MOX–Report No. 05/2010

A complete model for non-Newtonian sedimentary basins in presence of faults and compaction phenomena

Matteo Longoni, A.C.I. Malossi, Alfio Quarteroni, Andrea Villa

MOX, Dipartimento di Matematica "F. Brioschi"
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

mox@mate.polimi.it http://mox.polimi.it

A complete model for non-Newtonian sedimentary basins in presence of faults and compaction phenomena. *

M. Longoni [♯], A. C. I. Malossi [♮], A. Quarteroni [§], A. Villa [♭]

January 30, 2010

[♯] MOX– Modellistica e Calcolo Scientifico
Dipartimento di Matematica "F. Brioschi"
Politecnico di Milano
via Bonardi 9, 20133 Milano, Italy
matteo.longoni@mate.polimi.it

[♮] Chair of Modelling and Scientific Computing (CMCS)
Mathematics Institute of Computational Science and Engineering (MATHICSE)
École Polytechnique Fédérale de Lausanne
Station 8, CH-1015, Lausanne, Switzerland
cristiano.malossi@epfl.ch

[§] MOX– Modellistica e Calcolo Scientifico
Dipartimento di Matematica "F. Brioschi"
Politecnico di Milano
via Bonardi 9, 20133 Milano, Italy
and Chair of Modelling and Scientific Computing (CMCS)
Mathematics Institute of Computational Science and Engineering (MATHICSE)
École Polytechnique Fédérale de Lausanne
Station 8, CH-1015, Lausanne, Switzerland
alfio.quarteroni@epfl.ch

[♭] ENEA Ricerca sul Sistema Elettrico (ERSE)
Via Rubattino 54, 20134 Milano, Italy
andrea.villa@erse-web.it

Keywords: Computational geology, Fractures and faults, Non-Newtonian fluids, Compaction of granular systems.

PACS Subject Classification: 45.70.Cc, 47.11.Fg, 47.50.-d, 83.80.Nb, 91.55.Jk

*This work has been supported by the Project: Steam3D - ENI

Abstract

In this paper we present a model and a complete numerical tool for simulating the three-dimensional dynamics of realistic stratified sedimentary basins. We developed dedicated mathematical algorithms to include most of the key physical aspects such as the movement of the basement, the non-Newtonian behavior of the sediments, the effects of faults and the presence of compaction phenomena. This approach is mandatory to capture all the three-dimensional effects of realistic evolution dynamics, whose duration is about millions of years and to provide reliable results. In this work we apply our methods to a realistic and topologically complex sedimentary system.

1 Numerical models for advanced basin simulations

The numerical simulation of the geological evolution of a sedimentary basin represents a challenging aspect of applied numerical mathematics. Many physical aspects such as the compaction, the basement movement, the rheology, and reciprocal interactions between the sediment layers should be taken into account for a reliable simulation. In some cases most of these aspects have been considered, see [30], while in many other cases (especially in 3D see [22, 28]), only some simplified approaches have been implemented. In this work we aim at creating a complete numerical tool that includes the physical aspects analyzed in [30], extended to a three-dimensional realistic framework. We model the sediments with a classical stratified creeping flow problem. In particular, we complement the techniques already developed in [28] to include the non-Newtonian rheology of the sediments, the faults, the compaction, the movement of the basement and of the free surface. Although, a complete physical validation of the numerical scheme is beyond the scope of this work, yet we show some numerical results that are qualitatively consistent with the observed configuration of sediment layers.
Let's consider now the innovations added in the model. First of all we introduce the non-Newtonian rheology. A complete and exhaustive theoretical analysis of the sediment rheology is still missing. Experiments have shown different behaviors of the sediments, such as elastic, elasto-plastic, visco-plastic and visco-elastic, see [11, 48]. Semi-empirical relations are widespreadly adopted, since the theory can explain only a few mechanisms (an example is the viscous-fluid rheology of a crystalline structure, see [47]). In [40] a tensor splitting technique is exploited to adapt the simulation to almost every type of isotropic rheology. Here we concentrate on visco-plastic rheologies, as they account for the two main deformation mechanisms on geological timescales and we completely neglect the elastic behavior, as it is often related to shorter-period phenomena such as earthquakes. This approach has already been considered in [30] but, in this work, we consider a

broader choice of pseudo-plastic rheologies: the Carreau, Cross, Powell, and Yeleswarapu relations and we figure out their impact on the final geological timing and geometry, with regards to the stress field.
Let's 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 [10], or even neglected (see [22, 40] and [52]). In [31] a new splitting algorithm is introduced: the divergence of the solid flow field is computed according to some experimental compaction curves. 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 directly the modeling of the compaction function, i.e., the function that measures the rate of decrease of the solid volume (see [33]) 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, only in a few works it has been considered (see [29, 30, 40] and [43]). 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. Faults are fracture zones where damaged rock creates sliding planes. A possible way to model them as a fluid 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 [30]). This approach has two potential disadvantages: 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 [5, 26, 27, 41] and [51].
Let's 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 as the surrounding soil moves with Earth plates. 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 [22] the basin is a fixed box. This geometrical constraint is not acceptable for many applications and several works, such as [30, 31, 40], 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 [36] and applied to computational geology [34, 37]. 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 [13]. 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 [50]). 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 [1, 16, 21, 24, 35, 39] and [46]. 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 [2]) 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 [2, 3, 32]. 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 [15]). The second method, widespreadly adopted in aeroelastic analysis, is the spring method (see [7, 8]). 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 [23]). 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 give an overview on the geometric and mathematical model of the sedimentary basin, together with some physical key features of the problem. A temporal time splitting technique is introduced in 3, and the discretization and the spaces related to the physical variables are introduced in Section 4. Then in Sections 5 and 6 we deal with the two main new features we have introduced, the adaptive grid movements algorithm and the faults modeling. Finally some numerical results obtained from the simulation of a realistic case, are presented.

2 Physical and mathematical models

2.1 Nomenclature

Let's introduce the geometric model of a sedimentary basin (Figure 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's introduce some nomenclature: $\vec{X} = (x_1, x_2, x_3) \in \Omega$ indicates a point in the spatial domain of coordinates x_i, with $i = 1, 2, 3$, $(\widehat{x}_1, \widehat{x}_2, \widehat{x}_3)$ 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 Rheological models

Rocks and sediments usually exhibit a wide range of rheological behaviors. In a basin scale framework we find convenient to adopt some pseudo-plastic relations since they embed the fluid and plastic behaviors. These are the

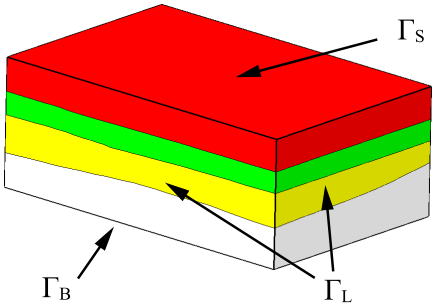


Figure 1: External shape of the domain Ω. The external boundary Γ is divided into three parts: the basement Γ_B, the free surface Γ_S, and the lateral contour Γ_L.

most relevant effects on the geological time scales, but no laws have been derived in literature specifically for rocks at such a scale. Anyway rocks behavior has proved to be far more complicated than a Newtonian one, so it comes necessary to consider non-Newtonian relationships, see [40]. The pseudo-plastic relations we have considered are well known and widely adopted in different fields, such as the simulation of blood flow (that also shows shear thinning), and provide reliable results in several applications. These laws are empirically derived and have to be fitted case by case, setting the depending parameter according to the considered fluid, though in this field reliable values are difficult to be validated. All of them are generalized Newtonian laws with a stress dependent viscosity. We have considered five different rheological relations for the apparent viscosity μ_i:

$$\mu_i = \begin{cases} \mu_i^\infty = \mu_i^0 & \text{Newton,} \\ \mu_i^\infty + \dfrac{\mu_i^0 - \mu_i^\infty}{(1+\xi_i\gamma)^{(2-r)/r}} & \text{Carreau,} \\ \mu_i^\infty + \dfrac{\mu_i^0 - \mu_i^\infty}{1+(\xi_i\gamma)^r} & \text{Cross,} \\ \mu_i^\infty + (\mu_i^0 - \mu_i^\infty)\dfrac{\sinh^{-1}(\xi_i\gamma)}{\xi_i\gamma} & \text{Powell,} \\ \mu_i^\infty + (\mu_i^0 - \mu_i^\infty)\dfrac{1+\ln(1+\xi_i\gamma)}{1+\xi_i\gamma} & \text{Yeleswarapu,} \end{cases} \tag{1}$$

where μ_i^0 is the reference unstressed viscosity, μ_i^∞ is the asymptotic viscosity for $\gamma \to \infty$, with γ the squared Frobenius norm of the symmetric gradient $\vec{\nabla}\vec{u} + (\vec{\nabla}\vec{u})^T$; moreover $\vec{u}$ is the macroscopic velocity of the rocks, r is a positive coefficient, also known as the power law coefficient, and ξ_i is a material dependent parameter. For some reference see [43, 44]. We can now

define an averaged viscosity as

$$\mu = \psi \sum_{i=1}^{n_s} \lambda_i \mu_i, \tag{2}$$

where λ_i are the characteristic functions of the layers and $\psi(t, \vec{X}) \leq 1$ is the viscosity abatement function and it takes into account the possible presence of faults (a complete description of the fault model is provided in Section 6). In other terms $\psi(t, \vec{X}) = 1$ outside the faulted region while $\psi << 1$ in the faulted region. The stress tensor is, then, given by

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

where μ is a function of the shear stress.

2.3 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 also called the compaction function, see [33].
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 $\widehat{x}_3$ 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 Figure 2).

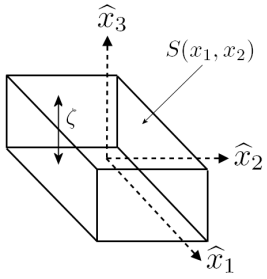


Figure 2: An outline of the sedimentary basin with a reference frame having axes $(\widehat{x}_1, \widehat{x}_2, \widehat{x}_3)$. The surface position is indicated with $S(x_1, x_2)$, the depth with ζ.

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

$$\phi = \exp(\mathcal{B}\zeta) \left(\sum_{i=1}^{n_s} \lambda_i \phi_i^0 \right), \tag{3}$$

where ϕ_i^0 is the reference porosity of the i-th layer and $\mathcal{B}$ is an empirical constant. We wish to determine Φ through (3). From [43] we get

$$\vec{\nabla} \cdot \vec{u} = \frac{1}{1-\phi} \frac{D\phi}{Dt}, \tag{4}$$

where $D/Dt = \partial/\partial t + \vec{u} \cdot \vec{\nabla}$ is the material derivative. Plugging equation (3) into (4) yields

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

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, $\mathcal{B}\left(\phi/(1-\phi)\right) u_3$, 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 by the geologists on the basis of some conjectures on the history of the basin.

2.4 The model

Now we can summarize the complete model in the advective ALE form, including all the new features we have introduced so far

$$\begin{cases} \vec{\nabla} \cdot \bar{\bar{\sigma}}(\mu, \vec{u}) - \vec{\nabla} P + \rho \vec{g} = 0 & \text{in} \quad \Omega \times [0,T], \\ \vec{\nabla} \cdot \vec{u} = \Phi(\phi) & \text{in} \quad \Omega \times [0,T], \\ \dfrac{\partial \lambda_i}{\partial t} + (\vec{u} - \vec{u}_g) \cdot \vec{\nabla} \lambda_i = 0 & \text{in} \quad \Omega \times (0,T], \\ \dfrac{\partial \psi}{\partial t} + (\vec{u} - \vec{u}_g) \cdot \vec{\nabla} \psi = 0 & \text{in} \quad \Omega \times (0,T], \\ \bar{\bar{\sigma}} = \mu(\vec{u}, \psi, \xi_i)(\vec{\nabla} \vec{u} + (\vec{\nabla} \vec{u})^T) & \text{in} \quad \Omega \times (0,T], \\ \lambda_i = \overline{\lambda}_i, \quad \psi = \overline{\psi} & \text{in} \quad \Omega \times \{0\}, \\ \vec{u} = \overline{u} & \text{on} \quad \Gamma_B, \\ (\bar{\bar{\sigma}} - P\bar{\bar{I}}) \cdot \widehat{n} = 0 & \text{on} \quad \Gamma_S, \\ u_1 = \overline{u}_1, \quad u_2 = \overline{u}_2, \quad ((\bar{\bar{\sigma}} - P\bar{\bar{I}}) \cdot \widehat{n}) \cdot \widehat{x}_3 = 0 & \text{on} \quad \Gamma_L, \end{cases} \tag{5}$$

where P is the total pressure of the solid-fluid mixture, $\vec{g}$ is the gravitational acceleration vector, $\overline{\lambda}_i$ and $\overline{\psi}$ are a suitable set of initial conditions, $\overline{u}$ is

a boundary velocity field, and $\vec{u}_g$ is the grid velocity linked to the ALE treatment of the boundary movement. Moreover the density ρ is given by

$$\rho = \phi\rho_f + (1-\phi)\left(\sum_{i=1}^{n_s}\lambda_i\rho_i\right), \tag{6}$$

where ρ_f is the fluid density and ρ_i is the reference density of each sedimentary layer. The system above consists of two evolution equations: the fault tracking function equation and the partial volume equation. For the latter we use an implicit tracking method, for details see [50], while for the fault tracking we employ a modified level set method, to be discussed in detail in Section 6. The remaining equations form a Stokes problem with non-solenoidal velocity, which is approximated using a finite element discretization and solved by a preconditioned iterative scheme, see [28].

3 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 n-th 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 the following scheme:

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

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

Then we use the following numerical scheme to solve problem (5):

1. *from λ_i^n and relation (3) compute ϕ^n;*
2. *from λ_i^n, ϕ^n and relation (6) compute ρ^n;*
3. *from λ_i^n, ψ^n, $\vec{u}^{n-1}$ and relations (1) and (2) compute μ^n.*
   ***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*

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

   where m is the iteration apex and $\vec{u}^{(0,0)}$ has been set equal to 0;
4. *solve a linearized Stokes problem for $\vec{u}^n$ and P^n:*

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

5. *a time step Δt^n is computed. This requires three steps: firstly, from the velocity $\vec{u}^n$, a maximum $\Delta t^n_{\mathrm{ALE}}$ is estimated such that the movement of the boundary elements does not exceed a certain threshold; secondly a maximum $\Delta t^n_{\mathcal{C}}$ 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*

$$\Delta t^n = \min(\Delta t^n_{\mathrm{ALE}}, \Delta t^n_{\mathcal{C}}); \tag{7}$$

6. *from Δt^n and the adaptive grid movement algorithm (see Section 5) the grid velocity $\vec{u}^n_g$ is evaluated;*

7. *compute λ_i^{n+1} and ψ^{n+1} by solving the following evolution equations in $(t^n, t^{n+1}]$:*

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

8. *with the grid velocity $\vec{u}^n_g$ 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:

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

4 Spatial discretization

To describe the spatial discretization scheme, we introduce the geometric approximation of the domain Ω. Let $\mathcal{T}^G_\Delta$ be a simplicial tetrahedral grid containing n^G_e elements e^G_r (with $r = 1, \ldots, n^G_e$) and n^G_p nodes $\vec{x}^G_k$ (with $k = 1, \ldots, n^G_p$), where the subscript Δ stands for the maximum diameter of the grid elements. From $\mathcal{T}^G_\Delta$ we build the mini-grid $\mathcal{T}^M_\Delta$ by adding n^G_e barycentric nodes; hence $\mathcal{T}^M_\Delta$ has $n^M_p = n^G_p + n^G_e$ nodes $\vec{x}^M_k$ and $n^M_e = 4n^G_e$ elements e^M_r. Moreover, with a uniform refinement of $\mathcal{T}^G_\Delta$ carried out for N_R-times, we create a conformal grid $\mathcal{T}^S_\Delta$, which has $n^S_e = 8N_R n^G_e$ elements e^S_r and n^S_p nodes $\vec{x}^S_k$. In the following we will refer to $\mathcal{T}^G_\Delta$ as the grid, to $\mathcal{T}^M_\Delta$ as the mini-grid, and to $\mathcal{T}^S_\Delta$ as the sub-grid. Let $\mathcal{F}^G_{r,j}$, $\mathcal{F}^M_{r,j}$ and $\mathcal{F}^S_{r,j}$ with $j = 1, \ldots, 4$ be the set of the four faces surrounding the r-th tetrahedron of the grid, mini-grid and sub-grid respectively. The element sharing the face $\mathcal{F}^S_{r,j}$ with the element e^S_r will be denoted with $e^S_{r_j}$. We also define the map $(r, j) \to \bar{j}$, such that, given the indices r and j, $\bar{j} : \mathcal{F}^S_{r_j,\bar{j}} = \mathcal{F}^S_{r,j}$ (see

Figure 3). In other words, every common face between the element e_r^S and its neighbors $e_{r_j}^S$ is identified by $\mathcal{F}_{r,j}^S$ or $\mathcal{F}_{r_j,\bar{j}}^S$ depending whether it is a face of e_r^S or $e_{r_j}^S$. Finally let's introuce the discrete variables and introduce the

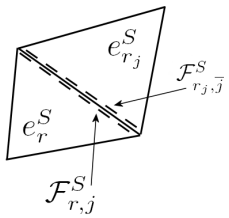


Figure 3: The element e_r^S and its j-th neighbor $e_{r_j}^S$. The j-th face of the element e_r^S is $\mathcal{F}_{r,j}^S$ and corresponds to the $\bar{j}$-th face of $e_{r_j}^S$, that is $\mathcal{F}_{r,j}^S = \mathcal{F}_{r_j,\bar{j}}^S$.

related discrete spaces:

$$
\begin{aligned}
\vec{u}_\Delta^n \in \overline{\mathbb{V}}_1^M : \overline{\mathbb{V}}_1^M &= \left\{ \vec{\varphi}_\Delta \in \mathbb{C}^0(\Omega) : \vec{\varphi}_\Delta|_{e_r^M} \in \mathbb{P}^1, \quad r = 1, \dots, n_e^M \right\}, \\
P_\Delta^n \in \mathbb{V}_1^G : \mathbb{V}_1^G &= \left\{ \varphi_\Delta \in \mathbb{C}^0(\Omega) : \varphi_\Delta|_{e_r^G} \in \mathbb{P}^1, \quad r = 1, \dots, n_e^G \right\}, \\
\bar{\bar{\sigma}}_\Delta^n \in \overline{\overline{\mathbb{W}}}_0^M : \overline{\overline{\mathbb{W}}}_0^M &= \left\{ \bar{\bar{\varphi}}_\Delta \in L^2(\Omega) : \bar{\bar{\varphi}}_\Delta|_{e_r^M} \in \mathbb{P}^0, \quad r = 1, \dots, n_e^M \right\}, \\
\vec{u}_{g,\Delta}^n \in \overline{\mathbb{V}}_1^G : \overline{\mathbb{V}}_1^G &= \left\{ \vec{\varphi}_\Delta \in \mathbb{C}^0(\Omega) : \vec{\varphi}_\Delta|_{e_r^G} \in \mathbb{P}^1, \quad r = 1, \dots, n_e^G \right\}, \\
\rho_\Delta^n,\, \mu_\Delta^n,\, \phi_\Delta^n \in \mathbb{W}_0^G : \mathbb{W}_0^G &= \left\{ \varphi_\Delta \in L^2(\Omega) : \varphi_\Delta|_{e_r^G} \in \mathbb{P}^0, \quad r = 1, \dots, n_e^G \right\}, \\
\psi_\Delta^n \in \mathbb{W}_0^S : \mathbb{W}_0^S &= \left\{ \varphi_\Delta \in L^2(\Omega) : \varphi_\Delta|_{e_r^S} \in \mathbb{P}^0, \quad r = 1, \dots, n_e^S \right\}, \\
\lambda_{i,\Delta}^n \in \mathbb{V}_0^S : \mathbb{V}_0^S &= \{ \varphi_\Delta \in L^2(\Omega) : \varphi_\Delta|_{\tau_k^S} \in \mathbb{P}^0, \quad k = 1, \dots, n_p^S \},
\end{aligned}
\tag{8}
$$

where τ_k^S are the cells of the dual sub-grid (see [50] for more details on the geometrical construction of the dual grid). In (8) the notation follows these conventions, $\mathbb{V}$ and $\mathbb{W}$ denote a finite element space with, respectively, node-related and cell-related degrees of freedom (DOF), the suffixes G, M, S refer to the grid discrete space, and the indices 0 and 1 indicate the degree of the basis.

5 Adaptive grid movement

5.1 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.

More precisely, 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 $\vec{S}^n : \Omega^n \to \Omega^{n+1}$ such that

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

To implement our mesh movement-adaption scheme, we adopt the ideas of [19], [20], that is we seek a best fit solution in $H^1(\Omega)$ to the alignment and to the equal distribution condition

$$(\vec{\nabla}\vec{X})^T \cdot \bar{\bar{M}} \cdot (\vec{\nabla}\vec{X}) = \bar{\bar{I}} \left(\frac{1}{|\Omega|} \int_\Omega \sqrt{\det(\bar{\bar{M}})} \right)^{2/3}, \tag{9}$$

with

$$\int_\Omega \sqrt{\det(\bar{\bar{M}})} = |\Omega|, \tag{10}$$

where $\bar{\bar{M}}$ 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 $t = 0$. A best fit solution of the alignment condition is given by the minimization of

$$\min_{\vec{X} \in H^1(\Omega)} \frac{1}{2} \|(\vec{\nabla}\vec{X})^T \cdot \bar{\bar{M}} \cdot (\vec{\nabla}\vec{X}) - \bar{\bar{I}}\|^2_{\bar{\bar{\mathcal{D}}}}, \tag{11}$$

where $\|\bar{\bar{\mathcal{C}}}\|^2_{\bar{\bar{\mathcal{D}}}} = \int_{\Omega^n} \bar{\bar{\mathcal{C}}} \cdot \bar{\bar{\mathcal{D}}} \cdot \bar{\bar{\mathcal{C}}}$, $\bar{\bar{\mathcal{D}}}$ 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 [2] and [23] we choose $E_L|_{e_r} = K_L|_{e_r} = 1/|e^G_r|$. The optimality conditions of (11) 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 (11) with respect to the coordinate system Y^n is

$$\min_{\vec{S}^n \in H^1(\Omega^n)} \frac{1}{2} \int_{\Omega^n} \bar{\bar{\epsilon}}^n \cdot \bar{\bar{\mathcal{D}}} \cdot \bar{\bar{\epsilon}}^n + \int_{\Omega^n} \bar{\bar{\sigma}}^n_0 : \bar{\bar{\epsilon}}^n, \tag{12}$$

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

$$\bar{\bar{\epsilon}}^n = \frac{1}{2} \left((\vec{\nabla}\vec{X}^n)^T \cdot \bar{\bar{M}}^n \cdot \vec{\nabla}\vec{X}^n - \bar{\bar{I}} \right), \tag{13}$$

is the strain tensor. Plugging

$$\vec{X}^n = \vec{Y}^n + \vec{S}^n \tag{14}$$

into (13) we get

$$\bar{\bar{\epsilon}}^n = \underbrace{\frac{1}{2}\left((\vec{\nabla}\vec{S}^n)^T \cdot \bar{\bar{M}}^n \cdot (\vec{\nabla}\vec{S}^n)\right)}_{\bar{\bar{\epsilon}}_2^n} + \underbrace{\frac{1}{2}\left((\vec{\nabla}\vec{S}^n)^T \cdot \bar{\bar{M}}^n + \bar{\bar{M}}^n \cdot (\vec{\nabla}\vec{S}^n)\right)}_{\bar{\bar{\epsilon}}_1^n} + \underbrace{\frac{1}{2}\left(\bar{\bar{M}}^n - \bar{\bar{I}}\right)}_{\bar{\bar{\epsilon}}_0^n}.$$

By differentiating equation (12) with respect to the displacement we obtain

$$\int_{\Omega^n} (\bar{\bar{w}}_1 + \bar{\bar{w}}_2) \cdot \bar{\bar{\mathcal{D}}} \cdot (\bar{\bar{\epsilon}}_0^n + \bar{\bar{\epsilon}}_1^n + \bar{\bar{\epsilon}}_2^n) + \int_{\Omega^n} \bar{\bar{\sigma}}_0^n : (\bar{\bar{w}}_1 + \bar{\bar{w}}_2) = 0 \qquad \forall \vec{v} \in H^1(\Omega),$$

where

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

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

$$\int_{\Omega^n} \bar{\bar{w}}_1 \cdot \bar{\bar{\mathcal{D}}} \cdot \bar{\bar{\epsilon}}_1^n + \int_{\Omega^n} (\bar{\bar{\sigma}}_0^n + \bar{\bar{\mathcal{D}}} \cdot \bar{\bar{\epsilon}}_0^n) : (\bar{\bar{w}}_1 + \bar{\bar{w}}_2) = 0 \qquad \forall \vec{v} \in H^1(\Omega).$$

For simplicity, we set $\bar{\bar{\Sigma}}_0^n = \bar{\bar{\sigma}}_0^n + \bar{\bar{\mathcal{D}}} \cdot \bar{\bar{\epsilon}}_0^n$. Integrating by parts, we obtain the following balance equation for the displacements

$$\begin{aligned} -\vec{\nabla} \cdot \Big[E_L(\vec{\nabla}\vec{S}^n : \bar{\bar{M}}^n)\bar{\bar{I}} \;&+\; K_L\left((\vec{\nabla}\vec{S}^n)^T \cdot \bar{\bar{M}}^n + \bar{\bar{M}}^n \cdot (\vec{\nabla}\vec{S}^n)\right) \\ &+\; \bar{\bar{\Sigma}}_0^n \cdot (\vec{\nabla}\vec{S}^n + \bar{\bar{I}}) \cdot \bar{\bar{M}}^n \Big] = 0, \end{aligned} \tag{15}$$

which is complemented by the following set of boundary conditions (distributed as in Figure 4):

$$\begin{cases} \vec{S}^n = \vec{u}_\Delta^n \Delta t^n & \text{on} \quad \Gamma_B, \\ S_1^n = u_{\Delta,1}^n \Delta t^n, \quad S_2^n = u_{\Delta,2}^n \Delta t^n, \quad (\bar{\bar{\Sigma}}_0^n \cdot \vec{n}) \cdot \widehat{x}_3 = 0 & \text{on} \quad \Gamma_L, \\ \vec{S}^n \cdot \vec{n} = (\vec{u}_\Delta^n \cdot \vec{n})\Delta t^n, \quad (\bar{\bar{\Sigma}}_0^n \cdot \vec{n}) \cdot \vec{n} = 0 & \text{on} \quad \Gamma_S. \end{cases}$$

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 we introduce now the discrete weak formulation, i.e., we seek $\vec{S}_\Delta^n \in \overline{\mathbb{V}}_1^G \cap H_\Gamma^1$ such that:

$$a_{\mathrm{ALE}}(\vec{S_\Delta}^n, \vec{v}_\Delta) = F_{\mathrm{ALE}}(\vec{v}_\Delta) \quad \forall \vec{v}_\Delta \in \overline{\mathbb{V}}_1^G \cap H_\Gamma^1, \tag{16}$$

where H_Γ^1 is the set of the functions $\vec{v}$ belonging to H^1 such that

$$\begin{cases} \vec{v} = \vec{u}_\Delta^n \Delta t^n & \text{on} \quad \Gamma_B, \\ v_1 = u_{\Delta,1}^n \Delta t^n, \quad v_2 = u_{\Delta,2}^n \Delta t^n & \text{on} \quad \Gamma_L, \\ \vec{v} \cdot \vec{n} = (\vec{u}_\Delta^n \cdot \vec{n})\Delta t^n & \text{on} \quad \Gamma_S. \end{cases}$$

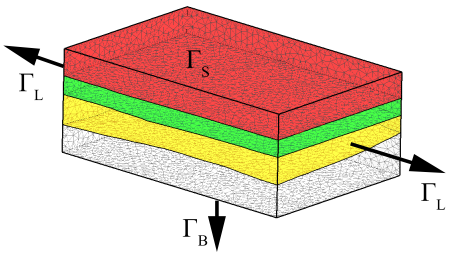


Figure 4: An open three-dimensional view of a sedimentary basin with the boundary conditions distribution and their physical effects.

Moreover $\vec{S_\Delta}^n \in \overline{\mathbb{V}}_1^G$ is the discrete counterpart of $\vec{S}^n$ and

$$\begin{cases} a_{\mathrm{ALE}}(\vec{S_\Delta}^n, \vec{v}_\Delta) = \displaystyle\int_{\Omega^n} \Big[E_L(\vec{\nabla}\vec{S}^n : \bar{\bar{M}}^n)\bar{\bar{I}} \quad + K_L \left((\vec{\nabla}\vec{S}^n)^T \cdot \bar{\bar{M}}^n + \bar{\bar{M}}^n \cdot (\vec{\nabla}\vec{S}^n) \right) \\ \qquad\qquad\qquad\qquad\qquad\qquad + \bar{\bar{\Sigma}}_0^n \cdot \vec{\nabla}\vec{S}^n \cdot \bar{\bar{M}}^n \Big] : \vec{\nabla}\vec{v}_\Delta, \\ F_{\mathrm{ALE}}(\vec{v}_\Delta) = -\displaystyle\int_{\Omega^n} \bar{\bar{\Sigma}}_0^n \cdot \bar{\bar{M}}^n. \end{cases}$$

If $K_L, E_L > 0$, problem (16) is coercive. Moreover if $\bar{\bar{\Sigma}}_0^n, \bar{\bar{M}}^n, K_L, E_L \in L^\infty(\Omega)$, then the bilinear form $a_{\mathrm{ALE}}(\cdot,\cdot)$ and the linear functional $F_{\mathrm{ALE}}(\cdot)$ are bounded and problem (16) has a unique solution (see [14] and [42]).
The pre-stress term is updated as follows

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

Once the displacement field is computed, the grid speed is calculated as

$$\vec{u}_{g,\Delta}^n = \frac{\vec{S}_\Delta^n}{\Delta t^n}.$$

5.2 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 $\bar{\bar{M}}^n$ relying upon a residual error estimate $\mathcal{E}_r^n = \Delta^2 (R_r^n)^2 + \Delta (J_r^n)^2 + (D_r^n)^2$ where

$$\begin{cases} R_r^n = \|\vec{\nabla} P_\Delta^n - \vec{\nabla} \cdot \bar{\bar{\sigma}}_\Delta^n - \rho_\Delta^n \vec{g}\|_{L^2(e_r^G)}, \\ D_r^n = \|\Phi - \vec{\nabla} \cdot \vec{u}_\Delta^n\|_{L^2(e_r^G)}, \\ J_r^n = \dfrac{1}{2} \sum_{j=1}^4 \|[(\bar{\bar{\sigma}}_\Delta^n - P_\Delta^n \bar{\bar{I}}) \cdot \widehat{n}_{r,j}]\|_{L^2(\mathcal{F}_{r,j}^G)}. \end{cases} \tag{17}$$

and $\widehat{n}_{r,j}$ is the outward normal of the j-th face of the r-th element.
We choose an isotropic metric $\bar{\bar{M}}^n = \bar{\bar{I}}\eta^n$. By doing so, the problem reduces to finding a suitable field η^n. Then, we define an auxiliary variable

$\beta_r^n = (\eta^n)^{3/2}$ that represents the local volumetric deformation induced by the metric $\bar{\bar{M}}^n$. We define β_r^n as the solution of the following minimization problem

$$\min_{\beta_r^n \in \mathbb{R}^{n_e^G}} \frac{1}{2}(\beta_r^n - \widehat{\beta}_r^n)^2 + \frac{\delta}{2}(\beta_r^n - \widetilde{\beta}_r^n)^2, \tag{18}$$

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

$$\widetilde{\beta}_r^n = \widetilde{K}^n \mathcal{R}_r^n, \qquad \widehat{\beta}_r^n = \frac{\sqrt{K_{\mathcal{E}}^n}}{\mathcal{E}_r^n}, \tag{19}$$

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

$$\widetilde{K}^n = \frac{|\Omega|}{\sum_{r=1}^{n_e^G} \mathcal{R}_r^n |e_r^{G,n}|}, \qquad \sqrt{K_{\mathcal{E}}^n} = \frac{|\Omega|}{\sum_{r=1}^{n_e^G} |e_r^{G,n}|/\mathcal{E}_r^n}. \tag{20}$$

As we will see shortly, the solution of (18) 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 $(\beta_r^n - \widehat{\beta}_r^n)^2/2$ in problem (18) triggers the equidistribution of the error. Indeed we would like that the contribution to the error of each element in the grid at time t^n be the same, that is

$$\left(\mathcal{E}_r^n \frac{|e_r^{G,n+1}|}{|e_r^{G,n}|}\right)^2 = (\mathcal{E}_r^n \beta_r^n)^2 = K_{\mathcal{E}}^n \qquad r = 1, \ldots, n_e^G,$$

where we have supposed that, for small grid deformations, the ratio $|\mathcal{E}_r|/|e_r^{G,n-1}|$ is almost independent of the grid geometry.
The term $\delta(\beta_r^n - \widetilde{\beta}_r^n)^2/2$ in problem (18) instead 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.,

$$\beta_r^n = \widetilde{K} \mathcal{R}_r^n \qquad r = 1, \ldots, n_e^G.$$

The solution of (18) is

$$\beta_r = \frac{\delta \widetilde{\beta}_r^n + \widehat{\beta}_r^n}{1 + \delta}. \tag{21}$$

We can show that (21) satisfies (10), in fact

$$\int_{\Omega^n} \sqrt{\det(\bar{\bar{M}}^n)} = \sum_{r=1}^{n_e^G} \beta_r |e_r^{G,n}|. \tag{22}$$

Then using (19) and (20) we get:

$$\int_{\Omega^n} \sqrt{\det(\bar{\bar{M}}^n)} = \frac{1}{1+\delta}\left(\delta\widetilde{K}^n \sum_{r=1}^{n_e^G} \mathcal{R}_r^n |e_r^{G,n}| + \sqrt{K_{\mathcal{E}}^n} \sum_{r=1}^{n_e^G} \frac{|e_r^{G,n}|}{\mathcal{E}_r^n}\right) = \frac{1}{1+\delta}\left(\delta|\Omega| + |\Omega|\right). \quad (23)$$

As regards the weight δ we choose

$$\delta > \max\left(\frac{\widehat{\beta}_{\max}^n - \mathcal{U}}{\mathcal{U} - \widetilde{\beta}_{\max}^n}, \frac{\mathcal{L} - \widehat{\beta}_{\min}^n}{\widetilde{\beta}_{\min}^n - \mathcal{L}}, 0\right), \quad (24)$$

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

Proposition 1. *If* $\mathcal{U} > \widetilde{\beta}_{\max}^n$, $\mathcal{L} < \widetilde{\beta}_{\min}^n$, *then (21) and (24) imply* $\mathcal{L} \leq \beta_r^n \leq \mathcal{U}$.

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

$$\frac{\partial\beta_r^n}{\partial\widehat{\beta}_r^n} = \frac{1}{1+\delta} > 0, \qquad \frac{\partial\beta_r^n}{\partial\widetilde{\beta}_r^n} = \frac{\delta}{1+\delta} > 0.$$

Therefore, from (21) it follows that $\beta_r^n < (\widehat{\beta}_{\max}^n + \delta\widetilde{\beta}_{\max}^n)/(1+\delta)$. Since (24) implies $(\widehat{\beta}_{\max}^n + \delta\widetilde{\beta}_{\max}^n)(1+\delta) < \mathcal{U}$, we get the first part of the bound. With similar arguments the other bound holds. ■

6 Handling of faults

6.1 A finite volume scheme

Here we introduce an innovative scheme for fault tracking. Since we do not aim at representing the fault surface precisely at this stage, we consider a piecewise-constant indicator function, defined on the elements of the subgrid, i.e., the uniformly refined grid. In other terms we represent the fault-damaged region rather than the fault surface. This choice goes towards the increase of the accuracy of the discrete solution. Indeed, three-dimensional grids usually have many more elements than nodes, therefore the use of element-related unknowns guarantees a lot of degrees of freedom in the fault region. Moreover our scheme is particulary computationally efficient since, as we will see in Section 8, the fault function (i.e., the level set-like function

that implicitly defines the faulted region) can be reconstructed only a few times. We now define the fault function as

$$\begin{cases} \lambda^{\mathcal{F}} > 1/2 & \text{in } \Omega^{\mathcal{F}}, \\ \lambda^{\mathcal{F}} \leq 1/2 & \text{in } \Omega \backslash \Omega^{\mathcal{F}}. \end{cases} \tag{25}$$

Then, we define the viscosity abatement function ψ, already introduced in (2), as

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

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

$$\frac{\partial \lambda^{\mathcal{F}}}{\partial t} + (\vec{u} - \vec{u}_{g,\Delta}) \cdot \vec{\nabla} \lambda^{\mathcal{F}} = 0. \tag{26}$$

The discrete counterpart of $\lambda^{\mathcal{F}}(t, \cdot)$ is $\lambda^{\mathcal{F}}_{\Delta}(t, \cdot) \in \mathbb{W}^S_0$ and is piecewise constant on the elements. Its degrees of freedom at time t^n are indicated by $\lambda_r^{\mathcal{F},n}$. We solve (26) with an implicit tracking method (see [50]) i.e.,

$$\lambda_r^{\mathcal{F},n+1} = \left(1 + \sum_{j=1}^{4} \nu^n_{r,j} \right) \lambda_r^{\mathcal{F},n} - \sum_{j=1}^{4} F^n_{r,j},$$

where

$$\nu^n_{r,j} = \frac{\Delta t^{\mathcal{F},n}}{|e^S_r|} \int_{\mathcal{F}^S_{r,j}} (\vec{u}(t^n, \cdot) - \vec{u}_{g,\Delta}(t^n, \cdot)) \cdot \widehat{n}$$

are the interface Courant numbers, and $\Delta t^{\mathcal{F},n}$ is the time step. The time step is chosen such that $\nu^n_{r,j}$ satisfies the following condition

$$\nu^n_{r,j} < \frac{1}{4} \qquad \forall r = 1, \ldots, n^S_e, \quad \forall j = 1, \ldots, 4.$$

Usually, the time stepping required for the stability of the main scheme, Δt^n, is bigger then $\Delta t^{\mathcal{F},n}$. Therefore $\Delta t^{\mathcal{F},n}$ is chosen as a submultiple of Δt^n and a sub-time stepping is performed. Finally, $F^n_{r,j}$ are the numerical fluxes

$$F^n_{r,j} = \begin{cases} \nu^n_{r,j} \lambda^{\mathcal{F},n}_{r,j} & \text{if} \quad \nu^n_{r,j} \geq 0, \\ \nu^n_{r,j} \lambda^{\mathcal{F},n}_{r,\bar{j}} & \text{otherwise,} \end{cases}$$

where

$$\lambda^{\mathcal{F},n}_{r,j} = \lambda^{\mathcal{F},n}_r + \gamma^n_{r,j} \Delta \lambda^{\mathcal{F},n}_{r,j}. \tag{27}$$

The limiter $\gamma_{r,j}^n$ of the LS–VT scheme (defined in (27)) is computed, for $j = 1, \ldots, 4$, in the following way

$$\gamma_{r,j}^n = \begin{cases} \min\left(1, \dfrac{1 + \sum_{j=1}^4 \nu_{r,j}^n - \nu_{r,j}^n}{\nu_{r,j}^n |\mathbb{J}_r| \Delta\lambda_{r,j}^{\mathcal{F},n}} \lambda_r^{\mathcal{F},n}, \dfrac{1 - \lambda_{r,j}^{\mathcal{F},n}}{\Delta\lambda_{r,j}^{\mathcal{F},n}}\right) & \text{if} \quad \Delta\lambda_{r,j}^{\mathcal{F},n} > 0, \\ \min\left(1, \dfrac{1 + \sum_{j=1}^4 \nu_{r,j}^n - \nu_{r,j}^n}{\nu_{r,j}^n |\mathbb{J}_r| \Delta\lambda_{r,j}^{\mathcal{F},n}} (\lambda_r^{\mathcal{F},n} - 1), \dfrac{-\lambda_{r,j}^{\mathcal{F},n}}{\Delta\lambda_{r,j}^{\mathcal{F},n}}\right) & \text{if} \quad \Delta\lambda_{r,j}^{\mathcal{F},n} < 0, \\ 1 & \text{if} \quad \Delta\lambda_{r,j}^{\mathcal{F},n} = 0, \end{cases}$$

where $\Delta\lambda_{r,j}^{\mathcal{F},n} = 1/2(\lambda_{r_j}^{\mathcal{F},n} - \lambda_r^{\mathcal{F},n})$, and $\mathbb{J}_r$ is the set of indexes of the outflow faces of the r-cell, i.e., $\mathbb{J}_r = \{j \in 1, \ldots, 4 : \nu_{r,j}^n > 0\}$. It is worth noting that the LS–VT coupled scheme is positive, i.e., $0 \leq \lambda_\Delta^{\mathcal{F}}(t, \cdot) \leq 1$ for all $t > 0$, see [50]. 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 [38, 45]).

6.2 Set reconstruction (continuous part)

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

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

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's now introduce the method in its continuous form. We define a coefficient $\alpha(t)$ as

$$\alpha(t) = \int_\Omega \left(\lambda^{\mathcal{F}}(t, \cdot) - 1/2\right) H(\lambda^{\mathcal{F}}(t, \cdot) - 1/2). \tag{28}$$

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

$$\begin{cases} J = \min\limits_{\theta \in L^2(\Omega)} \dfrac{1}{2} \displaystyle\int_\Omega (\theta - \lambda^{\mathcal{F}})^2, \\ \displaystyle\int_\Omega \theta = \alpha, \\ \theta \geq 0. \end{cases} \tag{29}$$

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

Proposition 2. *The solution of problem (29) is*

$$\overline{\theta} = \left(\lambda^{\mathcal{F}} - 1/2\right) H(\lambda^{\mathcal{F}} - 1/2). \tag{30}$$

Proof. We show that every perturbation of the solution (30) yields an increase of the functional J. Let's consider a small perturbation of the solution that satisfies the constraints $\theta = \overline{\theta} + \varepsilon\widetilde{\theta}$, where ε is a parameter that tends to zero. $\widetilde{\theta}$ is a perturbation function that satisfies the constraints in (29), therefore

$$\widetilde{\theta}(\vec{X}) \geq \begin{cases} 0 & \text{if} \quad \vec{X} \notin \Omega^{\mathcal{F}} \\ -\dfrac{\overline{\theta}(\vec{X})}{\varepsilon} & \text{if} \quad \vec{X} \in \Omega^{\mathcal{F}} \end{cases}, \qquad \int_{\Omega^{\mathcal{F}}} \widetilde{\theta} + \int_{\Omega/\Omega^{\mathcal{F}}} \widetilde{\theta} = 0. \tag{31}$$

Let's evaluate the functional J in $\overline{\theta} + \varepsilon\widetilde{\theta}$:

$$\begin{aligned} J(\overline{\theta} + \varepsilon\widetilde{\theta}) &= \frac{1}{2}\int_{\Omega} (\overline{\theta} - \lambda^{\mathcal{F}})^2 + \varepsilon \int_{\Omega} \widetilde{\theta}(\overline{\theta} - \lambda^{\mathcal{F}}) + O(\varepsilon^2) \\ &= J(\overline{\theta}) + \varepsilon \int_{\Omega^{\mathcal{F}}} \widetilde{\theta}(\overline{\theta} - \lambda^{\mathcal{F}}) + \varepsilon \int_{\Omega/\Omega^{\mathcal{F}}} \widetilde{\theta}(\overline{\theta} - \lambda^{\mathcal{F}}) + O(\varepsilon^2). \end{aligned}$$

From (30) we get

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

that, combined with the last equation of (31), leads to

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

The first order variation is positive as $\lambda^{\mathcal{F}} < 1/2$ and $\widetilde{\theta} \geq 0$ outside $\Omega^{\mathcal{F}}$, therefore $\overline{\theta}$ is a minimum for J. ■

In our particular case 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 (28). For instance, in the case $\vec{\nabla}\cdot\vec{u} = 0$, then $\alpha(t)$ is constant, and any conservative numerical scheme for (26) will conserve $\int_{\Omega}\lambda^{\mathcal{F}}$ and as a consequence $\alpha(t)$. Our tracking method is conservative (see [50]).

6.3 Set reconstruction (discrete part)

We approximate now θ with $\Theta_\Delta \in \mathbb{W}_0^S$, a piecewise constant function defined on the elements of the sub-grid. The discrete counterpart of problem (29) has the form

$$\min_{\Theta_\Delta \in \mathbb{W}_0^S} \frac{1}{2} \int_\Omega \left(\Theta_\Delta - \lambda_\Delta^{\mathcal{F}}\right)^2 + \eta \left(\int_\Omega \Theta_\Delta - \alpha \right) - \int_\Omega \mathcal{M}\Theta_\Delta - \int_\Omega \mathcal{N} \left(\frac{1}{2} - \Theta_\Delta \right), \tag{32}$$

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 (29), and $\Theta_\Delta \leq 1/2$. The latter condition is not present in the continuous form, but comes from (30), that imposes $\theta \leq 1/2$. Finally α is estimated as $\alpha = 1/2 \int_\Omega \lambda_\Delta^{\mathcal{F}}$. The equivalent optimality conditions for (32) are

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

where $\mathcal{M}_r, \mathcal{N}_r, \Theta_r \in \mathbb{W}_0^S$. This system is solvable by the Uzawa method (see [9] and [25]), as follows

$$\begin{cases} \Theta_r^{m+1} = 2\lambda_r^{\mathcal{F},m} + 2(\mathcal{M}_r^m - \mathcal{N}_r^m) - \eta^m, \\ \eta^m = \dfrac{2}{|\Omega|} \left(\displaystyle\sum_{r=1}^{n_e^S} \lambda_r^{\mathcal{F}} |e_r^S| + \sum_{r=1}^{n_e^S} |e_r^S| (\mathcal{M}_r^m - \mathcal{N}_r^m) - \alpha \right), \\ \mathcal{M}_r^m = \max(0, \mathcal{M}_r^{m-1} - \varrho \Theta_r^m), \\ \mathcal{N}_r^m = \max\left(0, \mathcal{N}_r^{m-1} - \varrho \left(\Theta_r^m - \frac{1}{2}\right)\right), \end{cases}$$

where ϱ is the acceleration factor. At the end of the iterative cycle, the function $\lambda_\Delta^{\mathcal{F}}$ is reconstructed by setting $\lambda_\Delta^{\mathcal{F}} = H(\Theta_\Delta)$.

6.4 The reconstruction algorithm applied to a simple case

In this subsection we report a numerical result regarding a simple one-dimensional 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 (29) has an analytical solution that is:

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

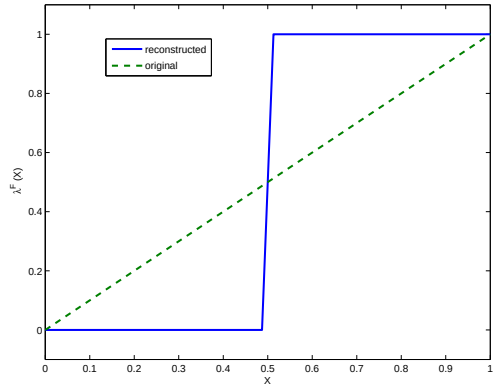


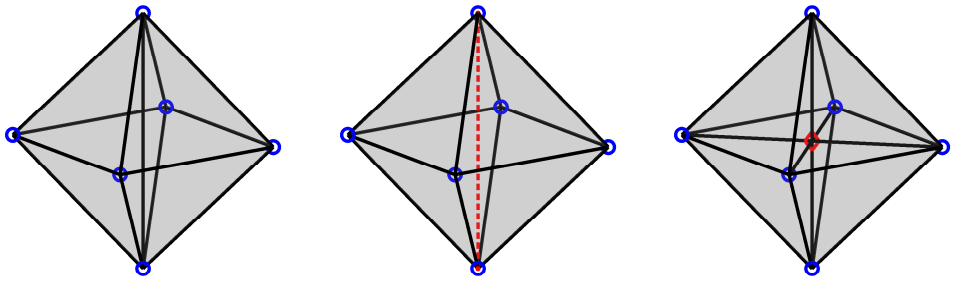
Figure 5: The reinitialization algorithm applied once (dashed line) and twice (solid line) to a highly diffused step function.

In Figure 5 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%.

6.5 Local grid refinement

In this section we recall the local mesh refinement algorithm applied near the fault region. Actually, we have implemented a general algorithm that is capable to refine an arbitrary number of elements, referred as marked elements. In our case, the marked elements correspond to those lying in the fault region but in general, we will be able to adapt the grid wherever it might be necessary, for example across the interfaces. We have considered two local refinement algorithms: the Red–Green and the bisection. The former exploits a uniform type refinement on the elements and manages the hanging nodes with dedicated regularization methods; the latter automatically generates a coherent grid, but only some of its variants guarantee a good mesh quality. A brief review of the latter techniques can be found in [5, 26, 27, 41] and [51]. We have implemented a recursive longest-edge refinement which splits the longest edge of the marked elements and assures a good mesh quality. In fact, to guarantee a coherent mesh, also the tetrahedra adjacent to the marked elements are bisected. The algorithm inputs are the maximum edge length $h_{\max}$, the maximum number of iterations $N_{\max}$ and the list $l_{\mathcal{T}}$ of the indexes of the elements to be refined. Then we proceed as follows:

Algorithm 2. *Mesh refinement:*



(a) (b) (c)

Figure 6: The bisection of four tetrahedrons. a) Four tetrahedrons, six vertices (blue circles); b) The refinement edge (red dashed); c) Eight tetrahedrons, with the new vertex (red diamond).

1. *find the longest edge of* e_r^G*, with* $r \in l_{\mathcal{T}}$*. For simplicity, we label it with the letter* $\mathcal{G}$*;*
2. *if* $\mathcal{G}$ *is shorter than* $h_{\max}$ *terminate, otherwise iterate* $N_{\max}$*-times the following steps:*
   - *find the longest edge belonging to the elements connected to* $\mathcal{G}$*;*
   - *if this edge is* $\mathcal{G}$ *go to the next step, otherwise label the new edge as the new* $\mathcal{G}$*;*
3. *refine the elements connected with* $\mathcal{G}$*.*

This algorithm terminates in a finite number of steps and produces a good mesh quality although, for the 3D case, there are no theoretical statements that back this heuristical result. At the end of the refinement phase, the physical data are transferred from the unrefined to the refined grid and, as new boundary elements may be created, in case some more boundary data are generated. Finally, the mini-grid and the sub-grid are built from the refined grid.

7 Numerical results

We present now some numerical tests to demonstrate the capabilities of our mathematical modeling set up through some realistic examples of simulation. For the time being we only assure that the numerical techniques we have chosen work properly, as a complete physical validation test is beyond the scope of this work. Indeed it is a lengthy process since there are few experimental data and most of them are incomplete and affected by a high degree of uncertainty. However a physical discussion of the numerical result is included.

The first test we have considered is the kinematical evolution of a faulted basin. Here we want to show that our numerical scheme is able to cope with the mesh distortion introduced by a fault. A two-dimensional example of a fault test case can be found in [30]. In order to maximize the distortion

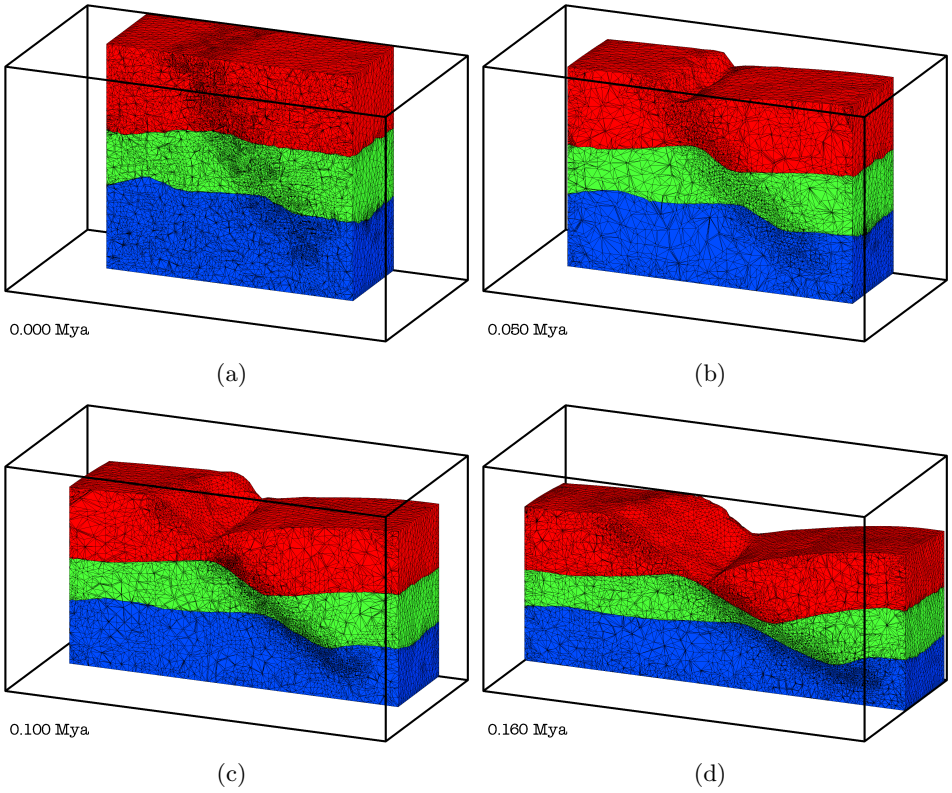


Figure 7: The evolution of a three-layered faulted basin (vertical cut section at $x = 6204$ km).

effects we have have set, for all the three layers in Figure 7, $\mu_i^0 = \mu_i^\infty = 10^{22}$ Pa s. In other terms we have considered a consolidated rock behavior. This is a quite unrealistic assumption, since usually the surface sediments are less compacted and weaker. However the main object of this test is to demonstrate the code robustness even in these extreme conditions. We also set the sediment density to 3000 kg/m^3 and a Newtonian rheological law.
In Figure 7 we can see the sediment evolution: the fault behaves as a sliding plane triggering the subduction of the right part of the basin. In this test case only three surface remeshing steps are needed to prevent mesh tangles and the overall quality is kept high. Our result can be compared with the observed qualitative behavior of faults in an extensional sedimentary basins, see [47]. The qualitative trend of the interaction between the sediment layers and the fault evolution are captured.

In the second test case we look into the effect of the rheology on the evolution of a salt diapir. We have considered a unit cube filled with three sedimentary layers, whose thicknesses are 0.2 for the lower and the upper layers, and 0.6 for the middle one. A small perturbation is applied to the interface between the first and second layer (whose characteristics are described in Table 1) in order to trigger a gravitational instability.

Table 1: Physical characteristics of the three layers for the diapir growth simulation. r and ξ_i values come from literature and have been chosen to stress the plastic effect, and are the same for all the corresponding rheological laws.

Sediment	ρ [10^3kg/m^3]	μ_i [10^{20}Pa s]	r	ξ_i
Bottom	2.14	1	2.0	0
Middle	2.60	100	1.2	70
Top	2.00	10	1.2	70

The rheology has profound influence on the evolution of the sedimentary basin. The most important effect is on the timing of the diapir growth, shown in Table 2.

Table 2: Diapir growth time (in Mya) with different rheological models.

Step	Newton	Carreau	Cross	Powell	Yeleswarapu
15	372	155	122	205	208
20	427	161	126	216	219
25	468	167	130	224	228
30	500	173	135	233	237
35	534	191	146	248	253

Clearly all the pseudoplastic shear-thinning rheologies speed up the diapir formation. Stronger plastic effects mean shorter times for the diapir evolution. If the timing of the sedimentary layer deposition is known with a reasonable precision this data could be interpreted to estimate the amount of plastic effects.
The rheology affects also the shape of the diapir, as shown in Figure 8. At present date this can be described very precisely, thanks to seismic analysis [18], therefore it provides another indicator for the choice of a proper rheological law. In particular in Figure 8 we can notice a strong difference between the Newtonian law and the plastic laws. Moreover the Carreau and the Cross laws trigger the fastest diapir evolution while the Powell and Yeleswarapu laws cause a slower evolution. In fact in the Carreau and in the Cross laws, the two rheological parameters r and ξ_i have been adjusted to describe strong plastic effects. In this simplified test case these two rheologies match the geological evolution timescales of the diapirs (see [48]) while

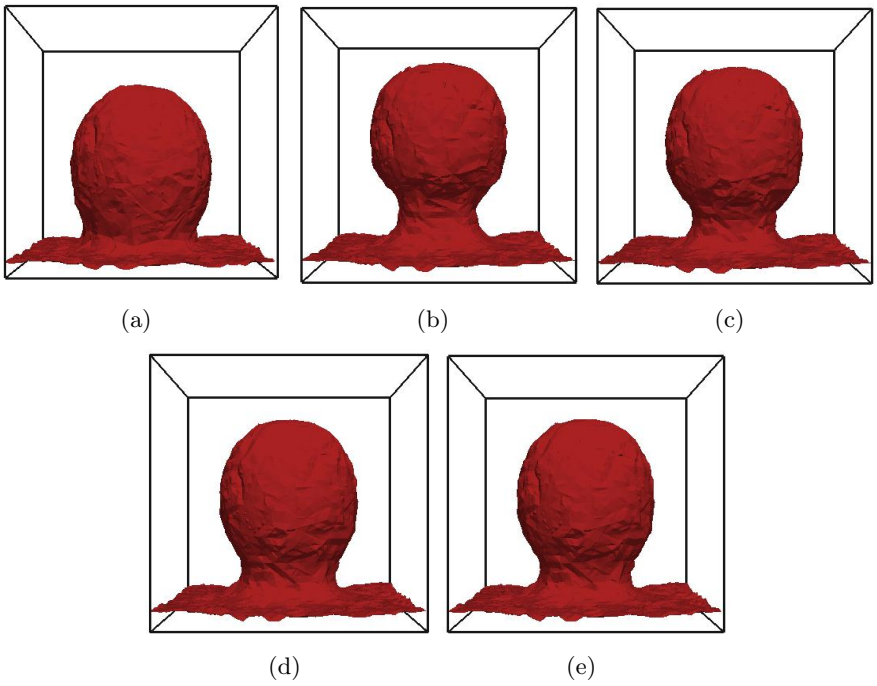


Figure 8: The evolution of a three layer box-shaped basin at time step 25 considering different rheologies: (a) Newton, (b) Carreau, (c) Cross, (d) Powell, (e) Yeleswarapu.

the other seem to prescribe a too slow evolution.
Finally, in Table 3 we can also analyze the stress field. For a possible comparison, some stress measures can be obtained through the mini-frac tests (see [6]).

Table 3: Maximum and minimum values of the second invariant of the stress tensor, expressed in 10^{14} Pa^2s^2.

max	0.0594	0.0237	0.0330	0.0343	0.0347
min	-0.4871	-0.1347	-0.1861	-0.2158	-0.2110

In these examples we have outlined some useful features to calibrate the physical properties of the model and to choose a proper rheological law. The last test case we present consists in a realistic three-layered sedimentary basin with two faults and a salt layer. This test case is important in order to prove the efficiency and the ability of the code to handle realistic and topologically complicated geological configurations. Here we put together all the ingredients presented in this work, in particular the tracking of the faults when they are strongly deformed by a rapidly rising salt diapir. The computations have been run on an AMD Opteron 8212 Dual-Core

2GHz processor; among the several simulations run, the one we present here has about 50K nodes corresponding to about 1 million of degrees of freedom requiring approximately 6.1Gb of RAM memory. The basin extension is $23.9 \times 16.8 \times 7.2$ km and it is subject to an extensional and downward movement; we consider compaction, sedimentation, and we adopted a Carreau rheology model. The characteristics of the layers are described in Table 4.

Table 4: Physical characteristics of the layers of the realistic sedimentary basin, with two faults and a salt layer.

Sediment	ρ [10^3kg/m^3]	μ_i [10^{20}Pa s]	ϕ^0
Salt	2.14	2	0.05
Oligocene	2.6	200	0.7
Eocene	2.6	200	0.55
Early-Miocene	2.6	200	0.7
Mid-Miocene	2.5	200	0.7
Late-Miocene	2.5	200	0.8
Pliocene	2.5	200	0.8

In Figure 9 we have outlined the evolution of the sedimentary basin from 55.8 to 5.3 Mya ago, corresponding to three sedimentation events occurred in the geological ages previously introduced. The white layer is the salt layer, the blue is the Oligocene, the green the Eocene, the yellow the Early-Miocene, the cyan the Mid-Miocene and the pink and orange are the Late-Miocene and Pliocene layers respectively. In the figures, the fault regions are transported by the velocity field and deformed by the rising diapir. As illustrated in Figure 10, the initialization process is able to capture a thin region across fault surfaces. Thanks to the efficiency of the fault tracking algorithm, faults are well followed and reconstructed, and no diffusion is evident, even if there is still room for improvement as regards the initialization process. The ALE scheme for grid movement follows the boundary movement as the lateral walls expand and the basement sinks. Unfortunately, this has proven to be sensitive to the initial quality of the mesh, insomuch as if the initial mesh is bad we have to limit the r-adaption or even to disable it. Anyway, in this case, no remeshes are needed, as the algorithm is able to cope with the mesh movement and the distortion induced by faults. Therefore, the results show how our code can deal with the interaction between the salt dynamics and the fault envisioned.

8 Conclusions

In this work we have introduced an efficient numerical framework for simulating the sedimentary basin evolution, endowed with the modeling of some relevant physical features: the compaction, the movement of the domain

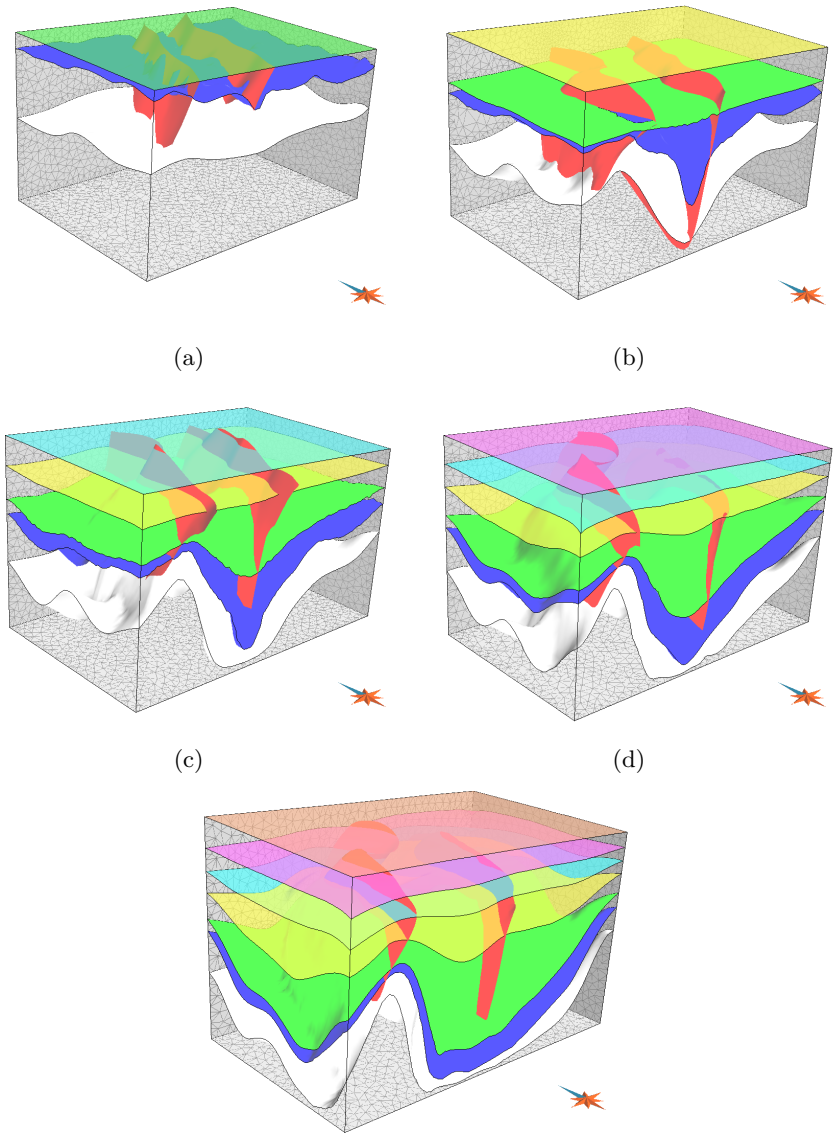
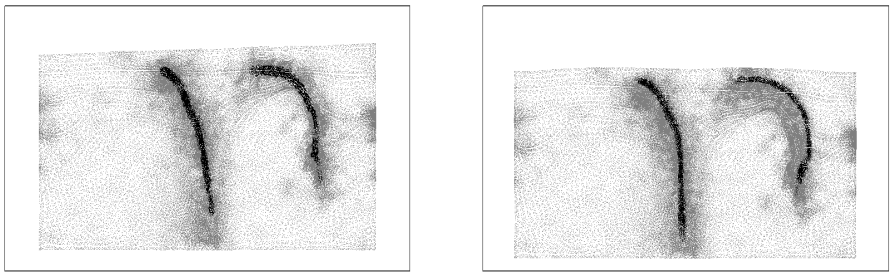


Figure 9: The evolution of a two-faulted (in red) sedimentary basin. (a) 55.8 Mya ago, 3 layers. (b) 23 Mya ago, 4 layers. (c) 16.1 Mya ago, 5 layers. (d) 11.6 Mya ago, 6 layers. (e) 5.3 Mya ago, 7 layers.



(a) (b)

Figure 10: Faults reconstruction. (a) Faults are initially identified with the level set function (black). (b) At the end of the simulation step, the fault regions have not been diffused, and still identified precisely. It can also be noticed the mesh refinement around faults.

boundaries, faults, and non-Newtonian rheologies.
This is coupled with a reliable mathematical support for grid handling and boundary movement. The ALE scheme has proven to be robust and efficient, and only a few remeshing are needed during long-run simulations. To prevent this, anyway, in the future the initial mesh quality should be carefully addressed, by means of dedicated regularization algorithms.
The fault tracking technique has proven to be low-diffusive and conservative in following faults evolution. It can also be applied successfully to all those applications where a precise reconstruction of the domain boundaries is not required, and therefore a discontinuous level set function can be used. Future developments can include a deeper theoretical analysis on this scheme. Last, but not least, the rheological properties of the materials pose the main difficulties to this geological time scale simulations. The confidence in the simulations is highly dependent on the behavior of the materials, and a more involving test phase should be carried out to figure out whether the computed evolution of the basin are coherent with the experimental data collected on the ground. A wider sensitivity analysis of the evolution behavior with respect to the rheological parameters is also envisioned.

References

[1] F. Armero and E. Love. An arbitrary Lagrangian-Eulerian finite element method for finite strain plasticity. International Journal for numerical methods in engineering, pp. 471–508, vol. 57, 2003.

[2] H. Askes and A. Rodriguez-Ferran. A combined rh-adaptivity scheme based on domain subdivision. Formulation and linear examples. International Journal for numerical methods in engineering, pp. 253–273, vol. 51, 2001.

[3] H. Askes, A. Rodriguez-Ferran and A. Huerta. Adaptive analysis of yeld line patterns in plates with the arbitrary Lagrangian–Eulerian method. Computers and Strutures, pp. 257–271, vol. 70, 1999.

[4] D.M. Audet and A.C. Fowler. A Mathematical model for compaction in sedimentary basins. Geophys. J. Int, pp. 577–590, vol. 110, 1992.

[5] J. Bey. Tetrahedral grid refinement. Mathematisches Institut Tubingen, Report nr. 18, 1995.

[6] J.S. Bell. Pratical methods for estimating in situ stresses for borehole stability applications in sedimentary basins. Journal of Petroleum Science and Engineering, pp. 111–119, vol. 38, issue 3-4, 2003.

[7] F. Blom. Consideration on the spring analogy. International Journal for Numerical Methods in Fluids, pp. 647–668, vol. 32, 1998.

[8] C.L. Bottasso, D. Detomi and R. Serra. The Ball-Vertex Method: a New Simple Spring Analogy Method for Unstructured Dynamic Meshes. Computer Methods in Applied Mechanics and Engineering, pp. 4244–4264, vol. 194, 2005.

[9] P.G. Ciarlet. Introduction to numerical linear algebra and optimization. Cambridge University Press, 1988.

[10] S. Cloetingh and Y. Podladchikov. Perspectives in tectonic modeling. Tectonophysics, pp. 169–173, vol. 320, 2000.

[11] I.F. Collins. Associated and Non-Associated aspects of Constitutive Laws for Coupled Elastic - Plastic materials. The international Journal of Geomechanics, pp. 259–267, vol. 2, no. 2, 2002.

[12] J.A.D. Connolly and Y. Podladchikov. Compaction-Driven Fluid Flow in Viscoelastic Rock. Geodinamica Acta, pp. 55–84, vol. 11, issue 2-3, 1998.

[13] R.M. Darlington, T.L. McAbee and G. Rodrigue. A study of ALE simulations of Rayleigh-Taylor instability. Computer Physics Communications, pp. 58–73, vol. 135, 2001.

[14] A. Ern and J.L. Guermond. Theory and Practice of Finite Elements. Springer, 2000.

[15] J.L. Farinatti Aymone. Mesh motion technique for the ALE formulation in 3D large deformation problems. International Journal for numerical methods in engineering, pp. 1879–1908, vol. 59, 2004.

[16] L. Formaggia and F. Nobile. Stability Analysis for the Arbitrary Lagrangian Eulerian Formulation with Finite Elements. East-West Journal of Numerical Mathematics, pp. 105–131, vol. 7, no. 2, 1999.

[17] A.C. Fowler and X. Yang. Fast and slow compaction in sedimentary basins. Siam J. Appl. Math, pp. 365–385, vol. 59, no. 1, 1998.

[18] G. Gardner. 3-D Seismic Imaging. Society of Exploration Geophysicists. website: `http://sepwww.stanford.edu/sep//biondo/3DSI_frame.html`

[19] W. Huang. Anisotropic Mesh Adaptation and Movement. Lecture notes for the workshop on: Adaptive Method, Theory and Application, Peking University, Beijing, China, June 20 - August 20, 2005.

[20] W. Huang, W. Sun. Variational mesh adption 2: error estimates and monitor functions. J. Comp. Phys., 184, 619-648. 2003.

[21] A. Huerta, A. Rodriguez-Ferran, P. Diez and J. Sarrate. Adaptive finite element strategies based on error assessment. International Journal for numerical methods in engineering, pp. 1803–1818, vol. 46, 1999.

[22] A. Ismail-Zadeh, Igor Tsepelev, Chistopher Talbot and Alexander 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, pp. 81–103, vol. 387, 2004.

[23] A.A. Johnson and T.E. Tezduyar. Mesh update strategies in parallel finite element computations of flow problems with moving boundaries and interfaces. Computational Methods in Applied Mechanical Engineering, pp. 73–94, vol. 119, 1994.

[24] M. Lesoinne and C. Farhat. Geometric conservation laws for flow problems with moving boundaries and deformable meshes, and their impact in aeroelastic computations. Computational Methods in Applied Mechanical Engineering, pp. 71–90, vol. 134, 1996.

[25] J.L. Lions. Optimal control of systems governed by partial differential equations. Springer, 1971.

[26] A. Liu and B. Joe. Quality local refinement of tetrahedral meshes based on bisection. SIAM Journal on Scientific Computing, pp. 1269–1291, 1995.

[27] A. Liu and B. Joe. Quality local refinement of tetrahedral meshes based on 8-subtetrahedron subdivision. Mathematics of computation, pp. 1183-1200, vol. 65, 1996.

[28] M. Longoni, A.C.I. Malossi, A. Villa. A robust and efficient conservative technique for simulating three-dimensional sedimentary basins dynamics. MOX report 26, 2009. website: `http://mox.polimi.it/it/progetti/pubblicazioni`

[29] P. Massimi. Perspective in computational salt tectonics. Ph.D. Thesis, Politecnico di Milano, 2006.

[30] P. Massimi, A. Quarteroni, F. Saleri and G. Scrofani. Modeling of salt tectonics. Comp. Meth. Appl. Mech. Eng., pp. 281–293, vol. 197, 2007.

[31] P. Massimi, A. Quarteroni and G. Scrofani. An adaptive finite element method for modeling salt diapirism. M3AS vol. 16, no. 4, 2006.

[32] A. Masud, M. Bhanabhagvanwala and R.A. Khurram. An adaptive mesh rezoning scheme for moving boundary flows and fluid-structure interaction. Computers & fluids, pp. 77–91, vol. 36, 2007.

[33] E. Miglio, A. Villa. A mathematical derivation of compaction and basin models. MOX report 13, 2009. website: `http://mox.polimi.it/it/progetti/pubblicazioni`

[34] L. Moresi, F. Dufour and H.B. Muhlhaus. A Lagrangian integration point finite element method for large deformation modeling of viscoelastic geomaterials. Journal of Computational Physics, pp. 476–497, vol. 184, 2003.

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

[36] E. Onate, S.R. Idelsohn, F. Del Pin and R. Aubry. The particle finite element method. An overview. International Journal of Computational Methods, pp. 267–307, vol. 1, no. 2, 2004.

[37] C. O'Neill, L. Moresi, D. Muller, R. Albert and F. Dufour. Ellipsis 3D. A particle-in-cell finite-element hybrid code for modelling mantle convection and lithosperic deformation. Computers and Geosciences, pp. 1769–1779, 2006.

[38] S. Osher and R.P. Fedkiw. Level Set Methods and Dynamic Implicit Surfaces. Springer, 2003.

[39] J.S. Peery and D.E. Carrol. Multi-Material ALE methods in unstructured grids. Computational Methods in Applied Mechanical Engineering, pp. 591–619, vol. 187, 2000.

[40] D. Peric and A.J.L. Crook. Computational strategies for predictive geology with reference to salt tectonics. Comp. Meth. Appl. Mech. Eng., pp. 5195–5222, vol. 193, 2004.

[41] A. Plaza and G.F. Carey. Local refinement of simplicial grids based on the skeleton. Applied Numerical Mathematics, pp. 195–218, vol. 32, 2000.

[42] A. Quarteroni and A. Valli. Numerical Approximation of Partial Differential Equations Second Edition. Springer–Verlag, 1997.

[43] G. Scrofani. Numerical Basins Modeling and Tectonics. Ph.D. Thesis, Politecnico di Milano, 2007. website: `http://mox.polimi.it/it/progetti/pubblicazioni/tesi/scrofani.pdf`

[44] A. Sequeira, J. Janela. An Overview of Some Mathematical Models of Blood Rheology. A Portrait of State-of-the-Art Research at the Technical University of Lisbon, 65-87. 2007.

[45] A. Sethian. Level Set Methods and Fast Marching Methods. Cambridge University Press, 1999.

[46] R.W. Smith. AUSM(Ale): A geometrical Conservative Arbitraary Lagrangian-Eulerian Flux Splitting Scheme. Journal Computational Physics, pp. 268–286, vol. 150, 1999.

[47] D.L. Turcotte and G. Schubert. Geodynamics, Cambridge University Press, 2001.

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

[49] R. Verfurth. A review of a posteriori error estimation and adaptive mesh refinement techniques. Wiley-Teubner, 1996.

[50] A. Villa. Implicit tracking for multi-fluid simulations. MOX report 11, 2009. website: `http://mox.polimi.it/it/progetti/pubblicazioni`

[51] W. Wessner. Mesh Refinement Techniques for TCAD Tools. Ph.D. Thesis, sec. 2.2, Technische Universität Wien, 2006. website: `http://www.iue.tuwien.ac.at/phd/wessner/node12.html`

[52] S. Zaleski and P. Julien. Numerical simulation of Rayleigh–Taylor instability for single and multiple salt diapirs. Tecnophysics, pp. 55–69, vol. 206, 1991.

MOX Technical Reports, last issues

Dipartimento di Matematica "F. Brioschi",
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

05/2010 Matteo Longoni, A.C.I. Malossi, Alfio Quarteroni, Andrea Villa:
A complete model for non-Newtonian sedimentary basins in presence of faults and compaction phenomena

04/2010 Marco Discacciati, Paola Gervasio, Alfio Quarteroni:
Heterogeneous mathematical models in fluid dynamics and associated solution algorithms

03/2010 P.E. Farrell, Stefano Micheletti, Simona Perotto:
A recovery-based error estimator for anisotropic mesh adaptation in CFD

02/2010 Pietro Barbieri, Niccolo' Grieco, Francesca Ieva, Anna Maria Paganoni and Piercesare Secchi:
Exploitation, integration and statistical analysis of Public Health Database and STEMI archive in Lombardia Region

01/2010 G.M. Porta, S. Perotto, F. Ballio:
Anisotropic Mesh Adaptation Driven by a Recovery-Based Error Estimator for Shallow Water Flow Modeling

35/2009 C. D'Angelo, P. Zunino:
Robust numerical approximation of coupled Stokes and Darcy flows applied to vascular hemodynamics and biochemical transport

34/2009 P. F. Antonietti, P. Houston:
A Class of Domain Decomposition Preconditioners for hp-Discontinuous Galerkin Finite Element Methods

33/2009 S. Vantini:
On the Definition of Phase and Amplitude Variability in Functional Data Analysis

32/2009 S. Micheletti, S. Perotto:
Anisotropic adaptation via a Zienkiewicz-Zhu error estimator for 2D elliptic problems

31/2009 Paola F. Antonietti, Paolo Biscari, Alaleh Tavakoli, Marco Verani, Maurizio Vianello:
Theoretical study and numerical simulation of textiles