American Journal of Science and Technology
2014; 1(1): 36-49
Published online March 30, 2014 (http://www.aascit.org/journal/ajst)

Keywords
CFD,
Heat Transfer,
Fluid Flow,
SIMPLE Algorithm

Received: March 03, 2014
Revised: March 17, 2014
Accepted: March 18, 2014

The process of heat transfer and fluid flow in CFD problems

Anestis Stylianos

Department of Oenology and Beverage Technology, Technological Educational Institute of Athens, Faculty of Food and Nutrition, Greece

Email address
sanestis@teiath.gr, Anestis@essolutions.eu

Citation
Anestis Stylianos. The Process of Heat Transfer and Fluid Flow in CFD Problems. *American Journal of Science and Technology.* Vol. 1, No. 1, 2014, pp. 36-49

Abstract
In many applications we need to use the CFD technology in order to model the problem and find exact solutions with less cost. In market can be found many special CFD software solutions where the user must programming according its problem and its special situations he wants to solve. In order to formulate the starting problem in a suitable form acceptable from the CFD package must follow a variety of steps which are no so clearly. Here, we try to present all these steps, and give the basic equations per step.

1. Introduction

Importance of heat transfer and fluid flow. This paper is concerned with heat and mass transfer, fluid flow, chemical reaction, and other related processes that occur in engineering equipment, in the natural environment, and in living organisms. That these processes play a vital role can be observed in a great variety of practical situations. Nearly all methods of power production involve fluid flow and heat transfer as essential processes. The same processes govern the heating and air conditioning of buildings. Major segments of the chemical and metallurgical industries use components such as furnaces, heat exchangers, condensers, and reactors, where thermo fluid processes are at work. Aircraft and rockets owe their functioning to fluid flow, heat transfer, and chemical reaction, (Sunden, 2013). In the design of electrical machinery and electronic circuits, heat transfer is often the limiting factor. The pollution of the natural environment is largely caused by heat and mass transfer, and so are storms, floods, and fires. In the face of changing weather conditions, the human body resorts to heat and mass transfer for its temperature control. The processes of heat transfer and fluid flow seem to pervade all aspects of our life.

The purpose of this paper is primarily to aim at developing a general method of prediction for heat and mass transfer, fluid flow, and related processes. As we shall shortly see, among the different methods of prediction, the *numerical* solution offers great promise. We shall present a numerical method having complete generality for predicting the processes of interest. We shall, therefore, refrain from accepting any final restrictions such as two-dimensionality, boundary-layer approximations, and constant-density flow. If any restrictions are temporarily adopted, it will be for ease of presentation and understanding and not because of any intrinsic limitation. Numerical solution therefore becomes a desirable approach for heat conduction under non-linear, complex geometric configurations, or complex boundary conditions. Instead of obtaining the

analytical expression of the temperature distribution, the results of numerical solutions are given at discrete points. The numerical solution involves three steps: (1) discretization of the computational domain, (2) discretization of the governing equations, and (3) solution of the algebraic equations (Murthy et al., 2006).

1.1. Basic Definitions

Prediction of heat transfer and fluid-flow processes can be obtained by two main methods: experimental investigation and theoretical calculation. We shall briefly consider each and then compare the two.

Experimental Investigation. The most reliable information about a physical process is often given by actual measurement. An experimental investigation involving full-scale equipment can be used to predict how identical copies of the equipment would perform under the same conditions. Such full-scale tests are, in most cases, prohibitively expensive and often impossible. The alternative then is to perform experiments on small-scale models. The resulting information, however, must be extrapolated to full scale, and general rules for doing this are often unavailable. Further, the small-scale models do not always simulate all the features of the full-scale equipment; frequently, important features such as combustion or boiling are omitted from the model tests. This further reduces the usefulness of the test results. Finally, it must be remembered that there are serious difficulties of measurement in many situations, and that the measuring instruments are not free from errors, [Hirsch,2007].

Theoretical Calculation. A theoretical prediction works out the consequences of a mathematical model, rather than those of an actual physical model. For the physical processes of interest here, the mathematical model mainly consists of a set of differential equations. If the methods of classical mathematics were to be used for solving these equations, there would be little hope of predicting many phenomena of practical interest. A look at a classical text on heat conduction or\ fluid mechanics leads to the conclusion that only a tiny fraction of the range of practical problems can be solved in closed form. Further, these solutions often contain infinite series, special functions, transcendental equations for eigenvalues, etc., so that their numerical evaluation may present a formidable task, (Patankar, 1975).

We shall now list the advantages that a theoretical calculation offers over a corresponding experimental investigation. These are low cost, speed, complete information, ability to simulate realistic and ideal conditions. The foregoing advantages are sufficiently impressive to stimulate enthusiasm about computer analysis. A blind enthusiasm for any cause is however undesirable. It is useful to be aware of the drawbacks and limitations. For the purpose of discussing the disadvantages of a theoretical calculation, it is, therefore, useful to divide all practical problems into two groups:

Group A: Problems for which an adequate mathematical description can be written. (Examples: heat conduction, laminar flows, and simple turbulent boundary layers.). For difficult problems involving complex geometry, strong nonlinearities, sensitive fluid-property variations, etc., a numerical solution may be hard to obtain and would be excessively expensive if at all possible. Extremely fast and small-scale phenomena such as turbulence, if they are to be computed in all their time-dependent detail by solving the unsteady Navier-Stokes equations, are still beyond the practical reach of computational methods.

Group B: Problems for which an adequate mathematical description has not yet been worked out. (Examples: complex turbulent flows, certain non-Newtonian flows, formation of nitric oxides in turbulent combustion, some two-phase flows.) This research consists of proposing a model, working out its implications by computer analysis, and comparing the results with experimental data. Thus, computational methods play a key role in this research.

The steps which must follow someone in order to use CFD software for the suitable solution to his problem can be presented next, Fig 1. These are:

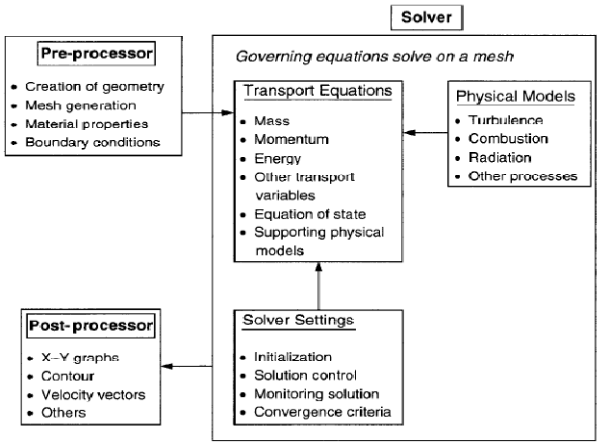

Fig 1. *The use of CFD Software.*

1. Specify the basic problems equations in 1D, 2D or 3D space. These are the mass, the momentum, the energy, the molecular and the temperature ones.
2. Specify the turbulence model if you have.
3. Solve the above equations and modify them so as their final form will take the general form of equation 8. Your scope will be to specify all the appropriate coefficients to input in the software.
4. Discretize the RANS equations following the finite elements or the Boundary elements or the Finite difference method.
    1. Specify the grid according to its geometry
    2. Choose the appropriate scheme
    3. Specify the over relaxation and/or under relaxation factors
    4. Specify the convection and/or diffusion

factors
5. Specify the pressure and the velocity corrections
1. Specify initial values and boundary conditions
2. Choose a computer algorithm
3. Choose a CFD software program.

2. Basic Equations

The numerical solution of heat transfer, fluid flow, and other related processes can begin when the laws governing these processes have been expressed in mathematical form, generally in terms of differential equations. For a detailed and complete derivation of these equations, the reader should turn to a standard textbook. Our purpose here is to develop familiarity with the form and the meaning of these equations. It will be shown that all the equations of relevance here possess a common form, the identification of which is the step toward constructing a general solution procedure, (Lauder and Spalding, 1972).

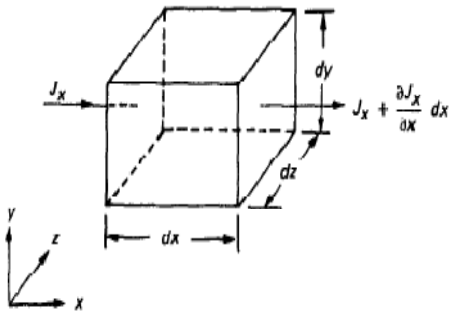

Fig 2. *Flux balance over a control volume.*

The individual differential equations that we shall encounter express a certain conservation principle. Each equation employs a certain physical quantity as its dependent variable and implies that there must be a balance among the various factors that influence the variable. The dependent variables of these differential equations are usually specific properties, i.e., quantities expressed on a unit-mass basis. Examples are mass fraction, velocity (i.e., momentum per unit mass), and specific enthalpy. The terms in a differential equation of this type denote influences on a unit-volume basis. An example will make this clear. Suppose J denotes a flux influencing a typical dependent variable φ. Let us consider the control volume of dimensions dx, dy, dz shown in Fig. 2. So, we can take the net balance per unit volume:

$$\frac{\partial J_x}{\partial x} + \frac{\partial J_y}{\partial y} + \frac{\partial J_z}{\partial z} = divJ \qquad (1)$$

Another example of a term expressed on a unit-volume basis is the rate-of-change term $\partial(\varphi\rho)/\partial t$. If φ is a specific property and ρ is the density, then φρ denotes the amount of the corresponding extensive property contained in a unit volume. Thus, $\partial(\varphi\rho)/\partial t$ is the rate of change of the relevant property per unit volume.

Let m_l denote the mass fraction of a chemical species. In the presence of a velocity field u, the conservation of m_l is expressed as:

$$\frac{\partial}{\partial t}(\rho m_1) + div(\rho u m_1) = div(\Gamma_l grad m_1) + R_l \qquad (2)$$

Where Γ_l is the diffusion coefficient, $\partial(\rho m_1)/\partial t$ denotes the rate of change of the mass of the chemical species per unit volume. Also, the quantity $\rho u m_l$ is the convection flux of the species, the R_l is the rate of generation of the chemical species per unit volume (negative, zero or positive,(Launder and Spalding, 1974).

The energy equation in its most general form contains a large number of influences. Since we are primarily interested in the form rather than in the details of the equation, it will be sufficient to consider some restricted cases. For a steady low-velocity flow with negligible viscous dissipation, for ideal gases, with c as a constant pressure specific heat and h = cT, the energy equation can be written as

$$div(\rho u T) = div\left(\frac{k}{c} grad\, T\right) + \frac{S_h}{c} \qquad (3)$$

The differential equation governing the conservation of momentum in a given direction for a Newtonian fluid can be written along similar lines; however, the complication is greater because both shear and normal stresses must be considered and because the Stokes viscosity law is more complicated than Fick's law or Fourier's law. With u denoting the x-direction velocity, we write the corresponding momentum equation as

$$\frac{\partial}{\partial t}(\rho u) + div(\rho u u) = div(\mu grad u) - \frac{\partial P}{\partial x} + B_x + V_x \qquad (4)$$

The momentum equation can be written in Cartesian coordinate system as:

$$\frac{\partial}{\partial x}(\rho u u) + \frac{\partial}{\partial y}(\rho u v) + \frac{\partial}{\partial z}(\rho u w) = -\frac{\partial P}{\partial x} + \frac{\partial}{\partial x}\left(\mu_{eff}\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mu_{eff}\frac{\partial u}{\partial y}\right) + \frac{\partial}{\partial z}\left(\mu_{eff}\frac{\partial u}{\partial z}\right) + \frac{\partial}{\partial x}\left(\mu_T\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mu_T\frac{\partial v}{\partial x}\right) + \frac{\partial}{\partial z}\left(\mu_T\frac{\partial w}{\partial x}\right) \qquad (5)$$

$$\frac{\partial}{\partial x}(\rho v u) + \frac{\partial}{\partial y}(\rho v v) + \frac{\partial}{\partial z}(\rho v w) = -\frac{\partial P}{\partial y} + \frac{\partial}{\partial x}\left(\mu_{eff}\frac{\partial v}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mu_{eff}\frac{\partial v}{\partial y}\right) + \frac{\partial}{\partial z}\left(\mu_{eff}\frac{\partial v}{\partial z}\right) + \frac{\partial}{\partial x}\left(\mu_T\frac{\partial u}{\partial y}\right) + \frac{\partial}{\partial y}\left(\mu_T\frac{\partial v}{\partial y}\right) + \frac{\partial}{\partial z}\left(\mu_T\frac{\partial w}{\partial y}\right) \qquad (6)$$

$$\frac{\partial}{\partial x}(\rho u w) + \frac{\partial}{\partial y}(\rho v w) + \frac{\partial}{\partial z}(\rho w w) = -\frac{\partial P}{\partial z} + \frac{\partial}{\partial x}\left(\mu_{eff}\frac{\partial w}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mu_{eff}\frac{\partial w}{\partial y}\right) + \frac{\partial}{\partial z}\left(\mu_{eff}\frac{\partial w}{\partial z}\right) + \frac{\partial}{\partial x}\left(\mu_T\frac{\partial u}{\partial z}\right) + \frac{\partial}{\partial y}\left(\mu_T\frac{\partial v}{\partial z}\right) + \frac{\partial}{\partial z}\left(\mu_T\frac{\partial w}{\partial z}\right) \qquad (7)$$

The density may be related, via an equation of state, to variables such as mass fraction and temperature. These variables and the velocity components obey the general differential equation. Further, the flow field should satisfy

an additional constraint, namely, the mass conservation or the continuity equation, which is

$$\frac{\partial}{\partial t}(\rho t) + \mathrm{div}(\rho u) = 0 \quad (8)$$

Another useful representation is the Cartesian-tensor form of these equations:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j}(\rho u_j) = 0 \quad (9)$$

Where the subscript j can take the values 1, 2, 3, denoting the three spaces coordinates. When a subscript is repeated in a term, a summation of three terms is implied; for example,

$$\frac{\partial}{\partial x_j}(\rho u_j) = \frac{\partial}{\partial x_1}(\rho u_1) + \frac{\partial}{\partial x_2}(\rho u_2) + \frac{\partial}{\partial x_3}(\rho u_3) \quad (10)$$

An immediate benefit of the Cartesian-tensor form is that the one-dimensional form of the equation is obtained by simply dropping the subscript j. The procedure for casting any particular differential equation into the general form (16) is to manipulate it until, for the chosen dependent variable, the unsteady term and the convection and diffusion terms conform to the standard form. The coefficient of grad φ in the diffusion term is then taken as the expression for Γ, and the remaining terms on the right-hand side are collectively defined as the source term S.

3. The Turbulence Model

The equations for unsteady laminar flow are converted into the time averaged equations for turbulent flow by an averaging operation in which it is assumed that there are rapid and random fluctuations about the mean value. The additional terms arising from this operation are the so-called Reynolds stresses, turbulent heat flux, turbulent diffusion flux, etc. To express these fluxes in terms of the mean properties of the flow is the task of a turbulence model.

Many turbulence models employ the concept of a turbulent viscosity or a turbulent diffusivity to express the turbulent stresses and fluxes. The result is that the time-averaged equations for turbulent flow have the same appearance as the equations for laminar flow, but the laminar exchange coefficients such as viscosity, diffusivity, and conductivity are replaced by *effective* (i.e., laminar plus turbulent) exchange coefficients. From a computational viewpoint, a turbulent flow within this framework is equivalent to a laminar flow with a rather complicated prescription of viscosity. (The same idea is applicable to non-Newtonian flows, which can be thought of as flows in which the viscosity depends on the velocity gradient. The currently popular "two-equation models" of turbulence (Launder and Spalding, 1972, 1974) employ, as one of the equations, the equation for the kinetic energy k of the fluctuating motion and the term (G-ρε) is the net source term, then we take:

$$\frac{\partial}{\partial t}(\rho k) + div(\rho u k) = div(\Gamma_k grad k) + G - \rho\varepsilon \quad (11)$$

The numerical procedures which are associated with turbulence models to make complete calculation methods can be divided into integral and differential types. Differential methods involve direct assumptions for the Reynolds stresses at a point and seek the solution of the governing equations in their partial differential form. Integral methods involve the integral parameters of the shear layer momentum thickness, shape parameter, skin friction coefficient, etc. One solves a system of ordinary differential equations (for 2-D flows), whose dependent variables are the profile parameters and independent variable is x; in 3-D flows, the equations are the partial differential equations in the plane of the layer. The important distinction between calculation methods is the type of turbulence model rather than the type of numerical procedure. The advantage of differential methods is that the restrictions and inaccuracy that arise from the need to parameterize the velocity profiles are avoided. Differential methods introduce substantially more detailed information about turbulence.

The turbulence models can be classified in several ways. The one most often used is that arranged in order of the number of differential equations solved in addition to the mean flow equations [Reynolds, 1976].

(I) Zero equation models
(II) One equation models
(III) Two equation models
(IV) Stress equation models

Most of the models, classes (I)-(III), use Boussinesq eddy viscosity model. Bradshaw et al. (1967), however, assume the constancy of the r/pk ratio, in boundary layer flows. Here z is the shear stress and k is the turbulent kinetic energy. It is important to note here that in this case the mean momentum and continuity equations form a hyperbolic system in contrast to the parabolic system obtained with the use of eddy viscosity models [Cebeci et al., 1980]. Other models which do not use the eddy viscosity assumption (class IV) obtain the Reynolds stress from a differential equation.

Zero equation model, which uses only the partial differential equation for the mean flow field and no transport equations for turbulence quantities, is also called "mean field" closure [Mellor and Herring, 1973]. The classes (II) to (IV) are called "transport equation" closures. Bradshaw (1972) describes the interplay between the development of models and the experiments.

Turbulence models for general-purpose CFD must be frame-invariant – i.e. independent of any particular coordinate system – and hence must be expressed in tensor form. These rules out simpler models of boundary-layer type (e.g. mixing-length models). Turbulent flows are computed either by solving the Reynolds-averaged Navier-Stokes equations with suitable models for turbulent fluxes or by computing the fluctuating quantities directly. The main approaches are summarized below.

Reynolds-Averaged Navier-Stokes (RANS) Models
- Eddy-viscosity models (EVM)
  - assume the (deviatory) turbulent stress proportional to the mean rate of strain;
  - Eddy viscosity derived from turbulent transport equations (usually k + one other).
- Non-linear eddy-viscosity models (NLEVM)
  - turbulent stress modelled as a non-linear function of mean velocity gradients;
  - turbulent scales determined by solving transport equations (usually k + one other)
  - Mimic response of turbulence to certain important types of strain.
- Differential stress models (DSM)
  - aka Reynolds-stress transport models (RSTM) or second-order closure (SOC);
  - Solve transport equations for all turbulent stresses.

Computation of fluctuating quantities
- Large-eddy simulation (LES)
  - Compute time-varying flow, but model sub-grid-scale motions.
- Direct numerical simulation (DNS)
  - No modelling; resolve the smallest scales of the flow.

The monograph by Launder and Spalding (1972) gives the mathematical concepts of turbulence models. Bradshaw and Cebeci (1978) describe the calculation methods for various classes of turbulent flows. Gosman et al. (1969) present various aspects of computation of recirculating flows. Several reviews have appeared concentrating on different aspects of turbulence modeling (Lauder and Spalding, 1972; Rotta, 1975; Lumley, 1978; Rodi, 1982; Spalding, 1982). A general review of turbulence models and their applications can be found in Refs [Lumley, 1978; Rodi, 1982; Spalding, 1982; Launder and Spalding, 1974; Meller and Yamada, 1982]. Launder and Spalding (1972) review briefly the turbulence models and their applications to internal flows, concentrating on their work till 1972. The reviews by Marvin (1983), Murphy (1984) and Wilcox and Rubesin (1980) have concentrated on turbulence models and their applications to external flows. In particular, they concentrated on the models used by NASA Ames groups. Lumley (1978, 1983) and Launder (1982) discuss the prospects of higher order closure models. Turbulence models and their application to atmospheric turbulence are reviewed by Donaldson (1972). Launder et al (1984) discuss second-moment closures in great detail. Meller and Yamada (1982) described the development and application of closure models to atmospheric and oceanic boundary layers. Rodi (1984) describes in detail the various turbulence models and their application to hydraulics problems. He evaluates the models with regard to their predictive capability and computational effort. Turbulence models available for the prediction of complex 3-D flows with curvature, rotation and flow separation are reviewed by Lekshminarayana (1985). Patel *et al.(1985)* provide an exhaustive review of turbulence models for near-wall and low Reynolds number flows. The present review concentrates on turbulence models and their applications to a number of internal flows. The flows considered include 2- and 3-D flows, flows with and without swirl and flows with secondary motion. In the following, the main classes of turbulence models are described briefly.

The standard high Reynolds number k-ε turbulence model has been widely used in industrial applications to predict the overall performance of a device. The model has been proved to be very robust and economical from the view point of computer time because of the use of standard wall functions. However, it has been observed that in recirculating flow, the prediction of near wall quantity using the k-ε model does not compare very well with other low Reynolds number models. So, for the accurate prediction of overall quantity (the mean residence time, mixed volume and dead volume in a situation where mixing is of importance) in a device, modified forms of the standard k-ε model have been developed in the last decade. However, such modified k-ε models has not been used very extensively for industrial cases except its validation with simple experiments. It has been the main motivation of the present work to use the standard k-ε model of Launder and Spalding (1972) along with its modifications, RNG (Yahkot and Orszag, 1992), Chen-Kim (CK) and k-ε with Yap correction (k-ε Yap) (Monson et al., 1990). The four turbulence models listed above are all of high Reynolds number form and are, of course, restricted to situations in which the Reynolds number is sufficiently high for the viscous effect to be unimportant. However, to describe the flow close to a solid wall where the Reynolds number is not sufficiently high, low Reynolds number turbulence models with near-wall modifications have been reported to perform better (Patel et al., 1986). In many computations, we intend to use various low Reynolds number models like the Lam-Bremhorst model (LB model) (Lam and Bremharet, 1981), the Chen-Kim low Reynolds number (CK low Re) model (with and without Yap correction) and the simplest of the turbulence models the constant effective viscosity model.

Conservation equation for the turbulence kinetic energy

$$\frac{\partial}{\partial x}(\rho uk)+\frac{\partial}{\partial y}(\rho vk)+\frac{\partial}{\partial z}(\rho wk)=\frac{\partial}{\partial x}\left(\frac{\mu_{eff}}{\sigma_k}\frac{\partial k}{\partial x}\right)+\frac{\partial}{\partial y}\left(\frac{\mu_{eff}}{\sigma_k}\frac{\partial k}{\partial y}\right)+\frac{\partial}{\partial z}\left(\frac{\mu_{eff}}{\sigma_k}\frac{\partial k}{\partial z}\right)+G_k-\rho\varepsilon \quad (12)$$

Conservation equations for the dissipation rate of turbulence kinetic energy

$$\frac{\partial}{\partial x}(\rho uk)+\frac{\partial}{\partial y}(\rho vk)+\frac{\partial}{\partial z}(\rho wk)=\frac{\partial}{\partial x}\left(\frac{\mu_{eff}}{\sigma_k}\frac{\partial k}{\partial x}\right)+\frac{\partial}{\partial y}\left(\frac{\mu_{eff}}{\sigma_k}\frac{\partial k}{\partial y}\right)+\frac{\partial}{\partial z}\left(\frac{\mu_{eff}}{\sigma_k}\frac{\partial k}{\partial z}\right)+G_k-\rho\varepsilon \quad (13)$$

In these equations G_k represents the generation of turbulence kinetic energy due to the mean velocity gradients and can be expressed as;

$$G_k = 2\mu_T\left[\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 + \left(\frac{\partial w}{\partial z}\right)^2\right] + \left[\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right)^2\right] \quad (14)$$

The turbulent or eddy viscosity μ_T, is computed by combining k and ε as follows:

$$\mu_T = \frac{c_\mu \rho k^2}{\varepsilon} \quad (15)$$

Finally, c_1, c_2, c_μ, σ_k, σ_ε are empirical constants with standard values [Launder and Spalding, 1974] as:

$$C_1 = 1.43,\ c_2 = 1.92,\ c_\mu = 009,\ \sigma_k = 1.0,\ \sigma_\varepsilon = 1.3$$

4. The General Form of Equations

This brief journey through some of the relevant differential equations has indicated that all the dependent variables of interest here seem to obey a generalized conservation principle. If the dependent variable is denoted by Φ=Φ(x,y,z,t) the general differential equation is

$$\frac{\partial}{\partial t}(\rho\Phi) + \mathrm{div}(\rho\Phi u) = \mathrm{div}(\Gamma \mathrm{grad}\Phi) + S \quad (16)$$

$$\frac{\partial}{\partial t}(\rho\Phi) + \frac{\partial}{\partial x_j}(\rho u_j \Phi) = \frac{\partial}{\partial x_j}\left(\Gamma \frac{\partial \Phi}{\partial x_j}\right) + S \quad (17)$$

The four terms in the general differential equation (16) are the unsteady term, the convection term, the diffusion term, and the source term. The dependent variable φ can stand for a variety of different quantities, such as the mass fraction of a chemical species, the enthalpy or the temperature, a velocity component, the turbulence kinetic energy, or a turbulence length scale. Accordingly, for each of these variables, an appropriate meaning will have to be given to the diffusion coefficient rand the source terms. - Not all diffusion fluxes are governed by the gradient of the relevant variable. The use of div(Γgradφ) as the diffusion term does not, however, limit the general φ equation to gradient-driven diffusion processes. Whatever cannot be fitted into the *nominal* diffusion term can always be expressed as a part of the source term; in fact, the diffusion coefficient Γ can even be set equal to zero if desired. A gradient-diffusion term has been explicitly included in the general φ equation because most dependent variables do require a prominent diffusion term of this nature.

Many times we can rewrite the eq.16 in equal definitions forms like:

$$\frac{\partial}{\partial t}(\rho\Phi) + \mathrm{div}(\rho\Phi u - \Gamma \mathrm{grad}\Phi) = S \quad (18)$$

$$\frac{\partial}{\partial t}(\rho\Phi) + \mathrm{div}(\rho\Phi u + J) = S \quad (19)$$

In eq.19 we replace the diffusion term by the Fick law equation where J=-ΓgradΦ. Many times in Fick law we use the diffusion coefficient D = Γ/ρ. Using the eq. 16 and changing the values in Φ, we can take the mass, the momentum, the energy, the temperature, the turbulence model equations.

Table 1. *Coefficients value for the equations 16.*

Equation	Φ	$\Gamma_{\varphi i}$	S
Mass	1	0	0
Chemical species	m_i	$-\Gamma_i$ gradm$_i$	R_i
Momentum	u,v,w	μgradu	$-\frac{\partial P}{\partial x} + B_x + V_x$
Energy	T_i	K/c gradT	S_h/c
Turbulence kinetic energy	k	$\mu + \frac{\nu_t}{\sigma_k}$	ρ(G-ε)
Dissipation rate of turbulence kinetic rate	ε	$\mu + \frac{\nu_t}{\sigma_\varepsilon}$	$\rho\frac{k}{\varepsilon}(c_{e1}G - c_{e2}\varepsilon)$

Another useful representation is the Cartesian-tensor form of these equations. Where the subscript *j* can take the values 1, 2, 3, denoting the three spaces coordinates. When a subscript is repeated in a term, a summation of three terms is implied; for example,

$$\frac{\partial}{\partial x_j}\left(\Gamma \frac{\partial \phi}{\partial x_j}\right) = \frac{\partial}{\partial x_1}\left(\Gamma \frac{\partial \phi}{\partial x_1}\right) + \frac{\partial}{\partial x_2}\left(\Gamma \frac{\partial \phi}{\partial x_2}\right) + \frac{\partial}{\partial x_3}\left(\Gamma \frac{\partial \phi}{\partial x_3}\right) \quad (20)$$

Definitions. A two-way coordinate is such that the conditions at a given location in that coordinate are influenced by changes in conditions on either side of that location. A one-way coordinate is such that the conditions at a given location in that coordinate are influenced by changes in conditions on only one side of that location, (Patankar and Spalding, 1970).

Parabolic, elliptic, hyperbolic. It appears that the mathematical terms *parabolic* and *elliptic,* which are used for the classification of differential equations, correspond to our computational concepts of one-way and two-way coordinates. The term parabolic indicates a one-way behavior, while elliptic signifies the two-way concept. A situation is parabolic if there exists at least one one-way coordinate; otherwise, it is elliptic. A flow with one one-way space coordinate is sometimes called a boundary-layer-type flow, while a flow with all two-way coordinates is referred to as a recirculating flow (Patankar and Spalding 1970; Gosman at al., 1969).

Discretization. For a given differential equation, the required discretization equations may be derived in many ways such as Taylor Series Formulation, Variational Formulation, Method of weighted Residuals, Control Volume Formulation, etc. The result of the discretization must satisfy the four basic rules, which are a) the consistency at control volume faces, b) the positive coefficients, c) the negative slope linearization of the source terms and d) the sum of the neighbor coefficients.

Steady one-dimensional

We have already derived the discretization equation for steady conduction in one dimension. The governing differential equation, where with S we show the rate of heat generation per unit volume, Fig. 3, is:

$$\frac{d}{dx}\left(k\frac{dT}{dx}\right) + S = 0 \quad (21)$$

This leads to the discretization equation

$$a_P T_P = a_E T_E + a_W T_W + b \,, \tag{22}$$

Unsteady one-dimensional

With reference to the general differential equation for φ, we have now seen, at least in the one-dimensional context, how to handle the diffusion term and the source term. Here, we turn to the unsteady term and temporarily drop the source term, since nothing new needs to be said about it. Thus, we seek to solve the unsteady one-dimensional heat-conduction equation

$$\rho c \frac{\partial T}{\partial t} = \frac{\partial}{\partial x}\left(k \frac{\partial T}{\partial x}\right) \tag{23}$$

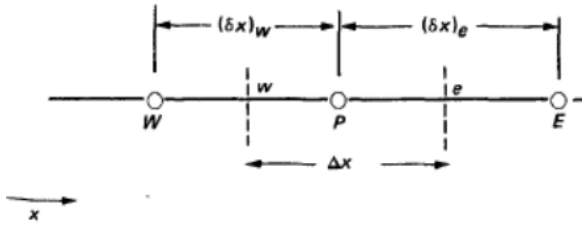

Fig 3. *Grid point cluster for the one dimensional problem.*

The discretization equation is now derived by integrating Eq. (23) over the control volume shown in Fig.3 and over the time interval from t to (t+Δt). Thus

$$\rho c \int_w^e \int_t^{t+\Delta t} \frac{\partial T}{\partial t}\, dt\, dx = \int_t^{t+\Delta t} \int_w^e \frac{\partial}{\partial x}\left(k \frac{\partial T}{\partial x}\right) dx\, dt \tag{24}$$

After representation of some terms, assumptions and replacements we take as final result:

$$a_P T_P = a_E \left[f T_E + (1-f) T_E^0\right] + a_W \left[f T_W + (1-f) T_W^0\right] + \left[a_P^0 - (1-f) a_E - (1-f) a_W\right] T_P^0 \tag{25}$$

Discretization Equation for Two Dimensions

A portion of a two-dimensional grid is shown in Fig.4. For the grid point P, points E and W are its x-direction neighbors, while N and S (denoting north and south) are the y-direction neighbors. The control volume around P is shown by dashed lines. Its thickness in the z direction is assumed to be unity. The nomenclature introduced in Fig.3 for distances Δx, $(\delta x)_e$, etc. is to be extended to two dimensions here. Locating them exactly midway between the neighboring grid points is an obvious possibility, but other practices can also be employed. We have seen how to calculate the heat flux Q_e at the control-volume face between P and E. We shall assume that Q_e, thus obtained, prevails over the entire face of area (Δy*1). Heat flow rates through the other faces can be obtained in a similar fashion. In this manner, the differential equation

$$\rho c \frac{\partial T}{\partial t} = \frac{\partial}{\partial x}\left(k \frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k \frac{\partial T}{\partial y}\right) + S \tag{26}$$

Can be instantly turned into the discretization equation

$$a_P T_P = a_E T_E + a_W T_W + a_N T_N + a_S T_S + b \tag{27}$$

Discretization Equation for Three Dimensions

Finally, we add two more neighbors T and B (top and bottom) for the z direction to complete the three-dimensional configuration. The discretization equation can easily be seen to be

$$a_P T_P = a_E T_E + a_W T_W + a_N T_N + a_S T_S + a_T T_T + a_B T_B + b \tag{28}$$

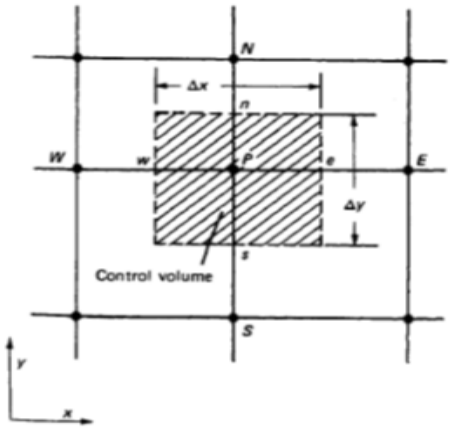

Fig 4. *Control volume for the two dimensional problem.*

4.1. The Use of a Grid

Finite difference equations are most efficiently solved in a rectangular domain (for 2-D applications and an equivalent hexahedral domain for 3-D applications) with equal grid spacing. Unfortunately, the majority of physical domains encountered are nonrectangular in shape. Thus, it is necessary to transform the nonrectangular physical domain to a rectangular computational domain where grid points are distributed at equal spacing. It is also important to note that the transformation allows the alignment of one of the coordinates along the body, thus facilitating the implementation of the boundary conditions. The objective of grid generation is then to identify the location of the grid points in the computational domain and the location of the corresponding grid points in the physical space. Furthermore, the metrics and Jacobian of transformation which are required for the solution of flow equations are computed within the grid generation routine, [Hoffman and Chiang, 2000].

Typically, grid generation schemes may be categorized as algebraic methods or differential methods. In the latter case, the scheme is based on the solution of a set of PDEs and may be subcategorized as either an elliptic, parabolic, or hyperbolic grid generation. Either category of grid generation scheme should include the following

considerations.

1. A mapping which guarantees one-to-one correspondence ensuring grid lines of the same family do not cross each other;
2. Smoothness of the grid distribution;
3. Orthogonality or near orthogonality of the grid lines;
4. Options for grid clustering.

A brief summary of the advantages and disadvantages of each method is provided below.

1. Algebraic grids

The advantages of this category of grid generators are: (a) They are very fast computationally; (b) Metrics may be evaluated analytically, thus avoiding numerical errors; (c) The ability to cluster grid points in different regions can be easily implemented.

The disadvantages are: (a) Discontinuities at a boundary may propagate into the interior region which could lead to errors due to sudden changes in the metrics; (b) Smoothness and skewness may be difficult to control.

2. Elliptic grids

The advantages of this class of grid generators are: (a) Will provide smooth grid point distribution, i.e., if a boundary discontinuity point exists, it will be smoothed out in the interior domain; (b) Numerous options for grid clustering and surface orthogonality are available; (c) Method can be extended to 3-D problems.

The disadvantages of the method are: (a) Computation time is large (compared to algebraic methods or hyperbolic grid generators); (b) Specification of the forcing functions P and Q (or the constants used in these functions) is not easy; (c) Metrics must be computed numerically.

3. Hyperbolic grids

The advantages of hyperbolic grid generators are: (a) The grid system is orthogonal in two dimensions; (b) Since a marching scheme is used for the solution of the system, computationally they are much faster compared to elliptic systems; (c) Grid line spacing may be controlled by the cell area or arc-length functions.

The disadvantages are: (a) Boundary discontinuity may be propagated into the interior domain; (b) Specifying the cell-area or arc-length functions must be handled carefully. A bad selection of these functions easily leads to undesirable grid systems.

4.2. The Use of a Scheme

For certain specific values of the weighting factors f_i, the discretization equation reduces to one of the well-known schemes for parabolic differential equations, [Patel and Markatos, 1986]. In particular, f = 0 leads to the explicit scheme, f = 0.5 to the Crank-Nicolson scheme, and f = 1 to the fully implicit scheme. Here, we present the equations of the fully implicit scheme which is the most common in use.

$$a_P T_P = a_E T_E + a_W T_W + b \tag{29}$$

Where

$$a_E = \frac{k_e}{(\delta x)_e} \tag{30}$$

$$a_W = \frac{k_w}{(\delta x)_w} \tag{31}$$

$$a_P^0 = \frac{\rho c\ \Delta x}{\Delta t} \tag{32}$$

$$b = S_C\ \Delta x + a_P^0 T_P^0 \tag{33}$$

$$a_P = a_E + a_W + a_P^0 - S_P\ \Delta x \tag{34}$$

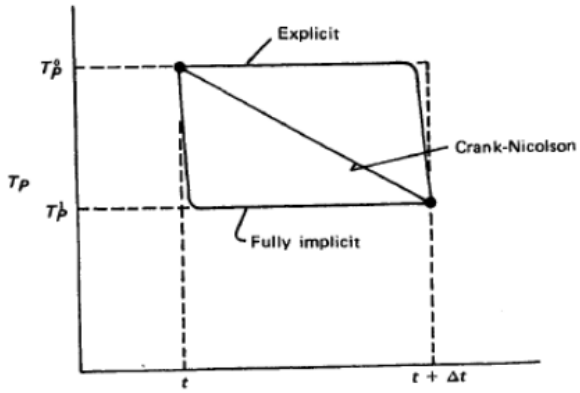

Fig 5. *Variation of temperature with time for different schemes.*

4.3. Over Relaxation and under Relaxation

In the iterative solution of the algebraic equations or in the overall iterative scheme employed for handling nonlinearity, it is often desirable to speed up or to slow down the changes, from iteration to iteration, in the values of the dependent variable. This process is called over laxation or under laxation depending on whether the variable changes are accelerated or slowed down: Over relaxation is often used in conjunction with the Gauss-Seidel method the resulting scheme being known as Successive Over-Relaxation (SOR). With the line-by-line method, the use of over relaxation is less common. Under relaxation is a very useful device for nonlinear problems. It is often employed to avoid divergence in the iterative solution of strongly nonlinear equations. There are many ways of introducing over relaxation or under relaxation. Some practices will be described here. We shall work with the general discretization equation of the form

$$a_P T_P = \Sigma\, a_{nb} T_{nb} + b \tag{35}$$

Another technique of over relaxation or under relaxation is to replace the discretization equation (35) with the next equation where i is the so called inertia and a is the suitable relaxation factor. For positive values of i the equation has the effect of under relaxation and for negative values of i produce over relaxation.

$$(a_P + i)T_P = \Sigma a_{nb} T_{nb} + b + iT_P^* \quad (36)$$

4.4. Convection and Diffusion

So far we have seen how to formulate the discretization equation from the general differential equation containing the unsteady term, the diffusion term, and the source term. The only omission has been the convection term, which we shall now include. We have also dealt with the methods of solving the algebraic equations; as long as the addition of the convection term does not alter the *form* of the discretization equation, the same methods continue to apply. The convection is created by fluid flow. Our task is to obtain a solution for φ in the presence of a *given flow field* (i.e., the velocity components and the density). The origin of the flow-field information is immaterial here. Having somehow acquired the flow field, we wish to calculate the temperature, concentration, enthalpy, or any such quantity that is represented by the general variable φ. Although convection is the only new term introduced in this chapter, its formulation is not very straightforward. The convection term has an inseparable connection with the diffusion term, and therefore, the two terms need to be handled as one unit.

5. Specify Initial Values and Boundary Conditions

The flow can be from the top left hand corner. The flow field is computed by solving the mass and momentum conservation equations in a boundary fitted coordinate system along with a set of realistic boundary conditions. The boundary conforms to a regular Cartesian system in case of no inclined wall. The free surface of the liquid was considered to be flat and the slag depth was considered to be insignificant. With these two assumptions the flow field was solved with the help of the above equations for all the turbulence models. The effect of natural convection is ignored because the ratio, Gr/Re2=0.044ΔT [Lopez-Ramirez et al., 2000], where ΔT, the driving force for natural convection is the temperature difference between the liquid steel at the top free surface and the bulk temperature of the liquid, which is much less than unity for all the cases that are computed here.

The formation of waves at the free surface was ignored. The free surface was assumed to be flat and mobile. Fluxes of all quantities across the free surface were assumed to be zero [Szekely et al,, 1987]. Therefore, normal velocity component (for convective flux) and normal gradients of all variables (for diffusive flux) were all set to zero, i.e.

$$w=0, \quad \frac{\partial u}{\partial z}=0, \quad \frac{\partial v}{\partial z}=0, \quad \frac{\partial k}{\partial z}=0, \quad \frac{\partial \varepsilon}{\partial z}=0, \quad (37)$$

The exit can be computationally treated as either a standard outflow or as a plane or surface, at which flow occurs at an ambient pressure (taken). At the outlets, both types of boundary conditions were applied in order to assess the similarity of the experimental results to model configuration.

At all the solid walls, the velocity components was set to zero, at both the side walls,

$$u=0, \quad v=0, \quad w=0, \quad k=0, \quad \varepsilon=0 \quad (38)$$

At both the frontal side walls,

$$u=0, \quad v=0, \quad w=0, \quad k=0, \quad \varepsilon=0 \quad (39)$$

At the bottom wall

$$u=0, \quad v=0, \quad w=0, \quad k=0, \quad \varepsilon=0 \quad (40)$$

6. Computer Program and CFD Software

It should be noted that, while constructing the discretization equations, we cast them into a linear form but did not assume that a particular method would be used for their solution. Therefore, any suitable solution method can be employed at this stage. It is useful to consider the derivation of the equations and their solution as two distinct operations, and there is no need for the choices in one to influence the other. In a computer program, the two operations can be conveniently performed in separate sections, and either section can be independently modified when desired.

So far, we have obtained the multidimensional discretization equations by a straightforward extension of the one-dimensional situation. One procedure that cannot so easily be extended to multiple dimensions is the tridiagonal matrix algorithm (TDMA). Direct methods (i.e., those requiring no iteration) for solving the algebraic equations arising in two- or three-dimensional problems are much more complicated and require rather large amounts of computer storage and time. For a linear problem, which requires the solution of the algebraic equations only once, a direct method may be acceptable; but in nonlinear problems, since the equations have to be solved repeatedly with updated coefficients, the use of a direct method is usually not economical. We shall, therefore, exclude direct methods from further consideration, except to say that a computer program for the direct solution of discretization equations in two dimensions. There are many iterative methods for solving algebraic equations.

6.1. The Tridiagonal Matrix Algorithm (TDMA)

In numerical linear algebra, the tridiagonal matrix algorithm, also known as the Thomas algorithm, is a

simplified form of Gaussian elimination that can be used to solve tridiagonal systems of equations. A tridiagonal system for n unknowns may be written as

$$a_i x_{i-1} + b_i x_i + c_i x_{i+1} = d_i,$$

Where $a_1 = 0$ and $c_n = 0$.

$$\begin{bmatrix} b_1 & c_1 & & & 0 \\ a_2 & b_2 & c_2 & & \\ & a_3 & b_3 & \ddots & \\ & & \ddots & \ddots & c_{n-1} \\ 0 & & & a_n & b_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_n \end{bmatrix}.$$

For such systems, the solution can be obtained in $O(n)$ operations instead of $O(n^3)$ required by Gaussian elimination. A first sweep eliminates the a_i's, and then an (abbreviated) backward substitution produces the solution. Examples of such matrices commonly arise from the discretization of 1D Poisson equation (e.g., the 1D diffusion problems) and natural cubic spline interpolation; similar systems of matrices arise in tight binding physics or nearest neighbor effects models.

6.2. The Algorithm for Numerical Simulation

In the numerical solution of incompressible fluid flow and heat transfer problems, the pressure-correction approach is the most popular method used in CFD/NHT community. The first pressure-correction algorithm was the SIMPLE proposed by Patankar and Spalding (1972). The acronym SIMPLE stands for semi-implicit method for the pressure-linked equation. The major approximations made in the SIMPLE algorithm are (Tao, 2001):

The initial pressure field and the initial velocity fields are independently assumed, hence the inherent interconnection between pressure and velocities are neglected, leading to some inconsistency between them. The effects of the pressure corrections of the neighboring grids are arbitrary dropped in order to simplify the solution procedure, thus make the algorithm semi-implicit.

These assumptions will not affect the final solutions if the iterative process converges (Tao, 2001; Patankar, 1980). However, they do affect the convergence rate. As described in Shyy and Mittal (1998), the great simplicity of the SIMPLE algorithm comes from the neglecting the terms that couples neighboring velocity values in the equation for the velocity correction. However, this can also cause slow convergence of the SIMPLE algorithm and it has been found this neglect tends to over predict the pressure correction and under relaxation for the pressure correction has to be resorted to in order to stabilize the iterative procedure. Therefore, a number of variants SIMPLE algorithms were proposed in order to overcome one or both of the approximations (Liu et al., 2005).

The SIMPLER algorithm (Patankar, 1981) successfully overcome the first approximation, and is widely used in the current CFD/NHT community. Even though there are more than ten variants of the SIMPLE-like algorithm, the second approximation, i.e. the drop of the neighboring grid effects, have not been successfully resolved so far and many attempts have been made to resolve the problem. Van Doormaal and Raithby (1984) proposed the SIMPLEC algorithm, in which by changing the definition of the coefficients of the velocity correction equation the effects of this drop is partially compensated. In the algorithm SIMPLEX (van Doormaal and Raithby, 1985; Raithby and Schneider, 1988), by solving a set of algebraic equation for the coefficients in the velocity correction equations, the effects of dropping the neighboring grids are also taken into account in some degree. In 1985, the PISO method is proposed by Issa (1985) to implement two or more correction steps of pressure correction. In 1986 Connell and Stow (1986) proposed two variants of pressure correction process. Chatwani and Turan (1991) proposed a pressure-velocity coupling algorithm to determine the under-relaxation factor in the pressure correction equation based on the minimization of the global mass residual norm. Lee and Tzong (1992) introduced an artificial source term into the pressure-linked equation to improve the convergence performance. Yen and Liu (1993) proposed the explicit correction step method to accelerate the convergence by making the velocity explicitly satisfy the momentum equation. For buoyancy driven fluid flows Sheng et al. (1998) introduced a temperature correction into the velocity correction equation. Yu et al.(2001) modified the SIMPLER algorithm by artificially changing the under-relaxation term to match the variable to be solved. The revised method was called MSIMPLER. All the above-mentioned algorithms and some others not mentioned above (for example, SIMPLESSEC, SIMPLESSE of Gjesdal and Lossius (1997), and the method proposed in Wen and Ingham (1993) are usually called SIMPLE-like or SIMPLE-family algorithm. The character common to all these algorithms is that a pressure correction term is introduced to the segregated solution process to improve the velocity and the effects of the pressure corrections of the neighboring grid points are neglected. Moukalled and Darwish (2000) made a comprehensive review and reorganization of the express format for all the pressure correction algorithms. It can be seen that the SIMPLER algorithm successfully overcomes the first approximation, while almost all other variants of the SIMPLE algorithm concentrate on overcoming the second approximation. There seems no such attempt in the literature to combine the SIMPLER algorithm and one of the other variants so that the effects of both of the two approximations can be alleviated in a better degree in one algorithm.

Liu et al., (2005) work, the idea of SIMPLEC was incorporated into the SIMPLER algorithm to overcome the second approximation in some extent. The revised algorithm is called consistent-SIMPLER (CSIMPLER) hereafter. Numerical experiments showed that CSIMPLER

can generally accelerate the rate of convergence, especially for natural convection and the cases with low under-relaxation factor.

In the last few years there has been continuous progress in the development of CFD codes. These codes can now cope with a high level of complexity in many research fields, which makes them attractive to use. However, CFD codes that are used in the food processing industry have not reached a relatively mature state, they still require to be improved in accuracy, ease of use, robustness and computational efficiency. Of all the commercial CFD codes there are some codes for general purpose as well as for specific applications (e.g. non-Newtonian flow). Most of these commercial CFD codes can be supported on the platforms of UNIX on workstations and WINDOWS or LINUX on high-end Intel Pentium PCs. Table 2 summarizes the main commercial CFD codes.

Table 2. *Commercial CFD code list.*

CFD code	Web site
CFX	http://www.software.aeat.com/cfx/
FLUET	http://www.fluent.com/
PHOENICS	http://www.cham.co.uk/
STAR-CD	http://www.cd.co.uk
FLOW-3D	http://www.flow3d.com
CFD-ACE	http://www.cfdrc.com
ICEM CFD	http://www.icemcfd.com/icepak.html
TECPLOT	http://www.amtec.com
PAM-FLOW	http://www.esi.com.au/
FLOVENT	http://www.flomerics.com/

6.2.1. The SIMPLER Algorithm

In all cases considered, the flow was assumed to be Newtonian, laminar and two-dimensional. Viscous dissipation is omitted. All thermo physical properties except density are presumed constant. The Boussinesq approximation is used for natural convection problem. Discretizing the governing equations by the finite volume method (Tao, 2001; Moukalled and Darwish, 2000) on a staggered grid system (Figure 6), we have another form of the equation (35)

$$a_P\Phi_P = \sum a_{\mathrm{nb}}\Phi_{\mathrm{nb}} + b$$

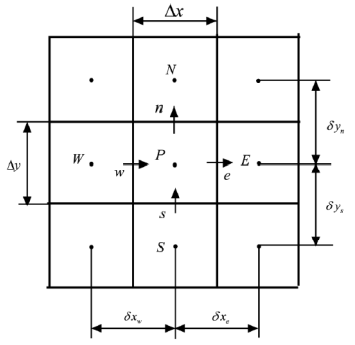

Figure 6. *Control volumes in 2D Cartesian coordinates.*

Where Φ is the general valuable standing for u and v, the subscripts P and n_b refer to the gird point P and its neighboring grids, respectively, a_p is the coefficient for the main grid point, a_{nb}'s are the coefficients of neighboring grid points and b is the source term. For the discretized momentum equations separating the pressure gradient term from the b-term and replace the general valuable by u or v, we have:

$$a_e u_e = \sum a_{\mathrm{nb}} u_{\mathrm{nb}} + (p_P - p_E)A_e + b \quad (42)$$

$$a_n v_n = \sum a_{\mathrm{nb}} v_{\mathrm{nb}} + (p_P - p_N)A_n + b \quad (43)$$

The discretized pressure equation is deduced from the momentum equations and the continuity equation and can be expressed as, (Tao, 2001; Patankar, 1980):

$$a_P p_P = \sum a_{\mathrm{nb}} p_{\mathrm{nb}} + b_P \quad (44)$$

By solving equations (42) and (43), we can obtained the intermediate solutions, symbolized by u^* and v^* which need to be improved such that the improved velocities can satisfy the mass conservation condition for each control volume. By introducing a pressure correction term, p', and the corresponding velocity correction terms u' and v', the improved velocities can be expressed by

$$u = u^* + u' \quad (45)$$

$$v = v^* + v' \quad (46)$$

These improved velocities are required to satisfy the continuity condition. The equations for the velocity correction terms, u', v', can be derived by some substitution and rearrangement (Tao, 2001; Patankar, 1980), and take the following form:

$$a_e u'_e = \sum a_{\mathrm{nb}} u'_{\mathrm{nb}} + (p'_P - p'_E)A_e \quad (47)$$

$$a_n v'_n = \sum a_{\mathrm{nb}} v'_{\mathrm{nb}} + (p'_P - p'_N)A_n \quad (48)$$

At this point an approximation, i.e. the second approximation in the SIMPLE algorithm mentioned above, is introduced: dropping the terms $\sum a_{\mathrm{nb}} u'_{\mathrm{nb}}$ and $\sum a_{\mathrm{nb}} v'_{\mathrm{nb}}$ in the above equations to simplify the expressions. Then we obtain

$$u'_e = d_e(p'_P - p'_E), \quad v'_e = d_n(p'_P - p'_N) \quad (49)$$

where d_e, d_n are defined as

$$d_e = \frac{A_e}{a_e}; \quad d_n = \frac{A_n}{a_n} \quad (50)$$

Then the improved velocities are rewritten as follows:

$$u_e = u_e^* + d_e(p'_P - p'_E)$$
$$v_n = v_n^* + d_n(p'_P - p'_N) \quad (51)$$

Substitution of the improved velocities of equations (50) and (51) into continuity equation, the equation for the pressure correction term is then derived

$$a_P p'_P = \sum a_{nb} p'_{nb} + b \quad (52)$$

where

$$b = (\rho u^* A)_w - (\rho u^* A)_e + (\rho v^* A)_s - (\rho v^* A)_n \quad (53)$$

In equation (53) the coefficients are the same as those in the equation (16) except the b-term, where the velocities take the values of the previous iteration, rather than the intermediate solutions.

The solution procedure of the SIMPLER algorithm is as follows:

- guess a initial velocity field u^0, v^0;
- calculate the coefficients of the discretized momentum equations and the pseudo-velocities $\hat{u}$ and $\hat{v}$ by following equations:

$$\hat{u} = \frac{\sum a_{nb} u_{nb}^0 + b}{a_e}; \quad \hat{v} = \frac{\sum a_{nb} v_{nb}^0 + b}{a_n} \quad (54)$$

- solve pressure equation to get p^*;
- solve the discretized momentum equations with p^* to get u^* and v^*;
- solve the pressure correction equation to get p';
- correct the velocities by equations (50) and (51);
- solve the discretized equations for other scalar variables if necessary; and
- Return to step 2 until convergence condition is satisfied.

It is to be noted that in the SIMPLER algorithm, the pressure correction term is only used to correct the velocities, but not used to correct the pressure. The pressure correction values are over predicted by solving equation (52), because the effects of the velocity corrections at neighboring grid points are totally neglected. The obtained pressure correction values are appropriate to correct the velocities, but not to pressure values. Since the discretized equations are all solved by iterative method, the solutions of velocities of the current iteration are based on the coefficients and source term determined by the solutions of the last iteration. In particular, the pressure field is solved according to the velocities of the previous iteration, and it is in this aspect that the solved velocity field and the pressure field are not consistent. It is the authors' consideration that the pressure filed may be further revised within the iteration that the consistency between the two fields can be refined.

6.2.2. The SIMPLEC Method

We obtain the following velocity correction equations:

$$u'_e = d'_e(p'_P - p'_E), \quad v'_e = d'_n(p'_P - p'_N) \quad (55)$$

where

$$d'_e = \frac{A_n}{(a_e - \sum a_{nb})}, \quad d'_n = \frac{A_n}{(a_n - \sum a_{nb})} \quad (56)$$

The SIMPLEC algorithm alleviates in some degree the effect of the second approximation in the SIMPLE algorithm. The pressure correction equation in the SIMPLEC algorithm is the same as that in the SIMPLE algorithm except that the d-terms are calculated from equation (50). The solution procedure of the SIMPLEC algorithm is identical to that of SIMPLE (Tao,2001; Patankar, 1980).

6.2.3. The CSIMPLER Algorithm

Now we incorporate the major idea of the SIMPLEC algorithm into the SIMPLER algorithms as follows:

The pressure correction equation of the SIMPLEC is adopted in the SIMPLER algorithm, i.e. the d-terms are calculated from equation (56); the pressure is also corrected after the pressure correction equation is solved:

$$p = p^* + \alpha_p p' \quad (57)$$

Where a_p is the relaxation factor for the pressure correction. When $a_p<1$ it is under relaxation of the pressure correction, while $a_p>1$ implies the over relaxation. Our practices have shown that over relaxation of the pressure correction term is often useful for the acceleration of the convergence procedure which will be discussed later.

By adopting above two treatments into SIMPLER algorithm and keeping the solution procedures the same, the resulting solution algorithm is called consistent SIMPLER, simplified by CSIMPLER, [Ashraya et al., 2007].

We consider that the introduction of the pressure correction term into the present pressure will improve the coupling between velocity and pressure, hence, may accelerate the convergence of the iterative process. It is to be noted that for any existing code based on the SIMPLER algorithm the implementation of the CSIMPLER algorithm is very simple and easy.

7. Conclusion

The objective of this article is to illustrate all steps to find the solution of a heat transfer and fluid flow problem using CFD packages. In order to present them we gave only the necessary information in theoretical fluid dynamics. Searching, we can find may research papers and

books specialize in this area.

Nomeclature

g: Acceleration due to gravity (m/s^2)
k: Thermal conductivity (W/m' K)
p: Pressure (Pa)
Pr : Prandtl number
Re: Reynolds number
Sc: Schmidt number
T: Temperature (K)
t: time (s)
u,v,w Velocity (m/s)
Γ: diffusion flux coefficient
R: rate of chemical generation
ρ: density (kg/m^3)
μ: viscosity (kgm/ms)
Φ: scalar variable
ε: dissipation rate of turbulent kinetic energy (m^2/s^3)
k: turbulent kinetic energy (m^2/s^2)
c_1,c_2 constants in the turbulence model
c_μ, σ_ε, σ_k empirical constants
c_p heat capacity (J/kgr k)
r volume fraction
σ Prandtl number

Subscripts and Superscripts

eff: Sum of molecular and turbulent contributions
ref: Reference quantity
i,j: the three spatial x,y,z direction
s south
n north
e east
w west
p center point
t top
b bottom

References

[1] Acharya S., B.R. Baliga, K. Kari, J.Y. Murthy, C. Prakash and S.P. Vanka. 2007. Pressure based finite volume methods in Computational Fluid dynamics. J. Heat Trasfer, 129(4), pp. 407-424

[2] Bradshaw P. and T. Cebeci, Engineering Calculation Methods. Academic Press, London (1978).

[3] Bradshaw P., D. H. Ferris and N. P. Atwell, Calculation of boundary-layer development using the turbulent energy equation. J. Fluid Mech. 28, 593~i16 (1967).

[4] Bradshaw P., The understanding and prediction of turbulent flow. Aeronaut. J. 76, 403-417 (1972).

[5] Cebeci T., K. C. Chang and P. Bradshaw, Solution of hyperbolic system of turbulence-model equations by the "Box" scheme. Comput. Meth. appl. Mech. Engng 22, 213-227 (1980).

[6] Chatwani, A.U. and Turan, A. (1991), "Improved pressure-velocity coupling algorithm based on global residual norm", Numer. Heat Transfer. Part B, Vol. 20, pp. 115-23.

[7] Connell, S.D. and Stow, P. (1986), "The pressure correction methods", Comput. Fluids, Vol. 14, pp. 1-10.

[8] Donaldson C. du P., Calculation of turbulent shear flows for atmospheric and vortex motions. AIAA J. 10, 4-12 (1972).

[9] Gjesdal, T. and Lossius, M.E.H. (1997), "Comparison of pressure correction smoothers for multigrid solution of incompressible flow", Int. J. Numer. Methods Fluids, Vol. 25, pp. 393-405.

[10] Gosman A. D., W. M. Pun, A. K. Runchal, D. B. Spalding and M. Wolfshtein, Heat and Mass Transfer in Recirculating Flows. Academic Press, London (1969).

[11] Hirsch Charles. 2007. Numerical computation of internal and external flows. The fundamentals of Computational Fluid Dynamics. John wiley and Sons Publ. LTd.

[12] Hoffman Klaus A. and Steve J. Chiang. Computational Fluid Dynamics, volume II. By Engineering Education System, USA, 2000

[13] Issa, R.I. (1985), "Solution of implicitly discretized fluid flow equation by operator-splitting", J. Comput. Physics, Vol. 62, pp. 40-65.

[14] Lam G.K.G. and K. Bremhorst. 1981. A modified form of the k-ε model for predicting wall turbulence. ASME, J. Fluids Eng. Vol.103, pp. 456-460

[15] Launder B. E. and D. B. Spalding, Turbulence models and their applications to the prediction of internal flows. Heat Fluid Flow 2, 43-54 (1972).

[16] Launder B. E., Stress transport closures--into the third generation. Syrup. Turbulent Shear Flows 3, 259-266 (1982).

[17] Launder B. E., W. C. Reynolds, W. Rodi, J. Mathieu and D. Jeandel, Turbulence Models and Their Applications, Vol. 2. Eyrolles, Paris (1984).

[18] Launder; B. E. and Spalding, D. B. (1974). The Numerical Computation of Turbulent Flow, Comp. Methods Appl. Mech. Eng., val. 3, p. 269-289.

[19] Lee, S.L. and Tzong, R.Y. (1992), "Artificial pressure for pressure-linked equation", Int. J. Heat Mass Transfer, Vol. 35, pp. 2705-16.

[20] Lekshminarayana B., Turbulence modeling for complex flows. AIAA. paper AIAA-85 1652 (1985).

[21] Liu X.L., W.Q. Tao and Y.L. He (2005). "A simple method for improving the SIMPLER algorithm for numerical simulations of incompressible fluid flow and heat transfer problems.", Int. J. heat Mass Transfer, vol. 22, no. 8, pp. 921-939

[22] Lopez-Ramirez S., R.D. Morales, J.A. Romero Serrano. 2000. Numerical simulation of the effects of buoyancy forces and flow control devices on fluid flow and heat transfer phenomena of liquid steel in a tundish. Num. Heat Transf., A, vol.37, pp. 37-69

[23] Lumley J. L., Computational modeling of turbulent flows. Adv. appl. Mech. 18, 123-176 (1978).

[24] Lumley J. L., Turbulence modeling. J. Appl. Mech. 50, pp. 1097-1103 (1983).

[25] Marvin J. G, Turbulence modeling for computational aerodynamics. AIAA J. 21, 941-955 (1983).

[26] Meller G. L. and T. Yamada, Development of a turbulence closure model for geophysical fluid problems. Rev. Geophys. space Phys. 20, 851-875 (1982).

[27] Mellor G. L, and H. J. Herring, A survey of the mean turbulent field closure models. AIAA J. 11, 590-599 (1973).

[28] Monson, D.J., Seegmiller, H.L., McConnaughey and Chen, Y.S. (1990), "Comparison of Experiment with calculations using curvature – corrected zero and two equations turbulence models for a two-dimensional U-duct", AIAA, pp. 1390-1484.

[29] Moukalled, F. and Darwish, M. (2000), "A unified formulation of the segregated class of algorithm for fluid flow at all speeds", Numer. Heat Transfer B, Vol. 37, pp. 103-39.

[30] Murphy J. D., Turbulence Modeling. NASA TM 85889 (1984).

[31] Murthy, J.Y., Minkowycz, W.J., Sparrow, E.M., and Mathur, S.R., 2007. "Survey of Numerical Methods," Handbook of Numerical Heat Transfer, 2nd ed., eds., Minkowycz, W.J., Sparrow, E.M., and Murthy, J.Y., pp. 3-51, Wiley, New York.

[32] Patankar S.V. (1980) "Numerical heat transfer and fluid flow". By Hemisphere Publishing Corporation.

[33] Patankar, S. V. and Spalding, D. B. (1970). Heat and Mass Transfer in Boundary Layers, 2d ed., Intertext, London.

[34] Patankar, S.V. (1981), "A calculation procedure for two-dimensional elliptic situations", Numer. Heat Transfer, Vol. 4, pp. 409-25.

[35] Patankar, S.V. and Spalding, D.B. (1972), "A calculation procedure for heat mass and momentum transfer in three dimensional parabolic flows", Int. J. Heat Mass Transfer, Vol. 15, p. 1787.

[36] Patankar, S.V. (1975). Numerical Prediction of Three Dimensional Flows, in B. E. Launder (ed.), Studies in Convection: Theory, Measurement and Applications, vol. 1, Academic, New York.

[37] Patel M.K. and N.C. Markatos. An evaluation of eight discretization schemes for two dimensional convection diffusion equations. Int. J. Num. Meth. Fluids, vol. 6, pp. 129-154, (1986).

[38] Patel V. C., W. Rodi and G. Scheuerer, Turbulence models for near wall and low Reynolds number flows: a review. AIAA J. 23, 1308-1319 (1985).

[39] Raithby, G.D. and Schneider, G.E. (1988), "Elliptic system: finite difference method II", in Minkowycz, W.J., Sparrow, E.M., Pletcher, R.H. and Schneider, G.E. (Eds), Handbook of Numerical Heat Transfer, Wiley, New York, NY, pp. 241-89.

[40] Reynolds W. C., Computation of turbulent flows. Ann. Rev. Fluid Mech. 8, 183-208 (1976).

[41] Rodi W., Examples of turbulence models for incompressible flows. AIAA J. 20, 872-879 (1982).

[42] Rotta J. C., Prediction methods for turbulent flows. Von Karman Institute for Fluid Dynamics, Lecture Series 76 (1975).

[43] Sheng, Y., Shoukri, M., Sheng, G. and Wood, P. (1998), "A modification to the SIMPLE method for buoyancy-driven flows", Numer. Heat Transfer, Vol. 33, pp. 65-78.

[44] Shyy, W. and Mittal, R. (1998), "Solution methods for the incompressible Navier-Stokes equations", in Johnson, R.W. (Ed.), The Handbook of Fluid Dynamics, CRC Press, Boca Raton, FL, pp. 31.1-31.33.

[45] Spalding D. B., Turbulence Models, a Lecture Course. Imperial College of Science and Technology (1982).

[46] Sunden B. 2013. On computational heat transfer in industrial applications. ASME 2013 Heat Transfer Summer Conf. Collocated with the ASME 2013 7th Int. Conf. on Energy Sustainability and the ASME 2013 11th Int. Conf. on Fuel Cell Science, Engineering and Technology, HT 2013

[47] Szekely J., O.J. Ilegbussi and N.El-Kaddah. 1987. The mathematical modeling of complex fluid flow phenomena in tundishes. Physicochemical Hydrodynamics, vol.9, pp. 453-472

[48] Tao, W.Q. (2001), Numerical Heat Transfer, 2nd ed., Xi'an Jiaotong University, Xi'an, p. 211. van Doormaal, J.P. and Raithby, G.D. (1984), "Enhancement of SIMPLE method for predicting incompressible fluid flows", Numer. Heat Transfer, Vol. 7, pp. 147-63.

[49] van Doormaal, J.P. and Raithby, G.D. (1985), "An evaluation of the segregated approach for predicting incompressible fluid flow", Paper 85-HT-9, ASME.

[50] Wen, X. and Ingham, D.B. (1993), "A new method for accelerating the rate of convergence of the SIMPLE-like algorithm", Int. J. Numer Methods Fluids, Vol. 17, pp. 385-400.

[51] Wilcox D. C. and M. W. Rubesin, Progress in turbulence modeling for complex flow fields including the effect of compressibility. NASA TP 1517 (1980).

[52] Yahkot Y. and Orszag S.A., 1992. Development of turbulence models for shear flows by a double expansion technique. Phys. Fluids A, vol. 4, pp. 1510-1520

[53] Yen, R.H. and Liu, C.H. (1993), "Enhancement of the SIMPLER algorithm by an additional explicit corrector step", Numer. Heat Transfer B, Vol. 24, pp. 127-41.

[54] Yu, B., Ozoe, H. and Tao, W.Q. (2001), "A modified pressure-correction scheme for the SIMPLER method, MSIMPLER", Numer. Heat Transfer B, Vol. 39, pp. 439-49. Further reading.