CHAPTER 4

FUNDAMENTALS OF COMPUTATIONAL FLUID DYNAMICS

In this chapter, a general description of mathematical modeling using computational fluid dynamics (CFD) is presented. The numerical approximation to the conservation equations is presented in accordance with the form used by the commercial CFD code "PHOENICS." The modifications required to adapt PHOENICS for the analysis of momentum and energy transfer in thermal sterilization are also presented.

4.1. INTRODUCTION TO COMPUTATIONAL FLUID DYNAMICS

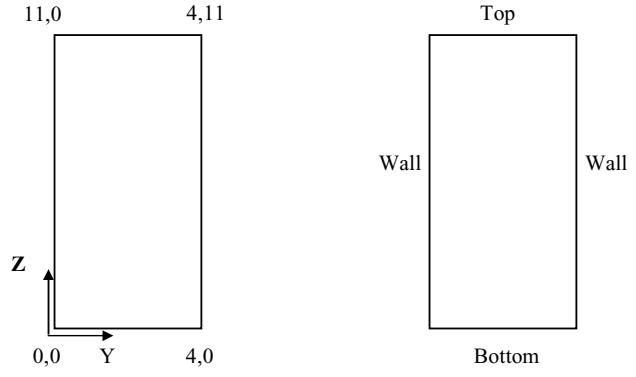
CFD is a numerical technique used for the solution of the equations governing fluid flow and heat transfer problems inside a defined flow geometry (Scott, 1994). CFD has wide applications in the areas of fluid and heat transfer within the aerospace and nuclear industries backed by the availability of powerful supercomputers. It has expanded into other industries such as the chemical and petrochemical industries. It is only in recent years that it has been applied to the food industry, with a limited variety of food-related problems being investigated (Scott and Richardson, 1997).

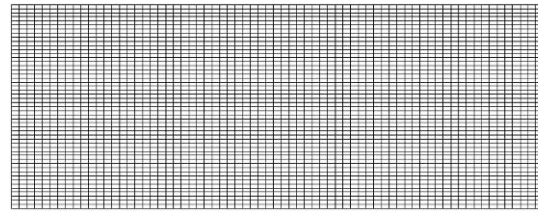
CFD offers a powerful design and investigative tool to the process engineer in many applications. However, at present little use of this technology has been reported in the food industry. Its application in such areas would be beneficial for the better understanding of the complex interactions occurring in food systems. The development of CFD packages came from a need to solve complex fluid flow problems of a general nature for a wide range of geometry and boundary conditions (Hatton and Carpenter, 1976). CFD works by dividing the physical environment of interest into a two or three-dimensional (3-D) grid or mesh. It contains a number of discrete cells and can evaluate fluid velocities, temperature, and pressure inside every one of the cells where fluid flows. This is done by the simultaneous solution of the equations describing fluid flow, heat, and mass transfer.

The use of CFD techniques to solve a fluid flow and heat transfer problem is split into three discrete parts: *pre processing, processing, and post processing*. In general, different computer programs that form the CFD code must undertake each of the three tasks.

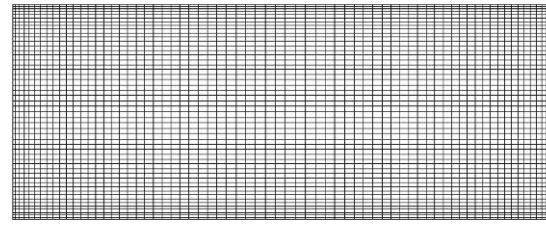
4.1.1. Definition of a CFD Problem (Preprocessor)

The first stage in solving a CFD problem is to define all the relevant parameters required by the CFD code prior to the numerical solution process, as follows:





(a) Orthogonal structured mesh with uniform spacing (50x70)



(b) Orthogonal structured mesh with non-uniform spacing (50x70)

Construction of a computational grid

Figure 4.1. Defining geometry for a CFD simulation.

1. Definition of the physical geometry of the environment in which the fluid flows, which is normally done by building up a geometric representation of the environment. "Create the shape of the problem domain that needs to be analysed."
2. Declaration of the boundary conditions of the physical environment. These boundary conditions will include defining certain areas such as the inlets and outlets for the fluid flow and the boundary areas of solids where heat transfer from or to the fluid can occur.
3. Construction of a mesh or grid as a geometric representation of the physical environment. This mesh or grid will form the computational grid that will be used in the solution of the problem by the powerful mathematical techniques around which CFD is based.

A uniform spacing mesh has regular rectangular cells with second-order accuracy. A nonuniform mesh (Figure 4.1) provides smaller cells where rapid changes are taking place within the solution domain (e.g., close to the boundaries); however, it is only of first-order accuracy (Mallinson, 1999).

In any CFD problem, the definition of boundary conditions is essential because they will define the flow conditions inside the flow domain. In general, at any top, bottom, inlets, or outlets, the flow domain needs to be defined to set the appropriate boundary conditions. For example, in the case of heating a fluid in a can, the velocity of the fluid and its temperature are set at the top, bottom, and wall of the can. Other types of boundary conditions that may need to be set are those for walls and solids particularly if they act as heat transfer boundaries (Scott and Richardson, 1997).

After defining the computational mesh and boundary conditions, the user needs to define the assumptions need to be made—for example, whether the flow is laminar or turbulent and whether heat transfer takes place or not. This usually involves writing a command file in PHOENICS that ties any other information required by the CFD solver, such as the type of model used and the number of iterations required.

There are two methods of generating the grid and the setup data for any problem. The first is to define the problem using FORTRAN. The second and more elegant method is based on a graphical user interface. In this method, the data is entered via a menu-driven procedure, which guides the user through all the separate stages in setting up a CFD simulation.

All CFD packages offer a variety of coordinate systems from which the computational grid is defined and generated, such as strict Cartesian, cylindrical polar, curvilinear, body fitted, or moving/rotating coordinate systems. The choice of the coordinate system, the resolution of the computational grid, and the method of its generation will be dependent on the complexity of the simulation.

4.1.2. Solution of the Problem (Processor)

The solution of the problem by the CFD code is where a host of mathematical techniques is used to approximate the differential equations into algebraic form, which can be solved directly or iteratively. Different CFD codes employ different solution techniques, but the physics is the same if it can be well defined and understood. The solution of the transport equations for the geometry under study is not a trivial matter and cannot be solved readily, if at all, by analytical techniques. CFD uses numerical techniques to solve discretized representations of the transport equations.

Direct or explicit numerical methods, which can be both extremely accurate and rapid, may be used if sufficient computing power is available. Many codes use iterative methods to solve the equations because they tend to be more robust, although they can take longer to converge. Standard texts are available that provide good background material on numerical simulation and CFD.

4.1.3. Analysis of the Results (Postprocessor)

The results can be analyzed both numerically and graphically. The postprocessor takes the numerical results and displays them as a visual representation. It displays a visual image of the physical geometry through which the fluid flows, with the option of printing a hard copy of all the results as tables of numbers and other means. It is possible to superimpose the velocity, pressure, and temperature distributions within the fluid. The format of this display is a graphical contour with the option of displaying scaled arrows for vector quantities. The output file can contain all sorts of information, including the spatial coordinates of all of the cells in the computational mesh and the solved transport variables for each cell. In the case of a large CFD problem, say greater than 100,000 cells, it is obvious that the user does not want to read through this file. Thus, the second method offers the user the ability to visualize the results. This method, often referred to as postprocessing, takes the results from the CFD solver and allows the user to display variables graphically on the computer screen, for all or part of the flow domain. These graphical presentation methods include vector plots

(a scaled arrow pointing in the direction of flow), contour plots on a two-dimensional (2-D) slice through the domain, and iso-surface plots (a 3-D surface on which a variable is constant). The user has the option to rotate the image in 3-D space or to zoom into areas of interest to extract the most useful information from the image. Combining both visual and numerical results allows the optimal solution to be achieved for the problem under investigation.

In this work, PHOENICS, a commercial CFD package developed by Concentration, Heat and Momentum Limited (CHAM), London, was used to solve the equations governing natural convection in liquid canned food during thermal sterilization. PHOENICS is a well-known CFD code. It is a computer code, which simulates fluid flow, heat transfer, chemical reactions, and related phenomena. It uses the finite volume method (FVM), which is one of several computational methods used for solving heat transfer and fluid flow problems. The details of this code and the various options and values of significant parameters are discussed in the PHOENICS Reference Manual.

For solving the equations governing natural convection, Nield and Bejan (1992), after examining a range of numerical techniques, concluded that the FVM is more appropriate than other methods of solution.

4.2. FINITE VOLUME METHOD AND PARTICULAR FEATURES OF PHOENICS

The FVM, which is also known as the finite domain method, is one of several computational methods for solving fluid flow among many other problems. The principle behind this numerical method is based on the control volume idea used in many fluid texts. This idea is applied on a cell basis and used to derive the conservation equations of mass, momentum, and energy from basic laws into a mathematical form known as finite volume equations (FVE).

In order to specify a problem, it is necessary to identify the computational domain, which totally covers the region of flow to be studied. The computational domain may contain some sections where there is no fluid flow since these may be blocked. The computational domain must then be subdivided into a number of divisions in the three dimensions (i.e. filled with NX $\times$ NY $\times$ NZ cells). In general, topologically Cartesian grids take three forms which are as follows:

a. *Strict Cartesian*
   A Cartesian grid is composed of cells formed by the intersection of three sets of mutually perpendicular parallel planes, on any one of which either x, y, or z is constant, these quantities being the distances in the three coordinate directions.
b. *Cylindrical (polar) grids*
   A cylindrical (polar) grid consists of cells formed by the intersection of
   - planes of constant z perpendicular to the axis of rotation
   - planes of constant x, which all pass through and thus intersect on that axis so that x now represents an angle in radius and not distance
   - concentric cylindrical surfaces of constant radial coordinate y
c. *Body-fitted coordinates (BFC)*
   A BFC is best imagined by supposing that a regular Cartesian grid is first embedded in a jellylike medium, which is then squeezed, bent, and twisted in an arbitrary way. All the cells in contact with another remain so.
   1. The position of each grid corner must be stored, which requires the storage of $3 \times (\mathrm{NX} + 1) \times (\mathrm{NY} + 1) \times (\mathrm{NZ} + 1)$ pieces of information, which represents the number of grids in the different direction values of I, J, and K for each corner in the cell.

2. The velocities are directed along lines joining cell nodes and are therefore perpendicular to the cell faces for strict Cartesian and cylindrical polar grids. In body-fitted grids, this is not necessarily true, and hence careful calculation of flows across faces is required.
3. In body-fitted grids, some form of a grid generation technique is required in order to specify the grid corner positions.

In cylindrical polar and body-fitted grids, the ideas behind the FVM are the same as those for strict Cartesian grids, but the mathematics is harder.

The FVM may treat time in a manner that is similar to that used for the other dimensions with a Cartesian grid. This means that the time dimension is subdivided into NT discrete time planes at which a solution is obtained. These may be spaced in any desired manner but should obviously be concentrated at times when the flow is changing rapidly, such as the fine mesh time steps used in our work at the beginning of heating. Three-dimensional notation will be discussed here for the geometry of the can and pouch being investigated in this book. Grid nomenclature used in PHOENICS is shown in Figure 4.2.

For the cell with its node at P, which adjoins cells with their nodes at E, W, N, S, H, and L (east, west, north, south, high, and low), the centers of the adjoining faces are e, w, n, s, h, and l having face areas of A_e, A_w, A_n, A_s, A_h, and A_l, respectively.

Most properties are stored as values at the cell nodes, while velocities are stored at the cell faces. This has computational advantages both in terms of stability and computational ease since the velocity related to flow across a face is specified at its center. This is done by assuming a staggered grid passing through the cell nodes and having the velocity at the face centers (e, w, n, s, h, and l), as shown in the dashed lines in Figure 4.3 for the XY plane.

In the course of defining the computational domain, the optimum meshing arrangement should be decided upon. The geometrical and time mesh subdivides the computational domain into small cells at which values of the medium properties, and solved variables, are stored (Mallinson, 1999).

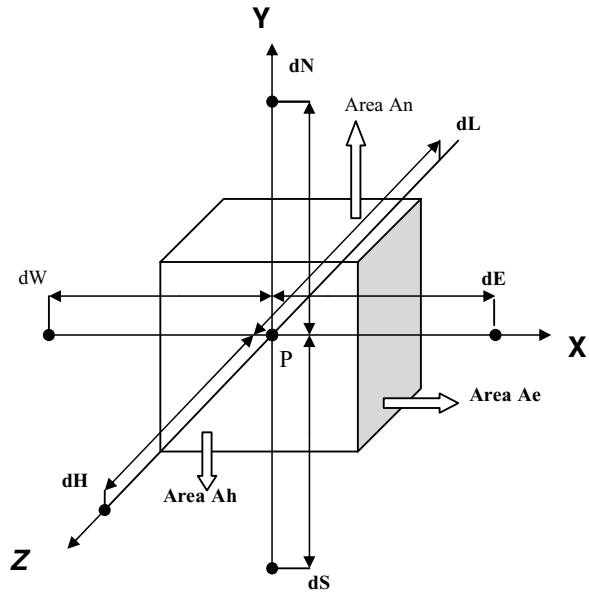


Figure 4.2. Cell nomenclature showing cell nodes and staggered grid.

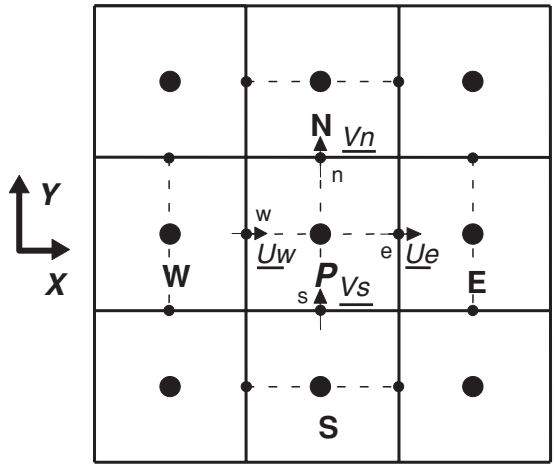


Figure 4.3. Cell nomenclature showing staggered grid.

Several meshing arrangements will be considered throughout this work. Most of the cases presented in this work were checked using both mesh types (uniform and nonuniform spacing) and for a different numbers of cells to prevent any discretization error and also to make sure that the results were consistent.

4.2.1. The Conservation Equations

In order to obtain a solution to a fluid flow problem, it is necessary to relate what is happening within one cell to that that is happening in its neighbours. This is achieved through the principles of conservation of mass, momentum, energy, etc. The basic principle of conservation as applied to a control volume is "The change in what is contained within a control volume must have come from or gone somewhere."

When deriving the conservation equation for cell P, the net change within the cell equals the net convection transfer plus the net diffusive transfer and any other sources. The transfer process is subdivided into three categories.

Convection:	The net amounts of a conserved property carried into or taken out of the control volume by the movement of fluid across the control surface (the cell faces)
Diffusion:	The transfer of a conserved property by diffusion
Other sources:	Sources from outside the solution domain as well as terms such as the pressure gradient–related source of momentum

The conservation equation for a control volume is therefore

$$\text{Net change within the control volume} = \text{Net convective transfer} + \text{Net diffusive transfer} + \text{Sources}$$

4.2.1.1. Conservation of Mass (Continuity)

In the software used in our work, we have only three components in the equation of conservation of mass:

1. The net change of mass in cell P during a time interval Δt is $\rho_P \forall_P - \rho_T \forall_T$, where subscript P denotes the present value and T denotes the previous value.
2. The net convective mass transfer through the faces of the cell is

$$(\rho_w A_w U_w - \rho_e A_e U_e + \rho_s A_s V_s - \rho_n A_n V_n + \rho_l A_l W_l - \rho_h A_h W_h)\,\delta t.$$

3. The net transfer from external sources, such as sources or sinks, is $m_{\text{source}}\delta t$.

Then the conservation of mass equation will be

$$\begin{aligned}\rho_P \forall_P &= \rho_T \forall_T + (\rho_w A_w U_w - \rho_e A_e U_e + \rho_s A_s V_s - \rho_n A_n V_n + \rho_l A_l W_l \\ &\quad - \rho_h A_h W_h + m_{\text{source}})\,\delta t \end{aligned} \tag{4.1}$$

4.2.1.2. Conservation of General Intensive Properties

For any general intensive property Φ (specific property used here for energy and concentration), the conservation equation can be written in a similar manner to equation (4.1):

1. The change of Φ in the cell is $\rho_P \forall_P \Phi_P - \rho_T \forall_T \Phi_T$.
2. The net convection is

$$(\rho_w A_w U_w \Phi_w - \rho_e A_e U_e \Phi_e + \rho_s A_s V_s \Phi_s - \rho_n A_n V_n \Phi_n + \rho_l A_l W_l \Phi_l - \rho_h A_h W_h \Phi_h)\,\delta t$$

3. The source of Φ associated with mass sources is $m_{\text{source}} \Phi_{\text{source}} \delta t$.
4. The other sources with no mass transfer is $\sum_f \text{CO}_f Tp_f (\text{Val}_f - \Phi_P)$.

The final term to be included is the diffusion term. In general the diffusion of Φ across any area A_f (f = e, w, n, s, h, or l) is equal to $\Gamma_f A_f \dfrac{\Phi_F - \Phi_P}{d_F} \delta t$.

If strong convection exists, the amount of diffusion is reducing. This is achieved by using

$$\text{Diffusion} = \langle 1 - D_f \rangle \Gamma_f A_f \frac{\Phi_F - \Phi_P}{d_F} \delta t$$

$$D_f = Dc\ ABS\left[\frac{\text{convection flux}}{\Gamma_f A_f / d_F}\right]$$

and where

$$D_f = Dc\ ABS\left[\frac{\text{convection flux}}{\Gamma_f A_f / d_F}\right]$$

Φ = specific property
CO = coefficient
d_F = the distance between nodes P and F (F = E, W, N, S, H, and L)
Dc = diffusion cut-off, which cuts the diffusion when convection becomes high
Tp = patch type multiplier (e.g., area or volume of the cell)
Val = value of Φ associated with the nonconvective source
$\forall$ = volume of the cell

δt = time interval
Γ = exchange coefficient = $\dfrac{\mu}{\mathrm{Pr}}$
Pr = Prandtl number = $\dfrac{\mu Cp}{k}$

Combining these five terms gives the full FVE (4.2):

$$\begin{aligned}\rho_{\mathrm{P}}\forall_{\mathrm{P}}\Phi_{\mathrm{P}} - \rho_{\mathrm{T}}\forall_{\mathrm{T}}\Phi_{\mathrm{T}} = (&\rho_{\mathrm{w}}A_{\mathrm{w}}U_{\mathrm{w}}\Phi_{\mathrm{w}} - \rho_{\mathrm{e}}A_{\mathrm{e}}U_{\mathrm{e}}\Phi_{\mathrm{e}} + \rho_{\mathrm{s}}A_{\mathrm{s}}V_{\mathrm{s}}\Phi_{\mathrm{s}} - \rho_{\mathrm{n}}A_{\mathrm{n}}V_{\mathrm{n}}\Phi_{\mathrm{n}}\\ &+ \rho_{\mathrm{l}}A_{\mathrm{l}}W_{\mathrm{l}}\Phi_{\mathrm{l}} - \rho_{\mathrm{h}}A_{\mathrm{h}}W_{\mathrm{h}}\Phi_{\mathrm{h}}\\ &+ \langle 1 - D_{\mathrm{w}}\rangle\Gamma_{\mathrm{w}}A_{\mathrm{w}}\left(\frac{\Phi_{\mathrm{W}} - \Phi_{\mathrm{P}}}{d_{\mathrm{W}}}\right) + \langle 1 - D_{\mathrm{e}}\rangle\Gamma_{\mathrm{e}}A_{\mathrm{e}}\left(\frac{\Phi_{\mathrm{E}} - \Phi_{\mathrm{P}}}{d_{\mathrm{E}}}\right)\\ &+ \langle 1 - D_{\mathrm{s}}\rangle\Gamma_{\mathrm{s}}A_{\mathrm{s}}\left(\frac{\Phi_{\mathrm{S}} - \Phi_{\mathrm{P}}}{d_{\mathrm{S}}}\right) + \langle 1 - D_{\mathrm{n}}\rangle\Gamma_{\mathrm{n}}A_{\mathrm{n}}\left(\frac{\Phi_{\mathrm{N}} - \Phi_{\mathrm{P}}}{d_{\mathrm{N}}}\right)\\ &+ \langle 1 - D_{\mathrm{l}}\rangle\Gamma_{\mathrm{l}}A_{\mathrm{l}}\left(\frac{\Phi_{\mathrm{L}} - \Phi_{\mathrm{P}}}{d_{\mathrm{L}}}\right) + \langle 1 - D_{\mathrm{h}}\rangle\Gamma_{\mathrm{h}}A_{\mathrm{h}}\left(\frac{\Phi_{\mathrm{H}} - \Phi_{\mathrm{P}}}{d_{\mathrm{H}}}\right)\\ &+ m_{\mathrm{source}}\Phi_{\mathrm{source}} + \mathrm{CO}Tp(\mathrm{Val} - \Phi_{\mathrm{P}})\delta t\end{aligned} \tag{4.2}$$

For the FVE to be solved in all the cells and for all the intensive properties, the coefficients of Φ_{P} must be coupled with all Φ_{F} at the nearby cells. This is performed by substituting Equation (4.1) into Equation (4.2) to eliminate $\rho_{\mathrm{P}}\forall_{\mathrm{P}}$ and then dividing by δt to get the FVE (4.3) of the form

$$\begin{aligned}0 = &\frac{\rho_{\mathrm{T}}\forall_{\mathrm{T}}}{\delta t}(\Phi_{\mathrm{T}} - \Phi_{\mathrm{P}})\\ &+ \rho_{\mathrm{w}}A_{\mathrm{w}}U_{\mathrm{w}}(\Phi_{\mathrm{w}} - \Phi_{\mathrm{P}}) - \rho_{\mathrm{e}}A_{\mathrm{e}}U_{\mathrm{e}}(\Phi_{\mathrm{e}} - \Phi_{\mathrm{P}})\\ &+ \rho_{\mathrm{s}}A_{\mathrm{s}}V_{\mathrm{s}}(\Phi_{\mathrm{s}} - \Phi_{\mathrm{P}}) - \rho_{\mathrm{n}}A_{\mathrm{n}}V_{\mathrm{n}}(\Phi_{\mathrm{n}} - \Phi_{\mathrm{P}})\\ &+ \rho_{\mathrm{l}}A_{\mathrm{l}}W_{\mathrm{l}}(\Phi_{\mathrm{l}} - \Phi_{\mathrm{P}}) - \rho_{\mathrm{h}}A_{\mathrm{h}}W_{\mathrm{h}}(\Phi_{\mathrm{h}} - \Phi_{\mathrm{P}})\\ &+ \langle 1 - D_{\mathrm{W}}\rangle\Gamma_{\mathrm{W}}A_{\mathrm{W}}\left(\frac{\Phi_{\mathrm{W}} - \Phi_{\mathrm{P}}}{d_{\mathrm{W}}}\right) + \langle 1 - D_{\mathrm{e}}\rangle\Gamma_{\mathrm{e}}A_{\mathrm{e}}\left(\frac{\Phi_{\mathrm{E}} - \Phi_{\mathrm{P}}}{d_{\mathrm{E}}}\right)\\ &+ \langle 1 - D_{\mathrm{s}}\rangle\Gamma_{\mathrm{s}}A_{\mathrm{s}}\left(\frac{\Phi_{\mathrm{S}} - \Phi_{\mathrm{P}}}{d_{\mathrm{S}}}\right) + \langle 1 - D_{\mathrm{n}}\rangle\Gamma_{\mathrm{n}}A_{\mathrm{n}}\left(\frac{\Phi_{\mathrm{N}} - \Phi_{\mathrm{P}}}{d_{\mathrm{N}}}\right)\\ &+ \langle 1 - D_{\mathrm{l}}\rangle\Gamma_{\mathrm{l}}A_{\mathrm{l}}\left(\frac{L_{\mathrm{L}} - \Phi_{\mathrm{P}}}{d_{\mathrm{L}}}\right) + \langle 1 - D_{\mathrm{h}}\rangle\Gamma_{\mathrm{h}}A_{\mathrm{h}}\left(\frac{\Phi_{\mathrm{H}} - \Phi_{\mathrm{P}}}{d_{\mathrm{H}}}\right)\\ &+ m_{\mathrm{source}}(\Phi_{\mathrm{source}} - \Phi_{\mathrm{P}}) + \mathrm{CO}Tp(\mathrm{Val} - \Phi_{\mathrm{P}})\end{aligned} \tag{4.3}$$

By normalizing the influence coefficients

$$a_{\mathrm{T}} = \frac{\rho_{\mathrm{T}}\forall_{\mathrm{T}}}{\delta t}$$

$$a_{\mathrm{F}} = \rho_{\mathrm{f}}A_{\mathrm{f}}\langle V_{\mathrm{f}}\rangle + \langle 1 - Dc_{\mathrm{f}}\rangle\frac{\Gamma_{\mathrm{f}}A_{\mathrm{f}}}{d_{\mathrm{F}}}$$

$$a_{\mathrm{P}} = a_{\mathrm{T}} + \sum_{\mathrm{F}} a_{\mathrm{F}} + m_{\mathrm{source}} + \mathrm{CO}Tp$$

where V_f is the velocity directed into the cell and F = W, E, S, N, L, and H, the FVE (4.3) becomes much simpler as follows:

$$a_P\Phi_P = a_T\Phi_T + \sum_F a_F\Phi_F + m_{source}\Phi_{source} + COTpVal$$

or

$$a_P\Phi_P = a_T\Phi_T + \sum_F a_F\Phi_F + b$$

4.2.1.3. Conservation of Momentum

The momentum equation is approximated in a manner similar to the mass and energy equations, except that the velocity is stored at the cell faces rather than the cell nodes, using the staggered grids (Figure 4.3) mentioned earlier. Simple mathematical interpolation must be carried out to obtain the values of fluxes and properties at these faces.

In addition to possible externally applied forces, the momentum equations also need a pressure-related source (generally automatically built into codes such as PHOENICS), for example for the V_s velocity, the momentum source for these terms has the form $A_s(P_P - P_S)$.

This is simple enough if the pressure is known, but so far we do not have an equation for the pressure. If the pressure is unknown, an additional equation of pressure (pressure correction equation) should be considered that affects all momentum equations and interlinks the three velocity components through the pressure.

To update the pressure, the momentum equations are solved using the in-store pressure field P^*. Then we can compute the error in the conservation Equation (4.1) as a rate of gaining or losing mass as follows:

$$\begin{aligned} R_{\dot{m}} &= \frac{\rho_P\forall_P - \rho_T\forall_T}{\delta t} \\ &\quad - (\rho_w A_w U_w - \rho_e A_e U_e + \rho_s A_s V_s - \rho_n A_n V_n + \rho_l A_l W_l - \rho_h A_h W_h) - \dot{m}_{source} \end{aligned}$$

If $R_{\dot{m}} > 0$ then the cell is gaining mass and the pressure needs to increase to counteract the flow and drive it out of the cell. If the cell is losing mass, $R_{\dot{m}} < 0$, dropping the pressure will induce mass inflow. The net change in the rate of gaining or losing mass induced by modifying the pressure field is calculated from

$$\begin{aligned} \Delta\dot{m} &= \rho_W A_W \frac{\partial U_W}{\partial(P_W - P_P)}\left(P'_W - P'_P\right) - \rho_e A_e \frac{\partial U_e}{\partial(P_P - P_E)}\left(P'_P - P'_E\right) \\ &\quad + \rho_s A_s \frac{\partial V_s}{\partial(P_S - P_P)}\left(P'_S - P'_P\right) - \rho_n A_n \frac{\partial V_n}{\partial(P_P - P_N)}\left(P'_P - P'_N\right) \\ &\quad + \rho_l A_l \frac{\partial W_l}{\partial(P_L - P_P)}\left(P'_L - P'_P\right) - \rho_h A_h \frac{\partial W_h}{\partial(P_P - P_H)}\left(P'_P - P'_H\right) \end{aligned}$$

Note that

$$P = P^* + P'$$

where P^* is the current in-store value of the pressure (from the previous sweep) and P' is the correction needed to obtain the new pressure (P).

The partial derivatives are obtained from the momentum equation corresponding to each velocity. For example the FVE for velocity component V_s is as follows:

$$a_{PV} V_s = A_s (P_P - P_S) + \text{Other terms}$$

so that $\frac{\partial V_s}{\partial (P_P - P_S)} = \frac{A_s}{a_{PV}}$.

By normalizing the influence coefficients similar to the energy and continuity equations, we can find a correction equation for P':

$$a_P P'_P = a_W P'_W + a_E P'_E + a_S P'_S + a_N P'_N + a_L P'_L + a_H P'_H + R_{\dot{m}} \tag{4.4}$$

This equation will be solved along with Equations (4.1) and (4.2).

Having obtained the pressure correction P', there are associated velocity corrections such as

$$V_s^{'} = \frac{A_s}{a_{PV}} (P_P^{'} - P_S^{'}) \tag{4.5}$$

which should be added to the velocities.

The normalized form of all the conservation equations can be solved explicitly by using a Jacobi point-by-point solver. PHOENICS has available a Jacobi point-by-point solver, which is not generally used, but it is useful when coefficients change greatly from "sweep to sweep." This method is easy to program but is generally slow to converge and may diverge. Alternatively, the most frequently used solution in PHOENICS is the slab-wise simultaneous solver, which is fully implicit. The slab-wise solution is one of the particular features of this code. In this technique a constant Z slab is solved simultaneously, while the off-slab values are treated as temporarily known (Richards, 1998). For example, consider the situation where conditions are specified at any one time for all blocks. For a particular horizontal row i, this solution method is solved by using old values for the rows $i + 1$ and $i - 1$ immediately above and below. When the solution moves to deal with row $i + 1$, the values used for row i are the old values not the new ones. When all rows are completed all old values are replaced with new ones and another "sweep" through the rows is undertaken.

4.2.2. The Transport Equations and Related Physics

The flow of any fluid can be described using transport equations that define the conservation of mass (continuity), momentum, and energy. They are derived by considering mass, momentum, and energy balances in an element of fluid as it flows. From these the appropriate partial differential equations are derived. These balances and equations are discussed in more detail by Bird et al. (1960). The transport equations are completed by adding two algebraic equations from thermodynamics, which are the equation of state and the constitutive equation.

4.2.2.1. Equation of State

The equation of state relates the density of a fluid to its thermodynamic state (temperature and pressure). A commonly used assumption for buoyancy problems is the Boussinesq approximation (Quarini, 1995). All fluid properties are assumed to be constant except for the density. The equation

of state (Boussinesq approximation) can be written as follows:

$$\rho = \rho_{\text{ref}}[1 - \beta(T - T_{\text{ref}})] \tag{4.6}$$

where β is the thermal expansion coefficient of the liquid, and T_{ref} and ρ_{ref} are the temperature and density at reference condition.

4.2.2.2. Constitutive Equation

The constitutive equation relates the static enthalpy of a fluid to its thermodynamic state. A discussion of such relationships can be found in any standard thermodynamics textbook such as that by Smith and Van Ness (1975).

4.2.2.3. Body Force

The body force (B) depends on the type of flow. For the Boussinesq buoyancy approximation, the body force becomes $B = \rho g$. Other body forces could include rotation forces, electrostatic forces, and resistances, such as those imposed on a fluid as it flows through a porous medium.

4.2.2.4. Turbulence

The transport equations presented can be applied to both laminar and turbulent flow conditions. Where turbulent flow conditions prevail, a suitable turbulence model is required to describe the turbulence and its influence on flow conditions. Several turbulence models are available in PHOENICS that attempt to solve the time-dependent nature of turbulence flow. The inclusion of turbulence in a CFD problem makes its solution more complex because turbulence modeling is difficult.

4.2.2.5. Non-Newtonian Fluid Behavior

When considering the flow of food products, it is often necessary to take the rheological nature of food into account because this will dictate its flow behavior. Most foods exhibit some form of non-Newtonian behavior, and many different fluid models have been used to describe such behavior (Holdsworth, 1993).

NOMENCLATURE

a	normalized influence coefficient
b	source and sink term
CO	coefficient
d_{F}	distance between nodes P and F (F = S, N, L, and H).
Dc	diffusion cut-off
P^*	current in-store pressure
P'	the correction needed to obtain the new pressure
R_m	rate of gaining or losing mass
Tp	patch type multiplier (e.g., area or volume of the cell)
V	velocity in the horizontal direction (y-direction)
Val	value
W	velocity in the vertical direction (z-direction)

Greek Symbols

δ_t time interval
Φ specific property
$\forall$ volume of the cell
Γ exchange coefficient $= \frac{\mu}{\mathrm{Pr}}$

Subscripts

F S, N, L, and H for the adjoining cell in the mesh
f s, n, l, and h for the adjoining faces in the mesh
H high cell
h high face
L low cell
l low face
N north cell
n north face
P present best-known value
S south cell
s south face
source external source term
T past best-known value
o initial state

REFERENCES

Bird, R.B., Stewart, W.S., & Lightfoot, E.N. (1960). *Transport phenomena*. New York: John Wiley & Sons.

Hatton, A.P., & Carpenter, C. (1976). *Sixth thermodynamics and fluid mechanics convention* (Institution of Mechanical Engineers, pp 81–89). London and New York: Mechanical Engineering Publication.

Holdsworth, S.D. (1993). Rheological model used in the prediction of the flow properties of food products. A literature Review. *Transaction of Institute of Chemical Engineers, 71*(C4), 139–179.

Mallinson, G.D. (1999). Computational fluid dynamics lecture notes. Department of Mechanical Engineering, University of Auckland.

Nield, D.A., & Bejan, A. (1992). *Convection in porous media*. New York: Springer-Verlag.

PHOENICS Reference Manual, Part A: PIL. TR 200 A, Bakery House, London SW 19 5AU, U.K: CHAM.

Quarini, G.L. (1995). Application of computational fluid dynamics in food and beverage production. *Journal of Food Science Technology Today, 4*, 234–236.

Richards, P.J. (1998). Computational fluid dynamics lecture notes. Department of Mechanical Engineering, University of Auckland.

Scott, G.M. (1994). Computational fluid dynamics for the food industry. *Journal of Food Technology International Europe,* 49–51.

Scott, G.M., & Richardson, P. (1997). The application of computational fluid dynamics in the food industry. *Trends in Food Science & Technology, 8*, 119–124.

Smith, J.M., & Van Ness, H.C. (1975). *Introduction to chemical engineering thermodynamic*. McGraw-Hill International.