VISUALIZATION FOR COMPUTATIONAL FLUID DYNAMICS AND THE COMPARISON WITH EXPERIMENTS

AIAA-90-3031-CP

Yoshiaki Tamura • and Kozo Fujii †

The Institute of Space and Astronautical Science,
Yoshinodai 3-1-1, Sagamihara
Kanagawa, 229 Japan

Abstract

Three aspects of the postprocess of computational fluid dynamics (CFD) and their requirements are described. One aspect is visualizing flow fields as one method of processing numerous data. Various ways of displaying various flow functions and user-friendly man-machine interface is emphasized in this aspect. The second aspect is illuminating the physics of the computed flow fields. Streamline tracing is an example for this purpose. The third aspect is validating computer simulations. Emphasis is made on the method to simulate the process of experimental visualizations for computed results in order to compare the results with pictures obtained in experiments. It is concluded that the real postprocess of the CFD is not only to visualize computed flow fields but also to "cook" the computed flow field to illuminate the flow characteristics, and generating nice and beautiful pictures without flow physics is out of the CFD research.

1. Introduction

In recent years, the computational fluid dynamics (CFD) have made a great progress by virtue of the development of supercomputers. Even three-dimensional Navier-Stokes equations can be solved within reasonable time and accuracy, and still more accurate and reliable methods for complex practical problems are being pursued nowadays.

The process of flow field simulation in the CFD can be decomposed into three stages. The first stage is so-called preprocess and in general it requires a grid generation. The second stage is main process, in other words, solving the governing equations, using a discretization procedure. The last stage is postprocess where the computed results are analyzed and certain form of required data is output. Visualized image is often used here[1-3].

All these three stages are important. However compared to the discussions on the main process, pre- and postprocess have not been well discussed. There may be not much to be discussed so far as simple body configurations and two-dimensional steady flow fields are concerned. Considering the recent trend of the CFD to treat more complex flow fields, we have to discuss more about the development of good pre- and postprocess.

To survey the history of the postprocess of the CFD, the first and one of the famous postprocessor may be the one called "PLOT3D", followed by "GAS" and "RIP", developed at NASA Ames Research Center[4-5]. The "PLOT3D" interactively plots contour lines and surfaces of a lot of flow functions as well as velocity vectors and streamlines using grid data and raw computed results. It was developed on VAX using the graphics library called "DISSPLA" initially and converted to the IRIS workstation later. The "GAS" shows animations on a display of the workstation based on the pictures made by "PLOT3D". The "RIP" is a special particle path tracer, which traces particles on a supercomputer and displays image on a workstation display.

Although these programs are very useful for postprocess of the CFD results, they may not satisfy some of the requirements necessary for the postprocessor. The postprocess of the CFD has several aspects. One aspect is to visualize the data as one method of processing numerous data. Visualization is a useful tool to understand the entire flow field represented by the huge numbers of data and the "PLOT3D" might be categorized in this group.

The second aspect is to illuminate the physics of the computed flow fields. For this purpose, functions to extract the features of flow fields are necessary as well as those of the first aspect. One of these functions is streamline (particle path trace) and there are more functions required. For example, Bitz and Zabusky[6] recently developed a new concept of postprocessor named "David" and they try to extract the feature of the calculated flow fields by reducing the dimensions, tracing the extrema of a certain flow function and taking digital data (function values) from the pictures. Their concept is different from "PLOT3D" and interesting even though "David" currently treats only time-dependent two-dimensional simulations.

The third aspect is validating computer simulations. It cannot be separated from the previous two but the main feature here is comparison of computed results with results of other computations and/or experiments. It is easy to compare two computed results by displaying their pictures at the same time. In order to compare computed results with experiments, there are much to consider. Especially when the experimental data is visualized image such as schlieren photograph, image of general contour plots such as density contour is not enough for the comparison. For instance, density contour plots of the computed results are often compared with the schlieren picture of the experiments but it is not fair and accurate because (in the case of two-dimensional flow) schlieren photograph shows not the density itself but the density gradient perpendicular to the knife edge. In three-dimensional flow, the situation is more complicated. It is necessary to simulate the visualization processes in experiment in order to compare computation with experiment.

As described above, the real postprocess of the CFD is not only the visualization but interpretation of computed

• Research Associate.
† Associate Professor Member AIAA.

results. There are still many things to be discussed for the good postprocessor of the CFD. The "PLOT3D" was designed more than ten years ago. What have been developed in this field since then ? The most important one is the advent of graphics workstations. The graphics workstations have window systems and graphics libraries with advanced computer graphics (CG). The window systems offer user-friendly environments such as menu selections, mouse input and multi-windows. The advanced CG enables drawing three-dimensional objects easily. The graphics workstations are thus strong tool for the postprocess. There seems to be a tendency, however, to generate more beautiful pictures and animations on the video without considering flow physics. Although the current CG techniques can generate even "photo-realistic" images, it is not essential for the CFD. The "photo-realistic" images do not necessarily give better interpretations of flow fields than less beautiful pictures with simple CG techniques. In addition, it takes a long time to generate such beautiful image and it spoils the responsiveness of the process, which is one advantage of the workstations. The CG techniques should be carefully used for the postprocessors of the CFD.

Developing a postprocessor is not a main task of the CFD researchers. However, since no available postprocessors like the "PLOT3D" was available in Japan, a postprocessor running on a graphics workstation was developed by the present authors[7]. It was designed to help the CFD researchers understanding flow physics rather than just visualizing the computed results. It is also designed to have user-friendly man-machine interface (MMI) with the window system of the workstation.

In the present paper, the three aspects of the postprocess of the CFD and their requirements are described using the present authors' postprocessor as an example of a possible postprocessor on a graphics workstation. The general functions of postprocess and user-friendly MMI are described in the next section. As an example of extracting features of flow fields, the methods to compute streamlines are discussed in the third section. In the fourth section, the methods to simulate the experimental visualizations on the computed results are presented.

2. Basic Requirements for the Postprocessor

The main purpose of the postprocess is to interpret the computed results and the visualization is useful for that purpose. For the CFD research, contour lines and surfaces of a certain scalar flow function and arrows (and streamlines described later) of a vector function are often plotted as well as the body surface grids. In order to show contour levels, colors are useful. Shading the objects according to the light sources is also necessary to figure out the three-dimensional configurations. So-called Gauraud shading technique (simple one) is sufficient for this purpose and more complicated shading techniques to generate photo-realistic images such as metallic textures of the fighter bodies or the semi-transparent cockpits are insignificant. On the other hand, this kind of realistic images might be important when these pictures are presented to the people out of the CFD field because they might pay more attentions and feel more familiarity to the problems. If these photo-realistic images could have been generated as fast as simply shaded ones, more beautiful pictures would be preferable. Because of the limitation of the CPU power of workstations photo-realistic images require quite a long time and sometimes spoil the responsiveness of the process. There are two types of postprocess depending on the purpose. If the visualization is used for creating the pictures to be presented at the general audience, it belongs to the category of "presentation graphics". On the other hand, if visualization is used to make the flow analysis easy, then it belongs to the category of "personal graphics". The software for these two purposes should be considered separately and since we are fluid dynamicists, the following discussion is only for "personal graphics".

For "personal graphics", it is important to be able to show various aspects of the computed flow fields and generating nice and beautiful pictures are not important. The various ways of displaying various flow functions and simplicity to obtain these pictures are the key features.

Various Ways of Displaying Various Flow Functions

There are a lot of flow functions. Density, velocity, pressure, of course, and total pressure, dynamic pressure, vorticity, enthalpy and many other functions exist. The necessary functions are different for each flow field. Among them, the basic flow functions are required to be computed automatically with raw data, that are conservative variables (ρ, ρu, ρv, ρw, e) for example. It is desirable that user-define function can also be handled.

Several ways of showing flow functions are also required. Contour line plots and contour surface plots, for instance, are usually used to show a scalar function, but these plots seem to be used without the consciousness that the informations obtained from these plots are different although the flow function is the same. Figure 1 shows the contour line plots and the contour surface plots of the density of the flow in a supersonic intake[8]. The contour surface plots show the function values themselves, in other words, show where the density is high and where the density is low. On the other hand, the contour line plots firstly show the concentrations and rarefactions of the lines, that is, the gradients of the density. When you would like to see the shock waves in supersonic flows, it is better to plot the contour lines rather than the contour surfaces.

Window system is one of the advantages of graphics workstations. Use of multi-windows enable us to show various pictures at the same time. For example, it is necessary to see three-dimensional objects from different view points in order to figure them out. To move the view point dynamically is one way but the other way is to show two or more pictures of the same object from the different view points at the same time. This can easily be done with multi-windows. Two or more results are often compared for the analysis. Sometimes plots of two different flow functions of the same results are required, sometimes the same function plots of the different results are required. These are also realized using multi-windows. Actually Figure 1 shows an image of multi-windows displaying two different plots.

Usability

We have to decide what flow functions to plot, which planes to show. View point and/or view direction should

also be determined. The decision depends on the pictures to appear on the screen and can not be made in advance. Thus such decision should be made interactively by command input. Command input means not only keyboard input but also mice or the dials or input from any other devices. These input can be handled in the window system.

The choice of input device (the keyboard, the mice, the dials and so on) requires another decision. The solution depends on what to be input. When moving the view point, the mouse or the dials may be better. On the other hand, it is probably better to use the keyboard to set the contour levels because it requires definite numbers. Contour colors corresponding to the contour levels should be set by the mouse rather than keyboard because it is difficult to represent colors by numbers. Figure 1 shows an example. In Fig. 1, so-called "contour bar editor" is placed on the right window which shows the density contour plots inside the supersonic intake. The colors of the contour bar editor can shift, expand and shrink by the movement of the mouse, the contour surface plots on the screen reflect new correspondence of colors and levels, and new color image appears on the screen immediately after picking "set" button.

Any interactive process should also be as simple as the case of contour bar editor. For instance, when the flow function is changed, the contour surface plots should be changed to those of the new function. When the levels of the contour lines are changed, the contour lines should be automatically re-plotted with the new levels. Planes of contours can also be edited, that is, added and deleted easily. This kind of convenience is important to reduce the burden of the users.

There might be a case that the procedure is predetermined even if interactive process is basic. It is quite tedious to input commands especially when repeating the same or similar process as done before. The function to record the history of the command process, to edit the recorded command process if necessary and to run it automatically (so-called "history function") are necessary then. In the case of the present postprocessor, the command history is recorded in a text file called history file. The contents of the file are command lines that are equivalent to the processes done. Then the history file can easily be edited with general text editor.

Another aspect of the interactiveness is rapidity of creating image. One factor of rapidity is the algorithm to prepare data to pass to the graphics libraries such as contour line data, streamline data and contour surface data with colors indicating the function values. The other factor is the speed of the graphics libraries and the CPU power. Even when a postprocessor is programmed as an interactive one, it does not become an interactive postprocessor if the speed of creating images is quite slow. This is the reason why photo-realistic images are not welcomed for "personal graphics".

Extracting Digital Data

Although visualization plays an important role of the postprocess of the CFD and is useful to grasp the whole image of the flow field, digital data such as pressure distribution on an airfoil and pressure value at the center of the vortex are also important. In the case of the center of the vortex, the coordinates or the indices of the grid should be pointed out on the screen and the necessary digital data should be extracted from the screen. Figure 2 shows the three-dimensional positioning on the screen. Two windows are displayed. On the left window, the hair cursors are pointing the center of the vortex over the double-delta wing. The cursors, however, cannot specify the three-dimensional position because the window is two-dimensional and the cursors actually indicate a line that is perpendicular to the window and goes through the center of the vortex. We have to specify the depth of the position along the line. The right window is opened for that purpose. The right window shows the picture viewing from the right-hand side of the left window. So the horizontal cursor on the right window could be considered as the line that the cursors on the left window specifies. Then the vertical cursor on the right window, that can move from left to right or from right to left, can specify the depth of the position along the line. Or practically it is simply done to specify the three-dimensional position just by setting the cursors at the point you want on both windows. After getting the three-dimensional position, the values of primitive variables and flow functions as well as the position are displayed on the text window by a certain command.

Examples

Followings are the examples showing the required features of the postprocessor described above.

Figure 3 is an example to show two functions at the same time. Figure 3 (a) shows the pressure contour plots over the double-delta wing[9] and Figure 3 (b) shows the total pressure contour plots.

Figure 4 shows two different results. Figure 4 (a) shows the total pressure contour plots over the double-delta wing at an angle-of-attack of 12 degree and Figure 4 (b) shows the total pressure contour plots at an angle-of-attack of 30 degree. In this case, the same computational grid is used but it is possible to show two different results using different grids.

Velocity vector plots and near-surface streamlines are shown in Fig. 5. The near-surface streamlines are obtained by computing the streamlines within a specified grid surface near the wing.

Another example is shown in Fig. 6. The total pressure contours are plotted on several cross-sections on the right small window and they indicate the interaction of vortices from the strake and the wing leading edges. Better way to show the vortex interaction is moving the cross-section on which the total pressure contours are plotted forward or backward. The co-author, who calculated this flow field, made a video animation of the moving contour surface plots[10] before. It revealed the phenomenon of the vortex interaction (and also the vortex breakdown) clearly. In video animations, however, no parameters such as viewing, the position of cross-section, flow functions cannot be changed once the animations are recorded. Here, on the left window in Fig. 6, the cross-section can move forward and backward by the mouse movement. As this is done on the workstation, any parameters can be changed inter-

actively. When you have two-dimensional time-dependent computation results, unsteady animations can be seen with the pseudo-three-dimensional data constructed of the original two-dimensional data by the idea that the time-axis is considered to be the z-axis.

Figure 7 shows the example of the interactiveness. Each process is done by the selection of the left menu or command input in the right-bottom window. The same process can be done by both ways. The reason why two ways of input are prepared is that the menu selection is easier for beginners but the command input is finally quicker, especially for the trained people. The window between the menu and the window for commands is called "File Browser" on which the computation grid and result files can be specified. The right-bottom window showing the picture of the mouse is called "Mouse Guide" and it shows the functions of the mouse buttons according to the position of the mouse pointer. The other windows are the lists that control data read in and created pictures. On the lists it is possible to delete or save the data and pictures. The "Interactive View" menu appears by the "Interactive View" button on the menu and it makes possible to change the viewing of the windows by the interactive mouse input.

3. Method to Trace Streamlines

As well as the basic requirements for the postprocessor for the CFD described in the previous section, functions to illuminate the physics of the flow fields, in other words, to extract the features of the flow fields, are important. One of these functions is to trace streamlines. Streamlines are useful especially for the vortical flows. They reveal the behaviors of the vortices.

Off-Surface Streamline

Three-dimensional streamlines are generally obtained by integration of the velocity vectors and the velocity of a certain position is obtained by the interpolation of the velocities at the grid points near there. The plots only using interpolations such as contour lines essentially have little error, error of order of one grid size at most. On the other hand, integrations generally accumulate errors. For the visualizations, errors can be allowed if they cannot be recognized on the pictures. Errors of the streamlines, however, could exceed this allowance. Figures 8 (a) and 8 (b) show the off-surface streamlines over a double-delta wing. This is the case of the vortex interaction ($\alpha = 12°$). The streamlines are plotted with the same computational result but with different integration method and they show so different pictures that we might deduce wrong interpretation of the flow characteristics.

The points of the method to trace streamlines are summarized in the following three items.

- Integration domain (physical/computational)
- Interpolation method of velocities and coordinates
- Integration method and integration step

In physical domain, x, y, z-coordinates and physical velocity components (u, v, w) are used and in computational domain, computational coordinates (ξ, η, ζ) and contravariant velocity vector components (U, V, W) are used. For example, simple Euler-explicit-like integrations in these domains can be written as,

$$x(t + \Delta t) = x(t) + u(t)\Delta t \tag{1}$$

$$\xi(t + \Delta t) = \xi(t) + U(t)\Delta t \tag{2}$$

It is obvious that the accuracy of the integration in physical domain is better than that in computational domain because there is an error of the coordinate transformation in computational domain, in other words, the error of the evaluation of metrics. Consider the case of uniform flow. Contravariant velocity can take wide range of values if computational grid is clustered and stretched as in the case of body-fitted coordinate system. This may cause the difficulty of the programming of streamline tracing.

There is no information about the non-grid points when the finite difference method is used for the computation. As far as the governing equations are not solved at the non-grid points, no interpolations are physically accurate. So we choose a simple linear interpolation method. A linear interpolation has a good nature that no extrema exist within a cell but at a grid point although it is only first order.

There are several choices for the integration method. Two methods are examined here. The first integration method is sort of "Crank-Nicolson implicit" method, that can be written as,

$$x(t + \Delta t) = x(t) + 0.5\{u(t) + u(t + \Delta t)\}\Delta t \tag{3}$$

in physical domain. Figure 8 (a) shows the streamlines using this method. The other method is simple Euler-explicit-like, that can be written as,

$$x(t + \Delta t) = x(t) + u(t)\Delta t \tag{4}$$

The streamlines in Fig. 8 (b) are plotted with this explicit method using the same integration step as Fig. 8 (a).

Because of the nature of numerical integrations, as the integration step becomes smaller, the integration becomes more accurate. Figure 8 (c) shows the streamlines integrated by the Euler-explicit-like method with 1/20 integration step of the previous two examples and that gives much better results. That means error becomes negligible on the screen even though it exists and is accumulated with integration. This is only an example but the followings are generally concluded.

- Resultant streamline is (basically) not dependent on the way of integration if the integration step is small enough.
- The adequate integration step much differs depending on the integration method.

Since the choice of the integration step affects the accuracy, streamline tracing programs have to be tested before practical use. A good way to check the program is to trace streamline in an analytically defined flow field where the streamlines are known in advance. As streamline tracing is often used for a vortical flow, one example of the test flow field is a spiral flow such as,

$$\begin{cases} u = -y \\ v = x \\ w = const \end{cases} \tag{5}$$

This flow field is good for examining the capability to follow the curvature of the streamline. If the curvature is underestimated (this often occur when a simple explicit integration method is used), the spiral goes outward. If the curvature is overestimated, the spiral go inward. It is preferable to keep the radius of the spiral for a long time.

Near-Surface Streamline

Another function to extract the features of the flow field is near-surface streamline tracing. Although near-surface streamline is not the same as oil-flow-pattern in the experiments, near-surface streamline is useful to detect the separations and flow behavior in the boundary layer.

Near-surface streamline is obtained as the same method as off-surface streamline but the integration domain is confined to the surface near the body and the grid surface near the body is usually selected in the case of the finite difference methods. The key issues of tracing near-surface streamline is thus almost the same as those of off-surface streamline. Selection of the surface to trace streamlines within is the only difference. Figure 9 (a) shows an example of near-surface streamlines.

Streamline is one method to show the flow physics. There must be many other methods. The question of what to plot to understand the flow physics is what is the essence of the flow field. There are more to be discussed.

4. Methods to Simulate Experiments

It is important to compare computed results with experimental ones for the validations of the computational simulations. Digital data such as forces or pressure distributions on airfoils are often used for comparisons. When the entire flow fields are to be compared, visualized images are used because there are few methods except visualizations to show entire flow fields in experiments. The ordinary visualized images such as density contour plots have often been compared with schlieren photographs and other kinds of pictures taken in experiments. This comparison is, however, not fair and accurate even though they show similar images because the process to generate images from computed results does not follow the principles of visualizations in experiments. To make the comparison fair and accurate, the methods to simulate visualization processes of experiments should be considered. The methods to simulate three sorts of optical visualization in experiments are discussed in the following of this section first.

Schlieren Photograph

The first optical visualization is schlieren photograph. In two-dimensional flow, the intensity of each point of schlieren photographs is proportional to the density gradient perpendicular to the knife edge because the deflection rate of ray is proportional to the density gradient[11](Fig. 10). When the knife edge is set to be perpendicular to the x axis, for instance, the intensity I is written as

$$I \propto \frac{\partial \rho}{\partial x} \tag{6}$$

and color schlieren photographs are also simulated using hue (H) instead of intensity.

$$H \propto \frac{\partial \rho}{\partial x} \tag{7}$$

Figure 11 shows an example of the simulated color schlieren photograph. The flow field is two-dimensional supersonic intake and Mach number is 3 which is the design condition. Figure 11 (a) is color schlieren photograph[12] of the experiment, and Figure 11 (b) shows the simulated schlieren photograph using the computed result[8]. Figure 11 (c) is the density contour surface plots and the difference between (b) and (c) is obvious.

In three-dimensional flow, since the magnitude of density gradient varies along one ray of light, the total deflection rate of the ray (that is equal to intensity) is proportional to the integration of the density gradient along the ray.

$$I \propto \int_{ray} \frac{\partial \rho}{\partial x}\, dl \tag{8}$$

Numerical integration is done by the following manner. At first, the structured grid that we usually use is divided into tetrahedra to simplify the problem. Then tetrahedra that the traced ray of light goes through are searched. The ray comes into the tetrahedron through one face and goes out through one of the other three faces. Since the interpolation inside a tetrahedron can be linear, the integration in the tetrahedron I_t can simply be obtained as Eq. (9),

$$I_t = \frac{1}{2}\left(\left(\frac{\partial \rho}{\partial x}\right)_{in} + \left(\frac{\partial \rho}{\partial x}\right)_{out}\right) \times l \tag{9}$$

where $()_{in}$ denotes the point where the ray comes in, $()_{out}$ denotes the point where the ray goes out and l is the length between these points (Fig. 12). This integration is carried out in every tetrahedron that the ray goes through and the each integrated value is summed up. Finally the intensity of that point becomes

$$I = \sum I_t \tag{10}$$

If the rays from all of pixels on the window of the graphic display are traced, it is equivalent to the so-called volume rendering. In order to save the computation time, here the number of rays are reduced. The basic idea of reducing the rays is choosing some of the pixels, pixels corresponding to the specified grid points for instance, tracing the rays from these pixels and interpolating the intensities at the other pixels that are not traced.

An example of a three-dimensional flow field is shown in Fig. 13. A cone of 15 degree is placed in the supersonic flow ($M_\infty = 3$) with 0 degree angle-of-attack. The simulated three-dimensional schlieren photograph image is shown on the upper half and the simulated two-dimensional schlieren photograph image within a plane of symmetry is shown in the lower half. No density gradients in the area between the shock wave and the cone are observed in two-dimensional schlieren photograph. But since the rays of light through that area experience density gradients when crossing the shock waves, that area has a different color from the uniform flow region in three-dimensional schlieren photograph. This effect may become important in the case of a more complex flow field.

Shadowgraph

Shadowgraphs in experiments are obtained in the similar manner as schlieren photographs, but without knife edges. The intensity is, now, proportional to the gradient of the deflection rate on the screen because the concentration and rarefaction of the rays occur according to the relative deflection rate of the rays. The screen does not become bright nor dark when the deflection rates are constant[11]. In two-dimensional flows, it can be written as,

$$I \propto \frac{\partial^2 \rho}{\partial x^2} + \frac{\partial^2 \rho}{\partial y^2} \tag{11}$$

Figure 14 (a) shows the simulated shadowgraph pattern of the previous intake simulation.

In three-dimensional flow, it is not the integration of Eq. (11) but the gradient of the integration of $grad\,\rho$ in the directions perpendicular to the direction of the ray of the light. So Equations (8)-(10) are substituted by Eqs. (12)-(14).

$$I \propto grad \left(\int_{ray} grad\,\rho \, dl \right) \tag{12}$$

$$I_t = \frac{1}{2}(grad\,\rho_{in} + grad\,\rho_{out}) \times l \tag{13}$$

$$I = grad(\sum I_t) \tag{14}$$

where,

$$grad \; : \; \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial z} \right)$$

Figure 15 shows an example of three-dimensional shadowgraph simulation of the same supersonic flow over a cone. The simulated three-dimensional shadowgraph pattern and the two-dimensional shadowgraph pattern within a plane of symmetry are plotted. No evident difference is observed in the region between the shock wave and the cone because the second derivatives of the density in the region is not large (this can be verified by the fact that the color in the region in Fig. 13 is almost constant).

Unlike schlieren photographs, the film or the screen is not placed at the focal plane for shadowgraphs and the image is more or less distorted in actual experiments. This effect is not considered here.

Interferogram

Interferogram shows the light and shade pattern according to the difference of the light-path length that goes through the test section and the light-path length that goes through the reference area[11]. Since the light-path length depends on the density, the light and shade pattern corresponds to the difference of the density between the test section and the reference area. In two-dimensional flow, the numbers of drift of the light and shade stripe N is proportional to the difference of the density(Eq.(15)).

$$N \propto \Delta\rho \tag{15}$$

where,

$$\Delta\rho = \rho - \rho_{ref}$$

and ρ_{ref} is the density in the reference area. Figure 14 (b) shows the example of two-dimensional interferogram pattern of the intake.

In three-dimensional flow, N is proportional to the integration of the difference of the density that can be written as,

$$N \propto \int_{ray} \Delta\rho \, dl \tag{16}$$

The simulated interferogram pattern of the cone is shown in Fig. 16. Again the difference between two- and three-dimension is obvious.

In the case of two-dimensional simulations of visualization process, errors are considered to be as small as those of ordinary contour plots because interpolation is carried out within a grid cell. On the other hand, errors tend to be accumulated in the case of three-dimensional simulations because of the numerical interpolation and integration. The situation is similar to that of streamline tracing and the accuracy should also be checked. Figure 17 shows an example of the test of the accuracy of the three-dimensional schlieren photograph simulation. The density distribution over the cone is given by an analytic (although not physical) function and the color schlieren pattern is analytically calculated in Fig. 17 (a). In Fig. 17 (b), the density distribution over the cone is here given at the discretized grid points and the color schlieren pattern is calculated by the present method. The difference cannot be recognized from these pictures and the accuracy of this case may be considered to be sufficient from the view point of visualizations although it is not possible to draw conclusion from one example.

These methods, especially two-dimensional simulations, have another advantage to ordinary visualization methods. Computed schlieren photograph can show compressions and expansions much more clearly than density contour surface plots and it makes understanding the flow fields easier. In the case of the three-dimensional methods to simulate experiments, the story is not so simple. Firstly, there exists the problem of accuracy. Secondly the obtained image may lose the information along the direction of the integration because the integration is a sort of averaging operation. Nevertheless we think these three-dimensional methods are worthy when computed results are compared with pictures taken in experiments, and the flow fields can be understood with these methods as well as the other methods described above.

Oil Flow Pattern

There are another kinds of visualization method in experiments; particle path trace and oil flow pattern. Strictly speaking, particle path in the experiments is not off-surface streamline because the particles have a volume and a specific gravity that is different from air and non-equilibrium effect may not be negligible. Similarly oil flow pattern is not a near-surface streamline because the oil has its own viscosity. It is not easy to estimate these effects but still necessary when we want to compare experimental photos with pictures of computation results, even though particle path and oil flow may not give pure physical information than streamlines because of these effects.

Particle path, however, seems to be close to the off-surface streamline when the flow field is steady. On the other hand, oil flow pattern cannot be considered to be the same as near-surface streamlines. Figure 9 (a) shows the near-surface streamlines on the double-delta wing at the angle-of-attack of 30 degree and 9 (b) shows the oil flow pattern of an experiment[13]. These pictures do not match, and as well as the positions of the separation lines, the pattern near the separation line is quite different. Although main reason for the discrepancy between the computed result and the experiment may be due to the transition to turbulent flow and the grid resolution, the effect of the viscosity of the oil may also contribute. Because of the viscosity of the oil, streamlines seem not to be able to turn so suddenly or, in other words, oil has some effect of smoothing the velocities. Based on this idea, we correct the velocity by adding the second derivatives of the velocity to the velocity and draw near-surface streamline with it. Figure 9 (c) shows an example of near-surface streamline using the corrected velocity. The pattern becomes closer to the experiment. This is only a trial and we have no intention to say this is the true computed oil flow pattern but at least we have to do something like this to compare computation results with the experiments of oil flow.

5. Concluding Remarks

Postprocess of the flow simulation is important for understanding flow fields. The basic function of postprocess of the CFD is to visualize computed results as a method of data processing. The capability to visualize various flow functions with various way is required. Interactive and simple usability is also important.

Another purpose of the postprocess is to highlight the flow physics. Off-surface and near-surface streamline tracing are examples. There must be another methods and the choice of the methods to extract the feature of the flow fields depend on the problem. The postprocessors should have capability to be customized or to be implemented a new visualization method into.

For the validation of computations, the methods to simulate processes in experimental visualizations are also important to compare the computed results with experimental pictures. These methods, especially optical ones such as schlieren photograph, have another importance. They were discussed only from an engineering point of view, that is a comparison of computations with experiments. Mathematically speaking, however, these methods are considered as ones that show certain functions of density, for instance one component of first derivative of density in the case of the schlieren photograph. Since we have complete information about the flow field in computation, derivatives of any other flow functions can also be plotted. This kind of mathematical process may have possibility to illuminate features of flow fields.

It seems that too much attention is focused on generating nice and beautiful pictures that sometimes indicate no physical information. The real postprocess is not only to visualize computed flow field but also to cook the computed flow field to illuminate the flow characteristics by means of mathematical approach such as integration and differentiation. To find out how to cook the computed results, of course, we have to have good physical insight into the flow field.

References

[1] Edwards, D. E., "Three-Dimensional Visualization of Fluid Dynamic Problems," AIAA Paper 89-0136, 1989.

[2] Modiano, D. L., Giles, M. B. and Murman, E. M., "Visualization of Three-Dimensional CFD Solutions," AIAA Paper 89-0138, 1989.

[3] Smith, M. H., Van Dalsem, W. R., Dougherty, F. C. and Buning, P. G., "Analysis and Visualization of Complex Unsteady Three-Dimensional Flows," AIAA Paper 89-0139, 1989.

[4] Buning, P. G. and Steger, J. L., "Graphics and Flow Visualization in Computational Fluid Dynamics," AIAA Paper 85-1507-CP, 1985.

[5] Lasinski, T., Buning, P., Choi, D., Rogers, S., Bancroft, G. and Merritt, F., "Flow Visualization of CFD Using Graphics Workstations," AIAA Paper 87-1180, 1987.

[6] Bitz, F. J. and Zabusky, N. J., "David and "Visiometrics": Visualizing, Diagnosing and Quantifying Evolving Amorphous Objects," *Computers in Physics*, 1989.

[7] Tamura, Y. and Fujii, K., "Use of Graphic Workstation for Computational Fluid Dynamics," *A Collection of Technical Papers 3rd ISCFD-Nagoya*, 1989.

[8] Kuroda, S. and Fujii, K., "Supersonic Inlet Flow Computations Using Fortified Navier-Stokes Approach," *3rd Symposium on CFD*, 1989, in Japanese.

[9] Fujii, K. and Obayashi, S., "Use of High-Resolution Upwind Scheme for Vortical Flow Simulations," *AIAA Journal of Aircraft*, Vol. 26, No. 12, December 1989, pp. 1123-1129.

[10] Fujii, K. and Schiff, L. B., The NASA video used at AIAA 19th Fluid and Plasma Conference (AIAA Paper 87-1229), June 1987.

[11] Liepmann, H. W. and Roshko, A., *Element of Gas Dynamics*, John Wiley, New York, 1956, pp. 153-179.

[12] Sakata, K., Yanagi, R., Shindo, S., Sakakibara, S., Hara, N., Honami, S., Shiraishi, K., Yasu, S. and Tanaka, A., "Flow Visualization of Supersonic Air Intake," *An Album of Flow Visualization*, No.6, 1989, in Japanese.

[13] Brennenstuhl, U. and Hummel, D., "Vortex Formulation Over Double-Delta Wing," *Proceedings of the 13th Congress of the International Council of the aeronautical Sciences*, ICAS Paper 82-6.6.3, August 1982.

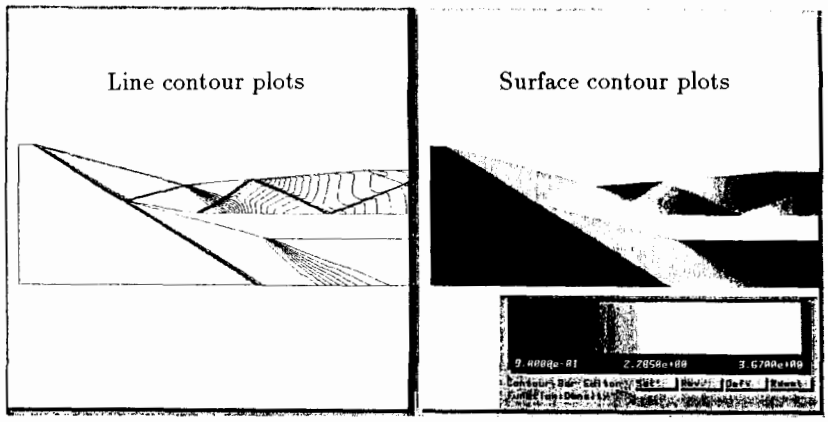


Figure 1 Two-dimensional supersonic intake ($M_\infty = 3.0$)

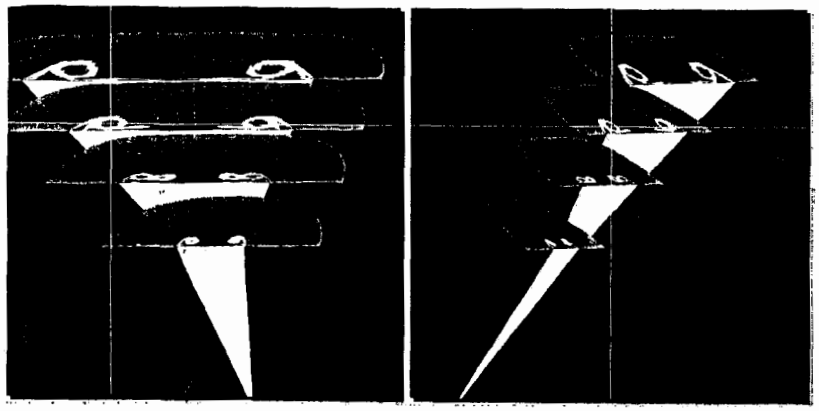
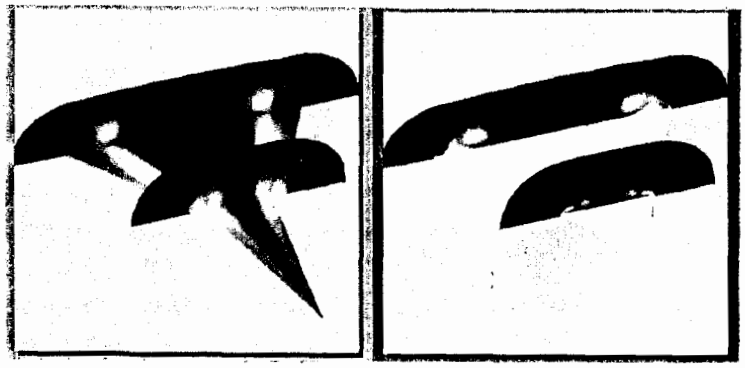
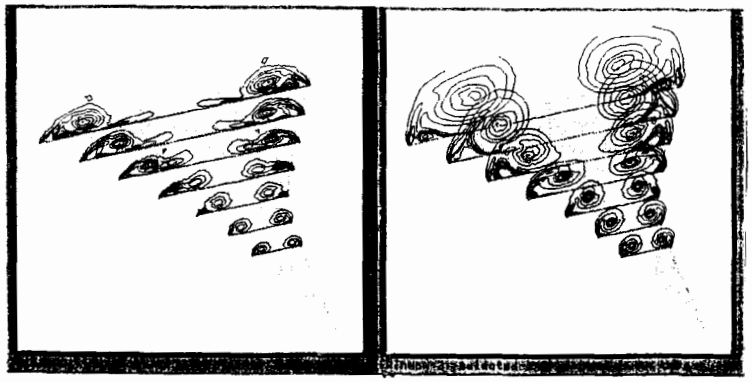


Figure 2 Example of three-dimensional positioning



(a) pressure contour surface plots (b) total-pressure contour surface plots

Figure 3 Contour surface plots over double-delta wing ($M_\infty = 0.3, \alpha = 12°, Re = 1.3 \times 10^6$)



(a) angle-of-attack of 12 degree (b) angle-of-attack of 30 degree

Figure 4 Comparison of two computation results

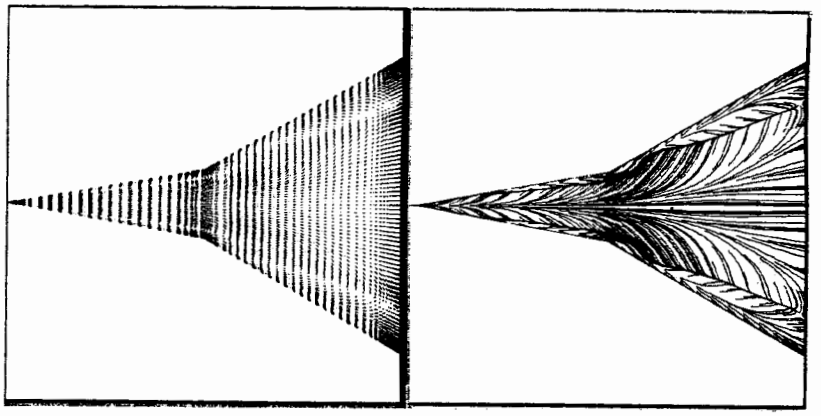


Figure 5 Velocity vectors and near-surface streamlines near double-delta wing

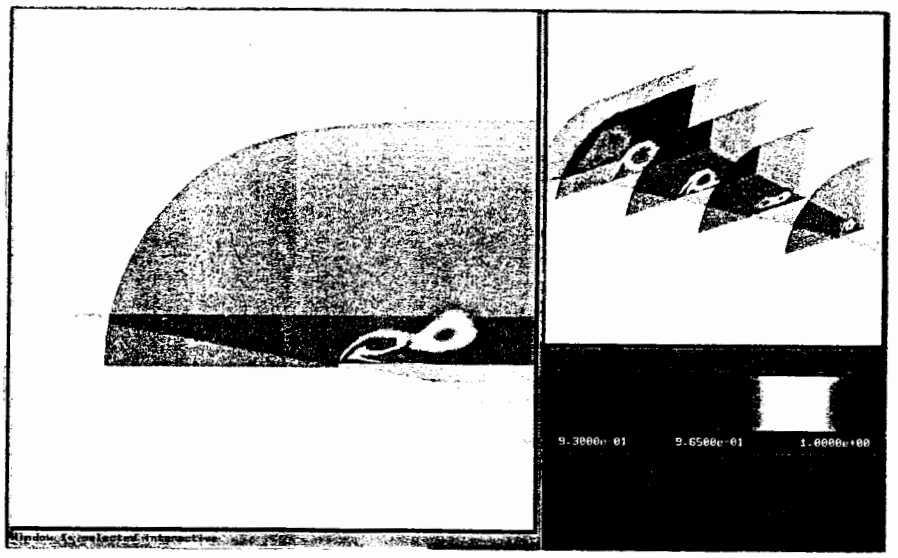


Figure 6 Contour surface sweeping

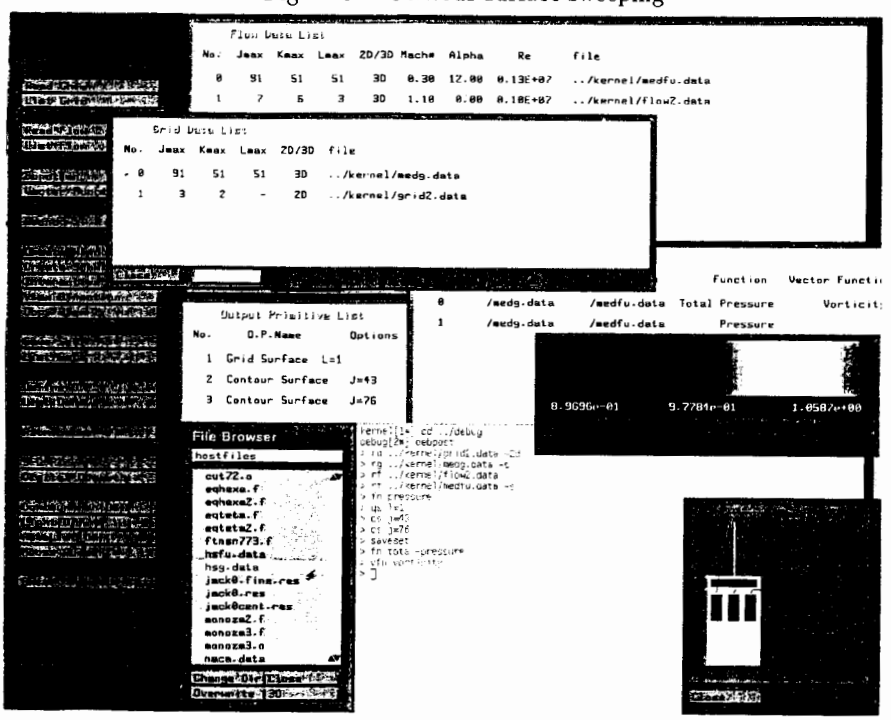
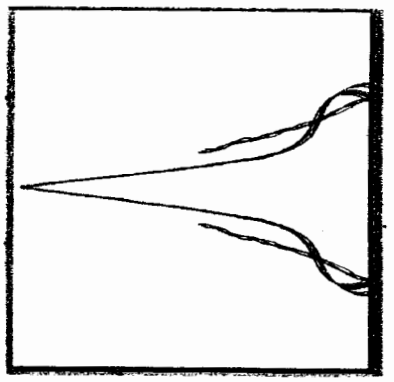
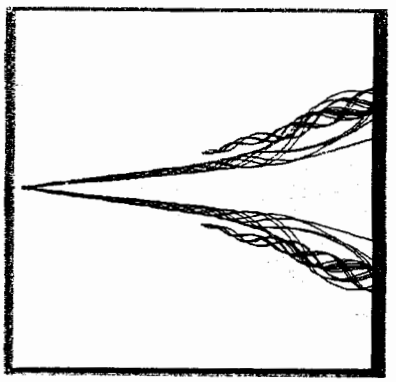


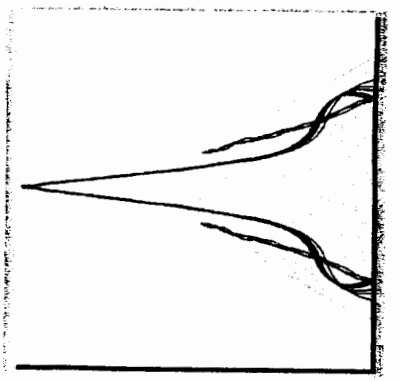
Figure 7 Menus, windows and lists



(a) Crank-Nicolson-like method (Δt_0)

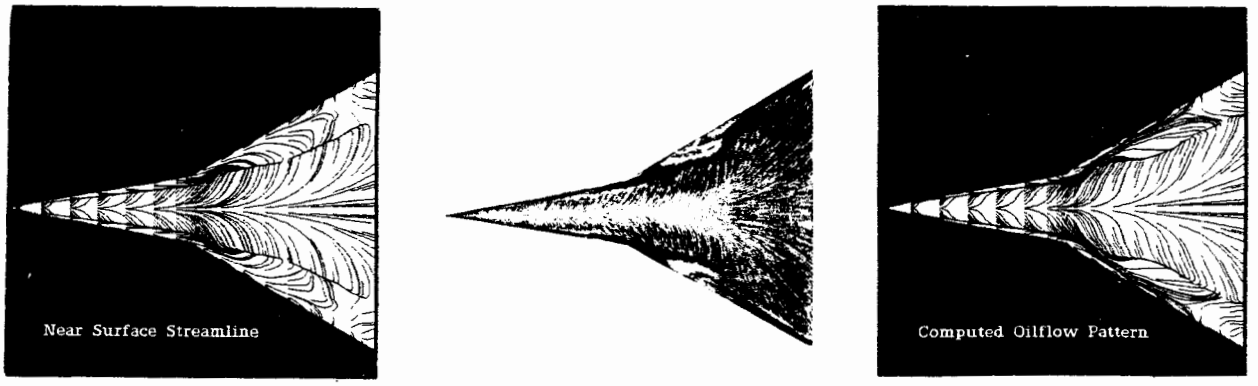


(b) Euler-explicit-like method (Δt_0)



(c) Euler-explicit-like method ($\frac{1}{20}\Delta t_0$)

Figure 8 Off-surface streamline over double-delta wing ($M_\infty = 0.3, \alpha = 12°, Re = 1.3 \times 10^6$)



(a) computed near-surface streamlines (b) experimental oil flow pattern[13] (c) computed oil flow pattern

Figure 9 Near-surface streamlines and oil flow pattern

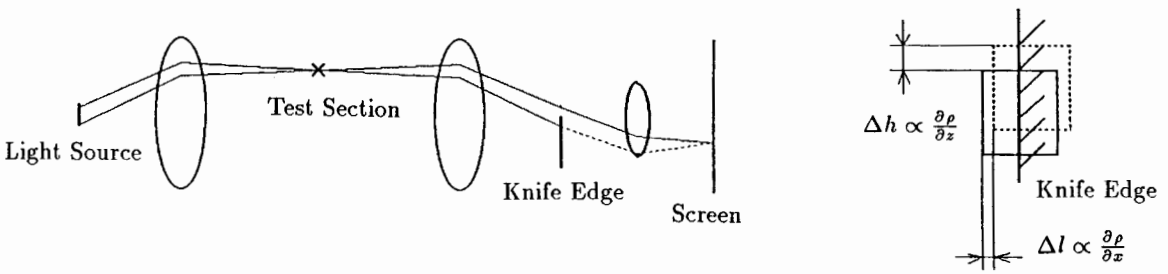


Figure 10 Principle of schlieren photograph

(a) experimental color schlieren photograph[12] (b) computer simulated schlieren photograph

(c) density contour plots

Figure 11 Two-dimensional supersonic intake

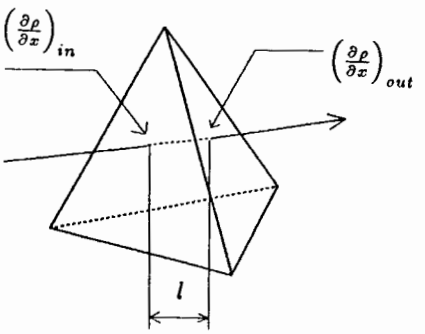


Figure 12 Numerical integration procedure within a tetrahedron

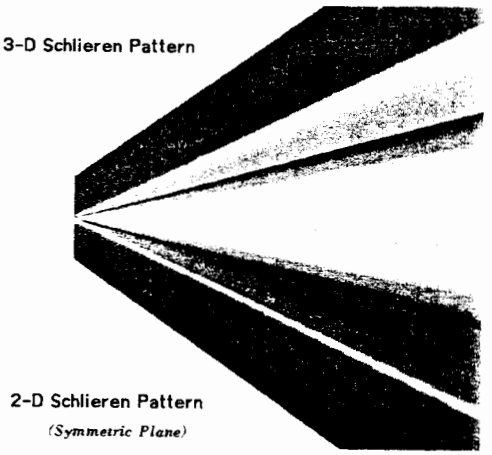
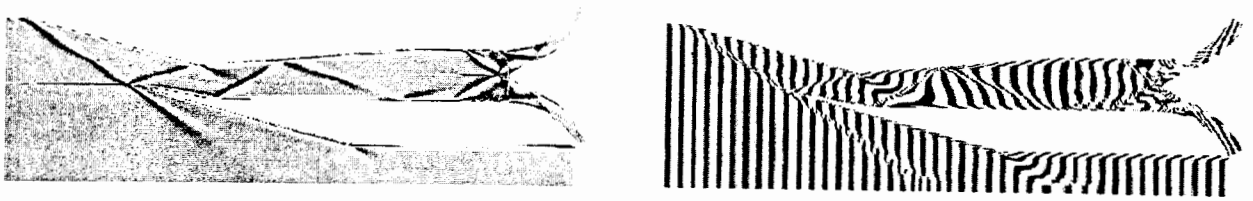


Figure 13 Computed schlieren photographs of a cone in the supersonic flow $(M_\infty = 3.0, \alpha = 0.0°)$



(a) shadowgraph (b) interferogram

Figure 14 Computer simulated shadowgraph and interferogram pattern

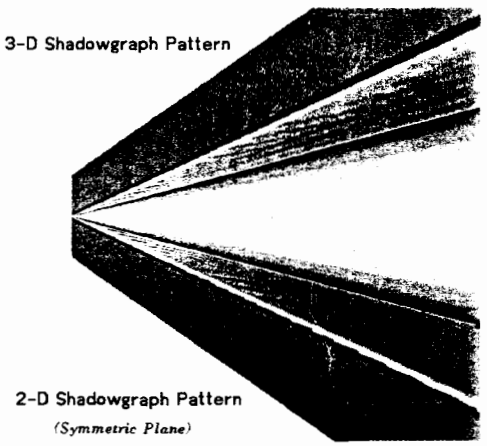


Figure 15 Computed shadowgraphs of a cone in the supersonic flow $(M_\infty = 3.0, \alpha = 0.0°)$

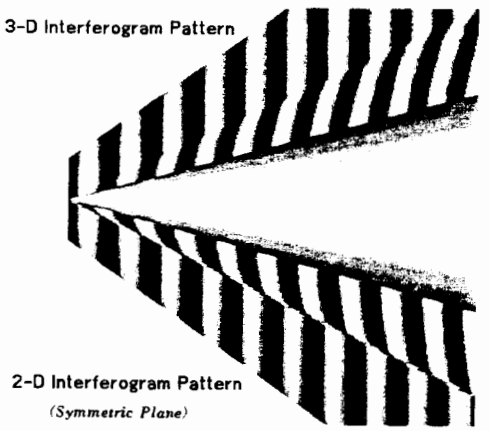


Figure 16 Computed interferogram pattern of a cone in the supersonic flow $(M_\infty = 3.0, \alpha = 0.0°)$

(a) analytically calculated pattern (b) numerically computed pattern by the present method

Figure 17 Test of accuracy of the three-dimensional schlieren photograph simulation