

Andrea Villa

Three Dimensional Geophysical Modeling:
from Physics to Numerical Simulation

Ledizioni LediPublishing

Andrea Villa
Department of Mathematics
University of Milan
Via C. Saldini n.50
20133 Milan - Italy

Series Editors:

Vincenzo Capasso,
Coordinator of the PhD Program in Mathematics and Statistics for
Computational Sciences;
Alfredo Lorenzi,
Coordinator of the PhD Program in Mathematics;
Bernhard Ruf,
Director of the Graduate School.

Editorial Assistant: Stefania Leonardi

© 2010 Andrea Villa
Open Access copy available at air.unimi.it
First Edition: September 2010
ISBN 978-88-95994-14-7
All rights reserved for the print copy.
For the digital copy rights please refer to air.unimi.it

Ledizioni LediPublishing
Via Alamanni 11 – 20141 Milano – Italy
<http://www.ledipublishing.com>
info@ledipublishing.com

Andrea Villa

Three Dimensional Geophysical Modeling: from Physics to Numerical Simulation

September 15, 2010

In memory of Fausto Saleri

Acknowledgements

I wish to acknowledge the actual and former junior members of the numerical geology group at MOX: Anna Scotti, Davide Soresinetti, Matteo Longoni, Franco Dassi, Cristiano Malossi, Marco Cuffaro and Alessio Fumagalli.

I am grateful to the senior members of the numerical geology group at MOX: Prof. Alfio Quarteroni, Prof. Luca Formaggia, Dr. Fabio Nobile, Dr. Edie Miglio, Dr. Simona Perotto, Dr. Carlo D'Angelo.

Many thanks to P. Ruffo and G. Scrofani for having supported our research program. I wish also to thank Laura Barbareschi and Gabriele Villa for their linguistic help and advices.

Contents

1	Introduction	1
2	An overview on basin geology	5
2.1	An insight into sedimentary basins	5
2.2	Sedimentary basin geology	6
2.3	Motivations for numerical simulations	9
2.4	Uncertainty	9
3	Mathematical models for basin simulation and compaction	11
3.1	Compaction models	12
3.1.1	The geometry of the basin	12
3.1.2	Volume averaging theorems	13
3.1.3	Momentum equation average	16
3.1.4	Mass equation averaging	21
3.1.5	Averaged energy balance	22
3.1.6	Constitutive equations	23
3.1.7	The complete macroscopic models	25
3.1.8	Approximated models	27
3.2	Numerical solution for compaction models	29
3.2.1	Time discretization algorithm for (3.50)	29
3.2.2	Time discretization algorithm for (3.51)	30
3.2.3	Algorithm analysis	31
3.3	Numerical solution for a one-dimensional case	33
3.3.1	The one-dimensional case and numerical techniques	33
3.3.2	Numerical results	34
3.4	Nomenclature	39
4	Multi-fluid tracking	41
4.1	A brief review of the literature	41
4.2	The proposed method	42
4.3	Analysis of the method	48

4.4	Convergence analysis of the one dimensional case	52
4.5	Numerical results	55
4.6	Results	55
4.7	Nomenclature	59
5	Numerical models for basic basin simulations	61
5.1	Physical and mathematical models	62
5.2	Time discretization	63
5.3	Velocity field solver	64
5.4	Tracking	71
5.5	Algorithm workflow	76
5.6	Numerical results	77
5.7	Nomenclature	84
6	Numerical Models for advanced basin simulations	87
6.1	Physical and mathematical models	89
6.1.1	Nomenclature	89
6.1.2	Rheological models	90
6.1.3	Modeling the compaction	91
6.1.4	The model	93
6.2	Numerical discretization	93
6.3	Adaptive grid movement	95
6.3.1	Grid movement equations	95
6.3.2	Error estimation	98
6.3.3	Metric definition	101
6.4	The handling of faults	103
6.4.1	A finite volume scheme	103
6.4.2	Set reconstruction (continuous part)	104
6.4.3	Set reconstruction (discrete part)	106
6.4.4	The reconstruction algorithm applied to a simple case	107
6.4.5	Local grid refinement	108
6.5	Algorithm workflow	109
6.6	Numerical results	110
7	Conclusions	117
	References	119

Chapter 1

Introduction

The main object of this thesis is to provide a comprehensive numerical tool for the three-dimensional simulation of sedimentary basins [94]. Sedimentary basins, in particular salt basins, are the best places to find oil, natural gas and to store dangerous nuclear waste material. The low permeability of salt guarantees low water leakage which is the main concern for the safety of a nuclear waste storage. For this reason one of the best places for a nuclear waste depository is a salt mine. These two applications call for a thorough knowledge of the basin evolution on geological time scales. Until now sedimentary basin studies have been based mainly on geological interpretation: experienced specialists estimate the history of a basin on the basis of common knowledge. More often, they provide a list of possible scenarios. An appropriate numerical simulator can provide the right tool to choose, among these scenarios, the correct one or, at least, the most realistic.

A starting point of this work has been a previous activity on two dimensional basin modeling and simulation discussed in [57, 56]. However the numerical techniques adopted there were not suited for the more complex 3D situation. In fact a complete reformulation has been required. The first part of our thesis is devoted to reviewing the physical models currently available in literature and to providing a common framework to the theory of the geological evolution of a sedimentary basin. Here we consider two kinds of physical models, the compaction models and the basin scale models. The former are physical models that represent the basin as a mechanically interacting two phase mixture: the fluid and the solid part are usually taken into account. The latter are reduced versions of the compaction models where all the quantities related to the fluid phase are modeled with some lumped parameter relations. The compaction models are used to predict overpressure conditions and, more generally, to simulate pressure and temperature fields inside the basin, see for instance [7]. Indeed, the safety of an oil drill is highly dependent on precise data regarding pressure and temperature.

In our work we extend the volume averaging method [12, 102] to a model of two moving-phases. This approach allows to obtain more information than other compaction models. In particular, a procedure to obtain the permeability and the thermal conductivity tensors from the microscopic structure is outlined. Moreover, the vol-

ume averaging method provides automatically the compaction model, that is the set of equations that describe the rate of loss of porosity as the layers are buried. Finally, we derive a couple of physical models and analyze their relations to the compaction and basin models found in literature. The compaction models consist of a set of evolution equations and a coupled Stokes-Darcy system. A preliminary analysis of the well posedness of this pde has been carried out.

Afterwards, we have considered the numerical discretization of the problem. Since the underlying problem is very complex, we have divided this step into three parts. In the first part we tackle the multi-fluid tracking problem. Since the rock rheology on geological timescales is akin to that of a viscous fluid, we have to solve a stratified fluid problem (see [57, 56]). To be more precise, we deal with a creeping flow in a Stokes regime. In the second part we combine the tracking algorithm with an iterative preconditioned Stokes solver [32, 73]. In the third part we add some more distinctive features such as the fault tracking, the movement of the basin boundary and other physical features, such as a simplified compaction relation, which makes our models suitable for geological simulation. Let's now review in detail these three parts.

In the first part (Chapter 4) we illustrate a method to track separated interfaces among immiscible fluids when a large number of fluids is involved. Our aim is to construct a robust method, effective even when the interfaces experience a strong deformation, with good mass conservation properties and that can be used on (2D and 3D) unstructured meshes. In literature many techniques regarding the two fluid problem are reported but they often cannot be extended readily to the multi-fluid problem and they are computationally expensive or they lack conservative properties. A few works are devoted to the multi-fluid simulation (see, for instance, [109, 110]) however many of them are specifically designed for the curvature-driven flows and are not readily applicable to the advection-driven case. While others involve some level set topological reconstruction steps that are computationally costly.

In this work we present an original coupled volume tracking - level set [65] method which gives a good compromise between accuracy and computational cost and has good conservation properties. Moreover, we introduce a flux-limited MUSCL scheme (see [10, 50]) which is capable to work on unstructured meshes and has a low numerical diffusion.

In the second part (Chapter 5) the tracking algorithm, combined with a Stokes solver, provides a solution method for a stratified fluid problem. In fact, actually, we use a classical operator splitting approach that divides the computation of pressure and velocity fields from the interface tracking. For three dimensional problems, the number of degrees of freedom (DOF) required for an accurate discretization of the Stokes problem is very high and the adoption of iterative schemes is necessary. In the present application, in addition, the value of the viscosity of the sediments may vary over a range of about five orders of magnitude (as shown in [56]), and this leads to a badly conditioned algebraic problem. Thus, we introduce a simple but effective preconditioner able to reduce sensibly the number of iterations required to solve the Stokes system in presence of high viscosity jumps. Numerical tests show that this preconditioner is efficient, although an optimality proof is still missing.

Moreover, in order to improve the efficiency of the code, we present an enhanced implementation of the tracking algorithm.

In the third part (Chapter 6) we introduce the implementation of non-Newtonian rheological relations, the compaction effects, the basement movement and the presence of faults. Basically, we have included the physical aspects illustrated in [57] into our three-dimensional framework. At the same time, we have developed some numerical techniques that overcome the limit of the fully Lagrangian approach used in [56].

A challenging aspect of this part of the work has been the implementation of the movement of the basin basement and of the lateral contour. The external boundary is subject to displacements as the surrounding soil moves with the Earth plates. This effect has a paramount importance in basin evolution as it is one of the driving forces for fault formation. 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 the boundary movement and an Eulerian scheme for the internal layers. This mixture between the Lagrangian and the Eulerian approaches leads to the so called Arbitrary Lagrangian Eulerian method (ALE), where the displacement of the grid is prescribed only on the boundary, while for the internal nodes a suitable movement law is considered, for example, to minimize the mesh distortion. The definition of the numerical algorithm for the computation of the internal grid movement is a critical part: we want, at the same time, to adapt the grid size, where necessary, without losing its quality. To achieve this goal, we choose the so called r -adaptivity (see [5]) combined with the ALE scheme. This technique is a cheaper choice in terms of computational costs than the h -adaptivity, although it may be less effective. We exploit then the information given by a residual-type error estimator [97] to construct an error-dependent metric, which drives the ALE scheme in adapting the grid size according to the minimization of the estimated error.

This work provides a comprehensive multi-physics numerical tool for the basin-scale simulation. The code has been validated in some realistic cases.

Even if it is not a specific subject of this thesis we mention the importance of a parallel implementation for a software of such a complexity. So far only a preliminary work on the scalability features of the linear system has been carried out. The work on this field is still ongoing.

Chapter 2

An overview on basin geology

2.1 An insight into sedimentary basins

In this thesis we will focus on the geological evolution of the thin layer of the Earth crust that composes the sedimentary basins. The Earth interior is made up of different layers. The most interior part is the core, see Figure 2.1(b): it is split into an inner solid core and an outer liquid part. The mantle forms a buffer zone between the core and the surface crust which is the outer and solid layer. The crust is less dense than the mantle and floats over it. The interaction between the mantle and the crust causes the well known continental plate drift and many geological phenomena such as the orogenesis and the subsidence. These two aspects are related to the uplift

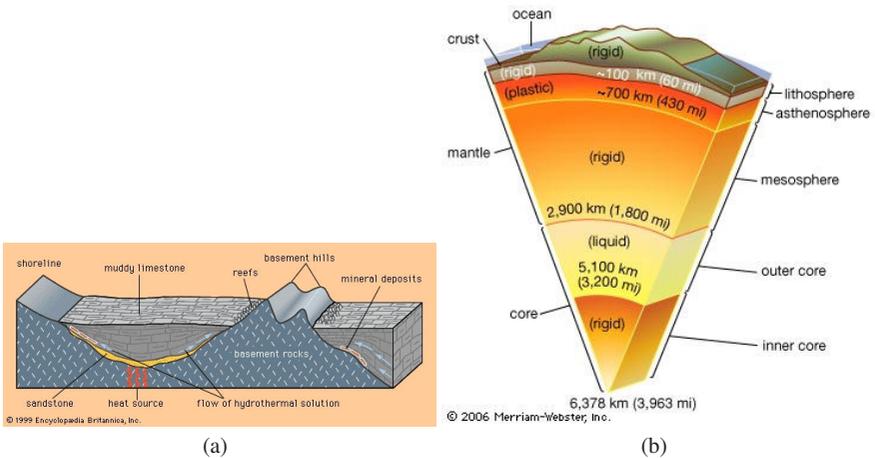


Fig. 2.1: (a) A schematic view of a sedimentary basin. (b) The earth interior.

and burial of parts of the Earth crust.

The term sedimentary basin is used to refer to any geographical feature exhibiting subsidence and consequent infilling by sedimentation. In Figure 2.1(a) we have outlined some of its geometrical features. The sedimentary basins lie on the bottom of depressed areas where the erosion processes deposit various kinds of materials. Typically the type of sediments deposited varies quite suddenly in a geological time scale framework. This forms some well defined sedimentary layers and the mixing between the different materials is very limited. The separating interfaces between the layers are called the horizons.

The sedimentary basin has, usually, three kinds of boundaries: the basement, the surface and the lateral contour which conventionally limit the area of interest. The basement is a solid compacted layer and can be either continental or oceanic (many sedimentary basins develop under the surface of the oceans). It is the bed over which the other sediments (the overburden) lie and it has much stronger mechanical characteristics than the overburden. The surface is the place we are more accustomed to and is the upper part of the basin. The lateral contour is usually an arbitrary boundary which delimits the area of interest and in many cases it has no physical meaning. The size of typical basins is of the order of 100 by 100 *km* in the horizontal plane and 10 *km* in depth.

2.2 Sedimentary basin geology

The sedimentary basins are stockpiles of gravel, sand, rocks and biological remains that have been transported by the wind, the rivers and, sometimes, by the sea. A typical example of a sedimentary basin is an alluvial plain where the erosion of the surrounding mountain ranges provides much of the deposited sediments. The physical characteristics of the layers change with the geological eras; every era lasting tens of millions of years. The deposition of different kinds of sediments causes the typical stratified configuration that is clearly visible in Figure 2.2.

Sedimentary basins are not static as it would seem at first sight. During the geological time scales they can experience strong deformations and even topological changes of the geometry of the layers. One of the main driving forces is the sedimentation that is the continuous deposition of debris. Nevertheless there are many forces applied from the inside and outside of the basin. The movement of the basement and of the lateral edges has a great impact on the morphology of the basin. From a tectonic viewpoint the basin is located on the upper part of the crust. The movements of the crust and the continental drift thus affect the movement of the boundary of the basin. This evolution of the basin can, in its turn, modify the local evolution of the crust: in fact the steady deposition of sediments causes the sinking of the basement by several kilometers in tens of millions of years. This phenomenon is also known as flexural lithostasis.

Many internal phenomena trigger the evolution of the basin, in particular the buoyancy of the lighter layers on denser ones. For example, the rocksalt is one of the lightest components, in fact it is one of the primary internal driving forces of the

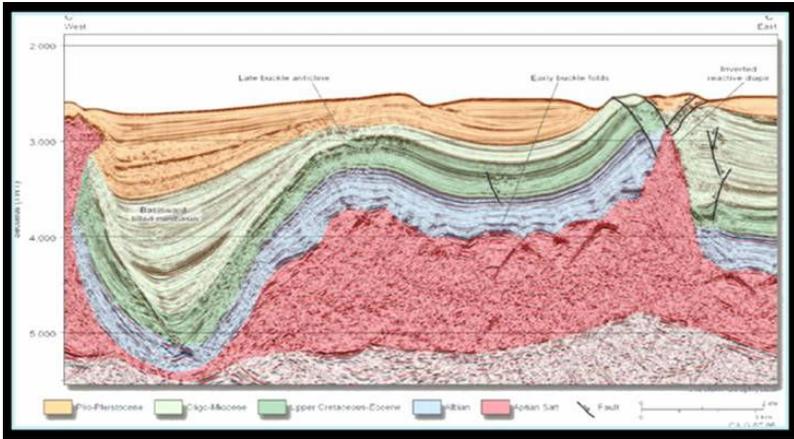


Fig. 2.2: A seismic analysis of a sedimentary basin. Seismic data can provide a detailed description of a basin cross section.

sedimentary basin dynamics. To be more precise at a depth of few hundred meters the other sediments, having expelled most of their water content, become heavier than salt. Also other components such as lightweight shale can be affected by buoyancy effects. In other words the sediments and rocks behave, on geological time scales, as viscous fluids, whose main physical characteristics are outlined in table (2.1).

The fluid behavior of the rocks can be explained at least by two physical arguments.

Sediment	Density (Kg/m^3)	Viscosity ($Pa \cdot s$)
Shale	2300	10^{21}
Undercompacted Shale	2200	10^{20}
Sandstone	2400	10^{21}
Limestone	2500	10^{22}
Rocksalt	1800	10^{19}
Rock	2500	10^{21}

Table 2.1: Typical density and viscosity values for some sediments.

The first one deals with the crystalline structure of the rocks and the movement of voids and imperfections. The rock structure contains many voids in the crystalline lattice, their position changes on geological time scales and their distribution can be statistically determined. Without applied loads, the movement of the voids is equally probable in all the directions. But if a load is applied, their movement becomes more probable in some specific directions; for more details see [94]. The macroscopic net effect is a fluid behavior associated to a Newtonian rheology. The second theory is based on the solubility of rock components in water. It is well known that pressure

affects the solubility of solids in water, therefore where a load is applied, the solubility is greater. In the contact regions among grains the rock dissolves, and then deposits on unstressed areas. Once again this phenomenon is represented macroscopically by a shear flow of a Newtonian fluid, see [94]. Nevertheless not all the effects can be explained by viscous fluid models. Also plasticity plays a key role. In fact rocks, under lithostatic pressure (i.e. the pressure applied by the overburden), tend to fracture and to modify their reciprocal position. These effects are usually modeled with a plastic-type rheological law; see [79]. Also the fault formation can be regarded as a plastic effect. In the faulted regions the soil is highly damaged and it cannot sustain the applied stress. All the three aspects just described are active in a sedimentary basin and none of them alone can fully explain the rheological behavior of the sediments. So far, a comprehensive model that links the stress to the strain and the strain to the velocity is missing, therefore semi-empirical relations are widely used; see [94], [55], [79], [95].

Let's now examine another important phenomenon: the compaction. Superficial sediments have up to 50% of void space which is filled by air or by water (sea water if the basin is in correspondance with the sea bed). As the layers are progressively buried by the accumulation of sediments, the overburden pressure rises. This triggers the reduction of the pore spaces and the liquid phase is expelled from the porous media. In fact, below few hundred meters the pore spaces are saturated by water. In some cases the water can be trapped by impermeable traps, in that case the fluid pressure rises as part of the overburden is supported by the liquid phase. This case is also known as overpressure and it has many consequences on the safety of oil-field exploitation. Luckily, strong overpressures are rare, in most of the cases the rock supports all the lithostatic weight though maintaining a porosity of 5 – 10%. The chemical reactions are another key element, in fact they modify the composition of rocks and create mechanical links between grains: cementification is an example. Chemistry is involved in the diagenesis that is the formation, starting from separated grains, of a coherent rock layer. The oldest sediment layers are made of compacted rock with much stronger mechanical characteristics than the shallow undercompacted sediments. Another important chain of chemical reactions is the formation of natural oil and gas. The last important feature we may consider is the temperature distribution inside the basin. The thermal gradient is about thirty degrees per kilometer, therefore the basement could reach three hundred degrees Celsius. The main effect is the heat diffusion from the lower layers and from the upper mantle, but also the transport of heat carried by the water is an important aspect. The temperature greatly affects the chemical reactions and the rheology, should the solid part reach the melting point. Lastly, salt is a good heat conductor and this has many implications in the oil formation.

2.3 Motivations for numerical simulations

Sedimentary basins, in particular salt basins, are among the best places to find petroleum, natural gas and to store nuclear waste material. In fact the low permeability of salt guarantees low water leakages that are the main concern for the safety of nuclear waste storage. For this reason one of the best places for a nuclear waste depository is a salt mine. Precise data regarding the basin evolution on geological timescales are required to solve the problems emphasized by these two applications. The history of the basin has a deep impact on the characteristics of the oil generated: in particular the geometrical evolution and the temperature experienced by the sediments determine localization, quantity and quality of the oil. For example temperature is a key aspect that controls the petroleum - gas ratio; the latter being less valuable than the former. Another example is the geometry of the cap-Rock (the sealing layer that triggers the formation of oilfields), which is the sealing layer of the oilfield. Oil usually floats and collects near the cap-Rock. In other terms, to have detailed information about an oilfield, we must have information about the past history of it.

Till now sedimentary basin studies have been based on the geological interpretation of experienced specialists. Geologists, usually, can outline several evolution scenarios of the basin. Therefore we must choose among them the ones which are coherent from a physical viewpoint. Numerical simulation could provide the tool for choosing the right scenario. Moreover, it can provide quantitative information (for instance the stress field) which are difficult to estimate by other means.

The great interest in numerical tools is boosted by the technical difficulties to carry out analogical experiments. Actually it is difficult to scale correctly all the physical quantities in a relative small model. Sand-box experiments [88] provide useful information regarding the brittle behavior of grains but can not represent all the viscous creeping mechanisms which require millions of years to produce a measurable effect. The experiments devoted to investigate the sediment rheology are difficult to carry out too. As a matter of fact it is necessary, at the same time, to reach extreme values of pressure and to measure very small displacements.

Also the nuclear industry is interested in structural geology simulations. Here the main interest is oriented to simulate future evolution of the basins and, in particular, the stability of a deposit in a time frame comparable to the half life of the isotopes to be stored.

2.4 Uncertainty

The mechanics of a sedimentary basin is very complex and depends on parameters which are very difficult, if not impossible, to determine with precision. The main uncertainty source is the rheology of the sediments, in fact, though the fluid behavior of the rocks is a well known phenomenon, the precise measure of the viscosity coefficients is a challenging aspect. From many points of view the Geology is sim-

ilar to Cosmology. In both cases direct measures are not available and there are only indirect data. For instance we cannot know the past deformation of a sedimentary basin and neither can we observe its evolution on geological time scales. The only available data is the configuration of present layers obtained with seismic data, the pore pressure and the temperature in a few points that correspond to the wells. Some other data can include the presence of fossil materia in some layers. The organic matter can be dated by measuring the quantity of the radioactive isotopes of the carbon. This technique gives an insight into the timing of the deposition of the layers. Finally also the presence of some chemical compounds can give some information on the temperature history. Actually some reactions are possible only if the temperature is higher than a prescribed threshold.

A complete validation of our computational results is beyond the scope of this thesis. We have concentrated in the development of proper numerical techniques to make the numerical simulation of a real 3D basin a reality. Therefore we have included the best and more promising physical models that are used now in literature. However some preliminar tests have been carried out and are displayed in Chapter 6.

Chapter 3

Mathematical models for basin simulation and compaction

In this chapter we introduce the numerical modeling of sedimentary basins, in particular, we discuss the most popular models and their applicability. We introduce the compaction models and the simpler basin scale models. Regarding the compaction models we provide a class of numerical algorithms and we discuss their implementation with a one dimensional model.

In the past years the numerical simulation of the geological evolution of sedimentary basins has gained an increasing relevance in the study of the safety of nuclear waste disposal, in the exploitation of geothermal energy and in oilfield studies.

In this Chapter we illustrate and analyze mathematical models for the simulation of the geological evolution of a sedimentary basin. These models include the evolution of the solid part (treated in [24, 107, 57, 63]) and the simulation of the fluid part (treated in [28, 84, 100]). However, for the simulations at the basin scale we will see that a detailed description for the fluid is not necessary and, in many cases, simplified models can be used.

The simulation of both the solid and fluid part is required, for instance, to have accurate results regarding the heat flow inside the basin and the temperature distribution. The temperature is a key aspect for petroleum formation since the chemical reactions that transform the biological remains into oil are strongly dependent on temperature. A few papers treat both the structural and fluid evolution of a basin with the same accuracy. An example is [7, 98, 99, 54] where linear elastic, and elasto-plastic models for the solid matrix are considered. Many works are devoted mainly to the fluid evolution: in [27] a viscoelastic matrix is examined though the total pressure of the mixture is considered lithostatic and the deviatoric components of the stress tensor of the solid part are neglected. A similar approach is used in [100, 101] and, through some manipulations, the solid matrix displacement and stress are eliminated from the model. In [36, 89] a one dimensional case is considered, in this situation structure of the equations is greatly simplified and, once again, the shear stress of the solid part is neglected. In [83] a complete three dimensional model for the fluid part is considered while a vertical balance equation is employed for the solid matrix. Many other works are mainly devoted to the structural evolution only, see, for instance, [108, 24, 37]. In particular, some works are specifically devoted to the salt

tectonics [55, 79, 107] while some other works simulate the visco-elastic features of the upper crust [60, 63].

The two phase and one phase models are also known as compaction models and basin scale models respectively. The former are physical models which describe the basin as a mechanically interacting two phase mixture: the fluid and the solid part are usually taken into account. The latter are reduced versions of the compaction models where the fluid phase is modeled with some algebraic relations. The compaction models are used to predict overpressure conditions and, more in general, to estimate pressure and temperature fields inside the basin; see for instance [16]. The safety of the oil drill is highly dependent on precise data regarding the presence of overpressures.

The physical model we illustrate in this Chapter is accurate both for the fluid and solid phases. This model is not completely new, it resembles very closely those in [7, 14, 12, 13, 75, 96, 99], yet it differs in many details. In fact, we extend the volume averaging method of [102] to a two moving-phase model: this approach allows to extract some more information. To be more precise, we describe a procedure to obtain the permeability and the thermal conductivity tensors from the microscopic structure. Moreover the volume averaging method here described provides a compaction model directly, which is the set of equations that describes the rate of loss of porosity as the layers are buried. Finally, we derive a couple of physical models and we analyze their relations with respect to other compaction and basin models found in the literature.

3.1 Compaction models

3.1.1 The geometry of the basin

In Figure 3.1 the geometry of a representative basin is depicted. We denote by Ω the volume of the basin and with $\partial\Omega$ its boundary. The latter is divided into three parts: the basement Γ_B , the upper surface Γ_S , and the lateral sides Γ_L . We consider n_s layers inside Ω , denoted with Ω_i . Each Ω_i is an open measurable subdomain of Ω such that $\Omega_i \cap \Omega_j = \emptyset$ if $i \neq j$ and $\cup_{i=1}^{n_s} \overline{\Omega}_i = \overline{\Omega}$. Every sedimentary layer is composed of a solid part and an interstitial space completely saturated by fluids: in our case we consider only water as the quantities of other species are often negligible. The solid part of every Ω_i is denoted with Ω_i^s and the fluid part with Ω_i^f , for simplicity we use Ω_i^k with $k \in \{s, f\}$. In general, for an arbitrary variable, or constant, a , we use the notation a^k with $k \in \{s, f\}$. The solid part of Ω is defined as Ω^s with $\overline{\Omega}^s = \cup_{i=1}^{n_s} \overline{\Omega}_i^s$ and the fluid part as Ω^f with $\overline{\Omega}^f = \cup_{i=1}^{n_s} \overline{\Omega}_i^f$. Each sedimentary layer is associated with some constant physical properties such as the reference density ρ_i^s and the apparent viscosity μ_i^s . Actually the variation of the physical coefficients inside each layer can be neglected while they can change of orders of magnitude between the layers. In the same way we assign a density ρ^f and a viscosity μ^f to the fluid

phase, which can be considered constant in the whole domain. Let's now attend

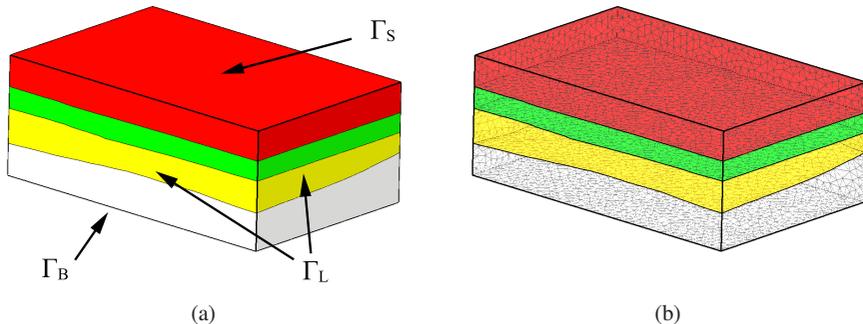


Fig. 3.1: a) A typical external shape of the domain Ω . The external boundary is divided into three parts: the basement Γ_B , the upper surface Γ_S and the lateral surface Γ_L . b) An open three dimensional view of the sedimentary basin which contains three horizons and four layers.

to the estimation of the characteristic lengths of the basin: the solid part consists of grains with dimensions ranging from millimeters to a few meters. Usually the gravel dimensions are a few centimeters. The characteristic scale of a basin is about ten kilometers, therefore the characteristic scale of the grains is well separated from that of the basin and it is possible to extract a volume $\omega \in \Omega$ that is "infinitesimal" from a macroscopic viewpoint yet it contains enough grains to get a meaningful spatial averaging. We can distinguish a solid part ω^s and a fluid part ω^f inside ω . We define $\varepsilon^k = \frac{|\omega^k|}{|\omega|}$ as the volume fraction of the k -th species; in particular ε^f is also known as the porosity and often indicated with ϕ . As a consequence, we get $\varepsilon^s = 1 - \phi$. We can also define the volume fraction of the i -th solid species present in ω^s as $\lambda_i = \frac{|\Omega_i \cap \omega^s|}{|\omega^s|}$. Finally we denote with \vec{n} the outgoing normal of Ω , with \vec{n}^k the outgoing normal of Ω^k , with \vec{X} the position vector, with \hat{x}_i , $i = 1, 2, 3$ the three cartesian basis vectors oriented along the two horizontal axes and the vertical axis respectively, and with $t \in [0, t^{end}]$ the time being $[0, t^{end}]$ the simulation interval. We adopt the notation of indexing the components of a generic vector \vec{V} as V_i with $i = 1, 2, 3$.

3.1.2 Volume averaging theorems

We start by recalling some fundamental results and definitions of the volume average theory developed by S. Whitaker in [102]. As customary in this case, for the derivation of the averaged equation we will neglect the fact that our actual domain is

bounded, so that we can ignore, at this stage, the presence of the boundary $\partial\Omega$. At each point \vec{X} we indicate with $\omega(\vec{X})$ a measurable averaging volume centered in \vec{X} such that $\omega(\vec{X}) = \{\vec{Z} : \vec{Z} = \vec{X} + \vec{Y}, \vec{Y} \in B\}$, being B a ball centered in zero with a radius sufficiently large to be used for averaging out microscale features, yet small at a large scale. In this context \vec{Y} can be considered a microscale coordinate. Then we define, for an arbitrary variable a^k related to the k -th phase (it could be a scalar, a vector or a tensor), using the notation of [102], the superficial average on $\omega(\vec{X})$ as

$$\langle a^k \rangle(\vec{X}) = \frac{1}{|\omega(\vec{X})|} \int_{\omega(\vec{X})} a^k(\vec{X} + \vec{Y}) d\vec{Y}. \quad (3.1)$$

where the variable a^k is extended outside Ω_k by zero. The definition (3.1) indicates that the average value $\langle a^k \rangle$ is associated with the centroid of $\omega(\vec{X})$. We can also define the intrinsic average as

$$[a^k](\vec{X}) = \frac{1}{|\omega^k(\vec{X})|} \int_{\omega^k(\vec{X})} a^k(\vec{X} + \vec{Y}) d\vec{Y}.$$

Clearly the domain of $\langle a^k \rangle$ and $[a^k]$ is the whole Ω . Since $\varepsilon^k = \frac{|\omega^k|}{|\omega|}$ the superficial and the intrinsic average are linked by the following relation

$$\langle a^k \rangle = \varepsilon^k [a^k]. \quad (3.2)$$

We can now introduce the volume averaging theorem for a scalar field a^k (these results can be extended readily to vector and tensor fields):

Proposition 3.1. *If $a^k \in L^2(\Omega) \cap H^1(\Omega^k)$, $k = s, f$, then the superficial average satisfies the following relation in a distributional sense*

$$\langle \vec{\nabla} a^k \rangle = \vec{\nabla} \langle a^k \rangle + \frac{1}{|\omega|} \oint_{\partial\omega^k} a^k \vec{n}. \quad (3.3)$$

Proof. Using the distributional definition of the gradient we have

$$\int_{\omega} \vec{\nabla} a^k \cdot \vec{v} = - \int_{\omega} a^k (\vec{\nabla} \cdot \vec{v}) + \oint_{\partial\omega} a^k (\vec{v} \cdot \vec{n}) \quad \forall \vec{v} \in \mathbb{C}_0^\infty(\overline{\Omega}). \quad (3.4)$$

Since a^k is null outside ω^k we get $\int_{\omega} a^k (\vec{\nabla} \cdot \vec{v}) = \int_{\omega^k} a^k (\vec{\nabla} \cdot \vec{v})$ and integrating by parts the first term of the right hand side of (3.4) we obtain

$$\int_{\omega} \vec{\nabla} a^k \cdot \vec{v} = \int_{\omega^k} \vec{\nabla} a^k \cdot \vec{v} - \oint_{\gamma^k} a^k (\vec{v} \cdot \vec{n}) + \oint_{\partial\omega} a^k (\vec{v} \cdot \vec{n}) \quad \forall \vec{v} \in \mathbb{C}_0^\infty(\overline{\Omega})$$

where γ^k is defined in Figure 3.2. Switching the gradient and the integral in the first term of the right hand side and since $-\oint_{\gamma^k} a^k (\vec{v} \cdot \vec{n}) + \oint_{\partial\omega} a^k (\vec{v} \cdot \vec{n}) = \oint_{\partial\omega^k} a^k (\vec{v} \cdot \vec{n})$ we have

$$\int_{\omega} \vec{\nabla} a^k \cdot \vec{v} = \vec{\nabla} \cdot \int_{\omega^k} a^k \vec{v} + \oint_{\partial\omega^k} a^k (\vec{v} \cdot \vec{n}) \quad \forall \vec{v} \in \mathbb{C}_0^\infty(\overline{\Omega}). \quad (3.5)$$

Dividing by $|\omega|$ we get the proof.

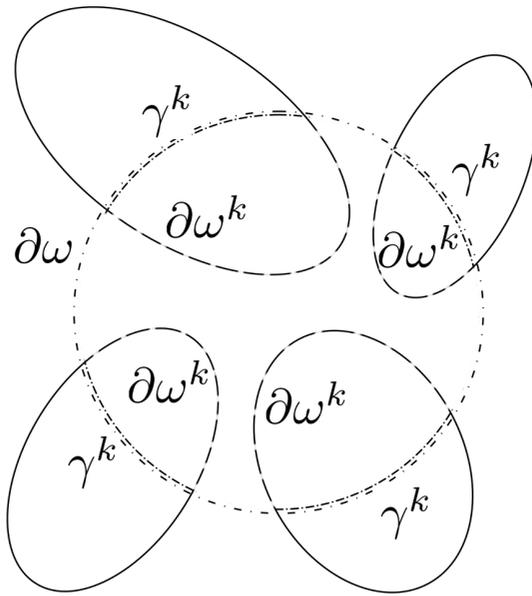


Fig. 3.2: An example of the averaging volume. Four grains are depicted and the boundary $\partial\omega$ of a circular averaging volume (dash-dot line) is outlined. The portion of the boundary of the grains inside the averaging volume (dashed line) is denoted by $\partial\omega^k$. Finally γ^k is the portion of $\partial\omega$ inside the grains (double dash dot line).

Proposition 3.1 has an important consequence:

Proposition 3.2. *The following relation holds*

$$\frac{1}{|\omega|} \oint_{\partial\omega^k} \vec{n} = -\vec{\nabla} \varepsilon^k.$$

Proof. Substituting $a^k = 1$ in (3.4) we get

$$\int_{\omega} \vec{\nabla} a^k \cdot \vec{v} = - \int_{\omega} \vec{\nabla} \cdot \vec{v} + \oint_{\partial\omega} (\vec{v} \cdot \vec{n}) = 0 \quad \forall \vec{v} \in \mathbb{C}_0^\infty(\overline{\Omega}).$$

Combining with (3.5) we obtain

$$0 = \vec{\nabla} \cdot \int_{\omega^k} \vec{v} + \oint_{\partial\omega^k} (\vec{v} \cdot \vec{n}) \quad \forall \vec{v} \in \mathbb{C}_0^\infty(\overline{\Omega}) \quad (3.6)$$

If we choose $\vec{v} = 1$ then

$$0 = \vec{v} \cdot \left(\vec{\nabla} \int_{\omega^k} 1 + \oint_{\partial\omega^k} \vec{n} \right)$$

Dividing by $|\omega|$ and using the definition of ε^k we have

$$0 = \vec{v} \cdot \left(\vec{\nabla} \left(\varepsilon^k \frac{1}{|\omega|} \int_{\omega} 1 \right) + \frac{1}{|\omega|} \oint_{\partial\omega^k} \vec{n} \right) \quad \forall \vec{v}$$

Since $\frac{1}{|\omega|} \int_{\omega} 1 = 1$ we get the proof.

Using Proposition 3.2, if the macroscopic and microscopic length-scales are well separated, as $[a^k]$ can be considered a constant in ω , we can make the following approximation (see [102])

$$\frac{1}{|\omega|} \oint_{\partial\omega^k} [a^k] \vec{n} = -[a^k] \vec{\nabla} \varepsilon^k. \quad (3.7)$$

Finally, we define the variations with respect to the intrinsic average as

$$\delta a^k = a^k - [a^k]. \quad (3.8)$$

3.1.3 Momentum equation average

Here we assume a creeping flow for both the fluid and the rock matrix and completely disregard the inertial terms. This assumption is reasonable given the time scales of the the basin evolution, see [56]. The momentum balance equations expressed in a Eulerian framework at a microscale level are

$$\begin{cases} \vec{\nabla} \cdot \vec{u}^f = 0 & \text{in } \Omega^f \\ \vec{\nabla} \cdot \vec{u}^s = 0 & \text{in } \Omega^s \\ \vec{\nabla} \cdot \bar{\bar{\sigma}}^f - \vec{\nabla} P^f + \rho^f \vec{g} = 0 & \text{in } \Omega^f \\ \vec{\nabla} \cdot \bar{\bar{\sigma}}^s - \vec{\nabla} P^s + \rho^s \vec{g} = 0 & \text{in } \Omega^s \\ \vec{u}^f \cdot \vec{n}^f + \vec{u}^s \cdot \vec{n}^s = 0 & \text{on } \partial\Omega^f \cap \partial\Omega^s \\ (\bar{\bar{\sigma}}^f - P^f \bar{\bar{I}}) \cdot \vec{n}^f + (\bar{\bar{\sigma}}^s - P^s \bar{\bar{I}}) \cdot \vec{n}^s = 0 & \text{on } \partial\Omega^f \cap \partial\Omega^s, \end{cases} \quad (3.9)$$

where \vec{u}^f, \vec{u}^s are the velocity fields, $\bar{\bar{\sigma}}^f, \bar{\bar{\sigma}}^s$ are the deviatoric parts of the stress tensors and P^f, P^s are the fluid and solid pressure, respectively (in particular P^f is also called the pore pressure). The gravity acceleration is \vec{g} , $\bar{\bar{I}}$ is the identity tensor and the density of the solid is defined by

$$\rho^s = \sum_{i=1}^{n_s} \lambda_i \rho_i^s. \quad (3.10)$$

Here we have not considered the boundary conditions. In fact they have no influence on the derivation of the averaged model and we will introduce them later. To deduce an averaged model for equations (3.9), we apply the superficial averaging operator to the first and the second equation and using equation (3.3) we get

$$\vec{\nabla} \cdot \langle \vec{u}^k \rangle + \frac{1}{|\omega|} \oint_{\partial\omega^k} \vec{u}^k \cdot \vec{n}^k = 0. \quad (3.11)$$

Using (3.2) and (3.8), we obtain

$$\vec{\nabla} \cdot (\varepsilon^k [\vec{u}^k]) + \frac{1}{|\omega|} \oint_{\partial\omega^k} ([\vec{u}^k] + \delta \vec{u}^k) \cdot \vec{n}^k = 0.$$

Lastly, by combining it with (3.7) and (3.11) we get

$$\vec{\nabla} \cdot [\vec{u}^k] + \frac{1}{|\omega^k|} \oint_{\partial\omega^k} \delta \vec{u}^k \cdot \vec{n}^k = 0. \quad (3.12)$$

We can operate similarly using the third and fourth equations of (3.9). Here we have the superficial average

$$\vec{\nabla} \cdot (\langle \bar{\sigma}^k \rangle - \langle P^k \rangle \bar{I}) + \frac{1}{|\omega^k|} \oint_{\partial\omega^k} (\bar{\sigma}^k - P^k \bar{I}) \cdot \vec{n}^k + \varepsilon^k \rho^k \vec{g} = 0 \quad (3.13)$$

and the corresponding intrinsic average

$$\vec{\nabla} \cdot ([\bar{\sigma}^k] - [P^k] \bar{I}) + \frac{1}{|\omega^k|} \oint_{\partial\omega^k} (\delta \bar{\sigma}^k - \delta P^k \bar{I}) \cdot \vec{n}^k + \rho^k \vec{g} = 0. \quad (3.14)$$

We replace equation (3.13) with $k = s$ with the sum of (3.13) with $k = s, f$

$$\vec{\nabla} \cdot (\langle \bar{\sigma}^s \rangle + \langle \bar{\sigma}^f \rangle) + \vec{\nabla} \cdot (\langle P^s \rangle + \langle P^f \rangle) + \rho^T \vec{g} = 0, \quad (3.15)$$

where $\rho^T = \phi \rho^f + (1 - \phi) \rho^s$. Equation (3.15) contrary to (3.13) does not contain any interface term and then it does not need a closure. Besides, the orders of magnitude in Table 3.1 suggest that the fluid stress tensor is negligible, indeed $\bar{\sigma}^f = O(\frac{1}{7} \mu^f) \simeq 10^{-16}$ while $\bar{\sigma}^s = O(\frac{1}{7} \mu^s) \simeq 10^7$. Using equations (3.11) and (3.15) we obtain the following macroscopic model

$$\begin{cases} \vec{\nabla} \cdot \langle \vec{u}^s \rangle = \Phi \\ \vec{\nabla} \cdot \langle \vec{u}^f \rangle = -\Phi \\ \vec{\nabla} \cdot \langle \bar{\sigma}^s \rangle - \vec{\nabla} \cdot \langle P^T \rangle + \rho^T \vec{g} = 0 \\ -\vec{\nabla} [P^f] + \rho^f \vec{g} + \frac{1}{|\omega^f|} \oint_{\partial\omega^f} (\delta \bar{\sigma}^f - \delta P^f \bar{I}) \cdot \vec{n}^f = 0, \end{cases} \quad (3.16)$$

Variable	Symbol	Magnitude
time	t	$10^{13} s$
space	\vec{X}	$10^3 m$
density	ρ^k	$10^3 kg/m^3$
fluid viscosity	μ^f	$10^{-3} Pa \cdot s$
solid viscosity	μ^s	$10^{20} Pa \cdot s$
pressure	P^k	$10^7 Pa$
temperature	T^k	$10^2 K$
thermal conductivity	H^k	$1W/(Km)$
thermal capacity	c^k	$10^3 J/(KgK)$

Table 3.1: Orders of magnitude of the main physical variables and constants.

where $\langle P^T \rangle = \langle P^s \rangle + \langle P^f \rangle$ is the total pressure and

$$\Phi = -\frac{1}{|\omega|} \oint_{\partial\omega^s} \vec{u}^s \cdot \vec{n}^s = \frac{1}{|\omega|} \oint_{\partial\omega^f} \vec{u}^f \cdot \vec{n}^f,$$

which, from now on, will be called the compaction function, as it provides the rate of decrease of porosity. We introduce, now, the fluid rheology that will be used for the model closure. We consider a Newtonian law,

$$\bar{\sigma}^f = \mu^f \left(\vec{\nabla} \vec{u}^f + (\vec{\nabla} \vec{u}^f)^T \right). \quad (3.17)$$

Whose corresponding averaged equations are

$$\langle \bar{\sigma}^f \rangle = \mu^f \left(\vec{\nabla} \langle \vec{u}^f \rangle + (\vec{\nabla} \langle \vec{u}^f \rangle)^T \right) + \frac{\mu^f}{|\omega|} \oint_{\partial\omega^f} \vec{u}^f \vec{n}^f + (\vec{u}^f \vec{n}^f)^T$$

and

$$[\bar{\sigma}^f] = \mu^f \left(\vec{\nabla} [\vec{u}^f] + (\vec{\nabla} [\vec{u}^f])^T \right) + \frac{\mu^f}{|\omega^f|} \oint_{\partial\omega^f} \delta \vec{u}^f \vec{n}^f + (\delta \vec{u}^f \vec{n}^f)^T. \quad (3.18)$$

respectively.

To obtain a closed form for (3.16) it is necessary to derive a differential problem for $\delta \vec{u}^k$, δp^k and $\delta \bar{\sigma}^k$. Subtracting (3.14) to the third and the fourth equation of (3.9), subtracting (3.18) to (3.17) and breaking up the interface conditions of (3.9) into their mean components and variations we get the following problem for $\delta \vec{u}^f$, $\delta \vec{u}^s$, δP^f , δP^s , $\delta \bar{\sigma}^f$, $\delta \bar{\sigma}^s$:

$$\left\{ \begin{array}{l}
\overline{\nabla} \cdot \delta \vec{u}^f = \frac{1}{|\omega^f|} \oint_{\partial \omega^f} \delta \vec{u}^f \cdot \vec{n}^f \text{ in } \Omega^f \\
\overline{\nabla} \cdot \delta \vec{u}^s = \frac{1}{|\omega^s|} \oint_{\partial \omega^s} \delta \vec{u}^s \cdot \vec{n}^s \text{ in } \Omega^s \\
\overline{\nabla} \cdot (\delta \bar{\sigma}^f - \delta P^f \bar{I}) = \frac{1}{|\omega^f|} \oint_{\partial \omega^f} (\delta \bar{\sigma}^f - \delta P^f \bar{I}) \cdot \vec{n}^f \text{ in } \Omega^f \\
\overline{\nabla} \cdot (\delta \bar{\sigma}^s - \delta P^s \bar{I}) = \frac{1}{|\omega^s|} \oint_{\partial \omega^s} (\delta \bar{\sigma}^s - \delta P^s \bar{I}) \cdot \vec{n}^s \text{ in } \Omega^s \\
\delta \bar{\sigma}^f = \mu^f (\overline{\nabla} \delta \vec{u}^f + (\overline{\nabla} \delta \vec{u}^f)^T) - \mu^f \frac{1}{|\omega^f|} \oint_{\partial \omega^f} \delta \vec{u}^f \vec{n}^f + (\delta \vec{u}^f \vec{n}^f)^T \text{ in } \Omega^f \\
\delta \vec{u}^f - \delta \vec{u}^s = -[\vec{u}^r] \text{ on } \partial \Omega^f \cap \partial \Omega^s \\
(\delta \bar{\sigma}^f - \delta P^f \bar{I}) \cdot \vec{n}^f + (\delta \bar{\sigma}^s - \delta P^s \bar{I}) \cdot \vec{n}^s = \\
-([\bar{\sigma}^f] - [P^f] \bar{I}) \cdot \vec{n}^f + ([\bar{\sigma}^s] - [P^s] \bar{I}) \cdot \vec{n}^s \text{ on } \partial \Omega^f \cap \partial \Omega^s \\
[\delta \vec{u}^f - \delta \vec{u}^s] = 0 \text{ in } \Omega,
\end{array} \right. \quad (3.19)$$

where $[\vec{u}^r] = [\vec{u}^f] - [\vec{u}^s]$ is the relative speed. The last equation of (3.19) state that the averaged variations are null. It has been introduced by Whitaker in [102] to obtain the hydraulic permeability tensor \bar{K} . From this system it is possible to extract a closure for the fluid balance equation.

In the literature many closure models have already been proposed for system (3.21), such as the closure model of [102], so we postulate that

$$\delta \vec{u}^f = -\delta \vec{u}^s = \frac{1}{2} \bar{B} \cdot [\vec{u}^r], \quad \delta P^f = \mu^f \vec{b} \cdot [\vec{u}^r], \quad (3.20)$$

where $\bar{B} : \Omega \rightarrow \mathbb{R}^3 \times \mathbb{R}^3$ and $\vec{b} : \Omega \rightarrow \mathbb{R}^3$. Moreover problem (3.19) is solved in the unit cell ω using periodic boundary conditions:

$$\left\{ \begin{array}{l}
\overline{\nabla} \cdot \delta \vec{u}^f = \frac{1}{|\omega^f|} \oint_{\partial \omega^f} \delta \vec{u}^f \cdot \vec{n}^f \text{ in } \omega^f \\
\overline{\nabla} \cdot \delta \vec{u}^s = \frac{1}{|\omega^s|} \oint_{\partial \omega^s} \delta \vec{u}^s \cdot \vec{n}^s \text{ in } \omega^s \\
\overline{\nabla} \cdot (\delta \bar{\sigma}^f - \delta P^f \bar{I}) = \frac{1}{|\omega^f|} \oint_{\partial \omega^f} (\delta \bar{\sigma}^f - \delta P^f \bar{I}) \cdot \vec{n}^f \text{ in } \omega^f \\
\overline{\nabla} \cdot (\delta \bar{\sigma}^s - \delta P^s \bar{I}) = \frac{1}{|\omega^s|} \oint_{\partial \omega^s} (\delta \bar{\sigma}^s - \delta P^s \bar{I}) \cdot \vec{n}^s \text{ in } \omega^s \\
\delta \bar{\sigma}^f = \mu^f (\overline{\nabla} \delta \vec{u}^f + (\overline{\nabla} \delta \vec{u}^f)^T) - \mu^f \frac{1}{|\omega^f|} \oint_{\partial \omega^f} \delta \vec{u}^f \vec{n}^f + (\delta \vec{u}^f \vec{n}^f)^T \text{ in } \omega^f \\
\delta \vec{u}^f - \delta \vec{u}^s = -[\vec{u}^r] \text{ on } \partial \omega^f \cap \partial \omega^s \\
(\delta \bar{\sigma}^f - \delta P^f \bar{I}) \cdot \vec{n}^f + (\delta \bar{\sigma}^s - \delta P^s \bar{I}) \cdot \vec{n}^s = \\
-([\bar{\sigma}^f] - [P^f] \bar{I}) \cdot \vec{n}^f + ([\bar{\sigma}^s] - [P^s] \bar{I}) \cdot \vec{n}^s \text{ on } \partial \omega^f \cap \partial \omega^s \\
\delta \vec{u}^k(\vec{X} + \mathcal{C}_i \hat{x}_i) = \delta \vec{u}^k(\vec{X}) \text{ on } \partial \omega \text{ with } i = 1, 2, 3; k \in \{s, f\} \\
\delta P^k(\vec{X} + \mathcal{C}_i \hat{x}_i) = \delta P^k(\vec{X}) \text{ on } \partial \omega \text{ with } i = 1, 2, 3; k \in \{s, f\} \\
[\delta \vec{u}^f - \delta \vec{u}^s] = 0 \text{ in } \omega,
\end{array} \right. \quad (3.21)$$

where \mathcal{C}_i with $i = 1, 2, 3$ is the periodicity length in the three directions. Inserting equations (3.20) in the first boundary condition of (3.21) we get

$$\bar{B} = -\bar{I} \text{ on } \partial \omega^f \cap \partial \omega^s, \quad (3.22)$$

Subtracting the first two equations in (3.19) and inserting (3.20) we obtain

$$(\vec{\nabla} \cdot \vec{B}) \cdot [\vec{u}^r] = \left(\frac{1}{2|\omega^f|} \oint_{\partial\omega^f} \vec{B} \cdot \vec{n}^f + \frac{1}{2|\omega^s|} \oint_{\partial\omega^s} \vec{B} \cdot \vec{n}^s \right) \cdot [\vec{u}^r]. \quad (3.23)$$

Using equation (3.22), and exploiting the fact that we are using a spatially periodic model (see [102]), we obtain

$$\begin{aligned} \left(\frac{1}{2|\omega^f|} \oint_{\partial\omega^f} \vec{B} \cdot \vec{n}^f + \frac{1}{2|\omega^s|} \oint_{\partial\omega^s} \vec{B} \cdot \vec{n}^s \right) \cdot [\vec{u}^r] = \\ - \left(\frac{1}{2|\omega^f|} \oint_{\partial\omega^f} \vec{n}^f + \frac{1}{2|\omega^s|} \oint_{\partial\omega^s} \vec{n}^s \right) \cdot [\vec{u}^r] = 0. \end{aligned} \quad (3.24)$$

Hence, equation (3.24) reduces to:

$$\vec{\nabla} \cdot \vec{B} = 0. \quad (3.25)$$

From the fifth equation of (3.19) we get

$$\delta \bar{\sigma}^f = \mu^f (\vec{\nabla} \vec{B} + (\vec{\nabla} \vec{B})^T) \cdot [\vec{u}^r] - \mu^f \frac{1}{2|\omega^f|} \oint_{\partial\omega^f} (\vec{B} \cdot \vec{n}) [\vec{u}^r] + \left((\vec{B} \cdot \vec{n}) [\vec{u}^r] \right)^T.$$

The last term is null by virtue of (3.22). Combining the latter and equation (3.20) with the third of (3.19) we obtain

$$\vec{\nabla} \cdot \left(\mu^f (\vec{\nabla} \vec{B} + (\vec{\nabla} \vec{B})^T) - \mu^f \vec{b} \right) \cdot [\vec{u}^r] = -\phi \mu^f \bar{K}^{-1} \cdot [\vec{u}^r], \quad (3.26)$$

where \bar{K} is the permeability tensor that has been defined by the following equation

$$-\phi \mu^f \bar{K}^{-1} \cdot [\vec{u}^r] = \frac{1}{|\omega^f|} \oint_{\omega^f} (\delta \bar{\sigma}^f - P^f \bar{I}) \cdot \vec{n}^f. \quad (3.27)$$

Equation (3.26), by virtue of (3.25) (and since μ_f is constant), can be expressed as

$$\nabla^2 \vec{B} - \vec{\nabla} \vec{b} + \phi \bar{K}^{-1} = 0. \quad (3.28)$$

Using (3.28), (3.25) and (3.22) we get the following system for \vec{B} and \vec{b}

$$\begin{cases} \nabla^2 \vec{B} - \vec{\nabla} \vec{b} + \phi \bar{K}^{-1} = 0 \text{ in } \omega^f \\ \vec{\nabla} \cdot \vec{B} = 0 \text{ in } \omega^f \\ \vec{B} = -\bar{I} \text{ on } \partial\omega^s \cap \partial\omega^f \\ \vec{b}(\vec{X} + \mathcal{C}_i \hat{x}_i) = \vec{b}(\vec{X}) \text{ on } \partial\omega \text{ with } i = 1, 2, 3 \\ \vec{B}(\vec{X} + \mathcal{C}_i \hat{x}_i) = \vec{B}(\vec{X}) \text{ on } \partial\omega \text{ with } i = 1, 2, 3 \\ \left[\vec{B} \right] = 0 \text{ in } \omega^f. \end{cases} \quad (3.29)$$

which is the same model derived in [102] for a still porous media. Using the following change of variables

$$\vec{d} = \frac{\vec{b} \cdot \vec{K}}{\phi}, \quad \vec{D} = \frac{(\vec{B} + \vec{I}) \cdot \vec{K}}{\phi},$$

we get the following system

$$\begin{cases} \vec{\nabla}^2 \vec{D} - \vec{\nabla} \vec{d} = \vec{I} \text{ in } \omega^f \\ \vec{\nabla} \cdot \vec{D} = 0 \text{ in } \omega^f \\ \vec{D} = 0 \text{ on } \partial \omega^s \cap \partial \omega^f \\ \vec{d}(\vec{X} + \mathcal{C}_i \hat{x}_i) = \vec{d}(\vec{X}) \text{ on } \partial \omega \text{ with } i = 1, 2, 3 \\ \vec{D}(\vec{X} + \mathcal{C}_i \hat{x}_i) = \vec{D}(\vec{X}) \text{ on } \partial \omega \text{ with } i = 1, 2, 3. \end{cases} \quad (3.30)$$

And the permeability tensor can be computed as

$$\vec{K} = \phi \left[\vec{D} \right]. \quad (3.31)$$

The first four equations of system (3.30) form a tensorial Stokes problem for the unknowns \vec{D} , \vec{d} . For its solution see [33, 73]. Once solved, it is possible to derive the permeability tensor, using equation (3.31). We are now ready to formulate the closure of equations (3.16). Let's substitute equation (3.27) in the fourth of (3.16), we obtain

$$\begin{cases} \vec{\nabla} \cdot \langle \vec{u}^s \rangle = \Phi \text{ in } \Omega \\ \vec{\nabla} \cdot \langle \vec{u}^f \rangle = -\Phi \text{ in } \Omega \\ \vec{\nabla} \cdot \langle \vec{\sigma}^s \rangle - \vec{\nabla} \langle P^T \rangle + \rho^T \vec{g} = 0 \text{ in } \Omega \\ \frac{\vec{K}}{\mu^r} \left(-\vec{\nabla} [P^f] + \rho^f \vec{g} \right) = \phi [\vec{u}^r] \text{ in } \Omega. \end{cases} \quad (3.32)$$

It has to be complemented by a suitable set of boundary conditions and by an explicit definition of Φ which is going to be discussed later. In other terms, this system of equations states the incompressibility of the two phases and the conservation of the momentum from a macroscopic viewpoint.

3.1.4 Mass equation averaging

In this subsection we derive the mass conservation equations and also an evolution equation for the porosity. Let's start with the derivation of the mass conservation equation of the solid part. Clearly, in the solid part of Ω , we have

$$\frac{\partial}{\partial t} (\rho_i^s) + \vec{\nabla} \cdot (\rho_i^s \vec{u}^s) = 0.$$

Using the superficial average operator we get

$$\frac{\partial}{\partial t} \langle \rho_i^s \rangle + \vec{\nabla} \cdot \langle \rho_i^s \vec{u}^s \rangle + \frac{1}{|\omega|} \oint_{\partial\omega^s} \rho_i^s \vec{u}^s \cdot \vec{n}^s = 0.$$

Since there is no mass flow across $\partial\omega^s$ the last term in the former equation is null. Moreover we have $\langle \rho_i^s \rangle = (1 - \phi)[\rho_i^s] = (1 - \phi)\rho_i^s$ and $\langle \rho_i^s \vec{u}^s \rangle = \rho_i^s \langle \vec{u}^s \rangle = \rho_i^s(1 - \phi)[\vec{u}^s]$, therefore we obtain

$$\frac{\partial}{\partial t} ((1 - \phi)\rho_i^s) + \vec{\nabla} \cdot ((1 - \phi)\rho_i^s[\vec{u}^s]) = 0. \quad (3.33)$$

In the same way we can obtain the mass conservation equation of the fluid part

$$\frac{\partial}{\partial t} (\phi\rho^f) + \vec{\nabla} \cdot (\phi\rho^f[\vec{u}^f]) = 0.$$

Dividing by ρ^f we can get a porosity evolution equation

$$\frac{\partial \phi}{\partial t} + \vec{\nabla} \cdot (\phi[\vec{u}^f]) = 0. \quad (3.34)$$

3.1.5 Averaged energy balance

As we will see later on, in the rest of this work we neglect the thermal effects, however, for completeness, in this section we briefly show the main passages for the derivation of the macroscopic thermal equation.

We start from the equation of the energy balance at the microscale

$$\begin{cases} \frac{\partial}{\partial t}(\rho^k c^k T^k) + \vec{\nabla} \cdot (\rho^k c^k T^k \vec{u}^k) = \\ \vec{\nabla} \cdot (\vec{u}^k \cdot (\vec{\sigma}^k - P^k \vec{I})) - \vec{\nabla} \cdot \vec{w}^k + \rho^k \vec{g} \cdot \vec{u}^k \text{ in } \Omega^k, k = \{s, f\} \\ \vec{w}^k = -H^k \vec{\nabla} T^k \text{ in } \Omega^k, k = \{s, f\} \\ T^s = T^f \text{ on } \partial\Omega^s \cup \partial\Omega^f \\ \vec{w}^s = \vec{w}^f \text{ on } \partial\Omega^s \cup \partial\Omega^f \end{cases} \quad (3.35)$$

where T^k and \vec{w}^k are, respectively, the temperature and the heat flux of the k -th phase. While c^f and H^f are respectively the specific heat and the thermal conductivity of fluid, c^s and H^s are the equivalent values for the solid part. The latter ones can be computed as a volumetric average of the values of each solid species thanks to

$$c^s = \sum_{i=1}^2 \lambda_i c_i^s, \quad H^s = \sum_{i=1}^2 \lambda_i H_i^s,$$

where c_i^s and H_i^s are the thermal capacity and conductivity of the i -th solid species. Moreover equation (3.35) can be complemented by:

$$\begin{cases} \vec{u}^s \cdot \vec{n}^s + \vec{u}^f \cdot \vec{n}^f = 0 \text{ on } \partial\Omega^s \cup \partial\Omega^f \\ (\vec{\sigma}^f - P^f \vec{I}) \cdot \vec{n}^f + (\vec{\sigma}^s - P^s \vec{I}) \cdot \vec{n}^s = 0 \text{ on } \partial\Omega^s \cup \partial\Omega^f \end{cases}$$

Using the same volume averaging techniques described in the previous sections we get the following macroscopic model:

$$\frac{\partial}{\partial t} \left(\sum_{k \in \{s, f\}} \varepsilon^k \rho^k c^k [T] \right) + \vec{\nabla} \cdot \left(\sum_{k \in \{s, f\}} \varepsilon^k \rho^k c^k [T] [\vec{u}^k] \right) = -\vec{\nabla} \cdot (H_{Eff} \vec{\nabla} [T]). \quad (3.36)$$

where $[T]$ is the mean macroscopic temperature and

$$H_{Eff} = \sum_{k \in \{s, f\}} \left(\varepsilon^k H^k + H^k \oint_{\partial\omega^k} \vec{d}^k \vec{n}^k \right)$$

where $\vec{d}^k = \vec{d}_0^k + \vec{A}_0^k \cdot \vec{c}$. The function \vec{d}_0^k can be determined solving the following problem on a periodic unit cell

$$\begin{cases} H^k \vec{\nabla}^2 \vec{d}_0^k = 0 \text{ in } \omega^k, k = \{s, f\} \\ \vec{d}_0^f = \vec{d}_0^s \text{ on } \partial\omega^f \cap \partial\omega^s \\ H^f (\vec{\nabla} \vec{d}_0^f \cdot \vec{n}^f) + H^s (\vec{\nabla} \vec{d}_0^s \cdot \vec{n}^s) = \\ - (H^f \vec{n}^f + H^s \vec{n}^s) \text{ on } \partial\omega^f \cap \partial\omega^s \\ \vec{d}_0^k (\vec{X} + \mathcal{C}_i \vec{x}_i) = \vec{d}_0^k (\vec{X}) \text{ on } \partial\omega \text{ with } i = 1, 2, 3. \end{cases}$$

And the tensor \vec{A}_0^k solving

$$\begin{cases} H^k \vec{\nabla}^2 \vec{A}_0^k = \frac{1}{\varepsilon^k} \vec{I} \text{ in } \omega^k, k = \{s, f\} \\ \vec{A}_0^f = \vec{A}_0^s \text{ on } \partial\omega^f \cap \partial\omega^s \\ H^f (\vec{\nabla} \vec{A}_0^f \cdot \vec{n}^f) + H^s (\vec{\nabla} \vec{A}_0^s \cdot \vec{n}^s) = 0 \text{ on } \partial\omega^f \cap \partial\omega^s \\ \vec{A}_0^k (\vec{X} + \mathcal{C}_i \vec{x}_i) = \vec{A}_0^k (\vec{X}) \text{ on } \partial\omega \text{ with } i = 1, 2, 3. \end{cases}$$

3.1.6 Constitutive equations

In this section we provide the constitutive relations that complement the balance equations. In particular we describe a rheological law for the solid part that links the stress to the deformation velocity. Regarding the rheological relation, in this chapter, we consider only a Newtonian law for the structure. Some more realistic rheologies

will be considered in the forthcoming chapters. The Newtonian relation is:

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

where μ^s is the apparent viscosity of the solid phase. It is possible to compute it as a volume average of μ_i^s that are the species viscosities. Therefore the definition of μ^s is

$$\mu^s = \sum_{i=1}^{n_s} \lambda_i \mu_i^s. \quad (3.37)$$

We point out that the absence of a satisfactory microscopic interaction model for the grains affects the possibility to obtain a macroscopical rheology model by homogenization. Therefore some heuristic models are used.

Let's now pass to the definition of the compaction function Φ . There are two ways to define it:

1. we can relate it directly to the other simulation variables by empirical relations;
2. otherwise, we can define a relation between the porosity ϕ and the other simulation variables. The function Φ is then automatically derived as we will see later on.

The first approach is adopted typically when dealing with a viscous compaction model, see, for example, [27, 36, 98, 99]. One of the mostly used, semi-empirical, relations is

$$\Phi(\phi, [P^e]) = -\frac{\phi - \phi_0}{\eta} [P^e], \quad (3.38)$$

where ϕ_0 is a minimum value for the porosity, $[P^e]$ is the so called effective pressure and η is a parameter which has the same dimensions of the viscosity and describes the resistance of the porous matrix to compaction. It is usually taken as the mean of the coefficients η_i of every single layer, i.e

$$\eta = \sum_{i=1}^{n_s} \lambda_i \eta_i.$$

The coefficients η_i can be found in the literature [98]. Equation (3.38) represents a simple viscous compaction process and is only an example of the constitutive relations that can be considered. There are many definitions of the effective pressure, see [36, 98], among which we mention

$$\begin{cases} [P^e] = \langle P^T \rangle - [P^f] \\ [P^e] = \langle P^T \rangle - (1 - \beta) [P^f] \\ [P^e] = \langle P^T \rangle - \phi [P^f], \end{cases} \quad (3.39)$$

where β is a positive parameter to be calibrated by experiments. In the case we base a constitutive relation directly on Φ it is possible to obtain the porosity ϕ using equation (3.34).

Otherwise, in the second approach (used, for instance, with elastic compaction

laws), we define the dependence of the porosity starting from other variables, such as the effective pressure. Then, using the volume averaging theory, it is possible to derive a relation that links the porosity to the compaction function. In fact, from its definition and from (3.20) we have

$$\Phi = -\frac{1}{|\omega|} \oint_{\partial\omega^s} ([\vec{u}^s] + \delta \vec{u}^s) \cdot \vec{n}^s = -\frac{1}{|\omega|} \oint_{\partial\omega^s} \left([\vec{u}^s] - \frac{1}{2} \vec{\bar{B}} \cdot [\vec{u}^r] \right) \cdot \vec{n}^s.$$

Using equation (3.22) and (3.7) we get

$$\Phi = -\frac{1}{2|\omega|} \oint_{\partial\omega^s} ([\vec{u}^f] - [\vec{u}^s]) \cdot \vec{n}^s = \frac{1}{2} \vec{\nabla} (1 - \phi) \cdot [\vec{u}^r] = -\frac{1}{2} \vec{\nabla} \phi \cdot [\vec{u}^r]. \quad (3.40)$$

The compaction function is then obtained if we define a constitutive relation for the porosity. This latter can depend on depth or on the effective pressure; one of the mostly used relations is the Athy compaction law [6]

$$\phi = \alpha e^{\gamma [P^e]}, \quad (3.41)$$

where α and γ are suitable coefficients to be calibrated. There are many other ϕ -stress relations that may be found in the literature, see [84].

3.1.7 The complete macroscopic models

In this section we rearrange the averaged balance equations, derived in the previous paragraphs, to get a complete and easy-to-discretize model. After formulating the complete model, we discuss some simplified versions used for basin modeling.

Let's start with the mass conservation equation for the solid phase. Since the sedimentary layers are treated as immiscible fluids, the λ_i functions coincide, at a macroscopic level, with the characteristic functions of the subdomains Ω_i namely

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

Therefore equation (3.10) can also be written as

$$\rho^s = \lambda_i \hat{\rho}_i^s.$$

Combining this one with (3.33) we get

$$\hat{\rho}_i^s \frac{\partial}{\partial t} ((1 - \phi) \lambda_i) + \hat{\rho}_i^s \vec{\nabla} \cdot (\lambda_i (1 - \phi) [\vec{u}^s]) = 0, \quad i = 1, \dots, n_s. \quad (3.42)$$

Dividing by $\hat{\rho}_i^s$ and summing all the components we obtain

$$\frac{\partial(1-\phi)}{\partial t} + \vec{\nabla} \cdot ((1-\phi)[\vec{u}^s]) = 0. \quad (3.43)$$

This latter equation is equivalent to (3.34), although $[\vec{u}^s]$ is usually smaller than $[\vec{u}^f]$. Therefore the numerical solution of (3.43), using explicit time advancing finite volume schemes, allows for longer time steps to satisfy the CFL condition. Subtracting (3.43) from (3.42) we get

$$\frac{\partial \lambda_i}{\partial t} + [\vec{u}^s] \cdot \vec{\nabla} \lambda_i = 0. \quad (3.44)$$

This latter relation is a volume transport equation that can be solved using, for instance, the finite volume methods [50] or the discontinuous Galerkin methods [25]. Combining equation (3.43) with (3.34) we obtain

$$-\vec{\nabla} \cdot ((1-\phi)[\vec{u}^s]) = \vec{\nabla} \cdot (\phi[\vec{u}^f]),$$

which may be written as

$$\vec{\nabla} \cdot [\vec{q}] = -\vec{\nabla} \cdot [\vec{u}^s], \quad (3.45)$$

where $[\vec{q}] = \phi[\vec{u}^f]$. We are now able to write two complete averaged models:

1. the first model employs a Φ -stress constitutive equation, like (3.38);
2. the second one employs a porosity-stress constitutive equation, like (3.41).

Using equations (3.32), (3.43), (3.44), (3.36) and (3.45) we obtain for $t > 0$ and in Ω

$$\left\{ \begin{array}{l} \frac{\partial(1-\phi)}{\partial t} + \vec{\nabla} \cdot ((1-\phi)[\vec{u}^s]) = 0 \\ \frac{\partial \lambda_i}{\partial t} + [\vec{u}^s] \cdot \vec{\nabla} \lambda_i = 0 \\ \vec{\nabla} \cdot \langle \vec{u}^s \rangle = \Phi([P^e]) \\ \vec{\nabla} \cdot \langle \bar{\sigma}^s \rangle - \vec{\nabla} \cdot \langle P^T \rangle + \rho^T \vec{g} = 0 \\ \vec{\nabla} \cdot [\vec{q}] = -\vec{\nabla} \cdot [\vec{u}^s] \\ \frac{\bar{k}(\phi)}{\mu^f} \left(-\vec{\nabla} [P^f] + \rho^f \vec{g} \right) = [\vec{q}] \\ \frac{\partial}{\partial t} \left((\sum_{k \in \{s,f\}} \epsilon^k \rho^k c^k) [T] \right) + \vec{\nabla} \cdot \left((\sum_{k \in \{s,f\}} \epsilon^k \rho^k c^k [\vec{u}^k]) [T] \right) = \vec{\nabla} \cdot \left(H_{Eff} \vec{\nabla} [T] \right), \end{array} \right. \quad (3.46)$$

where $\langle \bar{\sigma}^s \rangle$ is, in general, a function of $\langle \vec{u}^s \rangle$ and $[T]$. While, in the second case, we get for $t > 0$ and in Ω

$$\left\{ \begin{array}{l} \phi = \phi([P_e]) \\ \frac{\partial \lambda_i}{\partial t} + [\vec{u}^s] \cdot \vec{\nabla} \lambda_i = 0 \\ \vec{\nabla} \cdot \langle \vec{u}^s \rangle = -\frac{1}{2} \vec{\nabla} \phi \cdot [\vec{u}^r] \\ \vec{\nabla} \cdot \langle \vec{\sigma}^s \rangle - \vec{\nabla} \cdot \langle P^T \rangle + \rho^T \vec{g} = 0 \\ \vec{\nabla} \cdot [\vec{q}] = -\vec{\nabla} \cdot [\vec{u}^s] \\ \frac{\bar{k}(\phi)}{\mu^f} \left(-\vec{\nabla} [P^f] + \rho^f \vec{g} \right) = [\vec{q}] \\ \frac{\partial}{\partial t} \left((\sum_{k \in \{s,f\}} \epsilon^k \rho^k c^k) [T] \right) + \vec{\nabla} \cdot \left((\sum_{k \in \{s,f\}} \epsilon^k \rho^k c^k [\vec{u}^k]) [T] \right) = \vec{\nabla} \cdot \left(H_{Eff} \vec{\nabla} [T] \right). \end{array} \right. \quad (3.47)$$

Both systems have to be supplemented by appropriate boundary conditions. We can outline a common structure in problems (3.46) and (3.47): actually they both represent a set of evolution equations. In the first case the unknowns are the porosity, the mass and the energy, while, in the second case, only the last two are present. The third and fourth equation of (3.46) and (3.47) form a Stokes problem, while the fifth and the sixth equation, (since the velocity $[\vec{u}^s]$ is known by the solution of the Stokes problem), form a Darcy problem. Systems (3.46), (3.47) are similar to models already found in the literature but many details are different. First of all we have stressed in the fourth equation of (3.46) and of (3.47) the presence of the divergence of the superficial volume average of the stress $\langle \vec{\sigma}^s \rangle$. The macroscopical rheological relations usually link the intrinsic mean value of the stress $[\vec{\sigma}^s]$ with the deformation tensor or the deformation velocity tensor. The difference between the two averages is not always considered, although there are works like [12, 13, 75] where the difference is fully taken into account. The second feature of our formulation is the introduction of the compaction function Φ and its dependence on the relative velocity $[\vec{u}^r]$. In many works the compaction function is computed in terms of the material derivative of the effective pressure, see for instance [56]. Here, on the contrary, we have exploited the volume averaging technique to get the functional form of the compaction function. We have also extended the theory of [102] to a moving matrix case, and we have provided a procedure for the computation of the hydraulic permeability and of the effective thermal coefficient.

3.1.8 Approximated models

The compaction models we have derived are similar to the models used, for instance, in [107, 56, 57] and with other compaction models such as [7]. However, the models are still complex and they can be simplified further by adopting some additional hypothesis which are reasonable for an analysis at the scale of a sedimentary basin. The first hypothesis we introduce is that the rheology is temperature independent, therefore we can discard the temperature equations in (3.46) and (3.47). Indeed, the bottom of a sedimentary basin usually lies at a depth of less than ten kilometers where the mean temperature is approximately of three hundred degrees Celsius. In

many cases the variation of the temperature does not alter the rheological properties significantly. Then, we introduce another hypothesis: we neglect the effects of the gradient of the porosity on the rheology and on the compaction rate. This hypothesis is reasonable except in the thin interface regions where the porosity rapidly changes. These effects are usually neglected in basin scale simulations.

Moreover we consider the porosity being prescribed by an experimental law, for instance the Athy law (3.41). For the time being, we neglect only the effects of the gradient of the porosity and consider

$$\vec{\nabla} \cdot \langle \vec{u}^s \rangle = (1 - \phi) \left(\vec{\nabla} \cdot [\vec{u}^s] \right) - [\vec{u}^s] \cdot \left(\vec{\nabla} \phi \right) = \Phi.$$

From this equation we can derive an approximated form for the divergence of the velocity field of the solid phase, namely

$$\vec{\nabla} \cdot [\vec{u}^s] = \frac{\Phi}{1 - \phi}. \quad (3.48)$$

We can perform the same approximation on the rheology, considering a Newtonian law (but the procedure could be straightforwardly extended to the pseudo-plastic relations, see [57]), to get

$$\begin{aligned} \langle \bar{\sigma}^s \rangle = \mu^s \left(\vec{\nabla} \cdot \langle \vec{u}^s \rangle + (\vec{\nabla} \cdot \langle \vec{u}^s \rangle)^T \right) &= (1 - \phi) \mu^s \left(\vec{\nabla} [\vec{u}^s] + (\vec{\nabla} [\vec{u}^s])^T \right) - \\ &\mu^s \left([\vec{u}^s] \vec{\nabla} \phi + ([\vec{u}^s] \vec{\nabla} \phi)^T \right). \end{aligned}$$

Once again, neglecting the effects of the gradient of the porosity field, we obtain

$$\langle \bar{\sigma}^s \rangle = (1 - \phi) \mu^s \left(\vec{\nabla} [\vec{u}^s] + (\vec{\nabla} [\vec{u}^s])^T \right). \quad (3.49)$$

Inserting (3.48) and (3.49) in (3.46) and (3.47) we get two possible simplified models

$$\left\{ \begin{array}{l} \frac{\partial(1 - \phi)}{\partial t} + \vec{\nabla} \cdot ((1 - \phi)[\vec{u}^s]) = 0 \\ \frac{\partial \lambda_i}{\partial t} + [\vec{u}^s] \cdot \vec{\nabla} \lambda_i = 0 \\ \vec{\nabla} \cdot [\vec{u}^s] = \frac{\Phi([P^e])}{1 - \phi} \\ \vec{\nabla} \cdot \langle \bar{\sigma}^s \rangle - \vec{\nabla} \cdot \langle P^T \rangle + \rho^T \vec{g} = 0 \\ \vec{\nabla} \cdot [\vec{q}] = -\frac{\Phi([P^e])}{1 - \phi} \\ \frac{\bar{\kappa}(\phi)}{\mu^f} \left(-\vec{\nabla} [P^f] + \rho^f \vec{g} \right) = [\vec{q}] \end{array} \right. \quad (3.50)$$

and

$$\left\{ \begin{array}{l} \phi = \phi([P_e]) \\ \frac{\partial \lambda_i}{\partial t} + [\vec{u}^s] \cdot \vec{\nabla} \lambda_i = 0 \\ \vec{\nabla} \cdot [\vec{u}^s] = -\frac{1}{2} \frac{\vec{\nabla} \phi \cdot [\vec{u}^r]}{1-\phi} \\ \vec{\nabla} \cdot \langle \vec{\sigma}^s \rangle - \vec{\nabla} \cdot \langle P^T \rangle + \rho^T \vec{g} = 0 \\ \vec{\nabla} \cdot [\vec{q}] = \frac{1}{2} \frac{\vec{\nabla} \phi \cdot [\vec{u}^r]}{1-\phi} \\ \frac{\bar{k}(\phi)}{\mu^f} \left(-\vec{\nabla} [P^f] + \rho^f \vec{g} \right) = [\vec{q}]. \end{array} \right. \quad (3.51)$$

Systems (3.50) and (3.51) closely resemble some compaction models present in the literature, such as [96] and [7], where an elastic and a viscoelastic medium are treated. With respect to those works, we have used a different volume averaging technique that makes possible to model the compaction function directly. Moreover in our systems (3.50) and (3.51) a variable number of sedimentary layers is considered. As we have already stressed, systems (3.50) and (3.51) belong to the class of compaction models. To derive a basin-scale model we eliminate the fifth and sixth equations in (3.51), obtaining the reduced system

$$\left\{ \begin{array}{l} \phi = \phi([P^e]) \\ \frac{\partial \lambda_i}{\partial t} + [\vec{u}^s] \cdot \vec{\nabla} \lambda_i = 0 \\ \vec{\nabla} \cdot [\vec{u}^s] = \frac{\Phi([P^e])}{1-\phi} \\ \vec{\nabla} \cdot \langle \vec{\sigma}^s \rangle - \vec{\nabla} \cdot \langle P^T \rangle + \rho^T \vec{g} = 0. \end{array} \right. \quad (3.52)$$

The latter system is the classical model widely used in geological applications [107, 56, 60, 108]. In this model the liquid phase is not simulated, therefore the compaction effects are imposed a priori using experimentally derived compaction curves. In particular, the effective pressure is assumed to be the lithostatic one and Φ must be estimated using only the overburden pressure; for more details see, for example, [56].

The basin scale models can be used in all the cases in which the relative velocity and the pore pressure have weak variations. In that case the compaction function can be estimated precisely without we need to use the fluid velocity and pressure, thus allowing a decoupling between the evolution of the solid matrix and the evolution of the fluids.

3.2 Numerical solution for compaction models

3.2.1 Time discretization algorithm for (3.50)

In this section we introduce a class of numerical schemes for the solution of problem (3.50) based on a temporal splitting technique combined with a fixed point iteration.

We introduce a partition of the temporal interval $[0, t^{fin}]$, $[0, t^1, \dots, t^n, \dots, t^{fin}]$ with $\Delta t^n = t^{n+1} - t^n$. For simplicity we define, for a general variable $a^k(t, \vec{X})$ relative to the phase k ,

$$a^{k,n}(\vec{X}) = a^k(t^n, \vec{X}).$$

with $t \in [0, t^{fin}]$, $\vec{X} \in \Omega$. We define the following algorithm:

1. given ϕ^n and λ_i^n , we compute the density and the viscosity of the solid part $\rho^{s,n}$, $\mu^{s,n}$ using (3.10) and (3.37);
2. we set $[\vec{u}^{s,(0)}] = [\vec{u}^{s,n-1}]$, compute $\Phi^{(0)} = \Phi(\phi^n, [P^{e,(0)}])$ and $[P^{e,(0)}]$, which is a function of $\langle P^{T,n-1} \rangle$, $[P^{f,n-1}]$, using (3.39);
3. we iterate on m :

- a. we solve the Stokes problem

$$\begin{cases} \vec{\nabla} \cdot \langle \bar{\sigma}^{s,(m+1)} \rangle - \vec{\nabla} \cdot \langle P^{T,(m+1)} \rangle + \rho^{T,n} \vec{g} = 0 \\ \vec{\nabla} \cdot [\vec{u}^{s,(m+1)}] = \Phi(\phi^n, [P^{e,(m)}]), \end{cases} \quad (3.53)$$

where $\langle \bar{\sigma}^{s,(m+1)} \rangle$ is a function of $[\vec{u}^{s,(m+1)}]$;

- b. we solve the Darcy problem

$$\begin{cases} [\vec{q}^{(m+1)}] = -\frac{\bar{k}}{\mu^f} \left(\vec{\nabla} [P^{f,(m+1)}] + \rho^{f,n} \vec{g} \right) \\ \vec{\nabla} \cdot [\vec{q}^{(m+1)}] = -\Phi(\phi^n, [P^{e,(m)}]); \end{cases}$$

- c. we update the effective pressure $[P^{e,(m+1)}]$ using one of (3.39), and the compaction function $\Phi(\phi^n, [P^{e,(m+1)}])$;
4. if $\|[P^{e,(m+1)}] - [P^{e,(m)}]\|_{L^2(\Omega)}$, is smaller than a prescribed tolerance, we stop and set, for all variables a^k , $a^{k,n} = a^{k,(m+1)}$;
5. we solve the following evolution equations from t^n to t^{n+1} obtaining λ_i^{n+1} and ϕ^{n+1}

$$\begin{cases} \frac{\partial \lambda_i}{\partial t} + [\vec{u}^s] \cdot \vec{\nabla} \lambda_i = 0 \\ \frac{\partial (1 - \phi)}{\partial t} + \vec{\nabla} \cdot ((1 - \phi)[\vec{u}^s]) = 0. \end{cases}$$

3.2.2 Time discretization algorithm for (3.51)

We modify slightly the previous algorithm in order to adapt it to (3.51):

1. given ϕ^n and λ_i^n we compute the density and the viscosity of the solid part $\rho^{s,n}$, $\mu^{s,n}$;
2. we set $[\vec{u}^{s,(0)}] = [\vec{u}^{s,n-1}]$, $[\vec{u}^{T,(0)}] = \frac{[\vec{q}]^{n-1}}{\phi^{n-1}}$, $\phi^{(0)} = \phi([P^{e,0}])$ and compute $[P^{e,(0)}]$ using $\langle P^{T,n-1} \rangle$, $[P^{f,n-1}]$;

3. then we iterate on m :

a. we solve the Stokes problem

$$\begin{cases} \vec{\nabla} \cdot \langle \bar{\sigma}^{s,(m+1)} \rangle - \vec{\nabla} \cdot \langle P^{T,(m+1)} \rangle + \rho^{T,n} \vec{g} = 0 \\ \vec{\nabla} \cdot [\vec{u}^{s,(m+1)}] = -\frac{1}{2} \frac{\vec{\nabla} \phi^{(m)}}{(1-\phi^{(m)})} \cdot [\vec{u}^{r,(m)}], \end{cases}$$

where $\langle \bar{\sigma}^{s,(m+1)} \rangle$ is a function of $[\vec{u}^{s,(m+1)}]$.

b. we solve the Darcy problem

$$\begin{cases} [\vec{q}^{(m+1)}] = -\frac{\bar{K}}{\mu^f} \left(\vec{\nabla} [P^{f,(m+1)}] + \rho^{f,n} \vec{g} \right) \\ \vec{\nabla} \cdot [\vec{q}^{(m+1)}] = \frac{1}{2} \frac{\vec{\nabla} \phi^{(m)}}{(1-\phi^{(m)})} \cdot [\vec{u}^{r,(m)}]; \end{cases}$$

c. we update the effective pressure $[P^{e,(m+1)}]$ (using one of the (3.39)), the porosity $\phi^{(m+1)} = \phi([P^{e,(m+1)}])$ and the relative speed $[\vec{u}^{r,(m+1)}] = \frac{[\vec{q}^{(m+1)}]}{\phi^{(m+1)}}$;

4. if $\|[P^{e,(m+1)}] - [P^{e,(m)}]\|_{L^2(\Omega)}$ is smaller than a prescribed tolerance, we stop and set, for the variable a^k , $a^{k,n} = a^{k,(m+1)}$;

5. we solve the following evolution equation from t^n to t^{n+1} and we get λ_i^{n+1}

$$\frac{\partial \lambda_i}{\partial t} + [\vec{u}^s] \cdot \vec{\nabla} \lambda_i = 0.$$

3.2.3 Algorithm analysis

In the former section we have outlined that the compaction problem requires the solution of a Stokes and a Darcy problem. In this section we will study the conditions under which these two problems are well posed. For a Newtonian rheology the necessary existence conditions on the data for the Stokes problem are

$$\mu^s > 0, \quad \mu^s, \rho^T \in L^\infty, \quad (3.54)$$

see [73], [33]. The well posedness conditions for the Darcy problem are

$$\bar{K}, \mu^f > 0, \quad \bar{K}, \mu^f, \rho^f \in L^\infty. \quad (3.55)$$

Conditions (3.54) and (3.55) are satisfied if the porosity is strictly positive $\phi > 0$. If we use a ϕ -stress relation it is sufficient to make sure that the dependence of the porosity on pressure implies the positivity of ϕ . If a compaction law involving the compaction function Φ is used, it is necessary to verify, case by case, the positivity of the porosity. If relation (3.38) is used it is possible to guarantee the following:

Proposition 3.3. *The relation (3.38), if $\phi_0 > 0$, and $[P_e]$ is bounded (i.e. $\exists M_p > 0$ finite such that $[P_e] < M_p$), implies the strict positivity of ϕ .*

Proof. The boundness of $[P_e]$ is physically reasonable but, till now, we can not prove it mathematically.

Equation (3.43) can also be written as

$$\frac{D\phi}{Dt} = (1 - \phi) \vec{\nabla} \cdot [\vec{u}^s]$$

and combining it with (3.48) we get

$$\frac{D\phi}{Dt} = \Phi. \quad (3.56)$$

being $\frac{D}{Dt} = \frac{\partial}{\partial t} + [\vec{u}^s] \cdot \vec{\nabla}$. Then using a standard characteristic equation argument (see [50]) if $[\vec{u}^s]$ is Lipschitz continuous uniformly on t there is a variable change $(t, \vec{X}) \rightarrow (t, \vec{P})$ such that

$$\begin{cases} \frac{d}{dt} \vec{P}(t, \vec{X}) = [\vec{u}^s](t, \vec{X}) \\ \vec{P}(0, \vec{X}) = \vec{X}. \end{cases}$$

Then, the problem (3.56) in the (t, \vec{P}) coordinates reads

$$\begin{cases} \frac{d\phi}{dt} = \Phi \\ \phi(0, \cdot) = \phi_0 \end{cases}$$

where ϕ_0 is the initial condition for the porosity. Clearly we have $\phi \geq \bar{\phi}$ where $\bar{\phi}$ satisfies

$$\begin{cases} \frac{d\bar{\phi}}{dt} = -\frac{\bar{\phi} - \phi_0}{\eta} M_p \\ \bar{\phi}(0, \cdot) = \phi(0, \cdot) \end{cases}.$$

The analytical solution of this problem is $\bar{\phi}(t, \vec{P}) = \phi_0 + (\phi(0, \vec{P}) - \phi_0) e^{-\frac{M_p}{\eta} t}$, therefore

$$\phi(t, \vec{P}) \geq \phi_0 + (\phi(0, \vec{P}) - \phi_0) e^{-\frac{M_p}{\eta} t} \geq \phi_0 > 0 \quad \forall t > 0$$

And we get the thesis.

3.3 Numerical solution for a one-dimensional case

3.3.1 The one-dimensional case and numerical techniques

We solve problem (3.50) using the compaction function (3.38) neglecting the thermal effects. We consider only vertical compaction, consequently, the corresponding one-dimensional model to be solved is

$$\left\{ \begin{array}{l} \frac{\partial(1-\phi)}{\partial t} + \frac{\partial}{\partial x}((1-\phi)[u^s]) = 0 \\ \frac{\partial \lambda_i}{\partial t} + [u^s] \frac{\partial \lambda_i}{\partial x} = 0 \\ \frac{\partial [u^s]}{\partial x} = -\frac{\phi - \phi_0}{\eta} [P^e] \\ \frac{\partial}{\partial x} \left(2\mu^s \frac{\partial [u^s]}{\partial x} \right) - \frac{\partial \langle P^T \rangle}{\partial x} + \rho^T g = 0 \\ \frac{\partial [q]}{\partial x} = \frac{\phi - \phi_0}{\eta} [P^e] \\ \frac{\kappa}{\mu^f} \left(-\frac{\partial [P^f]}{\partial x} + \rho^f g \right) = [q] \\ [u^s(\cdot, 0)] = 0, \quad [q(\cdot, 0)] = 0 \\ \langle P^T \rangle(\cdot, L) = 0, \quad [P^f](\cdot, L) = 0 \\ \phi(0, X) = \bar{\phi}(X), \quad \lambda_i(0, X) = \bar{\lambda}_i(X), \end{array} \right. \quad (3.57)$$

in $(0, 1) \times (0, t^{fin}]$. Here $\bar{\phi}$ and $\bar{\lambda}_i$ are suitable initial values and L is the surface of the basin, which is determined by the following Cauchy problem

$$\left\{ \begin{array}{l} \frac{dL}{dt} = [u^s] \\ L(0) = L_0 \end{array} \right.$$

where L_0 is the initial thickness of the basin. In this one-dimensional case it is not necessary to use a finite element method in order to solve the Stokes and Darcy problems associated with (3.57). Indeed, problem (3.53) is equivalent to

$$\left\{ \begin{array}{l} \frac{\partial [u^{s,(m+1)}]}{\partial x} = -\frac{\phi^n - \phi_0}{\eta} [P^{e,(m)}] \\ \frac{\partial}{\partial x} \left(2\mu^s \frac{\partial [u^{s,(m+1)}]}{\partial x} \right) - \frac{\partial \langle P^{T,(m+1)} \rangle}{\partial x} + \rho^{T,n} g = 0 \\ [u^{s,(m+1)}](\cdot, 0) = 0, \quad \langle P^{T,(m+1)} \rangle(\cdot, L) = 0. \end{array} \right. \quad (3.58)$$

The first equation of (3.58), combined with the last one, forms a Cauchy problem. Once determined $[u^{s,(m+1)}]$, we can compute the pressure from the second equation of (3.58). A similar procedure leads to the solution of the Darcy problem

$$\begin{cases} \frac{\partial [q]}{\partial x} = \frac{\phi - \phi_0}{\eta} [P^e] \\ \frac{K}{\mu^f} \left(-\frac{\partial [P^f]}{\partial x} + \rho^f g \right) = [q] \\ [q(\cdot, 0)] = 0, [P^f](\cdot, L) = 0 \end{cases} \quad (3.59)$$

The first equation of (3.59) complemented by $[q](\cdot, 0) = 0$ forms a Cauchy problem for $[q]$, then $[P^f]$ can be computed by the second equation of (3.59) with the final condition $[P^f](\cdot, L) = 0$.

To sum up, the solution of (3.57) can be obtained with an ordinary differential equation (ODE) integrator and a Godunov finite volume solver applied to the first two evolution equations of (3.57). In our case we have used an explicit Euler scheme as the one dimensional case is not computationally demanding.

3.3.2 Numerical results

We have considered two significant numerical tests. In the first one, we consider a compaction test case with a low variation in the physical coefficients while, in the second one, we consider a case where a high variation in the coefficients leads to the formation of extreme overpressure conditions. We consider three sedimentary layers (see Figure 3.3), the deepest one is located between the basement and 450m, the second one between 450m and 1Km and the third one between and 1Km and 2Km: the total depth of the basin is 2Km. The physical characteristics of the materials for

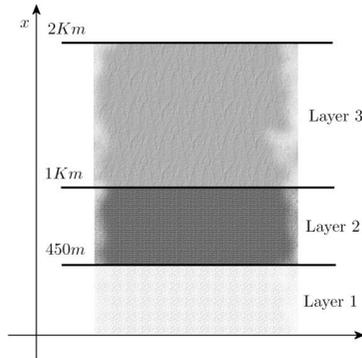


Fig. 3.3: An outline of the basin stratification.

the first test case are

Species	1	2	3
Viscosity ($10^{20} Pa \cdot s$)	3	1	3
Density ($10^3 Kg/m^3$)	1	1	3
η ($10^{20} Pa \cdot s$)	100	30	100

The remaining physical data are $\phi_0 = 0.01$, $\rho_f = 10^3 kg/m^3$, $\mu^f = 10^{-5} Pa \cdot s$ and in this case we have considered an empirical relation for the permeability tensor [28]

$$\bar{K} = \bar{K}_0 \left(\frac{\phi}{\phi_0} \right)^m,$$

where K_0 and m are experimental coefficients. We show in Figure 3.4 and in Figure

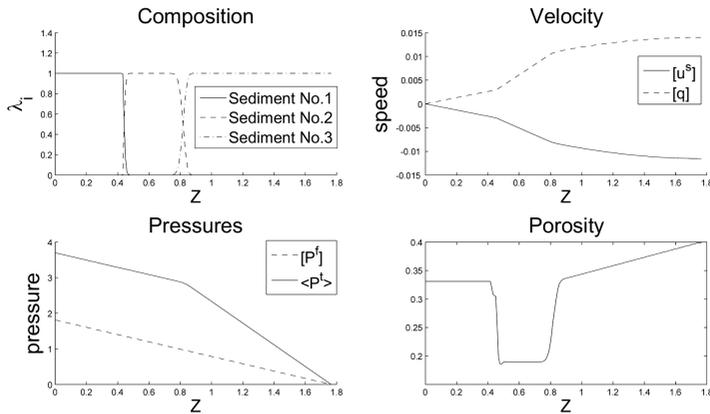


Fig. 3.4: Basin evolution at 0.49 million years with a moderate variation of the coefficients. In the upper-left corner the functions λ_i , that are the volumetric fractions of the sediments, are displayed. In the upper right corner the solid velocity and the relative flux are displayed. In the lower-left corner the total and pore pressures are displayed. Lastly in the lower right corner the porosity is displayed.

3.5 the evolution of the basin at 0.49 and 94.7 million years respectively. Since there are no strong variations in the coefficients, the evolution of the basin is quite smooth, the central layer compacts first, as expected, but the compaction rate is not very high. Therefore the fluid in the deepest layer has plenty of time to escape and no overpressure is formed. In fact we can see in Figures 3.4 and 3.5 that the fluid pressure maintains a nearly hydrostatic behavior.

In the second test case we consider the same geometry of the first one, while we change the strength of the middle layer:

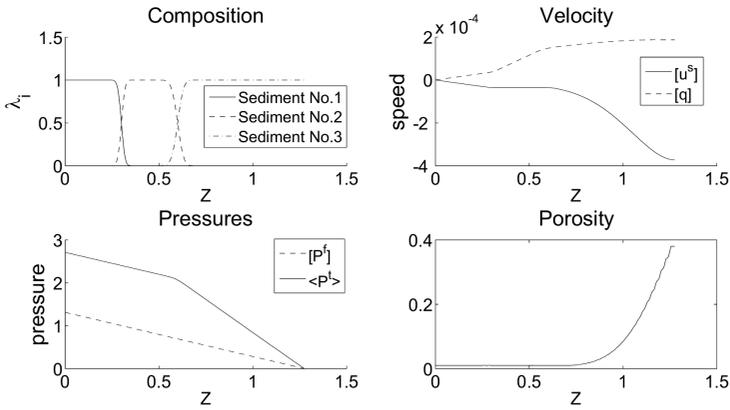


Fig. 3.5: Basin evolution at 94.7 million years with a moderate variation of the coefficients.

Species	1	2	3
Viscosity ($10^{20} Pa \cdot s$)	3	1	3
Density ($10^3 Kg/m^3$)	1	1	3
η ($10^{20} Pa \cdot s$)	100	1	100

In other terms, the central layer is very weak and compacts quickly. An over-compacted layer behaves as an impermeable zone and the liquids below it are trapped. In these conditions the pore pressure of the deepest layer reaches high values. In Figure 3.6 the basin evolution at 0.27 million years is depicted. In that

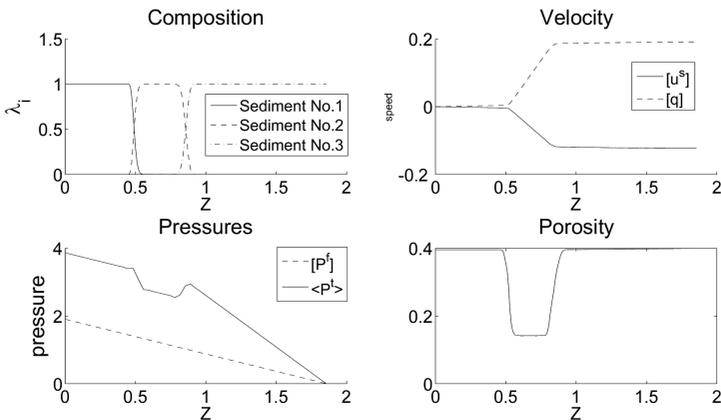


Fig. 3.6: Basin evolution at 0.27 million years with high variation of the coefficients.

period the central region collapses and reaches the minimum porosity. The water is retained only in the lowest part of the basin. Then its lowest part is compacted, as

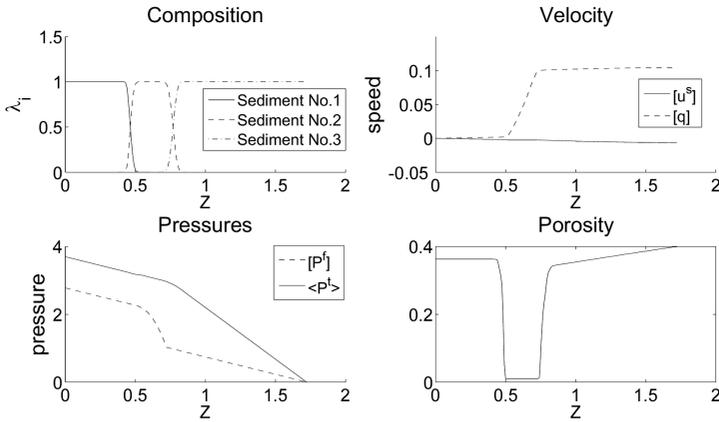


Fig. 3.7: Basin evolution at 4.2 million years with high variation of the coefficients.

we can see in Figure 3.7. Water tries to escape from the deepest layer but the middle, over-compacted, layer prevents this event and the pore pressure rises. The pressure $[P^f]$ deviates from the hydrostatic behavior and reaches $2.8 \cdot 10^7 Pa$. At 86.8 million

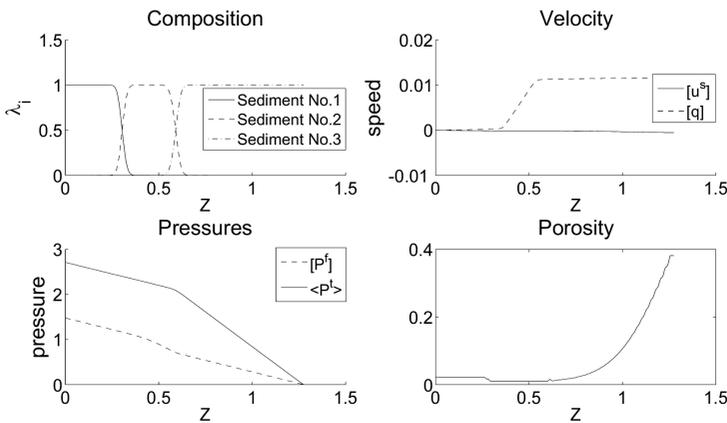


Fig. 3.8: Basin evolution at 86.8 million years with high variation of the coefficients.

years, as shown in Figure 3.8, the compaction of the deepest layer terminates and only the shallowest region remains un-compacted. it is worth noting that the basin

thickness is reduced from $2Km$ to $1.3Km$. In this last test case we have shown that our algorithm remains stable even under extreme conditions.

3.4 Nomenclature

Index	Meaning	Interval
k	Phase - 1 (fluid), 2(solid)	$[1, 2]$
i	Layer	$[1, n_s]$
n	Time step	-
(m)	Iteration	-

Table 3.2: Table of the indexes.

Symbol	Meaning
Ω	The sedimentary basin volume - The numerical domain
Ω^k	The k -th phase volume
Ω_i^s	The i -th solid layer volume
ω	Averaging volume
ω^k	The k -th phase part of the averaging volume
\vec{X}	Position vector
L	Macroscopical (Basin) scale
l	Microscopical (Grain) scale
\vec{n}	The normal ω / Ω^k
\vec{n}^k	The normal to ω^k / Ω^k
∂	The boundary

Table 3.3: Table of the geometry symbols.

Symbol	Meaning
μ^k	The viscosity of the k -th phase
μ_i^s	The viscosity of the i -th sediment layer
η	The bulk viscosity
η_i	The bulk viscosity of the i -th sediment layer
ρ^k	The density of the k -th phase
ρ^T	The total density
P^k	The pressure of the k -th phase
P^T	The total pressure
P^e	The effective pressure
$\bar{\sigma}^k$	The viscous stress of the k -th phase
T	The temperature
c^k	The thermal capacity of the k -th phase
c_i^s	The thermal capacity of the i -th layer
H^k	The thermal conductivity of the k -th phase
c_i^s	The thermal conductivity of the i -th layer
\vec{u}^k	The velocity of the k -th phase
\vec{K}	The permeability tensor

Table 3.4: Table of the physical variables.

Symbol	Meaning
ε^k	The volume fraction of the k -th phase
ϕ	The porosity
λ_i	The volume fraction of the i -th layer or the characteristic function of Ω_i
a^k	A generic k -th phase variable
$\langle a^k \rangle$	The superficial average
$[a^k]$	The intrinsic average
$\tilde{\delta} a^k$	The variation with respect to the intrinsic average
n_s	The number of the layers
t	The time
Φ	The compaction function

Table 3.5: Table of the other variables.

Chapter 4

Multi-fluid tracking

In the former chapter we have derived some physical models for the compaction and the basin-scale dynamics. This chapter is entirely devoted to the numerical solution of problem (3.44) in three dimensions. To this aim it is necessary to tackle the problem of tracing the geometrical evolution of a large number of sedimentary layers. We disregard the other equations in (3.46) and we focus just on problem (3.44). We assume that the velocity field $[\vec{u}^s]$ is known. For the sake of simplicity we drop the index s and the brackets $[\cdot]$ as, from now on, we neglect the evolution of the fluid part and we consider only macroscale quantities. We illustrate a method to track separating interfaces among immiscible fluids when a large number of fluids is involved. Our aim is to construct an efficient and robust method, effective even when the interfaces experience a strong deformation, with good mass conservation properties, that can be used on (2D and 3D) unstructured meshes.

4.1 A brief review of the literature

Many techniques regarding the two fluid problem are reported in the literature but they cannot often be extended readily to the multi-fluid problem and, moreover, do not match our requirements. Tracking methods can be roughly subdivided into two categories: Lagrangian and Eulerian methods. The former track the interfaces explicitly, while the latter reconstruct them with a post-processing procedure. Among the many Lagrangian tracking algorithms (see, for instance, [59, 93, 72]), some move all the nodes of the volume mesh. Some others, on the contrary, track just the interface points and reconstruct the mesh in the interior at every time step, or whenever necessary. The Lagrangian approach presents some difficulties, particularly in three-dimensional computations, such as the treatment of possible, physical or numerically induced, topological changes. Furthermore, sophisticated adaption algorithms should be used to maintain a sufficient mesh quality. If a topological change occurs, complex topology correction algorithms are needed (see, for instance, the one in [64]), and it is almost impossible to prove the algorithm robustness

with respect to all topological changes that may happen in complex and realistic 3D situations. Some works, for example [91, 82], have been proposed to tackle the topological change problem. These approaches require a fixed background grid for the solution of a geometry regularization equation. Though interesting, they can be applied only to two fluid simulations and are not mass-preserving.

Though the Lagrangian methods have an explicit and immediate representation of the interfaces (see [59, 72, 39, 93]) they are not conservative. There are works, like [62], which present procedures to enforce mass conservation, but fail to be robust for topological changes.

The complexity of Lagrangian methods triggered the development of the Eulerian implicit tracking methods: an overview can be found in [48] and [21]. We remind the most effective methods, namely the volume of fluid (VOF) and the level-set (LS). Other mixed Eulerian-Lagrangian methods exist, such as the ALE methods [29] or the particle methods [65], but they do not have the characteristics we are looking for (conservativeness, robustness with respect to topological changes). The LS [30, 86, 65, 81] is a robust method and it is easy to code. But in many cases it does not fit the multi fluid framework and, in general, it does not conserve the mass. A few works are devoted to the multi-fluid simulation (see, for instance, [109, 110]) however, the first is specifically designed for curvature-driven flows while the other entails a nested LS structure and is not mass conservative.

Many works are devoted to fix the LS non conservativeness like [90] and [69], yet all of them consider only the case of two fluids. The mass conservation issue can be partially solved by refining the grid adaptively, as pointed out in [1] and [2].

VOF methods are mass conservative by construction and relatively robust although they are usually designed to track only two fluids and moreover they have, in general, an irregular reconstruction of interfaces. The principal difficulty is again the check of the reconstruction of the interfaces when more than two fluids are involved. Interface reconstruction using the VOF methods is a major topic and many works such as [9, 8] are devoted to it. However the multi-fluid case is not usually treated and many VOF algorithms require a structured mesh.

One of the most applicable methods for multi-fluid simulations is the partial volume tracking method (VT) (see [48] for a brief description) which consists in discretizing, with high order schemes, the volume transport equation. As stated in [21], this approach has a moderate success, since the discontinuous initial solutions are quickly diffused even if high resolution methods are used.

4.2 The proposed method

We consider a domain $\Omega \subset \mathbb{R}^d$, with $d = 1, 2, 3$, with a regular boundary $\partial\Omega$; this domain is filled with n_s immiscible fluid species, such that every subdomain $\Omega_i \subset \Omega$, corresponding to a species, does not overlap with the others, i.e., $\Omega_i \cap \Omega_j = \emptyset$ if $i \neq j$ and $\overline{\Omega} = \bigcup_{i=0}^{n_s} \overline{\Omega}_i$. The subdomains Ω_i depend on time, i.e., $\Omega_i = \Omega_i(t)$, since they are advected by a time dependent velocity field $\vec{u}(t, \vec{X})$, $\vec{X} \in \Omega$ and $t \geq 0$,

whose trace on $\partial\Omega$ has zero normal component, i.e., $\vec{u} \cdot \vec{n} = 0$ on $\partial\Omega$, being \vec{n} the boundary normal vector. In other terms, for the sake of simplicity, we have neglected the boundary conditions without losing generality. We also assume that \vec{u} is sufficiently regular, in particular we assume at each time t $\vec{u}(t, \vec{X}) \in H^1(\Omega)$. We define $\lambda_i^0 \in L^2(\Omega)$ as the characteristic function of subdomain Ω_i at initial time, i.e., $\lambda_i^0(\vec{X}) = \chi_{\Omega_i(0)}$ where

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

for $i = 0, \dots, n_s$. Therefore the following relation holds: $\sum_{i=1}^{n_s} \lambda_i^0 = 1$ almost everywhere in Ω . The VT equation for a given vector field \vec{u} is

$$\begin{cases} \frac{\partial \lambda_i}{\partial t} + \vec{\nabla} \cdot (\lambda_i \vec{u}) - \lambda_i (\vec{\nabla} \cdot \vec{u}) = 0 & t > 0, \quad i = 1, \dots, n_s \\ \lambda_i = \lambda_i^0; & t = 0, \end{cases} \quad (4.1)$$

where $\lambda_i(t, \cdot) \in L^2_\Omega$ is a weak solution of (4.1). This equation is equivalent to the

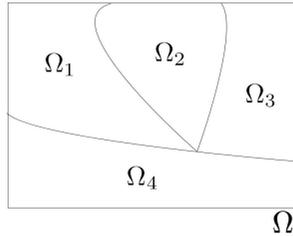


Fig. 4.1: Domain Ω and its subdomains Ω_i .

transport equation

$$\frac{\partial \lambda_i}{\partial t} + \vec{u} \cdot \vec{\nabla} \lambda_i = 0 \quad (4.2)$$

Equation (4.1) has been considered instead of (4.2) because, as we will see, its corresponding discrete form has better properties.

Problem (4.1) has some properties we wish to recall.

Proposition 4.1. *If the initial condition satisfies $\sum_{i=1}^{n_s} \lambda_i^0 = 1$ almost everywhere in Ω then $\sum_{i=1}^{n_s} \lambda_i = 1$ almost everywhere in Ω for $\forall t > 0$.*

Proof. We can proceed formally by summing up the i -th equations in (4.1), obtaining

$$\frac{\partial}{\partial t} \left(\sum_{i=1}^{n_s} \lambda_i \right) + \vec{\nabla} \cdot \left(\vec{u} \sum_{i=1}^{n_s} \lambda_i \right) - (\vec{\nabla} \cdot \vec{u}) \sum_{i=1}^{n_s} \lambda_i = 0. \quad (4.3)$$

By inspection it can be verified that $\sum_{i=1}^{n_s} \lambda_i = 1, \forall t \geq 0$ is a solution of (4.3) and it satisfies the initial condition. From the linearity of the differential problem it follows that the solution is also unique.

Proposition 4.2. *If the initial condition satisfies $0 \leq \lambda_i^0 \leq 1$ almost everywhere in Ω and the velocity field $\vec{u}(t, \vec{X})$ is Lipschitz continuous uniformly on t for $\vec{X} \in \Omega$ then $0 \leq \lambda_i \leq 1, \forall t > 0$ almost everywhere in Ω .*

Proof. We use a standard characteristic theory argument. Let $(t, \vec{P}(t))$ be a time-dependent variable change $(t, \vec{X}) \rightarrow (t, \vec{P})$ defined implicitly by the following time-backward ODE problem

$$\begin{cases} \frac{d\vec{P}}{ds} = -\vec{u}(s, \vec{P}(s)) \text{ on } (t, 0] \\ \vec{P}(t) = \vec{X}. \end{cases} \quad (4.4)$$

The Cauchy problem (4.4), thanks to the hypotheses on the velocity field, has a unique solution. Equation (4.2) written in the (t, \vec{P}) coordinates reads

$$\frac{\partial \lambda_i}{\partial t} + \frac{d\vec{P}}{dt} \cdot \vec{\nabla} \lambda_i + \vec{u} \cdot \vec{\nabla} \lambda_i = 0, \quad (4.5)$$

where, for the sake of simplicity, we have used $\vec{\nabla}$ to denote the gradient with respect to \vec{P} . Substituting the first of (4.4) into (4.5) we get $\frac{d}{dt} \lambda_i(t, \vec{P}(t)) = 0$ almost everywhere in Ω therefore for almost all \vec{X} there exists a $\vec{P}(0)$ such that $\lambda_i(t, \vec{X}) = \lambda_i(t, \vec{P}(0))$ and the thesis follows.

Let's now consider the definition of a level set description of the same subdomains. We define $\phi_i : \mathbb{R}^+ \times \overline{\Omega} \rightarrow \mathbb{R}$, with $\phi_i(t, \cdot) \in C^0(\overline{\Omega}) \quad \forall t > 0, i = 1, \dots, n_s$, some level set functions such that $\Omega_i(t) = \{\vec{X} \in \overline{\Omega} : \phi_i(t, \vec{X}) > \frac{1}{2}\}$ and consequently $\partial \Omega_i(t) = \{\vec{X} \in \overline{\Omega} : \phi_i(t, \vec{X}) = \frac{1}{2}\}$. This particular value of the set will be useful when we find an analogy between the discrete forms of LS and the VT equations. We can write the following evolution equation for each ϕ_i

$$\begin{cases} \frac{\partial \phi_i}{\partial t} + \vec{\nabla} \cdot (\phi_i \vec{u}) - \phi_i (\vec{\nabla} \cdot \vec{u}) = 0; & t > 0 \\ \phi_i = \phi_i^0; & t = 0, \end{cases} \quad (4.6)$$

by which, at all times, $\lambda_i = H(\phi_i - \frac{1}{2})$, where

$$H(\rho) = \begin{cases} 1 & \text{if } \rho > 0 \\ 0 & \text{otherwise} \end{cases}$$

is the Heaviside function, and ϕ_i^0 is the initial condition. In other words, at the continuous level, equations (4.1) and (4.6) are two equivalent ways to describe the interface motion. However, in the discrete setting we will use two different spaces

for the discrete λ_i and ϕ_i , leading to our numerical scheme.

We now introduce the discrete form of the equations: let \mathcal{T}_Δ a conforming (structured or unstructured) grid on Ω made of either simplex or quad elements. The grid \mathcal{T}_Δ has n_e elements indicated by $e_r, r = 1, \dots, n_e$ and n_p nodes denoted by $\vec{x}_k, k = 1, \dots, n_p$. Let Δ be the maximum diameter of the elements. Consider the dual mesh made of $n_c = n_p$ cells $\tau_k, k = 1, \dots, n_c$, centered on the nodes \vec{x}_k , and built by connecting the barycenters of the elements to the barycenters of the faces, see Figure 4.2. Let $\mathbb{I}_k^C = \{k_j, j = 1, \dots, |\mathbb{I}_k^C|\}$, be the set of the indexes of the cells surrounding cell τ_k , and let $\{\tau_{k_j}\}, j = 1, \dots, |\mathbb{I}_k^C|$, be the set of cells surrounding τ_k . The common surface between τ_k and τ_{k_j} is indicated by l_k^j . We also indicate by ι the index such that, given the indexes k and j , $\iota : l_{k_j}^\iota = l_k^j$. In other words every interface between the cells τ_k and τ_{k_j} can be identified by two indices j and ι respectively, depending whether it is a face of τ_k or τ_{k_j} : see Figure 4.2.

For the sake of clarity, we will adopt in this chapter the following convention: the index i will always refer to the fluid species, k to the cell related quantities, j to the interface related values, r to the elements, and n to the time steps. Let

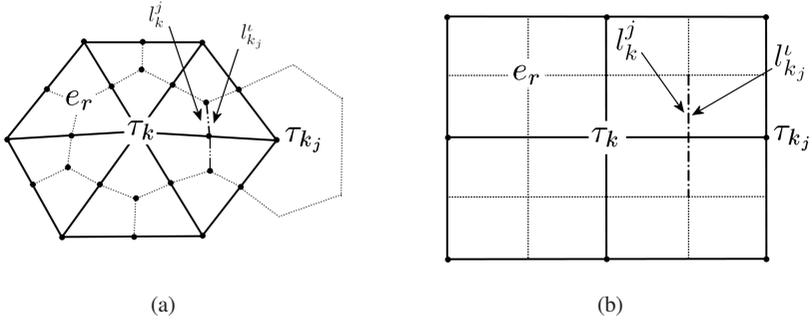


Fig. 4.2: An example of unstructured (a), and structured (b), two dimensional meshes with the dual meshes (dotted). The j -th neighboring cell of τ_k is τ_{k_j} , the common interface between τ_k and τ_{k_j} is called l_k^j . There exists a ι such that the ι -th interface of τ_{k_j} is equal to l_k^j .

us introduce the semi-discrete counterparts of λ_i and ϕ_i denoted by $\lambda_{i,\Delta}(t) \in \mathbb{V}^0$, $\phi_{i,\Delta}(t) \in \mathbb{V}^1$, respectively, where $\mathbb{V}^0 = \{\lambda \in L^2(\Omega) : \lambda|_{\tau_k} \in \mathbb{P}^0(\tau_k), k = 1, \dots, n_c\}$, $\mathbb{V}^1 = \{\phi \in \mathbb{C}^0(\Omega) : \phi|_{e_r} \in \mathbb{Q}^1(e_r), r = 0, \dots, n_c\}$ in the case of a rectangular grid and $\mathbb{V}^1 = \{\phi \in \mathbb{C}^0(\Omega) : \phi|_{e_r} \in \mathbb{P}^1(e_r), r = 0, \dots, n_c\}$ on a simplicial mesh. Here $\mathbb{P}^s(\Omega)$ denotes the space of polynomials of order at most s on Ω , and $\mathbb{Q}^s(\Omega)$ is the space of the tensor product of polynomials of order at most one. We consider the canonical basis $\{\vartheta_k^0\}$ for \mathbb{V}^0 and $\{\vartheta_k^1\}$ for \mathbb{V}^1 , therefore

$$\lambda_{i,\Delta}(t, \vec{X}) = \sum_{k=1}^{n_p} \lambda_{i,k}(t) \vartheta_k^0(\vec{X}), \quad \phi_{i,\Delta}(t, \vec{X}) = \sum_{k=0}^{n_c} \phi_{i,k}(t) \vartheta_k^1(\vec{X}), \quad (4.7)$$

where $\lambda_{i,k}$ is the mean volume fraction of the species i in the cell τ_k (we will denote, from now on, $\lambda_{i,k}$ as the composition) and $\phi_{i,k}$ are the values of the discrete level set function at node \vec{x}_k .

We introduce a rather simple coupling between LS and VT equations, by choosing the level set function as the piecewise linear interpolation from the dual mesh to the original one, i.e. $\phi_{i,\Delta} = \mathbf{I}_\Delta^1 \lambda_{i,\Delta}$ where $\mathbf{I}_\Delta^1 : \mathbb{V}^0 \rightarrow \mathbb{V}^1$ is the linear interpolation operator on the \mathcal{T}_Δ grid. In other terms we set

$$\phi_{i,k} = \lambda_{i,k} \quad k = 1, \dots, n_p, \quad i = 1, \dots, n_s. \quad (4.8)$$

The evolution of the interfaces is carried out by advancing the $\lambda_{i,\Delta}$ by a discrete version of (4.1), using the information carried by $\phi_{i,\Delta}$ to build the numerical fluxes. Finally we reconstruct the level set as a postprocessing. This choice implies an error concerning the representation of the initial conditions as, in general, $\lambda_{i,\Delta} \neq H(\phi_{i,\Delta} - \frac{1}{2})$. This difference can be bounded as we state in the following:

Proposition 4.3. *Let us assume that $\lambda \in \mathbb{V}^0$ and has the image in the set $\{1, 0\}$. And, moreover, let us consider $\phi = \mathbf{I}_\Delta^1 \lambda$. Then*

$$\int_{\Omega} \left(\lambda - H \left(\phi - \frac{1}{2} \right) \right) = O(\Delta).$$

Proof. Let $S_b = \{k \in [1, n_c] : \int_{\tau_k} (\lambda - H(\phi - \frac{1}{2})) \neq 0\}$. Since this set is also the set of the cells that are crossed by the boundary of Ω_i its cardinality is $O(\Delta^{1-d})$. Moreover $\int_{\tau_k} (\lambda - H(\phi - \frac{1}{2})) = O(\Delta^d) \forall k \in S_b$, therefore $\int_{\Omega} (\lambda - H(\phi - \frac{1}{2})) = O(\Delta^{1-d})O(\Delta^d)$ and we obtain the thesis.

We use a finite volume method together with an explicit Euler scheme to advance $\lambda_{i,\Delta}$

$$\lambda_{i,k}^{n+1} = (1 + D_{\Delta,k}^n) \lambda_{i,k}^n - \sum_{j=1}^{|\mathbb{C}_k^C|} F_{i,k}^{n,j}, \quad (4.9)$$

where $\lambda_{i,k}^n = \lambda_{i,k}(t^n)$ and $t^0, t^1, \dots, t^n, t^{n+1}$ is a sequence of time steps with $t^{n+1} = t^n + \Delta t^n$. The quantity $D_{\Delta,k}^n = \sum_{j=1}^{|\mathbb{C}_k^C|} v_k^{n,j}$ is the dimensionless discrete divergence factor of element τ_k (i.e. $D_{\Delta,k}^n$ is the discrete approximation of $\frac{\Delta t^n}{|\tau_k|} \oint_{\partial \tau_k} \vec{u} \cdot \vec{n}$) and

$$v_k^{n,j} = \frac{\Delta t^n}{|\tau_k|} \int_{l_k^j} \vec{u} \cdot \vec{n}$$

is a dimensionless quantity which can be considered as the interface Courant number. Finally

$$F_{i,k}^{n,j} = v_k^{n,j} \Phi(\widehat{\lambda}_{i,k}^{n,j}, \widehat{\lambda}_{i,k_j}^{n,l}) \quad (4.10)$$

are the interface fluxes, where $\Phi(\widehat{\lambda}_{i,k}^{n,j}, \widehat{\lambda}_{i,k_j}^{n,l})$ is the upwind function

$$\Phi(\widehat{\lambda}_{i,k}^{n,j}, \widehat{\lambda}_{i,k_j}^{n,l}) = \begin{cases} \widehat{\lambda}_{i,k}^{n,j} & \text{if } v_k^{n,j} \geq 0 \\ \widehat{\lambda}_{i,k_j}^{n,l} & \text{if } v_k^{n,j} < 0, \end{cases} \quad (4.11)$$

where $\widehat{\lambda}_{i,k}^{n,j}, \widehat{\lambda}_{i,k_j}^{n,l}$ are suitable approximations of the composition $\lambda_{i,\Delta}^n$ at the faces at l_k^j and $l_{k_j}^l$ respectively: see Figure 4.3

The stability of method (4.9) entails the following time step restriction,

$$\Delta t^n \leq \frac{|\tau_k|}{|\mathbb{I}_k^C|} \frac{1}{|\int_{j_k^j} \vec{u} \cdot \vec{n}|} \quad k = 1, \dots, n_c, \quad j = 1, \dots, |\mathbb{I}_k^C|, \quad (4.12)$$

that is

$$|v_k^{n,j}| \leq \frac{1}{|\mathbb{I}_k^C|} \quad k = 1, \dots, n_c, \quad j = 1, \dots, |\mathbb{I}_k^C|. \quad (4.13)$$

To define $\widehat{\lambda}_{i,k}^{n,j}$ we have used the following relation

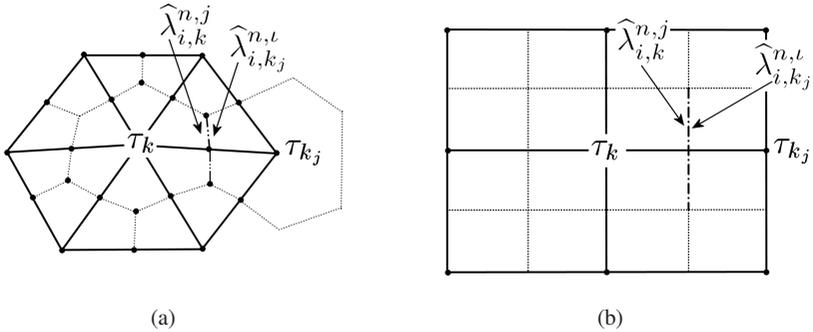


Fig. 4.3: An example of the boundary compositions $\widehat{\lambda}_{i,k}^{n,j}$ and $\widehat{\lambda}_{i,k_j}^{n,l}$ on an unstructured (a), and structured (b) grid. The first is an approximation of the composition from inside τ_k while the other is an approximation from the neighboring cell τ_{k_j} .

$$\widehat{\lambda}_{i,k}^{n,j} = \lambda_{i,k}^n + \delta \lambda_{i,k}^{n,j}, \quad (4.14)$$

where the variables $\delta \lambda_{i,k}^{n,j}$ are defined as the solution of the following constrained minimization problem to be solved for all $k = 1, \dots, n_c$ and for all $j \in \mathbb{I}_k$ where $\mathbb{I}_k = \left\{ j \in 1, \dots, |\mathbb{I}_k^C| : v_k^{n,j} \geq 0 \right\}$ is the set of the indices of the outflow faces l_k^j of the k -th cell

$$\begin{cases} \min_{\delta\lambda_{i,k}^{n,j}} \frac{1}{2} \sum_{i=1}^{n_s} (\lambda_{i,k}^n - \phi_{i,k}^{n,j} + \delta\lambda_{i,k}^{n,j})^2 \\ \sum_{i=1}^{n_s} \delta\lambda_{i,k}^{n,j} = 0; \\ \delta\lambda_{i,k,min}^{n,j} \leq \delta\lambda_{i,k}^{n,j} \leq \delta\lambda_{i,k,max}^{n,j}; \end{cases} \quad (4.15)$$

where $\phi_{i,k}^{n,j} = \frac{1}{|I_k^j|} \int_{I_k^j} \phi_{i,\Delta}(t^n, \vec{X})$ and

$$\begin{cases} \delta\lambda_{i,k,min}^{n,j} = -\lambda_{i,k}^n \\ \delta\lambda_{i,k,max}^{n,j} = \min \left(\frac{(1+D_{\Delta,k}^n) - v_k^{n,j} |\mathbb{J}_k|}{v_k^{n,j} |\mathbb{J}_k|} \lambda_{i,k}^n, 1 - \lambda_{i,k}^n \right). \end{cases} \quad (4.16)$$

We need also to build an algorithm to reinitalize the LS function. As we have dropped the usual definition of the distance function we need to define a proper algorithm for the reconstruction of the LS:

Algorithm 1 *If there is an index \bar{i} such that:*

$$\lambda_{\bar{i},k}^n > \frac{1}{2} \text{ and } \lambda_{i,k_j}^n > \frac{1}{2} \quad \forall k_j \in \mathbb{I}_k^C, \quad (4.17)$$

then we set $\lambda_{\bar{i},k}^n = 1$ and $\lambda_{i,k}^n = 0$ with $i = 1, \dots, n_s, i \neq \bar{i}$. Otherwise we maintain the nodal value $\lambda_{i,k}^n$. Then the level set function is updated using (4.8).

This algorithm doesn't modify the LS function in all the elements where $\phi_{i,\Delta}^n$ equals $\frac{1}{2}$: in other words, the interface position is not modified by this algorithm. Actually, if equation (4.17) is satisfied, from (4.8) we get $\tau_k \in \Omega_i$ and therefore we may set $\lambda_{\bar{i},k}^n = 1$. Since the evolution of the interfaces is independent of the set function (see [65], [81]) this algorithm doesn't introduce any error from the LS point of view. The Algorithm 1 does not guarantee the mass conservation in the sense that $\sum_{k=1}^{n_c} \lambda_{i,k}^n$ is not conserved. However we will see in the following Chapter that the mass discrepancy introduced by the reconstruction algorithm is small. Having concluded the definition of our method we devote the next section to its analysis.

4.3 Analysis of the method

Proposition 4.4. *If $0 \leq \lambda_{i,\Delta}^n \leq 1$ and if $v_k^{n,j} \geq 0$ (i.e. we are considering an outflow sub-cell) then problem (4.15) has a unique solution.*

Proof. First of all we will show that the feasible region for the $\delta\lambda_{i,k}^{n,j}$ defined by the constrains in problem (4.15) is a convex nonempty subset of \mathbb{R}^{n_s} . Since the first of (4.15) is a convex minimization problem we can conclude (see [66]) that the problem has a unique solution.

As we have stated previously, the feasible region is nonempty, in fact, we can bound all the terms in (4.16):

$$\delta\lambda_{i,k,min}^{n,j} = -\lambda_{i,k}^n \leq 0. \quad (4.18)$$

Moreover

$$1 - \lambda_{i,k}^n \geq 0. \quad (4.19)$$

Finally, since

$$D_{\Delta,k}^n = \sum_{j=1}^{|\mathbb{I}_k^C|} v_k^{n,j} \geq \sum_{j \notin \mathbb{J}_k} v_k^{n,j},$$

using (4.13) we get

$$D_{\Delta,k}^n \geq - \sum_{j \notin \mathbb{J}_k} \frac{1}{|\mathbb{I}_k^C|} \geq - \frac{|\mathbb{I}_k^C| - |\mathbb{J}_k|}{|\mathbb{I}_k^C|}.$$

We bound the second term in the second equation of (4.16)

$$\begin{aligned} \frac{(1 + D_{\Delta,k}^n) - v_k^{n,j} |\mathbb{J}_k|}{v_k^{n,j} |\mathbb{J}_k|} \lambda_{i,k}^n &\geq \frac{1 - \frac{|\mathbb{I}_k^C| - |\mathbb{J}_k|}{|\mathbb{I}_k^C|} - v_k^{n,j} |\mathbb{J}_k|}{v_k^{n,j} |\mathbb{J}_k|} \lambda_{i,k}^n = \\ &= \frac{|\mathbb{J}_k| - v_k^{n,j} |\mathbb{J}_k| |\mathbb{I}_k^C|}{v_k^{n,j} |\mathbb{J}_k| |\mathbb{I}_k^C|} \lambda_{i,k}^n = \frac{1 - v_k^{n,j} |\mathbb{I}_k^C|}{v_k^{n,j} |\mathbb{I}_k^C|} \lambda_{i,k}^n. \end{aligned}$$

Then from (4.13) we obtain

$$\frac{(1 + D_{\Delta,k}^n) - v_k^{n,j} |\mathbb{J}_k|}{v_k^{n,j} |\mathbb{J}_k|} \lambda_{i,k}^n \geq 0. \quad (4.20)$$

Combining (4.19) and (4.20) we have $\delta\lambda_{i,k,max}^{n,j} \geq 0$ and combining with (4.18) we get that the feasible set is not empty.

Let's now prove the convexity of the feasible region. Consider two vectors δ_i, ζ_i with $i = 1, \dots, n_s$ belonging to the feasible region. Let $\alpha\delta_i + (1 - \alpha)\zeta_i$ with $\alpha \in [0, 1]$ be a convex combination of the vectors. We have to show that the linear combination belongs to the feasible region. We have

$$\begin{cases} \sum_{i=1}^{n_s} (\alpha\delta_i + (1 - \alpha)\zeta_i) = \alpha \sum_{i=1}^{n_s} \delta_i + (1 - \alpha) \sum_{i=1}^{n_s} \zeta_i = 0 \\ (\alpha\delta_i + (1 - \alpha)\zeta_i) \leq \alpha \delta\lambda_{i,k,max}^{n,j} + (1 - \alpha) \delta\lambda_{i,k,max}^{n,j} = \delta\lambda_{i,k,max}^{n,j} \\ (\alpha\delta_i + (1 - \alpha)\zeta_i) \geq \alpha \delta\lambda_{i,k,min}^{n,j} + (1 - \alpha) \delta\lambda_{i,k,min}^{n,j} = \delta\lambda_{i,k,min}^{n,j}. \end{cases}$$

This completes the proof.

Proposition 4.5. *The method defined by (4.14), (4.15) (4.16) is positive, i.e. $\lambda_{i,k}^n \geq 0 \forall n i = 1, \dots, n_s k = 1, \dots, n_c$.*

Proof. Let's proceed by induction, we suppose $\lambda_{i,k}^n \geq 0$. Then using (4.9) we get

$$\lambda_{i,k}^{n+1} = (1 + D_{\Delta,k}^n) \lambda_{i,k}^n - \sum_{j=1}^{|\mathbb{I}_k^C|} F_{i,k}^{n,j} \geq (1 + D_{\Delta,k}^n) \lambda_{i,k}^n - \sum_{j \in \mathbb{J}_k} F_{i,k}^{n,j},$$

and combining with equations (4.10) and (4.14) we obtain

$$\begin{aligned} \lambda_{i,k}^{n+1} &\geq (1 + D_{\Delta,k}^n) \lambda_{i,k}^n - \sum_{j \in \mathbb{J}_k} v_k^{n,j} \left(\lambda_{i,k}^n + \delta \lambda_{i,k}^{n,j} \right) = \\ &\quad (1 + D_{\Delta,k}^n) \lambda_{i,k}^n - \lambda_{i,k}^n \sum_{j \in \mathbb{J}_k} v_k^{n,j} - \sum_{j \in \mathbb{J}_k} v_k^{n,j} \delta \lambda_{i,k}^{n,j}. \end{aligned}$$

Finally using (4.16) we have

$$\begin{aligned} \lambda_{i,k}^{n+1} &\geq (1 + D_{\Delta,k}^n) \lambda_{i,k}^n - \lambda_{i,k}^n \sum_{j \in \mathbb{J}_k} v_k^{n,j} - \sum_{j \in \mathbb{J}_k} \frac{(1 + D_{\Delta,k}^n) - v_k^{n,j} |\mathbb{J}_k|}{|\mathbb{J}_k|} \lambda_{i,k}^n = \\ &\quad (1 + D_{\Delta,k}^n) \lambda_{i,k}^n - \lambda_{i,k}^n \sum_{j \in \mathbb{J}_k} v_k^{n,j} - (1 + D_{\Delta,k}^n) \lambda_{i,k}^n + \lambda_{i,k}^n \sum_{j \in \mathbb{J}_k} v_k^{n,j} = 0, \end{aligned}$$

And, since $\lambda_{i,\Delta}^0 \geq 0$ the proof follows.

Proposition 4.6. *The sum of partial volumes on every cell at every time step equals one*

$$\sum_{i=1}^{n_s} \lambda_{i,k}^n = 1 \quad \forall n, k = 1, \dots, n_c. \quad (4.21)$$

Analogously the sum of the level set functions is everywhere equal to one

$$\sum_{i=1}^{n_s} \phi_{i,\Delta}^n = 1 \quad \forall n.$$

Proof. Let's use the induction principle. At $n = 0$, this condition is satisfied and we assume that the condition is satisfied at time t^n . We have

$$\begin{aligned} \sum_{i=1}^{n_s} \lambda_{i,k}^{n+1} &= \sum_{i=1}^{n_s} \left[(1 + D_{\Delta,k}^n) \lambda_{i,k}^n - \sum_{j=1}^{|\mathbb{I}_k^C|} F_{i,k}^{n,j} \right] = \\ &\quad (1 + D_{\Delta,k}^n) - \sum_{i=1}^{n_s} \sum_{j=1}^{|\mathbb{I}_k^C|} v_k^{n,j} \Phi(\widehat{\lambda}_{i,k}^{n,j}, \widehat{\lambda}_{i,k_j}^{n,l}). \quad (4.22) \end{aligned}$$

From the second equation of (4.9) and from the inductive hypothesis we have

$$\sum_{i=1}^{n_s} \widehat{\lambda}_{i,k}^{n,j} = \sum_{i=1}^{n_s} \lambda_{i,k}^n + \delta \lambda_{i,k}^n = 1,$$

and therefore

$$\sum_{i=1}^{n_s} \Phi(\widehat{\lambda}_{i,k}^{n,j}, \widehat{\lambda}_{i,k_j}^{n,t}) = 1. \quad (4.23)$$

Then plugging (4.23) into (4.22) we get

$$\sum_{i=1}^{n_s} \lambda_{i,k}^{n+1} = (1 + D_{\Delta,k}^n) - \sum_{j=1}^{|\mathbb{I}_k^C|} \mathbf{v}_k^{n,j},$$

Recalling that $D_{\Delta,k}^n = \sum_{j=1}^{|\mathbb{I}_k^C|} \mathbf{v}_k^{n,j}$ we obtain the first part of the thesis.

Finally, since $\phi_{i,\Delta}^n = \sum_{k=1}^{n_c} \lambda_{i,k}^n \vartheta_k^1$ we get

$$\sum_{i=1}^{n_s} \phi_{i,\Delta}^n = \sum_{i=1}^{n_s} \sum_{k=1}^{n_c} \lambda_{i,k}^n \vartheta_k^1 = \sum_{k=1}^{n_c} \vartheta_k^1 = 1.$$

We can also show a consistency result of the interface fluxes. Let $\psi : \lambda_{i,k}^n, \lambda_{i,k_j}^n, j = 1, \dots, |\mathbb{I}_k^C| \rightarrow \widehat{\lambda}_{i,k}^{n,j}$ be the map from the composition of the k -th cell $\lambda_{i,k}^n$ and from the compositions of its neighboring cells $\lambda_{i,k_j}^n, j = 1, \dots, |\mathbb{I}_k^C|$ to the j -th interface composition of the k -th cell i.e

$$\widehat{\lambda}_{i,k}^{n,j} = \psi \left(\lambda_{i,k}^n, \lambda_{i,k_1}^n, \dots, \lambda_{i,k_{|\mathbb{I}_k^C|}}^n \right)$$

Then we can show that:

Proposition 4.7. *If $\lambda_{i,k_j}^n = \lambda_{i,k}^n \quad j = 1, \dots, |\mathbb{I}_k^C|$ then*

$$\lambda_{i,k}^n = \psi(\lambda_{i,k}^n, \lambda_{i,k_1}^n, \dots, \lambda_{i,k_{|\mathbb{I}_k^C|}}^n)$$

Proof. In this particular case we have $\phi_{i,k}^{n,j} = \lambda_{i,k}^n \quad j = 1, \dots, |\mathbb{I}_k^C|$ and the optimal solution of (4.9) is $\delta \lambda_{i,k}^{n,j} = 0$. Actually this solution minimizes the objective function and clearly satisfies the equality constraint: in fact from Proposition 4.4 we also know that $\delta \lambda_{i,k}^{n,j} = 0$ is always in the feasible set.

Then from (4.10), (4.11) we get that the numerical flux is consistent too.

Finally we can also prove the following statement:

Proposition 4.8. *Every discrete subdomain does not overlap with the others, i.e*

$$\widetilde{\Omega}_{i,\Delta}(t) \cap \widetilde{\Omega}_{j,\Delta}(t) = \emptyset \quad \forall i = 1, \dots, n_s, \quad \forall j = 1, \dots, n_s, j \neq i \quad \forall t > 0$$

and given a subregion $\widetilde{\Omega}$ containing only two species identified by the indices i_1, i_2 , we have

$$\overline{\widetilde{\Omega}}_{i_1,\Delta} \cup \overline{\widetilde{\Omega}}_{i_2,\Delta} = \overline{\widetilde{\Omega}}.$$

Proof. If $\vec{X} \in \Omega_{i,\Delta}(t)$ then $\phi_{i,\Delta}(t, \vec{X}) > \frac{1}{2}$, from proposition (4.6) we get

$$\sum_{j=1, j \neq i}^{n_s} \phi_{j,\Delta}(t, \vec{X}) < \frac{1}{2}$$

and since the level set functions are the piecewise linear interpolation of a positive function (i.e. the volume fractions $\lambda_{i,\Delta}$) we get $\phi_{j,\Delta}(t, \vec{X}) < \frac{1}{2} \quad \forall j \neq i$ namely $\vec{X} \notin \Omega_{j,\Delta}(t) \quad \forall j \neq i$. In the special case in which in a subregion $\tilde{\Omega}$ there are only two species we get from the general case that there are no overlaps between the two subdomains $\tilde{\Omega}_{i_1,\Delta}, \tilde{\Omega}_{i_2,\Delta}$; we have only to prove that

$$\vec{X} \in \tilde{\Omega}_{i_1,\Delta} \text{ or } \vec{X} \in \tilde{\Omega}_{i_2,\Delta} \text{ or } \vec{X} \in \overline{\tilde{\Omega}}_{i_1,\Delta} \cap \overline{\tilde{\Omega}}_{i_2,\Delta} \quad \forall \vec{X} \in \tilde{\Omega}.$$

We consider the three cases,

1. $\phi_{i_1,\Delta}(t, \vec{X}) > \frac{1}{2}$, then $\vec{X} \in \tilde{\Omega}_{i_1,\Delta}$;
2. $\phi_{i_1,\Delta}(t, \vec{X}) < \frac{1}{2}$, so $\vec{X} \in \tilde{\Omega}_{i_2,\Delta}$, in fact: $\phi_{i_2,\Delta}(t, \vec{X}) = 1 - \phi_{i_1,\Delta}(t, \vec{X})$;
3. $\phi_{i_1,\Delta}(t, \vec{X}) = \frac{1}{2}$, consequently, the point $\vec{X} \in \overline{\tilde{\Omega}}_{i_1,\Delta} \cap \overline{\tilde{\Omega}}_{i_2,\Delta}$.

Therefore, we obtain the thesis.

4.4 Convergence analysis of the one dimensional case

In one dimension a more detailed analysis is possible. Let \mathcal{T}_Δ be a uniformly Δx -spaced 1D mesh (see Figure 4.4) with elements $e_0, \dots, e_{r-1}, e_r, e_{r+1}, \dots, e_{n_e}$ and let consider its dual mesh endowed with an ordered sequence of cells $\tau_0, \dots, \tau_{k-1}, \tau_k, \tau_{k+1}, \dots, \tau_{n_c}$. For the sake of simplicity let u be a constant, positive velocity field (i.e. we are treating a null divergence case), and let $v^n = \frac{\Delta t^n}{\Delta x} u$ be the Courant number. Notice that, in this case, all the Courant numbers are equivalent to v^n . Besides, every cell is associated to a mean composition $\lambda_{i,0}^n, \dots, \lambda_{i,k-1}^n, \lambda_{i,k}^n, \lambda_{i,k+1}^n, \dots, \lambda_{i,n_c}^n$ and has two boundary sub-cell compositions $\hat{\lambda}_{i,k}^{n,j}$ with $j = 1, 2$. Using a more explicit

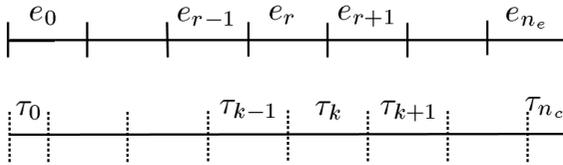


Fig. 4.4: The one dimensional mesh. In the first part are depicted the mesh and the elements e_r while in the second part the dual mesh is shown along with its cells τ_k .

notation, for the 1D case we can define the upwind subcell compositions as $\widehat{\lambda}_{i,k}^{n,+}$ and the downwind subcell as $\widehat{\lambda}_{i,k}^{n,-}$. Method (4.9) takes the form

$$\begin{cases} \lambda_{i,k}^{n+1} = \lambda_{i,k}^n - v(\widehat{\lambda}_{i,k}^{n,+} - \widehat{\lambda}_{i,k-1}^{n,+}) \\ \widehat{\lambda}_{i,k}^{n,+} = \lambda_{i,k}^n + \delta\lambda_{i,k}^{n,+}. \end{cases} \quad (4.24)$$

While the minimization problem (4.15) becomes

$$\begin{cases} \min_{\delta\lambda_{i,k}^{n,+}} \frac{1}{2} \sum_{i=1}^{n_s} \left(\frac{1}{2} (\lambda_{i,k}^n - \lambda_{i,k+1}^n) + \delta\lambda_{i,k}^{n,+} \right)^2 \\ \sum_{i=1}^{n_s} \delta\lambda_{i,k}^{n,+} = 0 \\ \delta\lambda_{i,k,\min}^{n,+} \leq \delta\lambda_{i,k}^{n,+} \leq \delta\lambda_{i,k,\max}^{n,+}, \end{cases} \quad (4.25)$$

where

$$\begin{cases} \delta\lambda_{i,k,\min}^{n,+} = -\lambda_{i,k}^n \\ \delta\lambda_{i,k,\max}^{n,+} = \min \left(\frac{(1+D_{\Delta,k}^n) - v|\mathbb{J}_k|}{v|\mathbb{J}_k|} \lambda_{i,k}^n, 1 - \lambda_{i,k}^n \right). \end{cases} \quad (4.26)$$

In the one dimensional case it is possible, using the modified equation technique, to carry out a convergence analysis. Let $U_i(t, \vec{X})$ be the modified solution such that:

$$\lambda_{i,k}^n = U_i(t^n, \vec{x}_k) \quad \forall n, i = 1, \dots, n_s, k = 1, \dots, n_c.$$

Moreover we suppose that $U_i \in \mathbb{C}^2(\mathbb{R}^+ \times \Omega)$ then

$$\begin{cases} \lambda_{i,k+1}^n = U_i(t^n, x_k) + \frac{\partial}{\partial x} U_i(t^n, x_k) \Delta x + O(\Delta x^2) \\ \lambda_{i,k-1}^n = U_i(t^n, x_k) - \frac{\partial}{\partial x} U_i(t^n, x_k) \Delta x + O(\Delta x^2) \\ \lambda_{i,k}^{n+1} = U_i(t^n, x_k) + \frac{\partial}{\partial t} U_i(t^n, x_k) \Delta t + O(\Delta t^2). \end{cases} \quad (4.27)$$

For simplicity we also denote:

$$\begin{aligned} U_{i,k}^n &= U_i(t^n, \vec{x}_k), & \frac{\partial U_{i,k}^n}{\partial x} &= \frac{\partial}{\partial x} U_i(t^n, \vec{x}_k), \\ & & \frac{\partial^2 U_{i,k}^n}{\partial x^2} &= \frac{\partial^2}{\partial x^2} U_i(t^n, \vec{x}_k), & \frac{\partial U_{i,k}^n}{\partial t} &= \frac{\partial}{\partial t} U_i(t^n, \vec{x}_k). \end{aligned}$$

Let's now attend to the convergence analysis:

Proposition 4.9. *The method defined by (4.25) is second order accurate in space and first order accurate in time.*

Proof. We first show that

$$\delta\lambda_{i,k}^{n,+} = \frac{1}{2} \frac{\partial U_{i,k}^n}{\partial x} \Delta x \quad (4.28)$$

is the optimal solution of problem (4.25) for $\Delta x \rightarrow 0$. Substituting (4.27) in the first of (4.25) we get

$$\frac{1}{2} \sum_{i=1}^{n_s} \left(\frac{1}{2} U_{i,k}^n - \frac{1}{2} U_{i,k}^n - \frac{1}{2} \frac{\partial U_{i,k}^n}{\partial x} \Delta x + O(\Delta x^2) + \frac{1}{2} \frac{\partial U_{i,k}^n}{\partial x} \Delta x \right) = n_s O(\Delta x^2).$$

Therefore the functional is minimized for $\Delta x \rightarrow 0$. It is also possible to prove that (4.28) satisfies the constraints of (4.25), in fact, plugging (4.28) in the second equation of (4.25) we get

$$\sum_{i=1}^{n_s} \delta \lambda_{i,k}^n = \sum_{i=1}^{n_s} \frac{1}{2} \frac{\partial U_{i,k}^n}{\partial x} \Delta x. \quad (4.29)$$

Then, summing up the components of the Taylor expansion $U_{i,k+1}^n = U_{i,k}^n + \frac{\partial}{\partial x} U_{i,k}^n \Delta x + O(\Delta x^2)$ and thanks to (4.21) we get

$$\sum_{i=1}^{n_s} \frac{\partial}{\partial x} U_{i,k}^n \Delta x = O(\Delta x^2). \quad (4.30)$$

Plugging (4.30) into (4.29) we obtain that (4.28) satisfies the first constraint of (4.25). Then substituting the Taylor expansions (4.27) into (4.26) we get

$$\begin{cases} \delta \lambda_{i,k,min}^{n,+} = -U_{i,k}^n \\ \delta \lambda_{i,k,max}^{n,+} = \min \left(\frac{(1+D_{\Delta,k}^n) - v^n |\mathbb{J}_k|}{v^n |\mathbb{J}_k|} U_{i,k}^n, 1 - U_{i,k}^n \right). \end{cases}$$

If Δx is small enough, the conditions above are equivalent to

$$-\frac{U_{i,k}^n}{\Delta x} \leq \frac{1}{2} \frac{\partial U_{i,k}^n}{\partial x} \leq \min \left(\frac{(1+D_{\Delta,k}^n) - v^n |\mathbb{J}_k|}{v^n |\mathbb{J}_k|} \frac{U_{i,k}^n}{\Delta x}, \frac{1 - U_{i,k}^n}{\Delta x} \right).$$

Consequently it follows that, for $\Delta x \rightarrow 0$, (4.28) is consistent with the second constraint of (4.25).

We can now estimate the convergence order of our method. Substituting (4.27) and (4.28) in the second of (4.24) we obtain

$$\begin{cases} \widehat{\lambda}_{i,k}^{n,+} = U_{i,k}^n + \frac{1}{2} \frac{\partial U_{i,k}^n}{\partial x} \Delta x \\ \widehat{\lambda}_{i,k-1}^{n,+} = U_{i,k-1}^n + \frac{1}{2} \frac{\partial U_{i,k-1}^n}{\partial x} \Delta x \end{cases}$$

and combining them with the first of (4.24) we get

$$\Delta t^n \frac{\partial U_{i,k}^n}{\partial t} + O((\Delta t^n)^2) = -\frac{\Delta t^n}{\Delta x} u (U_{i,k}^n - U_{i,k-1}^n + \frac{1}{2} \frac{\partial}{\partial x} U_{i,k}^n \Delta x - \frac{1}{2} \frac{\partial}{\partial x} U_{i,k-1}^n \Delta x). \quad (4.31)$$

From (4.27) we obtain

$$\begin{cases} U_{i,k}^n - U_{i,k-1}^n = -\frac{\partial U_{i,k}^n}{\partial x} \Delta x + \frac{1}{2} \frac{\partial^2 U_{i,k}^n}{\partial x^2} \Delta x^2 + O(\Delta x^3) \\ \frac{1}{2} \left(\frac{\partial U_{i,k}^n}{\partial x} - \frac{\partial U_{i,k-1}^n}{\partial x} \right) \Delta x = -\frac{1}{2} \frac{\partial^2 U_{i,k}^n}{\partial x^2} \Delta x^2 + O(\Delta x^3) \end{cases} \quad (4.32)$$

Plugging (4.32) into (4.31) and dividing by Δt^n , we obtain

$$\frac{\partial U_{i,k}^n}{\partial t} + u \frac{\partial U_{i,k}^n}{\partial x} = O(\Delta t^n) + O(\Delta x^2) \quad \forall n, i = 1, \dots, n_s, k = 1, \dots, n_c$$

and the proof follows.

4.5 Numerical results

4.6 Results

In this section we introduce some numerical experiments which aim to illustrate the quality of the numerical scheme proposed here. The first one is the convergence result in one dimension. In this case, in order to accommodate the boundary conditions, we have introduced a slight modification into the algorithm. In the description of this method we have, so far, neglected the boundary conditions since we wanted to focus on the properties of the method which are not dependent on them. We consider a test case with a constant transport speed i.e. $v = 1$ and the domain Ω is the interval $[0, 1]$. The initial conditions are $\lambda_{1,k}^0 = 1, \lambda_{2,k}^0 = 0 \quad \forall k$, while $\lambda_1^b(t) = 0, \lambda_2^b(t) = 1 \quad \forall t \in [0, T]$ are the boundary conditions on the left (inflow) side. The problem

$$\frac{\partial \lambda_i}{\partial t} + u \frac{\partial \lambda_i}{\partial x} = 0$$

has the following analytical solution:

$$\lambda_1(t, x) = H(x - ut), \quad \lambda_2(t, x) = 1 - H(x - ut).$$

Furthermore it is possible to compute the L^1 error on $(0, T) \times (0, 1)$ defined as:

$$E_{L^1} = \sum_{i=1}^{n_s} \int_0^T \int_a^b |\lambda_i(t, x) - \lambda_{i,\Delta}(t, x)|$$

In Figure 4.5 we show the L^1 error of the proposed method compared with a high resolution Discontinuous Galerkin (DG) method with a MinMod Limiter, see [25] and with the Godunov (G) method. Our method compares favorably with the DG method though the regularity of the solution limits the convergence rate. In fact, in this case, both our method and the DG one are only first order accurate.

Let's now consider some classical examples in two dimensions; in Figure 4.6a we outline some results obtained with a rotational field, $\vec{V}(\vec{X}) = [-X_2 - 1, X_1 - 1]$

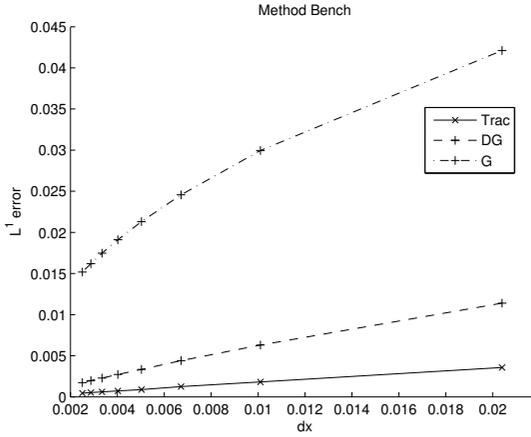


Fig. 4.5: One dimensional convergence of the proposed method (Trac) compared with a Discontinuous Galerkin (DG) method and with the Godunov (G) one.

where X_1, X_2 are the cartesian components of \vec{X} . A square is filled with a fluid tagged as A , the remaining space is filled with a fluid tagged as B ; the square lower left corner coordinates are $[0.8, 0.2]$ and the upper right corner coordinates are $[1.2, 0.6]$. If we compare with the comprehensive benchmark analysis performed

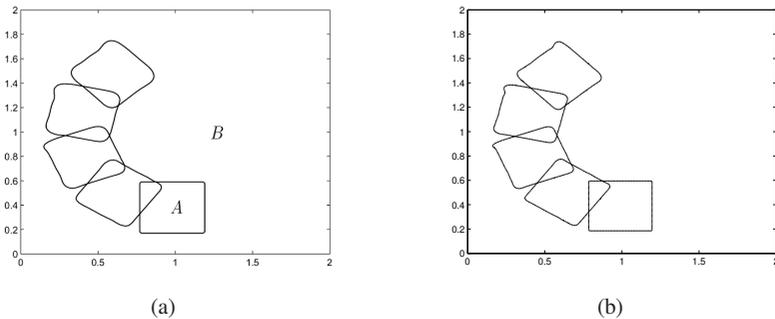


Fig. 4.6: Tracking of a square with (a) 10000 degrees of freedom and $cfl = \frac{1}{10}$ and (b) 40000 degrees of freedom and $cfl = \frac{1}{10}$.

in [30] we see that our results are intermediate nevertheless our method has the possibility to track a large number of fluids as we show in Figure 4.7. Here we consider the same case but with three fluids: the first, tagged as A , fills the inner square, the outer is filled with fluid B and the remaining space in the domain is filled with fluid

C , see Figure 4.7a. The inner square lower left corner and upper right coordinates are: $[0.9, 0.3]$, $[1.1, 0.5]$ respectively while the outer square corners coordinates are $[0.8, 0.2]$, $[1.2, 0.6]$. As we can see from Figure 4.7 the tracking performances are independent on the number of the species being tracked.

In Figure 4.8 we track three non-nested fluids showing the coherence between the

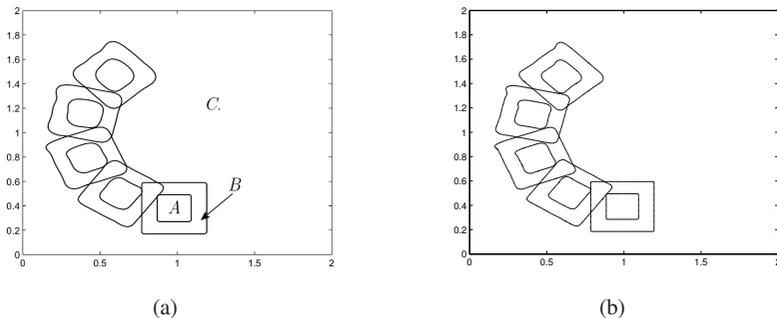


Fig. 4.7: Tracking of two nested squares with (a) 10000 degrees of freedom and $cfl = \frac{1}{10}$ and (b) 40000 degrees of freedom and $cfl = \frac{1}{10}$. In this case three species are involved: the inner square is filled with the species A , the outer with species B and the rest of the domain with species C .

three tracked interfaces; the small rectangle filled with fluid C has the following corner coordinates: $[0.75, 1]$, $[1.25, 1.45]$. The fluid B fills a more complex region, that is the complementary part of the rectangle C in a rectangle with corner coordinates $[0.5, 0.45]$, $[1.5, 1.45]$ respectively. The remaining part of the computational domain is filled with the fluid A . In Figure 4.9 the mass ratio between the volume occupied by the subdomains A and B and the total volume is shown. As we see, the volume is almost perfectly conserved.

In the next chapter we will present some three dimensional applications of our tracking method.

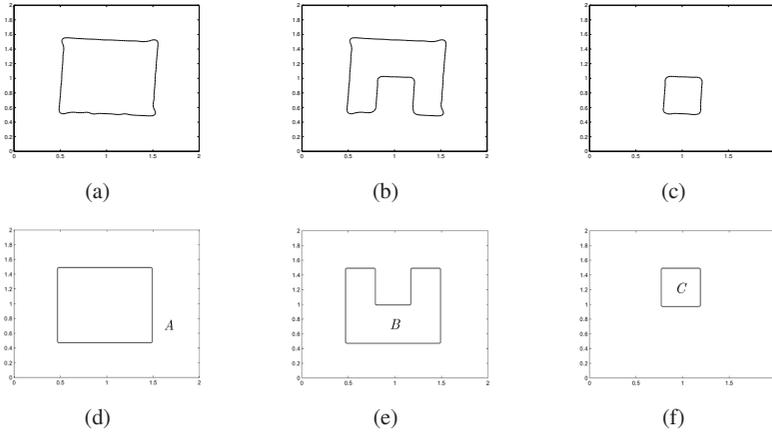


Fig. 4.8: Multi-fluid tracking using 10000 degrees of freedom and $cfl = \frac{1}{5}$. (a),(b),(c) are the computed interfaces after half a turn and (d),(e),(f) are the initial configurations.

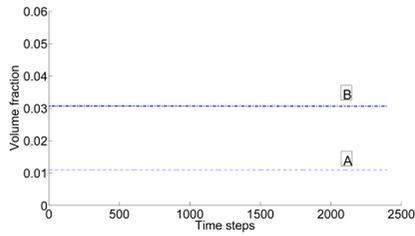


Fig. 4.9: The conservation of the mass fraction of species A (dashed line) and B (dash-dot line) plotted against the time steps.

4.7 Nomenclature

Index	Meaning	Interval
i	Layer index	$[1, n_s]$
k	Cell index	$[1, n_c]$
j	Face index	$[1, \mathbb{I}_k^C]$
r	Element index	$[1, n_e]$
n	Time step	-

Table 4.1: Table of the indexes.

Symbol	Meaning
Ω	The computational domain
$\partial\Omega$	The boundary of the computational domain
Ω_i	The i -th layer volume
\vec{X}	The position vector
\vec{n}	The normal to $\partial\Omega$
$\tilde{\Omega}_{i,\delta}$	The discrete i -th layer volume

Table 4.2: Table of the geometric quantities

Symbol	Meaning
\mathcal{T}_Δ	The mesh
Δ	The mesh characteristic length
n_e	Number of elements
n_p	Number of points
n_c	Number of cells
τ_k	The k -th cell
τ_k^j	The j -th neighboring cell of the k -th cell
e_r	The r -th element
l_k^j	The l -th interface of the k -th cell
\mathbb{I}_k^C	the set of indices of the neighboring cell of the k -th cell
\vec{x}_k	the k -th node of the mesh

Table 4.3: Table of the mesh quantities

Symbol	Meaning
λ_i	The characteristic function of Ω_i
ϕ_i	The level set function of Ω_i
λ_i^0	The characteristic function of Ω_i at $t = 0$
$\lambda_{i,\Delta}$	The discrete counterpart of λ_i
$\phi_{i,\Delta}$	The discrete counterpart of ϕ_i
$\widehat{\lambda}_{i,k}^{n,j}$	Interface compositions
$\Delta\phi_{i,k}^{n,j}$	The j -th interface, k -th cell variation of the level set
$\gamma_{i,k}^{n,j}$	The j -th interface, k -th cell Courant number
$\gamma_{i,k}^{n,j,max}$	The j -th interface, k -th cell Courant number upper bound

Table 4.4: Table of the variables

Symbol	Meaning
$\{\vartheta_k^0\}$	The space of the piece-wise constant functions on the cells
$\{\vartheta_k^1\}$	The space of the piece-wise linear functions on the elements
t	The time
t^n	The time at time-step n
Δt^n	The time step
Φ	The upwind function
\mathbb{J}_k	The set of the outflow interfaces of the k -th cell
$D_{\Delta,k}^n$	The sum of the courant numbers of the k -th cell
U_i	The modified solution
χ_{Ω_i}	The characteristic function of Ω_i
H	The Heaviside function

Table 4.5: Table of the other symbols

Chapter 5

Numerical models for basic basin simulations

In this Chapter we introduce the numerical solution for the mathematical model (3.51). The use of this simplified model is justified by the fact that, in many cases, the fluid pressure behaves as a hydrostatic one. Moreover if the rheology is not highly dependent on the temperature, the thermal effects can be decoupled from the computation of the basin evolution. In other terms, the temperature field can be computed a posteriori once the geometrical evolution of the basin is known. Moreover we consider some simplified boundary conditions and we postpone to the Chapter 6 the handling of more complex ones.

In this Chapter we describe the main features of a mathematical model able to simulate the evolution of salt sedimentary basins and its efficient implementation. In particular, we put together the models developed in the first chapter with the layer tracking techniques described in the second chapter. Combining these techniques with a Stokes solver we provide a numerical tool for the simulation of the gravitational Rayleigh–Taylor instability associated with salt diapirism. We postpone to the Chapter 6 the handling of more complex issues such as faults, non-Newtonian rheologies and sedimentation processes. As we have outlined in Chapter 3, the adopted model is composed of a series of layers, modeled as incompressible and immiscible fluids in Stokes regime. This choice, that is justified by the timescales of the phenomena which are comparable to the geological ages (as shown for example in [24, 94]), allows geologists to study the Rayleigh–Taylor instabilities associated with salt diapirism (as described in [55, 56, 79, 104, 107, 108]). To solve this kind of problems we choose a classical approach that divides the computation of pressure and velocity fields from interface tracking.

Moreover, to improve the efficiency of the code, we introduce an enhanced implementation of the tracking algorithm presented in Chapter 4 and a preconditioned iterative scheme for the Stokes solver. During the past few years, the finite element method has played an important role in the solution of the mass and force balance equations (as shown in [55, 57, 56, 60, 68, 79]), as it permits to solve the fluid dynamic problem in complex geometrical cases with high accuracy. For realistic three dimensional 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 this case the value of the viscosity of the sediments, modeled as fluids, varies over a range of about five orders of magnitude (as shown in [56, 79]), and this leads to a badly conditioned algebraic problem. Without a proper preconditioning, such a complex problem would require too many iterations to reach a reasonable tolerance. Here we propose a proper preconditioning technique in order to minimize the computational burden. As regards the tracking phase we exploit the scheme illustrated in Chapter 4.

5.1 Physical and mathematical models

Let's now recall briefly the geometric model of the sedimentary basin (Figure 5.1). 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 of 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 the velocity field with a Dirichlet condition, and the lateral contour Γ_L , that we suppose vertical for simplicity and where we have imposed a vertical no-stress condition. More general boundary conditions will be introduced in the following Chapter. In addition, the horizons between the subdomains are defined as $\Gamma_{i,j} = \Omega_i \cap \Omega_j$ and we define \vec{n} as the outward normal of domain Ω . Finally $\hat{x}_i, i = 1, 2, 3$ are the three unit

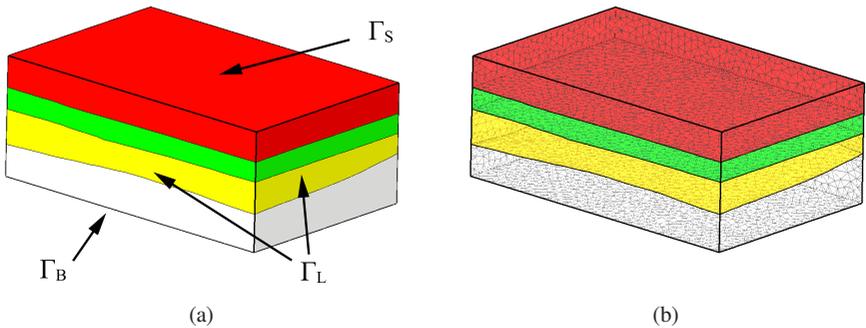


Fig. 5.1: a) External shape of the domain Ω . The external boundary Γ is divided into three parts: the basement Γ_B , the free surface Γ_S and the lateral contour Γ_L . b) An open three dimensional view of the sedimentary basin which contains three horizons and four layers.

vectors, $t \in (0, T]$ is the time coordinate, and $\vec{X} \in \Omega$, indicates a spatial point in the domain. The mathematical model (3.51) that describes the geological evolution of the basin, modeled as a stratified fluid, in which the layers are immiscible and have

constant properties reads

$$\left\{ \begin{array}{ll} \vec{\nabla} \cdot \vec{\sigma}(\mu) - \vec{\nabla} P + \rho \vec{g} = 0 & \text{in } \Omega \times [0, T] \\ \vec{\nabla} \cdot \vec{u} = 0 & \text{in } \Omega \times [0, T] \\ \frac{\partial \rho}{\partial t} + \vec{u} \cdot \vec{\nabla} \rho = 0 \quad \frac{\partial \mu}{\partial t} + \vec{u} \cdot \vec{\nabla} \mu = 0 & \text{in } \Omega \times (0, T] \\ \rho = \rho^0, \mu = \mu^0 & \text{in } \Omega \times \{0\} \\ \vec{u} = \vec{u} & \text{on } \Gamma_B \cup \Gamma_S \times [0, T] \\ u_1 = \bar{u}_1, u_2 = \bar{u}_2, (\vec{\sigma} \cdot \vec{n}) \cdot \hat{x}_3 = 0, & \text{on } \Gamma_L \times [0, T], \end{array} \right. \quad (5.1)$$

where the unknowns are the velocity and the pressure fields, \vec{u} and P , respectively and \vec{g} is the gravitational acceleration. Moreover \vec{u} is the velocity imposed on Γ_B and Γ_S , while on Γ_L only the \hat{x}_3 -orthogonal components are fixed through \bar{u}_1 and \bar{u}_2 . The functions μ^0 and ρ^0 are the initial conditions for the viscosity and the density fields respectively. As we have already seen in chapter 4, through the functions λ_i , it is possible to write the density and the dynamic viscosity fields as:

$$\rho = \sum_{i=1}^{n_s} \lambda_i \rho_i, \quad \mu = \sum_{i=1}^{n_s} \lambda_i \mu_i. \quad (5.2)$$

Setting $\lambda_i^0(\vec{X}) = \lambda(0, \vec{X})$ we have

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

We also assume, for now, a Newtonian law: $\vec{\sigma} = \mu(\vec{\nabla} \vec{u} + (\vec{\nabla} \vec{u})^T)$. This relation may seem not very representative of the rheological complexity of the sediments, however it is accepted in literature as a sound base model to study Rayleigh–Taylor instabilities associated with diapirism ([56, 106, 107]). Some more realistic rheological relations will be considered in the following Chapter.

5.2 Time discretization

To solve equation (5.1) we must perform a numerical approximation of the problem. First of all we introduce the following time discretization in the interval $[0, T]$: $[0, t^1, t^2, \dots, T]$ with $\Delta t^n = t^{n+1} - t^n$. Then, for a generical variable $a(t, \vec{X})$, we also set $a^n(\vec{X}) = a(t^n, \vec{X})$. In order to solve the numerical problem we have implemented the splitting algorithm presented in [56], which reads

1 step: knowing ρ^n and μ^n , solve the following Stokes problem for \vec{u}^n and P^n

$$\begin{cases} \vec{\nabla} \cdot \vec{\sigma}^n - \vec{\nabla} P^n + \rho^n \vec{g} = 0 & \text{in } \Omega \\ \vec{\nabla} \cdot \vec{u}^n = 0 & \text{in } \Omega \\ \vec{u}^n = \vec{u} & \text{on } \Gamma_B \cup \Gamma_S \\ u_1^n = \bar{u}_1 \quad u_2^n = \bar{u}_2, (\vec{\sigma}^n \cdot \vec{n}) \cdot \hat{z} = 0 & \text{on } \Gamma_L. \end{cases} \quad (5.3)$$

2 step: having computed \vec{u}^n and P^n , the quantities ρ^{n+1} and μ^{n+1} can be computed solving the following hyperbolic equations:

$$\frac{\partial \rho}{\partial t} + \vec{u}^n \cdot \vec{\nabla} \rho = 0, \quad \frac{\partial \mu}{\partial t} + \vec{u}^n \cdot \vec{\nabla} \mu = 0, \quad (5.4)$$

in $(t^n, t^{n+1}] \times \Omega$.

The first step of the algorithm is a classic Stokes problem that can be solved efficiently with a finite element method. The second step is equivalent (see (3.44)) to a set of transport equations for λ_i i.e.:

$$\begin{cases} \frac{\partial \lambda_i}{\partial t} + \vec{u}^n \cdot \vec{\nabla} \lambda_i = 0 & \text{in } \Omega \times (t^n, t^{n+1}] \\ \lambda_i(t^n, \vec{X}) = \lambda^n(\vec{X}) & \text{in } \Omega \end{cases}$$

After providing the basic description of the splitting algorithm, in the next two sections we will analyze one by one both the steps of this scheme.

5.3 Velocity field solver

In this section we discuss the difficulties related to the solution of the Stokes problem. Its finite element formulation is equivalent to an algebraic system of equations and for realistic 3D simulations we have a large system, typically sparse. The strong variability of the viscosity coefficient affects the conditioning of the finite element matrix and imposes the use of a preconditioned iterative method.

We first define the discrete problem where we have adopted a conformal finite element scheme based on the mini-elements [73]. Then we describe a proper preconditioning technique.

Let \mathcal{T}_Δ^G be a simplicial grid that approximates Ω , let be n_e^G the number of elements e_r^G with $r = 1, \dots, n_e^G$ and let n_p^G be the number of nodes \vec{x}_k^G , with $k = 1, \dots, n_p^G$. We define Δ the maximum diameter of the elements and \mathcal{T}_Δ^M the mini-elements grid which has $n_e^M = 4n_e^G$ elements e_r^M , obtained by adding n_e^G barycentric nodes \vec{X}^b (see Figure 5.2) and $n_p^M = n_p^G + n_e^G$ nodes \vec{x}_k^M . We define $\vec{u}_\Delta^n \in \mathbb{V}_1^M$ and $P_\Delta^n \in \mathbb{V}_1^G$ as the approximations of \vec{u}^n and P^n where:

$$\begin{cases} \mathbb{V}_1^M = \left\{ \vec{\varphi}_\Delta \in \mathbb{C}^0(\Omega) : \vec{\varphi}_\Delta|_{e_r^M} \in \mathbb{P}^1 \quad \forall r = 1, \dots, n_e^M \right\} \\ \mathbb{V}_1^G = \left\{ \vec{\varphi}_\Delta \in \mathbb{C}^0(\Omega) : \vec{\varphi}_\Delta|_{e_r^G} \in \mathbb{P}^1 \quad \forall r = 1, \dots, n_e^G \right\}. \end{cases} \quad (5.5)$$

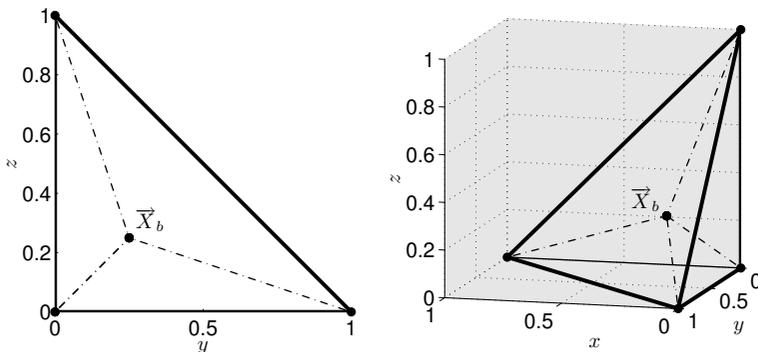


Fig. 5.2: A sketch of the \mathcal{T}_Δ^M grid obtained by the refinement of the \mathcal{T}_Δ^G grid. An element e_r^G is depicted with a solid line while the four elements e_r^M are depicted with a dashed line. The latter are obtained by adding a barycentric node inside the grid element and connecting it to the vertices of e_r^G .

\mathbb{V}_1^M and \mathbb{V}_1^G are intended as subspaces of the couple H_0^1 and L_0^2 , in which the continuous solution of the Stokes problem is searched. We introduce the weak formulation of problem (5.3):

find $\vec{u}_\Delta^n \in H_0^1(\Omega)$, $P_\Delta^n \in L_0^2(\Omega)$ such that

$$\begin{cases} a(\vec{u}_\Delta^n, \vec{v}_\Delta) + b(P_\Delta^n, \vec{v}_\Delta) = F(\vec{v}_\Delta) & \forall \vec{v}_\Delta \in \mathbb{V}_1^M \\ b(q_\Delta, \vec{u}_\Delta^n) = 0 & \forall q_\Delta \in \mathbb{V}_1^G, \end{cases} \quad (5.6)$$

where

$$\begin{aligned} a(\vec{u}_\Delta^n, \vec{v}_\Delta) &= - \int \mu (\vec{\nabla} \vec{u}_\Delta^n + (\vec{\nabla} \vec{u}_\Delta^n)^T) : (\vec{\nabla} \vec{v}_\Delta) \\ b(P_\Delta^n, \vec{v}_\Delta) &= \int \vec{P}_\Delta^n (\vec{\nabla} \cdot \vec{v}_\Delta) \\ F(\vec{v}_\Delta) &= - \int \rho_\Delta (\vec{g} \cdot \vec{v}_\Delta). \end{aligned}$$

and equation (5.6) is the discretization of the Stokes problem using the mini-elements. Let $\mu > 0$, $\mu \in L^\infty$, consequently the bilinear form $a(\cdot, \cdot)$ is coercive and the bilinear form $b(\cdot, \cdot)$ satisfies the inf-sup condition

$$\inf_{q \in L_0^2(\Omega), q \neq 0} \sup_{\vec{v} \in H_0^1(\Omega), \vec{v} \neq 0} \frac{b(q, \vec{v})}{|\vec{v}|_{H^1(\Omega)} \|q\|_{L^2(\Omega)}} \geq \beta, \quad (5.7)$$

for a $\beta > 0$. Furthermore, the finite element couple we have chosen satisfies the discrete inf-sup condition (see [32])

$$\min_{q_\Delta \in \mathbb{V}_1^G, q_\Delta \neq 0} \max_{\vec{v}_\Delta \in \mathbb{V}_1^M, \vec{v}_\Delta \neq 0} \frac{b(q_\Delta, \vec{v}_\Delta)}{|\vec{v}_\Delta|_{H^1(\Omega)} \|q_\Delta\|_{L^2(\Omega)}} \geq \beta_\Delta, \quad (5.8)$$

for a $\beta_\Delta > 0$. Finally, assuming $\rho \in L^\infty$, the functional $F(\cdot)$ is continuous. Thus, problem (5.6) has a unique solution (see [73]). Let $\{\varphi_k^M\} \in \mathbb{V}_1^M$ and $\{\varphi_k^G\} \in \mathbb{V}_1^G$ be the Lagrangian basis defined, respectively, on the grids \mathcal{T}_Δ^M and \mathcal{T}_Δ^G , then the discrete solution can be expanded as

$$\vec{u}_\Delta^n = \sum_{k=1}^{n_p^M} \vec{u}_k^n \varphi_k^M, \quad P_\Delta^n = \sum_{k=1}^{n_p^G} P_k^n \varphi_k^G. \quad (5.9)$$

Substituting these formulae in (5.6) we get the following algebraic problem:

$$\begin{bmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{V} \\ \mathbf{P} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_v \\ \mathbf{0} \end{bmatrix}, \quad (5.10)$$

where:

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

\mathbf{A} is a positive definite matrix and \mathbf{V} , \mathbf{P} are the vectors of the degrees of freedom. We solve this algebraic system by an algebraic fractional step scheme. First we compute \mathbf{P} by solving the pressure Schur complement

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

This problem is solved using a nested cycle iterative system to avoid an explicit inversion of matrix \mathbf{A} . Let's now analyze the conditioning of \mathbf{A} and of the Schur complement $\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T$. To this aim it is useful to recall some standard results. Let $\vec{v}_\Delta \in \mathbb{V}_1^M$, $q_\Delta \in \mathbb{V}_1^G$ and let $D\vec{u} = \vec{v}_\Delta \vec{u}_\Delta + (\vec{v}_\Delta \vec{u}_\Delta)^T$. The Korn inequality (see [31]) gives:

$$\|\vec{v}_\Delta \vec{u}_\Delta\|_{L^2(\Omega)} \leq C_1 \|D\vec{u}_\Delta\|_{L^2(\Omega)}. \quad (5.11)$$

Moreover we have

$$\|D\vec{u}_\Delta\|_{L^2(\Omega)} \leq C_2 \|\vec{v}_\Delta \vec{u}_\Delta\|_{L^2(\Omega)} \quad (5.12)$$

and (see [32])

$$\|\vec{v}_\Delta \cdot \vec{u}_\Delta\|_{L^2(\Omega)} \leq C_3 \|\vec{v}_\Delta \vec{u}_\Delta\|_{L^2(\Omega)}. \quad (5.13)$$

The inverse inequality (see [33]) leads to

$$\|\vec{v}_\Delta \vec{u}_\Delta\|_{L^2(\Omega)} \leq \frac{C_4}{\Delta} \|\vec{u}_\Delta\|_{L^2(\Omega)} \quad (5.14)$$

and finally using the Poincar inequality we get (see [33])

$$\|\vec{u}_\Delta\|_{L^2(\Omega)} \leq C_p \|\vec{v}_\Delta \vec{u}_\Delta\|_{L^2(\Omega)} \quad (5.15)$$

Let's now estimate the conditioning number of the Schur complement:

Proposition 5.1. *The conditioning number of the Schur complement $\mathcal{K}(\mathbf{BA}^{-1}\mathbf{B}^T)$ does not depend on Δ , more explicitly:*

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

Proof. Combining the discrete inf-sup condition (5.8) and equation (5.12) we get

$$\min_{q_\Delta \in \mathbb{V}_1^G} \max_{\vec{v}_\Delta \in \mathbb{V}_1^M} \frac{|(q_\Delta, \vec{\nabla} \cdot \vec{u}_\Delta)|}{\|D \vec{u}_\Delta\|_{L^2(\Omega)} \|q_\Delta\|_{L^2(\Omega)}} \geq \frac{\beta_\Delta}{C_2} \quad (5.16)$$

Furthermore

$$\|D \vec{u}_\Delta\|_{L^2(\Omega)} = \left\| \frac{1}{\sqrt{\mu}} \sqrt{\mu} D \vec{u}_\Delta \right\|_{L^2(\Omega)} \geq \frac{1}{\max(\sqrt{\mu})} \|\sqrt{\mu} D \vec{u}_\Delta\|_{L^2(\Omega)}. \quad (5.17)$$

Plugging equation (5.17) into (5.16) we get

$$\min_{q_\Delta \in \mathbb{V}_1^G} \max_{\vec{v}_\Delta \in \mathbb{V}_1^M} \frac{|(q_\Delta, \vec{\nabla} \cdot \vec{u}_\Delta)|}{\|\sqrt{\mu} D \vec{u}_\Delta\|_{L^2(\Omega)} \|q_\Delta\|_{L^2(\Omega)}} \geq \frac{1}{\sup(\sqrt{\mu})} \frac{\beta_\Delta}{C_2}. \quad (5.18)$$

Using the expansions (5.9) we have that

$$(q_\Delta, \vec{\nabla} \cdot \vec{u}_\Delta) = \mathbf{P}^T \mathbf{B} \mathbf{V}, \quad \|q_\Delta\|_{L^2(\Omega)} = \sqrt{\mathbf{P}^T \mathbf{Q} \mathbf{P}},$$

$$\|\sqrt{\mu} D \vec{u}_\Delta\|_{L^2(\Omega)} = \sqrt{\mathbf{V}^T \mathbf{A} \mathbf{V}},$$

where \mathbf{Q} is the mass matrix of the pressure field. Plugging this into (5.18) we get

$$\min_{\mathbf{P} \neq 1} \max_{\mathbf{V} \neq 0} \frac{|(\mathbf{P}^T \mathbf{B} \mathbf{V})|}{\sqrt{\mathbf{V}^T \mathbf{A} \mathbf{V}} \sqrt{\mathbf{P}^T \mathbf{Q} \mathbf{P}}} \geq \frac{1}{\sup(\sqrt{\mu})} \frac{\beta_\Delta}{C_2}. \quad (5.19)$$

Since the maximum is attained for $\mathbf{W} = \mathbf{A}^{1/2} \mathbf{V}$ (see [32]) we obtain

$$\frac{1}{\sup(\sqrt{\mu})} \frac{\beta_\Delta}{C_2} \leq \min_{\mathbf{P} \neq 1} \min_{\mathbf{P}} \max_{\mathbf{W}} \frac{|\mathbf{P}^T \mathbf{B} \mathbf{A}^{-1/2} \mathbf{W}|}{\sqrt{\mathbf{W}^T \mathbf{W}} \sqrt{\mathbf{P}^T \mathbf{Q} \mathbf{P}}}. \quad (5.20)$$

This equation implies $\mathbf{W} = \mathbf{A}^{-1/2} \mathbf{B}^T \mathbf{P}$ and plugged into (5.20) yields

$$\frac{1}{\max(\sqrt{\mu})} \frac{\beta_\Delta}{C_2} \leq \frac{\sqrt{|\mathbf{P}^T \mathbf{B} \mathbf{A}^{-1} \mathbf{B}^T \mathbf{P}|}}{\sqrt{\mathbf{P}^T \mathbf{Q} \mathbf{P}}}. \quad (5.21)$$

This completes the lower bound estimate. As regards the upper bound, combining (5.13), (5.11) with the Cauchy-Schwartz inequality, we have that $\forall \vec{u}_\Delta \in \mathbb{V}_1^M, \forall p_\Delta \in$

\mathbb{V}_1^G

$$\begin{aligned}
|(q_\Delta, \vec{\nabla} \cdot \vec{u}_\Delta)| &\leq \|q_\Delta\|_{L^2(\Omega)} \|\vec{\nabla} \cdot \vec{u}_\Delta\|_{L^2(\Omega)} \\
&\leq C_1 C_3 \|q_\Delta\|_{L^2(\Omega)} \|D \vec{u}_\Delta\|_{L^2(\Omega)} \\
&\leq \frac{C_1 C_3}{\inf(\sqrt{\mu})} \|q_\Delta\|_{L^2(\Omega)} \|\sqrt{\mu} D \vec{u}_\Delta\|_{L^2(\Omega)},
\end{aligned}$$

or the equivalent form

$$\frac{|(q_\Delta, \vec{\nabla} \cdot \vec{u}_\Delta)|}{\|q_\Delta\|_{L^2(\Omega)} \|\sqrt{\mu} D \vec{u}_\Delta\|_{L^2(\Omega)}} \leq \frac{C_1 C_3}{\min(\sqrt{\mu})}.$$

Applying the same argument used for the lower bound we get

$$\frac{\sqrt{|\mathbf{P}^T \mathbf{B} \mathbf{A}^{-1} \mathbf{B}^T \mathbf{P}|}}{\sqrt{\mathbf{P}^T \mathbf{Q} \mathbf{P}}} \leq \frac{C_1 C_3}{\min(\sqrt{\mu})}. \quad (5.22)$$

Then, combining the lower bound (5.21) and the upper bound (5.22) we obtain

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

that, multiplied by $\frac{\mathbf{P}^T \mathbf{Q} \mathbf{P}}{\mathbf{P}^T \mathbf{P}}$ leads to

$$\frac{1}{\max(\mu)} \frac{\beta_\Delta^2}{C_2^2} \frac{\mathbf{P}^T \mathbf{Q} \mathbf{P}}{\mathbf{P}^T \mathbf{P}} \leq \frac{|\mathbf{P}^T \mathbf{B} \mathbf{A}^{-1} \mathbf{B}^T \mathbf{P}|}{\mathbf{P}^T \mathbf{P}} \leq \frac{C_1^2 C_3^2}{\min(\mu)} \frac{\mathbf{P}^T \mathbf{Q} \mathbf{P}}{\mathbf{P}^T \mathbf{P}}. \quad (5.23)$$

Since (see [32]) we have

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

we obtain

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

And we get the proof. ■

We now perform the analysis of the \mathbf{A} matrix. Since (see [32])

$$\Delta^2 C_7 \leq \frac{\mathbf{V}^T \mathbf{Q}_V \mathbf{V}}{\mathbf{V}^T \mathbf{V}} \leq C_8 \Delta^2, \quad (5.24)$$

we have:

Proposition 5.2. *The conditioning number of \mathbf{A} is bounded by:*

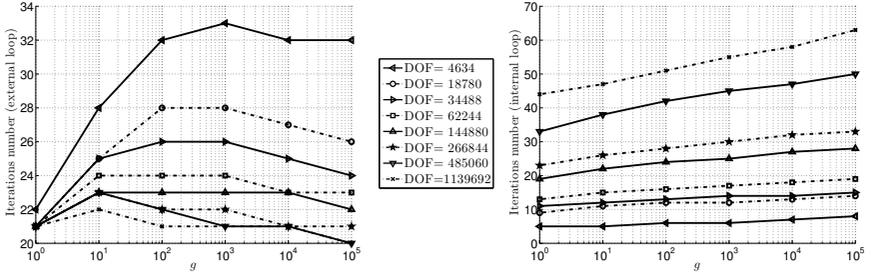


Fig. 5.3: On the left we show the number of iterations of the external cycle for different viscosity jumps $g = \max(\mu)/\min(\mu)$: the dependence on g or on Δ is not so strong. On the right we show the inner cycle iterations: there is an acceptable rise in the number of iterations.

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

Proof. Using equations (5.12) and (5.14) we have

$$\begin{aligned} \|\sqrt{\mu} D \vec{u}_\Delta\|_{L^2(\Omega)} &\leq \max(\sqrt{\mu}) \|D \vec{u}_\Delta\|_{L^2(\Omega)} \\ &\leq C_2 \max(\sqrt{\mu}) \|\vec{\nabla} \vec{u}_\Delta\|_{L^2(\Omega)} \leq \frac{C_2 C_4}{\Delta} \max(\sqrt{\mu}) \|\vec{u}_\Delta\|_{L^2(\Omega)}. \end{aligned} \quad (5.25)$$

Using (5.11) and (5.15) we can also estimate a lower bound for the symmetric gradient

$$\begin{aligned} \|\sqrt{\mu} D \vec{u}_\Delta\|_{L^2(\Omega)} &\geq \min(\sqrt{\mu}) \|D \vec{u}_\Delta\|_{L^2(\Omega)} \\ &\geq \frac{\min(\sqrt{\mu})}{C_1} \|\vec{\nabla} \vec{u}_\Delta\|_{L^2(\Omega)} \geq \frac{\min(\sqrt{\mu})}{C_1 C_p} \|\vec{u}_\Delta\|_{L^2(\Omega)}. \end{aligned} \quad (5.26)$$

Combining (5.25) and (5.26) we get

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

and from (5.9) we obtain

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

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

$$\frac{\mathbf{V}^T \mathbf{Q}_V \mathbf{V}}{\mathbf{V}^T \mathbf{V}} \frac{\min(\mu)}{C_1^2 C_p^2} \leq \frac{(\mathbf{V}^T \mathbf{A} \mathbf{V})}{(\mathbf{V}^T \mathbf{V})} \leq \frac{C_2^2 C_4^2}{\Delta^2} \max(\mu) \frac{\mathbf{V}^T \mathbf{Q}_V \mathbf{V}}{\mathbf{V}^T \mathbf{V}}. \quad (5.27)$$

Then from (5.27) and (5.24) we obtain:

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

and we get the proof. ■

These results show that the conditioning number of the two linear systems we wish to solve is high. In particular, in the inner cycle, where \mathbf{A} is used, the matrix conditioning is affected by both the grid spacing and the viscosity jumps across the interfaces. However, in this case, we can use several standard techniques such as, for instance, the incomplete LU factorizations. The outer cycle, characterized by the Schur complement, has a relatively better conditioning number but it is still dependent on the jumps of the viscosity coefficient. Moreover we can not use standard

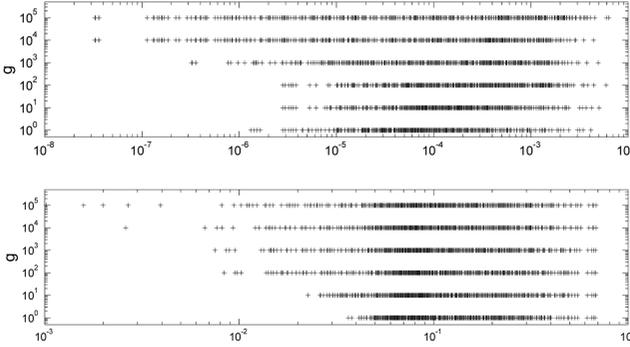


Fig. 5.4: The spectra of the Schur complement (in the upper part) with 4634 dofs for different viscosity jumps g . In the lower part we show the preconditioned spectra.

preconditioning techniques since they require an explicit assembling of the Schur complement and this is not affordable from a computational point of view. Therefore we look for a spectral equivalent to the Schur complement, and, to this end, we use a scaled mass matrix:

$$\mathbf{M}_{ij} = \int_{\Omega} \frac{1}{\mu} \phi_i^G \phi_j^G. \quad (5.28)$$

We are able only to give a rough explanation about the reasons that make this preconditioning matrix suitable to our purpose since a complete analysis on the spectral equivalence of the Schur complement with the matrix (5.28) is still missing. The Schur complement is the product of the matrix \mathbf{B} , that represents the divergence operator, the matrix \mathbf{A}^{-1} , that is the discrete representation of the inverse of

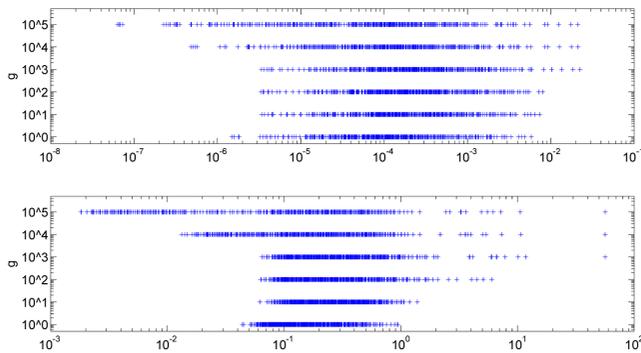


Fig. 5.5: The spectra of the Schur complement (in the upper part) with 4634 dofs for different viscosity jumps g . In the lower part we show the preconditioned spectra using non-scaled mass matrix.

the Laplace operator, and the matrix \mathbf{B}^T , that represents the gradient. Since the application of the gradient to the divergence yields the Laplace operator, the Schur complement resembles the identity operator. In our case the matrix \mathbf{A}^{-1} represents $\frac{1}{\mu}(\nabla^2)^{-1}$, therefore we expect that the Schur complement is spectrally similar to the mass matrix, apart from the $\frac{1}{\mu}$ factor. The numerical results in Figure 5.3 show that the preconditioner is effective and has a better behavior when the grid spacing is reduced. In Figure 5.4 the spectra of the Schur complement and of the preconditioned Schur complement are shown. We can see that the preconditioning matrix affects positively the ratio between the maximum and minimum eigenvalue. However there is still a dependence on the viscosity jump g . A theoretical result regarding the preconditioner performances is still missing and the research in this field is ongoing. However, as we can see in Figure 5.5, the use of a classical (not-scaled) mass matrix lead to worst results.

The linear system related to the \mathbf{A} matrix is solved with a preconditioned Krylov method. We use an incomplete LU factorization that has good performances even when the viscosity has a strong variability. However the incomplete factorization techniques are not optimal and some parameters such as the drop-off tolerance have to be adjusted case by case.

5.4 Tracking

In this section we study an efficient implementation of a numerical solver for problem (4.15) and we also give an estimate of its computational cost. In particular we

use a gradient method slightly adapted in order to solve this particular problem. We pass to the solver the following inputs for every outflow boundary of every cell:

- the cell composition, λ_i ;
- the vector ϕ_i representing the mean value of the level set function on each outflow sub-cell τ_k^j ;
- D_Δ , the divergence factor;
- ν , the Courant interface number;
- $|\mathbb{J}|$, the number of the outflow interfaces.

Here we have dropped the indices k, j as we illustrate the algorithm applied to a generical cell and interface. We use the following scheme:

Algorithm 2 *We compute the following quantities*

$$\begin{cases} \delta\lambda_{i,max} = \min\left(\frac{(1+D_\Delta)-\nu|\mathbb{J}|}{\nu|\mathbb{J}|}\lambda_i, 1-\lambda_i\right) \\ \delta\lambda_{i,min} = -\lambda_i. \end{cases} \quad (5.29)$$

and we set $\delta\lambda_i^{(0)} = 0, i = 1, \dots, n_s$. Then for $m = 0, 1, \dots$ we compute

$$\begin{cases} N_i^{(m)} = \begin{cases} 0 & \text{if } \delta\lambda_i^{(m)} = \delta\lambda_{i,min} \text{ and } (\lambda_i - \phi_i + \delta\lambda_i^{(m)}) > 0 \\ 0 & \text{if } \delta\lambda_i^{(m)} = \delta\lambda_{i,max} \text{ and } (\lambda_i - \phi_i + \delta\lambda_i^{(m)}) < 0 \\ 1 & \text{otherwise,} \end{cases} \\ \tilde{G}_i^{(m)} = (\phi_i - \lambda_i - \delta\lambda_i^{(m)})N_i^{(m)}, \\ G_i^{(m)} = \tilde{G}_i^{(m)} - \frac{\sum_{j=1}^{n_s} \tilde{G}_j^{(m)} N_j^{(m)}}{\sum_{j=1}^{n_s} (N_j^{(m)})^2} N_i^{(m)}, \\ \alpha = \begin{cases} \min\left(1, \frac{\delta\lambda_{i,max} - \delta\lambda_i^{(m)}}{G_i^{(m)}}\right) & \text{if } G_i^{(m)} \geq 0 \\ \min\left(1, \frac{\delta\lambda_{i,min} - \delta\lambda_i^{(m)}}{G_i^{(m)}}\right) & \text{if } G_i^{(m)} < 0, \end{cases} \\ \delta\lambda_i^{(m+1)} = \alpha G_i^{(m)} + \delta\lambda_i^{(m)}. \end{cases} \quad (5.30)$$

If $G_i^{(m)} = 0, i = 1, \dots, n_s$ we stop the iterations.

We give a brief overview of the Algorithm above: $N_i^{(m)}$ is zero if the corresponding component is constrained. We define $F_S^{(m)} = \{i \in \mathbb{R}^{n_s} : N_i^{(m)} = 1\}$ as the set of the active components and, in the same way, we define $NF_S^{(m)} = \{i \in \mathbb{R}^{n_s} : N_i^{(m)} = 0\}$ the set of the constrained components. The $\tilde{G}_i^{(m)}$ vector is the gradient of the objective function multiplied by $N_i^{(m)}$, therefore $\tilde{G}_i^{(m)}$ has no components relative to the constrained set NF_S . In order to have a descent direction with a zero mean value, the vector $G_i^{(m)}$ is computed using a Gram Schmidt method. In other terms $G_i^{(m)}$ satisfies

$$\sum_{i=1}^{n_s} G_i^{(m)} = 0. \quad (5.31)$$

This procedure is important in order to satisfy the first constraint of (4.15). We point out that if $\tilde{G}_i^{(m)}$ has already a zero mean value then $G_i^{(m)} = \tilde{G}_i^{(m)}$. Finally the α coefficient is chosen to satisfy

$$\delta\lambda_{i,\min} \leq \alpha G_i^{(m)} + \delta\lambda_i^{(m)} \leq \delta\lambda_{i,\max}. \quad (5.32)$$

We can show that this algorithm has some interesting properties:

Proposition 5.3. *If $\delta\lambda_{i,\min} \leq \phi_i - \lambda_i \leq \delta\lambda_{i,\max}$ then Algorithm 2 terminates in two steps.*

Proof. Since $\phi_i - \lambda_i$ is in the feasible region and $\delta\lambda_i^{(0)} = 0$ $i = 1, \dots, n_s$, from (5.32) we have

$$\left\{ \begin{array}{l} \text{if } \delta\lambda_{i,\min} = 0 \text{ then } \phi_i - \lambda_i \geq 0 \\ \text{if } \delta\lambda_{i,\max} = 0 \text{ then } \phi_i - \lambda_i \leq 0, \end{array} \right.$$

and we can see from (5.30) that $N_i^{(m)} = 1$, $i = 1, \dots, n_s$. Therefore since,

$$\sum_{i=1}^{n_s} \tilde{G}_i^{(m)} N_i^{(m)} = \sum_{i=1}^{n_s} \tilde{G}_i^{(m)} = \sum_{i=1}^{n_s} (\phi_i - \lambda_i) = 0,$$

we have

$$G_i^{(m)} = \tilde{G}_i^{(m)} = \phi_i - \lambda_i.$$

Finally, since $\phi_i - \lambda_i$ is in the feasible region and $\delta\lambda_i^{(0)} = 0$, we get

$$\delta\lambda_i^{m+1} = \alpha G_i^{(m)} + \delta\lambda_i^{(m)} = \phi_i - \lambda_i$$

At the iteration $m = 1$ we have $\tilde{G}_i^{(m)} = G_i^{(m)} = 0$, $i = 1, \dots, n_s$ and we stop.

We can also prove a more general stopping estimate:

Proposition 5.4. *Algorithm 2 terminates in less than $n_s - |NF_S^{(0)}| + 1$ steps.*

Proof. First we show that if $i \in NF_S^{(m)}$ then $i \in NF_S^{(m+1)}$. Actually from (5.30) we have

$$N_i^{(m)} = 0, \quad \tilde{G}_i^{(m)} = 0, \quad G_i^{(m)} = 0, \quad \delta\lambda_i^{(m+1)} = \delta\lambda_i^{(m)}, \quad \forall i \in NF_S^{(m)}.$$

At every iteration step the Algorithm 2 can add a constrained component to $NF_S^{(m)}$ or not. A maximum of $n_s - |NF_S^{(0)}|$ constraints can be added. If all the components are constrained then, for an $m \leq n_s - |NF_S^{(0)}| + 1$, we have $\tilde{G}_i^{(m)} = G_i^{(m)} = 0$ and the iterations stop.

Otherwise, if no constraint is added we can show that the iterations stop in the next step. In fact if at $m + 1$ no constraint is added then $\alpha = 1$ and we have

$$\delta\lambda_i^{(m+1)} = G_i^{(m)} + \delta\lambda_i^{(m)}, \text{ and } N_i^{(m+1)} = N_i^{(m)}, \quad i = 1, \dots, n_s,$$

Moreover

$$\tilde{G}_i^{(m+1)} = (\phi_i - \lambda_i - G_i^{(m)} - \delta\lambda_i^{(m)})N_i^{(m)},$$

since from (5.30) $(\lambda_i - \phi_i + \delta\lambda_i^{(m)})N_i^{(m)} = \tilde{G}_i^{(m)}$ we get

$$\tilde{G}_i^{(m+1)} = -G_i^{(m)}N_i + \tilde{G}_i^{(m)},$$

and since $G_i^{(m)}N_i = G_i^{(m)}$ we obtain

$$\tilde{G}_i^{(m+1)} = -G_i^{(m)} + \tilde{G}_i^{(m)}. \quad (5.33)$$

Finally plugging (5.33) into the definition of $G_i^{(m+1)}$ we get

$$G_i^{(m+1)} = \tilde{G}_i^{(m)} - \frac{\sum_{j=1}^{n_s} \tilde{G}_j^{(m)} N_j^{(m)}}{\sum_{j=1}^{n_s} (N_j^{(m)})^2} N_i^{(m)} - G_i^{(m)} + \frac{\sum_{j=1}^{n_s} G_j^{(m)} N_j^{(m)}}{\sum_{j=1}^{n_s} (N_j^{(m)})^2} N_i^{(m)}. \quad (5.34)$$

Since $G_j^{(m)}$ has a zero mean value, last term in (5.34) is null. Moreover since

$$\tilde{G}_i^{(m)} - \frac{\sum_{j=1}^{n_s} \tilde{G}_j^{(m)} N_j^{(m)}}{\sum_{j=1}^{n_s} (N_j^{(m)})^2} N_i^{(m)} = G_i^{(m)},$$

we have

$$G_i^{(m+1)} = 0. \quad (5.35)$$

And the Algorithm terminates.

We can also prove that Algorithm 2 finds the minimum solution of (4.15):

Proposition 5.5. *Algorithm 2 finds the optimal solution of (4.15)*

Proof. First we show that the iterations $\delta\lambda_i^{(m)}$ are always contained in the feasible region. The inequality constraints of (4.15) are enforced choosing a proper α as stated in (5.32).

Now we show that the equality constraint is satisfied by our algorithm. We use an induction argument and we suppose $\sum_{i=1}^{n_s} \delta\lambda_i^{(m)} = 0$ then

$$\sum_{i=1}^{n_s} \delta\lambda_i^{(m+1)} = \sum_{i=1}^{n_s} \alpha G_i^{(m)}.$$

Therefore

$$\sum_{i=1}^{n_s} \alpha G_i^{(m)} = \alpha \left(\sum_{i=1}^{n_s} \tilde{G}_i^{(m)} - \frac{\sum_{j=1}^{n_s} \tilde{G}_j^{(m)} N_j^{(m)}}{\sum_{j=1}^{n_s} (N_j^{(m)})^2} \sum_{i=1}^{n_s} N_i \right).$$

Since the N_i components can only be either 1 or 0 we get

$$\sum_{i=1}^{n_s} \alpha G_i^{(m)} = \alpha \left(\sum_{i=1}^{n_s} \tilde{G}_i^{(m)} - \sum_{i=1}^{n_c} \tilde{G}_i^{(m)} N_i^{(m)} \right).$$

Since $\tilde{G}_i^{(m)}$ is non-zero only if $N_i^{(m)} \neq 0$ we get $\sum_{i=1}^{n_s} \alpha G_i^{(m)} = 0$ and, since $\sum_{i=1}^{n_s} \delta \lambda_i^{(0)} = 0$, we can conclude the first part of the proof.

Finally we have to show that Algorithm 2 finds the minimum solution. We show that the Algorithm finds a local minimum then from the convexity and differentiability of the objective function and of the constraints it follows that minimum solution is also the global one.

Let's now show that if $G_i^{(m)} = 0$, $i = 1, \dots, n_s$ then there is no improving direction allowed. The allowed improving direction is $\tilde{G}_i^{(m)}$ but since $G_i^{(m)} = 0$, $i = 1, \dots, n_s$ we have that $\tilde{G}_i^{(m)}$ is orthogonal to the equality constraint of (4.15). Therefore there are no improving directions allowed.

Let's now give some more detailed estimates of the computational cost. First of all we show that in many cells the composition has not to be updated. Let us introduce the following condition

$$\lambda_{i,k}^n = \lambda_{i,k_j}^n \quad \forall j = 1, \dots, |\mathbb{I}_k^S|. \quad (5.36)$$

At the beginning of a numerical run most 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 Figure 5.6. Every cell that satisfies (5.36) has a trivial solution $\lambda_{i,k}^{n+1} = \lambda_{i,k}^n$, therefore we have only to update the composition of cells that don't not satisfy (5.36). In Figure 5.6 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 Section 5.6. Moreover the great majority (more than 98% in the test case of Figure 5.6) of the cells have less than two species and a more refined upper bound for the iterations of Algorithm 2 can be computed. In fact in those cells we have $\delta \lambda_{i,min} = \delta \lambda_{i,max} = 0 \quad \forall i \in \mathbb{R}^{n_s} : \lambda_i = 0$ therefore $|NF_S^{(0)}| = n_s - 2$ and, thanks to Proposition 5.4, the maximum number of iterations is equal to 3.

To conclude, in Figure 5.6 we can show good conservation properties even during the reconstruction of $\phi_{i,\Delta}^n$ at time steps 50 and 100. The mass is not conserved exactly because the velocity field is only weakly divergence free. The reconstruction algorithm is applied when the ratio between the cells that have only one species and the total number of the cells becomes lower than an assigned threshold.

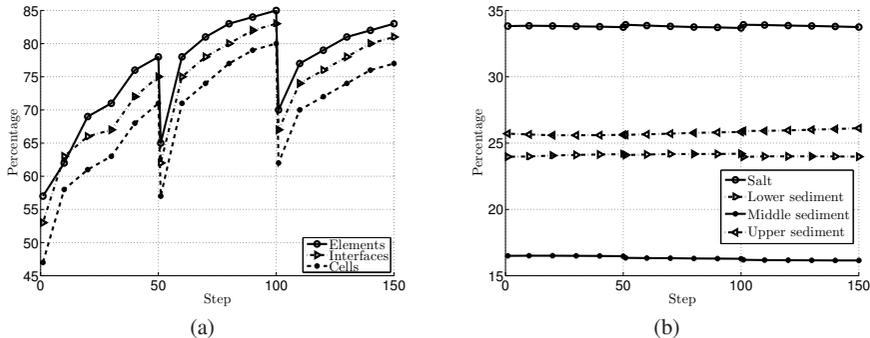
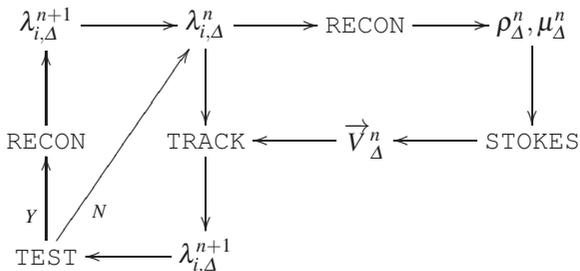


Fig. 5.6: In Figure (a) we show the percentage of the cells, elements, and interfaces inside the active band. The active band is composed by the cells that do not satisfy (5.36). In this case we have used a coarse grid therefore a lot of cells lie near an interface. However we can still achieve an average 20% reduction of the computational cost. In Figure (b) we show the percentage volume of the four species. At steps 50 and 100 a set reconstruction algorithm is applied.

5.5 Algorithm workflow

In this section we briefly describe the workflow of the algorithm we have developed. The first step is to define the physical properties of the sedimentary layers and their initial positions. These data are translated in the volume fractions $\lambda_{i,k}^n$, with $n = 0$. The code advances in time using the following scheme:



First of all the quantity $\lambda_{i,\Delta}^n$ is copied and the RECON Algorithm 1, is applied. This algorithm reduces the diffusion layer to the size of a cell. This procedure is important in order to limit the effects of numerical diffusion. From the cell partial volumes, the density and viscosity fields are created and the Stokes solver STOKES computes the velocity and pressure fields. The velocity field is used to track the compositions and find $\lambda_{i,k}^{n+1}$ using the algorithm TRACK. The latter algorithm consists in several steps:

1. construction of the cell-to-be-updated database;

2. computation of the time step so that (4.12) is satisfied. Then the number of the intermediate steps necessary to evolve the compositions from t^n to t^{n+1} is computed;
3. finally the interface fluxes are computed and the cell partial volumes updated.

The diffusion of the $\lambda_{i,k}^{n+1}$ is evaluated in `TEST` and the number of the cells that do not satisfy (5.36) is computed. If there are too many diffused cells the algorithm 1 is applied to $\lambda_{i,\Delta}^{n+1}$.

5.6 Numerical results

The test case we analyze in this chapter was already presented in Figure 5.1, and represents 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 evolution time is equal to 34.35 Mya. Density and viscosity have been taken equal to $2.2 \cdot 10^3$ kg/m³ and $0.1 \cdot 10^{20}$ Pas respectively for the salt layer and equal to $2.0\text{--}2.6 \cdot 10^3$ kg/m³ and $10^{20}\text{--}10^{21}$ Pas for the overbearing layers. These are reasonable values, physically speaking. Among the several simulation runs, the one we present here has about 900k unknowns, and requires approximately 4.2Gb of RAM. The computations have been run on an AMD Opteron 8212 Dual-Core 2GHz processor.

Only a few examples of three-dimensional cases on sedimentary basins are found in the literature, for example in [46], or Zadeh [107]. The main reason of such a lack of references resides in the dramatic rise of the computational cost that the switch to 3D requires. Yet a three-dimensional model is necessary to capture the basin dynamics completely. Anyway, as for the 2D case, the domain boundaries may introduce some undesired effects, worsening the quality of the results.

The simulation runs reported in this work not only represent an enrichment of the set of the three-dimensional cases, but also are provided with 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 three-dimensional multiframe 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 interface surfaces representing realistic sedimentary basins, and, as a consequence, the perturbations causing the Rayleigh–Taylor instabilities are not imposed *a priori* on a plane surface, but originate from the physical shape of the horizons (see Figure 5.7).

The ease of handling complicated geometries is enhanced by the use of unstructured meshes as discretizing tools. Figure 5.8 shows subsequent steps of the basin evolution, that lead to the shaping of some salt diapirs.

Secondly, the interface surfaces have been drawn with an 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

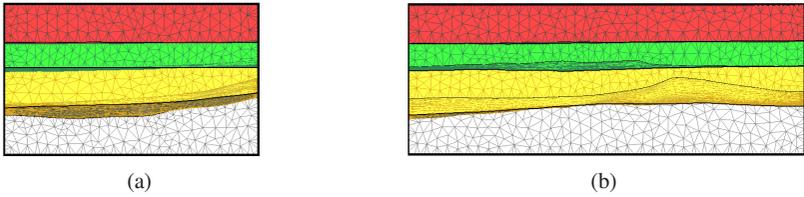


Fig. 5.7: A couple of lateral views of the basin used in the simulations. Salt is in white while heavier sediments are in yellow and green. In Figure (b) is visible the bump between the white and the yellow layers which leads to the diapir growth as a consequence of the Rayleigh–Taylor instability.

and in a fully automatic way phenomena of key importance in the sedimentary basin evolution, such as salt diapir detachments (see Figure 5.8(e)) and possible horizon intersections (see Figure 5.9(b)–5.9(d)). In the following figures we illustrate the evolution of the surface between the salt and the overburden layers, that finally ends by forming a diapir.

The code also computes other quantities of physical interest, as velocity, pressure and strain fields. As an example, in Figure 5.11 and 5.12 we show the distribution of vertical velocity and streamlines on a sectioning plane. In the first one 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, after that, the evolution is almost stationary. In the second figure we can appreciate the streamlines representing the flow motion leading to the main diapir formation, at a time step of major basin activity.

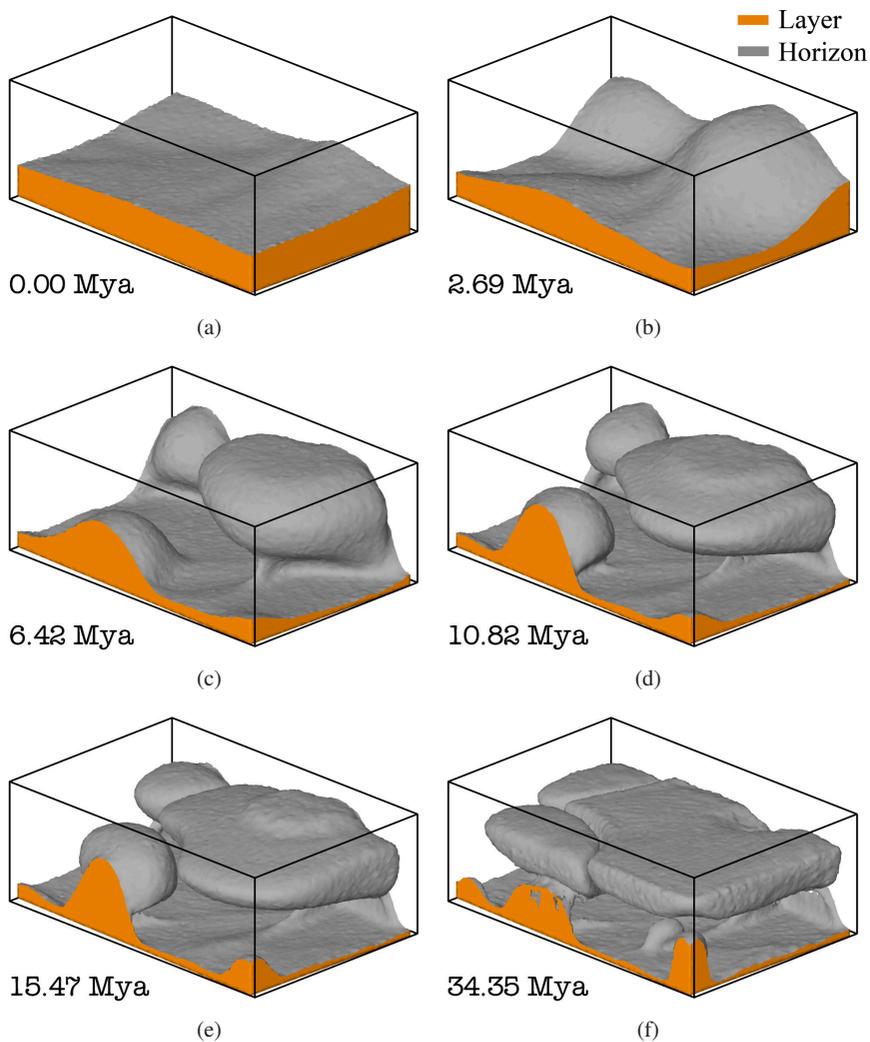


Fig. 5.8: Progressive evolution of salt diapirs: in about 34 Mya the growth of the three diapirs is almost complete.

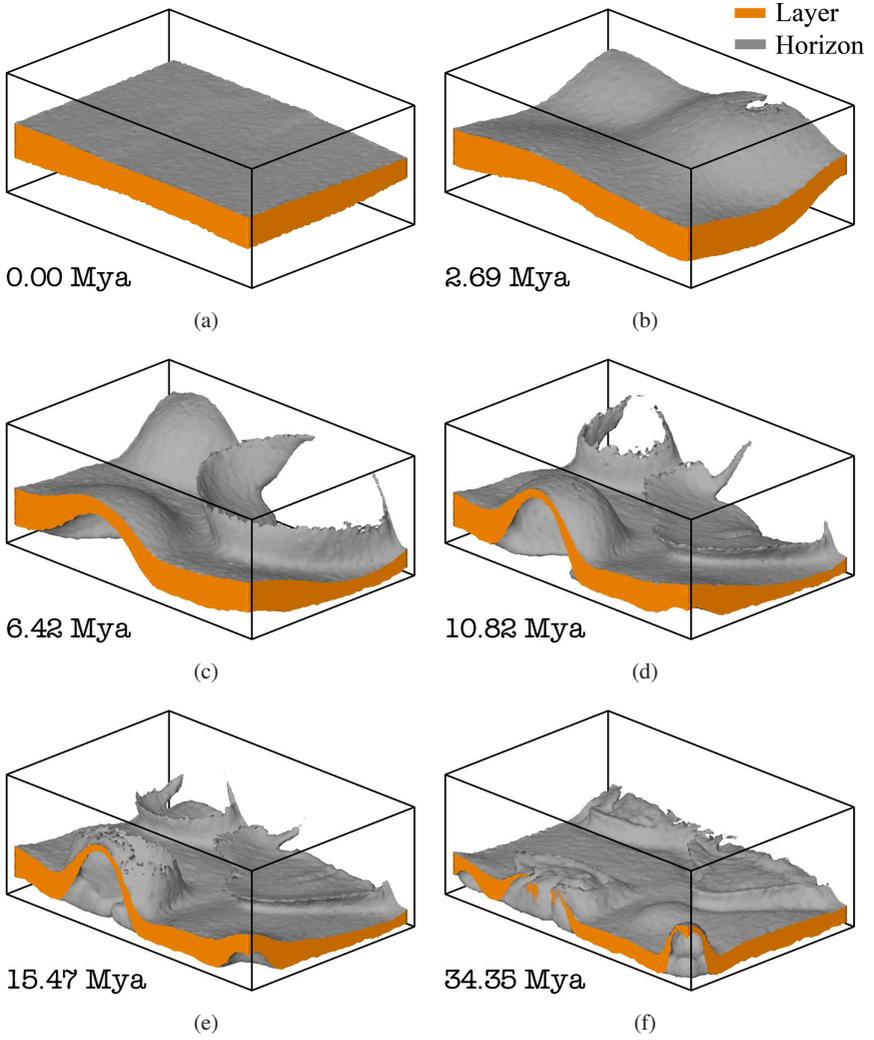


Fig. 5.9: Progressive evolution of lower sediment: salt rise perforates this sediment in three regions.

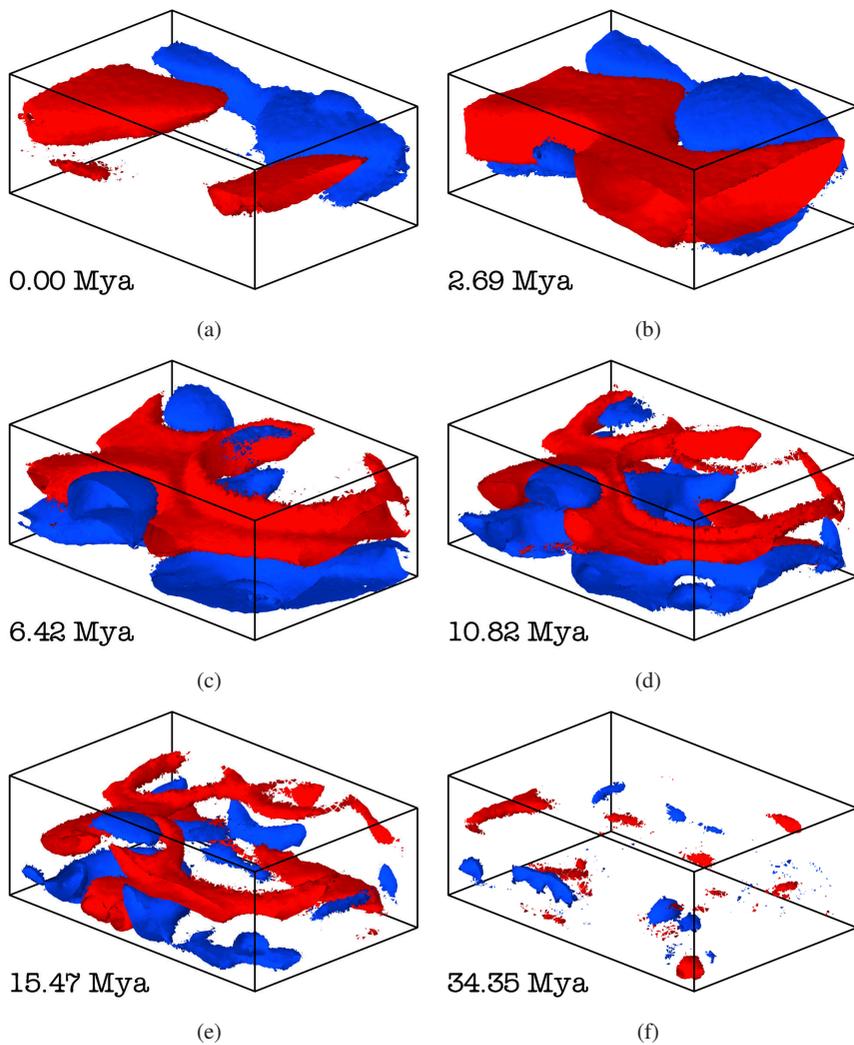


Fig. 5.10: Two isosurfaces showing the distribution of the vertical stress component σ_{zz} . (Red: 2.0 MPa , Blue: -2.0 MPa)

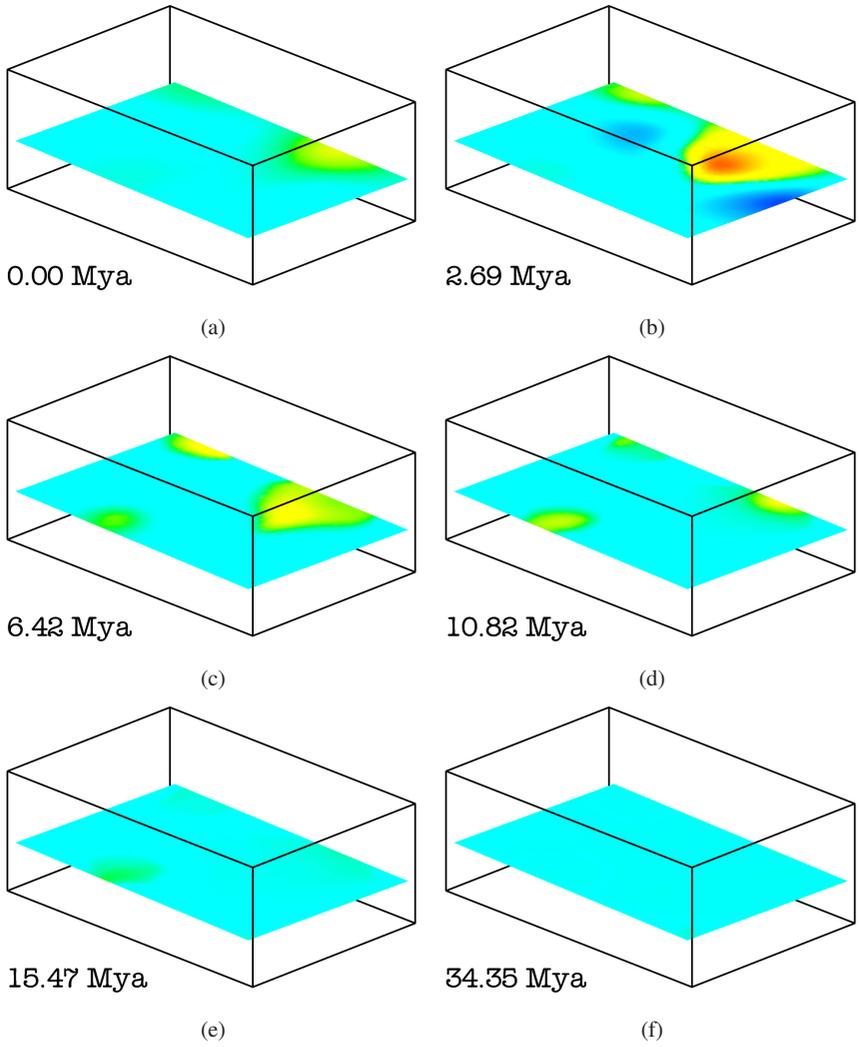


Fig. 5.11: A series of slabs of vertical velocity V_z , at $z = 2.18$ km from the bottom of the basin: the most part of the development of the diapirs happens in the first 11 Mya, while in the last 24 Mya the flows become gradually stationary. (Red: 1.64 km/Mya, Cyan: 0.0 km/Mya, Blue: -0.82 km/Mya)

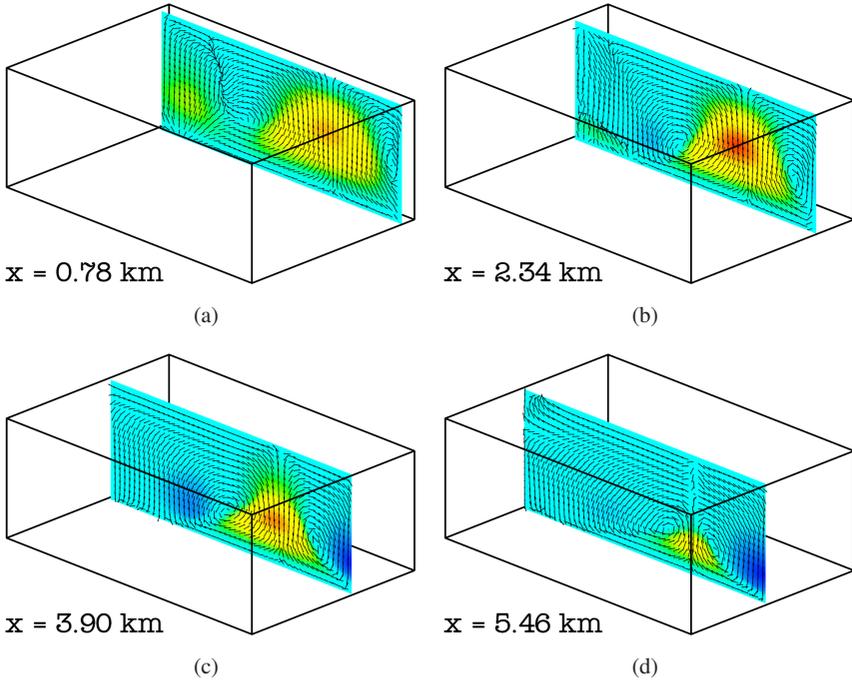


Fig. 5.12: A series of slabs of vertical velocity V_z and stream-lines at $t = 2.69$ Mya: two main vortices are visible near the biggest diapir. (Red: 1.64 km/Mya, Cyan: 0.0 km/Mya, Blue: -0.82 km/Mya)

5.7 Nomenclature

Index	Meaning	Interval
i	Layer index	$[1, n_s]$
k	Cell index	$[1, n_c]$
j	Face index	$[1, n_k^C]$
r	Element index	$[1, n_e]$
n	Time step	-

Table 5.1: Table of the indexes.

Symbol	Meaning
Ω	The computational domain
$\partial\Omega$	The boundary of the computational domain
Ω_i	The i -th layer volume
\vec{X}	The position vector
\vec{n}	The normal to $\partial\Omega$
\hat{x}_i	The three unit vectors
Γ	The boundary of the computational domain
$\Gamma_B, \Gamma_L, \Gamma_S$	The three portions of Γ

Table 5.2: Table of the geometric quantities

Symbol	Meaning
\mathcal{T}_Δ^G	The grid
Δ	The mesh characteristic length
n_e^G	Number of elements of the grid
n_p^G	Number of points of the grid
n_c^G	Number of cells of the grid
e_r^G	The r -th element of the grid
\vec{x}_k^G	the k -th node of the grid
\mathcal{T}_Δ^M	The mini-grid
n_e^M	Number of elements of the mini-grid
n_p^M	Number of points of the mini-grid
n_c^M	Number of cells of the mini-grid
e_r^M	The r -th element of the mini-grid
\vec{x}_k^M	the k -th node of the mini-grid

Table 5.3: Table of the mesh quantities

Symbol	Meaning
λ_i	The characteristic function of Ω_i
$\bar{\sigma}$	The stress tensor
P	The pressure
ρ	The density
ρ_i	The density of the i -th layer
\vec{g}	The gravity
μ	The viscosity
μ_i	The viscosity of the i -th layer

Table 5.4: Table of the variables

Symbol	Meaning
t	The time
t^n	The time at time-step n
Δt^n	The time step
T	The total simulation time

Table 5.5: Table of the other symbols

Chapter 6

Numerical Models for advanced basin simulations

In this Chapter we include the physical aspects analyzed in [57], extended to a three-dimensional realistic framework.

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, viscoplastic and visco-elastic ones. Semi-empirical relations are widely adopted, since the theory can explain only a few mechanisms (an example is the viscous-fluid rheology of a crystalline structure, see [94]). In [68] a tensor splitting technique is exploited to adapt the simulation to almost every type of isotropic rheology. We concentrate on visco-plastic rheology, as it accounts 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 [57] but, in this work, we consider a wider choice of pseudo-plastic rheologies: the Carreau, Cross, Powell and Yeleswarapu relations. All these models can be handled numerically either with a temporal linearization or with a fixed point iteration technique, being the former a cheaper choice and the latter a more accurate one.

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 [24], or even neglected (see [107, 68] and [108]). In [56] 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 and we solve a Stokes problem with a non null divergence. We stress that, if the fluid part is not simulated, as we have done in Chapter 3, it is mandatory to use the empirically-derived compaction curves.

Fault modeling is seldom included in geological basin simulations and, until now, only a few works consider this topic (see [55, 57, 68] and [79]). Fault location and

time of appearance are hardly predictable from the mechanical point of view, but fortunately seismic and well data are able to provide sufficiently accurate information as their age and location. Hence, we assume to know the location and the time of appearance of the faults and hence we concentrate on the modeling of their effects. In other words our rheology model is not able to predict the formation of the fault but since we know where and when they are formed, we are able to simulate accurately their effects.

Faults are fracture zones where the damaged rock creates sliding planes. A possible way to model them in a fluid framework is to reduce the fluid viscosity in the damaged area. For an active fault, the viscosity in a thin region around its sliding plane is reduced by several orders of magnitude (see [57]). This approach has two primary numerical problems: 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). To face these problems, we have implemented an implicit tracking algorithm and a local recursive bisection algorithm. For a review of the mesh refinement techniques in three-dimensions see [11, 52, 53, 70] and [111].

Let's now consider another important aspect, namely, the movement of the basin basement, of the free surface and of the lateral contour. The external boundary is subject to displacements as the surrounding soil moves with Earth plates. This effect has a key 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 [107] the basin is modeled as a fixed box. This geometrical constraint is not acceptable for many applications and several works, such as [57, 56, 68], have a more general geometrical treatment. All of them use a Lagrangian approach combined with frequent remeshing, as it handles naturally the movement of the boundary. However, the application of classical Lagrangian methods to a real three-dimensional case is expensive. Some new types of Lagrangian method, such as the Particle Finite Element methods (PFEM), have been introduced in mechanical engineering [64] and applied to computational geology [60, 63]. In these works, the Particle In a Cell technique is used. All these Lagrangian methods require a frequent mesh regeneration, making the cost of the algorithm critically dependent on the efficiency of the mesher. To overcome this aspect we have chosen, on the contrary, a method similar to the one described in [29]. The reason for this choice resides in the fact 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 the scheme described in Chapter 4 for the internal layers. The fusion of the two approaches gives rise to the so called Arbitrary Lagrangian Eulerian method (ALE), where the displacement of the grid is only prescribed on the boundary, while, for the internal nodes, a suitable movement law is considered, for example to minimize the mesh distortion. The ALE method has found several applications, see for instance [3, 5, 4, 34, 35, 43, 45, 49, 58, 61, 67], and [85]. The definition of the numerical algorithm for the computation of the internal grid move-

ment is the most critical part: we want, at the same time, to adapt the grid size, where necessary, without loosing the grid quality. To achieve this goal, we choose the so called r -adaptivity (see [5]) combined with the ALE scheme. This technique, which may also be regarded as a mesh fine-tuning technique, is a cheaper choice in terms of computational cost than the h -adaptivity, although less effective. Being the computational cost an important issue for our simulation, the r -adaptivity has been considered 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 bound error.

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 [5, 4, 58]. Its merits are the low computational cost and the compatibility with a metric-type adaptivity. On the other hand, it could fail if high curvatures are present on the domain boundary; in particular, non convex regions could induce mesh tangles, that's why sometimes this method is combined with a smoothing technique (see [34]). The second method, widely adopted in aeroelastic analysis, is the spring method (see [17, 18]). 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 [45]). 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.

6.1 Physical and mathematical models

6.1.1 Nomenclature

Let's recall the geometric model of the sedimentary basin (Figure 6.1). The domain $\Omega \in \mathbb{R}^3$ is divided into n_s subdomains Ω_i (without overlapping regions), 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 , that we suppose vertical for simplicity and where we impose a Dirichlet condition on the horizontal plane and a slip condition in the vertical direction. With respect to the previous chapters we have introduced the free surface condition for Γ_S . To complete our overview let

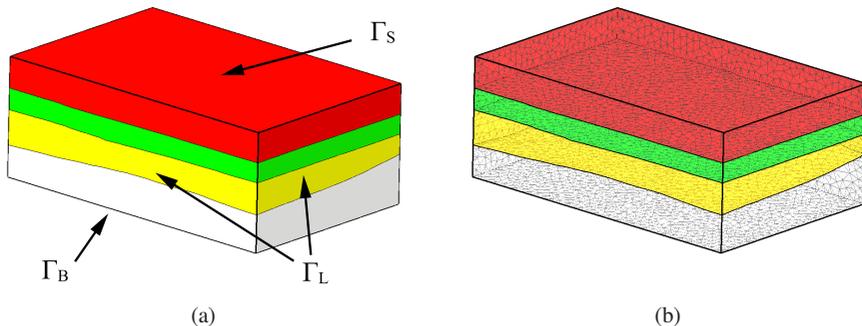


Fig. 6.1: a) External shape of the domain Ω . The external boundary Γ is divided into three parts: the basement Γ_B , the free surface Γ_S and the lateral contour Γ_L . b) An open three-dimensional view of a sedimentary basin containing three horizons and four layers.

us recall some nomenclature: $\vec{X} = (x_1, x_2, x_3) \in \Omega$ indicates a point in the spatial domain of coordinates x_i , with $i = 1, 2, 3$, $(\hat{x}_1, \hat{x}_2, \hat{x}_3)$ are the unit vectors of the coordinate system, \vec{n} is the domain outward normal and $t \in (0, T]$ is the time coordinate. For a generic vector \vec{u} , we denote its components with (u_1, u_2, u_3) .

6.1.2 Rheological models

We introduce the pseudo-plastic relations we have implemented. All of them are generalized Newtonian laws with a stress dependent viscosity. In this way, the structure of the equations is maintained. For each layer, we have considered five different rheological laws for the apparent viscosity μ_i for each component

$$\mu_i = \begin{cases} \mu_i^\infty = \mu_i^0 & \text{Newton} \\ \mu_i^\infty + \frac{\mu_i^0 - \mu_i^\infty}{(1 + \xi_i \gamma)^{(2-r)/r}} & \text{Carreau, [79]} \\ \mu_i^\infty + \frac{\mu_i^0 - \mu_i^\infty}{1 + (\xi_i \gamma)^r} & \text{Cross, [80]} \\ \mu_i^\infty + (\mu_i^0 - \mu_i^\infty) \frac{\sinh^{-1}(\xi_i \gamma)}{\xi_i \gamma} & \text{Powell, [80]} \\ \mu_i^\infty + (\mu_i^0 - \mu_i^\infty) \frac{1 + \ln(1 + \xi_i \gamma)}{1 + \xi_i \gamma} & \text{Yeleswarapu, [80]}, \end{cases} \quad (6.1)$$

where μ_i^0 is the reference unstressed viscosity, μ_i^∞ the asymptotic viscosity for $\gamma \rightarrow \infty$, γ is the squared Frobenius norm of the symmetric gradient $\vec{\nabla} \vec{u} + (\vec{\nabla} \vec{u})^T$, r is a positive coefficient, also known as the power law coefficient, ξ_i is a parameter and \vec{u} is the macroscopical velocity of the rock. We can now define an averaged viscosity by introducing the characteristic functions of the layers λ_i as

$$\mu = \psi \sum_{i=1}^{n_s} \lambda_i \mu_i, \quad (6.2)$$

where the viscosity abatement function $\psi(t, \vec{X})$ takes into account the possible presence of faults (a complete description of the fault model is provided in Section 6.4). The stress tensor is given by

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

where μ is now a function of the shear stress.

6.1.3 Modeling the compaction

We now detail the compaction model we have implemented for our simulations. In Chapter 3 we have discussed some compaction models that require the solution of the Darcy flow in the basin. Instead of these more demanding models, we have chosen a simple scheme, similar to the one in [57]. Namely, we model the porosity decrease in the deep layers imposing a non-solenoidal velocity of the sediments, in other terms

$$\vec{\nabla} \cdot \vec{u} = \Phi,$$

where Φ is the compaction function.

We here find a relation for Φ , without using the variables related to the fluid part, since they have been neglected. Let $S(x_1, x_2)$ be the relative height of the free surface, that is the distance along the \hat{x}_3 direction of the free surface from a reference plane. We can define the depth ζ as $\zeta = x_3 - S$ (see Figure 6.2). Now the objective is to model Φ using only a ϕ -depth relation. Let's consider, in particular, the Athy

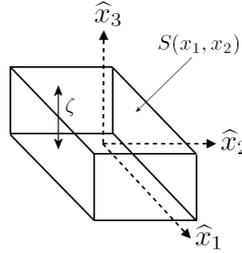


Fig. 6.2: An outline of the sedimentary basin with a reference frame having axes $(\hat{x}_1, \hat{x}_2, \hat{x}_3)$. The surface position is indicated with $S(x_1, x_2)$, the depth with ζ .

compaction law for the porosity ϕ

$$\phi = \exp(\mathcal{B}\zeta) \left(\sum_{i=1}^{n_s} \lambda_i \phi_i^0 \right), \quad (6.3)$$

where ϕ_i^0 is the reference porosity of the layers and \mathcal{B} is an empirical constant. We wish to determine Φ through (6.3). From [79] we get

$$\vec{\nabla} \cdot \vec{u} = \frac{1}{1-\phi} \frac{D\phi}{Dt}, \quad (6.4)$$

where $D/Dt = \partial/\partial t + \vec{u} \cdot \vec{\nabla}$ is the material derivative. Plugging equation (6.3) into (6.4) it becomes

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

Since $Dx_3/Dt = u_3$ and $\vec{\nabla} S$ is usually small (but this contribution could be included if necessary), the above equation can be approximated as

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

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

6.1.4 The model

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

$$\left\{ \begin{array}{l} \vec{\nabla} \cdot \vec{\sigma}(\mu, \vec{u}) - \vec{\nabla} P + \rho \vec{g} = 0 \text{ on } \Omega \times [0, T] \\ \vec{\nabla} \cdot \vec{u} = \Phi(\phi(\zeta)) \text{ on } \Omega \times [0, T] \\ \frac{\partial \lambda_i}{\partial t} + (\vec{u} - \vec{u}_g) \cdot \vec{\nabla} \lambda_i = 0 \text{ on } \Omega \times (0, T] \\ \frac{\partial \psi}{\partial t} + (\vec{u} - \vec{u}_g) \cdot \vec{\nabla} \psi = 0 \text{ on } \Omega \times (0, T] \\ \vec{\sigma} = \mu(\vec{u}, \psi, \xi_i) (\vec{\nabla} \vec{u} + (\vec{\nabla} \vec{u})^T) \text{ on } \Omega \times (0, T] \\ \lambda_i = \bar{\lambda}_i, \quad \psi = \bar{\psi} \text{ on } \Omega \times \{0\} \\ \vec{u} = \bar{u} \text{ on } \Gamma_B \\ (\vec{\sigma} - P\bar{I}) \cdot \hat{n} = 0 \text{ on } \Gamma_S \\ u_1 = \bar{u}_1, \quad u_2 = \bar{u}_2, \quad ((\vec{\sigma} - P\bar{I}) \cdot \hat{n}) \cdot \hat{x}_3 = 0 \text{ on } \Gamma_L, \end{array} \right. \quad (6.5)$$

where P is the total pressure of the solid-fluid mixture, $\bar{\psi}$ and $\bar{\lambda}_i$ are a suitable set of initial conditions, \bar{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 is given by

$$\rho = \phi \rho_f + (1 - \phi) \left(\sum_{i=1}^{n_s} \lambda_i \rho_i \right), \quad (6.6)$$

where ρ_f is the fluid density and ρ_i is the reference density of each sedimentary layer. The system above consists in two evolution equations: the fault tracking function equation and the partial volume equation. For the latter we use the method developed in Chapter 4, while for the fault tracking one we employ a modified level set method, to be discussed in detail in Section 6.4. The remaining equations form a Stokes problem with non solenoidal velocity, which is solved numerically with the techniques illustrated in Chapter 5.

6.2 Numerical discretization

In this section we recall briefly the spatial discretization of the model. First of all, we split the time interval $[0, T]$ into $[0, t^1, \dots, t^n, t^{n+1}, \dots, T]$ where $t^{n+1} = t^n + \Delta t^n$ and Δt^n is the n -th time step. Then, we indicate, for the generic variable $a(t, \vec{X})$, $a^n(\vec{X}) = a(t^n, \vec{X})$.

In order to describe the spatial discretization scheme, we introduce the geometric approximation of the domain Ω . Let \mathcal{T}_Δ^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 \vec{x}_k^G (with $k = 1, \dots, n_p^G$), where

the subscript Δ stands for the maximum diameter of the grid elements. From \mathcal{T}_Δ^G we build the mini-grid \mathcal{T}_Δ^M by adding n_e^G barycentric nodes; hence \mathcal{T}_Δ^M has $n_p^M = n_p^G + n_e^G$ nodes \vec{x}_k^M and $n_e^M = 4n_e^G$ elements e_r^M . Moreover, with a uniform refinement of \mathcal{T}_Δ^G carried out for N_R -times, we create a conformal grid \mathcal{T}_Δ^S , which has $n_e^S = 8N_R n_e^G$ elements e_r^S and n_p^S nodes \vec{x}_k^S . In the following we will refer to \mathcal{T}_Δ^G as the grid, to \mathcal{T}_Δ^M as the mini-grid, and to \mathcal{T}_Δ^S as the sub-grid. Let $\mathcal{F}_{r,j}^G$, $\mathcal{F}_{r,j}^M$ and $\mathcal{F}_{r,j}^S$ with $j = 1, \dots, 4$ 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 $\mathcal{F}_{r,j}^S$ with the element e_r^S will be denoted with $e_{r_j}^S$. We also define the map $[r, j] \rightarrow \bar{j}_{[r,j]}$, (where the subscript $[r, j]$ is appended in order to stress the dependence of \bar{j} on the couple r, j) such that, given the indices r and j , $\bar{j}_{[r,j]} : \mathcal{F}_{r_j, \bar{j}_{[r,j]}}^S = \mathcal{F}_{r,j}^S$ (see Figure 6.3). In other words, every interface between the elements e_r^S and $e_{r_j}^S$ is identified by two different local indexes j : once the local index j in e_r^S is identified, the other one, in the element $e_{r_j}^S$ is denoted by $\bar{j}_{[r,j]}$. Finally let us recall the discrete variables and

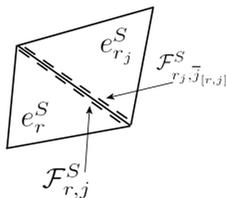


Fig. 6.3: The element e_r^S and its j -th neighbor $e_{r_j}^S$. The j -th face of the element e_r^S is $\mathcal{F}_{r,j}^S$ and corresponds to the $\bar{j}_{[r,j]}$ -th face of $e_{r_j}^S$, that is $\mathcal{F}_{r,j}^S = \mathcal{F}_{r_j, \bar{j}_{[r,j]}}^S$.

introduce the related discrete spaces:

$$\begin{aligned}
\vec{u}_\Delta^n &\in \overline{\mathbb{V}}_1^M : \overline{\mathbb{V}}_1^M = \left\{ \vec{\varphi}_\Delta \in (\mathbb{C}^0(\Omega))^3 : \vec{\varphi}_\Delta|_{e_r^M} \in (\mathbb{P}^1)^3, \quad \forall r = 1, \dots, n_e^M \right\} \\
P_\Delta^n &\in \mathbb{V}_1^G : \mathbb{V}_1^G = \left\{ \varphi_\Delta \in \mathbb{C}^0(\Omega) : \varphi_\Delta|_{e_r^G} \in \mathbb{P}^1, \quad \forall r = 1, \dots, n_e^G \right\} \\
\vec{\sigma}_\Delta^n &\in \overline{\mathbb{W}}_0^M : \overline{\mathbb{W}}_0^M = \left\{ \vec{\bar{\varphi}}_\Delta \in (L^2(\Omega))^{3 \times 3} : \vec{\bar{\varphi}}_\Delta|_{e_r^M} \in (\mathbb{P}^0)^{3 \times 3}, \quad \forall r = 1, \dots, n_e^M \right\} \\
\vec{u}_{g,\Delta}^n &\in \overline{\mathbb{V}}_1^G : \overline{\mathbb{V}}_1^G = \left\{ \vec{\varphi}_\Delta \in (\mathbb{C}^0(\Omega))^3 : \vec{\varphi}_\Delta|_{e_r^G} \in (\mathbb{P}^1)^3, \quad \forall r = 1, \dots, n_e^G \right\} \\
\rho_\Delta^n, \mu_\Delta^n, \phi_\Delta^n &\in \mathbb{W}_0^G : \mathbb{W}_0^G = \left\{ \varphi_\Delta \in L^2(\Omega) : \varphi_\Delta|_{e_r^G} \in \mathbb{P}^0, \quad \forall r = 1, \dots, n_e^G \right\} \\
\psi_\Delta^n &\in \mathbb{W}_0^S : \mathbb{W}_0^S = \left\{ \varphi_\Delta \in L^2(\Omega) : \varphi_\Delta|_{e_r^S} \in \mathbb{P}^0, \quad \forall r = 1, \dots, n_e^S \right\} \\
\lambda_{i,\Delta}^n &\in \mathbb{V}_0^S : \mathbb{V}_0^S = \left\{ \varphi_\Delta \in L^2(\Omega) : \varphi_\Delta|_{\tau_k^S} \in \mathbb{P}^0, \quad \forall k = 1, \dots, n_p^S \right\},
\end{aligned}$$

where 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, while the indices 0 and 1 indicate the degree of the basis.

6.3 Adaptive grid movement

6.3.1 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 we let $\vec{Y}^n \in \Omega^n$ be the position vector in the current reference system. We want to build a smooth displacement field $\vec{S}^n : \Omega^n \rightarrow \Omega^{n+1}$ such that

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

To implement our mesh movement-adaption scheme, we adopt the ideas of [41], [42], in other terms, we seek a best fit solution in $H^1(\Omega)$ to the alignment and to the equal distribution condition

$$(\vec{\nabla} \vec{X})^T \cdot \vec{M} \cdot (\vec{\nabla} \vec{X}) = \bar{I} \left(\frac{1}{|\Omega|} \int_\Omega \sqrt{\det(\vec{M})} \right)^{2/3} \quad (6.7)$$

with

$$\int_\Omega \sqrt{\det(\vec{M})} = |\Omega|, \quad (6.8)$$

where $\bar{\bar{M}}$ is a positive definite second order tensor which will be linked to the adaption process and \vec{X} in 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

$$\min_{\vec{X} \in H^1(\Omega)} \frac{1}{2} \|(\vec{\nabla} \vec{X})^T \cdot \bar{\bar{M}} \cdot (\vec{\nabla} \vec{X}) - \bar{I}\|_{\mathcal{D}}^2, \quad (6.9)$$

where $\|\bar{\bar{C}}\|_{\mathcal{D}}^2 = \int_{\Omega^n} \bar{\bar{C}} \cdot \bar{\mathcal{D}} \cdot \bar{\bar{C}}$, $\bar{\mathcal{D}}$ 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 [5] and [45] we choose $E_L|_{e_r} = K_L|_{e_r} = 1/|e_r^G|$. The optimality conditions of (6.9) can be transformed into a non linear partial differential equation. Since the displacement between two time steps is kept small, we consider a linearized form of the PDE. Moreover to simplify its derivation, we neglect, for now, the boundary conditions as they will be introduced later on.

The linearized form of (6.9) with respect to the coordinate system Y^n is

$$\min_{\vec{S}^n \in H^1(\Omega^n)} \frac{1}{2} \int_{\Omega^n} \bar{\bar{\epsilon}}^n \cdot \bar{\mathcal{D}} \cdot \bar{\bar{\epsilon}}^n + \int_{\Omega^n} \bar{\bar{\sigma}}_0^n : \bar{\bar{\epsilon}}^n, \quad (6.10)$$

where $\bar{\bar{\sigma}}_0^n$ is the pre-stress at time t^n . At $n = 0$, $\bar{\bar{\sigma}}_0^n = 0$ and we see shortly how to update it. Finally

$$\bar{\bar{\epsilon}}^n = \frac{1}{2} \left((\vec{\nabla} \vec{X}^n)^T \cdot \bar{\bar{M}}^n \cdot \vec{\nabla} \vec{X}^n - \bar{I} \right) \quad (6.11)$$

is the strain tensor. Plugging

$$\vec{X}^n = \vec{Y}^n + \vec{S}^n \quad (6.12)$$

into (6.11) we get

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

By differentiating equation (6.10) with respect to the displacement we obtain

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

where

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

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

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

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

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

$$-\vec{\nabla} \cdot \left[E_L(\vec{\nabla} \vec{S}^n : \bar{M}^n) \bar{I} + K_L \left((\vec{\nabla} \vec{S}^n)^T \cdot \bar{M}^n + \bar{M}^n \cdot (\vec{\nabla} \vec{S}^n) \right) + \bar{\Sigma}_0^n \cdot (\vec{\nabla} \vec{S}^n + \bar{I}) \cdot \bar{M}^n \right] = 0 \quad (6.13)$$

which is complemented by the following set of boundary conditions

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

This is a linear elastic-type equation with a pre-stress term that comes out from the previous deformation of the grid. We seek a solution using the finite element method, therefore we introduce now the discrete weak formulation

$$a_{\text{ALE}}(\vec{S}_{\Delta}^n, \vec{v}_{\Delta}) = F_{\text{ALE}}(\vec{v}_{\Delta}) \quad \forall \vec{v}_{\Delta} \in \bar{\nabla}_1^G, \quad (6.14)$$

where $\vec{S}_{\Delta}^n \in \bar{\nabla}_1^G$ is the discrete counterpart of \vec{S}^n and

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

If $K_L, E_L > 0$, problem (6.14) is coercive. Moreover if $\bar{\Sigma}_0^n, \bar{M}^n, K_L, E_L \in L^\infty(\Omega)$, then the bilinear form $a_{\text{ALE}}(\cdot, \cdot)$ and the linear functional $F_{\text{ALE}}(\cdot)$ are bounded and problem (6.14) has a unique solution (see [33] and [73]).

It can also be proved that the discrete problem is equivalent to a linear system with n_e^G unknowns and it is worth noting that this is a largely smaller size with respect to that of the Stokes problem, whose number of unknowns is equal to $3n_e^G + 4n_p^G$. Therefore, the computational cost of this part of the algorithm is small (as it will be shown in Section 6.5). The pre-stress term is updated as follows

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

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

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

Remark 6.1. The computation of Δt^n requires three steps: firstly, from the velocity \vec{u}^n , a maximum Δt_{ALE}^n is estimated such that the movement of the boundary elements does not exceed a certain threshold; secondly a maximum $\Delta t_{\mathcal{C}}^n$ is estimated such that the movement of the horizons is less than a prescribed length. Finally, the time step of the n -iteration is computed as

$$\Delta t^n = \min(\Delta t_{\text{ALE}}^n, \Delta t_{\mathcal{C}}^n). \quad (6.15)$$

■

6.3.2 Error estimation

In this section, we introduce the criterion for the mesh adaptation which is an a posteriori error estimator for the linearized Stokes problem. This problem has already been addressed in [97], considering a Newtonian law and fixed coefficients, but we need to modify slightly that derivation to suit the non-Newtonian relations which we linearize with respect to the previous time step. The discrete form of the Stokes problem is

$$\begin{cases} a(\vec{u}_{\Delta}^n, \vec{v}_{\Delta}) + b(P_{\Delta}^n, \vec{v}_{\Delta}) = F(\vec{v}_{\Delta}) & \forall \vec{v}_{\Delta} \in \mathbb{V}_1^M \\ b(q_{\Delta}, \vec{u}_{\Delta}^n) = G(q_{\Delta}), & \forall q_{\Delta} \in \mathbb{V}_1^G, \end{cases} \quad (6.16)$$

with

$$\begin{cases} a(\vec{u}_{\Delta}^n, \vec{v}_{\Delta}) = \int_{\Omega} \mu_{\Delta}^n (\vec{\nabla} \vec{u}_{\Delta}^n + (\vec{\nabla} \vec{u}_{\Delta}^n)^T) : (\vec{\nabla} \vec{v}_{\Delta}) \\ b(P_{\Delta}^n, \vec{v}_{\Delta}) = - \int_{\Omega} P_{\Delta}^n (\vec{\nabla} \cdot \vec{v}_{\Delta}) \\ F(\vec{v}_{\Delta}) = \int_{\Omega} \rho_{\Delta} (\vec{g} \cdot \vec{v}_{\Delta}) \\ G(q_{\Delta}) = \int_{\Omega} \Phi(\phi_{\Delta}) q_{\Delta} \end{cases}$$

where $\mu_{\Delta}^n = \mu_{\Delta}^n(\vec{u}^{n-1}, \psi^{n-1}, \xi_i)$ is computed with the velocity field of the previous time step.

The error estimator derivation follows the technique illustrated in [97]. Here, for the sake of simplicity, we drop the suffix n . Starting from (6.16) and introducing $\vec{v} \in H^1$ and $q \in L^2$ we get

$$\begin{aligned} a_e(U_{\Delta} - U, V) = & - \int_{\Omega} \rho_{\Delta} \vec{g} \cdot (\vec{v} - \vec{v}_{\Delta}) + \int_{\Omega} \Phi(q - q_{\Delta}) + \int_{\Omega} \bar{\sigma}_{\Delta} : \vec{\nabla} (\vec{v} - \vec{v}_{\Delta}) + \\ & - \int_{\Omega} P_{\Delta} \vec{\nabla} \cdot (\vec{v} - \vec{v}_{\Delta}) - \int_{\Omega} (q - q_{\Delta}) \vec{\nabla} \cdot \vec{u}_{\Delta}, \end{aligned}$$

where

$$a_e(U_{\Delta} - U, V) = - \int_{\Omega} (\bar{\sigma} - \bar{\sigma}_{\Delta}) : (\vec{\nabla} \vec{v}) + \int_{\Omega} (P - P_{\Delta}) (\vec{\nabla} \cdot \vec{v}) + \int_{\Omega} q \vec{\nabla} \cdot (\vec{u} - \vec{u}_{\Delta})$$

and

$$V = \left\{ \begin{array}{c} \vec{v} \\ q \end{array} \right\}, \quad U = \left\{ \begin{array}{c} \vec{u} \\ P \end{array} \right\}, \quad U_\Delta = \left\{ \begin{array}{c} \vec{u}_\Delta \\ P_\Delta \end{array} \right\}.$$

Finally,

$$\bar{\sigma} = \mu^n(\vec{u}^{n-1}, \psi^{n-1}, \xi_i)(\vec{\nabla} \vec{u} + (\vec{\nabla} \vec{u})^T), \quad \bar{\sigma}_\Delta = \mu_\Delta^n(\vec{u}_\Delta^{n-1}, \psi_\Delta^{n-1}, \xi_i)(\vec{\nabla} \vec{u}_\Delta + (\vec{\nabla} \vec{u}_\Delta)^T).$$

Resolving the contributions of each element and integrating by parts, we achieve this result

$$\begin{aligned} a_e(U_\Delta - U, V) &= \sum_{r=1}^{n_e^G} \left(- \int_{e_r^G} \rho_\Delta \vec{g} \cdot (\vec{v} - \vec{v}_\Delta) + \int_{e_r^G} \Phi(q - q_\Delta) \right. \\ &\quad \left. - \int_{e_r^G} (\vec{\nabla} \cdot \bar{\sigma}_\Delta) \cdot (\vec{v} - \vec{v}_\Delta) + \int_{e_r^G} \vec{\nabla} P_\Delta \cdot (\vec{v} - \vec{v}_\Delta) \right. \\ &\quad \left. + \oint_{\partial e_r^G} (\bar{\sigma}_\Delta \cdot \hat{n}) \cdot (\vec{v} - \vec{v}_\Delta) - \oint_{\partial e_r^G} P_\Delta (\vec{v} - \vec{v}_\Delta) \cdot \hat{n} - \int_{e_r^G} (q - q_\Delta) \vec{\nabla} \cdot \vec{u}_\Delta \right). \end{aligned}$$

Rearranging the boundary element terms and exploiting the Cauchy-Schwartz inequality we get

$$\begin{aligned} a_e(U_\Delta - U, V) &\leq \sum_{r=1}^{n_e^G} \left(\|\vec{\nabla} P_\Delta - \vec{\nabla} \cdot \bar{\sigma}_\Delta - \rho \vec{g}\|_{L^2(e_r^G)} \|\vec{v} - \vec{v}_\Delta\|_{L^2(e_r^G)} + \right. \\ &\quad \left. + \sum_{j=1}^4 \frac{1}{2} \|\llbracket (\bar{\sigma}_\Delta - P_\Delta \vec{I}) \cdot \hat{n}_{r,j} \rrbracket\|_{L^2(\mathcal{F}_{r,j}^G)} \|\vec{v} - \vec{v}_\Delta\|_{L^2(\mathcal{F}_{r,j}^G)} + \right. \\ &\quad \left. + \|\Phi - \vec{\nabla} \cdot \vec{u}_\Delta\|_{L^2(e_r^G)} \|q - q_\Delta\|_{L^2(e_r^G)} \right), \end{aligned}$$

where $\llbracket \cdot \rrbracket$ is the jump of a variable on the face $\mathcal{F}_{r,j}^G$ and $\hat{n}_{r,j}$ is the outward normal of the j -th face of the r -th element. We choose $\vec{v}_\Delta = \mathcal{C}_\Delta \vec{v}$ and $q_\Delta = \mathcal{C}_\Delta q$, where the Clement interpolation operator \mathcal{C}_Δ satisfies the following properties (see [33]) :

$$\begin{aligned} \|\mathcal{C}_\Delta \vec{v}\|_{L^2(\Omega)} &\leq C \|\vec{v}\|_{L^2(\Omega)} \quad \forall \Delta, \forall \vec{v} \in L^2(\Omega), \\ \|\vec{v} - \mathcal{C}_\Delta \vec{v}\|_{L^2(e_r^G)} &\leq C\Delta \|\vec{v}\|_{H^1(\tilde{\mathcal{C}}_r^G)} \quad \forall \Delta, \forall \vec{v} \in H^1(\Omega), \\ \|\vec{v} - \mathcal{C}_\Delta \vec{v}\|_{L^2(\mathcal{F}_{r,j}^G)} &\leq C\Delta^{1/2} \|\vec{v}\|_{H^1(\tilde{\mathcal{C}}_r^G)} \quad \forall \Delta, \forall \vec{v} \in H^1(\Omega), \end{aligned}$$

where $\tilde{\mathcal{C}}_r^G$ is the patch of the elements surrounding e_r^G and C is a generic constant (in the following, we will always use C to indicate any constant). These inequalities lead to

$$a_e(U_\Delta - U, V) \leq C \sum_{k=1}^{n_e^G} \left(\Delta R_r \|\vec{v}\|_{H^1(\mathcal{E}_r^G)} + D_r \|q\|_{L^2(\mathcal{E}_r^G)} + \Delta^{1/2} J_r \|\vec{v}\|_{H^1(\mathcal{E}_r^G)} \right), \quad (6.17)$$

with

$$\begin{cases} R_r = \|\vec{\nabla} P_\Delta - \vec{\nabla} \cdot \vec{\sigma}_\Delta - \rho_\Delta \vec{g}\|_{L^2(\mathcal{E}_r^G)} \\ D_r = \|\Phi - \vec{\nabla} \cdot \vec{u}_\Delta\|_{L^2(\mathcal{E}_r^G)} \\ J_r = \frac{1}{2} \sum_{j=1}^4 \|[(\vec{\sigma}_\Delta - P_\Delta \vec{I}) \cdot \hat{n}_{r,j}]\|_{L^2(\mathcal{F}_{r,j}^G)}. \end{cases} \quad (6.18)$$

The Cauchy-Schwartz inequality applied to (6.17) yields

$$a_e(U_\Delta - U, V) \leq C \left(\sqrt{\sum_{r=1}^{n_e^G} (\Delta R_r + \Delta^{1/2} J_r)^2} \sqrt{\sum_{r=1}^{n_e^G} \|\vec{v}\|_{H^1(\mathcal{E}_r^G)}^2} + \sqrt{\sum_{r=1}^{n_e^G} D_r^2} \sqrt{\sum_{r=1}^{n_e^G} \|q\|_{L^2(\mathcal{E}_r^G)}^2} \right).$$

Combining this result with the following relations

$$\begin{cases} (\Delta R_r + \Delta^{1/2} J_r)^2 \leq 2(\Delta^2 R_r^2 + \Delta J_r^2) \\ \sum_{r=1}^{n_e^G} \|\vec{v}\|_{H^1(\mathcal{E}_r^G)}^2 \leq C \|\vec{v}\|_{H^1(\Omega)}^2, \end{cases}$$

we get

$$\begin{aligned} a_e(U_\Delta - U, V) &\leq C \left(\left(\sqrt{\sum_{r=1}^{n_e^G} \Delta^2 R_r^2 + \Delta J_r^2} \right) \|\vec{v}\|_{H^1(\Omega)} + \left(\sqrt{\sum_{r=1}^{n_e^G} D_r^2} \right) \|q\|_{L^2(\Omega)} \right) \\ &\leq C \sqrt{\sum_{r=1}^{n_e^G} (\Delta^2 R_r^2 + \Delta J_r^2 + D_r^2)} \left(\|\vec{v}\|_{H^1(\Omega)} + \|q\|_{L^2(\Omega)} \right). \end{aligned}$$

Let's now introduce the direct product space $\mathcal{V}(\Omega) = H^1(\Omega) \times L^2(\Omega)$ equipped with the norm $\|V\|_{\mathcal{V}(\Omega)} = \|\vec{v}\|_{H^1(\Omega)} + \|q\|_{L^2(\Omega)}$. From the inf-sup condition (see [73, 33, 38]) we obtain

$$\|U_\Delta - U\|_{\mathcal{V}(\Omega)} \leq \sup_{V \in \mathcal{V}(\Omega)} \frac{a_e(U - U_\Delta, V)}{\|V\|_{\mathcal{V}(\Omega)}} \leq C \left(\sum_{r=1}^{n_e^G} (\Delta^2 R_r^2 + \Delta J_r^2 + D_r^2) \right)^{1/2},$$

and therefore

$$\|\vec{u} - \vec{u}_\Delta\|_{H^1(\Omega)} + \|P - P_\Delta\|_{L^2(\Omega)} \leq C \sqrt{\sum_{r=1}^{n_e^G} \mathcal{E}_r^2}, \quad (6.19)$$

with $\mathcal{E}_r^2 = \Delta^2 R_r^2 + \Delta J_r^2 + D_r^2$. In conclusion, expression (6.19) represents the residual error estimator $\mathcal{E}_r^n(\vec{u}_\Delta^n, P_\Delta^n, \mu_\Delta^n, \rho_\Delta^n, \phi_\Delta^n)$ that, from the physical data and the solution at time step n , can provide a local error upper bound.

6.3.3 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{M} relying upon the error estimate \mathcal{E}_r . We choose an isotropic metric $\bar{M} = \bar{I}\eta$. By doing so, the problem comes down to finding a suitable field η . Then, we define an auxiliary variable $\beta_r = \eta^{3/2}$ that represents the local volumetric deformation induced by the metric \bar{M} . β_r is defined as the solution of the following minimization problem

$$\begin{cases} \min_{\beta_r \in \mathbb{R}^{n_e^G}} \frac{1}{2}(\beta_r - \hat{\beta}_r)^2 + \frac{\delta}{2}(\beta_r - \tilde{\beta}_r)^2 \\ \sum_{r=1}^{n_e^G} \beta_r |e_r^{G,n-1}| = |\Omega|, \end{cases} \quad (6.20)$$

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

$$\tilde{\beta}_r = \tilde{K} \mathcal{R}_r, \quad \hat{\beta}_r = \frac{\sqrt{K_\mathcal{E}}}{\mathcal{E}_r}, \quad (6.21)$$

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

$$\tilde{K} = \frac{|\Omega|}{\sum_{r=1}^{n_e^G} \mathcal{R}_r}, \quad \sqrt{K_\mathcal{E}} = \frac{|\Omega|}{\sum_{r=1}^{n_e^G} 1/\mathcal{E}_r}. \quad (6.22)$$

As we will see shortly, the solution of (6.20) represents a compromise between the aim to distribute the error equally along the cells and the necessity to maintain the overall mesh quality, by means of the weight δ . The constraint in (6.20) is equivalent to the one in (6.7).

The term $(\beta_r - \hat{\beta}_r)^2/2$ in problem (6.20) triggers the equidistribution of the error. Indeed, considering (6.19), we would like that the contribution to the error of each element in the grid at time t^n be the same, that is

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

where we have supposed that, for small grid deformations, the ratio $|e_r^G|/|e_r^{G,n-1}|$ is almost independent of the grid geometry.

The term $\delta(\beta_r - \tilde{\beta}_r)^2/2$ in problem (6.20) ensures that the grid does not experi-

ence an excessive deformation, and in particular, we would like to impose that the elements deform with the same volume variation, i.e.

$$\beta_r = \tilde{K} \mathcal{R}_r \quad r = 1, \dots, n_e^G.$$

The constraint in (6.20) imposes that the metric \tilde{M} does not change the overall volume of the domain Ω .

The solution of (6.20) is

$$\begin{cases} \beta_r = \frac{\delta \tilde{\beta}_r + \hat{\beta}_r}{1 + \delta} - \Lambda \frac{|e_r^G|}{1 + \delta} \\ \Lambda = \frac{\sum_{r=1}^{n_e^G} (\delta \tilde{\beta}_r + \hat{\beta}_r) - |\Omega|(1 + \delta)}{\sum_{r=1}^{n_e^G} |e_r^G|^2}, \end{cases} \quad (6.23)$$

where Λ is the Lagrange multiplier of the constraint in (6.20).

We can also show that $\Lambda = 0$, in fact, using (6.22) and (6.21) we get

$$\Lambda = \frac{\delta \tilde{K} \sum_{r=1}^{n_e^G} \mathcal{R}_r + \sqrt{K_{\mathcal{E}}} \sum_{r=1}^{n_e^G} \frac{1}{\mathcal{E}_r} - |\Omega|(1 + \delta)}{\sum_{r=1}^{n_e^G} |e_r^G|^2} = \frac{\delta |\Omega| + |\Omega| - |\Omega|(1 + \delta)}{\sum_{r=1}^{n_e^G} |e_r^G|^2} = 0 \quad (6.24)$$

As regards the weight δ we choose

$$\delta > \max \left(\frac{\hat{\beta}_{\max} - \mathcal{U}}{\mathcal{U} - \tilde{\beta}_{\max}}, \frac{\mathcal{L} - \hat{\beta}_{\min}}{\tilde{\beta}_{\min} - \mathcal{L}}, 0 \right), \quad (6.25)$$

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

Proposition 6.1. *If $\mathcal{U} > \tilde{\beta}_{\max}$, $\mathcal{L} < \tilde{\beta}_{\min}$, then (6.23) and (6.25) imply $\mathcal{L} \leq \beta_r \leq \mathcal{U}$.*

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

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

Therefore, from (6.23) it follows that $\beta_r < (\hat{\beta}_{\max} + \delta \tilde{\beta}_{\max}) / (1 + \delta)$. Since (6.25) implies $(\hat{\beta}_{\max} + \delta \tilde{\beta}_{\max}) / (1 + \delta) > \mathcal{U}$, we get the first part of the bound. With similar arguments the other bound holds.

6.4 The handling of faults

6.4.1 A finite volume scheme

The time of appearance of a fault is given yet, faults must be tracked while the basin moves and they evolve during the movement. Faults are described by a small area with a decreased viscosity.

The tracking method relies on the knowledge of the elements whose viscosity is reduced, or, in other terms, of the grid elements the fault region $\Omega^{\mathcal{F}}$ goes through. Since we do not aim at representing the fault surface precisely, we consider a piecewise-constant indicator function, defined on the elements of the sub-grid, i.e. the uniformly refined grid. 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. We now define the fault function as

$$\begin{cases} \lambda^{\mathcal{F}} > 1/2 & \text{in } \Omega^{\mathcal{F}}, \\ \lambda^{\mathcal{F}} \leq 1/2 & \text{in } \Omega \setminus \Omega^{\mathcal{F}}. \end{cases} \quad (6.26)$$

Moreover, we define the viscosity abatement function ψ , already introduced in (6.2), as

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

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

$$\frac{\partial \lambda^{\mathcal{F}}}{\partial t} + (\vec{u} - \vec{u}_{g,\Delta}) \cdot \vec{\nabla} \lambda^{\mathcal{F}} = 0. \quad (6.27)$$

The discrete counterpart of $\lambda^{\mathcal{F}}(t, \cdot)$ is $\lambda_{\Delta}^{\mathcal{F}}(t, \cdot) \in \mathbb{W}_0^S$ 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 (6.27) with the coupled LS–VT method developed in Chapter 4, i.e.

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

where

$$v_{r,j}^n = \frac{\Delta t^{\mathcal{F},n}}{|e_r^S|} \int_{\mathcal{F}_{r,j}^S} (\vec{u}(t^n, \cdot) - \vec{u}_{g,\Delta}(t^n, \cdot)) \cdot \hat{n}$$

are the interface Courant numbers, and $\Delta t^{\mathcal{F},n}$ is the time step satisfying the following condition

$$v_{r,j}^n < \frac{1}{4} \quad \forall r = 1, \dots, n_e^S, \quad \forall j = 1, \dots, 4.$$

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

$$F_{r,j}^n = \begin{cases} v_{r,j}^n \lambda_{r,j}^{\mathcal{F},n} & \text{if } v_{r,j}^n \geq 0 \\ v_{r,j}^n \lambda_{r,\bar{j}[r,j]}^{\mathcal{F},n} & \text{otherwise,} \end{cases}$$

where

$$\lambda_{r,j}^{\mathcal{F},n} = \lambda_r^{\mathcal{F},n} + \delta \lambda_{r,j}^{\mathcal{F},n}. \quad (6.28)$$

The limiter $\gamma_{r,j}^n$ of the LS-VT scheme (defined in (6.28)) is computed, for $j = 1, \dots, 4$, in the following way

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

where

$$\Delta \lambda_{r,j}^{\mathcal{F},n} = \frac{1}{2} (\lambda_{r,j}^{\mathcal{F},n} - \lambda_r^{\mathcal{F},n})$$

\mathbb{J}_k is the set of indexes of the outflow faces of the r -cell, i.e. $\mathbb{J} = \{j \in 1, \dots, 4 : v_{r,j}^n > 0\}$. 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 $t > 0$. 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 [65, 81]). Moreover, in contrast with what we have done in Chapter 4, here we have used a discontinuous LS function.

6.4.2 Set reconstruction (continuous part)

The set reconstruction problem can be seen as follows: given $\lambda_{\Delta}^{\mathcal{F}}$ and $\Omega^{\mathcal{F}} = H(\lambda_{\Delta}^{\mathcal{F}} - \frac{1}{2})$, 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 has better properties, i.e., that is less diffused. In this section we construct, in the continuous framework a method, given $\lambda^{\mathcal{F}}$ and $\Omega^{\mathcal{F}}$, to find a function θ such that $H(\theta) = \Omega^{\mathcal{F}}$. 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

$$\alpha(t) = \frac{|\Omega^{\mathcal{F}}(t)|}{\int_{\Omega} \left(\lambda^{\mathcal{F}}(t, \cdot) - \frac{1}{2} \right) H(\lambda^{\mathcal{F}}(t, \cdot) - 1/2)}. \quad (6.29)$$

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

$$\begin{cases} J = \min_{\theta \in L^2(\Omega)} \frac{1}{2} \int_{\Omega} (\theta - \lambda^{\mathcal{F}})^2 \\ \int_{\Omega} \theta = \frac{|\Omega^{\mathcal{F}}|}{\alpha} \\ \theta \geq 0. \end{cases} \quad (6.30)$$

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

Proposition 6.2. *The solution of problem (6.30) is*

$$\bar{\theta} = \left(\lambda^{\mathcal{F}} - \frac{1}{2} \right) H(\lambda^{\mathcal{F}} - 1/2). \quad (6.31)$$

Proof. We show that every perturbation of the solution (6.31) yields an increase of the functional J . Let's consider a small perturbation of the solution that satisfies the constraints $\theta = \bar{\theta} + \varepsilon \tilde{\theta}$, where ε is a parameter that tends to zero. $\tilde{\theta}$ is a perturbation function that satisfies the constraints in (6.30), therefore

$$\tilde{\theta}(\vec{X}) \geq \begin{cases} 0 & \text{if } \vec{X} \notin \Omega^{\mathcal{F}} \\ -\frac{\bar{\theta}(\vec{X})}{\varepsilon} & \text{if } \vec{X} \in \Omega^{\mathcal{F}} \end{cases}, \quad \int_{\Omega^{\mathcal{F}}} \tilde{\theta} + \int_{\Omega/\Omega^{\mathcal{F}}} \tilde{\theta} = 0. \quad (6.32)$$

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

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

From (6.31) we get

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

that, combined with the last equation of (6.32), leads to

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

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

Moreover, if we neglect the compaction effects acting on the fault dislocation volume, we can prove that α does not depend upon time.

Proposition 6.3. *If the velocity field is divergence free, α is time independent.*

Proof. Let us compute α such that (6.29) is satisfied at $t = 0$. We show that if $\vec{\nabla} \cdot \vec{u} = 0$ the temporal variation of both the numerator and denominator in (6.29) are null. Indeed, for the numerator we have:

$$\frac{d}{dt} |\Omega^{\mathcal{F}}| = \frac{d}{dt} \int_{\Omega} H(\lambda^{\mathcal{F}} - 1/2) = \frac{d}{dt} \int_{\Omega^{\mathcal{F}}} 1 = \int_{\Omega^{\mathcal{F}}} \vec{\nabla} \cdot \vec{u} = 0,$$

and for the denominator

$$\frac{d}{dt} \int_{\Omega} \left(\lambda^{\mathcal{F}} - \frac{1}{2} \right) H(\lambda^{\mathcal{F}} - 1/2) = \int_{\Omega^{\mathcal{F}}} \frac{\partial \lambda^{\mathcal{F}}}{\partial t} + \vec{u} \cdot \vec{\nabla} \lambda^{\mathcal{F}} + \left(\lambda^{\mathcal{F}} - \frac{1}{2} \right) \vec{\nabla} \cdot \vec{u} = 0.$$

Therefore we get the proof.

6.4.3 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 (6.30) has the form

$$\min_{\Theta_{\Delta} \in \mathbb{W}_0^S} \frac{1}{2} \int_{\Omega} \left(\Theta_{\Delta} - \lambda^{\mathcal{F}} \right)^2 + \eta \left(\int_{\Omega} \Theta_{\Delta} - \frac{|\Omega^{\mathcal{F}}|}{\alpha} \right) - \int_{\Omega} \mathcal{M} \Theta_{\Delta} - \int_{\Omega} \mathcal{N} \left(\frac{1}{2} - \Theta_{\Delta} \right), \quad (6.33)$$

where $\eta \in \mathbb{R}$, $\mathcal{M} : \Omega \rightarrow \mathbb{R}$ and $\mathcal{N} : \Omega \rightarrow \mathbb{R}$ are the three Lagrange multipliers that force respectively the first and the second constraint in (6.30), and $\Theta_{\Delta} \leq 1/2$. The latter condition is not present in the continuous form, but comes from (6.31) that imposes $\theta < 1/2$. The equivalent optimality conditions for (6.33) are

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

where $\mathcal{M}_r, \mathcal{N}_r, \Theta_r \in \mathbb{W}_0^S$. This system is solvable with the Uzawa method (see [23] and [51]), as follows

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

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

6.4.4 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

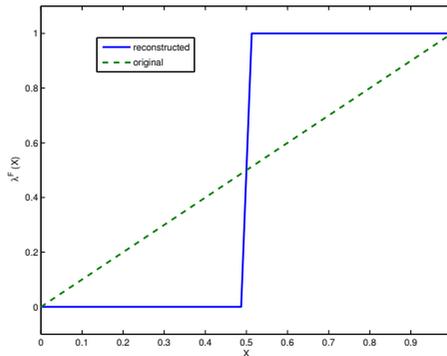


Fig. 6.4: The reinitialization algorithm applied once (dashed line) and twice (solid line) to a highly diffused function.

minimization problem (6.30) has an analytical solution that is:

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

In Figure 6.4 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%

6.4.5 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 may 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

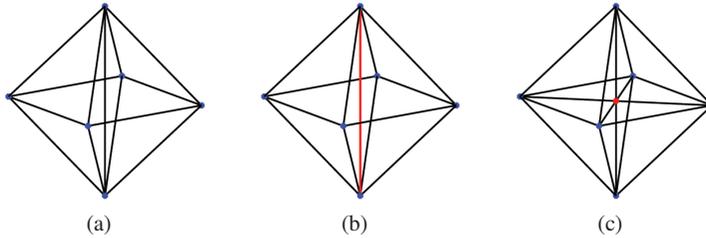


Fig. 6.5: The bisection of four tetrahedrons. a) Four tetrahedrons; b) The refinement edge (in red); c) Eight tetrahedrons, with the new vertex (in red).

automatically generates a coherent grid, but only some variants that belong to this class of methods, guarantee a good mesh quality. A brief review of the latter techniques can be found in [11, 52, 53, 70] and [111]. 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 $l_{\mathcal{G}}$ of the indexes of the elements to be refined. Then we proceed according to the following algorithm:

Algorithm 3 Mesh refinement

1. find the longest edge of $e_r^{\mathcal{G}}$, with $r \in l_{\mathcal{G}}$. 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, some more boundary data are generated. Finally, the mini-grid and the sub-grid are built from the refined grid.

6.5 Algorithm workflow

We outline the structure of the program for the solution of (6.5).

Algorithm 4 *At the generic time step n , we know the following variables of the problem:*

$$\left\{ \vec{u}_{\Delta}^{n-1}, P_{\Delta}^{n-1}, \vec{u}_{g,\Delta}^{n-1}, \lambda_{i,\Delta}^n, \lambda_{\Delta}^{\mathcal{F},n}, \rho_{\Delta}^{n-1}, \mu_{\Delta}^{n-1}, \phi^{n-1}, \mathcal{E}_{\Delta}^{n-1} \Delta t^{n-1} \right\}.$$

Starting from this situation, we use the following numerical scheme to solve problem (6.5):

1. from λ_i^n and relation (6.3) compute ϕ^n ;
2. from λ_i^n , ϕ^n and relation (6.6) compute ρ^n ;
3. from λ_i^n , ψ^n , \vec{u}^{n-1} and relations (6.1) and (6.2) compute μ^n ;
4. solve the 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. from relation (6.15) compute the time step Δt^n ;
6. from Δt^n and the adaptive grid movement algorithm (see Section 6.3) the grid velocity \vec{u}_g^n is evaluated;
7. from the coupled LS-VT method and from the modified level set method (described in Section 6.4), compute λ_i^{n+1} and ψ^{n+1} by solving the following evolution equations in $(t^n, t^{n+1}]$:

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

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

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

$$\left\{ \vec{u}_\Delta^n, P_\Delta^n, \vec{u}_{g,\Delta}^n, \lambda_{i,\Delta}^{n+1}, \lambda_{\Delta}^{\mathcal{F},n+1}, \rho_\Delta^n, \mu_\Delta^n, \phi^n, \mathcal{E}_\Delta^n, \Delta t^n \right\}.$$

Regarding the implementation of Algorithm 4, note that operations at steps (1), (2), and (3) require the computation of the layer reference density and viscosity. This operation requires the construction of a mean composition on every element e_r^G of the grid which is accomplished thanks to an interpolation of the node-based composition from the sub-grid to the elements of the grid, operation that is quite easy to perform since the sub-grid is nested in the grid. The element-based composition is filtered by the algorithm RECON (see Chapter 5), and saved apart from the original composition.

6.6 Numerical results

In this test we want to illustrate the capabilities of our simulation code. For the time being we only assure that the numerical techniques we have chosen work properly. 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 affected by a high degree of uncertainty.

The first test we have considered is the cinemactical evolution of a faulted basin. Here we would like to see whether the 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 [57]. In order to maximize the distortion effects we have set, for all the three layers in Figure 6.6, $\mu_i^0 = \mu_i^\infty = 10^{22} Pa \cdot s$. In other terms we have considered a consolidated rock behavior. This is quite unrealistic, usually the surface sediments are less compacted and weaker. However the main objective 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 is considered.

In Figure 6.6 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 and the mesh quality is kept high. Our result can be compared with the observed qualitative behavior of the faults in the extensional sedimentary basins (see Figure 6.7). As we can see, the qualitative behavior of the interaction between the sediment layers and the fault evolution are captured.

The second test case is focused on the ability of the code to simulate the evolution of a sedimentary basin when some sedimentation events occur. We have considered the following set of physical characteristics:

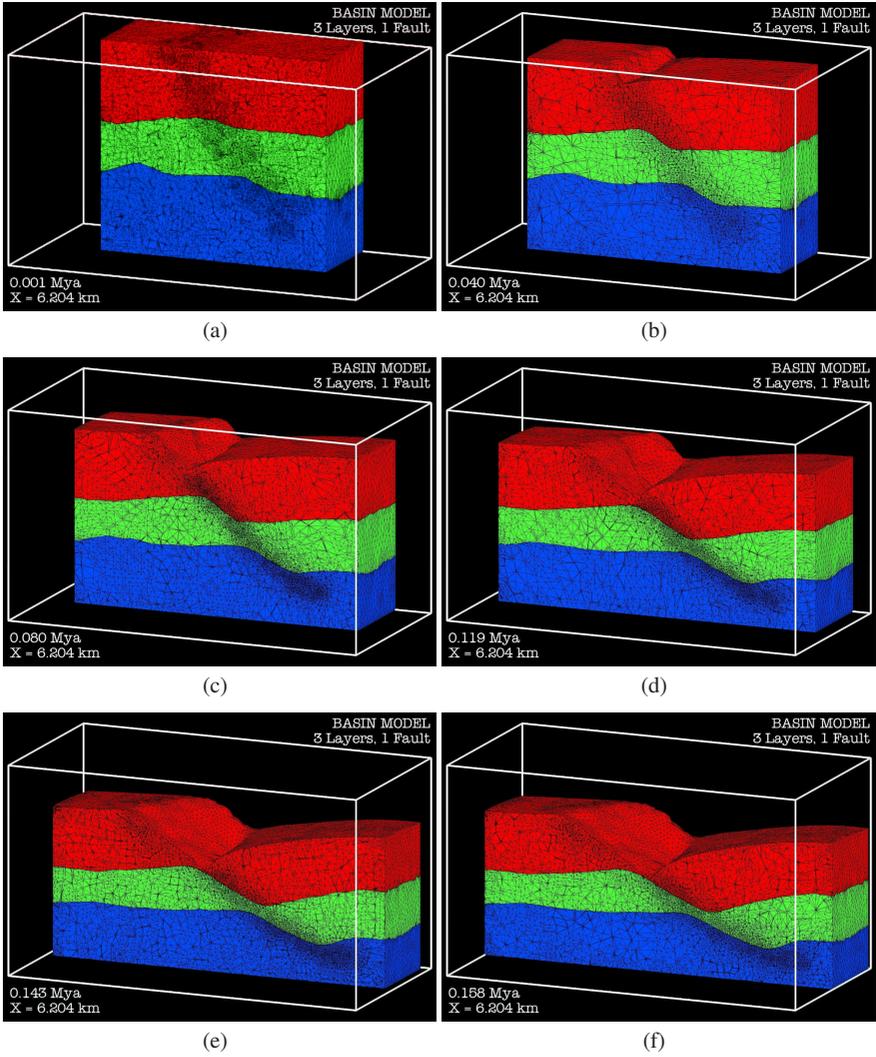


Fig. 6.6: The evolution of a faulted region with three layers.

Sediment	$\rho \ 10^3 \text{ kg/m}^3$	$\mu_i \ 10^{20} \text{ Pas}$	ϕ^0
Salt	2.14	1	0.05
Eocene	2.5	100	0.6
Oligocene	2.5	100	0.78
Early-Miocene	2.55	100	0.78
Mid-Miocene	2.6	100	0.78
Late-Miocene	2.6	100	0.78

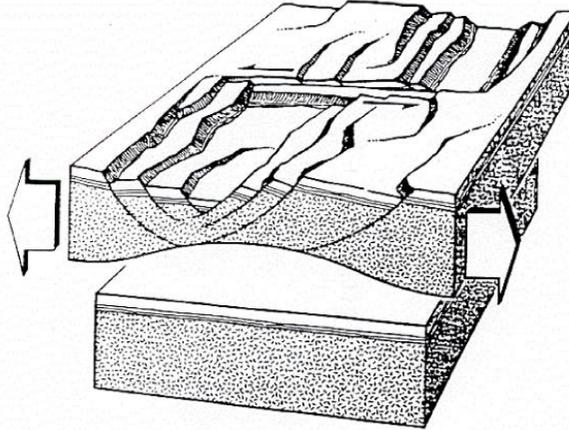


Figure 10. Extensional (rift) basin formation. From Koop and Stonely (1982).

Fig. 6.7: The effects of a fault system in an extensional sedimentary basin.

At the first time step (55.8Ma - ago) only three layers are present, the others are deposited according the following time-schedule:

Sediment	Time
Early-Miocene	23 Ma
Mid-Miocene	16.1 Ma
Late-Miocene	11.6 Ma

In Figure 6.8 we can see the whole evolution of the salt diapir, the subsequent deposition of the sedimentary layers and the horizons which are the separating interfaces between the layers. Also compaction effects are taken into account. We can see in Figure 6.9 the evolution of the porosity and its effects on the density field in Figure 6.10. The last test case we have considered is a sedimentary basin with two faults and a salt layer. This test case is important in order to prove the ability of the code to track the faults when they are strongly deformed by a rapidly rising salt diapir. Here we have considered the following layer characteristics:

Sediment	$\rho \cdot 10^3 \text{ kg/m}^3$	$\mu_i \cdot 10^{20} \text{ Pas}$	ϕ^0
Salt	2.14	1	0.05
Eocene	2.5	100	0.6
Oligocene	2.5	100	0.78

In Figure 6.11 we have outlined the evolution of the sedimentary basin at 55.8, 51.64, 45.2, 40.81, 38.25, 35.23, 32.23 and 29.67 millions years ago. The blue layer is the salt layer while the green layer is the Eocene layer and the red color is associated to the Oligocene layer. The horizons are depicted in yellow and the damaged fault-areas are the black-shaded regions. As we can see the fault regions are transported by the velocity field and are deformed by the rising diapir.

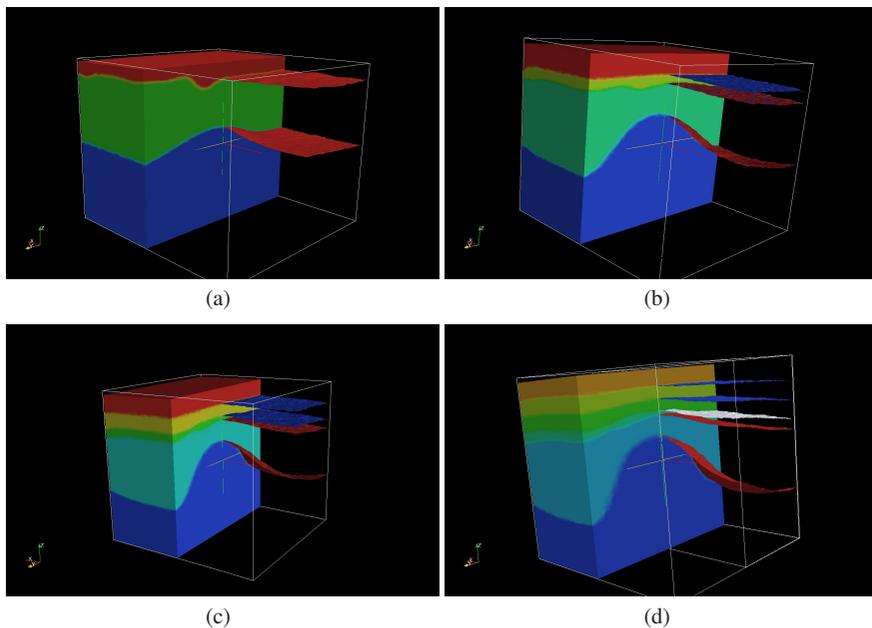


Fig. 6.8: The evolution of a salt diapir.

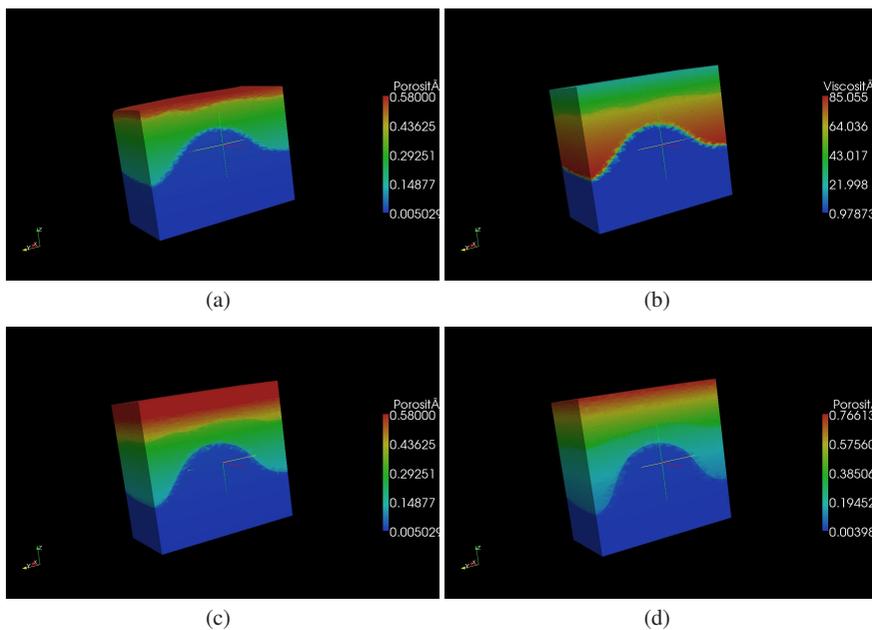


Fig. 6.9: The porosity field.

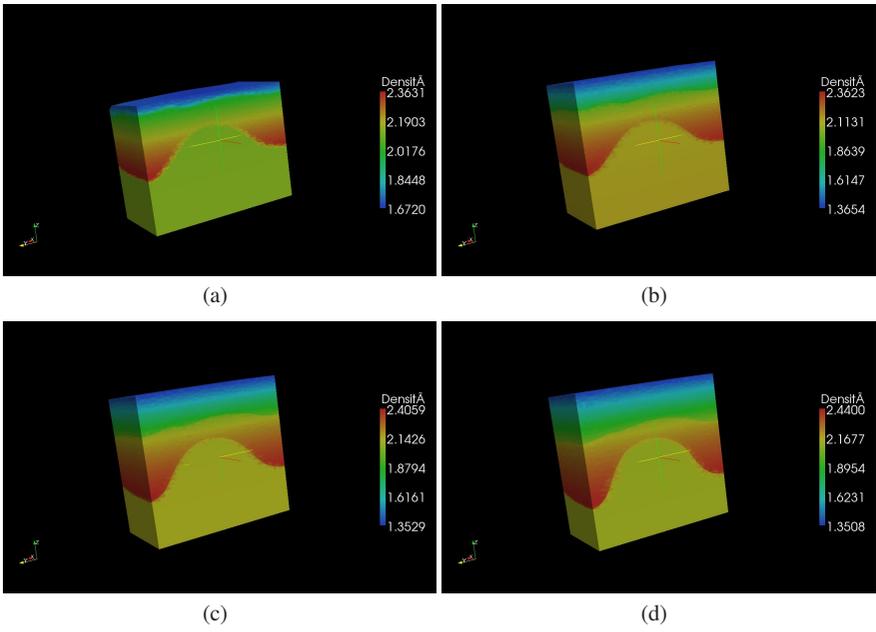


Fig. 6.10: The density field 10^3 kg/m^3 .

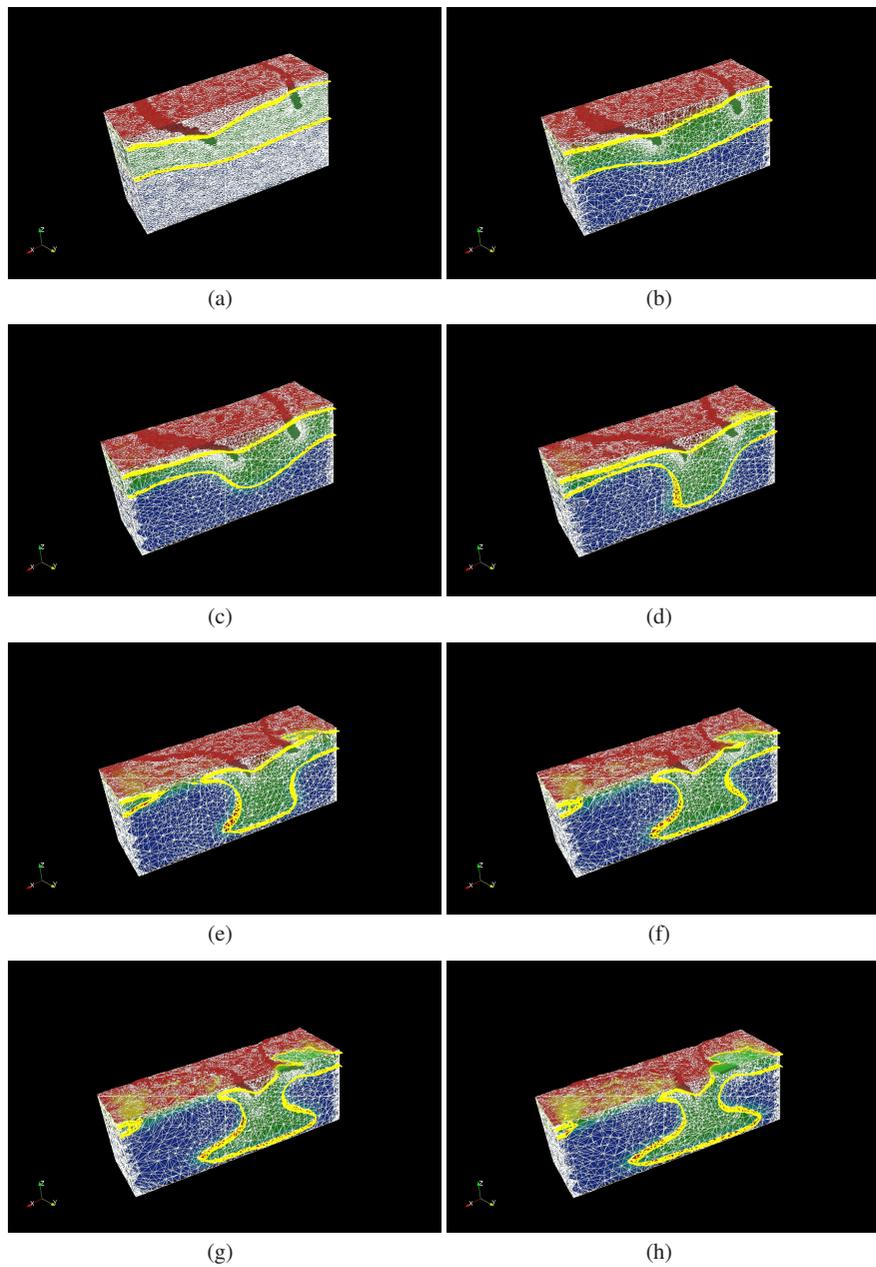


Fig. 6.11: The evolution of two faults and three layers.

Chapter 7

Conclusions

In this thesis we have developed all the numerical building blocks to simulate a three dimensional sedimentary basin. Some physical models have been outlined and a simplified model has been selected which consider only the evolution of the solid sediments. By now this seems a good compromise between the computational burden and the accuracy. However, as soon as high performance computers and parallel codes are available, we think that also the fluid evolution will be taken into account. In the third Chapter a new coupled LS-VT method is introduced. This method is computationally efficient and very robust. Moreover it seems promising for many other applications involving the conservative tracking of multi-fluid flows. From the theoretical point of view we are still seeking some global consistency results along with the total variation boundness properties and a convergence result on arbitrary meshes.

In Chapter 5 we have introduced the Stokes problem and we have examined its conditioning properties. There we have shown good computational results but we cannot prove theoretically the spectral equivalence between the Schur complement and the preconditioning matrix. Moreover the preconditioner of the stiffness matrix could be optimized. By now a ILU type of preconditioner is used but its performances are shown to be dependent on the viscosity jumps. The domain decomposition technique has been used for the construction of many optimal preconditioners with respect to the variations in the coefficients. The additive Schwartz method has been successfully applied to cases where the coefficients have small variations except, in some small isolated regions, see [78]. On the contrary, the substructuring schemes are very effective in all the cases in which the coefficient is almost piecewise constant. The regions where a coefficient is constant are also called subdomains in the domain decomposition theory. If we can construct a mesh such that all its elements belong to only one subdomain, then the substructuring methods are optimal with respect to the coefficient jumps, see [92]. As long as a non-Newtonian relation is used, the viscosity is not constant inside the subdomains and can be highly variable. Even though a Newtonian law is employed, in our case, the subdomains are represented by some level set functions and their boundaries, usually, cross the elements of the grid. In other terms, by now, no optimal technique seems to apply

to our case.

In Chapter 6 some other important features are introduced such as: the compaction, the movement of the domain boundaries, the fault tracking, the non-Newtonian rheologies.

The compaction effects are both treated in Chapter 4 and 6, in the latter one some simplified forms are used as we have neglected the evolution of the fluid part. However when proper parallel and efficient codes are developed, the models of Chapter 4 will be used.

The ALE scheme for the grid movement has been proven to be robust and efficient in fact a few remeshing are needed during long period simulations. In particular the scheme is very sensitive with respect to the initial quality of the mesh. If the initial mesh is bad, we have to limit the r -adaption or even to disable it. The initial mesh quality should be addressed, in the future, using dedicated mesh regularization algorithms.

The fault tracking technique seems also interesting in all the applications where we are not interested in a precise reconstruction of the domain boundaries and, therefore, a discontinuous level set function can be used. Future developments can include a more deep theoretical analysis of 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. Moreover, 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.

References

1. A. Anderson, X. Zheng, V. Cristini. Adaptive unstructured volume remeshing - 1: The method. *J. Comp. Physics*, 208, 616-625. 2005.
2. A. Anderson, J. Lowengrub, X. Zheng, V. Cristini. Adaptive unstructured volume remeshing - 2: Applications to two- and three-dimensional level-set simulations of multiphase flow. *J. Comp. Physics*, 208, 626-650. 2005.
3. F. Armero, E. Love. An arbitrary Lagrangian-Eulerian finite element method for finite strain plasticity. *Int. J. Numer. Methods Eng.*, 471-508, 57. 2003.
4. H. Askes, A. Rodriguez-Ferran, A. Huerta. Adaptive analysis of yield line patterns in plates with the arbitrary Lagrangian-Eulerian method. *Comput. and Struct.*, 70, 257-271. 1999.
5. H. Askes, A. Rodriguez-Ferran. A combined rh-adaptivity scheme based on domain subdivision. Formulation and linear examples. *Int. J. Numer. Methods Eng.*, 253-273, 51. 2001.
6. L.F. Athy. Density, porosity, and compaction of sedimentary rocks. *AAPG Bulletin*, 14, 1-22. 1930.
7. D.M. Audet, A. C. Fowler. A mathematical model for compaction in sedimentary basins. *Geophys. J. Int.*, 110, 577-590. 1992
8. E. Aulisa, S. Manservigi, R. Scardovelli. A mixed markers and volume-of-fluid method for the reconstruction and advection of interfaces in two-phases and free-boundary flows. *J. Comp. Physics* 188, 611-639. 2003.
9. E. Aulisa, S. Manservigi, R. Scardovelli. A surface marker algorithm coupled to an area-preserving marker redistribution method for three-dimensional interface tracking. *J. Comp. Physics* 197, 555-584. 2004.
10. P. Batten, C. Lambert, D.M. Causon. Positively conservative high resolution convection scheme for unstructured elements. *Int. J. Numer. Methods Eng.* 39, 1821-1838. 1996.
11. J. Bey. Tetrahedral grid refinement. *Mathematisches Institut Tubingen*, Report Nr. 18. 1995.
12. D. Bercovici, Y. Ricard, G. Schubert. A two-phase model for compaction and damage I. General Theory. *J. of geophys. research*, 106, B5, 8887-8906. 2001.
13. D. Bercovici, Y. Ricard, G. Schubert. *J. of geophys. research*, 106, B5, 8925-8939. 2001.
14. D. Bercovici, Y. Ricard. Two-phase damage theory and crustal rock failure. *Geophys. J. Int.*, 155, 1057-1064. 2003.
15. D. Bernaud, L. Dormieux, S. Maghous. A constitutive and numerical model for mechanical compaction in sedimentary basins. *Computers and Geotechnics*, Vol 33, pp 316-329. 2006.
16. K. Bitzer. Modeling consolidation and fluid flow in sedimentary basins. *Comput. and Geosci.*, 22, 467-478. 1996.
17. F. Blom. Consideration on the spring analogy. *Int. J. Numer. Methods Fluids*, 32, 647-668. 1998.
18. C.L. Bottasso. The Ball-Vertex Method: a New Simple Spring Analogy Method for Unstructured Dynamic Meshes. *Comput. Methods Appl. Mech. Engrg.*, 194, 39-41, 4244-4264. 2005.
19. H. Brezis. *Analisi funzionale. Teoria e applicazioni*. Liguori, 1986.

20. F. Brezzi, M. Fortin. Mixed and hybrid finite element methods. Springer. 1991.
21. A. Caboussat. Numerical Simulation of Two-Phase Free Surface Flows. Arch. Comput. Meth. Eng. 12, 2, 165-224. 2005.
22. N.L. Carter and F.D. Hanesen. Creep of rocksalt. Tectonophysics, 92, 275-333. 1983.
23. P.G. Ciarlet. Introduction to numerical linear algebra and optimization. Cambridge University Press. 1988.
24. S. Cloetingh, Y. Podladchikov. Perspectives in tectonic modeling. Tectonophysics, 320, 169-173, 2000.
25. B. Cockburn, C.W. Shu. TVB Runge Kutta local projection discontinuous Galerkin finite element method for conservation laws 2: general framework. Math. of Computing, 52, 411-435. 1989.
26. I.F. Collins. Associated and Non-Associated aspects of Constitutive Laws for Coupled Elastic - Plastic materials. Int. J. Geomech., 2, 2, 259-267. 2002.
27. J.A.D. Connolly, Yu. Yu. Podladchikov. Compaction-Driven Fluid Flow in Viscoelastic Rock. Geodinamica Acta. 11, 2-3, 55-84. 1998.
28. J. A.D. Connolly, Yu. Yu. Podladchikov. Temperature-dependent viscoelastic compaction and compartmentalization in sedimentary basins. Tectonophysics, 324, 137-168. 2000.
29. R.M. Darlington, T. L. McAbee, G. Rodrigue. A study of ALE simulations of Rayleigh-Taylor instability. Comput. Phys. Commun., 135, 58-73. 2001.
30. D.A. Di Pietro, S. Loforte, N. parolini. Mass preserving finite element implementations of the level set method. Applied numerical mathematics, 56, 1179-1195. 2006.
31. R.G. Duran, M.A. Muschietti. The Korn inequality for Jones domains. Electron. J. of Differential equations. 127, 1-10. 2004.
32. C. Elman, D.J. Silvester, A.J. Wathen. Finite Elements and Fast Iterative Solvers. Oxford University Press. 2005.
33. A. Ern, J.L. Guermond, Theory and Practice of Finite Elements, Springer. 2000.
34. J.L. Farinatti Aymone. Mesh motion technique for the ALE formulation in 3D large deformation problems. Int. J. Numer. Methods Eng., 59, 1879-1908. 2004.
35. L. Formaggia, F. Nobile. A Stability Analysis for the Arbitrary Lagrangian Eulerian Formulation with Finite Elements. East-West Journal of Numerical Mathematics, 7, 2-5, 105-131. 1998.
36. A.C. Fowler, Xin-She Yang. Fast and slow compaction in sedimentary basins. Siam J. Appl. Math, 59, 1, 365-385. 1998.
37. P. Fullsack, An arbitrary lagrangian eulerian formulation for creeping flows and its application in tectonic models. Geophys. J. Int, 120, 1-23. 1995.
38. V. Girault, P. A. Raviart. Finite Element Methods for Navier-Stokes Equations: Theory and Algorithm. Springer Verlag, 1986.
39. J. Glimm, J. W. Grove, X. L. LI, K. Shyue, Y. Zeng, Q. Zhang. Three-dimensional front tracking. Siam J. Sci. Computing 19, 703-727. 1998.
40. M. Gutierrez, M. Wangen. Modeling of compaction and over pressuring in sedimentary basins. Marine and Petroleum Geology, 22, 351-363. 2005.
41. 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.
42. W. Huang, W. Sun. Variational mesh adaption 2: error estimates and monitor functions. J. Comp. Phys., 184, 619-648. 2003.
43. A. Huerta, A. Rodriguez-Ferran, P. Diez, J. Sarrate. Adaptive finite element strategies based on error assessment. Int. J. Numer. Methods Eng., 46, 1803-1818. 1999.
44. A. Ibrahibegovic, F. Gharzeddine, L. Chorfi. Classical plasticity and viscoplasticity models reformulated: theoretical basis and numerical implementation. Int. J. Numer. Methods Eng., 42, 1499-1535. 1998.
45. A.A. Johnson, T.E. Tezduyar. Mesh update strategies in parallel finite element computations of flow problems with moving boundaries and interfaces. Comput. Methods Appl. Mech. Engrg, 119, 73-94. 1994.
46. B.J.P. Kaus and Y. Y. Podladchikov, Forward and reverse modeling of the three-dimensional viscous Rayleigh–Taylor instability. Geophysical Research Letters, 28, 1095-1098, 2001.

47. P.E. van Keken, C. J. Spiers, A. P. van den Berg, E. J. Muyzert. The effective viscosity of rock-salt: implementation of steady-state creep laws in numerical models of salt diapirism. *Tectonophysics*, 225, 457-476. 1993.
48. D.B. Kothe. Perspective on Eulerian Finite Volume Methods for Incompressible Interfacial Flows. Los Alamos report.
49. M. Lesoinne, C. Farhat. Geometric conservation laws for flow problems with moving boundaries and deformable meshes, and their impact in aeroelastic computations. *Comput. Methods Appl. Mech. Engrg.*, 134, 71-90. 1996.
50. R.J. LeVeque. *Numerical Methods for Conservation Laws*. Birkhauser. 1999.
51. J.L. Lions. *Optimal control of systems governed by partial differential equations*. Springer. 1971.
52. A. Liu, B. Joe. Quality Local refinement of tetrahedral meshes based on bisection. *SISC*, 16, 6, 1269-1291. 1994.
53. A. Liu, B. Joe. Quality local refinement of tetrahedral meshes based on 8-subtetrahedron subdivision. *Mathematics of Computation*, 65, 215, 1183, 1200. 1994.
54. X. Luo, G. Vasseur, A. Pouya, V. Lamoureux, A. Poliakov. Elastoplastic deformation of porous media applied to the modelling of compaction at basin scale. *Marine and Petroleum Geology*, 15, 145-162. 1998.
55. P. Massimi, *Perspective in computational salt tectonics*, Ph.D. Thesis, Politecnico di Milano, 2006.
56. P. Massimi, A. Quarteroni, G. Scrofani, An adaptive finite element method for modeling salt diapirism, *M3AS*, 16, 4, 2006.
57. P. Massimi, A. Quarteroni, F. Saleri, G. Scrofani, Modeling of salt tectonics, *Comp. Meth. Appl. Mech. Engrg.*, 197, 281-293, 2007.
58. A. Masud, M. Bhanabhagwanwala, R. A. Khurram. An adaptive mesh rezoning scheme for moving boundary flows and fluid-structure interaction. *Computers & fluids*, 36, 77-91. 2007.
59. P.D. Mineev, T. Chen, K. Nandakumar. A finite element technique for multigrid incompressible flow using eulerian grids. *J. Comp. Physics*, 187, 255-273. 2003.
60. L. Moresi, F. Dufour, H.B. Muhlhaus, A Lagrangian integration point finite element method for large deformation modeling of viscoelastic geomaterials. *J. Comp. Physics*, 184, 476-497, 2003.
61. M. Murayama, K. Nakahashi, K. Matsushima. Unstructured mesh for large movement and deformation. *AIAA Paper 2002-0122*, 40 th. Aerospace Sciences Meeting and Exhibit.
62. K. Ohmori, N. Saito. Flux-free Finite Element Method with Lagrange Multipliers for Two-fluid Flows. *Journal of Scientific Computing* 32, 2, 147-163. 2004.
63. C. O'Neill, L. Moresi, D. Muller, R. Albert, F. Dufour, Ellipsis 3D. A particle-in-cell finite-element hybrid code for modelling mantle convection and lithospheric deformation. *Computers and Geosciences*, 32, 10, 1769-1779. 2006.
64. E. Onate, S.R. Idelsohn, F. Del Pin, R. Aubry. The particle finite element method. An overview. *Int. J. Comput. Methods*, 1, 2, 267-307. 2004.
65. S. Osher, R. P. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer 2003.
66. P. Pedragal. *Introduction to optimization*. Springer 2004.
67. J.S. Peery, D. E. Carrol. Multi-Material ALE methods in unstructured grids. *Comput. Methods Appl. Mech. Engrg.*, 187, 591-619. 2000.
68. D. Peric, A. J. L. Crook, Computational strategies for predictive geology with reference to salt tectonics, *Comput. Methods Appl. Mech. Engrg.*, 193, 5195-5222, 2004.
69. S.P. van der Pijl, A Segal, C. Vuik, P. Wesseling. A mass-conserving level-Set method for modelling of multi-phase flows. *Int. J. Numer. Methods Fluids*, 47, 339-361. 2005.
70. A. Plaza, G.F. Carey. Local refinement of simplicial grids based on the skeleton. *Applied Numerical Mathematics*, 32, 195-218. 2000.
71. A.N.B. Poliakov, Yu. Yu. Podladchikov, E. Ch. Dawson. C. J. Talbot. Salt diapirism with simultaneous brittle faulting and viscous flow. *Special Publication of Geological Society: Salt Tectonics*. 1995.
72. S. Popinet, S. Zaleski. A front-tracking algorithm for accurate representation of surface tension. Report Univesitit Pierre et Marie Curie.

73. A. Quarteroni, A. Valli. Numerical Approximation of Partial Differential Equations Second Ed., Springer-Verlag, 1997.
74. A. Quarteroni, R. Sacco, F. Saleri. Matematica Numerica. Springer. 2000.
75. Y. Ricard, D. Bercovici, G. Schubert. A two-phase model for compaction and damage 2. Applications to compaction, deformation, and the role of interfacial surface tension. Journal of geophysical research, 106, B5, 8907-8924. 2001.
76. K. Runesson, M. Ristinmaa, L. Mahler. A comparison of viscoplasticity formats and algorithms. Mechanics of cohesive-frictional materials, 4, 75-98. 1999.
77. Y. Saad. Iterative methods for sparse linear systems, <http://www-users.cs.umn.edu/saad/books.html>, 1996.
78. R. Scheichl, E. Vainikko. Additive Schwarz with aggregation-based coarsening for elliptic problems with highly variable coefficients. Computing, 80, 319-343. 2007.
79. G. Scrofani. Numerical Basins Modeling and Tectonics, Ph.D. Thesis, <http://mox.polimi.it/it/progetti/publicazioni/tesi/scrofani.pdf>, Politecnico di Milano, 2007.
80. 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.
81. A. Sethian. Level Set Methods and Fast Marching Methods. Cambridge University Press. 1999.
82. S. Shin, D. Juric. Modeling Three-Dimensional Multiphase Flow Using a level Contour reconstruction Method for Front Tracking without Connectivity. J. Comp. Physics, 180, 427-470. 2002.
83. F. Schneider, J.L. Potdevin, S. Wolf, I. Faille. Mechanical and chemical compaction model for sedimentary basin simulators. Tectonophysics, 262, 307-317. 1996.
84. A.W. Skempton. The consolidation of clays by gravitational compaction. Q. J. Geol. Soc. Lond., 125, 373-412. 1970.
85. R.W. Smith. AUSM(Ale): A geometrical Conservative Arbitrary Lagrangian-Eulerian Flux Splitting Scheme. J. Comp. Physics, 150, 268-286. 1999.
86. A. Smolianski. Finite-Element/level-Set/Operator-Splitting (FELSOS) approach for computing two-fluid unsteady flows with free moving interfaces. Int. J. Numer. Methods Fluids, 24, 231-269. 2004.
87. SOPALE, a finite-element computer code for the computation of visco-plastic creeping flows with applications to tectonics, <http://geodynamics.oceanography.dal.ca/sopaledoc.html>.
88. F. Storti, F. Salvini and K. McClay. Fault-related folding in sandbox analogue models of thrust wedges. Journal of Structural Geology, 19, 3-4, 583-602. 1997.
89. E. Suetnova, G. Vasseur. 1-D Modelling rock compaction in sedimentary basins using a visco-elastic rheology. Earth and planetary science letters, 178, 373-383. 2000.
90. M. Sussman, E. G. Puckett. A Coupled level set and Volume-of-Fluid Method for Computing 3D and Axisymmetric Incompressible Two-Phase Flows. J. Comp. Physics, 162, 301-337. 2000.
91. D.J. Torres, J. U. Brackbill. The Point-Set Method: Front-Tracking without Connectivity. J. Comp. Physics, 165, 620-644. 2000.
92. A. Toselli, O. Widlund. Domain Decomposition Methods—Algorithms and Theory. 2004.
93. G. Tryggvason, B. Brunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, Y.-J Jant. A Front-Traking Method for the Computations of Multiphase Flow. J. Comp. Physics 169, 708-759. 2001.
94. D.L. Turcotte, G. Schubert, Geodynamics. Cambridge University Press. 2002.
95. J.L. Urai, C.J. Spiers, H.J. Zwart, G.S. Lister. Weakening of rock salt by water during long-term creep. Nature 324, 554-557. 1986.
96. V. Vasilyev, Y.Y. Podladchikov, D.A. Yuen. Modeling of compaction driven flow in poro-viscoelastic medium using adaptive wavelet collocation method. Geophys. res. lett, 25, 17, 3239-3242. 1998.
97. R. Verfurth. A review of a posteriori error estimation and adaptive mesh refinement techniques. Wiley-Teubner. 1996.
98. X.-S. Yang. Mathematical modelling of compaction and Diagenesis in sedimentary basins. Phd Thesis, Univestity of Oxford. 1997.

99. X.-S. Yang. A unified approach to mechanical compaction, pressure solution, mineral reactions and the temperature distribution in hydrocarbon basins. *Tectonophysics*, 330, 141-151. 2001.
100. M. Wangen, B. Antonsen, B. Fossum, L. K. Alm. A model for compaction of sedimentary basins. *Appl. Math. Modelling*, 14, 10, 506-517. 1990.
101. M. Wangen. A quantitative comparison of some mechanism generating overpressure in sedimentary basins. *Tectonophysics*, 334, 211-234. 2001.
102. S. Whitaker, *The Method of Volume Averaging*. Kluwer Academic Publishers. 1999.
103. J.K. Woods, P.D.M. Spelt, P.D. Lee, T.Selerand, C.J.Lawrence. Creeping flows of power-law fluids through periodic arrays of elliptical cylinders. *J. Non-Newtonian Fluid Mech.*, 111, 211-228. 2003.
104. A. Ismail-Zadeh, A.T., Korotkii, A.I., Naimark, B.M., Suetov, A.P., and Tsepelev, I.A. Implementation of a three-dimensional hydrodynamic model for evolution of sedimentary basins. *Comput. Math. and Math. Phys.*, 38, 7, 1138-1151. 1998.
105. A. Ismail-Zadeh, A.T., Tsepelev, I.A., Talbot, C., and Oster, P. A numerical method and parallel algorithm for three-dimensional modeling of salt diapirism (in Russian), in: *Problems in Dynamics and Seismicity of the Earth* (Eds. V.I. Keilis-Borok and G.M. Molchan). *Computational Seismology*, 31, 62-76, GEOS, Moscow. 2000.
106. A. Ismail-Zadeh, C. Talbot, Y. A. Volozh, Dynamic restoration of profiles across diapiric salt structures: numerical approach and it's applications, *Tectonophysics*, 337, 23-38, 2001.
107. A. Ismail-Zadeh, I. Tsepelev, C. Talbot, A. Korotkii, Three-dimensional forward and backward modelling of diapirism: numerical approach and its applicability to the evolution of salt structures in the Pricaspian basin. *Tectonophysics*, 387, 81-103. 2004.
108. S. Zaleski, P. Julien. Numerical simulation of Rayleigh–Taylor instability for single and multiple salt diapirs, *Tectonophysics*, 206, 55-69. 1991.
109. H-K. Zhao, T. Chan, B. Merriman, S. Osher. A variational level set approach to multiphase motion. *J. Comp. Phys.*, 127, 179-195. 1996.
110. S. Zlotnik, P. Diez. Hierarchical X-FEM for n-phase flow ($n > 2$). *Comput. Meth. Appl. Mech. Eng.*, 198, 2329-2338. 2009.
111. <http://www.iue.tuwien.ac.at/phd/wessner/node12.html>

