Primitive Variable CFD Solver
MAE 6263

Charles O'Neill

April 29, 2002

Abstract

A 3d transient viscous flow solver based on the `ge3df` code was used to predict the flow past an NACA 0012 airfoil in air. The modified `ge3df` code uses primitive pressure and velocity variables and the appropriate finite difference continuity and momentum expressions. The results show that an unrefined finite difference grid correctly determines the overall flow patterns. However, the flow solution near the airfoil is distorted and corrupted due to the discrete steps in geometry. Grid refinement is needed.

1 Introduction

The objective is to solve a 3D Cartesian transient viscous flows. The governing equations for continuity and momentum are reviewed[1]. Next, the modified `ge3df` FORTRAN program is discussed. Then, the geometry and fluid properties for the airfoil flow problem are shown. Results are given and discussed. Finally, conclusions are made.

2 Governing Equations

The fluid flow is solved for a transient, constant-density 3D Cartesian coordinate system based on velocities and pressures. Finally, the flow relationships are transformed to discrete representations for input into a computer routine.

2.1 Continuity

Continuity for a 3D Cartesian coordinate system is given as,

$$\frac{du}{dx} + \frac{dv}{dy} + \frac{dw}{dz} = 0$$

Continuity is enforced by adjusting the local pressure. The cell velocities are adjusted by a pressure correction which is applied at the cell faces. The velocity correction is:

$$u_{i,j,k} = u_{i,j,k} + \Delta t \Delta p / \Delta x$$

$$u_{i-1,j,k} = u_{i-1,j,k} - \Delta t \Delta p / \Delta x$$

$$v_{i,j,k} = v_{i,j,k} + \Delta t \Delta p / \Delta y$$

$$v_{i,j-1,k} = v_{i,j-1,k} - \Delta t \Delta p / \Delta y$$

$$w_{i,j,k} = w_{i,j,k} + \Delta t \Delta p / \Delta z$$

$$w_{i,j,k-1} = w_{i,j,k-1} - \Delta t \Delta p / \Delta z$$

where the error in continuity is given the value D. Thus, the pressure change required to enforce contintuity is

$$\Delta p = -D/(2\Delta t(1/\Delta x^2 + 1/\Delta y^2 + 1/\Delta z^2))$$

2.2 Momentum

Momentum is enforced by the conservative form of the Navier Stokes equation. The continuous forms of the Navier Stokes equation are:

$$\frac{du}{dt} + \frac{du^2}{dx} + \frac{d(uv)}{dy} + \frac{d(uw)}{dz} = -\frac{dP}{dx} + \nu \nabla^2 u$$

$$\frac{dv}{dt} + \frac{d(uv)}{dx} + \frac{dv^2}{dy} + \frac{d(vw)}{dz} = -\frac{dP}{dy} + \nu \nabla^2 v$$

$$\frac{dw}{dt} + \frac{d(uw)}{dx} + \frac{d(wv)}{dy} + \frac{dw^2}{dz} = -\frac{dP}{dz} + \nu \nabla^2 w$$

3 Computer Code

The `ge3df` FORTRAN computer code was modified to solve the 3D Cartesian transient governing equations. The general program steps are given below.

1. **Problem Initilization:** Geometry and variable initialization occurs. General iteration and problem constants are read and displayed. Initial flow values are computed. Coordinates for the airfoil boundaries are calculated and tagged.

2. **Initial Boundary Conditions:** The initial boundary conditions are created for the airfoil and the outer solution domain.

3. **Time Step Loop:** The program enters the time step loop.

4. **Momentum:** Velocities and viscous terms are computed to enforce momentum conservation. The velocity values are updated to the next time step.

5. **Boundary Conditions** The overall boundary conditions for both the computational domain and the airfoil are determined. Boundary conditions are enforced on velocity.

6. **Continuity Iteration:** Continuity is enforced by iterating pressure and velocity terms. A Δp is calculated from the velocity divergence. The local velocites are updated from this Δp.

7. **Enroute Output:** The iteration numbers and residuals are calculated and printed out.

8. **Repeat Time March:** The time march is repeated.

9. **Output Results:** Final pressure and velocity values are output.

4 Problem Geometry

The problem geometry consists of a 2D NACA 0012 airfoil. The fluid is air at 20 degrees Celcius. Airfoil coordinates are calculated from the four digit NACA airfoil expression given in Abbott and von Doenhoff[2]. The airfoil extends across the entire flow field in the z direction. A typical 150 by 100 geometrical mesh is shown in Figure 1. This figure shows a two dimensional slice; however, the entire flow domain is three-dimensional.

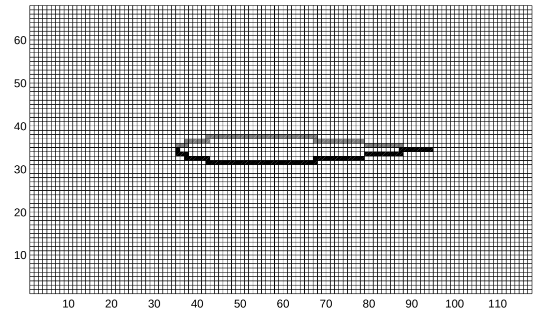

Figure 1: Solution Grid

5 Results and Discussion

The computer program was modified and run as previously described. All computations were performed on a NACA 0012 airfoil shown above. The computations were restricted to an angle of attack of zero degrees.

The flow solver was run until the residuals leveled off. Figure 2 shows the residuals versus the number of iterations. The residuals spike downward at regular intervals, however the power of maximum residual envelope decreases linearly.

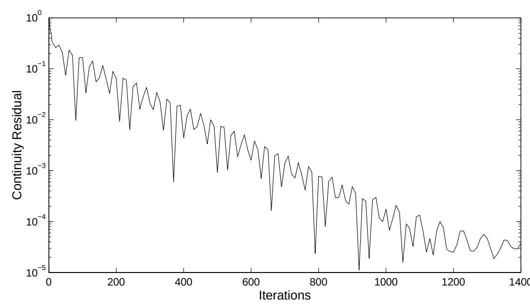

Figure 2: Continuity Residual vs. Iteration

The velocity distribution for the NACA 0012 is given in Figure 3. The overall distribution appears correct. There is the expected stagnation point forming off the leading edge. The flow is accelerating as it moves past the high curvature parts of the

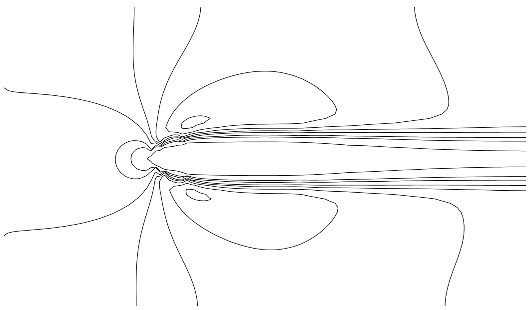

Figure 3: Velocity Distribution

airfoil. Finally, a wake is developed aft of the airfoil. However, the wake appears too large. Perhaps the flow is separating as it moves past the discrete steps in the geometry. Increasing the grid resolution required vastly more computational power. Worse still, the same boundary normal problem still exists with a fine grid.

The pressure distribution is given in Figure 4. Once again, the flow solver seems to find the general trend, however the near field distribution of pressure is distorted. The chordwise pressure field

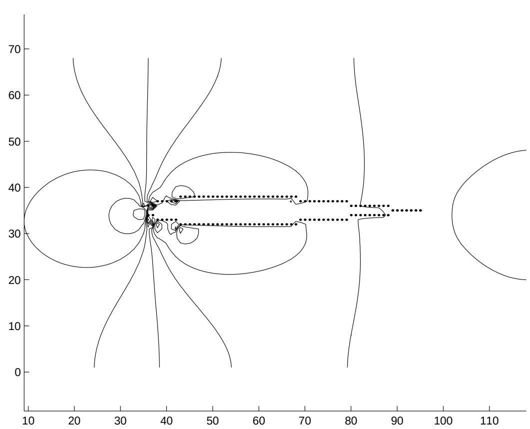

Figure 4: Pressure Distribution

on the airfoil surface is shown in Figure 5. The general trend in pressure distribution is correct but the magnitudes and shapes are incorrect. The leading edge pressure distribution is badly corrupted by the coarse geometrical mesh. The solver fails to find the maximum surface pressure due to “flow separation”

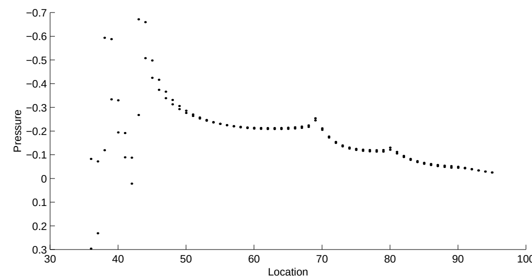

Figure 5: Surface Pressure Distribution

effects of the mesh. The discrete steps around the 3/4 chord are visible in the surface pressure distribution.

Computations were tried with a modified mesh geometry. Increasing the grid resolution helped the flow solution; however, local separations were still present. For some solutions, the separation was so severe that the airfoil started shedding alternating vorticies! It appears that the largest problem with the rectangular finite difference method is modeling the correct boundary conditions. This particular finite difference program is unable to incorporate boundary normal vectors other than north, south, east, west.

6 Conclusions

A modified version of the`ge3df` flow solver was used to predict the transient 3D viscous solutions around an airfoil. The flow solution work well away for any discrete geometric disturbances. Around the airfoil boundary and especially around the leading edge, the finite difference grid can not handle large flow gradients. Additionally, the coarse leading edge combined with the discrete rearward steps seems to separate the flow. In short, an effective and efficient CFD routine for airfoil prediction needs a refined grid.

References

[1] Lilley, D. G., *Computational Fluid Dynamics. Vol I-III*, Stillwater, OK, 1992.

[2] Abbott and von Doenhoff, *Theory of Wing Sections*, Dover, 1959.