Journal of Computational Physics 230 (2011) 3230–3248

Contents lists available at ScienceDirect

Journal of Computational Physics

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

An ALE-based numerical technique for modeling sedimentary basin evolution featuring layer deformations and faults

Matteo Longoni [a], A. Cristiano I. Malossi [b], Alfio Quarteroni [a,b], Andrea Villa [c,*,1], Paolo Ruffo [d]

[a] MOX, Modeling and Scientific Computing, Department of Mathematics, Politecnico di Milano, Via Bonardi 9, Milano, Italy
[b] CMCS, Chair of Modelling and Scientific Computing, MATHICSE, Mathematics Institute of Computational Science and Engineering, École Polytechnique Fédérale de Lausanne, Station 8, CH-1015 Lausanne, Switzerland
[c] Università degli Studi di Milano, Department of Mathematics, Via Saldini 50, 20133 Milano, Italy
[d] ENI, Ente Nazionale Idrocarburi, E&P Division, GEBA Department, Via Emilia 1, 20097 San Donato Milanese (MI), Italy

ARTICLE INFO

Article history:
Received 4 February 2010
Received in revised form 18 January 2011
Accepted 18 January 2011
Available online 27 January 2011

Keywords:
Computational geology
Fractures and faults
Non-Newtonian fluids
Compaction of granular systems
Finite element method
Level set method

ABSTRACT

In this paper we present a numerical tool to simulate dynamics of stratified sedimentary basins, i.e. depressions on the Earth's surface filled by sediments. The basins are usually complicated by crustal deformations and faulting of the sediments. The balance equations, the non-Newtonian rheology of the sediments, and the depth-porosity compaction laws describe here a model of basin evolution. We propose numerical schemes for the basin boundary movement and for the fault tracking. In addition, a time splitting algorithm is employed to reduce the original model into some simpler mathematical problems. The numerical stability and the other features of the developed methodology are shown using simple test cases and some realistic configurations of sedimentary basins.

1. Introduction

The numerical simulation of the geological evolution of a sedimentary basin represents a challenging aspect of applied numerical mathematics, especially when a full three-dimensional (3D) model is employed. In order to deal with such a complex problem, in this work we set up a numerical framework aimed to reproduce the 3D dynamics of realistic stratified sedimentary basins. This work is the natural follow up of [1], where the main features of a mathematical model for simulating the evolution of salt sedimentary basins have been discussed and an efficient implementation has been proposed. In particular, in [1] the authors present a technique to describe accurately the interfaces between layers (usually referred to as horizons), based on a level set approach. On the other hand, in the present work we extend the methodology devised in [1] taking into account the non-Newtonian rheology of the sediments, the depth-porosity compaction laws, the basin boundary movement, and the handling of faults.

As regards the modeling counterpart, we consider well-known geological models coming from the literature, and collect them in a wider framework that simplifies their numerical treatment. The sediments are modeled through a classical stratified creeping flow problem, a widely diffused approach for such a problem (see, for instance, [1–5]) especially for its

[*] Corresponding author.
E-mail addresses: matteo.longoni@mate.polimi.it (M. Longoni), cristiano.malossi@epfl.ch (A.C.I. Malossi), alfio.quarteroni@epfl.ch (A. Quarteroni), andrea.villa81@fastwebnet.it, andrea.villa@erse-web.it (A. Villa), paolo.ruffo@eni.com (P. Ruffo).
[1] Present address: RSE, Ricerca Sistema Energetico, Via Rubattino 54, 20134 Milano, Italy.

0021-9991/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2011.01.027

flexibility. Indeed, it can be easily enriched with models representing the non-Newtonian rheology of the sediments, the faults, the compaction and the movement of the basement and of the free surface. Let us review all these aspects one by one.

A complete and exhaustive theoretical analysis of the sediments rheology is still missing. Experiments have shown different behaviors of the sediments, such as the elastic, elasto-plastic, visco-plastic and visco-elastic ones (see [6,7]). Semi-empirical relations are usually adopted, since the theory can explain only a few mechanisms (an example is the pressure solution, see [8]). Even though in some work (such as [9]) all the rheological aspects are taken into account, the typical approach is to consider the fluid behavior, and to add the plastic effects through a pseudo-plastic relation, in a power law form [7,8,10]. This is the approach we adopt in our work, since it has proven to be a good trade-off between accuracy and computational complexity.

Let us now consider the porosity and compaction modeling. In various works the compaction is modeled in a simplified way, for example, as a vertical reduction of the volume occupied by the sediments [11], or even neglected (see [2,5,9]). In [12] a new splitting algorithm is introduced: the divergence of the solid flow field is computed according to an experimental compaction curve. Then, under the hypothesis of vertical compaction, the problem is reduced to a linear stationary hyperbolic equation. Here, we do not make any assumption about the direction of the compaction, but we address ourselves directly to the modeling of the compaction function, i.e., the function that measures the rate of decrease of the solid volume and we solve a Stokes problem with a non-null divergence. We stress that, if the fluid part is not simulated, it is mandatory to consider some empirically derived compaction curves.

Fault modeling is seldom included in geological basin simulations and, until now, it has been considered only in a few works (see [4,9,12,13]). Fault location and time of appearance in the geological history are hardly predictable from the mechanical point of view, but fortunately seismic and well data are able to provide sufficiently accurate information. Hence, we assume to know the location and the time of appearance of the faults and we concentrate on the modeling of their effects. A possible way to model faults in a fluid framework is to reduce the fluid viscosity in the damaged area. For an active fault, the viscosity in a thin region around its sliding plane is reduced by several orders of magnitude (see [13]). This approach has two potential drawbacks: the identification of the elements in the grid where the viscosity has to be reduced and the grid local refinement necessary to make the element size match the fault thickness (which is of the order of tens of meters, while a typical mesh element is about hundreds of meters in a basin scale simulation). To face these problems we have implemented an innovative implicit tracking algorithm based on a level set function and a local recursive bisection algorithm. For a review of the mesh refinement techniques in three-dimensions see [14–18].

Let us now consider another important aspect, namely, the movement of the lateral basin boundary. The basement, the free surface, and the lateral contour are subject to displacements due to deformations of the Earth's crust. This effect is of upmost importance in basin evolution as it is one of the driving forces for fault formation and movement. Not all the numerical schemes developed till now allow the extension or contraction of the basement, for example in [2] the basin is a fixed box. This geometrical constraint is not acceptable for many applications and several works, such as [9,12,13], have a more general geometrical treatment. All of them use a Lagrangian approach combined with frequent remeshing, as it handles the movement of the boundary naturally. However, the application of classical Lagrangian methods to a real three-dimensional case is computationally very expensive. Some new types of Lagrangian methods, such as the Particle Finite Element Methods (PFEM), have been introduced in mechanical engineering [19] and applied to computational geology [20,21]. In these latter works, the Particle-In-Cell (PIC) technique is used. All these Lagrangian methods require a frequent mesh regeneration, that makes the cost of the algorithm critically dependent on the efficiency of the grid generator. To overcome this problem we have chosen an Arbitrary Lagrangian Eulerian method (ALE), similar to the one described in [22]. The reason is that, since the deformation of the basin boundary is usually small compared with that of the internal layers, we can decouple the two problems and use a Lagrangian scheme to reconstruct only the boundary movement and an implicit tracking algorithm for the internal layers (see [23]). The displacement of the grid is only prescribed on the boundary, while a suitable movement law is considered for the internal nodes, for example to minimize the mesh distortion. The ALE method has found several applications, see for instance [24–30]. The definition of the numerical algorithm for the computation of the internal grid movement is the most critical part: indeed we want at the same time to adapt the grid size, where necessary, and to preserve its overall quality. To achieve these goals, we choose the so called *r*-adaptivity (see [31]) combined with the ALE scheme. This technique is cheaper than the *h*-adaptivity, although less effective. Anyway, as the computational cost is an important issue for our purposes, the *r*-adaptivity is a reasonable choice. We exploit then the information given by a residual-type error estimator to construct an error-dependent metric, which drives the ALE scheme in adapting the grid size, according to the minimization of the estimated upper error bound.

Three descriptions of the internal grid movement have been introduced so far. The first two require the solution of a Laplace-type problem and the solution of a net of connected springs respectively, while the third one models the grid as a continuum elastic body. The first approach is used in [31–33]. Its merits are the low computational cost and the compatibility with a metric-type adaptivity. On the other hand, it could fail if high curvatures are present on the domain boundary; in particular, non convex regions could induce mesh tangles, that's why sometimes this method is combined with a smoothing technique (see [34]). The second method, often used in aeroelastic analysis, is the spring method (see [35,36]). The mesh is considered as a net of nodes linked by springs, whose topology varies among the methods. One of its most appreciated qualities is the robustness, as the mesh tanglement is (in the most advanced variants) always prevented. However, it is very expensive and several simplified versions have been developed, in which, for example, the nodes are moved one by one. This latter approach is very effective in aeroelastic simulations, where the boundary movement is usually concentrated in a small region at the center of the computational domain (which could represent an airfoil, an aircraft, etc.), but it is less effective in geological simulations where the boundary movement is more distributed. The last method is based on an elastic model (see [37]). It is more robust than the Laplace-type approach, although more expensive from the computational point of view. Therefore, we choose to implement a linearized version, as a compromise between robustness and computational efficiency. In particular we derive the elastic equation from an optimization problem, so that the r-metric adaptivity can be directly embedded in the model.

This paper is organized as follows: in Section 2 we briefly review the physical model, we discuss its relations with the literature, and we introduce a mathematical formulation well suited for numerical discretization. In Section 3, using a time splitting technique, we decouple the physical model into some smaller mathematical problems, and we provide for each one a dedicated numerical treatment; in particular, for two of them, namely the basin boundary movement and the fault tracking we propose some original ad hoc schemes that are detailed in Sections 4 and 5. Finally, in Section 6, we show the numerical results obtained from the simulation of nearly realistic cases.

2. Physical and mathematical models

2.1. Geometrical model

Let us introduce the geometric model of a sedimentary basin (Fig. 1). The domain $\Omega \subset \mathbb{R}^3$ is divided into n_s disjoint subdomains Ω_i (without overlaps), which represent different sedimentary layers characterized by different physical properties. The external boundary Γ of the domain Ω is divided into three parts: the basement Γ_B, where we apply a Dirichlet condition for the velocity field, the top of the basin Γ_S, with a free surface condition, and the lateral contour Γ_L, where we impose a Dirichlet condition on the horizontal plane and a slip condition in the vertical direction. Moreover, we suppose Γ_L vertical for simplicity. To complete our overview let us introduce some nomenclature: $\vec{X} = (x_1, x_2, x_3) \in \Omega$ is a point in the spatial domain of coordinates x_i, with i = 1, 2, 3, $(\hat{x}_1, \hat{x}_2, \hat{x}_3)$ are the unit vectors of the coordinate system, $\vec{n}$ is the domain outward normal and $t \in (0, T]$ is the time coordinate. For a generic vector $\vec{u}$, we denote its components with (u_1, u_2, u_3).

2.2. Mass balance equation

The mass conservation of the solid part of a basin can be expressed as (see [38]):

$$\frac{\partial}{\partial t}((1-\phi)\rho_i\lambda_i) + \vec{\nabla}\cdot((1-\phi)\rho_i\lambda_i\vec{u}) = 0, \quad i = 1,\ldots,n_s, \tag{1}$$

where ρ_i is the density of the ith layer, ϕ is the porosity, $\vec{u}$ is the sediment advection speed and

$$\lambda_i(\vec{X}) = \begin{cases} 1, & \text{if } \vec{X} \in \Omega_i, \\ 0, & \text{otherwise}, \end{cases}$$

is the characteristic function of the ith subdomain. We can assume that ρ_i is constant, with i = 1, . . ., n_s. Dividing (1) by ρ_i and summing up with respect to the number or layers we obtain

$$\sum_{i=1}^{n_s}\left(\frac{\partial}{\partial t}((1-\phi)\lambda_i)\right) + \sum_{i=1}^{n_s}(\vec{\nabla}\cdot((1-\phi)\lambda_i\vec{u})) = 0,$$

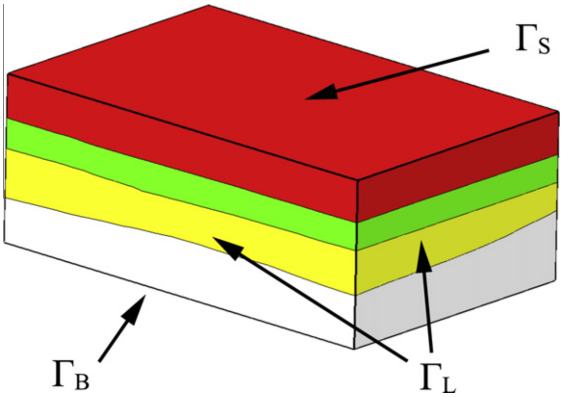


Fig. 1. Simplified sketch of the external shape of the domain Ω. The external boundary Γ is divided into three parts: the basement Γ_B, the free surface Γ_S, and the lateral contour Γ_L.

that, since $\sum_{i=1}^{n_s}\lambda_i = 1$ almost everywhere in Ω, can be simplified as

$$\frac{\partial}{\partial t}(1-\phi) + \vec{\nabla}\cdot((1-\phi)\vec{u}) = 0. \tag{2}$$

Finally, subtracting (2) from (1) divided by ρ_i we get

$$\frac{\partial \lambda_i}{\partial t} + \vec{u}\cdot\vec{\nabla}\lambda_i = 0, \quad i = 1,\ldots,n_s, \tag{3}$$

which represents an advection equation for the subdomains that can be tackled by multi-component level set methods, as detailed in [23,39].

Remark 1. The level set type Eq. (3) has been derived with the only assumption of the incompressibility of the solid matrix. Another way to derive the transport equation for the subdomains is to introduce a divergence-free hypothesis (see, for instance, [40]). However this approach prevents a consistent treatment of the compaction since the sediment volume reduction is not compatible with the divergence-free hypothesis.

2.3. Momentum balance equation and rheological models

The balance of momentum for a creeping fluid is stated by:

$$\vec{\nabla}\cdot\bar{\bar{\sigma}}(\mu,\vec{u}) - \vec{\nabla}P + \rho\vec{g} = 0,$$

where $\bar{\bar{\sigma}}$ is the stress tensor, P is the pressure, $\vec{g}$ is the gravitational acceleration vector. The density ρ is given by

$$\rho = \phi\rho_f + (1-\phi)\left(\sum_{i=1}^{n_s}\lambda_i\rho_i^0\right), \tag{4}$$

where ρ_i^0 is the reference density of the i-layer and ρ_f is the water density (in general the basin is nearly saturated by water). The expression of the stress tensor $\bar{\bar{\sigma}}$ depends on the choice of the rheological model. In this work, we consider a couple of pseudoplastic laws largely diffused in the literature [8,13]:

$$\mu_i = \begin{cases} \mu_i^0\gamma^{(r-2)/2} & \text{power-law}, \\ \mu_i^0(1+\xi_i\gamma)^{(r-2)/2} & \text{Carreau law}. \end{cases} \tag{5}$$

In these relations, μ_i is the apparent viscosity of each layer, μ_i^0 is the reference unstressed viscosity, γ is the squared Frobenius norm of the symmetric gradient $\vec{\nabla}\vec{u} + (\vec{\nabla}\vec{u})^T$, r is a positive coefficient, also known as the power-law coefficient, and ξ_i is a material dependent parameter. For $r = 2$ the power-law model reduces to the Newtonian behaviour. For $\gamma = 0$ it implies $\mu_i = 0$, a non-acceptable condition from the numerical point of view. This issue can be addressed by imposing a lower bound (see for instance [40]) or by using the Carreau law (see [13]), which can be considered a regularized form of the power-law. Both these approaches lead to the same expression for the stress tensor

$$\bar{\bar{\sigma}} = \mu(\vec{\nabla}\vec{u} + (\vec{\nabla}\vec{u})^T),$$

where the viscosity μ is defined as

$$\mu = \sum_{i=1}^{n_s}\lambda_i\mu_i. \tag{6}$$

In other words, this means that we assign a different viscosity to each layer through a level set approach.

Remark 2. In general the viscosity of the solid depends on the pressure and on the inverse of the temperature through an exponential relation (see, for instance, [8,13]). Anyway, in this paper we neglect the thermal effects on the dynamic of the basins, that will be addressed in future works. Nevertheless, the inclusion of a thermal dependent rheology for the sediments is straightforward and can be easily plugged into the model.

2.4. Modeling the compaction

We now detail the compaction model we have implemented for our simulations. We model the porosity decrease in the deep layers imposing a non-solenoidal velocity of the sediments, in other terms $\vec{\nabla}\cdot\vec{u} = \Phi$, where Φ is the compaction function. To find a relation for Φ we denote by $S(x_1,x_2)$ the relative height of the free surface, that is the distance along the $\hat{x}_3$ direction of the free surface from the x_1–x_2 plane. Hence, we can define the depth ζ as $\zeta = x_3 - S$ (see Fig. 2).

The objective is to model Φ using only a ϕ-depth relation. Let us consider, in particular, the so-called Athy compaction law for the porosity ϕ

$$\phi = \exp(\mathcal{B}\zeta)\left(\sum_{i=1}^{n_s}\lambda_i\phi_i^0\right), \tag{7}$$

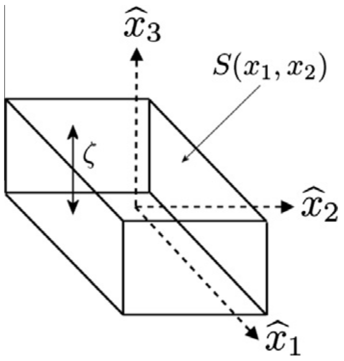


Fig. 2. An outline of the sedimentary basin with a reference frame having axes $(\hat{x}_1, \hat{x}_2, \hat{x}_3)$. The surface position is denoted with $S(x_1,x_2)$, the depth with ζ.

where ϕ_i^0 is the reference porosity of the ith layer and $\mathcal{B}$ is an empirical constant. We wish to determine Φ through (7). From [12] and (7) we get

$$\vec{\nabla}\cdot\vec{u} = \frac{1}{1-\phi}\frac{D\phi}{Dt} = \frac{1}{1-\phi}\frac{D\phi}{D\zeta}\frac{D\zeta}{Dt} = \mathcal{B}\frac{\phi}{1-\phi}\left(\frac{Dx_3}{Dt} - \frac{DS}{Dt}\right),$$

where $D/Dt = \partial/\partial t + \vec{u}\cdot\vec{\nabla}$ is the material derivative. Since $Dx_3/Dt = u_3$ and $\vec{\nabla}S$ is usually small (however this contribution could be included, if necessary), the above equation can be approximated as

$$\vec{\nabla}\cdot\vec{u} = \mathcal{B}\frac{\phi}{1-\phi}\left(u_3 - \frac{\partial S}{\partial t}\right).$$

From this relation we find that the compaction is the sum of two contributions: the first one, $\mathcal{B}(\phi/(1-\phi))u_3$, refers to the relative position of the layers and the latter, $-\mathcal{B}\phi/(1-\phi)\partial S/\partial t$, is related to the burial of the entire basin. In particular $\partial S/\partial t$ is known as the sedimentation speed and is provided, for example, by the geologists on the basis of some conjectures on the history of the basin.

This methodology is cheaper than the fully coupled fluid-structural one described in [41], and retains a consistent approach. In other works, such as [2,40], the sediments are compacted and de-compacted keeping a null divergence velocity field, which is not fully consistent.

2.5. Modeling the faults

From the geological point of view, faults modeling is of key importance. Faults are described using *geological restoration*, which is an inversion technique that allows to recover the timing and the characteristics of faults creation and evolution, starting from present day geological models and using only a kinematical approach. By the geological restoration we get the basin evolution over the time, represented as a set of geological snapshots of different ages. These snapshots provide the constraints of the dynamical modeling, which is the subject of the present work. In addition, the direct simulation of the dynamic of the model provides a validation tool for the geological restoration. For the time being the fulfilment of all physical constraints on the fault evolution (e.g. maximum dip allowed for a fault) are not included in the theoretical formulation, assuming that the fault description given by the geologists is correct and that local anomalies of fault behaviour during dynamic evolution may be acceptable while looking to the global basin evolution. In addition, we can verify whether the time of appearance of a fault triggers a basin evolution that is compatible with the present day configuration.

Faults are usually modeled as highly plasticized regions [9] or even as cracks [42]. As outlined in [11], the fluid model can describe accurately the stress field and the evolution of salt diapirism, though it can not describe faults formation. In other words, faults must be added separately in the model. A possible coherent way to include faults within a fluid framework is to decrease drastically the viscosity in a thin region of the domain, which represents, from the physical point of view, the fault-damaged region. Through this approach Eq. (6) becomes

$$\mu = \psi\sum_{i=1}^{n_s}\lambda_i\mu_i,$$

where $\psi \ll 1$ in the faulted region $\Omega^{\mathcal{F}}$ and equals the unity outside. The fault time of appearance and initial position are imposed a priori, for example from local structural analyses or restoration results. From this configuration, the fault is moved accordingly to the following relation:

$$\frac{\partial\psi}{\partial t} + \vec{u}\cdot\vec{\nabla}\psi = 0,$$

which imposes a null net sediment flux across it. This technique has been already used in [12,13] providing good results.

2.6. The model

We can now summarize the complete model in the advective ALE form, including all the features described so far

$$\begin{cases} \vec{\nabla}\cdot\bar{\bar{\sigma}}(\mu,\vec{u})-\vec{\nabla}P+\rho\vec{g}=0 & \text{in } \Omega\times[0,T], \\ \vec{\nabla}\cdot\vec{u}=\Phi(\phi) & \text{in } \Omega\times[0,T], \\ \frac{\partial\lambda_i}{\partial t}+(\vec{u}-\vec{u}_g)\cdot\vec{\nabla}\lambda_i=0 & \text{in } \Omega\times(0,T], \\ \frac{\partial\psi}{\partial t}+(\vec{u}-\vec{u}_g)\cdot\vec{\nabla}\psi=0 & \text{in } \Omega\times(0,T], \\ \bar{\bar{\sigma}}=\mu(\vec{u},\psi,\xi_i)(\vec{\nabla}\vec{u}+(\vec{\nabla}\vec{u})^T) & \text{in } \Omega\times(0,T], \\ \lambda_i=\bar{\lambda}_i, \quad \psi=\bar{\psi} & \text{in } \Omega\times\{0\}, \\ \vec{u}=\bar{u} & \text{on } \Gamma_B, \\ (\bar{\bar{\sigma}}-P\bar{\bar{I}})\cdot\hat{n}=0 & \text{on } \Gamma_S, \\ u_1=\bar{u}_1, \quad u_2=\bar{u}_2, \quad ((\bar{\bar{\sigma}}-P\bar{\bar{I}})\cdot\hat{n})\cdot\hat{x}_3=0 & \text{on } \Gamma_L, \end{cases} \tag{8}$$

where $\bar{u}$ is a boundary velocity field, $\vec{u}_g$ is the grid velocity linked to the ALE treatment of the boundary movement, and $\overline{\lambda_i}$ and $\bar{\psi}$ are a suitable set of initial conditions. From the modeling point of view, problem (8) is a collection of models already present in the literature. However, the assembling of the different models, together with the numerical techniques described in the forthcoming sections, is original and provides a large number of information in a coherent manner.

3. Numerical techniques

3.1. Time splitting algorithm

In this section we introduce the temporal discretization of the model. First of all we split the time interval $[0,T]$ into subintervals whose endpoints are $t^0 = 0, t^1, \ldots, t^n, t^{n+1}, \ldots, T$, where $t^{n+1} = t^n + \Delta t^n$ and Δt^n is the nth time step. Then, for the generic variable $a(t,\vec{X})$, we indicate for simplicity $a^n(\vec{X})=a(t^n,\vec{X})$. We evolve the solution according to the following scheme:

Algorithm 1. At the generic time step, n, we know the following variables of the problem:

$$\left\{\vec{u}^{n-1},P^{n-1},\vec{u}_g^{n-1},\lambda_i^n,\psi^n,\rho^{n-1},\mu^{n-1},\phi^{n-1}\right\} \quad \text{with } n \geqslant 1.$$

Then we use the following numerical scheme to solve problem (8):

1. from λ_i^n and relation (7) compute ϕ^n;
2. from λ_i^n, ϕ^n and relation (4) compute ρ^n;
3. from λ_i^n, ψ^n, $\vec{u}^{n-1}$ and relations (5) and (6) compute μ^n;
4. solve a linearized Stokes problem for $\vec{u}^n$ and P^n:

   $$\begin{cases} \vec{\nabla}\cdot(\mu^n(\vec{\nabla}\vec{u}^n+(\vec{\nabla}\vec{u}^n)^T))-\vec{\nabla}P^n+\rho^n\vec{g}=0, \\ \vec{\nabla}\cdot\vec{u}^n=\Phi(\phi^n), \end{cases}$$

   Note: at the first step ($n = 0$) $\vec{u}^{n-1}$ is not available. Therefore, only in this case, a non linear Stokes problem is solved using a fixed point iterative technique

   $$\begin{cases} \vec{\nabla}\cdot\left(\mu(\vec{u}^{(0,m)})(\vec{\nabla}\vec{u}^{(0,m+1)}+(\vec{\nabla}\vec{u}^{(0,m+1)})^T)\right)-\vec{\nabla}P^{(0,m+1)}+\rho^{(0)}\vec{g}=0, \\ \vec{\nabla}\cdot\vec{u}^{(0,m+1)}=\Phi, \end{cases}$$

   where m is the iteration apex and $\vec{u}^{(0,0)}$ has been set equal to 0;
5. a time step Δt^n is computed. This requires three steps: firstly, from the velocity $\vec{u}^n$, a maximum Δt^n_{ALE} is estimated such that the movement of the boundary elements does not exceed a certain threshold. Secondly a maximum Δt^n_C is estimated such that the movement of the internal sediment layers is less than a prescribed length. Finally, the time step of the n-iteration is computed as

   $$\Delta t^n=\min\left(\Delta t^n_{\text{ALE}},\Delta t^n_C\right); \tag{9}$$

6. from Δt^n solving a mesh adapting elastic equation (see Section 4) the grid velocity $\vec{u}_g^n$ is evaluated;
7. compute λ_i^{n+1} and ψ^{n+1} by solving the following evolution equations in $(t^n,t^{n+1}]$:

   $$\begin{cases} \frac{\partial\lambda_i^{n+1}}{\partial t}+(\vec{u}^n-\vec{u}_g^n)\cdot\vec{\nabla}\lambda_i^{n+1}=0 \quad \text{for } i=1,\ldots,n_s, \\ \frac{\partial\psi^{n+1}}{\partial t}+(\vec{u}^n-\vec{u}_g^n)\cdot\vec{\nabla}\psi^{n+1}=0; \end{cases}$$

8. with the grid velocity $\vec{u}_g^n$ and Δt^n the grid is moved to a new configuration.

At the end of the time step, we get the update variable vector of the problem:

$$\left\{\vec{u}^n, P^n, \vec{u}_g^n, \lambda_i^{n+1}, \psi^{n+1}, \rho^n, \mu^n, \phi^n\right\}.$$

Algorithm 1 decouples problem (8) into smaller standard mathematical problems, which can be addressed more easily one by one. In particular:

- steps 1, 2, and 3 involve algebraic equations. From the numerical viewpoint, they require at most some interpolations between the different discrete counterparts of ϕ, ρ, μ, λ_i, and $\vec{u}$;
- step 4 involves the solution of a linearized Stokes problem, which is accomplished by an explicit time scheme. This formulation reduces the computational burden with respect to a nonlinear Stokes problem (see for instance [10]). In this work, we solve the Stokes problem using piecewise linear finite elements (FE), implementing the mini-element enrichment to satisfy the inf-sup condition (see for details [43,44]). The resulting linear problem can be tackled with iterative Krylov-type techniques [45–47];
- step 6 involves the solution of an artificial elasticity problem to compute the grid velocity $\vec{u}_g$ according to a prescribed displacement of the lateral boundaries. The extension of the basin and the *r*-type adaption are two of the most innovative parts of this work and are addressed in detail in Section 4;
- finally, step 7 involves the solution of $n_s + 1$ transport equations, which are addressed by the conservative coupled Level Set-Volume Tracking (LS-VT) method developed in [23]. In addition, we adopt an element-based variant for fault tracking, for which a new set-reconstruction algorithm has been derived (see Section 5.1).

3.2. Spatial discretization

The geometric approximation of the domain Ω is described by three different computational grids. Most of the discrete variables are defined on the coarsest grid, to minimize the computational burden. A second finer grid is introduced to define the velocity field in order to satisfy the inf–sup condition. Finally, the tracking-related variables ψ and λ_i are defined on a third grid, the finest, to minimize numerical diffusion. For the sake of clarity, in the following we refer to a generic grid $\mathcal{T}_\Delta$ containing n_e elements e_r (with $r = 1,\ldots,n_e$) and n_p nodes $\vec{x}_k$ (with $k = 1,\ldots,n_p$), where the subscript Δ stands for the maximum diameter of the grid elements. For more details on the spatial discretization, see [1].

4. Adaptive grid movement

In this section we introduce an original ad hoc numerical technique to deal with the movement of the lateral boundaries of the domain Ω. In particular, this methodology refers to the modeling of the so called *extensional tectonics*, which basically deals with simulations where the lateral boundaries of the basin are subjected to extensional or compressional movements. To this purpose, Section 4.1 describes an ALE formulation, where the displacement of the lateral walls is computed by imposing a normal velocity field at the boundaries. This approach is successively completed in Section 4.2 by introducing the so called *r*-adaptivity technique (see [31]) for the computation of the grid movement. This strategy, which can be easily combined with the ALE formulation, is able to adapt the grid size relying upon an estimate on the solution error, and to preserve the overall quality.

4.1. Grid movement equations

To move the mesh according to a displacement of the lateral boundary, we define an artificial elasticity problem and use a solution-dependent metric. This can be considered as a numerical tool designed to preserve and adapt the quality of the mesh. Its purpose is to compute an optimized new mesh after the deformation, and has no influence on the physical behavior of the sediments or on the physical movement of the basin.

Let Ω^n be the domain at time t^n and let $\vec{Y}^n \in \Omega^n$ be the position vector in the current reference system. We want to build a smooth displacement field $\vec{S}^n : \Omega^n \rightarrow \Omega^{n+1}$ such that

$$\begin{cases} \vec{S}^n = \vec{u}_\Delta^n \Delta t^n & \text{on } \Gamma_B, \\ S_1^n = u_{\Delta,1}^n \Delta t^n, \quad S_2^n = u_{\Delta,2}^n \Delta t^n & \text{on } \Gamma_L, \\ \vec{S}^n \cdot \vec{n} = (\vec{u}_\Delta^n \cdot \vec{n}) \Delta t^n & \text{on } \Gamma_S. \end{cases} \tag{10}$$

To implement our mesh movement-adaption scheme, we adopt the ideas developed in [48,49]. We seek a best fit solution in $H^1(\Omega)$, where H^1 is the classical Sobolev space of the squared-integrable functions with an integrable weak derivative (see [50]), to the alignment and to the equal distribution condition

$$(\vec{\nabla}\vec{X})^T \cdot \overline{\overline{M}} \cdot (\vec{\nabla}\vec{X}) = \bar{\bar{I}}\left(\frac{1}{|\Omega|}\int_\Omega \sqrt{\det(\overline{\overline{M}})}\right)^{2/3},$$

where $\overline{\overline{M}}$ is a positive definite second order tensor which will be linked to the adaption process and $\vec{X}$ is the position with respect to the initial configuration Y at $t = 0$. In addition, we impose the constraint that the volume $|\Omega|$ does not change, i.e.,

$$\int_\Omega \sqrt{\det(\overline{\overline{M}})} = |\Omega|, \tag{11}$$

where the left-hand side represents the volume of the domain after the movement of the mesh. A best fit solution of the alignment condition is given by the minimization of

$$\min_{\vec{X}\in H^1(\Omega)} \frac{1}{2}\|(\vec{\nabla}\vec{X})^T \cdot \overline{\overline{M}} \cdot (\vec{\nabla}\vec{X}) - \bar{\bar{I}}\|^2_{\mathcal{D}}, \tag{12}$$

where $\|\overline{\overline{\mathcal{C}}}\|^2_{\mathcal{D}} = \int_{\Omega^n} \overline{\overline{\mathcal{C}}} \cdot \overline{\overline{\mathcal{D}}} \cdot \overline{\overline{\mathcal{C}}}, \overline{\overline{\mathcal{D}}}$ is a tensor with components $\mathcal{D}_{ijhk} = E_L\delta_{ij}\delta_{hk} + 2K_L\delta_{ik}\delta_{jh}$, and δ_{ij}, E_L, K_L are respectively the Kronecker delta, and the fictitious shear and bulk elastic moduli. According to [31,37] we choose $E_L|_{e_r} = K_L|_{e_r} = 1/|e^G_r|$. The optimality conditions of (12) can be transformed into a non linear partial differential equation, but, since the displacement between two time steps is kept small, we can consider a linearized form of the resulting PDE. Moreover, to simplify its derivation, we neglect, for now, the boundary conditions as they will be introduced later on.

The linearized form of (12) with respect to the coordinate system Y^n is

$$\min_{\vec{S}^n\in H^1(\Omega^n)} \frac{1}{2}\int_{\Omega^n} \bar{\bar{\epsilon}}^n \cdot \overline{\overline{\mathcal{D}}} \cdot \bar{\bar{\epsilon}}^n + \int_{\Omega^n} \bar{\bar{\sigma}}^n_0 : \bar{\bar{\epsilon}}^n, \tag{13}$$

where $\bar{\bar{\sigma}}^n_0$ is the pre-stress at time t^n. At $n = 0$ we have $\bar{\bar{\sigma}}^n_0 = 0$, and shortly we will define how to update it. Then

$$\bar{\bar{\epsilon}}^n = \frac{1}{2}\left((\vec{\nabla}\vec{X}^n)^T \cdot \overline{\overline{M}}^n \cdot \vec{\nabla}\vec{X}^n - \bar{\bar{I}}\right), \tag{14}$$

is the strain tensor. Plugging

$$\vec{X}^n = \vec{Y}^n + \vec{S}^n$$

into (14) we get

$$\bar{\bar{\epsilon}}^n = \underbrace{\frac{1}{2}\left((\vec{\nabla}\vec{S}^n)^T \cdot \overline{\overline{M}}^n \cdot (\vec{\nabla}\vec{S}^n)\right)}_{\bar{\bar{\epsilon}}^n_2} + \underbrace{\frac{1}{2}\left((\vec{\nabla}\vec{S}^n)^T \cdot \overline{\overline{M}}^n + \overline{\overline{M}}^n \cdot (\vec{\nabla}\vec{S}^n)\right)}_{\bar{\bar{\epsilon}}^n_1} + \underbrace{\frac{1}{2}\left(\overline{\overline{M}}^n - \bar{\bar{I}}\right)}_{\bar{\bar{\epsilon}}^n_0}.$$

By differentiating Eq. (13) with respect to the displacement we obtain

$$\int_{\Omega^n} (\bar{\bar{w}}_1 + \bar{\bar{w}}_2) \cdot \overline{\overline{\mathcal{D}}} \cdot (\bar{\bar{\epsilon}}^n_0 + \bar{\bar{\epsilon}}^n_1 + \bar{\bar{\epsilon}}^n_2) + \int_{\Omega^n} \bar{\bar{\sigma}}^n_0 : (\bar{\bar{w}}_1 + \bar{\bar{w}}_2) = 0 \quad \forall \vec{v} \in H^1(\Omega),$$

where

$$\bar{\bar{w}}_1 = \tfrac{1}{2}\left((\vec{\nabla}\vec{v})^T \cdot \overline{\overline{M}}^n + \overline{\overline{M}}^n \cdot (\vec{\nabla}\vec{v})\right),$$
$$\bar{\bar{w}}_2 = \tfrac{1}{2}\left((\vec{\nabla}\vec{S}^n)^T \cdot \overline{\overline{M}}^n \cdot (\vec{\nabla}\vec{v}) + (\vec{\nabla}\vec{v})^T \cdot \overline{\overline{M}}^n \cdot (\vec{\nabla}\vec{S}^n)\right),$$

and $\vec{v} \in H^1(\Omega)$ is a test function. Considering only the linear parts, we get

$$\int_{\Omega^n} \bar{\bar{w}}_1 \cdot \overline{\overline{\mathcal{D}}} \cdot \bar{\bar{\epsilon}}^n_1 + \int_{\Omega^n} \left(\bar{\bar{\sigma}}^n_0 + \overline{\overline{\mathcal{D}}} \cdot \bar{\bar{\epsilon}}^n_0\right) : \left(\bar{\bar{w}}_1 + \bar{\bar{w}}_2\right) = 0 \quad \forall \vec{v} \in H^1(\Omega).$$

For simplicity, we set $\overline{\overline{\Sigma}}^n_0 = \bar{\bar{\sigma}}^n_0 + \overline{\overline{\mathcal{D}}} \cdot \bar{\bar{\epsilon}}^n_0$. Integrating by parts, we obtain the following balance equation for the displacements:

$$-\vec{\nabla} \cdot \left[E_L(\vec{\nabla}\vec{S}^n : \overline{\overline{M}}^n)\bar{\bar{I}} + K_L\left((\vec{\nabla}\vec{S}^n)^T \cdot \overline{\overline{M}}^n + \overline{\overline{M}}^n \cdot (\vec{\nabla}\vec{S}^n)\right) + \overline{\overline{\Sigma}}^n_0 \cdot (\vec{\nabla}\vec{S}^n + \bar{\bar{I}}) \cdot \overline{\overline{M}}^n\right] = 0,$$

which is complemented by the following set of boundary conditions (see Fig. 1):

$$\begin{cases} \vec{S}^n = \vec{u}^n_\Delta \Delta t^n & \text{on } \Gamma_B, \\ S^n_1 = u^n_{\Delta,1}\Delta t^n, \quad S^n_2 = u^n_{\Delta,2}\Delta t^n, \quad (\overline{\overline{\Sigma}}^n_0 \cdot \vec{n}) \cdot \hat{x}_3 = 0 & \text{on } \Gamma_L, \\ \vec{S}^n \cdot \vec{n} = (\vec{u}^n_\Delta \cdot \vec{n})\Delta t^n, \quad (\overline{\overline{\Sigma}}^n_0 \cdot \vec{n}) \cdot \vec{n} = 0 & \text{on } \Gamma_S. \end{cases}$$

This is a linear elastic-type equation with a pre-stress term that arises from the previous deformation of the grid. We seek a solution using the finite element method, therefore now we introduce the discrete weak formulation, i.e., we seek $\vec{S}^n_\Delta \in \mathbb{V}_1 \cap H^1_\Gamma$ such that:

$$a_{\mathrm{ALE}}(\vec{S_\Delta}^n, \vec{v}_\Delta) = F_{\mathrm{ALE}}(\vec{v}_\Delta) \quad \forall \vec{v}_\Delta \in \mathbb{V}_1 \cap H^1_\Gamma, \tag{15}$$

where H^1_Γ is the set of the functions $\vec{v}$ belonging to H^1 that satisfy the Dirichlet boundary conditions (10) and $\mathbb{V}_1 = \{\vec{\varphi}_\Delta \in \mathbb{C}^0(\Omega) : \vec{\varphi}_\Delta|_{e_r} \in \mathbb{P}^1,\ r = 1, \ldots, n_e\}$. Moreover $\vec{S}^n_\Delta \in \mathbb{V}_1$ is the discrete counterpart of $\vec{S}^n$ and

$$\begin{cases} a_{\mathrm{ALE}}(\vec{S_\Delta}^n, \vec{v}_\Delta) = \int_{\Omega^n} \left[E_L(\vec{\nabla}\vec{S}^n : \overline{\overline{M}}^n)\bar{\bar{I}} + K_L\left((\vec{\nabla}\vec{S}^n)^T \cdot \overline{\overline{M}}^n + \overline{\overline{M}}^n \cdot (\vec{\nabla}\vec{S}^n)\right) + \overline{\overline{\Sigma}}^n_0 \cdot \vec{\nabla}\vec{S}^n \cdot \overline{\overline{M}}^n\right] : \vec{\nabla}\vec{v}_\Delta, \\ F_{\mathrm{ALE}}(\vec{v}_\Delta) = -\int_{\Omega^n} \overline{\overline{\Sigma}}^n_0 \cdot \overline{\overline{M}}^n. \end{cases}$$

If $K_L, E_L > 0$, problem (15) is coercive. Moreover if $\overline{\overline{\Sigma}}^n_0, \overline{\overline{M}}^n, K_L, E_L \in L^\infty(\Omega)$, then the bilinear form $a_{\mathrm{ALE}}(\cdot,\cdot)$ and the linear functional $F_{\mathrm{ALE}}(\cdot)$ are bounded and problem (15) has a unique solution (see [43,44,51]).

The pre-stress term is updated as follows

$$\bar{\bar{\sigma}}^{n+1}_0 = (\vec{\nabla}\vec{S}^n)^{-T} \cdot \overline{\overline{\Sigma}}^n_0 \cdot (\vec{\nabla}\vec{S}^n)^{-1}.$$

Once the displacement field is computed, the grid speed is calculated as $\vec{u}^n_{g,\Delta} = \vec{S}^n_\Delta / \Delta t^n$.

4.2. Metric definition

A good metric definition is expected to preserve the mesh quality and, at the same time, relocate the nodes to get a more accurate solution. Our goal is to construct the metric tensor $\overline{\overline{M}}^n$ relying upon a residual error estimate $\mathcal{E}^n_r = \Delta^2(R^n_r)^2 + \Delta(J^n_r)^2 + (D^n_r)^2$ where

$$R^n_r = \left\|\vec{\nabla}P^n_\Delta - \vec{\nabla}\cdot\bar{\bar{\sigma}}^n_\Delta - \rho^n_\Delta\vec{g}\right\|_{L^2\left(e^G_r\right)},$$

$$D^n_r = \left\|\Phi - \vec{\nabla}\cdot\vec{u}^n_\Delta\right\|_{L^2\left(e^G_r\right)},$$

$$J^n_r = \tfrac{1}{2}\sum_{j=1}^{4}\left\|\left[\left(\bar{\bar{\sigma}}^n_\Delta - P^n_\Delta\bar{\bar{I}}\right)\cdot\hat{n}_{r,j}\right]\right\|_{L^2\left(\mathcal{F}^G_{r,j}\right)},$$

$\hat{n}_{r,j}$ is the outward normal of the jth face of the rth element, $\|\cdot\|_{L^2}$ is the L^2 norm, and L^2 is the space of the squared Lebesgue-integrable functions, see [44].

We choose an isotropic metric $\overline{\overline{M}}^n = \bar{\bar{I}}\eta^n$. By doing so, the problem reduces to finding a suitable field η^n. Then, we define an auxiliary variable $\beta^n_r = (\eta^n)^{3/2}$ that represents the local volumetric deformation induced by the metric $\overline{\overline{M}}^n$. We define β^n_r as the solution of the following minimization problem

$$\min_{\beta^n_r \in \mathbb{R}^{n^G_e}} \frac{1}{2}\left(\beta^n_r - \hat{\beta}^n_r\right)^2 + \frac{\delta}{2}\left(\beta^n_r - \tilde{\beta}^n_r\right)^2, \tag{16}$$

where δ is an appropriate weight factor (that will be defined later) and

$$\tilde{\beta}^n_r = \widetilde{K}^n\mathcal{R}^n_r, \quad \hat{\beta}^n_r = \frac{\sqrt{K^n_{\mathcal{E}}}}{\mathcal{E}^n_r}, \tag{17}$$

$\mathcal{R}^n_r$ being the ratio between the volume of the element e^G_r at time t^0 and at time t^{n-1}. Then

$$\widetilde{K}^n = \frac{|\Omega|}{\sum_{r=1}^{n^G_e}\mathcal{R}^n_r\left|e^{G,n}_r\right|}, \quad \sqrt{K^n_{\mathcal{E}}} = \frac{|\Omega|}{\sum_{r=1}^{n^G_e}\left|e^{G,n}_r\right| \Big/ \mathcal{E}^n_r}. \tag{18}$$

As we will see shortly, the solution of (16) represents a compromise between the aim to distribute the error uniformly along the cells and the necessity to maintain the overall mesh quality, by means of the weight δ.

The term $(\beta^n_r - \hat{\beta}^n_r)^2/2$ in problem (16) triggers the equidistribution of the error. Indeed our aim is to obtain that the contribution to the error of each element in the grid at time t^n is the same, that is

$$\left(\mathcal{E}^n_r\frac{\left|e^{G,n+1}_r\right|}{\left|e^{G,n}_r\right|}\right)^2 = (\mathcal{E}^n_r\beta^n_r)^2 = K^n_{\mathcal{E}} \quad r = 1, \ldots, n^G_e,$$

where we have supposed that, for small grid deformations, the ratio $|\mathcal{E}_r| / \left|e^{G,n-1}_r\right|$ is almost independent of the grid geometry.

On the contrary the term $\delta(\beta^n_r - \tilde{\beta}^n_r)^2/2$ in problem (16) ensures that the grid does not experience an excessive deformation, and in particular, we would like to impose that the elements deform with the same volume variation, i.e.,

$$\beta^n_r = \widetilde{K}\mathcal{R}^n_r \quad r = 1, \ldots, n^G_e.$$

The solution of (16) is

$$\beta_r = \frac{\delta\tilde{\beta}_r^n + \hat{\beta}_r^n}{1+\delta}. \tag{19}$$

We can show that (19) satisfies (11), in fact

$$\int_{\Omega^n} \sqrt{\det(\overline{\overline{M}}^n)} = \sum_{r=1}^{n_e^G} \beta_r |e_r^{G,n}|.$$

Then using (17) and (18) we get:

$$\int_{\Omega^n} \sqrt{\det(\overline{\overline{M}}^n)} = \frac{1}{1+\delta}\left(\delta\widetilde{K}^n \sum_{r=1}^{n_e^G} \mathcal{R}_r^n |e_r^{G,n}| + \sqrt{K_{\mathcal{E}}^n} \sum_{r=1}^{n_e^G} \frac{|e_r^{G,n}|}{\mathcal{E}_r^n}\right) = \frac{1}{1+\delta}(\delta|\Omega| + |\Omega|) = |\Omega|.$$

As regards the weight δ, we choose

$$\delta > \max\left(\frac{\hat{\beta}_{\max}^n - \mathcal{U}}{\mathcal{U} - \tilde{\beta}_{\max}^n}, \frac{\mathcal{L} - \hat{\beta}_{\min}^n}{\tilde{\beta}_{\min}^n - \mathcal{L}}, 0\right), \tag{20}$$

where $\hat{\beta}_{\max}^n, \tilde{\beta}_{\max}^n$ are the maximum values of $\hat{\beta}_r^n, \tilde{\beta}_r^n$ and $\hat{\beta}_{\min}^n, \tilde{\beta}_{\min}^n$ are the minimum values of $\hat{\beta}_r^n, \tilde{\beta}_r^n$ and $\mathcal{U}, \mathcal{L}$ are the upper and lower bounds we want to impose on β_r. Finally we have:

Proposition 1. *If $\mathcal{U} > \tilde{\beta}_{\max}^n$, $\mathcal{L} < \tilde{\beta}_{\min}^n$, then (19) and (20) imply $\mathcal{L} \leqslant \beta_r^n \leqslant \mathcal{U}$.*

Proof. β_r^n is an increasing function of $\hat{\beta}_r^n$ and $\tilde{\beta}_r^n$, as

$$\frac{\partial\beta_r^n}{\partial\hat{\beta}_r^n} = \frac{1}{1+\delta} > 0, \quad \frac{\partial\beta_r^n}{\partial\tilde{\beta}_r^n} = \frac{\delta}{1+\delta} > 0.$$

Therefore from (19) it follows that $\beta_r^n < \left(\hat{\beta}_{\max}^n + \delta\tilde{\beta}_{\max}^n\right)/(1+\delta)$. Since (20) implies $\left(\hat{\beta}_{\max}^n + \delta\tilde{\beta}_{\max}^n\right)(1+\delta) < \mathcal{U}$, we get the first part of the bound. With similar arguments the other bound holds. □

5. Handling of faults

In this section we address the problem of faults tracking. The nature of the fluid approach imposes to track the position of a very thin layer representing the fault damaged volume. Many numerical approaches can be employed, such as the Lagrangian [52] or the Eulerian [53] ones, but a complete review of all the tracking algorithms is beyond the scope of this work. Moreover, all these techniques are designed to compute the evolution of surfaces, while here we are interested in reconstruction of volumes. Hence, these approaches would have implied to compute the distance between the fault surface and all the mesh elements at each time step to identify the fault volume, a strategy that, from the computational viewpoint, could be very expensive.

In order to devise an efficient approach, instead, we have developed a novel scheme which is able to track the damaged volume directly. In particular, our approach is specifically designed to track volumes with a very high aspect ratio, which is the case of the damaged fault volumes, and in addition, it retains the volume-conservation properties of the finite volume methods.

5.1. A finite volume scheme

Let us introduce the fault function $\lambda^{\mathcal{F}}$ such that

$$\begin{cases} \lambda^{\mathcal{F}} > 1/2 & \text{in } \Omega^{\mathcal{F}}, \\ \lambda^{\mathcal{F}} \leqslant 1/2 & \text{in } \Omega \setminus \Omega^{\mathcal{F}}. \end{cases}$$

In addition, we define the viscosity abatement function ψ, already introduced in (6), as

$$\psi = \begin{cases} \mathcal{A} & \text{in } \Omega^{\mathcal{F}}, \\ 1 & \text{in } \Omega \setminus \Omega^{\mathcal{F}}, \end{cases}$$

where $0 < \mathcal{A} \leqslant 1$ is the viscosity abatement factor. The evolution of the fault function is determined by the following transport equation in an ALE form:

$$\frac{\partial\lambda^{\mathcal{F}}}{\partial t} + (\vec{u} - \vec{u}_{g,\Delta}) \cdot \vec{\nabla}\lambda^{\mathcal{F}} = 0. \tag{21}$$

The discrete counterpart of $\lambda^{\mathcal{F}}(t,\cdot)$ is $\lambda^{\mathcal{F}}_{\Delta}(t,\cdot) \in \mathbb{W}_0$, where $\mathbb{W}_0 = \{\varphi_\Delta \in L^2(\Omega) : \varphi_\Delta|_{e_r} \in \mathbb{P}^0,\ r = 1,\ldots,n_e\}$, and is piecewise constant on the elements. Its degrees of freedom at time t^n are denoted with $\lambda_r^{\mathcal{F},n}$. We solve (21) with an element-based version of the implicit tracking method developed in [23]. If it is necessary, the tracking is performed on a finer grid and then interpolated back onto the original one.

It is worth noting that the LS-VT coupled scheme is positive, i.e., $0 \leqslant \lambda^{\mathcal{F}}_{\Delta}(t,\cdot) \leqslant 1$ for all $t > 0$. The next step is to develop a proper set reconstruction technique, since the definition of the level set function here is different from the standard distance function (see [53,54]).

5.2. Set reconstruction (continuous part)

The set reconstruction problem can be posed as follows: given $\lambda^{\mathcal{F}}_{\Delta}$ and $\Omega^{\mathcal{F}} = H(\lambda^{\mathcal{F}}_{\Delta} - 1/2)$, where H is the Heaviside function

$$H(x) = \begin{cases} 1 & \text{if } x > 0, \\ 0 & \text{if } x \leqslant 0, \end{cases}$$

find a new fault region function that is less diffused. In this section we construct a method in the continuous framework to find a function θ such that $H(\theta) = \Omega^{\mathcal{F}}$, given $\lambda^{\mathcal{F}}$ and $\Omega^{\mathcal{F}}$. This is a tautology in the continuous framework, however, in the discrete one $H(\theta)$ can be used as the reconstruction of the fault function.

Let us now introduce the method in its continuous form. We define a coefficient $\alpha(t)$ as

$$\alpha(t) = \int_\Omega (\lambda^{\mathcal{F}}(t,\cdot) - 1/2) H(\lambda^{\mathcal{F}}(t,\cdot) - 1/2). \tag{22}$$

The fault region can be found as $H(\theta)$, where θ satisfies

$$\begin{cases} J = \min\limits_{\theta \in L^2(\Omega)} \frac{1}{2} \int_\Omega (\theta - \lambda^{\mathcal{F}})^2, \\ \int_\Omega \theta = \alpha, \\ \theta \geqslant 0. \end{cases} \tag{23}$$

We will see in a while that $H(\theta) = H(\lambda^{\mathcal{F}} - 1/2)$.

Proposition 2. *The solution of problem (23) is*

$$\bar{\theta} = (\lambda^{\mathcal{F}} - 1/2) H(\lambda^{\mathcal{F}} - 1/2).$$

Proof. See Appendix A. □

In our particular situation we choose $\lambda^{\mathcal{F}} = \chi_{\Omega^{\mathcal{F}}}$, where $\chi_{\Omega^{\mathcal{F}}}$ is the characteristic function of the domain $\Omega^{\mathcal{F}}$. In this case we have $\alpha = 1/2|\Omega^{\mathcal{F}}| = 1/2 \int_\Omega \lambda^{\mathcal{F}}$. This choice has profound consequences on the discretized method, that we are going to introduce shortly, since, in the discrete setting, $1/2 \int_\Omega \lambda^{\mathcal{F}}$ can be estimated much better than (22). For instance, in the case $\vec{\nabla} \cdot \vec{u} = 0$, $\alpha(t)$ is constant, and any conservative numerical scheme for (21) will conserve $\int_\Omega \lambda^{\mathcal{F}}$ and, as a consequence, $\alpha(t)$. Our tracking method is conservative (see [23]).

5.3. Set reconstruction (discrete part)

We approximate now θ with $\Theta_\Delta \in \mathbb{W}_0$, a piecewise constant function defined on the elements of the sub-grid. The discrete counterpart of problem (23) has the form

$$\min_{\Theta_\Delta \in \mathbb{W}_0} \frac{1}{2} \int_\Omega \left(\Theta_\Delta - \lambda^{\mathcal{F}}_{\Delta}\right)^2 + \eta\left(\int_\Omega \Theta_\Delta - \alpha\right) - \int_\Omega \mathcal{M}\Theta_\Delta - \int_\Omega \mathcal{N}\left(\frac{1}{2} - \Theta_\Delta\right), \tag{24}$$

where $\eta \in \mathbb{R}$, $\mathcal{M} : \Omega \to \mathbb{R}$ and $\mathcal{N} : \Omega \to \mathbb{R}$ are the three Lagrange multipliers that force respectively the first and the second constraint in (23), and $\Theta_\Delta \leqslant 1/2$. The latter condition is not present in the continuous form, but comes from Proposition 2, that imposes $\theta \leqslant 1/2$. Finally α is estimated as $\alpha = 1/2 \int_\Omega \lambda^{\mathcal{F}}_{\Delta}$.

The equivalent optimality conditions for (24) are:

$$\begin{cases} \Theta_r = \lambda_r^{\mathcal{F}} + (\mathcal{M}_r - \mathcal{N}_r - \eta), \\ \eta = \frac{1}{|\Omega|}\left(\sum\limits_{r=1}^{n_e^S} \lambda_r^{\mathcal{F}} |e_r^S| + \sum\limits_{r=1}^{n_e^S} |e_r^S| (\mathcal{M}_r - \mathcal{N}_r) - \alpha\right), \end{cases}$$

where $\mathcal{M}_r, \mathcal{N}_r, \Theta_r \in \mathbb{W}_0$. This system is solved using the Uzawa method (see [55,56]). At the end of the iterative cycle, the function $\lambda^{\mathcal{F}}_{\Delta}$ is reconstructed by setting $\lambda^{\mathcal{F}}_{\Delta} = H(\Theta_\Delta)$. A one-dimensional (1D) example of the reconstruction algorithm is provided in Appendix B.

6. Numerical results

We present now some numerical examples to demonstrate the capabilities of our mathematical modeling set up through some realistic simulations. For the time being we only want to make sure that the numerical techniques we have chosen work properly, as a complete physical validation test is beyond the scope of this work. However a physical discussion of the numerical results is included.

6.1. Diapir growth

In the first example we perform a comparison between the results coming from our numerical simulations of the Rayleigh–Taylor instability (associated with diapirism) and published data. We consider a non-dimensional cube domain divided into three layers, as shown in Fig. 3(a). The discretized geometry have roughly 12,000 nodes and 60,000 elements, for a total amount of 228,000 DOF. For each layer we consider different geometrical and physical configurations, as outlined in Table 1. In particular, we vary the viscosity of the middle layer in cases 1 and 2 and its thickness in cases 3 and 4. The physical characteristics of the middle and the upper layers are left unchanged. Although they could be treated as a single layer, we prefer to keep them separate to highlight our code robustness when dealing with highly deformed multiple layers. As regards the boundary conditions, we apply a vertical free slip condition on the vertical walls and an homogeneous Dirichlet condition on the top and on the bottom of the cube. The results are shown in Fig. 4 which can be directly compared with Figure 8 of [40]: they show a good agreement between the two sets of results. In addition, in Fig. 3(b) we have an outline of the layer geometries of case 1, in an advanced stage: this proves that our implicit tracking algorithm is able to treat highly deformed geometries associated with mature diapirs. Finally, to give an idea of the computational efficiency, we remark that all these test cases require a few hours on an Intel Core2 Duo laptop computer (processor T9300, 2.5 GHz).

6.2. Fault evolution

In this section we consider the 3D kinematic evolution of a faulted basin through two different examples, to highlight the accuracy and the robustness of the methodology described in Section 5.

6.2.1. Fault accuracy

First of all we validate our approach comparing our results with those published in [13]. Since a two-dimensional case is treated there, we choose a proper initial geometrical configuration that limits the effects of the third dimension. In addition,

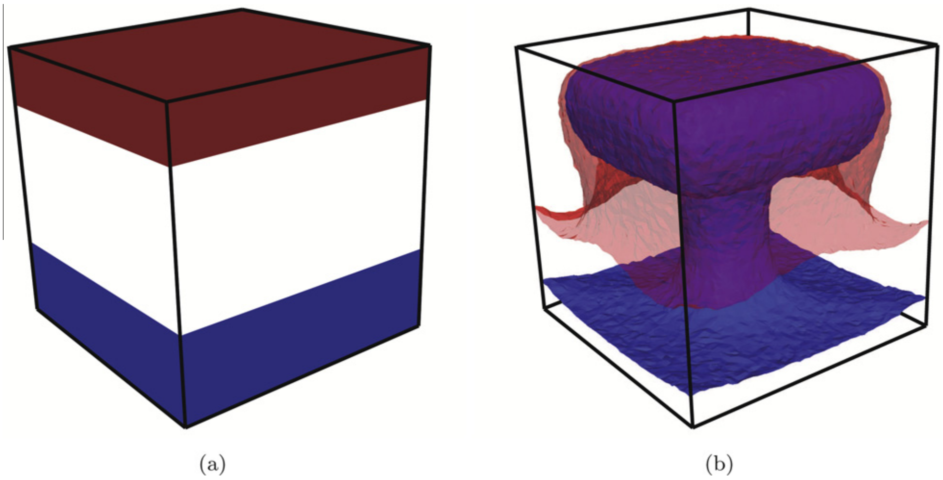


Fig. 3. Rayleigh–Taylor instability simulation. (a) Initial condition. (b) Final configuration: the interfaces between the three layers are depicted.

Table 1
Geometrical and physical configurations for the four test cases. All the values are given in non-dimensional units.

Case	Lower layer			Middle layer			Upper layer		
	Density	Viscosity	Thickness	Density	Viscosity	Thickness	Density	Viscosity	Thickness
1	1	100.0	0.300	1.003	1	0.50	1.003	1	0.2
2	1	0.01	0.300	1.003	1	0.50	1.003	1	0.2
3	1	1.00	0.100	1.003	1	0.70	1.003	1	0.2
4	1	1.00	0.005	1.003	1	0.75	1.003	1	0.2

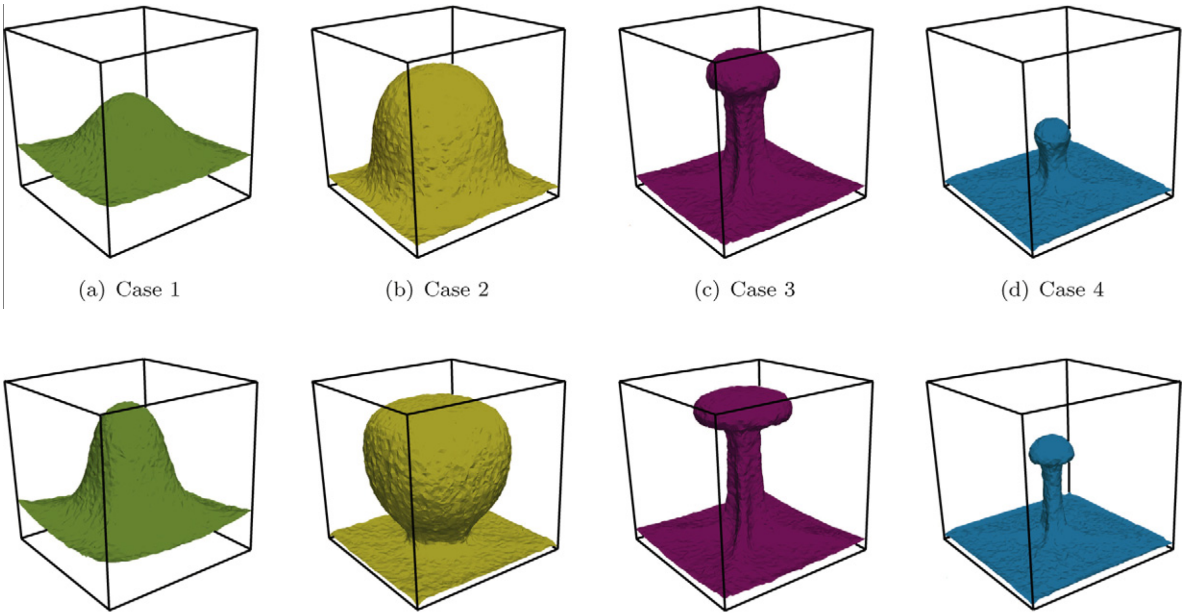


Fig. 4. Two snapshots of the diapir evolution for the four cases of Table 1. As highlighted in [40], the level set technique for the tracking of the sediments induces a slight noise in the representation of the sediment boundaries. This can be processed with some a posteriori regularization algorithms and, in general, does not affect the accuracy of the solution.

the fault is placed in the center cross section, where the distance from the boundary is maximum. The basin has three layers as in the diapir growth test case: the center layer has a rock-like behavior, while the other two have a salt behavior. In Fig. 5 only a portion of the basin is depicted, corresponding to the two upper layers. The results we obtain in Fig. 5(a) and (b) are very similar to those of the extensional case presented in Fig. 3 of [13]. In addition, in Fig. 5(c) and (d) a compressional case is presented.

6.2.2. Fault robustness

To highlight the robustness of our methodology in handling faults and large mesh distortions we impose a strong (and unrealistic) lateral displacement to the boundaries of the domain, by applying a lateral velocity field, through a Dirichlet boundary condition. In addition, to maximize the distortion effects, we set $\mu_i^0 = 10^{22}$ Pa s for all the layers; in other words we consider a consolidated rock behavior, which is quite an unrealistic assumption, since usually the shallower sediments are weaker and less compacted. We also set the sediments density to 3000 kg/m^3 and a Newtonian rheological law. In Fig. 6 we can observe the sediments evolution. In particular Fig. 6(b) proves that even in presence of large distortions the methodology devised in Section 5 is robust. Moreover, from the computational viewpoint, only a couple of remeshings steps are required; they are made automatically when the estimated mesh quality goes beyond a prescribed threshold.

6.3. Realistic basin

The last example we present is a nearly realistic three-layered sedimentary basin with two faults and a salt layer. In presence of salt layers, the above mentioned geological restoration approach is not able (for the time being) to support the

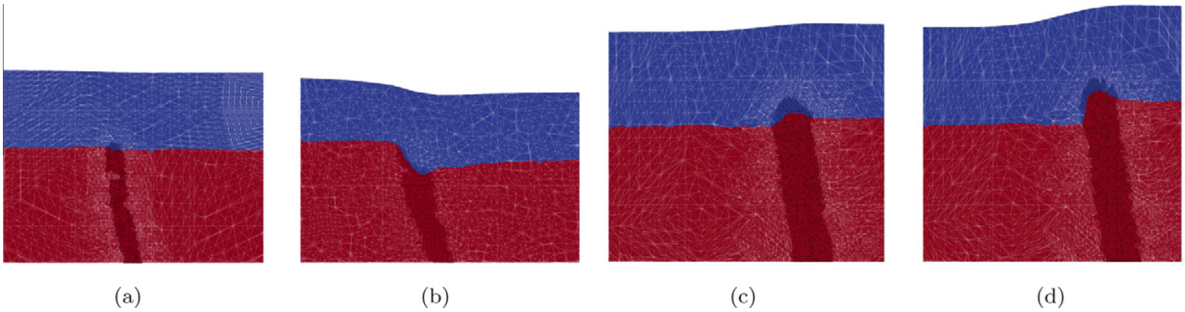


Fig. 5. 2D view of the extensional (a) and (b) and compressional (c) and (d) examples; only the top-central portion of the basins is shown. The blue color represents the salt layer while the red one represents the rock layer. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

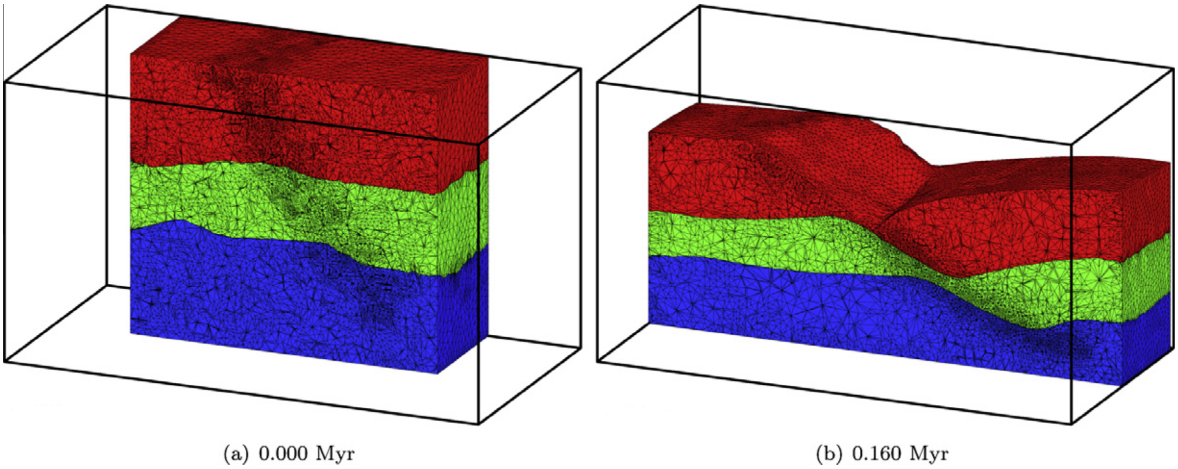


Fig. 6. The evolution of a three-layered faulted basin (vertical cut section at $x = 6204$ km).

geological interpreter in the modeling of the salt evolution. This issue makes the present case extremely interesting, as it represents a good example to show how the methodology devised here can be used to help the interpreter in understanding the salt behaviour. This process provides a more realistic description of the basin salt tectonics, improving the overall petroleum system modeling. Even if the goal of numerical modeling in detail a real 3D basin is not yet achieved, it is worthwhile to test our approach on a nearly realistic case. This would yield more insight about the directions to follow in order to pursue the final goal of a realistic geological dynamic modeling. This example is also important in order to prove the efficiency and the ability of our code to handle more realistic and topologically complicated geological configurations, than just synthetic ones. Here we put together all the ingredients presented in this work, in particular the tracking of the faults and of the subdomain horizons when they are strongly deformed by rapidly rising salt diapirs. Computations have been run on an AMD Opteron 8212 Dual-Core 2 GHz processor; among the several simulation runs, the one we present here has about 52 K nodes corresponding to about 1 million of degrees of freedom requiring approximately 6.2 Gb of RAM memory. The basin extension is $17.0 \times 24.2 \times 7.3$ km^3 and it is subject to an extensional and downward movement; we consider compaction, sedimentation, and we adopt a Carreau rheology model. In Fig. 7 we have outlined the evolution of the sedimentary basin from 55.8 to 5.3 Myr ago, corresponding to three sedimentation events occurred in the geological ages previously introduced. The five snapshots illustrated in Fig. 7 correspond to the five steps resulting from geological restoration interpretation. In absence of a comparison between the modeled numerical dynamic forward evolution and the kinematical inverse restoration, and due to the mentioned limitations of the geological restoration, the results of the dynamical forward modeling must be validated by the geological interpreter according to a suitable geological conceptual model (based on know-how, analogues, etc.), driving in this sense the understanding of the salt behaviour.

At the very beginning of the simulation ($t = 56$ Myr ago, Fig. 7(a)) the model includes three layers labeled as Salt (in yellow[2]), Oligocene (in red), and Eocene (in blue). From this configuration, the simulation advances in time till 24 Myr ago. Then we post-process the resulting configuration, and through the help of a dedicated model editing tool, we add the Early-Miocene layer (in green, see Fig. 7(b)). From this new configuration we restart the simulation advancing till $t = 15.9$ Myr ago, when a new layer, the Mid-Miocene one (in orange), has been added (see Fig. 7(c)). With the same approach, we subsequently add the Late-Miocene layer ($t = 11.3$ Myr ago, cyan in Fig. 7(d)) and the Pliocene layer ($t = 5.1$ Myr ago, magenta in Fig. 7(e)). The layers properties are described in Table 2.

Remark 3. In our approach, the sedimentation process is discrete. In particular, the layers are added as blocks. This is a simplified technique, and represents an approximation of the real sedimentation process, where the sediments are deposited more continuously on the surface. Anyway, the ALE approach can easily handle a continuous sedimentary process for the new layers, whose inclusion in the model will be considered in future works.

A particular care must be paid to the choice of a suitable time stepping restriction. The condition (9) imposes controlled movements to the free surface and to the sediment horizons; in particular, it constrains the maximum oscillations between layers. Moreover, a limit on the maximum time step increment is imposed by the user, according, by now, to the geological scenario. In this simulation a 0.1 Myr maximum time step has been chosen, however, since the dynamic of the basin is relatively fast, this choice has proven to be less restrictive, and the actual simulation timestep has been set by (9).

[2] For interpretation of color in Fig. 7, the reader is referred to the web version of this article.

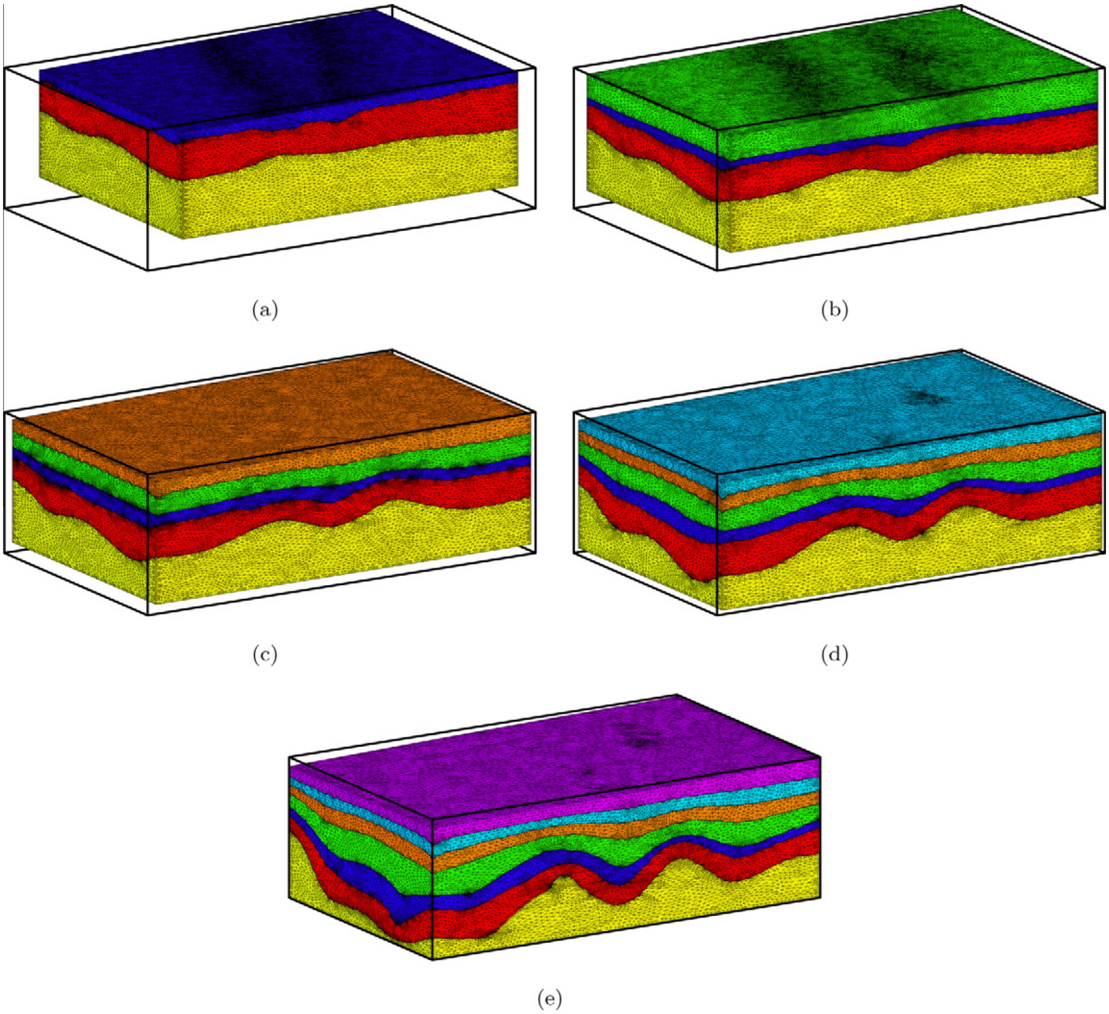


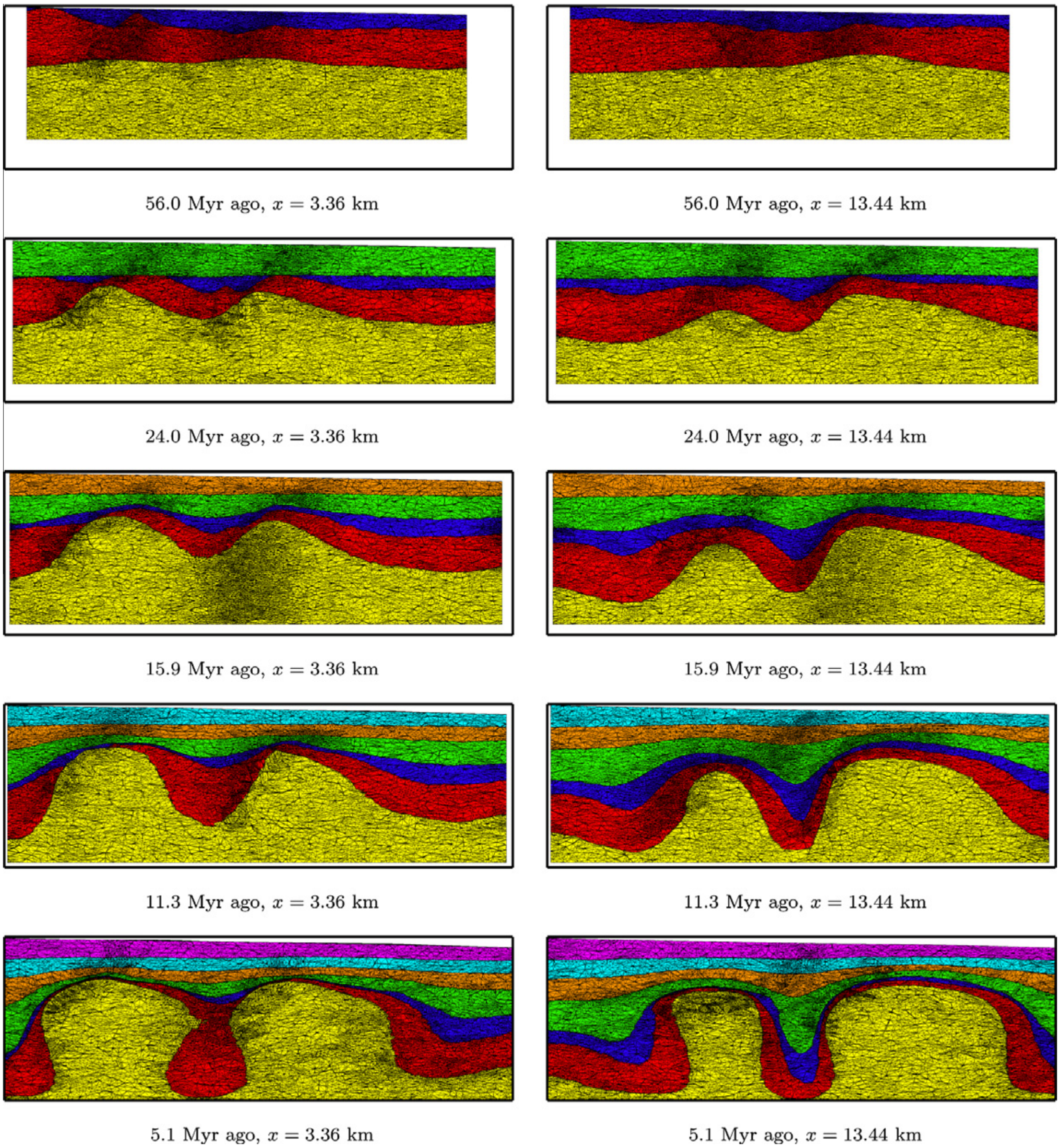
Fig. 7. The evolution of a two-faulted sedimentary basin. (a) 56 Myr ago, 3 layers. (b) 24 Myr ago, 4 layers. (c) 15.9 Myr ago, 5 layers. (d) 11.3 Myr ago, 6 layers. (e) 5.1 Myr ago, 7 layers.

Table 2
Physical characteristics of the layers of the realistic sedimentary basin, with two faults and a salt layer.

Sediment	ρ^0 [10^3 kg/m^3]	μ^0 [10^{20}Pa s]	ϕ^0
Salt	2.15	2	0.05
Oligocene	2.65	205	0.68
Eocene	2.65	210	0.51
Early-Miocene	2.60	205	0.73
Mid-Miocene	2.55	195	0.74
Late-Miocene	2.55	200	0.82
Pliocene	2.50	195	0.83

In Fig. 8 we illustrate a couple of two-dimensional sections of the basin that outline the diapir growth and the overburden deformation. The growth is very slow from 55.8 to 16.15 Myr ago. Then the evolution becomes very rapid and in about ten - Myr the diapirs reach a mature stage.

Almost all the heavier overburden layers are pierced: this outlines the robustness of our approach with respect to topological changes. The growth at 5.3 Myr ago slowed down since most of the upper layers are under-compacted and are lighter that the salt itself this effect flatten upper part of the diapirs. Their shape is compatible with the shapes shown in literature (see [57]).



56.0 Myr ago, $x = 3.36$ km

56.0 Myr ago, $x = 13.44$ km

24.0 Myr ago, $x = 3.36$ km

24.0 Myr ago, $x = 13.44$ km

15.9 Myr ago, $x = 3.36$ km

15.9 Myr ago, $x = 13.44$ km

11.3 Myr ago, $x = 3.36$ km

11.3 Myr ago, $x = 13.44$ km

5.1 Myr ago, $x = 3.36$ km

5.1 Myr ago, $x = 13.44$ km

Fig. 8. A 2D view of two different cutting sections of the sedimentary basin.

7. Conclusions

In this work we have introduced an efficient numerical framework for simulating sedimentary basins evolution, endowed with the modeling of some relevant physical features: compaction, movement of the domain boundaries, faults, and non-Newtonian rheologies. The code has been tested specifically on each features through realistic numerical simulations, to demonstrate its capability to correctly reproduce the physical behaviour. This is coupled with a reliable mathematical support for grid handling and boundary movement. The ALE scheme has proved to be robust and efficient, and only a few remeshings are needed during long-run simulations.

Even if the approach is not yet applicable to real studies, it is however, in our opinion, more appropriate than other approaches present in the literature that are based on the application of already existing general purpose numerical libraries. In the latter case all the effort is put in modifying the geological model characteristics so to fulfil the numerical constraints of the library in use. Instead, in our case we are gradually increasing the capabilities of the proposed numerical approach so to fully respect the geological characteristics and behaviour of the basin model. Although no real study have been performed yet, some test cases regarding relatively small partitions of realistic basins suggest that our technique is promising. Certainly the computational burden when dealing with a real basin is very high, however our scheme has demonstrated an excellent robustness even with very complex geometries. At the same time we are gaining more insight, from the issues that arise in this step by step approach, about the direction to follow for our future developments. Further improvements are foreseen as regards our time-splitting approach, whose architecture allows the substitution of a sub-block quite easily. In particular, the Stokes solver can be improved by a more physical comprehensive approach, including for example the thermal effects on the evolution of the dynamics, and elasticity and plasticity. In particular, these last aspects would allow to predict the formation of faults, that is an essential aspect in many practical cases.

Finally, another important issue is the physical validation of our code. The confidence in the simulations is highly dependent on the precision of the implemented models and on the accuracy of the numerical techniques by which they are solved. A more involving test phase should be carried out at all levels to figure out whether the computed evolution of the basin is coherent with well, seismic and experimental data collected on the ground.

Acknowledgments

This research has been founded by Eni S.p.A.; the advise and collaboration of Dr. Giovanni Scrofani from Eni GEBA Department is kindly acknowledged.

Appendix A. Solution of the fault problem

The solution of problem (23) is

$$\bar{\theta} = (\lambda^{\mathcal{F}} - 1/2)H(\lambda^{\mathcal{F}} - 1/2). \tag{A.1}$$

Proof. We show that every perturbation of the solution (A.1) yields an increase of the functional J. Let us consider a small perturbation of the solution that satisfies the constraints $\theta = \bar{\theta} + \varepsilon\tilde{\theta}$, where ε is a parameter that tends to zero, $\tilde{\theta}$ is a perturbation function that satisfies the constraints in (23). Therefore

$$\tilde{\theta}(\vec{X}) \geqslant \begin{cases} 0 & \text{if } \vec{X} \notin \Omega^{\mathcal{F}} \\ -\frac{\bar{\theta}(\vec{X})}{\varepsilon} & \text{if } \vec{X} \in \Omega^{\mathcal{F}} \end{cases}, \quad \int_{\Omega^{\mathcal{F}}} \tilde{\theta} + \int_{\Omega/\Omega^{\mathcal{F}}} \tilde{\theta} = 0. \tag{A.2}$$

Let us evaluate the functional J in $\bar{\theta} + \varepsilon\tilde{\theta}$:

$$J(\bar{\theta} + \varepsilon\tilde{\theta}) = \frac{1}{2}\int_{\Omega}(\bar{\theta} - \lambda^{\mathcal{F}})^2 + \varepsilon\int_{\Omega}\tilde{\theta}(\bar{\theta} - \lambda^{\mathcal{F}}) + O(\varepsilon^2) = J(\bar{\theta}) + \varepsilon\int_{\Omega^{\mathcal{F}}}\tilde{\theta}(\bar{\theta} - \lambda^{\mathcal{F}}) + \varepsilon\int_{\Omega/\Omega^{\mathcal{F}}}\tilde{\theta}(\bar{\theta} - \lambda^{\mathcal{F}}) + O(\varepsilon^2).$$

From (A.1) we get

$$J(\bar{\theta} + \varepsilon\tilde{\theta}) = J(\bar{\theta}) - \frac{1}{2}\varepsilon\int_{\Omega^{\mathcal{F}}}\tilde{\theta} - \varepsilon\int_{\Omega/\Omega^{\mathcal{F}}}\tilde{\theta}\lambda^{\mathcal{F}} + O(\varepsilon^2),$$

that, combined with the last equation of (A.2), leads to

$$J(\bar{\theta} + \varepsilon\tilde{\theta}) = J(\bar{\theta}) + \varepsilon\int_{\Omega/\Omega^{\mathcal{F}}}\tilde{\theta}\left(\frac{1}{2} - \lambda^{\mathcal{F}}\right) + O(\varepsilon^2).$$

The first order variation is positive as $\lambda^{\mathcal{F}} < 1/2$ and $\tilde{\theta} \geqslant 0$ outside $\Omega^{\mathcal{F}}$, therefore $\bar{\theta}$ is a minimum for J. □

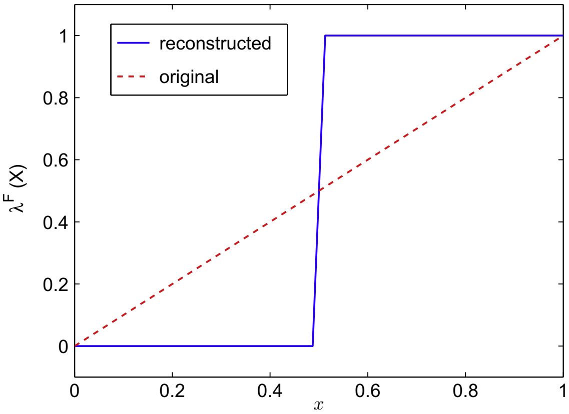


Fig. B.9. The reinitialization algorithm applied once (dashed line) and twice (solid line) to a highly diffused step function.

Appendix B. The reconstruction algorithm applied to a simple case

Here we report a numerical result regarding a simple 1D case. We consider the function $\lambda(x) = x$ that can represent a highly diffused step function. As we are interested in the advection of characteristic functions, we have to deal with its reconstruction. Usually the sharp profile of a step function is diffused by the finite volume scheme and gets much smoother. Clearly, in this case, our minimization problem (23) has an analytical solution that is:

$$\lambda(x) = \begin{cases} 0 & \text{for } x \in [0, 0.5], \\ 1 & \text{for } x \in (0.5, 1]. \end{cases}$$

In Fig. B.9 a comparison between the original function before and after the reinitialization is shown. The algorithm provides a good reconstruction and a conservative behavior as the mass is conserved up to the 1%.

References

[1] M. Longoni, A. Malossi, A. Villa, A robust and efficient conservative technique for simulating three-dimensional sedimentary basins, Comput. Fluids 39 (2010) 1964–1976.

[2] A. Ismail-Zadeh, I. Tsepelev, C. Talbot, A. Korotkii, Three-dimensional forward and backward modelling of diapirism: numerical approach and its applicability to the evolution of salt structures in the Pricaspian basin, Tectonophysics 387 (2004) 81–103.

[3] A. Ismail-Zadeh, D. Krupskii, Extrusion and gravity current of a fluid: Implications for salt tectonics, Phys. Solid Earth 42 (2006) 999–1006.

[4] A. Poliakov, Y. Podladchikov, E. Dawson, C. Talbot, Salt diapirism with simultaneous brittle faulting and viscous flow, Geol. Soc. London Spec. Pub. (Salt Tectonics) 100 (1996) 291–302.

[5] S. Zaleski, P. Julien, Numerical simulation of Rayleigh–Taylor instability for single and multiple salt diapirs, Tectonophysics 206 (1992) 55–69.

[6] I. Collins, Associated and non-associated aspects of constitutive laws for coupled elastic - plastic materials, Int. J. Geomech. 2 (2002) 259–267.

[7] P. van Keken, C. Spiers, A. van den Berg, E. Muyzert, The effective viscosity of rocksalt: implementation of steady-state creep laws in numerical models of salt diapirism, Tectonophysics 225 (1993) 457–476.

[8] D. Turcotte, G. Schubert, Geodynamics, Cambridge University Press, 2001.

[9] D. Peric, A. Crook, Computational strategies for predictive geology with reference to salt tectonics, Comput. Methods Appl. Mech. Eng. 193 (2004) 5195–5222.

[10] L. Lefton, D. Wei, A penalty method for approximations of the stationary power-law Stokes problem, Electron. J. Diff. Equat. 7 (2001) 1–12.

[11] S. Cloetingh, Y. Podladchikov, Perspectives in tectonic modeling, Tectonophysics 320 (2000) 169–173.

[12] P. Massimi, A. Quarteroni, G. Scrofani, An adaptive finite element method for modeling salt diapirism, Math. Mod. Methods Appl. Sci. 16 (2007) 587–614.

[13] P. Massimi, A. Quarteroni, F. Saleri, G. Scrofani, Modeling of salt tectonics, Comput. Methods Appl. Mech. Eng. 197 (2007) 281–293.

[14] J. Bey, Tetrahedral grid refinement, Computing 55 (1995) 355–378.

[15] A. Liu, B. Joe, Quality local refinement of tetrahedral meshes based on bisection, SIAM J. Sci. Comput. 16 (1995) 1269–1291.

[16] A. Liu, B. Joe, Quality local refinement of tetrahedral meshes based on 8-subtetrahedron subdivision, SIAM J. Sci. Comput. 65 (1996) 1183–1200.

[17] A. Plaza, G. Carey, Local refinement of simplicial grids based on the skeleton, Appl. Numer. Math. 32 (2000) 195–218.

[18] W. Wessner, Mesh refinement techniques for TCAD Tools, Ph.D. Thesis, Technische Universität Wien, 2006.

[19] E. Onate, S. Idelsohn, F.D. Pin, R. Aubry, The particle finite element method. An overview, Int. J. Comput. Methods 1 (2004) 267–307.

[20] L. Moresi, F. Dufour, H. Muhlhaus, A Lagrangian integration point finite element method for large deformation modeling of viscoelastic geomaterials, J. Comput. Phys. 184 (2003) 476–497.

[21] C. O'Neill, L. Moresi, D. Muller, R. Albert, F. Dufour, Ellipsis 3D. a particle-in-cell finite-element hybrid code for modelling mantle convection and lithosperic deformation, Comput. Geosci. 1 (2006) 1769–1779.

[22] R. Darlington, T. McAbee, G. Rodrigue, A study of ALE simulations of Rayleigh–Taylor instability, Comput. Phys. Commun. 135 (2001) 58–73.

[23] A. Villa, L. Formaggia, Implicit tracking for multi-fluid simulations, J. Comput. Phys. 229 (2010) 5788–5802.

[24] F. Armero, E. Love, An arbitrary Lagrangian–Eulerian finite element method for finite strain plasticity, Int. J. Num. Methods Eng. 57 (2003) 471–508.

[25] L. Formaggia, F. Nobile, Stability analysis for the arbitrary Lagrangian Eulerian formulation with finite elements, East-West J. Numer. Math. 7 (1999) 105–131.

[26] A. Huerta, A. Rodriguez-Ferran, P. Diez, J. Sarrate, Adaptive finite element strategies based on error assessment, Int. J. Num. Methods Eng. 46 (1999) 1803–1818.

[27] M. Lesoinne, C. Farhat, Geometric conservation laws for flow problems with moving boundaries and deformable meshes, and their impact in aeroelastic computations, Comput. Methods Appl. Mech. Eng. 134 (1996) 71–90.

[28] M. Murayama, K. Nakahashi, K. Matsushima, Unstructured dynamic mesh for large movement and deformation, AIAA Paper 2002-0122, 2002.

[29] J. Peery, D. Carrol, Multi-material ALE methods in unstructured grids, Comput. Methods Appl. Mech. Eng. 187 (2000) 591–619.

[30] R. Smith, Ausm(ale): a geometrical conservative arbitraary Lagrangian–Eulerian flux splitting scheme, J. Comput. Phys. 150 (1999) 268–286.

[31] H. Askes, A. Rodriguez-Ferran, A combined rh-adaptivity scheme based on domain subdivision. Formulation and linear examples, Int. J. Num. Methods Eng. 51 (2001) 253–273.

[32] H. Askes, A. Rodriguez-Ferran, A. Huerta, Adaptive analysis of yeld line patterns in plates with the arbitrary Lagrangian–Eulerian method, Comput. Struct. 70 (1999) 257–271.

[33] A. Masud, M. Bhanabhagvanwala, R. Khurram, An adaptive mesh rezoning scheme for moving boundary flows and fluid-structure interaction, Comput. Fluids 36 (2007) 77–91.

[34] J.F. Aymone, Mesh motion technique for the ALE formulation in 3D large deformation problems, Comput. Methods Appl. Mech. Eng. 59 (2004) 1879–1908.

[35] F. Blom, Consideration on the spring analogy, Int. J. Numer. Methods Fluids 32 (1998) 647–668.

[36] C. Bottasso, D. Detomi, R. Serra, The ball-vertex method: a new simple spring analogy method for unstructured dynamic meshes, Comput. Methods Appl. Mech. Eng. 194 (2005) 4244–4264.

[37] A. Johnson, T. Tezduyar, Mesh update strategies in parallel finite element computations of flow problems with moving boundaries and interfaces, Comput. Methods Appl. Mech. Eng. 119 (1994) 73–94.

[38] D. Audet, A. Fowler, A mathematical model for compaction in sedimentary basins, Geophys. J. Int. 110 (1992) 577–590.

[39] S. Zlotnik, P. Diez, Hierarchical X-FEM for n-phase flow ($n > 2$), Comput. Methods Appl. Mech. Eng. 198 (2009) 2329–2338.

[40] S. Zlotnik, M. Fernandez, P. Diez, J. Verges, Modelling gravitational instabilities: slab break-off and Rayleigh–Taylor diapirism, Pure Appl. Geophys. 165 (2008) 1491–1510.

[41] A. Fowler, X. Yang, Fast and slow compaction in sedimentary basins, SIAM J. Appl. Math. 59 (1998) 365–385.

[42] J.C. Neto, P. Wawrzynek, M. Carvalho, L. Martha, A. Ingraffea, An algorithm for three-dimensional mesh generation for arbitrary regions with cracks, Eng. Comput. 17 (2001) 75–91.

[43] A. Quarteroni, Numerical Models for Differential Problems, Springer–Verlag, 2009.
[44] A. Quarteroni, A. Valli, Numerical Approximation of Partial Differential Equations, Second ed., Springer–Verlag, 1997.
[45] D. May, L. Moresi, Preconditioned iterative methods for Stokes flow problems arising in computational geodynamics, Phys. Earth Planet In. 171 (2008) 33–47.
[46] L. Moresi, S. Zhong, M. Gurnis, The accuracy of finite element solution of Stokes flow with strongly varying viscosity, Phys. Earth Planet In. 97 (1996) 83–94.
[47] M. Olshanskii, A. Reusken, Analysis of a Stokes interface problem, Numer. Math. 103 (2006) 129–149.
[48] W. Huang, Anisotropic mesh adaptation and movement, Lecture Notes for the Workshop on: Adaptive Method, Theory and Application, Peking University, Beijing, China, 2005.
[49] W. Huang, W. Sun, Variational mesh adption 2: error estimates and monitor functions, J. Comput. Phys. 184 (2003) 619–648.
[50] R. Adams, J. Fournier, Sobolev Spaces, Elsevier Science, 2003.
[51] A. Ern, J. Guermond, Theory and Practice of Finite Elements, Springer, 2000.
[52] T. Brochu, R. Bridson, Robust topological operations for dynamic explicit surfaces, SIAM J. Sci. Comput. 31 (2009) 2472–2493.
[53] S. Osher, R. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, Springer, 2003.
[54] A. Sethian, Level Set Methods and Fast Marching Methods, Cambridge University Press, 1999.
[55] P. Ciarlet, Introduction to Numerical Linear Algebra and Optimization, Cambridge University Press, 1988.
[56] J. Lions, Optimal Control of Systems Governed by Partial Differential Equations, Springer, 1971.
[57] A. Ismail-Zadeh, C. Talbot, Y. Volozh, Dynamic restoration of profiles across diapiric salt structures: numerical approach and its applications, Tectonophysics 337 (2001) 21–36.