Domain Decomposition Finite Volume Method for Three-dimensional Inviscid Flow Calculations

Tiejun Nie and Jianhu Feng

1 Introduction

For a fighter aircraft, the flow quality at the inlet can have a large impact on the performance of the propulsion system. It is obvious that the flow quality depends not only on the shape of the inlet, but also on the shape of the forebody and the inlet mounting position. So the integrated solution of internal and external flows is required.

Solutions of the Euler equations can give a more physical representation of inviscid subsonic, transonic and supersonic flowfields than potential flow methods. Some important phenomena such as the vortex in a S-shaped inlet, separation induced by shock, can be nicely modeled with the use of Euler equations. Furthermore, solutions of Euler equations are considered as a stepping stone before the solution of time averaged Navier-Stokes equations can be attempted.

However, there are many difficulties to solve the 3-D Euler equations for complex geometries, such as grid generation, computer memory size, speed limitation and so on. This paper presents an Euler code designed for integrated combinations. The code is based on a finite volume Runge-Kutta method [1, 2]. In order to relieve the computer memory requirement, a domain decomposition technique is employed. This also makes the mesh generation much easier. Iterations are performed for each zone in turn. A newly-designed model of a fighter forebody-inlet combination and a missile model have been used to validate the code. Satisfactory results have been obtained and the solution cost is affordable.

[1] Department of Applied Mathematics Northwestern Polytechnical University, Xi'an, Shaanxi, 710072, P.R.China

Domain Decomposition Methods in Sciences and Engineering, edited by R. Glowinski *et al.*

2 Numerical Algorithm

The finite volume scheme is derived from the integral form of the Euler equations

$$\frac{\partial}{\partial t}\int_{\Omega}\vec{W}d\Omega + \oint_{\partial\Omega}\vec{F}\cdot\vec{n}ds = 0 \tag{1}$$

where Ω denotes a fixed region with boundary $\partial\Omega$ and outward normal $\vec{n}$, $\vec{W}$ represents the vector of conserved quantities, and $\vec{F}$ is the corresponding flux tensor,

$$\vec{W} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{bmatrix}, \quad \vec{F} = \begin{bmatrix} \rho u\vec{i} + \rho v\vec{j} + \rho w\vec{k} \\ (\rho uu + p)\vec{i} + \rho uv\vec{j} + \rho uw\vec{k} \\ \rho vu\vec{i} + (\rho vv + p)\vec{j} + \rho vw\vec{k} \\ \rho wu\vec{i} + \rho wv\vec{j} + (\rho ww + p)\vec{k} \\ (e+p)u\vec{i} + (e+p)v\vec{j} + (e+p)w\vec{k} \end{bmatrix} \tag{2}$$

where ρ and p are the fluid density and pressure, u, v, w are the Cartesian velocity components, $\vec{i}, \vec{j}, \vec{k}$ denote the unit vectors of the Cartesian coordinate system, and e, the total energy linked with other variables by the equation of state

$$p = (\gamma - 1)(e - \frac{1}{2}\rho(u^2 + v^2 + w^2)) \tag{3}$$

γ is the ratio of specific heats.

The computational domain Ω is divided into hexahedral cells. Values of the dependent variable $\vec{W}$ are defined at cell centers. A discrete approximation to the spatial terms yields

$$\frac{d}{dt}(V_{i,j,k}\cdot\vec{W}_{i,j,k}) + \vec{Q}_{i,j,k} = 0 \tag{4}$$

where $V_{i,j,k}$ is the (i,j,k) cell volume. This equation represents the discrete flux balance. The discrete flux term $\vec{Q}_{i,j,k}$ represents the net flux out of the cell,

$$\vec{Q}_{i,j,k} = \sum_{l=1}^{6}(\vec{F}\cdot\vec{S})_l \tag{5}$$

where $\vec{S}_l(l = 1, 2, \cdots, 6)$ denotes the lth surface vector of the (i,j,k) cell, and the flux $\vec{F}_l$ are evaluated at the lth surface of the (i,j,k) cell. It is usually calculated using the averages of the quantities $\vec{W}_{i,j,k}$ from adjacent cell centers.

The present finite volume spatial discretization reduces to a central-difference scheme which is formally of second-order accuracy for smooth grids. In order to suppress its well-known tendency for odd-even points decoupling, to capture shocks and to minimize pre- and post-shock oscillations, an adaptive dissipation term $\vec{D}_{i,j,k}$ is added to the system

$$\frac{d}{dt}(V_{i,j,k}\cdot\vec{W}_{i,j,k}) + \vec{Q}_{i,j,k} - \vec{D}_{i,j,k} = 0' \tag{6}$$

This dissipation formulation uses blended second and forth differences in each of the three parametric directions [1].

To integrate the system (6) of ordinary differential equations, the present code uses Runge-Kutta time stepping schemes [1, 2]. Several convergence acceleration techniques like local time stepping, enthalpy damping and implicit residual smoothing are adopted [2].

3 Grid System and Flow Solution Coupling

In the present approach, several computational zones are used to model integrated flowfield. For example, zone (I) is the subdomain forward of the inlet highlight; zone (II) is the subdomain of the internal inlet and diffuser; zone (III), the external region aft of the highlight. The computational grid system of each zone is generated independently by means of TTM [3].

The flow solutions are execute on each zone in turn. Separate solutions are coupled by interpolating the fluxes across the interface between adjacent zones during the in two steps. First, for each zonal boundary cell, a ghost point is introduced in the neighboring zone. The ghost point is located in the plane formed by the neighboring zone boundary cell centers but has the same y, z values as the present zonal boundary cell centers. The flow values at ghost points are determined by interpolation using the values of neighboring zone. The second step evaluates the fluxes across the interface by interpolation using values of the point of the present cell and its ghost point.

The treatment of other boundary conditions is the same as that in the literature [4].

4 Flow Solution Algorithm

The flow computations are initiated by assuming uniform in coming flow quantities. If this coming flow is supersonic, then the flow field inside the inlet is initiated with a subsonic uniform flow assuming there is a normal shock located right at inlet entrance. After that several iterations have been performed for one subdomain, the flow values are updated and the flow solution is stored in a data file. Then the computation shifts to another subdomain. A sequence of iterative updates for all subdomains is referred to as a 'cycle'. Usually hundreds of cycles are required to achieve a satisfactory, converged solution.

5 Numerical Results

The first example is a forebody-inlet combination model. The flight conditions are: $M_\infty = 2.047$, angle of attack $\alpha = 0^0$, and the engine mass ratio $\bar{m} = 0.94$. The S-shaped inlet is a rectangle at the entrance and changes into circle at the exit. Even though the grid is very coarse (13 × 17 × 15 for zone (I), 9 × 23 × 15 for zone (II), and 43 × 19 × 10 for zone (III)), the result agrees with the design requirement.

The second example is a total missile-inlet combination model. The computational domain is divided into eight subdomains. The total grid number is 38495. The following

flight conditions have been modeled: 0 $M_\infty = 2.0, \alpha = 0^0, \bar{m} = 0.98$
$M_\infty = 2.0, \alpha = 1^0, \bar{m} = 0.98$
$M_\infty = 2.0, \alpha = 2^0, \bar{m} = 0.98$
This result is also very satisfactory.

REFERENCES

[1] Jameson A. (1985) Numerical Solution of the Euler Equations for Compressible Inviscid Fluids. in*Numerical Methods for the Euler Equations of fluid Dynamics,* SIAM, 199-245.

[2] Turkel E. (1984) Acceleration to a Steady State for the Euler Equations, ICASE Report No. 84-32.

[3] Thompson J. F., Thames F. C. and Mastin C. S. (1974) Boundary-fitted Curvilinear Coordinate Systems for Solution of Partial Differential Equations Containing any Number of Arbitrary Two Dimensional Bodies, *J. Comp. Phys.,* Vol. 15: 299-319.

[4] Feng Jianhu (1994) *On Numerical Simulation of Three-Dimensional Aircraft Inlet Systems,* Ph. D. Dissertation, Northwestern Polytechnical University.