A Structured Grid Method in Simulating a Flow Around Supersonic Transports

Shunji ENOMOTO

Aeroengine Division
National Aerospace Laboratory

eno@nal.go.jp

Introduction

In the SST development project, it is expected to perform numerical simulations of the configuration of the airframe and the engine(Fig.1). But it is not easy task to solve these flow flowfield using existing CFD codes, because the geometry of the flow passage is very complicated. Development of a CFD code that makes easy to solve complex flowfield is expected.

UPACS is a CFD code development project , which is performed in National Aerospace Laboratory since 1997. A structured grid, multiblock, multiprocessor CFD solver is under development as the first CFD solver of this project. The basic idea of the code and some points of issue will be discussed in this report.

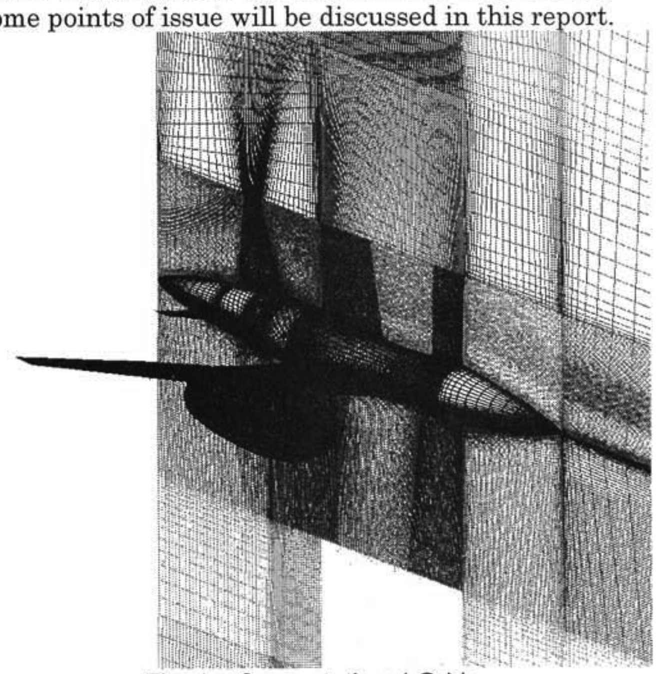

Fig. 1 Computational Grids of the SST configuration.

Numerical methods for complex topologies

The topologies of flow passages in practical problems are often very complicated. The task of making body-fitted structured grids and computational programs for them is often very troublesome. Today, the most difficult problem is the grid generation. Making body-fitted structured grid will not be automated in the near future. There are some good commercial grid generation tools, but it is still not very easy task to make grids using these tools.

Unstructured or overset grid methods are expected to be the solution for the problem, and they seem promising. Using these methods, it will be easy to make a body-fitted grid around the complex configuration, a solution adapted grid, and a moving grid.

However, comparing to these other methods, structured grid methods still possess some advantages. It is easier to implement higher order differential schemes and turbulence models in structured grid systems. It also has the advantage of efficiency and accuracy, especially for the case that the grid has good orthogonality, since most of higher order differential schemes and turbulence models are originally developed on a structured grid.

When the accuracy of the solution has the first priority, a structured grid method may be a good choice. But we have to devise some way to solve the problem of geometry complexity.

Multiblock

To apply body-fitted structured grids to a complicated flow passage, a multi-block decomposition method has to be used. A single-block skewed grid may be converted into multi-block grids that have good orthogonality. It also provides us a straightforward way to use multi processor computers. The disadvantage is that we have to transfer data between blocks, so the programming for the multiblock grid system is not very simple.

In order go maintain the accuracy on block boundaries, grid points on the block boundary have to be one by one. The orthogonality of the grids has to be good, too. To achieve these requirements, we have to use unstructured block connection. Figure 2 shows the examples of structured connections and unstructured connections.

Structured connection

The connection between blocks categorized into two types, structured and unstructured (Fig.2).

The structured connection can be defined as the connection where four blocks meet at one point. The unstructured connection can be defined as the connection other than the structured connection, for example, the connection where three or five blocks meet at one point.

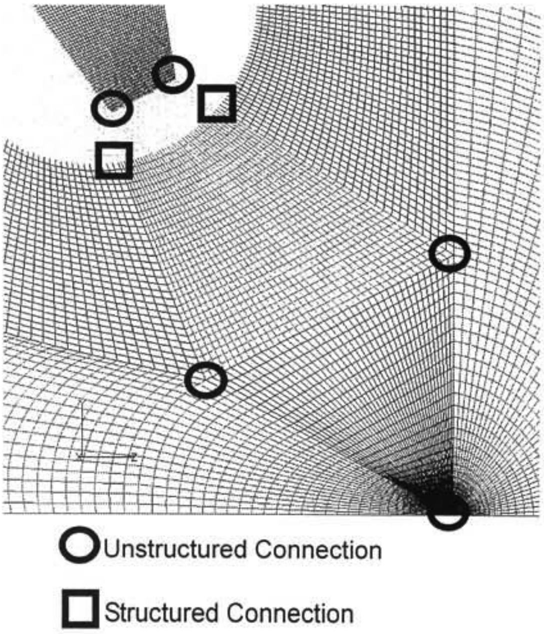

Fig.2 Structured connection
and unstructured connection

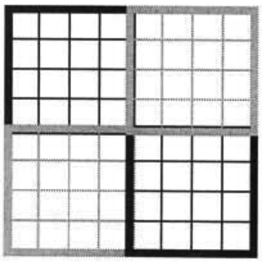

Fig.3 Structured connection

Figure 3 shows the typical example of structured block connections. Since the accuracy of the solution has the first priority, we do not want to let the block connection deteriorate the accuracy of the solution. In order to maintain the accuracy on the boundary, some overlapped region are defined and the data on that are transferred each other. The width of the overlapped region depends on the length of the stencil of the numerical scheme.

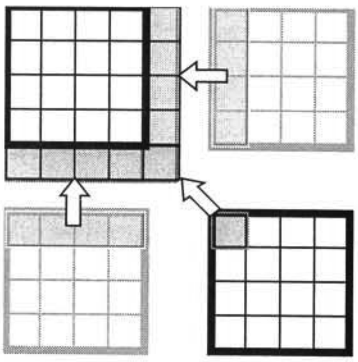

Fig.4 Overlapped region and data transfer

In order to maintain the accuracy near the block connection, each block have some overlapped points, and data of the adjacent block are transferred each other. For the accuracy of viscous terms, corner data have to be transferred, too(Fig.4). In three dimension, there are three type data transfer such as , 6 faces, 12 edges and 8 corners.

Unstructured connection

Figure 5 shows the typical example of unstructured block connections.

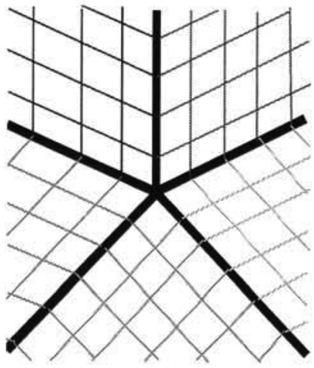

Fig.5 Unstructured connection

For the unstructured connection case, we cannot define the value of the corner point of the overlapped region. So we have to use some interpolation, and it makes the accuracy of viscous term a little worse.

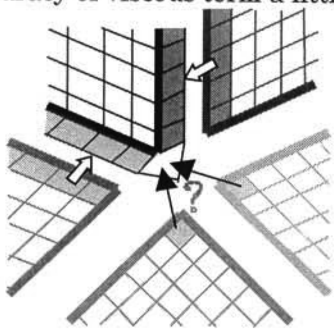

Fig.6 Data transfer
for unstructured-like connections

Unstructured-like block

For some cases, we may not be able to fill all geometry with hexahedron blocks. For such case, the unstructured-like block are used. Figure 7 shows an example of such blocks.

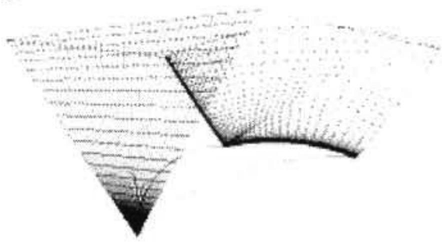

Fig.7 Unstructured-like block

In unstructured-like block, an edge line may degenerate into a point, or a surface plain may degenerate into a line. The concentrated grid might limit the time step and the accuracy of the computation.

Requirement for a multiblock solver

The requirement for a multiblock solver is that the result is not deteriorated by the block decomposition.

1. The accuracy is not affected on the structured connections, no matter how long the length of the stencil of the numerical scheme.
2. The accuracy around the unstructured connection should be maintained as much as possible.
3. The time accuracy of the unsteady calculation is not affected by the block decomposition.

In order to satisfy these requirements, data transfer between adjacent blocks has to be performed frequently. Since we may use multi processor computers, the programming of data transfer have to be written in some multi-processing library, such as MPI. Compared with traditional single processor programming, the multi processor programming is not very easy and flexible task

Data transfer

Basic equation of a conservation law can be written in this generalized form:

$$\frac{\partial}{\partial t}\int Q dV + \int F dA = 0$$

The discretized form of the basic equation can be written as like this form:

$$\Delta Q = \Delta t \cdot RHS(Q)$$

Generally, the flow chart of the CFD solver can be written like figure 8.

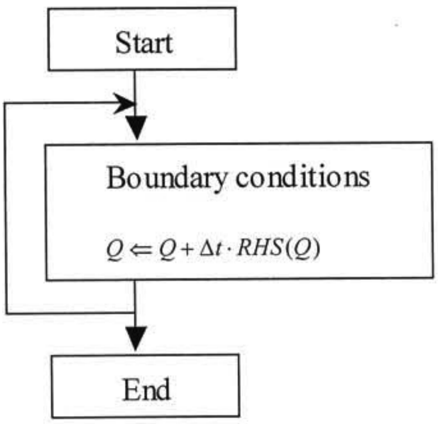

Fig. 8 Flow chart of CFD solver

For multiblock program, data transfer between blocks have to be performed somewhere in the flow chart. First, we thought that all we have to do is to transfer data between blocks one time per one iteration (Fig.9).

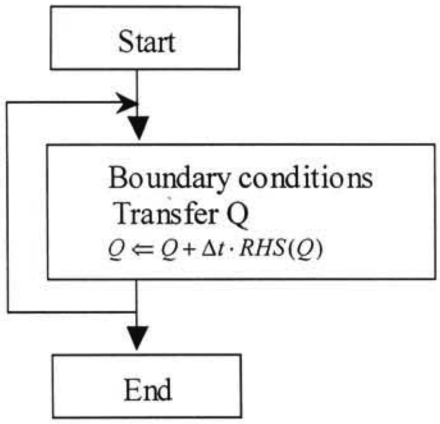

Fig.9 Flow chart of simple multiblock program

But soon we found that was not enough. We need transfer not only flow data(Q), but also grid metrics. If we use Runge-Kutta method, we need to transfer physical variables several times in one iteration. When we use a turbulence model, some other data have to be transferred between blocks, too.

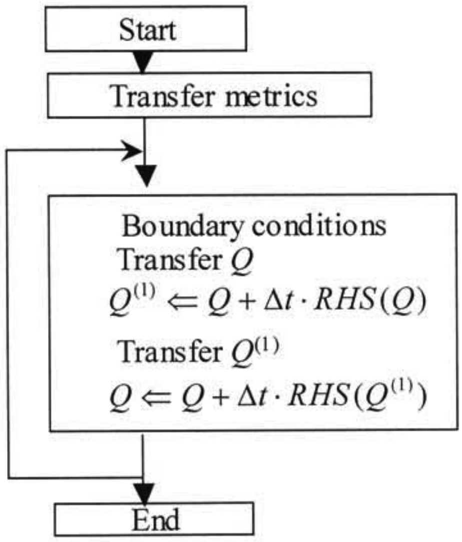

Fig.10 Flow chard of practical multiblock program

UPACS

Under these circumstances, UPACS has been developing. We think that there are two requirements for UPACS. One is that it to be a tool for CFD researchers. The other is to be a general-purpose flow solver for fluid-machine designers.

Multi-block, multi-processor programs tend to be very complicated. On the other hand, computer programs for researchers need to be simple, because the programs are continuously modified in order to try new ideas. To satisfy these requirements, UPACS has been developing as a platform or an environment for CFD tools. It has following features:

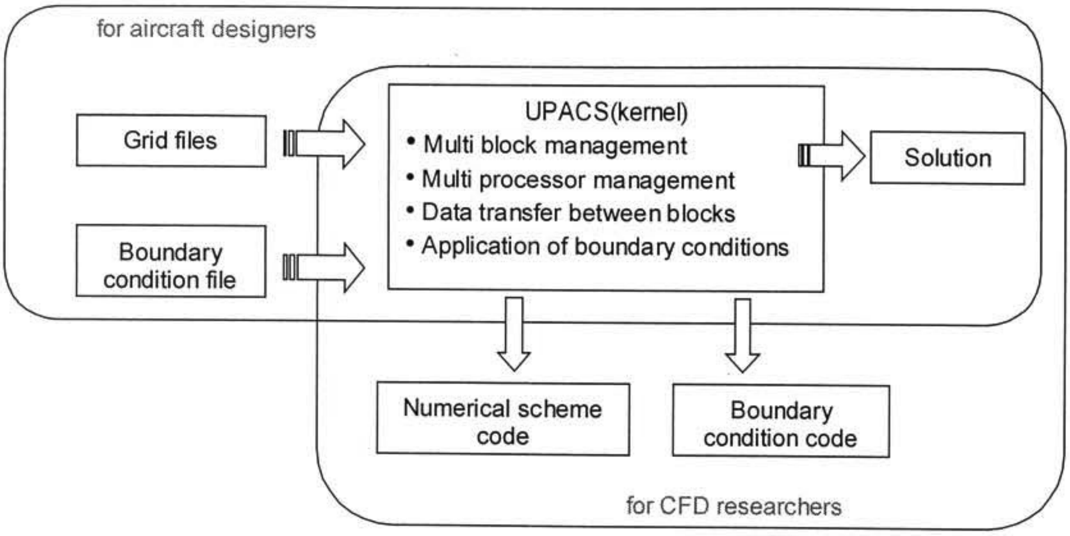

Fig. 10 UPACS environment

1. The multi-block, multi-processor part of the program and the CFD part are separated from each other. It makes easier for CFD researchers to modify and improve CFD part.
2. Existing single-block CFD solvers can be introduced into the program easily. In other words, CFD researcher writes CFD solvers for a single block and do not have to care about multiblocks. UPACS will apply the solver for all blocks.
3. CFD researcher writes generalized boundary condition routines. UPACS will apply the boundary condition to appropriate boundary.
4. Data transfer between blocks is performed automatically. CFD researcher only has to specify the data name. UPACS will transfer data between blocks.

Current CFD solver in UPACS is originally from already existed single block CFD solver. It is based on compressible Navier-Stokes equation, and has some options.

- Convective term: Roe scheme, AUSMDV, etc.
- Time integral: Runge-Kutta, MFGS, etc.
- Turbulence model: Baldwin-Lomax, Spalart-Allmaras

The program is made for general-purpose as much as possible. Many control parameters and control files made it possible. Since these control files are generated automatically, CFD designers are free from complicated tasks to apply the program to new flow geometry. Therefore, UPACS for aircraft designers is easy to use, like commercial CFD codes. Users do not need to make or edit a program to apply each problem. Computational grid and some parameter files are only required to perform CFD analysis.

Conclusion

In the SST project, UPACS is going to be used to simulate complicated flow phenomena around airframe and engine integration with higher-order turbulence models. It will be used to evaluate the configuration of SST experimental models. Through these activities, we have many opportunities to compare experimental results and the simulations. So it also will be a test bed of turbulence models.

The latest version of UPACS has succeeded inviscid calculation around a SST configuration with 105 blocks.

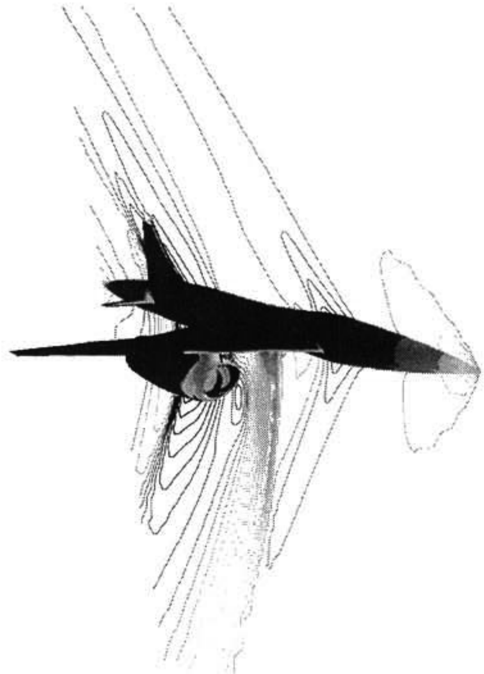

Fig. 11 Pressure contour