

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^{§¶}

June 2, 2020

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. In order to accurately discretize it while maintaining the simplicity of finite difference discretizations, the domain is described using a level-set technique. 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 scheme 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 [22]. 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 [51], 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 [8]; 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 [15] or treat appropriately the heterogeneity of the crust layer [46]. Modelling of layered material with moving interfaces has been exploited also for copper corrosion in [14].

^{*}(Corresponding author) School of Engineering, Computing and Mathematics, Oxford Brookes University, Wheatley campus, Oxford

[†]Dipartimento di Matematica, Università di Torino, via C. Alberto, 10, Torino

[‡]Centro Speciale di Scienze e Simbolica dei Beni Culturali, Università dell'Insubria, Chiostro di S. Abbondio, via Regina Teodolinda, Como

[§]Department of Humanities and Innovation, Università dell'Insubria, via Bossi 5, 22100 Como

[¶]Centro Speciale di Scienze e Simbolica dei Beni Culturali, Università dell'Insubria, Chiostro di S. Abbondio, via Regina Teodolinda, Como

In this paper we focus on the model of [4], since it is quite simple but yet contains the more relevant numerical difficulties. Previous numerical work on this model have considered the one-dimensional version of the model [33] or Cartesian grids in two space dimensions [52]. 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 [52], the computational model have shown the importance of two-dimensional effects near corners and other sharp features of the domain. In a real case Ω 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 Ω by small cuboids, whose size and shape are eventually adapted close to the boundary of Ω in such a way that the most external corners lie on the boundary. This approach is adopted by the Shortley-Weller discretization [54], providing the simplest approach falling under the class of boundary-fitted grid methods, where the grid is suitably adapted to the boundary of the domain. Although the Shortley-Weller discretization is designed for Dirichlet boundary conditions and can be highly accurate for the solution and its gradient [53], the extension to Neumann boundary conditions is not straightforward.

More accurate and well-known boundary-fitted methods are represented by the Finite Element Methods (FEM), successfully adopted in several scientific contexts (e.g., [10, 11, 36, 6, 9, 41, 38, 27, 37, 23]). However, fitting the mesh to a complex domain with several corners or highly variable curvature might be computationally demanding. Another alternative for working with a Cartesian structure is furnished by the Isogeometric approach, adopted for example in [57, 40], 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 [15, 46, 14].

The first methods falling under this category were the Immersed Boundary Method [50] and the Immersed Interface Methods [43], proposed to model blood flows in the heart.

More recent numerical approaches to discretise partial differential equations on complex domains in a Cartesian grid are the Ghost-Fluid Methods proposed in [29, 34, 35, 45], where the solution and the Dirichlet boundary condition are extrapolated to define ghost values outside the domain in order to maintain a standard discretization stencil on internal grid points without compromising the overall accuracy order.

An improved version that accounts for Neumann boundary conditions was proposed by Coco and Russo in [19, 21, 18]. In this method a high order accuracy is achieved not only for the solution but also for the gradient of the solution.

Other recent methods for Neumann boundary conditions were proposed in [39], while the accuracy order for the gradient of the solution has been improved in [7].

Among the other unfitted-boundary approaches, we mention the matched interface (MIB) method [58], the Immersed Finite Volume Methods (IFVM) [47, 28], the arbitrary Lagrangian Eulerian method (ALE) [30, 26], and the penalization methods [42, 3, 12].

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 [52, 24, 25] and on the level-set domain description and handling of boundary conditions of [19]. As in [52], 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 [24]. 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 $\varphi : \mathbb{R}^N \rightarrow \mathbb{R}$ is known at least at the grid nodes. The grid nodes are defined, according to φ , as internal (those inside Ω), ghosts (first layer of points around the internal

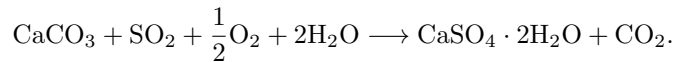
ones) and external. The method of [24] is applied only on the internal grid points. In order to close the method, the resulting nonlinear system of equations is augmented, as in [19], by the equations expressing the fulfillment of the boundary conditions on $\partial\Omega$ in terms of the ghost values and of their first internal neighbour points. The resulting system is then solved by Newton-Raphson and the special smoothing technique of [19] 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



to account for the transformation of 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 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:

$$\left\{ \begin{array}{ll} \frac{\partial (\phi(c)s)}{\partial t} = -\frac{a}{m_c} \phi(c) s c + d \nabla \cdot (\phi(c) \nabla s) & \text{in } \Omega \times [0, T] \\ \frac{\partial c}{\partial t} = -\frac{a}{m_s} \phi(c) s c & \text{in } \Omega \times [0, T] \\ s(\mathbf{x}, t) = s_b & \text{for } (\mathbf{x}, t) \in \partial\Omega \times [0, T] \\ s(\mathbf{x}, 0) = s_0(\mathbf{x}) & \text{for } \mathbf{x} \in \Omega \\ c(\mathbf{x}, 0) = c_0(\mathbf{x}) & \text{for } \mathbf{x} \in \Omega. \end{array} \right. \quad (1)$$

where $c(t, \mathbf{x})$ represents the marble concentration, initially set to c_0 , and $s(t, \mathbf{x})$ is the gas concentration, initially set to s_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 Ω , representing the pollution level of the surrounding air. It is assumed that, the gypsum concentration is $1 - c(t, \mathbf{x})$ and so, as time goes by, the calcium carbonate concentration is reduced from the initial value c_0 , as CaCO_3 is progressively replaced by a gypsum crust that forms on the outer shell of the monument and whose thickness and evolution are of interest to the managers of cultural heritage.

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

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

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 equation is the boundary condition for s and describes the condition surrounding the work of art. We assume for simplicity a Dirichlet boundary condition, although the method proposed in this paper can be extended to more sophisticated boundary conditions on fluxes such as in the free boundary model proposed in 1D in [15]. We observe that no boundary condition is needed for c , since the second equation of (1) does not involve spatial derivatives. Last two equations represent the initial conditions.

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 Ω and to describe the exact location and outward normal for the boundary of Ω .

3 Numerical method

3.1 Time discretization

Equations (1) are discretized in time using the second order accurate Crank-Nicolson scheme

$$\left\{ \begin{array}{ll} \frac{\phi(c^{(n+1)})s^{(n+1)} - \phi(c^{(n)})s^{(n)}}{\Delta t} = \frac{\mathcal{L}^s(s^{(n)}, c^{(n)}) + \mathcal{L}^s(s^{(n+1)}, c^{(n+1)})}{2} & \text{in } \Omega \\ \frac{c^{(n+1)} - c^{(n)}}{\Delta t} = \frac{\mathcal{L}^c(s^{(n)}, c^{(n)}) + \mathcal{L}^c(s^{(n+1)}, c^{(n+1)})}{2} & \text{in } \Omega \\ s^{(n+1)} = s_b & \text{on } \partial\Omega \\ s^{(0)} = s_0 \\ c^{(0)} = c_0 \end{array} \right. \quad (2)$$

where

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

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

3.2 Level-set function

The domain Ω is represented by an auxiliary level-set function $\varphi: D \rightarrow \mathbb{R}$:

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

Level-set methods have been firstly proposed by Sethian and Osher [49] and widely used since then to solve PDEs in complex-shaped geometries and moving domains [48]. The level-set function is convenient to gather geometric information, such as the outward unit normal vector \mathbf{n} and the curvature κ of the boundary Γ :

$$\mathbf{n} = \frac{\nabla \varphi}{|\nabla \varphi|}, \quad \kappa = \nabla \cdot \mathbf{n}. \quad (3)$$

Among the infinite level-set functions that can describe a domain Ω , the signed distance function $\phi_d(\mathbf{x}) = \pm \text{dist}(\mathbf{x}, \Gamma)$ has the advantage to maintain a stable algorithm since steep or shallow gradients are avoided. If the signed distance function ϕ_d is not available, it can be approximated from a generic level-set function ϕ by approximating the solution of the following PDE up to steady state (reinitialization step):

$$\frac{\partial \phi_d}{\partial \mu} = \text{sgn}(\phi) (1 - |\nabla \phi_d|), \quad \phi_d = \phi \quad \text{when } \mu = 0.$$

where μ is a fictitious time parameter [48]. The numerical methods proposed in this paper are designed for generic level-set function ϕ and therefore do not rely on the availability of the signed-distance function.

We observe that in real applications the level-set function must be derived from image data, usually starting from a cloud of points given by a 3D laser scanner. The level-set function can be obtained as follows. First, a provisional level-set function is defined by setting $\phi = -1$ inside the domain and $\phi = 1$ outside the domain, introducing a stepwise effects on the boundary. Then, the level-set function is slightly diffused (few iterations of diffusion equation), and the stepwise effects on the boundary disappear without affecting the important features of the monument. In this paper, we adopt the simplified 2D version of this technique to pixelated black and white images representing a woman head profile, a shark and a necklace (see §4.3).

3.3 Spatial discretization

Discretization in space is performed by a standard finite-difference scheme for the grid points lying inside the domain Ω 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 $D_h = \{(x_i, y_j) \in \mathbb{R}^2 : x_i = -L + ih, y_j = -L + jh, \text{ for } i, j = 0, \dots, N\}$. Let $\Omega_h = \Omega \cap D_h$ be the discrete counterpart of Ω .

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:

$$(x_i, y_j) \notin \Omega_h, \quad \{(x_i \pm h, y_j), (x_i, y_j \pm h)\} \cap \Omega_h \neq \emptyset. \quad (4)$$

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

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

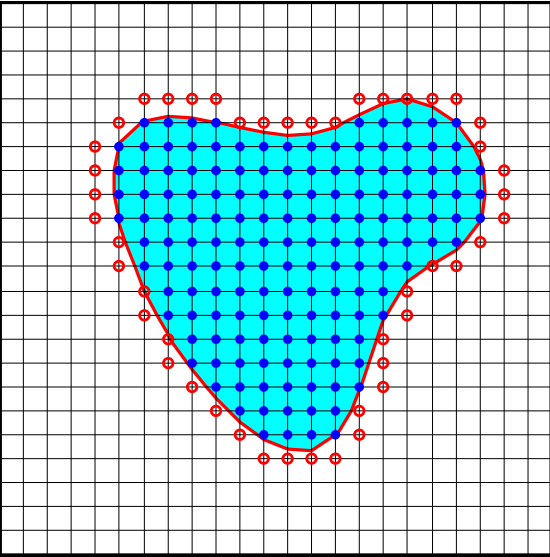


Fig. 1: Internal grid points Ω_h (blu filled circle) and ghost points Γ_h (red empty circle), according to the definition of ghost points (4).

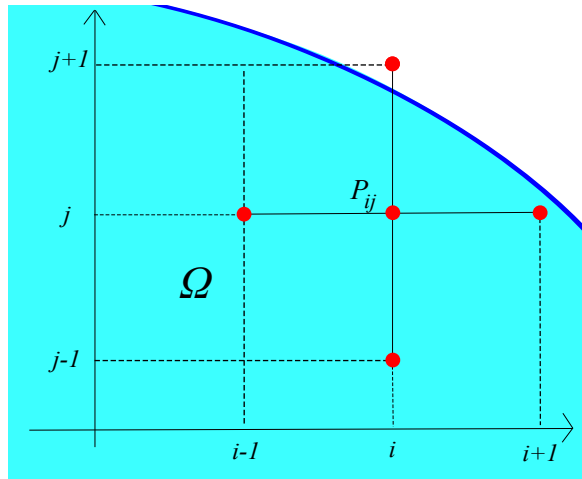


Fig. 2: Five-point stencil adopted for the finite-difference discretization on inside grid points (6).

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

The components of W are ordered by choosing a mapping

$$\mathcal{M}: \{1, 2, \dots, N_i + N_g\} \longrightarrow \mathcal{I}_\Omega \cup \mathcal{I}_\Gamma. \quad (5)$$

For the purpose of describing the numerical method, we order all s variables before the c ones, i.e. $W^{(n)} = (\mathbf{s}^{(n)}, \mathbf{c}^{(n)}) \in \mathbb{R}^{2(N_i + N_g)}$. Of course, the actual layout of the vector in the computer memory will be chosen to achieve optimal efficiency, e.g. like $W = (s_1, c_1, s_2, c_2, \dots, s_{N_i}, c_{N_i}, \dots, s_{N_i + N_g}, c_{N_i + N_g})$.

In order to compute $W^{(n+1)}$ from $W^{(n)}$, a system of $2(N_i + N_g)$ nonlinear equations in $2(N_i + N_g)$ unknowns must be solved at each time step. The nonlinear system is obtained as follows. The $2N_i$ 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:

$$\frac{\phi(c_{ij}^{(n+1)}) s_{ij}^{(n+1)} - \phi(c_{ij}^{(n)}) s_{ij}^{(n)}}{\Delta t} - \frac{\mathcal{L}_h^s(\mathbf{s}^{(n)}, \mathbf{c}^{(n)}) + \mathcal{L}_h^s(\mathbf{s}^{(n+1)}, \mathbf{c}^{(n+1)})}{2} = 0, \quad (6)$$

$$\frac{c_{ij}^{(n+1)} - c_{ij}^{(n)}}{\Delta t} - \frac{\mathcal{L}_h^c(\mathbf{s}^{(n)}, \mathbf{c}^{(n)}) + \mathcal{L}_h^c(\mathbf{s}^{(n+1)}, \mathbf{c}^{(n+1)})}{2} = 0 \quad (7)$$

where

$$\mathcal{L}_h^s(\mathbf{s}, \mathbf{c}) = -\frac{a}{m_c} \phi(c_{ij}) s_{ij} c_{ij} + \frac{d}{2} \sum_{(i^*, j^*) \in \mathcal{N}_{ij}} (\phi(c_{i^* j^*}) + \phi(c_{ij})) (s_{i^* j^*} - s_{ij})$$

$$\mathcal{L}_h^c(\mathbf{s}, \mathbf{c}) = -\frac{a}{m_s} \phi(c_{ij}) s_{ij} c_{ij},$$

and \mathcal{N}_{ij} is the set of four neighbouring index pairs for (i, j) , namely:

$$\mathcal{N}_{ij} = \{(i \pm 1, j), (i, j \pm 1)\}.$$

Eq. (7) is discretized also on ghost points (provided that the initial conditions for c and s are extrapolated outside the domain with second order accuracy, using for example the extrapolation technique proposed in [5]), leading to additional N_g nonlinear equations. The remaining N_g equations are obtained by enforcing the boundary condition for s on Γ

$$s^{(n+1)} = s_b \text{ on } \partial\Omega \quad (8)$$

using a ghost point extrapolation technique that was already successfully adopted in other contexts (elliptic [19, 21, 18, 17, 16] and hyperbolic [13, 20] equations) and described as follows. Let $G_{ij} = (x_i, y_j) \in \Gamma_h$ be a ghost point and $\mathbf{n}_{ij} = (n_x, n_y)$ be the approximated outward unit normal vector computed by a central finite-difference discretization of Eq. (3):

$$n_x = \frac{\tilde{n}_x}{\sqrt{\tilde{n}_x^2 + \tilde{n}_y^2}}, \quad n_y = \frac{\tilde{n}_y}{\sqrt{\tilde{n}_x^2 + \tilde{n}_y^2}}, \quad \text{with} \quad \tilde{n}_x = \frac{\varphi_{i+1,j} - \varphi_{i-1,j}}{2h}, \quad \tilde{n}_y = \frac{\varphi_{i,j+1} - \varphi_{i,j-1}}{2h}, \quad \varphi_{ij} = \varphi(x_i, y_j). \quad (9)$$

Let ST_{ij} be the nine-point stencil in the upwind direction with respect to \mathbf{n}_{ij} . More precisely,

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

where $m_x = \text{SIGN}(n_x)$ and $m_y = \text{SIGN}(n_y)$, with $\text{SIGN}(x) = 1$ or -1 if $x > 0$ or $x < 0$, respectively, and (conventionally) $\text{SIGN}(0) = 1$. Since this nine-point stencil is taken in the upwind direction with respect to

the normal \mathbf{n}_{ij} , it can be easily proven that $ST_9 \subseteq \Omega_h \cup \Gamma_h$, provided that the grid is sufficiently fine (i.e. h is sufficiently small).

The linear equations are obtained by prescribing the boundary condition (8) on $B_{ij} \in \Gamma$, where B_{ij} is the normal projection of G_{ij} onto Γ , obtained by the following algorithm (see Fig. 3):

- compute $P_{ij} = G_{ij} - 2h \mathbf{n}_{ij}$ (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 φ on the stencil ST_{ij} ;
- compute $B_{ij} = P_{ij} + \alpha(G_{ij} - P_{ij})$.

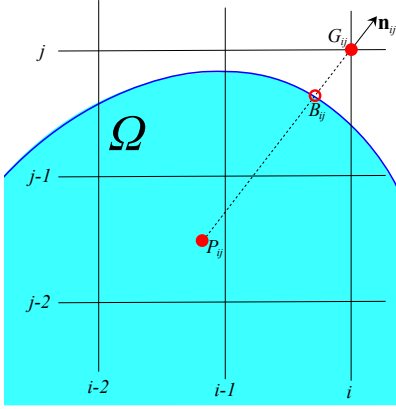


Fig. 3: Boundary point B_{ij} obtained from the ghost point G_{ij} by solving $\varphi(P_{ij} + \alpha(G_{ij} - P_{ij})) = 0$ for $\alpha \in [0, 1]$, where $P_{ij} = G_{ij} - 2h \mathbf{n}_{ij}$.

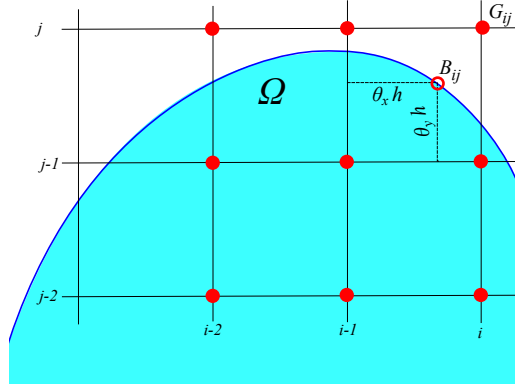


Fig. 4: Nine-point stencil ST_9 (red filled circles) associated with the ghost point G_{ij} to discretize the boundary condition on B_{ij} by Eq. (15).

Finally, the linear equations are:

$$s_b - \tilde{s}^{(n+1)}(B_{ij}) = 0, \quad (10)$$

where $\tilde{s}^{(n+1)}$ is the biquadratic interpolations of $\mathbf{s}^{(n+1)}$ on the stencil ST_{ij} .

We observe that when only Dirichlet boundary conditions are considered, other boundary discretization approaches can result more efficient and easier to implement, such as Shortley-Weller methods [54] or the symmetric positive-definite discretization proposed in [34]. However, the method proposed here is robust for all types of boundary conditions (Dirichlet, Neumann, Robin, mixed) and an extension to Neumann boundary conditions is proposed in §3.6 and tested in §4.1 (TEST 1N AND TEST 2N), making the approach suitable for an extension to more complex models of monument conservation, such as the free boundary models proposed in 1D in [15], where a non-homogenous free boundary conditions involving fluxes is proposed to account for the swelling of the external gypsum layer, The 3D implementation is currently under investigation.

3.4 Newton-Raphson method

Eqs. (6), (7), (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), (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 $k = 1, \dots$, repeat the steps:

(a) solve the linear system

$$J_F(W^{(n+1,k)}) \cdot \Delta W = F(W^{(n+1,k)}), \quad (11)$$

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.:

$$\min \left\{ \left\| F(W^{(n+1,k+1)}) \right\|_{\infty}, \frac{\left\| W^{(n+1,k+1)} - W^{(n+1,k)} \right\|_{\infty}}{\left\| W^{(n+1,k)} \right\|_{\infty}} \right\} < 10^{-9}.$$

The Jacobian matrix can be represented in the compact form

$$J_F(W^{(n+1,k)}) = \begin{pmatrix} J^{ss}(W^{(n+1,k)}) & J^{sc}(W^{(n+1,k)}) \\ J^{cs}(W^{(n+1,k)}) & J^{cc}(W^{(n+1,k)}) \end{pmatrix}, \quad (12)$$

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 3×3 stencils in the following way: we say that the stencil

$$\begin{bmatrix} a_{-1,1} & a_{0,1} & a_{1,1} \\ a_{-1,0} & \boxed{a_{0,0}} & a_{1,0} \\ a_{-1,-1} & a_{0,-1} & a_{1,-1} \end{bmatrix}$$

represents the r -th row of a $(N_i + N_g) \times (N_i + N_g)$ matrix A , with $\mathcal{M}(r) = (i, j)$, if for any $1 \leq q \leq N_i + N_g$ we have that

$$A_{rq} = \begin{cases} a_{k_i, k_j} & \text{if } \mathcal{M}(q) = (i + k_i, j + k_j), \text{ with } k_i, k_j \in \{-1, 0, 1\} \\ 0 & \text{otherwise.} \end{cases}$$

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 3×3 stencil. Therefore, we use for instance the stencil

$$\begin{bmatrix} \boxed{a_{0,0}} & a_{1,0} & a_{2,0} \\ a_{0,-1} & a_{1,-1} & a_{2,-1} \\ a_{0,-2} & a_{1,-2} & a_{2,-2} \end{bmatrix}$$

if we want to represent the r -th row of A as:

$$A_{rq} = \begin{cases} a_{k_i, k_j} & \text{if } \mathcal{M}(q) = (i + k_i, j + k_j), \text{ with } k_i \in \{0, 1, 2\}, k_j \in \{-2, -1, 0\} \\ 0 & \text{otherwise.} \end{cases}$$

Representation of the matrix $J^{ss}(W^{(n+1,k)})$. Using this notation, if $\mathcal{M}(r) \in \mathcal{I}_{\Omega}$ the r -th row of $J^{ss}(W^{(n+1,k)})$ is:

$$\begin{bmatrix} 0 & & 0 \\ 0 & \boxed{\frac{\phi(c_{ij}^{(n+1)})}{\Delta t} + \frac{a}{2m_c} \phi(c_{ij}^{(n+1)}) c_{ij}^{(n+1)}} & 0 \\ 0 & & 0 \end{bmatrix}$$

$$+\frac{d}{4} \begin{bmatrix} 0 & -\phi\left(c_{i,j+1}^{(n+1)}\right) + \phi\left(c_{ij}^{(n+1)}\right) & 0 \\ -\phi\left(c_{i-1,j}^{(n+1)}\right) + \phi\left(c_{ij}^{(n+1)}\right) & \boxed{\sum_{(i^*,j^*) \in \mathcal{N}_{ij}} \left(\phi\left(c_{i^*,j^*}^{(n+1)}\right) + \phi\left(c_{ij}^{(n+1)}\right)\right)} & -\phi\left(c_{i+1,j}^{(n+1)}\right) + \phi\left(c_{ij}^{(n+1)}\right) \\ 0 & -\frac{d}{2} \left(\phi\left(c_{i,j-1}^{(n+1)}\right) + \phi\left(c_{ij}^{(n+1)}\right)\right) & 0 \end{bmatrix}. \quad (13)$$

Let $\mathcal{M}(r) \in \mathcal{I}_\Gamma$ and $n_x \geq 0$, $n_y \geq 0$ (see Fig. 4 and Eq. (9)). Referring to Fig. 5, observe that the three coefficients

$$\left[\frac{\vartheta(\vartheta-1)}{2}, \quad (1-\vartheta)(1+\vartheta), \quad \frac{\vartheta(1+\vartheta)}{2} \right] \quad (14)$$

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

The 2D biquadratic interpolation (15) is obtained as dimension by dimension quadratic interpolations. Therefore, the r -th row of $J^{ss}(W^{(n+1,k)})$ is represented by

$$\begin{bmatrix} \boxed{\frac{\vartheta_y(1+\vartheta_y)}{2}} \\ (1-\vartheta_y)(1+\vartheta_y) \\ \frac{\vartheta_y(\vartheta_y-1)}{2} \end{bmatrix} \cdot \left[\frac{\vartheta_x(\vartheta_x-1)}{2} \quad (1-\vartheta_x)(1+\vartheta_x) \quad \boxed{\frac{\vartheta_x(1+\vartheta_x)}{2}} \right] =$$

$$\begin{bmatrix} \frac{\vartheta_x(\vartheta_x-1)\vartheta_y(1+\vartheta_y)}{4} & \frac{(1-\vartheta_x)(1+\vartheta_x)\vartheta_y(1+\vartheta_y)}{2} & \boxed{\frac{\vartheta_x(1+\vartheta_x)\vartheta_y(1+\vartheta_y)}{4}} \\ \frac{\vartheta_x(\vartheta_x-1)(1-\vartheta_y)(1+\vartheta_y)}{2} & (1-\vartheta_x)(1+\vartheta_x)(1-\vartheta_y)(1+\vartheta_y) & \frac{\vartheta_x(1+\vartheta_x)(1-\vartheta_y)(1+\vartheta_y)}{2} \\ \frac{\vartheta_x(\vartheta_x-1)\vartheta_y(\vartheta_y-1)}{4} & \frac{(1-\vartheta_x)(1+\vartheta_x)\vartheta_y(\vartheta_y-1)}{2} & \frac{\vartheta_x(1+\vartheta_x)\vartheta_y(\vartheta_y-1)}{4} \end{bmatrix}, \quad (15)$$

with

$$(\vartheta_x, \vartheta_y) = \frac{|B_{ij} - (x_{i+1}, y_{j-1})|}{h}.$$

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.

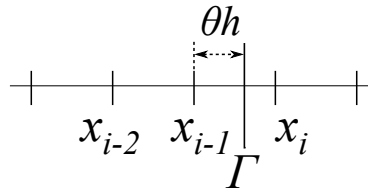


Fig. 5: One-dimensional interpolation on Γ using nodes x_{i-2} , x_{i-1} and x_i (Eqs. (14) and (26)).

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

$$\begin{aligned}
& \begin{bmatrix} 0 & & 0 \\ 0 & \boxed{\frac{\phi' \left(c_{ij}^{(n+1)} \right) s_{ij}^{(n+1)}}{\Delta t} + \frac{a}{2m_c} \left(\phi' \left(c_{ij}^{(n+1)} \right) s_{ij}^{(n+1)} c_{ij}^{(n+1)} + \phi \left(c_{ij}^{(n+1)} \right) s_{ij}^{(n+1)} \right)} & 0 \\ 0 & & 0 \end{bmatrix} \\
& + \frac{d}{4} \sum_{(i^*, j^*) \in \mathcal{N}_{ij}} \left(s_{ij}^{(n+1)} - s_{i^* j^*}^{(n+1)} \right) \begin{bmatrix} 0 & \phi' \left(c_{i,j+1}^{(n+1)} \right) & 0 \\ \phi' \left(c_{i-1,j}^{(n+1)} \right) & \boxed{\phi' \left(c_{i,j}^{(n+1)} \right)} & \phi' \left(c_{i+1,j}^{(n+1)} \right) \\ 0 & \phi' \left(c_{i,j-1}^{(n+1)} \right) & 0 \end{bmatrix} \quad (16)
\end{aligned}$$

if $\mathcal{M}(r) \in \mathcal{I}_\Omega$, or a null row if $\mathcal{M}(r) \in \mathcal{I}_\Gamma$.

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

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & \boxed{\frac{a}{m_s} \phi \left(c_{ij}^{(n+1)} \right) s_{ij}^{(n+1)} c_{ij}^{(n+1)}} & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (17)$$

both if $\mathcal{M}(r) \in \mathcal{I}_\Omega$ or $\mathcal{M}(r) \in \mathcal{I}_\Gamma$.

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

$$\begin{bmatrix} 0 & & 0 \\ 0 & \boxed{\frac{1}{\Delta t} + \frac{a}{m_s} \left(\phi' \left(c_{ij}^{(n+1)} \right) s_{ij}^{(n+1)} c_{ij}^{(n+1)} + \phi \left(c_{ij}^{(n+1)} \right) s_{ij}^{(n+1)} \right)} & 0 \\ 0 & & 0 \end{bmatrix}. \quad (18)$$

both if $\mathcal{M}(r) \in \mathcal{I}_\Omega$ or $\mathcal{M}(r) \in \mathcal{I}_\Gamma$.

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

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

$$J^{cs}(W^{(n+1,k)}) = D_{cs}(W^{(n+1,k)}), \quad J^{cc}(W^{(n+1,k)}) = D_{cc}(W^{(n+1,k)}),$$

where $D_{ss}(W^{(n+1,k)})$, $D_{sc}(W^{(n+1,k)})$, $D_{cs}(W^{(n+1,k)})$, $D_{cc}(W^{(n+1,k)})$ are four $(N_i + N_g) \times (N_i + N_g)$ diagonal matrices, $M_{ss}(W^{(n+1,k)})$, $M_{sc}(W^{(n+1,k)})$ are two $(N_i + N_g) \times (N_i + N_g)$ penta-diagonal matrices, while R_{ss} is a $(N_i + N_g) \times (N_i + N_g)$ nine-diagonal matrix that does 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 technique, as described in this section. In particular, we will introduce the relaxation operator (§3.5.1) and the transfer (restriction and interpolation) operators (§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 [56]). 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 [56, 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:

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

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 [56, 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 ([56, Ch. 8]). In this paper we implement a collective Gauss-Seidel scheme, i.e. a 2×2 linear system is solved at each internal grid point to update Δs_{ij} and Δc_{ij} simultaneously, while an appropriate relaxation is performed on ghost points ([19, 21, 18]). In detail, the relaxation scheme spans all $r = 1, \dots, N_i + N_g$. If $\mathcal{M}(r) = (i, j) \in \mathcal{I}_\Omega$, then Δs_{ij} and Δc_{ij} are updated as follows:

$$\begin{bmatrix} (\Delta s^{(m+1)})_r \\ (\Delta c^{(m+1)})_r \end{bmatrix} = \begin{bmatrix} (\Delta s^{(m)})_r \\ (\Delta c^{(m)})_r \end{bmatrix} + P^{-1} \cdot \begin{bmatrix} F^s(W^{(n+1,k)})_r - (J^{ss}(W^{(n+1,k)})_r \cdot \Delta \mathbf{s}^* + J^{sc}(W^{(n+1,k)})_r \cdot \Delta \mathbf{c}^*) \\ F^c(W^{(n+1,k)})_r - (J^{cs}(W^{(n+1,k)})_r \cdot \Delta \mathbf{s}^* + J^{cc}(W^{(n+1,k)})_r \cdot \Delta \mathbf{c}^*) \end{bmatrix},$$

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 $\Delta \mathbf{s}^*$ and $\Delta \mathbf{c}^*$ the current approximations of the Gauss-seidel iteration, i.e. they are vectors whose q -th component is defined by:

$$(\Delta \mathbf{s}^*)_q = \begin{cases} (\Delta \mathbf{s}^{(m+1)})_q & \text{if } 1 \leq q < r \\ (\Delta \mathbf{s}^{(m)})_q & \text{if } r \leq q \leq N_i + N_g \end{cases}, \quad (\Delta \mathbf{c}^*)_q = \begin{cases} (\Delta \mathbf{c}^{(m+1)})_q & \text{if } 1 \leq q < r \\ (\Delta \mathbf{c}^{(m)})_q & \text{if } r \leq q \leq N_i + N_g \end{cases},$$

and

$$P = \begin{bmatrix} (J^{ss}(W^{(n+1,k)}))_{rr} & (J^{sc}(W^{(n+1,k)}))_{rr} \\ (J^{cs}(W^{(n+1,k)}))_{rr} & (J^{cc}(W^{(n+1,k)}))_{rr} \end{bmatrix}. \quad (19)$$

We observe that the classical (pointwise) Gauss-Seidel scheme can be obtained by replacing the matrix (19) with

$$P = \begin{bmatrix} (J^{ss}(W^{(n+1,k)}))_{rr} & 0 \\ 0 & (J^{cc}(W^{(n+1,k)}))_{rr} \end{bmatrix}. \quad (20)$$

If $\mathcal{M}(r) = (i, j) \in \mathcal{I}_\Gamma$, then Δs_{ij} and Δc_{ij} are updated as follows:

$$\left(\Delta s^{(m+1)} \right)_r = \left(\Delta s^{(m)} \right)_r + \tau^s \left(F^s(W^{(n+1,k)})_r - J^{ss}(W^{(n+1,k)})_r \cdot \Delta \mathbf{s}^* \right) \quad (21)$$

$$\begin{aligned} \left(\Delta_{\mathbf{c}}^{(m+1)}\right)_r &= \left(\Delta_{\mathbf{c}}^{(m)}\right)_r \\ &+ \left(J^{cc}(W^{(n+1,k)})\right)_{rr}^{-1} \left(F^c(W^{(n+1,k)})_r - \left(J^{cs}(W^{(n+1,k)})_r \cdot \Delta \mathbf{s}^* + J^{cc}(W^{(n+1,k)})_r \cdot \Delta \mathbf{c}^*\right)\right), \end{aligned}$$

where $J^{\{ss\}}(W^{(n+1,k)})_r$ is the r -th row of the matrix $J^{\{ss\}}(W^{(n+1,k)})$ (see Eqs. (15)). If we choose τ^s as in (20) to have a Gauss-Seidel iteration, i.e.

$$\tau^s = \left(\frac{\vartheta_x(1+\vartheta_x)\vartheta_y(1+\vartheta_y)}{4}\right)^{-1},$$

the relaxation scheme may not converge (we observe for example that rows (15) are not diagonally dominant). Following the idea proposed in [19, 21, 18], the parameter τ^s is chosen in such a way that a proper CFL condition is satisfied for the iterations (21). In particular, we want to ensure that the absolute value of the coefficient of $(\Delta(s)^{(m)})_r$ in the right-hand side of (21) is smaller than one, i.e.

$$\left|1 - \tau^s \frac{\vartheta_x(1+\vartheta_x)\vartheta_y(1+\vartheta_y)}{4}\right| < 1 \quad (22)$$

We choose a parameter τ^s that satisfies the condition (22) for any $0 \leq \vartheta_x, \vartheta_y \leq 1$. This is achieved by: $0 < \tau^s < 1$. For practical purposes, we use $\tau^s = 0.9$.

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 ν_1 pre-relaxation iterations (§3.5.1), we compute the defects \mathbf{r}_h^s and \mathbf{r}_h^c :

$$\begin{aligned} \mathbf{r}_h^s &= F^s(W^{(n+1,k)}) - (J^{ss}(W^{(n+1,k)}) \cdot \Delta \mathbf{s}^{(\nu_1)} + J^{sc}(W^{(n+1,k)}) \cdot \Delta \mathbf{c}^{(\nu_1)}) \\ \mathbf{r}_h^c &= F^c(W^{(n+1,k)}) - (J^{cs}(W^{(n+1,k)}) \cdot \Delta \mathbf{s}^{(\nu_1)} + J^{cc}(W^{(n+1,k)}) \cdot \Delta \mathbf{c}^{(\nu_1)}) \end{aligned}$$

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

$$\mathbf{r}_{2h}^s = \mathcal{I}_{2h}^{h(s)} \mathbf{r}_h^s, \quad \mathbf{r}_{2h}^c = \mathcal{I}_{2h}^{h(c)} \mathbf{r}_h^c.$$

We observe that the defect \mathbf{r}_h^s is discontinuous across the boundary, because the defect on \mathcal{I}_Ω is related to the internal equations, while its values on \mathcal{I}_Γ are referred to the boundary conditions. For this reason, the restriction of the internal equations must use values only from \mathcal{I}_Ω . (see [19]) To this purpose, the restriction operator $\mathcal{I}_{2h}^{h(s)}$ is appropriately modified for internal grid points in the vicinity of $\Gamma = \partial\Omega$. We recall the full-weighting restriction operator (see [56, Ch. 2.3.3]):

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

In general, by the stencil notation

$$\mathcal{I}_{2h}^h = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & s_{-1,-1} & s_{-1,0} & s_{-1,1} & \cdots \\ \cdots & s_{0,-1} & s_{0,0} & s_{0,1} & \cdots \\ \cdots & s_{1,-1} & s_{1,0} & s_{1,1} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}_{2h}^h \quad (23)$$

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

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

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

Following the same idea of [19, 21], we modify the restriction operator close to the boundary in order to disregard the values of the defects on ghost points (see Fig. 6 and [19, 21] for more details).

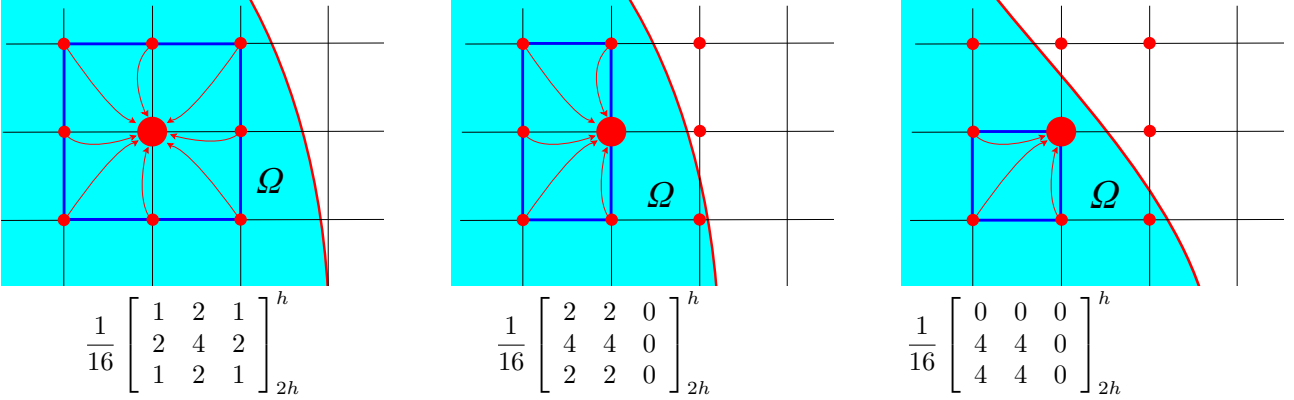


Fig. 6: Top: nine point stencil (small circles) around a grid point on the coarse grid (bold circle). The rectangle is a visual representation of the fine grid points requested in the restriction operator. Bottom: the respective stencils of the restriction operator in matrix form (Eq. (23)).

The restriction of the boundary condition 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 defect \mathbf{r}_h^s is firstly defined in the neighbouring inactive points by extrapolating the ghost value constantly along the normal direction to the boundary Γ . This can be achieved by solving the transport equations

$$\frac{\partial \mathbf{r}_h^s}{\partial \tau} + \nabla \mathbf{r}_h^s \cdot \mathbf{n} = 0.$$

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

3.5.3 Interpolation operator

The defect equations

$$\begin{bmatrix} J^{ss}(W^{(n+1,k)}) & J^{sc}(W^{(n+1,k)}) \\ J^{cs}(W^{(n+1,k)}) & J^{cc}(W^{(n+1,k)}) \end{bmatrix} \cdot \begin{bmatrix} \mathbf{e}_{2h}^s \\ \mathbf{e}_{2h}^c \end{bmatrix} = \begin{bmatrix} \mathbf{r}_{2h}^s \\ \mathbf{r}_{2h}^c \end{bmatrix}$$

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:

$$\mathbf{e}_h^s = \mathcal{I}_h^{2h} \mathbf{e}_{2h}^s, \quad \mathbf{e}_h^c = \mathcal{I}_h^{2h} \mathbf{e}_{2h}^c.$$

Since errors \mathbf{e}_{2h}^s and \mathbf{e}_{2h}^c are continuous across the boundary, we can use values from both sides of the boundary in the same stencil and therefore we can adopt a standard linear interpolation operator for all points:

$$\mathcal{I}_h^{2h} = \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \begin{matrix} \left[\right. \\ \\ \left. \right]_{2h} \end{matrix}.$$

Finally, ν_2 post-relaxation iterations (§3.5.1) are performed on the finer grid.

3.6 Neumann boundary conditions

In this section we extend the numerical approach to the case of Neumann boundary conditions for s . The implementation of more sophisticated boundary conditions (Robin, mixed, etc.) will then be straightforward. We consider the Neumann boundary condition

$$\frac{\partial s^{(n+1)}}{\partial n} = s_b \text{ on } \partial\Omega$$

instead of (8).

The spatial discretization (10) becomes:

$$\left(\nabla \tilde{s}^{(n+1)} \cdot \frac{\nabla \tilde{\varphi}}{|\nabla \tilde{\varphi}|} \right) \Big|_{B_{ij}} = s_b(B_{ij}), \quad (24)$$

where $\tilde{s}^{(n+1)}$ and $\tilde{\varphi}$ are the the biquadratic interpolations of $\mathbf{s}^{(n+1)}$ and φ , respectively, on the stencil ST_{ij} . The r -th row of $J^{ss}(W^{(n+1,k)})$ when $\mathcal{M}(r) \in \mathcal{I}_\Gamma$ (Eq. (15)) is obtained by using the condition (24):

$$\tilde{n}_x \frac{\partial \tilde{s}}{\partial x} + \tilde{n}_y \frac{\partial \tilde{s}}{\partial y} = 0, \quad \text{with } \tilde{n}_x = \frac{\tilde{n}_x^*}{\sqrt{(\tilde{n}_x^*)^2 + (\tilde{n}_y^*)^2}}, \quad \tilde{n}_y = \frac{\tilde{n}_y^*}{\sqrt{(\tilde{n}_x^*)^2 + (\tilde{n}_y^*)^2}}, \quad \tilde{n}_x^* = \frac{\partial \tilde{\varphi}}{\partial x}, \quad \tilde{n}_y^* = \frac{\partial \tilde{\varphi}}{\partial y}. \quad (25)$$

Since the coefficients of the 1D quadratic approximation of the first derivative on Γ are (see Fig. 5)

$$\frac{1}{h} \left[\vartheta - \frac{1}{2}, \quad -2\vartheta, \quad \vartheta + \frac{1}{2} \right] \quad (26)$$

for grid points x_{i-2}, x_{i-1}, x_i , respectively, then the r -th row of $J^{ss}(W^{(n+1,k)})$ is represented by (from (25), see Fig. 4):

$$\begin{aligned} & \tilde{n}_x \begin{bmatrix} \boxed{\frac{\vartheta_y(1+\vartheta_y)}{2}} \\ (1-\vartheta_y)(1+\vartheta_y) \\ \boxed{\frac{\vartheta_y(\vartheta_y-1)}{2}} \end{bmatrix} \cdot \frac{1}{h} \left[\vartheta_x - \frac{1}{2} \quad -2\vartheta_x \quad \boxed{\vartheta_x + \frac{1}{2}} \right] \\ & + \tilde{n}_y \frac{1}{h} \begin{bmatrix} \boxed{\vartheta_y + \frac{1}{2}} \\ -2\vartheta_y \\ \boxed{\vartheta_y - \frac{1}{2}} \end{bmatrix} \cdot \left[\boxed{\frac{\vartheta_x(\vartheta_x-1)}{2}} \quad (1-\vartheta_x)(1+\vartheta_x) \quad \boxed{\frac{\vartheta_x(1+\vartheta_x)}{2}} \right] \end{aligned}$$

$$\begin{aligned}
&= \frac{\tilde{n}_x}{h} \begin{bmatrix} \frac{\vartheta_y(1+\vartheta_y)}{2} \left(\vartheta_x - \frac{1}{2} \right) & -\vartheta_x \vartheta_y (1+\vartheta_y) & \boxed{\frac{\vartheta_y(1+\vartheta_y)}{2} \left(\vartheta_x + \frac{1}{2} \right)} \\ (1-\vartheta_y)(1+\vartheta_y) \left(\vartheta_x - \frac{1}{2} \right) & -2\vartheta_x(1-\vartheta_y)(1+\vartheta_y) & (1-\vartheta_y)(1+\vartheta_y) \left(\vartheta_x + \frac{1}{2} \right) \\ \frac{\vartheta_y(\vartheta_y-1)}{2} \left(\vartheta_x - \frac{1}{2} \right) & -\vartheta_x \vartheta_y (\vartheta_y-1) & \frac{\vartheta_y(\vartheta_y-1)}{2} \left(\vartheta_x + \frac{1}{2} \right) \end{bmatrix} \\
&+ \frac{\tilde{n}_y}{h} \begin{bmatrix} \frac{\vartheta_x(\vartheta_x-1)}{2} \left(\vartheta_y + \frac{1}{2} \right) & \left(\vartheta_y + \frac{1}{2} \right) (1-\vartheta_x)(1+\vartheta_x) & \boxed{\frac{\vartheta_x(1+\vartheta_x)}{2} \left(\vartheta_y + \frac{1}{2} \right)} \\ -\vartheta_y \vartheta_x (\vartheta_x-1) & -2\vartheta_y(1-\vartheta_x)(1+\vartheta_x) & -\vartheta_y \vartheta_x (1+\vartheta_x) \\ \frac{\vartheta_x(\vartheta_x-1)}{2} \left(\vartheta_y - \frac{1}{2} \right) & (1-\vartheta_x)(1+\vartheta_x) \left(\vartheta_y - \frac{1}{2} \right) & \frac{\vartheta_x(1+\vartheta_x)}{2} \left(\vartheta_y - \frac{1}{2} \right) \end{bmatrix},
\end{aligned}$$

with

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

$$\begin{aligned}
\tilde{n}_x^* &= \frac{1}{h} \begin{bmatrix} \varphi_{i-2,j} & \varphi_{i-1,j} & \varphi_{i,j} \\ \varphi_{i-2,j-1} & \varphi_{i-1,j-1} & \varphi_{i,j-1} \\ \varphi_{i-2,j-2} & \varphi_{i-1,j-2} & \varphi_{i,j-2} \end{bmatrix} \odot \begin{bmatrix} \frac{\vartheta_y(1+\vartheta_y)}{2} \left(\vartheta_x - \frac{1}{2} \right) & -\vartheta_x \vartheta_y (1+\vartheta_y) & \frac{\vartheta_y(1+\vartheta_y)}{2} \left(\vartheta_x + \frac{1}{2} \right) \\ (1-\vartheta_y)(1+\vartheta_y) \left(\vartheta_x - \frac{1}{2} \right) & -2\vartheta_x(1-\vartheta_y)(1+\vartheta_y) & (1-\vartheta_y)(1+\vartheta_y) \left(\vartheta_x + \frac{1}{2} \right) \\ \frac{\vartheta_y(\vartheta_y-1)}{2} \left(\vartheta_x - \frac{1}{2} \right) & -\vartheta_x \vartheta_y (\vartheta_y-1) & \frac{\vartheta_y(\vartheta_y-1)}{2} \left(\vartheta_x + \frac{1}{2} \right) \end{bmatrix}, \\
\tilde{n}_y^* &= \frac{1}{h} \begin{bmatrix} \varphi_{i-2,j} & \varphi_{i-1,j} & \varphi_{i,j} \\ \varphi_{i-2,j-1} & \varphi_{i-1,j-1} & \varphi_{i,j-1} \\ \varphi_{i-2,j-2} & \varphi_{i-1,j-2} & \varphi_{i,j-2} \end{bmatrix} \odot \begin{bmatrix} \frac{\vartheta_x(\vartheta_x-1)}{2} \left(\vartheta_y + \frac{1}{2} \right) & \left(\vartheta_y + \frac{1}{2} \right) (1-\vartheta_x)(1+\vartheta_x) & \frac{\vartheta_x(1+\vartheta_x)}{2} \left(\vartheta_y + \frac{1}{2} \right) \\ -\vartheta_y \vartheta_x (\vartheta_x-1) & -2\vartheta_y(1-\vartheta_x)(1+\vartheta_x) & -\vartheta_y \vartheta_x (1+\vartheta_x) \\ \frac{\vartheta_x(\vartheta_x-1)}{2} \left(\vartheta_y - \frac{1}{2} \right) & (1-\vartheta_x)(1+\vartheta_x) \left(\vartheta_y - \frac{1}{2} \right) & \frac{\vartheta_x(1+\vartheta_x)}{2} \left(\vartheta_y - \frac{1}{2} \right) \end{bmatrix},
\end{aligned}$$

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

The parameter τ^s of Eq. (22) must satisfy the condition:

$$\left| 1 - \tau^s \left(\frac{|\tilde{n}_x|}{h} \frac{\vartheta_y(1+\vartheta_y)}{2} \left(\vartheta_x + \frac{1}{2} \right) + \frac{|\tilde{n}_y|}{h} \frac{\vartheta_x(1+\vartheta_x)}{2} \left(\vartheta_y + \frac{1}{2} \right) \right) \right| < 1. \quad (27)$$

We choose a parameter τ^s that satisfies the condition (27) 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: $0 < \tau^s < \frac{2\sqrt{2}h}{3}$. For practical purposes, we use $\tau^s = 0.9 \frac{2\sqrt{2}h}{3}$.

4 Numerical tests

In this section we confirm numerically the second order accuracy of the numerical method and we evaluate the efficiency of the multigrid approach. We choose the following parameters in Eq. (1) (see [52]):

$$a = 10^4, \quad d = 0.1, \quad m_s = 64.06, \quad m_c = 100.09, \quad \phi(c) = 0.1 + 0.01c. \quad (28)$$

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 8×8 grid as the coarsest grid.

4.1 Accuracy tests

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

$$\begin{cases} \frac{\partial(\phi(c)s)}{\partial t} = -\frac{a}{m_c}\phi(c)sc + d\nabla \cdot (\phi(c)\nabla s) + f_1 & \text{in } \Omega \times [0, T] \\ \frac{\partial c}{\partial t} = -\frac{a}{m_s}\phi(c)sc + f_2 & \text{in } \Omega \times [0, T] \\ s(\mathbf{x}, t) = s_b(\mathbf{x}, t) & \text{for } (\mathbf{x}, t) \in \partial\Omega \times [0, T] \\ s(\mathbf{x}, 0) = s_0(\mathbf{x}) & \text{for } \mathbf{x} \in \Omega \\ c(\mathbf{x}, 0) = c_0(\mathbf{x}) & \text{for } \mathbf{x} \in \Omega \end{cases}$$

We choose f_1, f_2, s_b in such a way that the exact solutions are:

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

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

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

and on the gradients

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

where s_h and c_h are the numerical solutions and ∇s_h and ∇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:

$$\varphi(x, y) = \sqrt{(x - x_0)^2 + (y - y_0)^2} - R, \quad \text{where } x_0 = \sqrt{2}/30, \quad y_0 = \sqrt{3}/40, \quad R = 1.486. \quad (29)$$

In TEST 2 the domain is represented by the union of a square and four circles centred on the vertices

$$\varphi(x, y) = \min \{ \varphi_1(x, y), \varphi_2(x, y) \} \quad (30)$$

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^∞ 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 repeat the two tests for the case of Neumann boundary conditions for s as described in §3.6) (TEST 1N and TEST 2N) and present the results in Tables 5, 6, 7 and 8 and in Figures 12, 13, 14 and 15. We note that in all cases second order convergence is achieved.

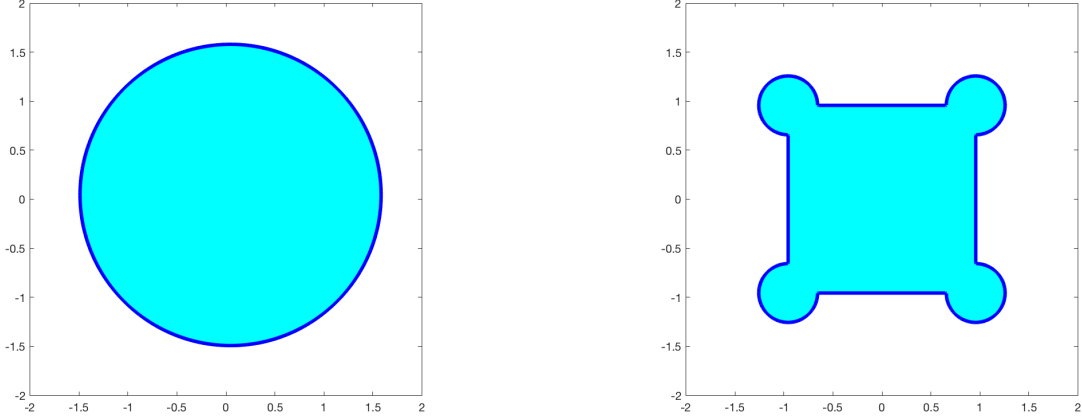


Fig. 7: Domain Ω for TEST 1, TEST 1N AND 3 (left) and TEST 2, TEST 2N AND 4 (right).

Table 1: TEST 1. 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^∞ error of s	order
16×16	$2.00 \cdot 10^{-6}$	-	$1.93 \cdot 10^{-5}$	-
32×32	$4.03 \cdot 10^{-7}$	2.31	$4.67 \cdot 10^{-6}$	2.05
64×64	$7.38 \cdot 10^{-8}$	2.45	$8.35 \cdot 10^{-7}$	2.48
128×128	$1.43 \cdot 10^{-8}$	2.37	$1.42 \cdot 10^{-7}$	2.56
256×256	$3.03 \cdot 10^{-9}$	2.24	$1.95 \cdot 10^{-8}$	2.86
No. of points	L^1 error of $ \nabla s $	order	L^∞ error of $ \nabla s $	order
16×16	$1.20 \cdot 10^{-3}$	-	$1.30 \cdot 10^{-3}$	-
32×32	$2.88 \cdot 10^{-4}$	2.06	$3.25 \cdot 10^{-4}$	2.00
64×64	$7.03 \cdot 10^{-5}$	2.03	$8.13 \cdot 10^{-5}$	2.00
128×128	$1.73 \cdot 10^{-5}$	2.02	$2.03 \cdot 10^{-5}$	2.00
256×256	$4.30 \cdot 10^{-6}$	2.01	$5.08 \cdot 10^{-6}$	2.00

4.2 Multigrid efficiency

In this section we solve Eq. (1) with data (28) and the following initial and Dirichlet boundary conditions:

$$s_0(\mathbf{x}) = 0, \quad c_0(\mathbf{x}) = 10, \quad s_B = 1. \quad (31)$$

We perform two tests: TEST 3 and TEST 4, with the domain represented by the level-sets (29) and (30), respectively (see Fig. 7).

Solutions at time $t = 1$ are plotted in Figs. 16 and 17. For each W-cycle of the multigrid method, we compute the convergence factor as:

$$\rho_{\text{MG}}^{(q)} = \frac{\left\| \mathbf{r}_h^{(q)} \right\|_\infty}{\left\| \mathbf{r}_h^{(q-1)} \right\|_\infty},$$

where

$$\mathbf{r}_h^{(q)} = \begin{bmatrix} \mathbf{r}_h^s \\ \mathbf{r}_h^c \end{bmatrix} = \begin{bmatrix} F^s(W^{(n+1,k)}) - (J^{ss}(W^{(n+1,k)}) \cdot \Delta \mathbf{s} + J^{sc}(W^{(n+1,k)}) \cdot \Delta \mathbf{c}) \\ F^c(W^{(n+1,k)}) - (J^{cs}(W^{(n+1,k)}) \cdot \Delta \mathbf{s} + J^{cc}(W^{(n+1,k)}) \cdot \Delta \mathbf{c}) \end{bmatrix}$$

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^∞ error of c	order
16×16	$2.04 \cdot 10^{-6}$	-	$1.92 \cdot 10^{-5}$	-
32×32	$4.60 \cdot 10^{-7}$	2.15	$5.10 \cdot 10^{-6}$	1.91
64×64	$8.52 \cdot 10^{-8}$	2.43	$1.12 \cdot 10^{-6}$	2.19
128×128	$1.63 \cdot 10^{-8}$	2.39	$1.98 \cdot 10^{-7}$	2.50
256×256	$3.45 \cdot 10^{-9}$	2.24	$2.80 \cdot 10^{-8}$	2.82
No. of points	L^1 error of $ \nabla c $	order	L^∞ error of $ \nabla c $	order
16×16	$6.36 \cdot 10^{-4}$	-	$6.89 \cdot 10^{-4}$	-
32×32	$1.60 \cdot 10^{-4}$	1.99	$2.08 \cdot 10^{-4}$	1.73
64×64	$4.06 \cdot 10^{-5}$	1.98	$5.74 \cdot 10^{-5}$	1.86
128×128	$1.02 \cdot 10^{-5}$	1.99	$1.52 \cdot 10^{-5}$	1.92
256×256	$2.57 \cdot 10^{-6}$	1.99	$3.92 \cdot 10^{-6}$	1.95

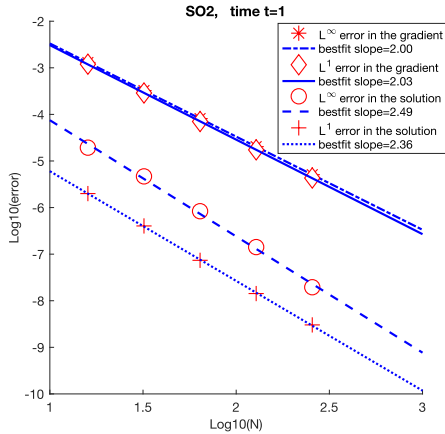


Fig. 8: TEST 1: bestfit lines of the errors in the solution and in the gradient for SO_2 in L^1 and L^∞ norms (Table 1).

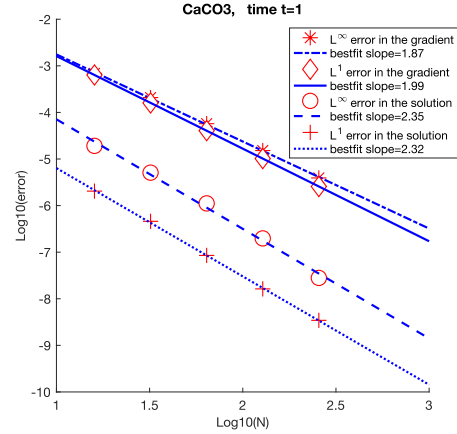


Fig. 9: TEST 1: bestfit lines of the errors in the solution and in the gradient for CaCO_3 in L^1 and L^∞ norms (Table 2).

is the defect after q W-cycles. Convergence factors are plotted in Fig. 18 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 [56, 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 use data (28) and (31) and 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. 19, 21 and 23, respectively, while zooms on some relevant regions,

Table 3: TEST 2. Accuracy order in the solution (top) and in the gradient (bottom) for s (SO₂).

No. of points	L^1 error of s	order	L^∞ error of s	order
16×16	$1.01 \cdot 10^{-4}$	-	$1.16 \cdot 10^{-3}$	-
32×32	$1.83 \cdot 10^{-5}$	2.47	$1.35 \cdot 10^{-4}$	3.11
64×64	$2.88 \cdot 10^{-6}$	2.67	$8.83 \cdot 10^{-5}$	0.61
128×128	$1.65 \cdot 10^{-7}$	4.13	$2.36 \cdot 10^{-6}$	5.22
256×256	$1.10 \cdot 10^{-7}$	0.59	$1.59 \cdot 10^{-7}$	3.89
No. of points	L^1 error of $ \nabla s $	order	L^∞ error of $ \nabla s $	order
16×16	$1.12 \cdot 10^{-3}$	-	$2.81 \cdot 10^{-3}$	-
32×32	$2.86 \cdot 10^{-4}$	1.97	$6.87 \cdot 10^{-4}$	2.03
64×64	$7.49 \cdot 10^{-5}$	1.93	$7.64 \cdot 10^{-4}$	-0.15
128×128	$1.75 \cdot 10^{-5}$	2.10	$5.31 \cdot 10^{-5}$	3.85
256×256	$4.30 \cdot 10^{-6}$	2.03	$6.92 \cdot 10^{-6}$	2.94

Table 4: TEST 2. Accuracy order in the solution (top) and in the gradient (bottom) for c (CaCO₃).

No. of points	L^1 error of c	order	L^∞ error of c	order
16×16	$7.20 \cdot 10^{-5}$	-	$1.20 \cdot 10^{-3}$	-
32×32	$4.13 \cdot 10^{-6}$	4.12	$6.76 \cdot 10^{-5}$	4.15
64×64	$7.37 \cdot 10^{-7}$	2.49	$1.15 \cdot 10^{-5}$	2.55
128×128	$1.10 \cdot 10^{-7}$	2.74	$1.50 \cdot 10^{-6}$	2.94
256×256	$4.96 \cdot 10^{-8}$	1.15	$5.57 \cdot 10^{-7}$	1.43
No. of points	L^1 error of $ \nabla c $	order	L^∞ error of $ \nabla c $	order
16×16	$7.20 \cdot 10^{-4}$	-	$1.86 \cdot 10^{-3}$	-
32×32	$1.66 \cdot 10^{-4}$	2.12	$2.94 \cdot 10^{-4}$	2.66
64×64	$4.19 \cdot 10^{-5}$	1.99	$8.68 \cdot 10^{-5}$	1.76
128×128	$1.05 \cdot 10^{-5}$	1.99	$2.27 \cdot 10^{-5}$	1.94
256×256	$2.67 \cdot 10^{-6}$	1.98	$6.35 \cdot 10^{-6}$	1.84

with the contour plot of CaCO₃ at times $t = 0.25$, $t = 0.5$, $t = 0.75$ and $t = 1$, are represented in Figs. 20, 22 and 24, 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. 20), 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. 20). The gypsum formation is also quicker around reentrant corners, as we can see in the bottom-left and bottom-right plots of Fig. 20, for example. Similar conclusions may be drawn for the shark in TEST 6, where the teeth (top-right plot of Fig. 22) and the secondary dorsal fin (top-left plot of Fig. 22) are transformed quickly. The damage on the caudal fin is quicker around the tips (top-left plot of Fig. 22) and around the reentrant corner of the pectoral fin (bottom-left plot of Fig. 22) and of the primary dorsal fin (bottom-right plot of Fig. 22). 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

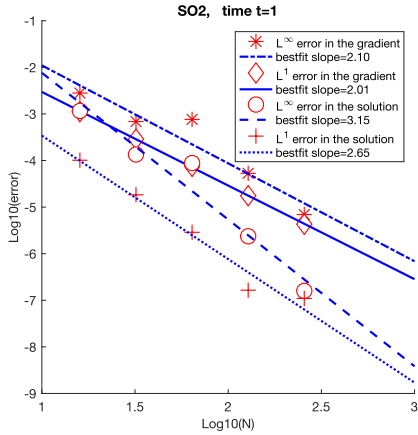


Fig. 10: TEST 2: bestfit lines of the errors in the solution and in the gradient for SO_2 in L^1 and L^∞ norms (Table 3).

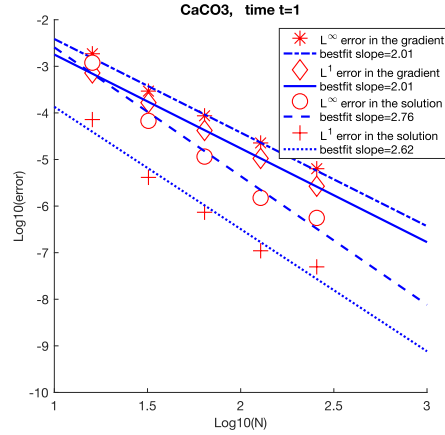


Fig. 11: TEST 2: bestfit lines of the errors in the solution and in the gradient for CaCO_3 in L^1 and L^∞ norms (Table 4).

Table 5: TEST 1N. 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^∞ error of s	order
16×16	$7.43 \cdot 10^{-5}$	-	$9.10 \cdot 10^{-5}$	-
32×32	$1.53 \cdot 10^{-5}$	2.28	$2.41 \cdot 10^{-5}$	1.92
64×64	$2.14 \cdot 10^{-6}$	2.84	$4.79 \cdot 10^{-6}$	2.33
128×128	$1.45 \cdot 10^{-7}$	3.88	$6.37 \cdot 10^{-7}$	2.91
256×256	$1.11 \cdot 10^{-7}$	0.39	$1.59 \cdot 10^{-7}$	2.00
No. of points	L^1 error of $ \nabla s $	order	L^∞ error of $ \nabla s $	order
16×16	$1.19 \cdot 10^{-3}$	-	$1.29 \cdot 10^{-3}$	-
32×32	$2.89 \cdot 10^{-4}$	2.04	$3.26 \cdot 10^{-4}$	1.98
64×64	$7.01 \cdot 10^{-5}$	2.04	$8.11 \cdot 10^{-5}$	2.01
128×128	$1.73 \cdot 10^{-5}$	2.02	$2.03 \cdot 10^{-5}$	2.00
256×256	$4.31 \cdot 10^{-6}$	2.01	$5.44 \cdot 10^{-6}$	1.90

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. 23 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).

Table 6: TEST 1N. 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^∞ error of c	order
16×16	$1.82 \cdot 10^{-5}$	-	$5.37 \cdot 10^{-5}$	-
32×32	$3.00 \cdot 10^{-6}$	2.60	$1.02 \cdot 10^{-5}$	2.40
64×64	$5.63 \cdot 10^{-7}$	2.41	$5.85 \cdot 10^{-6}$	0.80
128×128	$8.83 \cdot 10^{-8}$	2.67	$1.50 \cdot 10^{-6}$	1.96
256×256	$4.53 \cdot 10^{-8}$	0.96	$4.59 \cdot 10^{-7}$	1.71
No. of points	L^1 error of $ \nabla c $	order	L^∞ error of $ \nabla c $	order
16×16	$6.35 \cdot 10^{-4}$	-	$6.88 \cdot 10^{-4}$	-
32×32	$1.60 \cdot 10^{-4}$	1.99	$2.09 \cdot 10^{-4}$	1.72
64×64	$4.06 \cdot 10^{-5}$	1.98	$5.92 \cdot 10^{-5}$	1.82
128×128	$1.03 \cdot 10^{-5}$	1.98	$1.92 \cdot 10^{-5}$	1.63
256×256	$2.62 \cdot 10^{-6}$	1.97	$6.02 \cdot 10^{-6}$	1.67

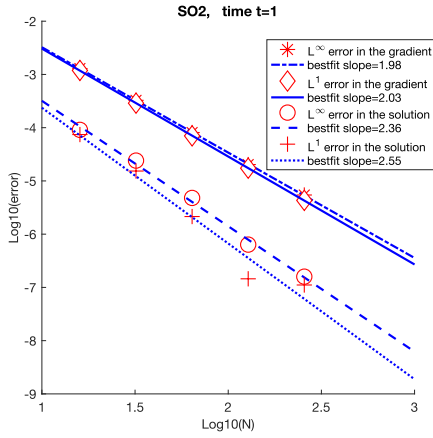


Fig. 12: TEST 1N: bestfit lines of the errors in the solution and in the gradient for SO_2 in L^1 and L^∞ norms (Table 5).

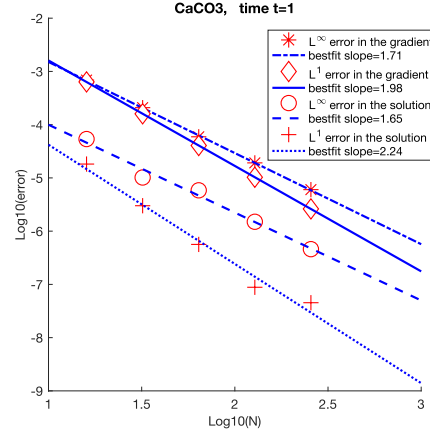


Fig. 13: TEST 1N: bestfit lines of the errors in the solution and in the gradient for CaCO_3 in L^1 and L^∞ norms (Table 6).

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 [31, 32] 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. [15]) or that include internal moving interfaces among layers of materials with different properties (e.g. [14, 46]), to take into account some effects like swelling and corrosion. 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. 23.

Table 7: TEST 2N. 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^∞ error of s	order
16×16	$1.16 \cdot 10^{-4}$	-	$1.74 \cdot 10^{-3}$	-
32×32	$2.70 \cdot 10^{-5}$	2.10	$7.48 \cdot 10^{-4}$	1.22
64×64	$3.18 \cdot 10^{-6}$	3.09	$7.79 \cdot 10^{-5}$	3.26
128×128	$1.74 \cdot 10^{-7}$	4.19	$2.64 \cdot 10^{-6}$	4.89
256×256	$1.07 \cdot 10^{-7}$	0.69	$1.57 \cdot 10^{-7}$	4.07
No. of points	L^1 error of $ \nabla s $	order	L^∞ error of $ \nabla s $	order
16×16	$1.16 \cdot 10^{-3}$	-	$3.77 \cdot 10^{-3}$	-
32×32	$3.32 \cdot 10^{-4}$	1.81	$2.86 \cdot 10^{-3}$	0.40
64×64	$7.89 \cdot 10^{-5}$	2.07	$7.05 \cdot 10^{-4}$	2.02
128×128	$1.81 \cdot 10^{-5}$	2.12	$5.91 \cdot 10^{-5}$	3.58
256×256	$4.44 \cdot 10^{-6}$	2.03	$7.03 \cdot 10^{-6}$	3.07

Table 8: TEST 2N. 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^∞ error of c	order
16×16	$8.57 \cdot 10^{-5}$	-	$1.82 \cdot 10^{-3}$	-
32×32	$8.35 \cdot 10^{-6}$	3.36	$4.00 \cdot 10^{-4}$	2.19
64×64	$8.09 \cdot 10^{-7}$	3.37	$1.16 \cdot 10^{-5}$	5.11
128×128	$1.23 \cdot 10^{-7}$	2.71	$1.61 \cdot 10^{-6}$	2.85
256×256	$5.32 \cdot 10^{-8}$	1.22	$5.62 \cdot 10^{-7}$	1.52
No. of points	L^1 error of $ \nabla c $	order	L^∞ error of $ \nabla c $	order
16×16	$7.53 \cdot 10^{-4}$	-	$3.51 \cdot 10^{-3}$	-
32×32	$1.81 \cdot 10^{-4}$	2.05	$1.48 \cdot 10^{-3}$	1.24
64×64	$4.22 \cdot 10^{-5}$	2.11	$9.41 \cdot 10^{-5}$	3.98
128×128	$1.06 \cdot 10^{-5}$	1.99	$2.44 \cdot 10^{-5}$	1.94
256×256	$2.68 \cdot 10^{-6}$	1.98	$6.84 \cdot 10^{-6}$	1.84

From a computational point of view, the method will be extended to three dimensional problems, where a realistic piece of work will be modelled from a laser scanner 3D reconstruction. To reduce the computational cost, the computational strategy will include an Adaptive Meshing Refinement (AMR) approach, where the Cartesian mesh will be refined around critical regions such as the boundary and/or the internal moving interface, by extending AMR strategies already developed for simpler problems [44, 55].

Acknowledgments

The work has been partially supported by the London Mathematical Society Computer Science Small Grants – Scheme 7 (Ref. SC7-1617-02) and the Research in Pairs – Scheme 4 (Ref. 41739).

References

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

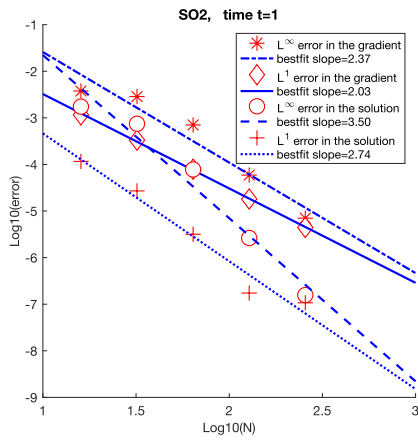


Fig. 14: TEST 2N: bestfit lines of the errors in the solution and in the gradient for SO_2 in L^1 and L^∞ norms (Table 7).

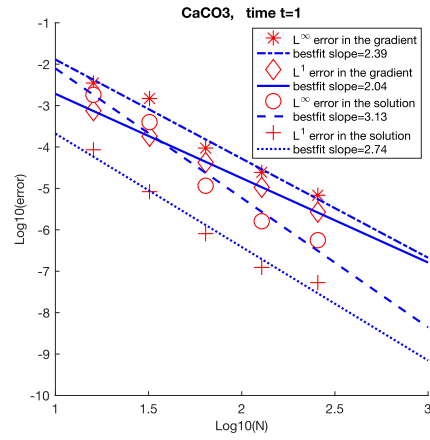


Fig. 15: TEST 2N: bestfit lines of the errors in the solution and in the gradient for CaCO_3 in L^1 and L^∞ norms (Table 8).

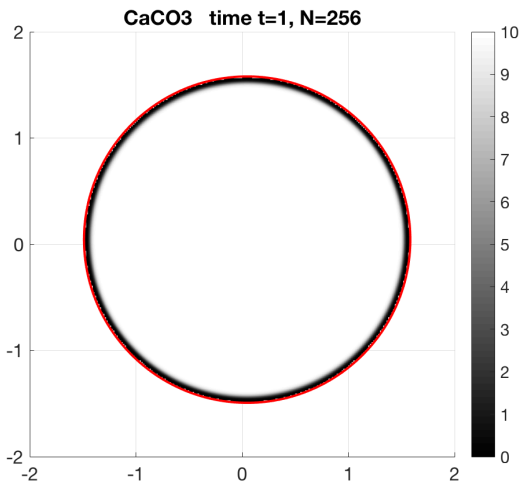
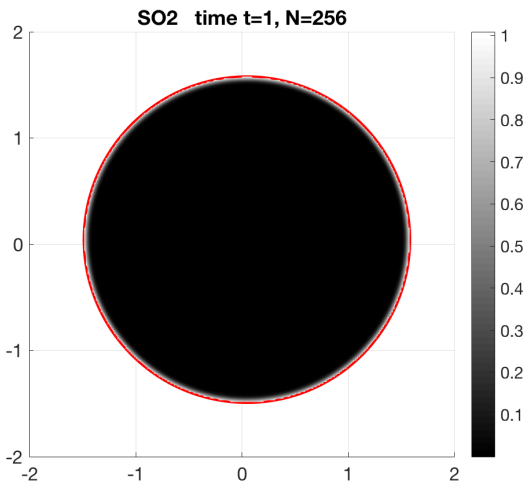


Fig. 16: Solutions for TEST 3 at time $t = 1$ with $N = 256$.

- [2] G. Alì, V. Furuholt, R. Natalini, and I. Torcicollo. 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.
- [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.

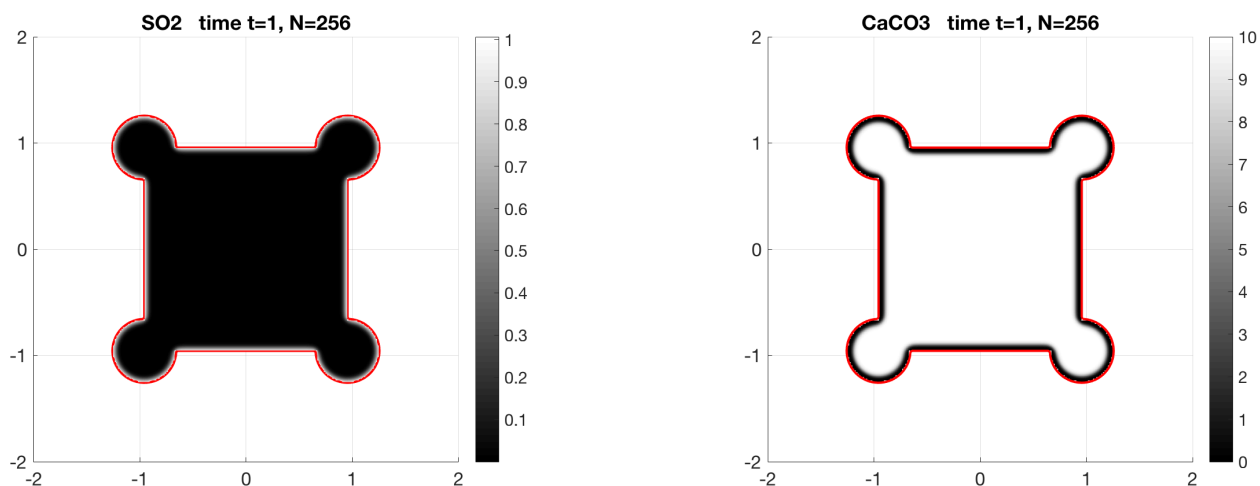


Fig. 17: Solutions for TEST 4 at time $t = 1$ with $N = 256$.

- [5] T. D. Aslam. A partial differential equation approach to multidimensional extrapolation. *J. Comput. Phys.*, 193(1):349–355, 2004.
- [6] I. Babuška. The finite element method for elliptic equations with discontinuous coefficients. *Comput.*, 5:207–213, 1970.
- [7] D. Bochkov and F. Gibou. Solving poisson-type equations with robin boundary conditions on piecewise smooth interfaces. *J. Comput. Phys.*, 376:1156–1198, 2019.
- [8] 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.
- [9] 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.
- [10] S. Brenner and R. Scott. *The mathematical theory of finite element methods*, volume 15. Springer Science & Business Media, 2007.
- [11] F. Brezzi and M. Fortin. *Mixed and hybrid finite element methods*, volume 15. Springer Science & Business Media, 2012.
- [12] 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.
- [13] 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.
- [14] F. Clarelli, B. De Filippo, and R. Natalini. Mathematical model of copper corrosion. *Appl. mathematical modelling*, 38(19-20):4804–4816, 2014.
- [15] 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.
- [16] 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.

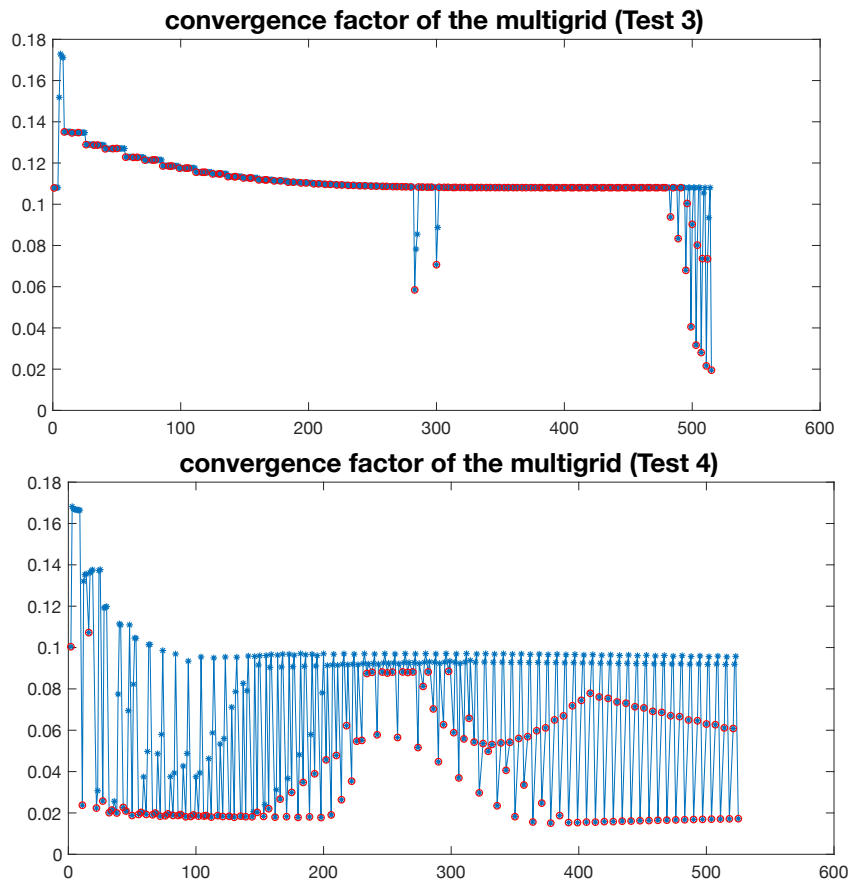


Fig. 18: Convergence factors versus the W-cycle iterations for TEST 3 (top) and TEST 4 (bottom). The first convergence factor obtained in each linear system (11) (either of the same Newton-Raphson step or a new time step) is circled (in red). The convergence factors of the first few linear systems are 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, showing that the multigrid efficiency has not been degraded by the non-rectangular domain and the ghost-point approach.

- [17] 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.
- [18] 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.
- [19] 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.
- [20] 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.

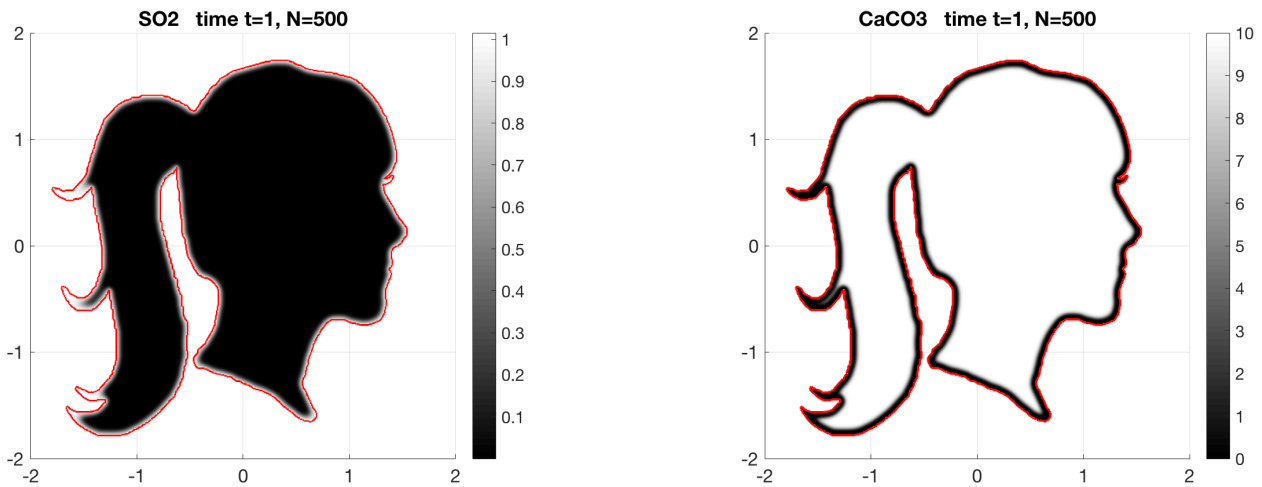


Fig. 19: Solutions for TEST 5 at time $t = 1$ with $N = 512$.

- [21] 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.
- [22] 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.
- [23] J. Dolbow and I. Harari. An efficient finite element method for embedded interface problems. *Int. J. for Num. Meth. Eng.*, 78(229–252), 2009.
- [24] 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.
- [25] 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.
- [26] J. Donea. An arbitrary Lagrangian-Eulerian finite element method for transient fluid-structure interactions. *Comput. Methods Appl. Mech. Eng.*, 33:689–723, 1982.
- [27] M. Dryja. A neumann-neumann algorithm for mortar discretization of elliptic problems with discontinuous coefficients. *Num. Math.*, 99(645–656), 2005.
- [28] 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.
- [29] 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.
- [30] 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.
- [31] C. Garoni and S. Serra-Capizzano. *Generalized locally Toeplitz sequences: theory and applications. Vol. I.* Springer, Cham, New York, 2017.

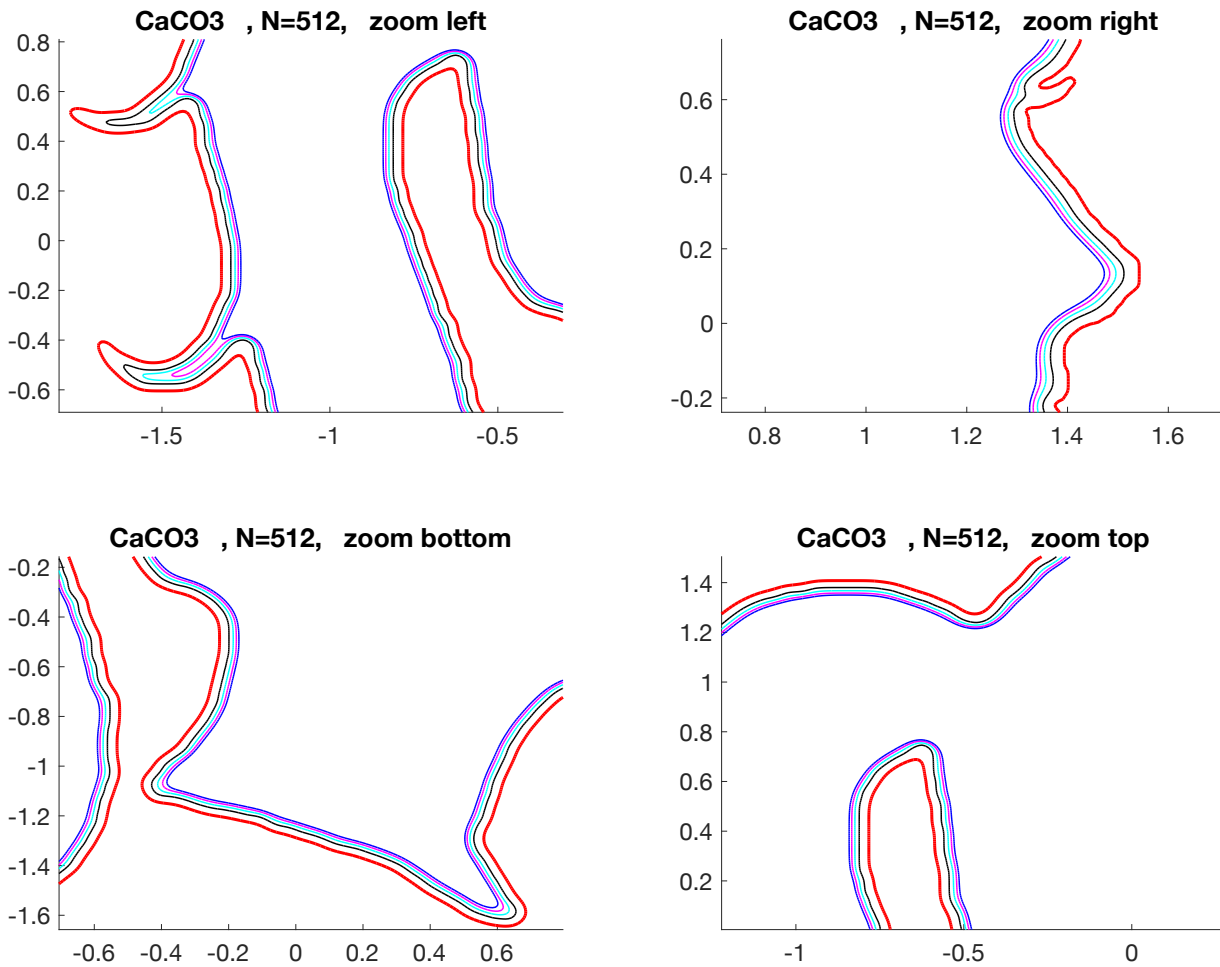


Fig. 20: Contour plot of $c = 5$ (CaCO_3) for TEST 5 at times $t = 0.25$, $t = 0.5$, $t = 0.75$ and $t = 1$. The boundary of the domain is the most external line, while the four contour lines can be easily recognised as they move away from the boundary as t increases. In the electronic version of the 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.

- [32] C. Garoni and S. Serra-Capizzano. *Generalized locally Toeplitz sequences: theory and applications. Vol. II*. Springer, Cham, New York, 2018.
- [33] 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.
- [34] 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.
- [35] 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.
- [36] V. Girault and P.-A. Raviart. *Finite element methods for Navier-Stokes equations: theory and algorithms*, volume 5. Springer Science & Business Media, 2012.

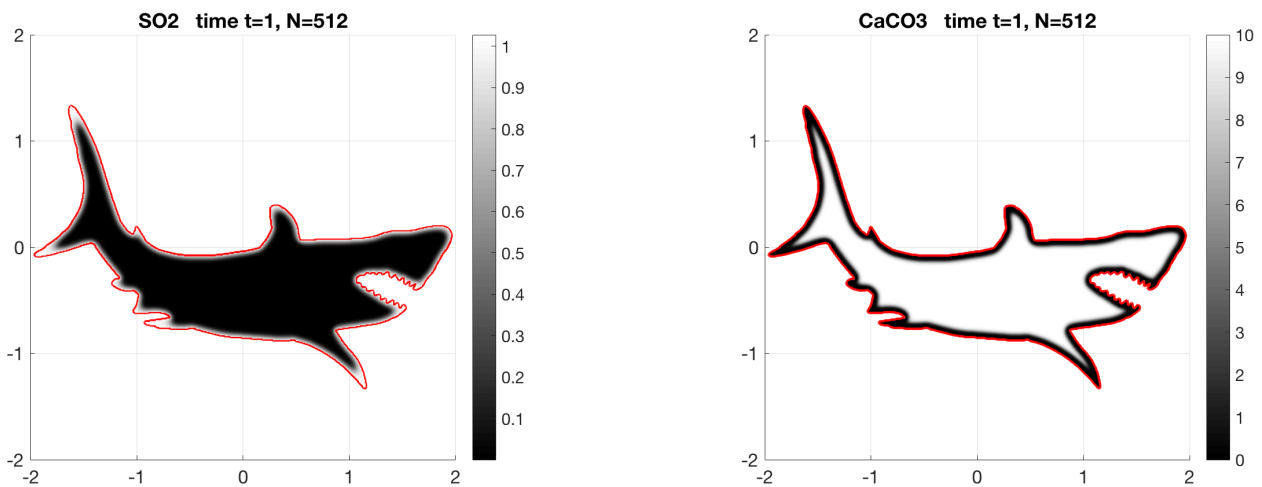


Fig. 21: Solutions for TEST 6 at time $t = 1$ with $N = 512$.

- [37] S. Groí and A. Reusken. An extended pressure finite element space for two-phase incompressible flows with surface tension. *J. Comput. Phys.*, 224(40–58), 2007.
- [38] 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.
- [39] 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.
- [40] C. Hesch and P. Betsch. Isogeometric analysis and domain decomposition methods. *Comput. Methods Appl. Mech. Eng.*, 213:104–112, 2012.
- [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] M. Mirzadeh, M. Theillard, and F. Gibou. A second-order discretization of the nonlinear poisson–boltzmann equation over irregular geometries using non-graded adaptive cartesian grids. *J. Comput. Phys.*, 230(5):2125–2140, 2011.
- [45] 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.
- [46] C. Nikolopoulos. Mathematical modelling of a mushy region formation during sulphation of calcium carbonate. *Netw. Heterog. Media*, 9(4):635–654, 2014.
- [47] 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.

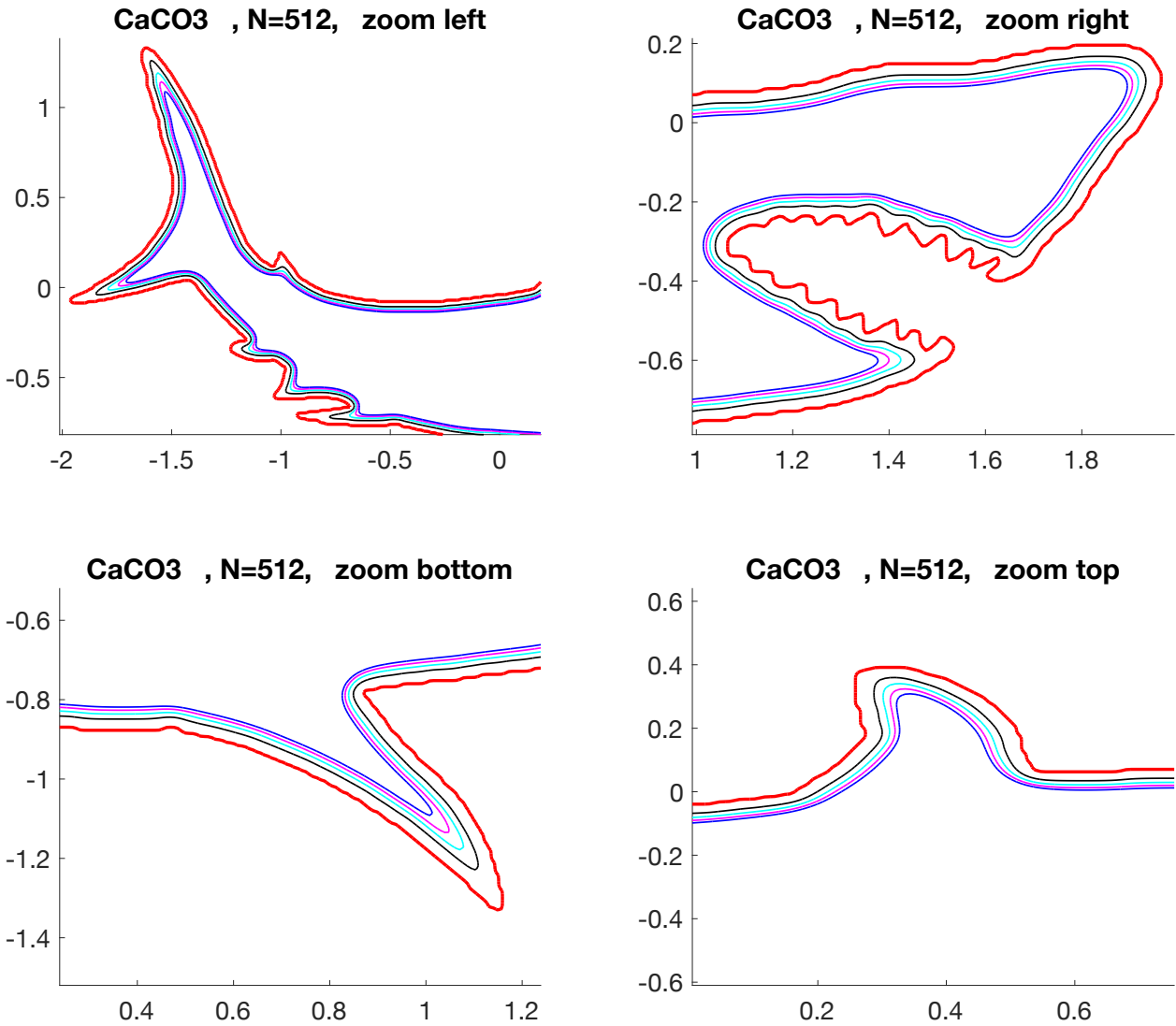


Fig. 22: Contour plot of $c = 5$ (CaCO₃) for TEST 6 at times $t = 0.25$, $t = 0.5$, $t = 0.75$ and $t = 1$. The boundary of the domain is the most external line, while the four contour lines can be easily recognised as they move away from the boundary as t increases. In the electronic version of the 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.

- [48] S. Osher and R. Fedkiw. *Level set methods and dynamic implicit surfaces*, volume 153. Springer Science & Business Media, 2006.
- [49] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *J. computational physics*, 79(1):12–49, 1988.
- [50] C. S. Peskin. Numerical analysis of blood flow in the heart. *J. Comput. Phys.*, 25:220–252, 1977.
- [51] M. Saba, E. Quiñones-Bolaños, and A. Barbosa López. A review of the mathematical models used for simulation of calcareous stone deterioration in historical buildings. *Atmospheric Environ.*, 180:156–166,

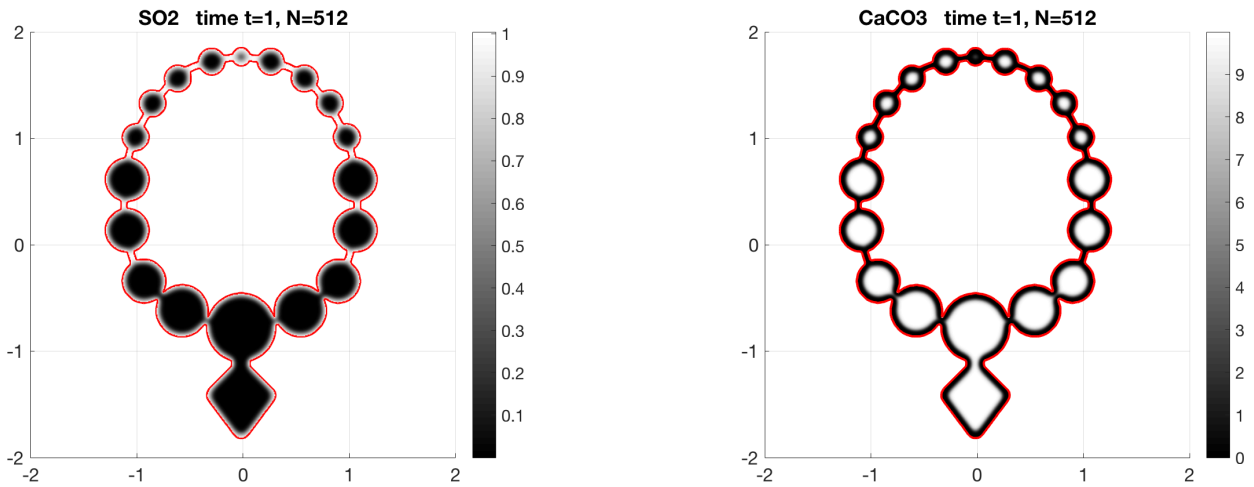


Fig. 23: Solutions for TEST 7 at time $t = 1$ with $N = 512$.

2018.

- [52] M. Semplice. Preconditioned implicit solvers for nonlinear PDEs in monument conservation. *SIAM J. Sci. Comput.*, 32(5):3071–3091, 2010.
- [53] J. Seo, S.-y. Ha, and C. Min. Convergence analysis in the maximum norm of the numerical gradient of the shortley–weller method. *J. Sci. Comput.*, 74(2):631–639, 2018.
- [54] G. H. Shortley and R. Weller. The numerical solution of Laplace’s equation. *J. Appl. Phys.*, 9:334–348, 1938.
- [55] M. Theillard, C. H. Rycroft, and F. Gibou. A multigrid method on non-graded adaptive octree and quadtree cartesian grids. *J. Sci. Comput.*, 55(1):1–15, 2013.
- [56] U. Trottemberg, C. Oosterlee, and A. Schuller. *Multigrid*. Academic Press, 2000.
- [57] 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.
- [58] 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.

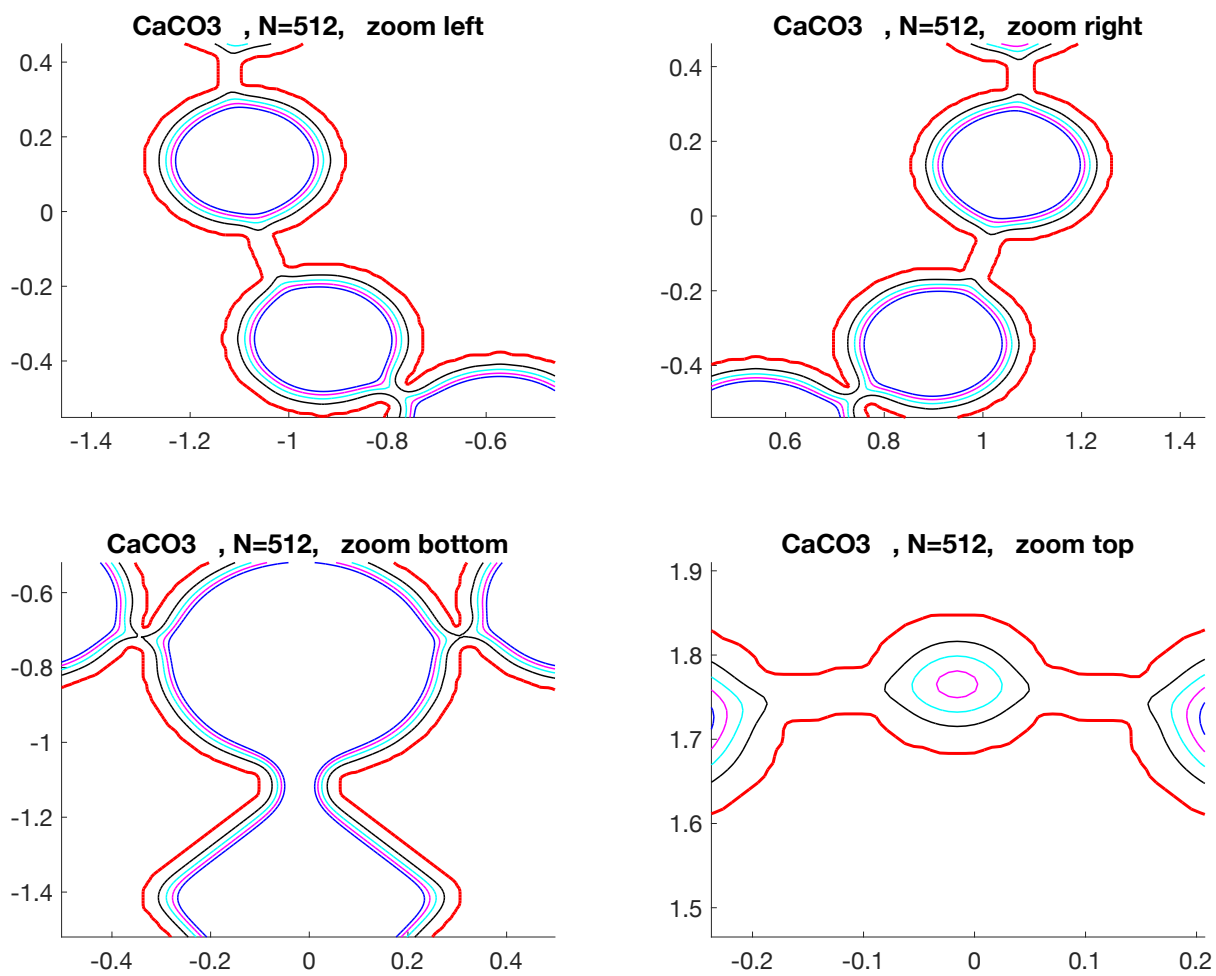


Fig. 24: Contour plot of $c = 5$ (CaCO₃) for TEST 7 at times $t = 0.25$, $t = 0.5$, $t = 0.75$ and $t = 1$. The boundary of the domain is the most external line, while the four contour lines can be easily recognised as they move away from the boundary as t increases. In the electronic version of the 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.