



A SHALLOW WATER EVENT-DRIVEN APPROACH TO SIMULATE
TURBIDITY CURRENTS AT STRATIGRAPHIC SCALE

Túlio Ligneul Santos

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia Civil, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia Civil.

Orientador: Alvaro Luiz Gayoso de Azeredo
Coutinho

Rio de Janeiro
Abril de 2020

A SHALLOW WATER EVENT-DRIVEN APPROACH TO SIMULATE
TURBIDITY CURRENTS AT STRATIGRAPHIC SCALE

Túlio Ligneul Santos

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA CIVIL.

Orientador: Alvaro Luiz Gayoso de Azeredo Coutinho

Aprovada por: Prof. Alvaro Luiz Gayoso de Azeredo Coutinho

Prof. Fernando Alves Rochinha

Prof. Jorge Hugo Silvestrini

Prof. Luiz Landau

Prof. Renato Nascimento Elias

RIO DE JANEIRO, RJ – BRASIL

ABRIL DE 2020

Santos, Túlio Ligneul

A shallow water event-driven approach to simulate turbidity currents at stratigraphic scale/Túlio Ligneul Santos. – Rio de Janeiro: UFRJ/COPPE, 2020.

XXIII, 145 p.: il.; 29,7cm.

Orientador: Alvaro Luiz Gayoso de Azeredo Coutinho
Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia Civil, 2020.

Referências Bibliográficas: p. 112 – 127.

1. Stabilized finite elements. 2. Flux-corrected transport. 3. Shallow water model. 4. Turbidity currents. 5. Event-driven simulation. 6. Sedimentation. 7. Stratigraphy. I. Coutinho, Alvaro Luiz Gayoso de Azeredo. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Civil. III. Título.

To my grandfather Renato.

Acknowledgments

First, I would like to thank my grandfather Renato for always encouraging me to pursue a career in engineering and technology and to further engage in the academic world. I am also extremely grateful to my parents, Gisele and José Bento, for their encouragement and guidance throughout my life. A special thanks to my brother, Erick, for the union and life example. To my friends Camila Rebello, Erick Tavares, Jorge Valdivia, Guilherme Martins, Gabriel Portugal and Thiago Xavier for the support and companionship received over the last years. Above all, I must thank my wife Raisa for her love and for being incredibly supportive during the extent of this endeavor.

I am grateful to my thesis advisor Alvaro Coutinho for his guidance and teachings during my doctoral studies and the execution of this work. I would also like to express my sincere gratitude to the professors Fernando Rochinha, Jorge Hugo Silvestrini, Luiz Landau and Renato Nascimento for their presence at the examining committee. I am also gratefully indebted to the other teachers throughout my academic education for their contribution to my learning and self-development.

A very special gratitude goes out to all down at PetroSoft for the technical and moral support in my professional life, especially to Alexandre Lopes for the continuous encouragement and clear trust in my work. I am also grateful to Petrobras. Special thanks to the geologist José Eduardo Faccion for sharing his knowledge and experience in geology and stratigraphy, and to the former PROMOB coordinator Marco Moraes for the opportunity to develop this thesis.

And finally, last but by no means least, to the Brazilian people who contributed significantly to my education and stay at this University. This thesis is a small way to repay the investment and trust in me.

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

UMA ABORDAGEM DE ÁGUAS-RASAS ORIENTADA A EVENTOS PARA SIMULAR CORRENTES DE TURBIDEZ EM ESCALA ESTRATIGRÁFICA

Túlio Ligneul Santos

Abril/2020

Orientador: Alvaro Luiz Gayoso de Azeredo Coutinho

Programa: Engenharia Civil

Apresenta-se uma nova abordagem orientada a eventos para simular a formação de sistemas deposicionais turbidíticos em escala estratigráfica. Combina-se um modelo de escoamento baseado na aproximação de águas-rasas com um algoritmo de sedimentação prático. Equações que governam correntes de turbidez são resolvidas usando um novo esquema de transporte corrigido por fluxo embasado no método dos elementos finitos. Sua formulação de baixa ordem é construída adicionando, à expressão usual obtida pelo método de Galerkin, um componente difusivo semelhante à dissipação escalar de Rusanov, escalado por um operador de captura de choques. Obtém-se o sistema de alta ordem incluindo fluxos anti-difusivos linearizados em torno da solução de baixa ordem e limitados pelo procedimento de Zalesak, seguindo um pré-limitador minmod. A integração temporal é feita implicitamente com passos de tempo adaptativos, e usando um algoritmo não-linear iterativo que lineariza os termos fonte. A sedimentação é realizada carregando cinco frações granulométricas através de linhas de emissão evoluídas durante o escoamento e espalhando sedimentos radialmente. Estes depositam, preenchendo o espaço disponível e compactando as camadas inferiores. O escoamento é calculado enquanto a descarga de um evento estiver ativa, continuando até a corrente atingir um estado de equilíbrio. Em seguida, a etapa de deposição do evento é executada. Resultados do simulador do escoamento apresentaram boa concordância com soluções exatas e bibliográficas disponíveis. A estratégia adotada também demonstrou melhor desempenho em relação à formulação de elementos finitos estabilizados testada. Além disso, os depósitos de sedimentos reproduzidos sugerem que a abordagem proposta é adequada para realizar simulações em escala estratigráfica.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

A SHALLOW WATER EVENT-DRIVEN APPROACH TO SIMULATE
TURBIDITY CURRENTS AT STRATIGRAPHIC SCALE

Túlio Ligneul Santos

April/2020

Advisor: Alvaro Luiz Gayoso de Azeredo Coutinho

Department: Civil Engineering

We present a new event-driven approach that combines a shallow water flow model with a practical sedimentation technique to simulate the formation of turbidite depositional systems at a stratigraphic scale. Equations that govern turbidity currents dynamics are solved using a new finite element flux-corrected transport scheme. In this sense, the low-order formulation is built by adding a novel Rusanov-like scalar dissipation scaled by a shock-capturing operator to standard Galerkin equations. From it, the high-order system is obtained by including anti-diffusive fluxes linearized around the low-order solution and limited with the Zalesak's algorithm, following a minmod prelimitter. Implicit time integration with adaptive time steps is performed with an iterative nonlinear scheme that linearizes source terms. Sedimentation is implemented by carrying five granulometric fractions (clay, silt, and fine, medium and coarse sands) along evolved streaklines and radially scattering sediments that deposit filling the available space and compacting the underneath sediment layers. The flow is computed while an event discharge into an area of interest is active, or the inflow current has not reached an equilibrium state. Afterward, the event deposition step is executed. Numerical results of our flow solver presented a good agreement with available exact and literature solutions. It is also compared with a stabilized finite element formulation, producing better outcomes, especially in scenarios with complex drying/wetting dynamics. Also, the simulated sediment deposits suggest that our approach is well suited for stratigraphic scale simulations.

Contents

List of Figures	x
List of Symbols	xix
1 Introduction	1
2 Flow Simulation	6
2.1 Governing Equations	6
2.1.1 Boundary Conditions	7
2.1.2 Scale Considerations	8
2.1.3 Mass Conservation Equation	11
2.1.4 Momentum Conservation Equation	12
2.1.5 Bed Friction Forces	14
2.1.6 Equation System in Convection-Diffusion Form	16
2.2 Initial Value Problem	17
2.3 Stabilized Formulation	21
2.4 FCT Scheme	25
2.4.1 Positivity Preservation	25
2.4.2 Low-Order Method	31
2.4.3 Source Term Linearization	33
2.4.4 High-Order Method	33
2.4.5 Flux Limiting	35
2.5 Adaptive Time Step Control	36
2.6 Dry/Wet Handling	37
3 Event-Driven Simulation	40
3.1 Recurrent Events	40
3.2 Streakline Computation	42
3.3 Mesh Generation	45
3.4 Sedimentation	48
4 Implementation Aspects	58

5	Numerical Results	64
5.1	1D Dam Break	64
5.2	Transcritical Flow with a Shock	66
5.3	1D Flow over an Irregular Bed	68
5.4	Asymmetric Dam Break	70
5.5	Dam Break over a Channel with Bumps	73
5.6	Inflow from a Sloped Channel	77
5.7	Turbidity Current Flow over an Irregular Bed	78
5.8	Sedimentation on a Flat Bed	82
5.9	Sedimentation on a Sloping Bed	89
5.10	Sedimentation on an Irregular Terrain	96
6	Conclusions and Future Work	108
	Bibliography	112
A	Useful Definitions	128
A.1	Derivatives at the Flux and Transformation Jacobian Matrices	128
A.2	Matrix Vectorization	129
A.3	Kronecker Product	129
B	Hypotheses Evaluations	131
B.1	SPD Jacobian Matrix of the Transformation from Conservation to Entropy Variables	131
B.2	Mass-Preserving Lumped Mass Matrix	132
B.3	Zero Row-Sum Property of the Generalized Stiffness Matrix	133
B.4	Non-Negative Diagonal Entries of Positive Definite Matrices	134
C	Parametric Streakline Curves	135
D	Predictor/Multi-corrector Algorithm	136
E	Exact Solution of the Transcritical Flow with a Shock Problem	139

List of Figures

2.1	Schematic representation of the adopted shallow water model. It shows a turbidity current with density ρ and thickness $h(x, y, t)$ that flows over a terrain, whose elevation is $z_b(x, y)$, and advances through a motionless fluid of density ρ_w	7
2.2	Representation of a non-zero vector \mathbf{v} and its transformation $\mathbf{P}\mathbf{v}$ using the positive definite matrix \mathbf{P}	29
2.3	Representations of the fluid's free surface in a 1D element with nodes i and j . (a) Real physical state; (b) Linear finite element representation; (c) Corrected linear finite element representation.	38
3.1	Streakline segment $\overline{\mathbf{p}_1\mathbf{p}_2}$ and adjacent points \mathbf{p}_0 and \mathbf{p}_3	42
3.2	The circle circumscribed in the triangle with vertices \mathbf{p}_1 , \mathbf{p}_2 and \mathbf{p}_3 that is used to compute the curvature at \mathbf{p}_2	43
3.3	Streakline segment, the associated central angle and a inscribed angle.	44
3.4	Example of an initial depositional surface (in brown) with the starting water level (in blue) at the external stratigraphic application.	46
3.5	Detail of the domain's contour in the original (a) and generated (b) meshes. The black dots in (a) are the control points used to build the red B-splines in (b).	47
3.6	Mesh generated for the underwater region illustrated in Figure 3.4.	47
3.7	(a) Example of a simulation scenario on the stratigraphic grid. The brown cells are dry while the light blue cells are underwater. Also, we identify two green coastlines, a blue open boundary, and a pink border that takes event inflows. (b) Mesh produced for the setup in (a). The B-splines in red were created using the black dots as control points.	48
3.8	Contents of the Gmsh script file used to create the mesh in Figure 3.7b. We highlight the definition of the domain's boundaries with the same colors used in Figure 3.7a: coast boundaries in green , open borders in blue , and contours with events in pink	49

3.9	Different parasequence stacking patterns. Here the numbers denote the distinct parasequences in a set. (a) Retrogradational parasequence set; (b) Aggradational parasequence set; (c) Progradational parasequence set. Adapted from WAGONER <i>et al.</i> [110].	50
3.10	Representation of the order in which we advance one step along each streakline. This sequence goes from the outermost to the innermost lines and is given by the numbers beside each streakline source in yellow . Here the cells with the blue highlight are associated with the same sediment supply.	53
3.11	Schematic representation of a possible result of the sedimentation of three events using three sub-steps per event.	55
3.12	Representation of the first steps of the flood-fill algorithm employed to scatter sediments from the hatched grid cell. The green color denotes the cells that were visited by the algorithm and the yellow color stands for the cells waiting in the queue to be visited. The numbers show the order in which the cells are to be visited.	56
4.1	Example of a hanging node (in red) created from the refinement of the original left cell. While it belongs to the elements on its left side, it is not part of the element on the right. The nodes in blue define the coarse elements, and the green nodes were also created by the refinement step.	59
5.1	Color table used to represent the distinct granulometric fractions (and the associated lithology types).	64
5.2	Schematic representations of the 1D dam break problem (a) and its analytical solution (b). The bold numbers represent the different regions of the initial state and the analytical solution.	65
5.3	Exact and simulated solutions at $t = 7.5$ s of the 1D dam break problem. (a) Fluid height; (b) Water velocity.	66
5.4	Exact and simulated water elevation solutions of the transcritical flow with shock problem.	68
5.5	Water surface obtained at key simulation times of the 1D flow over an irregular bed problem. The results of the FCT approach are in red and the ones obtained using the CAU and $YZ\beta$ operators are, respectively, in green and blue	69
5.6	Initial configuration of the asymmetric dam break problem. (a) Domain's geometry and initial fluid height distribution; (b) Detail of the mesh and the initial water heights near the dam.	71

5.7	3D and map views of the water surface with height contours obtained at $t = 7.2$ s of the asymmetric dam break problem. The ones with the green and blue borders were computed with the stabilized approach using, respectively, the CAU and $YZ\beta$ shock-capturing operators. Those with the red borders are outcomes of the FCT scheme.	71
5.8	Computed and reference solutions for the asymmetric dam break problem plotted along two sections at $t = 7.2$ s. In (a), the reference is the result of RICCHIUTO [129] at $y = 132$ m. In (b), the reference is the solution of RICCHIUTO <i>et al.</i> [128] at $y = 160$ m. . .	72
5.9	Detail of the specific discharge vectors obtained near the channel at $t = 7.2$ s of the asymmetric dam break problem. The maps with the green and blue borders were computed using the CAU and $YZ\beta$ operators while the one with the red border was generated by the FCT method.	73
5.10	3D view of the initial state (a) and 2D map of the simulation mesh (b) of the dam break over a channel with bumps problem.	74
5.11	Map views of the obtained water heights at key simulation times of the dam break over a channel with bumps problem. The green and blue borders indicate results computed with the stabilized approach using, respectively, the CAU and $YZ\beta$ shock-capturing operators. The maps with red borders are outcomes of the FCT scheme.	75
5.12	Water surface elevation along a section at $y = 15$ m and $t = 300$ s (a) and detail of the free surface near the highest bump (b) of the dam break over a channel with bumps problem.	75
5.13	3D and map views at the same times illustrated by LIANG and BORTHWICK [98] of the FCT solution for the dam break over a channel with bumps problem.	76
5.14	Geometric representation of the domain studied in the inflow from a sloped channel problem. (a) Domain's shape and lengths; (b) Detail of a vertical slice over the red dashed line in (a), showing the channel slope and the water inflow direction.	77
5.15	Detail of the position of the streakline sources (dots in yellow) at the start of the channel of the inflow from a sloped channel problem.	77
5.16	Solution at $t = 1.8$ s of the inflow from a channel problem, showing the streaklines in white and their individual points in yellow . (a) 3D view of the bed and water surfaces, and the streaklines; (b) Detail of the distribution of points along the streaklines near the end of the channel.	78

5.17	States of the streaklines computed by the ParaView [119] application (lines in pink) and our approach (lines in white and points in yellow) at $t = 6.19$ s of the inflow from a sloped channel problem. Computations were performed using two meshes: one with low-resolution (a) and another one with higher resolution (b).	79
5.18	State of the evolved streaklines at $t = 13$ s of the inflow from a sloped channel problem.	79
5.19	Initial configuration of the turbidity current flow over an irregular bed problem. (a) 3D view of the seabed indicating the entry of the Almirante Câmara canyon. (b) Detailed view of the mesh and the inflow discharges near the boundary by the canyon's entry, and the streakline sources (dots in yellow).	80
5.20	Simulated turbidity current at key times of the turbidity current flow over an irregular bed problem. The pink lines in middle of the current portray the computed streaklines.	81
5.21	Location of the sediment supply used in the sedimentation on a flat bed problem. Here, the columns beside the boundary cells highlighted in blue represent the sediment supplies.	82
5.22	Map views of the top of the resulting deposits after each event of the sedimentation on a flat bed problem, when solved using a single sub-step per event and $f_s = 1$. The maps (a), (b) and (c) refer to the state of the bed after the first, second and last events.	83
5.23	Map views of the top of the resulting deposits after each event of the sedimentation on a flat bed problem, when solved using a single sub-step per event. The maps (a), (b) and (c) refer to the state of the bed after the first, second and third events.	83
5.24	Map views of the sedimentary bed after each event and sub-step of the sedimentation on a flat bed problem, when solved using three sub-steps per event. (a), (b) and (c) represent depositions at each event simulation, being (a) associated with the first event and (c) with the last. The progression of the maps from left to right denote the sequence of sub-steps in an event's deposition.	84
5.25	View, along a vertical section parallel to the x-direction and placed in the middle of the simulated region, of the stacking pattern produced at the sedimentation on a flat bed problem, when solved using one sub-step and $f_s = 1$. (a) Original section in terms of the deposited lithology blocks; (b) Smoothed section. Here the black lines separate the strata of each event.	85

5.26	View, along a vertical section parallel to the x-direction and placed in the middle of the simulated region, of the stacking pattern produced at the sedimentation on a flat bed problem, when solved using one sub-step. (a) Original section in terms of the deposited lithology blocks; (b) Smoothed section. Here the black lines separate the strata of each event.	85
5.27	Maps of the thickness of the accumulated sediment deposits after each event of the sedimentation on a flat bed problem, when solved using one sub-step per event. The maps on the left depict the result after the first event, while the maps at the middle and the right columns respectively represent the states after the second and third events. Each maps row shows the accumulated thickness of the different components: (a) All lithology types; (b) Coarse sand strata; (c) Medium sand deposits; (d) Silt material.	86
5.28	View, along a vertical section parallel to the x-direction and placed in the middle of the simulated region, of the stacking pattern produced at the sedimentation on a flat bed problem, when solved using three sub-steps. (a) Original section in terms of the deposited lithology blocks; (b) Smoothed section. Here the black lines separate the strata of each event.	87
5.29	View, along a vertical section parallel to the x-direction and placed in the middle of the simulated region, of the stacking pattern produced at the sedimentation on a flat bed problem, when solved using five lithology types and three sub-steps. (a) Original section in terms of the deposited lithology blocks; (b) Smoothed section. Here the black lines separate the strata of each event.	87
5.30	Maps of the thickness of the accumulated sediment deposits after each event of the sedimentation on a flat bed problem, when solved using three sub-steps per event. The maps on the left depict the result after the first event, while the maps at the middle and the right columns respectively represent the states after the second and third events. Each maps row shows the accumulated thickness of the different components: (a) All lithology types; (b) Coarse sand strata; (c) Medium sand deposits; (d) Silt material.	88
5.31	Schematic representation of the region simulated in the sedimentation on a sloping bed problem.	89

5.32	Location of the sea level and the sediment supply used to simulate each turbidite event of the sedimentation on a sloping bed problem. The arrangements (a), (b) and (c) were used to simulate, respectively, the first, second and last events. Here, the columns beside the boundary cells highlighted in blue represent the sediment supplies.	90
5.33	Map views of the resulting deposits after each event of the sedimentation on a sloping bed problem, when solved using a single sub-step per event. The maps (a), (b) and (c) refer to the state of the bed after the first, second and last events.	91
5.34	View, along a vertical section parallel to the x -direction and placed in the middle of the simulated region, of the stacking pattern produced at the sedimentation on a sloping bed problem, when solved using a single sub-step per event. (a) Original section; (b) Section with scaled deposit thicknesses. Here the different types of black lines separate the strata of each event.	91
5.35	Maps of the thickness of the accumulated sediment deposits after each event of the sedimentation on a sloping bed problem, when solved using a single sub-step per event. The maps on the left depict the result after the first event, while the maps at the middle and the right columns respectively represent the states after the second and third events. Each maps row shows the accumulated thickness of the different components: (a) All lithology types; (b) Coarse sand strata; (c) Medium sand deposits; (d) Silt material.	92
5.36	Map views of the sedimentary bed after each event and sub-step of the sedimentation on a sloping bed problem, when solved using three sub-steps per event. (a), (b) and (c) represent depositions at each event simulation, being (a) associated with the first event and (c) with the last. The progression of the maps from left to right denote the sequence of sub-steps in an event's deposition.	93
5.37	View, along a vertical section parallel to the x -direction and placed in the middle of the simulated region, of the stacking pattern produced at the sedimentation on a sloping bed problem, when solved using three sub-steps. (a) Original section; (b) Section with scaled deposit thicknesses. The thinnest black lines represent the limits of a sub-step deposit, while the different types of thicker black lines separate the strata of each event.	93

5.38	Maps of the thickness of the accumulated sediment deposits after each event of the sedimentation on a sloping bed problem, when solved using three sub-steps per event. The maps on the left depict the result after the first event, while the maps at the middle and the right columns respectively represent the states after the second and third events. Each maps row shows the accumulated thickness of the different components: (a) All lithology types; (b) Coarse sand strata; (c) Medium sand deposits; (d) Silt material.	94
5.39	View, along a vertical section parallel to the x -direction and placed in the middle of the simulated region, of the stacking pattern produced at the sedimentation on a sloping bed problem, when solved using five lithology types and three sub-steps. (a) Original section. (b) Smoothed section with scaled deposit thicknesses. The thinnest black lines represent the limits of a sub-step deposit, while the different types of thicker black lines separate the strata of each event.	95
5.40	Map views of the top of the resulting deposits after each event of the sedimentation on an irregular terrain problem, computed imposing event progradation and using one sedimentation sub-step. Here the progression (a) through (e) represents the sequence of the simulated events.	97
5.41	3D views of the obtained deposit thicknesses after each event of the sedimentation on an irregular terrain problem, computed imposing event progradation and using one sedimentation sub-step. Here the progression (a) through (e) represents the sequence of the simulated events.	98
5.42	3D views of the final deposit thicknesses associated with each sediment type of the sedimentation on an irregular terrain problem, computed imposing event progradation and using one sedimentation sub-step. (a) Coarse sand; (b) Medium sand; (c) Fine sand; (d) Silt; (e) Clay.	99
5.43	(a) shows the position of the white line along which we extracted the vertical section displayed in (b), containing the stacking pattern obtained at the sedimentation on an irregular terrain problem, computed imposing event progradation and using one sedimentation sub-step. Layer thicknesses in (b) were scaled by a factor of 50 to aid visualization.	100

5.44	Map views of the top of the resulting deposits after each event of the sedimentation on an irregular terrain problem, computed imposing event progradation and using three sub-steps. Here the progression (a) through (e) represents the sequence of the simulated events, while the maps from left to right mark the sub-steps succession.	101
5.45	3D views of the obtained deposit thicknesses after each event of the sedimentation on an irregular terrain problem, computed imposing event progradation and using three sub-steps. Here the progression (a) through (e) represents the sequence of the simulated events. . . .	102
5.46	3D views of the final deposit thicknesses associated with each sediment type of the sedimentation on an irregular terrain problem, computed imposing event progradation and using three sub-steps. (a) Coarse sand; (b) Medium sand; (c) Fine sand; (d) Silt; (e) Clay. . . .	103
5.47	(a) shows the position of the white line along which we extracted the vertical section displayed in (b), containing the stacking pattern obtained at the sedimentation on an irregular terrain problem, computed imposing event progradation and using three sub-steps. Layer thicknesses in (b) were scaled by a factor of 50 to aid visualization. . . .	104
5.48	3D views of the final deposit thicknesses associated with each sediment type of the sedimentation on an irregular terrain problem, computed using three sub-steps per event. (a) Coarse sand; (b) Medium sand; (c) Fine sand; (d) Silt; (e) Clay.	105
5.49	(a) shows the position of the white line along which we extracted the vertical section displayed in (b), containing the stacking pattern obtained at the sedimentation on an irregular terrain problem, computed using three sub-steps per event. Here the thicker lines mark the transition between each stratigraphic application's simulation step. Layer thicknesses in (b) were scaled by a factor of 50 to aid visualization.	106
5.50	(a) Map view of the deposit thickness by the end of the simulation of the sedimentation on an irregular terrain problem, computed using three sub-steps per event; (b) Overlay of the real turbidite system presented by MACHADO <i>et al.</i> [112] (in black) onto the obtained deposit thickness map.	107
E.1	Plot of the energy-height curve for flows in a 1D channel. It show the critical height and energy values, and the intervals of the curve that denote subcritical or supercritical regimes.	141

E.2	Representation of the solution of the subcritical flow in a 1D channel with a bump.	142
E.3	Energy-height diagram for the subcritical flow in a 1D channel over a bump. The points <i>A</i> and <i>B</i> indicate different states of the fluid along the channel.	142
E.4	Representation of the solution of the transcritical flow in a 1D channel with a bump.	143
E.5	Energy-height diagram for the transcritical flow in a 1D channel over a bump. The points <i>A</i> , <i>B</i> and <i>C</i> indicate different states of the fluid along the channel.	143
E.6	Representation of the solution of the transcritical flow with a shock in a 1D channel with a bump.	144
E.7	Energy-height diagram for the transcritical flow with a shock in a 1D channel over a bump. The points <i>A</i> through <i>E</i> indicate different states of the fluid along the channel.	145

List of Symbols

Roman Symbols

$\mathbf{A} = \begin{bmatrix} \mathbf{A}_x & \mathbf{A}_y \end{bmatrix}$ Advection matrix of the generalized convection-diffusion equation that governs the turbidity current dynamics, p. 16.

$\tilde{\mathbf{A}}_0$ Jacobian matrix of the transformation from entropy to conservation variables, p. 22.

\mathbf{A} Left-hand side matrix of the linear system solved in the FCT scheme, p. 30.

$\mathbf{b} = (b_x, b_y, b_z)$ Body forces acting on the current, p. 7.

\mathbf{B} Right-hand side matrix of the linear system solved in the FCT scheme, p. 30.

c Propagation speed of a perturbation on the current free-surface, p. 22.

c_b Compaction coefficient controlling the decrease in ϕ as the buried depth increases, p. 56.

c_d Drag coefficient associated with the frictional forces acting between the current and the bed, p. 15.

C_u Velocity cut-off height value, p. 37.

\mathbf{C} Artificial diffusion matrix used to build the low-order equation of the FCT scheme, p. 31.

\mathbf{D} Generalized stiffness matrix of the solved semi-discrete system, p. 20.

f_s Factor that controls the height profile and the maximum extent of the deposits created as a result of scattering sediments from a single cell, p. 54.

\mathbf{F}	Source term of the solved semi-discrete system, p. 20.
\mathcal{F}_i	Anti-diffusive flux acting on the i -th node, p. 34.
\mathcal{F}_{ij}	Contribution of the j -th node to the total anti-diffusive flux acting on the node i , p. 34.
g	Module of the gravitational acceleration, p. 10.
g'	Reduced gravity inside the current due to buoyancy forces, p. 12.
\mathbf{G}	Solution value prescribed at the boundary Γ_e , p. 17.
h	Turbidity current height (or thickness), p. 6.
h_{dry}	Current height above which a node is classified as being wet, p. 37.
h_{ij}	Hydrostatic reconstruction of the current's height at the i -th node with respect to the j -th one, p. 31.
h_{max}	Point-wise maximum current height obtained during the flow simulation of an event, p. 53.
$H^{1h}(\bar{\Omega})$	Finite-dimensional first-order Hilbert space defined in the domain $\bar{\Omega}$, p. 18.
\mathbf{I}_3	Third-order identity matrix, p. 19.
\mathbf{j}	Normalized gradient of the current free surface elevation, p. 22.
k_{mean}	Mean curvature of a streakline segment, p. 43.
\mathbf{K}	Matrix with the diffusion coefficients of the generalized convection-diffusion equation that governs the turbidity current dynamics, p. 16.
l_e	Characteristic length of the e -th finite element, p. 36.
l_i^d	Largest diagonal of the element that contains the i -th point of a streakline, p. 42.
L_{max}	Maximum length of a streakline segment, p. 44.
\mathbf{L}	Matrix obtained by adding the artificial diffusion matrix \mathbf{C} to the generalized stiffness matrix \mathbf{D} , p. 31.
\mathbf{M}	Consistent mass matrix of the solved semi-discrete system, p. 20.

\mathbf{M}_L	Lumped mass matrix, p. 28.
n	Manning's roughness coefficient, p. 15.
n_{el}	Number of elements into which the domain is partitioned, p. 20.
n_{npe}	Number of nodes per element, p. 22.
n_p^k	Number of points to be added to a streakline segment as a result of it having a too high curvature, p. 44.
n_p^l	Number of points to be added to a streakline segment as a result of it being too long, p. 44.
\mathbf{n}	Outward-pointing unit normal at the boundary Γ , p. 17.
\mathbf{n}_b	Normal vector at the bed pointing into the current, p. 8.
\mathbf{n}_η	Normal vector at the current free surface pointing outwards., p. 7.
N_i	Finite element shape function associated with the i -th node, p. 19.
\mathbf{N}	Matrix containing the diagonal matrices \mathbf{N}_i , relative to each finite element shape function, p. 19.
\mathbf{N}_i	Diagonal matrix whose non-null entries are N_i , p. 19.
p	Fluid pressure, p. 7.
$\mathbf{p}(x, y)$	Point/particle at a streakline, p. 42.
q_{ij}	Reconstructed specific discharge at the node i with respect to the j -th one, p. 31.
$\mathbf{q} = (q_x, q_y)$	Current specific discharges (or discharges per unit width), p. 12.
\mathbf{R}^h	Finite element approximation to the residual of the convection-diffusion equation that governs turbidity current dynamics, p. 18.
$S(i)$	Set of the neighboring nodes of the node i , which includes all those belonging to the elements adjacent to i , p. 32.
\mathcal{S}^h	Finite-dimensional space of test functions, p. 18.
\mathbf{S}	Source term of the generalized convection-diffusion equations that governs the turbidity current dynamics, p. 16.
t	Time instant, p. 6.

- t_f Final time of the flow simulation, p. 17.
- $\Delta t = t^{n+1} - t^n$ Current time step, p. 22.
- TOL_1, TOL_2 Tolerance values used by the stopping criteria of the nonlinear iterations performed on the stabilized and FCT approaches, p. 24.
- $\mathbf{u} = (u, v, w)$ Turbidity current velocity, p. 7.
- \mathbf{u}_b Current velocity at the bed, p. 8.
- \mathbf{u}_η Current velocity at its free surface, p. 8.
- $\mathbf{U} = \begin{bmatrix} h & q_x & q_y \end{bmatrix}^T$ Vector of conservation variables, p. 16.
- V_{eA}^n Accumulated volume injected by the event e by the end of the time step n , p. 41.
- ΔV_e^n Volume injected into the domain by the event e at the time step n , p. 41.
- V_{eT} Total volume initially available to the event e , p. 41.
- \mathbf{V} Vector with all nodal solution values, p. 19.
- $\mathbf{V}_i = \begin{bmatrix} h_i & q_{xi} & q_{yi} \end{bmatrix}$ Solution at the i -th node, p. 19.
- \mathbf{V}^L Low-order nodal solutions of the FCT scheme, p. 32.
- \mathcal{V}^h Finite-dimensional space of trial functions, p. 18.
- Vol Current fluid volume stored in the domain, p. 64.
- Vol_{err} Relative error of the fluid volume stored in the domain, p. 64.
- \mathbf{W}^h Vector with the finite element trial functions, p. 18.
- $\mathbf{x} = (x, y, z)$ Position in the Cartesian coordinate system, p. 6.
- z_b Bed elevation from a datum fixed at the bed's lowest point, p. 6.

Greek Symbols

- α Ratio of the drag forces at the top of the flow and at the bed, p. 15.
- α_{ij} Scaling factor of the anti-diffusive flux \mathcal{F}_{ij} , p. 35.
- δ_{ij} Kronecker delta, p. 27.

δ_{Shock}	Coefficient of the shock-capturing operator in the stabilized finite element formulation, p. 21.
η	Elevation of the current-water interface, p. 7.
γ	Parameter associated with the frictional shear stresses acting on the current, p. 15.
Γ	Boundary of the domain where the simulation takes place, p. 17.
Γ_e	Subset of the domain's boundary where essential boundary conditions are enforced, p. 17.
$\partial_e \Gamma_e$	Subset of the contour Γ_e where the event e is active, p. 41.
Γ_n	Subset of the domain's boundary where natural boundary conditions are enforced, p. 17.
μ	Turbidity current dynamic viscosity, p. 6.
μ_w	Water dynamic viscosity, p. 6.
Ω	Internal region of the simulation domain, p. 17.
$\bar{\Omega}$	Closed domain where the simulation takes place, p. 17.
Ω_e	Internal region of the e -th finite element, p. 20.
ϕ	Average porosity of a deposited sediment (or lithology) layer, p. 56.
ϕ_0	Initial/surface porosity of a lithology type, p. 56.
ρ	Turbidity current density, p. 6.
ρ_w	Water density, p. 6.
τ_{SUPG}	Coefficient of the SUPG operator in the stabilized finite element formulation, p. 21.
$\boldsymbol{\tau}_{\mathbf{b}} = (\tau_{b_x}, \tau_{b_y})$	Shear stress incurred by the current at the bed, p. 8.
$\boldsymbol{\tau}_{\boldsymbol{\eta}} = (\tau_{\eta_x}, \tau_{\eta_y})$	Shear stress incurred by the current at its free interface, p. 8.
θ	Parameter that controls the stability and precision of the numerical time integration method, p. 24.
ζ_{max}	Maximum circle sector angle of a streakline segment, p. 44.

Chapter 1

Introduction

In the petroleum geology field, one of the main concerns is understanding the formation and evolution of potential hydrocarbon reservoirs. Besides, the characterization of siliciclastic areas in sedimentary basins and, on a smaller scale, the depositional features whose architectures form reservoirs is of particular interest. It is as essential to the exploration and prospection of new reserves as to the optimization of oil production. At present, important hydrocarbon reservoirs discovered in deep-water sediments are trapped into turbidite reservoirs, whose majority are in siliciclastic turbidites and hyper-concentrated flow deposits [1].

By definition, turbidite deposits are formed by the accumulation of the sediments carried by turbidity currents. These gravitational flows are broadly affected by turbulence effects. In fact, the more turbulent is the flow, the more the sediments are drawn by the current and, with the increased density, it gains more speed, which, as a result, increases turbulence. On this subject, MEIBURG and KNELLER [2] present an elegant review focused on fluid mechanics. In general, a turbidity current can be induced by sediment-laden river outflows, earthquakes, volcanic activities, storms, tsunami, and other sudden geological events [3]. However, when referring to a turbidite system, we are usually concerned with the outcome of a succession of these events, distributed over a long geologic period.

One way of studying sedimentary basins is through the point of view of stratigraphy. This discipline is mainly concerned with the formation of rock layers (strata) and the layering process (stratification). Its goal is to correlate strata separated in space and time in regard to their geometric relationships and to estimate the characteristics of the original depositional environment. However, stratigraphic studies are never straightforward and usually require different approaches and pieces of evidence as none are unbiased or yield completely unambiguous results. Thus, this field is traditionally subdivided into some branches according to their different strategies [4]. For example, chronostratigraphy seeks to assign absolute dates to the depositional sequence of all rocks within a region, providing an entire geological

record. Lithostratigraphy is mainly concerned with physical contrasts between rock types (lithologies). In turn, sequence stratigraphy surveys the succession of genetically related strata, i.e., of layers that were deposited in a single cycle of relative sea-level change. Undoubtedly, the evolution of the sea level might condition significant lateral shifts in the depositional pattern, which can form alternating layers of porous and permeable materials and sealing media. These structures can build oil traps and create reservoirs.

In complement to these and the other stratigraphy branches, numerical simulations of the sedimentary process should provide more data to reduce the inaccuracies of the whole stratigraphic model. Indeed, they can help understand the formation of a specific area, test a qualitative hypothesis, or provide quantitative data like the deposited rocks' properties (composition, porosity, permeability, texture). For such purposes, simulators based on sequence stratigraphy operate with time steps chosen according to the order of the sea level variation cycles that should be captured, usually ranging from 0.1 Ma to 10 Ma. Also, in basin-scale models, the studied regions have areas in the order of $100 \text{ km} \times 100 \text{ km}$, which are frequently decomposed into 1 km^2 elements (or cells) [5–7].

For instance, CARVALHO *et al.* [7–9] describe an algorithm for the transport and deposition of sediments in platform, slope and basin environments. Their technique applies quantitative sequence stratigraphy concepts as a sedimentation control mechanism, combined with a 2D steady-state flow simulation of an inviscid fluid that highly depends on the region's bathymetry. Given velocity-based boundary conditions, they compute a velocity field from which streamlines are obtained. These lines that originate from predetermined sediment supplies are used to transport sediments. The deposition itself is controlled by the angle of stability of each granulometric fraction and the available accommodation space. Additionally, other stratigraphic simulation techniques have been proposed. SHAFIE and MADON [10] present a comprehensive review of the main ones. STROBEL *et al.* [11] perform 2D simulations based on geometric rules, LEE and HARBAUGH [12] apply the particle-in-cell method to simulate 3D flows, and GRANJEON [5] describes a diffusion-based model.

Therefore, the main objective of this work is to develop a solution capable of simulating the formation of sedimentary basins in terms of the layering of different strata as a result of the sediment transport and the hydrodynamics of turbidity currents. For this purpose, we seek to improve the approach of CARVALHO *et al.* [7–9] in two senses. First, we employ a more realistic flow model. Possible advances could be to perform a transient analysis or to consider bed friction and turbulence effects. Second, we allow the simulation of turbidity currents, i.e., we must take into account the flow of sediment currents across an aqueous body, instead of just accounting for

the dynamics of the water itself.

To simulate the flow, we assume that the shallow water hypothesis holds. This approximation presupposes the horizontal extent of the flow is significantly higher than its vertical scale. In our case, the height of a turbidity current is many times smaller than the size of a basin. Thus, simplifications can be made on standard Navier-Stokes equations, producing a so-called depth-averaged or shallow water model. These modifications are fundamental to reduce computational costs, enabling the simulation of vast regions over long periods. On the other hand, depth-resolving strategies can provide more accurate results at the expense of additional computational effort. In this sense, MEIBURG *et al.* [13] discuss the strengths and challenges associated with the different approaches. Finite element methods have been used successfully [14–17] in depth-resolving models, and more recent studies include uncertainty quantification [18, 19].

Shallow water models have found their way into the study of geophysical problems such as atmospheric [20] and ocean [21] modeling, where extensive areas need to be examined. Besides, some have been derived for compressible flows, such as for gas motion [22] and magnetohydrodynamics [23]. Regarding morphodynamic approaches, some authors consider movable beds [24, 25] or flow and sediment/particle coupled models [26], with applications in the simulation of floodings [27, 28] and turbidity currents [29, 30]. Also, GROENENBERG *et al.* [31, 32] have shown usage in turbidite fan stratigraphy.

To numerically solve the shallow water equations, often finite difference [31] and finite volume [33] methods are employed. Also, several discontinuous [34] and continuous [35–37] finite element techniques have been applied over the years. Within the finite element group, stabilized formulations, formed by adding consistent and numerically stabilizing terms to the Galerkin method, have achieved considerable success [38–40]. Usually, they use a Streamline Upwind Petrov-Galerkin (SUPG) [41] term combined with a shock-capturing operator, such as the Consistent Approximate Upwind (CAU) [42, 43] operator. SANTOS and COUTINHO [37] evaluate the use of different SUPG and shock-capturing techniques to solve the shallow water equations. Similarly, regarding continuous finite elements, flux-corrected transport (FCT) [44–47] methods constitute a relevant subgroup. Classical FCT schemes employ a low-order method imbued with enough artificial diffusion to obtain a non-oscillatory smooth solution, which is then corrected by anti-diffusive fluxes limited to avoid the creation or growth of extrema. An alternative procedure is to blend the high- and low-order fluxes' approximations in a high-resolution scheme [48].

In this work, we adopt a semi-discrete finite element FCT technique with implicit time integration. Our low-order formulation is built by adding a novel Rusanov-like

scalar dissipation to standard Galerkin equations. Another possible approach would be to add, to a Taylor-Galerkin finite element model, artificial diffusion proportional to the difference between the consistent and lumped mass matrices [46, 49]. Next, the high-order equation is composed by adding limited anti-diffusive fluxes to the low-order equations. Here, we linearize and limit the fluxes around the low-order solution so these computations can be performed once per time step. Limiting is done with a Zalesak [50] limiter, and a prelimiter is also enforced.

To better evaluate the proposed FCT scheme, we test it against a stabilized finite element formulation. The examined stabilized approach comes from the work of SANTOS and COUTINHO [37]. We use the SUPG operator proposed by TEZDUYAR [51], later adapted by TAKASE *et al.* [52] to the shallow water equations. Plus, we employ the $YZ\beta$ [53, 54] and CAU [42, 43] shock-capturing operators. All implementations related to the finite element method were aided by the deal.II library [55], and thus we only use quadrilateral elements.

In many real-world applications of a shallow water simulator, the area in question may experience transitions between wet and dry states. For instance, such are the cases of simulations of coastal regions with periodic tidal oscillations [56, 57], dam breaks [58] and floodings [59]. For turbidity currents [31], the wet and dry states refer to the region being wet by the current or not. Thus, this type of simulation is always treated as the flooding of a “wet” current into a “dry” ambient fluid. Near fronts of dry/wet transition, the current’s height tends to zero, giving rise to some numerical problems that can lead to instabilities, mass unbalances or nonphysical behaviors. This is especially true when dealing with complex bathymetries. For this reason, appropriate drying/wetting techniques must be applied.

With a better flow model, the sedimentation algorithm still needs to be adapted. Hence, we start by transporting sediments through streaklines evolved during the flow simulation and radially scattering them. In this context, we propose a method to adaptively add and remove points from the streaklines to ensure accuracy and reduce computational costs. Then, during the deposition operation, the available accommodation space is limited by the point-wise maximum current heights. Additionally, to try to reproduce lithology stacking patterns seen in nature, we allow the sedimentation to be made in sub-steps that have even more limited accommodation spaces. Meanwhile, as new rock layers are deposited, we compact those underneath due to the overburden.

To make our approach more suitable to simulate the creation of turbidite systems, we have chosen to work with event-driven simulations. Accordingly, sedimentation and flow computations are performed on a per-event basis. In this work, we show how these events can be defined and monitored over time. However, the simulation of distinct events has been addressed in the past by GROENENBERG *et al.* [31, 32].

In their works, the simulation of an event is executed while the suspended sediment concentration is above a given threshold. Also, the quiet intervals between events can incur user-specified tectonic activity and a background sedimentation that can cover the turbidite deposits. Although they need to specify the inflow sediment volume and a recurrence frequency to define an event, the fact that they simulate the flow and sedimentation processes in a more strongly coupled manner causes its definition and usage to be slightly different from how we handle events in our simulations.

All in all, we evolve the stratigraphic simulation procedure introduced by CARVALHO *et al.* [7–9] with the ability to simulate the construction of turbidite deposits due to the development of several turbidity current events distributed over long periods. Furthermore, the shallow water flow model employed in the current dynamics is solved with an original FCT scheme. The artificial diffusion dependent on the hydrostatic reconstruction of current height and specific discharges is one of the main contributions of the proposed FCT scheme. It is reduced near dry/wet fronts as a result of the enforced velocity cut-offs, avoiding displacing some fluid incorrectly into a dry region. Also, the shock detector we present shows good results in the tested cases. Plus, while we assessed that the anti-diffusive correction factors computed with the proposed flux limiter should be synchronized into a single component, we observe that the chosen prelimiter should be applied individually to each flux component. Finally, to the best of our knowledge, no other approach has combined a process-based flow simulation with a sedimentation procedure as we propose.

The remainder of this thesis is structured as follows: In Chapter 2, we present our approach to simulate the flow of turbidity currents. Next, in Chapter 3, we describe how we integrate it with a practical sedimentation procedure to simulate the formation of turbidite systems resulting from recurrent events in an event-driven framework. In sequence, Chapter 4 presents some implementation details, while the obtained results are discussed in Chapter 5. At last, Chapter 6 wraps up with our conclusions and some future work suggestions.

Chapter 2

Flow Simulation

In this chapter, we introduce how we perform numerical simulations of turbidity current flows. First, we present the adopted shallow water model, together with its governing equations. Next, we enunciate the continuous and the semi-discrete initial value problems to be solved. Then, we show the stabilized finite element formulation and the FCT scheme employed to obtain numerical solutions and display how we adaptively control times step lengths during simulations. Finally, we examine the techniques used to avoid numerical instabilities and nonphysical behaviors as the current's height tends to zero.

2.1 Governing Equations

A schematic representation of the employed shallow water model can be seen in Figure 2.1. It depicts a sediment current of constant density ρ that flows over a terrain across a quiescent ambient fluid of density ρ_w . In this work, we let the ambient fluid and terrain be, respectively, the seabed and the seawater. $h(x, y, t)$ represents the current's thickness, while $z_b(x, y)$ is the seabed elevation from a datum fixed at the bed's lowest point. In addition, we assume that the turbidity current is diluted enough to behave as a Newtonian fluid and to allow its dynamic viscosity to be approximately equal to the water viscosity ($\mu = \mu_w = 10^{-3}$ Pa s).

As suggested by GROENENBERG [32], we consider the ambient fluid to be infinitely deep, so we can use a single-layer model (as opposed to multi-layer versions [60–62]), which is valid when the thickness of the current is much smaller than the depth of the real ambient fluid. In this scenario, the water column is so high that current height variations do not result in significant relative pressure changes in the ambient fluid along the current-water interface. Plus, the drag forces exerted by the current to the ambient water can be disregarded since it has no expressive impact if we consider the whole water column. As a result, the ambient pressure of an initially still water body will always have no dynamic component, and we can

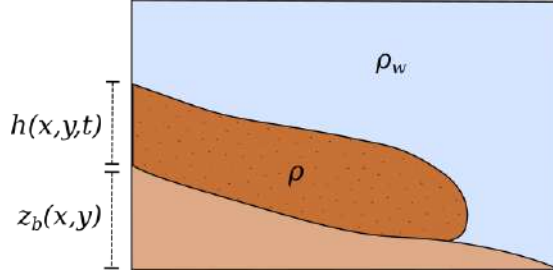


Figure 2.1: Schematic representation of the adopted shallow water model. It shows a turbidity current with density ρ and thickness $h(x, y, t)$ that flows over a terrain, whose elevation is $z_b(x, y)$, and advances through a motionless fluid of density ρ_w .

assume that the fluid remains motionless as the current flows.

To derive the equations governing the dynamics of a turbidity current, we start from the Navier-Stokes set of equations for incompressible flows. It comprises the mass conservation equation (or the continuity equation):

$$\nabla \cdot \mathbf{u} = 0, \quad (2.1)$$

and the momentum conservation equation:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\frac{1}{\rho} \nabla p + \frac{\mu}{\rho} \nabla^2 \mathbf{u} + \mathbf{b}, \quad (2.2)$$

where $\mathbf{u} = (u, v, w)$ is the fluid's velocity, p indicates the pressure, and the source term $\mathbf{b} = (b_x, b_y, b_z)$ covers the body forces applied to the fluid.

2.1.1 Boundary Conditions

The first step to derive the equations that govern the model is to define some boundary conditions. Initially, we define a kinematic condition for the interface between the current and the ambient fluid. Let $\eta = h + z_b$ be the elevation of the current free surface. The equation that defines it is $z = \eta(x, y, t)$, which can also be written as the level surface $\Phi(x, y, z, t) = 0$, with $\Phi(x, y, z, t) = z - \eta(x, y, t)$. Thus, the outward-pointing normal at the surface is:

$$\mathbf{n}_\eta = \nabla \Phi = \left(-\frac{\partial \eta}{\partial x}, -\frac{\partial \eta}{\partial y}, 1 \right), \quad (2.3)$$

which is not normalized at this point. Then, assuming that the surface is impermeable, $d\Phi/dt = 0$ holds in the Lagrangian perspective. This implies that a fluid particle initially on the surface will always remains attached to it. If the velocity of

a point on the free surface is $\mathbf{u}_\eta = (u_\eta, v_\eta, w_\eta)$, we have:

$$\begin{aligned}\frac{d\Phi}{dt} &= \frac{dz}{dt} - \frac{\partial\eta}{\partial x} \frac{dx}{dt} - \frac{\partial\eta}{\partial y} \frac{dy}{dt} - \frac{\partial\eta}{\partial t} = 0 \\ \therefore w_\eta - \frac{\partial\eta}{\partial x} u_\eta - \frac{\partial\eta}{\partial y} v_\eta - \frac{\partial\eta}{\partial t} &= 0 \\ \therefore \frac{\partial\eta}{\partial t} - \mathbf{u}_\eta \cdot \mathbf{n}_\eta &= 0.\end{aligned}\tag{2.4}$$

We can also perform an analogous procedure to obtain a kinematic condition for the bed surface. In this case, the normal vector pointing out of the bed is:

$$\mathbf{n}_b = \left(-\frac{\partial z_b}{\partial x}, -\frac{\partial z_b}{\partial y}, 1 \right).\tag{2.5}$$

Hence, if the fluid velocity on the bed is $\mathbf{u}_b = (u_b, v_b, w_b)$, we can arrive to:

$$\frac{\partial z_b}{\partial t} - \mathbf{u}_b \cdot \mathbf{n}_b = 0.\tag{2.6}$$

However, in this work, we consider that the bed elevation does not change during the flow simulation, i.e.:

$$\frac{\partial z_b}{\partial t} = \mathbf{u}_b \cdot \mathbf{n}_b = 0.\tag{2.7}$$

In sequence, we define dynamic boundary conditions for the current free surface and the bed. We impose the following tangential stress on the current-water interface:

$$\boldsymbol{\tau}_\eta = (\tau_{\eta_x}, \tau_{\eta_y}) = -\mu \frac{\partial \mathbf{u}_\eta}{\partial \hat{\mathbf{n}}_\eta} = -\mu (\hat{\mathbf{n}}_\eta \cdot \nabla) \mathbf{u}_\eta,\tag{2.8}$$

where $\hat{\mathbf{n}}_\eta$ is the unit normal vector on the current-water interface, pointing to the outside of the current. In addition, supposing that the surface is smooth, i.e. $\partial\eta/\partial x \ll 1$ and $\partial\eta/\partial y \ll 1$, we can assign $\hat{\mathbf{n}}_\eta = \mathbf{n}_\eta$.

Similarly, we impose the bottom shear stress acting on the current:

$$\boldsymbol{\tau}_b = (\tau_{b_x}, \tau_{b_y}) = \mu \frac{\partial \mathbf{u}_b}{\partial \hat{\mathbf{n}}_b} = \mu (\hat{\mathbf{n}}_b \cdot \nabla) \mathbf{u}_b,\tag{2.9}$$

where $\hat{\mathbf{n}}_b$ is the unit normal vector on the bed, pointing into the current. Once more, we assume the surface is smooth, i.e. $\partial z_b/\partial x \ll 1$ and $\partial z_b/\partial y \ll 1$, and adopt $\hat{\mathbf{n}}_b = \mathbf{n}_b$.

2.1.2 Scale Considerations

Now we introduce some scale considerations regarding shallow water models and examine their impact on the momentum conservation equation (2.2). First, let L_x ,

U and L_x/U be the flow's characteristic length, speed and time scales along the x -direction. For the y - and z -directions, the correspondent scales are L_y , V , L_y/V , L_z , W and L_z/W . Also, define the operator $O(\text{scale})$ to indicate the order of magnitude of a desired scale. In the study that follows, consider that $O(L_x) = O(L_y)$, using $L_x = L_y = L$. Then, the shallow water hypothesis states that:

$$\frac{L_z}{L} = \varepsilon \ll 1. \quad (2.10)$$

In fact, TAN [63] says it is usually required that $L_z/L < 10^{-3}-10^{-4}$, while SLOBODICOV [64] and VREUGDENHIL [65] consider that the ratio $L_z/L = 0.05$ is the upper limit on which the shallow water theory is valid. This value comes from the fact that a long shallow water wave is produced if the ratio between water depth and wavelength is less than approximately 0.05 [66]. All in all, the general idea is that the extent of the region to be simulated is considerably greater than the height of the turbidity current. Additionally, if we consider that the flow has the same characteristic time scale T on all directions (x , y and z), the proposition (2.10) implies that:

$$\begin{aligned} T &= \frac{L}{U} = \frac{L}{V} = \frac{L_z}{W} \\ \therefore V &= U \quad \text{and} \quad W = \frac{L_z}{L}U = \varepsilon U, \end{aligned} \quad (2.11)$$

i.e., the z -component of the flow velocity is also much smaller than the x - and y -components.

Next, based on the relations (2.10) and (2.11), we assess the main contributions to the fluid pressure variation. We initially examine the order of magnitude of the terms of the momentum equation (2.2) along the x -direction. For the transient and advective terms:

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} \longrightarrow O \left(\frac{U}{T} + \frac{U^2}{L} + \frac{U^2}{L} + \frac{WU}{L_z} \right) = O \left(\frac{U^2}{L} \right), \quad (2.12)$$

and for the diffusive term:

$$\frac{\mu}{\rho} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) \longrightarrow \frac{\mu}{\rho} O \left(\frac{U^2}{L^2} + \frac{U^2}{L^2} + \frac{U^2}{L_z^2} \right) = \frac{\mu}{\rho} O \left(\frac{U^2}{\varepsilon^2 L^2} \right). \quad (2.13)$$

Likewise, the same orders of magnitude can be obtained for the analogous terms of the momentum equation along the y -direction. However, if we regard the balance (2.11) in the z -direction, we get different orders of magnitude for the transient and

advective terms:

$$\frac{\partial w}{\partial t} + u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z} \longrightarrow O \left(\frac{W}{T} + \frac{UW}{L} + \frac{UW}{L} + \frac{W^2}{L_z} \right) = O \left(\frac{\varepsilon U^2}{L} \right), \quad (2.14)$$

and for the diffusive term:

$$\frac{\mu}{\rho} \left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2} \right) \longrightarrow \frac{\mu}{\rho} O \left(\frac{W^2}{L^2} + \frac{W^2}{L^2} + \frac{W^2}{L_z^2} \right) = \frac{\mu}{\rho} O \left(\frac{U^2}{L^2} \right). \quad (2.15)$$

We can observe that the order of the transient, advective and viscous terms in the z -direction is many times smaller than that of their respective counterparts in the x - and y -directions. This means that the dynamic components of the pressure inside the current has a much greater contribution coming from the horizontal directions than from the vertical. Thus, we can neglect the dynamic components along the current height, obtaining hydrostatic pressure profiles in this direction. In this case, the associated momentum balance equation is reduced to:

$$\frac{\partial p}{\partial z} = -\rho g, \quad (2.16)$$

where $g = 9.81 \text{ m s}^{-2}$ is the gravitational acceleration.

Therefore, to obtain the pressure at a position z inside the current, we can integrate (2.16) along the z -direction:

$$\begin{aligned} \int_z^\eta \frac{\partial p}{\partial z} dz &= - \int_z^\eta \rho g dz \\ \therefore p_\eta - p &= -\rho g(\eta - z) \\ \therefore p &= p_\eta + \rho g(\eta - z), \end{aligned} \quad (2.17)$$

where p_η is the pressure at the current-water interface. As the ambient fluid is at rest, the pressure is hydrostatic and can be determined. So, we arrive at:

$$\begin{aligned} p &= p_{\text{atm}} + \int_\eta^{\eta_w} \rho_w g dz + \rho g(\eta - z) \\ \therefore p &= p_{\text{atm}} + \rho_w g(\eta_w - \eta) + \rho g(\eta - z), \end{aligned} \quad (2.18)$$

where p_{atm} is the atmospheric pressure and η_w is the elevation of the water free surface. At last, considering that p_{atm} and η_w are constants, we differentiate the previous equation in respect to x and y to obtain the pressure gradients we will employ in the momentum conservation equation (2.2):

$$\frac{\partial p}{\partial x} = (\rho - \rho_w) g \frac{\partial \eta}{\partial x}, \quad (2.19)$$

where the terms $h\bar{u}$ and $h\bar{v}$ are the fluid specific discharges (or discharges per unit width) along the x - and y -directions. They are often denoted by $\mathbf{q} = (q_x, q_y) = (h\bar{u}, h\bar{v})$.

2.1.4 Momentum Conservation Equation

We can arrive at the model's momentum conservation equation by integrating Equation (2.2) over the current's height. For the equation in the x -direction:

$$\int_{z_b}^{\eta} \left(\frac{\partial u}{\partial t} + \frac{\partial u^2}{\partial x} + \frac{\partial(uv)}{\partial y} + \frac{\partial(uw)}{\partial z} \right) dz = \int_{z_b}^{\eta} \left(-\frac{1}{\rho} \frac{\partial p}{\partial x} + \frac{\mu}{\rho} \nabla^2 u + b_x \right) dz. \quad (2.28)$$

Similar to the definition of the depth-averaged velocities in (2.26), the contribution of the body forces is:

$$\int_{z_b}^{\eta} b_x dz = h\bar{b}_x, \quad (2.29)$$

where \bar{b}_x is the depth-averaged x -component of the volume forces. Nevertheless, in this work, we account for no such forces along the x - and y -directions.

In sequence, using Equation (2.19), we expand the pressure term into:

$$\int_{z_b}^{\eta} \frac{1}{\rho} \frac{\partial p}{\partial x} dz = \int_{z_b}^{\eta} \left(\frac{\rho - \rho_w}{\rho} \right) g \frac{\partial \eta}{\partial x} dz = \left(\frac{\rho - \rho_w}{\rho} \right) gh \frac{\partial \eta}{\partial x}. \quad (2.30)$$

Then, we consider that $g' = (\rho - \rho_w)g/\rho$ is the reduced gravity due to buoyancy forces and replace $\eta = h + z_b$ to obtain:

$$\int_{z_b}^{\eta} \frac{1}{\rho} \frac{\partial p}{\partial x} dz = g'h \frac{\partial h}{\partial x} + g'h \frac{\partial z_b}{\partial x} = \frac{\partial}{\partial x} \left(\frac{g'h^2}{2} \right) + g'h \frac{\partial z_b}{\partial x}. \quad (2.31)$$

In the case of the viscous term, we have:

$$\begin{aligned} \int_{z_b}^{\eta} \frac{\mu}{\rho} \nabla^2 u dz &= \frac{1}{\rho} \int_{z_b}^{\eta} \left[\frac{\partial}{\partial x} \left(\mu \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left(\mu \frac{\partial u}{\partial y} \right) + \frac{\partial}{\partial z} \left(\mu \frac{\partial u}{\partial z} \right) \right] dz = \\ &= \frac{1}{\rho} \left\{ \left[\frac{\partial}{\partial x} \left(\int_{z_b}^{\eta} \mu \frac{\partial u}{\partial x} dz \right) - \mu \frac{\partial u_{\eta}}{\partial x} \frac{\partial \eta}{\partial x} + \mu \frac{\partial u_b}{\partial x} \frac{\partial z_b}{\partial x} \right] + \left[\frac{\partial}{\partial y} \left(\int_{z_b}^{\eta} \mu \frac{\partial u}{\partial y} dz \right) + \right. \right. \\ &\quad \left. \left. - \mu \frac{\partial u_{\eta}}{\partial y} \frac{\partial \eta}{\partial y} + \mu \frac{\partial u_b}{\partial y} \frac{\partial z_b}{\partial y} \right] + \left[\mu \frac{\partial u_{\eta}}{\partial z} - \mu \frac{\partial u_b}{\partial z} \right] \right\} = \frac{1}{\rho} \left[\frac{\partial}{\partial x} \left(\int_{z_b}^{\eta} \mu \frac{\partial u}{\partial x} dz \right) + \right. \\ &\quad \left. + \frac{\partial}{\partial y} \left(\int_{z_b}^{\eta} \mu \frac{\partial u}{\partial y} dz \right) + \left(-\mu \frac{\partial u_{\eta}}{\partial x} \frac{\partial \eta}{\partial x} - \mu \frac{\partial u_{\eta}}{\partial y} \frac{\partial \eta}{\partial y} + \mu \frac{\partial u_{\eta}}{\partial z} \right) + \left(\mu \frac{\partial u_b}{\partial x} \frac{\partial z_b}{\partial x} + \right. \right. \\ &\quad \left. \left. + \mu \frac{\partial u_b}{\partial y} \frac{\partial z_b}{\partial y} - \mu \frac{\partial u_b}{\partial z} \right) \right] = \frac{\mu}{\rho} \frac{\partial}{\partial x} \left(\int_{z_b}^{\eta} \frac{\partial u}{\partial x} dz \right) + \frac{\mu}{\rho} \frac{\partial}{\partial y} \left(\int_{z_b}^{\eta} \frac{\partial u}{\partial y} dz \right) - \frac{\tau_{\eta x} + \tau_{b x}}{\rho}, \end{aligned} \quad (2.32)$$

where we have employed the Leibniz integral rule and the dynamic boundary con-

ditions (2.8) and (2.9) at the current free surface and at the bed. The first integral term can be expanded into:

$$\begin{aligned} \frac{\partial}{\partial x} \left(\int_{z_b}^{\eta} \frac{\partial u}{\partial x} dz \right) &= \frac{\partial}{\partial x} \left[\frac{\partial}{\partial x} \left(\int_{z_b}^{\eta} u dz \right) - \frac{\partial u_{\eta}}{\partial x} \frac{\partial \eta}{\partial x} + \frac{\partial u_b}{\partial x} \frac{\partial z_b}{\partial x} \right] = \\ &= \frac{\partial}{\partial x} \left(\frac{\partial(h\bar{u})}{\partial x} \right) + \frac{\partial}{\partial x} \left(-\frac{\partial u_{\eta}}{\partial x} \frac{\partial \eta}{\partial x} + \frac{\partial u_b}{\partial x} \frac{\partial z_b}{\partial x} \right), \end{aligned} \quad (2.33)$$

which, if we consider that the current free surface and the bed are smooth, is reduced to:

$$\frac{\partial}{\partial x} \left(\int_{z_b}^{\eta} \frac{\partial u}{\partial x} dz \right) = \frac{\partial}{\partial x} \left(\frac{\partial(h\bar{u})}{\partial x} \right). \quad (2.34)$$

Analogously, we have:

$$\frac{\partial}{\partial y} \left(\int_{z_b}^{\eta} \frac{\partial u}{\partial y} dz \right) = \frac{\partial}{\partial y} \left(\frac{\partial(h\bar{u})}{\partial y} \right). \quad (2.35)$$

Therefore, replacing the relations (2.34) and (2.35) into Equation (2.32), we get the contribution of the viscous term:

$$\int_{z_b}^{\eta} \frac{\mu}{\rho} \nabla^2 u dz = \frac{\mu}{\rho} \frac{\partial^2(h\bar{u})}{\partial x^2} + \frac{\mu}{\rho} \frac{\partial^2(h\bar{u})}{\partial y^2} - \frac{\tau_{\eta x} + \tau_{b x}}{\rho}. \quad (2.36)$$

Going back to Equation (2.28) and applying the Leibniz rule to its left-hand side, we obtain:

$$\begin{aligned} \int_{z_b}^{\eta} \left(\frac{\partial u}{\partial t} + \frac{\partial u^2}{\partial x} + \frac{\partial(uv)}{\partial y} + \frac{\partial(uw)}{\partial z} \right) dz &= \left[\frac{\partial}{\partial t} \left(\int_{z_b}^{\eta} u dz \right) - u_{\eta} \frac{\partial \eta}{\partial t} + u_b \frac{\partial z_b}{\partial t} \right] + \\ + \left[\frac{\partial}{\partial x} \left(\int_{z_b}^{\eta} u^2 dz \right) - u_{\eta}^2 \frac{\partial \eta}{\partial x} + u_b^2 \frac{\partial z_b}{\partial x} \right] &+ \left[\frac{\partial}{\partial y} \left(\int_{z_b}^{\eta} uv dz \right) - u_{\eta} v_{\eta} \frac{\partial \eta}{\partial y} + u_b v_b \frac{\partial z_b}{\partial y} \right] + \\ + \left[u_{\eta} w_{\eta} - u_b w_b \right] &= \frac{\partial}{\partial t} \left(\int_{z_b}^{\eta} u dz \right) + \frac{\partial}{\partial x} \left(\int_{z_b}^{\eta} u^2 dz \right) + \frac{\partial}{\partial y} \left(\int_{z_b}^{\eta} uv dz \right) + \\ - u_{\eta} \left(\frac{\partial \eta}{\partial t} + u_{\eta} \frac{\partial \eta}{\partial x} + v_{\eta} \frac{\partial \eta}{\partial y} - w_{\eta} \right) &+ u_b \left(\frac{\partial z_b}{\partial t} + u_b \frac{\partial z_b}{\partial x} + v_b \frac{\partial z_b}{\partial y} - w_b \right) = \\ = \frac{\partial(h\bar{u})}{\partial t} + \frac{\partial}{\partial x} \left(\int_{z_b}^{\eta} u^2 dz \right) + \frac{\partial}{\partial y} \left(\int_{z_b}^{\eta} uv dz \right), \end{aligned} \quad (2.37)$$

where we have canceled out some terms due to the kinematic boundary conditions (2.4) and (2.6). To simplify the previous expression, we define the form factors:

$$\zeta_1 = \frac{1}{h\bar{u}^2} \int_{z_b}^{\eta} u^2 dz = 1 + \frac{1}{h} \int_{z_b}^{\eta} \left(1 - \frac{u}{\bar{u}} \right)^2 dz, \quad (2.38)$$

$$\zeta_2 = \frac{1}{h\bar{u}\bar{v}} \int_{z_b}^{\eta} uv dz = 1 + \frac{1}{h} \int_{z_b}^{\eta} \left(1 - \frac{u}{\bar{u}} \right) \left(1 - \frac{v}{\bar{v}} \right) dz, \quad (2.39)$$

so we obtain:

$$\int_{z_b}^{\eta} \left(\frac{\partial u}{\partial t} + \frac{\partial u^2}{\partial x} + \frac{\partial(uv)}{\partial y} + \frac{\partial(uw)}{\partial z} \right) dz = \frac{\partial(h\bar{u})}{\partial t} + \frac{\partial(\zeta_1 h\bar{u}^2)}{\partial x} + \frac{\partial(\zeta_2 h\bar{u}\bar{v})}{\partial y}.$$

Now we assume that ζ_1 and ζ_2 are sufficiently close to 1 so they can be considered equal to the unit. As it can be seen through Equations (2.38) and (2.39), if u and v do not change along the z -direction, we have $u = \bar{u}$ and $v = \bar{v}$ by definition, and thus $\zeta_1 = \zeta_2 = 1$. Also, as another example, if u and v are 20% different from their respective averages \bar{u} and \bar{v} , ζ_1 and ζ_2 would only vary 4%. Consequently, in most cases, it is valid the simplification $\zeta_1 = \zeta_2 = 1$, which results in:

$$\int_{z_b}^{\eta} \left(\frac{\partial u}{\partial t} + \frac{\partial u^2}{\partial x} + \frac{\partial(uv)}{\partial y} + \frac{\partial(uw)}{\partial z} \right) dz = \frac{\partial(h\bar{u})}{\partial t} + \frac{\partial(h\bar{u}^2)}{\partial x} + \frac{\partial(h\bar{u}\bar{v})}{\partial y}. \quad (2.40)$$

Therefore, we can assemble the terms in Equations (2.29), (2.31), (2.36) and (2.40) into (2.28) to define the complete equation for the momentum conservation of the adopted shallow water model along the x -direction:

$$\begin{aligned} \frac{\partial(hu)}{\partial t} + \frac{\partial}{\partial x} \left(hu^2 + \frac{1}{2}g'h^2 \right) + \frac{\partial(huv)}{\partial y} = -g'h \frac{\partial z_b}{\partial x} + \frac{\mu}{\rho} \left(\frac{\partial^2(hu)}{\partial x^2} + \frac{\partial^2(hu)}{\partial y^2} \right) + \\ - \frac{\tau_{\eta_x} + \tau_{b_x}}{\rho} + hb_x, \end{aligned} \quad (2.41)$$

where the bars over the depth-averaged quantities were omitted to simplify the notation. From this point onward, we will always refer to $\mathbf{u} = (u, v)$ as the depth-averaged velocity. Plus, other space dependent vectors and differential operators are also evaluated in a two-dimensional sense. For instance, $\mathbf{x} = (x, y)$ and $\nabla = (\partial/\partial x, \partial/\partial y)$. Following a procedure similar to the one employed to arrive at Equation (2.41), we can also obtain an equation for the balance along the y -direction:

$$\begin{aligned} \frac{\partial(hv)}{\partial t} + \frac{\partial(huv)}{\partial x} + \frac{\partial}{\partial y} \left(hv^2 + \frac{1}{2}g'h^2 \right) = -g'h \frac{\partial z_b}{\partial y} + \frac{\mu}{\rho} \left(\frac{\partial^2(hv)}{\partial x^2} + \frac{\partial^2(hv)}{\partial y^2} \right) + \\ - \frac{\tau_{\eta_y} + \tau_{b_y}}{\rho} + hb_y. \end{aligned} \quad (2.42)$$

2.1.5 Bed Friction Forces

In this section, we identify the frictional shear stresses acting on the current at its free surface and at the bed. In one-dimensional flows, it is acknowledged that the

bottom shear stress τ_b is a quadratic function of the fluid velocity:

$$\tau_b = \rho(u_*)^2. \quad (2.43)$$

From this expression, it comes the definition of the shear velocity $u_* = (\tau_b/\rho)^{1/2}$, which is an alternative quantity for expressing the friction in velocity units. Moreover, according to SOULSBY [67], the shear stress is related to the depth-averaged current velocity through the relation:

$$\tau_b = \rho c'_d u^2, \quad (2.44)$$

where $c'_d = gn^2 h^{-1/3}$ is a non-dimensional drag coefficient based on the Manning coefficient n . As an aid to choose plausible values for n , works such as the one from ARCEMENT and SCHNEIDER [68] can be seen. For two-dimensional shallow water models, TAN [63] defines:

$$\boldsymbol{\tau}_b = \rho c'_d \|\mathbf{u}\| \mathbf{u}. \quad (2.45)$$

This stress can be written as a function of the specific discharge \mathbf{q} :

$$\boldsymbol{\tau}_b = \rho c_d \|\mathbf{q}\| \mathbf{q}, \quad (2.46)$$

with $c_d = gn^2 h^{-7/3}$.

Furthermore, we use the approach employed by KUBO and NAKAJIMA [69] and DE LUNA *et al.* [30] that defines the friction between the current and the ambient fluid as a fraction of the shear stress at the bed:

$$\boldsymbol{\tau}_\eta = \alpha \boldsymbol{\tau}_b. \quad (2.47)$$

In this case, α is the ratio of the drag forces at the top of the flow and at the bed. We adopt $\alpha = 0.43$, as suggested by KUBO and NAKAJIMA [69] and KOMAR [70]. However, ideally, α should vary with the flow's Froude number [71]. Although we employ the same parameterization for the bed and free surface stresses, BÁRCENAS *et al.* [72] have improved this approximation by considering different rules for each contribution. At last, in this work, to simplify the notation, we combine the effects of the friction at the bottom and at the top of the current into a single term:

$$\boldsymbol{\tau} = \boldsymbol{\tau}_\eta + \boldsymbol{\tau}_b = \rho \gamma \mathbf{q}, \quad (2.48)$$

with $\gamma = (1 + \alpha)c_d \|\mathbf{q}\|$.

2.1.6 Equation System in Convection-Diffusion Form

We can write the equation system composed of Equations (2.27), (2.41) and (2.42) in the conservation form:

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F} = \mathbf{S}_{\text{con}}, \quad (2.49)$$

with:

$$\mathbf{U} = \begin{bmatrix} h \\ hu \\ hv \end{bmatrix} = \begin{bmatrix} h \\ q_x \\ q_y \end{bmatrix}, \quad \mathbf{S}_{\text{con}} = \begin{bmatrix} 0 \\ -g'h \frac{\partial z_b}{\partial x} + \frac{\mu}{\rho} \left(\frac{\partial^2 q_x}{\partial x^2} + \frac{\partial^2 q_x}{\partial y^2} \right) - \gamma q_x \\ -g'h \frac{\partial z_b}{\partial y} + \frac{\mu}{\rho} \left(\frac{\partial^2 q_y}{\partial x^2} + \frac{\partial^2 q_y}{\partial y^2} \right) - \gamma q_y \end{bmatrix}, \quad (2.50)$$

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_x & \mathbf{F}_y \end{bmatrix}, \quad \mathbf{F}_x = \begin{bmatrix} hu \\ hu^2 + \frac{1}{2}g'h^2 \\ huv \end{bmatrix}, \quad \mathbf{F}_y = \begin{bmatrix} hv \\ huv \\ hv^2 + \frac{1}{2}g'h^2 \end{bmatrix}, \quad (2.51)$$

where \mathbf{U} is the vector of conservation variables, \mathbf{F}_x and \mathbf{F}_y are hydraulic fluxes, and \mathbf{S}_{con} is a source term. This system can also be cast as a quasi-linear equation if we employ the differentiation chain rule to get:

$$\frac{\partial \mathbf{F}_x}{\partial x} = \frac{\partial \mathbf{F}_x}{\partial \mathbf{U}} \frac{\partial \mathbf{U}}{\partial x} = \mathbf{A}_x \frac{\partial \mathbf{U}}{\partial x} \quad \text{and} \quad \frac{\partial \mathbf{F}_y}{\partial y} = \frac{\partial \mathbf{F}_y}{\partial \mathbf{U}} \frac{\partial \mathbf{U}}{\partial y} = \mathbf{A}_y \frac{\partial \mathbf{U}}{\partial y}, \quad (2.52)$$

where \mathbf{A}_x and \mathbf{A}_y are flux Jacobian matrices. We provide some insights on the computation of the derivatives with respect to \mathbf{U} in Appendix A.1. Besides, after some manipulations, the set of equations that govern the flow can be given in the form of a generalized convection-diffusion equation:

$$\frac{\partial \mathbf{U}}{\partial t} + (\mathbf{A} \cdot \nabla) \mathbf{U} - \nabla \cdot (\mathbf{K} \nabla \mathbf{U}) = \mathbf{S}, \quad (2.53)$$

where:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_x & \mathbf{A}_y \end{bmatrix}, \quad \mathbf{A}_x = \begin{bmatrix} 0 & 1 & 0 \\ g'h - u^2 & 2u & 0 \\ -uv & v & u \end{bmatrix}, \quad \mathbf{A}_y = \begin{bmatrix} 0 & 0 & 1 \\ -uv & v & u \\ g'h - v^2 & 0 & 2v \end{bmatrix}, \quad (2.54)$$

$$\mathbf{K} = \frac{\mu}{\rho} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{S} = \begin{bmatrix} 0 \\ -g' \frac{\partial z_b}{\partial x} h - \gamma q_x \\ -g' \frac{\partial z_b}{\partial y} h - \gamma q_y \end{bmatrix}. \quad (2.55)$$

We remark that this system can also be used to simulate classic shallow water flows (instead of turbidity currents) if we set $g' = g$, $\rho = \rho_w$ and compute γ with $\alpha = 0$.

2.2 Initial Value Problem

The initial value problem in question is: given the closed domain $\bar{\Omega} \in \mathbb{R}^2$, of interior region Ω and boundary $\Gamma = \Gamma_e \cup \Gamma_n$, with $\Gamma_e \cap \Gamma_n = \emptyset$, we solve the system of equations (2.53) for $\mathbf{U}(\mathbf{x}, t)$, $\forall t \in]0, t_f]$, subject to the initial condition:

$$\mathbf{U}(\mathbf{x}, 0) = \mathbf{U}^0(\mathbf{x}), \quad (2.56)$$

and to the essential and natural boundary conditions:

$$\mathbf{U} = \mathbf{G} \text{ on } \Gamma_e, \quad (2.57)$$

$$\mathbf{K}(\mathbf{n} \cdot \nabla) \mathbf{U} = \mathbf{0} \text{ on } \Gamma_n, \quad (2.58)$$

where \mathbf{n} is the outward-pointing unit normal at the boundary. In (2.58), the diffusive flux across the boundary Γ_n is null. This is a common approach in convective-diffusive physical models. In this case, all the flow across the boundary is advective. Another type of boundary condition that can be employed is the non-penetration condition $\mathbf{q} \cdot \mathbf{n} = 0$ on Γ_e . When the associated contour is aligned with a coordinate axis, we simply impose that the component of the specific discharge orthogonal to it is zero-valued. Otherwise, we take advantage of the deal.II's `VectorTools::compute_no_normal_flux_constraints(...)` function. Thus, we refer to the work of BANGERTH and KAYSER-HEROLD [73] for further details on how these constraints are applied.

To obtain the variational formulation of the problem, we initially define the sets of test and trial functions respectively as $\mathcal{S} = \{\mathbf{U} \in (H^1(\bar{\Omega}))^3 \mid \mathbf{U} = \mathbf{G} \text{ on } \Gamma_e\}$ and $\mathcal{V} = \{\mathbf{W} \in (H^1(\bar{\Omega}))^3 \mid \mathbf{W} = \mathbf{0} \text{ on } \Gamma_e\}$, where $H^1(\bar{\Omega})$ is the first order Hilbert space, defined in the closed domain $\bar{\Omega}$. Then, using the weighted residuals method, the solution of the differential equation (2.53) can be computed by solving:

$$\int_{\Omega} \mathbf{W}^T \mathbf{R} d\Omega = \mathbf{0}, \quad (2.59)$$

where:

$$\mathbf{R} = \frac{\partial \mathbf{U}}{\partial t} + (\mathbf{A} \cdot \nabla) \mathbf{U} - \nabla \cdot (\mathbf{K} \nabla \mathbf{U}) - \mathbf{S} \quad (2.60)$$

is the residual of the system (2.53). Here, the diffusive term can be integrated by parts:

$$\int_{\Omega} \mathbf{W}^T \nabla \cdot (\mathbf{K} \nabla \mathbf{U}) d\Omega = \int_{\Gamma} \mathbf{W}^T \mathbf{K}(\mathbf{n} \cdot \nabla) \mathbf{U} d\Gamma - \int_{\Omega} \nabla \mathbf{W}^T \cdot (\mathbf{K} \nabla \mathbf{U}) d\Omega. \quad (2.61)$$

As $\mathbf{W} = \mathbf{0}$ on Γ_e :

$$\int_{\Omega} \mathbf{W}^T \nabla \cdot (\mathbf{K} \nabla \mathbf{U}) d\Omega = \int_{\Gamma_n} \mathbf{W}^T \mathbf{K} (\mathbf{n} \cdot \nabla) \mathbf{U} d\Gamma_n - \int_{\Omega} \nabla \mathbf{W}^T \cdot (\mathbf{K} \nabla \mathbf{U}) d\Omega, \quad (2.62)$$

and, using the natural boundary conditions (2.58):

$$\int_{\Omega} \mathbf{W}^T \nabla \cdot (\mathbf{K} \nabla \mathbf{U}) d\Omega = - \int_{\Omega} \nabla \mathbf{W}^T \cdot (\mathbf{K} \nabla \mathbf{U}) d\Omega. \quad (2.63)$$

Thus, Equation (2.59) becomes:

$$\int_{\Omega} \mathbf{W}^T \left(\frac{\partial \mathbf{U}}{\partial t} + (\mathbf{A} \cdot \nabla) \mathbf{U} - \mathbf{S} \right) d\Omega + \int_{\Omega} \nabla \mathbf{W}^T \cdot (\mathbf{K} \nabla \mathbf{U}) d\Omega = 0. \quad (2.64)$$

In sequence, consider the finite-dimensional spaces of trial and test functions respectively defined as $\mathcal{S}^h = \{\mathbf{U}^h \in (H^{1h}(\bar{\Omega}))^3 \mid \mathbf{U}^h = \mathbf{G}$ on $\Gamma_e\}$ and $\mathcal{V}^h = \{\mathbf{W}^h \in (H^{1h}(\bar{\Omega}))^3 \mid \mathbf{W}^h = \mathbf{0}$ on $\Gamma_e\}$, with $\mathcal{S}^h \subset \mathcal{S}$ and $\mathcal{V}^h \subset \mathcal{V}$. Here, $H^{1h}(\bar{\Omega})$ is the finite-dimensional first-order Hilbert space, specified in the closed domain $\bar{\Omega}$. Then, we can perform a finite-dimensional approximation to the residual (2.60):

$$\mathbf{R}^h = \frac{\partial \mathbf{U}^h}{\partial t} + (\mathbf{A}^h \cdot \nabla) \mathbf{U}^h - \nabla \cdot (\mathbf{K} \nabla \mathbf{U}^h) - \mathbf{S}^h, \quad (2.65)$$

where the forms \mathbf{A}^h and \mathbf{S}^h indicate that their respective matrices should be computed based on \mathbf{U}^h . Similarly, Equation (2.64) can be brought to the finite-dimensional space:

$$\int_{\Omega} (\mathbf{W}^h)^T \left(\frac{\partial \mathbf{U}^h}{\partial t} + (\mathbf{A}^h \cdot \nabla) \mathbf{U}^h - \mathbf{S}^h \right) d\Omega + \int_{\Omega} \nabla (\mathbf{W}^h)^T \cdot (\mathbf{K} \nabla \mathbf{U}^h) d\Omega = 0. \quad (2.66)$$

Next, employing the standard Galerkin method, we initially consider $\mathbf{U}^h = \mathbf{V} \mathbf{N}$ and $\mathbf{W}^h = \mathbf{C} \mathbf{N}$, where \mathbf{V} contains the nodal values of the solution, \mathbf{C} defines arbitrary constants and \mathbf{N} holds the finite element basis (or shape) functions. For example, in the case of a finite space of dimension 3, we would have:

$$\mathbf{U}^h = \begin{bmatrix} h^h \\ q_x^h \\ q_y^h \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 \\ q_{x1} & q_{x2} & q_{x3} \\ q_{y1} & q_{y2} & q_{y3} \end{bmatrix} \begin{bmatrix} N_1 \\ N_2 \\ N_3 \end{bmatrix} = \begin{bmatrix} N_1 h_1 + N_2 h_2 + N_3 h_3 \\ N_1 q_{x1} + N_2 q_{x2} + N_3 q_{x3} \\ N_1 q_{y1} + N_2 q_{y2} + N_3 q_{y3} \end{bmatrix}, \quad (2.67)$$

where the subscripts indicate to which node each value is associated. To represent the multidimensional nodal solutions in a single vector, which will effectively be a linear system solution, we employ known properties of the vectorization operator

$\text{vec}(\cdot)$, introduced in Appendix A.2, to write:

$$\mathbf{U}^h = \text{vec}(\mathbf{U}^h) = \text{vec}(\mathbf{V}\mathbf{N}) = (\mathbf{N}^T \otimes \mathbf{I}_3) \text{vec}(\mathbf{V}), \quad (2.68)$$

where \mathbf{I}_3 is the third-order identity matrix, and \otimes denotes the Kronecker product, which we describe in Appendix A.3. For example, using an element with three nodes, we would have:

$$\begin{aligned} \mathbf{U}^h &= \begin{bmatrix} h^h \\ q_x^h \\ q_y^h \end{bmatrix} = \left(\begin{bmatrix} N_1 & N_2 & N_3 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right) \text{vec} \left(\begin{bmatrix} h_1 & h_2 & h_3 \\ q_{x1} & q_{x2} & q_{x3} \\ q_{y1} & q_{y2} & q_{y3} \end{bmatrix} \right) = \\ &= \begin{bmatrix} N_1 & 0 & 0 & N_2 & 0 & 0 & N_3 & 0 & 0 \\ 0 & N_1 & 0 & 0 & N_2 & 0 & 0 & N_3 & 0 \\ 0 & 0 & N_1 & 0 & 0 & N_2 & 0 & 0 & N_3 \end{bmatrix} \begin{bmatrix} h_1 \\ q_{x1} \\ q_{y1} \\ h_2 \\ q_{x2} \\ q_{y2} \\ h_3 \\ q_{x3} \\ q_{y3} \end{bmatrix} = \begin{bmatrix} N_1 h_1 + N_2 h_2 + N_3 h_3 \\ N_1 q_{x1} + N_2 q_{x2} + N_3 q_{x3} \\ N_1 q_{y1} + N_2 q_{y2} + N_3 q_{y3} \end{bmatrix}. \end{aligned} \quad (2.69)$$

Note, however, that the elements we use in this work have four nodes (i.e., they are quadrilaterals) because the deal.II library does not support triangular elements. Based on the previous transformation, for the deductions that follow, we will employ the definitions:

$$\mathbf{U}^h = \mathbf{N}\mathbf{V}, \quad \mathbf{W}^h = \mathbf{N}\mathbf{C}, \quad (2.70)$$

where:

$$\mathbf{N} = \begin{bmatrix} \mathbf{N}_1 & \mathbf{N}_2 & \dots & \mathbf{N}_{d_S} \end{bmatrix}, \quad \mathbf{N}_i = N_i \mathbf{I}_3, \quad (2.71)$$

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_1^T & \mathbf{V}_2^T & \dots & \mathbf{V}_{d_S}^T \end{bmatrix}^T, \quad \mathbf{V}_i = \begin{bmatrix} h_i & q_{xi} & q_{yi} \end{bmatrix}^T, \quad (2.72)$$

and \mathbf{C} contains arbitrary constants, while d_S is the dimension of the space \mathcal{S}^h .

Therefore, we can expand Equation (2.66) to obtain:

$$\begin{aligned} &\left(\mathbf{C}^T \int_{\Omega} \mathbf{N}^T \mathbf{N} d\Omega \right) \frac{\partial \mathbf{V}}{\partial t} + \left(\mathbf{C}^T \int_{\Omega} \mathbf{N}^T (\mathbf{A}^h \cdot \nabla) \mathbf{N} d\Omega \right) \mathbf{V} + \\ &+ \left(\mathbf{C}^T \int_{\Omega} \nabla \mathbf{N}^T \cdot (\mathbf{K} \nabla \mathbf{N}) d\Omega \right) \mathbf{V} = \mathbf{C}^T \int_{\Omega} \mathbf{N}^T \mathbf{S}^h d\Omega. \end{aligned} \quad (2.73)$$

With suitable choices for the constants in \mathbf{C} , we can define the semi-discrete system:

$$\mathbf{M} \frac{\partial \mathbf{V}}{\partial t} = -\mathbf{D}\mathbf{V} + \mathbf{F}, \quad (2.74)$$

where \mathbf{M} and \mathbf{D} are generalized mass and stiffness matrices, and \mathbf{F} is a source term. In this case:

$$\mathbf{M} = \int_{\Omega} \mathbf{N}^T \mathbf{N} d\Omega, \quad \mathbf{F} = \int_{\Omega} \mathbf{N}^T \mathbf{S}^h d\Omega, \quad (2.75)$$

$$\mathbf{D} = \int_{\Omega} [\mathbf{N}^T (\mathbf{A}^h \cdot \nabla) \mathbf{N} + \nabla \mathbf{N}^T \cdot (\mathbf{K} \nabla \mathbf{N})] d\Omega. \quad (2.76)$$

Here, it should be noted that \mathbf{D} and \mathbf{F} depend on \mathbf{V} , which confers nonlinearity to the system. Next, we let the finite element basis functions be locally determined piecewise linear functions with compact support. In this sense, the shape function associated with a mesh node is non-null only inside the elements surrounding the node. Therefore, we can write the matrices of the dynamic system (2.74) as sums of element contributions:

$$\mathbf{M} = \sum_{e=1}^{n_{el}} \int_{\Omega^e} \mathbf{N}^T \mathbf{N} d\Omega^e, \quad \mathbf{F} = \sum_{e=1}^{n_{el}} \int_{\Omega^e} \mathbf{N}^T \mathbf{S}^h d\Omega^e, \quad (2.77)$$

$$\mathbf{D} = \sum_{e=1}^{n_{el}} \int_{\Omega^e} [\mathbf{N}^T (\mathbf{A}^h \cdot \nabla) \mathbf{N} + \nabla \mathbf{N}^T \cdot (\mathbf{K} \nabla \mathbf{N})] d\Omega^e, \quad (2.78)$$

where Ω^e , $e = 1, \dots, n_{el}$ are the n_{el} elements that comprise the domain Ω , with $\Omega^i \cap \Omega^j = \emptyset, \forall (i, j)$. At last, the discrete problem specification is completed by considering discrete versions of the initial and boundary conditions defined in Equations (2.56)-(2.58):

$$\mathbf{V}(0) = \mathbf{V}^0, \quad (2.79)$$

$$\mathbf{V} = \mathbf{G} \text{ on } \Gamma_e \times]0, t_f], \quad (2.80)$$

$$\mathbf{K} (\mathbf{n} \cdot \nabla) \mathbf{U}^h = \mathbf{0} \text{ on } \Gamma_n \times]0, t_f]. \quad (2.81)$$

Besides, from this point onward, to simplify notation, we drop the superscript h from the components of the finite element approximation to the solution \mathbf{U} , so we have $\mathbf{U}^h = \begin{bmatrix} h & q_x & q_y \end{bmatrix}^T$. Do observe that this notation differs from the one used to represent the components of the nodal value solutions - introduced in Equation (2.72) - by not having subscripts.

2.3 Stabilized Formulation

For the stabilized formulation, the test function space is augmented using the Petrov-Galerkin method, by which $\hat{\mathbf{W}} = \mathbf{W} + \mathbf{P}_{\mathcal{G}}$. Following the Streamline Upwind Petrov-Galerkin (SUPG) technique, we incorporate effects associated with the convective term as we define $\mathbf{P}_{\mathcal{G}} = \tau_{\text{SUPG}}(\mathbf{A} \cdot \nabla)\mathbf{W}$. This additional function can be interpreted as a directional derivative associated with the convection matrix \mathbf{A} , having the coefficient τ_{SUPG} as an intrinsic time scale. Thus, the weighted residuals equation (2.59) becomes:

$$\int_{\Omega} \hat{\mathbf{W}}^T \mathbf{R} d\Omega = \mathbf{0} \quad (2.82)$$

$$\therefore \int_{\Omega} [\mathbf{W} + \tau_{\text{SUPG}}(\mathbf{A} \cdot \nabla)\mathbf{W}]^T \mathbf{R} d\Omega = \mathbf{0}. \quad (2.83)$$

Because \mathbf{W} has continuity C^0 and $\hat{\mathbf{W}}$ depends on $\nabla\mathbf{W}$, $\hat{\mathbf{W}}$ is discontinuous. For the same reason, we cannot integrate by parts the SUPG contribution.

In complement to the SUPG term, a shock-capturing (or discontinuity-capturing) contribution is also added to the weighting function to better represent sharp shocks, obtaining smooth but crisp solution profiles near discontinuities. This new term acts in the direction of the solution gradient instead of the streamline direction [74]. Its contribution can be included in Equation (2.83) by adding, to its left-hand side, the operator:

$$\int_{\Omega} \delta_{\text{Shock}} \nabla \mathbf{W}^T \cdot \nabla U^h d\Omega,$$

where δ_{Shock} is the shock-capturing operator's coefficient.

Therefore, the stabilized formulation is obtained by adding, to their respective counterparts in Equations (2.77) and (2.78), the matrices:

$$\mathbf{M}_{\text{SUPG}} = \sum_{e=1}^{n_{el}} \int_{\Omega^e} [\tau_{\text{SUPG}}(\mathbf{A}^h \cdot \nabla)\mathbf{N}]^T \mathbf{N} d\Omega^e, \quad (2.84)$$

$$\mathbf{F}_{\text{SUPG}} = \sum_{e=1}^{n_{el}} \int_{\Omega^e} [\tau_{\text{SUPG}}(\mathbf{A}^h \cdot \nabla)\mathbf{N}]^T \mathbf{S}^h d\Omega^e, \quad (2.85)$$

$$\mathbf{D}_{\text{SUPG}} = \sum_{e=1}^{n_{el}} \int_{\Omega^e} [\tau_{\text{SUPG}}(\mathbf{A}^h \cdot \nabla)\mathbf{N}]^T [(\mathbf{A}^h \cdot \nabla)\mathbf{N} - \mathbf{K}\nabla^2\mathbf{N}] d\Omega^e, \quad (2.86)$$

$$\mathbf{D}_{\text{Shock}} = \sum_{e=1}^{n_{el}} \int_{\Omega^e} \delta_{\text{Shock}} \nabla \mathbf{N}^T \cdot \nabla \mathbf{N} d\Omega^e. \quad (2.87)$$

The adopted SUPG technique, which was proposed by TEZDUYAR [51] and

adapted by TAKASE *et al.* [52] to the shallow water equations, defines:

$$\tau_{\text{SUPG}} = \left[\frac{1}{(\tau_{\text{SUGN1}})^2} + \frac{1}{(\tau_{\text{SUGN2}})^2} \right]^{-1/2}, \quad (2.88)$$

with:

$$\tau_{\text{SUGN1}} = \left(\sum_{a=1}^{n_{\text{npe}}} c |\mathbf{j} \cdot \nabla N_a| + |\mathbf{u} \cdot \nabla N_a| \right)^{-1}, \quad \tau_{\text{SUGN2}} = \frac{\Delta t}{2}, \quad (2.89)$$

where Δt is the current time step, $n_{\text{npe}} = 4$ is the number of nodes per element, $c = \sqrt{g'h}$ is the propagation speed of a perturbation on the current surface, and $\mathbf{j} = \nabla\eta/|\nabla\eta|$ is the normalized gradient of the current free surface elevation.

As a complement, we use the Consistent Approximate Upwind (CAU) shock-capturing operator introduced by GALEÃO and DO CARMO [42] to solve the scalar advection-diffusion equation, and later extended to the compressible Euler and Navier-Stokes equations by ALMEIDA and GALEÃO [43]. Because the operator is defined in terms of entropy variables, first consider the transformation from entropy to conservation variables $T_S : \mathbf{U}_S \rightarrow \mathbf{U}$, whose Jacobian matrix is $\tilde{\mathbf{A}}_0$, i.e., $\nabla\mathbf{U} = \tilde{\mathbf{A}}_0 \nabla\mathbf{U}_S$. Then, we can write the residual (2.60) as a function of the entropy variables \mathbf{U}_S :

$$\mathbf{R} = \tilde{\mathbf{A}}_0 \frac{\partial \mathbf{U}_S}{\partial t} + \left(\mathbf{A} \tilde{\mathbf{A}}_0 \cdot \nabla \right) \mathbf{U}_S - \nabla \cdot \left(\mathbf{K} \tilde{\mathbf{A}}_0 \nabla \mathbf{U}_S \right) - \mathbf{S} = \mathbf{R}_S, \quad (2.90)$$

which is in the same form used by ALMEIDA and GALEÃO [43]. Additionally, in the original definition, the matrix $\tilde{\mathbf{A}}_0$ (and also its inverse $\tilde{\mathbf{A}}_0^{-1}$) plays the role of a metric tensor in \mathbb{R}^3 , being employed, for instance, in the norm $\|\nabla\mathbf{V}^h\|_{\mathbf{A}_0}$. Here, for a vector \mathbf{v} and a matrix \mathbf{M} , $\|\mathbf{v}\|_{\mathbf{M}}$ denotes the norm $(\mathbf{v}^T \mathbf{M} \mathbf{v})^{1/2}$. Such a norm of the gradient of the entropy variables can be given in terms of the conservation quantities:

$$\begin{aligned} \|\nabla\mathbf{U}_S^h\|_{\mathbf{A}_0} &= \left(\nabla\mathbf{U}_S^h \right)^T \tilde{\mathbf{A}}_0 \left(\nabla\mathbf{U}_S^h \right) = \left(\tilde{\mathbf{A}}_0^{-1} \nabla\mathbf{U}^h \right)^T \tilde{\mathbf{A}}_0 \left(\tilde{\mathbf{A}}_0^{-1} \nabla\mathbf{U}^h \right) = \\ &= \left(\nabla\mathbf{U}^h \right)^T \tilde{\mathbf{A}}_0^{-1} \left(\nabla\mathbf{U}^h \right) = \|\nabla\mathbf{U}^h\|_{\mathbf{A}_0^{-1}}. \end{aligned} \quad (2.91)$$

Therefore, considering the transformation T_S , the Equations (2.90) and (2.91) and assuming the flow has a high Péclet number, we can write, in terms of the conservation variables, the coefficient of the CAU shock-capturing operator to be

applied to the semi-discrete system (2.74):

$$\delta_{\text{Shock}} = \max \left\{ 0, \frac{\|\mathbf{R}^h\|_{\mathbf{A}_0^{-1}}}{\|\hat{\nabla}\mathbf{U}^h\|_{\mathbf{A}_0^{-1}}} - \frac{[\tau_{\text{SUPG}}(\mathbf{A}^h \cdot \nabla)\mathbf{U}^h]^T \mathbf{R}^h}{\|\nabla\mathbf{U}^h\|_{\mathbf{A}_0^{-1}}^2} \right\}, \quad (2.92)$$

with:

$$\|\nabla\mathbf{U}^h\|_{\mathbf{A}_0^{-1}}^2 = \left\| \frac{\partial\mathbf{U}^h}{\partial x} \right\|_{\mathbf{A}_0^{-1}}^2 + \left\| \frac{\partial\mathbf{U}^h}{\partial y} \right\|_{\mathbf{A}_0^{-1}}^2, \quad (2.93)$$

$$\|\hat{\nabla}\mathbf{U}^h\|_{\mathbf{A}_0^{-1}} = \left(\left\| \frac{\partial\hat{x}}{\partial x} \frac{\partial\mathbf{U}^h}{\partial x} + \frac{\partial\hat{x}}{\partial y} \frac{\partial\mathbf{U}^h}{\partial y} \right\|_{\mathbf{A}_0^{-1}}^2 + \left\| \frac{\partial\hat{y}}{\partial x} \frac{\partial\mathbf{U}^h}{\partial x} + \frac{\partial\hat{y}}{\partial y} \frac{\partial\mathbf{U}^h}{\partial y} \right\|_{\mathbf{A}_0^{-1}}^2 \right)^{1/2}, \quad (2.94)$$

where $\hat{\mathbf{x}} = (\hat{x}, \hat{y})$ is a position in the canonical/local reference system. Thus, the derivatives of $\hat{\mathbf{x}}$ are entries of the Jacobian matrix of the transformation between the real and the canonical coordinate systems. When computing the coefficient, we assign $\delta_{\text{Shock}} = 0$ if $\|\nabla\mathbf{U}^h\|_{\mathbf{A}_0^{-1}}^2 = 0$. Also, we observed that the operator produces better results if we neglect the transient term in the residual \mathbf{R}^h . In fact, profiles obtained with the modified residual are smoother, while the use of its complete form (2.65) best represents discontinuities at the expense of having some spurious artifacts in the produced solutions. Thus, we employ the modified residual in all examples presented in this work.

To complete the definition of the operator, we still need to define the matrix $\tilde{\mathbf{A}}_0^{-1}$. Hence we consider the energy functional:

$$E = \frac{g'h^2 + u^2h + v^2h}{2}, \quad (2.95)$$

from which the entropy variables \mathbf{U}_S and the matrix $\tilde{\mathbf{A}}_0^{-1}$ can be defined [75]:

$$\mathbf{U}_S = \frac{\partial E}{\partial \mathbf{U}} = \begin{bmatrix} g'h - \frac{u^2 + v^2}{2} \\ u \\ v \end{bmatrix}, \quad (2.96)$$

$$\tilde{\mathbf{A}}_0^{-1} = \frac{\partial \mathbf{U}_S}{\partial \mathbf{U}} = \frac{1}{h} \begin{bmatrix} g'h + u^2 + v^2 & -u & -v \\ -u & 1 & 0 \\ -v & 0 & 1 \end{bmatrix}. \quad (2.97)$$

Here, the derivatives with respect to \mathbf{U} were computed in the same manner as discussed in Appendix A.1. As we show in Appendix B.1, $\tilde{\mathbf{A}}_0^{-1}$ is symmetric positive definite if $h > 0$, which should be true by definition. Also, to compute the coefficient (2.92), we determine $\tilde{\mathbf{A}}_0^{-1}$ based on the components of \mathbf{U}^h .

As an alternative to the CAU technique, we employ the $YZ\beta$ [53] shock-capturing operator, whose coefficient is:

$$\delta_{\text{Shock}} = \frac{\delta_{\beta=1} + \delta_{\beta=2}}{2}, \quad (2.98)$$

with:

$$\delta_{\beta} = \|\mathbf{Y}^{-1}\mathbf{Z}\| \left(\left\| \mathbf{Y}^{-1} \frac{\partial \mathbf{U}^h}{\partial x} \right\|^2 + \left\| \mathbf{Y}^{-1} \frac{\partial \mathbf{U}^h}{\partial y} \right\|^2 \right)^{0.5\beta-1} \|\mathbf{Y}^{-1}\mathbf{U}^h\|^{1-\beta} \left(\frac{h_{\text{Shock}}}{2} \right)^{\beta}, \quad (2.99)$$

$$\mathbf{Y} = \begin{bmatrix} (h)_{\text{Ref}} & 0 & 0 \\ 0 & (q_x)_{\text{Ref}} & 0 \\ 0 & 0 & (q_y)_{\text{Ref}} \end{bmatrix}, \quad \mathbf{Z} = (\mathbf{A}^h \cdot \nabla) \mathbf{U}^h, \quad (2.100)$$

$$h_{\text{Shock}} = 2 \left(\sum_{a=1}^{n_{\text{np}}} |\mathbf{j} \cdot \nabla N_a| \right)^{-1}. \quad (2.101)$$

In this case, $(h)_{\text{Ref}}$, $(q_x)_{\text{Ref}}$ and $(q_y)_{\text{Ref}}$ are reference values for the variables h , q_x and q_y . Alternatively, it could be used $\delta_{\text{Shock}} = \delta_{\beta=1}$ or $\delta_{\text{Shock}} = \delta_{\beta=2}$, for smoother or sharper shocks, respectively. Recently, TEN EIKELDER *et al.* [76] showed the connection of the $YZ\beta$ operator to the variational multiscale method.

Following a semi-discrete approach, we use the predictor/multi-corrector algorithm introduced by ALIABADI and TEZDUYAR [77] to advance nodal values over a time step $\Delta t = \Delta t^n = t^{n+1} - t^n$, from a state \mathbf{V}^n to the next \mathbf{V}^{n+1} . Here the superscripts n and $n+1$ are used denote the current and next states of any given quantity. We summarize the technique in Algorithm 1 and provide a full deduction in Appendix D. In this algorithm, $\theta \in [0, 1]$ is a parameter that controls the stability and precision of the method. $\theta = 0$ characterizes the forward Euler method, $\theta = 1$ defines the implicit backward Euler and $\theta = 1/2$ constitutes the trapezoidal rule. In this work, we employ $\theta = 1/2$. Also, we stop the nonlinear correction iterations if the l^2 -norm of linear system's residual is less than TOL_1 times its value at the first iteration, or the relative difference between its current and previous values is less than TOL_2 . Here we set $TOL_1 = 10^{-4}$ and $TOL_2 = 10^{-6}$, while limiting the number of iterations to 100. The linear system is solved with the iterative GMRes (Generalized Minimal Residual) algorithm [78], using a Krylov space of dimension 35 and the ILU(0) preconditioner. To speed up the execution of the linear system's solver, we reorder the degrees of freedom using the algorithm proposed by CUTHILL and MCKEE [79].

Algorithm 1 Predictor/multi-corrector algorithm employed for the time integration in the stabilized method. Here we assign $\mathbf{A}_c = \partial \mathbf{V} / \partial t$.

- Prediction phase:
 - 1: $\mathbf{A}_c^{(0)} = \mathbf{0}$.
 - 2: $\mathbf{V}^{(0)} = \mathbf{V}^n + (1 - \theta)\Delta t \mathbf{A}_c^n$.
 - Correction phase
 - 3: **for** $m = 0, 1, 2, \dots$ until the convergence criteria is met, **do**:
 - 4: $\mathbf{R}^{(m)} = -(\mathbf{M} \mathbf{A}_c^{(m)} + \mathbf{D}^{(m)} \mathbf{V}^{(m)}) + \mathbf{F}^{(m)}$.
 - 5: $\mathbf{M}^* = \mathbf{M} + \theta \Delta t \mathbf{D}^{(m)}$.
 - 6: Solve $\mathbf{M}^*(\Delta \mathbf{A}_c^{(m)}) = \mathbf{R}^{(m)}$.
 - 7: $\mathbf{A}_c^{(m+1)} = \mathbf{A}_c^{(m)} + \Delta \mathbf{A}_c^{(m)}$.
 - 8: $\mathbf{V}^{(m+1)} = \mathbf{V}^{(m)} + \theta \Delta t \Delta \mathbf{A}_c^{(m)}$.
 - 9: **end for**.
 - 10: $\mathbf{A}_c^{n+1} = \mathbf{A}_c^{(m+1)}$
 - 11: $\mathbf{V}^{n+1} = \mathbf{V}^{(m+1)}$
-

2.4 FCT Scheme

The semi-discrete system (2.74) is also numerically solved by a flux-corrected transport scheme [44, 45, 48]. The main idea is to adaptively swap a high-order method that might have spurious oscillations and a low-order approach that is both non-oscillatory and positivity preserving. In this work, the addition of anti-diffusive fluxes to the low-order system of equations recovers the high-order method. However, we scale these fluxes to optimize the transition between both methods, improving the overall stability and accuracy.

2.4.1 Positivity Preservation

Before venturing into the low-order method, it is necessary to understand the properties it must hold. The main concern is that the numerical scheme should respect the positivity constraints intrinsic to the continuous physical model. That is, it should not produce nonphysical negative values. In our case, the current thickness should always be non-negative.

However, to be complete and, at the same time, provide a clearer explanation for the multidimensional system (2.74), we have chosen to first examine an scalar analogous equation. In this case, we consider that $\mathbf{M} = \{m_{ij}\}$, $\mathbf{V} = \{v_i\}$, $\mathbf{D} = \{d_{ij}\}$ and $\mathbf{F} = \{f_i\}$, obtaining:

$$\sum_j m_{ij} \frac{\partial v_j}{\partial t} = - \sum_j d_{ij} v_j + f_i, \quad \forall i. \quad (2.102)$$

Initially, we replace the consistent mass matrix \mathbf{M} with its diagonal counterpart

\mathbf{M}_L , formed by summing and lumping the values in each row on the diagonal:

$$\mathbf{M}_L = \text{diag}\{m_i\}, \quad \text{with} \quad m_i = \sum_j m_{ij}, \quad \forall i, \quad (2.103)$$

where the operator $\text{diag}(\cdot)$ stands for the diagonal matrix whose diagonal entries, ordered from the upper left corner, are given by its operand. So, we have:

$$m_i \frac{\partial v_i}{\partial t} = - \sum_j d_{ij} v_j + f_i, \quad \forall i. \quad (2.104)$$

This procedure can be used to get a diagonal mass matrix and obtain an explicit solution for the semi-discrete system. HANSBO [80] showed that a lumped mass matrix only preserves mass in flow conservation laws if it is built by the adopted row-sum strategy. Indeed, it preserves mass in the sense that, for bilinear elements:

$$\sum_i m_i v_i = \sum_i \sum_j m_{ij} v_j, \quad \forall i, \quad (2.105)$$

as shown in Appendix B.2. The effect of mass lumping corresponds to adding artificial diffusion to the system [80], which should aid in suppressing undesired oscillations. Moreover, because the basis functions of bilinear elements are always positive inside the elements, $m_i > 0$, $\forall i$, which is a property explored in the deductions that follow.

In addition, observe that $\mathbf{D} = \{d_{ij}\}$ has zero row sums, as we demonstrate in Appendix B.3. Hence $d_{ii} = - \sum_{j \neq i} d_{ij}$ and:

$$\sum_j d_{ij} v_j = d_{ii} v_i + \sum_{j \neq i} d_{ij} v_j = - \sum_{j \neq i} d_{ij} v_i + \sum_{j \neq i} d_{ij} v_j = \sum_{j \neq i} d_{ij} (v_j - v_i), \quad \forall i, \quad (2.106)$$

which allows us to write (2.104) as:

$$m_i \frac{\partial v_i}{\partial t} = - \sum_{j \neq i} d_{ij} (v_j - v_i) + f_i, \quad \forall i. \quad (2.107)$$

For the scalar problem, the positivity criterion states that, if $v_i(t=0) \geq 0$, $\forall i$, then $v_i(t) \geq 0$, $\forall i$, $\forall t$. Therefore, the semi-discrete system (2.107) preserves positivity if:

$$m_i > 0, \quad d_{ij} \leq 0, \quad f_i \geq 0, \quad \forall i, \forall j \neq i. \quad (2.108)$$

For instance, consider that $v_i = 0$ and the previous conditions hold. In this case, it is clear that $\partial v_i / \partial t \geq 0$. As another example, consider that the source term is null and v_i is a local maximum. Then $- \sum_{j \neq i} d_{ij} (v_j - v_i) \leq 0$ and, as a result, $\partial v_i / \partial t \leq 0$. That is, a local maximum cannot increase (unless due to external

forces). Similarly, a local minimum cannot decrease. This shows that the scheme for the scalar equation is Local Extremum Diminishing (LED). This criterion coined by JAMESON [81] assures positivity because, as v_i is non-negative, so is the global minimum, which cannot decrease further.

After showing that the semi-discrete equation (2.107) is positivity preserving and LED under the assumptions (2.108), we still need to examine the employed time discretization procedure. In the present study, we employ the generalized trapezoidal method, arriving at a system in the form:

$$\mathbf{A}\mathbf{V}^{n+1} = \mathbf{B}\mathbf{V}^n + \mathbf{F}^{n+\theta}, \quad (2.109)$$

where $\mathbf{A} = \{a_{ij}\}$ and $\mathbf{B} = \{b_{ij}\}$, with:

$$a_{ij} = m_i \delta_{ij} + \theta \Delta t d_{ij}^{n+1}, \quad \forall i, \forall j, \quad (2.110)$$

$$b_{ij} = m_i \delta_{ij} - (1 - \theta) \Delta t d_{ij}^n, \quad \forall i, \forall j, \quad (2.111)$$

where δ_{ij} is the Kronecker delta. Here, the monotonicity of \mathbf{A} is an important property, since it implies that, if $(\mathbf{A}\mathbf{V}^{n+1})_i \geq 0, \forall i$, then $v_i \geq 0, \forall i$. Thus, we restrict \mathbf{A} to be an M-matrix, which is a class of monotone matrices that satisfies [82]:

$$a_{ii} > 0, \quad \forall i, \quad (2.112)$$

$$a_{ij} \leq 0, \quad \forall i, \forall j \neq i, \quad (2.113)$$

$$\sum_j a_{ij} \geq 0, \quad \forall i. \quad (2.114)$$

Therefore, assuming the conditions (2.108) are met, the fully discrete scheme is positivity preserving if \mathbf{A} meets the requirements (2.112)-(2.114) and \mathbf{B} does not have negative values ($b_{ij} \geq 0, \forall i, \forall j$). Hence we examine the matrices \mathbf{A} and \mathbf{B} to see if these properties hold. First, we observe that their off-diagonal entries have the correct sign:

$$a_{ij} = \theta \Delta t d_{ij}^{n+1} \leq 0, \quad \forall i, \forall j \neq i, \quad (2.115)$$

$$b_{ij} = -(1 - \theta) \Delta t d_{ij}^n \geq 0, \quad \forall i, \forall j \neq i. \quad (2.116)$$

Plus, the diagonal values of \mathbf{A} also have the appropriate sign:

$$a_{ii} = m_i + \theta \Delta t d_{ii}^{n+1} = m_i - \theta \Delta t \sum_{j \neq i} d_{ij}^{n+1} > 0, \quad \forall i. \quad (2.117)$$

However, for \mathbf{B} to only have non-negative diagonal values, we have to impose a

restriction on the time step:

$$\begin{aligned} b_{ii} &= m_i - (1 - \theta)\Delta t d_{ii}^n \geq 0 \\ \therefore \Delta t &\leq \frac{m_i}{(1 - \theta)d_{ii}^n}, \quad \forall i. \end{aligned} \quad (2.118)$$

Now, we return to the multidimensional Equation (2.74) and perform a similar analysis. To start, we remark that the system matrices are organized by blocks, in the sense that:

$$\mathbf{M} = \{\mathbf{M}_{ij}\} = \{m_{ij}\mathbf{I}_3\}, \quad \text{with} \quad m_{ij} = \int_{\Omega} N_i N_j d\Omega, \quad (2.119)$$

$$\mathbf{D} = \{\mathbf{D}_{ij}\}, \quad \mathbf{F} = \{\mathbf{F}_i\}, \quad (2.120)$$

where \mathbf{M}_{ij} and \mathbf{D}_{ij} are 3×3 matrices that are not null only if the nodes i and j are adjacent. Thus, \mathbf{M} and \mathbf{D} are block sparse matrices. Also, \mathbf{F}_i is a three-dimensional vector with the source term contributions associated to each component of \mathbf{V}_i . In addition, for this system, the lumped mass matrix is defined by block-wise row sums:

$$\mathbf{M}_L = \text{diag}\{\mathbf{M}_{Li}\} = \text{diag}\{m_i\mathbf{I}_3\}, \quad \mathbf{M}_{Li} = \sum_j \mathbf{M}_{ij}, \quad \forall i. \quad (2.121)$$

Therefore, we can replace the consistent mass matrix in (2.74) with the block diagonal lumped mass matrix, and rewrite the system as:

$$\mathbf{M}_{Li} \frac{\partial \mathbf{V}_i}{\partial t} = - \sum_j \mathbf{D}_{ij} \mathbf{V}_j + \mathbf{F}_i, \quad \forall i. \quad (2.122)$$

Next, observe that \mathbf{D} has zero block-wise row sums, as we demonstrate in Appendix B.3. Then, $\mathbf{D}_{ii} = - \sum_{j \neq i} \mathbf{D}_{ij}$, $\forall i$, and:

$$\begin{aligned} \sum_j \mathbf{D}_{ij} \mathbf{V}_j &= \mathbf{D}_{ii} \mathbf{V}_i + \sum_{j \neq i} \mathbf{D}_{ij} \mathbf{V}_j = - \sum_{j \neq i} \mathbf{D}_{ij} \mathbf{V}_i + \sum_{j \neq i} \mathbf{D}_{ij} \mathbf{V}_j = \\ &= \sum_{j \neq i} \mathbf{D}_{ij} (\mathbf{V}_j - \mathbf{V}_i), \quad \forall i, \end{aligned} \quad (2.123)$$

which we use to write Equation (2.122) as:

$$\mathbf{M}_{Li} \frac{\partial \mathbf{V}_i}{\partial t} = - \sum_{j \neq i} \mathbf{D}_{ij} (\mathbf{V}_j - \mathbf{V}_i) + \mathbf{F}_i, \quad \forall i. \quad (2.124)$$

In sequence, we initially consider the same positivity criterion employed in the scalar problem. It states that, if $\mathbf{V}_i^0 \geq 0$, $\forall i$, then $\mathbf{V}_i \geq 0$, $\forall i$, $\forall t > 0$. Here, all matrix/vector inequalities should be treated as element-wise comparisons. At this

point, it is important to emphasize that, if $\mathbf{V}_j \geq 0$ is not held for a node $j \neq i$, then the numerical solution for \mathbf{V}_i is not forced to be positive. In fact, positivity preservation means that the numerical scheme cannot produce nonphysical negative values [48]. This notion is particularly important in multidimensional problems, such as the present one, where the fluid discharges can obviously become negative over time.

Following KUZMIN *et al.* [83], we employ a natural generalization of the LED criterion used in the scalar transport problem, requiring all off-diagonal blocks of \mathbf{D} to be negative semi-definite. Alternatively, we suppose $-\mathbf{D}_{ij}, \forall i, \forall j \neq i$, is positive semi-definite. This requirement is much less restrictive than assuming that all off-diagonal entries of \mathbf{D} are non-positive [84]. To better understand why $-\mathbf{D}_{ij}, \forall i, \forall j \neq i$, should be positive semi-definite, consider a positive definite matrix \mathbf{P} and a non-zero vector \mathbf{v} . Then, the product $\mathbf{P}\mathbf{v}$ is the result of a linear transformation on \mathbf{v} , being a vector of different norm and direction, as depicted in Figure 2.2. From the definition of a positive definite matrix, we know that $\mathbf{v}^T \mathbf{P}\mathbf{v} > 0$. This

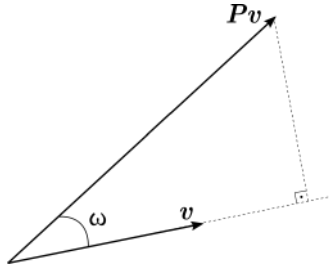


Figure 2.2: Representation of a non-zero vector \mathbf{v} and its transformation $\mathbf{P}\mathbf{v}$ using the positive definite matrix \mathbf{P} .

operation also represents the scalar product between \mathbf{v} and $\mathbf{P}\mathbf{v}$, so that:

$$\mathbf{v}^T \mathbf{P}\mathbf{v} = \|\mathbf{v}\| \|\mathbf{P}\mathbf{v}\| \cos(\omega) > 0, \quad (2.125)$$

where ω is the angle between \mathbf{v} and $\mathbf{P}\mathbf{v}$. As a result, $\omega \in [-\pi/2, \pi/2]$, which means that $\mathbf{P}\mathbf{v}$ points to the same general direction as \mathbf{v} or, equivalently, \mathbf{P} cannot reverse the overall direction of \mathbf{v} . Thereby, as $m_i > 0, \forall i$ and we presuppose $\mathbf{F}_i \geq 0, \forall i$, we can show that the scheme is LED if $\mathbf{D}_{ij}, \forall i, \forall j \neq i$, is negative semi-definite. For example, consider that \mathbf{V}_i is a local minimum. Then, $-\sum_{j \neq i} \mathbf{D}_{ij} (\mathbf{V}_j - \mathbf{V}_i)$ points to the general direction of growth of the magnitude of \mathbf{V}_i . Thus a minimum cannot decrease and, similarly, a maximum cannot increase.

Afterward, we discretize Equation (2.122) in time using the generalized trape-

zoidal method, obtaining:

$$\sum_j \left(\mathbf{M}_{Li} \delta_{ij} + \theta \Delta t \mathbf{D}_{ij}^{n+1} \right) \mathbf{V}_j^{n+1} = \sum_j \left[\mathbf{M}_{Li} \delta_{ij} - (1 - \theta) \Delta t \mathbf{D}_{ij}^n \right] \mathbf{V}_j^n + \Delta t \mathbf{F}_i^{n+\theta}, \quad \forall i. \quad (2.126)$$

If we disregard the source term, it can also be written as:

$$\mathbf{A} \mathbf{V}^{n+1} = \mathbf{B} \mathbf{V}^n, \quad (2.127)$$

where $\mathbf{A} = \{\mathbf{A}_{ij}\}$ and $\mathbf{B} = \{\mathbf{B}_{ij}\}$. Then, as a natural extension of the scalar problem's approach, we let \mathbf{A} be a generalized M-matrix [85, 86] where $\mathbf{A}_{ii}, \forall i$, is positive definite, $\mathbf{A}_{ij}, \forall i, \forall j \neq i$, is negative semi-definite and \mathbf{A} is block-dominant, in the sense that $\mathbf{A}_{ii} + 0.5 \sum_j (\mathbf{A}_{ij} + \mathbf{A}_{ji}) \geq 0, \forall i$. Also, we assume that $\mathbf{B}_{ij}, \forall i, \forall j$, is positive semi-definite so that $\mathbf{B}_{ij} \mathbf{V}_j^n$ does not change the overall direction of \mathbf{V}_j^n .

Next, we propose to examine the blocks $\mathbf{A}_{ij} = \{a_{ij}^{kl}\}$ and $\mathbf{B}_{ij} = \{b_{ij}^{kl}\}$ - which are 3×3 matrices - for additional necessary conditions. We basis these requirements on the fact that positive definite matrices must have non-negative diagonal entries and, likewise, negative definite matrices only have non-positive values at their diagonals, as briefly discussed in Appendix B.4. Thus, as we need the diagonal blocks of \mathbf{A} to be positive definite, we must have:

$$\begin{aligned} a_{ii}^{kk} &= m_i + \theta \Delta t (d_{ii}^{kk})^{n+1} > 0 \\ \therefore \Delta t &> \frac{-m_i}{\theta (d_{ii}^{kk})^{n+1}}, \quad \forall i, \forall k, \end{aligned} \quad (2.128)$$

which should be true by construction, since $\mathbf{D}_{ii} = -\sum_{j \neq i} \mathbf{D}_{ij}, \forall i$, and we assume $\mathbf{D}_{ij}, \forall i, \forall j \neq i$, is negative semi-definite. Similarly, for the negative semi-definite off-diagonal blocks:

$$a_{ij}^{kk} = \theta \Delta t (d_{ij}^{kk})^{n+1} \leq 0, \quad \forall i, \forall j \neq i, \forall k, \quad (2.129)$$

which is also met by default. In the case of the positive semi-definite blocks \mathbf{B}_{ij} , for the off-diagonal ones, the condition:

$$b_{ij}^{kk} = -(1 - \theta) \Delta t (d_{ij}^{kk})^n \geq 0, \quad \forall i, \forall j \neq i, \forall k \quad (2.130)$$

is always met. Meanwhile, for those along the main diagonal:

$$\begin{aligned} b_{ii}^{kk} &= m_i - (1 - \theta) \Delta t (d_{ii}^{kk})^n \geq 0, \\ \therefore \Delta t &\leq \frac{m_i}{(1 - \theta) (d_{ii}^{kk})^n}, \quad \forall i, \forall k, \end{aligned} \quad (2.131)$$

which is a constraint we need to impose on the simulation time step. For this reason, as we advance the simulation in time, we adaptively choose time steps that obey the condition:

$$\Delta t \leq \frac{1}{1 - \theta} \min_{i,k} \left\{ \frac{m_i}{(d_{ii}^{kk})^n} \mid (d_{ii}^{kk})^n > 0 \right\}, \quad (2.132)$$

where the operator $\min_{i,k} [f(i, k)]$ returns the lowest value of the function $f(i, k)$, considering all the values of i and k . The imposition of similar upper bounds on the time step also has been contemplated in the works of KUZMIN *et al.* [45] and KUZMIN [48].

2.4.2 Low-Order Method

Up to this stage, we have replaced \mathbf{M} in (2.74) with \mathbf{M}_L , arriving at the semi-discrete system (2.122), on which we employed the generalized trapezoidal method to obtain the fully discrete Equation (2.126). The next step into the construction of our low-order method is to add artificial diffusion to \mathbf{D} , attempting to eliminate the positive eigenvalues of the off-diagonal matrix blocks, which are rendered negative semi-definite. First, following GUERMOND *et al.* [87] and AUDUSSE *et al.* [88], let the hydrostatic reconstruction of the current's height at the i -th node with respect to the j -th one be:

$$h_{ij} = \max [0, h_i + z_{bi} - \max(z_{bi}, z_{bj})], \quad (2.133)$$

where z_{bi} denotes the bed elevation evaluated at the i -th node, and the operator $\max(\cdot)$ returns the highest of its operands. Then, the associated reconstructed nodal specific discharge vector is $\mathbf{q}_{ij} = h_{ij} \mathbf{u}_i$, where the velocity \mathbf{u}_i is computed from \mathbf{V}_i .

Now, based on the Rusanov-like scalar dissipation operator applied by KUZMIN *et al.* [89], we propose a new one, dependent on the reconstructed height and specific discharge. The latter is included to reduce the artificial diffusion near dry/wet fronts, avoiding the fluid of nonphysically stepping into a dry region, as detailed in Section 2.6. We define $\mathbf{L} = \mathbf{D} + \mathbf{C}$, where the blocks of the novel dissipation matrix $\mathbf{C} = \{\mathbf{C}_{ij}\}$, are given by:

$$\begin{aligned} \mathbf{C}_{ij} &= -\max(c_{ij}, c_{ji}) \mathbf{I}_3, \quad \forall i, \forall j \neq i, \\ \mathbf{C}_{ii} &= -\sum_{j \neq i} \mathbf{C}_{ij}, \quad \forall i, \end{aligned} \quad (2.134)$$

with:

$$c_{ij} = |\mathbf{e}_{ij} \cdot \mathbf{u}_j| + |\mathbf{e}_{ij} \cdot \mathbf{q}_{ji}| \sqrt{gh_{ji}}, \quad \mathbf{e}_{ij} = \int_{\Omega} N_i \nabla N_j d\Omega, \quad \forall i, \forall j \neq i. \quad (2.135)$$

Formed this way, the matrix \mathbf{C} has zero block-wise row and column sums, and thus, being a generalized diffusion operator, conserves mass. However, as stated by KUZMIN [48] in regard to the Euler equations, the nonlinearity of the problem makes it impossible to prove that the artificial diffusion makes the semi-discrete Equation (2.122) LED or positivity preserving. Notwithstanding, most times it is dissipative enough to avoid spurious oscillations near shocks and discontinuities.

In complement, inspired by the works of BASTING and KUZMIN [90] and GUERMOND *et al.* [87], we employ a shock detector, so more diffusion is added near discontinuities, while regions where the solution is smooth receive less. Consider $\Delta h_{ij} = h_i - h_j$ and the function $\text{sign}(x)$ that returns 1 if $x \geq 0$ and -1 , otherwise. Then, we propose to use the operator:

$$\psi_i = \begin{cases} \frac{|\sum_{j \in S(i)} \text{sign}(\Delta h_{ij})(\Delta h_{ij} - r)^4|}{\sum_{j \in S(i)} (\Delta h_{ij} - r)^4}, & \text{if } \sum_{j \in S(i)} (\Delta h_{ij} - r)^4 \neq 0, \\ 0, & \text{otherwise,} \end{cases} \quad (2.136)$$

associated with a node i , where $r = 0.001$ and $S(i)$ is the set of the neighboring nodes of i , which includes all those belonging to the elements adjacent to i . Thereby, we scale the submatrices of \mathbf{C} :

$$\begin{aligned} \mathbf{C}_{ij} &= -\max(\psi_i, \psi_j) \max(c_{ij}, c_{ji}) \mathbf{I}_3, \quad \forall i, \forall j \neq i, \\ \mathbf{C}_{ii} &= -\sum_{j \neq i} \mathbf{C}_{ij}, \quad \forall i. \end{aligned} \quad (2.137)$$

Altogether, our results indicate that this operator produces better solutions for the tested cases than the earlier ones.

Then, we can write the equation for the low-order scheme as:

$$\sum_j \left(\mathbf{M}_{Li} \delta_{ij} + \theta \Delta t \mathbf{L}_{ij}^{(m)} \right) \mathbf{V}_j^{(m+1)} = \sum_j \left[\mathbf{M}_{Li} \delta_{ij} - (1 - \theta) \Delta t \mathbf{L}_{ij}^n \right] \mathbf{V}_j^n + \Delta t \mathbf{F}_i^{n+\theta}, \quad \forall i, \quad (2.138)$$

which we solve iteratively using the same stopping criteria and linear system solver used with the stabilized approach. After convergence, we assign $\mathbf{V}^L = \mathbf{V}^{(m+1)}$. Do observe that, in this case, the time step constraint (2.132) must be computed based on the entries of \mathbf{L} , instead of those from \mathbf{D} .

2.4.3 Source Term Linearization

To prevent a negative source term from violating the positivity constraint, while also following the general recommendation to treat negative sources implicitly and positive sources explicitly [91], we linearize source terms around \mathbf{V}^{n+1} , as suggested by PATANKAR [92] and KUZMIN *et al.* [45]:

$$\mathbf{F}_i = \mathbf{F}_{C_i} + \mathbf{F}_{P_i} \mathbf{V}_i^{n+1}, \quad \forall i, \quad (2.139)$$

with $\mathbf{F}_{C_i} > 0$ and $\mathbf{F}_{P_i} \leq 0$, $\forall i$. This splitting can be done in the context of the iterative scheme with:

$$\mathbf{F}_i = \mathbf{F}_i^+ + \left(\frac{\mathbf{F}_i^-}{\mathbf{V}_i^{(m)}} \right) \mathbf{V}_i^{(m+1)}, \quad (2.140)$$

where \mathbf{F}_i^+ and \mathbf{F}_i^- are, respectively, the positive and negative parts of the source term. Thus we have $\mathbf{F}_{C_i} = \mathbf{F}_i^+$ and $\mathbf{F}_{P_i} = \mathbf{F}_i^- / \mathbf{V}_i^{(m)}$. In this work, this separation is performed individually for each quadrature point contribution. Besides, considering that $\mathbf{F}_i^{n+\theta} = \theta \mathbf{F}_i^{n+1} + (1-\theta) \mathbf{F}_i^n$, we split the contribution of \mathbf{F}_i^{n+1} into $\mathbf{F}_{C_i}^{(m)}$ and $\mathbf{F}_{P_i}^{(m)}$, and divide \mathbf{F}_i^n into $\mathbf{F}_{C_i}^n$ and $\mathbf{F}_{P_i}^{n(m)}$.

Therefore, the final low-order equation is:

$$\begin{aligned} \sum_j \left[\mathbf{M}_{L_i} \delta_{ij} + \theta \Delta t \left(\mathbf{L}_{ij}^{(m)} - \mathbf{F}_{P_i}^{(m)} \delta_{ij} \right) - (1-\theta) \Delta t \mathbf{F}_{P_i}^{n(m)} \delta_{ij} \right] \mathbf{V}_j^{(m+1)} = \\ = \sum_j \left[\mathbf{M}_{L_i} \delta_{ij} - (1-\theta) \Delta t \mathbf{L}_{ij}^n \right] \mathbf{V}_j^n + \theta \Delta t \mathbf{F}_{C_i}^{(m)} + (1-\theta) \Delta t \mathbf{F}_{C_i}^n, \quad \forall i. \end{aligned} \quad (2.141)$$

Do note that, because we have $-\mathbf{F}_{P_i}^{(m)} \geq 0$ and $-\mathbf{F}_{P_i}^{n(m)} \geq 0$, $\forall i$, this type of linearization reinforces the dominance of the block matrices on the diagonal and contributes to having non-negative diagonal entries on the left-hand side matrix. Plus, the fact that $\mathbf{F}_{C_i}^{(m)} \geq 0$ and $\mathbf{F}_{C_i}^n \geq 0$, $\forall i$, on the right-hand side also avoids the source term of spuriously changing the sign of the solution.

2.4.4 High-Order Method

To obtain the high-order equation, we directly employ the generalized trapezoidal method for the time discretization of Equation (2.74), arriving at:

$$\left(\mathbf{M}_{ij} + \theta \Delta t \mathbf{D}_{ij}^{n+1} \right) \mathbf{V}_j^{n+1} = \left[\mathbf{M}_{ij} - (1-\theta) \Delta t \mathbf{D}_i^n \right] \mathbf{V}_j^n + \Delta t \mathbf{F}_i^{n+\theta}, \quad \forall i. \quad (2.142)$$

Then, we build the high-order method by adding anti-diffusive fluxes \mathcal{F}_i to the right-hand side of the low-order system (2.141), recovering the previous high-order

set of equations:

$$\begin{aligned}
& \sum_j \left[\mathbf{M}_{Li} \delta_{ij} + \theta \Delta t \left(\mathbf{L}_{ij}^{(m)} - \mathbf{F}_{P_i}^{(m)} \delta_{ij} \right) - (1 - \theta) \Delta t \mathbf{F}_{P_i}^{n(m)} \delta_{ij} \right] \mathbf{V}_j^{(m+1)} = \\
& = \sum_j \left[\mathbf{M}_{Li} \delta_{ij} - (1 - \theta) \Delta t \mathbf{L}_{ij}^n \right] \mathbf{V}_j^n + \theta \Delta t \mathbf{F}_{C_i}^{(m)} + (1 - \theta) \Delta t \mathbf{F}_{C_i}^n + \mathcal{F}_i, \quad \forall i.
\end{aligned} \tag{2.143}$$

In this case, the definition of the anti-diffusive term comes from the difference between the high-order Equation (2.142) and the low-order Equation (2.138), which results in:

$$\begin{aligned}
\mathcal{F}_i(\mathbf{V}^{(m)}, \mathbf{V}^n) = & \sum_j \left(\mathbf{M}_{Li} \delta_{ij} - \mathbf{M}_{ij} \right) \left(\mathbf{V}_j^{(m)} - \mathbf{V}_j^n \right) + \\
& + \sum_j \left[\theta \Delta t \mathbf{C}_{ij}^{(m)} \mathbf{V}_j^{(m)} + (1 - \theta) \Delta t \mathbf{C}_{ij}^n \mathbf{V}_j^n \right], \quad \forall i.
\end{aligned} \tag{2.144}$$

Owing to the fact that we use the diagonally lumped mass matrix and also because \mathbf{C} has zero block-wise row sums, the following relations hold:

$$\begin{aligned}
\sum_j \left(\mathbf{M}_{Li} \delta_{ij} - \mathbf{M}_{ij} \right) \mathbf{V}_j & = \mathbf{M}_{Li} \mathbf{V}_i - \sum_j \mathbf{M}_{ij} \mathbf{V}_j = \\
& = \left(\sum_j \mathbf{M}_{ij} \right) \mathbf{V}_i - \sum_j \mathbf{M}_{ij} \mathbf{V}_j = \sum_{j \neq i} \mathbf{M}_{ij} \left(\mathbf{V}_i - \mathbf{V}_j \right), \quad \forall i,
\end{aligned} \tag{2.145}$$

$$\begin{aligned}
\sum_j \mathbf{C}_{ij} \mathbf{V}_j & = \mathbf{C}_{ii} \mathbf{V}_i + \sum_{j \neq i} \mathbf{C}_{ij} \mathbf{V}_j = \left(- \sum_{j \neq i} \mathbf{C}_{ij} \right) \mathbf{V}_i + \sum_{j \neq i} \mathbf{C}_{ij} \mathbf{V}_j = \\
& = \sum_{j \neq i} \mathbf{C}_{ij} \left(\mathbf{V}_j - \mathbf{V}_i \right), \quad \forall i,
\end{aligned} \tag{2.146}$$

allowing us to write the anti-diffusive flux in terms of the adjacent nodes' contributions:

$$\mathcal{F}_i = \sum_{j \in S(i)} \mathcal{F}_{ij}, \quad \forall i, \tag{2.147}$$

with:

$$\mathcal{F}_{ij} = \left(\mathbf{M}_{ij} - \theta \Delta t \mathbf{C}_{ij}^{(m)} \right) \left(\mathbf{V}_i^{(m)} - \mathbf{V}_j^{(m)} \right) - \left[\mathbf{M}_{ij} + (1 - \theta) \Delta t \mathbf{C}_{ij}^n \right] \left(\mathbf{V}_i^n - \mathbf{V}_j^n \right). \tag{2.148}$$

2.4.5 Flux Limiting

Now we adaptively switch between the low- and high- order methods by limiting how much anti-diffusion is added. This is done by scaling the edge contributions:

$$\mathcal{F}_i = \sum_{j \in S(i)} \alpha_{ij} \mathcal{F}_{ij}, \quad \forall i. \quad (2.149)$$

Following KUZMIN [48], the use of the multidimensional Zalesak's flux limiter [50] yields an individual correction factor for each solution component (α_{ij}^h , α_{ij}^{qx} and α_{ij}^{qy}). In this case, a synchronization technique is needed to obtain a single α_{ij} and avoid inappropriate results, such as the lack of mass conservation. Indeed, we found the best results are obtained when $\alpha_{ij} = \alpha_{ij}^h$. Thus, let \mathcal{F}_{ij}^h be the anti-diffusive flux component related to h . We propose to employ a limiter for \mathcal{F}_{ij}^h that also depends on the hydrostatic reconstruction h_{ji} :

$$P_i^+ = \sum_{j \in S(i)} \max(0, \mathcal{F}_{ij}^h), \quad P_i^- = \sum_{j \in S(i)} \min(0, \mathcal{F}_{ij}^h), \quad (2.150)$$

$$Q_i^+ = \max \left[0, \max_{j \in S(i)} \left(h_{ji}^{(m)} - h_i^{(m)} \right) \right], \quad Q_i^- = \min \left[0, \min_{j \in S(i)} \left(h_{ji}^{(m)} - h_i^{(m)} \right) \right], \quad (2.151)$$

$$R_i^+ = \min \left(1, \frac{m_i Q_i^+}{P_i^+} \right), \quad R_i^- = \min \left(1, \frac{m_i Q_i^-}{P_i^-} \right), \quad (2.152)$$

$$\alpha_{ij} = \alpha_{ij}^h = \begin{cases} \min(R_i^+, R_j^-), & \text{if } \mathcal{F}_{ij}^h > 0, \\ \min(R_i^-, R_j^+), & \text{otherwise.} \end{cases} \quad (2.153)$$

Here it is important to observe that $\alpha_{ij} = \alpha_{ji}$, so that the limiter preserves the flux balance between the nodes i and j .

Additionally, we prelimit the fluxes before computing the correction factors. Hence, as suggested by KUZMIN [48], we adopted a minmod strategy to prevent the consistent mass matrix of reversing the sign of \mathcal{F}_{ij} or increasing its magnitude:

$$\mathcal{F}_{ij} = \text{minmod} \left(\mathcal{F}_{ij}, \mathcal{F}_{ij}^* \right), \quad (2.154)$$

with:

$$\mathcal{F}_{ij}^* = -\theta \Delta t \sum_j \mathbf{C}_{ij}^{(m)} \left(\mathbf{V}_i^{(m)} - \mathbf{V}_j^{(m)} \right) - (1 - \theta) \Delta t \sum_j \mathbf{C}_{ij}^n \left(\mathbf{V}_i^n - \mathbf{V}_j^n \right). \quad (2.155)$$

By definition, the minmod function returns zero if the arguments have opposite signs, or the argument with the smallest magnitude, otherwise. We observed that, while the correction factors must be synchronized, the minmod prelimiter works

better if applied to each flux component individually.

To avoid having to compute and limit the anti-diffusive fluxes at each iteration of Equation (2.143), we use \mathbf{V}^L as an approximation to $\mathbf{V}^{(m)}$ - and, thus, to \mathbf{V}^{n+1} - when computing the fluxes and correction factors. Hence, both have to be computed only once per time step. Nonetheless, the low-order solution must be obtained prior to solving Equation (2.143). We present a summary of the full FCT scheme in Algorithm 2.

Algorithm 2 Summary of the adopted FCT scheme.

- 1: Choose Δt based on the *CFL* condition (2.157).
 - 2: Assemble the low-order system.
 - 3: **if** condition (2.132) is not met **then** choose new Δt and reassemble the system.
 - 4: Obtain the low-order solution \mathbf{V}^L .
 - 5: Compute the anti-diffusive fluxes $\mathcal{F}(\mathbf{V}^L, \mathbf{V}^n)$.
 - 6: Prelimit the fluxes.
 - 7: Compute the α_{ij} factors and limit the fluxes.
 - 8: Compute the high-order solution \mathbf{V}^H and set $\mathbf{V}^{n+1} = \mathbf{V}^H$.
-

2.5 Adaptive Time Step Control

As we advance the solution in time, we constrain the time step to limit the maximum *CFL* number of an element e , defined as:

$$CFL_e = \left(|\mathbf{u}_e| + \sqrt{g'h_e} \right) \frac{\Delta t}{l_e}, \quad (2.156)$$

where l_e is the element's characteristic length, defined here as the square root of its area. Also, the speeds $|\mathbf{u}_e|$ and $\sqrt{g'h_e}$ are evaluated at the element's barycentre. Then, the related restriction is given by:

$$\Delta t \leq CFL \min_e \left(\frac{l_e}{|\mathbf{u}_e| + \sqrt{g'h_e}} \right). \quad (2.157)$$

For the simulations ran in this work, $CFL = 0.5$. Thus, as we march through time, we adaptively choose time steps that comply with condition (2.157). Do note that the adopted FCT scheme further constrains the time step to help preserve positivity, choosing time steps that also obey the requirement (2.132).

2.6 Dry/Wet Handling

To reinforce the model’s applicability to real-world scenarios where turbidity currents flow over irregular beds, we need to mitigate the instabilities and the nonphysical behavior that might arise near transitions between dry and wet states. Hence, initially, it conveys to distinguish between wet and dry elements. We classify a node as being wet if its height h is greater than the threshold $h_{\text{dry}} = 0.01$ m. Then, an element is classified as wet or dry according to whether it has only wet or dry nodes. Otherwise, it is in a dry/wet front. This type of procedure is commonly applied to shallow water models [93–96].

Another frequent conduct is to define a cut-off height value, under which point velocities are considered null. Often this value is the same as the one used for the dry/wet element classification [96]. Still, no cut-off is considered for the height h . Based on the approach employed by RICCHIUTO and BOLLERMANN [95], we adopt:

$$\mathbf{u} = \begin{cases} \frac{\mathbf{q}}{h}, & \text{if } h \geq C_{\mathbf{u}}, \\ \mathbf{0}, & \text{otherwise,} \end{cases} \quad (2.158)$$

where $C_{\mathbf{u}} = l_e/L_{\text{ref}}$, with $L_{\text{ref}} = \max_{i,j \in \Omega} (\|\mathbf{x}_i - \mathbf{x}_j\|)$ and l_e is the characteristic length of the associated element.

Moreover, spurious velocities can arise near dry/wet fronts and violate mass conservation, as a result of trying to simulate a continuous surface elevation on a discrete mesh. This is particularly important at elements with adverse slopes. An element is said to have an adverse slope if it has a wet node i and a dry node j , where $z_{b_j} > z_{b_i}$ [95]. For fronts over flat or downward sloping surfaces, the discrete equilibrium would correctly induce the flow in the direction of the dry nodes.

To better understand the problem with adverse slopes, consider, for simplicity, the case of a still fluid in 1D depicted in Figure 2.3. We take into account the wet node i and the dry node j , where the fluid’s free surface ($\eta = h + z_b$) at i is lower than the bed elevation at j . The real physical state in equilibrium is illustrated in Figure 2.3a, where we have a flat free surface and a still fluid. However, the discrete linear finite element approximation seen in Figure 2.3b ends up causing the momentum balance to produce spurious velocities downslope.

This undesired behavior can be avoided by the solution proposed by BRUFAU and GARCÍA-NAVARRO [93], where the bed gradient is locally redefined to obey the equilibrium condition $\nabla z_b = -\nabla h$. This requirement can be easily obtained by nullifying the velocities of the momentum conservation equation in (2.53). For an element with an adverse slope, the bed elevation of its dry nodes is updated as:

$$z_{b_{\text{dry}}} = \max_{i \text{ is Wet}} (z_{b_i} + h_i). \quad (2.159)$$

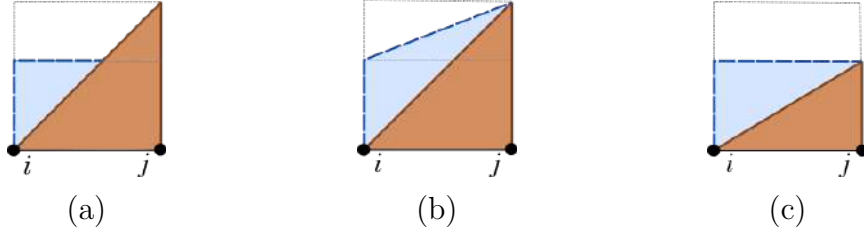


Figure 2.3: Representations of the fluid’s free surface in a 1D element with nodes i and j . (a) Real physical state; (b) Linear finite element representation; (c) Corrected linear finite element representation.

Then, for the given example, this adjustment results in the state shown in Figure 2.3c. In earlier works, this procedure has been applied to triangular elements. The fact that we use quadrilateral elements could jeopardize the correction since the polynomial interpolations in linear triangular and quadrilateral elements are different. The free surface is always approximated by a plane inside triangular elements, which is not true for quadrilaterals. However, we found that the technique also produces excellent results with quadrilateral elements. Additionally, we observed that the update can be performed locally. When assembling an element’s matrix contributions, we use nodal bed elevations corrected based only on the element’s nodes. There is no need to iterate through all the elements adjusting all nodal bed elevations.

Therefore, the adopted correction algorithm keeps the same discrete fluid volume and preserves the mass and the steady state as it avoids the creation of nonphysical velocities. For fronts propagating over adverse slopes, we apply the same procedure. However, after each time step, to avoid some fluid quickly jumping to a dry node, we zero the solution discharges using the same cut-off condition used for the velocities in Equation (2.158). This has been done by other authors, such as JIANG and WAI [97], that zero dry node discharges. Also, with the same objective, we have included the reconstructed specific discharge \mathbf{q}_{ji} in the second term of the artificial diffusion operator (2.135). In this case, the artificial diffusion is nullified for current heights smaller than the velocity cut-off C_u .

A final aspect to be considered concerns how the friction term varies when h tends to zero. As we use the Manning’s formula to compute the drag coefficient, it could become arbitrarily large and bring numerical problems. To avoid this, LIANG and BORTHWICK [98] zeroes the drag coefficient at dry quadrature points. LIANG and MARCHE [99] limit the friction force, so its maximum effect is to stop the fluid, i.e., it cannot reverse the flow. Conversely, other works increase friction when $h \rightarrow 0$ as a means to decrease and stabilize the flow. In fact, GOURGUE *et al.* [57] linearly increase the bottom stress and eddy viscosity inside the so-called “buffer” layer, defined by the region where $h < h_{\text{buffer}}$, being h_{buffer} a constant. WANG *et al.* [100] also

increase the drag coefficient, using $h_{\text{buffer}} = 0.05$ m and $h_{\text{dry}} = 0.002$ m. With the goal to freeze the flow in dry areas, HENICHE *et al.* [101] linearly vary the Manning coefficient at dry nodes: $n_{\text{dry}} = n[1 + \beta(h_{\text{dry}} - h)]$, with $\beta \in [10^2, 10^4]$. We follow HENICHE *et al.* [101], using $\beta = 10^2$.

Chapter 3

Event-Driven Simulation

In the context of turbidity currents, an event-driven simulation comes as a natural approach, since it is known that many geological features are the result of not one but many turbidite events. Here, the event-driven concept refers to the fact that simulation is only run during event discharges into the area of interest and while the currents carrying the inflow sediments have not reached an equilibrium state. Thus, when these criteria are not met, the simulation halts and time is skipped to the start of the next event occurrence, when the simulation is resumed.

This plan of action can find its use in the field of stratigraphy, where the simulations performed to build a complete stratigraphic model might need to span millions of years. Frequently, this process is split into sub-intervals that have their own parameter configuration and usually extend for at least a hundred thousand years. In contrast, a single turbidity current event range from hours to months [102]. So, knowing or estimating when they occur can considerably reduce simulation costs.

Therefore, in this chapter, we introduce our approach to perform an event-driven simulation of the sedimentary fill of an area. We start by showing how we define and track turbidite events, which are the base unit of the simulation. Then, we present how we build and evolve a set of streaklines for each event. In sequence, we show how the mesh used in the flow simulation is constructed from an external stratigraphic grid. Finally, we put forward the event-wise sedimentation algorithm.

3.1 Recurrent Events

To simplify the definition of events over long periods, we have chosen to work with recurrent episodes, namely an event with given features that have several occurrences over time. The event itself is defined by the usual Dirichlet boundary conditions, i.e., by setting $\mathbf{U}^T = \begin{bmatrix} h_e & q_{x_e} & q_{y_e} \end{bmatrix}$ on a section of the contour, and the total amount of volume it can supply to the domain. However, we have found that, instead of directly specifying \mathbf{q}_e , it is more intuitive to prescribe h_e on a sub-boundary and

define the event volume and the interval when its inflow is active. In this case, to compute the specific discharges required for the associated boundary condition, first, we obtain the event's flow rate by dividing its total volume by its duration. Then the specific discharge magnitude is obtained by the ratio between the flow rate and the sum of the lengths of the element edges along the sub-boundary. At last, \mathbf{q}_e is locally determined at each edge by the product between the magnitude and the edge's inward-pointing unit normal. In turn, event recurrence is described by its quantity and the time interval between the start of each episode.

When an event e has introduced all its available volume V_{eT} into the domain, its inflow discharges must be deactivated. Hence, we apply the new boundary condition $\mathbf{q}_e = \mathbf{0}$. However, to know when this must happen, we need to track the inflow volume of each active event. This is done by accumulating:

$$\Delta V_e^n = -\Delta t \int_{\partial_e \Gamma_e} [(1 - \theta) (\mathbf{q}_e^n \cdot \mathbf{n}) + \theta (\mathbf{q}_e^{n+1} \cdot \mathbf{n})] d\Gamma_e, \quad (3.1)$$

where $\partial_e \Gamma_e$ refers to the subset of the boundary Γ_e on which the event is active and \mathbf{n} is the outward-pointing unit normal vector at the boundary. Specifically, $V_{eA}^n = \sum_n \Delta V_e^n$ is the accumulated inflow volume. Note that we consider \mathbf{q}_e at the start and end of a time step. Consequently, the event's discharges can be given as a function of time.

Furthermore, the time step must be constrained, so no event injects into the domain more volume than it originally had available, and the instant that another event is to be activated is not exceeded. For the former case, the time step is limited based on the previous step's inflow volume:

$$\Delta t \leq \frac{V_{eT} - V_{eA}^{n-1}}{\Delta V_e^{n-1}} \Delta t^{n-1}. \quad (3.2)$$

In the latter context:

$$\Delta t \leq t_{eS} - t^n, \quad (3.3)$$

where t_{eS} is the time when an event is scheduled to start and t is the current time.

Another relevant aspect is how to define when we must end the simulation of an event that is no longer discharging material into the domain. In these cases, computation is interrupted when the flow has stopped or reached an equilibrium state. Thus, we test if the averaged l^2 -norm of the thickness derivative of all wet nodes is sufficiently close to zero, i.e., if:

$$\frac{1}{n_W} \sqrt{\sum_{i \text{ is wet}} \left(\frac{h_i^{n+1} - h_i^n}{\Delta t} \right)^2} \leq 10^{-6}, \quad (3.4)$$

with n_W being the number of wet nodes.

3.2 Streakline Computation

One of the inputs to the sedimentation procedure is the streaklines evolved during each event, as they are needed to transport the supplied sediments throughout the studied region. Thus, we link streakline sources to cells with predetermined sediment supplies. From these fixed points, the lines will emerge as the flow simulation advances. To allow a good integration between the streaklines and the flow and sedimentation solvers, we have chosen to implement our own streakline computation solution.

The approach we adopt is inspired by the works of DRITSCHER [103] and MILLS [104] in the sense that points are added or removed from streaklines based on distance and curvature metrics, and are advected with a Runge-Kutta scheme using velocities interpolated from a mesh. Also, the continuous curve between two nodes is approximated by a locally-determined cubic polynomial. Additionally, after moving all streakline nodes, here we remove the ones that have left the domain.

Streaklines are initially built by inserting new points/particles at their beginnings until each one has, at least, 10 points. Then, more points $\mathbf{p}(x, y)$ are adaptively added or removed to prevent an excessive number of particles, while ensuring accuracy. For this purpose, each streakline is traversed from start to end, considering each segment $\overline{\mathbf{p}_1\mathbf{p}_2}$. As depicted in Figure 3.1, the previous and following points \mathbf{p}_0 and \mathbf{p}_3 are also considered. If \mathbf{p}_1 is the first point of a streakline, \mathbf{p}_0 is extrapolated from the $\overline{\mathbf{p}_1\mathbf{p}_2}$ segment by defining $\mathbf{p}_0 = \mathbf{p}_1 - (\mathbf{p}_2 - \mathbf{p}_1)$. The same is done for \mathbf{p}_3 if \mathbf{p}_2 is the last point.

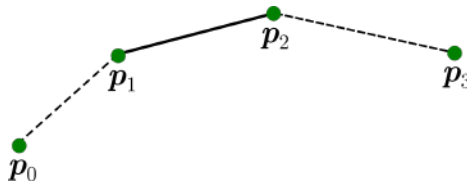


Figure 3.1: Streakline segment $\overline{\mathbf{p}_1\mathbf{p}_2}$ and adjacent points \mathbf{p}_0 and \mathbf{p}_3 .

As we proceed along a line, we remove a segment if its length is less than $0.02 l_1^d$, with l_1^d being the length of the largest diagonal of the element that contains \mathbf{p}_1 . In this case, \mathbf{p}_1 is removed and $\mathbf{p}_2 := 0.5(\mathbf{p}_1 + \mathbf{p}_2)$. Otherwise, the segment can be subdivided according to its mean curvature.

The curvature at a point is computed as the inverse of the radius of the circle circumscribing the triangle defined by the point and its immediate adjacent points. This and other curvature approximations for discrete curves were examined

by WILLIAMS and SHAH [105]. In this context, consider the point \mathbf{p}_2 illustrated in Figure 3.2. To compute the associated curvature, regard the triangle formed by

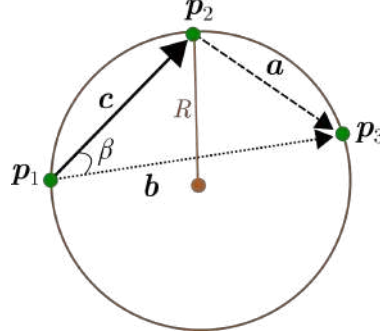


Figure 3.2: The circle circumscribed in the triangle with vertices \mathbf{p}_1 , \mathbf{p}_2 and \mathbf{p}_3 that is used to compute the curvature at \mathbf{p}_2 .

the points \mathbf{p}_1 , \mathbf{p}_2 and \mathbf{p}_3 . Also, let \mathbf{a} , \mathbf{b} and \mathbf{c} be vectors respectively pointing from \mathbf{p}_2 to \mathbf{p}_3 , \mathbf{p}_1 to \mathbf{p}_3 and \mathbf{p}_1 to \mathbf{p}_2 . Plus, R is the circumcircle's radius and β is the angle between \mathbf{b} and \mathbf{c} . Then, from the law of sines:

$$\frac{\|\mathbf{a}\|}{\sin(\beta)} = 2R, \quad (3.5)$$

and the cross-product relation:

$$\|\mathbf{b} \times \mathbf{c}\| = \|\mathbf{b}\| \|\mathbf{c}\| \sin(\beta) \quad (3.6)$$

$$\therefore \sin(\beta) = \frac{\|\mathbf{b} \times \mathbf{c}\|}{\|\mathbf{b}\| \|\mathbf{c}\|}, \quad (3.7)$$

we can determine the curvature at \mathbf{p}_2 :

$$k(\mathbf{p}_2) = \frac{1}{R} = \frac{2\|\mathbf{b} \times \mathbf{c}\|}{\|\mathbf{a}\| \|\mathbf{b}\| \|\mathbf{c}\|}. \quad (3.8)$$

Consequently, the mean curvature of the segment $\overline{\mathbf{p}_1\mathbf{p}_2}$ can be computed as:

$$k_{\text{mean}} = \frac{k(\mathbf{p}_1) + k(\mathbf{p}_2)}{2}, \quad (3.9)$$

where $k(\mathbf{p}_1)$ is determined by applying (3.8) to the previous streakline segment.

Next, we define the circle sector angle relative to the chord \mathbf{c} . Consider the inscribed and central angles ω and ζ depicted in Figure 3.3. From the sine law:

$$\frac{\|\mathbf{c}\|}{\sin(\omega)} = 2R \quad (3.10)$$

$$\therefore \omega = \arcsin\left(\frac{\|\mathbf{c}\|}{2R}\right),$$

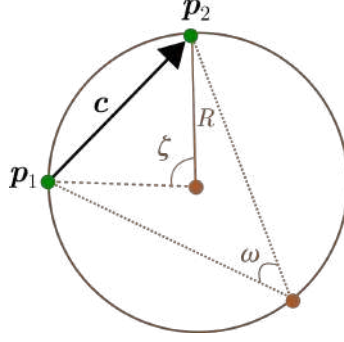


Figure 3.3: Streakline segment, the associated central angle and a inscribed angle.

and the inscribed angle property that states that $\zeta = 2\omega$, we have:

$$\zeta = 2 \arcsin \left(\frac{\|\mathbf{c}\|}{2R} \right), \quad (3.11)$$

which becomes, using a small-angle approximation:

$$\zeta = 2 \frac{\|\mathbf{c}\|}{2R} = \|\mathbf{c}\| \frac{1}{R} = \|\mathbf{c}\| k. \quad (3.12)$$

Here we adopt $\zeta = \|\mathbf{c}\| k_{\text{mean}}$ as the circle sector angle associated with the streakline segment $\overline{\mathbf{p}_1 \mathbf{p}_2}$.

Therefore, we define the number of points that should be added between \mathbf{p}_1 and \mathbf{p}_2 as:

$$n_p^k = \text{ceil} \left(\frac{\zeta}{\zeta_{\text{max}}} \right) - 1, \quad (3.13)$$

where $\zeta_{\text{max}} = 1^\circ = \pi/180 \text{ rad}$, and the function $\text{ceil}(x)$ returns the smallest integer value that is bigger than or equal to x . The positions of the new points are determined by fitting a parametric centripetal Catmull-Rom spline [106] through \mathbf{p}_0 , \mathbf{p}_1 , \mathbf{p}_2 and \mathbf{p}_3 , and evaluating points at equally spaced intervals of the parametric variable. This type of spline guarantees that the tangents of the generated curve are continuous across segments and that it does not form cusps or self-intersections within curve segments [107].

Afterward, if the segment is not subdivided due to curvature restrictions, it can be partitioned if its length is more than $L_{\text{max}} = 3l_1^d$. In this case, the number of points to be added is:

$$n_p^l = \text{ceil} \left(\frac{\|\mathbf{p}_2 - \mathbf{p}_1\|}{L_{\text{max}}} \right) - 1. \quad (3.14)$$

Here the positions of the new points are defined by a linear interpolation between \mathbf{p}_1 and \mathbf{p}_2 . For completeness, we present in Appendix C the equations of the parametric curves used in the interpolations.

After the simulation of each time step, the particles that compose each streakline

are displaced by solving the equation of motion:

$$\frac{\partial^2 \mathbf{X}}{\partial t^2}(\mathbf{x}_0, t) = \mathbf{A}_c(\mathbf{x}_0, t), \quad (3.15)$$

where \mathbf{x}_0 is the initial position of a particle, $\mathbf{X}(\mathbf{x}_0, t)$ is the Lagrangian description of the particle's position over time and $\mathbf{A}_c(\mathbf{x}_0, t)$ is its acceleration. As our flow solver is Eulerian-based, we employ the mapping:

$$\frac{\partial \mathbf{X}}{\partial t}(\mathbf{x}_0, t) = \mathbf{u}[\mathbf{X}(\mathbf{x}_0, t), t], \quad (3.16)$$

and apply the following Runge-Kutta scheme:

$$\mathbf{X}(\mathbf{x}_0, t^{n+1}) = \mathbf{x}_0 + \mathbf{u}(\mathbf{x}_0, t^n)\Delta t + \left(\frac{\sum_{k=0}^3 c_k \mathbf{a}_k}{\sum_{k=0}^3 c_k} \right) \frac{\Delta t^2}{2}, \quad (3.17)$$

where:

$$\mathbf{x}_k = \mathbf{X}(\mathbf{x}_0, t^n) + d_k \left[\mathbf{u}(\mathbf{x}_0, t^n) \frac{\Delta t}{c_k} + \mathbf{a}_{k-1} \frac{(\Delta t/c_k)^2}{2} \right], \quad (3.18)$$

$$\mathbf{a}_k = \mathbf{A}_c(\mathbf{x}_0, t^n + d_k \Delta t/c_k) = \frac{\mathbf{u}(\mathbf{x}_k, t^{n+1}) - \mathbf{u}(\mathbf{x}_0, t)}{\Delta t/c_k}, \quad (3.19)$$

with constants $d_0 = 0$, $d_{k \neq 0} = 1$, $c_0 = c_3 = 1$ and $c_1 = c_2 = 2$. At last, after displacing all the streakline points, we remove the ones that have left the mesh. We provide a summary of the streakline computation procedure in Algorithm 3.

3.3 Mesh Generation

The mesh on which we carry out the flow simulation comes from an external application specialized in stratigraphic simulations based on the works of CARVALHO *et al.* [7–9]. Roughly, its data-structure is a 2D regular grid representing a surface in 3D over which sediment layers are deposited, building pillar-like arrangements on each cell. This type of grid is also used in other stratigraphic simulators [10, 11]. Besides, at any time, the collection of the top of these structures compose the current depositional surface, although only wet regions can undergo morphological changes. An example of an initial surface with the starting water level can be seen in Figure 3.4.

In this context, our objective is to simulate the flow of a turbidity current inside an underwater region. Therefore, we apply suitable boundary conditions to different parts of the wet domain's contour. In fact, we enforce the proper Dirichlet boundary conditions where an event is active, impose non-penetration constraints on boundaries along a coastline, and leave the remaining sub-contours open, only applying

Algorithm 3 Summary of the algorithm used to build streaklines and advance them in time from t^n to t^{n+1} .

```

1: for each streakline do
2:   if it contains less than 10 points then
3:     Add new point at the same position of the streakline's source.
4:   else
5:     for each segment  $\overline{\mathbf{p}_1\mathbf{p}_2}$  do
6:       if  $\|\overline{\mathbf{p}_1\mathbf{p}_2}\| < 0.02 l_1^d$  then ▷ Is it too short?
7:         Remove  $\mathbf{p}_1$ ;
8:          $\mathbf{p}_2 := 0.5(\mathbf{p}_1 + \mathbf{p}_2)$ .
9:       else if  $n_p^k > 0$  then ▷ Is its curvature too high?
10:        Create  $n_p^k$  streakline points between  $\mathbf{p}_1$  and  $\mathbf{p}_2$  at positions
        interpolated using a fitted Catmull-Rom spline.
11:      else if  $n_p^l > 0$  then ▷ Is it too long?
12:        Create  $n_p^l$  streakline points between  $\mathbf{p}_1$  and  $\mathbf{p}_2$  at linearly
        interpolated positions.
13:      end if
14:    end for
15:  end if
16:  Displace all the streakline's points besides its source using Equation (3.17).
17:  Remove all streakline points that have left the mesh.
18: end for

```

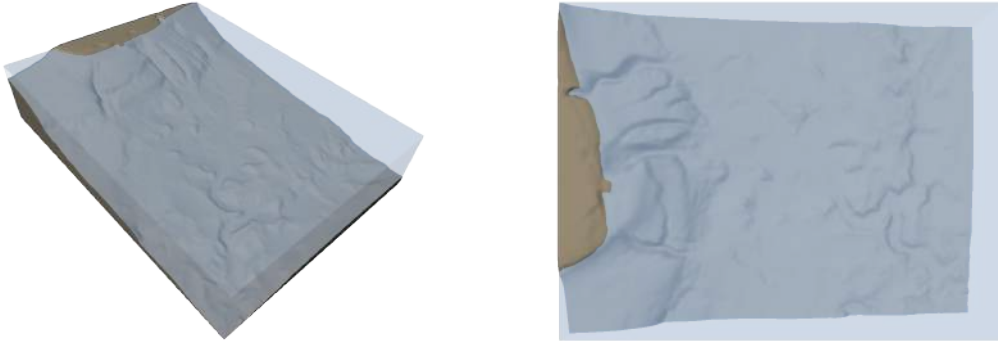


Figure 3.4: Example of an initial depositional surface (in **brown**) with the starting water level (in **blue**) at the external stratigraphic application.

the natural boundary conditions (2.58). Additionally, to work with a mesh that has a smooth boundary, which supposedly can better represent natural features, the actual mesh used in the flow simulation is created by remeshing the wet domain using the Gmsh [108] application (version 4.2.0). Nonetheless, we still create quadrilateral elements due to deal.II's requirements.

With Gmsh, we mesh a 2D surface delimited by patches of B-splines whose control points are obtained from the original grid. In general, we get these locations from the outer faces of the domain's boundary cells, as depicted by the black lines and dots in Figure 3.5a. Corner cells are diagonally traversed to avoid introducing corners in the new triangulation. Also, we create a spline patch for each subdivision

of the contour containing cells with the same boundary condition. This way, we properly delimit the transition between different boundary constraints in the resulting mesh, identifying them for use by the simulator. Figure 3.5b shows the splines and mesh generated for the area in Figure 3.5a. An example of a complete mesh can be seen in Figure 3.6.

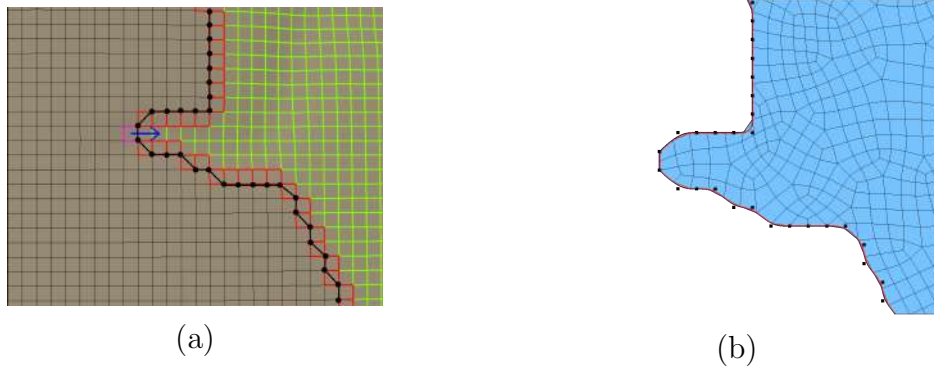


Figure 3.5: Detail of the domain’s contour in the original (a) and generated (b) meshes. The **black** dots in (a) are the control points used to build the **red** B-splines in (b).

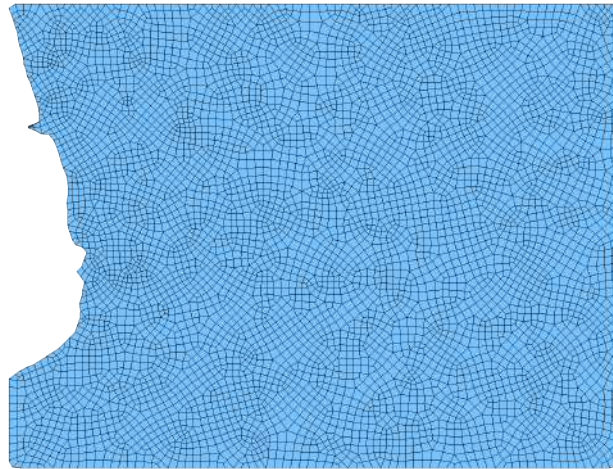


Figure 3.6: Mesh generated for the underwater region illustrated in Figure 3.4.

Next, consider the practical example of the region illustrated in Figure 3.7a, which could represent an island near a coastal region. In this scenario, we have arbitrarily positioned a turbidite event inflow along the **pink** sub-contour. Plus, we have two disconnected coast boundaries (in **green**) and a large **blue** open border. While traversing the wet grid cells and selecting control points for the splines, we make sure the positions at the transitions between two boundary types are included so we properly delimit them. Moreover, observe that, in this case, we also need to define an internal boundary for the domain. To allow Gmsh to distinguish between internal and external curves, producing “holes” in the resulting mesh, we sort the

points that define them, respectively, in clockwise and counter-clockwise orders. The generated mesh for this configuration can be seen in Figure 3.7b.

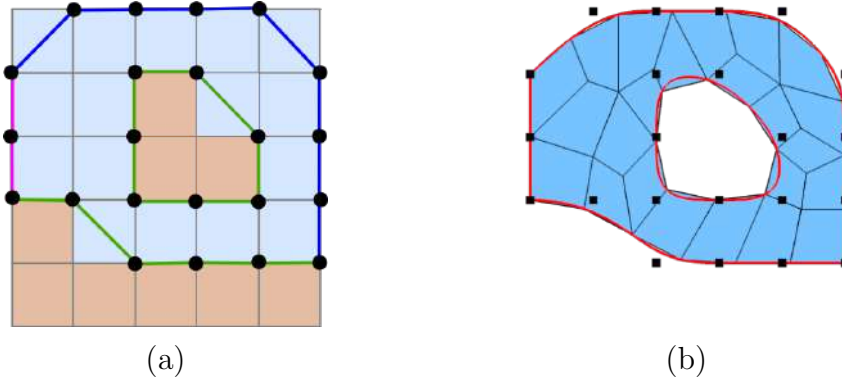


Figure 3.7: (a) Example of a simulation scenario on the stratigraphic grid. The **brown** cells are dry while the **light blue** cells are underwater. Also, we identify two **green** coastlines, a **blue** open boundary, and a **pink** border that takes event inflows. (b) Mesh produced for the setup in (a). The B-splines in **red** were created using the **black** dots as control points.

When the external stratigraphic application requests a flow simulation, we evaluate the current scenario and create a geometry script file with the `.geo` extension. This file will contain the instructions, in Gmsh’s built-in scripting language, needed to create the mesh. As an example, for the case presented in Figure 3.7, the produced script file will have the content displayed in Figure 3.8. It starts with the definition of a reference element size for the produced mesh, which we usually set as equal to the mean edge length of the cells in the stratigraphic grid. Then, we list the coordinates of all control points and define the distinct B-spline patches. Also, by associating splines with physical lines, we identify each boundary with an *id* that we can use to enforce the appropriate constraint on each section of the contour. For a fact, in the resulting mesh, the edges on the domain’s boundary will indicate to which boundary they belong. In sequence, we use the definition of the line loops to determine the surface to be triangulated. As Gmsh creates triangular meshes by default, we have to indicate that it should recombine the triangles into quads. At last, we export the generated mesh using the `.msh2` format so we can import it into our solver using deal.II’s API.

3.4 Sedimentation

In this section, we propose a novel sedimentation algorithm to simulate the formation of turbidite systems in stratigraphic scales. To this end, we employ all the toolsets previously discussed in the thesis. However, first, it is important to review some sequence stratigraphy concepts and introduce the depositional features our

```

// Reference element size:
size = 1;
// Points on external boundary:
Point(1) = {0, 2, 0, size};
Point(2) = {1, 2, 0, size};
Point(3) = {2, 1, 0, size};
Point(4) = {3, 1, 0, size};
Point(5) = {4, 1, 0, size};
Point(6) = {5, 1, 0, size};
Point(7) = {5, 2, 0, size};
Point(8) = {5, 3, 0, size};
Point(9) = {5, 4, 0, size};
Point(10) = {4, 5, 0, size};
Point(11) = {3, 5, 0, size};
Point(12) = {2, 5, 0, size};
Point(13) = {1, 5, 0, size};
Point(14) = {0, 4, 0, size};
Point(15) = {0, 3, 0, size};
// Points on internal boundary:
Point(16) = {2, 2, 0, size};
Point(17) = {2, 3, 0, size};
Point(18) = {2, 4, 0, size};
Point(19) = {3, 4, 0, size};
Point(20) = {4, 3, 0, size};
Point(21) = {4, 2, 0, size};
Point(22) = {3, 2, 0, size};
// Coast boundaries:
BSpline(1) = {1, 2, 3, 4, 5, 6};
Physical Line(1) = {1};
BSpline(4) = {16, 17, 18, 19, 20, 21, 22, 16};
Physical Line(4) = {4};
// Open boundary:
BSpline(2) = {6, 7, 8, 9, 10, 11, 12, 13, 14};
Physical Line(2) = {2};
// Event boundary:
BSpline(3) = {14, 15, 1};
Physical Line(3) = {3};
// External boundary line loop:
Line Loop(5) = {1,2,3};
// Internal boundary line loop:
Line Loop(6) = {4};
// Define plane surface:
Plane Surface(1) = {5,6};
Recombine Surface{1};
Physical Surface(1) = {1};
// Create the mesh:
Mesh.Algorithm = 8;
Mesh.RecombinationAlgorithm = 1;
Mesh 2;

```

Figure 3.8: Contents of the Gmsh script file used to create the mesh in Figure 3.7b. We highlight the definition of the domain’s boundaries with the same colors used in Figure 3.7a: coast boundaries in **green**, open borders in **blue**, and contours with events in **pink**.

approach aims to reproduce. As its name implies, the base working unit of sequence stratigraphy is called a sequence, which consists of a series of genetically related strata. This means that the sediments were deposited in a single cycle of relative sea-level change and that they have lateral continuity, obeying Walther’s law [109]. Sequences are often decomposed into the lower order parasequences, which are separated by marine flooding events.

In turn, parasequences are classified according to their depositional stacking patterns. A major conditioning factor for the different stacking patterns is the relation between the available accommodation space and the sediment inflow rate. In the case of the potential space for sedimentation, its variation is determined, for instance, by changes of the seafloor (subsidence) and the sea level (eustasy). If we consider the sea level as the single modifying factor of the accommodation space, if it rises and we have a low sediment influx rate, we get a retrogradational stacking pattern. In this case, deeper water deposits move towards the land, laying

over shallower water deposits. Meanwhile, if the sediment flux rate matches the sea level rise, we obtain an aggradational sequence in which sediments are deposited atop others of the same type. Also, if the influx sediment rate exceeds the sea level increase, we see a progradational arrangement where shore and nearshore deposits move into the ocean, overlying deeper-water deposits. All these stacking patterns are illustrated in Figure 3.9.

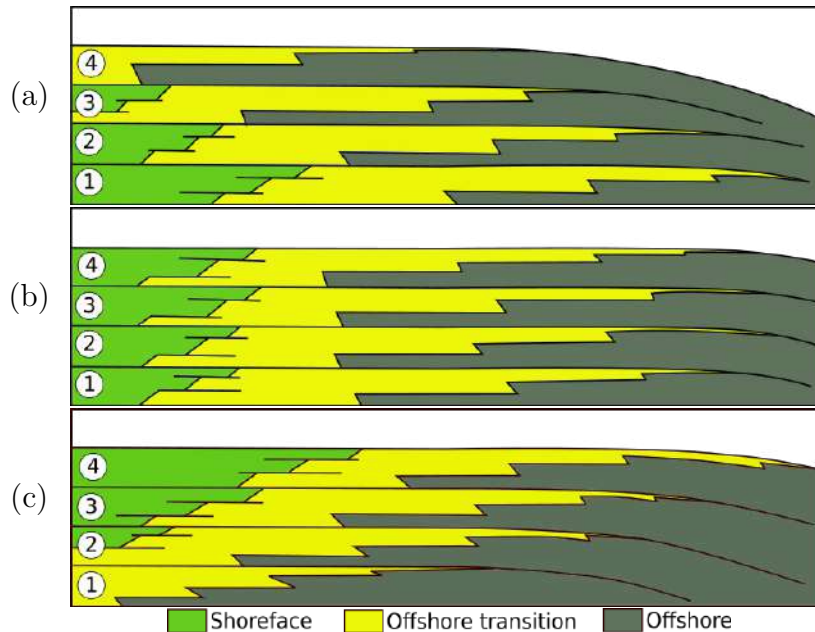


Figure 3.9: Different parasequence stacking patterns. Here the numbers denote the distinct parasequences in a set. (a) Retrogradational parasequence set; (b) Aggradational parasequence set; (c) Progradational parasequence set. Adapted from WAGONER *et al.* [110].

For turbidity currents, the Bouma sequence [111] describes the ideal vertical succession of strata deposited by low-density turbidity currents. One of its characteristics is that the sediments get finer-grained from the bottom to the top of the deposit. Laterally, this produces stacking patterns analogous to retrogradational parasequence sets. Besides, as a sequence of turbidite events takes place, progradational patterns emerge at the distal portion of the turbidite system, as the bulbs created by each event merge in progradation [112]. These progressive advances, together with channel avulsions, build the radial pattern of the depositional lobe. Here it is important to observe that, in this work, we do not intend to reproduce the intricate structure of the bulbs or the lobe channels, but the overall stacking pattern on a larger scale.

Next, we provide a brief review of the technique proposed by CARVALHO *et al.* [7–9], since it was the starting point of our research in what concerns the sedimentation algorithm. At first, it is important to observe that their approach sought to simulate the transport and deposition of siliciclastic sediments in platform, slope

and basin environments, without any concern on turbidite systems. Plus, since their method is also based on sequence stratigraphy, the available depositional space and sediment influx rate are essential controlling factors of the sedimentation.

The accommodation space is changed by applying, to the local sea level, eustatic variations determined by the Haq sea-level curve [113]. Moreover, to account for tectonic/thermal subsidence effects, it is employed a table with subsidence rates per geologic time that is sampled at a few positions throughout the studied region. These data can come, for instance, from previous well log studies, and are extrapolated to the entire domain so we can displace the whole sea bed. In turn, a sediment inflow is imposed by placing sediment supplies along the coastline. Each supply provides a user-defined volume of sand, silt and clay, being usually associated with an inflow boundary condition in the solved flow problem. To aid the estimation of these quantities, CARVALHO [8] provides a set of tables with reference values for the sediment discharges in different environments, as well as the individual volume concentration of each lithology type.

For each sedimentation step, after updating the sea level and the sea bed elevations, CARVALHO [8] solves the steady-state flow problem:

$$\frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2} = f(x, y), \quad (3.20)$$

where $f(x, y)$ is the sea bottom bathymetry and φ is the flow velocity potential, from which we can compute the flow velocities $\mathbf{u} = \nabla \varphi$. In fact, he solves a boundary value problem on a 2D regular grid using the finite difference method. Neumann boundary conditions are enforced so we can impose velocity values on the contour, although only the velocity component orthogonal to a boundary face can be prescribed. Then, the resulting velocity field is used to compute streamlines that begin from grid cells that contain sediment supplies. In this case, a 4-th order Runge-Kutta integration algorithm is employed. Afterward, the streamlines are used to transport and deposit sediments. Initially, each line receives an equal part of the total sediment available at the corresponding sediment supply. In sequence, each streamline is traversed from start to end while trying to deposit sediments in each grid cell crossed by it. Sediments will deposit if the local bed elevation slope along the line is less than a predetermined stability angle for each lithologic fraction. When a deposition should happen, sediments are deposited up to the wave base level, which is the maximum depth that waves would erode the deposits, or until the streamline does not carry any more sediments. The deposition itself is performed following the decreasing order of the granulometric curve (i.e., in the order: sand, silt and clay) and creating blocks of the deposited sediment type over the current seabed. After the deposition, a sediment compaction phase takes place.

Altogether, the approach introduced by CARVALHO [8] is intrinsically different from ours, since we focus on the simulation of turbidity currents. Besides, one of the main differences between them is that, instead of carrying sediments along streamlines built from a steady flow, we perform this task based on streaklines evolved during the simulation of transient flows produced by recurrent turbidite events. In addition, we radially scatter sediments from the cells crossed by the streaklines and impose additional constraints on the accumulation space.

In this context, we link sediment supplies to the boundary cells from which currents are to enter the domain. We consider that a supply can have five granulometric fractions - clay, silt and fine, medium and coarse sands - although each supply can have different proportions of them. Moreover, we place streakline sources in each of these boundary cells along a line that crosses the cell center, and that is parallel to the external boundary. In this case, we let the number of sources per cell be an input parameter. For the sedimentation examples presented in this work, we always assign five streakline sources to each cell.

After an event's flow simulation, the final state of the associated streaklines is used in the sedimentation procedure. By doing so, it is implied that, for the time and space scales employed in stratigraphic studies, the terminal streakline state is a sufficiently good approximation to the average path of the sediments carried by the current. For instance, if we consider a stratigraphic simulation with a time step of 0.1 Ma and a grid with $1 \text{ km} \times 1 \text{ km}$ cells, changes that happen in the order of seconds or hours and that displace the pathlines by some meters may not significantly impact the quality of the results. We remember that, with a similar rationale, CARVALHO [8] performed steady-state simulations. That is, in this perspective, these changes are so small that the time dependence can be neglected.

With this in mind, the first step of the sedimentation algorithm is to evenly distribute the volume available at the supplies - which may not be strictly equivalent to the simulated event's volume - between the associated streaklines. Thus, the more streaklines we have per supply, the less amount each one carries. Then, the lines are jointly traversed from start to end, while sediments are carried, scattered and deposited. The general idea of this process is to visit each cell crossed by a specific streakline, deposit sediments up to the cell's available depositional (or accommodation) space, radially spread sediments to neighboring cells, leaving the remaining sediments to the next cell on the line. Do note that the order in which the streaklines are used for the deposition can affect the resulting deposits. To reduce the lateral bias that can be induced by such ordering, the sequence in which we advance along each streakline progresses from the outermost lines to the innermost, as depicted in Figure 3.10.

This whole operation is done for each granulometric fraction in the decreasing

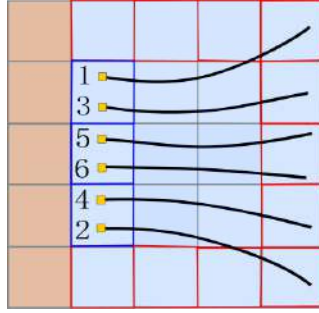


Figure 3.10: Representation of the order in which we advance one step along each streakline. This sequence goes from the outermost to the innermost lines and is given by the numbers beside each streakline source in **yellow**. Here the cells with the **blue** highlight are associated with the same sediment supply.

order of the granulometric curve, i.e., in the sequence: coarse, medium and fine sands, silt and clay. When switching from one grain size to the next, the transport process is resumed from the point on the streakline where the deposition of the previous lithology type ended. An overview of the full algorithm for the transport and deposition of sediments along the streaklines can be seen in Algorithm 4.

The accommodation space available for the deposition of the sediments injected by a turbidite event is initially defined by the volume between the initial seabed and the water level. Alternatively, when it is known the paleobathymetric surface chrono-correlated to the final time of the stratigraphic time step, it can be used as an upper limit to the sediments' deposits. Notwithstanding, we propose to impose another constraint on the accommodation space. We only allow the deposit of a single event to be as thick as the simulated current's point-wise maximum height. To obtain this property, after each time step of the flow simulation, we update the maximum current height of each mesh node so we have, by the simulation end, $h_{\max}(x, y) = \max_t [h(x, y, t)]$. To use this scalar field in the definition of the available space, we evaluate it at the center of each stratigraphic grid cell using finite element interpolations. Still, spaces not occupied by an event's deposits are left as available for the next event.

For completeness, it should be noted that, in the works of CARVALHO *et al.* [7–9], the deposition of the sediments carried from one cell to the next along a streamline is conditioned by the bathymetric gradient. It only occurs if the bed surface gradient is less than the stability angle of at least one lithology type. However, in this work, we consider that deposition always happens as it is already limited by the maximum current heights. At the same time, to mimic the effect of a stability angle, and following the work of RAYMOND [114], when we spread the sediments from a given cell, we reduce the accommodation volume as we distance from it. This is done by

Algorithm 4 Summary of the algorithm used to transport and deposit sediments along the streaklines. Here, `availableVolume(l_t, s)` holds the volume of the lithology type l_t currently available in the streakline s ; `nextUnvisitedCell(s)` returns the next cell to be visited along the streakline s ; `accommodationVolume(c)` determines the current accommodation volume at the cell c ; `depositSediments(vol, l_t, s_u, c)` deposits a sediment volume vol of the type l_t from the streakline s_u into the cell c ; and `spreadSediments(l_t, c, s_u)` radially spreads sediments of the type l_t from the cell c , considering the volume available at the line s_u .

```

1: Compute availableVolume( $l_t, s$ ),  $\forall l_t, \forall s$ , from the sediment supplies' data.
2: for each lithology type  $l_t$  do
3:   while exists an untraversed streakline in respect to  $l_t$  do
4:     for each untraversed streakline  $s_u$  do
5:       if availableVolume( $l_t, s_u$ ) = 0 then
6:         Set  $s_u$  as traversed in respect to  $l_t$ ;
7:         continue.
8:       end if
9:        $c = \text{nextUnvisitedCell}(s_u)$ ;
10:       $vol = \min[\text{availableVolume}(l_t, s_u), \text{accommodationVolume}(c)]$ ;
11:      depositSediments( $vol, l_t, s_u, c$ );
12:      spreadSediments( $l_t, c, s_u$ );
13:      Mark cell  $c$  as visited by the traversal of the line  $s_u$ ;
14:      if  $c$  is the last cell crossed by  $s_u$  then
15:        Set  $s_u$  as traversed in respect to  $l_t$ .
16:      end if
17:    end for
18:  end while
19: end for

```

the scaling, during a sediment spread operation:

$$\text{accommodationVolume}(c) = \frac{\text{accommodationVolume}(c)}{f_s^{\text{dist}(c, c_s)}}, \quad (3.21)$$

where c is the current cell being visited and c_s is the cell that is the source of the sediment spread process that reached c . Also, the function $\text{dist}(c, c_s)$ measures the Euclidean distance between the indexes (i, j) of the cells c and c_s on the regular stratigraphic grid. f_s is a factor that controls the height profile and the maximum extent of the deposits created as a result of scattering sediments from a single cell. Although this parameter can be given a different value for each lithology type, we assign $f_s = 2$.

Optionally, we allow the deposition of an event to be made in sub-steps that have limited accommodation space with respect to the one available for the entire event. We split the maximum depositional space equally between the sub-steps, although spaces not occupied at one sub-step is left available for the next. Allied

to the fact that the injected sediment volume is not split between the sub-steps, the imposed proportion between sediment influx and accommodation volume conditions the resulting event deposits to have a retrogradational-like stacking pattern. It makes the sediments finer-grained from the bottom to the top of the deposit, as in the well known Bouma sequence. Otherwise, without using sub-steps, the different grain-sizes settle in blocks, and the only stratifications we see inside an event's deposit are due to the factor f_s . Therefore, as illustrated in Figure 3.11, with the proper setup, we can perform a simulation where the deposits of a sequence of events produce a stacking pattern analogous to the progradational parasequence sets. Still, the deposits of every single event exhibit fining-upwards trends.

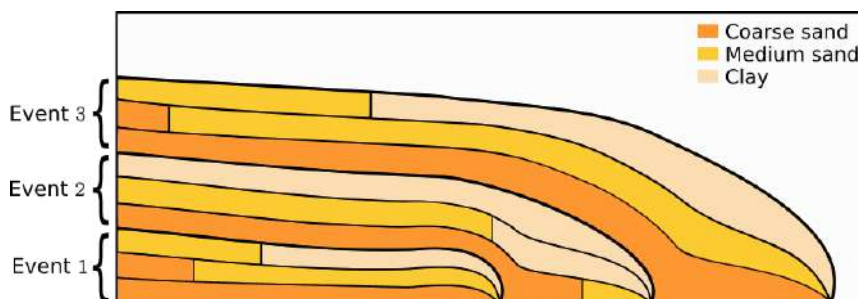


Figure 3.11: Schematic representation of a possible result of the sedimentation of three events using three sub-steps per event.

The deposition at a cell is accomplished by computing the volume of the specific granulometric fraction to be deposited, removing this amount from the streakline in question and adding a pillar of the same volume and material to the top of the cell's deposit. The amount of sediments to be accumulated is specified trying to fill all the space available in the cell, while being constrained by the volume still carried by the streakline. Also, the procedure that radially spreads sediments from a cell crossed by a streakline visits the cell's neighbors in a queue-based flood-fill algorithm where, when a cell is visited, all its neighbors are immediately added to the queue. In Figure 3.12, we show in which sequence the flood-fill algorithm will visit cells, starting from the hatched one. The left-most group of cells depicts the order that the immediate neighbors of the hatched cell are added to the queue. Next, as the cell 1 is visited, its neighbors are included in the queue, if they are not already there. In this case, the cells adjacent to 1 are enqueued in the same sequence as the neighbors of the hatched cell. Then, in the following three cell groups, we visit the cells 2, 3 and 4, and add their respective surrounding cells to the queue. It is important to observe that, in this process, while the deposition on each visited cell is constrained by its accommodation space, it is also limited by the volume available at the streakline that crossed the initial cell.

After the deposition phase of each event, we compact the accumulated sediments as a result of the weight added by the new layers deposited. That is, the layers below

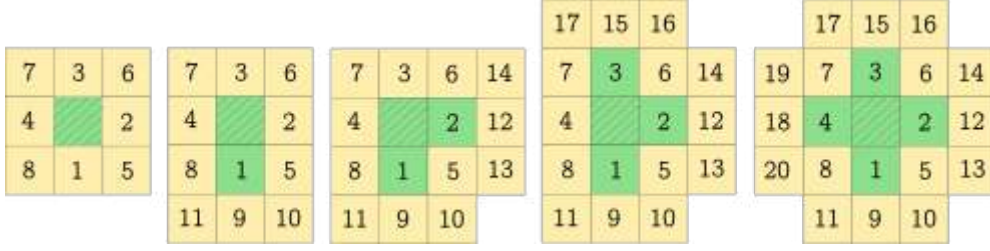


Figure 3.12: Representation of the first steps of the flood-fill algorithm employed to scatter sediments from the hatched grid cell. The **green** color denotes the cells that were visited by the algorithm and the **yellow** color stands for the cells waiting in the queue to be visited. The numbers show the order in which the cells are to be visited.

the top one are compressed due to the decrease in porosity caused by the weight overload. Following the works of GILES *et al.* [115] and CARVALHO *et al.* [7–9], we consider the porosity ϕ to be a function of the buried depth z , being defined as:

$$\phi(z) = \phi_0 e^{-c_b z}, \quad (3.22)$$

where ϕ_0 is the initial/surface porosity and c_b is the compaction coefficient. We obtain the initial values for the porosity of each granulometric fraction and the compaction coefficients from HANTSCHHEL and KAUEAUF [116]. Thus, we adopt $\phi_0^{\text{clay}} = 0.7$, $c_b^{\text{clay}} = 8.3 \times 10^{-4} \text{ m}^{-1}$, $\phi_0^{\text{silt}} = 0.55$, $c_b^{\text{silt}} = 5.1 \times 10^{-4} \text{ m}^{-1}$ and, for all sands, $\phi_0^{\text{sand}} = 0.41$, $c_b^{\text{sand}} = 3.1 \times 10^{-4} \text{ m}^{-1}$.

The idea here is to compact the pillar of sediments in each cell independently. Starting from the second layer of sediments in the top, going down to all the subsequent layers, we fix its top depth z_t and compute a new depth z_b for its bottom. To conserve mass, we require that the total sediment mass:

$$m_{\text{sed}} = \int_{z_b}^{z_t} \rho_{\text{sed}} (1 - \phi) dz, \quad (3.23)$$

of density ρ_{sed} , does not change in the process. In this sense, the mass balance between the states 1 and 2, which refer, respectively, to the configuration of a sediment layer before the deposition of the top layer and after the compaction, is given by:

$$\begin{aligned} \int_{z_b^1}^{z_t^1} \rho_{\text{sed}} (1 - \phi) dz &= \int_{z_b^2}^{z_t^2} \rho_{\text{sed}} (1 - \phi) dz, \\ \therefore \int_{z_b^1}^{z_t^1} \rho_{\text{sed}} (1 - \phi_0 e^{-c_b z}) dz &= \int_{z_b^2}^{z_t^2} \rho_{\text{sed}} (1 - \phi_0 e^{-c_b z}) dz \\ \therefore z_t^1 - z_b^1 + \frac{\phi_0}{c_b} \left(e^{-c_b z_t^1} - e^{-c_b z_b^1} \right) &= z_t^2 - z_b^2 + \frac{\phi_0}{c_b} \left(e^{-c_b z_t^2} - e^{-c_b z_b^2} \right). \end{aligned} \quad (3.24)$$

From this balance, we compute the new buried depth z_b of a sediment layer iteratively:

$$z_b^2 = z_t^2 - z_t^1 + z_b^1 + \frac{\phi_0}{c_b} \left(e^{-cz_t^2} - e^{-c_b z_b^2} - e^{-c_b z_t^1} + e^{-c_b z_b^1} \right), \quad (3.25)$$

At last, we remark that the adopted sedimentation technique exactly conserves the sediment mass even if the flow simulation were to present some mass error. In our practical algorithm, all supplied sediments are transported and should deposit up to the available depositional space, which is limited by the maximum envelope of the associated turbidity current throughout the flow simulation. However, if an event has an excessive amount of volume, the current (and the sediments) may leave the area of interest. Still, the outflow sediment volume can be easily tracked. Also, the compaction operation conserves mass as Equation (3.25) is solved up to machine precision.

Chapter 4

Implementation Aspects

Although the focus of this work is the proposed ensemble of methods and techniques that allows simulating the formation of turbidite system in stratigraphic scales, and not the computational implementation per se, in the present chapter, we provide some insights regarding how it was done. Notably, we regard the flow solver based on the proposed FCT scheme as well as the streakline evolution procedure. In contrast, we do not examine the sedimentation algorithm since it consists of a much simpler and straightforward implementation. Besides, it was operationalized in a stratigraphic application whose base algorithms and data-structures come from the work of CARVALHO *et al.* [7–9].

All the solver’s finite element related implementations were aided by the deal.II [55] C++ software library (version 9.0.0). Its use speeds up implementations while providing efficient data-structures for the finite element and linear system routines. For an analysis of the efficiency of the algorithms and data structures provided by deal.ii, we refer to the works of BANGERTH and KAYSER-HEROLD [73] and BANGERTH *et al.* [117]. Another deciding factor for its use is it being multiplatform, so our program would run natively on both Windows and Linux systems. deal.II supports quadrangular finite elements of different types and orders, although here we only use bilinear Lagrange elements. Another advantage of deal.II is that it handles mesh refinement/coarsening transparently to the user. Besides, it also supports parallel computing, either with shared or distributed memory machines. Among other tools used, we find it important to highlight the Gmsh [108] mesh generator we employ to build the meshes for the flow solver, and the Visualization ToolKit (VTK) [118], whose application programming interface we use to export the solution’s mesh and streakline data, which we take to the ParaView [119] application for analysis.

Now, we present key aspects of the base configuration of the solver using deal.II. Thus, unless stated otherwise, all classes and functions we discuss are part of its library, being defined in the `dealii` namespace. We start from the mesh data-

structure, which is of type `Triangulation<2>`, defining a surface immersed in the two-dimensional real space. As an automated procedure generates the meshes used in the simulations, we found it useful to refine them anisotropically to avoid elements with high aspect ratios using `GridTools::remove_anisotropy(..)`. Also, we remove all hanging nodes in the mesh with `GridTools::remove_hanging_nodes(..)`, which performs another refinement step. Hanging nodes are usually obtained during mesh refinements, being nodes that do not belong to all adjacent elements, as depicted in Figure 4.1.

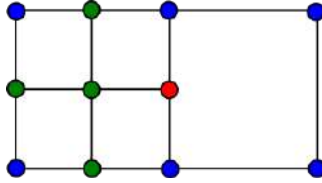


Figure 4.1: Example of a hanging node (in **red**) created from the refinement of the original left cell. While it belongs to the elements on its left side, it is not part of the element on the right. The nodes in **blue** define the coarse elements, and the **green** nodes were also created by the refinement step.

In `deal.II`, objects of the type `DoFHandler<2>` can be used to enumerate the degrees of freedom on all vertices. We create one object of this type for the problem solution and another one for the bed elevation, since the former is vector-valued and the latter contains scalar quantities. Also, the nodal values of both the solution and the bed elevation are stored in separate `Vector<double>` objects. The main finite element structure is defined by `FESystem<2>(FE_Q<2>(1),3)`, which groups 3 Lagrange finite elements of degree 1, defined on the $2D$ space. Similarly, for the bed elevation, we use `FESystem<2>(FE_Q<2>(1))`. Numerical integration is performed using the Gauss quadrature `QGauss<2>(2)` that produces 2 quadrature points in each space direction. In turn, the evaluation of a finite element in a cell's quadrature points is stored into a `FEValues<2>` object. Initial conditions are enforced using the function `VectorTools::interpolate(..)`. One of its parameters is a function object of type `Function<2,double>`, responsible for computing the initial solution at each position (x,y) . To define the nodal bed elevations, we apply the same procedure.

While the right-hand side of the assembled linear system is a standard vector (of type `Vector<double>`), its left-hand side is a sparse matrix (i.e., a `SparseMatrix<double>` object). Its sparsity pattern is built using the function `DoFTools::make_sparsity_pattern(..)`, which takes the current `DoFHandler` as an argument. However, prior to creating the sparsity pattern, we reorder the degrees of freedom using `DoFRenumbering::Cuthill_McKee(..)` that implements the algorithm proposed by CUTHILL and MCKEE [79]. To deal with constraints

on degrees of freedom, we use the `AffineConstraints<double>` class. Dirichlet boundary conditions (including those associated with events) are accounted into the constraints by `VectorTools::interpolate_boundary_values(...)`. In addition, the non-penetration condition imposed on the mesh contours associated with coastal lines are defined by `VectorTools::compute_no_normal_flux_constraints(...)`. At last, the linear system itself is solved using the preconditioner `SparseILU<double>` and the solver `SolverGMRES<>`.

Before giving more details about the implementation of the FCT scheme used to advance a solution in time, it is important to know how we initially define the time steps employed in the time marching algorithm. To compute them according to the condition (2.157), we use a quadrature rule with a single point positioned on the unit cell barycentre, being defined as `Quadrature<2>(Point<2>(0.5, 0.5))`. The use of the quadrature value instead of the nodal ones is because the constraint is based on the current's velocity, computed from the specific discharges using Equation (2.158), which depends on the cell area. Thus, we choose the maximum time step that makes the requirement (2.157) hold for the barycentre of each mesh cell.

In regard to the FCT scheme, both the low- and high-order equations are solved in two main steps. First, we compute the terms dependent on \mathbf{V}^n . For the low-order system, this is where we verify if the condition (2.132) on the time step is met. If not, we choose a suitable time step and reassemble the matrices. For the high-order system, we compute and limit the anti-diffusive fluxes. They are built into a global sparse matrix where each entry (i, j) denotes the flux from the i -th to the j -th degree of freedom. Thus, it can only have non-null values where the associated degrees of freedom belong to adjacent nodes. Besides, the fluxes are computed based on previously assembled sparse matrices, namely the artificial diffusion matrix and the consistent mass matrix. This operation is performed iterating through each node and its adjacent ones in parallel. To help identify the adjacent nodes of any node in the mesh, we take advantage of the function `GridTools::get_dof_to_support_patch_map(...)`, which gives us a map of each degree of freedom to the cells that share them. In sequence, the second step consists of the non-linear iterations where we assemble the terms that depend on \mathbf{V}^{n+1} and solve the linear system to obtain a solution.

To speed-up the computations needed for the FCT scheme, we employed `deal.II`'s module for parallel computing in shared memory devices. In particular, we benefit from the `WorkStream` namespace that allows operating on streams of cells/elements in parallel, while making all the synchronization and load balance tasks transparent to the user. For instance, when assembling the linear system matrices, it enables the local contributions to be computed in parallel. In contrast, their copy to the global matrices is done sequentially in a stable and reproducible order. To allow the worker

functions to use data that should not be shared between threads, deal.II defines a template object called `ScratchData`, which threads can use as private data. For example, we use it to store the local solution and the bed elevation evaluated at an element's quadrature points. Also, we keep the local copy of the nodal bed elevations that will be corrected using (2.159). In this context, the bed elevations at the quadrature points are computed from the corrected nodal values. In complement to the `ScratchData`, worker functions can store their results in objects of the template type `CopyData`, which can be used by a separate function to transfer the results to the global matrices. As an example, we store in a structure of such type, the local element contributions to the linear system matrices and the artificial diffusion matrix, as well as the mapping from the local to the global degrees of freedom indexes. After objects of the types `ScratchData` and `CopyData` have been used by a thread, they may be reused by other threads, avoiding spending computational time allocating and deallocating memory.

In terms of the recurrent events, we always keep a record of the number of recurrences left and the time of the next episode start. After activating an event, we start tracking the total volume it has injected into the domain, so we can know when to deactivate it. Thus, we need compute the increments defined in Equation (3.1). To compute this integral along the boundary edges where an event is active, we use the quadrature rule `QGauss<1>(2)` together with an object of type `FEFaceValues<2>` to obtain the evaluation of a finite element in quadrature points on an element face. In addition, when an event is activated or deactivated, we reset our instance of `AffineConstraints<double>`, so the new boundary conditions can be correctly enforced.

In sequence, we make some remarks about the streakline computation. In this sense, we observe that, to compute the velocities needed in the Runge-Kutta scheme (3.17) used to displace streakline points, we need to know the element where each one resides and whose nodal values will be interpolated for a desired position. The simplest way to perform this is to search the entire mesh at each value computation. However, the performance is increased if we store, for each particle, the current cell in which it resides. Thus, our streakline data-structure consists of an ordered list of pairs that hold a `Point<2>` object and an iterator of type `DoFHandler<2>::active_cell_iterator` that points to the container element. Hence, when moving a particle, we search for the new container cell starting from the current cell, going to its neighboring elements.

First, we find out to which vertex of the current element the desired position is closest. Then, using a cached map of every vertex to its adjacent elements, we test whether the position is inside any of the neighboring cells. The order in which the cells are tested depends on the direction between the particle's position and the

nearest vertex. The test itself is made by projecting the position to local unit cell coordinates and verifying if they are in the range $[0, 1]$. If none of the neighbors contains the particle, then we search the whole mesh for the vertex closest to the particle and proceed to check its adjacent elements. If still no cell is found to contain the particle, we obtain the one it is nearest to and its projection into the cell in local unit coordinates. In addition, we mark the cell as being outside the mesh, so it can be removed if its final state after the displacement is still out of bounds.

One issue that might arise in the previous procedure is that polynomial mappings from unit to real cell coordinates might not be invertible for points outside the cell, as pointed out by the deal.II's documentation on the `Mapping::transform_real_to_unit_cell(...)` function¹. In these cases, the point is classified as outside the cell, and its projection to the local unit coordinates is the coordinates of the nearest vertex of the cell. This implementation is aided by the deal.II's `GridTools::Cache<2>` class and the search itself is a modified version of the `GridTools::find_active_cell_around_point(...)` function.

When we perform a simulation requested by the external stratigraphic application, we need to define the initial state of the solution on each mesh node. For the solution itself, we always assign null values since, initially, there is no turbidite current flowing inside the domain. However, bed elevation values still need to be interpolated from the bathymetric data of the stratigraphic grid. Thus, for a given node, we find out the cell of the regular grid where it resides and use the bed elevations of the centers of the cell and its adjacent ones to obtain a value for the desired node. In this matter, the adopted interpolation method is the well-known inverse squared distance weighting [120].

After simulating the flow of an event, the evolved streaklines are brought to the stratigraphic application, where the event sedimentation phase will take place. At this point, for each streakline, we build an ordered list with the cells crossed by it. This is done as we iterate through all the streakline points and compute the grid cell in which they are located. Also, to prevent a low point density along any section of a streakline from making a cell crossed by it not to be detected and included in the ordered list, we increase the number of points in long segments. Thus, we subdivide the streakline segments whose lengths are greater than half the diagonal of the uniform grid cells. In addition to the streaklines, we also take the computed max current heights to the external application. This is done by using the function `VectorTools::point_value(...)` to obtain a value for the centre of each grid cell. Moreover, to export a solution that we want to import into the ParaView application, we use deal.II's API, employing a `DataOut<2, DoFHandler<2>>` object

¹<https://www.dealii.org/current/doxygen/deal.II/classMapping.html#a38ba6aaa1745359910e1b465a0f5fb27>, visited on January 18th, 2020.

to save a VTK [118] unstructured grid (.vtu files) with the solution of each time step, and the function `DataOutBase::write_pvd_record(..)` to group the files that make up the solution in time. For the streaklines, we export them as VTK polygonal data (.vtp files) that are also indexed with the pvd record file.

Chapter 5

Numerical Results

In this chapter, we present the numerical results of some test cases and make comparisons with analytical and literature available solutions. For each example, the input mesh is created with the Gmsh [108] tool, and the results were brought to the ParaView [119] application for analysis, when not directly examined in the stratigraphic software program. We consider that $\rho_w = 10^3 \text{ kg m}^{-3}$, $\mu = \mu_w = 10^{-3} \text{ Pa s}$ and $n = 0.018 \text{ sm}^{-1/3}$. Also, when it is relevant to compute the mass/volume relative error Vol_{err} , we use the expression:

$$Vol_{err}(t) = \frac{Vol(t) - Vol(0)}{Vol(0)}, \quad (5.1)$$

where $Vol(t)$ is the volume stored in the domain at the time t . For the cases where we simulate the sedimentation process, we consider the color table illustrated in Figure 5.1 to portray the distinct granulometric fractions (and the associated lithology types).






	Coarse sand
	Medium sand
	Fine sand
	Silt
	Clay

Figure 5.1: Color table used to represent the distinct granulometric fractions (and the associated lithology types).

5.1 1D Dam Break

We regard a 1D dam break configuration that describes two reservoirs that contain water and are separated by a dam. One reservoir has fluid with depth h_1 , while, in the other, the fluid height is h_0 , with $h_1 > h_0$. At a given time, the barrier is

instantly removed. Thus, we calculate the subsequent fluid flow. For this arrangement, it is possible to obtain an analytical solution for the state of the fluid after the release. STOKER [121] presented the solution adopted in this work. A schematic representation of the solution and the initial state of the problem can be seen in Figure 5.2.

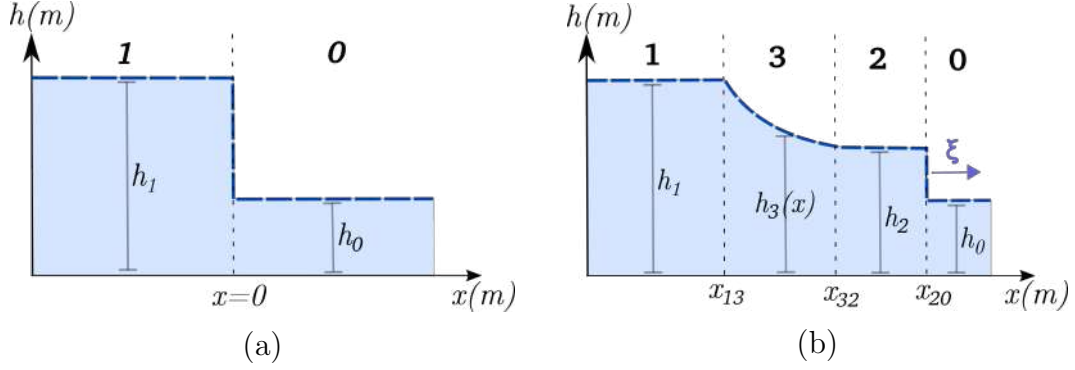


Figure 5.2: Schematic representations of the 1D dam break problem (a) and its analytical solution (b). The bold numbers represent the different regions of the initial state and the analytical solution.

The analytical solution can be divided into 4 regions. In the transition between regions 2 and 0, a shock wave propagates to the right with speed ξ . Also, the fluid has zero velocity in regions 1 and 0 ($u_1 = u_0 = 0 \text{ m s}^{-1}$). Let the propagation speed of a perturbation on the surface of each region be defined as $c_i = \sqrt{gh_i}$, with $i = 0, 1, 2, 3$. Then, the shock wave speed is obtained by solving the nonlinear equation:

$$\frac{\xi}{2} + \frac{c_0^2}{8\xi} \left[1 + \sqrt{1 + 8 \left(\frac{\xi}{c_0} \right)^2} \right] + \left[\frac{c_0^2}{2} \left(\sqrt{1 + 8 \left(\frac{\xi}{c_0} \right)^2} - 1 \right) \right]^{1/2} = c_1, \quad (5.2)$$

and the result is used as an input to the exact solution computation.

In sequence, it is defined:

$$c_2 = c_0 \left[\frac{1}{2} \sqrt{1 + 8 \left(\frac{\xi}{c_0} \right)^2} - \frac{1}{2} \right]^{1/2}, \quad c_3 = \frac{1}{3} \left(2c_1 - \frac{x}{t} \right), \quad (5.3)$$

$$u_2 = \xi - \frac{c_0^2}{4\xi} \left(1 + \sqrt{1 + 8 \left(\frac{\xi}{c_0} \right)^2} \right), \quad u_3 = \frac{2}{3} \left(c_1 + \frac{x}{t} \right), \quad (5.4)$$

where t is the elapsed time since the dam's removal. So, the coordinates of each transition between regions can be defined as:

$$x_{13} = -c_1 t, \quad x_{32} = (u_2 - c_2)t, \quad x_{20} = \xi t. \quad (5.5)$$

Here we consider $x \in [-50, 50]$ m, $t \in [0, 10]$ s, $h_1 = 2$ m and $h_0 = 1$ m, and compute $\xi = 4.183 \text{ m s}^{-1}$. The simulation is performed on a 2D regular mesh with 302×4 rectangular elements comprising a $100 \text{ m} \times 1 \text{ m}$ area. Non-penetration boundary conditions are applied at the limits of the numerical domain. Also, frictional forces are not considered. As FENNEMA and CHAUDHRY [122] mention, this assumption prevents the source term from introducing additional damping to the result. To compute the $YZ\beta$ operator in the stabilized formulation, we use the reference values $(h)_{\text{Ref}} = h_1$, $(q_x)_{\text{Ref}} = h_1 c_1$ and $(q_y)_{\text{Ref}} = 10^{10} \text{ m}^2 \text{ s}^{-1}$.

Figure 5.3 presents the exact solution and the results obtained with the stabilized and FCT methods at $t = 7.5$ s. We observe that the results obtained with the FCT scheme are in better consonance with the analytical solution than the ones produced by the stabilized techniques that use the CAU and $YZ\beta$ shock-capturing operators.

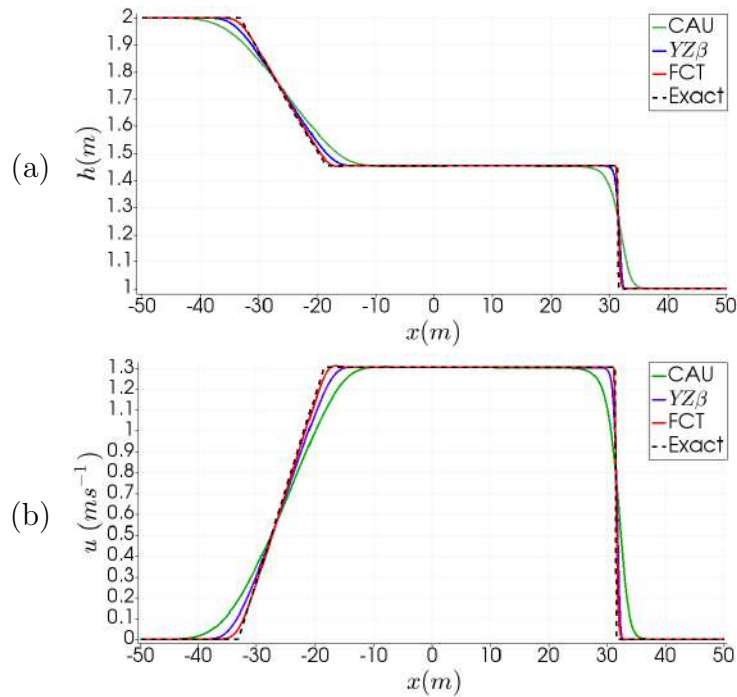


Figure 5.3: Exact and simulated solutions at $t = 7.5$ s of the 1D dam break problem. (a) Fluid height; (b) Water velocity.

5.2 Transcritical Flow with a Shock

We simulate the steady-state solution of a transcritical flow that presents a hydraulic shock. Following DELESTRE *et al.* [123], we neglect frictional forces and viscous stresses ($\mu = 0 \text{ Pa s}$). The simulation domain is a $25 \text{ m} \times 5 \text{ m}$ area whose bed elevation

is defined as:

$$z_b(x) = \begin{cases} 0.2 - 0.05(x - 10)^2, & \text{if } 8 \text{ m} < x < 12 \text{ m}, \\ 0, & \text{otherwise.} \end{cases} \quad (5.6)$$

The adopted boundary conditions are $q(x = 0) = q_0 = 0.18 \text{ m}^2 \text{ s}^{-1}$ and $h(x = L = 25) = h_L = 0.33 \text{ m}$, while the initial state is $q(x) = q_0$ and $h(x) = h_L$. According to DELESTRE *et al.* [123], for this configuration, the exact solution at the steady state can be computed by solving:

$$h(x)^3 + \left(z(x) - \frac{q_0^2}{2gh_c} - h_c - z_{\max} \right) h(x)^2 + \frac{q_0^2}{2g} = 0, \quad \forall x \in [0, x_{\text{shock}}), \quad (5.7)$$

$$h(x)^3 + \left(z(x) - \frac{q_0^2}{2gh_L} - h_L \right) h(x)^2 + \frac{q_0^2}{2g} = 0, \quad \forall x \in (x_{\text{shock}}, L], \quad (5.8)$$

$$q_0^2 \left(\frac{1}{h_+^2} - \frac{1}{h_-^2} \right) + \frac{g}{2} (h_-^2 - h_+^2) = 0, \quad \text{for } x = x_{\text{shock}}, \quad (5.9)$$

where $h_c = (q_0^2/g)^{1/3}$ is the critical water level at the subcritical to supercritical transition, $z_{\max} = 0.2 \text{ m}$ is the maximum bed elevation, and $h_-(x = x_{\text{shock}})$ and $h_+(x = x_{\text{shock}})$ are the water height upstream and downstream of the shock. The shock position x_{shock} can be obtained by solving Equation (5.9). However, the identification of the physically sound roots of the polynomials in Equations (5.7)-(5.9) requires further knowledge about the problem. For this reason, we provide a more complete explanation of the problem and its exact solution in Appendix E. Nonetheless, for the studied configuration, we implemented an Octave [124] script to solve the previous equations and obtain the solution. In this case, $x_{\text{shock}} \approx 11.666 \text{ m}$ and $h_c \approx 0.149 \text{ m}$. Computation is performed on a mesh with 200×10 elements. To calculate the $YZ\beta$ operator, we use the reference values $(h)_{\text{Ref}} = h_L$ and $(q_x)_{\text{Ref}} = (q_y)_{\text{Ref}} = 10^{10} \text{ m}^2 \text{ s}^{-1}$. We also tested it with $(h)_{\text{Ref}} = h_L$, $(q_x)_{\text{Ref}} = q_0$ and $(q_y)_{\text{Ref}} = 10^{10} \text{ m}^2 \text{ s}^{-1}$, but the obtained solution was too oscillatory.

A comparison between the analytical and simulated solutions can be seen in Figure 5.4. We observe that the use of the CAU operator produced an undesired oscillation to the left of the mound. In contrast, the $YZ\beta$ approach created a nonphysical peak at the downstream side of the shock. At last, the FCT scheme generated the best results, which are in good agreement with the exact solution.

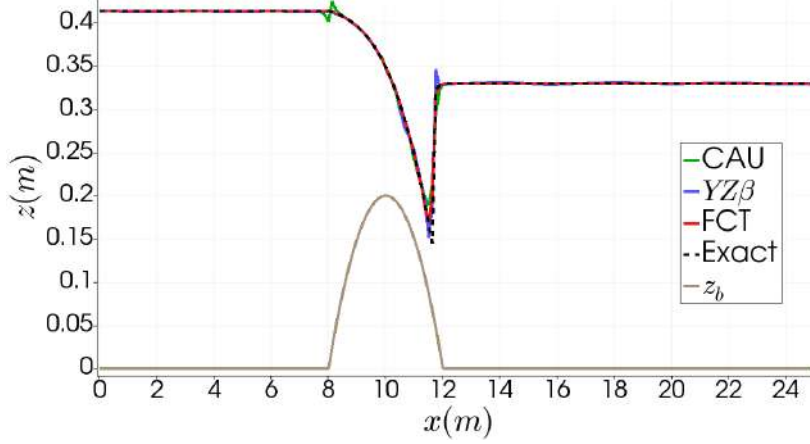


Figure 5.4: Exact and simulated water elevation solutions of the transcritical flow with shock problem.

5.3 1D Flow over an Irregular Bed

We set up a scenario based on a test case of GEORGE [125], where it is considered the flow over an irregular bed. In general, the resulting dynamics encompass dry/wet transitions, shocks and a stationary bore, before ending up in a motionless state. We define the bed elevation as:

$$z_b(x) = \begin{cases} 3e^{-6(0.1x)^2}, & \text{if } x < 15.064 \text{ m,} \\ 1.5e^{-(0.08x-4.8)^2}, & \text{if } 15.064 \text{ m, } \leq x < 89.717 \text{ m,} \\ 3e^{-6(0.1x-10)^2}, & \text{if } x \geq 89.717 \text{ m,} \end{cases} \quad (5.10)$$

while the initial conditions are $\mathbf{q} = \mathbf{0}$ and:

$$\eta(x) = h(x) + z_b(x) = \begin{cases} 2.0, & x < 40 \text{ m,} \\ 1.0, & x \geq 40 \text{ m.} \end{cases} \quad (5.11)$$

Also, non-penetration conditions are enforced on the whole boundary. The simulation is carried on a 2D regular mesh composed of 1 m^2 elements distributed along a $100 \text{ m} \times 4 \text{ m}$ area, and covers a period of 700 s. To compute the $YZ\beta$ shock-capturing operator in the stabilized method, we employ the reference values $(h)_{\text{Ref}} = 1.5 \text{ m}$ and $(q_x)_{\text{Ref}} = (q_y)_{\text{Ref}} = 10^{10} \text{ m}^2 \text{ s}^{-1}$.

We present the water surface profiles obtained at key simulation times in Figure 5.5. At first, the initial shock wave propagates to the right covering the mound, while the right-most reservoir maintains its steady-state equilibrium. As the wave surpasses the mound, its arrival at the second reservoir produces a stationary bore and a shock wave that moves to the right. Then, this wave is reflected by the right-most hill and comes back to the left, shifting and extinguishing the

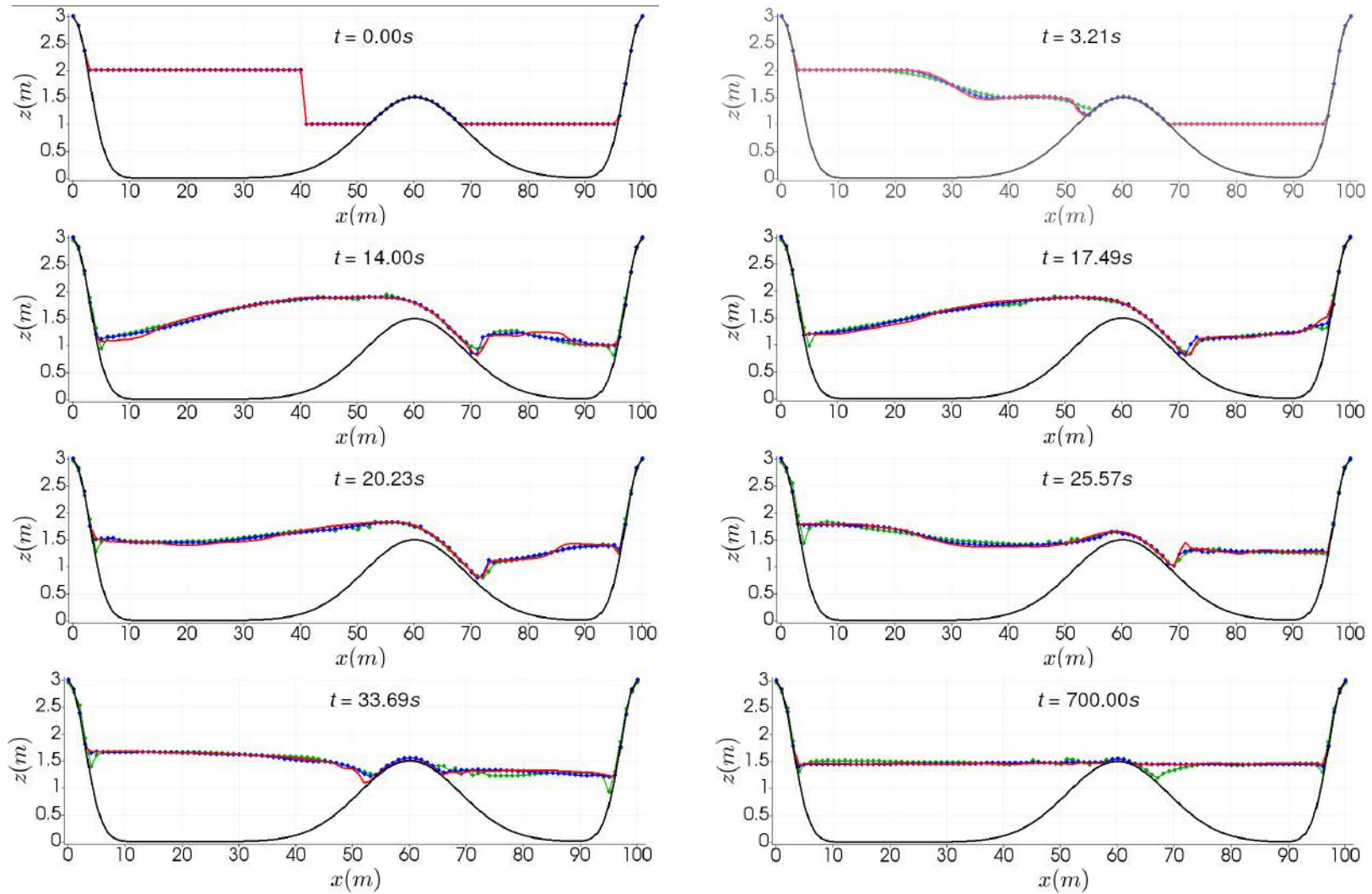


Figure 5.5: Water surface obtained at key simulation times of the 1D flow over an irregular bed problem. The results of the FCT approach are in **red** and the ones obtained using the CAU and $YZ\beta$ operators are, respectively, in **green** and **blue**.

hydraulic shock. Meanwhile, the mound is dried out as the first reservoir’s wet front is retracted back into it. In sequence, a series of similar dynamics that involve the exchange of water between both reservoirs take place, until the fluid stops and its surface remains flat.

Among the tested approaches, the FCT scheme showed the sharpest profiles. Also, the use of the CAU operator failed to reproduce the water elevation near the highest mounds precisely, and it did not reach a state with a flat water surface at $t = 700$ s. By the end of the simulation, the volume error of the FCT scheme was in the order of 10^{-10} , while the error of the $YZ\beta$ and CAU stabilized approaches were in the orders of 10^{-5} and 10^{-3} , respectively.

Moreover, the results of the proposed method are in good consonance with the overall simulation of GEORGE [125]. Note, however, that he does not consider the fluid’s viscosity nor bed friction forces. Hence our solutions could not be the same. Notwithstanding, our simulation shows our approach’s ability to maintain an initial still state (right-most reservoir before receiving the initial wave), and to reach a motionless equilibrium. Besides, dry/wet transitions are correctly handled even at the steeper slopes of the left- and right-most hills while keeping the smallest volume error between the assessed techniques.

5.4 Asymmetric Dam Break

Following previous studies [46, 58, 126–129], we simulate the frictionless inviscid flow triggered by the instantaneous break of the dam separating two reservoirs connected by a channel. The initial water height at the left and right reservoirs are, respectively, $h_L = 10$ m and $h_R = 5$ m. Here, we run the simulation up to $t = 7.2$ s and enforce non-penetration constraints at the whole domain boundary. A diagram of the domain’s geometry, dam’s placement and the initial fluid height distribution can be seen in Figure 5.6a. Additionally, Figure 5.6b depicts a detailed view of the employed 13488-element mesh and the initial water heights near the dam. In this case, to compute the $YZ\beta$ operator, let $h_2 = 7.27$ m and $u_2 = 2.92$ m s⁻¹ be the fluid height and velocity at the region 2 of the 1D dam break problem that has an initial fluid height distribution analogous to the present 2D case. Then, we use $(h)_{\text{Ref}} = h_2$ and $(q_x)_{\text{Ref}} = (q_y)_{\text{Ref}} = h_2 u_2$ as the reference values in the $YZ\beta$ term.

Figure 5.7 shows the final water height distribution with 40 contours between $h = 5$ m and $h = 10$ m. We observe that both stabilized formulations produced undesired water height perturbations past the wave front in the right-most reservoir, where the water surface should be flat, as reproduced by the FCT scheme. Figure 5.8 compares our results with the solutions obtained by RICCHIUTO *et al.* [128] and RICCHIUTO [129]. Among the tested approaches, the one with the CAU operator

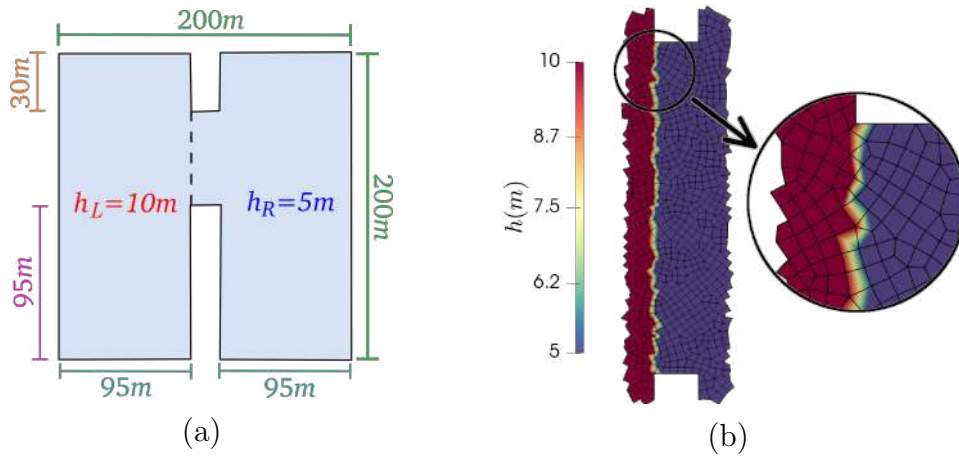


Figure 5.6: Initial configuration of the asymmetric dam break problem. (a) Domain's geometry and initial fluid height distribution; (b) Detail of the mesh and the initial water heights near the dam.

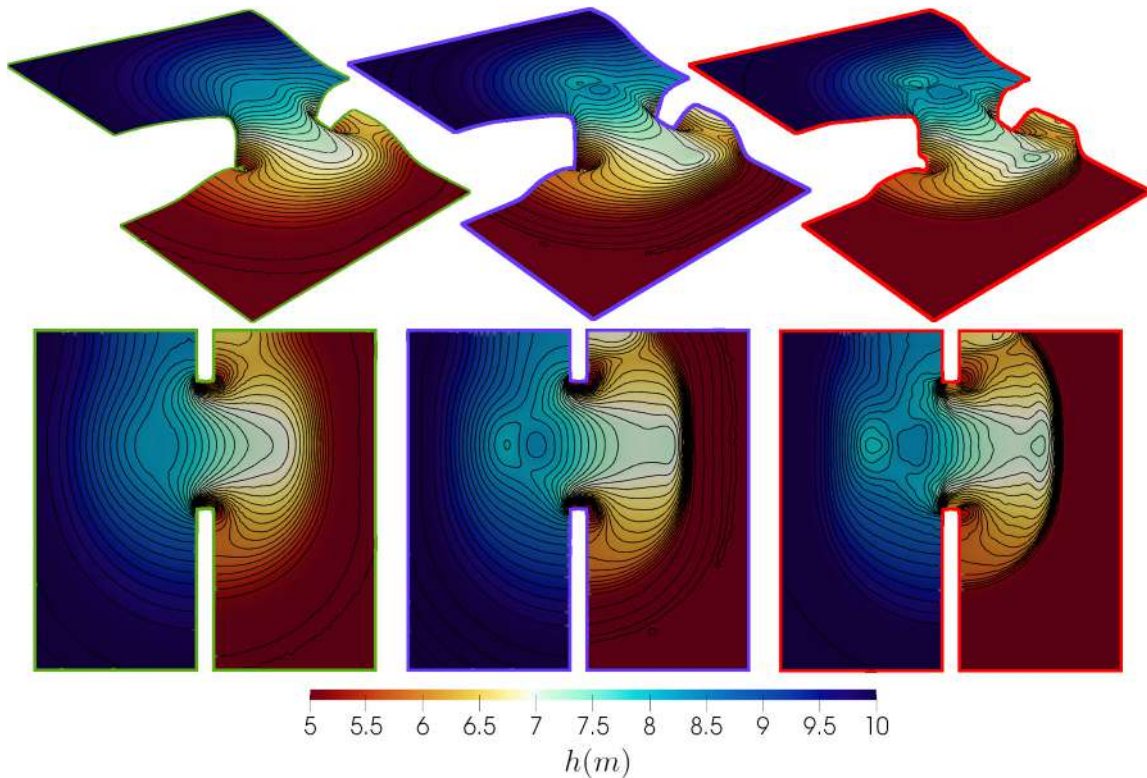


Figure 5.7: 3D and map views of the water surface with height contours obtained at $t = 7.2s$ of the asymmetric dam break problem. The ones with the **green** and **blue** borders were computed with the stabilized approach using, respectively, the CAU and $YZ\beta$ shock-capturing operators. Those with the **red** borders are outcomes of the FCT scheme.

produced a smoother profile, and the FCT scheme created the sharpest. In the first section (Figure 5.8a), which roughly crosses the center of the depression formed due to the interaction between the two corner rarefaction waves, we can see that the FCT solution presented a smoother profile than the reference result, having less pronounced kinks. In the second section (Figure 5.8b), our simulated front fell a little behind in relation to the reference solution.

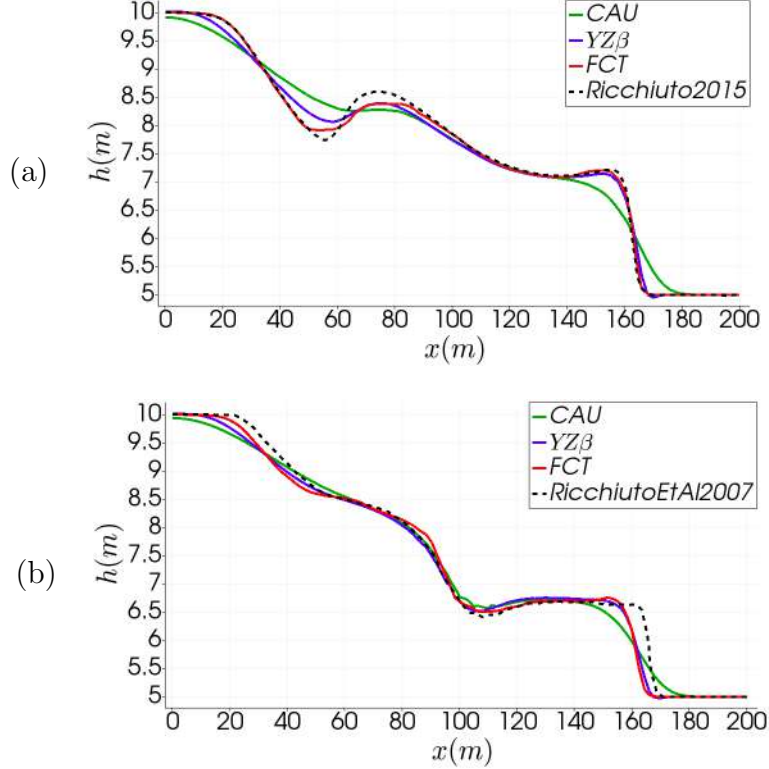


Figure 5.8: Computed and reference solutions for the asymmetric dam break problem plotted along two sections at $t = 7.2$ s. In (a), the reference is the result of RICCHIUTO [129] at $y = 132$ m. In (b), the reference is the solution of RICCHIUTO *et al.* [128] at $y = 160$ m.

In Figure 5.9, we show the computed specific discharge vectors inside a small area near the channel at $t = 7.2$ s. Again we observe that the CAU solution is the smoothest one, as can be seen by the mild color gradient in the right-most reservoir. Plus, we notice that the solution of the $YZ\beta$ operator produced higher discharge values near the channel walls. Throughout the simulations, the stabilized methods' volume errors remained under 10^{-13} , while the FCT's volume error stayed under 10^{-9} . All in all, we remark that the FCT technique produced better results as it did not create spurious perturbations in regions where the shock wave still had not arrived, and the obtained height profiles are closer to the reference solutions.

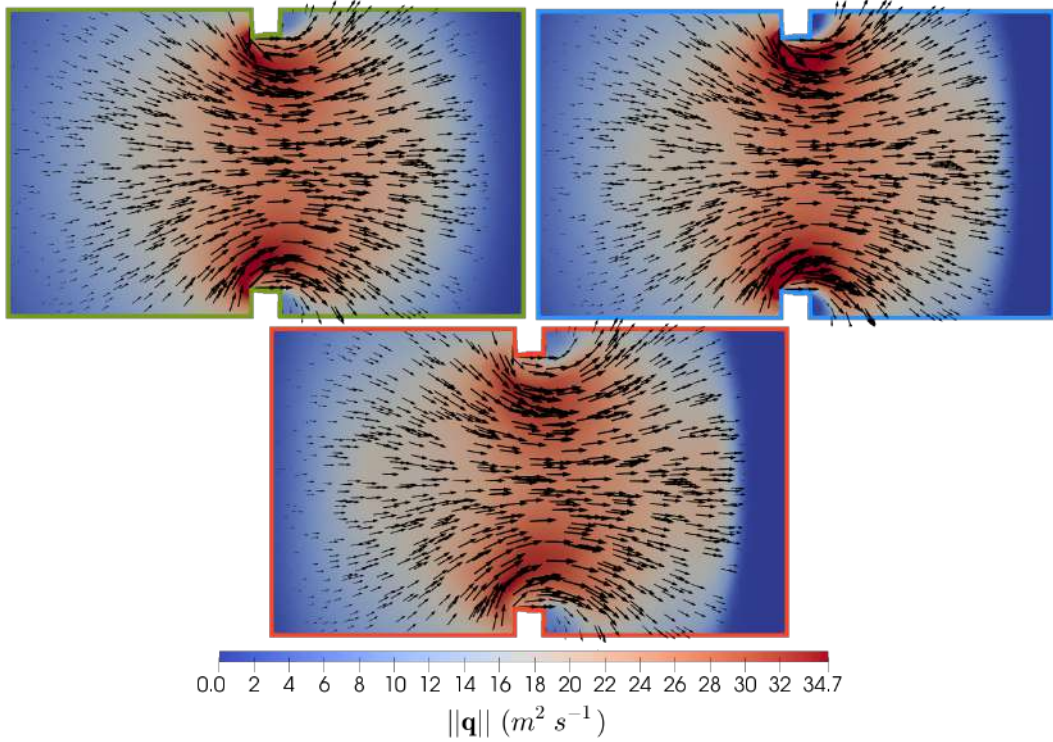


Figure 5.9: Detail of the specific discharge vectors obtained near the channel at $t = 7.2$ s of the asymmetric dam break problem. The maps with the **green** and **blue** borders were computed using the CAU and $YZ\beta$ operators while the one with the **red** border was generated by the FCT method.

5.5 Dam Break over a Channel with Bumps

We simulate the flow produced by a dam break over a dry bed that presents three bumps. This problem was introduced by KAWAHARA and UMETSU [130], being later revisited by BRUFAU and GARCÍA-NAVARRO [93], and LIANG and BORTHWICK [98]. The initial dam encloses a reservoir 16 m long that contains water 1.875 m deep. Here, the $75 \text{ m} \times 30 \text{ m}$ domain is discretized with 1 m^2 elements. Also, non-penetration boundary conditions are enforced during the simulations carried out until $t = 300$ s. As employed by LIANG and BORTHWICK [98], the bed elevation is defined by:

$$z_b(x, y) = \max \left[0, z_1(x, y), z_2(x, y), z_3(x, y) \right], \quad (5.12)$$

with:

$$z_1(x, y) = 1 - \frac{1}{8} \sqrt{(x - 30)^2 + (y - 6)^2}, \quad (5.13)$$

$$z_2(x, y) = 1 - \frac{1}{8} \sqrt{(x - 30)^2 + (y - 24)^2}, \quad (5.14)$$

$$z_3(x, y) = 3 - \frac{3}{10} \sqrt{(x - 47.5)^2 + (y - 15)^2}. \quad (5.15)$$

A 3D view of the problem's initial state can be seen in Figure 5.10a. Figure 5.10b depicts the simulation mesh and the initial water surface elevation. In this example, we use the reference values $(h)_{\text{Ref}} = 1.875 \text{ m}$ and $(q_x)_{\text{Ref}} = (q_y)_{\text{Ref}} = 10^{10} \text{ m}^2 \text{ s}^{-1}$ to compute the $YZ\beta$ operator.

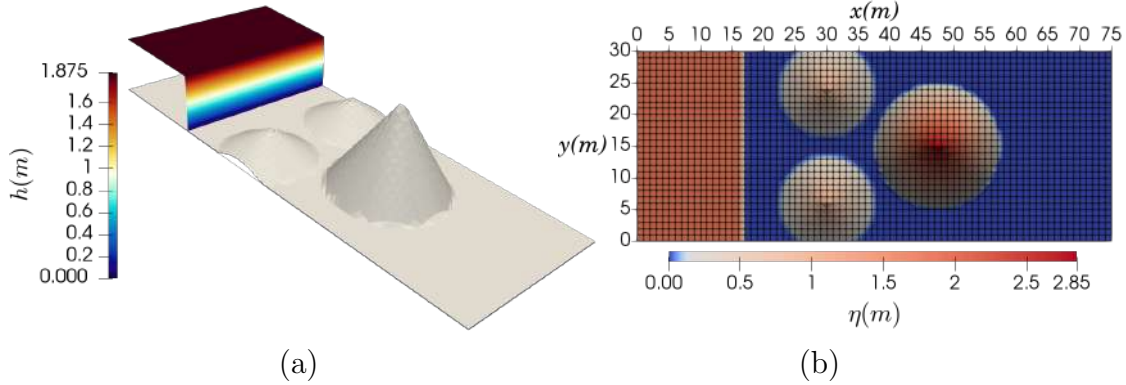


Figure 5.10: 3D view of the initial state (a) and 2D map of the simulation mesh (b) of the dam break over a channel with bumps problem.

Results obtained at key simulation times are presented in Figure 5.11, while Figure 5.12 shows free surface profiles computed along a section at $y = 15 \text{ m}$ and $t = 300 \text{ s}$. After the dam release, the flooding front advances, covering the smallest bumps and generating reflection waves in the upstream direction. At $t = 12 \text{ s}$, as the fluid that passed between the bumps slows down climbing the highest hill, it flows sideways, skirting the largest mound and moving downstream along the sides of the closed domain. Meanwhile, the reflection of the leading flood wave by the highest hill produces another upstream directed wave. By $t = 30 \text{ s}$, the reflection of the first wave on the right wall is climbing the largest bump from its right side. Then, at $t = 300 \text{ s}$, after a series of reflections and dry/wet transitions, all bumps are left partially submerged, and the water surface remains flat.

In general, the results achieved with the $YZ\beta$ and FCT techniques are in better agreement with the works of LIANG and BORTHWICK [98] and GUERMOND *et al.* [87]. At $t = 12 \text{ s}$, among the tested approaches, the FCT scheme produced the sharpest water surface, as can be seen by the greater maximum fluid height and the more detailed fringes that spread downstream along the top and bottom walls. At $t = 30 \text{ s}$, we observe that the FCT scheme presented some ripples to the left of the mounds, while the other methods' surfaces are flat. By the end of the simulation, the solution of the CAU technique has fluid in all the domain, and the $YZ\beta$ formulation has made some fluid go up the highest bump, creating negative fluid heights at a few points. In this case, the FCT method best represented the dry and wet regions and their transitions. This can be seen in Figure 5.11c by the water height distribution near and at the bumps' regions, and the water surface profile in Figure 5.12. In terms of the volume error, the CAU and FCT approaches presented

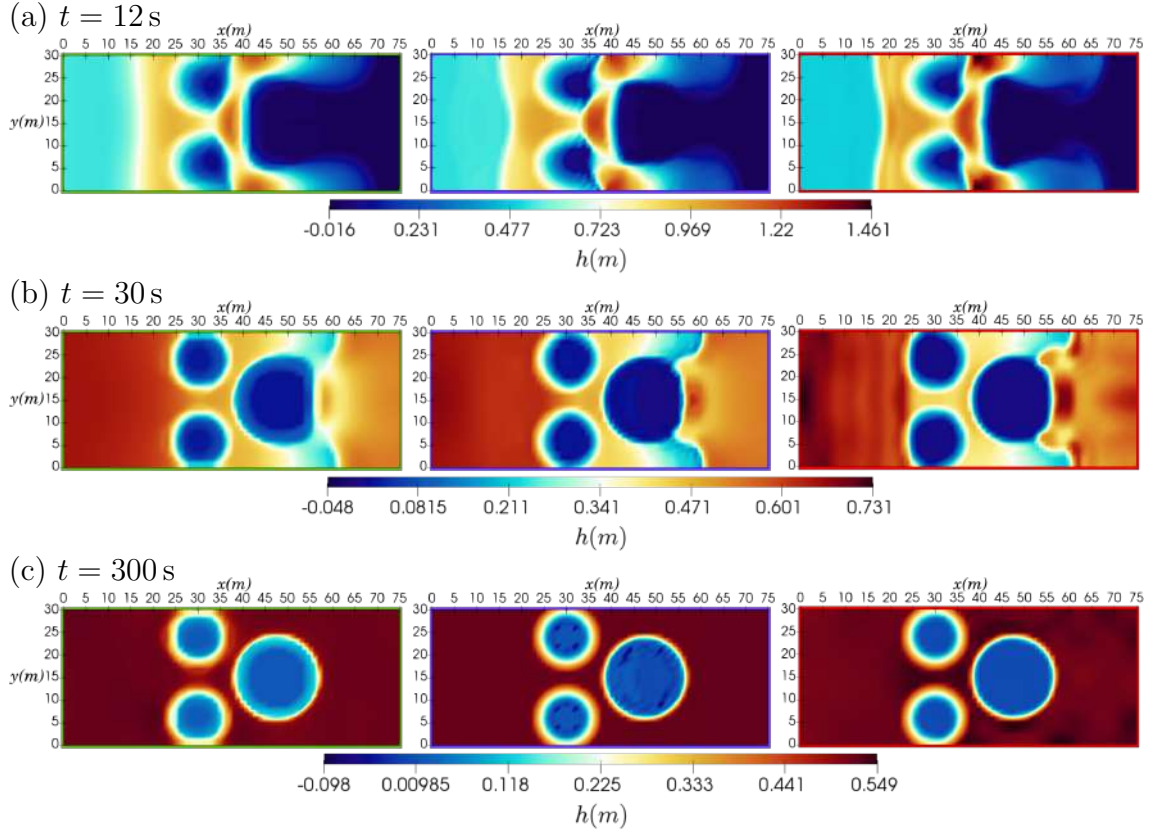


Figure 5.11: Map views of the obtained water heights at key simulation times of the dam break over a channel with bumps problem. The green and blue borders indicate results computed with the stabilized approach using, respectively, the CAU and $YZ\beta$ shock-capturing operators. The maps with red borders are outcomes of the FCT scheme.

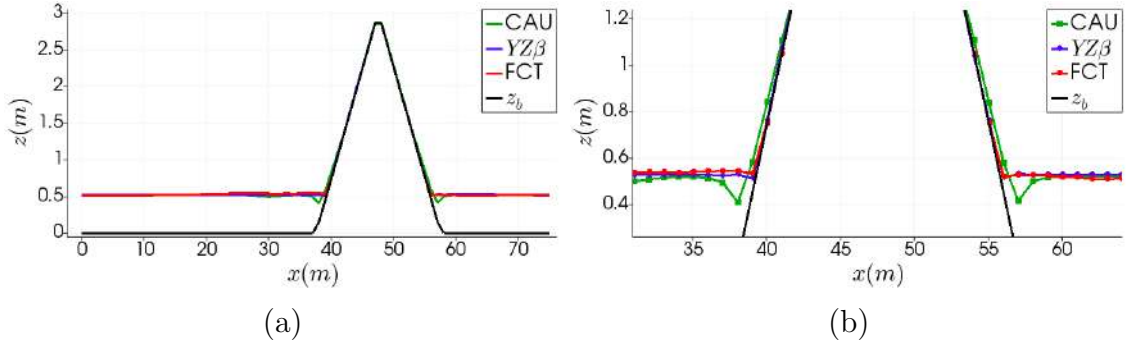


Figure 5.12: Water surface elevation along a section at $y = 15$ m and $t = 300$ s (a) and detail of the free surface near the highest bump (b) of the dam break over a channel with bumps problem.

similar errors of the order of 10^{-10} , while the $YZ\beta$ technique showed an error of the order of 10^{-2} . These facts support the understanding that our approach can properly handle complex wetting and drying processes. In this sense, we present, in Figure 5.13, 3D and map views of the FCT solution at the same times illustrated by LIANG and BORTHWICK [98], showing that our results are in a consonance with theirs. Furthermore, for the next problems studied, we only employ the proposed FCT formulation.

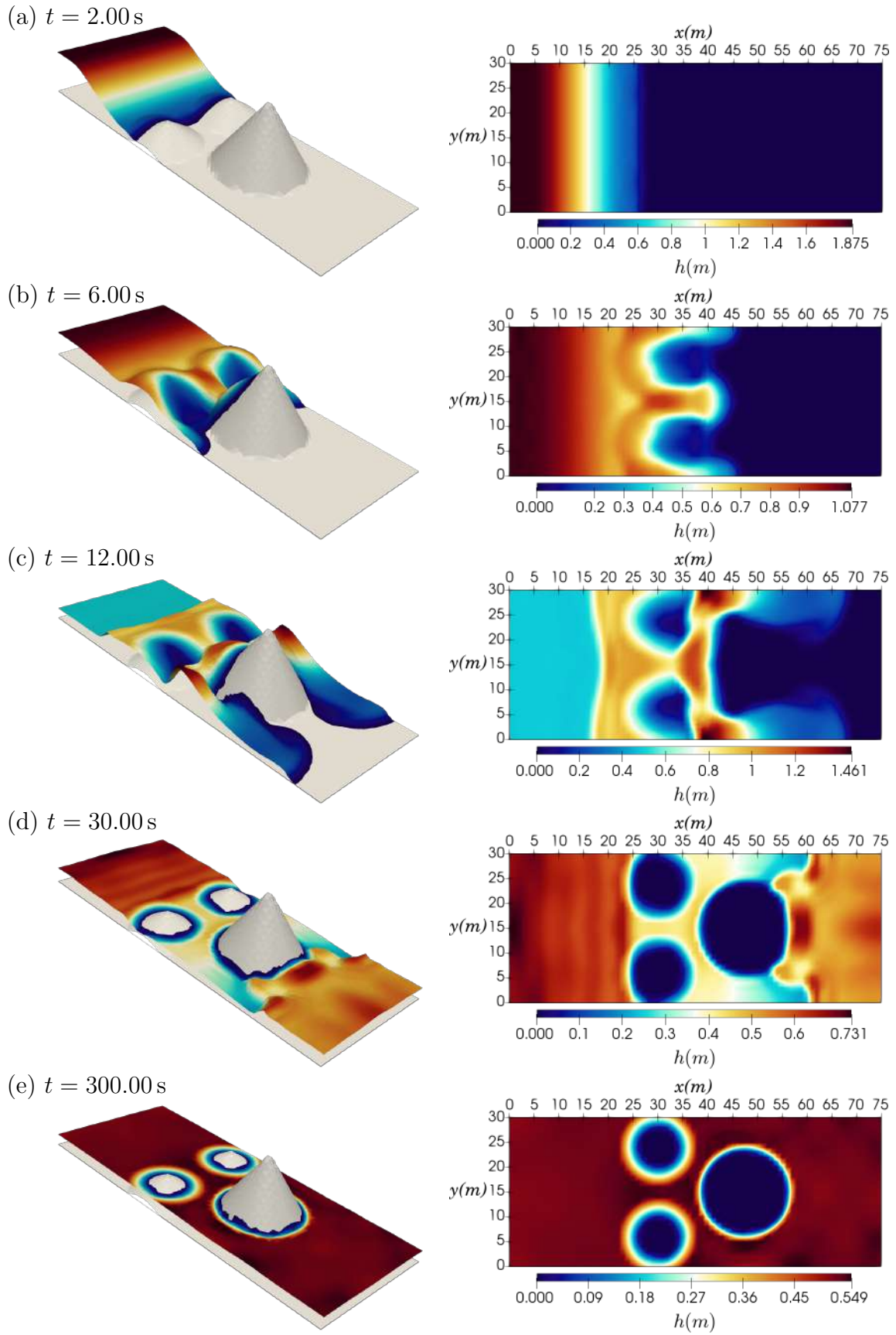


Figure 5.13: 3D and map views at the same times illustrated by LIANG and BORTHWICK [98] of the FCT solution for the dam break over a channel with bumps problem.

5.6 Inflow from a Sloped Channel

We simulate the inflow of water from a slanted channel into a dry reservoir. This problem is based on a case studied by CAMATA *et al.* [131] in which morphodynamic interactions were the focus. However, here the objective is to evaluate the technique proposed to build and evolve streaklines during flow simulations.

Diagrams of the studied region's geometry can be seen in Figure 5.14. We discretize the domain using two meshes: one with 4788 regular elements and another one built by a refinement step in which every element of the first mesh is broken into four smaller cells. Also, we consider that water enters the channel from its left side with $h = 0.5$ m and $\mathbf{u}^T = [10^{-3}, 0]$ m s⁻¹. The rest of the boundary is closed, being subjected to non-penetration conditions. Additionally, 10 streakline sources are positioned along the right faces of the elements at the start of the channel, as depicted in Figure 5.15.

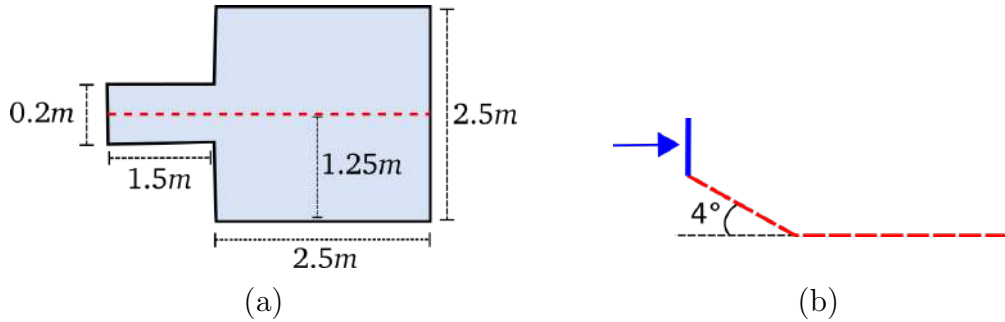


Figure 5.14: Geometric representation of the domain studied in the inflow from a sloped channel problem. (a) Domain's shape and lengths; (b) Detail of a vertical slice over the red dashed line in (a), showing the channel slope and the water inflow direction.

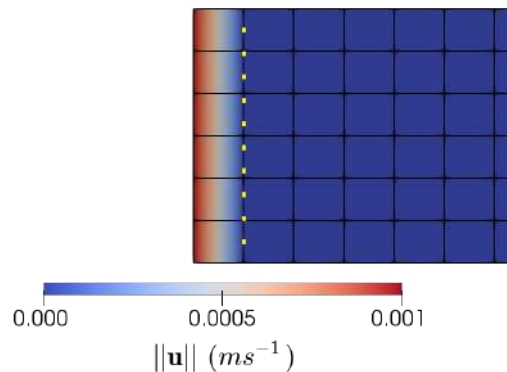


Figure 5.15: Detail of the position of the streakline sources (dots in yellow) at the start of the channel of the inflow from a sloped channel problem.

We present the solution of the problem and the state of the evolved streaklines at $t = 1.8$ s in Figure 5.16. Also, we zoom into the end of the channel to show the distribution of points along the streaklines. Observe that more points are used in regions with higher curvature to the detriment of sections where the lines are straight. This allows to focus computational efforts in regions where the line (or the flow) direction is rapidly changing.

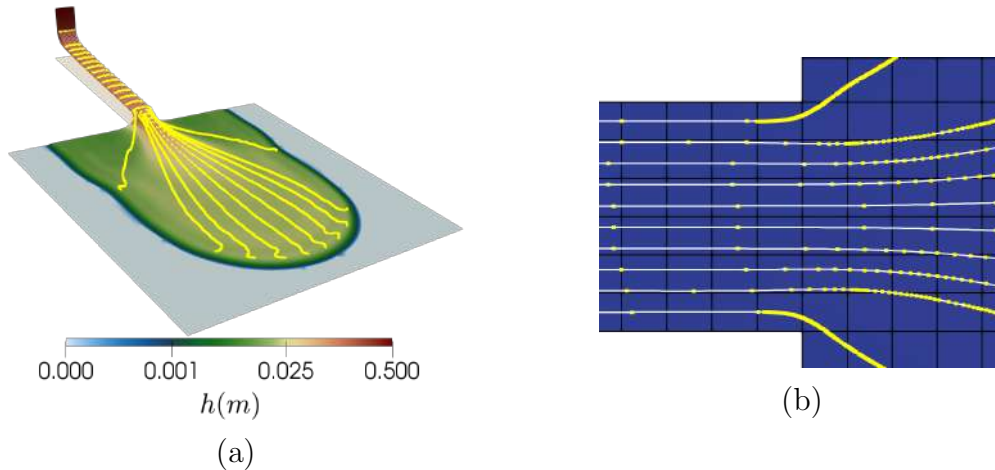


Figure 5.16: Solution at $t = 1.8$ s of the inflow from a channel problem, showing the streaklines in white and their individual points in **yellow**. (a) 3D view of the bed and water surfaces, and the streaklines; (b) Detail of the distribution of points along the streaklines near the end of the channel.

As a benchmark, we use the streaklines computed by the ParaView [119] application. In the finest mesh, the lines built with both approaches are in good consonance. However, when we use the coarsest mesh, the results differ. Our technique seems to produce smoother lines as a consequence of the adaptive procedure used to add and remove points throughout the simulation. Also, as we use splines to determine new point locations at high curvature regions, these are potential sources of discrepancy between the solutions. Figure 5.17 presents the streaklines computed by both approaches at $t = 6.19$ s. Finally, we observe that the present approach is able to produce complex line geometries, like the ones depicted in Figure 5.18.

5.7 Turbidity Current Flow over an Irregular Bed

We simulate the continuous flow of a turbidity current over an irregular topography. We focus on the turbidite system originated from the mouth of the Almirante Câmara canyon, located in the Campos Basin in Brazil. This deep-water marine sedimentary environment is part of the Carapebus formation. It is known that the turbidite beds in this region are the result of several events over long geological periods [112]. However, here we only simulate a single continuous inflow to evaluate the

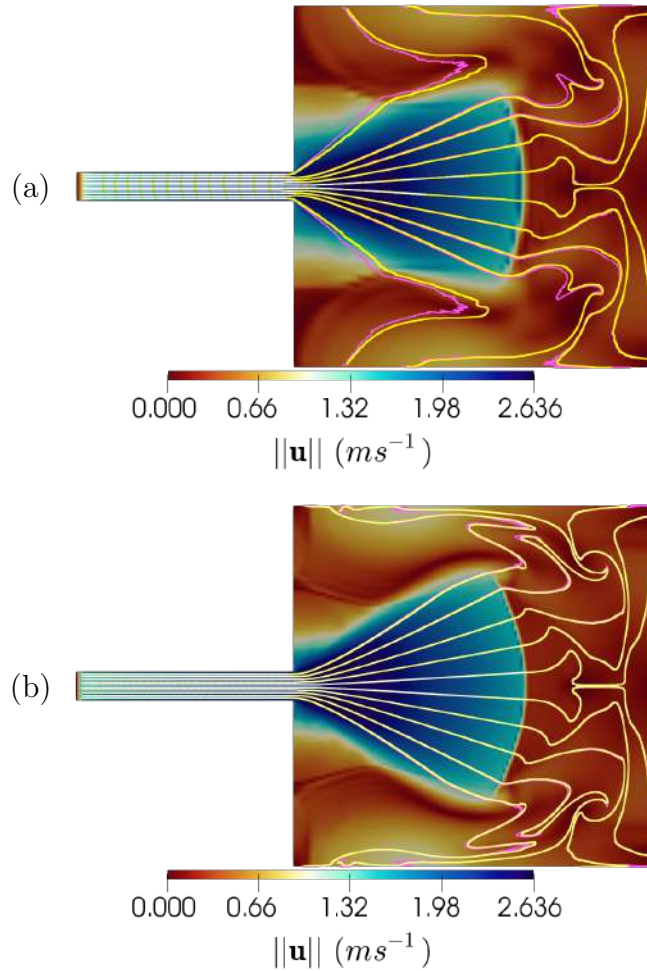


Figure 5.17: States of the streaklines computed by the ParaView [119] application (lines in pink) and our approach (lines in white and points in yellow) at $t = 6.19$ s of the inflow from a sloped channel problem. Computations were performed using two meshes: one with low-resolution (a) and another one with higher resolution (b).

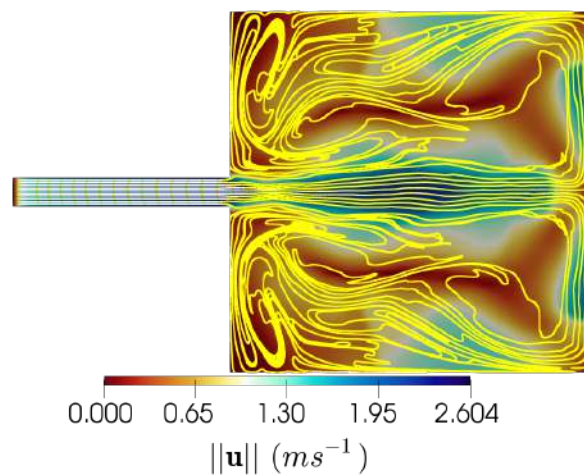


Figure 5.18: State of the evolved streaklines at $t = 13$ s of the inflow from a sloped channel problem.

evolution of the current and the related streaklines, as well as the overall stability of our approach. Also, as paleosurfaces of this region are unavailable, we perform the simulation over the modern seabed. This might be a valid approximation since the system is considered recent from a geological perspective.

The examined region covers an area of about $141 \text{ km} \times 107 \text{ km}$, subdivided by quadrilateral elements with the average characteristic length of 2 km . A 3D view of the seabed and the entry of the Almirante Câmara canyon can be seen in Figure 5.19a. In our simulation, we consider the current to have density $\rho = 1100 \text{ kg m}^{-3}$. Also, we prescribe an inflow with $h = 5 \text{ m}$, $q_x = 37.037 \text{ m}^2 \text{ s}^{-1}$ and $q_y = 0 \text{ m}^2 \text{ s}^{-1}$ at the start of the channel. Figure 5.19b shows a detailed view of the mesh near the canyon's entry and the inflow discharges. Here, we apply non-penetration constraints to the portion of the coast that does not receive the current's inflow and leave the remaining boundary open. In addition, 5 streakline sources are placed near the canyon's entry.

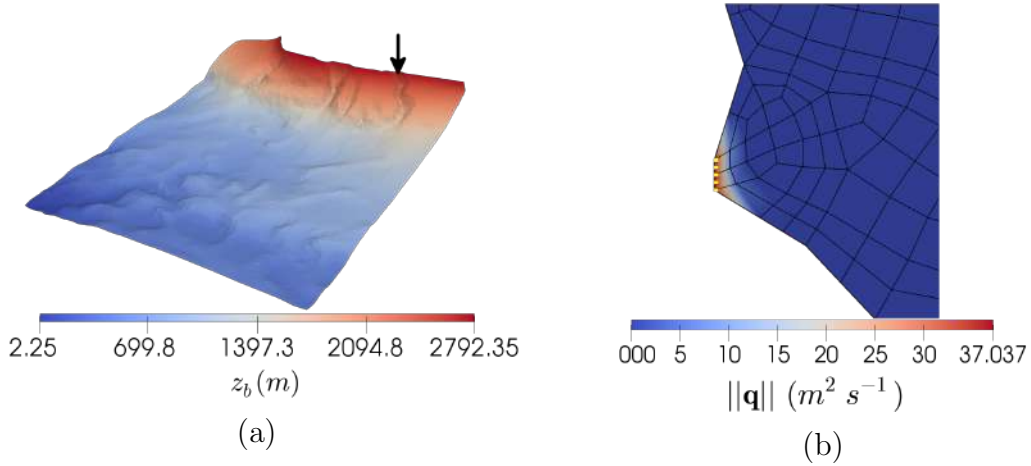


Figure 5.19: Initial configuration of the turbidity current flow over an irregular bed problem. (a) 3D view of the seabed indicating the entry of the Almirante Câmara canyon. (b) Detailed view of the mesh and the inflow discharges near the boundary by the canyon's entry, and the streakline sources (dots in **yellow**).

Obtained results at key simulation times are presented in Figure 5.20. In the initial steps of the simulation, the flow is channeled into the canyon, and the streaklines merge into one. After leaving the valley, the current advances with little variations in width until it forms a lobe before being forced back into a channel. As it traverses the trough, it finds an accumulation point in a turn. Then, following the channel, it leaves the area of interest. We observe that the path followed by the simulated current is in a good agreement with the real Almirante Câmara turbidite system studied by MACHADO *et al.* [112]. Also, the positions of the lobe and the accumulation point resemble the real system features.

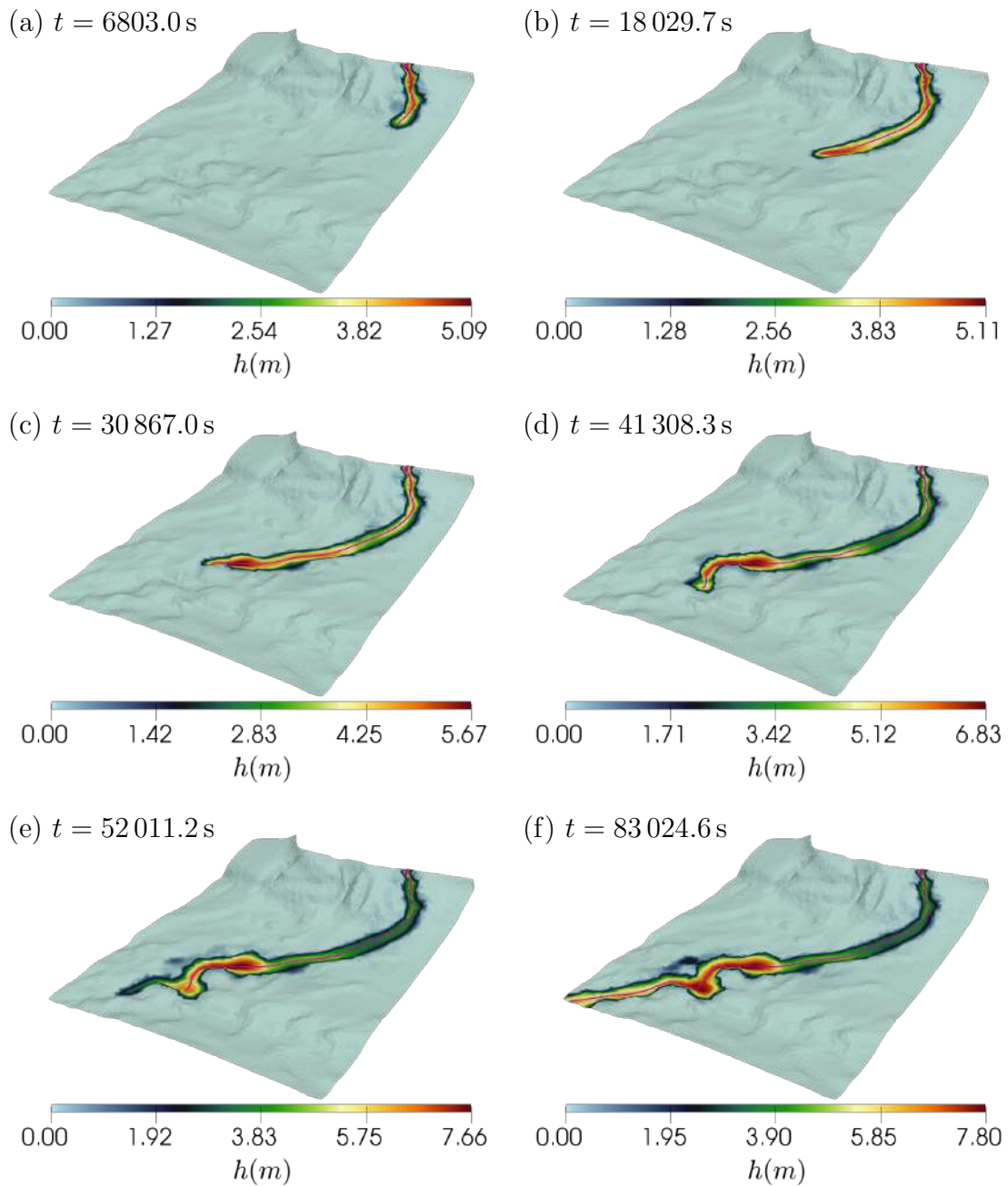


Figure 5.20: Simulated turbidity current at key times of the turbidity current flow over an irregular bed problem. The pink lines in middle of the current portray the computed streaklines.

5.8 Sedimentation on a Flat Bed

In this section, we simulate a basic scenario, trying to reproduce stacking patterns similar to the one illustrated in Figure 3.11, where we see a progradational trend between events and a retrogradational-like arrangement in an event's deposits. Furthermore, we evaluate the use of the decay factor f_s , introduced in Equation 3.21 to reduce the accommodation space when we radially scatter sediments from a cell crossed by a streakline. Plus, we assess the impact of using sub-steps with yet more limited depositional spaces.

We consider a $10 \text{ km} \times 10 \text{ km}$ area subdivided into 20×20 cells, and position the sediment supply (and the turbidity current inflow) in the middle of the left-most boundary, as depicted in Figure 5.21. For the present example, we simulate three events that have the same inflow current height of $h = 4 \text{ m}$. In each case, the associated sediment supply provides the same proportion of the different sediment types: 33.3% silt, 33.3% medium sand and 33.4% coarse sand. However, we vary the total amount of sediments supplied to each event, as well as the event's volume and duration. The first flow event holds 0.2 Mm^3 , having an inflow duration of 5 min. Then, the second and thirds events have, respectively, two and four times more volume and duration than the first one. The amount of sediments available to the three events are, respectively, 16.98 Mm^3 , 49.8 Mm^3 and 84.9 Mm^3 .

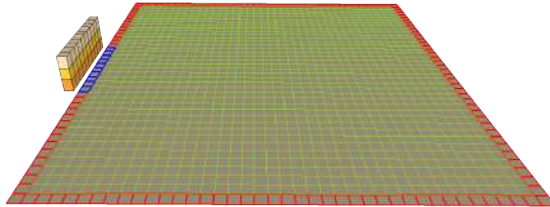


Figure 5.21: Location of the sediment supply used in the sedimentation on a flat bed problem. Here, the columns beside the boundary cells highlighted in **blue** represent the sediment supplies.

At first, we perform a simulation using one sub-step per event and setting $f_s = 1$ to disable the factor's effect on the available accumulation space. We show maps of the top of the resulting deposits after each event simulation in Figure 5.22. Be aware that, if examined alone, these maps can be misleading in the sense that they only show the strata at the top of the bed, however thin they may be. Thus, to aid the result analysis, we show, in Figure 5.25, the deposited lithologies along a vertical section that is parallel to the x-direction and crosses the middle of the domain.

Next, we run a simulation using one sub-step per event and the standard value of the spreading factor $f_s = 2$. Map views of the produced deposits are shown in Figure 5.23, while we present, in Figure 5.26, the obtained stacking pattern along the same section of the previous case. Map views of the deposits' thickness after each event simulation can be seen in Figure 5.27. We observe that the use of the factor

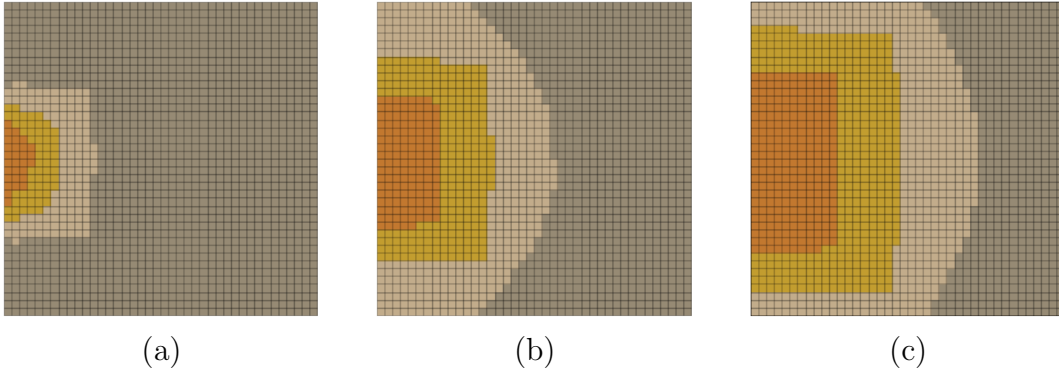


Figure 5.22: Map views of the top of the resulting deposits after each event of the sedimentation on a flat bed problem, when solved using a single sub-step per event and $f_s = 1$. The maps (a), (b) and (c) refer to the state of the bed after the first, second and last events.

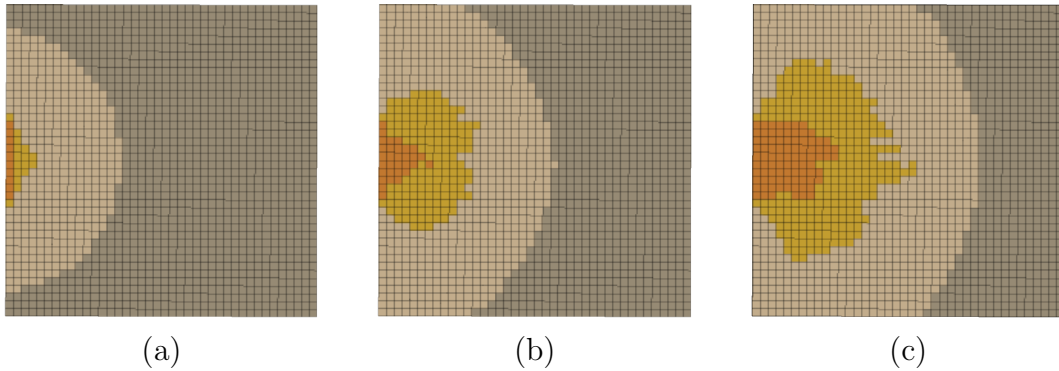


Figure 5.23: Map views of the top of the resulting deposits after each event of the sedimentation on a flat bed problem, when solved using a single sub-step per event. The maps (a), (b) and (c) refer to the state of the bed after the first, second and third events.

f_s produces more natural lateral distributions of sediments, avoiding the deposits of having squared shapes, such as the ones seen in Figure 5.22. Besides, its use yields more realistic stacking patterns, as the transitions between the different lithologies is less abrupt. We can also notice fining-upwards trends in an event's deposits near the interfaces between different lithology types.

Afterward, we simulate the current scenario using three sub-steps per event. Maps of the top of the depositional bed after each sub-step can be seen in Figure 5.24. In addition, we present in Figure 5.28 the stacking pattern obtained at the same vertical section previously examined. Then, the evolution of the thickness of the deposits of each lithology type is illustrated in Figure 5.30.

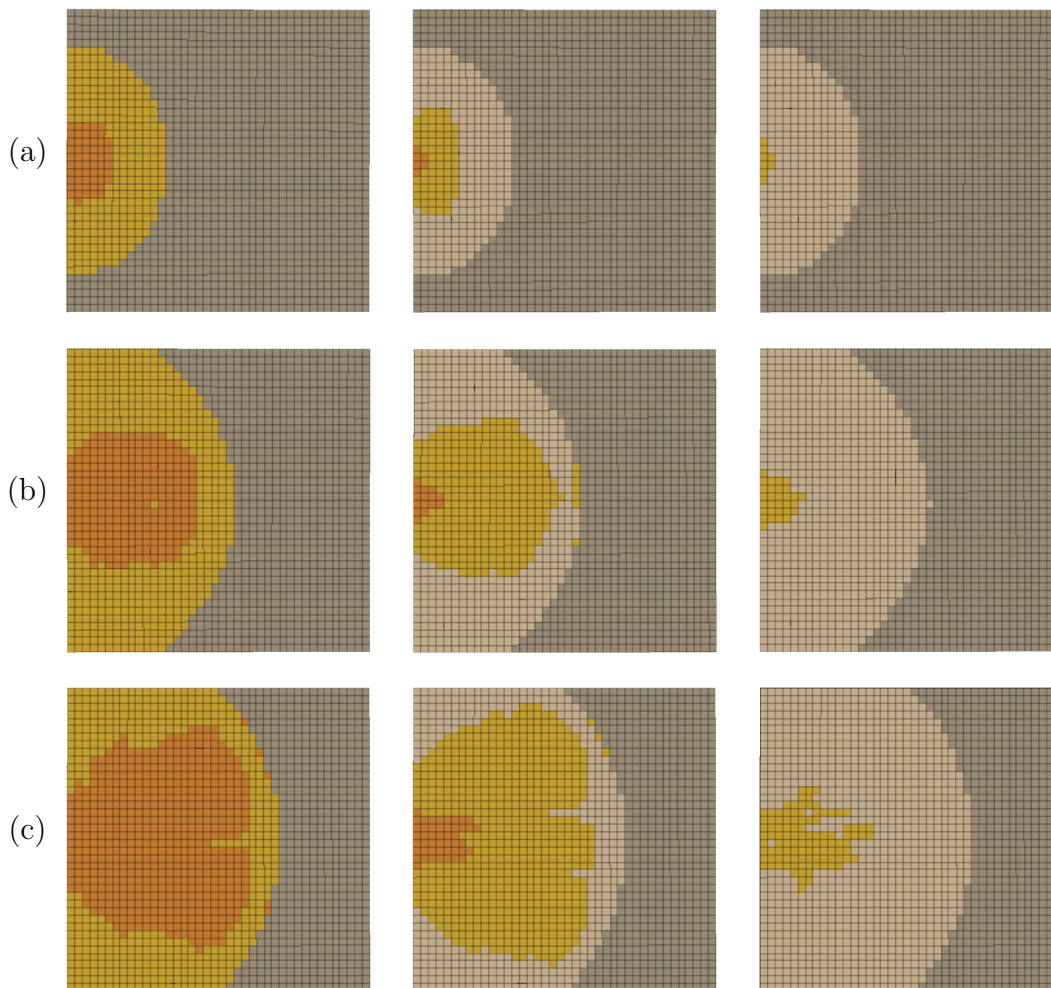
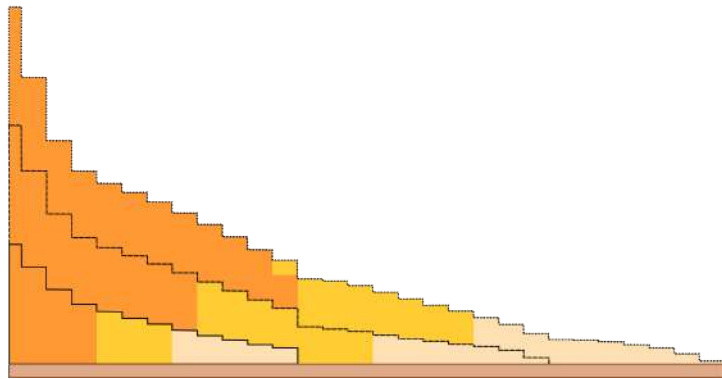
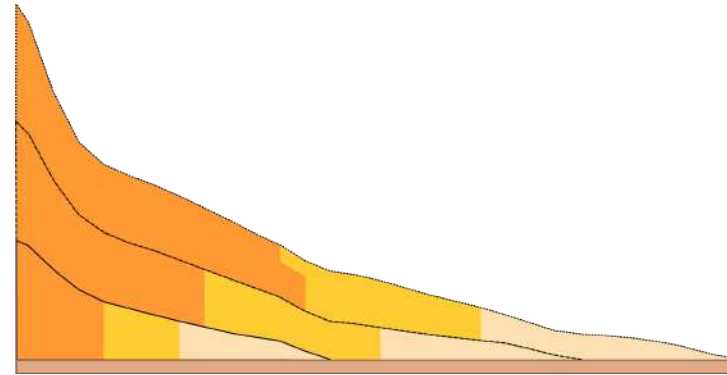


Figure 5.24: Map views of the sedimentary bed after each event and sub-step of the sedimentation on a flat bed problem, when solved using three sub-steps per event. (a), (b) and (c) represent depositions at each event simulation, being (a) associated with the first event and (c) with the last. The progression of the maps from left to right denote the sequence of sub-steps in an event's deposition.



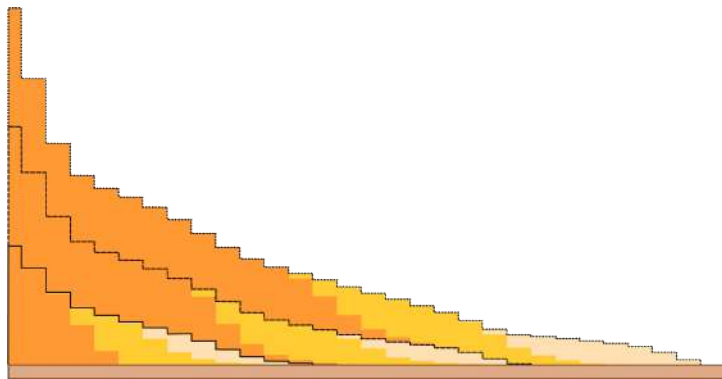
(a)



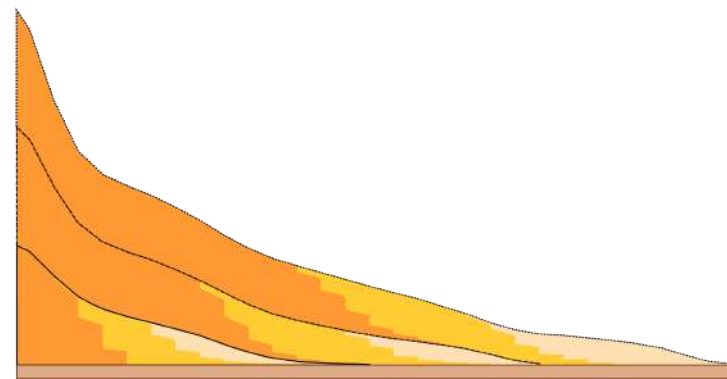
(b)

Figure 5.25: View, along a vertical section parallel to the x-direction and placed in the middle of the simulated region, of the stacking pattern produced at the sedimentation on a flat bed problem, when solved using one sub-step and $f_s = 1$. (a) Original section in terms of the deposited lithology blocks; (b) Smoothed section. Here the black lines separate the strata of each event.

85



(a)



(b)

Figure 5.26: View, along a vertical section parallel to the x-direction and placed in the middle of the simulated region, of the stacking pattern produced at the sedimentation on a flat bed problem, when solved using one sub-step. (a) Original section in terms of the deposited lithology blocks; (b) Smoothed section. Here the black lines separate the strata of each event.

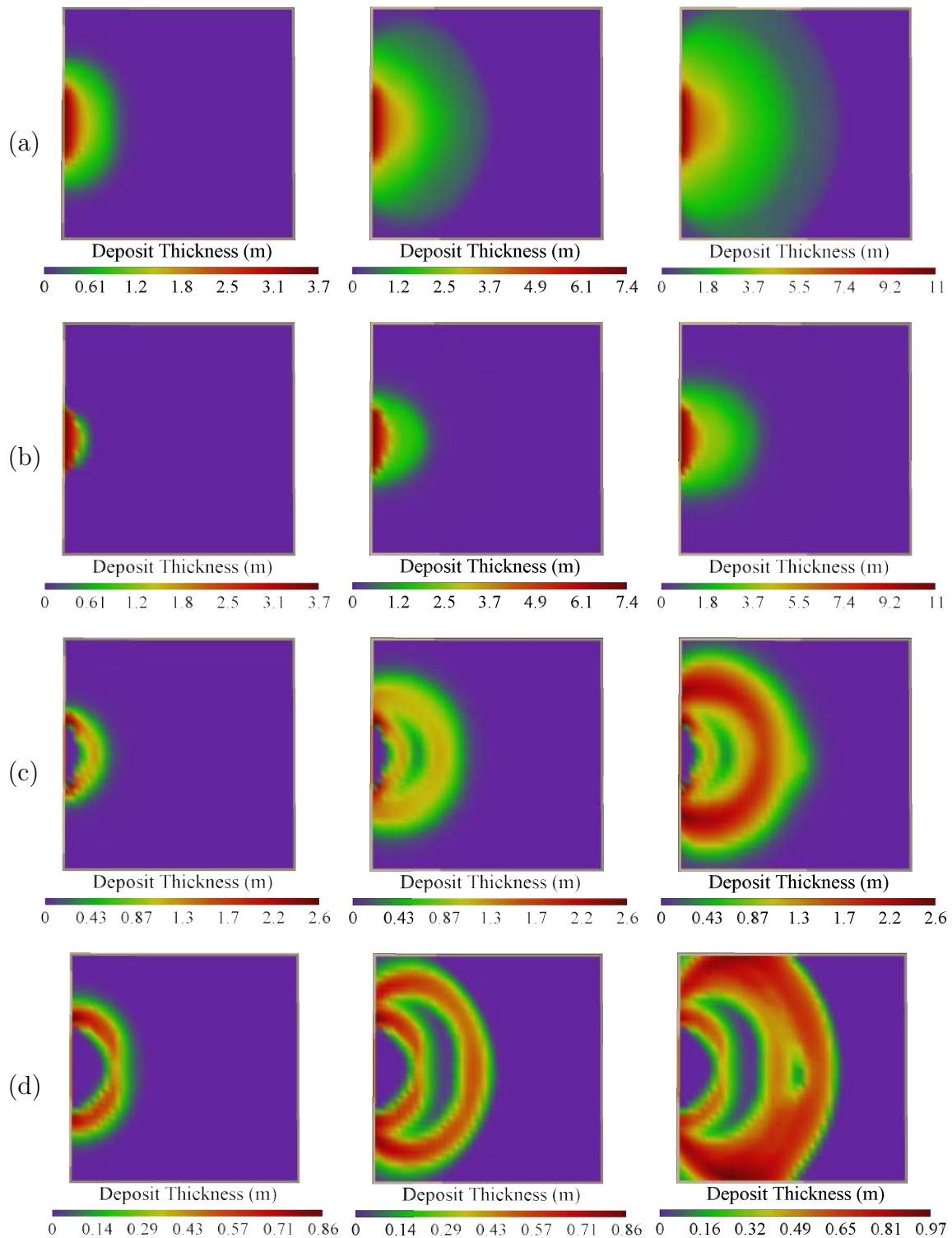
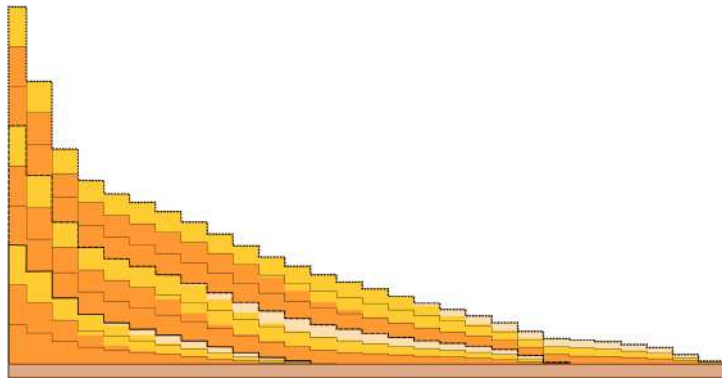
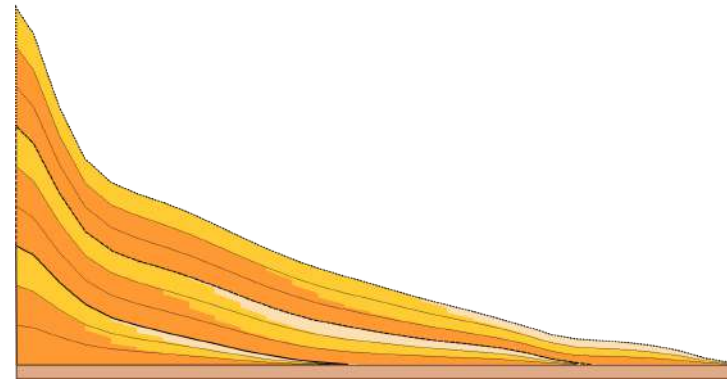


Figure 5.27: Maps of the thickness of the accumulated sediment deposits after each event of the sedimentation on a flat bed problem, when solved using one sub-step per event. The maps on the left depict the result after the first event, while the maps at the middle and the right columns respectively represent the states after the second and third events. Each maps row shows the accumulated thickness of the different components: (a) All lithology types; (b) Coarse sand strata; (c) Medium sand deposits; (d) Silt material.

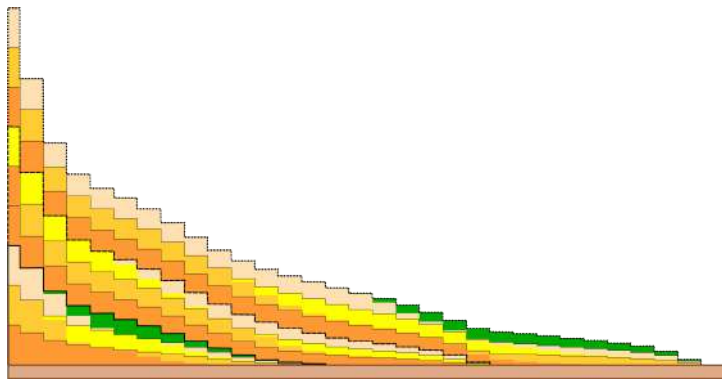


(a)

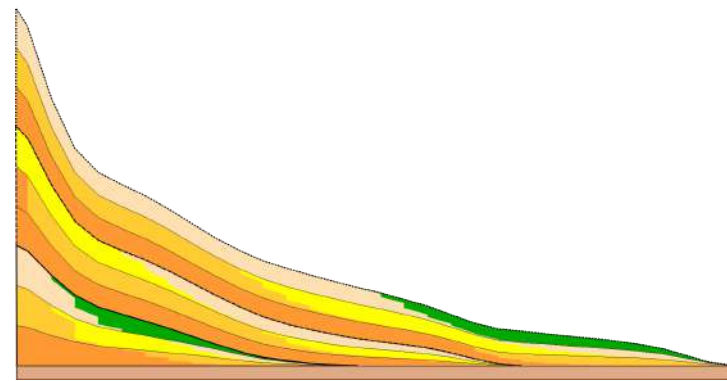


(b)

Figure 5.28: View, along a vertical section parallel to the x-direction and placed in the middle of the simulated region, of the stacking pattern produced at the sedimentation on a flat bed problem, when solved using three sub-steps. (a) Original section in terms of the deposited lithology blocks; (b) Smoothed section. Here the black lines separate the strata of each event.



(a)



(b)

Figure 5.29: View, along a vertical section parallel to the x-direction and placed in the middle of the simulated region, of the stacking pattern produced at the sedimentation on a flat bed problem, when solved using five lithology types and three sub-steps. (a) Original section in terms of the deposited lithology blocks; (b) Smoothed section. Here the black lines separate the strata of each event.

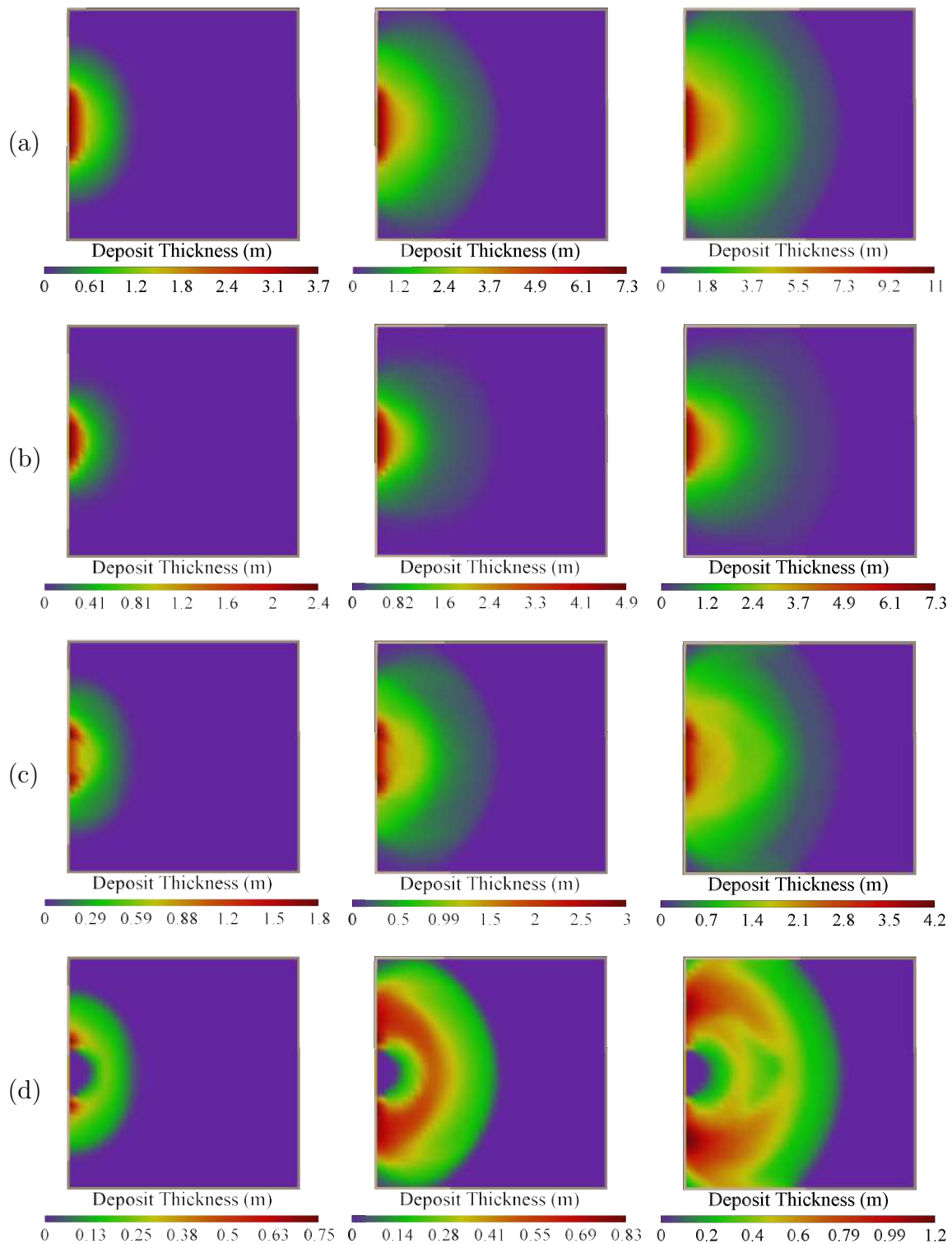


Figure 5.30: Maps of the thickness of the accumulated sediment deposits after each event of the sedimentation on a flat bed problem, when solved using three sub-steps per event. The maps on the left depict the result after the first event, while the maps at the middle and the right columns respectively represent the states after the second and third events. Each maps row shows the accumulated thickness of the different components: (a) All lithology types; (b) Coarse sand strata; (c) Medium sand deposits; (d) Silt material.

From the deposits reproduced by the simulations using one and three sub-steps, we can see that the use of a single step causes the deposition of each lithology type to be given in blocks. Indeed, there are clear lateral transitions between them, as we can distinguish in Figures 5.26 and 5.27. However, in this case, the coarser grain sizes deposit mostly near the inflow region, creating tall strata and filling all the available accommodation space. Also, we observe that the modest retrogradational trend inside each layer is only due to the spreading factor f_s . In contrast, if we use more sub-steps per event, we allow the coarser fractions to progress further downstream, leaving, in the upstream area, more accommodation space available to the finer sediments' deposition. As we can see in Figures 5.28 and 5.30, this engenders more pronounced fining-upwards biases in an event's deposits. Finally, we remark that, regardless of the number of sub-steps employed, we could reproduce stacking patterns similar to the one presented in Figure 3.11, where each event progrades in respect to the previous one.

At last, to help elucidate that our approach can reproduce more complex stacking patterns, we present in Figure 5.29 the layering of different strata along a section, computed considering that the sediment supply provides an equal amount of each of the five granulometric fractions.

5.9 Sedimentation on a Sloping Bed

In this example, we simulate three turbidite events initiated as a result of consecutive sudden sea-level drops in a region where part of the sea bed is sloped. A schematic representation of the simulated area and the initial bed elevation can be seen in Figure 5.31.

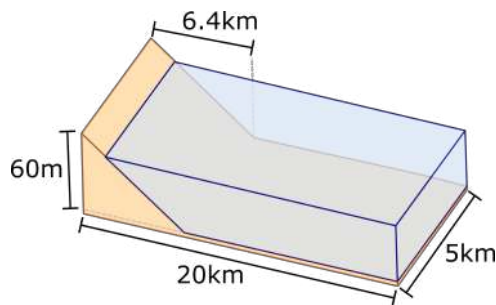


Figure 5.31: Schematic representation of the region simulated in the sedimentation on a sloping bed problem.

At the stratigraphic application, the simulation grid is composed of 50×20 cells. We show the state of the sea level and the position of the sediment supply (or the associated current inflow) used in the simulation of each event in Figure 5.32. For the first event, the sea level is positioned at 45 m from the bed's lowest point.

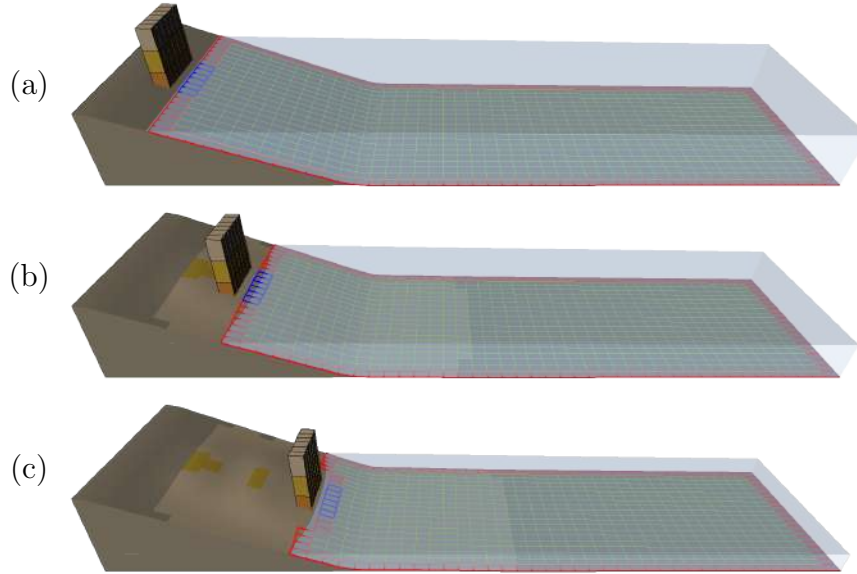


Figure 5.32: Location of the sea level and the sediment supply used to simulate each turbidite event of the sedimentation on a sloping bed problem. The arrangements (a), (b) and (c) were used to simulate, respectively, the first, second and last events. Here, the columns beside the boundary cells highlighted in blue represent the sediment supplies.

Afterward, we decrease it by 15 m to simulate the second event, and reduce it again by the same amount to compute the last one. Here, all events have the same characteristics: current inflow height of $h = 4$ m, available volume of 0.2 Mm^3 and inflow duration of 5 min. Also, at each event, the sediment supply provides a total of 35.5 Mm^3 of which 33.3% is silt, 33.3% is medium sand and 33.4% is coarse sand.

At first, we perform the simulation using one sub-step per event. Map views of the resulting deposits after each event can be seen in Figure 5.33. To allow the examination of the produced stacking patterns, we show, in Figure 5.34, the deposited lithologies along a vertical section that is parallel to the x -direction and positioned in the middle of the simulated area. Next, we present maps of the thickness of the accumulated sediment deposits after each event in Figure 5.35, even showing the contribution of each granulometric fraction.

In sequence, we perform a simulation using three sub-steps per event. We show map views of the accumulated deposits after each event and sub-step in Figure 5.36. As in the previous case, we present, in Figure 5.37, the deposited lithologies along a vertical section parallel to the x -direction and placed in the middle of the simulated area. Then, Figure 5.38 shows maps of the thickness of the accumulated sediment deposits after each event simulation.

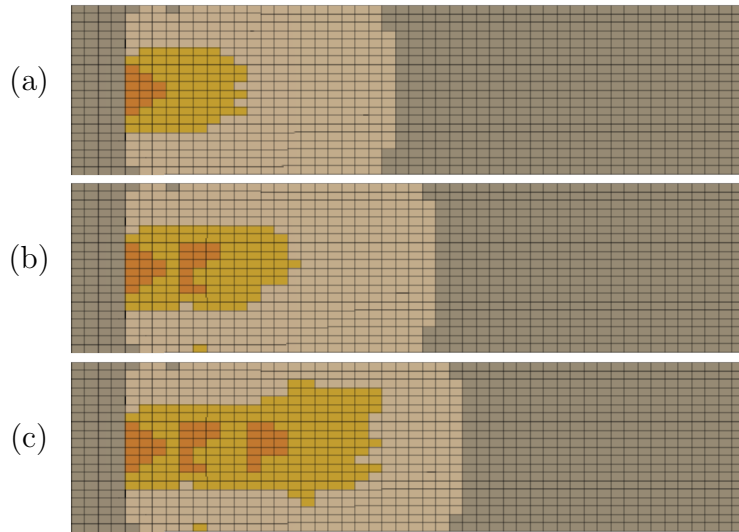


Figure 5.33: Map views of the resulting deposits after each event of the sedimentation on a sloping bed problem, when solved using a single sub-step per event. The maps (a), (b) and (c) refer to the state of the bed after the first, second and last events.

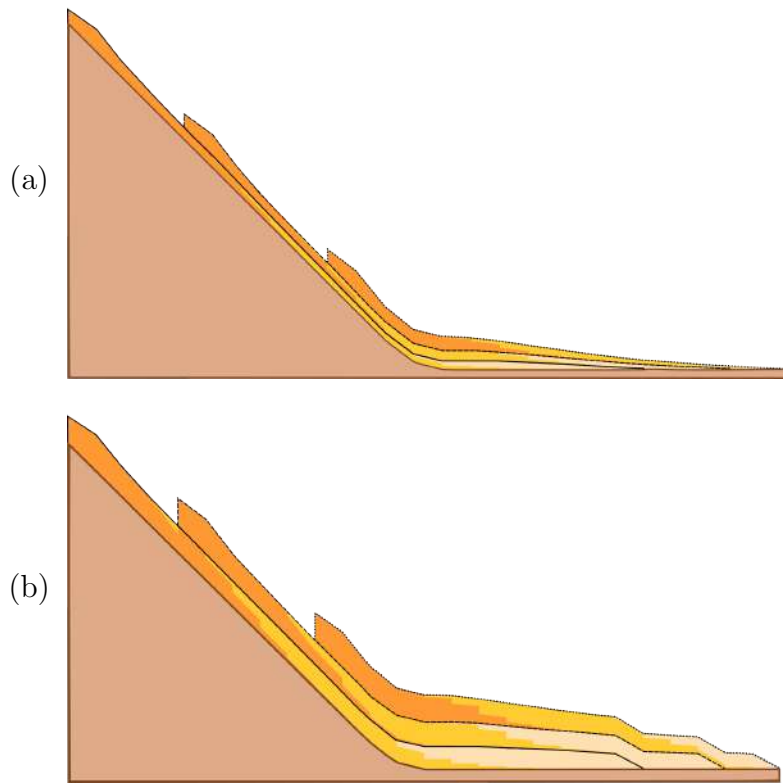


Figure 5.34: View, along a vertical section parallel to the x -direction and placed in the middle of the simulated region, of the stacking pattern produced at the sedimentation on a sloping bed problem, when solved using a single sub-step per event. (a) Original section; (b) Section with scaled deposit thicknesses. Here the different types of black lines separate the strata of each event.

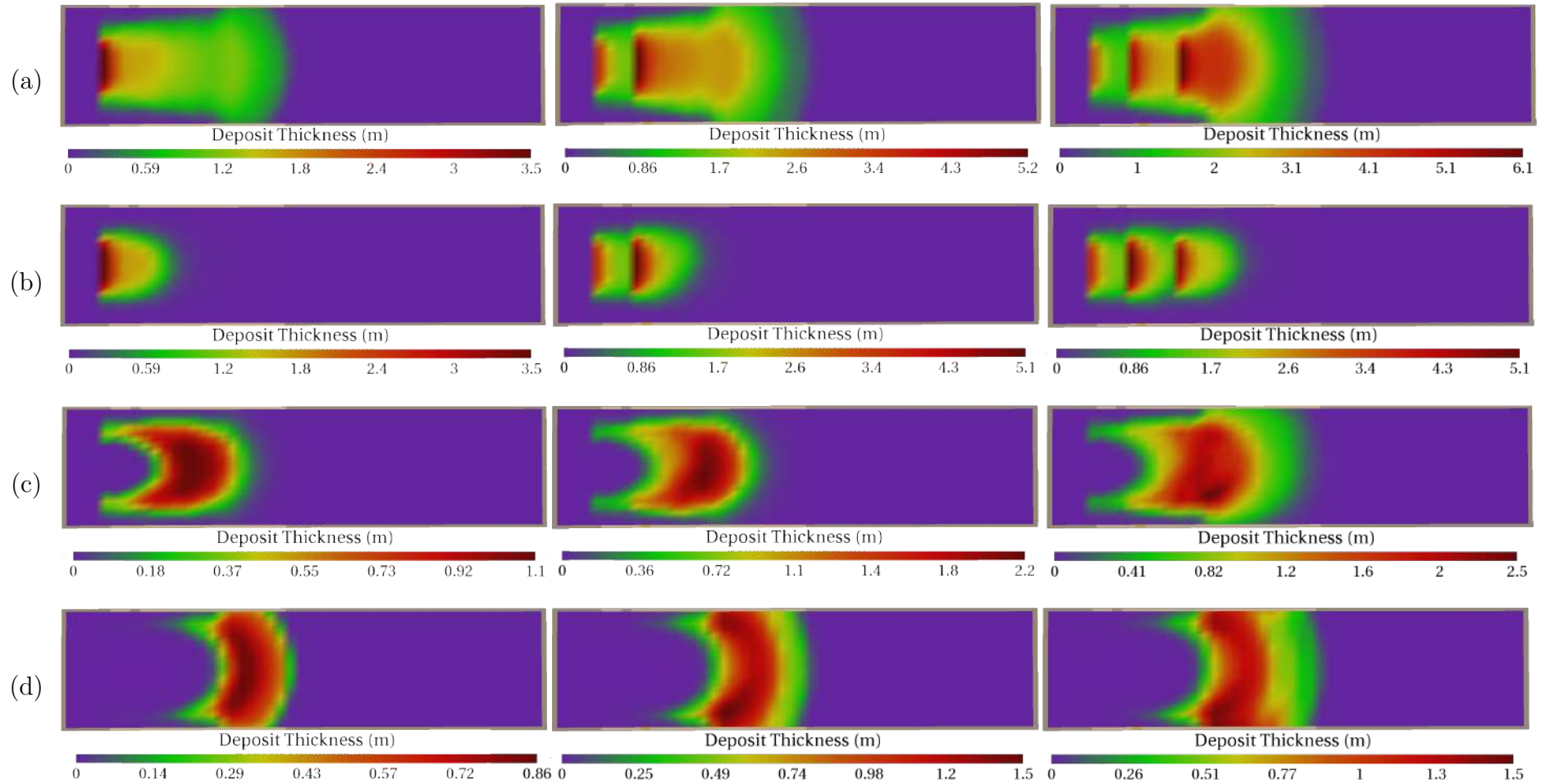


Figure 5.35: Maps of the thickness of the accumulated sediment deposits after each event of the sedimentation on a sloping bed problem, when solved using a single sub-step per event. The maps on the left depict the result after the first event, while the maps at the middle and the right columns respectively represent the states after the second and third events. Each maps row shows the accumulated thickness of the different components: (a) All lithology types; (b) Coarse sand strata; (c) Medium sand deposits; (d) Silt material.

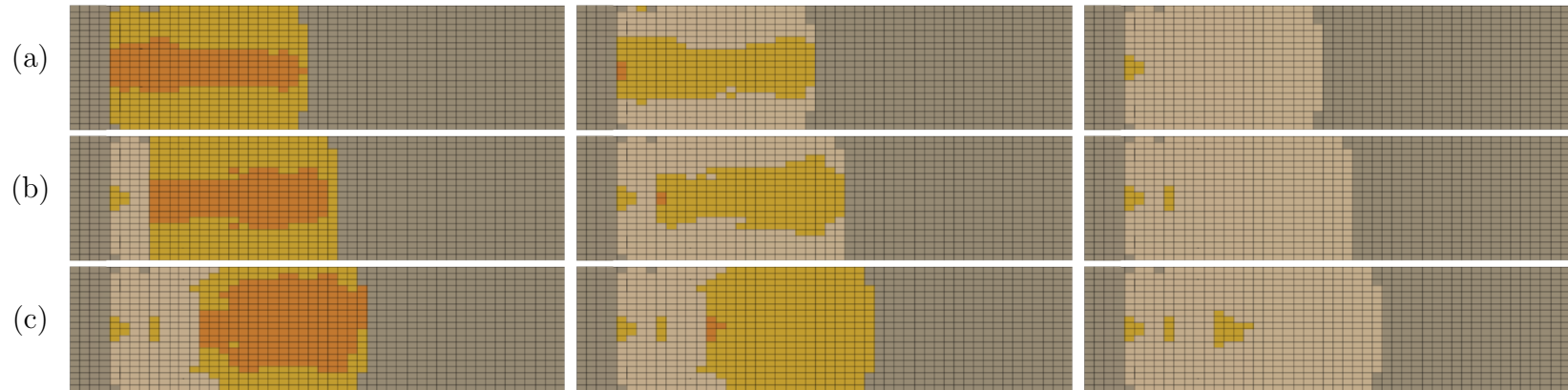


Figure 5.36: Map views of the sedimentary bed after each event and sub-step of the sedimentation on a sloping bed problem, when solved using three sub-steps per event. (a), (b) and (c) represent depositions at each event simulation, being (a) associated with the first event and (c) with the last. The progression of the maps from left to right denote the sequence of sub-steps in an event's deposition.

93

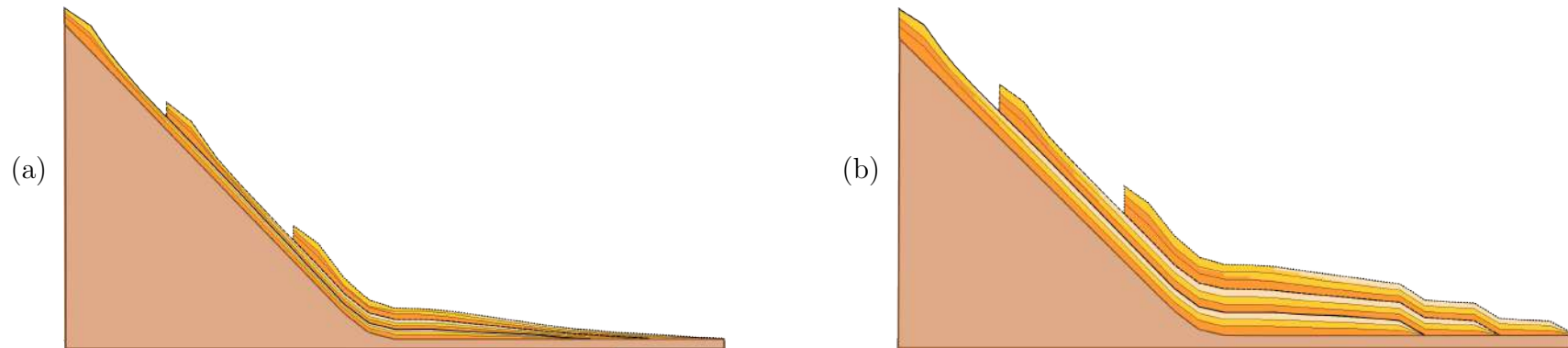


Figure 5.37: View, along a vertical section parallel to the x -direction and placed in the middle of the simulated region, of the stacking pattern produced at the sedimentation on a sloping bed problem, when solved using three sub-steps. (a) Original section; (b) Section with scaled deposit thicknesses. The thinnest black lines represent the limits of a sub-step deposit, while the different types of thicker black lines separate the strata of each event.

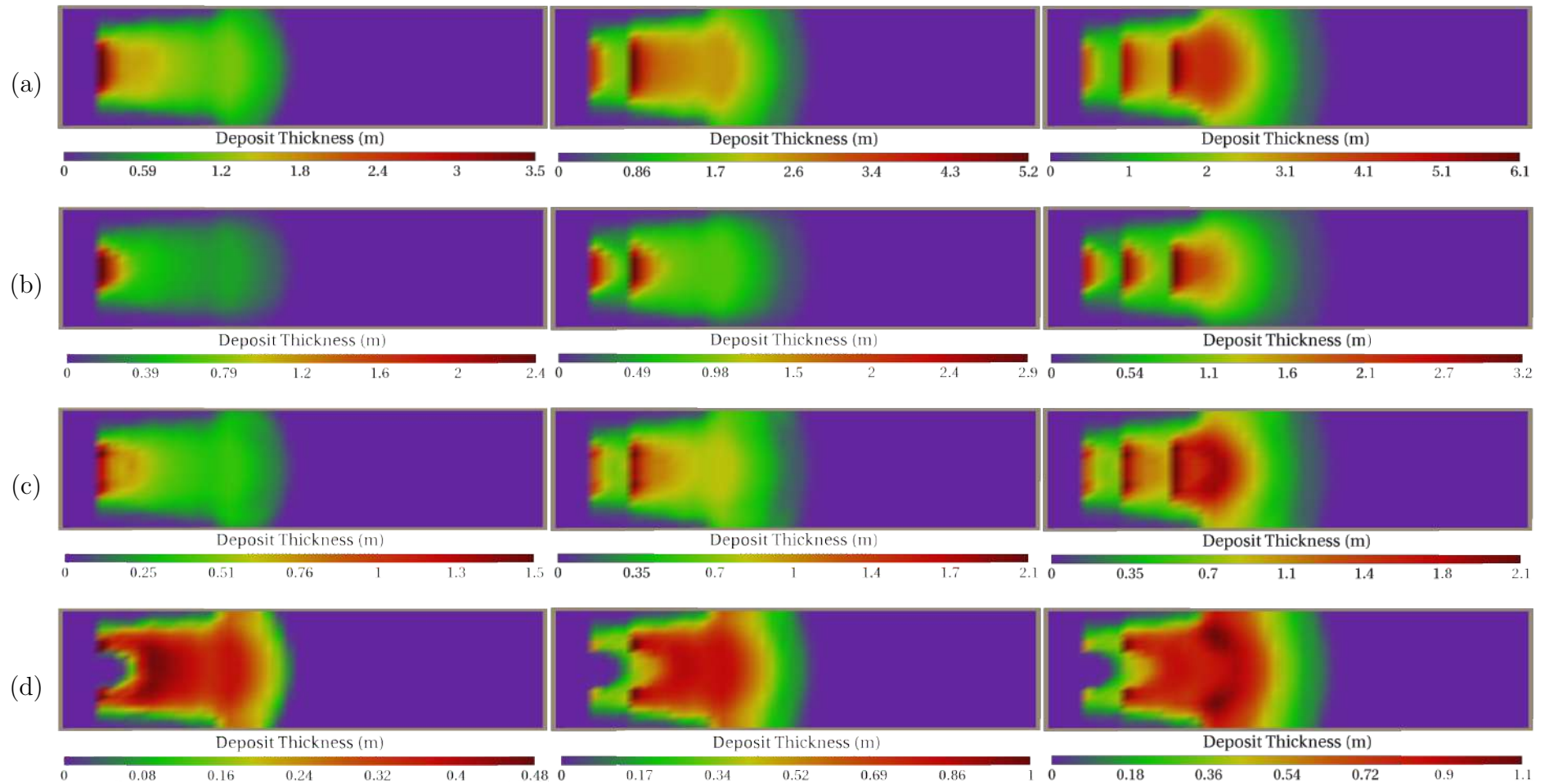


Figure 5.38: Maps of the thickness of the accumulated sediment deposits after each event of the sedimentation on a sloping bed problem, when solved using three sub-steps per event. The maps on the left depict the result after the first event, while the maps at the middle and the right columns respectively represent the states after the second and third events. Each maps row shows the accumulated thickness of the different components: (a) All lithology types; (b) Coarse sand strata; (c) Medium sand deposits; (d) Silt material.

From the results obtained for the simulations using one or three sub-steps, we observe that, in the former case, the deposition of each granulometric fraction happens in blocks within each event, having a clearer horizontal transition between them, as we can see in Figures 5.34 and 5.35. Besides, most of the coarser fractions deposit near the inlet region, filling all the available depositional space. In this case, the small retrogradational-like pattern inside each layer is due to the decay factor f_s . When we use more sub-steps for each event, the coarser grains advance further along the flow, allowing the finer fractions to deposit more volume in the upstream area, as seen in Figures 5.37 and 5.38. This produces more pronounced fining-upwards trends in an event's deposits. Also, we remark that, independent to the number of sub-steps employed, the results clearly show the progradation of each event in regard to the previous one. Furthermore, we argue that the number of sub-steps can be a convenient tool in reproducing different stacking patterns, specially if allied to the fact that a sediment supply can provide different amounts of each sediment type.

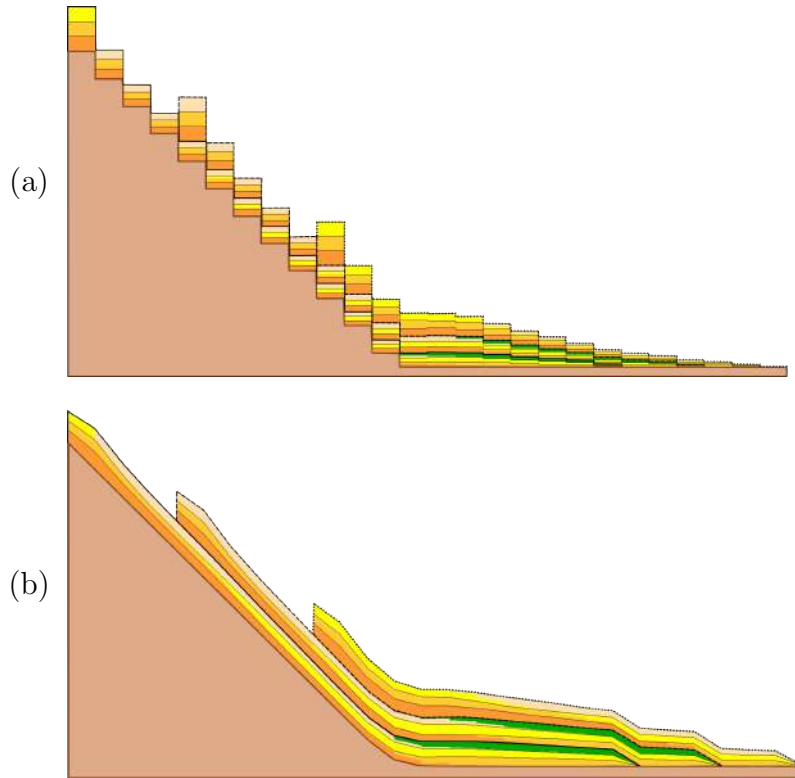


Figure 5.39: View, along a vertical section parallel to the x -direction and placed in the middle of the simulated region, of the stacking pattern produced at the sedimentation on a sloping bed problem, when solved using five lithology types and three sub-steps. (a) Original section. (b) Smoothed section with scaled deposit thicknesses. The thinnest black lines represent the limits of a sub-step deposit, while the different types of thicker black lines separate the strata of each event.

To exemplify a more complex stacking pattern that can be reproduced, we simulate the current scenario using all 5 granulometric fractions, but still considering

the same total sediment volume, and an equal amount of each sediment type. We show the resulting stacking pattern along a section in Figure 5.39. In this case, together with a scaled version of bed's smoothed surface, we also present a view of the deposited lithology blocks, representing the actual data-structure used in the algorithm.

5.10 Sedimentation on an Irregular Terrain

We perform event-driven simulations of the formation of the deposits generated by the accommodation of the sediments brought by a series of turbidity currents. The region studied is the same one contemplated at Section 5.7, i.e., we simulate flows arising from the Almirante Câmara canyon. Although the real turbidite deposits in this region are mainly sandy, we have chosen to employ all five previously mentioned granulometric fractions (clay, silt and fine, medium and coarse sands) so we can see the vertical interchange of the different lithologies. Here, remember that, for the current approach, the properties of the sediments have no impact on the flow and, hence, on the geometric shape of the deposits. In nature, it is estimated that it takes thousands of turbidite events along at least a million years to create a single depositional lobe or seismic horizon in this particular system [112]. However, for a stratigraphic analysis at a basin scale, the smaller channels and bulbs can be disregarded. Indeed, we seek to reproduce the larger-scale structural formations using fewer events.

Initially, we perform two simulations at the stratigraphic application in which we impose a progradational trend between five events by gradually increasing the volume available at each one, as done in Section 5.8. In these cases, we consider that the turbidity currents enter the domain with height $h = 5$ m. Also, we establish that the first event has an inflow duration of 10 min, injecting 6 Mm^3 into the simulated region. Then, as we simulate the next four events, the inflow volume and duration of an event are set as double the amounts conferred to its antecedent. Also, the total sediment volume supplied to each event is, in sequence, 147 Mm^3 , 267 Mm^3 , 508 Mm^3 , 828 Mm^3 , 2259 Mm^3 , which is composed of an equal part of each of the five employed sediment types.

In the first case, we carry out the simulation using a single sedimentation sub-step per event. Figure 5.40 shows the map of the top of the deposits after each event, while Figure 5.41 portrays the obtained deposit thickness. In turn, Figure 5.42 depicts the final thicknesses of the deposits of each sediment type. Then, we show in Figure 5.43b the stacking pattern we obtained along a vertical section that follows the white line in Figure 5.43a. The thicknesses of the strata along the section were scaled by a factor of 50 to aid visualization. Do observe that the layers in the

bottom are, in general, thinner than the ones near the top of the deposit. This happens as a result of the compaction of the layers according to their burial.

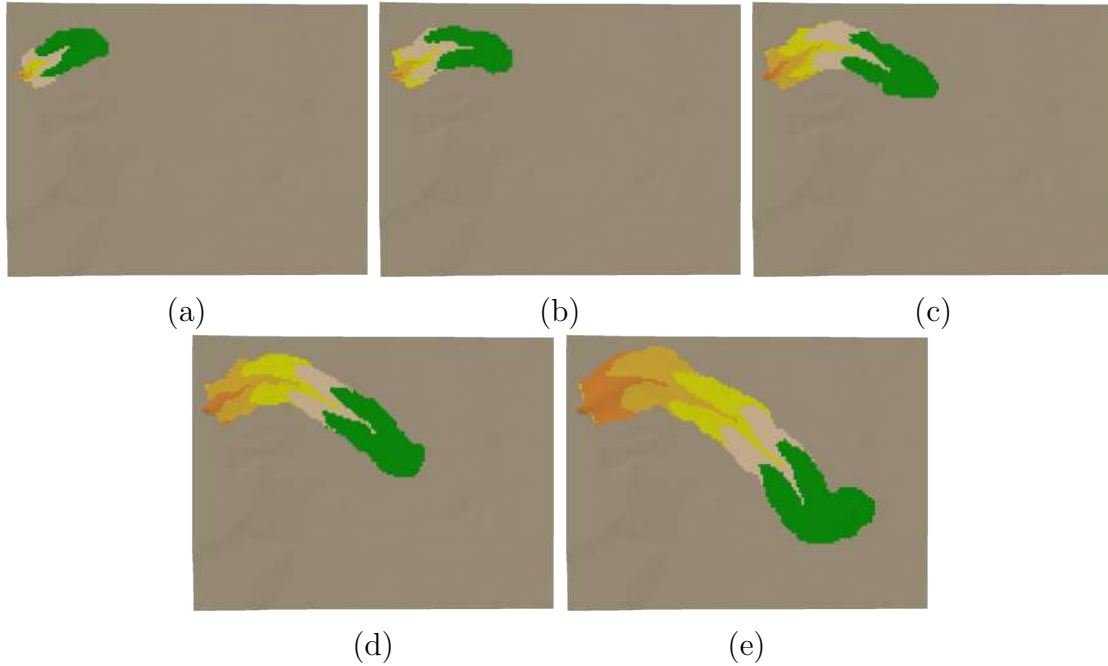


Figure 5.40: Map views of the top of the resulting deposits after each event of the sedimentation on an irregular terrain problem, computed imposing event progradation and using one sedimentation sub-step. Here the progression (a) through (e) represents the sequence of the simulated events.

In sequence, for the second case, we execute the simulation using three sub-steps per event. We show, in Figure 5.44, the maps of the top of the deposits after each sub-step, while Figure 5.45 exhibits the obtained deposit thickness. Also, Figure 5.46 displays the final thickness related to each deposit type. Then, we present, in Figure 5.47b, the resulting stratification along a vertical section that follows the white line in Figure 5.47a. Again, we scaled the strata along the section by a factor of 50 to improve the visualization.

We can observe that the obtained results using one and three sub-steps are in agreement with the simulations performed in Sections 5.8 and 5.9, where simpler terrains were considered. In fact, using a single sub-step, the sediments settle in blocks, producing more localized concentrations of the distinct sediment types, as we can see in Figure 5.46. On the other hand, with more sub-steps, the deposits of each sediment type are more spread out horizontally, as seen in Figure 5.46. Besides, in this case, the fining-upwards trends within each event's deposits are more pronounced. It is worth to remember that these trends are a characteristic of the Bouma sequence [111], which describes the ideal vertical succession of strata deposited by low-density turbidity currents.

Next, we run a simulation on the stratigraphic application using events with the

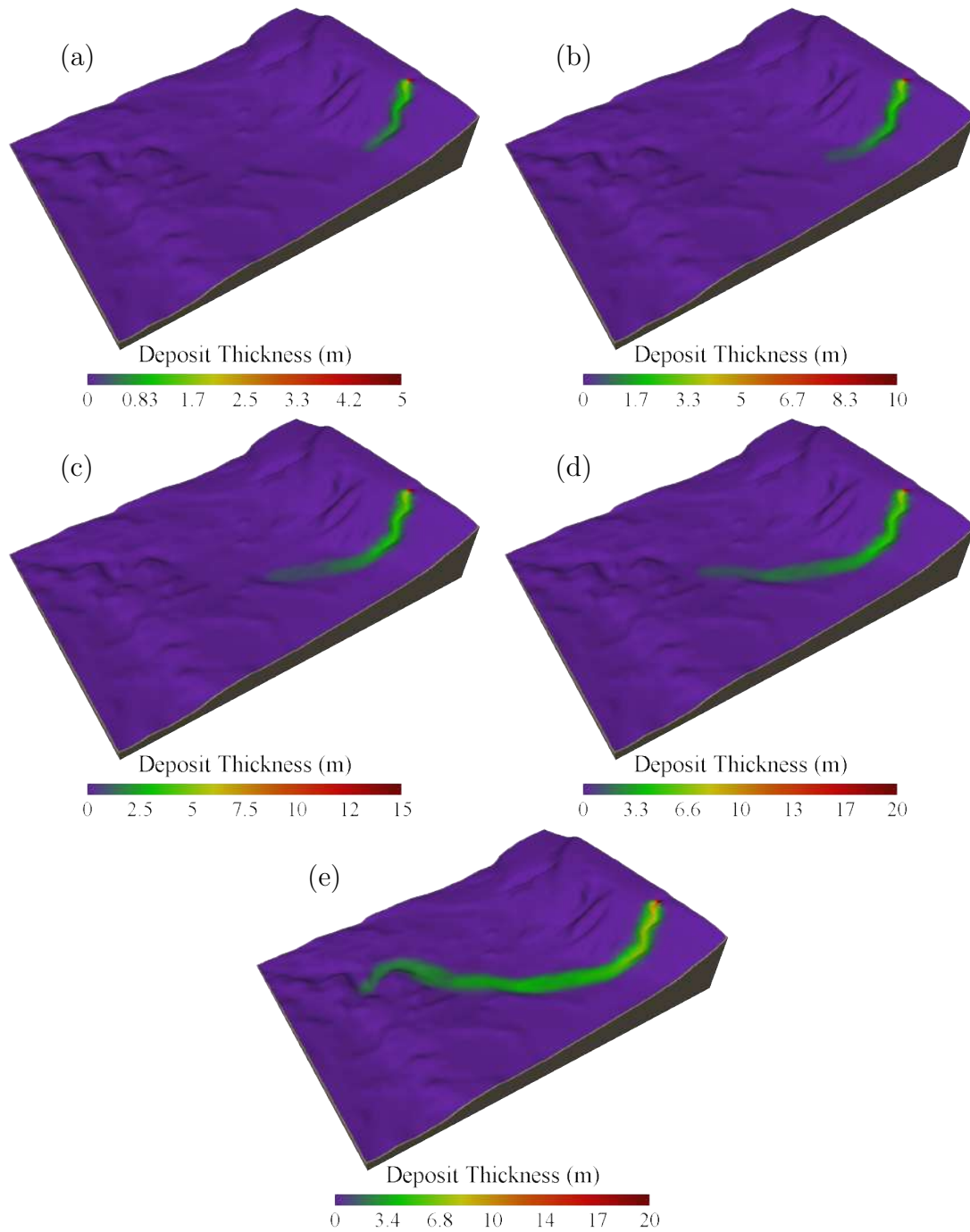


Figure 5.41: 3D views of the obtained deposit thicknesses after each event of the sedimentation on an irregular terrain problem, computed imposing event progradation and using one sedimentation sub-step. Here the progression (a) through (e) represents the sequence of the simulated events.

same base characteristics and three sedimentation sub-steps. Here, the current associated with each event enters the domain with $h = 5$ m, injecting a total of 96 Mm^3 into the region in 160 min. We perform six simulation steps of the stratigraphic application, considering a single event at the each of the first four steps. In these steps, we provide to each event a total sediment volume of 1167 Mm^3 , which is evenly split between all five sediment types. Then, in the fifth step, we simulate two events

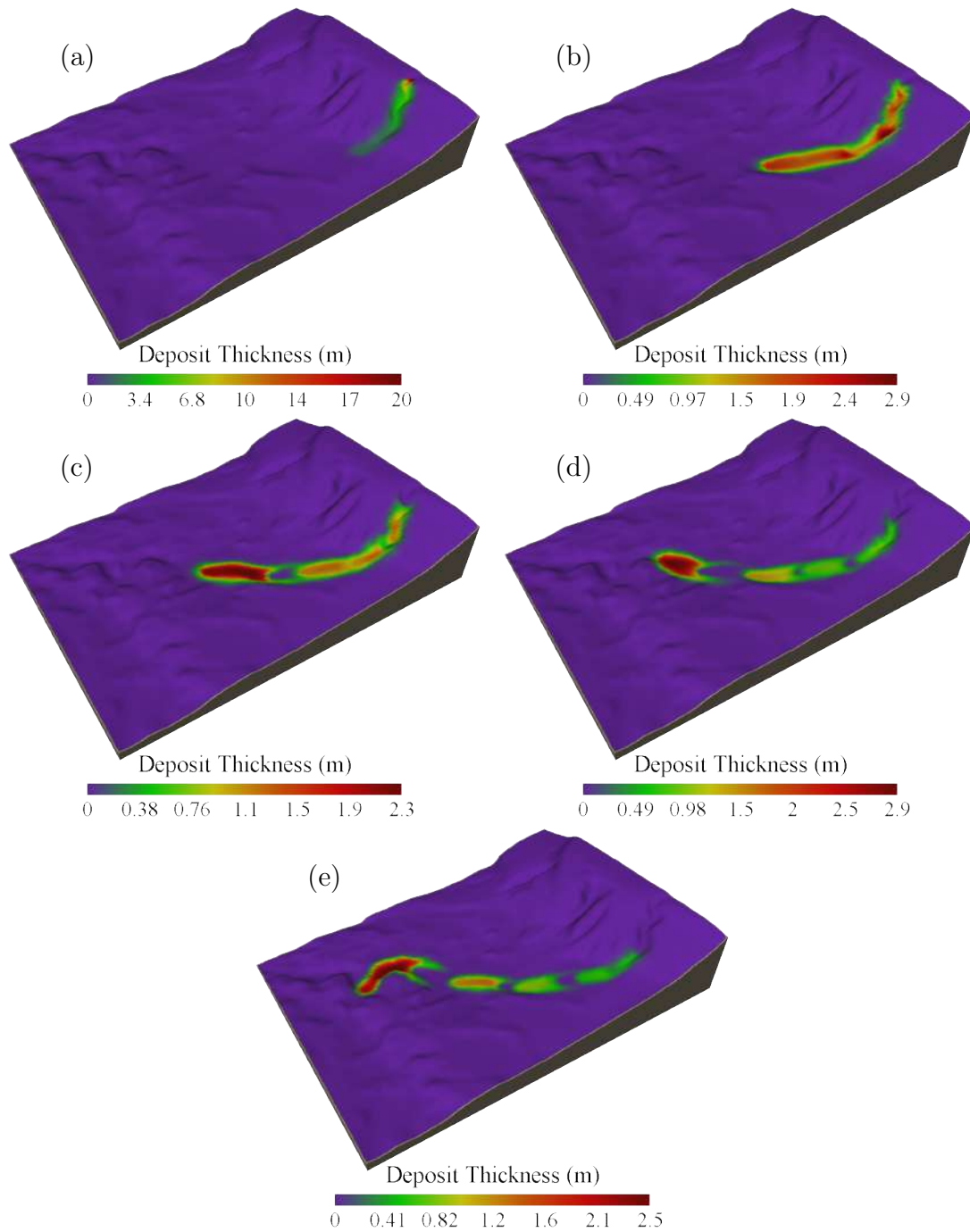


Figure 5.42: 3D views of the final deposit thicknesses associated with each sediment type of the sedimentation on an irregular terrain problem, computed imposing event progradation and using one sedimentation sub-step. (a) Coarse sand; (b) Medium sand; (c) Fine sand; (d) Silt; (e) Clay.

with the recurrence of four years. In this case, the simulated current advanced further than in the previous events. Hence we provided more sediments to the event, composing a total of 2334 Mm^3 , so that we fill most of the accommodation space determined by the maximum point-wise current heights. At last, in the final step, we took into account four events, using the same parameters employed in the previous one. We show the final deposit thickness associated with each sediment type

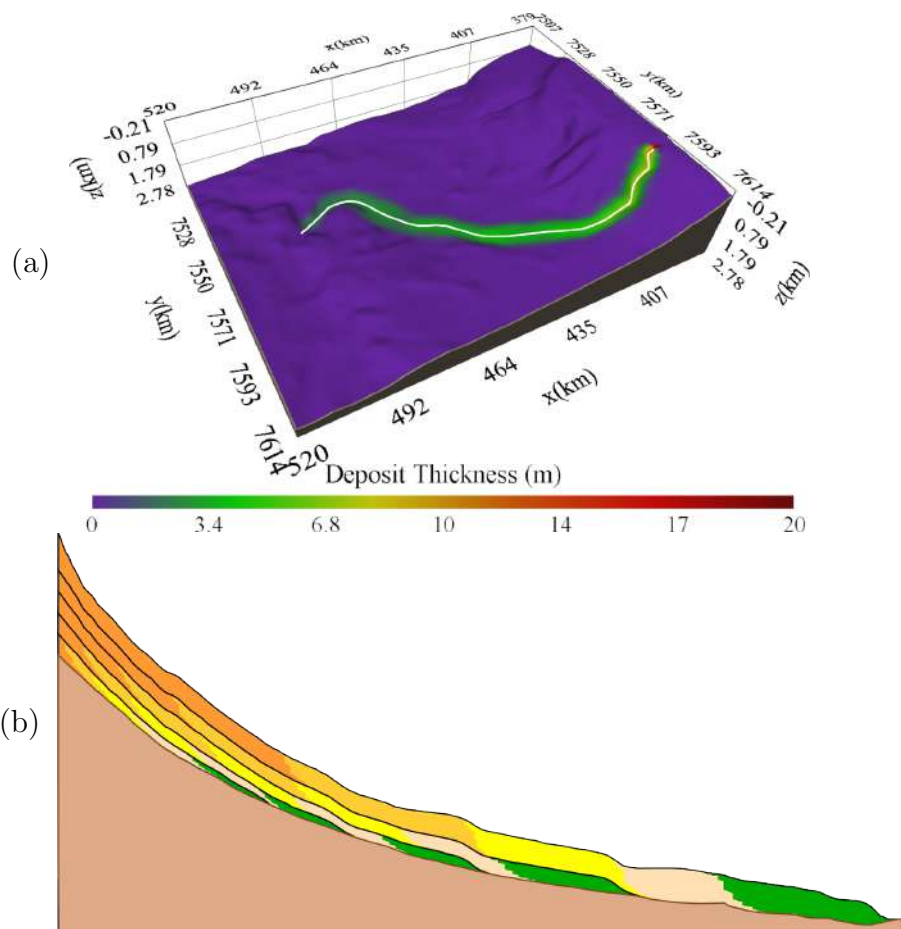


Figure 5.43: (a) shows the position of the white line along which we extracted the vertical section displayed in (b), containing the stacking pattern obtained at the sedimentation on an irregular terrain problem, computed imposing event progradation and using one sedimentation sub-step. Layer thicknesses in (b) were scaled by a factor of 50 to aid visualization.

in Figure 5.48 and show the stacking pattern obtained along a vertical section in Figure 5.49. Although we did not impose a progradational trend between events, we obtained progradational depositional patterns, while also keeping a fining-upwards trend within each event's deposits.

We compare our result with the map of sedimentary facies at the modern seabed presented by MACHADO *et al.* [112] in Figure 5.50, which overlays the shape of the real turbidite system onto the deposit thickness map we obtained. We observe that the actual path of the proximal trough agrees with the simulated deposits. Additionally, the real turbidite lobe is well represented by the increase in the deposition area before the current reenters a channel. In our simulation, the flow did not advance further due to the chosen boundary conditions. However, we remark that different deposit geometries and stacking patterns can be recreated by wisely selecting the input parameters. For instance, increasing (or decreasing) the event volume or specific discharges, or changing the available sediment supply would pro-

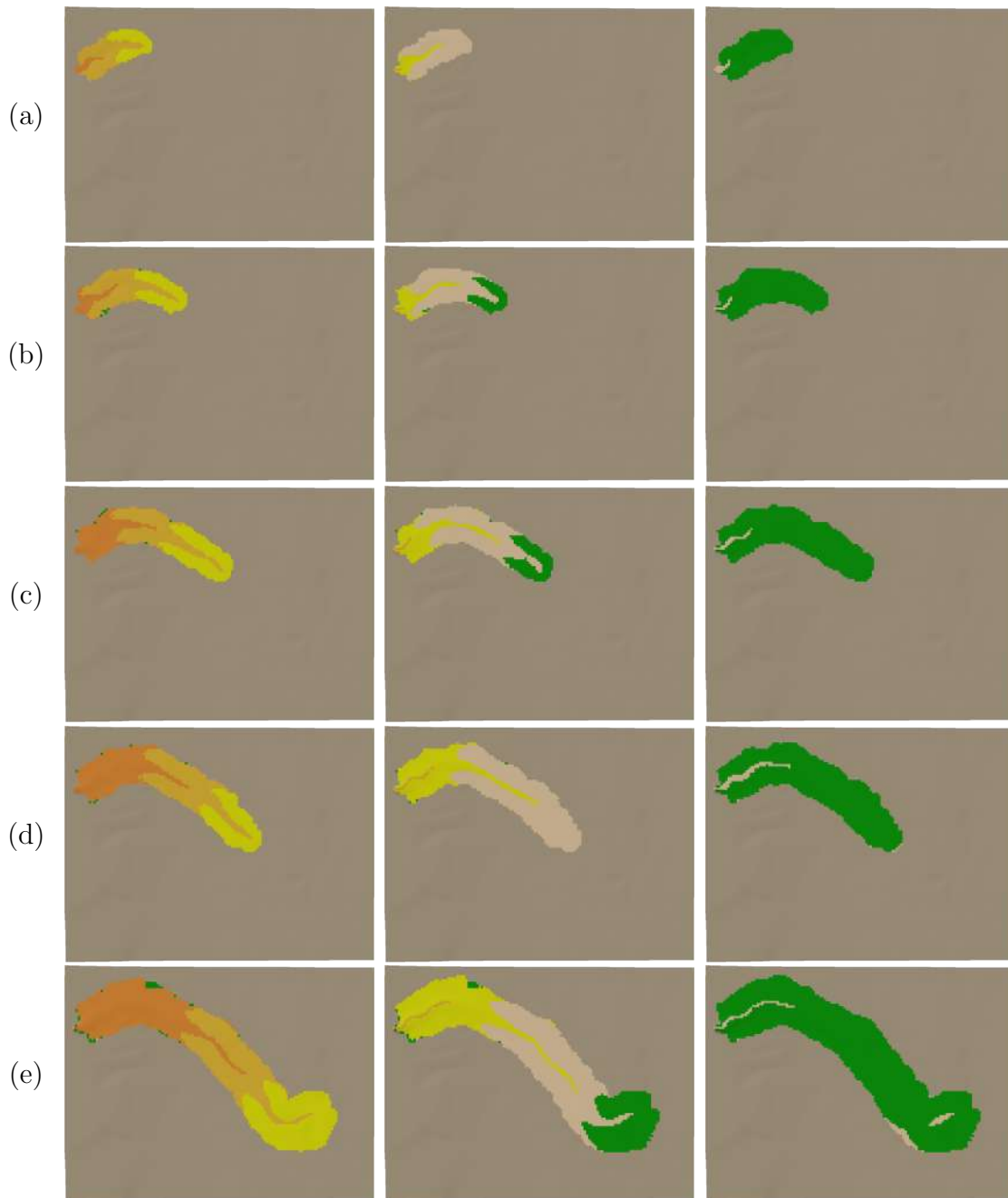


Figure 5.44: Map views of the top of the resulting deposits after each event of the sedimentation on an irregular terrain problem, computed imposing event progradation and using three sub-steps. Here the progression (a) through (e) represents the sequence of the simulated events, while the maps from left to right mark the sub-steps succession.

duce distinct outcomes. Therefore, we ascertain that our approach can reproduce stacking patterns common in sequence stratigraphy, and can recreate depositional features of turbidite systems.

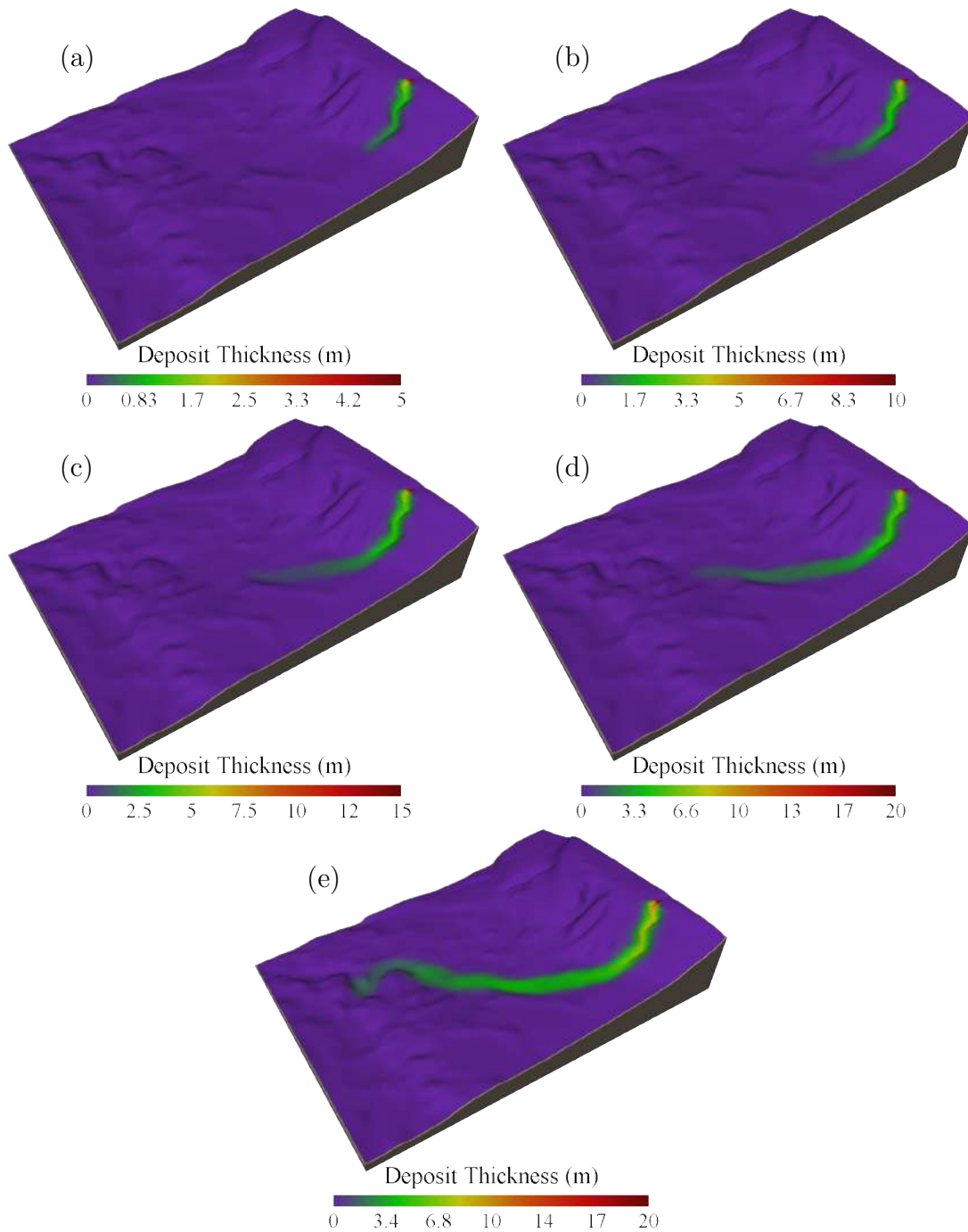


Figure 5.45: 3D views of the obtained deposit thicknesses after each event of the sedimentation on an irregular terrain problem, computed imposing event progradation and using three sub-steps. Here the progression (a) through (e) represents the sequence of the simulated events.

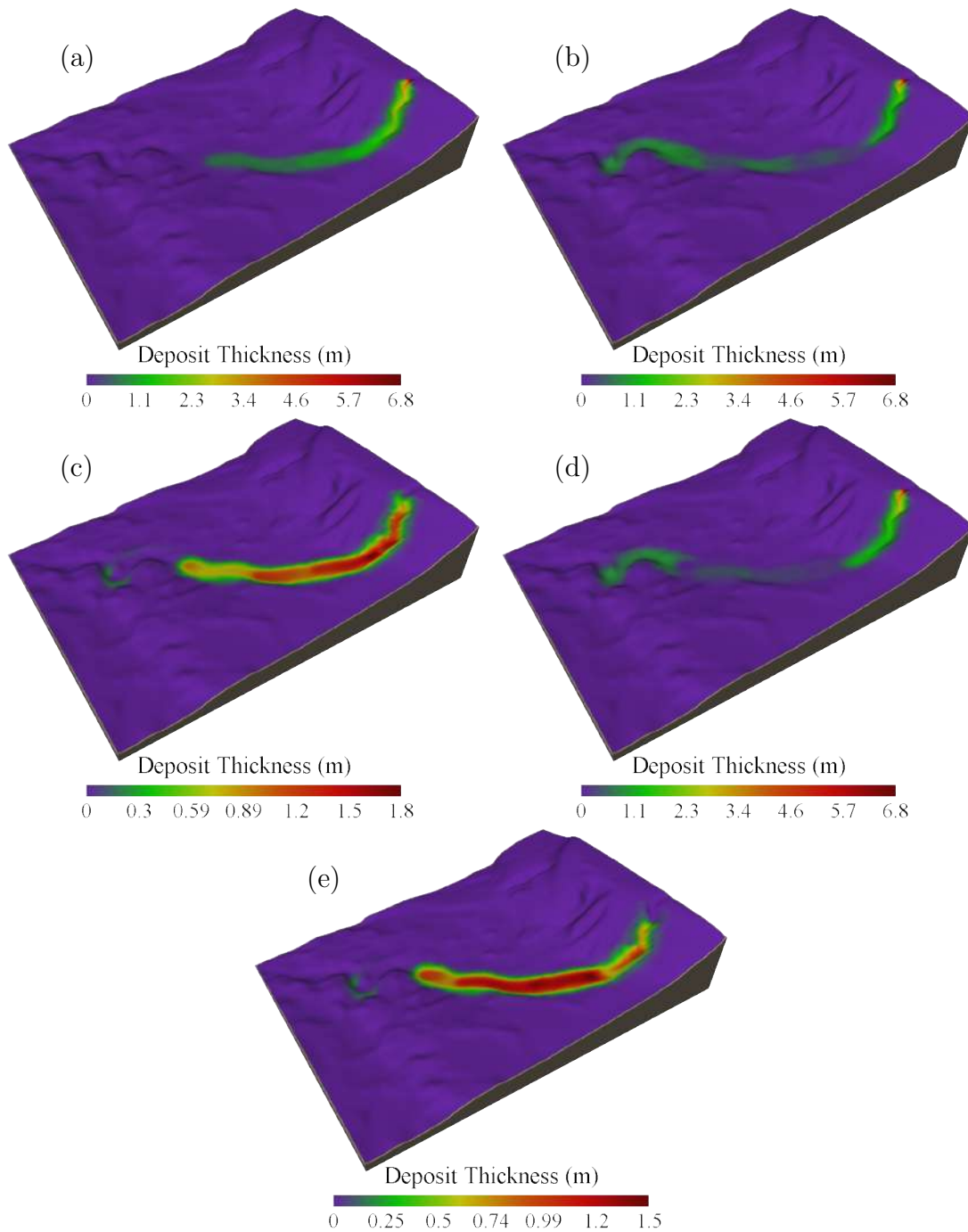


Figure 5.46: 3D views of the final deposit thicknesses associated with each sediment type of the sedimentation on an irregular terrain problem, computed imposing event progradation and using three sub-steps. (a) Coarse sand; (b) Medium sand; (c) Fine sand; (d) Silt; (e) Clay.

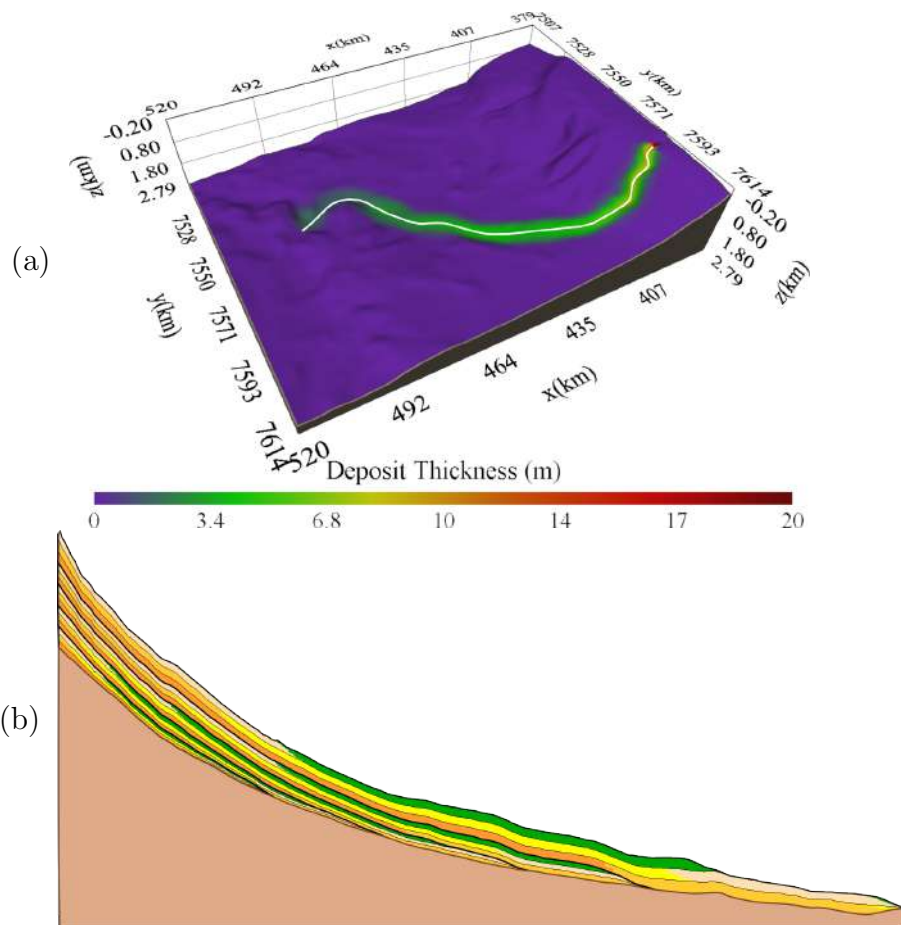


Figure 5.47: (a) shows the position of the white line along which we extracted the vertical section displayed in (b), containing the stacking pattern obtained at the sedimentation on an irregular terrain problem, computed imposing event progradation and using three sub-steps. Layer thicknesses in (b) were scaled by a factor of 50 to aid visualization.

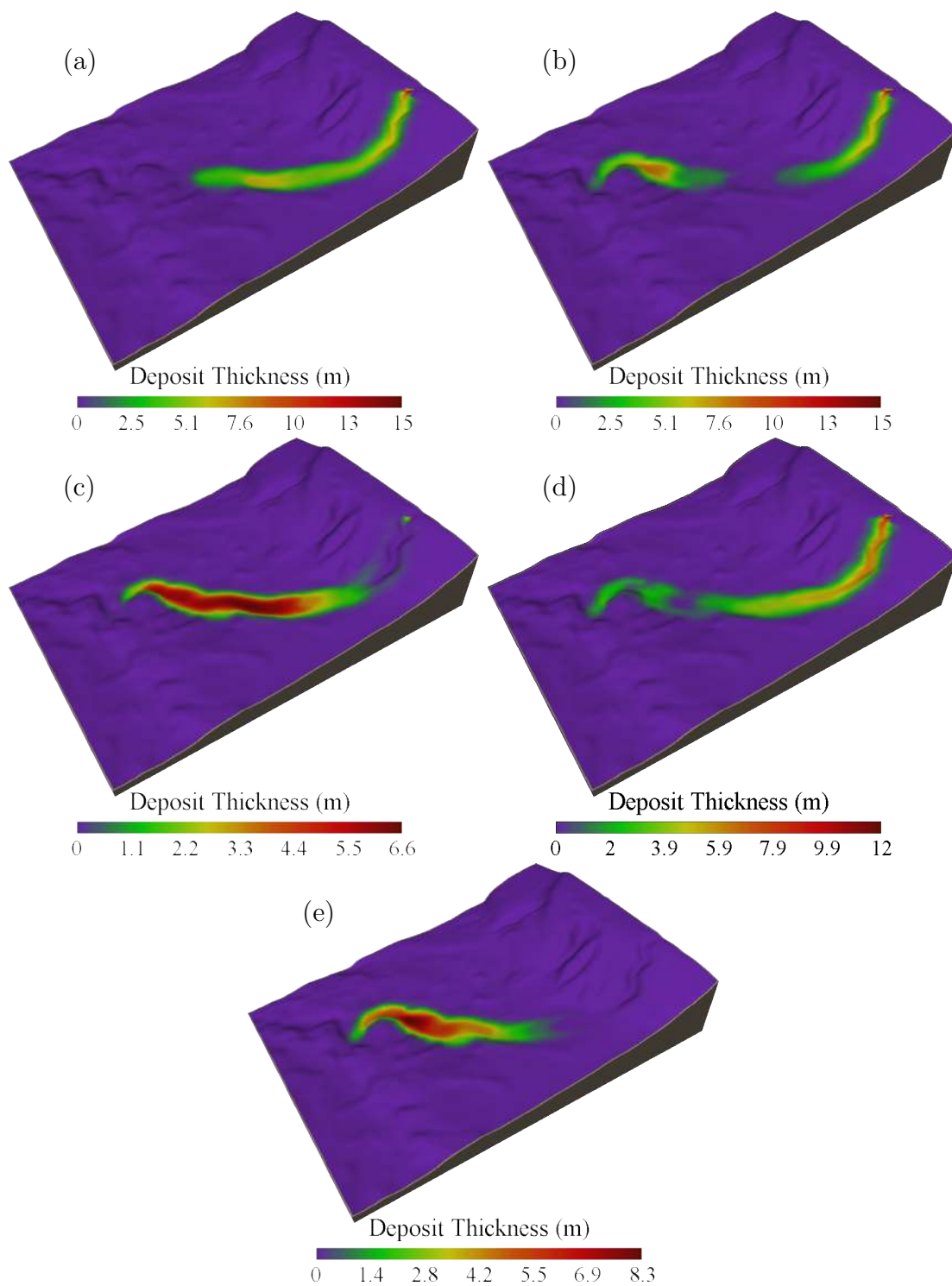


Figure 5.48: 3D views of the final deposit thicknesses associated with each sediment type of the sedimentation on an irregular terrain problem, computed using three sub-steps per event. (a) Coarse sand; (b) Medium sand; (c) Fine sand; (d) Silt; (e) Clay.

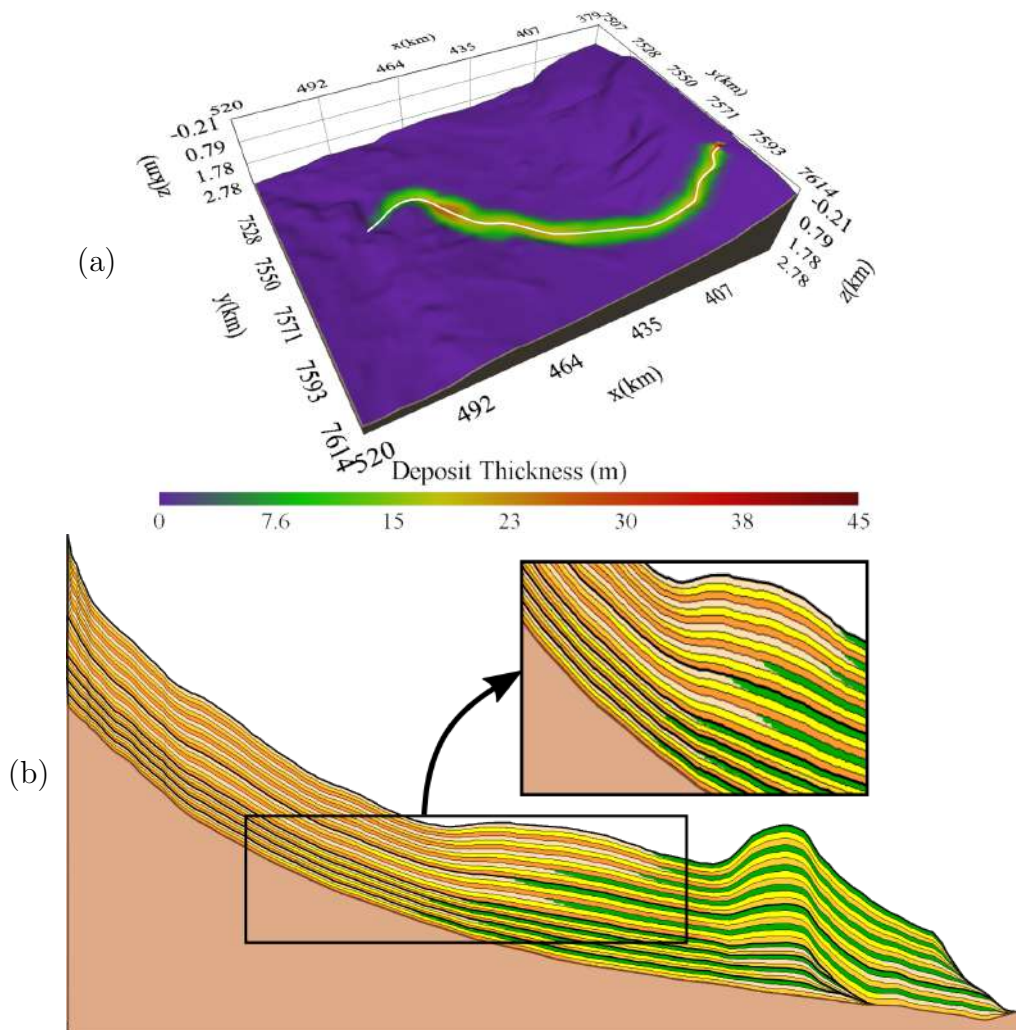


Figure 5.49: (a) shows the position of the white line along which we extracted the vertical section displayed in (b), containing the stacking pattern obtained at the sedimentation on an irregular terrain problem, computed using three sub-steps per event. Here the thicker lines mark the transition between each stratigraphic application's simulation step. Layer thicknesses in (b) were scaled by a factor of 50 to aid visualization.

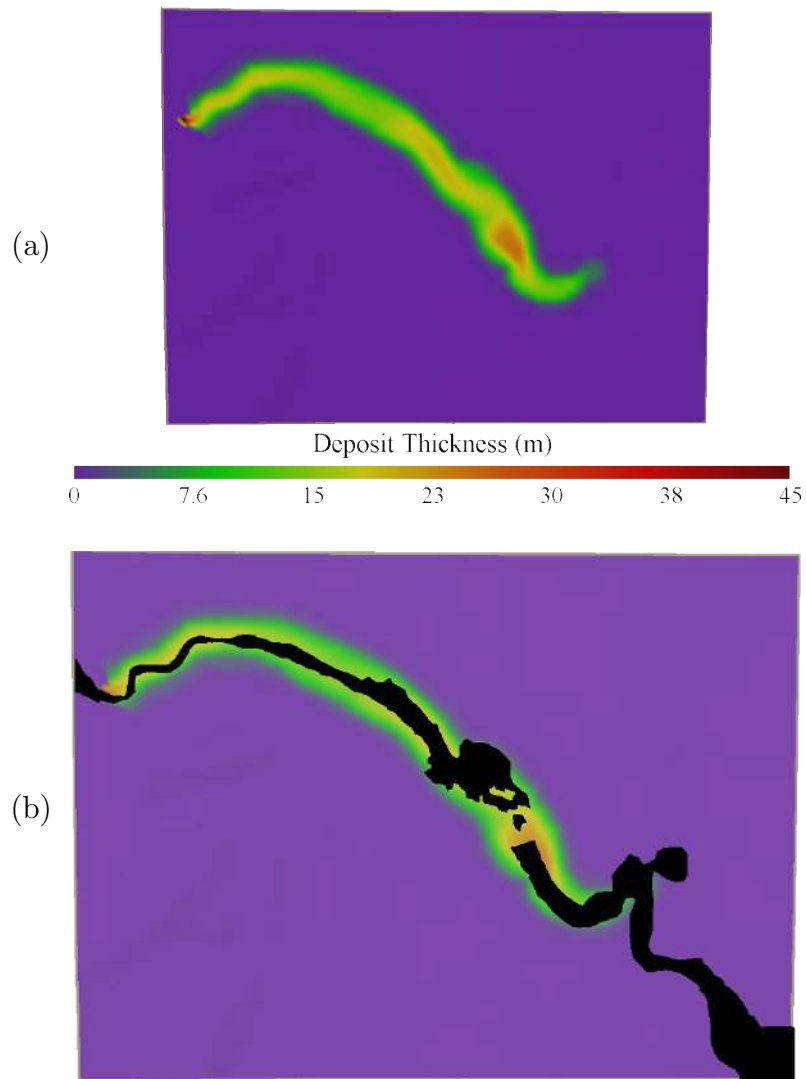


Figure 5.50: (a) Map view of the deposit thickness by the end of the simulation of the sedimentation on an irregular terrain problem, computed using three sub-steps per event; (b) Overlay of the real turbidite system presented by MACHADO *et al.* [112] (in black) onto the obtained deposit thickness map.

Chapter 6

Conclusions and Future Work

We proposed a new event-driven approach to simulate the formation of turbidite depositional systems at stratigraphic scale. Indeed, we couple a process-based dynamic flow model with a practical sedimentation technique. The obtained results suggest good applicability to the stratigraphic scale.

The adopted turbidity current flow model is based on the shallow water approximation and considers an augmented friction coefficient to account for the drag forces at the water-current and current-bed interfaces. In this context, to solve the governing equations, we developed an original finite element flux-corrected transport scheme. Its low-order formulation is built by adding a novel Rusanov-like scalar dissipation scaled by a shock-capturing operator to standard Galerkin equations. This artificial diffusion depends on the hydrostatic reconstruction of the current height and specific discharges. The use of the latter reduces the diffusion near dry/wet fronts to avoid fluid incorrectly jumping into dry regions. Indeed, it nullifies the diffusion for current heights smaller than the cut-off discussed in Section 2.6. Besides, our results suggest that the presented shock detector yields better results than the ones introduced by BASTING and KUZMIN [90] and GUERMOND *et al.* [87].

Next, the high-order system is composed by summing, to the low-order one, limited anti-diffusive fluxes linearized around the low-order solution. Limiting is performed with a Zalesak-type flux limiter that considers the hydrostatic reconstruction of the fluid's height, together with a minmod prelimiter. In this regard, we concluded that, for the shallow water equations, the minmod prelimiter should be applied to each flux component individually. In contrast, the correction factors computed by the flux limiter should be synchronized by the factor relative to the current height component. An iterative nonlinear implicit time integration scheme with source term linearization is employed in both the high- and low-order systems. Besides, in this work, all finite element related implementations were assisted by the deal.II [55] library and its module for parallel computing with shared memory devices.

During a simulation, as the fluid’s height tends to zero, velocities are desingularized using a cut-off value based on the local ratio between element and mesh sizes. Also, we correct the bed elevation at dry nodes to avoid unnatural dynamics due to the discretization of the current and bed surfaces. We showed that this procedure, initially applied by BRUFAU and GARCÍA-NAVARRO [93] to triangular elements, also produces excellent results with quadrilateral elements. Plus, we found that the correction can be performed for each element individually. When assembling an element’s matrix contributions, the correction can be based only on its nodal values, without the need to ever iterate through all the elements adjusting each nodal bed elevation. In addition, we linearly vary the bed friction near the bed to help stabilize the flow and adaptively update the time steps throughout the simulation to enforce a maximum CFL constraint and to help preserve the current height positivity.

We compared our approach with a stabilized finite element formulation and found that the FCT scheme is more robust, presenting good results in all the tested cases. Regarding the stabilized method, the $YZ\beta$ technique also produced plausible results. However, its usage requires some tweaking with the reference values for the variables h , q_x and q_y . We remark that the perturbations it created past the advancing wavefront in Section 5.4 might have a more significant impact on simulations with dry and wet cells. For the example in Section 5.5, it made some fluid go up the larger bump and produced negative height values at some dry points. Thus, this method might be unsuitable to regions with more irregular terrains and more complex flow dynamics. All in all, the results of our flow solver indicate mass conservation and a good agreement with analytical solutions and simulations performed by others.

In sequence, the new sedimentation algorithm, which was based on the works of CARVALHO *et al.* [7–9], is coupled to the flow solver. In this context, one of our main contributions is using more accurate flow lines since our flow simulation is based on the actual physical transient process, also accounting for bed friction effects, instead of considering a simplified steady-state equilibrium. Plus, our approach allows the simulation of currents entering an aqueous body, as opposed to only computing the flow of a water body. These features, allied to the proposed event-driven framework, makes our work more suitable for turbidite systems.

Moreover, as we carry sediments along streaklines, we radially scatter them from each grid cell traversed, instead of only depositing along the lines. Also, here we further constrain the space available for sediment accommodation by each event’s point-wise maximum current height. Nevertheless, for the deposition, we do not assess the stability angle of each lithology, since we assume it is indirectly enforced by the limit on the vacant depositional space. However, when we radially spread sediments from a cell crossed by a streakline, we gradually reduce the accommodation space as we distance from it. In practice, this constraint has an effect similar

to considering a stability angle. In addition, to try to reproduce stacking patterns seen in real turbidite systems, we allow the deposition of a single event to be made in sub-steps that yet have more limited accommodation spaces. Finally, after the deposition step of each event, we compact the deposited sediment layers as a result of the new overburden.

The streaklines used in the sedimentation operation are computed with a new strategy in which points/particles are adaptively added or removed. New particle positions are interpolated using centripetal Catmull-Rom splines and evolved through time with a 4th-order Runge-Kutta scheme. An efficient method to track the cell that contains each point is adopted to speed up the interpolation of properties on a point's position for its displacement.

All the meshes used in the test problems were created with the Gmsh [108] application. In the Almirante Câmara's case, the employed mesh is created by remeshing the underwater region of a 2D regular grid coming from an external stratigraphic application. To this end, we developed a procedure that defines the boundary of the simulation mesh using B-spline patches whose control points come from the original grid.

In terms of future works, one possible approach is to enhance the adopted physical and numerical models. For instance, it is known that turbulence effects constitute a relevant mechanism in turbidity current dynamics. Thus, a turbulence model, such as the Smagorinsky scheme [132], could improve our results. Another action would be to compute the turbidity current's density and viscosity based on its water/sediment composition. Also, erosive forces could even be contemplated. In this case, a potential plan of attack is to create a pseudo-heuristic based on the Hjulström diagram [133] and the point-wise maximum flow speed during each event.

Furthermore, we also could derive and implement a two-layer shallow-water model where the hydrostatic hypothesis is applied within each layer. This type of model can be employed to simulate flows with different densities [36] and tsunamis generated by landslides [134]. In the latter case, one bottom layer refers to the landslide, considered as an incompressible liquid, and the other is the seawater. One known issue with this kind of model is that the system is conditionally hyperbolic due to the coupling terms between the layers [60–62]. Also, if the velocities of the two layers are too different, it is expected that Kelvin-Helmholtz instabilities arise in the interface between them.

Another suggestion is to consider a depth-averaged non-hydrostatic extension for the shallow water equations. In this sense, JESCHKE *et al.* [135] proved that the use of a quadratic vertical pressure profile makes the system correspondent to the fully nonlinear weakly dispersive Boussinesq-type equations. A distinct issue concerns the imposition of consistent absorbing (or transparent) boundary condi-

tions throughout the simulation, which is usually done based on the analysis of the incoming characteristic waves. These constraints can reduce the simulation error and improve convergence rates [136, 137]. However, their implementation can be a particularly daunting task in complex simulations where different parts of the boundary switch between inlet/outlet or subcritical/supercritical states.

We can find additional courses for improvements in speeding up the simulation. For example, we could adaptively refine/coarse the mesh to optimally distribute computational effort between different regions. Following a similar rationale, dry elements far from dry/wet interfaces could be left out of computation, not being assigned any degree of freedom. Thus, as the current is flooding into an initially dry region, a much smaller system of equations would have to be assembled and solved. At last, although we employed deal.II's module for parallelism on shared memory devices, the use of clusters with distributed memory still has not been addressed.

Finally, we suggest further study should be performed to better understand the impact of different parameters and boundary conditions on the resulting deposits. For instance, using the same bathymetry, we could evaluate how the deposits change if we simulate more events with shorter durations, instead of less and longer events. This type of analysis may need to be region dependent, leading to the assessment of the model's applicability in different areas of interest.

Bibliography

- [1] IMBERT, P. “Chapter 10 - Industrial application of deep-sea sediments”. In: HüNeke, H., Mulder, T. (Eds.), *Deep-Sea Sediments*, v. 63, *Developments in Sedimentology*, Elsevier, pp. 715–764, 2011. doi: 10.1016/B978-0-444-53000-4.00010-X.
- [2] MEIBURG, E., KNELLER, B. “Turbidity currents and their deposits”, *Annual Review of Fluid Mechanics*, v. 42, n. 1, pp. 135–156, 2010. doi: 10.1146/annurev-fluid-121108-145618.
- [3] YINCAN, Y., OTHERS. “Chapter 15 - Development laws of geological hazards and hazard geology regionalization of China seas”. In: Yincan, Y., others (Eds.), *Marine Geo-Hazards in China*, Elsevier, pp. 657–687, 2017. ISBN: 978-0-12-812726-1. doi: 10.1016/B978-0-12-812726-1.00015-2.
- [4] MURPHY, M. A., SALVADOR, A. “International stratigraphic guide - An abridged version”, *Episodes*, v. 22, n. 4, pp. 255–271, Dec. 1999.
- [5] GRANJEON, D. “3D forward modelling of the impact of sediment transport and base level cycles on continental margins and incised valleys”. In: *From Depositional Systems to Sedimentary Successions on the Norwegian Continental Margin*, cap. 16, pp. 453–472, John Wiley & Sons, Ltd, 2014. ISBN: 9781118920435. doi: 10.1002/9781118920435.ch16.
- [6] HUANG, X., LIU, K., ZOU, C., et al. “Forward stratigraphic modelling of the shallow-water delta system in the Poyang Lake, southern China”, *Journal of Geochemical Exploration*, v. 144, pp. 74–83, 2014. ISSN: 0375-6742. doi: 10.1016/j.gexplo.2014.01.019. Computational modeling of fluid flow and geochemical processes in ore-forming and geoenvironmental systems.
- [7] CARVALHO, C. V. A., LOPES, A. A. O., MARTHA, L. F., et al. “Simulation of transport and deposition of siliciclastic sediments in platform, slope, and basin environments”, *Proceedings of the 23rd GOCAD Meeting, Nancy, France*, pp. 13–28, 2003.

- [8] CARVALHO, C. V. A. *Simulação de transporte e deposição de sedimentos siliciclásticos em ambientes de plataforma, talude e bacia*. PhD thesis, Civil Engineering Department, Pontifical Catholic University of Rio de Janeiro (PUC-RIO), 2002. Available at: <<https://www.maxwell.vrac.puc-rio.br/colecao.php?strSecao=resultado&nrSeq=3529@2>>. Accessed on: 1 mar. 2020.
- [9] CARVALHO, C. V. A., MARTHA, L. F., JÚNIOR, E. D. A. V. “Transport of sediments in numerical simulation of sedimentary basins”, *Proceedings of the 4th International WorkShop to ACMGE, Ouro Preto, Brasil*, pp. 17–20, 2003.
- [10] SHAFIE, K. R. K., MADON, M. “A review of stratigraphic simulation techniques and their applications in sequence stratigraphy and basin analysis”, *Bulletin of the Geological Society of Malaysia*, v. 54, Nov. 2008. doi: 10.7186/bgsm54200814.
- [11] STROBEL, J., CANNON, R., CHRISTOPHER, G. S., et al. “Interactive (SEDPAK) simulation of clastic and carbonate sediments in shelf to basin settings”, *Computers & Geosciences*, v. 15, n. 8, pp. 1279–1290, 1989. ISSN: 0098-3004. doi: 10.1016/0098-3004(89)90092-7.
- [12] LEE, Y.-H., HARBAUGH, J. W. “Stanford’s sedsim project: dynamic three-dimensional simulation of geologic processes that affect clastic sediments”. In: Pflug, R., Harbaugh, J. W. (Eds.), *Computer Graphics in Geology*, pp. 113–127. Springer, Berlin, Heidelberg, 1992. ISBN: 978-3-540-46768-7. doi: 10.1007/BFb0117791.
- [13] MEIBURG, E., RADHAKRISHNAN, S., NASR-AZADANI, M. “Modeling gravity and turbidity currents: computational approaches and challenges”, *Applied Mechanics Reviews*, v. 67, n. 4, pp. 040802, Jul. 2015. doi: 10.1115/1.4031040.
- [14] CAMATA, J. J., SILVA, V., VALDURIEZ, P., et al. “In situ visualization and data analysis for turbidity currents simulation”, *Computers & Geosciences*, v. 110, pp. 23–31, 2018. ISSN: 0098-3004. doi: 10.1016/j.cageo.2017.09.013.
- [15] ROSSA, A. L., COUTINHO, A. L. G. A. “Parallel adaptive simulation of gravity currents on the lock-exchange problem”, *Computers & Fluids*, v. 88, pp. 782–794, 2013. ISSN: 0045-7930. doi: 10.1016/j.compfluid.2013.06.008.

- [16] GUERRA, G. M., ZIO, S., CAMATA, J. J., et al. “Numerical simulation of particle-laden flows by the residual-based variational multiscale method”, *International Journal for Numerical Methods in Fluids*, v. 73, n. 8, pp. 729–749, 2013. doi: 10.1002/fld.3820.
- [17] ELIAS, R. N., PARAIZO, P. L. B., COUTINHO, A. L. G. A. “Stabilized edge-based finite element computation of gravity currents in lock-exchange configurations”, *International Journal for Numerical Methods in Fluids*, v. 57, n. 9, pp. 1137–1152, 2008. doi: 10.1002/fld.1781.
- [18] ZIO, S., DA COSTA, H. F., GUERRA, G. M., et al. “Bayesian assessment of uncertainty in viscosity closure models for turbidity currents computations”, *Computer Methods in Applied Mechanics and Engineering*, v. 342, pp. 653–673, 2018. ISSN: 0045-7825. doi: 10.1016/j.cma.2018.08.023.
- [19] GUERRA, G. M., ZIO, S., CAMATA, J. J., et al. “Uncertainty quantification in numerical simulation of particle-laden flows”, *Computational Geosciences*, v. 20, n. 1, pp. 265–281, Feb 2016. ISSN: 1573-1499. doi: 10.1007/s10596-016-9563-6.
- [20] GIRALDO, F. X. “Continuous and discontinuous Galerkin methods for atmospheric modeling”. In: *Seminar on Recent Developments in Numerical Methods for Atmosphere and Ocean Modelling, 2-5 September 2013*, pp. 167–181, Shinfield Park, Reading, 2014. ECMWF.
- [21] SÁRMÁNY, D., HUBBARD, M. E. “Upwind residual distribution for shallow-water ocean modelling”, *Ocean Modelling*, v. 64, pp. 1–11, 2013. ISSN: 1463-5003. doi: 10.1016/j.ocemod.2012.12.013.
- [22] KAREL’SII, K. V., PETROSYAN, A. S., CHERNYAK, A. V. “Nonlinear theory of the compressible gas flows over a nonuniform boundary in the gravitational field in the shallow-water approximation”, *Journal of Experimental and Theoretical Physics*, v. 116, n. 4, pp. 680–697, Apr. 2013. ISSN: 1090-6509. doi: 10.1134/S1063776113030175.
- [23] KLIMACHKOV, D. A., PETROSYAN, A. S. “Nonlinear theory of magnetohydrodynamic flows of a compressible fluid in the shallow water approximation”, *Journal of Experimental and Theoretical Physics*, v. 123, n. 3, pp. 520–539, Sept. 2016. ISSN: 1090-6509. doi: 10.1134/S1063776116070098.
- [24] DÍAZ, M. J. C., FERNÁNDEZ-NIETO, E. D., FERREIRO, A. M., et al. “Two-dimensional sediment transport models in shallow water equations. A sec-

- ond order finite volume approach on unstructured meshes”, *Computer Methods in Applied Mechanics and Engineering*, v. 198, n. 33, pp. 2520–2538, 2009. ISSN: 0045-7825. doi: 10.1016/j.cma.2009.03.001.
- [25] JUEZ, C., MURILLO, J., GARCÍA-NAVARRO, P. “A 2D weakly-coupled and efficient numerical model for transient shallow flow and movable bed”, *Advances in Water Resources*, v. 71, pp. 93–109, 2014. ISSN: 0309-1708. doi: 10.1016/j.advwatres.2014.05.014.
- [26] LIU, X., SEDANO, J. A. I., MOHAMMADIAN, A. “A robust coupled 2-D model for rapidly varying flows over erodible bed using central-upwind method with wetting and drying”, *Canadian Journal of Civil Engineering*, v. 42, n. 8, pp. 530–543, 2015. doi: 10.1139/cjce-2014-0524.
- [27] APOSTOLIDOU, I.-G. *Variable density shallow flow model for flood simulation*. PhD thesis, University of Oxford, 2011.
- [28] CREED, M., APOSTOLIDOU, I.-G., TAYLOR, P., et al. “A finite volume shock-capturing solver of the fully coupled shallow water-sediment equations”, *International Journal for Numerical Methods in Fluids*, 2016. ISSN: 1097-0363. doi: 10.1002/flid.4359. Fld.4359.
- [29] ELFIMOV, V. I., KHAKZAD, H. “Evaluation of flow regime of turbidity currents entering Dez Reservoir using extended shallow water model”, *Water Science and Engineering*, v. 7, n. 3, pp. 267–276, 2014. ISSN: 1674-2370. doi: 10.3882/j.issn.1674-2370.2014.03.003.
- [30] DE LUNA, T. M., CASTRO-DÍAZ, M. J., MADROÑAL, C. P., et al. “On a shallow water model for the simulation of turbidity currents”, *Communications in Computational Physics*, pp. 848–882, 2009.
- [31] GROENENBERG, R. M., SLOFF, K., WELTJE, G. J. “A high-resolution 2-DH numerical scheme for process-based modeling of 3-D turbidite fan stratigraphy”, *Computers & Geosciences*, v. 35, n. 8, pp. 1686–1700, 2009. ISSN: 0098-3004. doi: 10.1016/j.cageo.2009.01.004.
- [32] GROENENBERG, R. M. *Process-based modelling of turbidity-current hydrodynamics and sedimentation*. PhD thesis, Delft University of Technology, 2007.
- [33] HOU, J., LIANG, Q., ZHANG, H., et al. “An efficient unstructured MUSCL scheme for solving the 2D shallow water equations”, *Environ. Model. Softw.*, v. 66, n. C, pp. 131–152, Apr. 2015. ISSN: 1364-8152. doi: 10.1016/j.envsoft.2014.12.007.

- [34] AMBATI, V. R., BOKHOVE, O. “Space–time discontinuous Galerkin finite element method for shallow water flows”, *Journal of Computational and Applied Mathematics*, v. 204, n. 2, pp. 452–462, 2007. ISSN: 0377-0427. doi: 10.1016/j.cam.2006.01.047. Special Issue: The Seventh International Conference on Mathematical and Numerical Aspects of Waves (WAVES’05).
- [35] HERVOUET, J.-M. *Hydrodynamics of free surface flows: modelling with the finite element method*. Chichester, UK, John Wiley & Sons, 2007. ISBN: 0470319623.
- [36] CASTRO, M. J., GARCÍA-RODRÍGUEZ, J. A., GONZÁLEZ-VIDA, J. M., et al. “Improved FVM for two-layer shallow-water models: Application to the strait of Gibraltar”, *Advances in Engineering Software*, v. 38, n. 6, pp. 386–398, 2007. ISSN: 0965-9978. doi: 10.1016/j.advengsoft.2006.09.012.
- [37] SANTOS, T. L., COUTINHO, A. L. G. A. “A continuous finite element approach to the well-balanced shallow water equations”, *CILAMCE 2017 – XXXVIII Ibero-Latin American Congress on Computational Methods in Engineering*, Nov. 2017. ISSN: 2178-4949. doi: 10.20906/CPS/CILAMCE2017-0712.
- [38] BEHZADI, F. *Solution of fully-coupled shallow water equations and contaminant transport using a primitive variable Riemann solver and a semi-discrete SUPG method*. PhD thesis, University of Tennessee, 2016.
- [39] CASTRO, R. S. “Space-time finite element formulation for shallow water equations with shock-capturing operator”, *Pesquimat*, v. 3, n. 1, 2014. ISSN: 1609-8439.
- [40] TAKASE, S., KASHIYAMA, K., TANAKA, S., et al. “Space-time SUPG finite element computation of shallow-water flows with moving shorelines”, *Computational Mechanics*, v. 48, n. 3, pp. 293–306, 2011. ISSN: 0178-7675. doi: 10.1007/s00466-011-0618-1.
- [41] HUGHES, T. J. R., MALLET, M. “A new finite element formulation for computational fluid dynamics: III. The generalized streamline operator for multidimensional advective-diffusive systems”, *Computer Methods in Applied Mechanics and Engineering*, v. 58, n. 3, pp. 305–328, 1986. ISSN: 0045-7825. doi: 10.1016/0045-7825(86)90152-0.
- [42] GALEÃO, A. C., DO CARMO, E. G. D. “A consistent approximate upwind Petrov-Galerkin method for convection-dominated problems”, *Computer*

Methods in Applied Mechanics and Engineering, v. 68, n. 1, pp. 83 – 95, 1988. ISSN: 0045-7825. doi: 10.1016/0045-7825(88)90108-9.

- [43] ALMEIDA, R. C., GALEÃO, A. C. “An adaptive Petrov-Galerkin formulation for the compressible Euler and Navier-Stokes equations”, *Computer Methods in Applied Mechanics and Engineering*, v. 129, n. 1-2, pp. 157–176, 1996. ISSN: 0045-7825. doi: 10.1016/0045-7825(95)00858-6.
- [44] BORIS, J. P., BOOK, D. L. “Flux-corrected transport. I. SHASTA, a fluid transport algorithm that works”, *Journal of Computational Physics*, v. 11, n. 1, pp. 38–69, 1973. ISSN: 0021-9991. doi: 10.1016/0021-9991(73)90147-2.
- [45] KUZMIN, D., MÖLLER, M., TUREK, S. “Multidimensional FEM-FCT schemes for arbitrary time stepping”, *International Journal for Numerical Methods in Fluids*, v. 42, n. 3, pp. 265–295, 2003. doi: 10.1002/flid.493.
- [46] SHEU, T. W. H., FANG, C. C. “High resolution finite-element analysis of shallow water equations in two dimensions”, *Computer Methods in Applied Mechanics and Engineering*, v. 190, n. 20–21, pp. 2581–2601, 2001. ISSN: 0045-7825. doi: 10.1016/S0045-7825(00)00255-3.
- [47] ORTIZ, P., ANGUIA, J., RIVEIRO, M. “Free surface flows over partially erodible beds by a continuous finite element method”, *Environmental Earth Sciences*, v. 74, n. 11, pp. 7357–7370, Dec. 2015. ISSN: 1866-6299. doi: 10.1007/s12665-015-4730-y.
- [48] KUZMIN, D. “Explicit and implicit FEM-FCT algorithms with flux linearization”, *Journal of Computational Physics*, v. 228, n. 7, pp. 2517–2534, 2009. ISSN: 0021-9991. doi: 10.1016/j.jcp.2008.12.011.
- [49] LÖHNER, R., MORGAN, K., PERAIRE, J., et al. “Finite element flux-corrected transport (FEM-FCT) for the euler and Navier-Stokes equations”, *International Journal for Numerical Methods in Fluids*, v. 7, n. 10, pp. 1093–1109, 1987. doi: 10.1002/flid.1650071007.
- [50] ZALESKAK, S. T. “Fully multidimensional flux-corrected transport algorithms for fluids”, *Journal of Computational Physics*, v. 31, n. 3, pp. 335–362, 1979. ISSN: 0021-9991. doi: 10.1016/0021-9991(79)90051-2.
- [51] TEZDUYAR, T. E. “Finite element methods for fluid dynamics with moving boundaries and interfaces”. In: *Encyclopedia of Computational Mechanics*, cap. 17, John Wiley & Sons, Ltd, 2004. ISBN: 9780470091357. doi: 10.1002/0470091355.ecm069.

- [52] TAKASE, S., KASHIYAMA, K., TANAKA, S., et al. “Space-time SUPG formulation of the shallow-water equations”, *International Journal for Numerical Methods in Fluids*, v. 64, n. 10-12, pp. 1379–1394, 2010.
- [53] RISPOLI, F., SAAVEDRA, R., CORSINI, A., et al. “Computation of inviscid compressible flows with the V-SGS stabilization and $YZ\beta$ shock-capturing”, *International Journal for Numerical Methods in Fluids*, v. 54, n. 6-8, pp. 695–706, 2007. ISSN: 0271-2091. doi: 10.1002/flid.1447.
- [54] TEZDUYAR, T. E., SENGA, M. “SUPG finite element computation of inviscid supersonic flows with $YZ\beta$ shock-Capturing”, *Computers & Fluids*, v. 36, n. 1, pp. 147 – 159, 2007. ISSN: 0045-7930. doi: 10.1016/j.compfluid.2005.07.009. Challenges and Advances in Flow Simulation and Modeling.
- [55] BANGERTH, W., HARTMANN, R., KANSCHAT, G. “deal.II – a general-purpose object-oriented finite element library”, *ACM Transactions on Mathematical Software*, v. 33, n. 4, Aug. 2007. ISSN: 0098-3500. doi: 10.1145/1268776.1268779.
- [56] CARNIELLO, L., DEFINA, A., D’ALPAOS, L. “Modeling sand-mud transport induced by tidal currents and wind waves in shallow microtidal basins: Application to the Venice lagoon (Italy)”, *Estuarine, Coastal and Shelf Science*, v. 102, pp. 105–115, 2012. ISSN: 0272-7714. doi: 10.1016/j.ecss.2012.03.016.
- [57] GOURGUE, O., COMBLEN, R., LAMBRECHTS, J., et al. “A flux-limiting wetting–drying method for finite-element shallow-water models, with application to the Scheldt Estuary”, *Advances in Water Resources*, v. 32, n. 12, pp. 1726–1739, 2009. ISSN: 0309-1708. doi: 10.1016/j.advwatres.2009.09.005.
- [58] ZOPPOU, C., ROBERTS, S. “Numerical solution of the two-dimensional unsteady dam break”, *Applied Mathematical Modelling*, v. 24, n. 7, pp. 457–475, 2000. ISSN: 0307-904X. doi: 10.1016/S0307-904X(99)00056-6.
- [59] VACONDIO, R., PALÙ, A. D., MIGNOSA, P. “GPU-enhanced finite volume shallow water solver for fast flood simulations”, *Environmental Modelling & Software*, v. 57, pp. 60–75, 2014. ISSN: 1364-8152. doi: 10.1016/j.envsoft.2014.02.003.
- [60] KIM, J., LEVEQUE, R. J. “Two-layer shallow water system and its applications”. In: *Proceedings of the Twelfth International Conference on Hyperbolic Problems, Maryland*, 2008.

- [61] ABGRALL, R., KARNI, S. “Two-layer shallow water system: A relaxation approach”, *SIAM Journal on Scientific Computing*, v. 31, n. 3, pp. 1603–1627, 2009. doi: 10.1137/06067167X.
- [62] PETCU, M., TEMAM, R. “An interface problem: The two-layer shallow water equations”, *Discrete and Continuous Dynamical Systems*, v. 33, n. 11/12, pp. 5327–5345, 2013. ISSN: 1078-0947. doi: 10.3934/dcds.2013.33.5327.
- [63] TAN, W. *Shallow water hydrodynamics: mathematical theory and numerical solution for a two-dimensional system of shallow-water equations*. Elsevier Oceanography Series. Amsterdam, The Netherlands, Elsevier Science, 1992. ISBN: 9780080870939.
- [64] SLOBODCICOV, I. *Implementação em paralelo do método dos elementos finitos para as equações de águas rasas*. Master’s thesis, Civil Engineering Program, Federal University of Rio de Janeiro (UFRJ), 2003.
- [65] VREUGDENHIL, C. *Numerical methods for shallow-water flow*, v. 13, *Water Science and Technology Library*. 1 ed. Heidelberg, Germany, Springer Netherlands, 1994. ISBN: 9780792331643.
- [66] LE MÉHAUTÉ, B. “An introduction to hydrodynamics and water waves”, *Ocean. Front. Coast. Process*, 1969.
- [67] SOULSBY, R. *Dynamics of marine sands: a manual for practical applications*. Telford, 1997. ISBN: 9780727725844.
- [68] ARCEMENT, G. J., SCHNEIDER, V. R. *Guide for selecting Manning’s roughness coefficients for natural channels and flood plains*. US Government Printing Office Washington, DC, USA, 1989.
- [69] KUBO, Y., NAKAJIMA, T. “Laboratory experiments and numerical simulation of sediment-wave formation by turbidity currents”, *Marine Geology*, v. 192, n. 1, pp. 105–121, 2002. ISSN: 0025-3227. doi: 10.1016/S0025-3227(02)00551-0.
- [70] KOMAR, P. D. “The channelized flow of turbidity currents with application to Monterey deep-sea fan channel”, *Journal of Geophysical Research (1896-1977)*, v. 74, n. 18, pp. 4544–4558, 1969. doi: 10.1029/JC074i018p04544.
- [71] MIDDLETON, G. V. “Experiments on density and turbidity currents: II. Uniform flow of density currents.” *Canadian Journal of Earth Sciences*, v. 3, n. 5, pp. 627–637, 1966. doi: 10.1139/e66-044.

- [72] BÁRCENAS, P., FERNÁNDEZ-SALAS, L. M., MACÍAS, J., et al. “Improvements on turbidity current models. Application to Adra river (southern Spain)”, *EGU General Assembly Conference Abstracts*, v. 14, pp. 8695, 2014.
- [73] BANGERTH, W., KAYSER-HEROLD, O. “Data structures and requirements for *hp* finite element software”, *ACM Trans. Math. Softw.*, v. 36, n. 1, pp. 4:1–4:31, Mar. 2009. ISSN: 0098-3500. doi: 10.1145/1486525.1486529.
- [74] HUGHES, T. J., MALLET, M., AKIRA, M. “A new finite element formulation for computational fluid dynamics: II. Beyond SUPG”, *Computer Methods in Applied Mechanics and Engineering*, v. 54, n. 3, pp. 341–355, 1986. ISSN: 0045-7825. doi: 10.1016/0045-7825(86)90110-6.
- [75] TADMOR, E., ZHONG, W. “Energy-preserving and stable approximations for the two-dimensional shallow water equations”. In: *Mathematics and Computation, a Contemporary View: The Abel Symposium 2006 Proceedings of the Third Abel Symposium, Alesund, Norway, May 25–27, 2006*, pp. 67–94, Springer Berlin Heidelberg, 2008. ISBN: 978-3-540-68850-1. doi: 10.1007/978-3-540-68850-1_4.
- [76] TEN EIKELDER, M. F. P., BAZILEVS, Y., AKKERMAN, I. “A theoretical framework for discontinuity capturing: joining variational multiscale analysis and variation entropy theory”, *Computer Methods in Applied Mechanics and Engineering*, v. 359, pp. 112664, 2020. ISSN: 0045-7825. doi: <https://doi.org/10.1016/j.cma.2019.112664>.
- [77] ALIABADI, S. K., TEZDUYAR, T. E. “Parallel fluid dynamics computations in aerospace applications”, *International Journal for Numerical Methods in Fluids*, v. 21, n. 10, pp. 783–805, 1995. ISSN: 1097-0363. doi: 10.1002/flid.1650211003.
- [78] SAAD, Y., SCHULTZ, M. H. “GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems”, *SIAM J. Sci. Stat. Comput.*, v. 7, n. 3, pp. 856–869, Jul. 1986. ISSN: 0196-5204. doi: 10.1137/0907058.
- [79] CUTHILL, E., MCKEE, J. “Reducing the bandwidth of sparse symmetric matrices”. In: *Proceedings of the 1969 24th National Conference*, ACM ’69, pp. 157–172, New York, NY, USA, 1969. ACM. doi: 10.1145/800195.805928.

- [80] HANSBO, P. “Aspects of conservation in finite element flow computations”, *Computer Methods in Applied Mechanics and Engineering*, v. 117, n. 3, pp. 423–437, 1994. ISSN: 0045-7825. doi: [https://doi.org/10.1016/0045-7825\(94\)90127-9](https://doi.org/10.1016/0045-7825(94)90127-9).
- [81] JAMESON, A. “Computational algorithms for aerodynamic analysis and design”, *Applied Numerical Mathematics*, v. 13, n. 5, pp. 383 – 422, 1993. ISSN: 0168-9274. doi: [https://doi.org/10.1016/0168-9274\(93\)90096-A](https://doi.org/10.1016/0168-9274(93)90096-A).
- [82] QUARTERONI, A., VALLI, A. “Numerical solution of linear systems”. In: *Numerical Approximation of Partial Differential Equations*, cap. 2, pp. 17–71, Springer Berlin Heidelberg, 1994. ISBN: 978-3-540-85268-1. doi: [10.1007/978-3-540-85268-1_2](https://doi.org/10.1007/978-3-540-85268-1_2).
- [83] KUZMIN, D., MÖLLER, M., TUREK, S. “High-resolution FEM–FCT schemes for multidimensional conservation laws”, *Computer Methods in Applied Mechanics and Engineering*, v. 193, n. 45, pp. 4915 – 4946, 2004. ISSN: 0045-7825. doi: <https://doi.org/10.1016/j.cma.2004.05.009>.
- [84] KUZMIN, D., MÖLLER, M. “Algebraic flux correction II. Compressible Euler equations”. In: *Flux-Corrected Transport: Principles, Algorithms, and Applications*, pp. 207–250, Springer Berlin Heidelberg, 2005. ISBN: 978-3-540-27206-9. doi: [10.1007/3-540-27206-2_7](https://doi.org/10.1007/3-540-27206-2_7).
- [85] ELSNER, L., MEHRMANN, V. “Convergence of block iterative methods for linear systems arising in the numerical solution of Euler equations”, *Numerische Mathematik*, v. 59, n. 1, pp. 541–559, Dec 1991. ISSN: 0945-3245. doi: [10.1007/BF01385795](https://doi.org/10.1007/BF01385795).
- [86] YIP, E. L. “A necessary and sufficient condition for M-matrices and its relation to block LU factorization”, *Linear Algebra and its Applications*, v. 235, pp. 261–274, 1996. ISSN: 0024-3795. doi: [10.1016/0024-3795\(94\)00164-2](https://doi.org/10.1016/0024-3795(94)00164-2).
- [87] GUERMOND, J.-L., DE LUNA, M. Q., POPOV, B., et al. “Well-balanced second-order finite element approximation of the shallow water equations with friction”, *SIAM Journal on Scientific Computing*, v. 40, n. 6, pp. A3873–A3901, 2018. doi: [10.1137/17M1156162](https://doi.org/10.1137/17M1156162).
- [88] AUDUSSE, E., BOUCHUT, F., BRISTEAU, M.-O., et al. “A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows”, *SIAM Journal on Scientific Computing*, v. 25, n. 6, pp. 2050–2065, June 2004. ISSN: 1064-8275. doi: [10.1137/S1064827503431090](https://doi.org/10.1137/S1064827503431090).

- [89] KUZMIN, D., MÖLLER, M., SHADID, J. N., et al. “Failsafe flux limiting and constrained data projections for equations of gas dynamics”, *Journal of Computational Physics*, v. 229, n. 23, pp. 8766–8779, 2010. ISSN: 0021-9991. doi: 10.1016/j.jcp.2010.08.009.
- [90] BASTING, M., KUZMIN, D. “An FCT finite element scheme for ideal MHD equations in 1D and 2D”, *Journal of Computational Physics*, v. 338, pp. 585–605, 2017. ISSN: 0021-9991. doi: 10.1016/j.jcp.2017.02.051.
- [91] LIAN, C., XIA, G., MERKLE, C. L. “Impact of source terms on reliability of CFD algorithms”, *Computers & Fluids*, v. 39, n. 10, pp. 1909–1922, 2010. ISSN: 0045-7930. doi: <https://doi.org/10.1016/j.compfluid.2010.06.021>.
- [92] PATANKAR, S. V. *Numerical heat transfer and fluid flow*. CRC press, 1980. doi: 10.1201/9781482234213.
- [93] BRUFAU, P., GARCÍA-NAVARRO, P. “Unsteady free surface flow simulation over complex topography with a multidimensional upwind technique”, *Journal of Computational Physics*, v. 186, n. 2, pp. 503–526, 2003. ISSN: 0021-9991. doi: 10.1016/S0021-9991(03)00072-X.
- [94] ZHENG, L., CHEN, C., LIU, H. “A modeling study of the Satilla river estuary, Georgia. I: Flooding-drying process and water exchange over the salt marsh-estuary-shelf complex”, *Estuaries*, v. 26, n. 3, pp. 651–669, 2003. ISSN: 0160-8347. doi: 10.1007/BF02711977.
- [95] RICCHIUTO, M., BOLLERMANN, A. “Stabilized residual distribution for shallow water simulations”, *Journal of Computational Physics*, v. 228, n. 4, pp. 1071–1115, 2009. ISSN: 0021-9991. doi: 10.1016/j.jcp.2008.10.020.
- [96] KESSERWANI, G., LIANG, Q. “Well-balanced RKDG2 solutions to the shallow water equations over irregular domains with wetting and drying”, *Computers & Fluids*, v. 39, n. 10, pp. 2040–2050, 2010. ISSN: 0045-7930. doi: 10.1016/j.compfluid.2010.07.008.
- [97] JIANG, Y. W., WAI, O. W. H. “Drying–wetting approach for 3D finite element sigma coordinate model for estuaries with large tidal flats”, *Advances in Water Resources*, v. 28, n. 8, pp. 779–792, 2005. ISSN: 0309-1708. doi: 10.1016/j.advwatres.2005.02.004.
- [98] LIANG, Q., BORTHWICK, A. G. L. “Adaptive quadtree simulation of shallow flows with wet–dry fronts over complex topography”, *Computers & Fluids*,

- v. 38, n. 2, pp. 221–234, 2009. ISSN: 0045-7930. doi: 10.1016/j.compfluid.2008.02.008.
- [99] LIANG, Q., MARCHE, F. “Numerical resolution of well-balanced shallow water equations with complex source terms”, *Advances in Water Resources*, v. 32, n. 6, pp. 873–884, 2009. ISSN: 0309-1708. doi: 10.1016/j.advwatres.2009.02.010.
- [100] WANG, B., FRINGER, O. B., GIDDINGS, S. N., et al. “High-resolution simulations of a macrotidal estuary using SUNTANS”, *Ocean Modelling*, v. 26, n. 1, pp. 60–85, 2009. ISSN: 1463-5003. doi: 10.1016/j.ocemod.2008.08.006.
- [101] HENICHE, M., SECRETAN, Y., BOUDREAU, P., et al. “A two-dimensional finite element drying-wetting shallow water model for rivers and estuaries”, *Advances in Water Resources*, v. 23, n. 4, pp. 359–372, 2000. ISSN: 0309-1708. doi: 10.1016/S0309-1708(99)00031-7.
- [102] MULDER, T., HÜNEKE, H., LOON, A. J. V. “Chapter 1 - Progress in deep-sea sedimentology”. In: HüNeke, H., Mulder, T. (Eds.), *Deep-Sea Sediments*, v. 63, *Developments in Sedimentology*, Elsevier, pp. 1–24, 2011. doi: 10.1016/B978-0-444-53000-4.00001-9.
- [103] DRITSCHEL, D. G. “Contour surgery: a topological reconnection scheme for extended integrations using contour dynamics”, *Journal of Computational Physics*, v. 77, n. 1, pp. 240–266, 1988. ISSN: 0021-9991. doi: 10.1016/0021-9991(88)90165-9.
- [104] MILLS, P. “Isoline retrieval: an optimal sounding method for validation of advected contours”, *Computers & Geosciences*, v. 35, n. 10, pp. 2020–2031, 2009. doi: 10.1016/j.cageo.2008.12.015.
- [105] WILLIAMS, D. J., SHAH, M. “A fast algorithm for active contours and curvature estimation”, *CVGIP: Image Understanding*, v. 55, n. 1, pp. 14–26, 1992. ISSN: 1049-9660. doi: 10.1016/1049-9660(92)90003-L.
- [106] BARRY, P. J., GOLDMAN, R. N. “A recursive evaluation algorithm for a class of Catmull-Rom splines”, *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 199–204, 1988. doi: 10.1145/54852.378511.
- [107] YUKSEL, C., SCHAEFER, S., KEYSER, J. “On the parameterization of Catmull-Rom curves”, *2009 SIAM/ACM Joint Conference on Geometric and Physical Modeling*, pp. 47–53, 2009. doi: 10.1145/1629255.1629262.

- [108] GEUZAINÉ, C., REMACLE, J.-F. “Gmsh: a 3-D finite element mesh generator with built-in pre- and post-processing facilities”, *Int. J. Numer. Meth. Engng.*, v. 79, n. 11, pp. 1309–1331, Sept. 2009. ISSN: 0029-5981. doi: 10.1002/nme.2579.
- [109] LÓPEZ, G. I. “Walther’s law of facies”. In: *Encyclopedia of Scientific Dating Methods*, pp. 1–2, Springer Netherlands, 2013. ISBN: 978-94-007-6326-5. doi: 10.1007/978-94-007-6326-5_30-1.
- [110] WAGONER, J. C. V., MITCHUM, R. M., CAMPION, K. M., et al. *Siliciclastic sequence stratigraphy in well logs, cores, and outcrops: concepts for high-resolution correlation of time and facies*. American Association of Petroleum Geologists, 01 1990. ISBN: 9781629810164. doi: 10.1306/Mth7510.
- [111] BOUMA, A. H., KUENEN, P. H., SHEPARD, F. P. *Sedimentology of some Flysch deposits : a graphic approach to facies interpretation*. Amsterdam, The Netherlands, Amsterdam : Elsevier, 1962.
- [112] MACHADO, L. C. R., KOWSMANN, R. O., ALMEIDA-JR, W., et al. “Geometria da porção proximal do sistema deposicional turbidítico moderno da formação Carapebus, Bacia de Campos: modelo para heterogeneidades de reservatório”, *Boletim de Geociências da Petrobras*, v. 12, n. 2, pp. 287–315, 2004. Available at: https://www.researchgate.net/publication/285769176_Geometria_da_porcao_proximal_do_sistema_deposicional_turbiditico_moderno_da_Formacao_Carapebus_Bacia_de_Campos_Modelo_para_heterogeneidades_de_reservatorio. Accessed on: 1 mar. 2020.
- [113] HAQ, B. U., HARDENBOL, J., VAIL, P. R. “Chronology of fluctuating sea levels since the triassic”, *Science*, v. 235, n. 4793, pp. 1156–1167, 1987. ISSN: 0036-8075. doi: 10.1126/science.235.4793.1156.
- [114] RAYMOND, A. S. *Stratigraphic sedimentary inversion using paths in graphs*. Master’s thesis, Computer and Systems Engineering Program, Federal University of Rio de Janeiro (UFRJ), Mar. 2017.
- [115] GILES, M. R., INDRELID, S. L., JAMES, D. M. D. “Compaction — the great unknown in basin modelling”, *Geological Society, London, Special Publications*, v. 141, n. 1, pp. 15–43, 1998. ISSN: 0305-8719. doi: 10.1144/GSL.SP.1998.141.01.02.

- [116] HANTSCHHEL, T., KAUERAUF, A. I. “Pore pressure, compaction and tectonics”. In: *Fundamentals of Basin and Petroleum Systems Modeling*, cap. 2, pp. 31–101, Springer Berlin Heidelberg, 2009. ISBN: 978-3-540-72318-9. doi: 10.1007/978-3-540-72318-9_2.
- [117] BANGERTH, W., BURSTEDDE, C., HEISTER, T., et al. “Algorithms and data structures for massively parallel generic adaptive finite element codes”, *ACM Transactions on Mathematical Software*, v. 38, n. 2, Jan. 2012. ISSN: 0098-3500. doi: 10.1145/2049673.2049678.
- [118] SCHROEDER, W. *The visualization toolkit : an object-oriented approach to 3D graphics*. Clifton Park, N.Y, Kitware, 2006. ISBN: 9781930934191.
- [119] AHRENS, J., GEVECI, B., LAW, C. “ParaView: an end-user tool for large-data visualization”. In: Hansen, C. D., Johnson, C. R. (Eds.), *Visualization Handbook*, Butterworth-Heinemann, pp. 717–731, 2005. ISBN: 978-0-12-387582-2. doi: 10.1016/B978-012387582-2/50038-1.
- [120] ACHILLEOS, G. “The inverse distance weighted interpolation method and error propagation mechanism – creating a DEM from an analogue topographical map”, *Journal of Spatial Science*, v. 56, pp. 283–304, 12 2011. doi: 10.1080/14498596.2011.623348.
- [121] STOKER, J. J. *Water waves: The mathematical theory with applications*, v. 36. Chichester, UK, John Wiley & Sons, Inc., 1992. ISBN: 9781118033159.
- [122] FENNEMA, R. J., CHAUDHRY, M. H. “Explicit methods for 2-D transient free surface flows”, *Journal of Hydraulic Engineering*, v. 116, n. 8, pp. 1013–1034, 1990. doi: 10.1061/(ASCE)0733-9429(1990)116:8(1013).
- [123] DELESTRE, O., LUCAS, C., KSINANT, P.-A., et al. “SWASHES: a compilation of shallow water analytic solutions for hydraulic and environmental studies”, *International Journal for Numerical Methods in Fluids*, v. 72, pp. 269–300, May 2013. doi: 10.1002/flid.3741.
- [124] EATON, J. W., BATEMAN, D., HAUBERG, S., et al. “GNU Octave version 4.0.0 manual: a high-level interactive language for numerical computations”. 2015. Available at: <<http://www.gnu.org/software/octave/doc/interpreter>>. Accessed on: 1 mar. 2020.
- [125] GEORGE, D. L. *Numerical approximation of the nonlinear shallow water equations with topography and dry beds: a Godunov-type scheme*. Master’s thesis, University of Washington, 2004.

- [126] VALIANI, A., CALEFFI, V., ZANNI, A. “Finite volume scheme for 2D shallow-water equations. Application to the Malpasset dam-break”, *The 4th CADAM Workshop, Zaragoza*, pp. 63–94, 1999.
- [127] SEAÏD, M. “Non-oscillatory relaxation methods for the shallow-water equations in one and two space dimensions”, *International Journal for Numerical Methods in Fluids*, v. 46, n. 5, pp. 457–484, 2004. doi: 10.1002/fld.766.
- [128] RICCHIUTO, M., ABGRALL, R., DECONINCK, H. “Application of conservative residual distribution schemes to the solution of the shallow water equations on unstructured meshes”, *Journal of Computational Physics*, v. 222, n. 1, pp. 287–331, 2007. ISSN: 0021-9991. doi: 10.1016/j.jcp.2006.06.024.
- [129] RICCHIUTO, M. “An explicit residual based approach for shallow water flows”, *Journal of Computational Physics*, v. 280, pp. 306–344, 2015. ISSN: 0021-9991. doi: 10.1016/j.jcp.2014.09.027.
- [130] KAWAHARA, M., UMETSU, T. “Finite element method for moving boundary problems in river flow”, *International Journal for Numerical Methods in Fluids*, v. 6, n. 6, pp. 365–386, 1986. ISSN: 1097-0363. doi: 10.1002/fld.1650060605.
- [131] CAMATA, J. J., ELIAS, R. N., COUTINHO, A. L. G. A. “FEM simulation of coupled flow and bed morphodynamic interactions due to sediment transport phenomena”, *Journal of Computational Science and Technology*, v. 7, pp. 306–321, 2013. doi: 10.1299/jcst.7.306.
- [132] BLAISE, S., DELEERSNIJDER, E., WHITE, L., et al. “Influence of the turbulence closure scheme on the finite-element simulation of the upwelling in the wake of a shallow-water island”, *Continental Shelf Research*, v. 27, n. 18, pp. 2329–2345, 2007. ISSN: 0278-4343. doi: 10.1016/j.csr.2007.06.003.
- [133] HJULSTRÖM, F. *Studies of the morphological activity of rivers as illustrated by the river Fyris*. Bulletin of geological institution, University of Uppsala. Almqvist & Wiksells, 1935.
- [134] OSTAPENKO, V. V. “Numerical simulation of wave flows caused by a shore-side landslide”, *Journal of Applied Mechanics and Technical Physics*, v. 40, n. 4, pp. 647–654, 1999. ISSN: 1573-8620. doi: 10.1007/BF02468439.

- [135] JESCHKE, A., PEDERSEN, G. K., VATER, S., et al. “Depth-averaged non-hydrostatic extension for shallow water equations with quadratic vertical pressure profile: equivalence to boussinesq-type equations”, *International Journal for Numerical Methods in Fluids*, 2017. ISSN: 1097-0363. doi: 10.1002/fld.4361. Fld.4361.
- [136] STORTI, M. A., NIGRO, N. M., PAZ, R. R., et al. “Dynamic boundary conditions in computational fluid dynamics”, *Computer Methods in Applied Mechanics and Engineering*, v. 197, n. 13, pp. 1219–1232, 2008. ISSN: 0045-7825. doi: 10.1016/j.cma.2007.10.014.
- [137] PAZ, R. R., STORTI, M. A., GARELLI, L. “Absorbing boundary condition for nonlinear hyperbolic partial differential equations with unknown Riemann invariants”, *Mecánica Computacional*, v. 28, pp. 1593–1620, Nov. 2009.
- [138] HUGHES, T., TEZDUYAR, T. “Finite element methods for first-order hyperbolic systems with particular emphasis on the compressible euler equations”, *Computer Methods in Applied Mechanics and Engineering*, v. 45, n. 1–3, pp. 217–284, 1984. ISSN: 0045-7825. doi: 10.1016/0045-7825(84)90157-9.

Appendix A

Useful Definitions

In this chapter, we present some definitions that are useful to improve the understanding of certain parts of the thesis.

A.1 Derivatives at the Flux and Transformation Jacobian Matrices

To give the reader the gist of how to compute the derivatives with respect to U that define the Jacobian fluxes presented in Section 2.1.6 and the inverse Jacobian matrix of the transformation from entropy to conservation variables introduced in Section 2.3, we demonstrate the derivation of some of the required terms. Altogether, the crucial point is not to forget that the depth-averaged velocities \bar{u} and \bar{v} depend on h . Therefore, we exemplify:

$$\frac{\partial \bar{u}}{\partial h} = \frac{\partial}{\partial h} \left(\frac{1}{h} \int_{z_b}^{\eta} u dz \right) = -\frac{1}{h^2} \int_{z_b}^{\eta} u dz = -\frac{\bar{u}}{h}, \quad (\text{A.1})$$

$$\frac{\partial \bar{u}}{\partial(h\bar{u})} = \frac{\partial \left(\frac{1}{h} \int_{z_b}^{\eta} u dz \right)}{\partial \left(\int_{z_b}^{\eta} u dz \right)} = \frac{1}{h}, \quad (\text{A.2})$$

$$\begin{aligned} \frac{\partial(h\bar{u}^2)}{\partial h} &= \frac{\partial}{\partial h} \left[\frac{1}{h} \left(\int_{z_b}^{\eta} u dz \right)^2 \right] = \left(\int_{z_b}^{\eta} u dz \right)^2 \frac{\partial h^{-1}}{\partial h} = \\ &= -\frac{1}{h^2} \left(\int_{z_b}^{\eta} u dz \right)^2 = -\bar{u}^2, \end{aligned} \quad (\text{A.3})$$

$$\frac{\partial(h\bar{u}^2)}{\partial(h\bar{u})} = \frac{\partial \left[h \left(\frac{1}{h} \int_{z_b}^{\eta} u dz \right)^2 \right]}{\partial \left[h \left(\frac{1}{h} \int_{z_b}^{\eta} u dz \right) \right]} = \frac{1}{h} \frac{\partial \left(\int_{z_b}^{\eta} u dz \right)^2}{\partial \left(\int_{z_b}^{\eta} u dz \right)} = \frac{2}{h} \int_{z_b}^{\eta} u dz = 2\bar{u}, \quad (\text{A.4})$$

$$\begin{aligned}\frac{\partial(h\bar{u}\bar{v})}{\partial h} &= \frac{\partial}{\partial h} \left[\frac{1}{h} \left(\int_{z_b}^{\eta} u dz \right) \left(\int_{z_b}^{\eta} v dz \right) \right] = \left(\int_{z_b}^{\eta} u dz \right) \left(\int_{z_b}^{\eta} v dz \right) \frac{\partial h^{-1}}{\partial h} = \\ &= -\frac{1}{h^2} \left(\int_{z_b}^{\eta} u dz \right) \left(\int_{z_b}^{\eta} v dz \right) = -\bar{u}\bar{v},\end{aligned}\tag{A.5}$$

$$\frac{\partial(h\bar{u}\bar{v})}{\partial(h\bar{u})} = \frac{\partial(h\bar{u})}{\partial(h\bar{u})}\bar{v} + h\bar{u} \frac{\partial(\bar{v})}{\partial(h\bar{u})} = \bar{v}.\tag{A.6}$$

A.2 Matrix Vectorization

The vectorization of a matrix is a linear transformation that converts it into a column vector. Let \mathbf{A} be a matrix of dimensions $m \times n$. Its vectorization, denoted by $\text{vec}(\mathbf{A})$ is the column vector of size mn obtained by the juxtaposition of all its columns. For example, if we have:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix},\tag{A.7}$$

the vectorized matrix is:

$$\text{vec}(\mathbf{A}) = \begin{bmatrix} a_{11} & a_{21} & a_{12} & a_{22} \end{bmatrix}^T.\tag{A.8}$$

The vectorization operation can be employed together with the Kronecker product - which is denoted by \otimes and introduced in Section A.3 - to express matrix multiplications as a linear transformation. In this sense, if we also consider the matrices \mathbf{B} and \mathbf{C} , whose respective dimensions are $n \times p$ and $p \times q$, the following equations are valid:

$$\text{vec}(\mathbf{AB}) = (\mathbf{I}_p \otimes \mathbf{A}) \text{vec}(\mathbf{B}) = (\mathbf{B}^T \otimes \mathbf{I}_m) \text{vec}(\mathbf{A}),\tag{A.9}$$

$$\begin{aligned}\text{vec}(\mathbf{ABC}) &= (\mathbf{C}^T \otimes \mathbf{A}) \text{vec}(\mathbf{B}) = (\mathbf{I}_q \otimes \mathbf{AB}) \text{vec}(\mathbf{C}) = \\ &= (\mathbf{C}^T \mathbf{B}^T \otimes \mathbf{I}_m) \text{vec}(\mathbf{A}),\end{aligned}\tag{A.10}$$

where \mathbf{I}_m denotes the m -th order identity matrix.

A.3 Kronecker Product

The Kronecker product between two matrices is an operation that results in a block matrix. It is closely related to the standard tensor product. Thus, using the index notation, initially consider the basis vectors $\mathbf{e}_a, \mathbf{e}_b, \dots, \mathbf{e}_y$ and \mathbf{e}_z , and the tensors $\mathbf{A} = a_{ab..l} \mathbf{e}_a \mathbf{e}_b \cdots \mathbf{e}_l$ and $\mathbf{B} = b_{mn..z} \mathbf{e}_m \mathbf{e}_n \cdots \mathbf{e}_z$. The tensor product between \mathbf{A} and

\mathbf{B} is given by:

$$\mathbf{A} \otimes \mathbf{B} = a_{ab..l} b_{mn..z} \mathbf{e}_a \mathbf{e}_b \cdots \mathbf{e}_z. \quad (\text{A.11})$$

If these tensors are of second-order, or, equivalently, if they represent matrices, we have $\mathbf{A} = a_{ij} \mathbf{e}_i \mathbf{e}_j$ and $\mathbf{B} = b_{kl} \mathbf{e}_k \mathbf{e}_l$, and hence:

$$\mathbf{A} \otimes \mathbf{B} = a_{ij} b_{kl} \mathbf{e}_i \mathbf{e}_j \mathbf{e}_k \mathbf{e}_l. \quad (\text{A.12})$$

The resulting tensor can be written in matrix notation if we employ the Kronecker product. If \mathbf{A} and \mathbf{B} are, respectively, matrices of dimensions $m \times n$ and $p \times q$, we get:

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11} \mathbf{B} & \cdots & a_{1m} \mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1} \mathbf{B} & \cdots & a_{mn} \mathbf{B} \end{bmatrix}. \quad (\text{A.13})$$

At this point, we remark that the Kronecker product is not strictly a tensor product. It provides the result of the operation on a standard basis so that it can be represented as a matrix. In the previous example, the product $\mathbf{A} \otimes \mathbf{B}$ should have returned a fourth-order tensor under the bases \mathbf{e}_i , \mathbf{e}_j , \mathbf{e}_k and \mathbf{e}_l . However, we obtained a second-order tensor (a matrix) with respect other two bases. To exemplify the use of the operator, we can consider the 2×2 matrices:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}, \quad (\text{A.14})$$

and compute:

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} & a_{12} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \\ a_{21} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} & a_{22} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} a_{11} b_{11} & a_{11} b_{12} & a_{12} b_{11} & a_{12} b_{12} \\ a_{11} b_{21} & a_{11} b_{22} & a_{12} b_{21} & a_{12} b_{22} \\ a_{21} b_{11} & a_{21} b_{12} & a_{22} b_{11} & a_{22} b_{12} \\ a_{21} b_{21} & a_{21} b_{22} & a_{22} b_{21} & a_{22} b_{22} \end{bmatrix}. \quad (\text{A.15})$$

At last, among the properties of the Kronecker product, we can list:

- Inverse matrix: $(\mathbf{A} \otimes \mathbf{B})^{-1} = \mathbf{A}^{-1} \otimes \mathbf{B}^{-1}. \quad (\text{A.16})$

- Transpose matrix: $(\mathbf{A} \otimes \mathbf{B})^T = \mathbf{A}^T \otimes \mathbf{B}^T. \quad (\text{A.17})$

- Determinant: For the square matrices \mathbf{A} and \mathbf{B} , of dimensions $m \times m$ and $p \times p$: $|\mathbf{A} \otimes \mathbf{B}| = |\mathbf{A}|^m |\mathbf{B}|^p. \quad (\text{A.18})$

Appendix B

Hypotheses Evaluations

This chapter comprises the evaluation of some hypotheses made in previous chapters of the thesis.

B.1 SPD Jacobian Matrix of the Transformation from Conservation to Entropy Variables

We show that the Jacobian matrix of the transformation from conservation to entropy variables is symmetric positive definite (SPD) for positive current heights, i.e., for $h > 0$. The matrix in question is:

$$\tilde{\mathbf{A}}_0^{-1} = \frac{1}{h} \begin{bmatrix} g'h + u^2 + v^2 & -u & -v \\ -u & 1 & 0 \\ -v & 0 & 1 \end{bmatrix}. \quad (\text{B.1})$$

We compute its eigenvalues λ_1 , λ_2 and λ_3 , which are the solutions of the equation:

$$\det(h\tilde{\mathbf{A}}_0^{-1} - h\lambda\mathbf{I}_3) = 0, \quad (\text{B.2})$$

where $\det(\dots)$ is the determinant operator. Thus, we solve:

$$\begin{vmatrix} (g'h + u^2 + v^2) - h\lambda & -u & -v \\ -u & 1 - h\lambda & 0 \\ -v & 0 & 1 - h\lambda \end{vmatrix} = 0, \\ \therefore (1 - h\lambda) [(1 - h\lambda)(g'h + u^2 + v^2 - h\lambda) - u^2 - v^2] = 0. \quad (\text{B.3})$$

So, if $(1 - h\lambda) = 0$, the equation holds for the eigenvalue:

$$\lambda_1 = \frac{1}{h}. \quad (\text{B.4})$$

Otherwise, we have:

$$(1 - h\lambda)(g'h + u^2 + v^2 - h\lambda) - u^2 - v^2 = 0$$

$$\therefore h\lambda^2 - \alpha\lambda + g'h = 0, \quad (\text{B.5})$$

with $\alpha = (g'h + u^2 + v^2 + h) > 0$. Solving the previous quadratic equation, we obtain the other two eigenvalues:

$$\lambda_2 = \frac{\alpha + \sqrt{\alpha^2 - 4g'h^2}}{2h} \quad \text{and} \quad \lambda_3 = \frac{\alpha - \sqrt{\alpha^2 - 4g'h^2}}{2h}. \quad (\text{B.6})$$

As $\tilde{\mathbf{A}}_0^{-1}$ is symmetric and all its entries are real-valued, the Spectral Theorem states that all its eigenvalues are also real numbers. This fact can be easily verified for a symmetric matrix \mathbf{S} . Let the superscript $*$ denote the element-wise complex conjugate function and take into account the eigenequation $\mathbf{S}\mathbf{v} = \lambda\mathbf{v}$. Then, note that $(\mathbf{S}\mathbf{v})^* = \mathbf{S}\mathbf{v}^* = \lambda^*\mathbf{v}^*$, since \mathbf{S} only has real entries. In this context, consider the following relation:

$$\lambda(\mathbf{v}^T\mathbf{v}^*) = (\lambda\mathbf{v})^T\mathbf{v}^* = (\mathbf{S}\mathbf{v})^T\mathbf{v}^* = \mathbf{v}^T\mathbf{S}^T\mathbf{v}^* = \mathbf{v}^T\mathbf{S}\mathbf{v}^* = \mathbf{v}^T\lambda^*\mathbf{v}^* = \lambda^*(\mathbf{v}^T\mathbf{v}^*)$$

$$\therefore \lambda = \lambda^*, \quad (\text{B.7})$$

which is only true if $\lambda \in \mathbb{R}$. As a result, $\alpha^2 - 4gh^2 \geq 0$ and, thus, λ_2 is a positive real value. Next, observing that $\sqrt{\alpha^2 - 4gh^2} \leq \sqrt{\alpha^2} \leq \alpha$, we can also see that $\lambda_3 > 0$. Therefore, all the matrix's eigenvalues are positive real values and hence $\tilde{\mathbf{A}}_0^{-1}$ is SPD for $h > 0$.

B.2 Mass-Preserving Lumped Mass Matrix

We show that we can replace the consistent mass matrix in Equation (2.102) with the row-sum lumped mass matrix while preserving the global mass. That is, we can rewrite:

$$\sum_j m_{ij} \frac{\partial v_j}{\partial t} = - \sum_j d_{ij} v_j + f_i \quad (\text{B.8})$$

as:

$$m_i \frac{\partial v_i}{\partial t} = - \sum_j d_{ij} v_j + f_i, \quad (\text{B.9})$$

where $m_i = \sum_j m_{ij}$.

Initially, we observe that the entries of the consistent mass matrix are defined by:

$$m_{ij} = \int_{\Omega} N_i N_j d\Omega, \quad (\text{B.10})$$

where N_i is the shape function associated with the i -th mesh node. For finite elements, we have $\sum_i N_i = 1$ at any point inside the domain Ω . In turn, the values of the lumped diagonal matrix are given by:

$$m_i = \sum_j m_{ij} = \sum_j \int_{\Omega} N_i N_j d\Omega = \int_{\Omega} N_i \left(\sum_j N_j \right) d\Omega = \int_{\Omega} N_i d\Omega. \quad (\text{B.11})$$

Therefore, the global mass is preserved when switching between the consistent and lumped mass matrices in the sense that:

$$\begin{aligned} \sum_i \sum_j m_{ij} \mathbf{v}_j &= \sum_i \sum_j \left(\int_{\Omega} N_i N_j d\Omega \right) \mathbf{v}_j = \sum_j \left[\int_{\Omega} \left(\sum_i N_i \right) N_j d\Omega \right] \mathbf{v}_j = \\ &= \sum_j \left(\int_{\Omega} N_j d\Omega \right) \mathbf{v}_j = \sum_j m_j \mathbf{v}_j = \sum_i m_i \mathbf{v}_i. \end{aligned} \quad (\text{B.12})$$

B.3 Zero Row-Sum Property of the Generalized Stiffness Matrix

We show that the generalized stiffness matrix:

$$\mathbf{D} = \int_{\Omega} [\mathbf{N}^T (\mathbf{A}^h \cdot \nabla) \mathbf{N} + \nabla \mathbf{N}^T \cdot (\mathbf{K} \nabla \mathbf{N})] d\Omega, \quad (\text{B.13})$$

defined in Section 2.2, has zero block-wise row sums.

However, we initially examine the simpler stiffness matrix of a scalar equation analogous to the multidimensional system (2.74). In this case, $\mathbf{N} = \{N_i\}$, $\mathbf{D} = \{d_{ij}\}$ and \mathbf{a}^h and k are the scalar problem equivalents to the \mathbf{A}^h advection matrix and the \mathbf{K} diffusivity matrix. We can perform row-wise sums of the advection contribution:

$$\sum_j \int_{\Omega} N_i (\mathbf{a}^h \cdot \nabla) N_j d\Omega = \int_{\Omega} N_i (\mathbf{a}^h \cdot \nabla) \sum_j N_j d\Omega = \int_{\Omega} N_i (\mathbf{a}^h \cdot \nabla) 1 d\Omega = 0, \quad (\text{B.14})$$

and of the diffusion component:

$$\sum_j \int_{\Omega} \nabla N_i \cdot (k \nabla N_j) d\Omega = \int_{\Omega} \nabla N_i \cdot \left[k \nabla \left(\sum_j N_j \right) \right] d\Omega = \int_{\Omega} \nabla N_i \cdot (k \nabla 1) d\Omega = 0, \quad (\text{B.15})$$

Together, Equations B.14 and B.15 prove that the stiffness matrix of the scalar problem has zero row sums, i.e., $\sum_j d_{ij} = 0$. Additionally, note that we cannot guarantee that \mathbf{D} has zero column sums because, unlike the diffusion contribution, the advection term does not meet this condition.

In sequence, going back to the original multidimensional problem, observe that $\mathbf{N} = \{\mathbf{N}_i\} = \{N_i \mathbf{I}_3\}$ and $\mathbf{D} = \{\mathbf{D}_{ij}\}$, where \mathbf{D}_{ij} are 3×3 matrices. We argue that $\sum_j \mathbf{D}_{ij} = \mathbf{0}$, $\forall i$. On these grounds, we sum the rows of the advection part:

$$\sum_j \int_{\Omega} \mathbf{N}_i(\mathbf{A}^h \cdot \nabla) \mathbf{N}_j d\Omega = \int_{\Omega} \mathbf{N}_i(\mathbf{A}^h \cdot \nabla) \sum_j \mathbf{N}_j d\Omega = \int_{\Omega} \mathbf{N}_i(\mathbf{A}^h \cdot \nabla) \mathbf{I}_3 d\Omega = \mathbf{0}, \quad (\text{B.16})$$

and of the diffusion contribution:

$$\begin{aligned} \sum_j \int_{\Omega} \nabla \mathbf{N}_i \cdot (\mathbf{K} \nabla \mathbf{N}_j) d\Omega &= \int_{\Omega} \nabla \mathbf{N}_i \cdot \left[\mathbf{K} \nabla \left(\sum_j \mathbf{N}_j \right) \right] d\Omega = \\ &= \int_{\Omega} \nabla \mathbf{N}_i \cdot (\mathbf{K} \nabla \mathbf{I}_3) d\Omega = \mathbf{0}, \end{aligned} \quad (\text{B.17})$$

showing that the matrix \mathbf{D} has zero block-wise row sums.

B.4 Non-Negative Diagonal Entries of Positive Definite Matrices

We show that positive definite matrices must have non-negative diagonal entries. By definition, a matrix $\mathbf{M} = \{m_{ij}\}$ is positive definite if:

$$\mathbf{v}^T \mathbf{M} \mathbf{v} > 0, \quad \forall \mathbf{v} \in \mathbb{R}^n \setminus \mathbf{0}. \quad (\text{B.18})$$

Now, suppose an arbitrary diagonal entry is non-positive, i.e., $m_{ii} \leq 0$ for a given i . Then, we can choose a vector \mathbf{v} whose values are zero, except for the i -th entry, which assumes a positive value. In this case, the product $\mathbf{v}^T \mathbf{M} \mathbf{v}$ is non-positive. As an example, for a 3×3 matrix \mathbf{M} , if $\mathbf{v} = [0 \ 1 \ 0]^T$ and $m_{22} = -1$:

$$\begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & -1 & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = -1 \leq 0. \quad (\text{B.19})$$

Therefore, as condition (B.18) must hold for all non-zero vectors \mathbf{v} , we conclude that positive definite matrices never have non-positive diagonal values. Otherwise, it is always possible to choose a vector \mathbf{v} that renders the product $\mathbf{v}^T \mathbf{M} \mathbf{v}$ non-positive. Similarly, the same rationale can be employed to show that negative definite matrices always have non-positive diagonal entries.

Appendix C

Parametric Streakline Curves

We define the parametric curves used to interpolate new streakline nodes, as introduced in Section 3.2. Consider the line segment $\overline{\mathbf{p}_1\mathbf{p}_2}$ that has adjacent points \mathbf{p}_0 and \mathbf{p}_3 , and is illustrated in Figure 3.1 . We approximate the continuous curve $f : \mathbb{R} \rightarrow \mathbb{R}^2$ between \mathbf{p}_1 and \mathbf{p}_2 with a centripetal Catmull-Rom spline [106], defined by:

$$f(s) = \frac{r_2 - r}{r_2 - r_1} \mathbf{b}_1 + \frac{r - r_1}{r_2 - r_1} \mathbf{b}_2, \quad \forall s \in [0, 1], \quad (\text{C.1})$$

where:

$$\begin{aligned} r &= (1 - s)r_1 + s r_2, & r_1 &= \|\mathbf{p}_1 - \mathbf{p}_0\|, \\ r_2 &= \|\mathbf{p}_2 - \mathbf{p}_1\| + r_1, & r_3 &= \|\mathbf{p}_3 - \mathbf{p}_2\| + r_2, \\ \mathbf{b}_1 &= \frac{r_2 - r}{r_2} \mathbf{a}_1 + \frac{r}{r_2} \mathbf{a}_2, & \mathbf{b}_2 &= \frac{r_3 - r}{r_3 - r_1} \mathbf{a}_2 + \frac{r - r_1}{r_3 - r_1} \mathbf{a}_3, \\ \mathbf{a}_1 &= \frac{r_1 - r}{r_1} \mathbf{p}_0 + \frac{r}{r_1} \mathbf{p}_1, & \mathbf{a}_2 &= \frac{r_2 - r}{r_2 - r_1} \mathbf{p}_1 + \frac{r - r_1}{r_2 - r_1} \mathbf{p}_2, \\ & & \mathbf{a}_3 &= \frac{r_3 - r}{r_3 - r_2} \mathbf{p}_2 + \frac{r - r_2}{r_3 - r_2} \mathbf{p}_3. \end{aligned} \quad (\text{C.2})$$

In the cases we linearly interpolate between \mathbf{p}_1 and \mathbf{p}_2 , we consider the function:

$$f(s) = \mathbf{p}_1 + s(\mathbf{p}_2 - \mathbf{p}_1), \quad \forall s \in [0, 1]. \quad (\text{C.3})$$

Do note that $f(0) = \mathbf{p}_1$ and $f(1) = \mathbf{p}_2$ in both curve types.

Appendix D

Predictor/Multi-corrector Algorithm

We present a deduction for the predictor/multi-corrector algorithm introduced by ALIABADI and TEZDUYAR [77]. For an analysis of the stability and precision of such algorithm, we refer to the work of HUGHES and TEZDUYAR [138]. We consider the semi-discrete dynamic system:

$$\mathbf{M}\mathbf{A}_c = -\mathbf{D}\mathbf{V} + \mathbf{F}, \quad (\text{D.1})$$

where \mathbf{V} contains the problem's unknowns whose time derivatives are denoted by \mathbf{A}_c . Also, \mathbf{M} is a generalized mass matrix and \mathbf{D} and \mathbf{F} , which might depend on \mathbf{V} , are, respectively, generalized stiffness and source matrices.

We employ the predictor/multi-corrector algorithm to iteratively advance the solution a time step of $\Delta t = t^{n+1} - t^n$ from its current state \mathbf{V}^n to the next \mathbf{V}^{n+1} . For each m -th iteration, with $m \in \{0, 1, 2, \dots\}$, it solves the equation:

$$\mathbf{M}\mathbf{A}_c^{(m+1)} = -\mathbf{D}^{(m)}\mathbf{V}^{(m+1)} + \mathbf{F}^{(m)}, \quad (\text{D.2})$$

obtaining the $(m+1)$ th solution. After the last iteration, we assign $\mathbf{V}^{n+1} = \mathbf{V}^{(m+1)}$. With this objective, first we use the generalized trapezoidal method to discretize $\partial\mathbf{V}/\partial t = \mathbf{A}_c$ between the current and next solution states:

$$\mathbf{V}^{n+1} = \mathbf{V}^n + (1 - \theta)\Delta t\mathbf{A}_c^n + \theta\Delta t\mathbf{A}_c^{n+1}, \quad (\text{D.3})$$

where $\theta = 0.5$ is a parameter that controls the stability and precision of the method. Then, we define \mathbf{A}_c^{n+1} as the sum of the consecutive corrections on \mathbf{A}_c , which are

performed in each iteration:

$$\mathbf{A}_c^{n+1} = \sum_m \Delta \mathbf{A}_c^{(m)}, \quad \Delta \mathbf{A}_c^{(m)} = \mathbf{A}_c^{(m+1)} - \mathbf{A}_c^{(m)}. \quad (\text{D.4})$$

Thus, in each iteration, its current value can be updated by:

$$\mathbf{A}_c^{(m+1)} = \mathbf{A}_c^{(m)} + \Delta \mathbf{A}_c^{(m)}. \quad (\text{D.5})$$

For consistency with (D.3), we define

$$\mathbf{A}_c^{(0)} = \mathbf{0}. \quad (\text{D.6})$$

Hence, we get:

$$\mathbf{V}^{n+1} = \mathbf{V}^n + (1 - \theta)\Delta t \mathbf{A}_c^n + \theta \Delta t \sum_m \Delta \mathbf{A}_c^{(m)}. \quad (\text{D.7})$$

If we define:

$$\mathbf{V}^{(0)} = \mathbf{V}^n + (1 - \theta)\Delta t \mathbf{A}_c^n, \quad (\text{D.8})$$

we can obtain an expression for the update on \mathbf{V} in each iteration:

$$\mathbf{V}^{(m+1)} = \mathbf{V}^{(m)} + \theta \Delta t \Delta \mathbf{A}_c^{(m)} = \mathbf{V}^{(m)} + \theta \Delta t \left(\mathbf{A}_c^{(m+1)} - \mathbf{A}_c^{(m)} \right). \quad (\text{D.9})$$

Then, from (D.2) and (D.9):

$$\begin{aligned} \mathbf{M} \mathbf{A}_c^{(m+1)} &= -\mathbf{D}^{(m)} \left[\mathbf{V}^{(m)} + \theta \Delta t \left(\mathbf{A}_c^{(m+1)} - \mathbf{A}_c^{(m)} \right) \right] + \mathbf{F}^{(m)} \\ \therefore \mathbf{M} \mathbf{A}_c^{(m+1)} &= -\mathbf{D}^{(m)} \mathbf{V}^{(m)} - \theta \Delta t \mathbf{D}^{(m)} \left(\mathbf{A}_c^{(m+1)} - \mathbf{A}_c^{(m)} \right) + \mathbf{F}^{(m)}. \end{aligned} \quad (\text{D.10})$$

Subtracting $\mathbf{M} \mathbf{A}_c^{(m)}$ from both sides of the equation:

$$\begin{aligned} \mathbf{M} \left(\mathbf{A}_c^{(m+1)} - \mathbf{A}_c^{(m)} \right) + \theta \Delta t \mathbf{D}^{(m)} \left(\mathbf{A}_c^{(m+1)} - \mathbf{A}_c^{(m)} \right) &= \\ &= - \left(\mathbf{M} \mathbf{A}_c^{(m)} + \mathbf{D}^{(m)} \mathbf{V}^{(m)} \right) + \mathbf{F}^{(m)} \\ \therefore \left(\mathbf{M} + \theta \Delta t \mathbf{D}^{(m)} \right) \left(\mathbf{A}_c^{(m+1)} - \mathbf{A}_c^{(m)} \right) &= - \left(\mathbf{M} \mathbf{A}_c^{(m)} + \mathbf{D}^{(m)} \mathbf{V}^{(m)} \right) + \mathbf{F}^{(m)} \\ \therefore \mathbf{M}^* \left(\Delta \mathbf{A}_c^{(m)} \right) &= \mathbf{R}^{(m)}, \end{aligned} \quad (\text{D.11})$$

with:

$$\mathbf{M}^* = \mathbf{M} + \theta \Delta t \mathbf{D}^{(m)}, \quad (\text{D.12})$$

$$\mathbf{R}^{(m)} = - \left(\mathbf{M} \mathbf{A}_c^{(m)} + \mathbf{D}^{(m)} \mathbf{V}^{(m)} \right) + \mathbf{F}^{(m)}, \quad (\text{D.13})$$

where \mathbf{M}^* is a effective mass matrix and $\mathbf{R}^{(m)}$ is the residual of (D.1) at the m -th iteration.

Therefore, in the prediction phase of the algorithm, we perform the initializations in (D.6) and (D.8). Then, during the correction phase, at each iteration, we assemble and solve (D.11) for $\Delta\mathbf{A}_c^{(m)}$. The result is used to update the current solution utilizing (D.5) and (D.9). A summary of the algorithm is presented in Algorithm 1, which is reproduced here for clarity.

Algorithm 1 Predictor/multi-corrector algorithm employed for the time integration in the stabilized method. Here we assign $\mathbf{A}_c = \partial\mathbf{V}/\partial t$.

- Prediction phase:
 - 1: $\mathbf{A}_c^{(0)} = \mathbf{0}$.
 - 2: $\mathbf{V}^{(0)} = \mathbf{V}^n + (1 - \theta)\Delta t\mathbf{A}_c^n$.
 - Correction phase
 - 3: **for** $m = 0, 1, 2, \dots$ until the convergence criteria is met, **do**:
 - 4: $\mathbf{R}^{(m)} = -(\mathbf{M}\mathbf{A}_c^{(m)} + \mathbf{D}^{(m)}\mathbf{V}^{(m)}) + \mathbf{F}^{(m)}$.
 - 5: $\mathbf{M}^* = \mathbf{M} + \theta\Delta t\mathbf{D}^{(m)}$.
 - 6: Solve $\mathbf{M}^*(\Delta\mathbf{A}_c^{(m)}) = \mathbf{R}^{(m)}$.
 - 7: $\mathbf{A}_c^{(m+1)} = \mathbf{A}_c^{(m)} + \Delta\mathbf{A}_c^{(m)}$.
 - 8: $\mathbf{V}^{(m+1)} = \mathbf{V}^{(m)} + \theta\Delta t\Delta\mathbf{A}_c^{(m)}$.
 - 9: **end for**.
 - 10: $\mathbf{A}_c^{n+1} = \mathbf{A}_c^{(m+1)}$
 - 11: $\mathbf{V}^{n+1} = \mathbf{V}^{(m+1)}$
-

Appendix E

Exact Solution of the Transcritical Flow with a Shock Problem

Here we discuss the exact solution of the transcritical flow with a shock problem introduced in Section 5.2. We consider a 1D region delimited by $x \in [0, 25]$ m and whose bed elevation is:

$$z_b(x) = \begin{cases} 0.2 - 0.05(x - 10)^2, & \text{if } 8 \text{ m} < x < 12 \text{ m}, \\ 0, & \text{otherwise.} \end{cases} \quad (\text{E.1})$$

As previously stated, we can solve Equations (5.7)-(5.9) to obtain a solution. However, doing so requires further knowledge about the problem, which is often not provided in other works that also present numerical solutions. Therefore, we derive the exact steady-solution of a 1D transcritical flow that presents a hydraulic shock. We start from the steady shallow water equations:

$$\frac{\partial q}{\partial x} = 0, \quad (\text{E.2})$$

$$\frac{\partial}{\partial x} \left(hu^2 + \frac{gh^2}{2} \right) = -gh \frac{\partial z_b}{\partial x}, \quad (\text{E.3})$$

where u is the fluid's depth-averaged velocity, $q = hu$ is its specific discharge, z_b is the bed elevation and g is the gravitational acceleration. If we consider the domain $x \in [0, L]$ and prescribe $q(x = 0) = q_0$, Equation (E.2) results in $q = q_0$. Then,

Equation (E.3) can be written as:

$$\begin{aligned}
(-u + gh) \frac{\partial h}{\partial x} &= -gh \frac{\partial z_b}{\partial x} \\
\therefore \left(-\frac{q^2}{h^2} + gh \right) \frac{\partial h}{\partial x} &= -gh \frac{\partial z_b}{\partial x} \\
\therefore \left(\frac{q_0^2}{gh^3} - 1 \right) \frac{\partial h}{\partial x} &= \frac{\partial z_b}{\partial x} \\
\therefore -\frac{q_0^2}{2g} \frac{\partial h^{-2}}{\partial x} - \frac{\partial h}{\partial x} &= \frac{\partial z_b}{\partial x}.
\end{aligned} \tag{E.4}$$

Integrating (E.4) between two positions labeled 1 and 2, we obtain:

$$-\frac{q_0^2}{2gh_2^2} + \frac{q_0^2}{2gh_1^2} - h_2 + h_1 = z_{b2} - z_{b1}, \tag{E.5}$$

from which we infer that:

$$h + \frac{q_0^2}{2gh^2} + z_b = \text{constant} \tag{E.6}$$

throughout the domain. Also, if we consider the specific energy:

$$E = h + \frac{q_0^2}{2gh^2}, \tag{E.7}$$

we have:

$$E + z_b = \text{constant}. \tag{E.8}$$

This implies that if we increase the bed elevation by Δz from one position 1 to another 2 along the flow, we are decreasing the specific energy by the same amount, i.e., $E_2 = E_1 - \Delta z$.

For a given q_0 , Figure E.1 presents the general form of the curve (E.7). The point with specific energy E_c and fluid height h_c marks the transition between subcritical and supercritical flow states. The related critical value is represented by the ratio of the flow speed and the velocity of a small amplitude gravity wave, which is given by the Froude number:

$$Fr = \frac{u}{\sqrt{gh}} = \frac{q}{\sqrt{gh^3}}. \tag{E.9}$$

Flows with $Fr > 1$ are said to be supercritical while those with $Fr < 1$ are subcritical. Subcritical flows are deeper and slower, having more potential than kinetic energy. Meanwhile, supercritical flows are shallower and faster, with more kinetic than potential energy.

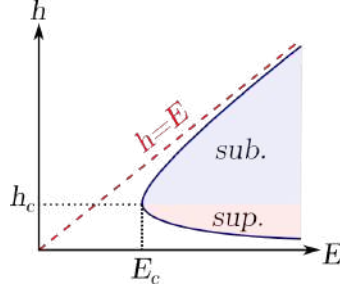


Figure E.1: Plot of the energy-height curve for flows in a 1D channel. It show the critical height and energy values, and the intervals of the curve that denote subcritical or supercritical regimes.

The critical height h_c can be computed by minimizing the energy equation (E.7):

$$\frac{dE}{dh} = 0 \quad \therefore \quad 1 - \frac{q_0^2}{gh_c^2} = 0 \quad \therefore \quad h_c = \left(\frac{q_0^2}{g} \right)^{1/3}. \quad (\text{E.10})$$

Next, we replace the relation $q_0^2 = gh_c^3$ obtained from (E.10) into (E.7) to define the critical specific energy as a function of the critical height:

$$E_c = \frac{3}{2}h_c. \quad (\text{E.11})$$

Before defining the exact solution for the case with a transcritical flow with a shock, we initially examine the solution of a subcritical flow and a transcritical flow, building up to the more complex solution of the target scenario.

For the subcritical flow, consider the Figure E.2 that depicts the expected profile of the solution. We consider the boundary conditions $q(x = 0) = q_0 = 4.42 \text{ m}^2 \text{ s}^{-1}$ and $h(x = L) = 2 \text{ m}$ and the initial conditions $q(x) = q_0$ and $h(x) = h_L$. If we apply the Bernoulli equation (E.6) between $x = L$ and any other position, we get:

$$h + \frac{q_0^2}{2gh^2} + z_b = h_L + \frac{q_0^2}{2gL^2} \\ \therefore h^3 + \left(z_b - \frac{q_0^2}{2gh_L^2} - h_L \right) h^2 + \frac{q_0^2}{2g} = 0, \quad \forall x \in [0, L], \quad (\text{E.12})$$

from where we can obtain three values for $h(x)$. One solution is negative and can be discarded because it does not have any physical meaning. The other two represent solutions with subcritical and supercritical regimes. Hence, we need further knowledge about the resulting flow regime to select the actual solution value for h .

First, we compute the critical fluid height $h_c = 1.2581$ from (E.10) and observe that the initial fluid heights at the positions 0, 1 and 2 in Figure E.2 are subcritical. Then we compute the solution $h_0 = 2 \text{ m}$ from (E.12) to keep the flow subcritical and the associated specific energy $E_0 = 2.2489 \text{ m}$. Downstream of this point,

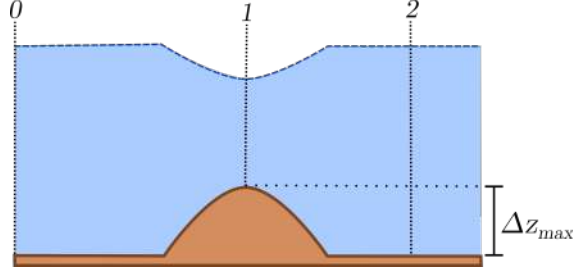


Figure E.2: Representation of the solution of the subcritical flow in a 1D channel with a bump.

the flow can only become supercritical if the energy is minimized to the critical value $E_c = 1.8872$ m, computed with (E.11). However, the maximum bed elevation $\Delta z_{\max} = 0.2$ m is not enough to reduce E_0 to E_c and thus the flow remains subcritical in all the domain. For the present case, in the energy-height diagram in Figure E.3, the flow goes from the state A to B as it climbs the mound, coming back to A downstream.

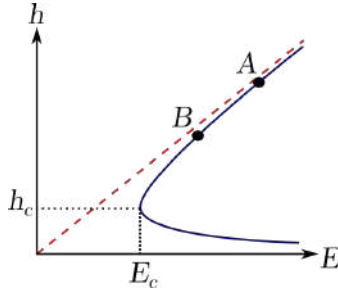


Figure E.3: Energy-height diagram for the subcritical flow in a 1D channel over a bump. The points A and B indicate different states of the fluid along the channel.

The next case we examine is the transcritical flow whose solution profile is illustrated in Figure E.4. We apply the initial conditions $q(x) = q_0 = 1.53 \text{ m}^2 \text{ s}^{-1}$ and $h(x) = h_L = 0.66$ m and the boundary condition $q(x = 0) = q_0$. In addition, while the flow is subcritical, we enforce $h(x = L) = h_L$. To start, we compute the critical values $h_c = 0.6202$ m and $E_c = 0.9303$ s. Then, we note that, as the fluid flows over the bump, the specific energy is reduced by $\Delta z_{\max} = 0.2$ m before reaching the position 1 in Figure E.4. Also, using the initial fluid height, we compute the specific energy $E_0 = 0.9339$ s at the position 0 by the start of channel in Figure E.4. These data show that, at first, as the initial subcritical flow passes over the bump, the specific energy would be less than the critical value ($E_0 - \Delta z_{\max} < E_c$), indicating that it does not have the energy needed to surpass the bump. Thus, as the fluid inflows, h_0 increases until h_c (and E_c) is obtained at the position 1 and the fluid overpasses the mound in a supercritical regime.

To compute the exact solution, we apply the Bernoulli equation (E.6) between

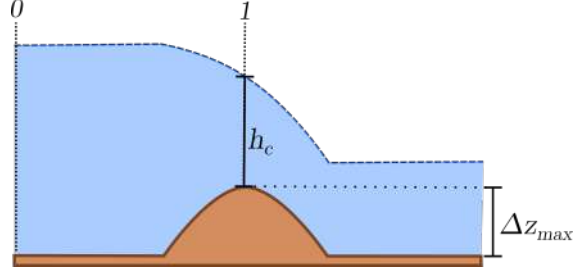


Figure E.4: Representation of the solution of the transcritical flow in a 1D channel with a bump.

position 1, denoted as x_c , and any other position and get:

$$h + \frac{q_0^2}{2gh^2} + z_b = h_c + \frac{q_0^2}{2gh_c^2} + z_{b_{\max}}$$

$$\therefore h^3 + \left(z_b - \frac{q_0^2}{2gh_c^2} - h_c - z_{b_{\max}} \right) h^2 + \frac{q_0^2}{2g} = 0, \quad \forall x \in [0, L], \quad (\text{E.13})$$

which we solve, obtaining three values for h . The negative solution is discarded by default, while we choose the value relative to a subcritical or a supercritical state according to whether it is associated with a position, respectively, upstream or downstream of x_c . In Figure E.5, the resulting steady-state flow starts from the subcritical state A , reaching C at the top of the mound, where the flow becomes supercritical. As the flow advances down the bump, it regains specific energy, reaching B .

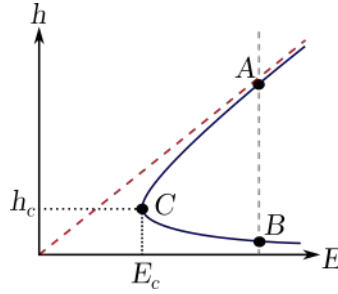


Figure E.5: Energy-height diagram for the transcritical flow in a 1D channel over a bump. The points A , B and C indicate different states of the fluid along the channel.

The last case we examine is the steady-state solution of a transcritical flow that presents a hydraulic shock and whose profile is depicted in Figure E.6. We start from the subcritical state $q(x) = q_0 = 0.18 \text{ m}^2 \text{ s}^{-1}$ and $h(x) = h_L = 0.33 \text{ m}$ and enforce the boundary conditions $q(x=0) = q_0$ and $h(x=L) = h_L$. The computed critical height and energy are $h_c = 0.1489 \text{ m}$ and $E_c = 0.22 \text{ m}$. As in the previous case, the initial specific energy $E_0 = 0.3451 \text{ m}$ of the position 0 at the start of the channel is not enough to make the fluid surpass the mound. Thus, the fluid height upstream of

the mound will increase and the fluid will reach the critical height h_c at the position 1 and transition into a supercritical regime. In sequence, the boundary condition at $x = L$ indicates that the flow should change back to a subcritical regime. However, in a supercritical flow between 1 and 3, the energy rises and the height diminishes, making it impossible to reach the critical point. Thus, a hydraulic shock takes place at a position x_{shock} between 1 and 3, evolving the flow to a subcritical state.

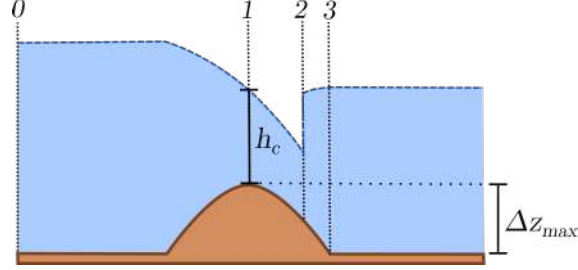


Figure E.6: Representation of the solution of the transcritical flow with a shock in a 1D channel with a bump.

The exact solution upstream of the shock can be computed with the Equation (E.13) used in the previous case:

$$h^3 + \left(z_b - \frac{q_0^2}{2gh_c^2} - h_c - z_{b_{\max}} \right) h^2 + \frac{q_0^2}{2g} = 0, \quad \forall x \in [0, x_{\text{shock}}), \quad (\text{E.14})$$

choosing solution values according to the flow regime. In addition, the exact solution downstream of the shock is obtained by solving the equation (E.12) of the first scenario:

$$h^3 + \left(z_b - \frac{q_0^2}{2gh_L^2} - h_L \right) h^2 + \frac{q_0^2}{2g} = 0, \quad \forall x \in (x_{\text{shock}}, L]. \quad (\text{E.15})$$

At last, we need to define the exact position of the shock. Let the fluid momentum be $p = \rho qu = \rho q^2/h$ and consider the hydrostatic force $f = \int_0^h \rho gh dh = \rho gh^2/2$. Then, the momentum change across the shock should be balanced by the difference in the hydrostatic force:

$$\begin{aligned} \rho \frac{q_0^2}{h_+} - \rho \frac{q_0^2}{h_-} &= \rho g \frac{h_+^2}{2} - \rho g \frac{h_-^2}{2} \\ \therefore q_0^2 \left(\frac{1}{h_+^2} - \frac{1}{h_-^2} \right) + \frac{g}{2} (h_-^2 - h_+^2) &= 0, \quad \text{for } x = x_{\text{shock}}, \end{aligned} \quad (\text{E.16})$$

where $h_-(x = x_{\text{shock}})$ and $h_+(x = x_{\text{shock}})$ are the water height upstream and downstream of the shock. We have implemented an Octave [124] script to solve this equation and obtain $x_{\text{shock}} \approx 11.6656$ m.

Figure E.7 shows an energy-height diagram with the different states of the fluid in the present case. The resulting steady-state flow starts from the subcritical state A , reaching C at the top of the mound, where the flow becomes supercritical. As

the flow advances down the bump, it regains specific energy reaching B . Then, an hydraulic shock changes the fluid to the subcritical state D , which gains energy down the mound to E .

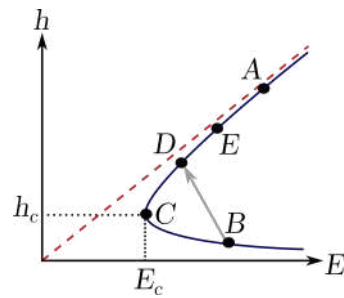


Figure E.7: Energy-height diagram for the transcritical flow with a shock in a 1D channel over a bump. The points A through E indicate different states of the fluid along the channel.