Escola de Camins
Escola Tècnica Superior d'Enginyeria de Camins, Canals i Ports
UPC BARCELONATECH

MASTER THESIS

Master

Numerical Methods in Engineering

Title

COUPLED MODELS IN DAM ENGINEERING

Author

María de los Ángeles González Gutiérrez

Tutor

Antonia Larese and Pablo Becker

Date

September 28, 2016

ACKNOWLEDGEMENTS

This thesis represents the end of an enriching period for my personal and professional life. In this educational experience, there are people who deserve special recognition because without their valuable support and help this work would not have been possible.

First, I would like to thank God for giving me strength and inspiration in moments of fatigue.

I would like to thanks to my tutors: Antonia Larese and Pablo Becker for their patience and support.

My thanks also go to my colleague Mauricio for his advices.

And last but not least I want to express my special gratitude to my husband Abrahan and to my parents: Elide and Jesús for their unconditional support.

ABSTRACT

Keywords = **rockfill dams, overtopping, PFEM, PFEM-2, Free-surface, Eulerian approach**.

Rockfill dams are certainly one of the most important engineering structures due to their economic advantages and flexible design. Unfortunately their vulnerability to overtopping still remains their weakest point. For that reason, several research groups are interested in both the numerical and experimental analysis of this phenomenon. In this work we focused on the numerical analysis.

The previous work developed in CIMNE on a coupled PFEM-Eulerian free surface Computational Fluid Dynamic (CFD-PFEM) technique for the simulation of porous problems has been studied and extended in the present work.

The Particle Finite Element second generation method (PFEM-2) has been used as an alternative to the Particle Finite Element Method (PFEM). Hence a new coupling strategy, called CFD-PFEM-2, has been developed. On one hand, the fluid free surface problem outside and inside the rockfill slope is treated using a unique Eulerian fixed mesh formulation. On the other hand, the structural dam response is analyzed using a mixed Eulerian–Lagrangian framework (the PFEM-2 method).

A comparison between the coupled strategies CFD-PFEM, and CFD-PFEM-2 has been performed. Additionally, all the numerical results are compared with experiments on prototype rockfill dams.

The results obtained with the new strategy show a good agreement with both the experimental results and the CFD-PFEM code. Furthermore, the computational efficiency between the two numerical strategies was found to be almost identical. Given the similar accuracy and required computational power, it is concluded that the CFD-PFEM-2 is a competitive strategy to simulate the phenomenon.

RESUMEN

Palabras claves: **presas de escollera, sobrevertido, PFEM, PFEM-2, superficie libre CFD** .

Las presas de escollera son consideras unas de las obras ingenieriles más importantes en la actualidad. Las dos principales razones son: su ventaja económica y la flexibilidad en el diseño. No obstante, la principal desventaja que poseen es su vulnerabilidad frente a un proceso de sobrevertido. Es por esta razón, que una gran cantidad de grupos de investigación se encuentran interesados en estudiar este fenómeno. Esta tesis se ha enfocado sólo en el análisis numérico.

El trabajo previo desarrollado en CIMNE llamado a coupled PFEM-Eulerian free surface Computational Fluid Dynamic (CFD-PFEM) technique ha sido estudiado y extendido en esta tesis.

El método Particle Finite Element segunda generación (PFEM-2) ha sido usado como alternativa al método PFEM. Por lo tanto, una nueva estrategia de acoplamiento llamada CFD-PFEM-2 ha sido desarrollada y evaluada. Por una parte, el problema del fluido tanto adentro como fuera de la presa es tratado usando una única formulación Euleriana con malla fija. Por otra parte, la respuesta estructural de la presa ha sido analizada usando una formulación Euleriana- Lagrangiana.

Una comparación entre ambas estrategias CFD-PFEM y CFD-PFEM-2 fue realizada. Además todos los resultados numéricos obtenidos fueron comparados con los experimentales.

Los resultados obtenidos con la nueva estrategia muestran un buen acuerdo tanto con los resultados experimentales como con el código CFD-PFEM. Además, se encontró que la eficiencia computacional entre las dos estrategias numéricas es similar. Por lo tanto, dadas las similitudes entre ambos métodos, tanto en precisión como en la potencia de cálculo requerida, se concluye que el método CFD-PFEM-2 presenta las características necesarias para hacer un estudio del fenómeno comentado.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Embankment dams are certainly one of the most important engineering structures, both for their technical complexity, and for the services that they provide. For that reason, the safety analysis of new dams and rehabilitation of the existing ones are nowadays open fields of research. In fact in many countries the design criteria of these structures have recently been reviewed with the intention of increasing the safety level facing an exceptional flooding [1]. This can be justified considering that usually the cause of the overflows is an extreme meteorological event, which could be produced by climate change induced by global warming.

While in a concrete dam, an overflow does not easily affect the integrity of the structure, in an embankment dam, in most cases, it compromises the dam body [9]. If a dam fails, loss of life and economic damage could be direct consequences of such event [1]. Hence, early warning is crucial for saving lives in flood-prone areas. This is the reason why there exists an increasing interest on the study of stability of embankment dams in overtopping situations. Overtopping occurs when water overpasses the crest of the dam (Figure 1.1).

Figure 1.1: Overtopping in a concrete dam. Image taken from [18].

Several examples of dam failures as the consequence of overtopping can be found in the literature, as is the case of the Tous dam in Spain. In October 1982, a tsunami of 20 million of m^3 of water flowed through the Valencia community (Figure 1.2). In that case the cause of the exceptional flooding was a particular meteorological condition called "gota fria" which

consists of a cold high-altitude depression surrounded by warm air with high moisture content that leads to extremely heavy rain fall in the hinterland of the Mediterranean cost of Spain [1].

Figure 1.2: Images of Tous dam after the overtopping of October, 1982. Images taken from [1].

This kind of environmental phenomena makes necessary to study the main characteristics of embankment dams and their behaviour during extreme events. The first point to remark is the dam's classification. The embankment dams are divided in rockfill and earthfill dams, where the main difference is the size and the frictional internal angles of the material of the mount. Particularly in the case of rockfill dams the permeability is extremely high. This makes this kind of structures particularly vulnerable to overtopping. In fact, as soon as water overpasses the crest of the dam, an extremely fast and turbulent seepage process starts. Seepage can rapidly compromise the stability of the downstream toe and trigger the dam progressive failure.

Earthfill dams, on the contrary, are made of materials with lower permeabilities, and the short duration of an overtopping is not sufficient to see a deep seepage occurrence. In this case superficial erosion is the critical process to be studied to define the failure mechanism.

In both cases a deeper knowledge of the failure mechanism is extremely important to improve the safety and define emergency measures. This is the reason why several research groups are interested in both the numerical and experimental analysis of these phenomena.

In this work we focused on the numerical analysis of overtopping in rockfill dams only. We recover and extend the previous work done by Larese and co workers [1–5] and we extend it so to investigate alternative numerical approaches and possible improvements. Larese proposed an Eulerian free surface computational fluid dynamic (CFD) code [1, 4]. This free surface code is able to consider the presence of a porous medium and to handle variable discharge conditions. The model is conceived to represent seepage problems in porous medium like rockfill or other materials with extremely high permeability [1, 4]. The governing equations are solved using a stabilized Finite Element method (FEM) in an Eulerian Framework.

Moreover, in order to simulate correctly the behaviour of rockfill dams during overtopping it is not enough just to model the seepage process but the evaluation of the dam body response is also needed. The dam body is simulated as a non-Newtonian material. The technique used is PFEM, a Lagrangian technique that can naturally handle large deformations. This structural code is coupled to the fluid in order to simulate the overtopping and failure

mechanism.

The results obtained with the coupled code have shown a good agreement between experimental and numerical values, principally for 2D models. However, limitations in 3D coupled CFD-PFEM models were observed both in terms of computational efficiency (the remeshing process in PFEM is not parallelizable) and accuracy of the results. These aspects will be discussed in the following chapters. Therefore, an improvement in this coupled strategy is needed in order to reach a more realistic simulation in the analysed problem.

For that reason, **this work deals with the evaluation of a new method called Particle Finite Element Method Second Generation (PFEM-2) as a new strategy to simulate the structural response of the dam in the coupled problem when an overtopping occurs**. In order to achieve the main objective a computational comparison between PFEM and PFEM-2 is performed. The main difference between PFEM and PFEM-2 is that PFEM-2 uses a fixed finite element mesh avoiding the re-meshing process.

Additionally, it is important to point out that PFEM-2 has been validated for several coupled problems such as multi-fluids, and fluid-structure interaction (FSI) [7, 8]. However, it has not been used for modelling the behaviour of rockfill dams during overtopping scenarios. Therefore, the current work is a starting point in this field of research.

1.1 Embankment dams

Given their economic benefits and flexible design, that makes possible the use of different shapes and materials that can be found in situ, the embankment dams are widely used. They are usually chosen on sites with wide valleys and can be built in areas with relatively poor bearing capacity given the fact that the shear forces that they generated are smaller than the forces generated by concrete dams, which always need qualified foundations. This kind of dams only work as gravity dams, meaning that their own weights are responsible for resisting the water's pressure (Figure 1.3).

Figure 1.3: Gravity dams. Image taken from [20].

Types of embankment dams

The embankment dams are commonly divided in earthfill and rockfill dams. The fundamental difference between one and the other lies in the material used for its construction:

- Earthfill dam: this is made of materials with low friction angle. Its height is limited by the large amount of material needed. The usual relations between their dimensions are presented in Figure 1.4.

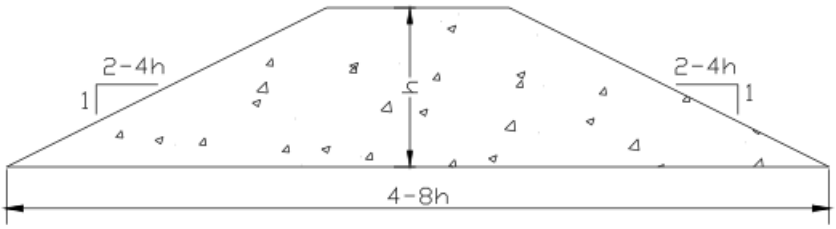


Figure 1.4: Usual dimensions in earthfill dams. Image taken from [12].

- Rockfill dam: this is made of materials with high friction angle, null cohesion and high permeability. The usual relations between its dimensions are presented in Fig 1.5.

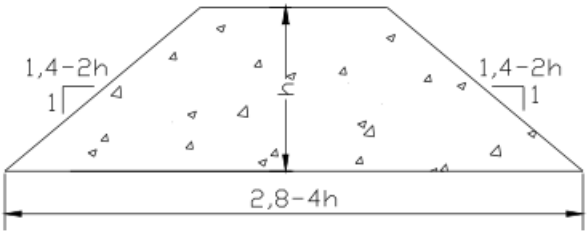
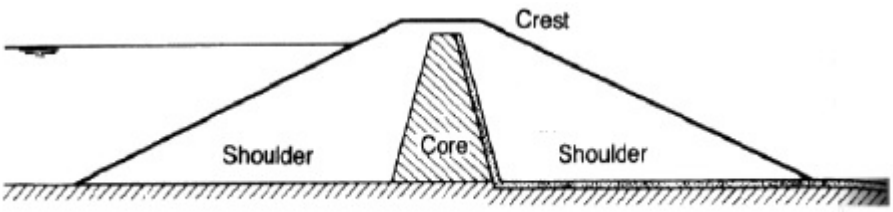


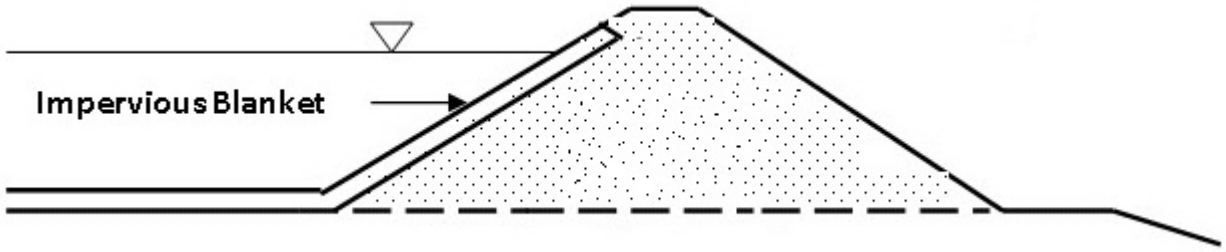
Figure 1.5: Usual dimensions in rockfill dams. Image taken from [12].

The extremely high permeability of the rockfill medium makes necessary the presence of an impervious structure inside or above the dam body. The most relevant standard procedures are: impervious core (Figure 1.6a) and upstream impervious screen (Figure 1.6b).

From now on we will concentrate only on rockfill dams, leaving earthfill dams for future works.



(a) Dam with an internal core



(b) Dam with an impermeable cover on the upstream face

Figure 1.6: Types of rockfill dams. Images taken from [19].

Disadvantages

Despite the advantages offered by the embankment dams, their weakest point is the vulnerability of the structural body when an overtopping occurs. In fact, according to the ICOLD bulletin [9], this is their principal or secondary cause of failure in 31% (rockfill dams) and 18% (earthfill dams) of cases respectively.

When an overtopping occurs in a rockfill dam a seepage process begins in the downstream slope. Resistance is not instantaneously compromised, but it is progressively reduced. The first breach usually appears at the toe of the dam, where the resistance is lower. There are two main mechanics failures: mass slinding and superficial erosion [10, 11]. This two mechanisms usually act in a combined way depending on the failure process [10].

- Mass slinding or loss of stability of a part of the downstream region. This is the predominant failure mechanism when the downstream slope is very steep. The saturation of the rockfill leads to a reduction of effective stresses that induces the formation of a failure circle that abruptly crumbles (Figure 1.7a) .
- Superficial dragging of rockfill particles. This is the predominat failure when the downstream slope is flat, for example 1V: 3H. The water coming out from the toe of the dam drags away the superficial rocks (Figure 1.7b).

(a) Mass sliding failure

(b) Superficial dragging failure

Figure 1.7: Mechanics of failure in embankment dams. Images taken from [1].

1.2 Objectives

Good results were observed in the literature using the PFEM method for the simulation of the dam response, however some limitations have been observed principally in the 3D models [1], [2]. On the one hand, PFEM requires a continuous remeshing of the deformed domain to avoid excessive mesh distortion. Remeshing is a serial process, not easy parallelizable. This aspect seriously compromise the computational efficiency in 3D cases. On the other hand lack of sufficient accuracy was also observed in 3D cases due to the noise generated by remeshing. For those reasons, **this thesis is focused on evaluating the possibility of using PFEM-2 method as a valid strategy for the simulation of the initial stage of failure in a prototype rockfill dams when an overtopping occurs**. PFEM-2 in fact, avoids remeshing using a parallel projection strategy between the moving particles and a fixed background grid.

In order to fulfill the main purpose the following steps have been set:

- Characterize the Particle Finite Element (PFEM) and the Particle Finite Element Second Generation (PFEM-2) methods.
- Compare the methods PFEM, and PFEM-2 using a set of Two-Dimensional (2D) and Three-Dimensional (3D) examples.
- Develop a tool to couple the free surface/seepage CFD code with PFEM-2.
- Simulate two prototypes small scale of rockfill dams using the coupled models.
- Compare the obtained numerical results with the given experimental data.

The experimental data of the XPRES [13] and E-DAMS [14] projects are used in this work to validate the numerical approach of the coupled models. All the algorithms used are implemented in Kratos [16], a framework for developing finite element codes for multiphysics problems. The pre and post process visualization is performed using the GID program [15].

1.3 Layout of the document

This work is divided in 5 chapters:

Chapter 1. The introduction. The motivation and main objectives of the work are presented.

Chapter 2. The numerical approach. Both the fluid problem as the structural problem are described. The governing equations, the weak form, the stabilization techniques, the time scheme and the strategies to solve the problems are presented.

Chapter 3. Methodology. The steps and tools used in order to achieve the principal objective are indicated.

Chapter 4. Numerical results and conclusions. The numerical examples are presented and analyzed. A evaluation of the coupled models are performed using an experimental small-scale dams.

Chapter 5. The conclusions and future work are explained.

Appendix A and B. The coupling algorithm between the CFD code and PFEM-2 is presented.

CHAPTER 2

NUMERICAL APPROACH

2.1 Introduction

The fluid and the structural problem presented in this work are treated with two different approaches. The fluid problem is solved using an Eulerian approach (level set technique) whereas for the structural one a Lagrangian approach is preferred (PFEM and PFEM-2 techniques). These choices are the consequence of the following considerations: the structure suffers large deformation and shape changes. On the other hand the fluid problem is treated using a fixed mesh and a level set technique to track the evolution of the free surface.

In this chapter a brief overview of the main characteristics of the different numerical techniques used in this work is presented. It should be noted that for a complete analysis of the details of the methods the external references mentioned in the text should be consulted. The objective of the next sections is to help the reader understanding the main ingredients of the different techniques used.

At the end of the chapter both the pre existing CFD-PFEM and the newly developed CFD-PFEM-2 coupled strategies designed for the fluid structure interaction problem are also described.

2.2 The fluid problem

2.2.1 Flow in rockfill material

The flow of fluids through porous media is traditionally studied using the well known Darcy law, which was formulated by the French civil engineer Henry Darcy in 1856 on the basis of his experiment on vertical water filtration through a sand-filled column. He discovered that the pressure drop ($\bar{i}$) and the velocity of water inside a porous material (u) are linearly related. The formulation of the empirical Darcy law is:

$$\bar{i} = \frac{\mu}{k} u \tag{2.1}$$

where μ is the water dynamic viscosity and k is the permeability of the porous media [21] and is defined as

$$k = \frac{n^3 D_p^2}{5(1-n)^2\theta} \tag{2.2}$$

where D_p is an equivalent diameter of the porous material and θ is a shape coefficient of the particles.

Remark 1. It is important define two different measures for the fluid velocity. The first one is the Darcy velocity u being the fluid velocity averaged over the total control volume Ω. The second one is the fluid velocity $\bar{u}$ being the averaged over the empty part of Ω. Their relation is stated by the Dupuit-Forchheimer equation [22]:

$$u = n\bar{u} \tag{2.3}$$

where n is the porosity, defined as the relation between the empty volume and the total control volume:

$$n = \frac{\Omega_E}{\Omega} \tag{2.4}$$

Figure 2.1 presents a graphical description of the velocities.

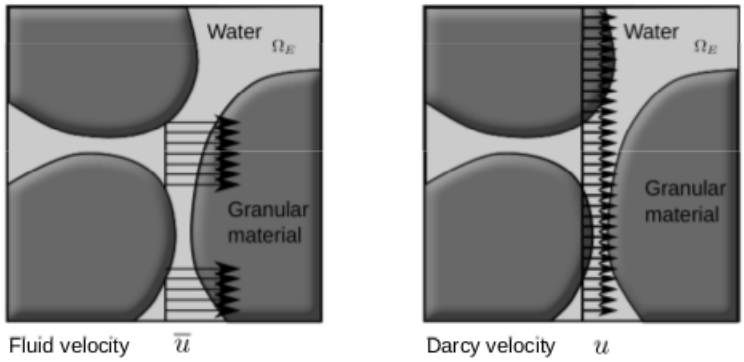


Figure 2.1: Graphical description of fluid velocity $\bar{u}$ and Darcy velocity u. Image taken from [1]

Despite the simplicity of equation 2.1, this formula provides an accurate approach in many geotechnical applications. However, this linear behaviour is only valid for low Reynolds numbers, therefore limited to low velocities and small grain sizes. For flux through rockfill material, the local fluid velocities were observed not be linearly related to the pressure drop, meaning that the Darcy law is not applicable to the analyzed problem. Instead, an analogy between the flow in pipes and the flow in rockfill material is performed with the purpose to derive a suitable non-linear resistance law [1].

2.2.2 Resistance laws

It is widely accepted to consider the flow in the pores of rock particles essentially similar to flow in pipe network but with a more complicated configuration [10, 23]. All the empirical formulae to evaluate the pressure drop due to friction in pipe have been used and adapted to get similar empirical relationships in the case of porous material [24, 25].

Forchheimer (1901) was one of the first in proposed a quadratic resistance law:

$$\mathbf{i} = \alpha \mathbf{u} + \beta \mathbf{u}^2 \tag{2.5}$$

where constants α and β depend only on the characteristics of rockfill material.
Prony (1804) and Jeager (1956) proposed and exponential law like

$$\mathbf{i} = \gamma \mathbf{u}^\eta \tag{2.6}$$

where constants γ and η depend on the flow condition, the characteristics of the porous medium and the fluid.

Nowadays, both equations are widely used and all efforts have been addressed in determining the α, β, γ and η constants, as is the case of Ergun (1952), Wilkins (1956), McCorquodale (1978) and others. A comprehensive overview of the different models can be found in [10, 23, 24].

Larese [1, 4] adopted the Ergun's law as the non-linear resistance for her algorithm. Two conditions were taken into account. The first condition is that the resistance law it should go automatically to zero when $n = 1$, in order to simulate the free surface flow through the rockfill and outside. The second one is that a quadratic form is easier to implement than the exponential resistance law. The pressure drop for Ergun's law yields

$$\bar{\mathbf{i}} = E_1 \mathbf{u} + E_2 \mathbf{u}^2 \tag{2.7}$$

where the constant are defined as:

$$E_1 = 150 \frac{(1-n)^2}{n^3} \frac{\mu}{D_p^2} \tag{2.8}$$

and

$$E_2 = 1.75 \frac{(1-n)}{n^3} \frac{\rho}{D_p} \tag{2.9}$$

D_p is the average diameter of the granular material ($D_p = D_{50}$).

Defining the permeability shape coefficient $\theta = 30$, the permeability k can be calculated as a function of n and D_p

$$k = \frac{n^3 D_p^2}{150(1-n)^2} \tag{2.10}$$

The final form of the resistance law chosen in this work is :

$$\boxed{\bar{\mathbf{i}} = \frac{\mu}{k}\mathbf{u} + \frac{1.75}{\sqrt{150}} \frac{\rho}{\sqrt{k} n^{3/2}} \mathbf{u}^2} \tag{2.11}$$

It is important to note that the linear part of equation 2.11 is equivalent to the Darcy's law.

2.2.3 Balance equations

The balance of linear momentum and the continuity equation for an incompressible fluid can be derived due to the resistance law has been chosen. It is important to stress that the governing equations have to be able to reproduce the free surface flow in a variable porosity medium, considering air as a porous medium with $n = 1$.

The unknowns of the problem are:

$\mathbf{u}$: Darcy velocity.
p : fluid pressure.

The other parameters involved in the governing equations are:

ρ : is the fluid density.
μ: is the fluid dynamic viscosity.
n : is the porosity.

Continuity equation

The continuity equation is

$$\frac{\partial u}{dx} + \frac{\partial v}{dy} = 0 \tag{2.12}$$

It can be also rewritten as:

$$\boxed{\nabla \cdot \mathbf{u} = 0} \tag{2.13}$$

Momentum equation

The balance equation of the linear momentum can be written as

$$\boxed{\rho \partial_t \mathbf{u} + \rho \bar{\mathbf{u}} \cdot \nabla \mathbf{u} + n \nabla p - 2\nabla \cdot \mu \nabla^s \mathbf{u} - \rho \mathbf{b} n + \mathbf{D} = 0} \tag{2.14}$$

where $\mathbf{D}$ is the matricial form of the resistance law (equation 2.11). It represents the dissipative effects due to the interaction between the solid and the fluid part.

$$\mathbf{D} = \frac{n\mu}{k}\mathbf{u} + \frac{1.75}{\sqrt{150}} \frac{\rho n}{\sqrt{k}} \frac{\mid \mathbf{u} \mid}{n^{3/2}} \mathbf{u} \tag{2.15}$$

The modified form of the Navier-Stokes problem are

$$\begin{aligned} \rho \partial_t \mathbf{u} + \rho \bar{\mathbf{u}} \cdot \nabla \mathbf{u} + n \nabla p - 2\nabla \cdot \mu \nabla^s \mathbf{u} - \rho \mathbf{b} n + E_1 \mathbf{u} + E_2 \mid \mathbf{u} \mid \mathbf{u} = \mathbf{0} \text{ in } \Omega, \text{ t} \in (0, \text{ T}); \\ \nabla \cdot \mathbf{u} = 0 \text{ in } \Omega, \text{ t} \in (0, \text{ T}); \end{aligned} \tag{2.16}$$

being $\Omega \subset \mathbb{R}^d$ the fluid domain in a time interval $(0, T)$. The boundary and initial condition of the previous problem are:

$$\begin{aligned} \mathbf{u}(\mathbf{x},0) &= \mathbf{u}_0(\mathbf{x}) \text{ in } \Omega; \\ \mathbf{u}(\mathbf{x},t) &= \mathbf{g}(\mathbf{x},t) \text{ on } \partial\Omega_D,\ t \in (0,T); \\ \mathbf{n}\cdot\sigma(\mathbf{x},t) &= \mathbf{t}(\mathbf{x},t) \text{ on } \partial\Omega_N,\ t \in (0,T); \end{aligned} \tag{2.17}$$

where σ is defined by equation 2.18 and Ω_D and Ω_N are the Dirichlet and Neumann boundary respectively.

$$\sigma = -np\mathbf{I} + 2\mu\nabla^s\bar{\mathbf{u}} \tag{2.18}$$

Remark 2. It is important to note that $\mathbf{n}$ indicates the outer unit normal vector whereas n is defined as porosity.

2.2.4 Weak form

The weak form of equations 2.16 is derived next using a Galerkin formulation. A mixed finite element method is obtained, that is the approximation of both the velocity components and the pressure need to be introduced. The weak form yields,

$$\begin{aligned} &\int_\Omega \mathbf{w}\rho\partial_t\mathbf{u}d\Omega + \int_\Omega \mathbf{w}\rho\bar{\mathbf{u}}\cdot\nabla\mathbf{u}d\Omega + \int_\Omega \mathbf{w}n\nabla p d\Omega \\ &- \int_\Omega \mathbf{w}\nabla\cdot 2\mu\nabla^s\mathbf{u}d\Omega + \int_\Omega \mathbf{w}(E_1\mathbf{u}+E_2 \mid \mathbf{u} \mid \mathbf{u})d\Omega - \int_\Omega \mathbf{w}\rho n\mathbf{b}d\Omega = \mathbf{0}\ \ \forall\mathbf{w}\in\mathcal{V}; \\ &\int_\Omega q\nabla\cdot\mathbf{u} = 0\ \forall q\in\mathcal{Q} \end{aligned} \tag{2.19}$$

where for a fixed $t \in (0,T)$, $\mathbf{u}$ is assumed to belong to the velocity space $\mathcal{V} \in [\mathbf{H}^1(\Omega)]^d$ of vector functions whose components and their first derivatives are square-integrable, and p belongs to the pressure space $\mathcal{Q} \in \mathbf{L}_2$ of square-integrable functions. The velocity and pressure weightings functions are $\mathbf{w}$ and q respectively.

Integrating by parts the pressure and convective terms and calling $\Gamma = \partial\Omega$ gives

$$\begin{aligned} \int_\Omega \mathbf{w}n\nabla p d\Omega &= -\int_\Omega np\nabla\cdot\mathbf{w}d\Omega + \int_\Gamma \mathbf{w}\cdot np\mathbf{n}d\Gamma \\ \int_\Omega \mathbf{w}\nabla\cdot 2\mu\nabla^s\mathbf{u}d\Omega &= -2\int_\Omega \nabla\mathbf{w}:\mu\nabla^s\mathbf{u}d\Omega + \int_\Gamma \mathbf{w}\cdot(2\mu\mathbf{n}\cdot\nabla^s\mathbf{u})d\Gamma \end{aligned} \tag{2.20}$$

where $\mathbf{n}$ is the outer normal vector. Substituting equation 2.20 into equations 2.19 and considering the Neumann boundary condition, the system to be solved becomes

$$\begin{aligned} &\int_\Omega \mathbf{w}\rho\partial_t\mathbf{u}d\Omega + \int_\Omega \mathbf{w}\rho\bar{\mathbf{u}}\cdot\nabla\mathbf{u}d\Omega - \int_\Omega np\nabla\cdot\mathbf{w}d\Omega \\ &+2\int_\Omega \nabla\mathbf{w}:\mu\nabla^s\mathbf{u}d\Omega + \int_\Omega \mathbf{w}(E_1\mathbf{u}+E_2 \mid \mathbf{u} \mid \mathbf{u})d\Omega - \int_\Omega \mathbf{w}\rho n\mathbf{b}d\Omega - \int_\Gamma \mathbf{w}\cdot\mathbf{t}d\Gamma = \mathbf{0}\ \ \forall\mathbf{w}\in\mathcal{V}; \\ &\int_\Omega q\nabla\cdot\mathbf{u}d\Omega = 0\ \forall q\in\mathcal{Q} \end{aligned} \tag{2.21}$$

Let $\mathcal{V}_h$ be a finite element space to approximate $\mathcal{V}$ and $\mathcal{Q}_h$ a finite element approximation to $\mathcal{Q}$. Now, the problem is finding $\mathbf{u}_h \in \mathcal{V}_h$ and $p_h \in \mathcal{Q}_h$ such that

$$\begin{aligned}
&\int_\Omega \mathbf{w}_h \rho \partial_t \mathbf{u}_h d\Omega + \int_\Omega \mathbf{w}_h \rho \bar{\mathbf{u}}_h \cdot \nabla \mathbf{u}_h d\Omega - \int_\Omega n p_h \nabla \cdot \mathbf{w}_h d\Omega \\
&+2\int_\Omega \nabla \mathbf{w}_h : \mu \nabla^s \mathbf{u}_h d\Omega + \int_\Omega \mathbf{w}_h (E_1 \mathbf{u}_h + E_2 \mid \mathbf{u}_h \mid \mathbf{u}_h) d\Omega - \int_\Omega \mathbf{w}_h \rho n \mathbf{b} d\Omega - \int_\Gamma \mathbf{w}_h \cdot \mathbf{t}_h d\Gamma = \mathbf{0} \\
&\forall \mathbf{w}_h \in \mathcal{V}_h; \\
&\int_\Omega q_h \nabla \cdot \mathbf{u}_h d\Omega = 0 \ \forall q_h \in \mathcal{Q}_h
\end{aligned} \tag{2.22}$$

2.2.5 Free surface and seepage Fluid Dynamic Code (CFD)

Larese [1–3] developed a code to solve the free surface and seepage for the current problem using a level set technique. The CFD code is able to consider the presence of a porous medium and to handle variable discharge condition. Since the CFD is based on the use of a fixed-grid approach, the solution method has to be completed by the choice of a tracking method for the free surface. For that reason, the level set method is used to follow the moving interfaces. The interface is implicitly defined as the zero-valued iso-surface of a given smooth function, in this case a distance function. A detailed description of the procedure can be found in [1,34].

Stabilized formulation

Since only simplicial P1/P1 elements are used in the free surface and seepage fluid dynamic code, as suitable stabilization technique is necessary to overcome pressure and convective instabilities. The Orthogonal sub-grid scale OSS method introduced by Codina [30, 31] is used. In this case the space for the sub-grid scale is taken orthogonal to the finite element one.

Fractional step solver using an explicit 4^{th} order Runge Kutta time scheme

The modified form of the Navier-Stokes equations are solved using a pressure-splitting approach, proposed by Codina [35]. This option is convenient due to its high computational efficiency for flows at hight Reynolds numbers (Re). The fundamental idea is to solve the momentum equation keeping fixed the pressure and later correcting the pressure so as to guarantee the satisfaction of the divergence constraint. An explicit form of the fractional step solver is used, that is a 4^{th} order Runge Kutta scheme. The advantages of this scheme are: high parallelization and lower computational cost.

For more information consult [1].

2.2.6 The algorithm

Finally the algorithm created by Larese [1] is presented:

Algorithm 1 CFD algorithm

1: Given the level set function φ^n, extrapolate velocity, pressure and pressure gradient so to obtain $\mathbf{u}^n_{ext}, p^n_{ext}$ and ∇p^n_{ext} define as the velocity, pressure and pressure gradient over the extrapolate domain.
2: Convect the level set function φ defining the new free surface at t^{n+1} using $\mathbf{u}^n$ and $\mathbf{u}^n_{ext}$. Note that the extrapolated values are only required within a limited number of layers which are the ones on which the convection will be actually performed.
3: Re-compute (if needed) the distance in the whole domain starting from the zero of the level set function at t^{n+1} obtained at step 2.
4: Solve the momentum equation in equation.
5: Set the approximate pressure boundary conditions on $\partial\Omega^{n+1}$ so to guarantee that the pressure is approximately zero at the position indicated by the zero of the level set function. In order to do that, the geometric distance is evaluated on the first non fluid layer of nodes external to the free surface.
6: Solve for the pressure equation 2.39.
7: Solve for the correction equation 2.40.
8: Move to the next time step.

2.3 The structural problem

The behaviour of the granular non-cohesive material used in rockfill dams is presented in the current section. Two numerical strategies are presented following the work by Larese [1, 6] and Becker [7, 8], respectively.

The simulation of the structural response of a slope made of rockfill materials has been treated using a continuous approach. This is an acceptable choice under the assumption that the rockfill size is small with respect to the overall size of the structure. A non-Newtonian constitutive law has been adopted for the rockfill body due to the capability of the rockfill slope to support a certain amount of shear stress with negligible elastic strain, before starting large deformation, that is the behavior of the yielded material is more similar to a fluid than to the process of deformation of a solid. The constitutive model chosen is a variable visco-rigid model.

2.3.1 Constitutive law

A continuous approach is used to simulate the structural response of a slope made of granular material. A non-Newtonian high viscosity material is chosen due to the high deformation that the structure might be subjected. Following the work by Larese [1, 6] , the constitutive model adopted is a variable yield visco-rigid model.

The regularized model adopted has its origin in a classical Bingham constitutive relation but the yield stress τ_0 is pressure sensitive and it is defined using a Mohr-Coulomb failure criteria without cohesion.

$$\tau_0 = p'_s tg(\phi) \tag{2.23}$$

where p'_s is the effective pressure and ϕ is the internal friction angle. The shear stress τ yields

$$\tau = 2\left[\mu_s + \frac{p'_s tg(\phi)}{\dot{\gamma}}\left(1 - e^{-m\dot{\gamma}}\right)\right]\boldsymbol{\varepsilon}(\mathbf{u}) \tag{2.24}$$

where $\dot{\gamma}$ is the rate of strain, μ is the dynamic viscosity, m is a regularized parameter that controls the approximation to the bilinear model (Figure 2.4) and $\boldsymbol{\varepsilon}$ is the rate of strain tensor.

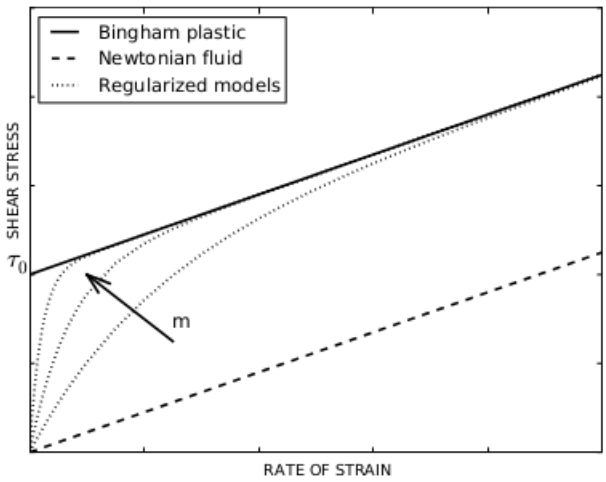


Figure 2.2: Newtonian and Bingham fluid compared with the regularized model for increasing values of m parameter. Image taken from [1]

The resulting apparent viscosity is

$$\tilde{\mu}(\dot{\gamma}) = \mu_s + \frac{p'_s tg(\phi)}{\dot{\gamma}} \left(1 - e^{-m\dot{\gamma}}\right) \tag{2.25}$$

Remark 3. For the structural model in absence of water, the Mohr Coulomb failure is written as

$$\tau_0 = p_s tg(\phi) \tag{2.26}$$

2.3.2 Balance equation

The balance equations are represented by the Navier-Stokes equations for a non-Newtonian fluid.

The unknowns of the problem are:

$\mathbf{u}_s$: velocity of the structure.
p_s : total pressure of the structure.
p'_s : effective pressure of the structure defined as

$$p'_s = p_s - p \tag{2.27}$$

where p is the fluid pressure.

Other parameters involve in the governing equations are:

ρ_s : is the solid dry density.
μ_s: is the dynamic viscosity of the yield material (when Newtonian behaviour is recovered).
$\tilde{\mu}_s$: is the dynamic apparent viscosity (see equation 2.25).

Remark 4. The s sub-index indicates the non-Newtonian variables and parameters. The presented model has been developed in both Lagrangian and Eulerian framework. Hence the convective velocity $\mathbf{a}_s$ is defined in its more general form as

$$\mathbf{a}_s = \mathbf{u}_s - \mathbf{u}_s^M \tag{2.28}$$

where $\mathbf{u}_s^M$ is the mesh velocity. Therefore, $\mathbf{a}_s = 0$ for a Langragian framework and $\mathbf{a}_s = \mathbf{u}_s$ in an Eulerian framework.

The modified Navier-Stokes equations yields

$$\begin{aligned} \rho_s \partial_t \mathbf{u}_s + \rho_s \mathbf{a}_s \cdot \nabla^s \mathbf{u}_s + \nabla p_s - 2\nabla \cdot \tilde{\mu} \nabla \mathbf{u}_s - \rho_s \mathbf{b} = \mathbf{0} \text{ in } \Omega_s, \text{ t} \in (0, \text{ T}); \\ \nabla \cdot \mathbf{u}_s = 0 \text{ in } \Omega_s, \text{ t} \in (0, \text{ T}); \end{aligned} \tag{2.29}$$

where $\Omega_s \subset \mathbb{R}^d$ is the structural domain in a time interval $(0, T)$. The boundary and initial condition of the previous problem are:

$$\begin{aligned} \mathbf{u}_s(\mathbf{x}, 0) = \mathbf{u}_{s0}(\mathbf{x}) \text{ in } \Omega; \\ \mathbf{u}_s(\mathbf{x}, t) = \mathbf{g}_s(\mathbf{x}, t) \text{ on } \partial\Omega_D, \ t \in (0, T); \\ \mathbf{n} \cdot \sigma_s(\mathbf{x}, t) = \mathbf{t}_s(\mathbf{x}, t) \text{ on } \partial\Omega_N, \ t \in (0, T); \end{aligned} \tag{2.30}$$

Ω_D and Ω_N are the Dirichlet and Neumann boundary respectively.

2.3.3 Weak form

The weak form of equations 2.29 is derived using a Galerkin formulation.

$$\begin{aligned} \int_\Omega \mathbf{w} \rho_s \partial_t \mathbf{u}_s d\Omega + \int_\Omega \mathbf{w} \rho_s \mathbf{a}_s \cdot \nabla \mathbf{u}_s d\Omega + \int_\Omega \mathbf{w} \nabla p_s d\Omega \\ - \int_\Omega \mathbf{w} \nabla \cdot 2\tilde{\mu} \nabla^s \mathbf{u}_s d\Omega - \int_\Omega \mathbf{w} \rho_s \mathbf{b} d\Omega = \mathbf{0} \ \ \forall \mathbf{w} \in \mathcal{V}; \\ \int_\Omega q \nabla \cdot \mathbf{u}_s = 0 \ \forall q \in \mathcal{Q} \end{aligned} \tag{2.31}$$

where for a fixed $t \in (0, T)$, $\mathbf{u}_s$ is assumed to belong to the velocity space $\mathcal{V} \in [\mathbf{H}^1(\Omega)]^d$ of vector functions whose components and their first derivatives are square-integrable, and p_s belongs to the pressure space $\mathcal{Q} \in \mathbf{L}_2$ of square-integrable functions. The velocity and pressure weightings functions are $\mathbf{w}$ and q respectively.

Performing the integration by parts of the pressure and the viscous terms as explained in the fluid problem, then the weak form yields

$$\begin{aligned} \int_\Omega \mathbf{w} \rho_s \partial_t \mathbf{u}_s d\Omega + \int_\Omega \mathbf{w} \rho_s \mathbf{a}_s \cdot \nabla \mathbf{u}_s d\Omega - \int_\Omega p_s \nabla \cdot \mathbf{w} d\Omega + 2 \int_\Omega \nabla \mathbf{w} : \tilde{\mu} \nabla^s \mathbf{u}_s d\Omega \\ - \int_\Omega \mathbf{w} \rho_s \mathbf{b} d\Omega - \int_\Gamma \mathbf{w} \cdot \mathbf{t} d\Gamma = \mathbf{0} \ \ \forall \mathbf{w} \in \mathcal{V}; \\ \int_\Omega q \nabla \cdot \mathbf{u}_s d\Omega = 0 \ \forall q \in \mathcal{Q} \end{aligned} \tag{2.32}$$

Let $\mathcal{V}_h$ be a finite element space to approximate $\mathcal{V}$ and $\mathcal{Q}_h$ a finite element approximation to $\mathcal{Q}$. Now, the problem is finding $\mathbf{u}_h \in \mathcal{V}_h$ and $p_h \in \mathcal{Q}_h$ such that

$$\begin{aligned}\int_\Omega \mathbf{w}_h \rho_s \partial_t \mathbf{u}_{sh} d\Omega + \int_\Omega \mathbf{w}_h \rho_s \bar{\mathbf{u}}_{sh} \cdot \nabla \mathbf{u}_{sh} d\Omega - \int_\Omega p_{sh} \nabla \cdot \mathbf{w}_h d\Omega \\ +2\int_\Omega \nabla \mathbf{w}_h : \tilde{\mu} \nabla^s \mathbf{u}_{sh} d\Omega - \int_\Omega \mathbf{w}_h \rho_s \mathbf{b} d\Omega - \int_\Gamma \mathbf{w}_h \cdot \mathbf{t}_h d\Gamma = \mathbf{0} \quad \forall \mathbf{w}_h \in \mathcal{V}_h; \\ \int_\Omega q_h \nabla \cdot \mathbf{u}_{sh} d\Omega = 0 \ \forall q_h \in \mathcal{Q}_h\end{aligned} \tag{2.33}$$

2.3.4 PFEM method

Since the structural domain is expected to undergo severe deformation as the failure progresses, the kinematic model has to adapt dynamically to such deformations. For that reason, the Particle Finite Element Method (PFEM) has been chosen in the work by Larese [1, 5] for its flexibility and reliability.

The PFEM is a numerical method that uses a Finite Element mesh to discrete the physical domain and to integrate the differential governing equations. In PFEM the domain is modeled using an Updated Lagrangian Formulation. That is all the variables are assumed to be known in the current configuration at time t and they are brought in the next configuration at time $t + dt$. The finite element method is used to solve the continuum equations in a mesh built up from the underlying nodes (the particles).

As all Lagrangian methods, the PFEM offers a more natural solution to problems where the "particles" of the domain can move freely. Unlike the level set method, there is no need to recalculate the surface since the location of the interface is obtained trivially. This is useful to model the separation of solid particles from the bed surface and to follow their subsequent motion as individual particles with a known density, an initial acceleration and a velocity subject to gravity forces.

It is important to stress that in PFEM each particle is treated as a material point characterized by the density of the solid domain to which it belongs to. The global mass is obtained by integrating density at the different material points over the domain. The quality of the numerical solution depends on the discretization chosen as in the standard FEM. Adaptive mesh refinement techniques can be used to improve the solution in zones where large gradients of the fluid or the structure variables occur [1].

The basic ingredients of PFEM can be summarized in:

- An Updated Lagrangian kinematical description of the motion, where variables are described in the current configuration Ω^0 at time t^0.
- A fast remeshing algorithm.
- A boundary recognition method (alpha-shape).
- FEM for the solution of the governing equations.

2.3.4.1 Remeshing algorithm

In a Lagrangian approach the need of an efficient remeshing algorithm together with the difficulty of parallelizing this procedure are the biggest drawback. The mesh moves in accordance to the material points and large deformations occur. The code implemented by Larese used external libraries to remesh the domain, which are the Triangle and TetGen for 2D and 3D cases respectively.

TetGen is a program to generate tetrahedral meshes of any 3D polyhedral domains. TetGen generates exact constrained Delaunay tetrahedralizations, boundary conforming Delaunay meshes, and Voronoi partitions. The following Figure 2.5 respectively illustrate a 3D polyhedral domain (left), a boundary conforming Delaunay tetrahedral mesh (middle), and its dual - a Voronoi partition (right). For more detail see [26].

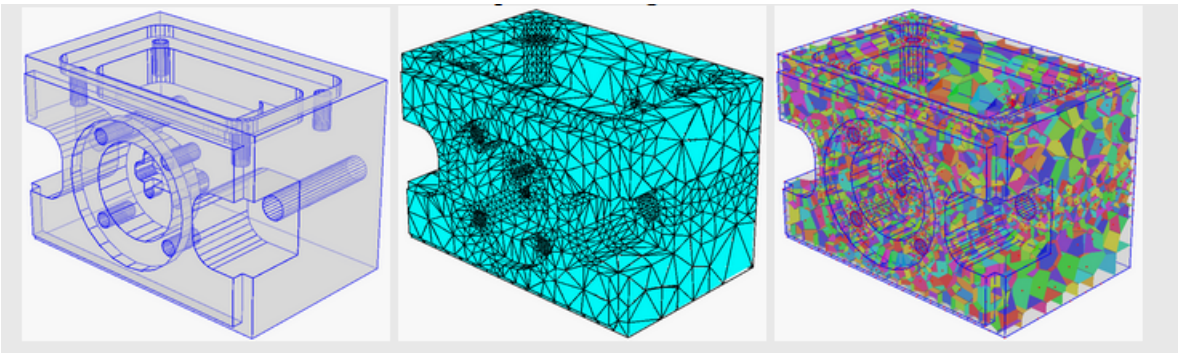


Figure 2.3: Example of a tetrahedral mesh of a 3D polyhedral domain. Image taken from [26]

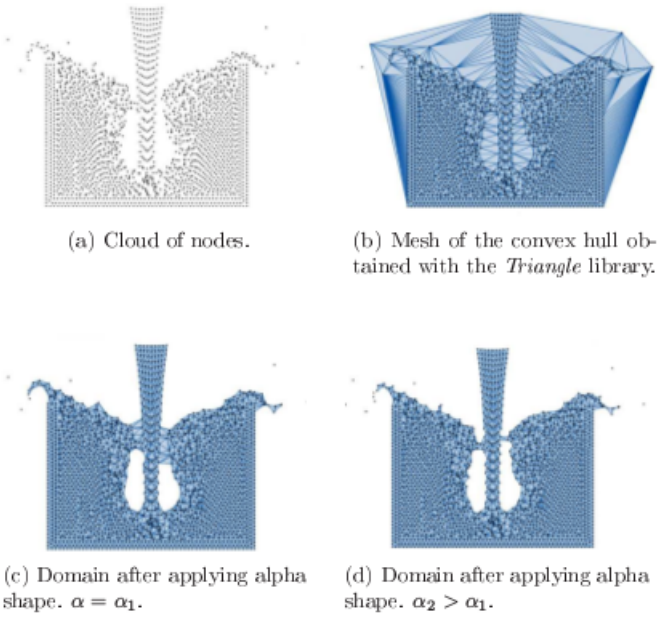
2.3.4.2 Boundary recognition method: alpha-shape method

The α-shape method is used to define the free surfaces and the boundaries on the material domain once it is partitioned using the TetGen library. Each node i of the domain has its own dimension h_i determined as the average distance of node i from its neighbours. In the same way, an elemental dimension h_{el} can be defined for each element as the average of the h_i of its node. Finally, depending on the precision wanted, an α custom parameter greater but close to one is defined. If the radius of the sphere that circumscribes the element (r) is bigger than $\alpha \cdot h_{el}$ then the element is eliminated. Different values of the alpha shape parameter can lead to different accuracy on the mesh boundaries as shown in Figure 2.6 where different values if the alpha parameter are used.

2.3.4.3 FEM

For the FEM solution, a monolithic strategy to solve the governing equations and a Bossak time integration scheme is used.

Stabilized formulation The choice of adopting equal order linear elements (P1/P1) for the velocity and pressure, despite its simplicity, entails the necessity of using a stabilization technique. An Algebraic Subgrid scale (ASGS) stabilization technique is employed for that purpose [36]. In this case the entire space for the sub-grid scale is taken.



(a) Cloud of nodes.

(b) Mesh of the convex hull obtained with the *Triangle* library.

(c) Domain after applying alpha shape. $\alpha = \alpha_1$.

(d) Domain after applying alpha shape. $\alpha_2 > \alpha_1$.

Figure 2.4: Possible boundaries of a cloud of nodes using alpha shapes method. Image taken from [1]

Bossak time integration scheme

A Bossak time integration scheme is used to advance in time the momentum equations due to its presents some implementation advantages for non-linear problems. For a more detail about this topic see [1].

2.3.4.4 The algorithm

Finally the algorithm of PFEM is presented:

Algorithm 2 PFEM algorithm

1: Imposition of mesh velocity at time step n. $\mathbf{u}_{sM} = \mathbf{u}^n$.
2: Laplacian smoothing (free surface kept fixed). It is a geometrical technique that allows a more homogenous redistribution of the nodes inside the analysis domain without changing the connectivities between nodes.
3: Remesh.
4: Solve the monolithic system.
5: Back to step 1.

2.3.5 PFEM-2 method

Despite the advantages of the PFEM method, limitations on 3D models of the current analyzed problem were observed due to the remeshing process [1]. Therefore, a new strategy to evaluate the structural response of the dam has been chosen, which is PFEM-2. It is based on the use of Lagrangian particles with no connectivities to convect all the material properties combined with a fixed mesh, leading to a Lagrangian formulation without the need of deforming the spatial mesh nor remeshing [7]. The principal features are presented as follow:

2.3.5.1 Circumventing the difficulties of convective transport

The strategy is based on a mixed Eulerian- Lagrangian framework, combining advantages of both methods. This is achieved by using a set of particles combined with a fixed FEM mesh. The main steps of the method are obtained as follows. For a given Lagragian particle p and integrating across the streamlines, the position and the velocity at a given time step $n+1$ are defined as

$$\begin{aligned} \mathbf{x}_p^{n+1} &= \mathbf{x}_p^n + \int_n^{n+1} \mathbf{V}^t dt \\ \mathbf{V}_p^{n+1} &= \mathbf{V}_p^n + \int_n^{n+1} \mathbf{a}^t dt \end{aligned} \tag{2.34}$$

where $\mathbf{V}$ is the velocity, $\mathbf{a}$ is the acceleration, and the superscript t indicates that both variables have to be evaluated continuously in time.

Equation 2.34 cannot be solved directly in the way they are presented. On one hand they are linked together since once is the derivative of the other. The convection of the particles depends on the velocity at every moment, while the forces (accelerations) of the momentum equation depend on the position of each of the particles. On the other hand, the need of a time discretization to solve the equation implies that the variables are not known for any arbitrary time t, but only at discrete time steps $n-1$, n, $n+1$, etc.

For the convection of the material points, these issues can be assessed by using an explicit strategy in 2.48a, uncoupling it from the momentum equations.

$$\mathbf{x}_p^{n+1} \approx \mathbf{x}_p^n + \int_n^{n+1} \mathbf{V}^n(\mathbf{x}_p^t) dt \tag{2.35}$$

On the other hand, the strategy applied for the momentum equation 2.48b is fully implicit and evaluated only at the final configuration, rather than across the streamlines . Then, the momentum equation is expressed as:

$$\mathbf{V}_p^{n+1} \approx \mathbf{V}_p^n + \mathbf{a}^{n+1}\Delta t \tag{2.36}$$

In order to solve the acceleration a fixed mesh is used.

Combining the explicit contribution of equation with the implicit term of equation, the general algorithm of the PFEM-2 with implicit correction is obtained. This simplifies the implementation of the method since the implicit terms are added directly in the final configuration. Then the final set of equations including the convection reads:

$$\begin{aligned} \mathbf{x}_p^{n+1} &\approx \mathbf{x}_p^n + \int_n^{n+1} \mathbf{V}^n(\mathbf{x}_p^t)dt \\ \mathbf{V}_p^{n+1} &\approx \mathbf{V}_p^n + \mathbf{a}^{n+1}\Delta t \end{aligned} \tag{2.37}$$

With only explicit contributions across the streamline integration, it is possible to devise a strategy with two distinct stages. First, the explicit streamline integration is performed. Having done this, results are projected into a fixed mesh to solve the implicit system of momentum equations and finally the corrections are passed to the Lagrangian material points. Then the complete scheme required to solve a step becomes:

Convect particles $\Longrightarrow$ Project information $\Longrightarrow$ solve equations (mesh stage) $\Longrightarrow$ Update particles.

The particles used in this scheme do not represent a fixed amount of mass, but rather material points with certain properties and velocity. This allows for a variable number of particles per element depending on the zone, to ensure a better accuracy on selected areas. It must be noted that in the algorithm implemented, the particles are only used to transport information (solve the convective term), meaning that the forces will not be integrated across the streamlines. Figure 2.7 shows the velocity streamline integration for a single particle.

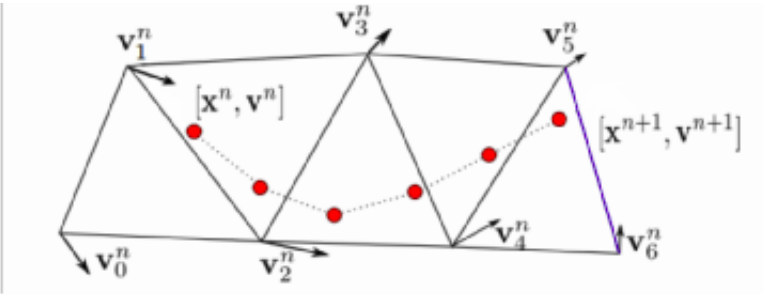


Figure 2.5: PFEM-2 streamline integration. Image taken from [7]

Once all the particles have been convected, information is projected into a mesh. The interface is drawn by determining the exact place where the material properties change. This is defined in the mesh by an auxiliary pseudo-level set scalar function φ that is zero at these lines (2D) or surfaces (3D).

A correct location of the interface is critical, since the finite element mesh is enriched at the interface as it will explained later. The projection kernel to transfer the information from the particles to the mesh is a critical part of the strategy. Since no information is stored in the mesh, this kernel must be accurate enough as to guarantee that the projected field of the variables resembles the field of the original domain (the particles).

Having detected the interface, the system of equation is assembled and solved implicitly. Finally, once the full system of equations has been solved, corrections are passed to the particles and the cycle is restarted. The basic ingredient of PFEM is the fixed mesh domain.

2.3.5.2 Fixed mesh domain

The chosen spatial discretization procedure is the finite element method (FEM). It consists on dividing the domain into elements, whose geometry is defined by nodes. In order to solve the Lagrangian equations in the mesh, first it is necessary to transfer the information from the particles. To do so, a projection algorithm is used. The simplest projection kernel consists on directly using the shape functions of the elements.

The unknown variables are the values at the nodes and the solution is interpolated from the nodal values inside each element. In this numerical approach liner shape functions will be used for all the variables.

The equations to solve the problem are Lagrangian, therefore are equal to the ones explained in section 2.3.4.3 for PFEM. The only difference is that the free surface now is in the middle of the elements. The stabilization used is a OSS (Orthogonal Subgrid Scale) [27].

2.3.5.3 Special considerations on the interface

PFEM-2 was developed to solve multi-fluids, meaning that it is not able to solve single phase free surface problem as the case of PFEM. In the current work the two fluids are the granular material and the air. Since these phases have a density ratio larger than 1000, special care must be taken when computing the pressure using a fixed Finite Element mesh with moving interfaces.

Standard FE formulations only allow for continuous interpolations of the variables inside the elements. However discontinuities in the material properties can cause jumps in the gradients of the variables and even complete discontinuities. To overcome this issue the Finite Element space must be modified to allow for a more accurate reproduction of the solution. A first alternative would be remeshing the zone. However, this requires new degrees of freedom in the new nodes and new connectivites, which is computationally expensive. For that reason, the alternative chosen is an enrichment of the FE space in the pressure field. It is able to grasp this phenomenon in an elementary level. Hence, the global system does not change the number of unknowns.

For this cases in which the material discontinuities only produce a kink in the gradients of the unknowns, it is necessary to add new shape functions that are continuous in the variables but discontinuous in the gradient. The function used is the one proposed By Coppola-Owen and Codina [104] (Figure 2.58). This function was developed to improve the pressure field in two-fluid problems with low viscosities.

This enrichment function must be used together with the standard linear interpolation functions of the elements. The combination of these two fields can capture a smooth field with a sharp change in the slope at the location of the interface.

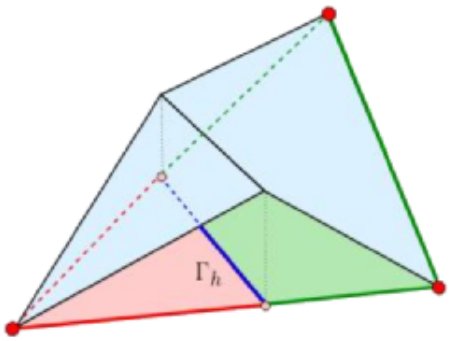


Figure 2.6: Enrichments for discontinuous gradient. Image taken from [7]

2.3.5.4 The algorithm

Finally the algorithm of PFEM-2 proposed by Becker and co-workers [7, 8] is presented:

Algorithm 3 PFEM-2 algorithm

1: Convect the particles using the streamlines.
2: Project information on the mesh.
3: Detect the interface elements.
4: Assemble the system of equations.
5: Solve the system of equations.
6: Recover condensed DoFs.
7: Update the velocity.
8: Back to step 1.

2.3.6 Summarizing the main characteristics of PFEM and PFEM-2

The main characteristics of both coupled strategies are presented as follow:

	PFEM	PFEM-2
Framework	Updated Lagrangian formulation	Mixed Eulerian-Lagrangian formulation. Eulerian for the nodes, Lagrangian for the particles.
Pressure/Velocity interpolation elements	P1/P1 piecewise linear elements	P1/P1 piecewise linear elements plus enrichment shape functions for the pressure
Mesh	Need a fast remeshing algorithm	No need remeshing algorithm (fixed mesh). A projection/interpolation algorithm between particle and nodes is needed instead.
Structural constitutive law	Variable yield visco-rigid model	Variable yield visco-rigid model
Approach	Monolithic	Monolithic
Stabilization technique	ASGS(Algebraic Sub-Grid scale)	OSS (Orthogonal Subgrid Scale)
Solution strategy	Residual based strategy	Residual based strategy
Interface/boundary recognition method	Alpha shape method	Pseudo Level set function
Time scheme	Bossak time	Backward Euler
Boundary conditions	Inlet velocity, definition of the boundaries and free surfaces	Inlet velocity, definition of the boundaries and free surfaces
Principal parameters	Internal friction angle, solid density and a bounding box	Internal friction angle, solid density and a point with fixed pressure
Type of problem	Single material (Air is not simulated)	Two Materials (Air is simulated)
Parallelization	The computation on each particle cannot be parallelized	The computation on each particle can be parallelized

Table 2.1: Main characteristics of PFEM and PFEM-2 methods

2.4 The Coupled Problem

The following sections are devoted to the presentation of the coupling strategies. Firstly the existing strategy that couples the free surface/seepage Eulerian code with PFEM, developed by Larese, is presented. Secondly, the newly developed strategy that coupled the fluid code with PFEM2 is described.

2.4.1 CFD-PFEM strategy

Two models are created in order to simulated correctly the behavior of rockfill dams when an overtopping occurs. On one hand the fluid model is defined only with the control domain and with the inflow discharge. On the other hand, the structural model is defined only with the granular non-cohesive material domain and its mechanical properties. It is important to note that the meshes are different in both models.

An explicit staggered coupling is used. The structural Lagrangian model is mapped on the Eulerian fixed mesh domain where, at the beginning of the simulation, the only available information is the inflow discharge and the control domain. The idea is that the fluid analysis step is evaluated once the distribution of porosity is mapped from the structural domain. The solution of the fluid problems is then mapped on the Lagrangian structural mesh. It is necessary to know the fluid pressure and the Darcy forces in order to evaluate correctly the external forces term of the momentum equation. Once this is done, the structural response can be calculated. Therefore, the granular domain deforms accordingly to the obtained velocity and pressure fields. This new deformed granular domain is finally mapped onto the Eulerian mesh in order to solve for the subsequent time step.

An element transfer method (EMT) is implement for the mapping of the variables between the Eulerian fixed and the PFEM moving meshes. For more detail see [1,45]. The searching algorithm developed uses a kd-tree data structure. The flow chart of the algorithm is presented as follow:

Algorithm 4 Coupling algorithm

1: Map the configuration of the rockfill material in terms of Porosity and diameter distribution on the Eulerian fluid domain.
2: Calculate the free surface and seepage problem.
3: Map the water pressure fields on the Lagrangian structural mesh.
4: Map the non linear Darcy term on the Lagrangian structural mesh.
5: Calculate the structural response in a Lagrangian mesh using PFEM.
6: Back to step 1.

Figure 2.7 shows graphically the summary of the coupled CFD-PFEM strategy.

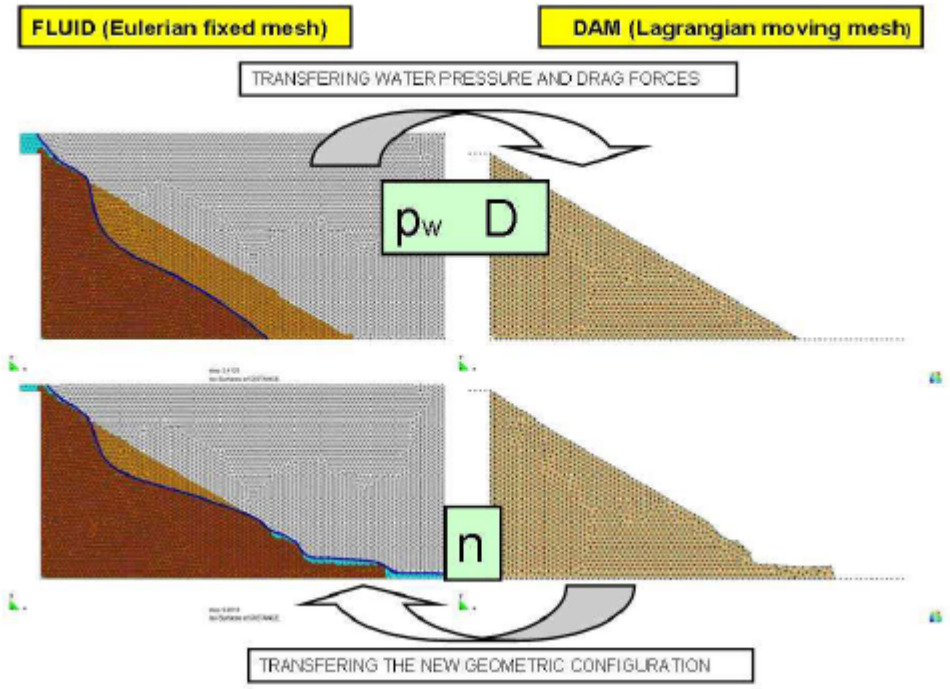


Figure 2.7: Graphical summary of the coupled CFD-PFEM strategy. Image taken from [1]

2.4.2 CFD-PFEM-2 strategy

The CFD-PFEM-2 strategy has been developed in the current work.

Similarly to the coupled CFD-PFEM strategy two models are created in order to simulated the analyzed problem. However, for this coupled strategy the fluid model is defined with the entire information of the granular non-cohesive material (porosity, diameter, internal friction angle) at the beginning. On the other hand, the structural model is defined using the same control domain and mesh that the fluid problem with the purpose to simplify the data transmission between both models.

An explicit staggered coupling is used. The fluid model has the information of porous material, the inflow discharge and the control domain. Then, the fluid is evaluated and the solution of the fluid problem is copied on the PFEM-2 structural mesh. It is necessary to know the fluid pressure and the Darcy forces in order to evaluate correctly the external forces term of the momentum equation. Once this is done, the structural response can be calculated. Therefore, the granular domain deforms accordingly to the obtained velocity and pressure fields. This new deformed granular domain is finally copied onto the Eulerian mesh in order to solve for the subsequent time step.

The flow chart of the algorithm is presented as follow:

Algorithm 5 Coupling algorithm

1: Calculate the fluid problem.
2: Copy the fluid velocity and pressure fields from the Eulerian fixed mesh to the PFEM-2 structural fixed mesh (these two meshes are matching).
3: Copy the non linear Darcy term to the PFEM-2 structural fixed mesh.
4: Calculate the structural response in a fixed mesh using PFEM-2.
5: Copy the information in terms of porosity to the Eulerian fixed mesh.
6: Back to step 1.

Figure 2.8 shows graphically the summary of the coupled CFD-PFEM strategy.

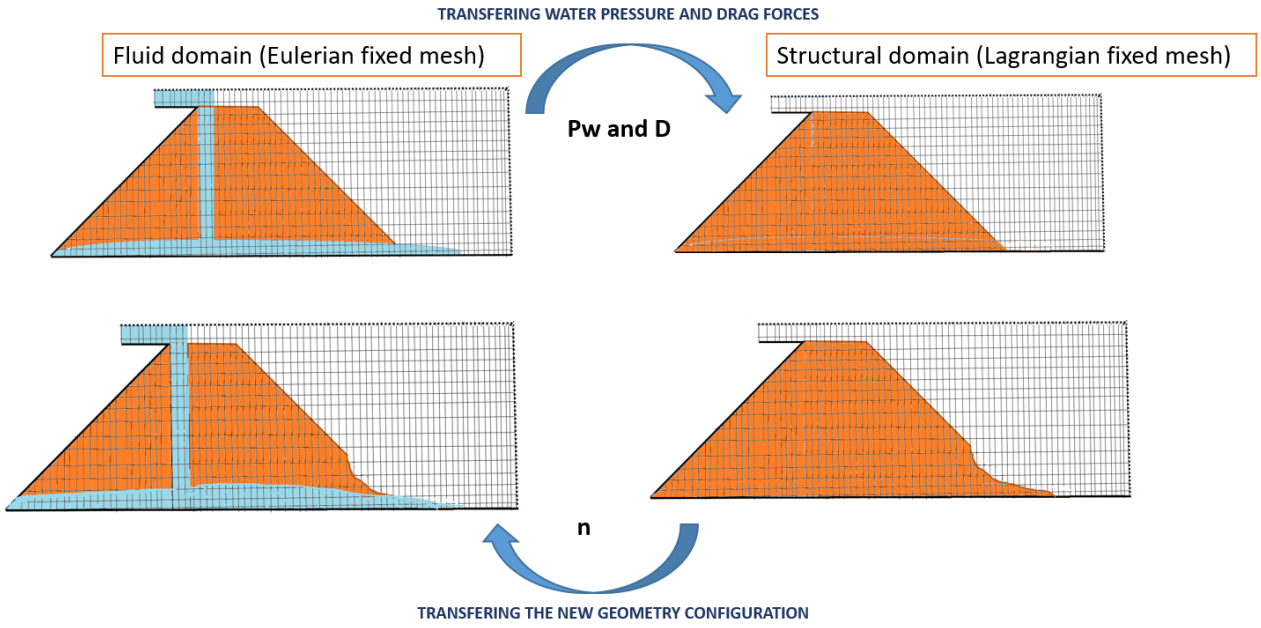


Figure 2.8: Graphical summary of the coupled CFD-PFEM-2 strategy

CHAPTER 3

THE METHODOLOGY

3.1 Introduction

In this chapter, the methodology followed to achieve the objectives stated in Chapter 1 is explained. Two principal stages are differentiated. The first stage presents the assessment of the existing CDF-PFEM coupled model, where the main idea is to be familiarize with the code and with the tools used for the simulation. The second stage presents the development of the CDF-PFEM-2 coupled strategy and its subsequent evaluation.

3.2 Existing CFD-PFEM coupled model assessment

The existing code presented in section 2.3.1 has been assessed reproducing a set of 2D small-scale dam models with and without failure. Moreover, a set of simple examples (still rockfill dam, and settlement of a vertical rockfill dam) have been reproduced with the purpose to study the behavior of the PFEM strategy separately. The Kratos Multiphysics [16] and GID [15] programs developed by CIMNE are used for the modeling and visualization of the numerical examples.

Kratos is a framework for building multi-disciplinary finite element codes as well as a common platform for natural interaction of these modules in different ways. It is written in C++ language. It uses the facilities of Python language [17] for IO data transmission.

GID is a universal, adaptive and user-friendly pre and post processor for numerical simulations in science and engineering.

Integration of any solver inside GID can be carried out in an easy way. The data required by the solver must be specified to GID in an xml file, and GID will automatically create the corresponding windows and the graphical tree containing all the information useful for the user of the simulation code. This is called "The problem type" in GID. During the calculation, the solver can send information to GID in order to update its status, and at the end the results are passed to GID in order to be post-processed and visualized. For more detail see Appendix A.

The coupled CFD-PFEM strategy aims to simulate the seepage line and the overtopping flow

while following the evolution of the breach in the dam material. Two models are generated for this strategy:
- The fluid Eulerian model. The control volume is built large enough in order to not influence the solution. This model is characterize by the absence of any porous material. This information is passed during the calculation, by the PFEM model.

- The PFEM structural model. The dam is constructed in a Lagrangian framework. This implies modeling only the material domain (i.e free surface and walls). The definition of a bounding box is required. It sets the maximum calculation domain. If a node exits the bounding box is no longer calculated and is deleted.

Finally, the numerical results obtained were compared with the experimental values in order to evaluate the correctness simulation of the models.

3.3 Development of the CFD-PFEM-2 coupled strategy

Once the existing methodology (CFD-PFEM and PFEM-2) have been used, verified and understood then the implementation of the coupled strategy between PFEM-2 and Free Surface/Seepage CFD has been performed using the programming language Python.

As is mentioned in Chapter 2, one of the main features of PFEM-2 is that, unlike the PFEM method, it uses a fixed mesh. This property was used in this work to simplify the coupled algorithm because the remeshing process is avoided. This is the reason why lack of accuracy and expensive computational cost in 3D small scale dam models has been observed using the CFD-PFEM strategy [1, 5, 6].

Due to the fluid domain and the domain of the granular material occupy a similar volume calculation we will use for PFEM-2 the same mesh that was generated for the fluid. This greatly simplifies the transmission of information between the two domains, since the positions of the nodes and elements coincide exactly. It should be noted that both modules (PFEM-2 and fluid) used the velocity and pressure as main variables for the calculation. For this reason, two independent domains (models) are used. Otherwise one solver would overwrite the information of the other. Coupling variables will be transfer between domains only when necessary, using a utility transmission node to node (node mapper). The main parts of the code can be summarize as follow:

1 part. The path of Kratos is indicated. Moreover, the libraries, applications and solvers to be used are imported.

2 part. The model part (nodes and elements), problem settings and nodal conditions are read.

3 part. The time step for the fluid and structural problem are calculated.

4 part. The algorithm to solve the coupled problem is presented:

Step 1 Solve the water free surface flow problem calculating the VELOCITY and PRESSURE field in a Eulerian framework using the edgebased level set solver.

Step 2 Copy the fluid velocity and pressure fields on the structural fixed mesh using the function *"mapper.ScalarMap"*.

Step 3 Copy the non linear Darcy term on the structural fixed mesh using the function *"mapper.ScalarMap"*.

Step 4 Calculate the STRUCTURAL RESPONSE in a fixed mesh using the PFEM-2 solver ($pfem\_2\_solver\_monolithic\_fluid$).

Step 5 Copy the configuration of the rockfill material in terms of porosity distribution on the Eulerian fluid domain using the function *mapper.InverseScalarMap*.

Step 6 Return to step 1 until the maximum time is reached.

5 part. Prints the results calculated.

The newly developed coupled strategy will be used in the next chapters together with the existing PFEM-CFD strategy to reproduce a series of examples and experimental test. This will allow compare the two strategy and evaluate whether PFEM2 is suitable for the considered coupled problem. The complete code can be seen in Appendix B.

CHAPTER 4

NUMERICAL RESULTS AND DISCUSSIONS

A set of 2D and 3D models are performed with the purpose of identifying principally the strengths and weakness of both PFEM and PFEM-2 methods in the simulation of overtopping in rockfill dams. As a first step in the analysis of the fluid and structural models, the author reproduced examples for the uncoupled solvers. The main idea is evaluated the stability of the methods. As a second step the coupled strategies are analyzed using a set of 2D scale-rockfill dams.

4.1 Still rockfill tank

A 2D analysis is carried out. The aim of this simple example is to asses the calculation of the pressure distribution when no velocity is presented for both PFEM and PFEM-2 algorithms.

4.1.1 Geometry

The geometry is presented in Figure 4.1.

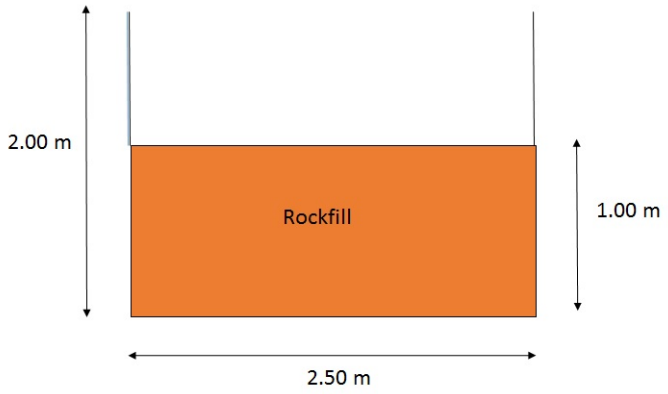


Figure 4.1: Geometry of the still rockfill tank

4.1.2 Parameters

Table 4.1 shows the principal parameters used in the calculation of the solid pressure.

Parameter	Value
Internal friction angle ϕ	30^{o}
Solid density	1490 kg/m^3

Table 4.1: Properties of the rockfill material

4.1.3 Mesh

The mesh used has 338 linear triangular elements (Figure 4.2) for both models.

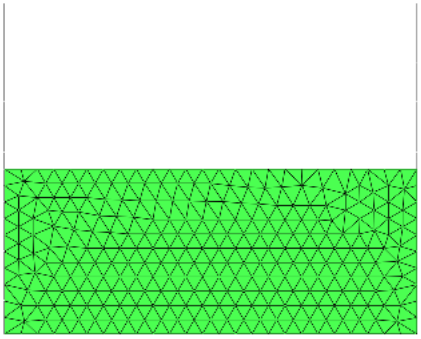


Figure 4.2: Mesh of the still rockfill model

4.1.4 Results

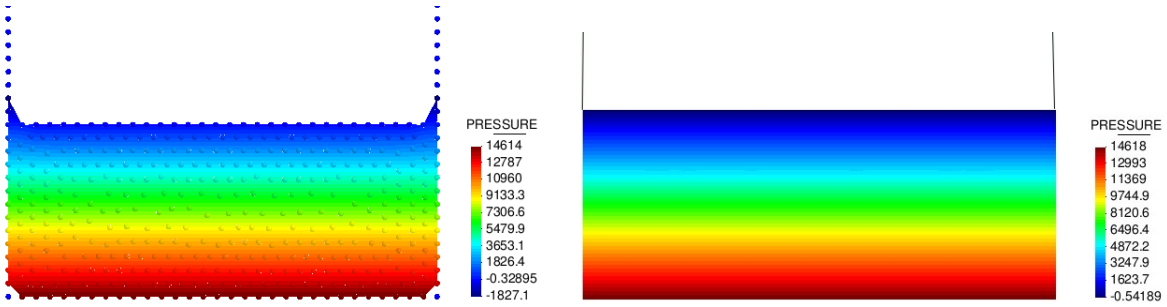
The results of the still rockfill tank example for PFEM and PFEM-2 methods are shown in Figures 4.3a and 4.3b, respectively. It can be seen that both methods have a good performance in the calculation of the pressure distribution when no velocity is presented. Knowing that the pressure distribution is calculated as:

$$P = \rho_s * g * h \tag{4.1}$$

Then the theoretical result for the bottom pressure is :

$$P = 1490 * 9.81 * 1 = 14617 Kg/m^3 \tag{4.2}$$

which is equal to the values in the bottom of the numerical models. Additionally, no pressure variations were observed in time.



(a) PFEM (b) PFEM-2

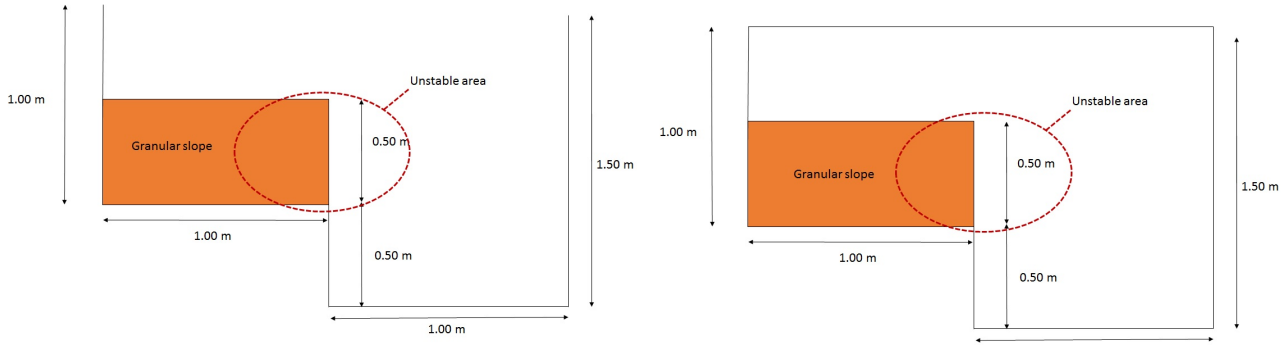
Figure 4.3: Pressure distribution

4.2 Settlement of a vertical rockfill slope

2D and 3D models are performed. This example is used to verify the reproduction of the internal friction angle and the dependency of the stable configuration of the mesh size in both the PFEM and PFEM-2 algorithms. It is important to point out that PFEM-2 has been implemented for multifluids problem. For that reason it cannot calculate only one stage as PFEM (leaving empty spaces). Instead, the air is also calculated.

4.2.1 Geometry

The geometry of PFEM and PFEM-2 is presented in Figures 4.4a and 4.4b, respectively. It can be observed that the PFEM-2 model needs a close domain in order to calculate the air and the granular material.



(a) PFEM model (b) PFEM-2 model

Figure 4.4: Settlement of a vertical slope. Geometry.

4.2.2 Parameters

The parameters on both granular material and air are presented in Table 4.2. Additionally, Table 4.3 indicates the different internal friction angles simulated.

Material	Density Kg/m^3	Viscocity (Pa s)
Rockfill material	1490	0.01
Air	1	0.0001

Table 4.2: Properties of the rockfill material and air

Case	Internal friction angle ϕ
Case 1	30°
Case 2	35°
Case 3	40°
Case 4	45°

Table 4.3: Internal friction angles

4.2.3 Sensitivity mesh analysis

Let us consider an internal friction angle $\phi = 30^{\circ}$. An analysis of mesh sensitivity has been performed using three different sizes of elements for the simulation:

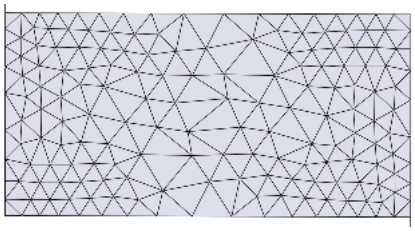
Mesh	Size (m)	N° elements
Mesh 1	0.10	306
Mesh 2	0.05	512
Mesh 3	0.02	2856

Table 4.4: Different mesh sizes taken into account for the current example

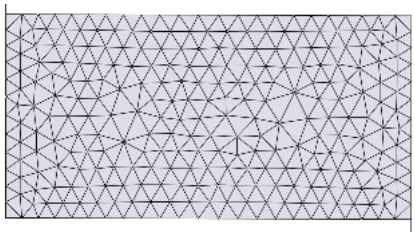
They are shown in Figure 4.5.

4.2.4 Results

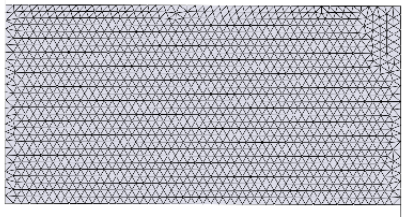
The evolution of the settlement in PFEM and PFEM-2 is shown in Figure 4.6 for the above mentioned meshes . As expected, the more accurate and realistic settlement process is obtained with the finer mesh but not relevant differences in the angle are observed in the courser meshes. Mesh 3 is used for the simulation of the others cases of frictional internal angles. The results are presented in Figure 4.7. It can be observed that angles $\phi = 35^{\circ}$ and $\phi = 40^{\circ}$ are reproduced exactly in both cases. However, the case of angle $\phi = 45^{\circ}$ using PFEM is closer to the value but not exact. Following the work by Larese [1] this value of angle ϕ is in the limit of validity of the model. Despite, it can be observed that PFEM-2 reproduces the angle correctly. For that reason another internal friction angles is studied, $\phi = 47^{\circ}$. Figure 4.8 shows that PFEM cannot reproduces the angle, the body behaves as rigid. On the other hand, PFEM-2 can reproduces the angle.



(a) Mesh1

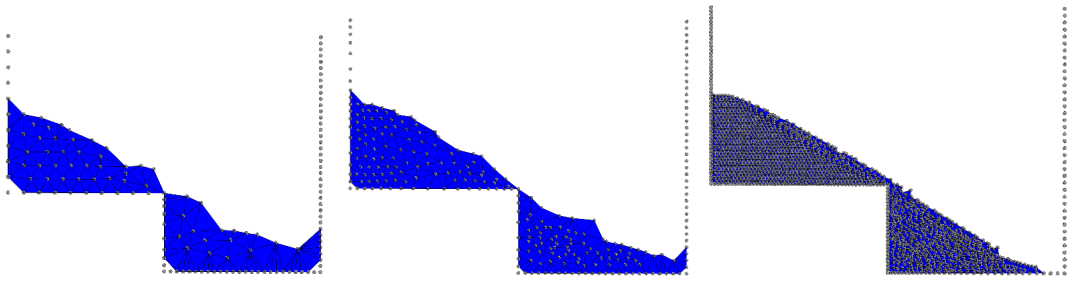


(b) Mesh2

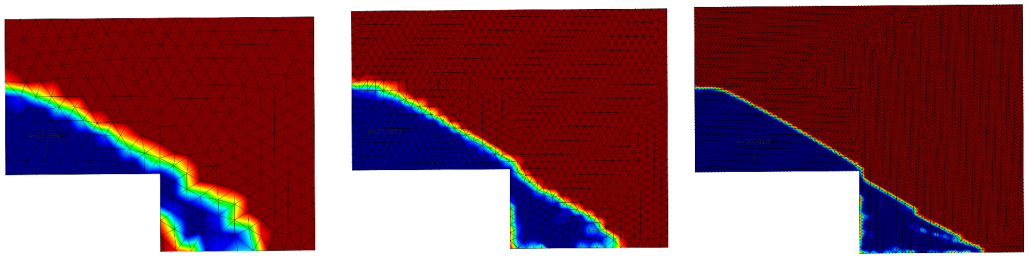


(c) Mesh3

Figure 4.5: Different mesh sizes taken into account for the current example

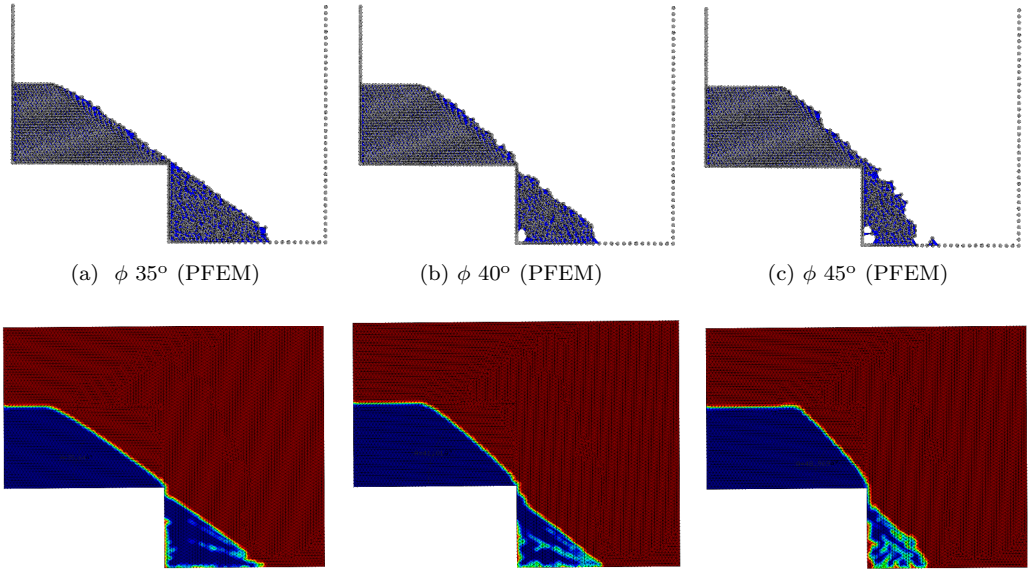


(a) Mesh1 PFEM (b) Mesh2 PFEM (c) Mesh3 PFEM



(d) Mesh1 PFEM-2 (e) Mesh2 PFEM-2 (f) Mesh3 PFEM-2

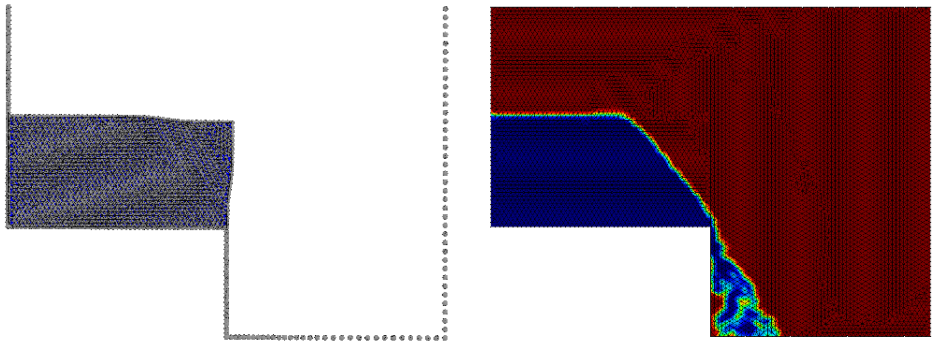
Figure 4.6: Evolution of the settlement for the analyzed meshes



(a) ϕ 35° (PFEM) (b) ϕ 40° (PFEM) (c) ϕ 45° (PFEM)

(d) ϕ 35° (PFEM-2) (e) ϕ 40° (PFEM-2) (f) ϕ 45° (PFEM-2)

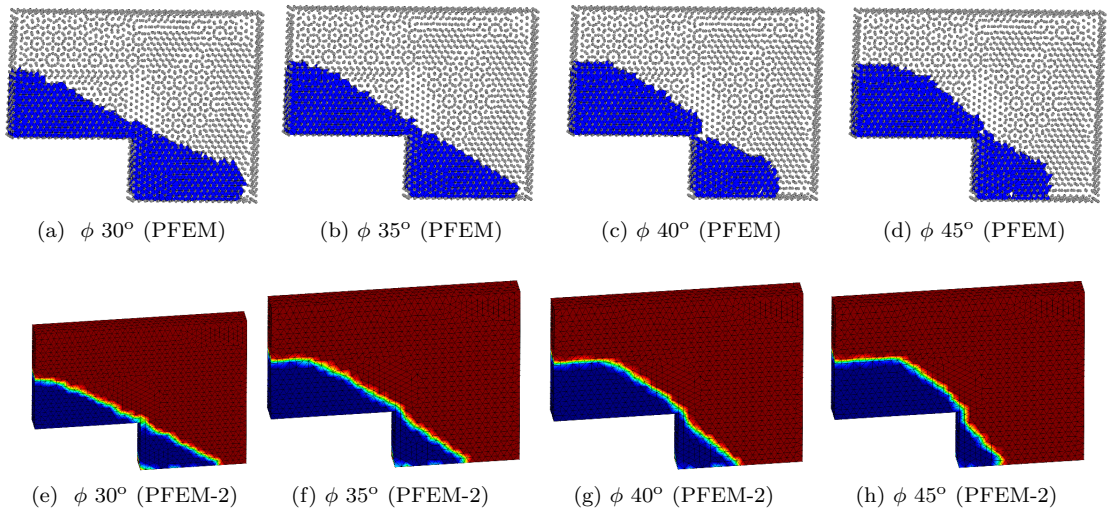
Figure 4.7: Evolution of the settlement for variables internal friction angles for 2D models



(a) ϕ 47° (PFEM) (b) ϕ 47° (PFEM)

Figure 4.8: Evolution of the settlement for the internal friction angle 47°.

The settlement process is also reproduced in 3D models. Figure 4.9 shows the settlement of the analyzed internal friction angles. It can be seen a good agreement between the numerical results and the real values. Only the case of angle $\phi = 45°$ using PFEM is closer to the value but not exact.



(a) ϕ 30° (PFEM) (b) ϕ 35° (PFEM) (c) ϕ 40° (PFEM) (d) ϕ 45° (PFEM)

(e) ϕ 30° (PFEM-2) (f) ϕ 35° (PFEM-2) (g) ϕ 40° (PFEM-2) (h) ϕ 45° (PFEM-2)

Figure 4.9: Evolution of the settlement for variables internal friction angles for 3D models

4.3 Small-scale Rockfill dams during overtopping scenarios

The evolution of seepage and beginning of failure in two different types of dam is simulated: (A) an homogeneous dam, and (B) a dam with an impervious screen. All the dams considered have the same downstream slope 1.5H:1V. This geometrical aspect does not have any influence in the modeling of seepage but strongly determines the deformation of the rockfill. In fact, for this kind of slope mass sliding is the predominant failure. Both strategies CFD-PFEM and CFD-PFEM-2 have been conceived for representing this type of failure.

4.3.1 Parameters

Only one material has been analyzed, its characteristic are summarize in Table 4.5.

Parameter	Symbology	Value
Porosity	n	0.4052
Average diameter	D_{50}	35.04 mm
Dry density	ρ_s	1490 kg/m^3
Saturated density	ρ_{sat}	1910 kg/m^3
Apparent specific weight	W	2500 kg/m^3
Pore index	P_i	0.68
Internal friction angle	ϕ	[37° - 42.5°]

Table 4.5: Properties of rockfill material for small scale rockfill dams

4.3.2 Upstream discharge

The discharge Q (l/s) for every simulated case is shown in Table 4.6.

	Model A Homogeneous dam	Model B Impervious face dam
Without Failure	A.1 $Q = 25.46 l/s$	B.1 $Q = 25.46 l/s$
With	A.2.1 $Q = 51.75 l/s$	B.2.1 $Q = 15.36 l/s$
Failure	A.2.2 $Q = 69.07 l/s$	B.2.2 $Q = 25.05 l/s$
	A.2.3 $Q = 90.68 l/s$	B.2.3 $Q = 30.27 l/s$

Table 4.6: Discharges for the studied cases

4.3.3 Case A: Homogeneous dam

The first example reproduces a dam without any internal core or impervious screen.

4.3.3.1 Geometry

The geometry of the prototype dam is presented in Figure 4.10, where also the distribution of the bottom pressure sensors is indicated.

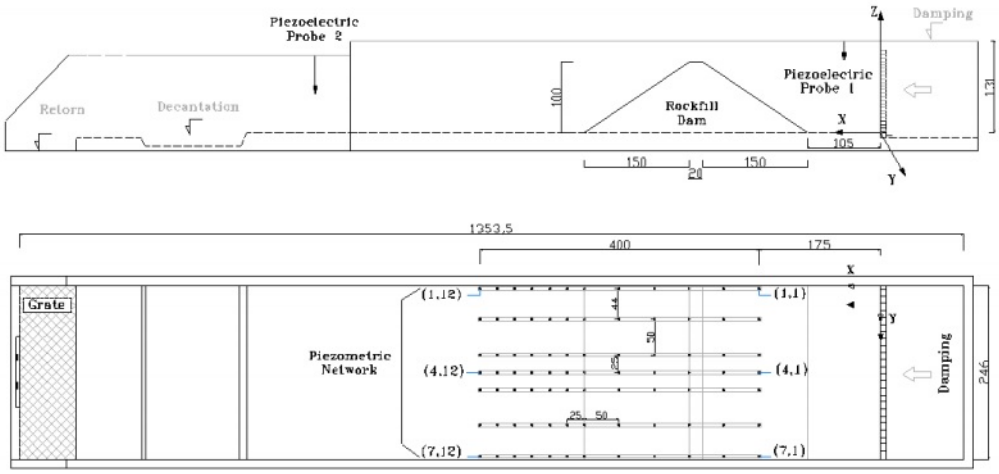


Figure 4.10: Case A: Homogeneous dam. Image taken from [1]

4.3.3.2 Case A1: Seepage analysis

The numerical model is built following the geometry of the experiment. The control volume of the Eulerian fluid model has to be large enough in order not to influence the solution.

Concerning the Dirichlet boundary conditions, an inlet with fixed velocity is set in the left side of the control volume and a slip boundary condition is imposed on the walls as shown in Figure 4.11. Concerning the Neumann boundary a fixed pressure is set on the top and the right side.

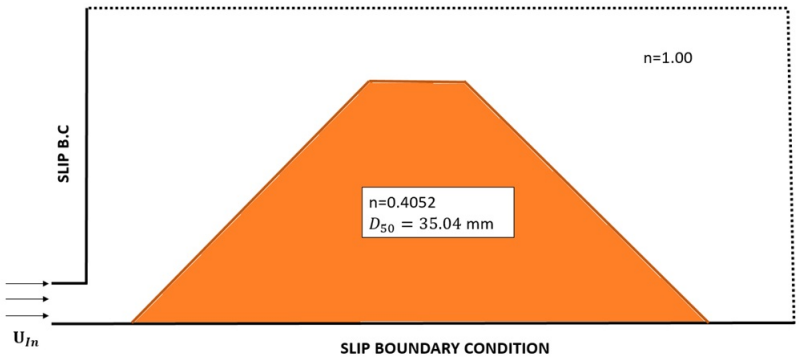


Figure 4.11: Qualitative model geometry and boundary conditions for Case A1

It is important to indicated that no sensitivity analysis is performed for the mesh due to Larese and co-worker demonstrated [1] that the mesh size does not affect relevantly the quality of the results in the current model. The mesh used for the simulation has 16347 linear triangular elements (Figure 4.12).

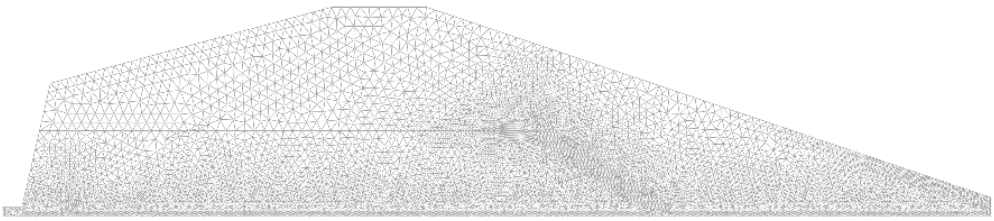


Figure 4.12: Mesh used in the current model.

In Figure 4.13 the comparison between numerical and experimental head of pressure is shown. A good agreement is observed in both experimental and numerical results. However, a lower value of pressure is presented in the numerical results. This is the consequence of the model chosen for the resistance law (Ergun's law).

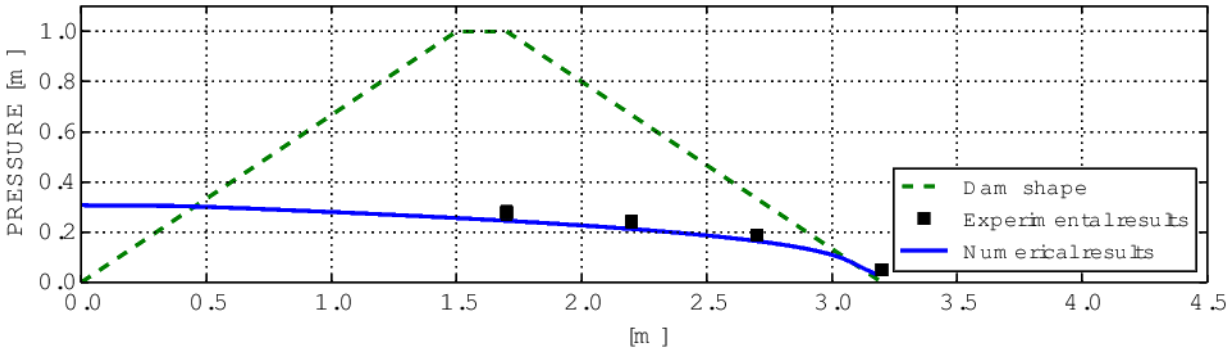


Figure 4.13: Comparison between numerical and experimental bottom pressure in Case A1.

4.3.3.3 Case A2: Failure analysis

The coupled models aim to simulate the seepage line and the overtopping flow while following the evolution of the breach in the dam material. As is explained in chapter 3, two models are generated for the coupled strategies. From now on the mesh used in the discretization is the same as case A1.

PFEM

Figure 4.14 and 4.15 show the Fluid qualitative model and the structural qualitative model of the analyzed problem. It can be observed that the fluid model has the same boundary conditions as case A1. However, the main difference of this model is the absence of any porous material. This information is passed during the calculation, by the PFEM model. On the other hand, it can be seen that the structural model only represented the granular material domain.

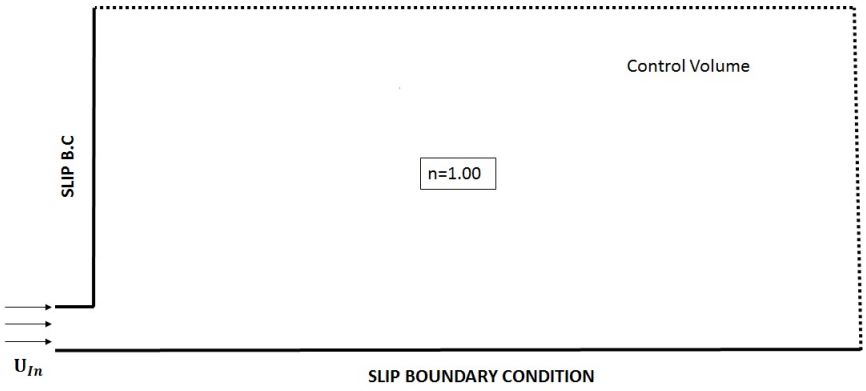


Figure 4.14: Fluid qualitative model for Case A2

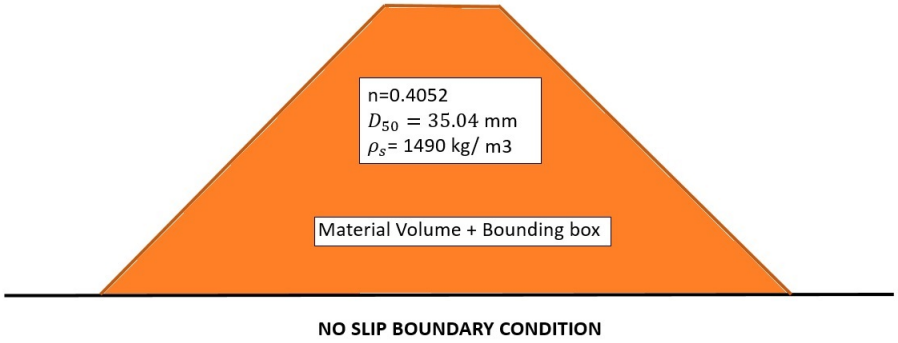


Figure 4.15: Dam qualitative model for Case A2 using PFEM

PFEM-2

As was explained in chapter 3, PFEM-2 used the same mesh as the fluid problem in order to simplify the transfers of information between both models. The construction of the fluid Eulerian model is analogous to the uncoupled Free surface/ Seepage CFD case (Figure 4.11). On the other hand, the structural domain is created cloning the fluid Eulerian model and changing the elements with the purpose to maintain the independence in the variables. Then, a transfers in term of porosity is performed from the Eulerian model to the structural model in order to define the rockfill body (free surface).Additionally, The boundary conditions of PFEM-2 model are presented in Figure 4.16 .

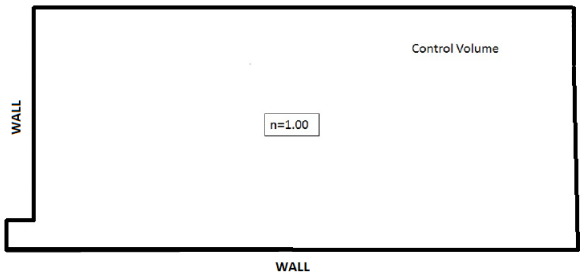


Figure 4.16: Dam qualitative model for Case A2 using PFEM-2

Results

- Case A21

  The simulation of the current analyzed model using PFEM is performed and a good agreement is observed comparing with the experimental results, as was expected. The results will be presented later with the results of PFEM-2.

  A first model was calculated using the CFD-PFEM-2 strategy with the purpose to asses the correctness implementation of the development script. It was seen that instabilities in the pressure field on the structural domain were presented. Hence, a detailed study of the possible causes was carried out. The main causes found were: incorrect input data, and boundary conditions.

Once these problems were solved a good agreement between experimental and numerical results of the bottom pressure were observed. However, the strategy chosen for the relation between the fluid timestep and the solid timestep was not suitable for the computational cost of the simulation. This strategy was $dt_{solid} = 10dt_{fluid}$, which is used in the coupled CFD-PFEM method. A comparison between the computational time of PFEM and PFEM-2 models was performed. The PFEM-2 model spent approximately 10 hours in order to finish the simulation, while the PFEM model spent approximately 2 hours.

Therefore, a new strategy was defined for the CFD-PFEM-2 model. The main idea is that for the initials time steps the relation between fluid and solid can be large ($dt_{solid} = 100dt_{fluid}$) due to the structural domain yet fails. Then, when the initial stage of the failure begins the relation will be the same as the old strategy ($dt_{solid} = 10dt_{fluid}$). This changed allows to decrease the computational time for the current analyzed model approximately. 2 hours and 20 minutes spent the PFEM-2 model in the simulation.

Finally, the results of both strategies are compared again. Figure 4.17 shows a good agreement between the numerical results and the experimental ones for both strategies.

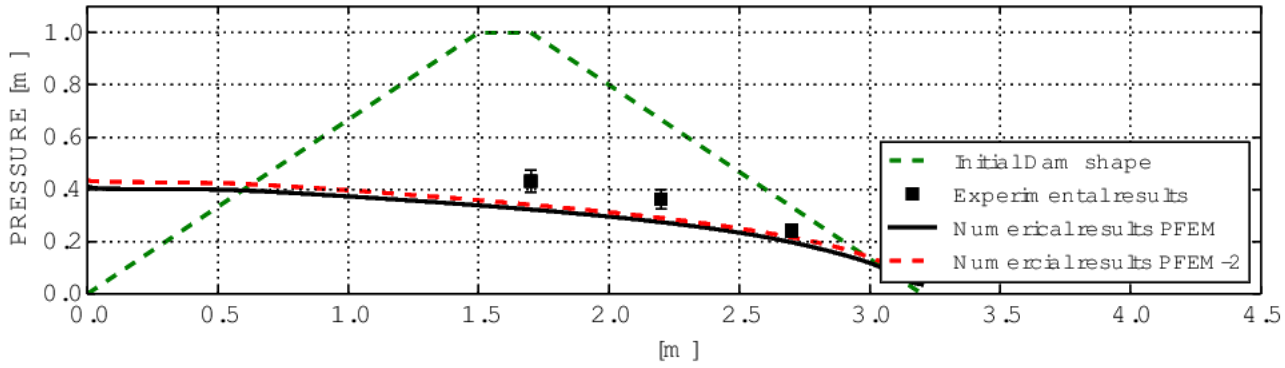


Figure 4.17: Comparison between numerical and experimental bottom pressure in Case A21 using PFEM and PFEM-2

Additionally, the advance of failure in the dam body using PEM-2 is presented in Figure 4.18. The experimental advance of failure "B" is by definition the horizontal projection of the position of the higher particles that moves. This movement is not quantified in the present work. Instead it was assumed that a particle is to be considered "moved" if its total displacement is higher than the average dimension of the granular material (3 cm). The contour fill indicates the displacement of the domain in meters (Figure XX).

Table 4.7 presents the values of the length of failure. The relative error between PFEM-2 and the experimental values is lower than 0.4%. On the contrary, the relative error between PFEM and the experimental length "B" is approximately 4%.

It is important to point out that the internal friction angle used is $\phi = 38^{o}$ due to was the most suitable value for the simulation of the PFEM-2 strategy.

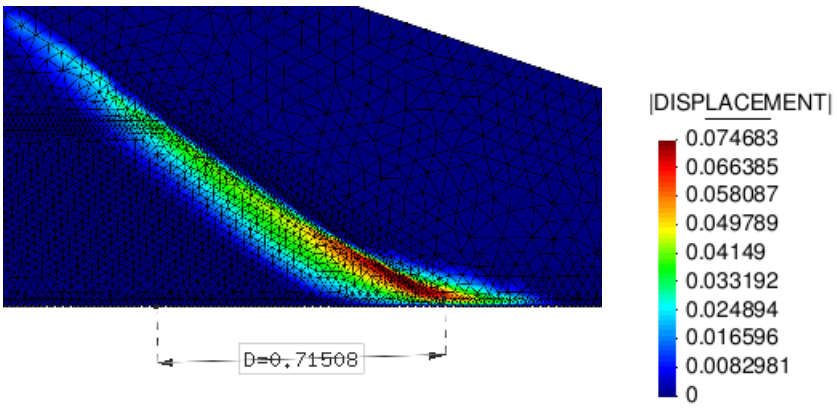


Figure 4.18: Numerical length of failure for Case A21 using PFEM-2

$B_{experimental}$ (m)	$B_{numercial}^{PFEM-2}$ (m)	$B_{numercial}^{PFEM}$
0.7100	0.7150	0.6800

Table 4.7: Comparison between experimental and numerical length of failure in Case A21

- Case A22

  Following the same procedure of Case A21. Figure 4.19 shows the comparison of pressure between the numerical results and the experimental. The computational cost were approximately of 2 hours and 30 minutes for PFEM and approximately 2 hours and fifteen minutes for PFEM-2. The plots of the pressure drop presents a good agreement between the experimental and numerical results. Additionally, it can be observed in Table 4.8 that the advance of failure "B" are closer to the experimental one but lower, that is the body behaves more rigid than the real case. The relative errors is approximately 5 % in both strategies.

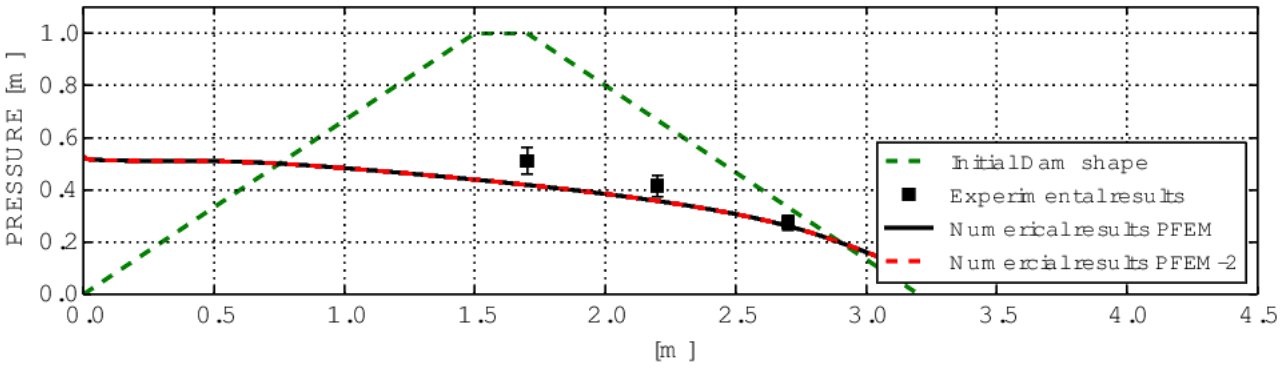


Figure 4.19: Comparison between numerical and experimental bottom pressure in Case A22 using PFEM and PFEM-2 .

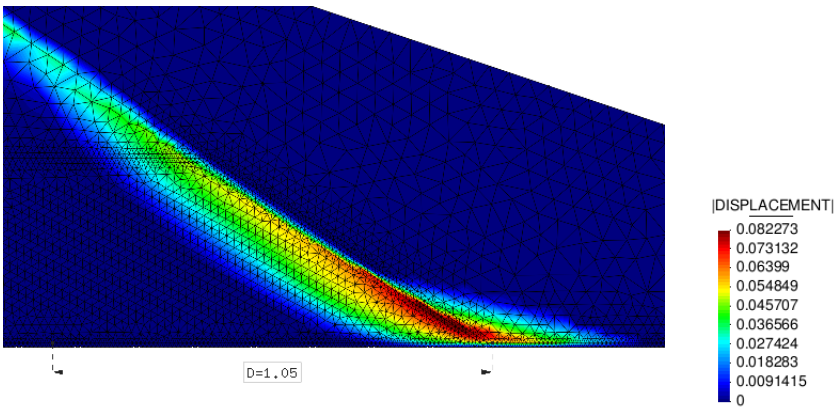


Figure 4.20: Numerical advance of failure for Case A22 using PFEM-2

$B_{experimental}$ (m)	$B^{PFEM-2}_{numercial}$ (m)	$B^{PFEM}_{numercial}$
1.08	1.05	1.04

Table 4.8: Comparison between experimental and numerical advance of failure in Case A22

- Case A23

  Figure 4.21 presents the comparison of the head pressure between the experimental and numerical results. The plots of the pressure drop shows a good agreement between the experimental and numerical results.

  On the other hand, Table 4.9 shows the numerical and experimental advance of failure. It can be observed that the numerical advance calculated by PFEM-2 is lower than the experimental data (relative error approximately equal to 19 %).It is important to point out that a first of PFEM-2 model was performed and instabilities in the structural domain was observed. The body forces increased significantly and for that reason the model collapsed. Hence, a limitation in this values was defined. The top of the module between the body forces in X direction and Y direction could not exceed the value of 20. Once, this changed was implemented, the model behaves stable but with some limitations in the calculation of the advance failure value (Figure 4.22).

$B_{experimental}$ (m)	$B^{PFEM-2}_{numercial}$ (m)	$B^{PFEM}_{numercial}$
1.56	1.25	1.58

Table 4.9: Comparison between experimental and numerical length of failure in Case A23

4.3.4 Case B: a dam with impervious screen

4.3.4.1 Geometry

The details of the geometry of the experimental setting can be seen in Figure N 4.23 , where the pressure sensors distribution are also shown. The red rectangles indicate the three lines

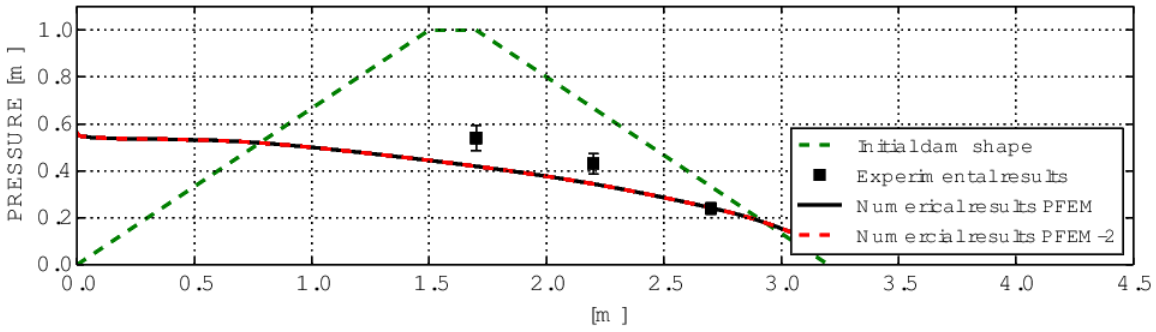


Figure 4.21: Comparison between numerical and experimental bottom pressure in Case A23 using PFEM and PFEM-2

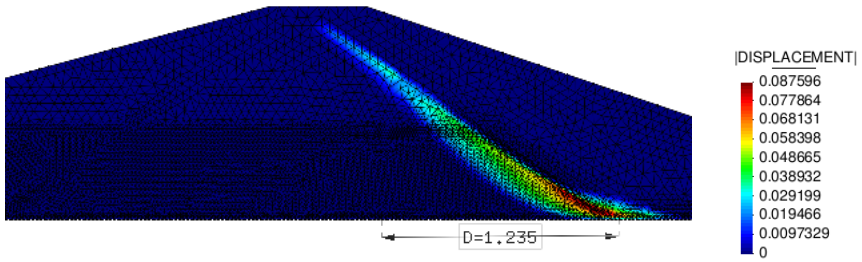


Figure 4.22: Numerical length of failure for Case A23 using PFEM-2

of sensors activated. They are respectively at $Y = 0.3m$, $Y = 0.5m$, and $Y = 0.7m$. The comparison is performed using the first line of sensor ($Y = 0.3m$).

4.3.4.2 Case B1: Seepage analysis

In order to optimize the computational domain, the upstream reservoir is not simulated and the entrance of water is directly set in the upper left part, at the crest level, as shown in Figure 4.24. The upstream screen is therefore considered perfectly impermeable and it is simulated as a rigid wall with a slip condition.
Following the work by Larese [1], the minimum element size that yields a correct conservation of the fluid volume need to be equal o lower than 0.015 m. For that reason, in the area where jet falls the size used is 0.015 m. The rest domain has element of 0.03 m. (Figure 4.25)

In Figure 4.26 the comparison between numerical and experimental head of pressure is shown. The agreement is good even if the numerical code underestimated the experimental values. This is the consequence of the model chosen for the resistance law (Ergun's law).

4.3.4.3 Case B2: Failure analysis

The mesh used in the discretization is equal to the case B1.

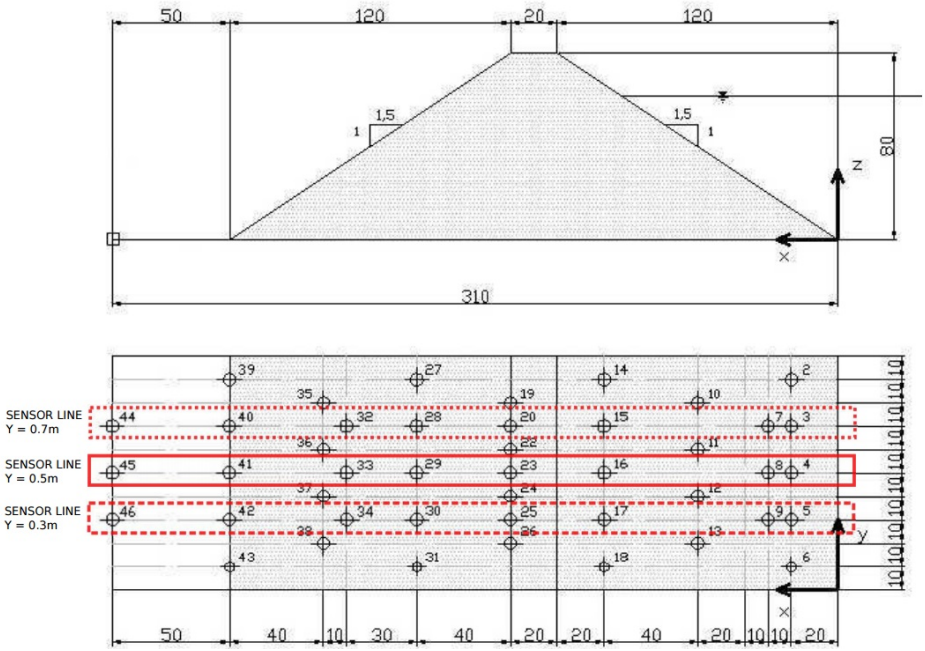


Figure 4.23: Model C: a dam with impervious screen

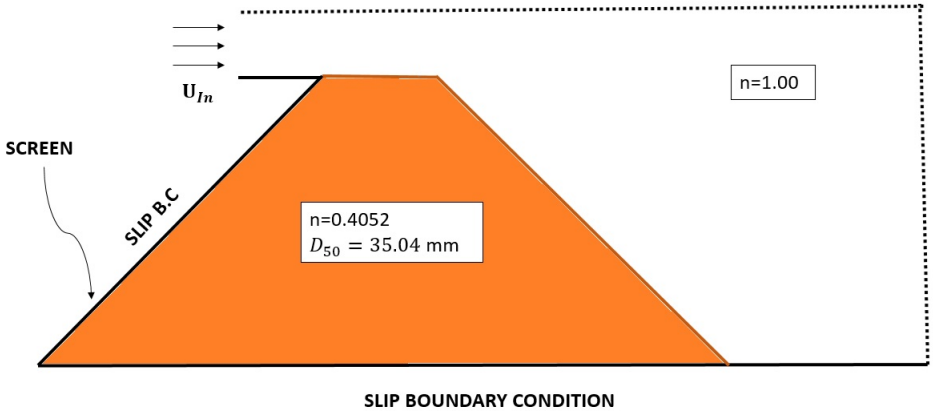


Figure 4.24: Qualitative model geometry and boundary conditions for case B1

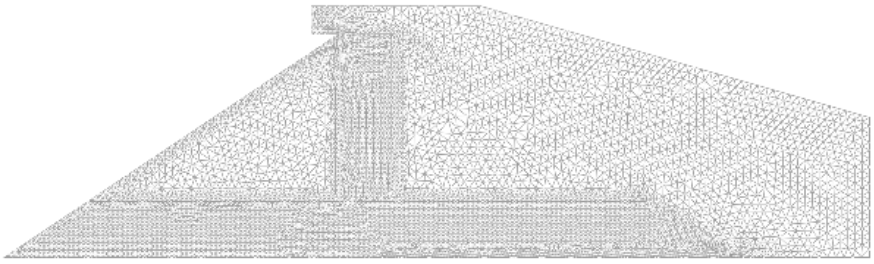


Figure 4.25: Mesh used in case B: a dam with impervious screen

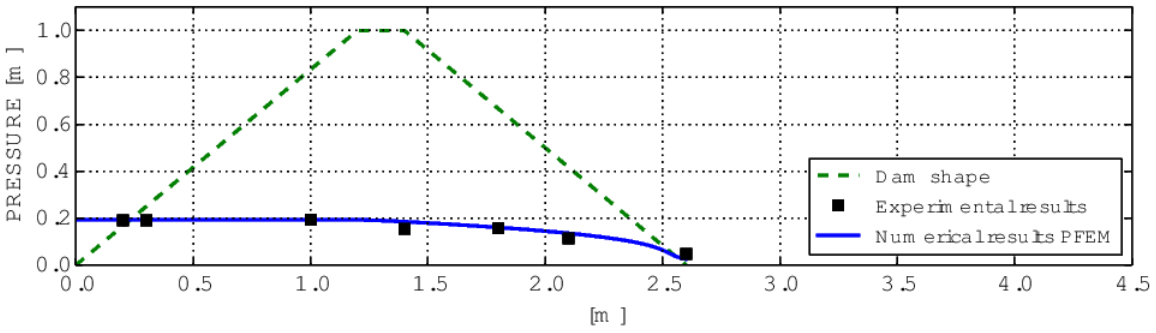


Figure 4.26: Comparison between numerical and experimental bottom pressures

PFEM

It is composes of two parts:

Figure 4.27 and 4.28 show the Fluid qualitative model of the analysed dam and the structural qualitative model for the analyzed model. It can be observed that the fluid model has the same boundary conditions as case B21. Being the principal difference the absence of any porous material. Moreover, it can be observed that the structural model only defined the granular material domain.

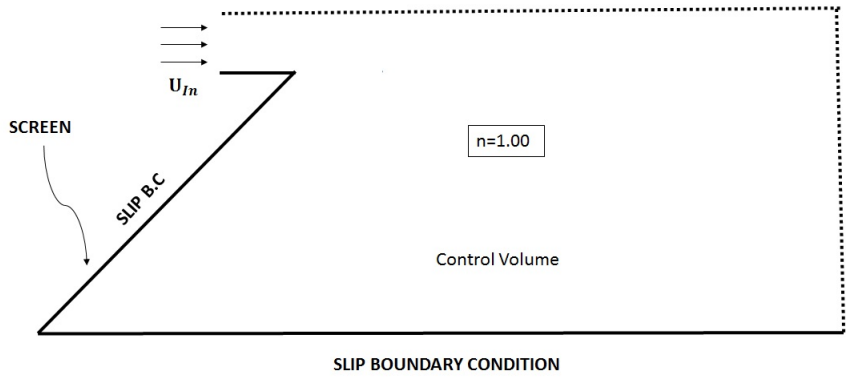


Figure 4.27: Fluid qualitative model for Case B2

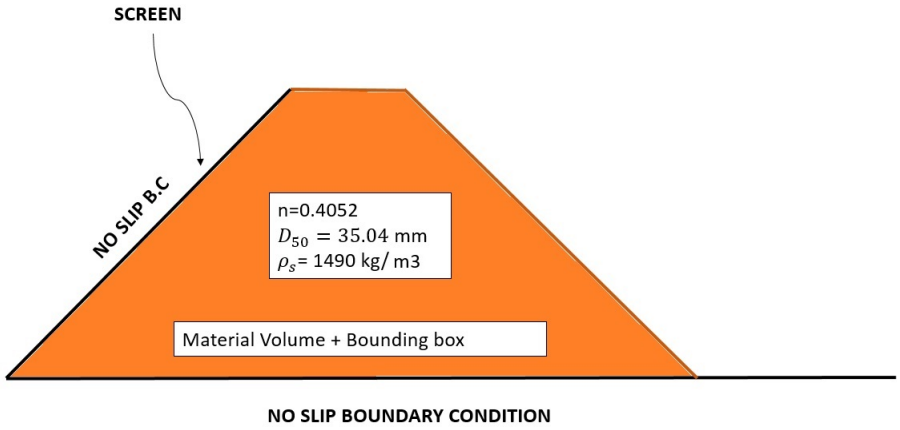


Figure 4.28: Dam qualitative model for Case B2

PFEM-2

The construction of the fluid Eulerian model is analogous to the case B21. Figure 4.27 shows the fluid qualitative model of the analyzed dam. The structural domain is created copied the fluid Eulerian model and changing the element type. No porous material is observed due to this value is transferred from the fluid domain in the calculation (Figure 4.29).

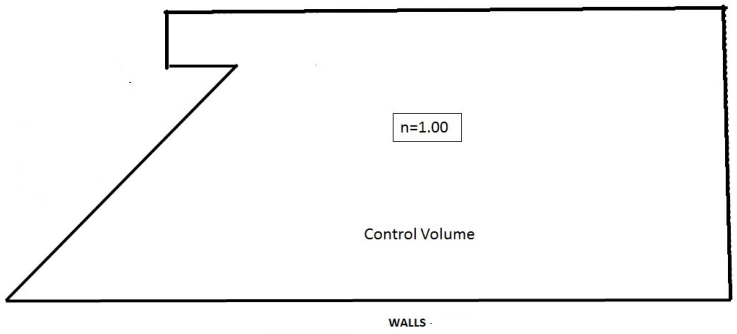


Figure 4.29: Dam qualitative model for case B2

Results

- Case B21

  Figure 4.30 shows the comparison between the numerical and experimental head of pressure. It can be observed a good agreement for both strategies. As is expected, the values are lower than the experimental one. Table 4.10 presents the advance of failure for both methods. The advance "B" obtained with PFEM-2 has a good accordance with the experimental. On the other hand, following the work by Larese [1] for this case the PFEM method not reproduced correctly the length but an improvement are observed with the other incoming discharge.

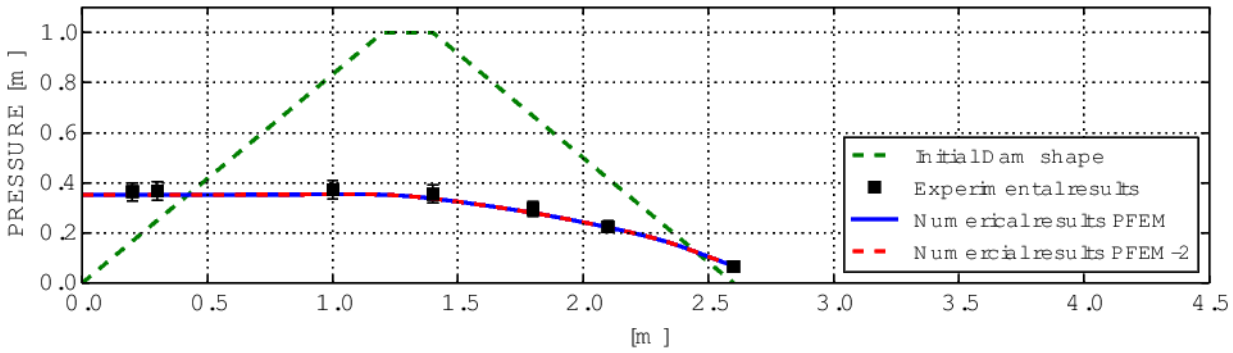


Figure 4.30: Comparison between numerical and experimental bottom pressure in Case B21 using PFEM-2 and PFEM strategy

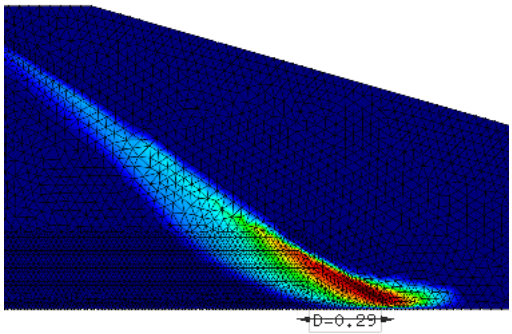


Figure 4.31: Numerical length of failure for case B21 using PFEM-2.

$B_{experimental}$ (m)	$B^{PFEM-2}_{numercial}$ (m)	$B^{PFEM}_{numercial}$
0.2400	0.2900	0.5800

Table 4.10: Comparison between experimental and numerical length of failure in case B21

It is important to stress that for the current case the internal friction angle $\phi = 40^o$ was used for the simulation because is the most suitable value obtained after a sensitivity analysis.

- Case B22

  Figure 4.32 shows the comparison between the numerical and experimental head of pressure. It can be observed a good agreement for both strategies. As is expected, the values are lower than the experimental one. Figure 4.33 show the advance of failure in meters.

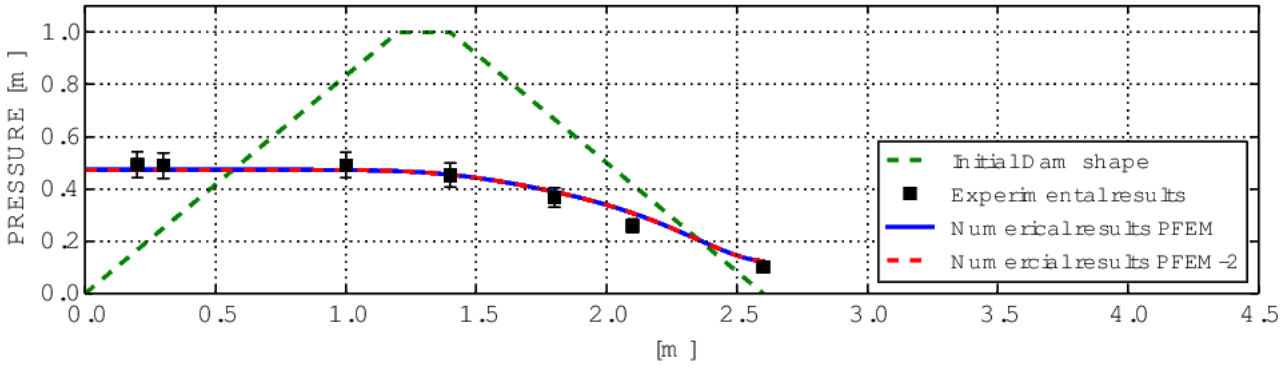


Figure 4.32: Comparison between numerical and experimental bottom pressure in case B22

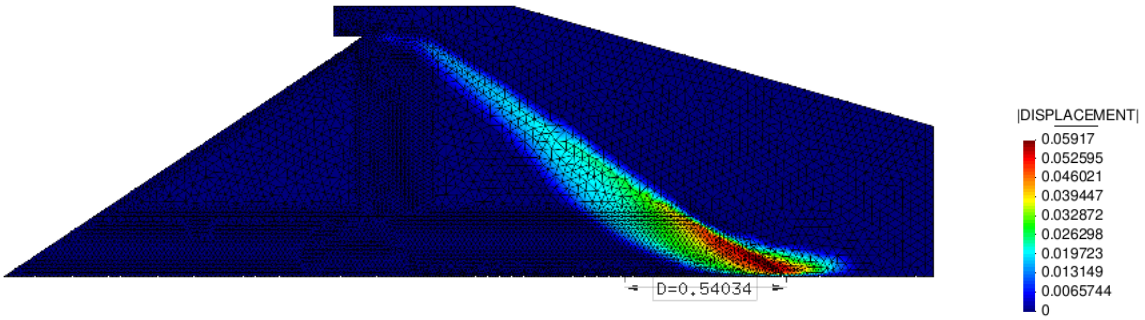


Figure 4.33: Numerical length of failure for Case B22 using PFEM-2

Table 4.11 presents the length of failure for both methods. It can be observed a good accordance between experimental and numerical results.

$B_{experimental}$ (m)	$B^{PFEM-2}_{numercial}$ (m)	$B^{PFEM}_{numercial}$
1.08	1.05	1.04

Table 4.11: Comparison between experimental and numerical length of failure in Case B22

- Case B23

  Figure 4.34 shows the comparison between the numerical and experimental head of pressure. It can be observed a good agreement for both strategies. As is expected, the

values are lower than the experimental one. Figure 4.35 show the advance of failure in meters. For this case the advance failure reached the crest of the dam.

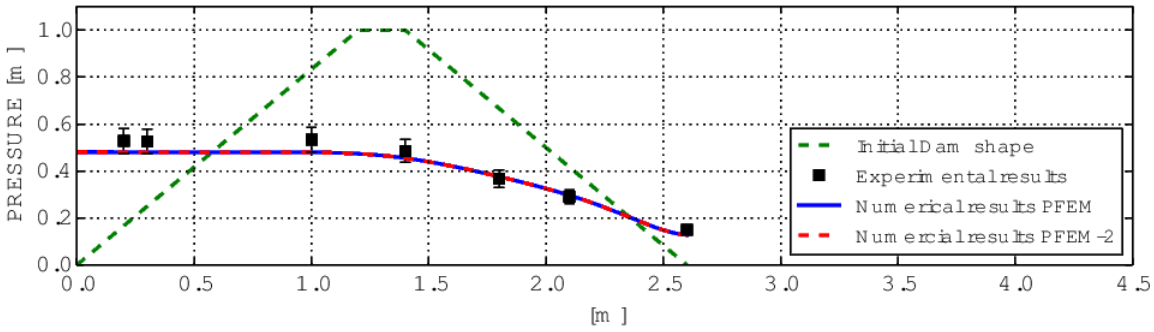


Figure 4.34: Comparison between numerical and experimental bottom pressure in case B23

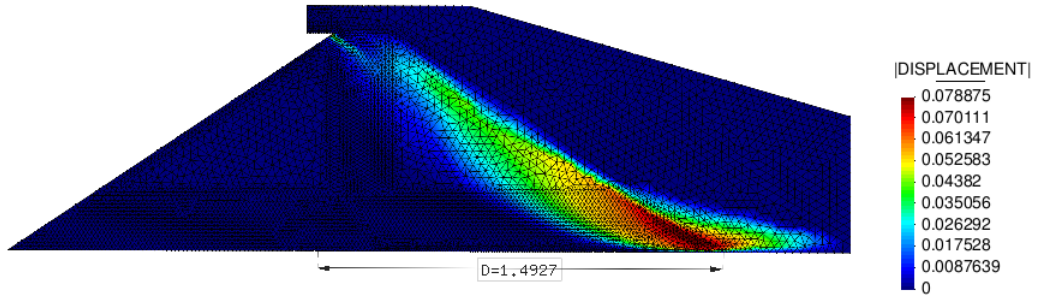


Figure 4.35: Numerical length of failure for Case B23 using PFEM-2

Table 4.12 presents the length of failure for both methods. It can be observed a good accordance between experimental and numerical results. The advance of failure with PFEM-2 is greater than the experimental. The cause can be the internal friction angle taken ($\phi = 40^o$).

$B_{experimental}$ (m)	$B^{PFEM-2}_{numercial}$ (m)	$B^{PFEM}_{numercial}$
1.44	1.49	1.40

Table 4.12: Comparison between experimental and numerical length of failure in Case B23

Additionally is important to note that in the experimental cases A23 and B23 the failure reached the crest of dam. The numerical B obtained with PFEM-2 did not reproduce the real value for case A23, however for case B23 the expected advance of failure is reproduced. This leads to the conclusion that some parameter for instance the body forces in case A23 need to be reviewed and modified.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

In this work a new coupled strategy (CFD-PFEM2) for the simulation of failure in rockfill dams during overtopping is presented. The dynamic evolution of seepage and the free surface is done using the edge-based code presented in section 2.1. The structural response is evaluated with a visco-rigid constitutive model. The rockfill is treated as a highly viscous non-Newtonian fluid and it is solved using the Particle Finite Element method second generation PFEM-2. In order to evaluate this new strategy, a comparison between the existing coupled strategy CFD-PFEM was performed using small scale rockfill dam models. Finally, the conclusions and future works can be summarize as follow:

1. As expected, the values are lower than the experimental due to the chosen resistance law. The choice of the suitable resistance law was performed in [1, 2] and it is beyond the scope of this work. There is a good agreement between experimental and numerical pressure heads in the studied models.

2. Unlike PFEM, PFEM-2 is able to reproduce internal friction angles greater than 45°. This behavior can be observed in section 4.2.4. This is not so relevant for rockfill dam simulation because 45° is a upper limit of the possible internal friction angle in a rockfill slope.

3. The CFD-PFEM-2 code reproduces correctly the advance of failure when the value of the incoming discharge is low and medium. Moreover, it is able to reproduce the advance of failure when reached the crest of the dam. However, a lower value of advance failure is presented in case A23. The cause could be the restriction imposed in the body forces. For that reason, a deep analysis on the development code should be performed as future work.

4. The computational time by the CFD-PFEM-2 strategy was higher than expected; since the PFEM-2 does not require neither remeshing nor creating a new system matrix, it was expected to outperform the PFEM strategy. However, the computational time of both methods was similar. A possible explanation for this is that the PFEM mesh is actually coarser than then fixed mesh, requiring less computational power. On the other hand, the current implementation of the PFEM-2 must compute the air phase when there is no granular material present (i.e. it is a two-phase solver). These two factors could easily diminish the advantages of the PFEM-2, finally resulting in a similar efficiency to the PFEM. However, we expect PFEM2 to be much more efficient

in 3D simulation were serious problems arises in PFEM remeshing. This should be analyzed in the future research on this topic.

5. In the future, the possibility of comparison between other type of rockfill dams as core dam should be performed with the purpose to evaluate all the scenarios about rockfill dam behavior. Additionally,the comparison between a 3D CFD-PFEM model and 3D CFD-PFEM-2 model should be investigated because a lack of sufficient accuracy had been observed in 3D case for the CFD-PFEM strategy [1].

Finally, it can be said that CFD-PFEM-2 is a promising strategy for the simulation of the analyzed problem due to times and results obtained are similar to the PFEM coupled strategy.

APPENDIX A

GID PRE AND POST PROCESS

The steps performed for the simulation of the examples in GID are:

1 step. The geometry is drawn (see Figure 3.1), boundary conditions and parameters are assigned, which depend of the analyzed problem.

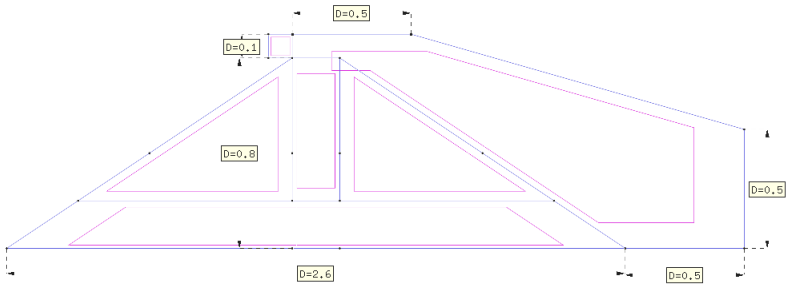


Figure A.1: An example of geometry in GID

2 setp. The mesh is generated . The elements used in this work are linear triangular (3 nodes) for 2D models while for the 3D models are linear tetrahedral (4 nodes).

GID allows to define fine meshes by zones, therefore the size of the mesh is changed depending on the problem (see Figure 3.2). It should be noted that a refined mesh produces high computational cost and therefore it is advisable to optimize the mesh depending on the needs of the calculation. For this work the mesh is refined in areas where the flow of water pass (see Figure 3.3).

3 step. The problem type is executed. GID generates a series of files. The most relevant files used for the calculation are:

<name of the file>.mdpa : presents the information about the nodes, elements and boundary conditions generated in the model.
problem_settings.py : presents the principal parameters to be use.

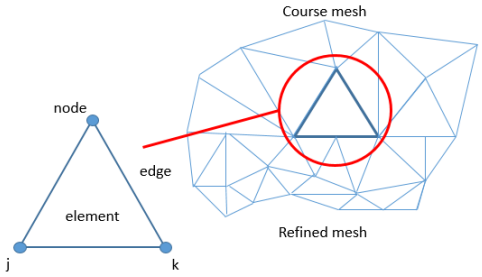


Figure A.2: Discretization of the geometry

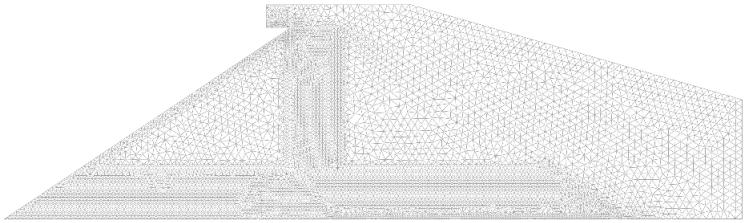


Figure A.3: An example of mesh generated in GID

script.py : defines the calculation to be performed in Python language.

In the calculation stage the Python file is run, which calls Kratos, passing the necessary information from the files generated in the pre-process by GID.

4 step. Results are displayed in GID in the phase of post-process (see Figure 3.4).

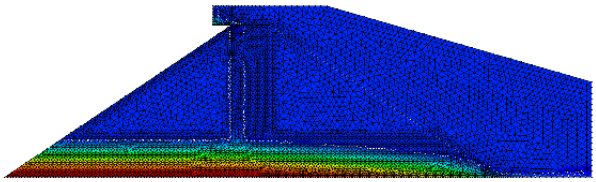


Figure A.4: Visualization of results in GID

Graphically this scheme can be summarized as:

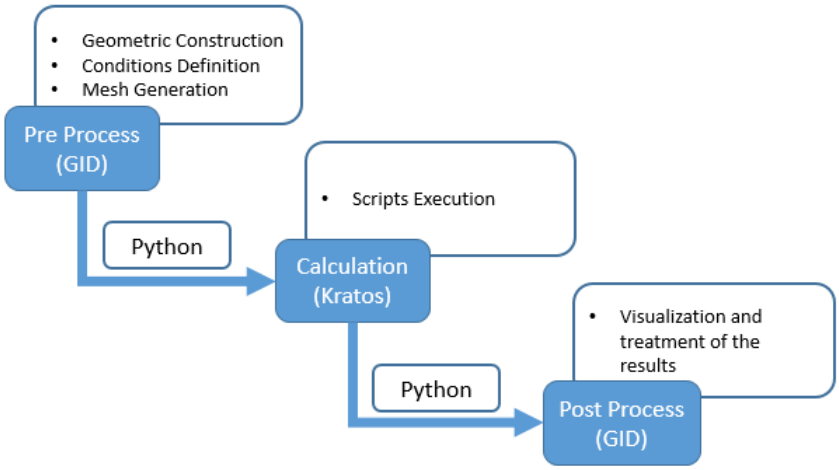


Figure A.5: Visualization of results in GID

APPENDIX B

THE COUPLED CFD-PFEM2 STRATEGY

```
from __future__ import print_function, absolute_import, division
# import the configuration data as read from the GiD
import problem_settings
import time as timer
##################################################################
##################################################################
# setting the domain size for the problem to be solved
domain_size = problem_settings.domain_size

##################################################################
##################################################################
# ATTENTION: here the order is important

# including kratos path
import sys
sys.path.append(problem_settings.kratos_path)

# importing Kratos main library
from KratosMultiphysics import *
from KratosMultiphysics.IncompressibleFluidApplication import *
from KratosMultiphysics.PFEM2Application import *
from KratosMultiphysics.PFEMApplication import *
from KratosMultiphysics.ExternalSolversApplication import*
from KratosMultiphysics.FSIApplication import*


# from now on the order is not anymore crucial
import math

# IMPORT BOTH FLUID AND STRUCTURAL SOLVERS
import edgebased_levelset_solver
import pfem_2_solver_monolithic_fluid_maria as pfem2_solver
```

```
# defining the TWO MODEL PARTS: pfem_model_part for the STRUCTURE
#and fixed_model_part for the FLUID
pfem_model_part = ModelPart("PfemFluidPart")
fixed_model_part = ModelPart("FixedFluidPart")

# importing the fluid_solver files and adding the variablesv
edgebased_levelset_solver.AddVariables(fixed_model_part)
fixed_model_part.AddNodalSolutionStepVariable(FLAG_VARIABLE)
fixed_model_part.AddNodalSolutionStepVariable(POROSITY)
fixed_model_part.AddNodalSolutionStepVariable(DIAMETER)
fixed_model_part.AddNodalSolutionStepVariable(SEEPAGE_DRAG)
fixed_model_part.AddNodalSolutionStepVariable(STRUCTURE_VELOCITY)
fixed_model_part.AddNodalSolutionStepVariable(MESH_VELOCITY)

# importing the structural_solver files and adding the variables
SolverType = "pfem2"
pfem2_solver.AddVariables(pfem_model_part)
pfem_model_part.AddNodalSolutionStepVariable(YIELD_STRESS)
pfem_model_part.AddNodalSolutionStepVariable(INTERNAL_FRICTION_ANGLE)
pfem_model_part.AddNodalSolutionStepVariable(PRESS_PROJ_NO_RO)
pfem_model_part.AddNodalSolutionStepVariable(SEEPAGE_DRAG)
pfem_model_part.AddNodalSolutionStepVariable(WATER_PRESSURE)
pfem_model_part.AddNodalSolutionStepVariable(DENSITY)
pfem_model_part.AddNodalSolutionStepVariable(POROSITY)
pfem_model_part.AddNodalSolutionStepVariable(MESH_VELOCITY)
pfem_model_part.AddNodalSolutionStepVariable(DISPLACEMENT)
# reading the models
name_pfem = problem_settings.problem_name + "_pfem2"
name_fixed = problem_settings.problem_name

gid_mode = GiDPostMode.GiD_PostBinary#Ascii
multifile = MultiFileFlag.SingleFile
deformed_mesh_flag = WriteDeformedMeshFlag.WriteUndeformed
write_conditions = WriteConditionsFlag.WriteElementsOnly

gid_io_fluid = GidIO(name_fixed, gid_mode, multifile,
deformed_mesh_flag, write_conditions)
gid_io_pfem = GidIO(name_pfem, gid_mode, multifile,
deformed_mesh_flag, write_conditions)

##### reading the pfem model part
model_part_io_structure = ModelPartIO(name_pfem)
model_part_io_structure.ReadModelPart(pfem_model_part)
print("pfem2 model read correctly")

##### reading the fixed model part with the new problem type format
model_part_io_fluid = ModelPartIO(name_fixed)
model_part_io_fluid.ReadModelPart(fixed_model_part)
```

```
print("fixed model read correctly")

##### sharing the parameters from the fluid model to the structural model
mapper = SharedPointsMapper(fixed_model_part.Nodes,
pfem_model_part.Nodes, 1.0/1000000.0)

##### Adding the degrees of freedom
edgebased_levelset_solver.AddDofs(fixed_model_part)
pfem2_solver.AddDofs(pfem_model_part)

# GET NODAL CONDITIONS FROM the PFEM model
body_force = Vector(3)
body_force[0] = problem_settings.body_force_x
body_force[1] = problem_settings.body_force_y
body_force[2] = problem_settings.body_force_z
pfem_model_part.ProcessInfo.SetValue(GRAVITY_X, 0.0 );
pfem_model_part.ProcessInfo.SetValue(GRAVITY_Y, -9.81)
pfem_model_part.ProcessInfo.SetValue(GRAVITY_Z, 0.0 );
pfem_model_part.ProcessInfo.SetValue(VISCOSITY_WATER, 0.01 );
#--> solid,  to change value by hand
pfem_model_part.ProcessInfo.SetValue(DENSITY_WATER, 1490.0);
#--> solid,  to change value by hand
pfem_model_part.ProcessInfo.SetValue(VISCOSITY_AIR, 0.0001 );
#--> air
pfem_model_part.ProcessInfo.SetValue(DENSITY_AIR, 1.0);
#--> air

# GET NODAL CONDITIONS FROM the FIXED model
# we assume here that all of the internal nodes are marked with a negative
# distance set the distance of all of the internal nodes to a small value
small_value = 0.0001
n_active = 0
for node in fixed_model_part.Nodes:
    dist = node.GetSolutionStepValue(DISTANCE)
    if(dist < 0.0):
        n_active = n_active + 1
        node.SetSolutionStepValue(DISTANCE, 0, -small_value)
    else:
        node.SetSolutionStepValue(DISTANCE, 0, small_value)
if(n_active == 0):
    raise "ERROR. At least one node has to be initialized
    with a distance lesser than 0"

# make sure that the porosity is not zero on any node
#(set by default to fluid only)
porosity_problem = 0.0
for node in fixed_model_part.Nodes:
    if(node.GetSolutionStepValue(POROSITY) == 0.0):
```

```
        node.SetSolutionStepValue(POROSITY, 0, 1.0)
    if(node.GetSolutionStepValue(DIAMETER) == 0.0):
        node.SetSolutionStepValue(DIAMETER, 0, 1.0)


###### takes the last value of the porosity
    if(node.GetSolutionStepValue(POROSITY) < 0.99999):
        porosity_problem = node.GetSolutionStepValue(POROSITY)


###### copying the porosity to the pfem_model_part
#VectorMap(rOriginVariable,rDestinationVariable)
mapper.ScalarMap(POROSITY,DISTANCE)

for node in pfem_model_part.Nodes:
        dist=node.GetSolutionStepValue(DISTANCE)
        if dist>0.99999:
            dist= 1
        else:
            dist= -1
        node.SetSolutionStepValue(DISTANCE,0,dist)

###### Pression need to be fixed in one point
        if node.Y>0.6999 and node.X<-0.04999:  ###--> to be change by hand
             node.Fix(PRESSURE)

        node.SetSolutionStepValue(BODY_FORCE_Y,0,-9.81)
        node.SetSolutionStepValue(DENSITY,0,
        pfem_model_part.ProcessInfo.GetValue(DENSITY_WATER))
        node.SetSolutionStepValue(INTERNAL_FRICTION_ANGLE,0,0.68)   ####--> by hand


for condition in pfem_model_part.Conditions:
      for node in condition.GetNodes():
           node.Fix(VELOCITY_X)
           node.Fix(VELOCITY_Y)
         #  if node.X<1.2 and node.Y<0.79 and node.Y>0.6:
         #     #node.Fix(PRESSURE)
         #     node.Free(VELOCITY_X)
         #     node.Free(VELOCITY_Y)



# the buffer size should be set up here after the mesh is read for the first time
pfem_model_part.SetBufferSize(2)
fixed_model_part.SetBufferSize(2)
```

```
# FLUID SOLVER CONSTRUCTION
fluid_viscosity = problem_settings.viscosity
fluid_density = problem_settings.density
solid_density = pfem_model_part.ProcessInfo.GetValue(DENSITY_WATER)
fluid_solver = edgebased_levelset_solver.EdgeBasedLevelSetSolver(fixed_model_part,
domain_size, body_force, fluid_viscosity, fluid_density)
fluid_solver.redistance_frequency = problem_settings.redistance_frequency
fluid_solver.extrapolation_layers = int(problem_settings.extrapolation_layers)
fluid_solver.stabdt_pressure_factor = problem_settings.stabdt_pressure_factor
fluid_solver.stabdt_convection_factor = problem_settings.stabdt_convection_factor
fluid_solver.tau2_factor = problem_settings.tau2_factor
fluid_solver.edge_detection_angle = problem_settings.edge_detection_angle
fluid_solver.assume_constant_pressure = problem_settings.assume_constant_pressure
fluid_solver.compute_porous_resistance_law =
int(problem_settings.compute_porous_resistance_law)
# 0 = None; 1 = Ergun; 2 = Custom;

# FLUID SOLVER INITIALIZATION
fluid_solver.Initialize()
if(problem_settings.wall_law_y > 1e-10):
    fluid_solver.fluid_solver.ActivateWallResistance(problem_settings.wall_law_y);


# STRUCTURAL SOLVER CONSTRUCTION
# adding dofs to the structural solver
# defining the structural solver
maximum_nonlin_iterations=10 #nonlinear problem
structural_solver = pfem2_solver.PFEM2Solver(pfem_model_part,domain_size,
maximum_nonlin_iterations)
# setting solver parameters
structural_solver.time_order = 1
structural_solver.echo_level = 1
# problem parameters to be changed
# time setting pfem
# pfem2 implicit
min_dt = 0.001
max_dt = 0.1

# edgebased explicit
max_Dt = problem_settings.max_time_step
initial_Dt_ls = 0.001 * max_Dt
final_time = problem_settings.max_time
output_dt = problem_settings.output_dt
safety_factor = problem_settings.safety_factor
number_of_inital_steps = problem_settings.number_of_inital_steps
initial_time_step = problem_settings.initial_time_step

# STRUCTURAL SOLVER INITIALIZATION
```

```
structural_solver.Initialize()

##PfemUtils = PfemUtils();
DragUtils = DragUtils();

#-----------------------------------------------------
gid_io_fluid.InitializeMesh(0.0);
gid_io_fluid.WriteMesh(fixed_model_part.GetMesh())
gid_io_fluid.FinalizeMesh();

gid_io_pfem.InitializeMesh(0.0);
gid_io_pfem.WriteMesh(pfem_model_part.GetMesh())
gid_io_pfem.FinalizeMesh();


gid_io_fluid.InitializeResults(0.0,(fixed_model_part).GetMesh())
gid_io_pfem.InitializeResults(0.0,(pfem_model_part).GetMesh())

out = 0
#out_step=1

original_max_dt = max_Dt
time = 0.0
step = 0
next_output_time = output_dt
max_safety_factor = safety_factor
check_structure_dt = True
print_structure = True
next_structure_step = time

gid_io_pfem.WriteNodalResults(VELOCITY,pfem_model_part.Nodes,0,0)
gid_io_pfem.WriteNodalResults(DISTANCE,pfem_model_part.Nodes,0,0)
gid_io_pfem.WriteNodalResults(PRESSURE,pfem_model_part.Nodes,0,0)


t0 = timer.time()
mapper.InverseScalarMap(POROSITY,DISTANCE)

detected_structure_velocity=False

while(time < final_time):
# print "line49"

    if(step < number_of_inital_steps):
        max_Dt = initial_time_step
    else:
        max_Dt = original_max_dt
        # progressively increment the safety factor
```

```
    # in the steps that follow a reduction of it
    safety_factor = safety_factor * 1.2
    if(safety_factor > max_safety_factor):
        safety_factor = max_safety_factor

# CALCULATE FLUID_DT AND STRUCTURE_DT
# check which of the time step is more restrictive
fluid_dt = fluid_solver.EstimateTimeStep(safety_factor, max_Dt)
struct_dt_factor=10.0
if time>0.05 and time<30.0 and detected_structure_velocity==False:
      struct_dt_factor=100.0
if(check_structure_dt == True):
    if(step < 2.0 * number_of_inital_steps):
      structure_dt = min_dt
    else:
      structure_dt = struct_dt_factor *  fluid_dt



    # check which of the time step is more restrictive
    if(structure_dt <= fluid_dt):
      Dt = structure_dt
      next_structure_step = time
    else:
      Dt = fluid_dt
      if(structure_dt * safety_factor > struct_dt_factor * fluid_dt):
         next_structure_step = time + struct_dt_factor * fluid_dt
      else:
         next_structure_step = time + safety_factor * structure_dt
         check_structure_dt = False
else:
    Dt = fluid_dt

print("fluid_dt = ", fluid_dt)
print("structure_dt = ", structure_dt)
print("Dt = ", Dt)

print("next_structure_step =", next_structure_step)

time = time + Dt
fixed_model_part.CloneTimeStep(time)

print("******** CURRENT TIME = ", time)

if(step > 3):
    #------------------------------------------------------------
    mapper.InverseVectorMap(STRUCTURE_VELOCITY,VELOCITY)
```

```
# solving the fluid problem ..........................
print("starting solving fluid edgebased~~~~~~~~~~~~~~~~~")
fluid_solver.Solve();
print("finished solving fluid edgebased")

print("start checking the time stepping")
check_dt = fluid_solver.EstimateTimeStep(0.95, max_Dt)

if(check_dt < Dt):
    print("************************************************************")
    print("************************************************************")
    print("************************************************************")
    print("            *** REDUCING THE TIME STEP ***")
    print("************************************************************")
    print("************************************************************")
    print("************************************************************")

    # we found a velocity too large! we need to reduce the time step
    fluid_solver.fluid_solver.ReduceTimeStep(fixed_model_part, time)
    # this is to set the database to the value at the beginning of the step

    safety_factor *= problem_settings.reduction_on_failure
    reduced_dt = fluid_solver.EstimateTimeStep(safety_factor, max_Dt)

    print("time before reduction= ", time)
    time = time - Dt + reduced_dt
    print("reduced time = ", time)
    print("Dt = ", Dt)
    print("reduced_dt =4 ", reduced_dt)

    fluid_solver.fluid_solver.ReduceTimeStep(fixed_model_part, time)
    # this is to set the database to the value at the beginning of the step
    fluid_solver.Solve()
print("finished checking the time stepping")
print(fixed_model_part)

if(time >= next_structure_step):

    pfem_model_part.CloneTimeStep(time)
    if(time > 0.0):
        print(pfem_model_part)
        # solving the structural problem
        print("starting solving structure~~~~~~~~~~~~~~~~~~~~~")

        #-------------------------------------------------------
        # Calculating Darcy non linear term
        DragUtils.CalculateFluidDrag(fixed_model_part,
        SEEPAGE_DRAG, fluid_viscosity, fluid_density, solid_density)
```

```
                mapper.ScalarMap(PRESSURE,WATER_PRESSURE)
                mapper.VectorMap(PRESS_PROJ,PRESS_PROJ_NO_RO)
                mapper.VectorMap(SEEPAGE_DRAG,SEEPAGE_DRAG)

                # Adding Darcy non linear term and Water weight to
                # the external force
##
                DragUtils.AddDrag(pfem_model_part, SEEPAGE_DRAG, BODY_FORCE,
                PRESS_PROJ_NO_RO, body_force)
                for node in pfem_model_part.Nodes:
                    if node.GetSolutionStepValue(DISTANCE)>0.0:
                        node.SetSolutionStepValue(BODY_FORCE_X,0,body_force[0])
                        node.SetSolutionStepValue(BODY_FORCE_Y,0,body_force[1])
                        node.SetSolutionStepValue(BODY_FORCE_Z,0,body_force[2])
##
                    nodal_body_force = node.GetSolutionStepValue(BODY_FORCE)
                    nodal_body_force_modulus = math.sqrt(nodal_body_force[0]
                    * nodal_body_force[0]
                    + nodal_body_force[1]*nodal_body_force[1]
                    + nodal_body_force[2]*nodal_body_force[2])
                    if nodal_body_force_modulus>20.0 :
                         if node.GetSolutionStepValue(DISTANCE)>0.0:
                            print("body_force=",body_force)
                         nodal_body_force = (20.0/nodal_body_force_modulus)
                         * nodal_body_force
                         node.SetSolutionStepValue(BODY_FORCE,nodal_body_force)


                structural_solver.Solve();


                check_structure_dt = True
                print_structure = True
                print("finished solving structure")

                print("PROJECT POROSITY")
                mapper.InverseScalarMap(POROSITY,DISTANCE)
                for node in fixed_model_part.Nodes:
                    porosity=node.GetSolutionStepValue(POROSITY)
                    if porosity>0.0:
                        porosity = 1
                    else:
                        porosity = porosity_problem
                    node.SetSolutionStepValue(POROSITY,0,porosity)

                nodes_with_velocity = 0
                for node in pfem_model_part.Nodes:
                    if (node.GetSolutionStepValue(DISTANCE)<0.0):
```

```
                        vel = node.GetSolutionStepValue(VELOCITY)
                        new_disp = node.GetSolutionStepValue(DISPLACEMENT)
                        + vel * Dt
                        node.SetSolutionStepValue(DISPLACEMENT,new_disp)
                        #new part, we check if there is movement
                        # to reduce the timestep
                        vel_modulus = math.sqrt(vel[0]*vel[0] + vel[1]*vel[1]
                         + vel[2]*vel[2])
                        if vel_modulus>0.001:
                             nodes_with_velocity+=1
                if nodes_with_velocity>2:
                      detected_structure_velocity=True
                else:
                      detected_structure_velocity=False



##################################
    if(time >= next_output_time):

        # a change in the output name is needed!!!!

# print "fluid output ****************************************************"
        gid_io_fluid.WriteNodalResults(VELOCITY, fixed_model_part.Nodes, time, 0)
      # gid_io.WriteNodalResults(CONV_VELOCITY,fixed_model_part.Nodes,time,0)
        gid_io_fluid.WriteNodalResults(PRESSURE, fixed_model_part.Nodes, time, 0)
        gid_io_fluid.WriteNodalResults(DISTANCE, fixed_model_part.Nodes, time, 0)
        gid_io_fluid.WriteNodalResults(PRESS_PROJ, fixed_model_part.Nodes, time, 0)
        gid_io_fluid.WriteNodalResults(POROSITY, fixed_model_part.Nodes, time, 0)
        gid_io_fluid.WriteNodalResults(DIAMETER, fixed_model_part.Nodes, time, 0)
        gid_io_fluid.WriteNodalResults(SEEPAGE_DRAG, fixed_model_part.Nodes, time, 0)
        gid_io_fluid.WriteNodalResults(STRUCTURE_VELOCITY,fixed_model_part.Nodes,time,0)

        gid_io_fluid.Flush()

        if(print_structure == True):

# print "structure output ************************************************"
            gid_io_pfem.WriteNodalResults(VELOCITY,pfem_model_part.Nodes,
            time,0)
            gid_io_pfem.WriteNodalResults(PRESSURE,pfem_model_part.Nodes,
            time,0)
            gid_io_pfem.WriteNodalResults(DISTANCE,pfem_model_part.Nodes,
            time,0)
            gid_io_pfem.WriteNodalResults(PRESS_PROJ,pfem_model_part.Nodes,
            time,0)
##          gid_io_pfem.WriteNodalResults(VISCOSITY, pfem_model_part.Nodes,
time,0)
```

```
##          gid_io_pfem.WriteNodalResults(DENSITY, pfem_model_part.Nodes,
time,0)
            gid_io_pfem.WriteNodalResults(BODY_FORCE, pfem_model_part.Nodes,
            time,0)
            gid_io_pfem.WriteNodalResults(DISPLACEMENT, pfem_model_part.Nodes,
            time,0)
            gid_io_pfem.WriteNodalResults(POROSITY, pfem_model_part.Nodes,
            time,0)
##          gid_io_pfem.WriteNodalResults(DIAMETER, pfem_model_part.Nodes,
time,0)
            gid_io_pfem.WriteNodalResults(SEEPAGE_DRAG, pfem_model_part.Nodes,
            time,0)
            gid_io_pfem.WriteNodalResults(WATER_PRESSURE, pfem_model_part.Nodes,
            time,0)
##          gid_io_pfem.PrintOnGaussPoints(EQ_STRAIN_RATE, pfem_model_part,
time)
##          gid_io_pfem.PrintOnGaussPoints(MU, pfem_model_part,
time)
##          gid_io_pfem.PrintOnGaussPoints(TAU, pfem_model_part,
time)


            gid_io_pfem.Flush()

            print_structure = False

        next_output_time = time + output_dt

        out = 0

    out = out + 1
    step = step + 1
    print("step finished",step)

gid_io_fluid.FinalizeResults()
gid_io_pfem.FinalizeResults()

print("solution finished")
t1 = timer.time()
print("Solution time with the new projection module   ", t1 - t0)
```

BIBLIOGRAPHY

[1] A.Larese, E. Oñate, and R. Rossi. *A coupled Eulerian- PFEM model for the simulation of overtopping in rockfill dams*. Monograph CIMNE. PhD thesis. Barcelona-Spain.

[2] A.Larese, R. Rossi, E. Oñate, and S.R Idelsohn. *A coupled PFEM-Eulerian approach for the simulation of porous FSI problems.*Comput Mech (2012) 50:805–819. DOI 10.1007/s00466-012-0768-9.

[3] A.Larese, R. Rossi , and E. Oñate. *Finite Element Modeling of Free Surface Flow in a variable porosity media*. Arch Computat Methods Eng. DOI 10.1007/s11831-014-9140-x.

[4] R. Rossi, A.Larese, P. Dadvand , and E. Oñate . *An efficient Edge-Based level set Finite Element Method for free surface flow problems*. International Journal for Numerical Methods in Fluids. Int. J. Numer. Meth. Fluids 2013; 71:687–716.

[5] A.Larese, R. Rossi, E. Oñate, M.A Toledo, R. Moran, and H. Campos . *Numerical and Experimental study of overtopping and failure of rockfill dams*. American Society of Civil Engineers.DOI: 10.1061/(ASCE) GM.1943-5622.0000345. 2014.

[6] A.Larese. *A Langragian PFEM approach for non-Newtonian viscoplastic materials*. CIMNE.

[7] P. Becker, S. Idelsohn and E. Oñate. *An enhanced Particle Finite Element Method with special emphasis on landslides and debris flows*. Monograph CIMNE. PhD thesis. Barcelona-Spain.

[8] P. Becker, S. Idelsohn and E. Oñate. *A unified monolithic approach for multi-fluid flows and fluid-structure interaction using the Particle Finite Element Method with fixed mesh.*. Computational Mechanics. 55:1091-1104. DOI 10.1007/s00466-014-1107-0. 2015.

[9] ICOLD. *Bulletin 99, Dam Failures Statistical Analysis*. ICOLD, 1995.

[10] M.A Toledo. *Presas de escollera sometidas a sobrevertido. Estudio del movimiento del agua a través de la escollera y de la estabilidad frente al deslizamiento en masa.*. Monograph UPM. PhD thesis. Madrid- Spain. 1997.

[11] M.A Toledo. *Safety of rockfill dams subject to overtopping.*Proceding of the International Symposium on new trend on guidelines on dam safety. 1998.

[12] R. Juan, A. Larese and E. Oñate. *Análisis numérico de los fénomenos hidrodinámicos en escolleras con aplicaciones a presas de materiales sueltos.* Final Project. UPC. 2010

[13] XPRES. *Desarrollo de un método para el estudio del proceso de rotura de presas de escollera por sobrevertido combinado con técnicas de elementos finitos y partículas.*Project of the national plan R+D of the Spanish Ministry of Science and Innovation I+D BIA 2007-68120-CO3-01, 2007-2010.

[14] E-DAMS. *Numerical and experimental techniques for safety assessment and protection of embankment dams in overtopping scenarios.*Project of the national plan R+D of the Spanish Ministry of Science and Innovation I+D BIA 2010-21350-CO3-00, 2010-2013.

[15] GID the personal pre ans post processor . *http://www.gidhome.com/.*

[16] Kratos, multiphysics open source fem code . *http://kratos-wiki.cimne.upc.edu.*

[17] Python programming language . *http://www.python.org/.*

[18] San Clemente Dam . *http://ccows.csumb.edu/wiki/index.php/San_Clemente_Dam.*

[19] Presas de Terraplén . *http://es.slideshare.net/sjnavarro/presas-de-terraplen-presentation.*

[20] Tipos de cortinas o presas hidráulicas . *http://ingenieriareal.com/tipos-de-cortinas-o-presas-hidraulicas/.*

[21] J.Bear. *Dynamic of Fluids in Porous Media.* Courier Dover Publications. 1988.

[22] D.A Nield and A Bejan. *Convection in porous media.* Springer. 1992.

[23] D.W Taylor . *Fundamentals of soil mechanics.* John Wiley and Sons, New York.1948.

[24] B. Li. *Flow through and overtopped rockfill dams.*PhD tehsis. University of Ottawa. 1995.

[25] J.C López. *Caracterización del proceso de saturación de una presa de escollera mediante métodos unidimensionales.* Diploma de Estudios Avanzados. Universidad Politécnica de Madrid. 2005.

[26] A Quality Tetrahedral Mesh Generator and a 3D Delaunay Triangulator . *http://wias-berlin.de/software/tetgen/.*

[27] R. Codina. *Stabilization of incompresibility and convection through orthogonal subscales in finite element methods.*Comput Method Appl Mech 190:1579-1599, 2000.

[28] P. Bussetam, R.Roman and J.P Ponthot *Comparison of data transfer methods between meshes in the frame of the arbitrary Lagrangian Eulerian formalism.* Proceeding of the V international conference on Advances Computational Methods in Engineering (ACOMEN 2011).

[29] A.J Chorin *A numerical method for solving incompressible viscous problems.* Journal of Computational physics. 2:12, 1967.

[30] R. Codina *A nodal-based implementation of a stabilized finite element method for incompressible flow problems.*International journal for numerical methods in fluids. 33:737-766, 2000.

[31] R. Codina *Stabilized finite element approximation of transient incompressible flows using orthogonal subscales*. Computer Methods in Applied Mechanics and Engineering. 191:4295-4321,2002.

[32] O. Soto, R. Lohner, J. Cebral and F. Camelli *A stabilized edge-based implicit incompressible flow formulation*. Computer Methods in Applied Mechanics and Engineering.

[33] R. Juon and W.H. Hager *Flip bucket with and without deflectors*.Journal of Hydraulic Engineering.

[34] R. Elias, M. Martins R. and A. Coutinho *Parallel edge-based solution of viscoplastic flows with the sup/pspg formulation*. Computational mechanics. 38:365-381, 2006. 10.1007/s00466-005-0012-y.

[35] R. Codina *Pressure stability in fractional step finite element methods for incompressible flows*. Journal of Computational Physics. 170:112-140, 2000.

[36] R. Codina. *A Stabilized finite element method for generalized stationary incompressible flows*.Comput Method Appl Mech submitted. Pages:1-36.