Yade Technical Archive (2018)
Document Link
www.yade-dem.org

YADE 1D vertical VANS fluid resolution: numerical resolution details

Raphael Maurin[1,*]

[1]Institut de Mécanique des Fluides de Toulouse, Univ. Toulouse, Toulouse , France
[*]raphael.maurin@imft.fr

ABSTRACT

In the present note, the volume-averaged Navier-Stokes (VANS) 1D fluid resolution performed in YADE is detailed. Starting from unidirectional VANS equation, the closures considered are detailed in order to obtain the equation to solve. The equation is then discretized in time and space following an upwind staggered grid resolution scheme, and written as a simple matricial system to solve. The numerical scheme is then detailed at the boundary and the implementation in YADE is fully explicited.

Keywords: Yade; fluid-DEM coupling; 1D VANS; numerical implementation; turbulent bedload transport

1 Introduction

The goal of the present note is to detail the volume-averaged Navier-Stokes (VANS) 1D fluid resolution performed in YADE (16), and which can be coupled with the DEM part. The theoretical basis of the momentum balance equation solved and the practical details of the coupling with the DEM in YADE can be found respectively in refs (9) and (10). The coupling presented here has been validated with classical configuration and fine experiments in (11).

The fluid code has been adapted by Raphael Maurin (IMFT) (8; 12) from the two-phase continuous code of Julien Chauchat (LEGI) (1; 2; 15); translating it from fortran77 to c++ with the help of Julien Chauchat, and adapting the coupling with the DEM granular phase instead of the continuous one.

This document deals with the practical details of the fluid resolution, i.e. the closures adopted for the fluid momentum balance equation, its discretization, its matricial form and its resolution.

2 1D fluid momentum balance equation

2.1 Fluid equations

As shown previously in a document about the theoretical bases of volume-averaged fluid resolution (10), the 3D volume-averaged momentum balance for a fluid in interaction with a particle phase reads:

$$\rho^f \varepsilon \left(\frac{\partial \langle u_i \rangle^f}{\partial t} + \langle u_k \rangle^f \frac{\partial}{\partial x_k} \langle u_i \rangle^f \right) = \frac{\partial}{\partial x_k} \left(\sigma_{ik}^{eff} + \varepsilon R_{ik}^f \right) + \rho^f \varepsilon\, g_i - n \left\langle f_i^f \right\rangle^p, \tag{1}$$

Where ρ^f is the fluid density, $\varepsilon = 1 - \phi$ is the fluid volume fraction, u_i is the i-th component of the velocity, $\langle \bullet \rangle^f$ and $\langle \bullet \rangle^p$ denote the volume averaging with respect to the fluid and particle phase respectively, σ_{ij}^{eff} is the effective fluid stress tensor, R_{ij}^f is the Reynolds stress tensor, g_i is the i-th component of the acceleration of gravity, and $n \left\langle f_i^f \right\rangle^p$ is the volume-averaged fluid-particle momentum transfer associated with the hydrodynamic forces.

In the case of steady uniform bedload transport, the problem is unidirectional so that the fluid velocity depends only on the wall-normal component z and is aligned with the streamwise direction (see figure 1), $\langle \mathbf{u} \rangle^f (\mathbf{x}) = \langle u_x \rangle^f (z) \mathbf{e_x}$. Therefore, equations (1) simplifies into a 1D vertical momentum balance (12):

$$\rho^f \varepsilon \frac{\partial \langle u_x \rangle^f}{\partial t} = \frac{\partial}{\partial z}\left(\sigma_{xz}^{eff} + \varepsilon R_{xz}^f\right) + \rho^f \varepsilon\, g_x - n\left\langle f_x^f \right\rangle^p \tag{2}$$

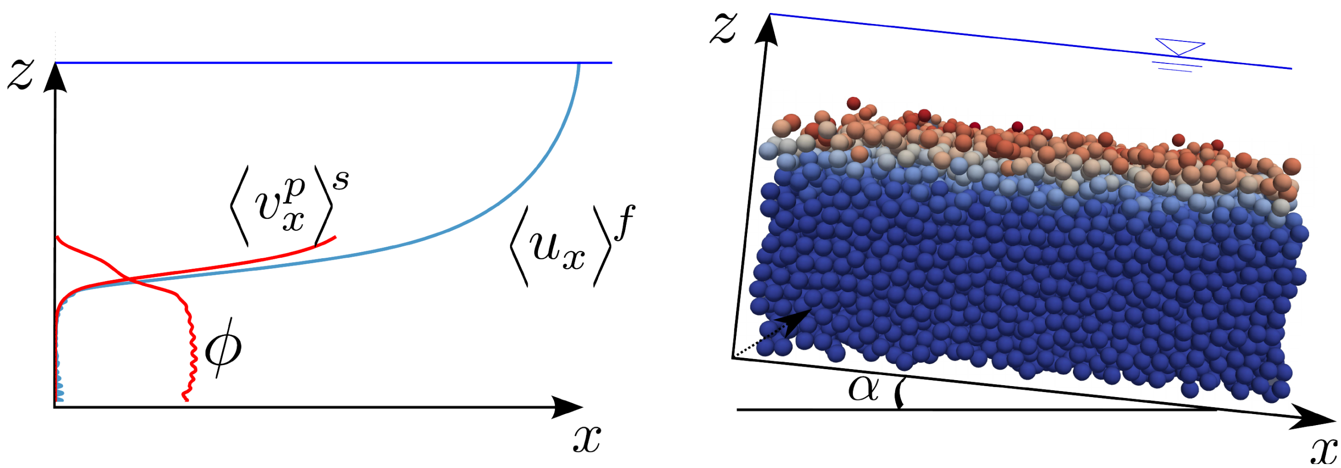

Figure 1. Scheme of the bedload transport configuration considered and its equivalent average unidirectional picture with typical fluid velocity $\langle \mathbf{u}^f \rangle = \left\langle u_x^f \right\rangle(z)\, \mathbf{e_x}$, solid volume fraction ϕ, and solid velocity $\langle \mathbf{v}^p \rangle^s = \left\langle v_x^p \right\rangle^s(z)\, \mathbf{e_x}$ depth profiles. After (13; 14).

2.2 Closures adopted in YADE

Considering the effective fluid stress tensor to be characterized by an effective viscosity ν^e, it reads:

$$\sigma_{xz}^{eff} = \rho^f \nu^e \frac{\partial \langle u_x \rangle}{\partial z}, \tag{3}$$

Where $\langle u_x \rangle = \phi \langle u_x \rangle^s + \varepsilon \langle u_x \rangle^f$ as explained in (5) and recalled in section 2.1.2 of (10).The effective viscosity adopted can be taken either as the Einstein effective viscosity $\nu^e = \nu(1+2.5\phi)$, or as clear fluid $\nu^e = \nu$ (no effect of the particles on the fluid rheology).

The Reynolds stress tensor is closed using an eddy viscosity formulation with a mixing length approach:

$$R_{xz}^f = \rho^f\ \nu^t \frac{\partial \langle u_x \rangle^f}{\partial z} \ \text{ with }\ \nu^t = l_m^2 \left| \frac{\partial \langle u_x \rangle^f}{\partial z} \right|, \tag{4}$$

in which the mixing length l_m formulation proposed by ref. (7) is used:

$$l_m(z) = \kappa \int_0^z \frac{\phi^{max} - \phi(\zeta)}{\phi^{max}}\, d\zeta, \tag{5}$$

where $\kappa = 0.41$ represents the von Karman constant.

Lastly, the fluid-particle interaction term is restricted to drag and buoyancy, $\mathbf{f}^f = \mathbf{f}_D^f + \mathbf{f}_b^f$. Following refs (6) and (14), the buoyancy $f_{b,i}^f$ is taken as the generalized Archimedes force:

$$n\left\langle f_i^f \right\rangle^p = n\left\langle f_{b,i}^f \right\rangle^p + n\left\langle f_{D,i}^f \right\rangle^p = \phi \frac{\partial \sigma_{ij}^{eff}}{\partial x_j} + n\left\langle f_{D,i}^f \right\rangle^p. \tag{6}$$

As a consequence, equation (1) is modified:

$$\varepsilon \rho^f \frac{\partial \langle u_x \rangle^f}{\partial t} = (1-\phi)\frac{\partial \sigma_{xz}^{eff}}{\partial z} + \frac{\partial (\varepsilon R_{xz}^f)}{\partial z} + \varepsilon \rho^f g\ \sin\alpha - n\left\langle f_{D,i}^f \right\rangle^p. \tag{7}$$

The drag term is evaluated in the DEM simulations from averaging of the drag force applied to each particle. The average term is therefore expressed as[1]:

$$n\langle f_{D,x}\rangle = \beta\ \left(\langle u_x\rangle^f - \langle v_x^p\rangle^s\right), \tag{8}$$

with β evaluated from the DEM, as well as the average granular velocity, $\langle v_x^p\rangle^s$.

These different closures, altogether reported in equation (1) lead to the following 1D fluid equation to solve:

$$\varepsilon\rho^f\frac{\partial\langle u_x\rangle^f}{\partial t} = \rho^f\varepsilon\frac{\partial}{\partial z}\left(\nu^e\frac{\partial}{\partial z}\left[\phi\langle u_x\rangle^s + \varepsilon\langle u_x\rangle^f\right]\right) + \rho^f\frac{\partial}{\partial z}\left(\varepsilon\nu^t\frac{\partial\langle u_x\rangle^f}{\partial z}\right) + \varepsilon\rho^f g\ \sin\alpha - \beta\left(\langle u_x\rangle^f - \langle v_x^p\rangle^s\right) \tag{9}$$

3 Numerical resolution

This 1D equation is to be solved on a vertical grid at different vertical points. The fluid resolution is based on an implicit first order Euler time scheme, together with an upwind scheme on a staggered grid. The scheme presented in figure 2 gives an overview of the variables definitions and position. The regular mesh defines nodes in between which the equations are solved. At these nodes, the scalar variables such as the solid volume fraction are defined. The velocities are defined in between to allow for a better precision of the numerical scheme, so that the spatial derivatives of the velocity are defined at the node and the second spatial derivative are defined in between the node (staggered grid). The fluid momentum balance equation (9) is expressed in terms of velocity and second derivative of the velocity so that it is solved at the velocity nodes, i.e. at position i+1/2.

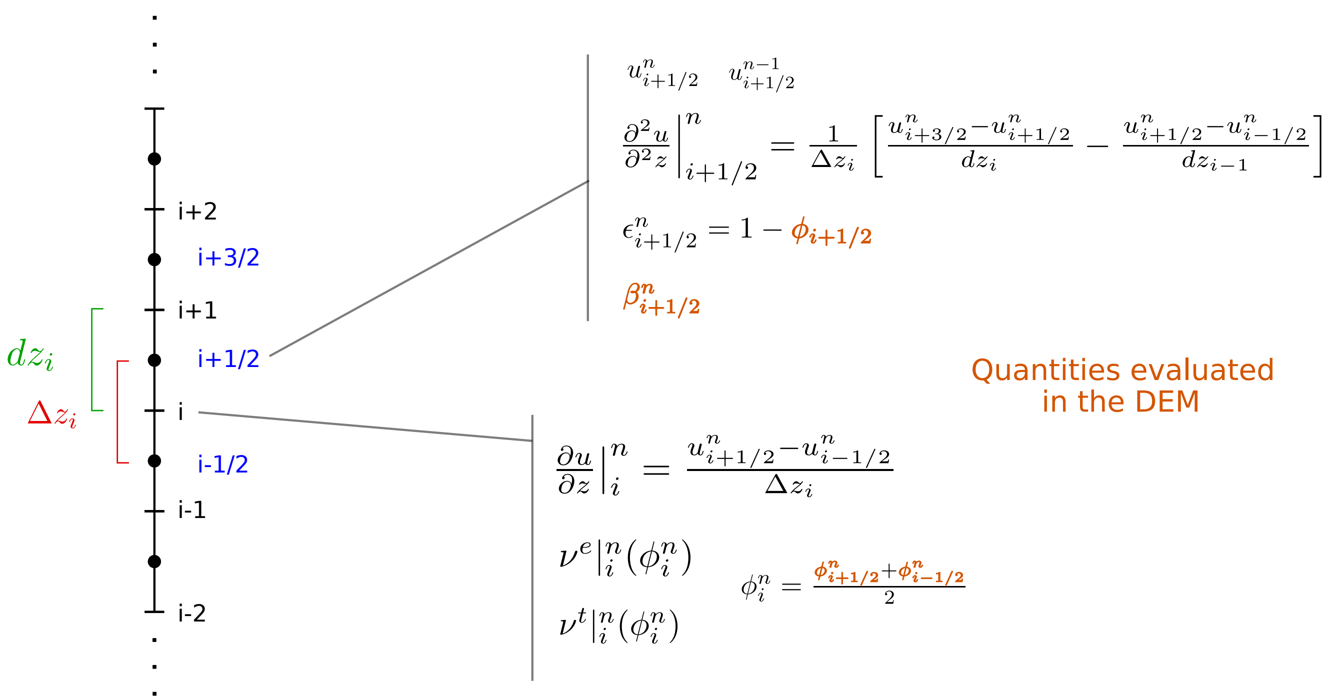

Figure 2. Schematical picture of the numerical fluid resolution and variables definition with a regular mesh. All the definitions still holds for a mesh with variable spatial step.

3.1 Time and space discretization

The time derivative can then be written as (implicit first order Euler time scheme):

$$\left.\frac{\partial u}{\partial t}\right|^n = \frac{u^n - u^{n-1}}{\Delta t}, \tag{10}$$

[1]Therefore, the consequence of this formulation is that beta is explicit while the rest is implicit. The longer the fluid resolution without updating the granular phase, the larger the associated error with this assumption.

where for simplicity the averaged streamwise fluid velocity is written u, and u^n denotes the (unknown) velocity at the present time step, while u^{n-1} denotes the (known) velocity at the previous time step. The equation should also be discretized spatially, and the spatial derivative reads for an upwind scheme on a staggered grid:

$$\left.\frac{\partial u}{\partial z}\right|_i^n = \frac{u^n_{i+1/2} - u^n_{i-1/2}}{\Delta z_i}. \tag{11}$$

Where Δz_i is the space between the two velocity $u^n_{i+1/2}$ and $u^n_{i/2}$. With these expressions, equation (9) expressed at the velocity node $i+1/2$ reads:

$$\rho^f \varepsilon^n_{i+1/2} \left.\frac{\partial u}{\partial t}\right|^n_{i+1/2} = \rho^f \varepsilon^n_{i+1/2} \frac{\partial}{\partial z}\left(\nu^e \frac{\partial\left[\varepsilon u + \phi v\right]}{\partial z}\right)\Bigg|^n_{i+1/2} + \rho^f \frac{\partial}{\partial z}\left(\varepsilon \nu^t \frac{\partial u}{\partial z}\right)\Bigg|^n_{i+1/2}$$
$$+ \varepsilon^n_{i+1/2}\, \rho^f g\, \sin\alpha - \beta^n_{i+1/2}\left(u^n_{i+1/2} - v^n_{i+1/2}\right) \tag{12}$$

The first term on the left hand side can be expanded as:

$$\rho^f \varepsilon^n_{i+1/2} \left.\frac{\partial u}{\partial t}\right|^n_{i+1/2} = \rho^f \varepsilon^n_{i+1/2} \frac{u^n_{i+1/2} - u^{n-1}_{i+1/2}}{\Delta t} \tag{13}$$

For the first term on the right hand side, the equations are solved at the velocity node so that the full derived term should be evaluated at $i+1/2$. This way, it can be expanded (upwind scheme):

$$\frac{\partial}{\partial z}\left(\nu^e \frac{\partial(\varepsilon u)}{\partial z}\right)\Bigg|^n_{i+1/2} = \frac{1}{dz_i}\left(\nu^e \left.\frac{\partial(\varepsilon u)}{\partial z}\right|^n_{i+1} - \nu^e \left.\frac{\partial(\varepsilon u)}{\partial z}\right|^n_i\right)$$
$$= \frac{1}{dz_i}\left[\nu^e|^n_{i+1} \frac{\varepsilon^n_{i+3/2}u^n_{i+3/2} - \varepsilon^n_{i+1/2}u^n_{i+1/2}}{\Delta z_{i+1}} - \nu^e|^n_i \frac{\varepsilon^n_{i+1/2}u^n_{i+1/2} - \varepsilon^n_{i-1/2}u^n_{i-1/2}}{\Delta z_i}\right] \tag{14}$$

where dz_i is the space between node i and $i+1$ defined by the adopted mesh, and Δz_i is the space between the velocity positions defined by (see figure 2):

$$\Delta z_i = \frac{dz_i + dz_{i-1}}{2}. \tag{15}$$

This distinction might seem secondary when considering a regular grid. However, it will appear to play a (non-negligible) role when considering the boundaries. Doing the same with ϕv, we obtain the second part of the expression:

$$\frac{\partial}{\partial z}\left(\nu^e \frac{\partial(\phi v)}{\partial z}\right)\Bigg|^n_{i+1/2} = \frac{1}{dz_i}\left(\nu^e \left.\frac{\partial(\phi v)}{\partial z}\right|^n_{i+1} - \nu^e \left.\frac{\partial(\phi v)}{\partial z}\right|^n_i\right)$$
$$= \frac{1}{dz_i}\left[\nu^e|^n_{i+1} \frac{\phi^n_{i+3/2}v^n_{i+3/2} - \phi^n_{i+1/2}v^n_{i+1/2}}{\Delta z_{i+1}} - \nu^e|^n_i \frac{\phi^n_{i+1/2}v^n_{i+1/2} - \phi^n_{i-1/2}v^n_{i-1/2}}{\Delta z_i}\right] \tag{16}$$

Considering a similar logic we obtain for the second term on the right hand side of equation (12):

$$\frac{\partial}{\partial z}\left(\varepsilon \nu^t \frac{\partial u}{\partial z}\right)\Bigg|^n_{i+1/2} = \frac{1}{dz_i}\left(\varepsilon \nu^t \left.\frac{\partial u}{\partial z}\right|^n_{i+1} - \varepsilon \nu^t \left.\frac{\partial u}{\partial z}\right|^n_i\right)$$
$$= \frac{1}{dz_i}\left[\varepsilon^n_{i+1}\ \nu^t|^n_{i+1} \frac{u^n_{i+3/2} - u^n_{i+1/2}}{\Delta z_{i+1}} - \varepsilon^n_i \nu^t|^n_i \frac{u^n_{i+1/2} - u^n_{i-1/2}}{\Delta z_i}\right] \tag{17}$$

With this expansion and considering a similar logic for the second term on the right hand side, equation (12) can be rewritten:

$$
\begin{aligned}
&\rho^f \varepsilon^n_{i+1/2} \frac{u^n_{i+1/2}-u^{n-1}_{i+1/2}}{\Delta t} = \\
&\frac{\rho^f \varepsilon^n_{i+1/2}}{dz_i}\left[\nu^e|^n_{i+1} \frac{\varepsilon^n_{i+3/2}u^n_{i+3/2}-\varepsilon^n_{i+1/2}u^n_{i+1/2}}{\Delta z_{i+1}} - \nu^e|^n_i \frac{\varepsilon^n_{i+1/2}u^n_{i+1/2}-\varepsilon^n_{i-1/2}u^n_{i-1/2}}{\Delta z_i}\right] \\
&+\frac{\rho^f \varepsilon^n_{i+1/2}}{dz_i}\left[\nu^e|^n_{i+1} \frac{\phi^n_{i+3/2}v^n_{i+3/2}-\phi^n_{i+1/2}v^n_{i+1/2}}{\Delta z_{i+1}} - \nu^e|^n_i \frac{\phi^n_{i+1/2}v^n_{i+1/2}-\phi^n_{i-1/2}v^n_{i-1/2}}{\Delta z_i}\right] \\
&+\frac{\rho^f \varepsilon^n_{i+1/2}}{dz_i}+\frac{\rho^f}{dz_i}\left[\varepsilon^n_{i+1}\ \nu^t\big|^n_{i+1} \frac{u^n_{i+3/2}-u^n_{i+1/2}}{\Delta z_{i+1}} - \varepsilon^n_i \nu^t\big|^n_i \frac{u^n_{i+1/2}-u^n_{i-1/2}}{\Delta z_i}\right] \\
&\qquad\qquad\qquad\qquad + \varepsilon^n_{i+1/2}\ \rho^f g\ \sin\alpha - \beta^n_{i+1/2}\left(u^n_{i+1/2}-v^n_{i+1/2}\right)
\end{aligned} \tag{18}
$$

The last thing is now to express the effective and turbulent viscosities in this frame. Recalling the formulation of the turbulent viscosity (equation 4) and assuming a linearly dependant Einstein's effective viscosity (4) we get:

$$
\nu^e|^n_i = \nu^f(1+2.5\phi)\big|^n_i = \nu^f(1+2.5\phi^n_i) \tag{19}
$$

$$
\nu^t|^n_i = \ l_m^2 \frac{\partial u}{\partial z}\bigg|^n_i = (l_m|^n_i)^2 \frac{u^n_{i+1/2}-u^n_{i-1/2}}{\Delta z_i} \tag{20}
$$

with

$$
l_m|^n_i = \kappa \sum_{j=0}^{i} \frac{\phi^{max}-\phi^n_{j-1/2}}{\phi^{max}} dz_{j-1} \tag{21}
$$

and $\phi^n_{j-1/2}$ the solid volume fraction defined between node $j-1$ and j, and dz_{j-1} the space between node $j-1$ and j.

3.2 Matricial system to solve

From equations (18) to (21), we can form a matricial system to solve the new fluid velocity profile. Indeed, gathering first the element as a function of the fluid velocity components, it gives:

$$
\begin{aligned}
&\frac{\rho^f}{\Delta t}\left(u^n_{i+1/2}\ \varepsilon^n_{i+1/2} - u^{n-1}_{i+1/2}\ \varepsilon^n_{i+1/2}\right) = u^n_{i+3/2}\ \left(\frac{\rho^f \varepsilon^n_{i+1/2}}{dz_i}\ \frac{\nu^e|^n_{i+1}\ \varepsilon^n_{i+3/2}}{\Delta z_{i+1}} + \frac{\rho^f}{dz_i}\ \ \frac{\varepsilon^n_{i+1}\nu^t|^n_{i+1}}{\Delta z_{i+1}}\right) \\
&+u^n_{i+1/2}\ \left(-\frac{\rho^f \varepsilon^n_{i+1/2}}{dz_i}\ \frac{\nu^e|^n_{i+1}\ \varepsilon^n_{i+1/2}}{\Delta z_{i+1}} - \frac{\rho^f \varepsilon^n_{i+1/2}}{dz_i}\ \frac{\nu^e|^n_i\ \varepsilon^n_{i+1/2}}{\Delta z_i} - \frac{\rho^f}{dz_i}\ \frac{\varepsilon^n_{i+1}\nu^t|^n_{i+1}}{\Delta z_{i+1}} - \frac{\rho^f}{dz_i}\ \frac{\varepsilon^n_i \nu^t|^n_i}{\Delta z_i} - \beta^n_{i+1/2}\right) \\
&+u^n_{i-1/2}\ \left(\frac{\rho^f \varepsilon^n_{i+1/2}}{dz_i}\ \frac{\nu^e|^n_i\ \varepsilon^n_{i-1/2}}{\Delta z_i} + \frac{\rho^f}{dz_i}\ \frac{\varepsilon^n_i \nu^t|^n_i}{\Delta z_i}\right) +\ \varepsilon^n_{i+1/2}\ \rho^f g\ \sin\alpha +\ \beta^n_{i+1/2}\ v^n_{i+1/2} \\
&\qquad\qquad\qquad + \frac{\rho^f \varepsilon^n_{i+1/2}}{dz_i}\left[\nu^e|^n_{i+1} \frac{\phi^n_{i+3/2}v^n_{i+3/2}-\phi^n_{i+1/2}v^n_{i+1/2}}{\Delta z_{i+1}} - \nu^e|^n_i \frac{\phi^n_{i+1/2}v^n_{i+1/2}-\phi^n_{i-1/2}v^n_{i-1/2}}{\Delta z_i}\right]
\end{aligned} \tag{22}
$$

Multiplying by Δt, dividing by ρ^f, moving all the terms at time n to the left and the rest to the right, we obtain:

$$
\begin{aligned}
& u^n_{i-1/2}\left(-\frac{\Delta t\ \varepsilon^n_{i+1/2}}{dz_i}\ \frac{\nu^e|^n_i\ \varepsilon^n_{i-1/2}}{\Delta z_i}-\frac{\Delta t}{dz_i}\ \frac{\varepsilon^n_i \nu^t|^n_i}{\Delta z_i}\right)\\
&+u^n_{i+1/2}\left(\frac{\Delta t\ \varepsilon^n_{i+1/2}}{dz_i}\ \frac{\nu^e|^n_{i+1}\ \varepsilon^n_{i+1/2}}{\Delta z_{i+1}}+\frac{\Delta t\ \varepsilon^n_{i+1/2}}{dz_i}\ \frac{\nu^e|^n_i\ \varepsilon^n_{i+1/2}}{\Delta z_i}+\frac{1}{\Delta t\ dz_i}\ \frac{\varepsilon^n_{i+1}\nu^t|^n_{i+1}}{\Delta z_{i+1}}+\frac{\Delta t}{dz_i}\ \frac{\varepsilon^n_i\nu^t|^n_i}{\Delta z_i}+\varepsilon^n_{i+1/2}+\frac{\Delta t}{\rho^f}\ \beta^n_{i+1/2}\right)\\
&+u^n_{i+3/2}\left(-\frac{\Delta t\ \varepsilon^n_{i+1/2}}{dz_i}\ \frac{\nu^e|^n_{i+1}\ \varepsilon^n_{i+3/2}}{\Delta z_{i+1}}-\frac{\Delta t}{dz_i}\ \frac{\varepsilon^n_{i+1}\nu^t|^n_{i+1}}{\Delta z_{i+1}}\right)\\
&=u^{n-1}_{i+1/2}\ \varepsilon^n_{i+1/2}+\ \varepsilon^n_{i+1/2}\Delta t\ g\ \sin\alpha+\ \frac{\Delta t}{\rho^f}\beta^n_{i+1/2}\ v^n_{i+1/2}\\
&\qquad+\frac{\Delta t\ \varepsilon^n_{i+1/2}}{dz_i}\left[\nu^e|^n_{i+1}\frac{\phi^n_{i+3/2}v^n_{i+3/2}-\phi^n_{i+1/2}v^n_{i+1/2}}{\Delta z_{i+1}}-\nu^e|^n_i\frac{\phi^n_{i+1/2}v^n_{i+1/2}-\phi^n_{i-1/2}v^n_{i-1/2}}{\Delta z_i}\right]
\end{aligned}
\tag{23}
$$

And in order to respect the formulation at the boundaries (see below), we can write it as:

$$
a[i+1]\ u^n_{i-1/2}+b[i+1]\ u^n_{i+1/2}+c[i+1]\ u^n_{i+3/2}=s[i+1] \tag{24}
$$

with

$$
a[i+1]=-\frac{\Delta t\ \varepsilon^n_{i+1/2}}{dz_i}\ \frac{\nu^e|^n_i\ \varepsilon^n_{i-1/2}}{\Delta z_i}-\frac{\Delta t}{dz_i}\ \frac{\varepsilon^n_i\nu^t|^n_i}{\Delta z_i} \tag{25}
$$

$$
b[i+1]=\frac{\Delta t\ \varepsilon^n_{i+1/2}}{dz_i}\ \frac{\nu^e|^n_{i+1}\ \varepsilon^n_{i+1/2}}{\Delta z_{i+1}}+\frac{\Delta t\ \varepsilon^n_{i+1/2}}{dz_i}\ \frac{\nu^e|^n_i\ \varepsilon^n_{i+1/2}}{\Delta z_i}+\frac{\Delta t}{dz_i}\ \frac{\varepsilon^n_{i+1}\nu^t|^n_{i+1}}{\Delta z_{i+1}}+\frac{\Delta t}{dz_i}\ \frac{\varepsilon^n_i\nu^t|^n_i}{\Delta z_i}+\varepsilon^n_{i+1/2}+\frac{\Delta t}{\rho^f}\ \beta^n_{i+1/2} \tag{26}
$$

$$
c[i+1]=-\frac{\Delta t\ \varepsilon^n_{i+1/2}}{dz_i}\ \frac{\nu^e|^n_{i+1}\ \varepsilon^n_{i+3/2}}{\Delta z_{i+1}}-\frac{\Delta t}{dz_i}\ \frac{\varepsilon^n_{i+1}\nu^t|^n_{i+1}}{\Delta z_{i+1}} \tag{27}
$$

$$
\begin{aligned}
s[i+1]&=u^{n-1}_{i+1/2}\ \varepsilon^n_{i+1/2}+\ \varepsilon^n_{i+1/2}\Delta t\ g\ \sin\alpha+\ \frac{\Delta t}{\rho^f}\beta^n_{i+1/2}\ v^n_{i+1/2}\\
&\qquad+\frac{\Delta t\ \varepsilon^n_{i+1/2}}{dz_i}\left[\nu^e|^n_{i+1}\frac{\phi^n_{i+3/2}v^n_{i+3/2}-\phi^n_{i+1/2}v^n_{i+1/2}}{\Delta z_{i+1}}-\nu^e|^n_i\frac{\phi^n_{i+1/2}v^n_{i+1/2}-\phi^n_{i-1/2}v^n_{i-1/2}}{\Delta z_i}\right]
\end{aligned}
\tag{28}
$$

This system can be written in a matricial form

$$
MU^n=S, \tag{29}
$$

Where M is a $(ndimz+1)$ x $(ndimz+1)$ tri-diagonal matrix formed by:

$$
M=\begin{bmatrix}
a[0] & b[0] & c[0] & 0 & 0 & 0 & 0 & ... & 0\\
0 & a[1] & b[1] & c[1] & 0 & 0 & 0 & ... & 0\\
... & ... & ... & ... & ... & ... & ... & ... & ...\\
0 & ... & 0 & a[i] & b[i] & c[i] & 0 & ... & 0\\
... & ... & ... & ... & ... & ... & ... & ... & ...\\
0 & ... & 0 & 0 & 0 & 0 & a[ndimz] & b[ndimz] & c[ndimz]
\end{bmatrix} \tag{30}
$$

and U^n the (unknown) fluid velocity vector of size $ndimz+1$

$$
\begin{aligned}
U^n&=(U^n_0,U^n_1,U^n_2,...,U^n_{i-1},U^n_i,U^n_{i+1},...,U^n_{ndimz-1},U^n_{ndimz})\\
&=(u^n_0,u^n_{1/2},u^n_{1+1/2},...u^n_{i-1/2},u^n_{i+1/2},u^n_{i+3/2},...u^n_{ndimz-1/2},u^n_{ndimz})
\end{aligned}
\tag{31}
$$

$$S = (S_0, S_1, S_2, ..., S_{i-1}, S_i, S_{i+1}, ..., S_{ndimz-1}, S_{ndimz})$$
$$= (s[0], s[1], s[2], ..., s[i-1], s[i], s[i+1], ..., s[ndimz-1], s[ndimz]) \quad (32)$$

Where $ndimz+1$ is the number of mesh points, and 0 and $ndimz$ denotes the bottom and top boundary conditions respectively. From equation (29), we can deduce the fluid velocity by inverting the tri-diagonal matrix M and evaluating:

$$U^n = M^{-1}S. \quad (33)$$

The inversion of a tri-diagonal matrix is rather simple and is done in YADE with a double-sweep algorithm (3).

3.3 Boundaries and boundary conditions

The velocities are defined in between the nodes; however, at the boundaries, the velocity needs to be defined at the nodes to prescribe boundary conditions (see figure 3). Therefore as can be seen on figure 3 for a regular mesh, the formulation of Δz_i is changing for 0 and ndimz -1:

$$\Delta z_0 = \frac{dz_0}{2}, \quad (34)$$

$$\Delta z_{ndimz-1} = \frac{dz_{ndimz-2}}{2}. \quad (35)$$

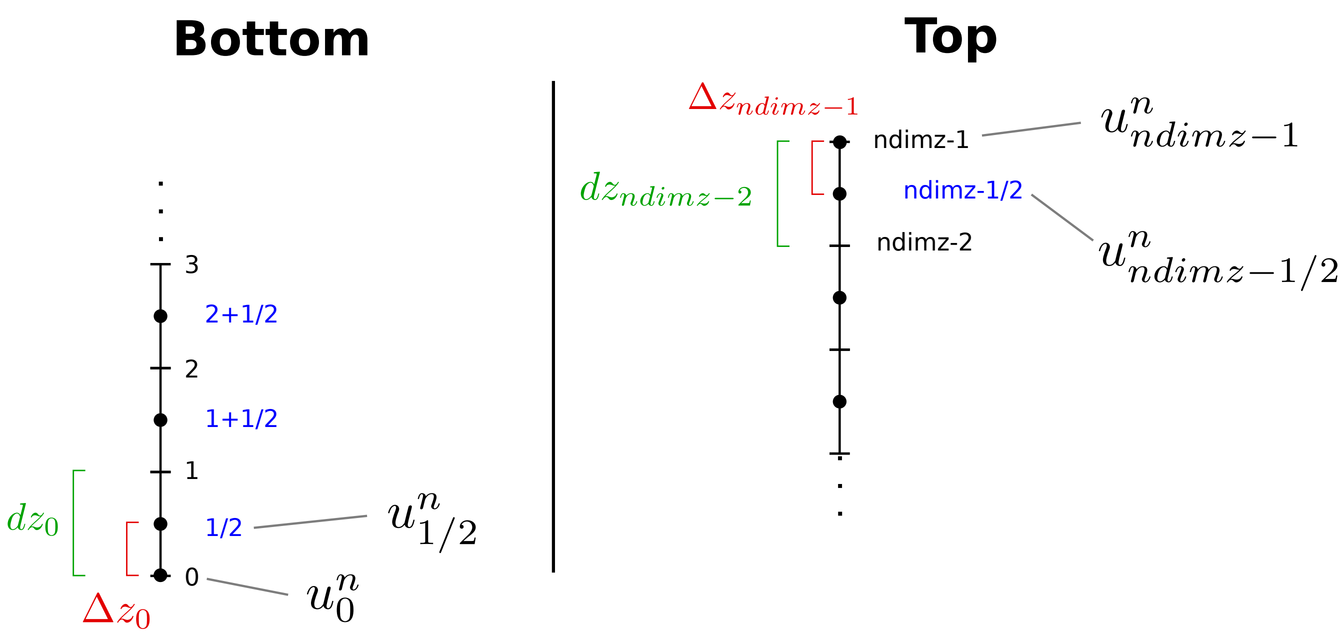

Figure 3. Schematical picture of the numerical fluid resolution and variables definition at the boundaries with a regular mesh. All the definitions still holds for a mesh with variable spatial step.

In addition, at the bottom, the matrical equation becomes:

$$a[0]\ U^n_{-3/2} + b[0]\ U^n_{-1/2} + c[0]\ U^n_{1/2} = s[0], \quad (36)$$

which can be rewritten by identifying $U^n_{-1/2} = U^n_0$:

$$a[0]\ U^n_{-3/2} + b[0]\ U^n_0 + c[0]\ U^n_{1/2} = s[0], \quad (37)$$

where $a[0]$ should be kept to zero in order to remove the non-existing term in $-3/2$. Similarly, we have at the top:

$$a[ndimz]\ U^n_{ndimz-1/2} + b[ndimz]\ U^n_{ndimz} + c[ndimz]\ U^n_{ndimz+3/2} = s[ndimz], \quad (38)$$

with the condition $c[ndimz] = 0$. In order to impose either a velocity (Dirichlet, fixed) or zero velocity-gradient (Neumann, free-surface) at the boundaries , we need to have:

$$U_0^n = U_{imposed}^{bot} \text{ or } U_0^n = U_{1/2}^n \text{ at the bottom,} \tag{39}$$

$$U_{ndimz-1}^n = U_{imposed}^{top} \text{ or } U_{ndimz-1}^n = U_{ndimz-2}^n \text{ at the top.} \tag{40}$$

In terms of matricial form, this means for the bottom

$$a[0] = 0 \text{ , } b[0] = 1 \text{ , } c[0] = 0 \text{ and } s[0] = U_{imposed}^{bot}, \tag{41}$$

for an imposed velocity, or

$$a[0] = 0 \text{ , } b[0] = 1 \text{ , } c[0] = -1 \text{ and } s[0] = 0, \tag{42}$$

for a zero velocity gradient. Similarly, we can impose a velocity at the top by setting

$$a[ndimz] = 0 \text{ , } b[ndimz] = 1 \text{ , } c[ndimz] = 0 \text{ and } s[ndimz] = U_{imposed}^{top}, \tag{43}$$

or impose the velocity gradient to be zero by setting

$$a[ndimz] = 1 \text{ , } b[ndimz] = -1 \text{ , } c[ndimz] = 0 \text{ and } s[ndimz] = 0. \tag{44}$$

Application example Commonly, we use a no-slip boundary condition at the bottom, and a free-surface one at the top. In this case, let us express the equations at the top and at the bottom, in order to image clearly the resolution. At the bottom, we have for $\mathbf{i} = \mathbf{0}$:

$$a[0]U_{-3/2}^n + b[0]U_0^n + c[0]U_{1/2}^n = s[0] \tag{45}$$

with

$$a[0] = 0 \text{ , } b[0] = 1 \text{ , } c[0] = 0 \text{ and } s[0] = U_{imposed}^{bot}, \tag{46}$$

For $\mathbf{i} = \mathbf{1}$,

$$a[1]U_0^n + b[1]U_{1/2}^n + c[1]U_{3/2}^n = s[1], \tag{47}$$

with $a[1]$, $b[1]$, $c[1]$, $s[1]$ given by equation 25-28 with $i = 0$, reading :

$$a[1] = -\frac{\Delta t\ \varepsilon_{1/2}^n}{dz_0}\ \frac{\nu^e|_0^n\ \varepsilon_0^n}{\Delta z_0} - \frac{\Delta t}{dz_0}\ \frac{\varepsilon_0^n \nu^t|_0^n}{\Delta z_0} \tag{48}$$

$$b[1] = \frac{\Delta t\ \varepsilon_{1/2}^n}{dz_0}\ \frac{\nu^e|_1^n\ \varepsilon_{1/2}^n}{\Delta z_1} + \frac{\Delta t\ \varepsilon_{1/2}^n}{dz_0}\ \frac{\nu^e|_0^n\ \varepsilon_{1/2}^n}{\Delta z_0} + \frac{\Delta t}{dz_0}\ \frac{\varepsilon_1^n \nu^t|_1^n}{\Delta z_1} + \frac{\Delta t}{dz_0}\ \frac{\varepsilon_0^n \nu^t|_0^n}{\Delta z_0} + \varepsilon_{1/2}^n + \frac{\Delta t}{\rho^f}\ \beta_{1/2}^n \tag{49}$$

$$c[1] = -\frac{\Delta t\ \varepsilon_{1/2}^n}{dz_0}\ \frac{\nu^e|_1^n \varepsilon_{3/2}^n}{\Delta z_1} - \frac{\Delta t}{dz_0}\ \frac{\varepsilon_1^n \nu^t|_1^n}{\Delta z_1} \tag{50}$$

$$s[1] = u_{1/2}^{n-1}\ \varepsilon_{1/2}^n +\ \varepsilon_{1/2}^n \Delta t\ g\ \sin\alpha + \frac{\Delta t}{\rho^f}\beta_{1/2}^n\ v_{1/2}^n + \frac{\rho^f \varepsilon_{1/2}^n}{dz_0}\left[\nu^e|_1^n \frac{\phi_{3/2}^n v_{3/2}^n - \phi_{1/2}^n v_{1/2}^n}{\Delta z_1} - \nu^e|_0^n \frac{\phi_{1/2}^n v_{1/2}^n - \phi_0^n v_0^n}{\Delta z_0}\right] \tag{51}$$

In which the value of Δz_0 should be taken as $dz_0/2$, and the evaluation of ϕ_0^n at the node 0 should be made in order to determine ε_0^n and $\nu^e|_0^n$. In the latter two cases, ϕ_0^n is taken equal to $\phi_{1/2}^n$, and all the expressions can be evaluated. At step $i = 2$, there are no more differences with respect to the classical evaluation of the matrices, as no terms in $_0$ appears. At the top of the sample,

the last equation is performed on node $ndimz-1$, i.e.
for $\mathbf{i} = \mathbf{ndimz} - \mathbf{1}$:

$$a[ndimz]\ u^n_{ndimz-3/2} + b[ndimz]\ u^n_{ndimz-1/2} + c[i+1]\ u^n_{ndimz+1/2} = s[ndimz], \tag{52}$$

i.e.,

$$a[ndimz]\ u^n_{ndimz-3/2} + b[ndimz]\ u^n_{ndimz-1} + c[ndimz]\ u^n_{ndimz+1/2} = s[ndimz]. \tag{53}$$

where we impose a free-surface by setting $a[ndimz] = 1$, $b[ndimz] = -1$, $c[ndimz] = 0$, $s[ndimz] = 0$.

For $\mathbf{i} = \mathbf{ndimz} - \mathbf{2}$, we have then:

$$a[ndimz-1]\ u^n_{ndimz-5/2} + b[ndimz-1]\ u^n_{ndimz-3/2} + c[ndimz-1]\ u^n_{ndimz-1} = s[ndimz-1], \tag{54}$$

with

$$a[ndimz-1] = -\frac{\Delta t\ \varepsilon^n_{ndimz-3/2}}{dz_{ndimz-2}}\ \frac{\nu^e|^n_{ndimz-2}\ \varepsilon^n_{ndimz-5/2}}{\Delta z_{ndimz-2}} - \frac{\Delta t}{dz_{ndimz-2}}\ \frac{\varepsilon^n_{ndimz-2}\nu^t|^n_{ndimz-2}}{\Delta z_{ndimz-2}} \tag{55}$$

$$b[ndimz-1] = \frac{\Delta t\ \varepsilon^n_{ndimz-3/2}}{dz_{ndimz-2}}\ \frac{\nu^e|^n_{ndimz-1}\ \varepsilon^n_{ndimz-3/2}}{\Delta z_{ndimz-1}} + \frac{\Delta t\ \varepsilon^n_{ndimz-3/2}}{dz_{ndimz-2}}\ \frac{\nu^e|^n_{ndimz-2}\ \varepsilon^n_{ndimz-3/2}}{\Delta z_i} + \frac{\Delta t}{dz_{ndimz-2}}\ \frac{\varepsilon^n_{ndimz-1}\nu^t|^n_{ndimz-1}}{\Delta z_{ndimz-1}} + \frac{\Delta t}{dz_{ndimz-2}}\ \frac{\varepsilon^n_i \nu^t|^n_i}{\Delta z_i} + \varepsilon^n_{ndimz-3/2} + \frac{\Delta t}{\rho^f}\ \beta^n_{ndimz-3/2} \tag{56}$$

$$c[ndimz-1] = -\frac{\Delta t\ \varepsilon^n_{ndimz-3/2}}{dz_{ndimz-2}}\ \frac{\nu^e|^n_{ndimz-1}\ \varepsilon^n_{ndimz-1}}{\Delta z_{ndimz-1}} - \frac{\Delta t}{dz_{ndimz-2}}\ \frac{\varepsilon^n_{ndimz-1}\nu^t|^n_{ndimz-1}}{\Delta z_{ndimz-1}} \tag{57}$$

$$s[ndimz-1] = u^{n-1}_{ndimz-3/2}\ \varepsilon^n_{ndimz-3/2} + \ \varepsilon^n_{ndimz-3/2}\Delta t\ g\ \sin\alpha + \frac{\Delta t}{\rho^f}\beta^n_{ndimz-3/2}\ v^n_{ndimz-3/2} + \frac{\rho^f \varepsilon^n_{ndimz-3/2}}{dz_{ndimz-2}}\left[\nu^e|^n_{ndimz-1}\ \frac{\phi^n_{ndimz-1}v^n_{ndimz-1} - \phi^n_{ndimz-3/2}v^n_{ndimz-3/2}}{\Delta z_{ndimz-1}} - \nu^e|^n_{ndimz-2}\ \frac{\phi^n_{ndimz-3/2}v^n_{ndimz-3/2} - \phi^n_{ndimz-1}v^n_{ndimz-1}}{\Delta z_{ndimz-2}}\right] \tag{58}$$

In which the value of $\Delta z_{ndimz-1}$ should be taken as $dz_{ndimz-2}/2$ (see explanation above), and the evaluation of $\phi^n_{ndimz-1}$ at the node $ndimz-1$ should be made in order to determine $\varepsilon^n_{ndimz-1}$ and $\nu^e|^n_{ndimz-1}$. In the latter two cases, $\phi^n_{ndimz-1}$ is taken equal to $\phi^n_{ndimz-3/2}$, and all the expressions can be evaluated. At step $i = ndimz-3$, there are no more differences with respect to the classical evaluation of the matrices, as no terms in $_{ndimz-1}$ appears.

3.4 Implementation

The issue of the implementation is to write the matricial system following the exact description given above, with vectors defined at different position and of different sizes. Indeed, the fluid velocity for example is defined in between the nodes but also at the two boundaries (see e.g. figure 4), so that its length is $ndimz-1+2 = ndimz+1$, with $ndimz$ the number of scalar nodes. In comparison, the solid (or fluid) volume fraction is evaluated only at the velocity nodes so that its length is $ndimz-1$. But more importantly, $phi[j]$ does not correspond to $ufn[j]$, as the velocity starts at zero while the solid volume fraction starts at 1/2. To image this, we can write the solid volume fraction and velocity vectors

$$\phi^n = (\phi^n_0, \phi^n_1, \phi^n_2, ..., \phi^n_{i-1}, \phi^n_i, \phi^n_{i+1}, ..., \phi^n_{ndimz-2}) = (\phi^n_{1/2}, \phi^n_{1+1/2}, ..., \phi^n_{i+1/2}, \phi^n_{i+3/2}, \phi^n_{i+5/2}, ...\phi^n_{ndimz-3/2}) \tag{59}$$

$$U^n = (U^n_0, U^n_1, U^n_2, ..., U^n_{i-1}, U^n_i, U^n_{i+1}, ..., U^n_{ndimz-1}, U^n_{ndimz}) = (u^n_0, u^n_{1/2}, u^n_{1+1/2}, ...u^n_{i-1/2}, u^n_{i+1/2}, u^n_{i+3/2}, ...u^n_{ndimz-1/2}, u^n_{ndimz}) \tag{60}$$

We clearly see there that $\phi_0^n = \phi_{1/2}^n$ while $U_0^n = u_0^n$ and $U_1^n = u_{1/2}^n$, or in a more general form that, $\phi_j^n = \phi_{j+1/2}^n$ while $U_j^n = u_{j-1/2}^n$. All the quantities evaluated in the DEM, i.e. the solid volume fraction ϕ^n, the averaged drag term β^n, the solid velocity v^n, as well as the step definition vector (*dsig* in the code) are of size $ndimz-1$ and follow the same behavior as the solid volume fraction presented here. Meanwhile, the fluid velocity at time step n and $n+1$ are of dimension $ndimz+1$ and follows the pattern exposed above. In between, the scalar node position vector (called *sig* in the code) is of size *ndimz*, as well as the turbulent and effective viscosity vectors. The latter are defined as

$$\nu^n = (\nu_0^n, \nu_1^n, \nu_2^n, ..., \nu_{i-1}^n, \nu_i^n, \nu_{i+1}^n, ..., \nu_{ndimz-1}^n) = (\nu_0^n, \nu_1^n, \nu_2^n, ... \nu_{i-1}^n, \nu_i^n, \nu_{i+1}^n, ... \nu_{ndimz-2}^n, \nu_{ndimz-1}^n) \tag{61}$$

as they are defined at the scalar nodes.

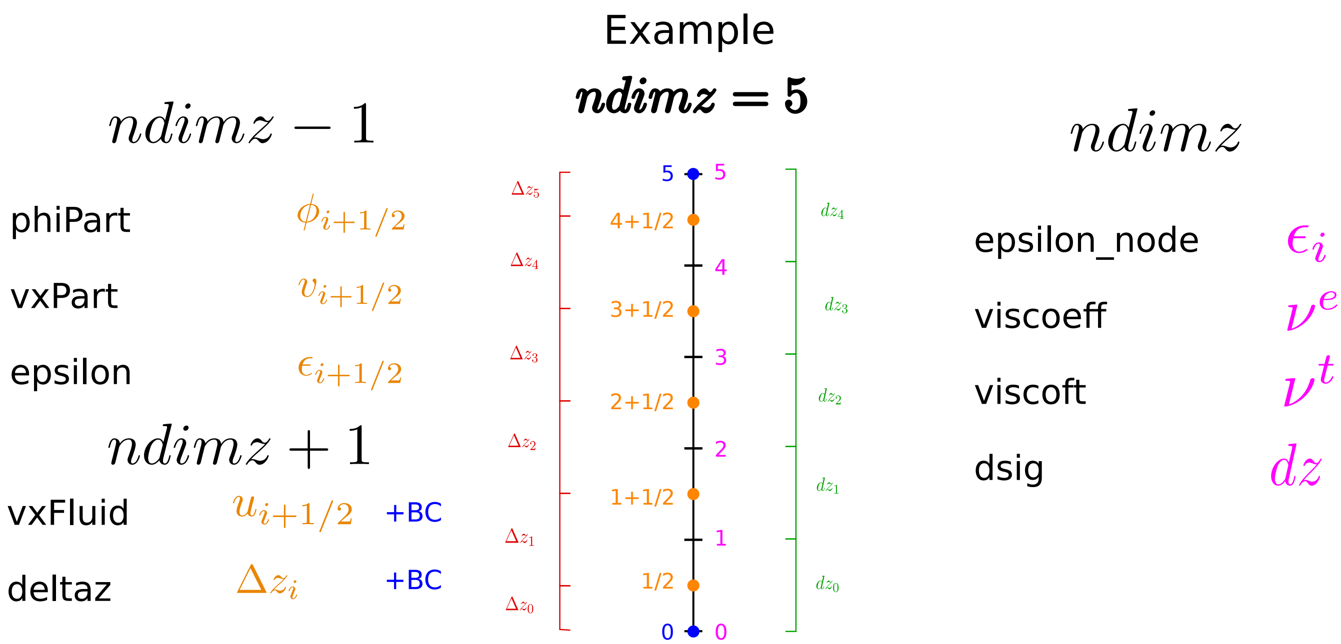

Figure 4. Example of mesh, and corresponding vectors definition. This illustrate the issue of the implementation through the size of the vectors obtained and the position where they are obtained.

Therefore, the fluid volume fraction at the scalar nodes (epsilon_node) as well as the step between two computational nodes (deltaz) are evaluated in the code before the computation of the matricial system

```
for (j=0;j<nCell;j++){
  if ((j!=0)&&(j!=nCell-1)){
    deltaz[j] = 0.5*(dsig[j-1]+dsig[j]);
    epsilon_node[j] = 0.5*(epsilon[j-1]+epsilon[j]);
  }
  else if (j==0) {
    deltaz[j] = 0.5*dsig[j];
    epsilon_node[j] = epsilon[j];
  }
  else if (j==nCell-1){
    deltaz[j] = 0.5*dsig[j-1];
    epsilon_node[j] = epsilon[j-1];
  }
}
```

Here, the fluid volume fraction is extrapolated linearly from the values at the two surrounding velocity nodes, and the same method of calculation is taken for deltaz. At the boundaries ($j == 0$ and $j == nCell-1$), deltaz is taken as a half of the cell size, and the solid volume fraction is taken to be the one from the nearest value (above and below respectively).
Following equations (25)-(28) and taking into account the extrapolations made above, the code is written as

```
for(j=0;j<nCell-1;j++){
  termeVisco_j = dt*epsilon[j]/dsig[j]*viscoeff[j]/deltaz[j];
  termeVisco_jp1= dt*epsilon[j]/dsig[j]*viscoeff[j+1]/deltaz[j+1];
  termeTurb_j = dt/dsig[j]*epsilon_node[j]*viscoft[j]/deltaz[j];
  termeTurb_jp1 = dt/dsig[j]*epsilon_node[j+1]*viscoft[j+1]/deltaz[j+1];

  if (j==0){ejm1 = epsilon[j]; phijm1 = phiPart[j]; upjm1 = vxPart[j];}
  else {ejm1 = epsilon[j-1]; phijm1 = phiPart[j-1]; upjm1 = vxPart[j-1];}
  if (j==nCell-1) {ejp1 = epsilon[j];phijp1=phiPart[j]; upjp1 = vxPart[j];}
  else {ejp1 = epsilon[j+1];phijp1=phiPart[j+1]; upjp1 = vxPart[j+1];}

  secondMemberPart = dt*epsilon[j]/dsig[j]*
        (viscoeff[j+1]/deltaz[j+1]*(phijp1*upjp1-phiPart[j]*vxPart[j])
         -viscoeff[j]/deltaz[j]*(phiPart[j]*vxPart[j]-phijm1*upjm1));

  a[j+1] = - termeVisco_j*ejm1 - termeTurb_j;
  b[j+1] =  termeVisco_jp1*epsilon[j] + termeVisco_j*epsilon[j]
           + termeTurb_jp1 + termeTurb_j + epsilon[j] + dt*taufsi[j];
  c[j+1] = - termeVisco_jp1*ejp1 - termeTurb_jp1;

  s[j+1]= ufn[j+1]*epsilon[j] + epsilon[j]*dt*std::abs(gravity[0])
         + dt*taufsi[j]*vxPart[j] + secondMemberPart;
}
```

This is supposed[2] to be the exact transcription of equation (25)-(28), where viscous and turbulent contribution are of the same shape and have been identified as *termVisco* and *TermTurb*. Also, the viscous terms are indeed multiplied by ε^2 (*termeVisco_j* $*\, ej$), and $s[j+1]$ includes a contribution from the solid phase (*secondMemberPart*) due to the closure of the effective stress tensor with the total velocity.

4 Conclusion

Building the matricial system from the 1D fluid momentum balance equation and solving it, we can obtain the volume-averaged fluid velocity profile at the next time step. Therefore, this process can be performed over a given number of time step to simulate the evolution of the fluid velocity profile over a given time. This is what is done in YADE when calling the fluidResolution(t_{simu},dt) of HydroForceEngine, which perform the described numerical fluid resolution $N = t_{simu}/dt$ times with a time step dt. The practical detail of the coupling with DEM using YADE and of the use of HydroForceEngine are detailed in (9).

Acknowledgements

I want to thank Julien Chauchat for developping the fluid code and for help in the adaptation to fluid-DEM coupling and in the translation to C++. I also want to thank Yade community for the support during development.

References

1. J. Chauchat. *Contribution à la modélisation diphasique du transport sédimentaire en milieux côtiers et estuariens*. PhD thesis, Université de Caen, 2007.

2. J. Chauchat. A comprehensive two-phase flow model for unidirectional sheet-flows. *Journal of Hydraulic Research*, 2017.

3. J. Chauchat, S. Guillou, D. Pham Van Bang, and K. Dan Nguyen. Modelling sedimentation-consolidation in the framework of a one-dimensional two-phase flow model. *Journal of Hydraulic Research*, 51(3):293–305, 2013.

4. A. Einstein. Eine neue Bestimmung der Molekül Dimensionen. *Ann. Physik*, 19:289–306, 1906.

[2]Even though I checked multiple times, I welcome any other confirmation!

5. R. Jackson. Locally averaged equations of motion for a mixture of identical spherical particles and a newtonian fluid. *Chemical Engineering Science*, 52(15):2457 – 2469, 1997.

6. R. Jackson. *The dynamics of fluidized particles*. Cambridge University Press, 2000.

7. L. Li and M. Sawamoto. Multi-phase model on sediment transport in sheet-flow regime under oscillatory flow. *Coastal engineering Japan*, 38:157–178, 1995.

8. R. Maurin. *Investigation of granular behavior in bedload transport using an Eulerian-Lagrangian model*. PhD thesis, Université Grenoble Alpes, 2015.

9. R. Maurin. *YADE 1D vertical VANS fluid resolution : Practical details*. Yade Technical Archive, 2018.

10. R. Maurin. *YADE 1D vertical VANS fluid resolution : Theoretical bases*. Yade Technical Archive, 2018.

11. R. Maurin. *YADE 1D vertical VANS fluid resolution : validations*. Yade Technical Archive, 2018.

12. R. Maurin, J. Chauchat, B. Chareyre, and P. Frey. A minimal coupled fluid-discrete element model for bedload transport. *Physics of Fluids*, 27(11):113302, 2015.

13. R. Maurin, J. Chauchat, and P. Frey. Dense granular flow rheology in turbulent bedload transport. *Journal of Fluid Mechanics*, 804:490–512, 10 2016.

14. R. Maurin, J. Chauchat, and P. Frey. Revisiting slope influence in turbulent bedload transport: consequences for vertical flow structure and transport rate scaling. *Journal of Fluid Mechanics*, 839:135–156, 2018.

15. T. Revil-Baudard and J. Chauchat. A two-phase model for sheet flow regime based on dense granular flow rheology. *Journal of Geophysical Research: Oceans*, 118(2):619–634, 2013.

16. Šmilauer et al. *Yade Documentation 2nd ed. The Yade Project (http://yade-dem.org/doc/) DOI: 10.5281/zenodo.34073*, 2015.