fluids

MDPI

Article

An Efficient Strategy to Deliver Understanding of Both Numerical and Practical Aspects When Using Navier-Stokes Equations to Solve Fluid Mechanics Problems

Desmond Adair [1,*] and Martin Jaeger [2]

[1] School of Engineering and Digital Sciences, Nazarbayev University, Nur-Sultan 010000, Kazakhstan
[2] College of Sciences and Engineering, University of Tasmania, TAS 7001, Australia; m.jaeger@ack.edu.kw
* Correspondence: desmond.adair@utas.edu.au; Tel.: +7-7172-70-6531

Received: 7 July 2019; Accepted: 27 September 2019; Published: 1 October 2019

Abstract: An efficient and thorough strategy to introduce undergraduate students to a numerical approach of calculating flow is outlined. First, the basic steps, especially discretization, involved when solving Navier-Stokes equations using a finite-volume method for incompressible steady-state flow are developed with the main aim being for the students to follow through from the mathematical description of a given problem to the final solution of the governing equations in a transparent way. The well-known 'driven-cavity' problem is used as the problem for testing coding written by the students, and the Navier-Stokes equations are initially cast in the vorticity-streamfunction form. This is followed by moving on to a solution method using the primitive variables and discussion of details such as, closure of the Navier-Stokes equations using turbulence modelling, appropriate meshing within the computation domain, various boundary conditions, properties of fluids, and the important methods for determining that a convergence solution has been reached. Such a course is found to be an efficient and transparent approach for introducing students to computational fluid dynamics.

Keywords: computational fluid dynamics; finite-volume method; discretization; turbulence modelling; meshing

1. Introduction

Computational Fluid Dynamics (CFD) is the simulation of transport phenomena, reacting systems, heat transfer, etc., using modelling, i.e., mathematical physical problem formulation and numerical solution, which include discretization methods, solvers, numerical parameters and mesh generation. The efficient and thorough development, implementation and evaluation of a suitable curriculum, which incorporates both the theoretical and practical aspects of CFD, for undergraduate students as an extension of their knowledge in the thermo-fluids area is warranted and necessary in modern engineering programmes. Numerical methods have an advantage over analytical methods of solution in that analytical methods tend to be restrictive by commonly requiring simple geometries, unrealistic assumptions and until lately assumed linearity, whereas real engineering problems usually have quite complex geometries and highly non-linear phenomena. However, despite the advantages of numerical methods, the latter may miss some finer aspects of fluid dynamics which, from the equations of analytical methods, may be more transparent, for example the question of uniqueness of weak solutions to Navier-Stokes equations in three dimensions. Analytical methods also often task a student to consider the treatment of any nonlinear terms found in partial differential equations, hence encouraging a student's interest in this important topic.

Many areas of engineering education already use mathematical modelling as a teaching tool, for example, for diesel engine studies [1], in fluid mechanics and heat transfer [2–4] and in chemical reactions [5]. It is important when introducing a complex topic such as CFD to have educational user-friendly interfaces, usually in the form of dialogue graphical user interfaces (GUIs), and some work has already been done on this [6,7], where the general aspects of the three main processes of CFD, i.e., building the problem using a pre-processor, solving the problem, and displaying the results using a post-processor were considered. However, although the education benefits associated with integrating computer-assisted learning and simulation technology into undergraduate engineering course are great [8], and, computational fluid dynamics has revolutionized approaches to research and design, its incorporation into the teaching of undergraduate transport phenomena has still difficulties. One of these difficulties, and the one addressed in this paper, is to efficiently and thoroughly combine a sufficient understanding of the theoretical aspects of CFD with its practical use. Another is to engender a critical understanding of obtained results.

To help with an initial understanding of the process of solving Navier-Stokes and related transport equations the symbolic software, *Mathematica* [9] was employed to demonstrate the mathematical aspects of some of the basic steps involved, for example meshing, discretization, application of boundary conditions and the development of a simple solution algorithm. Two-dimensional incompressible flows were calculated for simplicity and the Navier-Stokes, at this stage of the course, were recast in an alternative form, i.e., in terms of streamfunction and vorticity. In many applications, the vorticity-streamfunction form of the Navier-Stokes equations provides a better insight into the physical mechanisms driving the flow than the "primitive variable" formulation which is in terms of the mean velocities u, v and pressure, p. The streamfunction and vorticity formulation is also useful for numerical work, since it avoids some problems resulting from discretization of the continuity equation.

The second part of this work outlines how to best introduce the practical aspects and emphasizes concerns which must be remembered when using CFD. Issues arise when introducing simulation into a curriculum, including learning vs. research objectives, usability vs. predetermined objectives and student demographics. A proper balance must be sought for between these competing objectives; for example, it is just as important that students are taught the systematic and practical ways of using a CFD package in a general sense, as well as achieving a specific result. There is evidence from previous studies that using simulation enhances the curriculum [4]; there is increased learning efficiency and understanding [10,11]; there is effectiveness of hands-on learning methods [12]; and, it is effective to use a combination of physical and simulation laboratories [7].

Other concerns when introducing CFD to students are, for example, when to introduce during the programme, how much to introduce and how much mathematical and thermo-fluid backgrounds are necessary [13]? When a student first commences studying CFD a lot of new knowledge and required skills descend on them from several different curriculum areas, which must be married together hence rendering a steep learning curve. If careful planning is not carried out, this curve may become overwhelming for many of the students. It is also important to ensure that the CFD GUIs are not treated as 'black boxes', without knowledge of what is going on inside the computer software. This is especially true as many commercial codes now provide the opportunity of attaching in-house coding to the main core of the software hence providing a gateway for enhanced research and code development [14]. Without knowledge of CFD fundamentals and the core coding, code development would not be possible. It is important that students maintain a 'feeling' for physical phenomena, remember the necessary assumptions made for the mathematical modelling, and, very importantly, know of the need to verify and validate the calculated results [15]. It has been observed that successful implementation of CFD often requires a re-focus of course objectives and skills taught, and a re-structuring of the course curriculum [16]. An important practical consideration is to engender into the students a healthy skepticism concerning the calculated results and a willingness to be critical of their results until validated [17].

In this paper the mathematical aspects of CFD are first introduced, with a particular emphasize on how to discretize equations. This is followed by a discussion of practical details such as closure of the Navier-Stokes equations using turbulence modelling, appropriate meshing within the computation domain, various boundary condition, properties of fluids, and the important methods for determining that a convergence solution has been reached. An overall description of the course is then given. The paper finishes with conclusions drawn and possible work needed for future development.

2. Numerical Elements of Computational Fluid Dynamics (CFD) Teaching

2.1. Introduction

In this part of the course, which lasted for four weeks, the purpose was to develop the basic steps involved in solving Navier-Stokes equations using a finite-volume approach for incompressible steady-state flow. The students can actually see the development of the mathematical description linked to a programming environment solution process, so often hidden in commercial code used for training CFD students. The students start with discretization of the governing equations, write their own coding to generate a mesh, apply boundary conditions, provide a suitable solution algorithm and to generate contour and x-y plots.

The students had already successfully completed courses on matrix algebra, vector calculus, ordinary and partial differential equations, with the latter solved using a variety of numerical methods, as well as preliminary courses on fluid mechanics and heat transfer. The students also have reasonable experience of using *Mathematica*, which is a necessary prerequisite [18]. Required reading for the course included LeVeque [19] and Ferziger and Peric [20].

In the following sections the theory taught to the students in preparation for them writing appropriate coding is summarized.

2.2. Navier-Stokes Equations

The governing equations for modelling fluid flow are the Navier-Stokes equations and the continuity equation. These laws of motion are valid for all phases including liquids, gases and solids, but there is a fundamental difference between fluids and solids in that fluids have no limit to their distortion. The analysis of fluid flow needs to take into account such distortions. The Navier-Stokes equations can be derived by the consideration of the dynamic equilibrium of a small fluid element and they state that surface and body forces are in equilibrium with inertial forces acting on the fluid element. On consideration of incompressible flow, where the density is constant, the Navier-Stokes equations for three-dimensional flow are as shown in Equations (1)–(4). Here body forces have not been included.

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}=0 \tag{1}$$

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}+w\frac{\partial u}{\partial z}=-\frac{1}{\rho}\frac{\partial p}{\partial x}+\nu\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+\frac{\partial^2 u}{\partial z^2}\right) \tag{2}$$

$$\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}+w\frac{\partial v}{\partial z}=-\frac{1}{\rho}\frac{\partial p}{\partial y}+\nu\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}+\frac{\partial^2 v}{\partial z^2}\right) \tag{3}$$

$$\frac{\partial w}{\partial t}+u\frac{\partial w}{\partial x}+v\frac{\partial w}{\partial y}+w\frac{\partial w}{\partial z}=-\frac{1}{\rho}\frac{\partial p}{\partial z}+\nu\left(\frac{\partial^2 w}{\partial x^2}+\frac{\partial^2 w}{\partial y^2}+\frac{\partial^2 w}{\partial z^2}\right) \tag{4}$$

where u, v, w are the velocity components in the x, y, z directions, ρ is the fluid density, p is the pressure and ν is the kinematic viscosity.

Most flows encountered in engineering problems are turbulent because it is mostly impossible to stop transition from laminar to turbulent flow, or, by intention, as turbulence may aid the engineering application. For turbulent flows however, the variation of the quantities to be calculated with time is often so rapid and with a high degree of randomness that this variation is of little engineering relevance. Hence the quantities are averaged and most often the Reynolds-averaged Navier-Stokes equations as shown in Equations (5)–(8) are used.

$$\frac{\partial \overline{u}}{\partial x}+\frac{\partial \overline{v}}{\partial y}+\frac{\partial \overline{w}}{\partial z}=0 \tag{5}$$

$$\frac{\partial \overline{u}}{\partial t}+\overline{u}\frac{\partial \overline{u}}{\partial x}+\overline{v}\frac{\partial \overline{u}}{\partial y}+\overline{w}\frac{\partial \overline{u}}{\partial z}=-\frac{1}{\rho}\frac{\partial \overline{p}}{\partial x}+\frac{\partial}{\partial x}\left(\nu\frac{\partial \overline{u}}{\partial x}-\overline{u'u'}\right)+\frac{\partial}{\partial y}\left(\nu\frac{\partial \overline{u}}{\partial \overline{y}}-\overline{u'v'}\right)+\frac{\partial}{\partial z}\left(\nu\frac{\partial \overline{u}}{\partial z}-\overline{u'w'}\right) \tag{6}$$

$$\frac{\partial \overline{v}}{\partial t}+\overline{u}\frac{\partial \overline{v}}{\partial x}+\overline{v}\frac{\partial \overline{v}}{\partial y}+\overline{w}\frac{\partial \overline{v}}{\partial z}=-\frac{1}{\rho}\frac{\partial \overline{p}}{\partial y}+\frac{\partial}{\partial x}\left(\nu\frac{\partial \overline{v}}{\partial x}-\overline{u'v'}\right)+\frac{\partial}{\partial y}\left(\nu\frac{\partial \overline{v}}{\partial \overline{y}}-\overline{v'v'}\right)+\frac{\partial}{\partial z}\left(\nu\frac{\partial \overline{v}}{\partial z}-\overline{v'w'}\right) \tag{7}$$

$$\frac{\partial \overline{w}}{\partial t}+\overline{u}\frac{\partial \overline{w}}{\partial x}+\overline{v}\frac{\partial \overline{w}}{\partial y}+\overline{w}\frac{\partial \overline{w}}{\partial z}=-\frac{1}{\rho}\frac{\partial \overline{p}}{\partial y}+\frac{\partial}{\partial x}\left(\nu\frac{\partial \overline{w}}{\partial x}-\overline{u'w'}\right)+\frac{\partial}{\partial y}\left(\nu\frac{\partial \overline{w}}{\partial \overline{y}}-\overline{v'w'}\right)+\frac{\partial}{\partial z}\left(\nu\frac{\partial \overline{w}}{\partial z}-\overline{w'w'}\right) \tag{8}$$

As can be observed, the instantaneous quantities are replaced by their corresponding time-averaged quantities. Extra terms appear due to the averaging process, e.g., $-\overline{u'v'}$ and these behave like stress terms, often called Reynolds stresses, which need further equations if the system of equations is to be closed. There have, over this last 60–70 years, been many suggestions as to the best way for equation closure, none of which as yet are completely satisfactory.

2.3. Vorticity-Streamfunction Governing Equations

For this work, the Navier-Stokes equations are caste into the vorticity-streamfunction form, where the streamfunction is defined as

$$\psi_A(P)=\int_A^P \mathbf{u}\cdot\mathbf{n}\,\mathrm{d}s \tag{9}$$

and where the integral has to be evaluated along a curve C from the arbitrary but fixed point A to point P, $\mathbf{u}$ is the velocity vector, and $\mathbf{n}$ is the unit normal on the curve from A to P. The streamfunction is regarded as a function of the spatial coordinates only. Streamlines are lines that are everywhere tangential to the velocity field, and hence ψ is constant along the streamlines. Invoking the integral incompressibility constraint for an infinitesimally small triangle shows that ψ is related to the two Cartesian velocity components u and v via

$$u=\frac{\partial\psi}{\partial y};\quad v=-\frac{\partial\psi}{\partial x} \tag{10}$$

Flows that are specified by a streamfunction automatically satisfy the continuity equation since

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=\frac{\partial}{\partial x}\left(\frac{\partial\psi}{\partial y}\right)-\frac{\partial}{\partial y}\left(\frac{\partial\psi}{\partial x}\right)=0 \tag{11}$$

For two-dimensional flows, the vorticity vector $\omega=\nabla\times\mathbf{u}$ only has one non-zero component (in the z direction; i.e., $\omega=\omega e_z$ where

$$\omega=\frac{\partial v}{\partial x}-\frac{\partial u}{\partial y} \tag{12}$$

Using the definition of the velocities in terms of the streamfunction shows that

$$\omega=\frac{\partial}{\partial x}\left(-\frac{\partial\psi}{\partial x}\right)-\frac{\partial}{\partial y}\left(\frac{\partial\psi}{\partial y}\right)=-\nabla^2\psi \tag{13}$$

The Navier-Stokes equations for incompressible steady-state flow in vorticity-streamfunction form are

$$\frac{\partial\psi}{\partial x}\frac{\partial\omega}{\partial y}-\frac{\partial\psi}{\partial y}\frac{\partial\omega}{\partial x}=\frac{1}{\mathrm{Re}}\left(\frac{\partial^2\omega}{\partial x^2}+\frac{\partial^2\omega}{\partial y^2}\right) \tag{14}$$

$$\frac{\partial^2\psi}{\partial x^2}+\frac{\partial^2\psi}{\partial y^2}=-\omega \tag{15}$$

where Re is the Reynolds number.

It should be mentioned that the above formulation is only valid in two-dimensions and some alternation is needed when three-dimensional flow is considered.

2.4. Boundary Conditions

The boundary conditions needed to solve Equations (14) and (15) for the classical computational fluid dynamics case, namely the lid-cavity flow [21], are shown on Figure 1.

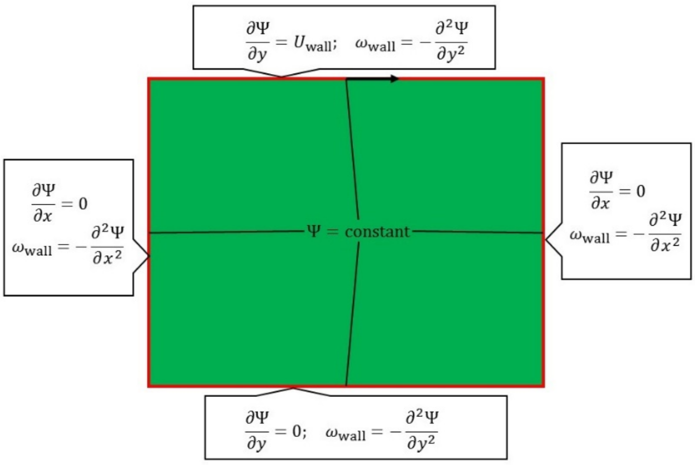
$\frac{\partial\Psi}{\partial y}=U_{\mathrm{wall}};\quad \omega_{\mathrm{wall}}=-\frac{\partial^2\Psi}{\partial y^2}$
$\frac{\partial\Psi}{\partial x}=0$
$\omega_{\mathrm{wall}}=-\frac{\partial^2\Psi}{\partial x^2}$
$\Psi=\mathrm{constant}$
$\frac{\partial\Psi}{\partial x}=0$
$\omega_{\mathrm{wall}}=-\frac{\partial^2\Psi}{\partial x^2}$
$\frac{\partial\Psi}{\partial y}=0;\quad \omega_{\mathrm{wall}}=-\frac{\partial^2\Psi}{\partial y^2}$

Figure 1. Summary of boundary conditions for cavity with moving lid. (Reproduced with permission from [22]).

A rectangular box, where the lid is allowed to move in the horizontal plane from left to right, is simulated with the moving lid driving the fluid within the box. In Figure 1 the boundary conditions needed for solution are summarized in terms of vorticity and streamfunction. The streamfunction is constant at the walls, as its gradient is velocity, which is zero relative to a given wall (the no-slip condition). The vorticity boundary condition for each wall is also shown in each of the outer boxes and are derived from the streamfunction.

2.5. Discretization of Transport Equations

Before describing the discretization of the governing equations, it is important to note that for convenience Equations (14) and (15) together with the boundary conditions were non-dimensionalized using

$$\widetilde{x}=\frac{x}{b};\ \widetilde{y}=\frac{y}{a};\ \widetilde{\psi}=\frac{\psi}{Vb};\ \widetilde{\omega}=\frac{\omega b}{V};\ \gamma=\frac{b}{a};\ \mathrm{Re}=\frac{Vb}{\nu}$$

where a and b are the height and width of the cavity, respectively, and V is a reference velocity. The resultant governing equations in vorticity-streamfunction format become

$$\frac{\partial^2\widetilde{\omega}}{\partial\widetilde{x}^2}+\gamma^2\frac{\partial^2\widetilde{\omega}}{\partial\widetilde{y}^2}=Re\gamma\left(\frac{\partial\widetilde{\psi}}{\partial\widetilde{y}}\frac{\partial\widetilde{\omega}}{\partial\widetilde{x}}-\frac{\partial\widetilde{\psi}}{\partial\widetilde{x}}\frac{\partial\widetilde{\omega}}{\partial\widetilde{y}}\right) \tag{16}$$

$$\frac{\partial^2\widetilde{\psi}}{\partial\widetilde{x}^2}+\gamma^2\frac{\partial^2\widetilde{\psi}}{\partial\widetilde{y}^2}=-\widetilde{\omega} \tag{17}$$

defined in the domain $0 \leq \widetilde{x} \leq 1,\ 0 \leq \widetilde{y} \leq 1$.

So as to include the boundary nodes of the cavity, central differencing is used to discretize Equations (16) and (17) with the number of nodes in the x and y directions set at n_x and n_y, respectively. The mesh is a regular Castesian grid with the nodes equally spaced at a distance h, as illustrated on Figure 2.

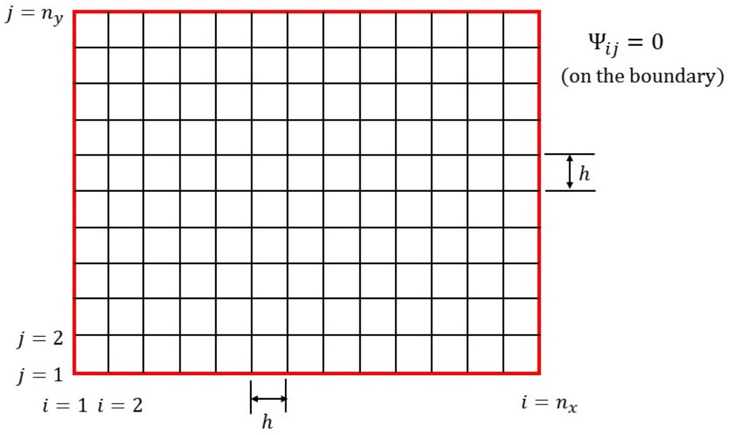

Figure 2. Typical Cartesian mesh used for the lid-cavity flow. (Reproduced with permission from [21]).

To discretize Equations (16) and (17) the following central-difference approximations were introduced for spatial dimensions, obtained by adding or subtracting one Taylor series from another and invoking the intermediate value theorem. As can be seen from Equation (18), the order of accuracy for both the first and second derivatives is $O(h^2)$.

$$\begin{aligned} \frac{\partial f(x)}{\partial x} &= \frac{f(x+h) - f(x-h)}{2\,h} + O\left(h^2\right) \\ \frac{\partial f(x)}{\partial x} &= \frac{f(x+h) - f(x-h)}{2\,h} + O\left(h^2\right) \end{aligned} \tag{18}$$

Using the approximations from Equation (18), the discretized advection-diffusion equation (Equation (16)) at the (i, j) node is

$$\begin{aligned} &\frac{\omega_{i+1,j} - 2\omega_{i,j} + \omega_{i-1,j}}{h^2} + \gamma^2 \frac{\omega_{i,j+1} - 2\omega_{i,j} + \omega_{i,j-1}}{h^2} \\ &\qquad = \mathrm{Re}\gamma\left(\frac{\psi_{i,j+1} - \psi_{i,j-1}}{2\,h}\frac{\omega_{i+1,j} - \omega_{i-1,j}}{2\,h} - \frac{\psi_{i+1,j} - \psi_{i-1,j}}{2\,h}\frac{\omega_{i,j+1} - \omega_{i,j-1}}{2\,h}\right) \end{aligned} \tag{19}$$

The elliptic equation (Equation (17)) at the (i, j) node is

$$\frac{\psi_{i+1,j} - 2\psi_{i,j} + \psi_{i-1,j}}{h^2} + \gamma^2 \frac{\psi_{i,j+1} - 2\psi_{i,j} + \psi_{i,j-1}}{h^2} = -\omega\left(x_i, y_j\right) \tag{20}$$

The equations are applied at the internal nodes of the Cartesian mesh; i.e., $2 \leq i \leq n_{x-1}$ and $2 \leq i \leq n_{y-1}$.

2.6. Solution Algorithm

The iterative solution used here is based on Burggraf's proposal [23]. The solution method uses residual functions; i.e., if the values of $\psi_{i,j}$ and $\omega_{i,j}$ are exact on the nodes spanned by the residual functions $\mathcal{R}_{ij}$ and $\mathcal{L}_{ij}$, then

$$\mathcal{R}_{ij}^{k+1} = \mathcal{L}_{ij}^{k+1} = 0 \tag{21}$$

where

$$\mathcal{R}_{ij}^{k+1} = \frac{1}{2(1+\gamma^2)}\left(\psi_{i+1,j}^{k} + \psi_{i-1,j}^{k+1} + \gamma^2\left(\psi_{i,j+1}^{k+1} + \psi_{i,j-1}^{k+1}\right)\right) + h^2\omega_{i,j}^{k} - \psi_{i,j}^{k} \tag{22}$$

$$\begin{aligned}\mathcal{L}_{ij}^{k+1} = \frac{1}{2(1+\gamma^2)}&\left(\left(\omega_{i+1,j}^{k} + \omega_{i-1,j}^{k} + \gamma^2\left(\omega_{i,j+1}^{k} + \omega_{i,j-1}^{k}\right)\right)\right.\\ &\left.-\frac{\mathrm{Re}}{4}\gamma\left(\left(\psi_{i,j+1}^{k} - \psi_{i,j-1}^{k+1}\right)\left(\omega_{i+1,j}^{k} - \omega_{i-1,j}^{k+1}\right) - \left(\psi_{i+1,j}^{k} - \psi_{i-1,j}^{k+1}\right)\left(\omega_{i,j+1}^{k} - \omega_{i,j-1}^{k+1}\right)\right)\right) - \omega_{i,j}^{k}\end{aligned} \tag{23}$$

The following fixed-point iterative procedure, based on the Gauss-Seidel scheme (see Appendix A for a short general explanation), is then constructed

$$\begin{aligned}\psi_{i,j}^{k+1} &= \mathcal{F}^{k}\left(\psi^{(k)}, \psi^{(k+1)}, \omega^{(k)}, \omega^{(k+1)}\right) \equiv \psi_{i,j}^{k} + p\mathcal{R}_{ij}^{k}\\ \omega_{i,j}^{k+1} &= \mathcal{G}^{k}\left(\psi^{(k)}, \psi^{(k+1)}, \omega^{(k)}, \omega^{(k+1)}\right) \equiv \omega_{i,j}^{k} + p\mathcal{L}_{ij}^{k}\end{aligned} \tag{24}$$

where p is a relaxation parameter lying in the range $0 < p \leq 1$, and $k+1$ and k refer to the respective iterations. The use of this relaxation parameter is to improve the stability of a particular computation, particularly when solving steady-state problems.

The boundary conditions now need to be determined. For the streamfunction

$$\psi_{i,1} = 0 \text{ for } i = 1, \ldots, n_x,$$

$$\psi_{n_x,j} = 0 \text{ for } j = 1, \ldots, n_y,$$

$$\psi_{i,n_y} = 0 \text{ for } i = 1, \ldots, n_x,$$

$$\psi_{1,j} = 0 \text{ for } j = 1, \ldots, n_y.$$

For vorticity, the following needs to be considered for, say, the left wall of the cavity with one node outside the computational domain.

$$\omega_{i,j} = \frac{1}{h^2}\left(\psi_{2,j} - 2\psi_{1,j} + \psi_{0,j} + \gamma^2\left(\psi_{1,j+1} - 2\psi_{1,j} + \psi_{1,j-1}\right)\right) \tag{25}$$

The value of $\psi_{0,j}$ can be accounted for by using the no-slip boundary condition on the cavity wall, i.e.,

$$v = -\frac{\partial\psi}{\partial x} = 0, \text{ at } x = 0 \tag{26}$$

Using central differencing, this condition can be written as

$$-\left(\frac{\partial\psi}{\partial x}\right)_{1,j} = 0 \Rightarrow -\left(\frac{\psi_{2,j} - \psi_{0,j}}{2\,h}\right) = 0 \Rightarrow \psi_{0,j} = \psi_{2,j} \tag{27}$$

where again the order of accuracy is $O\left(h^2\right)$.

This finding, together with $\psi_{1,j} = \psi_{1,j+1} = \psi_{1,j-1} = 0$ gives

$$\omega_{1,j} = -\frac{2}{h^2}\left(\psi_{2,j}\right) \text{ for } j = 2, n_{y-1} \tag{28}$$

Similarly, for the other walls of the cavity the boundary conditions for vorticity can be deduced from

$$\omega(x_i, y_i) = -\left(\frac{\psi_{i+1,j} - 2\psi_{i,j} + \psi_{i-1,j}}{h^2} + \gamma^2\frac{\psi_{i,j+1} - 2\psi_{i,j} + \psi_{i,j-1}}{h^2}\right) \tag{29}$$

to give

$$
\begin{aligned}
\omega_{i,1} &= -\frac{2}{h^2}\gamma^2(\psi_{i,2}) \text{ for } i = 1,\ldots,\ n_x \\
\omega_{n_x,j} &= -\frac{2}{h^2}\gamma^2\left(\psi_{n_{x-1},j}\right) \text{ for } j = 2,\ldots,\ n_{y-1} \\
\omega_{i,n_y} &= -\frac{2}{h^2}\gamma^2\left(\psi_{i,n_{y-1}} + \tfrac{h}{\gamma}\right) \text{ for } i = 1,\ldots,\ n_x
\end{aligned}
\tag{30}
$$

It should be noted that the final boundary condition shown in Equation (30) is for the moving lid, and the extra term is due to the tangential velocity being non-zero.

3. Practical Elements of CFD Teaching

3.1. Introduction

For Computational Fluid Dynamics laboratories, each student had a desktop computer on which was loaded software which provides the interface as a 'three-dimensional' fully interactive environment. The interface uses the concept of 'virtual reality' and allows a student to simulate a flow, including geometry, properties, boundary conditions, closure of equations, numerics, etc, from beginning to end without having to resort to specialized codes, or code writing. The code used can be used for professional engineering consultancy, so giving students useful skills and experience which contribute to their preparation for the workplace. The virtual reality (VR) environment is designed as a general purpose CFD interface, which can handle such problems as fluid flow, heat transfer, reacting flows, or a combination of all, and consists of the VR-Editor (pre-processor), the VR-Viewer (post-processor) and the solver module which performs the actual calculations. The VR-Editor allows a student to set the overall computational domain size, define the position, size and properties of object introduced into the domain, specify the material properties which occupy the domain, specify inlet and outlet boundary conditions, specify initial conditions, select turbulence models (if necessary), specify the fineness and type of computational mesh, and set the numerics by choosing a suitable solution algorithm and which influence the speed of calculation to obtain a convergence solution. Once the student is satisfied with problem set-up in the pre-processor, calculations are made using the solver module and the progress of the calculations is clearly monitored on-screen until convergence is reached. The results of the flow simulation are then displayed on the VR-Viewer graphically or they can be analysed from the raw numerical data. The post-processing graphical capabilities are vector plots, contour plots, iso-surfaces, streamlines and x-y plots.

3.2. Interface Design Features

The Computational Fluid Dynamics laboratory is designed in a user-friendly way, the computational features are easy to access and implement. It is important to keep parameter settings at the top level, and not to use multi-levels, as much as possible, as the setting of vital parameters necessary to the success of a particular simulation may well get hidden or even forgotten about, thus frustrating the student with strange results. Also it is important to show students that CFD methodology needs to be systematic and rigorous, as the laboratory must reflect what is found in engineering practice.

The CFD interface was designed, as shown on Figure 3, so that the features systematically inform, it is vocationally sound and is easy to use.

Each simulation process reflected the situation in engineering practice with students systematically setting up the problem, solving and finally analysing the results. The students interact with the software using mouse and keyboard input with no requirement for the distraction of computer language coding so enabling them to concentrate on the CFD methodology and procedures and how close to reality their results are. An important feature of the CFD interface is that it is stand-alone, by which is meant that the problem-building, solving and post-processing are all combined in the same virtual reality environment. Also important is that all the data representing the results can be moved to the familiar Microsoft Office environment for further processing and reporting. The CFD software package operates on the Windows OS platform using PCs with relatively low computer power, and so the solvers have to be reasonably fast because the students have limited guided time in the laboratories, and also to keep their level of interest heightened, calculated results should come back reasonably quickly. In addition to producing contour plots, vector plots, etc., the post-processor could also produce animation for transient flows.

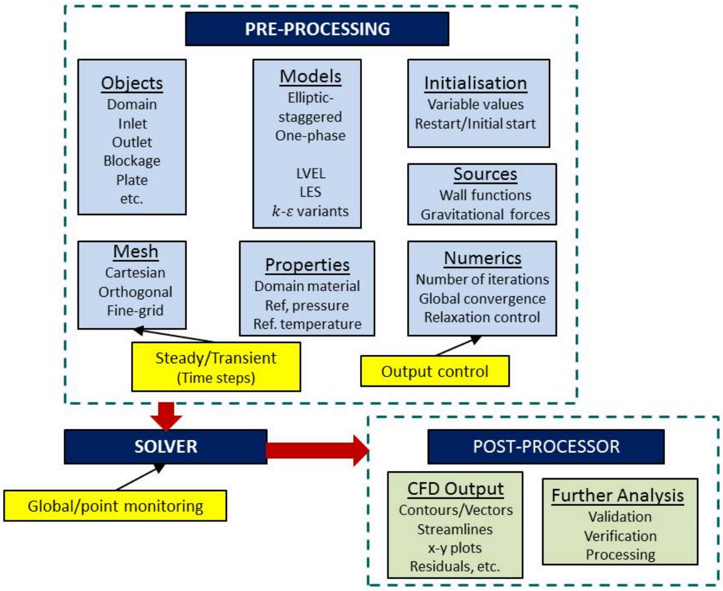

Figure 3. Summary of Computational Fluid Dynamics (CFD) interface. (Reproduced with permission from [24]).

3.3. Example to illustrate use of the Interface

An example of flow through a labyrinth with heat exchange is given to illustrate the simplicity of the interface. The geometry is 2D with dimensions and the content of the computation domain is summarized on Figure 4.

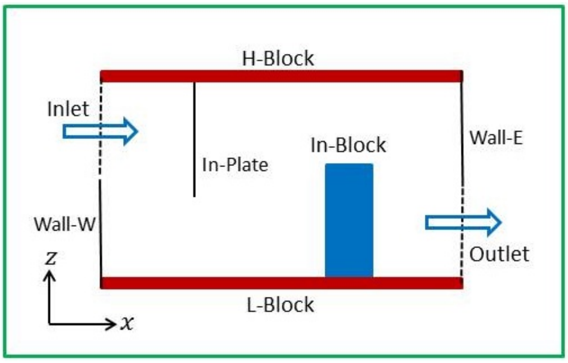

Figure 4. Geometry and contents of a 2D computational domain with steady turbulent flow of air and heat exchange.

Initially the pre-processing virtual reality GUI is opened and the case title immediately entered. With reference to Figure 3, under 'Objects' the x-domain size is changed to 2.0 m and under 'Mesh' a Cartesian grid is chosen. Under 'Properties', the user now checks that the default properties for this case are chosen, i.e., 'Air at 20 °C, 1 atm', the ambient air is set at 20 °C, and the ambient pressure is set at 0 Pa relative to 1 atm.

Under 'Models' the 'solution for velocities and pressure' is left ON, the 'energy equation' is clicked and 'Temperature' is selected. The turbulence model selection is left as the KE variant 'KECHEN'.

Under 'Objects' each of the objects within the domain are selected, sized and positioned and where necessary given the relevant attributes. The objects introduced in order are, the H-BLOCK whose attribute is 'ALUMINIUM at 27 °C. the L-BLOCK whose attribute is 'ALUMINIUM and which is a HEAT SOURCE of 100 W', the WALL-E which is a plate, the WALL-W which is a plate, the IN-PLATE, the IN-BLOCK whose attribute is 'COPPER', the INLET whose attribute is velocity in the x-direction at 0.5 m/s and whose temperature is 'AMBIENT', the OUTLET which has 'PRESSURE' as its attribute.

Once the basic geometry is in place, a computational grid is automatically suggested and can be inspected by the 'MESH' toggle, which for the present case, generates a picture as shown on Figure 5. The grid can be customized to the user's wishes simply by clicking on 'Grid Mesh Setting' and adjusting the default numbers.

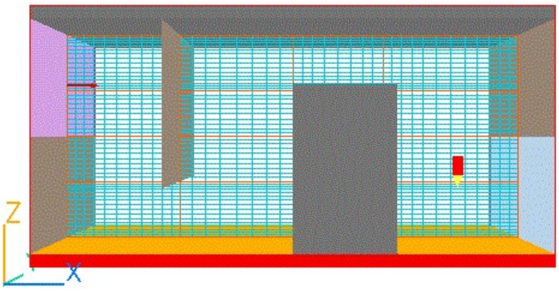

Figure 5. Grid automatically generated for labyrinth case.

In this example, 'Initialisation' and 'Sources' are taken as the defaults, and so only 'Numerics' is left to inspect and adjust. Here the number of iterations is set and the convergence tolerances are left at default. Importantly the probe used as one measure of convergence is appropriately positioned, i.e., not too close to the inlet or not placed in an anticipated region of recirculation.

The solver is now activated to perform the simulation. The progress of the calculations is displayed in relation to convergence, number of sweeps of the computation domain completed and the estimated time left to complete the simulation.

Once the solver has closed, the GUI post-processor can now be opened to view the results. Toggles are available to view 'Vectors', 'Contours', 'Streamlines', Iso-surfaces', and further toggles are available to select which calculated variable is of interest. The results can be viewed on any plane or at any angle by the use of direction and rotation buttons respectively. A typical result is shown on Figure 6 for a vector plot.

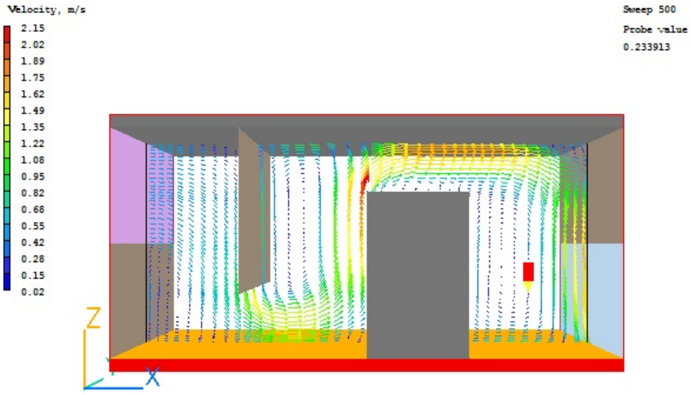

Figure 6. A typical vector plot.

The students are shown how to save their work, and how to transfer results suitable for reports and further analysis.

3.4. Implementation of the CFD Laboratories

The Computational Fluid Dynamics laboratories were introduced using a combination of short-lectures on back-ground theory, demonstrations, student group work and discussion, and assisted hands-on computer usage over an intensive four week period. Gradually, as time passed, the students became more confident and were able to navigate their way around the software efficiently and with assurance. The format of the demonstrations was that the instructor built simple simulations while students, watching a large overhead screen, did the same operations on their own computer. Group work was also used where perhaps a group of four students were given an exercise and asked to build a simulation. This proved a very effective and efficient way of learning. Gradually the method of delivering instruction changed over the four weeks, from demonstrations, to the use of detailed instruction sheets fully supported by faculty and teaching assistant intervention, to the distribution of descriptions of problems without instructions or hints and only the occasional help from support staff. There was never any restriction in seeking help from other students (again something which happens in engineering practice).

The main learning outcomes during this part of the course were to become familiar with the equations that govern steady and transient fluid flow and heat transport (conservation of mass, momentum, heat, species and energy) and be able to apply these equations to a range of practical problems in the areas of fluid flow, heat transfer and species transport. Seminars were given to inform students of what was expected during the CFD laboratory. It was stressed that the students should not compartmentalize theory, experiments and computational fluid dynamics but rather view them as mutually complimentary. The main bulk of these seminars was the introduction of the students to standard CFD methodology and procedures and the splitting of the CFD process into three distinct procedures, i.e., pre-processing, solving and post-processing.

During this four week section of the course, theoretical fundamental topics covered included:

- Getting started;
- CFD notation;
- CFD equations-continuity, momentum, energy, concentration of species;
- Finite differencing;
- The finite-volume method;
- Boundary conditions;
- Accounting for the pressure term;
- Closure of averaged equations;
- Time-stepping techniques; and,
- Properties of numerical methods.

This section of the course included discussions on CFD good practices and importantly the students were taught to be critical of their results and how to examine if what they got as results is what they might have expected. During the hands-on sessions, the following areas listed in Table 1 were emphasized.

Table 1. Areas for systematic consideration.

Geometry	**Solid and Other Fluid Boundaries, Special Shapes of Objects.**
Physics	Incompressible/compressible fluid, which quantities to be solved for, closure of equations, materials within the computational domain, initial and boundary conditions
Mesh	The choice here is Cartesian meshing or orthogonal meshing. The Cartesian mesh can be automatically generated or built by hand, and can be refined close to solid objects or in areas of high velocity gradient. Fine-grid embedding can also be used when appropriate.
Numerics	Convergence monitoring, selection of numerical scheme, maximum number of iterations, convergence criteria for each variable.
Post-processing	Flow visualization, analysis, validation using published experimental data or analytical calculation data.

4. Discussion

4.1. Overall Course Description

The main learning outcomes of this course are to understand how to apply the equations of transport governing conservation of mass, momentum and energy in an open and transparent way, and to be able to apply them correctly to a range of practical projects, including

- predicting drag forces on bluff, streamline bodies and flat plates;
- analyzing the flow in pipe systems;
- analyzing performance of radial flow pumps and turbines; and,
- matching pumps and turbines for particular applications.

The overall course last twelve weeks and was conducted during the fifth semester for students of mechanical and aerospace engineering, with 38 students taking the course. The first four weeks were used to teach the numerical aspects of computational fluid dynamics using close integration between theoretical lectures and hand-on computer laboratories, using as already described, mainly *Mathematica*, although some exercises using the versatile Python computer language were also introduced. The course during the second four weeks still had theoretical aspects within it, mainly in the form of turbulence modelling, but tended to concentrate on giving the students skills so as to solve real, though simple practical problems. The last four weeks of the course involved the students carrying out the series of projects outlined above, either individually or in small groups, which at first were supplemented with guidance and hints from the instructors on how to solve, but gradually this moved to problem description only, and it was left entirely to the student or students on how to enact a solution. In addition to set class time during these four weeks, the laboratory was open to students at any time, and discussion between the students was actively encouraged.

Assessment of students was by a series of bi-weekly quizzes and by assignment. No final examination was used.

4.2. Use of Mathematica

Based on the governing equations and boundary conditions established in Section 2, the students had to individually write coding using the symbolic software, *Mathematica* [9], to solve the problem of a lid-cavity flow [21]. The students had to write coding which set-up the required geometry, the mesh parameters and initial and boundary conditions. Next came coding for Eqns. [14] and [15], i.e., for $\mathcal{R}_{ij}^{k+1}$ and $\mathcal{L}_{ij}^{k+1}$ followed by coding for the iterative algorithm.

The mesh size for this problem and grid spacing had to be defined, and initially ω and ψ were set to zero for all nodes, except the top wall, where the vorticity is non-zero. The Reynolds number was set to 100, the relaxation parameter, p to 1 and the aspect ratio γ to 1. The students also had to set an appropriate residual value e and carefully nest the loops used to execute the iterative algorithm. At every iteration step k, if the maximum value of the absolute value of $\mathcal{L}[i,j]$ is less than e, then the calculations are halted. The iterative loops were wrapped with the Mathematica function Timing to give an estimate of the time taken to do the calculations.

Following completion of the calculations, students had to then write coding to display contour and vector plots of their results. An example of a contour plot obtained for the streamfunction is given on Figure 7.

Extraction of the centerline velocities was also instructive for the students in that first, students were given more experience of viewing velocity profiles, and second there are ample calculations and measurements of the lid-cavity flow for comparison with their results [25–27]. The centerline velocities u (in the x direction) and v (in the y direction) were derived from the streamfunction values using the equations

$$u_{i,j} = \frac{\partial \psi}{\partial y} = \frac{\psi_{i,j+1} - \psi_{i,j-1}}{2h} \quad j = 2, \ldots, n_y \tag{31}$$

$$v_{i,j} = -\frac{\partial \psi}{\partial x} = -\frac{\psi_{i+1,j} - \psi_{i-1,j}}{2h} \quad i = 2, \ldots, n_x \tag{32}$$

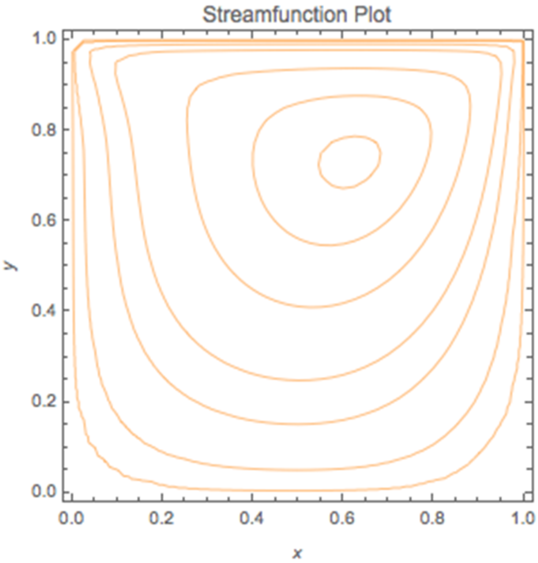

Figure 7. Contour plot of streamfunction for lid-cavity problem. (Reproduced with permission from [22]).

The velocity distributions for u and v in Figure 8 are drawn along the vertical and horizontal centerlines, respectively.

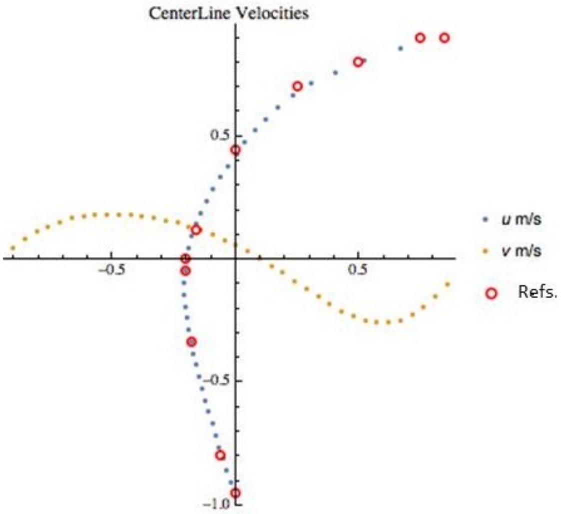

Figure 8. Calculated centerline velocity results compared with those of Refs. [25] and [27]. (Reproduced with permission from [21]).

Importantly, the students were made aware that the profiles they calculated must be compared, preferably with experimental measurements, to test the validity of their calculations. As can be seen from Figure 8, the calculations for the horizontal centerline velocities compare very favourably with those reported by [25,27] for Re = 100. Finally, the students were asked to produce coding which displayed the formation of the streamfunction contour plot with time.

4.3. *Computational Fluid Dynamics methodology*

In the second third of the course, the students moved on to the skills involved when using CFD in a professional consultancy capacity, while still learning necessary underpinning theory. The idea of fluid mechanics theory, experimental work and the current CFD skills and knowledge being acquired as complementary was reinforced. CFD now has well established methodology coupled with good practice and procedures, and it was emphasized that following these carefully would produce results much more efficiently and with a better degree of confidence. The students were taught how to use CFD judicially and appropriately, and to check their work against experimentation where possible. The idea of breaking the CFD process into three distinct tasks was again reinforced, i.e., using the pre-processor to build the problem, using the solver to obtain a converged solution and the use of the post-processor to obtain informative results. The numerical process of discretization was especially reinforced, as was how to activate the acquired conservation equations, geometrical aspects, fluid and solid properties, activation of sources, especially the gravity or body force, numerical details, which included initialization, the iterative process and how to use under-relaxation when necessary. Setting of the boundary conditions correctly was taught as was how to close the conservation equations by using an appropriate turbulence model [28,29]. When results were achieved it was extremely important that students could interpret what they had got in light of, say, geometric considerations, and also to be open-minded and critical.

During the demonstrations on how to use the GUI interface it was felt important that students could have 'hand-on' experience rather than just watching. Several simple three-dimensional flows were used as exemplars to give an over-arching view of the CFD process and the virtual reality environment which facilitates the VR-Editor (pre-processor), the VR-Viewer (post-processor) and the solver module which performs the calculations was introduced.

It was demonstrated how the VR-Editor is used to set the size of the computational domain, define the position, size and properties of objects introduced into the domain, specify the materials which occupy the domain, specify inlet(s) and outlet(s) and other boundary conditions, specify chosen initial conditions, select a turbulence model (when appropriate), specify the fineness of the computational grid and set parameters which define convergence and also influence the speed of convergence of the solution procedure. Viewing movement controls were also discussed which zoom, rotate, and allow for vertical and horizontal translations. It was emphasized how to systematically create a new environment for a particular simulation and finally set an appropriate number for the maximum number of iterations, set the position of the monitor for convergence. Finally, the students were encouraged to review the computational domain, especially in regard to position and attributes of the objects and the refinement of the grid close to solid surfaces, which at this stage of the demonstration would not be appropriate.

The convergence to a solution was monitored in two ways. First a point in the computation domain was chosen and values of the variables updated after each successive sweep. Once the values became constant it was assumed a converged solution was obtained. This was backed up by calculating the 'global errors' or changes between successive sweeps through the computation domain for each variable and when these errors were deemed sufficiently small, convergence was declared.

Once a converged solution was obtained, the simulation results were viewed by way of the post-processor facilities which displayed vector plots, contour plots, iso-surfaces, streamlines and x-y plots. The students were encouraged to find the optimum way of presenting their results and were also instructed on how to print the results. Very importantly, they were taught to always save and catalogue all input and output files.

4.4. Individual and Group Projects

For the last four weeks of the course the students were given a series of assignments which got progressively more difficult. The projects were wide ranging and covered important topics, such as the use of orthogonal grids, fine-grid embedding, cut-cell techniques, further exploration of turbulence models and how appropriate they are, the volume of fluid technique and two-phase flows (solid/gas and liquid/gas). Several projects covered how to import complicated geometries, built using a computer-aided drawing (CAD) package. In addition to problems involving purely fluid flow, projects were included which expanded to flow combined with heat transfer, the modelling of species transport and gaseous combustion. Use was made of the non-premixed combustion model where a natural gas combustion problem was set and solved using the non-premixed combustion model for the reaction chemistry. A project was also included which modelled an evaporating liquid spray where several discrete-phase models, including wall surface reactions were included.

During this part of the course the teaching model used emphasized active and collaborative learning in addition to students and instructors discovering and constructing knowledge and techniques together. Outside of assignments set for grading, the students were encouraged to collaborate and cooperate as much as they though appropriate.

4.5. Online Questionnaire

An anonymous online survey was conducted before students who had taken the course were given their final grades for the course to help with its formative evaluation. The fifteen questions/statements listed in Table 2 were used during the survey, with each student requested to mark as they thought appropriate, on a five-point Likert scale, whether they: strongly agree, agree, neutral, disagree, strongly disagree with the question/statement. There was also a column to mark if they had no opinion, and a space to make short comments for qualitative feedback.

Table 2. A list of questions/statements used in the survey for students' feedback.

No.	Question/Statement
1	I have used CFD modelling before.
2	I found the use of vorticity and streamfunction to be instructive.
3	I could relate vorticity and streamfunction to the primitive variables easily.
4	I found the use of collaboration with other students useful.
5	I preferred working on my own to working in groups.
6	Do you think the balance between CFD theory and practice was correct?
7	This CFD course enhances my understanding of fluid mechanics theory.
8	This CFD course is a useful addition to the fluid mechanics laboratories.
9	The 'hands-on' aspects of this CFD course has taught me extra skills.
10	Do you feel you can continue to model basic flows without much help?
11	Does this course give you confidence that you can model more difficult flows?
12	By using CFD I have learned things that could not be taught through traditional theory or experiments.
13	I have now a sound knowledge of CFD good practice.
14	I have enjoyed this course.
15	I would recommend this CFD course to others.

Generally, student feedback surveys have a very low response rate [30,31]. However the response rate here was relatively high (77%) and overall the results of the survey showed a positive attitude of the students towards the course. The numerical responses to the survey are shown on Figure 9 and indicate that students felt they had benefited from their exposure to this CFD course. In additional comments made by the students the view was expressed that the amount of material introduce was about right, although some felt that some of the exercises were unnecessarily long. Students did not seem to worry about the concept of streamfunction being used but some did express concern about their understanding of vorticity in a physical sense. The students were generally appreciative that they could easily visualize flow using contour and vector plots and generally agreed that the combination of basic theory combined with computer coding gave them more confidence and understanding. Student showed some enthusiasm about learning more about CFD and in particular how they could incorporate CFD into projects from other subjects and capstone projects.

It was noted that the students liked the hands-on and self-discover approach, although at times some frustration was also noted, and they were fairly neutral on working in groups as opposed to working on their own. In fact it was very noticeable that students took pride in performing the simulations by themselves with a real sense of achievement ensuing.

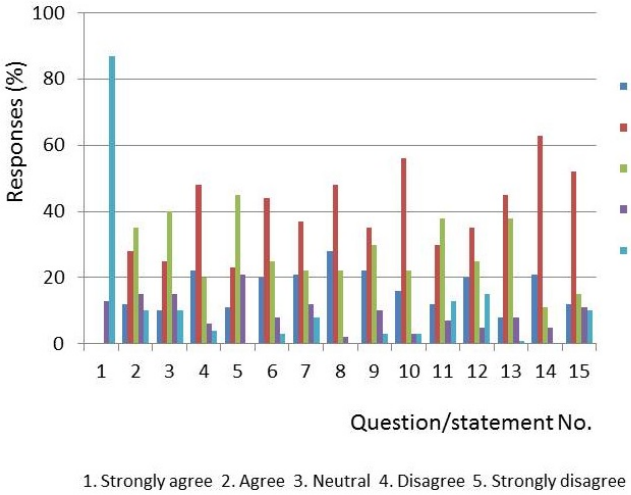

Figure 9. Chart showing survey results (N = 29).

Several points should also be made: the traditional view of CFD is that it has a steep learning gradient, but it has been shown here that with proper interfaces and a limited depth of flow difficulty that this gradient can be brought to an acceptable level. Also, it should be pointed out that for the practical software used no mention of code development was made to the students, as the learning outcome was to become proficient in the use of the existing coding. This would need to be remedied by a later course which involves ways of interacting with the existing, quite complicated core coding.

Evaluation of the course will continue in the future and hence be developed according to results. Only student views have been collected up to now but it is intended to take into account both student and instructor views in the future. It is felt more attention to how interest is aroused in general concerning the use of CFD is needed in future surveys as is the promotion of the use of CFD as a research tool during say capstone projects and post-graduate work. There will also be an attempt to elicit feedback from students who are at the 'beyond-completion' stage of the course.

5. Conclusions

In this paper the approach to delivering understanding of the numerical and practical aspects of the use of Navier-Stokes equations to solve fluid mechanics problems is outlined. For the numerical understanding, a well-defined sequence of steps needed to solve the Navier-Stokes equation cast in vorticity-streamfunction form was outlined. The sequence was integrated with the software package *Mathematica* at appropriate stages to take care of tedious computations, and hence allowed the students to concentrate on the overall details of the solution process. In addition, it is now important generally to integrate computer technology so as to complete and compliment lectures and theory. This has the advantages of helping with the computations, aiding presentation for reports and the ability to continue with further analysis as well as motivation for the students. Incorporating Mathematica also takes away the 'black-box' approach so often encountered by students with full CFD commercial codes, which give no real understanding of the numerical mathematics involved.

In addition to teaching the students some of the numerical aspects of CFD, this course also allowed for an introduction to the type of CFD carried out in professional offices, and taught additional topics like turbulence modelling, the idea of obtaining a grid independent solution and how critical it is to validate results obtained by the CFD approach. One of the main reasons for the inclusion of the second part of this course was to give the students practical skills and it was found that the simplified interface design does provide students with hands-on experience, gained through an interactive and user-friendly environment, and encourages student self-learning. It was noted from a survey that students liked getting hands-on experience and the self-discovery approach, although at times some frustration was noted when setting a problem up.

Author Contributions: Conceptual, D.A. and M.J.; methodology and software D.A.; validation and analysis, D.A. and M.J.; original draft preparation, D.A.; review and editing, M.J.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

In general, when solving for the nodal quantities, this is in effect solving a system of n linear equations, where n is the total number of nodes in the system. The system of equations can easily be described by

$$A\mathbf{x} = \mathbf{b} \tag{A1}$$

where A is a $n \times n$ matrix of coefficients, $\mathbf{x}$ is a vector of n unknown values, and $\mathbf{b}$ is a vector of n constants. Rather than inverting the matrix A directly, the Gauss-Seidel method can be used to solve the equations iteratively, which is especially desirable for large systems. In general the Gauss-Seidel method employs forward substitution and is given by

$$x_i^{k+1} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{k+1} - \sum_{j=i+1}^{n} a_{ij}x_j^{k}\right) \tag{A2}$$

for $i = 1, 2, \ldots, n$. The iterations are performed until the difference of the quantity value, often called the residual error (E), between iterations reaches a specified tolerance.

References

1. Assanis, D.N.; Heywood, J.B. Development and use of a computer simulation of turbo-compounded Diesel system for engine performance and component heat transfer studies. *SAE Trans.* **1986**, *2*, 451–476.
2. Devenport, W.J.; Schetz, J.A. Boundary layer codes for students in Java. In Proceedings of the ASME Fluids Engineering Division Summer Meeting, Washington, DC, USA, 21–25 June 1998.
3. Zheng, H.; Keith, J.M. JAVA-based heat transfer visualization tools. *Chem. Eng. Educ.* **2004**, *38*, 282–290.
4. Rozza, G.; Huynh, D.B.P.; Nguyen, N.C.; Patera, A.T. Real-time reliable simulation of heat transfer phenomena. In Proceedings of the ASME-American Society of Mechanical Engineers-Heat Transfer Summer Conference, San Franscisco, CA, USA, 19–23 July 2009.
5. Qian, X.; Tinker, R. Molecular dynamics simulations of chemical reactions for use in education. *J. Chem. Educ.* **2006**, *81*, 77–90.
6. Pieritz, R.A.; Mendes, R.; Da Silva, R.F.A.F.; Maliska, C.R. CFD: An educational software package for CFD analysis and design. *Comput. Appl. Eng. Educ.* **2004**, *12*, 20–30. [CrossRef]
7. Stern, F.; Xing, T.; Yarbrough, D.B.; Rothmayer, A.; Rajagopalan, G.; Otta, S.P.; Caughey, D.; Bhaskaran, R.; Smith, S.; Hutchings, B.; et al. Hands-on CFD educational interface for engineering courses and laboratories. *J. Eng. Educ.* **2006**, *95*, 63–83. [CrossRef]
8. Curtis, J.C. Integration of CFD into undergraduate chemical engineering curriculum. In Proceedings of the Annual AIChE Meeting, Philadelphia, PA, USA, 16–21 November 2008.
9. Wolfram Research, Inc. *Mathematica, Version 12.0*; Wolfram Research, Inc.: Champaign, IL, USA, 2019.
10. Jaeger, M.; Adair, D. Human factors simulation for construction management. *Eur. J. Eng. Educ.* **2010**, *35*, 299–310. [CrossRef]
11. Jaeger, M.; Adair, D. Communication simulation in construction management simulation: Evaluating the learning effectiveness. *Australas. J. Eng. Educ.* **2012**, *18*, 1–14. [CrossRef]
12. Patil, A.; Mann, L.; Howard, P.; Martin, F. Assessment of hands-on activities to enhance students' learning for the first year engineering skills course. In Proceedings of the 20th Australasian Association for Engineering Education Conference, Adelaide, Australia, 6–9 December 2009; pp. 286–292.
13. Edgar, T.F. Enhancing the undergraduate computing experience. *Chem. Eng. Educ.* **2006**, *40*, 231–238.
14. Adair, D.; Jaeger, M. Incorporating computational fluid dynamics code development into an undergraduate engineering course. *Int. J. Mech. Eng. Educ.* **2015**, *43*, 153–167. [CrossRef]

15. Parulekar, S.J. Numerical problem solving using Mathcad in undergraduate reaction engineering. *Chem. Educ. Eng.* **2006**, *40*, 14–23.
16. Rockstraw, D.A. Aspen Plus in the curriculum-suitable course content and teaching methodology. *Chem. Eng. Educ.* **2005**, *39*, 68–75.
17. Finlayson, B.A.; Rosendall, B.M. Reactor/transport models for design: How to teach students and practitioners to use the computer wisely. In *AIChE Symposium Series*; American Institute of Chemical Engineers: New York, NY, USA, 2000; pp. 176–191.
18. Pomeranz, S. Using a computer algebra system to teach the finite element method. *Int. J. Eng. Educ.* **2000**, *16*, 362–368.
19. LeVeque, R.J. *Finite Difference Methods for Ordinary and Partical Differential Equations: Steady-State and Time-Dependent Problems*; Siam: Philadelphia, PA, USA, 2007.
20. Ferziger, J.H.; Peric, M. *Computational Methods for Fluid Dynamics*; Springer: New York, NY, USA, 1996.
21. Bozeman, J.D.; Dalton, C. Numerical study of viscous flow in a cavity. *J. Comput. Phys.* **1973**, *12*, 348–363. [CrossRef]
22. Adair, D.; Jaeger, M. Developing an understanding of the steps involved in solving Navier-Stokes equations. *Math. J.* **2015**, *17*, 1–19. [CrossRef]
23. Burggraf, O.R. Analytical and numerical studies of the structure of steady separated flows. *J. Fluid Mech.* **1966**, *24*, 113–151. [CrossRef]
24. Adair, D. Incorporation of Computational Fluid Dynamics into a Fluid Mechanics Curriculum. In *Advances in Modeling of Fluid Dynamics*; InTech: Berlin, Germany, 2012; ISBN 980-953-307-311.
25. Gia, U.; Ghia, K.N.; Shin, C.T. High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *J. Comput. Phys.* **1982**, *48*, 387–411. [CrossRef]
26. Spotz, W.F. Accuracy and performance of numerical wall boundary conditions for steady, 2D, incompressible streamfunction vorticity. *Int. J. Numer. Methods Fluids* **1998**, *28*, 737–757. [CrossRef]
27. Wan, D.C.; Zhou, Y.C.; Wei, G.W. Numerical solution of incompressible flows by discrete singular convolution. *Int. J. Numer. Methods Fluids* **2002**, *38*, 789–810. [CrossRef]
28. Launder, B.E.; Spalding, B. *Mathematical Models of Turbulence*; Academic Press: London, UK, 1972.
29. Agonafer, D.; Liao, D.G.; Spalding, B. *The LVEL Turbulence Model for Conjugate Heat Transfer at Low Reynolds Number, Concentration*; Heat and Momentum Ltd.: London, UK, 2008.
30. Gamliel, E.; Davidovitz, L. Online versus traditional teaching evaluation: Mode can mater. *Assess. Eval. High. Educ.* **2005**, *30*, 581–592. [CrossRef]
31. Nulty, D.D. The adequacy of response rates to outline and paper surveys: What can be done? *Assess. Eval. High. Educ.* **2008**, *33*, 301–314. [CrossRef]