ESI-TR-98-02

IN-31
048154
p. 95

INJECTOR DESIGN TOOL IMPROVEMENTS

Purchase Order Number: H-28621D

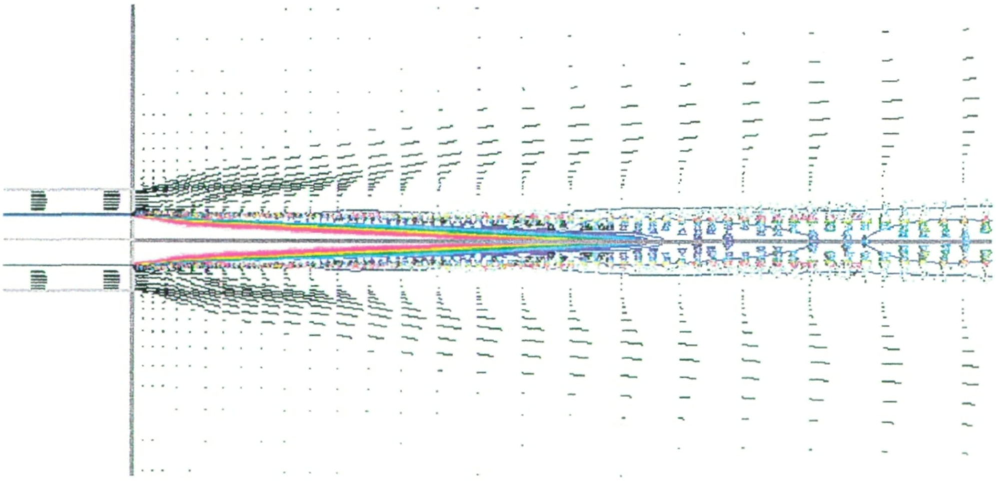
Final Technical Report

and

User's Manual for FDNS V.4.5

Prepared for

National Aeronautics & Space Administration
George C. Marshall Space Flight Center
Marshall Space Flight Center, AL 35812



by

Engineering Sciences, Inc.
1900 Golf Road, Suite D
Huntsville, AL35802

April 4, 1998

TABLE OF CONTENTS

1. PARALLEL CFD FDNS MODEL

1.1 ABSTRACT

The first section of this report describes the numerical implementation of parallel computing algorithm for a pressure-based, general-purpose flow solver, FDNS, using PVM libraries. The single-program multiple-data (SPMD) model of computing is implemented by the domain-decomposition technique for a multi-block Computational Fluid Dynamics (CFD) code. A general interface solver suitable for single or parallel computers has been developed. Numerical tests have been performed for several test cases and high parallel efficiency has been achieved in the current implementation.

1.2 INTRODUCTION

In recent years, Computational Fluid Dynamics (CFD) has became an important tool for engineering analyses to improve flow system performance. The demands of high performance and efficiency require a tremendous growth and availability of large scale computing applied to analysis and design. However, the rapid growth in the speed and storage of single processor computers has slowed down in recent years It now appears that using multiple processors, i.e. parallel computers, to work on the same problem is the only alternative. Parallel computers use standard chips and are therefore cheaper to produce than conventional supercomputers. CFD applications can now achieve significant performance improvements through the combined computational resources of a collection of computers [1-3].

There are two major developments of parallel computers: massively parallel processors (MPPs), such as Cray T3D and IBM SP2, and the widespread use of distributed computing. MPPs combine hundreds to thousands CPUs in a single cabinet shared same large memory. They offers enormous computational power and are used to solve computational grand challenge problems. Distributed computing is a process whereby a set of homogenous or heterogeneous computers connected by a network are used collectively to solve a single large problem. Unequaled computational power can be achieved by combing several MPPs for distributed computing.

There are several software packages have been developed for distributed computing. Among the most well known effort is Parallel Virtual Machine (PVM) software system [4] developed at the University of Tennessee and Oak Ridge National Laboratory (ORNL). It is a standard massage passing interface and enables distributed computing across a wide variety of computer types, including MPPs. PVM is built around the concept of a virtual machine which is a dynamic collection of (homogenous or heterogeneous) computational resource managed as a large single parallel computer So far, PVM is available for 40 different architectures combining UNIX workstations, shared memory machines, MPPs, and even WINDOWS 95/NT personal computers to one single parallel virtual machine. The most powerful feature of PVM is that it provides the message passing interface which lets the application assume to run on one single machine. PVM contains resource management and process control functions that are important for creating portable applications that run on clusters of workstations and MPPs. PVM is distributed freely and has been widely used in the computational applications communities around the world.

The FDNS [5] flow solver has been developed using single processor computers for fluid dynamics analyses. The algorithms developed for traditional serial computers may not run efficiently on parallel computer systems. Since FDNS is a multi-block code, a natural and efficient choice for parallel computing is to use domain decomposition. A pre-processing program for domain decomposition and generation of FDNS input data for all processes to be performed has been developed. The post-processor developed can collect the distributed output restart and plot files Load balance for network computers can be controlled by user input based on the grid point for each processor. A general interface solver suitable for single or parallel computers has been developed Different implementations have been investigated to achieve high performance, efficiency, and computational stability for FDNS code The message passing library PVM has been employed for information exchange of parallel computers The developed routines can be easily adapted to other message passing libraries

1.3 THE FDNS FLOW SOLVER

The FDNS [5] flow solver is a finite difference method for solving non-linear governing equations using non-staggered curvilinear grid system. The governing equations of FDNS are given below:

$$\frac{\partial \rho U}{\partial t} + \frac{\partial}{\partial x_i}\left(\rho u_i U + \mu_e \frac{\partial U}{\partial x_i} \right) = S_U \tag{1}$$

where the source terms including multi-phase effects are written as:

$$S_U = \begin{Bmatrix} M_p \\ -\frac{\partial P}{\partial x_j} + \frac{\partial}{\partial x_i}\left(\mu_e \frac{\partial u_i}{\partial x_j} \right) - \frac{2}{3}\frac{\partial}{\partial x_j}\left(\mu_e \frac{\partial u_i}{\partial x_i} \right) + D_i + M_p V_{p_i} \\ \frac{DP}{Dt} + \Phi + Q_i - V_{p_i} D_i + M_p\left(hv + Ur^2/2 \right) \\ \rho(\mathrm{P}_r - \varepsilon) \\ \rho\frac{\varepsilon}{k}\left[\left(C_1 + C_3 \mathrm{P}_r/\varepsilon \right)\mathrm{P}_r - C_2\varepsilon \right] \\ \omega_n, n = 1, \ldots\ldots N \end{Bmatrix}$$

This code provides multi-zone multi-block option for multiple species and finite rate chemistry reacting flow by solving the Navier-Stokes equations for the simulation of complex geometry flow problems. A Lagrangian-Eulerian statistical particle tracking method for spray combustion is employed in the FDNS to provide effects of momentum and energy exchanges between the gas phase and the particle phase. The particle trajectories are calculated using a one-step implicit method for several groups of particle sizes by which the drag forces and heat fluxes are then coupled with the gas phase equations The physical sub-models including droplet-turbulence interaction, particle-wall boundary conditions and particle accumulation rates, etc , are also incorporated A third-order TVD (Total Variation Diminishing) scheme similar to that of Chakravarthy and Osher [6] is employed to approximate the convection terms in the momentum, energy and species equations Viscous fluxes and source terms are discretized using a second-order central difference approximation. The time domain discretization of the present method allows the finite difference equations to be arranged into delta form for time-marching integration. Time-centered implicit time-marching schemes are employed for time accurate computations. A CFL number conditioned non-uniform time marching option can also be used for efficient steady-state solutions

For completeness, the time-marching scheme in FDNS is described below For convenience, transformed equations (from Xi to ξi system with J as the Jacobian of coordinate transformation) of Eq 1 is written as

$$\frac{1}{J}\frac{\partial \rho U}{\partial t} = -\frac{\partial F_i}{\partial \xi_i} + S_U = R_U \tag{2}$$

where F represents convection and diffusion fluxes First, Eq. 2 is discretized in time with a second-order time-centered scheme. That is

$$\frac{1}{J\Delta t}\left\{(\rho U)^{n+1} - (\rho U)^{n}\right\} = \frac{1}{2}\left({R_U}^{n+1} + {R_U}^{n}\right)$$

where superscripts n and n+1 represent old and new time levels respectively. If a sub-iteration procedure within a time step is applied, the following linearization can be incorporated

$$(\rho U)^{k+1} = (\rho U)^{k} + \rho^{k}\Delta U^{k}$$
$${R_U}^{k+1} = \left(\frac{\partial R_U}{\partial U}\right)^{k}\Delta U^{k} + {R_U}^{k}$$

where the superscript k denotes the k-th sub-iteration. With the above approximations, the final form of the time-marching scheme can be written as

$$\left\{\frac{\rho^k}{J\Delta t}-\left(\frac{\partial R_U}{\partial U}\right)^k\right\}\Delta U^k=-\frac{(\rho U)^k-(\rho U)^n}{J\Delta t}+\frac{R_U{}^k+R_U{}^n}{2}$$

The solutions at time level n+1 is then updated by:

$$U^{n+1}=U^{k+1}=U^k+\Delta U^k$$

When k = 1 is selected, a non-iterative time-marching scheme with a multi-corrector solution method can provide time accurate solutions for unsteady flow problems. The pressure based multi-corrector solution method is formulated using simplified perturbed momentum and continuity equations The simplified velocity correction equation can be written as.

$$\frac{\partial \rho u_i}{\partial t}\approx -\nabla P'$$

or, in discrete form,

$$u_i'\approx -\beta\frac{\Delta t}{\rho}\nabla P' \quad and \quad P^{n+1}=P^n+P' \tag{3}$$

where β represents a pressure relaxation parameter. The velocity and density fields in the continuity equation are then perturbed to form a correction equation Higher order terms are neglected. That is,

$$\frac{\partial \rho'}{\partial t}+\nabla(u_i\rho')+\nabla(\rho u_i')=-\left(\frac{\partial \rho}{\partial t}\right)^n-\nabla(\rho u_i)^n \tag{4}$$

Substituting Eq 3 into 4 and letting ρ' = P'/RT, the following all speed pressure correction equation is obtained

$$\frac{1}{RT}\frac{\partial P'}{\partial t}+\nabla\left(\frac{u_i}{RT}P'\right)-\nabla\left(\beta\Delta t\nabla P'\right)=-\left(\frac{\partial\rho}{\partial t}\right)^n-\nabla\left(\rho u_i\right)^n \qquad (5)$$

To provide smooth shock solutions the adaptive dissipation terms based on the pressure field is added to the right hand side of Eq. 5. Once solution of Eq. 5 is obtained, the velocity and pressure fields are updated using Eq. 3. The density field is then updated through the equation of state. The temperature field can also be modified by using a perturbed temperature correction equation. The entire corrector step is repeated 3 or 4 times such that the mass conservation condition is enforced before marching to the next time level

The discretized finite-volume equations can be represented by a set of linear algebra equations, which are non-symmetric, banded and positive-definite matrix system with sparsity patterns The preconditioned Bi-CGSTAB matrix solver is used to efficiently solve the linear algebra equations [7].

1.4 DOMAIN DECOMPOSITION AND COMMUNICATION

Domain Decomposition

In the domain decomposition approach, blocks containing large grid points can be subdivided to subblocks, and hence one or several blocks/subblocks can be assigned to each processor Maximum efficiency can be achieved by giving each processor the amount of work to do according its CPU speed and memory size. Therefore, the same code runs on all processors on its own set of data Exchange of data between processors and/or storage is necessary to enforce the boundary conditions at the divided interfaces.

An example of square 2D computational domain and mesh is shown in Fig 1. To use 4 same speed processors for parallel computing, 4 equal blocks are divided as shown in Fig. 2 The divided interfaces present zonal boundaries in multi-block FDNS code. The algorithm employed in FDNS code for updating zonal boundary conditions has be re-constructed to allow the data exchange between each twin interfaces at one divided interface during each solution iteration.

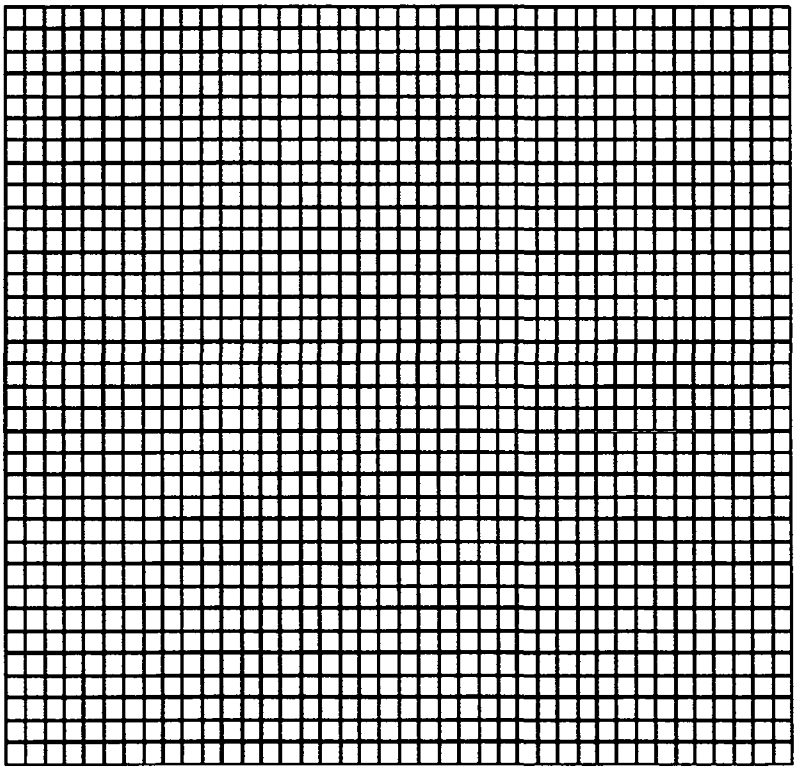


Fig. 1 Initial computational domain and mesh.

Communications

The communication routines for parallel computing include the following sections:

1. Process initialization:
    - Initialize the master (or parent) process by starting the code on the master machine.

- Start the other (slave or children) processes on the other machines ın the network by the master process.
- Obtain the task identifiers for each process

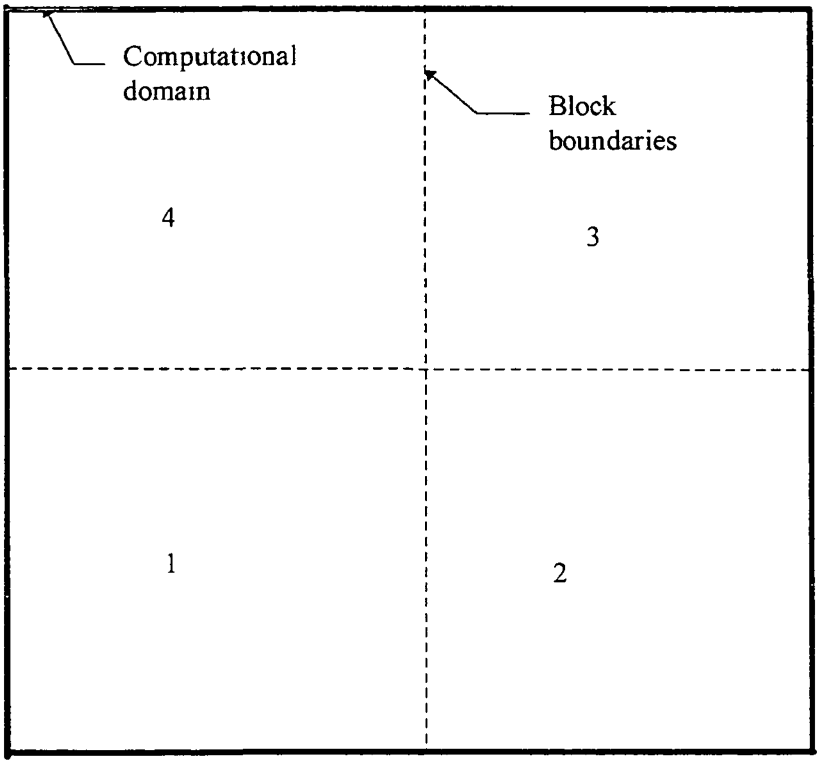


Fig. 2 Illustration of a square 2D computational domain divided into 4 blocks.

2. Information sending:
   - Prepare the buffer which ıs to accept the ınformation to be transferred.
   - Pack the information in an array.
   - Send the buffer to the process with specified task ıdentifier.
3. Information receıving:
   - Receıve the massage from the process with matched task identifier
   - Unpack the message in exactly the same way as it was packed in the sending process
4. Message broadcastıng.
   - Broadcast some messages from the master to all other processes

5. Message collecting·
   - Collect some values from all other processes by the master process
6. Information exchange·
   - Exchange the zonal boundary conditions for solution field variables, such as density, velocity and pressure etc. during the iterative or time marching process. The coupling terms between velocity and pressure correction also must be exchanged at the zonal interface for the solution of pressure correction equation. Because FDNS is an implicit code, the information exchange for the zonal boundaries must be included in the matrix solver to achieve high convergence of the solution.
7. Process exit
   - Collect output files and remove temporary files.
   - Terminate the execution at each processor.
   - Exit the parallel process from the network.

For distributed computing, communication between processor is necessary. Each processor needs to store data from one or more layers of cells on the other side of the interface. *Local communication* takes place between processors operating on the twin blocks. Each process will build up a send table, which contains all cells from which information is needed by another block. The corresponding process ID, where the data has to be sent to, must be specified During the iterative process the two tables allow an efficient and secure communication of boundary values through PVM libraries. *Global communication* collects some information from all processes in a master processor and broadcasting some information back to other processors. Such information can be residuals and reference values.

The information exchanges performed in section 2,3 and 6 are local communication between each pair processors. The message exchanges passed in section 4 and 5 are global communications. Parallel efficiency can be improved by minimizing both communication overheads for systems with slow communication network

Exchange of Zonal Boundary Conditions

The new boundary information exchange subroutines have been developed and can be applied for single or parallel computers based on the current FDNS code For a general zonal interface boundary, twin interfaces are located at each neighboring block. If the other block is located in the same process, no communication is required to exchange the boundary conditions. Otherwise, the interfaces involved in the two different processes at distributed computing have to exchange data through network communications A general interface solver suitable for parallel computing is described below

1 Specify each process number for the two interfaces at each zonal boundary This can be done during the data input preparing stage through a user controlled program for domain decomposition

2. Pack data at the inner layer 3 or 4 if the interface 1 or 2 is located in the current process. Fig 3 illustrates the interfaces and their inner layers

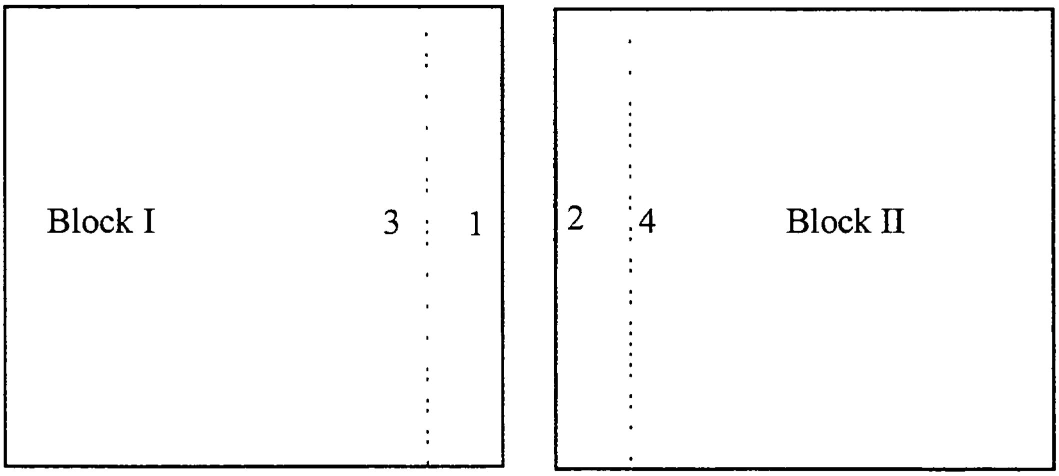


Fig. 3 Boundary interfaces and inner layers.

3 Send the packed information to other process and at the same time receive the packed information sent from the other process only if this interface in another process.

- The zonal boundary is defined as at least one interface is located in the current process
- No communication or data exchange is required if both interfaces 1 and 2 are in the current process.
- If interface 1 is in the current process and interface 2 is not, the data at inner layer 3 of interface 1 is packed and sent to other process which contains interface 2. The packed data for inner layer 4 of interface 2 sent from another process will be received by the current process.
- The similar operation takes place when interface 2 is in the current process and interface 1 is not.

4. Interpolate the boundary values through linear interpolation using the grid spacing
5 Update the boundary solutions accordingly.

Flowchart for the Parallel Computing

An overview flowchart for the parallel domain decomposition approach is presented in Fig 4. The flowchart illustrates the procedures for parallel computing and massage passing calls for use in parallel computers.

The Parallel Efficiency

The two parameters, speed-up factor and efficiency, as described by Ferziger and Perić [1], are usually used to measure the performance of parallel programs:

$$S_n = \frac{T_s}{T_n}, \qquad E_n = \frac{T_s}{nT_n}$$

Here T_s is the execution time for the best serial algorithm on a single processor and T_n is the execution time for the paralleled algorithm using n processors

1.5 NUMERICAL EXAMPLES

The current methodology ıs developed only for continues matched interfaces The computer code has been ımplemented on maȷor UNIX workstations, such as SGI, SUN, and IBM, and PC workstatıons using LINUX and WINDOWS NT operating systems. The parallel efficiency will be addressed through the following numerical test cases.

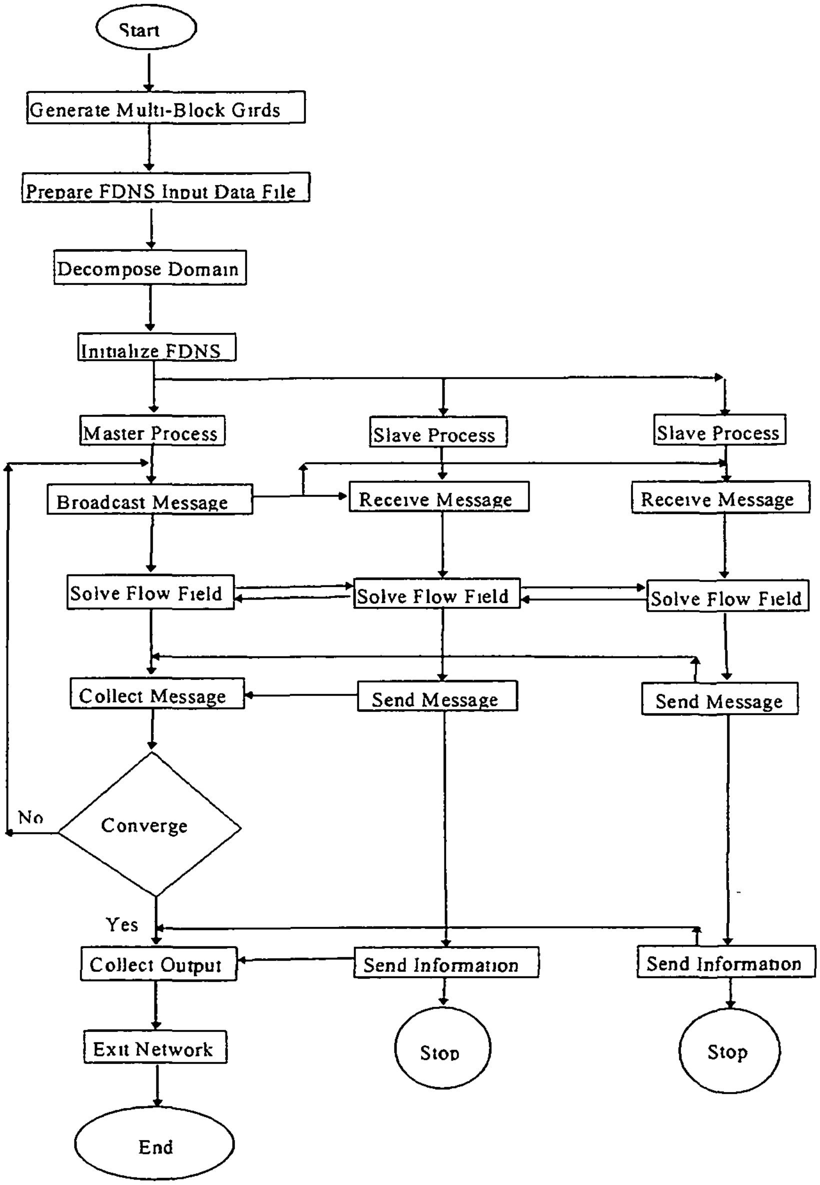


Fig 4 Overvıew flowchart for FDNS code usıng parallel computıng

Two-Dimensional Driven Cavity Flow

A four block incompressible laminar driven cavity flow with Reynolds number 1000 is used for two-dimensional test cas Fig. 5 presents the velocity vectors solved on single host. The same problem is solved using parallel computing method with 4 zones using 4 processors. The computed velocity vectors are shown in Fig. 6. It can seen that both solutions using serial and parallel computing technique agree very well.

Two grid sizes with 81×81 and 161×161 have been conducted for 1000 and 200 iterations respectively. The results are shown in Tab. 1 It is found that higher speed-up factor and efficiency have been achieved for large grid size The communication times between zonal interfaces take about 35 8% and 23 3% for small and large grid sizes used in this study.

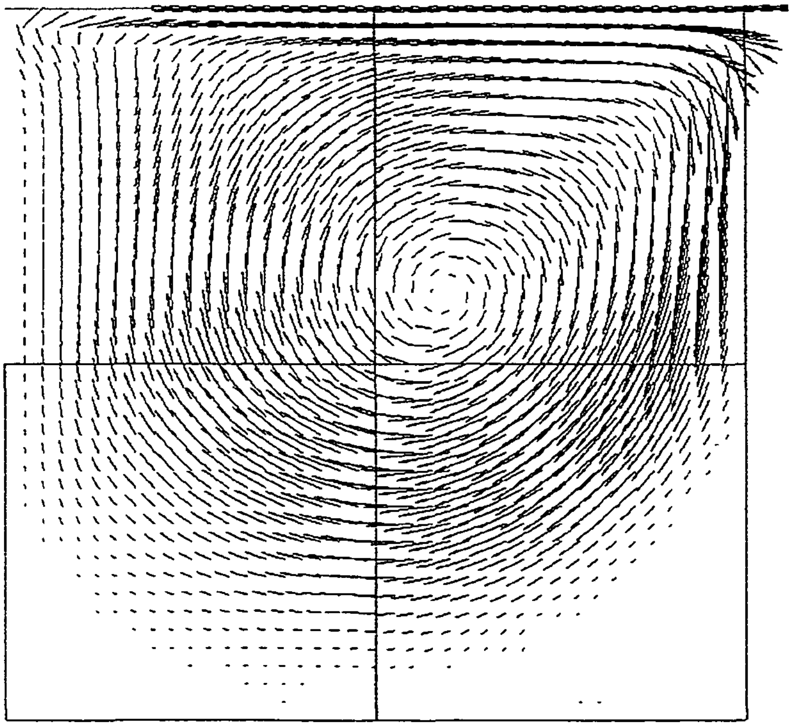


Fig. 5 Velocity vectors computed using single host for driven cavity flow.

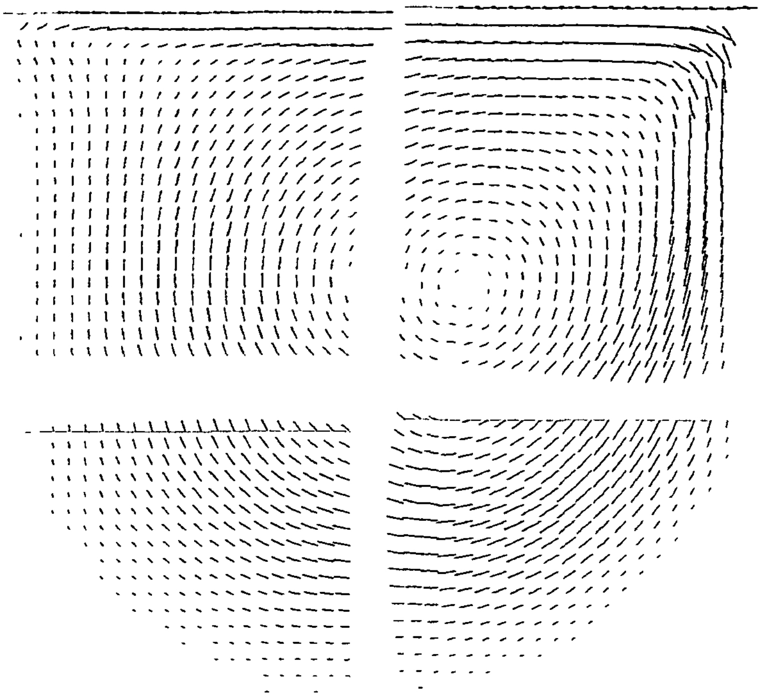


Fig 6 Velocity vectors calculated by parallel computing method.

Tab. 1 Performance of parallel computing using 4 processors on SGI's Power Challenge computer.

Grid	T_s(s)	T_n(s)	S_n	E_n (%)
81×81	503.0	1007.7	2.00	50.0.
161×161	207.0	592 2	2.86	71.5

Three-Dimensional Blunt Body Flow

Parallel computing for FDNS code has been used for compressible flow over a 3-D blunt body. Three block grids (41×61×81 each block) with total 607,743 grid points and twelve block grids (41×61×21 each block) with total 630,252 grid points have been used to perform this calculation using 3 and 12 processors in SGI's Power Challenge computer respectively. The computed mach number contours and block outlines for 3 processors are presented in Fig. 7. The block outlines for 12 processors are shown in Fig. 8.

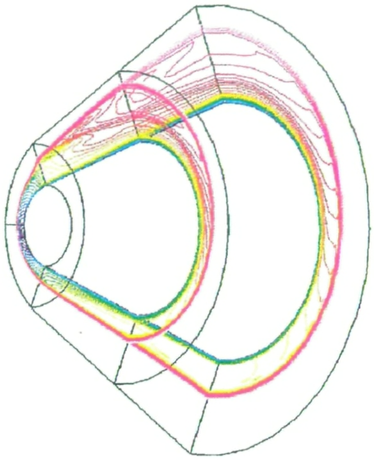


Fig. 7 Mach number contour fo ra 3-D blunt body flow (M=4.0, Attack Angle=10degree)

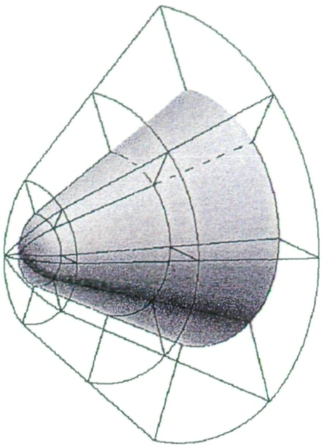


Fig. 8. Illustration of 12 blocks used for blunt body calculation

The test results at 70 iterations are presented in Tab.2 using 3 and 12 processors. The parallel speed-up factors are 2.82 and 10.73 respectively, which corresponding the parallel efficiency 94.0 and 89.4 respectively. It is shown in Tab. 2 that the execution time for 3 blocks is longer than 12 blocks both running at serial (single CPU) computing. It is attributed to the CG solver implemented in FDNS code. Generally speaking, the CG solver converges faster for each individual block and hence takes less global CPU time while the computational domain is decomposed because the solutions at the decomposed domain interfaces are updated explicitly. This may affect the global convergence rate of CFD code based on the nature of the flow fields, time step size or relaxation factors employed. For this particular super sonic case, it is found that the explicit exchange of boundary conditions at the zonal interfaces slows down global convergence rate slightly.

Tab 2. Performance of parallel computing for 3-d blunt body flow on SGI's Power Challenge computer.

Processors	T_s(s)	T_n(s)	S_n	E_n (%)
3	6781	2410	2.82	94.0
12	6074	566	10.73	89.4

Three-Dimensional Propeller Flow

The parallel efficiency is also examined for heterogeneous network computers and compared with the shared memory SGI's Power Challenge machine. The test case ıs a three-dimensional incompressible propeller flow as shown in Fig 9 Only 1/5 domain is computed due to the cyclic boundary condition. The network computers used are SGI's Indigo2, SUN's Ultra-2 wıth dual CPUs, and two generıc PCs with Pentium II 266MHz processor and WINDOWS NT/4.0 operating system. The RAMs are enough to make sure no swapping occurred during the computing. The low speed ethernet cards with the transferring rate 10MB/s are used to connect the network computers due to availability by authors. The grid points, CPU times for 800 iterations and network computers are shown in Table 3. The executıon tıme is 15,162s. Due to the grid numbers and CPU speeds are not balanced for this calculation, ıt ıs hard to compare the parallel efficiency and speed-up factor It is traced that SGI's machıne is at idle status in most of execution tıme due to very low load. It can be found from Table 3 that SUN's machine is a little faster than PCs. Because the most load or busy process is at SUN CPU1, the computıng efficiency is determined by this process. The fair estmate of the parallel effıcıency for this case can be obtained by comparing the most CPU time and execution tıme. Such efficiency is 84.1% for thıs network computers running at different load.

Table. 3. Grıd points and CPU times for network computers.

	SGI	SUN	SUN	PC1	PC2
Grid	51x10x	51x51	51x45x	51x45x	51x45
CPU	4,578	12,750	10,674	12,432	11,712

The same test case is also conducted in SGI's Power Challenge machine. The most CPU time spent compared with the execution time is 95.4%. Hence, the higher parallel efficiency is achieved in shared memory computer due to its fast communications.

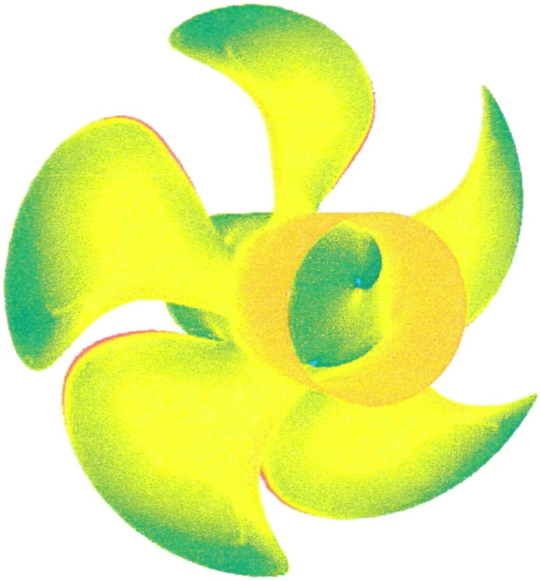


Fig. 9. Pressure contour on propeller surface.

1.6 REFERENCES

1. Ferziger, J.H., and Perić, *Computational Methods for Fluid Dynamics*, ISBN 3-540-59434-5 Springer-Verlag Berlin Heidelberg New York1996.
2. Blosch, E.L., and Shyy, W., "Scalability and Performance of Data-Parallel Pressure-Based Multigrid Methods for Viscous Flows," *J. of Computational Physics,* Vol. 125, pp.338-353, 1996.
3. Sawley, M.L., and Tegnér, J.K., "A Comparison of Parallel Programming Models for Multiblock Flow Computations," *J. of Computational Physics,* Vol. 122, pp.280-290, 1995.

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. Chen, Y S , "FDNS - a General Purpose CFD Code, User's Guide, Version 3 0," ESI-TR-93-01, May 1993.

6. Chakravarthy, S.R., and Osher, S , "A New Class of High Accuracy TVD Schemes for Hyperbolic Conservation Laws," AIAA Paper 85-0363, 1985.

7. Van Der Vorst, H.A., "BI-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems," *SIAM J. Sci. Stat. Comput.*, Vol. 13, No.2, pp.631-644, Jan. 1992.

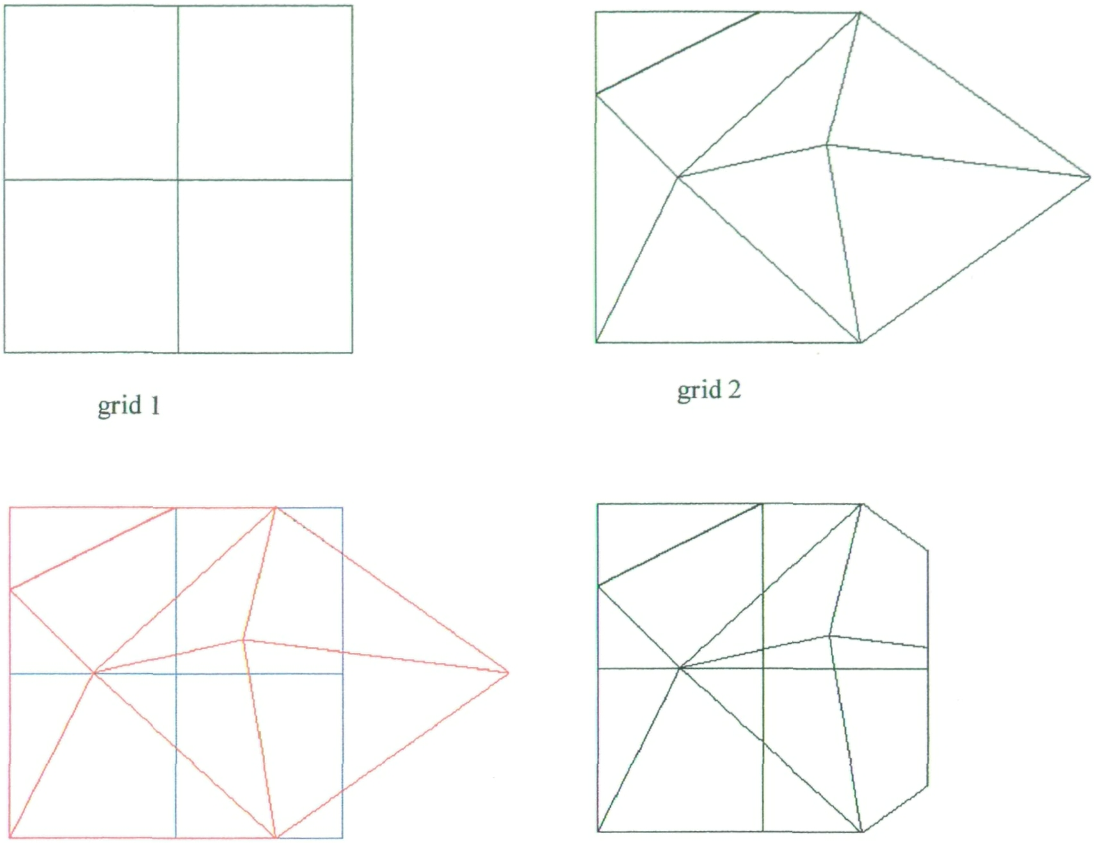
2. GENERAL 3-D PATCHED GRID MODEL

For complex geometry flow problems, local high resolutions mesh blocks are commonly used to reveal detailed flow physics. This mesh block may be surrounded by course-grid blocks with discontinuous grid lines across the block interface. Major concern for this type of block interface treatment lies in the mass conservation property that direct interpolation does not always guaranty conservation. The only way that the conservation property can be ensured are described in the following steps.

1. Reconstruct a new interface based on the original pair of interfaces by identifying the interface cells that are bounded by the mesh lines of the original interfaces. This will results in the best cell resolutions to describe the new interface.
2. Integrate mass fluxes at the interface based on the cells identified on the new interface. This ensures that the best resolutions of mass fluxes are preserved.
3. Use the new mass fluxes for the continuity, momentum and other transport equations.

The high-cell-resolution interface reconstruction procedure is required only once at the start of CFD problem initialization. Therefore, the procedure can still be very efficient.

A general patched-grid reconstruction module is developed. This module takes interface data from separate mesh blocks and reconstruct the common interface using polygon elements based on the mesh line interceptions of the two original interface patches. Figure 10 illustrates a sample case of getting interface data from Grid 1 and Grid 2 to find polygon information and obtain the final common interface for Grid 1 and Grid 2. This common interface gives the highest resolution one can obtain based on the two grids. Therefore, conservation is achieved using this common interface for flux calculations.



grid 1

grid 2

overlap of grid 1, 2

interface

Figure 10: Patched grid plane surface interface

The patched grid interface module developed by ESI is a 3-D general-purpose module using a fully conservative method. The multi-block patched interfaces are used as input to this module to construct an unstructured-cell common interface such that conservative feature of the flow variables can be enforces across the patched boundaries. This module is mostly self-sufficient and can be easily modified to integrate with other existing CFD codes. The main features are summarized below:

1. Permits the use of discontinuous as well as continuous multi-block grids.
2. Takes both structured and unstructured grids.

3. Applicable for both 2-D and 3-D calculations.
4. The interfaces can be planner or curved.
5. The interfaces number can be as many as necessary.
6. Applicable to turbomachinery cyclic boundary and stator-rotor moving boundary conditions.

The major subroutines that are interfaced with the FDNS flow solver are described in the following sections.

2.1 SUBROUTINES FOR CONSTRUCTING INTERFACE

Face Data Preparation

1. RGFDNS(IDIM,L,M,N,IDSN,IJSN,IJSS,IJST,IKSS,IKST,
   X,Y,Z,IT2,JT2,X2,Y2,Z2,XREF)
   ----------Get structured face grid from FDNS

Variables In:

IDIM : dimension

L, M, N, IDSN, IJSN, IJSS, IJST, IKSS, IKST : parameters in FDNS used to determine the block boundaries

L : total grid number of the previous zone

M : I-max of current zone

N : total grid number on one I-J plane in current zone

IDSN : boundary facing index 1: east, 2: west, 3: north, 4: south,
5: top, 6: bottom

IJSN : I, J, or K location (depends on IDSN) of the boundary

IJSS, IJST : boundary starting and ending indices (for I or J)

IKSS, IKST : boundary starting and ending indices (for J or K)

X( ), Y( ), Z( ) : grid coordinates from FDNS

XREF : reference length from FDNS

Variables Out:

IT2, JT2 : I-max and J-max of the boundary face 2

X2( ), Y2( ), Z2( ): coordinates of the boundary face 2

2. UNSTRT(IDIM,IT2,JT2,NODE2,NELE2,INODE2,IDNODE2,NINODE2)

----------Change the structured grid from FDNS to unstructured grid data format

Variables In:

IDIM, IT2, JT2

Variables Out:

NODE2 : node number in face 2

NELE2 : cell number in face 2

INODE2( ) : record the node number on each cell in face 2

IDNODE2( ) : record the node ID on each cell in face 2

NINODE2 : sum of the node number on every cell in face 2

3. VECTNR(NELE1,INODE1,IDNODE1,X1,Y1,Z1,VECTN1)

----------Calculate the normal vector on every cell for 3-D curved face application

Variables In:

NELE1, INODE1( ), IDNODE1( ), X1( ), Y1( ), Z1( )

Variables Out:

VECTN1( ): normal vector on each cell for face 1

Construct Interface

1. PATCH2(NODE1,NELE1,INODE1,IDNODE1,X1,Y1,
NODE2,NELE2,INODE2,IDNODE2,X2,Y2,
NELE,NODE,INODE,IDNODE,NINODE,XND,YND,
XAREA,ISUB1,ISUB2)

---------Construct the interface (straight line or curved line) for 2-D cases

Variables In:

NODE1 : node number in face 1
NELE1 : cell number in face 1
INODE1( ) : record the node number on each cell in face 1
IDNODE1( ) : record the node ID on each cell in face 1
X1( ), Y1( ) : x, y coordinates for grid in face 1
NODE2, NELE2, INODE2, IDNODE2, X2( ), Y2( ) used for face 2

Variables Out:

NODE, NELE, INODE( ), IDNODE( ), NINODE : used for interface
XND( ) , YND( ) : x, y, coordinates for every interface node
XAREA( ) : area of every cell element on the interface
ISUB1( ) : pointer to indicate which cell on face 1 this interface element corresponds to
ISUB2( ) : used for face 2

2. PATCH3(NODE1,NELE1,INODE1,IDNODE1,X1,Y1,Z1,VECTN1,
NODE2,NELE2,INODE2,IDNODE2,X2,Y2,Z2,VECTN2,
IT2,JT2,NODE,NELE,INODE,IDNODE,NINODE,XND,YND,ZND,
XAREA,ISUB2,ISUB1,ICUR)

---------Construct the interface (planner or curved face) for 3-D case

Variables In:

NODE1 : node number in face 1
NELE1 : cell number in face 1

INODE1( ) record the node number on each cell ın face 1

IDNODE1( ) : record the node ID on each cell ın face 1

VECTN1( ) : normal vector on each cell in face 1

NODE2,NELE2,INODE2,IDNODE2,X2,Y2,Z2,VECTN2 used for face 2

IT2, JT2 : maxımum index ın I and J dırection

ICUR . 0---planner face for 3-D case, 1---curved face for 3-D case

Variables Out:

NODE, NELE, INODE( ), IDNODE( ), NINODE : used for interface

XND( ) , YND( ) , ZND( ) : x, y, z coordınates for every ınterface node

XAREA( ), ISUB1( ), ISUB2( ) same meaning as ın PATCH2

Pass Interface Information To Main Solver

1 PATCHGZ(IZF,IDSIN,IDIM,ICUR,X,Y,Z,ANGLE,IDF,
IZS,JZS,KZS,IZT,JZT,KZT,
IZB1,IZF1,IJZ11,IJZ12,JKZ11,JKZ12,
IZB2,IZF2,IJZ21,IJZ22,JKZ21,JKZ22,
NELE1,NELE2,NELE,ISUB1,ISUB2,XAREA,
AREA1,AREA1P,AREA2,AREA2P,IC,
DELANG,DELDIS,IROT)

------------Get patched grid information on zonal ınterface in FDNS

Variables In:

IDIM, ICUR

IZF · Process ID for PVM

IDSIN : interface ID

IZS( ), JZS( ), KZS( ), IZT( ), JZT( ), KZT( ) : parameters from FDNS

IZS( ) : total grid number of the zones before current zone

JZS( ) · I-max of current zone

KZS( ) : grıd number on I-J plane ın current zone

IZT( ) : I-MAX

JZT( ) : J-MAX

KZT( ) : K-MAX

IZB1, IZF1, IJZ11, IJZ12, JKZ11, JKZ12

IZB2, IZF2, IJZ21, IJZ22, JKZ21, JKZ22

zonal indices defined in FDNS for interface boundaries

IZB1 : zone ID

IZF1 : boundary facing index 1: east, 2: west, 3: north, 4: south
5: top, 6: bottom

IJZ11, IJZ12 : boundary starting and ending indices (for I or J)

JKZ11, JKZ12 : boundary starting and ending indices (for J or K)

X( ), Y( ), Z( ) : grid coordinates from FDNS

ANGLE, IDF, IC, DELANG, DELDIS are used for turbomachinery cases

Variables Out:

NELE1, NELE2, NELE, ISUB1( ), ISUB2( ), XAREA( )

AREA1( ) : area of every cell element in face1

AREA1P( ) : sum of the interface cell area that correspond to the same cell in face 1

AREA2( ), AREA2P( ) : used for face 2

IROT( ) : used for turbomachinery

2.2 SUBROUTINE FOR CONSERVATIVE VARIABLE TRANSFORMATION

Get Variables at Cell

1. FACEVL(L,M,N,IDSN,IJSN,IJSS,IJST,IKSS,IKST,F,IJLO,F2,IC)

-----------Get variable value at boundary face from FDNS

Variables In:

L, M, N, IDSN, IJSN, IJSS, IJST, IKSS, IKST

F( ) : variable at each grid in whole domain from FDNS

IJLO( ) : node type(inner node, boundary node, wall node) specifier from FDNS

IC : 0 -- for interface FDNS with other code (e.g. SINDA)

1 -- for zonal interface in FDNS

Variable Out:

F2( ) : variable value at each cell in face 2

Information Change Across The Boundary

1. FCHANGZ((IC,NELE,XAREA,IDF,AREA1,AREA1P,AREA2,AREA2P,
ISUB1,NELE1,F1,ISUB2,NELE2,F2)

-----------Information change across zonal boundary in FDNS

Variables In:

IC: 0---for p, t 1---for variables u, q 2----for v 3---for w

NELE, XAREA( ), IDF, AREA1( ), AREA1P( ), AREA2( ), AREA2P( )

ISUB1( ), NELE1, F1( )

ISUB2( ), NELE2, F2( )

Variables Out:

F1( ), F2( )

Pass Updated Variables To Main Solver

1. GRIDVL(L,M,N,IDSN,IJSN,IJSS,IJST,IKSS,IKST,F2,IJLO,F,IC)

--------Translate the updated variable from face value array to domain value array

Variables In:

L, M, N, IDSN, IJSN, IJSS, IJST, IKSS, IKST, F2( ), IJLO( ), IC

Variables Out:

F( )

2. PATCHFZ(IZF,IDSIN,IDIM,F,VM,ANGLE,IDF ,
IZS,JZS,KZS,IZT,JZT,KZT,IJLO,
IZB1,IZF1,IJZ11,IJZ12,JKZ11,JKZ12,
IZB2,IZF2,IJZ21,IJZ22,JKZ21,JKZ22,
NELE1,NELE2,NELE,ISUB1,ISUB2,XAREA,
AREA1,AREA1P,AREA2,AREA2P,IROT,DELANG,IC)

------------Do conservative variable interpolation across zonal interface in FDNS

Variables In:

IC: 0---for variable p, t 1----for u, q 2---for v 3---for w

IZF, IDSIN, IDIM, F( ) , ANGLE, IDF ,
IZS( ), JZS( ), KZS( ), IZT( ), JZT( ), KZT( ), IJLO( ),
IZB1, IZF1, IJZ11, IJZ12, JKZ11, JKZ12,
IZB2, IZF2, IJZ21, IJZ22, JKZ21, JKZ22,
NELE1, NELE2, NELE, ISUB1( ), ISUB2( ), XAREA( ),
AREA1( ), AREA1P( ), AREA2( ), AREA2P( ), IROT( ), DELANG
VW( ) : v or w from FDNS used for turbomachinery

Variables Out:

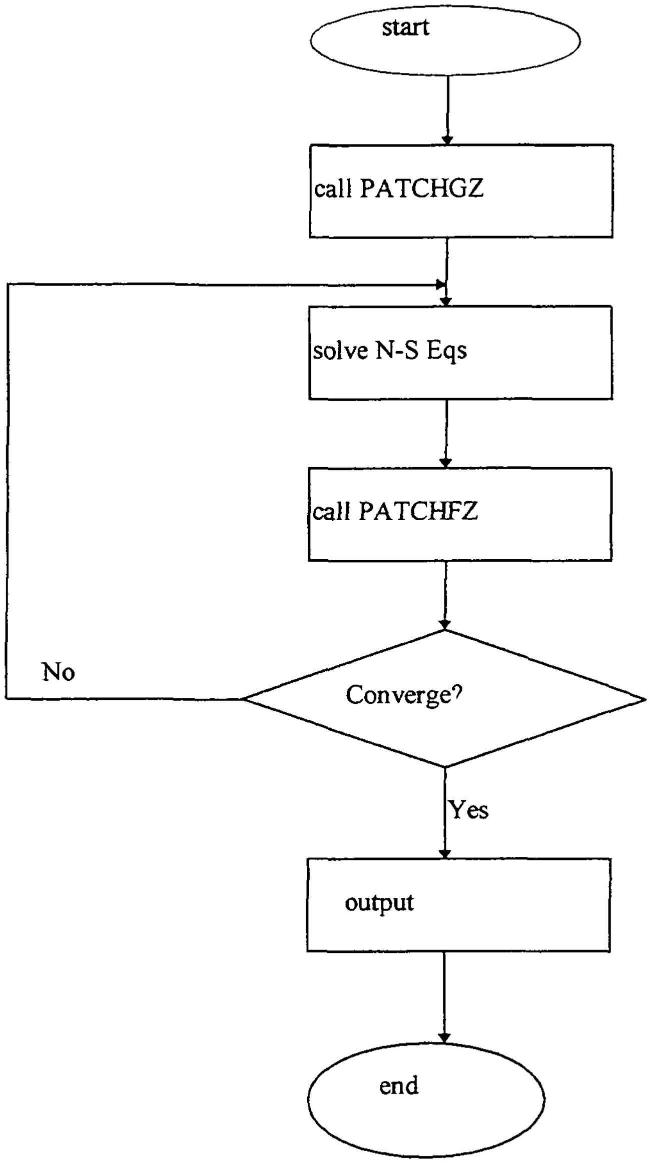
F( )

IDF=20~30 corresponds to the cases with continuous-grid interfaces;

IDF=21 or 31: general zonal interface cases;

IDF=23 or 33: for supersonic marching cases (not tested yet);

For turbomachinery applications, IDF=22 or 32 is for cyclic boundary conditions; IDF=34 or 35 is for stator-rotor moving boundary, while IDF=35 is for the case when grid lines are radially inlined, i.e. 3D interfaces are treated as 2D line interfaces.



(Flow chart for multi-zone patched-grid approach in FDNS)

Benchmark validation cases for the current patched grid method is given in the follow sections which include a wide range of 2-D and 3-D test problems.

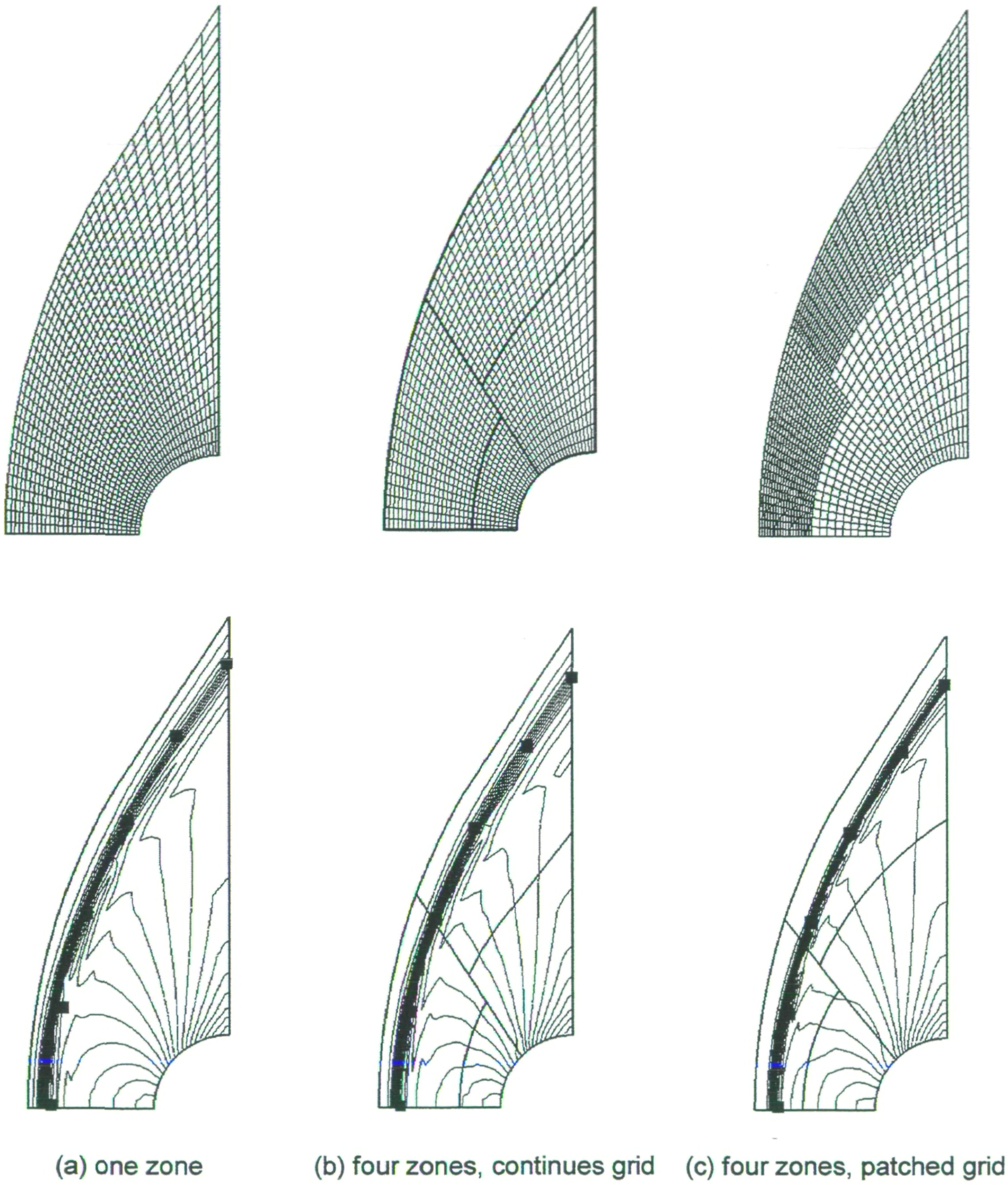
2.3 BENCHMARK TEST CASES FOR THE PATCHED GRID MODEL

Supersonic Stream Past A Cylinder

A cylinder in supersonic stream (M = 2) is calculated using different mesh system. The grid and corresponding pressure contour lines at steady state are shown in Figure 11~13. The captured shock location is compared with the prediction by a correlation[1] based on experiment (dark square)

The single zone with grid number of 41x31 is first calculated as in Figure 11 (a). Then, the calculation domain is cut into four zones and the total grid number is kept as same as the single zone situation. In this case (shown in Figure 11(b)), the grid lines are continuous across the zonal interface In Figure 11(c), the domain is also divided into four zones but the grid lines are discontinuous at the zonal boundary (patched grid lines). Finer grid lines are put in the zones that contain the shock. In this case, some interface is designed as straight line and some as curved line to demonstrate its capability of handling both kind of boundary shapes. The initial variable values at all grid points are set as the free stream values. The shock appears at body surface first and then move outward cross zonal boundary to its steady state position. From Figure 11, it can be seen that the shock can be captured accurately by using all these three grid meshes. For the patched grid case, the contour lines are almost continuous cross the patched zonal interfaces. Because of the conservative nature of this scheme, the patched grid approach can get same good results as the single zone approach. Beside, it can capture shock more accurately by using finer grid in the shock region.

This module is robust for any grid number ratio between the two side of interface. To demonstrate the capability of handling large grid ratio situation, the domain is divided into two zones with grid number ratio of 1:10. Figure 12 shows the grid configuration and the Figure 13 gives the predicted pressure contours.


(a) one zone

(b) four zones, continues grid

(c) four zones, patched grid

Figure 11. Pressure contour for cylinder in supersonic free stream, M=2

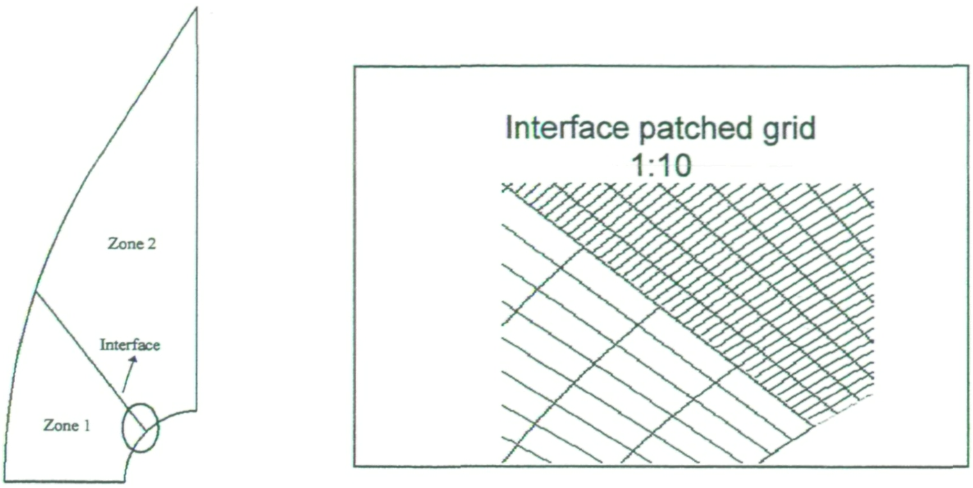
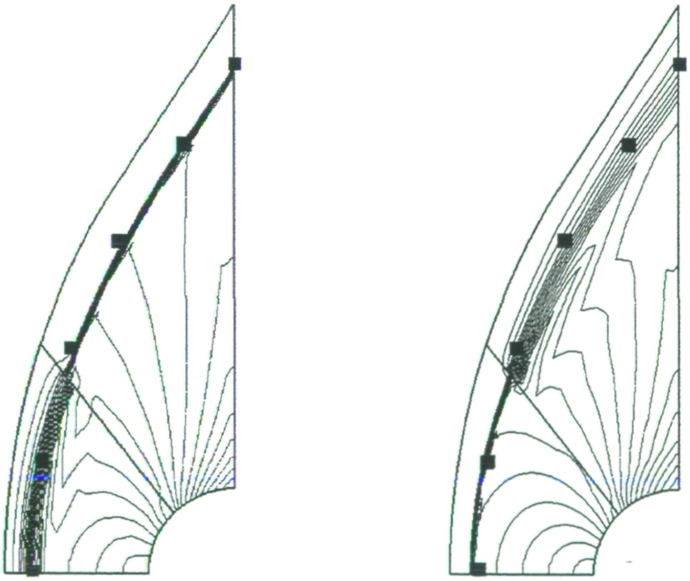


Figure 12. Grid configuration of two-zone domain for cylinder in supersonic flow



(a) Zone1: 26x21
Zone2: 26x201

(b) Zone1: 26x201
Zone2: 26x21

Figure 13. Pressure contour for cylinder in supersonic free stream when using two zones

Backward Facing Step Laminar Flow

Backward facing step flow is a simple but important benchmark test case. The 2-D laminar flow over a backward facing step with 1:2 expansion ration is examined in this section. The reattachment length and the size of secondary recirculation zone change with the inlet flow Reynolds number (based on the inlet bulk velocity and the twice of the inlet channel width). The numerical prediction is compared with the experimental data by Armaly et. al.[2]. The experiment is conducted in a channel with rectangular cross section (aspect ratio is 10). It was reported[3] that the two-dimensionality was hard to maintain after the Reynolds number exceed 500. So, the computational results may depart from the measurements when Reynolds number is bigger than 500.

The computational domain starts from 2 step heights upstream of the expansion corner to 45 step height downstream of the expansion corner. A fully developed laminar velocity profile is imposed at the inlet. The computational domain is divided into three zones. A part of the patched zonal interface is demonstrated in Figure 14. The grid lines are discontinuous at the interface. The flow fields with inlet Reynolds number from 100 to 800 have been calculated. Figure 15 shows the stream line pattern for the case of Reynolds number equal to 300 and 800 respectively. The secondary separation zone start to appear at Reynolds number around 500. The reattachment length and the size of the secondary separation zone are drawn versus Reynolds number and compared with the experimental data in Figure 16. When Reynolds number is less than 400, the prediction of reattachment length matches the experimental data very well. After the Reynolds number exceeds 500, the calculation results are depart from the measurements as expected.

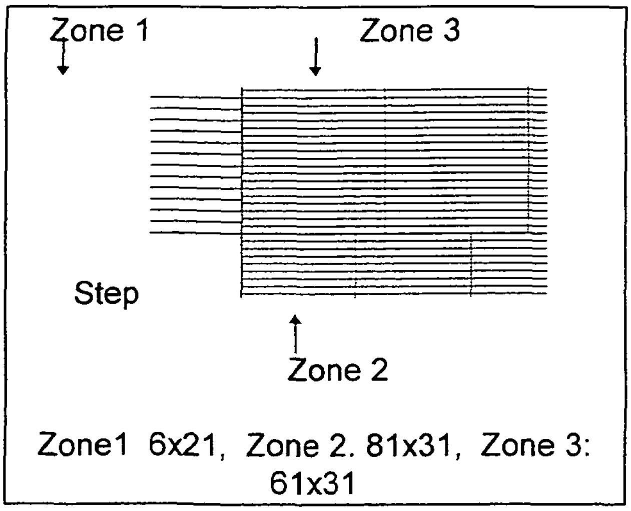
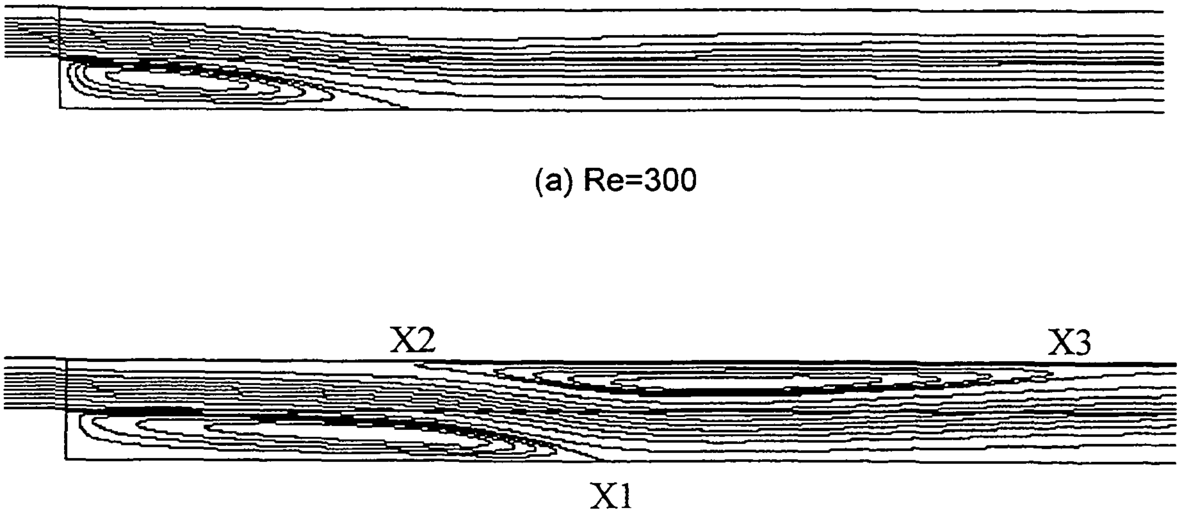


Figure 14. Patched grid at the zonal interface for backward facing step flow



(a) Re=300

(b) Re=800

Figure 15. Backward facing step laminar flow stream line patterns

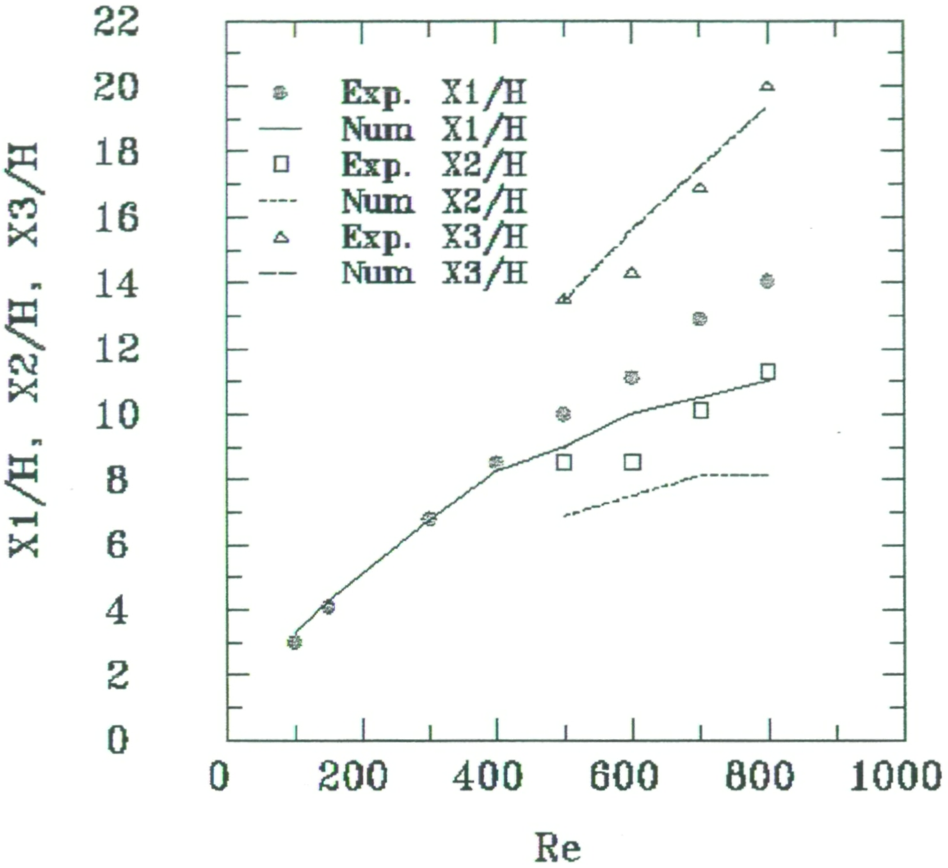


Figure 16. Separation zone size comparison

Laminar Flow over a Cylinder

Coutanceau and Bouard[4, 5] conducted a series of experiments to determine the main features of the viscous flow (steady and unsteady) in the wake of a circular cylinder in uniform translation. A set of flows with different Reynolds number were tested. The transient flow of Reynolds number (based on the free stream velocity and cylinder diameter) equal to 40 is numerically calculated in this study. The uniform free stream is imposed to the initial fields. The region of interest for the cylinder is divided into four zones. The zones and the grids in each zone are shown in Figure 17. Figure 18 shows the flow pattern and wake region at three different time stages. The stream lines continuously cross the zonal interfaces. Figure 19 illustrates the comparison of wake length between the present prediction and the experimental data

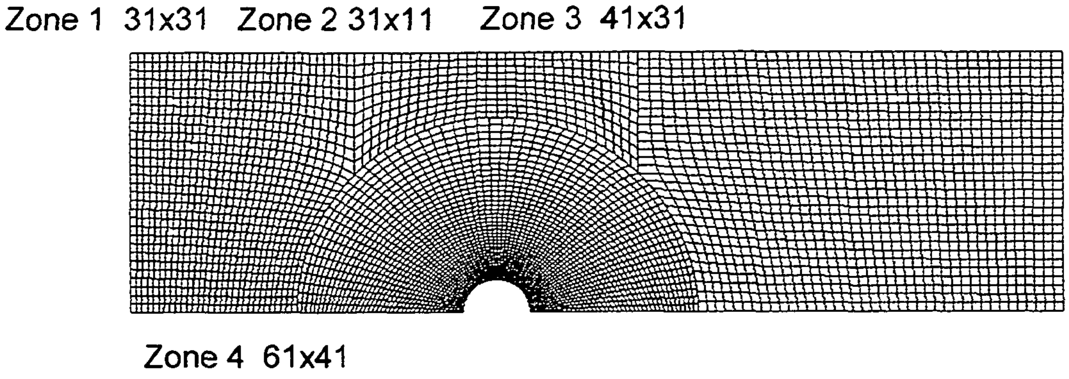
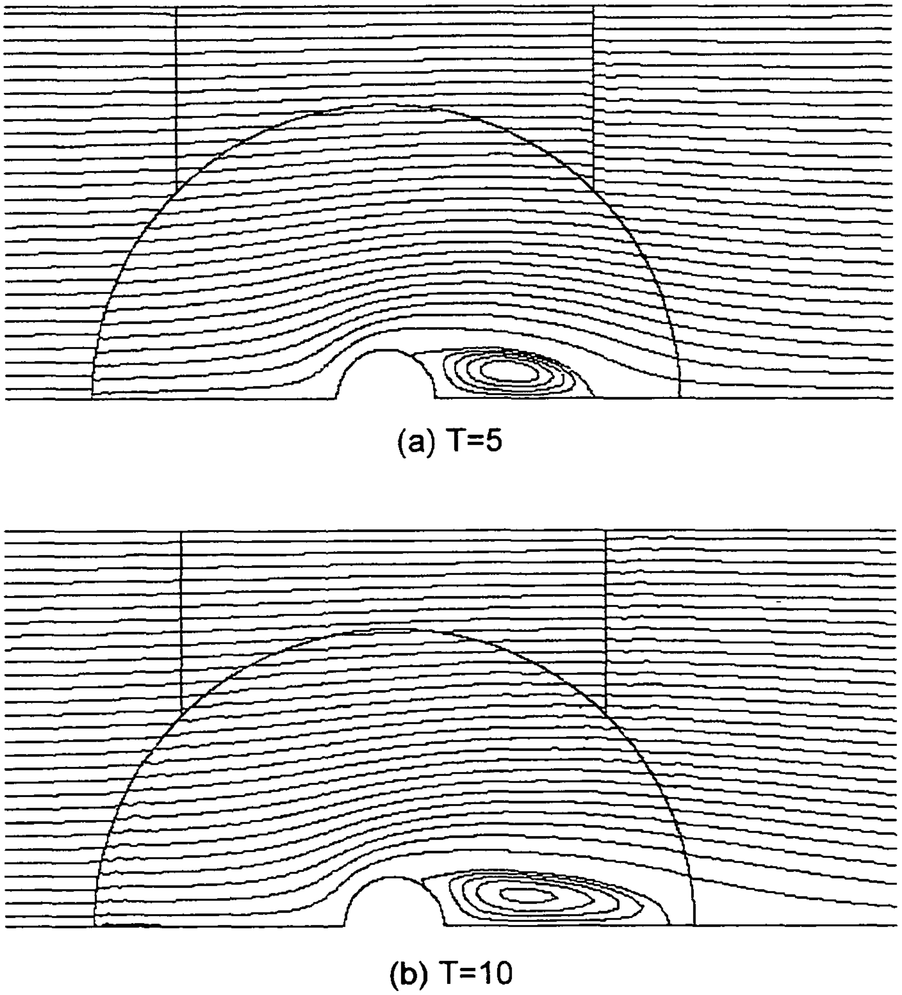
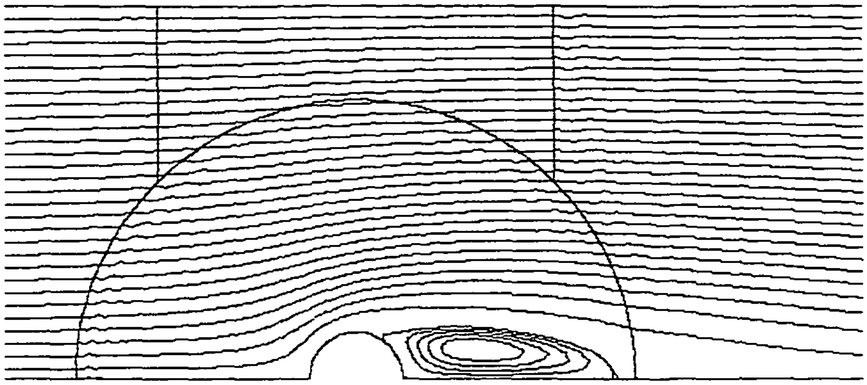


Figure 17. Patched grid for laminar flow past a cylinder


(a) T=5

(b) T=10



(c) T=14

Figure 18. Stream line contours of transient laminar flow past a circular cylinder, Re=40

Where, L is the wake length beginning from the cylinder tail wall, D is the cylinder diameter, T is the nondimensional time, $T=\frac{tU_0}{D}$, and U_0 is the free stream velocity, t is the real time. It can be seen that the calculation results excellently match the experiment measurements

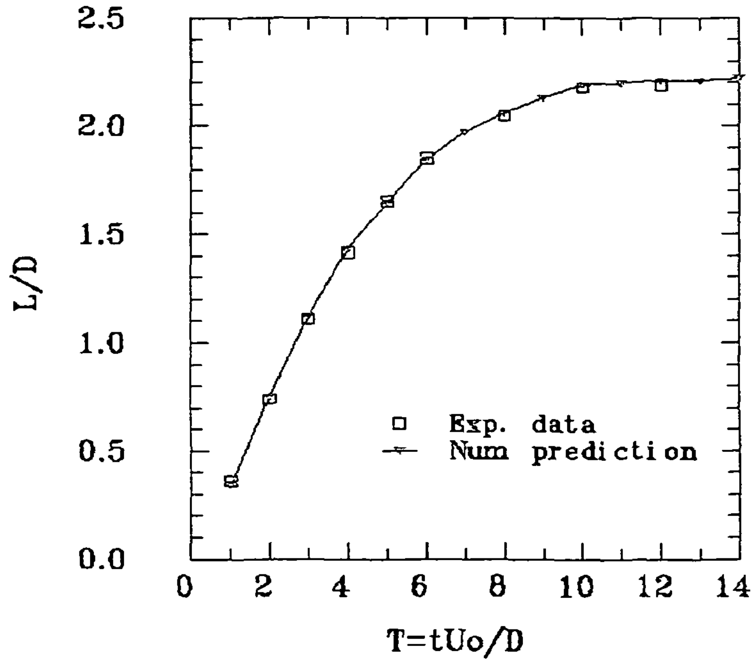
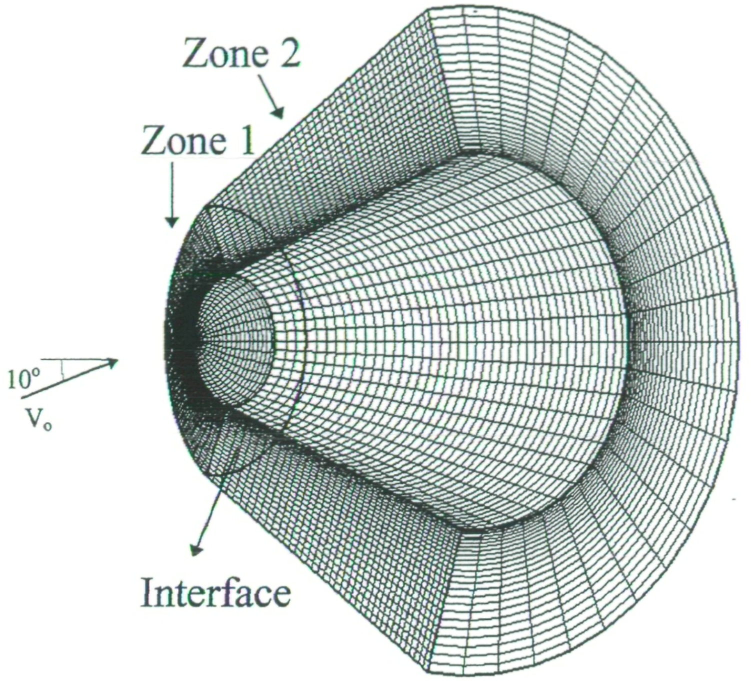


Figure 19 Comparison of time dependent wake growth



Zone 1: 21x31x19
Zone 2: 41x25x25

Figure 20. Grid mesh of 3-D blunt body supersonic flow

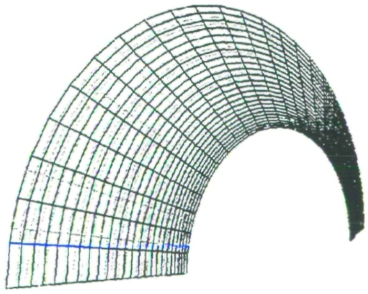
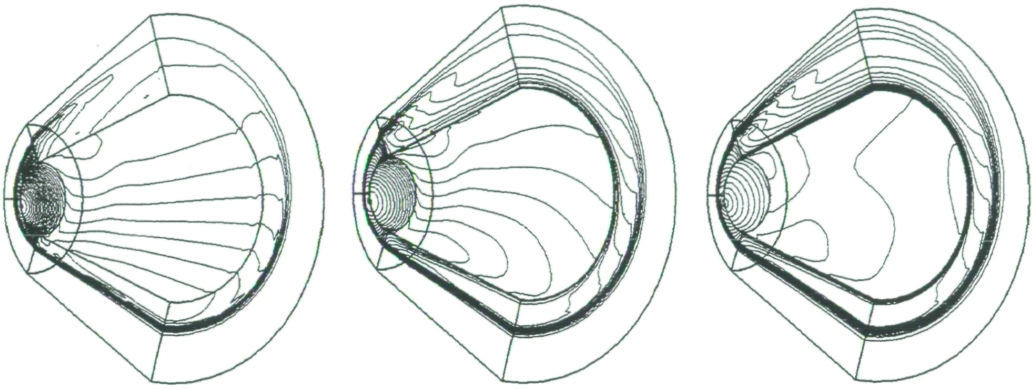


Figure 21. Curved interface for 3-D blunt body

3-D Supersonic Flow Past Blunt Body

A axisymmetric blunt body with spherical nose is put in the supersonic free stream of the Mach number equals to 4. The free stream comes in the direction where the angle of attack equals to 10 degree. So the flow is three dimensional. The calculation domain contains two zones with a curved face as the zonal interface. The zones and their grids are shown in Figure 20. Figure 21 gives the curved zonal interface. The dark grid lines belong to zone1 and the light grid lines belong to zone 2. The isothermal (with 250 K) wall boundary condition is employed to solve the energy equation. The turbulence model used is k-e model.

Figure 22 illustrates the pressure contours, temperature contours, and Mach number contours in the symmetry face, tail outlet face and one grid up the body face. The shock is stronger in the upwind side than that in the downwind side because of the bigger compression. The contour lines are reasonable and smooth across the zonal interface.



(a) Pressure (b) Temperature (c) Mach number

Figure 22. 3-D blunt body in supersonic flow (M=4.0, Angle=10°)

REFERTENCES

1. John David, Anderson: Hypersonic and High Temperature Gas Dynamics, McGraw-Hill Series in Aeronautical and Aerospace Engineering, 1989.
2. B. F. Armaly, F. Dust, J. C. F. Pereira and B. Schonung: “Experimantal and Theoretical Investigation of Backward-Facing Step Flow,” J. Fluid Mech., Vol. 127, pp. 473-496, 1983.
3. Y. S. Chen: Viscous Flow Computations Using A Second-Order Upwind Differencing Scheme,” AIAA 88-0417, 1988.
4. M. Coutanceau and R. Bouard: “Experimental Determination of the main Features Of The Viscous Flow In The Wake Of A Circular Cylinder In Uniform Translation, Part 1. Steady Flow,” J. Fluid Mech., Vol. 79, Part 2, pp. 231-256, 1977.
5. M. Coutanceau and R. Bouard: “Experimental Determination of the main Features Of The Viscous Flow In The Wake Of A Circular Cylinder In Uniform Translation, Part 2. Unsteady Flow,” J. Fluid Mech., Vol. 79, Part 2, pp. 2571-272, 1977.

3. FDNS USER CONTROL AND INITIALIZATION FEATURES

Besides the aforementioned new model developed for the FDNS code, other flow control, problem specification and data preparation new features developed in present effort are described in this section. These new features include: (1) multiple outlet pressure conditions implementation; (2) a new pre-processor for data preparation, conversion and assignment; and (3) new input/output data control specification by the user.

3.1 MULTIPLE OUTLET PRESSURE CONDITIONS

The current version (V.4.5) of FDNS expands the flexibility in specifying the flow outlet pressure conditions. Multiple outlet pressure conditions can be specified with the new input format. The flow outlet pressure level parameter, PRAT, has been changed to an array (instead of a constant in precious versions) in correspondence with the flow exit boundary specification input lines. Each flow exit plane can be specified at its own outlet pressure level and anchored with respect to any point in the flow domain. This new input format for the outlet pressure is described in Section 6.2 in this report.

3.2 FDNS PRE-PROCESSOR FEATURES

The FDNS pre-processor of the current version reads the initial input data file, fort.11 and initial grid and flow files, fort.12, fort.13 and fort.14, prepared by the user (may be by other grid generator and FORTRAN flowfield initialization programs), and prepare file decomposition for a parallel FDNS model based on a user assigned domain decomposition scheme specified. This pre-processor makes all necessary process number assignment in the input data file. Other utility codes are also available to convert data files from previous versions into formats of the current version. Some error checking schemes are also incorporated in the utility programs to provide the user a problem diagnostics tool.

3.3 NEW INPUT/OUTPUT CONTROLS

Since the current parallel CFD model utilizes method of domain decomposition, input/output file decomposition and recombination processes are necessary in running a typical case. Some new input and output control programs are developed for the task Selections of input and output data formats are made through parameters specified in the input data file, fort.11. Other new features such as physical submodel specifications by name searching, freedom in inserting comment cards and symbolic chemical reaction equations input are part of the results of the current research effort. Detailed descriptions of these new features are given in Sections 6.1 and 6 2

4. GRASP RADIATION CODE COMPUTATIONAL EFFICIENCY IMPROVEMENT

4.1 ABSTRACT

The conventional radiative transfer equation (RTE) and the even parity formulation (EPF) of the RTE in body-fitted coordinates have been developed and they are used to simulate multi-dimensional radiative heat transfer in irregular geometries by the discrete ordinates method (DOM). The discrete ordinates equations for the conventional RTE and the EPF are first-order and second-order differential equations, respectively, and they are spatially discretized using different schemes. By investigating five two-dimensional and three-dimensional benchmark problems with absorbing-emitting and scattering media enclosed by irregular walls, the EPF of the DOM is found to be more sensitive to the grid layout and it requires a clustered grid near the wall in order to provide accurate results in radiative wall heat flux. With an appropriate selection of a grid, the even parity solution generally has an accuracy close to the conventional discrete ordinates solution. However, it usually requires more CPU time and iterations to converge. The present study indicates that the conventional RTE of the DOM is a more computationally robust radiation model than the EPF of the DOM.

NOMENCLATURE

F	Sum of opposite direction intensities
ix, iy, iz	volume index numbers
I	radiative intensity
I_b	black body radiative intensity
J	Jacobian matrix
M	total discrete direction numbers over a solid angle of 4π
N_ξ	maximal volume index number along ξ coordinate
$\mathbf{n}$	surface normal vector
$q(\mathbf{r_w})$	irradiation on the wall
q	net radiative wall heat flux

r	location vector
S_b	Additional term in two-dimensional axisymmetric coordinates
x_1, x_2, x_3	Cartesian coordinates
w	weight of a discrete direction

Greek

γ_1, γ_2, γ_3	direction cosines
ε	wall emissivity
Φ	scattering phase function
κ	absorption coefficient
ξ_1, ξ_2, ξ_3	body-fitted coordinates
σ	scattering coefficient
Ω	direction vector

Subscripts

m	discrete direction number
w	at the wall

Superscripts

′	incoming direction
+	leaving direction from the wall
-	arriving direction on the wall

4.2 INTRODUCTION

Radiative heat transfer is often a dominant heat transfer mode in many high-temperature systems involving propulsion, materials processing and manufacturing, etc Design, control, and optimization of these systems require accurate modeling of radiative heat transfer through the hot media In the last three decades, many approximate methods have been developed for solving the radiative transfer equation (RTE) and they include the zone method, Monte Carlo method, flux method, the P-N method, discrete ordinates method (DOM), etc. Each of these methods has its strengths and weaknesses and a review of these methods is given by Howell (1988) and Modest (1993). Due to the strongly coupling between radiative heat transfer and other physical processes in most high-temperature systems, a desirable radiation model should be accurate, computationally efficient, and compatible with fluid dynamics solver. Among the various radiation models, the DOM is one of the few models that have a potential to achieve all these capabilities and meet practical modeling needs.

Like several other radiation models, the DOM was originally applied in the field of nuclear engineering. Currently, two formulations of the DOM have been developed and they are all very successful in solving the neutron transport problems. The first formulation is based on the conventional RTE, which is a first-order differential equation, and the second is based on the even parity formulation (EPF) of the RTE, which is a second-order differential equation. Applications of the DOM in heat transfer problems began in the early of 1980s and most studies have been focused on solving the conventional RTE ever since. From the early of 1990s, the EPF of the DO method began to receive attentions in the heat transfer community due to its unique features. Song and Park (1992) are the first to apply this formulation in radiative heat transfer problems. They transformed the conventional RTE into a second-order differential equation with integral scattering terms and investigated two problems to demonstrate the solution accuracy. Fiveland and Jessee (1994, 1995) formulated the conventional RTE and the EPF with the finite element method and compared the even parity solution with other solutions for different cases. Their results indicated that the accuracy of the even parity predictions degraded as the optical thickness and wall emissivity were increased. Cheong and Song (1995) examined several spatial discretization schemes in the EFP of the DOM. Sample computation for a square enclosure indicated that the control-line approach was the most promising spatial discretization scheme. Liu et al. (1997) investigated the accuracy and efficiency of the EPF by considering a simple two-dimensional enclosure. They found that the EPF of the DOM was not as robust as the conventional DOM.

Because of the completely different mathematical structures, some numerical features in the EPF of the DOM are quite different from those in the conventional DOM and, their examination is necessary for determining the application of the EPF in heat transfer problems. While nearly all realistic problems are of irregular geometries, most current studies on the EPF have been only limited to orthogonal grids in Cartesian coordinates and the application of the EPF in body-fitted coordinates has been rarely discussed. The objective of the present study is to develop the conventional and even parity formulations of the DOM in a general body-fitted coordinate system and evaluate their accuracy and efficiency by considering several benchmark problems with irregular geometries. This work represents another effort to explore the unique numerical features of the EPF.

4.3 MATHEMATICAL FORMULATIONS

Radiative Transfer Equations

Consider the conventional radiative transfer equation (RTE) in a Cartesian coordinate system as shown in Fig. 23. The balance of energy passing in a specified direction Ω through a small differential volume in an absorbing-emitting and scattering gray medium can be written as

$$(\Omega \cdot \nabla) I(\mathbf{r},\Omega) = -(\kappa+\sigma) I(\mathbf{r},\Omega) + \kappa I_b(\mathbf{r}) + \frac{\sigma}{4\pi} \int_{\Omega'=4\pi} I(\mathbf{r},\Omega')\Phi(\Omega' \to \Omega)\, d\Omega' \qquad (1)$$

where $I(r,\Omega)$ is the radiative intensity, which is a function of position and direction; $I_b(r)$ is the blackbody radiative intensity at the temperature of the medium; κ and σ are the absorption and scattering coefficients, respectively; and $\Phi(\Omega' \to \Omega)$ is the scattering phase function from the incoming Ω' direction to the outgoing direction Ω.

If the wall bounding the medium is assumed gray and emits and reflects diffusely, then the radiative boundary condition for Eq. (1) is given by

$$I(\mathbf{r}_w,\Omega^+) = \varepsilon\, I_b(\mathbf{r}_w) + \frac{(1-\varepsilon)}{\pi} q(\mathbf{r}_w) \qquad (2a)$$

with

$$q(\mathbf{r}_w) = \int_{\mathbf{n}\,\Omega^-<0} I(\mathbf{r}_w,\Omega^-)\left|\mathbf{n}\cdot\Omega^-\right| d\Omega^- \qquad (2b)$$

where **n** represents the unit normal vector at the surface; ε is the wall emissivity, Ω^+ and Ω^- denote the leaving and arriving radiative intensity directions, respectively; and $q(r_w)$ is the irradiation on the wall.

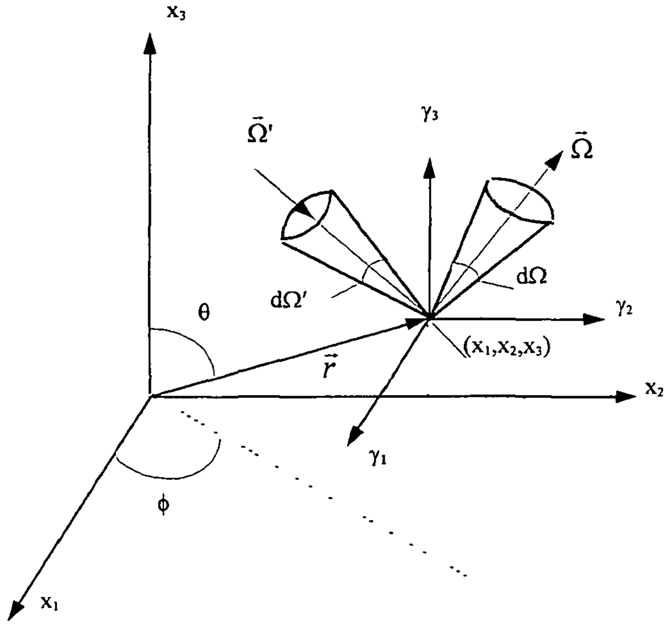


Figure 23. Coordinate system for radiative transfer equation

The even parity formulation (EPF) of the RTE was derived by Liu et al. (1997) and it is expressed as

$$(\Omega\ \nabla)\frac{1}{\kappa+\sigma}(\Omega\cdot\nabla)F(\mathbf{r},\Omega) = (\kappa+\sigma)F(\mathbf{r},\Omega) - 2\kappa I_b(\mathbf{r}) - \frac{\sigma}{4\pi}\int_{\Omega'=2\pi} 2F(\mathbf{r},\Omega')d\Omega' \qquad (3)$$

where F represents the sum of opposite direction intensities. The above equation is a second-order equation that has been shown to be positive definite and self adjoint (Kaplan and Davis, 1967). The corresponding boundary condition is

$$\frac{(\mathbf{n}\cdot\Omega)}{|\mathbf{n}\cdot\Omega|}\frac{1}{\kappa+\sigma}(\Omega\cdot\nabla)F(\mathbf{r}_w,\Omega)=F(\mathbf{r}_w,\Omega)-2\varepsilon I_b(\mathbf{r}_w)-\frac{2(1-\varepsilon)}{\pi}q(\mathbf{r}_w) \quad (4)$$

Coordinate Transformation

The preceding conventional RTE, Eqs. (1) and (2), and even parity RTE, Eqs. (3) and (4), are presented in the vector forms and they can be transformed into the equations in terms of orthogonal Cartesian coordinates x_1, x_2, and x_3 by substituting each vector with its vector components and executing the vector operators. For a problem with a complex geometry, a differential equation in orthogonal Cartesian coordinates must be transformed into the equation in general body-fitted coordinates for the convenience of solution. By referring to the common coordinate transformation procedure as that in the computational fluid dynamics (CFD) (Thompson et al., 1985), the full conservative forms of the conventional RTE and the even parity RTE in a general body-fitted coordinate system (ξ_1, ξ_2, ξ_3) can be easily obtained. For the sake of brevity, the tensor indices are used in the following mathematical formulations in which the repeated subscripts indicate the summation. The tensor index changes from 1, 2 to 3 for three-dimensional problems and from 1 to 2 for two-dimensional problems. For the conventional RTE, Eq. (1), the coordinate transformation in a specific direction Ω with the direction cosines γ_1, γ_2 and γ_3 results in

$$\frac{1}{J}\frac{\partial}{\partial\xi_i}(IJ\gamma_j\frac{\partial\xi_i}{\partial x_j})=-(\kappa+\sigma)I(\mathbf{r},\Omega)+\kappa I_b(\mathbf{r})+\frac{\sigma}{4\pi}\int_{\Omega'=4\pi}I(\mathbf{r},\Omega')\Phi(\Omega'\rightarrow\Omega)d\Omega' \quad (5)$$

where J is the Jacobian of the coordinate transformation and is given as

$$J=\frac{\partial(x_1,x_2,x_3)}{\partial(\xi_1,\xi_2,\xi_3)} \quad (6)$$

For two-dimensional planar or axisymmetric problems, the Jacobian is replaced by

$$J=x_2^{IAX-1}x_3^{2-IAX}\frac{\partial(x_1,x_2)}{\partial(\xi_1,\xi_2)} \quad (7)$$

where IAX =1 is used for two-dimensional planar problems and IAX=2 is specified for two-dimensional axisymmetric problems. For two-dimensional axisymmetric problems, the radial direction is assumed to be along the positive x_2-axis. The x_3-coordinate here simulates the effects of three-dimensional volume changes with a default value of unity. To account for the effect of the angular derivative in two-dimensional axisymmetric problems, an additional term must be added to the left hand side of Eq (5) and it is expressed as

$$S_b = -\frac{1}{x_2}\frac{\partial(\gamma_3 I)}{\partial\psi} \qquad (8)$$

where γ_2 is the radial direction cosine; γ_3 is the cosine in azimuthal angle direction for x_1-x_2 equation of transfer and ψ is the azimuthal angle for the direction Ω.

For the EPF of the RTE, Eq (3), the coordinate transformation in a specific direction Ω results in

$$\frac{1}{J}\frac{\partial}{\partial\xi_i}(\frac{J}{\kappa+\sigma}\gamma_k\gamma_l\frac{\partial\xi_i}{\partial x_k}\frac{\partial\xi_j}{\partial x_l}\frac{\partial F}{\partial x_j}) = (\kappa+\sigma)F(\mathbf{r},\Omega) - 2\kappa I_b(\mathbf{r}) - \frac{\sigma}{4\pi}\int_{\Omega=2\pi} 2F(\mathbf{r},\Omega')d\Omega' \, (9)$$

The following additional term is also added on the left hand side of Eq. (9) to account for the effect of the angular derivative in two-dimensional axisymmetric problems

$$\begin{aligned} S = &\frac{1}{J}\frac{\partial}{\partial\xi_i}\{\frac{J}{\kappa+\sigma}\gamma_k\frac{\partial\xi_i}{\partial x_k}\frac{1}{x_2}[\gamma_2 F - \frac{\partial(\gamma_3 F)}{\partial\psi}]\} \\ &+\frac{1}{x_2}\frac{\partial}{\partial\psi}\{\frac{\gamma_3}{\kappa+\sigma}[\gamma_k\frac{\partial\xi_i}{\partial x_k}\frac{\partial F}{\partial\xi_i} - \frac{1}{x_2}(\gamma_2 F - \frac{\partial(\gamma_3 F)}{\partial\psi})]\} \qquad (10) \end{aligned}$$

In general body-fitted coordinates, the EPF at boundary, Eq. (4), becomes

$$\frac{(n\ \Omega)}{|n\cdot\Omega|}\frac{1}{\kappa+\sigma}\frac{1}{J}\frac{\partial}{\partial\xi_i}(J\gamma_k\frac{\partial\xi_i}{\partial x_k}F) = F(r_w,\Omega) - 2\varepsilon\, I_b(r_w) - \frac{2(1-\varepsilon)}{\pi}q(r_w) \, (11)$$

For two-dimensional axisymmetric problems, the additional term needed on the right hand side of Eq. (11) to account for the effect of the angular derivative is

$$S_b = -\frac{1}{x_2}\frac{\partial(\gamma_3 F)}{\partial\psi} \qquad (12)$$

4.4 DISCRETE ORDINATES METHOD

Discrete Ordinates Equations

The conventional RTE and even parity RTE in a body-fitted coordinate system involve not only spatial differentiation but also the angular integration over the solid angle Ω. To solve these equations numerically, the angular dependence is first removed. In the DOM, the RTE is replaced by a set of equations for a finite number of ordinate directions For a specific ordinate direction m, defined by $\Omega^m = (\gamma_1^m, \gamma_2^m, \gamma_3^m)$, the integrals in Eqs. (5) and (9) are replaced by quadratures of order M and M/2 respectively with the appropriate angular weights w^m

$$\frac{1}{J}\frac{\partial}{\partial \xi_i}(I^m J\gamma_j \frac{\partial \xi_i}{\partial x_j}) = -(\kappa + \sigma)I^m + \kappa\ I_b + \frac{\sigma}{4\pi}\sum_{m'=1}^{M} w^{m'}\Phi^{mm'} I^m \qquad (13)$$

$$\frac{1}{J}\frac{\partial}{\partial \xi_i}(\frac{J}{\kappa+\sigma}\gamma_k\gamma_l \frac{\partial \xi_i}{\partial x_k}\frac{\partial \xi_j}{\partial x_l}\frac{\partial F^m}{\partial x_j}) = (\kappa + \sigma)F^m - 2\kappa\ I_b - \frac{\sigma}{4\pi}\sum_{m'=1}^{M/2} 2w^{m'}F^{m'} \quad (14)$$

It is noted that the integration in the even parity RTE, Eq. (14), is made over 2π solid angles, thus it is necessary to solve the equations for only half of the directions that would be required if the conventional RTE, Eq. (13), were applied. The selection of the direction cosines and corresponding weights, that is the quadrature scheme, is arbitrary, although restrictions arise from the need to preserve symmetries and invariance properties of the physical system. In this study, the level symmetric S$_n$ quadrature scheme (Fiveland, 1988) was employed.

For the discrete direction Ω_m, the radiative boundary conditions for Eqs. (13) and (14) can be written as

$$I_w^m = \varepsilon\, I_b + \frac{(1-\varepsilon)}{\pi}q \qquad (15)$$

$$\frac{(n\ \Omega^m)}{\left|n\ \Omega^m\right|}\frac{1}{\kappa+\sigma}\frac{1}{J}\frac{\partial}{\partial \xi_i}(J\gamma_k \frac{\partial \xi_i}{\partial x_k}F^m) = F^m - 2\varepsilon\, I_b - \frac{2(1-\varepsilon)}{\pi}q \quad (16)$$

The above equations also involve the angular integration which is implicitly included in the calculation of the irradiation q

Spatial Discretization

To solve the discrete ordinates equations, the spatial domain is divided into small control volumes. Within each control volume, the spatially discretized equations along the discrete direction Ω_m are derived. In the following analysis, ix, iy and iz are used to represent the volume nodal index numbers along the ξ_1, ξ_2 and ξ_3 coordinates, respectively.

The discrete ordinates equation for the conventional RTE, Eq. (13), is a first-order differential equation. For a representative control volume, the first-order spatial derivative term along a specific coordinate ξ_1 is approximated by the values on the volumes surfaces, that is

$$\left.\frac{\partial}{\partial \xi_1}(I^m J\gamma_j \frac{\partial \xi_1}{\partial x_j})\right|_{ix} = (I^m J\gamma_j \frac{\partial \xi_1}{\partial x_j})_{ix+1/2} - (I^m J\gamma_j \frac{\partial \xi_1}{\partial x_j})_{ix-1/2} \tag{17}$$

To close the above equation, relations are needed between the intensities on the volume surfaces and the nodal intensities. One appropriate closure relation for complicated geometries is based on the step scheme (Chai et al., 1995) which sets the downstream surface intensities equal to the upstream nodal intensities. Then, the final discretized equation for the DOM can be readily obtained as

$$\left.\frac{\partial}{\partial \xi_1}(I^m J\gamma_j \frac{\partial \xi_1}{\partial x_j})\right|_{ix} = -\max[(J\gamma_j \frac{\partial \xi_1}{\partial x_j})_{ix-1/2}, 0] I^m_{ix-1}$$

$$-\{\min[(J\gamma_j \frac{\partial \xi_1}{\partial x_j})_{ix-1/2}, 0] - \max[(J\gamma_j \frac{\partial \xi_1}{\partial x_j})_{ix+1/2}, 0]\} I^m_{ix} + \min[(J\gamma_j \frac{\partial \xi_1}{\partial x_j})_{ix+1/2}, 0] I^m_{ix+1} \tag{18}$$

The spatial discretization for the derivative terms along the other coordinates can be conducted in a similar fashion. For two-dimensional axisymmetric problems, the azimuthal angular difference in Eq. (8) is treated by introducing the angular differencing coefficients and detailed procedure is provided by Carlson and Lathrop (1968).

The discrete ordinates equation for the EPF of the RTE, Eq. (14), is a second-order differential equation which is similar to the transport equations found in fluid dynamics, thus it can be discretized spatially in a way similar to that used in CFD (Hirsch, 1990). In a control volume, the second-order spatial derivative terms in Eq. (14) are discretized using the second-order central difference scheme while the terms on the right hand side of Eq. (14) are treated as source terms. The discretized terms are divided into non-mixed derivative and mixed derivative parts. The non-mixed derivative terms are treated implicitly and the mixed derivative terms are lumped into the source terms. This arrangement reduces the computer memory requirements compared to the fully implicit treatment. For simplicity, only the spatial derivative term along the coordinate ξ_1 is presented and it is

$$\begin{aligned}\frac{\partial}{\partial\xi_1}\left(\frac{J}{\kappa+\sigma}\gamma_k\gamma_l\frac{\partial\xi_1}{\partial x_k}\frac{\partial\xi_j}{\partial x_l}\frac{\partial F^m}{\partial\xi_j}\right)\Bigg|_{ix} &= \left(\frac{J}{\kappa+\sigma}\gamma_k\gamma_l\frac{\partial\xi_1}{\partial x_k}\frac{\partial\xi_1}{\partial x_l}\frac{\partial F^m}{\partial\xi_1}\right)_{ix+1/2}-\left(\frac{J}{\kappa+\sigma}\gamma_k\gamma_l\frac{\partial\xi_1}{\partial x_k}\frac{\partial\xi_1}{\partial x_l}\frac{\partial F^m}{\partial\xi_1}\right)_{ix-1/2}\\ &+\left(\frac{J}{\kappa+\sigma}\gamma_k\gamma_l\frac{\partial\xi_1}{\partial x_k}\frac{\partial\xi_2}{\partial x_l}\frac{\partial F^m}{\partial\xi_2}\right)_{ix+1/2}-\left(\frac{J}{\kappa+\sigma}\gamma_k\gamma_l\frac{\partial\xi_1}{\partial x_k}\frac{\partial\xi_2}{\partial x_l}\frac{\partial F^m}{\partial\xi_2}\right)_{ix-1/2}\\ &+\left(\frac{J}{\kappa+\sigma}\gamma_k\gamma_l\frac{\partial\xi_1}{\partial x_k}\frac{\partial\xi_3}{\partial x_l}\frac{\partial F^m}{\partial\xi_3}\right)_{ix+1/2}-\left(\frac{J}{\kappa+\sigma}\gamma_k\gamma_l\frac{\partial\xi_1}{\partial x_k}\frac{\partial\xi_3}{\partial x_l}\frac{\partial F^m}{\partial\xi_3}\right)_{ix-1/2}\end{aligned}\qquad(19)$$

where the terms associated with $\partial F^m/\partial\xi_1$ are the orthogonal terms and the terms involving $\partial F^m/\partial\xi_2$ and $\partial F^m/\partial\xi_3$ represent the non-orthogonal terms. The orthogonal and non-orthogonal terms on the ix+1/2 control volume interface can be written as

$$\left(\frac{J}{\kappa+\sigma}\gamma_k\gamma_l\frac{\partial\xi_1}{\partial x_k}\frac{\partial\xi_1}{\partial x_l}\frac{\partial F^m}{\partial\xi_1}\right)_{ix+1/2}=\left(\frac{J}{\kappa+\sigma}\gamma_k\gamma_l\frac{\partial\xi_1}{\partial x_k}\frac{\partial\xi_1}{\partial x_l}\right)_{ix+1/2}\times\left(F^m_{ix+1,iy,iz}-F^m_{ix,iy,iz}\right)\quad(20)$$

$$\begin{aligned}&\left(\frac{J}{\kappa+\sigma}\gamma_k\gamma_l\frac{\partial\xi_1}{\partial x_k}\frac{\partial\xi_2}{\partial x_l}\frac{\partial F^m}{\partial\xi_2}\right)_{ix+1/2}=\frac{1}{4}\left(\frac{J}{\kappa+\sigma}\gamma_k\gamma_l\frac{\partial\xi_1}{\partial x_k}\frac{\partial\xi_2}{\partial x_l}\right)_{ix+1/2}\\ &\times\left(F^m_{ix+1,iy+1,iz}+F^m_{ix,iy+1,iz}-F^m_{ix+1,iy-1,iz}-F^m_{ix,iy-1,iz}\right)\end{aligned}\qquad(21)$$

$$\begin{aligned}&\left(\frac{J}{\kappa+\sigma}\gamma_k\gamma_l\frac{\partial\xi_1}{\partial x_k}\frac{\partial\xi_3}{\partial x_l}\frac{\partial F^m}{\partial\xi_3}\right)_{ix+1/2}=\frac{1}{4}\left(\frac{J}{\kappa+\sigma}\gamma_k\gamma_l\frac{\partial\xi_1}{\partial x_k}\frac{\partial\xi_3}{\partial x_l}\right)_{ix+1/2}\\ &\times\left(F^m_{ix+1,iy,iz+1}+F^m_{ix,iy,iz+1}-F^m_{ix+1,iy,iz-1}-F^m_{ix,iy,iz-1}\right)\end{aligned}\qquad(22)$$

For two-dimensional problems, the terms involving $\partial F^m/\partial\xi_3$ vanish. For two-dimensional axisymmetric problems, the additional term arising from the angular derivative, Eq. (10), is treated as a part of the source terms. The numerical treatment of the azimuthal angular difference follows the original procedure described by Carlson and Lathrop (1968) by introducing the angular differencing coefficients.

Discretization of the EPF of the radiative boundary condition, Eq. (16), requires special attention because it is a first-order differential equation and is likely to induce instability in the numerical procedure (Hirsch, 1990; Koch et al., 1995). To overcome this problem, an upwind scheme is used. The order of this upwind scheme is selected to be second or third-order in order to be compatible with the scheme used for the interior control volume. The detailed discretization formulation can be derived by following the same procedure described by Liu et al. (1997).

4.5 SOLUTION METHOD

The preceding spatial discretization is carried out in one discrete direction. The same discretization procedure is applied to all discrete directions. This forms a finite number of systems of non-symmetric algebraic equations. In this study, each direction is solved independently. Before the solution procedure begins, the temperature and partial pressure of the radiatively participating medium in each control volume are provided, and the absorption and scattering coefficients are calculated. Due to the dependence of the source terms and the boundary conditions on the variable F as well as the intensity, global iterations are necessary for solving the conventional RTE and even parity RTE. As the first step of solution procedure, an initial solution is assumed. Then the source terms and irradiation are calculated and a system of equations for each direction is solved. The new solution replaces the previous solution and the iterative procedure continues until convergence is obtained. The solver of the discretized equations is directly taken from a general CFD code, the FDNS code (Chen, 1989; Wang and Chen, 1993). This matrix solver is based on the conjugate gradients squared method. An incomplete factorization serves as preconditioner to accelerate convergence rate.

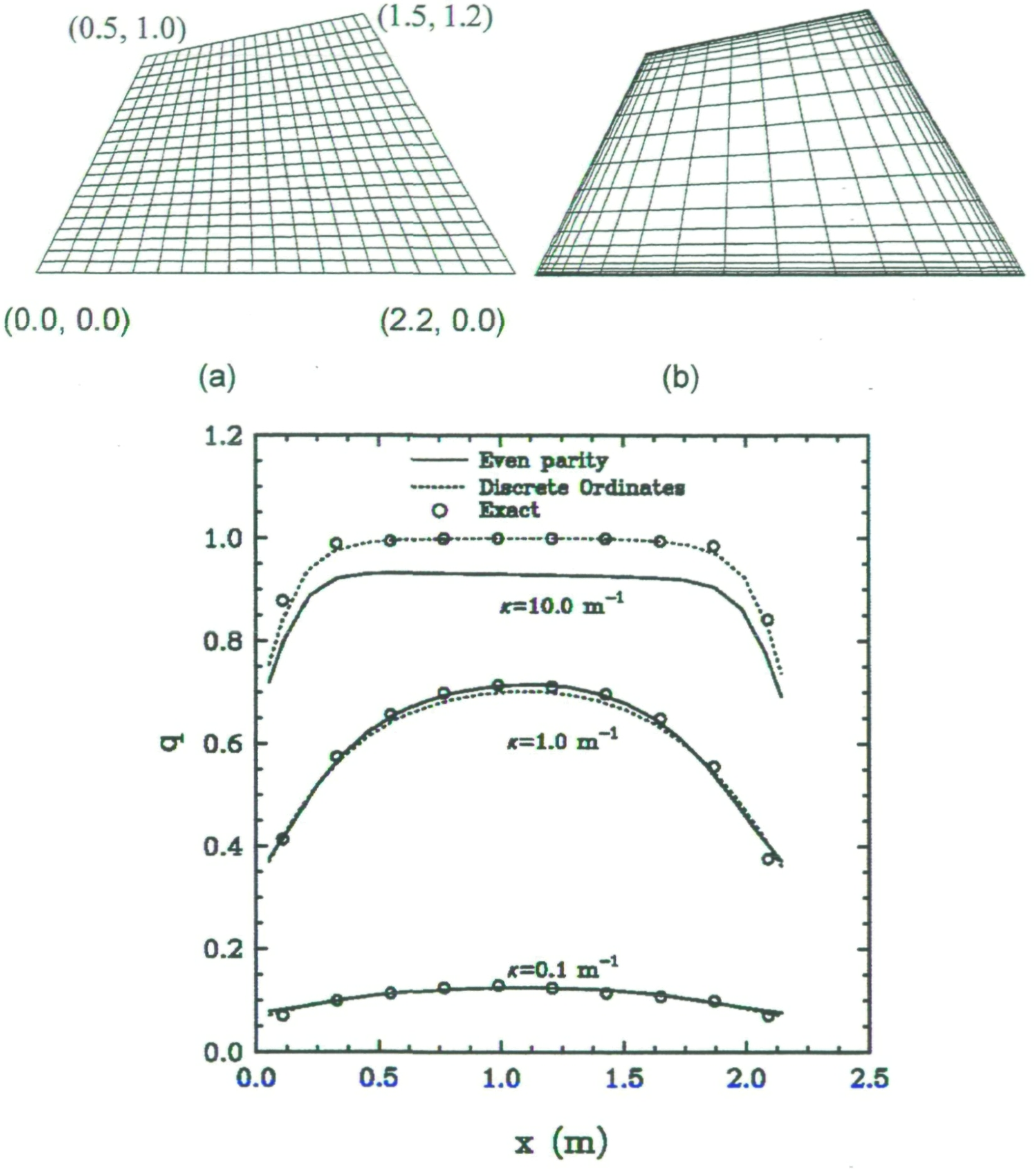
4.6 RESULTS AND DISCUSSION

Based on the above theoretical and numerical analyses, two computer programs have been developed which are capable of simulating two-dimensional planar, two-dimensional axisymmetric or three-dimensional radiative heat transfer containing gray or nongray absorbing-emitting and scattering media in a general body-fitted coordinate system. In the first program, the conventional RTE is used with the DOM and the corresponding solution is termed the discrete ordinates solution. In the second program, the EPF of the RTE is used with the DOM and the corresponding solution is termed the even parity solution. To evaluate the accuracy and computational efficiency of the conventional RTE and the EPF of the RTE in general body-fitted coordinates, five benchmark problems were selected and the discrete ordinates and even parity solutions were compared with other available solutions. For the three two-dimensional problems, the S_8 quadrature scheme was used to define the direction cosines and corresponding weights. For the two three-dimensional problems, the S_4 quadrature scheme was used. In each problem the same spatial grid was applied to both the discrete ordinates and even parity solutions. All computation was conducted on the IBM RISC/6000 machine and the numerical solution was considered to be convergent when the relative incident radiation change was less than 0.01%.

Two-dimensional Quadrilateral Enclosure

The first problem examined is an irregular quadrilateral enclosure as shown in Fig. 24a (all dimensions are in meter). This problem consists of an absorbing-emitting medium maintained at an emissive power of unity and cold and black walls. The selected medium κ varies from 0.1 to 1.0 to 10.0 m^{-1}. This benchmark problem has been studied before and an exact solution (Chai et al., 1995) for radiative heat flux along the bottom wall is available to validate the present solutions.

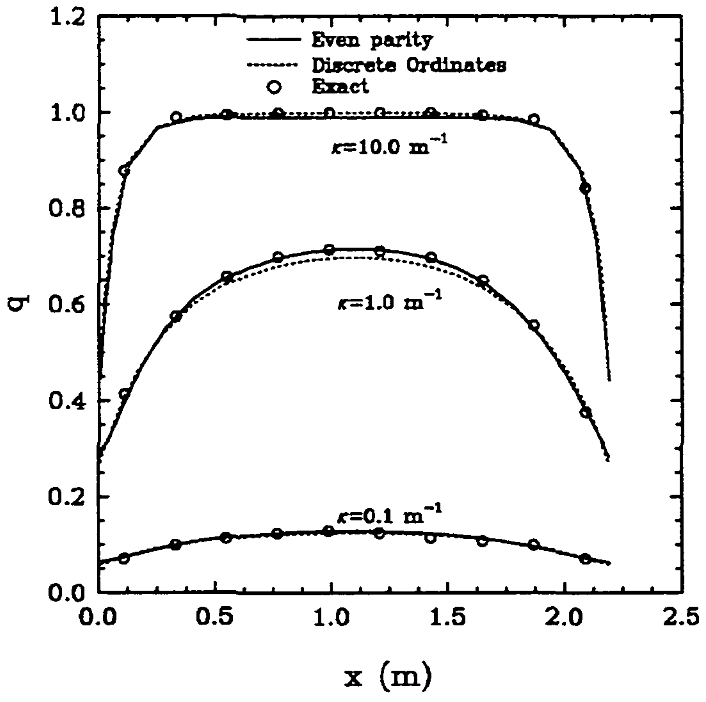
To examine the sensitivity of the discrete ordinate solution and the even parity solution to the grid layout, the non-clustered grid (Fig. 24a) and clustered grid near the walls (Fig. 24b) are considered. Both the grids are body-fitted with $N_{\xi_1} \times N_{\xi_2}$=21×21. Here, N_{ξ_1}, N_{ξ_2} and N_{ξ_3} represent the total grid numbers along the ξ_1, ξ_2 and ξ_3 coordinates, respectively. Figure 24c demonstrates the radiative heat flux distributions along the bottom wall for the non-clustered grid at different absorption coefficients. At κ of 0.1 and 1.0, both the discrete ordinates and even parity solutions are in good agreement with the exact solution. As κ increases to 10.0, the discrete ordinate solution continues to reproduce the exact results very well while the even parity solution becomes about 10% below the exact solution. The reason for this difference is that the variable F in the EFP usually changes rapidly near the wall, especially for optically thick media, and its prediction based on a non-clustered can become inaccurate. Since the radiative wall flux is a function of the gradient of the variable F, an inaccurate prediction of the variable F can only result in an inaccurate prediction of the radiative wall flux. Figure 24d shows the results for the clustered grid. Unlike the non-clustered grid, both the discrete ordinates and



(a)

(b)

(c)



(d)

Figure 24. Two-dimensional quadrilateral enclosure: (a) non-clustered grid, (b) clustered grid, (c) radiative heat flux distributions on the bottom wall for the non-clustered grid; (d) radiative heat flux distributions on the bottom wall for the clustered grid.

even parity solutions are found to match the exact solution at each of three absorption coefficients. The present results demonstrate that the EPF is more sensitive to the grid layout than the conventional RTE and it requires a grid clustered near the wall in order to obtain accurate results in radiative wall heat flux. A similar study was performed for the remaining problems and they all reflect the same trend. The present finding is consistent with the previous studies (Fiveland and Jessee, 1994, Fiveland and Jessee, 1995; Liu et al., 1997) on the regular two-dimensional geometry In the rest of this study, only clustered grids are used for the convenience of comparison between the discrete ordinates and even parity solutions.

Two-dimensional Curved Geometry No. 1

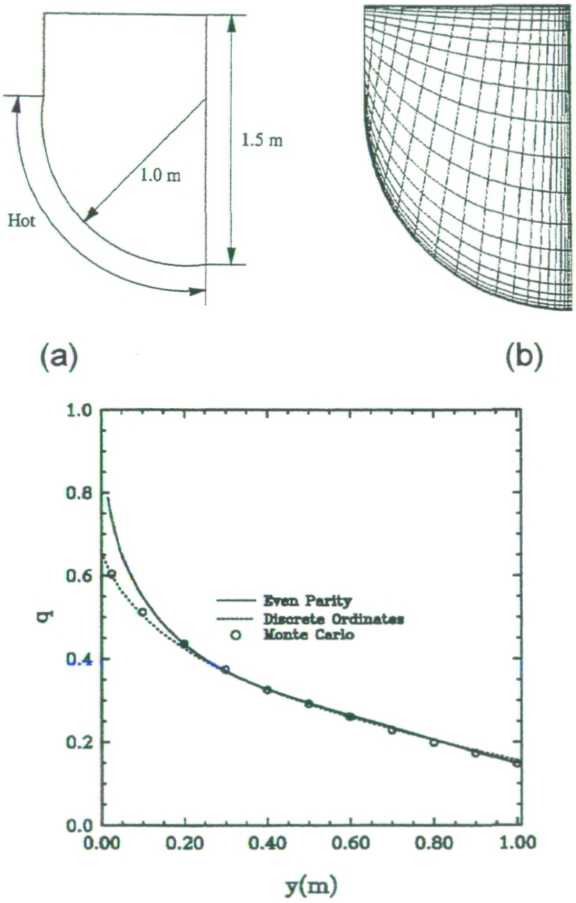
The second problem considered is a quarter of a circle with a rectangular region added to the top as shown in Fig. 25a. The curve wall is hot and black and it has an emissive power of unity, while the other walls are cold and black. The medium is cold and purely absorbing with κ=1.0 m^{-1}. This problem is different from the first problem in that there exists a discontinuous emissive power near the joint points between the curved wall and the right and left walls. The Monte Carlo method (Parthasarathy et al., 1994) has been used to investigate this problem before and its solution was used to validate the present solutions

For a body-fitted grid ($N_{\xi_1} \times N_{\xi_2}$=21×21) as shown in Fig. 25b, the radiative heat flux distribution on the right wall from the discrete ordinates and the even parity solutions are presented in Fig. 25c. The discrete ordinates solution is seen to essentially coincide with the Monte Carlo solution along the entire right wall However, the even parity solution is found to gradually deviate from the Monte Carlo solution as the location on the right wall is approaching the curved wall. The similar phenomenon has been reported by Liu et al. (1997) who investigated a square enclosure with discontinuous emissive power at the corners. Liu et al. (1997) attributed this deviation to the physically unrealistic negative intensities around the corner regions and they used a negative intensity fix-up procedure to improve the accuracy of the results The same fix-up procedure was tried on the present problem and the even parity solution was found never convergent. The reason for the divergence can be found by monitoring the change of the F values in an iterative solution procedure. In a certain solution iteration and along a specific discrete direction, the discretized even parity equations, whose unknown values include those not only on the interior nodes but also on the boundary points, must be solved simultaneously This results in a set of F values. If the intensities, which are the derivative quantity of the variable F, become negative in a corner region, a new set of F values are then obtained by setting the negative intensities equal to zero. If the changes of the F values obtained from solving the discretized equation and setting negative intensities equal to zero keep the same trend in the rest iterations, the final solution would be gradually convergent. Otherwise, the even parity solution becomes divergent as seen in this problem. Therefore, it is concluded that the negative intensity fix-up procedure may be not always suitable for the EPF.

Two-dimensional Curved Geometry No. 2

The schematic and body-fitted grid of the third problem are showed in Fig. 26a. The top wall is located at y=1.0 m. The bottom wall varies according to the following function

$$y = \frac{1}{2}[\tanh(2-3x) - \tanh(2)] \qquad 0 \le x \le \frac{10}{3} \qquad (23)$$



(a)

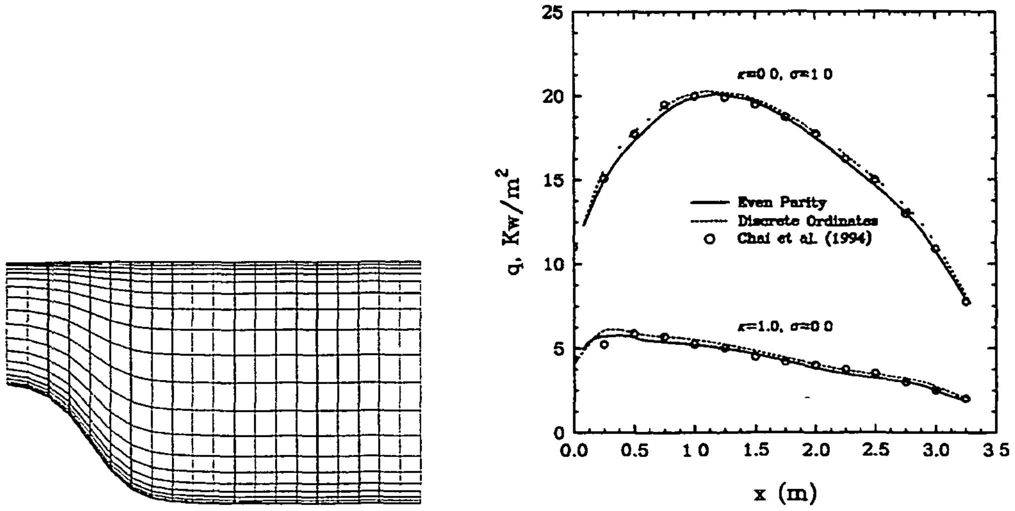
(b)

(c)

Figure 25. Two-dimensional curved geometry No. 1: (a) schematic; (b) grid; (c) radiative heat flux distributions on the right wall.

The bottom black wall is maintained at 1000 K while the other black walls are kept at 0 K. The medium is cold (0 K) and it is purely absorbing with κ=1 0 m^{-1} for the first case and purely isotropically scattering with σ=1.0 m^{-1} for the second case. This problem was considered by Chai et al (1994) using the FVM with the blocked-off region treatment Use of the blocked-off region treatment means that a fine grid must be employed in the region with a curve wall in order to assure the accuracy of the numerical results. Such a restriction can be, however, relieved if a body-fitted grid is employed as in this problem.

Figure 26b shows the radiative heat flux distributions on the top wall with a grid of $N_{\xi} \times N_{\zeta}$=21×21. For each case, both the discrete ordinates and even parity solutions are seen to match the solution from Chai et al. (1994) very well Because the presented results are along the top wall and there is no discontinuous emissive power on the two upper corner regions, the discrepancy between the even parity solution and other solutions observed in the previous problem is not found in this problem



(a)

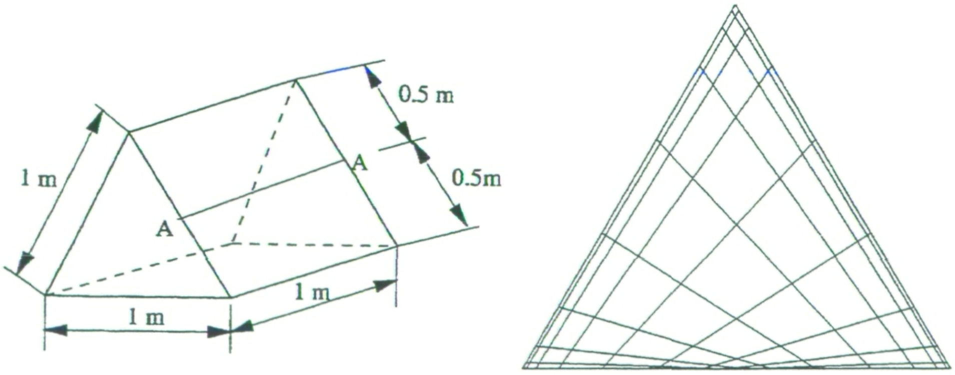
(b)

Figure 26 Two-dimensional curved geometry No. 2· (a) schematic; (b) radiative heat flux distributions on the top wall.

Three-dimensional Equilateral Triangular Enclosure

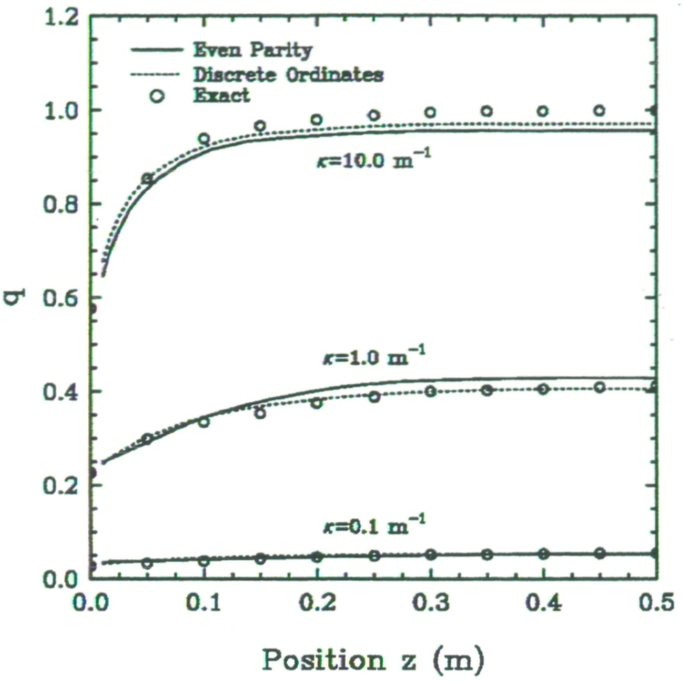
The fourth problem examined is a three-dimensional equilateral triangular enclosure as shown in Fig. 27a. In this problem, all walls are black and cold. The medium is purely absorbing-emitting and maintained at an emissive power of unity. The selected medium absorption coefficient κ varies from 0.1, 1.0 to 10. m^{-1}. Chai et al. (1996) investigated this problem and they obtained the exact solution by integrating the RTE over the spatial and angular domains.

Since the grid in the ξ_3 direction is perpendicular to the ξ_1-ξ_2 plane, Fig. 27b only shows the body-fitted grid at one ξ_1-ξ_2 cross-section for the sake of brevity. A grid of $N_{\xi_1} \times N_{\xi_2} \times N_{\xi_3}$=10×10×11 was used in the computation. Figure 27c shows the radiation heat flux distributions along the A-A line (see Fig. 27a) for three different values of κ. Due to symmetry, heat fluxes are only plotted for half of the enclosure. Compared to the exact solution, both the discrete ordinates and even parity solutions are seen very satisfactory at κ of 0.1 and 1.0. At κ equal to 10.0, the two present solutions are slightly below the exact solution. These discrepancies were found to disappear if the grid number and the order of quadrature scheme were slightly increased.



(a)

(b)



(c)

Figure 27. Three-dimensional equilateral triangular enclosure: (a) schematic; (b) grid; (c) radiative heat flux distributions along the A-A line.

Three-dimensional Elliptical Enclosure

The last problem considered is a three-dimensional elliptical enclosure as shown in Fig. 28a. The radius R, width W, and length H are 1.0 m, 1.5 m, and 2.0 m, respectively. All walls are black and cold. The medium is maintained at an emissive power of unity. Among the three cases considered, the first case has an absorbing-emitting and isotropically scattering medium with κ= 0.5 m^{-1} and σ= 0.5 m^{-1} and other two cases have a purely absorbing-emitting medium with κ= 1.0 and 10.0 m^{-1}, respectively. Chai et al. (1996) investigated this problem using the finite volume method and their solution was used to validate the present solutions.

Since the grid in the ξ_3 direction is perpendicular to the ξ_1-ξ_2 plane, Fig. 28b only shows the body-fitted grid at one ξ_1-ξ_2 cross-section for the sake of brevity. A grid of $N_{\xi_1} \times N_{\xi_2} \times N_{\xi_3}$=20×20×21 was used in the computation. Only half of the enclosure was studied for this problem due to symmetry. Figure 28c shows the radiation heat flux distributions along the A-A line (see Fig. 28a) for three different cases. At κ equal to 10.0, the discrete ordinates solution is in good agreement with the finite volume solution, while the even parity solution is slightly below the finite volume solution. For the other two cases, both discrete ordinates and even parity solutions are seen to be very close to the finite volume solution.

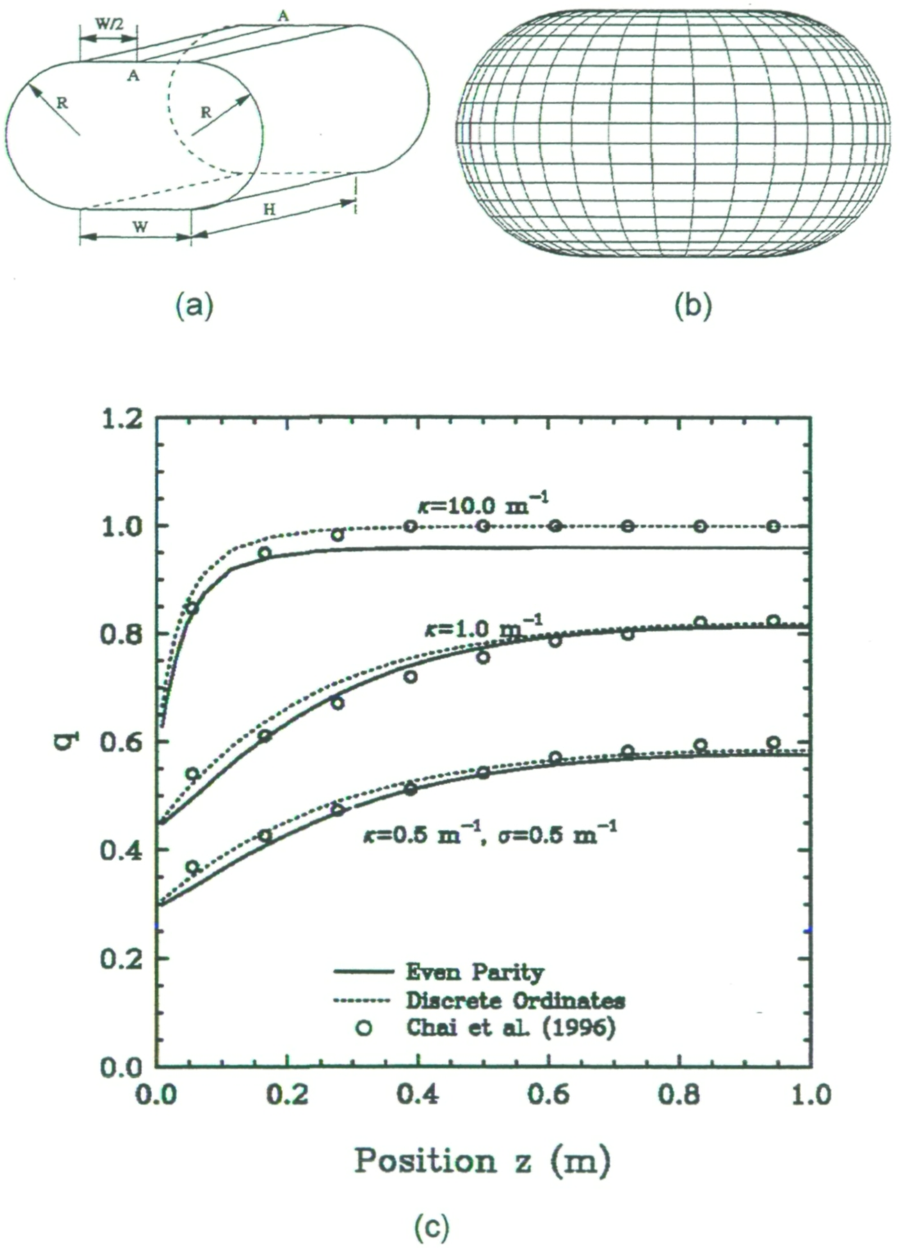


Figure 28. Three-dimensional elliptical enclosure: (a) schematic; (b) grid; (c) radiative heat flux distributions along the A-A line.

Table 1 lists the required CPU time and iterations for the discrete ordinates and even parity solutions for the above five problems. The data in the first problem are based on the clustered grid. For the cases without a scattering medium, the source terms in the discretized equations and boundary conditions are independent on the unknown intensities in the discrete ordinates solutions. Therefore, only two iterations are required to obtain the convergent solutions. In contrast, the EPF of the RTE has a totally different mathematical formulation from the conventional RTE and the resulting source terms in the discretized equations and boundary conditions become strongly dependent on the variable F in the even parity solutions. Thus, the CPU time and iterations are seen considerably higher in the even parity solutions, especially for the case with a small optical thickness. For the cases with a scattering medium, the CPU time and iterations are seen to increase in the discrete ordinates solution because the unknown intensities begin to appear in the source terms. However, they are still smaller than those in the even parity solution.

Table 1 CPU time and iterations

Problems	κ	σ	Even parity		Discrete Ordinates	
			CPU Time, s	Iterations	CPU Time, s	Iterations
2D Quadrilateral Enclosure	0.1	0.0	90.47	99	2.40	2
	1.0	0.0	28.30	34	2.32	2
	10.0	0.0	10.96	15	2.23	2
2D Curved Geometry No. 1	1.0	0.0	75.07	81	2.36	2
2D Curved Geometry No. 2	0.0	1.0	26.69	35	11.41	14
	1.0	0.0	41.46	51	2.14	2
3D Equilateral Triangular Enclosure	0.1	0.0	1018.40	495	3.53	2
	1.0	0.0	204.89	101	3.50	2
	10.0	0.0	46.31	23	3.39	2
3D Elliptical Enclosure	0.5	0.5	1830.51	124	60.22	6
	1.0	0.0	1737.52	113	30.87	2
	10.0	0.0	317.41	21	32.73	2

4.7 REFERENCES

1. Carlson, B. G., and Lathrop, K. D., 1968, "Transport Theory - The Method of Discrete Ordinates," *Computing Methods in Reactor Physics,* edited by H. Greenspan, C. N. Kelber, and D. Okrent, Gordon & Breach, New York, pp. 165-266.
2. Chai, J. C., Lee, H. S., and Patankar, S. V., 1994, "Treatment of Irregular Geometries Using a Cartesian Coordinates Finite-Volume Radiation Heat Transfer Procedure," *Numerical Heat Transfer*, Part B, Vol. 26, pp. 225-235.
3. Chai, J. C., Parthasarathy, G., Lee, H. S., and Patankar, S. V., 1995, "Finite Volume Radiative Heat Transfer Procedure for Irregular Geometries," *Journal of Thermophysics and Heat Transfer,* Vol. 9, pp. 410-415.
4. Chai, J. C., Moder, J. P., and Parthasarathy, G., 1996, "Radiation Heat Transfer in Three-Dimensional Irregular Geometries," AIAA Paper 96-1889.
5. Chen, Y. S., 1989, "Compressible and Incompressible Flow Computations with a Pressure Based Method," AIAA Paper 89-0286.
6. Cheong, K. B. And Song, T. H., 1995, "Examination of Solution Methods for the Second-order Discrete Ordinate Formulation," *Numerical Heat Transfer, Part B*, Vol. 27, pp. 155-173.
7. Dupont, T., Kendall, R. P., and Rachford, M. M., 1968, "An Approximate Factorization Procedure for Solving Self-Adjoint Elliptic Difference Equations," *SIAM Journal of Numerical Analysis,* Vol. 5, No. 3, pp. 559-573.
8. Fiveland, W. A., 1988, "Three-Dimensional Radiative Heat Transfer Solutions by the Discrete-Ordinates Method," *Journal of Thermophysics and Heat Transfer*, Vol.2, No. 4, pp. 309-316.
9. Fiveland, W. A., and Jessee, J. P., 1994, "Finite Element Formulations of the Discrete-Ordinates Method for Multidimensional Geometries," *Journal of Thermophysics and Heat Transfer*, Vol. 8, No. 3, pp. 426-433.
10. Fiveland, W. A., and Jessee, J. P., 1995, "Comparison of Discrete Ordinates Formulations for Radiative Heat Transfer in Multidimensional Geometries," *Journal of Thermophysics and Heat Transfer*, Vol. 9, No.1, pp. 47-54.

11. Hirsch, C., 1990, *Numerical Computation of Internal and External Flows*, Vol. 2, John Wiley & Sons, New York.
12. Kaplan, S., and Davis, J. A., 1967, "Canonical and Involuntary Transformations of the Variational Problems of Transport Theory," *Nuclear Science and Engineering*, Vol. 28, No. 1, pp. 166-167.
13. Langtangen, H. P., 1989, "Conjugate Gradient Methods and ILU Preconditioning of Non-Symmetric Matrix Systems with Arbitrary Sparsity Patterns," *International Journal for Numerical Methods in Fluid,* Vol. 9, No. 2, pp. 213-233.
14. Lewis, E., E., and Miller, W. F., 1984, *Computational Methods of Neutron Transport*, John Wiley & Sons, New York.
15. Liu, J., Shang, H. M., Chen, Y. S., and Wang, T. S., 1997, "Analysis of Discrete Ordinates Method with Even Parity Formulation," *Journal of Thermophysics and Heat Transfer*, Vol. 11, No. 2, pp. 253-260.
16. Parthasarathy, G., Lee, H. S., Chai, J. C., and Patankar, S. V., 1994, "Monte Carlo Solutions for Radiative Heat Transfer in Irregular Two-Dimensional Geometries," *Radiative Heat Transfer: Current Research*, HTD-Vol. 276, pp. 191-199.
17. Song, T. H., and Park, C. W., 1992, "Formulation and Application of the Second Order Discrete Ordinate Method," *Transport Phenomena Science and Technology 1992*, edited by Bu-Xuan Wang, Higher Education Press, Beijing, China, pp. 833-841.
18. Thompson, J. F., Warsi, Z. U. A., and Mastin, C. W., 1985, *Numerical Grid Generation - Foundation and Applications*, North-Holland, New York.
19. Viskanta, R. and Menguc, M. P., 1987, "Radiation Heat Transfer in Combustion Systems," *Progress in Energy and Combustion Science*, Vol. 13, No. 2, pp. 97-160.
20. Wang, T. S., and Chen, Y. S., 1993, "Unified Navier-Stokes Flowfield and Performance Analysis of Liquid Rocket Engines," *Journal of Propulsion and Power,* Vol. 9, No. 5, pp. 678-685.

5. INJECTOR SPRAY COMBUSTION BENCHMARK CASE STUDY

The present FDNS model is used to solve a LOX/GH2 coaxial injector spray combustion flowfield that was one of the cases investigated experimentally at Penn State University. The case considered has O/F mass flow rate ratio around 8 (LOX and H2 flow rates are 0.393 lbm/sec and 0.0489 lbm/sec respectively). A two-block grid (with sizes of 21 x 31 and 201 x 71) was used for numerical computation. A cold flow solution with VOF (Volume-of-Fluid) and spray models turned on was first estabilished. Ignition heat source was then introduced near the injector nozzle lip to establish the flame. This way of igniting the flame is more effective since the flame holding flow pattern can be maintained early in the computation. Thus, the total computational time for a case can be reduced. It is also found important to properly initialize the flowfield before starting the reacting flow. Since the experiment was conducted with nitrogen and air in the chamber initially, it would be a better representation of the flowfield if it is initialized the same way numerically. The reason for this is that any species trapped in the recirculating flow near the backstep is found to take a very long time before it can be replaced by the injected species. Thus, the flame shape and temperature behind the backstep are affected by the residual species that was there initially.

Figure 23 shows the flowfield solution, LOX jet and particle patterns near the injector. The hydrogen injection flow is greatly affected by the spray and mass addition as a result of evaporating LOX droplets. Figure 24 illustrates the solution of the temperature field, oxygen concentration and OH concentration contours. The results indicate that a substantial amount of unburned oxygen is exiting the nozzle as a result of poor mixing even though the injection O/F ratio is about stoichiomatric.

O/F RATIO = 8.0308 (LOX/GH2 INJECTOR)

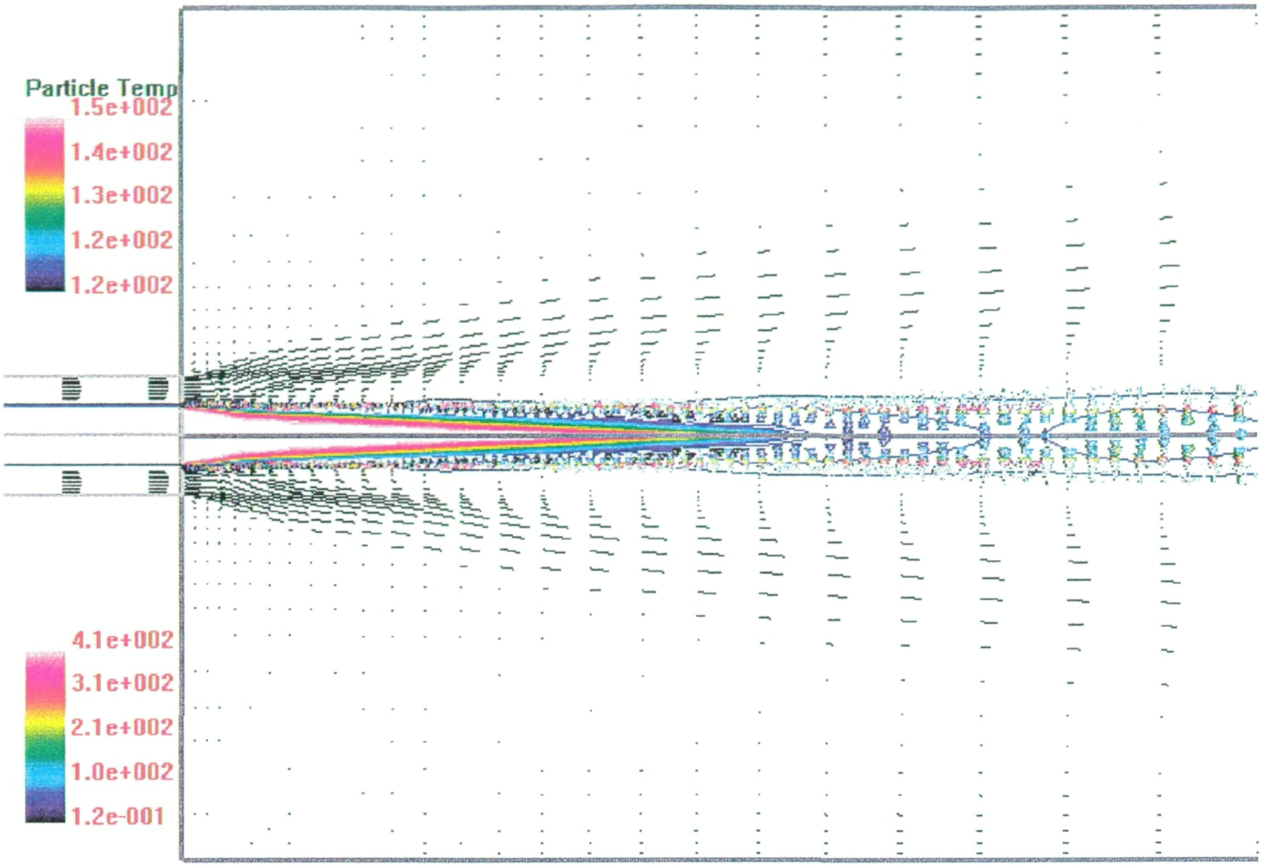
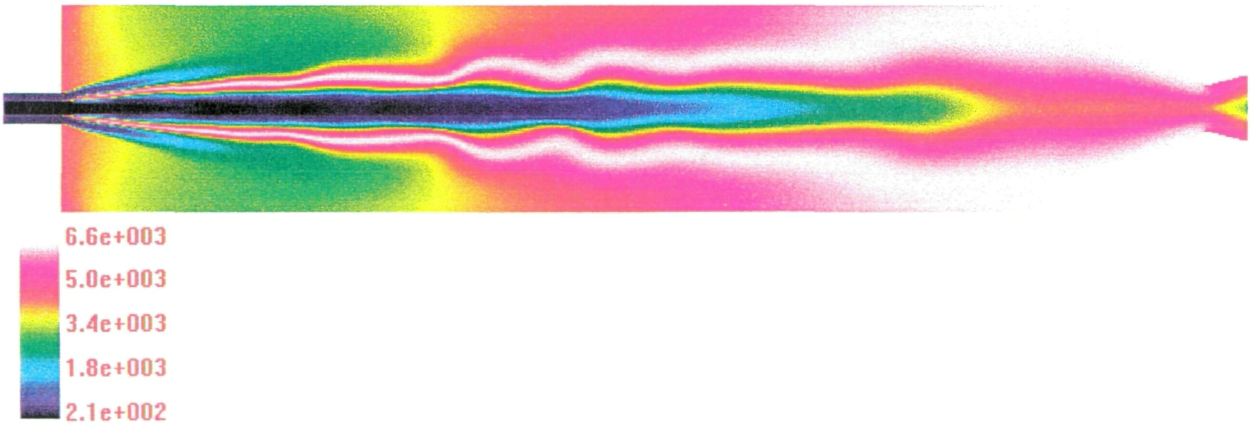
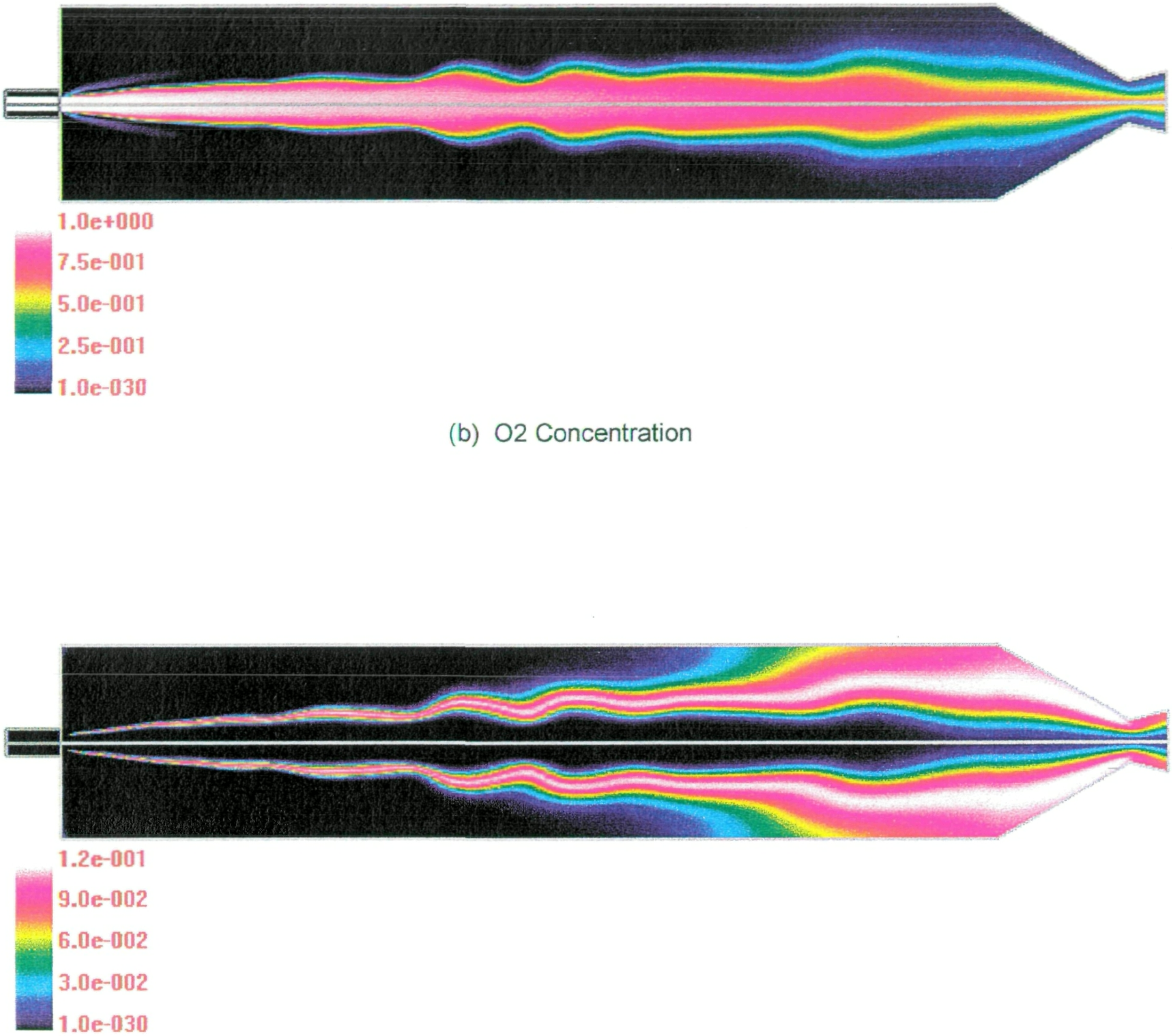


Figure 23. Liquid jet, velocity vectors and particle patterns of a injector spray.



(a) Temperature Contours (R)



(b) O2 Concentration

(c) OH Concentration

Figure 24. Temperature field, O2 concentration and OH concentration contours of a burning LOX/GH2 spray.

6. USER'S GUIDE TO THE NEW PARALLEL FDNS MODEL

The following two sections are very important for the user in order to setup the FDNS version 4.5 correctly. Section 6.1 describes the necessary steps to setup a parallel computing environment for the current FDNS model. The single processor option is maintained as was for the previous versions. Functions of a new FDNS pre-processor and data conversion utility programs are also defined. Next, in Section 6.2, a detailed description of the new input data format is included.

6.1 PARALLEL COMPUTATIONAL ENVIRONMENT SETUP

The current FDNS parallel version requires assignment of computer resource to be used for the computation (specify how many CPUs or how many computers in user's local network). The PVM library needs to setup the computational environment for message passing as a mechanism for domain boundary data transfer. The following is a description of steps needed to setup the environment (which is the same as the readme file in the software package of FDNS version 4.5).

List of the readme file:

Parallel Computing of FDNS Using PVM

===

by Huan-Min Shang, Engineering Sciences, Inc.

Revised: 2/9/1998

===

1. Set your .cshrc file by inserting the following lines:

```
#set for PVM
 setenv   PVM_ROOT    '/home2/PVM/pvm3'  !!! depend on where
                                             pvm installed
 if ($?PVM_ROOT) then
```

```
      setenv PVM_ARCH `$PVM_ROOT/lib/pvmgetarch`
      set path=($path $PVM_ROOT/lib)
      endif
```

2. Modify your rhosts file to include the hosts to be used

3. Prepare the directories to be used at each host, such as "parallel"

4 Prepare "hostfile" file in the master directory.

```
#  note: "dx" must be included for win32 hosts
   iris.esi.inc
   iris2.esi.inc
   ibm.esi.inc
   ibm2.esi.inc
   snow.esi.inc
   solarwind.esi.inc
   typhoon.esi.inc    dx=c:\pvm3\lib\win32\pvmd3.exe
   flurry.esi.inc     dx=d:\pvm3\lib\win32\pvmd3.exe
   hurricane.esi.inc  dx=c:\pvm3\lib\win32\pvmd3.exe
   storm.esi.inc      dx=c:\pvm3\lib\win32\pvmd3.exe
   cloud.esi.inc      dx=c:\pvm3\lib\win32\pvmd3.exe
```

5 Modifiy fort.11 file (only boldfaced data shown below affect the parallel model)

```
# IZT, JZT, KZT,  LPROC,  CBG1, CBV2
  51   51   33        1, -1400.  0.0
  51   45   33        2, -1400.  0.0
.............
#THCYCX, IZB1, IZF1,IJZ11,IJZ12,JKZ11,JKZ12,INONUF,IPROC1
#        IZB2, IZF2,IJZ21,IJZ22,JKZ21,JKZ22,IDFACE,IPROC2
 72.00     2     6     1     51     15     45        0      0
           2     5     1     51     15     45        0      0
```

```
0.00     4     3     1     51     1     33      31     0
         3     4     1     51     1     33       0     0
0.00     1     2     1     51     1     33      31     0
         2     4     1     51     1     33       0     0
. . . . . . . . . . . . . . . . . . . . . .
```

```
# where LPROC means the process number that this block is involved with.
  IDFACE, IPROC1 and IPROC2 are 0 in the master directory. They will be updated
  in the slave directories after runing the FDNS pre-processor, xprep, program
  IDFACE is the global index of the zonal interface.  IPROC1 and IPROC2 indicate
  process numbers for each piece of the interface surfaces in which it is located.  If
  IPROC1 = IPROC2, both interfaces are within the same process.
# In the current version, INONUF=0  is used for fdns4.0 interface methods.  In this
  case, both interfaces must be in the same host.
# INONUF = 21 is used for patched (matched)  grids in parallel/serial computing.
# INONUF = 22 is used for patched (matched)  grids in parallel/serial computing
  with cyclic bc.
# INONUF = 31 is used for general patched grids in parallel/serial computing.
# INONUF = 32 is used for general patched grids in parallel/serial computing with
  cyclic bc.
# INONUF = 34 is used for patched grids in turbomachinery parallel/serial
  computing with moving  interface. If thcycx=0.0, for 2-d case, CBV is not zero.  If
  thcycx.ne.0, for 3-d case, CBG must be provided.
# INONUF = 35 Same as INONUF=34 for 3-d turbomachinery flows, but grids are
  matched in the radial direction.  This setting can save CPU time for simpified
  interfaces
# INONUF = 51 is used for matched patched grids in parallel/serial computing as
  INONUF = 21, but with different method of interpolating interface values.
#
#=============================================================#
```

Also include parallel input section:

```
#
parallel
1   host=iris2.esi.inc
    edir=/home3/shang/pvm/fdns402/src
    wdir=/home3/shang/pvm/fdns402/pro5/parallel
2   host=solarwind.esi.inc
    edir=/home3/shang/pvm/fdns402/src/sun
    wdir=/export/home/shang/parallel/pro5
3   host=iris.esi.inc
    edir=/home3/shang/pvm/fdns402/src
    wdir=/home/shang/parallel/pro5
4-5 host=solarwind.esi.inc
    edir=/home3/shang/pvm/fdns402/src/sun
    wdir=/export/home/shang/parallel/pro5
6   host=hurricane.esi.inc
    wdir=c:\pvm3\temp\shang\pro5
7   host=flurry.esi.inc
    wdir=d:\pvm3\temp\shang\pro5
end of parallel

#
#==========================================================================#
# where 'host=' is the machine's hostname
#       'edir=' is the FDNS source file directory. It is assumed that all
#               the other executable files are also in the same directory
#               for this host. For Win32, all executables must be copied
#               to $pvm_root\bin\win32 because the current win32 version
#               can not identify a specified directory for executables.
#       'wdir=' is the working directory
#        note:  1. the master host must be the first host
#               2. the key-words "parallel" and "end" must be included
#               3. under "pvm>" prompt, check the host names. both must
#                  be exactly the same to run pvm.
#               4. do not exceed 80 column in this input section
```

#===#

6. Type 'pvmd hostfile &' in the master directory/host to start PVM.

7. Parallel pre-processor & utility program: **xprep**

   pvmd must be started before runing xprep

   a hostfile must be provided to run xprep

   type '../src/xprep' to run xprep

   control parameters:

```
ICONT = 0: eixt
        1: distribute FDNS input file:   fort.11
        2: distribute plot3d grid file:  fort.12
        3: distribute restart file:      fort.13
        4: collect plot3d grid file:     fort.22
        5: collect restart file:         fort.23
        6: collect plot3d solution files: fort.92, fort.93, fort.94, etc.
        7: mv fort.22 and fort.12 are in the current (master) dir
        8: mv fort.23 and fort.13 are in the current (master) dir
        9: mv f(proc#).22 f(proc#).12 for restarting in the slave dir
       10: mv f(proc#).23 f(proc#).13 for restarting in the slave dir
```

   steps 1-3 are used to build parallel input data

   steps 4-6 are used to collet parallel output data

   steps 7-9 are used to restart FDNS

8. Type '../src/xfdns -p ' in the master directory to run parallel FDNS.

   Type '../src/xfdns -np' in the master directory to run non-parallel FDNS.

9. When the project is finished, type the following command to kill PVM:

```
pvm (Return)
>  halt (Return)
```

or the following command to reset (kill all jobs, but not exit pvm):

```
pvm (Return)
>  reset (Return)
```

10. Example problems:

cav4:	2d	4-zone driven cavity problem
back3:	2d	3-zone backward facing step incompressible laminar flow
casefel:	2d	3-zone flame holding
ssme:	2d	1-zone ssme nozzle flow (not parallel)
pro5:	3d	5-zone propeller flow problem
Propeller:	3d	4-zone propeller flow problem
blunt3d:	3d	2-zone supersonic blunt body flow
blu12:	3d	12-zone supersonic blunt body flow
port3d:	3d	5-zone port flow

11. Related modification in FDNS can be found using " grep 'hms-pvm' *.f " in the directory src. The main PVM routine is in pvm.f. The interface treatment and CG solver are in cgs.f. File fpvm h must be provided in the src directory.

12. WIN32 problems:

PVM for Microsoft NT version, WIN32, has some differeces with the unix version:

1) "dx=c:\pvm3\lib\win32\pvmd3.exe" must be included in host file or when add WIN32 host
2) The executable files must be placed in c:\pvm3\bin\win32 directory and set empty for "edir"

3) copy fort 11 fort.12 fort 13 to run xprep when using NT as master

13 Some utility programs:

1) xfcon: file convertor

ICONT= 1· Convert fort.11 file from old format to new format

ICONT= 2 Convert restart file from old format to new format
and from unf to fmt or fmt to unf

ICONT= 3 Convert plot3dg file from unf to fmt or fmt to unf

ICONT= 4 Convert plot3dq file from unf to fmt or fmt to unf

(unf and fmt stand for the unformatted and formatted file formats respectively)

2). xdecm domain decomposition

ICONT= 1· Domain decomposition for fort.11 file (to be developed)

ICONT= 2 Domain decomposition for grid file

ICONT= 3 Domain decomposition for restart file

ICONT= 4 Restore restart file back to the initial grid

input file for xdecm: decm.in

```
#IZON1 ! NUMBER OF ZONES & GRID SIZES OF THE ORIGINAL GRID
     3
#   IZ1  IZT1  JZT1  KZT1
     1    51    51    33
     2   151    45    33
     3    51    10    33
#IZON2  ! NUMBER OF ZONES AFTER GRID DECOMPOSED
     5  ! IZ2: NEW ZONE NUMBER; IZ1: ORIGINAL ZONE NUMBER
#   IZ2   IZ1   II1   II2   JJ1   JJ2   KK1   KK2
     1     1     1    51     1    51     1    33
     2     2     1    51     1    45     1    33
     3     3     1    51     1    10     1    33
     4     2    51   101     1    45     1    33
     5     2   101   151     1    45     1    33
```

6.2 INPUT DATA FILE FORMAT FOR FDNS VERSION 4.5

The input data format for the FDNS version 4.5 is different from that of previous versions. In addition to the changes necessary for specifying process number in parallel computation mode as described in Section 6.1, multiple outlet pressure conditions, new '#' sign comment card feature, name searching for physical submodels and symbolic input specification for chemical reaction equations are among the new improved input data features for the current model. A sample input data file, fort.11, is first listed below, following by a description defining the input parameters. In the current input file, any line start with a '#' sign can be inserted as comment card or for clarity of the file without affecting the data read sequence.

List of a sample fort.11:

```
################################################################################
TITLE: (2D SSME N., 100% PL, Finite-rate Case, 9-Step)
################################################################################
#IDIM
    2
#IZON,IZFACE,  IBND,    ID, ISNGL, INPNT
    1      0      3      1      0      2
#  IZT,  JZT,  KZT,LPROC,      CBG1,      CBV2
   101    71     1     1  0.000E+00  0.000E+00
#THCYCX, IZB1, IZF1,IJZ11,IJZ12,JKZ11,JKZ12,INONUF,IPROC1
#        IZB2, IZF2,IJZ21,IJZ22,JKZ21,JKZ22,IDFACE,IPROC2
#IBCZON,IDBC,ITYBC,IJBB,IJBS,IJBT,JKBS,JKBT,IVFINT,    =PRAT,=IPZ,=IPI,=IPJ,=IPK
      1    1     2  101    1   71    1    1      0 -1.000E+00   1  101   70    1
      1    2     1    1    1   71    1    1      0 -1.000E+00   1  101   70    1
      1    4     3    1    1  101    1    1      0 -1.000E+00   1  101   70    1
#IWBZON,  L1,  L2,  M1,  M2,  N1,  N2,IWTM,      HQDOX,IWALL,    DENNX,    VISWX
      1    1  101   71   71    1    1   -1   0.000E+00     0 1.000E+00 1.000E+00
#ISNZON, ISNBC, ISNAX, ISNBS, ISNBT
#IPRZON,  LP1,  LP2,  MP1,  MP2,  NP1,  NP2
      1     1   101     1     1     1     1
      1     1   101    71    71     1     1
#IDATA, IGEO, =ITT,=ITPNT,=ICOUP,NLIMT,  IAX, ICYC
     1    43   500     50      1     1     2     0
#      =DTT,=IREC,  =REC,=THETA,=BETAP,=IEXX
  1.000E-03     4 0.2500 1.0000 1.0000     1
#   IPC,   JPC,  =IMN,  =JMN,   =GFORC1,   =GFORC2,   =GFORC3
    102      1    201      1  0.000E+00  0.000E+00  0.000E+00
#      VISC,IG,ITURB,   AMC,  GAMA,      =CBE,      =CBH,    =EREXT
  2.821E-07  2     2 0.1984 1.1957  0.000E+00  0.000E+00  1.000E-06
#=ISWU,=ISWP,=ISWK,=ISKEW
    93    93    93     1
#INSO(IEQ):
# U, V, W, T,DK,DE,07,08,09,VS,FM,SP
  1  1  0  1  1  1  0  0  0  1  1 21
```

```
#IUNIT,     DENREF,      UREF,       TREF,       XREF
     2  1.731E-02  1.058E+03  5.400E+02  1.000E+00
#=IGDINN,=IOFINN,=IOFOUT,=IOP3DOUT (1:Unf, 2:Fmt) ! IO FORMAT CONTROL
       1       1       1         1
********************************************************************************
PTFREE,         TMFREE,    XMFREE, ISPN2, ISPO2
  1.000E+00  3.000E+02  1.000E-01      7      2
********************************************************************************
ISPVOF
   0
#DENVOF(K),K=1,ISPVOF

#IVFPTL(K),K=1,ISPVOF

********************************************************************************
NGAS
   7
H2O                 J 3/79H  2.O  1.   0.   0.G   300.000  5000.000     18.01520
  .26340650E+01  .31121900E-02 -.90278450E-06  .12673050E-09 -.69164730E-14
 -.29876260E+05  .70823870E+01  .41675560E+01 -.18106870E-02  .59450880E-05
 -.48670870E-08  .15284140E-11 -.30289550E+05 -.73088000E+00
O2                  J 3/77O  2.   0.   0.   0.G   300.000  5000.000     31.99880
  .36122140E+01  .74853170E-03 -.19820650E-06  .33749010E-10 -.23907370E-14
 -.11978150E+04  .36703310E+01  .37837140E+01 -.30233630E-02  .99492750E-05
 -.98189100E-08  .33031830E-11 -.10638110E+04  .36416340E+01
H2                  J 3/77H  2.   0.   0.   0.G   300.000  5000.000      2.01580
  .30558120E+01  .59740400E-03 -.16747470E-08 -.21247540E-10  .25195490E-14
 -.86168480E+03 -.17207070E+01  .29432330E+01  .34815510E-02 -.77713820E-05
  .74997490E-08 -.25203380E-11 -.97695410E+03 -.18186140E+01
O                   J 3/77O  1.   0.   0.   0.G   300.000  5000.000     15.99940
  .25342960E+01 -.12478170E-04 -.12562720E-07  .69029860E-11 -.63797100E-15
  .29231110E+05  .49628590E+01  .30309400E+01 -.22525850E-02  .39824540E-05
 -.32604920E-08  .10152040E-11  .29136530E+05  .26099340E+01
H                   J 3/77H  1.   0.   0.   0.G   300.000  5000.000      1.00790
  .25000000E+01  .00000000E+00  .00000000E+00  .00000000E+00  .00000000E+00
  .25474390E+05 -.45989840E+00  .25000000E+01  .00000000E+00  .00000000E+00
  .00000000E+00  .00000000E+00  .25474390E+05 -.45989840E+00
OH                  J 6/77O  1.H  1.   0.   0.G   300.000  5000.000     17.00730
  .28897810E+01  .10005880E-02 -.22048810E-06  .20191290E-10 -.39409830E-15
  .38857040E+04  .55566430E+01  .38737300E+01 -.13393770E-02  .16348350E-05
 -.52133640E-09  .41826970E-13  .35802350E+04  .34202410E+00
N2                  J 3/77N  2.   0.   0.   0.G   300.000  5000.000     28.01340
  .28532900E+01  .16022130E-02 -.62936890E-06  .11441020E-09 -.78057470E-14
 -.89008090E+03  .63964900E+01  .37044180E+01 -.14218750E-02  .28670390E-05
 -.12028880E-08 -.13954680E-13 -.10640790E+04  .22336290E+01
CO                  J 9/65C  1.O  1.   0.   0.G   300.000  5000.000     28.01040
  .29840700E+01  .14891390E-02 -.57899680E-06  .10364580E-09 -.69353550E-14
 -.14245230E+05  .63479160E+01  .37100930E+01 -.16190960E-02  .36923590E-05
 -.20319670E-08  .23953340E-12 -.14356310E+05  .29555350E+01
CO2                 J 9/65C  1.O  2.   0.   0.G   300.000  5000.000     44.00980
  .44608040E+01  .30981720E-02 -.12392570E-05  .22741330E-09 -.15525950E-13
 -.48961440E+05 -.98635980E+00  .24007800E+01  .87350960E-02 -.66070880E-05
  .20021860E-08  .63274040E-15 -.48377530E+05  .96951460E+01
********************************************************************************
NREACT
   9
#CHEMICAL REACTIONS:
    1   .1700E+14    .0000E+00    .2407E+05     0    0
                                                        O2 + H2 = 2 OH
    2   .2190E+14    .0000E+00    .2590E+04     0    0
                                                        H2 + OH = H2O + H
    3   .6023E+13    .0000E+00    .5500E+03     0    0
                                                        2 OH = H2O + O
    4   .1800E+11    .1000E+01    .4480E+04     0    0
                                                        H2 + O = OH + H
    5   .1220E+18   -.9100E+00    .8369E+04     0    0
                                                        O2 + H = OH + O
    6   .1000E+17    .0000E+00   .0000E+00  999     0
                                                      H + O = OH
    7   .2550E+19   -.1000E+01   .5939E+05  999     0
                                                      2 O = O2
    8   .5000E+16    .0000E+00   .0000E+00  999     0
                                                      2 H = H2
    9   .8400E+22   -.2000E+01   .0000E+00  999     0
                                                      OH + H = H2O
********************************************************************************
```

All parameter names listed above that are preceded by an equal sign '=' are run-time modifiable (updated every 10 time steps) by editing the fort.11 file.

FDNS Version 4.5 Input Data Format:

Card Group #1. gives the case title and identifies whether the problem is 2-D or 3-D

Format

#IDIM,

<————(one data line)

Definition:

IDIM 2 for 2-dimensional flow problems

3 for 3-dimensional flow problems

Card Group #2 specifies zonal information and number of flow and wall boundaries

Format:

#IZON, IZFACE, IBND, ID, ISNGL, INPNT,

<————(one data line)

Definition·

IZON number of zones or mesh blocks

IZFACE number of patched interfaces

IBND number of flow boundaries (e g inlet, outlet or symmetry planes)

ID number of wall elements (blocks)

ISNGL number of singularity lines

INPNT number of data output specifications along grid lines

Card Group #3 specifies zonal grid size and zonal rotational/translational speeds

Format

```
#IZT,   JZT,   KZT,  CBG1, CBV2,
                                         <---------(IZ = 1, IZON)
```

Definition:

IZT(IZ) I-max in zone IZ

JZT(IZ) J-max in zone IZ

KZT(IZ) K-max in zone IZ

CBG1(IZ) rotational speed (RΩx/Uref) of zone IZ about X-axis

CBV2(IZ) translational speed of zone IZ in Y-direction

Card Group #4: identifies the zonal interface matching indices.

Format:

```
#THCYCX, IZB1,  IZF1,  IJZ1,  IJZ2,  JKZ1,  JKZ2, INONUF,IPROC1,
#              IZB2,  IZF2,  IJZ1,  IJZ2,  JKZ1,  JKZ2, IDFACE,IPROC2,
                                         <---------(2*IZFACE data lines)
```

Definition:

THCYCX angle (in degrees) between two interfaces (positive for Right-Hand-Rule along X-axis for turbomachinery applications, 0.0 for regular continuous interfaces)

IZB1 zonal index of interface plane # 1

IZF1 interface plane identifier for plane #1

1: I = I-max or East

2: I = 1 or West

3: J = J-max or North

4: J = 1 or South

5: K = K-max or Top

6: K = 1 or Bottom

IZB2 zonal index of interface plane # 2

IZF2 interface plane identifier for plane #2

IJZ1 the starting point of the first running index on the interface plane

IJZ2 the ending point of the first running index on the interface plane

JKZ1 the starting point of the second running index on the interface plane

JKZ2 the ending point of the second running index on the interface plane

Example — If IZF1 or IZF2 is either 1 or 2 then IJZ1 and IJZ2 are the indices in J-direction and JKZ1 and JKZ2 are the indices in K-direction

If IZF1 or IZF2 is either 3 or 4 then IJZ1 and IJZ2 are the indices in I-direction and JKZ1 and JKZ2 are the indices in K-direction.

If IZF1 or IZF2 is either 5 or 6 then IJZ1 and IJZ2 are the indices in I-direction and JKZ1 and JKZ2 are the indices in J-direction

Notice — The interface patching surface indices for planes #1 and #2 (i e IJZ1, IJZ2, JKZ1, JKZ2) must have consistent running order

INONUF — patched grid options (please refer to Section 6 1 for details)

IDFACE — global index for the interface (to be assigned by the pre-processor, please refer to Section 6 1)

IPROC1 — interface #1 process number to be assigned in the pre-processor (set = 0 initially as user input)

IPROC2 — interface #2 process number to be assigned in the pre-processor (set = 0 initially as user input)

Care Group #5 — specifies flow boundaries (inlet, outlet, symmetry)

Format

```
#IBCZON, IDBC, ITYBC, IJBB, IJBS, IJBT, IKBS, IKBT, IVFINT, IPZ, IPI,IPJ,IPK,
                                      <---------(IBND data lines)
```

Definition

IBCZON — zonal index for the flow boundary

IDBC — boundary facing index

1	I = I-max	or East
2	I = 1	or West
3	J = J-max	or North
4	J = 1	or South
5	K = K-max	or Top
6	K = 1	or Bottom

ITYBC — identifies boundary type

-2 inlet fixing everything except pressure

-1: inlet fixing mass flow rates (e.g. solid fuel blowing surfaces)

0: inlet fixing everything (e.g. supersonic)

1: inlet fixing total pressure (compressible flow only)

2: outlet boundary

3: symmetry plane (can also be used for slip wall boundary conditions)

IJBB I, J or K location (depends on IDBC) of the boundary

IJBS, IJBT boundary starting and ending indices (for I or J)

JKBS,JKBT boundary starting and ending indices (for J or K)

IVFINT VOF species index for inlet boundaries

PRAT specifies outlet boundary condition options

-1.0: for supersonic outlet b. c.

0.0: for outlet mass conservation b. c.

0.0: for outlet fix pressure (in ATM) b. c. The outlet pressure reference point, (IPZN, IPI, IPJ, IPK) is used for anchoring the pressure specified.

IPZ zonal index for the pressure anchoring point

IPI I-index of the pressure anchoring point in zone IPZ

IPJ J-index of the pressure anchoring point in zone IPZ

IPK K-index of the pressure anchoring point in zone IPZ

Card Group #6: specifies wall block indices.

Format:

#IWBZON,L1,L2,M1,M2,N1,N2, IWTM, HQDOX, IWALL, DENNX, VISWX,

<---------(ID data lines)

Definition:

IWBZON zonal index for the wall block

L1, L2 starting and ending indices in the I-direction

M1, M2 starting and ending indices in the J-direction

N1, N2 starting and ending indices in the K-direction

IWTM solid-wall thermal boundary condition options

-1: for fixed wall-temperature

1: for heat-flux (= HQDOX) b. c.

HQDOX wall heat flux when IWTM = 1,

positive from wall to fluid (in Watts/m**2)

IWALL solid wall heat conduction option

0: to deactivate; 1: to activate

DENNX solid wall density*Cp (in joule/m**3-K)

VISWX solid wall thermal conductivity (in Watts/m-K)

*** When IWALL = 1 is selected, the program will set IWTM = -1, since this is a correct combination.**

Card Group #7: specifies the singularity lines.

Format:

#ISNZON, ISNBC, ISNAX, ISNBS, ISNBT,

<---------(ISNGL data lines)

Definition:

ISNZON zonal index for the singularity lines

ISNBC singularity line boundary facing index

1: I = I-max	or East
2: I = 1	or West
3: J = J-max	or North
4: J = 1	or South
5: K = K-max	or Top
6: K = 1	or Bottom

ISNAX orientation of the singularity line axis

for example: on I-J plane (ISNBC = 5 or 6)

ISNAX = 1 for I-axis

ISNAX = 2 for J-axis

on J-K plane (ISNBC = 1 or 2)

ISNAX = 1 for J-axis

ISNAX = 2 for K-axis

on K-I plane (ISNBC = 3 or 4)

ISNAX = 1 for I-axis

ISNAX = 2 for K-axis

ISNBS, ISNBT   starting and ending indices along ISNAX

Card Group #8   specifies the data output alogn grid lines

Format

#IPRZON, LP1, LP2, MP1, MP2, NP1, NP2,

<---------(INPNT data lines)

Definition

IPRZON   zonal index for the data print lines

LP1, LP2   I-index range for the output specification

MP1, MP2   J-index range for the output specification

NP1, NP2   K-index range for the output specification

Card Group #9   I/O parameters and problem control parameters.

Format

#IDATA, IGEO, ITT, ITPNT, ICOUP, NLIMT, IAX, ICYC,

<---------(one data line)

Definition:

IDATA   restart options

IDATA = 1 for regular restart runs Restart grid and flow files for 12 and fort 13 must be made available

IDATA = 2 for example start run Initial grid and flow data must be made available in the fexmp01 include file

IGEO   geometry parameter (for user applications)

IGEO = 1 is specifically for problems without inlets and outlets (e g cavity flows)

ITT   number of time steps limit

ITPNT   the frequency on printing out solutions (through files fort 22, fort.23, fort 91, fort 92 and fort.93)

ICOUP   number of pressure correctors (typically 1 for steady-state applications and 3-6 for transient or rough initial start applications)

NLIMT   typically 1, 0 for printing out the initial or restart files without going through solution procedures

IAX 1 for 2-D planar or 3-D flows

2 for 2-D axisymmetric flow problems

ICYC cyclic or periodic boundary conditions identifier

Currently, only ICYC = 3 is active for turbomachinery applications

Card Group #10 time-step size, upwind schemes and time-marching scheme selections.

Format

#DTT, IREC, REC, THETA, BETAP, IEXX,

<———(one data line)

Definition:

DTT non-dimensional time step size, DT*Uref/Xref

IREC selects upwind scheme options

0 for second-order upwind scheme

1 for third-order upwind scheme

2 for second-order central scheme

REC upwind damping parameter (0 1 recommended)

0 0 for second-order accuracy

1 0 for first-order upwind scheme

THETA time-marching scheme θ parameter

1 0 for steady-state applications

99 for implicit-Euler transient applications

0 5 for Crank-Nicholson second-order accurate transient applications

BETAP pressure updating under-relaxation parameter

typically 1 0, small values can be used to reduce the amount on pressure corrections for rough start initial runs

IEXX outlet extrapolation parameter for scalar quantities

1. for zero-gradient extrapolation

2: for linear extrapolation

Card Group #11 specifies inlet, outlet pressure points and data monitoring point

Format

#IPC, JPC, IMN, JMN, GFROC1, GFORC2, GFORC3

<————(one data line)

Definition:

IPC, JPC flowfield reference point

IPC grid index in zone JPC

(not the global grid index)

IMN, JMN solution monitoring point

GFORC1 X-direction gravitational acceleration (in m/sec**2)

GFORC2 Y-direction gravitational acceleration (in m/sec**2)

GFORC3 Z-direction gravitational acceleration (in m/sec**2)

Card Group #12. gives reference viscosity, Mach number and options of turbulence models

Format:

#VISC, IG, ITURB, AMC, GAMA, CBE, CBH, EREXT,

<————(one data line)

Definition:

VISC non-dimensional fluid viscosity

= 1/(Reynolds number)

= vis-ref/(den-ref)/(u-rerf)/(x-ref)

IG 1 for laminar flow option

2 for turbulent flow option

ITURB for turbulence model selection

1. for standard high-Re k-ε model

2 for extended high-Re k-ε model

3 for L-B low-Re k-ε model

4 for H-G low-Re k-ε model

AMC reference Mach number, = (u-ref)/(ref. sound speed)

GAMA reference specific heat ratio

CBE non-dimensional buoyancy force parameter

= Gr/Re**2, where Gr stands for the Grashoff number and Re is the flow Reynolds number

CBH used to activate compressibility corrections for the k-ε turbulence models

-1.0: for k-corrected model

-2.0: for ε-corrected model

(CBH = 0.1 to 0.7 is used for setting VOF liquid surface tension data listed in f4.f)

EREXT convergence criterion (typically 5.0E-05 for steady-state solutions)

Card Group #13: specifies number of zonal iterations in the matrix solver when INFACE is used for overlaid grid zonal interface interpolations and indicates orthogonal or non-orthogonal grid options.

Format:

#ISWU, ISWP, ISWK, ISKEW,

<----------(one data line)

Definition:

ISWU martix solver selector for the momentum and energy equations, typically 93

ISWP matrix solver selector for the pressure correction equation typically 93

ISWK matrix solver selector for other scalar equations typically 93

ISKEW non-orthogonal grid viscous flux option

0: for orthogonal grid

1: for non-orthogonal grid

Card Group #14: specifies which equations are to be solved.

Format:

#INSO(IEQ):

#U, V, W, TM, DK, DE, 7, 8, 9, VS, FM, SP,

<----------(one data line)

Definition: (0 to deactivate; 1 to activate)

U, V, W	for the momentum equations
TM	for the energy equation
DK, DE	for the turbulence model
7	for VOF transport equaiton
8, 9	not used
VS	for updating the turbulence eddy viscosity
FM	for the species mass-fraction equations
SP	for calculating the gas thermal properties *(Also, SP = 1 is for penalty-function chemical reaction model and SP = 21 is for point-implicit chemical reaction model and the later one is recommended)*

Card Group #15 specifies number of gas species and reactions, and gives the reference conditions

Format

#IUNIT, DENREF, UREF, TREF, XREF,

<---------(one data line)

Definition

IUNIT	1	for SI-unit reference conditions
	2	for English-unit reference conditions
DENREF	reference density (in kg/m**3 or slug/ft**3)	
UREF	reference velocity (in m/sec or ft/sec)	
TREF	reference temperature (in °K or °R)	
XREF	reference length (in m or ft)	

Card Group #16: specifies I/O data file formats

Format

#IGDINN,IOFINN,IOFOUT,IOP3DOUT,

<---------(one data line)

Definition:

(1 for unformatted data and 2 for formatted data for the following parameters)

IGDINN	grid input file (in PLOT3D format), fort 12
IOFINN	FDNS restart input Q-files, fort 13 & fort 14

IOFOUT	FDNS output Q-files, fort.13 & fort.14
IOP3DOUT	PLOT3D format output files, fort.22, fort.92 & fort.93, etc.

Card Group #17: specifies the freestream conditions

Format:

#PTFREE, TMFREE, XMFREE, ISPN2, ISPO2,

<---------(one data line)

Definition:

PTFREE	freestream total pressure (in ATM)
TMFREE	freestream temperature (in deg-K)
XMFREE	freestream Mach number
ISPN2	index for N2 in CEC species list of group #19
ISPO2	index for O2 in CEC species list of group #19

Card Group #18: identifies the VOF species properties

(This group is only necessary when INSO(7) = 1 in Card Group #14)

Format:

#ISPVOF,

<---------(one data line)

#(DENVOF(K), K = 1,ISPVOF)

<---------(one data line)

#(IVFPTL(K), K = 1, ISPVOF)

<---------(one data line)

Definition:

ISPVOF	number of VOF species to be read
DENVOF	density of VOF species (same unit as reference values)
IVFPTL	gas-index for the VOF particle species (use the index sequence in Card Groups #19)

Card Group #19 include the CEC thermodynamics data here

Format:

NGAS (This title must always kept unaltered)

<———(one data line)

Name, Molecular Weight, Coefficients (7 x 2)

<———(4*NGAS lines)

Definition

NGAS number of gas species CEC tables to be read

= 0 for ideal gas cases

> 0 for CEC real gas cases

FDNS reads in the data in CEC format.

Card Group #20: specifies the finite-rate reaction steps

Format

NREACT (This title must always kept unaltered)

<———(one data line)

#REACTION Species Names, N = 1,NGAS (this is a title line)

IREACT, A, B, E/RT, ITHIRD, IGLOB

Reaction Equation for STOCEF(N, IREACT)

Reaction Equation for STOCEG(N, IREACT) — If IGLOB = 2

<———(NREACT sets)

Definition.

NREACT number on reaction steps to be used

= 0 for non-reacting flow

> 0 for finite-rate reacting flow

IREACT reaction step counter

A reaction rate leading constant

B reaction rate temperature exponent

E/RT reaction rate activation energy constant

ITHIRD third-body reaction indicator

0· deactivated

N for using the Nth species as third-body

999: for global (every species) third-body

IGLOB global reaction model indicator

0: deactivated

1: type 1 use only one line of STCOEF

2: type 2 use STCOEF plus STCOEG for power dependency effects

Equation for STCOEF, e.g. `O2 +  2  H2 = 2 H2O`

Equation for STCOEG, e.g. `0.1 O2 + 0.5 H2 =`

Card Group #21: provides particle input control **(Card Groups #21 and #22 are one set)**

Format:

#IDPTCL, IPREAD, ISPRAY,

<----------(one data line)

Definition:

IDPTCL number on particle initial condition input lines

0: to deactivate particulate phase option

IPREAD 1 for reading in particle data (fort.14) from upstream domain (this allows transferring the outlet particle data from the upstream domain solutions to the inlet boundary for succeeding domain computations -- especially useful for multi-phase rocket plume simulations)

ISPRAY to turn on spray atomization particle models

1: to activate

0: to deactivate (steady-state model used)

New Particle Input for Impinging Injectors:

Card Group #22: for reading in particle initial conditions (the following input data are for steady-state runs only)

Format:

#IPTZON,IDBCPT,LPTCL1,LPTCL2,MPTCL1,MPTCL2,NPTCL1,NPTCL2,

#XXNC,YYCN,ZZCN,UUPTCL,VVPTCL,WWPTCL,XTHE1,XTHE2,XTHE3,

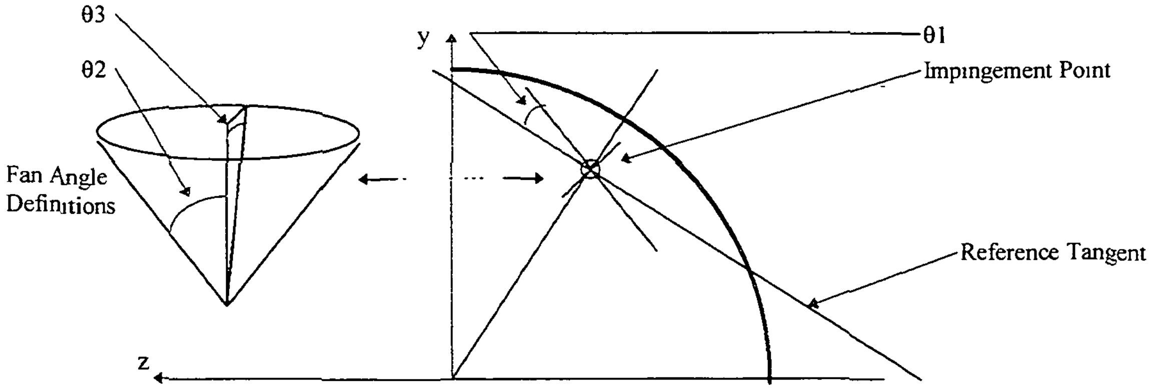
#ITPTCL,DDPTCL,DNPTCL,WDMASS,HTPTCL,ISPTCL,

<---------(3*IDPTCL data lines)

Definition:

IPTZON zonal index for the particle initial position

IDBCPT	I-, J- or K-plane identifier
	1 for I-plane (plane normal to I lines)
	2 for J-plane (plane normal to J lines)
	3 for K-plane (plane normal ro K lines)
LPTCL1, LPTCL2	I-interval for the particle initial position
MPTCL1,MPTCL2	J-interval for the particle initial position
NPTCL1,NPTCL2	K-interval for the particle initial position
XXCN	initial particle x-coordinate (non-dimensional)
YYCN	initial particle y-coordinate (non-dimensional)
ZZCN	initial particle z-coordinate (non-dimensional)
UUPTCL	initial particle u-velocity (m/sec)
VVPTCL	initial particle v-velocity (m/sec)
WWPTCL	initial particle w-velocity (m/sec)
XTHE1 (θ1, deg)	spray fan plane angle from tangent (see schematics below)
XTHE2 (θ2, deg)	spray open angle in the plane of fan (see schematics below)
XTHE3 (θ3, deg)	spray angle off the plane of fan (see schematics below)
ITPTCL	*number of particle groups (trajectories) starting from each* grid cell
DDPTCL	particle diameter in μm
DNPTCL	particle density in kg/m**3
WDMASS	particle mass flow rates for the current particle group and area involve of the current input line (kg/sec)
HTPTCL	particle initial temperature in deg-K
ISPTCL	particle species (0:AL2O3, 1 RP1, 2 LOX)



Card Group #23 for identifying vehicle bodies to calculate drag, thrust and momentum
(This data set can be placed anywhere between data groups below Card Group #16)

Format

```
#IDBODY, IDUNIT
                                          <--------(1 data line)
# ----IDBODY sets for the following input group
#          IDNUMB, IINLET
                                          <--------(1 data line)
#                   IDFORC, (I = 1, IDNUMB)
                                          <--------(1 data line)
#                   IDMONT, (I = 1, IINLET)
                                          <--------(1 data line)
```

Definition

IDBODY	number of vehicle bodies to be specified
IDUNIT	units of the forces 1 for SI units and 2 for English units
IDNUMB	number of wall segments to represent the body
IINLET	number of inlet boundaries involved in this body force set
IDFORC	wall block indices as in the order appeared in Card Group #6
IDMONT	inlet boundary indices as in the order appeared in Card Group #5

7. CONCLUSIONS

A parallel computing algorithm has been developed for the FDNS, pressure-based, general-purpose CFD code using domain decomposition technique. A general interface solver suitable for single or parallel computers has been developed. The message passing library PVM has been employed for information exchange of parallel computers. Numerical applications have been conducted for two and three dimensional fluid dynamics problems. Impressive parallel efficiency has been reached for computational intense cases. Further extension to the present study regarding parallel computational modeling can be focused on high performance, load balancing and user friendly interface implementations.

A fully conservative general 3-D patched grid interface module has been developed and validated in the current study. The interface module consists of subroutines that translates patched structured interface grids into a new set of unstructured cells representing a common interface for the original pair of interfaces. This method guaranteed the conservation of flow variables through the interface both locally and globally.

Preparation of the new input and control parameters for the FDNS has been incorporated in a pre-processor and a error checking module. Domain decomposition assignment, CPU/host computer assignment and input/flow file conversion utilities are included in the pre-processor.

In the radiation model study, the conventional RTE and the EPF of the RTE have been employed to investigate multi-dimensional radiative heat transfer problems containing absorbing-emitting and scattering media in general body-fitted coordinates by using the DOM. Compared to the conventional RTE, the EPF is more sensitive to the grid layout and it requires a clustered grid near the wall in order to provide accurate results in radiative wall heat flux. With an appropriate selection of a grid, the even parity solution generally has an accuracy close to the conventional discrete ordinates solution in a body-fitted coordinate system. However, it usually requires much more CPU time and iterations to converge, especially for the case with a small optical thickness, although the even parity solution only needs to consider half of the discrete directions as used in the discrete ordinates solution The present study clearly demonstrates that the DOM with the conventional RTE is more robust than the DOM with the EPF Therefore, the future development of an accurate and efficient radiative transfer model in a body-fitted coordinate system should only focus on the conventional RTE with the DOM or other radiation methods

Example 2-D and 3-D flow cases using the present parallel algorithm have been studied to assess the factor of computational speed up of the current approach. A benchmark injector spray combustion case has been analyzed with the present model. The results have shown a large gain in project turnaround speed with efficiency reaching 90 percent for 3-D cases. This proves the effectiveness of current parallel CFD method in reducing the computational turnaround time for all test cases included in this report and that the present model will be very useful in general engineering design analysis applications.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
April 4, 1998

3. REPORT TYPE AND DATES COVERED
Final Technical Report

4. TITLE AND SUBTITLE
INJECTOR DESIGN TOOL IMPROVEMENTS

5. FUNDING NUMBERS
H-28621D

6. AUTHOR(S)
Yen-Sen Chen, Huan-Min Shang, Hong Wei
and Jiwen Liu

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Engineering Sciences, Inc.
1900 Golf Road, Suite D
Huntsville, AL 35802

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
George C. Marshall Space Flight Center
Marshall Space Flight Center, AL 35812

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
Kevin Tucker / Technical Monitor

12a. DISTRIBUTION/AVAILABILITY STATEMENT
UNLIMITED

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The major emphasis of the current effort is in the development and validation of an efficient parallel machine computational model, based on the FDNS code, to analyze the fluid dynamics of a wide variety of liquid jet configurations for general liquid rocket engine injection system applications. This model includes physical models for droplet atomization, breakup/coalescence, evaporation, turbulence mixing and gas-phase combustion. Benchmark validation cases for liquid rocket engine chamber combustion conditions will be performed for model validation purpose. Test cases may include shear coaxial, swirl coaxial and impinging injection systems with combinations LOX/H2 or LOX/RP-1 propellant injector elements used in rocket engine designs. As a final goal of this project, a well tested parallel CFD performance methodology together with a user's operation description in a final technical report will be reported at the end of the proposed research effort.

14. SUBJECT TERMS
Injector Spray Combustion
Parallel Computing, General Patched-Grid Method

15. NUMBER OF PAGES
94

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
unclassified	unclassified	unclassified	unlimited