A STUDY OF PLANE FLOWS USING MICROCOMPUTERS

By A.O.Sherif[*] & I.M.Shabaka[*]

ABSTRACT

This paper is a part of a plan to explore the possibility of utilizing the economic, yet powerful, microcomputers for the analysis and education of numerical fliud dynamics' techniques.

The stream function equation has been discretized and solved with a simple line (tri-diagonal) solver using an S.O.R. iteration scheme. The solution Algorithm was programmed in FORTRAN77 for the IBM PC equipped with the Intel 8087 math-coprocessor. Some results are presented to illustrate some limitations and tips in the use of such numerical devices and machinesin the Computational Fluid Dynamics.

* Associate Professors, Department of Aeronautics, Cairo University.

1. Introduction:

Computational Fluid Dynamics' research studies and education is known to be regretably lagging in Egypt. In spite of its importance, it seems that efforts have been failing in deriving our educational institutions to catch up with the continuous progress in this field. Few efforts were successful in understanding and performing useful work, but no systematic local policy has been agreed upon in this field yet. The most important reason for this was the unavailability of computing facilities in our institutions. Some of the institutions do not have in-house computers. Others have them, but the demand is so high that their practical availability is not high enough to allow for serious CFD work to be performed. In addition, its know how that is laking. We know some of the theory but the local practical know how is questionable. The low cost, powerful micro-computers of today, we thought, should help alleviate this shortcoming.

A plan was put in the Department of Aeronautics in Cairo University to explore the feasibilty and limitations of utilizing micro- computers in serious CFD computatioin. The plan agreed upon covers the different types of flow and can be summaraized in the three well known types of flow, namely:

1. One dimensional flows.
2. Two dimensional flows under the following flow conditions:
    a) Incompressible conditions.
    b) Compressible conditions.
    c) Transonic conditions.
3. Three dimensional flows.

The present paper falls within the scope of the second stage of the plan. The plane flow stream function equation has been used in the study. Its solution was investigated using a conservative finite difference scheme on an IBM PC.

2. Problem Formulation:

The stream function equation is a powerful tool in computing plane flows. Over the potential function equation, it has the advantage of capability tohandel rotational flow effects. The equation has been successfuly solved in transonic flow applications by Hafez & Lovell [1]. It has been extended to solve three-dimensional transonic flows over wings, and wing-body combinations by Sherif & Hafez [2]. The extension to solve viscous flows has been described by Sherif [3].

The governing equation for the stream function is obtained from the definition of vorticity. The obtained equation automatically satisfies the continuity equation. Using Einestien notation [4], the equation may be written in generalized coordinates as:

$$\left(\frac{g_{11}}{\rho J} \psi_{y^2} - \frac{g_{12}}{\rho J} \psi_{y^1} \right)_{y^2} - \left(\frac{g_{12}}{\rho J} \psi_{y^2} - \frac{g_{11}}{\rho J} \psi_{y^1} \right)_{y^1} = -J\xi \quad (1)$$

where,

$$g_{ij} = \frac{\partial x^{\alpha}}{\partial y^{i}} \quad \frac{\partial x^{\alpha}}{\partial y^{j}} \qquad i,j=1,2 \qquad (2)$$

defines the metric of the transformation between the computaional plane $x^i$and the physical plane y^j,

$$J = g = | g_{ij} | \qquad (3)$$

is the Jacobian of transformation,

ρ is the fluid density,

ψ is the stream function,

ξ is the vorticity, and

ψ_{y^i} denotes $\partial\psi / \partial y^i$.

Similar expressions apply for x^i,s.

The contravariant velocity components are defined by:

$$\left.\begin{aligned} \psi_{y^1} &= -J\ V^2 \\ \psi_{y^2} &= -J\ V^1 \end{aligned}\right\} \qquad (4)$$

Using a system of orthogonal coordinates, thus: $g_{ij} = 0$ for $i \neq j$, the equations simplifies to:

$$\left(\frac{g_{11}}{\rho J} \psi_{y^2} \right)_{y^2} + \left(\frac{g_{22}}{\rho J} \psi_{y^1} \right)_{y^1} = - J\ \xi \qquad (5)$$

and in Cartesian coordinates it is:

$$\left(\frac{\psi_{y^1}}{\rho} \right)_{y^1} + \left(\frac{\psi_{y^2}}{\rho} \right)_{y^2} = - \xi \qquad (6)$$

with $$\psi_{y^1} = -\rho v \quad \& \quad \psi_{y^2} = \rho u \qquad (7)$$

In compressible flow computation, the density is variable and is determined using the energy equation. For details of the process see [1 & 2].

3. Discretization of the Equation:

Equation (6) is in conservative form. The standard method for discretizing conservative forms is to use central differencing. The discretized form may be written as:

$$\overleftarrow{\delta}_{y^1}\left(\frac{\overrightarrow{\delta}_{y^1}\psi}{\rho}\right)^{n+1}_{i-\frac{1}{2}} + \overleftarrow{\delta}_{y^2}\left(\frac{\overrightarrow{\delta}_{y^2}\psi}{\rho}\right)^{n+1}_{j-\frac{1}{2}} = -\xi \qquad (8)$$

where:

$$\overleftarrow{\delta}_{y^1} = (\psi_{i+1,j} - \psi_{i,j}) / \Delta y^1_i$$

$$\overleftarrow{\delta}_{y^1} = (\psi_{i,j} - \psi_{i-1,j}) / \Delta y^1_{i-1} \qquad (9)$$

With similar expressions for the y^2 direction.

The scheme described by (8) is second order accurate, and the resulting system of equations is benta-diagonal.

In the following section a simple line over relaxation scheme for solving the equations is described.

4. The Iteration Scheme:

A typical point relaxation scheme may be described by:

$$L [\underline{\psi}^{n+1}] = \underline{\xi}^n \qquad (10)$$

Let

$$\underline{\psi}^{n+1} = \underline{\psi}^n + \underline{C}^n$$

Therefore:

$$L [\underline{C}^n] = \underline{\xi}^n - L [\underline{\psi}^n]$$

$$L [\underline{C}^n] = - \underline{R}^n \qquad (11)$$

where:

$\underline{R}^n$ is the residual at the nth. iteration, and

$\underline{C}^n$ is the correction to the solution at the nth. iteration.

In general, the operator L, operating on the nth. correction $\underline{C}^n$, can be replaced by the appropriate operator N defined as:

$$N = (\alpha - A\,\delta^2_{y^1})\,(\alpha - B\,\delta^2_{y^2}) \qquad (12)$$

where:

$$A \simeq \left(1 - \frac{u^2}{u^2+v^2}\right), \qquad B \simeq \left(1 - \frac{v^2}{u^2+v^2}\right)$$

and is selected in such a way to ensure the stability of the scheme. (For more details on this point see [5].)

In the line scheme used in this work the y^2-term has been evaluated from the values of the stream function at iteration number n. Only a tri-diagonal system of difference equation is left to solve. This has to be solved along the y^1-direction for each y^2-level.

The values of the stream function is then corrected

using: $$\psi^{n+1} = \psi^n + \omega\, C^n \qquad (13)$$

where: ω is an over relaxation factor $1 < \omega < 2$.

The spectral radius of convergence for the scheme may be expressed as:

$$r = 1 - \alpha h^{\sigma} \tag{14}$$

where:

α is a constant,

h is a characteristic mesh size, and,

σ is a constant.

It is known that $\sigma = 2$ for Jaccobi and Gauss-Seidel methods, $\sigma=1$ for S.O.R. iterations using optimum relaxation parameter, ω_{opt}. $\omega < 1$ for the Zebra scheme used in [1, 2 & 5].

The values for α & σ are sensitive to:

- the step size h,
- the aspect ratio of the mesh $\Delta x_1 / \Delta x_2$,
- the variation of the mesh aspect ratio in the field of computation, and,
- the location of the outer boundary.

5. Results and Conclusions:

The first part of the plan, of which the present paper is but a first step is to investigate the performance of the simple solver described in the previous section when operated on micro-computers. It is well known that the convergence of such scheme, when applied to incompressible flows depends on:

- The size of the mesh and the mesh aspect ratio.
- The relaxation factor.
- The used level of precision (whether single or double).
- The amount of stretching to which the mesh is subjected (if any).

A set of computational experiments were performed to study the first three of these factors. The effect of the forth item is still under investigation. Incompressible flow over a cylinder in a uniform stream was selected as a test case. It was decided to perform the following sequence of runs:

1. For ω =1.0 , under both single and double precision arithmetic, nine grid configurations (17x17, 33x17, 65x17, 33x33, 65x33, 65x65, 17x33, 17x65, 33x65) were tested.
2. Grids 33x33, and 65x33 were then selected to test the effect of a sequence of relaxation factors: ω = 1.1, 1.2, 1.3, 1.4, 1.5, and 1.7.

Some of the important features were presented in three figures. Fig.(1) shows the convergernce histories for the first set of runs using single precision arithmetic and a relaxation factor of one (Jaccobi-scheme). The fastest convergence rate being for grid 17x17 and the convergence rate detoeriorates as the mesh size increases, something which is expected. Convergent solutions showed oscillatory behaviour after the first few iterations. It was not possible, to have converged solutions for the cases where the mesh aspect ratio ($\Delta x/\Delta y$) were greater than one (i.e., for grids 17x33, 33x65, and 17x65). This is due to the very elementary used solver. A cure is possible by introducing a corrective cycle between iterations or by using a more sophesticated solver.

Fig.(2) depicts the convergence histories for the same set computed using double precision arithmetic and with no relaxation. The undesirable oscillations are now smoothed out. Again, the grid 17x17 has the highest convergence rate and the rate of convergence deteriorates with decrease in mesh size.

Fig.(3) shows the effect of varying the relaxation parameter on the rate of convergence for the 33x33 grid. Increasing the relaxation factor increases the rate of convergence until $\omega > 1.5$, where oscillations start to appear and then the rate of convergence deteriorates. This value of the relaxation parameter is lower than the values used with potential flow solvers (ω=1.85).

Similar results were obtained using a 65x65 grid, but in this case the residuals were reduced to lower values of the order of 10^{-11}.

In all cases, the average computing time was about 0.01 sec/point/iteration using an IBM PC equipped with the Intel 8087 math-coprocessor, and a Microsoft FORTRAN77 Compiler. This is an indication of a reasonably acceptable computaion rate and an ecomomical computer cost.