

## FULLY-COUPLED SOLUTION OF FLUID FLOW AND ENERGY EQUATIONS

Dr. Mat Nawi Wan Hassan  
Faculty of Mechanical Engineering  
Universiti Teknologi Malaysia.

### Sinopsis

*Satu skim numerik yang cekap telah dibina untuk menyelesaikan persamaan aliran bendalir dan tenaga. Persamaan pembeza separa eliptik dari persamaan pengawal diubah kepada skim beza terhingga dengan menggunakan grid berselang dan pendekatan isipadu terkawal. Aturcara ini menyelesaikan satu pasang persamaan serentak taklinear dengan menggunakan cara Newton dan teknik matriks renggang. Cara ini tidak memerlukan pengendur bawah atau teknik cepat menumpu sebagaimana digunakan di dalam skim lelar biasa. Walau bagaimanapun, pengendur bawah yang sederhana dapat digunakan, dengan demikian jumlah masa pengiraan makin berkurang. Satu aliran gerakan apungan diaruhkan ke dalam suatu lingkungan dinding isothermal pugak yang panas dan sejuk telah diselesaikan pada  $Ra = 10^6$ . Cara yang digunakan cepat menumpu dan cara ini dapat dilakukan untuk nombor Rayleigh yang lebih besar dan mungkin untuk jenis aliran lain. Kertas ini melaporkan aturcara, hasil dari perkiraan dan perbandingan dengan data numerik yang telah diterbitkan.*

### Synopsis

*An efficient numerical scheme has been developed for the solution of fluid flow and energy equations. The elliptical partial differential equations of the governing equations are transformed into finite difference scheme using staggered grid and control volume approach. The algorithm solves the set of nonlinear simultaneous equations using Newton's method and a sparse matrix technique. The method does not require any underrelaxation or other convergence-enhancing techniques as employed in the normal iterative schemes. However, simple underrelaxation method can be used, thus further reduces the total computational time. A buoyancy driven flow induced in an enclosure by isothermally hot and cold vertical walls has been computed at  $Ra = 10^6$ . The present procedure is rapidly convergent and the method can be extended for higher Rayleigh numbers and possibly for other types of flow. This paper reports the algorithm, the results of the computation and the comparisons with the published numerical data.*

### Introduction

In the past, both experimental and analytical methods have been used to solve some of the fluid flow and heat transfer problems. With the advent of the powerful computer, it is not surprising that, fluid dynamics and heat transfer are contributing to and benefiting from the current development in finite difference numerical analysis.

The numerical solution to a problem depends on:

- (a) the number of equations that governs the system
- (b) the forms of the partial differential equations — parabolic, hyperbolic or elliptic
- (c) the linearity of the equation — linear or nonlinear
- (d) the system of the equations — coupling or not
- (e) the source term in the equations

Generally, for the multidimensional fluid flow and heat transfer phenomena, the governing equations are coupled, nonlinear and elliptic. The partial differential equations (pde) obtained are derived from the continuity, momentum and energy equations. These equations express the transport of mass, momentum, heat and other related variables by the mechanism of convection, diffusion and internal source generations.



In recent years, several finite difference schemes has been proposed and developed. Some methods have used the primitive variables, while some have solved the equations in terms of vorticity and stream function as the dependent variables. The governing equations are often transformed into the nondimensional form. The advantage is that it is more convenient to work with dimensionless variables. The characteristic parameters such as Reynold number, Prandtl number and Rayleigh number can be varied independently. Furthermore, by nondimensionalizing the equations, the flow parameters such as velocity and temperature, are normalized so that their values can be adjusted to fall between certain prescribed limits. A number of general-purpose computer programs using finite difference methods has been developed. Some of these programs have relied on the works of the Los Alamos group<sup>1</sup>, while some have employed the SIMPLE<sup>2</sup> based on the works at Imperial College. For aerospace applications, other schemes and computer programs are also available.

Most of the algorithm using finite difference technique developed for the solution of the fluid flow and heat transfer equations have utilized the procedure based on decoupled technique. If the velocity fields are given, other parameters can be calculated easily. However, in many cases, the flow field is not specified; the local velocity components and the density field must be calculated. There are two main problems of solving these equations: the nonlinearity of the momentum equations and the unknown pressure field. Traditionally, the nonlinearity is handled by iteration. Starting with a guessed temperature and velocity fields, the individual equation of the governing equations is solved iteratively until the converged solution for the velocity components and temperature is obtained. The unknown pressure poses difficulty in the computation of the velocity field and hence the temperature distribution. The pressure gradient of the momentum equations forms a part of the source term. If the pressure is not given, a direct mean of obtaining the pressure is not easy as there is no obvious equation for the pressure field. In SIMPLE algorithm of Patankar and Spalding<sup>2</sup>, the finite difference equations for the  $x$ ,  $y$  (and  $z$ ) momenta and  $T$  of the energy equation are solved in sequence using a previously guessed pressure field. The pressure field is subsequently corrected through the solution of a pressure-correction equation.

The difficulty associated with the determination of the pressure field has led to a new method that eliminates the pressure gradient in the momentum equations. In two dimensional problem, this is done by cross differentiation of the two momentum equations. The result is a vorticity-transport equation. This method forms a basis of the stream function-vorticity procedure as described in literature. The method has some attractive features. As there is no need to solve for the pressure field, only two equations are required to be solved to obtain the stream function and the vorticity; instead of dealing with the continuity and two momentum equations. For simple boundary flow behaviour, such as when an external irrotational flow lies adjacent to the computation domain, the boundary vorticity can easily be set equal to zero. However, when the boundary condition at a wall is needed, the stream function-vorticity method poses some major disadvantages. The value of the vorticity at a wall is difficult to specify and often causes difficulty to obtain a converged solution. Other major disadvantages of using this technique are that rather frequently the pressure term has been eliminated, which unfortunately happens to be a vital parameter for calculation of density and other parameters. The efforts of finding the pressure from the vorticity formulation, then offset the computational savings obtained. The vorticity method is limited to two dimensional flow and cannot easily be extended to three dimensional problem where the stream function does not exist.

This paper deals with the development and application of a robust and efficient scheme for solving finite difference form of fluid flow and heat transfer equations. In the past, as the number of the nonlinear simultaneous equations formed after discretization of the modelling equations is large, an iterative technique is used to update the flow variables from one time state to the next. In the case of steady flow, the variables are updated line by line and are solved independently. The strong coupling among the fluid flow and heat transfer parameters is handled through iterative means. The present method is based on a fully coupled solution of the governing equations.

Newton's method and sparse matrix method are used. The finite difference continuity, momentum and energy equations are solved in the nondimensional form without a pressure or



pressure correction equation. This is done by combining the four equations into one large set, and a Newton-Raphson method is then used to solve the combined set of nonlinear equations. The linearized equations at each Newton iteration are solved directly using a sparse matrix package<sup>3</sup>.

### Governing Differential Equations

The two-dimensional Navier-Stokes and energy equations that governs the flow in an enclosure may be conventionally written as:

Mass Continuity

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

x-momentum

$$\frac{\partial}{\partial x} (\rho uu) + \frac{\partial}{\partial y} (\rho vu) = - \frac{\partial p}{\partial x} + \frac{\partial}{\partial x} (\mu \frac{\partial u}{\partial x}) + \frac{\partial}{\partial y} (\mu \frac{\partial u}{\partial y}) + S_u$$

y-momentum

$$\frac{\partial}{\partial x} (\rho uv) + \frac{\partial}{\partial y} (\rho vv) = - \frac{\partial p}{\partial y} + \frac{\partial}{\partial x} (\mu \frac{\partial v}{\partial x}) + \frac{\partial}{\partial y} (\mu \frac{\partial v}{\partial y}) + S_v$$

energy

$$\frac{\partial}{\partial x} (\rho C_p u T) + \frac{\partial}{\partial y} (\rho C_p v T) = \frac{\partial}{\partial x} (K \frac{\partial T}{\partial x}) + \frac{\partial}{\partial y} (K \frac{\partial T}{\partial y})$$

In the above equations,  $u$  and  $v$  are the  $x$  and  $y$  components of the velocity,  $p$  is the pressure and  $\rho$  is the density,  $\mu$  is the viscosity and  $S_v$  and  $S_u$  represent the other terms not included in the equations. For the laminar buoyancy driven flow  $S_u = 0$  and  $S_v = \rho g$ . For turbulent flow it is necessary to solve the transport equations of turbulence quantities depending on the type of model used. The currently popular "two equation models" employ the equation for kinetic energy of turbulence,  $k$  and its dissipation rate,  $\epsilon$ . The solution of the turbulent flow will be discussed in another paper<sup>4</sup>.

### Finite Difference Equations

There are several procedures for developing the finite-difference equations from a given set of partial differential equations. Among these are: Taylor series expansion, polynomial fitting, integral method and control-volume approach. The control volume approach is used here although other method is possible.

In the development of the control-volume approach, the governing pde's are first transformed into divergence form. If the dependent variables ( $u$ ,  $v$  and  $T$ ) are denoted by  $\phi$ , the general differential equation can be written as:

$$\text{div} (\rho u \phi) = \text{div} (\text{grad } \phi) + S$$

where  $\Gamma$  is the diffusion coefficient.

or

$$\text{div} (\rho u \phi - \text{grad } \phi) = S$$

Using Gauss divergence theorem

$$\begin{aligned} \iiint_V \text{div} (\rho u \phi - \Gamma \text{grad } \phi) dV &= \iint_S (\rho u \phi - \Gamma \text{grad } \phi) \cdot n dS \\ &= \iiint_S dV \end{aligned}$$

Alternatively:

$$\frac{\partial J_x}{\partial x} + \frac{\partial J_y}{\partial y} = S$$

where

$$J_x = \rho u \phi - \Gamma \frac{\partial \phi}{\partial x}$$

$$J_y = \rho v \phi - \Gamma \frac{\partial \phi}{\partial y}$$

The finite difference equations are obtained by integrating the above equation over discrete subdomains of the flow called control volume. A staggered mesh system is employed to locate the flow variables ( $u$  and  $v$ ). The  $u$  velocities are located midway between two pressure nodes in the  $x$  direction, and the  $v$  velocities are located midway between two pressures in the  $y$  direction. The details of the grids and control volumes for the four equations are shown in Figure 1.

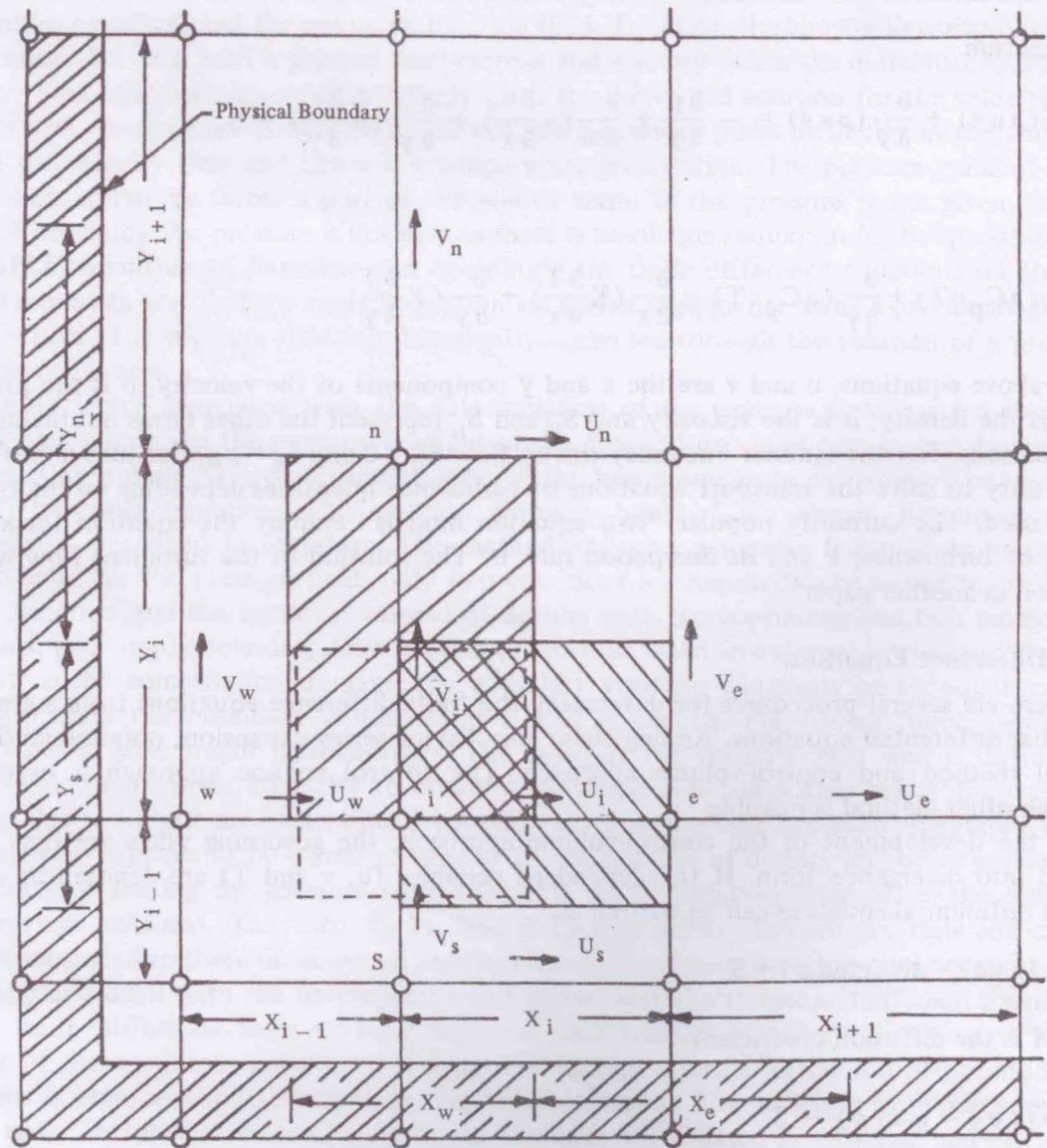


Figure 1 Details of grids and control volume for:  
 x-Momentum ( ▨ ), y-Momentum ( ▩ ), Continuity  
 and energy ( ▭ ) equations.



To integrate the fluxes at the surfaces of the control volume, it is necessary to know an interpolated expression for the influx and efflux at the surface of each control volume. In the present work, this expression is obtained from the solution of a one dimensional convective-diffusive equation without any source term of the form:

$$\rho U \frac{d\phi}{dx} - \Gamma \frac{d^2\phi}{dx^2} = 0$$

However, without modification, the expression contains an exponential function which is relatively expensive to compute. In addition, the actual equation to be solved contains a source term and is two dimensional, and hence has a different solution. The extra expense of computing the exponentials is not justified. A simple scheme that has the same qualitative behaviour known as hybrid scheme [5], is adopted. The following section describes the formulation of the scheme.

### The Hybrid Scheme

Consider the one dimensional convective-diffusive equation of the above form:

$$\frac{d}{dx} \left[ \rho U \phi - \Gamma \frac{d\phi}{dx} \right] = 0 \text{ or } \frac{dJ_x}{dx} = 0, J_x = \rho U \phi - \Gamma \frac{d\phi}{dx}$$

where  $\phi$  represents any transport variable, U, V or T.

For the boundary conditions:

$$x = 0, \quad \phi = \phi_i$$

$$x = X_e, \quad \phi = \phi_e$$

the solution can be written as:

$$\frac{\phi - \phi_i}{\phi_e - \phi_i} = \frac{\text{Exp}(PeX/X_e) - 1}{\text{Exp}(Pe) - 1}$$

where:

$$Pe = \frac{\rho U Y_s}{\Gamma Y_s/X_e} = \frac{F_E}{D_E}$$

= Strength of convection/strength of diffusion

= Peclet number

In the above solution  $\rho U$  and  $\Gamma$  are assumed to be constant. Integrating the equation for east and west surfaces (Figure 2) gives:

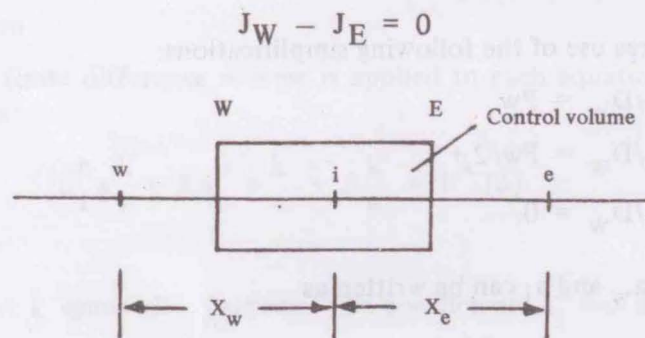


Figure 2 Grid-point cluster for one dimensional problem.

Substituting the solution for  $J_E$  and  $J_W$  we get

$$F_e \left( \phi_i + \frac{\phi_i - \phi_e}{\text{Exp}(Pe) - 1} \right) - F_w \left( \phi_w + \frac{\phi_w - \phi_w}{\text{Exp}(Pw) - 1} \right) = 0$$

where:

$$F_e = (\rho U)_E Y_s$$

$$F_w = (\rho U)_W Y_s$$

Simplification of the above equation leads to:

$$a_i \phi_i = a_e \phi_e + a_w \phi_w$$

where:

$$a_e = \frac{F_e}{\text{Exp}(Pe) - 1}$$

$$a_w = \frac{F_w \text{Exp}(Pw)}{\text{Exp}(Pw) - 1}$$

$$a_i = a_e + a_w + (F_e - F_w)$$

The variation of  $a_w$  is shown below:

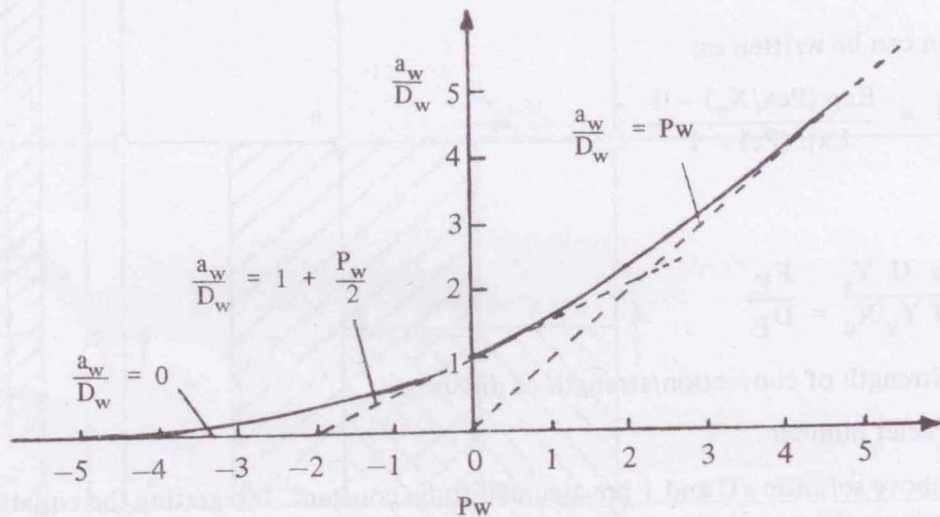


Figure 3 Variation of coefficient  $a_w$  with Peclet number.

The hybrid scheme makes use of the following simplifications:

$$P_w > 2. \quad a_w/D_w = P_w$$

$$-2. < P_w < 2. \quad a_w/D_w = P_w/2 + 1.$$

$$P_w < -2. \quad a_w/D_w = 0.$$

The final result for  $a_e$ ,  $a_w$  and  $a_i$  can be written as

$$a_e = \text{AMAXI}(-F_e, D_e - F_e/2, 0)$$

$$a_w = \text{AMAXI}(F_w, D_w + F_w/2, 0)$$



$$a_i = a_e + a_w + (F_e - F_w)$$

Application of the above scheme to two dimensional flow of the form:

$$\frac{\partial J_x}{\partial x} + \frac{\partial J_y}{\partial y} = S$$

leads to the following results:

$$F_\phi = a_i \phi_i + a_n \phi_n + a_s \phi_s + a_e \phi_e + a_w \phi_w + S_\phi$$

where:

$$F_n = \rho V_N X_w, \quad D_n = \Gamma_N X_w / Y_i, \quad P_n = F_n / D_n$$

$$F_s = \rho V_S X_w, \quad D_s = \Gamma_S X_w / Y_{i-1}, \quad P_s = F_s / D_s$$

$$F_e = \rho U_E Y_s, \quad D_e = \Gamma_E Y_s / X_i, \quad P_e = F_e / D_e$$

$$F_w = \rho U_W Y_s, \quad D_w = \Gamma_W Y_s / X_{i-1}, \quad P_w = F_w / D_w$$

$$a_n = \frac{F_n}{\text{Exp}(P_n) - 1} = \text{AMAX1}(-F_n, D_n - F_n/2, 0)$$

$$a_s = \frac{F_s \text{Exp}(P_s)}{\text{Exp}(P_s) - 1} = \text{AMAX1}(F_s, D_s + F_s/2, 0)$$

$$a_e = \frac{F_e}{\text{Exp}(P_e) - 1} = \text{AMAX1}(-F_e, D_e - F_e/2, 0)$$

$$a_w = \frac{F_w \text{Exp}(P_w)}{\text{Exp}(P_w) - 1} = \text{AMAX1}(F_w, D_w + F_w/2, 0)$$

$$a_i = -F_n - \frac{F_n}{\text{Exp}(P_n) - 1} - \frac{F_s}{\text{Exp}(P_s) - 1}$$

$$-F_e - \frac{F_e}{\text{Exp}(P_e) - 1} - \frac{F_w}{\text{Exp}(P_w) - 1}$$

$$= -(a_n + a_s + a_e + a_w) + (F_w - F_e) + (F_s - F_n)$$

$$S_\phi = \int_{cv} S \, dx dy$$

#### Method of solution

When the above finite difference scheme is applied to each equation, the result is a set of equations of the form:

$$a_i^k \phi_i^k + \sum a_n^k \phi_n^k + S_\phi^k = F^k(\phi)$$

where the superscript  $k$  spans all equations. The coefficients  $a_i^k$  and  $a_n^k$  are constants for the continuity equation, while in other equations these coefficients contain velocities and diffusive coefficients and are thus variables.

Each set of equations is coupled with each other, either through intervariable coupling, transport effects or source terms. To solve these equations a scheme as suggested by Vanka [6] is adopted. The set of continuity, momentum and energy equations is solved in a coupled manner using Newton's method. For each grid point the equations are arranged in the above order to produce a matrix of a narrow bandwidth. The system of equations thus obtained is then solved directly using a sparse matrix solver [3] without an ordering routine.

The number of equations formed for a n by n grid is:

Continuity Equation: [n] x [n]

X-Momentum Equation: [n-1] x [n]

Y-Momentum Equation: [n] x [n-1]

Energy Equation: [n] x [n]

Total number of equations is 2 [nxn + n(n-1)]

If the continuity, x-momentum, y-momentum and energy equations for all nodes are represented respectively by FC, FU, FV and FE and F(x) is the large set of these equations, then we can write:

$$F(\underline{x}) = [FC, FU, FV, FE]^T = 0$$

$$\underline{x} = [U, V, P, T]^T$$

Consider a one dimensional problem of the form  $f(x) = 0$ . Suppose  $x_0$  is the first approximation to the solution. We can write.

$$f(x) = f(x_0) + (x-x_0) f'(x_0) + (x-x_0)^2 f''(x_0)/2 + \dots$$

An improved approximation to the root can be obtained by setting  $f(x) = 0$  and taking the first two terms of the expansion giving:

$$x = x_0 - f(x_0)/f'(x_0)$$

The general formulation for n iterations can be written as:

$$x^{n+1} = x^n + f(x^n)/f'(x^n)$$

Extending the method to a multi-component system, we have:

$$\underline{x}^{n+1} = \underline{x}^n + \underline{p}^n$$

where

$$\underline{p}^n = -J(\underline{x}^n)^{-1} f(\underline{x}^n)$$

$J(\underline{x}^n)$  is the Jacobian of size [s] x [s] of  $f(\underline{x})$  with components

$$J_{ij}(\underline{x}) = \partial f_i(\underline{x})/\partial x_j$$

Starting from an initial estimate for U, V, P, T and boundary conditions, the iteration process proceeds until convergence. The convergence criteria used in this case is that the infinity norms of the solution be less than  $1 \times 10^{-2}$ . After a certain number of iterations the Jacobian matrix can be frozen to save computing time. The method is further improved by the introduction of a damping factor  $\alpha$  in the following manner.

$$\underline{x}^{n+1} = \underline{x}^n + \alpha^n \underline{p}^n$$

and  $\alpha$  is chosen such that

$$\|f(\underline{x}^{n+1})\| < \lambda \|f(\underline{x}^n)\| \text{ with } 0 < \lambda < 1$$

There are many ways of choosing  $\alpha$ ; the method used as suggested by Armijo [7] in the context of minimization of residuals is

$$\alpha^n = 2^{-m}, m = 0, 1, 2, 3, \dots$$

where m is the smallest integer such that

$$\|f(\underline{x}^n + 2^{-m} \underline{p}^n)\| < \lambda \|f(\underline{x}^n)\|$$



### Solution Algorithm

The sequence of the steps in solving the equations is:

1. Discretize the equations using the control volume approach into finite difference equations using non-uniform grids.
2. With proper boundary conditions, the initial values of  $U$ ,  $V$ ,  $P$  and  $T$  are assumed.
3. The Jacobian of the continuity, x-momentum, y-momentum and energy equation and right hand side of the simultaneous equations is determined. It is stored according to the required sparse storage scheme. The system of equation is then solved using sparse package solver.
4. Step 3 is repeated until  $U$ ,  $V$ ,  $P$  and  $T$  satisfy the convergence criterion. The matrix can be frozen after a few iterations to reduce the total CPU time by 10 – 15 percent.

### The Test Results

The above algorithm has been tested for the calculation of a two dimensional buoyancy flow in a square cavity. At low Rayleigh numbers, the Newton's iteration converged rapidly. However, no comparisons of the CPU time were made, as data using other methods are not available. The results of the calculation performed to find the flow patterns, the temperature distributions, the average Nusselt number and other data are presented as required by reference [8]. Comparisons of results with the work of de Vahl Davis, Jones I., Portier and Quon were made for  $Ra = 10^5$  and  $Ra = 10^6$ . We have performed calculations with  $21 \times 21$  grids. Figure 4, 5 and 6 show the distribution of the nondimensional  $T$ ,  $V$  and  $U$ . The flow and the temperature fields show a similar pattern as computed by other authors. The comparison of the data as in Table 1 shows that the present results seem to agree very well.

### Summary

In this study, an alternative development of a new solution procedure for engineering application of laminar flow are presented. The method can be combined with other available procedure to solve more complicated flow such as turbulent flows where solutions to the turbulent parameters are required. The algorithm was based on Newton's method and sparse matrix technique for direct solution of the system of equations. The calculation procedure has been applied to buoyancy driven flow in a square cavity. The disadvantage of the present procedure is its requirement of more computer storage over that required by other algorithm.

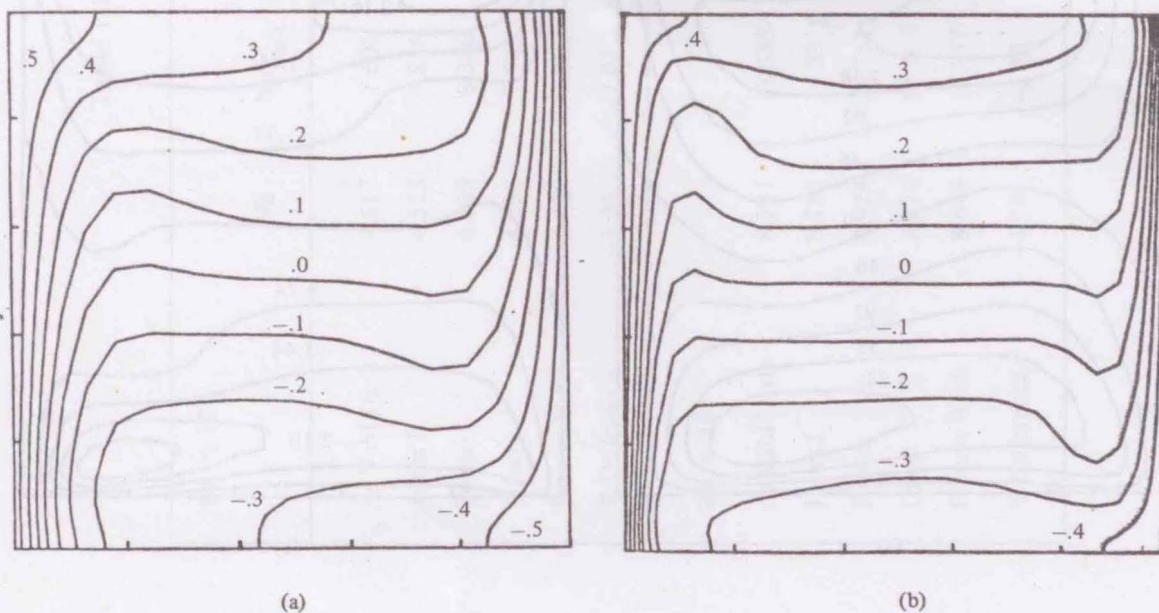


Figure 4 Contour maps of non-dimensional temperature  $\theta$  : (a)  $Ra = 10^5$ , (b)  $Ra = 10^6$ .



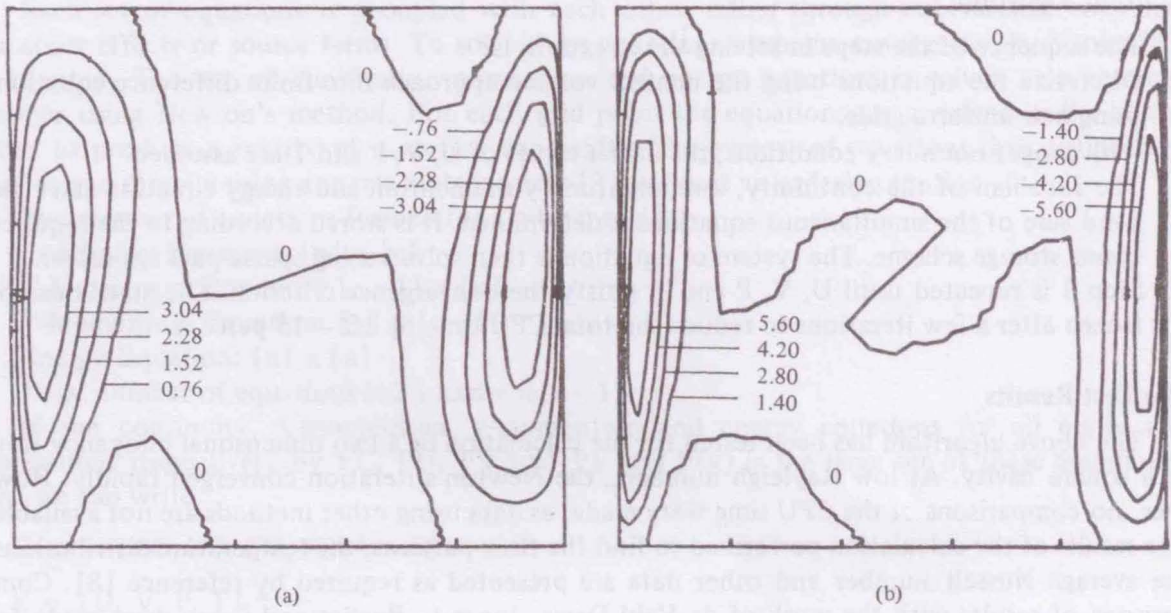


Figure 5 Contour maps of non-dimensional vertical velocity  $V$ : (a)  $Ra = 10^5$ , (b)  $Ra = 10^6$ .

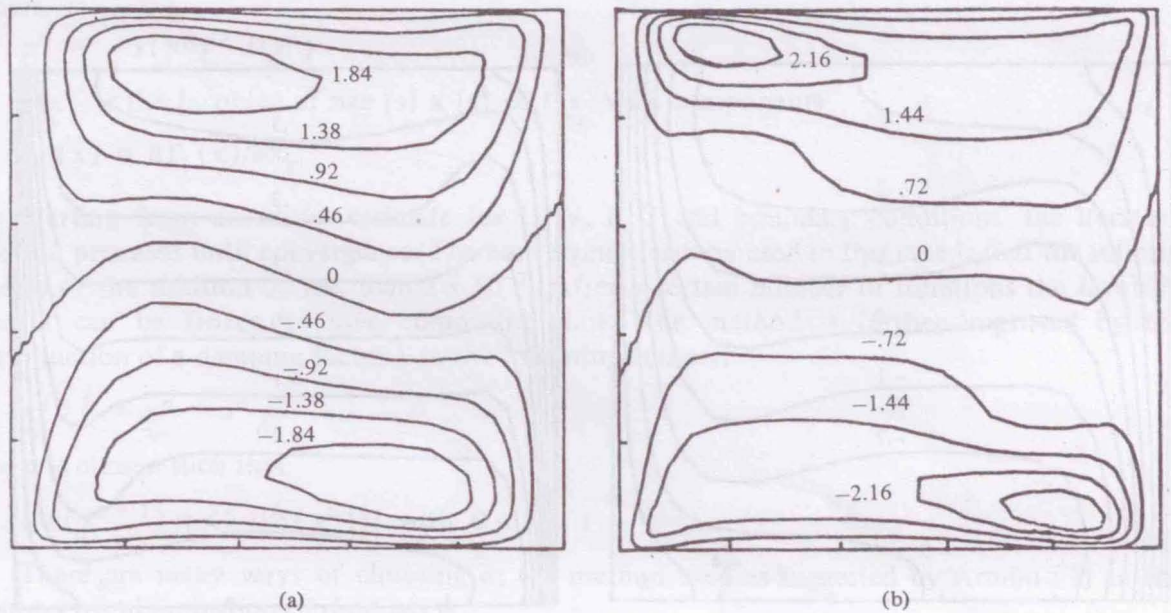


Figure 6 Contour maps of non-dimensional horizontal velocity  $U$ : (a)  $Ra = 10^5$ , (b)  $Ra = 10^6$ .



Table 1 Comparisons of Nu,  $\bar{U}$  and  $\bar{V}$  at  $10^3 < Ra < 10^6$ .

<b>Ra = 10<sup>5</sup></b>									
	Nu	Nu <sub>max</sub>	$\bar{Y}$	Nu <sub>min</sub>	$\bar{Y}$	$\bar{U}_{max}$	$\bar{Y}$	$\bar{V}_{max}$	$\bar{X}$
de Vahl Davis	4.517	7.605	0.083	0.739	1.000	34.81	0.855	68.68	0.066
Jones I	4.523	7.816	0.088	0.734	1.000	35.36	0.855	68.65	0.067
Portier	4.869	8.386	0.075	0.705	1.000	35.16	0.855	68.14	0.067
Quon	4.538	7.568	0.087	0.808	0.996	34.18	0.846	68.26	0.068
Present Work	4.429	7.686	0.073	0.775	1.000	34.41	0.861	67.00	0.056
% Difference	-3.96	-2.01		3.74		-1.34		-2.09	
<b>Ra = 10<sup>6</sup></b>									
de Vahl Davis	8.797	18.635	0.039	1.065	1.000	64.96	0.85	221.28	0.038
Jones I	8.783	16.89	0.058	1.011	1.000	69.07	0.85	219.5	0.040
Portier	9.924	20.761	0.042	0.832	1.000	65.46	0.858	217.2	0.040
Quon	8.871	17.765	0.033	1.011	0.996	65.59	0.857	219.74	0.033
Present Work	8.648	17.857	0.026	1.053	1.000	69.17	0.866	222.51	0.035
% Difference	-4.90	-3.54		7.44		4.37		1.40	



## References

1. A. A. Amsden, and F. H. Harlow, "The SMAC Method: A Numerical Technique for Calculating Incompressible Fluid Flows," Los Alamos Scientific Laboratory Report, LA-4370, 1970.
2. S. V. Patankar and D. B. Spalding, "A Calculation Procedure for Heat, Mass, and Momentum Transfer in Three-dimensional Parabolic Flows," Int. J. Heat Mass Trans., Vol 15, pp. 1978-1806, 1972.
3. Eisenstat S. C., M. C. Gursky, M. H. Schultz and A. H. Sherman, "Yale Sparse Matrix Package II: The Nonsymmetric Codes", Res. Rep 114, 1977.
4. Wan Hassan, Mat Nawi, "A Numerical Prediction of Turbulent Flow using  $k\epsilon$  model". (To be published)
5. Patankar, Suhas V., Numerical Heat Transfer and Fluid Flow, McGraw-Hill Book Company, Hemisphere Publishing Corporation, 1980.
6. Vanka, S. P., G. K. Leaf, "Fully-Coupled Solution of Pressure-Linked Fluid Flow Equations", Argonne National Laboratory, Argonne, Illinois, 1983.
7. Armijo, Larry, "Minimization of Functions Having Lipschitz Continuous First Partial Derivatives", Pacific Journal of Mathematics, Vol. 16, No. 1, 1966.
8. Davis, G. De Vahl and I. P. Jones, "Natural Convection In A Square Cavity: A Comparison Exercise", Internat. Journal for Numerical Methods in Fluids", Vol. 3, pp. 227-248, 1983.