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

TREBALL FINAL DE GRAU

Reduced-Order Modelling of water flow in embankment dams. A parametric study.

Treball realitzat per:
Enric William Astals Buck

Dirigit per:
Alberto García González
Sergio Zlotnik Martinez

Grau en:
Enginyeria Geològica

Barcelona, 30 de juny de 2020

Departament d'Enginyeria Civil i Ambiental

Abstract

Tailings dams are enormous structures built to store the by-products of mining activities. Unlike water-retaining dams, this type of structure is raised gradually throughout its life span, in answer to the productivity of the mine. This can eventually lead to stability problems.

The high failure rate of tailings dams is a major problem across the globe, causing numerous casualties every year and devastating effects from toxic waste on the surrounding environment.

Sensor technology is currently used to monitor the pressure and stress state in their embankments, but as they transmit localised measurements, they are often difficult to interpret.

The work presented in this document serves as a contribution to a European project, which is designing a model that takes sensor data from a tailings dam, and creates a visual display of the pore water pressure and stress state across its entire body.

As hazardous conditions can develop within minutes, the model needs to operate in near real-time. To achieve this, it relies on Reduced-Order Modelling techniques, which are used to minimize its run time.

This document describes different parametric analyses carried out in an attempt to understand how the numerical techniques affect the model's speed and accuracy, and to establish how they can be applied to improve its performance, in an uncoupled approach.

The results of the investigations determine under which problem conditions the use of Reduced-Order Modelling techniques benefits the model, and show how its accuracy behaves in different circumstances.

Key Words: tailings dams, Reduced-Order Modelling, uncoupled approach, parametric analyses

Resumen

Las presas de relaves son enormes estructuras construidas para almacenar los desechos mineros. En contraposición a las presas de retención de agua, este tipo de estructura es elevada progresivamente a lo largo de su vida útil como respuesta a la productividad de la mina, pudiendo conducir a problemas de estabilidad a largo plazo.

La alta tasa de falla presentada por las presas de relaves es un serio problema en todo el mundo, ocasionando, anualmente, numerosos fallecimientos y la devastación del medio ambiente.

La tecnología de sensores se utiliza hoy en día para monitorizar el estado de presiones y esfuerzos en sus terraplenes, pero debido a que transmiten mediciones locales, tienden a ser difíciles de interpretar.

El trabajo presentado en este documento es una contribución a un proyecto europeo, que está llevando a cabo el diseño de un modelo que, de entrada, recoge datos de los sensores de una presa y, como salida, permite visualizar el estado de esfuerzos y presiones de la estructura entera.

Puesto que las condiciones de riesgo pueden desarrollarse en cuestión de minutos, es necesario que el modelo opere en tiempo real. Para lograrlo, emplea métodos de reducción de modelos, que son empleados para minimizar su tiempo de ejecución.

El documento describe diferentes análisis paramétricos llevados a cabo para entender cómo afectan los métodos numéricos a la precisión del modelo, y establecer cómo deben ser aplicados con tal de maximizar su rendimiento, en un enfoque desacoplado.

Los resultados de las investigaciones permiten conocer bajo qué condiciones la implementación de los métodos beneficia al modelo, y muestran cómo se comporta la exactitud de éste en diferentes circunstancias.

Palabras clave: presas de relaves, reducción de modelos, enfoque desacoplado, análisis paramétricos

Acknowledgements

To Christina, Berto, Sergio for offering me the possibility of working in this field, and for providing guidance in times of need.

To Peter, Linda and Lucy, for your unconditional help.

To Alba, for all the patience and support, and for encouraging me every step of the way.

To my father, Lluís Astals,
whom I love and dearly miss.
My aspirations amount to being
as kind a person as you were.

Table of Contents

List of Figures

List of Tables

Table of Abbreviations

Abbreviation	Description	Abbreviation	Description
EJD	European Joint Doctorate	**PDEs**	Partial Differential Equations
FEM	Finite Element Model	**POD**	Proper Orthogonal Decomposition
ITN	Innovative Training Network	**ROM**	Reduced-Order Modelling
MSCA	Marie Słodowska-Curie Actions	**SVD**	Singular Value Decomposition

Table of Parameters

Symbol	Description	Units
b_w	Vector of water body forces	Nm^{-2}
C	Specific moisture capacity	-
h	Pressure head	m
k	Hydraulic conductivity	ms^{-1}
k_{sat}	Saturated hydraulic conductivity	ms^{-1}
L	Length	M
m	Van Genuchten Model fitting parameter	-
n	Van Genuchten Model curve fitting parameter	-
p	Pore water pressure	Pa
Se	Effective saturation	-
T	Time dimension	S
t	Time	s
tol	Tolerance	%
α	Van Genuchten Model parameter	m^{-1}
γ_w	Specific weight of water	Nm^{-3}
θ	Angle of rotation	rad
θr	Residual water content	-
θ_s	Saturated water content	-

Table of Notations

Symbol	Description
A	Snapshot matrix
B	Reduced basis
B^T	Transposed of the reduced basis
Bmatrix	Snapshot matrix in the Python codes
E	Power of 10
$E_m{}^T$	Error produced by the composite trapezoidal rule
f	Load vector
f	Total number of Picard iterations
f	Function
h	Size of the subinterval
I	Composite Trapezoidal Rule
i	Spatial node
i	Subscript of a vector or matrix
j	Subscript of a vector or matrix
K	Stiffness matrix
k	Number of columns of the reduced basis / modes
m	Number of rows of the snapshot matrix and the singular values matrix
m	Total dimension to which an integral is applied
n	Temporal node
n	Size of the nonlinear system of algebraic equations
q	Number of columns of the snapshot matrix and the singular values matrix
RM	Reduced-order model
U	Left eigenvector matrix
u	Example vector space
ū	Solutions vector of the full-order model
V	Orthonormal matrix of vector directions
V^T	Right eigenvector matrix
v	Example vector space
x	Spatial node of the discretized dam
α	Solutions vector of the reduced-order model
Σ	Singular values matrix
σ	Stretch and compression factors / Singular values

1. Introduction

1.1. Framework

The following document collects and displays all the information relevant to an end-of-degree project, authored by Enric William Astals Buck, which consists of the study of the numerical techniques applied to a computational model built to improve the safety of tailings dams.

The paper intends to serve as a contribution to a larger European scientific project: *'Sensor data assimilation supporting decision making in the assessment of embankment dams'* conducted by Early Stage Researcher Christina Nasika [1]. This project belongs to a European Joint Doctorate programme (EJD) called ProTechTion, funded by the Marie Słodowska-Curie Actions (MSCA) and the Innovative Training Network (ITN), granted by the European Comission.

Figure 1-1: Emblem of ProTechTion and the Marie Słodowska-Curie Actions.

It is important to note that this project is in progress, meaning that there is very little information in the public domain.

The need to carry out the project arose from the high-rate of tailings dam failures across the globe, which cause numerous deaths every year, and release tremendous quantities of toxic waste.

Two main variables that affect the stability of a tailings dam are the distribution of the pore water pressure and the stress state of the structure [2][3][4].

The main objective of the project is to develop a numerical model that can be used to estimate the stress and pressure state of the entire structure of a tailings dam, and to ascertain whether or not its stability is at risk.

The following subsection will provide a broader introduction on the theoretical background of the European project and will introduce the need to carry out the study presented in this document.

1.2. Background

Tailings dams are huge structures used to store water and solid waste, also known as tailings, which are the by-products of mining activities, in their impoundment[1].

This type of embankment dams[2] are built in close proximity to their corresponding mining site, and usually account for the waste produced in the entire life of a mine. Unlike traditional water retention dams, tailings dams are not necessarily built across river valleys [5][6].

Rather than reinforced concrete, this type of dam uses rock fill and compacted soil to create the initial barrage. The embankment of the dam is usually partially constructed with tailings material, due to it being readily available and considerably cheaper than imported fill. This material is employed as a combination of waste particles and processed water, which form a semi-liquid mixture that has the appearance of mud. Many of the tailings deposited in the embankment and the impoundment of the dam are highly toxic [7].

As tailings dams are used as storing facilities for mining sites, their final size is not easily determined beforehand. Unlike most other types of dams, their structures are gradually raised, employing the use of tailings themselves to make the barrier higher, in order to accommodate more and more waste [5][8]. This overfill increases the chances of the dams developing hazardous conditions [4][5].

On average, between three and four of the world's total of 3500 tailings dams fail every year [3]. The probability of a tailings dam failure is several times higher than other conventional water-retaining dams [9], whose failure rate amounts roughly to only one in 10000 [3].

Due to the bulk and the toxicity of the material contained within their structure, the collapse of tailings dams produces catastrophic effects on the surrounding environment and human life. One of the most calamitous tailings dam failures happened only last year (2019) in Brazil, leaving more than 250 people dead and spreading approximately 12 million cubic metres of tailings across hundreds of square km of the regional river system and land [10].

[1] Impoundment: The area found behind the barrier of a dam, usually filled with water or, in the case of a tailings dam, mining waste.

[2] Embankment dams: A type of dam whose barrier is primarily constituted of earth and rock fragments.

Figure 1-2: Satellite images of the Brumadinho tailings dam in Brazil, before and after its failure [11].

In an attempt to respond to the growing necessity of finding a way to reduce the rate of failure of this type of dam, the European project was born. The project consists of the design of computational tools that, given observational data from a tailings dam as an input, develop a numerical model which permits the user to visualize, in real-time, a simulation of the stress and pressure state in its entire structure.

The importance of developing a model that works in real-time is because hazardous conditions can develop within hours or even minutes [12].

Partial Differential Equations (PDEs) constitute the foundation upon which the numerical model is built. They consist of mathematical expressions that attempt to describe the real phenomena that occur within a dam. These equations have a complex nature and require efficient approximation methods.

The project at hand uses the Finite Element Method (FEM) to estimate a solution for the system of PDEs through a series of loops[3].

Due to the complexity associated with solving the system of PDEs, there arose the need to utilize computational tools that help to minimize its run time, so the solutions are found at a much higher speed.

In order for the designed model to simulate the groundwater flow through the porous material of the dam in real-time, the project incorporates a Reduced-Order Modelling (ROM) technique, based on the so-called Proper Orthogonal Decomposition method (POD) [12]. This ROM method, which will be explained more thoroughly later on in the document, is implemented to increase the model's performance (its speed) by creating a so-called reduced basis of the model that simplifies the problem by decreasing the size (or order) of the system of PDEs that the model has to solve [13].

[3] Loop: Programming structure constituted by a sequence of instructions that are continually repeated until a certain condition is met.

It is important to note that a disadvantage that comes with reducing the order of the model is that a certain amount of information that describes the solutions of the model is lost in the process, meaning that the reduced basis cannot find the solutions of the system of PDEs as accurately as the original-sized model.

The purpose that the project intends to give to the constructed numerical model is for it to be used to analyse the state of real tailings dams, and raise an alarm if their integrity is at risk.

Sensor technology is a recently developed methodology for monitoring tailings dams, and has already been installed on various existing structures [14]. The problem of using the data obtained from these observational instruments without combining it with the proposed model is that it is very difficult to detect critical situations. Even though they are often programmed to transmit an alert when the value of a certain property is exceeded, they only hold localised measurements and, therefore, they lack information of the stress and pressure state of the entire body of the dam.

The idea is for the model to receive data in near real-time from the wireless monitoring sensor network installed on a dam, compute the value of the parameters that come into play across all of the structure and show how they evolve over time, and finally output a visual display of the whole dam's hydromechanical behaviour [1].

A current problem that the project is facing in relation to the employment of the ROM technique is that the performance of the model is only enhanced if the physical parameters entered are found within a certain value range (which is different for each parameter), meaning that it is highly dependent on the parametric values.

This constitutes a considerable drawback for the project, due to it becoming heavily limited by the characteristics of the dam, prohibiting its application to dams whose parametric values do not coincide with that particular value range.

Moreover, the larger project currently has a general lack of quantifiable knowledge in terms of how the model behaves in response to the creation of a reduced basis.

The work presented in this document intends to address the problems explained in the previous paragraphs, and to therefore assist the European project in achieving its goals.

The motivation to take part in this larger project is due to the alarming problem that tailings dams' high failure rate constitutes for human life and for the planet.

1.3. Aim and Objectives

The aim of this end-of-degree project is to contribute to the European project by studying the viability of applying numerical techniques and computational tools to increase the speed of the designed model, and investigate the parametric dependency that has been identified to occur under certain problem conditions.

The following objectives are set forth in order to fulfil this aim:

i. To propose methodologies that permit quantifying how the speed and the accuracy of the model behave in response to the creation of different reduced bases.
ii. To investigate the viability of reducing the order of the model to increase its speed.
iii. To delimit the conditions that have to be satisfied for the speed of the model to be enhanced without its accuracy being compromised.
iv. To analyse how the performance and the accuracy of the model behaves in response to different ranges of parametric inputs.
v. To gather information which enables the behaviour of the reduced bases of the model for different problem conditions to be understood.

Having defined the framework, the background and the objectives of the work, it is important to briefly describe the structure followed by the rest of the document.

The second chapter in the document (*Computational Model in Study*) will expand upon the description of the designed model and discuss how it works. It will also review the literature on the subjects that are relevant to the work.

After that, chapter 3 (*Computational Cost Analysis*) and chapter 4 (*Parametric Analysis*) will describe the analyses carried out towards fulfilling the objectives of the work, and discuss the obtained results.

Finally, the fifth chapter (*Project Conclusions*) will consist of a brief summary of the main accomplishments of the end-of-degree project.

The sources of information used to carry out the project can be found in the final chapter.

The appendices of the project can be found in a file named *Appendices_TFG_Enric_Astals.zip*. The file contains a PDF file with the lists of the simulations carried out and tables with the obtained results. Additionally, the last appendix comprises a programming code, which is found both at the end of the PDF file and in its original form, as a Python file (extension .py).

2. Computational Model in Study

Before delving into the practical side of the end-of-degree project, it is important to be aware of the programming tools that are used to build the model. Said model comprises a series of codes (written by the author of the European project), that when executed, work through the system of equations and iteration schemes, and outputs a set of solutions.

The computing platform used to build the model is FEniCS, which consists of a collection of programs which serve the purpose of constructing Finite Element models [15]. The programming language with which the codes are written is Python. The extension of said codes is '*.py*'.

Further description of the programming tools will be provided later in the chapter.

To understand the role that the piece of work presented in this document plays in the overall scheme of the European project, it is first necessary to briefly go through the hypotheses around which the designed numerical model is built and the most important aspects regarding its execution.

In order to construct a model that is sufficiently representative of the natural phenomena that describe groundwater flow through a tailings dam's structure, it is necessary to navigate through soil mechanics theorems, and consider the coupled relationship between soil and water.

The existence of a hydromechanical coupled effect means that the fluid (water) alters the properties of the soil it flows through, modifying its structure, and at the same time, the soil properties (such as the porosity) affect the behaviour of the fluid [16]. In short, it describes the interaction between soil and water.

Given that the introduction of a coupled effect to the studied problem adds a new layer of complexity, the present work will tackle the problem solely considering the water flow through the embankment of the dam. The uncoupled problem (or hydrological problem) does not, therefore, take into account the mechanical processes involved.

The subsequent subsection describes how the uncoupled problem is implemented in the model.

To understand how the model works, it is important to firstly come to grips with the equations that have to be solved in order for the model to output the sought results, and observe which parameters come into play.

2.1. Uncoupled problem

The goal of the model designed for the uncoupled (or hydrological) problem is to determine the pressure state in the entire structure of a tailings dam, given its geometrical properties and the value of the relevant physical parameters.

The equations displayed in the current subsection have been obtained, through personal communication, from the European project [17]. The information exposed on the functioning of the model is also obtained from the referenced source.

The governing equation for the hydrological problem is the water mass conservation, defined as follows:

$$\nabla^T \left[\frac{k(p)}{\gamma_w} (\nabla p + b_w) \right] + C(p) \frac{\partial p}{\partial t} = 0 \qquad (Equation\ 2-1)$$

Where,

p – Pore water pressure [Pa]

k – Hydraulic conductivity [ms^{-1}]

γ_w – Specific weight of water [Nm^{-3}]

b_w – Vector of water body forces [Nm^{-2}]

C – Specific moisture capacity [-]

t – Time [s]

The pore water pressure (*p*) is the most important variable of this first approach to the problem, as it defines the pressure state of the dam. In the above equation it is possible to identify that both the hydraulic conductivity (*k(p)*) and the specific moisture capacity (*C(p)*) are functions of *p*.

The expression that defines the hydraulic conductivity in these given conditions is the following:

$$k(p) = k_{sat} \sqrt{Se(p)} \left[1 - (1 - Se(p)^{\frac{n}{n-1}})^m \right]^2 \qquad (Equation\ 2-2)$$

Where,

k_{sat} – Saturated hydraulic conductivity [ms^{-1}]

n – Van Genuchten Model Fitting parameter [-]

m – Van Genuchten Model Curve fitting parameter [-]

Se – Effective saturation [-]

Se(p) is a variable that belongs to a hydraulic model, proposed by Van Genuchten, which characterizes the relationship between the water content and the pore water pressure in unsaturated soils [18]. It should be noted that the maximum value it can take is 1.

This model is called the water retention curve, in which the effective saturation is given by:

$$Se(p) = \begin{cases} \dfrac{1}{[1 + (\alpha \left| \dfrac{p}{\gamma_w} \right|^{\frac{m-1}{m}}]^m} & p < 0 \\ 1 & p \geq 0 \end{cases} \qquad (Equation\ 2-3)$$

Where,

α – Van Genuchten model parameter [m^{-1}]

The expression given to define the specific moisture capacity, C(p), is:

$$C(p) = \frac{-\alpha m(\theta_s - \theta_r)}{1 - m} Se(p)^{\frac{1}{m}}(1 - Se(p)^{\frac{1}{m}})^m \qquad (Equation\ 2-4)$$

Where,

θ_s – Saturated water content [-]

θ_r – Residual water content [-]

It is important to understand that the equations *2-2*, *2-3* and *2-4* form a system of partial differential equations (PDEs) that govern the uncoupled problem through *Equation 2-1*.

Having discussed the system of PDEs that defines the problem, the next important step is to explain how they are incorporated and resolved by the model. In the uncoupled problem, the code used to calculate the pore water pressure of the entire dam is *Hydro_Unsaturated.py*.

The code begins by inputting the geometrical properties belonging to an imaginary dam, and creating a discrete mesh[4], as seen in *Figure 2-1*.

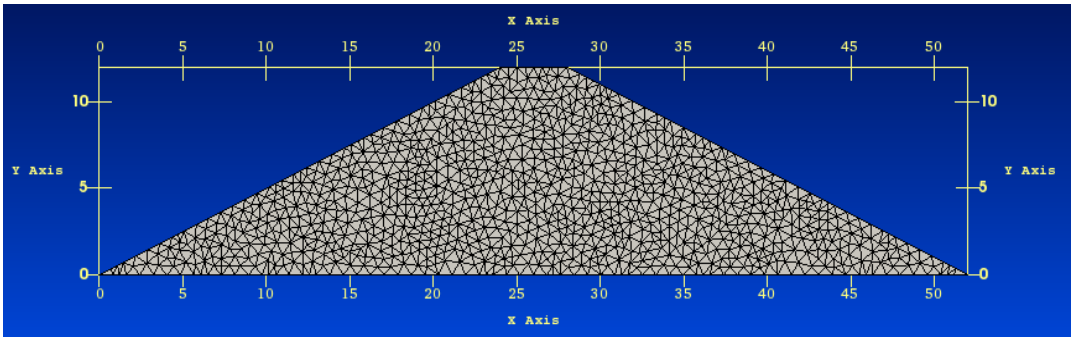

Figure 2-1: Discrete mesh of the dam used to construct the model of the European project. Paraview 5.2.0.

The currently studied mesh is formed by 1036 vertices (or nodes), which are the spatial points of the dam where the pore water pressure will be calculated [17]. It consists of an example which has been defined and constructed in order to enable the model to work. It is important to note that the number of spatial nodes depends on the geometry of the dam.

[4] Discrete mesh: A geometric objects formed by a finite number of subdivisions (or cells).

As the interest of the model is not only to determine the pore water pressure in the spatial dimension of the dam, but also obtain how it evolves over time, it is necessary to define the number of time steps and their magnitude.

This necessity to study the time-dimension of the dam originates from the fact that the water level on the upstream side of the dam fluctuates over time (it rises due to rainfall and the continual storage of tailings in the impoundment, and sinks due, mainly, to evaporation).

The time integration currently defined in the model, comprises 200 time steps (or temporal nodes), that each consists of 8640 seconds (2.4 hours). Therefore, the total measured time is 20 days (200·8640 seconds) [17].

Towards obtaining the pore water pressure of the entire body of the tailings dam in each of the defined time steps, the code first states that the pore water pressure is a known quantity for all of the spatial points of the dam at the initial moment. This initial condition is defined by the existence of an initial state of the water level, whose upstream fluctuation later causes the upstream and downstream boundary conditions to vary over time [17].

The study of how the system evolves is essential in order to evaluate hazardous pressure states.

At the beginning, the whole dam is unsaturated, being that the initial condition selected is the moment before the water starts flowing through the embankment.

At every time step, there is a loop (iteration sequence) encapsulated in another loop, the outer one being the time iteration process, and the inner being the solution convergence loop. Both of these will be explained more thoroughly in the following paragraphs.

From the starting point of having the initial values for the pore water pressure in every spatial point of the dam, the code calculates *Equation 2-2*, *Equation 2-3* and *Equation 2-4*, and then transfers the obtained values for k(p), Se(p) and C(p) into the governing equation, trying to drive it as close enough to zero in order for it to be coherent at both ends of the equal sign [17].

If the residuals of the equation are negligible (smaller than the error tolerance, defined as 10^{-4} in decimal expression [17]) for every point of the dam, convergence has occurred for that time step and the code jumps to the next time step, using a method called Crank-Nicolson to assign a new value of p to every point of the dam, in function of the state of the water-level curve at that moment.

On the other hand, if the governing equation doesn't approximate to zero for some (or all) of the spatial nodes, the FEM is applied to solve the system of PDEs, following the standard approach of formulating a weak integral form of the governing equation and discretizing it via the Galerkin Method [19].

This results in a nonlinear system of algebraic equations, which is then resolved using the Picard iteration method. The nonlinearity of the system is due to the presence of complicated boundary conditions. [17]

The Picard solver tackles the nonlinear system of equations by transforming it into a linear system that has to be evaluated a repeated number of times.

At each iteration of the Picard method, a new solution for the pore water pressure (p) is calculated, for every point where convergence wasn't achieved. Then, the loop starts again, by attacking the system of PDEs with the new pore water pressure values.

If after a certain number of iterations (specifically 25), the code doesn't converge on a solution for every point of the dam, or if the difference between the results obtained in consecutive iterations is lower than the error tolerance of 10^{-4}, the code warns the user that convergence was not successful in the current time step and the next time step is approached.

The process ends when the solution convergence loop finishes iterating for the last temporal node (n=200 in this case), at which point, the model outputs a matrix with the solutions (the values of pore water pressure in every node of the mesh in the complete time dimension) with 1036 rows and 200 columns.

In summary, the inner loop (or solution convergence loop) consists of an iterative process that solves an algebraic system of equations, derived from the system of PDEs by the FEM.

The outer loop (or time loop) describes the transitory character of the problem at hand. Due to the mentioned fluctuation of the water level upstream, the water table changes over time, causing variations in the pore water pressure in the dam.

To understand how the code works more fully, the following page presents a diagram which shows the dynamics of the iterative processes that take place in the *Hydro_Unsaturated.py* code.

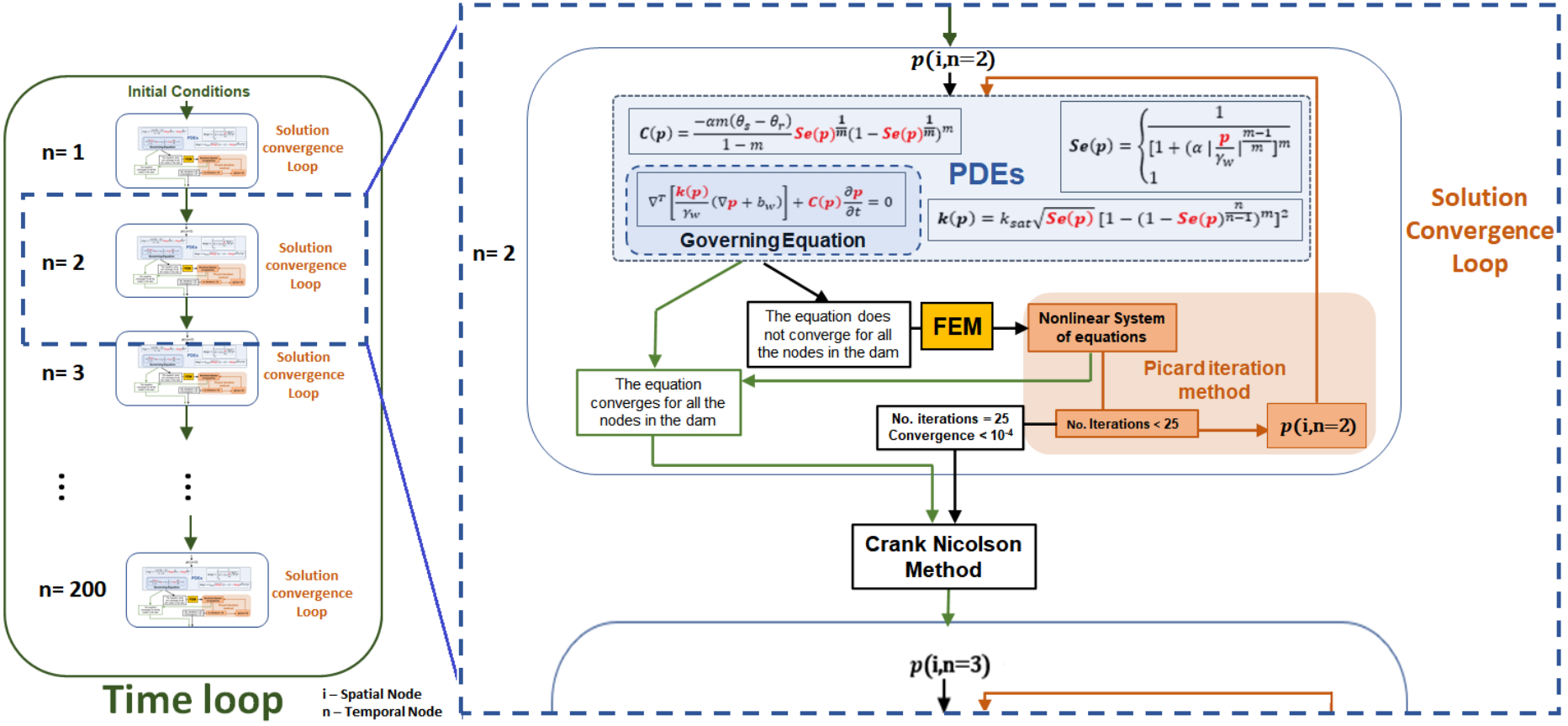

Figure 2-2: Diagram that displays how the model carries out the calculation of the evolution of the pore water pressure of the entire dam.

2.2. Reduced-Order Modelling

As can be observed in the content of the previous pages, the system of PDEs that has to be resolved via the FEM in order to obtain the desired output, is vastly complicated, due to the different iteration sequences and boundary conditions that come into play.

In a physical sense, the complexity associated with the model's functionality translates into it needing a high number of vectors to describe its solutions and therefore, taking a large amount of time to operate [12].

As one of the fundamental objectives of the European project is to achieve the construction of a model that processes monitoring data and outputs a visual schematic of the stress and pressure state of the dam in near real-time, its speed has to be maximized.

Hence, there is a strong need for implementing sophisticated analytical and numerical tools that yield low-dimensional approximations that are capable of reproducing the dynamics of the full-order system [20].

These small-dimensional simplifications are what is generally known as reduced-order models. Reduced-order models are low-dimensional constructions derived from the full set of nonlinear equations that the full-order model has to resolve [21][22].

Although the objective of using reduced order models is for the model to run in near real-time, their assembly can be computationally expensive. They require the full-order model[5] to be run a certain number of times in order to feed the model enough relevant information on the desired output, for it to then have the capacity of accurately predicting the solutions of the model, while including a much lower number of vectors.

This accumulation of solutions is achieved by the employment of the method of snapshots. To clarify, a snapshot is the state of a system for a particular parametric value [22].

The method of snapshots consists of solving the high-dimensional full-order model a fixed number of times, where the only difference between them is the value of a chosen parameter, and then storing the snapshots alongside one another in order to construct a matrix that effectively characterizes how the system responds [23].

Once the snapshot matrix is assembled, an algebraic factorization technique called the Singular Value Decomposition (SVD) is applied to it, which results in the defined matrix being broken up into the product of three matrices.

[5] Full-order model: Version of the designed model which has not interacted with any order-reduction techniques.

The importance derived from obtaining these matrices lies with the fact that two of them only hold information related to the vector directions that describe the solutions of the model, while the third describes how much of the system's information is provided by each vector direction.

Therefore, by using the SVD method, it is possible to quantify the minimum number of vector directions (or elements) that the model has to include in order to solve the PDEs system with an adequate precision.

After having decomposed the snapshot matrix into the product of three matrices, and identified which vector directions together integrate practically all of the information on the model's dynamics, the key is to use this knowledge to minimize the number of vector directions in accordance with their relative importance in the system.

The minimized matrix of vector directions is what is referred to as the reduced basis [13].

The concept of the reduced basis is distinctly important in this end-of-degree project, as its definition directly influences the creation of the reduced-order model.

It is important to note that the combined use of the SVD and the generation of a reduced basis are englobed in a POD-based ROM technique, as mentioned in *Background*. Nevertheless, the document will only use the terms SVD and reduced basis from now on, for the sake of clarity.

To further understand the ideas put forward in these paragraphs, it is necessary to gain a wider perspective on the workings of the SVD and the reduced basis.

2.2.1. Singular Value Decomposition

The current subsection is dedicated to offer a fuller explanation on the SVD method, used in order to obtain a version of the studied model that performs with greater speed.

SVD is a data analysis tool that is based on linear algebra [24][25]. This concept, at its core, originates from the effect a matrix has on a vector when they are multiplied [25][26].

To understand this idea, the following example vector and matrix are defined:

$$\boldsymbol{v} = \begin{pmatrix} 1 \\ 3 \end{pmatrix} \quad , \quad A = \begin{pmatrix} 1 & 1 \\ -1 & 2 \end{pmatrix}$$

Where,

v – One-dimensional array that has two rows and a single column (vector)
A – Two-dimensional array that has two rows and two columns (matrix)

The result of multiplying matrix *A* with vector ***v*** is a second vector, ***u***:

$$\boldsymbol{u} = \begin{pmatrix} 1 & 1 \\ -1 & 2 \end{pmatrix} \times \begin{pmatrix} 1 \\ 3 \end{pmatrix} = \begin{pmatrix} 4 \\ 5 \end{pmatrix}$$

By multiplication, matrix *A* transforms vector ***v*** into vector ***u***.

It is easier to perceive the physical meaning of this transformation if the example is treated graphically. The following graph (*Figure 2-3*) shows how vector ***v*** becomes vector ***u***.

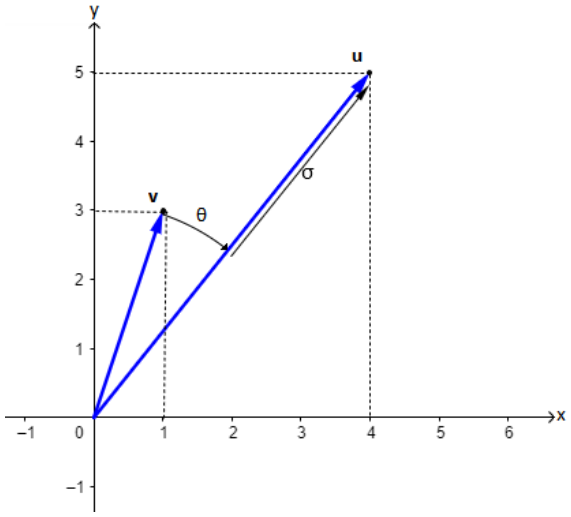

*Figure 2-3: Graph representing the transformation from a vector **v** into another vector **u**. GeoGebra.*

In essence, what occurs is that matrix *A* rotates and stretches vector ***v***, transforming it into vector ***u***. The rotation of vector ***v*** is defined by an angle of rotation θ, and the stretching depends on a stretching (or compressing) factor σ [27]. Both of these phenomena can be observed taking place in the above graph.

So, this is what a matrix does to an individual vector when multiplied.

Another useful example is to think about what happens to an n-dimensional sphere under a matrix multiplication. To begin with, it is practical to imagine a 2-dimensional unit circle which defines a vector space formed by orthonormal vectors[6] $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$ [25][28]. It is easy to see that $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$ are, therefore, both unit vectors.

Analogous to the example studied in the previous page, when the vectors are multiplied by a matrix, their magnitude and their direction change, forming new vectors $\boldsymbol{u_1}$ and $\boldsymbol{u_2}$. In other words, they stretch or compress and rotate.

Due to the stretching/compressing of the vectors, the new vectors are not unitary. However, they conserve the property of being orthogonal. They define a new vector space; whose shape is that of an ellipse. The vectors constitute the major and minor axes that define the ellipse

The following illustration shows the example that has been discussed in the previous paragraphs.

[6] Orthonormal vectors: A set of unit vectors that are orthogonal.

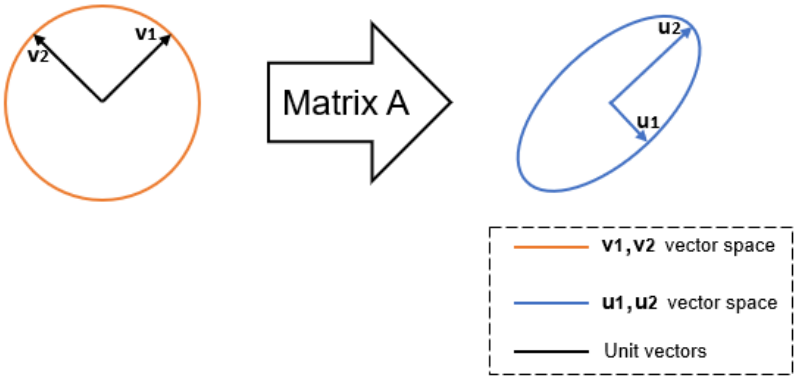

Figure 2-4: Matrix multiplication of a 2-dimensional unit vector space.

The new vectors $\boldsymbol{u_1}$ and $\boldsymbol{u_2}$ can be made unitary by redefining the major and minor axes of the ellipse as the product of a unit vector $\boldsymbol{u_i}$ and a stretching or compressing factor $\boldsymbol{\sigma_i}$.

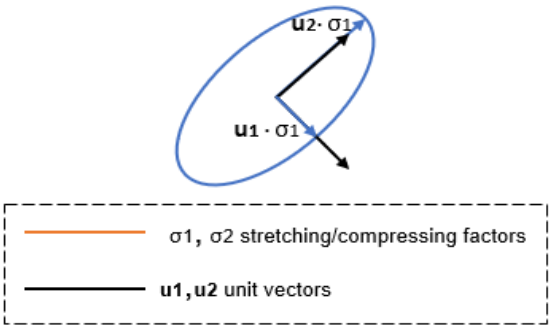

Figure 2-5: Redefinition of the major and minor axes of the ellipse.

In mathematical terms, the shown transformation from one orthonormal vector space to another, is described by the following expression:

$$A \cdot \boldsymbol{v_i} = \sigma_i \cdot \boldsymbol{u_i}$$

So, if matrix *A* acts on a single unit vector $\boldsymbol{v_i}$, it produces another unit vector $\boldsymbol{u_i}$, multiplied by a scalar σ_i which holds the magnitude the vector had, before being made unitary.

The two-dimensional example studied in the previous page sets the scene for interpreting the same phenomena at a much larger scale, as is the case of the problem studied in the project. The following step is to apply the same principle to an n-dimensional vector space, formed by *n* unit vectors $\boldsymbol{v_i}$.

By multiplying the vectors by a matrix with *m* rows and *n* columns, the expression becomes:

$$\begin{bmatrix} A_{1,1} & \cdots & A_{1,n} \\ \vdots & \ddots & \vdots \\ A_{m,1} & \cdots & A_{m,n} \end{bmatrix} \cdot \left[\begin{pmatrix} \vdots \\ \boldsymbol{v}_1 \\ \vdots \end{pmatrix} \begin{pmatrix} \vdots \\ \boldsymbol{v}_2 \\ \vdots \end{pmatrix} \dots \begin{pmatrix} \vdots \\ \boldsymbol{v}_n \\ \vdots \end{pmatrix} \right] = \left[\begin{pmatrix} \vdots \\ \boldsymbol{u}_1 \\ \vdots \end{pmatrix} \begin{pmatrix} \vdots \\ \boldsymbol{u}_2 \\ \vdots \end{pmatrix} \dots \begin{pmatrix} \vdots \\ \boldsymbol{u}_n \\ \vdots \end{pmatrix} \right] \cdot \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_n \end{bmatrix}$$

Which can also be expressed as:

$$A \cdot V = U \cdot \Sigma$$

Where,

A – Matrix with m rows and n columns

V – Orthonormal matrix of vector directions ($\boldsymbol{v_i}$)

U – Orthonormal matrix of vector directions ($\boldsymbol{u_i}$)

Σ – Diagonal matrix of vector magnitudes (σ_i)

An important property that needs to be taken into account is that the inverse of an orthonormal matrix is equal to its transposed self.

Using this information, the above expression can be rearranged to match the general formulation of the SVD method [28], as follows:

$$A = U \cdot \Sigma \cdot V^T \qquad (Equation\ 2-5)$$

Where,

V^T – Transposed Orthonormal matrix of vector directions (v_i)

Equation 2-5 shows that a matrix can be factorized into the product of three matrices mentioned in *Reduced-Order Modelling*. This decomposition can be carried out for any matrix [25].

In the project, the SVD method is applied to the matrix of snapshots, which comprises a set of solution matrices obtained from solving the full-order model a low number of times, for different values of a parameter.

The following illustration (*Figure 2-6*) shows a snapshot matrix formed by ten solutions of the full-order model (the intermediate of which are represented by a single vertical line for simplicity). The snapshots are taken for different logarithmically spaced values of saturated hydraulic conductivity (k_{sat}) measured in ms^{-1}, found in *Equation 2-2*.

The solutions that assemble the matrix are commonly referred to as the sampled solutions [13].

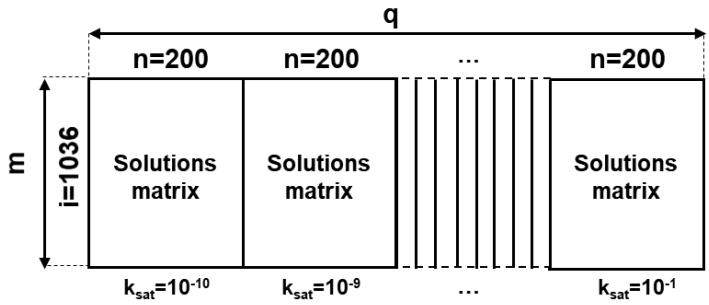

Figure 2-6: Illustration of a snapshot matrix comprising 10 solutions of the full-order model, solved for different values of saturated hydraulic conductivity between 10^{-10} and 10^{-1} ms^{-1}.

Having specified that the snapshots matrix in *Figure 2-6* contains 10 solutions of the full-order model, each constituted by 200 time steps, its total dimensions are 1036 rows (*m*) and 2000 columns (*q*).

By applying SVD to the defined snapshot matrix, the resulting dimensions of the product of matrices are as follows:

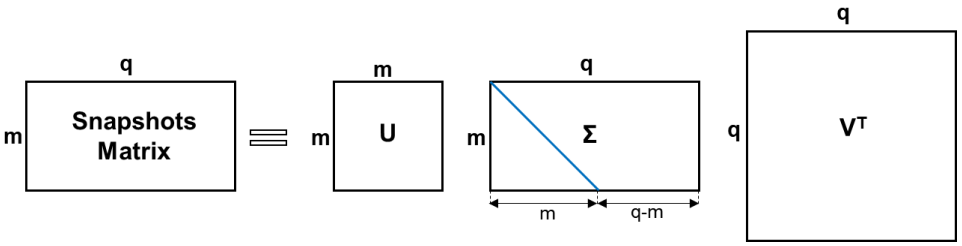

Figure 2-7: Illustration of the dimensions of the product of matrices U, Σ and V^T.

The diagonality of matrix ***Σ*** is represented by a blue line. As the matrix is not square, the last non-zero value of the matrix is found on column 1036. The remaining columns of the matrix (*q-m*) only contain zeros.

It is important to recall that matrix ***Σ*** contains the original magnitude of the orthonormal vectors found in matrix *U*.

Significant properties of matrix ***Σ*** are that the values found on the diagonal are all positive and are found in decreasing order [13]:

$$\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \ldots \geq \sigma_m > 0$$

Due to the properties stated above, Σ is commonly named the singular values matrix as its components (σ_i) are singular values[7] (or eigenvalues) [29].

Given the similarities between the SVD and another linear algebra theorem, the matrices U and V^T are commonly called the left and right eigenvector matrices, respectively [30].

2.2.2. Reduced Basis

Having talked about how the SVD works, it is now important to focus on the next step that has to be taken in order to reduce the order of the model.

It is important to remember that the singular values matrix ***Σ*** holds the magnitudes of the left and right eigenvector matrices *U* and V^T, in decreasing order.

For the purposes of this work, solely the ***Σ*** and the *U* matrices are of interest.

[7] Singular Values: Scalars that are used to stretch (or compress) an eigenvector.

As the vectors serve the purpose of describing the solutions assembled in the snapshot matrix, their magnitude can be translated into their relative importance. In other words, the vectors that have larger magnitudes describe more information of the model's solutions than those with smaller magnitudes.

Taking this fact into consideration, it is possible to simplify the system of algebraic equations that has to be solved by the model, by describing its solutions with fewer vectors.

If, for example, the first ten singular values of matrix Σ account for 99% of total sum of the magnitudes found in the diagonal matrix, the solutions of the model can be estimated to a precision of 1%, by solving a much simpler system of equations.

Therefore, the idea is to reduce the number of vectors used to describe the values of pore water pressure output by the model, by truncating the columns of both the singular values matrix Σ and the left eigenvector matrix U, as shown in the following illustration:

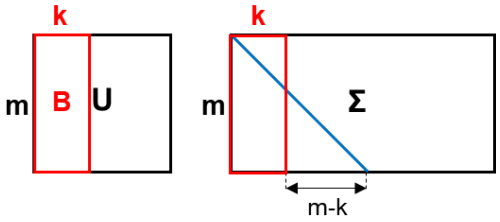

Figure 2-8: Truncation of the singular values matrix and the left eigenvector matrix.

The truncated left eigenvector matrix (*B*) is what is known as the reduced basis of the model. It has *m* rows and *k* columns. The columns of the reduced basis are often referenced as *modes*.

The importance of this finding is that the solutions of the model can be predicted by only taking into account the vector directions that form the reduced basis.

In order to understand this fact, it is important to briefly describe the nonlinear algebraic system of equations that has to be resolved using the Picard method (as shown in *Figure 2-2)*.

To simplify the explanation, the transient problem will not be taken into account. The simplified discretized nonlinear system of equations takes the following form [31]:

$$K \cdot \bar{\boldsymbol{u}} = \boldsymbol{f} \qquad (Equation\ 2-6)$$

Where,

K – Stifness matrix which holds all of the variables of the problem

ū– Solutions vector

f – Load vector which contains the initial and boundary conditions of the problem

The solutions vector ($\bar{u}$) has 1036 rows and 1 column. Towards applying the described truncation to the system of equations, the basis of $\bar{u}$ has to be changed.

To do so, the vector is redefined as the linear combination of a new vector (*α*) that has k rows and 1 column, and the reduced basis (*B*) [12].

$$\bar{\boldsymbol{u}} = B \cdot \boldsymbol{\alpha} \qquad (Equation\ 2-7)$$

Where,

B – Reduced basis of the model

α – Solutions vector of the reduced-order model

By changing the base of the solutions vector $\bar{u}$, the full-order model is transformed into a reduced-order model [12], which only takes into account the vector directions defined in the reduced basis.

The system of equations can therefore be rewritten as:

$$K \cdot B \cdot \boldsymbol{\alpha} = \boldsymbol{f} \qquad (Equation\ 2-8)$$

In order to obtain the desired dimensions, the left and right side of *Equation 2-8* have to be multiplied by B^T (the transposed reduced basis) [12].

The following illustration shows more clearly the transformations described in the previous paragraphs and the dimensions of the arrays.

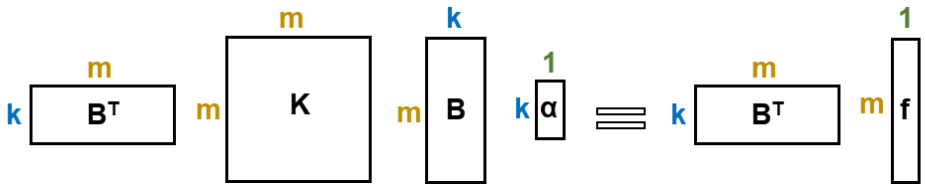

Figure 2-9: System of equations that has to be resolved after truncating the number of vector directions that describe the solutions of the model.

It can be seen that the resulting dimensions on both sides of the equation in *Figure 2-9* are those corresponding to vector *α*, meaning that the number of equations in the system is equivalent to the number of columns (*k*) that were preserved in the creation of the reduced basis.

These findings can be extended to the whole time dimension, which was previously omitted from the explanation.

The idea is for the Picard iterative method to be applied to this smaller system of equations in order to find the solutions of the reduced-order model (vector *α*). Having done that, the solutions vector of the full-order model for a certain parameter value can be calculated using *Equation 2-7*.

In summary, the creation of a reduced basis enables reducing the size of the nonlinear system of algebraic equations that has to be solved by the model. The information described by the vector directions that form the reduced basis can then be used towards predicting the solutions that would be obtained for the full system of equations, meaning that the pore water pressure in the entire space-time dimension can be approximated without having to solve such a large system of equations.

2.3. Uncoupled Codes

After discussing how the full-order model solves the uncoupled problem, and introducing the numerical techniques used towards increasing the speed with which the model operates, it is necessary to provide a brief description on the computational tools used by the European project to carry out these ideas.

As mentioned in the beginning of the chapter, the codes used have been written in Python language.

The codes belonging to the project for the uncoupled problem have the following file names [1]:

Table 2-1: Programming codes used by the project, divided into online and offline stage.

Uncoupled Codes	
Online Stage	**Offline Stage**
Hydro_Unsaturated.py	*SVD.py*
RB_Solver.py	*Snapshots.py*

The online stage codes are those which are directly used to solve the uncoupled problem and obtain the solutions of the model, both in the case of the full-order model (*Hydro_Unsaturated.py*) and the reduced-order model (*RB_Solver.py*). They have to be executed each time the problem has to be solved [32][33].

On the other hand, the offline stage codes are those that are necessary towards carrying out the reduced-order model [32][33], as they are responsible for collecting a set of solutions of the full-order model and generating a snapshots matrix (*Snapshots.py*), and later applying the Singular Value Decomposition towards creating a reduced basis (*SVD.py*). They can be computationally expensive [33], but do not have to be run every time the problem is solved.

The procedure followed by the *Hydro_Unsaturated.py* code has already been explained (see *Figure 2-2*). In the case of the remaining codes, a few remarks have to be made about how each one works.

Snapshots.py

The Snapshots.py code has a very similar content to the *Hydro_Unsaturated.py* code in the sense that it tackles the uncoupled problem by applying the FEM to the PDEs system and then solves the full-sized nonlinear algebraic system using the Picard method.

The difference between them is that the *Snapshots.py* code solves the problem multiple times, changing the value of a parameter each time it does. The code then assembles a snapshot matrix with the sampled solutions (see *Figure 2-6*).

The parameter that is currently used in the code to generate different snapshots is the saturated hydraulic conductivity (k_{sat}), as illustrated in *Figure 2-6*.

Figure 2-10 shows how the snapshot matrix is assembled for a range of k_{sat} values. In this case, the snapshot matrix consists of 10 solutions associated to logarithmically spaced k_{sat} values between 10^{-10} and 10^{-1} ms^{-1}.

The variable *Bmatrix* found on the third line of code (*Figure 2-10*) defines the matrix where the sampled solutions are eventually stored alongside one another.

```
param_domain = 9
snap_step = 1
Bmatrix =  np.zeros( (len(p_.vector()[:]),  math.ceil(steps*(((param_domain)/snap_step)+1))))

for u in range(0, param_domain+snap_step, snap_step):

    ksat = (10**u) * 1e-10

    print('Snapshot for parametric value:', ksat)
```

Figure 2-10: Part of the Snapshots.py code that is responsible of defining the snapshots. [1]

SVD.py

SVD.py is a short code which begins by inputting the snapshot matrix assembled by the *Snapshots.py* code (*Bmatrix*), and applying the SVD to it, via a simple inbuilt operation.

Then, the singular values matrix is exported to MATLAB, where a graph is plotted that permits visualizing the relationship between the relative importance of the singular values and the number of columns that form the matrix, in order to decide how many columns of the matrix should be discarded.

Having done this, the singular values and the left eigenvector matrices are truncated to the selected number of columns (as seen in *Figure 2-8*), thus creating the reduced basis of the model.

RB_Solver.py

Finally, the *RB_Solver.py* code tackles the PDEs system with the FEM, obtaining a nonlinear system of algebraic equations, as in the case of *Hydro_Unsaturated.py*.

It then inputs the reduced basis created by *SVD.py*, and changes the base of the solutions vector (see *Equation 2-7*), reducing the size of the nonlinear system of equations so it matches the number of columns conserved by the reduced basis.

After applying the Picard method to find the solutions for the reduced-order model, the solutions corresponding to the full-order model are calculated with *Equation 2-7*.

3. Computational Cost Analysis

The objective of this first chapter of the experimental part is to undergo a study of the computational cost associated with the model of the uncoupled problem, and analyse how it behaves in response to the reduction of the model's order.

As previously explained, the purpose of applying the SVD technique to the model is to reduce its dimension (i.e. the number of vector directions that describe the solutions of the original model), which in turn generates the so-called reduced basis, and produces a low-order model (or reduced-order model) that can effectively find the solutions to the nonlinear problem more quickly.

The computational cost is therefore related to the complexity of the system of equations that needs to be solved [34]. As the overall idea of the European project is for the model to input instrumental data from a real dam and permit the user to visualize the pressure and stress state of the entire structure in near real-time, finding a way to quantify the rapidity (or performance) of the reduced-order model is key.

A first obstacle encountered by the proposed analysis is that the computational cost cannot be directly translated into the amount of time it takes for the Python codes that form the model to reach their solutions. This notion is counter-intuitive, given that the objective of lowering the computational cost is for the model to minimize the time it takes to resolve the system.

The reason why the run time of the codes is not a good way to measure the computational cost is that it does not only depend on the complexity of the problem, but also on the level of code optimization[8].

So, in order to initiate a study of the computational cost, it is first necessary to define a method that only depends on the mentioned system complexity.

The standard of measurement proposed by this end-of-degree project is to quantify the number of operations undergone by the reduced-order model, in comparison to that of the full-order model.

To understand this idea, it is necessary to think back briefly on how the *Hydro_Unsaturated.py* code works (*Figure 2-2*).

At every temporal node, the model has to solve a system of PDEs, governed by *Equation 2-1*. Then, the FEM finds the weak form of the problem and discretizes it, obtaining a system of nonlinear algebraic equations, which are then resolved by the Picard iterative method.

[8] Code optimization: The process of modifying a code in order to enhance its efficiency.

As shown in *Reduced Basis*, the size of the system of equations that the Picard method has to resolve at each iteration, in the case of the full-order model, corresponds to the number of vector directions of the full-sized system of equations (1036 in the current discretization).

Meanwhile, in the case of a reduced-order model, the size (or number of equations) of the system is dictated by the number of columns conserved by the reduced basis.

Every time the Picard method resolves the nonlinear system of equations, the resulting solutions (pore water pressure) are then transferred into the governing equation.

When the Picard method manages to find the pore water pressure of all of the points of the dam, that validate the governing equation, convergence is reached and the code passes on to the next temporal node. If after 25 attempts, the Picard method does not reach convergence, or if the difference between the solutions it finds at each iteration is under a tolerance of 10^{-4}, the code passes on to the next temporal node [17].

To avoid confusion, the total number of systems of PDEs corresponds to the number of time steps. The total number of Picard iterations (or number of times a system is resolved) is found by multiplying the number of iterations the model needs to reach convergence, by the number of systems.

The term 'number of systems resolved' refers to the total number of iterations. It is important to clarify that solving a system does not imply that convergence is reached. It means that the model has undergone a Picard iteration.

Therefore, the idea in this chapter is to estimate the computational cost by evaluating the number of operations undergone by the model from start to finish.

This quantity is a result of three factors:

- The size of the system of equations that it has to solve with the Picard method at each iteration.
- The number of operations needed to solve a system.
- The total number of times a system is solved.

The following equation (*Equation 3-1*), adapted from the computational efficiency of the Gaussian Elimination method [35], links the mentioned factors in order to calculate the computational cost.

$$Computational\ Cost\ = f \times \frac{2}{3} n^3 \qquad (Equation\ 3-1)$$

Where,

f – Total number of Picard iterations (number of linear systems solved) [-]

n – Number of equations (or size) of the nonlinear system of equations [-]

The units given to the computational cost by the two variables do not have a physical sense, and will therefore not be included in the equation.

Equation 3-1 shows that the size of the system of equations that the model has to resolve carries more weight than the number of times it has to be resolved in order to reach convergence.

Taking into account the condition set upon the code which limits the total number of iterations per time step to 25, and having seen that, in its current design, the model runs through 200 time steps, it can be found that the maximum value that *f* can have in *Equation 3-1* is 5000:

$$f_{max} = 25\frac{iterations}{timestep} \times 200\ timesteps = 5000\ iterations$$

The calculated maximum is expressed in 'iterations' for the sake of clarity.

The maximum size the system can have is 1036, as corresponds to the full-order model.

$$n_{max} = 1036\ equations$$

So, the upper limit of the computational cost is the following:

$$Computational\ Cost\ = f_{max} \times \frac{2}{3}{n_{max}}^3 = 5000 \times \frac{2}{3}(1036)^3 = 3.71 \times 10^{12}$$

Although it is unclear how the obtained value relates to the actual speed with which the model functions, the proposed equation makes it possible to compare the computational costs of the full-order model with different sized models.

As for the lower limit, it can easily be found that the minimum number of iterations that can be made by the nonlinear solver in order for there to be convergence is 200 (one per time step), and that the minimum size that the system can have is 1, setting the limit at the following value:

$$Computational\ Cost\ = f_{min} \times \frac{2}{3}{n_{min}}^3 = 200 \times \frac{2}{3}(1)^3 = 133.\overline{3}$$

An important matter that will be analysed in the chapter is whether a lower computational cost is always beneficial.

The following subsections will approach the described analysis, explain how it has been put into practice, and discuss the obtained results.

3.1. Methodology

As explained in the introductory pages of the chapter, the method proposed for determining the computational cost of the full-order and reduced-order models is the application of *Equation 3-1*,

shaped by the number of Picard iterations that take place, from start to finish, and the size of the system of PDEs.

Being that the quantification of the number of iterations mainly consists of counting the number of linear systems that are solved in each model, it does not require further introduction.

In terms of putting this into practice, some minor modifications have to be made to the online stage codes (*Hydro_Unsaturated.py* and *RB_Solver.py*), in order for there to be additional outputs for each code, which calculate the number of Picard iterations undergone (the number of linear systems resolved) and calculate the computational cost of the model.

In relation to the size of the system of equations that needs to be resolved at each iteration, it is important to remember that, in the case of a reduced-order model, this quantity corresponds to the number of vector directions that form the reduced basis.

Another idea that needs to be recalled is that the size of the reduced basis is a direct consequence of the truncation of the singular values matrix, which indicates the magnitude (or importance), of the vectors that form the left eigenvector matrix. As the singular values matrix is ordered in decreasing importance, its truncation is indicative of how much of the original information is conserved by the reduced basis.

Taking these ideas into account, the idea is to significantly modify the offline stage code *SVD.py* so that it permits the user to define a tolerance that truncates the singular values matrix in terms of the importance of its components and then applies said truncation to the left eigenvector matrix, creating the reduced basis of the problem and indicating its size.

In order to transform the components of the singular values matrix into percentage values of importance, it is necessary to divide each one by the total sum of the singular values ($\Sigma(\sigma_i)$), and multiply them by 100.

$$\begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_q \end{bmatrix} \Rightarrow \begin{bmatrix} \frac{\sigma_1}{\Sigma(\sigma_i)} \times 100 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{\sigma_q}{\Sigma(\sigma_i)} \times 100 \end{bmatrix}$$

Figure 3-1: Transformation from a matrix holding the magnitudes of the vectors that form the decomposed snapshot matrix, to values that indicate the global importance of each corresponding vector direction.

The mentioned tolerance measures how much information of the full-order model's sampled solutions is lost in the process of generating a reduced basis by the truncation of its components. In other words, it is the difference between the information held by the full-order model (100%) and the information maintained by the reduced-order model.

The expression of the tolerance, in percentage form, is the following:

$$Tolerance\ [\%]\ =\ 100\ -\ \sum_{i=1}^{k}\left(\frac{\sigma_i}{\sum_{j=1}^{q}(\sigma_j)}\times 100\right) \qquad (Equation\ 3-2)$$

Where,

σ – Singular values

k – Number of columns of the truncated singular values matrix

q – Number of columns of the full singular values matrix

It is important to note that the subtrahend in *Equation 3-2* is referenced throughout the text as the level of information considered by the reduced-order model.

It is also significant to recall that the number of singular values conserved after carrying out the described truncation (σ_i) matches the number of vector directions that form the reduced basis. Therefore, the tolerance and the size of the reduced basis are inversely proportional.

Having described how the number of iterations and the size of the reduced basis affect the analysis, it is important to clarify that the two factors do not work independently. This notion refers to the fact that the number of linear systems that the model goes through is influenced by the level of information taken into account.

The expected behaviour of these two factors is for them to hold a proportional relationship to one another, meaning that, as the size of the reduced basis increases, so does the number of times that the model needs to iterate in order to reach convergence. Or, put differently, when the system of algebraic equations holds more information, and is therefore more complex, a larger number of iterations is needed for it to be resolved.

After modifying the codes appropriately, the idea is to run a series of simulations with the objective of analysing how the computational cost is affected by different tolerances (and therefore differently-sized reduced bases).

It is also important to assign part of the simulations to analysing the linked behaviour between the number of iterations (or behaviour of the nonlinear solver) and the size of the reduced basis.

Lastly, as the offline stage code *Snapshots.py* has been designed to take snapshots of the full-order model's solutions for different values of the saturated hydraulic conductivity (k_{sat}) parameter, the analysis will also take advantage of this fact and study the computational cost associated with different values of the parameter, in order to cover more ground and see how it behaves for different problem conditions.

As the purpose of the analysis is to study how the computational cost varies in different circumstances in order to improve the construction of the model, the physical representativeness

of the chosen saturated hydraulic conductivity values is not of immediate interest. In an attempt to include a large range of different possibilities, the selected values are the following:

Table 3-1: List of saturated hydraulic conductivity values that define the different problem.

Saturated hydraulic Conductivity (k_{sat}) [ms^{-1}]	
10^{-1}	10^{-2}
10^{-3}	10^{-4}
10^{-5}	10^{-6}
10^{-7}	10^{-8}
10^{-9}	10^{-10}

3.2. Study Delimitations

In order for this study to be conclusive, it is necessary to specify the conditions under which it is run. These delimitations describe the boundaries set up for the study.

In other words, they define the scope of the experiment. By defining fixed boundaries in the study, the number of variables is reduced, which permits focusing the experiments so they explore the specific interests put forward.

Clearly stating these boundaries in each analysis is a very important step, as it permits the reader to perceive the relationship between the simulations that take place and reality.

The delimitations chosen for the analysis of the computational cost of the uncoupled problem are as follows:

- The problem is studied under uncoupled conditions.
- The computational cost is measured as a function of the number of linear systems (or Picard iterations) the model resolves and the size of the system of algebraic equations.
- The size of the system that has to be solved by the full-order model is of 1036 equations.
- In the case of a reduced-order model, the size of the system of equations is determined by a chosen tolerance, which quantifies the level of information held by its corresponding reduced basis.
- The analysis is implemented for 10 different values of the k_{sat} parameter, logarithmically spaced in the range from 10^{-10} to 10^{-1} ms^{-1}.
- The snapshot matrix generated (in *Snapshots.py*) comprises the sampled solution matrices of the full-order model for the 10 chosen values of k_{sat}, stated in the point above.

3.3. Implementation

As briefly explained in the *Methodology* of the current chapter, in order for the analysis to take place, the first step that has to be taken is to modify the codes that form the uncoupled model.

The following table identifies the main points of the programming of the codes which has been carried out towards preparing the simulations.

The complete list of simulations carried out in the chapter can be found in *Appendix A.1.*

Table 3-2: Modifications made to the uncoupled codes.

Code	Modifications
Hydro_Unsaturated.py	- Creation of a variable that counts the number of Picard iterations the code runs through, every time it is executed. - Modification, every time the code is run, of the value of saturated hydraulic conductivity, in accordance with the listed values in *Table 3-1.* - Creation of a variable that calculates and outputs the computational cost of the full-order model, for the assigned value of saturated hydraulic conductivity.
Snapshots.py	- Modification of the range of values of saturated hydraulic conductivity to be considered for the generation of the matrix of snapshots.
SVD.py	- Modification of the functionality of the code, so that instead of the user deciding where to truncate the singular values matrix based on a graph exported to MATLAB, (as explained in *Uncoupled Codes*), a tolerance is input, which then truncates the matrices formed by the SVD and outputs the size of the resulting reduced basis.
RB_Solver.py	- Creation of a variable that counts the number of Picard iterations the code runs through, every time it is executed. - Modification, every time the code is run, of the value of saturated hydraulic conductivity, in accordance with the listed values in *Table 3-1*. - Creation of a variable that calculates and outputs the computational cost of the reduced-order model, for the defined tolerance and the assigned value of saturated hydraulic conductivity.

An important aspect of the practical work that the reader needs to be bear in mind is that, even though the computational cost is not quantifiable by the time the codes take to run, the run time does act as a limitation to the work carried out.

For the purpose of comprehending how the limitation affects the work, the following table (*Table 3-3*) gives an estimate of the approximate interval of run time (in minutes) associated with each code.

It is important to perceive that the run time of the code *RB_Solver.py* greatly depends on the size of the reduced basis (and therefore the size of the nonlinear system it has to resolve at each time step). The table includes some examples of the time it takes to run *RB_Solver.py*, for different values of tolerance defined in *SVD.py*.

Table 3-3: Run time intervals for the codes used to carry out the simulations.

Codes		**Run time [min]**
Hydro_Unsaturated.py		1.5 – 2
Snapshots_Liu.py		20 – 30
SVD.py		0.25 – 0.5
RB_Solver.py	**Tol: 1%**	1 – 4
	Tol: 10^{-3}%	5 – 15
	Tol: 10^{-5}%	32 – 50
	Tol: 10^{-10}%	110 – 135
	Tol: 10^{-15}%	195 – 215

The important message that the table tries to transmit is that the practical work carried out in the end-of-degree project is costly in terms of time. Therefore, the number of computational experiments (or simulations) that can be performed is influenced by this restriction.

After having made the specified changes to the codes and discussed the limitations of the practical work, it is possible to start the analysis.

The ensuing diagram presents the sequence of actions followed to run the simulations of the experimental work (*Figure 3-2*).

The full-order model's code *Hydro_Unsaturated.py* is run 10 times (as seen in the top right of *Figure 3-2*, modifying the value of the saturated hydraulic conductivity (k_{sat}) each time. As a result, the number of linear systems resolved and the resulting computational cost of the code are output.

In parallel, the *Snapshots.py* code is modified and executed.

As it can be seen in the diagram below, the first step in the process is to define the values of k_{sat} in *Snapshots.py*. When the code is run, it outputs a matrix of snapshots that comprises the sampled solutions of the full-order model, for each of the 10 logarithmically spaced values of k_{sat}.

After that, the chosen tolerance is written into the *SVD.py* code, and the code is executed. When run, *SVD.py* inputs the matrix of snapshots created by *Snapshots.py* and decomposes it into the product of three matrices and truncates the singular values matrix and the left eigenvector matrix, redefining the latter as the reduced basis.

The next step is to run the *RB_Solver.py* code 10 times, changing the value of the saturated hydraulic conductivity each time, in accordance with the values in *Table 3-1*.

Every time it is executed, the code inputs the reduced basis matrix created by the code *SVD.py*, and outputs the number of linear systems resolved from beginning to end, which is then used to calculate the computational cost, which is also output.

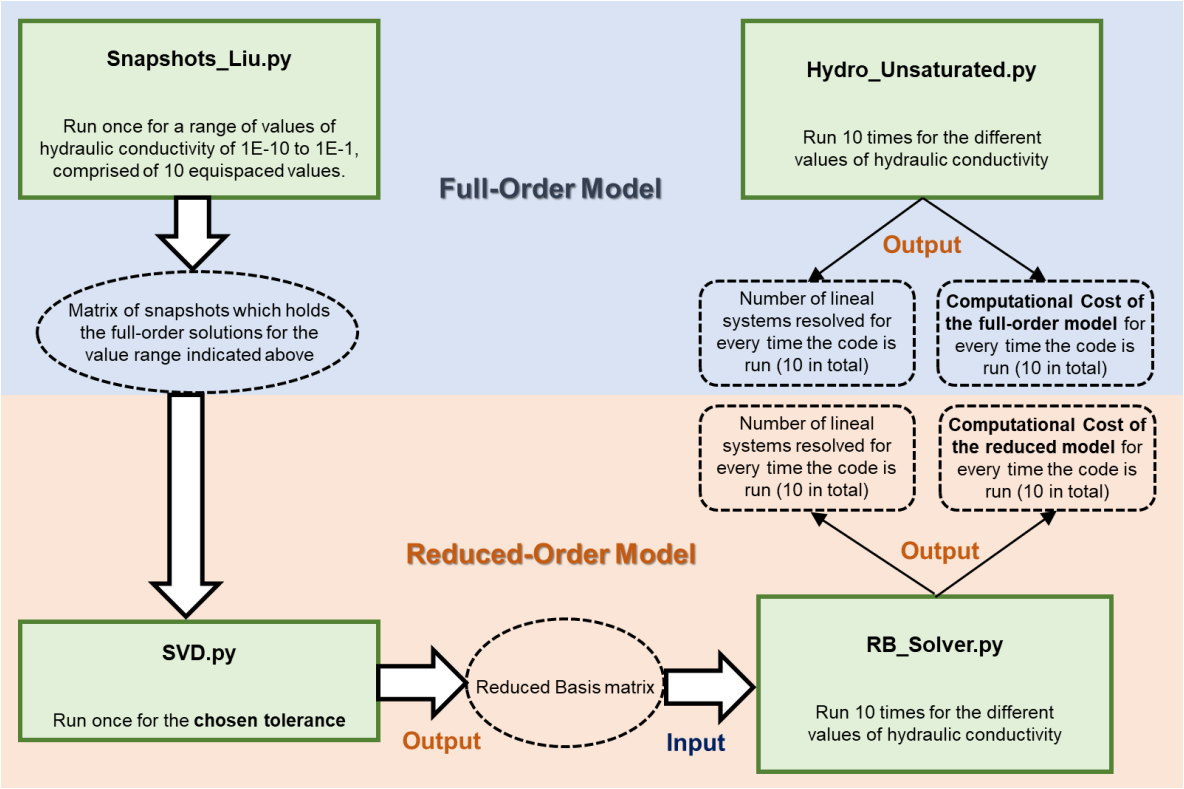

Figure 3-2: Diagram that illustrates the code execution sequence followed to build a set of simulations.

In order to repeat the same analysis for different tolerances, only part of the explained cycle has to be iterated for each value of the tolerance, as shown in the diagram below.

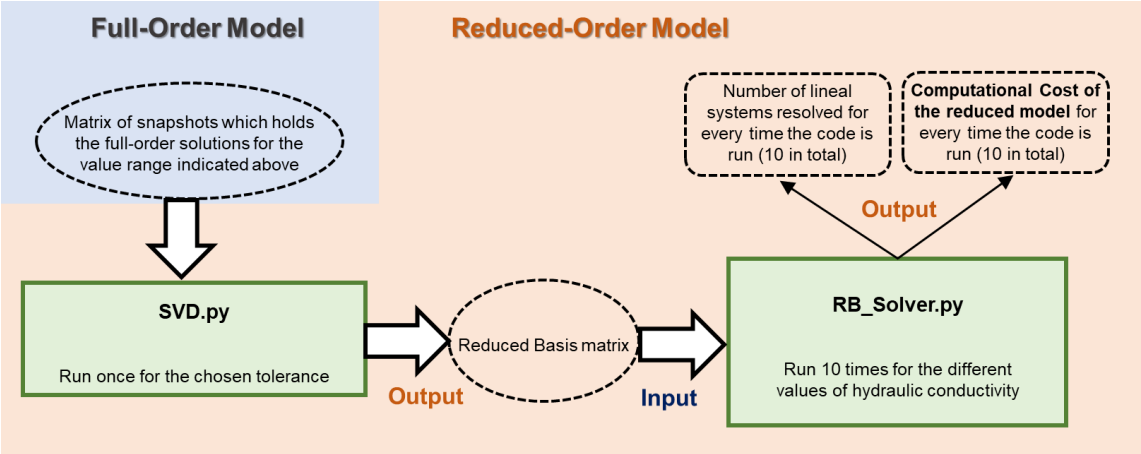

Figure 3-3: Cycle followed for every value of tolerance used to truncate the left eigenvector matrix.

3.3.1. Focus Questions

The current subsection intends to serve the function of formulating questions, to be answered in the remaining part of the chapter. The idea, as shown in the following diagram (*Figure 3-4)*, is to initiate a rigorous investigative process where the answer found for a specific question leads to asking and answering more questions, which in the long run, allows general conclusions to be drawn.

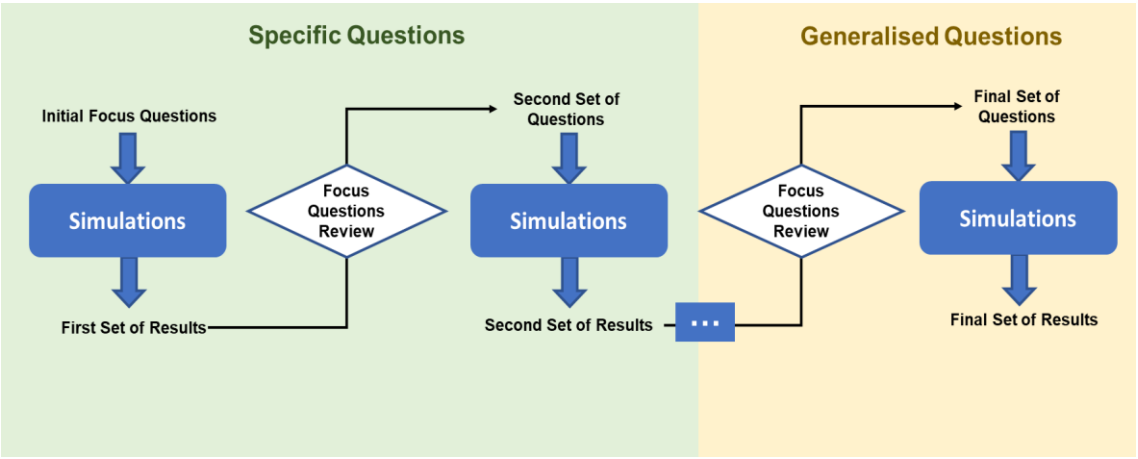

Figure 3-4: Diagram representing the investigative process followed in the analysis.

The diagram shows that the investigative process requires multiple iterations of the questions-answers cycle in order to obtain general conclusions. In the document, however, the entire sequence is simplified into two cycles (beginning and end).

The simplified version of the procedure, which also defines the structure followed in the rest of the chapter, incorporates the initial specific focus questions (defined in the current subsection), the corresponding first set of results (*First Set of Results)*, a review and reformulation of the focus

questions, which integrates all of the intermediate analysis carried out (*Focus Questions Review*) and the final set of general results (*Final Set of Results*).

It is clear that the criteria used to formulate the first set of questions are, to a certain extent, arbitrary. The importance of this exercise, however, lies with following up the initial questions, (after having found their answers), with other related questions. This enables a pattern to be established which, in turn, leads to being capable of analysing the behaviour of the computational cost associated with the reduced-order model in a more general sense.

The initial specific focus questions are the following:

- **Focus Question 1:** How many elements form the reduced basis when the tolerance is 1%?
- **Focus Question 2:** How many elements form the reduced basis when the tolerance is 0.1%?
- **Focus Question 3:** How many iterations of the Picard method are needed to solve the reduced-order model, for a tolerance of 1%?
- **Focus Question 4:** How does the value obtained as a result of the previous focus question compare to the number of linear systems solved by the full-order model, under the same problem conditions?
- **Focus Question 5:** For a tolerance of 1%, is the number of linear systems resolved by the reduced-order model the same for different values of saturated hydraulic conductivity?
- **Focus Question 6:** What is the computational cost of the reduced-order model, for a tolerance of 1%?
- **Focus Question 7:** What is the computational cost of the full-order model, under the same conditions as the previous question?

3.3.2. First Set of Results

After having run the first series of simulations, with the objective of finding the solutions to the initial focus questions, the current subsection summarizes the obtained results.

Apart from answering the specific questions presented in the previous subsection, the following pages also incorporate observations made during the simulations, which can potentially lead to the formulation of new focus questions, later in the document.

For clarity, the questions asked in *Focus Questions* are included again, followed by their answers.

Focus Question 1	How many elements form the reduced basis when the tolerance is 1%?

By running the modified *SVD.py* code, for a tolerance of 1%, it is possible to find that the reduced basis is made up of 4 elements (or vector directions).

In other words, only 4 vectors hold 99% of the information that describes the solutions of the full-order model.

Remembering that the full-order model is defined by a total of 1036 vector directions, the observation made above can be quantified in percentage form, such that:

$$\frac{4\ vector\ directions}{1036\ vector\ directions} \cdot 100 = 0.386\%$$

This means that 0.386% of the vectors can describe 99% of the information output by the model (the reduced-order model can predict the solutions of the full-sized model to a precision of 1%).

Focus Question 2	How many elements form the reduced basis when the tolerance is 0.1%?

For a tolerance of 0.1%, the reduced basis is formed by 14 vector directions. In other words, 14 elements hold 99.9% of the information.

$$\frac{14\ vector\ directions}{1036\ vector\ directions} \cdot 100 = 1.351\%$$

It is immediately noticeable that the number of elements that constitute the reduced basis is significantly bigger for 0.1% than for 1%, even though the quantity is still a very small portion of the total.

Focus Question 3	How many iterations of the Picard method are needed to solve the reduced-order model, for a tolerance of 1%?

As the question does not specify what value of saturated hydraulic conductivity needs to be considered, the following table (*Table 3-4)* presents the number of linear systems resolved, associated with a reduced basis defined by a tolerance of 1%, for the different values of k_{sat} used to generate the snapshot matrix (*Table 3-1).*

Table 3-4: Behaviour of the nonlinear solver of the Reduced-order model, for a tolerance of 1% and different values of saturated hydraulic conductivity.

Reduced-Order Model (99% of the information)										
Saturated Hydraulic Conductivity [ms^{-1}]	**10^{-10}**	**10^{-9}**	**10^{-8}**	**10^{-7}**	**10^{-6}**	**10^{-5}**	**10^{-4}**	**10^{-3}**	**10^{-2}**	**10^{-1}**
Nonlinear solver behaviour [No. of linear systems]	612	612	614	610	600	646	718	902	1043	1093

It is possible to perceive that for k_{sat} values between 10^{-10} and 10^{-6}, the nonlinear system behaves in a similar way, needing to resolve the same amount of linear systems for different values of k_{sat}.

Focus Question 4	**How does the value obtained as a result of the previous focus question compare to the number of linear systems solved by the full-order model, under the same problem conditions?**

Under the same conditions, the number of iterations undergone by the full-order model is as follows:

Table 3-5: Number of linear systems solved by the full-order model, for different values of saturated hydraulic conductivity.

Full-Order Model (100% of the information)										
Saturated Hydraulic Conductivity [ms^{-1}]	**10^{-10}**	**10^{-9}**	**10^{-8}**	**10^{-7}**	**10^{-6}**	**10^{-5}**	**10^{-4}**	**10^{-3}**	**10^{-2}**	**10^{-1}**
Nonlinear solver behaviour [No. of linear systems]	1166	1405	1759	2228	2022	2341	2617	2808	2901	3037

The similarity observed in the previous question between the number of linear systems solved by the reduced-order model for the lower limit of the range of saturated hydraulic conductivities, does not hold true with the full-order model. The number of iterations presents a high level of variation, and increases for higher values of k_{sat}.

By comparing the results obtained in the full-order model with the number of iterations solved by a reduced-order model that holds 99% of the original information, it is clear that the undergone reduction in the size of the model (which was originally characterised by 1036 vector directions, and is now only defined by 4), significantly lowers the number of iterations.

This comparison between the behaviour of the nonlinear solver in both cases can be seen more clearly in the following graph (*Figure 3-5*):

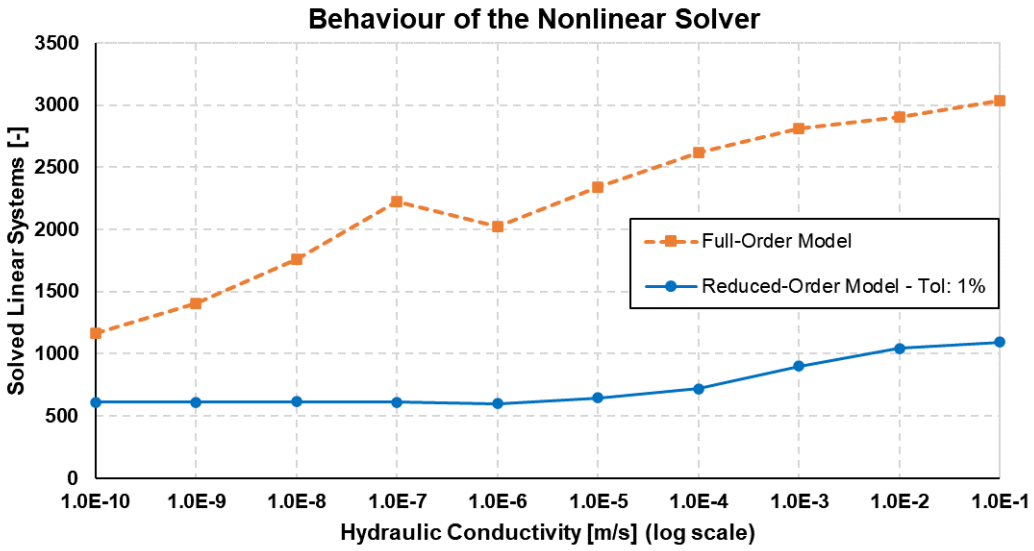

Figure 3-5: Comparison between the behaviour of the nonlinear solver in the full-order model and in the reduced-order model, generated by a tolerance of 1%.

It is easily noticeable that for both models, the computational cost has a tendency to grow for higher values of k_{sat}.

Focus Question 5	**For a tolerance of 1%, is the number of linear systems resolved by the reduced-order model the same for different values of saturated hydraulic conductivity?**

The answer to *Focus Question 3* has inadvertently found the response to this question.

So, as can be seen in *Table 3-4*, the number of Picard iterations run through by of the reduced-order model is generally not the same for different values of k_{sat}, although the lower values of k_{sat} present similarities. This can also be seen in *Figure 3-5.*

Focus Question 6	**What is the computational cost of the reduced-order model, for a tolerance of 1%?**

Thanks to having answered *Focus Question 1* and *Focus Question 3*, the computational cost for the reduced-order model generated by a tolerance defined at 1% can be calculated by means of *Equation 3-1*.

The results obtained are shown in *Table 3-6*:

Table 3-6: Computational cost of the reduced-order model, for a tolerance of 1%, for different values of saturated hydraulic conductivity.

Reduced-Order Model (99% of the information)										
Saturated Hydraulic Conductivity [ms^{-1}]	**10^{-10}**	**10^{-9}**	**10^{-8}**	**10^{-7}**	**10^{-6}**	**10^{-5}**	**10^{-4}**	**10^{-3}**	**10^{-2}**	**10^{-1}**
Computational Cost [-]	2448	2448	2456	2440	2400	2584	2872	3608	4172	4372

As explained in the introductory subsection of the chapter, the physical meaning of the obtained measurements of computational cost are somewhat vague. The importance of the chosen method, however, is that it enables comparisons to be made between the results obtained for one model and another.

Given that the computational cost is calculated by multiplying the number of Picard iterations by the number of operations needed in order to resolve the nonlinear system, whose size is determined by the applied tolerance, which remains constant for all of the different values of k_{sat}, the trends followed by the computational cost of the model will match those observed for the behaviour of the nonlinear solver.

Focus Question 7	**What is the computational cost of the full-order model, under the same conditions as the previous question?**

The computational cost of the full-order model is quantified in *Table 3-7*:

Table 3-7: Computational cost of the full-order model, for different values of saturated hydraulic conductivity.

Full-Order Model (100% of the information)										
Saturated Hydraulic Conductivity [ms^{-1}]	**10^{-10}**	**10^{-9}**	**10^{-8}**	**10^{-7}**	**10^{-6}**	**10^{-5}**	**10^{-4}**	**10^{-3}**	**10^{-2}**	**10^{-1}**
Computational Cost [-]	$1.2 \cdot 10^6$	$1.5 \cdot 10^6$	$1.8 \cdot 10^6$	$2.3 \cdot 10^6$	$2.1 \cdot 10^6$	$2.4 \cdot 10^6$	$2.7 \cdot 10^6$	$2.9 \cdot 10^6$	$3.0 \cdot 10^6$	$3.1 \cdot 10^6$

In the case of the full-order model, as the size of the system resolved by the nonlinear solver at each iteration is made up of 1036 equations, the resulting computational cost values are much higher than in the case of the reduced-order model for a tolerance of 1%, whose size is made up of 4 equations.

3.3.3. Focus Questions Review

Having discussed the first questions-answers cycle in the previous subsections, the current subsection serves the function of summarizing all of the experimental work, by providing a general review that permits the reader to perceive the directions followed in the intermediate simulations carried out. It also includes less-general conclusions drawn along the way and the formulation of the final set of focus questions, whose aim is to obtain a set of general conclusions to the computational cost analysis.

Being that this subsection does not follow the immediate events of the previous subsections, it is important to start the review by providing a brief explanation that describes the path followed since finding the answers to the first set of focus questions.

From the starting point of having found that, in answer to *Focus Question 1* and *Focus Question 2*, the size of the reduced basis grows substantially between the tolerances of 1% and 0.1%, the interest of the subsequent simulations is that of finding the size of the reduced basis for other values of the tolerance, and plot a graph that permits to visualize this relationship (*Figure 3-6*).

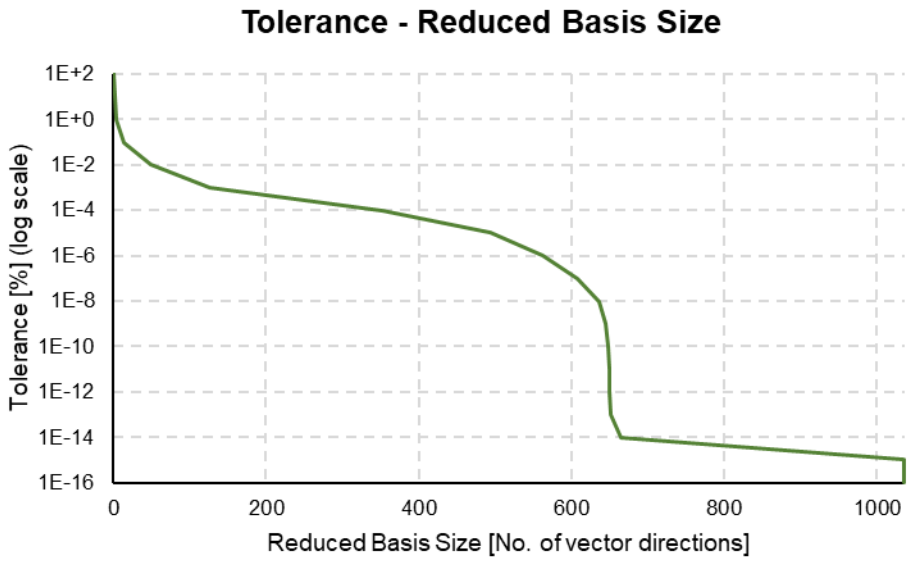

Figure 3-6: Relationship between the value of the tolerance and the size of the reduced basis.

It is important to notice that the vertical axis, which holds the values of the tolerances, is a base-10 logarithmic scale.

Table 3-8 quantifies the number of columns (or elements) that form the reduced basis for different values of tolerance.

Table 3-8: Relationship between the tolerance and the size of the reduced basis.

Tolerance [%]	100	10	1	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}
RB Size [No. elements]	0	2	4	14	50	126	352	494	563	607
Tolerance [%]	10^{-8}	10^{-9}	10^{-10}	10^{-11}	10^{-12}	10^{-13}	10^{-14}	10^{-15}	10^{-16}	10^{-100}
RB Size [No. elements]	637	644	648	649	650	651	664	1035	1036	1036

It is made clear that for a tolerance of 100%, which is obtained by considering 0% of the information described by the vector directions in the full-order model (see *Equation 3-2*), the reduced basis has zero elements.

Nevertheless, it is interesting to see the limits presented by the values of the tolerance, in order to see how the size of the reduced basis behaves between those boundaries.

At the other end of the spectrum, for a tolerance of 10^{-16}%, the size of the reduced basis is the same as the original model. In other words, it reaches its maximum size. This fact is corroborated by finding that the number of elements for an extremely low tolerance of 10^{-100}% is still 1036.

So, the size of the reduced basis varies in the range of tolerances from 100 to 10^{-16} %. It makes no sense, therefore, to analyse the computational cost outside this interval of tolerances.

In *Table 3-8* it can be observed that for tolerances between 100% and 1%, very few vector directions define the reduced basis.

Between percentage values of tolerance of 1 and 10^{-2}, the number of elements that form the reduced basis grow in significant proportions but continue being small percentages of the total. This can also be seen by taking a look at the gradient of the curve represented in *Figure 3-6* which is slightly higher between 1 and 10^{-1}. It is important to bear in mind that the tolerances are represented in a base-10 logarithmic scale.

Between the tolerances of 10^{-2} to 10^{-8} %, the number of vector directions increase more abruptly (with lower gradients in *Figure 3-6*).

Then, between 10^{-8} and 10^{-14}, the size of the reduced basis hardly changes, meaning that almost all of the vector directions that are left to account for constitute less than 10^{-14}% of the total information.

Finally, between the percentage tolerance values of 10^{-14} and 10^{-15}, there are a large number of vector directions (a total of 371), leaving the reduced basis with 1035 elements for a tolerance of 10^{-15}%, indicating that a single vector direction is found between the tolerances 10^{-15} and 10^{-16}%.

It is unclear whether or not the size of the reduced basis is influenced by how the matrix of snapshots, created in *Snapshots.py*, is assembled. As it is necessary to gather as much information as possible in order for the analysis to be indicative, this possibility is worth investigating in further simulations (see *page 45*).

Another way of visualizing what has been discussed in the previous paragraphs is found by transposing the axis of the graph in *Figure 3-6* and transforming the axis that characterises the size of the reduced basis into percentages.

This alternative is represented in the following bar chart:

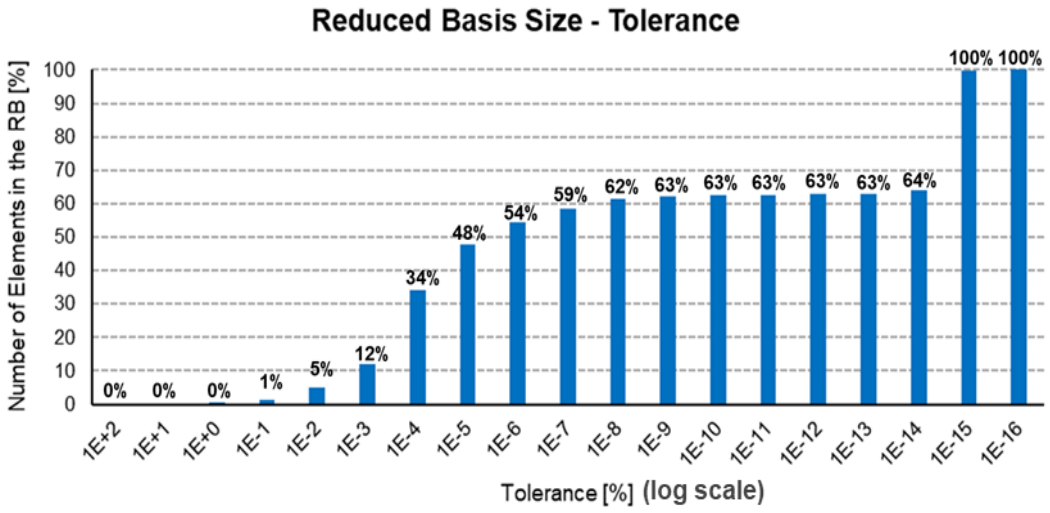

Figure 3-7: Relationship between the percentage size of the reduced basis (out of a total of 1036 vector directions) and the tolerance.

The previous representation permits stating that, under the conditions of the present study, in order to maintain, for example, 99.999% of the information held by the assembled matrix of full-order solutions (the matrix of snapshots), which corresponds to a tolerance of 10^{-3}%, only 12% of the elements that comprise the full-order model have to be used.

Having previously demonstrated that, alongside the number of elements employed to describe the solutions matrix of the model, another factor that directly influences the computational cost is the number of linear systems the model has to resolve, the next important matter that has to be approached is that of quantifying this number for different situations.

The answers to the initial *Focus Question 3* and *Focus Question 4* lead the investigation to draw the graph found in *Figure 3-5,* which shows that the number of linear systems solved by the reduced-order model, evaluated for ten values of k_{sat} and defined by a tolerance of 1%, is lower than that of the full-order model.

After that, further simulations were performed, in order to make analogous comparisons, for different tolerances. The full results of these simulations can be consulted in *Appendix A.2*.

There are some observations concerning these simulations that should be highlighted.

The first set of solutions for this comparison, represented in Figure 3-5, shows that the number of Picard iterations resolved by both models grows progressively for higher values of saturated hydraulic conductivity.

By looking at how the nonlinear solver behaves for different tolerances, as seen in the following table (*Table 3-9*), it can be noted that for the reduced-order models defined by tolerances in the range of 10^{-1} and 10%, the results seemingly obey this established trend, although there are minor local discrepancies.

In the case of the tolerances of 10^{-2} and 10^{-3}%, the mentioned tendency is not followed for the lower values of saturated hydraulic conductivity. Even so, in the case of both these tolerances, it does occur for higher values of saturated hydraulic conductivity.

In fact, all of the tolerances studied in *Table 3-9* corroborate the existence of this trend when only considering the interval of saturated hydraulic conductivity values comprised between 10^{-6} and 10^{-1}. This is made easier to identify in the table through the employment of green shades.

Table 3-9: Number of linear systems solved by the reduced-order model, for tolerances between 10^{-3} and 10%, and for different values of saturated hydraulic conductivity.

Reduced-Order Model											
Tolerance [%] \ **Saturated Hydraulic Conductivity [ms^{-1}]**		**10^{-10}**	**10^{-9}**	**10^{-8}**	**10^{-7}**	**10^{-6}**	**10^{-5}**	**10^{-4}**	**10^{-3}**	**10^{-2}**	**10^{-1}**
10	**Linear systems solved**	452	452	452	450	450	474	556	592	653	816
1	**Linear systems solved**	612	612	614	610	600	646	718	902	1043	1093
10^{-1}	**Linear systems solved**	990	1001	981	1054	1395	1478	1480	1819	1805	1907
10^{-2}	**Linear systems solved**	1689	2920	2142	1839	1603	1789	1892	2194	2107	2356
10^{-3}	**Linear systems solved**	4843	4773	2316	1822	1806	1806	2291	2682	2525	2723

For saturated hydraulic conductivity values equal or lower than 10^{-7}%, another observation that can be made based on the table is that, for the higher tolerance values of 10^{-1} and 10%, the number of systems resolved by the reduced-order model is approximately constant. In the table, blue shades are applied to these values, for clarity.

On the other hand, for lower tolerances, the number of solved systems becomes highly variable for different values in the range of saturated hydraulic conductivity. This phenomenon can be identified by the red shades in the table.

The following graph (*Figure 3-8*) presents the information found in *Table 3-9* in a more visual way, conserving its colour formats.

The graph also includes the number of linear systems resolved by the full-order model, in the studied range of saturated hydraulic conductivities.

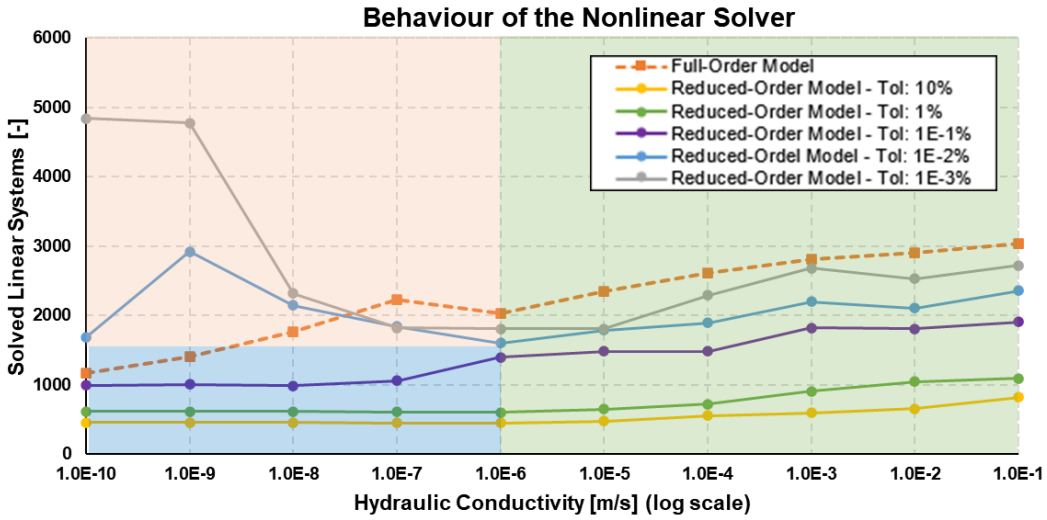

Figure 3-8: Behaviour of the nonlinear solver of the reduced-order models defined by different tolerances, with respect to the behaviour of the full-order model, for different values of saturated hydraulic conductivity logarithmically spaced between 10^{-10} to 10^{-1} ms^{-1}.

The graph clearly proves a previously mentioned hypothesis, which inferred that for a specific value of saturated hydraulic conductivity, as the tolerance shrinks in orders of magnitude (and therefore the size of the reduced basis increases), the reduced-order model requires more iterations in order to converge. In other words, as the nonlinear system that needs solving at every time step is enriched, the iterative process has a harder time of driving the governing equation to zero.

By comparing the behaviour of the nonlinear system in the reduced-order models to the behaviour of the full-order model, the following observations can be made in relation to the differently coloured areas found in the graph:

1. **Green Area:** For all of the range of saturated hydraulic conductivities superior to 10^{-6} ms^{-1}, the studied reduced-order models, (which essentially differ due to their unique reduced bases, defined by the tolerances), do not require a higher number of iterations to solve the nonlinear problem, than the full-order model.

   It is obvious that the values of tolerance that constitute each of the reduced bases play an important role in this, and that, as previously described in the first subsections of the chapter, as the size of the reduced basis increases, so does the number of iterations.

   It is therefore important to investigate how the nonlinear solver behaves when the tolerance that defines the reduced basis is much lower.

   shows a comparison between the number of iterations that are needed to solve the full-order model, against the iterations needed for the reduced-order model, for a tolerance of 10^{-10}%.

   In *Figure 3-7* it was demonstrated that a tolerance of 10^{-10}% generates a reduced basis formed by 63% of the vector directions that describe the solutions of the full-order model.

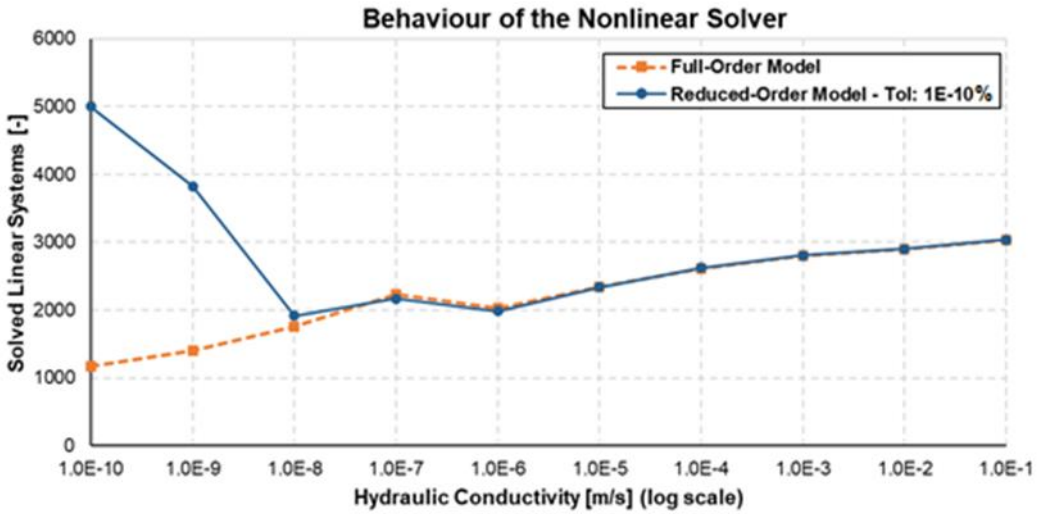

Figure 3-9: Comparison between the number of linear systems solved by the full-order model and by the reduced-order model, defined by a tolerance of 10-10%, for different values of saturated hydraulic conductivity.

The above graph verifies that for higher values of saturated hydraulic conductivity than 10^{-6} ms^{-1}, the Picard method (the nonlinear solver) performs as well for the reduced-order model as the full-order model.

In fact, it can be seen that for the saturated hydraulic conductivity value of 10^{-7} ms^{-1}, the reduced-order model is also not penalized by the nonlinear solver.

For the value of 10^{-8} ms^{-1}, the number of systems that need to be solved by the reduced-order model is slightly higher than the quantity of the full-order model. This phenomenon can also be observed happening in *Figure 3-8* for the tolerance values of 10^{-2} and 10^{-3}%.

So, in essence, it can be seen that the nonlinear solver performs as well in the reduced-order model, as it does in the case of the full-order model, for values of k_{sat} equal or higher than 10^{-7}%.

This revelation is important, given that the Picard method is the worst nonlinear solver in terms of efficiency and accuracy, as it tends to need a great number of iterations in order to reach convergence [36]. The reason for its usage in the European project is that it is the easiest to implement [17].

So, given that, performance-wise, the Picard nonlinear method tends to be poor, the fact that the number of iterations that the reduced-order models need to go through in order to reach convergence, with respect to the full-order model, is similar, is good news.

For the sake of clarity, the studied graph is newly shown (Figure 3-10), with the corresponding adjustments made to the coloured areas, and including the reduced-order model for a tolerance of 10^{-10}%.

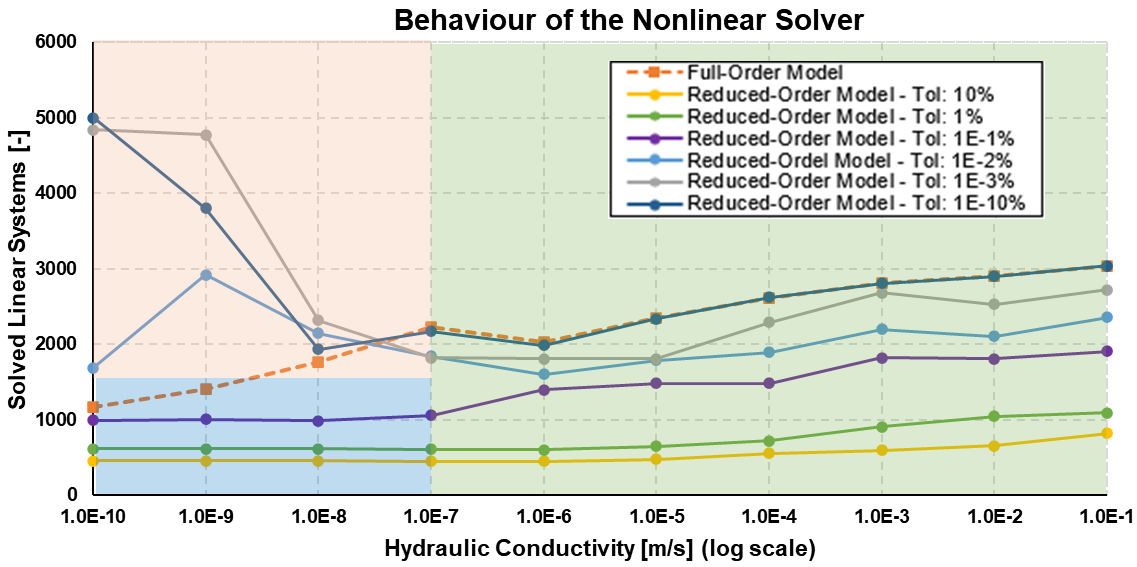

Figure 3-10: Behaviour of the nonlinear solver of the reduced-order models defined by different tolerances, with respect to the behaviour of the full-order model, for different values of saturated hydraulic conductivity.

2. **Blue Area:** In *Figure 3-10* it is easily observed that the blue area shows that for high tolerances (from 10^{-1} until 10%) the number of systems that the reduced-order model needs to resolve is approximately constant for low values of k_{sat}.

   Following up on the first observation made about *Figure 3-8*, the low variability of the number of iterations between different values of saturated hydraulic conductivity for high values of tolerance can indicate that, as the size of the reduced basis is significantly low, consisting of, at most, 1% of the original size (as seen in *Figure 3-7*), the model has an easier job of making the governing equation converge, as it accepts a wider scope of values to be reasonable.

   Hence, if the Picard method applied to solve the linear system (in a certain temporal node) finds a solution which produces a convergence error in the governing equation which is lower than the tolerance, the solution is considered acceptable, and no more linear systems are resolved in that temporal node.

   In other words, the reduced-order models created for such high tolerance values lack precision.

3. **Red Area:** The red-coloured area in *Figure 3-10* shows how the Picard solver behaves for low values of k_{sat} and lower tolerances than those seen in the *Blue Area*.

   It is immediately noticeable that the tendency found in the *Blue Area*, which shows that the number of linear system solved by the reduced-order models are approximately constant for the different values of saturated hydraulic conductivity, does not hold true when the tolerance values pass under 10^{-1}%. As the tolerance decreases, the number of Picard iterations become ever more variable.

For lower tolerance values than 10^{-1}% and lower values of saturated hydraulic conductivity than 10^{-7} ms^{-1}, the reduced-order model needs to solve more linear systems than the full-order model.

However, this does not imply that the computational cost of the reduced-order model for low values of saturated hydraulic conductivity is higher than that of the full-order model. As stated in the initial formulation of *Equation 3-1*, the size of the system of equations is more influential towards the cost than the number of iterations of the system.

The study of how the nonlinear system behaves in solving the full-order model in comparison to the different reduced-order models, lead to concluding that its performance is equal for both models when the saturated hydraulic conductivity is higher than 10^{-7} ms^{-1}. In contrast, for lower values, the number of systems solved by the reduced-order model becomes increasingly variable for gradually lower values of tolerance, passing above the number of iterations needed by the full-order model.

Based on these discoveries, the need arises to study more closely how the solver behaves for low values of k_{sat}.

Bearing this in mind, a new snapshot matrix is assembled, comprising 10 values of k_{sat} between 10^{-9} and 10^{-7} ms^{-1}. The idea is to carry out the same analysis, for the new matrix of sampled solutions.

As the new range of values is much smaller, the 10 values of k_{sat} sampled solutions (or snapshots) are selected such that they are equally spaced out in the chosen range (as a posed to the logarithmic equispaced values employed until now). These values are the following:

Table 3-10: Values of saturated hydraulic conductivity that correspond to the snapshots of the new analysis.

Saturated Hydraulic Conductivity [ms^{-1}]				
10^{-9}	$1.2 \cdot 10^{-8}$	$2.3 \cdot 10^{-8}$	$3.4 \cdot 10^{-8}$	$4.5 \cdot 10^{-8}$
$5.6 \cdot 10^{-8}$	$6.7 \cdot 10^{-8}$	$7.8 \cdot 10^{-8}$	$8.9 \cdot 10^{-8}$	10^{-7}

So, the new delimitations of the problem, which build upon the points seen in *Study Delimitations*, while replacing those which become contradictory, are:

- The analysis is implemented for 10 different values of the saturated hydraulic conductivity parameter, equally spaced in the range from 10^{-9} to 10^{-7} ms^{-1}.
- The snapshot matrix generated (in *Snapshots.py*) comprises the solution matrices of the full-order model for the 10 chosen values of saturated hydraulic conductivity, stated in the point above.

Given that a new matrix of snapshots has been defined for the analysis (*Table 3-10*), the opportunity arises to investigate an idea put forward at the beginning of the subsection, to learn whether the size of the reduced basis is influenced by how the matrix of snapshots is assembled.

The first thing that has to be done is to newly follow the process described in *Figure 3-2*, in order to answer the same focus questions posed in the original analysis, and collect enough information to be able to continue the work followed from then on.

Relevant to the hypothesis discussed in the previous paragraphs, a first observation that can be made is that for the new matrix of snapshots, the relationship between the tolerance and the percentage size of the reduced basis has changed, as can be seen in *Figure 3-11*.

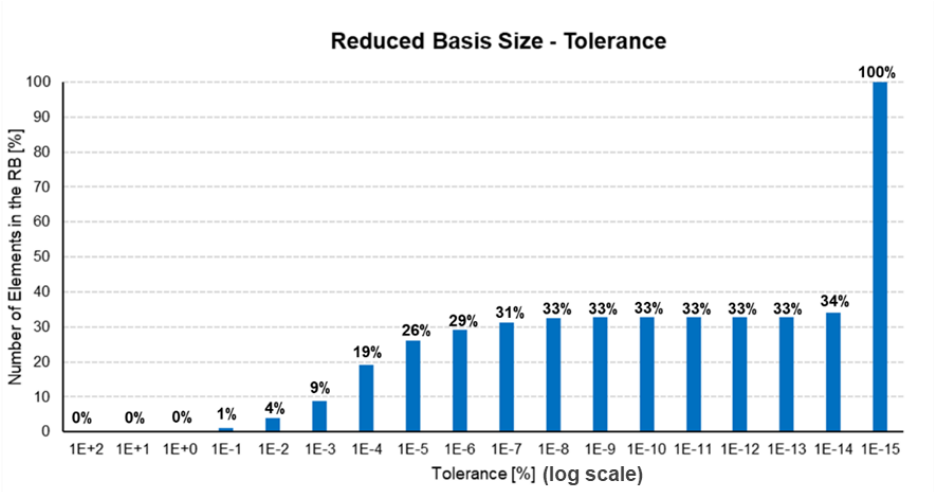

Figure 3-11: Relationship between the percentage size of the reduced basis and the tolerance, for a matrix of snapshots generated from a range of saturated hydraulic conductivity from 10^{-9} to 10^{-7} ms^{-1}.

By comparing this bar chart with the one found in *Figure 3-7*, it is possible to distinguish two main differences with respect to the relationship between the size of the reduced basis and the tolerance, for the differently defined snapshot matrices:

1. For the newly assembled snapshot matrix, the reduced basis reaches its full size for a tolerance of 10^{-15}, instead of 10^{-16}.
2. For the new matrix, only 34% of the vectors that define the matrix of the full-order model's solutions (snapshot matrix) are needed for a tolerance of 10^{-14}%. In the case of the matrix defined for the k_{sat} value range from 10^{-10} to 10^{-1}, 64% of the vector directions were necessary to describe the model's solutions to the same level of precision.

A provisional conclusion that can be reached from this is that a reduced basis can be defined, to the same precision, with fewer elements if the snapshot matrix is generated by solving a set of full-order model solutions whose range of parametric values is smaller (more narrowly focused).

In order to corroborate this hypothesis, at least one more snapshot matrix needs to be assembled. To do so, it is necessary to formulate a focus question that concentrates on concluding this matter:

- **Final Focus Question 1:** What relationship is there between the number of elements needed in order to define a reduced basis, and the range of parametric values for which the matrix of snapshots is formed?

The answer to the question will be presented in the next subsection (*Final Set of Results*).

Concerning how the nonlinear solver behaves in relation to the new range of saturated hydraulic conductivity values, the graph in *Figure 3-12* is plotted.

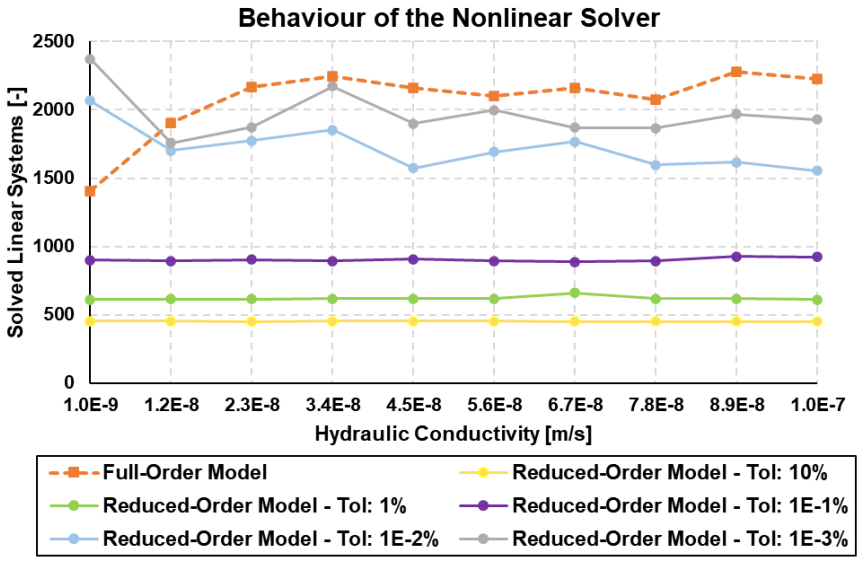

Figure 3-12: Behaviour of the nonlinear solver of the reduced-order models defined by different tolerances, with respect to the behaviour of the full-order model, for different values of saturated hydraulic conductivity for the newly defined snapshot matrix ranging from 10^{-9} to 10^{-7} ms^{-1}.

A first observation that can be made is that the number of iterations solved by a reduced-order model defined by a tolerance of, for example, 10^{-3}% is not the same for the k_{sat} value of 10^{-9} ms^{-1} in the new snapshot matrix (*Figure 3-12*) than it was for the original snapshot matrix (*Figure 3-12*).

The following table quantifies this comparison for the k_{sat} values of 10^{-9} and 10^{-7} ms^{-1}, which are the only two solutions that are sampled in both snapshot matrices.

Table 3-11: Comparison between the number of Picard iterations solved by different reduced bases generated from truncating two distinct snapshot matrices.

Snapshot matrix				
Tolerance [%]	**10^{-10} to 10^{-1} ms^{-1} (10 logarithmically spaced values)**		**10^{-9} to 10^{-7} ms^{-1} (10 equally spaced values)**	
	$k_{sat}=10^{-9}$ ms^{-1}	**$k_{sat}=10^{-7}$ ms^{-1}**	**$k_{sat}=10^{-9}$ ms^{-1}**	**$k_{sat}=10^{-7}$ ms^{-1}**
10	452	450	452	450
1	612	610	612	610
10^{-1}	1001	1054	900	924
10^{-2}	2920	1839	2067	1553
10^{-3}	4773	1822	2374	1930

Table 3-11 shows that the nonlinear solver has less difficulty in reaching convergence when snapshot matrix is assembled for a smaller range of k_{sat} values.

This can be understood by looking back at *Reduced Basis*. The reduced basis has the task of reproducing, (to the level of precision defined by the tolerance), the solutions of the full-order model that constitute the matrix of snapshots, and to estimate the solutions for any k_{sat} value found inside of the studied range (sampled or not), in basis of the sampled information.

It is therefore intuitive that the vector directions that define a snapshot matrix that centres upon a smaller domain of k_{sat} values, will have a greater capacity of predicting the solutions for that range of values, than what is expected of the vector directions that define a snapshot matrix that englobes a much larger variety of values.

Another statement that can be made based on *Figure 3-12* is that it proves that the reduced-order model has more difficulty in solving the nonlinear system than the full-order model, for lower k_{sat} values than 10^{-8}, but that the difference between the iterations undergone by the models is much smaller when considering snapshot matrices that are more narrowly focused.

On the other hand, for high values of tolerance, the number of systems resolved for the same value of k_{sat} is coincidental when it is evaluated using different snapshot matrices.

Joining this last idea with the observations made in *Blue Area*, it can be can be concluded that if the number of systems that are solved by a reduced-order model defined by a high-tolerance, are the same for different k_{sat} values lower than 10^{-7} ms^{-1}, the tolerance is too low, and the model is not sensitive enough to be affected by changes in the value of k_{sat}, nor by the change of the assembled snapshot matrix.

Given that the number of systems has been defined as directly proportional to the computational cost, it can be concluded that a lower computational cost is not always optimal. Taking this into consideration, it is clear that for the analysis to be more beneficial towards the European project, there is a great need for quantifying the relationship between the computational cost and the tolerance, in order to find how the reduced-order model has to be defined towards minimizing the computational cost while maintaining an acceptable level of information.

To accomplish this goal, the subsequent question is posed:

- **Final Focus Question 2: What is the relationship between the computational cost of the full-order and the reduced-order model, for different tolerances?**

Finally, a last question that needs to be answered, which focuses on one of the main objectives of the project, is the following:

- **Final Focus Question 3: Is it beneficial to reduce the order of the model?**

3.3.4. Final Set of Results

After a long number of simulations, discussed in the previous subsection, it is time to find the answers to the final set of focus questions, also formulated in the last pages, and draw general conclusions from the whole analysis.

Final Focus Question 1	**What relationship is there between the number of elements needed in order to define a reduced basis, and the range of parametric values for which the matrix of snapshots is formed?**

In the previous subsection, a transition was made from using a snapshot matrix comprising 10 solutions of the full-order model, in the vast range of saturated hydraulic conductivity values between 10^{-10} to 10^{-1} ms^{-1}, to using a matrix formed by 10 solutions in a much smaller range, from 10^{-9} to 10^{-7}.

By observing the relationship between the tolerance applied to the decomposed product of matrices and the resulting size of the truncated (or reduced) basis, a hypothesis was put forward, which pointed at the possibility of an existing correlation between the size of the evaluated parameter value range, and the number of elements that is then needed to characterise the reduced basis, in order to conserve the same amount of information from the solutions of the original-sized model.

Put more simply, the idea is to validate if it is true that fewer vector directions are needed to define a reduced-basis that has been created from the decomposition of the full-order model solutions matrices evaluated for smaller value ranges, for a same value of tolerance.

To corroborate this hypothesis, the range of values is modified once again in *Snapshots.py*. The new matrix, also constituted by solving the full-order model for 10 equally spaced values of saturated hydraulic conductivity, ranges from k_{sat} values of 10^{-9} to 10^{-8} ms^{-1}.

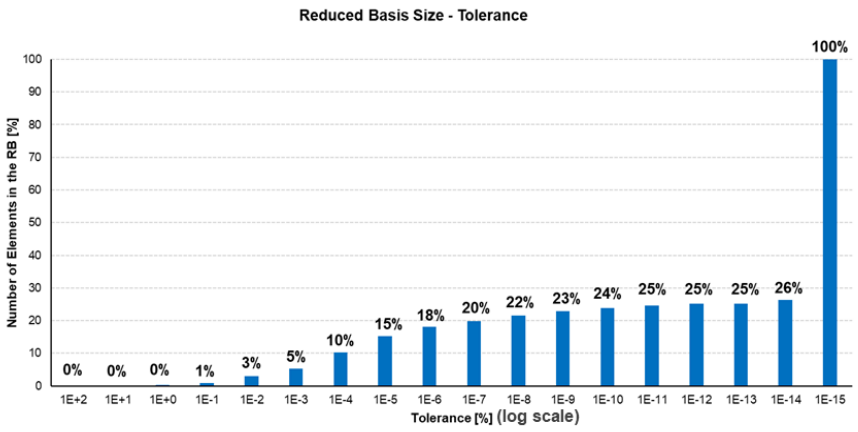

Figure 3-13: Relationship between the percentage size of the reduced basis and the tolerance, for a matrix of snapshots generated from a range of saturated hydraulic conductivity from 10^{-9} to 10^{-8} ms^{-1}.

As can be observed, the new range of values confirms the posed hypothesis.

The following table offers a clearer visualization of the validated correlation, by comparing the percentage number of elements needed to create the reduced basis in every case, for different values of the applied tolerance.

Table 3-12: Relationship between the range of values evaluated in the snapshot matrix and the number of elements needed to describe the resulting reduced basis.

Percentage number of elements needed to define the reduced basis		**Saturated Hydraulic Conductivity Value Range [ms^{-1}]**		
		$(10^{-10} – 10^{-1})$ ms^{-1}	$(10^{-9} – 10^{-7})$ ms^{-1}	$(10^{-9} – 10^{-8})$ ms^{-1}
Tolerance [%]	10^{-1} %	1%	1%	1%
	10^{-3} %	12%	9%	5%
	10^{-6} %	54%	29%	18%
	10^{-10} %	63%	33%	24%
	10^{-14} %	64%	34%	26%

Once again it can be seen that higher values of tolerance are less sensitive to changes of the snapshot matrix.

Having also concluded during the previous subsection that for narrower focused ranges of k_{sat} values, the nonlinear solver has to iterate less to reach convergence, it is clear that the computational cost of the reduced-order model will also be lower for a smaller range of k_{sat}.

This general conclusion on the importance of defining a delimited matrix of snapshots makes it necessary for the European project to define such a matrix that is limited to an interval of k_{sat} values that can be expected in a tailings dam.

Final Focus Question 2	**What is the relationship between the computational cost of the full-order and the reduced-order model, for different tolerances?**

Figure 3-14 compares the computational cost between the full-order model and the reduced-order model for a tolerance value of 10^{-15} %. It is worth remembering that in both cases, the size of the linear system to be resolved is of 1036 vector directions.

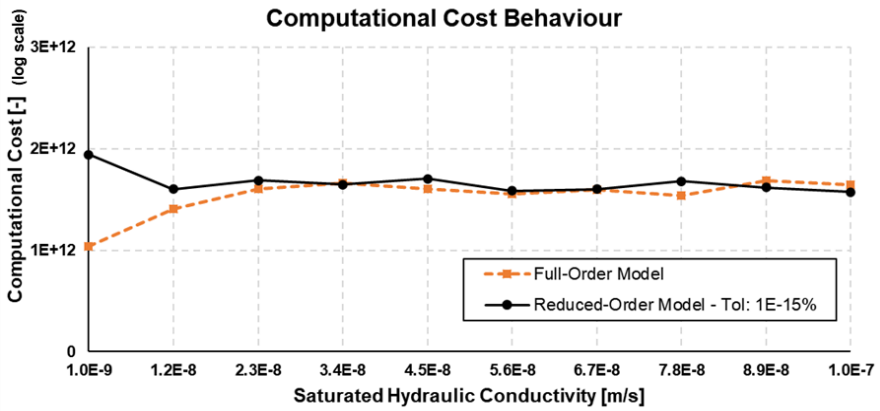

Figure 3-14: Comparison between the computational cost of the full-order model and the reduced-order model for a tolerance value of 10^{-15} %, for a k_{sat} range from 10^{-9} to 10^{-7} ms^{-1}.

The graph shows that even though the computational cost of both models has the same order of magnitude, the reduced-order model is, generally, computationally more expensive than the full-order model. This can be understood by remembering how *Equation 3-1* is defined.

So, while the system that both models have to resolve is equal is size, the number of iterations they need in order to do so is different.

The relationship between the computational cost of the models follows the same patterns described for the behaviour of the nonlinear solver:

- For higher k_{sat} values than 10^{-8} ms^{-1}, the computational cost of the reduced-order model behaves similarly to that of the full model.
- For equal or lower values than 10^{-8} ms^{-1}, the computational cost of the reduced-order model grows significantly above that of the full-order model, doubling it for a k_{sat} value of 10^{-9} ms^{-1}.

Having found that the computational cost of the reduced-order model is generally slightly higher than the original, for the complete size made out of 1036 vector directions, it is important to see how this cost behaves in response to the increase of the tolerance's value.

The following graph (*Figure 3-15*) shows how the computational cost of the reduced-order model varies in function of the defined tolerance. It is important to bear in mind that the vertical axis is represented in a base-10 logarithmic scale. In the legend, RM means reduced-order model.

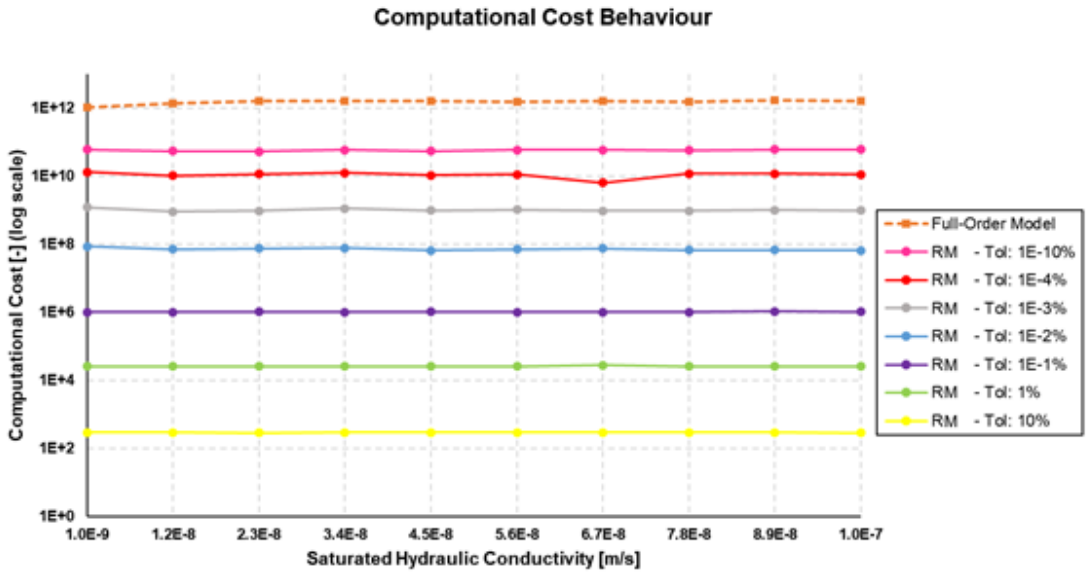

Figure 3-15: Comparison between the computational cost of the full-order model and the computational cost of the reduced-order model, for different percentage values of tolerance.

It is plainly visible that for gradually higher percentage values of tolerance, the computational cost decreases in orders of magnitude.

The graph makes it possible to learn how the computational cost decreases in function of the tolerance, as can be perceived more clearly with the following examples:

- When the reduced basis is truncated to a tolerance of 10^{-10}%, it is constituted by a third of the original size (see *Figure 3-7*), and its computational cost is half that of the original model.

- When the reduced basis is truncated to a tolerance of 10^{-4}%, it is constituted by a fifth of the original size (see *Figure 3-7*), and its computational cost is 1% of the original model.

A first notion could be to think that the best scenario is produced by the highest tolerance value, as the computational cost is the lowest.

Nevertheless, it has also been demonstrated that a higher tolerance value signifies a lower precision, and therefore, a balance between the two ideas has to be found.

The ideal result is clearly for the computational cost of the model to be minimised, while maximising its level of precision.

Towards having the capacity to choose which tolerance yields this optimal result, it is necessary to find a way to actually quantify the reduced-order model's accuracy.

Final Focus Question 3	**Is it beneficial to reduce the order of the model?**

In terms of how costly the model is, applying the POD-based technique, as described throughout the document, is clearly advantageous.

But as it has been concluded that a balance needs to be found between the computational cost and the exact accuracy of the model, the information available at the end of the current chapter is insufficient to conclude whether the reduction of the model is a viable option for the European project to continue implementing.

It is clearly necessary to carry out further analyses that permit gathering the required information.

3.4. Concluding Remarks

Having discussed the analysis extensively, it is advisable to summarize the obtained results, in order to gain a clearer understanding of the study, and to be able to connect different ideas.

Towards achieving this goal, the following table (*Table 3-13*) lists the conclusions reached throughout the analysis.

Table 3-13: Summary of the main conclusions drawn from the computational cost analysis.

<table>
<tr><th colspan="3">Conclusions</th></tr>
<tr><td>3-1</td><td colspan="2">Creating a reduced basis for reasonably low values of tolerance makes it possible to create a reduced-order model that is capable of predicting the solutions of the original model to a high level of precision, while having to solve much smaller systems of equations.</td></tr>
<tr><td>3-2</td><td colspan="2">The number of iterations needed by the model to reach convergence in a time step is proportional to the size of the reduced basis, and inversely proportional to the defined tolerance.</td></tr>
<tr><td rowspan="1">$k_{sat} > 10^{-7}$ ms^{-1}</td><td>3-3</td><td>The nonlinear solver (Picard iterative method) performs as well for the reduced-order model as it does for the full-order model.</td></tr>
<tr><td rowspan="2">$k_{sat} \leq 10^{-7}$ ms^{-1}</td><td>3-4</td><td>If the tolerance that defines the reduced-order model is too high, the model presents a deficient level of precision, which is perceptible thanks to different phenomena:
The function that describes how many iterations of the nonlinear solver are needed for the reduced model to converge hardly oscillates for the successive k_{sat} values in study.
The number of vector directions needed to describe the reduced basis to a level of precision is very similar for different snapshot matrices.</td></tr>
<tr><td>3-5</td><td>For reasonably low values of tolerance, the number of systems that the reduced-order model has to resolve exceeds that of the full-sized model, and becomes increasingly variable as the tolerance decreases.
The performance of the nonlinear solver degrades for lower values of k_{sat}.</td></tr>
<tr><td>3-6</td><td colspan="2">A snapshots matrix assembled for smaller ranges of k_{sat} values yields better results in terms of the nonlinear solver and the computational cost.</td></tr>
<tr><td>3-7</td><td colspan="2">A lower computational cost is not always optimal. It is necessary to find a balance between the cost of the reduced-order model and the accuracy with which it predicts the pore water pressure in the whole dam.</td></tr>
</table>

The last conclusion makes it clear that to achieve the objectives set out by the end-of-degree project, another analysis has to be carried out, focused on quantifying the accuracy of the reduced-order model and studying its behaviour through further simulations.

Given the close relationship between the analyses undertaken in this chapter and the next, some of the conclusions drawn in *Table 3-13* may be expanded upon in the following chapter.

4. Parametric Analysis

After concluding in the previous analysis (*Computational Cost Analysis*) that there is a need to quantify how well the reduced-order model predicts the values of pore water pressure across the entire dam, the current chapter aims to propose a method that accomplishes this objective, and analyse how this accuracy behaves in different circumstances.

It is important to bear in mind that, as the aim of the European project is to prevent further casualties related to the failure of tailings dams across the globe, the accuracy with which the designed model estimates the pore water pressure is a highly delicate matter.

A fundamental assumption that has to be made in order for the proposed method to make sense is that the full-order model is a sufficiently accurate portrayal of reality.

Bearing this in mind, the calibre of the reduced-order model's performance will be determined by how well the obtained values of the pore water pressure approximate the results obtained in the full-order model.

In other words, the accuracy of the reduced-order model will be measured by the difference between the solutions that it presents, and the solutions of the full-order model. The methodology followed to calculate the mentioned difference (or error) will be approached in the following subsection.

So, the idea around which the chapter is constructed is that of studying how accurately the reduced-order model predicts the evolution of the pore water pressure of the dam, in function of how the reduced basis of the model is defined.

The factors that define a reduced basis derive from:

- The characteristics of the matrix of snapshots, whose decomposition results in the construction of the eigenvector matrices and the singular values matrix.
- The criteria used for the truncation of the singular values matrix, which then transforms the left eigenvector matrix into the reduced basis.

Regarding the first point, the information held by the snapshot matrix depends on numerous components, related to the construction of uncoupled problem.

It is important to remember that the snapshot matrix is one that stores the solution matrices resulting from solving the full-order problem a certain number of times, for different values of a certain parameter.

Having this in mind, it can be found that an important component which influences the reduced basis is the parametric delimitations used to build the snapshot matrix. Specifically, the range of parametric values for which the full-order model is solved, and the number of points in said range.

The idea is to run simulations that analyse how the reduced basis behaves (how it varies) for different ranges and numbers of points in the range, of a chosen parameter.

In order to be more efficient with the experimental work, the parameter selected for this analysis is the saturated hydraulic conductivity (k_{sat}). As explained previously in the document, the code that solves the full-order problem a given number of times, and stores the solutions (or snapshots) in a matrix, is configured to follow this procedure with the saturated hydraulic conductivity parameter (see *Snapshots.py*).

Another component that affects the matrix of snapshots is the definition of the uncoupled problem. In other words, the values of the physical and numerical parameters that are involved in the procedure explained in *Figure 2-2*.

Towards studying how the variation of a problem parameter affects the reduced basis, the chosen variable is the total time of study (or time dimension).

A fuller description of the considerations made to study the parameters will be given in a later subsection (*Study Parameters*).

As for the second listed factor, the truncation of the left eigenvector matrix is performed in response to the tolerance defined in the previous chapter (see *Equation 3-2*).

Taking this into consideration, the analysis will need to take into account how the accuracy of the reduced model behaves for reduced bases defined by different values of tolerance.

So, in summary, the present analysis will study the reduced bases that the uncoupled problem produces thanks to the application of the singular value decomposition, for different conditions defined by the variation of the following parameters:

Table 4-1: Parameters studied in the analysis.

Study Parameters	Saturated Hydraulic Conductivity (ksat)
	Time Dimension (T)
	Tolerance (tol)

An important comment that has to be made, concerning the parameter analysis carried out in this chapter is that the different scenarios that will be presented by the variation of the saturated hydraulic conductivity and the time dimension, will be studied in independent sets of simulations, following the workflow seen in the computational cost analysis *Figure 3-4*.

Meanwhile, the tolerance applied to the singular values matrix, is partially studied in both of the previously mentioned subsections. The reasoning for this is that the variation of the tolerance

influences the definition of the reduced basis in a more substantial way, than the changes caused by the other two parameters, as it directly determines its size.

A complication encountered by the proposed analysis is that the process of calculating the global error (the error in every spatial node of the dam) for numerous different problem conditions would be very costly in terms of time.

Having already discussed the limitation set on the work by the run time of the codes (see *Table 3-3*), it is necessary to simplify the dimensions of the study, towards analysing the error produced by the reduced-order model at a smaller scale.

Based on the conclusions drawn from the analysis at a local level, it might be possible to then extrapolate the results to be representative of the whole dimension of the dam.

To carry out the mentioned simplification, a single spatial point of the dam is selected for the analysis, reducing the dam's dimension to a 1-dimensional space.

By doing this, the solutions of the model pass from being matrices that hold the pore water pressure of every point of the dam, for all of the studied time, to vectors that consist of the pore water pressure of a single point of the dam, for every temporal node (or time step).

The following diagram shows the transformation undergone by the matrix of solutions of the model, from a 2-dimensional space to a 1-dimensional space. The dimensions of the 2-dimensional model's solutions matrix coincide with those described in the *Uncoupled problem*.

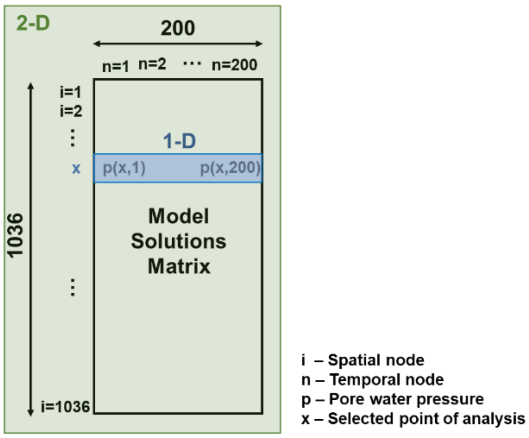

Figure 4-1: Diagram showing the conversion of the solutions matrix to a 1-dimensional space.

It is important to note that, as the dam is discretized using the Finite Element Method (FEM), the selected point will correspond to one of the nodes that conform the dam's mesh.

The criteria followed in the selection of the spatial node subjected to this analysis are the following:

- It should be closer to the downstream boundary than to the upstream boundary, so that its pressure state is influenced by a considerable part of the dam's structure. Therefore, it will be more representative of other spatial points in the dam.

- Its position should not be such that it is never saturated (i.e. that the flow of water is unable to reach its height).
- It should not be found on a boundary of the dam, so it is not subjected to boundary conditions.

The chosen point is number 63 (of the 1036 spatial points that form the discretization of the dam). The following illustration shows the position of the selected point, marked with a pink dot, in a 2-dimensional representation of the dam's domain. The program used to visualize the dam's discretization is ParaView [37].

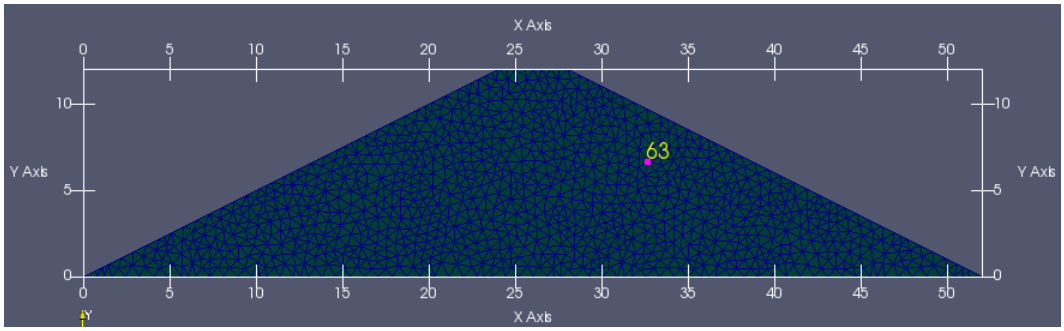

Figure 4-2: 2-dimensional illustration of the discrete mesh of the dam, with the position of the selected spatial node. ParaView 5.2.0.

Concentrating on a single node facilitates the ability to compare the solutions obtained with the full-order model against those of the reduced-order model, as graphs can be plotted showing the evolution of the pore water pressure in the point of study.

For example, the following graph shows the evolution of the pore water pressure in node 63, found by the full-order model and by the reduced-order model defined by a tolerance of 10^{-3}%, for a k_{sat} value of 10^{-9} ms^{-1} and a time dimension of 20 days.

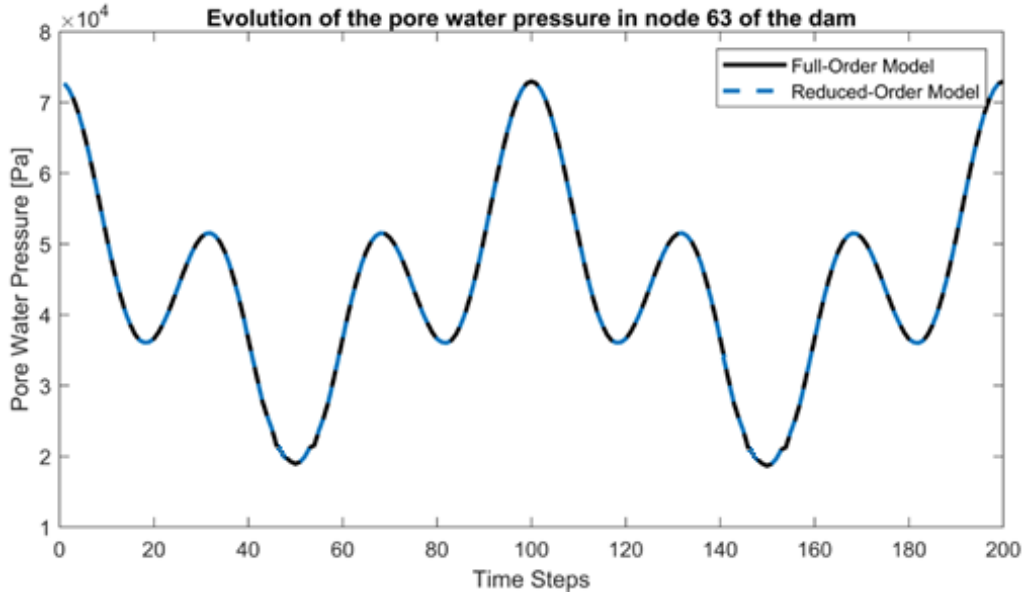

Figure 4-3: Pore water pressure versus Time steps graph, for a certain parametric scenario in spatial node 63. MATLAB.

At a first glance of the graph, the solutions derived from the full-order model and reduced-order model seem equivalent, but the order of magnitude of the graph has to be taken into account. The pore water pressure axis is multiplied by 10^4, so the differences between the solutions of both models are not visible.

An important feature that can be observed in the graph is the relation held between the selected spatial node and the phreatic level of the water.

Given that the pore water pressure fluctuates between the approximate values of $2 \cdot 10^4$ and $7 \cdot 10^4$ pascal, it can be concluded that spatial point 63 is never found above the water table (or phreatic level) of the dam's embankment, as positive pore water pressure values indicate total saturation [38].

As the selected point of analysis is only representative of the saturated portion of the dam, the study needs to also study how the accuracy of the reduced-order model behaves in a spatial point which presents unsaturated conditions.

The inclusion of the unsaturated area of the dam is particularly important for the analysis of the coupled problem, as most of the processes involving soil-water interaction occur in unsaturated conditions [39][40].

In order to fulfil this, a second spatial node is selected, following the same criteria as the first, but adding a last guideline:

- It should not be fully saturated at all times.

The point selected is node 275, found approximately 3 metres above node 63. Point 275 is found part-time saturated and part-time unsaturated.

Having included another spatial point of study in the analysis, it is clear that other parametric scenarios can be studied, in relation to the saturation of the point in study.

In summary, the main objective of the analysis is to study the accuracy with which the reduced-order model estimates the pore water pressure of the dam, in relation to different problem conditions (or parametric scenarios), created by a series of study parameters.

As the structure of the current chapter mirrors the sequence followed in the *Computational Cost Analysis*, the purpose of each subsection does not require clarification.

4.1. Methodology

Having proposed that the accuracy of the reduced-order model solutions be quantified by the difference between the solutions of the reduced-order model and the full-order model, the current subsection will describe how this error is defined and calculated.

As the analysis has been simplified to be carried out in a 1-dimensional space, the mentioned error will relate to the difference between the resulting values of pore water pressure for each model at a single spatial point, at every time step.

In order to put this into effect, the first step will be to execute the code corresponding to each model (*Hydro_Unsaturated.py* for the full-order model and *RB_Solver.py* for the reduced-order model), for a certain parametric scenario, bearing in mind that the defined problem conditions should coincide for both models.

The resulting solutions consist of two matrices (one per model) that hold the solutions of the pore water pressure in every spatial point of the discretized dam, for all the studied time steps.

Then, the solutions pertaining to every point of the dam, except the selected node of analysis, are discarded, converting the solutions of each model to a 1-dimensional space, as shown in *Figure 4-1*.

The following step is to subtract the solution functions obtained in each model at every temporal node, as examplified in *Figure 4-2*. In order to avoid negative results, the subtracted values are squared.

Next, a numerical integration method is applied to the squared subtracted solutions, in order to get a single number. The numerical method chosen to carry out the integration is called the 'Composite Trapezoidal Rule', which is especially suitable when the interval to be integrated is large, as it presents a fairly low error, compared to other methods [41].

As the dimension on which the integral wants to be applied is constituted by 200 time steps, which each consist of 8640 seconds, the selected method is appropriate for the task.

The expression of the 'Composite Trapezoidal Rule' is the following [41][42]:

$$I = \int_a^b f(x)dx = \frac{h}{2}\left(f(x_0) + 2\sum_{i=1}^{m-1} f(x_i) + f(x_m)\right) + E_m^T \qquad (Equation\ 4-1)$$

Where,

h – Size of each subinterval [-]

m – Total dimension to which the integral is applied [-]

E_m^T – Error caused by the method [-]

By introducing the values of the current study to *Equation 4-1*, the expression becomes:

$$I = \frac{8640}{2}\left(f(x_1) + 2\sum_{i=2}^{199} f(x_i) + f(x_{200})\right)$$

The error caused by the method is not taken into account.

After integrating the squared subtracted solutions, the following step is to apply the square root to the obtained number (to counter the previously implemented exponent). The result is a scalar whose physical meaning is that of the absolute error of the reduced-order model's performance.

The reason for it being absolute is that it has units of pressure.

Towards obtaining a relative error, (which is easier to work with, given that it has no units, and can be represented in decimal form), it has to be divided by a reference quantity, which in this case is found by integrating the values of the temporal nodes of the full-order model's function.

The below chart (*Figure 4-4)* summarises the steps followed in order to calculate the relative error caused by the reduced-order model in spatial node 63, and shows the transformations undergone. The same procedure is also followed for point 275, for unsaturated conditions.

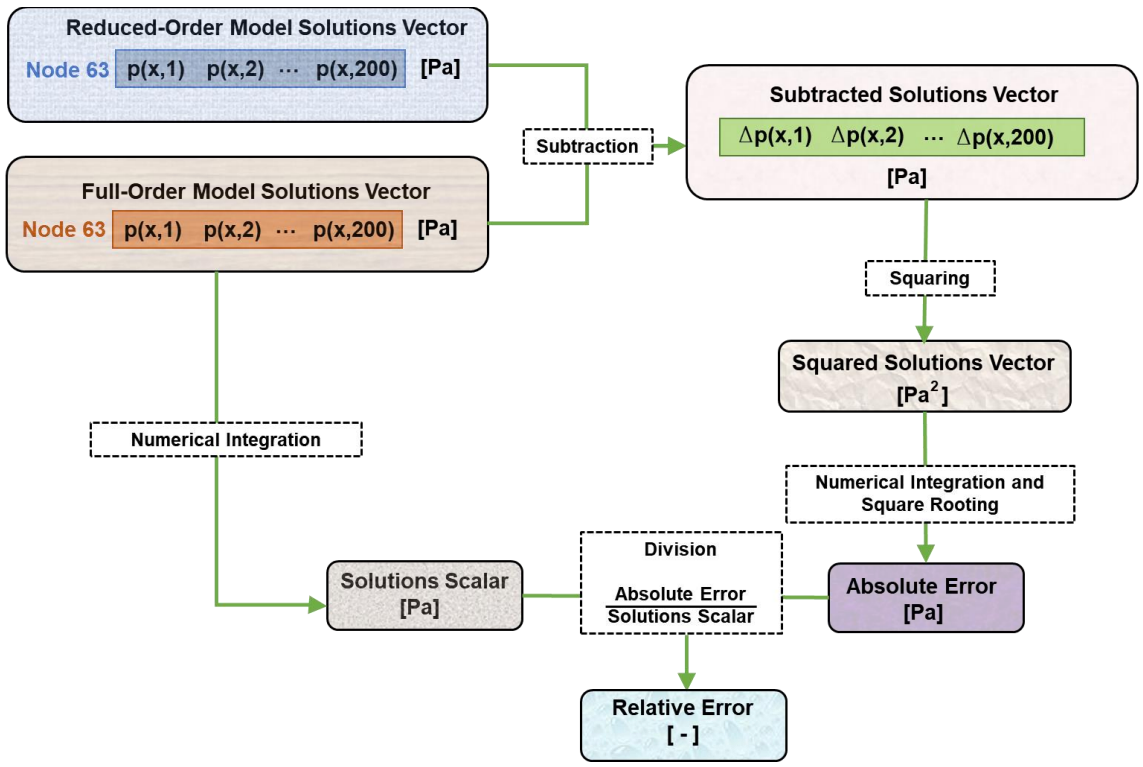

Figure 4-4: Computation of the relative error produced by the reduced-order model when predicting the evolution of the pore water pressure at a single spatial node of the dam.

To clarify, the colours present in the diagram are used for aesthetics. They do not carry additional information.

In short, the methodology described in this subsection makes it possible to take two vastly complex solution matrices, and collapse all of the difference between them into a scalar, which represents the relative error of the reduced-order model's performance with respect to that of the full-order model, which has been taken as reference.

As the analysis focuses on measuring the error caused by the reduced-order model for different problem conditions, it is essential to define the margin of error set on the analysis.

In other words, in order for the study to be conclusive, it is necessary to establish the maximum value that the relative error can have for it to still be considered acceptable.

Given that the long-term objective of the uncoupled model is to estimate the pore water pressure in the entire dam, from the monitoring network installed on the dam, the margin of acceptable error will be related to the instrumental accuracy of the sensors.

Most literature indicates that piezometers, which are currently the most common devices used to monitor the pressure head[9] levels in embankment dams [8], tend to have an inbuilt accuracy of around 0.001 to 0.002 meters [43][44][45]. Meanwhile, other sources point at a higher value of 0.005 metres [46]. Nevertheless, there is a consensus about the maximum accuracy being generally limited to 1 millimetre.

As the model works with pore water pressure (P), it is important to show that there is a directly proportional relationship between this concept and that of pressure head(h), found in hydrostatic conditions [47]:

$$h = \frac{p}{\gamma_w} \qquad (Equation\ 4-2)$$

Where,

h – Pressure head [m]

p – Pore water pressure [Pa]

γ_w – Specific Weight of Water [Nm^{-3}]

Bearing in mind that the instrumental accuracy has been estimated to an absolute value of 0.001 metres, and that the pressure head, as defined in *Equation 4-2* is measured in metres, the relative error can be found to be of 0.1%.

As there is no point in having a more accurate model than the measurements of the sensor network found on the dam, the aim of the analysis will be to study how the reduced-order model has to be constituted in order for the relative error of its solutions to be maximised without overpassing 0.1%.

Given that the relative error has been defined to be expressed as a decimal and not a percentage, the acceptable error for the problem has a value of 0.001.

[9] Pressure head: Height of the vertical column of a fluid over a point that has the same pressure as that caused by the overlying column.

4.2. Study Delimitations

In order to pursue the goals explained in the beginning of the chapter, there is the necessity to establish, once again, the delimitations (or assumptions) taken into account for the study:

- The problem is studied under uncoupled conditions.
- The full-order model is a sufficiently accurate portrayal of reality.
- The analysis is set in a 1-dimensional space, in the sense that the solutions of a single spatial point are studied, for the full length of time.
- The accuracy of the solutions of the reduced-order model are quantified by the relative error between them and the solutions of the full-order model.
- The errors produced by the applied numerical integration method (E_m^T) are not taken into account.
- The parameters in study are the tolerance, the saturated hydraulic conductivity, the time dimension and the saturation.

4.3. Implementation

In order to carry out the parametric analysis, in addition to the existing codes, a new script has to be written to compute the relative error, following the procedure described in *Methodology* and summarized by *Figure 4-4*.

The original version of the new code, named *ErrorQuantifier.py* and also written in the Python language, can be found in *Appendix C*.

The full list of simulations carried out in the chapter can be found in *Appendix B.1*.

The key functionalities of the code are summarized in the following table:

Table 4-2: Summary of the content of new Python code 'ErrorQuantifier.py'.

Code	Written Code
ErrorQuantifier.py	- Importation of the solutions files produced by the full-order model (by running *Hydro_Unsaturated.py*) and the reduced-order model (by running *RB_Solver.py*). - Creation of a vector that holds the solutions of the full-order model of a single spatial point (by only maintaining one row of the solutions matrix). - Creation of a vector that holds the solutions of the reduced-order model of a single spatial point (the same point as the previous variable). - Application of the basic arithmetic described in the *Methodology* subsection, and employment of the *'Composite Trapezoidal Rule'*. - Creation of a variable that holds the absolute error and another that contains the resulting relative error.

Towards having a clearer understanding, the following figure (*Figure 4-5*) shows an example of what is obtained when *ErrorQuantifier.py* is executed.

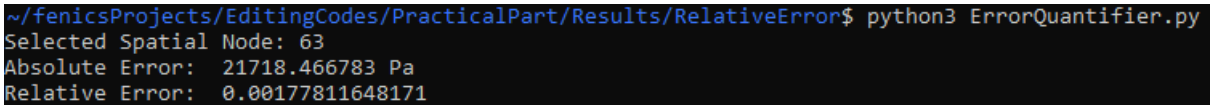

Figure 4-5: Result of running the new code ErrorQuantifier.py for a certain parametric scenario.

For clarity, the first line of blue-coloured text that can be seen in the image makes reference to the path where the code file is found. The white-coloured writing at the end of that line is the command that enables the code to be run.

The other three lines are output messages from the code, which indicate the spatial point of study, the absolute error expressed in pascal (Pa), and the relative error, which has no units.

Towards creating the different parametric scenarios discussed in the beginning of the chapter, a series of modifications have to be systematically made to the existing codes of the uncoupled problem (building on the changes seen in *Implementation*) , and the new code *ErrorQuantifier.py*.

The ensuing table (*Table 4-3*) summarizes the changes that have to be made to the codes, and also indicates to which study parameters the modifications correspond.

Table 4-3: Modifications that have to be made systematically to the codes in order to construct the different parametric scenarios of the study.

Code	Study Parameters	Systematic Modifications
Hydro_Unsaturated.py	Time Dimension	- Modification of the value of the variable that defines the total time dimension.
RB_Solver.py	Time Dimension	- Modification of the value of the variable that defines the total time dimension.
Snapshots.py	Saturated Hydraulic Conductivity	- Modification of the range of values of saturated hydraulic conductivity to be considered for the generation of the snapshot matrix.
SVD.py	Tolerance	- Modification of the value of the tolerance.
ErrorQuantifier.py	Saturation	- Modification of the spatial point of study.

4.4. Study Parameters

Having briefly discussed the use of different study parameters to provide different problem conditions for the analysis to be carried out, the current subsection will give a closer examination on the parameters and the criteria behind the scope of different values they will take into account.

As this study also follows the sequence defined and followed in the analysis of the previous chapter (see *Figure 3-4*), it will give a more compact account of the process gone through in order to draw conclusions.

As mentioned in the introductory pages of the chapter, the investigation process described in the previous paragraph will be carried out for every study parameter, meaning that each parameter will have its own subsection in which the analysis will be performed for different parametric scenarios created by the modification of the corresponding parameter.

The exception is the saturation, as it only presents two possible parametric scenarios, and is therefore easily included in every separate analysis. Doing this will permit perceiving the difference in the results obtained for the different problem conditions of the study parameters, found for the situations:

- Saturated conditions (by studying node 63 of the discretized dam)
- Unsaturated conditions (by studying node 275)

The explicit formulation of focus questions will only be included in the case that they add fundamental information on the directions to follow in the analysis. The full questions-answers cycles will not be shown in this chapter, for conciseness.

4.4.1. Tolerance

Given that the overall work hopes to explore how the computational cost can be minimised, without causing a significant threat to the accuracy of the solutions, and having defined the margin of accepted error at a decimal value of 0.001, the aim of the current subsection is to study the relationship between the tolerance and the relative error.

The motivation for analysing how the error behaves in function of the tolerance is to perceive how small the reduced basis can be (how high the tolerance can be set) without surpassing the maximum error value. It will then be important to observe how the obtained results behave with the variation of other study parameters, in order for the analysis to be representative of a larger range of possibilities.

Towards studying the different scenarios defined by the variation of the tolerance, the idea is to fix the values of the rest of study parameters, as defined in the ensuing table (*Table 4-4*).

Table 4-4: Fixed values given to the study parameters which are not variable in the current analysis.

Fixed values of the study parameters	
Saturated Hydraulic Conductivity	Range from 10^{-9} to 10^{-7} ms^{-1} (10 points in the range)
Time Dimension	20·86400 s = 20 days

As can be observed in the table, the values given to the fixed study parameters correspond to the latest values defined in the document.

Saturated Conditions

Starting with the saturated scenario (and therefore analysing spatial node 63), the following graph (*Figure 4-6*) shows how the relative error behaves for the logarithmically spaced values of tolerance between 10^{-4} and 1%.

The reason for analysing equal or higher tolerance values than 10^{-4}% is that the computational cost associated with this value is approximately 1% of the cost of the full-order model (see *Figure 3-15*).

Although a graph representing the behaviour of the relative error for the full range of possible tolerances, (from 10^{-15} to 100%), would be more ideal, it is not possible to carry out so many simulations for low values of tolerance, as demonstrated in *Table 3-3*.

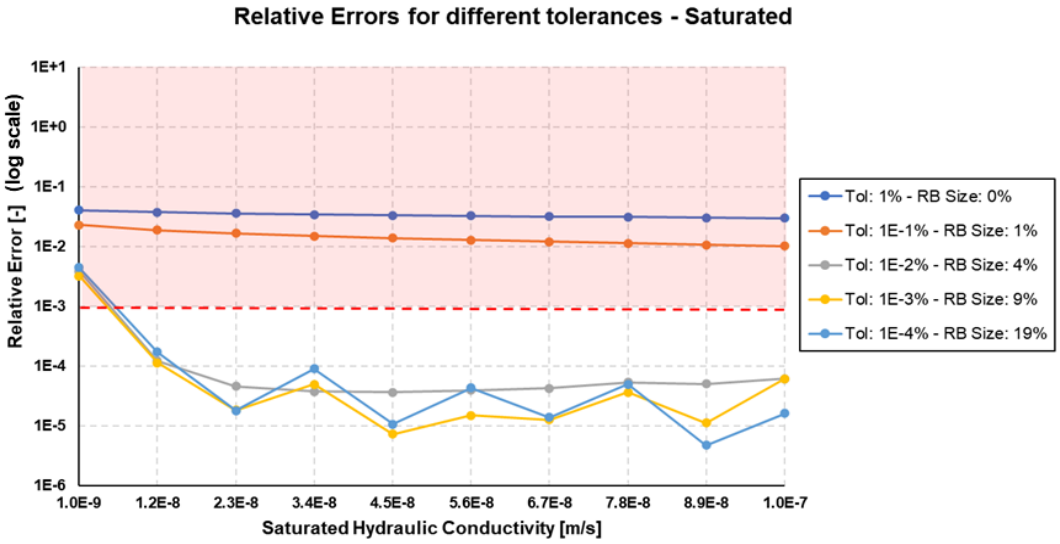

Figure 4-6: Behaviour of the relative errors for different percentage values of tolerance, and different saturated hydraulic conductivities in the range from 10^{-9} to 10^{-7}, in saturated conditions.

In the graph, the red discontinuous line marks the maximum accepted value of the relative error. The area found above the value is shaded in red in order to clearly show that the errors found in that portion of the graph do not satisfy the set condition and are, therefore, too high.

The legend of *Figure 4-6* presents the percentage value of the tolerance associated with each function, alongside the percentage size of the reduced basis, as defined in *Figure 3-11*.

A first remark that can be made in view of *Figure 4-6* is that the relative errors obtained by the reduced-order models defined by tolerance values of 10^{-1} and 1% are significantly higher than the maximum accepted value, for all of the different values of k_{sat}.

Another observation that can be made is that, for all of the studied tolerances, the relative errors have a tendency to decrease with the growth of the values of k_{sat}.

The solutions of the reduced-order model obtained for lower tolerances than 10^{-2} % yield relative errors that are below the maximum, for every value of k_{sat}, except for 10^{-9} ms^{-1}.

It was already concluded in the previous chapter that the nonlinear solver has more difficulty in converging for lower values of k_{sat} (*Conclusion 3-5*). The fact that the error associated with the lowest value of k_{sat} is higher than the error obtained for the rest of values, and that the functions decrease for higher values of k_{sat}, is consistent with this deduction.

In an attempt to lower the value of the tolerance to obtain an acceptable error for the k_{sat} value of 10^{-9} ms^{-1}, the following table offers the relative errors found for the tolerance values of 10^{-5} and 10^{-10} %:

Table 4-5: Relative errors of the reduced-order model's solutions, obtained for a saturated hydraulic conductivity value of 10^{-9} ms^{-1}, for tolerances of 10^{-5} and 10^{-10} %.

Saturated Hydraulic Conductivity [ms^{-1}]	Tolerance [%]	Relative Error [-]
10^{-9}	10^{-5}	0.004
	10^{-10}	0.002

The error associated with a k_{sat} value of 10^{-9} ms^{-1} continues being higher than the limit for the tolerance of 10^{-10}%. As it makes no sense to truncate the model using such low values of tolerance, given that the computational cost would not differ in a significant enough way from that of the full-order model, other parameters have to be used to observe how the error behaves.

In order to see how the relative error behaves for higher and lower ranges of k_{sat}, the next subsection (*Saturated Hydraulic Conductivity*) will build different snapshot matrices that permit to continue this line of research.

A third thing that stands out from *Figure 4-6* is that the higher tolerance values present approximately constant relative errors for different values of k_{sat}, while the lower tolerances show a certain variability.

It can be noticed that the tolerance value of 10^{-2}% marks the transition between the constancy and the variability, as its function does not present significant changes between the errors measured for higher k_{sat} values than approximately $2.3 \cdot 10^{-8}$ ms^{-1}, but does clearly have a higher error value for the value of 10^{-9} ms^{-1} than for the rest.

This observation can also be linked to a provisional conclusion drawn in the previous chapter, that pointed out that the non-variability of the number of systems solved by the reduced-order model for high tolerance values is due to a lack of precision. The mentioned lack of precision translates to a lack of accuracy, proven by *Figure 4-6*, as it shows that the reduced basis of the model needs to incorporate more information in order for its accuracy to be acceptable.

As the tolerance of 10^{-2}% also shows a lack of variability between the errors found for different values of saturated hydraulic conductivity, and is still a relatively high, it is possible to perceive that, under the problem conditions specified in the beginning of the subsection, 10^{-3}% is the highest tolerance value that obtains an adequate pattern of error for different values of k_{sat} (with the exception mentioned in paragraphs above).

Hence, the idea is to use the tolerance of 10^{-3}% as a fixed value in the subsequent subsections, when dealing with saturated conditions. The high tolerance of 10^{-1}% will also be implemented, in order to continue perceiving the transition between a tolerance that is too high and one that is reasonably low.

Having discussed the variability of the relative errors, it is important to reflect on the optimal result sought by the overall parametric analysis.

The desirable outcome of the analysis would be to find that the solutions of the reduced-order model present an error which is lower than the accepted value of 10^{-3}, and is independent of the value of the parameter (i.e. that it does not oscillate for different values of k_{sat}).

Whilst the constancy of the error functions presented by high tolerance values have been proven to be linked to a lack of precision and accuracy, the constancy of a function whose error is found under the maximum accepted value can mean that the reduced basis has the capacity to describe the solutions of the model well enough for whichever value of k_{sat} found between the range of values (sampled or not).

Although the tolerance of 10^{-2}% presents an error function which is constant for different values of k_{sat}, and is found below the defined acceptable value, the fact that the lower tolerances of 10^{-3} and 10^{-4}% do not maintain this constancy, seems to indicate that the reason why the function drawn by the errors obtained for a tolerance of 10^{-2}% does not vary from value to value is due to it being the transition between the higher and the lower values, as mentioned in the above paragraphs.

Unsaturated Conditions

In the case of the unsaturated scenario, (which corresponds to the analysis of node 275), an equivalent graph to *Figure 4-6* is drawn, with the aim to see how the change in saturation affects the relative errors produced by the reduced-order model, for different values of k_{sat} (*Figure 4-7*).

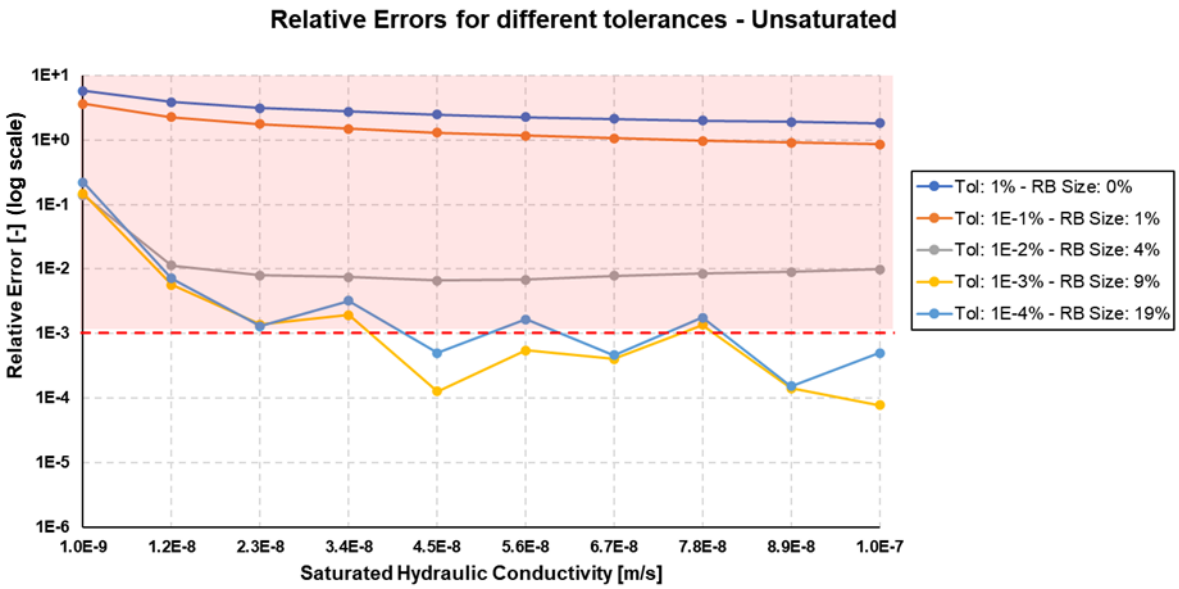

Figure 4-7: Behaviour of the relative errors for different percentage values of tolerance, and different saturated hydraulic conductivities in the range from 10^{-9} to 10^{-7}, in unsaturated conditions.

An immediate observation that can be made is that the tolerance values between 10^{-4} and 10^{-2}%, which, for saturated conditions, presented lower relative errors than 10^{-3} for all of the solutions of the reduced-order model, (except the one corresponding to a k_{sat} of 10^{-9} ms^{-1}), no longer maintain an acceptable level of accuracy.

The lowest value of tolerance, which should correspond to the highest accuracy present in the graph, is mostly found under the maximum error, for k_{sat} values between $4.5 \cdot 10^{-8}$ to 10^{-7} ms^{-1}. Even so, half of the values that form the function are higher than 10^{-3}, causing the result to be considered unacceptable.

By comparing this graph with the one found for saturated conditions (*Figure 4-6*), it can be perceived that the functions drawn in both representations hold many similarities. This is seen clearest for the tolerance of 10^{-4}%.

The functions in both conditions have similar shapes, but are found vertically shifted between 1 and 3 orders of magnitude.

The reason for this phenomenon could be related to the difference between the solutions of the reduced-order model obtained in the studied spatial nodes and the definition of the relative error.

As mentioned in the beginning of the chapter, the pore water pressure in point 63 (total saturation) oscillates between values of $2 \cdot 10^4$ and $7 \cdot 10^4$ Pa, for the lowest k_{sat} value of 10^{-9} ms^{-1}.

In the case of the unsaturated spatial node 275, the values of pore water pressure oscillate between $-1.5 \cdot 10^4$ and $3.5 \cdot 10^4$, which makes sense, (based on *Equation 4-2*), as the point is found approximately 3 metres above point 63. Negative values of pore water pressure result from node 275 being found, at times, above the water table [38].

The absolute error in both spatial points should be analogous, as it is a measure of the difference between the solutions found by the reduced-order and the full-order model.

Meanwhile, the numerical integration of the pore water pressure obtained by the full-order model in unsaturated conditions yields a lower result than for saturated conditions, due to the oscillating values being closer to zero.

As the relative error is calculated by dividing the absolute error by the integral of the full-order solutions (see *Figure 4-4*), the error associated with unsaturated conditions will be higher, as the denominator is lower.

The issue, then, seems to be that the spatial point selected to analyse the unsaturated conditions is too close to the water table for its relative error to be correctly quantified. The change of the model's solutions (values of pore water pressure) between positive and negative values also propitiates the denominator to be low in comparison to that of the saturated conditions.

In order to find a way around the problem, it needs to be also thought of in a physical sense.

When working in the field, monitoring devices such as piezometers are rarely installed at a shallow depth in relation to the water table, as the state change between saturated and unsaturated could damage the instrumentation [48].

As the accepted relative error has been set in accordance with the margin of error of piezometers, and these are usually installed at depth under the water table, the decimal value of 0.001 only makes sense when it is applied to positive hydrostatic pressure.

In conclusion, the acceptable value is only valid when studying saturated conditions, and therefore, a certain amount of leeway needs to be given to the unsaturated analysis.

The important information that can be obtained from comparing *Figure 4-6* and *Figure 4-7* is that the errors in both pressure states follow similar tendencies for different values of k_{sat}, although the unsaturated case presents a slightly higher oscillation.

.

Another important observation that can be made from this comparison is that the graph drawn for the unsaturated conditions is consistent with the saturated one in the sense that the variability of the errors presented by tolerance values equal or lower to 10^{-2}%, is relatively low (within the same order of magnitude) for k_{sat} higher or equal to $2.3 \cdot 10^{-8}$ ms^{-1}.

This opens the door to a very important line of research in the analysis, introduced at the end of the analysis of saturated conditions in the current subsection.

Up until now, the simulations carried out have been aimed at studying the viability of the proposed methodology to compute the accuracy with which the reduced-order model predicts the pore water pressure at a single point of the dam, and to investigate how this accuracy compares to the defined maximum accepted error.

Having found that the use of the proposed methodology to quantify the errors offers consistent results to various conclusions drawn in the *Computational Cost Analysis* when applied to pore water pressure that are not too close to zero, it is now necessary to focus the attention of the analysis on investigating how capable the model is of predicting the pore water pressure of k_{sat} values that have not been sampled by the snapshot matrix.

This aptitude is related to the lack of oscillation presented by the function that connects the errors associated with the sampled solutions (for the 10 k_{sat} values equally spaced in the range 10^{-9} to 10^{-7} ms^{-1}).

The reasoning behind this statement is the following:

- If the function drawn by connecting the errors associated with the solutions sampled by the matrix of snapshots is flat (in the sense that it does not oscillate significantly for different values of k_{sat}, the reduced basis has the capacity to describe the solutions of the intermediate (non-sampled) values of k_{sat}, by linear interpolation.
- If, on the contrary, the function is not smooth, but presents relative errors that oscillate significantly from one sampled solution to the next, the intermediate solutions would be expected to present higher values of error than its surrounding sampled solutions, as the reduced basis would have more difficulty in predicting the solutions that are not accounted for in the snapshot matrix.

Therefore, in order to determine how well the reduced-order model (and its corresponding reduced basis) describes the non-sampled solutions, the following focus questions has to be answered:

Focus Question 1	**How do the relative errors caused by the reduced-order model behave when it solves the uncoupled problem for non-sampled values of k_{sat}?**

To obtain a response to this question, further simulations have to be run.

As explained before, the reduced-order model in study is created by a reduced basis truncated at a tolerance of 10^{-3}% (which is formed by 9% of the vector directions of the original-sized model (see Figure 3-11)).

The idea is to run the *Hydro_Unsaturated.py* and the *RB_Solver.py* codes for values of k_{sat} which have not been previously used to assemble the snapshot matrix in *Snapshots.py*.

For the selection of the non-sampled values of k_{sat}, an interesting idea is to consider a value that is lower than $2.3 \cdot 10^{-8}$ ms^{-1} and another which is higher.

The chosen values are the following:

Table 4-6: Selected non-sampled values of saturated hydraulic conductivity.

Intermediate Values of Saturated Hydraulic Conductivity [ms^{-1}]	
10^{-8}	$5 \cdot 10^{-8}$

By modifying *Figure 4-7* so it only includes the errors associated with a tolerance of 10^{-3}% and the intermediate values presented in *Table 4-6*, the following graph can be plotted:

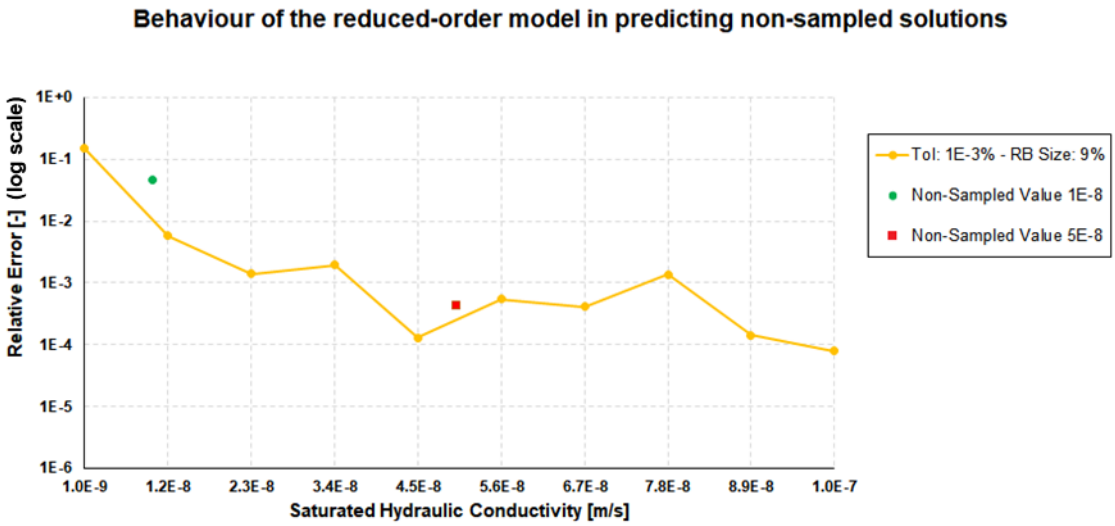

Figure 4-8: Capacity of the reduced-order model, created by a reduced basis of 10^{-3}%, of predicting non-sampled solutions, for a k_{sat} range between 10^{-9} and 10^{-7} ms^{-1}, in unsaturated conditions.

The graph shows that, as expected, the error presented by the non-sampled solution found for a k_{sat} value under $2.3 \cdot 10^{-8}$ ms^{-1} is found far above the line that connects the sampled k_{sat} solutions that surround it.

In the case of the intermediate value that is higher than $2.3 \cdot 10^{-8}$, the error is closer to the line that connects its surrounding sampled solutions, but still presents a higher error than the function that connects the sampled values of $4.5 \cdot 10^{-8}$ and $5.6 \cdot 10^{-8}$ ms^{-1}.

This indicates that the oscillation presented by the k_{sat} values higher than $2.3 \cdot 10^{-8}$ continues to be high.

In the previous chapter, it was proven that the nonlinear solver works as well for the reduced-order model as it does for the full-order model for k_{sat} values higher than 10^{-7} ms^{-1}, whilst lower values present a higher oscillation, and generally need more iterations to reach convergence (*Conclusion 3-3* and *Conclusion 3-5*).

This mentioned oscillation could be connected to the variability of the errors seen in the above graph (*Figure 4-8*). In order to investigate this, it is necessary to change the range of saturated hydraulic conductivity values so that it considers higher values than 10^{-7} ms^{-1}.

As another aspect to be studied is how the relative errors of the reduced basis behave in relation to the accepted margin of error, when the range of saturated hydraulic conductivity values is defined so that it considers higher values than 10^{-9} ms^{-1}, it is clearly necessary to pass on to study the different parametric scenarios created by varying the ranges of k_{sat}, while fixing the values of the rest of the study parameters.

4.4.2. Saturated Hydraulic Conductivity

Among the results of the previous chapter (*Final Set of Results*), one of the conclusions that was drawn was that it is necessary to study the typical values that the saturated hydraulic conductivity is expected to have in a real tailings dam, in order to have the sufficient knowledge to build a snapshot matrix that is limited to that interval of values, and obtain better computational cost results.

By looking at *Equation 2-2* it is possible to perceive that the value of k_{sat} will be equal or higher than the value of hydraulic conductivity (k), as k_{sat} is multiplied by the effective saturation (S_e) whose maximum value is 1 (*Equation 2-3*).

Therefore, in order for k_{sat} to be equal to k, the effective saturation has to be 1, which is the case of the saturated conditions:

$$k(p) = k_{sat}\sqrt{Se_{max}(p)}\,[1-(1-Se_{max}(p)^{\frac{n}{n-1}})^{m}]^2$$

$$k(p) = k_{sat}\cdot\sqrt{1}\,[1-(1-1^{\frac{1.23}{1.23-1}})^{0.18}]^2 = k_{sat}$$

For the unsaturated conditions, S_e takes a lower, (positive value) than 1, so that:

$$k(p) \leq k_{sat}$$

The literature on tailings dams indicates that the saturated hydraulic conductivity (k_{sat}) is highly prone to differ from one case to another, and can also vary in orders of magnitude in the same embankment [49].

This high variability is due to the fact that the magnitude of the parameter depends on many factors, such as the grain size of the material found in the dam, (which due to its heterogeneity, usually varies significantly throughout the structure), the type of tailings found in the embankment of the dam, the water content and the degree of consolidation.

The k_{sat} in a tailings dam usually ranges from 10^{-9} to 10^{-5} ms^{-1} [50][51]. Even lower values can be reached in extreme circumstances [52].

Having concluded in the previous chapter that the nonlinear solver works as well for the reduced-order model as it does for the full-order model for higher k_{sat} values than 10^{-7} ms^{-1}, the range of solutions for which the snapshot matrix should be built is 10^{-9} to 10^{-7}, which coincidentally, is the range that has been used until now.

In the previous pages, a need was expressed to investigate how the relative error of the reduced-order model behaves in response to varying the range of saturated hydraulic conductivities. As the sampled k_{sat} value that presents problematic errors in the analysis, for reduced bases defined by tolerance values equal or lower to 10^{-2}%, is 10^{-9} ms^{-1}, a new snapshot matrix is built that ranges from 10^{-10} to 10^{-8} ms^{-1}.

In the previous subsection it was shown that the transition between the error functions being found under and above the accepted value of 10^{-3} is found between the k_{sat} values of 10^{-9} and $1.2 \cdot 10^{-8}$ ms^{-1} (see *Figure 4-6 and Figure 4-7*) .

The idea of assembling a matrix with a smaller range between 10^{-10} and 10^{-8} ms^{-1} is to find for which k_{sat} value, the error surpasses the accepted value, or if the change of the snapshot matrix causes there to be a change in the behaviour of the relative errors.

It is also a good idea to track whether the relationship between the errors obtained for saturated conditions and unsaturated conditions is maintained, and if higher tolerances than 10^{-2}% continue yielding approximately constant relative errors that do not satisfy the accepted value, for different ranges of k_{sat}.

The following graph (*Figure 4-9*) displays the behaviour of the relative errors for a k_{sat} range from 10^{-10} to 10^{-8} (also containing 10 sampled solutions).

In order to perceive more fully the different aspects discussed in the previous paragraphs, the following differentiations are made in the plot:

- **Saturated conditions:** Continuous lines.
- **Unsaturated conditions:** Lines made of hyphens and dots.
- **Tolerance of 10^{-1}%:** Blue lines.
- **Tolerance of 10^{-3}%:** Orange lines.

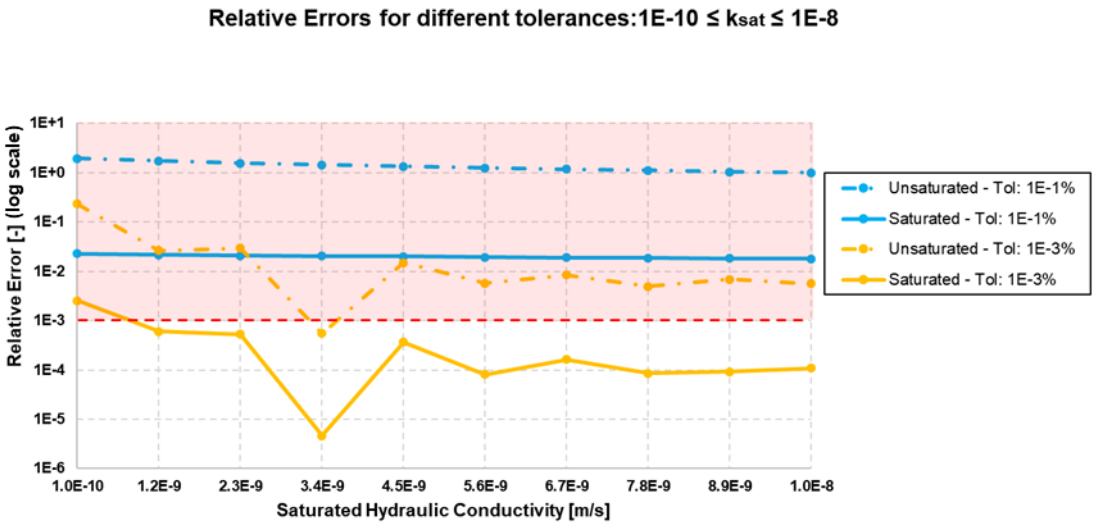

Figure 4-9: Behaviour of the relative errors for different tolerance values of 10^{-1} and 10^{-3}%, and for different saturated hydraulic conductivities in the range from 10^{-10} to 10^{-8} ms^{-1}, in both saturated and unsaturated conditions.

By comparing the above graph to the figures displayed for the saturated and unsaturated conditions in *Tolerance*, the following observations can be made:

- The relative errors presented by the tolerance values of 10^{-1}% have not changed significantly. Both in the case of the saturated and the unsaturated conditions, the values and the shapes of the lines are maintained for both k_{sat} ranges.

- For the tolerances of 10^{-3}%, the shape of both lines does change significantly. Meanwhile, it can be seen that the orders of magnitude presented by the errors in both snapshot matrices are consistent.

- In the case of the k_{sat} range from 10^{-10} to 10^{-8} ms^{-1}, the function drawn for the unsaturated conditions oscillates less than the one found for total saturation. This finding contradicts what was seen, in this respect, for a range from 10^{-9} to 10^{-8}.

The first observation is consistent with an observation made from *Table 3-11* which stated that high values of tolerance do not present different results for differently assembled snapshot matrices.

As the study of high tolerance values has not provided any new information to the analysis other than confirming what has already been concluded in *Conclusion 3-4*, they will not be included in the subsequent simulations.

In the case of the second observation, it is clear that the relative error differs due to the solutions being found for different values of k_{sat}.

In terms of values, *Figure 4-9* proves once again that the reduced-order model has a harder time resolving the governing equation for lower values of k_{sat}.

In order to perceive the similarities between the orders of magnitude presented by the functions of both snapshot matrices, the following figure shows the functions drawn by the relative errors obtained by the reduced-order model for a tolerance value of 10^{-3}% in saturated conditions, for both snapshot matrices.

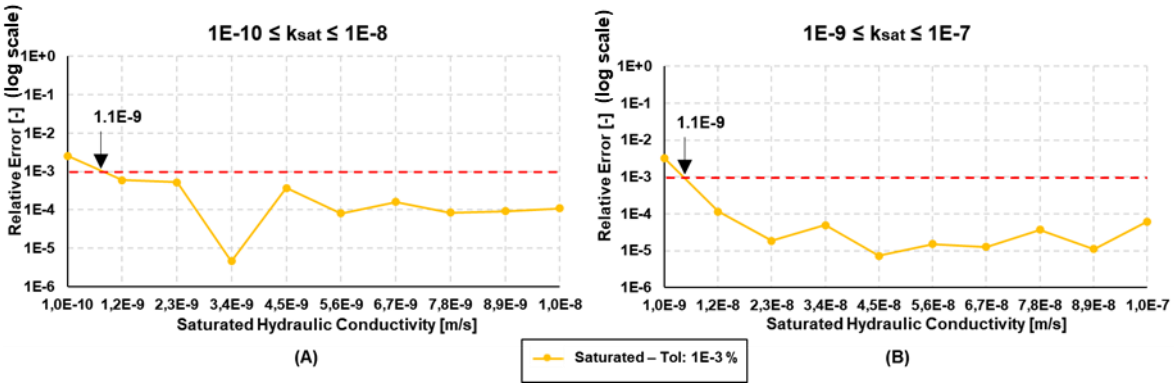

Figure 4-10: (A) Graph that shows the behaviour of the relative errors for a snapshot matrix from 10^{-10} to 10^{-8} ms^{-1}. (B) Analogous graph to the left, which displays the behaviour of the relative errors for a snapshot matrix from 10^{-9} to 10^{-7} ms^{-1}.

By comparing both functions, it is easily seen that the error presented by the reduced-order model exceeds the accepted value for the approximate k_{sat} value of $1.1 \cdot 10^{-9}$ ms^{-1}.

The magnitude of the measurements of relative error in both cases are similar. This can be observed by seeing that the order of magnitude presented by the value of 10^{-8} ms^{-1} in graph (A) and the value of $1.2 \cdot 10^{-8}$ ms^{-1} in graph (B), is the same.

Another observation that can be made by looking at both graphs in *Figure 4-10* is that the gradient of the line that connects the first and the second measured relative errors is higher in the case of graph (B). The reason for this is that the snapshot matrix corresponding to graph (B) is constituted by a larger range of k_{sat} values than is the case of graph (A).

Moreover, the line that connects the first and the second sampled relative errors in graph (B) corresponds to almost the whole of the function presented in graph (A).

By changing the snapshot matrix to a range of k_{sat} from 10^{-8} to 10^{-6}, the following graph is constructed:

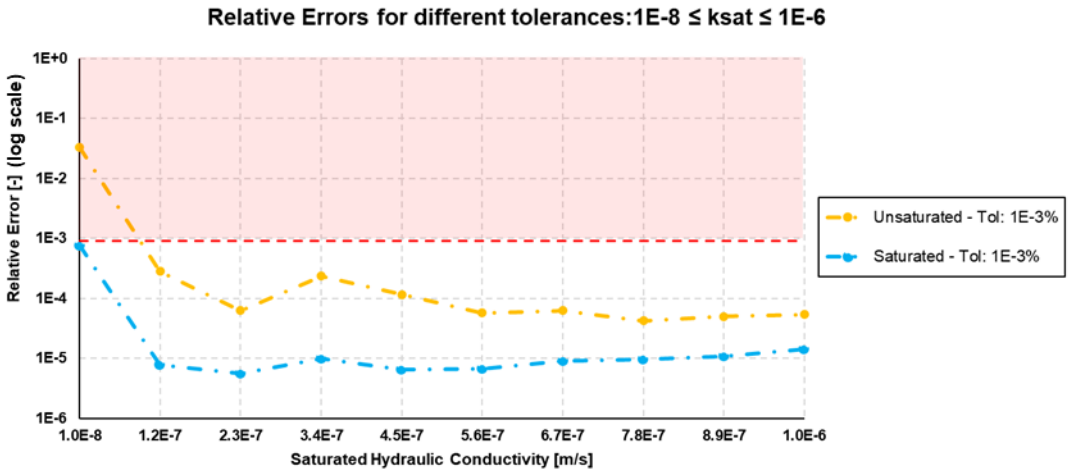

Figure 4-11: Behaviour of the relative errors for a tolerance value of 10^{-3}%, and for different saturated hydraulic conductivities in the range from 10^{-8} to 10^{-6}, in both saturated and unsaturated conditions.

As expected, the saturated function is entirely underneath the maximum error value, although the lowest value of k_{sat} (10^{-8} ms^{-1}) is very close to the limit.

Analogous to the previous graph (see *Figure 4-9*), the unsaturated function oscillates in a similar way to the saturated function, but is found shifted upwards, meaning that it crosses the margin of error for the lowest value of k_{sat}.

Even though the shape of both functions is similar, it can be perceivable that, once again, the one drawn for the saturated conditions oscillates less.

Another line of study brought up in the previous subsection (*Focus Question 1*) hypothesized that there could possibly exist a correlation between the variability presented by the number of linear systems that the reduced-order model has to resolve, for different values of k_{sat} below 10^{-7} ms^{-1}, and the oscillation seen for the relative errors, under the same k_{sat} value.

In order to study this possibility, a new parametric scenario has been constructed by defining a snapshot matrix that ranges from 10^{-6} to 10^{-8} ms^{-1} (*Figure 4-11*).

For the mentioned hypothesis to be validated, the reduced-order model should be capable of predicting non-sampled solutions that correspond to values of k_{sat} higher than 10^{-7} ms^{-1} with more accuracy than in the case of values found under 10^{-7} ms^{-1}.

In other words, the error of the non-sampled solutions should fall nearer to the function that connects the sampled solutions.

For that to happen, another aspect that has to be satisfied is that lower k_{sat} values than 10^{-7} ms^{-1} present lower oscillation between the errors. In this respect, *Figure 4-11* clearly shows that the oscillation presented by the errors is generally lower than the oscillation seen in *Figure 4-8.*

The chosen intermediate values are presented in the following table:

Table 4-7: Selected non-sampled values of saturated hydraulic conductivity.

Intermediate Values of Saturated Hydraulic Conductivity [ms^{-1}]	
10^{-7}	$5 \cdot 10^{-7}$

Having observed in the previous subsection that the saturated conditions can present lower oscillations than the unsaturated conditions under certain conditions, and vice versa, it is interesting to compare how the non-sampled values (*Table 4-7*) fare in both situations.

Unsaturated Conditions

For the unsaturated conditions, the following graph is plotted, which displays the function of the errors caused by the reduced-order model, and the intermediate solutions found in *Table 4-7*.

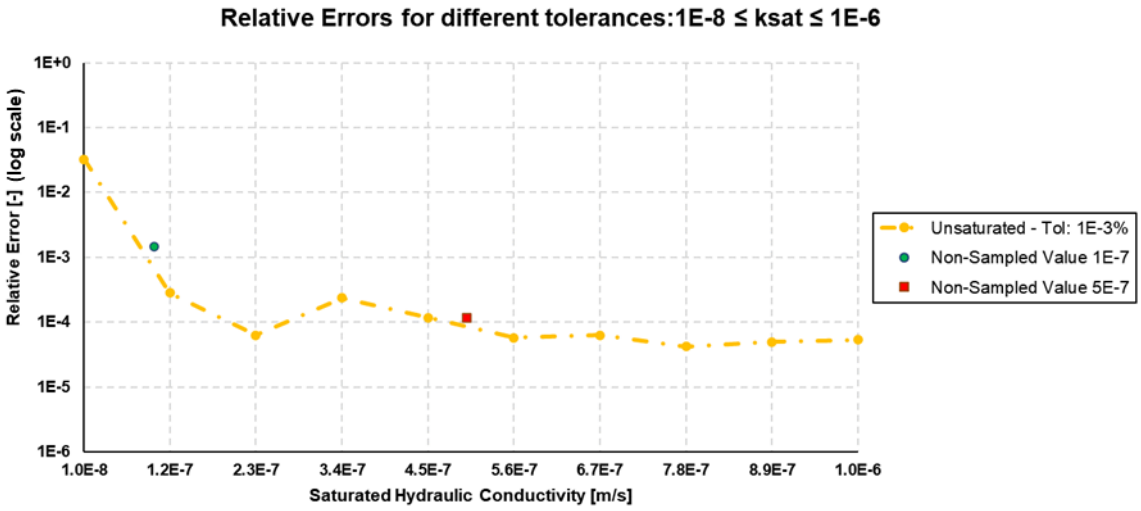

Figure 4-12: Capacity of the reduced-order model, created by a reduced basis of 10-3%, of predicting non-sampled solutions, for a k_{sat} range between 10^{-8} and 10^{-6} ms^{-1}, in unsaturated conditions.

Given that the error presented by the intermediate value of 10^{-7} ms^{-1} does not fall on the line that connects the errors of the sampled solutions, it can be seen that the k_{sat} value for which the error stops oscillating is higher than the predicted value of 10^{-7} ms^{-1}.

As the intermediate value of $5 \cdot 10^{-7}$ is closer to the function, but still presents a slightly higher error than the one that would be found by linear interpolation between the surrounding sampled solutions, it can be seen that the minimum k_{sat} value from which the reduced basis is capable of predicting the solutions of the non-sampled values by linear interpolation, is higher than $5 \cdot 10^{-7}$ ms^{-1}.

By looking at the graph (*Figure 4-12*), it can be seen that the function that connects the errors of the sampled solutions becomes approximately constant from the value of $5.6 \cdot 10^{-7}$ ms^{-1}.

It is important to perceive and understand that the sampled solutions in the range of k_{sat} values from 10^{-8} to 10^{-6} ms^{-1} are each separated by a distance of $1.1 \cdot 10^{-7}$. This means that all of the values (except for the first) in the range from 10^{-9} and 10^{-7} ms^{-1}, whose error oscillation can be seen in *Figure 4-8,* are found between the first and second value of the range.

By comparing *Figure 4-12* with *Figure 4-8,* it can be seen that the non-sampled solutions are found increasingly further away from the function, as the k_{sat} value decreases.

Saturated Conditions

An analogous graph is plotted for saturated conditions:

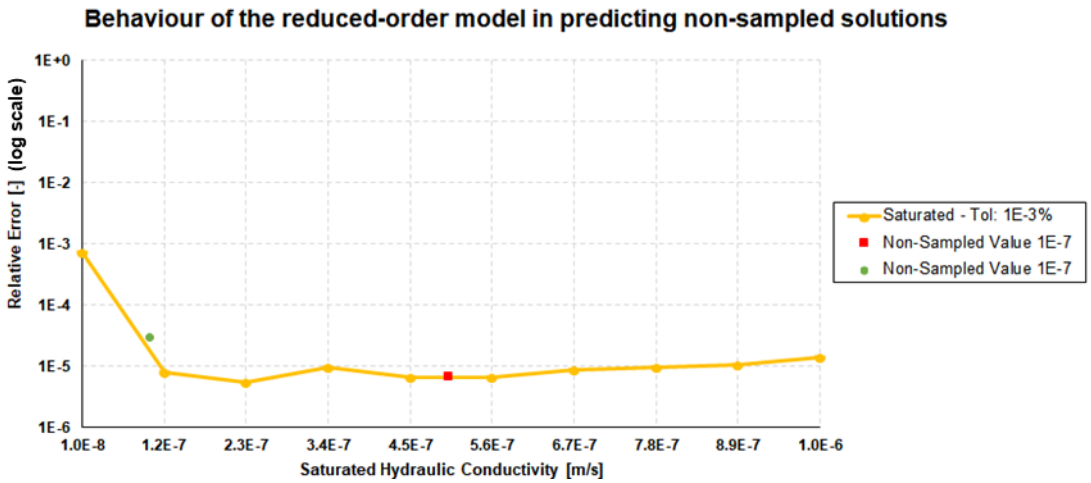

Figure 4-13: Capacity of the reduced-order model, created by a reduced basis of 10^{-3}%, of predicting non-sampled solutions, for a k_{sat} range between 10^{-8} and 10^{-6} ms^{-1}, in saturated conditions.

The fact that the intermediate value of 10^{-7} slightly differs from the function, and being able to observe that the steadiness of the function is not reached until k_{sat} is equal or higher than approximately $1.2 \cdot 10^{-7}$, it can be concluded that the value of $1.2 \cdot 10^{-7}$ ms^{-1} is the transition, for saturated conditions, between the oscillation and the constancy of the function.

On the other hand, the fact that the non-sampled value of $5 \cdot 10^{-7}$ is directly on the function, and that the function in itself clearly oscillates significantly less than in *Figure 4-12*, corroborates that the reduced-order model has more difficulty in predicting the non-sampled solutions that are found between solutions whose error changes drastically, than those who maintain error values in the same order of magnitude. In other words, it has more difficulty in finding the solutions of intermediate values found on steeper slopes.

It is important to remember that the graph is semi-logarithmic, so the steepness criteria can only be used when comparing graphs with the same axes.

It is also significant to be aware of the fact that the snapshot matrices whose value ranges are bigger will generally present steeper slopes than the smaller ranges, so using the criteria of the 'smoothness' of a function to compare differently-sized snapshot matrices can be misleading.

By looking back at the graph plotted for the unsaturated conditions (*Figure 4-12*), it can be perceived that this constancy is reached for higher values of k_{sat}, at approximately $5.6 \cdot 10^{-8}$ ms^{-1}. Meanwhile, for the saturated conditions, the function becomes approximately constant for a k_{sat} value of $1.2 \cdot 10^{-7}$ ms^{-1}or higher.

Comparing these values to the transition concluded in *Computational Cost Analysis* that showed that the nonlinear solver has more difficulty in converging in the entire dam for a reduced-order model associated to equal or lower k_{sat} values than 10^{-7} ms^{-1}, it can be seen that the saturated conditions are more representative of the whole dam than the unsaturated.

By unifying these boundaries, it can be concluded that the oscillations observed for lower k_{sat} values than $1.2 \cdot 10^{-7}$ ms^{-1} are caused by the difficulty in converging encountered by the nonlinear solver for those values.

A line of research that should be considered in the future is to find the reason why the nonlinear solver has more difficulty in finding the solutions for lower values of k_{sat} than $1.2 \cdot 10^{-7}$ ms^{-1}.

A last observation that can be made based on the graph in Figure 4-13 is that when the function that connects the errors becomes independent of the value of k_{sat} (when it stops oscillating drastically), the order of magnitude presented by the errors is equal to the tolerance value.

This remark is made harder to perceive by the fact that the tolerance is expressed as a percentage while the errors are calculated in decimal form. In order to unify the orders of magnitude, the tolerance can be expressed as a decimal, as follows:

$$Tolerance\,[-] = Tolerance\,[\%] \cdot \frac{1}{100} = 10^{-3}\,\% \cdot \frac{1}{100} = 10^{-5}$$

The observed correlation between the value of tolerance and the relative error produced by the reduced-order model makes sense, as can be seen by comparing their meaning:

Table 4-8: Comparison between the definitions of tolerance and relative error, as established in the study.

Tolerance	Relative Error
Value that measures the amount of information from the solutions of the full-order model that is discarded in the process of creating the reduced basis, (and subsequentially), the reduced-order model.	Value that measures the difference between the solutions of the full-order model and the reduced-order model.

Table 4-8 permits noticing that the difference between the concepts is that the tolerance measures the level of information that is used from the original model to calculate the solutions of the reduced-order model, and therefore indicates the level of precision that the reduced model is expected to achieve, whilst the relative error quantifies the exact difference between the solutions of the reduced model, when compared to the original.

In other words, the tolerance is a chosen value that is imposed on the model in order to generate a certain reduced basis. It is an input that attempts to control the maximum precision to which the reduced-order model can aspire. It affects the model at a global scale.

On the other hand, the relative error, as defined in the chapter, is the value that is output by the model. It defines the actual accuracy with which the reduced-order model predicts the solutions of the spatial node in study.

Consequently, there is no guarantee that the relative errors match the defined value of tolerance, but the order of magnitude of the tolerance is indicative on how low the relative error can be expected to go.

As seen, the tolerance and the relative error differ significantly for low values of k_{sat}, (for which the nonlinear solver needs to solve a higher number of linear systems in order for the reduced-order model to converge).

It can be concluded, therefore, that for high values of k_{sat}, there reaches a point where the solutions stop oscillating drastically from value to value, and become independent of k_{sat}, which can be identified by the values of the relative errors being very similar to that of the tolerance.

When this phenomenon occurs, the reduced basis of the model is capable of finding the solution of the uncoupled problem for any intermediate values of k_{sat}, as accurately as it finds its sampled solutions.

4.4.3. Time Dimension

The final parameter used to study the relative errors of the reduced-order model for different problem conditions is the time dimension, which refers to the total measured time.

Until now, the simulations carried out have used 8640·200 seconds (or 20 days) as the time dimension, due to it being the value used to build the model [17].

The idea in this subsection is to vary the total time of measurement, and see how the relative error behaves.

It is important to clarify that the size of the time steps (8640 seconds) remains the same for the different parametric scenarios. The reason for this is that it is much easier to compare results if only one variable is changed.

The importance of this exercise is that the reduced-order model has to be capable of not only increasing the speed of the model, but also of maintaining an adequate accuracy for different problem descriptions. This is highly significant, given that the model is aimed to one day be used to monitor different dams across the globe, which will be unique in terms of their geometry, their geology, their capacity, and so on.

The value of the time dimension should ideally be established as the time it takes a drop of water to travel through the embankment of the dam, from the upstream to the downstream.

To get a rough estimate on the amount of time it would take to cross a totally permeable homogeneous dam in a straight line, the linear equation of velocity can be used, which assumes that the velocity of the water is equal to the hydraulic conductivity:

$$T = \frac{L}{k} \qquad (Equation\ 4-3)$$

where,

T – Time dimension [s]

L – Length of the dam's embankment [m]

k – Hydraulic conductivity [ms^{-1}]

As seen in *Figure 2-1*, the length of the dam in the designed model is of approximately 50 meters.

In *Saturated Hydraulic Conductivity*, it was stated that a typical range of values for the hydraulic conductivity (*k*) of a tailings dam is equal or lower to the value of k_{sat}. Typical values of *k* in a tailings dam usually range between 10^{-5} and 10^{-9} ms^{-1} [53].

By taking the higher, less restrictive value of k_{sat}, the time dimension can be estimated to be equal to:

$$T = \frac{L}{k} = \frac{50\ m}{10^{-5}\ ms^{-1}} = 5 \cdot 10^{6}\ s\ \approx\ 60\ days$$

The value obtained is three times the current value of the model. Still, considering that the applied value of hydraulic conductivity is the minimum of the given range, the studied time dimension is considerably low.

In order to get a better approximation, other information would have to be taken into account, such as the hydraulic gradient, the effective porosity and the water level fluctuation.

As the purpose of the subsection is to analyse how the measured error behaves in different circumstances, the exact time dimension is not relevant.

Also, by changing the time dimension in the model, the number of time steps also changes, and the simulations take a proportional time to run. So, it is good to note that the values implemented in the study are generally low in comparison to reality, but towards having the capacity of carrying out the simulations, the different parametric scenarios will have values of time dimension with the same order of magnitude as the original.

The values that will be given to the time dimension are the following:

Table 4-9: Values of the total time dimension used to build different parametric scenarios.

Time Dimension [days]		
10	40	80

The reasoning behind the selected values is that of being able to see how the errors will change when the time dimension is halved, doubled and multiplied by four.

Saturated Conditions

The following graph (*Figure 4-14*) shows the behaviour of the relative error for the different time dimension values found in *Table 4-9*, for a tolerance of 10^{-3}%.

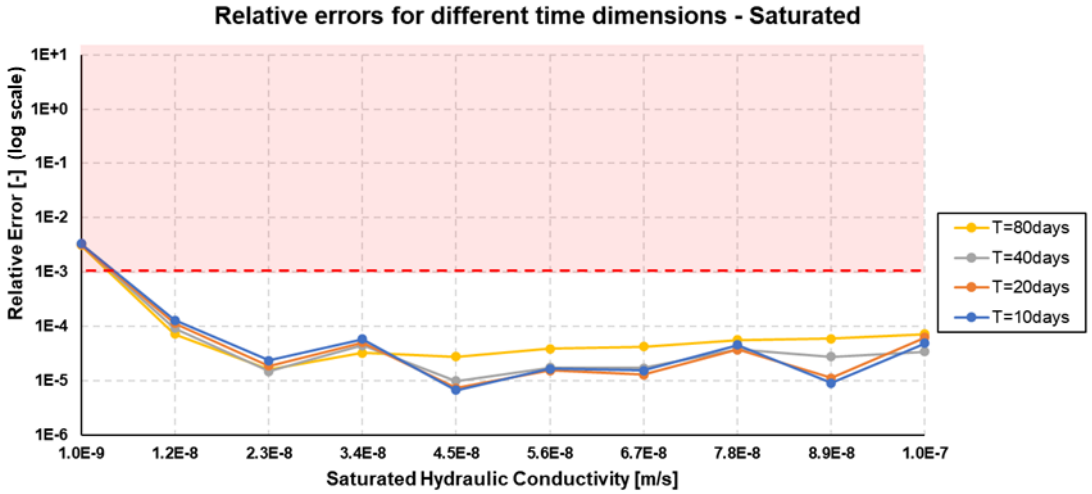

Figure 4-14: Behaviour of the relative error of the reduced-order model defined by a tolerance of 10^{-3}%, for different time dimensions, in saturated conditions.

It can be seen that for low values of k_{sat}, the relative error follows the same path for all of the studied time dimensions, presenting the oscillation seen in *Figure 4-6* for a tolerance of 10^{-3}%.

As the values of k_{sat} grow, the error associated with the highest time dimension of 80 days differentiates itself from the rest. The function it draws becomes approximately constant for equal and higher k_{sat} values than $2.3 \cdot 10^{-8}$ ms^{-1}, with a higher error value than the rest of the time dimensions. When it reaches $7.8 \cdot 10^{-8}$ ms^{-1}, the second highest time dimension (40 days) also stops oscillating and becomes parallel to the first.

As seen in previous subsections, the fact that the relative error is approximately constant for different values of k_{sat} can be an indication of one of the following phenomena.

- A lack in the accuracy of the solutions.
- A gain in the steadiness of the solutions.

In order to perceive which of the two possibilities corresponds to what is happening, it should be remembered that the constant error values that translate into a lack of accuracy are also found above the maximum accepted value of 10^{-3} (expressed as a decimal).

In the previous subsection it was found that a way to check whether the constancy of a function of relative errors is due to a gain in steadiness, is to see whether the order of magnitude of the errors correspond to the value of the defined tolerance.

The applied tolerance is 10^{-3}% (10^{-5} in decimal form), and the errors presented by the time dimensions of 80 and 40 days for the high values of k_{sat} in the range from 10^{-9} to 10^{-7} are found between the values of 10^{-5} and 10^{-4} of the vertical axis of the graph.

The constancy presented by the error functions of 80 days and 40 days seems to be due to a gain in steadiness, as they are found under the maximum accepted value of 10^{-3}.

Having said this, it is necessary to reflect on why these functions become independent from the k_{sat} values for higher values of error than that of the constant function drawn by a time dimension of 20 days.

It is also worth investigating why the error functions of 80 and 40 days become constant for lower values of k_{sat} (lower transition k_{sat} values) than the function drawn for the original time dimension of 20 days.

These differences are quantified in the following table:

Table 4-10: Differences in the behaviour of error functions drawn by the reduced-order model, for different time dimensions, in saturated conditions.

Time dimension [d]	Transition k_{sat} value [ms^{-1}]	Relative Error [-]
20	$1.2 \cdot 10^{-7}$	10^{-5}
40	$7.8 \cdot 10^{-8}$	$3 \cdot 10^{-5}$
80	$2.3 \cdot 10^{-8}$	$6 \cdot 10^{-5}$

It is important to remember that the axis that measures the relative errors in the studied graphs (e.g. *Figure 4-14*) is logarithmic, so it is hard to quantify the relative errors from pure observation. The results of the simulations carried out in the chapter can be found in *Appendix B.2*.

Table 4-10 shows that the errors presented by higher time dimensions are not much higher than those associated to lower time dimensions.

As the only difference between the mentioned functions is related to the total time of measurement, it is logical to reason that the functions of a higher time dimension become constant for lower values of k_{sat} due to the fact that the time loop (*Figure 2-2*) takes into account a lot more information, and is therefore enriched.

The reason for the function becoming constant for higher error values can be related to the errors introduced by the method used to estimate the variation of the pore water pressure over time (the Crank Nicolson method, as mentioned in *Uncoupled problem*).

So, by increasing the total time, the reduced basis of the model is capable of finding the non-sampled solutions to the uncoupled problem for lower values of k_{sat}, but does so more inaccurately, due to the accumulative error associated to the time integration.

In order to see if this deduction holds true for other points of the dam, an analogous graph is plotted for the unsaturated conditions.

Unsaturated Conditions

The following graph (*Figure 4-15*) displays the behaviour of the relative error for the different time dimensions values defined in *Table 4-9*, for a tolerance of 10^{-3}%, in unsaturated conditions.

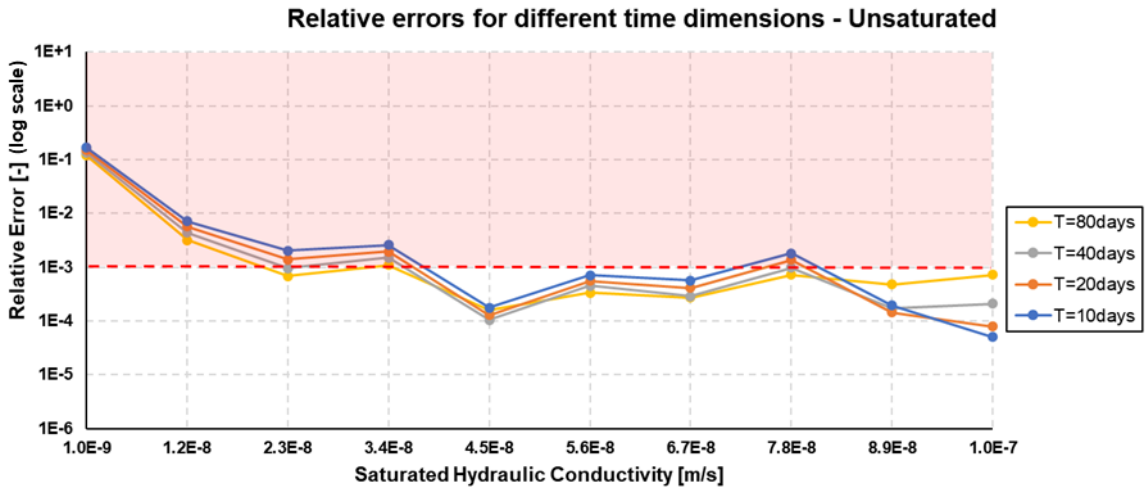

Figure 4-15: Behaviour of the relative error of the reduced-order model defined by a tolerance of 10^{-3}%, for different time dimensions, in unsaturated conditions.

By comparing *Figure 4-14* and *Figure 4-15* it can be seen that the functions drawn for unsaturated conditions present a slightly higher oscillation than those found for saturated conditions, making it possible to conclude that in the current study, the reduced basis of the model is generally more capable of predicting the intermediate values of pore water pressure for saturated conditions than for unsaturated conditions. As the representativeness of the unsaturated conditions given by node 275 has been questioned, in the future, it would be prudent to investigate these phenomena further, for different unsaturated spatial nodes.

On the other hand, the errors obtained by the reduced-order model for higher time dimensions become constant at higher values of k_{sat}, as quantified in the following table:

Table 4-11: Differences in the behaviour of error functions drawn by the reduced-order model, for different time dimensions, in unsaturated conditions.

Time dimension [d]	Transition k_{sat} value [ms^{-1}]	Relative Error [-]
20	$5.6 \cdot 10^{-8}$	$7 \cdot 10^{-5}$
40	$8.9 \cdot 10^{-8}$	$2 \cdot 10^{-4}$
80	$7.8 \cdot 10^{-8}$	$6 \cdot 10^{-4}$

An overall conclusion that can be drawn from the subsection is that the solutions of the model behave very similarly for different time dimensions, for low values of k_{sat}.

As the values of k_{sat} grow, the solutions obtained for problems defined by higher time dimensions become independent of the value of k_{sat}, and present higher relative errors.

his transition occurs for lower values in saturated conditions than for unsaturated conditions.

4.5. Concluding Remarks

After having discussed the parametric analysis carried out, the current subsection serves the function of collecting its main results.

Although numerous observations and deductions are made throughout the chapter, only those which are conclusive will be mentioned.

Accordingly, it is important to note that as it has been found that the behaviour of the relative error associated to the spatial point which is fully saturated at all times is more representative at a global scale than the results obtained from studying the unsaturated conditions, the conclusions will mainly focus on the former.

Table 4-12 recapitulates the principal conclusions drawn from the analysis. Given that most deductions describe how the errors behave for different parametric scenarios, the table will include divisions that facilitate its comprehension.

Table 4-12: Summary of the main conclusions drawn from the parametric analysis.

<table>
<tr><th colspan="6">Conclusions</th></tr>
<tr><td rowspan="7">Saturated</td><td rowspan="5">T = 20 days</td><td>4-1</td><td colspan="3">The reduced basis of the model is generally more capable of predicting the intermediate values of pore water pressure for saturated conditions than for unsaturated conditions. The results obtained for the saturated conditions are more representative of the dam.</td></tr>
<tr><td rowspan="4">$k_{sat} > 1.1 \cdot 10^{-9}$ ms^{-1}</td><td>4-2</td><td colspan="2">The size of the reduced basis created for a tolerance value of 10^{-3} % (10^{-5} in decimal) is the minimum that yields an acceptable accuracy.</td></tr>
<tr><td rowspan="2">$k_{sat} \geq 1.2 \cdot 10^{-7}$ ms^{-1}</td><td>4-3</td><td>The functions drawn by the relative errors of the sampled solutions present a steady behaviour, meaning that the error becomes independent from the different values of k_{sat}.
It can be identified by the similarity between the constant value of the relative error and the value of the tolerance, both expressed as decimals.</td></tr>
<tr><td>4-4</td><td>The reduced basis of the model is capable of finding the solutions of the uncoupled problem for non-sampled values of k_{sat} as accurately as it finds the sampled solutions, due to the steadiness of the error functions.</td></tr>
<tr><td>$k_{sat} < 1.2 \cdot 10^{-7}$ ms^{-1}</td><td>4-5</td><td>The function that connects the relative errors of the sampled solutions oscillates for different values of k_{sat}, due to the difficulty that the nonlinear solver presents in making the governing equation converge.</td></tr>
<tr><td>$k_{sat} \leq 1.1 \cdot 10^{-9}$ ms^{-1}</td><td>4-6</td><td colspan="2">The relative error produced by the reduced-order model exceeds the acceptable value of 10^{-3}.</td></tr>
<tr><td rowspan="2">T > 20 days</td><td>4-7</td><td colspan="3">As the time dimension grows, the reduced basis of the model becomes more capable of finding the non-sampled solutions to the uncoupled problem for lower values of k_{sat} (the functions go steady for lower values of k_{sat}), but yields slightly higher relative errors.</td></tr>
<tr><td>4-8</td><td colspan="3">Before the error functions become steady, the solutions of the model behave very similarly for different time dimensions.</td></tr>
<tr><td>Unsaturated</td><td colspan="2">4-9</td><td colspan="3">The proposed methodology for quantifying the relative errors produced by the reduced-order model is unsuitable for points of the dam where the conditions of the pore water pressure alternate between saturated and unsaturated.
It is also not practical for points that present pore water pressure values that are relatively close to zero.</td></tr>
</table>

An important observation which can be perceived by considering the conclusions drawn in *Table 4-12*, is that the reduced-order model generally manages to maintain an acceptable accuracy for different problem descriptions.

Taking into account the results obtained from the *Computational Cost Analysis* and the *Parametric Analysis*, it is now possible to draw general conclusions from the practical work of the project:

- For tailings dams whose saturated hydraulic conductivity ranges between $1.1 \cdot 10^{-9}$ and 10^{-5} ms^{-1}, the employment of the SVD and the subsequent creation of a reduced basis, make it possible to significantly reduce the computational cost of the model, without compromising its accuracy.

Having found that the highest tolerance value for which this holds true is 10^{-3} %, it is easy to see that the reduced basis created for a tolerance of 10^{-3} % is optimal, as it offers the minimal computational cost that yields error values within the established margin.

Figure 4-16 shows the computational cost of the reduced-order model for the mentioned tolerance, and the error it presents for different values of k_{sat}.

It can be seen that the computational cost of the model constitutes only 0.1% of the cost of the full-order model. The errors are generally lower than 10^{-4} (expressed as decimals).

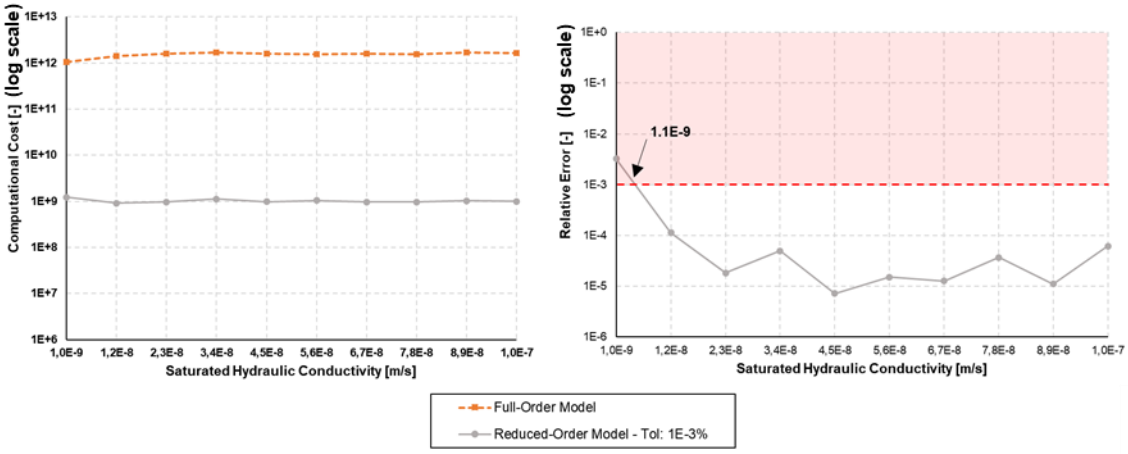

Figure 4-16: Optimal balance between the computational cost and the accuracy of the reduced-order model.

- For tailings dams with very low values of saturated hydraulic conductivity, the construction of a reduced basis can threaten the accuracy with which the model predicts the pore water pressure throughout the dam.

5. Project Conclusions

5.1. Conclusions

In the work presented in this document, different investigations were carried out in order to respond to the objectives defined in *Aim and Objectives*.

Firstly, a set of simulations were designed to study the speed with which the reduced-order model of the uncoupled problem can operate in comparison to the full-sized model.

To achieve this, a method was proposed, enabling the calculation of the computational cost of a model, which takes into account the behaviour of the solver used to tackle the nonlinear system of equations in each case, and the size of said system.

Additionally, the code in charge of creating the reduced basis of the problem was distinctly modified in order for it to use an input value of tolerance to carry out the truncation of the matrices.

From this investigation it was possible to conclude that the nonlinear solver used to resolve the algebraic system of equations (Picard iterative method), works as well for the reduced-order model as it does for the original-sized model, for higher values of the saturated hydraulic conductivity parameter.

For lower values, it was found that the solver requires a larger number of iterations to solve the system in the reduced-order model, meaning that it has a greater difficulty in making the governing equation converge.

Nevertheless, given that in defining the magnitude of the computational cost, the size of the reduced basis is more important than the number of iterations undertaken by the solver, the performance of the model was verified to be significantly improved by the reduction of its order, independently of the value of saturated hydraulic conductivity. It was also found that reduced bases formed by smaller value ranges yield better results in terms of speed.

On the other hand, it was demonstrated that the accuracy of the model is proportional to the size of the reduced basis, meaning that a balance has to be established between the computational cost of the model and the accuracy with which it calculates the pore water pressure state of the dam.

In order to study the accuracy associated with the reduced-order model, a parametric analysis was carried out, through another set of simulations, varying the value of the tolerance applied in the creation of the reduced basis, the saturated hydraulic conductivity, the total time dimension and the saturation.

A methodology was proposed to calculate the errors produced by the reduced-order model in comparison to the full-sized model, and a minimum acceptable accuracy was also defined.

This second investigation yielded numerous results that contributed to understanding how the accuracy of the reduced-order model behaves for different problem conditions.

With regard to the troubling parametric dependence identified to be affecting the performance of the reduced basis, it was found that for significantly low values of saturated hydraulic conductivity, the reduced basis is not capable of predicting the solutions of the full-order model to an acceptable accuracy.

By later joining the results obtained in both analyses, it was concluded that the high errors found for low saturated hydraulic conductivity values are due to the difficulty presented by the nonlinear solver in reaching convergence.

In the case of the other studied parameters, the investigation yielded a maximum tolerance value for which the generated reduced basis is such that the computational cost is the lowest while maintaining an acceptable accuracy. It also demonstrated that the proposed system of measurement is not valid for studying spatial points whose state of saturation changes over time. Lastly, the study showed that by considering longer time dimensions, the reduced basis becomes more capable of predicting the values of pore water pressure that are not sampled in the snapshot matrix, but does so slightly less accurately.

All of the results that have been found throughout the experimental work provide substantial knowledge of the behaviour of the reduced basis, the nonlinear solver and the computational model itself. The conclusions drawn from the analyses are therefore significantly relevant to the development of the European project.

Thanks to these analyses, the objectives set at the beginning of the work have been achieved.

5.2. Future Research

Even though many findings were made in the work presented in this document, there are still many aspects of the overall European project that are in need of further research.

Having approached the study of the uncoupled problem, the following step that should be taken, proposed by the author of this document in basis of the experience gained, is to investigate whether the conclusions drawn in the analyses carried out also hold true for the coupled scenario.

After that, the model should be modified in such a way as to take into account the deductions made in this work, in order to improve its performance.

Another line of research that should be followed is that of learning why the nonlinear solver has difficulty in making the system of partial differential equations converge for lower values of saturated hydraulic conductivity, and finding a means of solving the problem.

Lastly, the behaviour of the reduced model should be studied further for unsaturated conditions.

6. References

[1] Nasika C. et al. (2018-in progress) *Sensor data assimilation supporting decision making in the assessment of embankment dams*. PhD thesis in progress. CIMNE, Universitat Politècnica de Catalunya, Université Libre de Bruxelles & World Sensing. ProTechTion. Personal Communication. Available at: *https://www.cimne.com/vnews/10456/vac-2018-12-%E2%80%93--phd-pos-protechtion---sensor-data-assimilation-supporting-decision-making-in-the-assessment-of-embankment-dams-*

[2] Bauer, G. E. (1982). Stability of Tailing Structures - Pore Water Pressure Distribution and Monitoring. *Organ by Hung Min and Metall Soc, Budapest and Cent Inst for Min Dev of Hung*, 323–333. Pp. 323-324

[3] Lyu, Z. et al. (2019). A Comprehensive Review on Reasons for Tailings Dam Failures Based on Case History. *Advances in Civil Engineering*, *2019*

[4] Zardari, M. A. (2011) *Stability of Tailings Dams – Focus on Numerical Modelling, Licentiate Thesis, PhD*, P. 79

[5] Owen, J. R. et al. (2020) *Catastrophic tailings dam failures and disaster risk disclosure*, International Journal of Disaster Risk Reduction, 42, p. 101361. doi: *10.1016/j.ijdrr.2019.101361*

[6] Wills, B. A. and Finch, J. A. (2016) Chapter 16 - Tailings Disposal', in Wills, B. A. and Finch, J. A. B. T.-W. M. P. T. (Eighth E. (eds). Boston: Butterworth-Heinemann, pp. 439–448. doi: *10.1016/B978-0-08-097053-0.00016-9*

[7] The British Dam Society (1991) *The Embankment Dam*. *Proceedings of the sixth conference of the British Dam Society held in Nottingham on 12-15 September 1990.* United Kingdom. 2. Thomas Telford Ltd. ISBN: 9780727716477. Pp. 94-96

[8] Vanden-Berghe, J. F. et al. (2011) *Geotechnical risks related to tailings dam operations.* Proceedings of the 15th international conference on tailings and mine waste, 2010

[9] Zhang, L.M. et al. (2014) *Geotechnical Safety and Risk IV. PROCEEDINGS OF THE FOURTH INTERNATIONAL SYMPOSIUM ON GEOTECHNICAL SAFETY AND RISK (4TH ed.)* Taylor & Francis Group, LLC. p. 19

[10] Thompson, F. et al. (2020) *Severe impacts of the Brumadinho dam failure (Minas Gerais, Brazil) on the water quality of the Paraopeba River*, Science of The Total Environment, 705, p. 135914. doi: *10.1016/j.scitotenv.2019.135914*

[11] Raman, A. and Liu, F. (2019) *An investigation of the Brumadinho Dam Break with HEC RAS simulation*. New Jersey Science Academy. Available at: *http://arxiv.org/abs/1911.05219*

[12] Nasika, C. et al. (2019) *Reduced Model Order Techniques for Data Assimilation in Tailings Dams Monitoring*. MORTech 2019, 5th International workshop on Reduced Basis, POD and PGD Model Order Reduction Techniques. Personal Communication, Paris, 2019

[13] Kasolis, F. and Clemens, M. (2020) 'Maximum Entropy Snapshot Sampling for Reduced Basis Generation', pp. 1–14. Available at: *http://arxiv.org/abs/2005.01280*

[14] Abancó, C. et al. (2016) *Wireless low power real-time solutions for tailings dams – a case study*. APSSIM 2016, Australian Centre for Geomechanics

[15] Alnaes, M.S. et al. (2015) *The FEniCS Project Version 1.5.* Archive of Numerical Software, vol. 3. Available at: *https://journals.ub.uniheidelberg.de/index.php/ans/article/view/20553*

[16] Rutqvist, J. and Stephansson, O. (2003) *The role of hydromechanical coupling in fractured rock engineering*. *Hydrogeology Journal* 11, 7–40. *doi: 10.1007/s10040-002-0241-5*

[17] Nasika, C., Zlotnik, S. and Díez, P. (2019) *A simple model of the hydro-mechanical behaviour of earthfill dams to support decision making regarding tailings dams' safety.* International Conference on Adaptive Modelling and Simulation. Personal Communication. ADMOS 2019, Alicante, Spain

[18] Van Genuchten, M.T. (1980) *A closed-form equation for predicting the hydraulic conductivity of unsaturated soils*. Soil Sci. Soc. Am. J., 44:892-898.

[19] Kalashnikova, I. and Barone, M. F. (2011) *Stable and efficient galerkin reduced order models for non-linear fluid flow*, 6th AIAA Theoretical Fluid Mechanics Conference, pp. 3–4

[20] Pinnau, René. (2008). *Model Reduction via Proper Orthogonal Decomposition*. doi: *10.1007/978-3-540-78841-6_5*

[21] Schilders, W. H. a, Vorst, H. a Van Der and Rommes, J. (2008) Model Order Reduction: Theory, Research Aspects and Applications, Methods. doi: *10.1007/978-3-540-78841-6*, pp. 3-8

[22] Osterroth, S. (2018) *Mathematical Models for the Simulation of Combined Depth and Cake Filtration Processes*, Fraunhofer Verlag, Informationszentrum Raum und Bau, IRB, Stuttgart, p. 173. ISBN: 978-3-8396-1297-2

[23] Willcox, K. and Perairet, J. (2001) *Balanced model reduction via the proper orthogonal decomposition*, 15th AIAA Computational Fluid Dynamics Conference, 40(11), pp. 2323-2326. doi: *10.2514/3.15326*

[24] Henry, E. R. and Hofrichter, J. B. T.-M. in E. (1992*) [8] Singular value decomposition: Application to analysis of experimental data*, in Numerical Computer Methods. Academic Press, pp. 129–192. doi: *10.1016/0076-6879(92)10010-B*

[25] Kalman, D. (1996) *A Singularly Value Decomposition: The SVD of a matrix.* The College Mathematics Journal, vol. 27, Lecture NO. 1, January 1996 p. 22

[26] Tzeng, J. (2013) *Split-and-Combine Singular Value Decomposition for Large-Scale Matrix'*, Journal of Applied Mathematics. Edited by N. Mastronardi. Hindawi Publishing Corporation, 2013, p. 683053. doi: *10.1155/2013/683053*

[27] Evans, P. R. (2001) *Rotations and rotation matrices*, Acta Crystallographica - Section D Biological Crystallography, 57(10), pp. 1355–1359. doi: *10.1107/S0907444901012410*

[28] Tomasi, C. (2014) The Singular Value Decomposition, Introduction to Ground Penetrating Radar: Inverse Scattering and Data Processing, 9781118305003(1), pp. 229–241. doi: *10.1002/9781118835647.ch14*

[29] Golub, G. H. and van der Vorst, H. A. (2000) Eigenvalue computation in the 20th century, Journal of Computational and Applied Mathematics, 123(1), pp. 35–65. doi: *10.1016/S0377-0427(00)00413-1*

[30] Huang, X., Ghodsi, M. and Hassani, H. (2016) *A Novel similarity measure based on eigenvalue distribution*, Transactions of A. Razmadze Mathematical Institute, 170(3), pp. 352–362. doi: *10.1016/j.trmi.2016.08.001*

[31] Nikishkov, G. P. (2004) *Introduction to the Finite Element Method*. Lecture Notes, University of Aizu, Aizu-Wakamatsu 965-8580, p. 7

[32] Radu, F.A. et al. (2019) *Numerical Mathematics and Advanced Applications ENUMATH 2017.* Lecture Notes in Computational Science and Engineering, p. 21-22, ISBN: 978-3-319-96415-7

[33] Frei S. et al. (2017) *Fluid-Structure Interaction*. *Modelling, Adaptive Discretizations and Solvers.* Walter de Gruyter GmbH, Berlin/Boston, p. 243, ISBN: 978-3-11-049426-6

[34] Tamrakar, A. et al. (2016) *Implementation of hybrid DEM-PBM approach to reduce the computational cost of powder mixing modeling*, in Kravanja, Z. and Bogataj, M. B. T.-C. A. C. E. (eds) 26 European Symposium on Computer Aided Process Engineering. Elsevier, pp. 1267–1273. doi: *10.1016/B978-0-444-63428-3.50216-2*

[35] Gockenbach, Mark S. (2010) *Finite-Dimensional Linear Algebra.* Taylor & Francis Group, LLC, p. 515. ISBN: 978-1-4398-8287-0

[36] Auclair, J.-P. et al. (2017) *Implementation of Newton's method with an analytical Jacobian to solve the 1D sea ice momentum equation*, Journal of Computational Physics, 340, pp. 69–84. doi: *10.1016/j.jcp.2017.02.065*

[37] Liu, Q., Li, J. and Liu, J. (2017) ParaView visualization of Abaqus output on the mechanical deformation of complex microstructures, Computers & Geosciences, 99, pp. 135–144. doi: *10.1016/j.cageo.2016.11.008*

[38] Likos, W. and Lu, N. (2014) *Hystersis of capillary cohesion in unsaturated soils*. 15th ASCE Engineering Mechanics Conference, June 2-5, 2002, Columbia University, New York, NY, p. 5

[39] Liu, X. and Xu, M. (2017) 'The Unsaturated Hydromechanical Coupling Model of Rock Slope Considering Rainfall Infiltration Using DDA', Geofluids, 2017, pp. 1-2, doi: *10.1155/2017/1513421*

[40] Ponte, K.J. (2004) *Retaining Soil Moisture in the American Southwest*. Sunstone Press, Santa Fe, p. 66, ISBN: 978-0-8653-4411-2

[41] Talvila, E. and Wiersma, M. (2011) 'Simple derivation of basic quadrature formulas', Atlantic electronic journal of mathematics pp. 1–4

[42] Koenig, H.A. (1998) *Modern Computational Methods. Series in Computational and Physical Processes in Mechanics and Thermal Sciences.* Taylor & Francis Group, LLC. Chapter 4.2. Simple Integration Formulas

[43] Rau, G. C. et al. (2019) 'Error in hydraulic head and gradient time-series measurements: a quantitative appraisal', Hydrology and Earth System Sciences Discussions, (April), pp. 1–41. doi: *10.5194/hess-2019-182*

[44] Contreras, I., Grosser, A.T. and Ver Strate, R.H. (2008) *The Use of the Fully-grouted Method for Piezometer Installation*, Geotechnical News, vol. 26, June 2008, pp. 30-32

[45] Hutchison, B.A. and Hick, B.B. (1985) *The Forest-Atmosphere Interaction*. Proceedings of the Forest Environmental Measurements Conference held at Oak Ridge, Tennessee, October 23-28, 1983. D. Reidel Publishing Company, p. 261, ISBN: 978-94-010-8843-5

[46] Technical Service Center Denver and Colorado. (2015) *Procedure for Using Piezometers to Monitor Water Pressure in a Rock Mass*. Materials Engineering and Research Laboratory, code 86-68180, Technical Service Center, Denver, Colorado. Designation USBR 6515, p. 4

[47] McKenzie, N., Coughlan, K. and Creswell, H. (2002) *Soil Physical Measurement and Interpretation for Land Evaluation.* Australian soil and land survey handbook series, vol. 5, CSIRO Publishing, pp. 178-179, ISBN-10: 0-643-06767-1

[48] Tarantino, A., Ridley, M.R. and Toll, D.G. (2008) *Field Measurement of Suction, Water Content, and Water Permeability*. Geotech Geol Eng 26, 751-782, doi: *10.1007/s10706-008-9205-4*

[49] Milczarek, M.A. et al. (2006) *Saturated and unsaturated hydraulic properties characterization at mine facilities: Are we doing it right*, 7th International Conference on Acid Rock Drainage 2006, ICARD - Also Serves as the 23rd Annual Meetings of the American Society of Mining and Reclamation, 2, pp. 11-2, doi: *10.21000/jasmr06021273*

[50] Williams, D. J. (2015) Chapter 2 - Placing Soil Covers on Soft Mine Tailings, in Indraratna, B., Chu, J., and Rujikiatkamjorn, C. B. T.-G. I. C. H. (eds). San Diego: Butterworth-Heinemann, pp. 51–81. doi: *10.1016/B978-0-08-100698-6.00002-7*

[51] Abitew, A. (2012) Hydro-Geotechnical Study of Tailings at Aitik Mine Hydro-Geotechnical Study of Tailings at AITIK Mine. Master's Thesis, Department of Civil, Environmental and Natural Resources Engineering, Lulea University, pp. 23-24

[52] Jantzer, I. (2009) *Critical Hydraulic Gradients in Tailings Dams Comparison Analogies Comparison to natural analogies*. Licentiate Thesis, Department of Civil and Environmental Engineering, Division of Mining and Geotechnical Engineering, Lulea University, pp. 63-65. ISBN: 978-91-7439-055-1

[53] Adajar, M.A. and Zarko, M. (2014) *An empirical model for predicting hydraulic conductivity of mine tailings.* International Journal of GEOMATE, vol. 7, p. 1056, doi: *10.21660/2014.14.140630*