Direct Numerical Simulation of Natural Convection Flows

Guillermo Andres Oyarzun Altamirano
guillermo@cttc.upc.edu

Spain, 2007

Abstract

Natural convection is a problem found in many applications, such as solar energy, cooling of electronic equipment, air conditioning, etc. Despite its importance, for the case of turbulent natural convection flows, the available turbulence models (RANS and LES) are still not capable of reproducing important features of the flows such as the transition points for certain ranges of Rayleigh numbers. The best tool to understand the physics involved in this type of turbulent flows is Direct Numerical Simulation (DNS). In the followings sections will expose a step by step way of discretize the governing equations. And this techniques will be applied to the *Differential Heated Cavity Problem.*

1 The Governing Equations

The governing equations of the heat transfer by convection are a state equation (relation among pressure, temperature and density) and conservation equations of mass, linear momentum and energy. Assuming:

- Bidimensional model
- Laminar flow
- Incompressible flow
- Newtonian fluid
- Boussinesq hypothesis (constant physical properties everywhere except in the body forces term)

- Negligible viscous dissipation
- Negligible compression or expansion work
- Non-participating medium in radiation
- Mono-component and mono-phase fluid

the governing equations written in Cartesian coordinates are:

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

$$\rho\frac{\partial u}{\partial t} + \rho u\frac{\partial u}{\partial x} + \rho v\frac{\partial u}{\partial y} = -\frac{\partial p_d}{\partial x} + \mu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \tag{2}$$

$$\rho\frac{\partial v}{\partial t} + \rho u\frac{\partial v}{\partial x} + \rho v\frac{\partial v}{\partial y} = -\frac{\partial p_d}{\partial y} + \mu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) + \rho g\beta(T - T_\infty) \tag{3}$$

$$\rho\frac{\partial T}{\partial t} + \rho u\frac{\partial T}{\partial x} + \rho v\frac{\partial T}{\partial y} = \frac{\lambda}{c_p}\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right) + \frac{\Phi}{c_p} \tag{4}$$

where,

- β Coefficient of volumetric thermal expansion.
- c_p Specific heat.
- λ Conductivity.
- g Gravitational acceleration.
- μ Viscosity.
- p_d Dynamic pressure.
- ρ Density.
- t Time.
- T Temperture.
- u, v Velocity components.
- x, y Spatial coordinates.
- Φ Heat source.

They are partial coupled partial differential equations. The 4 unknowns are the pressure, temperature, and the two velocity components u and v. An appropriate boundary and initial conditions are required to close the problem. Two strong coupling characterize this equations system:

- pressure-velocity. There is no specific pressure equation. For incompressible flows, the pressure is the field that makes the velocity accomplish the mass conservation equation.
- temperature-velocity. This coupling is only present for natural convection, mixed convection or temperature dependent physical properties. In forced convection and constant physical properties, the velocity field does not depend on the temperature field.

All the equations written above $(1-4)$ can be summarized in the convection-diffusion equation:

$$\frac{\partial \rho\phi}{\partial t} + \nabla(\rho\vec{v}\phi) = \nabla(\Gamma\nabla\phi) + S \tag{5}$$

or in Cartesian coordiantes, incompressible flow and constant physical properties:

$$\rho\frac{\partial\phi}{\partial t} + \rho u\frac{\partial\phi}{\partial x} + \rho v\frac{\partial\phi}{\partial y} = \frac{\lambda}{c_p}\left(\frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2}\right) + S \tag{6}$$

The accumulation of ϕ, plus the net convective flow has to be the net diffusive flow plus the generation of ϕ per unit of volume. The diffusive term flows from greater to smaller value of ϕ. According to the convection diffusion equation, we can write a table with the appropriate parameters in order to reproduce the governing equations. See the Table.

Equation	ϕ	Γ	S
Continuity	1	0	0
Momentum in x	u	μ	$-\frac{\partial p_d}{\partial x}$
Momentum in y	v	μ	$-\frac{\partial p_d}{\partial y} + \rho g\beta(T - T_\infty)$
Energy(constant c_p)	T	$\frac{\lambda}{c_p}$	$\frac{\Phi}{c_p}$

2 Finite-Volume Discretization

Integrating the convection-diffusion equation into a rectangular finite volume (see Fig. 1), the discretized equation can be written:[HKV95]

$$\frac{(\rho\phi)_P^{n+1} - (\rho\phi)_P^n}{\Delta t}\Delta x\Delta y + [(\rho u\phi)_e^{n+1} - (\rho u\phi)_w^{n+1}]\Delta y + [(\rho v\phi)_n^{n+1} - (\rho v\phi)_s^{n+1}]\Delta x =$$

$$= \left[\left(\Gamma\frac{\partial\phi}{\partial x}\right)_e^{n+1} - \left(\Gamma\frac{\partial\phi}{\partial x}\right)_w^{n+1}\right]\Delta y + \left[\left(\Gamma\frac{\partial\phi}{\partial y}\right)_n^{n+1} - \left(\Gamma\frac{\partial\phi}{\partial y}\right)_s^{n+1}\right]\Delta x +$$

$$S_P^{n+1}\Delta x\Delta y \tag{7}$$

where the following hypothesis are done:

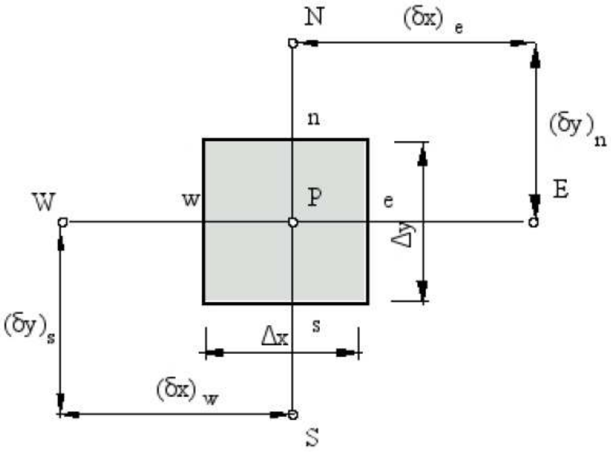


Figure 1: Finite Volume

1. In the integration process, the convective and diffusive flows have been considered constant through each face of the control volume.

2. (spatial deviation)^n=(spatial deviation)$^{n+1}$ (spatial deviation)^w=(spatial deviation)^e (spatial deviation)^s=(spatial deviation)^n

3. The source term: $\int_n^{n+1} \int_w^e \int_s^n S dydxdt = S_P^{n+1} \Delta x \Delta y \Delta t$

2.1 Consistency, stability and convergence

A numerical approximation is consistent when the discretized equations are solution of the differential equation when the spatial and temporal grid tend to zero. Thus, as the grid is refined, truncation errors must tend to zero.[Fed84]

A numerical approximation is stable if the solution obtained is the solution of the discrete equations. Possible roundoff errors, equations couplings, etc can produce instabilities.

A convergent solution is a stable solution that tends to the solution of the differential equations as the the meshes are finer. So, consistency and stability are two necessary and sufficient conditions to get convergence.

3 Analytical solution of the convection-diffusion equation.One dimensional,null source term and steady state solution

For this case the convection - diffusion equation is:[Seg97]

$$\frac{\partial \rho \phi}{\partial t} + \nabla(J) = S \tag{8}$$

where $J = \rho v \phi - \Gamma \nabla \phi$ and is the total flow: convective and diffusive. Rewriting J for the specified conditions:

$$J = \rho u \phi - \Gamma \frac{d\phi}{dx} \tag{9}$$

Assuming a null source term S, equation 8 reduces to:

$$\frac{dJ}{dx} = 0 \tag{10}$$

and integrating over the control volume of figure 2:

$$J_e - J_w = 0 \tag{11}$$

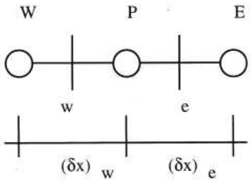


Figure 2: Finite Control Volume for one dimensional problem

The resulting discrete equation can be easily solved imposing the following boundary conditions:

in $x = 0 \rightarrow \phi = \phi_0$
in $x = L \rightarrow \phi = \phi_L$

The analytical solution is:

$$\frac{\phi - \phi_0}{\phi_L - \phi_0} = \frac{exp(Px/L) - 1}{exp(P) - 1} \tag{12}$$

where P is the Peclet number and is defined as $P = \rho u L / \Gamma$ Graphical representation of equation 12 can be seen in figure 3.

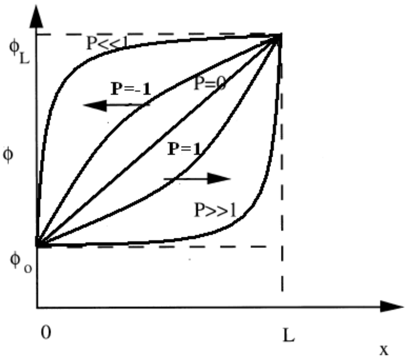


Figure 3: Graphical representation of equation 12

Taking into account this exact analytical profile available, the flows in the faces of the Control Volumes are:

$$J_e = F_e(\phi_P + \frac{\phi_P - \phi_E}{exp(P_e) - 1}) \tag{13}$$

where $P_e = \frac{(\rho u)_e \delta x_e}{\Gamma_e} = \frac{F_e}{D_e}$

Substituting this flows in the discrete equation, we get an equations like:

$$a_P\phi_P = a_E\phi_E + a_W\phi_W \tag{14}$$

The solution of this equation gives the exact solution of the differential convection-diffusion equation in the nodal points whatever the discretization mesh is.

4 General case. Bidimensional and Transcient

The convection-diffusion equation in this situation reads:[Seg98]

$$\frac{\partial(\rho\phi)}{\partial t} + \frac{\partial J_x}{\partial x} + \frac{\partial J_y}{\partial y} = S \tag{15}$$

where $J_x = \rho u\phi - \Gamma\frac{\partial\phi}{\partial x}$ and $J_y = \rho v\phi - \Gamma\frac{\partial\phi}{\partial y}$

Integrating this equation over a control volume, assuming and implicit criteria for the temporal integration and a constant flow at each face of the control volume:

$$\frac{(\rho\phi)_P - (\rho\phi)_P^0}{\Delta t}\Delta x\Delta y + J_e - J_w + J_n - J_s = S_P^n + 1\Delta x\Delta y \tag{16}$$

In order to assure convergence, is better to introduce the continuity equation in the discretized equation:

$$\frac{\partial\rho}{\partial t} + \frac{\partial\rho u}{\partial x} + \frac{\partial\rho v}{\partial y} = 0 \tag{17}$$

which is integrated over a control volume:

$$\frac{\rho_P - \rho_P^0}{\Delta t}\Delta x\Delta y + F_e - F_w + F_n - F_s = 0 \tag{18}$$

where $F = \rho v S$, S is the surface vector of the control volume. Then, we can substract Eq 16 and 18$\cdot\phi_p$, and obtain:

$$\begin{aligned}\frac{(\phi_P - \phi_P^0)\rho_P^0}{\Delta t}\Delta x\Delta y + (J_e - F_e\phi_P) - (J_w - F_w\phi_P) + \\ +(J_n - F_n\phi_P) - (J_s - F_s\phi_P) = S_P^{n+1}\Delta x\Delta y\end{aligned} \tag{19}$$

5 Numerical Schemes

It can be seen that in the discretizated convection-diffusion equation 7 convective and diffusive terms are evaluated at the cell faces, whereas dependent variable ϕ is known at the cell center. The evaluation of the variable at the cell face is carried out by numerical schemes [Tri07].

Conductive flux is calculated as an arithmetic mean:

$$\left(\frac{\partial\phi}{\partial x}\right)_e = \frac{\phi_E - \phi_P}{\delta x_e} or \left(\frac{\partial\phi}{\partial y}\right)_n = \frac{\phi_N - \phi_P}{\delta y_n} \tag{20}$$

The *order* of a numerical scheme is the number of neighboring nodes that are used to evaluate dependent variable at the cell face.

5.1 Low order numerical schemes

These numerical schemes evaluate the variable using nearest nodes: east (E), west (W), north (N) and south (S). Their order is 1 or 2. Most significant low order numerical schemes are:

- Central Difference Scheme (CDS): It is a second order scheme, variable at the cell face is calculated as an arithmetic mean. That is:

$$\phi_e = \frac{1}{2}(\phi_P + \phi_E) \tag{21}$$

- Upwind Difference Scheme (UDS): It is a first order scheme and the value of ϕ at the cell face is equal to the value of ϕ at the grid point on the upwind side of the face. That is:

$$\phi_e = \phi_P \quad if \quad F_e > 0 \tag{22}$$
$$\phi_e = \phi_E \quad if \quad F_e < 0 \tag{23}$$

- Hybrid Difference Scheme (HDS): Uses CDS for low velocities and UDS for high velocities.
- Exponencial Difference Scheme (EDS): It is a second order scheme and the evaluation of the dependent variable at the cell face comes from the exact solution of the convection-diffusion equation in one-dimensional, null source term and steady problem.
- Powerlaw Difference Scheme (PLDS): It is a second order scheme and variable at the cell face is calculated with an approximation of the EDS by a polynomial of fifth degree.

If numerical schemes are introduced in the integrated discretized convection-diffusion equation, it is obtained an algebraic equation for each control volume:

$$a_P\phi_P = a_E\phi_E + a_S\phi_S + a_W\phi_W + a_N\phi_N + b \tag{24}$$

Coefficients a_i can be evaluated with:

$$a_E = D_e \cdot A(|Pe_e|) + max(-F_e, 0) \tag{25}$$
$$a_W = D_w \cdot A(|Pe_w|) + max(F_w, 0) \tag{26}$$
$$a_N = D_n \cdot A(|Pe_n|) + max(-F_n, 0) \tag{27}$$
$$a_S = D_s \cdot A(|Pe_s|) + max(F_s, 0) \tag{28}$$
$$a_P = a_E + a_W + a_N + a_S + \rho_P^n \frac{\Delta x \Delta y}{\Delta t} \tag{29}$$
$$b = \rho_P^n \frac{\Delta x \Delta y}{\Delta t}\phi_P^n + S_P^{n+1}\Delta x \Delta y \tag{30}$$

where:

$$\begin{array}{cccc} D_e = \frac{\Gamma_e \Delta_y}{(\delta x)_e} & D_w = \frac{\Gamma_w \Delta y}{(\delta x)_w} & D_n = \frac{\Gamma_n \Delta x}{(\delta y)_n} & D_s = \frac{\Gamma_s \Delta x}{(\delta y)_s} \\ F_e = (\rho u)_e \Delta y & F_w = (\rho u)_w \Delta y & F_n = (\rho v)_n \Delta x & F_s = (\rho v)_s \Delta x \end{array}$$

and the Peclet number evaluated at the face of the control volume (f) is:

$$P_f = \frac{F_f}{D_f}$$

Numerical Scheme	A(|P|)
UDS	1
CDS	$1 - 0.5(\|P\|)$
HDS	max(0,$(1 - 0.5\|P\|)$)
EDS	$\|P\|/(e^{\|P\|} - 1)$
PLDS	max(0,$(1 - 0.5\|P\|)^5$)

6 The Mesh

6.1 Spacial discretization of the domain

One of the important steps of the Finite Difference method is the spacial domain discretization : This mean divide the domain into a finite number of control volumes. This partition is known like *Mesh* and there are different ways of do it. The three main types are: *Orthogonal Structured Mesh*, *Adaptable No Orthogonal Structured Mesh* and *Unstructured Mesh.*

The choice of the proper mesh is a key factor in the numerical resolution of CFD problems. A lot of problems can be simplify by the choice of a the right mesh. Unstability and convergency problems can be solved as well.

6.1.1 Densification of the Mesh

To improve the simulation of some physics phenomena, is necessary the use of special meshes, more dense that allow us more accurate results. However, it makes no sense use a denser mesh in the whole domain if it is only needed in a specific region.

It's common densify the meshes in the zones with a particular interest and in this way save computational cost. In the following image it's possible watch an example of the method applied in this work. (4).

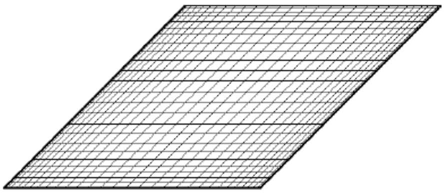


Figure 4: Densification of the mesh in the proximities of the walls.

This technique can be applied to any type of mesh, but we focus our efforts in the orthogonal structured meshes because is a choice that have good results in the *Differential Heated Cavity Problem.* The method is describe it as follows:

Imagine that you want to discretize a region of the space r like in the Figure 5. His length is L_r and is localized at a distance x_r from the origin of the coordinates. The sides of his N control volumes are distributed following the expression:

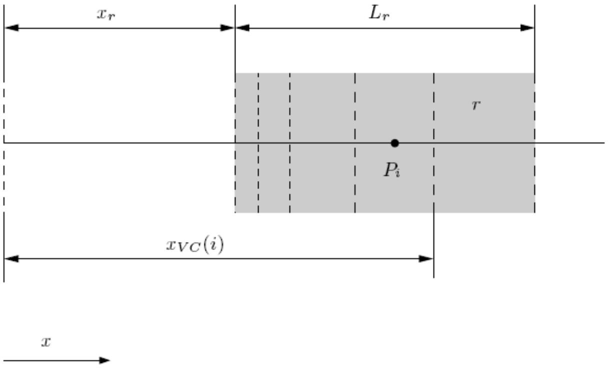


Figure 5: Densification of the mesh for a particular region of the space r.

$$x_{VC}(i) = \delta + \frac{\Delta\delta}{2}\left[1 + tanh\left(2k_{x(r)}\frac{\tilde{i}-1}{n} - k_{x(r)}\right)/tanh(k_{x(r)})\right] \qquad i = i_r, i_r{+}1, \ldots, i_r{+}N_r \tag{31}$$

Where $x_{VC}(i)$ is the cartesian coordinate of the side of the control volume i in the x direction; i_r is the first index of the i control volumes of this region r; $k_{x(r)}$ is the densification factor of the mesh for the region r in the direction x. His value use to be between 0.5 and 1.5.

This expression allows make a symmetrical densification on both sides or just for one of the sides.

- For a symmetrical densification: $n = N_r$, $\delta = x_r$, $\Delta\delta = L_r$ and $\tilde{i} = i - i_r$
- For densify just the right side: $n = 2N_r$, $\delta = x_r$, $\Delta\delta = 2L_r$ and $\tilde{i} = i - i_r + 1$
- For densify just the left side: $n = 2N_r$, $\delta = x_r - L_r$, $\Delta\delta = 2L_r$ and $\tilde{i} = i + N_r - i_r + 1$

In the same way this method can be applied to the other directions y and z.

7 Simple algorithm

Now, that we describe the algorithm used to solve the set of linear equation systems involved in the simulation of the problem. To achieve this we use the iterative algorithm proposed by Patankar [Pat80] and is briefly describe it in the following steps.

1. Data Set: This is the process in which the constants, the velocities,temperatures,pressures maps are initializated, plus some parameters of the program like discretization squemes, convergency numbers, dimensions of the mesh, etc.
2. Iterative Loop: In this section, at every iteration the variables U,V and T stay constants.
3. Numerical Resolution: The momentum equations are solved using the algorithm *TDMA* with a *Gauss-Seidel*. The resolution is relaxed, obtaining new values of U^* and V^* which will be use in the following steps.
4. Pressure Correction Resolution: This step is the numerical resolution of the pressure correction system. Obtaining the values of p' for each node.
5. Pressure and Velocity Correction: The velocities and the pressures are corrected in this way, $p = p^* + p'$, $u = u^* + u'$, $v = v^* + v'$.
6. Temperature Resolution: In the same way that the velocities, the temperature is resolved with the algorithm *TDMA* with a *Gauss-Seidel*. Obtaining the new maps of temperatures T.
7. Finally if the algorithm converges the algorithm stops and if it not, a new iteration is computed.

In the figure 6 is shown the flux diagram of the algorithm.

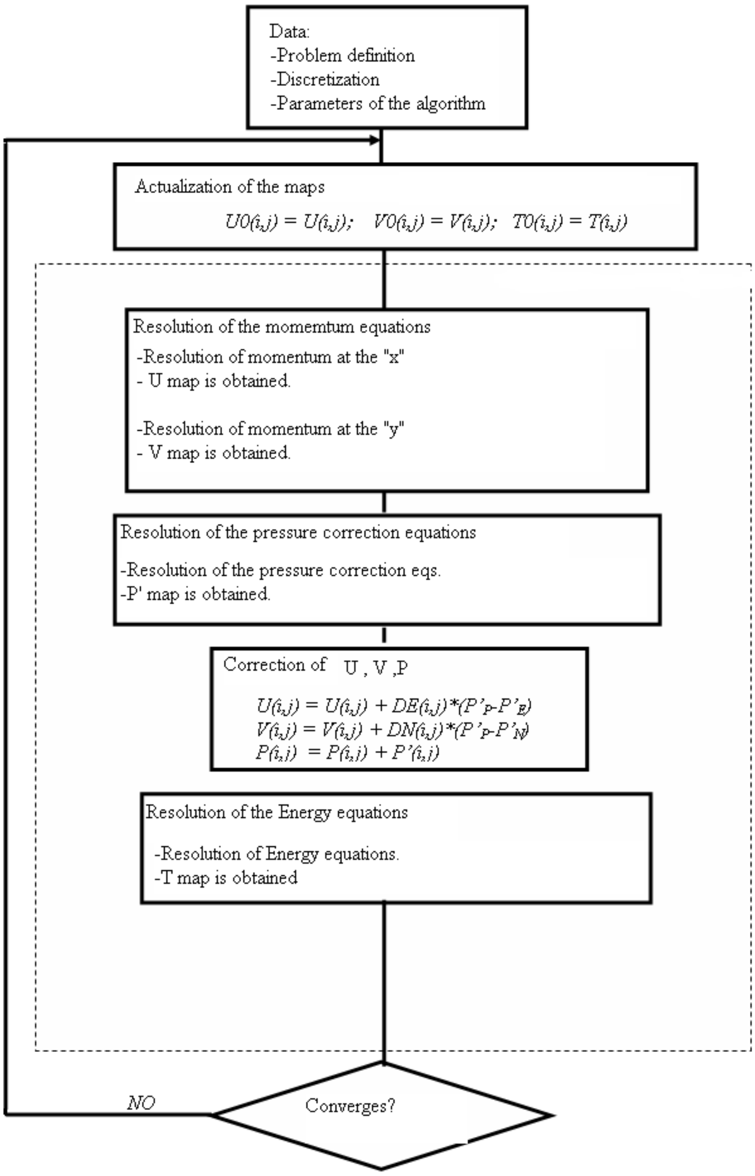
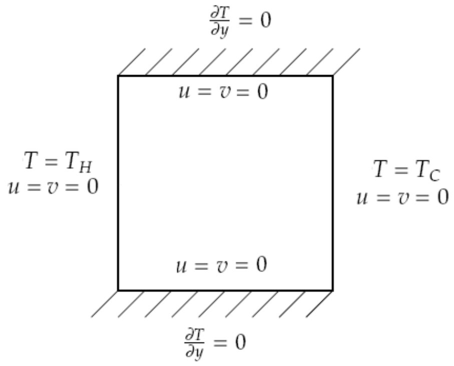

Data:
-Problem definition
-Discretization
-Parameters of the algorithm
Actualization of the maps
$U0(i,j) = U(i,j); \quad V0(i,j) = V(i,j); \quad T0(i,j) = T(i,j)$
Resolution of the momemtum equations
-Resolution of momentum at the "x"
- U map is obtained.
-Resolution of momentum at the "y"
- V map is obtained.
Resolution of the pressure correction equations
-Resolution of the pressure correction eqs.
-P' map is obtained.
Correction of U , V ,P
$U(i,j) = U(i,j) + DE(i,j)*(P'_P-P'_E)$
$V(i,j) = V(i,j) + DN(i,j)*(P'_P-P'_N)$
$P(i,j) = P(i,j) + P'(i,j)$
Resolution of the Energy equations
-Resolution of Energy equations.
-T map is obtained
NO
Converges?

Figure 6: Simple algorithm.

8 Differential Heated Cavity Problem

Another Benchmark used to validate the global code is the one known as *Differential Heated Cavity Problem.* This case consist in the simulation of a laminar flow inside of a square cavity where two opposite faces have different temperatures. The boundary conditions are shown in the Figure.



$\frac{\partial T}{\partial y} = 0$
$u = v = 0$
$T = T_H$
$u = v = 0$
$T = T_C$
$u = v = 0$
$u = v = 0$
$\frac{\partial T}{\partial y} = 0$

Figure 7: Geometry of the problem

The Differential Heated Cavity is a benchmark of natural convection. The velocity field is generated because of the gradient of the temperatures. This case allow verify the governing equations of continuity, of conservation of momentum and the conservation of energy. Also is useful to probe the *Boussinesq* hypothesis.

The problem has been adimensionalized with the following variables:

$$x^* = \frac{x}{L_x} \tag{32}$$

$$y^* = \frac{y}{L_x} \tag{33}$$

$$u^* = \frac{uL_x}{a} \tag{34}$$

$$v^* = \frac{vL_x}{a} \tag{35}$$

$$T^* = \frac{T - T_H}{\Delta T} \tag{36}$$

Where,

$$a = \frac{\lambda}{\rho c_p}$$

Substituting the adimensional variables in the governing equations is easily probe that the problem depends only of the adimensional numbers of Prandtl and Rayleigh:

$$\begin{aligned} Pr &= \frac{\mu c_p}{\lambda} \qquad (37) \\ Ra &= \frac{g\beta\rho^2 \Delta T L^3 c_p}{\lambda\mu} \qquad (38) \end{aligned}$$

The mesh: For the discretization of the domain has been used an orthogonal mesh. This mesh has been densify close to the left and right sides of the cavity, because in that places is where the gradients of temperatures are more strong, also the fact that the velocities are bigger, what implies that a bigger accuracy is needed. In this way is possible get better results than with a uniform mesh with the same number of nodes. The results where tested on three different meshes with different number of nodes : 23 × 23, 43 × 43 y 83 × 83.

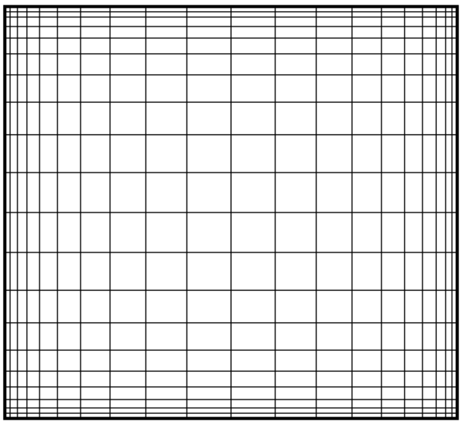


Figure 8: Discretization of the spacial domain with a orthogonal mesh with symmetric densification.

For the validation of the algorithm the results were compared with the benchmark proposed by De Vahl Davis (1983) [Dav83]. Each case were tested with the same Prandtl number ($Pr = 0,71$) and for different Rayleigh numbers, that were obtained modifying the value of ρ

The results compared are:

- u^*_{max},y^*_{max}: Maximum value and position of the horizontal component of the velocity u at the vertical symmetry line of the cavity.
- v^*_{max},x^*_{max}: Maximum value and position of the vertical component of the velocity v at the horizontal symmetry line of the cavity.
- Nu_{max},$y^*_{Nu_{max}}$ Maximum value and position of the local Nusselt at the hot wall.
- Nu_{min},$y^*_{Nu_{min}}$ Minimum value and position of the local Nusselt at the hot wall.
- Nu_{mig} Average of the Nusselt number at the hot wall.

The following Figures show the numerical results obtained and the relative error respect the Benchmark.

$Ra = 10^3$							
	mesh 23*x*23	*error*	*mesh* 43*x*43	*error*	*mesh* 83*x*83	*error*	*Benchmark*
u^*_{max}	0.564	0.845	0.738	0.798	0.970	0.734	3.649
y^*_{umax}	0.922	0.134	0.910	0.119	0.892	0.097	0.813
v^*_{max}	0.571	0.845	0.741	0.800	0.971	0.737	3.697
x^*_{vmax}	0.922	4.178	0.910	4.111	0.892	4.012	0.178
Nu_{max}	1.081	0.282	1.099	0.270	1.124	0.253	1.505
$y^*_{Nu_{max}}$	0.117	0.275	0.135	0.463	0.159	0.724	0.092
Nu_{min}	0.939	0.357	0.920	0.330	0.897	0.297	0.692
$y^*_{Nu_{min}}$	0.971	0.029	0.988	0.012	0.995	0.005	1.000
Nu_{mig}	1.010	0.097	1.009	0.097	1.011	0.096	1.118

Table 1: Comparison of the numerical results for a $Ra = 10^3$.

$Ra = 10^4$							
	mesh 23*x*23	*error*	*mesh* 43*x*43	*error*	*mesh* 83*x*83	*error*	*Benchmark*
u^*_{max}	5.307	0.672	6.811	0.579	8.532	0.473	16.178
y^*_{umax}	0.922	0.120	0.910	0.105	0.892	0.084	0.823
v^*_{max}	5.620	0.714	7.011	0.643	8.627	0.560	19.617
x^*_{vmax}	0.922	6.745	0.927	6.793	0.903	6.586	0.119
Nu_{max}	1.924	0.455	2.115	0.400	2.341	0.337	3.528
$y^*_{Nu_{max}}$	0.169	0.180	0.193	0.348	0.224	0.569	0.143
Nu_{min}	0.644	0.099	0.599	0.022	0.583	0.005	0.586
$y^*_{Nu_{min}}$	1.000	0.000	0.998	0.002	0.999	0.001	1.000
Nu_{mig}	1.271	0.433	1.351	0.398	1.453	0.352	2.243

Table 2: Comparison of the numerical results for a $Ra = 10^4$.

It can be concluded that while the mesh gets denser, the results improve, and converge to the values of the benchmark, reducing the relative error. This

$Ra = 10^5$							
$Ra = 10^5$	*mesh* 23*x*23	*error*	*mesh* 43*x*43	*error*	*mesh* 83*x*83	*error*	*Benchmark*
u^*_{max}	34.403	0.009	39.635	0.141	43.265	0.246	34.730
y^*_{umax}	0.922	0.078	0.910	0.064	0.922	0.078	0.855
v^*_{max}	43.320	0.368	46.032	0.329	48.313	0.296	68.590
x^*_{vmax}	0.950	13.398	0.955	13.470	0.951	13.412	0.066
Nu_{max}	5.781	0.251	5.821	0.246	5.920	0.233	7.717
$y^*_{Nu_{max}}$	0.169	1.083	0.227	1.807	0.243	2.002	0.081
Nu_{min}	0.802	0.100	0.801	0.099	0.806	0.106	0.729
$y^*_{Nu_{min}}$	1.000	0.000	0.998	0.002	1.000	0.000	1.000
Nu_{mig}	3.280	0.274	3.332	0.263	3.355	0.258	4.519

Table 3: Comparison of the numerical results for a $Ra = 10^5$.

$Ra = 10^6$							
	mesh 23*x*23	*error*	*mesh* 43*x*43	*error*	*mesh* 83*x*83	*error*	*Benchmark*
u^*_{max}	71.100	0.100	66.950	0.036	65.735	0.017	64.630
y^*_{umax}	0.883	0.038	0.865	0.018	0.855	0.006	0.850
v^*_{max}	214.275	0.023	219.048	0.001	220.103	0.003	219.360
x^*_{vmax}	0.029	0.234	0.034	0.097	0.037	0.017	0.038
Nu_{max}	17.420	0.028	17.594	0.018	17.526	0.022	17.925
$y^*_{Nu_{max}}$	0.050	0.309	0.045	0.184	0.037	0.017	0.038
Nu_{min}	1.273	0.288	1.145	0.158	1.064	0.076	0.989
$y^*_{Nu_{min}}$	0.971	0.029	0.988	0.012	0.997	0.003	1.000
Nu_{mig}	8.723	0.009	8.863	0.007	8.922	0.014	8.800

Table 4: Comparison of the numerical results for a $Ra = 10^6$.

proofs that the algorithm works for this kind of problems, but some errors are found for the case $Ra = 10^5$, and probably are caused for a oscillatory convergence.

The next images show the distribution of temperatures for each one of the simulated cases. The following pictures show the coupling existing between the equations of conservation of momentum and the equation of conservation of energy. As the Rayleigh number increases, the velocities and the temperature gradient increase as well. This is produced because of the contribution of the Boussinesq term that become more important with high Rayleigh numbers.

In the cases with lower Rayleigh numbers the diffusion is more important than the convection, but while this number increases, the convection start getting more importance and the temperature distribution get stratified. This has as result that the gradient for temperatures increases at the walls and so the Nusselt number.

The flow lines are shown for the different cases resolved. It can be seen the evolution of the flow as the number Rayleigh increases and the coupling between temperatures and velocities.

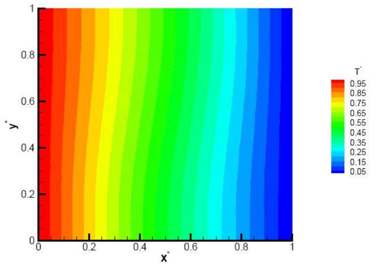


Figure 9: Temperatures distribution T^* for a $Ra = 10^3$.

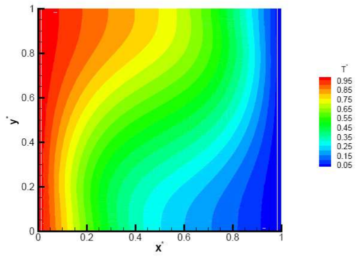


Figure 10: Temperatures distribution T^* for a$Ra = 10^4$.

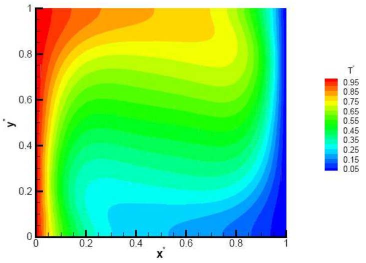


Figure 11: Temperatures distribution T^* for a $Ra = 10^5$.

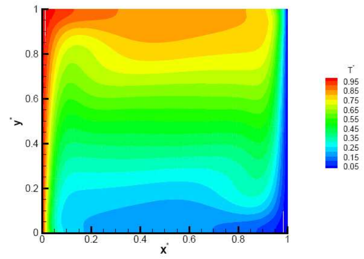


Figure 12: Temperatures distribution T^* for a $Ra = 10^6$.

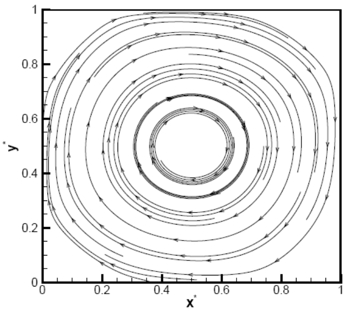


Figure 13: Flow lines for a $Ra = 10^3$.

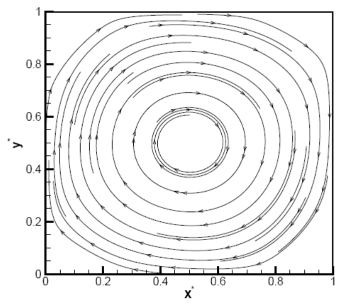


Figure 14: Flow lines for a $Ra = 10^4$.

Even if the errors increases for the case ($Ra = 10^5$), the algorithm can simulate the physics of the problem. In the following images it can be shown the evolution of the temperatures flows. The different temperatures at each side wall of the cavity produces a heat flow and this flow produces a convective movement in the fluid inside the cavity. Which totally fits with the expected

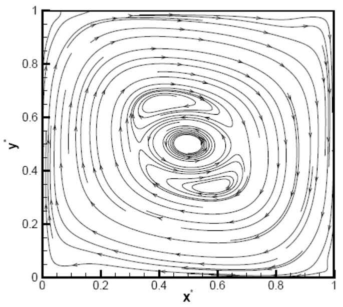


Figure 15: Flow lines for a $Ra = 10^5$.

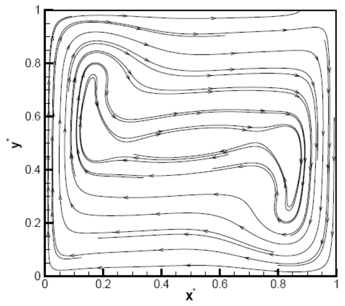


Figure 16: Flow lines for a $Ra = 10^6$.

behavior.

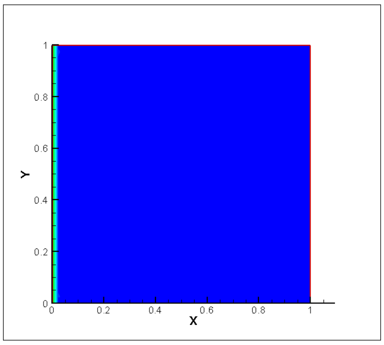


Figure 17: Differential Heated cavity. Initial Temperature distribution.

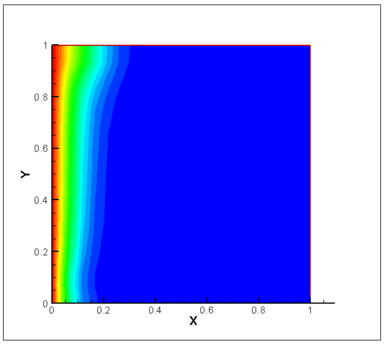


Figure 18: Differential Heated cavity. Iteration 25.

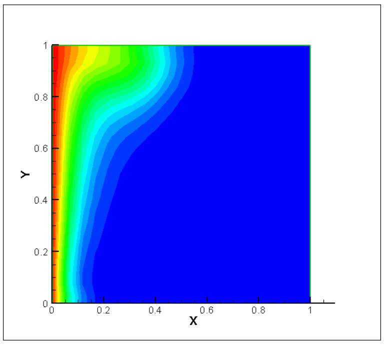


Figure 19: Differential Heated cavity. Iteration 50.

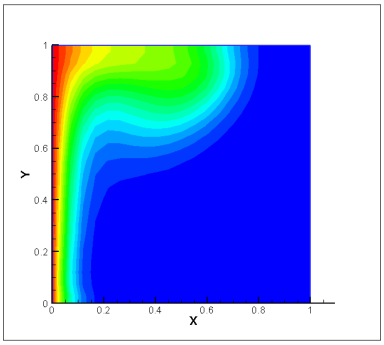


Figure 20: Differential Heated cavity. Iteration 75.

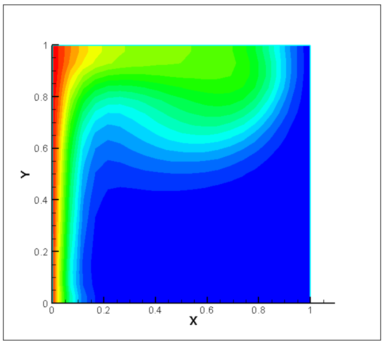


Figure 21: Differential Heated cavity. Iteration 100.

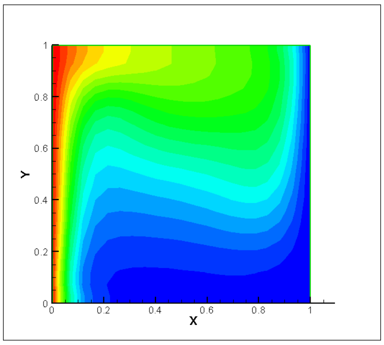


Figure 22: Differential Heated cavity. Iteration 200.

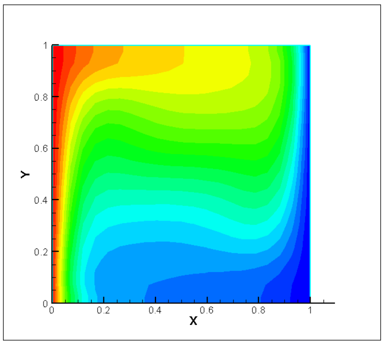


Figure 23: Differential Heated cavity. Iteration 400.

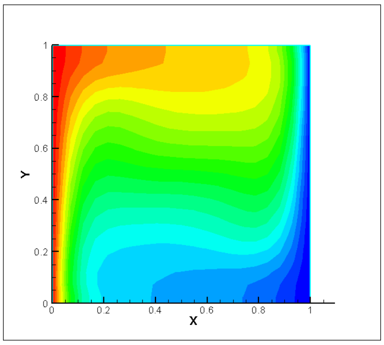


Figure 24: Differential Heated cavity. Iteration 800.

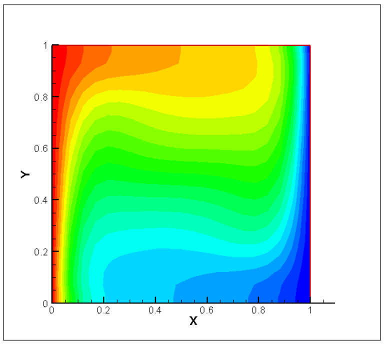


Figure 25: Differential Heated cavity. Iteration 1000.

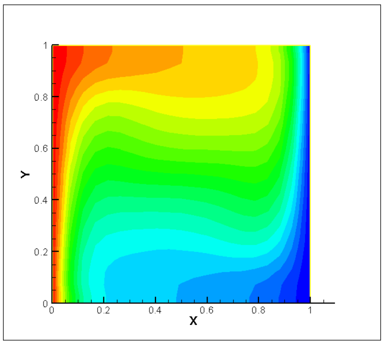


Figure 26: Differential Heated cavity. Iteration 1200.

8.1 Conclusions and Future Work

The *CFD* methods are a primary tool in the engineering and in the research world. The CFD problems are presents all around the world, sometimes in the most common things, like the design of radiators, design of cars or the design of airplanes.

This work was focused in the testing of different discretization squemes, concluding that the *Central Difference Squeme* present some convergency problems for high Peclet numbers.

Also where tested different mesh methods, obtaining better results with a mesh more dense at the sides of the cavity, however this depends on the problem, and a proper choose of the mesh can improve the results.

The experiments probe that the simple algorithm is useful to resolve problems with simple geometry and also the simulations results were agree with the theoretical physic behavior.

The numerical resolution on problems of Heat Transfer and Fluid Dynamics is a field that constantly evolve. Every year new methods appear and in that way is possible resolve new and more difficult problems.

It will be very interesting add new features at the code developed. In particular, some precision problems has been detected, and the use of new tools could be fix them. The use of high order squemes can be useful to improve the simulation.

How ever, the most important work by now will be adapt the code, taking advantage of the parallel computing and in that way improve the quality of the solutions obtained. It is important too, test the code for other kind of problems, with a more practical view.

References

[Dav83] G. De Vahl Davis. Natural convection of air in a square cavity: a benchmark numerical solution. International Journal for Numerical Methods in Fluids, pages 3:249264, 1983.

[Fed84] R.P. Fedorenko. The speed of convergence of one itereative process. U.S.S.R. Computational Mathematics and Mathematical Physics,4:227-235, 1984.

[HKV95] W. Malalasekera H. K. Versteeg. *An Introduction to Computational Fluid Dynamics: The Finite Volume Method.* Research Studies Pr, 1995.

[Pat80] Suhas V. Patankar. *Numerical Heat Transfer and Fluid Flow.* McGraw-Hill Inc.,US, 1980.

[Seg97] C. D. Pérez Segarra. Topics on fluid mechanics heat transfer mathematical formulation (ii). diferential formulation. Centre Tecnològic de Transferència de Calor (CTTC). Univeritat Politècnica de Catalunya (UPC), 1997.

[Seg98] C. D. Pérez Segarra. Topics on fluid mechanics heat transfer mathematical formulation (i). integral formulation. Centre Tecnològic de Transferència de Calor (CTTC). Univeritat Politècnica de Catalunya (UPC), 1998.

[Tri07] Xavi Trias. Apunts de seminari de mètodes numèrics en convecció-difusió. Centre Tecnològic de Transferència de Calor (CTTC). Univeritat Politècnica de Catalunya (UPC), 2007.