



## COMPUTATION OF COUPLED THERMAL-FLUID PROBLEMS IN DISTRIBUTED MEMORY ENVIRONMENT

H. Wei, H. M. Shang, Y. S. Chen  
Engineering Sciences Inc., Huntsville, AL  
Phone: (256) 883-6233, fax: (256) 883-6267, e-mail: hwei@esi-al.com

### ABSTRACT

The thermal-fluid coupling problems are very important to aerospace and engineering applications. In stead of analyzing heat transfer and fluid flow separately, this study merged two well-accepted engineering solution methods, SINDA for thermal analysis and FDNS for fluid flow simulation, into a unified multi-disciplinary thermal-fluid prediction method. A fully conservative patched grid interface algorithm for arbitrary two-dimensional and three-dimensional geometry has been developed. The state-of-the-art parallel computing concept was used to couple SINDA and FDNS for the communication of boundary conditions through PVM (Parallel Virtual Machine) libraries. Therefore, the thermal analysis performed by SINDA and the fluid flow calculated by FDNS are fully coupled to obtain steady state or transient solutions. The natural convection between two thick-walled eccentric tubes was calculated and the predicted results match the experiment data perfectly. A 3-D rocket engine model and a real 3-D SSME geometry were used to test the current model, and the reasonable temperature field was obtained.

### INTRODUCTION

Modeling of the thermal-fluid coupling effects plays an important role in the design and problem diagnostics of liquid rocket engine systems and the sub-systems, such as combustion chamber regenerative cooling channels compatibility, cryogenic fluid management with passive recirculation, etc. The heat transfer between different material and fluid media is also commonly encountered in the engineering practices. The applications include the cooling of electric equipment, material processing and compact heat exchangers. Conventional approach for the thermal-fluid coupling solution very often requires two separate analyses that involve different ways of practice and complexity in each discipline. This study is to merge two well-accepted engineering solution methods, SINDA and FDNS, into a unified multi-disciplinary thermal-fluid analysis method with the aid of patched grid and parallel computing techniques. In the resulting method, the thermal simulating by SINDA and the flow fields calculating by FDNS are fully coupled to get the steady state or transient solutions.

SINDA (Systems Improved Numerical Differencing Analyzer) [1] is a widely accepted thermal analysis software for simulating solid components energy balance using method of conductor-capacitor networks. Other models such as wall radiation heat transfer and one-dimensional fluid flow equations are used to provide boundary conditions for complex systems. On the other hand, many practical applications in rocket engine flow analysis require CFD models, such as the FDNS (Finite Difference Navier-Stokes) code [2], for better predictions of the flow fields which can not be modeled properly with the simplified method used in SINDA. Therefore, merge of these two disciplines into one unified analytical model will enhance the productivity and prediction capability of the thermal-fluid design community.

Since the grids for CFD and SINDA are generated independently, the grid lines of two adjoining regions may align (continuous grids) or may not align (discontinuous grids) with each other. Generally, the CFD model requires finer grids to accurately predict flow fields than the grids used for thermal analyses. So, the grid lines are mostly discontinuous at the interface for most applications. The boundary solution translation procedure must be conservative, stable, and robust for the integrated system. The patched grid approach [3] was used for interface linkage between SINDA and FDNS. We keep all the grid lines and collect the smallest cells. When the heat flux or temperature are exchanged across the interface, the local energy conservation is achieved by integrating upon cell areas. In the iterative procedure, SINDA and FDNS communicate and exchange boundary conditions through the boundary heat transfer coefficient and temperature.

A unique feature of this method is the usage of implicit coupling of SINDA and FDNS by the parallel computing technique. It makes this method better efficiency and stability, and applicable for both steady state and transient solutions. In the integrated SINDA and FDNS system using the PVM library [4], SINDA is run as the master (parent) process, and can initialize several FDNS copies for the slave (children) processes. Both SINDA and FDNS are run with their own input and control data, and are communicating and exchanging boundary conditions with each other through PVM.

The developed numerical method is first tested by a benchmark problem. A 2-D case of the natural convection between two thick-walled eccentric tubes has been simulated. The predicted results match the experiment data perfectly. Then, a 3-D rocket engine model and a real 3-D SSME geometry are used to test the current model. The nozzle flow, the solid wall and the cooling channel flow are all coupled together during the computation.

## NUMERICAL METHODS

### A FULLY CONSERVATIVE PATCHED GRID INTERFACE ALGORITHM

For 2-D cases, the interface boundary is determined by all of the face grid points of both adjoining zones as shown in Fig. 1. This enables that all the individual points from both zones lie in the interface line and then the interface is unique and accurate.

For 3-D cases, the cell elements are defined using the original grid points in both zones plus the intersection points. So, all of the grid points are used to construct the interface surface. The intersection point of these two set of grid meshes is defined as the intersection of one grid mesh with the image of the nearby cells of another grid mesh on it. For higher accuracy, the finer mesh is selected as the base mesh and the image of the other mesh is calculated based on every cell of the base mesh. If the interface is planar surface, the image can be calculated only once based on the base surface.

Figure 2 shows that a planar surface structured mesh intersect with another unstructured mesh. We keep all the grid lines and collect the resulting cells. They may be no longer quadrilaterals or triangular, but polygons with the edge number less than eight (the maximum edge number of a polygon is eight in the case of two structured grids mesh interface). The polygon doesn't need to be triangulated under the memory and speed consideration. Figure 3 is a cylinder face as a simple example of curved interface in 3-D application. The 6x6 mesh (dark lines) is the base mesh and the 4x4 mesh (light lines) is projected based on every cell of the base mesh to construct the interface mesh.

We use the unstructured grid data format to manage cell element at the interface. The surface in 3-D domain should first be translated into x-y plane by translation and rotation processes. Then cell elements (polygon) are detected. The process includes calculating the intersect point, determining the vortex of polygon, calculating the cell area, defining a pointer to indicate its corresponding cell ID in the original interface meshes of two zones respectively. When communicating across the interface, the local mass and energy conservation is enforced through integration upon cell areas.

### ACCELERATE PATCH GRID GEOMETRIC SEARCH BY USING BINARY SEARCH TREE ALGORITHM

In order to construct interface elements, every cell in one mesh must be checked with every cell in another mesh to see if they have intersections. The number of search is the first mesh cell number times the second mesh cell number,  $n_1 \times n_2$ . However, if we build a binary tree [5] to organize the geometric domain of one mesh, and search the intersected cell by using tree traverse technique, the search effort will be tremendously reduced. Assume every terminal node of the tree holds five cells, then the comparison times needed for one cell in mesh 1 to get the interested cell group in mesh 2 is only  $\log_2(n_2/5)$ .

Here we use the two end points coordinates of the block diagonal as the key to build the geometric binary search tree. First find out minimum and maximum x, y, z over all the grids in mesh 2. The root represents the cube A(min x, min y, min z)-B(max x, max y, max z). This block is bisected across the x axis and the region for which  $X_a < X < (X_a + X_b)/2$  is assigned to left son and the region for which  $(X_a + X_b)/2 < X < X_b$  is assigned to the right son.

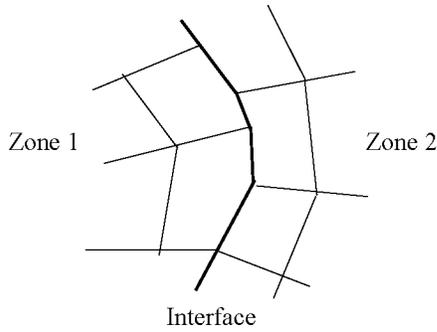


Figure 1. Interface is defined by all of the grid points in 2-D case

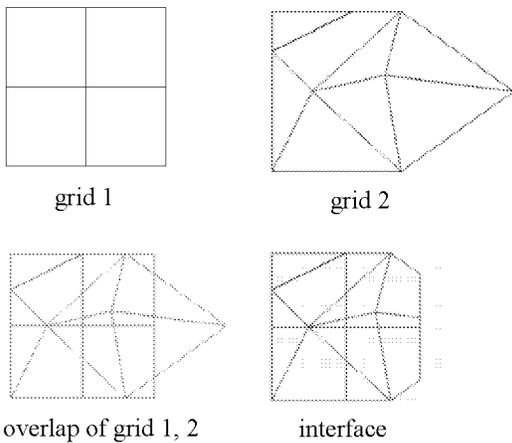


Figure 2 : Patched grid planar surface interface

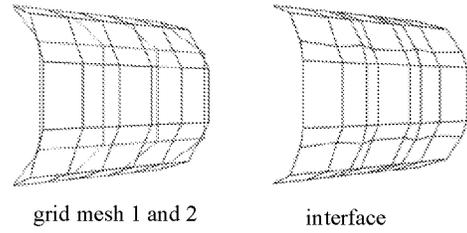


Figure 3. Patched grid curve surface interface

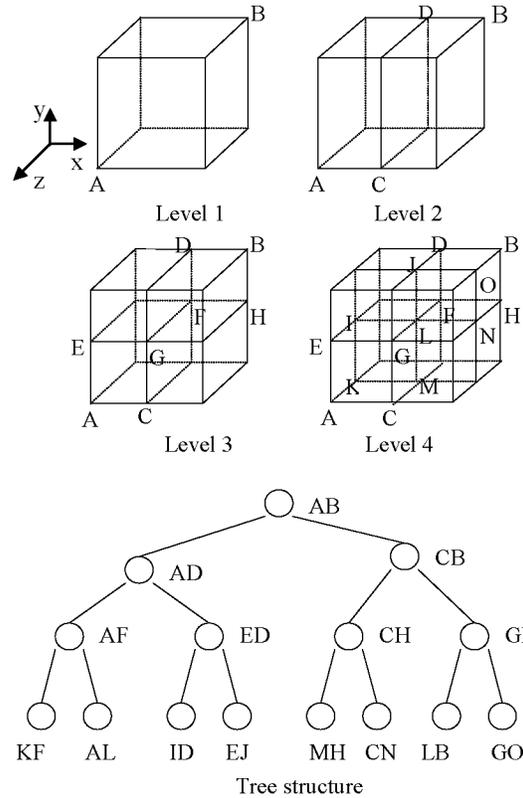


Figure 4. Geometric binary search tree structure.

For each of the node, repeat the process across Y-axis, and then do it across Z-axis. The process is continued by choosing X Y Z in cyclic order. Figure 4 shows the procedure.

Since the region represented by son node is covered by the region represented by its parent, so if a cell is not overlapped with a region represented by a node, the complete set of the regions stored in the sub tree of this node can be disregarded from the search. The geometric search algorithm can be displayed by a recursive procedure as:

1. Check if the cell overlapped with the region represented by the root.
2. If the cell overlapped with the left sub region, search the left tree.
3. If the cell overlapped with the right sub region, search the right tree.

The binary search tree is implemented by using C++ language and coupled with the patch grid subroutine in FORTRAN. For the 3-D nozzle case, there exist two interfaces between nozzle flow and solid wall. One interface is 231 cells in fluid side and 56 cells in solid side, another interface is 891 cells in fluid side and 56 cells in solid side. On IRIS workstation, the CPU time used for constructing the interface is 2.13s with binary search tree and 4.08s without binary search tree. The patched grid process is greatly speed up, and the larger the grid size, the more efficient of this method.

#### SINDA/FDNS PRE-PROCESSOR

The comprehensive CFD techniques including geometry modeling, grid generation, flow solver and post-processor have been well developed at Engineering Sciences. For the SINDA input file, it generally requires tedious hand calculations of nodal capacitance and conductance. To couple CFD model with SINDA, a preprocessor must be developed first to generate a SINDA input deck for subsequent finite difference analyses. We developed a preprocessor that has two main tasks. The first task is to detach interface boundary grids from SINDA domain grids to prepare boundary grid data for the patched grid manipulation as described above. The second task is to generate SINDA input file, which includes calculation of nodal capacitance and conductance for SINDA network model, setting boundary conditions, taking the CFD-SINDA communication subroutines and finally, writing out the SINDA input file according to its required format. Our preprocessor can take both structured and unstructured grids. It can also take the grid and boundary information directly from the PATRAN neutral file to generate the SINDA input file. And, the post-processor can print out SINDA temperature field in plot3d format, to be viewed through the CFD post-processor.

#### SINDA AND FDNS COUPLING WITH PVM

The communication between SINDA and FDNS will be achieved through parallel computing approach. In the integrated SINDA and FDNS system using PVM libraries, SINDA will run as master (parent) process, and can initialize several FDNS copies as other slave (children) processes. Both SINDA and FDNS run with their own input and control data, and communicate and exchange boundary conditions each other through PVM. SINDA calculates the conduction heat flux and temperature within the wall nodes, based on the boundary heat transfer coefficient provided by the CFD model. On the other hand, the CFD model uses the SINDA-calculated wall temperatures as fixed boundary temperatures, and solves the energy equation to calculate the fluid temperatures, heat flux and heat transfer coefficient at the boundary. This process is coupled and repeated for steady state or transient solutions.

The program communications between SINDA and FDNS are shown in Fig. 5. After starting PVM daemon, executing SINDA code will automatically start FDNS. The whole process will stop when the iteration number for steady state solution or time progressing for transient calculation exceed the specified values.

#### ENHANCEMENT OF THERMAL-FLUID COUPLING

In the iterative procedure, SINDA and FDNS communicate and exchange boundary conditions through the boundary heat flux or temperature. We have experienced convergence problem when passing heat flux directly from fluid side to solid side if the thermal conductivity is very small compared to fluid side's effective thermal conductivity. The temperature fields close to solid-fluid interface oscillate during the iteration procedure unless small time step is used. It is found that the numerical stability can be enhanced if we pass the heat transfer coefficient and temperature instead of heat flux.

The heat flux calculated in fluid side can be expressed as:

$$q = h(T_f - T_w)$$

and

$$q = k(T_f - T_w) / \Delta_s$$

where  $q$  is the heat flux,  $h$  is the heat transfer coefficient,  $T_f$  is the fluid temperature at the grid adjacent to the wall,  $T_w$  is the wall temperature,  $k$  is the thermal conductivity and  $\Delta_s$  is the normal distance between the grid and wall. So the heat transfer coefficient of laminar flow can be calculated as

$$h = k / \Delta_s$$

where, for turbulent flow,  $h$  is given by the turbulence model.

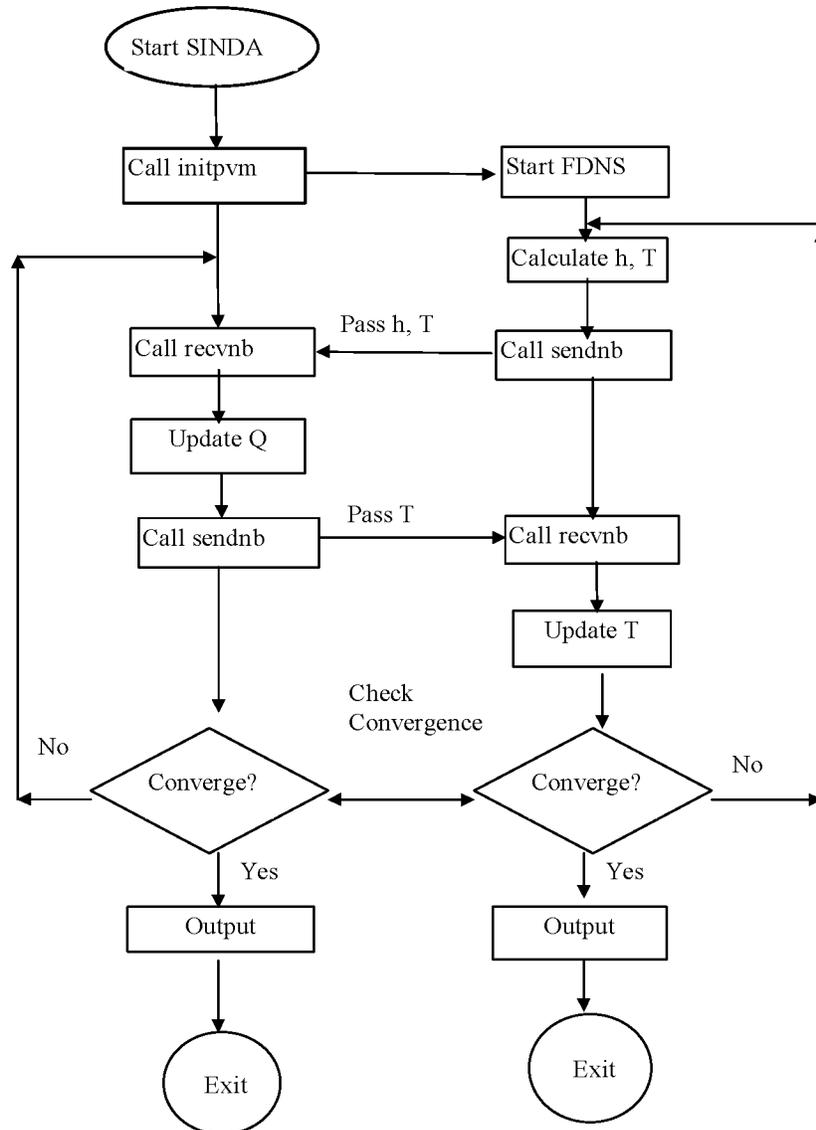


Figure 5. Communication between SINDA and FDNS

## NUMERICAL RESULTS

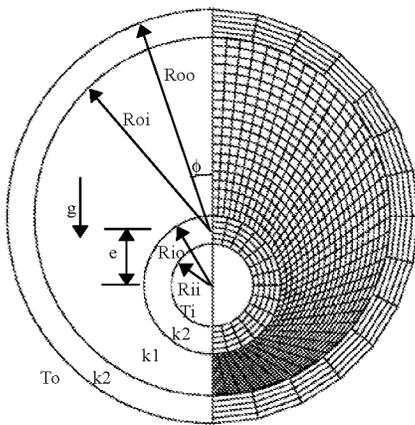
### THICK-WALLED ECCENTRIC TUBES CONJUGATE HEAT TRANSFER

The conjugate heat transfer between eccentric tubes is calculated by the integrated CFD-SINDA model, where the FDNS code is used to solve the natural convection and SINDA is used to solve the tube wall heat conduction. Figure 6 gives the geometry and the boundary conditions for this problem. The inner surface of the inner tube and the outer surface of the outer tube are kept at different fixed temperatures,  $T_i$  and  $T_o$ . The flow between the tubes is induced through the buoyancy force caused by temperature gradient. According to the experiments conducted by Kuehn and Goldstein [6], the Prandtl number of the fluid is 0.7, the Rayleigh number based on the length scale  $(R_{oi}-R_{io})$  and the temperature difference  $(T_i-T_o)$  is taken to be  $4.93 \times 10^4$ .

Two types of grid systems denoted as Grid 1 and Grid 2 are used. In Grid 1, the grid mesh is  $41 \times 21$  for the fluid and  $41 \times 6$  for inner and outer tube walls respectively. The grid lines are continuous at the interface. In Grid 2, the grid size is  $41 \times 21$  for the fluid but  $15 \times 6$  is used for the solid walls. The grid lines at the interface are discontinuous for Grid 2 and the patched grid technique is utilized.

For very large conductivity ratio  $C$  between solid and fluid (i.e.  $C=10^4$  for copper: air), the tubes are indicated to be isothermal. The numerically predicted temperature distributions at  $\phi=0^\circ$  and  $\phi=180^\circ$  are compared with the experimental data of Kuehn and Goldstein [6] in Fig. 7.  $\eta$  represents the distance from the innermost tube wall, which is normalized by the distance between the outermost and the innermost walls. The dimensionless temperature is defined as  $(T-T_o)/(T_i-T_o)$ . From the figure we can see that the Grid 2 can give same results as Grid 1 and all are in good agreement with experimental data.

We simulated two different conductivity ratio cases, one for  $C$  equals to 1 and another for  $C$  equals to  $10^4$ , the latter one corresponding to copper and air. Figure 8 shows the streamlines and the temperature contours. It is clear that the bigger the conductivity ratio the smaller the temperature gradient across the wall. When the  $C$  value is very large, the Bi number (defined as thermal conductivity ratio of solid to fluid) is very small and the wall is almost isothermal. That is the case for copper and air. This is verified in both Fig. 7 and Fig. 8 for different conductivity ratios. For the case of  $C=1$ , the transient solutions at  $t=1s, 5s, 10s$  are shown in Fig. 9.



$$\begin{aligned} R_{oi}/R_{io} &= 2.6, & R_{ii}/R_{io} &= 0.6, \\ R_{oo}/R_{io} &= 3.0, & e/(R_{oi}-R_{io}) &= 0.623 \end{aligned}$$

Figure 6. Geometry and Grid 2 for thick-walled eccentric tubes conjugate heat transfer.

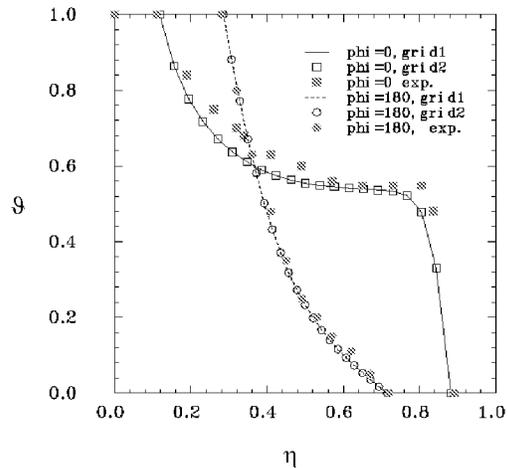


Figure 7. The dimensionless temperature vs. normalized distance between the tube walls for  $\phi=0^\circ$  and  $180^\circ$ .

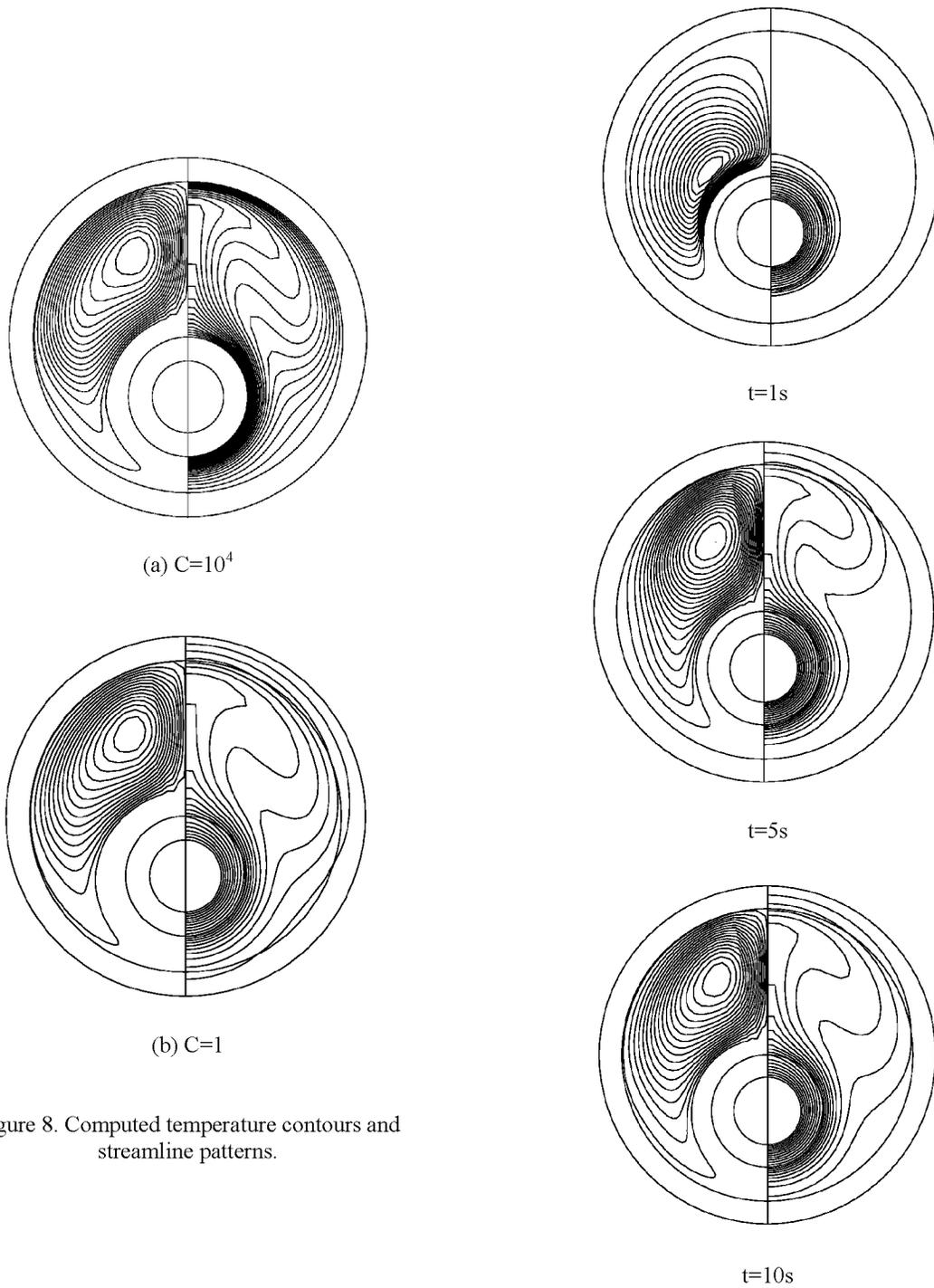


Figure 8. Computed temperature contours and streamline patterns.

Figure 9. Flow pattern and temperature field developing transient process for  $C=1$ .

### 3-D ROCKET ENGINE

Development of the algorithm for running multiple copies of FDNS for flow fields with a SINDA model for heat conduction is the solid component. The 3-D rocket engine model is selected as a test case. As shown in Fig. 10, this model consists of a hot gas flow part for FDNS nozzle flow model, a solid metal wall part for SINDA thermal model and two outer flow passages for FDNS cooling channel model. The nozzle gas flow is a compressible flow, and the cooling channel water flow is an incompressible flow. The grid size and initial conditions for each model are shown in Fig.10. For better observation, solid wall and cooling channel is showed apart from the nozzle.

SINDA runs with two copies of FDNS simultaneously. SINDA and each copy of FDNS use their own input and control data. SINDA runs as a master process and initialize two FDNS children processes. Figure 11 (a) shows the overall view of the temperature field including the hot gas nozzle flow solved by FDNS, the solid wall block solved by SINDA, and the outer cooling channel flow solved by FDNS. Enlarged views near location b, c, and d, indicated in Fig. 11(a), are shown in Fig. 11(b), 11(c), and 11(d) respectively. The gapes shown in Fig. 11 are caused by the differences in grid densities between the FDNS model and the SINDA model. Computationally, there is no gapes between these two models. The patched grid interface model enforces the energy conservation across arbitrarily patched interface grids between FDNS model and SINDA model. Reasonable temperature contours and variations across the fluid-solid interfaces are observed in Fig. 11.

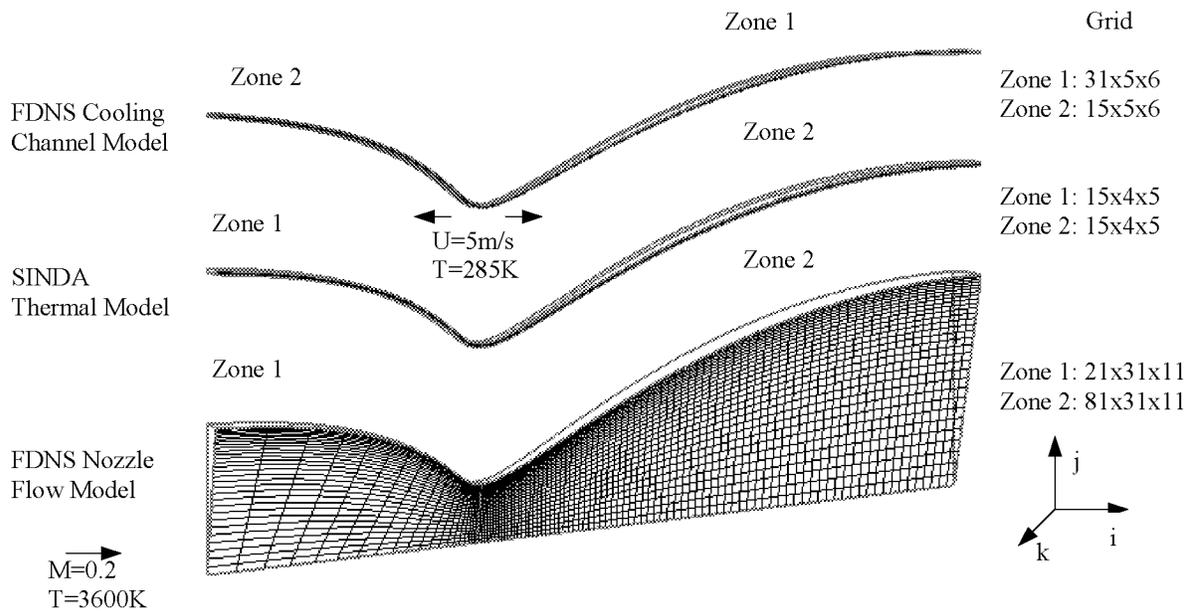


Figure 10: SSME nozzle and coolant channel flow configuration

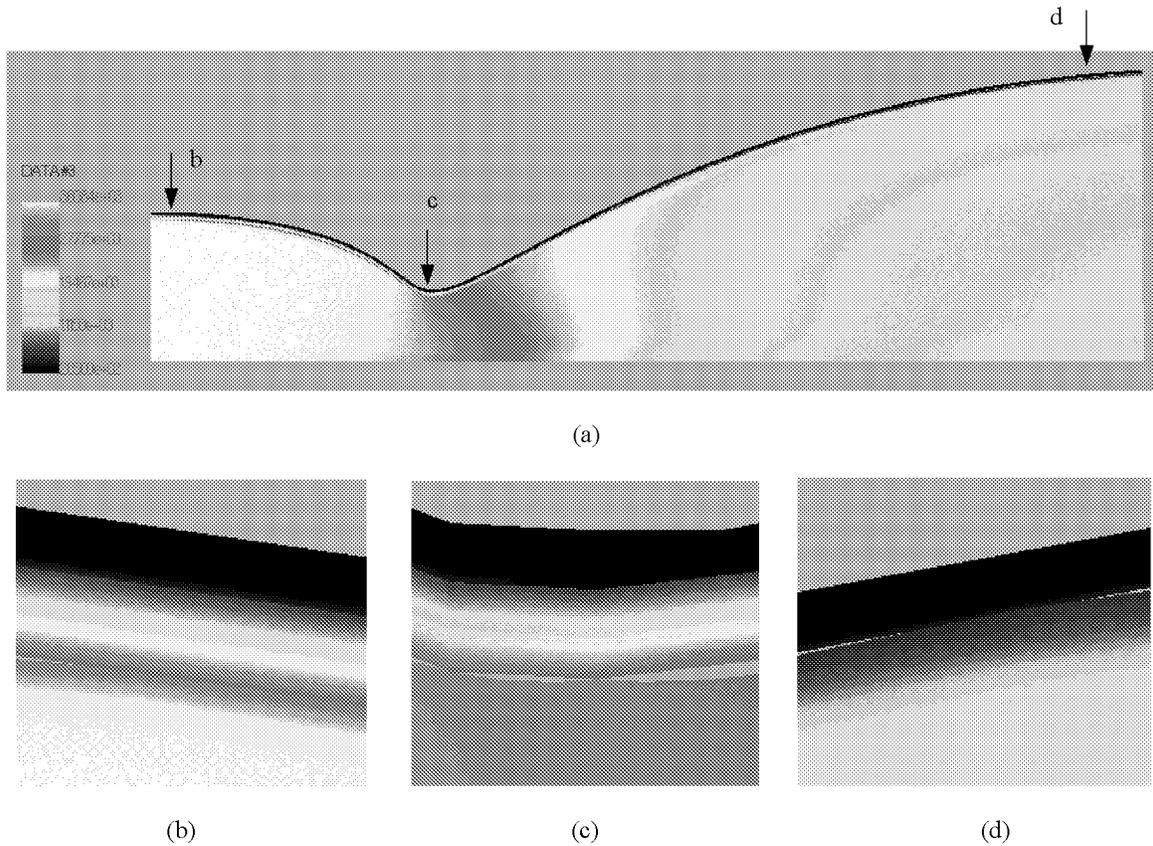


Figure 11: Temperature field of a 3-D SINDA/FDNS coupled solution.  
 (a). Full view. (b) (c) (d): Coolant, solid and main flow interfaces at different sections.

### 3-D SSME

A real geometry SSME nozzle flow with coolant channel flow and channel solid heat conduction is calculated using the new developed SINDA/multiple-FDNS simulation tool. Figure 12 shows the system configuration and grid mesh distribution. The coolant channel is showed apart from the SSME nozzle for clear observation. The hot gas flows in at uniform velocity with the Mach number of 0.2 and temperature of 3600K. Liquid hydrogen enters the coolant channel at velocity of 0.51m/s, which results in the coolant mass flow rate of 29.45lb/s for the 550-channel design. The hydrogen properties at pressure of 5000 psia and temperature of 54K are used in the calculations.

Figure 13 shows the temperature fields of 3-D SSME hot gas flow, coolant flow and coolant channel wall heat conduction by SINDA/multiple-FDNS coupled solution. Where, Fig. 13(a) is the full view in the middle cut of x-y plane. Fig. 13(b) gives the enlarged view of coolant, solid and main flow interfaces at the location (b) indicated in Fig. 13(a). Fig. 13(c) is the cross section temperature contours (in y-z plane) at the location (c). Due to the grid lines are discontinuous at the interfaces of coolant flow and channel wall, the temperature contour lines also show some discontinuity by the graphics package. The gape between hot gas flow and the channel wall is because of the x-location of the hot gas cross section and x-location of the channel wall and coolant flow cross section are not exactly the same.

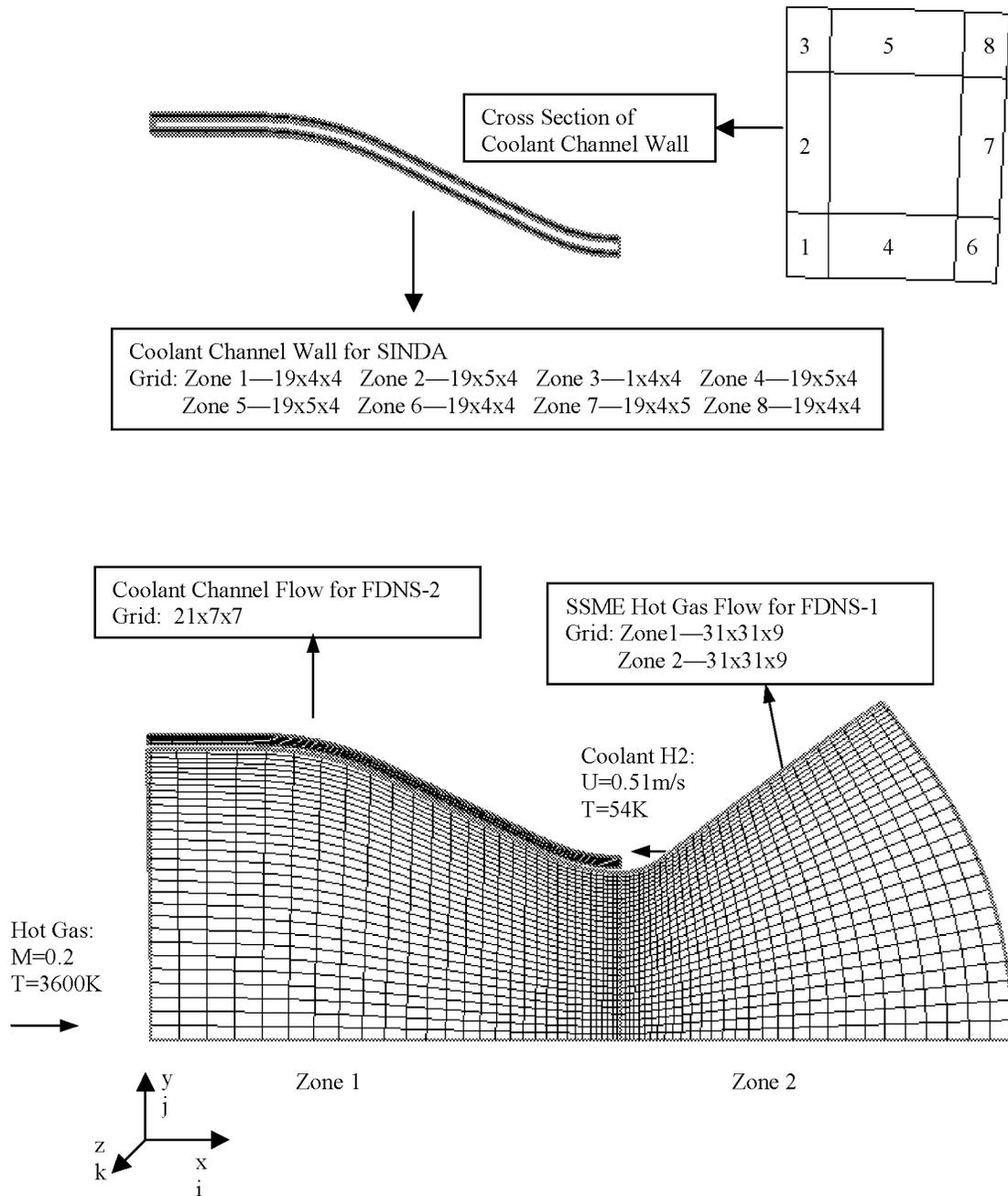
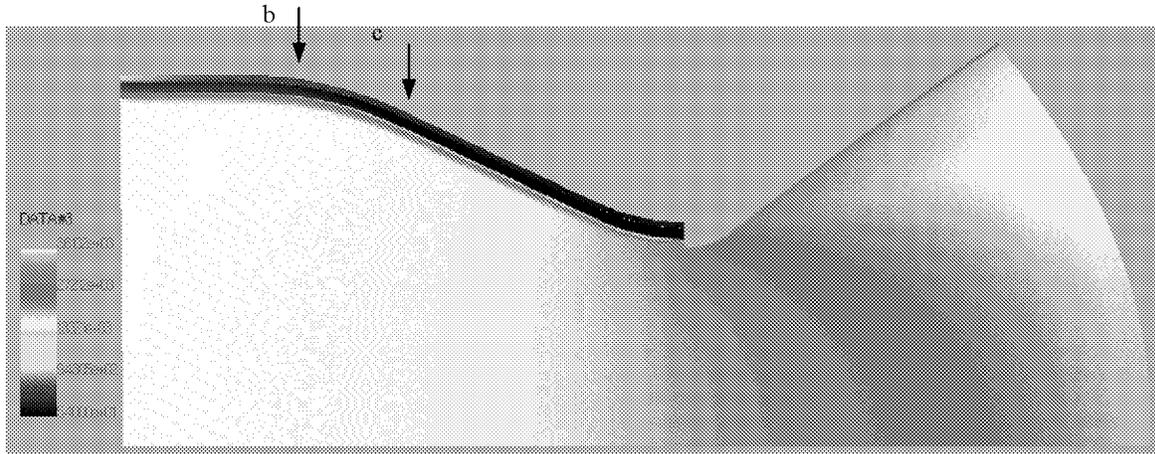
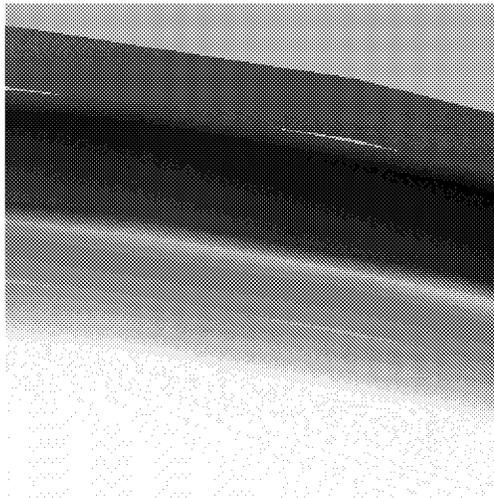


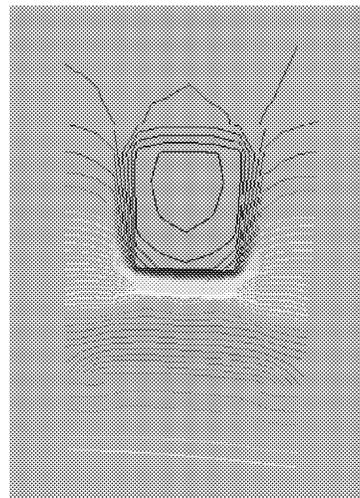
Figure 12. 3-D SSME nozzle flow with coolant channel configuration for SINDA/multiple-FDNS simulation.



(a)



(b)



(c)

Figure 13: Temperature fields of 3-D SSME hot gas flow, coolant flow and coolant channel wall heat conduction by SINDA/multiple-FDNS coupled solution.  
 (a) Full view. (b) Coolant, solid and main flow interfaces. (c) cross section temperature contours

## CONCLUSIONS

The integrated SINDA-FDNS model can effectively solve the coupled thermal-fluid problems. The fully conserved patched grid algorithm can ensure the energy conservation across the solid-fluid interfaces. The state-of-art parallel computing technique makes SINDA and FDNS running and exchanging information every time step. The successful implementation of SINDA model starting multiple copies of FDNS completed the entire model building. The resulted thermal-fluid model will be used for typical liquid rocket engine thermal-fluid analysis. It can serve as a reliable modeling tool in the aerospace and civil engineering industry.

## ACKNOWLEDGMENT

This work is performed under the support from NASA Marshall Space Flight Center with the contract NAS8-98066. The authors would like to acknowledge the technical suggestions from Dr. T. S. Wang of NASA Marshall.

## REFERENCES

1. SINDA/FLUINT-Systems Improved Numerical Differencing Analyzer and Fluid Integrator, Version 2.6, NASA/Cosmic Library.
2. Y.S. Chen: "FDNS-a General Purpose CFD Code, User's Guide, Version 3.0," ESI-93-01, May 1993.
3. G. H. Klopfer and G. A. Molvik: "Conservative Multizonal Interface Algorithm for the 3-D Navier-Stokes Equations," AIAA-91-1601.
4. Geist, A., Beguelin, A., Dongarra, J., Jiang, W., Manchek, R., and Sunderam, V., "PVM: Parallel Virtual Machine - A Users' Guide and Tutorial for Networked Parallel Computing," The MIT Press, Cambridge, MA, 1994.
5. J. Bonet and J. Peraire: "An Alternating Digital Tree (ADT) Algorithm for 3D Geometric Searching and Intersection Problem," International Journal for Numerical Methods in Engineering, Vol. 31, pp. 1-17, 1991.
6. Kuehn, J.H., and Goldstein, R.J., "An Experimental Study of Natural Convection Heat Transfer in Concentric Horizontal Cylindrical Annuli," *Journal of Heat Transfer*, Vol. 100, pp. 635-640, 1978.