

**FIELD SCALE FLOW MODELING OF THIXOTROPIC  
MINE TAILINGS USING THE MATERIAL POINT  
METHOD**

by

Étienne Parent

A thesis submitted to the Faculty of Graduate and Postdoctoral Affairs in partial fulfillment of the requirements for the degree of

Master of Applied Science

in

Environmental Engineering

Carleton University  
Ottawa, Ontario

© 2019, Étienne Parent

## **Abstract**

Understanding free surface flow of mine tailings is necessary to plan deposition operations in tailings' storage facilities (TSF), and is important to predict the consequences of a dam break. Proper understanding of deposition operations has probably hindered adoption of alternative tailings technologies, whereas recent failures of tailings impoundments have had catastrophic consequences. Flow behavior is crucial to understand for effective risk mitigation.

Tailings exhibit thixotropic properties – their behavior is a function of the material's stress history. A thixotropic constitutive model has been implemented into an open-source Material Point Method (MPM) framework. The results were validated with previous experimental data at laboratory scale and compared with Bingham rheology with good agreements.

Large-scale simulations were done for 2D and 3D cases. The thixotropic model was able to model self-forming channels seen during field scale deposition, which Bingham rheology is unable to capture. Dam break events also scaled differently from laboratory scale experiments with thixotropy. This can possibly be an element that explains longer runouts than initially predicted with standard rheology.

## **Acknowledgements**

First, I would like to acknowledge the guidance of my supervisor, Professor Paul Simms. Thanks to him, I got opportunities to tackle crucial engineering problematics in the field of tailings management. His supervision was useful to develop skills by participating in a variety of projects which allowed me to present this thesis.

I would also like to thank Professors John Goldak (Carleton University) and Arno Talmon (TU Delft) for their input on numerical strategies as well as Ryan Taylor (Carleton University) for the technical support.

I also recognize the positive influence of my peers at the Carleton Tailings Research Group and the Deltares team on my thesis and work experiences that will long be remembered: Luca Sittoni, Jill Hanssen, Jan van Kester and many more.

Finally, I am grateful to my family, boyfriend and friends for their constant encouragements and support.

## Table of Contents

<b>Abstract.....</b>	<b>ii</b>
<b>Acknowledgements .....</b>	<b>iii</b>
<b>Table of Contents .....</b>	<b>iv</b>
<b>List of Tables .....</b>	<b>vii</b>
<b>List of Figures.....</b>	<b>viii</b>
<b>1: Introduction .....</b>	<b>1</b>
1.1    Background.....	1
1.2    Scope .....	6
1.3    Objectives.....	6
1.4    Novelty .....	7
1.5    Outline of the Thesis .....	8
<b>2: Literature Review .....</b>	<b>9</b>
2.1    Deposition and management of tailings .....	9
2.1.1    Conventional deposition.....	9
2.1.2    Post-deposition behavior of tailings.....	11
2.1.3    Beach slope prediction .....	13
2.1.4    Dam breaks.....	15
2.1.5    Slurry and debris flow modeling.....	16
2.2    Rheology.....	17
2.2.1    Viscosity.....	18
2.2.2    Yield stress .....	19
2.2.3    Thixotropy.....	19
2.2.4    Constitutive models.....	21

2.3	Numerical modeling .....	24
2.3.1	Finite Elements Methods - Eulerian and Lagrangian descriptions.....	24
2.3.2	Mesh-free methods.....	25
2.4	Material Point Method (MPM).....	27
2.4.1	Summary of MPM.....	27
2.4.2	Governing equations .....	29
2.4.3	Spatial and temporal discretization .....	30
2.4.4	Shape functions .....	34
2.4.5	Overview of MPM characteristics.....	39
2.5	Summary.....	40
<b>3:</b>	<b>Modeling Methodology.....</b>	<b>43</b>
3.1	Uintah-MPM framework .....	43
3.1.1	Compiling the Code .....	44
3.1.2	Running the code.....	46
3.2	Pre-processing .....	47
3.2.1	Geometry initialization.....	47
3.2.2	Continuous deposition.....	49
3.3	Post-processing.....	51
3.3.1	Parsing tools .....	51
3.3.2	Advanced rendering .....	52
3.3.3	Point-based visualisation.....	53
3.4	Implementation of constitutive models .....	59
3.4.1	Thixotropic constitutive equations .....	59
3.4.2	Algorithmic Implementation.....	64
<b>4:</b>	<b>Validation .....</b>	<b>67</b>

4.1	Poiseuille flow .....	67
4.2	Dambreak experiment with Newtonian fluids.....	69
4.3	Bingham rheology .....	77
4.3.1	Dam break scenario.....	78
4.3.2	Pipe deposition scenarios .....	80
4.4	Thixotropic rheology .....	83
4.4.1	Experimental validation .....	83
4.4.2	Investigation of aging speed.....	88
4.5	Parametric analysis .....	92
4.5.1	Sensitivity analysis to numerical parameters .....	92
4.5.2	Analysis of numerical sensitivity .....	103
4.5.3	Sensitivity to physical parameters.....	105
<b>5:</b>	<b>Large scale 3D modeling scenarios .....</b>	<b>115</b>
5.1	Scalability of rheological models to field scale.....	115
5.2	Large scale dam breaks .....	120
5.3	Large scale pipe deposition .....	132
<b>6:</b>	<b>Conclusion .....</b>	<b>142</b>
<b>7:</b>	<b>Recommendations.....</b>	<b>145</b>
	<b>Bibliography .....</b>	<b>146</b>

## List of Tables

Table 1: Qualitative description of different types of rheological fluids.....	22
Table 2: Physical and numerical parameters for Poiseuille flow simulation.....	68
Table 3: Parameters for Newtonian fluid dam break experiments .....	70
Table 4: Bingham model for gold tailings measured under different conditions .....	77
Table 5: Coussot model parameters for tailings .....	84
Table 6: Coussot model parameters for tailings .....	89
Table 7: Overview of parameters varied in the numerical sensitivity analysis .....	94
Table 8: Overview of parameters varied in the numerical sensitivity analysis .....	106
Table 9: Baseline material parameters for numerial analysis .....	116
Table 10 : Physical and numerical parameters for sensitivity analysis .....	129
Table 11: Physical properties for large scale pipe deposition simulation.....	133

## List of Figures

Figure 1: End of pipe deposition of mine tailings (Carleton Tailings Research Group) ....	4
Figure 2: Embankment types (Adapted from Vick & al., 1983) .....	10
Figure 3: Sketch of self-forming channels in tailings flow (Adapted from MPA Williams, 2010) .....	12
Figure 4: Viscoplastic flow down an inclined plane (Adapted from Lui & Mei, 1990)...	14
Figure 5: Comparison of experimental data and non-thixotropic viscoplastic model.....	21
Figure 6: Comparison of common rheological models.....	23
Figure 7: Evolution of viscosity as a function of time for different shear rates; Coussot thixotropic fluid .....	23
Figure 8: SPH pipe deposition of Bingham mine tailings (Adapted from Babaoglu, 2014)	
.....	26
Figure 9: Solution procedure for the MPM.....	28
Figure 10: Shape function and its gradient for traditional MPM (Adapted from Guilkey & al., 2009) .....	36
Figure 11: Shape function and its gradient for GIMP (Adapted from Guilkey, & al., 2009)	
.....	38
Figure 12: Initial particle distribution for two colliding disks (Adapted from Guilkey & al., 2009) .....	48
Figure 13: Extrusion simulations with Uintah-MPM (Adapted from Guilkey & al., 2009)	
.....	50
Figure 14: Visualization of point-based data using MatLab scatter3 plot .....	55
Figure 15: Advanced visualization of point-based data.....	58

Figure 16: Comparison of numerical simulation and Poiseuille analytical solution .....	68
Figure 17: Water dam break, validation with experimental results .....	71
Figure 18: Shampoo dam break, validation with experimental results.....	72
Figure 19: Comparison between experimental results, author's results and MPM model for a water dam break. Dimensionless height (y/L) at left wall, middle (0.27m) and right wall .....	74
Figure 20: Comparison between experimental results, author's results and MPM model for a shampoo dam break. Dimensionless height (y/L) at the left wall, middle (0.27m), and right wall. ....	75
Figure 21: Flume setup for deposition experiments (Adapted from Henriquez, 2009)....	78
Figure 22: Bench scale dam break runout distance, experimental data and predictions ..	79
Figure 23: Bench scale pipe deposition runout distance, experimental data and predictions .....	80
Figure 24: Transient runout of mine tailings after 3, 6 and 9 seconds.....	82
Figure 25: Experimental set-up for thixotropic free surface flow modeling (Adapted from Mizani, 2016b).....	84
Figure 26 : Time-dependant viscosity profile with Mizani (2016b) parameters .....	85
Figure 27: Predicted and experimental beaching profiles of thixotropic oil sand tailings	85
Figure 28: 2D Flow profile at different time steps: 2 min 0 s,7 min. 40 s, 13 min. 40 s, 21 min. 20 s.....	87
Figure 29: Final runout modeled using a Bingham rheological model .....	88
Figure 30: Time-dependant viscosity profiles with increased aging parameter .....	89
Figure 31: Final flow profile for different aging speeds.....	91

Figure 32: Initial setup for parametric analysis .....	92
Figure 33: Viscosity and runout as a function of time. Grid resolution sensitivity at time t = 1s. From top to bottom: 0.1m, 0.05m, 0.033m, 0.025m squared cells with $2^3$ points per cell.....	95
Figure 34: Viscosity and runout as a function of time. Grid resolution sensitivity at time t = 5s. From top to bottom: 0.01m, 0.005m, 0.0033m, 0.0025m squared cells with $2^3$ points per cell.....	96
Figure 35: Viscosity and runout as a function of time. Grid resolution sensitivity at time t = 15s. From top to bottom: 0.01m, 0.005m, 0.0033m, 0.0025m squared cells with $2^3$ points per cell.....	97
Figure 36: Viscosity and runout as a function of time. Points per cell sensitivity at times t = 1s. From top to bottom: $2^3$ $3^3$ , $4^3$ points on grid cells of 0.01m and $3^3$ points on grid cells of 0.0033m.....	98
Figure 37: Viscosity and runout as a function of time. Integration points sensitivity at times t = 5s. From top to bottom: $2^3$ $3^3$ , $4^3$ points on grid cells of 0.01m and $3^3$ points on grid cells of 0.0033m.....	99
Figure 38: Viscosity and runout as a function of time. Integration points sensitivity at times t = 15s. From top to bottom: $2^3$ $3^3$ , $4^3$ points on grid cells of 0.01m and $3^3$ points on grid cells of 0.0033m.....	100
Figure 39: Viscosity and runout as a function of time. Sensitivity to timesteps at t = 1s and t = 13s. From top to bottom, 20% and 90% of the stability criteria.....	101
Figure 40: Viscosity and runout as a function of time. Sensitivity to interpolation method: 3 <sup>rd</sup> order B-Splines at times t = 1s, t = 5s and t = 15s.....	102

Figure 41: Base test case for sensitivity analysis to physical parameters. Viscosity and runout as a function of time. From top to bottom, at times t= 1s and 1 = 15s.....	107
Figure 42: Sensitivity to minimum viscosity (0.1 Pa*s) at time t = 1s and t = 15s. ....	109
Figure 43: Sensitivity to minimum viscosity (1 Pa*s) at time t= 1s and t = 15s. ....	109
Figure 44: Runout at 1 s and 15 s for different alpha values (0.2, 2, 10 respectively) ...	110
Figure 45 : Runout at 1 s and 15 s for different n values (1.5, 2, 3 respectively).....	111
Figure 46: Runout at 1 s and 15 s for different alpha values (1, 2, 10 respectively) .....	113
Figure 47: Runout distance as a function of scale and constitutive model used.....	117
Figure 48: Time to static equilibrium for different constitutive models at different scales .....	117
Figure 49: Mobilized fraction of the impoundment.....	118
Figure 50: State of the tailings after runout (Photos from watcher.news, 2016) .....	121
Figure 51: Satellite imagery of the tailings runout, before and after the dam collapse ..	122
Figure 52: Topography of the valley surrounding the TSF .....	123
Figure 53: Altimetroc profile of the vally at different cross sections .....	123
Figure 54: Initial conditions for 3D dambreak analysis.....	124
Figure 55: Cross sectional view of dam break flow at 30, 60, 90, 150, 250 and 330 seconds .....	125
Figure 56: Plan view of dam break flow at 30, 60, 90, 150, 250, 330 seconds .....	126
Figure 57 : Sensitivity of runout length to physical parameters .....	130
Figure 58 : Sketch of the large scale pipe deposition scenario .....	132
Figure 59: Pipe deposition of thixotropic tailings for field scale conditions .....	135
Figure 60 : Pipe deposition of thixotropic tailings for field scale conditions, plan view	137

Figure 61: Pipe deposition of thixotropic tailings for field scale conditions, section view

..... 139

# **1: Introduction**

## **1.1 Background**

The mining sector play an essential role in the Canadian economy. The industry employs approximately 426,000 workers and contributed \$97 billion, 5%, to Canada's GDP (MAC, 2018). In the process of natural resource extraction, however, large quantities of by-products are produced. These include waste rock, blocks of large diameter typically found in open-pit mines as the soil cover is removed to reach the underlying ore deposit, and tailings, the by-product of the refinery operations. In the hard rock mining industry, this slurry-like material is mostly comprised of ground rock and water from the mineral extraction process, the rock is typically silt to sand size due to requirements of particle size for efficient mineral grain separation. For commodities extracted from overburden, such as for oil sands bitumen extraction and bauxite mining, the slurry is a mixture of sand, fine clay particles, and water.

In most cases, the tailings are typically transported by pipe as a slurry with a relatively high water content to an impoundment site called tailings storage facility (TSF). Tailing storage facilities make up some of the largest man-made structures in the world. In fact, some TSF are so big they are part of the few structures that can be seen from the International Space Station, such as the oil sand TSF in Northern Alberta. In the public eye, social and environmental accountability of mine operators is important. Impact mitigation measures are an essential part of mining projects and health and safety are a priority for all stakeholders.

The containment structures in the TSF can be a combination of dams, embankments and natural formations (e.g. rock outcrops). Dams are built from waste rock left by open-pit mining operations or other nearby geomaterials. These structures can be problematic because of their size and external factors such as: poor construction techniques, earthquakes, heavy rainfall and poor water control (Vick, 1983). Unlike hydroelectric dams or other types of concrete dams, the design process is not as strict and regulated for tailing retention structures. Flaws in the construction process can lead to failure of the dam and subsequent runout of tailings. In such events, the catastrophic impacts are economical, environmental and social. They can even result in loss of human life, such as the Brumadinho dam disaster in 2019. The lives of 270 people were claimed (Orduz Roja & al., 2019).

In response to the high occurrence of collapses, considerable efforts are made by the industry and researchers to better understand the rheology and geotechnical behavior of the tailings and the surrounding dams and avoid these catastrophic events (De Stephano & al., 2019; Llano-Serna & al., 2017; Rafael & al., 2017).

Rheological modeling of tailings is also important for daily operations of the TSF. It is used by engineers to predict the final stack geometry of the deposited tailings. One important design parameter in managing tailings is the thickness of the layers formed by the flow. The thickness will affect significantly dewatering properties of the tailings by consolidation and desiccation. Better dewaterability can increase the effective stress and favor overall stability of the impoundment. The height of the final runout and its uniformity are also important to maximize space usage in the TSF for a given footprint (Simms & al., 2011, Simms & al., 2017).

Recent research has shown tailings to be a thixotropic material (Nguyen & al., 1998, Mizani & al., 2017, Mizani and Simms 2016); one that gains or loses strength as a function of the shearing efforts to which it is subjected. This process is dictated by the reversible formation of microstructural bonds between attractive particles. This implies that the observed physical properties of the material aren't solely a function of stress state, but of stress history.

Therefore, with thixotropy, the variation of properties as a function of material history is known to occur in tailings. The observed behavior of tailings at field scale is dependant on the age of the deposit. Newer deposits will flow will spread uniformly while older deposits would have a tendency to flow in channels and have more convex beaching profiles (Simms & al., 2011).

It is possible that thixotropy can represent this unique flow behavior as it seems to be linked to an aging process. Thixotropic constitutive equations inherently include this time-dependency in their formulations. Several models have been proposed in the past with different degrees of complexity to account for phenomenon like hysteresis (Coussot, 2002; Hewitt & Blamforth, 2013)

Bingham rheology (a fluid that will stop flowing under a certain shear limit) is routinely used to model the flow and stoppage of mine tailings. However, the accuracy of this model is only valid for small scale experiments (Henriquez, 2009).

One of the problems with the use of a Bingham model is that the measured yield stress is largely dependant on the measurement technique and range of shear strain values (Nguyen & al., 2006, Mizani & al., 2017, 2013). Since the variation in applied shear is important when going between bench scale and field scale flow, some discrepancies can occur. A thixotropic model arguably offers a better fit to data on mine tailing rheology.

Figure 1 represents the variety of states under which the tailings will be found in during its lifetime and can provide better visual understanding of the flow behavior of fresh mine tailings over an aged surface. This complex behavior relies on multiple physical processes other than thixotropy such as: desiccation, consolidation, sedimentation, erosion. It is however hypothesised that thixotropy plays an important role in explaining the observed phenomenology.



**Figure 1: End of pipe deposition of mine tailings (Carleton Tailings Research Group)**

In light of this, efforts should be made to assess the effect of using thixotropic rheology to more complex situations by incorporating it as a constitutive model in numerical solvers (Finite Elements, Finite Difference, etc.).

In traditional methods of computational fluid dynamics (CFD), tracking material history is hard to accomplish as the discretization methods typically rely on numerical mass advection. In the contrary, Lagrangian methods that are commonly used for solids analysis allows easy tracking of the material's stress history, but often fail numerically when large deformations are introduced due to mesh entanglement. Since mine tailings are a rheological material whose properties oscillate between the two spectrums – they aren't definitely a liquid, but neither are they definitely a solid – other means of numerical analysis must be considered.

The Material Point Method (MPM) is a particle-based Finite Element Method (FEM), which conjoins properties of Eulerian and Lagrangian space discretization (Sulsky & al., 1994). This allows the analysis of large deformation problems while considering material history.

MPM would therefore be a good candidate to model thixotropy with a reasonable degree of physical accuracy. The method is also scalable in terms of computational costs to 3D modeling unlike mesh-free methods such as Smooth Particle Hydrodynamics (SPH) that relies on expensive neighbour searches (Dominquez, 2011).

Also, the MPM has been extensively used to models geomaterials (Yerro & al., 2013). Therefore, MPM could extend naturally to the in-place state of aged tailings where the behavior is best described with geotechnical models (Daliri & al., 2014) and possibility consider the transition between the two states.

## 1.2 Scope

The scope of this work is to investigate the potential of thixotropic rheology to explain some of the behavior seen in field scale flow of mine tailings. Comparison with Bingham rheology will also be accomplished to further highlight differences between the two rheological models.

Both set of constitutive equations are implemented into Uintah-MPM, an open-source modeling framework developed mainly by the University of Utah and Los Alamos National Laboratory (Guilkey & al., 2009).

In order to validate the numerical implementation and its ability to represent physical parameters accurately, extensive validation with experimental data is done.

Once a strong argument for the applicability of MPM to rheological materials modeling is built, simulations of field scale scenarios will be done to see how both constitutive models perform under different conditions.

## 1.3 Objectives

The goal of this project is to investigate new numerical methods that are capable of modeling the history-dependant behavior of mine tailings. The method needs to be scalable in size and in time to incorporate 3D simulations.

Hopefully, phenomenology seen on site but not fully understood or reproducible in small scale laboratory experiments and 2D numerical simulations will be highlighted. Particularly, the capacity of flowing tailings to form channelized flow as they age is of interest. The scalability of rheological models is also of interest.

The material point method (MPM) is a good candidate in terms of modeling capabilities and scalability to perform these simulations. Once rheological constitutive models are implemented, extensive validation should be done to validate its performance under a variety of conditions. Firstly, simulation of experimental data for bench scale studies will be carried out. This includes Newtonian (Cruchaga & al., 2007), Bingham (Gao & Fourie, 2018; Babaoglu, 2014; Henriquez & Simms, 2009), and thixotropic (Mizani, 2017) constitutive models.

A sensitivity analysis to numerical parameters is also done to study the accuracy of the model under different parametric conditions.

Once validation is completed, field scale simulations will be done for dam break and field deposition scenarios with certainty about the representativity of the results. The main hypothesis is that the thixotropic model, incorporated in a robust 3D MPM framework will be sufficient to model self-forming channels, which are seen at field scale. Effects of constitutive model on dam breaks of field scale size (up to 100 meters in height) will also be investigated.

## 1.4 Novelty

The complex rheological behavior of mine tailings and the transition between a highly deformable (fluid) state and a solid (geotechnical) state requires new modeling methods. A thixotropic rheological model can possibly provide answers to phenomenology seen at field scale such as self-forming channel during deposition or the much longer runouts measured when dam breaks occur, when compared with back calculated values.

The material point method (MPM) is often used in geotechnical engineering (Soga & al., 2015) but its capacity to model rheological or Newtonian fluids is still an active zone of research. Also, the scalability of MPM to bigger computer systems makes it possible to compute large scale 3D situation with complex physical processes. Bingham and thixotropic rheology will be compared in their capability to simulate field scale tailing deposition scenarios.

## 1.5 Outline of the Thesis

The outline of the thesis is presented as follows;

Chapter 2: A review of tailings behavior and their management strategies, rheology of non-Newtonian fluids and different numerical modeling methods.

Chapter 3: Mathematical and algorithmic implementation of Bingham and thixotropic rheology in a MPM framework, instructions on pre- and post-processing.

Chapter 4: Validation of Newtonian, Bingham, Thixotropic models with laboratory data. Sensitivity analysis to physical and numerical parameters.

Chapter 5: Field scale, 3D scenario analysis for dam breaks and pipe deposition

Chapter 6: Conclusion

Chapter 7: Recommendations for future work

## **2: Literature Review**

Three area of interest have been identified as relevant to the project. Therefore, scientific literature pertaining to the state of the art in these areas will be reviewed in depth. The identified areas are: deposition of tailings; advanced rheology; and numerical modeling methods.

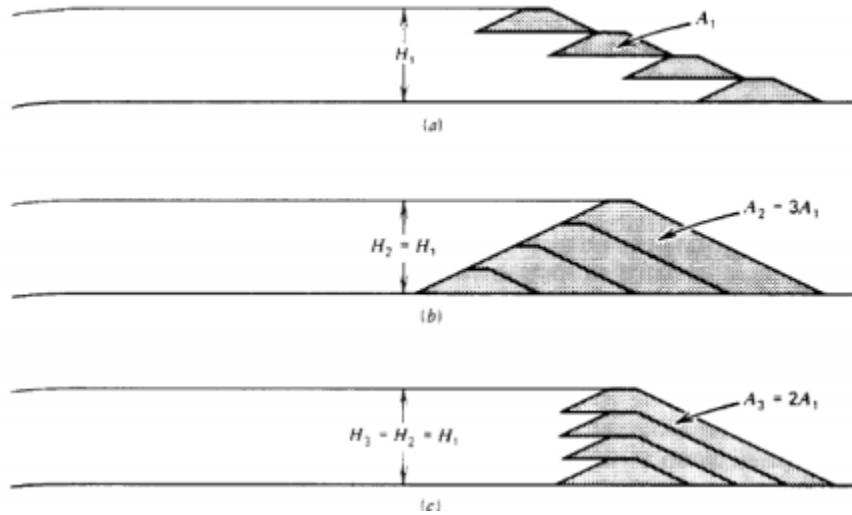
### **2.1 Deposition and management of tailings**

#### **2.1.1 Conventional deposition**

Tailings, the material left over after minerals of value have been extracted from the ore is a by-product of mineral extraction activities. It requires to be managed efficiently during the lifetime of the mine and its final state must ensure geotechnical, chemical and physical stability in time in the closure and post-closure phases of the mine's life.

Traditionally, tailings are pumped at a high water content and stored behind dams in the tailings storage facility (TSF). Once they reach the end of the pipe, their free surface flow behavior determines what part of the impoundment they will reach. Several engineering parameters can be controlled to provide uniform layers of material with good spatial distribution. Engineers can consider modifying the deposition flow rate, number of spigots and the time before changing the deposition point are. It is also common to rely on natural formations of land such as hillsides to provide partial containment of the tailings (Blight, 2009). The containment structure usually begins with the construction of a small dyke which will be progressively raised as more tailings are stored (Vicks, 1983).

Three methods are used to raise the height of the structure. Figure 2 presents the three most common methods namely upstream, downstream and centerline construction.



**Figure 2: Embankment types (Adapted from Vick & al., 1983)**

A notable difference between the three construction methods is the amount of material needed to build the retaining structure. For upstream dams, the total volume of material is significantly lower than for the downstream technique. The centerline construction method is perhaps a compromise between the first two.

It should be mentioned that the upstream dyke construction method is becoming less and less popular as it is linked to poor performance and greater risk as it makes up 76% of failures in the world and 47% of failures in Europe (Rico & al., 2008). Since the higher lifts are built directly on the tailings, problems can occur if the tailings are not in an appropriate geotechnical state. The consolidation process needs to be advanced to provide sufficient effective stress to ensure stability, which can be operationally challenging (Martin & McRoberts, 1999; Kossoff & al., 2014).

Determining the final beaching profile and slope angle after each pour is a relevant engineering parameter for a variety of operational activities. It provides information used to design the dam and plan total stored volume and period lifts (Fitton, 2006; Henriquez & Simms, 2009). Therefore, being able to predict the height and the shape of the stack throughout its lifetime is necessary for engineers (Fourie & Gauw, 2010; Williams & al., 2008).

Other deposition solutions exist, such as subaqueous methods. They will not be investigated here although they can be considered as an alternative to dam storage in some cases.

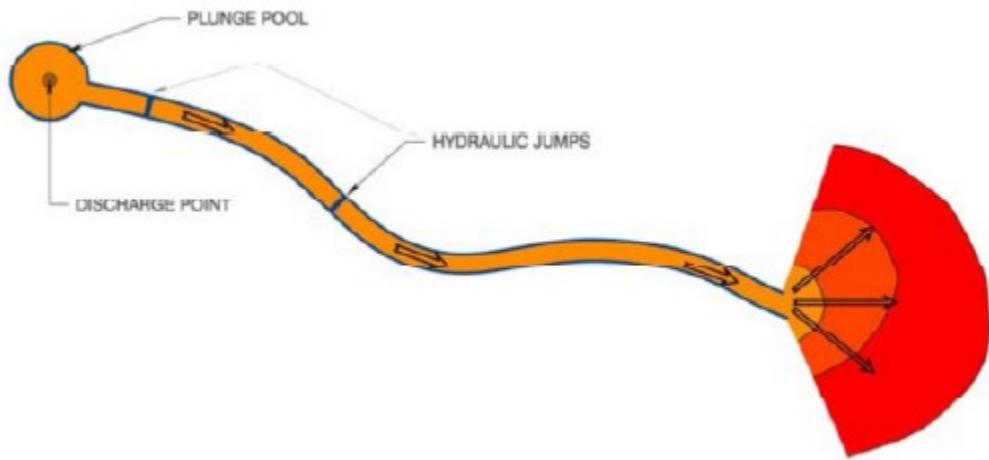
### **2.1.2 Post-deposition behavior of tailings**

Tailings can be deposited under a variety of rheological states. Conventional deposition has very high water content (sometimes ~100% gravimetric water content for hard rock tailings and ~200% for oil sands and bauxite tailings). Attempts to optimize water usage has led to the adoption of high density tailings, which are pumped in the laminar range and can behave differently under free surface flow conditions. Along with other factors like mineralogy, processing and additives, the flow behavior of the tailings is altered.

From field observations, tailings first flow in a thin sheet that spreads out in a lobe shape from the deposition point. During this initial phase of the flow, the beaching profile can be described as a parabola, with a larger slope at the front and a relatively gentle slope of the rest of the beach with an average slope that varies between 0.5% and 4% (Fourie, 2010; Simms, 2013; Fitton, 2006).

However, the self-eroding nature of the tailings, aging and other physical processes such as sedimentation, segregation and desiccation translates into more erratic behavior at field scale for mature tailings.

After a certain amount of time, the tailings exiting the pipe will form a channel in the previous layer of tailings. Tailings will be transported that way until the end of the channel, where a new lobe will form. Figure 3 represents this process schematically. Eventually, the tailings in the channel will back up as they are blocked by the lobe and a new channel will form, repeating the process (MPA Williams, 2010).



**Figure 3: Sketch of self-forming channels in tailings flow (Adapted from MPA Williams, 2010)**

Proper management of stack geometry is necessary for several engineering priorities such as the optimization of space usage in the impoundment or the long-term dewaterability of the tailings. Therefore, understanding flow behavior of tailings, especially with the adoption of high-density deposition technologies, is necessary.

### 2.1.3 Beach slope prediction

In order to plan the use of the TSF for the duration of its lifetime, it is important to know the beaching properties of the material. Several semi-empirical methods have been developed in an attempt to provide engineering tools for daily operations in the industry.

Fitton (2006) proposed a method based on equilibrium conditions between erosion and sedimentation processes. The beaching profile and angle is estimated from properties of the channel in which tailings flow such as the hydraulic radius and average velocity. The equation assumes non-segregating, turbulent flow. Empirically, it was determined that

$$d = 12.2Q^{0.6} \quad (2.1)$$

where  $d$  is the depth of the channel in millimeters and  $Q$  is the flow rate in liters per second.

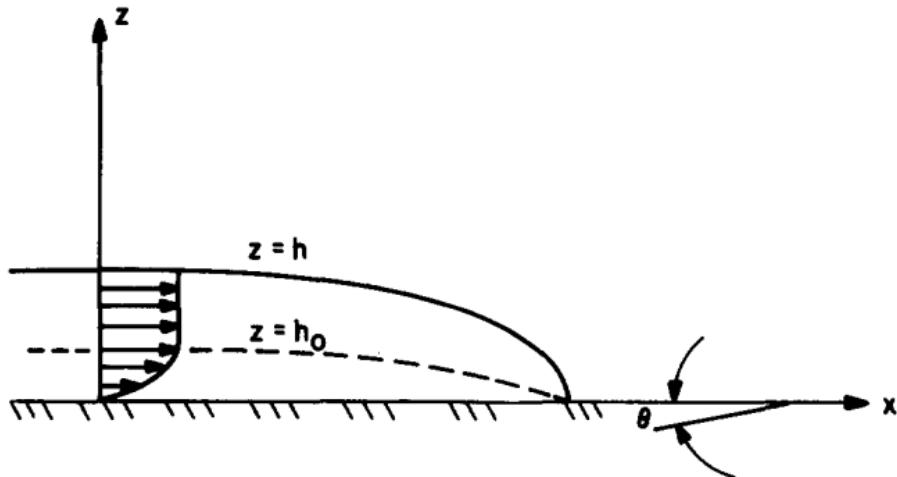
The predicted depth is then used to estimate a hydraulic radius and determine the beaching angle:

$$i = \frac{100 \tan(\arcsin(0.073V^2))}{2R_H g \left( \frac{8}{\tau_y + K \left( \frac{2v}{R_H} \right)} \right)^{0.25}} \quad (2.2)$$

where  $i$  is the slope in degrees,  $V$  is the critical velocity,  $\tau_y$  is the yield stress,  $R_H$  is the hydraulic radius,  $g$  is the gravitational constant,  $K$  is a parameter based on Herschel-Bulkley rheology and  $v$  is the average velocity in the channel.

Good agreements are found between predictions using Fitton's method and field scale results. However, the equation is limited to a subset of tailings flow by its assumptions of non-segregation and turbulent transport.

Lubrication theory is another approximation that can be used to predict stack geometry in TSF. By ignoring the inertia term in Navier-Stokes momentum and continuity equations, analytical formulations can be derived for cases under specific geometric conditions. Liu & Mei (1990) provide analytical equations for the slow spreading of thin plastic flow of Bingham fluid down an inclined plane as shown on the following sketch.



**Figure 4: Viscoplastic flow down an inclined plane (Adapted from Lui & Mei, 1990)**

The following equation represents the height of the plug flow as a function of the shear profile in the fluid:

$$\tau_y(x - x_o) = \frac{\rho g}{2} (h^2 - h_0^2) \quad (2.3)$$

where  $h_0$  is the height of the plug,  $\rho$  is the density and  $x$  is the longitudinal dimension. Previous research has proven this theory's applicability to deposition of tailings (Henriquez, 2009; Mizani & al., 2013). The predictions were accurate for single layer deposition. However, for multi-layer deposition, the profile got progressively more convex and the approximation of lubrication theory was not valid. Predictions were valid only for pipe deposition as the inertia present in dam break scenarios cannot be represented by lubrication theory (Gao & Fourie, 2018).

#### 2.1.4 Dam breaks

Tailings, the fine-grained slurry material resulting from mineral extraction, is typically discharged in a designated area known as the tailings storage facility (TSF) (Vick, 1983). This material with no economic value is typically discharge as a slurry, with a high water content. Characteristics of the slurry can vary greatly with original mineralogy, processing method and conditions during deposition.

Traditional deposition methods employ large dams and embankments to contain the slurry. However, due to poor control of the dam structures, they may fail and release the enclosed tailings with catastrophic consequences. With the ever-increasing size of mining operation in the last 100 years, tailings storage facility structures have grown proportionately. The risk associated with these dams has also increased as a function of their size.

As a result, catastrophic failure of TSF have increased in recent years (WISE, 2015) with significant environmental and economic impact. Loss of human life also occurs in a sizeable portion of these events (ICOLD, 2001; Rico & al., 2008).

Several geotechnical failure mechanisms can play a role in these events. Some of the most common occurrences are heavy rainfall, poor design and management (drainage structure, foundation, etc.) or problems with the natural soil under the TSF (Azam & Li, 2010).

To prevent such catastrophic events from happening, it is necessary to rethink engineering practices and update norms and regulations to better reflect the state of modern mining operations. Scenario analysis can provide useful insight during the design of a dam. However, determining the resulting runout requires good understanding of free surface flow of mine tailings.

One of the major difficulties is the determination of rheological properties of tailings as they flow after a dam break. Since the material has been in an immobile state for years, its properties can change significantly and might not be comparable to fresh tailings.

### **2.1.5 Slurry and debris flow modeling**

Advances in the field of geohazards management have led to similar strategies for flow modeling. A well researched model is DAN3D (Dynamic Analysis 3D) which has been used extensively to model runouts of landslides and debris flow (Hung & McDougall, 2009). The mathematical basis of the modeling tool relies on depth-averaging of the continuum equations. A semi-empirical approach is required as rheological parameters are back-fitted from previous landslide runout distances. Other tools relying on

similar methodologies have also been developed by researchers worldwide (Mergili & al., 2017; Hussin & al. 2012).

Timothy D. Stark and collaborators (Olson & Stark, 2002; Stark & al., 2005) have also accomplished work on using a geotechnical approach to runout modeling instead of a rheological one. Semi-empirical relationships were developed by comparing the ratio of liquefied shear stress to vertical effective stress. That way, a framework for liquefaction resistance was developed successfully and provided a way to estimate factors of safety for landslides and resulting runouts (Aaron & al., 2017; Chugh & Stark, 2005).

## 2.2 Rheology

Rheology is the science that studies fluid materials and soft solids that exhibit a particular set of characteristics: under an applied strain rate, their response is non-linear. This behavior contrasts with Newtonian fluids, like water or cooking oil, that exhibit a linear response to an applied effort.

Rheological materials, also termed non-Newtonian materials, are commonly found everywhere. Natural materials such as clay suspensions or biological fluids; household items like toothpaste or ketchup; industrial products and by-products like paint, fresh concrete and sludge are some examples of rheological substances.

### 2.2.1 Viscosity

For a fluid, the observed deformation behavior is linked to the relative movement of particles to each other at the microscopic scale dictated by physico-chemical interactions. For most applications in civil and environmental engineering, however, the macroscopic behavior is of interest. Observed physical deformations of a fluid is commonly described with a parameter called viscosity ( $\mu$ ) which links the rate of deformation (shear rate,  $\dot{\gamma}$ ) to the applied shear stress ( $\tau$ ). Viscosity is a scalar value commonly measured in Pascal-second (Pa·s) although several other units, such as the centipoise (cP) are often found in textbooks and viscosity value tables.

In one dimension, the shear rate can be expressed as:

$$\dot{\gamma} = \frac{\partial u}{\partial y} \quad (2.4)$$

where the differential term represents the local variation of velocity with height. The shear rate is measured with units of reciprocal seconds ( $s^{-1}$ ). The shear stress, measured in Pascals (Pa), is defined by a force (N) applied in parallel to a surface area of area ( $m^2$ ), as described in equation 2.5.

$$\tau = \frac{F}{A} \quad (2.5)$$

The viscosity of a material is highly dependant on its physico-chemical state. For example, in most fluids, viscosity can decrease by orders of magnitude for an increase in temperature. Several equations can be found in the scientific literature to describe the relation between the two.

### 2.2.2 Yield stress

Some fluids, described as viscoplastic, are able to withstand a certain level of stress before yielding and initiating flow. Mathematically, this can be described as:

$$\frac{\partial u}{\partial y} = \begin{cases} 0, & \tau < \tau_y \\ \frac{\tau - \tau_y}{\mu_y}, & \tau > \tau_y \end{cases} \quad (2.6)$$

In situations where stress is not constant, such as free surface flow of geomaterials, this can give rise to two zones of fluid with two drastically different behavior. The first region, where the shear stress is below the yield stress ( $\tau_y$ ), behaves as a solid material. This region is often referred to as a “plug”. The other region of the material, where yield stress exceeds that threshold, behaves as a fluid.

Although it proves itself to be an undeniably useful tool for a range of engineering applications, there are discussions when it comes to the physical existence of yield stress in the scientific community (Barnes, 2007).

### 2.2.3 Thixotropy

For materials exhibiting viscosity and yield stress properties, it can be said that their behavior is only dependant on their current stress state. However, to accurately model the behavior of some fluids, it is important to also consider their stress history.

A material is said to be thixotropic when its behavior is dependant on the development of its internal microstructure over time due to Brownian motion. When such a material is left at rest or at very low shear rates, structuration (aging) effects are allowed to develop (Mitchell & Soga, 2005). The gradual build-up of intermolecular bonding in the microstructure translates, at the macroscopic scale, into a progressive increase of apparent viscosity with time. Under certain conditions, the fluid's viscosity can even diverge towards infinite values and a gelled state can be observed.

Conversely, if a thixotropic material is sufficiently sheared, the inter-particular microstructural bonds will be destroyed as shearing effects become more important than aging effects (Mitchell & Soga, 2005).. The rate at which bonds are destroyed can be correlated to the intensity of the shear stress applied and the stress history. At a macroscopic scale, the material's observed viscosity will get lower with time. Some common thixotropic fluids are yoghurt and fresh concrete.

Thixotropic fluids have been an active area of research and several constitutive models have been developed with varying degrees of complexity to represent their behavior (Coussot, 2002; Barnes, 1997; Hewitt & Balmforth, 2013)

Coussot (2002) led experiments on gravity induced flow of thixotropic material on an inclined plane and illustrated experimentally the difference between a viscoplastic yielding fluid and a thixotropic fluid. Figure 5 shows the avalanche-like flow behavior of a clay suspension on an inclined plane. The length of the runout is plotted as a function of time. Results are compared to a simple viscoplastic constitutive model which does not, even on a qualitative basis, represent the behavior accurately.



**Figure 5: Comparison of experimental data and non-thixotropic viscoplastic model.**

Lastly, thixotropic materials do not have a yield stress per se. However, a diverging viscosity would closely resemble unyielded behavior as it is seen in yield stress fluids. In that sense, some of the physical characteristics and visual behavior of yield stress materials can be replicated with a thixotropic model, such as a non-yielding lump of material or the “pseudo-plug” occurring in free surface flows.

#### 2.2.4 Constitutive models

Several models have been proposed to quantify flow behavior of rheological substances. They can either include or not a yield stress and propose a law for evolution of the apparent viscosity as a function of shear rate. Several models can be used to describe a same material with different levels of accuracy. The choice of a model depends therefore on accuracy needs and availability of experimental data for calibration.

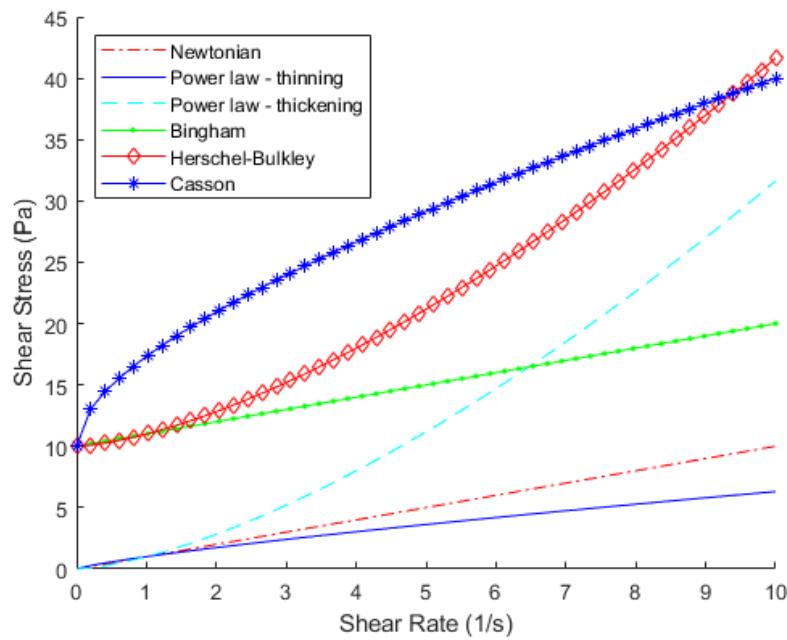
A qualitative description of each model, followed by the mathematical equations for shear stress, is presented in table 1 in order to compare their differences more easily.

**Table 1: Qualitative description of different types of rheological fluids**

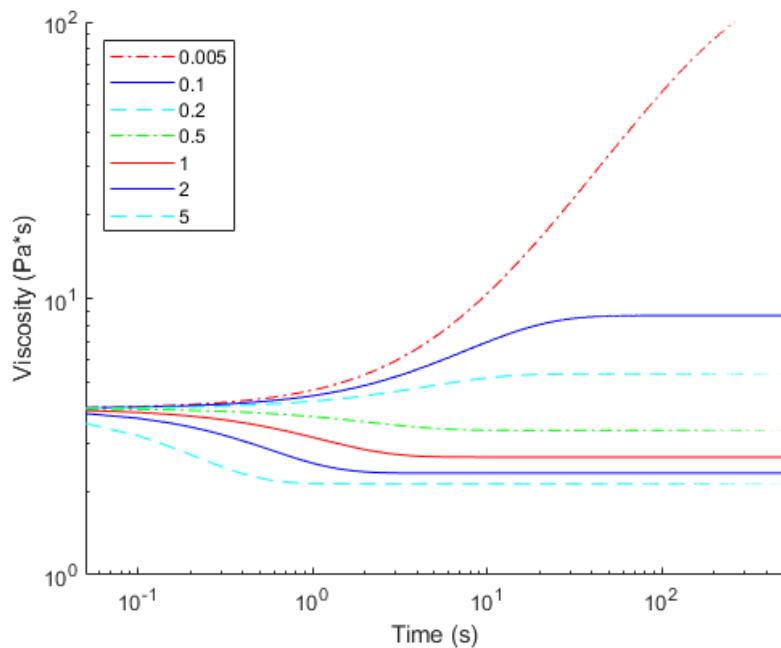
Fluid Type	Yield Stress ( $\tau_y$ )	Viscosity ( $\mu$ )	Material History	Equation	
Newtonian fluid	$\tau_y = 0$	Constant	None	$\tau = \mu \gamma$	(2.7)
Power Law: Shear thinning fluid	$\tau_y = 0$	Decreases as ; increases ( $0 < n < 1$ )	None	$\tau = \mu \gamma^n$	(2.8)
Power Law: Shear thickening fluid	$\tau_y = 0$	Increases as ; increases ( $n > 1$ )	None	$\tau = \mu \gamma^n$	(2.9)
Bingham fluid	$\tau_y > 0$	Constant	None	$\tau = \tau_y + \mu \gamma$	(2.10)
Herschel-Bulkley fluid	$\tau_y > 0$	Decreases ( $0 < n < 1$ ) or increases ( $n > 1$ ) as ; increases	None	$\tau = \tau_y + \mu \gamma^n$	(2.11)
Casson fluid	$\tau_y > 0$	Decreases as ; increases	None	$\sqrt{\tau} = \sqrt{\tau_y} + \sqrt{\mu \gamma}$	(2.12)
Coussot fluid	$\tau_y = 0$	Increases or decreases as a function of structure ( $\lambda$ )	Considered	$\tau = \mu \gamma$ $\frac{d\lambda}{dt} = \frac{1}{T} - \alpha \gamma \lambda$ $\mu = \mu_0(1 + \lambda^n)$	(2.13) (2.14) (2.15)

The equations 2.7 up to 2.15 are plotted as to qualitatively compare their evolution over an arbitrary range of shear strains in figure 6.

Figure 7 illustrates the evolution of viscosity in time for Coussot's thixotropic model at different shear rates.



**Figure 6: Comparison of common rheological models**



**Figure 7: Evolution of viscosity as a function of time for different shear rates; Coussot thixotropic fluid**

## 2.3 Numerical modeling

In this section, the main numerical modeling techniques will be briefly reviewed. A more in-depth review of the state of the art for the material point method will be given, as it is the focus of the present work.

### 2.3.1 Finite Elements Methods - Eulerian and Lagrangian descriptions

The two most common methods to discretize partial differential equations over their respective domains are by using Eulerian and Lagrangian schemes.

In the Eulerian method, the governing equations are discretized on a fixed grid through which the material can move freely. As more material accumulates or leaves a node, the relative size of a given node increases compared to its neighbours. Using a Eulerian mesh in continuum mechanics can easily deal with large deformation analysis because the grid is immobile, and no restrictions exist on the material's movement. However, it suffers major setbacks for certain applications.

Firstly, history dependency is hard to implement as material particles are not directly tracked. Secondly, the boundary nodes might not necessarily coincide with the material, making it difficult to apply boundary conditions on the material. Finally, there is a convective term present in the Eulerian formulation to describe the direction of material transport, which can be computationally expensive to handle (Zienkiewicz & al., 2005). Considering those characteristics, Eulerian methods are commonly used in fluid dynamics, both for gas and liquid phase problems.

In the Lagrangian frame of reference, the grid is attached to the material elements and gets deformed conjointly with the material. In this case, the setbacks of the Eulerian are greatly diminished as boundaries are easy to track and history-dependant constitutive models can be implemented. The main setback with the Lagrangian method is problems associated with mesh distortion for large deformation analysis (Belytschko & al., 2013).

Several advanced techniques have been developed for both the Eulerian and Lagrangian formulations to reduce their setbacks and analyze scenarios of increasing complexity such as the arbitrary Lagrangian-Eulerian (ALE). In this method, the mesh moves continuously and independently from the continuum as it deforms in an attempt to optimize shape functions and track boundaries (Hughes & al., 1981).

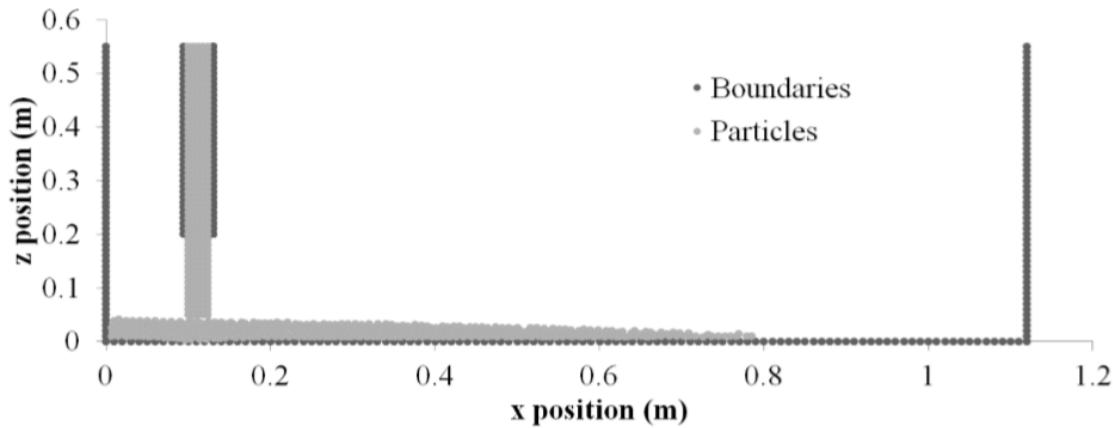
### 2.3.2 Mesh-free methods

Mesh-free methods are a set of numerical methods that discretize a continuum without relying on a background grid to map the information. Instead, the interaction between material elements are described using different techniques. Two of the most commonly used methods are the Discrete Elements Method (DEM) and Smoothed Particle Hydrodynamics (SPH).

SPH is mesh-free Lagrangian particle method that first appeared to simulate boundary-free problems in astrophysics (Gingold & Monaghan, 1977; Lucy, 1977). Since then, it has been extensively used in fluid dynamics and large deformation analysis. Physical quantities of every material point are updated using formulations that represent the domain of influence of particle. Weighted averaging methods can then be used to update the physical quantities of every node for the following timestep.

SPH also has its own setbacks. Firstly, enforcing boundary conditions is difficult. Immobile points are created to simulate boundaries, but it doesn't guarantee that they will be totally impenetrable. SPH is also computationally expensive and does not lend itself well to implicit solving methods due to the absence of a background grid. This can be problematic when timescales are an important factor in the engineering applications.

SPH was successfully used to compute mine tailings with Bingham rheology (Baboglu, 2014) to simulate multi-layer deposition of mine tailings. Kozemi & Simms (2017) showed SPH could also be used to model thixotropic tailings flow at small and large scale in two dimensions.



**Figure 8: SPH pipe deposition of Bingham mine tailings (Adapted from Babaoglu, 2014)**

## 2.4 Material Point Method (MPM)

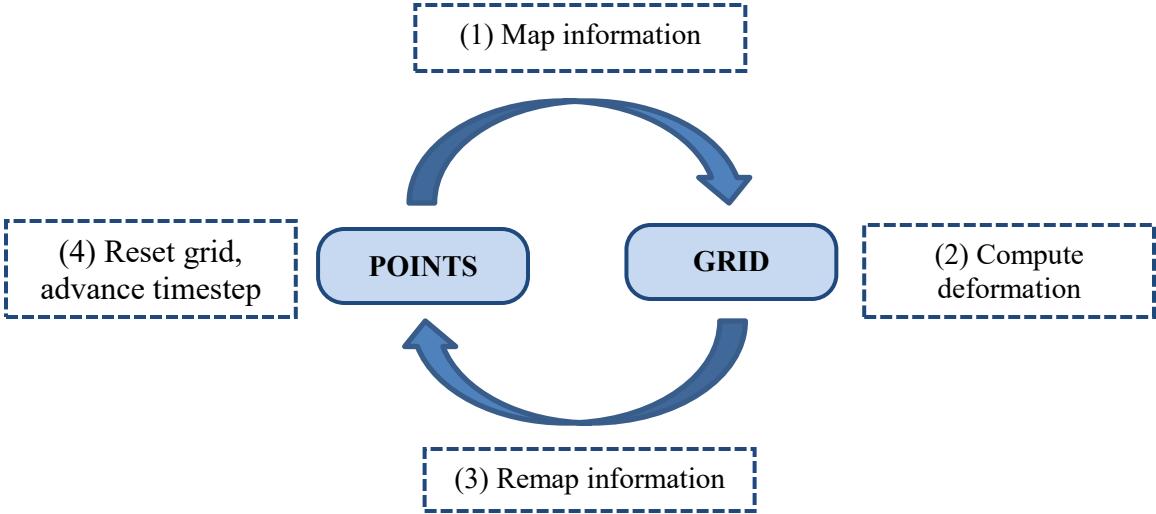
### 2.4.1 Summary of MPM

The material point method (MPM) is a particle-based finite element method (FEM).

Historically, it stems from particle-in-cell (PIC) (Harlow, 1964) and Fluid Implicit Particle (FLIP) (Brackbill & al., 1988). The methods have been improved and extended to both solids and fluids analysis by Sulsky & al. (1994) and has since then been referred to as the Material Point Method (MPM).

In the MPM, a continuum is discretized in a finite set of points that are tracked throughout the deformation process. All physical quantities such as the mass, density, velocity, Cauchy stress tensor and history parameters are stored in the points. Since the integration points' masses does not change in some formulations, mass conservation is enforced.

The MPM solution procedures follows the steps that are shown in figure 9. The steps can be split into four categories that represent a full iterative loop for a given time step.



**Figure 9: Solution procedure for the MPM**

After the initialization process, the first step is to transfer the information from the points to the nodes. The physical quantities are distributed to every node of the grid element in which the point is contained using weighted averaging procedures. For example, considering an 8-node cubic element, a material point situated exactly in the center of the element would assign 12.5% of its physical quantities to each surrounding node.

Once the information has been mapped from the particles to the grid, the second step is to solve the momentum equations in a Lagrangian FEM way for a given constitutive model. After the deformation is computed, the elements will deform, therefore updating the physical location of the points inside of it.

Then, information is mapped back from the grid nodes to the points, in a back-and-forth way with the mapping that occurs in first step.

The final step is to reinitialize the grid and update the information about points localization. For example, certain points could have crossed the grid at a given time which means that new active nodes and inactive nodes must be determined.

This numerical method is used for a wide array of applications. In the geoengineering field, landslide analysis (Andersen & Andersen, 2010), cracks propagation (Chen & al., 2002), pile driving (Phuong & al., 2016), multi-phase soil mechanics (Bandara & Soga, 2015), erosion and fluid-structure interaction (Onate & al., 2008; Parker & al., 2006) are some examples of problems that were tackled with the MPM. These developments were made possible by the fact that MPM overcomes some of the downfalls on traditional FEM analysis.

#### 2.4.2 Governing equations

The MPM solves the general set of equations of motion as with most computation methods. Under continuum mechanics and considering only mechanical forces (no heat transfers are considered), three conservation laws are used to describe deformation.

$$\frac{d\rho}{dt} + \rho \nabla \cdot \mathbf{v} = 0 \text{ (conservation of mass)} \quad (2.16)$$

$$\rho \frac{d\mathbf{v}}{dt} = \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{b} \text{ (conservation of momentum)} \quad (2.17)$$

$$\rho \frac{de}{dt} = \mathbf{D} : \boldsymbol{\sigma} \text{ (conservation of energy)} \quad (2.18)$$

where  $\rho$  is the density,  $\mathbf{v}$  is the velocity,  $\boldsymbol{\sigma}$  is the Cauchy stress tensor,  $\nabla$  is the gradient operator,  $\mathbf{b}$  is the specific body force,  $\mathbf{D}$  is the rate of deformation tensor.

The boundary value problems then consist of solving conservation equations for a deformed continuum according to a set of initial and boundary conditions

$$v(x, t = 0) = v_0 \text{ (initial velocity)} \quad (2.19)$$

$$\sigma(x, t = 0) = \sigma_0 \text{ (initial stress state)} \quad (2.20)$$

$$u = \bar{u} \text{ on } \Gamma_u \text{ (Dirichlet boundary condition)} \quad (2.21)$$

$$t = \bar{t} \text{ on } \Gamma_t \text{ (Neumann boundary condition)} \quad (2.22)$$

where Dirichlet boundary prescribes a finite value for the given physical parameter and Neumann conditions prescribe a finite value for the derivative of a physical parameter.

### 2.4.3 Spatial and temporal discretization

The material point method uses a discretization of momentum equation which is similar to the formulation in standard FEM analysis to account for non-smoothness of the solution. This is done by multiplying the momentum equation to a weighing function which is also known as the virtual work theorem:

$$\delta W = \delta W_{int} + \delta W_{kin} - \delta W_{ext} = 0 \quad (2.23)$$

$$\int_{\Omega} \rho \frac{\partial \delta u_i}{\partial x_j} \sigma_{ij}^s d\Omega + \int_{\Omega} \rho \delta u_i a_i d\Omega = \int_{\Omega} \rho \delta u_i b_i d\Omega + \int_{\Omega} \rho \delta u_i \underline{t}_i^s d\Gamma \quad (2.24)$$

where  $\Omega$  represents the current configuration of the continuum;  $\delta u_i$  is the test function (virtual displacement field);  $\sigma_{ij}^s$  is the specific stress tensor  $\sigma_{ij} / \rho$ ) and  $t$  is the specific traction tensor and  $b$  is the body forces. More details about weak formulation of conservation equations can be found in Belytschko & al. (2013).

In order to solve a boundary value problem, the continuum's domain is discretized into a certain number of points per cell. These "material points" are analog to Gauss integration points in finite element analysis. An appropriate number of points per cell (PPC) should be chosen to ensure convergence, while giving reasonable computation time.

This can be written in compact form as expressed by Newton's law

$$m\mathbf{a} = \mathbf{F}_{ext} - \mathbf{F}_{int} \quad (2.25)$$

where  $m$  is the mass,  $\mathbf{a}$  is the acceleration,  $\mathbf{F}$  is the force vector where the subscript "ext" refers to external forces (combination of body forces and traction) and the subscript "int" is the internal force vector resulting from the constitutive properties of the material.

The computation scheme starts, for a given timestep, by mapping all information from the points to the nodes to solve the continuity equations.

$$m_i = \sum_{p=1}^{n_p} S_{ip} m_p \quad (2.26)$$

$$\mathbf{v}_i = \frac{\sum_{p=1}^{n_p} S_{ip} m_p \mathbf{v}_p}{m_i} \quad (2.27)$$

$$\mathbf{F}_i^{ext} = \sum_{p=1}^{n_p} S_{ip} \mathbf{F}_p^{ext} \quad (2.28)$$

where the subscript " $p$ " is used to represent values associated with the particles. The subscript "i" is used to represent information that is saved under the background grid's nodes; This subscript notation is kept for the rest of the mathematical description unless stated otherwise.  $S_{ip}$  refers to the shape function of the node evaluated at the coordinates of the material points;  $\mathbf{v}$  is the velocity vector.

The last step to solve equation 2.29 is to compute the internal forces tensor.

$$\mathbf{F}_i^{int} = \sum_{p=1}^{n_p} G_{ip} \boldsymbol{\sigma}_p v_p \quad (2.29)$$

where  $G_{ip}$  is the gradient of the shape function of node  $i$  evaluated at point  $p$ ,  $\boldsymbol{\sigma}_p$  denotes the Cauchy stress tensor and  $v_p$  represents the volume.

Then, equation 2.30 is solved for acceleration of the grid

$$\mathbf{a}_i = \frac{\mathbf{F}_i^{ext} - \mathbf{F}_i^{int}}{m_i} \quad (2.30)$$

An explicit Eulerian method is used to advance the timestep

$$\mathbf{v}_i^{t+\Delta t} = \mathbf{v}_i + \mathbf{a}_i \Delta t \quad (2.31)$$

The updated grid velocity is then used to compute a velocity gradient  $\nabla \mathbf{v}$  for every particle. This velocity gradient will be used to compute the deformation. The equations in their discrete form are:

$$\nabla \mathbf{v}_p = \sum_{p=1}^{n_p} G_{ip} \mathbf{v}_i^{t+\Delta t} \quad (2.32)$$

$$\mathbf{dF}_p^{t+\Delta t} = (\mathbf{I} + \nabla \mathbf{v}_p \Delta t) \quad (2.33)$$

Then, the particle's volume and deformation gradient can be computed:

$$v_p^{t+\Delta t} = \text{Det}(\mathbf{dF}_p^{t+\Delta t}) \quad (2.34)$$

$$\mathbf{F}_p^{t+\Delta t} = \mathbf{dF}_p^{t+\Delta t} \mathbf{F}_p^t \quad (2.35)$$

The deformation gradient serves as an input to the constitutive model's set of equations. Solving these set of equations will be discussed in the third chapter of this thesis. The constitutive model will compute an updated internal stress state update for the particles.

Finally, the velocity and position information of the particles is updated according to the grid's deformation:

$$\mathbf{v}_p(t + \Delta t) = \mathbf{v}_p(t) + \sum_{i=1}^{n_i} S_{ip} \mathbf{a}_i \Delta t \quad (2.36)$$

$$\mathbf{x}_p(t + \Delta t) = \mathbf{x}_p(t) + \sum_{i=1}^{n_i} S_{ip} \mathbf{v}_i^{t+\Delta t} \Delta t \quad (2.37)$$

This set of equations complete the computation of one full timestep. Summarily, the MPM algorithm can be thought of as very similar to FEM with Gaussian integration points. The main difference is that, using MPM, these points are free to move.

After the grid is deformed, the particle's location is updated as well since their displacement is proportional to the deformation of the grid. However, once the grid is reset at the following time-step, the points' position does not get rolled back as well. This has the effect of simulating mass convection throughout the computational domain in a Eulerian way. This additional bookkeeping must be done to keep track of the particles' movements, especially when crossing element boundaries to ensure proper mapping and deactivate inactive elements to increase computational efficiency.

Both structured and unstructured grids are used in conjunction with the MPM (Wieckowski 2004, Jassim 2013). Unstructured grids provide the advantage of being able to deal with complex geometries and frictional contact problems more easily. However, tracking particles is more expensive as they cross the boundary of the element in which they reside. More complex - hence computationally expensive - algorithms need to be implemented to find which side of the element's face the particle has crossed and into which element it resides then.

By comparison, on a structured grid, a material point, given its coordinates  $\mathbf{x}_p = (x_p, y_p, z_p)$  can be assigned to an element of the mesh in a very straightforward way as the spacing on each of the three axis is known.

#### 2.4.4 Shape functions

In the MPM, traditionally, particles are considered as Dirac delta functions which means that the entirety of the mass and other physical properties associated to a small section of material are considered as a single point in space.

The shape functions are interpolation methods that describe mapping of information between the grid nodes and the material points. They serve as weighing functions to assign a certain percentage of physical values to the nodes of the elements as a function of its proximity.

Several methods exist, the most commonly used being “Generalized Interpolation Material Point” (GIMP) (Bardenhagen & al., 2004). Other alternatives such as “Convected Particle Domain Interpolation” (CPDI) (Sadeghirad & al., 2011), or high order B-spline functions are also available. Their particularities might make them interesting in particular scenarios. The most widespread method, GIMP, is described and compared with standard MPM to highlight the differences between both interpolation methods.

The effective shape function  $S_{ip}$  of a point  $\mathbf{x}_p$  is given by convolution of the characteristic function  $\chi_{ip}$  and the shape function  $S_i$  such as

$$S_{ip}(\mathbf{x}_p) = \frac{1}{V_p} \int_{\Omega_p \cap \Omega} \chi_p(x - \mathbf{x}_p) S_i(x) dx \quad (2.38)$$

The classical Dirac MPM characteristic function is

$$\chi_p(x) = \delta(x - \mathbf{x}_p) V_p \quad (2.39)$$

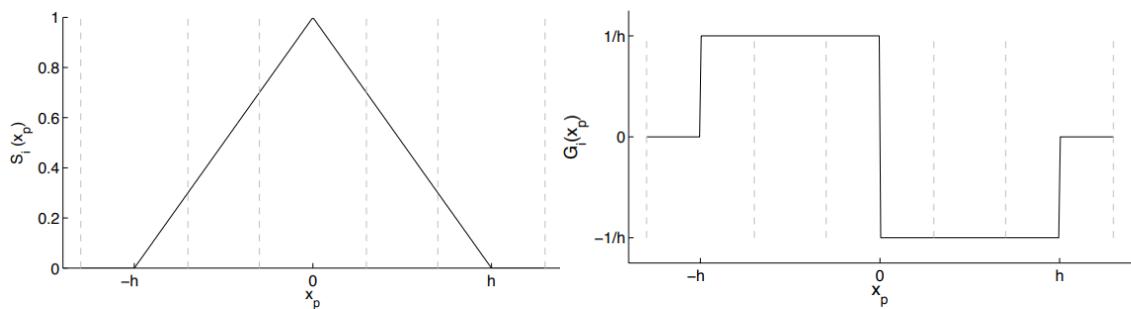
Although, technically, the user has latitude in choice of shape functions, a linear one is traditionally used such as

$$S_i(x_p) = \begin{cases} 1 + \frac{(x - x_i)}{h}, & -h < x - x_i \leq 0 \\ -\frac{1}{h}, & 0 < x - x_i \leq h \\ 0, & otherwise \end{cases} \quad (2.40)$$

After convolution of both equations in the three-dimensional form, the gradient of the effective shape function for traditional MPM can be expressed as

$$G_i(x_p) = \begin{cases} \frac{1}{h} - h, & -h < x - x_i \leq 0 \\ -\frac{1}{h}, & 0 < x - x_i \leq h \\ 0, & otherwise \end{cases} \quad (2.41)$$

Figure 9 presents a plot of equations 2.40 and 2.41 in a one dimensional field. The frequent discontinuities in the gradient function leads to poor numerical accuracy and even stability issues. Since the goal of MPM is to allow Lagrangian computation of large deformation problems, the stability of very mobile integration points is necessary as they can cross element boundaries several times during a computation.



**Figure 10: Shape function and its gradient for traditional MPM (Adapted from Guilkey & al., 2009)**

The Generalized Interpolation Material Point (GIMP) modifies slightly the mapping function near non-differentiable parts of the shape function. This is accomplished by introducing Heaviside's function (Guilkey & al., 2009) such as

$$H(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases} \quad (2.42)$$

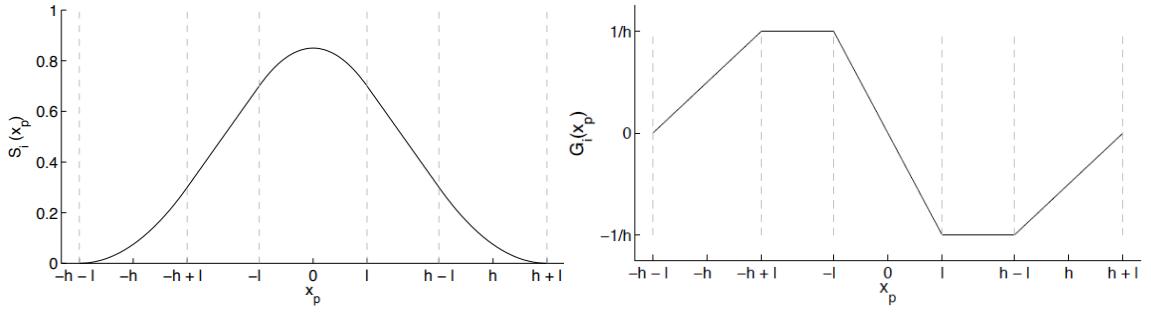
Therefore, the formulation of the characteristic function will be expressed as

$$\chi_p(x) = H\left(x - (x_p - l_p)\right) - H\left(x - (x_p + l_p)\right) \quad (2.43)$$

where  $l_p$  is the half-length of the particle. Then, convolution of equation 2.38 and 2.43 offers a new formulation for the effective shape function and its gradient, described respectively as

$$S_i(x_p) = \begin{cases} \frac{(h + l_p + (x_p - x_i))^2}{4hl_p} & -h - l_p < x_p - x_i \leq -h + l_p \\ 1 + \frac{x_p - x_i}{h} & -h + l_p < x_p - x_i \leq -l_p \\ 1 - (x_p - x_i)^2 + l_p^2 & -l_p < x_p - x_i \leq l_p \\ 1 - \frac{(x_p - x_i)}{h} & l_p < x_p - x_i \leq h - l_p \\ \frac{(h + l_p - (x_p - x_i))^2}{4hl_p} & h - l_p < x_p - x_i \leq h + l_p \\ 0 & otherwise \end{cases} \quad (2.44)$$

$$G_i(x_p) = \begin{cases} \frac{(h + l_p + (x_p - x_i))}{2hl_p} & -h - l_p < x_p - x_i \leq -h + l_p \\ \frac{1}{h} & -h + l_p < x_p - x_i \leq -l_p \\ -\frac{x_p - x_i}{hl_p} & -l_p < x_p - x_i \leq l_p \\ -\frac{1}{h} & l_p < x_p - x_i \leq h - l_p \\ -\frac{h + l_p - (x_p - x_i)}{2hl_p} & h - l_p < x_p - x_i \leq h + l_p \\ 0 & otherwise \end{cases} \quad (2.45)$$



**Figure 11: Shape function and its gradient for GIMP (Adapted from Guilkey, & al., 2009)**

Both equations are represented graphically in one dimension in figure 11. Although there are still discontinuities in the gradient, the gain in accuracy and stability is significant. The computational cost of using GIMP compared to MPM isn't important enough to justify using the traditional linear formulation of MPM over GIMP.

Other shape functions, especially methods that stem from Convected Particle Domain Interpolation (CPDI) offer promising results. The advantage of CPDI is that the material points are considered as deformable parallelogram. Therefore, the volume considered by the points can not overlap or be discontinuous (Sadeghirad & al., 2011).

#### 2.4.5 Overview of MPM characteristics

Nguyen (2014), gives an extensive description of the advantages and inconveniences of using the material point method over a FEM scheme. Some of the advantages that should be highlighted are:

- Any mesh distortion associated with Lagrangian FEM is eliminated, opening the door to large strain analysis;
- Problems that involve tracking interface or free surfaces are easily solved;
- No-slip and no penetration is enforced by default which facilitates modeling granular materials and applying certain types of boundary conditions;
- Boundary conditions are easily applied, unlike mesh-free methods;
- Solving fluid-structure interaction problems can be straightforward because the background structured grid can be used as an Eulerian grid for the fluid phase.

However, some of the downsides of the MPM are as follows:

- Constantly mapping data back and forth between particles and grid nodes increases computation time;
- More fast access storage (RAM) needed to keep track of both particles and a grid;
- Numerical artefacts can occur as the particles move. Some sudden increase in kinetic energy (having the effect of creating some “explosions” in the material) can occur if the implementation isn’t done carefully.
- Visualisation of data can be more challenging, as with meshless and other particle-based methods like SPH.

- No convergence test is currently widespread to test numerical accuracy (analogs to method of manufactured solutions in FEM). However, comparing with analytical equations can still be suited to validate numerical properties.

Other characteristics should be considered when considering MPM as a candidate to model rheological materials with tailings management priorities in mind:

- Geotechnical models are commonly used in MPM, opening the door to transition between rheology and geotechnical constitutive models;
- MPM is used to model crack propagation in several fields (rock mechanics, wood engineering), a phenomenon that occurs in drying tailings;
- The Eulerian-Lagrangian framework can include water and air physics in the Eulerian part for unsaturated flow or consolidation modeling or to consider the material as a 2 or 3 phase element (soil, water and air);
- Anisotropy can be included in the constitutive models with relative ease.

## 2.5 Summary

Prediction of beach profile of tailings are necessary to plan daily operations of the tailings storage facility (TSF). Also, recent catastrophic failures of TSF's retaining structures and the resulting runout of tailings has increased scientific interest in runout modeling for dam breach scenarios.

Several methods have been developed to determine the slope angle and the extent of the beach with semi-empirical tools. Fitton (2006) proposed a solution based on equilibrium condition as tailings flow in their channel. Hydraulic radius is evaluated empirically with properties of the tailings that are easier to measure.

For high-density tailings, lubrication theory has also been applied successfully to tailings modeling problems (Henriquez, 2009). Lubrication theory relies on an approximation of the Navier-Stokes equation in which the inertia term is ignored. Analytical equations can then be derived for particular geometric and initial conditions. This method was applied successfully, although discrepancies were found with dam breaks of a certain height since the inertia was becoming non-negligible.

Since tailings are not Newtonian materials, particular rheological models must be developed to explain their behavior. Most commonly, a Bingham rheology is considered, where the material does not deform under a certain stress threshold (Babaoglu, 2014; Henriquez, 2009). However, more advanced rheological models have proven to be a better fit for a broader range of conditions (Mizani & Simms, 2017). Thixotropic rheology relies on a stress-history dependant structure parameter to dictate the evolution of the material (Hewitt & Blamforth, 2013; Coussot & al., 2002).

Different numerical methods to solve the equations of motion in a discrete way and their capacities to model large scale thixotropic flows for mine tailings management application has been evaluated. Although SPH is a viable candidate, the Material Point Method (MPM) provides better scaling opportunities for 3D analysis by relying on a background grid (Sulsky & al., 1994).

This work investigates the capability of MPM to model rheological flow scenarios for tailing management applications. Dam breaks and pipe deposition are two scenarios in which flow behavior is important to understand. Both Bingham and Coussot's constitutive equations can be implemented in an MPM framework and their behavior can be compared to better the understanding of tailing's deformation behavior in the fluid domain.

### **3: Modeling Methodology**

In the following section, the Uintah-MPM framework will be discussed in detail. Afterwards, the numerical implementation of thixotropy as a constitutive model will be discussed, both from a mathematical and algorithmic point of view.

#### **3.1 Uintah-MPM framework**

The Uintah framework provides several numerical analysis tools for modeling purposes in different engineering fields. This open-source code developed by the University of Utah is written in object-oriented C++. For geotechnical engineering, three frameworks are worthy of mention. Uintah-MPM consists of a material point method code that proves useful for large deformation analysis of solids. Uintah-ICE (Implicit Continuous-fluid Eulerian) provides a full-physics solver for computational fluid dynamics (CFD) problems. It contains treatment of thermodynamics, fluid-structure interaction, phase change and turbulence.

A third software, Uintah-MPMICE, combines both frameworks to allow complex, multi-material analyses. The MPM portion takes care of the deformation of solid materials while the Eulerian part, more suited for fluid dynamics, updates the state of the liquid or gas portions.

In the context of this thesis, only the Uintah-MPM framework has been investigated and modified for the purpose of modeling free-surface thixotropic flows. Therefore, all instructions and descriptions of the software concern solely the Uintah-MPM portion.

### 3.1.1 Compiling the Code

In order to compile the Uintah-MPM framework, the user needs to have a Linux-based machine. Ubuntu is recommended as most library dependencies can be readily obtained. The following build instructions are for Ubuntu systems. CentOS is also a good alternative to use. It facilitates installation of software visualization tools (VisIt), as the binaries are available for download and have the required plugins to read Uintah-MPM data files directly.

Firstly, the source code can be downloaded from their website (<http://www.sci.utah.edu/download/uintah/>), or checked out directly using SVN.

```
svn co https://gforge.sci.utah.edu/svn/uintah/trunk Uintah
```

The second step is the install the required software dependencies and libraries that are used by Uintah-MPM.

```
sudo apt-get install subversion libhypre-dev petsc-dev \
libxml2-dev zlib1g-dev liblapack-dev cmake libglew-dev \
libxmu-dev g++ gfortran libboost-all-dev git \
libxrender-dev libxi-dev
```

Then, two build configurations can be made. The first configuration file is for debugging and development purposes; it contains static libraries and prints most debugging and error messages. GNU-Debugger or, more conveniently, VSCode can be used as development environments.

To prepare the debug configuration, the following command lines can be input where “path/to/Uintah-2.1.0/” refers to the folder in which the source code has been downloaded. The last command line builds the binaries where the optional argument “-j” followed by an integer specifies how many cores are to be used to build the binaries and accelerate the process.

```
cd /path/to/Uintah-2.1.0/
mkdir dbg
cd dbg
../src/configure --enable-mpm --with-mpi=built-in \
--enable-wasatch_3p --with-boost=/usr \
--with-mpi=/usr/lib/x86_64-linux-gnu/openmpi F77=gfortran \
--enable-debug --enable-static
make -j 16
```

A similar set of command prompts are used when building optimized binaries. They will provide minimal debugging and error messages information, but build the software in a way to make computations as fast as possible.

```
cd /path/to/Uintah-2.1.0/
mkdir opt
cd opt
../src/configure --enable-mpm --with-mpi=built-in \
--enable-wasatch_3p --with-boost=/usr \
--with-mpi=/usr/lib/x86_64-linux-gnu/openmpi F77=gfortran \
--enable-optimize
make -j 16
```

Once the code is compiled successfully, the executable, `sus.o`, which stands for Standalone Uintah Simulator, can be found in the folder called `StandAlone`.

### 3.1.2 Running the code

In order to run a simulation, an input file of XML format needs to be created. This file will provide all the necessary information to the simulation in a convenient and user-friendly way. It includes: the constitutive models and its parameters; the geometry of the material; the numerical parameters; and information about the desired output data saved. The input files use the extension UPS, which stands for Uintah Problem Specification. The creation of input files will not be discussed. However, more information on their creation can be obtained in the Uintah user guide (Guilkey & al, 2009). An example of input file for a simple dam break case is provided in Annex 2.

To run simulations on a local computer, the executable file is called with the UPS input file as an argument.

```
cd /path/to/Uintah-2.1.0  
cd opt/StandAlone  
.sus InputFile.ups
```

For parallel runs, the message parsing interface is used with the argument -np followed by an integer to specify the number of processors used. When the simulation is initiated using a remote connection to a server, the “nohup” command must also be specified in order to keep the computation going on after the user disconnects. The resulting command lines should be as follows.

```
cd /path/to/Uintah-2.1.0  
cd opt/StandAlone  
nohup mpirun -np 16 sus InputFile.ups &  
exit
```

### **3.2 Pre-processing**

In this section, particle creation strategies for MPM and more particularly the Uintah-MPM framework will be discussed. Strategies to simulate continuous deposition will also be considered

#### **3.2.1 Geometry initialization**

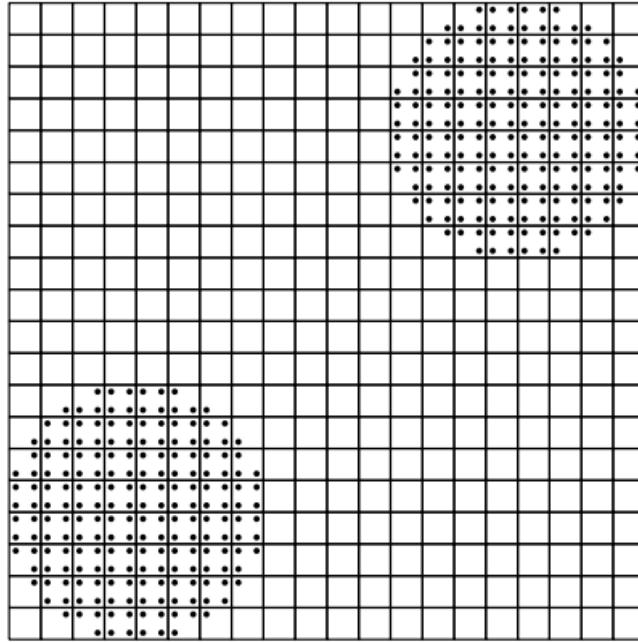
In the MPM, the continuum is discretized in a cloud of points over a regular or irregular grid. For the remainder of this thesis, only regular grids will be discussed as they provide characteristics that are more appropriate to solving the engineering issues at hand.

Care should be taken while selecting parameters for the grid cell's dimensions and number of points per cell. Current practice recommends at least  $2N$  points where  $N$  is the number of dimensions as to take full advantage of the particularities of MPM over FEM. Sensibility analysis comparing the grid refinement and points per cell can be conducted to investigate the trade-off between the two. On the other hand, tracking the simulation's runtime can further improve optimization.

For simple geometries, the cloud of points can be produced by incorporating basic shapes commands in the modeling tool, which is already done for most open source or commercial packages of MPM software. Furthermore, incorporating union “U” and intersection “ $\cap$ ” commands can increase the range of shape created to slightly more complex geometries. A wide range of applications can be created using these input-file commands.

By specifying basic properties of geometrical shapes, points can be created in a straightforward, analytical way. Cells inside the domain are fully filled by particles. For

cells in which the boundary of the element falls, they can be partially filled to accurately represent the modeled shape. Figure 12 shows initial configuration of two colliding disks with four points per cell (PPC). Partially filled elements makes it possible to represent a circle on a rectangular grid with more accuracy.



**Figure 12: Initial particle distribution for two colliding disks (Adapted from Guilkey & al., 2009)**

To create advanced or very complex geometries, digital images can be used to create the cloud of points. For example, in biomechanics, a CT scan of a human skeleton can be used to create a 3D model by successively transforming pixels of 2D slices of the skeleton in material points. Other examples of applications where this technique would be used in earth science would be by using microscopy images of granular materials or force

chain structures to compute small-scale deformation simulations. Topographical land-survey images could possibly be used to define shapes in large-scale simulations.

### 3.2.2 Continuous deposition

For pipe deposition scenarios, continuous particle insertion strategies must be developed. This is also necessary for different engineering applications that involve extrusion.

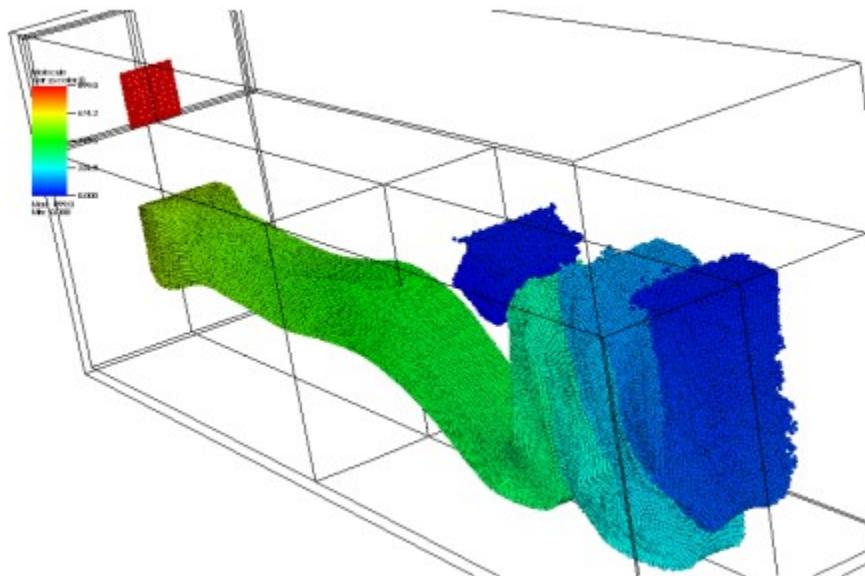
The MPM allows continuous insertion of particles fairly easily as the particles and the grid can be managed independently. However, the insertion of material needs to be discretized in time. The description that follows is the method used in Uintah-MPM as it is the software used in the context of this thesis.

The particularity with Uintah-MPM is that creation of particles can only occur at the initialization phase of the computation (i.e. at time  $t = 0$  s), unfortunately. Other MPM frameworks might have modules implemented that allows continuous creation of points. One of the advantages of such procedure would be a significant reduction of computational times at the initial phases of the insertion as the total number of material points would be minimal. This is one of the main disadvantages of the Uintah-MPM framework and should be one of the main targets of future work to increase computational efficiency.

To approximate continuous insertion of material, a translation function is used on successive discrete blocks of material. Every block of material is assigned a number and stored in a remote section of the computational domain which serves as a waiting area. . In the input file, the particle insertion specification file is given which provides information about block translations. At a given timestamp, a block with a specified number is

translated out of the “waiting area” into the actual computational domain by specifying a displacement vector. Initial properties such as the velocity can also be reset if desired.

Figure 13 provides a visual example of an extruded rubbery material deforming on itself. The overlapping blocks of material in the waiting area, represented in red, are progressively translated down to the extrusion tip, where green particles are located. Finally, the material deforms on itself once it reaches the boundaries of the computational domain as its initial velocity was specified. Some challenges arise from using such a method when incorporation weak materials and gravity fields as well as aging-dependency. The algorithm description in section 3.4 gives more detail about how these issues were overcome.



**Figure 13: Extrusion simulations with Uintah-MPM (Adapted from Guilkey & al., 2009)**

To provide a regular output of particles, it is a good idea to keep blocks small and moved at a timeframe in the order of seconds or fraction of sections. This can result in

input files of several hundred to thousands of blocks. For efficiency, a VBA-Excel code was written to create such files programmatically.

### 3.3 Post-processing

Once the computation is completed, different strategies are used to extract, analyze and present the data.

#### 3.3.1 Parsing tools

To treat data more efficiently with dedicated post-processing software such as Matlab or Excel, several tools are available in the Uintah libraries to facilitate data extraction.

The most commonly used command is “puda” which stands for Parsing Uintah Data Archives. It will print out the desired information about the simulation and desired timesteps into the Linux terminal window. It is more convenient to save it as a text file, using the following command lines, where “VarName” and “FolderName” refers to the name of the variable to be extracted and the folder containing the data archives under UDA format, respectively.

```
cd path/to/Uintah-2.1.0  
cd opt/StandAlone  
.puda -partvar VarName FolderName > OutputData.txt
```

Some physical quantities might be easier to visualize when evaluated at the grid nodes instead of at the integration points. To extract grid data, the command `lineextract`

is used for lines or three dimensional sections of the computational domain evaluated at grid nodes.

```
cd path/to/Uintah-2.1.0
cd opt/StandAlone
./lineextract -tlow N -thigh N \
-istart N N N -iend N N N FolderName > OutputData.txt
```

In the above code snippet, the letter N is used to refer to an integer. The arguments -tlow and -thigh indicate the first and last saved timesteps from which to extract the data. The arguments -istart and -iend indicate the grid elements that will be subjected to the extraction.

As with the point-based extraction method, the output is saved as a text file and can be more easily managed by the user.

### 3.3.2 Advanced rendering

One of the main challenges with particle-based continuum modeling tools is rendering data in a way that is clear, straight-forward and easily understandable to engineers and decision-makers.

The best rendering practices are to convert the cloud of points back to a surface to represent the continuum. This can be efficiently accomplished by using anisotropic kernel density functions (Yu & Turk, 2013). These methods are time-consuming and not

necessary for most engineering applications; they are however an active research area of numerical modeling for the graphics industry (movies and video games).

### 3.3.3 Point-based visualisation

Point-based visualisation refers to a set of techniques where the integration points in MPM (or particles in SPH) are modelled without converting the scattered points back to a surface. In the context of this thesis, MatLAB has been used as a post-processing software. The steps to convert the data extracted with “puda” command described in chapter 3.3.1 to a 3D plot is detailed. The following MatLAB code snippets can easily be adapted to plot any type of particle-based data from other modeling tools.

Firstly, the raw data in the output file Output.txt needs to be filtered and separated by timestep. Given N as the number of particles and T as the number of timesteps, a mother file with  $N \times T$  rows of data will be split in T files of N rows, which can be done with the following lines in MatLAB.

```
%input file location
filnamXYZ = "C:\Users\etienne\Desktop\OutputXYZ.txt";
filnamVis = "C:\Users\etienne\Desktop\OutputViscosity.txt";
folderOut = "C:\Users\etienne\Desktop\plots";

delimiter = " "; %separates input file columns
headerlines = 15; %qty of header lines in input

%load the data in MatLAB
iniXYZ = importdata(filnamXYZ,delimiter,headerlines);
iniVIS = importdata(filnamVIS,delimiter,headerlines);

%finds all unique time values
totSteps = unique(iniXYZ.data(:,1));
nSteps = size(totSteps(:));
nSteps = totSteps(1,1);

%creates structure objects
```

```

sXYZ = struct;
sVIS = struct;

%split the data by timestep in the structure objects
for j=1:totSteps
    sXYZ(j).val = iniXYZ.data(:,1)==totSteps(j),:);
    sVIS(j).val = iniVIS.data(:,1)==totSteps(j),:);
end

```

Once the data is split, specifying inputs for different plot types is easily done. It should however be noted that this method is limited by RAM capacities of the computer. For input files in the order of gigabytes, a first manual split in two or three files should be done.

The simplest, although not the most visually pleasing type of plot is simply a 3D scatter plot using the data of points. By changing the camera or using the “scatter” command instead of “scatter3”, two-dimensional cross-sections on any plane can be rendered by the following code snippet.

```

for k = 1:maxplot
    lenX = 10                      %length of plot image (inches)
    lenY = 4                        %height of plot image (inches)
    sz = 8                          %size of points on the plot
    f = figure(k)                   %opens a new figure object

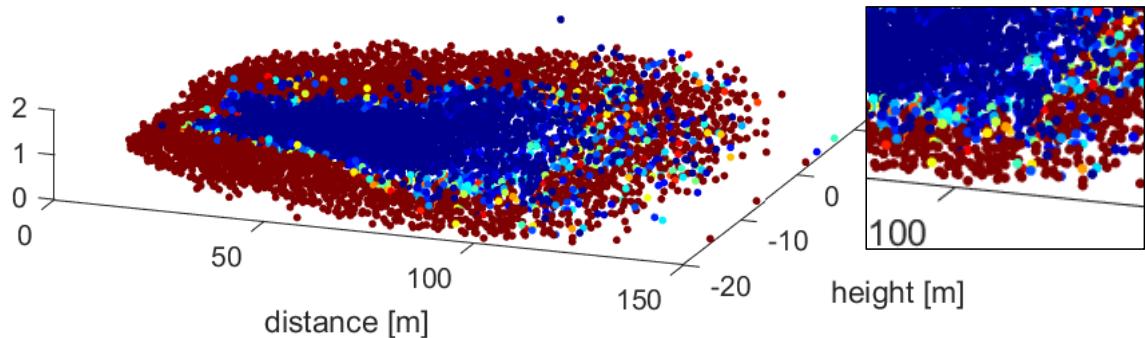
    set(gcf, 'PaperUnits', 'inches', 'PaperOrientation', ...
          'portrait', 'papertype', 'usletter', ...
          'paperposition',[1 1 lenX lenY]);

    scatter3(sXYZ(k).val(:,4), ...
              sXYZ(k).val(:,6), ...
              sXYZ(k).val(:,5), sz, ...
              sVIS(k).val(:,4), ...
              'filled');
end

```

Figure 14 shows an example of plot rendered using the scatter3 function in MatLAB. It should be noted that the main loop described above can be substantiated to

include axis titles and limits, camera's viewing angle and other elements. The disadvantage of using this method is that there is no possible modification of the rendered points besides color. Spatial distribution of points can be harder to grasp due to the lack of depth perception. No lighting nor texturing options can be applied which makes determining the exact location of the points in a 3D space somewhat difficult. For 2D simulations, however, it is largely satisfactory.



**Figure 14: Visualization of point-based data using MatLab scatter3 plot**

A more visually pleasing method is to render every point individually as a sphere. A custom function bubbleplot3 can be obtained from MatLAB's file sharing central on their website. The inputs for BUBBLEPLOT3 ( $x, y, z, r, c, \text{Alpha}$ ) are respectively the center coordinates, the radius, color and transparency. For every point, three values should be input for the radius. Plotting ellipsoids will counteract the effect of axis scaling. This plotting technique can be completed by the following algorithm in MatLAB.

```

for k = 1:totSteps

    dia      = 0.35;           %diameter of plot spheres
    scFac   = 1/20;            %scaling factor for the elevation
    alpha   = 0.75;            %transparency (1.0 is opaque)
    maxVis = 250;              %max viscosity for color scale
    FS      = 10;               %font size

```

```

col      = 32;           %number of shades in color scale
camHt   = 45;            %camera's elevation (degrees)
camBeg  = 20;            %beginning azimuth for camera's
                         view
camEnd  = 160;           %ending azimuth for camera's view
                         %put same value for immobile
                         camera

%build a custom color scale
color = zeros(length(sXYZ(k).val(:,4)),3);
colmap = colormap(parula(col));
cutOffs = zeros(length(sXYZ(k).val(:,4)),1);
cutOffs = round((sVIS(k).val(:,4)/maxVis)*(col-1));

%transforms parameters in array form
diameter = zeros(length(txyzvals(k).val(:,4)),3);
diameter(:,1)=1;
diameter(:,2)=1;
diameter(:,3)=scFac;
diameter = dia*diameter;

camAngles = linspace(camBeg,camEnd,totSteps);

for j = 1:length(txyzvals(k).val(:,4))
    color(j,1) = colmap(modulos(j)+1,1);
    color(j,2) = colmap(modulos(j)+1,2);
    color(j,3) = colmap(modulos(j)+1,3);
end

%plot the spheres
bubbleplot3(sXYZ(k).val(:,4), ...
            sXYZ(k).val(:,6), ...
            sXYZ(k).val(:,5), ...
            diameter,color,alpha);

%adjust axis and camera angles
ax = gca;
ax.DataAspectRatio = [1,1,scFac];
view(camAngles(k),camHt);

grid on
grid minor

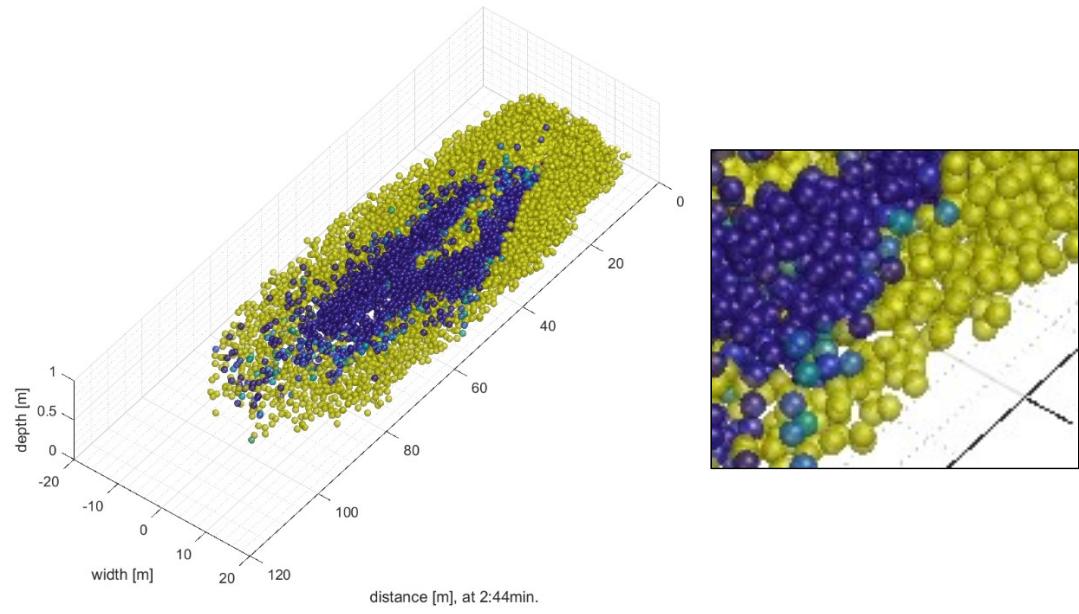
%add lighting effects
camlight;
lighting gouraud;

end

```

Finally, regardless of the plotting method chosen, the results will need to be saved as image files. The following code excerpt can be introduced at the end of the main loop that plots every frame. Figure 15 shows an example of plot using this advanced rendering methodology. This operation can be completed by the following snippet.

```
for k = 1:totSteps  
  
    %save the file as a png in a given folder  
    outfile = strcat(folderOut, "\plot", num2str(k));  
    saveas(figure(k), outfile, 'png');  
    close  
end
```



**Figure 15: Advanced visualization of point-based data**

### 3.4 Implementation of constitutive models

Two new constitutive models have been implemented into the Uintah-MPM framework – one for Bingham rheology and another one for thixotropy. The mathematical and algorithmic approach will be described in detail. The general formulation is based on the SPH Galerkin formulation for fluids proposed by Cueto-Felgueroso & al. (2004) with some adjustments for increased numerical stability.

#### 3.4.1 Thixotropic constitutive equations

The steps to solve the Cauchy stress tensor at every time step is presented. Equations along with code excerpts are presented for every step to facilitate referencing. All the equations are contained within the `computeStressTensor` method. More details about the algorithm as a whole are presented in the following section.

At each time step, a Cauchy stress tensor is computed, which holds the form

$$\boldsymbol{\sigma} = -p\mathbf{I} + 2\mu\mathbf{d}' \quad (3.1)$$

```
// viscous part of the stress
    Shear = DPrime*(2.*pViscosity_new[idx]);
// compute the total stress (volumetric + deviatoric)
    pStress_new[idx] = (Identity*(-p) + Shear);
```

where  $p$  is the hydrostatic pressure,  $\mu$  is the viscosity, and  $\mathbf{d}'$  is the deviatoric part of the rate of deformation tensor ( $\mathbf{d}$ ), which is given by

$$\mathbf{d}' = \mathbf{d} - \frac{1}{3} \text{tr}(\mathbf{d})\mathbf{I}, \quad \mathbf{d} = \frac{1}{2}(\nabla\mathbf{v} + \nabla\mathbf{v}^T) \quad (3.2)$$

```
// Calculate rate of deformation D, and deviatoric rate DPrime,
Matrix3 D = (velGrad[idx] + velGrad[idx].Transpose())*0.5;
Matrix3 DPrime = D - Identity*onethird*D.Trace();
```

The deformation gradient is taken to be the symmetric part of the velocity gradient, which is common in viscoplastic modeling or linear elastic applications.

Under matrix notation, the Cauchy stress tensor, identity matrix and velocity gradient can be represented respectively as

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix} \quad (3.3)$$

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

$$\nabla \mathbf{v} = \begin{bmatrix} \frac{\partial v_x}{\partial x} & \frac{\partial v_x}{\partial y} & \frac{\partial v_x}{\partial z} \\ \frac{\partial v_y}{\partial x} & \frac{\partial v_y}{\partial y} & \frac{\partial v_y}{\partial z} \\ \frac{\partial v_z}{\partial x} & \frac{\partial v_z}{\partial y} & \frac{\partial v_z}{\partial z} \end{bmatrix} \quad (3.5)$$

A thixotropic rheological model was used to compute the viscosity. As opposed to standard models of Newtonian fluid where viscosity is a constant parameter, it evolves with every time step as a function of material shear history. Several constitutive laws have been developed to express evolution of microstructure in the fluid to different degrees of detail. However, only the Coussot model (Coussot, 2002) has been implemented.

$$\frac{d\lambda}{dt} = \frac{1}{T} - \alpha \dot{\gamma} \lambda \quad (3.6)$$

$$\mu = \mu_0(1 + \lambda^n) \quad (3.7)$$

```
double viscosity = minvis * (1.0 +
    pow(pStructure_new[idx], expN));
pViscosity_new[idx] = viscosity;
```

In three dimensions, the second invariant of the deviatoric part of the rate of deformation tensor ( $II_{d'}$ ) is used as a uniform measure of strain, regardless of the direction of principal stresses, which can be obtained by

$$II_{d'} = \frac{1}{2} (tr(\mathbf{d}')^2 - tr(\mathbf{d}'^2)) \quad (3.8)$$

```
//get the second invariant of the deviatoric part
double SecInvar = DPrime(0,0)*DPrime(1,1) +
    DPrime(1,1)*DPrime(2,2) +
    DPrime(2,2)*DPrime(0,0) -
    DPrime(0,1)*DPrime(1,0) -
    DPrime(1,2)*DPrime(2,1) -
    DPrime(0,2)*DPrime(2,0);
SecInvar = abs(0.5*SecInvar);
```

Under its discretized form, the structure parameter can be solved numerically as follows in order to find the updated viscosity for the current timestep.

$$\lambda^{t+\Delta t} = \lambda^t + \Delta t \left( \frac{1}{T} - \alpha I I_d' \lambda^t \right) \quad (3.9)$$

```
//Compute thixotropic viscosity
double delLambda = delT * ((1.0/tParam) -
                           alpha*SecInvar*pStructure[idx]);
pStructure_new[idx] = pStructure[idx] + delLambda;
```

Then, Cueto-Felgueroso & al. proposes an equation of state to determine the hydrostatic part of the stress tensor.

$$p = \kappa \left[ \left( \frac{\rho}{\rho_0} \right)^\gamma - 1 \right] \quad (3.10)$$

where commonly used values in the literature for  $\gamma$  are 1.5 for air and 7 for water. However, this equation was linked to numerical instabilities when implemented into the Uintah-MPM framework due to numerical oscillations of hydrostatic pressure. Therefore, Tait's equation of state (Macdonald, 1969) was used instead.

$$p = C \cdot \kappa \left[ \exp \left( \frac{1}{C} \cdot (1 - J) \right) - 1 \right] \quad (3.11)$$

```
// get the hydrostatic part of the stress
double cconst = 0.0894;
p = cconst*bulk*(exp((1 / cconst)*(1 - J)) - 1);
```

The bulk modulus  $\kappa$  is chosen so that the fluid is nearly incompressible while still allowing for time steps of reasonable size.

For the Bingham rheological model, the general solution procedure is similar, but viscosity is updated according to a different set of rules. Out of the different Bingham formulations, a bi-viscosity model was used (Hosseini & al., 2006).

The bi-viscosity model can be conceptually described as follows. For the yielded portion of the material, a given viscosity value is used to calculate the Cauchy stress tensor (Eq. 3.1). For the unyielded portion, the viscosity is multiplied by a constant to simulate a highly viscous material that is quasi-indeformable. The advantage of using this method is to provide a smoother transition between the yielded and non-yielded domains. Therefore,

$$II_{d'} \leq \frac{\tau_y}{2\alpha\mu} \rightarrow \boldsymbol{\tau} = 2\alpha\mu \mathbf{d}' \quad (3.12)$$

$$II_{d'} > \frac{\tau_y}{2\alpha\mu} \rightarrow \boldsymbol{\tau} = \left( \frac{\tau_y}{II_{d'}} + 2\mu \right) \mathbf{d}' \quad (3.13)$$

using an alpha value of at least 100 is recommended (Hosseini & al., 2006). In the code, this is represented by:

```
// if not sheared enough, viscosity = minvis*alpha
// else, viscosity = minvis

if(SecInvar<(viscmax/(2*alpha*minvis))){ 
    viscosity = minvis*alpha;
    pViscosity_new[idx] = viscosity;

    Shear = DPrime*(2.*alpha*minvis);
} else{
    viscosity = minvis;
    pViscosity_new[idx] = viscosity;

    Shear = DPrime*((viscmax/SecInvar)+2*minvis);
}
```

### 3.4.2 Algorithmic Implementation

The equations were implemented as presented above as if the calculation was done for a single point. The rest of the MPM computation process, including mapping of the information between particles and grid nodes, scheduling and optimization is done in other classes in the Uintah-MPM software. This facilitates customization of the MPM framework to different applications.

The XML parsing libraries have been updated in order to recognize the new variables into the Uintah-MPM framework. Small changes and bug fixes were also necessary in a few occasions. However, the most important C++ method that was created is for the computation of the stress tensor: `Thixotropic::computeStressTensor` and its equivalent for the Bingham model.

The steps taken in the thixotropic stress tensor computation method are summarily described in Algorithm 1. The full code can be found in Annex 1.

Some additional steps have been implemented as a way to insure better numerical stability. Firstly, when viscosity diverges due to lack of significant shear efforts and aging of the material takes place, an additional parameter was created to let the user specify a cap over which viscosity will stop increasing. This avoids diverging to infinite values which would cause the computation to crash. This threshold should be chosen to be high enough to practically have no effect on the resulting behaviour of the fluid, for the studied time scale and stress ranges.

Another important change that has been made is that the particles are not allowed to age if their x-coordinate in the cartesian frame of reference is negative. This was necessary to simulate pipe deposition scenarios. To solve extrusion-type problems within

the Uintah-MPM framework, the user creates blocks of materials during the initialization (i.e. all particles are created at time  $t = 0$  s). More details about this process were previously presented in section 3.2.2.

In order to prevent particles from aging while they were still out of the free surface flow computational domain (they can be thought of as still in the pipe), this workaround was required. Future users should be aware that this means the effective computational domain is limited to positive x values.

<b>Algorithm 1: Stress tensor computation with thixotropic rheology</b>	
1:	For every patch in the model
2:	Get Particle Data from the previous timestep in the data warehouse
3:	Allocate memory in data warehouse for the new data
4:	For every particle in the patch
5:	If particle is at end pipe
6:	Allow aging and shearing to begin
7:	End If
8:	Compute new deformation gradient (Eq. 3.13)
9:	Compute rate of deformation tensor and its deviatoric part (Eq. 3.2)
10:	Compute updated structure parameter and viscosity (Eq. 3.7 to 3.9)
11:	If viscosity > max allowed
12:	viscosity = max allowed
13:	End If
14:	Compute shear stress (Eq. 3.1, in part)
15:	Compute current density
16:	Compute hydrostatic pressure (Eq. 3.11)
17:	Compute total stress (Eq. 3.1)
18:	Compute artificial viscosity, if specified
19:	End loop over particles
20:	Compute wave speed for a stable timestep
21:	End loop over patches

1:	For every patch in the model
2:	Get Particle Data from the previous timestep in the data warehouse
3:	Allocate memory in data warehouse for the new data
4:	For every particle in the patch
5:	If particle is at end pipe
6:	Allow aging and shearing to begin
7:	End If
8:	Compute new deformation gradient (Eq. 3.13)
9:	Compute rate of deformation tensor and its deviatoric part (Eq. 3.2)
10:	Compute updated structure parameter and viscosity (Eq. 3.7 to 3.9)
11:	If viscosity > max allowed
12:	viscosity = max allowed
13:	End If
14:	Compute shear stress (Eq. 3.1, in part)
15:	Compute current density
16:	Compute hydrostatic pressure (Eq. 3.11)
17:	Compute total stress (Eq. 3.1)
18:	Compute artificial viscosity, if specified
19:	End loop over particles
20:	Compute wave speed for a stable timestep
21:	End loop over patches

For Bingham rheology, computing the stress tensor is very similar. No workaround is necessary for pipe deposition since the material's properties will not change in time. Another difference is the determination of viscosity. Outside of these, two changes, the computation of the stress tensor is similar to Coussot's thixotropic model. The steps are summarized in Algorithm 2.

<b>Algorithm 2: Stress tensor computation with Bingham rheology</b>	
1:	For every patch in the model
2:	Get Particle data from the previous timestep in the data warehouse
3:	Allocate memory in data warehouse for the new data
4:	For every particle in the patch
5:	Compute new deformation gradient (Eq. 3.13)
6:	Compute rate of deformation tensor and its deviatoric part (Eq. 3.2)
7:	If shear rate > yield stress
8:	viscosity = viscosity
9:	Else If shear rate < yield stress
10:	viscosity = viscosity * alpha
11:	End If
12:	Compute current density
13:	Compute hydrostatic pressure (Eq. 3.11)
14:	Compute total stress (Eq. 3.1)
15:	Compute artificial viscosity, if specified
16:	End loop over particles
17:	Compute wave speed for a stable timestep
18:	End loop over patches

## 4: Validation

In order to validate the capability of the material point method to simulate fluid dynamics, some common problems tackled in the field have been studied. The problems range in complexity from simplest to hardest. Laboratory experiment on Newtonian, Bingham and thixotropic fluids have been replicated. Comparison was done with analytical equations given their availability.

### 4.1 Poiseuille flow

Poiseuille's law provides an analytical solution for the case of laminar flow between two parallel plates based on lubrication theory. Assuming an incompressible fluid and no-slip conditions at the boundaries, the velocity profile can be expressed as

$$v(z) = v_{max} \left( \frac{4z}{h} - \frac{4z^2}{h^2} \right) \quad (4.1)$$

$$v_{max} = -\frac{h^2}{8\mu} \cdot \frac{dp}{dx} \quad (4.2)$$

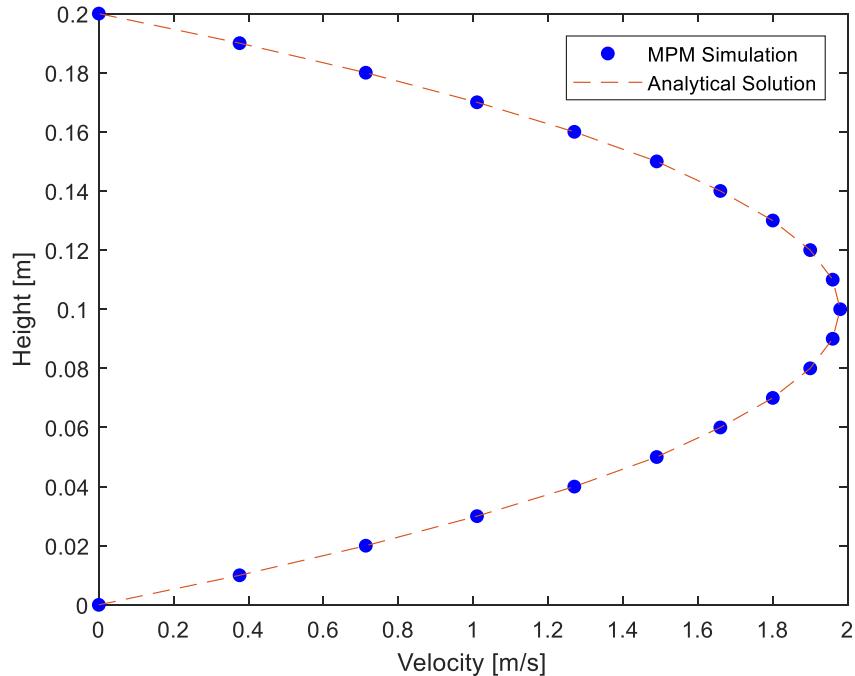
In order to simulate Poiseuille flow between two parallel plates numerically, the chosen parameters chosen are presented in table 2. A high bulk modulus was chosen in order to make the fluid nearly incompressible.

The total simulated time was two seconds, which was enough for the system to reach steady-state conditions. Viscosity and pressure gradient (gravity-driven flow) were chosen in such a way to avoid turbulence or high speed.

**Table 2: Physical and numerical parameters for Poiseuille flow simulation**

Parameter	Experiment #1
Substance	Fluid
Viscosity (Pa*s)	25
Bulk Modulus (Pa)	15 000
Density (kg/m <sup>3</sup> )	1000
Width (m)	0.4
Height (m)	0.2
Total time (s)	2
Acceleration (m/s <sup>2</sup> )	9.81

Figure 16 shows the velocity profile computed with the analytical equation and compared the velocities computed at grid nodes. A quasi-perfect fit is obtained between the two as differences are in the order of  $10^{-5}$  m/s. The velocities in the transverse direction were less than  $10^{-15}$  m/s which is negligible. This means that a bulk modulus of 15 000 Pa was sufficient to ensure incompressibility for the given conditions.



**Figure 16: Comparison of numerical simulation and Poiseuille analytical solution**

## 4.2 Dambreak experiment with Newtonian fluids

Cruchaga & al. (2007) have conducted dam break experiments on water and shampoo, both considered to be Newtonian fluids. They modeled a total of four test cases; two dam breaks experiments with water were carried out with an aspect ratio of 1H:1W and 2H:1W respectively. The same two physical configurations were also done for shampoo. The small timescale of the experiment allows to consider shampoo as a high viscosity Newtonian fluid. The physical properties of the two materials were measured using standard laboratory procedures for rheological testing.

Two out of their four experiments were chosen to be replicated with a MPM simulation in order to validate the behavior of the numerical model. For water, the case with an aspect ratio of 1:1 was used.

The other dam break case with water showed a high degree of turbulence. Treating fluid motion with very high Reynolds numbers would require specialized computational methods to be represented accurately. For example, particle-splitting algorithms could be used to improve the resolution of the velocity field with the presence of turbulence eddies when a small number of particles are present in a cell. However, it is not the main focus of this work.

For the shampoo dam break cases, an aspect ratio of 2H:1W was chosen. The high viscosity of the liquid made inertial effects barely present. Since laminar flow and lubrication theory (which assumes no inertial effects) has been validated with the study of Poiseuille flow previously presented, this case was deemed less pertinent. Secondly, traces

of shampoo were left of the plexiglass box which makes accurate determination of the fluid interface harder.

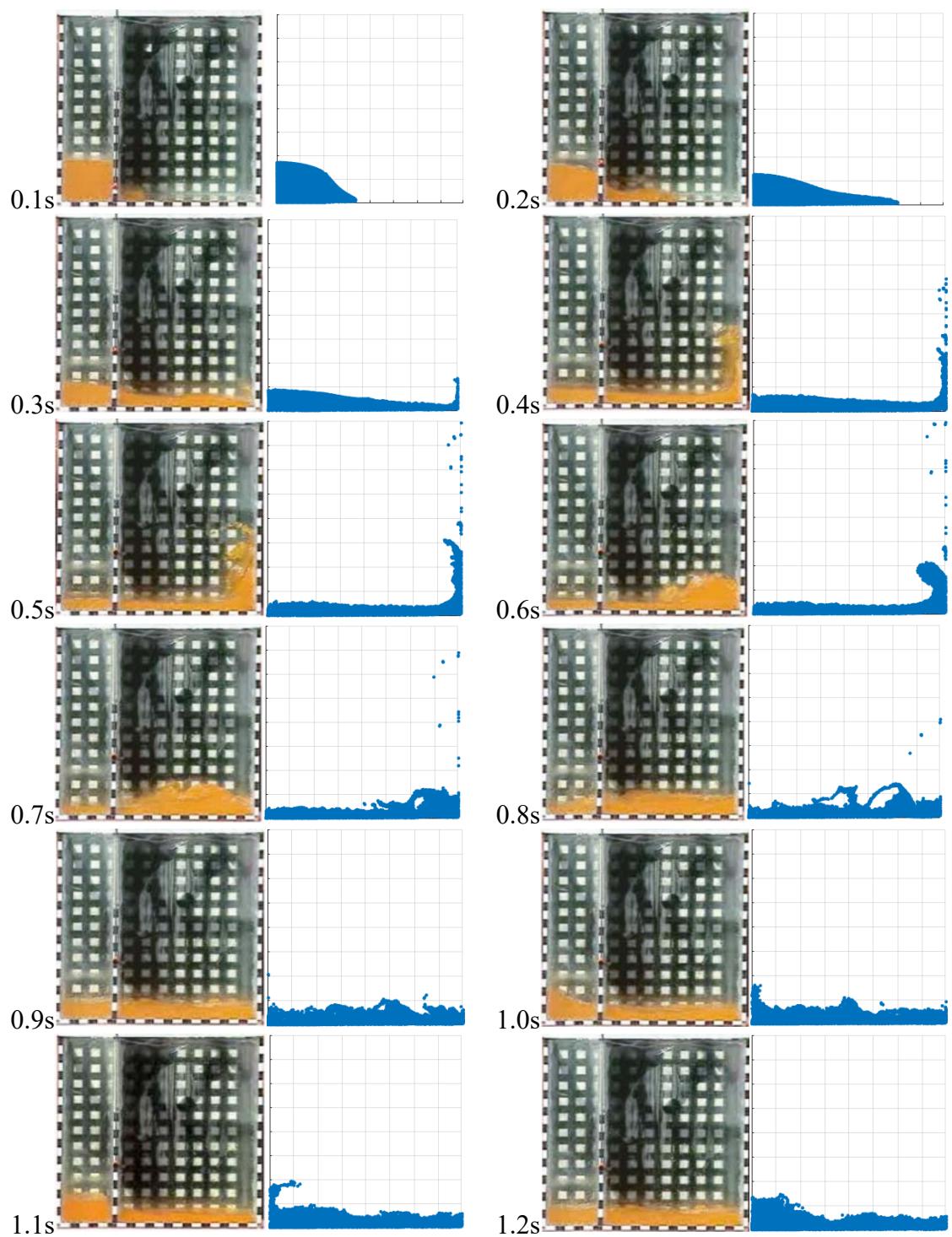
**Table 3: Parameters for Newtonian fluid dam break experiments**

Parameter	Experiment #1	Experiment #2
Substance	Water	Shampoo
Viscosity (Pa*s)	0.001	8
Density (kg/m3)	1000	1042
Width (m)	0.114	0.114
Height (m)	0.114	0.228
Aspect Ratio	1:1	2:1
Total time (s)	1.2	1.0
Box width (m)	0.42	0.42
Box height	0.44	0.44

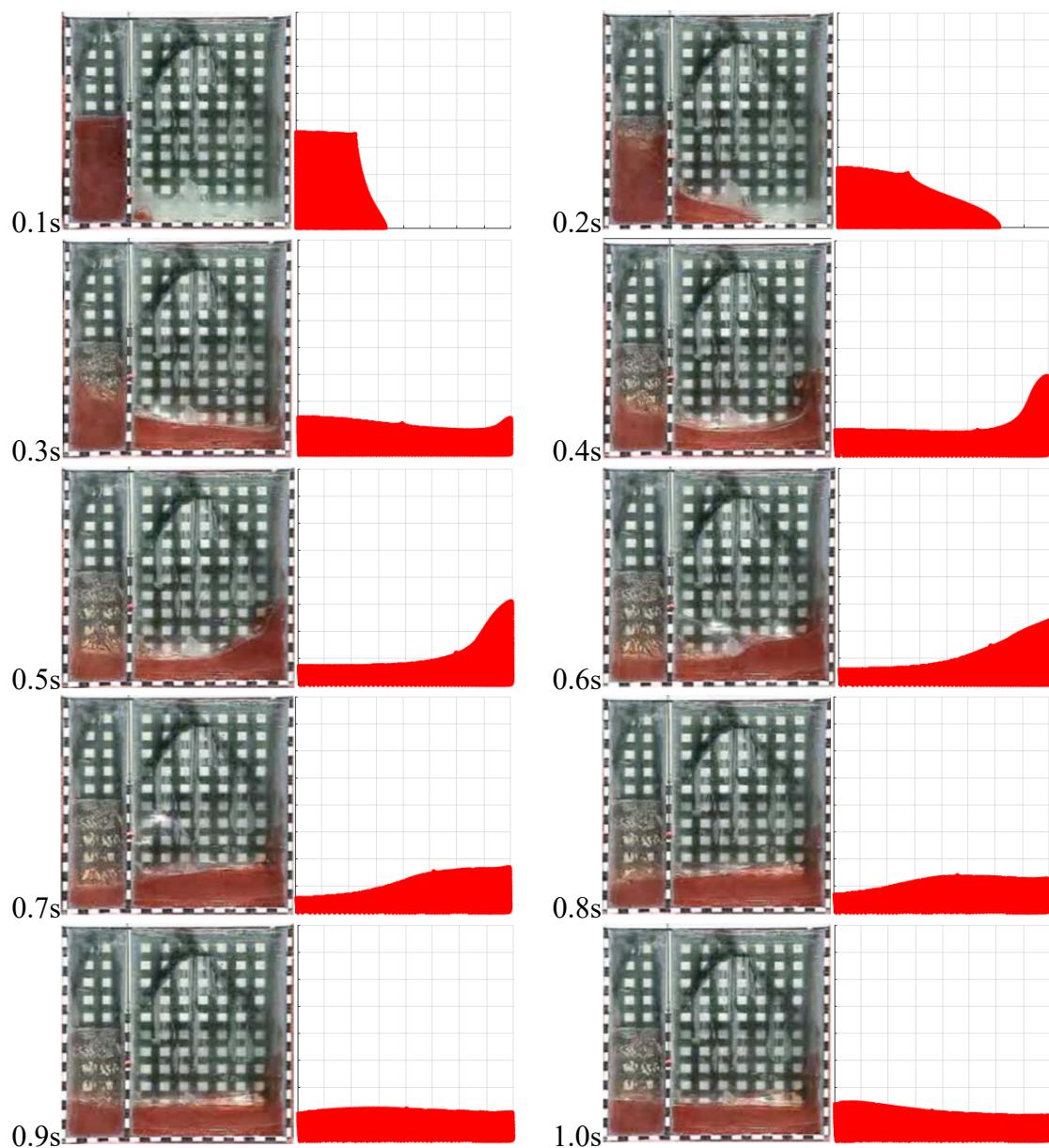
The experimental results of Cruchaga & al. (2007) with the implementation of the thixotropic constitutive model where aging was turned off to enforce constant viscosity. With the implementation of thixotropy in Uintah-MPM relying on positive x values to allow aging, this simulation was done where  $x < 0$  for the entire domain.

The pertinence of modelling Newtonian fluid is two-fold for the end goal of the thesis. Firstly, it relies on the thixotropic constitutive model, so it can provide a partial validation of its physics. Secondly, the high inertia and turbulence resulting from the dam break and a low viscosity fluid provides insight towards the behavior of the system under high energy conditions to tackle large-scale tailings dam break problems.

Screenshots are compared at an interval of 0.1 seconds until the fluids approach hydrostatic equilibrium in figure 17 and 18.



**Figure 17: Water dam break, validation with experimental results**



**Figure 18: Shampoo dam break, validation with experimental results**

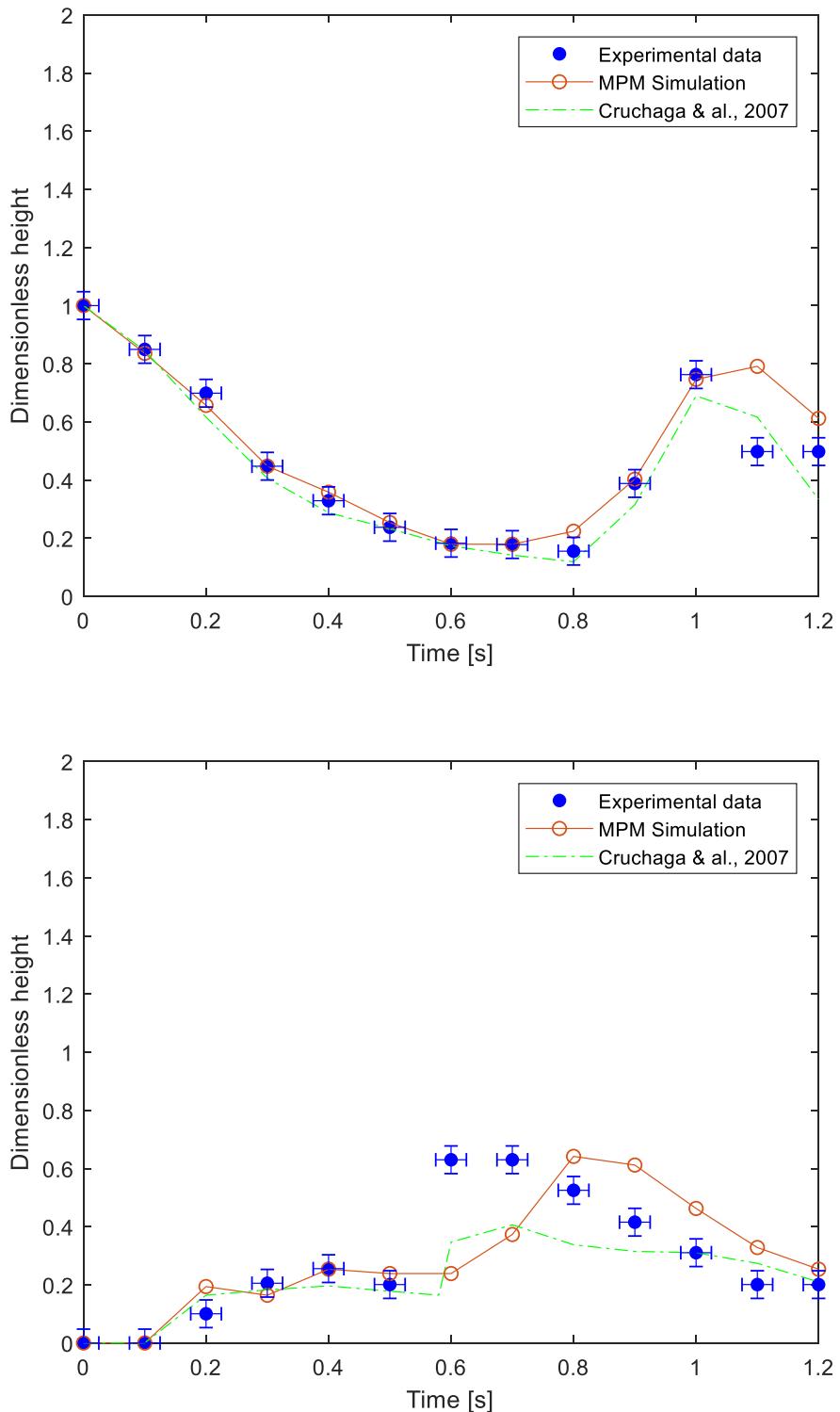
The MPM model was able to reproduce with a reasonably high degree of accuracy the flow behavior of both the low viscosity and high viscosity fluids for the majority of the time steps when compared on the basis of frame-by-frame observation. The turbulent behavior has been captured, at least in part.

For both test cases, every 0.1 seconds, the height of the fluid was measured at three points: the left wall; in the middle; and at the right wall. A three-way comparison is made with experimental data, the Uintah-MPM code, and the numerical model used by the authors. The author's numerical methods rely on a FEM method with edge-tracked interface locator technique (ETILT) to follow the evolution of free surface during the dam collapse.

Figures 19 and 20 shows the evolution in the material's height in time for both tested cases. The data is normalized with the initial length of the sample (0.105 m).

Some differences between the experimental setup and the numerical simulation can affect the accuracy. First on all, the gate was lifted at 0.83 m/s, but the collapse is considered instantaneous numerically. The effect of friction has not been considered. No-slip boundary conditions were used to describe the domain.

For the high viscosity fluid, the effect of the boundary condition was more noticeable. The use of a no-slip condition at the bottom produced excessive damping of velocity and results did not accurately compare with the laboratory experiments. A second simulation was done using a no-penetration boundary. For this case, particles at the boundary are met with an opposing force of equal magnitude so the resultant is null. Using this type of boundary condition produced numerical results that closely matched the experimental results.



**Figure 19: Comparison between experimental results, author's results and MPM model for a water dam break. Dimensionless height ( $y/L$ ) at left wall, middle (0.27m) and right wall**

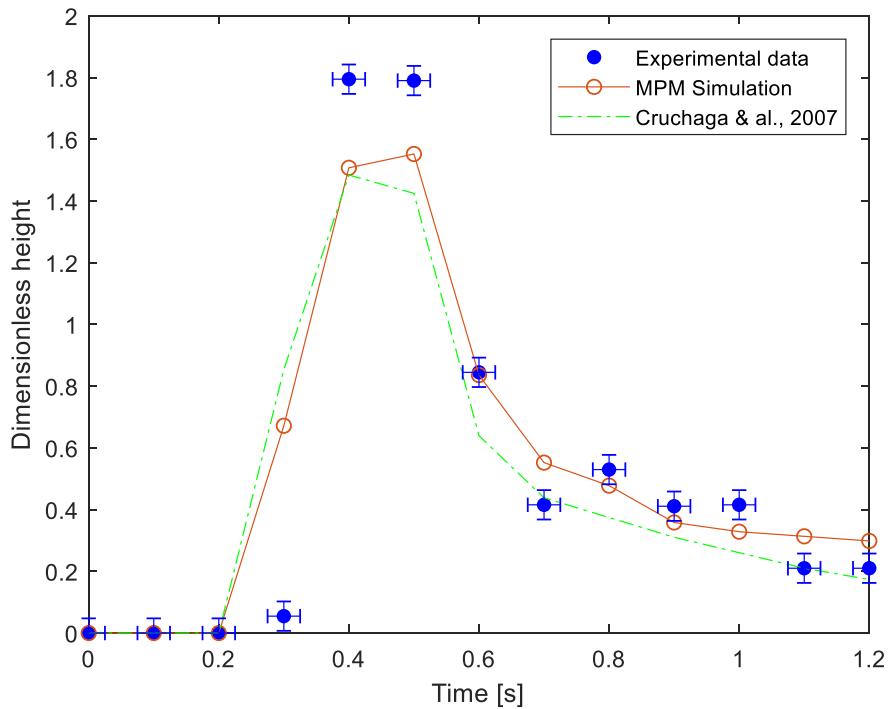


Figure 19 (cont'd)

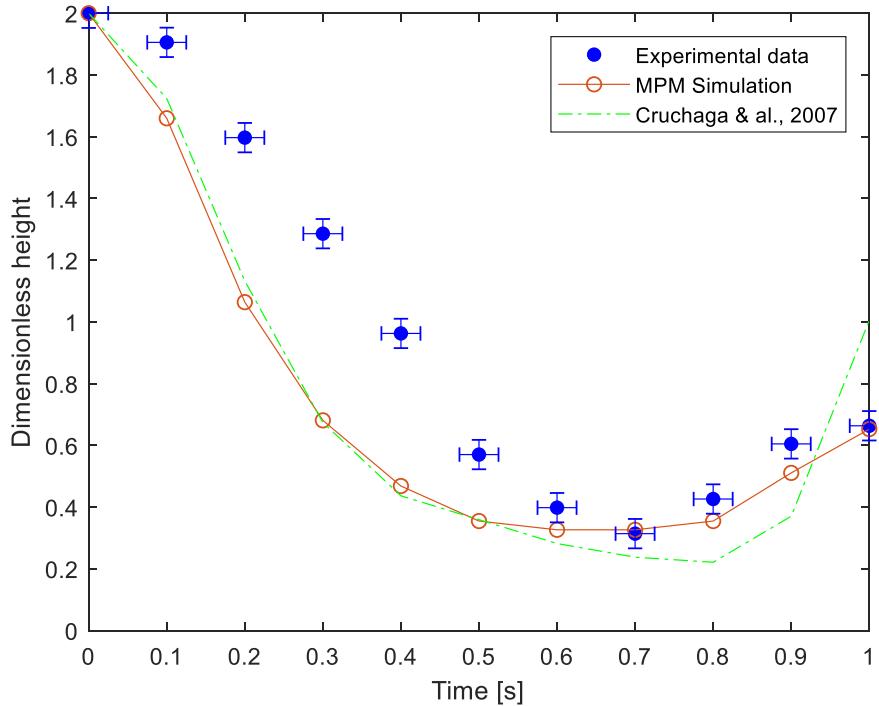
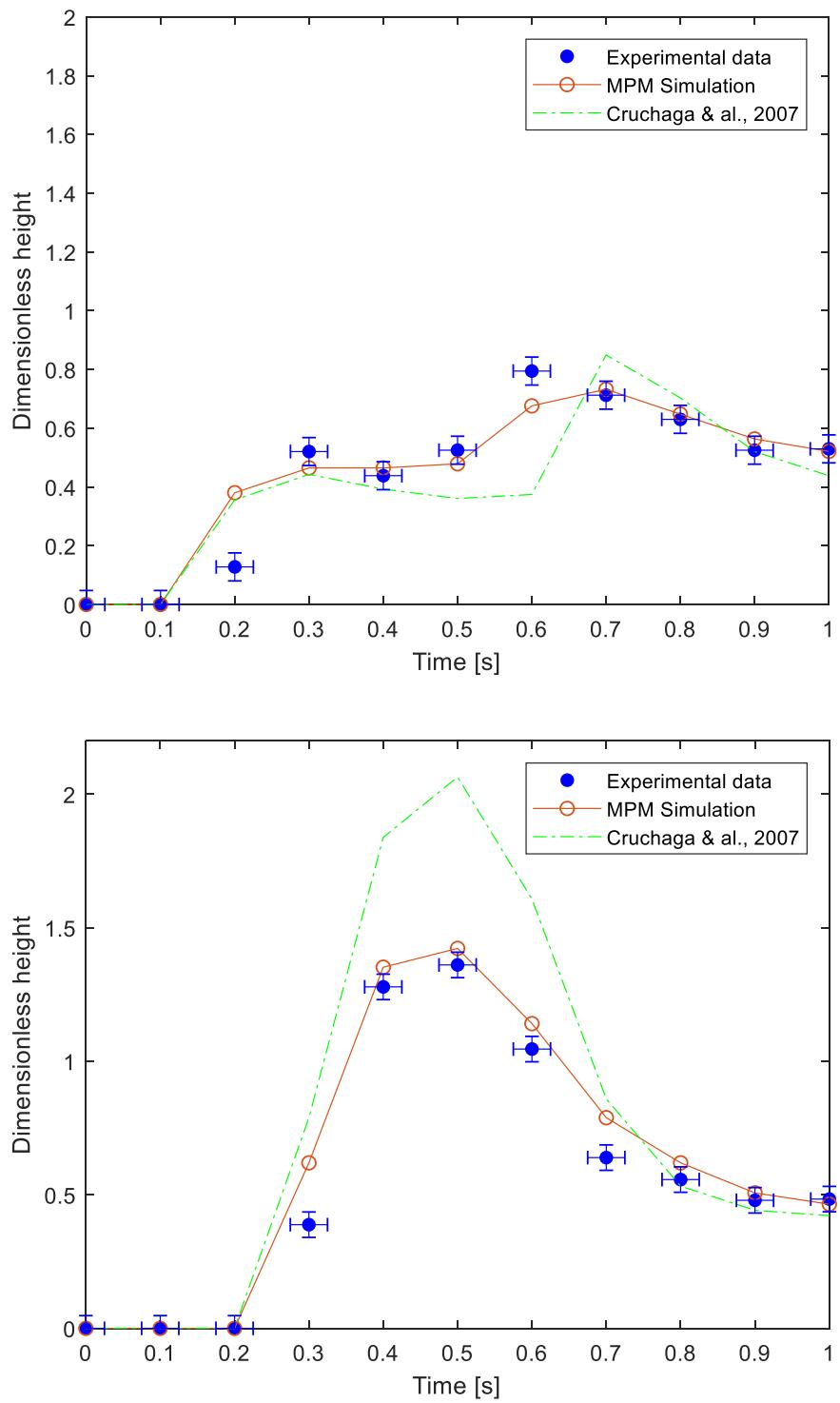


Figure 20: Comparison between experimental results, author's results and MPM model for a shampoo dam break. Dimensionless height ( $y/L$ ) at the left wall, middle (0.27m), and right wall.



**Figure 20 (cont'd)**

### 4.3 Bingham rheology

A set of simulations were done to replicate previous laboratory data on free surface flow of mine tailings. The experimental work conducted by Henriquez & Simms (2009) was done for both dam break and inertia-free pipe deposition scenarios. A Bingham rheological model was used to predict the final runout of the tailings.

The material tested are tailings from a gold mine provided by Barrick Gold. Their granulometry is standard to what would be expected for tailings, with an average grain size ( $d_{50}$ ) of 35  $\mu\text{m}$ .

In an attempt to find the most appropriate values for the Bingham parameters, several standardized rheological tests were done. A variability was expressed in shear rates, which were changed from 0 to 400  $\text{s}^{-1}$ . Higher shear rates are more common for the design of pipe transportation systems whereas lower shear rates are found during free surface flow in the impoundment. The expressed yield stress and viscosity changed considerably as expressed in table 4.

**Table 4: Bingham model for gold tailings measured under different conditions**

Methods	Viscosity (Pa*s)	Yield stress (Pa)
Rheometer 1 and slump tests	0.12	73
Rheometer 2	0.19	52
Rheometer, lab 2	0.65	36

The authors determined that the values of a 0.65 Pa\*s viscosity and a yield stress of 36 Pa were more representative of the low shear rates expected in bench scale experiments. Therefore, the same set of parameters will be chosen for validation of the Uintah-MPM Bingham rheological model. Figure 21 shows the laboratory setup that was used for the bench scale deposition scenarios.



**Figure 21: Flume setup for deposition experiments (Adapted from Henriquez, 2009)**

The laboratory experiments were simulated numerically with the rheological parameters that were determined by the authors. Comparison was also made with the previous modelling efforts of other researchers.

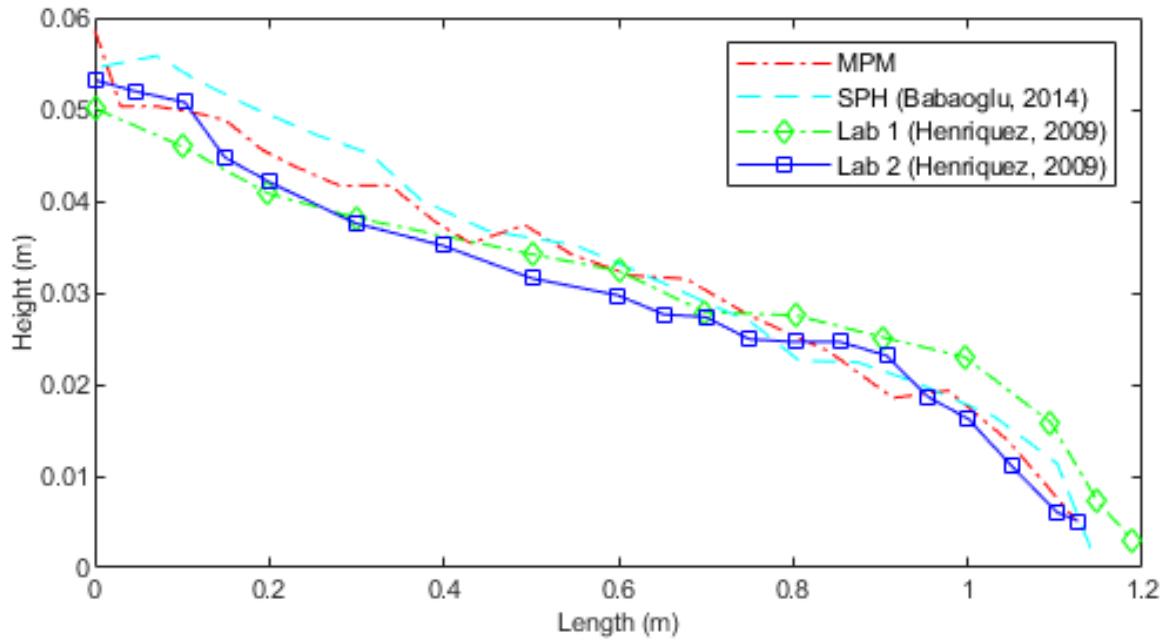
The deposition laboratory experiments were modelled successfully using smooth particle hydrodynamic (SPH) methodology (Babaoglu, 2014). The location of the free surface of this model will also be compared with MPM and physical results to further investigate the performance of MPM by comparing it to other common modeling tools.

#### **4.3.1 Dam break scenario**

Dam break scenarios were realized in the flume setup. The tests were performed at a gravimetric water content of 40%. The width of the flume is 15.2 cm. The width was chosen in order to negate the effects of side-wall friction.

For the dam break scenarios, the tailings were mixed and then placed behind a gate. The material reached total heights of 10 cm or 20 cm. An initial cross-section of 20 cm by 20 cm ( $400 \text{ cm}^2$ ) was considered. There were differences in the runout distance when they were compared to lubrication theory because of non-negligible inertial effects.

Figure 22 presents the difference in runout between dam breaks for both the author's experimental results, the MPM simulation and a SPH model.



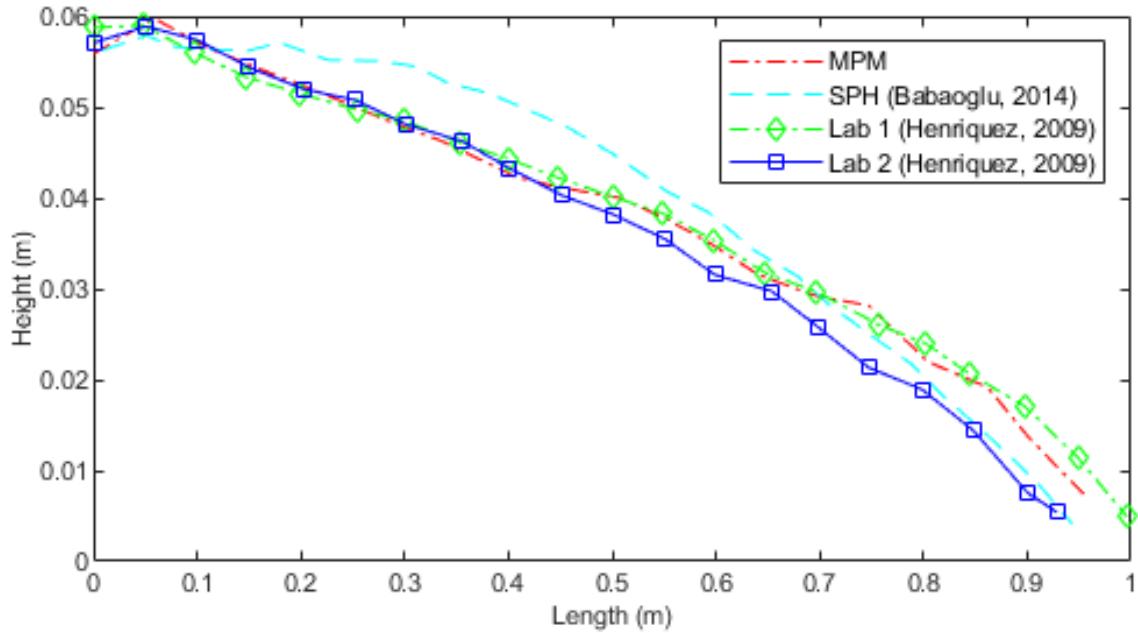
**Figure 22: Bench scale dam break runout distance, experimental data and predictions**

Good agreement is found between the runout that was measured experimentally and computed results with both MPM and SPH tools. With appropriate Bingham parameters as input, the runout of small scale experiments can be predicted. However, it seems both numerical models overestimated the first half of the runout profile compared to the laboratory experiments (for  $0 < x < 0.5 \text{ m}$ ).

#### 4.3.2 Pipe deposition scenarios

Inertia was found to have an effect on the runout distance. Dam breaks would reach further distances than material deposited with a funnel. Development of the analytical solutions in lubrication theory relies on removal of the inertia term to simplify the differential equations. The previously cited authors (Henriquez & Simms, 2009) also ran deposition experiments using a funnel to reduce the inertia until negligible.

The same volume of material was deposited with a funnel. The resulting profile and its comparison with SPH and MPM numerical models is presented in figure 23.



**Figure 23: Bench scale pipe deposition runout distance, experimental data and predictions**

In the previous figure, differences between experimental data and the MPM implementation of Bingham rheology are minimal. The SPH method seems to overestimate the first part of the runout profile. This was a characteristic detected in the simulations on dam break data and remains of little influence on the quality of the predictions.

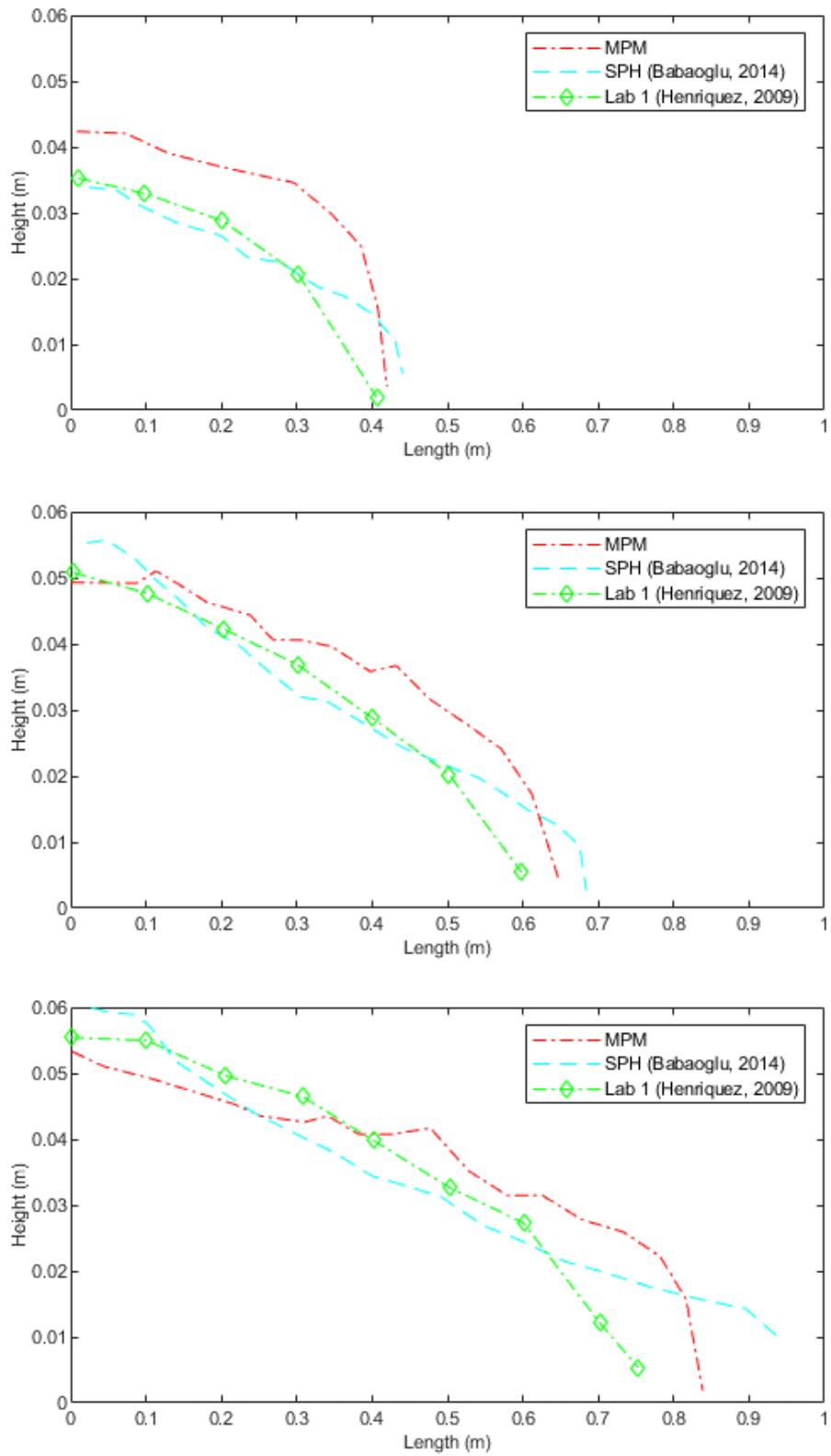
It can be hypothesised that some numerical settings could be optimized to calibrate the model to the experimental data.

More comparisons between laboratory and numerical analysis were done for pipe deposition under transient conditions. Figure 24 shows the runout distance after 3, 6 and 9 seconds. It should be mentioned that the final profile, shown in the previous figure, occurred after 11 seconds of flow.

SPH was able to reproduce the evolution up to the final profile with more accuracy than MPM until a certain point in time. After which, the MPM method was closer to the experimental results, as depicted in the final profile.

At the 3 and 6 seconds mark, the total volume of material present with the MPM simulation is larger than the volume of SPH and experimental results. The difference can be explained with a different in input flow rates. In the laboratory and SPH model, the material was deposited with a funnel. The material's velocity will therefore not be constant. The MPM simulation assumes a pump-like constant flow rate. The later phases of the flow are more accurately captured since the relative differences in deposited volumes are smaller.

The variability in the transient flow profile remains satisfactory for engineering purposes.



**Figure 24:** Transient runout of mine tailings after 3, 6 and 9 seconds

## **4.4 Thixotropic rheology**

### **4.4.1 Experimental validation**

Finally, Coussot's thixotropic constitutive model was validated with experimental data. The rheological dataset used was compiled by (Mizani, 2016b) where the behavior of oil sand tailings was fitted using Coussot's thixotropic model.

The deposited material was tailings resulting from the oil sand industry that exhibit a slightly different behavior than hard rock mine tailings that were used to validate the Bingham model . The tailings are a mix of sand, water, fine clay particles, residual bitumen and chemical additives that were added in the extraction process. Flocculents are also added as the optimal dosage was under investigation by the authors. Flocculent doses between 650 and 1100 g/ton were added.

One deposition scenario was chosen to be reproduced by numerical means. For that case, the deposition rate with the pump setup was 36 L/min for a total duration of 21 minutes and 17 seconds. The total volume deposited is 773 L and the width of the flume is 0.59 m. Figure 25 shows the experimental set-up, including the flume, tanks and pump.

In-situ testing showed a material with a yield stress in the range of 50 Pa if fitted with a Bingham rheological model for shear rates in the range of deposition values and immediately after mixing.



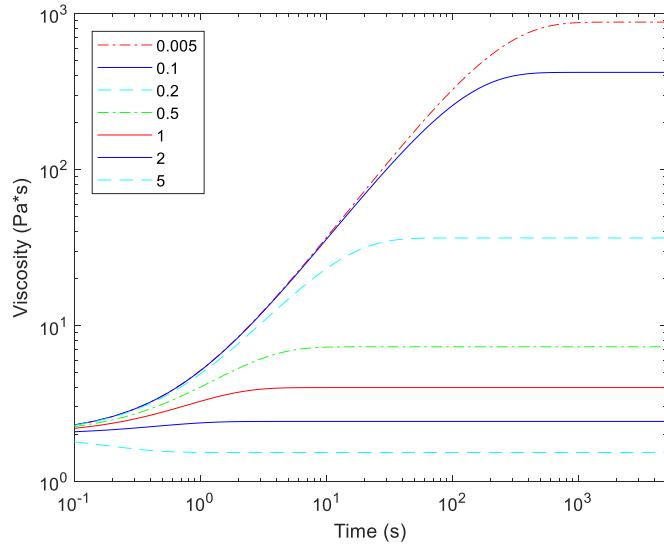
**Figure 25: Experimental set-up for thixotropic free surface flow modeling (Adapted from Mizani, 2016b)**

The oil sand tailings also contained polymer additives to facilitate flocculation. The parameters provided good qualitative fit with the data although there were some differences that required more complex thixotropic models to fit. The Coussot model was however largely satisfactory in its capacity to explain the time dependant nature of the tailings. The parameters are presented in table 5.

**Table 5: Coussot model parameters for tailings**

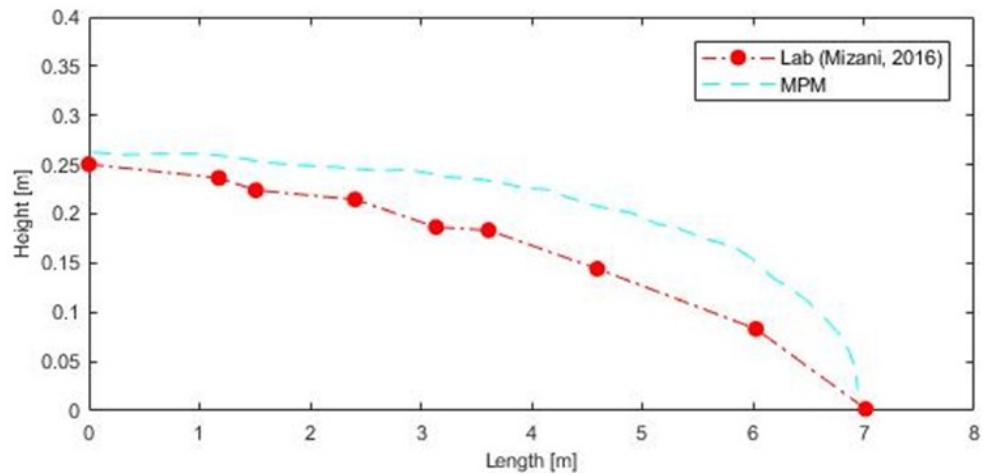
Parameter	Shabnam (2016b)
$T^*\alpha$	0.36
$\mu_0$ (Pa s)	1
n	1.074

Figure 26 presents the evolution of viscosity as a function of time for several constant shear rates for the given parameters in one dimension. Values vary as a function of aging primarily between 1 and 100 seconds before stabilizing at a constant value. Also, it can be noted that, at high shear rates, the minimum viscosity value is approached.



**Figure 26 : Time-dependant viscosity profile with Mizani (2016b) parameters**

The results were also reproduced with the thixotropic constitutive model with MPM. Results for the 2D simulation are presented in figure 27. It should be noted that the current implementation is explicit and time steps are limited by a stability criterion. Therefore, it was only possible to simulate 2D deposition to stay within the limits of reasonable computation times with the hardware that was used. Figure 28 presents the evolution of the viscosity at different time steps.

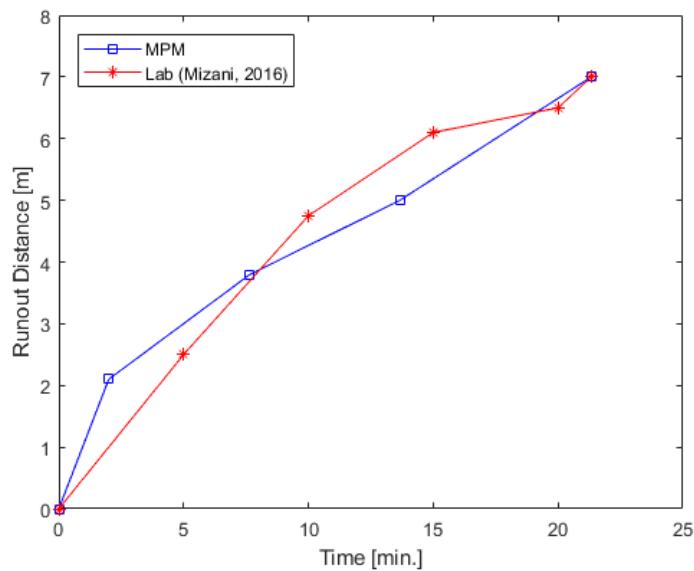


**Figure 27: Predicted and experimental beaching profiles of thixotropic oil sand tailings**

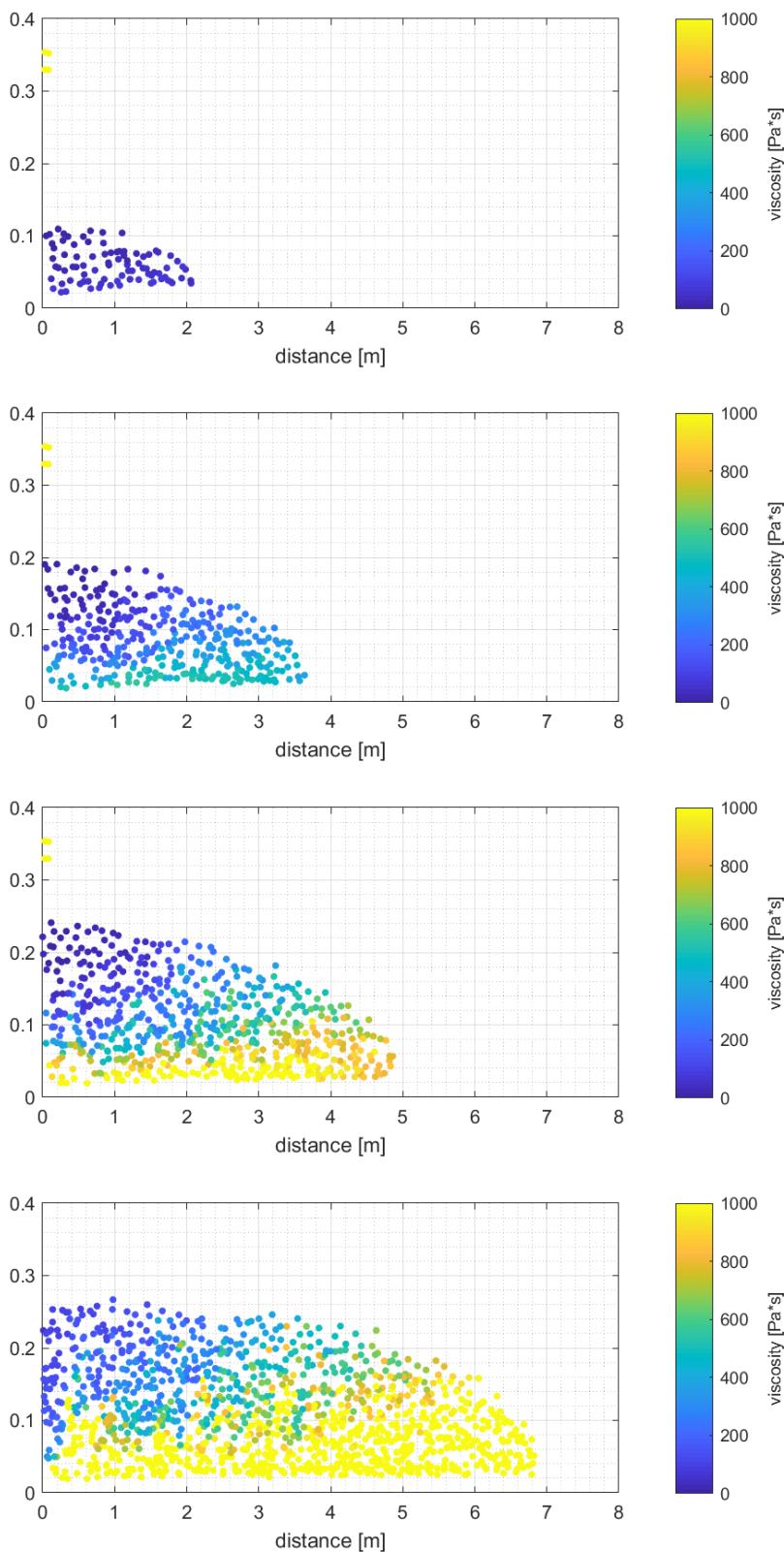
The final profile indicates good agreement between the numerical predictions and the laboratory results in terms of final height and runout distance. However, the total volume is significantly larger in the numerical simulation. The height of the runout varies from 10% to 60% over the middle portion of the final runout profile.

The difference can be explained because of secondary processes that occur during flows with significant length of time that are unaccounted for in Coussot's rheological model. A total of 14.74 liters of bleed water was collected during the experimental process. Self-weight consolidation is also expected to have an influence in reducing the total volume over the 20 minutes in which the flow process takes place.

The following figure shows the evolution of the beach at different points in time. The viscosity increases slowly in time. Inversely, the speed at which the runout advances slows down in time as the material gets harder. In the first 7 minutes, the beach advanced for approximately 3.8 meters. In the last 7 minutes of flow, the beach only progressed for about 1.8 meters.



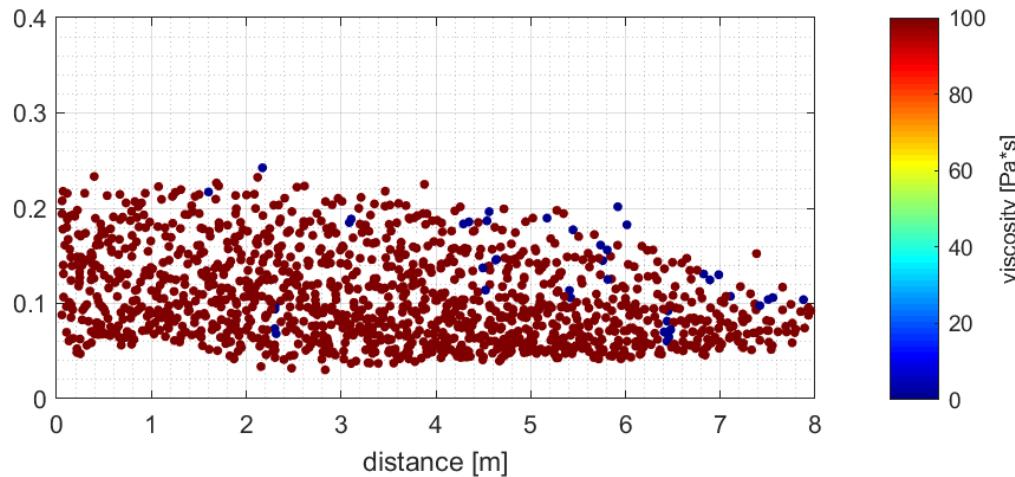
**Figure 28A: Runout of flow profile at different time steps**



**Figure 29B: 2D Flow profile at different time steps: 2 min 0 s, 7 min. 40 s, 13 min. 40 s, 21 min. 20 s**

The runout was also modeled with a Bingham constitutive model to highlight the differences between the two constitutive models. Figure 29 shows the runout using the parameters suggested by the author for an appropriate Bingham fit in the given shear rate range. The yield stress is 50 Pa and the viscosity is 0.5 Pa\*s. These parameters could be calibrated to have a better fit and some gains in accuracy could potentially be made.

The fit is satisfactory for engineering purposes. However, Mizani (2017) shows that this is not true for slower pours as the thixotropic effects become dominant, which changes the beach slope angle and geometry. Due to numerical limitations of the model, the slow pour, where thixotropic effects become significant, could not be reproduced. However, future work should focus on improving computational efficiency to model timescales of the order of hours or days. Recommendations towards this goal are presented in chapter 7.



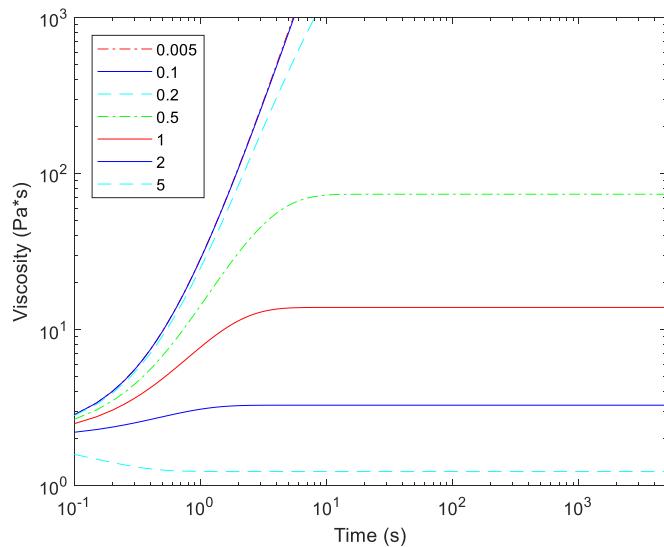
**Figure 30: Final runout modeled using a Bingham rheological model**

#### 4.4.2 Investigation of aging speed

The authors also proposed investigating a value for the n parameter between 2 and 3 as it was previously reported in literature that these parameters can accentuate the bifurcation point due to the exponential nature of the parameter and therefore simulate a flow that can be qualitatively described as a “pseudo-plug” flow by analogy to the Bingham rheological model and field observations of free surface flows. The following figure shows the evolution of viscosity using the same previous set of data with one exception; the n parameter is set to 2.5. This change was arbitrarily done as previous reporting (Mizani, 2016b) indicated ‘n’ as an important parameter in Bingham-like behavior of thixotropic fluids. An overview of the parameters is shown in table 6.

**Table 6: Coussot model parameters for tailings**

Parameter	Shabnam (2016b)	Fast-aging
$T^*\alpha$	0.36	0.36
$\mu_0$ (Pa s)	1	1
n	1.074	1.75



**Figure 31: Time-dependant viscosity profiles with increased aging parameter**

Two important differences can be noted from looking at the updated functions and how they contrast from the 1D constant shear viscosity profiles presented in the previous section. First of all, there is a slight effect on the maximum viscosity reached at every constant shear rate. For example, the viscosity once steady-state is reached under a  $1 \text{ s}^{-1}$  shear rate is  $10 \text{ Pa*s}$  instead of  $3 \text{ Pa*s}$ . However, the most notable effect is the speed at which structuration occurs. Steady states viscosity values are reached after  $10^1$  instead of  $10^2$  seconds.

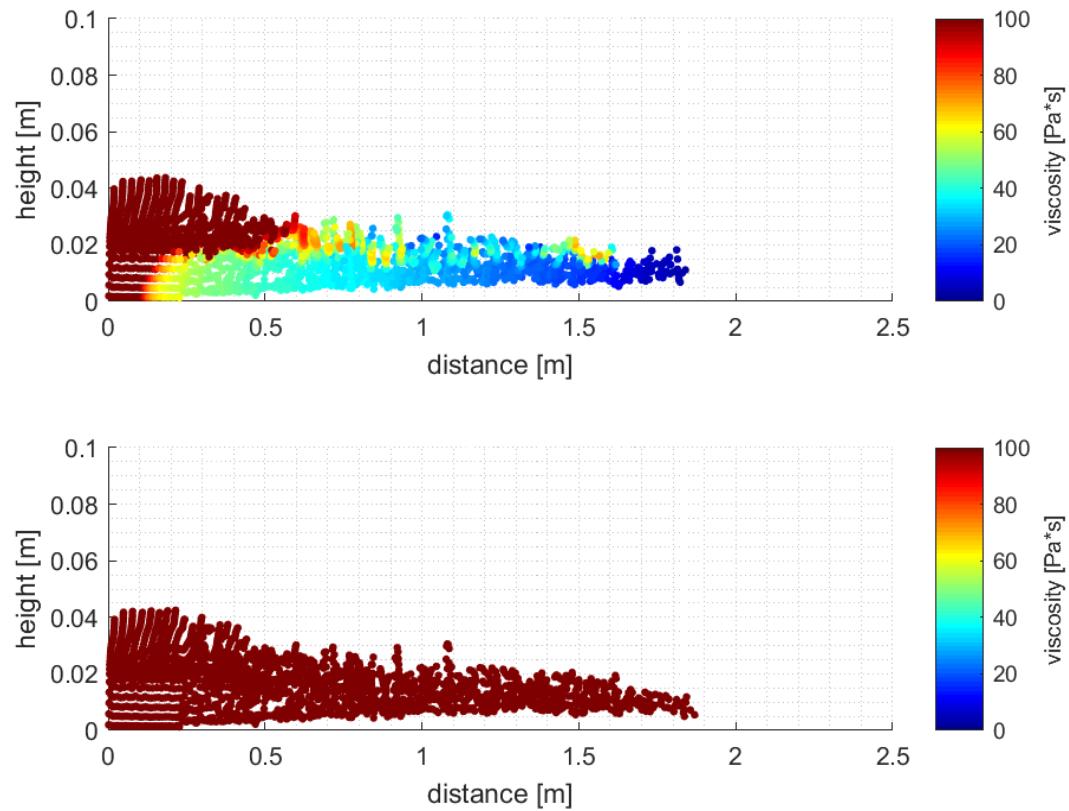
Since explicit Lagrangian numerical methods are limited by Courant-Friedrichs-Lowy (CFL) condition, the very small time step increments can make modeling large periods of time very expensive. Therefore, keeping the effect of the alpha parameter in mind is a good idea in reducing the required simulation time, however, not without losing a bit of representativity on the physical parameter side.

It is worthy of mention that viscosities over  $1000 \text{ Pa * s}$  are not considered in this analysis. The first reason is that differences in viscosity at the bench scale will not be affected significantly by the order of magnitude of viscosity at very large values.

Secondly, in the numerical implementation, there is a cap on viscosity values to prevent them to diverging towards infinity hence causing numerical errors. Thirdly, a tailing with very high viscosity value would probably be more accurately described by geotechnical elasto-plastic constitutive models. The “fast-aging” model will be used for calibration analysis, numerical trials and could also be used to accelerate the timescales for largescale 3D simulations.

The effect of other parameters will be investigated in section 4.6 where extensive validation of physical parameters has been carried out.

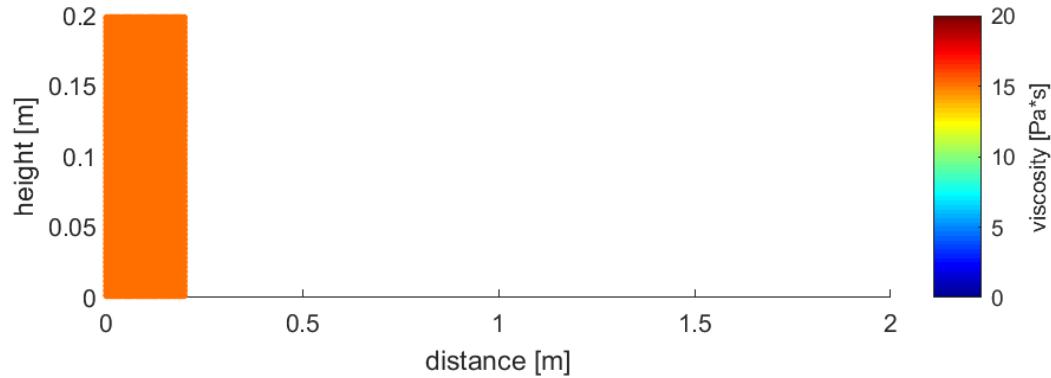
Figure 31 presents runout of the near-final resting profiles for both aging parameter “n” analyzed. The runout simulated with the fast-aging model was similar to the one with values backfitted from laboratory testing. The difference in measured runout at stoppage is therefore minimal at the bench scale. Section 5.1 presents a more in-depth look at the effect of aging at larger scales.



**Figure 32: Final flow profile for different aging speeds**

## 4.5 Parametric analysis

In this section, an analysis of the numerical and physical parameters is made to determine optimal running conditions for thixotropic free surface flow using the material point method. The test case studied is a two-dimensional laboratory scale dam break problem. A rectangular section of material with an area of 0.2 meter by 0.2 meter is released in a 2-meter-long enclosure. For the numerical portion of the analysis, all the parameters in Coussot's thixotropic model are set to values that would produce a runout of approximately one meter in length. A schematic representation of this test case is presented in figure 32.



**Figure 33: Initial setup for parametric analysis**

### 4.5.1 Sensitivity analysis to numerical parameters

To validate the thixotropic model, a sensitivity analysis to numerical parameters has been done. Numerical evaluations of partial differential equations in a non-trivial operation and, by definition, is an estimation. Therefore, it is possible that numerical parameters create undesired variability in the results.

Numerical parameters can also change the total simulation runtime by orders of magnitude with little to no gain in the accuracy of the results. Therefore, understanding the effect of these parameters is crucial in running accurate analyses in an optimal manner.

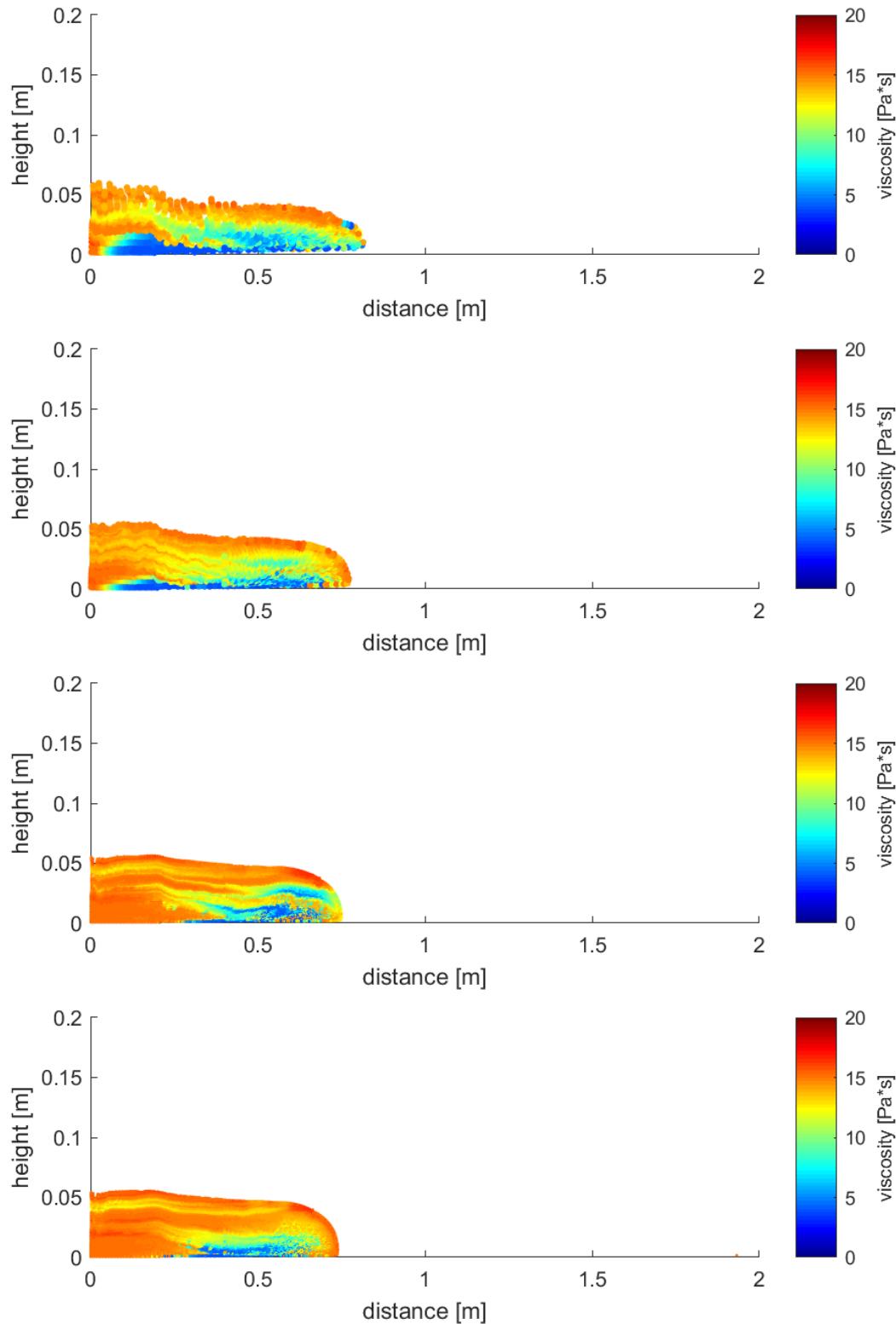
The effect of grid resolution and number of particles per cell has been analyzed. The effect of interpolation method and time steps length has also been investigated.

The timesteps vary throughout the simulation based on the stability criteria. Therefore, it is not possible to set it directly. The chosen timestep refers to a safety factor (a multiplier) associated with the stability criteria, which is dictated by the speed of sound through the material.

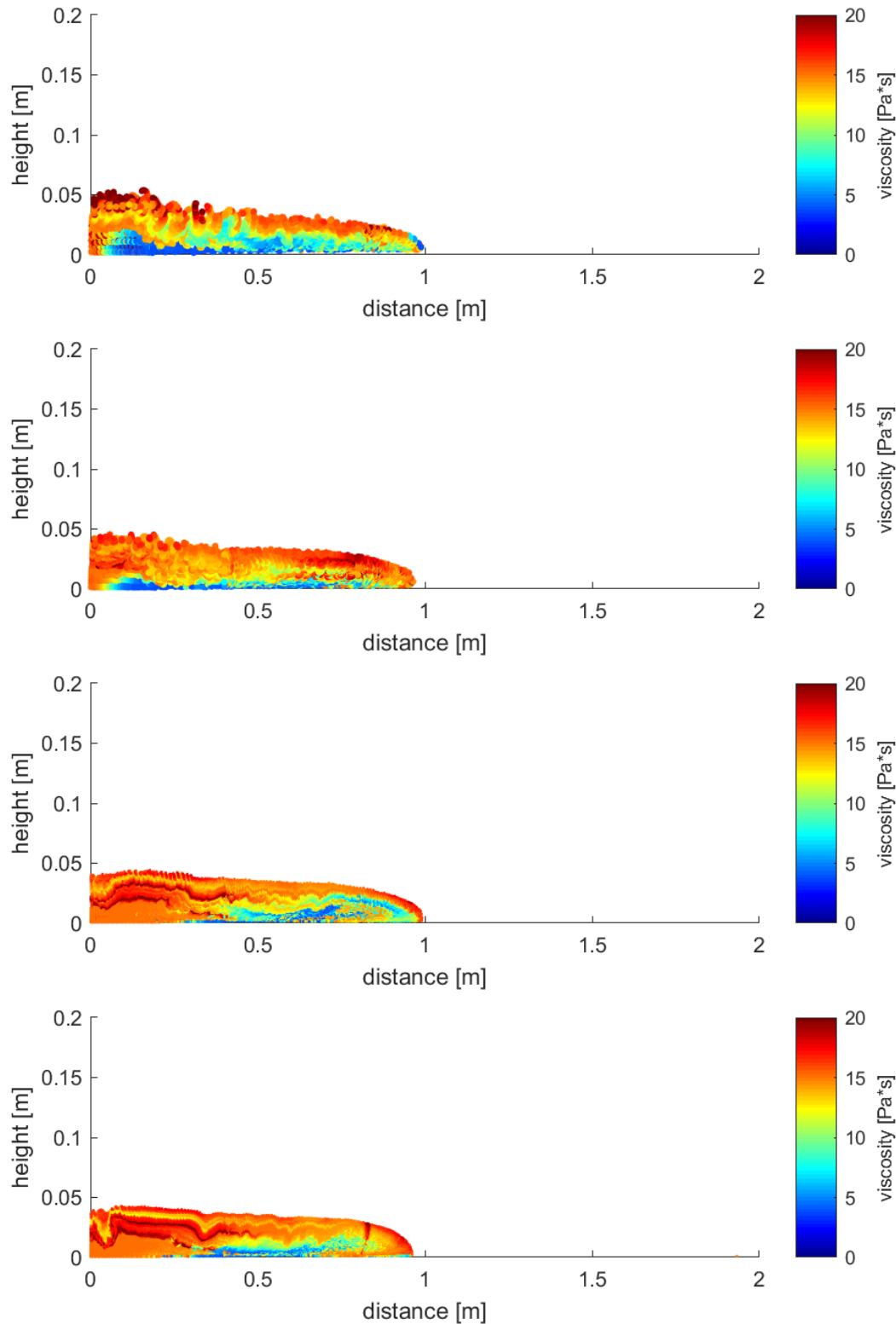
Table 7 provides an overview of the different parameters that have been analyzed. Then, in figures 33 to 40, resulting runout is shown after 1s, 5s, and 15s to show evolution of viscosity until near stoppage.

**Table 7: Overview of parameters varied in the numerical sensitivity analysis**

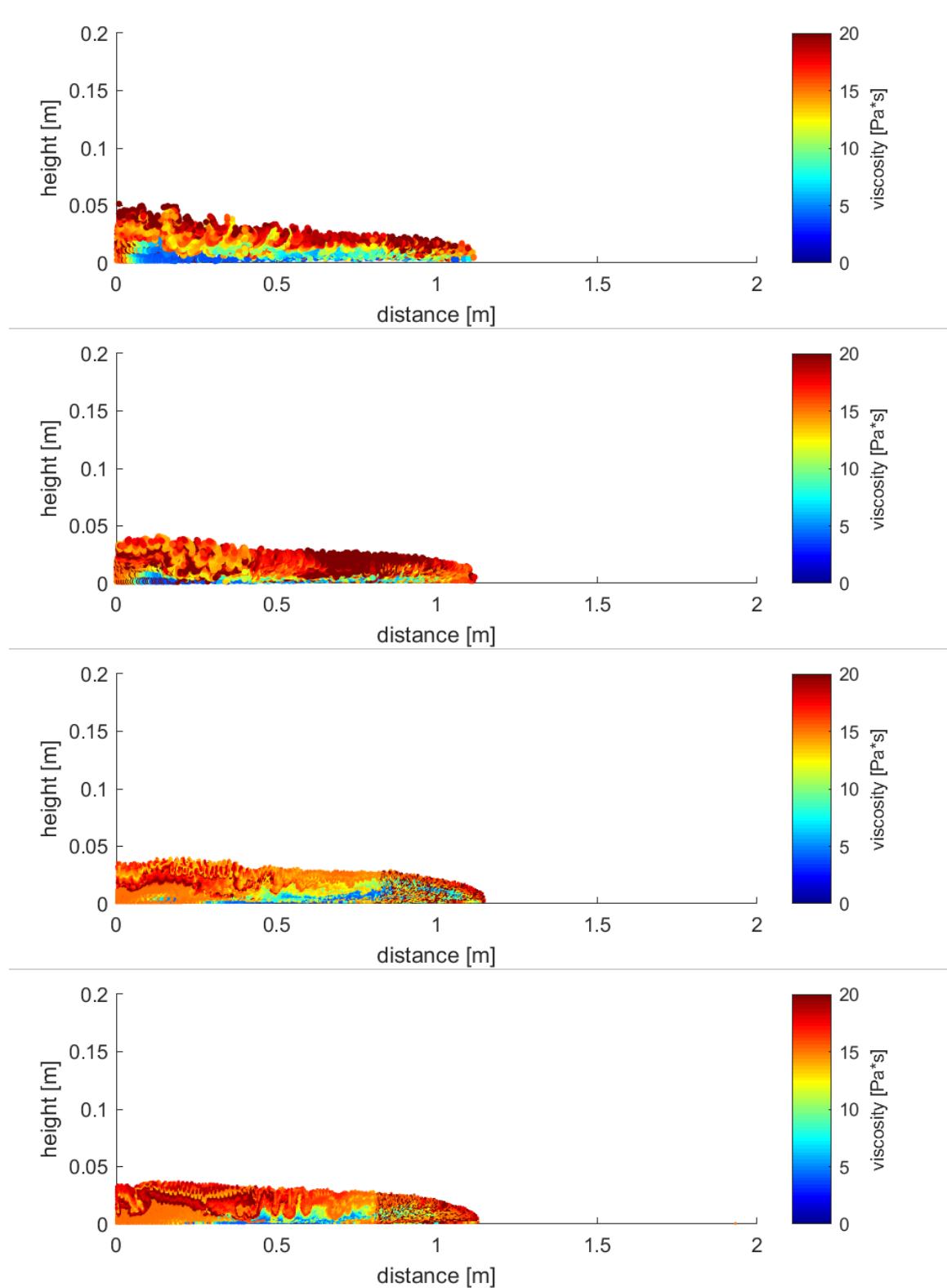
Run Number	Grid cell size (m)			Points per cell			Total Elements	Total Points	Interpolation method	Timestep (multiplier)
	X	y	z	X	y	z				
1a	<b>0.01</b>	<b>0.1</b>	0.1	2	2	2	4 800	3 200	GIMP	0.6
1b	<b>0.05</b>	<b>0.05</b>	0.1	2	2	2	10 800	7 200	GIMP	0.6
1c	<b>0.0033</b>	<b>0.0033</b>	0.1	2	2	2	43 200	28 800	GIMP	0.6
1d	<b>0.0025</b>	<b>0.0025</b>	0.1	2	2	2	76 800	51 200	GIMP	0.6
2a	0.01	0.1	0.1	<b>2</b>	<b>2</b>	<b>2</b>	4 800	3 200	GIMP	0.6
2b	0.01	0.1	0.1	<b>3</b>	<b>3</b>	<b>3</b>	4 800	10 800	GIMP	0.6
2c	0.01	0.1	0.1	<b>4</b>	<b>4</b>	<b>4</b>	4 800	25 600	GIMP	0.6
2d	<b>0.05</b>	<b>0.05</b>	0.01	<b>3</b>	<b>3</b>	<b>3</b>	4 800	24 300	GIMP	0.6
3a	0.01	0.1	0.1	2	2	2	4 800	3 200	GIMP	<b>0.2</b>
3b	0.01	0.1	0.1	2	2	2	4 800	3 200	GIMP	<b>0.9</b>
4a	0.01	0.1	0.1	2	2	2	4800	3 200	<b>CPDI</b>	0.6
4b	0.01	0.1	0.1	2	2	2	4800	3 200	<b>3<sup>rd</sup> order B-Spline</b>	0.6



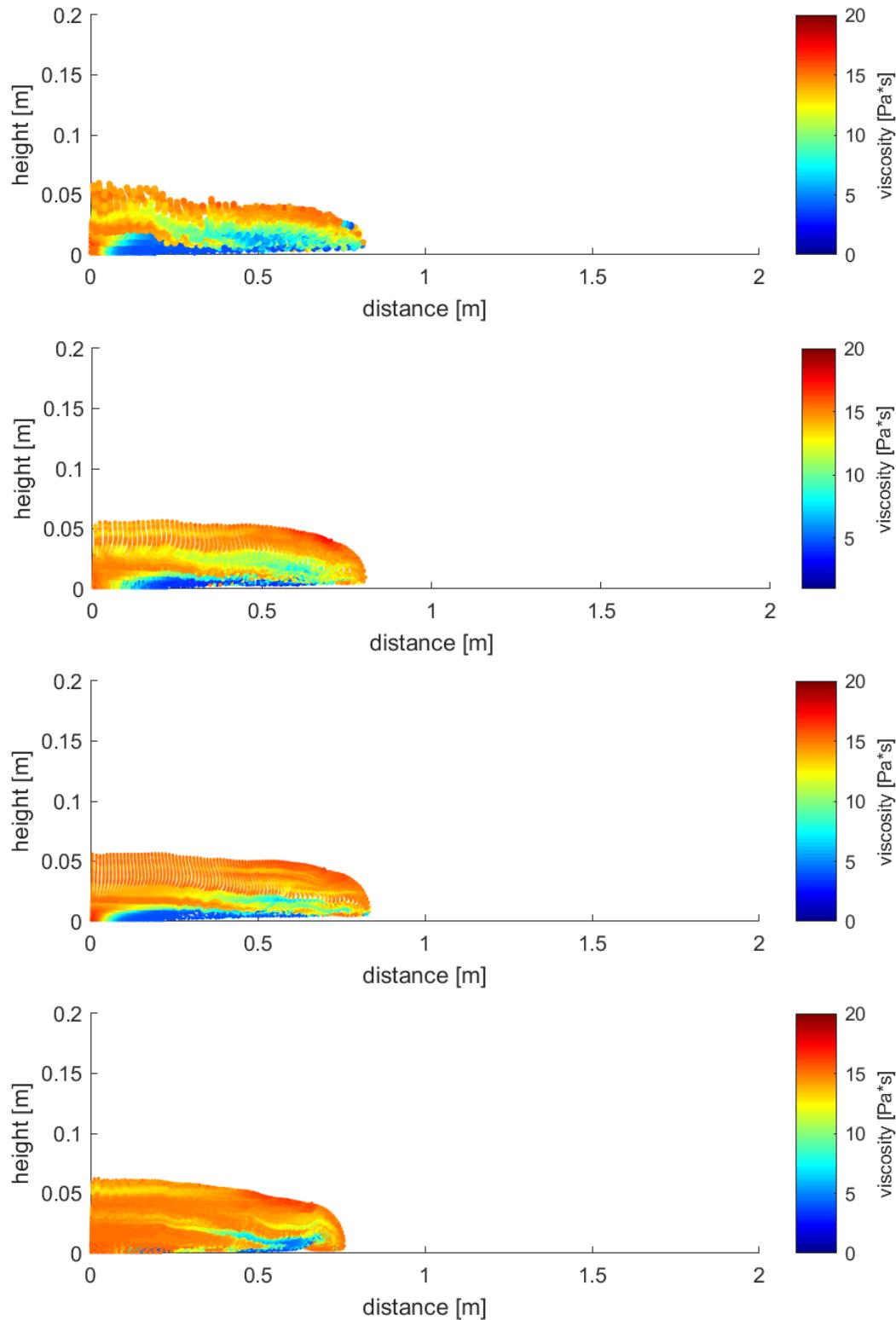
**Figure 34: Viscosity and runout as a function of time. Grid resolution sensitivity at time  $t = 1\text{ s}$ . From top to bottom: 0.1m, 0.05m, 0.033m, 0.025m squared cells with  $2^3$  points per cell.**



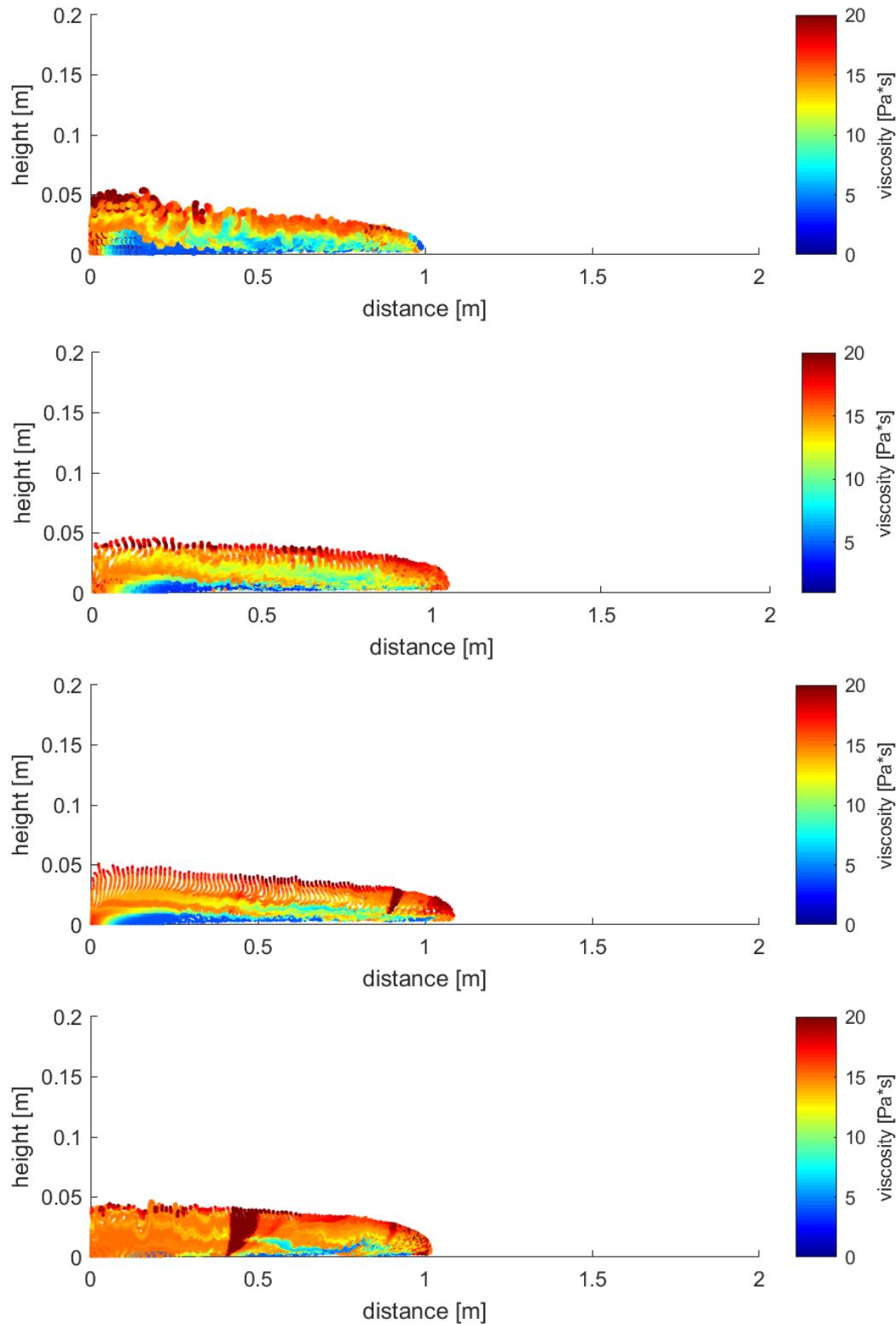
**Figure 35: Viscosity and runout as a function of time. Grid resolution sensitivity at time  $t = 5\text{s}$ . From top to bottom: 0.01m, 0.005m, 0.0033m, 0.0025m squared cells with  $2^3$  points per cell.**



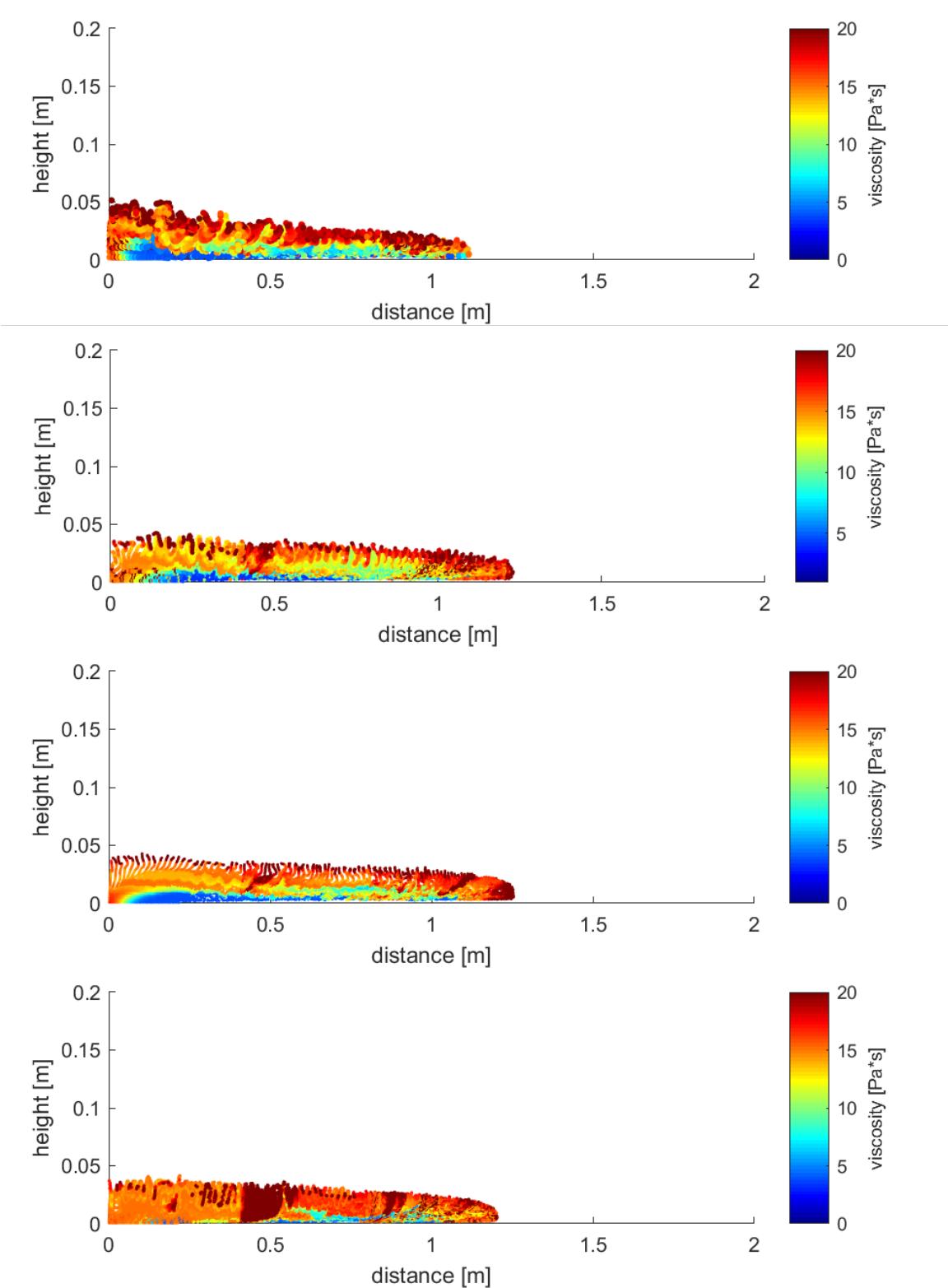
**Figure 36: Viscosity and runout as a function of time. Grid resolution sensitivity at time  $t = 15\text{s}$ .  
From top to bottom: 0.01m, 0.005m, 0.0033m, 0.0025m squared cells with  $2^3$  points per cell.**



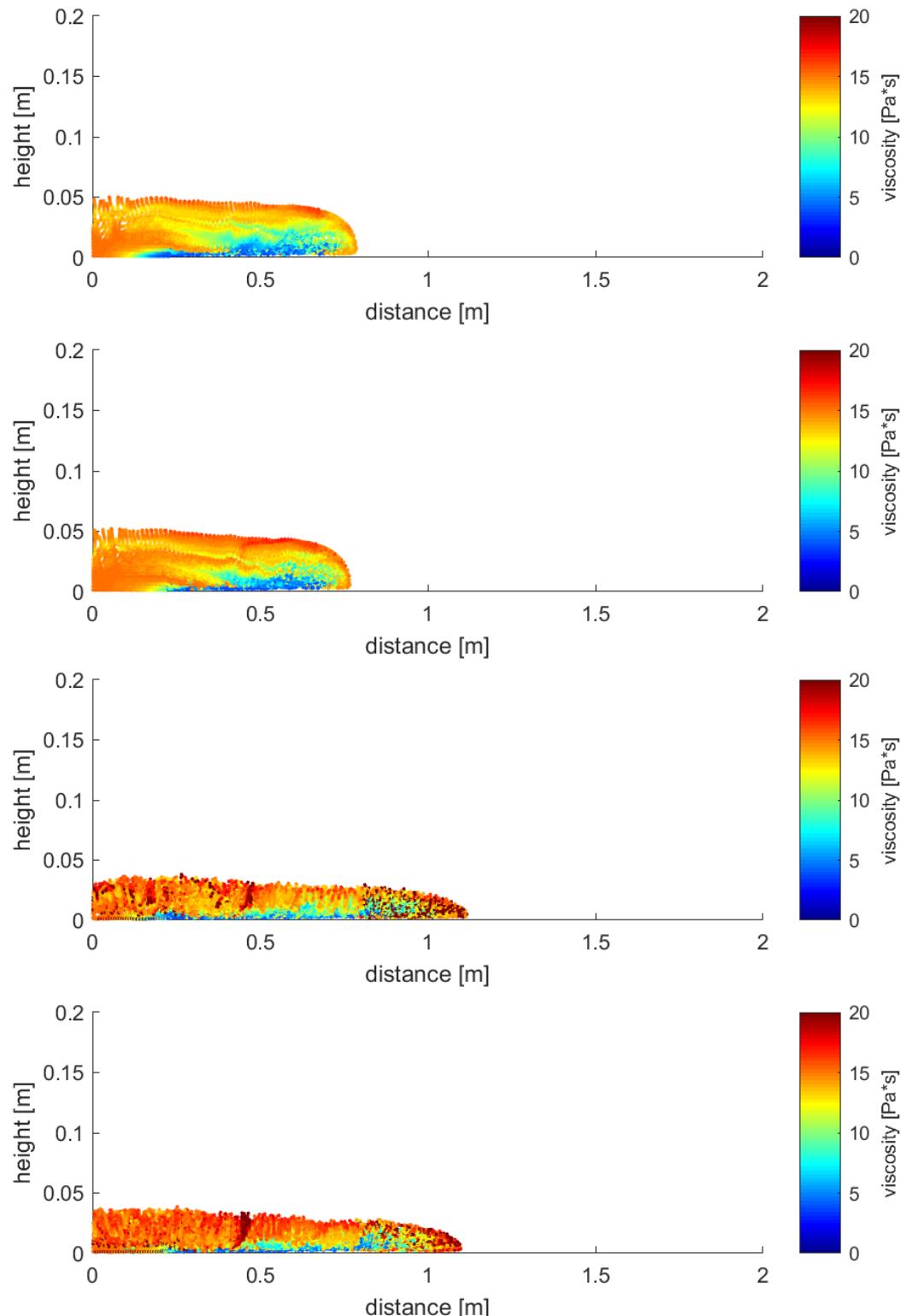
**Figure 37: Viscosity and runout as a function of time. Points per cell sensitivity at times  $t = 1\text{s}$ . From top to bottom: $2^3$ ,  $3^3$ ,  $4^3$  points on grid cells of  $0.01\text{m}$  and  $3^3$  points on grid cells of  $0.0033\text{m}$ .**



**Figure 38: Viscosity and runout as a function of time. Integration points sensitivity at times  $t = 5\text{s}$ . From top to bottom: $2^3$   $3^3$ ,  $4^3$  points on grid cells of  $0.01\text{m}$  and  $3^3$  points on grid cells of  $0.0033\text{m}$ .**

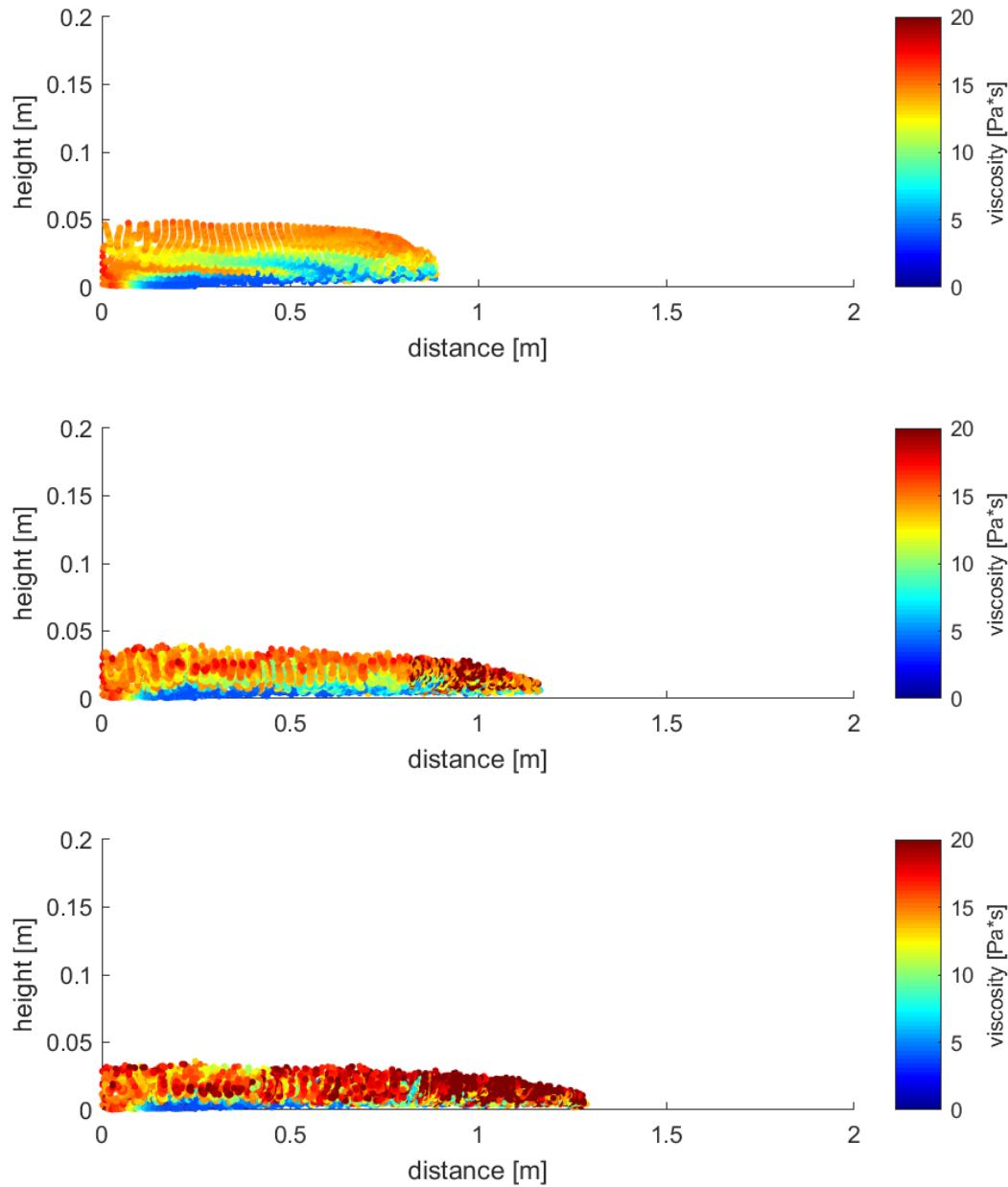


**Figure 39: Viscosity and runout as a function of time. Integration points sensitivity at times  $t = 15s$ .  
From top to bottom: $2^3 \times 3^3$ ,  $4^3$  points on grid cells of  $0.01m$  and  $3^3$  points on grid cells of  $0.0033m$ .**



**Figure 40: Viscosity and runout as a function of time. Sensitivity to timesteps at  $t = 1\text{s}$  and  $t = 13\text{s}$ .  
From top to bottom, 20% and 90% of the stability criteria.**

For CPDI interpolation, stability was not achieved for several tests that were ran. Therefore, no results are presented. These instabilities are however discussed in the next section.



**Figure 41: Viscosity and runout as a function of time. Sensitivity to interpolation method: 3<sup>rd</sup> order B-Splines at times  $t = 1\text{s}$ ,  $t = 5\text{s}$  and  $t = 15\text{s}$ .**

#### 4.5.2 Analysis of numerical sensitivity

For all the runs tested, the overall physical behavior observed seems reasonable. The initially high ( $15 \text{ Pa*s}$ ) viscosity value is quickly reduced to near-minimal values (4 to  $5 \text{ Pa*s}$ ) in the high shear zones near the bottom boundary as the element collapses. In other parts, viscosity is sheared to a lesser degree, according to the constitutive model implemented. When the flow progressively reaches a stop, the thixotropic behavior is accurately captured as the fluid regain its strength.

Firstly, the results were stable for a range of grid cell spacing. As shown on figure 36, the flume reached approximately 1.1 meters for all grid resolutions. The height profile is also closely comparable for the four analyses. A difference can be seen in the wave front, where more definition is found for the finest grid.

The distribution of viscosity values is also similar for all the resolutions tested. For the 0.0033 meter grid cells analysis, there is a patch of high viscosity that developed towards the end of the simulation. It is unclear what caused this sudden and localized increase in viscosity. One possibility is that it is simply a numerical artifact. It could also be hypothesized that the physical quantities in that region were close to a bifurcation point in the constitutive model. The effect on the quality of the results is minimal.

It should also be noted that there seems to be a stratification of viscosity values in the modeled runouts although the total variation is rather small.

The quality of results also increases with the number of particles used. Using  $3^3$  particles per cell on a coarse grid increases significantly the uniformity of the results and makes it comparable to using a high grid resolution. The same results were observed while using  $4^3$  particles per cell, but it is debatable whether the increase in accuracy can justify the increase in computational cost. In fact, the total number of particles jumped from 10 800 to 25 600 when using this increment in points per elements. For 2D analysis, a solution might be to use  $4 \times 4 \times 1$  points per element, but using such a technique in 3D analysis would create stability issues.

The plots shown for the stability to timestep are for 1 seconds and 13 seconds because the analysis with a timestep of 0.9 times the stability criteria crashed towards the end. Otherwise, little variation is seen between the two different timestep coefficients.

For CPDI interpolation, stability was not achieved for several tests that were ran regardless of the conditions. The reason is probably an increase to “kinematic anomalies”, a term coined by Professor Rebecca Brannon, who developed CPDI, to describe sudden increases in kinematic energy for no apparent reason. They are most often found when a fluid reaches near-equilibrium condition and their cause is not yet well understood. According to her personal website, using implicit time integration ensured stability for all conditions tested, further reinstating the need to develop an implicit version of the thixotropic model.

To conclude, it is safe to conclude that the thixotropic implementation in Uintah-MPM is unsensitive to variation in numerical parameters for a reasonable range of values.

### 4.5.3 Sensitivity to physical parameters

In this section, a sensitivity analysis to physical parameters was done. The analysis has been carried out on a grid of 0.01 meter elements with 3x3x1 material points per cell. Those numerical parameters were chosen following the analysis in the previous section. They provide optimum conditions in terms of accuracy of the results and computational cost for the current size and time scale.

Every analysis will be compared to a base scenario, which is the one used for the numerical sensitivity analysis. The goal is to determine qualitatively the influence of each parameter of the Coussot thixotropic model on lab-scale bench test experiments. Secondly, it is to determine a range of values that produce an observable difference in behavior at the given scale. Finally, it serves as a screening for bugs in the implementation of the equations.

In fact, a deeper understanding the variability of these parameters facilitates future calibration to experimental data as the user can have a basic understanding of their relative influence, which is most likely non-linear.

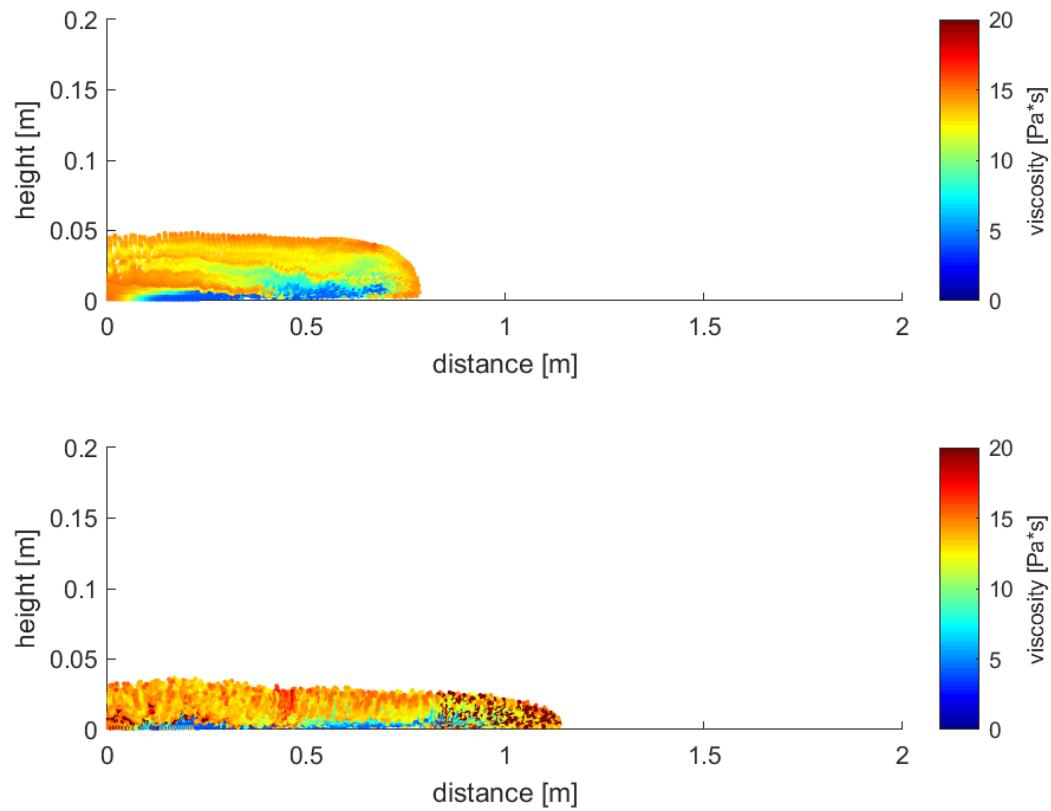
Only screenshots at 1 and 15 seconds will be presented for a selected set of case to reduce the total number of plots produced while still keeping track of phenomenology during the transient portion of the deformation.

Table 8 presents the overall set of simulations that were carried out to validate the different physical parameters reviewed. All the parameters show variation in line with what is expected when looking at the variation of the viscosity profiles under a 1D constant shear rate plot. Parameters that reduce the viscosity output from Coussot model shows longer runouts and the same is true for parameters that have the opposite effect.

The run “7c” crashed for unknown reasons. It can be speculated that the structure parameter was too high relatively to the other parameters as the numerical parameters should have been satisfactory to guarantee stability. Increasing the bulk modulus had no effect on the modeled runout distance. However, this might not be the case for large scale simulations, where hydrostatic pressure is more important. Since the bulk modulus affects the time step criteria, keeping in the range of smaller values is preferable, if it does not induce negative side effects. Higher density produced longer runouts compared to lower values. Finally, increasing the slope had a noticeable effect on runout distance and the material hit the bottom boundary before stoppage.

**Table 8: Overview of parameters varied in the numerical sensitivity analysis**

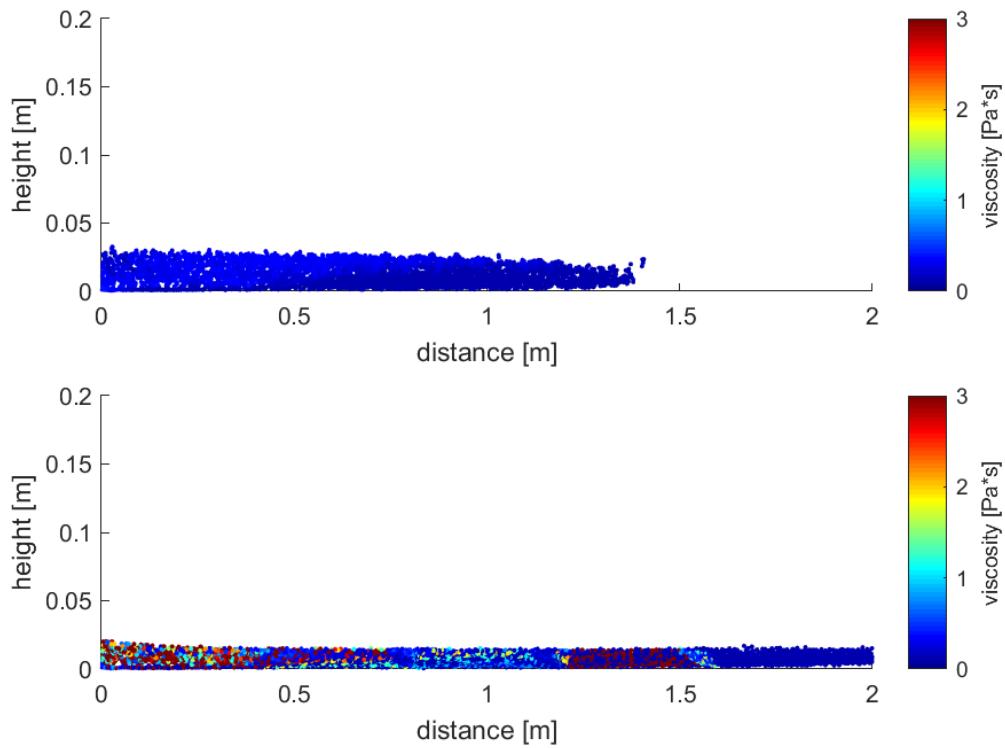
Run Number	Cousset Parameters					Bulk modulus (Pa)	Density (kg/m <sup>3</sup> )	Bottom slope angle (degrees)
	Minimum Viscosity	n	alpha	T	Init. Struct.			
<b>BASE</b>	4	2.5	1	0.5	1.5	15 000	1850	0deg
<b>5a</b>	0.1	2.5	1	0.5	1.5	15 000	1850	0deg
<b>5b</b>	1	2.5	1	0.5	1.5	15 000	1850	0deg
<b>6a</b>	4	1.5	1	0.5	1.5	15 000	1850	0deg
<b>6b</b>	4	2	1	0.5	1.5	15 000	1850	0deg
<b>6c</b>	4	3	1	0.5	1.5	15 000	1850	0deg
<b>7a</b>	4	2.5	0.5	0.5	1.5	15 000	1850	0deg
<b>7b</b>	4	2.5	2	0.5	1.5	15 000	1850	0deg
<b>7c</b>	4	2.5	10	0.5	1.5	15 000	1850	0deg
<b>8a</b>	4	2.5	1	0.5	1	15 000	1850	0deg
<b>8b</b>	4	2.5	1	0.5	2	15 000	1850	0deg
<b>8c</b>	4	2.5	1	0.5	10	15 000	1850	0deg
<b>9a</b>	4	2.5	1	0.5	1.5	1.5 *10 <sup>6</sup>	1850	0 deg
<b>10a</b>	4	2.5	1	0.5	1.5	15 000	1200	0 deg
<b>10b</b>	4	2.5	1	0.5	1.5	15 000	3000	0 deg



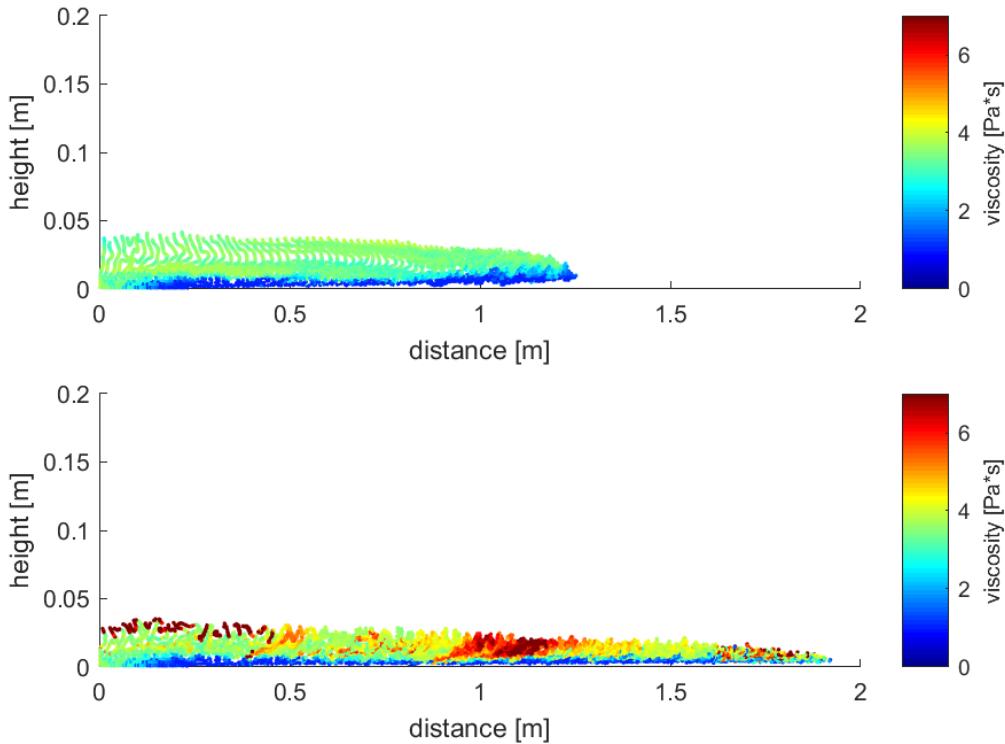
**Figure 42: Base test case for sensitivity analysis to physical parameters. Viscosity and runout as a function of time. From top to bottom, at times  $t = 1\text{s}$  and  $t = 15\text{s}$**

The physical meaning of the different parameters in Coussot's thixotropic model can be hard to define. All the parameters, except the minimum viscosity are empirical. The minimum viscosity could be obtained by rheological testing at very high shear rates. The initial structure controls the initial conditions of the material. It can probably be linked to resting time before shearing begins, but further testing would be required to validate this hypothesis.

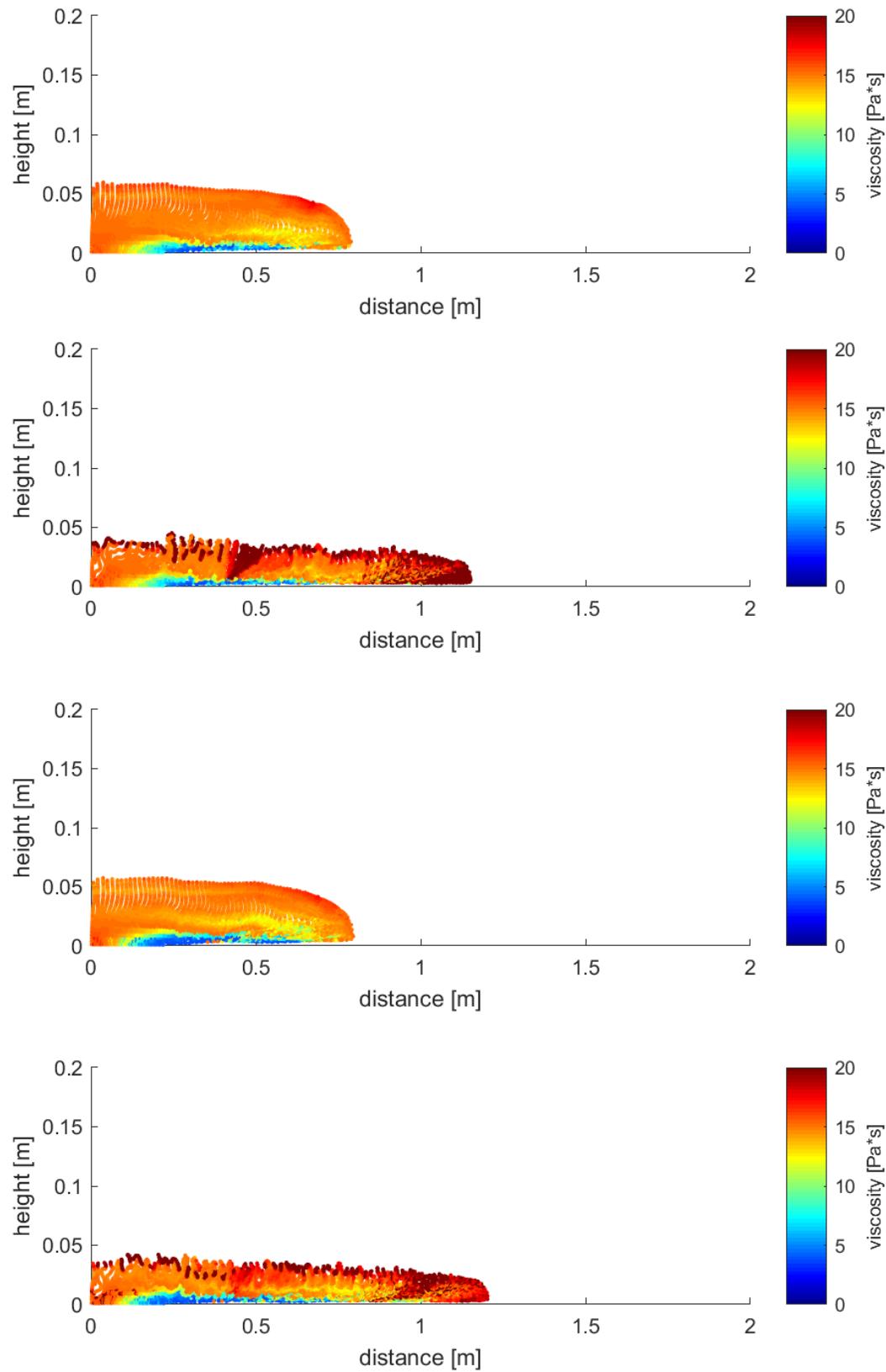
Finally, the other parameters are empirical, and seem to control the expressed viscosity to different degrees. The strongest parameter is the "n" exponent, which controls the rate of viscosity increase or decrease. The higher the "n" value, the greater the rate of viscosity decrease under high shear, and the greater the rate of ageing and viscosity increase under slow shear.



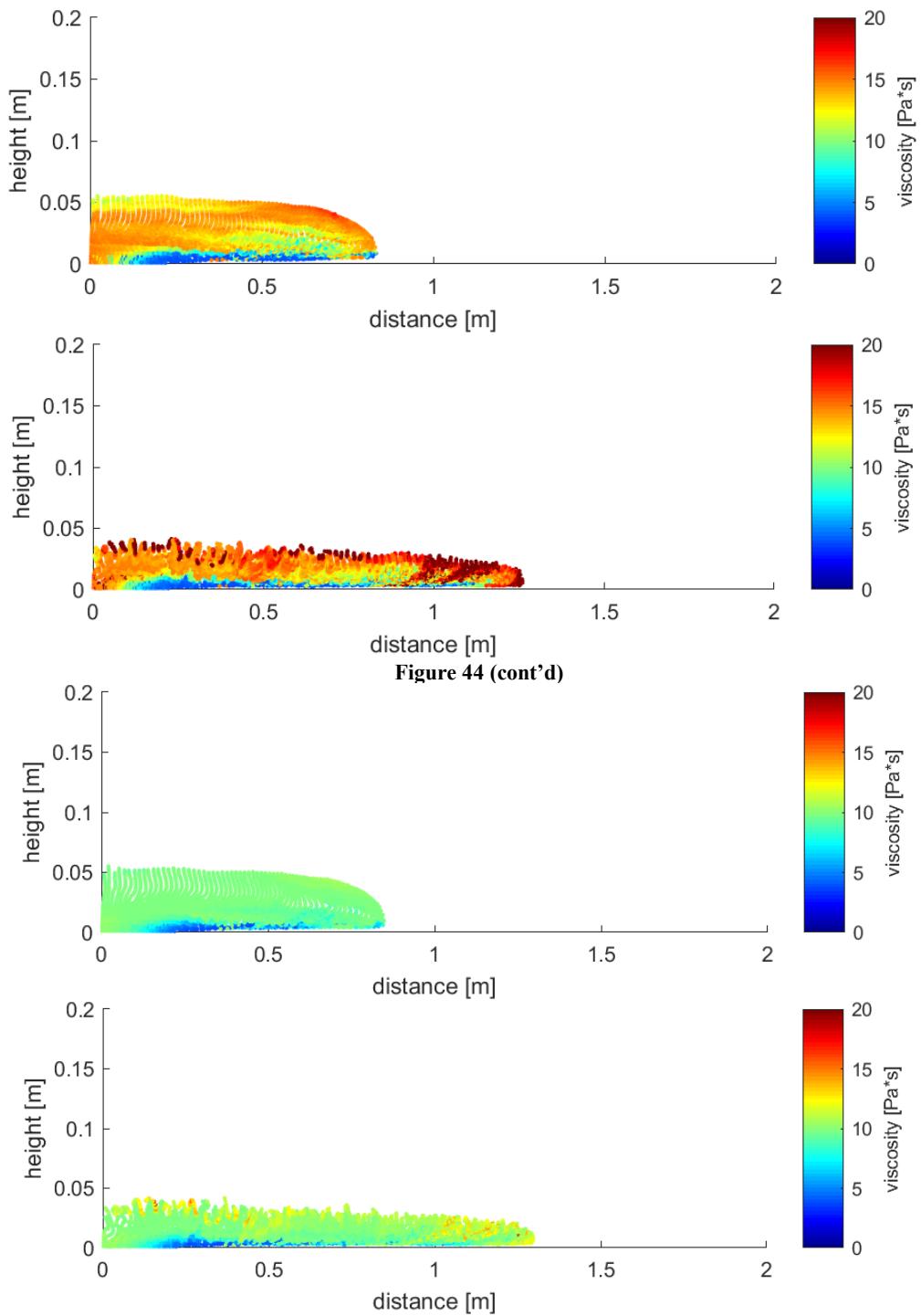
**Figure 43:** Sensitivity to minimum viscosity ( $0.1 \text{ Pa}^*\text{s}$ ) at time  $t = 1\text{s}$  and  $t = 15\text{s}$ .



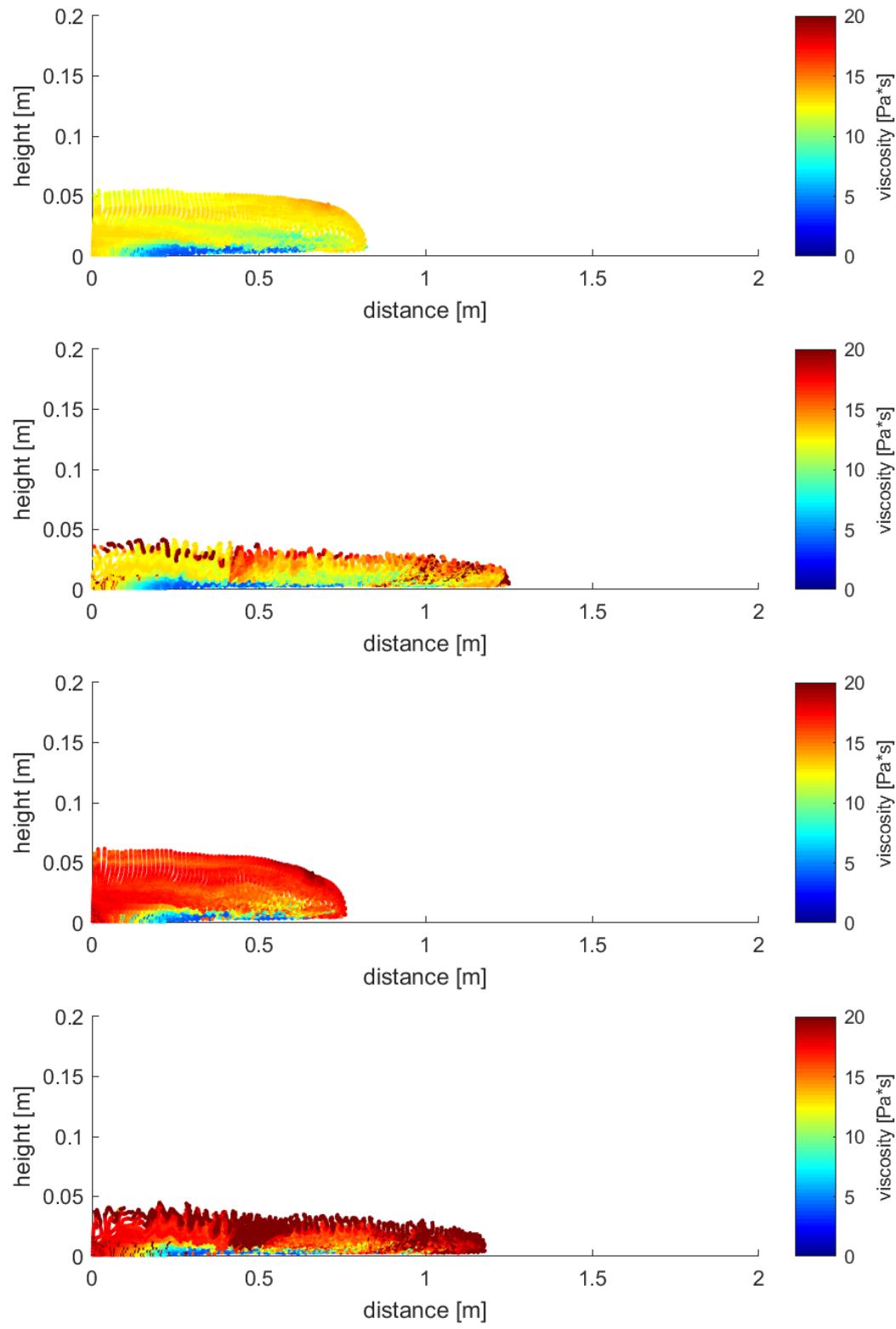
**Figure 44:** Sensitivity to minimum viscosity ( $1 \text{ Pa}^*\text{s}$ ) at time  $t = 1\text{s}$  and  $t = 15\text{s}$ .



**Figure 45: Runout at 1 s and 15 s for different alpha values (0.2, 2, 10 respectively)**



**Figure 46 : Runout at 1 s and 15 s for different  $n$  values (1.5, 2, 3 respectively)**



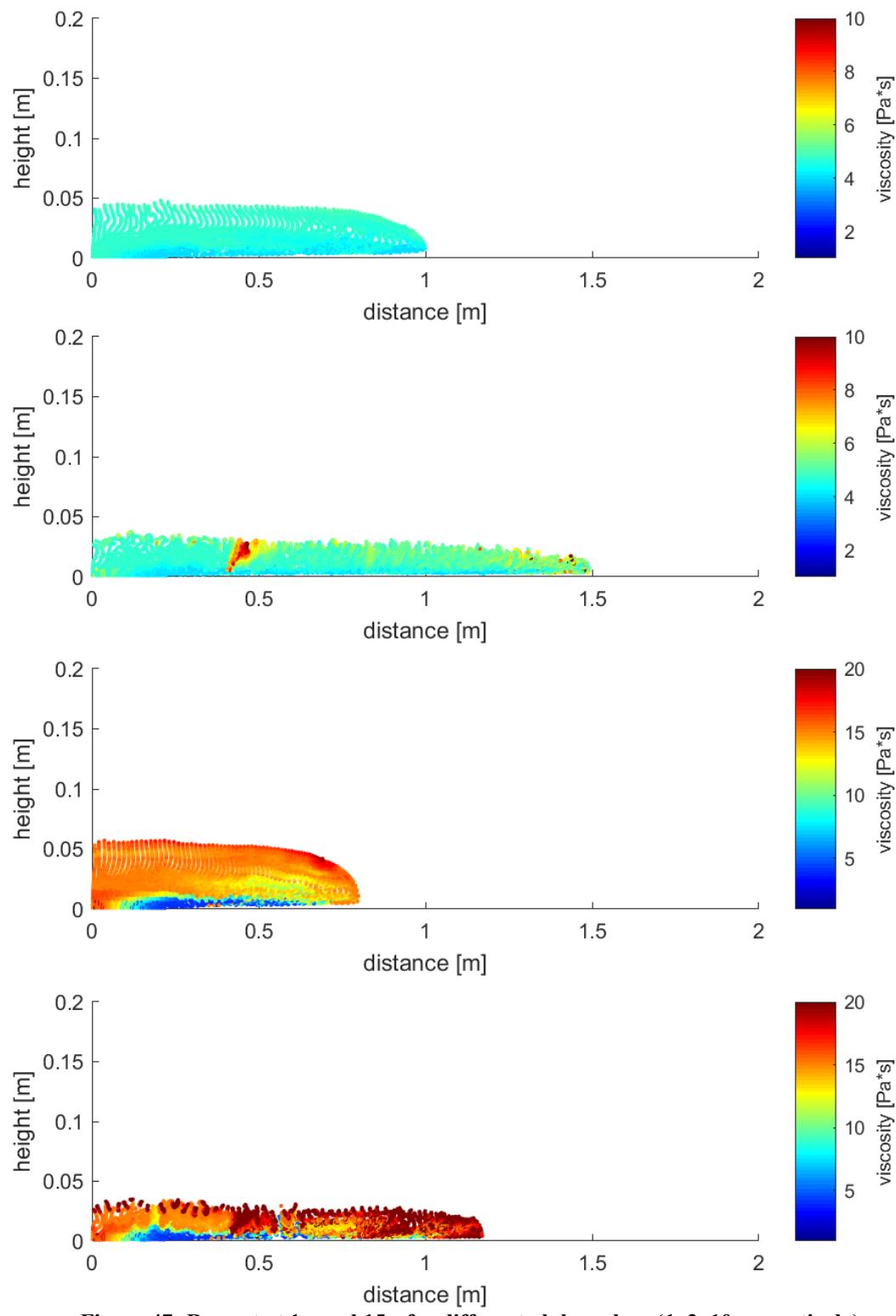


Figure 47: Runout at 1 s and 15 s for different alpha values (1, 2, 10 respectively)

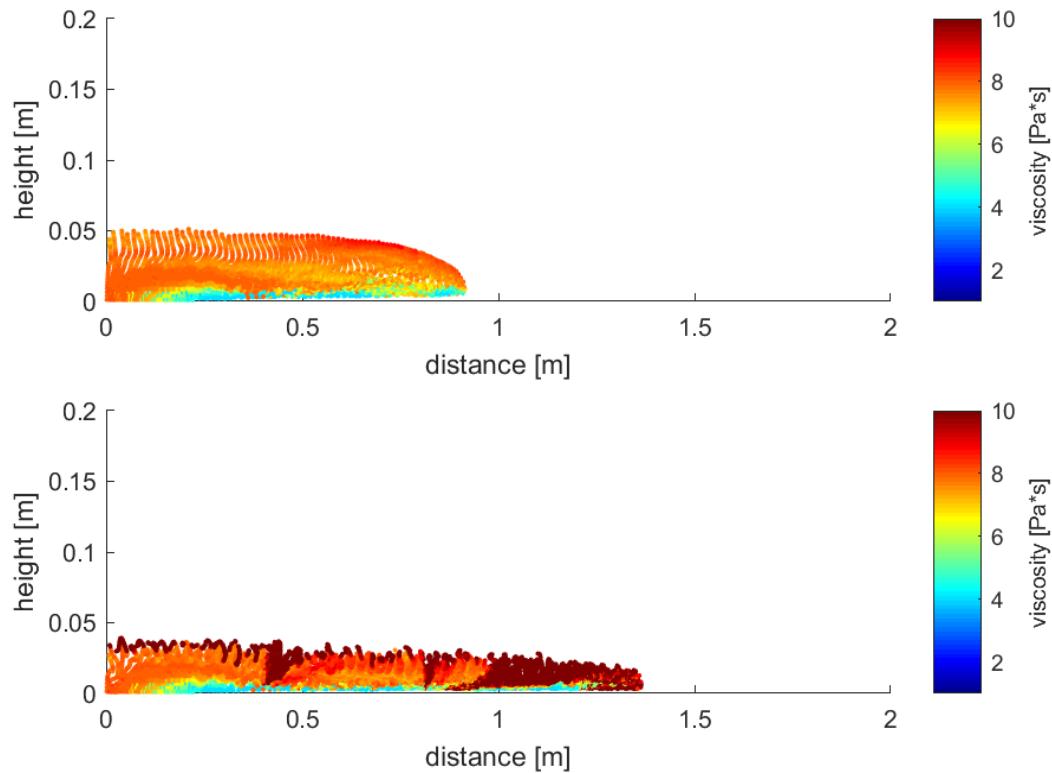


Figure 46 (cont'd)

## **5: Large scale 3D modeling scenarios**

With the calibration work presented in chapter 4 being conclusive, a set of more complex scenarios were simulated. They range in size from laboratory experiments to large scale deposition or dam break events. They also range in complexity and aim to model 2D and 3D cases.

The main goals of this chapter were to: (1) deepen the exploration of the thixotropic constitutive model by direct comparison to Bingham rheology, (2) explore flow behavior of thixotropic constitutive models at large scale and (3) validate robustness of the software by scaling the physical parameters to field scale values (high stress states, important velocities and significant length of time).

### **5.1 Scalability of rheological models to field scale**

First, a scalability analysis was done for the different rheological models implemented. This was done to validate assumptions that rheological models at bench scale will scale proportionally to large scale scenarios where stresses and strain rates are much more significant.

Three rheological models were considered. A Bingham model was compared with two thixotropic models. For the first thixotropic material, the aging parameter (exponent n) was put at 2.5 to simulate a faster divergence towards very high viscosity values. This was of interest as previous work presented in chapter 4 (sections 4.4 and 4.5.2) highlights the importance of ‘n’ in the response of the material, using Coussot’s thixotropic model.

The other thixotropic model is based on previous researcher's calibration of Coussot's model with laboratory data on oil sand tailings (Mizani, 2016). Table 9 presents a summary of the chosen parameters. The three models were chosen in such a way that they are within the range of normally accepted values for bench scale experiments on mine tailings. They are also chosen so that their runout, at small scale, is quasi-identical.

**Table 9: Baseline material parameters for numerical analysis**

Parameter	Bingham	Coussot 1	Coussot 2
Yield Stress (Pa)	30	-	-
Viscosity (Pa*s)	0.05	-	-
$\alpha*T$	-	0.36	0.36
n	-	2.5	1.076
Minimum Viscosity (Pa*s)	-	1	1

The total runout distance and the shape of the free-surface at stoppage are measured under the same conditions for 0.2 m, 1 m, 10 m and 100 m where the distances are normalized by the initial length.

The following figure presents the normalized runout distances as a function of scale for the three rheological models. Figure 48 also shows the time required to reach static equilibrium conditions.

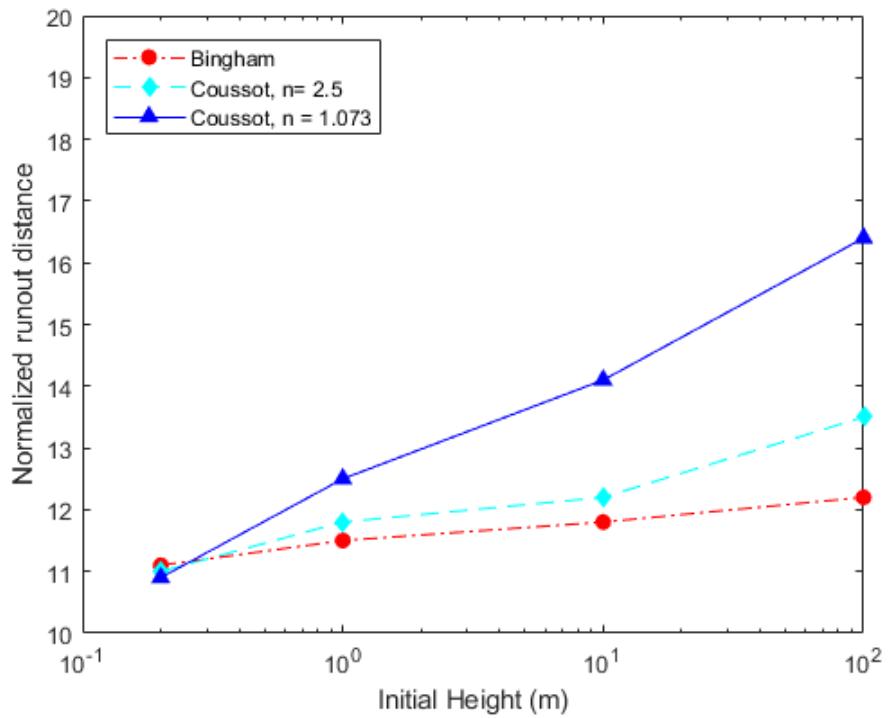


Figure 48: Runout distance as a function of scale and constitutive model used

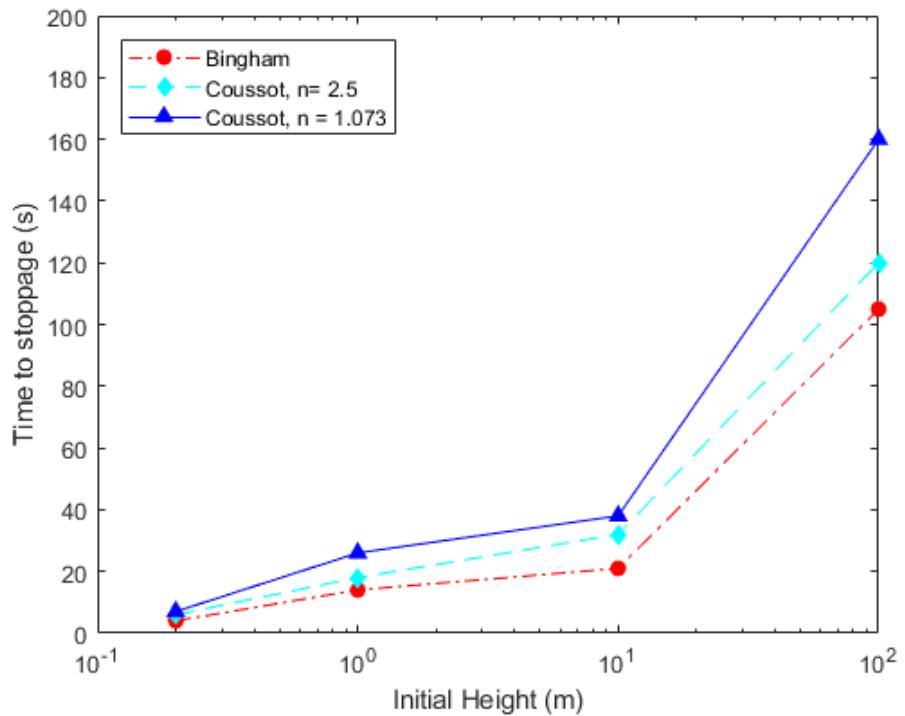
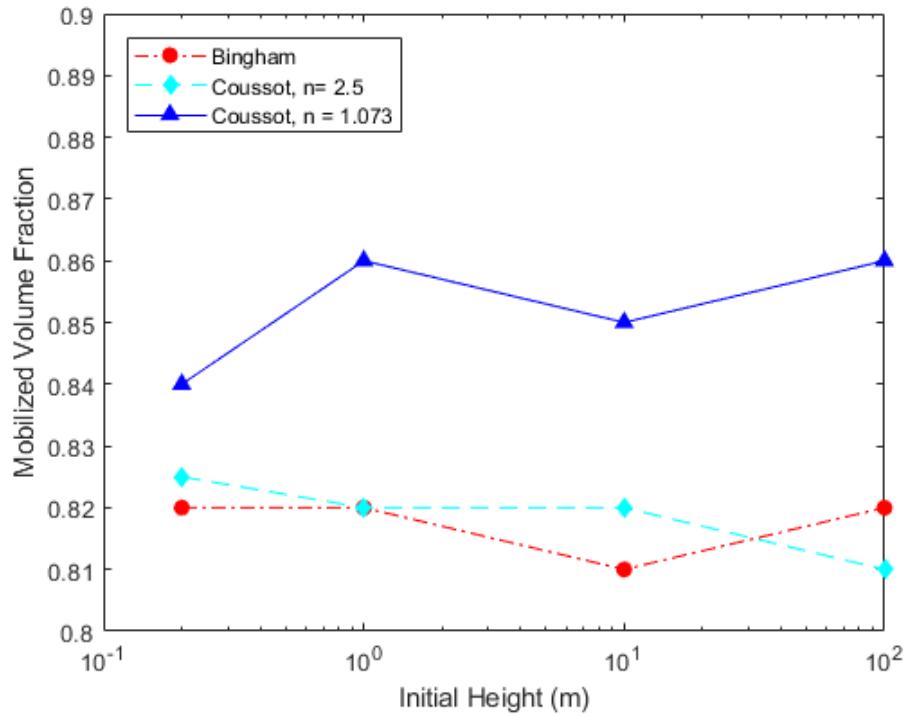


Figure 49: Time to static equilibrium for different constitutive models at different scales



**Figure 50: Mobilized fraction of the impoundment**

Although the three constitutive models created a runout distance of approximately 11 times the initial height, variation becomes more and more important as the scale grows. For the largest simulation, which are indicative of field scale conditions for dam break scenarios, the difference in runout distance is over 7.5% and 35%, respectively, when compared to the Bingham model that created identical results at small scale.

One of the factors contributing to this important difference is most likely the inability of the material to reach a yielded state quickly when the shear efforts are more important than in a laboratory setting. Previous research has shown that, for materials undergoing a high level of shearing, Bingham rheology might be inadequate (Mizani & al., 2016b) for which rheological behavior is better explained using an aging parameter to describe the evolution of viscosity.

Figure 49 presents the amount of tailings that is mobilized during the flow process. A certain fraction of the material will remain within the domain of the dam. This has an influence on the final runout as a greater volume of material could potentially be mobilized. However, the change in 2% is the total volume is relatively small when compared to the range of differences measured for runout length, with the different constitutive models.

Back calculation of dam breaks using Bingham rheology can sometimes be inconclusive as the runouts are longer than what would be expected. For example, back analysis of runouts using Bingham rheology presented at a 2019 Tailings and Mine Waste Dam Breach Workshop often showed quite low yield stress ( $< 20$  Pa), which appears physically unreasonable given the state of the tailings at the breach (e.g. tailings that are much denser than at initial deposition). Similarly, the methods of Stark & Olsen (2017), which uses liquefied strength ratios instead of yield stresses often reports low values; much lower than the expected residual strength ratio ( $< < 0.07$ ). The numerical simulations carried in this work seem to provide a partial answer to that question.

## 5.2 Large scale dam breaks

A set of large scale dam break simulations were carried out to reproduce an event that occurred at the Xiangjiang Wanji Aluminum plant in Luoyang, Henan Province in China. In 2016, 2 million cubic meters of red mud, a by-product of the aluminum production process, was released. Although important economic and environmental damage has been done to the area, no one was injured or killed as early evacuations have been carried out, reportedly.

Figure 50 presents the state of the area after the dam collapse. The total runout distance was approximately 1.5 km. The following figure shows the evolution of the landscape in the area throughout the years. Figure 51 shows the initial area, where tailings were stored in the smaller adjacent tailing storage facility in the early phases of the factory's operational phase. Then, a second TSF of higher capacity is built in the adjacent valley. It is used until 2016 where satellite imagery in 2017 show the extent of the runout.

Open-source satellite imagery from Google Earth's archives has been used to determine the topography on site. It is composed of a valley with approximately 200 meters in width.

This case was chosen for the relative simplicity of its boundary conditions. The geometry of the valley is easy to determine by satellite data. Secondly, it can be simplified with a constant slope while still providing a reasonable approximation. Altitude points have been measured for 3 cross section of the area affected. The average longitudinal slope was determined to be 2.6%. The total number of altimetric data points used was 45.



**Figure 51: State of the tailings after runout (Photos from watcher.news, 2016)**



**Figure 52:** Satellite imagery of the tailings runout, before and after the dam collapse

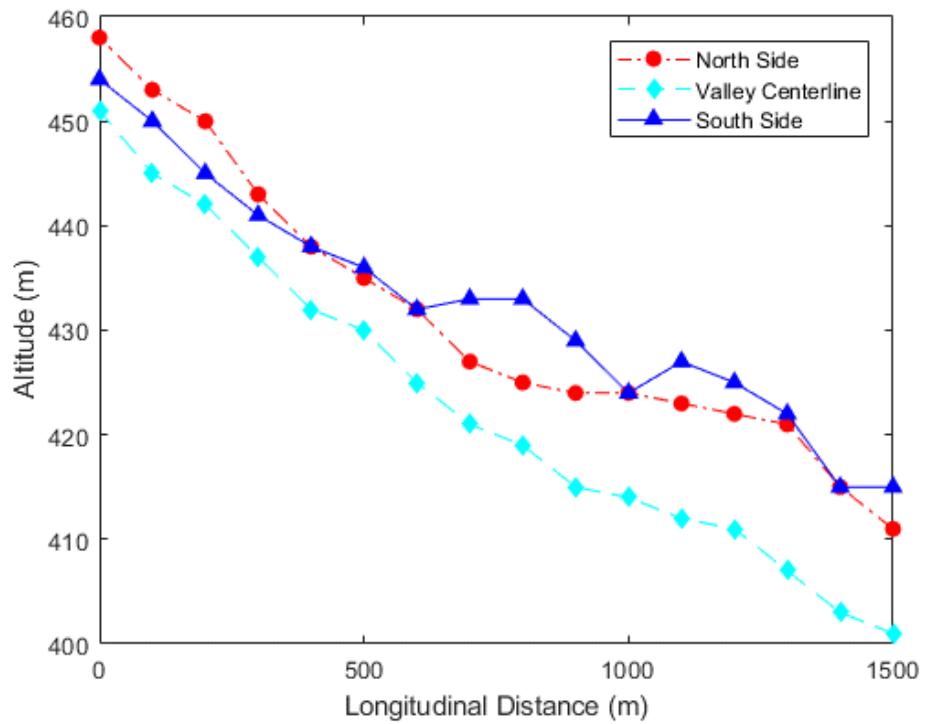


Figure 53: Topography of the valley surrounding the TSF

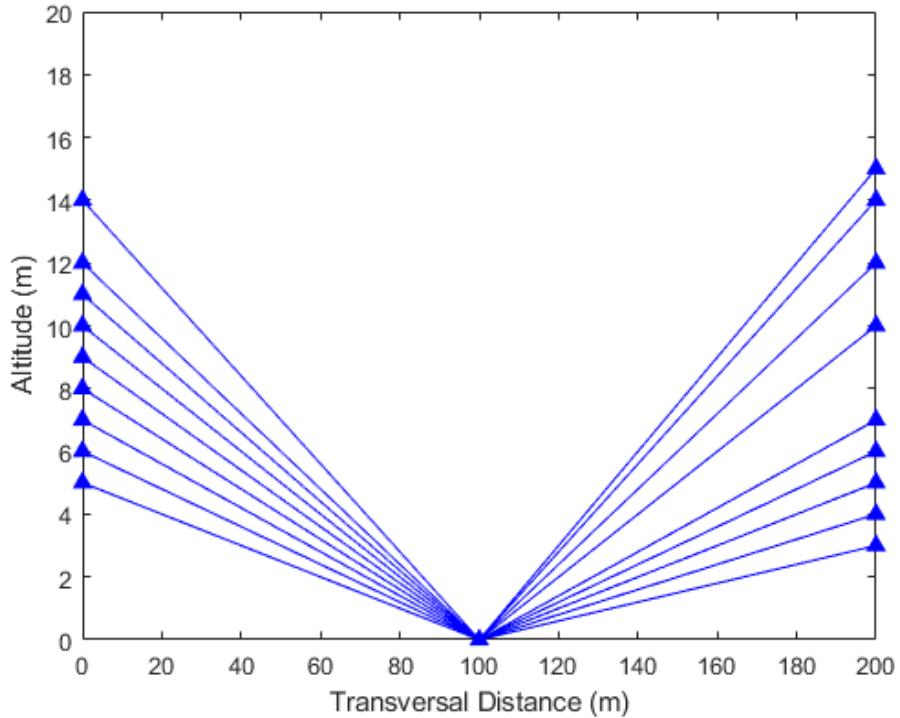
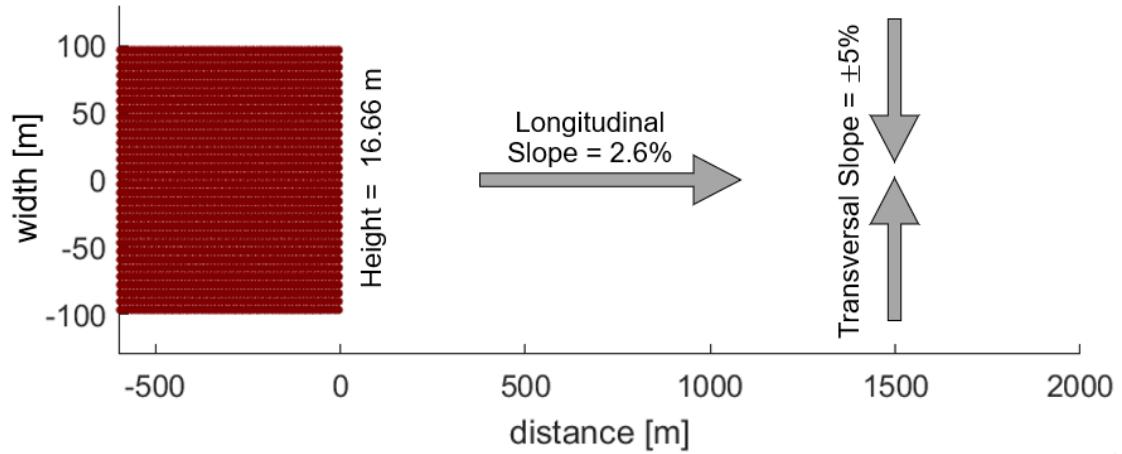


Figure 54: Altimetric profile of the valley at different cross sections

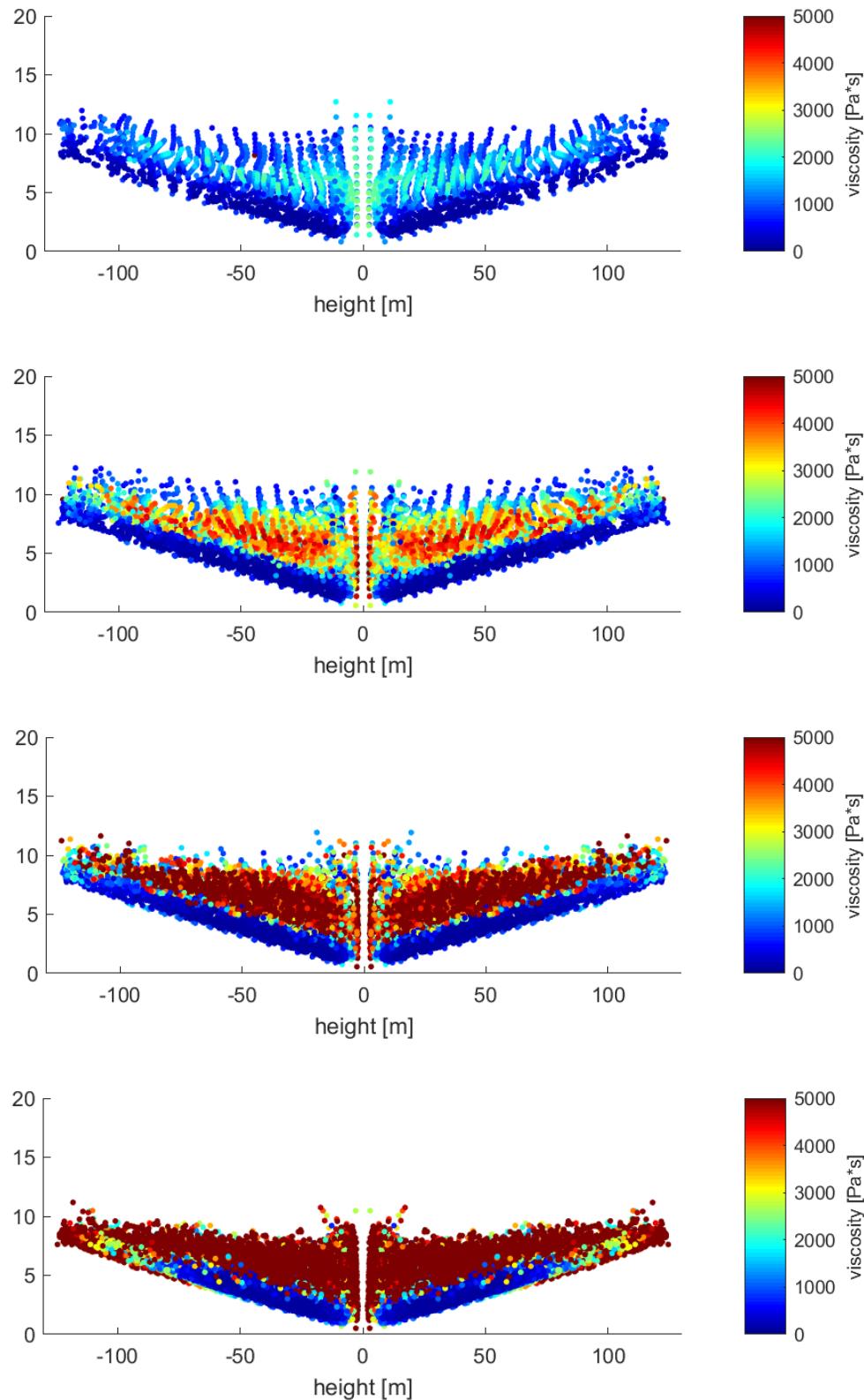
The transversal slope is smaller near the impoundment and it increases as the distance downstream progresses, as shown by figures 52 and 53. The slopes measured here are probably a lot steeper than they are on site due to the lack of datapoints. The valley is more likely to be U-shaped instead of V-shaped and the higher slopes are more pronounced further down the valley. However, it was not possible to access data with higher resolution. The longitudinal slope that was considered for the numerical model is 2.6%. For the transversal slope, 5% was chosen. The initial conditions are presented in figure 54.



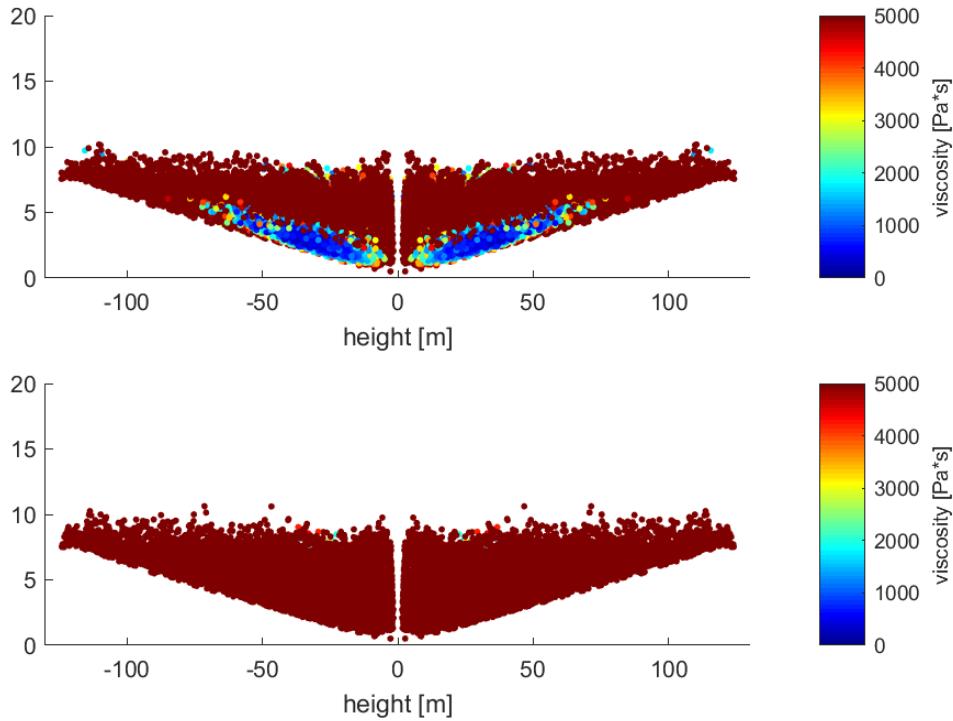
**Figure 55: Initial conditions for 3D dambreak analysis**

Rheological parameters that are typical of mine tailings were assumed for the 3D propagation case. The effect of each parameter on the runout distance has also been investigated afterwards.

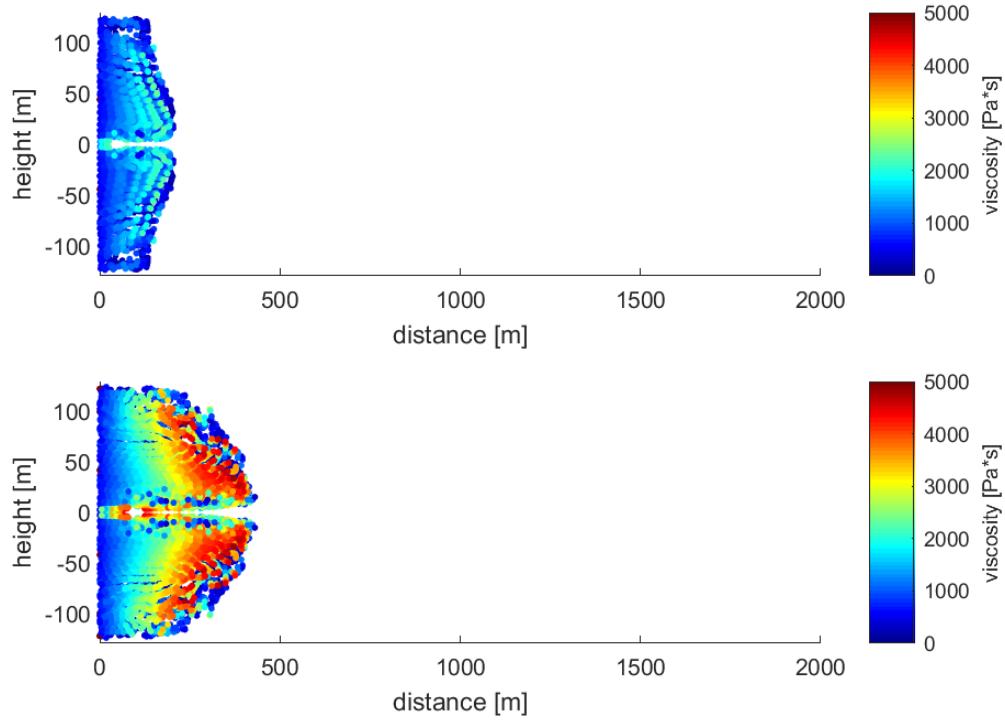
The following two figures show the evolution of the wave front at different time until stoppage. The figure presents snapshots at 30, 60, 90, 150, 250 and 330 seconds for both cross sectional and top views respectively. An axis of symmetry down the middle of the flume was used to reduce the number of points in the computation.



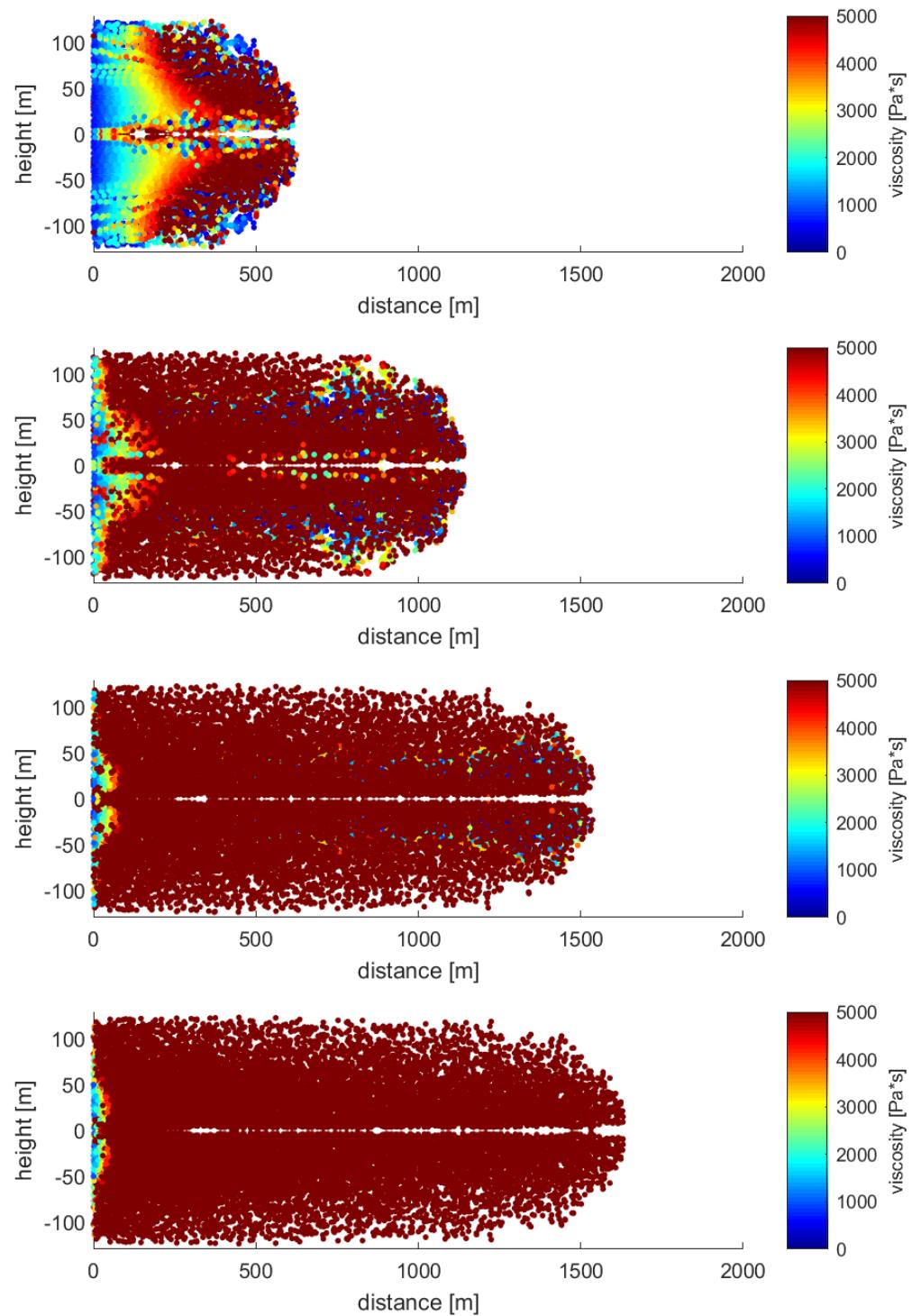
**Figure 56:** Cross sectional view of dam break flow at 30, 60, 90, 150, 250 and 330 seconds



**Figure 55 (cont'd)**



**Figure 57: Plan view of dam break flow at 30, 60, 90, 150, 250, 330 seconds**



**Figure 56 (cont'd)**

Interesting behavior can be noted as the extend of the runout develops in time. Firstly, at small timescales (the first 30 seconds) the range of values for viscosity is fairly small as the initially hardened material experiences enough shear to reduce viscosity to near-minimal values. However, as the material experiences less shearing efforts while flowing down the inclined plane, structuration effects are able to become dominant again and viscosity increases at a rate inversely proportional to the intensity of shearing efforts.

After 90 seconds, two distinctive areas appear in the runout. The bottom half, experiencing prolonged shearing, has minimum viscosity whereas the upper half doesn't experience enough stress to stay in that state. This is analog to plug flow observed in Bingham fluids where the material is divided in yielded and unyielded sections. As the flow slows down, viscosity increases until everything reaches stoppage, which occurs roughly after 330 seconds. The total runout is approximately 1550 meters, which is within 10 % of the observed runout, if accounting for precision of the measurements.

To investigate the influence of the different engineering parameters on the runout, a sensitivity analysis has been carried out. Since no rheological data was available on the properties of the tailings in scientific literature. Estimates have been made out of average geotechnical and rheological properties of bauxite tailings and how they compare to other types of tailings in the natural resource extraction sector such as hard rock or oil sand tailings. However, since a full analysis encompassed several simulations, the case has been reduced to a 2D scenario to accelerate computing.

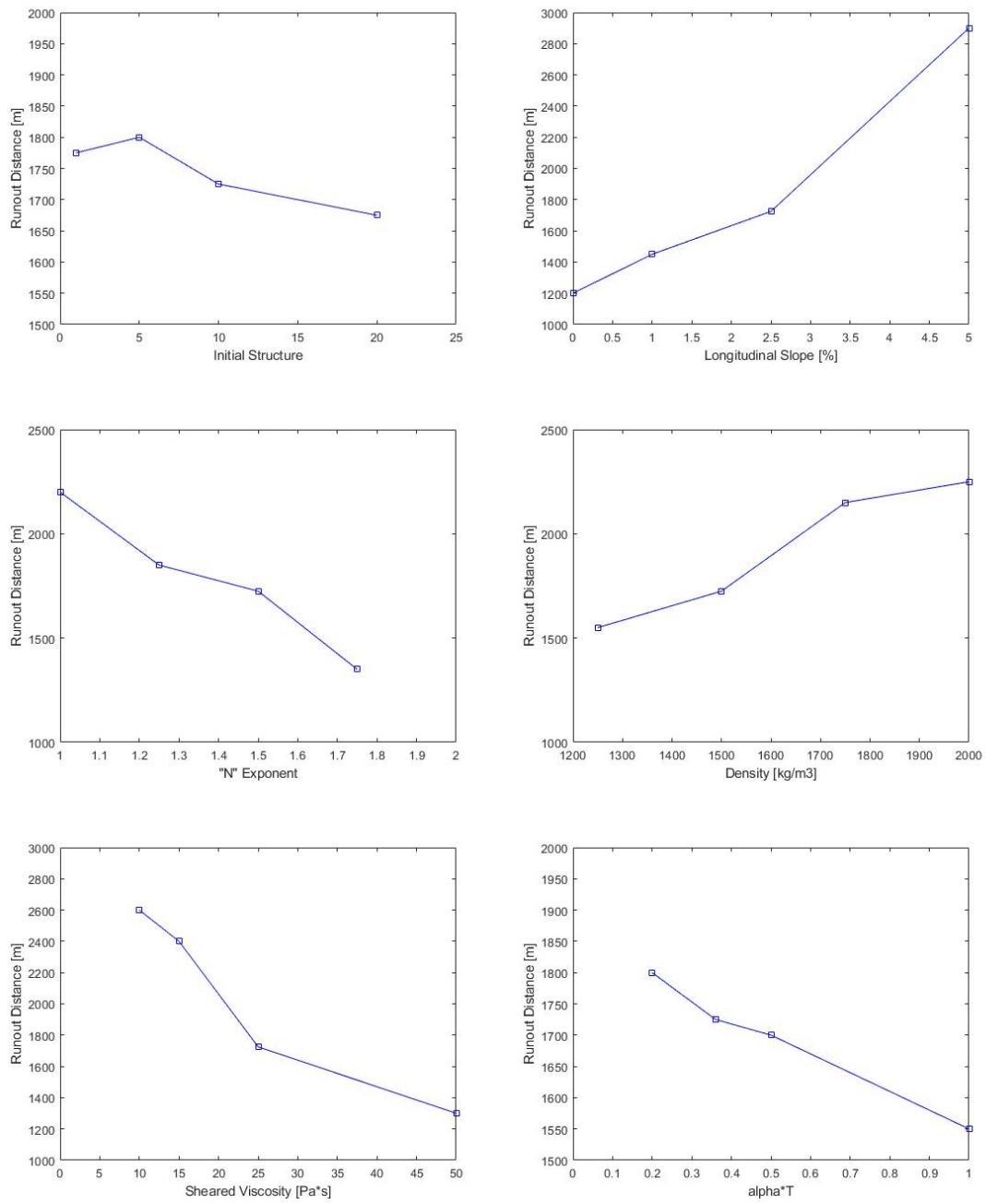
Bauxite tailings are considered a fine-grained, low plasticity material which falls under ML in the USCS classification system. It has low hydraulic conductivity and exhibits static liquefaction potential, like most tailings (Gore, 2017)

Therefore, the parameters previously used by Mizani are used. The minimum viscosity has been slightly augmented to account for the higher solids content than fresh oil sands tailings and irreversible dewatering that took place through consolidation and evaporation over time which is not accounted for in Coussot's thixotropic model.

Table 10 presents some physical and numerical parameters used in the computation. It is natural to say that the numerical simulations completed are more to serve as proof-of-concept and to explore possible ramifications of thixotropy on post-rupture flow phenomenology. For the sensitivity analysis, the parameters were varied in ranges that were similar to the sensitivity analysis to physical parameters carried out in section 4.5.2.

**Table 10 : Physical and numerical parameters for sensitivity analysis**

Parameter	Value
<b>Physical</b>	
Viscosity (Pa*s)	50
$\alpha$	0.36
T	1
n	1.25
Viscosity Cap (Pa*s)	25 000
Initial Length (m)	600
Initial Height (m)	16.67
Longitudinal slope (%)	2.6
<b>Numerical</b>	
Element width (m)	10
Element height (m)	2
Points per cell	$2^2$
Total points	2160



**Figure 58 : Sensitivity of runout length to physical parameters**

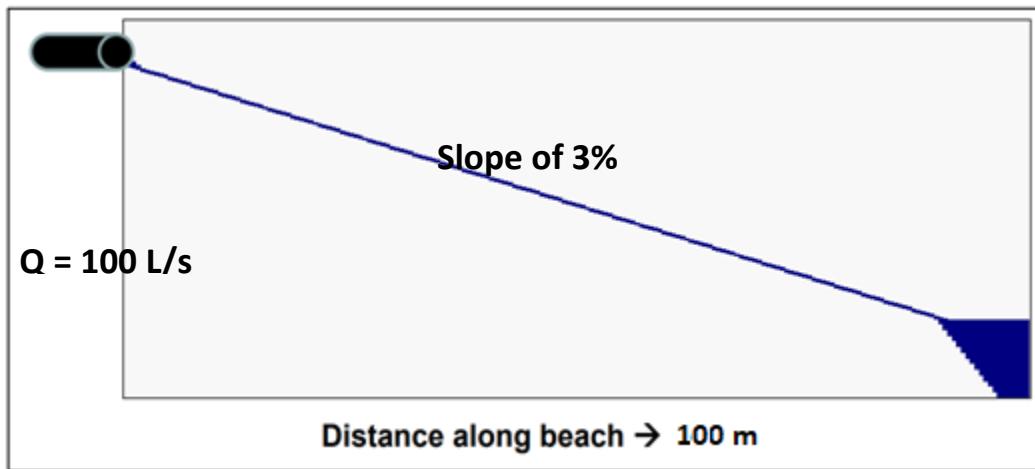
The parameters chosen, when used for a 2D simulation, provided runout that were a bit longer than in the three dimensional case.

In 2D, the total runout reached 1750 meters instead of 1550 meters. This is most likely due to increased dissipation of energy in the transversal direction in 3D simulations.

The values for the numerical simulations range from 1200 to 3000 meters. Some parameters such as the fully sheared viscosity or the slope have a significant influence on the runout distance. Other parameters, such as  $\alpha^*T$ , only affected the runout by a few hundred meters for the range of values tested. Even if the runouts are often much longer than the real-life result, it is better to be on the conservative side when it comes to modeling catastrophic events for which we lack a deep understanding.

### 5.3 Large scale pipe deposition

The problem of large scale flow from pipe deposition was also tackled with a thixotropic rheological model. This was done to investigate the free surface flow behavior of a thixotropic material in a setting that would closely resemble deposition in a tailings storage facility. Behavior is compared with observed behavior of tailings seen during deposition, as reviewed in chapter 2.1.



**Figure 59 : Sketch of the large scale pipe deposition scenario**

A pipe deposits the tailings at a rate of 100 L/s on an inclined plane with a slope of 3%. The flow lasts for a total of 180 seconds (three minutes) and then an extra 40 seconds is allocated to allow the material to fully age and reach static equilibrium conditions. The total simulated time is therefore 220 seconds.

The rheological parameters were chosen in order to closely match thixotropic parameters for mine tailings. Mizani (2016) provides a set of parameters that were used for a basepoint. However, the exponent “n” parameter was changed for two reasons. Firstly, it

is suggested that a high exponent can lead to the pseudo-plug flow that is often seen in mine tailings.

Secondly, it has the effect of accelerating aging in the material. Since the timescale are still somewhat limited by the explicit time steps, faster aging was used to see faster apparition of different phenomena while staying within the computation limits of the software in its current form. Table 11 provides a summary of the physical parameters chosen.

**Table 11: Physical properties for large scale pipe deposition simulation**

Parameter	Value
$\mu_0$ (Pa*s)	1
$\alpha T$	0.36
n	1.75
Density (kg/m <sup>3</sup> )	1850
Pipe area (m <sup>2</sup> )	0.0625
Flow rate (m <sup>3</sup> /s)	0.1
Bulk Modulus (Pa)	50 000

The first seconds of deposition are at an elevation of 0 to 0.3 meters then, the height of the end of the pipe is placed 1 meter above the surface. This change in vertical elevation of the deposition point is done in order to reduce splashing in the initial phase of the deposition. Since the mass represented by every point is rather small, dropping the particles from an important height will induce non-physical pressure oscillations of the material.

However, when particles fall on themselves, there is a damping effect that prevents the apparition of most of these instabilities.

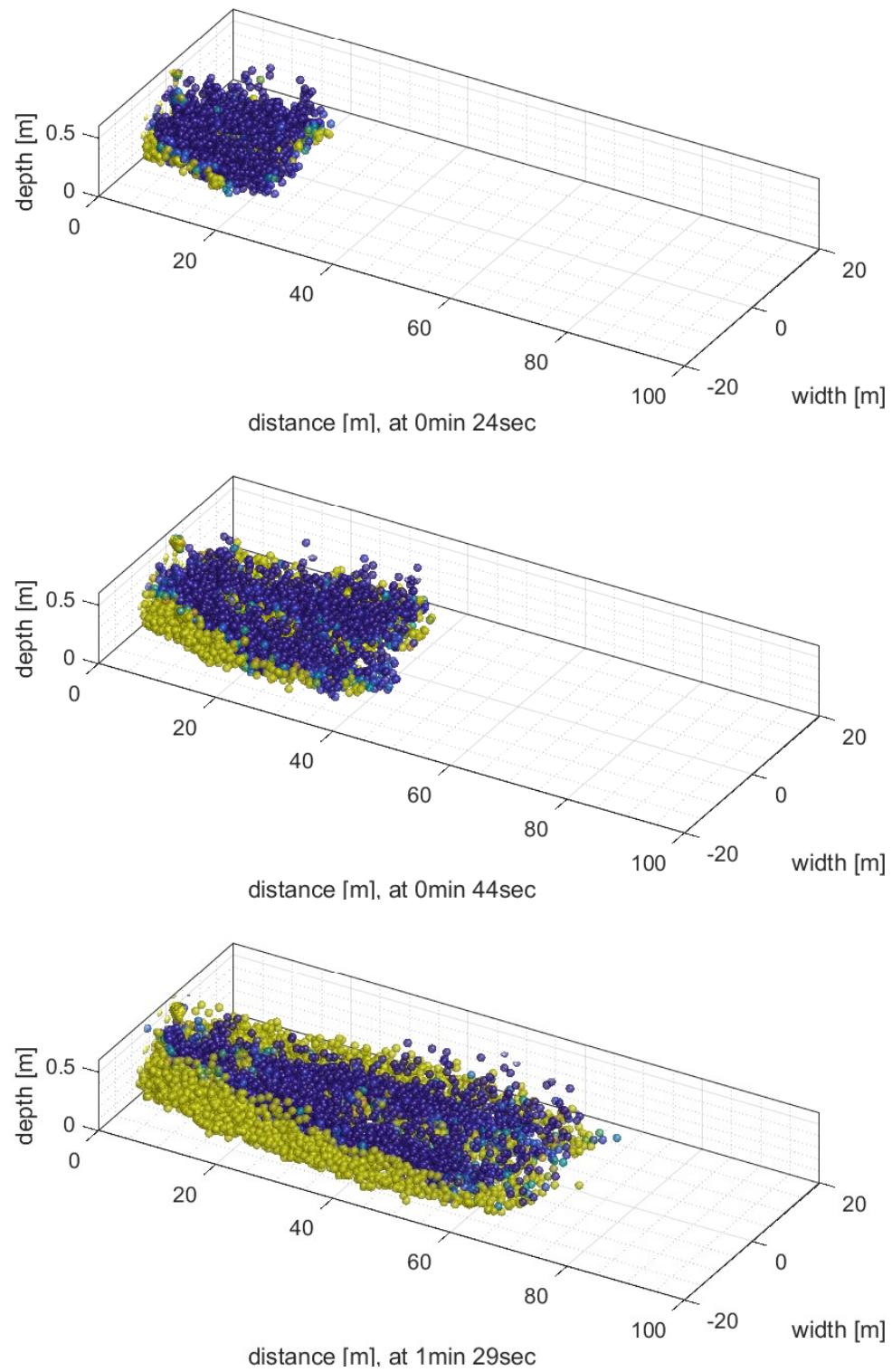
Another way to reduce pressure oscillations would be to increase the diameter of the pipe depositing the particles. The lower initial velocity would potentially reduce the undesired and non-physical splashing effects.

The boundaries of the computational domain are located transversally 25 meters on each side of the deposition point and 150 meters in the longitudinal direction. Therefore, the dimension of the box encompassing the entire computational domain are 150 m by 50 m by 2 m. This was big enough to model the entire runout without having it reach the boundaries.

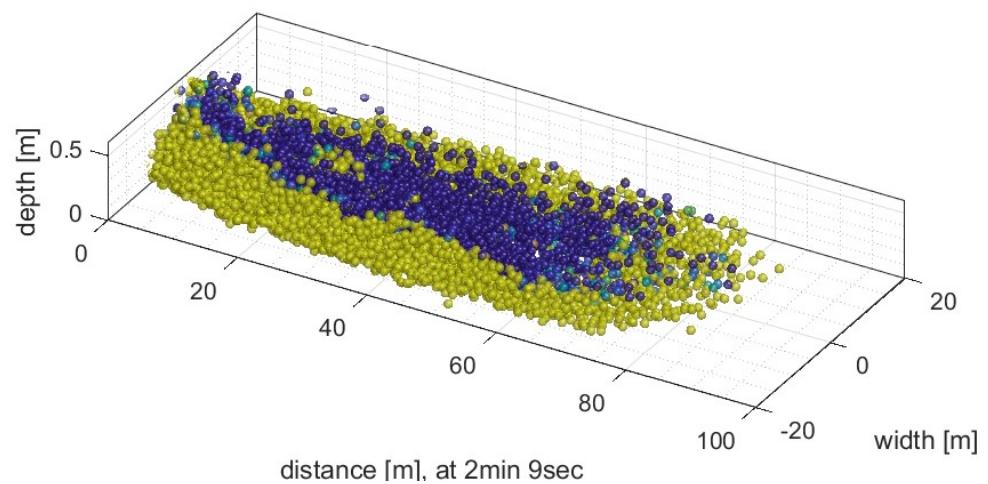
The given domain is discretized in 1m by 1m by 0.16m orthogonal elements. Therefore, the computational grid is composed of 90 000 cells. The material, given the current configuration and an initial particle density of  $3^3$  points per cell, is therefore discretized in 19 200 material points.

The bottom of the slope is described with a no-slip boundary condition (zero velocity Dirichlet BC). All other boundaries are reflective, no-flow boundaries which can be conceptualized as rigid walls. However, these boundaries are far enough from the runout that they will not come in contact with it and the lump of material will be able to develop without external influence.

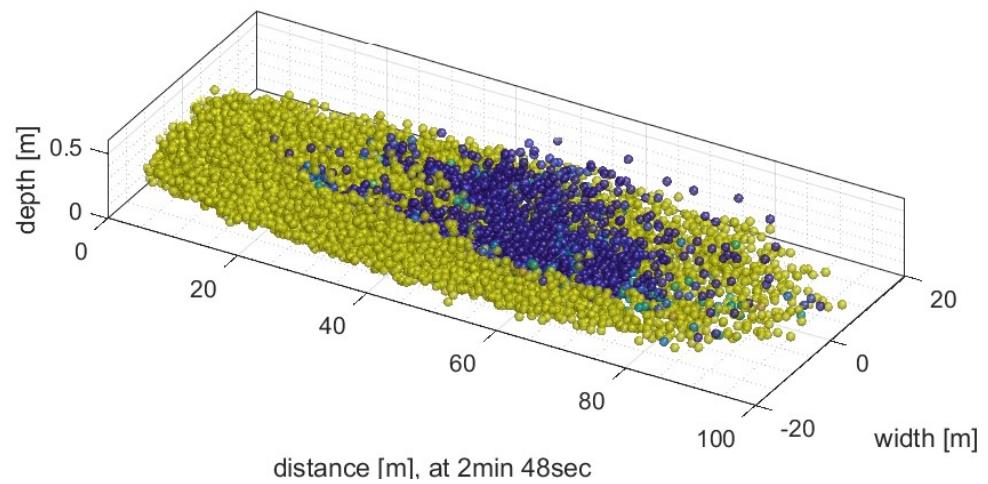
The following figures presents the evolution of the deposition lobe at different points in time for a 3D isometric view; a top view and a front view.



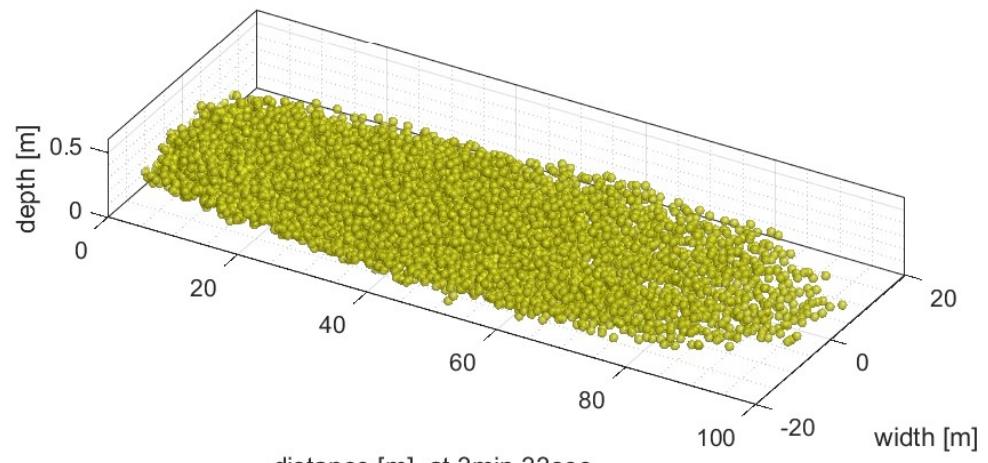
**Figure 60: Pipe deposition of thixotropic tailings for field scale conditions**



distance [m], at 2min 9sec

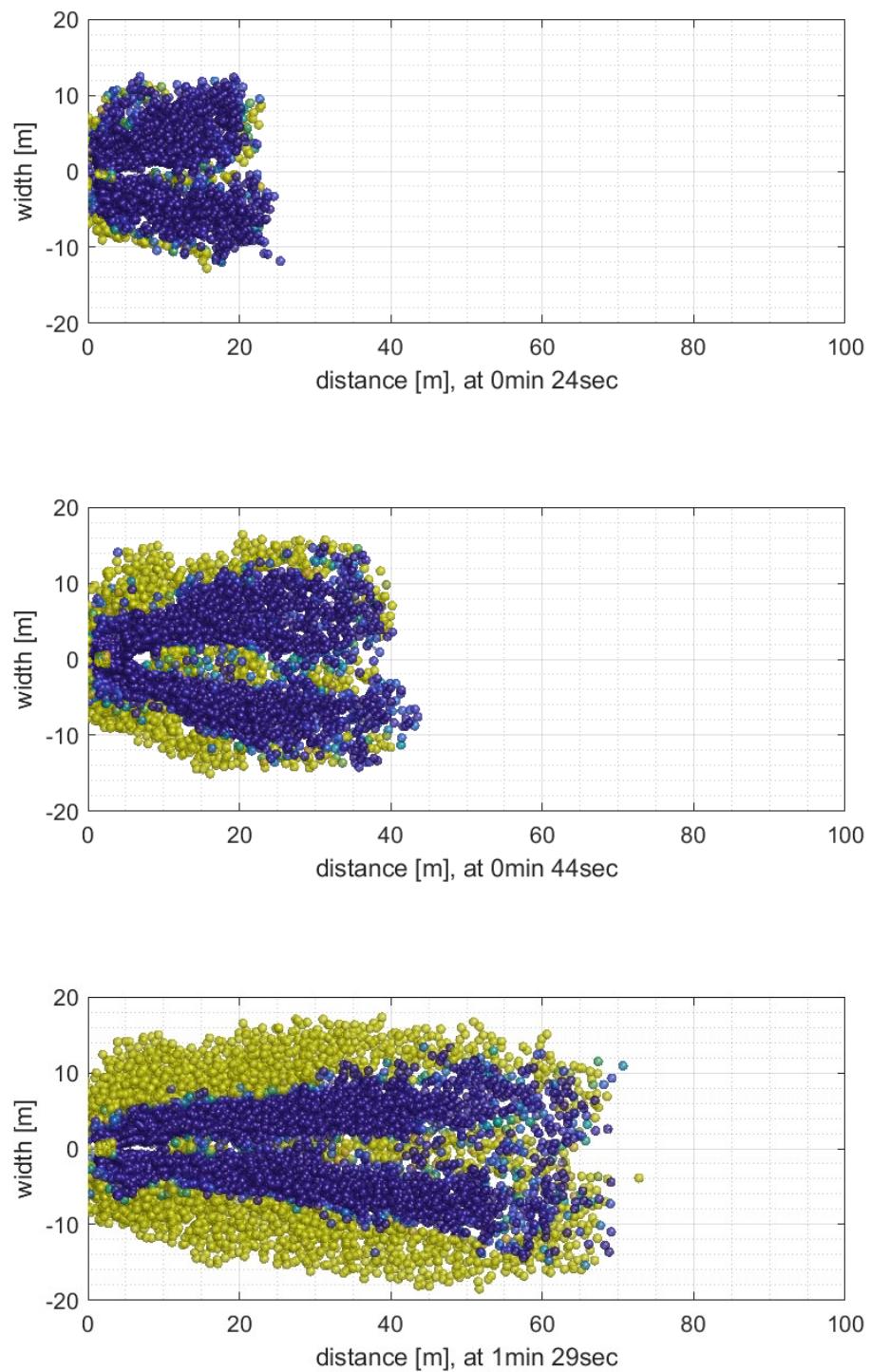


distance [m], at 2min 48sec

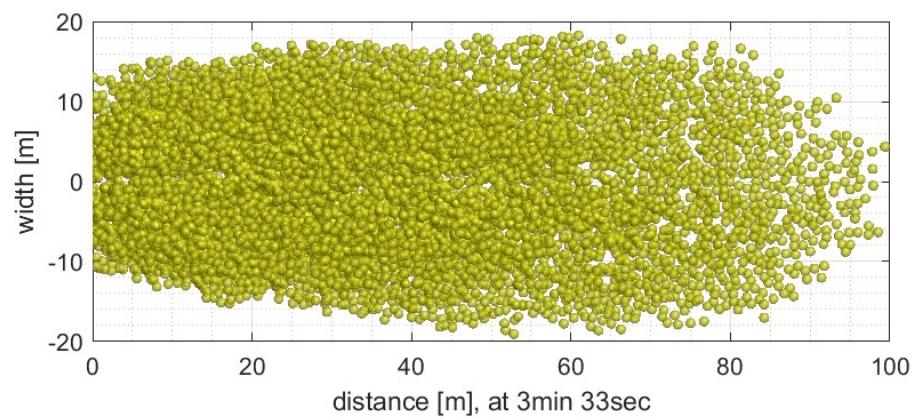
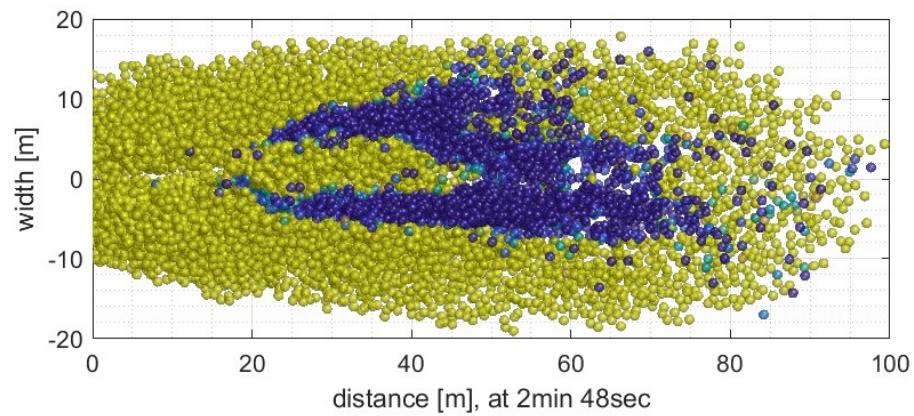
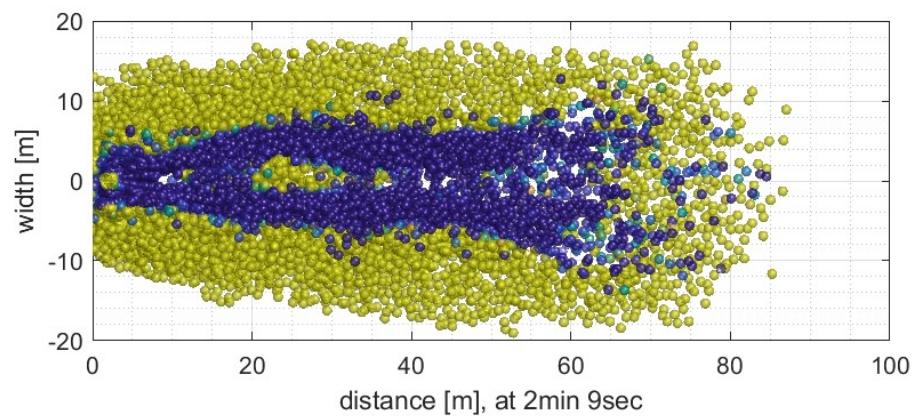


distance [m], at 3min 33sec

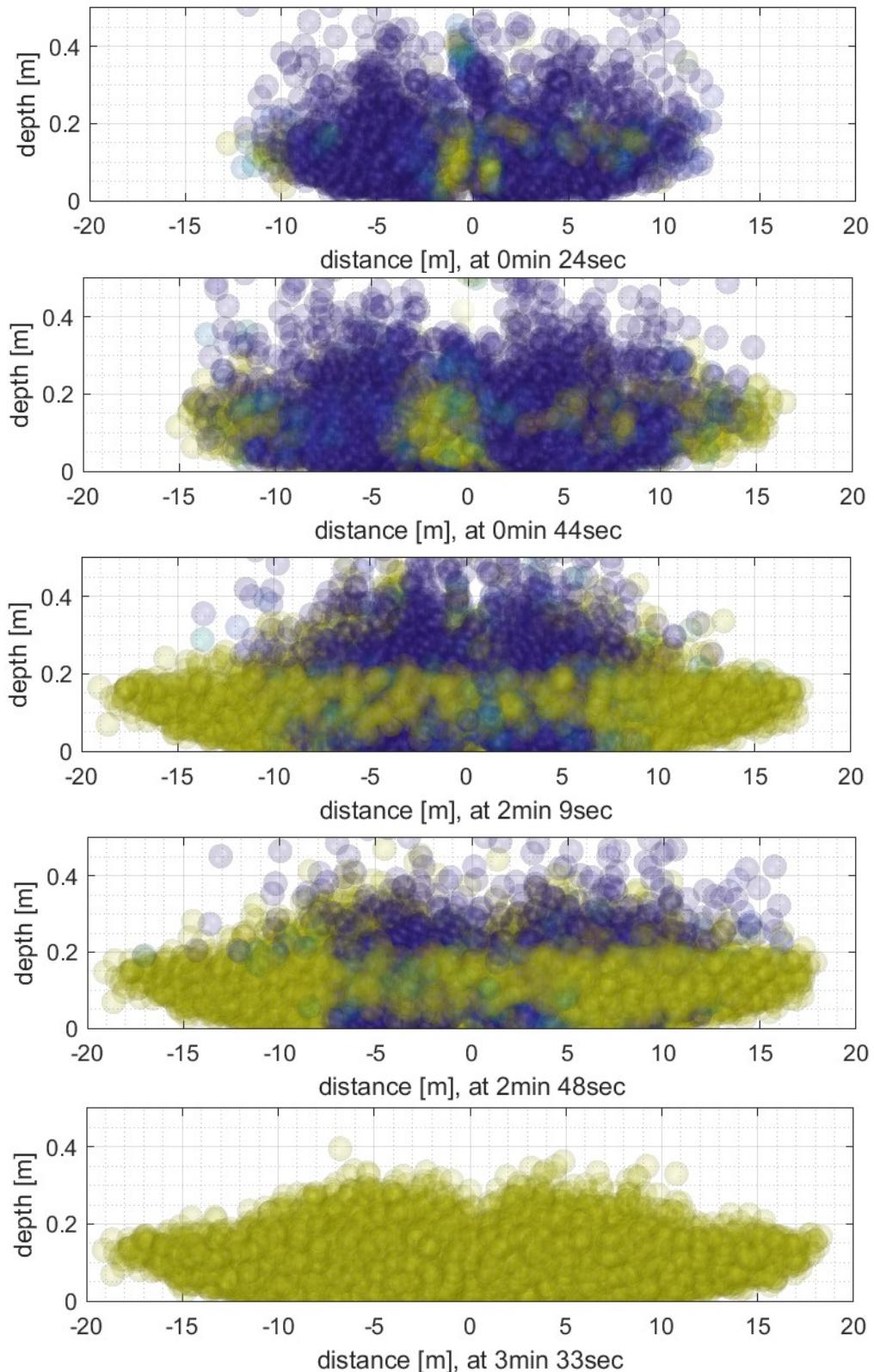
**Figure 59 (cont'd)**



**Figure 61 : Pipe deposition of thixotropic tailings for field scale conditions, plan view**



**Figure 60 (cont'd)**



**Figure 62: Pipe deposition of thixotropic tailings for field scale conditions, section view**

Initially, as seen in the first timestamp of the figures, the lobe develops in an elliptical shape. Since the material did not have any time to age and the shear stress profile is similar in all directions, no channels have formed yet.

However, after a few seconds, shear stresses in the exterior portions of the flow is subjected to small shear efforts which allows more rapid aging of the material, this isn't the case right in front of the deposition point as the continuous input of material with high velocity keeps the fresh tailings in a sheared state. Some differences in the expressed viscosity of the material can be seen due to the differences in aging speed.

After about 1 minute of flow time, one can see the channels being fully developed. The two channels are thought to be a side effect of the deposition height in the initial phase of the deposition, which allowed a lump of material to age quickly and therefore create a divergence point in the velocity vector field. The drastic difference in viscosity between the aged portion and the material inside the self-formed channel should be noted.

When deposition stops at the upstream, the channel progressively dries out. The last picture shows the final slump of fully aged material.

The splashing of particles that occur can create differences in kinetic energy between the modeled results and real-life cases. However, the differences should be minor. It is still possible to evaluate accurately the extent of the lobe at stoppage. The slope of the

material at rest is approximately 1.5% which is consistent with what is found in the literature (Fitton, 2006).

A rendering under video format of the simulation can be found at the following address : <https://youtu.be/uaTnPug9LVk>.

## 6: Conclusion

Several rheological models can be used to explain flow behavior of mine tailings. The Bingham model is the most widespread for its ease of calibration with experimental data and ability to represent observed behavior accurately, at laboratory scale. However, recent work has shown that thixotropy – the reversible gain in strength arising from microscopic bounds inside a material when it is left at rest – plays a significant role in the flow behavior of mine tailings. It also affects the expressed yield stress of a Bingham material as measurements under different shear conditions can make the yield stress vary by up to an order of magnitude.

In order to accurately represent large deformation problems in geoengineering, the material point method has proven itself to be a good candidate, overcoming some of the disadvantages of traditional Lagrangian or Eulerian discretization schemes. A thixotropic constitutive model has been implemented in Uintah, an open-source MPM framework. Indications were given on compiling and using the software as well as pre- and post-processing tools.

The robustness and validity of the implementation has been verified by replicating a series of previously published laboratory-scale experiments. Firstly, turbulent Newtonian flow was successfully modeled and the results were comparable with experiments as well as a FEM code with interface tracking. Analytical comparison with Poiseuille flow was also carried out for flow between two parallel plates.

Secondly, validation on deposition experiments of mine tailings has been done. Experimental data from Henriquez & Simms (2009) where both dam break and pipe deposition scenarios were studied was reproduced numerically. The final runout was reproduced accurately although some differences were found during the transient state.

The thixotropic constitutive model was validated with pour tests from Mizani (2016). The height and length of the runout were closely captured, but the total volume was larger in the MPM simulation. This is due to bleed water and self-weight consolidation that occurs over the 21 minutes span during which the tailings were deposited.

From a mathematical standpoint, the three constitutive models are related because they all rely on only calculating an elastic strain increment based on updated viscosity for each particle. For each material point, the viscosity is updated at every timestep following different rules, but no model requires splitting the deformation between elastic and plastic strain increments. This is highly efficient from a computing standpoint as it avoids the problem of return-mapping to the yield surface entirely.

Finally, a sensitivity analysis to physical and numerical parameters of the thixotropic constitutive model was done. No important variation of the results was found for a reasonable range of numerical parameters and the physical parameters had the expected effect on the runout distance.

With full validation of the code to reproduce rheological flow of tailings, different modeling scenarios at field scale were done. Scaling thixotropic and Bingham models at large scales provided different results. For identical runout at lab scale, thixotropic constitutive models had significantly longer runouts at large scales. This is most likely due to the time needed for the material to reach a non-deformable state which is considered instantaneous in the Bingham model under a certain shear level.

The thixotropic constitutive model was then used to model two scenarios at field scale in 3D conditions; one for dam break and one for pipe deposition.

A case study of the Wanji Aluminum Factory dam break was done and results were replicated in 3D. A sensitivity analysis of the various physical parameters also provided some additional insight in the modeling capabilities.

Finally, a continuous pipe deposition scenario was modeled to see how the thixotropic model would behave. Initially, a lobe develops but as aging takes over, some preferential flow channels appear. This behavior is seen on field scale and is the major hurdle is the adoption of thickened tailings deposition technology as it makes the thickness of lifts hard to predict.

The MPM is a promising tool for geotechnical engineering and has been used for static and dynamic analysis. It should also be considered for rheological modeling as it has proven to be reliable and capable of simulating complex scenarios. Since mine tailings oscillate between the two states, practicing engineers in the industry should consider it as a possible tool for runout modeling and other engineering needs.

## 7: Recommendations

The MPM was able to handle the large deformation and history-dependant constitutive model required for thixotropy. One of the main limitations for computational efficiency was the lack of implicit time integration. Explicitly, time steps are limited by the CFL condition. Since the constitutive model relies only on a linear elastic model, the fourth order stiffness tensor is easier to find due to its symmetry. Removing this constraint would allow significantly faster results for simulations in which the modeled time is in the range of 15 minutes and more in 3D with desktop computers.

Migration towards an MPM framework that is built with geotechnical or rheological problems in mind would also be better as treatment of complex topography and continuous particle creation for pipe deposition scenarios. One of the advantages of the Uintah framework is its proven scalability up to hundreds of thousands of CPU cores. However, parallelization in large clusters is less important with the current trend of using GPUs for scientific computations. Therefore, another framework providing different advantages might be more in line with future objectives.

Finally, tailings runout resulting both from deposition and dam breaks are a multi-physics process. Aged tailings are described accurately by geotechnical non-associated plasticity flow rules. The transition between geotechnical and rheological behavior should be investigated and the MPM is a good candidate to provide a single model that contains both sets of constitutive equations. Transition between the rheological and geotechnical states can be determined by second work order theory (Prime, 2014).

## Bibliography

- Aaron, J., Hungr, O., Stark, T. D., & Baghdady, A. K. (2017). Oso, Washington, Landslide of March 22, 2014: Dynamic Analysis. *Journal of Geotechnical and Geoenvironmental Engineering*, 143(9), 05017005.
- Andersen, S., & Andersen, L. (2010). Modelling of landslides with the material-point method. *Computational Geosciences*, 14(1), 137-147.
- Azam, S., & Li, Q. (2010). Tailings dam failures: a review of the last one hundred years. *Geotechnical news*, 28(4), 50-54.
- Babaoglu, Y. (2014). *Deposition Modelling of High Density Tailings Using Smoothed Particle Hydrodynamics* (Master's dissertation, Carleton University).
- Bandara, S., & Soga, K. (2015). Coupling of soil deformation and pore fluid flow using material point method. *Computers and geotechnics*, 63, 199-214.
- Bardenhagen, S. G., & Kober, E. M. (2004). The generalized interpolation material point method. *Computer Modeling in Engineering and Sciences*, 5(6), 477-496.
- Barnes, H. A. (1997). Thixotropy—a review. *Journal of Non-Newtonian fluid mechanics*, 70(1-2), 1-33.
- Barnes, H. A. (2007). The 'Yield stress myth?' paper -- 21 years on. *Applied Rheology*, 17(4), 43110-44250.
- Belytschko, T., Lui, W. K., Moran, B., & Elkhodary, K. (2013). *Nonlinear finite elements for continua and structures*. John Wiley & sons.
- Blight, G. E. (2009). *Geotechnical engineering for mine waste storage facilities*. CRC Press.

- Brackbill, J. U., Kothe, D. B., & Ruppel, H. M. (1988). FLIP: a low-dissipation, particle-in-cell method for fluid flow. *Computer Physics Communications*, 48(1), 25-38.
- Chen, Z., Hu, W., Shen, L., Xin, X., & Brannon, R. (2002). An evaluation of the MPM for simulating dynamic failure with damage diffusion. *Engineering Fracture Mechanics*, 69(17), 1873-1890.
- Coussot, P., Nguyen, Q. D., Huynh, H. T., & Bonn, D. (2002). Viscosity bifurcation in thixotropic, yielding fluids. *Journal of rheology*, 46(3), 573-589.
- Cruchaga, M. A., Celentano, D. J., & Tezduyar, T. E. (2007). Collapse of a liquid column: numerical simulation and experimental validation. *Computational Mechanics*, 39(4), 453-476.
- Cueto-Felgueroso, L., Colominas, I., Mosqueira, G., Navarrina, F., & Casteleiro, M. (2004). On the Galerkin formulation of the smoothed particle hydrodynamics method. *International Journal for Numerical Methods in Engineering*, 60(9), 1475-1512.
- Daliri, F., Kim, H., Simms, P., & Sivathayalan, S. (2014). Impact of desiccation on monotonic and cyclic shear strength of thickened gold tailings. *Journal of Geotechnical and Geoenvironmental Engineering*, 140(9), 04014048.
- De Stefano, M., Nadarajah, G., & Bleiker, D. (2019). Application of simplified/empirical framework to estimate runout from tailings dam failures. In *Sustainable and Safe Dams Around the World/Un monde de barrages durables et sécuritaires* (pp. 3410-3419). CRC Press.
- Domínguez, J. M., Crespo, A. J., Gómez-Gesteira, M., & Marongiu, J. C. (2011). Neighbour lists in smoothed particle hydrodynamics. *International Journal for Numerical Methods in Fluids*, 67(12), 2026-2042.

Fisseha, B., Bryan, R., & Simms, P. (2010). Evaporation, unsaturated flow, and salt accumulation in multilayer deposits of “paste” gold tailings. *Journal of geotechnical and geoenvironmental engineering*, 136(12), 1703-1712.

Fitton, T. G., Chryss, A. G., & Bhattacharya, S. N. (2006). Tailings beach slope prediction: a new rheological method. *International Journal of Mining, Reclamation and Environment*, 20(3), 181-202.

Fourie, A.B. and Gauw, S.K.Y. (2010) The validity of laboratory flume data for predicting beach slopes of thickened tailings deposits, in Proceedings 13th International Seminar on Paste and Thickened Tailings (Paste2010), R.J. Jewell and A.B. Fourie (eds), 3–6 May 2010, Toronto, Canada, Australian Centre for Geomechanics, Perth, pp. 241–254.

Gao, J., & Fourie, A. (2018). Studies on thickened tailings deposition in flume tests using the computational fluid dynamics (CFD) method. *Canadian Geotechnical Journal*, 56(2), 249-262.

Gingold, R. A., & Monaghan, J. J. (1977). Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly notices of the royal astronomical society*, 181(3), 375-389.

Gore, M. S., Gilbert, R. B., McMillan, I., & Isovitsch Parks, S. L. (2016). Geotechnical characterization of compacted Bauxite residue for use in Levees. In *Geo-Chicago 2016* (pp. 299-310).

Guilkey, J., Harman, T., Luitjens, J., Schmidt, J., Thornock, J., de St Germain, J. D., ... & Saad, T. (2009). Uintah user guide. In *Technical Report*. SCI Institute Technical Report.

Harlow, F. H. (1962). *The particle-in-cell method for numerical solution of problems in fluid dynamics* (No. LADC-5288). Los Alamos Scientific Lab., N. Mex..

- Henriquez, J., & Simms, P. (2009). Dynamic imaging and modelling of multilayer deposition of gold paste tailings. *Minerals Engineering*, 22(2), 128-139.
- Hewitt, D. R., & Balmforth, N. J. (2013). Thixotropic gravity currents. *Journal of Fluid Mechanics*, 727, 56-82.
- Hosseini, S. M., M. T. Manzari, and S. K. Hannani. "A fully explicit three-step SPH algorithm for simulation of non-Newtonian fluid flow." *International Journal of Numerical Methods for Heat & Fluid Flow* 17.7 (2007): 715-735.
- Hughes, T. J., Liu, W. K., & Zimmermann, T. K. (1981). Lagrangian-Eulerian finite element formulation for incompressible viscous flows. *Computer methods in applied mechanics and engineering*, 29(3), 329-349.
- Hungr, O., & McDougall, S. (2009). Two numerical models for landslide dynamic analysis. *Computers & Geosciences*, 35(5), 978-992.
- Hussin, H. Y., Quan Luna, B., Van Westen, C. J., Christen, M., Malet, J. P., & van Asch, T. W. (2012). Parameterization of a numerical 2-D debris flow model with entrainment: a case study of the Faucon catchment, Southern French Alps. *Natural hazards and earth system sciences*, 12(10), 3075-3090.
- ICOLD 2001. International Commission on Large Dams and the United Nations Environmental Programme (UNEP) Division of Technology, Industry and Economics (DTIE), “Tailings Dams Risks of Dangerous Occurrences Lessons Learned From Practical Experiences,” Bulletin 121, 2001
- Ito, M., & Azam, S. (2013). Large-strain consolidation modeling of mine waste tailings. *Environmental Systems Research*, 2(1), 7.

Jassim, I. (2013). *Formulation of a dynamic material point method (MPM) for geomechanical problems* (Doctoral dissertation, Ph. D. thesis. University of Stuttgart, Institute of Geotechnical Engineering).

Kossoff, D., Dubbin, W. E., Alfredsson, M., Edwards, S. J., Macklin, M. G., & Hudson-Edwards, K. A. (2014). Mine tailings dams: characteristics, failure, environmental impacts, and remediation. *Applied Geochemistry*, 51, 229-245.

Liu, K. F., & Mei, C. C. (1990). Approximate equations for the slow spreading of a thin sheet of Bingham plastic fluid. *Physics of Fluids A: Fluid Dynamics*, 2(1), 30-36.

Llano-Serna, M., Williams, D. J., & Ruest, M. (2017, January). Analysis of tailings dam-break and run-out. In *Fourth International Seminar on Tailings Management, Santiago, Chile* (pp. 12-14).

Lucy, L. B. (1977). A numerical approach to the testing of the fission hypothesis. *The astronomical journal*, 82, 1013-1024.

MAC. 2018. Facts & Figures 2018 Report. Mining Association of Canada.

Macdonald, J. R. (1969). Review of some experimental and analytical equations of state. *Reviews of Modern Physics*, 41(2), 316.

Martin, T. E., & McRoberts, E. C. (1999, January). Some considerations in the stability analysis of upstream tailings dams. In *Proceedings of the Sixth International Conference on Tailings and Mine Waste* (Vol. 99, pp. 287-302). Rotterdam, Netherlands: AA Balkema.

Mendes, P. R. S., & Dutra, E. S. (2004). Viscosity function for yield-stress liquids. *Applied Rheology*, 14(6), 296-302.

Meng, Q., Luitjens, J., & Berzins, M. (2010, November). Dynamic task scheduling for the uintah framework. In *2010 3rd Workshop on Many-Task Computing on Grids and Supercomputers* (pp. 1-10). IEEE.

Mergili, M., Jan-Thomas, F., Krenn, J., & Pudasaini, S. P. (2017). r. avaflow v1, an advanced open-source computational framework for the propagation and interaction of two-phase mass flows. *Geoscientific Model Development*, 10(2), 553.

Mewis, J., & Wagner, N. J. (2009). Current trends in suspension rheology. *Journal of Non-Newtonian Fluid Mechanics*, 157(3), 147-150.

Mitchell J.K. (1960). Fundamental Aspects of Thixotropy in Soils. *Journal of Soil Mechanics and Foundation Division, ASCE*, 86 (SM 3):19-52.

Mitchell, J.K., and Soga, K (2005). Fundamentals of Soil Behavior (3rd Edition) John Wiley & Sons.

Mizani, S., and Simms, P. (2016) .Rheology for deposition control of polymer amended oil sands tailings. Submitted to *Journal of Rheologica Acta*. Recommended for publishing pending revision, December 2016.

Mizani, S., (2016b) Experimental Study and Surface Deposition Modelling of Amended Oil Sands Tailings Products. Ph. D. Thesis Dissertation. Carleton University.

Mizani, S., Simms, P., & Wilson, W. (2017). Rheology for deposition control of polymer-amended oil sands tailings. *Rheologica Acta*, 56(7-8), 623-634.

Nguyen, Q. D., & Boger, D. V. (1998). Application of rheology to solving tailings disposal  
Nguyen, Q.D., Akroyd, T., De Kee, D.C., and Zhu, L. (2006). Yield stress measurements in suspensions: an inter-laboratory study. *Korea-Australia Rheology Journal*. 18 (1): 15–24.

- Nguyen, V. P., & University, C. (2014). Material point method: basics and applications
- Olson, S. M., & Stark, T. D. (2002). Liquefied strength ratio from liquefaction flow failure case histories. *Canadian Geotechnical Journal*, 39(3), 629-647.
- Onate, E., Idelsohn, S. R., Celigueta, M. A., & Rossi, R. (2008). Advances in the particle finite element method for the analysis of fluid–multibody interaction and bed erosion in free surface flows. *Computer methods in applied mechanics and engineering*, 197(19-20), 1777-1800.
- Orduz Rojas, C. M., Barros Pereira, D., & Bruno Dias, J. (2019). The conflicts of (Neo) extractivism: Mines, tailings dams, and protected areas in the metropolitan area of Belo Horizonte, Brazil. *Cybergeo: European Journal of Geography*.
- Papanastasiou, T., Georgiou, G., & Alexandrou, A. N. (1999). *Viscous fluid flow*. CRC press.
- Parker, S. G., Guilkey, J., & Harman, T. (2006). A component-based parallel infrastructure for the simulation of fluid–structure interaction. *Engineering with Computers*, 22(3-4), 277-292.
- Phuong, N. T. V., Van Tol, A. F., Elkadi, A. S. K., & Rohe, A. (2016). Numerical investigation of pile installation effects in sand using material point method. *Computers and Geotechnics*, 73, 58-71.
- Qiu, Yunxin, and D. C. Sego. "Laboratory properties of mine tailings." *Canadian Geotechnical Journal* 38.1 (2001): 183-190.
- Rafael, H. M. M., & Romanel, C. (2017, July). Static Liquefaction Analysis of the Limonar Tailings Dam in Peru. In *International Congress and Exhibition" Sustainable Civil Infrastructures: Innovative Infrastructure Geotechnology"* (pp. 41-56). Springer, Cham.

Rico, M., Benito, G., Salgueiro, A. R., Díez-Herrero, A., & Pereira, H. G. (2008). Reported tailings dam failures: a review of the European incidents in the worldwide context. *Journal of hazardous materials*, 152(2), 846-852.

Robertson, P. K. (2017, July). Evaluation of Flow Liquefaction: influence of high stresses. In *Proceedings of the 3rd International Conference on Performance Based Design (PBD-III)*.

Sadeghirad, A., Brannon, R. M., & Burghardt, J. (2011). A convected particle domain interpolation technique to extend applicability of the material point method for problems involving massive deformations. *International Journal for numerical methods in Engineering*, 86(12), 1435-1456.

Simms, P., Williams, M. P. A., Fitton, T. G., & McPhail, G. (2011, April). Beaching angles and evolution of stack geometry for thickened tailings-a review. In *Invited paper.] In Proceedings of the 14th International Conference on Paste and Thickened Tailings, Edited by R. Jewell and A. Foruie, Perth, Australia* (pp. 171-190)

Simms, P. (2016). 2013 Colloquium of the Canadian Geotechnical Society: Geotechnical and geoenvironmental behaviour of high-density tailings. *Canadian Geotechnical Journal*, 54(4), 455-468.

Simms, Paul & Dunmola, A. & Fisseha, Bereket. (2017). Generic Predictions of Drying Time in Surface Deposited Thickened Tailings in a "wet" Climate.

Soga, K., Alonso, E., Yerro, A., Kumar, K., & Bandara, S. (2015). Trends in large-deformation analysis of landslide mass movements with particular emphasis on the material point method. *Géotechnique*, 66(3), 248-273.

Stark, T. D., Choi, H., & McCone, S. (2005). Drained shear strength parameters for analysis of landslides. *Journal of Geotechnical and Geoenvironmental Engineering*, 131(5), 575-588.

Sulsky, D., Chen, Z., & Schreyer, H. L. (1994). A particle method for history-dependent materials. *Computer methods in applied mechanics and engineering*, 118(1-2), 179-196

Vick SG. 1983. Planing, Design and Analysis of Tailings Dams. New York: John Wiley & Sons, Inc.

Wieckowski, Z. (2004, August). The material point method in soil mechanics problems. In *The XXI International Congress of Theoretical and Applied Mechanics* (pp. 3-4).

WISE 2015. World Information Service on Energy Uranium Project, Chronology of Major Tailings Dam Failures, from 1960, updated 18 May 2015, <http://www.wise-uranium.org/mdaf.html>

Yerro Colom, A., Alonso Pérez de Agreda, E., & Pinyol Puigmartí, N. M. (2013). The Material Point Method: a promising computational tool in geotechnics. In *Challenges and Innovations in Geotechnics* (pp. 853-856).

Yu, J., & Turk, G. (2013). Reconstructing surfaces of particle-based fluids using anisotropic kernels. *ACM Transactions on Graphics (TOG)*, 32(1), 5.

Zienkiewicz, O. C., Taylor, R. L., & Zhu, J. Z. (2005). *The finite element method: its basis and fundamentals*. Elsevier.

## Annex 1 – Constitutive Model Class

### A. Thixotropic.h – Header file

```
/*
 * The MIT License
 *
 * Copyright (c) 1997-2017 The University of Utah
 *
 * Permission is hereby granted, free of charge, to any person
 * obtaining a copy
 * of this software and associated documentation files (the
 * "Software"), to
 * deal in the Software without restriction, including without
 * limitation the
 * rights to use, copy, modify, merge, publish, distribute,
 * sublicense, and/or
 * sell copies of the Software, and to permit persons to whom the
 * Software is
 * furnished to do so, subject to the following conditions:
 *
 * The above copyright notice and this permission notice shall be
 * included in
 * all copies or substantial portions of the Software.
 *
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
 * EXPRESS OR
 * IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
 * MERCHANTABILITY,
 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT
 * SHALL THE
 * AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR
 * OTHER
 * LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
 * ARISING
 * FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
 * OTHER DEALINGS
 * IN THE SOFTWARE. */

#ifndef __THIXOTROPIC_CONSTITUTIVE_MODEL_H__
#define __THIXOTROPIC_CONSTITUTIVE_MODEL_H__


#include <cmath>
#include "ConstitutiveModel.h"
#include <Core/Math/Matrix3.h>
#include <vector>
#include <Core/Disclosure/TypeDescription.h>

namespace Uintah {
    class Thixotropic : public ConstitutiveModel {
```

```

public:

    // Create datatype for storing model parameters
    struct CMDData {
        //THIXOTROPIC PARAMETERS
        double d_Bulk;           //bulk modulus
        double d_Gamma;          //for EOS, use 7
        double d_MinViscosity;   //fully sheared viscosity
        double d_Alpha;          //empirical parameter alpha
        double d_TimeParam;      //'T' time parameter for
        structuration
        double d_LambdaPar;      // structure at time t=0, constant
    for now
        double d_ExponentN;     // n exponent
        double d_ViscoCap;       // maximum allowed viscosity
    };
}

protected:

    CMDData d_initialData;
    bool d_useModifiedEOS;
    int d_8or27;

private:
    // Prevent copying of this class
    // copy constructor
    Thixotropic& operator=(const Thixotropic &cm);

public:
    // constructors
    Thixotropic(ProblemSpecP& ps, MPMFlags* flag);

    // destructor
    virtual ~Thixotropic();

    virtual void outputProblemSpec(ProblemSpecP& ps, bool
output_cm_tag = true);

    // clone
    Thixotropic* clone();

    virtual void addInitialComputesAndRequires(Task* task,
                                                const MPMMaterial*
matl,
                                                const PatchSet*
patches) const;

    // compute stable timestep for this patch
    virtual void computeStableTimestep(const Patch* patch,
                                         const MPMMaterial* matl,
                                         DataWarehouse* new_dw);

    // compute stress at each particle in the patch
    virtual void computeStressTensor(const PatchSubset* patches,

```

```

        const MPMMaterial* matl,
        DataWarehouse* old_dw,
        DataWarehouse* new_dw);

    // carry forward CM data for RigidMPM
    virtual void carryForward(const PatchSubset* patches,
                              const MPMMaterial* matl,
                              DataWarehouse* old_dw,
                              DataWarehouse* new_dw);

    // initialize each particle's constitutive model data
    virtual void initializeCMData(const Patch* patch,
                                  const MPMMaterial* matl,
                                  DataWarehouse* new_dw);

    virtual void addComputesAndRequires(Task* task,
                                        const MPMMaterial* matl,
                                        const PatchSet* patches)
const;

    virtual void addComputesAndRequires(Task* task,
                                        const MPMMaterial* matl,
                                        const PatchSet* patches,
                                        const bool recursion) const;

    virtual double computeRhoMicroCM(double pressure,
                                      const double p_ref,
                                      const MPMMaterial* matl,
                                      double temperature,
                                      double rho_guess);

    virtual void computePressEOSCM(double rho_m, double& press_eos,
                                   double p_ref,
                                   double& dp_drho, double& ss_new,
                                   const MPMMaterial* matl,
                                   double temperature);

    virtual double getCompressibility();

    virtual void addParticleState(std::vector<const VarLabel*>&
from,
                                std::vector<const VarLabel*>& to);

    void initializeLocalMPMLabels();

    //THIXOTROPY-SPECIFIC LABELS
    const VarLabel *p_structureLabel;
    const VarLabel *p_structureLabel_preReloc;
    const VarLabel *p_viscoLabel;
    const VarLabel *p_viscoLabel_preReloc;
    const VarLabel *p_existsLabel;
    const VarLabel *p_existsLabel_preReloc;

};

} // End namespace Uintah

```

```
#endif // __THIXOTROPIC_CONSTITUTIVE_MODEL_H__
```

## B. Thixotropic.cc

```
/*
 * The MIT License
 *
 * Copyright (c) 1997-2017 The University of Utah
 *
 * Permission is hereby granted, free of charge, to any person
 * obtaining a copy
 * of this software and associated documentation files (the
 * "Software"), to
 * deal in the Software without restriction, including without
 * limitation the
 * rights to use, copy, modify, merge, publish, distribute,
 * sublicense, and/or
 * sell copies of the Software, and to permit persons to whom the
 * Software is
 * furnished to do so, subject to the following conditions:
 *
 * The above copyright notice and this permission notice shall be
 * included in
 * all copies or substantial portions of the Software.
 *
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
 * EXPRESS OR
 * IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
 * MERCHANTABILITY,
 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT
 * SHALL THE
 * AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR
 * OTHER
 * LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
 * ARISING
 * FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
 * OTHER DEALINGS
 * IN THE SOFTWARE. */

#include <CCA/Components/MPM/ConstitutiveModel/Thixotropic.h>
#include <Core/Grid/Patch.h>
#include <CCA/Ports/DataWarehouse.h>
#include <Core/Grid/Variables/ParticleVariable.h>
#include <Core/Grid/Task.h>
#include <Core/Grid/Level.h>
#include <Core/Grid/Variables/VarLabel.h>
#include <Core/Grid/Variables/VarTypes.h>
#include <Core/Labels/MPMLabel.h>
#include <Core/Math/Matrix3.h>
#include <CCA/Components/MPM/ConstitutiveModel/MPMMaterial.h>
#include <Core/ProblemSpec/ProblemSpec.h>
#include <Core/Math/MinMax.h>
```

```

#include <Core/Malloc/Allocator.h>
#include <iostream>

using namespace std;
using namespace Uintah;

Thixotropic::Thixotropic(ProblemSpecP& ps, MPMFlags* Mflag)
    : ConstitutiveModel(Mflag)
{
    d_useModifiedEOS = false;

    //make sure all required inputs are specified
    ps->require("bulk_modulus",           d_initialData.d_Bulk);
    ps->require("gamma",                 d_initialData.d_Gamma);
    ps->require("minimum_viscosity",     d_initialData.d_MinViscosity);
    ps->require("alpha_parameter",       d_initialData.d_Alpha);
    ps->require("t_time_parameter",      d_initialData.d_TimeParam);
    ps->require("initial_structure",    d_initialData.d_LambdaPar);
    ps->require("exponent_n",           d_initialData.d_ExponentN);
    ps->require("max_viscosity",         d_initialData.d_ViscoCap);

    //create labels specific for thixotropy
    initializeLocalMPMLabels();
}

Thixotropic::~Thixotropic()
{
    //destroys custom labels
    VarLabel::destroy(p_structureLabel);
    VarLabel::destroy(p_structureLabel_preReloc);
    VarLabel::destroy(p_viscoLabel);
    VarLabel::destroy(p_viscoLabel_preReloc);

    //FOR PIPE DEPOSITION
    VarLabel::destroy(p_existsLabel);
    VarLabel::destroy(p_existsLabel_preReloc);
}

void Thixotropic::outputProblemSpec(ProblemSpecP& ps, bool
output_cm_tag)
{
    ProblemSpecP cm_ps = ps;
    if (output_cm_tag) {
        cm_ps = ps->appendChild("constitutive_model");
        cm_ps->setAttribute("type", "Thixotropic");
    }

    cm_ps->appendElement("bulk_modulus",           d_initialData.d_Bulk);
    cm_ps->appendElement("gamma",                 d_initialData.d_Gamma);
    cm_ps->appendElement("minimum_viscosity",     d_initialData.d_MinViscosity);
    cm_ps->appendElement("alpha_parameter",       d_initialData.d_Alpha);
}

```

```

        cm_ps->appendElement("t_time_parameter",
d_initialData.d_TimeParam);
        cm_ps->appendElement("initial_structure",
d_initialData.d_LambdaPar);
        cm_ps->appendElement("exponent_n",
d_initialData.d_ExponentN);
        cm_ps->appendElement("max_viscosity",
d_initialData.d_ViscoCap);
    }

Thixotropic* Thixotropic::clone()
{
    return scinew Thixotropic(*this);
}

void Thixotropic::initializeLocalMPMLabels()
{
    //labels for variables specific to thixotropic case
    p_structureLabel = VarLabel::create("p.structure",
ParticleVariable<double>::getTypeDescription() );
    p_structureLabel_preReloc = VarLabel::create("p.structure+",
ParticleVariable<double>::getTypeDescription() );
    p_viscoLabel = VarLabel::create("p.viscosity",
ParticleVariable<double>::getTypeDescription() );
    p_viscoLabel_preReloc = VarLabel::create("p.viscosity+",

    ParticleVariable<double>::getTypeDescription() );
    ParticleVariable<double>::getTypeDescription() );
    ParticleVariable<double>::getTypeDescription() );
    ParticleVariable<double>::getTypeDescription() );
    ParticleVariable<double>::getTypeDescription() );

    //FOR PIPE DEPOSITION
    p_existsLabel = VarLabel::create("p.exists",
ParticleVariable<int>::getTypeDescription()
);
    p_existsLabel_preReloc = VarLabel::create("p.exists+",
ParticleVariable<int>::getTypeDescription()
);
}

void Thixotropic::initializeCMData(const Patch* patch,
const MPMMaterial* matl,
DataWarehouse* new_dw)
{
    // Initialize the variables shared by all constitutive models
    // This method is defined in the ConstitutiveModel base class.
    initSharedDataForExplicit(patch, matl, new_dw);

    //Initialize structure parameters
    ParticleSubset* pset = new_dw->getParticleSubset(matl-
>getDWIndex(), patch);
    double bulk = d_initialData.d_Bulk;
    double gamma = d_initialData.d_Gamma;
}

```

```

double minvis = d_initialData.d_MinViscosity;
double alpha = d_initialData.d_Alpha;
double tParam = d_initialData.d_TimeParam;
double expN = d_initialData.d_ExponentN;
ParticleVariable<double> p_structure;
ParticleVariable<double> p_viscosity;
ParticleVariable<int> p_exists;
double rho_orig = matl->getInitialDensity();

//calculate viscosity at time t=0
double lambdaini = d_initialData.d_LambdaPar;
double viscini = minvis * (1.0 + pow(lambdaini, expN));
int state = 0;

new_dw->allocateAndPut(p_structure, p_structureLabel,           pset);
new_dw->allocateAndPut(p_viscosity, p_viscoLabel,             pset);
new_dw->allocateAndPut(p_exists,   p_existsLabel,              pset);

ParticleSubset::iterator iter = pset->begin();
for(;iter != pset->end(); iter++){
    p_structure[*iter] = lambdaini;
    p_viscosity[*iter] = viscini;
    p_exists[*iter] = state;
}

computeStableTimestep(patch, matl, new_dw);
}

void Thixotropic::computeStableTimestep(const Patch* patch,
                                         const MPMMaterial* matl,
                                         DataWarehouse* new_dw)
{
    // This is only called for the initial timestep - all other
    timesteps
    // are computed as a side-effect of computeStressTensor
    Vector dx = patch->dCell();
    int dwi = matl->getDWIndex();
    // Retrieve the array of constitutive parameters
    ParticleSubset* pset = new_dw->getParticleSubset(dwi, patch);
    constParticleVariable<double> pmass, pvolume;
    constParticleVariable<Vector> pvelocity;

    new_dw->get(pmass,      lb->pMassLabel,      pset);
    new_dw->get(pvolume,    lb->pVolumeLabel,    pset);
    new_dw->get(pvelocity, lb->pVelocityLabel, pset);

    double c_dil = 0.0;
    Vector WaveSpeed(1.e-12,1.e-12,1.e-12);

    double bulk = d_initialData.d_Bulk;
    for(ParticleSubset::iterator iter = pset->begin();iter != pset-
>end();iter++){
        particleIndex idx = *iter;
        // Compute wave speed at each particle, store the maximum
        c_dil = sqrt((bulk)*pvolume[idx]/pmass[idx]);
    }
}

```

```

WaveSpeed=Vector (Max(c_dil+fabs(pvelocity[idx].x()),WaveSpeed.x()),
Max(c_dil+fabs(pvelocity[idx].y()),WaveSpeed.y()),
Max(c_dil+fabs(pvelocity[idx].z()),WaveSpeed.z()));
}
WaveSpeed = dx/WaveSpeed;
double delt_new = WaveSpeed.minComponent();
new_dw->put(delt_vartype(delt_new), lb->deltLabel, patch-
>getLevel());
}

void Thixotropic::computeStressTensor(const PatchSubset* patches,
                                      const MPMMaterial* matl,
                                      DataWarehouse* old_dw,
                                      DataWarehouse* new_dw)
{
    for(int pp=0;pp<patches->size();pp++) {
        const Patch* patch = patches->get(pp);
        Matrix3 tensorL,Shear;
        double J, p,se=0.;
        double c_dil=0.0;
        Vector WaveSpeed(1.e-12,1.e-12,1.e-12);
        double onethird = (1.0/3.0);
        Matrix3 Identity; Identity.Identity();

        Vector dx = patch->dCell();

        int dwi = matl->getDWIndex();
        ParticleSubset* pset = old_dw->getParticleSubset(dwi, patch);
        //old data containers
        constParticleVariable<Matrix3> defGrad_new;
        constParticleVariable<Matrix3> defGrad;
        constParticleVariable<Matrix3> velGrad;
        ParticleVariable<Matrix3> pStress_new;
        constParticleVariable<double> pVolume;
        constParticleVariable<Vector> pVelocity;
        constParticleVariable<int> pLocalized;
        ParticleVariable<int> pLocalized_new;
        ParticleVariable<double> pdTdt,p_q;
        //Create structure variables + new data containers
        constParticleVariable<double> pStructure;
        ParticleVariable<double> pStructure_new;
        constParticleVariable<double> pViscosity;
        ParticleVariable<double> pViscosity_new;
        constParticleVariable<int> pExists;
        ParticleVariable<int> pExists_new;
        constParticleVariable<Point> pPos;

        delt_vartype delt;
        old_dw->get(delt, lb->deltLabel, getLevel(patches));
        //get data
    }
}

```

```

        old_dw->get(pVelocity,           lb->pVelocityLabel,
pset);          lb->pDeformationMeasureLabel,
        old_dw->get(defGrad,            pset);
        old_dw->get(pLocalized,         lb->pLocalizedMPMLabel,
pset);          p_structureLabel,
        old_dw->get(pStructure,         p_viscoLabel,
pset);          p_existsLabel,
        old_dw->get(pViscosity,         pset);
        old_dw->get(pExists,           lb->pXLabel,
pset);          pset);

        //allocations
        new_dw->allocateAndPut(pExists_new,
                                p_existsLabel_preReloc,
pset);          lb->pStressLabel_preReloc,
        new_dw->allocateAndPut(pStress_new,   pset);
        new_dw->allocateAndPut(pdTdt,       lb->pdTdtLabel,
pset);          pset);
        new_dw->allocateAndPut(p_q,         lb->p_qLabel_preReloc,
pset);          lb->pLocalizedMPMLabel_preReloc,
        new_dw->allocateAndPut(pLocalized_new,
                                pset);
        new_dw->allocateAndPut(pViscosity_new,
                                p_viscoLabel_preReloc,
pset);          lb->pStructure_label,
        new_dw->allocateAndPut(pStructure_new,
                                p_structureLabel_preReloc,
pset);          lb->pDeformationMeasureLabel_preReloc,
        new_dw->get(defGrad_new,
                    lb->pVolumeLabel_preReloc,
pset);          pset);
        new_dw->get(pVolume,             lb->pVelGradLabel_preReloc,
pset);          pset);

        // Get the particle IDs, useful in case a simulation goes belly
up
        constParticleVariable<long64> pParticleID;
        old_dw->get(pParticleID, lb->pParticleIDLabel, pset);

        //set constants
        double bulk = d_initialData.d_Bulk;
        double gamma = d_initialData.d_Gamma;
        double minvis = d_initialData.d_MinViscosity;
        double alpha = d_initialData.d_Alpha;
        double tParam = d_initialData.d_TimeParam;
        double expN = d_initialData.d_ExponentN;

```

```

    double viscmax = d_initialData.d_ViscoCap;

    double rho_orig = matl->getInitialDensity();

    for(ParticleSubset::iterator iter = pset->begin(); iter!=pset->end(); iter++){
        particleIndex idx = *iter;

        // Assign zero internal heating by default - modify if necessary.
        pdTdt[idx] = 0.0;

        // Carry forward the pLocalized tag for now, alter below
        pLocalized_new[idx] = pLocalized[idx];

        J = defGrad_new[idx].Determinant();

        if(!(J > 0.) || J > 1.e50){
            cerr << "**ERROR** Negative (or huge) Jacobian of
deformation gradient."
                << " Deleting particle in Thixotropic model" <<
pParticleID[idx] << endl;
            cerr << "l = " << velGrad[idx] << endl;
            cerr << "F_new = " << defGrad_new[idx] << endl;
            cerr << "J = " << J << endl;
            cerr << "DWI = " << matl->getDWIndex() << endl;
            pLocalized_new[idx]=-999;

            J=1;
        }

        //check if particle exists (in the positive part of x
        coordinate system)
        int state = pExists[idx];
        double posx = pPos[idx].x();
        if(state==0){
            if(posx<0){
                //start aging process
                pExists_new[idx] = 1;
            }
        }

        // Calculate rate of deformation D, and deviatoric rate
        DPrime,
        Matrix3 D = (velGrad[idx] + velGrad[idx].Transpose())*0.5;
        Matrix3 DPrime = D - Identity*onethird*D.Trace();

        //get the second invariant of the deviatoric part
        double SecInvar = DPrime(0,0)*DPrime(1,1) +
                        DPrime(1,1)*DPrime(2,2) +
                        DPrime(2,2)*DPrime(0,0) -
                        DPrime(0,1)*DPrime(1,0) -
                        DPrime(1,2)*DPrime(2,1) -
                        DPrime(0,2)*DPrime(2,0);
    }
}

```

```

SecInvar = abs(0.5*SecInvar);

//Compute thixotropic viscosity
double delLambda = delt * ((1.0/tParam) -
alpha*SecInvar*pStructure[idx]);

//if particle is allowed to age update structure parameter
if(state==1){
    pStructure_new[idx] = pStructure[idx] + delLambda;
}else{
    pStructure_new[idx] = pStructure[idx];
}

double viscosity = minvis * (1.0 +
pow(pStructure_new[idx],expN));
if(viscosity >= viscmax){
    viscosity = viscmax;
}

pViscosity_new[idx] = viscosity;

// Get the deformed volume and current density
double rho_cur = rho_orig/J;

// Viscous part of the stress
Shear = DPrime*(2.*pViscosity_new[idx]);

// get the hydrostatic part of the stress
double cconst = 0.0894;

p = cconst*bulk*(exp((1 / cconst)*(1 - J)) - 1);

// compute the total stress (volumetric + deviatoric)
pStress_new[idx] = (Identity*(-p) + Shear);

double jtotheminusgamma = pow(J, -gamma);
Vector pVelocity_idx = pVelocity[idx];
c_dil = sqrt((gamma*jtotheminusgamma*bulk)/rho_cur);

WaveSpeed=Vector (Max(c_dil+fabs(pVelocity_idx.x()),WaveSpeed.x()),
Max(c_dil+fabs(pVelocity_idx.y()),WaveSpeed.y()),
Max(c_dil+fabs(pVelocity_idx.z()),WaveSpeed.z()));

if(!(pStress_new[idx].Norm()>0) &&
!(pStress_new[idx].Norm()<1e15)){
    cout << "F_new = " << defGrad_new[idx] << endl;
    cout << "pressure = " << p << endl;
    cout << "shear = " << Shear << endl;
    cout << "pstress = " << pStress_new[idx] << endl;
    cout << "J = " << J << endl;
}

```

```

        // Compute artificial viscosity term
        if (flag->d_artificial_viscosity) {
            double dx_ave = (dx.x() + dx.y() + dx.z()) / 3.0;
            double c_bulk = sqrt(bulk/rho_cur);
            Matrix3 D=(velGrad[idx] + velGrad[idx].Transpose())*0.5;
            p_q[idx] = artificialBulkViscosity(D.Trace(), c_bulk,
rho_cur, dx_ave);
        } else {
            p_q[idx] = 0.0;
        }
    } // end loop over particles

    WaveSpeed = dx/WaveSpeed;
    double delT_new = WaveSpeed.minComponent();
    new_dw->put(delt_varType(delT_new), lb->delTLabel, patch-
>getLevel()));

    if (flag->d_reductionVars->accStrainEnergy ||
        flag->d_reductionVars->strainEnergy) {
        new_dw->put(sum_varType(se), lb->StrainEnergyLabel);
    }
}
}

void Thixotropic::addParticleState(std::vector<const VarLabel*>&
from,
                                    std::vector<const VarLabel*>& to)
{
    from.push_back(p_structureLabel);
    to.push_back(p_structureLabel_preReloc);

    from.push_back(p_viscoLabel);
    to.push_back(p_viscoLabel_preReloc);

    from.push_back(p_existsLabel);
    to.push_back(p_existsLabel_preReloc);
}

void Thixotropic::addInitialComputesAndRequires(Task* task,
                                                const MPMMaterial* matl,
                                                const PatchSet* patch)
const
{
}

void Thixotropic::carryForward(const PatchSubset* patches,
                               const MPMMaterial* matl,
                               DataWarehouse* old_dw,
                               DataWarehouse* new_dw)
{
    for(int p=0;p<patches->size();p++){
        const Patch* patch = patches->get(p);
        int dwi = matl->getDWIndex();
        ParticleSubset* pset = old_dw->getParticleSubset(dwi, patch);
    }
}

```

```

// Carry forward the data common to all constitutive models
// when using RigidMPM.
// This method is defined in the ConstitutiveModel base class.
carryForwardSharedData(pset, old_dw, new_dw, matl);

// Carry forward the data local to this constitutive model
new_dw->put(delt_vartype(1.e10), lb->delTLabel, patch-
>getLevel()));

ParticleVariable<double> p_structure_new;
const ParticleVariable<double> p_structure;
old_dw->get(p_structure, p_structureLabel,
pset);
new_dw->allocateAndPut(p_structure_new,
p_structureLabel_preReloc, pset);

ParticleVariable<double> p_viscosity_new;
const ParticleVariable<double> p_viscosity;
old_dw->get(p_viscosity, p_viscoLabel, pset);
new_dw->allocateAndPut(p_viscosity_new, p_viscoLabel_preReloc,
pset);

ParticleVariable<int> p_exists_new;
const ParticleVariable<double> p_exists;
old_dw->get(p_exists, p_existsLabel, pset);
new_dw->allocateAndPut(p_exists_new, p_existsLabel_preReloc,
pset);

for(ParticleSubset::iterator iter = pset->begin();
     iter != pset->end(); iter++) {
    particleIndex idx = *iter;
    p_structure_new[idx] = p_structure[idx];
    p_viscosity_new[idx] = p_viscosity[idx];
    p_exists_new[idx] = p_exists[idx];
}

if (flag->d_reductionVars->accStrainEnergy ||
    flag->d_reductionVars->strainEnergy) {
    new_dw->put(sum_vartype(0.), lb->StrainEnergyLabel);
}
}

void Thixotropic::addComputesAndRequires(Task* task,
                                         const MPMMaterial* matl,
                                         const PatchSet* patches) const
{
    // Add the computes and requires that are common to all explicit
    // constitutive models. The method is defined in the
    ConstitutiveModel
    // base class.
    const MaterialSubset* matlset = matl->thisMaterial();
    addSharedCRForExplicit(task, matlset, patches);
}

```

```

        Ghost::GhostType gnone = Ghost::None;
        task->requires(Task::OldDW, lb->pLocalizedMPMLabel, matlset,
gnone);
        task->requires(Task::OldDW, lb->pParticleIDLabel, matlset,
gnone);
        task->requires(Task::OldDW, lb->pXLabel, matlset,
gnone);
        task->requires(Task::OldDW, p_structureLabel, matlset,
gnone);
        task->requires(Task::OldDW, p_viscoLabel, matlset,
gnone);
        task->requires(Task::OldDW, p_existsLabel, matlset,
gnone);

        task->computes(lb->pLocalizedMPMLabel_preReloc, matlset);
        task->computes(p_viscoLabel_preReloc, matlset);
        task->computes(p_structureLabel_preReloc, matlset);
        task->computes(p_existsLabel_preReloc, matlset);
    }

void Thixotropic::addComputesAndRequires(Task* ,
                                         const MPMMaterial* ,
                                         const PatchSet* ,
                                         const bool ) const
{
}

// The "CM" versions use the pressure-volume relationship of the CNH
model
double Thixotropic::computeRhoMicroCM(double pressure,
                                       const double p_ref,
                                       const MPMMaterial* matl,
                                       double temperature,
                                       double rho_guess)
{
    double rho_orig = matl->getInitialDensity();
    double bulk = d_initialData.d_Bulk;

    double p_gauge = pressure - p_ref;
    double rho_cur;

    double p_g_over_bulk = p_gauge/bulk;
    rho_cur=rho_orig*(p_g_over_bulk + sqrt(p_g_over_bulk*p_g_over_bulk
+1.));

    return rho_cur;
}

void Thixotropic::computePressEOSCM(const double rho_cur,double&
pressure,
                                    const double p_ref,
                                    double& dp_drho, double& tmp,
                                    const MPMMaterial* matl,
                                    double temperature)

```

```

{
    double bulk = d_initialData.d_Bulk;
    double rho_orig = matl->getInitialDensity();

    double p_g = .5*bulk*(rho_cur/rho_orig - rho_orig/rho_cur);
    pressure = p_ref + p_g;
    dp_drho = .5*bulk*(rho_orig/(rho_cur*rho_cur) + 1./rho_orig);
    tmp = bulk/rho_cur; // speed of sound squared
}

double Thixotropic::getCompressibility()
{
    return 1.0/d_initialData.d_Bulk;
}

namespace Uintah {

#if 0
    static MPI_Datatype makeMPI_CMDData()
    {
        ASSERTEQ(sizeof(Thixotropic::StateData), sizeof(double)*0);
        MPI_Datatype mpitype;
        Uintah::MPI::Type_vector(1, 0, 0, MPI_DOUBLE, &mpitype);
        Uintah::MPI::Type_commit(&mpitype);
        return mpitype;
    }

    const TypeDescription*
fun_getTypeDescription(Thixotropic::StateData*)
{
    static TypeDescription* td = 0;
    if(!td){
        td = scinew TypeDescription(TypeDescription::Other,
                                    "Thixotropic::StateData",
                                    true, &makeMPI_CMDData);
    }
    return td;
}
#endif
} // End namespace Uintah

```

## Annex 2 – Sample Input File

```
<?xml version="1.0" encoding="iso-8859-1"?>
<Uintah_specification>

    <Meta>
        <title>poiseuille flow - use gravity as pressure
gradient</title>
    </Meta>

    <SimulationComponent type="mpm" />

    <Time>
        <maxTime>15</maxTime>
        <initTime>0.0</initTime>
        <delt_min>1.0e-7</delt_min>
        <delt_max>0.01</delt_max>
        <max_Timesteps>3000000</max_Timesteps>
        <timestep_multiplier>0.2</timestep_multiplier>
    </Time>

    <DataArchiver>
        <filebase>timesteptest.uda</filebase>
        <outputInterval>1</outputInterval>
        <save label="p.x"/>
        <save label="p.velocity"/>
        <save label="p.viscosity"/>
        <save label="g.mass"/>
        <save label="g.velocity"/>
    </DataArchiver>

    <MPM>
        <time_integrator>explicit</time_integrator>
        <interpolator>gimp</interpolator>
        <artificial_damping_coeff>0.0</artificial_damping_coeff>
    </MPM>

    <PhysicalConstants>
        <gravity>[0,-9.81,0]</gravity>
    </PhysicalConstants>

    <MaterialProperties>
        <MPM>
            <material>
                <density>1500</density>
                <constitutive_model type="thixotropic">
                    <bulk_modulus>15000</bulk_modulus>
                    <gamma>7.0</gamma>
                    <minimum_viscosity>4</minimum_viscosity>
                    <alpha_parameter>1</alpha_parameter>
                    <t_time_parameter>0.5</t_time_parameter>
                    <initial_structure>1.5</initial_structure>
                    <exponent_n>2.5</exponent_n>
                </material>
            </MPM>
        </MaterialProperties>
    </PhysicalConstants>
```

```

        <max_viscosity>5000</max_viscosity>
    </constitutive_model>
    <thermal_conductivity>0.0</thermal_conductivity>
    <specific_heat>5</specific_heat>
    <geom_object>
        <box label = "boxL">
            <min>[0, 0, 0]</min>
            <max>[0.2, 0.2, 0.1]</max>
        </box>
        <res>[2,2,2]</res>
        <velocity>[0.0,0.0,0.0]</velocity>
        <temperature>12</temperature>
    </geom_object>
</material>

<contact>
    <type>null</type>
</contact>
</MPM>
</MaterialProperties>

<Grid>
    <BoundaryConditions>
        <Face side = "x->">
            <BCType id = "all" label = "Symmetric" var =
"symmetry"> </BCType>
            </Face>
        <Face side = "x+>">
            <BCType id = "all" label = "Symmetric" var =
"symmetry"> </BCType>
            </Face>

        <Face side="y->">
            <BCType id="all" var="Dirichlet" label="Velocity">
                <value> [0.0,0.0,0.0] </value>
            </BCType>
            </Face>

        <Face side="y+>">
            <BCType id="all" var="Dirichlet" label="Velocity">
                <value> [0.0,0.0,0.0] </value>
            </BCType>
            </Face>

        <Face side = "z->">
            <BCType id = "all" label = "Symmetric" var =
"symmetry"> </BCType>
            </Face>
        <Face side = "z+>">
            <BCType id = "all" label = "Symmetric" var =
"symmetry"> </BCType>
            </Face>
    </BoundaryConditions>
    <Level>
        <Box label="1">

```

```
<lower>[0, 0, 0]</lower>
<upper>[2, 0.24, 0.1]</upper>
<resolution>[300, 36, 1]  </resolution>
<extraCells>[1,1,1]      </extraCells>
<patches>[5,5,1]         </patches>
</Box>
</Level>

</Grid>

</Uintah_specification>
```