Instituto Alberto Luiz Coimbra de
Pós-Graduação e Pesquisa de Engenharia

COPPE UFRJ

STRATIGRAPHIC SEDIMENTARY INVERSION USING PATHS IN GRAPHS

Alexandre Simões Raymond

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientador: Franklin de Lima Marquezino

Rio de Janeiro
Março de 2017

STRATIGRAPHIC SEDIMENTARY INVERSION USING PATHS IN GRAPHS

Alexandre Simões Raymond

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:

Prof. Franklin de Lima Marquezino, D.Sc.

Prof. Abílio Pereira de Lucena Filho, D.Sc.

Prof. Alexandre Antonio de Oliveira Lopes, D.Sc.

Prof. Raphael Carlos Santos Machado, D.Sc.

RIO DE JANEIRO, RJ – BRASIL
MARÇO DE 2017

Raymond, Alexandre Simões

Stratigraphic Sedimentary Inversion Using Paths in Graphs/Alexandre Simões Raymond. – Rio de Janeiro: UFRJ/COPPE, 2017.

XII, 61 p.: il.; 29,7cm.

Orientador: Franklin de Lima Marquezino

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2017.

Bibliografia: p. 58 – 61.

1. Stratigraphy Inversion. 2. Paths in Graphs. 3. Algorithms. I. Marquezino, Franklin de Lima. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

"An algorithm must be seen to be believed." - Donald Knuth

Acknowledgements

This dissertation marks the end of a long period of an intense transformation. Enrolling into this programme was a very enriching experience, and my perspective regarding science grew from curiosity into profound admiration. Many hardships were endured along the way, battles were fought and here I stand. There enters a student, and now, there exits an aspiring member of the community. I would like to thank the Federal University of Rio de Janeiro, the Systems Engineering and Computer Science programme, the Algorithms and Combinatorics laboratory, all the professors, professionals, and colleagues for providing the conditions to this development.

I would like to thank Prof. Franklin Marquezino for all the invaluable dedication and support throughout these years. He was a firm believer in what I could accomplish, even when I would question myself. There has never been a moment where I could not count with his support and prompt assistance to any query during this journey. I was his apprentice during this time, and I am glad that I made the right choice. I believe that we make a great team together, and I am looking forward to collaborating with him in the future, now as colleagues. I wholeheartedly express my highest appreciation and gratitude for his work.

I would like to thank Prof. Alexandre Lopes for being the first person to believe in my potential in the early stages of my career. We fought many battles together for many years and his support was the groundwork for my professional and academic development. Lopes vouched for my competence against the odds and placed high stakes on it. I could not have reached new heights without his impulse, and therefore I am very grateful for all his support.

I would like to give special thanks to Prof. Abílio Lucena for positively impacting my academic life. My experience as his student at the Linear Programming course has deeply transformed my perspective of education. Prof. Abílio is passionate and sensible about the art of teaching, and together we managed to flip a difficult situation into a strong success case. I will always remember the importance of that act and carry on this learning to my future academic life. Without his support, this study could not have been possible. I hope that this work makes him proud and serves as some kind of retribution.

I would like to thank José Wilson Pinto, Diego Nicodemos, Ana Luísa Carvalho, and Filipe Cabral (sorted in ascending order of length of homework lists), also known as the "Algorithms $\vee$ Combinatorics" study group, for being the best colleagues and friends I could have inside the university. We have constantly saved each other's lives, and I believe that our companionship made the whole challenge of academia worth enduring. My transformation and newfound admiration for the Mathematical Sciences is largely due to our time studying together.

I think that "friendship" fails to describe how precious our relationship is, but I cannot express in other words how much I would like to thank my lifetime partners André Sobral, Wagner Reck, Willian Pessoa, Adriana Buzzacchi, Priscila Faria, and Marcelle Magalhães for holding my back (literally) during hard times and being genuinely happy for my accomplishments. You all held me on your arms after I survived a dangerous surgery, took good care of me, and I hope to always make you proud. Our moments together have shown me that a family need not be bloodbound, and that I am one of the luckiest people alive for having you all around.

I would like to thank all my best friends and family for all the encouragement during these academic years. Naming them is beyond the scope of this work, but their support was precious and much appreciated.

I would like to thank the team at PetroSoft for supporting and contributing to my technical advancement throughout these years.

Lastly, I would like to thank the person that deeply impacted my life and shaped my character 12 years ago. With all my heart, I dedicate my life to her memory.

Love is eternal.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

INVERSÃO SEDIMENTAR ESTRATIGRÁFICA USANDO CAMINHOS EM GRAFOS

Alexandre Simões Raymond

Março/2017

Orientador: Franklin de Lima Marquezino

Programa: Engenharia de Sistemas e Computação

O problema de Inversão Sedimentar Estratigráfica consiste em analisar quantidades de dados sedimentos depositados em determinadas regiões em um espaço deposicional, calculando os volumes totais necessários para que os dados das amostras possam ser igualados em uma simulação deposicional *forward*. Esse é um problema central em geologia e exploração de petróleo. Tentativas de resolver esse problema foram conduzidas na ótica de técnicas de otimização e abordagens de tentativa-e-erro. Nós apresentamos um algoritmo de simulação *forward* deposicional utilizado em indústrias de petróleo, além de uma nova abordagem para algoritmos de inversão baseada em caminhos em grafos. Conjecturamos que este algoritmo gera resultados exatos quando há pelo menos uma solução disponível, ao invés de soluções aproximadas. Nós demonstramos uma execução do algoritmo em detalhes. Resultados são discutidos e futuros desenvolvimentos e extensões deste trabalho são sugeridas.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

STRATIGRAPHIC SEDIMENTARY INVERSION USING PATHS IN GRAPHS

Alexandre Simões Raymond

March/2017

Advisor: Franklin de Lima Marquezino

Department: Systems Engineering and Computer Science

The problem of Stratigraphic Sedimentary Inversion consists of analysing quantities of given sediments deposited in sampled regions of a depositional space and calculating the total volumes that are necessary to match the sampled data in a forward depositional simulation. This is a central problem in geology and oil exploration. Attempts to solve this problem were driven in the optics of optimisation techniques and trial-and-error approaches. We present a forward depositional simulation algorithm used in oil industries and a new counterpart inversion algorithm based on paths in graphs. It is conjectured that this algorithm can yield exact measurements when at least one feasible solution is available, instead of approximated solutions. We present a sample run of the algorithm in details. Results are discussed and future developments and extensions of this work are suggested.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The exploration and transformation of resources have always been one of the foundations for the development of mankind. Technological prowess granted humans the ability and the ever-increasing necessity to probe Earth's sources in all kingdoms of its *Systema Naturæ* [1]. Mineral resources played a central role in human advancement, through the mastery of materials such as iron, steel, coal, fossil fuel, and silicon.

Ergo, the inquisitive nature of humankind coupled with the eagerness for new means of development spurred the research of Earth sciences, in order to unravel the mechanisms after well-defined coherent models of nature. The science of geology ensued this urgency and yielded a deep scrutiny at the ontogeny of our home planet.

Transistors and Turing machines became the ultimate tool to harvest the population of what is arguably the fourth missing kingdom in [1]: *Regnum Informatio*[1]. Our updated understanding of nature linked with a new reality of quantifiable information originated digital representations of our physical world.

Albeit diverse, most digital portrayals of reality are profoundly limited by the boundaries of computability and available processing power. For this reason, our understanding of many natural systems is still narrow as to encompass its full complexity. Notwithstanding that, the contributions are still valid and can extend our available knowledge of the agencies of nature.

This dissertation is presented as the culmination of the studies of a Theoretical Computer Science student collaborating amongst the Geosciences' research and development team of a major oil company in Brazil.

The team was presented with the challenge of constructing a stratigraphic simulator to be applied in survey and exploration efforts. Spanning several years, the ongoing development of this tool enabled many different studies and applications on the improvement of geological analysis.

[1] The Kingdom of Information - free translation.

Drilling wells in deep waters can cost up to 50 million dollars for a single attempt, and the failure rate is significant enough to cause an important loss of investment in case the targeted area was not analysed correctly. The cost of developing multiple computational teams and funding research is, in many cases, lower than the cost of a single missed well. Thus, if research and development can save at least one instance of a deep water perforation from failure, the investment is already paid off.

However, in a competitive field for the world's most valuable resource, technology is well-kept and protected from prying eyes. Hence this work situates itself within a narrow, conceded exclusion zone of industrial secrecy. Some other developments and accomplishments were kept out of this work due to the risk of sensitive intellectual property breaches. Many systems and papers associated with the industry present their accomplishments exclusively in higher-level approaches, either because those systems possess very high complexities or for the very reason of maintaining certain results as easy to demonstrate, but hard to reproduce [2–4].

We present a novel, authorial, detailed, and low-level approach to the problem of stratigraphic inversion, departing from comprehensively different techniques, such as traversing paths in graphs, rather than ordinary optimisation methods. This dissertation does not make strong claims as to the applicability of these methods in current systems, despite providing a very well-defined algorithm that may be reproduced at diligence. It is noteworthy that the algorithms presented are stripped of many implementation details that could vastly extend beyond the scope of this work.

Chapter 2 covers the theoretical background required to understand this dissertation in full. Chapter 3 follows with a concise review of the efforts driven towards stratigraphic simulation and inversion. An algorithm for forward stratigraphic simulation encompassing sediment transfer is presented in Chapter 4, alongside the terminology chosen to represent new concepts in this work. The inversion algorithm is presented in Chapter 5, accompanying a thorough, detailed example run to illustrate the concepts explained. Chapter 6 presents our final considerations and the possibilities of future studies related to the topic.

Chapter 2

Fundamentals

2.1 Geology Background

In order to fully comprehend the geological background beneath the motivations for this study, we present an overview of the current literature. In-depth explanations can be found in the present bibliography. [5]

2.1.1 Sedimentology

The study of sediments and the processes regarding their distribution and deposition is called sedimentology. Several different factors play a key role on the formation of sedimentary structures. Physical, chemical and biological processes are constantly studied and observed by sedimentologists. In this study, we will focus only on the physical aspects of sedimentology.

Sedimentation is the process of accumulation after precipitation of mineral or organic matter on a given surface. Constant accumulation under pressure may lead settled particles to crack and form sedimentary rocks. Those can be divided in three primary types:

Carbonates: Composed of minerals containing the carbonate ion ($CO_3{}^{2-}$), carbonate rocks are sedimentary rocks that can be dissolved or precipitated by groundwater, depending on factors such as pH or temperature. The two main types of Carbonate rocks are limestone and dolostone.

Evaporites: Mineral sediments accumulated by chemical processes and precipitation. Sea salt is an example of an evaporite sedimentary rock. Evaporites are soluble in water, and tend to precipitate in high-concentration, saturated water bodies. The water evaporation leads to a higher concentration of salts in the water body, until saturation is reached and precipitation begins. The Dead Sea is an example of an evaporite depositional environment.

Size	Aggregate	Rock
256mm	Boulder	Boulder conglomerate
64mm	Cobble	Cobble conglomerate
4mm	Pebble	Pebble conglomerate
2mm	Granule	Granule conglomerate
1mm	Very coarse sand	Very coarse sandstone
1/2mm	Coarse sand	Coarse sandstone
1/4mm	Medium sand	Medium sandstone
1/8mm	Fine sand	Fine sandstone
1/16mm	Very fine sand	Very fine sandstone
1/256mm	Silt	Siltstone
$< 1/256$mm	Clay	Claystone

Table 2.1: Clastic grain sizes and their associated rock types.

Clastic: Composed of other minerals and rocks fragmented by weathering or erosion. Clastic sediments are transported by physical processes and accumulated in a low-energy environment. For example, a river flow transporting debris and fragments of rocks will eventually settle in the ocean, if the velocity is low enough. The term siliciclastic is used to denominate noncarbonate clastic rocks that contain silicon in its composition, such as sand, silt or clay(mud). Siliciclastic sediments will be the focus of this study. Clastic sediments can be classified based on grain size, as seen in Tab. 2.1. [6]

2.1.2 Stratigraphy

Over time, sedimentation processes or other geological happenstances (such as volcanism) lead to the formation of rock layers, or strata. The study of rock layers is called stratigraphy. It is important to point the scale difference between sedimentology and stratigraphy. Whilst the former deals with particles and the detailed composition of sediments, the latter deals with sediments in a widely manner, treating whole strata as the atomic units of study. Minor generalisations may apply.

Studies in stratigraphy can be categorised regarding the approach taken to their basic units. Lithostratigraphy interprets and correlates physical aspects of rocks (lithologies) and direct relationships between strata, not considering any chronological properties within deposition or accumulation. A sole approach on lithostratigraphy ignores breaks caused by unconformities, and can lead to mistakes such as correlating two similar lithologies that may have been deposited in very different depositional episodes.

Allostratigraphy is a complement of lithostratigraphy. An allostratigraphical unit is a sedimentary body defined by its bounding unconformities and discontinuities. The sedimentary section is mapped with time significance, often associated with discontinuous surfaces [7].

Biostratigraphy is strongly associated with paleontology, and is also called paleontologic stratigraphy. Its main focus is to correlate fossil evidence within rock layers in outspread locations. Biological material within rocks are useful to correlate and determine their relative age. Biostratigraphic studies played a central role on the development of the geological time scale [5]. There are also narrower fields of study whose specialisations are hinted by prefixes, such as chemostratigraphy, magnetostratigraphy, seismic stratigraphy, and chronostratigraphy [8].

The Principles of Steno

In 1669, Nicolas Steno, a Danish scientist, formulated rules after his observation of rocks and encompassed solids. These observations became the defining basis for the science of stratigraphy [9].

Principle of Superposition: *"At the time when any given stratum was being formed, all the matter resting upon it was fluid, and, therefore, at the time when the lower stratum was being formed, none of the upper strata existed."*

This principle is valid to most stratigraphic analyses. Different strata are stacked with younger ones on top (see Figure 2.1). Some exceptions apply to this rule, such as tectonic faulting or older layers collapsing.

Principle of Original Horizontality: *"(...) strata either perpendicular to the horizon or inclined to it, were at one time parallel to the horizon."* Layers of sediment are deposited horizontally due to gravity (see Figure 2.2). Other forces or processes may lead to further deformations. This principle led to the development of the plate tectonics theory.

Principle of Lateral Continuity: *"Materials forming any stratum were continuous over the surface of the Earth unless some other solid bodies stood in the way."* This principle states that layers of sediment are extended laterally until their continuity is broken by some other acting process, such as erosion or orogenetic movements, for example (see figure 2.3). The lateral extension of a layer is determined by the amount and composition of sediments deposited, along with spatial limitations from the sedimentary basin.

Principle of Cross-Cutting Relationships *"If a body or discontinuity cuts across a stratum, it must have formed after that stratum."* Generally speaking, cross-cutting relationships are those where a body cuts through or penetrates pre-existing rocks, such as fractures, faults, or even volcanic intrusions. Since the previously cited principles state that strata are deposited horizontally and

Figure 2.1: Stratigraphic column with visible different depositional episodes. (PD)

Figure 2.2: Chalk (carbonate) layers deposited horizontally. (PD)

are laterally continuous, any event that proceeds to penetrate or disrupt the original horizontality or lateral continuity must be a later episode regarding the formation of any sedimentary layers.

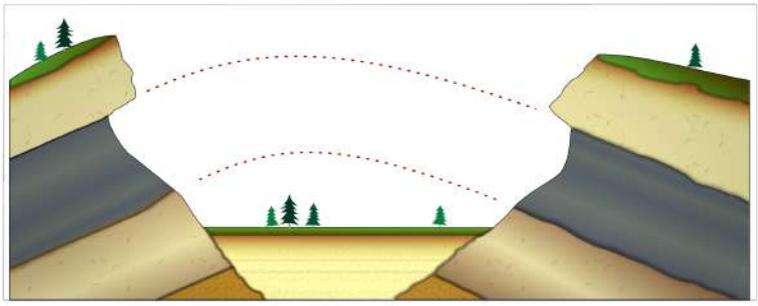

Figure 2.3: Laterally continuous layers separated by an erosional feature. Ⓟ

2.1.3 Sedimentary Transport

The formation of sedimentary layers depends heavily on a supply of depositional matter. Most sediments are transported by air or water currents, and deposited in low-energy environments. A Swedish geographer named Filip Hjulström devised a curve that described the thresholds for erosion and deposition of particles in water [10].

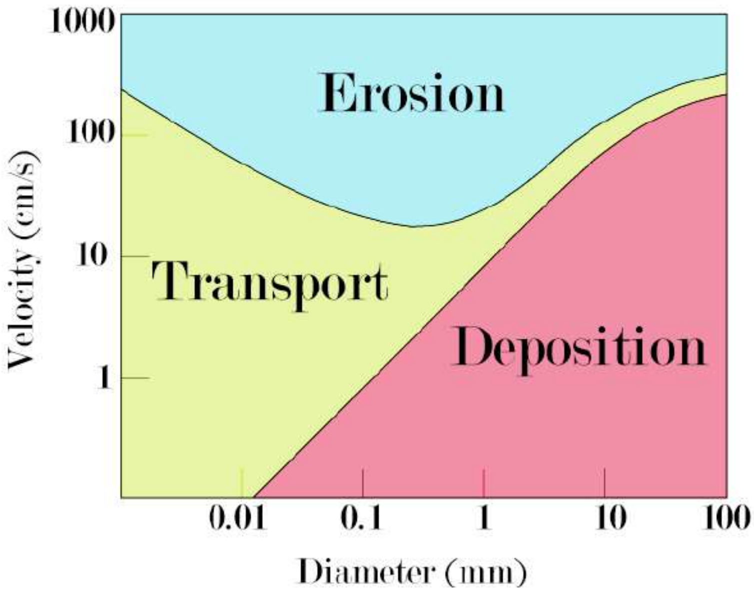

Figure 2.4: The Hjulström curve.

The Hjulström curve (see 2.4) indicates the velocity required to transport or erode particles, depending on their grain size (see table 2.1). If the stream velocity is too low, particles will be deposited. If the stream velocity is too high, previously deposited particles will be removed from the surface and dragged along the current (erosion). Intermediate velocities will transport suspended particles, but not erode the sedimentary surface.

Particle transport is not continuous. Fast, strong currents may deposit gravel, while keeping sands and mud in suspension. If the current weakens and decelerates, a layer of sand will be deposited on top of the gravel. If it stops completely, a layer of mud (silt and/or clay) will be deposited on top of the sand layer.

The physics of fluid dynamics are the ruling mechanism for understanding the behaviour of currents and streams [11]. An useful representation of fluid flow is through the concept of streamlines. A laminar flow is a type of fluid motion where the streamlines are parallel, and do not intersect paths. A stream of rain water running over a gutter is an example of laminar flow. Turbulent flow, on the other hand, has a chaotic movement pattern, with streamlines crossing their paths and forming spirals and whirls (see Figure 2.5).

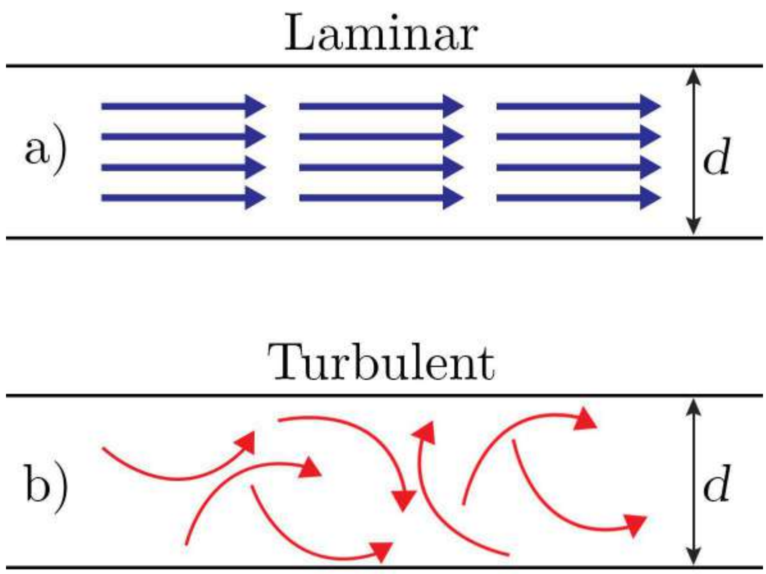

Figure 2.5: Representations of flow types: a) Laminar flow. b) Turbulent flow.

2.1.4 Eustasy

Sea level is not constant in geological history [12]. Most sedimentary structures present in continental bodies were, at one given point of time, within a water body. It is important to point out the sea level variation in a stratigraphic setting, so that depositional environments can be correctly identified and timestamped accordingly.

Local measurements of sea level, such as those regarding a land benchmark, are not precise enough to indicate global tendencies, as a local setting may be affected for extraneous factors, such as tectonism or all sorts of faulting.

Long term, global sea level changes are also called eustatic changes [13]. It is strongly related to global climate change, and is mostly controlled by the ratio of ice to water in the planet. Colder periods indicate a strong glaciation, and the volume of ocean water is diminished by the increase of frozen bodies. Analogously, the transition to hotter periods causes the melting of global icecaps, thus raising the sea level.

Eustatic sea level measurements are also dependent on benchmarks. However, differently from local sea level measurements, eustatic changes take global references, such as the centre of the Earth or an orbiting satellite.

> "It is impossible to determine the size of the variations in eustatic position that occurred during deposition sedimentary section. This is because of the position of the sea varies as a function of eustasy, tectonic behavior and sedimentary fill. It can be demonstrated that if the size of two of the variables are specified then third can be established. This presents a problem since the sizes of all three of these variables are unknown. The solution chosen by most earth scientists is to assume a 'reasonable' size for two of the three variables and solve for the third." [14]

2.2 Graph Theory

Graph Theory is the study of *graphs*, a mathematical structure used to describe and model connectivity between objects.

Definition 2.2.1 (Graph)**.** A graph G is an ordered pair $G = (V, E)$, where V denotes a set of *vertices*, and E denotes a set of *edges*. Every element of E represents a pair of elements in V.

Definition 2.2.2 (Undirected Graph)**.** If there is no special ordering for every pair described by the elements of E, we may denote G as *undirected*. Else, the graph is named a *directed graph* or *digraph*.

Definition 2.2.3 (Simple Graph)**.** If the elements of E are all distinct, we denote G as a *simple graph*. Else, if E has repeated edges, the graph is named a *multigraph*.

In this dissertation, we may assume *graph* as a *simple undirected graph* for convenience.

Definition 2.2.4 (Chebyshev Distance)**.** Chebyshev distance is a metric defined on a vector space where the distance between two vectors is the greatest of their differences along any coordinate dimension. [15] Let $M_1 = (A_1, d_1)$ and $M_2 =$

(A_2, d_2) be metric spaces. Let $A_1 \times A_2$ be the cartesian product of A_1 and A_2. The Chebyshev distance on $A_1 \times A_2$ is defined as

$$d_\infty(x, y) := \max\{d_1(x_1, y_1), d_2(x_2, y_2)\} \tag{2.1}$$

where $x = (x_1, x_2), y = (y_1, y_2) \in A_1 \times A_2$.

Definition 2.2.5 (Moore Neighbourhood)**.** The Moore neighbourhood of a point is defined as the set of points at a Chebyshev distance of 1. [16]

We proceed to construct a graph considering grid cells as vertices. The neighbourhood of any given vertex v is given by the Moore neighbourhood of v [17]. The resulting graph is a King's Graph [18].

Definition 2.2.6 (King's Graph)**.** The $m \times n$ King's Graph is a graph with mn vertices in which each vertex represents a square in an $m \times n$ chessboard, and each edge corresponds to a legal move by a king.

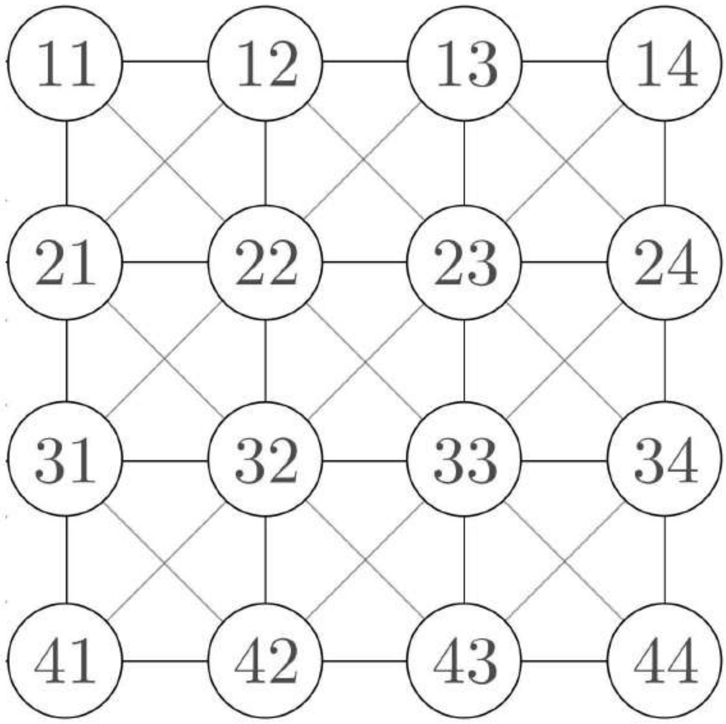

Figure 2.6: King's Graph with $m = 4$ and $n = 4$.

Theorem 2.2.1. *Every (m, n)-King's Graph with $m \geq 2$ and $n \geq 2$ is Hamiltonian and biconnected.*

Proof. Let G be an (m, n)-King's Graph. We can construct a cycle using the following procedure. G is Hamiltonian if and only if there exists a Hamiltonian cycle in G. It is trivial to find Hamiltonian cycles for $m \leq 3$ and $n \leq 3$. Suppose m is even. We can construct a cycle using the following procedure

1. Start in vertex $(1, 1)$ and move $n - 1$ steps right to vertex $(1, n)$.
2. Move $m - 1$ steps downward to vertex (m, n).
3. Move $n - 1$ steps left to vertex $(m, 1)$.
4. Move one step up to $(m - 1, 1)$ and move $n - 2$ steps right to $(m - 1, n - 1)$.
5. Move one step up to $(m - 2, n - 1)$ and move $n - 2$ steps left to $(m - 2, 1)$.
6. Since m is even, the two last steps can be repeated $k = m/2$ times, returning to vertex $(1, 1)$, closing the Hamiltonian cycle.

Suppose m is odd. We can construct a cycle using the following procedure.

1. Start in vertex $(1, 1)$ and move $n - 1$ steps right to vertex $(1, n)$.
2. Move $m - 1$ steps downward to vertex (m, n).
3. Move $n - 1$ steps left to vertex $(m, 1)$.
4. Move one step up to $(m - 1, 1)$ and move $n - 2$ steps right to $(m - 1, n - 1)$.
5. Move one step up to $(m - 2, n - 1)$ and move $n - 2$ steps left to $(m - 2, 1)$.
6. Repeat steps 4 and 5 until the vertex $(3, n - 1)$ is reached.
7. Move one step up to $(2, n - 1)$ and move one step to the south-west diagonal $(3, n - 2)$. Repeat this step until $(3, 1)$ is reached.
8. Move two steps up back to $(1, 1)$, closing the Hamiltonian cycle.

We have proved that G is Hamiltonian. As proven by [19], all Hamiltonian graphs are biconnected. Thus, G is biconnected. □

2.3 Forward and Backward Simulations

In general terms, a simulation can be regarded as the replication or reproduction of a process through a model of reality or a hypothetical model. The quality of the simulation is directly related to the adequacy, correctness and precision of the proposed model.

The very concept of computer simulation is inherently and conceptually attached to the notion of automata. A simulation can be regarded as a relationship between

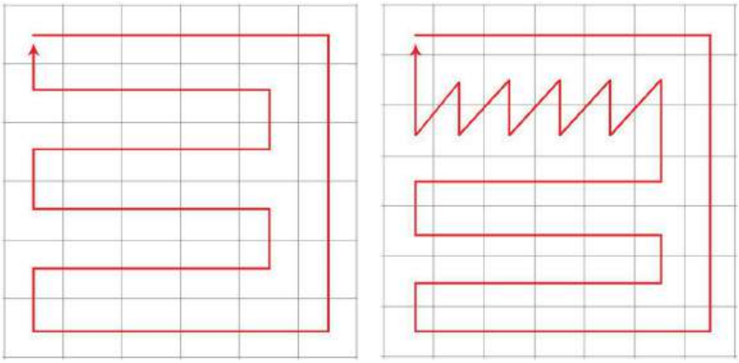

Figure 2.7: Hamiltonian cycles with even and odd m values.

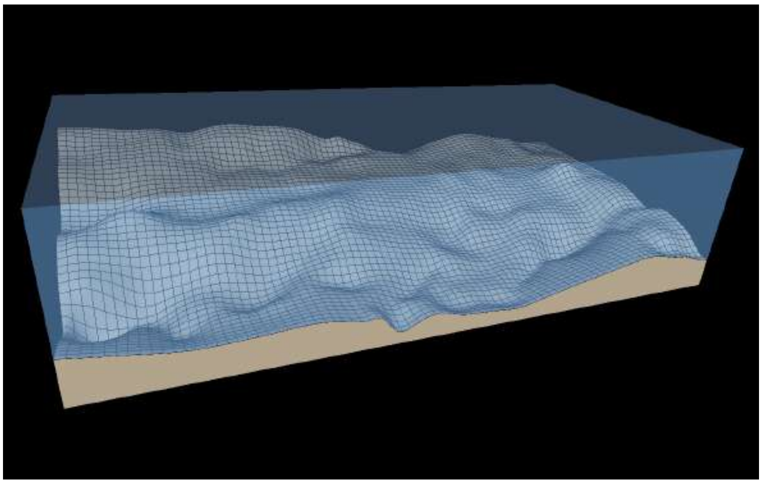

Figure 2.8: 3D depositional surface with $m = 80$ and $n = 50$.

states a and a' of a system, where a represents an input state and a' represents the state after the transition from the input space to the result state.

A more formal definition of types of simulation by the means of automata theory can be found in [20]. We shall attempt to adapt, alter and simplify some of those definitions in order to suit the scope of this work in its competence.

The relationship of states in a forward simulation is similar to a *homomorphism* in automata theory.

The existence of a forward simulation from state A to state B is defined as $A \leq_F B$.

Conceptually, backward simulations provide a way to a start state that would generate the current end state from a forward simulation. Lynch et al. defines:

> "In many respects, backward simulations are the dual of forward simulations. Whereas a forward simulation requires that *some* state in the image of each start state should be a start state, a backward simulation requires that *all* states in the image of a start state be start states. Also, a forward simulation requires that forward steps in the source au-

> tomaton can be simulated from related states in the target automaton, whereas the corresponding condition for a backward simulation requires that backward steps can be simulated. (...) From any given state, all the possible histories are finite executions, whereas the possible futures can be infinite." [20]

The existence of a backward simulation from state A to state B is defined as $A \leq_B B$.

We can extend these concepts to cover the notion of hybrid simulations - which combine in a single relation both a forward and a backward simulation. We are interested in the concept of backward-forward simulations.

We say that there is a backward-forward simulation between states B and A if $B \leq_B A$ and $A \leq_F B$.

In Chapters 4 and 5 we will present forward and backward simulation algorithms, respectively. Section 5.2 contains a backward-forward simulation—defined as such to conjecture that the backward method is capable of finding an adequate input space that generates its original state.

Chapter 3

A Review of Stratigraphic Simulation Methods

A large number of predictive science applications rely on simulations, especially computational simulations. A concise simulation model must be able to reproduce studied phenomena more efficiently than clear human observation and deduction. As a very large number of factors and parameters operate on real-world systems, most simulations are simplifications aiming to obtain predictive data under an expected error threshold.

One may reproduce a scenario in an artificial system (i.e., a simulation) and compare results with sampled data from the original system. If a clear relationship between simulated and real data is to be found, then evidence suggests that the simulation is accurate to the extent of its own produced information.

Forward simulations will be used to model events that happen forwardly in time. We define input parameters and conditions for the simulator, which outputs an outcome of such calculations. These data can represent a prediction over the initial parameters.

Simulation is an important tool to study and address geological hypotheses. The understanding of parameters and factors that lead to a certain scenario is crucial for effective analyses of real world cases. The problem of sedimentary transport has been addressed in several studies. [21] [22] [23] [24] Computational and numerical simulations are widely used in geological research, yielding results in several fields, such as geochemistry or sedimentology.

3.1 Forward Stratigraphic Simulation

Geological analysis and resource prospection in new regions often require an in-depth knowledge of the ancient natural processes that took place millions of years ago. The current setting of rock layers at large scales is studied through stratigraphic phenomena.

Stratigraphic simulation software came to existence in the late 1980s, with Stanford University's SEDSIM [25]. The interest in computer-assisted simulation techniques grew as geologists acknowledged the capabilities of simulation models being developed along the years. [26]

Although unanimity among geologists regarding the effectiveness of computerised stratigraphic simulation tools was hard to achieve in the early years, the ever-increasing adoption of such methods urges the field for further developments and improvements.

Most models for stratigraphic simulation work in a similar way, despite some minor differences in approach and internal methods. [27] In general terms, users input:

- The "basement" surface (in 2D or 3D representations)
- Values for sea levels
- Sediment supply parameters
- Flow configurations, such as border velocities and sediment transport rules

Optional inputs can also be:

- Paleobathymetries[1] as depositional limits
- Real-world well data for comparison
- Eustatic curves for dynamic sea level adjustment
- Wave base levels
- Erosion data

Combined with the rules of the forward stratigraphy simulator, these inputs produce an output state that contain varied outputs, depending on how each simulator is structured. It is very common to encounter a distribution of sediments, variations in eustasy, sediment compaction, tectonic and orogenetic movements, subsidence etc.

[1]Paleobathymetry is the study of past underwater depths.

3.2 Stratigraphic Inversion

In 1987, Burton et al. reasoned [14] that stratigraphic inversion was impossible to be carried out due to the complexity of the many variables involved in the process, such as eustasy, tectonics and sediment transport. The main argument was the non-uniqueness of possible states, which would render such a process impossible to predict even with a well-defined forward model.

Two years later, Carron [28] and Fabre et al. [29] presented their first models for stratigraphic inversion of seismic data. In what seemed to be the consideration of [14] as a challenge for the community, more and more efforts to prove the actual possibilities within the inversion and backward modelling of stratigraphic processes.

The year of 1991 introduced Lessenger et al. [30], a more complete model of stratigraphic inversion. Lessenger encompassed several variables in their simplified forward stratigraphy simulator, and provided the inversion capabilities through GLI[2]. Albeit a preliminary work, the extension of that study culminated in [32], a definitive refutal for the claims in [14]. The claim that stratigraphic inversion is feasible was accompanied by a more sophisticated method of forward simulation, contemplating temporal and spatial distributions of stratigraphic surfaces and sedimentation patterns atop synthetic basins. Inversion was accomplished by numerical optimisation techniques, such as gradient descent methods.

Lessenger et al. established the theoretical possibility of inversion, but her practical methods were not at par with the demanding requirements of the exploration industry. It is even suggested that a better algorithm for stratigraphic inversion should be tested in their model. Conjointly, the expediency of stratigraphic inversion drove the interest of industry and conferences in order to build accurate stratigraphic simulators.

A solid attempt to provide such an algorithm surfaced in 1998, with Bornholdt et al. [33]. The authors created a *quasi-backward* optimisation model for automatising the input parameters in a forward stratigraphic model using genetic algorithms. Granted the dissimilarities in approaches, our dissertation is very similar to this work scope-wise.

The methods applied in Bornholdt et al. [33] are approximative heuristics, hence the terminology *quasi-backward*. Fitness functions attempt to converge the variables in the input space to the best possible distribution of lithologies alongside the basin. In many cases, approximative solutions are sufficient for practical results.

It is worth recognising that most of the input data provided for real-world simulations are fairly noisy, imprecise, and incomplete. Geochronological information

[2] Generalised Linear Inversion [31].

extracted from samples have margins of error that span values beyond 1 Ma[3]. Simulating time steps that are orders of magnitude smaller than the margins of error of their input data is considerably challenging, yet acceptable for evaluating broader hypotheses and concepts.

Structuring and formalising the problem of stratigraphic inversion came with a very influential paper by Lessenger et al. [34]. The definition of an inverse model was established as:

1. Selecting a stratigraphic forward model;
2. Defining invertible mathematical abstractions that describe real stratigraphic processes;
3. Transcribing the output data of the forward model into an optimisation-friendly format;
4. Selecting an appropriate optimisation algorithm;
5. Binding previous steps into a stratigraphic inversion model.

Lessenger's work was also able to provide accurate stratigraphic predictions on real-world data, which can be presented as practical evidence of the claims made in [32].

Significant developments on the subject can be seen in Sharma [35]. The approach of using different parameterisations for different time intervals provided an extra insight on the coupling of different parameters. Source location and transport coefficients indicated significant coupling in [36].

A more recent study by Falivene et al. [2] incorporated a full three-dimensional stratigraphic inversion and further alterations to the optimisation models used in previous works, such as Neighbourhood Algorithms [37].

It is critical to point out that the field of geological simulation is deeply covered in industrial secrecy. Many of the more modern approaches to geological simulation are protected under intellectual property of industries in the fields of energy and resource exploration, mainly.

Moreover, it follows that some of the studies don't disclose inner mechanisms of their forward and backward models in sufficient detail for reproducing results; alternatively, they refer to those topics in a higher level of abstraction, either to preserve secrecy or to address purely the geological countenances—which might be the relevant subject for their intended audience.

In Chapters 4 and 5 we are going to present a different abstraction for the problem of stratigraphic simulation and sediment transfer. We choose to define our

[3]One million years.

forward model in terms of graph theory, and we shall provide an inversion algorithm that aims to provide exact inversion values for sediment supply totals.

Chapter 4

An Algorithm for Forward Stratigraphic Simulation

We present an algorithm used in the oil and gas industry for the simulation of siliciclastic sediment transport in stratigraphic scales, originally devised by Lopes et al. [38] [39] and adapted by the author. As stated, stratigraphic analyses are held in lower resolution settings, with large models and discrete units that may span over several hundred meters.

This algorithm aims to simulate the distribution of sediments in a basin over a specific period of time. The process can be repeated multiple times for each new period, or *time step*. It is important to point out that whilst the greater process is associated with time, each time step is considered a discrete unit. There can be steps of any granularity, but time is not counted *inside* the steps—it only progresses discretely.

Many different sediments can be deposited in the same time step. Geologically, sediment flow is commonly mixed and composed of different lithologies. However, according to Figure 2.4, larger grains require more energy for transportation, therefore, deposit earlier than smaller grains. A simplified way to represent this property is to run the deposition algorithm once for every sediment, in decreasing order of grain size. Since multiple-sediment runs are essentially extra iterations of the algorithm, we can assume, without loss of generality, that the algorithm for one sediment described below is valid for every number of sediments, by running it one time for every different sediment.

4.1 Algorithm Input

4.1.1 Vertices

The algorithm operates on a 3D depositional surface, given as input. Bathymetric[1] data is used to construct the surface. The surface is then partitioned in $|V| = m \times n$ vertices disposed in a regular grid. Real simulation instances may have over 250 thousand vertices.

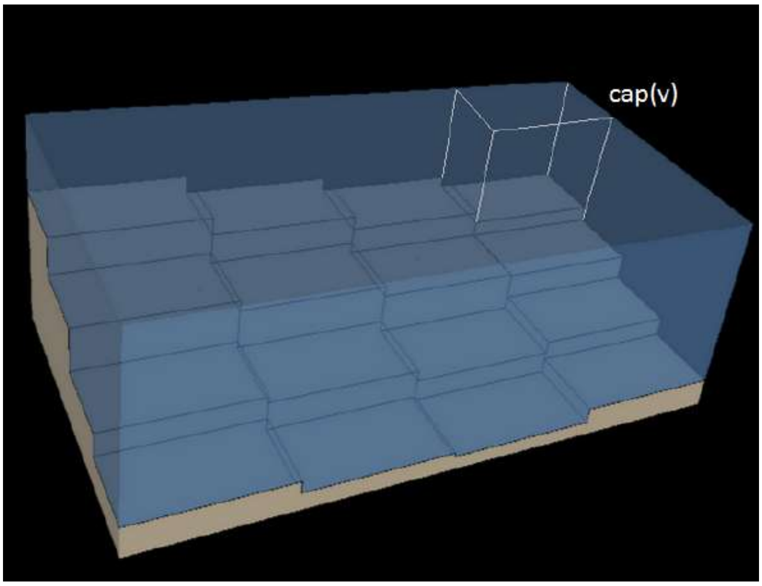

Figure 4.1: Bathymetry and the associated capacity of v.

A 2-dimensional grid $V_{m \times n}$ of vertices represents the sedimentary cells $v_{ij} \in V$. We denote $MAX(V)$ as the matrix of maximum capacities associated with V. $DEP(V)$ is the matrix of deposit amounts in every vertex v_{ij}.

Definition 4.1.1 (Vertex Capacity)**.** The matrix of vertex capacities is defined as $CAP(V) = MAX(V) - DEP(V)$, where $cap(v_{ij}) \geq 0$, $\forall i \leq m$ and $\forall j \leq n$.

The neighbourhood model for the vertices is the Moore neighbourhood, as discussed in Section 2.2.

4.1.2 Flow Configuration

All deposited matter follows a given flow throughout the grid, visiting cells and depositing amounts of sediment over them. Different modelling approaches in fluid dynamics can be used to define the physics of flow for effective and reliable simulations. While this is an interesting topic that has provided several theses and dissertations on the subject, it is not the focus of this work. We will consider all flow calculations and mechanisms as a valid input to define our streamlines.

[1]Bathymetry is the underwater equivalent to hypsometry or topography.

Definition 4.1.2 (Path)**.** A path is a finite sequence of edges between two vertices u and v without including any vertex more than once.

Definition 4.1.3 (Streamline)**.** A streamline $s_i = \{v_1, v_2, \ldots, v_n\} \in S$ is a path by which sediments will be transported and deposited. Different streamlines may share vertices and edges, but their paths cannot be crossed in the current topological representation of the graph. Every streamline s_i has an amount of sediments $sed(s_i) \geq 0$ associated to it.

Definition 4.1.4 (Path Step)**.** The k-th vertex present in the path of s_i will be denoted as s_i–k.

4.2 Deposition

Given the capacities of all vertices and a set S of streamlines with their associated volumes, we can now proceed to the transport of sediments in the graph. Let s_{max} be the longest streamline available.

We describe, in general terms, the forward deposition algorithm below.

Algorithm 4.1 Forward deposition algorithm

```
for k in 1 to |s_max| do
    for every s_i ∈ S do
        if k < |s_i| and dep(s_i) > 0 then
            s_i visits the k-th vertex in the path.
            Transfers sediment from sed(s_i) to the vertex in s_i–k.
            deposit := min(sed(s_i), cap(s_i–k)
            sed(s_i) := sed(s_i) − deposit
            dep(s_i–k) := dep(s_i–k) + deposit
            if sed(s_i) > 0 then
                SEDIMENT SPREAD(s_i–k, sed(s_i))
            end if
        end if
    end for
end for
```

Essentially, the algorithm visits the k-th vertex of every streamline, proceeds to deposit the maximum amount possible (limited by the capacity of the vertex) and then proceeds to the sediment spread procedure. After the sediment spread is finished, it visits the next streamline, until all streamlines were visited at the k-th round. It continues until all paths are fully visited or if there is no sediment left in streamlines. The sediment spread procedure will be discussed in depth in 4.3.

4.3 Sediment Spread

Since this forward modelling aims to reproduce a physical phenomenon, some approximations are made in order to efficiently reproduce some of the aspects and properties of sediment deposition and transport. Sediment has a tendency to spread along its deposition site, and it is important to represent this in the simulation.

One way to do so is to make small depositions around the neighbouring cells of the grid. The proportion of deposited sediments diminishes along with the distance of the deposition centre in an exponential manner.

Definition 4.3.1 (Spread Radius)**.** The spread radius r is the maximum Chebyshev distance (2.2.4) where sediment spread takes place.

Let v be the deposition centre and $N(v, r)$ the set of all neighbouring vertices at a distance $d < r$. We define the maximum amount $dep(u)$ spread to a given vertex $u \in N(v, r)$ as

$$dep(u) = cap(u) \cdot f^{-d}, \tag{4.1}$$

where f is a constant used as the linear factor for the exponential decrease. For example, if $f = 4$ and $r = 3$, all vertices at distance 1 from v would be able to receive 1/4 of its current capacity, as 1/16 and 1/64 would be the limits for $d = 2$ and $d = 3$, respectively. A higher value of f would deposit smaller amounts in neighbouring vertices, which allows for the desired calibration of spread behaviour.

However, sediment spread does not happen simultaneously for all vertices. All the spread is done consecutively. The visiting order for sediment spread is well-defined and can be given as input. This order considers in what sequence should the immediate ($d = 1$) Moore neighbourhood be visited. Larger neighbourhoods can be constructed using the procedure illustrated by Figures 4.2 to 4.10.

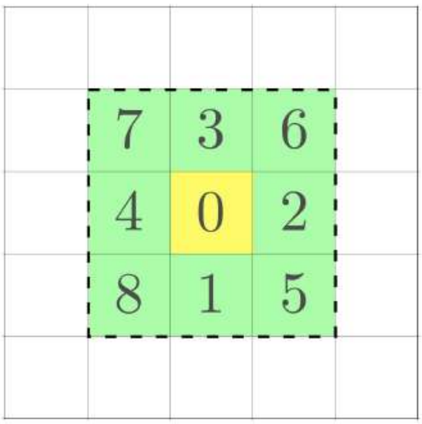

Figure 4.2: $N(v, 2)$ immediate Moore neighbourhood, defined as input up to $d = 1$.

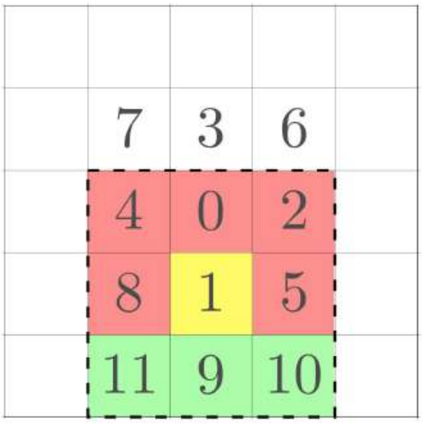

Figure 4.3: First neighbour is visited. All adjacent cells are added to N.

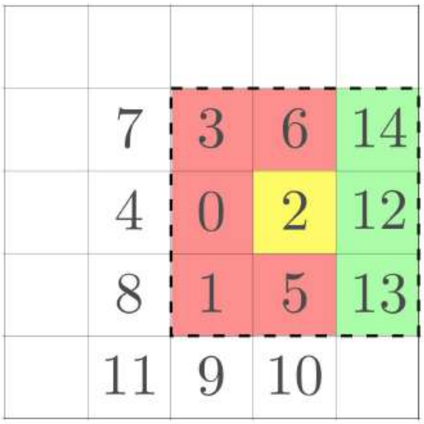

Figure 4.4: Second neighbour is visited. All adjacent cells are added to N.

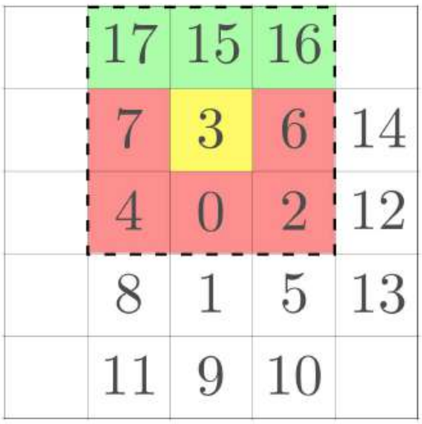

Figure 4.5: Third neighbour is visited. All adjacent cells are added to N.

It is important to notice that any vertex $u \in N(v, r)$ can only receive a maximum amount of $dep(u)$. However, as $dep(u)$ is a multiplication by a fraction of the available space (exponentially related to the Chebyshev distance), for all $d > 0$, no spread vertex will be completely filled by sediment spread. Thus, a vertex can only

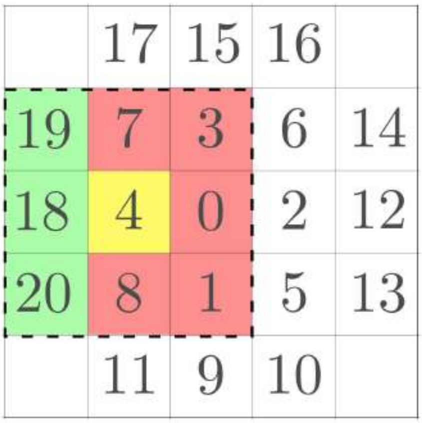

Figure 4.6: Fourth neighbour is visited. All adjacent cells are added to N.

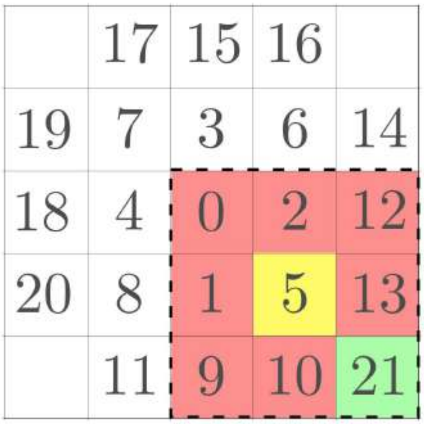

Figure 4.7: Fifth neighbour is visited. All adjacent cells are added to N.

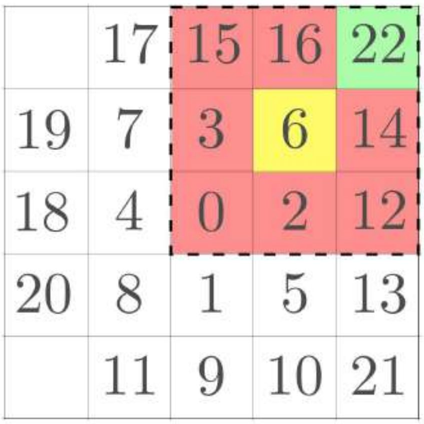

Figure 4.8: Sixth neighbour is visited. All adjacent cells are added to N.

be topped by direct deposition.

Figure 4.9: Seventh neighbour is visited. All adjacent cells are added to N.

Figure 4.10: The process repeats itself until the neighbourhood $N(v, 2)$ is totally filled.

4.4 Considerations

As in most large-scale, low-resolution geological simulations, it is important to notice that this algorithm is an extreme simplification of the real process, which is inherently complex and still a target of many researchers and geologists. That said, one can observe behaviours in specific cases that this algorithm will not correspond to the physical reality of geological settings.

For instance, this forward approach disregards phenomena such as the alteration of flow velocity throughout streamlines, which could cause sediment bypass (not depositing in available vertices due to high sediment velocity) or erosion (actually removing previously deposited sediment due to extreme sediment velocity). Also, other odd behaviours might be seen, such as sediment from lower regions being spread upwards, just because the vicinity criteria is not bounded by altitude.

One must point that, however precise the sedimentation simulator might be, it can only be as good as its input data. Most sediment dating is carried through the analysis of fossils, and those can have a margin of error as large as one million years. Considering that most simulations are carried in steps smaller than the margin of error of the input (hundred-thousand-year time steps, for example), it becomes evident that even the most sophisticated algorithm would not produce an exact geological representation of a given setting, mostly because the input data is rarely precise enough.

That said, the simulator can still provide general and low-resolution results that are satisfying to test or verify broad hypotheses for simulators. Users will work to obtain tendencies and rough estimates, which can already debunk flawed assumptions, such as the impossibility of a certain sediment coming through a path, or discovering a possible alternate source for a sediment.

Chapter 5

A New Backward Simulation Algorithm

In summary, the greater steps of the *forward* algorithm are listed as:

1. Graph construction.
2. Definition of flow paths.
3. Establishment of sediment supply volumes.
4. Sediment transport and deposition.

These steps will simulate the deposition of sediments along a grid of cells of different capacities. Streamlines will be traversed and spread proportional capacities to neighbouring vertices. A geologist might use the results from the forward algorithm to test geological hypotheses. This sort of test is normally performed through the comparison of expected results in vertices that contain real data extracted from wells. We shall denote any such vertex as a *restricted vertex*.

However, adjusting initial parameters manually until restrictions are met is a very thorough task. The forward algorithm is complex and not intuitive enough for a user to make appropriate guesses in reasonable time. The average user may take several days to manually find satisfiable initial parameters that meet the restrictions (yet still with significant error rates).

This chapter aims to propose an algorithm that will perform a backward traversal of the grid, starting from the restrictions. Vertices alongside the paths will add up to the total demand of sediment. The overall demand for each streamline will be obtained so that the forward algorithm, when set with the parameters found by the backward algorithm, will match all restrictions in just one run (with no error). This can greatly speed up analyses from geologists, making day-long manual trial-and-error attempts become automated executions that takes minutes to perform.

We will consider the same abstractions and structures previously defined in Chapter 4. The original implementation of the forward algorithm was not organised with those abstractions. They were devised for the purpose of this study by the author, and shall bear no distinctions from the original concept. Thereupon, the whole concept of the algorithm arose from the author's point of view whilst observing the original problem structured in the means of graph theory.

5.1 Core Concepts

The basis of the inversion algorithm revolves on the concept of *visitors*.

A Visiting Step represents the unitary and indivisible step of the deposition process.

Definition 5.1.1 (Visiting Step)**.** A Visiting Step is the conjunction of a Path Step with the associated spread index. It is written in the form s_i–k–e, or as a 3-tuple (i, k, e), where i represents the i-th streamline s_i, k represents the k-th step in the traversal of s_i, and e represents the e-th sediment spread step at s_i–k.

For a clearer understanding, Path Steps are illustrated in Figure 5.4, and some Visiting Steps are shown in Figure 5.5 (dotted arrows). The spread index is the same index referenced by the Moore neighbourhood (see Section 4.3), considering the current Path Step as the centre of the sediment spread.

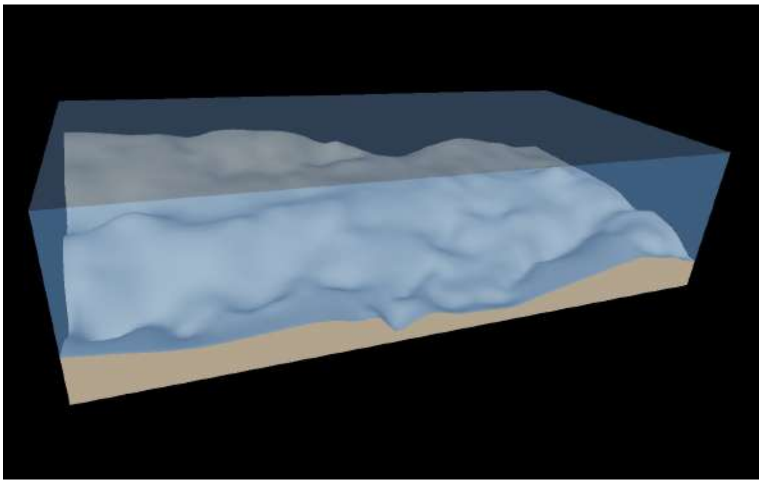

Figure 5.1: 3D depositional surface.

5.1.1 Path Restrictions

The earliest step of the inversion algorithm is the establishment of path restrictions. This is a pre-processing step for the algorithm. Streamlines are traversed individually, from start to end, and if a restricted vertex is found in a given Path Step during the traversal, all the following Path Steps will be marked with the heaviest possible sediment found in the restricted vertex's demand.

Path restrictions are necessary for two reasons:

- Simulations encompassing the inversion of different sediments must restrict their paths in order to prevent heavier sediments to be deposited on top of lighter sediments.
- Simulations with just one sediment and multiple restricted vertices may require path restrictions in order to prevent overfilling of demands.

5.1.2 Spread Restrictions

Analogously to Section 5.1.1, heavier sediments cannot be deposited on top of lighter sediments. Specifically, if a vertex has its demand fully satisfied but at incomplete capacity, it is vulnerable to other visitors that may deposit extra amounts of sediment to top the remaining capacity totals. It is noteworthy that spread restrictions are negligible in unrestricted vertices, yet very important on vertices where an exact distribution of sediments must be matched.

Therefore, the algorithm ought to verify, for every Path Step visited, what visitors have restricted sediment spread towards the vertex represented by the current Path Step. This is done by verifying the heaviest possible sediment found topping the restricted vertex's demand, and comparing with the visitor's spread sediment type. If the sediment to be spread by the visitor is heavier than the heaviest possible sediment at the restricted vertex, this visitor is removed from the visitor list for that specific vertex, and a spread lock (5.1.7) is placed on the visitor's Path Step.

5.1.3 Visitor Lists

Considering that streamlines are traversed iteratively, one step at a time, and with spreading steps, most vertices will be visited more than once by neighbouring streamlines. More than one visit for the same streamline can occur in different steps, and this phenomenon is normally associated with large spread radii or streamline geometries that may reach the same vertex in different spread steps. It is important to stress that every Visitor List in each vertex will only have one entry per Path Step.

In order to obtain Visitor Lists, it suffices to perform one run of the forward algorithm, registering every visit in a queue inside each vertex. Therefore, each vertex will contain its own Visitor List, i.e., a list of Path Steps that will proceed to reach the said vertex in case there is enough sediment at the time of its traversal.

5.1.4 Initial Vertex and Traversal Order

One may utilise the same run of the algorithm above to register the order of visited vertices. The initial vertex for the backward algorithm is the last *restricted vertex* visited in the forward algorithm. Repeated instances of restricted vertices are not considered for the definition of the initial vertex.

Choosing the appropriate initial vertex is very important for the correctness of the algorithm. Extra sediment volumes, or insufficient amounts can be accounted in case the initial vertex chosen is not correct.

The traversal order of the algorithm is the reversed order of visit from the forward algorithm, starting from the initial vertex.

5.1.5 Step Traversal

The traversal starts from the initial vertex. Since the initial vertex is a restricted vertex, there is a demand for sediments associated with it. The algorithm will then attempt to fulfil this demand by requesting sediments from other Path Steps present in its Visitor List.

It is noteworthy that non-direct sediment transfers are limited to the exponential factor f^{-d} associated with the decrease in distance during sediment spread. That said, if the first visitor in the Visitor List has distance d from the current step u, the maximum amount transferred cannot exceed $cap(u) \cdot f^{-d}$ per step. Also, $cap(u)$ changes at every sediment transfer.

Proposition 5.1.5.1. *A vertex can only be completely filled via direct transfer if* $f > 1$.

Proof. We define *direct transfer* as when a Path Step has distance 0 from the target vertex. Suppose there is a vertex v that can be completely filled via non-direct transfer $\{d \in \mathbb{Z}_{\geq 0} \mid d > 0\}$. Since $d \geq 1$ and $f > 1$, f^{-d} is strictly less than 1. Thus, $cap(v) \cdot f^{-d} < cap(v)$. The vertex is not filled at the first visit by a non-direct transfer. Repeating the steps, we denote $cap_n(v)$ as the remaining capacity after the n-th visit. If the $n{+}1$-th visit holds a non-direct transfer, the previous relationship holds, and $cap_n(v) \cdot f^{-d} < cap_n(v)$. Therefore, v cannot ever be completely filled via non-direct transfer for values of f greater than 1. Contradiction. □

After the capacity (or demand) of the vertex is filled, the vertex is marked as visited and the algorithm proceeds to the next vertex following the traversal order defined in section 5.1.4.

5.1.6 Auxiliary Visitors

In order to adequately invert the sediment spread procedure described in 4.3, the algorithm must account for other sediment transfers that occur between the start of the spread and the target vertex.

An *auxiliary visitor* is a Path Step that transfers sediments to a non-target vertex in a collateral manner. Auxiliary vertices are vertices that are not in the path of any streamline, and thus can only be accessed by sediment spread.

It is crucial that all vertices that participate in the depositional process, either auxiliary or not, are accounted for within the inversion algorithm. Otherwise, a slight deviation or skipped vertex may contaminate the whole result afterwards.

5.1.7 Spread Locks

When multiple streamlines reach the same vertex, the visiting order must be dutifully followed, respecting the appropriate deposition priorities. However, when a Path Step deposits the maximum capacity or demand for the target vertex *before* subsequent Path Steps can deposit into the same vertex, a *spread lock* is placed on all following Path Steps that visit the previously filled vertex.

Spread locks are necessary because:

- Only one Path Step is effectively going to fulfil the maximum capacity of a vertex in the forward algorithm, thus, the spread event can only be triggered by one visitor.
- The initial Visitor List does not consider which Path Step actually tops the vertex capacity. This is calculated during step traversal.
- It should not be possible to demand spread quantities from Path Steps that are not spreading, hence the necessity of a lock.

5.1.8 Continuation Locks

When a deposit takes place in a vertex where the restriction is met yet its capacity is not fulfilled, all other visitors must be specifically disallowed to deposit in that particular vertex. Therefore, all visitors that occur after a continuation lock are flagged as not allowed, and their depositions cannot continue after that point, hence the name *continuation lock*.

5.1.9 Demand readjustment and feasibility validation

Due to the topology of streamlines, there is a possibility that continuation or spread locks are added *after* some otherwise illegal visit is accounted for that particular vertex. In this case, it suffices to perform a single run per Path Step and rebalance the demands according to which demands are disallowed or required. For example, if a visit is deemed illegal at step k but registered in a previous step, the readjustment should remove this visitor and all the subsequent others that might depend on it.

If this process can be carried without leaving any contradictions in the visitor tables, it shows that the model is feasible and can produce a solution for the problem. Otherwise, the backward algorithm is unable to provide a solution and might require a change in the input parameters.

5.2 A Thorough Run of the Algorithm

In order to fully demonstrate the concepts listed in the previous sections, we will perform a detailed run of an instance of the problem. The backward simulation algorithm is presented in Algorithm 5.1.

5.2.1 Input Data

We will begin with a sample grid of size $4 \cdot 4 = 16$ vertices (Figure 5.2). The vertices have the following capacities:

$$CAP = \begin{bmatrix} cap(v_{1,1}) & cap(v_{1,2}) & cap(v_{1,3}) & cap(v_{1,4}) \\ cap(v_{2,1}) & cap(v_{2,2}) & cap(v_{2,3}) & cap(v_{2,4}) \\ cap(v_{3,1}) & cap(v_{3,2}) & cap(v_{3,3}) & cap(v_{3,4}) \\ cap(v_{4,1}) & cap(v_{4,2}) & cap(v_{4,3}) & cap(v_{4,4}) \end{bmatrix}$$

and

$$CAP = \begin{bmatrix} 2400 & 3300 & 2000 & 2900 \\ 1000 & 3300 & 3200 & 1600 \\ 2800 & 3400 & 1600 & 2000 \\ 1500 & 1700 & 3300 & 1800 \end{bmatrix}.$$

We will define 3 streamlines for this instance (Figure 5.3). Their paths are:

$$s_1 = \{v_{1,1}, v_{2,1}, v_{2,2}, v_{3,2}, v_{4,2}\},$$

$$s_2 = \{v_{1,2}, v_{2,2}, v_{2,3}, v_{3,3}, v_{3,4}\},$$

and

$$s_3 = \{v_{1,4}, v_{2,4}, v_{3,4}\}.$$

Algorithm 5.1 Simplified Backward Simulation Algorithm

```
Build the Visitor Lists for all vertices (5.1.3)
Update all Path Restrictions (5.1.1)
Store the ordered list of Path Steps in P
for every s_i–k ∈ P do
    Store the ordered list of visitors in V_is
    Enforce spread restrictions (5.1.2) and remove restricted visitors from V_is
    for every s'_i–k'–j ∈ V_is do
        d := CHEBYSHEV DISTANCE(s_i–k, s'_i–k')
        Add the restriction demand to demand.
        Add the spread demand from s_i–k to demand.
        if s_i = s'_i then
            Add the overall stream demand to demand.
        end if
        demand := demand − deposited
        if demand > 0 then
            if cap(s_i–k) > 0 then
                if demand < cap(s_i–k) · f^{−d} OR j = 0 then
                    Add demand to the overall stream amount at s_i.
                    if j > 0 AND s'_i–k' has spread lock then
                        continue
                    end if
                    cap(s_i–k) := cap(s_i–k) − demand
                    Set s_i–k topped by s'_i–k'–j
                else
                    if s_i–k has spread lock (5.1.7) AND spread lock is not s'_i–k'–j
then
                        continue
                    end if
                    deposit := cap(s_i–k) · f^{−d}
                    Add deposit to fixed demand amount of s'_i–k'
                    SET AUXILIARY DEMANDS(s'_i–k'–j)
                    deposited := deposited + deposit
                end if
            else
                Add demand to the overall stream amount at s_i.
            end if
        end if
    end for
end for
Add all auxiliary demands to the respective stream totals.
```

Algorithm 5.2 SET AUXILIARY DEMANDS(s_i–k–E_j)

r := maximum spread radius.
Create the spread neighbourhood of j items and radius r for s_i–k–j as in Figure 4.10 and store it in N.
for every spread step $z \leq j \in N$ **do**
    d := CHEBYSHEV DISTANCE(s_i–k–z, s_i–k–j)
    Add auxiliary visit to s_i–k–z with $cap(s_i\text{–}k\text{–}z) \cdot f^{-d}$ demand.
end for
Set s_i–k as the Spread Lock for this vertex.

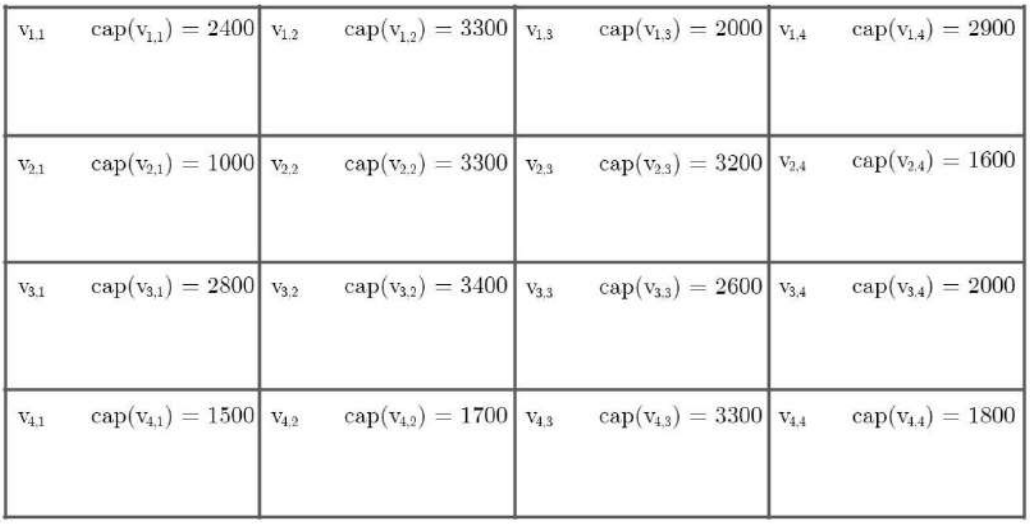

Figure 5.2: Sample depositional grid.

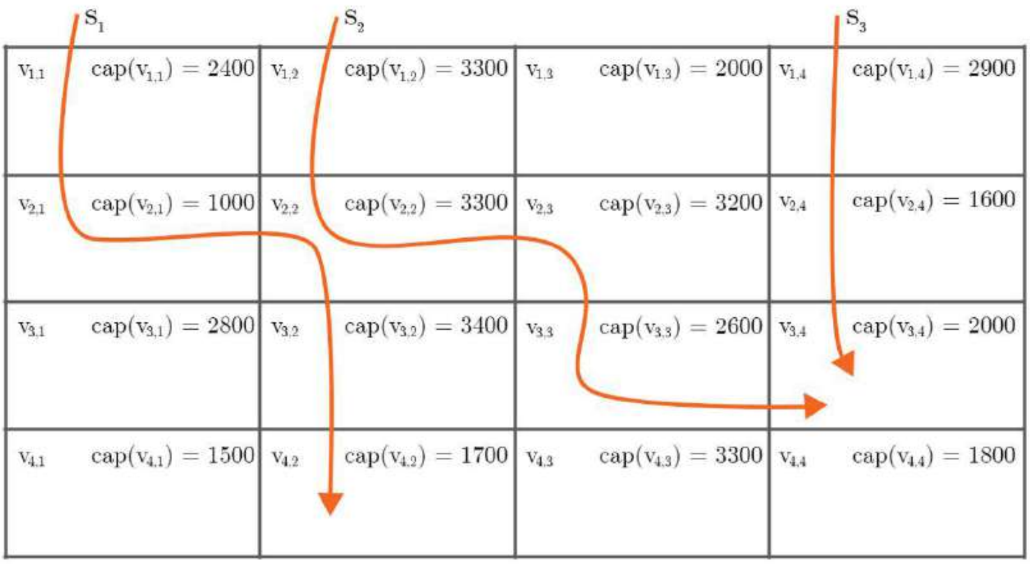

Figure 5.3: Sample depositional grid with streamlines s_1, s_2, and s_3.

We will establish 3 restrictions within this grid. Our simulation will encompass 3 different sediments, named w_1, w_2, and w_3, from coarser to finer. The backward algorithm must be able to find the initial amounts for each streamline s_i in order to match the required distribution of sediments at each restriction.

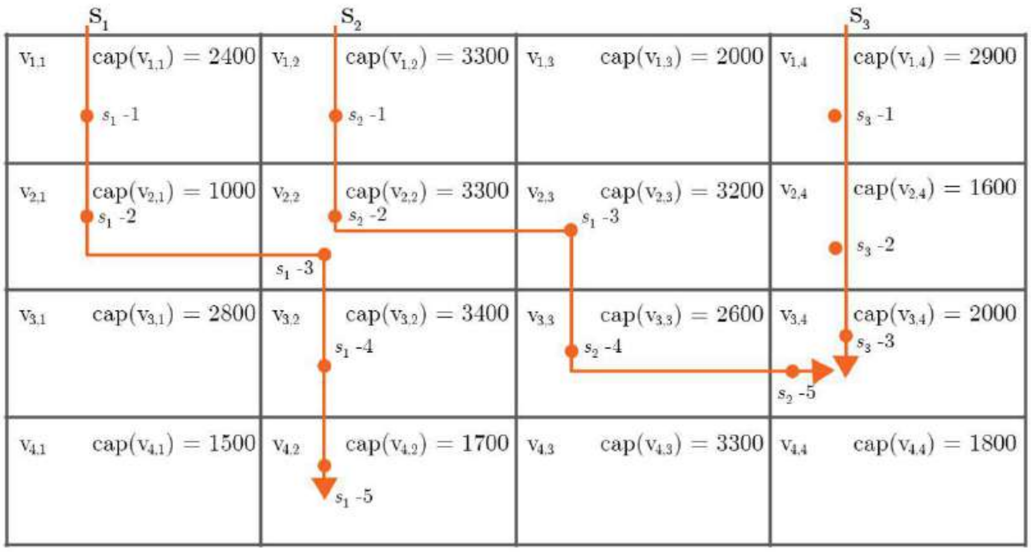

Figure 5.4: Path Steps of s_1, s_2, and s_3.

The restrictions are defined as:

$$R(v_{2,4}) = \{800w_1, 400w_2, 400w_3\}$$

$$R(v_{3,2}) = \{1200w_1, 1200w_2\}$$

$$R(v_{3,4}) = \{1500w_3\}$$

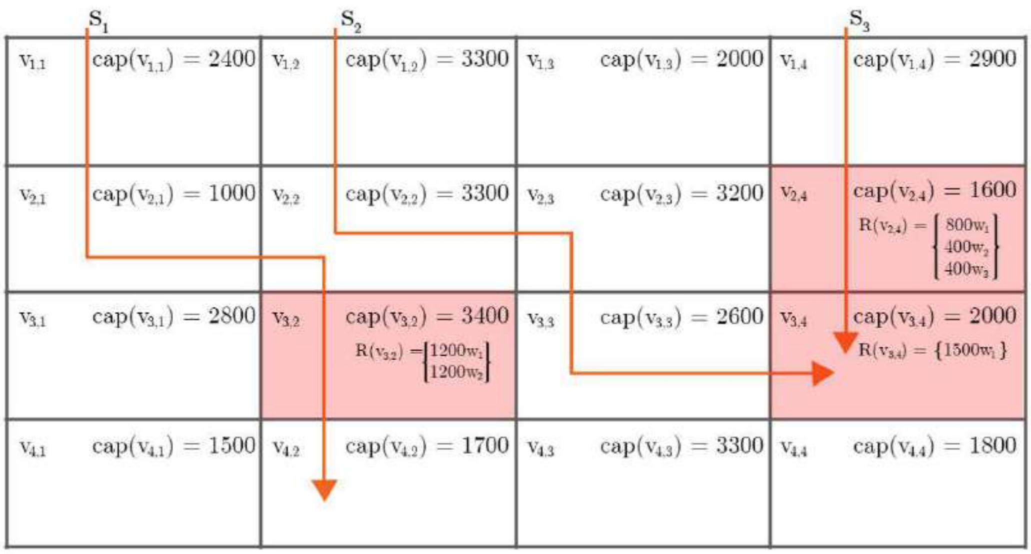

Figure 5.5: Restrictions $R(v_{2,4})$, $R(v_{3,2})$, and $R(v_{3,4}$ and their respective demands.

For this example, we will assume the maximum spread radius (4.3.1) $r = 1$, and the exponential decrease factor $f = 4$.

5.2.2 Visitor List

After the input data is established, we now proceed to build the list of visitors at each step of the simulation. This is achieved by simulating a forward run of all streamlines considering the maximum spread radius defined at the simulation. Vertices that cannot be accessed are excluded from the visitor list. The spread index of each visit will be listed under the according vertex. Index 0 represents direct deposition (not limited by $dep(u) = cap(u) \times f^{-d}$, since $d = 0$.)

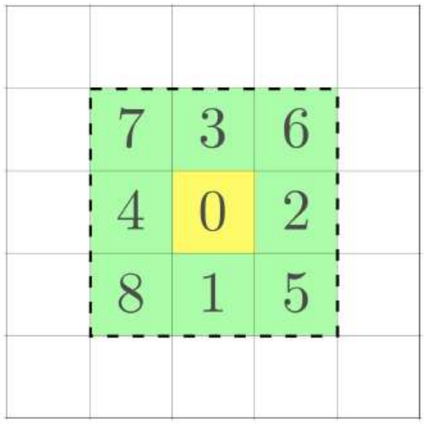

Figure 5.6: $N(v, 2)$ immediate Moore neighbourhood, defined as input up to $d = 1$. v has $d = 0$.

$$s_1\text{–}1 : \{\underset{0}{v_{1,1}}, \underset{1}{v_{2,1}}, \underset{2}{v_{1,2}}, \underset{5}{v_{2,2}}\}$$

$$s_2\text{–}1 : \{\underset{0}{v_{1,2}}, \underset{1}{v_{2,2}}, \underset{2}{v_{1,3}}, \underset{4}{v_{1,1}}, \underset{5}{v_{2,3}}, \underset{8}{v_{2,1}}\}$$

$$s_3\text{–}1 : \{\underset{0}{v_{1,4}}, \underset{1}{v_{2,4}}, \underset{4}{v_{1,3}}, \underset{8}{v_{2,3}}\}$$

$$s_1\text{–}2 : \{\underset{0}{v_{2,1}}, \underset{1}{v_{3,1}}, \underset{2}{v_{2,2}}, \underset{3}{v_{1,1}}, \underset{5}{v_{3,2}}, \underset{6}{v_{1,2}}\}$$

$$s_2\text{–}2 : \{\underset{0}{v_{2,2}}, \underset{1}{v_{3,2}}, \underset{2}{v_{2,3}}, \underset{3}{v_{1,2}}, \underset{4}{v_{2,1}}, \underset{5}{v_{3,3}}, \underset{6}{v_{1,3}}, \underset{7}{v_{1,1}}, \underset{8}{v_{3,1}}\}$$

$$s_3\text{–}2 : \{\underset{0}{v_{2,4}}, \underset{1}{v_{3,4}}, \underset{3}{v_{1,4}}, \underset{4}{v_{2,3}}, \underset{7}{v_{1,3}}, \underset{8}{v_{3,3}}\}$$

$$s_1\text{–}3 : \{\underset{0}{v_{2,2}}, \underset{1}{v_{3,2}}, \underset{2}{v_{2,3}}, \underset{3}{v_{1,2}}, \underset{4}{v_{2,1}}, \underset{5}{v_{3,3}}, \underset{6}{v_{1,3}}, \underset{7}{v_{1,1}}, \underset{8}{v_{3,1}}\}$$

$$s_2\text{–}3 : \{\underset{0}{v_{2,3}}, \underset{1}{v_{3,3}}, \underset{2}{v_{2,4}}, \underset{3}{v_{1,3}}, \underset{4}{v_{2,2}}, \underset{5}{v_{3,4}}, \underset{6}{v_{1,4}}, \underset{7}{v_{1,2}}, \underset{8}{v_{3,2}}\}$$

$$s_3\text{–}3 : \{\underset{0}{v_{3,4}}, \underset{1}{v_{4,4}}, \underset{3}{v_{2,4}}, \underset{4}{v_{3,3}}, \underset{7}{v_{2,3}}, \underset{8}{v_{4,3}}\}$$

$$s_1\text{–}4 : \{\underset{0}{v_{3,2}}, \underset{1}{v_{4,2}}, \underset{2}{v_{3,3}}, \underset{3}{v_{2,2}}, \underset{4}{v_{3,1}}, \underset{5}{v_{4,3}}, \underset{6}{v_{2,3}}, \underset{7}{v_{2,1}}, \underset{8}{v_{4,1}}\}$$

$$s_2\text{–}4 : \{\underset{0}{v_{3,3}}, \underset{1}{v_{4,3}}, \underset{2}{v_{3,4}}, \underset{3}{v_{2,3}}, \underset{4}{v_{3,2}}, \underset{5}{v_{4,4}}, \underset{6}{v_{2,4}}, \underset{7}{v_{2,2}}, \underset{8}{v_{4,2}}\}$$

$$s_1\text{–}5 : \{\underset{0}{v_{4,2}}, \underset{2}{v_{4,3}}, \underset{3}{v_{3,2}}, \underset{4}{v_{4,1}}, \underset{6}{v_{3,3}}, \underset{7}{v_{3,1}}\}$$

$$s_2\text{–}5 : \{\underset{0}{v_{3,4}}, \underset{1}{v_{4,4}}, \underset{3}{v_{2,4}}, \underset{4}{v_{3,3}}, \underset{7}{v_{2,3}}, \underset{8}{v_{4,3}}\}$$

The notation s_i–k–e verbally represents 'the e-th spread of the k-th step of streamline s_i. For instance, s_2–2–5 represents the moment where s_2–2 performs its fifth spread step, reaching vertex $v_{3,3}$.

5.2.3 Initial Vertex

The backward algorithm will be performed once for every sediment w_i available. Considering all zero-indexed visits from the previous step, we list:

$$\underset{s_1-1}{v_{1,1}}, \underset{s_2-1}{v_{1,2}}, \underset{s_3-1}{v_{1,4}}, \underset{s_1-2}{v_{2,1}}, \underset{s_2-2}{v_{2,2}}, \underset{s_3-2}{v_{2,4}}, \underset{s_1-3}{v_{2,2}}, \underset{s_2-3}{v_{2,3}}, \underset{s_3-3}{v_{3,4}}, \underset{s_1-4}{v_{3,2}}, \underset{s_2-4}{v_{3,3}}, \underset{s_1-5}{v_{4,2}}, \underset{s_2-5}{v_{3,4}}$$

Highlighting the vertices with w_1 demands, we have:

$$\underset{s_1-1}{v_{1,1}}, \underset{s_2-1}{v_{1,2}}, \underset{s_3-1}{v_{1,4}}, \underset{s_1-2}{v_{2,1}}, \underset{s_2-2}{v_{2,2}}, \underset{s_3-2}{v_{2,4}}, \underset{s_1-3}{v_{2,2}}, \underset{s_2-3}{v_{2,3}}, \underset{s_3-3}{v_{3,4}}, \underset{s_1-4}{v_{3,2}}, \underset{s_2-4}{v_{3,3}}, \underset{s_1-5}{v_{4,2}}, \underset{s_2-5}{v_{3,4}}$$

The initial vertex is $v_{3,2}$, starting at Path Step s_1–4.

The backward traversal order for w_1 inversion will be s_1–4, s_3–3, s_2–3, s_1–3, s_3–2, s_2–2, s_1–3, s_3–1, s_2–1, s_1–1.

Highlighting the vertices with w_2 demands, we have:

$$\underset{s_1-1}{v_{1,1}}, \underset{s_2-1}{v_{1,2}}, \underset{s_3-1}{v_{1,4}}, \underset{s_1-2}{v_{2,1}}, \underset{s_2-2}{v_{2,2}}, \underset{s_3-2}{v_{2,4}}, \underset{s_1-3}{v_{2,2}}, \underset{s_2-3}{v_{2,3}}, \underset{s_3-3}{v_{3,4}}, \underset{s_1-4}{v_{3,2}}, \underset{s_2-4}{v_{3,3}}, \underset{s_1-5}{v_{4,2}}, \underset{s_2-5}{v_{3,4}}$$

The initial vertex is $v_{3,2}$, starting at Path Step s_1–4.

The backward traversal order for w_2 inversion will be s_1–4, s_3–3, s_2–3, s_1–3, s_3–2, s_2–2, s_1–3, s_3–1, s_2–1, s_1–1, Highlighting the vertices with w_3 demands, we have:

$v_{1,1}$,	$v_{1,2}$,	$v_{1,4}$,	$v_{2,1}$,	$v_{2,2}$,	$v_{2,4}$,	$v_{2,2}$,	$v_{2,3}$,	$v_{3,4}$,	$v_{3,2}$,	$v_{3,3}$,	$v_{4,2}$,	$v_{3,4}$
s_1–1	s_2–1	s_3–1	s_1–2	s_2–2	s_3–2	s_1–3	s_2–3	s_3–3	s_1–4	s_2–4	s_1–5	s_2–5

The initial vertex is $v_{3,4}$, starting at Path Step s_3–3.

The backward traversal order for w_3 inversion will be s_1–4, s_3–3, s_2–3, s_1–3, s_3–2, s_2–2, s_1–3, s_3–1, s_2–1, s_1–1,

5.2.4 Step Traversal for w_1

Step s_1–4

The inversion starts at step s_1–4, on vertex $v_{3,2}$. This vertex is restricted and has a demand of $1200w_1$ sediment units. Querying the visitor list, we can obtain the list of visiting steps that reach $v_{3,2}$:

Visitor	Distance (d)	Deposit
s_1–2–5	1	$dep_1 := cap(v_{3,2}) \times f^{-d}$
s_2–2–1	1	$dep_2 := (cap(v_{3,2}) - dep_1) \times f^{-d}$
s_2–3–8	1	$dep_3 := (cap(v_{3,2}) - \sum_i^2 dep_i) \times f^{-d}$
$\mathbf{s_1}$**–4–0**	0	$dep_4 := (cap(v_{3,2}) - \sum_i^3 dep_i)$
s_2–4–4	1	$dep_3 := (cap(v_{3,2}) - \sum_i^4 dep_i) \times f^{-d}$
Total	——	$dep := \sum_i dep_i$

Replacing $cap(v_{3,2}) = 3400$ and $f = 4$, we have:

Visitor	Distance (d)	Deposit
s_1–2–5	1	$dep_1 := 850$
s_2–2–1	1	$dep_2 := 350$ ($1200w_1$ matched. Stop demands.)
s_2–3–8	1	not allowed
$\mathbf{s_1}$**–4–0**	0	not allowed
s_2–4–4	1	not allowed
Total	——	$dep := 1200w_1$

It only required 2 spread visitors to fulfil $v_{3,2}$'s restriction requirements. Note that since $v_{3,2}$ is a *restricted vertex*, we do not aim to fill the whole capacity, but instead satisfy the required amount. It is important to point out that s_2–2–1 did not use its full capacity to spread. This means that for all s_2–2–e where $e > 1$, the spread step will not be able to deposit any sediments to other vertices. Since the spread mechanism is essentially sequential, depositing on subsequent vertices would *require* s_2–2–1 to use its maximum capacity, which would exceed the restriction values for

$v_{3,2}$. To prevent this from happening, we will add a *continuation lock* to s_2–2–1, meaning that deposits are not allowed for any s_2–k–e where $k \geq 2$ and $e > 1$.

Step s_3–3

This step has no demand of w_1 and neither is an intermediary for other steps. We will skip to the next step.

Step s_2–3

Building up $v_{2,3}$'s visitor list, we identify an intermediary step, meaning that this demand must be fully satisfied in order to proceed to the final restriction.

Considering $cap(v_{2,3}) = 3200$:

Visitor	Distance (d)	Deposit
s_2–1–5	1	$dep_1 := 800$ (required for s_2–2–1 above).
s_3–1–8	1	no demand
s_2–2–2	1	locked (continuation lock at s_2–2–1)
s_3–2–4	1	no demand
s_1–3–2	1	no demand
$\mathbf{s_2}$–3–0	0	locked (continuation lock at s_2–2–1)
s_3–3–7	1	no demand
s_1–4–6	1	no demand
s_2–4–3	1	locked (continuation lock at s_2–2–1)
Total	——	$dep := 800w_1$

Step s_1–3

Building up $v_{2,2}$'s visitor list, we identify intermediary steps, meaning that those demands must be fully satisfied in order to proceed to the final restriction.

Considering $cap(v_{2,2}) = 3300$:

Visitor	Distance (d)	Deposit
s_1–1–5	1	$dep_1 := 825$ (required for s_1–2–5 above).
s_2–1–1	1	$dep_2 := 618.75$ (required for s_2–2–1 above).
s_1–2–2	1	$dep_3 := 464.0625$ (required for s_1–2–5 above).
$\mathbf{s_2}$**–2–0**	0	$dep_4 := 1392.1875$ (required for s_2–2–1 above). Maximum capacity reached. Stop demands.
$\mathbf{s_1}$**–3–0**	0	vertex is full (...)
...	...	...
Total	——	$dep := 3300w_1$

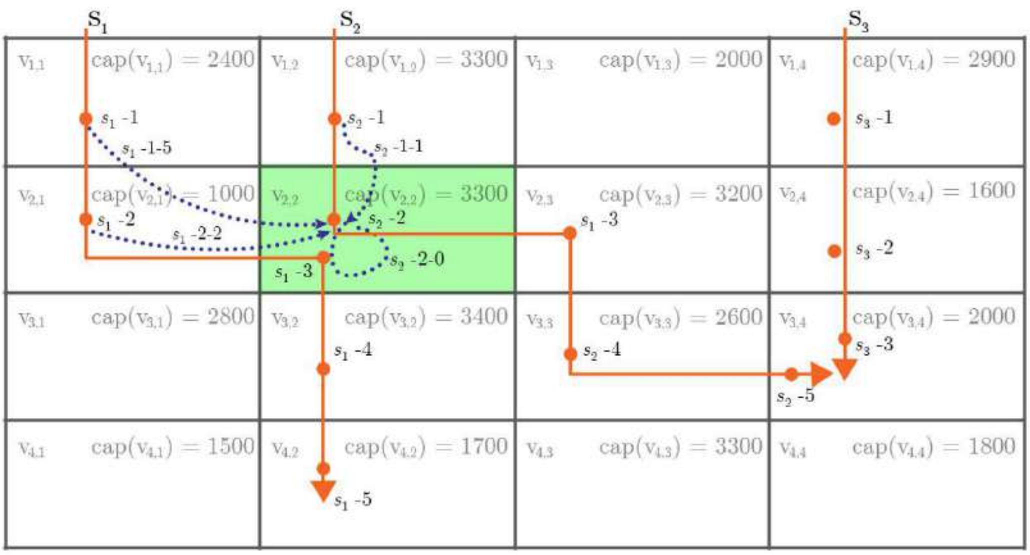

Figure 5.7: Visitors at $v_{2,2}$.

Step s_3–2

This vertex ($v_{2,4}$) is restricted and has a demand of $800w_1$ sediment units.

Considering $cap(v_{2,4}) = 1600$:

Visitor	Distance (d)	Deposit
s_3–1–1	1	$dep_1 := 400$
$\mathbf{s_3}$**–2–0**	0	$dep_2 := 400$ ($800w_1$ matched. Stop demands. Since the deposit was not the maximum amount, set continuation lock at s_3–2–0.)
s_2–3–2	1	not allowed (...)
...	...	...
Total	——	$dep := 800w_1$

Step s_2–2

Vertex $v_{2,2}$ was already visited at step s_1–3. The demands are the same.

Step s_1–2

Building up $v_{2,1}$'s visitor list, we identify intermediary steps, meaning that those demands must be fully satisfied in order to proceed to the final restriction.

Considering $cap(v_{2,1}) = 1000$:

Visitor	Distance (d)	Deposit
s_1–1–1	1	$dep_1 := 250$ (required for s_1–1–5 above).
s_2–1–8	1	$dep_2 := 187.5$ (required for s_2–2–0 above).
$\mathbf{s_1}$**–2–0**	0	$dep_3 := 562.5$ (required for s_1–2–2 above). Maximum capacity reached. Stop demands.
s_2–2–4	1	vertex is full (...)
...	...	...
Total	——	$dep := 1000w_1$

Step s_3–1

Building up $v_{1,4}$'s visitor list, we identify an intermediary step, meaning that this demand must be fully satisfied in order to proceed to the final restriction.

Considering $cap(v_{1,4}) = 2900$:

Visitor	Distance (d)	Deposit
$\mathbf{s_3}$**–1–0**	0	$dep_1 := 2900$ (required for s_3–1–1 above). Maximum capacity reached. Stop demands.
s_2–3–2	1	vertex is full (...)
...	...	...
Total	——	$dep := 1000w_1$

Step s_2–1

Building up $v_{1,2}$'s visitor list, we identify an intermediary step, meaning that this demand must be fully satisfied in order to proceed to the final restriction.

Considering $cap(v_{1,2}) = 3300$:

Visitor	Distance (d)	Deposit
s_1–1–2	1	$dep_1 := 825$ (required for s_1–1–5 above).
$\mathbf{s_2}$–1–0	0	$dep_1 := 2475$ (required for s_3–1–1 above). Maximum capacity reached. Stop demands.
s_1–2–6	1	vertex is full (...)
...	...	...
Total	——	$dep := 3300w_1$

Step s_1–1

Building up $v_{1,1}$'s visitor list, we identify an intermediary step, meaning that this demand must be fully satisfied in order to proceed to the final restriction.

Considering $cap(v_{1,1}) = 2400$:

Visitor	Distance (d)	Deposit
$\mathbf{s_1}$–1–0	0	$dep_1 := 2400$ (required for s_1–1–1 above). Maximum capacity reached. Stop demands.
s_2–1–4	1	vertex is full (...)
...	...	...
Total	——	$dep := 2400w_1$

5.2.5 Auxiliary Visitors for w_1

Although the step traversal is complete, this inversion did not account for other *collateral* deposits that may have occurred in vertices that are not part of any streamline, but must still have its deposits added to the total amount of sediments. Those visitors are called *auxiliary visitors*, and were described in section 5.1.6.

As it was pointed in section 4.3, it is not possible to deposit in a spread step e without depositing in all previous steps $e' < e$. The subroutine described in 5.2 was devised to run alongside the main step traversal inversion algorithm.

Once each auxiliary vertex has all their auxiliary visitors stored, they can traverse their own visitor list following the same precedence rules as seen on the step traversal procedure.

We can identify 3 auxiliary vertices: $v_{3.1}$, $v_{1,3}$, and $v_{3,3}$. Although $v_{3,3}$ has a streamline that crosses the vertex, since this vertex appears *after* the initial vertex, it is considered an auxiliary vertex for all effects.

Considering $cap(v_{3,1}) = 2800$:

Visitor	Distance (d)	Deposit
s_1–2–1	1	$dep_1 := 700$
Total	——	$dep := 700w_1$

Considering $cap(v_{1,3}) = 2000$:

Visitor	Distance (d)	Deposit
s_2–1–2	1	$dep_1 := 500$
s_3–1–4	1	$dep_2 := 375$
Total	——	$dep := 875w_1$

5.2.6 Demand readjustment and feasibility for w_1

According to Section 5.1.9, we must verify the necessity for readjustments for w_1. In this case, we have, for vertex $v_{2,3}$:

Visitor	Distance (d)	Deposit
s_2–1–5	1	$dep_1 := 800$ (required for s_2–2–1 above).
s_3–1–8	1	$dep_2 := 600$ (readjustment for s_3–2–0).
s_2–2–2	1	locked (continuation lock at s_2–2–1)
s_3–2–4	1	no demand
s_1–3–2	1	no demand
$\mathbf{s_2}$–3–0	0	no demand
s_3–3–7	1	no demand
s_1–4–6	1	no demand
s_2–4–3	1	locked (continuation lock at s_2–2–1)
Total	——	$dep := 1400w_1$

In this case, the demands could be correctly redistributed along the visitors, confirming the feasibility for w_1.

5.2.7 Results for w_1

Adding up all visitors from s_1, s_2, and s_3 we have a total of $6876.5625w_1$, $6323.4375w_1$, and $4675w_1$, respectively. These are the exact values of w_1 required to satisfy the restrictions $R(v_{2,4})$ and $R(v_{3,2})$.

5.2.8 Step traversal for w_2

Step s_1–4

The inversion starts at step s_1–4, on vertex $v_{3,2}$. This vertex is restricted and has a demand of $1200w_2$ sediment units. Querying the visitor list, we can obtain the list of visiting steps that reach $v_{3,2}$:

Replacing $cap(v_{3,2}) = 2200$ and $f = 4$, we have:

Visitor	Distance (d)	Deposit
s_1–2–5	1	locked (spread lock at $w_1 : s_1$–2–0).
s_2–2–1	1	locked (spread lock at $w_1 : s_2$–2–0).
s_2–3–8	1	$dep_1 := 550w_2$
$\mathbf{s_1}$–4–0	0	$dep_2 := 650w_2$
s_2–4–4	1	not allowed
Total	——	$dep := 1200w_2$

Step s_3–3

This step has no demand of w_2 and neither is an intermediary for other steps. We will skip to the next step.

Step s_2–3

Building up $v_{2,3}$'s visitor list, we identify intermediary steps, meaning that this demand must be fully satisfied in order to proceed to the final restriction.

Considering $cap(v_{2,3}) = 1800$:

Visitor	Distance (d)	Deposit
s_2–1–5	1	locked (spread lock at $w_1 : s_2$–1–0).
s_3–1–8	1	locked (spread lock at $w_1 : s_3$–1–0).
s_2–2–2	1	locked (spread lock at $w_1 : s_2$–2–0).
s_3–2–4	1	no demand
s_1–3–2	1	locked (spread lock at $w_1 : s_2$–2–0).
$\mathbf{s_2}$–3–0	0	$dep_1 := 1800$ (required for s_2–3–8 above). Maximum capacity reached. Stop demands.
s_3–3–7	1	vertex is full
s_1–4–6	1	vertex is full
s_2–4–3	1	vertex is full
Total	——	$dep := 1800w_2$

Step s_1–3

Vertex $v_{2,2}$ was already filled during the w_1 traversal. We shall skip to the next step.

Step s_3–2

This vertex ($v_{2,4}$) is restricted and has a demand of $400w_2$ sediment units.

Considering $cap(v_{2,4}) = 800$:

Visitor	Distance (d)	Deposit
s_3–1–1	1	locked (spread lock at $w_1 : s_3$–1–0).
$\mathbf{s_3}$**–2–0**	0	$dep_1 := 400$ ($400w_2$ matched. Stop demands. Since the deposit was not the maximum amount, set continuation lock at s_3–2–0.)
s_2–3–2	1	not allowed (...)
...	...	...
Total	——	$dep := 400w_2$

Step s_2–2

Vertex $v_{2,2}$ was already visited at step s_1–3.

Step s_1–2

Vertex $v_{2,1}$ was already filled during the w_1 traversal.

Step s_3–1

Vertex $v_{1,4}$ was already filled during the w_1 traversal.

Step s_2–1

Vertex $v_{1,2}$ was already filled during the w_1 traversal.

Step s_1–1

Vertex $v_{1,1}$ was already filled during the w_1 traversal.

5.2.9 Auxiliary Visitors for w_2

Considering $cap(v_{1,3}) = 843.75$:

Visitor	Distance (d)	Deposit
s_2–3–3	1	$dep_1 := 210.9375$
Total	——	$dep := 210.9375w_2$

Considering $cap(v_{3,3}) = 1200$:

Visitor	Distance (d)	Deposit
s_2–3–1	1	$dep_1 := 300$
Total	——	$dep := 300w_2$

5.2.10 Demand readjustment and feasibility for w_2

Analogously to Section 5.2.6, the same process of readjustment and feasibility check applies after the first traversal.

We can observe that step s_3–2 possesses a restriction fulfilled by visitor s_3–2–0. Since the vertex is not full at that stage, any other visitors that come after it are not allowed to deposit and interfere with the sediment amounts.

A conflict, then, comes up: s_2–3–2 is a visitor in step s_3–2 that appears after the fulfilment of the vertex. Consequently, s_2–3–2 must not carry any sediment. However, at step s_1–4, we see a visitor s_2–3–8 depositing $550w_2$ into the vertex. Since s_2–3–8 depends on all spread steps $e' < 8$ to deposit in their respective targets, it cannot have any sediment, for s_2–3–2 is not allowed to deposit. We can, then, fix $v_{3,2}$:

Visitor	Distance (d)	Deposit
s_1–2–5	1	locked (spread lock at $w_1 : s_1$–2–0).
s_2–2–1	1	locked (spread lock at $w_1 : s_2$–2–0).
s_2–3–8	1	~~$dep_1 := 550w_2$~~ not allowed
$\mathbf{s_1}$–4–0	0	~~$dep_2 := 650w_2$~~ $dep_1 := (650 + 550)w_2$. $1200w_2$ matched. Stop demands.
s_2–4–4	1	not allowed
Total	——	$dep := 1200w_2$

That change resonates in step s_2–3. Since there is no requirement for s_2–3–8 above, we can eliminate the demand for s_2–3–0:

Visitor	Distance (d)	Deposit
s_2–1–5	1	locked (spread lock at $w_1 : s_2$–1–0).
s_3–1–8	1	locked (spread lock at $w_1 : s_3$–1–0).
s_2–2–2	1	locked (spread lock at $w_1 : s_2$–2–0).
s_3–2–4	1	no demand
s_1–3–2	1	locked (spread lock at $w_1 : s_2$–2–0).
$\mathbf{s_2}$–3–0	0	~~$dep_1 := 1800$ (required for s_2–3–8 above).~~ no demand
s_3–3–7	1	no demand
s_1–4–6	1	no demand
s_2–4–3	1	no demand
Total	——	$dep := 0w_2$

Without the ability of spreading from s_2–3–0, both auxiliary visitors ($v_{1,3}$ and $v_{3,3}$) can be removed:

Considering $cap(v_{1,3}) = 843.75$:

Visitor	Distance (d)	Deposit
s_2–3–3	1	~~$dep_1 := 210.9375$~~ no demand
Total	——	$dep := 0w_2$

Considering $cap(v_{3,3}) = 1200$:

Visitor	Distance (d)	Deposit
s_2–3–1	1	~~$dep_1 := 300$~~ no demand
Total	——	$dep := 0w_2$

In this case, the demands could be correctly redistributed along the visitors, confirming the feasibility for w_2.

5.2.11 Results for w_2

Adding up all visitors from s_1, s_2, and s_3 we have a total of $1200w_2$, $0w_2$, and $400w_2$, respectively. These are the exact values of w_2 required to satisfy the restrictions $R(v_{2,4})$ and $R(v_{3,2})$.

5.2.12 Step traversal for w_3

Step s_2–5

The inversion starts at step s_2–5, on vertex $v_{3,4}$. This vertex is restricted and has a demand of $1500w_3$ sediment units. Querying the visitor list, we can obtain the list of visiting steps that reach $v_{3,4}$:

Considering $cap(v_{3,4}) = 2000$:

Visitor	Distance (d)	Deposit
s_3–2–1	1	$dep_1 := 500$
s_2–3–5	1	locked (spread lock at $w_2 : s_2$–3–0).
$\mathbf{s_3}$–3–0	0	$dep_2 := 1000$ ($1500w_3$ matched. Stop demands.)
...	...	...
Total	——	$dep := 1500w_3$

Step s_1–5

This step has no demand of w_3 and neither is an intermediary for other steps. We will skip to the next step.

Step s_2–4

This step has no demand of w_3 and neither is an intermediary for other steps. We will skip to the next step.

Step s_1–4

This step has no demand of w_3 and neither is an intermediary for other steps. We will skip to the next step.

Step s_3–3

Vertex $v_{3,4}$ was already visited at step s_2–5.

Step s_2–3

Vertex $v_{2,3}$ was already filled during the w_2 traversal.

Step s_1–3

Vertex $v_{2,2}$ was already filled during the w_1 traversal.

Step s_3–2

This vertex ($v_{2,4}$) is restricted and has a demand of $400w_3$ sediment units.

Considering $cap(v_{2,4}) = 400$:

Visitor	Distance (d)	Deposit
s_3–1–1	1	locked (spread lock at w_1 : s_3–1–0).
$\mathbf{s_3}$–2–0	0	$dep_1 := 400$ ($400w_3$ matched. Stop demands.)
s_2–3–2	1	no demand (...)
...	...	...
Total	——	$dep := 400w_3$

Step s_2–2

Vertex $v_{2,2}$ was already visited at step s_1–3.

Step s_1–2

Vertex $v_{2,1}$ was already filled during the w_1 traversal.

Step s_3–1

Vertex $v_{1,4}$ was already filled during the w_1 traversal.

Step s_2–1

Vertex $v_{1,2}$ was already filled during the w_1 traversal.

Step s_1–1

Vertex $v_{1,1}$ was already filled during the w_1 traversal.

5.2.13 Results for w_3

Adding up all visitors from s_1, s_2, and s_3 we have a total of $0w_3$, $0w_3$, and $1900w_3$, respectively. These are the exact values of w_3 required to satisfy the restrictions $R(v_{2,4})$ and $R(v_{3,4})$.

5.2.14 Final results

Adding up the demands from all 3 runs of the algorithm (one per sediment type w_i), we obtain:

- $s_1 : 6876.5625w_1, 650w_2, 0w_3$
- $s_2 : 7529.6875w_1, 3460.9375w_2, 0w_3$
- $s_3 : 4675w_1, 400w_2, 1900w_3$

5.3 Result Verification

Provided the final results of the backward simulation, we can now proceed with a forward run (i.e. a *backward-forward simulation*) to demonstrate that the results obtained by the backward algorithm are valid.

5.3.1 Forward w_1 run

For sediment w_1, streamlines s_1, s_2, and s_3 possess, respectively, $6876.5625w_1$, $6323.4375w_1$, and $4675w_1$ sediment units. We shall then perform a run of Algorithm 4.1 with these values, considering the original capacities of all vertices and visiting order, as defined in Section 5.2.1 and Section 5.2.2.

Step s_1–1

The amount of available sediment in s_1 before this step is $6876.5625w_1$.

Spread Visit	Target t	$cap(t)$	Distance (d)	Deposit ($cap(t) \cdot f^{-d}$)
$\mathbf{s_1}$**–1–0**	$v_{1,1}$	2400	0	$dep_1 := 2400$
s_1–1–1	$v_{2,1}$	1000	1	$dep_2 := 250$
s_1–1–2	$v_{1,2}$	3300	1	$dep_3 := 825$
s_1–1–5	$v_{2,2}$	3300	1	$dep_4 := 825$
Total	——	——	——	$dep_{s_1-1} := 4300w_1$

Step s_2–1

The amount of available sediment in s_2 before this step is $6323.4375w_1$.

Spread Visit	Target t	$cap(t)$	Distance (d)	Deposit ($cap(t) \cdot f^{-d}$)
$\mathbf{s_2}$**–1–0**	$v_{1,2}$	2475	0	$dep_1 := 2475$
s_2–1–1	$v_{2,2}$	2475	1	$dep_2 := 618.75$
s_2–1–2	$v_{1,3}$	2000	1	$dep_3 := 500$
s_1–1–3	$v_{1,1}$	0	1	$dep_4 := 0$ (vertex is full)
s_2–1–5	$v_{2,3}$	3200	1	$dep_5 := 800$
s_2–1–8	$v_{2,1}$	750	1	$dep_6 := 187.5$
Total	——	——	——	$dep_{s_2-1} := 4581.25w_1$

Step s_3–1

The amount of available sediment in s_3 before this step is $4675w_1$.

Spread Visit	Target t	$cap(t)$	Distance (d)	Deposit ($cap(t) \cdot f^{-d}$)
$\mathbf{s_3}$**–1–0**	$v_{1,4}$	2900	0	$dep_1 := 2900$
s_3–1–1	$v_{2,4}$	1600	1	$dep_2 := 400$
s_3–1–4	$v_{1,3}$	3300	1	$dep_3 := 375$
s_3–1–8	$v_{2,3}$	2400	1	$dep_5 := 600$
Total	——	——	——	$dep_{s_3-1} := 4275w_1$

Step s_1–2

The amount of available sediment in s_1 before this step is $2576.5625w_1$.

Spread Visit	Target t	$cap(t)$	Distance (d)	Deposit ($cap(t) \cdot f^{-d}$)
$\mathbf{s_1}$**–2–0**	$v_{2,1}$	562.5	0	$dep_1 := 562.5$
s_1–2–1	$v_{3,1}$	2800	1	$dep_2 := 700$
s_1–2–2	$v_{2,2}$	1856.25	1	$dep_3 := 464.0625$
s_1–2–3	$v_{1,1}$	0	1	$dep_4 := 0$ ($v_{1,1}$ is full)
s_1–2–5	$v_{3,2}$	3400	1	$dep_5 := 850$
s_1–2–6	$v_{1,2}$	0	1	$dep_6 := 0$ ($v_{1,2}$ is full)
Total	——	——	——	$dep_{s_1-2} := 2576.0625w_1$

Streamline s_1 is now empty.

Step s_2–2

The amount of available sediment in s_2 before this step is $1742.1875w_1$.

Spread Visit	Target t	$cap(t)$	Distance (d)	Deposit ($cap(t) \cdot f^{-d}$)
$\mathbf{s_2}$**–2–0**	$v_{2,2}$	1392.1875	0	$dep_1 := 1392.1875$
s_2–2–1	$v_{3,2}$	2550	1	$dep_2 := 350$
Total	——	——	——	$dep_{s_2-2} := 1742.1875w_1$

Streamline s_2 is now empty.

Step s_3–2

The amount of available sediment in s_3 before this step is $400w_1$.

Spread Visit	Target t	$cap(t)$	Distance (d)	Deposit ($cap(t) \cdot f^{-d}$)
$\mathbf{s_3}$**–2–0**	$v_{2,4}$	1200	0	$dep_1 := 400$
Total	——	——	——	$dep_{s_3-2} := 400w_1$

Streamline s_3 is now empty.

5.3.2 Forward w_2 run

For sediment w_2, streamlines s_1, s_2, and s_3 possess, respectively, $1200w_2$, $0w_2$, and $400w_2$ sediment units. Since s_2 is empty, we will ignore its steps.

Step s_1–1

This vertex is full.

Step s_3–1

This vertex is full.

Step s_1–2

This vertex is full.

Step s_3–2

The amount of available sediment in s_3 before this step is $400w_2$.

Spread Visit	Target t	$cap(t)$	Distance (d)	Deposit $(cap(t) \cdot f^{-d})$
s₃–2–0	$v_{1,4}$	800	0	$dep_1 := 400$
Total	——	——	——	$dep_{s_3-2} := 400w_2$

Streamline s_3 is now empty.

Step s_1–3

This vertex is full.

Step s_1–4

The amount of available sediment in s_1 before this step is $1200w_2$.

Spread Visit	Target t	$cap(t)$	Distance (d)	Deposit $(cap(t) \cdot f^{-d})$
s₁–4–0	$v_{3,2}$	2200	0	$dep_1 := 1200$
Total	——	——	——	$dep_{s_1-4} := 1200w_2$

Streamline s_1 is now empty.

5.3.3 Forward w_3 run

For sediment w_3, streamlines s_1, s_2, and s_3 possess, respectively, $0w_3$, $0w_3$, and $1900w_2$ sediment units. Since s_1 and s_2 are empty, we will ignore their steps.

Step s_3–1

This vertex is full.

Step s_3–2

The amount of available sediment in s_3 before this step is $1900w_2$.

Spread Visit	Target t	$cap(t)$	Distance (d)	Deposit $(cap(t) \cdot f^{-d})$
$\mathbf{s_3}$–2–0	$v_{2,4}$	400	0	$dep_1 := 400$
Total	——	——	——	$dep_{s_3-2} := 400w_3$

Step s_3–3

The amount of available sediment in s_3 before this step is $1500w_2$.

Spread Visit	Target t	$cap(t)$	Distance (d)	Deposit $(cap(t) \cdot f^{-d})$
$\mathbf{s_3}$–2–0	$v_{2,4}$	2000	0	$dep_1 := 1500$
Total	——	——	——	$dep_{s_3-3} := 1500w_3$

Streamline s_3 is now empty.

All demands have been successfully satisfied with the values generated by the backward simulation algorithm.

5.4 Complexity Estimate

The complexity of the backward simulation algorithm can be measured in terms of the *number of visitors* at each Path Step. The worst case scenario for this particular measurement is if all streamlines are present within the Moore neighbourhood $N(v, r)$, where r is the maximum spread radius, and v is the current vertex visited by the Path Step.

Proposition 5.4.0.1. $|N(v, r)| = (2r + 1)^2$

Proof. We prove by induction on r. The property is clearly satisfied for $r = 0$. For $r = 1$, there are 9 vertices including v, as shown in Figure 4.2. $|N(v, 1)| = 9$.

Assume that $|N(v, r)| = (2r + 1)^2$ holds for $r \leq z$.

For $r = z + 1$, we have:

$$|N(v, z + 1)| = |N(v, z)| + 4(2z + 1) + 4$$

$$|N(v, z + 1)| = |N(v, z)| + 8z + 8$$

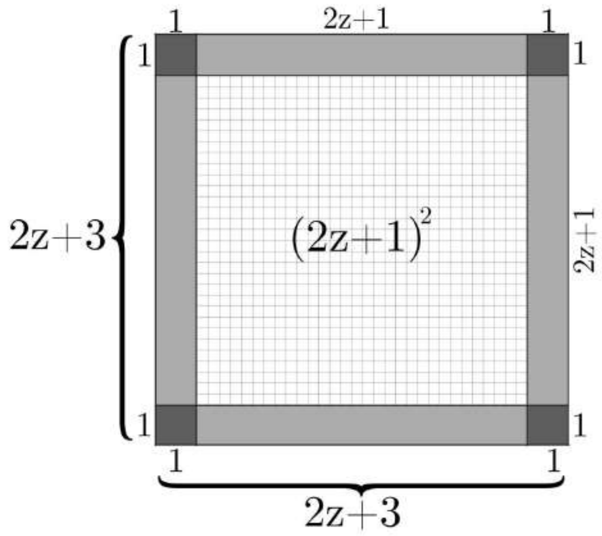

Figure 5.8: Growth of a Moore Neighbourhood $N(v, z+1)$

By the induction hypothesis, we can conclude (see Figure 5.8):

$$|N(v, z+1)| = (2z+1)^2 + 8z + 8$$

$$|N(v, z+1)| = 4z^2 + 12z + 9$$

$$|N(v, z+1)| = (2z+3)^2$$

$$|N(v, z+1)| = (2(z+1)+1)^2$$

□

Considering this result and following the definitions of sediment spread in Section 4.3, we know that sediment spread happens after the last Path Step deposits, fulfils, and exceeds the maximum capacity of the origin vertex. Therefore, for every vertex in $N(v, r)$, there is at most one Path Step that will visit and possibly deposit to v. Thus, the maximum number of valid visitors per Path Step is $|N(v, r)| = (2r+1)^2$.

Following Definition 4.1.4, the length of each streamline is measured in Path Steps. Let us assume that, for the worst case, all streamlines have the same maximum length l. If there are s streamlines with length l, we have a maximum number of valid visitors defined by $s \cdot l \cdot (2r+1)^2$. Removing the constants, we can conjecture that the worst-case time complexity is

$$O(s \cdot l \cdot r^2).$$

Chapter 6

Conclusion and Future Directions

This work presented a new contribution to the problem of stratigraphic inversion. There are known approaches that make use of optimisation methods to empower simulators to find the adequate sediment supply values for the forward simulation. These methods have been thoroughly validated and tested throughout the industry and the community.

We introduced an algorithm for sediment transfer over two dimensions. This algorithm empirically reproduces a similar behaviour to the progression of sediment spread according to basic principles of deposition [10]. Sediment priority is respected, and sediment spread can be simulated with various radii and exponential factors for increased detail and realism. Close variants of this algorithm are applied in some stratigraphic simulators in industry [39].

Many approaches in literature (see Chapter 3) attempt to attack the input space with advanced search and optimisation methods, then running the forward algorithm and converging the result towards the smallest possible error. This approach is algorithm-agnostic, since it is only bounded by an error function that does not depend on any particular implementation. However, the computational cost for optimisation methods rapidly increase with the number of variables. In a simulation grid with 250,000 cells, inverting values for 10 sediments for 1,000 streamlines with over 200 steps each, the amount of variables to be accounted for can exceed the computational resources available. Simulations then become either too lengthy or devoid of resolution.

Observing the problem as traversing separate paths in graphs and updating weights in vertices can yield a different solution. Our inversion algorithm presented in Chapter 5 is, in essence, the dual of our forward algorithm presented in Chapter 4. Nonetheless, one must notice that the concept of duality does not imply in the same level of simplicity as the original algorithm. A very simplified version is described in Algorithm 5.1, stripped down of many implementation details and data structures in order to be presented comprehensively.

The inherent difficulty in inverting Algorithm 4.1 lies in the process of sediment spread. Each path is not only affecting their own vertices, but also affecting neighbouring vertices and changing their maximum capacities. That, consequently, affects the demands of all other paths that attempt to deposit in that very same vertex. The inversion algorithm, then, must be capable of calculating if a said path will perform a visit to that particular vertex, and what amounts of sediments should be considered for every possible streamline.

Thus, one can identify the presence of multiple complex structures that rely on a long and interdependent ordering to establish the correct amounts throughout the inverse traversal of the streamlines. Therefore, the concept of *visitors* demonstrates its usefulness as to unify all the different orderings into one single abstraction.

The algorithm is presented in the form of a conjecture. It is believed that this algorithm is correct. However, there might be extreme instances and corner cases against which this algorithm was not tested. The very nature of the problem allows for exceedingly complex and unwieldy instances, some of them yet unconceived by the author. Those are not relevant to the practical applications of the problem, but are worth testing to evaluate the soundness of the conjecture. Further formal investigations around the correctness and invariants are beyond the scope of this work.

Our first main contribution is to provide a different perspective on the approach of inversion and optimisation problems. Understanding the structures of forward problems can yield interesting results. This method is, to this date, the first of its kind to provide stratigraphic inversion by using an algorithm as the inverse function of a problem. Also, we provide the capability for *exact* inversion values, instead of approximative solutions.

Secondly, the level of descriptiveness provided by both forward and backward procedures in this work is a strong contribution on itself, as it allows the community to reproduce step-by-step processes for stratigraphic inversion in a finer level of detail. It is not widespread practice to delve unto the mechanics of sediment transfer and inversion - at least algorithmically. Most methods in the literature are described in a higher level of abstraction.

Thirdly, the estimated complexity of this algorithm is polynomial, with low degrees, which is a good indication of efficiency for such a complex problem. It is not possible to provide any comparisons with other methods, since there are no common datasets or implementation details of the methods in literature. It is important to point out that most details about implementations in this field are highly protected in industrial secrecy.

Next versions of this stratigraphic simulation model can account for added layers of complexity. With variable velocity streamlines, we can evaluate the action for the sediments at every Path Step according to the behaviour of particle transport as shown by the Hjulström curve (see Figure 2.4), thus deciding if there happens a deposition, bypassing (no deposition), or erosion (negative deposition). Simulating erosion could be accomplished by the concept of negative demands, or visitors with negative deposit amounts.

Future works could also provide a performance and accuracy benchmark of different approaches for stratigraphic inversion, something that has not yet been seen in current literature.

Bibliography

[1] VON LINNÉ, C. *Systema naturae per regna tria naturae secundum classes, ordines, genera, species,...*, v. 1. Impensis Georg. Emanuel Beer, 1788.

[2] FALIVENE, O., FRASCATI, A., GESBERT, S., et al. "Automatic calibration of stratigraphic forward models for predicting reservoir presence in exploration", *AAPG Bulletin*, v. 98, n. 9, pp. 1811–1835, 2014.

[3] PAOLA, C. "Quantitative models of sedimentary basin filling", *Sedimentology*, v. 47, n. s1, pp. 121–178, 2000.

[4] PYRCZ, M. J., DEUTSCH, C. V. *Geostatistical reservoir modeling.* Oxford university press, 2014.

[5] NICHOLS, G. *Sedimentology and stratigraphy.* John Wiley & Sons, 2009.

[6] WENTWORTH, C. K. "A scale of grade and class terms for clastic sediments", *The Journal of Geology*, v. 30, n. 5, pp. 377–392, 1922.

[7] CATUNEANU, O. *Principles of sequence stratigraphy.* Elsevier, 2006.

[8] HAQ, B. U., HARDENBOL, J., VAIL, P. R. *Mesozoic and Cenozoic chronostratigraphy and cycles of sea-level change.* Special Publications of SEPM, 1988.

[9] KRAVITZ, G. "The Geohistorical Time Arrow: From Steno's Stratigraphic Principles to Boltzmann's Past Hypothesis", *Journal of Geoscience Education*, v. 62, n. 4, pp. 691–700, 2014.

[10] HJULSTROM, F. *Transportation of detritus by moving water: Part 1. Transportation.* AAPG Special Volumes, 1939.

[11] GEANKOPLIS, C. *Transport processes and separation process principles (includes unit operations).* Prentice Hall Press, 2003.

[12] HAQ, B. U., HARDENBOL, J., VAIL, P. R., et al. "Chronology of fluctuating sea levels since the Triassic", *Science*, v. 235, n. 4793, pp. 1156–1167, 1987.

[13] KENDALL, C. G. S. C., LERCHE, I. *The rise and fall of eustasy.* Special Publications of SEPM, 1988.

[14] BURTON, R., KENDALL, C. G. S. C., LERCHE, I. "Out of our depth: on the impossibility of fathoming eustasy from the stratigraphic record", *Earth-Science Reviews*, v. 24, n. 4, pp. 237–277, 1987.

[15] CANTRELL, C. D. *Modern mathematical methods for physicists and engineers.* Cambridge University Press, 2000.

[16] SCHIFF, J. "Two-Dimensional Automata". In: *Introduction to cellular automata*, cap. 4, University of Wisconsin, 2005.

[17] SMITH, A. R. "Two-dimensional formal languages and pattern recognition by cellular automata". In: *Switching and Automata Theory, 1971., 12th Annual Symposium on*, pp. 144–152. IEEE, 1971.

[18] CHANG, G. J. "Algorithmic aspects of domination in graphs", *Handbook of Combinatorial Optimization*, pp. 221–282, 2013.

[19] SKIENA, S. *Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica.* Boston, MA, USA, Addison-Wesley Longman Publishing Co., Inc., 1991. ISBN: 0-201-50943-1.

[20] LYNCH, N., VAANDRAGER, F. "Forward and backward simulations", *Information and Computation*, v. 121, n. 2, pp. 214–233, 1995.

[21] VAN RIJN, L. C. "Sediment transport, part I: bed load transport", *Journal of hydraulic engineering*, v. 110, n. 10, pp. 1431–1456, 1984.

[22] MORGAN, R., QUINTON, J., SMITH, R., et al. "The European Soil Erosion Model (EUROSEM): a dynamic approach for predicting sediment transport from fields and small catchments", *Earth surface processes and landforms*, v. 23, n. 6, pp. 527–544, 1998.

[23] WU, W., RODI, W., WENKA, T. "3D numerical modeling of flow and sediment transport in open channels", *Journal of Hydraulic Engineering*, v. 126, n. 1, pp. 4–15, 2000.

[24] MERRITT, W. S., LETCHER, R. A., JAKEMAN, A. J. "A review of erosion and sediment transport models", *Environmental Modelling & Software*, v. 18, n. 8, pp. 761–799, 2003.

[25] LEE, Y.-H., HARBAUGH, J. W. "Stanford's sedsim project: Dynamic three-dimensional simulation of geologic processes that affect clastic sediments". In: *Computer Graphics in Geology*, Springer, pp. 113–127, 1992.

[26] LAWRENCE, D. T., DOYLE, M., AIGNER, T. "Stratigraphic Simulation of Sedimentary Basins: Concepts and Calibration (1)", *Aapg Bulletin*, v. 74, n. 3, pp. 273–295, 1990.

[27] AIGNER, T., BRANDENBURG, A., VAN VLIET, A., et al. "Stratigraphic modelling of epicontinental basins: two applications", *Sedimentary Geology*, v. 69, n. 3-4, pp. 167–190, 1990.

[28] CARRON, D. "Well guided stratigraphic inversion of borehole and surface seismic sections". In: *SEG Technical Program Expanded Abstracts 1988*, Society of Exploration Geophysicists, pp. 837–840, 1988.

[29] FABRE, N., GLUCK, S., GUILLAUME, P., et al. "Robust multichannel stratigraphic inversion of stacked seismic traces". In: *SEG Technical Program Expanded Abstracts 1989*, Society of Exploration Geophysicists, pp. 943–945, 1989.

[30] LESSENGER, M. A., CROSS, T. A. "A stratigraphic inverse simulation model", *AAPG Bulletin (American Association of Petroleum Geologists);(United States)*, v. 75, n. CONF-910403–, 1991.

[31] COOKE, D. A., SCHNEIDER, W. A. "Generalized linear inversion of reflection seismic data", *Geophysics*, v. 48, n. 6, pp. 665–676, 1983.

[32] LESSENGER, M. A., CROSS, T. A. "An inverse stratigraphic simulation model–is stratigraphic inversion possible?" *Energy exploration & exploitation*, v. 14, n. 6, pp. 627–637, 1996.

[33] BORNHOLDT, S., WESTPHAL, H. "Automation of stratigraphic simulations: Quasi-backward modelling using genetic algorithms", *Geological Society, London, Special Publications*, v. 134, n. 1, pp. 371–379, 1998.

[34] CROSS, T. A., LESSENGER, M. A. "Construction and application of a stratigraphic inverse model", 1999.

[35] SHARMA, A. K. *Quantitative stratigraphic inversion*. Tese de Doutorado, Virginia Tech, 2006.

[36] SHARMA, A. K., IMHOF, M. G. "Quantitative stratigraphic inversion: numerical study". In: *SEG Technical Program Expanded Abstracts 2007*, Society of Exploration Geophysicists, pp. 1500–1504, 2007.

[37] SAMBRIDGE, M. “Geophysical inversion with a neighbourhood algorithm—II. Appraising the ensemble”, *Geophysical Journal International*, v. 138, n. 3, pp. 727–746, 1999.

[38] LOPES, A. A. O., FACCION, J. E., CARVALHO, C. V. A., et al. “Simulação Numérica para o Estudo da Formação de Bacias Sedimentares Usando Modelagem Estratigráfica”, *Proceedings of the XXVIII Iberian Latin American Congress on Computational Methods in Engineering*, pp. 17–18, 2006.

[39] DUMONT, N. A., LOPES, A. A. O., AGUILAR, C. A., et al. “On a depth-averaged numerical simulation of flow and transport phenomena in sedimentary basins”, *CILAMCE XXXVI Iberian Latin-American Congress on Computational Methods in Engineering*, pp. 1–17, 2015.