

# An Automatic Mesh Generator Based CFD System to be Used as a Design Tool

Tatsuhito Matsushima  
Software Cradle Co., Ltd., Japan

Copyright © 2001 Society of Automotive Engineers, Inc.

## ABSTRACT

A computational fluid dynamics (CFD) system is presented. The system, called SCRYU/Tetra, is based on an automatic tetrahedral mesh generator and has been developed as a design tool for engineers who are not necessarily fluid dynamics specialists, but are seeking better designs of their equipment through CFD analysis. In order to make the system useful for non-specialists, efforts are taken so that the system can handle complicated geometry from CAD systems, does not require any special skill which is not essential for the understanding of the flow phenomenon itself, and is accurate enough to correctly identify the best design candidate from others. Microsoft WINDOWS PC's are chosen as the main hardware platform to make the analysis cost low. The system has demonstrated an ability to complete analysis in a very short time (typically a few days) with satisfactory accuracy.

## INTRODUCTION

Lower cost and higher performance of personal computers have made CFD analysis increasingly easier for non-specialists. Accordingly, the demand from equipment designers to conduct CFD analysis by themselves in a reasonably short period of time is increasing rapidly. To make the CFD analysis a common practice for engineers, who are not necessarily fluid dynamics specialists, the CFD system should be designed carefully to eliminate unnecessary complications. First, the system should require no special skill that is not related to the understanding of the fluid phenomenon itself. One example is the hand construction skill of a computational mesh that is often required if the CFD analysis is carried out using a conventional hexahedral mesh. To construct a hexahedral mesh, the computational domain has to be decomposed into a set of blocks so that each block can be mapped to a parallelepiped without too much deformation. This procedure, called blocking, requires an intensive topological analysis of the geometry and requires considerable experience. The blocking is also a time consuming procedure. For example, it can take months to create a hexahedral mesh of a coolant fluid path in an engine. The process is also difficult to

automate reliably and completely, although research efforts are being made[1] to remove the difficulty. As a consequence, the hexahedral mesh generation typically requires a human intervention[2]. With the current status of the technology, we judge that such a procedure would degrade the quality of CFD system as a design tool and should be avoided.

Another desirable point for the CFD system as a design tool is that the system should be able to handle the complicated geometry of actually manufactured products. In many cases the geometry for the CFD analysis comes from a CAD system and it can be quite complicated because it is primarily generated for production purpose. The CFD system should be able to carry out the analysis with minimal geometrical simplifications and modifications.

To satisfy these requirements, we have chosen to base SCRYU/Tetra on hybrid meshes instead of more conventional hexahedral meshes. We define the hybrid mesh as a mesh that consists of tetrahedral, pyramidal, prismatic and hexahedral elements. Because a class of hybrid mesh – a tetrahedral mesh with prism layers inserted along the wall part of the geometry - can be generated automatically and reliably, this choice eliminates the necessity of manual mesh construction. The automatic mesh generator is robust enough to complete the mesh generation even if the geometry is complicated. Thus it allows the user to import complicated geometry from most CAD systems with minimal simplifications and modifications.

The choice of hybrid mesh makes the mesh generation simple and robust. However, a question arises if a good accuracy is obtained by using the hybrid mesh. Preliminary investigation shows that the insertion of prism layers along the wall greatly improves the accuracy. However it is not obvious if a satisfactory accuracy is obtained with this kind of mesh. We will show that with appropriate choice of the computational method, the flow analysis conducted using this kind of mesh can be as accurate as the analysis conducted using the hexahedral meshes.

It is a great advantage to the user if the mesh generation and the solver execution can be combined to one batch job. If this can be done, the operator does not need to wait for the completion of the mesh generation to set up the calculation conditions and start the solver execution. To realize the one batch job approach, all analysis parameters need to be set before the computational mesh is generated. We will present a way to prepare the analysis parameters to realize this approach.

## **CAD INTERFACE**

**STL FORMAT AS THE PRIMARY INTERFACE** – In situations where the CFD system is used for research, the geometry is often created by the researchers themselves using a geometry generator (modeler). A CFD system is often equipped with such a modeler and a new geometry is created in order to conduct a CFD analysis. However for design purposes, a designer is more likely to have a pre-assigned modeler that is tuned up to maximize the efficiency from the manufacturing point of view. The designer is also likely to have already defined a geometry, probably for purposes other than CFD analysis and wants to use it for a CFD analysis. For this group of people, the modeler equipped with the CFD system is of less importance and the robustness of geometry interface from the CAD system to the CFD system would be more important. From this consideration, we have chosen SCRYU/Tetra not to be equipped with a modeler, but rather import the surface data of the geometry from various CAD systems.

Several formats can be considered as an interface to import a surface geometry from CAD systems. Direct interface from each CAD system, IGES and STL format are among the possibilities.

The direct interface should be effective and cause relatively small number of problems when it is properly programmed. However it needs to be developed for each CAD system and therefore takes a huge development effort. The IGES format is standard for exporting and importing geometry between CAD systems. However it has differences in detail for various CAD systems and our experience suggests that it can require significant effort to import geometry reliably.

The STL (Stereolithography) format, which describes the geometry as a collection of small triangular patches, is very simple and is supported by most CAD systems. Unlike NURBS[3], which is often used in the IGES format, its description of geometry is straightforward and can be understood without any special training. Also it is relatively easy to detect and fix defects. For this reason, we have chosen the STL format as the primary geometry input interface to our system. We note that the STL format has ASCII and binary version. If the STL format is supported, other formats based on triangular and quadrilateral patches, e.g., DXF, NASTRAN and I-DEAS formats can be supported easily.

One drawback exists in adopting the STL format instead of the IGES format as the primary geometry input interface: the ability to express geometry accurately is lower for the STL format than for the IGES format. This is because the STL format essentially supports only the first order (planar) surface patches. The first order surface is a problem for p-type FEM stress analyses where both geometry and the governing equations need to be represented in high order. However, for CFD analyses conducted using SCRYU/Tetra, the discretization order is at most second order. In addition, most calculations are done using a turbulence model and the error caused by the modeling of turbulence probably surpasses the error caused by the geometry representation error. Hence the author believes that the drawback caused by the selection of STL format as the primary geometry interface is insignificant.

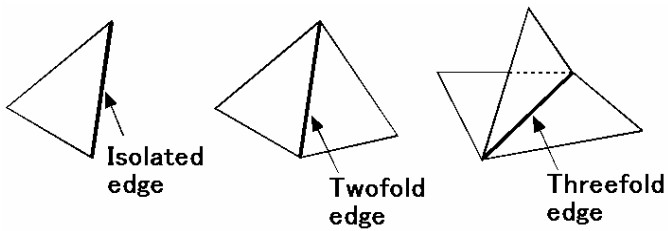
**QUALITY OF INPUT GEOMETRY** – The quality of STL output can be vastly different depending on the CAD system and how the geometry data is prepared. As a rule of thumb, the surface quality is good if the CAD system is a solid modeler and the geometry is generated in the CAD system from scratch. In this case, only a few surface corrections are expected to be necessary. On the other hand, if the CAD system is a surface modeler, the surface quality can be poor. The quality is potentially also poor if the geometry data is imported to a solid modeler as surface data. This is because a surface modeler needs to treat the connection between surface fragments in a much looser sense than a solid modeler does. Hence some surface fragments tend to be output independently from other neighboring surface fragments and the output geometry surface fails to be closed.

As one can see, this type of trouble is not due to the STL format itself, but it is due to the way the geometry is prepared. To fix the problem, the connection between surface fragments must be defined or recovered in the CAD system. The SCRYU/Tetra preprocessor provides some functionality to repair surface defects in case defects still remain. However it is best to repair the surface geometry as much as possible before it is output as a STL file.

## **MESH GENERATION**

**SURFACE CHECKING AND FIXING** – As discussed in the previous section, the quality of input geometry can be poor depending on how the geometry data is generated. One way to assess the quality of the input geometry is to analyze the connectivity of the triangular patches that make up the geometry. In Fig. 1, we show some possible ways in which neighboring triangular faces are connected. If the geometry is composed of a single material and the triangular patches make up a closed surface, then all edges of triangular patches should be the twofold edge. On the other hand, if the geometry is defective and has a hole, the triangles adjacent to the hole will have isolated edges. If the geometry is sectioned into two or more materials, threefold edges or edges of higher multiplicity may occur.

Hence, by analyzing the multiplicity of the edges of triangular patches one can obtain useful information about the quality of the surface geometry.



**Fig. 1 Edge multiplicity of triangular patches**

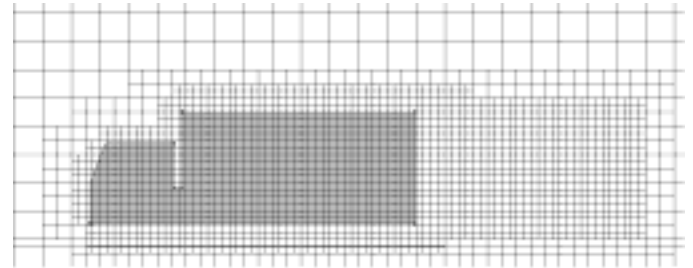
For the edge multiplicity analysis it is important to know the number of materials and the existence of panels in the geometry because it determines the possibility of the occurrence of edges whose multiplicity is higher than two. This is not a problem because the number of materials and the existence of panels are usually known when the surface geometry is generated.

To fix a defective surface geometry, one can add or delete triangular patches, split or swap edges, and add, delete or merge nodes. These elementary operations can be combined to form useful macro operations, for example, to fill holes and to sew isolated edges. The elementary operations and other useful macro operations are implemented in the SCRYU/Tetra preprocessor.

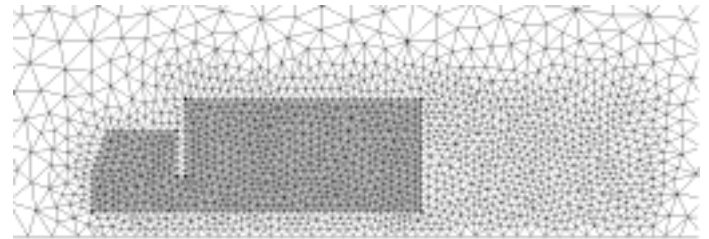
**MESH SIZE SPECIFICATION** – Versatile mesh size specification ability is needed for a CFD preprocessor. This is because for an accurate flow analysis, one needs to specify a small mesh size not only for geometrically complicated parts of the computational domain, but also where eddies, shear layers and other important fluid dynamical features exist. These flow features are not necessarily confined close to the boundary but can extend to inner part of the domain, as a jet blowing into a space or a car wake suggests. Therefore, beside the size specification according to the geometry, the preprocessor should provide a method to specify the mesh size freely without depending on the geometry of the computational domain.

One way to attain this is to use an octree[4] mesh size specification as shown in Fig. 2. Cubic boxes, called octants, form the octree. The size of the octant becomes approximately the size of the tetrahedral element at its location when the tetrahedral mesh is generated. One can specify an arbitrary mesh size to any part of the computational domain by mouse and keyboard operations on the condition that the size of the neighboring octants cannot differ by more than twice. Various functions to assist editing the octree are available to make flexible mesh size specification possible. For example, one can use a function to select octants that interfere with some specified part of the geometry and a function to select neighbor octants of already selected octants.

The manual mesh size specification is useful when the operator knows in which part of the computational geometry a fine mesh should be used. In fact it is the favored method if the operator has some experience and understanding of the flow field. However if this is not the case, some automatic mechanism to specify the mesh size would be useful. Currently we are experimenting with the adaptive mesh size specification and the results will be reported elsewhere.



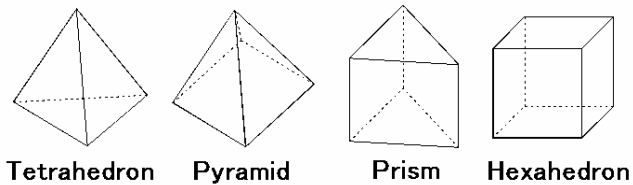
**Fig. 2 Octree mesh size specification**



**Fig. 3 Tetrahedral mesh generated using the octree shown in Fig. 2**

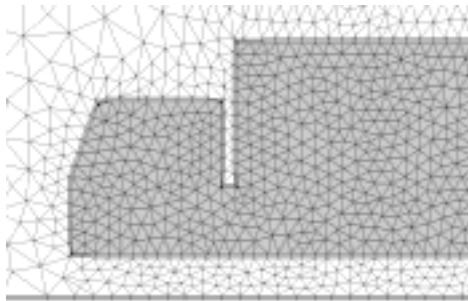
**MESH GENERATION** – Once the mesh size is specified by an octree, the surface part of the computational mesh is generated from the original geometry. Then using the surface mesh, the tetrahedral mesh is generated by the advancing front method[5]. In the advancing front method the computational domain is filled with tetrahedrons starting from the surface mesh to the inner part of the domain. It is possible to generate the tetrahedral mesh directly from the octree, however in our method the octree is used to specify the mesh size and is not used for the construction of the mesh. One reason for this is because the advancing front method tends to produce a better quality mesh compared to the octree method. A tetrahedral mesh generated using the octree in Fig. 2 is shown in Fig. 3.

**HYBRID MESH AND PRISM LAYERS** – Many of the engineering flows have a high Reynolds number, and the CFD system must provide a way to treat the boundary layer accurately. The insertion of prism layers along the wall part of the computational mesh surface is an effective way to treat the boundary layer and the SCRYU/Tetra adopts this method. This is possible because the system supports the hybrid mesh, which includes the prism elements as shown in Fig. 4.



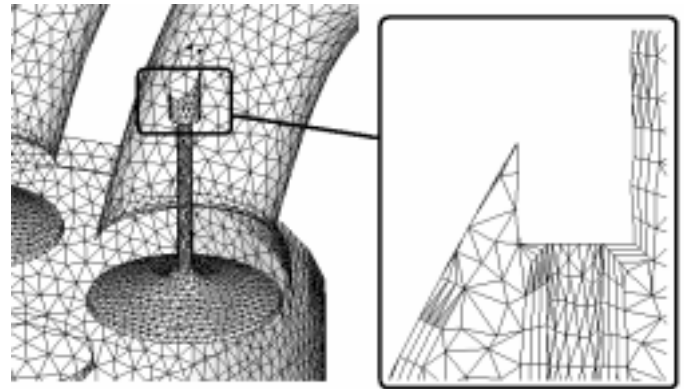
**Fig. 4 Elements of a hybrid mesh**

The prism layer insertion is done after the tetrahedral mesh is generated. The input parameters are the number of prism layers, the thickness of the first prism layer, and the thickness growth rate if multiple prism layers are to be inserted. The process first inserts a thin layer of prisms along the specified boundary of the tetrahedral mesh and grows its thickness until the thickness becomes the assigned size. If necessary, the tetrahedral elements adjacent to the prism layer are removed as the prism layer thickness grows and the resulting void is refilled with tetrahedrons. An example of the hybrid mesh is shown in Fig. 5. Two to three prism layers are usually inserted. The layer thickness is chosen so that the law of the wall condition is valid and the best accurately is obtained. Selecting the layer thickness as 30 to 100 in the non-dimensional distance from the wall ( $y^+$ ) appears to give a good result[6].



**Fig. 5 A hybrid mesh generated from the tetrahedral mesh in Fig. 3**

The prism insertion is not always possible if only tetrahedrons and prism elements are used to construct a hybrid mesh. An example occurs if the prism insertion needs to be canceled because of a sharp wedge shaped feature in the geometry. It can be shown that the prism layer can be terminated at an arbitrary location inside the mesh if pyramid elements are used to connect the prism elements and the tetrahedral elements. SCRYU/Tetra's preprocessor detects the necessity of pyramid elements and inserts the pyramid elements automatically. A result of such a procedure is shown in Fig. 6. The insertion of the prism layer is avoided due to the existence of a wedge shaped feature in the geometry.



**Fig. 6 Example of the use of pyramid elements. A section mesh pattern of the figure on the left hand side is shown on the right hand side.**

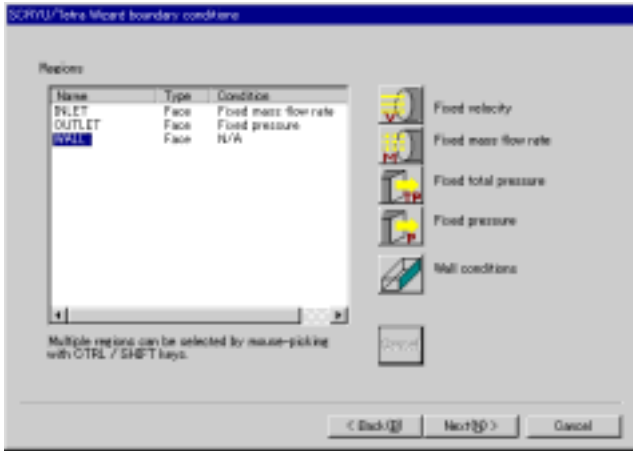
## ANALYSIS PARAMETERS

To set up the calculation parameters, one needs to specify the inlet, outlet, wall and other surfaces to apply the boundary conditions. A convenient way to do this is to assign a name to the appropriate part of the surface and associate the calculation parameters such as the velocity or pressure boundary conditions to the name. With this method to specify the boundary conditions, the calculation parameters become independent of the details of the hybrid mesh as long as the same boundary names are used. Consequently the file containing the calculation parameters can be reused if the hybrid mesh is regenerated.

A more convenient way to manipulate the boundary names is to assign names directly to the boundaries of the input geometry from the CAD system. The location of the boundaries and their names may be stored together with the geometry as a computer file. When the hybrid mesh is generated, the boundaries and the names registered in the input geometry are transferred to the hybrid mesh automatically. This method of registering the boundaries to the surface geometry has advantages over the method to register boundaries only to the computational mesh. First, one does not have to specify the boundary names each time the hybrid mesh is generated. Second, the complete description of the problem becomes possible only with the surface geometry file and the calculation parameter file because the parameters such as the inlet velocity value and the outlet pressure value can be associated with the name of the boundary. This implies that the consecutive execution of the mesh generation and the flow calculation becomes possible without human intervention.

We note that the method to name the boundary of the surface geometry can be extended to treat the volume region of the mesh if a closed surface is recognized as a candidate volume region to be named. The volume regions are used, for example, to specify the location of a filter and a heat generator and to specify the mesh speed when the ALE function is used.

To help to set up the calculation conditions, dialog panels can be used to set up all calculation parameters. Parameters used most often, for example the inlet velocity and the output pressure, can be input by a wizard type dialogues, i.e., by filling in input panels which ask simple questions one by one. An example of the boundary condition input dialog is shown in Fig. 7.

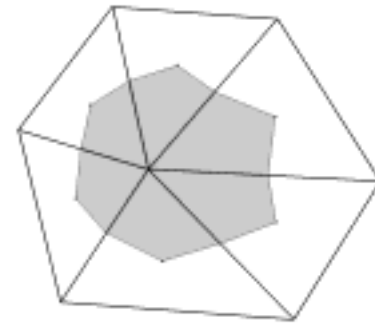


**Fig. 7 An example of boundary condition input dialog**

## FLOW SOLVER

**NODE BASED CONTROL VOLUME METHOD** – To solve the flow field by the control volume method using the hybrid mesh based on tetrahedrons and prisms, two different methods can be considered[7]. The first, the cell-centered scheme, is to use the elements of the hybrid mesh directly as the control volume. The second, the node-based scheme, is to define a control volume around each node of the hybrid mesh. A two dimensional example of the node-based control volume is shown in Fig. 8. In the figure, the control volume around the node at the center is defined using the centroids of the elements around the node and the midpoints of the edges emanating from the node. The extension to the three dimensional case is straightforward if one considers the element centroids, the element surface centroids and the element edge midpoints.

If the application of boundary conditions is ignored, the number of unknowns in the resulting matrix equation is equal to the number of elements in the cell-centered scheme and is equal to the number of nodes in the node-based scheme. In a hexahedral mesh, the number of nodes is greater than the number of elements. However in a hybrid mesh based on the tetrahedrons and prisms, the number of nodes is typically considerably smaller than the number of elements. Roughly estimated, the ratio is about 1 to 4. Hence for our case, the node-based scheme leads to a smaller number of unknowns in the resulting matrix equation and more efficiency is expected. For this reason the SCRYU/Tetra adopts the node-based scheme.



**Fig. 8 Control volume for node-based scheme**

**ACCURACY** – When a high Reynolds number flow is solved, it is important to treat the boundary layer near the wall accurately. A tetrahedral mesh is not suited for this purpose because the element height along the wall is not uniform. This is a major disadvantage of the tetrahedral mesh compared to the hexahedral mesh for which it is relatively easy to control the mesh size near the wall. However for a hybrid mesh, the prism insertion is possible and consequently it is easy to control the element height along the wall. The prism insertion has a dramatic effect on the accuracy of the analysis, as shown in [6].

Though hexahedral meshes are more difficult to generate compared to the hybrid meshes based on tetrahedrons and prisms, they are often favored because it is believed that the CFD analysis based on hexahedral meshes gives more accurate result compared to tetrahedron and prism based hybrid meshes. One reason to believe this is that if the hexahedral mesh is constructed such that its mesh pattern lies close to the stream line pattern of the flow, the false diffusion due to the discretization error can be contained to a low level. However, in general it is very hard to construct such a mesh unless the geometry under consideration is very simple. Consequently, some compromise is usually necessary and we must consider that this advantage is not always valid. In SCRYU/Tetra, a second order evaluation of the convection term (MUSCL[8]) is used for the momentum equation to contain the false diffusion to a low level. The effect of using the MUSCL will be shown later in the examples.

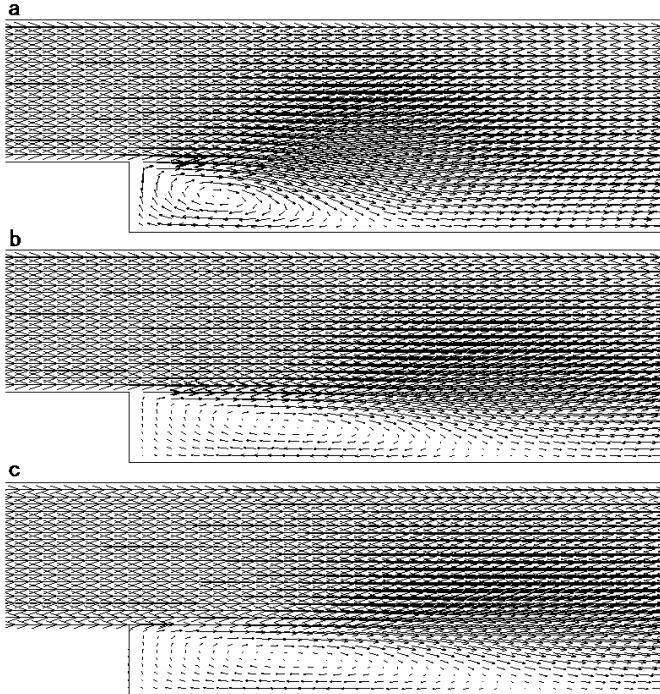
**EFFICIENCY** – We have noted that the choice between the cell-centered and the node-based scheme gives a significant difference in the efficiency of the flow solver if a hybrid mesh is used. On the other hand, the efficiency of the matrix solver itself is also important because a large portion of the computational time is spent solving the matrix equations derived from the governing flow equations. We have adopted the algebraic multi-grid matrix solver[9] and the ILUBiCG-STAB method[10] to solve the discretized momentum and mass conservation equation. In our experience, the algebraic multi-grid matrix solver often gives a slightly better performance to solve the mass conservation equation and is chosen as a default matrix solver for this.

FUNCTIONS - As a general purpose flow solver, the SCRYU/Tetra supports various functions, as summarized in Table 1. In high Reynolds number applications, the standard k-ε model with the law of the wall boundary conditions is used as a default.

**Table 1 Main analysis functions of SCRYU/Tetra**

3D incompressible / compressible fluids.
Laminar flow
Turbulent flow (k-ε method :Standard / RNG / MP)
Discontinuous mesh
Arbitrary Lagrangean-Eulerian (ALE) formulation
Fluid-solid thermal coupling
Diffusion of concentration species
Gas mixing
Radiation heat transfer
Periodic conditions

EXAMPLES – An example of a back step flow is shown in Fig. 9. In the figure, an eddy is formed behind the step at the bottom of the geometry. A hybrid mesh is used for the computation of Fig. 9(a) and Fig. 9 (b), while a hexahedral mesh is used in Fig. 9(c). In Fig. 9(a), the convection term is treated only as first order (upwind) and in Fig. 9(b), it is treated as second order (MUSCL). The effect of the false diffusion reduction is apparent in Fig. 9(b).

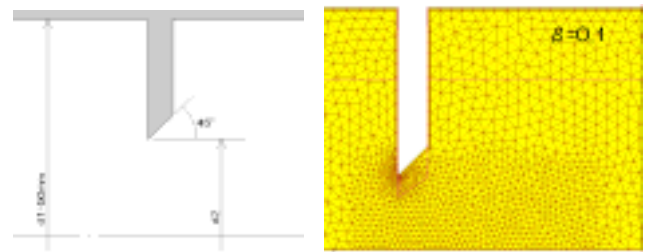


**Fig. 9 Velocity field behind a step shape. a. UPWIND, b. MUSCL, c. MUSCL (Hexahedral mesh). Re=46000 based on the step height and inlet velocity. The standard k-ε model is used.**

Fig. 9(c) shows a second order (MUSCL) computation result using a hexahedral mesh. The degree of freedom

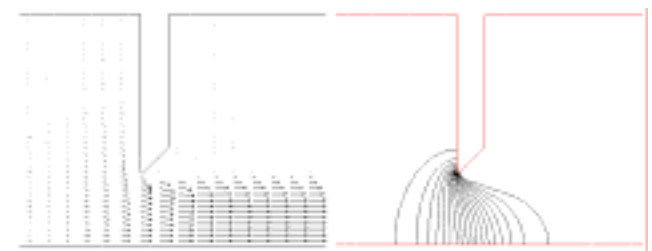
in the mesh in Fig. 9(c) is about 2800 and is comparable with the degree of freedom in the mesh in Fig. 9(a) and Fig. 9(b). The distance from the bottom of the step to the reattachment point is 3.5, 5.6 and 5.7 in the unit of step height for Fig. 9 (a), (b) and (c), respectively. The result in Fig. 9(c) is similar to that of Fig. 9(b) and shows that the hybrid mesh can give a comparable accuracy with the hexahedral mesh.

In the next example we show the result of orifice pressure drop calculations. The geometry of the orifice is shown on the left hand side of Fig. 10. The orifice is characterized by the area ratio  $\beta = (d_2/d_1)^2$ . The discharge coefficient  $C = \bar{U}_2 \sqrt{\rho/2\Delta P}$  is compared with experimental results. Here,  $\bar{U}_2$ ,  $\rho$  and  $\Delta P$  are the mean velocity at the orifice, fluid density and pressure drop across the orifice, respectively. An example of the computational mesh is shown for the case  $\beta = 0.1$  on the right hand side of Fig. 10.



**Fig. 10 Orifice geometry. The mesh is 1/8 model in the azimuthal direction and contains 37094 nodes.**

The velocity vector plot and the pressure contour lines for the case  $\beta = 0.1$  is shown in Fig. 11. As shown in the figure, a large pressure gradient occurs near the orifice edge and some concentration of node points is necessary there. In fact, the concentration of node points is also necessary in order to resolve the details of the orifice geometry at the edge.



**Fig. 11 Velocity and pressure field for  $\beta=0.1$ . Re=250000 (based on inlet velocity and pipe diameter).**

The computed discharge coefficient is compared with experiments in Table 2 for the cases  $\beta=0.1, 0.2$  and  $0.3$ . The maximum error occurs for the case  $\beta = 0.1$  and is 2.3%. We consider the result is accurate enough if the system is to be used as a design tool.

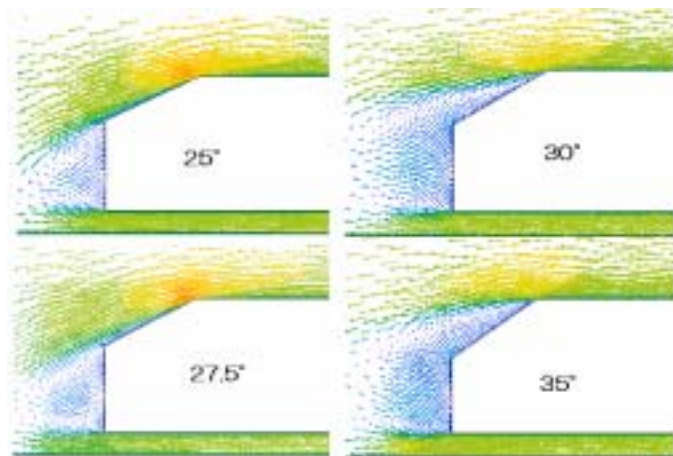
**Table 2 Comparison of computed and experimental discharge coefficient.**

$\beta$	C (Calc.)	C (Expt.)
0.1	0.616 (+2.3%)	0.602
0.2	0.623 (+1.3%)	0.615
0.3	0.645 (+1.7%)	0.634

In all cases the computed discharge coefficients are larger than the experimental value. This tendency becomes more apparent if the mesh resolution is lower. If the orifice is only a part of a more complicated geometry, it may not be possible to assign a fine resolution for the orifice part as in this example. Our investigation suggests that even in such a case the element size of about 1/15 of the orifice diameter  $d_2$  or a smaller size should be used near the orifice.

**FLOW SEPARATION** – To investigate the separation behavior of the flow behind a car body, Ahmed, et. al. [11] proposed a bluff body whose rear part has a slope at a slant angle  $\beta$ . The experiment shows that the flow separates from the body if the slant angle is greater than about 30 degrees. The calculation result by the SCRYU/Tetra is shown in Fig. 12. A good agreement of the separation behavior with the experiment is seen.

Compared to the example shown above, the prediction of the flow separation point from a smoothly curved surface is a more difficult problem because the separation point cannot be recognized from the geometry. If the  $k-\epsilon$  model and the law of the wall boundary conditions are used, the effect of adverse pressure gradient to the boundary layer and other effects are not treated rigorously and a considerable error in the prediction of flow separation point may result. Hence one should be careful if the accurate prediction of the separation point is critical for the successful flow analysis.



**Fig. 12 Separation behavior of the flow from a bluff body[11] at slant angles  $\beta = 25, 27.5, 30$  and  $35$  degrees.**

## HARDWARE PLATFORM

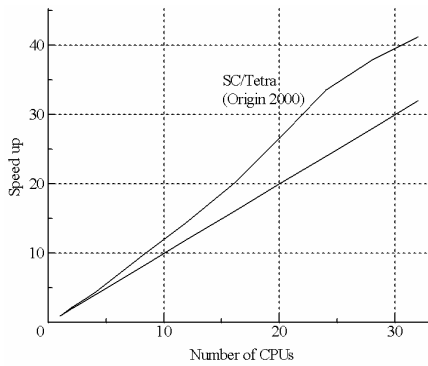
The preprocessor and the postprocessor are developed on the Microsoft WINDOWS NT system and the WINDOWS functions are used extensively to construct the graphical user interface. Thus the operation conforms to other WINDOWS applications and it should be easy to learn to use the preprocessor and the postprocessor. Because inexpensive WINDOWS hardware platforms can be used for SCRYU/Tetra, the cost of the system can be kept relatively low.

On the other hand, the flow solver is offered for both WINDOWS NT and UNIX operating systems. Hence a time consuming computation may be carried out as a UNIX background job. Some examples of typical computational resource usage are shown in Table 3. The mesh generation takes more memory than the solver execution and the requirement is roughly 250 bytes per element. The computational time is measured by using the 933MHz Pentium III processor. We note that the solver execution time can take longer if the discontinuous mesh and other functions are used. The memory requirement and the solver execution time are dependent on the ratio of the tetrahedral elements to the number of prism elements in the hybrid mesh.

**Table 3 Usage of computational resources. The computational time is measured by using Pentium III (933MHz). Solver execution time is for 200 iterations.**

a. Mesh generation			
Elements	Nodes	Memory	Time
976,398	205,366	244MB	0.4 hrs
2,205,529	438,166	550MB	0.8 hrs
b. Solver execution			
Elements	Nodes	Memory	Time
976,398	205,366	146MB	6.6 hrs
2,205,529	438,166	330MB	12.6 hrs

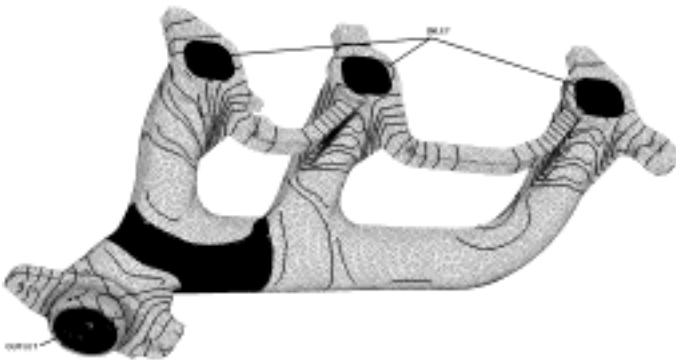
The parallel version of the solver is also available for the UNIX operating system, although currently some functions including the discontinuous mesh cannot be used. The parallel processing is realized using the MPI[12] library and the speed up for the case of SGI Origin 2000 (300MHz) is shown in Fig. 13. The actual speed up is greater than the ideal speed up, which is equal to the number of CPUs used in the calculation. This behavior is often observed if the parallel programming is done properly.



**Fig. 13 Speed up of parallel SCRYU/Tetra solver. The test problem contains 3,399,808 elements.**

## APPLICATION EXAMPLES

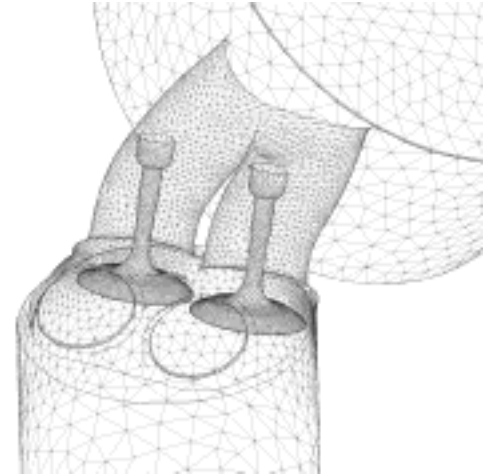
**FLOW IN A MANIFOLD** - The flow and temperature analysis result of a cast iron manifold is shown in Fig. 14. A heated gas with temperature 1223K enters each inlet at a constant mass flux rate, exchanges heat with the manifold, and leaves from the outlet. Free convection and radiation heat transfer boundary conditions are set for the outer surface of the manifold. The gas is treated as a compressible fluid. We note that the gas velocity increases as the inlet branches merge. As shown in the figure, the manifold surface temperature is higher near the root of inlet branches than near the gas inlets though the gas temperature decreases gradually as it flows from the inlet to the outlet. This results because the turbulent heat transfer coefficient at the inner surface of the manifold is high where the gas velocity is high. The phenomenon is also observed in real products.



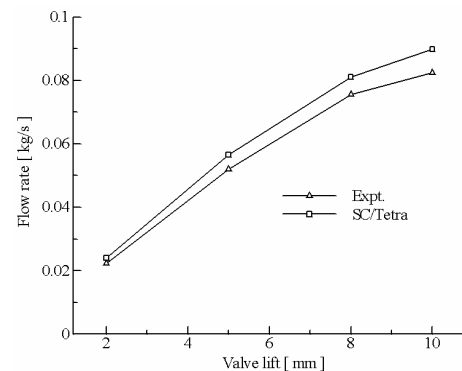
**Fig. 14 Temperature distribution of a cast iron manifold. Inlet gas temperature is 1223K. Surface region with  $T > 873K$  is shaded. The computational mesh contains 535,277 elements.**

The hybrid mesh used in the calculation above contains 535277 elements. With this size of the computational mesh, the mesh generation and the flow calculation can be carried out in one day using a reasonably fast WINDOWS PC provided that a good quality surface geometry from a CAD system is available. It would take much longer time to complete the analysis if a hexahedral mesh based CFD system is used.

**STEADY FLOW ANALYSIS OF ENGINE PORT** – In the next example we show a steady state flow analysis of an engine port. The computational mesh is shown in Fig. 15. We set a pressure difference (2937Pa) between the inlet and the outlet of the port and the mass flux flow rate is computed as a function of the valve lift. The result is shown in Fig. 16. The error in the mass flow rate prediction is within 10% of the experimental value.



**Fig. 15 Computational hybrid mesh of test engine port. In the figure, the valve lift is 10mm. The mesh contains 723,326 elements.**



**Fig. 16 Valve lift – flow rate relation**

As in the manifold example above, the mesh generation and the flow calculation can be carried out in one day if the surface geometry is available. To execute the analysis, one needs to specify (1) the location of the inlet, outlet and the wall boundaries, (2) the boundary conditions there, (3) the mesh size by an octree, (4) the number and the thickness of the prism layer and (5) some other simple parameters. The parameters may be saved to a file and may be reused for different valve lift cases. The octree can also be saved to a file and may be reused. In this case, the octree needs to be generated so that it suits for different valve locations. The use of the same octree minimizes the effect of differences in the computational mesh on the calculation results. The whole parameter setup takes no more than 15 minutes. Once the setup is complete, the mesh generation and the flow analysis are done consecutively and no human intervention is necessary until the output

file for the postprocessor is ready. Using the calculation parameter file and the octree file used for the first calculation, the subsequent computations with different valve lifts can be started in a matter of a few minutes.

**FAN CALCULATION** – Our last example shows the result of a fan calculation. The ALE function and a discontinuous mesh are used to rotate the fan around its axis. This solution method can be applied if the non-rotating part of the computational geometry is not axis-symmetric though in this example it is axis-symmetric. The geometry is shown in Fig. 17. Because the location of the fan blades changes in time, the calculation is inherently time-dependent and requires longer computational time compared to a steady state analysis where the changes of the flow field in time are irrelevant. The analysis is continued until the pressure difference across the fan becomes essentially time-independent. In this example, the mesh contains about 1.1 million elements and the procedure to generate the computational mesh is the same as that in earlier examples. However, the flow calculation takes 3 days because of the large number of time steps required to satisfy the condition stated above.

To shorten the analysis time, the parallel execution of the solver would be effective. However because this example uses the discontinuous mesh, the parallel execution of the solver is not available. We note that we are working to remove this limitation.



**Fig. 17 The geometry and a velocity plot of a fan**

## CONCLUSION

A CFD System based on an automatic mesh generator is discussed. Because the automatic mesh generator simplifies the mesh generation procedure greatly, no specialized skill is required to conduct a CFD analysis. The analysis is fast and accurate enough for engineering applications. The analysis cost is low because inexpensive Microsoft WINDOWS PCs can be used. For these reasons, the system developed in this paper is suited for use as an engineering design tool.

Results obtained in this paper have been implemented in a commercial code SCRYU/Tetra. It has been used by users from various manufacturing companies and universities[13][14][15].

## REFERENCES

1. Park, S., Lee, K., 'A new approach to automated multiblock decomposition for grid generation: a hypercube++ approach', in *Handbook of grid generation*, chapter 10, CRC-Press (1998).
2. Reymond, J.-D., Hauser, J., Xia, Y., 'Grid generation for 3D turbine configurations', in *Numerical Grid Generation in Computational Field Simulations*, ed. Soni, B. K., et.al., p451 (1996).
3. Piegl, L., Tiller, W., *The NURBS book*, Springer (1995).
4. Shephard, M. S., Coughny, H. L., et. al., 'Automatic grid generation using spatially based trees', in *Handbook of grid generation*, chapter 15, CRC-Press (1998).
5. Chan, C. T., Anastasiou, K., 'An automatic tetrahedral mesh generation scheme by the advancing front method', *Commun. numer. methods eng.*, 13, pp33-46 (1997).
6. Okumura, K., 'CFD simulation by automatically generated tetrahedral and prismatic cells for engine intake duct and coolant flow in three days', SAE paper 2000-01-0294 (2000).
7. Mavriplis, D. J., 'Unstructured grid techniques', *Annu. Rev. Fluid. Mech.*, 29, p473 (1997).
8. Van Leer, B., 'Towards the ultimate conservative difference scheme V. A second-order sequel to Godunov's method', *J. Comp. Phys.*, 32, p101 (1979).
9. Elias, S. R., Stubbley, G. D., Raithby, G. D., 'An adaptive agglomeration method for additive correction multigrid', *Int. J. Numer. Methods in Engng.*, 40, p887 (1997).
10. Dongarra, J., et. al., 'Templates for the solution of linear systems: building blocks for iterative methods', SIAM, Philadelphia, (1994)
11. Ahmed, S. R., Ramm, G., 'Some salient features of the time averaged ground vehicle wake', SAE paper 840300 (1984).
12. Snir, M., et. al., 'MPI: The complete reference (vol. 1), 2<sup>nd</sup> ed.', MIT Press, 1998.
13. Sano, T., Yoshida, Y., et. al., 'Numerical study of rotating stall in a pump vaned diffuser', to appear in ASME Fourth International Symposium on Pumping Machinery, May 29-June 1, 2001, New Orleans.
14. Oshio, F., et. al., 'Analysis of air ventilation performance based on aerodynamics simulation', SAE paper 2001-01-0296 (2001).
15. Ono, J., et. al., 'Development of underbody aerodynamic simulation using automatically generated tetrahedral and prismatic cells', SAE paper 2001-01-0704 (2001).