COBEM 99
XV Congresso Brasileiro de Engenharia Mecânica
15th Brazilian Congress of Mechanical Engineering
22 - 26 de Novembro de 1999 / November 22 - 26, 1999 Águas de Lindóia, São Paulo,

SPARSE-MATRIX NUMERICAL SOLUTION OF MULTIDIMENSIONAL CONVECTION-DIFFUSION PROBLEMS

Gustavo Di Fiore dos Santos
Marcelo J.S. de Lemos
Departamento de Energia - IEME
Instituto Tecnológico de Aeronáutica - ITA
12228-900 – São José dos Campos, SP, Brasil - E-mail: mdelemos@tecsat.com.br

Abstract. *This work presents comparisons of the required computational effort to solve two-dimensional convection-diffusion problems using different solution techniques. The effects of Peclet number and flow direction on algorithm performance are discussed upon. Iterative methods, such as the Strong Implicit Procedure, uses minimum computer time and were found to be rather insensitive to the flow direction. Sparse-matrix direct solvers followed in computational performance and showed some dependence on velocity direction.*

Key Words: *CFD, Iterative Methods, Sparse Matrix, Linear Systems*

1. INTRODUCTION

Fluid flow and heat transfer problems can today be tackled with the help of sophisticated numerical tools. Software vendors already added in their packages many different physical models and several well-established numerical algorithms. In spite of the ever-greater use of computational fluid dynamics, flow non-linearity, geometry complexity, advanced physical models and problem size may, often, cause difficulties to the reach of a final converged solution. This, consequently, motivates the search of more robust, less time-consuming computational algorithms.

Based on the above, this paper presents comparisons of different linear equation solvers when applied to the solution of multidimensional convection-diffusion problems. Incomplete and complete full-field matrix decomposition schemes were applied and compared. Emphasis was put on appropriate methods for solving large systems. The Doolittle method is based on the matrix decomposition idea and has the important advantage of avoiding unnecessary use of computer memory. Also applied is the well-known Strong Implicit Procedure SIP (Stone, 1968) which considers matrix decomposition and has proven to be appropriate for flow problems. A comparison of the necessary computational effort for solving identical problems is reported.

2. MATHEMATICAL MODEL AND NUMERICS

The present work considers a conduction-convection stationary laminar flow over a rectangular domain with temperature prescribed at the boundaries. The velocity field is known and kept constant and no internal heat sources are considered. A structured orthogonal regular mesh is employed for discretization along with a cell-centered finite-difference scheme. A schematic is shown in Fig. 1

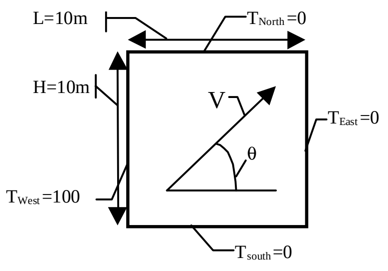

Figure 1 - Computational domain and boundary conditions.

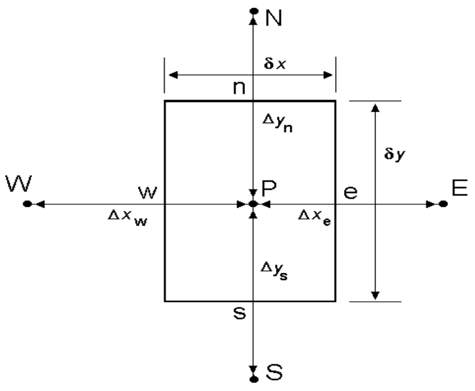

Figure 2 - Control volume for discretization.

As the hydrodynamic problem is assumed to be solved, it only remains to obtain the solution of the energy equation which, in the present case, can be written in Cartesian coordinates as,

$$\frac{\partial}{\partial x}(\rho u T)+\frac{\partial}{\partial y}(\rho v T)=\frac{\partial}{\partial x}\left(\frac{\mathrm{k}}{\mathrm{c_P}}\frac{\partial T}{\partial x}\right)+\frac{\partial}{\partial y}\left(\frac{\mathrm{k}}{\mathrm{c_P}}\frac{\partial T}{\partial y}\right) \tag{1}$$

where ρ is the fluid density, *u* and *v* are the *x* and *y* velocity components, respectively, *T* is the temperature, *k* is the thermal conductivity and c_p the specific heat at constant pressure. Except for T, all quantities are held constant.

The values adopted for the fluid properties are k/c_p=1 kg/m.s and ρ=1kg/m^3. Also analyzed herein is the strictly conductive problem (*u*=*v*=0) which, for the boundary conditions shown in Fig. 1, admits analytical solution (found in any textbook on Heat Transfer).

Algebraic equations are obtained by integrating Eq. (1) over a typical control volume as sketched in Fig. 2 (Patankar, 1980). Internodal interpolation follows the *Weighted Upstream Differencig Scheme* proposed by Raithby & Torrance (1974). It makes use of two coefficients α and β which weigh convective and diffusive processes. Taking the east face as an example, the temperature and its gradient at the interface are approximated by

$$T_\mathrm{e}=\left(\frac{1}{2}+\alpha_\mathrm{e}\right)T_\mathrm{P}+\left(\frac{1}{2}-\alpha_\mathrm{e}\right)T_\mathrm{E} \tag{2}$$

$$\left.\frac{\partial T}{\partial x}\right|_{\mathrm{e}} = \beta_{\mathrm{e}}\left(\frac{T_{\mathrm{E}} - T_{\mathrm{P}}}{\Delta x_{\mathrm{e}}}\right) \tag{3}$$

with coefficients being expressed as

$$\alpha_{\mathrm{e}} = \frac{\mathrm{Pe}_{\mathrm{e}}^{2}}{10 + 2\,\mathrm{Pe}_{\mathrm{e}}^{2}}\ ,\ \beta_{\mathrm{e}} = \frac{1 + 0.005\,\mathrm{Pe}_{\mathrm{e}}^{2}}{1 + 0.05\,\mathrm{Pe}_{\mathrm{e}}^{2}} \tag{4}$$

where Pe_e is the control volume *Peclet number* defined as

$$\mathrm{Pe}_{\mathrm{e}} = Pe\_x = \left.\frac{\rho\,\mathrm{u}\,\Delta x}{\mathrm{k}/\mathrm{c}_{\mathrm{P}}}\right|_{\mathrm{e}} \tag{5}$$

Note that in Eq. (5), due to the use of an orthogonal coordinate system and a regularly spaced grid, the Peclet number at the east face can also be characterized by the subscript *x*. A similar expression holds for the *y* direction:

$$\mathrm{Pe}_{\mathrm{n}} = Pe\_y = \left.\frac{\rho\,\mathrm{v}\,\Delta y}{\mathrm{k}/\mathrm{c}_{\mathrm{P}}}\right|_{\mathrm{n}} \tag{6}$$

Applying Eq. (2) and (3) to all four control volume faces in Fig. 2 one arrives at

$$A_{\mathrm{P}}T_{\mathrm{P}} = A_{\mathrm{E}}T_{\mathrm{E}} + A_{\mathrm{W}}T_{\mathrm{W}} + A_{\mathrm{N}}T_{\mathrm{N}} + A_{\mathrm{S}}T_{\mathrm{S}} \tag{7}$$

with the east coefficient being defined as

$$A_{\mathrm{E}} = -\left(\frac{1}{2} - \alpha_{\mathrm{e}}\right)C_{\mathrm{e}} + \beta_{\mathrm{e}}D_{\mathrm{e}} \tag{8}$$

where

$$C_{\mathrm{e}} = (\rho u)_{\mathrm{e}}\,\delta y \quad \text{and} \quad D_{\mathrm{e}} = \left.\frac{\mathrm{k}}{\mathrm{c}_{\mathrm{P}}}\frac{\delta y}{\Delta x}\right|_{\mathrm{e}} \tag{9}$$

are the convective and diffusive fluxes respectively. The remaining coefficients are similarly defined.

Equation (7), written into matrix form embracing all grid nodes represented in Fig. 2 on computational domain of Fig. 1, reads,

$$\mathbf{A}x = b \tag{10}$$

Matrix **A** in (10) is a N^2xN^2 sparse matrix, where N is the number of unknown temperatures in each direction. It can be solved by several methods. Due to hypotheses here assumed (fixed velocity, constant properties, see Fig. 1, **A** will be constant. In general, the larger the fluid flow problem is, the more sparse matrix **A** will be. Also, convective and diffusive terms weighted by coefficients (4) will depend on the flow direction such that distinct terms in Eq. (7) may be of quite different values even for regular meshes. This, in turn, will unbalance symmetric terms in **A** with corresponding consequence in the solution of (10). Next section briefly characterizes the methods used in solving such system followed by the results here shown.

3. SOLUTION OF LINEAR SYSTEMS

In this work, several methods for solving the linear system (10) were used. They are here recalled by the names: 1) **MARKOWITZ** Strategy, 2) Method of **DOOLITTLE** and 3) **SIP**, the Strongly Implicit Procedure of Stone (1968).

The Markowitz technique involved the use of IMSL (1991) standard routines that were designed, supposedly, with no particular system in mind. The Doolittle and SIP methods take advantage of matrix sparsity and, for that, are well suitable for numerical treatment of convection-diffusion problems. In the Markowiz and Doolittle methods the solution is sought after complete matrix decomposition so no iterations are necessary. On the other hand, the SIP method is iterative so that a solution refinement scheme is necessary before final converged calculation is reached. For those cases, a maximum normalized residual for Eq. (7) equal to $1x10^{-5}$ has been used.

Direct methods, at a first glance not suitable for solving large systems, find its way in connection with multigrid techniques (Rabi & de Lemos, 1998). There, coarse grid matrices are usually of relative small size justifying direct matrix decomposition. Also, the last three of the above methods take advantage of the system sparsity. The algorithm of Stone (1968) stores only non-zero entries of the coefficient matrices and for that it uses the minimum amount of memory. The other two (Markowitz and Doolittle) also preserve matrix sparsity by proper pivoting during decomposition (Markowitz 1957, Duff et al 1986). A brief description of these methods is presented below:

1) MARKOWITZ Strategy (IMSL routines LFTXG/ LFSXG):, This method solves a sparse system of linear equations given the *LU* factorization of the coefficients. The sparse coordinate format for matrix **A** requires one real and two integer vectors. The real array contains all the non-zeros in **A**. If the number of non-zeros is *nz*, then the two integer arrays *irow*() and *jcol*(), each of length *nz*, contain the *row* and *column* numbers for these entries in **A**. That is

$\mathbf{A}_{\mathbf{irow(i),jcol(i)}} = a(i), \quad i = 1, \ . \ . \ . \ , nz$

with all other entries of null value. The factorization is performed by calling first the routine LFTXG. A subsequent call to routine LFSXG follows in order to solve the linear equation set given its *LU* decomposition. The solution of the linear system is then found by standard forward and backward substitution. In summary, the algorithm can be expressed as $\mathbf{PAQ} = LU$ where **P** and **Q** are the row and column permutation matrices determined by the Markowitz strategy (Markowitz, 1957). This method was conceived in order to preserve matrix sparsity during pivoting (for details see Duff et al. 1986). Finally, the solution $x=\mathbf{A}^{-1}b$ is obtained by the following calculations: 1) $Lz = \mathbf{P}b$ 2) $Uy = z$ 3) $x = \mathbf{Q}y$ For more details, see Crowe et al. (1990).

2) Method of DOOLITTLE: Here a *LU* complete decomposition in also sought such that

$$\mathbf{A} = LU \tag{11}$$

with $l_{ij} = 0$ $(j>i)$ $u_{ij} = 0$ $(j<i)$. The elements of the main diagonal of the matrix *L* (or the matrix *U*) can be given an arbitrary value except zero. In the method of Doolittle one sets

$$l_{ij} = 1 \tag{12}$$

In a variant of the method (Crout algorithm) the condition (12) is replaced by u_{ij} = 1. Carrying on formally the multiplication (11) one obtains the following *n* independent systems of equations:

$$l_{11}u_{1j} = a_{ij}; \quad (\mathrm{j}=1,2,...,\mathrm{n}) \qquad (13)$$

$$\begin{cases} l_{21}u_{11} = a_{21} \\ l_{21}u_{1j} + l_{22}u_{2j} = a_{2j} \end{cases}; \quad (\mathrm{j}=2,3,...,\mathrm{n}) \qquad (14)$$

$$\begin{cases} l_{31}u_{11} = a_{31} \\ l_{31}u_{12} + l_{32}u_{22} = a_{32} \\ \sum_{k=1}^{2} l_{3k}u_{kj} + l_{33}u_{3j} = a_{3j} \end{cases}; \quad (\mathrm{j}=3,4,...,\mathrm{n}), \quad \text{etc...} \qquad (15)$$

Taking into account (12), the first system Eq. (13) yields the values of the elements of the first row of the matrix *U* ($u_{1j}=a_{ij}$); the second system (14) yields l_{21}, u_{22}, u_{23},...,u_{2n}; the third system (15) yields l_{31}, l_{32}, u_{33}, u_{34},..., u_{3n} and so on.

The solution of the systems (13)-(14)-(15) can be written in compact form:

$$u_{1j} = a_{1j}; \quad (\mathrm{j}=1,2,...,\mathrm{n}) \qquad (16)$$

$$l_{i1} = {}^{a_{i1}}\!/\!{}_{u_{11}}; \ (\mathrm{i}=1,2,...,\mathrm{n}) \qquad (17)$$

$$l_{ij} = \frac{\left(a_{ij} - \sum_{k=1}^{j-1} l_{ik}u_{kj}\right)}{u_{jj}} (\mathrm{i} = 3,4,...,\mathrm{n}), (\mathrm{j} = 2,3,...,\mathrm{i}-1) \qquad (18)$$

$$u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik}u_{kj}; \quad (\mathrm{i} = 2,3,...,\mathrm{n}), (\mathrm{j} = \mathrm{i},...,\mathrm{n}) \qquad (19)$$

This method, summarized by formulae (16)-(19), preserves the structure of the matrices *L* and *U* and is applicable only to regular matrices **A,** because, in this case, $u_{jj}\neq 0$ (*j*=1,2,...,n) However, it can be proved that for fluid problems as we deal here, the matrix **A** is and therefore regular. Also, information about memory management and storage was not available to the user.

3) <u>SIP</u> (*Strongly Implicit Procedure*): Stone (1968) has recognized that the sparsity and diagonal structure of the matrix **A** could be exploited to advantage. His idea was to look for two triangular matrices, one lower *L* and one upper *U*, whose product was a good approximation of the matrix **A** . This class of method is known as an Incomplete Factorization Scheme since the *LU* product is only an approximation of the full matrix **A** . In the SIP approach, the matrices *L* and *U* have only three diagonal and their product is arranged in such a way that it results also in a 5-diagonal matrix **A***. However, since this

matrix is not exactly the same as the one in (10), the solution being sought needs an iterative refinement. To this end, Eq. (10) may be rewritten as

$$\mathbf{A}^* x = \mathbf{A}^* x + \{\mathbf{A}x - b\} \qquad (20)$$

If we now introduce an iteration counter as superscript on x in (20), "n" on the left and "n-1" on the right hand side, and denote further the increment $\delta = x^n - x^{n-1}$ and the residue $R = \mathbf{A}x^{n-1} - b$, one finally gets,

$$\mathbf{A}^* \delta = R \qquad (21)$$

Due to the easy triangular and tridiagonal structure of matrices L and U, Eq. (21) gives δ for repetitive updating of x^n until $R \to 0$. Information about the iterative strategy and memory storage were not available to the user.

4. RESULTS AND DISCUSSION

4.1 Computational Details. The computer used was based on a Intel Pentium Pro 200MHz processor with 64Mb of RAM. For these three methods used, which reduce computer storage taking advantage of matrix sparsity (algorithms of Markowitz, Stone and Doolittle), grids up to 88 points in each direction were calculated. Not considering boundary values as unknown, this means $(90-2)^2 = 7744$ equations.

4.2 Effect of *Pe_x*. Calculations using the methods of DOOLITTLE, MARKOVITZ and SIP had the x-direction Peclet number, defined in (5), varied in the range $0 \le Pe_x \le 1000$ (flow from left to right, $\theta = 0°$ in Fig. 1). Fig. 3 shows the time necessary, in seconds, when the solution of system (10) is sought with the Markowitz Strategy. Apparently, the convective strength in the x-direction does not influence computing time with results being close to those of a pure diffusive case ($Pe_x = 0$). Fig. 4 shows similar calculations using Doolittle's method and the weight of convective and diffusive coefficients, given by the factors in Eq. (4), seems to increase the necessary computational effort as Pe_x increases. Results in Fig. 5 for the SIP method presents a strong reduction on t, even for very low Peclet numbers.

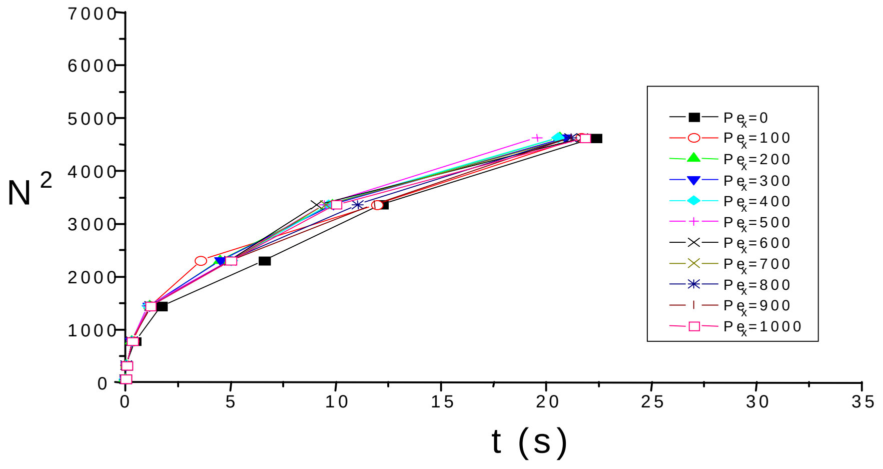

Figure 3 - Effect of N and Pe_x on computational effort- Markowitz Strategy.

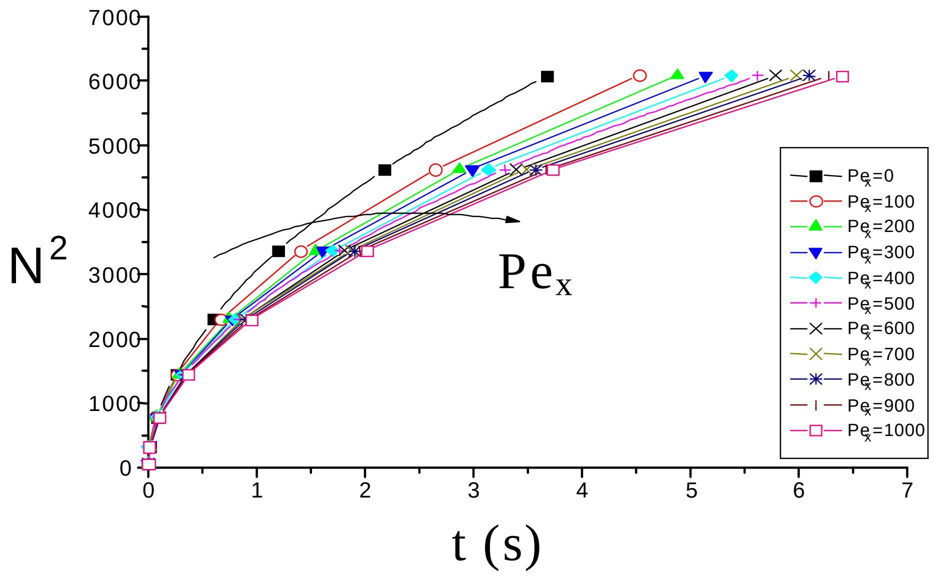

Figure 4 - Effect of *N* and *Pex* on computational effort - Doolittle Method.

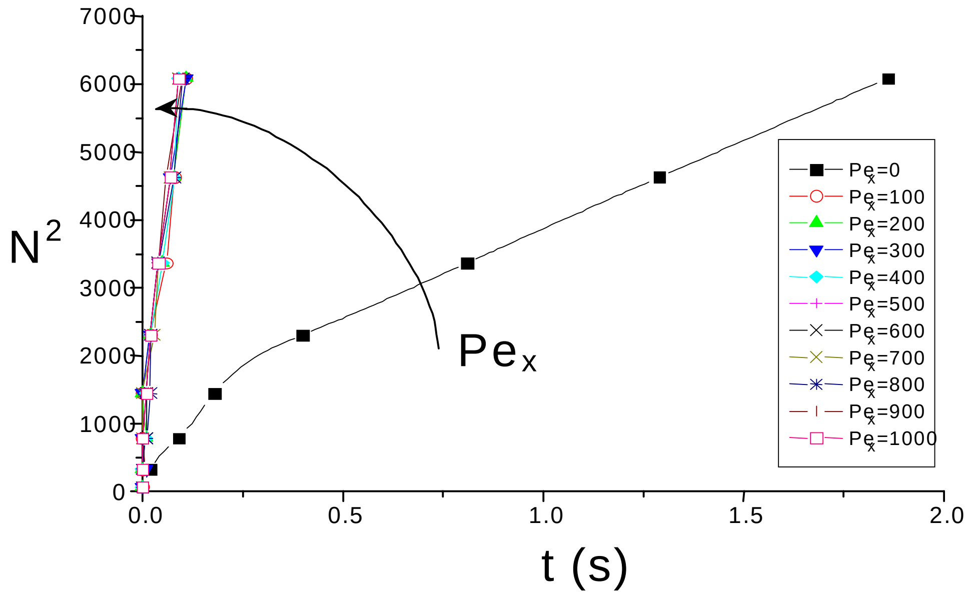

Figure 5 - Effect of *N* and *Pex* on computational effort- SIP.

4.3 Effect of numerical method. A more direct comparison among the algorithms here used is presented in Fig. 6. The SIP method of Stone (1967) produced the best performance followed by the Doolittle method with the Markowitz Strategy giving the worst results. One should mention that the system here analyzed corresponds to a simplified flow with fixed matrix coefficients and that, in real three-dimensional flow calculations, the relative advantages of one method over the others may change from the situation here observed.

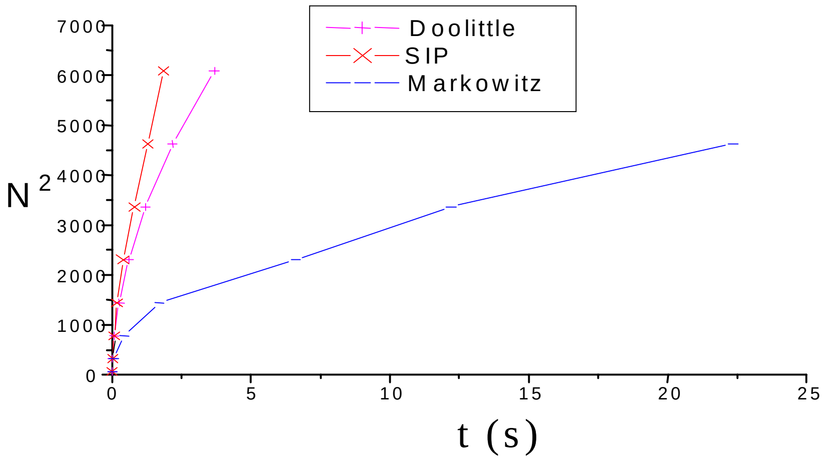

Figure 6 – Effect of numerical method on computational effort, Pe_x=Pe_y=0.

4.4 Effect of flow direction. The results shown so far considered the cases of velocity parallel to the *x*-axis. Another study presented in this work is the effect of velocity direction on the computational effort. In general, recirculating flows present velocity patterns such that both velocity components, in either direction, are of non-negligible sizes. That will, in turn, unbalance coefficients in matrix **A** depending upon flow direction.

Fig. 7 shows results for the computational time as a function of velocity angle θ (see Fig. 1). The curves are for different Pe_x when θ=0, here recalled as Pe_{xo}, and for N^2=4624 equations, that is 70 points of discretization. No conclusion about a specific behavior seems to be drawn from the curves, except that, when the velocity components are of equal size and different sign (θ=135° and 315°), the computing time is independent of Pe_{xo}. Also, flows aligned with the horizontal direction (θ=0° and 180°) needed more CPU time than those parallel to the vertical direction (θ=90° and 270°). That could be associated with positioning of boundary conditions used and with the nodal ordering for composing the matrix of coefficients. Similar results using the Doolittle method are presented in Fig. 8. Horizontal flows show a dependence on Pe_{xo} but in all other directions the time for solution is nearly independent of θ or Peclet. Here again the specific ordering of the linear equation set and the particular boundary conditions used could be the reason for such behavior. Finally, Fig. 9 shows the necessary computing time to when the SIP method is used. In this case, the minimum computational effort is when the velocity vector has both components of equal size and of positive sign (θ=45°). Also, the SI Procedure showed the least sensitive to θ and only a mild dependence on Pe_x. Comparing further Figs. 7-8-9 the SIP method presented the least overall computing time regardless of flow direction.

5. CONCLUDING REMARKS

Several numerical methods for solution of linear systems were applied to a two-dimensional convection-diffusion problem with given flow. The iterative algorithm of Stone

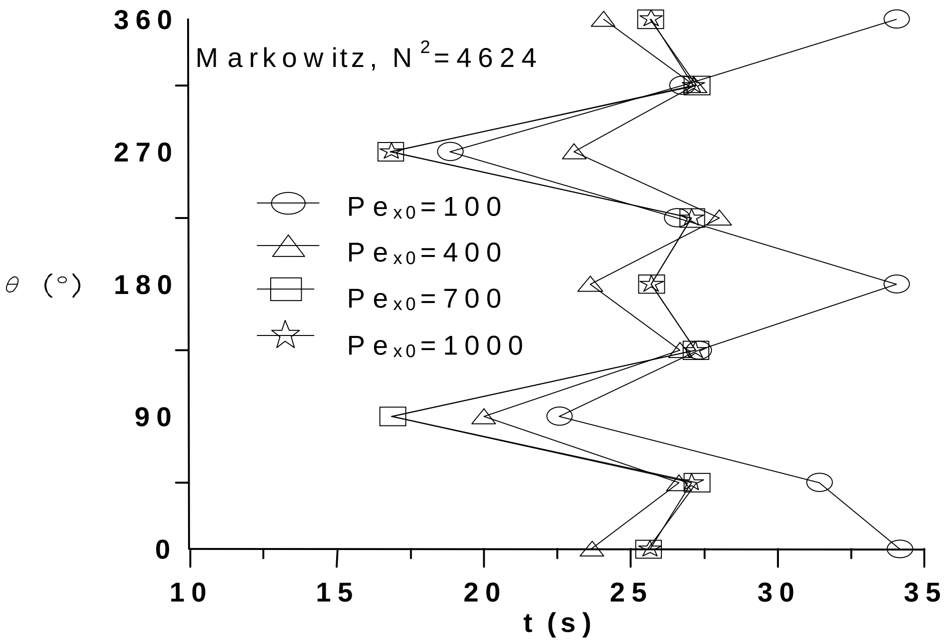

Figure 7 – Effect of θ and Pe_x on computational effort – Markovitz strategy.

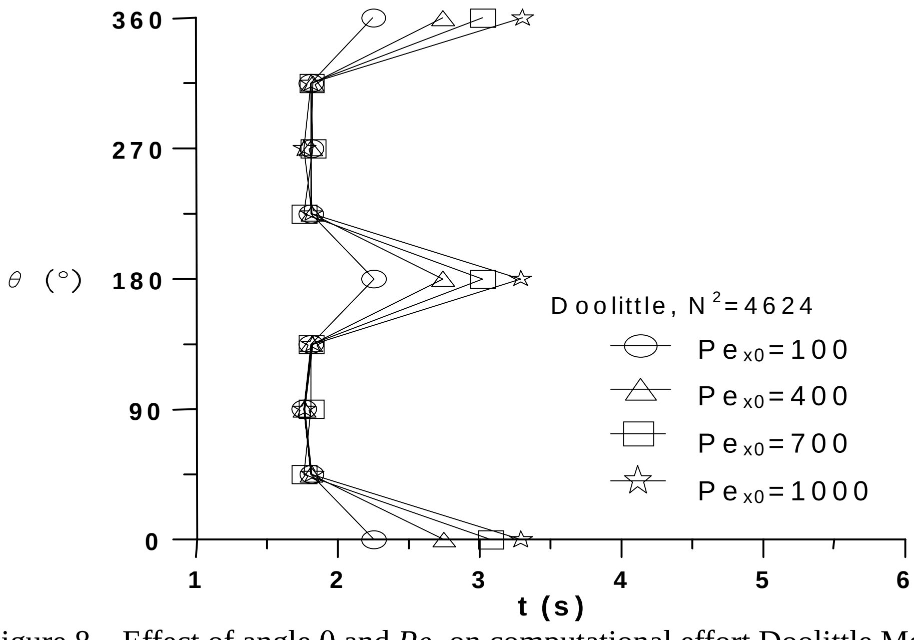

Figure 8 – Effect of angle θ and Pe_x on computational effort Doolittle Method

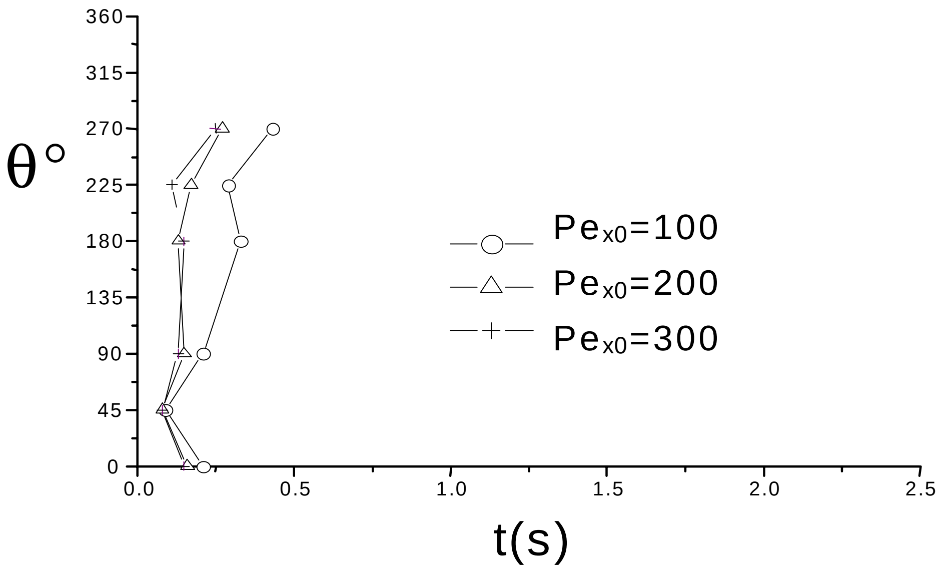

Figure 9 – Effect of θ and Pe_x on computational effort – SIP method.

uses the minimum amount of memory and showed the best performance. The direction of the velocity vector was felt with more intensity with the Markowitz Strategy.

6. ACKNOWLEDGMENTS

The authors are thankful to CNPq, Brazil, for their financial support during the course of this research.

7. REFERENCES

- Crowe, K., Fan, Y.A., Li, J., Neaderhouser, D., Smith, P., 1990, A direct sparse linear equation solver using linked list storage, IMSL Tech. Report 9006, IMSL, Houston.
- Duff, I.S., Erisman, A.M., and Reid, J.K., 1986, Direct Methods for Sparse Matrices, Clarendon Press, Oxford.
- Markowitz, H.M., (1957), The elimination form of the inverse and its application to linear programming, Management Sci., vol. 3, pp. 255-269.
- Patankar, S.V., 1980, Numerical Heat Transfer and Fluid Flow, Hemisphere Publishing Corporation.
- Rabi, J.A., de Lemos, M.J.S., 1998, The Effects of Peclet Number and Cycling Strategy On Multigrid Numerical Solutions of Convective-Conductive Problems, 7th AIAA/ASME Joint Thermcs & HT Conf., Albuquerque, NM, USA June 15-18, AIAA-98-2584.
- Raithby, G.D., Torrance, K.E., 1974, Upstream-Weighted Differencing Schemes and Their Application to Elliptic Problems Involving Fluid Flow, Comp. & Fluids, vol. 2, pp. 191-206.
- Stone, H. L., 1968, Iterative Solution of Implicit Approximations of Multi-Dimensional Partial Differential Equations, SIAM J. Num. Anl., 5, pp. 530-558