1 Introduction

Numerical simulation of flows with suspended particles is a task of great interest in Engineering. The applications involve rheology, coating and drying processes, fluidized suspensions and many others. The coupling between the particles' dynamics and the fluid's dynamics is necessary when we deal with particulate flows with moderate and high concentrations. The more particles are suspended in the liquid, the more their dynamics affect the flow. At low concentrations, uncoupled models can be used.

Previous works have proposed many techniques to solve this problem and they can be gathered in two types: the type that uses a moving mesh and the one with a fixed mesh.

The first type solves the Navier-Stokes equations in the fluid domain, as in the works of Hu et al (18, 17). The hydrodynamic forces that act on the particles are computed from the flow solution and applied on the particles' dynamics equations. This approach requires the mesh to wrap around the particles so that it covers only the fluid domain. It may be, however, computationally expensive as the mesh needs to be generated at each time step once the particles move. Besides, no regular meshes can be employed, as there are holes in the regions occupied by the particles.

The other type uses the Fictitious Domain Method, in which a fixed mesh is used. The mesh can be regular depending only on the domain's shape. The first work was proposed by Glowinsky et al (16). Other authors have improved this method such as (12).

Recently, in his PhD. thesis, Marcos Lage (21) introduced a modified Fictitious Domain Method to simulate suspended particles in incompressible flow in 2D. Our aim is to do so in three-dimensional space in order to get a more realistic description. As a first attempt in 3D space, we will use only one phase of fluid. So we do not take the capillarity force into account nor an interface between two immiscible fluids.

In this work, we present the simulation of Newtonian incompressible transient flows in three-dimensional space with and without immersed particles. By particle, we mean a very small rigid body. So, in the whole work, when the reader finds the word particle we will be referring to a spherical rigid body with its 6 degrees of freedom – 3 components of translational velocity and 3 of angular velocity.

The lid driven cavity problem is used to validate the code without particles. In order to enable the comparison between our results with those of Lage (21), the boundary conditions were adjusted to perform the simulation of two-dimensional flow with the three-dimensional code. This validation is done without particles, once the 2D particles are infinite cylinders in 3D and not spheres.

Figure 1.1 shows the flow domain with a 2D circular particle, which was studied in Lage's thesis (21). Defining the 3D axis as in figure 1.2, we have this particle as an infinite cylinder along the z axis (see Fig. 1.3(a)). However, this work simulates spheres in 3D space (Fig. 1.3(b)). It means that 2D results for circles do not match 3D results for spheres as the shapes of the particles are not the same. Despite this fact, the formulation presented in this work can be applied with arbitrary shapes. We just need, indeed, a function that tells whether a point lies inside the particle or not and a function to treat collision (in this work, we do not treat collision). Even so, for simplicity, we will solve the flow coupled only with spherical particles.

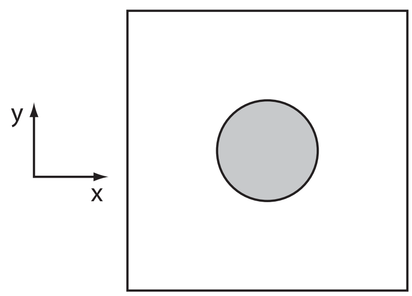


Figure 1.1: The 2D circular particle.

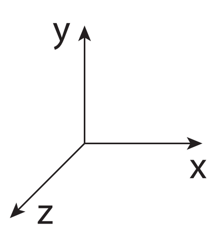
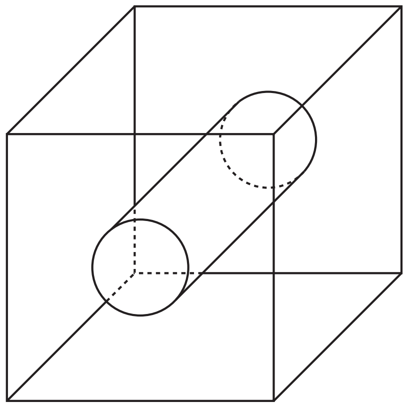
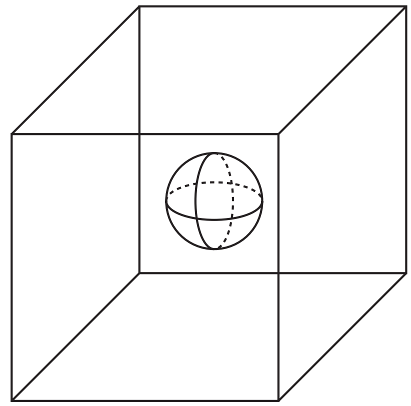


Figure 1.2: The 3D axis.

The formulation of the problem is based on the Momentum and Continuity equations (23, 6), as well as the Dynamics of Rigid Bodies. With these equations, the weak formulation is developed and solved by the finite elements' method.



1.3(a): 3D infinite cylinder



1.3(b): 3D sphere

Figure 1.3: Sketch of 2D and 3D particles.

1.1 Units of measurement

We adopted in this work the International System of Units (SI). In this entire work (code, files and interface of the software), the quantities are expressed in the units displayed in table 1.1.

Quantity	Unit	Symbol
length	meter	m
time	second	s
mass	kilogram	kg
velocity	meter per second	m/s
acceleration	meter per second squared	m/s^2
area	square meter	m^2
volume	cubic meter	m^3
force	Newton	N
pressure and stress	Pascal	Pa
dynamic viscosity	Pascal second	$Pa.s$
specific mass	kilogram per cubic meter	kg/m^3

Table 1.1: Units used for each quantity