# High Performance Computing in Basin Modeling

# Simulating Mechanical Compaction with the Level Set Method

Dissertation
zur
Erlangung des Doktorgrades (Dr. rer. nat.)
der
Mathematisch-Naturwissenschaftlichen Fakultät
der
Rheinischen Friedrich-Wilhelms-Universität Bonn

von
**Sean Thomas McGovern**
aus
Redwood City, CA, USA

Bonn, October 2019

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Rheinischen Friedrich-Wilhelms-Universität Bonn.

| | |
|---|---|
| 1. Gutachter: | Prof. Dr. Stefan Kollet |
| 2. Gutachter: | Prof. Dr. Andreas Kemna |

| | |
|---|---|
| Tag der Promotion: | 20.05.2020 |
| Erscheinungsjahr: | 2020 |

*Il n'y a pas un être dans l'univers qu'on ne puisse, à quelque égard, regarder comme le centre commun de tous les autres, autour duquel ils sont tous ordonnés, en sorte qu'ils sont tous réciproquement fins et moyens les uns relativement aux autres. L'esprit se confond et se perd dans cette infinité de rapports, dont pas un n'est confondu ni perdu dans la foule. Que d'absurdes suppositions pour déduire toute cette harmonie de l'aveugle mécanisme de la matière mue fortuitement!*

Jean-Jacques Rousseau

*Émile, ou De l'éducation, IV, §991*

# Acknowledgements

This work is dedicated to my parents.

I'd like to acknowledge the support of all my family and friends. I'd like to thank all the wonderful people that I met in my time in the Rhineland. This thesis would not have been possible without my advisor Stefan Kollet or my co-authors at the Shell company, Claudius Bürger, Olaf Podlaha and Ronnie Schwede. I learned a lot from Carsten Burstedde as a student in his numerics lecture courses. I remain very grateful that Wolfgang Bangerth participated in this work and was such a gracious host for my six-week visit to Fort Collins, CO, USA. Finally, I thank the folks at the Jülich Supercomputing Center for their computing expertise and technical support.

# Abstract

The evolution of sedimentary basins is a complex set of processes developing over millions of year. Often, tectonic activity will create accommodation space that is filled in with sediments. These sediments undergo changes as pressure and geothermal energy are applied to them. Typically, sedimentary basins will be in a marine or fluvial environment, thus the sediments are saturated with a pore fluid, water or brine. Such a depositional environment, treated as a saturated porous medium, will undergo loading and compaction, as more sediments eject pore fluid out of the pore spaces around the solid matrix. The geochemistry of these sediment layers, strata, is of interest for possible hydrocarbon resource production.

Using mathematical models to approximate the evolution of physical states in sedimentary basins, called basin modeling, has progressed in the last decades due to the increasing computational resources available. The computational approach to approximately solving the model equations, formulated as partial differential equations, relies on reducing the physical domain to a "cloud of points", known as a mesh or grid. Physical states are then only defined at these points (or objects derived from them, e.g. faces). The quality of the grid's representation of physical geometry is essential to the quality of the approximate solutions obtained computationally.

In geological basins, the layering of the strata can be very complex. Portions of the domain might receive more or less sediments of varying physical characteristics, such as permeability (the ability to allow fluid to pass through) or compressibility (the change in porosity with respect to a change in stress). The layered sediments will change their geometry as they compact during deposition. Other examples of complexity include sharply discontinuous material properties, e.g., one layer might have drastic differences in permeability compared to its neighbors. Resolving where this change in permeability occurs would then be required to produce accurate simulations. Even more pronounced deformation comes from various tectonic processes that might have caused folding or even fracturing of the rock layers, as is commonly revealed through seismic imaging. In this thesis, we will not consider tectonic processes or fracturing, though these are important features in many geological basins and prime examples of geometrical complexity.

One of the main challenges in accurately simulating physical processes in geological contexts is the ability of the discrete mesh to conform to the physical states and material properties, given that the geometry of the layers can be deformed in ways that make it challenging to line up on a grid. As basin modeling is concerned with the evolution of sediment layers, a time-dependent context, one approach to this problem is to frequently recreate or modify the grid. The physical changes in the domain are adapted to a correspondingly updated grid. This has the advantage that the phyical quantities and material properties will be conforming to the new grid for the solution at the next time step. Recreating the mesh can be an expensive operation (in computational work), and can increase the amount of communication time between processors, when dealing with distributed parallel computing.

As one of the goals of this project has been to consider model applications in the context of

High-Performance Computing (HPC), the approach of frequently modifying the grid presents a possible impediment to scaling to higher processor counts. The purpose of this doctoral thesis is to consider techniques to represent the complex geometries found in geological basins in a way that does not require re-meshing.

The idea is to implicitly represent the layered structure of a basin, beginning with a fixed global domain, through which sediments are deposited at a certain speed. The interfaces between layers are implicitly given, allowing the identification at a point in the domain of what rock type is there, and so what material properties are appropriate. A well-known technique to evolve an interface, encoded as a fixed value level set of an auxiliary function, is the level set method. The interface moves according to how a given "speed function" evolves the auxiliary potential function. Whether the potential function at a point is above or below the value (often taken to be 0), determines whether the point is "in or out", and thus we can reconstruct the identity of the material, what layer it is in, at that point.

The novel contribution of this work is to apply the numerical technique of the level set method for interfaces coupled to physical processes in the modeling of sedimentary basins. First, using the principle of vertical effective stress, the work considers in 1-D the fundamental coupling between (1) the layer interfaces equipped with a speed function and a (2) pore pressure, combining to influence the movement of a rock layer based upon the sedimentation rate and the rate of compaction. As there are no publicly available benchmarks or standard solutions by which to verify a basin simulator, the developed model has been verified against one of the only available analytical solutions relevant to sedimentary basins, formulated as porous media, the Gibson consolidation problem. Further examples of multi-layer setups are presented.

In the second part, we consider a finite element discretization of a pore fluid flow solver coupled to a representation of geologic interfaces through a level set method. A description of the model equations, their discretization and their implementation in the finite element library, deal.II, is given. Subsequently, the verification example is considered, as well as some elaborations to more complex demonstration cases in 2-D, including a basin with a temperature field. A scaling study has been performed on Forschungszentrum Juelich's supercomputer JSC/Juwels to profile the code's parallel performance with distributed memory.

In the third part, three elaborations are briefly considered: (1) a small 3-D example derived from the Gibson consolidation problem, (2) the possibility of incorporating an effect similar to buoyancy in the level set speed function, and (3) the foundations required for adaptive mesh refinement.

A conclusion and some possibilities for extension complete the thesis.

# Contents

# List of Figures

# List of Tables

CHAPTER 1

# Introduction

This thesis considers the numerical modeling of sedimentary basins. The potential complexity of geological basins, with layers of varying material properties and irregular geometries can lead to difficulties in the discretization required for the modeling of physical processes in the basin. In the following work, we consider an implicit representation of the positions of the material layers, through the level set method. The purpose of this is to keep a fixed mesh upon which the physical processes evolve, rather then update the mesh to reflect current rock geometry.

## 1.1 Geological Background

Sedimentary basins are geological formations that are important for understanding the subsurface structure in many parts of the world, see e.g., [1]. A basin is an opening or depression in the crust that is filled with sediments. These can extend on the order of many square kilometers wide and tens of kilometers deep. Over millions of years, the sediments compact and turn into sedimentary rock, a process called lithification. In Figure 1.1, a hewn rock exhibits the strata formed from rock layers of varying chemical compositions and material properties.

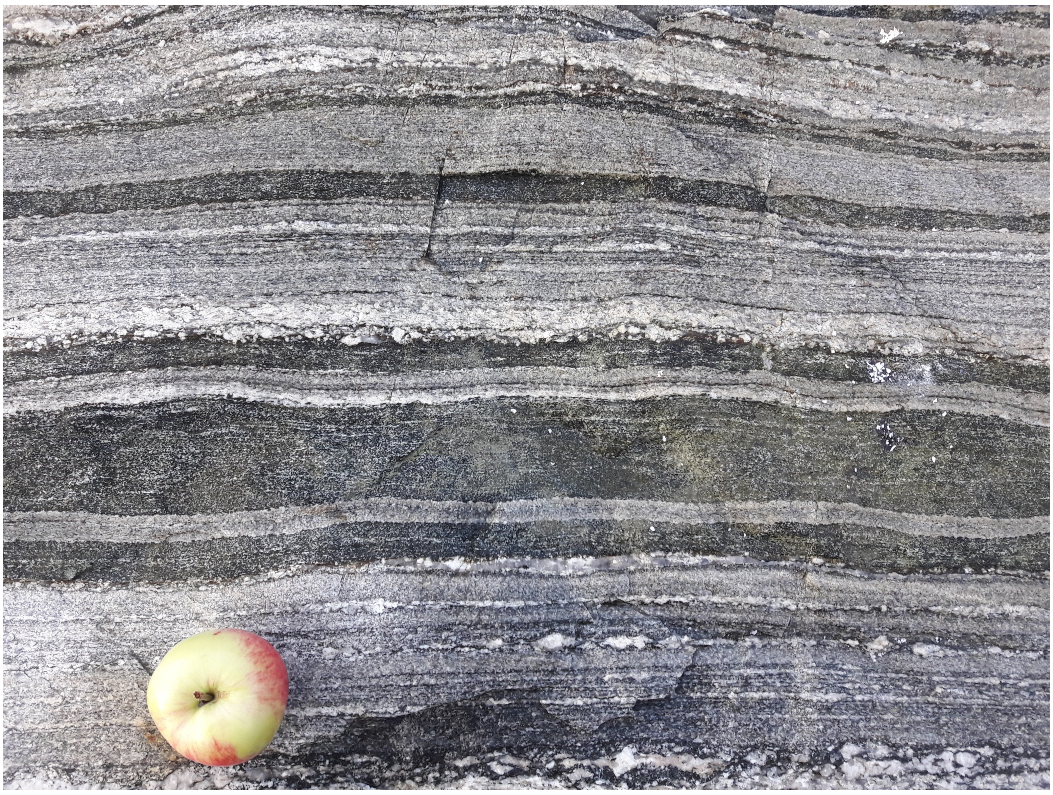

Figure 1.1: This photograph shows a close up of a sedimentary rock, hewn and displayed in Norway. The layering is here shown nicely up close, without significant weathering. The apple sets the scale. Photo taken by author.

As these basins are often in the ocean or at the mouth of rivers, they are saturated with pore fluid (water or brine). During the lithification process, pore fluid is ejected from the sediments. The processes of sedimentation and compaction and their effect on the fluid in the porous medium is the main focus of this work.

Basins can be very complex geometrically, with different rock layers, strata, deforming and folding. Another common feature of basins is fractures, though these are not considered here. The thermal history of basins is important to understanding the evolution of its chemical constituents. Basins can be a rich source of hydrocarbons, and the understanding of and prediction of the generation of oil and gas is an economic motivation for the study of specific sedimentary basins. Exploration is often done through seismic imaging and borehole drilling and analysis. With the development of computing technology, numerical basin modeling is increasingly used to produce estimates of the large scale structure of a basin.

## 1.2 Numerical Modeling

Basins have been modeled as porous media in a continuum mechanics framework, see e.g., [2]. The use of conservation laws is applied to simulating their development over time. The partial differential equations that form a model typically require discrete approximations to the equations in order to be tractable. This is commonly done by creating a mesh over the continous domain, where only a discrete set of values is well-defined. The physical values of the model then are taken to be the solutions of the discrete problem. This is where numerics and the use of computational resources is required.

When the geometry of the domain is irregular, there are consequences for the accuracy of the discrete approximations. In Figure 1.2, we see an example of a rock facade in the Italian Alps near Trento. The facade shows how sedimentary layers formed millions of year ago, persist until today, despite weathering and fracturing of the rock. The main horizontal divisions between layers are the inspiration for the type of interfaces that are to be represented in this model.

In a time-dependent simulation, the location of irregularities might move through the domain, and require a re-meshing to maintain convergent solutions. While there are techniques to re-mesh the domain so that the cells are conforming to the material properties, the current work pursues the idea of keeping the mesh the same, and evolving the material interfaces over the fixed mesh through the use of an implicit interface technique, the level set method. In short, an additional potential function is considered, where the locations of a fixed value (usually 0 value) of this function define the interface. The level set potential function moves according to a speed function, normal to the interface, that determines the trajectory. In this work, to specify the trajectory, we construct a speed function that depends on the dominant sedimentaion rate of added material and a coupling to the pore pressure to incorporate the amount of compaction in the layers. The coupling between the pore pressure and the rate of layer movement, due to compaction and sedimentation, is a novel contribution of this thesis.

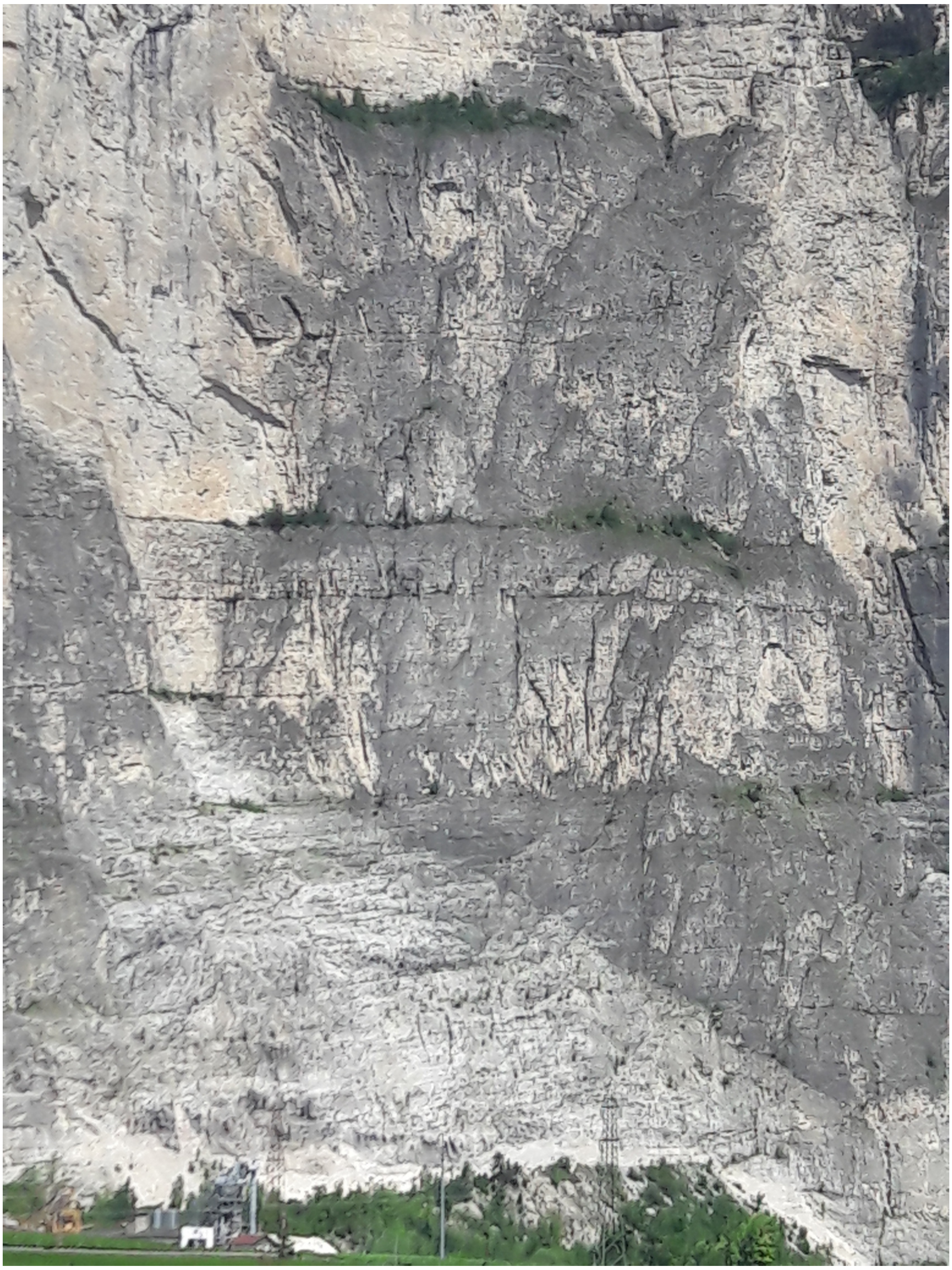

Figure 1.2: This photograph shows sedimentary rock, now exposed due to glaciation, along the Adige river in the Alps outside of Trento, Italy. A white house is visible at the bottom of the facade. Photo taken by author.

## 1.3 Organization

This doctoral thesis is structured in the following way:

Chapter 2 considers the demonstration of principle for the coupling of a physical quantity, i.e., overpressure, to the driver of geometrical change, the speed function for the level set. The setting is 1-D and the discretizations are done with finite differences. A version of this chapter has been published in the journal *Computational Geosciences* [3].

Chapter 3 is a transitional chapter that introduces the High Performance Computing (HPC) capabilities within the context of the generic finite element library deal.II. Its purpose is to provide glue towards keeping the thesis self-contained, and of course introducing notation and some context.

Chapter 4 is focused on elaborating the model presented in Chapter 2 into a finite element method context with 2 spatial dimensions, while maintaining the vertical effective stress assumption, inherent in 1-D. Further elaborations are also considered, including adding a temperature field. The code is structured to take advantage of distributed memory parallelism. A version of this chapter is in preparation for submission to the journal *Basin Research*.

Chapter 5 considers model developments along three axes: (a) code infrastructure for 3-D model runs, (b) the extension of the speed function beyond sedimentation and compaction to something akin to density dependent buoyancy, and (c) this chapter lays the foundations for running the model with dynamic local adaptivity. The last axis, adaptivity, requires some changes to the manipulation of the data structures and introduces some considerations for our explicit solvers. The Level Set solver code that is used has been adapted from Manuel Quezada de Luna's work [1] and, in the adaptive context, requires alterations to the algorithms to function that are beyond the scope of this doctoral thesis. A version of this chapter, describing the progress towards the development of a 3-D adaptive basin simulator, has been submitted to the journal *Geoscientific Model Development*.

Chapter 6 summarizes the doctoral thesis and lays out some extensions to the model that would provide further capabilities and interesting features, in the context of future work.

[1] https://dealii.org/developer/doxygen/deal.II/code_gallery_two_phase_flow.html

CHAPTER 2

# Coupling Geometry to Pore Pressure and Compaction

A version of this chapter was published in McGovern, et al. (2017).

## 2.1 Introduction

Basin modelling has become an increasingly important component to the identification of hydrocarbons in various geological settings, see e.g., [4], [5], [6], [2]. The thermal history and pore pressure of a basin as it evolves over millions of years is vital to estimating the generation, migration and trapping of gas and oil deposits.

The motivation of the present work is to focus on handling significant changes in the stratification of the sample system. Extensional and compressive forces can be large. For example, uplift and salt diapirs are potential causes of large deformations in the layering of the basin. These large deformations require specific computational techniques to render their inclusion in the model tractable. A well-known approach to computing the dynamical evolution of a system is the Lagrangian framework.

In this setting, the coordinate system of the grid or mesh is updated to track the changes in the material as it moves. The grid nodes are identified with material positions. If the material moves significantly, then the grid nodes will as well. This can lead to warps and distorted grid geometry, which degrade the numerical properties of the discrete solution to the governing differential equations. To correct this, one needs to re-grid, which, in general, is an expensive procedure. An alternative to the re-gridding process is to hold the coordinate system of the grid fixed, while moving the material through. Known as the Eulerian framework, this fixed grid approach has the numerical advantages of a well-behaved, regular grid, but has the complication that a material element will, in general, not be identified with a single node. Locating the material elements on the grid, leads to its own challenges. For a general review of the two computational approaches, see [7]. A popular method to address large deformations, the Arbitrary Lagrangian-Eulerian technique, employs a combination in order to seek the best of both worlds, see e.g., [8], [9] and [10].

In constrast, the level set method ([11], [12]) exploits the fixed grid, while introducing a potential function to implicitly locate the interfaces between material elements. Level set methods have been applied to a wide array of domains in the last decades, from computational fluid dynamics (e.g., [13], [14]), groundwater flow [15], crystal growth [16] to image processing [17]. This study uses the level set method to investigate the issue of deformation of the geometry of a sedimentary basin due to mechanical compaction.

As basin modelling is a multi-component and multi-scale approach to complex physico-chemical processes, the focus of the current study is restricted to the problem of fluid flow and compaction that a basin undergoes as layers of sediment contribute to an evolving pressure profile of the basin. The compaction, i.e., reduction in porosity, contributes to changes in pore pressure, thus driving flow, as a result of increasing sediment load and displacement of strata. Furthermore, in this paper, we consider only a one-dimensional model of single phase fluid flow under isothermal conditions. The present work draws on the following studies in compaction and fluid flow.

Schrefler and Scotta (2001) [18] present a fully coupled model of multiphase flow in a deformable poroelastic medium. They use a finite element method to solve for the displacements of the solid matrix and the pressures of the two phases. The other variables are related through constitutive equations. A

Galerkin method is used for the spatial discretization and a Newmark method is used for the temporal discretization. They validate the numerical results with three physical examples over small spatial scales and short time scales. In Minkoff et. al. (2003) [19], coupling is studied through two 3-D finite element codes. With the mixed finite element flow simulator, a black oil model is run with initial values for porosity and permeability to compute a pore pressure. This is, then, communicated to the quasi-static geomechanics simulation, which computes porosity and permeability. These results are passed to the flow simulator and the subsequent time step flow is calculated, and so on. The authors argue for a loose coupling approach essentially since it is computationally less expensive than a fully coupled method. One advantage is that the domains can be quite different for each system. In Bethke (1985) [20], the author provides a numerical model of compaction-driven flow for a 2-dimensional space. The conservation of rock mass, fluid flow and heat transfer each contribute an equation to the coupled system, which is solved iteratively using a finite difference discretization. The problem is formulated using a Lagrangian reference frame that moves with the rock mass. Material added to the top of the basin will form new nodes when sufficient material is added. See also [21] and [22]. Chen et. al (2002) [23] presents a similar model for compacting soils, but differs in that only compaction and fluid flow are considered. The triangulation of the basin is updated at each time step, i.e., the elements are functions of time. The coupled system is iteratively solved using the mixed finite element method. Kikinzon et al (2015) [24] follows the model set forth in [23], while modifying the algorithm for the time-dependent grid structure. They clearly divide the problem into physically distinct steps which are treated in sequence, an operator splitting approach. The presentation of the Theory section below draws on this exposition. Christopher (2006) [25] studies the Ursa region of the Gulf of Mexico by using a 1-D compaction model with finite difference solution method for the fluid pressure. In the present study, as in the above series of papers, an effective stress principle with only vertical compaction is assumed. The same basic model of compaction-driven fluid flow is considered here. Also, the sedimentation step is split from the fluid pressure solution step, pursuing an iterative approach to the coupled partial differential equations.

In contrast to the above series of papers, however, the level set method is introduced in this paper to implicitly track the evolving locations of the rock layers in the sedimentary basin. Work of M.

Longoni, et. al. [26], [27] has already used implicit interface tracking in the context of sedimentary basin modelling. This approach does not explicitly model mechanical compaction, as each layer is thought of as an immiscible non-Newtonian fluid. Solved with an adaptive finite element method, the Stokes equations form the core of their model, with a level set advection framework to keep the fluid layers separate and an arbitrary Lagrangian-Eulerian framework to represent the boundary motion of the basin under deposition. To the best of the authors' knowledge, the current work is the first application of the level set method to the problem of modelling a sedimentary basin as a porous media fluid-structure interaction problem.

In our application, we use the level set method in the following way. Adopting an operator splitting approach, first the layer interfaces are propagated in response to sedimentation. The state and material properties at the nodes are changed to reflect the adjusted interfaces. In other words, the rock properties are transported through the domain according to the level set interfaces, rather than the grid nodes being moved around. Then, the fluid pressure equation is solved and a compaction rule is applied to determine the contribution of compaction to the movement of the interface in the next time step.

Thus, the rock is effectively advected through the domain via the level set interfaces, with the goal of avoiding the complex and expensive update rules for moving grids. This is the novelty of the approach considered here. It has two main upshots: 1) simplicity of the grid and 2) flexibility of the speed function. The grid remains regular, which is advantageous from an implementation point of view. Secondly, the setup of the coupling between physics and geometry, allows for free choice of the physical models that move the interfaces. What is required is to have a speed function which determines the movement of the front at each point on the interface.

## 2.2 Theory

### 2.2.1 Effective Stress Models

The pores, or void spaces, in a solid material allow for fluid, liquid or gas, to occupy the space. The fraction of the pore volume to the total volume is named the porosity. As stress exerted on the material affects its porosity, this in turn has consequences for the fluid flow through the porous medium. A central tool to modelling flow in porous material is the concept of effective stress introduced in [28]. The effective stress, $\sigma'$ [ML$^{-1}$T$^{-2}$], is given by the total stress, $\sigma$ [ML$^{-1}$T$^{-2}$], minus the pore pressure, $p$ [ML$^{-1}$T$^{-2}$].

$$\sigma' = \sigma - p. \tag{2.1}$$

This approach assumes that the stress on the solid matrix is a result of solid particles contacting other particles. The pressure of the fluid in the solid matrix exerts a force on the constituent solids that is homogeneous and so does not contribute to deformation of the solid matrix. For this reason, the fluid pressure is subtracted from the total stress to isolate the effective stress that the solid matrix experiences. In the context of the one dimensional model described in this paper, we consider effective stress in only one direction, a scalar, which we refer to as the vertical effective stress.

The importance of the effective stress hypothesis is that it directly relates the fluid pressure, essential for describing the flow dynamics, to the stress which is exerted on the solid skeleton of the medium, the mechanical aspect. For an historical perspective on the development of porous media mechanics until recently, see [29].

### 2.2.2 Consolidation

Consider a consolidation problem, where sediment is added to the top of a soil column, as in [30]. The sediment stack deforms continuously under its own growing weight. The fluid pressure, above the hydrostatic pressure, in the saturated pore space holds open the pores, resisting compaction. Extending this to different types of sediment, i.e., with different lithological properties, that are added at different stages, creates the stratification that we conceptualize as rock layer interfaces.

Following [2], the Darcy flow relation between the discharge velocity of the fluid (relative to the solid) and the overpressure gradient, including gravity, is assumed,

$$\mathbf{v} = -\frac{\mathbf{k}}{\mu}\nabla\left(p - \rho_f g z\right), \tag{2.2}$$

where $\mathbf{v}$ [LT$^{-1}$] is the discharge velocity of the pore fluid, $\mathbf{k}$ [L$^2$] is the permeability of the rock, $\mu$ [ML$^{-1}$T$^{-1}$] is the viscosity of the fluid, $p$ [ML$^{-1}$T$^{-2}$] is the pore pressure, $\rho_f$ [ML$^{-3}$] is the density of the fluid, $z$ [L] is the spatial position, and $g$ [LT$^{-2}$] is the acceleration due to gravity.

The mass balance condition requires that any divergence be compensated by a reduction in the amount of mass contained,

$$\nabla \cdot \mathbf{v} = -\frac{1}{1-\phi}\frac{\partial \phi}{\partial t} + \frac{1}{\rho_f}\frac{\partial \rho_f}{\partial t}, \tag{2.3}$$

with $\phi$ [L$^3$L$^{-3}$] the porosity. The necessary reduction in mass comes from a change in the porosity, i.e., a reduction in volume for the fluid, or a change in fluid density. Neglecting changes in the fluid density, the last term in Eq. 2.3 is dropped, and inserting Eq. 4.2 into Eq. 2.3 gives

$$\nabla \cdot \left(-\frac{\mathbf{k}}{\mu}\nabla\left(p - \rho_f g z\right)\right) = -\frac{1}{1-\phi}\frac{\partial \phi}{\partial t}. \tag{2.4}$$

The change in porosity can be expressed as

$$\frac{\partial \phi}{\partial t} = \frac{\partial \phi}{\partial \sigma_z'}\frac{\partial \sigma_z'}{\partial t} = -C\frac{\partial \sigma_z'}{\partial t}, \tag{2.5}$$

where $C$ [M$^{-1}$LT$^2$] is the compressibility, a lithological property dependent on $\sigma_z'$, which is the effective stress in the z direction, as introduced previously. Recalling that effective stress is the total stress minus the pore pressure, $\sigma_z' = \sigma_z - p$, Eq. 4.6 is combined with Eq. 2.4 to arrive at the pore pressure equation

$$-\nabla \cdot \left(\frac{\mathbf{k}}{\mu}\nabla\left(p - \rho_f g z\right)\right) = \frac{C(\sigma_z')}{1-\phi}\frac{\partial\left(\sigma_z - p\right)}{\partial t}. \tag{2.6}$$

Recall that for the purposes of this paper we consider only a one-dimensional model:

$$-\frac{\partial}{\partial z}\left(\frac{k}{\mu}\frac{\partial}{\partial z}(p - \rho_f g z)\right) = \frac{C(\sigma_z')}{1-\phi}\frac{\partial(\sigma_z - p)}{\partial t}. \tag{2.7}$$

Introducing the overpressure, defined as pore pressure minus the hydrostatic pressure, $p_o = p - \rho_f g z$, for constant fluid density, and rearranging, we have

$$\frac{\partial p_o}{\partial t} - \frac{1-\phi}{C\left(\sigma_z'\right)}\frac{\partial}{\partial z}\left(\frac{k}{\mu}\frac{\partial}{\partial z}p_o\right) = \frac{\partial\left(\sigma_z - \rho_f g z\right)}{\partial t}. \tag{2.8}$$

The total overburden at a depth $z$ is given by the weight of the material above,

$$\sigma_z = \int_0^z \rho_b\left(z'\right) g\, dz', \tag{2.9}$$

where the bulk density, $\rho_b$, is the porosity-dependent average density of the material, i.e.,

$$\rho_b = (1-\phi)\rho_s + \phi\rho_f, \tag{2.10}$$

with $\rho_s$ the density of the rock matrix and $\rho_f$ the fluid density. The right hand side of Eq. 2.8 becomes

$$\frac{\partial\left(\sigma_z - \rho_f g z\right)}{\partial t} = \frac{\partial}{\partial t}\left(g\int_0^z \left(\rho_s - \rho_f\right)\left(1-\phi\left(z'\right)\right)\, dz'\right). \tag{2.11}$$

This is the source term for the overpressure generation, the rate of change of the difference between the overburden and the hydrostatic pressure. The equation includes three lithological properties: the compressibility, the permeability and the rock density, which can vary vertically.

For mathematical closure, boundary and initial conditions must be specified. The top boundary condition is chosen to be a fixed atmospheric pressure, $p_0$,

$$p(z=0) = p_0 \tag{2.12}$$

for all $t > 0$. The bottom boundary condition is chosen to be a no flow condition,

$$\frac{\partial p}{\partial z} = 0 \tag{2.13}$$

for all $t > 0$. For the initial condition, $t = 0$,

$$p(z) = p_0 \tag{2.14}$$

In addition to the pore pressure equation, it is necessary to estimate the porosity as a function of depth or effective stress. A number of phenomenological models are presented in [2]. A good selection is the simple exponential relation known as Athy's Law [31],

$$\phi = \phi_0 e^{-\beta\sigma'_z}, \tag{2.15}$$

where $\phi_0$ is an initial porosity and $\beta$ is a compaction parameter. [32] provide an analysis of the suitability of assuming this exponential relation. The main variables for the physical model considered here are summarized in Table 2.1.

| Physical quantity | Symbol | Dimension (SI unit) |
|---|---|---|
| pore pressure | $p$ | $ML^{-1}T^{-2}$ (Pa) |
| overpressure | $p_o$ | $ML^{-1}T^{-2}$ (Pa) |
| permeability | $k$ | $L^2$ ($m^2$) |
| viscosity | $\mu$ | $ML^{-1}T^{-1}$ (Pa s) |
| densities | $\rho$ | $ML^{-3}$ (kg $/m^3$) |
| vertical effective stress | $\sigma'_z$ | $ML^{-1}T^{-2}$ (Pa) |
| overburden | $\sigma_z$ | $ML^{-1}T^{-2}$ (Pa) |
| rock matrix compressibility | $C$ | $M^{-1}LT^2$ (Pa$^{-1}$) |
| porosity | $\phi$ | $L^3L^{-3}$ (-) |
| void ratio | $e$ | $L^3L^{-3}$ (-) |
| compaction parameters | $\alpha$,$\beta$ | $M^{-1}LT^2$ (Pa$^{-1}$) |

Table 2.1: Main physical variables considered.

### 2.2.3 Level Set Methods

When using level sets, the changing geometry is represented by evolving interfaces which are implicitly defined over a fixed grid [33]. Again, the motivating idea is that a fixed grid has the virtue of computational simplicity relative to frequently remapping the grid. Of course, there is a price to be paid, namely the introduction of additional partial differential equations to solve.

As mentioned above, the key idea is to implicitly track the evolution of interfaces through the use of an additional equation, namely that for a higher dimensional function, the level set potential function, [34]. The curve that is to be tracked, $\Gamma$, is embedded in the level set potential function as the 0 level set, meaning all the points where the potential function is 0 represent $\Gamma$. The level set function, $u$, is defined such that at $t = 0$, its value at $x$ is the signed distance, $d$, from $x$ to $\Gamma$, i.e.,

$$u(x, t = 0) = d. \tag{2.16}$$

The time evolution of $u$ is given by an advection equation for the level set function,

$$\frac{\partial u}{\partial t} + \vec{w} \cdot \nabla u = 0, \tag{2.17}$$

where $\vec{w}$ is the velocity field for the interface motion and $u$ is the level set function . The interface is moved along in the direction of the velocity field. So, at any time going forward, $\Gamma$ remains identified by all the points where the level set function has value 0.

Defining the speed function, $F$, to be the the magnitude of the vector field normal to the front surface, $F = \vec{w} \cdot \hat{n}$, and with the definition of the unit normal vector, $\hat{n} = \frac{\nabla u}{|\nabla u|}$, Eq. 4.24 can be rewritten as

$$u_t + F|\nabla u| = 0. \tag{2.18}$$

The velocity field for the interface motion, $\vec{w}$, or alternatively the speed function, $F$, plays a determinative role in this scheme. For any particular physical application, $F$ must come from the dynamics of the system. For applications that solve the Navier-Stokes equation, the speed function can be taken to be the fluid velocity, [35]. [36] use the level set method to track the evolution of resin being infused into a porous material. This problem is formulated on the assumption of Darcy flow, and the average fluid velocity is used as the speed function for the level set evolution.

The movement of the level set is determined by the relevant physics of the problem considered. This is a major feature of the method in that $F$ can be constructed to take into account various physical phenomena at different levels of description. In the current section, we propose a speed function to capture the dynamics of the consolidation problem considered. The interfaces to be tracked are defined by the different rock types. The speed at which these interfaces move is mostly determined by the sedimentation rate, i.e., the rate at which material is added to the column. Further, there should be a contribution from the compaction of the layers. The following procedure constructs a speed function that satisfies this dependence on sedimentation rate and rate of compaction of the layers. It is not offered as a rigorous derivation, but rather as a concrete formula for the coupling.

Beginning with Equation B.2 in [2], we have from the conservation of rock mass,

$$\nabla \cdot \mathbf{v_s} = \frac{1}{1-\phi}\frac{d\phi}{dt}, \tag{2.19}$$

where $\mathbf{v_s}$ is the velocity of the rock relative to the fixed coordinate system. Integrating in one dimension from the top of the column to a point $z'$ in depth,

$$\int_0^{z'} \frac{dv_s}{dz} dz = \int_0^{z'} \frac{1}{1-\phi}\frac{d\phi}{dt}\, dz. \tag{2.20}$$

On the left hand side, we have

$$v_s|_{z'} - v_s|_0, \tag{2.21}$$

the velocity at the depth $z'$ minus the velocity at the surface. With a constant time step and grid spacing, substituting the discrete differences for derivatives, the integration is then approximated as summation over the quantities evaluated on the discrete grid points, leading to

$$F = v_s|_{z'} = v_s|_0 + \sum_{i=0}^{z'} \frac{1}{1-\phi_i}\frac{\Delta\phi_i}{\Delta t}\, \Delta z. \tag{2.22}$$

The rock velocity is then used in the level set equation as the speed function.

For numerical reasons, the level set function deviates from being a signed distance function as the iterations of the advection equation increase [34], [37]. Therefore, [38] proposed a procedure to reinitialize the level set function back to being a signed distance function. In general, there will be some error introduced into the position of the interface from solving the reinitialization equation. For example, in a two-phase fluid problem with different fluid densities, this affects the mass conservation. Therefore, the recommended frequency with which this procedure should be applied depends on the application [39]. However, in our one dimensional model, reinitialization does not affect the zero level set, but only the values of the level set function off the interface (since we always know the distance to the interface on the line). When the slope around the interface becomes too flat or steep, depending on some tolerance, the level set is reinitialized.

Another technical aspect of the level set method that is used in our model is extending the interface speed [34], [40]. A speed must be assigned to regions neighboring the interface, as well as directly on it. In our case, the speed function computed within the layer above the bounding interface is extended into the layer below. This is required for smoothness reasons, since we have rock layers with heterogeneous properties, and is implemented by taking the slope in the speed function above the interface and extending the speed function linearly for a few nodes (a heuristic number has been the ratio of rock length added in a time step to the dicretization step). This amounts to something similar to upwinding of the speed of the rock layer above into the top of the layer below.

### One dimensional example

As a demonstration, for the one dimensional case of an interface moving with speed $F$, one time step in the propagation of the front labelled $\Gamma_1$, is illustated in Fig. 2.1. The interface between the blue section and the brown section, $\Gamma_1$, starts at a position, $z_0$, and moves to $z_1$ in the next time step. $z_1$ is computed through solving the advection equation for the level set function, $u$. The level set function, $u$, is plotted beneath a graph of the real coordinates. Given an intial level set function, $u^{t_0} = z - z_0$, Eq. 4.25, can be used to find the level set function at the next time step, $u^{t_1}$. Using a simple forward discretization in time gives,

$$\frac{u^{t_1} - u^{t_0}}{\Delta t} + F = 0, \tag{2.23}$$

recalling that $u$ was initially defined as a signed distance function, $|\nabla u| = 1$. Rearranging gives,

$$u^{t_1} = u^{t_0} - F\Delta t = z - z_0 - F\Delta t. \tag{2.24}$$

Inspection of $u$ for $u^{t_1} = 0$ gives the location of the interface. So, $z_1 = z_0 + F\Delta t$, and the interface is moved forward appropriately. The position, $z_1$, of the interface, $\Gamma_1$, is found through solving the advection equation for $u$, hence an implicit tracking method. The linear level set function gives the zero dimensional point of the interface. In two spatial dimensions, the level set function is also two

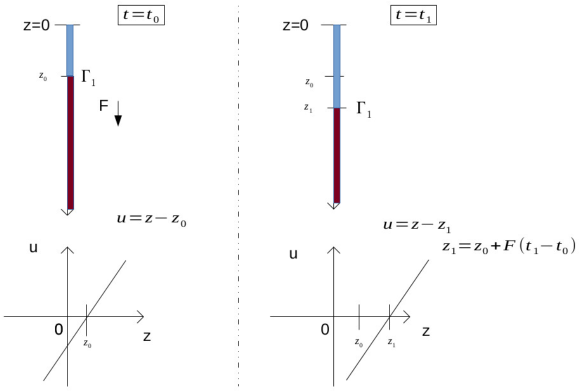

Figure 2.1: One time step in the level set evolution.

dimensional, with the interface taking values in a one dimensional space. This is the essence of the level set method, i.e., to manipulate a higher-dimensional function to infer information about the physically important front location (zero level set).

## 2.3 Methodology

### 2.3.1 Model Outline

The work flow that is implemented to carry out the consolidation simulation is shown in Fig. 2.2. First, the bulk rock material is deposited at a constant rate $\omega_b$, expressed directly in terms of length per time [m/s]. Second, the speed function is computed. Third, the level set equation is solved. Equation 4.25 is solved through a backward Euler implicit finite difference approximation. The level set function is initialized to a signed distance function at the beginning of the simulation and reinitialized depending on the slope around the 0 level set. Fourth, depending on the movement of the rock layer, the grid nodes are updated with the physical states. Fifth, the pressure equation, Eq. 2.7 or an analogous equation expressed in terms of a different pressure variable, is solved through a centered implicit finite difference approximation. In order to complete one time step iteration of the model, the effective stress and porosity are updated. Equation 4.5 or a linear compaction relation is then used to update the porosity distribution in the column. The next time step then begins again with sedimentation.

This operator splitting approach separates the sedimentation and the compaction steps into

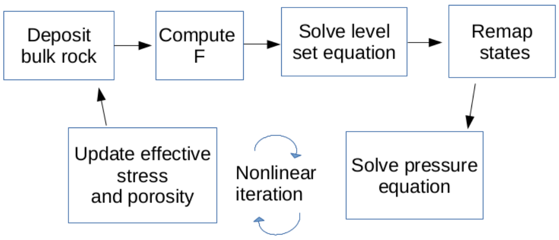

Figure 2.2: Flow chart of simulation, beginning at top left.

sequential events. In order to enforce consistency between the porosity distribution and the pore pressure distribution, a Picard iteration loop was implemented over the pore pressure solution and the compaction law update. The solution to Eq. 2.8 is used in computing the effective stress, Eq. 2.1, which determines the new porosity from Eq. 4.5 (or similar). This is done because of the nonlinearity introduced into Eq. 2.8 by considering the porosity to be a function of effective stress, as in Eq. 4.5. By using a known value from a previous step, the Picard iteration is used to linearize the nonlinear appearance of the pore pressure in the compaction law.

In the case of one layer with a linear compaction law, the accuracy of this model can be compared to an analytical solution available in the literature, and in particular as presented by Wangen (2010) [41]. The details of which are laid out next.

### 2.3.2 Verification with Analytical Solution

Assuming a linear dependence of the void ratio on the effective stress and a linear dependence of the permeability on the void ratio, there is one available analytical solution for the consolidation problem considered here. [41] uses a coordinate transform, depending on the local porosity, to remove the pore space from the equation. In these “compaction-free" coordinates, the solution can be expressed analytically.

The transform between the real coordinate, $z$, and the Lagrangian compaction-free coordinate is

$$z = \int_{\zeta_1}^{\zeta_2} \frac{d\zeta}{1-\phi} \tag{2.25}$$

This serves to scale the spatial coordinate by the porosity. In the $\zeta$ coordinates, there is no pore space, therefore the quantities of bulk density and bulk deposition rate need to be rendered as just the solid rock properties. Therefore, $\rho_b$ and $\omega_b$ are changed to $\rho_s$ and $\omega_s$, the density of rock and the deposition rate (still length/time) of pure rock, respectively.

The void ratio was chosen to scale linearly with the effective stress, $e = e_0 - \alpha\sigma'_z$, where $\alpha$ [M$^{-1}$LT$^2$] is a compaction coefficient and $e_0$ [L$^3$L$^{-3}$] is the void ratio at the surface. Since $e = \frac{\phi}{1-\phi}$, one can see that the porosity dependence on effective stress is slightly nonlinear. For the comparison with the numerical model considered, which is formulated in terms of the porosity, it is important to note that

$$\frac{\partial e}{\partial t} = \frac{1}{(1-\phi)^2}\frac{\partial \phi}{\partial t} \tag{2.26}$$

and

$$\frac{\partial e}{\partial \sigma'_z}\frac{\partial \sigma'_z}{\partial t} = \frac{1}{(1-\phi)^2}\frac{\partial \phi}{\partial \sigma'_z}\frac{\partial \sigma'_z}{\partial t}. \tag{2.27}$$

Therefore, the compressibility is

$$C(\sigma'_z) = \frac{\partial \phi}{\partial \sigma'_z} = -\alpha(1-\phi)^2. \tag{2.28}$$

The permeability is taken to be a function of the void ratio. The choice of $k = k_0(\frac{1+e}{1+e_0})$, simplifies the pressure equation, and mimics the reduction in permeability due to a decrease in the void ratio.

With these assumptions in place and exploiting the coordinate transform, the pressure equation considered is

$$\frac{\partial p_o}{\partial t} - \frac{k_0}{(1+e_0)\alpha\mu}\frac{\partial^2 p_o}{\partial \zeta^2} = (\rho_s - \rho_f)g\omega_s. \tag{2.29}$$

Equation 2.29 is a constant coefficient diffusion equation with a constant source that has a solution of the form:

$$\begin{aligned} p_o(\zeta,t) = \gamma\omega_s t - \gamma(\pi c t)^{-\frac{1}{2}}\exp\left(\frac{-\zeta^2}{4ct}\right)\ldots \\ \int_0^\infty \xi \tanh\left(\frac{\omega_s \xi}{2c}\right)\cosh\left(\frac{\zeta\xi}{2ct}\right)\exp\left(\frac{-\xi^2}{4ct}\right)d\xi, \end{aligned} \tag{2.30}$$

where $\gamma = (\rho_s - \rho_f)g$ and $c = \frac{k_0}{(1+e_0)\alpha\mu}$.

The coordinate transform can then be used to convert back to the real coordinates, $z$, using the porosity distribution computed from the pressure solution.

### Dimensionless Form

It is possible to perform a variable transform to non-dimensional coordinates. Scaling the Lagrangian depth coordinate by $\overline{\zeta} = \frac{\zeta}{\omega_s t_{tot}}$, where $t_{tot}$ is the total time period simulated and $\omega_s$ is still the deposition rate of pure rock. Time is scaled as $\overline{t} = \frac{t}{t_{tot}}$. The overpressure is scaled as $\overline{p_o} = \frac{p_o}{(\rho_s - \rho_f) g \omega_s t_{tot}}$. Similarly, the VES is scaled as $\overline{\sigma_z'} = \frac{\sigma_z'}{(\rho_s - \rho_f) g \omega_s t_{tot}}$. Carrying out the variable substitions in Eq. 2.29 leads to,

$$\frac{\partial \overline{p_o}}{\partial \overline{t}} - \overline{D} \frac{\partial^2 \overline{p_o}}{\partial \overline{\zeta}^2} = 1, \tag{2.31}$$

where $\overline{D} = \frac{k_0}{(1+e_0)\alpha\mu{\omega_s}^2 t_{tot}}$ is the dimensionless diffusivity, which is time independent. One more dimensionless quantity is necessary to describe the compaction law, $e = e_0 - \alpha\sigma_z'$. With the VES variable substitution, we get

$$e = e_0 - \overline{\alpha}\overline{\sigma_z'}, \tag{2.32}$$

where $\overline{\alpha} = \alpha(\rho_s - \rho_f) g \omega_s t_{tot}$ is the dimensionless compaction coefficient.

In this formulation, the behavior of the dynamical evolution can be studied based on the two dimensionless quantities, diffusivity and compaction coefficient, thus reducing the parameter space of the of the system. In Fig. 2.3, across five orders of magnitude in dimensionless diffusivity, we can see the effective behavior of the overpressure, spanning from almost hydrostatic to almost lithostatic. Similarly, in Fig. 2.4, one can see large compaction when the overpressure is small and essentially no compaction when the overpressure is large. Thus, for a chosen dimensionless compaction coefficient, we can see the “full" range of behavior through considering the dimensionless form of the equations.

In Wangen (2010) [41], a similar transformation is carried out, though it differs in the treatment of time. Wangen considers the characteristic pressure that would be needed to see noticeable compaction. From this characteristic pressure, a characteristic length and time are derived. In particular, Wangen introduces the so-called gravity number, $N = \frac{k_0(\rho_s - \rho_f)g}{(1+e_0)\alpha\mu}$ and the dimensionless time, $\tau = \alpha(\rho_s - \rho_f) g \omega_s t$. One can see that, up to the definition of time used, $\tau$ is the same as $\overline{\alpha}$ and,

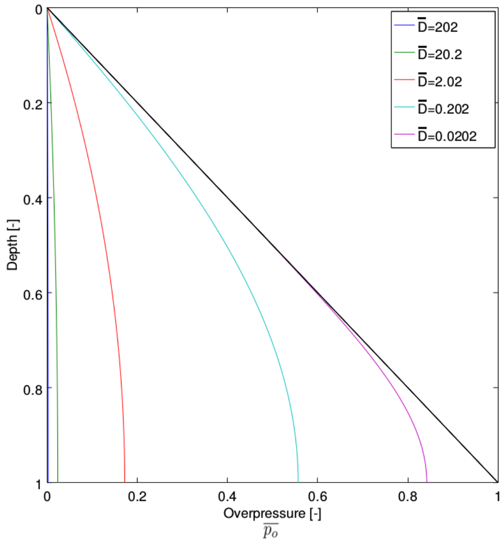

Figure 2.3: Overpressures for different dimensionless diffusivities at a dimensionless compaction coefficient of 0.26. A larger diffusivity gives a smaller overpressure. The black diagonal line represents the lithostatic pressure.

furthermore, $\frac{N}{\tau} = \overline{D}$. In [41] the formulation uses a characteristic time, thus making the scaling of the dimensionless variables time dependent. We use the total time of simulation to scale the variables, leading to a time-independent dimensionless diffusivity and compaction coefficient.

### 2.3.3 Plausibility Checks

For the multi-layer case, there is no analytical solution available. Therefore, we introduce some criteria for evaluating these simulations. First, given the assumed dynamics of consolidation, except when a layer is actively being deposited, the layers should always decrease in size. This translates into a constraint on the speed function. The speed at a given layer interface should always be less than the speed at the layer interface above it, in order to be physically consistent. This monotonic decrease of the speed function with layer interface depth can be violated, for example, with a choice of a very low porosity layer above a very high porosity layer. In this case, the bounding interface below the very high porosity layer would move faster than the bounding interface below the very low porosity layer, thus leading to an increasing thickness of the very low porosity layer. Second, as in the case of the single layer, if the pore pressure becomes larger than the lithostatic pressure, leading to a negative

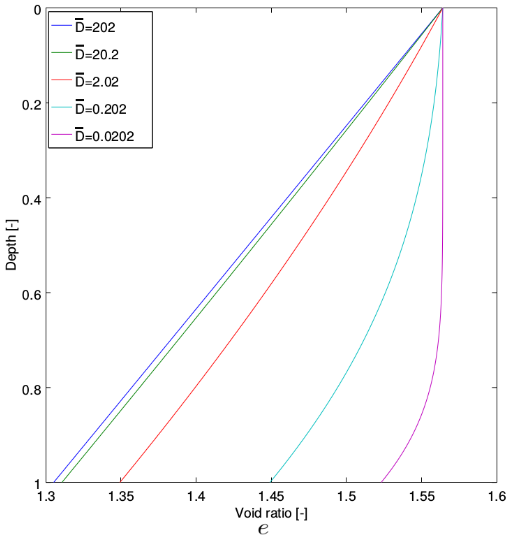

Figure 2.4: Void ratios for different dimensionless diffusivities at a dimensionless compaction coefficient of 0.26.

vertical effective stress, this represents an unphysical situation for the proposed model. Physically, if the pore pressure approaches the lithostatic pressure, compaction stops and hydrofracturing, the opening of the pore space, occurs. No faulting or fracturing processes are considered in our model.

Furthermore, by varying the compaction coefficients and the permeabilities of the layers, one can create conditions for overpressure buildup. A seal forms when the permeability of an overlying layer is significantly less than that of the layer beneath (e.g., an order of magnitude). The pore fluid in the lower layer has difficulty flowing upward, and thus, with continual deposition, the pore pressure increases.

## 2.4 Results and Discussion

The reference example for the following numerical experiments is based upon a similar set-up as in [41]. The test parameters are one kilometer of uncompacted rock deposited over one million years. The single rock type is deposited with an initial porosity of 0.61, and a surface permeability of $10^{-18}m^2$, with a rock density of $2720\frac{kg}{m^3}$. The compaction coefficient was chosen to be $5 \cdot 10^{-8}Pa^{-1}$. This parameter plays a crucial role, and essentially determines the coupling strength between the

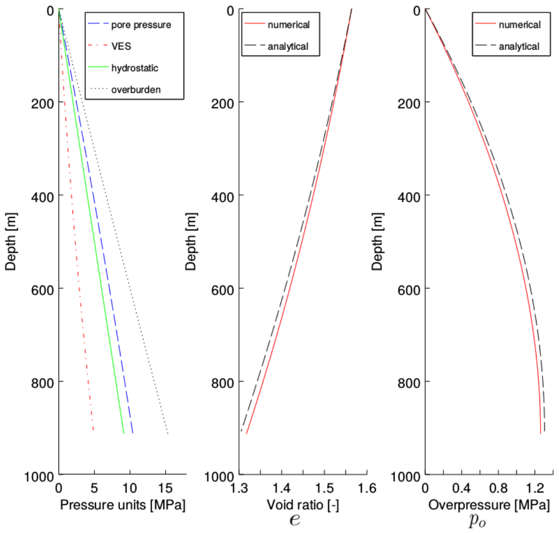

Figure 2.5: Physical variables after one million years of deposition: Linear compaction law. A uniform compaction coefficient of $5 \cdot 10^{-8} Pa^{-1}$ was used. The initial porosity was 0.61 and the initial permeability was $10^{-18} m^2$ with a discretization of 2.5m. Note, VES=vertical effective stress.

vertical effective stress (VES) and the porosity. For the linear void ratio compaction law, the depth of the basin was compared to the analytical value for different values of the dimensionless quantities defined in section 2.3.2.

### 2.4.1 Verification

#### Single Layer with Linear Compaction

First we present a plot of the relevant physical variables after deposition, using a discretization of 100 time steps and a grid spacing of 2.5 m. In Fig. 2.5, we see that the pore pressure is bounded between the lithostatic pressure and the hydrostatic pressure, which is physically consistent. Even though the solution variable is the overpressure in the implementation, adding the hydrostatic pressure results in the pore pressure, which is plotted here.

Using the solution for the overpressure in Lagrangian coordinates as put forward in [41], one can compute the VES, and the porosity distribution. Using the coordinate transform, one then arrives back at the real coordinates to find the height of the column. Thus, for the parameters used here, the height

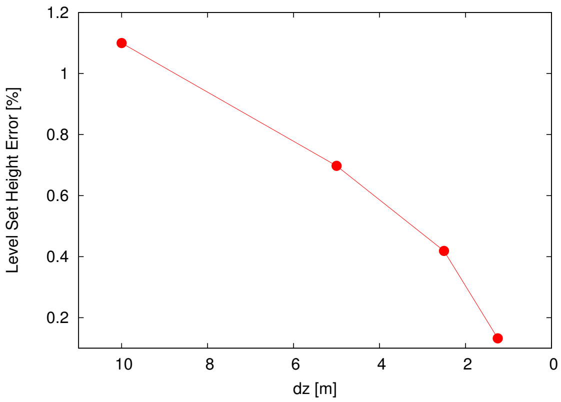

Figure 2.6: Error in the level set height calculation. Using 100 time steps per simulation, the spatial resolution, dz, was increased from 10m to 1.25m. The initial porosity was 0.61 and the initial permeability was $10^{-18} m^2$ and compaction coefficient of $5 \cdot 10^{-8} Pa^{-1}$ .

| $\overline{D}$ [-] | $\overline{\alpha}$ [-] | Height Error [%] |
|---|---|---|
| 202 | 0.260 | 1.24 |
| 20.2 | 0.260 | 1.10 |
| 2.02 | 0.260 | 0.295 |
| 0.202 | 0.260 | -0.939 |
| 0.0202 | 0.260 | -0.851 |
| 8.09 | 0.649 | 5.70 |
| 0.809 | 0.649 | 1.10 |
| 27.0 | 0.195 | 0.613 |
| 27.0 | 0.389 | 2.51 |
| 27.0 | 0.779 | 9.56 |
| 27.0 | 1.56 | 49.9 |

Table 2.2: Percent error for different values of the dimensionless diffusivity, $\overline{D}$, and dimensionless compaction coefficient $\overline{\alpha}$. Discretization of 2.5 m and 100 time steps used. The first five entries are the same $\overline{\alpha}$ and $\overline{D}$ values as the analytical solution shown in Fig. 2.3 and Fig. 2.4. Note, a positive percent difference is when the level set height is bigger than the analytical solution (i.e., less compaction). The two negative percentages indicate that the numerical solution was shorter (i.e, more compaction) than the analytical solution. The last entry in the table is a simulation at the limits of the analytical solution's validity.

of the column after one million years should be 907 m . The level set depth was 911 m for an error of 0.4%, which is acceptable in our opinion. The convergence of the numerical solution to the analytical solution for decreasing spatial discretization size is shown in Fig. 2.6.

To understand the parameter space of the simulation domain, the error of the level set height calculation for selected values of the dimensionless diffusivity and compaction coefficient are shown in Table 2.2.

It is necessary to consider the relationship between the diffusivity and the compaction coefficient in looking at the limits of the model using the linear compaction law. If the diffusivity is low, then the pore pressure will increase more quickly with compaction coefficient, leading to a situation where the pore pressure becomes higher than the lithostatic pressure, i.e. negative VES. On the other hand, when the overpressure is close to zero, then the VES is near its maximum and there is strong compaction. Depending on the choice of the compaction coefficient, this can lead to negative void ratios, hence invalid simulations. While that is clearly non-physical, even very small positive void ratios are outside the bounds of reasonable domain applicability of the model. The last entry in Table 2.2 corresponds to a simulation where at the bottom of the sediment stack, according to the analytical solution, the void ratio is 0.0347 or 3% in porosity. ($\overline{\alpha} = 1.56$ corresponds to $\alpha = 2.4 \cdot 10^{-7} Pa^{-1}$, given the other parameter values.) This porosity is clearly much smaller than pure mechanical compaction alone can reach. This unphysical situation would not occur with a compaction law where the compaction coefficient changes suitably with the void ratio, e.g. an exponentional compaction law. Therefore, since such large compaction coefficients are outside the scope of physical applicability, that simulation, and its large error, is largely irrelevant, and has been included for completeness.

In summary, the compaction coefficient plays a fundamental role in determining the strength of the nonlinear coupling. The method is more accurate the less compaction that occurs. This coheres with the fact that there is no iteration over the coupling between the level set movement and the pressure solution. Thus, the stronger the compaction, the more the divergence. For compaction coefficients smaller than say, $1 - 5 \cdot 10^{-7} Pa^{-1}$, depending on choice of diffusivity, we conclude the model shows agreement with the analytical solution. This includes the regime of typical physically reasonable lithological compaction coefficients.

### Multi-layer with Linear Compaction

In order to perfom a simulation with four rock types, at least four level sets are required. Each interface is assigned its own level set potential function, with the 0 value defining the location of the interface. When considering four rock types, we have the three interfaces between them and one between the

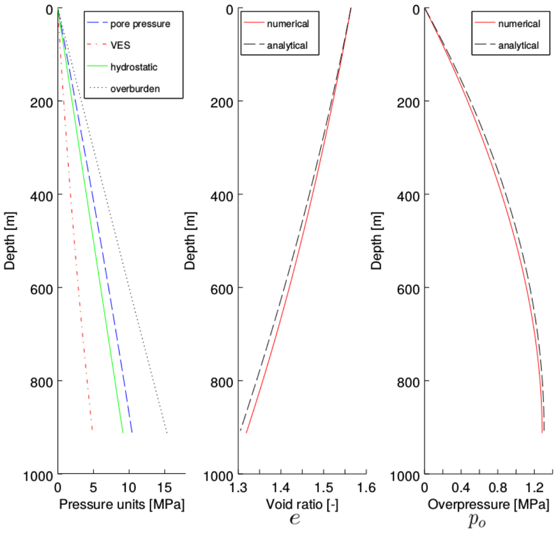

Figure 2.7: Multiple layers, single rock type: linear compaction law. A uniform compaction coefficient of $5 \cdot 10^{-8} Pa^{-1}$ was used.The initial porosity was 0.61 and the initial permeability was $10^{-18} m^2$ with a discretization of 2.5m.

“basement" and the deepest rock. Each level set is solved independently, though the same speed function relation, Eq. 2.22, is shared among them. For a consistency check, we compare the multiple layers, initialized all with the same rock type, in order to compare to the single layer case. For the setup with an analytical solution of 907 m, the multiple layer, single rock numerical model simulates 912m, compared to 911m for the explicitly single layer case above. The physical variables are plotted in Fig. 2.7. We conclude this is successful for the multiple layer implementation.

### 2.4.2 Plausibility Checks

#### Multi-Layer with Linear Compaction

Introducing a heterogeneous permeability distribution in the column, (Fig. 2.8c) leads to differences in the pore pressure shown in Fig. 2.8. If a layer of significantly lower permeability is above a layer of higher permeability, then overpressure will build up, as the pore fluid is trapped. We can see that the pore pressure approaches the lithostatic pressure (overburden) under the seal, located at circa 500 m, which comes from a 2 order of magnitude permeability difference between the two middle layers.

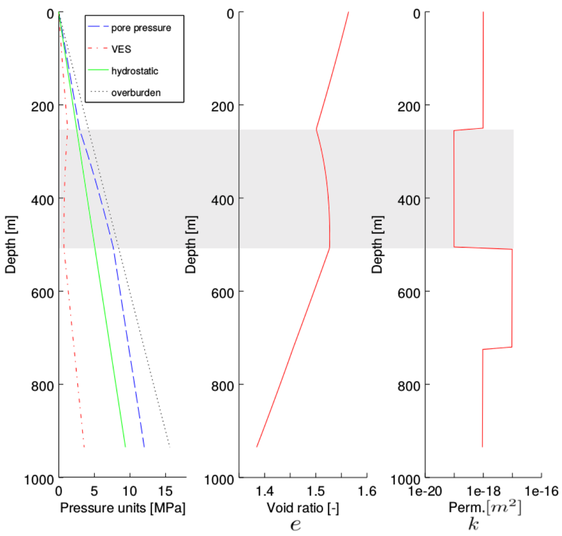

Figure 2.8: Strong permeability differences. Around 500m in depth, there is a build up of overpressure in the layer beneath the low permeability layer, indicative of the behavior of a seal.

This constrasts to to the pore pressure profile in Fig. 2.7, where there is a smooth permeability profile.

The different layers can also have different compaction coefficients. The differences in compaction are clearly seen in the void ratio plot in Fig. 2.9. We can see in Fig. 2.10 how the four layers grow during their period of deposition and how the layers are reduced at different rates, depending on their compaction coefficient. In the simulation, the second layer from the bottom has a larger compaction coefficient than the other layers.

The thicknesses of the different layers are given in Fig. 2.10. The second layer ends up with the smallest thickness, as we might expect given the compaction coefficients. The oscillations that can be seen in the layer thicknesses come from the relative definition of the layers. As the 0 level set of a layer approaches a node it might reach it slightly before or after another layer's 0 level set reaches its next node. The size of the oscillations are a discretization effect dependent on the ratio of length of rock added per time step over the spatial step size. Over the simulation time, the trend clearly shows a reduction in thickness. This decrease looks plausible when considering the compaction coefficients and permeabilities of the different layers.

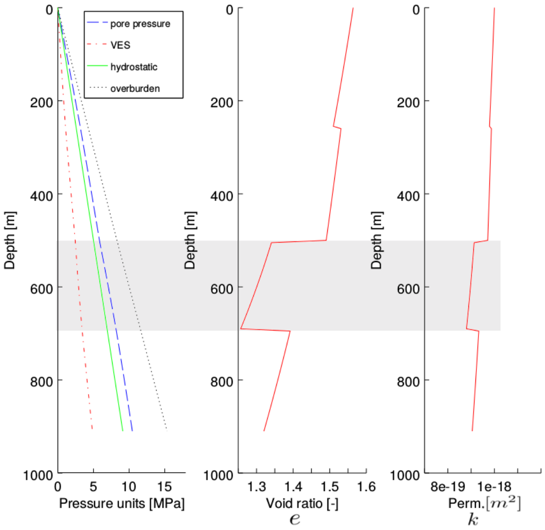

Figure 2.9: Compaction coefficient differences. From the lowest layer going up, the compaction coefficient is $5 \cdot 10^{-8} Pa^{-1}, 9 \cdot 10^{-8} Pa^{-1}, 3 \cdot 10^{-8} Pa^{-1}$ and $5 \cdot 10^{-8} Pa^{-1}$.

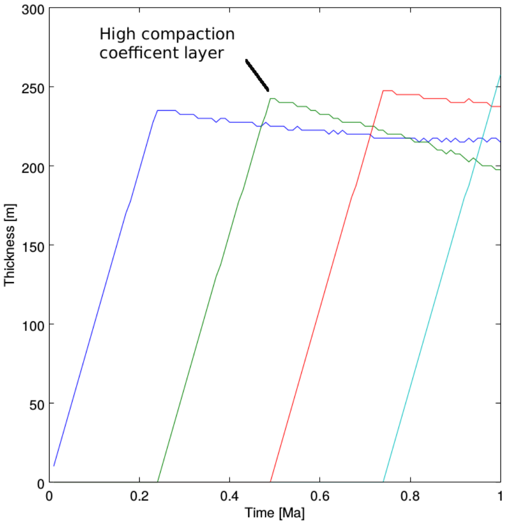

Figure 2.10: Layer Thicknesses for different compaction coefficients. From the first layer on the left, the compaction coefficient is $5 \cdot 10^{-8} Pa^{-1}, 9 \cdot 10^{-8} Pa^{-1}, 3 \cdot 10^{-8} Pa^{-1}$ and $5 \cdot 10^{-8} Pa^{-1}$.

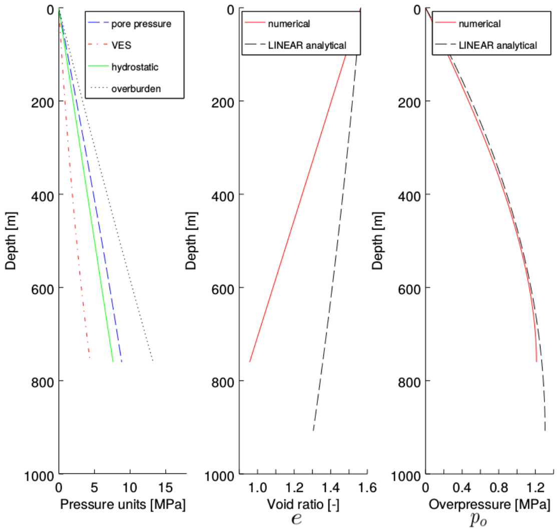

Figure 2.11: Multiple layers, single rock type: Athy's compaction law. Using the multiple layer framework, a uniform compaction coefficient of $5 \cdot 10^{-8} Pa^{-1}$ was used. The initial porosity was 0.61 and the initial permeability was $10^{-18} m^2$ with a discretization of 2.5m. As there is not an analytic expression in this case, the analytic solution for the linear compaction law is plotted for comparison only.

### Nonlinear Compaction Law

More realistic situations can be approached by modelling compaction using Athy's law, Eq. 4.5. As there is no known analytic solution for the consolidation problem using this compaction relation, we present the results of a simulation of multiple layers of one rock type to show the consistency of our modelling approach. In Fig. 2.11, we see the effect of the exponential compaction law. For comparison, the analytical solution, of course with linear compaction, using the same parameters shows the significant increase in compaction. The total length of the column after the simulation is over 100 m shorter than when the linear compaction law is used. For similar pore pressures and effective stresses, the geometrical difference as a result of the compaction law used is significant.

As a concluding test case, we demonstrate results from a simulation using Athy's law with layers defined by distinct depositional periods, initial porosities, permeabilities and compaction coefficients. The simulation parameters are included in Table 2.3. Looking at Fig. 2.12 and Fig. 2.13, one can see the fundamental interplay between compaction coefficient and permeability. Layer 3 compacts significantly, despite having the smallest compaction coefficient, since layer 4 above it has a large

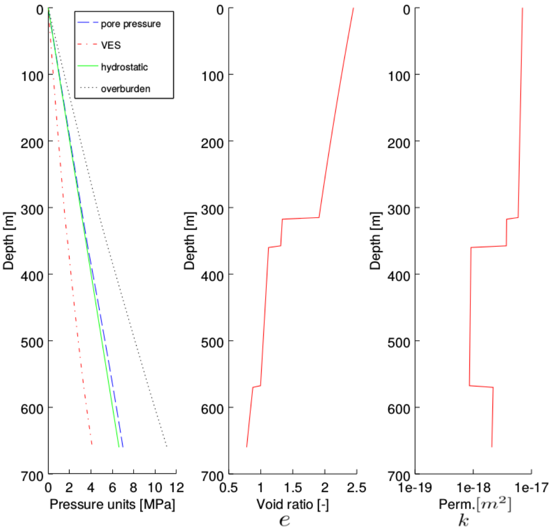

Figure 2.12: Multiple layers, multiple rock types: Athy's compaction law. Description of rock properties is found in Table 2.3.

permeability. Conversely, layer 2 has a low permeability, which stops layer 1 from compacting significantly even with its high compaction coefficient.

| Rock layer<br>(bottom up) | $\phi_0$<br>[-] | $k_0$<br>$[10^{-18}m^2]$ | $\beta$<br>$[10^{-8}Pa^{-1}]$ | Depositional<br>period [Ma] |
|---|---|---|---|---|
| 1 | 0.61 | 3 | 8 | 0.2 |
| 2 | 0.57 | 1 | 4 | 0.3 |
| 3 | 0.60 | 4 | 3 | 0.13 |
| 4 | 0.71 | 7 | 5 | 0.37 |

Table 2.3: Properties of rock layers in Fig. 2.12 and Fig. 2.13. $\phi_0$ is initial porosity, $k_0$ is initial permeability, $\beta$ is compaction coefficient.

In closing, the usefulness of the level set method in representing a sedimentary basin will be considered. By design, the level sets introduce an additional, though simple, advection equation for each interface to be tracked. Of course, this adds to the computational cost. However, there are two primary features: 1) simplicity of the grid and 2) flexibility of the speed function. Firstly, in adopting the Eulerian framework, the grid remains static. There is no need to regrid during the computation as the sedimentary layers evolve. The numerical advantages of a regular grid can be maintained even for larger and longer computations, well-suited for high performance computing. Secondly, the speed

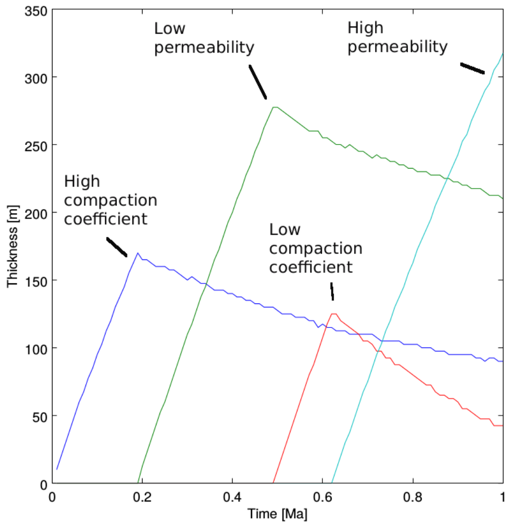

Figure 2.13: Multiple layers, multiple rock types: Athy's compaction law. Thickness of layers in time. Description of rock properties is found in Table 2.3.

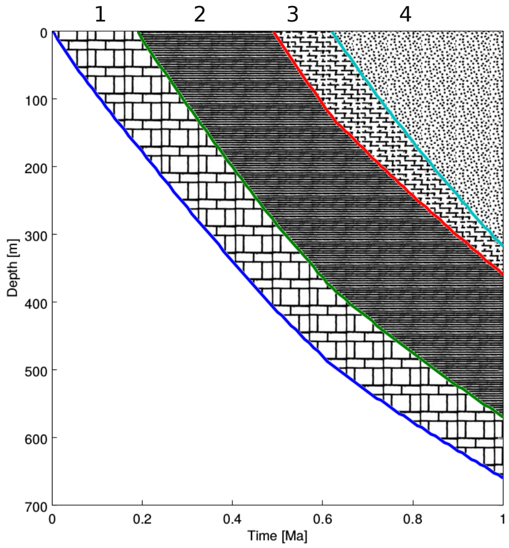

Figure 2.14: Multiple layers, multiple rock types: Athy's compaction law. Burial history. Layer 1 has a high compaction coefficient and layer 2 has a low permeability, while layer 3 has a low compaction coefficient and layer 4 has a high permeability. Description of rock properties is found in Table 2.3.

function which controls the evolution of the fronts can be modified ad hoc. It is possible to consider the physical processes one wishes. The form of the coupling of the physics to the geometry remains the same through the level set evolution. This assumes that the physics solved on the regular grid can be sufficiently well-represented on the static grid for all subsequent material arrangements.

## 2.5 Conclusion

In this paper, a model of single phase fluid flow under isothermal conditions is coupled to the movement of porous rock undergoing compaction due to sedimentation over long time scales. The novel ingredient is the use of the level set method to track the interfaces between rock types while keeping the numerical discretization grid unchanged. We proposed a speed function, dependent on sedimentation and compaction. Numerical experiments have been conducted to verify the model against the one available analytical solution, describing a restricted case of a single rock layer with a linear compaction law. In the more realistic cases of greater interest to basin modelling, namely Athy's nonlinear compaction law and multiple rock layers, we have presented results from simulations that are consistent with the plausibility checks discussed.

We have shown the flexibility of the speed function. Once the numerical model is set up, with the level set equations and speed function, we could change the compaction behavior by modifying only the compaction law used (linear in the void ratio or Athy's law) while the rest of the code was unchanged. It is also possible to add other terms into the speed function, such as for erosion or cementation.

Looking forward, in order to directly address large deformations, such as those due to folding or buoyancy effects, a multidimensional model that includes lateral stresses is needed. This would mean relaxing the *vertical* effective stress assumption. How the effective stress tensor influences the interface motion is determined through the speed function. It remains open how to construct a speed function which would represent the strains due to these stresses. Further elaborations could also include introducing heat flow and the temperature dependence of viscosity and density.

One area where the level set coupling model could be advantageous is overthrusting. Overthrusting and lateral stresses lead to a modification of the vertical effective stress into a mean effective stress. Thus allowing lateral movement, the multiple layer occurences that result are treated in practice with a domain decomposition into blocks. In principle, the topological flexibility of the implicit surface method would not require this block concept for the thrust belts.

CHAPTER 3

# Introducing the Language of Finite Elements and High Performance Computing

In this short chapter, we introduce the language of the finite element library deal.II [42]. This chapter is meant to introduce the notation relevant for our purposes and pave the conceptual path into the following chapter. It can be skipped by the reader familiar with deal.II. It is not meant to represent original contribution. The material that follows is mostly based on dealii.org and draws from [43].

## 3.1 Finite Element Discretizations

In this chapter, we briefly introduce the outlines of a finite element method (FEM) discretization. Rather than discretize differential operators directly, as in finite differences used in the last chapter, the finite element method begins with considering a variational form (also called weak form), and subsequent discretization restricts possible solution values to a representation in a discrete functional space.

### 3.1.1 Laplace's Equation

To demonstrate the solution of an elliptic problem with the FEM, we consider the Laplace equation on a square domain, $\Omega$, with 0 Dirichlet boundary conditions. We include a spatially varying coefficient, $a(\vec{x})$, as this is relevant to our purposes, which rely heavily on varying material properties, expressed in analogous coefficients. In its strong form, this equation is

$$-\nabla \cdot (a(\vec{x})\nabla u) = \rho. \tag{3.1}$$

For simplicity, one might set the source $\rho = 0$. The strong form of the equation is multiplied by a test function, a member $\varphi$ from a suitable functional space, and integrated over the whole domain, $\Omega$, giving the weak or variational form. In this case, the suitable functional space is the so-called Hilbert Sobolev space $H_0^1$, the set of functions that are square integrable, whose derivative is square integrable and which take the value 0 on the boundary. In this work, we use only the scalar Lagrange finite elements, $\mathbb{Q}_p$, the set of continuous, piecewise polynomials of degree $p$.

We seek the discrete solutions in this basis of scalar Lagrange finite elements

$$\int_\Omega -\varphi\nabla \cdot (a\,(\vec{x})\,\nabla u) = 0. \tag{3.2}$$

To reduce the continuity requirements on $u$ integration by parts is used, giving

$$\int_\Omega -\varphi\nabla \cdot (a(\vec{x})\nabla u) = \int_\Omega \nabla\varphi \cdot (a(\vec{x})\nabla u) - \int_{\partial\Omega_D} \varphi\hat{n} \cdot a(\vec{x})\nabla u = 0. \tag{3.3}$$

On the boundary, with $\hat{n}$ the outward pointing normal, $u = 0$, and this is enforced by a condition on the set of possible test functions, i.e., that all $\varphi = 0$ on the boundary. This is an example of an essential boundary condition, one that constrains the possible space of test functions. This leaves only the volume integral,

$$\int_\Omega \nabla\varphi \cdot (a(\vec{x})\nabla u) = 0. \tag{3.4}$$

Now, with the equation in the weak form, the spatial discretization is imposed. To form an approximation to the solution variable, we want to express $u$ as a sum of contributions from the discrete basis functions, so-called shape functions, with the expansion coefficients, $U_j$ as unknowns,

$$u = \sum_j U_j \varphi_j. \tag{3.5}$$

The shape functions, $\varphi_j$, are taken to be elements of the same space as the test functions. This leads to the weak form of the discrete problem, where we require for all $i$:

$$\int_\Omega \nabla\varphi_i \cdot \left(a\left(\vec{x}\right)\nabla\left(U_j\varphi_j\right)\right) = 0. \tag{3.6}$$

Rearranging gives

$$\int_\Omega a(\vec{x})\nabla\varphi_i \cdot \nabla\varphi_j U_j = 0. \tag{3.7}$$

This can be expressed in matrix form with $A_{ij} = \int_\Omega a(\vec{x})\nabla\varphi_i\nabla\varphi_j$, leading to the linear system,

$$A_{ij}U_j = 0. \tag{3.8}$$

$A$ is the discretization of the Laplacian operator, sometimes called the stiffness matrix. In this case, $A$ includes the spatially varying coefficient $a$. In order to solve this linear system numerically, two further steps are required. First, the domain $\Omega$ is broken up into a set of cells, $K$, such that

$$A = \int_\Omega a\nabla\varphi_i\nabla\varphi_j = \Sigma_k \int_k a\nabla\varphi_i\nabla\varphi_j. \tag{3.9}$$

Subsequently, the integral over the $k$th cell is approximated by numerical quadrature, and is evaluated at quadrature points, $x_q$, with weights, $w_q$

$$\int_k a\nabla\varphi_i\nabla\varphi_j \approx \Sigma_q a(x_q)\nabla\varphi_i(x_q)\nabla\varphi_j(x_q)w_q. \tag{3.10}$$

In practice, these quantities are computed on a reference cell, which is mapped to the real cell. This is also done for the shape functions. We use quadrilaterals so this is not a focus in this work, but just imagine a circular or cylindrical domain and these considerations can become crucial. Throughout this work, we use Gaussian quadrature and a multi-linear mapping, i.e., linear in each dimension.

Now that a linear system has been created with the components of the stiffness matrix computed, the system requires a solution. The solution vector is made of the degrees of freedom (DOFs) of the problem, which are to be determined. This could be accomplished with a direct solver, e.g. Gaussian elimination, or with an iterative solver. Linear solvers are discussed later.

The salient point is that the FEM takes a PDE and converts it into a linear system that is then suitable for numerical solution. With the maturation of computing in the 20th and 21st centuries, FEM has become a very powerful technique that can be applied to a wide array of PDEs.

### 3.1.2 Time-dependent Scalar Fields

In this work, we consider a time-dependent version of the above equation, a so-called diffusion equation, as the basis for our overpressure and temperature equations:

$$\frac{\partial u}{\partial t} - \nabla \cdot (a(\vec{x})\nabla u) = \rho. \tag{3.11}$$

There are two standard ways to treat the time dependence. The first is to spatially discretize the Laplacian, leaving only the time as a continuous variable. Then ordinary diffential equation techniques can be used to solve the system in a march. This is called the method of lines.

Alternatively, the time variable can discretized by finite differences, like implicit Euler, leaving a PDE to be solved at each time step. If considering dynamically adapted refinement, allowing for an independent spatial discretization at each time step is crucial, since the spatial discretization might change in (a priori) unforeseen ways, rendering it unsuited for one spatial discretization at the beginning of a time-march. If the mesh is unchanged throughout a simulation, the two methods should lead to similar results.

So, for the set of equations that we consider in this work, per time step, we solve each one sequentially, as opposed to a monolithic solution of a system of equations.

## 3.2 Parallel data structures

In parallel distributed computations, no processor has access to all the memory locations in a computation. Processors have access to their local data and if they require data outside of what they already have, they must communicate with another processor (and know which one!). Therefore, all processors must communicate values in response to other processors' requests. This is done through the message passing interface (MPI). This is the foundational protocol to have different processors communicate about data requests in a structured way. In deal.II, each processor is assigned an index set of degrees of freedom that it owns. Additionally, the processor will have the indices of degrees of freedom from neighboring processes that form the ghost layer, or halo, of possibly relevant values. With this labeling, each processor will perform computations on parts of the matrices and vectors that it owns. When necessary, a process can communicate with another to share updated values of degrees of freedom. In this framework, no processor has immediate access to all DOFs in the domain. This has important consequences for the algorithms discussed in the section on linear solvers, though this is not exposed to the user in deal.II.

In order to evaluate the performance of a parallel code, scaling studies are conducted. The study is meant to quantify how well the code takes advantage of the additional resources. In a so-called strong scaling study, a problem of a given size (namely, degrees of freedom that determine matrix size) is run with different numbers of processors. If the number of processors is doubled, the execution time should be reduced by half, in an ideal case. Often this is not the case, since not all parts of a code can operate in parallel. There are certain parts that are inherently sequential and cannot take advantage of the increased parallelism. Also, passing messages around requires time to communicate, and if the problem size becomes too small per processor, then this communication time can dominate. Another metric to evaluate parallel performance is weak scaling, where the problem size is kept constant per processor, as more and more processors are used.

Figure 3.1: This photograph shows sedimentary rock, now exposed due to uplift, outside Fort Collins, Colorado, USA. Photo taken by author.

## 3.3 Linear Solvers

### 3.3.1 Iterative Methods

We use the preconditioned conjugate gradient method in this work, as well as an extension of it in the form of the Generalized Minimal Residual (GMRes). These are iterated solvers. Rather than constructing a decomposition of the system as in a direct solver, these iterative methods are in the class of Krylov subspace methods. This means that a repeated application of the matrix is used to construct a subspace in which the solution is to be found. For a matrix of rank $n$, the exact solution is obtained in at most $n$ iterations. In the numerical context of floating point arithmetic, with large sparse matrices, this approach is often stopped after some dozens of iterations, assuming the residual decreases sufficiently. The speed of convergence is heavily dependent on the quality of the preconditioner.

Furthermore, there are restrictions for the use of these methods. For the conjugate gradient method, the matrix must be symmetric and positive definite. This means that the eigenvalues are real and positive. This is a prerequisite for the proper functioning of the conjugate gradient method. In our work, the overpressure and temperature fields use this solution technique. Furthermore, the system constructed in the level set solver also uses this. For systems that are not symmetric, GMRes can be applied.

### 3.3.2 Preconditioners

A preconditioner is a linear operator that is applied to a linear system, in order to render it easier to solve, usually by reducing the span between the largest and smallest eigenvalue of the original linear system's matrix. Constructing preconditioners that operate efficiently in parallel is a challenging issue and is the focus of ongoing research. In this work we use Hypre, an algebraic multigrid (AMG) preconditioner through PETSc. Also, a block Jacobi preconditioner is used for the overburden and speed function solutions, in the streamline upwind Petrov-Galerkin (SUPG) systems.

CHAPTER 4

# High Performance Computing in Basin Modeling

## 4.1 Introduction

Basin modeling approaches the evolution of sedimentary basins over geological timespans by considering, in some form, the varied physical processes including mechanical deformation, fluid flow, heat transport and source rock maturation kinetics [2]. In the following, we restrict ourselves to a single-phase fluid flow problem, focusing on mechanical deformation and the geometrical response that different stratigraphic layers undergo in the absence of chemical transport and reactions.

In some basins, there is evidence of significant folding, compacting, thrusting and other manifestations of large deformations, see e.g. [1]. Modeling such types of deformation requires appropriate numerical techniques, e.g. [44, 45]. The issue is that if the discrete numerical grid undergoes deformation, this can degrade the properties of the numerical solution. One successful technique, so-called Arbitrary-Lagrangian-Eulerian (ALE), is to use a mapping between the displacements of the material being modeled and the positions of grid nodes, such that the magnitudes of the node displacements can be kept small, while the solution accurately tracks material movement. At the continental scale, the ALE technique has been used extensively to model tectonics and crustal deformation, e.g., [9, 10, 46].

In the field of salt diapirism, there has also been work that simulates large deformation by considering the different strata as viscous fluids and solving a Stokes system of equations, [47–49]. [27] use a hybrid ALE method with level sets to keep the fluid layers distinct and [50] considers level sets generally in geodynamics.

In the following, we focus on the sedimentary basin as described by a poromechanical model using the effective stress concept, proposed by Terzaghi [28] and further elaborated by Biot [51]. We describe a model of an idealized compaction process during loading of a sedimentary basin. We focus on the use of the level set method, see e.g., [34, 52], to track interfaces between different facies during the evolution of the basin. Building on [3], the purpose of this work is to extend the poromechanical model into 2-D, using finite elements and distributed parallelism. The coupling between the pore pressure and the speed of the moving rock interfaces due to deposition and compaction remains based on the effective stress in only the vertical direction. We also introduce a temperature field into the simulation domain.

In the next section, some physical assumptions are laid out on the way to the mathematical model. Subsequently, the discretization and solution procedure for the system of coupled equations is described. Finally, the results of different synthetic test cases are presented and the parallel scalability is inspected.

## 4.2 Physical Motivation

Let us begin by outlining the equations we intend to solve to model the evolution of basins over time.

### 4.2.1 Fluid flow in and compaction of the porous medium

A central tool to modeling compaction coupled to fluid flow in porous material is the concept of effective stress, introduced by Karl von Terzaghi in 1925 [28]. The effective stress, $\sigma'$, is given by the total stress, $\sigma$, minus the pore pressure, $p$:

$$\sigma' = \sigma - p. \tag{4.1}$$

We assume a single phase saturated porous medium. The porosity, $\phi$, of a representative volume element, as defined by the structure of the solid matrix therein, has an effect on the ability of the saturating pore fluid to flow through that region. Through the assumption of the effective stress, the pore pressure also has an influence on the stress that the solid matrix undergoes. In basin modeling, it is common to consider only the stress in the vertical direction, $\sigma_z$, which can be assumed to be equivalent to the overburden, the weight of material above a point. Thus, the effective stress in the vertical direction is $\sigma_z' = \sigma_z - p$, called the vertical effective stress (VES).

Within the porous medium, we assume that the fluid that fills the pores behaves following Darcy's law. That is, the discharge velocity of the fluid, $\mathbf{v}$, is given by

$$\mathbf{v} = -\frac{\mathbf{k}}{\mu}\nabla\left(p - \rho_f g z\right), \tag{4.2}$$

where $\mathbf{k}$ is the permeability, $\mu$ is the viscosity of the fluid, $p$ is the pore pressure, $\rho_f$ is the density of fluid, $g$ is the acceleration due to gravity, and $z$ is a depth below the surface datum.

Assuming no change in fluid density, i.e., an incompressible fluid, the conservation of fluid mass can be stated as

$$\nabla \cdot \mathbf{v} = -\frac{1}{1-\phi}\frac{\partial \phi}{\partial t}. \tag{4.3}$$

From the conservation of fluid mass and Darcy's law, we have for the pore pressure at a depth $z$,

$$\nabla \cdot \mathbf{v} = -\nabla \cdot \left(\frac{\mathbf{k}}{\mu}\nabla\left(p - \rho_f g z\right)\right) = -\frac{1}{1-\phi}\frac{\partial \phi}{\partial t} = \frac{C(\sigma_z')}{1-\phi}\frac{\partial(\sigma_z - p)}{\partial t}, \tag{4.4}$$

where $C$ is the compressibility, defined as $C(\sigma') := -\frac{\partial \phi}{\partial \sigma_z'}$. The compressibility depends on the compaction rule used to relate the VES to the porosity. A compaction rule is an empirical relation that models the compaction of the porous medium (and the consequent reduction of the porosity) with depth as a consequence of the effective stress of the overlying material.

In this work, we will use two different parametrizations for how the porosity is related to the VES. The first compaction rule is the widely used Athy's law:

$$\phi = \phi_0 e^{-\beta\sigma_z'}, \tag{4.5}$$

where $\phi_0$ is an initial porosity value and $\beta$ is a compaction parameter. With Athy's law, the compressibility is given by

$$C(\sigma') := -\frac{\partial\phi}{\partial\sigma_z'} = \beta\phi. \tag{4.6}$$

The second compaction rule is one that is linear in the void ratio, $e$,

$$e = e_0 - \beta\sigma_z', \tag{4.7}$$

where $e_0$ is an initial void ratio value. Recall that the relationship between porosity and void ratio can be expressed as $\phi = \frac{e}{1+e}$. The compressibility in this case is then,

$$C(\sigma') := -\frac{\partial\phi}{\partial\sigma_z'} = \beta(1-\phi)^2. \tag{4.8}$$

This compaction rule is introduced for use in the verification example in the results section.

The pressure equation can alternatively be expressed through the overpressure at a depth $z$, $p_o = p - \rho_f g z$, leading to

$$\frac{C(\sigma_z')}{1-\phi}\frac{\partial p_o}{\partial t} - \nabla\cdot\left(\frac{\mathbf{k}}{\mu}\nabla p_o\right) = \frac{C(\sigma_z')}{1-\phi}\frac{\partial(\sigma_z - \rho_f g z)}{\partial t}. \tag{4.9}$$

Regions of overpressure buildup in a basin are of geological interest. In our computations we use the overpressure as the primary variable because it focuses on the pressure in excess of the hydrostatic pressure, which can be orders of magnitude larger than the overpressure.

To solve the partial differential equation, boundary and initial conditions must be specified. In Fig. 4.1, the definition of the domain Ω and the boundary surfaces are shown. The rectangular domain

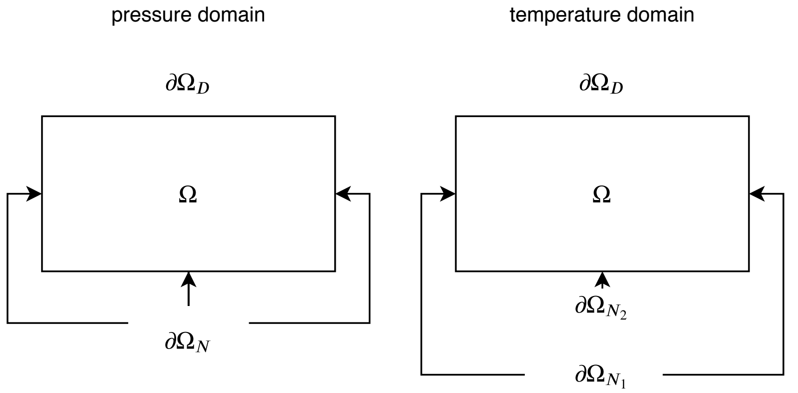

Figure 4.1: A schematic of the 2-D domain $\Omega$ and its boundary partitions for both the ovepressure equation and the temperature equation is shown.

is split into $\partial\Omega_D$ and $\partial\Omega_N$. $\partial\Omega_D$ is defined as the top surface of the domain, with $\partial\Omega_N$ comprised of the remaining three sides. The boundary conditions are:

$$p_o = 0 \tag{4.10}$$

on $\partial\Omega_D$ for all $t > 0$, and

$$\frac{\mathbf{k}}{\mu}(\nabla p_o) \cdot \hat{\mathbf{n}} = 0 \tag{4.11}$$

on $\partial\Omega_N$ for all $t > 0$, where $\hat{\mathbf{n}}$ is the normal vector pointing out.

For the initial condition at time $t = 0$, it is a hydrostatic condition,

$$p_o = 0 \tag{4.12}$$

in domain $\Omega$.

The source, on the right hand side of equation (4.9), is essentially the rate of change of the overburden, determined by the sedimentation rate, $\omega$ $[LT^{-1}]$. The total overburden at a depth $z$ is given by the weight of the material above it,

$$\sigma_z = \int_0^z \rho_b(s)\, g\, ds, \tag{4.13}$$

where $\rho_s$ is the solid rock density, the bulk density is $\rho_b = (1-\phi)\rho_s + \phi\rho_f$ and $\phi$ is a function of depth.

Beginning from equation (4.9),

$$\frac{C}{1-\phi}\frac{\partial p_o}{\partial t} - \nabla\cdot\left(\frac{\mathbf{k}}{\mu}\nabla p_o\right) = \frac{C}{1-\phi}\frac{\partial(\sigma_z - \rho_f g z)}{\partial t} \tag{4.14}$$

The source term on the right hand side contains shows the hydrostatic component that can be expressed as,

$$\frac{\partial(\sigma_z - \rho_f g z)}{\partial t} = \frac{\partial}{\partial t}(g\int_0^z(\rho_s - \rho_f)(1-\phi(z'))dz') \tag{4.15}$$

Remembering the assumption of a known sedimentation rate of bulk material with a known depositional porosity, we can express this as, using the definition of bulk density,

$$\frac{\partial(\sigma_z - \rho_f g z)}{\partial t} = (1-\phi)(\rho_s - \rho_f)g\omega[ML^{-1}T^{-3}] \tag{4.16}$$

$$\frac{\partial\sigma_z}{\partial t} = \rho_b|_{\phi_0} g\omega[ML^{-1}T^{-3}] \tag{4.17}$$

We call the right hand side $Q$ and call $\frac{k}{\nu}$ the diffusion coefficient, $a$.

$$\frac{C}{1-\phi}\frac{\partial p_o}{\partial t} - \nabla\cdot(a(x)\nabla p_o) = Q \tag{4.18}$$

### 4.2.2 Temperature of the medium

The thermal history of a basin is an important aspect of its evolution. The model from McKenzie [53] is an instructive example of how heat flow considerations can be connected to subsidence and stretching of the lithosphere. For our purposes, we will use a basic model for the temperature calculation in the basin. Specifically, using the conservation of energy and Fourier's law, we have that the temperature $T$ satisfies the equation

$$\rho_b c\frac{\partial T}{\partial t} - \nabla\cdot(\lambda\nabla T) = 0, \tag{4.19}$$

where $\rho_b$ is the bulk density, $c$ is the bulk heat capacity, and $\lambda$ is the thermal conductivity.

As above, the rectangular domain is split into $\partial\Omega_D$ and $\partial\Omega_N$. $\partial\Omega_D$ is defined as the top surface of the domain, with $\partial\Omega_N$ being further subdivided into $\partial\Omega_{N_1}$, the two sides, and $\partial\Omega_{N_2}$ comprised of just the bottom. The boundary conditions are:

$$T = T_{surface}, \tag{4.20}$$

where $T_{surface}$ is a fixed value on $\partial\Omega_D$ for all $t > 0$, and

$$\lambda \nabla T \cdot \hat{\mathbf{n}} = 0 \tag{4.21}$$

on $\partial\Omega_{N_1}$ for all $t > 0$, where $\hat{\mathbf{n}}$ is the normal vector pointing out. On $\partial\Omega_{N_2}$, the basal heat flux is enforced,

$$\lambda \nabla T \cdot \hat{\mathbf{n}} = q, \tag{4.22}$$

where $q$ is a fixed heat flux. For simplicity, the initial condition at time $t = 0$ is taken to be

$$T = 0 \tag{4.23}$$

in the domain $\Omega$.

We note a few possibilities for elaboration. We assume that the sides of the basin are insulated (no heat flux). This assumption can be improved upon by using a temperature profile based e.g., on the geothermal gradient, at the lateral boundaries. We model the temperature as a diffusive isotropic problem. A more complete thermal model would include the advection of heat by the fluid parcels of water, thereby introducing an advection term into the temperature equation. As this is only important in regimes of high Darcy flux [54], we omit the term and treat the temperature in the diffusive regime only. Also, source terms due to radiogenic heat production could be included. We leave this to further refinements.

### 4.2.3 Rock strata interface motion

The main contribution of this paper is to introduce an implicit representation of the interfaces between rock layers. The purpose is to eliminate re-meshing to restore conformity of the grid to the material layers. Rather, in the approach proposed here, we shift the layer interfaces by advecting with a prescribed speed function the auxiliary potential functions, whose priviledged level set value(normally taken to be 0) serves to define the interface. This is an implicit representation of interfaces with level set functions, from which we can reconstruct the identity of a point in the domain, i.e., what rock layer is it in and what are its material properties.

The level set method, going back to [11, 12], has found a wide array of applications in numerical models that rely on representing an evolving interface, e.g. [38, 55, 56].

The main idea is to embed the interface of interest into a potential function, whose evolution is determined by an advection equation,

$$\frac{\partial u}{\partial t} + \vec{w} \cdot \nabla u = 0, \tag{4.24}$$

where $\vec{w}$ is the velocity field for the interface motion and $u$ is the advected quantity (level set function). The interface is then defined to be all the points where $u = 0$.

Typically, the level set function is initialized to be a signed distance function, such that the distance from the interface to a point determines the value of the potential function at that point. If this property is maintained, then the following simplification is obtained $|\nabla u| = 1$. Retaining this property of the level set function is the purpose of the technique of reinitialization. Towards this goal, the procedure of reinitialization has been considered [37, 57]. The goal is to ensure the regularity of $|\nabla u|$ and so keeping the normal vector well-defined, given by $\hat{n} = \frac{\nabla u}{|\nabla u|}$

Defining the speed function, $F$, which represents the sedimentation rate, and possible compaction effects, to be the the magnitude of the vector field normal to the front surface, $F = \vec{w} \cdot \hat{n}$, and with the definition of the unit normal vector above, (4.24) can be rewritten as

$$\frac{\partial u}{\partial t} + F|\nabla u| = 0 \tag{4.25}$$

If we consider only the speed function $F$, as is done in the current work, we can clearly see that the interface only moves in the normal direction. This prevents a straightforward inclusion of tangential forces at the interface, e.g., shear stress on a surface or surface tension in the case of immiscible two-phase flow.

Level set methods have the benefit that they can implicitly track an interface as it undergoes possibly complex changes, including topology changes. The grid upon which this is done can be fixed. The nodes do not need to be moved, and this will not lead to degenerate geometries for which a regridding procedure is required. This fixed grid has desirable properties when considering the parallelization and distribution of the problem over multiple machines. Since each interface requires its own level set function, the cost of solving the advection equations can become significant. Local grid refinement can help toward reducing that cost. The level set method has the additional benefit that it is well suited for parallel implementation on octrees, which is crucial for local refinement around the interfaces [58, 59].

For applications with multiple intersecting interfaces, the method is less robust. Some work on multiphase flow problems has been done [60, 61]. However, in the current model, we do not have intersecting level sets. In other words, the model is a set of layers that will not pierce each other, much like a layer cake that can deform, but keeps the original connectivity intact.

### Determination of interface speed

We consider the speed normal to an interface, $F$ at a point in the column, $z'$ to be given by the following

$$F = F\left(p_o\right) = \omega + \int_0^{z'} \frac{1}{1-\phi} \frac{\partial \phi\left(p_o\right)}{\partial t} dz \tag{4.26}$$

which represents the layer moving at the rate of deposition, $\omega$, and incorporating the cumulative compaction rate of the column above a point in the profile. We note that the coupling between the geometry and the physical states is located here. $F$ determines how the layers move, dependent on the overpressure, and, in turn, the overpressure will depend on layer positions.

## 4.3 Solution Procedure

### 4.3.1 Summary of physical model

Shown in Fig. 4.2, the procedure that is implemented to carry out the basin simulation is as follows: (1) Material is deposited in the domain, which is done by (a) advancing the level set corresponding to the material layer and (b) updating the identity of all cells in the domain to reflect the new interface position. (2) The overburden equation is then solved to update the stress at each point. (3) The overpressure equation is then solved. (4) Steps 2 and 3 are repeated until the overpressure does not change more than within a certain tolerance. (5) The temperature field is solved. (6) A new speed function is then calculated, in order to repeat the steps in the next time step.

This operator splitting approach separates the sedimentation and the compaction steps into sequential events. Next, we examine the discretization and solution procedures of the equations that make up the basin model.

### 4.3.2 Discretization with finite elements

We first consider the discretization of the pressure equation, formulated with the overpressure variable.

#### Overpressure weak form

First, we discretize the time derivatives with an implicit Euler (or for higher order we can use Crank-Nicolson) finite difference approximation.

$$\frac{C}{1-\phi}\frac{p_o^{n+1}-p_o^n}{\Delta t} - \nabla \cdot (a(x)\nabla p_o^{n+1}) = Q^{n+1} \tag{4.27}$$

To convert the strong form of the PDE to the variational formulation, we multiply by an arbitrary member $\varphi$ from an appropriate set of test functions, and integrate over the domain. We seek the discrete solutions in a basis of scalar Lagrange finite elements, $\mathbb{Q}_p$, the set of continuous, piecewise

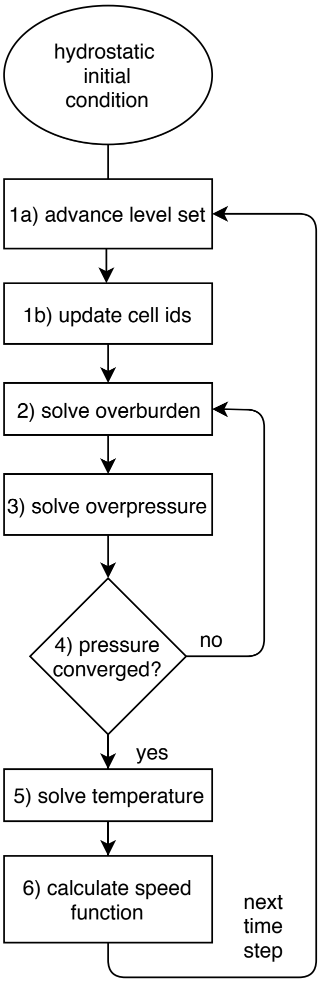

Figure 4.2: Schematic of the solution algorithm for the simulation

polynomials of degree $p$.

$$\int_\Omega \frac{C}{1-\phi}\varphi(\frac{p_o^{n+1}-p_o^n}{\Delta t}) - \int_\Omega \varphi(\nabla \cdot (a(x)\nabla p_o^{n+1})) = \int_\Omega \varphi Q^{n+1} \tag{4.28}$$

To reduce the continuity requirements on $p_o$, the second term on the left hand side is integrated by parts, giving

$$\int_\Omega \varphi(\frac{p_o^{n+1}-p_o^n}{\Delta t}) + \int_\Omega \nabla\varphi \cdot (a(x)\nabla p_o^{n+1}) - \int_{\partial\Omega} \varphi(a(x)\nabla p_o^{n+1}) \cdot \hat{n} = \int_\Omega \varphi Q^{n+1} \tag{4.29}$$

Examining the boundary term:

$$\int_{\partial\Omega} \varphi(a(x)\nabla p_o^{n+1}) \cdot \hat{n} = \int_{\partial\Omega_D} \varphi(a(x)\nabla p_o^{n+1}) \cdot \hat{n} + \int_{\partial\Omega_N} \varphi(a(x)\nabla p_o^{n+1}) \cdot \hat{n} \tag{4.30}$$

On the Dirichlet boundary, the test function comes from a discrete functional space, whose members are defined to be zero on that part of the boundary. So, that integral is zero. Applying the zero Neumann boundary condition to the surface integral eliminates it, and we have the weak formulation of the equation. To form an approximation to the solution variable, we want to express the $p_o$ as a sum of contributions from the discrete basis functions with the expansion coefficients to be determined,

$$p_o = \sum_j U_j \varphi_j \tag{4.31}$$

The shape functions, $\varphi_j$, are elements of the same space as the test functions. This leads to the weak form of the discrete problem, where we require for all $i$:

$$\int_\Omega \frac{C}{1-\phi}\varphi_i \left( \frac{(U_j\varphi_j)^{n+1} - (U_j\varphi_j)^n}{\Delta t} \right) + \int_\Omega \nabla\varphi_i \cdot (a(x)\nabla(U_j\varphi_j)^{n+1}) = \int_\Omega \varphi_i Q^{n+1} \tag{4.32}$$

Rearranging,

$$\int_\Omega \frac{C}{1-\phi}\varphi_i\varphi_j U_j^{n+1} - \int_\Omega \varphi_i\varphi_j U_j^n + \Delta t \int_\Omega a(x)\nabla\varphi_i \cdot \nabla\varphi_j U_j^{n+1}) = \Delta t \int_\Omega \varphi_i Q^{n+1} \tag{4.33}$$

Expressing the mass matrix with coefficient as $M' = \int_\Omega \frac{C}{1-\phi}\varphi_i\varphi_j$, the stiffness matrix (with the spatially dependent coefficient $a(x)$) as $A' = \int_\Omega a(x)\nabla\varphi_i\nabla\varphi_j$ and the source term vector $F = \int_\Omega \varphi_i Q^{n+1}$ leads to the linear system,

$$M'U^{n+1} - M'U^n + \Delta t A'U^{n+1} = \Delta t F^{n+1} \tag{4.34}$$

and finally grouping the known quantities on the right hand side leads to the linear system of equations to be solved at each time step,

$$(M' + \Delta t A')U^{n+1} = M'U^n + \Delta t F^{n+1} \tag{4.35}$$

Since $(M' + \Delta t A')$ is symmetric and positive definite, the preconditioned conjugate gradient method is used to iteratively solve the system.

As mentioned above, but for conciseness not shown, we can use a Crank-Nicolson time stepping which mixes from the current and last solution vectors. This is formally a second-order method. It is noted that in the results section 4.5, the Crank-Nicolson time-stepping was used.

### Temperature

A very similar procedure as above is followed for the temperature. There is no source term and the coefficients in front of the time derivative are subsumed, alongside the coefficient for the thermal conductivity, into $b$.

$$\frac{\partial T}{\partial t} - \nabla \cdot (b(x)\nabla T) = 0 \tag{4.36}$$

The only difference in the weak form comes from the non-zero Neumann condition at the bottom of the domain. In Eq. (4.29), the boundary term becomes

$$\int_{\partial\Omega} \varphi(b(x)\nabla T^{n+1}) \cdot \hat{n} = \int_{\partial\Omega_D} \varphi(b(x)\nabla T^{n+1}) \cdot \hat{n} + \int_{\partial\Omega_N} \varphi(b(x)\nabla T^{n+1}) \cdot \hat{n} \tag{4.37}$$

Integrating over $\partial\Omega_D$ still gives 0 since $\varphi \in H_0^1$, but while the flux through the sides remains 0, the bottom, $\partial\Omega_{bottom}$ has a fixed flux of $q$. So, with $T = \sum_j U_j \varphi_j$,

$$\int_{\Omega} \varphi_i (\frac{(U_j\varphi_j)^{n+1} - (U_j\varphi_j)^n}{\Delta t}) + \int_{\Omega} \nabla\varphi_i \cdot (b(x)\nabla(U_j\varphi_j)^{n+1}) = \int_{\partial\Omega_{bottom}} \varphi_i q \tag{4.38}$$

The same preconditioned conjugate gradient method is applied to the resulting linear system. As an implicit time discretization was used, the time step should not be bound by a stability condition, though in practice the time step is limited by the stability condition for the explicit level set solver, as described below in section 4.3.2.

### Overburden and speed function

Both equations (4.26) and (4.13) contain a line integral in the vertical direction. This is treated in the finite element context by applying the gradient operator to both sides of the equation and then taking the dot product with the normal vector in the vertical direction.

$$F = F(p_o) = \omega + \int_0^{z'} \frac{1}{1-\phi} \frac{\partial \phi(p_o)}{\partial t} dz \tag{4.39}$$

leading to

$$\hat{e_z} \cdot \nabla F = \frac{1}{1-\phi} \frac{\partial \phi}{\partial t} \tag{4.40}$$

and similarly for the overburden equation, leading to

$$\hat{e_z} \cdot \nabla \sigma_z = \rho_b g \tag{4.41}$$

Since these two advection equations lack a property called coercivity, see e.g.,Typical options are to introduce stabilization, such that the coercivity property is obtained and the standard Galerkin solution technique can be applied. We treat these two advection equations with a method that introduces some diffusion in the direction of the flow field, called the streamline upwind Petrov-Galerkin method (SUPG), originally proposed in [62]. The weighting functions are supplemented by the addition of a “streamline upwind perturbation", acting only in the flow direction. The modified weighting function is applied to all terms, so it is a consistent weighted residual construction. There are a number of stabilization schemes and alternatives to this approach, such as the discontinuous Galerkin method and a method based on the partition of unity, as introduced in [63].

We describe the formation of the weak form for this particular stabilization mechanism, using the generic variables, $u$ and $f$.

The inflow boundary, $\partial\Omega_{in}$ is taken to be the top layer, $z = 0$, with the flow field, $\vec{w}$ given by the deposition rate, $\omega$ in the normal direction. We formulate the following discrete problem at each time step.

$$\vec{e_z} \cdot \nabla u = f \tag{4.42}$$

We want to cast this equation in variational form, thus we multiply by a test function from the left and integrate over the domain. As mentioned above, the space of test functions is chosen such that each member has the form,

$$\varphi + \delta \vec{e_z} \cdot \nabla \varphi \tag{4.43}$$

where $\delta$ is a small number, dependent on the local mesh size. This gives

$$\int_\Omega (\varphi + \delta \vec{e_z} \cdot \nabla \varphi) \vec{e_z} \cdot \nabla u_h = \int_\Omega (\varphi + \delta \vec{e_z} \cdot \nabla \varphi) f \tag{4.44}$$

where $u_h = \sum_j U_j \varphi_j$ is the discrete expansion of the solution variable. The test functions and the basis functions come from different spaces, characteristic of the Petrov-Galerkin formulation. To incorporate the inflow boundary condition, the same quantity is subtracted from both sides, and on the right hand side the known value is inserted, yielding

$$\int_\Omega (\varphi + \delta \vec{e_z} \cdot \nabla \varphi) \vec{e_z} \cdot \nabla u_h - \int_{\partial\Omega_{in}} \vec{e_z} \cdot \hat{n} u_h = \int_\Omega (\varphi + \delta \vec{e_z} \cdot \nabla \varphi) f - \int_{\partial\Omega_{in}} \vec{e_z} \cdot \hat{n} \omega \tag{4.45}$$

This system is not symmetric positive definite, so the standard conjugate gradient method cannot be used. Instead, we use the Generalized Minimal Residual (GMRes) method.

### Level set equations

The level set equation is a hyperbolic PDE, and so the standard finite element Bubnov-Galerkin discretization method will lead to instabilities (non-physical oscillations) when applied. In the code, we use a level set solver that is based on higher-order stabilization, [64]. The particular level set solver used in this work was implemented in [65]. As it is quite sophisticated in using an entropy viscosity stabilization, as well as artificial compression to minimize numerical diffusion, we refer to the relevant literature for a detailed description. As indicated above, despite its simple appearance, the first-order hyperbolic advection equation has required special techniques to solve accurately in the finite element context, and is an area of active research.

We note that the level set solver code uses an explicit time-stepping, either forward Euler or a strong-stability preserving (SSP) higher-order time stepping. In either case, this requires attending to the limitations of a Courant-Friedrichs-Lewy (CFL) condition on the relation between the grid size and the time step. We set the CFL condition to 0.5 and iterate the level set solver two times per time step, corresponding to the physical field updates. The idea to have a time step for the explicit level set solver that is a fraction of the physical time step was inspired by [66]. This helps keep the level set solver stable amid the changes to the speed function, computed based on the physical fields.

### 4.3.3 System summary

We summarize the equations that determine the model here:

$$\begin{aligned}
\frac{\partial u_i}{\partial t} + \vec{w} \cdot \nabla u_i &= 0 \\
-\nabla \cdot \left( \frac{k}{\mu} \nabla (p - \rho_f g z) \right) &= \frac{C(\sigma_z')}{1-\phi} \frac{\partial (\sigma_z - p)}{\partial t} \\
\rho c \frac{\partial T}{\partial t} - \nabla \cdot \lambda \nabla T &= 0 \\
\hat{e}_z \cdot \nabla \sigma_z &= \rho_b g \\
\hat{e}_z \cdot \nabla F &= \frac{1}{1-\phi} \frac{\partial \phi}{\partial t},
\end{aligned}$$

with $\vec{w} = (0, 0, F)$.

## 4.4 Implementation

We use the deal.II library [42]. We use the parallel distributed capabilities [67], based on the p4est library [68], though at this stage we do not use the capabilities of adaptive local refinement provided therein. For the preconditioner to the Conjugate Gradient solver, we use the Algebraic Multigrid package Hypre provided through PETSc. The code is structured in the following way: a top-class provides the framework for reading in parameter data, initializing data structures, and organizing the time stepping loop, where the routines for the different parts of the solution procedures are called. This top-class interfaces with the level set solver, a separate class that is treated as a black box, through providing an initial condition for the potential function once and a velocity field at each time step. The level set solver returns the new potential function after advancing it corresponding to one time step. Each layer has its own instance of the level set class, while there is only one speed function used for all the layers.

Each cell in the domain is assigned a material type. When updating the cell identities (step (1b) in Fig. 4.2), for each quadrature point on a cell, to determine the identity at that point, a sum of all the level set potential values at that quadrature point is computed. This is done for each quadrature point on the cell and the sum of these results corresponds to a material type. For simplicity, we ignore the complexities of a possible intra-cell cut, and so each cell has just one material identity at a time.

### 4.4.1 Numerical experiments

The first reference setup is based on the Gibson consolidation problem, with the specific setup here we call the “Wangen" configuration, as presented in [41] and that has been used in [3]. This is the only relevant analytical solution available, and there are no publicly available benchmarks for verifying basin simulators. The consolidation problem is valuable to evaluate the coupling between the pressure solver and the rock layer movement. It is clear that consolidation will not lead to large deformation and so its interest is perhaps limited. Nevertheless it is useful for evaluating correctness and for profiling.

Subsequently, cases that introduce heterogeneity in various ways are elaborated. These examples depend on material properties that are meant to vary in certain ways. The code is structured with the material properties in a class separate from the rest, such that a more complicated or a more physically consistent material model for the relevant properties can modularly be introduced. In this work, we consider only abstract basins with synthetic material properties. In the following examples, we present (1) variations on the sedimentation function, (2) layers with varying permeabilities, and (3) a case of thermal flow with heterogeneous material properties.

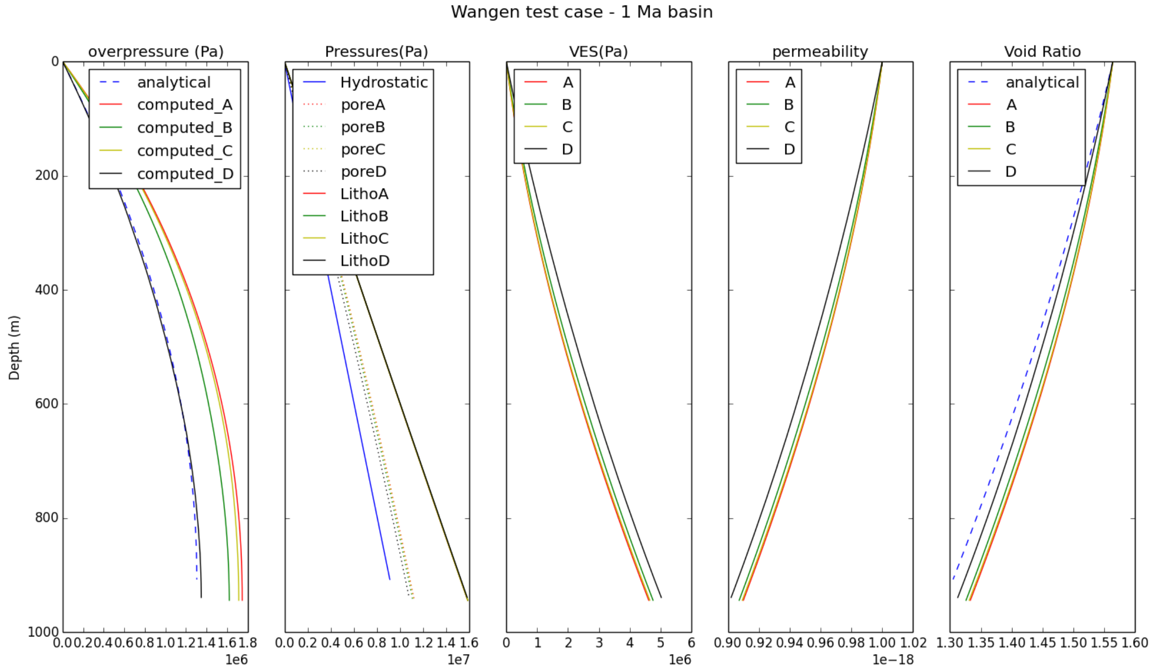

Figure 4.3: This presents four simulations with varying parameters against analytical Wangen solution. Run A: $dz = 5m$ and 10 nonlinear iterations. Run B: $dz = 5m$ and 100 nonlinear iterations. Run C: $dz = 2.5m$ and 10 nonlinear iterations. Run D: $dz = 2.5m$ and 100 nonlinear iterations.

## 4.5 Results and Discussion

### 4.5.1 Simulation cases

#### Multi-1D for verification

We seek to verify the code by comparison to an analytical solution in the case that there is one rock material added at a constant rate. We use a profile through the two dimensional domain for comparison with the one-dimensional analytic solution. This is perhaps best expressed as a “multi one-dimensional" framework, as there are no interactions in the lateral direction.

The simulation results depend on the number of nonlinear iterations and the mesh size. In the following figure, four separate simulations are carried out at the combinations of 2.5m/5m spatial resolution with 10 or 100 nonlinear iterations (over the overburden and overpressure solutions).

The depth of the sediment column is analytically 907 m and for runs A,B,C the depth extends to 950 m, or 4 percent error over a 1 km column (uncompacted depth). Run D is slightly better at 940 m, or  3 percent error.

The plots move toward the analytical overpressure solution with a higher resolution and with more nonlinear iterations. This suggests that the code is well-behaved in regards to this specific case.

It is noted that this case uses a particular compaction law, i.e., linear in the void ratio and the compressibility is given by Eq. (4.8).

### Differential sedimentation

In the section, we consider the effects of differential sedimentation on a hydrostatic basin. The following two simulations were run on the Forschungszentrum Jülich's JSC/Jureca supercomputer [69] over a few hundred MPI processes, each with just over 1 million degrees of freedom (DOFs). The permeabilities are chosen such that no overpressure builds up. In this context, we can manipulate the sedimentation rate over the basin laterally, as in Fig 4.4. Here the sedimentation rate is constant in time, i.e., $\omega(x)$, and scaled with a trigonometric function laterally.

In Figure 4.5, the sedimentation rate changed both in time and space, i.e., $\omega(x,t)$. A base rate of sedimentation is first increased with a Gaussian distribution localized around 500m, subsequently, this disappears and on either side of the central line, different Gaussian distributions slow down the sedimentation rate. The purpose of this conceit is to highlight the ability of the level set solvers to handle lateral deformation, while retaining a vertical effective stress rule. This figure serves as an answer to one of the fundamental questions of this work: namely, can we reproduce geometrical variations laterally under the assumption of vertical effective stress? This simulation contributes to a positive response.

### Heterogeneous permeability

Here we focus on the interfaces between homogeneous layers with permeabilities that differ by a few orders of magnitude, as in the case of a seal. Again in a 2-D basin, we examine a 1-D profile of the overpressure. In Fig. 4.6, we see four overlapping time slices of the basin overpressure around the time of a new layer's introduction. Since, each layer has a permeability that differs from its neighbor, this causes variations in the overpressure, as seen in the graph on the left side.

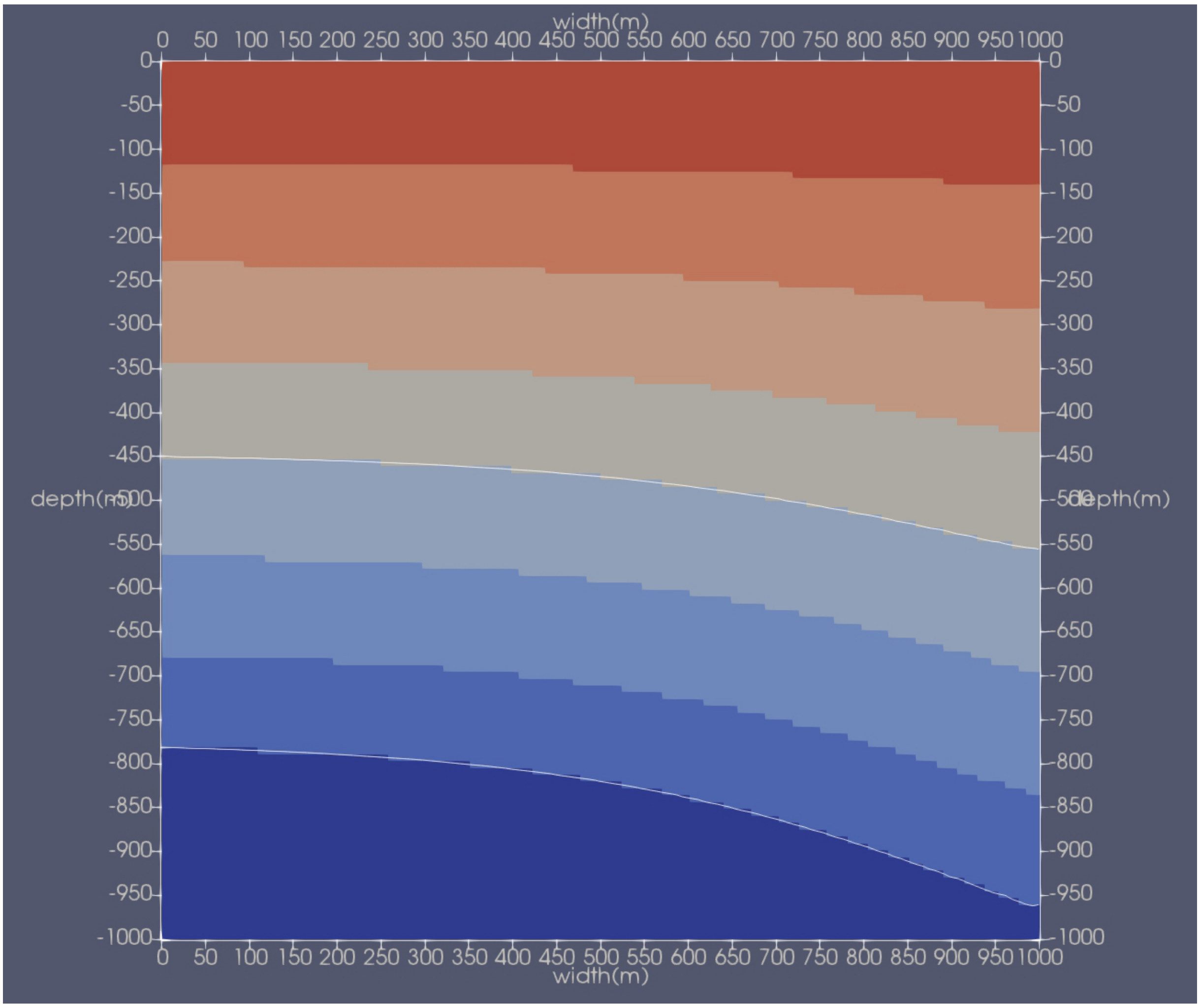

Figure 4.4: Eight layers with variable sedimentation rates left to right are shown. The bottom and fourth above interfaces are traced with a contour at the 0.5 level set. One can see that they are smooth, while the coarse mesh leaves the identification of that cell visible.

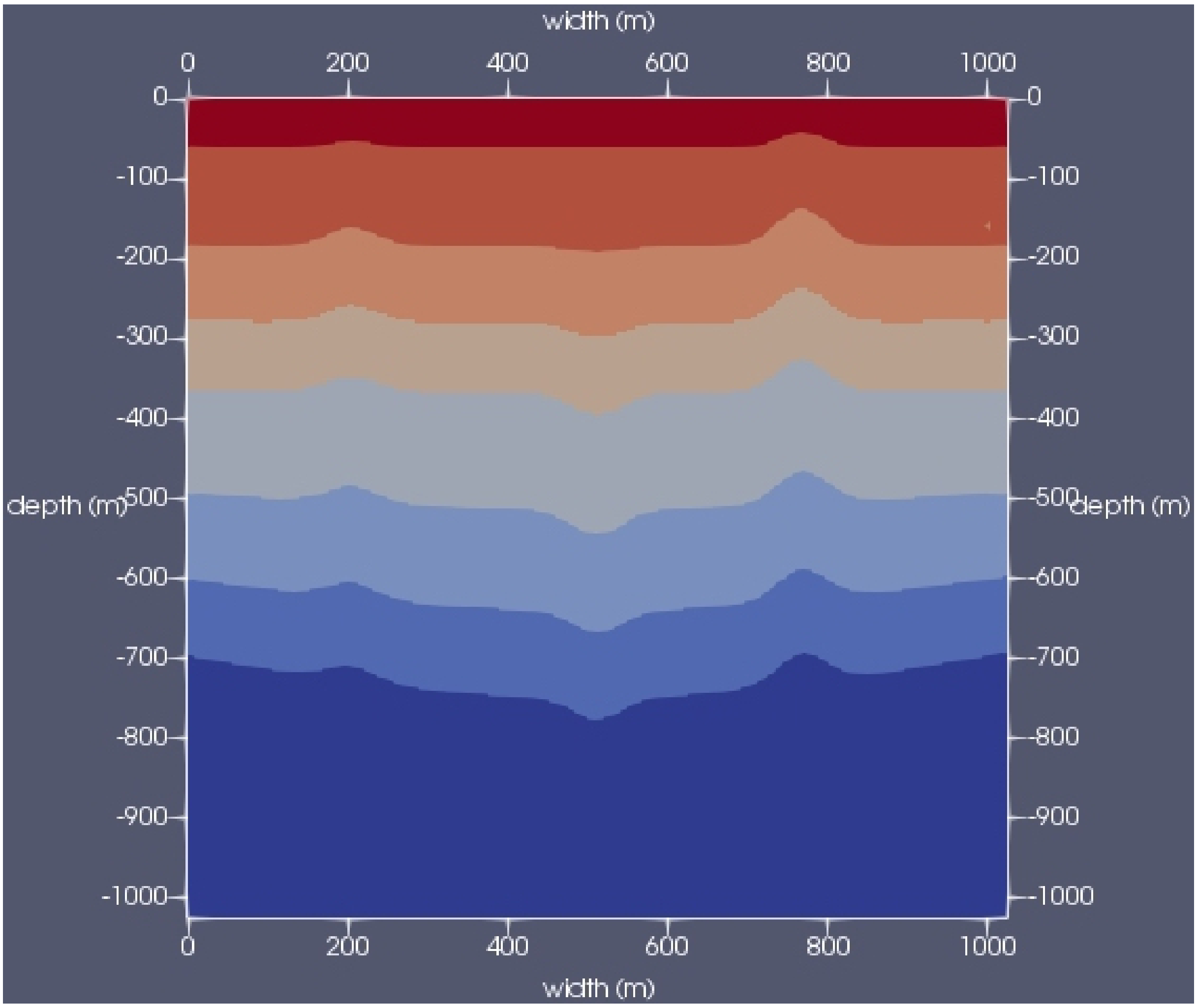

Figure 4.5: Eight layers with variable sedimentation rates in both space and time are shown.

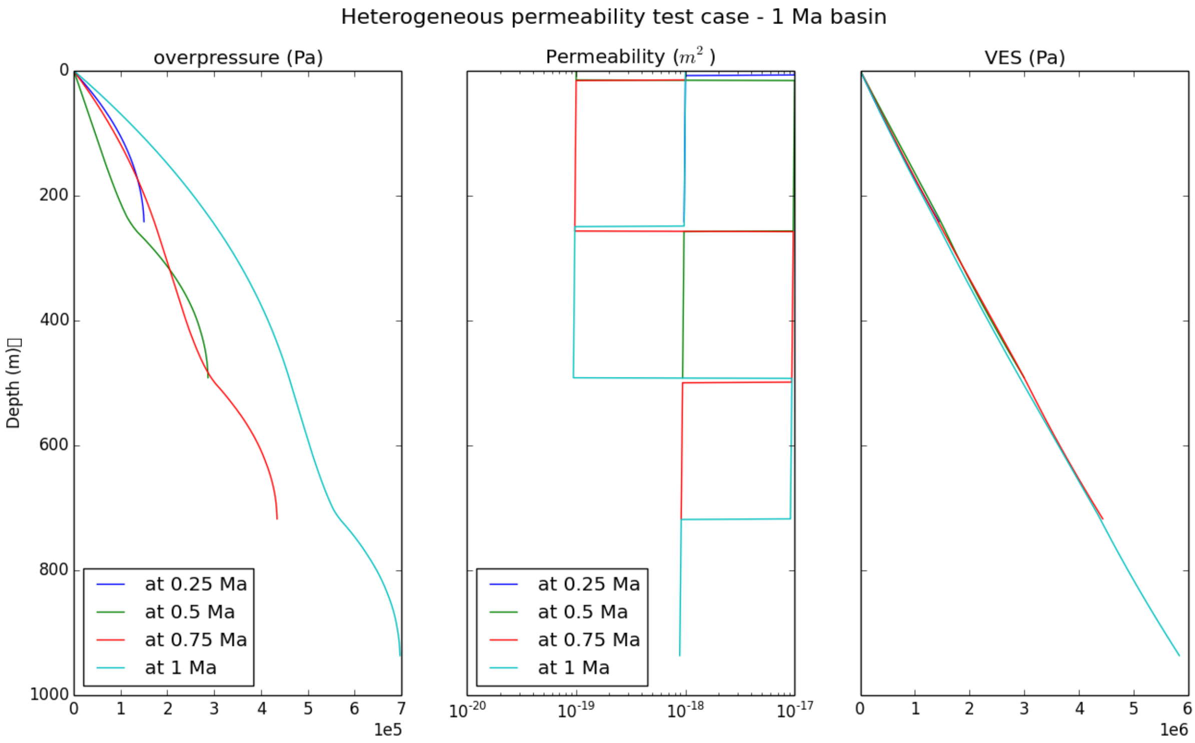

Figure 4.6: Overpressure and the permeability at four times in the basin evolution are shown. The colors represent the basin at a time around the introduction of a new layer and at the final 1 Ma time. The permeabilities of the layers from bottom to top is: $10^{-18}$, $10^{-17}$, $10^{-19}$, $10^{-18}m^2$.

### Thermal preferential flow

Here the point is to exhibit the temperature variations that come with heterogeneous thermodynamic properties, i.e., ratio of thermal conductivity to specific heat of different lithologies.

We consider a somewhat simplified model where we just use a diffusion coefficient in the assembly. We exploit the idea of having a spatially varying diffusion constant within a layer to highlight the 2-D nature of the temperature field, since typically there is an almost linear gradient to the basal heat flow. The physical properties are synthetic, namely, in the second material layer, we add two increasing gaussians to the value of the thermal diffusion, localized around 250 m and 750 m in the lateral dimension. The purpose of this artifice is to demonstrate lateral heterogeneity within the temperature field. We note that the material property *within* the material layer is varying. In Fig. 4.7, we can see that in regions of high thermal diffusion coefficient, the temperature drops compared to its

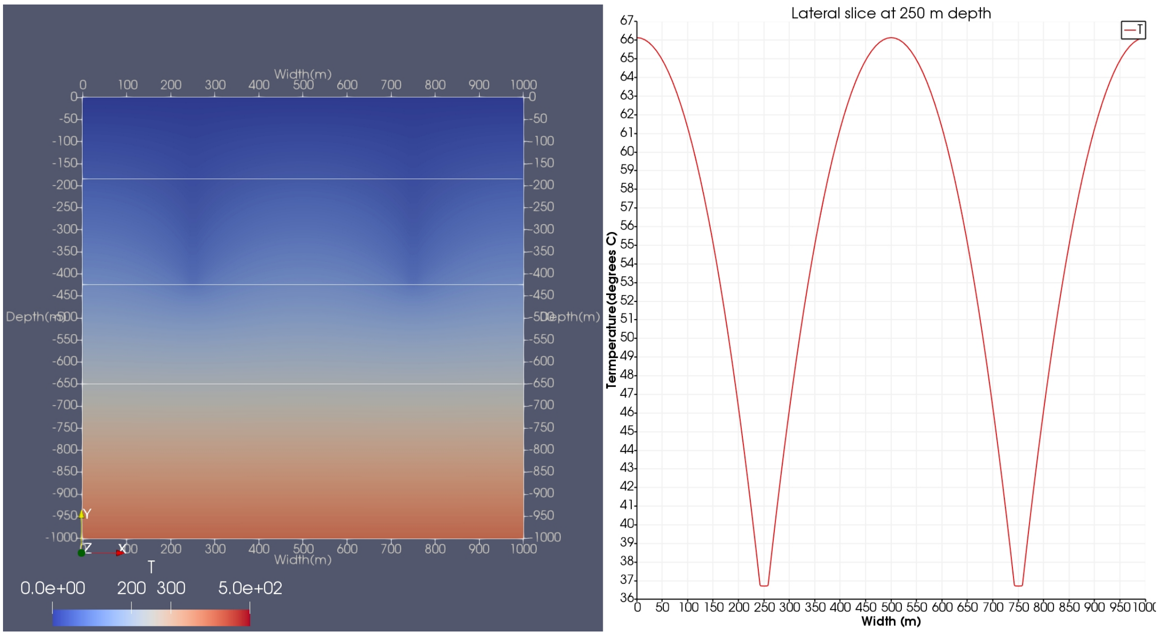

Figure 4.7: On the left, we see the temperature field in the 2D basin. The white lines are the interfaces between different material types. In the second layer, between 175 and 425 m depth, the thermal diffusion value is spiked at 250 and 750 m lateral position. This leads to the decrease in temperature around these regions, seen in the shape of the graph on the right.

surroundings, as expected. This is seen on the right side where there are two dips in the temperature profile taken across the basin at a depth of 250 m.

### 4.5.2 Parallel scalability

We have run simulations using the Jureca and the Juwels [70] supercomputers at Forschungszentrum Jülich (FZJ). The following scaling study on Juwels consists of running the Wangen test case, as described above, using a reduced sedimenation rate by half. Each of the regular nodes on Juwels is equipped with 48 cores. We have explored from 1 to 32 nodes (48-1536 cores). Additionally, the problem sizes were increased and run at different node counts to study weak scaling. As the core count was doubled, the problem size (number of DOFs) was also doubled. Starting with a 1km deep and wide basin, the basin width was doubled consecutively. This doubled the number of degrees of freedom, thus generating the series of problem sizes considered here, from 1 to 8 million DOFs. This allows for a combined study of strong and weak scaling.

We do not use hyperthreading and consider each core to be an MPI task or process. A breakdown of the simulation's timing by class method was obtained in addition to the total program run time. The data set includes three missing points (4 million DOFs at 48 and 768 proceses, 8 million DOFs at 768 proceses), which the scheduler did not accept for unknown reasons.

A feature of all the graphs is the dominance of communication overhead at the right side, beyond the 768 processes mark. In other words, there is not enough work per processor and communication time saturates the timing. The total program run times can be seen in Figure 4.8. The graph can be used to evaluate both strong and weak scaling. The theoretical limit to perfect strong scaling is the solid black line. The more parallel to this line a series at one problem size(one color) is, the better strong scaling efficiency. The horizontal grid labels can also be used to evaluate weak scaling, as the problem sizes double with doubling number of MPI processes. Thus, the more horizontal a series of increasing problem size points(multi-color), the better weak scaling efficiency. Before the turning point both the weak and strong scaling are good.

Next we consider the four individual solvers, not including temperature. Together these make up a significant portion of the total runtime. First, in figure 4.9 is the level set solver. The good weak and strong scaling results are an accomplishment for the algorithms used by Manuel Quezada de Luna and [65].

Second, in Figure 4.10, we inspect the overpressure solver. This graph demonstrates reasonable scaling for the DOFs/ MPI process ratio. It seems as though there would be more "room to run" at a higher DOFs/ MPI process ratio, i.e., shifting to the right.

Third, in Figure 4.11 and Figure 4.12, we look at the two SUPG solvers. Neither one suggests good scaling behavior in this DOFs/ MPI process range. This would be one place to look into for improvements.

The pressure and level set solver both use an AMG preconditioner with a conjugate gradient solver. The SUPG problems use a block Jacobi preconditioner with a GMRES solver. These differences should be considered when evaluating the scaling performance in this context of the former (good-relatively good) and the latter two (not good).

The ratio of DOFs/ MPI process should be increased in a future scaling study. Given constraints on

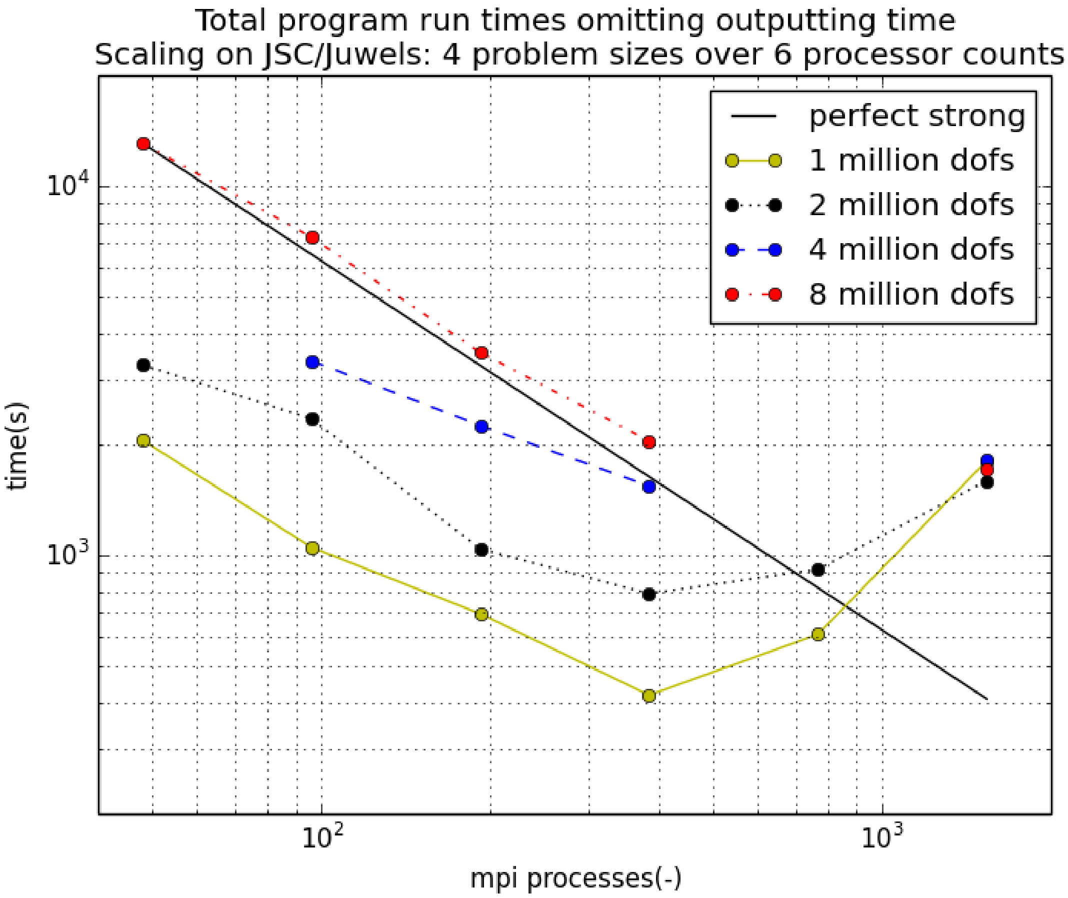

Figure 4.8: The total program run time, with the outputting time subtracted is shown. The graph can be used to evaluate both strong and weak scaling. The theoretical limit to perfect strong scaling is the solid black line. The more parallel to this line a series at one problem size(one color) is, the better strong scaling efficiency. The horizontal grid labels can also be used to evaluate weak scaling, as the problem sizes double with doubling number of MPI processes. Thus, the more horizontal a series of increasing problem size points(multi-color), the better weak scaling efficiency.

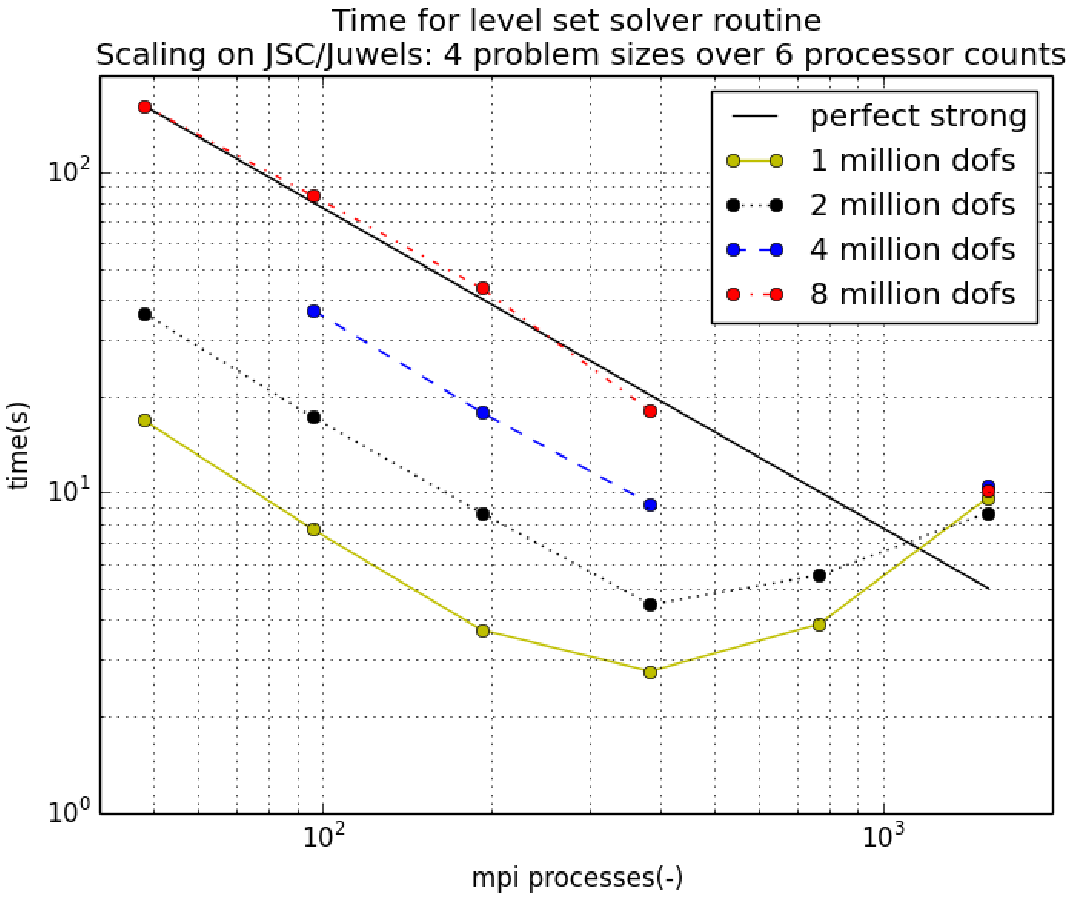

Figure 4.9: This presents timings for the level set solver. AMG preconditioner and conjugate gradient solver used. This graph demonstrates good scaling, both strong and weak.

resources, problem sizes and processor counts were conservative. This study provides an evidentiary basis for justifying the exploration of larger problem sizes at higher processor counts.

## 4.6 Conclusion and Outlook

In this work, we have presented a model using the finite element library deal.II that shows that geometrical complexity in rock layer facies can be represented by interfaces described through the level set method. The implicit representation of these interfaces allows for flexibility in modeling deformation in sedimentary basins. We have considered a general, abstract basin. In particular, we have shown that under the assumption of vertical effective stress, lateral variations in the layers can

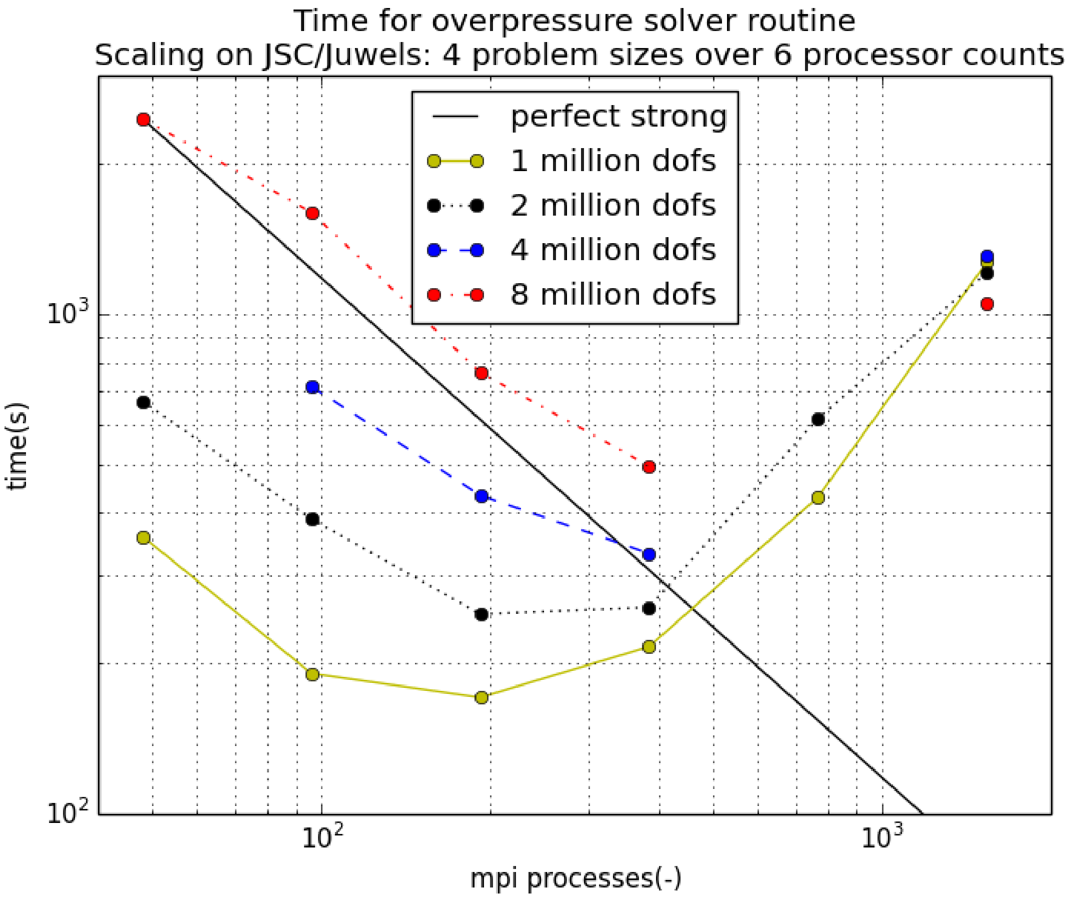

Figure 4.10: This presents timings for the overpressure solver. AMG preconditioner and conjugate gradient solver used. This graph demonstrates reasonable scaling for the DOFs/ MPI process ratio. It seems as though there would be more “room to run” at a higher DOFs/ MPI process ratio, i.e., shifting to the right.

be produced. We have shown that the model can approximate the overpressure that results from a consolidation process, where an analytical solution is available. We have shown that the model exhibits increases in overpressure in the presence of strong permeability constrasts that are in line with the behavior of low permeability rock seals. We have demonstrated that a choice of material properties, namely thermal diffusivity, can be used to exhibit lateral heterogeneity in the 2-D temperature field. Together, these demonstration cases of the model suggest that this implicit representation of the geometrical interfaces can be used to produce a basin model without modifying the mesh. Furthermore, we have conducted a preliminary scaling study to assess the parallel performance of the code when executing the verification example. The overall conclusion is that the code scales well in terms of

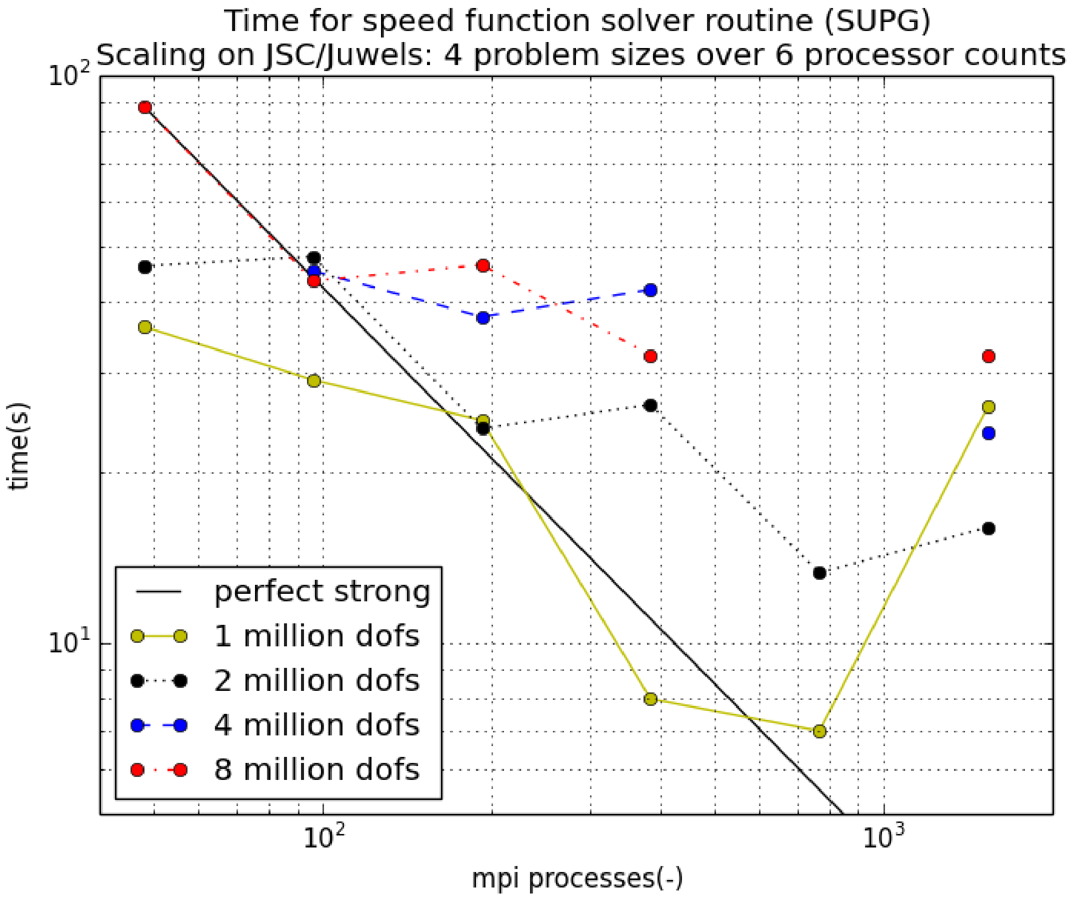

Figure 4.11: The speed function scaling is shown. The SUPG problems use a block Jacobi preconditioner with a GMRES solver. The results here do not indicate clear scaling patterns.

overall wallclock time, with the potential for improvement in individual physics and solver components of the coupled approach.

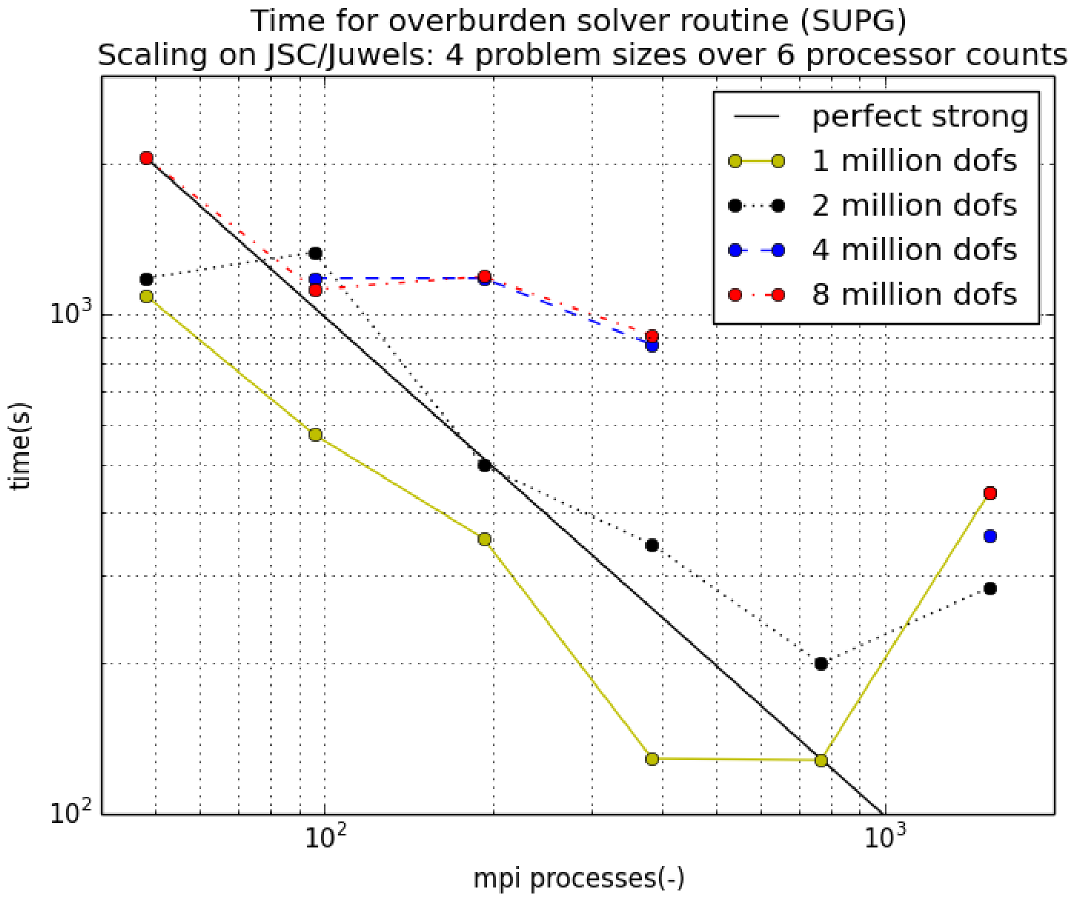

Figure 4.12: Here is the overburden solver scaling. The SUPG problems use a block Jacobi preconditioner with a GMRES solver. The results here do not indicate clear scaling patterns.

CHAPTER 5

# 3-D, Speed Function Modification and the Foundations for Dynamic Local Adaptivity

## 5.1 Introduction

In this chapter, we consider the elaboration of the simulation code along three axes. The first is to implement the infrastructure to run the code in 3-D. The second is to consider incorporating other physical processes in the speed function. The last is to consider dynamic local adaptivity.

## 5.2 3-D Code Infrastructure

Basin modelers are naturally interested in 3-D domains. The number of degrees of freedom rises very rapidly in 3-D and the computations can quickly become very expensive. A small toy version of a simple domain was run to just check if all the classes behaved properly in 3-D. As seen in Fig. 5.1, the interface advances about a quarter of the way through the domain. Then the computation was cut off. It is just an illustration that the code runs in a 3-D context.

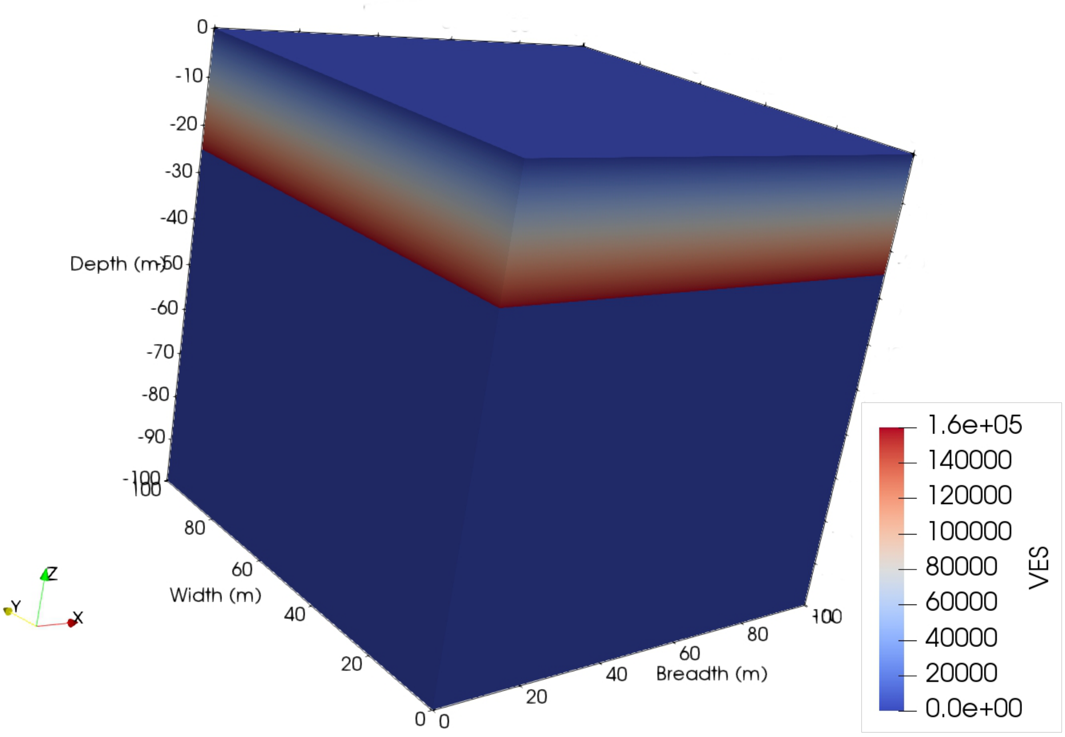

Figure 5.1: This presents a basin with one rock layer at the end of a depositional period of $0.1Ma$. The vertical effective stress (VES) in units of Pa is shown, with the basement, the bottom blue layer, taking the value 0. This small toy example exhibits the code's 3-D infrastructure.

This is mostly facilitated through the C++ templating used in deal.II. The dimension is passed as a template parameter to the constructors of the data types. Additionally, certain functions use switch statements to return the appropriate data types based on dimension. Additional solution vectors were also required.

## 5.3 Physical Process Extension

The second is to consider the addition of a physical process to the geometrical changes of rock layer interfaces. Inspired by the notion of buoyancy, we add an additional term to the speed function that determines the rock layer interface positions.

We consider the following basin setup with the standard speed function as earlier. A four layer sequence, each layer added at the same rate over 0.25 Ma. In Fig. 5.2, we see how a given basin with four homogeneous layers, all the same rock density, looks after 1 million years.

In the next setup, we change the rock density of one layer to contrast with its neighbors. Namely, it is made significantly lighter, such that we might expect a difference in final layer configuration. In Figure 5.3, we see the basin with four layers with the same material properties, except the second layer added has a solid rock density of 1/4 the others. One might intuitively expect that the lighter layer would experience a buoyant force upward due to its large density difference. This is not the case. The interface locations are the same as before. Given the general motivation to better understand salt dynamics, one might then look for a way to incorporate buoyancy-like effects in the model.

This motivates the following experiment to include a buoyancy inspired effect in the simulation code. This is done by examining the speed function, the driver of geometric change.

First, we look at an example of an extension of the speed function to include other physical processes. The concept of separating the geometry from the physical processes is one of the main aspects of this work. The speed function for the level set potential functions is where they interface. In the preceding chapters we have considered sedimentation and compaction as physical processes, encoded in Equation (2.22), reproduced here:

$$F = F(p_o) = \omega + \int_0^{z'} \frac{1}{1-\phi}\frac{\partial \phi(p_o)}{\partial t} dz \tag{5.1}$$

We add a term to Eq. (5.1) to relate the geometrical movement to the influence of a physical process inspired by buoyancy. We consider the intuitive idea that porous rock, with solid matrix that has a rock density much lower than that of a later addition of a denser rock, should experience something of an uplift compared to the speed of the denser rock. We look to the overburden to get a sense of the density variations in the region above a particular point. Roughly we will integrate above a point in the column, and look at the ratio to the slope of the overburden compared to the actively sedimenting material.

$$F = F(p_o) = \omega + \int_0^{z'} \frac{1}{1-\phi}\frac{\partial \phi(p_o)}{\partial t} dz - \alpha \int_0^{z'} (1 - \frac{\rho_b(z)}{\rho_b^\circ})\, dz, \tag{5.2}$$

where $\rho_b^\circ$ is the bulk density of the actively deposited material, and $\alpha$ $[T^{-1}]$, is a proportionality

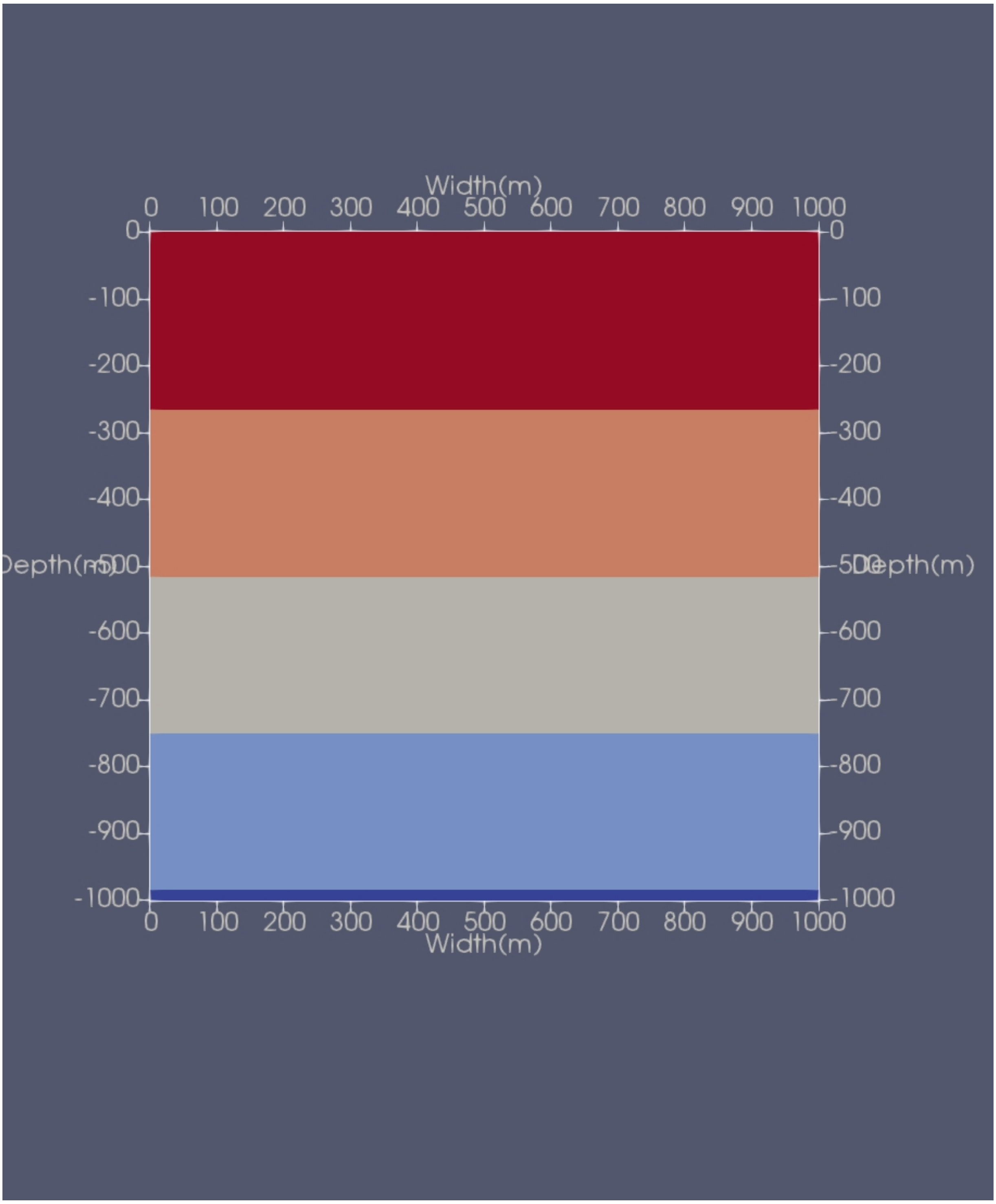

Figure 5.2: In this image, we see a basin at the end of its depostional period of $1Ma$. The lowest blue layer is the basement and the other colors represent different layers. The layers were added equitemporally and all have the same material properties.

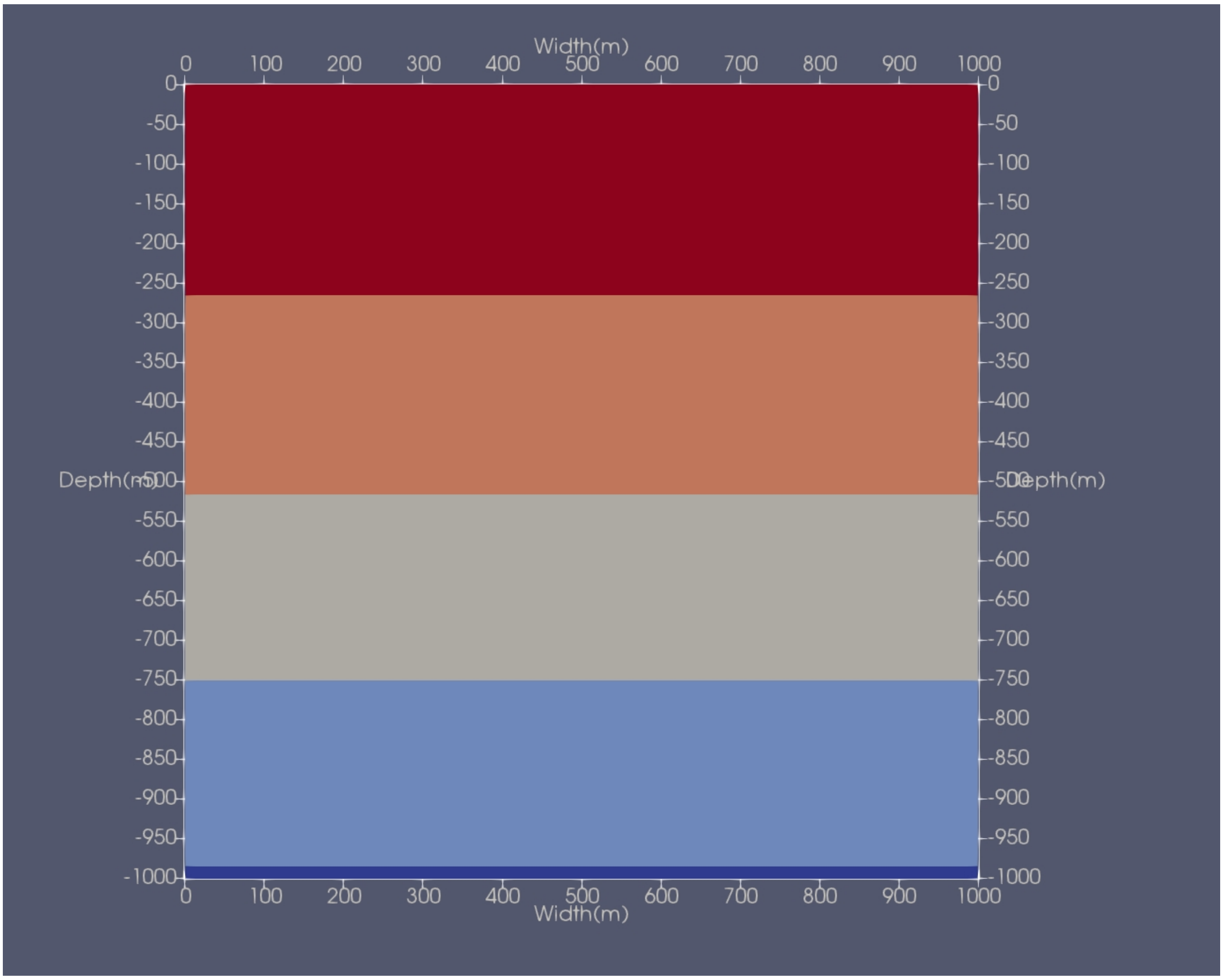

Figure 5.3: Here we see a basin at the end of its depostional period of $1Ma$. The lower blue layer is the basement and the other colors represent different layers. The layers were added equitemporally and have the same material properties, except for the second layer added, that has a solid rock density of $1100\frac{kg}{m^3}$, while all others have $4400\frac{kg}{m^3}$. One might intuitively expect that the lighter layer would experience a buoyant force upward due to its large density difference, however these layers are at the same places as in the previous image.

coefficient.

So, the idea is that if the bulk density at the current point is the same as the actively deposited material, there is no change. If the current bulk density is less than the actively deposited, then some speed is substracted from the total speed function, proportional to the coefficient $\alpha$.

This is just one form this could take. In a more comprehensive sense, we could add a general function, $g$, of densities to represent more complicated relations, if desired, as in,

$$F = F(p_o) = \omega + \int_0^{z'} \frac{1}{1-\phi} \frac{\partial \phi(p_o)}{\partial t} dz - \int_0^{z'} g(\rho_b(z))\, dz. \tag{5.3}$$

Subsequently, we use Eq. (5.2) for definiteness.

Following the procedure of converting this integral equation into a PDE, we apply the gradient operator, then dot it with the vertical unit vector, as done in Eq. (4.40), leading to

$$\hat{e_z} \cdot \nabla F = \frac{1}{1-\phi} \frac{\partial \phi}{\partial t} - \alpha(1 - \frac{\rho_b}{\rho_b^\circ}) \tag{5.4}$$

This is the speed function that is used to build the right-hand side in the assembly for this system's solution. To consider how this manipulation to the speed function affects a basin, we look at the following case.

Now, we use the modified speed function based off of Eq. (5.2). We note that the right hand side is on the order of $10^{-6}$, and so we set $\alpha$ to the same. We expect to see that the layers have not descended as much. In Figure 5.4, we see the result.

It appears as if the light rock has compacted. However, it has not, as confirmed in its porosity profile. This, then, is an example of a loss of mass conservation.

In this vertical effective stress (VES) context, if the porosity is reduced, this means that pore fluid has been expelled upward. In this case, we see that the layer is of a smaller thickness (thanks to the slower speed function), but we do not have a coherent account of why there is a reduction in thickness, as we do when compaction occurs. Using a speed function based off of Eq. (5.3) would have the same drawback, true for any non-trivial function $g$.

So, perhaps, to account for the volume reduction, and lower the porosity to be consistent, we could

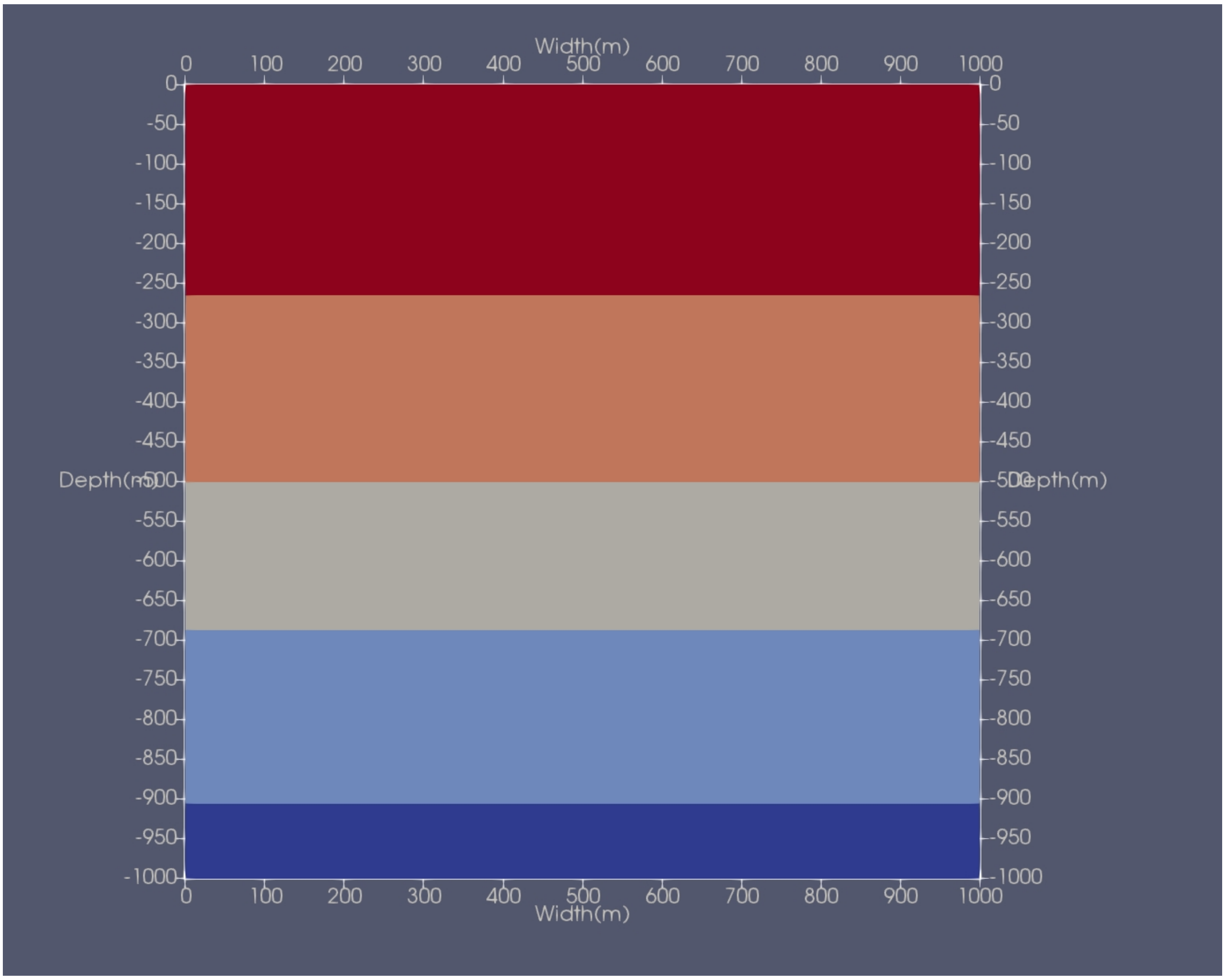

Figure 5.4: In the case of the modified speed function, this is the basin at the end of its depostional period of 1 Ma. The lowest blue layer is the basement and the other colors represent different layers. The layers were added equitemporally and have the same material properties, except for the second layer added, that has a solid rock density of $1100\frac{kg}{m^3}$, while all others have $4400\frac{kg}{m^3}$. One might intuitively expect that the lighter layer would experience a buoyant force upward due to its large density difference. We see that the second added layer has not descended as much, in fact it is significantly thinner than its $240m$ thickness.

increase the VES, by incorporting a buoyant force into the stress consideration. Namely, not only do we have an overburden acting downward, we have a buoyant force acting upward, both contributing to the stress. This would correspond to an increase in the effective stresses on a cell, and then to a larger compaction. But, this then would eliminate the need for the additional speed function term, since the buoyant effect would be affecting the porosity, acting through the definition VES. Changes in porosity are already accounted for in the unmodified speed function.

This leads to the idea of generalizing to a 2/3-D effective stress, namely where stresses on each face of the cell are considered (neglecting shear stresses). In our vertical effective stress setup, there is not the flexibility to allow a lighter layer to move up relative to another while maintaining mass conservation. If the lighter layer is to move up, pore fluid must be ejected.

A proper treatment of buoyancy would require, in the case of large density differences, materials to move past one another. So, one would be moving laterally to make way for the lighter one. As we are still operating with the assumption of vertical effective stress, this cannot be represented in the model as is.

### 5.3.1 Relaxing the vertical effective stress assumption

We can start by just considering the normal stresses in 3-D.

$$\sigma_{ij} = \sigma_{ij} - \mathbf{1}p_i \tag{5.5}$$

In the VES context, there is a compaction law, that makes a direct link between the stress and the strain, though in the form of porosity change. Porosity is a volumetric concept, while strains are linear in each dimension. If we are considering movement in only one direction, then porosity change can be interpreted as a movement in that direction. The compaction law requires either a reformulation in terms of strains, or an interpretation suitable such that a 3-D vector field can be derived from it.

Furthermore, a new interpretation of the level set potential function is required. Currently, the interfaces are treated as a layer cake, with a defined ordering, such that the material identity of a layer can be extracted at a point by considering a sum over all of the level set potential functions' values at

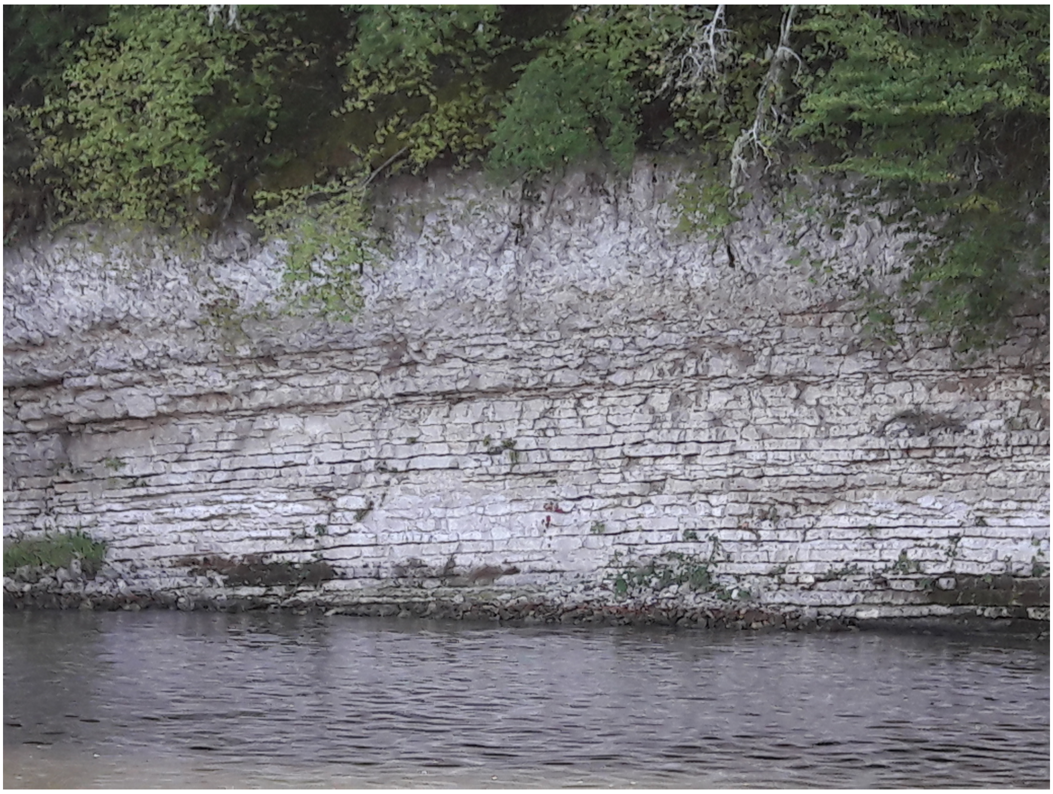

Figure 5.5: This photograph shows sedimentary rock, along the Dordogne river in central France, outside Sarlat. The tree branches give the scale. Photo taken by author.

that point. This effectively defines a material as being between two level set interfaces. If we consider the speed function as a 3-D vector field, then this ordering principle would need to change to one level set per layer. Presently, it's unclear how to do this in a coherent way.

## 5.4 Towards Dynamically Adaptive Mesh Refinement

### 5.4.1 Overview

In this section, we introduce the ideas that go along with dynamic local adaptive grid refinement. This means that the cells have varying width/area/vol. Algorithms that are dependent on the assumption of a uniform cell size, must be reorganized to include the change to the basis functions and the sizes of data structures. Deal.II has made most of these processes very user friendly (cf. tutorial step-6). Complications arise for parallel distributed setups and for time-dependent problems, e.g., tutorial step-32. In our case, we have a number of solution vectors which are not contained in a vector system. Each field is treated as a scalar and the solutions are obtained sequentially. Part of the reason for this was to facilitate interfacing with the level set solver, which was not written to use a vector structure (FESystem in dealii language). So, the challenge is to have the pore pressure, and associated fields, interact with the level set potential functions. There is some bookkeeping involved. This requires modifying the setup functions in the level set solver to be used repeatedly through the simulation, when a change of the grid occurs, rather than a one-time call at the beginning.

When considering adaptivity, where to adapt the grid is of special importance. In our case, we would proceed in one of two ways. The first is to adapt the grid around the interfaces between our layers, as they are regions of potentially rapid change in the fields due to the material discontinuities. The second approach is to use the a posteriori Kelly error estimator to focus on the cells where the gradient of the solution field is high.

### 5.4.2 Adaptive algorithm

The general framework for an adaptive mesh algorithm is the loop:

- Evaluate the solution.
- Mark the cells for refinement or coarsening based on the error estimator or location.
- Refine and coarsen cells, thereby generating a new mesh.
- Interpolate solutions onto the new mesh.

The level set potential is a structure that is an excellent candidate for adaptive grids, as interest is restricted around the 0 level set. What happens "far-away" is of much less importance. This has led to the narrow-band level set method, see e.g., [52], where the level set potential function is considered only over a fixed width (narrow-band) around the 0 level set.

In cases where explicit solvers are used, as in the level set solver, the CFL condition needs to be updated to reflect the smallest cell size. This reduces the time step, and might be a limit to how many levels of refinement are desirable, in order to keep a reasonable global time step. Another strategy, would be to have a hierarchy of time step sizes, such that in certain parts of the domain with high refinement, more time steps of smaller size would be taken in the span of one larger time step. This would be an area of complexity to explore, though maintaining a reasonable ratio between smallest time step and smallest cell size is straightforward enough.

The simulation code has been modified for adaptive refinement, though more work is needed to coax a reasonable result out of the level set solver. The following images show the mesh before and after refinement. In Fig. 5.6, we see the mesh lines of a uniformly refined domain. The resolution is the same over the domain. In Fig. 5.7, the mesh has been adaptively refined once, with the refinement occurring around the top of the domain, where the level set interface enters the domain. Away from the interface, the mesh is coarsened. The solution vectors have been interpolated onto the new mesh. This is done through the "SolutionTransfer" method in the deal.II library. Presently the limitation is that the level set solver does not return proper values on the new mesh.

This is attributed to the complex inner workings of the algorithms that go into it. Rather than reproducing details here, the reader is encouraged to read [65] for details on their method of considering entropy production to adjust artificial compression and entropy-viscosity stabilization for the transport equation. The paper states that their algorithms are suited to unstructured meshes, though the implementation obtained in the dealii code gallery is for uniform meshes. It is outside the scope of this thesis to adapt the coded algorithms for local adaptivity, and is left for future work.

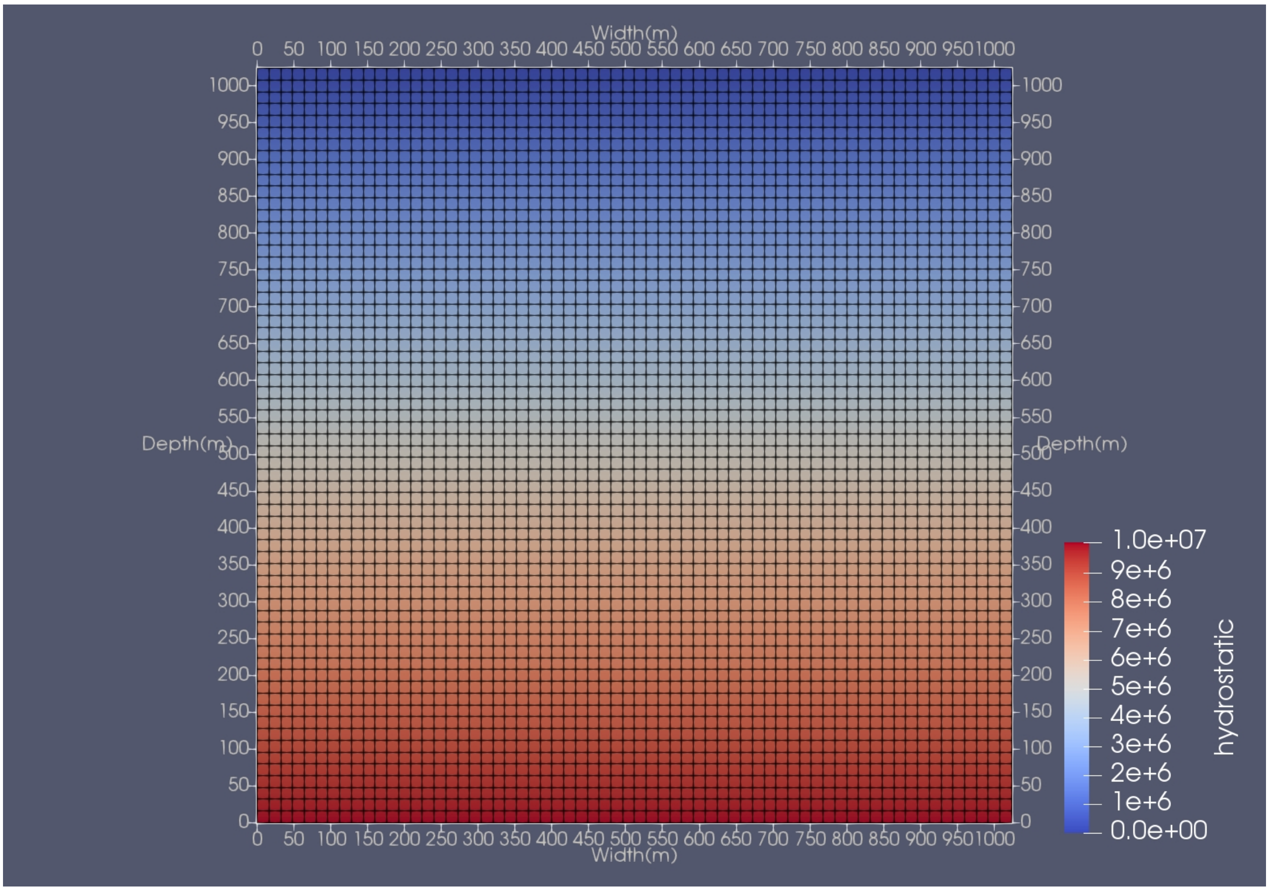

Figure 5.6: In this image we see the mesh lines in black over the domain, along with a hydrostatic pressure (Pa) field.

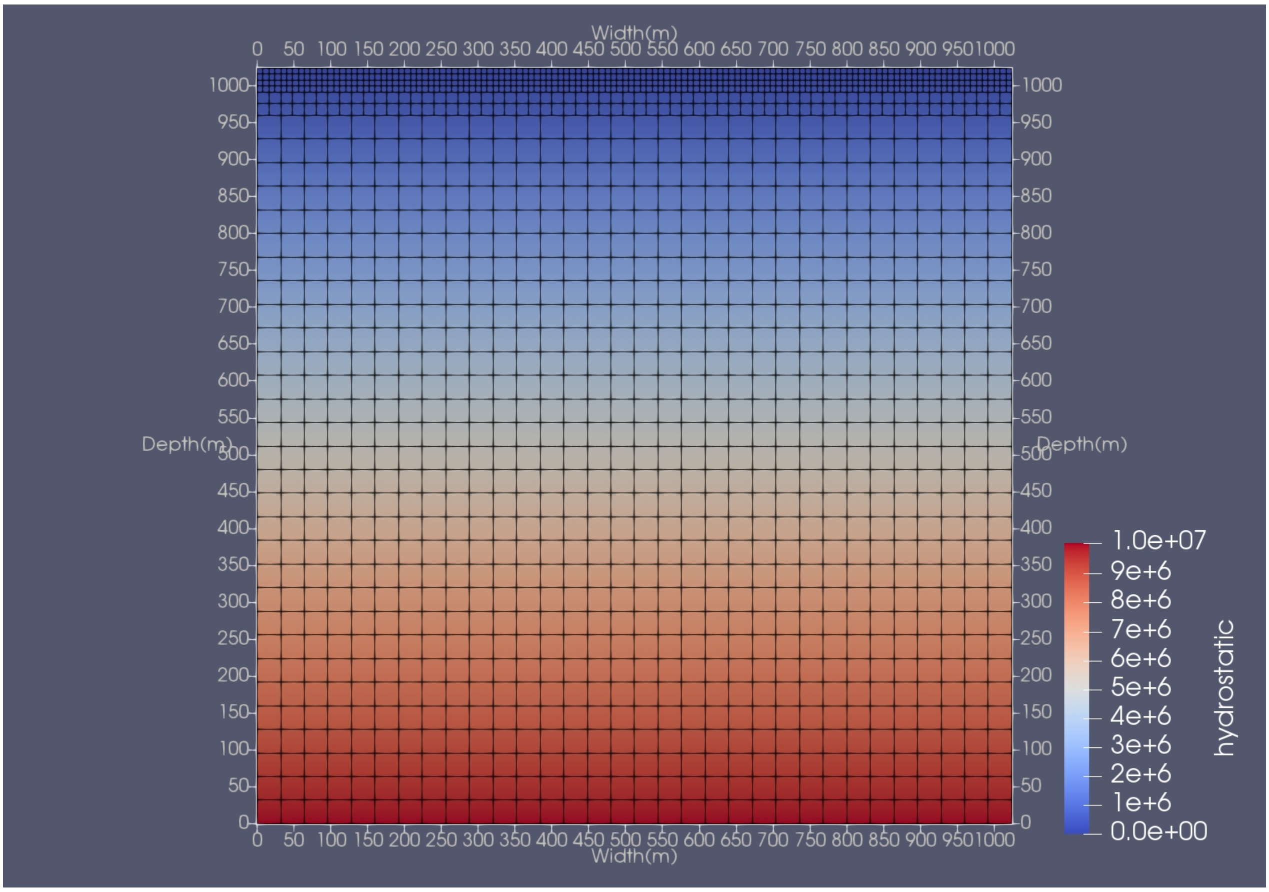

Figure 5.7: This image presents the adapted mesh lines in black over the domain, along with a hydrostatic pressure (Pa) field. We have marked cells near the first interface for refinement, and have coarsened the cells far away. We see here the deal.II strategy for hierarchical refinement, whereby each cell's neighbor cannot be more than 1 level refined or coarsened from it.

CHAPTER 6

# Summary and Outlook

## 6.1 Introduction

In this chapter, we conclude the work. We summarize the results obtained in the previous chapters and we lay out some possible paths for development.

## 6.2 Summary

In the preceding chapters, we have presented a novel concept to represent the layered geometry of sedimentary basins through application of the level set method. In was shown that under sedimentation the movement of compacting layers can be represented with a speed function, responsible for advancing the layer interfaces, that is coupled to a physical process, namely overpressure distribution. Furthermore, the material properties of the layers have effects on the basin's total physical state.

We have developed a finite element model of this coupling, with the library deal.II, that can be run in a parallel distributed memory setting. A scaling study has been conducted that evidences overall scalability, in the sense of reducing runtime with more resources. In some parts of the code the parallel

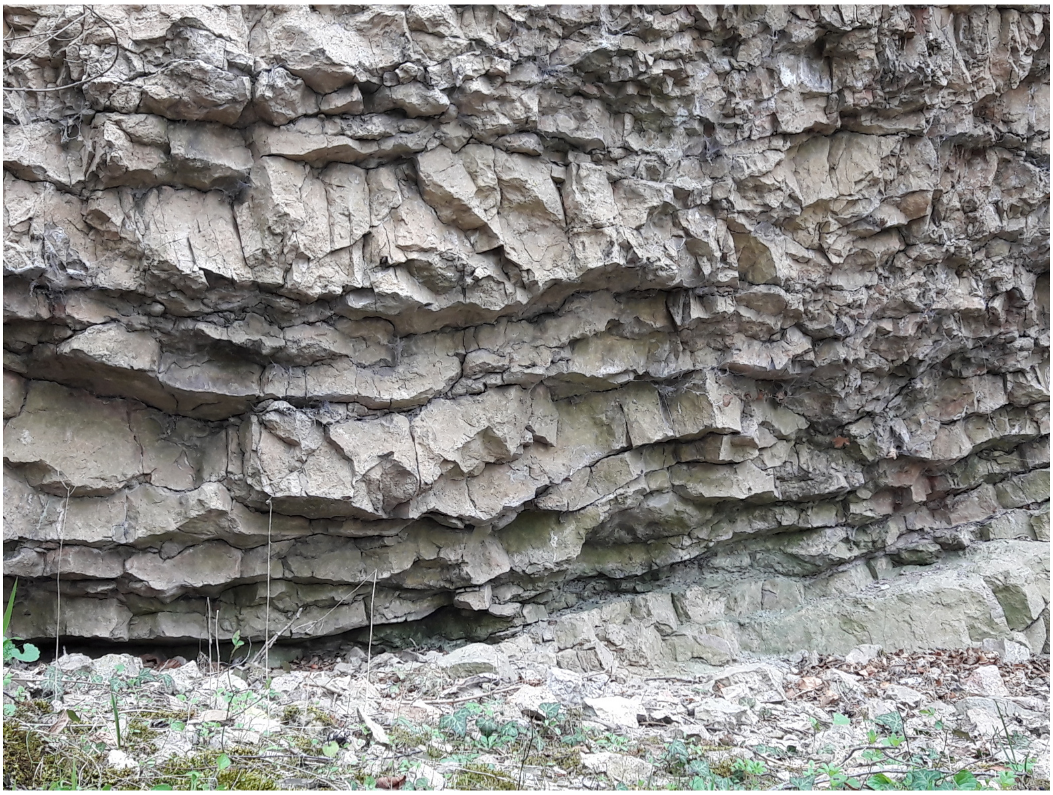

Figure 6.1: This photograph shows sedimentary rock, in the dolomite rock formations surrounding Gerolstein, Germany. The vegetation gives the scale. Photo taken by author.

efficiency is good, and in other parts would benefit from greater efficiency, through optimizing current algorithms or replacement with better scaling algorithms.

Additionally, the foundations for running the simulation with dynamic local adaptivity have been explored. Developments to the code to make use of this technique have been implemented for data structures, with the exception of the level set solver code, which has been treated as a black box in this thesis. Changes required to the time-stepping and other assumptions have been considered.

A temperature field is also included, since the thermal history of a basin is one of the determining factors for the chemical composition of its constituents. Depending on the depth of the basin, this is largely under the control of the basal heat flow from lower regions of the lithosphere. We have shown how material properties can have an effect (synthetic in our case) on the temperature distribution laterally.

## 6.3 Outlook

We consider some modifications to the simulation code that could be undertaken in the future. They are listed in order of how remote they are from the current status.

### 6.3.1 Larger Problems

While the code's infrastructure is set up for 3-D simulations, 3-D simulations require a much larger number of degrees of freedom to discretize the problem. Reducing the number of degrees of freedom through locally refining the mesh would be a great value to working in 3-D. Implementing the required changes to the assembly of the assorted matrices in the level set solver would enable the simulation to undergo local refinement. With attention to the time-stepping, this would open up new territory to explore.

### 6.3.2 More Comprehensive Material Models

The class that contains the material data is separated from the rest of the simulation's methods. This was chosen such that adding a more robust material model or set of data can be easily hooked into the simulation. As it is now, depending on the material identification number of a layer, the relevant physical property is returned, e.g., surface porosity, solid rock density. This is crucial to introduce varying material properties per layer.

One modification to the existing material class is to return a function rather than just a value. For example, in the temperature example in Chapter 4, the thermal diffusivity was not just a value based on the material id, but rather a function that depended on the x-coordinate of the cell in which its value was called. So, the thermal diffusivity could vary within a particular layer. This was accomplished by passing an x-coordinate, along with the material id as a parameter to the return function. The idea of a more complicated data structure to return material properties at certain places has a lot room to explore. In particular, it would go towards introducing a geological basin data set.

### 6.3.3 Initial Conditions of Layers

Along the lines of considering bringing data into the simulation, here we consider how to approach a situation where instead of evolving the basin from the surface, we can map a set of layers onto an existing complex geometry. For example, if there is an indication of the layer divisions from seismic, a selection of the layer interfaces can be mapped onto them.

The key idea is to change the initial condition of the level set function per layer, including the moment of entry into the simulation. From the data, one can extract a smooth curve, or interpolate one, and then those positions become the 0 level set of the function. We then solve a "pseudo-time" equation to, essentially reinitialize the level set potential function into a signed distance function.

$$\frac{d\phi}{dt'} = sign(\phi)(1 - |\nabla\phi|) \tag{6.1}$$

This would then be a level set with the interface over the extracted curve. Beginning with hydrostatic conditions, one could then see how the states evolve, or force them in certain ways, depending on the basin conditions.

### 6.3.4 Introduce Reinitialization

We have chosen to use the level set solver that includes two techniques, anti-diffusion and an entropy-viscosity smoothing. It does not include reinitialization, which is the periodic solution to the above equation, that essentially resets the level set potential function to a desired state, a signed-distance function. This technique has been used extensively in other level set contexts, though the level set solver here uses other methods to ensure accuracy and stability. Reinitialization should not affect the location of the interface, though in practice this has been an issue leading to a loss of mass conservation in two-phase flow, for example. As the field for first order hyperbolic equations evolves, advances from the literature could be incorporated into this crucial aspect of the simulation.

### 6.3.5 Relaxing the Vertical Effective Stress Assumption

As considered in the previous chapter, a generalization of the vertical effective stress to include lateral stresses is desirable. The general notion is straightforward in considering the stress tensor in 3D.

$$\sigma_{ij} = \sigma_{ij} - \mathbf{1}p_i \tag{6.2}$$

The complexity comes when trying to clarify how these stresses would relate to the speed function of the level sets.

### 6.3.6 Considering Faults and Fracturing

Faults and fracturing are an important part of geological systems. They play crucial roles in the fluid dynamics in a basin, as often faults are preferential pathways for fluid flow. At the reservoir scale, this is treated with double porosity/double permeability models. Level sets have been used to introduce discontinuities at fractures. Our use of level sets to represent dynamic layer interfaces is different. Furthermore, our use of an effective stress principle is a simplification of the rock solid mechanics. To get fracturing one would need to use a more comprehensive constitutive model. It is not clear presently how to extend the current model to incorporate these phenomena.

# References

[1] M. Tucker, *Sedimentary Petrology: An Introduction to the Origin of Sedimentary Rocks*, Wiley, 2009 (cit. on pp. 1, 43).

[2] T. Hantschel and A. Kauerauf, *Fundamentals of Basin and Petroleum Systems Modeling*, Springer, 2009 (cit. on pp. 3, 7, 11, 13, 16, 43).

[3] S. McGovern et al., *Novel basin modelling concept for simulating deformation from mechanical compaction using level sets*, Computational Geosciences **21** (2017) 835, ISSN: 1573-1499, URL: `https://doi.org/10.1007/s10596-017-9643-2` (cit. on pp. 5, 44, 59).

[4] P. H. Ungerer and R. Pelet, *Extrapolation of the kinetics of oil and gas formation from laboratory experiments to sedimentary basins*, (1987) (cit. on p. 7).

[5] H. S. Poelchau et al., "Basin Simulation and the Design of the Conceptual Basin Model", *Petroleum and Basin Evolution*, ed. by P. D. D. h. c. D. H. Welte, D. B. Horsfield and P. D. R. Baker, Springer Berlin Heidelberg, 1997 3 (cit. on p. 7).

[6] K. Tuncay and P. Ortoleva, *Quantitative basin modeling: present state and future developments towards predictability*, Geofluids **4** (2004) 23 (cit. on p. 7).

[7] D. J. Benson, *Computational methods in Lagrangian and Eulerian hydrocodes*, Computer methods in Applied mechanics and Engineering **99** (1992) 235 (cit. on p. 8).

[8] J. Donea et al., "Arbitrary Lagrangian Eulerian Methods", *Encyclopedia of Computational Mechanics*, John Wiley & Sons, 2004 (cit. on p. 8).

[9] P. Fullsack, *An arbitrary Lagrangian-Eulerian formulation for creeping flows and its application in tectonic models*, Geophysical Journal International **120** (1995) 1 (cit. on pp. 8, 44).

[10] M. Nazem et al.,
*Arbitrary Lagrangianâ–Eulerian method for large-strain consolidation problems*,
International Journal for Numerical and Analytical Methods in Geomechanics **32** (2008) 1023
(cit. on pp. 8, 44).

[11] S. Osher and J. A. Sethian, *Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations*, Journal of computational physics **79** (1988) 12
(cit. on pp. 8, 50).

[12] J. A. Sethian, *A review of recent numerical algorithms for hypersurfaces moving with curvature dependent speed*, J. Differential Geometry **31** (1989) 131 (cit. on pp. 8, 50).

[13] M. Sussman et al., *An adaptive level set approach for incompressible two-phase flows*,
Journal of Computational Physics **148** (1999) 81 (cit. on p. 8).

[14] A. Villa and L. Formaggia, *Implicit tracking for multi-fluid simulations*,
Journal of Computational Physics **229** (2010) 5788 (cit. on p. 8).

[15] P. Frolkovic, *Application of level set method for groundwater flow with moving boundary*,
Advances in Water Resources **47** (2012) 56 (cit. on p. 8).

[16] J. A. Sethian and J. Straint, *Crystal growth and dendritic solidification*,
Journal of Computational Physics **98** (1992) 231 (cit. on p. 8).

[17] R. Malladi and J. A. Sethian, *Image processing via level set curvature flow*,
Proceedings of the National Academy of Sciences **92** (1995) 7046 (cit. on p. 8).

[18] B. A. Schrefler and R. Scotta,
*A fully coupled dynamic model for two-phase fluid flow in deformable porous media*,
Computer methods in applied mechanics and engineering **190** (2001) 3223 (cit. on p. 8).

[19] S. E. Minkoff et al., *Coupled fluid flow and geomechanical deformation modeling*,
Journal of Petroleum Science and Engineering **38** (2003) 37 (cit. on p. 9).

[20] C. M. Bethke, *A numerical model of compaction-driven groundwater flow and heat transfer and its application to the paleohydrology of intracratonic sedimentary basins*, Journal of Geophysical Research: Solid Earth **90** (1985) 6817 (cit. on p. 9).

[21] C. M. Bethke, *Modeling subsurface flow in sedimentary basins*, Geologische Rundschau **78** (1989) 129 (cit. on p. 9).

[22] D. M. Audet and A. C. Fowler, *A mathematical model for compaction in sedimentary basins*, Geophysical Journal International **110** (1992) 577 (cit. on p. 9).

[23] Z. Chen et al., *Integrated Two-Dimensional Modeling of Fluid Flow and Compaction in a Sedimentary Basin*, Computational Geosciences **6** (2002) 545 (cit. on p. 9).

[24] E. Kikinzon et al., *Modeling fluid flow and compaction in sedimentary basins using mixed finite elements*, Computational Geosciences (2015) 1 (cit. on p. 9).

[25] L. M. Christopher, *Forward Modeling of Compaction and Fluid Flow in the Ursa Region, Mississippi Canyon Region, Gulf of Mexico*, PhD thesis: Pennsylvania State University, 2006 (cit. on p. 9).

[26] M. Longoni, A. C. I. Malossi and A. Villa, *A robust and efficient conservative technique for simulating three-dimensional sedimentary basins dynamics*, Computers & Fluids **39** (2010) 1964 (cit. on p. 10).

[27] M. Longoni et al., *An ALE-based numerical technique for modeling sedimentary basin evolution featuring layer deformations and faults*, Journal of Computational Physics **230** (2011) 3230 (cit. on pp. 10, 44).

[28] K. Terzaghi, *Erdbaumechanik auf bodenphysikalischer Grundlage*, Franz Deuticke, 1925 (cit. on pp. 11, 44).

[29] R. de Boer, *Theory of Porous Media – Past and Present*, ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift fuer Angewandte Mathematik und Mechanik **78** (1998) 441 (cit. on p. 11).

[30] R. E. Gibson, *The progress of consolidation in a clay layer increasing in thickness with time*, Geotechnique **8** (1958) 171 (cit. on p. 11).

[31] L. F. Athy, *Density, Porosity, and Compaction of Sedimentary Rocks*, AAPG Bulletin **14** (1930) 1 (cit. on p. 14).

[32] D. B. Bahr et al., *Exponential approximations to compacted sediment porosity profiles*, Computers & Geosciences **27** (2001) 691 (cit. on p. 14).

[33] J. A. Sethian, *Theory, algorithms, and applications of level set methods for propagating interfaces*, Acta numerica **5** (1996) 309 (cit. on p. 14).

[34] S. Osher and R. Fedkiw, *Level set methods and dynamic implicit surfaces*, vol. 153, Springer Science & Business Media, 2003 (cit. on pp. 14, 16, 17, 44).

[35] J.-D. Yu, *Two-phase viscoelastic jetting*, Lawrence Berkeley National Laboratory (2009) (cit. on p. 15).

[36] J. Vila, C. Gonzalez and J. LLorca, *A level set approach for the analysis of flow and compaction during resin infusion in composite materials*, Composites Part A: Applied Science and Manufacturing **67** (2014) 299 (cit. on p. 15).

[37] C. Min, *On reinitializing level set functions*, Journal of Computational Physics **229** (2010) 2764 (cit. on pp. 16, 50).

[38] M. Sussman, P. Smereka and S. Osher, *A level set approach for computing solutions to incompressible two-phase flow*, Journal of Computational physics **114** (1994) 146 (cit. on pp. 16, 50).

[39] D. Hartmann, M. Meinke and W. Schroeder,
*Differential equation based constrained reinitialization for level set methods*,
Journal of Computational Physics **227** (2008) 6821 (cit. on p. 16).

[40] D. Adalsteinsson and J. A. Sethian,
*The fast construction of extension velocities in level set methods*,
Journal of Computational Physics **148** (1999) 2 (cit. on p. 17).

[41] M. Wangen, *Physical Principles of Sedimentary Basin Analysis*,
Cambridge University Press, 2010 (cit. on pp. 19, 21–24, 59).

[42] W. Bangerth, R. Hartmann and G. Kanschat,
*deal. II - a general-purpose object-oriented finite element library*,
ACM Transactions on Mathematical Software (TOMS) **33** (2007) 24 (cit. on pp. 35, 58).

[43] A. Ern and J.-L. Guermond, *Theory and Practice of Finite Elements*, en,
Applied Mathematical Sciences, Springer-Verlag, 2004, ISBN: 978-0-387-20574-8
(cit. on p. 35).

[44] H. Sabetamal et al., *Large deformation dynamic analysis of saturated porous media with applications to penetration problems*, Computers and Geotechnics **55** (2014) 117
(cit. on p. 43).

[45] F. Irzal et al.,
*A large deformation formulation for fluid flow in a progressively fracturing porous material*,
Computer Methods in Applied Mechanics and Engineering **256** (2013) 29 (cit. on p. 43).

[46] S. Willett, C. Beaumont and P. Fullsack,
*Mechanical model for the tectonics of doubly vergent compressional orogens*,
Geology **21** (1993) 371 (cit. on p. 44).

[47] D. Peric and A. J. L. Crook,
*Computational strategies for predictive geology with reference to salt tectonics*,
Computer Methods in Applied Mechanics and Engineering **193** (2004) 5195 (cit. on p. 44).

[48] S. Gradmann and C. Beaumont,
*Coupled fluid flow and sediment deformation in margin-scale salt-tectonic systems: 2. Layered sediment models and application to the northwestern Gulf of Mexico*,
Tectonics **31** (2012) TC4011 (cit. on p. 44).

[49] S. Gradmann and C. Beaumont, *Numerical modelling study of mechanisms of mid-basin salt canopy evolution and their potential applications to the Northwestern Gulf of Mexico*,
Basin Research (2016) n/a (cit. on p. 44).

[50] B. Hillebrand et al., *Using the level set method in geodynamical modeling of multi-material flows and Earth's free surface*, Solid Earth **5** (2014) 1087 (cit. on p. 44).

[51] M. A. Biot, *General theory of three-dimensional consolidation*,
Journal of applied physics **12** (1941) 155 (cit. on p. 44).

[52] J. A. Sethian, *Evolution, implementation, and application of level set and fast marching methods for advancing fronts*, Journal of Computational Physics **169** (2001) 503
(cit. on pp. 44, 82).

[53] D. McKenzie, *Some remarks on the development of sedimentary basins*,
Earth and Planetary Science Letters **40** (1978) 25 (cit. on p. 48).

[54] M. Wangen, *A finite element formulation in Lagrangian co-ordinates for heat and fluid flow in compacting sedimentary basins*,
International Journal for Numerical and Analytical Methods in Geomechanics **17** (1993) 401
(cit. on p. 49).

[55] P. Frolkovic,
*Immersed interface method for a level set formulation of problems with moving boundaries*,
Proceedings of the Conference Algoritmy (2015) 32 (cit. on p. 50).

[56] J. Grooss and J. S. Hesthaven, *A level set discontinuous Galerkin method for free surface flows*,
Computer Methods in Applied Mechanics and Engineering **195** (2006) 3406 (cit. on p. 50).

[57] M. F. Trujillo, L. Anumolu and D. Ryddner,
*The distortion of the level set gradient under advection*,
Journal of Computational Physics **334** (2017) 81 (cit. on p. 50).

[58] M. Mirzadeh et al., *Parallel level-set methods on adaptive tree-based grids*,
Journal of Computational Physics **322** (2016) 345 (cit. on p. 51).

[59] J. Braun et al., *DOUAR: A new three-dimensional creeping flow numerical model for the solution of geological problems*, Physics of the Earth and Planetary Interiors **171** (2008) 76 (cit. on p. 51).

[60] B. Merriman, J. K. Bence and S. J. Osher, *Motion of multiple junctions: A level set approach*,
Journal of Computational Physics **112** (1994) 334 (cit. on p. 51).

[61] H.-K. Zhao et al., *A variational level set approach to multiphase motion*,
Journal of computational physics **127** (1996) 179 (cit. on p. 51).

[62] A. Brooks and T. Hughes, *Streamline Upwind/Petrov-Galerkin Formulations for Convection Dominated Flows with Particular Emphasis on the Incompressible Navier-Stokes Equations*,
Computer Methods in Applied Mechanics and Engineering **32** (1982) (cit. on p. 56).

[63] S. Valance et al., “A Finite Element Method for Level Sets”,
*ECCOMAS Multidisciplinary Jubilee Symposium*, ed. by P. D. J. Eberhardsteiner et al.,
Computational Methods in Applied Sciences 14, Springer Netherlands, 2009 95 (cit. on p. 56).

[64] J.-L. Guermond, R. Pasquetti and B. Popov,
*Entropy viscosity method for nonlinear conservation laws*,
Journal of Computational Physics **230** (2011) 4248 (cit. on p. 57).

[65] J.-L. Guermond, M. Q. de Luna and T. Thompson,
*An conservative anti-diffusion technique for the level set method*,
Journal of Computational and Applied Mathematics **321** (2017) 448 (cit. on pp. 57, 66, 83).

[66] D.-I. M. Bernauer,
*Motion Planning for the Two-Phase Stefan Problem in Level Set Formulation*, de, (2010),
URL: http://monarch.qucosa.de/landing-page/?tx_dlf[id]=http%3A%2F%2Fmonarch.qucosa.de%2Fapi%2Fqucosa%253A19430%2Fmets (visited on 28/05/2019)
(cit. on p. 58).

[67] W. Bangerth et al.,
*Algorithms and data structures for massively parallel generic adaptive finite element codes*,
ACM Trans. Math. Softw. (2011) (cit. on p. 58).

[68] C. Burstedde, L. C. Wilcox and O. Ghattas,
*p4est : Scalable Algorithms for Parallel Adaptive Mesh Refinement on Forests of Octrees*,
SIAM Journal on Scientific Computing **33** (2011) 1103 (cit. on p. 58).

[69] Jülich Supercomputing Centre,
*JURECA: General-purpose supercomputer at Jülich Supercomputing Centre*,
Journal of large-scale research facilities **2** (2016) (cit. on p. 61).

[70] Jülich Supercomputing Centre,
*JUWELS: Modular Tier-0/1 Supercomputer at the Jülich Supercomputing Centre*,
Journal of large-scale research facilities **5** (2019),
URL: http://dx.doi.org/10.17815/jlsrf-5-171 (cit. on p. 65).

# Appendix

APPENDIX A

# Appendix

This appendix has two parts. An alternative form of the fluid pressure equation, using the variable of the pore pressure (rather than the overpressure as in the main text) is presented. This was not used for the computations presented, but the FEM discretization was carried out and this may be of use in another context. It leads to a slightly different linear system, due to the difference in the weak form's statement.

The second part concerns the values of the physical fields in the domain during the updating of the material properties. If a cell snaps to another material type, what becomes of the pressure at that point in the domain for the next time step? In other words, if we move the material properties through the domain (through the material id of a cell), should not the physical fields (overpressure, temperature) also be advected along? Techniques below were introduced to address this conceptual question. However, this issue can be minimized by higher resolution, and the techniques below did not exhibit a marked influence on the verification problem, so the additional complexity was deemed not worth it for this thesis. As it mixes states from the current and next time step, Crank-Nicolson time-stepping was used in the thesis and seemed sufficient. Again, these materials are presented since they might be of use in another context, though it was not relevant to the computations presented in the main text.

## A.1 Pore pressure weak form

Beginning from Eq. (4.4),

$$\frac{C}{1-\phi}\frac{\partial p}{\partial t} - \nabla \cdot \left(\frac{\mathbf{k}}{\mu}(\nabla p - \rho_f g \hat{n}_z)\right) = \frac{C}{1-\phi}\frac{\partial \sigma_z}{\partial t} \tag{A.1}$$

where $\hat{n}_z$ is the unit vector in vertical direction. The right hand side can be expressed as

$$\frac{\partial \sigma_z}{\partial t} = \frac{\partial}{\partial t}\left(g\int_0^z \rho_b(\phi(z'))\, dz'\right) \tag{A.2}$$

As we assume the addition of bulk material with a known surface porosity, $\phi_0$ at a prescribed sedimentation rate $\omega$, we can express this as,

$$\frac{\partial \sigma_z}{\partial t} = \rho_b|_{\phi_0} g\omega[ML^{-1}T^{-3}] \tag{A.3}$$

We refer to the right hand side of Eq. (A.1) as $Q$ subsequently. To complete the partial differential equation, boundary and initial conditions must be specified. The square domain is split into $\partial\Omega_D$ and $\partial\Omega_N$. $\partial\Omega_D$ is defined as the top surface of the domain, with $\partial\Omega_N$ comprised of the remaining three sides. The boundary conditions are:

$$p = 0 \tag{A.4}$$

on $\partial\Omega_D$ for all $t > 0$, and

$$\frac{\mathbf{k}}{\mu}(\nabla p - \rho_f g \hat{n}_z) \cdot \hat{\mathbf{n}} = 0 \tag{A.5}$$

on $\partial\Omega_N$ for all $t > 0$, where $\hat{\mathbf{n}}$ is the normal vector pointing out.

For the initial condition at time $t = 0$, it is a hydrostatic condition,

$$p = \rho_f g z \tag{A.6}$$

in domain $\Omega$.

First, we discretize the time derivatives with an implicit Euler finite difference approximation.

$$\frac{C}{1-\phi}\left(\frac{p^{n+1}-p^{n}}{\Delta t}\right)-\nabla\cdot\left(\frac{\mathbf{k}}{\mu}\left(\nabla p^{n+1}-\rho_{f}g\hat{n_{z}}\right)\right)=Q^{n+1} \tag{A.7}$$

To convert the strong form of the PDE to the variational formulation, we multiply by an arbitrary member $\varphi$ from an appropriate set of test functions, and integrate over the domain. We seek the discrete solutions in a basis of scalar Lagrange finite elements, $\mathbb{Q}_p$, the set of continuous, piecewise polynomials of degree $p$.

$$\int_{\Omega}\frac{C}{1-\phi}\varphi\left(\frac{p^{n+1}-p^{n}}{\Delta t}\right)-\int_{\Omega}\varphi\nabla\cdot\left(\frac{\mathbf{k}}{\mu}\left(\nabla p^{n+1}-\rho_{f}g\hat{n_{z}}\right)\right)=\int_{\Omega}\varphi Q^{n+1} \tag{A.8}$$

To reduce the continuity requirements on $p$, the second term on the left hand side is integrated by parts. This goes through with the product rule and the divergence theorem. Namely

$$\begin{aligned}-\int_{\Omega}\varphi\nabla\cdot\left(\frac{\mathbf{k}}{\mu}\left(\nabla p^{n+1}-\rho_{f}g\hat{n_{z}}\right)\right)&=\int_{\Omega}\nabla\varphi\cdot\left(\frac{\mathbf{k}}{\mu}\left(\nabla p^{n+1}-\rho_{f}g\hat{n_{z}}\right)\right)\\&-\int_{\partial\Omega}\varphi\hat{n}\cdot\left(\frac{\mathbf{k}}{\mu}\left(\nabla p^{n+1}-\rho_{f}g\hat{n_{z}}\right)\right)\end{aligned} \tag{A.9}$$

Examining the boundary term:

$$\begin{aligned}\int_{\partial\Omega}\varphi\hat{n}\cdot(\frac{\mathbf{k}}{\mu}(\nabla p^{n+1}-\rho_{f}g\hat{n_{z}}))&=\int_{\partial\Omega_{D}}\varphi\hat{n}\cdot(\frac{\mathbf{k}}{\mu}(\nabla p^{n+1}-\rho_{f}g\hat{n_{z}}))\\&+\int_{\partial\Omega_{N}}\varphi\hat{n}\cdot(\frac{\mathbf{k}}{\mu}(\nabla p^{n+1}-\rho_{f}g\hat{n_{z}}))\end{aligned} \tag{A.10}$$

On the Dirichlet boundary, at the top of the domain, the pore pressure is 0. As an essential boundary condition, the test function must come from the space which reflects this condition. Applying the no flow Neumann boundary condition to that part of the the surface integral eliminates it, and we have the weak formulation of the equation.

$$\int_{\Omega}\frac{C}{1-\phi}\varphi(\frac{p^{n+1}-p^{n}}{\Delta t})+\int_{\Omega}\nabla\varphi\cdot(\frac{\mathbf{k}}{\mu}(\nabla p^{n+1}-\rho_{f}g\hat{n_{z}}))=\int_{\Omega}\varphi Q^{n+1} \tag{A.11}$$

To form an approximation to the solution variable, we want to express the $p$ as a sum of contributions from the discrete basis functions with the expansion coefficients to be determined,

$$p = \sum_j U_j \varphi_j \tag{A.12}$$

The shape functions, $\varphi_j$, are elements of the same space as the test functions. This characterizes the Bubnov-Galerkin discretization method. This leads to the weak form of the discrete problem, where we require for all $i$:

$$\int_\Omega \frac{C}{1-\phi}\varphi_i(\frac{(U_j\varphi_j)^{n+1} - (U_j\varphi_j)^n}{\Delta t}) + \int_\Omega \nabla\varphi_i \cdot (\frac{\mathbf{k}}{\mu}(\nabla(U_j\varphi_j)^{n+1} - \rho_f g\hat{n_z})) = \int_\Omega \varphi_i Q^{n+1} \tag{A.13}$$

Taking just the first term:

$$\int_\Omega \frac{C}{1-\phi}\varphi_i\left(\frac{(U_j\varphi_j)^{n+1} - (U_j\varphi_j)^n}{\Delta t}\right) = \frac{1}{\Delta t}\left(\int_\Omega \frac{C}{1-\phi}\varphi_i\varphi_j U_j^{n+1} - \int_\Omega \frac{C}{1-\phi}\varphi_i\varphi_j U_j^n\right) \tag{A.14}$$

The second term of Eq. (A.13):

$$\int_\Omega \nabla\varphi_i \cdot (\frac{\mathbf{k}}{\mu}(\nabla(U_j\varphi_j)^{n+1} - \rho_f g\hat{n_z})) = \int_\Omega \frac{\mathbf{k}}{\mu}(\nabla\varphi_i \cdot \nabla\varphi_j)U_j^{n+1} - \int_\Omega \frac{\mathbf{k}}{\mu}(\nabla\varphi_i \cdot \rho_f g\hat{n_z})) \tag{A.15}$$

Multiplying Eq. (A.13) through by the time step, and expressing the weighted mass matrix as $M' = \int_\Omega \frac{C}{1-\phi}\varphi_i\varphi_j$, the stiffness matrix (with the spatially dependent coefficient $\frac{\mathbf{k}}{\mu}$) as $A' = \int_\Omega \frac{\mathbf{k}}{\mu}\nabla\varphi_i\nabla\varphi_j$ and the source term vector $F^{n+1} = \int_\Omega \varphi_i Q^{n+1}$ leads to the linear system,

$$M'U^{n+1} - M'U^n + \Delta t A'U^{n+1} = \Delta t(F^{n+1} + \int_\Omega \frac{\mathbf{k}}{\mu}(\nabla\varphi_i \cdot \rho_f g\hat{n_z})) \tag{A.16}$$

and finally grouping the known quantities on the right hand side leads to the linear system of equations to be solved at each time step,

$$(M' + \Delta t A')U^{n+1} = M'U^n + \Delta t(F^{n+1} + \int_\Omega \frac{\mathbf{k}}{\mu}(\nabla\varphi_i \cdot \rho_f g\hat{n_z})) \tag{A.17}$$

These are the matrices and vectors that form the linear system and then, since $(M' + \Delta t A')$ is symmetric and positive definite, the preconditioned conjugate gradient method is used to iteratively solve the system.

## A.2 Identification of location in model domain across time step

As Eq. A.17 shows, the solution vector $U^n$ is required for the solution at the new time step $U^{n+1}$. Since we shift the model domain by the depth corresponding to the amount of sediment added, $\omega\Delta t$, the coefficient vector $U^n$ needs to be updated to reflect this shift, i.e. the coefficient weights refer to different basis functions in the computational grid. We describe two strategies to take this into account.

(1) Rather than attempting to manipulate the vector $U^n$ globally, during the assembly of the linear system in Eq. A.17, the right hand side will be built completely, as opposed to just using $U^n$ as existing from the previous time step's solve output.

The shifted $U^n$, referred to as $U^{n*}$ is created through the following steps. During the assembly phase at a time step $n$ (i.e. the creation of the linear system to solve for the solution at $n + 1$), as $M'$ and $A'$are being created through the process of computing the local contributions from one cell, and then copying to the global matrix, $U^{n*}$ will also be constructed, by using the values for the degrees of freedom from the above neighboring cell. By enforcing that in a time step exactly one cell width of material depth is added, using the solution values of the cell above reflects the shift in the model domain through the computational grid. This leads to the following linear system.

$$(M' + \Delta t A')U^{n+1} = M'U^{n*} + \Delta t(F^{n+1} + \int_{\Omega} \frac{\mathbf{k}}{\mu}(\nabla\varphi_i \cdot \rho_f g \hat{n_z})) \qquad (A.18)$$

(2) The second approach seeks to avoid the concreteness of the low-level manipulations of degrees of freedom by manipulating the vector globally. An operator approach is now described, the purpose of which is to "advance" the previous solution variables onto the computational grid of the current time step. This is done by a simple advection, in a known direction, vertical, by a known quantity, the value of the speed function, F, at that point, $\hat{w} = (0, 0, -F) = -F * \hat{n}$, where $\hat{n}$ is the unit vector

pointing up in the vertical direction.

$$\frac{\partial U}{\partial t} + \hat{w} \cdot \nabla U = 0 \tag{A.19}$$

and with an explicit Euler discretization for the time derivative,

$$\frac{U^{n+1} - U^n}{\Delta t} - F\hat{n} \cdot \nabla U^n = 0 \tag{A.20}$$

Multiplying from the left with a test function, integrating

$$\int_\Omega \varphi_i \frac{U^{n+1} - U^n}{\Delta t} - \int_\Omega \varphi_i F\hat{n} \cdot \nabla U^n = 0 \tag{A.21}$$

and expanding U in terms of the finite element basis, $U = \sum_j \chi_j \varphi_j$.

$$\int_\Omega \varphi_i (\chi_j \varphi_j)^{n+1} - \varphi_i (\chi_j \varphi_j)^n = \int_\Omega \varphi_i \Delta t F\hat{n} \cdot \nabla(\chi_j \varphi_j)^n \tag{A.22}$$

Rearranging, and using the mass matrix, $M = \int_\Omega \varphi_i \varphi_j$,

$$M\chi_j^{n+1} - M\chi_j^n = \int_\Omega \varphi_i \Delta t F(\hat{n} \cdot \nabla\varphi_j)\chi_j^n \tag{A.23}$$

Letting $B = \int_\Omega \varphi_i \Delta t F(\hat{n} \cdot \nabla \varphi_j)$

$$M\chi_j^{n+1} = (M + B)\chi_j^n \tag{A.24}$$

The last step is to solve this symmetric and positive definite system with CG to arrive at the updated vector $U^{n+1} = \sum_j \chi_j^{n+1} \varphi_j$.

This updated vector is used, as explained in the previous section, in the formation of the right hand side of a linear system like Eq. (4.35). So there are not consecutive time integrations of this advection equation, which would require stabilization of some kind.

A third approach has been been considered to elide this complexity. Namely, we use a Crank-Nicolson time stepping which mixes from the current and last solution vectors.

It is noted that in the results section 4.5, the Crank-Nicolson time-stepping was used.