

CFD STUDIO: AN EDUCATIONAL SOFTWARE FOR CFD ANALYSIS

R. A. Pieritz,
R. Mendes,
R. Ferraz and
C. R. Maliska

Federal University of Santa Catarina
 Mechanical Engineering Department
 SINMEC – Computational Fluid
 Dynamics Laboratory
 88040-900 - Florianópolis - SC – Brazil
 rap@sinmec.ufsc.br

ABSTRACT

The main goal of this paper is to demonstrate the general characteristics of the educational user-friendly CFD Studio package for CFD teaching. The package was designed for teaching 2D fluid mechanics and heat transfer process, including conduction, coupled conduction/convection, natural and forced convection, external and internal flows, among other phenomena. The finite volume methodology and its related topics can also be taught using the software. Therefore, general aspects of the three main modules, pre-processor, solver and post-processor are discussed aiming to show the generality of the tool. These modules are integrated in the application by a so-called “numerical problem project” which guide the student through the steps to obtain the solution. To approximate the partial differential equations the finite volume approach is employed using a fully-implicit formulation with the interpolation schemes CDS, UDS and WUDS. Mesh editing and non-orthogonal boundary-fitted mesh generation, using algebraic interpolation and elliptic equations, are important features of the package. Coupled heat transfer problems are handled using the “solid-block” formulation and the pressure-velocity coupling uses the SIMPLE and SIMPLER methods with non-staggered grids. To demonstrate the capabilities two fluid flow and heat transfer “problem projects” are presented.

INTRODUCTION

Computational Fluid Dynamics CFD is nowadays a powerful tool to solve scientific and engineering problems involving fluid flow and heat transfer. The frontiers of CFD use are already beyond the scientific applications, with more and more companies introducing the methodology in their designing procedures. The success of its utilization depends however, on the user's skills, both in the numerical technique and in the knowledge of the physical problem under consideration.

The computational implementation of a numerical solution consists in developing specific objects for grid generation, boundary conditions application, coupling of the discrete equations, solution of the linear systems, generation and control of the iterative procedures and post-processing, like scalar visualization, 3D graphics, iso-surfaces, velocity vectors etc.

In this paper it is introduced the educational freeware CFD Studio software. It was developed for educational purposes to allow undergraduate and graduate students to access finite volume numerical tools to be used in CFD and heat transfer courses like Maliska (1997) and Ribando (1997). It solves two-dimensional conduction-convection problems. The data manipulation in CFD Studio is entirely done by graphical interface, resulting in a very friendly interface between user and software, which helps in: i) reducing the time in the development cycle of a numerical solution, ii) avoiding the needed of high expertise in computer sciences, iii) reducing time and costs in research and development activities.

The numerical main features are: finite-volume based approach for discretizing of the differential equations; totally implicit formulation; CDS, UDS, WUDS interpolation schemes; mesh editing and mesh generation

(algebraic interpolation and solution of elliptical equations); non-orthogonal boundary-fitted framework; solid-fluid interaction using “Solid-Block” formulation; pressure-velocity coupling using SIMPLE and SIMPLER methods in non-staggered grids.

The CFD Studio was created using object oriented concepts in C++ for its CFD engine as described in Pieritz et al (2000), and for interfacing by the multiplatform library COI-lib 2.0-Classes and Objects (ESSS, 1998). This allows the use of the software in several different platforms: i) Unix/X-Window: IBM AIX; FreeBSD; Linux; SGI Irix (Silicon Graphics); SunSolaris; SunOS; HP-UX; DEC OSF; and ii) Microsoft Windows: 95/98/Me/NT4.0/2000/XP.

This paper is dedicated for demonstrating the concepts of the numerical methodology applied and the general features of the package. The software interface is described, introducing the main structure: work-pad, numerical set-up interface, pre-processor, processor and post-processor modules. An example is used to show the numerical capabilities and the quality of the results obtained comparing with available benchmark solutions. It is believed that this package will facilitate the introduction of CFD techniques in undergraduate courses and to greatly help students in the development of their research work. It can be used also as auxiliary teaching tool in heat transfer courses in undergraduate and graduate levels. In addition it is also a powerful tool that can be used by engineers in solving practical fluid flow and heat transfer problems.

MATHEMATICAL MODEL AND NUMERICAL METHODOLOGY

In the Cartesian coordinate (x, y) system, the differential equations that govern steady elliptic convection-diffusion problems used in the CFD Studio software are (Bejan 1984), the mass conservation, momentum and energy equations, given by

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u_j)}{\partial x_j} = 0 \tag{1}$$

$$\frac{\partial (\rho u_i)}{\partial t} + \frac{\partial (\rho u_j u_i)}{\partial x_j} = -\frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j} \left(\mu \frac{\partial u_i}{\partial x_j} \right) + S^{ui} \tag{2}$$

$$\frac{\partial (\rho T)}{\partial t} + \frac{\partial (\rho u_j T)}{\partial x_j} = \frac{\partial}{\partial x_j} \left(\frac{k}{c_p} \frac{\partial T}{\partial x_j} \right) + S^T \tag{3}$$

where *u* is the velocity component, ρ is the density, *k* is the thermal conductivity, *P* is the pressure, *c_p* is the specific heat at constant pressure, μ is the absolute viscosity, *T* is the temperature and *S* an appropriate source term.

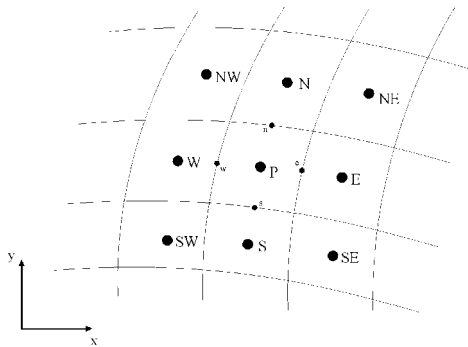


Figure 1. The numerical stencil used in CFD Studio.

The numerical method applied is the finite-volume method with boundary-fitted coordinates, as described in Maliska (1995). Figure (1) shows the elemental control P volume and their neighboring volumes in a structured grid framework, as used by the CFD Studio software.

Equations 1-3 can be written in a general form and write in a general curvilinear coordinate system, as shown in Eq. (4), allowing the numerical procedure to be done in the transformed plane (Maliska, 1995, Thompson et al, 1985).

$$\frac{\partial}{\partial t} \left(\rho \frac{\phi}{J} \right) + \frac{\partial}{\partial \xi} (\rho U \phi) + \frac{\partial}{\partial \eta} (\rho V \phi) = \frac{\partial}{\partial \xi} \left(\Gamma^\phi J \alpha \frac{\partial \phi}{\partial \xi} - \Gamma^\phi J \beta \frac{\partial \phi}{\partial \eta} \right) +$$

$$\frac{\partial}{\partial \eta} \left(\Gamma^\phi J \gamma \frac{\partial \phi}{\partial \eta} - \Gamma^\phi J \beta \frac{\partial \phi}{\partial \xi} \right) + P^\phi \tag{4}$$

where

$$U = u y_\eta - v x_\eta \tag{5}$$

$$V = v x_\xi - u y_\xi \tag{6}$$

are the contravariant velocities components of the velocity vector and

$$\alpha = (x_\eta^2 + y_\eta^2) \tag{7}$$

$$\gamma = (x_\xi^2 + y_\xi^2) \tag{8}$$

$$\beta = (x_\xi x_\eta + y_\xi y_\eta) \tag{9}$$

$$J = (x_\xi y_\eta - x_\eta y_\xi)^{-1} \tag{10}$$

Figure 2. Examples of CFD Studio interface windows: work-pad console, “numerical problem project”, numerical engine monitor, viewer of simulation residues convergence, solver

are the metrics and the Jacobian of the transformation, respectively. Table (1) shows the Γ^ϕ e P^ϕ parameters for the different conservation equations being solved.

To obtain the approximated equations, balances of the conserved properties are made or the divergence form of the conservation equations are integrated over the control volumes. The time discretization was done using a fully implicit formulation and the CDS, UDS and WUDS interpolation functions are used. Boundary conditions are applied using the fictitious volumes concept. A non-staggered grid arrangement was employed with the SIMPLE and SIMPLEC methods for

treating the pressure-velocity coupling, as described on Van Doormaal and Raithby (1984). The linear system (with a 9 diagonals matrix) can be solved by TDMA, Gauss-Seidel or Jacobi (for teaching purposes) and LU decomposition. The interested user can find full details of the coordinate transformation and of the finite-volume method used in CFD Studio in Maliska (1995).

Table 1. Γ^ϕ and S^ϕ for the different conservation equations

Conservation equation	ϕ	Γ^ϕ	P^ϕ
Global mass	1	0	0
Momentum in x axis	u	μ	$\frac{\partial P_\phi}{\partial \xi} y_\eta - \frac{\partial P_\phi}{\partial \eta} y_\xi$
Momentum in y axis	v	μ	$\frac{\partial P_\phi}{\partial \eta} x_\xi - \frac{\partial P_\phi}{\partial \xi} x_\eta - \frac{\mu v}{J} + \frac{\rho}{J} g \beta (T - T_\infty)$
Energy	T	k / cp	0

The data organization and the user interface were based on the basic steps that describe the usual approach in the solution of a CFD using finite-volume methods using the Sinflow Library code, as described in Pieritz et al (2000). These steps are: 1) the pre-process (definition of the geometry, medium and boundary conditions; 2) setting of the numerical parameters); 3) the process (the solution of the problem) and 4) the post-processing (data in/out and the visualization of the results).

The CFD Studio application is divided in modules, each one with its window interface. These modules communicate dynamically by “message post and sending” architecture, increasing performance and reliability. All software are implemented in Oriented Object Programming (OOP), written in C++ (ANSI 1998, Barton 1994). The input/output data are implemented in XML (Extensible Markup Language) - Hunter et al (2001), to be platform and binary code independent, standard, and allowing exchange data between different applications

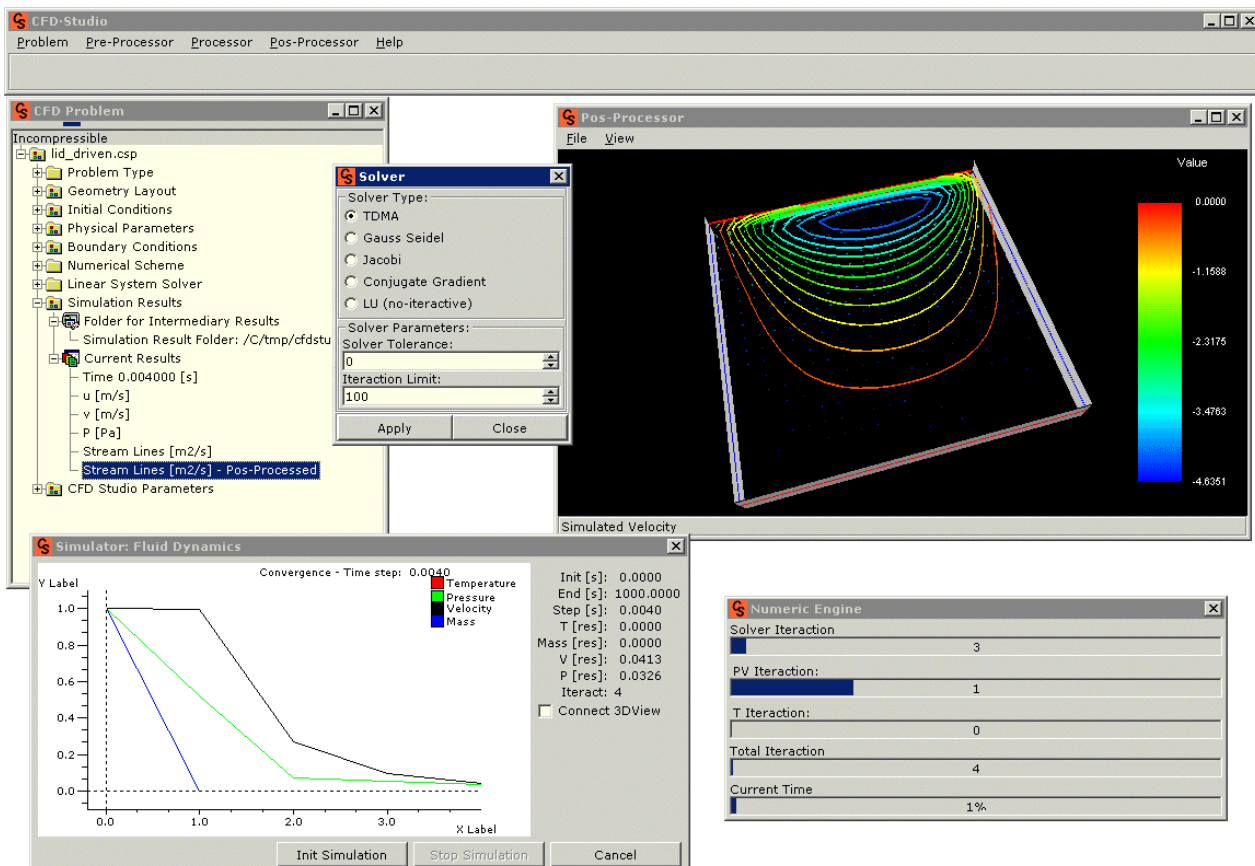


Figure 2. Examples of CFD Studio interface windows: work-pad console, “numerical problem project”, numerical engine monitor, viewer of simulation residues convergence, solver setting and 3D post-processing window.

THE SOFTWARE CFD STUDIO INTERFACE CONCEPT

The CFD Studio software was designed taking advantage of the modern computational resources to provide an exciting and intelligent learning environment - Fig. (2). The software is based in a dynamic internal data exchange and on its powerful graphical user interface.

MAIN CONTROL INTERFACE

CFD Studio main control center is called “work-pad”, Fig. (3), where the user has available the menus: problem (ordinary file in/out and setting parameters), pre-processor, process, post-processor and help modules access.

The main educational concept used in the interface is the “Numerical Problem Project” interface, Fig. (3), right. This interface shows all CFD parameters used in the current problem, allowing to the students quick access to all modules by mouse selection (all data and settings) step by step. For educational purposes, CFD Studio offers two options for setting-up a new problem: i) “CFD Problem Wizard” - the software guides interactively the user for setting-up the problem in 8 different steps; and ii) “CFD Problem” generates a default problem (the basic parameters are already pre-configured). Later the user can modify the problem parameters at his choice through the modules windows.

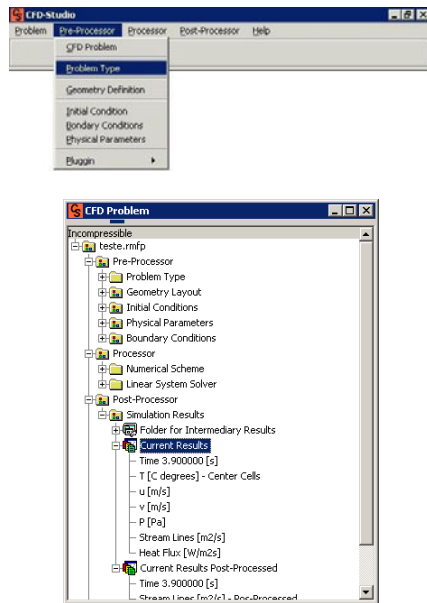


Figure 3. CFD Studio work-pad console (up) and the “Numerical Problem Project” interface (down).

PRE-PROCESSOR

The pre-processor main modules are: i) CFD problem definition like, “heat conduction”, “fluid flow” or “fluid flow and heat conduction”; ii) 2D mesh generator by solving elliptic equation or/and editor (fully interactive through the mouse), Fig. (4); iii) initial conditions setting – temperature, velocity, pressure; iv) boundary conditions – convection, heat flux, temperature, velocity, inlet, outlet, Marcondes et al (1999); v) physical parameters setting, Fig. (4), right, like thermal conductivity, heat generation, viscosity etc. Special cell called “solid blocks” can be defined to simulate the solid-fluid interaction; and vi) a special “plugging” to quickly start a CFD problem from a scanned gray level image or draft (“gray level image to mesh”).

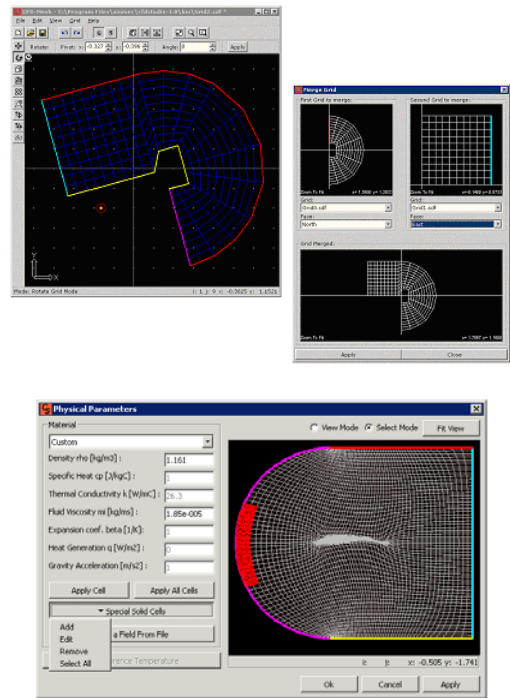


Figure 4. Pre-processor modules: mesh editor interface (up) and the physical parameters window (down).

PROCESSOR

The processor groups the modules for setting and controlling the numerical solution. The main modules are: i) numerical scheme interface; ii) 3D OpenGL hardware control center; iii) solver module setting: TDMA, Jacobi, Gauss Seidel, LU decomposition and Conjugate Gradients methods, Press (1992); iv) simulation residues convergence interface, as shown in Fig. (5), for mass, temperature, pressure and velocity; and v) the numerical engine status interface.

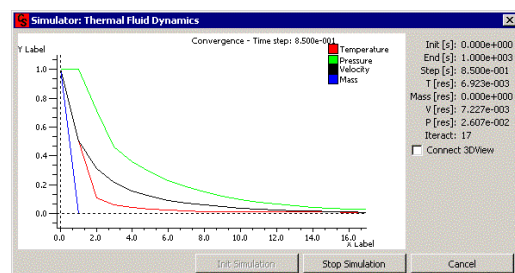


Figure 5. Processor module: simulation residues convergence interface.

POST-PROCESSOR

The 3D OpenGL hardware acceleration, Fig. (6), is used in the post-processor modules, allowing full visual interactive capability during the simulation steps. The main post-processor features are: i) iso-lines, iso-surfaces, scalars and vector field visualization in 3D; ii) interface customization (colors, preferences etc.); iii) data output in a standard ASCII format (data sheet); iv) capture screenshots and export in the TIFF and BMP image files; and v) on screen data sheets to control the simulation results.

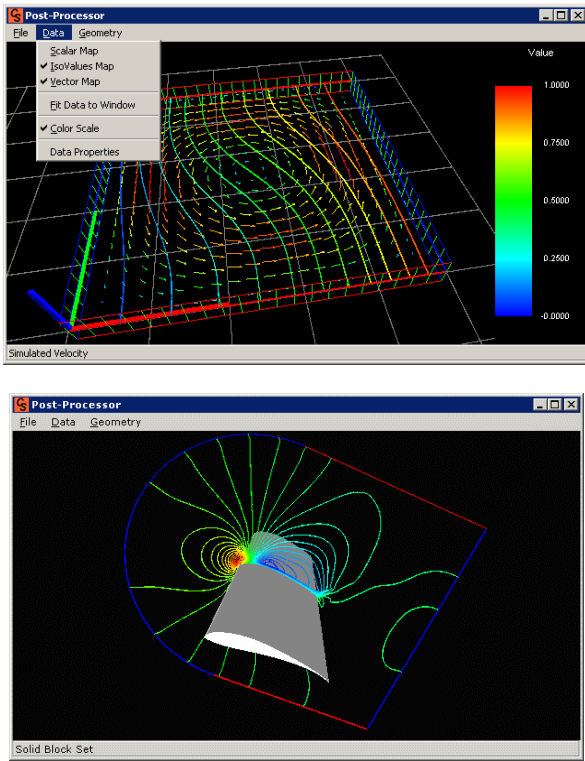


Figure 6. 3D post-process window interface – natural convection problem (up), and flow over a wing (pressure field -NACA profile) (down).

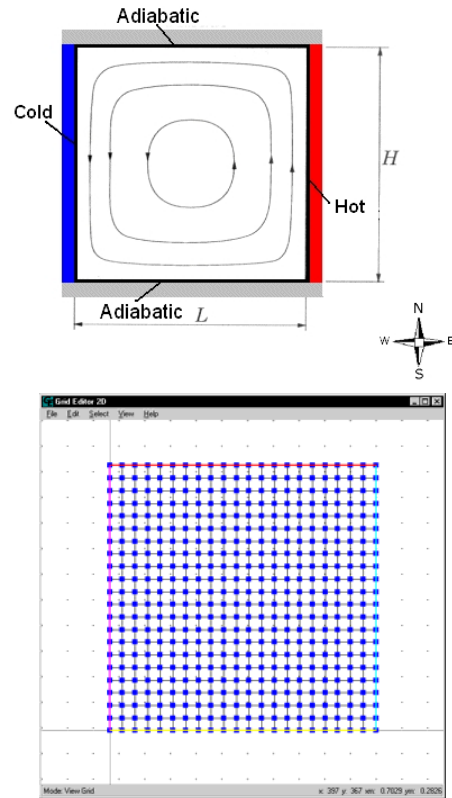


Figure 7. Natural convection in a square cavity (up) and the Cartesian grid (down)

**AN EXAMPLE:
BUOYANCY-DRIVEN CAVITY FLOW**

The objective of this example is to demonstrate the use of the CFD Studio in solving a fluid flow problem and its use by engineering students in their work assignments. The natural convection in a square cavity is frequently used for testing numerical algorithms, Vahl Davis (1983), Maliska(1995).

The solution was obtained for Prandtl equal to 0.71 and Ra equal to 103 and 104. To compare the results with the benchmark solution, the following dimensionless variables are used:

$$u^* = \frac{u \cdot L}{\alpha} \tag{11}$$

$$v^* = \frac{v \cdot L}{\alpha} \tag{12}$$

$$x^* = \frac{x}{L} \tag{13}$$

$$y^* = \frac{y}{L} \tag{14}$$

$$T^* = \frac{T - T_C}{T_H - T_C} \tag{15}$$

**EDUCATIONAL ASSIGNMENT:
SOFTWARE SETTINGS AND
NUMERICAL PARAMETERS**

The student starts setting step by step the geometric and numerical problem parameters through the “Numerical Problem Project” interface and modules, Fig. (3) and (4) – section 3. In this case, the numerical parameters used: $H = L = 1$; fluid mean temperature $T_0 = 0.5^\circ\text{C}$, $u_0 = v_0 = 0$; top and bottom insulated and ΔT (west = 0°C and east = 1°C); problem type “fluid flow and heat conduction”; Cartesian mesh 21×21 cells; CDS; SIMPLEC; simulation step time = 0.05s; maximum iteration PV (pressure-velocity coupling) = 20; maximum iteration in time = 1000; maximum error in the solver and for all variables = 10^{-5} ; error evaluation = center cells line in x and y; TDMA solver; solver maximum iteration = 200.

After that, the student starts the simulation by the “processor” main menu options (workpad – Fig (3)) and controls the simulation residues convergence by its interface – Fig. (5).

RESULTS

As stated, the goal of this section is to show how to use the CFD Studio without major preoccupation in obtaining a very fine mesh solution. In spite of that, it is important to show that the software solves correctly the conservation equations. Table (2) shows the results for the 21x21 grid compared with the results of the benchmark presented by Vahl Davis (1983).

Table 2. CFD Studio results compared with those presented by Vahl Davis (1983).

Ra	variable	Vahl Davis	CFD Studio
10 ³	u* _{Max}	3.649	3.553
	y*	0.813	0.833
	v* _{Max}	3.697	3.791
	x*	0.178	0.167
	Average Nu	1.118	1.117
	Nu _{max}	1.505	1.495
	y*	0.092	0.119
	Nu _{min}	0.692	0.702
	y*	1	0.976

Ra	Variable	Vahl Davis	CFD Studio
10 ⁴	u* _{Max}	16.178	16.135
	y*	0.823	0.817
	v* _{Max}	19.617	19.744
	x*	0.119	0.110
	Average Nu	2.243	2.236
	Nu _{max}	3.528	3.506
	y*	0.143	0.134
	Nu _{min}	0.586	0.613
	y*	1	0.988

SPECIAL CELLS: SOLID BLOCKS

The problem is now modified by the addition of solid blocks, obtaining two equivalent square cavities (with same dimensions and boundary conditions), as shown in Fig. (10). This modification allows to test the special cells ("solid blocks") used when the interaction between solid and fluid is of interest. The "solid blocks" method inside of the grid simulation allows modeling complex geometries, avoiding the use of more sophisticated methods like "multi-block", Wood (1993), Ghia (1992). The solid blocks temperature is prescribed as constant during the simulation.

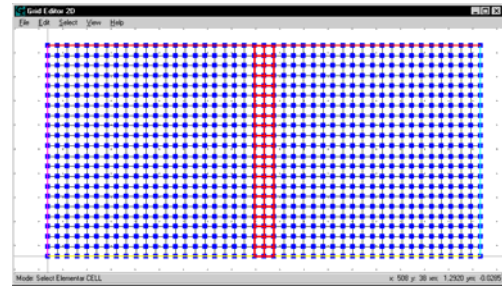
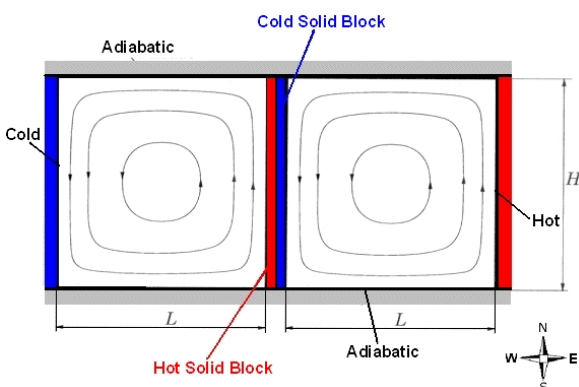


Figure 8. Square Cavity divided by solid blocks (up) and the simulation grid with solid blocks cells in red (down).

Figure (9) shows the temperature profiles T* in the half-height horizontal line for both solutions, single and double cavity. It can be seen that the results are the same, as expected.

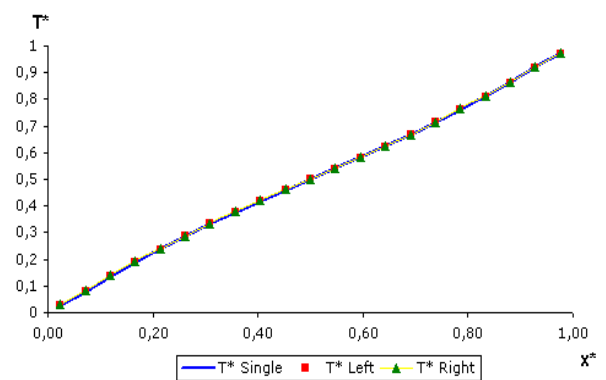
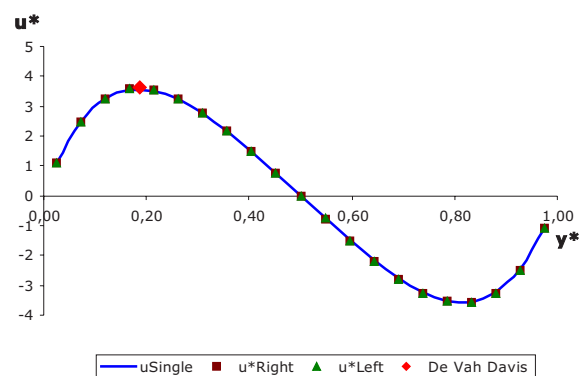


Figure 9. Temperatures profile T* in the half-height horizontal line for the different models.

Figure (10) presents the velocity profiles for the two cavities compared with the results obtained for the single cavity and the benchmark. In Tab. (3) the main results are also compared with the benchmark, showing a good agreement.



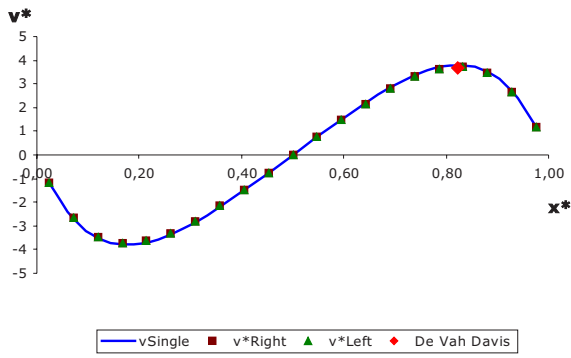


Figure 10. The u^* profile in the half-length vertical line for the different models (left) and v^* profile in the half-height horizontal line for the different models (right). In red are the points of maximum velocity reported by Vahl Davis (1983).

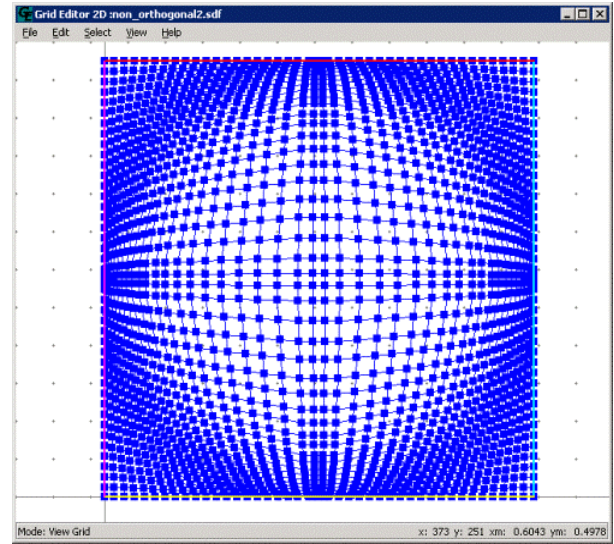


Figure 11. Non-orthogonal grid (43x43) used in the square cavity problem.

Table 3. The results obtained with CFD Studio compared with the Vahl Davis (1983) benchmark.

Ra	variable	Vahl Davis	left	right
10^3	u^*_{max}	3.649	3.578	3.578
	y^*	0.813	0.833	0.833
	v^*_{max}	3.697	3.733	3.733
	x^*	0.178	0.166	0.166
	Nu (average)	1.118	1.113	1.113
	Nu_{max}	1.505	1.500	1.500
	y^*	0.092	0.071	0.071
	Nu_{min}	0.692	0.692	0.692
	y^*	1	0.976	0.976

NON-ORTHOGONAL GRID

The natural convection example is now solved using a non-orthogonal grid to demonstrate the generality and flexibility of the CFD Studio package. Figure (11) shows the grid employed, Choi (1993)(1994a)(1994b). Several important calculations, like metric calculation, non-orthogonal terms discretization etc., can be tested with this example. Figure (12) shows the results for the benchmark, and the orthogonal and non-orthogonal grids of this paper. The agreement is good.

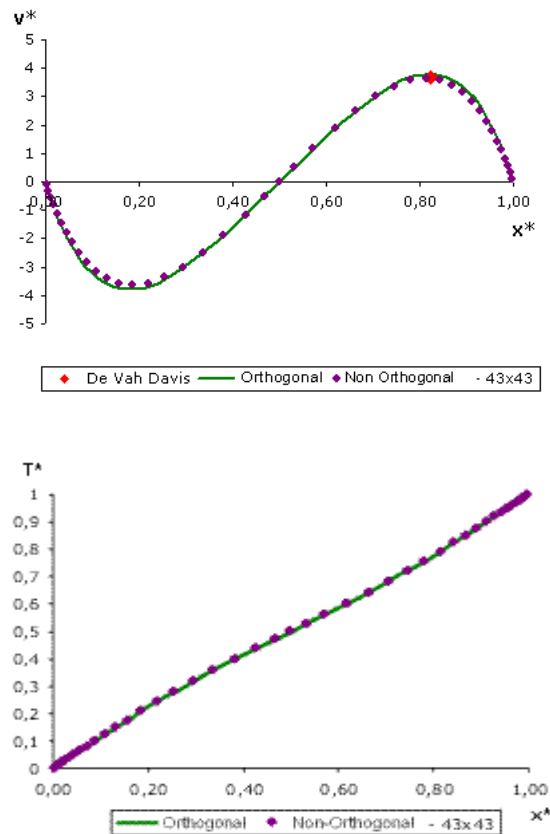


Figure 12. Velocity profile obtained with CFD Studio for $Ra 10^3$ (up) and the temperatures profile in half-height horizontal line for the different models (down).

Table 4. The results obtained with CFD Studio related to the benchmark - Vahl Davis (1983).

Ra	Variable	Vahl Davis	No-Orthogonal grid
10 ³	u _{max} *	3.649	3.326
	y*	0.813	0.814
	v _{max} *	3.697	3.646
	x*	0.178	0.186
	Nu (average)	1.118	1.115
	Nu _{max}	1.505	1.484
	y*	0.092	0.105
	Nu _{min}	0.692	0.707
	y*	1	0.965

FINAL REMARKS

This paper presented the educational freeware software CFD Studio to simulate and study complex thermal and fluid flow problems. The software architecture of this tool allows full visual interactive capability during the simulation steps. The software is highly versatile, with powerful graphical user interface and post-processor facilities. The software was evaluated by a numerical standard benchmark showing an excellent agreement. Modifications in the standard natural convection problem were also made to illustrate the possibilities of modeling complex geometries by the solid cells approach. The implementation of these CFD algorithms into teaching modules can provide a significant enhancement in the learning of heat transfer and fluid dynamics in undergraduate and graduate courses. The software can be used at home by students working in their assignments, where the student can develop his own numerical problems. Due to its generality, the software can also be used by engineers in the solution of many engineering problems.

ACKNOWLEDGEMENT

The authors are grateful to Axel Dihlmann, Márcio Monteiro, Marcelo Souza, Gerson Bridi, Ewaldo Schubert and Carlos Donatti for their contribution to this work. The software and the main related documents can be downloaded for free from the worldwide web site: <http://www.sinmec.ufsc.br/cfd>.

REFERENCES

- ANSI – American National Standards Institute, 1998, “Information Technology - Programming Languages - C++”, Document Number: ISO/IEC 14882.
- Barton, J.J., Nackman, L.R., 1994, “Scientific and Engineering C++, an Introduction with Advanced Techniques”, Addison-Wesley.
- Bejan, A., 1995, “Convection Heat Transfer”, John Wiley & Sons.
- Choi, S.K., Nam, H.Y., Cho, M., 1993, “Use of the Momentum Interpolation Method for Numerical Solution of Incompressible Flows in Complex Geometries: Choosing Cell Face Velocities”, Numerical Heat Transfer, Part B, Vol. 23, pp. 21-41.
- Choi, S.K., Nam, H.Y., Cho, M., 1994, “Use of Staggered and Nonstaggered grid arrangements for

Incompressible Flow Calculations on Nonorthogonal Grids”, Numerical Heat Transfer, Part A., Vol. 25, pp. 193-204.

Choi, S.K., Nam, H.Y., Cho, M., 1994, “Systematic Comparison of Finite-Volume Calculation Methods with Staggered and Nonstaggered Grid Arrangements”. Numerical Heat Transfer, Part B., Vol. 25, pp. 205-221.

COI-lib 2.0, 1998, “Reference Manual: Classes and Objects for Interfacing”, ESSS-Engineering Simulation and Scientific Software Press.

Ferziger, J.H., Peric, M., 1999, “Computational Methods for Fluid Mechanics”, Springer.

Ghia, U., Ghia, K. N., Shin, C.T., 1982, “High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and a Multigrid Method”, Journal of Computational Physics, Vol. 48, pp. 387-411.

Hunter, D., Rafter, J., Pinnock, J., Dix, C., Cagle, K., Kovack, R., 2001, “Beginning XML”, 2nd Edition, Wrox.

Maliska, C.R., 1995, “Transferência de Calor e Mecânica dos Fluidos Computacional”, LTC – Livros Técnicos e Científicos.

Maliska, C. R., Reis, M. V. F., Maliska Jr., C. R., Dihlmann, A., 1997, “Heat Transfer 1.0 – An Educational Software for Heat Conduction Teaching”, ASME Proceedings of the 32nd National Heat Transfer Conference, Baltimore, Vol. 6, pp 53-59.

Marcondes, F., Maliska, C.R., 1999, “Treatment of the Inlet Boundary Conditions in Natural-Convection Flows in Open-ended Channels”, Numerical Heat Transfer, Part B, Vol. 35.

Ribando, R. J., O’Leary, G. W., Carlson-Skalak, S., 1997, “General Numerical Scheme for Heat Exchanger Thermal Analysis and Design”, Computer Applications in Engineering Education, pp 231-242.

Patankar, S. V., 1980, “Numerical Heat Transfer and Fluid Flow”, McGraw-Hill.

Pieritz, R. A., Kuhn da Silva, A., Monteiro, M., 2000, “Fundamentos Teóricos da SinFlow Library”, Relatório Técnico, SINMEC, Departamento de Engenharia Mecânica, UFSC, Brasil.

Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P., 1992, “Numerical Recipes in C”, Cambridge University Press.

Thompson, F.F., Warsi, Z.U.A. and Mastin, C.W., 1985, “Numerical Grid Generation – Foundations and Applications”, Elsevier Science Publishing Corporation.

Wood, W.A., 1993, “Multigrid Approach to Incompressible viscous Cavity Flows”, NASA Report, Langley Research Center, USA.

Vahl Davis, G and Jones, I. P., 1983, “Natural Convection in a Square Cavity: A Comparison Exercise”, International Journal for Numerical Methods in Fluids, Vol. 3, pp. 227-248.

Van Doormaal, J.P., Raithby, G.D., 1984, “Enhancements of the Simple Method for Predicting Incompressible Fluid Flows”, Numerical Heat Transfer, Vol. 7, pp 147-163.