

A robust and efficient conservative technique for simulating three-dimensional sedimentary basins dynamics

Matteo Longoni^a, A. Cristiano I. Malossi^{a,1}, Andrea Villa^{b,*}

^a MOX, Modeling and Scientific Computing, Department of Mathematics, Politecnico di Milano, Via Bonardi 9, 20133 Milano, Italy

^b Department of Mathematics, Università degli Studi di Milano, Via Saldini 50, 20133 Milano, Italy

ARTICLE INFO

Article history:

Received 23 July 2009

Received in revised form 25 June 2010

Accepted 30 June 2010

Available online 9 August 2010

Keywords:

Diapir and diapirism

Computational geology

Level set method

Volume tracking method

Finite-element method

ABSTRACT

The development of new efficient numerical techniques is a key point in computational fluid dynamics, and as a consequence in geological simulations. In this paper we present a model for simulating the dynamic of three-dimensional stratified sedimentary basins. This kind of problem contains several numerical complexities such as the presence of high viscosity jumps, or the necessity of tracking multiple surfaces of interface (horizons) independently. To overcome these difficulties, we introduce a new preconditioner, that reduces significantly the amount of time required to solve the finite element linear system resulting from the Stokes problem, and a new tracking method. Using a coupled level set–volume tracking method, indeed, an unlimited number of layers can be tracked with good mass conservation properties. Finally, to prove the efficiency of these new techniques we present the results and the computation performances obtained in simulations of a realistic case with four horizons, together with a complete description of the main physical quantities involved.

© 2010 Elsevier Ltd. All rights reserved.

1. Introduction

Recent geological studies have shown a strong correlation between the presence of salt domes and the formation of oilfields, and as consequence, during the last few years, the interest towards inner dynamics of salt sedimentary basins has had a marked increase. Indeed, salt is a critical element for the dynamics of sedimentary basins, thanks to its high viscosity and its low permeability.

The development of mathematical models and numerical approximations has lead to impressive results in this research field, as a powerful tool to compare and/or explain seismic data. Nevertheless, the necessity of a three-dimensional (3D) approach is inevitable, as the geological structures assume complex morphologies which cannot be described completely through two-dimensional models. For this reason it is essential to employ robust and, above all, efficient methods to solve with accuracy and in a reasonable time, models composed by a considerable number of degrees of freedom, such as those associated with 3D discretizations.

In this article we describe the main features of a mathematical model for simulating the evolution of salt sedimentary basins, and its efficient implementation. The most innovative aspects on which we will focus on are the computational efficiency of the proposed method and the ability to describe accurately the interfaces between layers (usually referred as horizons). We postpone to future papers the handling of more complex issues such as faults, non-Newtonian rheologies and sedimentation processes.

The reference model is composed of a series of layers, modeled as incompressible and immiscible fluids in Stokes regime. This choice, justified by the phenomena of interest lifetime, that is comparable to geological ages (as shown for example by [2,28]), allows geologists to study the Rayleigh–Taylor instabilities associated with salt diapirism (as described by [10,14,16,25,30]). To solve this kind of problem we choose a classic approach that splits the computation of pressure and velocity field from the interface tracking. Moreover, to improve the efficiency and accuracy of the code, we have introduced some innovations both in the resolution of the linear systems and in the tracking algorithm.

During the past few years, the finite element (FE) method has played a predominant role in the solution of the mass and force balance equations (as shown by [14–16,18,21,25]), as it permits to solve the fluid dynamic problem in complex geometrical cases with high accuracy. For 3D problems, the number of degrees of freedom (DOF) required for an accurate discretization is so high that the adoption of iterative schemes to solve the linear system is necessary. In the present case, in addition, the value of the

* Corresponding author. Present address: ERSE, ENEA – Ricerca sul Sistema Elettrico, Via Rubattino 54, 20134, Milano, Italy.

E-mail addresses: matteo.longoni@mate.polimi.it (M. Longoni), cristiano.malossi@mate.polimi.it (A. Cristiano I. Malossi), andrea.villa@unimi.it, avilla@erse-web.it (A. Villa).

¹ Present address: 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.

viscosity of the sediments, which are modeled as fluids, varies over a range of about five orders of magnitude (as shown by [16,25]), and this leads to a bad conditioned algebraic problem. Such a complex problem is in general unsolvable as it is, or at least it requires too many iterations to converge within a reasonable tolerance. To overcome these critical points, in this paper we introduce an innovative efficient preconditioner, that is able to visibly reduce the number of iterations required to solve the linear system in presence of high viscosity jumps. A review of other preconditioning techniques applied to some two-dimensional cases can be found in [17].

As regards the tracking phase, no technique seems to be prevailing on the others. One of the first method employed to solve this kind of problems is the particle in a cell (PIC) (as shown by [8,18]), that has been applied successfully by Fullsack [7] and then extended by O'Neill et al. [19]. This method permits to follow the evolution of a large number of layers, by tracking a high number of particles at the same time, and to reconstruct the physical quantities a posteriori, on the mesh used to compute the velocity field. However, despite its strength, this technique is very expensive as the reconstruction of the physical quantities needs to find how many particles lie inside each cell. Furthermore, the PIC method cannot represent horizons, that instead provide a direct visualization of the geometry of the sediment layers.

An alternative to the PIC is represented by the volume of fluid methods [30]. This algorithm transports the partial volumes but does not contain any direct information regarding interface positions. These are extracted with post processing algorithms that are, usually, only applicable to the two fluid case. A similar way of proceeding, based on characteristic methods, was used by Ismail-Zadeh et al. [9–11] to simulate the growth of diapirs in two and three dimensions. With respect to PIC, this technique provides some small advantages in terms of computational cost, though neither can handle the reconstruction of the horizons. A third possibility is the level set (LS) method, that represents an interface as the zero level of an auxiliary function (see [20,26]). The advantage is that this method easily follows a shape that changes topology, for example when it splits into two parts or develops holes, but on the other hand it does not guarantee the conservation of the volumes. There are few examples of multi-fluid LS approaches, some of them are specifically devoted to the mean curvature flows ([23,27]) and cannot be applied readily to an advection-driven case. In [31] a nested LS approach is proposed: this approach relies on the standard signed distance function and is not conservative.

A different possibility, applied almost exclusively in two dimensions, is the Lagrangian tracking (as reported by [15,16,21]). This technique allows an explicit reconstruction of horizons with a good computational accuracy, paying a relative low computational cost. However, an automatic handling of topological changes is still quite complex, even in two dimensions, therefore this kind of methods have never been applied to realistic 3D geological problems. In our case, we have chosen an Eulerian approach, as it satisfies our objective to have a method that is robust, accurate and, above all, able to represent horizons evolution. In particular, we implement a coupled level set–volume tracking (LS–VT) method (see [29]), which is applied for the first time to a realistic 3D geological case, in presence of more than two layers. This new technique is nearly conservative and is able to give a precise and coherent reconstruction of all the horizons.

This paper is organized as follows: in Section 2 we introduce the mathematical model and the classic splitting approach, which consists in separating the Stokes problem from the hyperbolic equation of advection of horizons. Subsequently, in Section 3 we study in detail the algebraic structure of the Stokes problem and we show a way to build an appropriate preconditioner optimized for high-order viscosity jumps. Then, in Section 4 we present a slight modification of the tracking method proposed by Villa and

Formaggia [29], and we give a brief description of the approach used to reconstruct the Set function. In Section 5, we provide an overview of the algorithm workflow, and finally in Section 6 we illustrate and discuss the results of the simulation of a realistic 3D geological model.

2. Physical and mathematical model

In this section we present the physical model and we introduce some mathematical notation. A time splitting algorithm is introduced along with the time discretization.

2.1. Nomenclature

Let us introduce the geometric model of the sedimentary basin (see Fig. 1). The domain $\Omega \in \mathbb{R}^3$ is divided into n_s subdomains Ω_i (without overlapping regions), which represent different layers characterized by a specific value of density ρ_i and dynamic viscosity μ_i . The external boundary Γ of the domain Ω is divided into three parts: the basement Γ_B and the free surface Γ_S , where we have imposed a Dirichlet condition for the velocity field, and the lateral contour Γ_L , that we suppose vertical for simplicity and where we have imposed a vertical no-stress condition. In addition, the horizons between the subdomains are defined as $\Gamma_{ij} = \Omega_i \cap \Omega_j$.

To complete our overview let us introduce now some mathematical objects: $X = (x_1, x_2, x_3) \in \Omega$ indicates a generic point in the spatial domain of coordinates x_i , with $i = 1, 2, 3$, $(\hat{x}_1, \hat{x}_2, \hat{x}_3)$ the axial versors, \hat{n} the domain outward normal and $t \in (0, T]$ the time coordinate. Then for a generic vector \vec{u} we denote its cartesian orthogonal components with (u_1, u_2, u_3) .

Finally, we introduce the mathematical model that describes the geological evolution of the basin, modeled as a stratified fluid, in which the layers are immiscible and have constant properties:

$$\begin{cases} \vec{\nabla} \cdot \vec{\sigma}(\mu) - \vec{\nabla} P + \rho \vec{g} = 0 & \text{in } \Omega \times (0, T], \\ \vec{\nabla} \cdot \vec{u} = 0 & \text{in } \Omega \times (0, T], \\ \frac{\partial \rho}{\partial t} + \vec{u} \cdot \vec{\nabla} \rho = 0, \frac{\partial \mu}{\partial t} + \vec{u} \cdot \vec{\nabla} \mu = 0 & \text{in } \Omega \times (0, T], \\ \rho = \rho^0, \mu = \mu^0 & \text{in } \Omega \times \{0\}, \\ \vec{u} = \vec{u} & \text{on } \Gamma_B \cup \Gamma_S \times [0, T], \\ u_1 = \bar{u}_1, u_2 = \bar{u}_2, (\vec{\sigma} \cdot \hat{n}) \cdot \hat{x}_3 = 0 & \text{on } \Gamma_L \times [0, T]. \end{cases} \quad (1)$$

The unknowns are the velocity and pressure fields (respectively \vec{u} and P), together with the physical quantities (the density ρ and dynamic viscosity μ). \vec{g} is the gravitational acceleration vector. Problem (1) describes the motion of a creeping stratified fluid when the inertial terms can be neglected. Its first equation states the momentum balance, the second one the incompressibility condition, and the third one the transport of the physical coefficients. The transport equation for the density, under the incompressibility assumption, is equal to the mass conservation law. The remaining relations in (1) consist of the initial and the boundary conditions. Initial conditions are provided for the physical quantities through ρ^0 and μ^0 . As regards the boundary conditions we impose the velocity vector on the surface and on the bottom of the domain and, on the lateral boundary, a free slip condition in the vertical direction plus an horizontal plane velocity field are imposed. In particular, \vec{u} is the imposed velocity on Γ_B and Γ_S , while on Γ_L the velocity components orthogonal to \hat{x}_3 are fixed through \bar{u}_1 and \bar{u}_2 .

We also assume for now a Newtonian law for the stress tensor: $\vec{\sigma} = \mu(\vec{\nabla} \vec{u} + (\vec{\nabla} \vec{u})^T)$. This relation may not seem to be truly representative of the rheological complexity of the sediments, however it is widely accepted in literature as a solid base model to study the

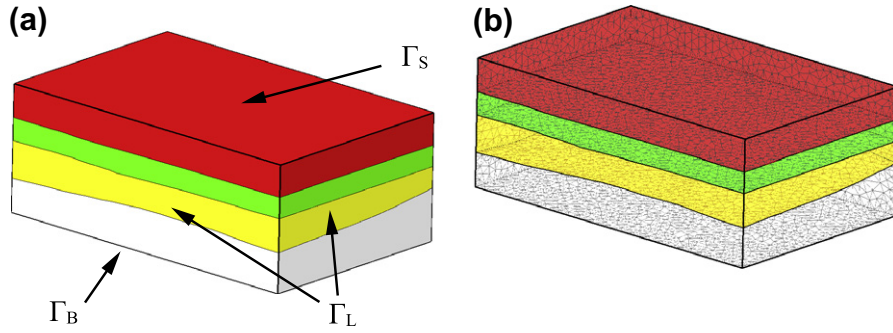


Fig. 1. (a) External shape of the domain Ω , which contains three horizons and four layers. The external boundary Γ is divided into three parts: the basement Γ_B , the free surface Γ_S and the lateral contour Γ_L . (b) An open 3D view of the sedimentary basin.

Rayleigh–Taylor instabilities associated with diapirism (see [9,10,16]).

2.2. Time discretization and splitting algorithm

To solve (1) we must discretize the time interval $[0, T]$; we introduce hence a variable time step, such that $\Delta t^n = t^{n+1} - t^n$, and we discretize the interval as $[0, t^1, t^2, \dots, T]$. Therefore, we can solve the time discrete problem through the following splitting algorithm (already adopted, for example, by [16,24]). The apex n indicates the value of a quantity at time t^n :

Algorithm 1. At the generic time step n we solve problem (1) in two steps:

- (i) given ρ^n and μ^n , we compute \bar{u}^n and P^n by solving the Stokes problem

$$\begin{cases} \bar{\nabla} \cdot \bar{\sigma}^n - \bar{\nabla} P^n + \rho^n \bar{g} = 0 & \text{in } \Omega, \\ \bar{\nabla} \cdot \bar{u}^n = 0 & \text{in } \Omega, \\ \bar{u}^n = \bar{u} & \text{on } \Gamma_B \cup \Gamma_S, \\ u_1^n = \bar{u}_1, u_2^n = \bar{u}_2, (\bar{\sigma}^n \cdot \bar{n}) \cdot \hat{x}_3 = 0 & \text{on } \Gamma_L; \end{cases} \quad (2)$$

- (ii) given \bar{u}^n and P^n , we obtain ρ^{n+1} and μ^{n+1} from the following hyperbolic equations

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \bar{u}^n \cdot \bar{\nabla} \rho = 0, \quad \frac{\partial \mu}{\partial t} + \bar{u}^n \cdot \bar{\nabla} \mu \\ = 0, \quad \text{in } \Omega \times (t^n, t^{n+1}]. \end{aligned} \quad (3)$$

The first step of the algorithm is a classic Stokes problem that can be solved efficiently with a FE method. The second step requires the solution of a linear hyperbolic problem.

Our splitting algorithm is first order accurate and is an efficient and computationally cheap approach, as we will demonstrate in the following sections. It is well suited to solve dynamic stratified flows such as those related to geophysical basin scale modeling and salt diapirism.

2.3. Characteristic function for physical quantities

We want now to describe efficiently the internal 3D structure of a domain made of immiscible layers characterized by different physical properties. At this purpose we introduce a set of characteristic functions λ_i (one for each subdomain Ω_i)

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

that we can gather in the composition vector λ . Through these functions we can write the density and the dynamic viscosity fields as

$$\rho = \sum_{i=1}^{n_s} \lambda_i \rho_i, \quad \mu = \sum_{i=1}^{n_s} \lambda_i \mu_i. \quad (4)$$

Denoting the value of λ_i at time t^0 with λ_i^0 , we can rewrite in the same way the initial condition for the physical quantities, as

$$\rho^0 = \sum_{i=1}^{n_s} \lambda_i^0 \rho_i, \quad \mu^0 = \sum_{i=1}^{n_s} \lambda_i^0 \mu_i.$$

We need now to rephrase the hyperbolic transport equation in a volume transport equation. For $t \in [t^n, t^{n+1}]$ we have

$$\begin{cases} \frac{\partial}{\partial t} \lambda + \bar{u}^n \cdot \bar{\nabla} \lambda = 0, \\ \lambda(0, \vec{X}) = \lambda^0(\vec{X}), \end{cases} \quad (5)$$

where λ^0 is the composition vector at time t^0 . It is easy to show that Eq. (5) is equivalent to the mass conservation law (3): multiplying (5) by ρ_i and performing a sum on the geological species we obtain

$$\sum_{i=1}^{n_s} \left(\rho_i \frac{\partial \lambda_i}{\partial t} + \rho_i \bar{u}^n \cdot \bar{\nabla} \lambda_i \right) = 0.$$

Hence, as the densities of the components ρ_i are constant, we get the first equation of Eq. (3). With a similar argument we can also prove the second of (3). Once we have provided the basic description of the splitting algorithm, in the next two sections we will analyze one by one both the steps of this scheme.

3. Velocity field solver

In this section we discuss the difficulties related to the solution of the Stokes problem. The FE formulation is equivalent to an algebraic system of equations, that, especially in 3D, has a very large number of unknowns. Besides, the strong variability of the viscosity coefficient badly affects the conditioning of the FE matrix. This imposes the usage of a preconditioned iterative method, which is the best solution when dealing with 3D problems. This technique has not been studied in detail yet in our particular case. In the following we define the discrete problem and we construct a suitable preconditioning technique.

Let \mathcal{T}_A^G be a simplicial tetrahedral grid containing n_e^G elements e_r^G (with $r = 1, \dots, n_e^G$) and n_p^G nodes \bar{x}_k^G (with $k = 1, \dots, n_p^G$), where the subscript A stands for the maximum diameter of the grid elements. From \mathcal{T}_A^G we build the mini-grid \mathcal{T}_A^M by adding n_e^G barycentric nodes (see Fig. 2); hence \mathcal{T}_A^M has $n_p^M = n_p^G + n_e^G$ nodes \bar{x}_k^M and $n_e^M = 4n_e^G$ elements e_r^M . Let us finally recall the discrete variables and introduce the related discrete spaces:

$$\begin{aligned} \bar{u}_A^n \in \bar{\mathbb{V}}_1^M : \bar{\mathbb{V}}_1^M = \{ \bar{\varphi}_A \in \mathbb{C}^0(\Omega) : \bar{\varphi}_A|_{e_r^M} \in \mathbb{P}^1, \forall r = 1, \dots, n_e^M \}, \\ P_A^n \in \mathbb{V}_1^G : \mathbb{V}_1^G = \{ \varphi_A \in \mathbb{C}^0(\Omega) : \varphi_A|_{e_r^G} \in \mathbb{P}^1, \forall r = 1, \dots, n_e^G \}, \end{aligned}$$

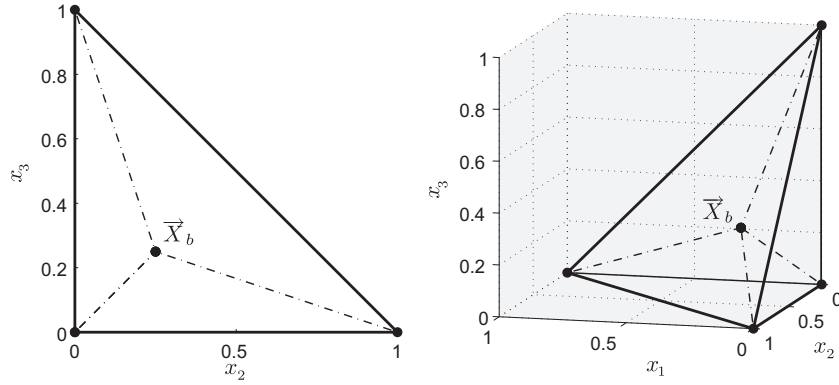


Fig. 2. A sketch of \mathcal{T}_A^M grid obtained by the refinement of the \mathcal{T}_A^C grid. An element e_r^C is depicted with a solid line while the four elements e_r^M are depicted with a dot-dashed line. The latter are obtained by adding a barycentric node inside the grid element, connected to the vertices of e_r^C .

where \mathbb{V}_1^M and \mathbb{V}_1^C are intended as subspaces of the couple H_0^1 and L_0^2 (i.e., the square integrable functions with zero mean value), in which the continuous solution of the Stokes problem is searched. Then we introduce the weak formulation of problem (2)

$$\begin{cases} a(\bar{u}_A^n, \bar{v}_A) + b(P_A^n, \bar{v}_A) = F(\bar{v}_A) & \forall \bar{v}_A \in \mathbb{V}_1^M, \\ b(q_A, \bar{u}_A^n) = 0 & \forall q_A \in \mathbb{V}_1^C, \end{cases} \quad (6)$$

where

$$\begin{aligned} a(\bar{u}_A^n, \bar{v}_A) &= - \int \mu (\bar{\nabla} \bar{u}_A^n + (\bar{\nabla} \bar{u}_A^n)^T) : (\bar{\nabla} \bar{v}_A), \\ b(P_A^n, \bar{v}_A) &= \int \bar{P}_A^n (\bar{\nabla} \cdot \bar{v}_A), \\ F(\bar{v}_A) &= - \int \rho_A (\bar{g} \cdot \bar{v}_A). \end{aligned}$$

Eq. (6) represents the discretization of the Stokes problem with the mini-elements. Let $\mu > 0$ and $\mu \in L^\infty$, therefore the bilinear form $a(\cdot, \cdot)$ is coercive and from the choice of the discrete spaces the following inf-sup condition holds:

$$\inf_{q \in L^2(\Omega), q \neq 0} \sup_{\bar{v} \in H^1(\Omega), \bar{v} \neq 0} \frac{b(q, \bar{v})}{|\bar{v}|_{H^1(\Omega)} \|q\|_{L^2(\Omega)}} \geq \beta.$$

Furthermore, the FE couple we have chosen satisfies the discrete inf-sup condition (see [5])

$$\min_{q_A \in \mathbb{V}_1^C, q_A \neq 0} \max_{\bar{v}_A \in \mathbb{V}_1^M, \bar{v}_A \neq 0} \frac{b(q_A, \bar{v}_A)}{|\bar{v}_A|_{H^1(\Omega)} \|q_A\|_{L^2(\Omega)}} \geq \beta_A. \quad (7)$$

At last, assuming $\rho \in L^\infty$, the functional $F(\cdot)$ is continuous. Thus, problem (6) has a unique solution (see [22]).

Remark 1. In the bilinear form $a(\cdot, \cdot)$ the velocity gradient is piecewise constant on the elements and we can slip it outside the integral sign. Therefore we have only to provide the viscosity mean value on every element.

Let $\{\varphi_k^M\} \in \mathbb{V}_1^M$ and $\{\varphi_k^C\} \in \mathbb{V}_1^C$ be the lagrangian basis defined on the grids \mathcal{T}_A^M and \mathcal{T}_A^C respectively, then the discrete solution can be expanded as

$$\bar{u}_A^n = \sum_{k=1}^{n_p^M} \bar{u}_k^n \varphi_k^M, \quad P_A^n = \sum_{k=1}^{n_p^C} P_k^n \varphi_k^C. \quad (8)$$

Through (8) and defining \mathbf{V} and \mathbf{P} as the vectors of the degrees of freedom of the velocity and pressure fields, we get the following algebraic problem:

$$\begin{bmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{V} \\ \mathbf{P} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_v \\ \mathbf{0} \end{bmatrix},$$

where

$$\mathbf{A}_{ij} = a(\varphi_i^M, \varphi_j^M), \quad \mathbf{B}_{ij} = b(\varphi_i^C, \varphi_j^M), \quad \mathbf{F}_{v,i} = F(\varphi_i^M),$$

and \mathbf{A} is a positive definite matrix. We solve this algebraic system by means of the pressure Schur complement

$$(\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T)\mathbf{P} = \mathbf{B}\mathbf{A}^{-1}\mathbf{F}_v.$$

This algebraic complement is solved through a nested cycle iterative system. The inner cycle is necessary to avoid an explicit inversion of matrix \mathbf{A} . let us now attend to the conditioning of \mathbf{A} and of the Schur complement $\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T$ that form the two linear systems of our iterative scheme. Before, we briefly recall some useful results applied to our case. Let $\bar{v}_A \in \mathbb{V}_1^M$ and $q_A \in \mathbb{V}_1^C$ then we consider the Korn inequality (see [4])

$$\|\bar{\nabla} \bar{u}_A\|_{L^2(\Omega)} \leq C_1 \|\bar{\nabla} \bar{u}_A + (\bar{\nabla} \bar{u}_A)^T\|_{L^2(\Omega)}, \quad (9)$$

and we also have

$$\|\bar{\nabla} \bar{u}_A + (\bar{\nabla} \bar{u}_A)^T\|_{L^2(\Omega)} \leq C_2 \|\bar{\nabla} \bar{u}_A\|_{L^2(\Omega)}. \quad (10)$$

From [5] we obtain

$$\|\bar{\nabla} \cdot \bar{u}_A\|_{L^2(\Omega)} \leq C_3 \|\bar{\nabla} \bar{u}_A\|_{L^2(\Omega)}, \quad (11)$$

and the inverse inequality (see [6]) leads to

$$\|\bar{\nabla} \bar{u}_A\|_{L^2(\Omega)} \leq \frac{C_4}{\Delta} \|\bar{u}_A\|_{L^2(\Omega)}. \quad (12)$$

We remark that Δ stands for the maximum diameter of the grid elements. Finally we recall the Poincaré inequality (see [6])

$$\|\bar{u}_A\|_{L^2(\Omega)} \leq C_p \|\bar{\nabla} \bar{u}_A\|_{L^2(\Omega)}. \quad (13)$$

An estimation of the conditioning number of the Schur complement and of matrix \mathbf{A} is provided by Propositions 1 and 2, respectively.

Proposition 1. The conditioning number of the Schur complement $\mathcal{K}(\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T)$ is not dependent upon Δ , more explicitly:

$$\mathcal{K}(\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T) \leq \frac{\sup(\mu)}{\inf(\mu)} \frac{C_6}{C_5} \left(\frac{C_1 C_2^2 C_3}{\beta_A^2} \right)^2.$$

Proof.

See Appendix A. \square

Proposition 2. The conditioning number of \mathbf{A} is bounded by

$$\mathcal{K}(\mathbf{A}) \leq \frac{\max(\mu)}{\inf(\mu)} \frac{1}{\Delta^2} \frac{C_8}{C_7} (C_1 C_2 C_5 C_p)^2.$$

Proof. See Appendix B. \square

These results show that the conditioning of the two linear systems we wish to solve is bad. In particular, in the inner cycle, where \mathbf{A} is involved, the matrix conditioning is affected by both the grid spacing and the viscosity jumps across the interfaces. However in this case several standard techniques such as the incomplete LU factorizations can be profitably applied. The outer cycle, involving the Schur complement, has instead a relatively better conditioning number but is still dependent upon the jumps of the viscosity coefficient. Moreover we cannot apply standard preconditioning techniques, since they would require an explicit assembling of the Schur complement, that is memory and time consuming. Therefore it is mandatory to find a spectral equivalent to the Schur complement, and at this scope we introduce the following matrix:

$$\mathbf{M}_{ij} = \int \frac{1}{\mu} \varphi_i^G \varphi_j^G. \quad (14)$$

We now give a rough explanation about the reasons that make this precondition matrix suitable to our purpose. We give only a heuristical explanation of the properties of this matrix, as a complete analysis on the spectral equivalence of the Schur complement with the matrix (14) is still missing. The Schur complement is the product between the matrix \mathbf{B} , that represents the divergence operator, the matrix \mathbf{A}^{-1} , that is the discrete representation of the inverse of the Laplace operator, and the matrix \mathbf{B}^T , that represents the gradient. As the application of the gradient to the divergence yields to the Laplace operator, the Schur complement resembles the identity operator. Indeed it is known (see [5]) that the Schur complement is not dependent on Δ . In our case the matrix \mathbf{A}^{-1} resembles more closely $(\nabla^2)^{-1}/\mu$, therefore we expect that the Schur complement is spectrally similar to the mass matrix, apart from the $1/\mu$ factor. The numerical results in Fig. 3 show that the preconditioner is effective and has a better behavior when the grid spacing is reduced. In Fig. 4 the spectra of the Schur complement and of the preconditioned Schur complement are compared. It can be noticed that the preconditioning matrix positively affects the ratio between the maximum and minimum eigenvalue, although there is still a dependence on the viscosity jump $g = \max(\mu)/\min(\mu)$. The numerical tests may suggest an asymptotic trend, but a theoretical result regarding the preconditioner performances is still missing and researches in this field are ongoing.

4. Tracking of the layers

As we stated previously, for the tracking phase we adopt the method developed in [24,29]. This consists in a coupled level

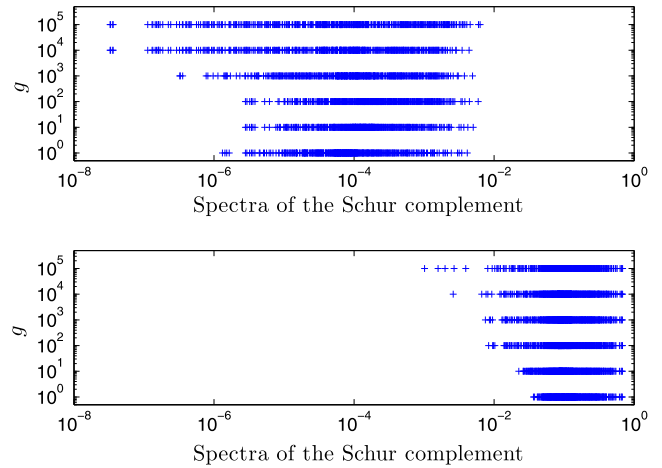


Fig. 4. The spectra of the Schur complement (upper figure) for the case with 4634 DOF, with different viscosity jumps g . In the lower figure we show the preconditioned spectra.

set-volume tracking technique capable of computing the movement of multiple sediment layers. In this section we recall the main feature of that method, we provide some technicalities to reduce the computational burden, and finally their performance efficiency.

Let us introduce the nomenclature: let \mathcal{F}_A^S be a conformal grid obtained as a N_R -times uniform refinement of the grid \mathcal{F}_A^G , as depicted in Fig. 5. \mathcal{F}_A^S has n_e^S elements e_r^S and n_p^S nodes \tilde{x}_k^S . Let us consider the dual mesh provided with n_p^S node-centered cells τ_k^S . As regards its connectivity, let $\{k_j\} = \{k_j^C, j = 1, \dots, |k^C|\}$ be the set of the indexes of the cells surrounding a τ_k^S cell, and let $\{\tau_{\bar{j}}^S\}$ be the set of cells surrounding τ_k^S (see Fig. 5). The common surface between τ_k^S and $\tau_{\bar{j}}^S$ is the interface $l_{k,\bar{j}}^S$. We can also define the map $\bar{j} = \bar{j}(k, j), j = 1, \dots, |k^C|$ such that $l_{k,\bar{j}}^S = l_{k,j}^S$, in other terms $l_{k,\bar{j}}^S$ is the interface of $\tau_{\bar{j}}^S$ that corresponds to the j th interface of τ_k^S . For the sake of clarity, the index i will always identify the sediment layer, k will refer to cells and j to quantities related to the interfaces.

Let us now introduce the discrete functional spaces for the coupled LS-VT method: as it has a two-fold interpretation it has two distinct discrete spaces. Consider the discretized form of λ^n : $\lambda_A^n \in \mathbb{V}_0^S$ where

$$\mathbb{V}_0^S = \{\varphi_A \in L^2(\Omega) : \varphi_A|_{\tau_k^S} \in \mathbb{P}^0\}.$$

Let \mathbb{V}_0^S be piecewise constant and equipped with the base of the characteristic functions $\{\chi_k\}$ of the cells τ_k^S . Therefore we can expand λ_A^n as

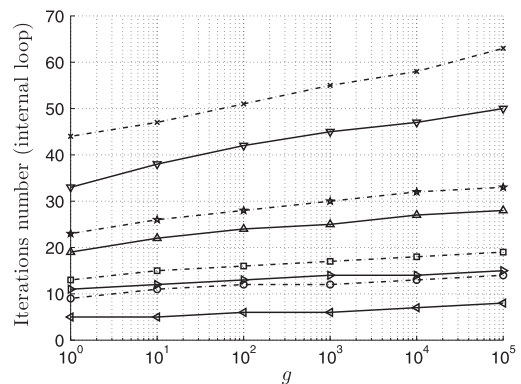
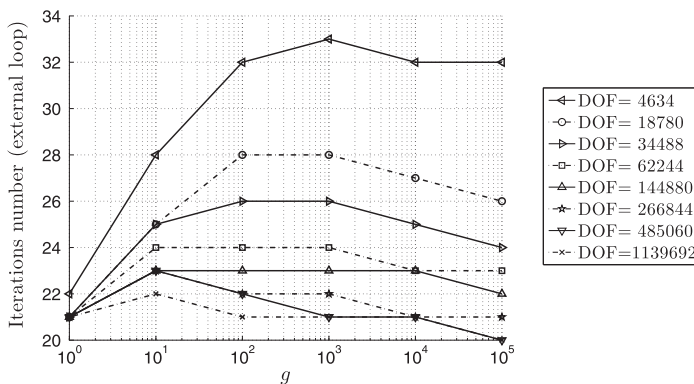


Fig. 3. On the left we show the number of iterations of the external cycle at different values of the viscosity jump g : the dependence upon g or Δ^G is not so evident. On the right we show the inner cycle iterations: there is an acceptable rise in the number of iterations.

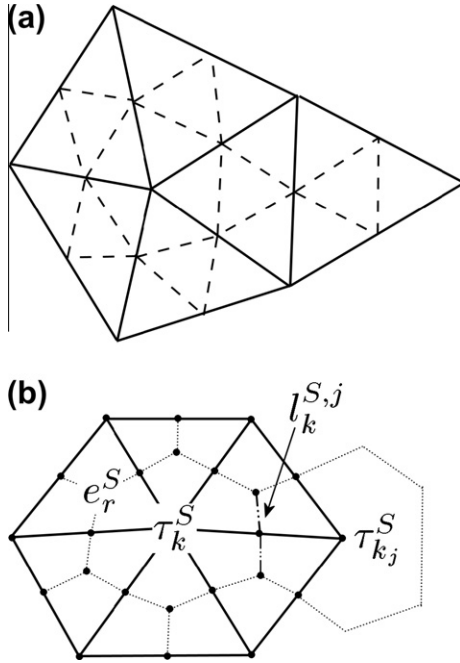


Fig. 5. We show two 2D sketches of the underlying meshes. Obviously we are handling their 3D counterparts. (a) The grid \mathcal{T}_Δ^S is identified by a solid line, and the dashed line defines the refinement procedure leading to \mathcal{T}_Δ^S . (b) A sketch of the \mathcal{T}_Δ^S mesh is shown, together with the cell τ_k^S , its neighbors τ_{kj}^S and the related interfaces l_{kj}^S . Lastly, a generic element e_r^S is depicted.

$$\lambda_\Delta^n = \sum_{k=1}^{n_p^S} \lambda_k^n \chi_k,$$

where λ_k^n are the cell mean values at time step n .

We consider then the first grade interpolant of λ_Δ^n , $\phi_\Delta^n \in \mathbb{V}_1^S$, with components $\phi_{i,\Delta}^n$, where $\mathbb{V}_1^S = \{\varphi|_{\mathcal{E}_r^S} \in \mathbb{P}^1\}$. Moreover we assume

$$\lambda_k^n = \phi_k^n, \tag{15}$$

where ϕ_k^n are the degrees of freedom of the level set field. Eq. (15) creates a relation between the LS and VT representations, making possible to accede to the information related to these methods from the solution of a single equation.

We now recall briefly the method described in [29]. Let us consider the following finite volume (FV) scheme

$$\lambda_k^{n+1} = \left(1 + \sum_{j=1}^{|\mathbb{J}_k^S|} v_{kj}^n \right) \lambda_k^n - \sum_{j=1}^{|\mathbb{J}_k^S|} \mathbf{F}_{kj}^n,$$

where v_{kj}^n are the interfaces Courant numbers

$$v_{kj}^n = \frac{\Delta t^n}{|\tau_k^S|} \int_{l_{kj}^S} \bar{\mathbf{u}}_\Delta^n \cdot \hat{\mathbf{n}}.$$

Here $\mathbf{F}_{kj}^n = v_{kj}^n \Phi(\hat{\lambda}_{kj}^n, \hat{\lambda}_{kj\bar{j}}^n)$ are the interface fluxes, and

$$\Phi(\hat{\lambda}_{kj}^n, \hat{\lambda}_{kj\bar{j}}^n) = \begin{cases} \hat{\lambda}_{kj}^n & \text{if } v_{kj}^n \geq 0, \\ \hat{\lambda}_{kj\bar{j}}^n & \text{if } v_{kj}^n < 0, \end{cases}$$

is the upwind function, where $\hat{\lambda}_{kj}^n$ and $\hat{\lambda}_{kj\bar{j}}^n$ are two suitable approximations of the composition near the interface l_{kj}^S from inside τ_k^S and τ_{kj}^S respectively. In other words, the last three equations define a FV scheme in which the interface fluxes are computed by means of an appropriate interface state $\hat{\lambda}_{kj}^n$, which is in general different from the cell mean value considered, for instance, in the Godunov meth-

od [13]. Our algorithm resembles more closely a MUSCL [1] or a Discontinuous Galerkin method [3] but differs from the latter for the definition of the interface states and the flux limiter. In fact we introduce the following definition of the interface states:

$$\hat{\lambda}_{i,kj}^n = \lambda_{i,k}^n + \gamma_{i,kj}^n \Delta \phi_{i,kj}^n,$$

where $\hat{\lambda}_{i,kj}^n$, $\lambda_{i,k}^n$ are i th components of $\hat{\lambda}_{k,j}^n$, λ_k^n . Moreover $\Delta \phi_{i,kj}^n$ is the i th component of the vector $\Delta \phi_{k,j}^n = \phi_{k,j}^n - \phi_{k\bar{j}}^n$ with

$$\phi_{k,j}^n = \frac{1}{|\tau_{k,j}^S|} \int_{\tau_{k,j}^S} \phi_\Delta^n, \quad \bar{\phi}_k^n = \frac{1}{|\tau_k^S|} \int_{\tau_k^S} \phi_\Delta^n.$$

Finally $\gamma_{i,kj}^n$ are the flux limiters and they are computed with the following maximization problem:

$$\begin{cases} J = \max \sum_{i=1}^{n_1} \gamma_{i,kj}^n, \\ \sum_{i=1}^{n_2} \gamma_{i,kj}^n \Delta \phi_{i,kj}^n = 0, \\ 0 \leq \gamma_{i,kj}^n \leq \gamma_{i,kj}^{n,\max}, \end{cases} \tag{16}$$

with

$$\begin{cases} \gamma_{i,kj}^{n,\max} = \min \left(1, \frac{\left(1 + \sum_{j=1}^{|\mathbb{J}_k^S|} v_{kj}^n\right) - v_{kj}^n |\mathbb{J}_k|}{v_{kj}^n |\mathbb{J}_k| \Delta \phi_{i,kj}^n} \lambda_{i,k}^n, \frac{1 - \lambda_{i,k}^n}{\Delta \phi_{i,kj}^n} \right) & \text{if } \Delta \phi_{i,kj}^n > 0, \\ \gamma_{i,kj}^{n,\max} = \min \left(1, -\frac{\lambda_{i,k}^n}{\Delta \phi_{i,kj}^n} \right) & \text{if } \Delta \phi_{i,kj}^n < 0, \end{cases} \tag{17}$$

where \mathbb{J}_k is the set of the indices of the outflow interfaces l_{kj}^S of the k th cell, i.e.,

$$\mathbb{J}_k = \{j \in 1, \dots, |\mathbb{J}_k^S| : v_{kj}^n \geq 0\}.$$

Problems (16) and (17) define implicitly the limiter $\gamma_{i,kj}^n \in [0, 1]$. We point out that if the limiter equals zero, our scheme equals the Godunov method. Now we define some techniques for reducing the computational cost. From Eqs. (16) and (17) it seems that we need to solve a minimization problem on all the interfaces at every time step. Actually, this is not necessary, as only the cells τ_k^S containing more than two species require an explicit solution of the minimization problem. In addition, most of the cells have only one or two species, whereas the remaining cells, that are near to three or more separating interfaces, are only a few. This is true also for more complicated models where the number of layers is higher or the layers themselves are thinner, since in these cases a finer FE discretization is required. Thus keeping the number of cells with three or more species much lower than the one with one or two species. Besides, for the two species-cells we can exploit a simplified version of (16):

$$\begin{cases} \gamma_{i_1,kj}^n = \min \left(1, \frac{\Delta \phi_{i_2,kj}^n}{\Delta \phi_{i_1,kj}^n} \gamma_{i_2,kj}^{n,\max}, \gamma_{i_1,kj}^{n,\max} \right), \\ \gamma_{i_2,kj}^n = \min \left(1, \frac{\Delta \phi_{i_1,kj}^n}{\Delta \phi_{i_2,kj}^n} \gamma_{i_1,kj}^{n,\max}, \gamma_{i_2,kj}^{n,\max} \right), \\ \gamma_{i,kj}^n = 1 \quad \text{with } i \neq i_1, i_2, \end{cases} \tag{18}$$

where i_1 and i_2 are the indices of the two species that fill the cell we are considering. The computational cost could be reduced further; let us consider the following condition:

$$\lambda_k^n = \lambda_{i,kj}^n \quad \forall j = 1, \dots, |\mathbb{J}_k^S|. \tag{19}$$

At the beginning of a numerical run most of the cells satisfy this condition as they are far away from the interfaces. Their number tends to decrease during the simulation run while it rapidly increases when the LS function is reinitialized, as we can see in

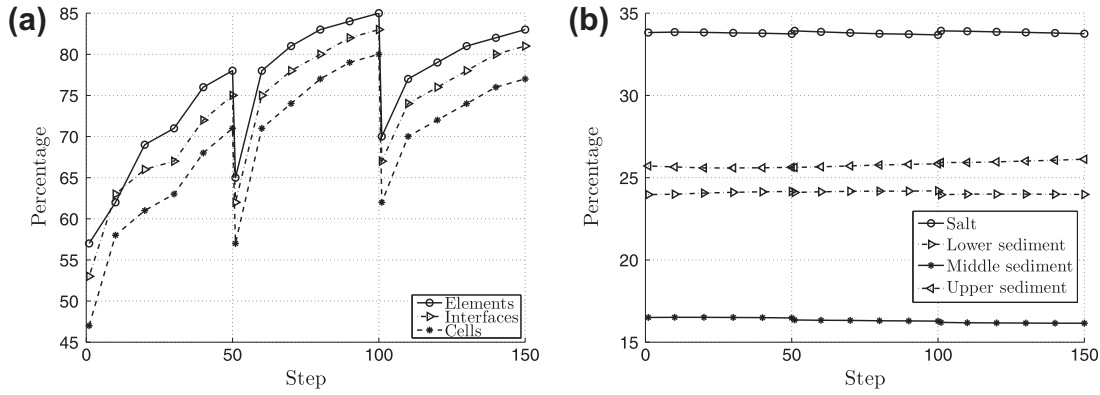


Fig. 6. In figure (a), we show the percentage of the cells, elements, and interfaces inside the active band. The active band is composed by the cells that do not satisfy (19). In this case the grid is coarse therefore a lot of cells lie near an interface. However we can still achieve an average 20% reduction of the computational cost. In figure (b), we show the percentage volume of the four species. At steps 50 and 100 a reinitialization algorithm is applied.

Fig. 6. Every cell that satisfies (19) has a trivial solution $\lambda_k^{n+1} = \lambda_k^n$, therefore we have only to update the composition of those cells that does not satisfy (19). In Fig. 6 we show some results about the computational cost reduction in a typical geological simulation with four sedimentary layers, that we will discuss in details in the numerical results section. The great majority (more than 98% in current test case) of the cells has less than two species and the simplified form (18) can be applied. Finally in Fig. 6b we can appreciate the good conservation properties even during the reinitialization of $\phi_{i,A}^n$ at time steps 50 and 100.

5. Algorithm workflow

In this section we describe briefly the workflow of the algorithm we have developed, which is schematically sketched in Fig. 7.

At the general time step t^n we know the composition vector λ_Δ^n , which contains the volume fractions $\lambda_{i,k}^n$ necessary to track the position of the sedimentary layers and in particular of the horizons.

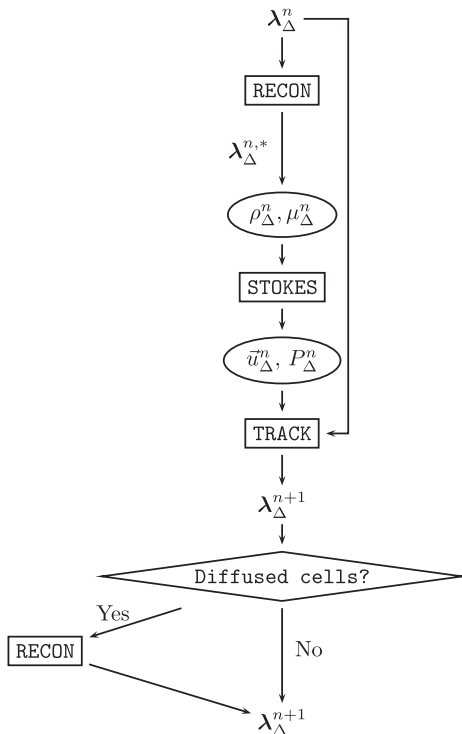


Fig. 7. Algorithm workflow.

From this information, the reconstruction algorithm **RECON** (described below) is applied to all the volume fractions $\lambda_{i,k}^n$, leading to an updated copy $\lambda_{i,k}^{n,*}$ of the composition vector. This procedure is important to limit the effects of numerical diffusion: indeed, it reduces the diffusion layer of the properties distribution to the size of a cell, thus enhancing the accuracy of the forthcoming computation of the velocity field. We briefly recall the algorithm **RECON** (described in [29]) in the following:

Algorithm 2. At the generic time step n , if there is an index \bar{i} such that

$$\lambda_{\bar{i},k}^n > \frac{1}{2} \quad \text{and} \quad \lambda_{\bar{i},k_j}^n > \frac{1}{2}, \quad j = 1, \dots, |\mathbb{U}_k^S|,$$

then set $\lambda_{\bar{i},k}^n = 1$ and $\lambda_{\bar{i},k}^n = 0$ with $i \neq \bar{i}$. Otherwise maintain the nodal value $\lambda_{i,k}^n$.

From the new vector $\lambda_{i,k}^{n,*}$ we compute the physical quantities ρ_A^n and μ_A^n through Eq. (4), and then solve the Stokes problem in the block **STOKES**. This solver generates the pressure and the velocity fields P_A^n and \bar{u}_A^n . The latter, together with $\lambda_{i,k}^n$, finally leads to the new composition vector $\lambda_{i,k}^{n+1}$, by means of the algorithm **TRACK**:

Algorithm 3. Starting from $\lambda_{i,k}^n$ and \bar{u}_A^n compute $\lambda_{i,k}^{n+1}$ following these three steps:

- (i) construct the database of the cell to be updated;
- (ii) compute the time step satisfying the condition that $\max_k \max_j v_{k,j}^n < \bar{v}$ where \bar{v} is set equal to 1/5. The number of intermediate time steps necessary to arrive to t^{n+1} from t^n is also computed;
- (iii) compute the interface fluxes, and update the cell partial volumes.

We remark that at this stage, the new composition is tracked starting from the original composition $\lambda_{i,k}^n$, to ensure the conservativity of our method. Nevertheless, in the last part of the code, before the beginning of the succeeding time step, we perform a test on the diffusion of the composition, as we need to keep it bounded inside the domain: if the number of diffused cells exceeds an imposed threshold, **Algorithm 2** is applied also to the new composition vector $\lambda_{i,k}^{n+1}$.

6. Numerical results

The test case we analyze has already been introduced in Fig. 1. It consists in a sedimentary basin divided into four layers by three horizons. The basin dimensions are $10.3 \times 15.6 \times 5.8$ km, and the

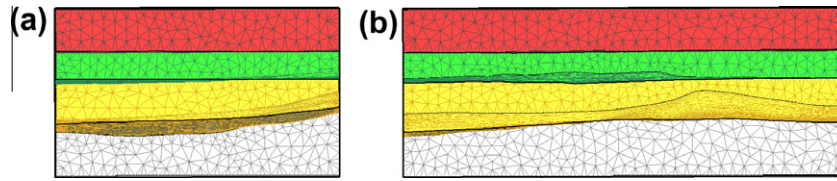


Fig. 8. A couple of lateral views of the basin considered in the simulations. Salt is in white, and heavier sediments are in yellow and green. In figure (b), a bump is visible between the white and the yellow layers, which will lead to a diapir growth as a consequence of the Rayleigh–Taylor instability. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

evolution time we consider is equal to 34.35 Mya. Density and viscosity have been taken respectively $2.2 \cdot 10^3 \text{ kg/m}^3$ and $0.1 \cdot 10^{20} \text{ Pa s}$ for the salt layer and $2.0\text{--}2.6 \cdot 10^3 \text{ kg/m}^3$ and $10^{20}\text{--}10^{21} \text{ Pa s}$ for the overbearing layers, that are reasonable values physically speaking. Among the several simulations run, the one we present here has about 900k unknowns, and requires approximately 4.2 Gb of RAM. The computations have been run on an AMD Opteron 8212 Dual-Core 2 GHz processor.

With respect to a wide variety of two-dimensional simulations, only a few examples of 3D cases on sedimentary basins already exist in literature, for example Zadeh [10] or Kaus and Podladchikov [12]. The main reason of such a lack of references resides in the dramatic rise of the computational cost that the switch to 3D implies. But then, if compared to a 2D domain, a 3D model becomes necessary to capture the complete basin dynamics. Anyway, as for the 2D case, the domain boundaries may introduce some undesired effects, worsening the quality of the results. This is mainly

due to the imposition of the boundary conditions on a limited domain, where the free slip condition in the vertical direction on the boundaries (the last equation in problem (1)) is a good approximation as long as the vertical stress in the real basin is low. In addition the exact values of velocity and stress and their spatial and temporal variation laws, coming from geological analysis or data, are still affected by large uncertainty.

In this sense, the simulation of this work not only represents an enrichment of the set of test cases in a 3D domain, but also provides specific features that lead to significant results both from the physical and mathematical point of view. First of all, the implementation of a numerical code for 3D multiflow simulations represents an innovation in the stratified fluid dynamics field, as, with respect to the above-mentioned two-layer 2D simulations, it can handle a model with an arbitrary number of layers. In addition, the model geometry is composed of interfaces representing realistic sedimentary layers, and therefore the perturbation causing the

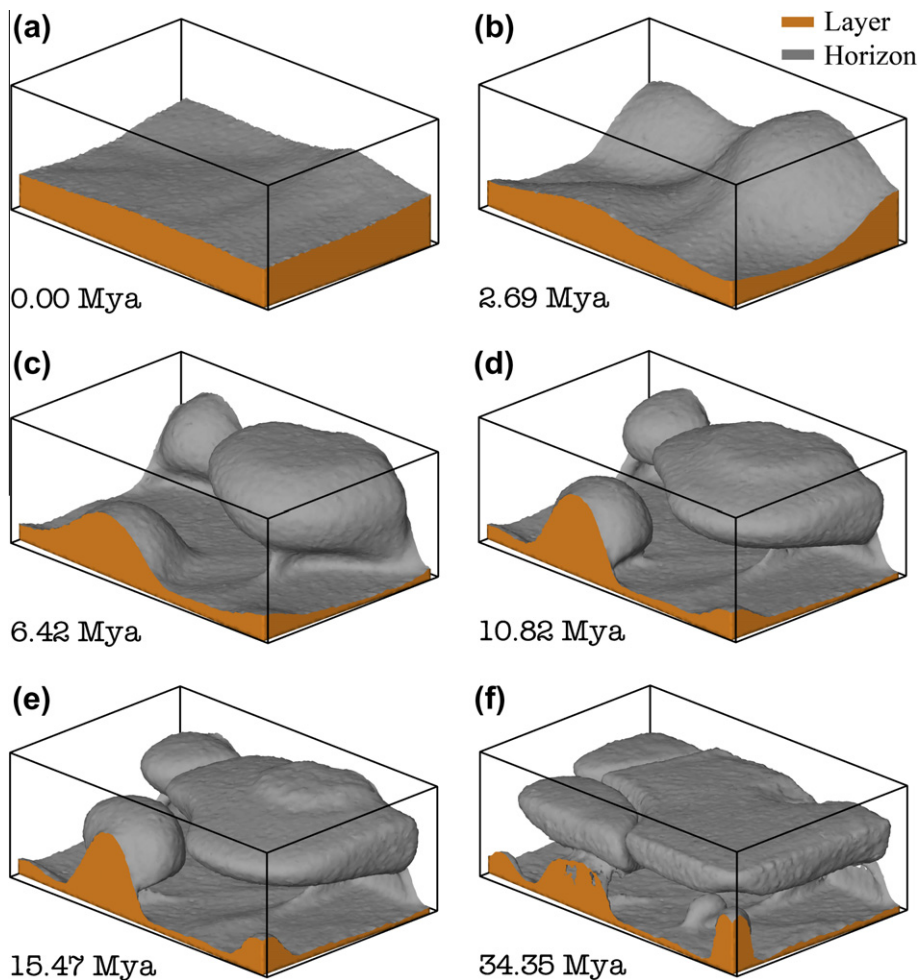


Fig. 9. Progressive evolution of salt diapirs: in about 34 Mya the growth of the three diapirs is almost complete.

Rayleigh–Taylor instabilities are not imposed *a priori* on a plane surface, but originate from the physical shape of the horizons (see Fig. 8).

The ease of handling complicated geometries is enhanced by the use of unstructured meshes as discretizing tool. Fig. 9 shows subsequent steps of the basin evolution, that leads to the shaping of some salt diapirs.

Secondly, the interface surfaces have been drawn with the innovative tracking algorithm, that is able to reconstruct the horizon positions efficiently, both from the geometric and the computational point of view. In particular, as these figures show, it is able to handle and represent topological changes, and so to simulate correctly and in a fully automatic way phenomena of key importance in the sedimentary basin evolution, such as salt diapir detachments (see Fig. 9e) and horizon intersections (see Fig. 10b–d). In the following figures we illustrate the evolution of the surface between the salt and the overburden layers, that ends with the formation of three main diapirs.

As well as the interfaces evolution, the distribution of other quantities of physical interest is computed, such as velocity, pressure and strain fields. As an example, in Fig. 11–13 we show the distribution of the vertical stress component, the vertical velocity and the streamlines on a sectioning plane. In the first we can recognize three different phases in the basin evolution: the most part of the diapir growth happens in the first 11 Mya, followed by a settlement phase lasting 24 Mya, and here-hence the evolution is almost stationary. In the second figure we can appreciate the streamlines representing the flow motion leading to the formation of the main diapir, at a time step of basin major activity.

From the computational viewpoint, this test case lasts 180 time steps, each one requiring the solution of one Stokes problem. Since the numerical scheme is time-adaptive, the single step length varies according to the evolution rate of the model, and in this simulation ranges from 0.07 Mya to 1.61 Mya. The mean time step is 0.2 Mya. As regards the tracking time step, we implement a sub-stepping procedure: for each Stokes solution we advance the horizons tracking using 40 substeps. Although a total of 7200 tracking steps were performed, the solution of the Stokes problem proved to be the most time-consuming part of the code.

7. Conclusions

In this article we presented an innovative numerical technique for the mathematical and physical modeling of stratified salt sedimentary basins. In particular we focused on numerical efficiency and computational time reduction, operating both on the solution of the linear system resolution and on the interface tracking algorithm.

Firstly, we faced the problem related to the resolution of the linear system, which is derived from the FE method adopted to solve the set of governing equation. The huge matrix coming from the high number of degrees of freedom shows a dependence from the viscosity, that may have a large range of variability among the different basin layers. The algorithm we propose exploits this dependence of the external loop to build a convenient preconditioning matrix based on the Schur complement, depending on the viscosity itself, that has provided a sensible gain in the calculation time.

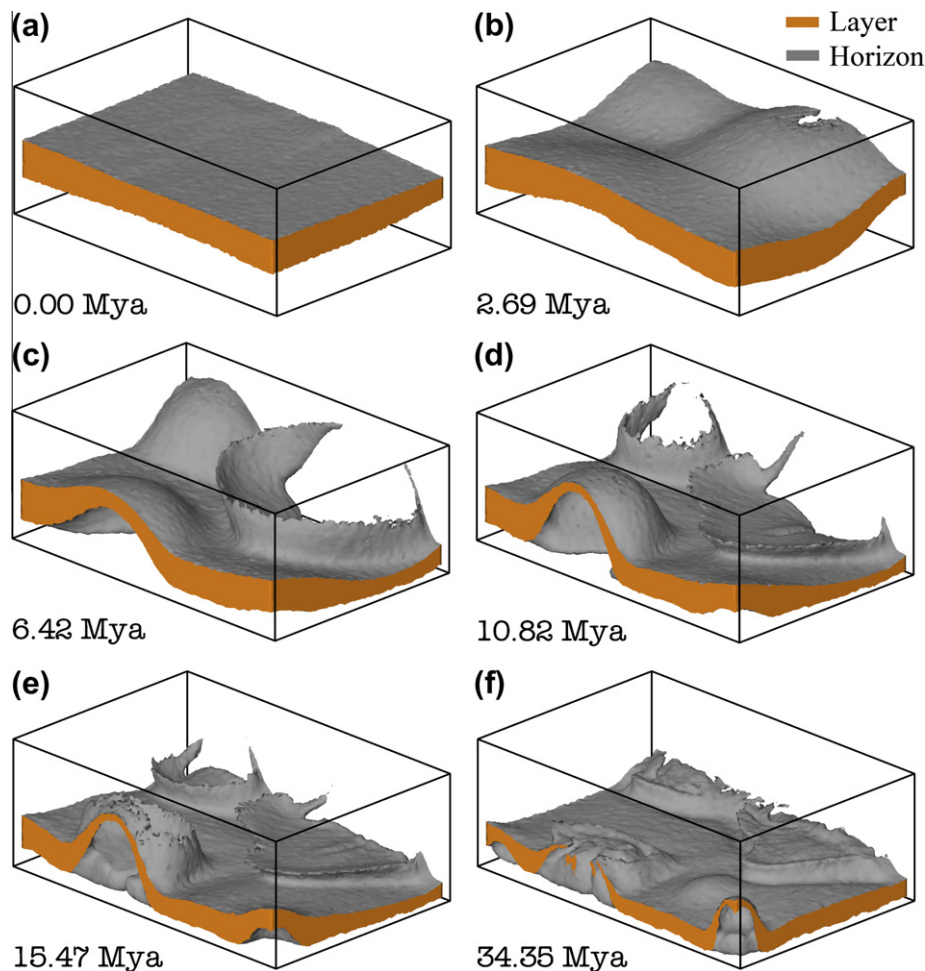


Fig. 10. Progressive evolution of the lower sediment: the salt rise perforates the overbearing sediment in three regions.

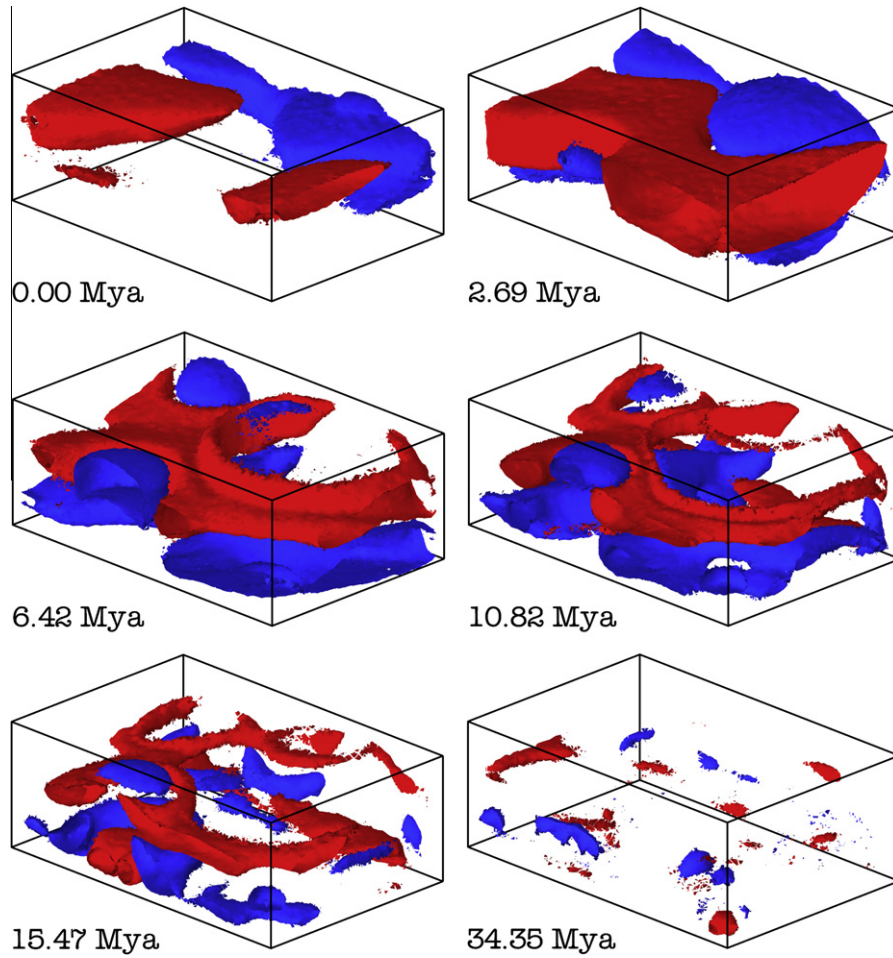


Fig. 11. Two isosurfaces showing the distribution of the vertical stress component $\bar{\sigma}_{zz}$. (Red: 2.0 MPa, Blue: -2.0 MPa). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Moreover we presented an evolution of the implicit horizon tracking method applied to a 3D case test, with a number of layers major than two. This algorithm puts together the advantages coming from volume tracking and level set methods, and can both accomplish the reconstruction of the interfaces movement with good accuracy, and solve the problems of the topological changes, extremely significant in a 3D domain. In particular, we focused on an efficient way to compute a simplified solution of the method limiter, that, coupled with the exploitation of the localized distribution of the involved cells, leads to a dramatic reduction of the global computational time.

Finally we applied these techniques to a test case, and the results we presented state the efficiency in simulating the evolution of a realistic sedimentary basin in a 3D domain.

Further development will concern the enrichment of the model and mathematical improvements. Interesting tasks will be the extension of the geometry representation to include faults and sedimentation and compaction processes, and the implementation of more sophisticated physical models such as non Newtonian rheologies.

Acknowledgments

The authors wish to acknowledge the support of the Italian MIUR (Grant PRIN 2007) and G. Scrofani and P. Ruffo for having supported such an interesting research project. They also wish to thank L. Formaggia, E. Miglio and F. Nobile from MOX for their precious advices and suggestions.

Appendix A. Shur complement conditioning number estimate

The conditioning number of the Schur complement $\mathcal{K}(\mathbf{BA}^{-1}\mathbf{B}^T)$ is not dependent upon Δ , more explicitly:

$$\mathcal{K}(\mathbf{BA}^{-1}\mathbf{B}^T) \leq \frac{\sup(\mu)}{\inf(\mu)} \frac{C_6}{C_5} \left(\frac{C_1 C_2^2 C_3}{\beta_A^2} \right)^2.$$

Proof. Combining the discrete inf-sup condition (7) and Eq. (10) we get:

$$\min_{q_A \in \mathcal{V}_A^Q} \max_{\bar{v}_A \in \mathcal{V}_A^M} \frac{|(q_A, \bar{\nabla} \cdot \bar{u}_A)|}{\|\bar{\nabla} \bar{u}_A + (\bar{\nabla} \bar{u}_A)^T\|_{L^2(\Omega)} \|q_A\|_{L^2(\Omega)}} \geq \frac{\beta_A}{C_2}. \quad (\text{A.1})$$

The norm of the weighted symmetric gradient can be bounded as follows:

$$\begin{aligned} \|\bar{\nabla} \bar{u}_A + (\bar{\nabla} \bar{u}_A)^T\|_{L^2(\Omega)} &= \left\| \frac{1}{\sqrt{\mu}} \sqrt{\mu} (\bar{\nabla} \bar{u}_A + (\bar{\nabla} \bar{u}_A)^T) \right\|_{L^2(\Omega)} \\ &\geq \frac{1}{\max(\sqrt{\mu})} \|\sqrt{\mu} (\bar{\nabla} \bar{u}_A + (\bar{\nabla} \bar{u}_A)^T)\|_{L^2(\Omega)}, \end{aligned} \quad (\text{A.2})$$

and plugging Eq. (A.2) in (A.1) we get

$$\begin{aligned} \min_{q_A \in \mathcal{V}_A^Q} \max_{\bar{v}_A \in \mathcal{V}_A^M} \frac{|(q_A, \bar{\nabla} \cdot \bar{u}_A)|}{\|\sqrt{\mu} (\bar{\nabla} \bar{u}_A + (\bar{\nabla} \bar{u}_A)^T)\|_{L^2(\Omega)} \|q_A\|_{L^2(\Omega)}} \\ \geq \frac{1}{\max(\sqrt{\mu})} \frac{\beta_A}{C_2}. \end{aligned} \quad (\text{A.3})$$

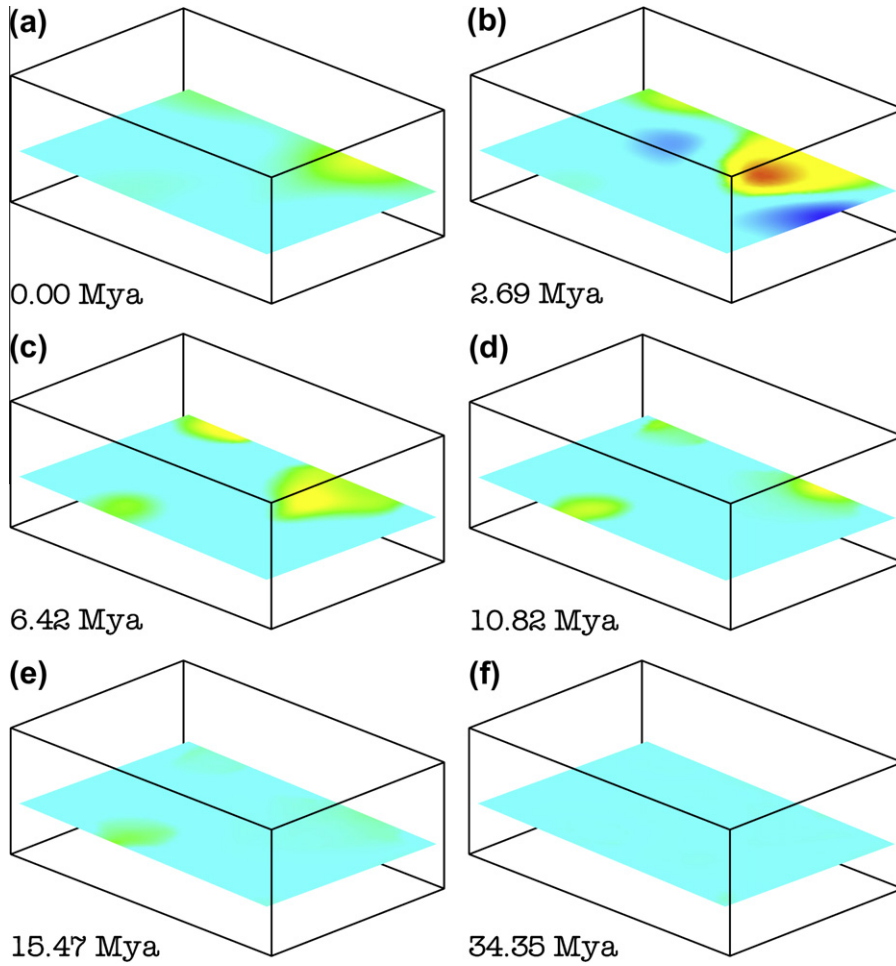


Fig. 12. A series of slabs of vertical velocity V_3 , at $x_3 = 2.18$ km from the bottom of the basin: the most part of the development of the diapirs happens in the first 11 Mya, while in the last 24 Mya the flows become gradually stationary. (Red: 1.64 km/Mya, Cyan: 0.0 km/Mya, Blue: -0.82 km/Mya). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

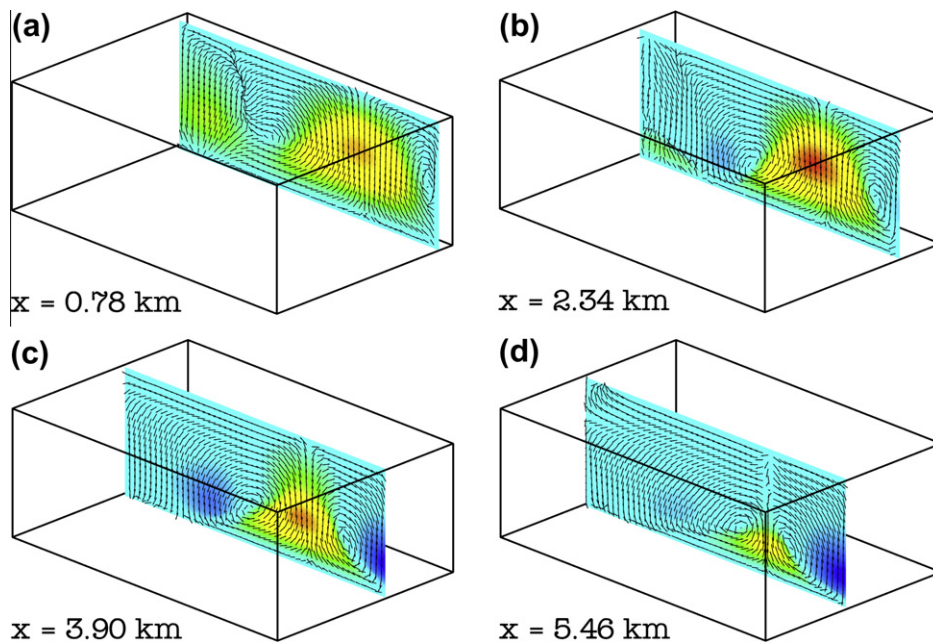


Fig. 13. A series of slabs of vertical velocity V_3 and stream-lines at $t = 2.69$ Mya: two main vortices are visible near the biggest diapir. (Red: 1.64 km/Mya, Cyan: 0.0 km/Mya, Blue: -0.82 km/Mya). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

From the expansions (8) we obtain

$$\begin{aligned} (q_A, \vec{\nabla} \cdot \vec{u}_A) &= \mathbf{P}^T \mathbf{B} \mathbf{V}, \\ \|q_A\|_{L^2(\Omega)} &= \sqrt{\mathbf{P}^T \mathbf{Q} \mathbf{P}}, \\ \|\sqrt{\mu}(\vec{\nabla} \vec{u}_A + (\vec{\nabla} \vec{u}_A)^T)\|_{L^2(\Omega)} &= \sqrt{\mathbf{V}^T \mathbf{A} \mathbf{V}}, \end{aligned}$$

where \mathbf{Q} is the mass matrix of the pressure field and $\mathbf{P} \in \mathbb{R}^{n_p}$, $\mathbf{V} \in \mathbb{R}^{3n_p}$. Hence, (A.3) can be rewritten as

$$\min_{\mathbf{P}} \max_{\mathbf{V}} \frac{|(\mathbf{P}^T \mathbf{B} \mathbf{V})|}{\sqrt{\mathbf{V}^T \mathbf{A} \mathbf{V}} \sqrt{\mathbf{P}^T \mathbf{Q} \mathbf{P}}} \geq \frac{1}{\max(\sqrt{\mu})} \frac{\beta_A}{C_2}.$$

Now let us introduce $\mathbf{W} = \mathbf{A}^{1/2} \mathbf{V}$, therefore

$$\frac{1}{\max(\sqrt{\mu})} \frac{\beta_A}{C_2} \leq \min_{\mathbf{P}} \max_{\mathbf{W}} \frac{|\mathbf{P}^T \mathbf{B} \mathbf{A}^{-1/2} \mathbf{W}|}{\sqrt{\mathbf{W}^T \mathbf{W}} \sqrt{\mathbf{P}^T \mathbf{Q} \mathbf{P}}}.$$

This equation implies $\mathbf{W} = \mathbf{A}^{-1/2} \mathbf{B}^T \mathbf{P}$, that yields to

$$\frac{1}{\max(\sqrt{\mu})} \frac{\beta_A}{C_2} \leq \frac{\sqrt{|\mathbf{P}^T \mathbf{B} \mathbf{A}^{-1} \mathbf{B}^T \mathbf{P}|}}{\sqrt{\mathbf{P}^T \mathbf{Q} \mathbf{P}}}, \tag{A.4}$$

thus completing the lower bound estimate. As regards the upper bound, combining (9) and (11) with the Cauchy-Schwartz inequality, we have

$$\begin{aligned} |(q_A, \vec{\nabla} \cdot \vec{u}_A)| &\leq \|q_A\|_{L^2(\Omega)} \|\vec{\nabla} \cdot \vec{u}_A\|_{L^2(\Omega)} \\ &\leq C_1 C_3 \|q_A\|_{L^2(\Omega)} \|\vec{\nabla} \vec{u}_A + (\vec{\nabla} \vec{u}_A)^T\|_{L^2(\Omega)} \\ &\leq \frac{C_1 C_3}{\min(\sqrt{\mu})} \|q_A\|_{L^2(\Omega)} \|\sqrt{\mu}(\vec{\nabla} \vec{u}_A + (\vec{\nabla} \vec{u}_A)^T)\|_{L^2(\Omega)}, \end{aligned}$$

or the equivalent form

$$\frac{|(q_A, \vec{\nabla} \cdot \vec{u}_A)|}{\|q_A\|_{L^2(\Omega)} \|\sqrt{\mu}(\vec{\nabla} \vec{u}_A + (\vec{\nabla} \vec{u}_A)^T)\|_{L^2(\Omega)}} \leq \frac{C_1 C_3}{\min(\sqrt{\mu})}.$$

Applying the same argument used for the lower bound we get:

$$\frac{\sqrt{|\mathbf{P}^T \mathbf{B} \mathbf{A}^{-1} \mathbf{B}^T \mathbf{P}|}}{\sqrt{\mathbf{P}^T \mathbf{Q} \mathbf{P}}} \leq \frac{C_1 C_3}{\min(\sqrt{\mu})}. \tag{A.5}$$

Now, combining the lower bound (A.4) and the upper bound (A.5) we obtain

$$\frac{1}{\max(\mu)} \frac{\beta_A^2}{C_2^2} \leq \frac{|\mathbf{P}^T \mathbf{B} \mathbf{A}^{-1} \mathbf{B}^T \mathbf{P}|}{\mathbf{P}^T \mathbf{Q} \mathbf{P}} \leq \frac{C_1^2 C_3^2}{\min(\mu)},$$

that, multiplied by $(\mathbf{P}^T \mathbf{Q} \mathbf{P})/(\mathbf{P}^T \mathbf{P})$, leads to

$$\frac{1}{\max(\mu)} \frac{\beta_A^2}{C_2^2} \frac{\mathbf{P}^T \mathbf{Q} \mathbf{P}}{\mathbf{P}^T \mathbf{P}} \leq \frac{|\mathbf{P}^T \mathbf{B} \mathbf{A}^{-1} \mathbf{B}^T \mathbf{P}|}{\mathbf{P}^T \mathbf{P}} \leq \frac{C_1^2 C_3^2}{\min(\mu)} \frac{\mathbf{P}^T \mathbf{Q} \mathbf{P}}{\mathbf{P}^T \mathbf{P}}. \tag{A.6}$$

From [5] we have

$$C_5 \Delta^2 \leq \frac{\mathbf{P}^T \mathbf{Q} \mathbf{P}}{\mathbf{P}^T \mathbf{P}} \leq C_6 \Delta^2,$$

that combined with (A.6) yields to

$$\frac{C_5}{\max(\mu)} \frac{\beta_A^2}{C_2^2} \Delta^2 \leq \frac{|\mathbf{P}^T \mathbf{B} \mathbf{A}^{-1} \mathbf{B}^T \mathbf{P}|}{\mathbf{P}^T \mathbf{P}} \leq \frac{C_1^2 C_3^2 C_6}{\min(\mu)} \Delta^2,$$

and the proof holds. \square

Appendix B. Stiffness matrix conditioning number estimate

The conditioning number of \mathbf{A} is bounded by

$$\mathcal{K}(\mathbf{A}) \leq \frac{\max(\mu)}{\inf(\mu)} \frac{1}{\Delta^2} \frac{C_8}{C_7} (C_1 C_2 C_5 C_p)^2.$$

Proof. We can bound the weighted symmetric gradient through Eqs. (10) and (12) for the upper bound

$$\begin{aligned} \|\sqrt{\mu}(\vec{\nabla} \vec{u}_A + (\vec{\nabla} \vec{u}_A)^T)\|_{L^2(\Omega)} &\leq \max(\sqrt{\mu}) \|\vec{\nabla} \vec{u}_A + (\vec{\nabla} \vec{u}_A)^T\|_{L^2(\Omega)} \\ &\leq C_2 \max(\sqrt{\mu}) \|\vec{\nabla} \vec{u}_A\|_{L^2(\Omega)} \\ &\leq \frac{C_2 C_4}{\Delta} \max(\sqrt{\mu}) \|\vec{u}_A\|_{L^2(\Omega)}, \end{aligned} \tag{B.1}$$

and through Eqs. (9) and (13) for the lower bound

$$\begin{aligned} \|\sqrt{\mu}(\vec{\nabla} \vec{u}_A + (\vec{\nabla} \vec{u}_A)^T)\|_{L^2(\Omega)} &\geq \min(\sqrt{\mu}) \|\vec{\nabla} \vec{u}_A + (\vec{\nabla} \vec{u}_A)^T\|_{L^2(\Omega)} \\ &\geq \frac{\min(\sqrt{\mu})}{C_1} \|\vec{\nabla} \vec{u}_A\|_{L^2(\Omega)} \\ &\geq \frac{\min(\sqrt{\mu})}{C_1 C_p} \|\vec{u}_A\|_{L^2(\Omega)}. \end{aligned} \tag{B.2}$$

The two inequalities (B.1) and (B.2) can be coupled as

$$\frac{\min(\mu)}{C_1^2 C_p^2} \leq \frac{\|\sqrt{\mu}(\vec{\nabla} \vec{u}_A + (\vec{\nabla} \vec{u}_A)^T)\|_{L^2(\Omega)}^2}{\|\vec{u}_A\|_{L^2(\Omega)}^2} \leq \frac{C_2^2 C_4^2}{\Delta^2} \max(\mu),$$

or, more conveniently, from (8)

$$\frac{\min(\mu)}{C_1^2 C_p^2} \leq \frac{(\mathbf{V}^T \mathbf{A} \mathbf{V})}{(\mathbf{V}^T \mathbf{Q} \mathbf{V})} \leq \frac{C_2^2 C_4^2}{\Delta^2} \max(\mu),$$

where \mathbf{Q}_V is the mass matrix of the velocity discrete field. Multiplying this by $(\mathbf{V}^T \mathbf{Q}_V \mathbf{V})/(\mathbf{V}^T \mathbf{V})$ we get

$$\frac{\mathbf{V}^T \mathbf{Q}_V \mathbf{V}}{\mathbf{V}^T \mathbf{V}} \frac{\min(\mu)}{C_1^2 C_p^2} \leq \frac{(\mathbf{V}^T \mathbf{A} \mathbf{V})}{(\mathbf{V}^T \mathbf{V})} \leq \frac{C_2^2 C_4^2}{\Delta^2} \max(\mu) \frac{\mathbf{V}^T \mathbf{Q}_V \mathbf{V}}{\mathbf{V}^T \mathbf{V}}. \tag{B.3}$$

From [5] we have

$$\Delta^2 C_7 \leq \frac{\mathbf{V}^T \mathbf{Q}_V \mathbf{V}}{\mathbf{V}^T \mathbf{V}} \leq C_8 \Delta^2,$$

that coupled with (B.3) leads to

$$C_7 \frac{\min(\mu)}{C_1^2 C_p^2} \Delta^2 \leq \frac{(\mathbf{V}^T \mathbf{A} \mathbf{V})}{(\mathbf{V}^T \mathbf{V})} \leq (C_2^2 C_4^2 C_8) \max(\mu),$$

and to the proof. \square

References

- [1] Batten P, Lamberta C, Causon DM. Positively conservative high resolution convection scheme for unstructured elements. *Int J Numer Methods Eng* 1996;39(3–4):1821–38.
- [2] Cloetingh S, Podladchikov Y. Perspectives in tectonic modeling. *Tectonophysics* 2000;320:169–73.
- [3] Cockburn B, Shu CW. TVB Runge Kutta local projection discontinuous Galerkin finite element method for conservation laws 2: general framework. *Math Comput* 1989;52:411–35.
- [4] Duran RG, Muschietti MA. The Korn inequality for Jones domains. *Electron J Diff Eqns* 2004;127:1–10.
- [5] Elman C, Silvester DJ, Wathen AJ. *Finite elements and fast iterative solvers*. Oxford University Press; 2005.
- [6] Ern A, Guermond JL. *Theory and practice of finite elements*. Springer; 2000.
- [7] Fulsack P. SOPALE: a finite-element computer code for the computation of visco-plastic creeping flows with applications to tectonics; 1995. <<http://geodynamics.oceanography.dal.ca/sopaledoc.html>>.
- [8] Fulsack P. An arbitrary lagrangian Eulerian formulation for creeping flows and its application in tectonic models. *Geophys J Int* 1995;120:1–23.

- [9] Ismail-Zadeh AT, Talbot CJ, Volozh YuA. Dynamic restoration of profiles across diapiric salt structures: numerical approach and its applications. *Tectonophysics* 2001;337:21–36.
- [10] Ismail-Zadeh AT, Tsepelev IA, Talbot C, Korotkii A. Three-dimensional forward and backward modelling of diapirism: numerical approach and its applicability to the evolution of salt structures in the Pricaspian basin. *Int J Numer Methods Eng* 2004;387:81–103.
- [11] Ismail-Zadeh AT, Tsepelev IA, Talbot C, Oster P. A numerical method and parallel algorithm for three-dimensional modeling of salt diapirism. *Problems in dynamics and seismicity of the earth*, vol. 31. GEOS; 2000. p. 62–76.
- [12] Kaus BJP, Podladchikov Y. Forward and reverse modeling of the three-dimensional viscous Rayleigh–Taylor instability. *Geophys Res Lett* 2001;28:1095–8.
- [13] LeVeque RJ. *Numerical methods for conservation laws*. Birkhauser; 1999.
- [14] Massimi P. *Perspective in computational salt tectonics*. PhD thesis, Politecnico di Milano; 2006.
- [15] Massimi P, Quarteroni A, Saleri F, Scrofani G. Modeling of salt tectonics. *Comput Methods Appl Mech Eng* 2007;197:281–93.
- [16] Massimi P, Quarteroni A, Scrofani G. An adaptive finite element method for modeling salt diapirism. *Math Mod Methods Appl Sci* 2007;16(4):587–614.
- [17] May DA, Moresi L. Preconditioned iterative methods for Stokes flow problems arising in computational geodynamics. *Phys Earth Planet In* 2008;171:33–47.
- [18] Moresi L, Dufour F, Muhlhaus HB. A Lagrangian integration point finite element method for large deformation modeling of viscoelastic geomaterials. *J Comput Phys* 2003;184:476–97.
- [19] O'Neill C, Moresi L, Muller D, Albert R, Dufour F, et al. Ellipsis 3D: a particle-in-cell finite-element hybrid code for modelling mantle convection. *Comput Geosci* 2006;1(2):1769–79.
- [20] Osher S, Fedkiw RP. *Level set methods and dynamic implicit surfaces*. Springer; 2003.
- [21] Peric D, Crook AJL. Computational strategies for predictive geology with reference to salt tectonics. *Comput Methods Appl Mech Eng* 2004;193:5195–222.
- [22] Quarteroni A, Valli A. *Numerical approximation of partial differential equations*. 2nd ed. Springer-Verlag; 1997.
- [23] Ruuth SJ. A diffusion-generated approach to multiphase motion. *J Comput Phys* 1998;145:166–92.
- [24] Scotti A, Villa A. Predictive numerical models of basin evolution and petroleum generation. In: *Applied and industrial mathematics in Italy III. Advances in mathematics for applied sciences*, vol. 82; 2009. p. 554–65.
- [25] Scrofani G. *Numerical basins modeling and tectonics*. PhD thesis, Politecnico di Milano; 2007.
- [26] Sethian A. *Level set methods and fast marching methods*. Cambridge University Press; 1999.
- [27] Tan L, Zabaras N. A level set simulation of dendritic solidification of multi-component alloy. *J Comput Phys* 2007;221:9–40.
- [28] Turcotte DL, Schubert G. *Geodynamics*. Cambridge University Press; 2001.
- [29] Villa A, Formaggia L. Implicit tracking for multi-fluid simulations. *J Comput Phys* 2010;229:5788–802.
- [30] Zaleski S, Julien P. Numerical simulation of Rayleigh–Taylor instability for single and multiple salt diapirs. *Tectonophysics* 1991;206:55–69.
- [31] Zlotnik S, Diez P. Hierarchical X-FEM for n-phase flow ($n > 2$). *Comp Methods Appl Mech Eng* 2009;198:2329–38.