Computational Fluid Dynamics of Active Fluids

Luca Biancofiore
Department of Mechanical Engineering, Bilkent University, Bilkent 06800, Ankara, Turkey.
(Dated: January 20, 2021)

I. EQUATIONS FOR AN ACTIVE FLUID

An active fluid denotes a collection of particles, cells, or macromolecules, typically suspended in a viscous fluid, which are able to convert chemical energy into mechanical work. There are several kinds of active fluids , e.g. suspensions of active particles, such as Janus particles for instance [1], microswimmers suspensions [2], or active nematic fluids [3]. Since the definition of active fluids contains a broad variety of fluid systems, it is not surprising that their equations in the continuum description change significantly depending on what we need to model. In some cases a Lagrangian approach is the best option to model the active particles moving in the fluid (that it will described by the Navier-Stokes equations), while in other cases an Eulerian model, to describe the ensemble of the particles plus the fluid acting as a continuum, is the most suitable choice. In the following sections we will describe the equations for three systems as examples.

A. Motion of a Janus particle in a fluid

Let's consider first the motion of an active Janus particle (JP) in a incompressible Newtonian fluid [1]. We can descrive the motion of the particle from the Newton's equation

$$M\frac{d\mathbf{V}}{dt} = -\int_{\Sigma(t)} \mathbf{P}(\mathbf{r},t)\cdot\mathbf{n}d\Sigma + \mathbf{F}_{ext} \tag{1}$$

where M is the mass of the Janus particle, $\Sigma(t)$ is its surface, $\mathbf{V}$ is its velocity, $\mathbf{P}(\mathbf{r},t)$ is the fluid pressure tensor, $\mathbf{F}_{ext}$ is the sum of all the external forces (if any), t is the time and $\mathbf{r}(x,y,z)$ is the position in the domain. Moreover, the fluid exerts also a torque on the JP and its angular velocity $\mathbf{\Omega}$ is given by

$$\mathcal{I}\cdot\frac{d\mathbf{\Omega}}{dt} = -\int_{\Sigma(t)} \nabla\mathbf{r}\times[\mathbf{P}(\mathbf{r},t)\cdot\mathbf{n}]d\Sigma + \mathbf{T}_{ext}, \tag{2}$$

where $\mathcal{I}$ the inertia tensor of the JP and $\mathbf{T}_{ext}$ is the sum of the external torques.

To find the fluid pressure acting on the particle surface we need to solve the fluctuating Navier-Stokes equations for an incompressible Newtonian fluid.

$$\boldsymbol{\nabla}\cdot\mathbf{u} = 0 \tag{3a}$$
$$\rho(\partial_t\mathbf{u} + \mathbf{u}\cdot\boldsymbol{\nabla}\mathbf{u}) = -\boldsymbol{\nabla}\cdot\mathbf{P}. \tag{3b}$$

Note that the pressure tensor can be expressed as $\mathbf{P} = p\mathbf{I} + \mathbf{\Pi}$, where p is the hydrostatic pressure and the viscous pressure tensor components are given by

$$\Pi_{ij} = -\mu(\partial_i v_j + \partial_j v_i) + \pi_{ij}. \tag{4}$$

In Eq. 4 we introduced the dynamic viscosity μ and a Gaussian white noise field π_{ij} characterised by

$$\langle\pi_{ij}(\mathbf{r},t)\rangle = 0 \tag{5a}$$
$$\langle\pi_{ij}(\mathbf{r},t)\pi_{kl}(\mathbf{r}',t')\rangle = $$
$$= 2k_bT\mu(\delta_i k\delta_j l + \delta_{il}\delta_{jk})\delta(\mathbf{r}-\mathbf{r}')\delta(t-t'), \tag{5b}$$

where k_b is the Boltzmann constant and T is the temperature. We need to couple this equation with the energy equation to solve for velocity, pressure and temperature as e.g. in Ref. [1]. Obviously all the equations need to be coupled with suitable boundary conditions.

We can extend, by using a suitable model for the particle collision (see for instance Ref. [4]) this model also to a group of particles. Using a Lagrangian description for the particles motion we gain in accuracy but it will be computationally unbearable if the number of particles start to be too large.

B. Active suspension flow - 1

A continuum model for active suspensions was developed first by Saintilllian and Shelly [5, 6]. Here we consider a version of this model for a suspension of active Brownian swimmers as described in Ref. [2]. The fluid motion particularly is assumed to be incompressible and Newtonian. Then the velocity and the pressure must satisfy

$$\rho(\partial_t\mathbf{u} + \mathbf{u}\cdot\boldsymbol{\nabla}\mathbf{u}) = -\boldsymbol{\nabla}p + \mu\boldsymbol{\nabla}^2\mathbf{u} + n\sigma_0\boldsymbol{\nabla}\mathbf{D}, \tag{6a}$$
$$\boldsymbol{\nabla}\cdot\mathbf{u} = 0. \tag{6b}$$

Eqs. 6 is the Navier-Stokes equations corrected by the addition of the nematic alignment tensor $\mathbf{D}$ (n is the mean number density of particles in the suspension, while σ_0 is the force dipole strength that arises from the first moment of the force distribution on the particle surface [7]). This term captures the effect of the force dipoles exerted by the swimmers on the fluid while self-propelling. Again we need to couple Eqs. 6 with a suitable boundary conditions. For instance in a confined domain it could be a no-slip conditions, i.e. $\mathbf{u} = 0$ at the boundaries. This corrected Navier-Stokes equations must be coupled with other transport equations for (i) the particle concentration c, (ii) the polarisation $\mathbf{m}$ and (iii) the already introduced tensor $\mathbf{D}$.

C. Active suspension flow - 2

A generalised Navier-Stokes equations for active suspension particularly adapted to shear flows was devel-

oped by Slomka and Dunkel [8–10].

$$\rho(\partial_t \mathbf{u} + \mathbf{u}\cdot\boldsymbol{\nabla}\mathbf{u}) = \\ = -\boldsymbol{\nabla}p + \Gamma_0\boldsymbol{\nabla}^2\mathbf{u} - \Gamma_2\boldsymbol{\nabla}^4\mathbf{u} + \Gamma_4\boldsymbol{\nabla}^6\mathbf{u} \quad (7a)$$

$$\boldsymbol{\nabla}\cdot\mathbf{u} = 0, \quad (7b)$$

where $\boldsymbol{\nabla}^{2n} \equiv (\boldsymbol{\nabla}^2)^n$, if $n > 2$ we take into account non-Newtonian effects. The constant Γ_0 and Γ_4 must be positive to ensure asymptotic stability, while the constant Γ_2 can have both signs. If $\Gamma_2 = \Gamma_4 = 0$ we find the standard Navier-Stokes equations in which the dynamics viscosity $\mu = \Gamma_0$.

II. NAVIER-STOKES EQUATIONS

The models presented in the previous sections (I A-I C) are three examples of descriptions suitable for different kinds of active systems (or different flow configurations). They have something in common though: they are all based on the incompressible Newtonian Navier-Stokes equations

$$\rho(\partial_t \mathbf{u} + \mathbf{u}\cdot\boldsymbol{\nabla}\mathbf{u}) = -\boldsymbol{\nabla}p + \mu\boldsymbol{\nabla}^2\mathbf{u} \quad (8a)$$

$$\boldsymbol{\nabla}\cdot\mathbf{u} = 0. \quad (8b)$$

The momentum equation (Eq. 8a) is a transport equation. The term $\partial_t\mathbf{u}$ is the transient term, $\mathbf{u}\cdot\boldsymbol{\nabla}\mathbf{u}$ is the advective (or convective term) term, $\boldsymbol{\nabla}p$ is the pressure term (which is a source/sink of momentum term) and $\mu\boldsymbol{\nabla}^2\mathbf{u}$ is the viscous diffusion term. The derivation of Eqs. 8 is out of the scope of this text but the interested reader is referred to Ref. [11].

The list of models presented in these notes is obviously not exhaustive. For instance we did not present all the models based on the Toner-Tu equations [12, 13], which are possible also to supplement with a Swift-Hohenberg-like term [3]. However also in these approaches based on the Toner-Tu model common for active nematic fluids, terms typical of the Navier-Stokes equations (e.g advection or viscous diffusion) are present. Briefly, the discretisation of a Navier-Stokes-type equation will be the minimum common factor of all the available models of active fluids from a continuum mechanics point of view. Therefore we believe that a course devoted to the numerics of active matter cannot lack of a chapter about the discretisation of the Navier-Stokes equations (and more in general transport equations).

A. Non-dimensionalisation

We can introduce the following non-dimensional quantities

$$\mathbf{u}^* = \mathbf{u}/U, \quad \mathbf{x}^* = \mathbf{x}/L, \quad t^* = tU/L, \\ \text{and} \quad p^* = p/(\mu U/L), \quad (9)$$

where U is the reference velocity and L the reference length. There are other possible references scales for the pressure but the viscous pressure $\mu U/L$ is the most suitable for most of the active matter applications. By substituting Eqs. 9 into Eqs. 8 we obtain the non-dimensional Navier-Stokes equations

$$Re(\partial_t^*\mathbf{u}^* + \mathbf{u}^*\cdot\boldsymbol{\nabla}^*\mathbf{u}^*) = -\boldsymbol{\nabla}^*p^* + \boldsymbol{\nabla}^{*2}\mathbf{u}^* \quad (10a)$$

$$\boldsymbol{\nabla}^*\cdot\mathbf{u}^* = 0. \quad (10b)$$

where we introduced the Reynolds number $Re = \frac{\rho UL}{\mu}$ which is the ratio of the inertial to the viscous forces. If the viscous forces are much stronger than the inertial ones, we can assume that $Re = 0$ to obtain the Stokes flow (or creeping flow) equations

$$-\boldsymbol{\nabla}^*p^* + \boldsymbol{\nabla}^{*2}\mathbf{u}^* = 0 \quad (11a)$$

$$\boldsymbol{\nabla}^*\cdot\mathbf{u}^* = 0. \quad (11b)$$

Stokes flow equations are widely used in microfluidics where the pressure gradient is counterbalanced from viscous diffusion. From now on, we will work on the non-dimensional Navier-Stokes equations (Eqs. 10) but neglecting the asterisks.

III. DISCRETISATION OF A GENERIC TRANSPORT EQUATION VIA THE FINITE VOLUME METHOD

Our goal is to arrive to the discretisation of the incompressible Navier-Stokes (NS) equations (Eqs. 8). To do so we prefer to start to work on the differential form of the transport equation for the generic transported quantity ϕ

$$\frac{\partial(\rho\phi)}{\partial t} + \boldsymbol{\nabla}\cdot(\rho\phi\mathbf{u}) = \boldsymbol{\nabla}\cdot(\Gamma\boldsymbol{\nabla}\phi) + S_\phi \quad (12)$$

where Γ is the diffusion coefficient relative to the quantity ϕ. In the left hand side (LHS) we have the rate of increase of ϕ in the fluid element ($\frac{\partial(\rho\phi)}{\partial t}$) and the net rate of flow of ϕ out of the fluid element $\boldsymbol{\nabla}\cdot(\rho\phi\mathbf{u})$. In the right hand side (RHS) we have rate of increase of ϕ due to the diffusion $\boldsymbol{\nabla}\cdot(\Gamma\boldsymbol{\nabla}\phi)$ and the rate of the increase due to sources (that can be due to the surface or body forces).

A. Finite Volume Method

In this book we use the Finite Volume Method (FVM) to discretize Eq. 12. We chose FVM since it is widely used in computational fluid dynamics in commercial (e.g. Ansys Fluent) or open-source software (e.g. OpenFoam). The interested reader is referred to Refs. [14–16] for a more comprehensive dissertation on this numerical method. Other possible discretisations could make use of finite difference or finite elements.

The key step of the FVM is to integrate Eq. 12 over a three-dimensional control volume (CV):

$$\int_{CV}\frac{\partial(\rho\phi)}{\partial t}dV + \int_{CV}\boldsymbol{\nabla}\cdot(\rho\phi\mathbf{u})dV = \\ = \int_{CV}\boldsymbol{\nabla}\cdot(\Gamma\boldsymbol{\nabla}\phi)dV + \int_{CV}S_\phi dV \quad (13)$$

We can use the Gauss' divergence theorem to rewrite the integral representing the convective and diffusive terms

over the bounding surface A of the CV

$$\int_{CV} \frac{\partial \rho\phi}{\partial t} dV + \int_A \mathbf{n} \cdot (\rho\phi\mathbf{u}) dA = \\ = \int_A \mathbf{n} \cdot (\Gamma \nabla\phi) dA + \int_{CV} S_\phi dV. \quad (14)$$

Let's describe shortly the four terms of Eq. 14

- $\int_{CV} \frac{\partial(\rho\phi)}{\partial t} dV$ represents the rate of change of the total amount of ϕ in the CV.
- In the term $\int_A \mathbf{n} \cdot (\rho\phi\mathbf{u}) dA$ we have $\mathbf{n} \cdot (\rho\phi\mathbf{u})$ that is the advective flux of ϕ due to the fluid flowing along the outward normal component $\mathbf{n}$. Thus the second term on the LHS is the net rate of decrease of ϕ in the CV due to the advection.
- To understand the first term of the RHS ($\int_A \mathbf{n} \cdot (\Gamma \boldsymbol{\nabla}\phi) dA$) we need first to remember that a diffusive flux is positive in the direction of a negative gradient (i.e. heat conducted from positive to negative temperature). For this reason the $\mathbf{n} \cdot (-\Gamma \boldsymbol{\nabla}\phi)$ is the diffusive flux along the outward normal vector, i.e. out of the CV. This means that $-\mathbf{n} \cdot (-\Gamma \boldsymbol{\nabla}\phi)$ can be seen as a positive diffusive flux along the inward normal vector (i.e. inside of the CV). In conclusion, the first of term of the RHS represents the net rate of increase of fluid property ϕ of the fluid element due to diffusion.
- $\int_{CV} S_\phi dV$ represents the rate of increase of of ϕ as a result of sources/sinks inside the CV.

Note that in this section we assume that the velocity field $\mathbf{u}$ is somehow known. Otherwise it would have been necessary to couple the transport equation with the Navier-Stokes equations (Eqs. 8).

1. *Advection-diffusion*

In this section we focus on the steady state of Eq. 14

$$\int_A \mathbf{n} \cdot (\rho\phi\mathbf{u}) dA = \int_A \mathbf{n} \cdot (\Gamma \nabla\phi) dA + \int_{CV} S_\phi dV. \quad (15)$$

This equation needs to be coupled with the continuity equation

$$\boldsymbol{\nabla} \cdot \mathbf{u} = 0. \quad (16)$$

First of all we need to define the CV. We work using the two-dimensional CV centered in P shown in figure 1. The analysed CV has four neighbour CVs defined with capital letters: northern N, southern S, eastern E and western W neighbours. The faces of the CV are defined with lowercase: northern n, southern s, eastern e and western w faces. We define the cross-sectional areas of the CV as $A_e = A_w = \Delta y$ and $A_n = A_s = \Delta x$.

Let's integrate Eqs. 15-16 over the CV to obtain

$$(\rho u A\phi)_e - (\rho u A\phi)_w + (\rho v A\phi)_n - (\rho v A\phi)_s = \\ = \left(\Gamma A \frac{\partial\phi}{\partial x}\right)_e - \left(\Gamma A \frac{\partial\phi}{\partial x}\right)_w + \\ + \left(\Gamma A \frac{\partial\phi}{\partial y}\right)_s - \left(\Gamma A \frac{\partial\phi}{\partial y}\right)_n + \bar{S}\Delta V, \quad \text{and} \quad (17)$$

$$(\rho u A)_e - (\rho u A)_w + (\rho v A)_n - (\rho v A)_s = 0, \quad (18)$$

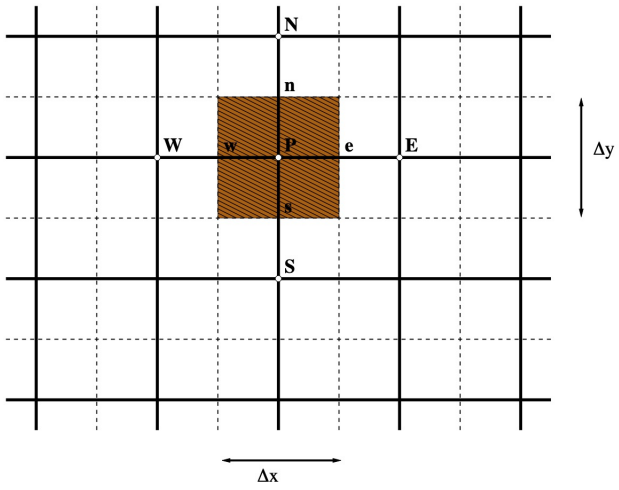

FIG. 1. The 2D control volume used to discretise Eq. 15.

where ΔV is the CV volume and $\bar{S}$ is the average value of the source S over the CV. We can assume that (i) Γ and ρ are constant along the domain and (ii) the grid is uniform (i.e all the CVs have the same size). In this manner we can write

$$\rho\Delta y u_e \phi_e - \rho\Delta y u_w \phi_w + \rho\Delta x u_n \phi_n - \rho\Delta x u_s \phi_s = \\ = \Gamma\Delta y \left(\frac{\partial\phi}{\partial x}\right)_e - \Gamma\Delta y \left(\frac{\partial\phi}{\partial x}\right)_w + \\ + \Gamma\Delta x \left(\frac{\partial\phi}{\partial y}\right)_n - \Gamma\Delta x \left(\frac{\partial\phi}{\partial y}\right)_s + \bar{S}\Delta V. \quad (19)$$

Now we have to decide how to approximate the values of ϕ and its derivatives at the faces. In this chapter we use for both the central differencing scheme, which corresponds to a linear interpolation. While for the diffusion term this scheme is perfectly suitable, for the convective terms it depends on the system and the parameters. We will briefly discuss about it later. Therefore we can approximate by using central differencing for the values of ϕ at the faces

$$\phi_e = \frac{\phi_E + \phi_P}{2}, \quad \phi_w = \frac{\phi_P + \phi_W}{2}, \\ \phi_n = \frac{\phi_N + \phi_P}{2}, \quad \text{and} \quad \phi_s = \frac{\phi_P + \phi_S}{2}, \quad (20)$$

and for the fluxes

$$\left(\frac{\partial\phi}{\partial x}\right)_e = \frac{\phi_E - \phi_P}{\Delta x}, \quad \left(\frac{\partial\phi}{\partial x}\right)_w = \frac{\phi_P - \phi_W}{\Delta x}, \\ \left(\frac{\partial\phi}{\partial y}\right)_n = \frac{\phi_N - \phi_P}{\Delta y}, \quad \text{and} \\ \left(\frac{\partial\phi}{\partial y}\right)_s = \frac{\phi_P - \phi_S}{\Delta y}. \quad (21)$$

Furthermore we can introduce to simplify the equation the convective fluxes F and the diffusive fluxes D as

$$F_e = \rho\Delta y u_e, \quad F_w = \rho\Delta y u_w, \\ F_n = \rho\Delta x v_n, \text{and} \quad F_s = \rho\Delta x v_s, \quad (22)$$

and

$$D_e = D_w = \frac{\Gamma\Delta y}{\Delta x} \quad \text{and} \quad D_n = D_s = \frac{\Gamma\Delta x}{\Delta y}. \quad (23)$$

Finally, we need to treat the source term. In practical situations S could be a function of the dependent variable

ϕ. In these cases FVM approximates the source term in a linear form

$$\bar{S}\Delta V = S_u + S_p\phi_P. \tag{24}$$

Replacing Eqs. 20-24 in Eqs. 18 and 19 we obtain

$$F_e - F_w + F_n - F_s = 0, \quad \text{and} \tag{25}$$

$$\begin{aligned}
&\frac{F_e}{2}(\phi_P + \phi_E) - \frac{F_w}{2}(\phi_W + \phi_P) + \\
&+ \frac{F_n}{2}(\phi_P + \phi_N) - \frac{F_s}{2}(\phi_S + \phi_P) = \\
&= D_e(\phi_E - \phi_P) - D_w(\phi_P - \phi_W) + \\
&+ D_n(\phi_N - \phi_P) - D_s(\phi_P - \phi_S) + \\
&+ S_u + S_P\phi_P.
\end{aligned} \tag{26}$$

We can rearrange Eq. 26

$$\begin{aligned}
&\left[\left(D_w - \frac{F_w}{2}\right) + \left(D_e + \frac{F_e}{2}\right) + \right. \\
&\left. + \left(D_s - \frac{F_s}{2}\right) + \left(D_n + \frac{F_n}{2}\right) - S_p\right]\phi_P = \\
&= \left(D_w + \frac{F_w}{2}\right)\phi_W + \left(D_e - \frac{F_e}{2}\right)\phi_E + \\
&+ \left(D_s + \frac{F_s}{2}\right)\phi_S + \left(D_n - \frac{F_n}{2}\right)\phi_N + S_u.
\end{aligned} \tag{27}$$

We can also let appear the continuity equation (Eq. 25) in the coefficient of ϕ_P

$$\begin{aligned}
&\left[\left(D_w + \frac{F_w}{2}\right) + \left(D_e - \frac{F_e}{2}\right) + \right. \\
&+ \left(D_s + \frac{F_s}{2}\right) + \left(D_n - \frac{F_n}{2}\right) + \\
&\left. + (F_e - F_w + F_n - F_s) - S_p\right]\phi_P = \\
&= \left(D_w + \frac{F_w}{2}\right)\phi_W + \left(D_e - \frac{F_e}{2}\right)\phi_E + \\
&+ \left(D_s + \frac{F_s}{2}\right)\phi_S + \left(D_n - \frac{F_n}{2}\right)\phi_N + S_u.
\end{aligned} \tag{28}$$

The notation of Eq. 28 can be simplified by

$$a_P\phi_P = a_W\phi_W + a_E\phi_E + a_N\phi_N + a_S\phi_S + S_u, \tag{29}$$

where $a_W = D_w + \frac{F_w}{2}$, $a_E = D_e - \frac{F_e}{2}$, $a_S = D_s + \frac{F_s}{2}$, $a_N = D_n - \frac{F_n}{2}$, and $a_P = a_W + a_E + a_S + a_N + (F_e - F_w + F_n - F_s) - S_p$.

The unknowns of Eq. 29 are ϕ_P, ϕ_E, ϕ_W, ϕ_N and ϕ_S. Since we need to write a similar equation for every grid point of the domain, we have transformed a partial differential equation (PDE) like the steady general transport equation (i.e. the steady state of Eq. 12) in a system of M-algebraic equations, where M is the number of grid points. If M is large the solution of this system can be computationally expensive. There are several methods to solve it such as the Tri-Diagonal Matrix Algorithm (TDMA or Thomas algorithm), Gauss-Seidel or Multigrid techniques to name a few. A detailed treatment of these (or other) algorithms are out of scope of this chapter. The interested reader is referred to [14] to have more information. Libraries (or also simple commands) to use this kind of algorithms are available in the most common computational languages.

In most of the practical applications however an iterative algorithm to solve the system of M-algebraic equations must be implemented. The convergence of the numerical solution is not ensured for all the numerical methods. Particularly a numerical scheme must be *bounded* to avoid divergence and converge to an accurate solution. More details can be found in appendix A. One way to check if a numerical scheme is bounded or not is to control if the coefficients of ϕ_P, ϕ_E, ϕ_W, ϕ_N and ϕ_S in the equivalent of Eq. 29 have the same sign. For central differencing this is clearly not guaranteed. Assuming u and v are positive for instance and, as a consequence, a_W and a_S are positive as well, a_E and a_N are positive only if $D_e > \frac{F_e}{2}$ and $D_n > \frac{F_n}{2}$, respectively. We define as Peclet number Pe the ratio between convective and diffusive fluxes $Pe = \frac{F}{D}$. Then the central differencing scheme is bounded only if the Pe at the eastern and the northern face is $Pe_e < 2$ and $Pe_n < 2$. Since the diffusive flux D does depend on the grid size, a grid refinement could help in ensuring boundedness. However a too fine grid means too expensive simulations. Furthermore, central differencing does not embed transportiveness at all (see appendix A). Transportiveness is an important feature for a numerical scheme if the advective term is important. In conclusion, if advection is strong in our flow system, it is preferable to discretize the convective terms with other schemes, such as e.g. upwind (see exercise 2), linear upwind or QUICK [14], depending on the degree of accuracy that we need in our simulations. Many applications in active matter though do not present a large Pe, making central differencing a suitable numerical scheme for both convection and diffusion terms.

2. *Boundary conditions*

In the previous section we have shown how it is possible to transform the steady general transport equation (i.e. the steady state of Eq. 12) into a systems of algebraic equations. However the generic equation for the node P (Eq. 29) is valid just for internal nodes. If the node is close to a boundary (like in figure 2 for instance) we need to take into account the presence of the boundary in our final algebraic equation for the boundary node P. Particularly two main kinds of boundary conditions are common in fluid mechanics

- Prescribed distribution of the boundary value of ϕ, i.e. $\phi = \phi_B$. It corresponds to a Dirichlet boundary condition.
- Prescribed distribution of the total (convective plus diffusive) flux at the boundary. It corresponds to a Robin boundary condition.

Let's work on the boundary CV depicted in Fig. 2 in which the east face of the cell coincides with the boundary B. The neighbouring CV lies in the boundary. In reality, this node does not exist in reality. The fluxes at the

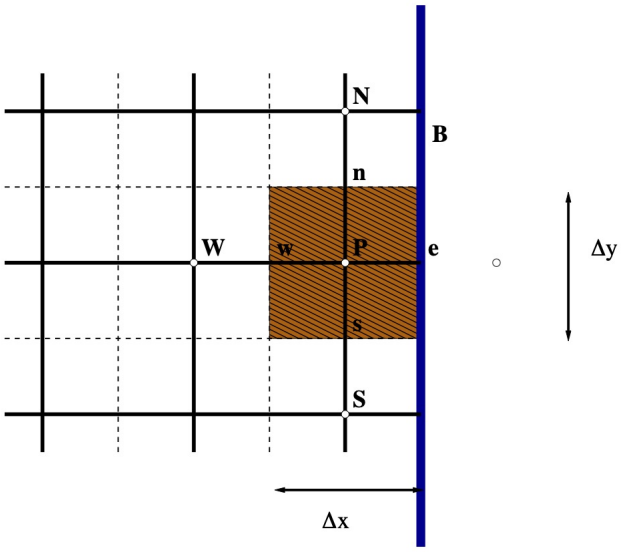

FIG. 2. A boundary control volume. Instead of the nodal neighbourg E, we have a boundary denoted with B.

boundary e can be written as

$$(\rho u\phi)_e - \Gamma \left(\frac{\partial \phi}{\partial x}\right)_e, \tag{30}$$

where evidently the cell face e coincides with the boundary. We cannot use the approximation used for internal node since we do not have a neighbouring node E to approximate the flux. However, an expression can easily be deduced from the relevant boundary condition.

- **Prescribed** ϕ. In this case, we need to modify the usual expression considering that the cell face and the boundary coincide, thus

$$(\rho u\phi)_e - \Gamma \left(\frac{\partial \phi}{\partial x}\right)_e = \rho u_B \phi_B - \Gamma \left(\frac{\phi_B - \phi_P}{\Delta x/2}\right). \tag{31}$$

  Note that in this expression we introduced also the velocity boundary condition u_B. In this scenario we introduce a source term $S_u + S_p\phi_P$, in which $S_u = \left(\frac{2\Gamma}{\Delta x} - \rho u_B\right)\Delta y = 2D_e - F_e$ and $S_p = \frac{2\Gamma}{\Delta x} = 2D_e$.

- **Prescribed total flux**. Since we know the total flux at the cell face e, the complete expression for the flux (Eq. 30) is simply replaced by the specified value. Note that the boundary value of ϕ_B does not enter in the solution directly. However we can evaluate it after the solution, using the expression for the flux and working out the value of ϕ_B.

3. Transient

In the previous sections we analysed the steady state of generic transport equation (Eq. 12). Now it's time to work on the transient term $\frac{\partial(\rho\phi)}{\partial t}$. We can approximate the transient term by using the FVM method

$$\int_{CV} \frac{\partial \rho\phi}{\partial t} dV \approx \rho \frac{d\phi_P}{dt} \Delta V. \tag{32}$$

However now we need to discretise the time derivative. There is a large variety of schemes to use, often based on finite-difference schemes. Let's first write Eq. 12 as

$$\rho \frac{d\phi_P}{dt} = f, \tag{33}$$

where f comes out from the steady discretised transport equation (Eq. 29)

$$a_P\phi_P - a_W\phi_W - a_E\phi_E - a_N\phi_N - a_S\phi_S - S_u = 0. \tag{34}$$

Since f depends on ϕ_P, ϕ_W, ϕ_E,ϕ_S, ϕ_N and t, we can write $f = f(\phi_I, t)$. In general we can discetise Eq. 32 as

$$\rho \frac{\phi_P^{n+1} - \phi_P^n}{\Delta t} = (1-\theta) f(\phi_I, t_n) + \theta f(\phi_I, t_{n+1}), \tag{35}$$

where $\theta \in [0, 1]$. However, we have three main options for choosing the value of θ.

1. $\theta = 0$. In this case we evaluate all the values of ϕ at the old time t_n. This is called explicit Euler method. This is the simplest way to discretise the temporal derivative. The main advantage is that the evaluation of ϕ_P^{n+1} does not depend on the neighbouring values at time t_{n+1}. This means that we can obtain the value of ϕ_P^{n+1} by straightforward substitution of the values ϕ_{nb}^n on the RHS. We can develop a simple algorithm for the explicit Euler method, starting from the known variable field at t_n and advancing with time Δt to the next time level t_{n+1}.

   (i) Assemble all the coeffienct a_I, and the source terms S_u and S_p.

   (ii) Scan the grid and compute ϕ_P^{n+1} at each point. The order in which each point is targeted is irrelevant, as the equation for ϕ_P^{n+1} does not depend on the neighbouring values at t_{n+1}.

   (iii) Advance the time from n to $n+1$ by adding Δt.

   The main setback of the explicit Euler method is that stable solutions can only be obtained under strict conditions (see exercise 4).

2. $\theta = 1$. In this case we evaluate all the values of ϕ at the old time t_{n+1}. We call this scheme the implicit Euler method. Each ϕ_P^{n+1} depends on its neighbours at the new time level ϕ_{nb}^{n+1}. This results in the impossibility of writing an explicit expression for ϕ_P^{n+1} in terms of known quantities. Rather, each ϕ_P^{n+1} is related implicitly to its neighbour values ϕ_{nb}^{n+1}. This means that a set of algebraic equations must be solved to determine the values of ϕ_P^{n+1}

3. $\theta = \frac{1}{2}$. We use both the old and the new time, equally weighted, to evaluate the values of ϕ. This is called the Crank-Nicolson scheme. This is the most accurate between the three presented scheme (it is second order accurate, while implicit and explicit Euler are only first order, see appendix A). The Crank-Nicolson algorithm is unconditionally stable [17] but boundedness is not ensured if the time step is too large (see exercise 4).

In computational fluid dynamics we need to respect the following condition

$$\frac{\mathcal{U}\Delta t}{\Delta x_{max}} \leq 1 \tag{36}$$

with $\mathcal{U}$ being the maximum propagation speed of the information, Δt is the time step and Δx_{max} is the maximum distance in the domain between grid points. Eq. 36 is called the Courant, Friedrichs and Lewy (CFL) condition and comes out from the principle of domain of dependence, i.e. the numerical solution in each time step must not advance further in space that the physical solution over the same time increment. If the CFL condition, which couples the numerical and the physical solution is not satisfied an explicit discretisation of the governing equation is not stable. Even if the numerical schemes are unconditionally stable (so they are allowed to exceed the CFL condition), violation of Eq. 36 may lead to unphysical results.

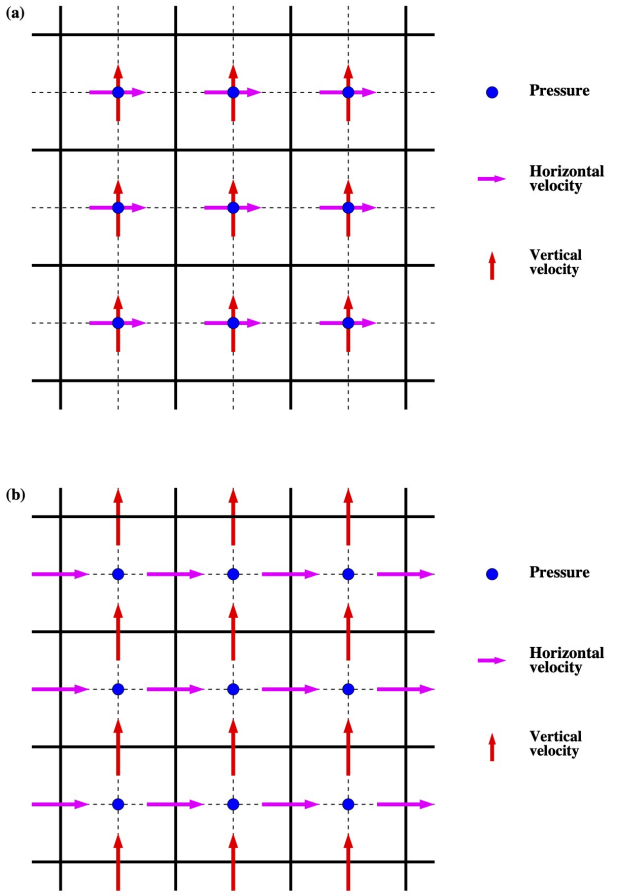

FIG. 3. Schematic of a (a) collacated grid and (b) a staggered grid. We show the storage of the (i) two velocity components u and v and (ii) the pressure p.

IV. DISCRETISATION OF NAVIER-STOKES

The incompressible Navier-Stokes (NS) equation momentum equation (Eq. 10a) is a transport equation of momentum. However there are two main complications in Eq. 10a with respect to the discretisation show in section III A: (i) the non-linearity in the advection term $(\mathbf{u}\cdot\boldsymbol{\nabla}\mathbf{u})$ and (ii) the role of the pressure.

The non-linearity in the advection term results from the convective fluxes F which depend on the unknowns $\mathbf{u}(u,v)$. There is then the need of initially "guessing" the values of the velocity in the convective fluxes and, through an iterative procedure, reach the final solution by solving both the momentum and the continuity equations (Eqs. 10). There are several schemes available in literature but before briefly presenting one of the most used (i.e. SIMPLE algorithm [18]) we must talk about the second challenge inherent in discretising the NS equations: the pressure gradient $(\boldsymbol{\nabla}p)$ and its role.

A. Pressure gradient

The main difficulty arising if the pressure field is not a priori known lies in the fact that we have no explicit equation for the pressure (for incompressible flows). The pressure merely appears as a gradient in the momentum equation. This means that the pressure field has to be chosen in such a way that the solution of the momentum equation gives a velocity field satisfying both the momentum and the continuity equation.

The first task is to decide where to store the pressure. In general the most obvious choice would be to store all the variables (velocities, pressure, etc.) at the nodal points, in the center of the CV . We call collocated grid this kind of arrangement (see Fig. 3a). Unfortunately a collocated grid presents some setbacks. In particular, if we use the central differencing scheme to approximate the pressure gradient, the dependency of the pressure node at the center falls out (see exercise 5). This could bring to oscillations and instabilities in the numerical scheme. There are ways to overcome this issue [16] but are out of the scope of this course.

However the easiest (and most common) solution is to apply a staggered grid in which the pressure values (and other scalar variables) are stored in the center of the CV, while the u- (v-) velocities are stored in the vertical (horizontal) faces of the CV (see Fig. 3b). In a staggered grid implementation, for instance, the approximation of

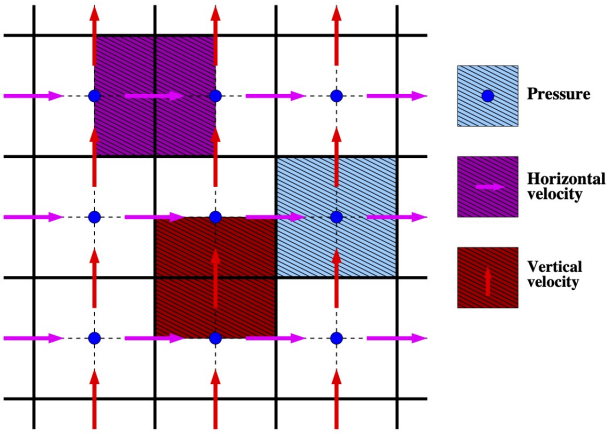

FIG. 4. The three CVs (u, v and p) used in a staggered grid to solve the NS equations.

the pressure gradient in the horizontal momentum equation can directly use the pressure values at the western and eastern faces since it is where they are defined. Furthermore the u- and v-control volumes are built directly around the velocity components. Thus, they are staggered by half a cell with respect to the CV of the pressure (and the other scalar variables) as it is possible to observe in Fig. 4.

B. Discretisation of the NS momentum equation

First of all we need to discretise the momentum (Eqs. 10a). For the u-momentum equation we can arrive

$$\begin{aligned}&\left(\rho\frac{du}{dt}\Delta x\Delta y + a_P^u\right)u_P = \\ &= a_N^u u_N + a_S^u u_S + a_E^u u_E + \\ &+ a_W^u u_W + b_P^u - (p_P - p_W)\Delta y\end{aligned} \tag{37}$$

where the coefficients are

$$a_N^u = \left(D_n - \frac{F_n}{2}\right) = \left(\frac{1}{\Delta y} - Re\frac{v_n}{2}\right)\Delta x, \tag{38a}$$

$$a_S^u = \left(D_s + \frac{F_s}{2}\right) = \left(\frac{1}{\Delta y} - Re\frac{v_s}{2}\right)\Delta x, \tag{38b}$$

$$a_E^u = \left(D_e - \frac{F_e}{2}\right) = \left(\frac{1}{\Delta x} - Re\frac{u_e}{2}\right)\Delta y, \tag{38c}$$

$$a_W^u = \left(D_w + \frac{F_w}{2}\right) = \left(\frac{1}{\Delta y} - Re\frac{u_w}{2}\right)\Delta y, \tag{38d}$$

$$a_P^u = a_N^u + a_S^u + a_E^u + a_W^u, \text{ and} \tag{38e}$$

$$b_P^u = S_P^u \Delta x\Delta y. \tag{38f}$$

If we use the central differencing scheme also for the convective mass fluxes we obtain for instance

$$F_n = \frac{\rho}{2}(v_N + v_P)\Delta x, \tag{39a}$$

$$F_e = \frac{\rho}{2}(u_E + u_P)\Delta y. \tag{39b}$$

Similarly we can obtain the discretise equation of the y-momentum equation

$$\begin{aligned}&\left(\rho\frac{dv}{dt}\Delta x\Delta y + a_P^v\right)u_P = \\ &= a_N^v v_N + a_S^v v_S + a_E^v v_E + \\ &+ a_W^v v_W + b_P^v - (p_P - p_W)\Delta y,\end{aligned} \tag{40}$$

where the expressions of the coefficients are left to be found by the interested reader (see exercise 6).

C. Discretisation of the continuity equation

The approximation of the continuity equation (Eq. 10b) through the FVM is straightforward. We use the CV around the scalar variables (see Fig. 4). After some calculations we obtain

$$(u_E - u_P)\Delta x + (v_N - v_P)\Delta y = 0 \tag{41}$$

D. SIMPLE algorithm

The *Semi-Implicit Method of Pressure Linked Equations* (SIMPLE) algorithm [18] has two different steps: (i) the predictor step and (ii) the corrector step. We leave the detailed discretisation to the reader, see exercise 6, but here we will give the general details.

In the predictor step we solve the momentum equations (Eqs. 37 and 40) to obtain u_P^* and v_P^* by using the pressure field and coefficients evaluated at the previous time-step or previous iteration, which ever is the most recent. This *predicted* velocity field closely satisfies the momentum equations but it will not necessary satisfies also the continuity equation (Eq. 41).

To solve this issue we need a corrector step. Let's introduce a pressure correction

$$p'_I = p_I^{n+1} - p_I^n \tag{42}$$

and after some calculations we can write expressions for the velocity corrections u' and v'

$$u'_P = u_P^{n+1} - u_P^* = -d_p^u(p'_P - p'_W), \tag{43a}$$

$$v'_P = v_P^{n+1} - v_P^* = -d_p^v(p'_P - p'_S) \tag{43b}$$

and similarly for the neighbour nodes, as for example

$$u_E^{n+1} - u_E^* = -d_p^u(p'_E - p'_P), \tag{43c}$$

$$v_N^{n+1} - v_N^* = -d_p^v(p'_N - p'_P). \tag{43d}$$

The d coefficients are given by

$$d_P^u = \frac{r\Delta y}{\rho\frac{du}{dt}\Delta x\Delta y + a_P^u}, \tag{44a}$$

$$d_P^v = \frac{r\Delta x}{\rho\frac{dv}{dt}\Delta x\Delta y + a_P^v}, \tag{44b}$$

$$d_E^u = \frac{r\Delta y}{\rho\frac{du}{dt}\Delta x\Delta y + a_E^u}, \tag{44c}$$

$$d_N^v = \frac{r\Delta x}{\rho\frac{dv}{dt}\Delta x\Delta y + a_N^v}. \tag{44d}$$

Note than in the SIMPLE algorithm the dependence on the neighbouring nodes is neglected. This is the major assumptions of SIMPLE and it could slow down the convergence speed but it does not infer with the accuracy of the solution.

In Eqs. 43 the only unknowns are the new velocity u^{n+1}, v^{n+1} and the pressure correction p'. The new velocities can be eliminated by inserting Eqs. 43 into the continuity equation (Eq. 41) to obtain a Poisson-type equation for the pressure correction

$$a_P^p p'_P = a_N^p p'_N + a_S^p p'_S + a_E^p p'_E + a_W^p p'_W + b_P, \tag{45}$$

where

$$a_N^p = d_N^v, \quad a_S^p = d_N^v, \quad a_E^p = d_E^u, \quad a_W^p = d_W^u, \tag{46a}$$
$$a_P^p = a_N^p + a_S^p + a_W^p + a_E^p, \text{ and} \tag{46b}$$
$$b_P = -(u_E^* - u_P^*) - (v_N^* - v_P^*). \tag{46c}$$

From Eq. 45 we can find the pressure correction since all the coefficients are known. We can find then the velocity correction from Eqs. 43 that can be used to correct the velocity field (that will automatically satisfies the continuity equation). However, it is not given that the corrected velocity field satisfies the momentum equations as well. For this reason we have to re-evaluate the coefficients of both momentum and continuity equations. In practice we need to repeat the entire procedure until reaching a velocity field which satisfies both continuity and momentum equations (at least to a reasonable level).

Finally we summarise below the main parts of the SIMPLE algorithm.

1. Initialise the velocity and the pressure fields.
2. Calculate the momentum equations coefficients (in particular d^u and d^v) and solve for the predictor velocity field u^* and v^* using the latest available pressure p^n.
3. Calculate the coefficients for the pressure correction (p') equation (Eq. 45) and solve for p'. With the pressure correction we can obtain p^{n+1} since $p^{n+1} = p^n + p'$.
4. Correct the velocity field by means of the pressure correction (Eqs. 43) and then the continuity equation is satisfied
5. Calculate the coefficients of the other scalar equations connected to the NS equations (energy, concentration, etc.) and solve for the new values.

The sequence 2 to 5 must be repeated until reaching convergence. SIMPLE is used by many CFD practitioners (and commercial and open source softwares as well) and it is observed to be quite robust despite there is no formal proof of its convergence. In general a necessary condition (but not sufficient) to achieve convergence is to have a decrease in the pressure correction while iterating.

There are several variants of SIMPLE available in literature (see [14] for further details). For instance in the SIMPLEC algorithm we do not neglect the dependence on the neighbouring nodes. This will increase the convergence speed. In SIMPLER there is a preliminary step, involving the so-called pseudo-velocities, to have a better initial value for the pressure to find the predicted velocities u^* and v^*. More in general, the SIMPLE algorithm has originally put forward for steady state problems, while PISO could be a better choice for unsteady flows.

V. EXERCISES

1. Show that the accuracy of central differencing scheme is second order. (Hint. How will you express the value of a function ϕ in E and W if you know the value of ϕ in P by using Taylor series approximations? Look at the CV in Fig. 1)
2. If the Peclet number $Pe > 2$ the central differencing scheme is not a suitable choice anymore since it is not bounded and a good transportiveness is required in this case. The easiest solution is to apply the so-called *upwind* scheme (or *donor cell*). In this scheme the values of the transported quantity ϕ, needed at the faces in the convective terms in Eq. 19, are approximated simply by the value of ϕ of the CV upwards (downwards) if the velocity is positive (negative). For instance if u is positive ϕ_e is approximated by ϕ_P. Please write the equivalent of 29 if we use upwind instead of central differencing to discretise the convective terms.
3. Obtain all the coefficients a, the convective and the diffusive fluxes (F and D) as we have done for Eq. 29 if (i) the density ρ and the diffusivity coefficients Γ are not constant along the domain and (ii) the grid is non-uniform.
4. Find what is the maximal allowed time step to ensure boundedness for the general transport equation (Eq. 12) without sources/sinks for the following temporal schemes
   (i) The explicit Euler scheme if only the diffusion term is present.
   (ii) For Crank-Nicolson scheme if only the diffusion term is present.
   (iii) For explicit Euler scheme if both advection and diffusion terms are present.
5. Assume that we have a checkerboard pressure field (i.e. the pressure is point-by-point oscillating between p_{max} and p_{min}). Calculate the pressure gradients $\frac{\partial p}{\partial x}$ and $\frac{\partial p}{\partial y}$ for two grids similar to 1 but the (i) the first one is collocated and (ii) the second one is staggered (see Fig. 3).
6. In this exercise we want to obtain step-by-step all the relevant equations for the SIMPLE algorithm. Note a staggered grid is used (see Fig. 3b).

(a) Starting from the non-dimensional NS equations (Eqs. 10) find Eqs. 37, 40 and 41 and all the relevant coefficients.

(b) Find the predictor velocities u^* and v^*.

(c) Write the velocities for the CV centred in the point P for the iteration level $n+1$ (u_P^{n+1} and v_P^{n+1}) by using Eqs. 37 and 40.

(d) Find the velocity correction u'_P and v'_P (subtract the expression for u^* and v^* found in (a) to the expressions for u_P^{n+1} and v_P^{n+1} found in (b)) by introducing the pressure correction (Eq. 42) and the d coefficients (Eqs. 44).

(e) Which term can we neglect in the expressions for u'_P and v'_P? Please neglect it to obtain Eqs. 43.

(f) Replace Eqs. 43 in the discretised continuity equation (Eq. 41) to find the pressure correction equation (Eq. 45).

(g) In general we use *under-relaxation* in SIMPLE to improve convergence. This means that we do not use directly the velocity field and the pressure obtained in the last time-step or iteration to solve the momentum equations to find $\mathbf{u}^*$ and $\mathbf{u}_P^{n+1}$ as done in (a) and (b). The solution for u, v and p after each iteration are regarded as provisional and so we can write

$$p_P^{n+1} = \alpha_p \hat{p}_P^{n+1} + (1-\alpha_p) p_P^n, \tag{47a}$$

$$u_P^{n+1} = \alpha_u \hat{u}_P^{n+1} + (1-\alpha_u) u_P^n, \text{ and} \tag{47b}$$

$$v_P^{n+1} = \alpha_v \hat{v}_P^{n+1} + (1-\alpha_v) v_P^n, \tag{47c}$$

where $\hat{p}$, $\hat{u}$ and $\hat{v}$ are the provisional solution and α_p, α_u and α_v are the under-relaxation factors for the pressure, horizontal and vertical velocities, respectively. The under-relaxation factors α_i can have values $0 \leq \alpha_i \leq 1$ (if they are larger than 1 they become over-relaxation factors, but its sure it is warmly discoureged for SIMPLE). Assume that $\alpha_p = \alpha_u = \alpha_v$ and write again all the SIMPLE algorithm introducing under-relaxation (in particular the expressions for the predicted velocities, the velocities for the iteration level $n+1$ and the d coefficients must be changed).

Appendix A: Properties of numerical schemes

In this appendix we present the most important numerical properties of numerical schemes.

1. Accuracy

Numerical solutions of fluid flows are only approximate solutions. To measure the degree of accuracy of a numerical method we should compare how much the approximated solution is close to the analytical one. However in many cases it is not possible to have an analytical solution, for this reason we have to find another method to systematically define the accuracy of a numerical scheme.

Let's use as an example the discretisation of the the term $\frac{d\phi}{dx}$. We need to approximate its value in P (i..e the center of the CV shown in Fig. 1)

$$\left(\frac{d\phi}{dx}\right)_P = F(\phi_I, \Delta x) + O(\Delta x^k), \tag{A1}$$

where F is a function depending on the values ϕ in some nodes (which nodes it depends on the numerical scheme) and the grid size along the x-direction (Δx), while $O(\Delta x^k)$ represents the truncated term. In practice the power k in the truncated term governs the rate at which the error tends to zero as the grid is refined and is called the order of the difference approximation. We define this power k as the degree of accuracy of a numerical scheme.

For the central differencing scheme we can write (see exercise 1)

$$\left(\frac{d\phi}{dx}\right)_P = \frac{\phi_E - \phi_W}{2\Delta x} + O(\Delta x^2). \tag{A2}$$

For this reason the accuracy of the central differencing scheme is second order.

2. Stability

A numerical solution is said to be stable if it does not magnify the errors appearing during the course of the numerical process. In the case of transient flows, stability guarantees to reach a bounded solution (see section A 3) if, of course, the exact solution is bounded. If iterations are involved, a simple definition of a stable numerical method is a scheme which does not diverge. If complex boundary conditions or - worse - non-linearities are present, it could be complicated to analyse the stability of a numerical method. However for these complex cases in general we can investigate the stability of the method for linear problems without boundary conditions. In general it was observed that the results obtained in this manner can be often used also for the more complex problems (but be careful since there are exceptions).

3. Boundedness

Numerical solutions should lie in proper bounds. Quantities that are always positive (such as density, viscosity, kinetic energy of turbulence, etc) have to be always positive. Concentrations, moreover, must lie between 0 and 1. If sources/sinks are not applied, some equations (like the heat equation for instance) want that the minimum and the maximum values of the variables are located at the boundaries of the domain. Therefore we would like ideally to have these constraints inherited by the numerical scheme. Schemes known to produce unbounded solutions could have easily problems in stability.

If the discretisation scheme does not satisfy boundedness, it is highly possible that the solution does not converge at all, or, if it does, that it contains oscillations.

There are two methods to control the boundedness of a numerical method

1. Check if the Scarborough's criterion is satisfied. This criterion states a sufficient condition for a convergent iterative method

$$\frac{\Sigma|a_{nb}|}{|a_P|} \begin{cases} \leq 1 & \text{at all nodes} \\ < 1 & \text{at one node at least} \end{cases} \tag{A3}$$

   This corresponds to have a diagonal dominant matrix of the coefficients a (see Eq. 29 for instance).

2. Control if all the coefficient of the discretised equations have the same sign (usually positive). Physically this implies that an increase in the variable ϕ at one node should result in an increase in ϕ at its neighbours.

4. Conservativeness

In fluid systems the equations to be solved are conservation laws. Ideally, the numerical schemes should respect these laws. In absence of sources/sinks at steady state, the amount of a quantify ϕ leaving a closed volume is equal to the amount of the same quantity going into the same volume. This is a fundamental property since it stabilises a constraint on the solution error. The error can only wrongly distribute conserved quantities (such as mass, momentum or energy for instance) over the domain. Non-conservative schemes can produce artificial sources and sinks. However if these schemes are consistent and stable can still lead to correct solutions if the grid is very fine (sometimes not easy to guarantee though). A good feature of the FVM is that - if coupled with the strong conservation form of the equations - guarantees conservativeness for each individual CV and for the solutions in the entire domain.

5. Transportiveness

Transportiveness for a numerical method measure how much the directionality of the flow is taken into account. Let's assume that our CV is represented in Fig. 1 and the velocity is positive and completely horizontal (i.e. $v = 0$). In this case the value of the quantity ϕ will be influenced more than the western neighbouring node (W) than the eastern neighbour (E). If the velocity is negative instead, we will have the opposite case (i.e. E more 'influential' than W). If a numerical scheme embeds this quality we say that it is a good transportiveness. Flows with strong advection (i.e. with high Peclet number, see section III A 1) need to be treated with schemes with good transportiveness, while if diffusion is dominant this property is less important. The central differencing scheme does not take account at all the flow direction. For this reason it is not a suitable scheme for the convective terms if the Peclet number is large.

[1] P. Gaspard and R. Kapral, Journal of Statistical Mechanics: Theory and Experiment **2019**, 074001 (2019).
[2] M. Theillard and D. Saintillan, Journal of Computational Physics **397**, 108841 (2019).
[3] H. H. Wensink, J. Dunkel, S. Heidenreich, K. Drescher, R. E. Goldstein, H. Löwen, and J. M. Yeomans, Proceedings of the National Academy of Sciences **109**, 14308 (2012).
[4] A. Callegari and G. Volpe, Flowing Matter , 211 (2019).
[5] D. Saintillan and M. J. Shelley, Physical Review Letters **100**, 178103 (2008).
[6] D. Saintillan and M. J. Shelley, Physics of Fluids **20**, 123304 (2008).
[7] D. Saintillan and M. J. Shelley, Physical review letters **99**, 058102 (2007).
[8] J. Słomka and J. Dunkel, The European Physical Journal Special Topics **224**, 1349 (2015).
[9] J. Słomka and J. Dunkel, Physical Review Fluids **2**, 043102 (2017).
[10] J. Słomka and J. Dunkel, Proceedings of the National Academy of Sciences **114**, 2119 (2017).
[11] P. Kundu, I. Cohen, and D. Dowling, "Fluid mechanics 6th edition," (2015).
[12] J. Toner and Y. Tu, Physical review E **58**, 4828 (1998).
[13] J. Toner, Y. Tu, and S. Ramaswamy, Annals of Physics **318**, 170 (2005).
[14] H. K. Versteeg and W. Malalasekera, *An introduction to computational fluid dynamics: the finite volume method* (Pearson education, 2007).
[15] B. Andersson, R. Andersson, L. Håkansson, M. Mortensen, R. Sudiyo, and B. Van Wachem, *Computational fluid dynamics for engineers* (Cambridge university press, 2011).
[16] F. Moukalled, L. Mangani, M. Darwish, *et al.*, *The finite volume method in computational fluid dynamics*, Vol. 6 (Springer, 2016).
[17] C. A. Fletcher, *Computational techniques for fluid dynamics 2: Specific techniques for different flow categories* (Springer Science & Business Media, 2012).
[18] S. V. Patankar and D. B. Spalding, in *Numerical prediction of flow, heat transfer, turbulence and combustion* (Elsevier, 1983) pp. 54–73.