CFD: 2-D Inviscid Flow Simulation

Michael McMahon and Saleh Elmohamed

NPAC, Syracuse University,

USA.

Abstract

This code simulates 2-D inviscid flow through an axisymmetric nozzle. The simulation yields contour plots of all flow variables, including velocity components, pressure, mach number, density and entropy. The plots show the location of any shock wave that would reside in the nozzle. The code finds the steady state solution to the 2-D Euler equations, seen below.

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{E}}{\partial x} + \frac{\partial \mathbf{F}}{\partial y} = \alpha \mathbf{H} \tag{1}$$

Here $\mathbf{U} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ e_t \end{pmatrix}$, $\mathbf{E} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ (e_t + p) u \end{pmatrix}$, and $\mathbf{F} = \begin{pmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ (e_t + p) v \end{pmatrix}$.

The source vector $\mathbf{H}$ is zero for this case.

The High Performance Fortran (HPF) code uses a 4 stage Runga-Kutta time-stepping algorithm and a finite volume central-difference technique to find the solution. A numerical dissipation model is employed to dampen any spurious oscillations and prevent the solution from blowing up in the presence of shock waves (Jameson et al, 1981). If the inlet and exit conditions are supersonic, the code requires about 1800 iterations to complete execution.

However, if subsonic boundary conditions are employed, the number of iterations required jumps to nearly 6000, as new errors are introduced into the domain by these conditions (Dang 1997). The output file info.dat contains the data for contour plots.

Implementation and Performance Issues

The code uses many `FORALL` statements. This construct is succint and guarentees no dependencies, so the compiler doesn't get confused. The arrays are distributed along their longest dimension, which corresponds to the lengthwise direction of the nozzle. Other distributions may be more efficient, depending on the number of processors available. However, this distribution allows any number of processors to be used. The `HPF` statements `DISTRIBUTE` and `ALIGN` are used to execute this mapping.

The code does require some communication, especially in the artifical dissipation routine, where second and fourth derivatives are computed. Computing the inlet and outlet boundary conditions creates a small load imbalance in computation. These two facts prevent perfect scaling, but excellent speedup can still be attained.

Documentation

Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes", AIAA 14th Fluid and Plasma Dynamics Conference, paper no. AIAA-81-1259, June 23-25, Palo Alto, California, 1981.

Dang, Thong Q. "Monograph in Computational Fluid Dynamics." Department of Mechanical, Aerospace and Manufacturing Engineering. Syracuse University, 1997.