

### 3

## Mathematical and Numerical Models

### 3.1

#### Governing Equations

The physics of fluid flow is governed by conservation equations based on the principles of conservation of mass (continuity), momentum (Newton's second law) and energy. These fundamental principles are expressed in terms of mathematical equations, which in their more general form are partial differential equations. By solving these equations, namely Navier-Stokes equations, it is possible to describe the fluid motion and its interaction with the surroundings, accounting the forces and the physical constraints that ultimately characterize the flow. For incompressible flow, neglecting body forces, and neglecting temperature changes the Navier-Stokes equations reduce to:

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j \partial x_j}, \quad (3.1.1)$$

$$\frac{\partial u_i}{\partial x_i} = 0, \quad (3.1.2)$$

where  $t$  is the time,  $u_i$  is the  $i$ -th component of the velocity vector,  $p$  is the thermodynamic pressure,  $\rho$  is the fluid density, and  $\nu$  is the kinematic viscosity.

#### 3.1.1

##### Turbulent Flow

Turbulent flows can be characterized using the same motion equations showed above, but as was pointed out, it would need massive computational resources to solve for all the scales that compound the complete flow. To deal with this problem, the governing equations can be filtered, so that only a relevant part of those scales are calculated. According to Wilcox [4] there exist three forms, used in turbulence-model research, in order to do that; the *time average*, the *spatial average*, and the *ensemble average*.

Starting from the Navier-Stokes equations, it is possible to derive new equations that describe the evolution of filtered magnitudes using any of the previous mentioned filtration process. The selection of the filtering process will

depend on the expected accuracy, the complexity of the flow, and the available computational resources. According to Wilcox [4], time averaging is the most appropriate approach because the majority of engineering applications are more concerned with the mean velocity field than the velocity fluctuations. Moreover, the relative low computational resources this approach demands is an advantage compared to the other methodologies.

The time averaging process is done representing the instantaneous velocity as the sum of a mean and a fluctuating part

$$u(x, t) = \bar{u}(x, t) + u'(x, t) , \quad (3.1.3)$$

where  $\bar{u}$  is the average of  $u$  defined by

$$\bar{u} = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T u \, dt , \quad (3.1.4)$$

and  $u'$  is the fluctuating part of  $u$ .

Replacing the previous velocity decomposition and taking the mean of the continuity equation 3.1.2 we obtain the mean-continuity equation for steady state:

$$\frac{\partial \bar{u}_i}{\partial x_i} = 0 \quad (3.1.5)$$

Replacing the previous velocity decomposition and taking the mean of the momentum equation 3.1.1 we obtain the mean-momentum equation:

$$\bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial \bar{p}}{\partial x_i} + \nu \frac{\partial^2 \bar{u}_i}{\partial x_j \partial x_j} - \frac{\partial \overline{u'_i u'_j}}{\partial x_j} \quad (3.1.6)$$

As can be seen, the non-linearity of the Navier Stokes equations leads to the appearance of a momentum flux that acts as an apparent stress throughout the flow. This term  $\overline{u'_i u'_j}$ , is unknown a priori and cannot be closed, so it is necessary to use additional equations to solve for the additional unknowns.

The approach used in the present study to represent the Reynolds stress term is based on the Boussinesq eddy-viscosity approximation, introduced by Boussinesq in 1977. In this approach, Boussinesq proposed an analogy between the turbulent stress and the viscous stress, in which the deviatoric part of the turbulent stress tensor can be calculated as a linear function of the mean strain rate as follows.

$$-\overline{u'_i u'_j} = -\frac{2}{3} k \delta_{ij} + \nu_t \left( \frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) , \quad (3.1.7)$$

where  $\nu_t$  is the kinematic turbulent viscosity and  $k$  is the turbulent kinetic energy.

Analogous to the molecular viscosity, the turbulent viscosity is defined as being the result of the momentum transfer between adjacent layers of fluid due to the movement of turbulent eddies at macroscopic level. Using dimensional analysis the turbulent viscosity can be calculated by the product of a characteristic velocity  $u^*(x, t)$  and a characteristic length  $l^*(x, t)$  scale as follows:

$$\nu_t = u^* l^* , \quad (3.1.8)$$

Therefore, to close this system of equations the characteristic values have to be computed in terms of known quantities obtained through modeled transport equations. The various models are classified in terms of number of transport equations solved in addition to the RANS equations. Among the most popular models used in aerodynamics, one and two equations models represent a good trade between low computational cost and accuracy. In the present study it was used one model from each class. The features of both turbulence models are described below together with their transport equations.

### 3.1.2

#### Spalart Allmaras Model

This model uses only one transport equation to compute for the turbulence kinetic viscosity  $\nu_t$ . Spalart and Allmaras in (1994) developed this model for aerodynamic applications, focusing to remove the incompleteness of algebraic and one equations models based on kinetic energy, and yet have a model computationally simpler than two equation models [15]. As the previous model, this have been largely tested modeling external aerodynamic flows such as transonic flow over airfoils, including boundary-layer separation. According to Godin P.,[9], the model provided quite successful results, but also showed limitations as other models in predicting boundary-layer separation due to adverse pressure gradients.

The equations used for this model are presented below.

$$\nu_t = \tilde{\nu} f_{v1} , \quad (3.1.9)$$

where

$$\begin{aligned} \frac{\partial \tilde{\nu}}{\partial t} + u_j \frac{\partial \tilde{\nu}}{\partial x_j} = & c_{b1} (1 - f_{t2}) \tilde{S} \tilde{\nu} - \left( c_{\omega 1} f_{\omega} - \frac{c_{b1}}{\kappa^2} f_{t2} \right) \left( \frac{\tilde{\nu}}{d} \right)^2 \\ & + \frac{1}{\sigma} \left\{ \frac{\partial}{\partial x_j} \left[ (\nu + \tilde{\nu}) \frac{\partial \tilde{\nu}}{\partial x_j} \right] + c_{b2} \frac{\partial \tilde{\nu}}{\partial x_i} \frac{\tilde{\nu}}{\partial x_i} \right\} , \end{aligned} \quad (3.1.10)$$

$$f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}, \quad \chi = \frac{\tilde{\nu}}{\nu}, \quad \tilde{S} = \Omega + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2}$$

and

$$\Omega = \sqrt{2W_{ij}W_{ij}}, \quad W_{ij} = \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} \right), \quad (3.1.11)$$

$\Omega$  is the magnitude of the vorticity  $W_{ij}$ ,  $d$  is the distance from the field point to the nearest wall. The parameters in the model equations for  $\tilde{\nu}$  are shown in table [3.1]

$f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}}$	$f_{\omega} = g \left[ \frac{1 + c_{\omega3}^2}{g^6 + c_{\omega3}^6} \right]^{1/6}$	$C_{b1} = 0.1355$	$c_{\omega3} = 2$
$g = r + c_{\omega2} (r^6 - r)$	$r = \min \left[ \frac{\tilde{\nu}}{\tilde{S} \kappa^2 d^2}, 10 \right]$	$C_{b2} = 0.622$	$\kappa = 0.41$
$c_{\omega1} = \frac{c_{b1}}{\kappa^2} + \frac{1 + c_{b2}}{\sigma}, \sigma = 2/3$	$\sigma_{\omega2} = 0.856$	$c_{v1} = 7.1$	$c_{t3} = 1.2$
$f_{t2} = c_{t3} \exp(-c_{t4} \chi^2)$	$c_{v2} = 5$	$c_{\omega2} = 0.3$	$c_{t4} = 0.5$

Table 3.1: Parameters in the model equation for  $\tilde{\nu}$

### 3.1.3

#### k- $\omega$ SST Model

The  $k - \omega$  SST model provides computation for  $u^*$  and  $l^*$  using the turbulent kinetic energy  $k$  and the vorticity  $\omega$  (specific dissipation rate) respectively. Menter in 1994 [5] proposed this model aiming at obtaining accurate prediction of aeronautic flows with strong pressure gradients and separation. Firstly he proposed to modify the definition of the turbulent viscosity to account for the effect of the transport of the principal turbulent shear stress. And secondly utilizing a zonal formulation, in which the original  $k - \omega$  model of Wilcox is used in the inner region of the boundary-layer, while a high Reynolds definition of the  $k - \varepsilon$  model is used in the outer region.

As a result, the performance of the mentioned models were improved by avoiding shortcomings such as the free-stream turbulence dependency of the  $k - \omega$  model and the inability of the  $k - \varepsilon$  model to deal with separated airfoil flows [25].

To meet that, the formulation of this model uses a blending function in order to shift from one model to the other. The blending function is designed to be one in the near-wall region, which activates the  $k - \omega$  model, and zero away from the surface, which activates the  $k - \varepsilon$  model. The complete formulation used for this model is presented below.

The kinematic turbulent eddy viscosity is defined as follows:

$$\nu_t = \frac{a_1 k}{\max(a_1 \omega, S F_2)} , \quad (3.1.12)$$

where  $S$  is the invariant measure of the mean strain rate and  $F_2$  is a second blending function:

$$\frac{\partial(\rho \bar{u}_i k)}{\partial x_i} = P_k - \beta^* \rho k \omega + \frac{\partial}{\partial x_i} \left[ \left( \mu + \frac{\mu_t}{\sigma_k} \right) \frac{\partial k}{\partial x_i} \right] , \quad (3.1.13)$$

where  $k$  is the turbulence kinetic energy, defined as  $k = \frac{1}{2} (\bar{u}'^2 + \bar{v}'^2 + \bar{w}'^2)$  the production of  $k$  is

$$P_k = \mu_t \frac{\partial \bar{u}_i}{\partial x_j} \left( \frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) , \quad (3.1.14)$$

$$\begin{aligned} \frac{\partial(\rho \omega)}{\partial t} + \frac{\partial(\rho \bar{u}_i \omega)}{\partial x_i} &= \frac{\gamma}{\nu_t} P_k - \beta \rho \omega^2 + \frac{\partial}{\partial x_i} \left[ (\mu + \sigma_\omega \mu_t) \frac{\partial k}{\partial x_i} \right] \\ &\quad + 2(1 - F_1) \rho \sigma_{\omega 2} \frac{1}{\omega} \frac{\partial k}{\partial x_i} \frac{\partial \omega}{\partial x_i} , \end{aligned} \quad (3.1.15)$$

where the blending function  $F_1$  is defined by:

$$F_1 = \tanh \left\{ \left\{ \min \left[ \max \left( \frac{\sqrt{k}}{\beta^* \omega y}, \frac{500 \nu}{y^2 \omega} \right); \frac{4 \rho \sigma_{\omega 2} k}{C D_{k\omega} y^2} \right] \right\}^4 \right\} , \quad (3.1.16)$$

with  $C D_{k\omega} = \max \left( 2 \rho \sigma_{\omega 2} \frac{1}{\omega} \frac{\partial k}{\partial x_i} \frac{\partial \omega}{\partial x_i}, 10^{-10} \right)$  and  $y$  is the distance to the nearest wall.

$$F_2 = \tanh \left\{ \left[ \max \left( \frac{2 \sqrt{k}}{\beta^* \omega y}, \frac{500 \nu}{y^2 \omega} \right) \right]^2 \right\} , \quad (3.1.17)$$

All the constants are computed by a blend from the corresponding constants of the  $k - \varepsilon$  and the  $k - \omega$  model via  $\alpha = \alpha_1 F + \alpha_2 (1 - F)$ . The constant for this models are presented below:

$\beta^* = 0.09$	$\gamma_1 = 5/9$	$\beta_1 = 3/40$
$\sigma_{k1} = 0.85$	$\sigma_{\omega 1} = 0.5$	$\gamma_2 = 0.44$
$\beta_2 = 0.0828$	$\sigma_{k2} = 1$	$\sigma_{\omega 2} = 0.856$
$a_1 = 0.31$		

Table 3.2: Parameters in the model equations for  $k$  and  $\omega$

### 3.2

#### Near-Wall Treatment

Turbulence models are usually derived to work outside the viscous sub-layer, so the effects of the walls are not taken into account during the solution of the transport equations. To circumvent this issue, most of the turbulence models have to be modified in order to allow them to represent the near-wall region. The modifications gave rise to a new classification of the turbulence models.

Low Reynolds models, which calculate the flow up to the wall using damping functions that reduce the effect of turbulence as the solution moves towards the wall. They require a very fine grid resolution, consequently, it is computationally more expensive and restrictive for most of the industrial applications.

High Reynolds models, which use wall functions to approximate the near wall flow by analytically calculating the boundary-layer in the near wall region from equations fitted to extensive experimental data. Since the flow near the wall is not solved, using wall functions reduces notably the quantity of cells used in the simulations and thus the computationally cost required to solve the equations. The use of standard wall functions is limited by several reasons. Most importantly, they do not hold for adverse pressure gradient, rough surfaces and for separated flows, what can result in spurious values in the solutions. However, it is possible to implement modified wall functions to account for the mentioned phenomena in order to improve the performance of the wall function.

### 3.3

#### Discretization

Since it is not always possible to find solutions for the Navier-Stokes equations, which can represent the continuous values of  $u, p, \rho$  in space and time, the continuous values are approximated by discrete expressions prescribing values only at a finite number of points.

The process by which these discrete expressions are obtained is named discretization. Different methods can be used to carry out this process, namely, Finite Difference Method (FDM), Finite Elements Method (FEM) or Finite Volume Method (FVM). Each of them uses its own schemes to approximate the equation's terms and divide the domain in smaller regions, usually trading complexity and computational cost for accuracy. The OpenFoam code uses a finite volume method for structured and unstructured grids.

**Spatial discretization.** The domain is divided into cells, resulting in a computational grid in which the solution of the governing equations is carried out.

**Governing equations discretization.** The set of governing equations (Navier-Stokes equations) are transformed into algebraic equations that represent the physical problem in the discrete points of the grid.

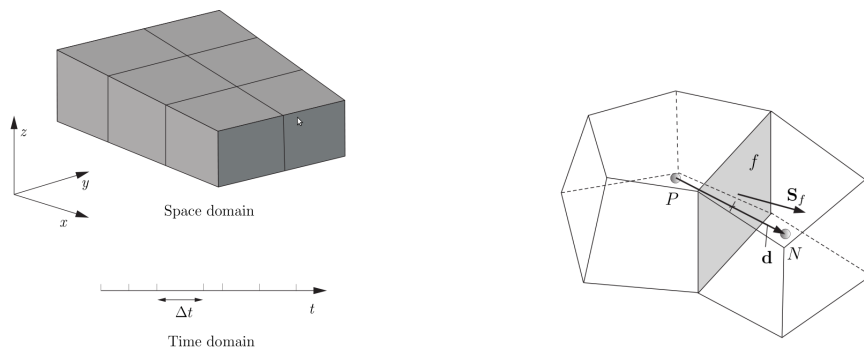
**Temporal discretization.** To account for transient effects, every term in the differential equations is integrated over time, so the time dimension is divided into a set of discrete time steps.

### 3.3.1

#### Spatial discretization

The resulting computational grid stores cells identification and relations between them. The cells define not only the spatial localization, but also allow the definition of the spatial derivatives on the grid points.

OpenFoam works with three dimensional polyhedral cells, holding the following features: The cells must be contiguous, i.e., they do not overlap each other and completely fill the domain. Each cell is bounded by a set of flat faces, given the generic label  $f$ . Faces can be internal or boundary faces; internal faces connect two cells inside the mesh and the boundary faces belongs to internal cells as well as the boundary of the domain. An example of the space, time domain and the geometric parameters involving two finite volumes are shown in Fig. [3.1].



a) Discretization of the domain

b) Cell geometric parameters

Figure 3.1: Discretization parameters [16]

The tool used to get the domain discretized is the utility supplied by OpenFoam named **snappyHexMesh**. As was explained this utility generates three-dimensions unstructured grids automatically, containing hexahedra and split-hexahedra, from a triangulated surface geometry in Stereolithography (STL) format.

The grid generation process on the (STL) geometry begins by creating a background hexagonal mesh that fills the entire domain as shown in Fig. [3.2]. This background mesh can be created using the **blockMesh** utility, provided that the following criteria are met.

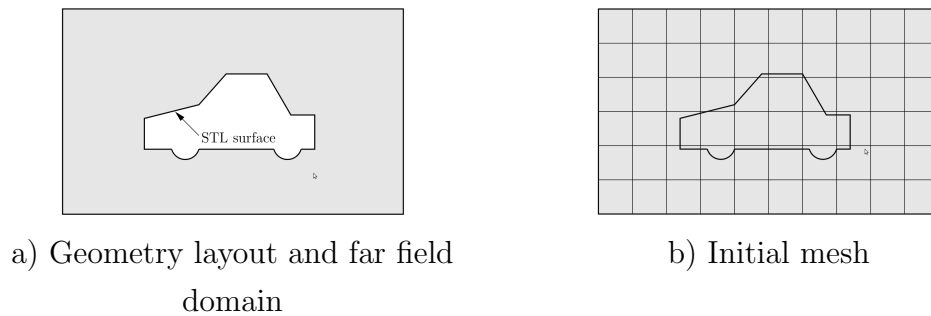


Figure 3.2: Grid generation step one and two [16]

- The mesh must consist purely of hexahedral cells.
- The cell aspect ratio should be approximately 1 to ensure the convergence of the snapping procedure.
- There must be at least one intersection of a cell edge with the STL surface.

After the definition of the background mesh, the domain's cells are split as shown Fig. [3.3] part (a), beginning at the STL surfaces according to a given refinement level. The level of refinement is defined through a minimum and a maximum value, which represents the number of times the initial cells are divided in two. The minimum value is applied generally across the surface and the maximum value is only applied in areas with larger curvature. Once the feature and surface splitting is complete a process of cell removal begins as shown in Fig. [3.3] part (b). Cell removal requires one or more regions enclosed entirely by a bounding surface within the domain. The region in which cells are retained are simply identified by a location vector within that region.



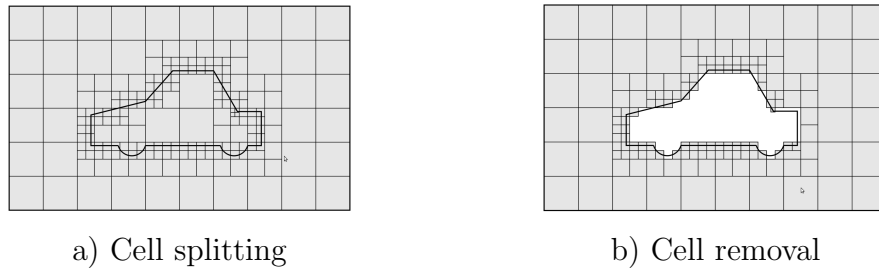


Figure 3.3: Grid generation step three and four [16]

In the next stage, to remove the jagged surface that remains from the later stage, the cell vertex points are snapped to the surface to sketch the original surface form as shown in Fig. [3.4] part (a). An additional stage that can help to get better results in high Reynolds simulations is the addition of layers of hexahedral cells around the geometry surface as shown in Fig. [3.4] part (b). This layers consist of cells aligned to the boundary surface representing the boundary-layer over the surface. The number of layers and its distributions is an important issue that has to be controlled when setting up the layers growth in order to achieve the  $y^+$  value desired for the first cell, according to each kind of wall treatment.

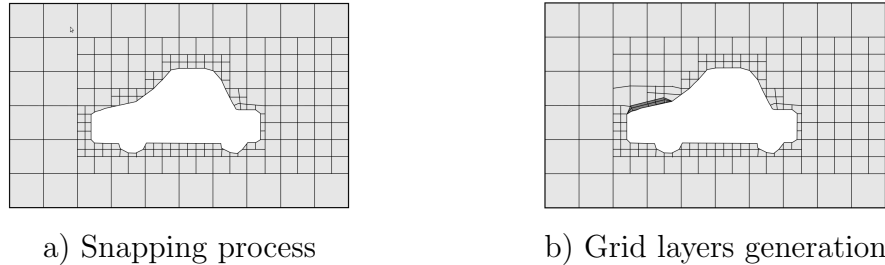


Figure 3.4: Grid generation step five and six [16]

### 3.3.2

#### Governing Equations Discretization

The Navier-Stokes equations (3.1.1) are discretized using the Finite Volume Method. In this method, the discretization of each term is formulated by first integrating the term over the cell, or control volume  $V$ . Volume and surface integrals are then linearized using appropriate schemes to be read during each job run-time. To begin the discretization process the volume integrals are transformed into surface integrals using the Gauss theorem, allowing to evaluate the values at the faces rather than at the volumes. The discretization of the different terms is carried out as follows:

$$\int_V \frac{\partial u_i}{\partial t} + \int_V \frac{\partial u_i u_j}{\partial x_j} = \int_V -\frac{1}{\rho} \frac{\partial p}{\partial x_i} + \int_V \nu \frac{\partial^2 u_i}{\partial x_j \partial x_j} , \quad (3.3.1)$$

### Convective Term

$$\int_V \vec{\nabla} \cdot (\rho \vec{u} \vec{u}) dV = \int_S d\vec{S} \cdot (\rho \vec{u} \vec{u}) \approx \sum_f \vec{S}_f \cdot (\rho \vec{u})_f \vec{u}_f , \quad (3.3.2)$$

where  $\vec{u}_f$  is evaluated using second order UPWIND scheme.

### Diffusive Term

$$\int_V \vec{\nabla} \cdot (\nu \nabla \vec{u}) dV = \int_S d\vec{S} \cdot (\nu \nabla \vec{u}) \approx \sum_f \nu_f \vec{S}_f \cdot (\nabla \vec{u})_f , \quad (3.3.3)$$

where the dot product can be calculated as follows:

$$\vec{S}_f \cdot (\nabla \vec{u})_f = \left| \vec{\Delta} \right| \frac{u_N - u_P}{\left| \vec{d} \right|} + \vec{k} \cdot (\nabla \vec{u})_f , \quad (3.3.4)$$

where the first term on the right is the orthogonal contribution and the second term is the non-orthogonal corrector.  $\vec{S}_f$  is the face area vector, and  $\left| \vec{d} \right|$  is the distance between two adjacent nodes as shown in Fig. [3.1]. The vector  $\vec{S}_f$  is decomposed in two components  $\vec{S}_f = \vec{\Delta} + \vec{k}$ , where the vector  $\vec{\Delta}$  is chosen to be parallel with  $\vec{d}$  and  $\vec{k}$  is the other component. P and N are the points around the face  $f$  and  $(\nabla \vec{u})_f$  is calculated as follows.

$$(\nabla \vec{u})_f = f_x (\nabla \vec{u})_P + (1 - f_x) (\nabla \vec{u})_N , \quad (3.3.5)$$

where  $f_x$  is an interpolation factor

### Pressure Term

$$\int_V \nabla p dV = \sum_f \vec{S}_f p , \quad (3.3.6)$$

### 3.3.3

#### Boundary Conditions

In order to solve the governing equations, boundary conditions have to be specified on all boundary faces. The boundary conditions used to set the simulation are two types:

- **Dirichlet** prescribes the value of the dependent variable on the boundary and is therefore termed fixed value.

- **Neumann** prescribes the gradient of the variable normal to the boundary and is therefore termed fixed gradient.

To specify the real boundary conditions of a flow, the later numerical description of the boundary conditions are applied differently on each boundary surface. In incompressible fluid flow there are the following physical boundaries.

- **Inlet** The velocity field at the inlet is supplied and, for consistency, the boundary condition on pressure is zero gradient.
- **Outlet** The pressure field at the outlet is supplied and a zero gradient boundary condition on velocity is specified.
- **No-slip, impermeable wall** The velocity of the fluid is equal to that of the wall itself, i.e., a fixed value condition can be specified. The pressure is specified zero gradient since the flux through the wall is zero.
- **Symmetry plane.** The symmetry plane condition specifies the component of the gradient normal to the plane should be zero.
- For turbulent flows, the inlet and outlet boundary conditions on turbulence variables ( $k$  and  $\omega$ , for example) are typically assigned to fixed values and zero gradients, respectively. The boundary conditions for the turbulence properties on the wall depend on the form of the selected turbulence model and the near-wall treatment [7].

### 3.3.4

#### Calculation of the flow field

The systems of partial differential equations are treated in the segregated way, meaning that they are solved one at a time, with the inter-equation coupling treated in the explicit manner. All dependent variables share the same control volumes, which is usually called the co-located or non-staggered variable arrangement.

The form of the presented equations shows a linear dependence of velocity on pressure and vice-versa. So, to solve the problem of pressure-velocity coupling, the Semi-Implicit Method for Pressure-Linked Equations (SIMPLE) is used.

The SIMPLE algorithm by Patankar [3] is formulated as follows:

1. Guess the pressure field.
2. Solve the momentum equations to obtain the velocity field.
3. Solve the pressure correction equation.

4. Calculate the pressure using the pressure correction and the guessed pressure.
5. Calculate the velocity field using the velocity-correction formulas.
6. Solve the discretization equations for the other magnitudes.
7. Treat the corrected pressure as a new guessed pressure, return to step two, and repeat the whole procedure until a converged solution is obtained.