



Scat Report September 2007 – August 2008

CTTC – Universitat Politècnica de Catalunya

Universidad Técnica Federico Santa María

Eduardo Sufan Piñeiro

Contents

List of Figures	5
1 Previous understandings	7
1.1 Control volume formulation	7
1.2 Linear algebraic equation systems solver	8
1.2.1 Gauss - Seidel	8
1.2.2 TDMA: TriDiagonal Matrix Algorithm	9
1.2.3 LU Decomposition	10
1.3 The Staggered Grid	12
2 Conduction	15
2.1 A Two-dimensional Transient Conduction Problem	17
2.1.1 Discretization equations	18
2.1.2 Results	22
3 Convection-Diffusion	25
3.1 A Two-Dimensional Steady Convection-Diffusion Equation: The Smith Hutton Problem	27
3.1.1 Discretization equations	28
3.1.2 Discussion of results	29
4 Navier-Stokes equations	37
4.1 Driven Cavity problem	38
4.1.1 Discretization	39
4.1.2 Discussion of results	44
Bibliography	55

List of Figures

1.1	a)Domain divided in volume controls, b) Control volume	7
1.2	Staggered grid	13
2.1	Grid cluster one-dimensional	15
2.2	Distances associated with the control volume face	16
2.3	Schema of conduction problem	17
2.4	Prescribed heat flow boundary condition	20
2.5	Convection boundary condition	21
2.6	Evolution of the solution for different mesh sizes	22
2.7	Temperature field at $t = 5000[s]$	23
2.8	Temperature in time for two different points on the domain	23
3.1	Convection-diffusion control volume	25
3.2	Effect of the <i>Peclet</i> number on convection-diffusion problems	27
3.3	Geometry specifications and streamlines	28
3.4	Discretization of the domain	29
3.5	Outlet ϕ for different mesh size using Upwind and $P = 1000$	30
3.6	Results for the outlet ϕ of the ideal numerical result and the one achieved in this work	31
3.7	Velocity field	31
3.8	ϕ in the domain for <i>Peclet</i> = 10	32
3.9	ϕ in the domain for <i>Peclet</i> = 1000	32
3.10	ϕ in the domain for <i>Peclet</i> = 10^6	33
3.11	ϕ at the output for <i>Peclet</i> = 10 and different numerical schemes	33
3.12	ϕ at the output for <i>Peclet</i> = 10^3 and different numerical schemes	34
3.13	False diffusion	34
3.14	3D ϕ for <i>Peclet</i> = 10	35
3.15	3D ϕ for <i>Peclet</i> = 1000	35
3.16	3D ϕ for <i>Peclet</i> = 10^6	36
4.1	Driven Cavity problem.	38
4.2	Structural mesh used in the simulation	39
4.3	Control volume for pressure variable	40
4.4	SIMPLE algorithm	42
4.5	Fractional step method algorithm	45
4.6	u and v results for different mesh sizes	45
4.7	Error between Benchmark and the result achieved using central difference and upwind schemes for u and v solutions	46

4.8	<i>u</i> results for different Reynolds numbers	48
4.9	<i>v</i> results for different Reynolds numbers	49
4.10	Stream lines for different Reynolds numbers	50
4.11	Zoom at eddies in cavity corners for $Re = 10000$	51
4.12	<i>u</i> results for all Reynolds numbers	51
4.13	<i>v</i> results for all Reynolds numbers	52
4.14	Comparison between Fractional Step and SIMPLE algorithms for <i>u</i> velocity	52
4.15	Comparison between Fractional Step and SIMPLE algorithms for <i>v</i> velocity	53

Chapter 1

Previous understandings

In this chapter a previous numerical introduction is explained to make the understanding of this work more efficient and easy. We will start with the control volume formulation for the discretization process and then the numerical solver to solve the linear system of equations generated with the discretization of the equations that govern the physical phenomena shown along this work.

1.1 Control volume formulation

In order to make a discretization of the equations governing a certain phenomena its necessary to choose a integration method such as *finite elements*, *finite volume* and others. In our case we work with finite volume method, so the domain will be divided in small volumes which contain a node that represets a piece of the whole domain and that has physical properties constant in all the control volume and any property ϕ remains constant also in the whole volume.

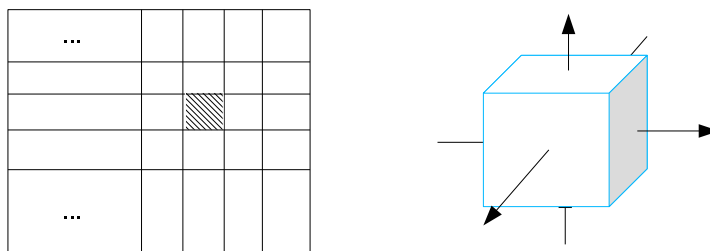


Figure 1.1: a) Domain divided in volume controls, b) Control volume

Figure 1.1a shows the typical domain of a determined problem divided into several control volumes, and figure 1.1b shows a particular control volume where the arrows represent the flows o a determined flux.

The main characteristics of the finite volume methods are the following

- + The control volume is surrounding a node point in a mesh that in addition generates the whole domain,

- + fluxes are calculated in the faces of each volume and
- + the flux leaving one control volume is identical to the flux entering the adjacent one.

The integration of the governing equations takes place on one control volume and the numerical approximations are made evaluating the properties in the walls of the control volume using the neighbors nodes. In this section we won't make the integral because during the rest of this work the discretization takes place for special cases that work as an example of how the finite volume method works.

1.2 Linear algebraic equation systems solver

In this section three different solvers are explained. The first two are iterative, that means that for reaching a solution of a linear system, the solution is first guessed and in each step or iteration the solver gets closer to the solution until an acceptable error is reached. The third one is a direct solver, in which the solution is reached at once, solving the whole matrix.

The equation system is the type $Ax = b$ or in its expanded form:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & \cdots & a_{2n} \\ & a_{32} & a_{33} & a_{34} & & \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ & & & a_{n-1,n-2} & a_{n-1,n-1} & a_{n-1,n} \\ a_{n,1} & \cdots & & & a_{n,n-1} & a_{n,n} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ \vdots \\ b_n \end{pmatrix}$$

Where the unknowns matrix is x .

1.2.1 Gauss - Seidel

The Gauss-Seidel algorithm is a point-by-point solver, meaning that the solution is calculated for one of the many unknowns each time, as the equation 1.1 shows for $i = 1$ to n .

$$x_i^{k+1} = \frac{1}{a_{ij}} \left(b_i - \sum_{j < i} a_{ij} x_j^{k+1} - \sum_{j > i} a_{ij} x_j^k \right) \quad (1.1)$$

where the superindex $k + 1$ means the value calculated in the actual iteration and the superindex k makes reference to the iteration calculated before. In other words the Gauss-Seidel algorithm makes use of the newest value known of the x variable.

Notice that the solution is calculated row by row putting all the unknowns at one side leaving only one to be solved. Then it passes into the next unknown until all are solved, passing then to the next row and so on. When all the matrix is solved the residual is calculated as in equation 1.2 expecting it to decrease

as the iterative motion runs until a sufficiently small residual is reached and it should be zero for an exact solution.

$$r = \sqrt{\sum_{i=1}^n \left(x_i - \left(b_i - \sum_{j \neq i} a_{ij} x_j \right) \right)^2} \quad (1.2)$$

This residual equation is used for all the iterative solutions in this work.

1.2.2 TDMA: TriDiagonal Matrix Algorithm

The name refers to the fact that it uses only the non-zero diagonals of the matrix A previously mentioned. In fact for one-dimensional cases when that matrix of coefficients is written, all the nonzero coefficients align themselves along three diagonals of the matrix.

For convenience of presenting the algorithm the nomenclature will be changed. The linear equations is as follow:

$$a_i x_i = b_i x_{i+1} + c_i x_{i-1} + d_i \quad (1.3)$$

where a represents the coefficients in the main diagonal, $-b$ represents the value of the coefficients in the superdiagonal and $-c$ represents the value of the coefficients in subdiagonal finely b represents the solution vector. Notice the minus sign included to make the equation fit to the standard notation of the type $Ax = b$.

This equation shows a relation between the variable x_i and its neighbors x_{i+1} and x_{i-1} for $i = 1, 2, 3 \dots, n$. But taking into account that $c_1 = 0$ and $b_n = 0$ the unknowns x_0 and x_{n+1} will not have real importance in the solution algorithm. These means that x_1 will only depend and will be solve in terms of x_2 and this relation reduces a relation between x_2 and x_3 . In other words x_2 can be obtain in terms of x_3 and so on until x_n where the meaning of x_{n+1} has no importance as said before, that means that the value of x_n will be actually known at this stage. Having this solution in hand, it enable us to make a backward substitution in x_{n-1} which is obtained in terms of x_n and so on.

In other words the TDMA algorithm seeks a relation of the type:

$$x_i = P_i x_{i+1} + Q_i \quad (1.4)$$

which making a substitution into equation 1.3 leads to

$$a_i x_i = b_i x_{i+1} + c_i (P_{i-1} x_i + Q_{i-1}) + d_i \quad (1.5)$$

rearranging we can obtain the values of the coefficients P_i and Q_i

$$P_i = \frac{b_i}{a_i - c_i P_{i-1}} \quad (1.6)$$

$$Q_i = \frac{d_i + c_i Q_{i-1}}{a_i - c_i P_{i-1}} \quad (1.7)$$

These are recurrence relations of the variables P_i and Q_i in terms of P_{i-1} and Q_{i-1} . So for start the process we use the fact that $c_1 = 0$ leading to

$$P_1 = \frac{b_1}{a_1} \quad \text{and} \quad Q_1 = \frac{d_1}{a_1} \quad (1.8)$$

At the end when $i = n$ we have that $b_n = 0$ this leads to $P_n = 0$ and we obtain

$$x_n = Q_n \quad (1.9)$$

And now with all the P_i and Q_i coefficients calculated using the recurrence relations 1.6 and 1.7, and taking as a first step for the solution equation 1.9 we can obtain all the solution in one step using a backward substitution in equation 1.4.

One important fact of this method is that it is not iterative but only work for one dimensional problems which fortunately have the coefficient matrix only with three nonzero diagonals.

For two or three dimensional cases, this algorithm can be used, but it changes its essence and became an iterative one. For example, in two dimensional cases the matrix A became a penta-diagonal, meaning that it has five diagonals. One for every neighbor and one for it selves. Then to be able of using the TDMA a tridiagonal matrix must be generated, for that reason the *north* and *south* neighbors are taken as known and moved into the d matrix. Then you can solve the problem as one-dimensional. Once all the unknowns are solved, you pass to the next line, solving now for the old *north* unknowns, and so on. That is performed line-by-line until all the domain is solved. Then you calculate the residual as in equation 1.2 and the same procedure is performed. Notice that you can make it going line-by-line in the x-direction or in the y-direction or in both alternatively. For a three dimensional case its the same but having more neighbors, the *top* and *bottom* ones.

1.2.3 LU Decomposition

The decomposition LU writes the matrix \mathbf{A} previously described at the beginning of this chapter as the product of two matrices.

$$\mathbf{L} \cdot \mathbf{U} = \mathbf{A} \quad (1.10)$$

where \mathbf{L} stands for the *lower triangular* having elements nonzero only on the diagonal and below, and \mathbf{U} is the *upper triangular* having elements nonzero only on the diagonal and above. An example of this decomposition is shown for a 4×4 matrix.

$$\begin{pmatrix} \alpha_{11} & 0 & 0 & 0 \\ \alpha_{21} & \alpha_{22} & 0 & 0 \\ \alpha_{31} & \alpha_{32} & \alpha_{33} & 0 \\ \alpha_{41} & \alpha_{42} & \alpha_{43} & \alpha_{44} \end{pmatrix} \cdot \begin{pmatrix} \beta_{11} & \beta_{12} & \beta_{13} & \beta_{14} \\ 0 & \beta_{22} & \beta_{23} & \beta_{24} \\ 0 & 0 & \beta_{33} & \beta_{34} \\ 0 & 0 & 0 & \beta_{44} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}$$

The main importance of this decomposition is the facility to solve the system once the decomposition is made. Because having a triangular set of equations

leads to a trivial solution of the system only by substitution starting from the corner which has only one coefficient nonzero, and solving from there.

Now solving the linear set taking the decomposition leads to two different linear systems which can be solved in a easier way.

$$A \cdot x = (L \cdot U) \cdot x = L \cdot (U \cdot x) = b \quad (1.11)$$

Using this decomposition we can solve the vector y such that

$$L \cdot y = b \quad (1.12)$$

and then solving

$$U \cdot x = y \quad (1.13)$$

The solution of the *lower triangular* (equation 1.12) can be solve by *forward substitution* as follows,

$$\begin{aligned} y_1 &= \frac{b_1}{\alpha_{11}} \\ y_i &= \frac{1}{\alpha_{ii}} \left[b_i - \sum_{j=1}^{i-1} \alpha_{ij} y_j \right] \quad i = 2, 3, \dots, n \end{aligned} \quad (1.14)$$

and 1.13 can be solve by *backsubstitution* as

$$\begin{aligned} x_n &= \frac{b_n}{\beta_{nn}} \\ x_i &= \frac{1}{\beta_{ii}} \left[y_i - \sum_{j=i+1}^n \beta_{ij} x_j \right] \quad i = n-1, n-2, \dots, 1 \end{aligned} \quad (1.15)$$

To perform the **LU** decomposition we can make use of *Court's algorithm* which solves the set of equations for all the α 's and β 's by arranging the equations in a certain order.

- Set $\alpha_{ii} = 1$ for $i = 1, \dots, n$
- For each $j = 1, 2, 3, \dots, n$ do two procedures:
 - First for $i = 1, 2, \dots, j$

$$\alpha_{i1}\beta_{1j} + \alpha_{i2}\beta_{2j} + \dots + \alpha_{ii}\beta_{jj} = a_{ij} \quad (1.16)$$

$$\alpha_{ii} = 1 \quad (1.17)$$

to solve β_{ij}

$$\beta_{ij} = a_{ij} - \sum_{k=1}^{i-1} \alpha_{ik}\beta_{kj} \quad (1.18)$$

– Second solve α_{ij} for $i = j + 1, j + 2, \dots, n$

$$\alpha_{ij} = \frac{1}{\beta_{jj}} \left(a_{ij} - \sum_{k=1}^{j-1} \alpha_{ik} \beta_{kj} \right) \quad (1.19)$$

Its important to do both procedures before going on to the next j .

The king of problems solved in this work leads to a matrix **A** band diagonal, that means that the nonzero elements are group in a diagonal band with $m1$ elements in the subdiagonal and $m2$ elements in the superdiagonal plus the diagonal it selves leaving the rest of the space occupied with zero. With this in mind, the code for solving the linear system use a smaller matrix called $\mathbf{a}_{n \times (m1+m2+1)}$ which has only the band diagonal but rotated in 45° saving space for computing¹.

1.3 The Staggered Grid

In problems of convection and diffusion. When both the velocity field and the pressure field are to be solved. A very well known problem appears when using the normal grid, where all quantities are in the main grid in the center of the control volume. The solution can converge but not necessarily reach physically realistic results, in particular, the pressure or the velocity fields can develop a *checkerboard* pattern which can satisfy continuity for the velocity and also the momentum equation in the pressure term. This pattern is the result of the discretization it selves, by taking the values not in the adjacent nodes but one into another. In the case of a staggered grid the nodes of the velocity are in the faces of the control volume, meaning that there is another mesh for the velocity in every direction. The figure 1.2 shows a grid node P in the main grid with the index I, J . The velocity in the x-coordinate is allocate in the faces *east* and *west* of the main control volume, in this case with the index i, J because this staggered grid is only rearrange in the x direction maintaining the alignment in the y-coordinate with the main grid. In the figure the control volume for u is with a hatching of 45° . The same occurs with the velocity in the y-direction, that maintains the index in the x-coordinates, while moves half control volume in the y-direction, for this staggered mesh in y-direction the index are I, j and in the figure the hatching is in -45° .

¹To see the C code enter:
<http://www.fizyka.umk.pl/nrbook/c2-4.pdf>

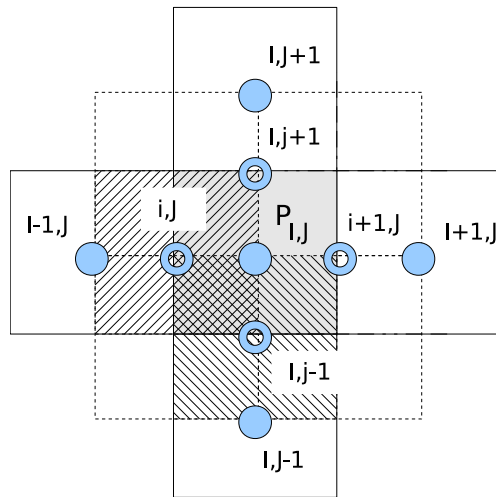


Figure 1.2: Staggered grid

Chapter 2

Conduction

Taking in consideration the steady state heat conduction phenomenon governed by

$$\nabla \cdot (k\nabla T) + S = 0 \quad (2.1)$$

where k is the thermal conductivity and S is the rate of heat generation per unit of volume.

To derive the discretization equation we employ the grid shown in figure 2.1 for a one-dimensional case, the two and three dimensional are easily generated starting from this case.

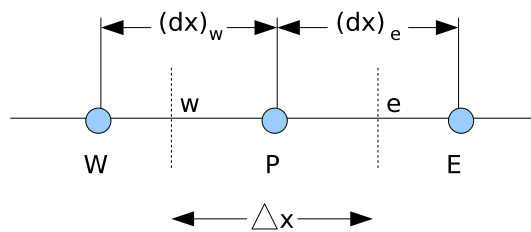


Figure 2.1: Grid cluster one-dimensional

The letters W and E represent the *west* and *east* nodes and the letter P represent the node in the control volume studied. While the letters w and e represent the faces of the control volume. The volume of this control volume is $dx \cdot 1 \cdot 1$

The equation to integrate over the control volume is the following

$$\frac{d}{dx} \left(k \frac{dT}{dx} \right) + S = 0 \quad (2.2)$$

resulting in

$$\left(k \frac{dT}{dx} \right)_e - \left(k \frac{dT}{dx} \right)_w + \int_w^e S dx = 0 \quad (2.3)$$

Now we have to evaluate dT/dx in the faces of the control volume. To do that a linear profile can be assumed, taking in to account that the value of T is supposed to prevail over the whole control volume surrounding the grid point. The resulting equation is finely

$$\frac{k_e (T_E - T_P)}{(dx)_e} - \frac{k_w (T_P - T_W)}{(dx)_w} + \bar{S}\Delta x = 0 \quad (2.4)$$

where \bar{S} is the average rate of heat generation inside the control volume.

In other word the discretization equation can be rewritten in the following form:

$$a_P T_P = a_E T_E + a_W T_W + b \quad (2.5)$$

where

$$a_E = \frac{k_e}{(dx)_e} \quad (2.6)$$

$$a_W = \frac{k_w}{(dx)_w} \quad (2.7)$$

$$a_P = a_E + a_W \quad (2.8)$$

$$b = \bar{S}\Delta x \quad (2.9)$$

It's important to take into account that the thermal conductivity k is defined in the wall of the control volume and when the heat flux is calculated from one control volume should be the same that the heat flux entering the adjacent control volume. To do this an equivalent heat conductivity can be calculated. Using figure 2.2 we are calculating the heat flux through the *east* face of the control volume. First we calculate an interpolation factor f_e defined in terms of the distances from the nodes to the wall of the control volume.

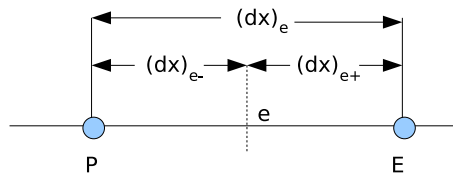


Figure 2.2: Distances associated with the control volume face

$$f_e = \frac{(dx)_{e+}}{(dx)_e} \quad (2.10)$$

Using this factor now we can calculate the thermal conductivity for the interface e

$$k_e = f_e k_p + (1 - f_e) k_E \quad (2.11)$$

2.1. A TWO-DIMENSIONAL TRANSIENT CONDUCTION PROBLEM

Leading to the discretized heat flux equation.

$$q_e = k_e \frac{(T_E - T_P)}{(dx)_e} \quad (2.12)$$

Equation 2.12 works as the heat flux through the east face of the control volume and as the west flux for the adjacent one.

In the following section a heat conduction problem is solved using the preceding discretization.

2.1 A Two-dimensional Transient Conduction Problem

A very long rod is composed of four different materials (M1 to M4), represented in figure 2.3 with different colours. The coordinates of the points p1 to p3 are given in table 2.1. The properties of the materials are given in table 2.2. Each of the four sides of the rod interact with the surrounding in a different manner, as described in table 2.3. The initial temperature field is $T = 11.00[^\circ C]$.

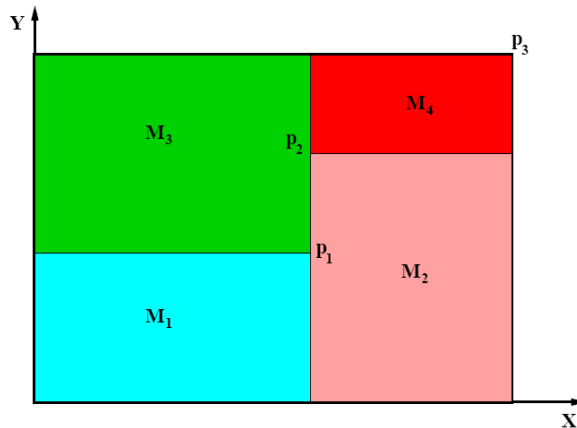


Figure 2.3: Schema of conduction problem

	$x[m]$	$y[m]$
p1	0.50	0.40
p2	0.50	0.80
p3	1.00	1.10

Table 2.1: Problem coordinates

The calculation is made for a transient development from time 0 to 10000 seconds. Solving for each instant of time using a time increment of 1 second.

	$\rho[\text{kg}/\text{m}^3]$	$c_p[\text{J}/\text{kgK}]$	$k[\text{W}/\text{mK}]$
M1	1600.00	770.00	210.00
M2	1900.00	810.00	200.00
M3	2700.00	980.00	190.00
M4	1500.00	860.00	170.00

Table 2.2: Physical properties

Wall	Boundary condition
Bottom	Isotherm at $T = 18.00[^\circ\text{C}]$
Top	Uniform $qflow = 64.00[\text{W}/\text{m}]$ length
Left	In contact with fluid at $T_g = 33.00[^\circ\text{C}]$ and heat transfer coefficient $8.00[\text{W}/\text{m}^2\text{K}]$
Right	Uniform temperature $T = 11.00 + 0.06 \cdot t[^\circ\text{C}]$ (where t is time in seconds)

Table 2.3: Boundary conditions

2.1.1 Discretization equations

For this problem the governing equation is

$$\rho c_p \frac{\partial T}{\partial t} = \nabla \cdot (k \nabla T) \quad (2.13)$$

where ρ is the density of the domain and c_p is the specific heat of the material.

Notice that the difference of equation 2.13 with 2.1 is that the source term is zero and there is heat accumulation due to the time dependent behavior.

For this problem the integration in a control volume must be made also in time. For the discretization in time the notation of $t + \Delta t$ will be used for the actual time and the t for the old time. According to this the temperature field will be calculated for the new instant using the *old* values of the preceding instant of time. The integration over a control volume will be made taking into account that inside a control volume the physical properties are taken as constant.

$$\rho c_p \int_V \int_t^{t+\Delta t} \frac{\partial T}{\partial t} dt dV = \int_t^{t+\Delta t} \int_V \nabla \cdot (k \nabla T) dV dt \quad (2.14)$$

The left side of equation 2.14 can be solved assuming that the temperature prevails throughout the hole control volume, leading to

$$\rho c_P \int_V \int_t^{t+\Delta t} \frac{\partial T}{\partial t} dt dV = \rho c_P V (T_P^1 - T_P^0) \quad (2.15)$$

Where V is the total volume of the control volume. The subindex P as explained in the previous section is the reference of the node studied. The superindex 1 and 0 are referred to the temperatures of the actual and the previous instant respectively.

2.1. A TWO-DIMENSIONAL TRANSIENT CONDUCTION PROBLEM

The spatial integration of the right side of equation was all ready performed in equation 2.4. Then for the time integration we have (*for space reasons the integration will be made as it was one-dimentional*)

$$\begin{aligned} \frac{\rho c_P V (T_P^{n+1} - T_P^n)}{dt} = & f \left[\frac{k_e (T_E^1 - T_P^1) dy}{(dx)_e} - \frac{k_w (T_P^1 - T_W^1) dy}{(dx)_w} + \right] \quad (2.16) \\ & + (1 - f) \left[\frac{k_e (T_E^0 - T_P^0) dy}{(dx)_e} - \frac{k_w (T_P^0 - T_W^0) dy}{(dx)_w} \right] \end{aligned}$$

where f is the weighting factor deffined as

- $f = 0$ for fully explicit, all the values are known from the previous time step (*old values*).
- $f = 1$ for fully implicit time integration, all the values are unknown and are solved for this time step.
- $f = 0.5$ for Crack-Nikolson time integration, the values of the previous and actual time are used.

To solve this particular problem we used the fully implicit because is unconditionally stable. Any time increment will reach convergence. That doesn't means that a physically realistic result will be reached.

Finally for the intern nodes the two-dimensional discretization equation is as follow

$$a_P T_P^1 = a_E T_E^1 + a_W T_W^1 + a_N T_N^1 + a_S T_S^1 + b_P \quad (2.17)$$

$$a_E = k_e \frac{dy}{(dx)_e} \quad (2.18)$$

$$a_W = k_w \frac{dy}{(dx)_w} \quad (2.19)$$

$$a_N = k_n \frac{dx}{(dy)_n} \quad (2.20)$$

$$a_S = k_s \frac{dx}{(dy)_s} \quad (2.21)$$

$$a_P^0 = \frac{\rho c_P dx dy}{dt} \quad (2.22)$$

$$a_P = a_E + a_W + a_N + a_S + a_P^0 \quad (2.23)$$

$$b_P = a_P^0 T_P^0 \quad (2.24)$$

The boundary conditions generate different discretized equations for the boundary nodes.

- + **Top:** according to the table 2.3 and as shown in figure 2.4 for the half control volume one of the faces has a constant flux of $64[W/m]$ while the others have related heat flows with the neighbors.

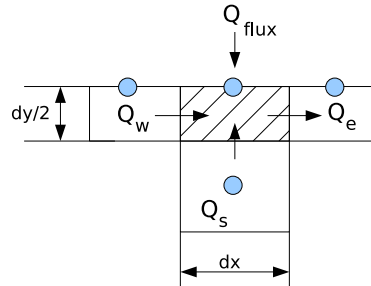


Figure 2.4: Prescribed heat flow boundary condition

The discretized equation has the same structure than equation 2.17 but the coefficients are defined as

$$a_E = k_e \frac{dy}{2(dx)_e} \quad (2.25)$$

$$a_W = k_w \frac{dy}{2(dx)_w} \quad (2.26)$$

$$a_N = 0.0 \quad (2.27)$$

$$a_S = k_s \frac{dx}{(dy)_s} \quad (2.28)$$

$$a_P^0 = \frac{\rho c_P dx dy}{2dt} \quad (2.29)$$

$$a_P = a_E + a_W + a_N + a_S + a_P^0 \quad (2.30)$$

$$b_P = a_P^0 T_P^0 + q_{flux} \cdot dx \quad (2.31)$$

Notice that the north coefficient is set to zero because there is no north node. And the extra 2 is related with the fact that is half of a normal control volume.

+ **Bottom:** The bottom boundary condition is constant temperature so the coefficients are

$$a_{nb} = 0.0 \quad (2.32)$$

$$a_P^0 = 0.0 \quad (2.33)$$

$$a_P = 1.0 \quad (2.34)$$

$$b_P = T_{bottom} \quad (2.35)$$

where the subindex nb means *neighbor*, that in this boundary condition don't make any influence over the node.

+ **Left:** according to table 2.3 the left side is in contact with a fluid of constant temperature T_g and heat transfer coefficient α that generates a

heat flux as shown in figure 2.5, this convective heat flux depends in this particular case only in the difference of temperature between the fluid surrounding the material and the temperature of the wall.

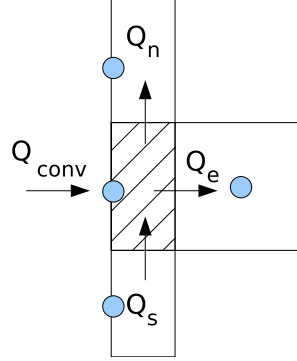


Figure 2.5: Convection boundary condition

The discretization for this boundary condition is the following

$$a_E = k_e \frac{dy}{(dx)_e} \quad (2.36)$$

$$a_W = 0.0 \quad (2.37)$$

$$a_N = k_s \frac{dx}{2(dy)_s} \quad (2.38)$$

$$a_S = k_s \frac{dx}{2(dy)_s} \quad (2.39)$$

$$a_P^0 = \frac{\rho c_P dx dy}{2dt} \quad (2.40)$$

$$a_P = a_E + a_W + a_N + a_S + a_P^0 + \alpha dy \quad (2.41)$$

$$b_P = a_P^0 T_P^0 + \alpha dy T_g \quad (2.42)$$

as in the top boundary condition the extra 2 in the *north* and *south* coefficients is due to the half control volume, and an extra term is added to the source.

- + **Right:** this boundary is very similar to the bottom one, but the main difference is that the source term depends on time.

$$a_{nb} = 0.0 \quad (2.43)$$

$$a_P^0 = 0.0 \quad (2.44)$$

$$a_P = 1.0 \quad (2.45)$$

$$b_P = 11.00 + 0.006 \cdot t \quad (2.46)$$

this means that the temperature of the wall varies as b_P because all the rest coefficients are set to zero

- + **Corners:** The corners can be associated to the closer boundary in the case of the bottom corners they can have the same discretization than the rest of the bottom nodes. The left-top corner can have the discretization of the left side. But the right-top corner has on one side constant heat flux and in the other side convection.

$$a_E = k_e \frac{dy}{2(dx)_e} \quad (2.47)$$

$$a_W = 0.0 \quad (2.48)$$

$$a_N = 0.0 \quad (2.49)$$

$$a_S = k_s \frac{dx}{2(dy)_s} \quad (2.50)$$

$$a_P^0 = \frac{\rho c_P dx dy}{4 dt} \quad (2.51)$$

$$a_P = a_E + a_W + a_N + a_S + a_P^0 + \alpha dy/2 \quad (2.52)$$

$$b_P = a_P^0 T_P^0 + \alpha \cdot dy \cdot T_g/2 + q_{flow} \cdot dx/2 \quad (2.53)$$

Notice that in this case the size of the control volume is the fourth part of a normal control volume.

2.1.2 Results

This problem was solved using the *TDMA* explained in section 1.2.2. For the transient solution a time step of 1 second was used and the mesh size was deffined according to the convergence of the solution of one point randomly chosen to a particular solution as the mesh size enlarge. This can be seen from figure 2.6 that for a number greater that 40000 the result changes in a very small proportion in contrast with the computing time. Ergo for this case a mesh of 48400 was used.

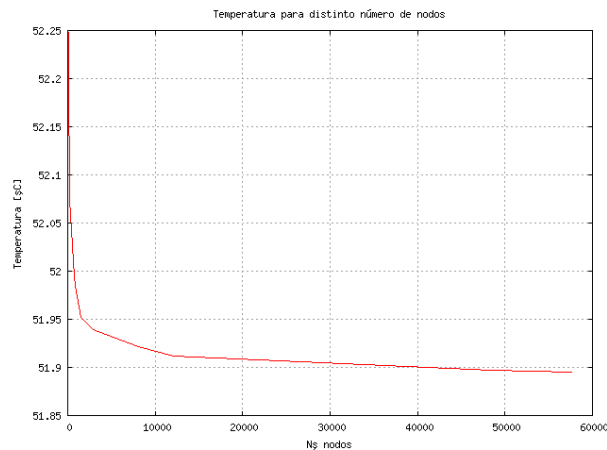


Figure 2.6: Evolution of the solution for different mesh sizes

2.1. A TWO-DIMENSIONAL TRANSIENT CONDUCTION PROBLEM

Figure 2.7 shows the temperature field in the solid once the time has reached the 5000[s]. There you can see that in the bottom a temperature of 18°C was maintained and that in the left side the temperature has a valued close to 40°C according to the evolution of this boundary condition.

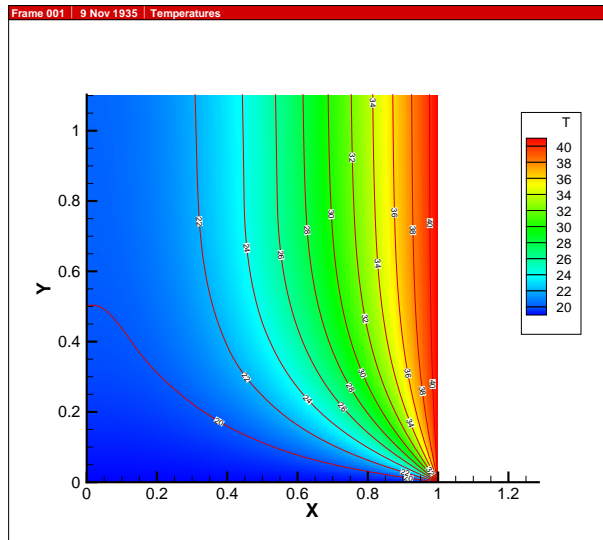


Figure 2.7: Temperature field at $t = 5000[\text{s}]$

Notice that this material will never reach a stationary state because on the left side the temperature changes on time, and then the temperature will always increase.

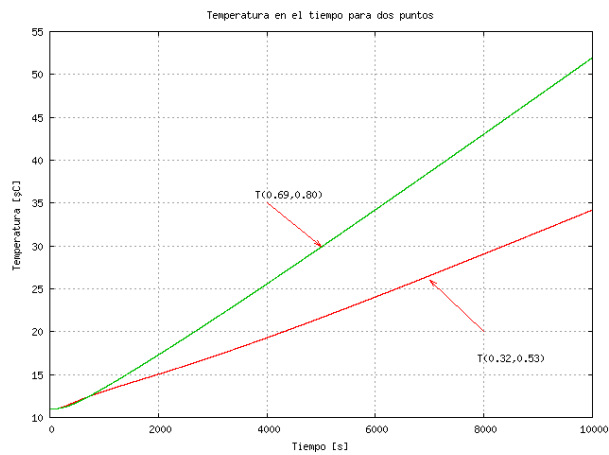


Figure 2.8: Temperature in time for two different points on the domain

Figure 2.8 shows the temperature in time for two points. There can be seen the the temperature is actually increasing in sustained way, but with a bigger value for the point that is closer to the left side of the domain.

Chapter 3

Convection-Diffusion

In problems where there is fluid motion a convective effect takes place. In fact the convection is the transport of a certain propertie by fluid motion.

From the transport equation for a general propertie ϕ and deleting the transient term we have the general convection-diffusion equation for steady state.

$$\nabla(\rho\mathbf{u}\phi) = \nabla \cdot (\Gamma\nabla\phi) + S_\phi \quad (3.1)$$

Where \mathbf{u} is the velocity vector and Γ is the diffusive coefficient and ρ is the density.

To make the integral over the control volume we need to use of the *Gauss divergence theorem*

$$\int_{CV} \text{div}(\mathbf{a})dV = \int_S \mathbf{n} \cdot \mathbf{a}dS \quad (3.2)$$

in words. The integral in the region inside a surface of the divergence of a vector field is equal to the net flux of the vector through the surface of the region.

Finally,

$$\int_S \mathbf{n} \cdot (\rho\mathbf{u}\phi)dS = \int_S \mathbf{n} \cdot (\Gamma\nabla\phi)dS + \int_{CV} S_\phi dV \quad (3.3)$$

The left side of this equation gives the convective term, while the right side is the diffusive and source term.

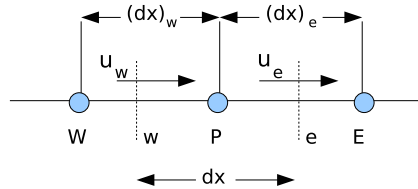


Figure 3.1: Convection-diffusion control volume

Now performing the integration over a control volume in figure 3.1. Assuming no sources and one-dimensional we have

$$(\rho u \phi)_e - (\rho u \phi)_w = \left(\Gamma \frac{d\phi}{dx} \right)_e - \left(\Gamma \frac{d\phi}{dx} \right)_w \quad (3.4)$$

from chapter 2 we know how to represent the integral of the right side of equation 3.4. For the convection term or left side of equation 3.4 a piecewise-linear profile is not always the best choice because it can lead to non-positive coefficients in the discretization equation.

To make the discretization of equation 3.4 we will define two variables F and D as the convective mass flux and the diffusion conductance respectively.

$$F_e = (\rho u)_e, \quad F_w = (\rho u)_w \quad (3.5)$$

$$D_e = \frac{\Gamma_e}{(dx)_e}, \quad D_w = \frac{\Gamma_w}{(dx)_w} \quad (3.6)$$

Now, using this new variables is important to make use of numerical schemes for the convective term, next a few are shown.

a *Upwind*: This scheme is based on the idea that the fluid flowing into the control volume brings the information of the control volume where it came and will not know anything about the control volume towards which it is heading. In other words, it uses the upwind value of ϕ . For example in the east side face we have

$$\begin{aligned} * F_e > 0 \text{ then } \phi_e &= \phi_P \\ * F_e < 0 \text{ then } \phi_e &= \phi_E \end{aligned}$$

b *Central Difference*: This scheme makes a piecewise-linear profile to approximation the value of ϕ in the face of the control volume.

$$\phi_e = \frac{\phi_E + \phi_P}{2} \quad (3.7)$$

There are more numerical schemes for solving the convection approximation, but to explain all escapes from the slop of this work, for more information please see bibliography [1].

Now we rewrite the discretized equation for convection-diffusion

$$a_P \phi_P = a_W \phi_W + a_E \phi_E + a_S \phi_S + a_N \phi_N + b_P \quad (3.8)$$

where the coefficients are

$$\begin{aligned} a_E &= D_e A(|P_e|) + [|-F_e, 0|] \\ a_W &= D_w A(|P_w|) + [|F_w, 0|] \\ a_N &= D_n A(|P_n|) + [|-F_n, 0|] \\ a_S &= D_s A(|P_s|) + [|F_s, 0|] \\ a_P &= a_W + a_E + a_S + a_N \\ b_P &= S_C dx dy \end{aligned}$$

3.1. A TWO-DIMENSIONAL STEADY CONVECTION-DIFFUSION
EQUATION: THE SMITH HUTTON PROBLEM

Where the symbol $[[A, B]]$ means the greater of A and B , and S_C its the source term. Term $A(|P|)$ is a value function of the *Peclet* number ($P = F/D$) which depends on the numerical scheme used for the convection term of the discretization equation.

The table 3.1 shows five numerical schemes used to approximate the convection term.

Scheme	Formula for $A(P)$
Central Difference	$1 - 0.5 P $
Upwind	1
Hybrid	$[[0, 1 - 0.5 P]]$
Power Law	$[[0, (1 - 0.1 P)^5]]$
Exponential	$ P /(exp(P) - 1)$

Table 3.1: The Function $A(|P|)$ for different schemes

The value of the *Peclet* number indicates the relative strengths of convection and diffusion, in other words it indicates when the problem is mainly diffusive or when its more convective. The figure 3.2 is a clear image of the relation between the *Peclet* number and the propriety exchange. As it can be seen for *Peclet* numbers close to zero the problem is purely diffusive, in other words it would be like a problem without mass flow. On the contrary when the *Peclet* number is big as tending to infinity, the problem becomes a purely convective, in those cases the transfer of a particular propertie is mainly because the mass flow and it doesn't gives time to the diffusion to make any influence.

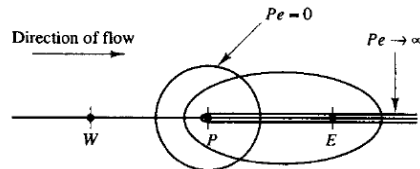


Figure 3.2: Effect of the *Peclet* number on convection-diffusion problems

In other words

- No convection and pure diffusion $P \rightarrow 0$
- No diffusion and pure convection $P \rightarrow \infty$

3.1 A Two-Dimensional Steady Convection-Diffusion Equation: The Smith Hutton Problem

The Smith and Hutton problem is a *test problem* with a strong streamline curvature, which tries to evaluate the numerical model used for the interpolation of the convection term in the energy equation. The problem is as fallow:

A two dimensional steady state solution of the equation 3.1 in a rectangular domain with a prescribed velocity. A schematic diagram is given in figure 3.3.

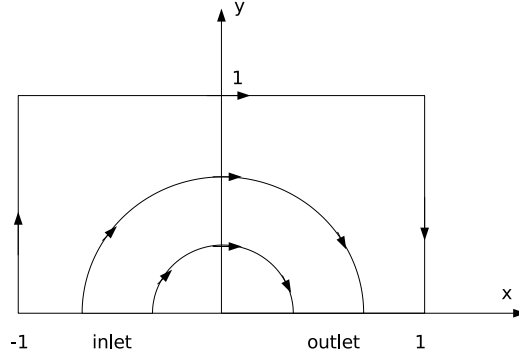


Figure 3.3: Geometry specifications and streamlines

$$u(x, y) = 2y(1 - x^2) \quad (3.9)$$

$$v(x, y) = -2x(1 - y^2) \quad (3.10)$$

and the following boundary condition for the variable ϕ

$$\begin{aligned} \phi &= 1 + \tanh(\alpha(2x + 1)) & y = 0 ; x \in (-1, 0) & \quad (\text{inlet}) \\ \frac{\partial \phi}{\partial y} &= 0 & y = 0 ; x \in (0, 1) & \quad (\text{outlet}) \\ \phi &= 1 - \tanh(\alpha) & & \quad (\text{elsewhere}) \end{aligned}$$

where $\alpha = 10$.

This simulation describe the numerical behavior of different numerical schemes (such as CD, Upwind, Hybrid...) and evaluate the results with different mesh size with the purpose of finding an appropriate one to obtain a good approximation of the correct solution.

This problem is treated as two-dimensional with all properties remaining constant and without sources. It's asked to solve the problem for a range of *Peclet* numbers: 10, 10^3 , 10^6 . The *Peclet* number represents the ratio of strengths of convection and diffusion, this implicates that for small values of P the convection its almost absent and the piecewise-linear profile approximation is appropriate but for large values the convection is present and it's necessary the use of a numerical scheme that can interpolate best the variable studied.

3.1.1 Discretization equations

Given the nature of the problem, it was chosen a mesh that enable us to make use of a determinate numerical scheme, that why the boundary was not a node

3.1. A TWO-DIMENSIONAL STEADY CONVECTION-DIFFUSION EQUATION: THE SMITH HUTTON PROBLEM

but one side of the control volume next to the contour of the domain. This discretization is shown in figure 3.4 where the internal nodes are shown with a filled dot and the boundary ones are with a ring, those last ones aren't really nodes but essentially are used to fulfill the standard numerical equation described in the subsequent section. An internal node is remarked and show us that the velocity is not on the main grid but in a secondary one that locates the values of the velocity in the corresponding wall of the control volume.

For this particular problem it was chosen to use a regular mesh with square elements of size $dx \cdot dy$.

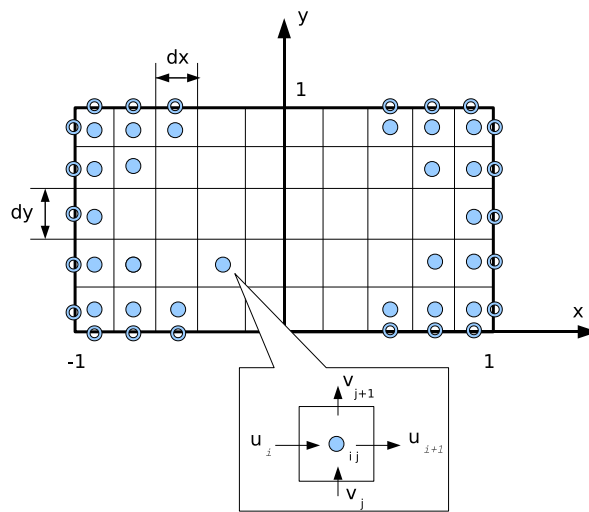


Figure 3.4: Discretization of the domain

The discretization equation was shown previously in equation 3.8

3.1.2 Discussion of results

After the results are achieved it's important to find a good mesh that give us a good approximation without consuming a large amount of time or computational effort. For that reason the program was run with different meshes but everyone using the upwind scheme showed at the beginning of this chapter and with the same solver, so the only difference is the mesh size. The results are in figure 4.2 where it can be seen that for a small number of elements the solution is far from the expected one, but while the number of elements is increased the solution of one and another are closer. So in this case we make use of a 200×100 mesh because if the size of the element are smaller the difference between this size and one much smaller are not significant.

The results achieved with the mesh selected and using Upwind scheme at the outlet are listed in table 3.2 where the column headed with *Expected* are the best results for this problem achieved by somebody else, and the column

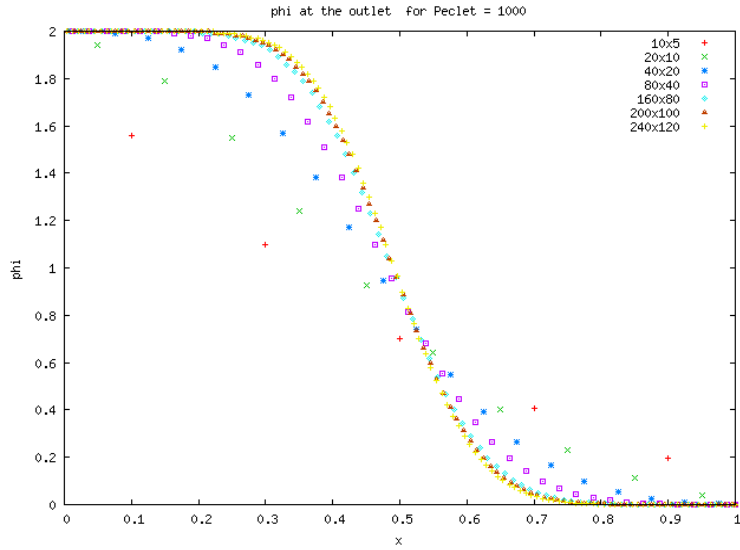


Figure 3.5: Outlet ϕ for different mesh size using Upwind and $P = 1000$

headed with *Achieved* are the values achieved by me.

$x - position$	$\rho/\Gamma = 10$		$\rho/\Gamma = 10^3$		$\rho/\Gamma = 10^6$	
	<i>Expected</i>	<i>Achieved</i>	<i>Expected</i>	<i>Achieved</i>	<i>Expected</i>	<i>Achieved</i>
0.0	1.989	1.9081	2.0000	2.0000	2.000	2.000
0.1	1.402	1.3877	1.9990	1.9999	2.000	2.000
0.2	1.146	1.1362	1.9997	1.9969	2.000	1.999
0.3	0.946	0.9388	1.9850	1.9442	1.999	1.982
0.4	0.775	0.7691	1.8410	1.6277	1.964	1.734
0.5	0.621	0.6169	0.9510	0.9248	1.000	0.937
0.6	0.480	0.4775	0.1540	0.2907	0.036	0.222
0.7	0.349	0.3481	0.0010	0.0446	0.000	0.019
0.8	0.227	0.2268	0.0000	0.0030	0.000	0.000
0.9	0.111	0.1117	0.0000	0.0000	0.000	0.000
1.0	0.000	0.0000	0.0000	0.0000	0.000	0.000

Table 3.2: Numerical results at the outlet for different ρ/Γ numbers

The figure 3.6 is a plot of the table 3.2 where the difference between the expected values and the ones achieved are best noted.

The figure 3.7 shows the velocity profile in the domain and figures 3.8, 3.9 and 3.10 shows the values of ϕ at the hole domain for *Peclet* number 10, 10^3 and 10^6 respectively. It can be seen that for small values of *Peclet* the solution is far from being the same that at the inlet, according to the diffusion strength, but as the value of P increase the solution becomes closer to the inlet and then more convective.

3.1. A TWO-DIMENSIONAL STEADY CONVECTION-DIFFUSION EQUATION: THE SMITH HUTTON PROBLEM

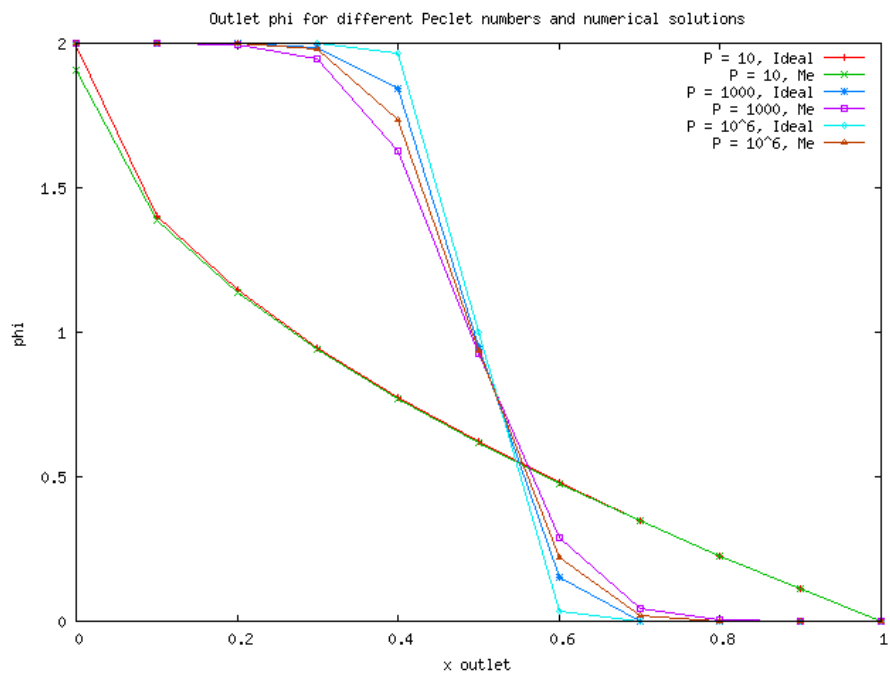


Figure 3.6: Results for the outlet ϕ of the ideal numerical result and the one achieved in this work

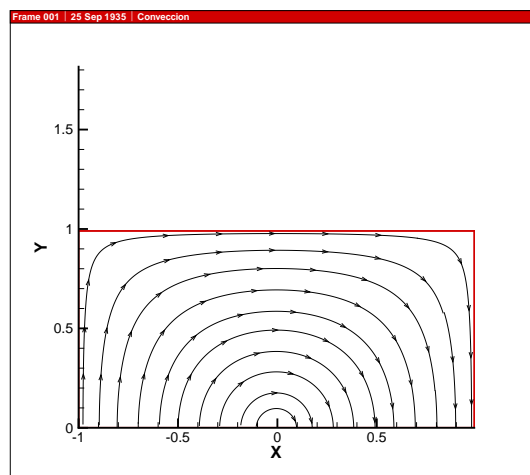


Figure 3.7: Velocity field

The different solutions achieved with the different numerical schemes are shown in figures 3.11 and 3.12 for *Peclet* numbers 10 and 1000 respectively. It can be seen that for the case of a small *Peclet* number the solution achieved using the central difference scheme is the same that with any other numerical

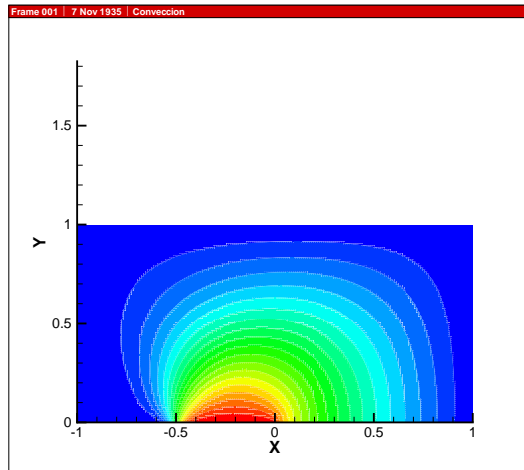


Figure 3.8: ϕ in the domain for $Peclet = 10$

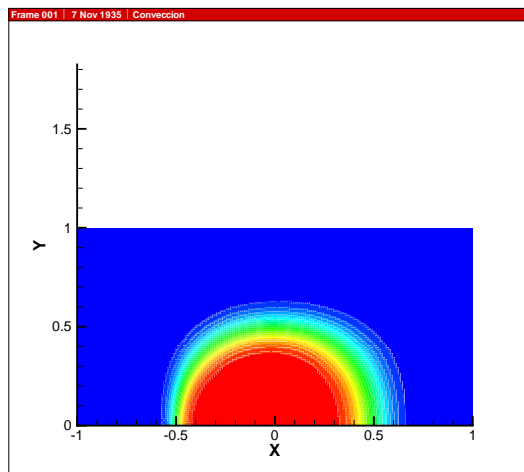


Figure 3.9: ϕ in the domain for $Peclet = 1000$

3.1. A TWO-DIMENSIONAL STEADY CONVECTION-DIFFUSION EQUATION: THE SMITH HUTTON PROBLEM

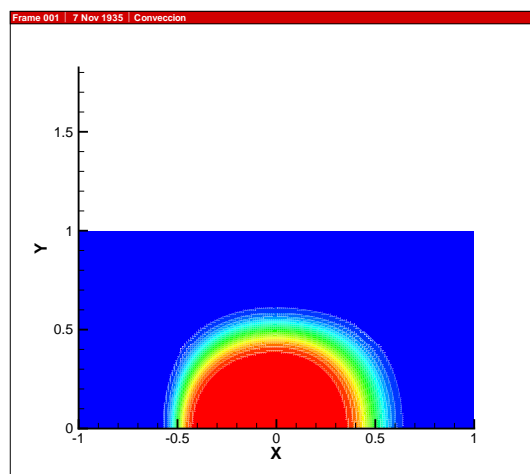


Figure 3.10: ϕ in the domain for $Peclet = 10^6$

scheme, that is because as we said before, a small value of peclet shows a strong diffusion so in that case a piecewise-linear profile in the interpolation of ϕ is good enough. But in case of a large values of peclet number the convection plays an important role so the central difference scheme don't work and give wrong values and for the exponential scheme a large values of $Peclet$ doesn't give right answers.

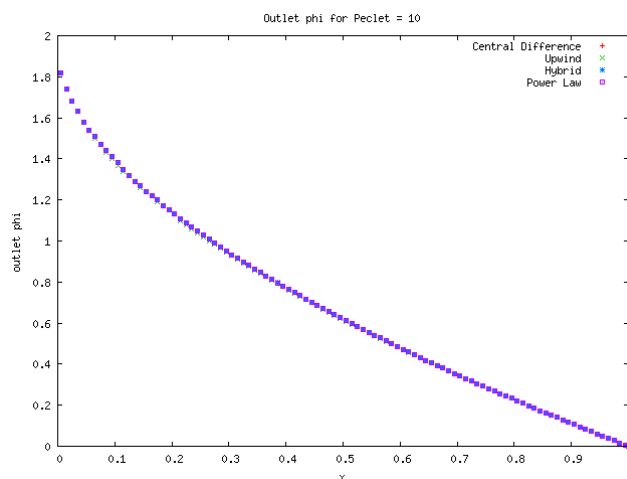


Figure 3.11: ϕ at the output for $Peclet = 10$ and different numerical schemes

For a sufficient large $Peclet$ number the effect of the diffusion will tend to zero while the main effect on transport of a determinated propertie ϕ is the convection. Never the less it can be seen in figure 3.13 that the comparation between the solution without diffusion and the one achieved using the upwind numerical scheme is that a false diffusion takes place. This false diffusion is the

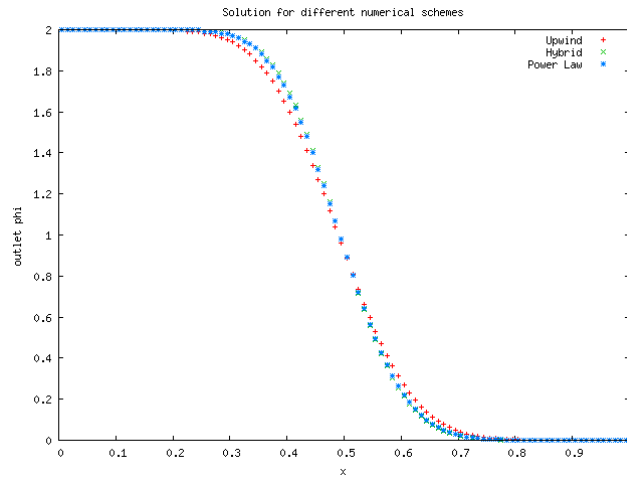


Figure 3.12: ϕ at the output for $Peclet = 10^3$ and different numerical schemes

consequence of the inclination of a diagonal incidence of the velocity with the wall if the control volume. The effect of a false diffusion would be eradicated in the case that the velocity is perpendicular to the wall, if not the value of the cell face will be calculated using the neighbors in a way that the properties calculated will start to spread in the surrounding even if there is no diffusion.

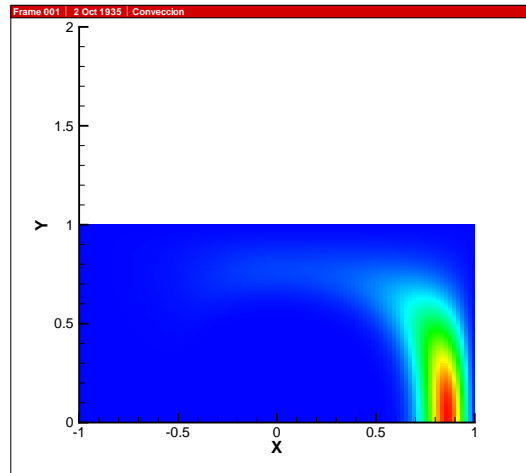


Figure 3.13: False diffusion

The figures 3.14, 3.15 and 3.16 show a three dimensional view of ϕ there we see the difference at the outlet while Peclet number changes.

3.1. A TWO-DIMENSIONAL STEADY CONVECTION-DIFFUSION EQUATION: THE SMITH HUTTON PROBLEM

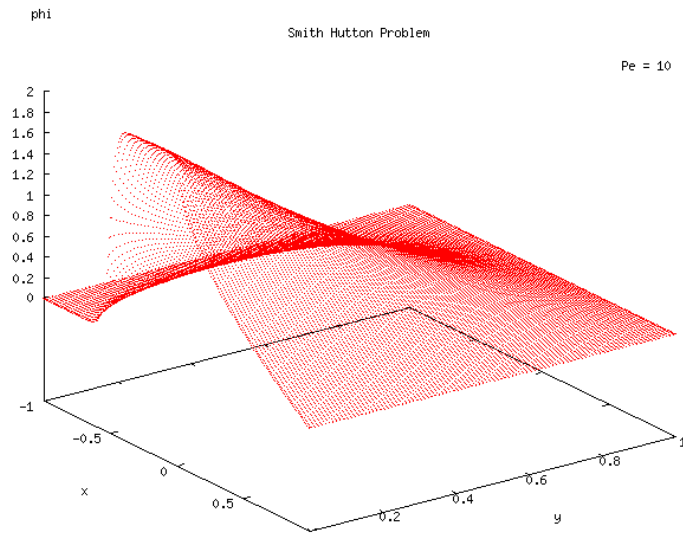


Figure 3.14: 3D ϕ for *Peclet* = 10

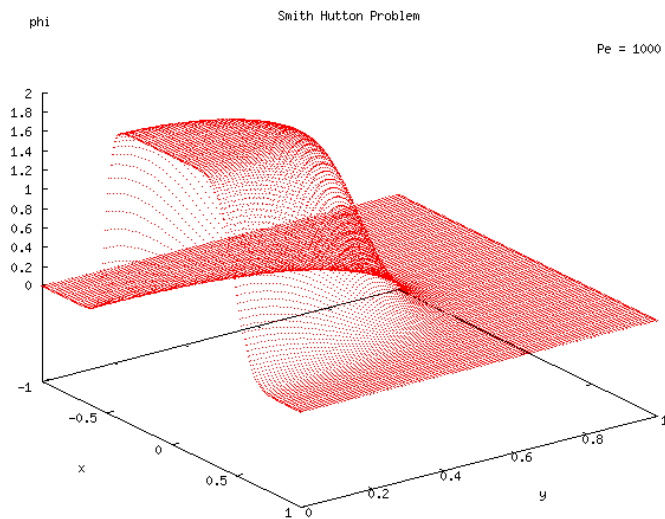


Figure 3.15: 3D ϕ for *Peclet* = 1000

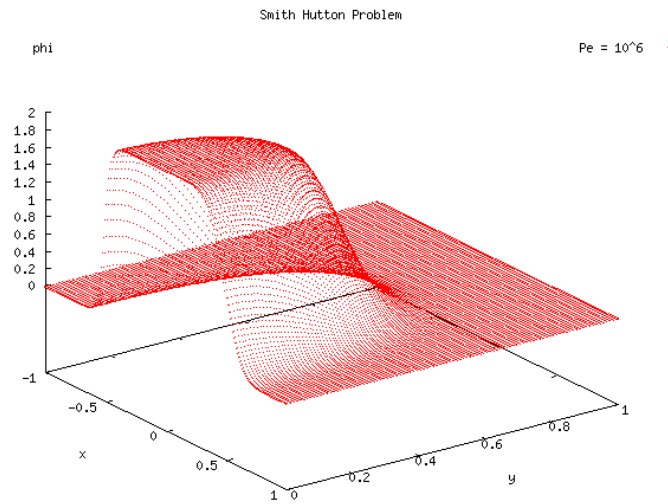


Figure 3.16: 3D ϕ for $Peclet = 10^6$

Chapter 4

Navier-Stokes equations

The Navier-Stokes equations describes the motion of fluids and the physical behavior of any propertie in it. The equation 4.1 represents the governing equation for the transport of the propertie ϕ in a certain fluid.

$$\frac{\partial \rho \phi}{\partial t} + \nabla \cdot (\rho \mathbf{u} \phi) = \nabla \cdot (\Gamma \nabla \phi) + S \quad (4.1)$$

where ρ is the density and Γ is the diffusion coefficient.

For the momentum equations, the propertie ϕ can be changed for the velocity in x or y in order to get the respective coordinate momentum equation. The diffusion coefficient Γ is changed for the viscosity μ and finely the source is replaced with the momentum source, in other words with the presure gradient as follow

x-momentum equation

$$\frac{\partial}{\partial x}(\rho u u) + \frac{\partial}{\partial y}(\rho v u) = \frac{\partial}{\partial x} \left(\mu \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left(\mu \frac{\partial u}{\partial y} \right) - \frac{\partial p}{\partial x} + S_u \quad (4.2)$$

y-momentum equation

$$\frac{\partial}{\partial x}(\rho u v) + \frac{\partial}{\partial y}(\rho v v) = \frac{\partial}{\partial x} \left(\mu \frac{\partial v}{\partial x} \right) + \frac{\partial}{\partial y} \left(\mu \frac{\partial v}{\partial y} \right) - \frac{\partial p}{\partial y} + S_v \quad (4.3)$$

Continuity equation

$$\frac{\partial}{\partial x}(\rho u) + \frac{\partial}{\partial y}(\rho v) = 0 \quad (4.4)$$

Nevertheless having the proper equation is not enough. As you can see there are no transport equation for the pressure, in other words the momentum equations can't be solved without any given or somehow estimated pressure field. And if the fluid is incompressible the pressure can't be obtained as a function of the density and temperature. This means that to in order to solve the velocity, the pressure field is needed and to solve the pressure the velocity field is needed. This is what we called the *pressure-velocity coupling*. To solve this coupling two different approaches are made in this work. First the SIMPLE algorithm

that somehow uses a guessed pressure and velocity fields and correct them in an iterative fashion in order to obtain a result realistic enough. The second one is the Fractional Step, that uses a mathematical approach to solve the velocity field in every time step.

In the next section a problem will be solved using both methods to compare them.

4.1 Driven Cavity problem

The driven cavity problem is defined as shown in figure 4.1, it consist in a square cavity filled with an incompressive fluid width all physical properties remaining constant. This problem consist in studying the vortex generation inside the cavity when the upper layer of fluid moves at u -Velocity of 1 while the remaining walls remain at null velocity (zero slip condition). This condition influence the movement inside the cavity that depending on the *Reynolds* number the size and quantity of vortex changes. In this work the simulations are made whit Reynolds numbers of 100, 400, 1000, 3200, 5000, 7500 and 10000.

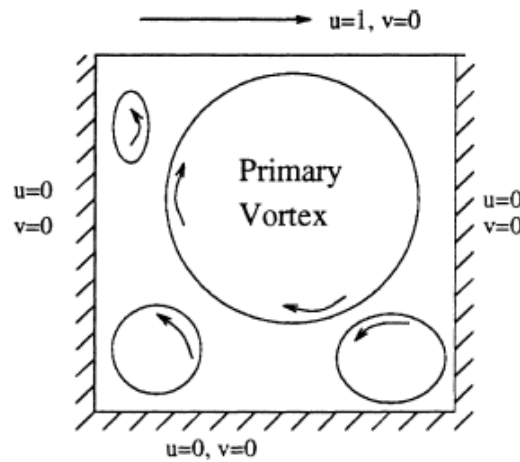


Figure 4.1: Driven Cavity problem.

where Re is the dimensionless Reynolds number defined as

$$Re = \frac{\rho V_0 L}{\mu} \quad (4.5)$$

where ρ and μ are the density and the viscosity of the working fluid respectively. L and V_0 are the characteristic length and velocity respectively.

According to the definition of the Reynolds number in equation 4.5, L correspond to the width of the cavity 1 and μ is fixed as 1 as well, so in order to change the Reynolds number, the density ρ was taking the values given previously, leaving the velocity at the boundary fixed as shown in figure 4.1.

4.1.1 Discretization

The mesh used was an structural one of $N \times M$ control volumes for the variables p and to compute u and v it was used a staggered grid in the x and y direction respectively as shown in figure 4.3.

The mesh generation was made using equation 4.6, so the region closer to the wall can have smaller control than the region at the center of the cavity, in fact the most interesting behavior occurs close to the walls where the vortex are formed and ergo a refined mesh its needed to compute the phenomenon as real as it can due to approximation used in the algorithm. In figure 4.2 a mesh of 129×129 its shown, notice that at the walls the mesh gets thinner as explained before.

$$x_{vc}(i) = \delta + \frac{\Delta\delta}{2} \left[1 + \tanh \left(2k_x \frac{i-1}{n} \right) / \tanh(k_x) \right] \quad (4.6)$$

Where x_{cv} is the cartesian coordinate of the i^{th} control volume line in the x -direction; $\Delta\delta$ is the zone length; k is the grid concentration factor that in this case takes the value of 4 and in this case $n = N$ for symmetrical concentration.

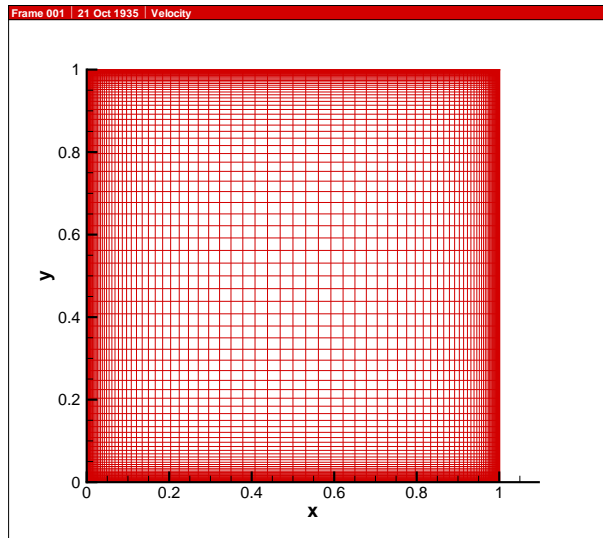


Figure 4.2: Structural mesh used in the simulation

Figure 4.3 shows the main grid used to calculate the pressure variable p . Notice that for the velocity components u and v a staggered in x and y was

used.

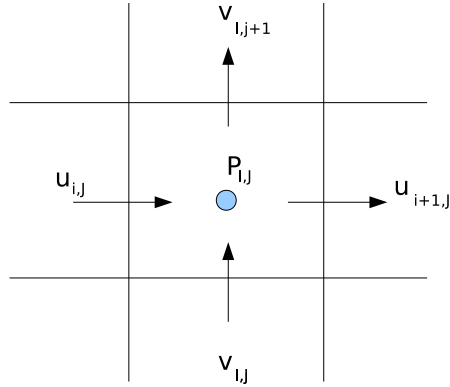


Figure 4.3: Control volume for pressure variable

§ The SIMPLE algorithm

SIMPLE stands for Semi-Implicit Method for Pressure-Linked Equations. It work as a guess-and-correct procedure for calculation of pressure.

For start a presure field must be guessed, then the momentum equations are solved using this guessed pressure field p^* obtaining the guessed velocity field. Calculating u^* and v^* guessed velocity components as follow.

$$a_{i,j}u_{i,j}^* = \sum a_{nb}u_{nb}^* + (p_{I-1,J}^* - p_{I,J}^*)A_{i,j} + b_{i,j} \quad (4.7)$$

$$a_{I,j}v_{I,j}^* = \sum a_{nb}v_{nb}^* + (p_{I,J-1}^* - p_{I,J}^*)A_{I,j} + b_{I,j} \quad (4.8)$$

Notice that the subindexes i and j are refereed to the staggered grid for x and y respectively, and the subindexes I and J are refereed to the main grid as shown in figure 4.3.

Where:

$$a_{i,j} = \sum a_{nb} + a_{i,j}^0$$

$$a_{I,j} = \sum a_{nb} + a_{I,j}^0$$

The neighbor coefficients nb are calculated as follow:

$$\begin{aligned}
a_E &= D_e A(|P_e|) + [| - F_e, 0|] \\
a_W &= D_w A(|P_w|) + [| F_w, 0|] \\
a_N &= D_n A(|P_n|) + [| - F_n, 0|] \\
a_S &= D_s A(|P_s|) + [| F_s, 0|] \\
a^0 &= \frac{\rho \Delta x \Delta y}{\Delta t}
\end{aligned}$$

where $[|A, B|]$ means the bigger of A and B, and the flow rates F and conductances D are defined as

$$\begin{aligned}
D_e &= \mu \Delta y / (\delta x)_e & F_e &= (\rho u)_e \Delta y \\
D_w &= \mu \Delta y / (\delta x)_w & F_w &= (\rho u)_w \Delta y \\
D_n &= \mu \Delta x / (\delta y)_n & F_n &= (\rho v)_n \Delta x \\
D_s &= \mu \Delta x / (\delta y)_s & F_s &= (\rho v)_s \Delta x
\end{aligned}$$

and $P_f = F_f / D_f$. In this work we compare the results using different numerical schemes to approximate the convection term of the governing equations. This means in terms of the equation that $A(|P|)$ takes the values of table 3.1

After the guessed velocity field is solved, pressure field must be corrected using the initial guessed pressure field and a corrector field p' :

$$p = p^* + p' \quad (4.9)$$

The derivation of the equation for p' will not be described in this work, but following the same idea used for correct the pressure field in equation 4.9 and deleting the neighbor guess velocity as an approximation one can get to the following equations.

$$u_{i,J} = u_{i,J}^* + d_{i,J}(p'_{I-1,J} - p'_{I,J}) \quad (4.10)$$

$$v_{I,j} = v_{I,j}^* + d_{I,j}(p'_{I-1,J} - p'_{I,J}) \quad (4.11)$$

where $d_{i,J} = \frac{A_{i,J}}{a_{i,J}}$ and $d_{I,j} = \frac{A_{I,j}}{a_{I,j}}$

$$a_{I,J} p'_{I,J} = a_{I+1,J} p'_{I+1,J} + a_{I-1,J} p'_{I-1,J} + a_{I,J+1} p'_{I,J+1} + a_{I,J-1} p'_{I,J-1} + b'_{I,J} \quad (4.12)$$

where $a_{I,J} = a_{I+1,J} + a_{I-1,J} + a_{I,J+1} + a_{I,J-1}$ and the coefficients are as follow:

$$\begin{aligned}
 a_{I+1,J} &= (\rho dA)_{i+1,J} \\
 a_{I-1,J} &= (\rho dA)_{i,J} \\
 a_{I,J+1} &= (\rho dA)_{I,j} \\
 a_{I,J-1} &= (\rho dA)_{I,j} \\
 b'_{I,J} &= (\rho u^* A)_{i,J} - (\rho u^* A)_{i+1,J} + (\rho v^* A)_{I,j} - (\rho v^* A)_{I,j+1}
 \end{aligned}$$

Using this equation the pressure correction its calculated and consequently the velocity components are corrected. Notice that the source term ($b'_{I,J}$) in equation 4.12 correspond to the continuity imbalance arising from the incorrect velocity field, so in the converged solution this term should tend to zero. Figure 4.4 shows the algorithm used in this work.

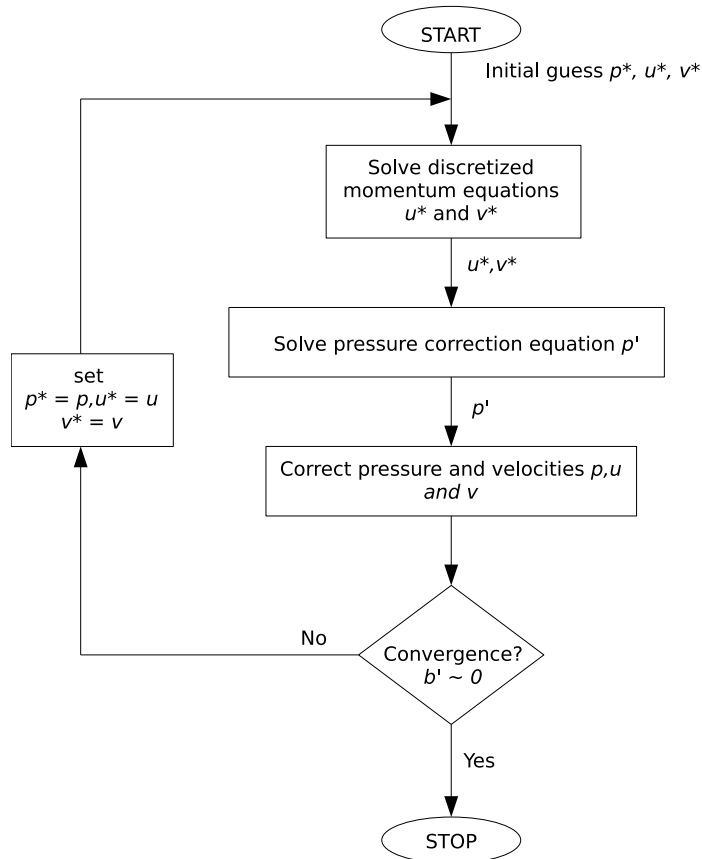


Figure 4.4: SIMPLE algorithm

Notice that this algorithm can be performed as a semi-transient or fully transient. The main difference is that when the convergence is reached you can go on with the time step for a fully transient, but on the other hand you can update the velocity in the precedent instant with the value calculated in the iteration before, that way you can get a semi-transient

behavior. But in that case you only get a solution at the end of the calculation for the stationary state.

§ The Fractional Step Method

For this formulation we consider the dimensionless Navier-Stokes equation for incompressible Newtonian fluids.

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = \frac{1}{Re} \Delta \mathbf{u} - \nabla p \quad (4.13)$$

$$\nabla \cdot \mathbf{u} = 0 \quad (4.14)$$

Then we can put together the convection and diffusion terms

$$\mathbf{R}(\mathbf{u}) \equiv -(\mathbf{u} \cdot \nabla) \mathbf{u} + \frac{1}{Re} \Delta \mathbf{u} \quad (4.15)$$

Then we can rewrite equation 4.13 in a reduced expression

$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{R}(\mathbf{u}) - \nabla p \quad (4.16)$$

then using central difference for the time derivative term we have

$$\left. \frac{\partial \mathbf{u}}{\partial t} \right|^{n+1/2} \approx \frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} + O(\Delta t^2) \quad (4.17)$$

and for $\mathbf{R}(\mathbf{u})$ we use a fully explicit second order Adams-Bashforth scheme

$$\mathbf{R}^{n+1/2}(\mathbf{u}) \approx \frac{3}{2} \mathbf{R}(\mathbf{u}^n) - \frac{1}{2} \mathbf{R}(\mathbf{u}^{n-1}) + O(\Delta t^2, \Delta x^m) \quad (4.18)$$

and for the pressure gradient a first-order backward Euler scheme. Leading to the semi-discretized Navier-Stokes equations

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = \frac{3}{2} \mathbf{R}(\mathbf{u}^n) - \frac{1}{2} \mathbf{R}(\mathbf{u}^{n-1}) - \nabla p^{n+1} \quad (4.19)$$

$$\nabla \cdot \mathbf{u}^{n+1} = 0 \quad (4.20)$$

Now as in the previous section, the problem is to solve the pressure-velocity coupling. To do that we use the fractional step method. In which we solve a time increment velocity field without regard of its incompressibility and then the pressure gradient force the incompressibility solving finely the velocity field. This procedure is derived from the Helmholtz-Hodge vector decomposition theorem¹. Here we have the velocity predictor u^p that can be uniquely decomposed into a divergence-free vector, u^{n+1} , and the gradient of a scalar field, $\nabla \tilde{p}$. Leading to equation 4.21

¹The deduction of this theorem goes out of the range of this work

$$u^p = u^{n+1} + \nabla \tilde{p} \quad (4.21)$$

the velocity predictor can be calculated as

$$u^p = u^n + \Delta t \left(\frac{3}{2} \mathbf{R}(u^n) - \frac{1}{2} \mathbf{R}(u^{n-1}) \right) \quad (4.22)$$

Where taking the divergence of equation 4.21, the term of u^{n+1} turns into zero due to its definition and we get the Poisson equation for the pseudo-pressure

$$\Delta \tilde{p} = \nabla \cdot u^p \quad (4.23)$$

where the pseudo-pressure its deffined in equation 4.24

$$\tilde{p} = \Delta t \cdot p^{n+1} \quad (4.24)$$

Once the solution is obtained, u^{n+1} can be calculated from

$$u^{n+1} = u^p - \nabla \tilde{p} \quad (4.25)$$

As a summary the fractional step algorithm is shown in figure 4.5. Where you can see that it is not a iterative procedure, in other words for each time instant it reach the solution at once, in contrast with the SIMPLE algorithm that has to make a loop until convergence is reach due to its guess-and-correct essence. For this reason one could say that for transient problems the fractional step method is better.

The final time increment is reached when a steady state is reached somehow comparing the velocity in the preceding instant with the actual and if the difference is not important, the simulation stops.

4.1.2 Discussion of results

§ SIMPLE algorithm

To begin with the results analysis we first choose the best mesh, for that reason the simulation was made using different number of control volumes. In figure 4.6 the results of u and v are plotted for different mesh sizes, starting from a 10x10 mesh until a 200x200, between this first one and the 129x129 mesh the differences are notorious. But the results achieved using a refined mesh are not significant on the other way the computing time gets bigger and makes the simulation slow and unavailable, for that reason the 129x129 mesh its used.

After the mesh size is selected, a comparison between the benchmark solution and the one reached in this work is made. For the sake of clarity instead of a data table a plot of the relative error was made for the both components of velocity u and v using only central difference and upwind scheme for numerical simulation as explained in ???. In figure 4.7 can

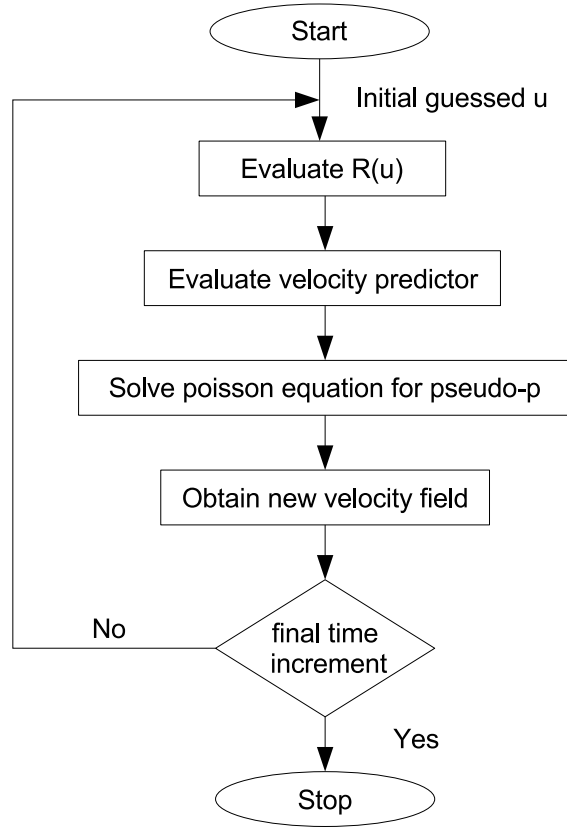


Figure 4.5: Fractional step method algorithm

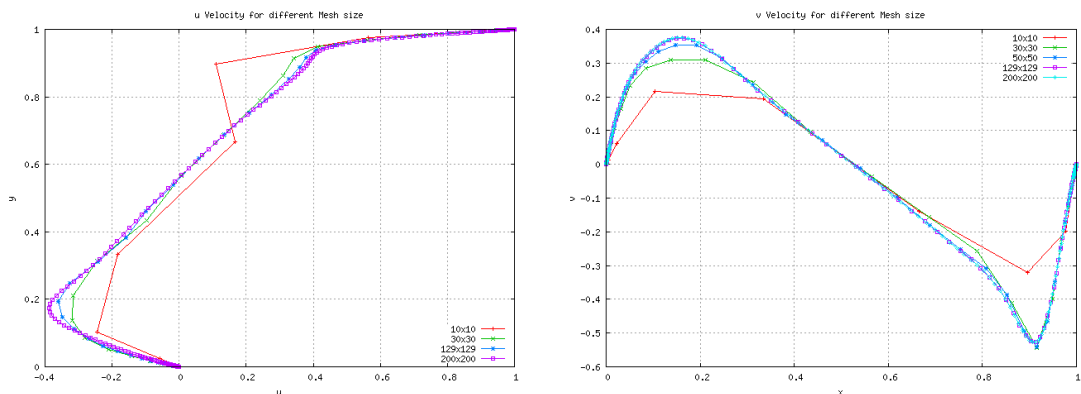


Figure 4.6: u and v results for different mesh sizes

be seen that for small Reynolds the error is less than 5% for the central difference approach being considerably smaller than the results achieved with upwind scheme. For larger Reynolds there are some difficulties at the center of the cavity making the error bigger but anyway smaller than with upwind, for this reason the results shown in figures 4.10 are using central difference scheme.

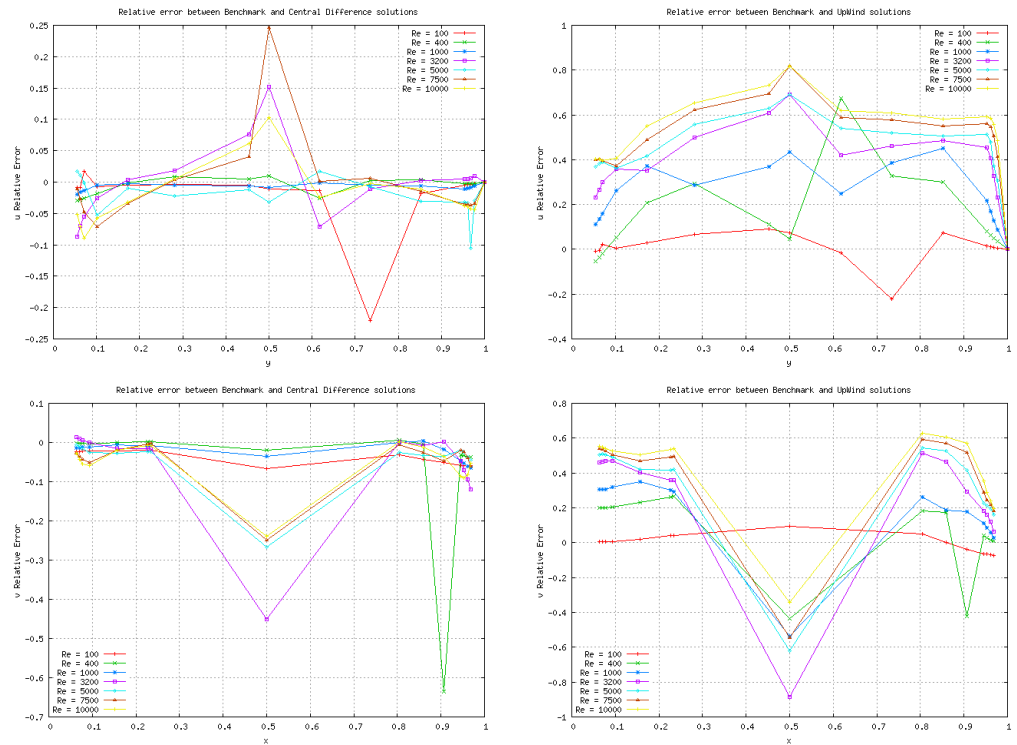


Figure 4.7: Error between Benchmark and the result achieved using central difference and upwind schemes for u and v solutions

Figures 4.8 show a superposition of the results achieved with central difference, upwind, hybrid and power law, and the benchmark solution for the u velocity component along the vertical line through the center of the cavity. There can be seen that for $Re = 100$ the results of all are very similar but as Reynolds number increase the difference between the results gets larger, but for central difference the solution remains in top of the benchmark almost in the entire y line and the closest after central difference is the hybrid followed by the power law, but in all cases the worst approximation to the benchmark solution is upwind, but as Reynolds increase all numerical schemes but central difference tend to the same solutions one that is far from the benchmark. This very same effect can be seen in the solution for v velocity at the horizontal line through the center of the cavity, figures 4.9 show the evolution of the solutions for different Reynolds numbers, and in the same case that for u velocity, the solution achieved

with upwind departs from the solution as the Reynolds increase and the other schemes get closer to the upwind while central difference remains close to the benchmark solution. This behavior was expected due to the first order accuracy in most of the numerical schemes used, but for Central difference a second order is achieved.

Figures 4.10 shows the stream lines at the cavity for the Reynolds numbers studied. The phenomenon is as follow. For small Reynolds the vortex generation its localized at the lower left cornet in which the first eddy start to give rise, then as Reynolds increase this eddy increase its size and another eddy its generated in the opposite corner of the cavity also at the bottom. Notice that in the right upper corner a stream line deformation is taking place where for a large Reynolds the third eddy is generated. Then for larger Reynolds in the same corner where all started a smaller eddy is generated on the back of the first eddy. But the most important is that in the middle there is the largest eddy, this one has been on the cavity for all Reynolds but at the beginning it was not in the center of the cavity but closer to the top and on the left side of the geometrical center and as the Reynolds number increases it was moving to the center of the cavity maybe helped with the effect of the other vortex movement. Figure 4.11 show the corner vortex for $Re = 10000$ there the size and shape of the vortex can be observed and the small vortex its also shown clearly. Notice that the size of the eddies occupied almost the 20% of the total width of the bottom of the cavity each one.

Figures 4.12 and 4.13 show the evolution of the velocity at the horizontal and vertical lines through the center of the cavity respectively. Notice that as Reynolds number increase the velocity profile gets almost linear at the center of the cavity while at the region close to the walls the velocity profile changes in a more abrupt fashion, this phenomenon follows the movement of the main vortex at the center of the cavity, that's because the bigger the Reynolds, the bigger the inertial forces, so the cylinder in the middle tries to govern the entire movement of the cavity, creating three smaller eddies at its corners to maintain the shape of a cylinder.

§ Fractional Step Method

Due to the non iterative behavior of this method, the time taken for the simulation is larger than the one taken for the SIMPLE algorithm because in this case every time step is solve. In comparison with the semi-transient solution of the SIMPLE. According to this the LU decomposition solver was used in order to achieve the solution faster. Due to the similarity of between the solutions achieved with SIMPLE and Fractional Step, we will only make a comparison. Figures 4.14 and 4.15 show the results of the simulation for $Re = 100$ and for a mesh size of 10×10 . In this figures you can see that the result achieved with both methods are very similar, but in this case we where only interested in the stationary solution, ergo the time steps where only usefull to make the calculation. For this reason the Fractional Step took a long time to achieve the expected solution, because as said before, it solves for every time step. In contrast the SIMPLE went

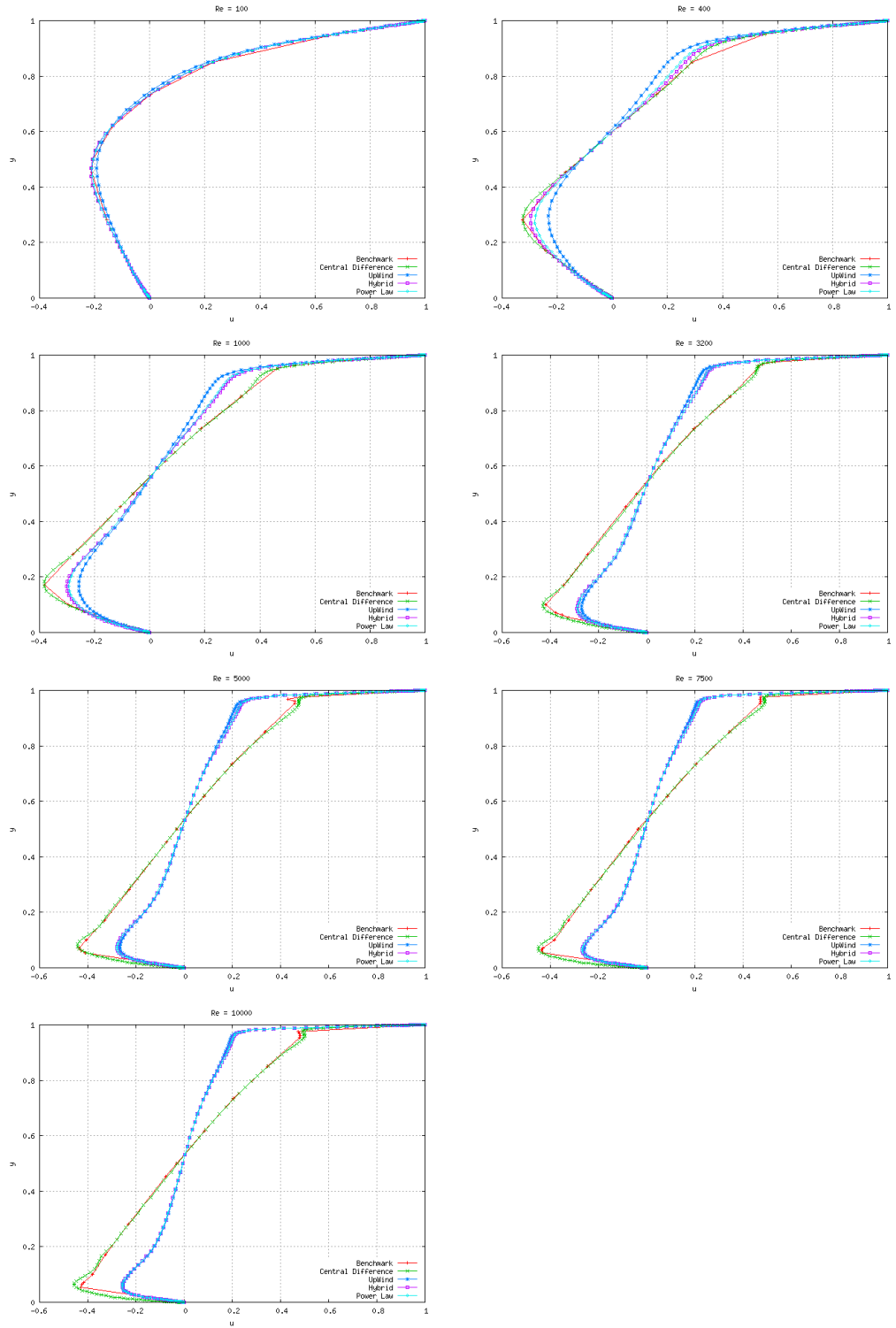


Figure 4.8: u results for different Reynolds numbers

4.1. DRIVEN CAVITY PROBLEM

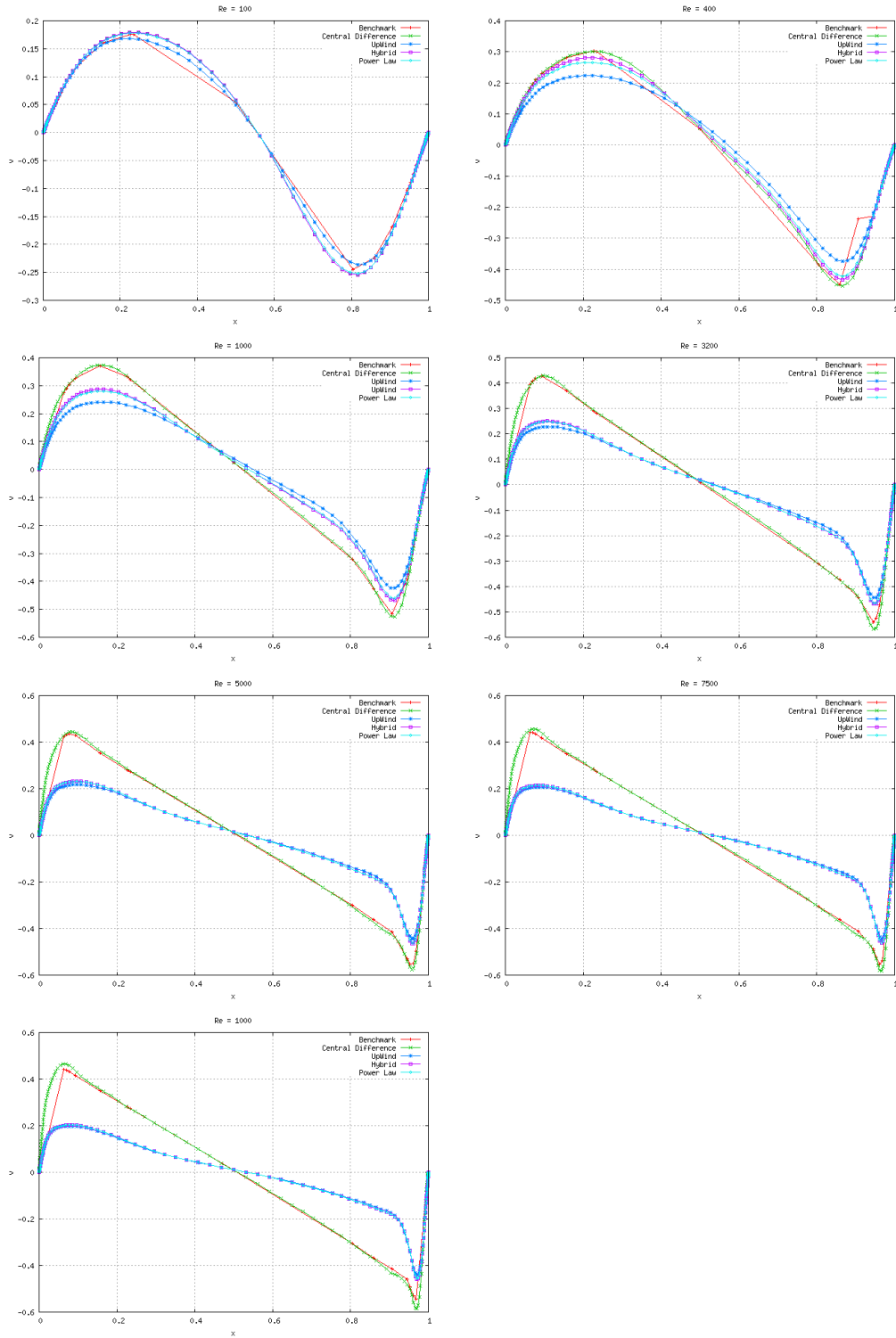


Figure 4.9: v results for different Reynolds numbers

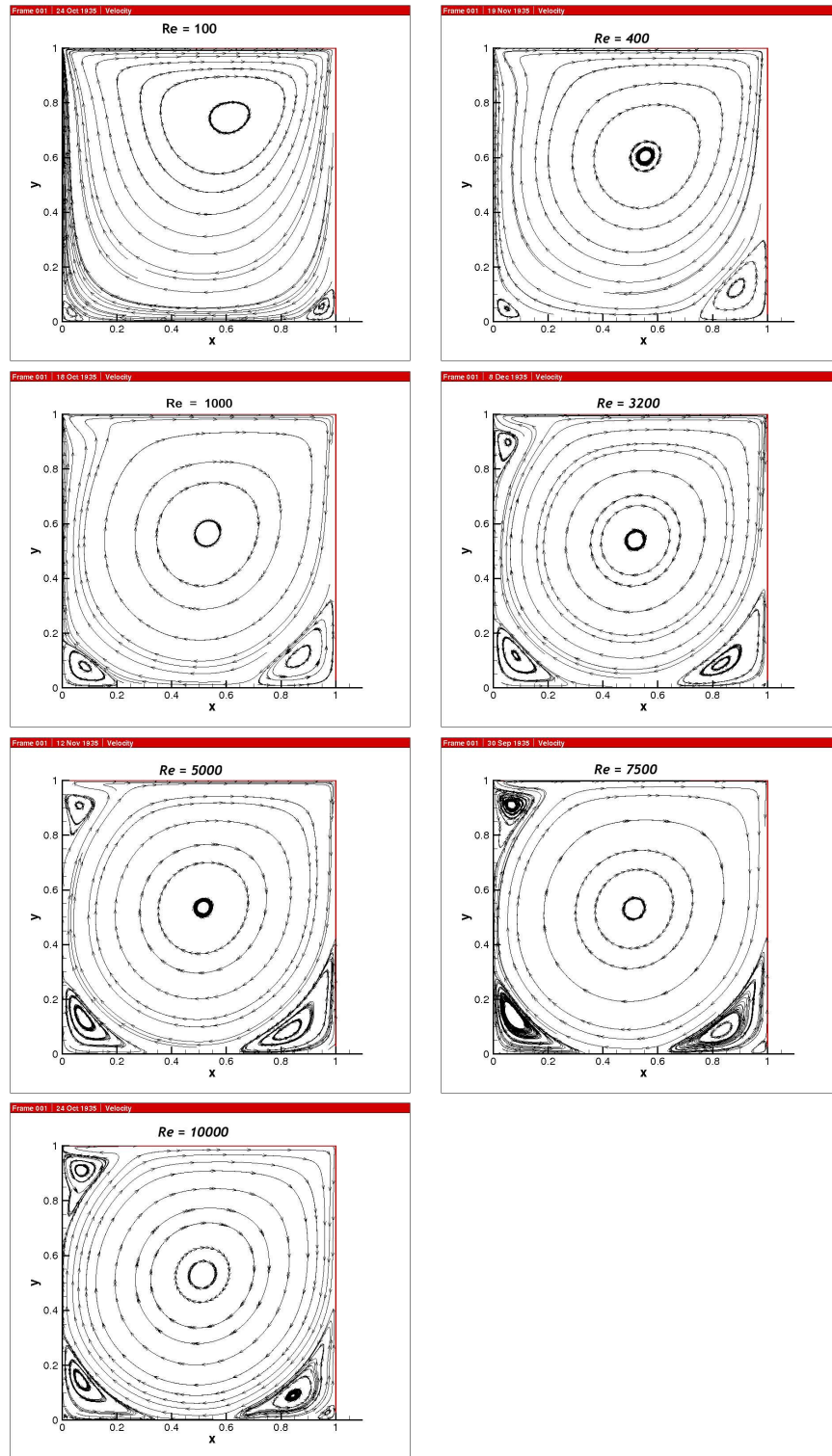


Figure 4.10: Stream lines for different Reynolds numbers

4.1. DRIVEN CAVITY PROBLEM

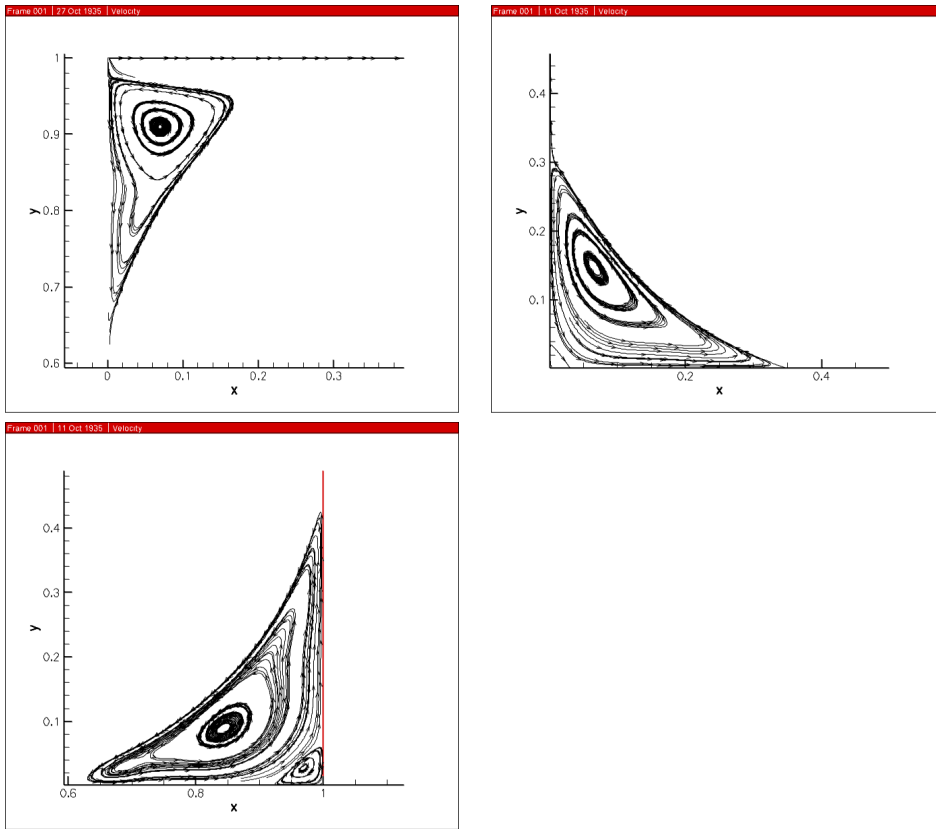


Figure 4.11: Zoom at eddies in cavity corners for $Re = 10000$

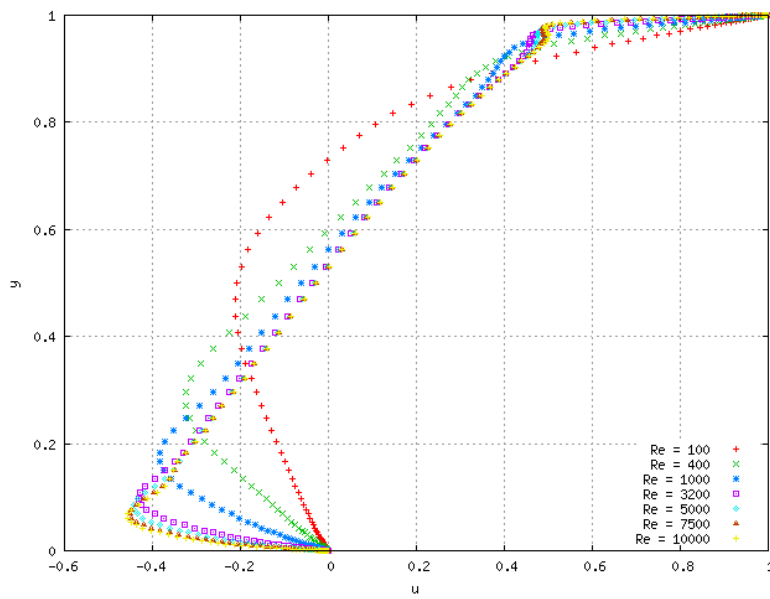


Figure 4.12: u results for all Reynolds numbers

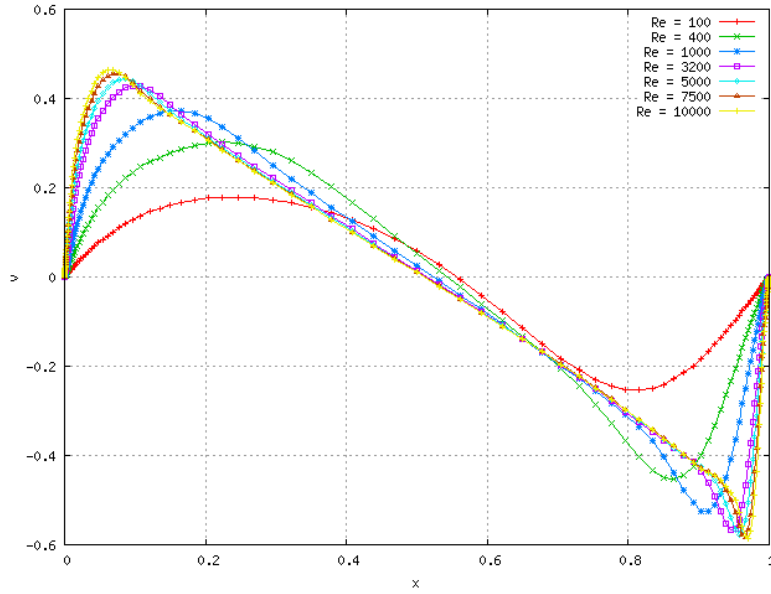


Figure 4.13: v results for all Reynolds numbers

directly to the stationary solution in a short amount of time, avoiding the time step results. SIMPLE in the most probably of the cases would take more time than Fractional Step to get into the stationary solution if the time step would reach the convergence due to its iterative behavior.

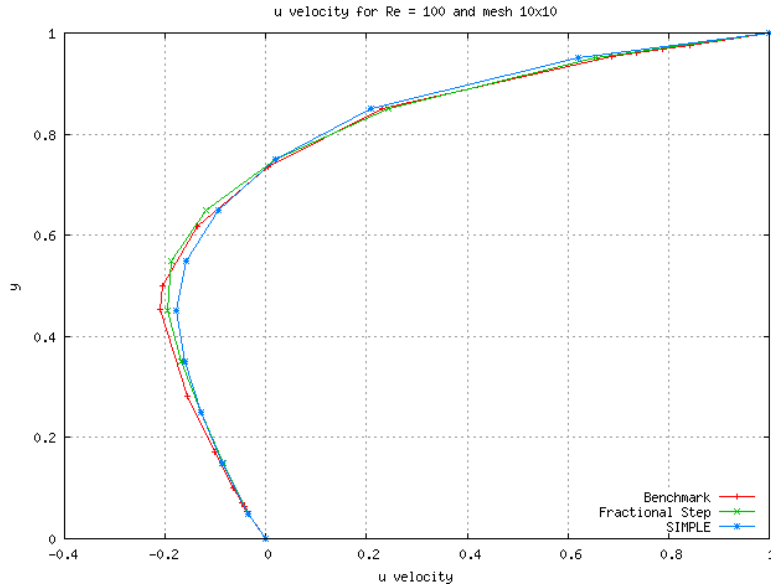


Figure 4.14: Comparison between Fractional Step and SIMPLE algorithms for u velocity

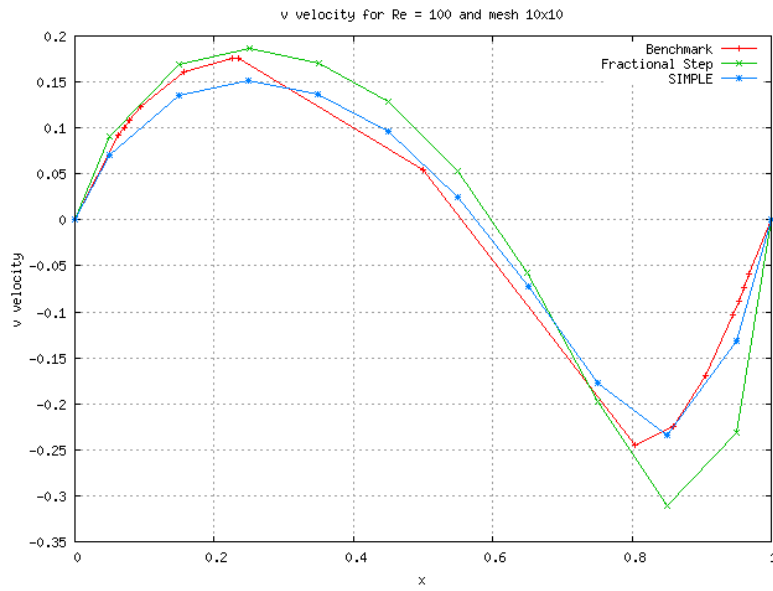


Figure 4.15: Comparison between Fractional Step and SIMPLE algorithms for v velocity

Bibliography

- [1] Suhas V. Patankar: *Numerical Heat Transfer and Fluid Flow*, Hemisphere Publishing Corp.(1980)
- [2] H. K. Versteeg and W. Malalasekera: *An Introduction to Computational Fluid Dynamics*, The finith volume method
- [3] M. S. Darwish and F. H. Moukalled: *Normalized Variable and Space Formulation Methodology for High-Resolution Schemes*, Numerical Heat Transfer, Part B, 26:79-96 (1994)
- [4] M.S. Darwish: *A New High-Resolution Scheme Based on the Normalized Variable Formulation*, American University of Beirut, Faculty of Engineering & Arquitecture (1993)
- [5] Siddharth Thakur and Wei Shyy: *Convection Schemes for Finite-Volume Formulations*, Department of Aerospace Engineering, Mechanical and Engineering Science, University of Florida (1993)
- [6] S. Armfield and R. Street: *Modified fractional-Step methods for the Navier-Stokes equations*, 2004
- [7] CTTC papper: *Introduction to the Fractional Step Method*
- [8] Frank P. Incropera and David P. DeWitt: *Introduction to Heat Transfer*, School of Mechanical Engineering. Purdue University
- [9] Patrick J. Roache: *Code Verification by the Method of Manufactured Solutions*, March 2002