Chapter 2

Direct numerical simulation in fully saturated porous media

C. Soulaine

This chapter is organized as follows. First, we introduce the equations of fluid mechanics, namely the Navier-Stokes equations. Then, we propose some recipes to face situations of practical interest like the simulation of flow in fully saturated micromodels or the computation of the absolute permeability from micro-CT images. Finally, we describe the most common numerical procedures to solve these equations.

2.1 Mathematical model

To estimate effective parameters such as permeability or dispersion tensors, it is crucial to compute the velocity field within the void space of the porous medium. In Figure 2.1 we have represented a schematic of a fully saturated porous medium. The void-space is denoted $V_\beta$ and the volume occupied by the solid is $V_\sigma$. The velocity profile of the fluid flowing through the void-space can be obtained solving the base equations of fluid mechanics, namely the Navier-Stokes equations. They are composed by a set of mass and momentum balance equations. For an incompressible fluid, they read,

\[
\nabla \cdot \mathbf{v}_\beta = 0 \text{ in } V_\beta,
\]

and

\[
\frac{\partial \rho_\beta \mathbf{v}_\beta}{\partial t} + \nabla \cdot (\rho_\beta \mathbf{v}_\beta \mathbf{v}_\beta) = -\nabla p_\beta + \rho_\beta \mathbf{g} + \mu_\beta \nabla^2 \mathbf{v}_\beta \text{ in } V_\beta,
\]
Figure 2.1: Schematic of a fully saturated porous medium. The volume occupied by the solid grains is denoted $V_\sigma$ and the void-space $V_\beta$. The interfacial area between the two domains is $A_{\beta\sigma}$.

where $\rho_\beta$, $v_\beta$, $p_\beta$, $g$ and $\mu_\beta$ respectively stand for the fluid density, the velocity profile, the pressure field, the gravity and the viscosity. The effect of stress in the fluid is represented by the pressure gradient and the viscous forces. The effect of the pressure gradient on the flow is to accelerate the fluid in the direction from high pressure to low pressure.

Besides the equations of motions a boundary condition on the fluid/solid surface area, $A_{\beta\sigma}$, is needed. The no-slip condition at a solid boundary states that for viscous fluids the fluid have zero velocity relative to the boundary,

$$v_\beta = 0 \text{ at } A_{\beta\sigma}.$$  \hspace{1cm} (2.3)

### 2.1.1 Non-dimensionalized Navier-Stokes equations

When dealing with flows in porous media, we often consider very slow mass flow rates and, as a matter of fact, the advective inertial forces in Navier-Stokes equations are small compared with viscous forces. To gain a greater insight into the relative size of various terms present in the Navier-Stokes it is useful to deal with non-dimensionalized equations. This process of non-dimensionalization requires the selection of reference values. Assuming $L_0$ the characteristic length of the system, $U_0$ the reference velocity, $t_0$ a characteristic time and $p_0$ the reference pressure, then we can introduce the dimensionless variables $t^* = \frac{t}{t_0}$, $\nabla^* = \frac{\nabla}{L_0}$, $v_\beta^* = \frac{v_\beta}{U_0}$ and $p^* = \frac{p}{p_0}$. Hence, the Navier-Stokes equation in absence of gravity can be recast into,

$$\frac{\rho_\beta U_0}{t_0} \frac{\partial v_\beta^*}{\partial t} + \frac{\rho_\beta U_0^2}{L_0} \nabla^* : (v_\beta^* v_\beta^*) = -\frac{p_0}{L_0} \nabla^* p_\beta^* + \frac{\mu_\beta U_0}{L_0^2} \nabla^{*2} v_\beta^* \text{ in } V_\beta.$$  \hspace{1cm} (2.4)
Figure 2.2: Schematic of flow past an obstacle. a) In the creeping flow regime, the flow is model by the Stokes equation and the streamlines embrace the solid contours. b) For higher mass flow rate, the inertia effects deform the streamlines and we notice flow recirculation after the obstacle in the streamwise.

By definition of the characteristic time we have $t_0 = \frac{L_0}{U_0}$. Moreover, for viscous dominated fluid, it is common to define the reference pressure as, $p_0 = \frac{\mu \beta U_0 L_0}{\rho \beta g}$. So the Equation (2.4) reads,

$$Re \left[ \frac{\partial \mathbf{v}^*_{\beta}}{\partial t} + \nabla^* \cdot (\mathbf{v}^*_{\beta} \mathbf{v}^*_{\beta}) \right] = -\nabla^* p^*_{\beta} + \nabla^* \mu^*_{\beta} \mathbf{v}^*_{\beta} \text{ in } V_{\beta},$$

(2.5)

where we have introduced the dimensionless Reynolds number, $Re = \frac{\rho \beta U_0 L_0}{\mu \beta}$. This number is defined as the ratio of inertial forces to viscous forces and consequently quantifies the relative importance of these two types of forces for given flow conditions. For very slow mass flow rates we have very low Reynolds number, i.e., $Re < 1$ so the inertia effects can be neglected and the momentum balance equation Eq (2.2) can be reformulated as the Stokes equation,

$$0 = -\nabla p_{\beta} + \rho \beta g + \mu \beta \nabla^2 \mathbf{v}_{\beta} \text{ in } V_{\beta}.$$  

(2.6)

In such flow regime also called the creeping flow regime, the streamlines embrace the porous solid structure (see Figure 2.2a). Actually a Stokes flow has no dependence on time and the solution only depends on the geometry of the void space. From a fluid mechanics point of view, the permeability is seen as the coefficient of the averaged drag force due to the viscous friction at the fluid/solid surface. On the other hand, for higher mass flow rate, when inertia effects depicted by the left-hand side of Eq (2.2), start to impact the solution, i.e. $10 < Re < 2300$, then the streamlines are deformed by the presence of the solid structure and flow recirculations are generated downstream the obstacles in the streamwise as illustrated in Figure 2.2b. This phenomena is responsible of the deviation to Darcy’s law for high mass flow rates. It is accounted for in the continuum representation by adding a correction to Darcy’s law.
2.1.2 Viscous flow between parallel plates

In some particular configurations, the Navier-Stokes equations have some simple analytical solutions as, for instance, in case of viscous flow between two parallel plates, see Figure 2.3. In such a case, we can suppose that there is a uniform effective pressure gradient in the \( x \)-direction, so that \( \nabla p = \frac{\Delta P}{L} \mathbf{e}_x \). Moreover, considering that the dimensions of the plates are much larger than the thickness \( h \), then we can consider that the flow is only carried by the \( x \) component and only depends on \( z \), i.e., \( v_\beta = v_x(z) \mathbf{e}_x \). Hence, the momentum equation reads

\[
\mu_\beta \frac{\partial^2 v_x}{\partial z^2} = \frac{\Delta P}{L},
\]

with the non-slip boundary conditions at the walls,

\[
v_x \left( z = \pm \frac{h}{2} \right) = 0.
\](2.8)

The integration of this boundary values problem leads to the following parabolic profile,

\[
v_z(y) = \frac{1}{\mu_\beta} \frac{h^2}{8} \frac{\Delta P}{L} \left[ 1 - \left( \frac{2y}{h} \right)^2 \right].
\](2.9)

It is the 2D solution of a Poiseuille flow. We can notice that the maximum velocity, \( v_{z,\text{max}} = \frac{h^2}{8\mu_\beta} \frac{\Delta P}{L} \), is reached in the middle of the channel. Though very simple, this parabolic equation has lot of...
applications in porous media modeling. If it is averaged over the thickness, we obtained the Hele-Shaw relation (\cite{10}),

\[
\langle v_x \rangle = \frac{1}{h} \int_{-h/2}^{h/2} v_x(y) \, dy = \frac{h^2}{12 \mu \beta} \frac{\Delta P}{L}.
\] (2.10)

The Hele-Shaw cells consist in two parallel plates fixed at a small distance sandwiching a viscous fluid. They are widely used in fluid mechanics experiments to visualize the flow patterns. In a sense, the micromodels are Hele-Shaw cells with obstacles to mimic a porous medium. This Hele-Shaw equation that relates the average velocity to the pressure gradient remind us the Darcy’s law with a permeability \( K = \frac{h^2}{12} \) and a porosity \( \varepsilon = 1 \). Actually, this simple integration was historically the first step towards the derivation of Darcy’s law from Stokes momentum equation (see \cite{12} for an interesting history review from Hele-Shaw first experiments to Darcy’s law formulation).

Equation 2.9 also has application for flows in fractured media. Indeed, at leading order, the fluid flow through fractures is conceptualized by using the assumption of laminar flow between parallel plates. The total flow between parallel, \( q = \langle v_x \rangle hw \) is equal to,

\[
q = \frac{h^3 w}{12 \mu \beta} \frac{\Delta P}{L}.
\]

We recognize, here, the cubic law where \( h \) represents the fracture aperture.

### 2.1.3 Viscous flow through a capillary

Another important analytical solution of the Navier-Stokes equations is the Poiseuille flow in a capillary, see Figure 2.4. This solution, first proposed by Poiseuille for the study of blood flows in capillary, has a lot of impacts in porous media research since it is at the origin of the well-known Kozeny-Carman correlation to estimate the permeability according to the porosity.

Considering that the tube is long enough compared to its radius and due to axial symmetry, we can assume that \( \nabla P = \frac{\Delta P}{L} \mathbf{e}_x \) and that the velocity is only carried by the x component that only depends on the radial position, \( v_\beta = v_z(r) \mathbf{e}_z \). In cylindrical coordinates, the Stokes equation
becomes,

\[ \mu_\beta \frac{1}{r} \frac{\partial}{\partial r} \left( r \frac{\partial v_z}{\partial r} \right) = \frac{\Delta P}{L}, \]  

(2.11)

with the non-slip boundary conditions at the walls,

\[ v_z(r = R) = 0. \]

The integration of this boundary values problem leads to the following parabolic profile,

\[ v_z(r) = \frac{R^2}{4\mu_\beta} \left( 1 - \left( \frac{r}{R} \right)^2 \right) \Delta \frac{P}{L}. \]

It is the 3D solution of a Poiseuille flow. We can notice that the maximum velocity, \( v_{z,\text{max}} = \frac{R^2}{4\mu_\beta} \Delta \frac{P}{L} \), is reached in the middle of the capillary. The intrinsic average of this equation yields in,

\[ \langle v_z \rangle_\beta = \frac{2}{R^2} \int_0^R v_z(r) r \, dr = \frac{R^2}{8\mu_\beta} \Delta \frac{P}{L}. \]

A very simplified view of a porous medium is a bundle of capillaries. This idealised porous medium is not really made up of actual straight tubes, so we can introduce a tortuosity effect, \( C_0 \), and define the hydraulic radius \( R_h \) instead as \( R \) to account for the non-circular shape of the capillary. With such considerations, the superficial average velocity reads,

\[ \langle v_x \rangle = C_0 \frac{\varepsilon R_h^2 \Delta P}{8\mu_\beta \frac{L}{V}}. \]

The hydraulic radius is defined as the ratio between the volume occupied by the fluid over the wetted surface, \( R_h = \frac{2\varepsilon}{a_s(1-\varepsilon)} \), where \( a_s \) is the specific surface area, \( i.e., \) the surface area of the pore space per unit volume of solid. For a porous medium made up of uniform spheres of diameter \( d_p \), we have \( a_s = \frac{6}{d_p^2} \). Considering the Kozeny constant, \( \frac{C_0}{2} = 5 \), we obtain the famous Kozeny-Carman equation,

\[ K = \frac{d_p^2 \varepsilon^3}{180 (1-\varepsilon)^2}, \]

that relates the permeability to the porosity. Actually, this permeability is just a gross approximation and is only valid for a very limited pore-scale structure. A more general method to evaluate the absolute permeability from a given porous structure is to solve numerically the Navier-Stokes equation within this complex geometry and then deduce the permeability by linking the volume average computed velocity and the pressure drop.
2.2 Numerical methods to solve Navier-Stokes equations

Beside very few situations, the Stokes and Navier-Stokes equations do not have simple solutions and need numerical approximations to be solved. Nowadays, all the commercial or open-source Computational Fluid Dynamics (CFD) codes embed very efficient Stokes or Navier-Stokes solvers that can deal with very large grids and complex geometries. For instance a Stokes flow simulation in a domain that contains up to 13 millions cells solved with the CFD software OpenFOAM® and performed with 256 cores converges within 5-10 minutes.

Before going further, it is important to mention that all the programs designed to solve Navier-Stokes can solve Stokes as well without any modifications: the inertia term will be negligible in front of the viscous term and will not impact the solution! To solve the pressure-velocity coupling, most of the fluid dynamics software are based on the Pressure Implicit Operator-Splitting (PISO) algorithm ([7]) for transient solutions or the Semi-Implicit Method for Pressure Linked Equations (SIMPLE) algorithm ([11]) for steady-state (see Table 2.1). These are predictor-corrector methods which means that the pressure and the velocity equations are solved in a sequential way. The principle of these algorithms is described in the last Section of this chapter. Note that because of the stationnarity of the Stokes equations, it is more relevant to use the SIMPLE algorithm.

<table>
<thead>
<tr>
<th>algorithm</th>
<th>transient</th>
<th>steady-state</th>
<th>comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>PISO</td>
<td>yes</td>
<td>yes</td>
<td>Can be used to find the stationary solution by solving all the time steps</td>
</tr>
<tr>
<td>SIMPLE</td>
<td>no</td>
<td>yes</td>
<td>Much faster than PISO to converge to the steady state</td>
</tr>
</tbody>
</table>

Table 2.1: Summary of the solution algorithms that can be used to solve Navier-Stokes according to the type of simulation.

The PISO algorithm is not unconditionally stable! To handle the instabilities, the time-step size required some restrictions. This is usually achieved using a small fixed time step or an adjustable $\delta t$ based on a CFL condition. Usually, CFL = 1 is enough.

The SIMPLE algorithm requires user-defined under-relaxation factors for the pressure, $\alpha_p$, and the velocity matrix, $\alpha_U$. The usual recommended values for $\alpha_p$ and $\alpha_U$ are 0.2 and 0.8 respectively. Actually these values depend on the Reynolds number. For Stokes problem, we have noticed that the simulations can converge up to twice faster with $\alpha_U = 0.9$.

Some CFD packages propose a solution algorithm that combine both the PISO and the SIMPLE features in order to solve the transient Navier-Stokes equations which larger time-steps up to ten times than those allowed by the PISO algorithm. Basically, they use a SIMPLE loop within each time step, in the framework of a PISO algorithm. In OpenFOAM®, this PISO/SIMPLE method is referred to as PIMPLE.
2.3 Practical example: flow simulation in a saturated micromodel

In this section, we illustrate how to run direct simulations in fully saturated micromodels with OpenFOAM®. Figure 2.5 represents a 2D slice of a sandstone, 442 pixels by 1274 pixels with a 1.5 µm/px (0.63 mm × 1.83 mm), obtained from a 3D micro-CT image following the protocol described in [4]. In this picture the black region represents the void space and the white area the solid grains. To perform microfluidic experiments, the porous medium pattern is usually etched onto silicon using standard photolithography techniques, with an etching depth, \( h \), varying in the range \([10 – 30 \mu m]\) and the etched silicon wafer is bonded to a glass plate in order to obtain a kind of Hele-Shaw cell (see Figure 2.5b). To reduce the computational efforts, the flow in this micromodel will be simulated in 2D. Actually, there are 3D effects that play an important role and must be accounted for. Basically we assume a Poiseuille profile in the depth of the micromodel. Hence, integrating the Stokes equation Eq 2.6 over the depth and neglecting gravity effect, we have the following momentum equation,

\[
0 = -\nabla p_\beta + \mu_\beta \nabla^2 v_\beta - \frac{12}{h^2} v_\beta \text{ in } V_\beta, \tag{2.12}
\]

where the Darcy-like term \( \mu_\beta \frac{12}{h^2} v_\beta \) is a Hele-Shaw correction ([10]). Solving this modified Stokes equation, all the simulation results are in fact averaged results over the micromodel vertical cross sections. The implementation of this additional term in the momentum equation is straightforward and one can refers to Section 2.6 to learn how implement an implicit source term into the momentum equation.

Figure 2.5: Image of the slice of a sandstone where the void space (in black) has been gridded

The binary image is first vectorialized, extruded and exported to a surface object format (.stl or...
.obj usually). Once the surface object is created, the pore space can be mesh using *snappyHexMesh* the automatic mesher of OpenFOAM®. The void space (in black) is then gridded with *snappyHexMesh*. Basically, this utility recognizes the intersections between the provided surface object and a background mesh that consists purely of hexes and then removes the cells occupied by solid (see Figure 2.6 for a schematic of the gridding workflow).

![Workflow of the gridding procedure](image)

**Figure 2.6**: Workflow of the gridding procedure, from a binary image to the final mesh. *(a)* The initial image is first binarized, *(b)* then vectorialized (with Inkscape for example), *(c)* the resulting image is transformed into a surface mesh (*.stl* or *.obj* format can be obtained with Blender for example), *(d)* the surface mesh is superimposed to a background cartesian grid and the intersection between the two mesh is recognized, *(e)* finally, the cells identified as solid grains are removed. The quality of the final mesh strongly depends on the level of refinement of the background grid.

The gridding process is of the greatest importance since the accuracy of the simulation results will strongly depends on it. To obtain a mesh of good quality, the following best practices must be kept in mind. First, in order to capture all the complex feature of the entire poral structure with such a workflow, the background grid must contain at least as much cells as the base image resolution, namely 440 × 1300 cells in this example. However, this rule only will not garantry that the numerical results will correspond to the actual flow solution. Indeed the base image resolution should not be mistaken with the mesh quality requirement! As we can notice in Figure 2.8a, the void-space of a porous medium is made of a conglomerate of narrow capillaries. In order to get
Figure 2.7: Mesh convergence study: plot of the highest velocity value according to the number of cells in the simulation domain

To solve the flow in the creeping flow regime, we have applied a pressure difference, $\Delta P$, between the top and the bottom of the model low enough to be sure that the inertia effects are neglectible. The right and left boundaries are considered as walls. We have used a Navier-Stokes solver based on the SIMPLE algorithm. The under-relaxation factors were set to 0.2 for pressure and 0.9 for velocity. The simulations were considered to be fully converged when the residuals were below $10^{-9}$. The pressure and velocity fields normalized by the highest value are plotted in Figure 2.8a and Figure 2.8b,c respectively. In this simulation, we clearly notice some areas of preferential flow and some dead-end zones. Then, in such configurations, the transport of a passive tracer will be dominated by convection into the fast region and by diffusion into the dead-end zones. If the heterogeneity of the local velocity profile is very emphasized, then we could observe non-fickian behavior at larger...
Figure 2.8: Plot of a Stokes simulation results in a sandstone replica. The resulting fields are normalized by the highest value. (a) Pressure field, (b) Magnitude of the velocity field, (c) Velocity vectors in a zoom in.

Note that to check if the simulation really corresponds to the creeping flow when using a Navier-Stokes solver, and then that the inertia term in the momentum is negligible, a simple test consists in inverting the pressure drop direction, solves the flow and volume integrates the resulting velocity field. If the values are not the same, this means that the difference comes from inertia effects.

2.4 On the use of cyclic boundary conditions

One of the most important application of pore-scale simulation is the estimation of effective properties like the permeability or the dispersion tensors. The very definition of these parameters relies on the concept of REV, which means that the porous medium structure is assumed to be represented locally by a periodic geometry. Two points must be treated carefully when running a cyclic simulation:

- First, to treat two boundaries as though they were physically connected, each pair of connect-
ing faces must have similar area to within a given tolerance. If this constraint can be easily satisfied for rather simple porous media as the one depicted in Fig. 2.9, the situation is much more complicated for realistic pore structure as the ones obtained from micro-CT images. We will see in this Section several tricks to overcome this difficulty.

- When solving Navier-Stokes equations with periodic conditions, the system unknowns at each cyclic pair will have the same values. In particular, this is a problem for the pressure field since it means that there is no pressure drop and therefore no flow in the domain... The trick consists in decomposing the pressure field into a value averaged over the REV and a deviation field $\tilde{p}_\beta$. Hence the pressure gradient becomes $\nabla p_\beta = \nabla \tilde{p}_\beta + \frac{\Delta P}{L} e_0$ where the body force $\frac{\Delta P}{L} e_0$ stands for a macroscopic pressure drop per unit length. The value of $\frac{\Delta P}{L}$ governs the magnitude of the mass flow rate while the vector $e_0$ determines the average flow field orientation. Substituting the new expression of the pressure gradient into the Stokes momentum equation leads to,

$$0 = -\nabla \tilde{p}_\beta + \mu_\beta \nabla^2 v_\beta + \frac{\Delta P}{L} e_0 \text{ in } V_\beta. \quad (2.13)$$

The new unknowns of the system are then $(v_\beta, \tilde{p}_\beta)$ and there is no longer any restriction to define periodic boundaries. Note that by construction, the average of $\tilde{p}_\beta$ over the computation domain is zero. Following the discretization procedure introduced in Section 2.6, the implementation of this Stokes-like problem is straightforward and the only required modification is to add the explicit source term $\frac{\Delta P}{L} e_0$ into the momentum equation.

### 2.4.1 Simulation in an homogeneous porous medium model

To illustrate how to run periodic simulation in REV, we have applied the above method to simulate the flow through a simplified porous medium made of homogeneously distributed solid cylinders. Figure 2.9 shows the unit-cell of such geometry and its dimension. The top/bottom and the left/right boundaries are defined as pair cyclic conditions respectively. The computational domain is gridded following the workflow described in the previous Section. In this example, the average flow is oriented by 45 from the horizontal axis which means that $e_0 = \frac{1}{\sqrt{2}} \left( \begin{array}{c} 1 \\ 1 \end{array} \right)$. The value of $\frac{\Delta P}{L}$ is small enough in order to be sure to remain in the creeping flow regime. The flow is solved with SIMPLE algorithm. The simulations were considered to be fully converged when the residuals was below $10^{-9}$. The results are plotted in Figure 2.9. They are normalized by the maximum value of the displayed domain. For the pressure, the results must be interpreted carefully. Indeed, the unknown we have solved using Eq (2.13) and plotted is not the physical pressure but the deviation to the averaged pressure\(^1\). To obtain the real pressure field, the pressure gradient relationship,

\(^1\)Actually when solving a Navier-Stokes problem, the pressure field is off by an additive constant. Most of the time, this constant is determined by a fixed value boundary condition. However, in case of periodic conditions the
Figure 2.9: Schematic of the flow past homogeneously distributed solid cylinders. The results are normalized by the highest value of the computational domain. (a) Geometry, dimensions and boundary conditions in the REV of this porous medium, (b) Plot of $\tilde{p}_\beta$, the deviation to the average pressure, (c) Plot of the velocity vectors in the REV.

$$\nabla p_\beta = \nabla \tilde{p}_\beta + \frac{\Delta P}{L} e_0,$$ should be integrated over the REV.

### 2.4.2 Predictions of absolute permeability from micro-CT images

Due to the recent improvements in pore-scale imaging, it is now possible to have access to the exact three dimensional geometry of a piece of the rock, to grid the void space and then to solve the flow within it. Assuming that the sample is REV, it is then possible to use characterization techniques based on homogenisation to obtain effective parameters such as the absolute permeability or the dispersion tensor. For instance, once the velocity field is computed, the absolute permeability tensor coefficients, $K_{ij}$, can be merely deduced by averaging the simulation results and inversing a Darcy’s law,

$$ K_{ij} = \mu_\beta \langle v_{\beta,i} \rangle \left( \frac{\Delta P}{L} \right)^{-1} i = x, y, z. \quad (2.14) $$

boundary conditions are of no use to fix this constant and in order to help the convergence, the trick consists to arbitrarily set a reference value to a cell of the mesh ($\tilde{p}_\beta[refCell] = 0$ most of the time). When post-processing the results, a preliminary stage consists to reconstruct the real $\tilde{p}_\beta$. Using the null average property of the deviations, we have: $\tilde{p}_\beta,real = \tilde{p}_\beta,calculated = \langle \tilde{p}_\beta,calculated \rangle^\beta$. 

13
To be consistent with the definition of REV, the flow should be computed with periodic boundary conditions. Usually in the CFD software, this kind of boundary conditions can only be applied to pairs of connecting faces with similar area. However, real pore-space images do not have this geometrical property. Several methods exist to overcome this difficulty. We list two of them:

- In order to obtain pairs of connecting faces with similar area, one possibility is to mirror the domain. If the periodic boundary conditions can be directly used with such a method, this is actually its only advantage. Indeed the computational domain will be multiplied by 4 in 2D and by 8 in 3D... Moreover, mirroring the REV biases the anisotropy of the original material. This method should be used carefully, knowing all its limitations.

- Another possibility consists in surrounding the representative pattern by a thin layer of void space. This way, it is straightforward to apply cyclic boundary conditions. This surrounding should thin enough to only have a low impact on the flow simulation results. Basically, the thickness should be of the same order of magnitude than the average pore throats dimension.

Figure 2.10: (a) Image of a Bentheimer sandstone obtained by micro-CT imaging. Only the solid grains are represented. (b) Clip of the grid that represents the void space. To simulate fluid flow into this sample with periodic boundary conditions, the void space has been surrounded by a small layer of cells.

Figure 2.10a represents the solid grains of a Bentheimer sandstone obtained by micro-CT imaging. The image contains $300^3$ voxels with a resolution of $3 \mu m/\text{voxel}$. Basically, to acquire these data, the sample is rotated with different angles and the absorption of X-rays is recorded in these different directions. The rock and void spaces are then differentiated using a segmentation algorithm. More details about the pore-scale imaging techniques can be found in the recent review paper by Martin Blunt ([2]). To compute the velocity profile within this sample, we mesh the void space following gridding workflow introduced in Section 2.3. In order to use periodic boundary conditions, the rock
sample is surrounded by a thin layer of void space (see Figure 2.10b). This is achieved considering a background grid a little bit larger than the sample. As illustrated in the previous section, one should no mix up the micro-CT resolution and the mesh size requirement which only depends on numerical considerations. Our final grid contains $8.6 \times 10^6$ cells. Pair cyclic conditions are specified for each opposed faces of the computational domain. The pressure gradient is imposed is the $x$ direction ($e_0 = e_x$) and the flow is solved by the SIMPLE algorithm with the values 0.2 and 0.9 for the pressure and velocity under-relaxation factors. The simulations were considered to be fully converged when the residuals was below $10^{-9}$. Results normalized by the highest displayed value are plotted in Figure 2.11. The Figure 2.11b represents the velocity vectors within the void space colored by the pressure field.

![Figure 2.11](image.png)

**Figure 2.11:** Simulation results of the flow within the rock sample. The results are normalized by the highest displayed value. (a) Fluid pressure, (b) Velocity vectors colored by the pressure field.

Once the flow is computed, the procedure to obtain the permeability coefficients $K_{xx}$, $K_{yx}$ and $K_{zx}$ is almost straightforward. First, the average velocity components $\langle v_{\beta,i} \rangle$ are computed for each component by integrating $v_{\beta,i}$ over the computational domain (it is important to exclude the artificial surrounding from the integration) and by dividing the resulting values by the volume of the sample (included both void and rock volumes). Then, $K_{xi}$ are deduced by applying Eq (2.14). To obtain the other coefficients of the permeability tensor, $K$, the flow has to be solved in the 2 other main directions ($e_y$ and $e_z$). Note that, in theory, $K$ must be symmetric ([5]).

## 2.5 Darcy-Brinkman-like formulation

In various situations, it is relevant to introduce a hybrid modeling formulation where the flow in a porous medium is modeled by a Darcy’s law and flow in the free area is governed by Navier-Stokes equations. This is case when a separation of scale is satisfied, which means that the length scale characteristic is the pore-throat of the porous medium is orders of magnitude smaller than those of the free zone. For instance, when studying flow in fractured porous media, it can be useful to
explicitly represent the flow in the fracture solving Navier-Stokes equations while the block matrix is considered as a continuum modeled by a Darcy’s law. From a numerical point of view, two different approaches exist to solve this problem:

1. The two domains approach, where two domains with different physics are considered and communicate through coupled boundary conditions. Actually, the kind of conditions one should impose at the interface between the free fluid and the porous part is still an open debate. Most of the time, the effective conditions experimentally proposed by [1] is used to simulate the pressure jump between the two domains.

2. The single domain approach in which it is assumed that a single equation holds in both the free-flow and the porous-medium domain. The equation to be solved for both domains is the Darcy-Brinkman equation ([3]). The model arises from the integration of Navier-Stokes equations over a control volume in presence of solid material. At the opposite of the first approach, this one does not involve coupling conditions. Indeed, because of the single equation used, it makes stresses and velocities continuous in the entire domain and the transition is denoted by spatial variation of properties such as permeability and porosity across a transition zone. Actually, with such a formulation, the physics at the interface between the domains is an approximation of the reality (no pressure jump). However, this approximation can often be considered as valid. Note that this is also the formulation that most of the CFD softwares (OpenFOAM, Fluent, COMSOL...) embed by default to deal with fully saturated porous media.

In this section, we adopt the second approach. Following [13] developments, we can show that the locally averaged Navier-Stokes equations read

\[ \nabla \cdot (\varepsilon_\beta \rho_\beta \bar{v}_\beta) = 0, \quad (2.15) \]

\[ \frac{\partial \varepsilon_\beta \rho_\beta \bar{v}_\beta}{\partial t} + \nabla \cdot (\varepsilon_\beta \rho_\beta \bar{v}_\beta \bar{v}_\beta) = -\varepsilon_\beta \nabla \bar{p}_\beta + \varepsilon_\beta \mu_\beta \nabla^2 \bar{v}_\beta - \mu_\beta k^{-1} \varepsilon_\beta \bar{v}_\beta^2. \quad (2.16) \]

where the locally averaged pressure and velocity field \( \bar{p} \) and \( \bar{v}_\beta \) are the unknown of the system. \( \varepsilon_\beta \) is the volume fraction of void space per cell and vary from 0 to 1. It is defined as a field with different values all over the domain. This variable differentiates the free zone and the porous medium: for a cell with \( \varepsilon_\beta = 1 \), then there is no solid in it, which corresponds to a free zone, on the other hand, \( 0 < \varepsilon_\beta < 1 \) denotes the porous medium domain. The latest term of the right-hand side of Eq (2.16) represents the momentum exchange term between the fluid and the solid phase, i.e, the Darcy resistance term. The permeability-like \( k \) also varies in space. It should be a function of \( \varepsilon_\beta \) such that the resistance flow term \( \mu_\beta k^{-1} \varepsilon_\beta^2 \bar{v}_\beta \) vanishes in the free zone and is dominant in the porous
medium. For instance, $k$ can be expressed with Kozeny-Carman relationship,

$$k^{-1} = k_0^{-1} \frac{(1 - \varepsilon \beta)^3}{\varepsilon \beta^2}, \quad (2.17)$$

with $k_0$ the permeability of the porous medium. Substituting Eq (2.17) into Eq (2.16), we notice that the locally averaged Navier-Stokes system tends towards

- Navier-Stokes when $\varepsilon \beta = 1$. In this case, $\bar{v}_\beta = v_\beta$.
- Darcy when $0 < \varepsilon \beta < 1$. In this case, the resistance flow term is dominant and we have,

$$\varepsilon \beta \bar{v}_\beta = -\frac{k}{\mu \beta} \nabla \bar{p}_\beta.$$

This mathematical model can be easily implemented in the framework of the PISO or SIMPLE algorithms introduced later in Section 2.6.

### 2.5.1 Flow in fractured porous media

To illustrate the potential of the above model, we simulate the flow in a fractured porous media. The fractures are explicitly represented and the flow within it is governed by Navier-Stokes equations. On the other hand the flow in the porous medium is governed by Darcy’s law. The porosity and permeability are $\varepsilon = 0.4$ and $\varepsilon = 8 \times 10^{-10}$ m$^2$ respectively. The geometry consists in a 2D box, 11 cm x 6.6 cm, gridded with 1280 x 800 cells. The fracture network is differentiated from the porous medium by assigning different values to the volume fraction field $\varepsilon \beta$ (see Figure 2.12). Blue color corresponds to the void space ($\varepsilon \beta = 1$) and red color to the porous medium with ($\varepsilon \beta = 0.4$). We can notice that the fracture networks are disconnected.

We apply a pressure gradient in the horizontal direction from the left to the right and solve the flow in steady-state. Simulation results are plotted in Figure 2.13. As expected the fluid preferentially flows through the fractures with an important flux within the porous area that connects the two networks. We notice that far from this area, the flux in the porous medium is orders of magnitude smaller than in the fractures network.

### 2.5.2 Pore-scale flow with penalized method

This model can also be used as a penalized method to simulate flows in complex pore-space with a fixed cartesian grid in the spirit of the Fictitious Domain method ([9]). Indeed one can consider the solid grain as a porous medium with a very low porosity and very low permeability. Hence, $\varepsilon \beta$ is seen as a mask function to differentiate the void space ($\varepsilon \beta = 1$) and the solid ($\varepsilon \beta \to 0$). With very low values for $k_0$, then the velocity in the solid is near zero and the interfacial boundary between the solid and the fluid is automatically approximated by $v_\beta \approx 0$. For example, the homogeneously distributed cylinders geometry introduced in Section 2.4.1 can be represented in a cartesian grid.
Figure 2.12: Fractured porous medium represented by mapping the void fraction function on a cartesian grid. Blue color corresponds to the void space ($\varepsilon_\beta = 1$) and red color to the porous medium with ($\varepsilon_\beta = 0.4$).

Figure 2.13: Simulation results in the fractured porous medium. The velocity vectors are colored by the pressure field. The fractures contour is displayed for information.
Figure 2.14: Homogeneously distributed cylinders represented by mapping the void fraction function on a cartesian grid. Blue color corresponds to the void space ($\epsilon_\beta = 1$) and red color to the solid with ($\epsilon_\beta = 0.0001$) by the void volume fraction $\epsilon_\beta$ as denoted in Figure 2.14. Blue color corresponds to the void space ($\epsilon_\beta = 1$) and red color to the solid ($\epsilon_\beta = 0.0001$). The flow past this computational domain can be solved by tuning the permeability to a very low value ($k_0 = 10^{-18} \text{ m}^2$). Steady state results are shown in Figure 2.15 and compared with the results obtained solving Stokes equations with boundary condition at the fluid/solid interface. Both results are in perfect agreement.

2.5.3 Solid degradation

This formulation can also be very useful to deal with solid degradation such as dissolution due to chemical reactions at the solid surface or such as thermal degradation like the pyrolysis phenomena in oil shale retorting. In this case, the frontier between the solid and the void space is no longer immobile but moving according to the rate of degradation of the solid, $\dot{m}$. To model that phenomenon we just add an other variable, $\epsilon_\sigma$, to the system with the trivial relationship, $\epsilon_\sigma + \epsilon_\beta = 1$. It corresponds to the volume solid fraction in each cell and it is governed by the following mass conservation equation:

$$
\frac{\partial \epsilon_\sigma \rho_\sigma}{\partial t} = -\dot{m}. \tag{2.18}
$$

where $\rho_\sigma$ is the density of the solid. The solid mass is transformed into fluid, so the conservation law for the fluid turns into,

$$
\frac{\partial \epsilon_\beta \rho_\beta}{\partial t} + \nabla \cdot (\epsilon_\beta \rho_\beta \bar{v}_\beta) = \dot{m}. \tag{2.19}
$$
2.6 Solution algorithms to solve Navier-Stokes equations

We do not pretend to provide an accurate description of the numerics to solve Navier-Stokes equation but to give enough materials to allow the reader to properly set up a Navier-Stokes simulation and understand the philosophy of the usual solution procedures. In the previous Sections we have seen some useful variations to Navier-Stokes equations in porous media. This section will provide simple recipies to customize existing solvers in order to consider these variations. This section aims to present the solution algorithms SIMPLE and PISO as programmed in OpenFOAM®. Readers who are interested to further details on the Navier-Stokes numerical solutions can refer to [6] or to [8] for details specific to OpenFOAM®.

2.6.1 Discretization of the momentum equation

We present in this part a discretisation procedure for the Navier-Stokes equations. The fluid is assumed to be incompressible so the momentum equation, Eq (2.2), is divided by a constant density. The non-linearity introduced by the advection term is overcomed by linearized it using an existing velocity field. The set of equations Eqs (2.1)-(2.2) is then transformed into a set of algebraic equations after application of any standard discretization procedure. Most of the modern CFD softwares use the finite volume method. Using the Euler implicit difference scheme and if $k$ and
\( k + 1 \) denote successive time levels, then the semi-discretized form of the momentum equation reads,

\[
\mathcal{V} \left( \frac{v^{k+1}_P - v^k_P}{\delta t} \right) = -a'_P v^{k+1}_P + \sum_{NP} a'_{NP} v^{k+1}_{NP} - \frac{1}{\rho} \nabla p + g. \tag{2.20}
\]

In this equation, \( \mathcal{V} \) and \( \delta t \) respectively stand for the cell volume and the time step, subscript \( P \) denotes values at the cell center, while the coefficients \( a'_{NP} \) account for the influence of neighbor nodes. They are mainly composed by convective and diffusive fluxes across cell faces. In the spirit of the Rhie and Chow procedure ([?]), the pressure gradient terms is not discretised at this stage.

Introducing the vector \( S = \frac{\mathcal{V}}{\delta t} - g \) that contains all the explicit source terms (the source part of the transient term and all the body force like the gravity acceleration) apart from the pressure gradient, Eq (2.20) can be rearranged as

\[
\left( \mathcal{V} \left( \frac{1}{\delta t} + a'_P \right) \right) v^{k+1}_P = \sum_{NP} a'_{NP} v^{k+1}_{NP} + S - \frac{1}{\rho} \nabla p. \tag{2.21}
\]

We can now introduce the diagonal term \( a_P = \frac{\mathcal{V}}{\delta t} + a'_P \) of the full matrix resulting from the momentum equation discretization, and the operator \( H_a(X) = \sum_{NP} a'_{NP} X_{NP} + S \). Therefore, Eq (2.21) turns to,

\[
a_P v^{k+1}_P = H(v^{k+1}) - \frac{1}{\rho} \nabla p. \tag{2.22}
\]

This semi-discretized equation is generic and the same formulation can be used to program variations of the Navier-Stokes momentum equation. For instance, we have seen in the sub-section 2.4 that in order to deal with periodic boundary conditions, it can be relevant to add a macroscopic pressure drop term, \( \Delta P_{Le0} \), in the momentum balance equation. This extra term is an explicit source term and therefore it will be included in the vector \( S \) in Eq (2.21). Another interesting variation of Navier-Stokes equation consists to consider a Hele-Shaw correction, \( \mu_\beta \frac{12}{l} v_\beta \), to account for the thickness of a micromodel in 2D simulation (see 2.3). For a better stability, this term has to be treated as an implicit source term. This is achieved by adding its contribution into the diagonal coefficients \( a'_P \) in Eq (2.20).

### 2.6.2 Derivation of the pressure equation

The pressure equation is derived from the total mass conservation and the momentum equation. This is achieved by dividing the semi-discretized equation, Eq (2.22), by \( a_P \) and then by taking the divergence of resulting expression. It results the following Poisson equation,

\[
\nabla \cdot \left( \frac{1}{\rho a_P} \nabla p^{k+1} \right) = \nabla \cdot \left( \frac{H(v^{k+1})}{a_P} \right). \tag{2.23}
\]

Such a procedure ensures that the solution of the pressure-velocity \((v, p)\) coupling satisfies the
mass conservation. To deal with more sophisticated mass conservation equation, that involves for example mass source or compressibility, one should derive a suitable pressure equation by substituting the semi-discretized form of $v^{k+1}$ into the mass balance equation.

Note that because of the introduction of an equation for the pressure, a boundary condition has to be specify at the solid boundary. In practice we use a zero gradient condition.

### 2.6.3 The PISO algorithm

We present in this sub-section the Pressure Implicit Operator-Splitting (PISO) algorithm designed by [7] to solve the transient Navier-Stokes equations. The algorithm solve the pressure-velocity $(v, p)$ coupling with a predictor-corrector method. The main steps are described below and sketched in Fig. 2.16. The algorithm we present does not follow exactly Issa’s original paper but is fully consistent with the PISO implementation in OpenFOAM®. We give here the main steps of the procedure.

1. The first stage consists in the prediction of the velocity field, $v^*$, by solving implicitly the momentum equations

   $$ a_P v_P^* = H(v^*) - \frac{1}{\rho} \nabla p^k, $$

   (2.24)

   where the gradient of the pressure field is evaluated from the values computed at the previous time step. This stage is called the momentum predictor. Actually, if the time step is small enough, it is not necessary to solve this equation and only an update of $a_P$ and $H(v)$ with the latest computed velocity profile is necessary.

2. The predicted velocity $v^*$ does not satisfy the continuity equation Eq (2.1) and have to be corrected. This is achieved by looking for $(v^{**}, p^{**})$ that obeys,

   $$ v_P^{**} = \frac{1}{a_P} \left[ H(v^*) - \frac{1}{\rho} \nabla p^* \right], $n

   (2.25)

   $$ \nabla . v^{**} = 0. $$

   (2.26)

Assembling these two equations, the pressure equation can be formulated as

$$ \nabla . \left( \frac{1}{\rho a_P} \nabla p^{**} \right) = \nabla . \left( \frac{H(v^*)}{a_P} \right), $$

(2.27)

and solved implicitly. The corrected velocity $v^{**}$ is then computed point-wise from Eq (2.25). This second step called the PISO loop may be repeated several times to force the convergence. [7] demonstrates that at least two iterations are necessary. In practice it is also sufficient. The resulting values are assimilated to $(v^{k+1}, p^{k+1})$ and then the algorithm increments the time and loops over the whole sequence.
Note that the PISO algorithm is not unconditionally stable! This is typical of the predictor-corrector feature of the method since as the solution to the equations is achieved approximately, the residual errors in these approximations may alter the stability characteristics of the overall scheme. To handle the instabilities, the time-step size required some restrictions. This is usually achieved using a small fixed time step or an adjustable $\delta t$ based on a CFL condition.

Note also that by construction of the algorithm the only way to get a steady state with PISO is to run a transient simulation until the solution reaches its stationarity.

### 2.6.4 The SIMPLE algorithm

When solving Navier-Stokes or Stokes flows in porous media, we are often interested in the fully established regime. Of course, the PISO algorithm can be used until the simulation reach a stationary solution but this process can be quite long since it needs to resolve all the time steps. An alternative approach consists to drop the transient term in the Navier-Stokes momentum equation (Stokes already has a steady-state formulation) and to directly look for a steady-state solution. The Semi-Implicit Method for Pressure Linked Equations (SIMPLE) algorithm by [11] directly reaches this stationary solution by iteratively guesses and corrects the pressure-velocity coupling. In this procedure, the superscript $k$ no longer refers to time level but to an iterative index. The SIMPLE algorithm is formulated as follow (see Fig 2.17),
1. A prediction of the velocity field, $v^*$, is obtained by solving the momentum equation,

$$a_P v_P^* = \mathbf{H}(v^*) - \frac{1}{\rho} \nabla p^k. \quad (2.28)$$

In this step, the pressure gradient term is calculated using the pressure distribution, $p^k$, from the previous iteration or from an initial guess. Due to the non-linear nature of Navier-Stokes equations, it is necessary to under-relax the momentum equation in an implicit manner with the velocity under-relaxation factor $\alpha_U$.

2. The pressure equation,

$$\nabla \cdot \left( \frac{1}{\rho a_P} \nabla p^* \right) = \nabla \cdot \left( \frac{\mathbf{H}(v^*)}{a_P} \right), \quad (2.29)$$

is implicitly solved in order to obtain the new pressure distribution, $p^*$. 

3. The pressure solution is under-relaxed in order to take into account the velocity part of the error,

$$p^{k+1} = p^k + \alpha_p(p^* - p^k), \quad (2.30)$$

where $p^{k+1}$ is the approximation of the pressure field that will be used in the next momentum predictor, $p^k$ is the pressure field used in the momentum predictor and $\alpha_p$ is the pressure under-relaxation factor, $(0 < \alpha_p \leq 1)$. Then the explicit velocity correction is performed with this new value of the pressure field,

$$v_P^{k+1} = \frac{1}{a_P} \left[ \mathbf{H}(v^*) - \frac{1}{\rho} \nabla p^{k+1} \right]. \quad (2.31)$$

4. Finally the algorithm increments the iterative index $k$ and loops over the whole sequence until the residuals go down below a predefined criterion.
The usual recommended values of the under-relaxation factors $\alpha_p$ and $\alpha_U$ are 0.2 and 0.8 respectively. Actually these values depend on the Reynolds number. For Stokes problem, the simulations can converge up to twice faster with $\alpha_U = 0.9$.

### 2.7 Alternative approach to compute the flow in the pore-space

An alternative approach that gains in popularity lately for computing the single phase flow in the void space of a porous medium is the lattice Boltzmann method. It is a particle-based technique with collision models that solves the discrete Boltzmann equation instead of solving the Navier–Stokes equations. The averaged behaviour can be shown to approximate the governing Navier-Stokes equations. Detailed considerations of this technique are beyond the scope of this document.
Bibliography


