UNIVERSITAT POLIÈCNICA DE CATALUNYA

BARCELONATECH

MSc. in Computational Mechanics

Industrial Training

Spring Semester 2017/2018

Submitted By:

Aditya Anand Mangaonkar

External Supervisor

Dr. Antonia Larese

Internal Supervisor

Prof. Josep Sarrate

ACKNOWLEDGEMENT

I would like to express my deepest gratitude towards Prof. Dr.Antonia Larese who has been the backbone for the successful completion of this industrial training. She always guided me in strenuous conditions and provided me unmatched mentorship to clarify my doubts regarding operation in diverse models and programs.

I would also like to oblige Prof. Sergio Zlotnik for his priceless advice and lecture regarding the forums and organizations which had to be approached to seek an internship as well as his guidance during the masters programme. His priceless assistance proved instrumental to make this internship possible.

Despite the industrial training tenure being short, I thoroughly found myself in the place of conquest to learn new programming skills along with total clarity of mathematical modeling. This industrial training as a whole has however developed me as a person in terms of technique and attitude. Hence, this industrial training shall always serve as the primary calibration for overcoming the future difficulties in my engineering career.

Foremost, I would like to thank my parents Anand Mangaonkar and Nutan Mangaonkar for their immense support and love which always upheld me to conquer my fears and RISE each time with unparalleled dedication.

Contents

1.Introduction

To study flow through porous media a model was introduced by Larese, Rossi And Oñate[1]. It is a modified form of Navier Stokes equations with the principle aim of simulating the seepage flow inside rockfill like porous material and the free surface flow in the clear fluid region. This problem is solved using a semi-explicit stabilized fractional step algorithm where velocity can be calculated using a 4th order Runge–Kutta scheme. The numerical formulation is developed in a Eulerian framework using a level set technique to track the evolution of the free surface. An edge-based data structure is employed.

The main aim of the model is to study flow in rockfill constituting a wall of the dam. According to traditional studies of flow in rockfill at a micro level, the flux between the rocks is assimilated to flow in pipes. This analogy is used for the derivation of the resistance law which is further used for the calculation of the hydraulic gradient at a macro scale due to seepage. The well-known Darcy's law is not applicable to the analysed problem. Here a non-linear resistance law should be adopted for accurate evaluation of the resistance forces made by the porous structure inclusive of the local turbulent flow in the pores. In the following section, the traditional mathematical models to evaluate the hydraulic gradient in rockfill are reviewed and discussed.

The main task in the current work is divided into four parts:

1. First- studying the model for flow through porous media.
2. The second part consists of studying the Kratos framework, installation, compilation of Kratos and execution of programs in Kratos.
3. The third part consists of setting the benchmark example to test the model, with the previous framework.
4. Fourth part includes migration to the updated model and validating it with test results

2.Mathematical Model Formulation

2.1 Seepage into Rockfill

The flow in porous media is traditionally studied using the empirical relation obtained by Darcy in 1856.

$$i \;= \frac{\mu}{k}\, u \tag{1}$$

where '$\boldsymbol{\mu}$' is the water dynamic viscosity and '$\mathbf{k}$' is the permeability of the porous media. Velocity u in Eq. (1) is by definition the Darcy velocity, i.e. the fluid velocity averaged over the total control volume $\boldsymbol{\Omega}_{\text{volume}}$), whereas the fluid velocity u is averaged over the empty part of $\boldsymbol{\Omega}$ (called $\boldsymbol{\Omega}_E$). Their relation is stated by the Dupuit–Forchheimer equation.

$$u \;=\; n\bar{u} \tag{2}$$

where n is the porosity that, by definition is

$$n = \frac{\Omega_E}{\Omega} \tag{3}$$

See Fig. 1 for a graphical view of the fluid and the Darcy definition of velocity. In the most general case n is a function of space and time:

$$n = n\,(\mathbf{X}, t) \tag{4}$$

where $\mathbf{X}$ is the material coordinate vector, In the present work, according to experimental analysis the variation of porosity in time can be neglected, considering only its variation in space, i.e.

$$n = n\,(\mathbf{X}) \tag{5}$$

The relation between velocity and pressure drop was observed to be nonlinear and it was experimentally demonstrated that over the certain average dimension of the particles, the Darcy equation is not any more valid. Many authors have deeply studied this aspect with essentially two objectives: to discover the range of validity of Darcy's law (Eq. (1))

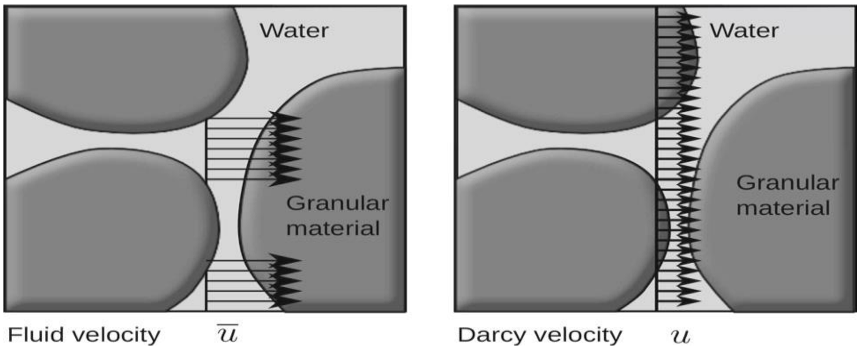

Figure 1 Graphical description of fluid velocity and Darcy velocity Image is taken from (1)

2.2 Analogy Between Flow in Rockfill and Pipes Flow

It is generally accepted to consider the flow in the pores of rock particles essentially similar to flow in a pipe network but with a more complicated configuration. Based on this assumption all the empirical formulae to evaluate the pressure drop due to friction in pipes have been used and adapted to get similar empirical relationships in the case of porous material. In the present work, a quadratic resistance law has been implemented. The user can define the **α** and **β** parameters depending on the type of material studied and on the flow conditions. This aspect will be discussed later in the sections devoted to the validation examples

2.3 Mathematical Model: The Modified Form of the Navier–Stokes Equations

1 Momentum Conservation Equation

The balance of linear momentum is

$$\partial_t \mathbf{u} + \bar{\mathbf{u}} \cdot \nabla \mathbf{u} + n\nabla p - 2\nabla \cdot \nu \nabla^s \mathbf{u} - bn + \mathbf{D} = \mathbf{0} \tag{6}$$

Where, $\partial_t \mathbf{u} = \frac{\partial \mathbf{u}}{\partial t}$ $.\mathbf{D}$. $\mathbf{D}$ is the matrix form of the resistance law and it represents the dissipative effects in the fluid flow due to the presence of the solid matrix. where b is the component of the body force and D represents the component of the hydraulic gradient due to seepage, e.g. the resistance law. The matrix form of the non-linear Darcy law is

$$\mathbf{D} = \alpha \mathbf{u} + \beta \mathbf{u} \cdot \mathbf{u}. \tag{7}$$

where α and β are constant coefficients.

2 The Weak Form of the Problem

The boundary and initial condition of the problem are:

$$\begin{aligned}
&\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}) && \text{in } \Omega; && \\
&\mathbf{u}(\mathbf{x}, t) = \mathbf{g}(\mathbf{x}, t) && \text{on } \partial\Omega_D, && t \in (0, T); \\
&\mathbf{n} \cdot \boldsymbol{\sigma}(\mathbf{x}, t) = \mathbf{t}(\mathbf{x}, t) && \text{on } \partial\Omega_N, && t \in (0, T);
\end{aligned} \tag{8}$$

$\boldsymbol{\Omega}_D$ and $\boldsymbol{\Omega}_N$ are the Dirichlet and Neumann boundary respectively. Note that **n** indicates the outer unit normal vector whereas n is the porosity. The weak form of the equation is derived next using a Galerkin formulation. A mixed finite element method is obtained, for which the approximation of both the velocity components and the pressure (and their respective weighting functions) is introduced. Multiplying by the weighting function and integrating by parts

$$\int_\Omega \boldsymbol{w} \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{t}} d\Omega + \int_\Omega \boldsymbol{w}\bar{\boldsymbol{u}} \cdot \nabla \boldsymbol{u}\, \boldsymbol{d}\Omega + \int_\Omega \boldsymbol{w} n \nabla\, p \boldsymbol{d}\boldsymbol{\Omega} - \mathbf{2} \int \boldsymbol{\Omega} \nabla \boldsymbol{w} : \nu \nabla^s \boldsymbol{u}\, \boldsymbol{d}\Omega$$
$$+ \int_\Omega \boldsymbol{w}(\alpha \boldsymbol{u} + \beta \boldsymbol{u} \cdot \boldsymbol{u}) \boldsymbol{d}\Omega - \int_\Omega \boldsymbol{w} n \boldsymbol{b} \boldsymbol{d}\Omega = \mathbf{0}$$

..........(9a)

$$\int_\Omega q\, \nabla \cdot \boldsymbol{u}\, \boldsymbol{d}\Omega = \mathbf{0}$$

.............(9b)

Stabilized Formulation

In this work low-order, simplicial elements will be used with the same linear interpolation for the velocity and pressure values. Hence, as usual for this type of finite element approximation, a stabilization technique is needed for both the convective term and the incompressibility constraint. In the present work, the *Orthogonal sub-grid scale (*OSS) method is used for stabilization.

Solution Strategy and Time Discretization Scheme

The modified form of the Navier–Stokes equations is solved using a fractional step (FS) algorithm. Pressure-splitting approaches of the fractional-step type are very convenient due to their high computational efficiency for flows at high *Re*. The fundamental idea is to split the momentum equation in two, decoupling the degrees of freedom of the problem. This allows solving the momentum equation keeping the pressure fixed and later to correct the pressure to guarantee the satisfaction of the mass balance constraint. For time integration fourth order Runge Kutta Method is used.

2.4 Edge-based approach.

The advantage of such technique for this specific case is that the pre-computation of operators allows a faster computation of the residual when compared with standard integration rules. The starting point in our work is the systematic use of the partition-of-unity property of the finite element shape functions, which provides the relations

$$\sum_i N_i = 1 \qquad \Rightarrow \qquad N_i = 1 - \sum_{j \neq i} N_j$$

and therefore

$$\sum_i \nabla N_i = 0 \qquad \Rightarrow \qquad \nabla N_i = -\sum_{j \neq i} \nabla N_j$$

The 'edge-based' approach is obtained by applying such relations for the computation of the discrete operators of interest.

2.5 Free Surface Tracking: The Level Set Method

the level set technique the free surface is represented by a signed distance function, the level set function ($\phi(\mathbf{x}, t)$) that takes the value 0 on the free surface, takes negative values in the fluid domain and positive values outside the fluid domain The fluid domain $\Omega(t)$ is evolving during the process and can move in the global domain of analysis Ω_0 according to the solution of the Navier–Stokes problem presented in the previous sections. The level set function, for a given time instant *t*, is defined as the level set function, for a given time instant *t* is defined as

$$\begin{aligned}
&\phi(\mathbf{x}) = d(\mathbf{x}) && \text{if } \mathbf{x} \in \Omega, && t \in (0, T); \\
&\phi(\mathbf{x}) = d(\mathbf{x}) = 0 && \text{if } \mathbf{x} \in \partial\Omega, && t \in (0, T); \qquad (10) \\
&\phi(\mathbf{x}) = -d(\mathbf{x}) && \text{if } \mathbf{x} \in \Omega, && t \in (0, T);
\end{aligned}$$

where $d(\mathbf{x}) = min|\mathbf{x} - \mathbf{x}_i|$ being $\mathbf{x}_i$ is any point on the free surface boundary $\partial\Omega$. The problem is thus transferred to the solution of the following transport problem

$$
\begin{aligned}
&\partial_t \phi + \mathbf{u} \cdot \nabla\phi = 0 && \text{in } \Omega^0, && t \in (0, T), \\
&\phi = \overline{\phi} && \text{on } \partial\Omega_{in}, && t \in (0, T), \qquad (11) \\
&\phi(x, 0) = \phi_0(x) && \text{in } \Omega_0,
\end{aligned}
$$

where $\partial\Omega_{in} := \{x \in \partial\Omega 0 \mid \mathbf{u} \cdot \mathbf{n} < 0\}$ is the inflow part of $\partial\Omega_m$.

2.6 Kratos Framework

Kratos is designed as a framework for building multi-disciplinary finite element programs. Generality in design and implementation is the first requirement. Flexibility and extensibility are other key points in this design, enabling developers to implement very different formulations and algorithms involving in the solution of multi-disciplinary problems. The detailed framework of Kratos is explained in the P. Dadvand et al.[2] and [4]

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

Step1. The geometry is drawn, boundary conditions and parameters are assigned, which depend on the analyzed problem.

Step2. The mesh is generated. The elements used in this work are linear triangular (3 nodes) for 2D models while for the 3D models are linear tetrahedral (4 nodes). GID allows to define fine meshes by zones, therefore the size of the mesh is changed depending on the problem. It should be noted that a refined mesh produces high computational cost and therefore it is advisable to optimize the mesh depending on the needs of the calculation. For this work, the mesh is refined in areas where the flow of water pass.

Step3. The problem type is executed. GID generates a series of files. The most relevant files used for the calculation are:

.mdpa: presents the information about the nodes, elements and boundary conditions generated in the model.

Problem_settings .py: presents the principal parameters to be used.

script.py: defines the calculation to be performed in the Python language. In the calculation stage, the Python file is run, which calls Kratos, passing the necessary information from the files generated in the pre-process by GID.

Step4. Results are displayed in GID in the phase of post-process

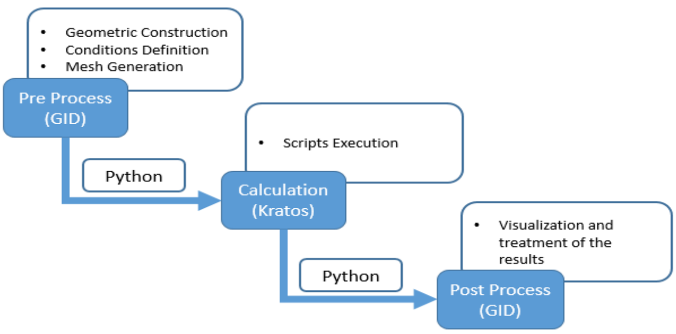

Figure 2 Visualization of results in GID

3. NUMERICAL EXAMPLES

The Examples are set to check the performance of the model in various technical terms. First, we run all the examples in old application -Incompressible Fluid application. Then a new application was updated and developed as per requirement of the current model framework by Antonia Larese which enhanced the performance of the code and proved that it took lesser time in certain simulations.

1.Still-Water

Series of examples are presented to show the good performance of the proposed method. The good capability of reproducing a hydrostatic pressure distribution in case of a still water simulation, without oscillations or loss of volume, is shown in the first example. Verification of the correct pressure calculation in the case of a tank half filled with water. The example is stationary, but in order to ensure that no spurious velocities appear, the simulation time is 10 s and the results are taken from the last step of the analysis. Incompressible water has the following characteristics: kinematic viscosity= 10^6 m^2/s, density =1000 kg/m^3, gravity is 9.81 m/s in the vertical direction.

From the results, we can claim that there is no deviation in both the results (Incompressible fluid application and Free surface application) and the tank is half filled so we have to change the python script manually in this example. We must design geometry according to the coordinate system. The level set function is zero at the boundary, negative in the fluid domain and positive outside the domain. The simple linear function is to be made to operate the nodes. The below table describes the pressure distribution in the tank which we can see in the post process file

rho	g	h	p
1000	9.81	1	9810
1000	9.81	2	19620
1000	9.81	3	29430
1000	9.81	4	39240
1000	9.81	5	49050

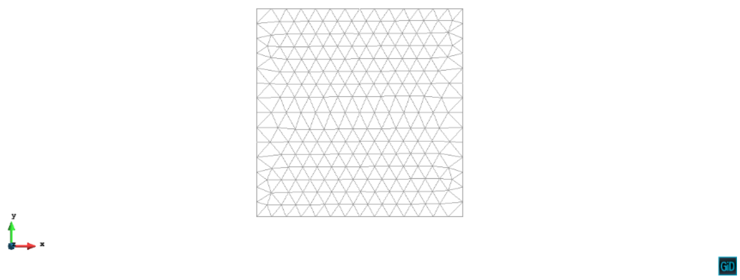

Figure 3 Mesh formation of still water tank

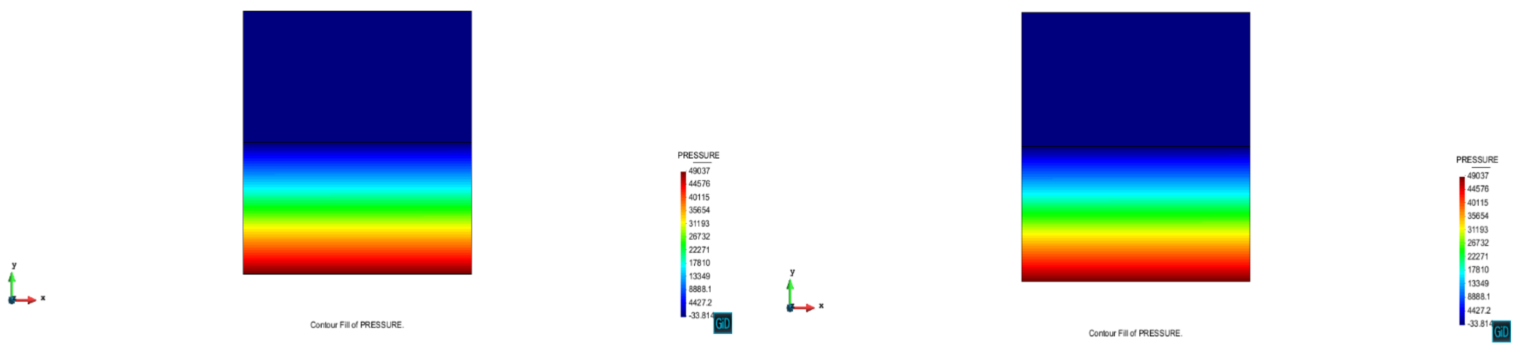

Figure 4 Postprocess results of still water example fig. on left from the Incompressible fluid application and fig. on right results from the free surface application of pressure inside the tank

2. Vertical Column

The second example is a vertical rectangular column constituting an inlet in the bottom side and an outlet on the top face. The inlet is 1 m^2 and its height is 5 m the flow of velocity is from downwards to upwards, so the total time required to fill the column totally is 5 sec.

The tricky part is the expressing the flow of velocity and the direction of body forces. In this case, the flow of body forces and the velocity should be expressed properly otherwise we will not be able to attain the desired results.

The distance function for the bottom surface of geometry is set to -1 and the top is set to be at a constant pressure which forms the exit of the geometry.

The total simulation is made to run for 10 sec with the timestep of 1. The results achieved by both the applications are same in case of column filling as well as pressure, so we can say that there is no variation in the output. The problem presents the same difficulties of maintenance of a flat free surface both in the transitory and in the stationary regime.

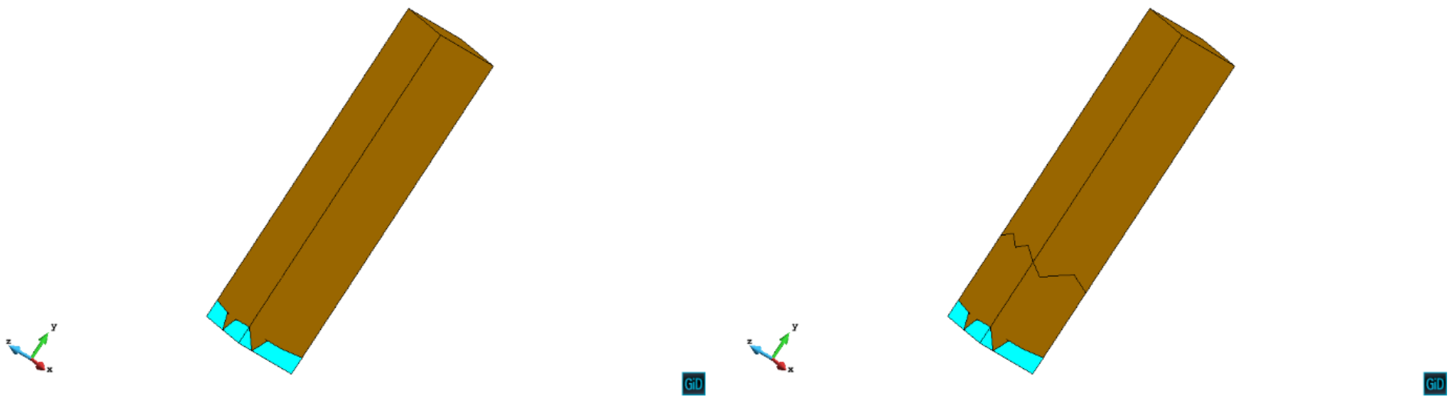

Figure 5 Initial Postprocessl results of water feeling of vertical column example fig. on left from the Incompressible fluid application and fig. on right results from the free surface application

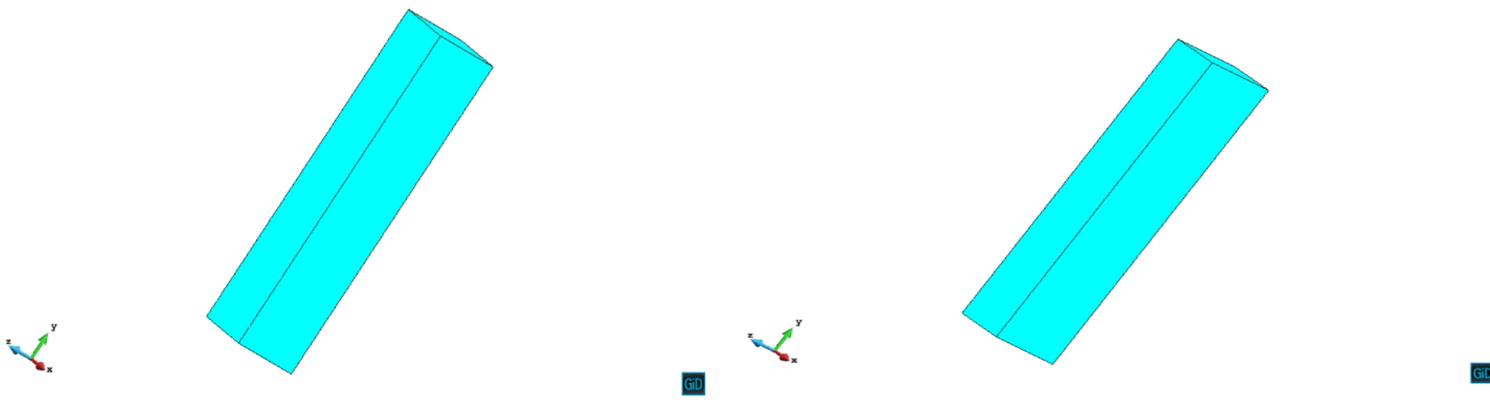

Figure 6 Final Postprocess results of water feeling of vertical column example fig. on left from the Incompressible fluid application and fig. on right results from the free surface application

3.Vertical Column with a lateral opening

Water filling

The above geometry is slightly changed and an opening of 0.2*0.2 is provided laterally, instead of bottom opening. So, to get fill up column vertically, the time is kept constant to 10 sec but the velocity is varied to 2.8 m/s in X direction which is the opening provided. But, the body forces are still acting in Y direction so first, the water comes in the X direction, the slip condition does not allow it to go further so it rises in Y direction which we can see below in the following results, hence it proves that directions in modeling are correct.

Now the estimated time to get filled the column is 10 seconds. The last image shows that the column is perfectly filled so we can say that volume conservation modeling is correct.

Figure 7 Initial Postprocess results of I water feeling of vertical col. with lateral opening example fig. on left from the Incompressible fluid application and fig. on right results from the free surface application

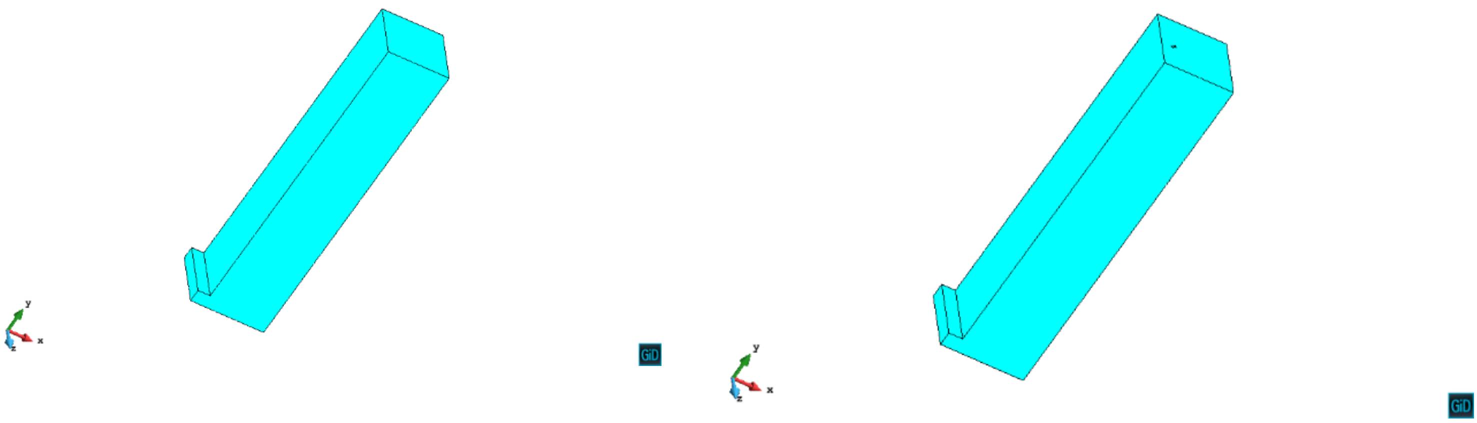

Figure 8 Final Postprocess results of initial water feeling of vertical col. with lateral opening example fig. on left from the Incompressible fluid application and fig. on right results from the free surface application

4.Dam with Porous wall

The wall of the dam is considered as porous material and the porosity considered as 0.5 and the diameter of the porous material consider as 0.035 m. The results are completely the same in both the cases we can see below results. The geometry is shown as figure. The fixed velocity, slip condition and the fixed pressure conditions are provided in the problem type. We have to say to the code to consider the presence of a porous medium. It is not sufficient to do it just setting porosity and diameter. We have to go to the problem parameter menu and set compute porous resistance law to 1 or 2 we have considered porous resistance law 2 here with linear Darcy co-efficient and Non-linear Darcy coefficient, as explained in the model above. The results achieved by the model are validated with the actual test results.

Both the results are almost same, but we can say that in case of incompressible fluid application different pressure ranges are seen than Free Surface application and in post process output minutely varied.

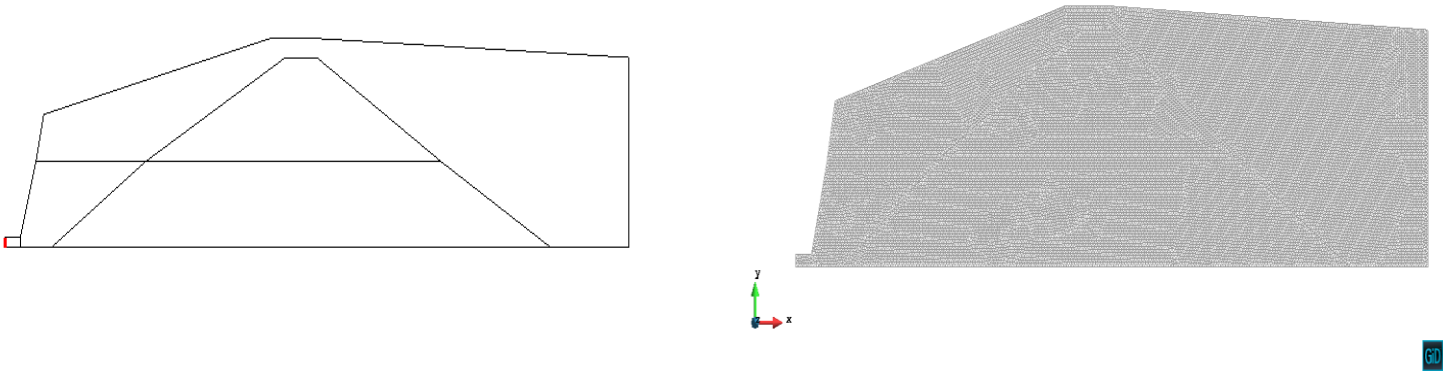

Figure 9 Geometry and mesh of Dam example

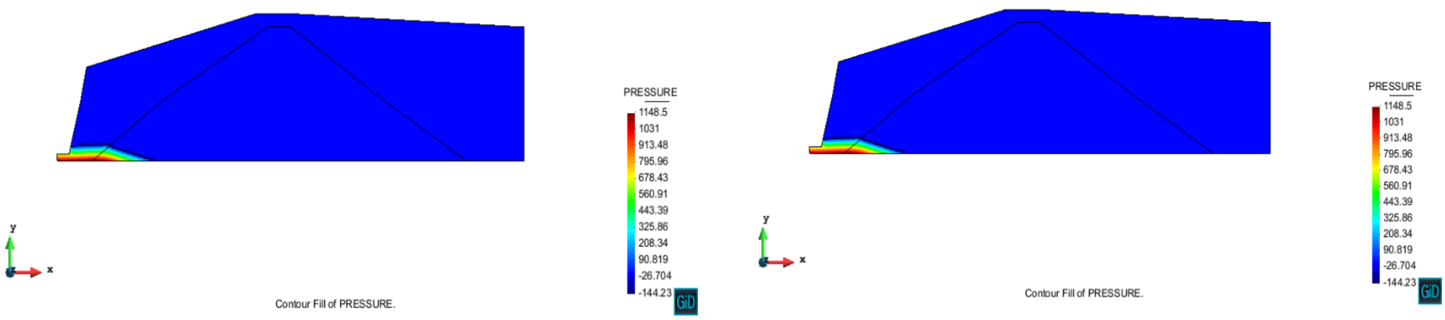

Figure 10 Initial Postprocess results of pressure inside dam wall fig. on left from the Incompressible fluid application and fig. on right results from the free surface application

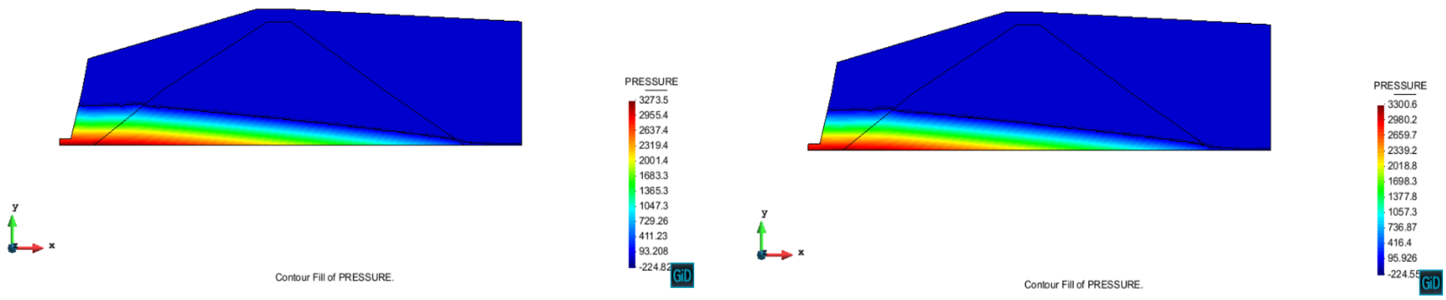

Figure 11 Final Postprocess results of dam example fig. on left from the Incompressible fluid application and fig. on right results from the free surface application

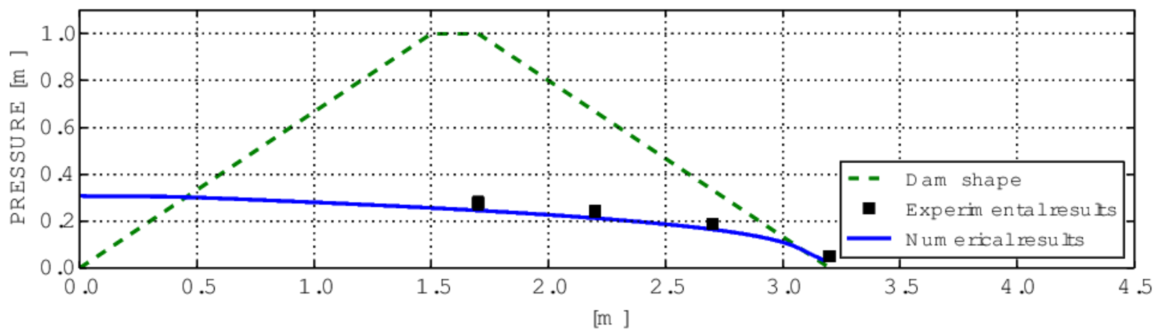

Figure 12 Comparison between the numerical and experimental pressure

4.Conclusion

In the present work, a model for the simulation of the seepage evolution in porous materials and free surface flow has been studied. A unified formulation of the Navier–Stokes equations has been derived to considering flow through the porous material. The formulation is stabilized using Orthogonal Sub Grid Scale stabilization method and solved using 4rth order Runge Kutta method. The model is developed to represent seepage problems that surpass Darcy law, including rockfill or other materials with extremely high permeability. For this purpose, a Forchheimer resistance law has been added as a dissipation term in the linear momentum equations. The prime aspect of the model is that the governing equations reduce to the classical Navier– Stokes equations in the clear fluid region after studying and understanding the level set method thoroughly along with the framework of the Kratos and GID interface. The Incompressible fluid application which forms an old application is also studied, some examples are set on it to evaluate the application and same examples are run into the newly updated application which is the free surface application.

After simulating the model in both- incompressible fluid application and free surface application we can conclude that free surface application has generated more accurate and time optimized model.

Both the results are approximately equal, but we can say that the generated results of incompressible fluid application vary minutely with respect to Free Surface application.

References

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

[2]Dadvand P, Rossi R, Oñate E (2010) An object-oriented environment for developing finite element codes for multi-disciplinary applications. Arch Computation Methods Eng 17:253–297

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

[4]GID the personal pre and post processor. http://www.gidhome.com/

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