UNIC-UNS USER’S MANUAL

A General Purpose CFD Code

Using Unstructured Mesh

Compatible Grid Formats: Gridgen/Fieldview, V-Grid, Patran and PLOT3D

ENGINEERING SCIENCES INCORPORATED
1900 Golf Road, Suite D
Huntsville, AL 35802, USA
(256) 883-6233 / www.unicfd.com
© Copyright since 2000. All rights reserved.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>AN INTEGRATED TOOL</td>
<td>2</td>
</tr>
<tr>
<td>SOLUTION METHOD</td>
<td>3</td>
</tr>
<tr>
<td>Governing Equations</td>
<td>3</td>
</tr>
<tr>
<td>Spatial Discretization</td>
<td>4</td>
</tr>
<tr>
<td>Time Integration</td>
<td>6</td>
</tr>
<tr>
<td>Pressure-Velocity-Density Coupling</td>
<td>6</td>
</tr>
<tr>
<td>Higher Order Schemes</td>
<td>7</td>
</tr>
<tr>
<td>Pressure Damping</td>
<td>9</td>
</tr>
<tr>
<td>Boundary Conditions</td>
<td>9</td>
</tr>
<tr>
<td>Linear Matrix Solver</td>
<td>10</td>
</tr>
<tr>
<td>Algebraic Multi-Grid Solver</td>
<td>10</td>
</tr>
<tr>
<td>Mathematical Formulations For Radiative Transfer Equation (RTE)</td>
<td>13</td>
</tr>
<tr>
<td>MESH ADAPTATION REFINEMENT</td>
<td>17</td>
</tr>
<tr>
<td>Identification of Cells for Refinement or Coarsening</td>
<td>17</td>
</tr>
<tr>
<td>Grid Refinement</td>
<td>19</td>
</tr>
<tr>
<td>Cell Coarsening</td>
<td>20</td>
</tr>
<tr>
<td>Smoothing Strategy</td>
<td>20</td>
</tr>
<tr>
<td>Parallelization</td>
<td>20</td>
</tr>
<tr>
<td>Domain Decomposition</td>
<td>21</td>
</tr>
<tr>
<td>Parallel Implementation</td>
<td>21</td>
</tr>
<tr>
<td>Implementation of Parallel Computing Algorithm</td>
<td>22</td>
</tr>
<tr>
<td>VALIDATIONS</td>
<td>23</td>
</tr>
<tr>
<td>Flow Solver Accuracy</td>
<td>23</td>
</tr>
<tr>
<td>Solution Adaptation Mesh Refinement</td>
<td>26</td>
</tr>
<tr>
<td>WALL FUNCTION HEAT TRANSFER MODELS</td>
<td>32</td>
</tr>
<tr>
<td>Hypersonic Heat Transfer Benchmark Test Case</td>
<td>32</td>
</tr>
<tr>
<td>UNIC-UNS INPUT GUIDE</td>
<td>35</td>
</tr>
<tr>
<td>DOMAIN DECOMPOSITION &amp; COMMUNICATION EXAMPLE</td>
<td>50</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>52</td>
</tr>
</tbody>
</table>
AN INTEGRATED TOOL

The procedure of predicting the general fluid dynamics environment involves the generation of computational mesh and the solutions of the complex flowfield that may involve radiative heat transfer effects. The integrated process is illustrated in Figure 1. The governing equations, numerical methodologies, radiative transfer model, mesh refinement, solution-adaptation, and parallel strategy are described in the following sections.

Fig. 1 Procedure for base-heating analysis
SOLUTION METHOD

Unstructured grid methods are characterized by their ease in handling completely unstructured meshes and have become widely used in computational fluid dynamics. The relative increase of computer memory and CPU time with unstructured grid methods is not trivial, but can be offset by using parallel techniques in which many processors are put together to work on the same problem. Furthermore, when unstructured grid method is facilitated with the multi-grid technique its convergence can be greatly enhanced to reach a converged solution. Moreover, flexible mesh adaptation is another attraction for the unstructured grid method where high resolution can be achieved by mesh refinement in high gradient regions such as shock waves.

Traditionally, numerical methods developed for compressible flow simulations use an unsteady form of the Navier-Stokes or Euler equations [1-6]. These methods use density as one of the primary variables and pressure is determined via an equation of state. Their application in cases of incompressible or low Mach number flows is questionable, since in low compressibility limit, the density changes are very small and the pressure-density coupling becomes very weak. Some possible ways to circumvent this difficulty lie in the use of a fictitious equation of state or artificial compressibility [7,8] such that the Jacobian matrix is not ill-conditioned. On the other hand, methods for incompressible flows are mostly of the pressure correction type and use pressure as the primary variable [9] for solving the continuity equation. These methods are well established, with many variations being possible depending upon the choice of the dependent variables and their arrangement, computational grid, pressure correction algorithm, differencing schemes, etc. [9,10].

Numerical methods, which are applicable to flows for all speed regimes are of special interest in the present research. An important step in this direction was reported by Hirt et al. [11]. Their method suffered from oscillations due to pressure-velocity decoupling. Recently, Karki and Patankar [12] and Chen [13] presented solution methods based on modified pressure-velocity coupling algorithms of the SIMPLE-type which include the compressibility effects and are applicable to flows at all speeds. Karki and Patankar [12] use locally fixed base vectors. Their method suffers from sensitivity to grid smoothness, due to presence (directly or indirectly) of curvature terms in the equations. The sensitivity is manifested by increased total pressure loss wherever the grid lines change their directions or surface areas.

In the present work, a cell-centered unstructured finite volume method was developed to predict the all speed flows [13-18], in which the primary variables are the Cartesian velocity components, pressure, total enthalpy, turbulence kinetic energy, turbulence dissipation and mass fractions of chemical species.

Governing Equations

The general form of mass conservation, Navier-Stokes equation, energy conservation and other transport equations can be written in Cartesian tensor form:
\[
\frac{\partial(\rho\phi)}{\partial t} + \frac{\partial}{\partial x_j}(\rho U_j\phi) = \frac{\partial}{\partial x_j}(\mu_s \frac{\partial \phi}{\partial x_j}) + S_s
\]  

(1)

where $\mu_s$ is an effective diffusion coefficient, $S_s$ denotes the source term, $\rho$ is the fluid density and $\phi = (1, u, v, w, h, k, \varepsilon)$ stands for the variables for the mass, momentum, total energy and turbulence equations, respectively. Detailed expressions for the $k-\varepsilon$ models and wall functions can be found in [19].

**Spatial Discretization**

The cell-centered scheme is employed here then the control volume surface can be represented by the cell surfaces and the coding structure can be much simplified. The transport equations can also be written in integral form as

\[
\frac{\partial}{\partial t} \int_{\Omega} \rho \phi d\Omega + \oint_{\Gamma} \vec{F} \cdot \hat{n} d\Gamma = \int_{\Omega} S_s d\Omega
\]  

(2)

where $\Omega$ is the domain of interest, $\Gamma$ the surrounding surface, $\hat{n}$ the unit normal in outward direction. The flux function $\vec{F}$ consists of the inviscid and the viscous parts:

\[
\vec{F} = \rho \vec{V} \phi - \mu_s \nabla \phi
\]  

(3)

The finite volume formulation of flux integral can be evaluated by the summation of the flux vectors over each face,

\[
\oint_{\Gamma} \vec{F} \cdot \hat{n} d\Gamma = \sum_{j \in k(i)} F_{i,j} \Delta \Gamma_j
\]  

(4)

where $k(i)$ is a list of faces of cell $i$, $F_{i,j}$ represents convection and diffusion fluxes through the interface between cell $i$ and $j$, $\Delta \Gamma_j$ is the cell-face area.

The viscous flux for the face $e$ between control volumes $P$ and $E$ as shown in Figure 2 can be approximated as:

\[
(\nabla \phi \cdot \vec{n})_e \approx \frac{\phi_e - \phi_p}{r_e - r_p} + \nabla \phi \cdot \left( \frac{\vec{n} - \vec{r}_e + \vec{r}_p}{r_e - r_p} \right)
\]  

(5)

That is based on the consideration that
\[ \phi_E - \phi_P \approx \nabla \phi \cdot (\vec{r}_E - \vec{r}_P) \]  \hspace{1cm} (6)

where \( \nabla \phi \) is interpolated from the neighbor cells E and P.

The inviscid flux is evaluated through the values at the upwind cell and a linear reconstruction procedure to achieve second order accuracy

\[ \phi_u = \phi_e + \Psi_e \nabla \phi_e \cdot (\vec{r}_e - \vec{r}_u) \]  \hspace{1cm} (7)

where the subscript u represents the upwind cell and \( \Psi_e \) is a flux limiter used to prevent from local extrema introduced by the data reconstruction. The flux limiter proposed by Barth [20] is employed in this work. Defining \( \phi_{\text{max}} = \max(\phi_e, \phi_u), \phi_{\text{min}} = \min(\phi_e, \phi_u) \), the scalar \( \Psi_e \) associated with the gradient at cell u due to edge e is

\[
\Psi_e = \begin{cases} 
\min(1, \frac{\phi_{\text{max}} - \phi_e}{\phi_e - \phi_u}) & \text{if } \phi_e - \phi > 0 \\
\min(1, \frac{\phi_{\text{min}} - \phi_e}{\phi_e - \phi_u}) & \text{if } \phi_e - \phi < 0 \\
1 & \text{other case}
\end{cases}
\]  \hspace{1cm} (8)

where \( \phi_e^0 \) is computed without the limiting condition (i.e. \( \Psi_e = 1 \))

![Figure 2. Unstructured control volume.](image-url)
Time Integration

A general implicit discretized time-marching scheme for the transport equations can be written as:

\[
\frac{(\rho^* + A_{\rho} \phi_{\rho}^{n+1})}{\Delta t} \sum_{m=1}^{NB} A_m \phi_{\rho, m}^{n+1} + \frac{(\rho \phi_{\rho})^*}{\Delta t} + S_\rho
\]

where NB means the neighbor cells of cell P. The high order differencing terms and cross diffusion terms are treated using known quantities and retained in the source term and updated explicitly.

The \( \Delta \)-form used for time-marching in this work can be written as:

\[
\frac{(\rho^n + A_{\rho}) \Delta \phi_{\rho}}{\Delta t} \sum_{m=1}^{NB} A_m \Delta \phi_{\rho, m} + SU_{\rho}
\]

where \( \theta \) is a time-marching control parameter which needs to specify. \( \theta = 1 \) and \( \theta = 0.5 \) are for implicit first-order Euler time-marching and second-order time-centered time-marching schemes. The above derivation is good for non-reacting flows. For general applications, a dual-time sub-iteration method is now used in UNIC-UNS for time-accurate time-marching computations.

Pressure-Velocity-Density Coupling

In an extended SIMPLE [12-18] family pressure-correction algorithm, the pressure correction equation for all-speed flow is formulated using the perturbed equation of state, momentum and continuity equations. The simplified formulation can be written as:

\[
\rho' = \frac{p'}{\gamma RT}; \quad \bar{u}' = -D_u \nabla p'; \quad \bar{u}^{n+1} = \bar{u}^n + \bar{u}'; \quad p^{n+1} = p^n + p'
\]

where \( Du \) is the pressure-velocity coupling coefficient. Substituting Eq. (12) into Eq. (13), the following all-speed pressure-correction equation is obtained,
\[
\frac{1}{\gamma RT} \rho' \Delta t + \nabla \cdot \left( \frac{\bar{u}}{\gamma RT} \rho' \right) - \nabla \cdot \left( \rho D_v \nabla \rho' \right) = -\left( \frac{\Delta \rho'}{\Delta t} \right) - \nabla \cdot \left( \rho \bar{u} \right)
\]  

(14)

For the cell-centered scheme, the flux integration is conducted along each face and its contribution is sent to the two cells on either side of the interface. Once the integration loop is performed along the face index, the discretization of the governing equations is completed. First, the momentum equation (9) is solved implicitly at the predictor step. Once the solution of pressure-correction equation (14) is obtained, the velocity, pressure and density fields are updated using Eq. (12). The entire corrector step is repeated 2 or 3 times so that the mass conservation is enforced. The scalar equations such as turbulence transport equations, species equations etc. are then solved sequentially. Then, the solution procedure marches to the next time level for transient calculations or global iteration for steady-state calculations. Unlike for incompressible flow, the pressure-correction equation, which contains both convective and diffusive terms is essentially transport-like. All treatments for inviscid and the viscous fluxes described above are applied to the corresponding parts in Eq. (14).

**Higher Order Schemes**

The challenge in constructing an effective higher-order scheme is to determine an accurate estimate of flux at the cell faces. Barth and Jespersen [21] proposed a multi-dimensional linear reconstruction approach, which forms the basis for the present scheme. In the cell reconstruction approach, higher-order accuracy is achieved by expanding the cell-centered solution to each cell face with a Taylor series:

\[
q(x, y, z) = q(x_c, y_c, z_c) + \nabla q_c \cdot \Delta \vec{r} + O(\Delta \vec{r}^2)
\]  

(15)

where

\[
q \equiv [\rho, u, v, w, t, dk, de]^T
\]  

(16)

This formulation requires that the solution gradient be known at the cell centers. Here a scheme proposed in [22] is employed to compute the gradients:

\[
\int q \nabla d\Omega = \oint q \vec{n} d\Gamma
\]  

(17)

The general approach was to: 1) coalesce surrounding cell information to the vertices or nodes of the candidate cell, then 2) apply the midpoint-trapezoidal rule to evaluate the surface integral of the gradient theorem

\[
\nabla q = \frac{1}{\Omega} \oint q \vec{n} d\Gamma
\]  

(18)
over the faces of each cells. Here \( \Omega \) denotes the volume enclosed by the surface \( \Gamma \).

It is possible to further simplify the method for triangle (2D) or tetrahedron (3D) cells such that Eq. (18) need not be evaluated explicitly. The simplification stems from the useful geometrical invariant features of triangle and tetrahedron. These features are illustrated for an arbitrary tetrahedral cell in Figure 3. Note that a line extending from a cell-vertex through the cell-centroid will always intersect the centroid of the opposing face. Furthermore, the distance from the cell-vertex to the cell centroid is always three-fourths of that from the vertex to the opposing face (For a triangle, the comparable ration of distance is two-thirds). By using these invariants along with the fact that \( \Delta r \) is the distance between them, Eq. (16) can be evaluated as:

\[
\nabla q_c \cdot \Delta \vec{r} = \frac{\partial q_c}{\partial r} \Delta r \approx \frac{1}{3} (q_{n1} + q_{n2} + q_{n3}) - q_{n4} \]

Thus Eq. (15) can be approximated for tetrahedral cells by the simple formula:

\[
q_{n1,2,3} = q_c + \frac{1}{4} \left[ \frac{1}{3} (q_{n1} + q_{n2} + q_{n3}) - q_{n4} \right]
\]

where as illustrated in Figure 3, the subscripts \( n_1, n_2 \) and \( n_3 \) denote the nodes comprising face \( f_{1,2,3} \) of cell \( c \) and \( n_4 \) corresponds to the opposite node. This modified scheme is analytically equivalent to that in [22], and results in a factor of two reductions in computational time of the flow solver.

The nodal quantities \( q_n \) are determined in the manner described in [22]. Accordingly, estimates of the solution are determined at each node by a weighted average of the surrounding cell-centered solution quantities. It is assumed in the nodal averaging procedure that the known values of the solution are concentrated at the cell centers, and that the contribution to a node from the surrounding cells is inversely proportional to the distance from each cell centroid to the node:

\[
q_c = \frac{\sum_{i=1}^{N} q_{ci} r_i}{\sum_{i=1}^{N} r_i}
\]

where

\[
r_i = \left[ (x_{c,i} - x_n)^2 + (y_{c,i} - y_n)^2 + (z_{c,i} - z_n)^2 \right]^{1/2}
\]

The nodal quantities \( q_n \) are determined in the manner described in [22]. Accordingly, estimates of the solution are determined at each node by a weighted average of the surrounding cell-centered solution quantities. It is assumed in the nodal averaging procedure that the known values of the solution are concentrated at the cell centers, and that the contribution to a node from the surrounding cells is inversely proportional to the distance from each cell centroid to the node:
Until recently, the sole approach for the reconstruction was a pseudo-Laplacian averaging scheme presented in [23]. This scheme offers the advantage of second-order accuracy in reconstructing data from surrounding cells to a node. However, there is a need to artificially 'clip' the weighting factors between 0 and 2 [24] to avert a violation of the positivity principle, which is necessary for solution stability. This artificial 'clipping' process does, unfortunately, compromise the formal second-order accuracy of the scheme to some extent. Recent experience of applying the pseudo-Lapacian scheme to Navier-Stokes computations has surfaced some anomalous behavior, which needs further investigation. Meanwhile, for the present work, we are temporarily reverting to the inverse-distance averaging of Eq. (21), which represents only acceptable accuracy, but will never violate the principle of positivity.

**Pressure Damping**

Following the concept of Rhie and Chow [10] developed for structured grid method to avoid the even-odd decoupling of velocity and pressure fields, a pressure damping term can also introduced for unstructured grid method when evaluating the interface mass flux. This form is written as:

$$U_{i} = \bar{U}_{i} - \bar{D}U_{i} \cdot \left( \frac{p_{c} - p_{p}}{r_{c} - r_{p}} - \bar{\nabla}p_{r} \cdot \frac{\bar{r}_{c} - \bar{r}_{p}}{|\bar{r}_{c} - \bar{r}_{p}|} \right)$$

where $\bar{U}_{i}$, $\bar{\nabla}p_{r}$ and $\bar{D}U_{i}$ are interpolated from the neighboring cells E and P, respectively. The last term on the right hand side is a higher order pressure damping term projected in the direction $\bar{PE}$.

**Boundary Conditions**

Several different types of boundaries may be encountered in flow calculations, such as inflow, outflow, impermeable wall and symmetry. In the case of viscous incompressible flows, the following boundary conditions usually apply:
• The velocities and temperature are prescribed at the inlet;
• Zero normal gradient for the parallel velocity component and for all scalar quantities, and zero normal velocity component are specified at symmetry planes or axes;
• No-slip condition and prescribed temperature or heat flux are specified at the walls;
• Zero (or constant non-zero) gradient of all variables is specified at the outlet.

In the case of compressible flows, some new boundary conditions may apply:

1. Prescribed total conditions (pressure, temperature) and flow direction at inflow;
2. Prescribed static pressure at outflow;
3. Supersonic outflow.

Some of these boundary conditions are straightforward to implement the detailed implementation is described in [13].

In order to facilitate the variations of inlet operating conditions in rocket engine applications, an inlet data mapping tool is developed. This inlet data-mapping tool accepts multiple patches of surface data. Each individual surface data patch is a structured grid that contains flowfield data such as velocity, pressure, temperature, turbulence quantities and species concentrations, etc. At start up, the UNIC code reads input data file and restart flowfield data, then it will check the existence of the inlet boundary condition file, bcdata.dat. If inlet data is obtained from the inlet boundary condition file, data mapping procedure is then performed to incorporate the new inlet conditions.

**Linear Matrix Solver**

The discretized finite-volume equations can be represented by a set of linear algebra equations, which are non-symmetric matrix system with arbitrary sparsity patterns. Due to the diagonal dominant for the matrixes of the transport equations, they can converge even through the classical iterative methods. However, the coefficient matrix for the pressure-correction equation may be ill conditioned and the classical iterative methods may break down or converge slowly. Because satisfaction of the continuity equation is of crucial importance to guarantee the overall convergence, most of the computing time in fluid flow calculation is spent on solving the pressure-correction equation by which the continuity-satisfying flow field is enforced. Therefore the preconditioned Bi-CGSTAB [25] and GMRES [26] matrix solvers are used to efficiently solve, respectively, transports equation and pressure-correction equation.

**Algebraic Multi-Grid Solver**

An algebraic multi-grid method (AMG) is also developed for the solution of the algebra equation resulting from the pressure-correction equation. The AMG method is often used to low speed flows with long domains, which are subject to long-wave errors.
The algorithm of the AMG method is outlined below. The final linear equation has the form of

$$a_i \phi_i - \sum_{\text{nb of } i} a_i^{nb} \phi_{nb} = b_i$$  \hspace{1cm} (24)

where nb means the neighbor cell of i. Assume the solution on coarse mesh cell $i$ which contains the fine mesh cell $i$ is $\Phi_{I/i}$. The correction on fine mesh from coarse mesh is

$$\tilde{\phi}_i = \phi_i + \Phi_{I/i}$$  \hspace{1cm} (25)

In order to enforce the residual sum in $I$ be zero,

$$\sum_{i \text{ in } I} \tilde{r}_i = 0$$  \hspace{1cm} (26)

The equation on coarse mesh comes out as

$$\sum_{i \text{ in } I} a_i \Phi_{I/i} - \sum_{\text{nb of } i} \sum_{i \text{ in } I} a_i^{nb} \Phi_{NB/nb} = \sum_{i \text{ in } I} \tilde{r}_i$$  \hspace{1cm} (27)

Rewrite the equation on coarse mesh in the form of

$$A_I \Phi_I - \sum_{\text{NB of } I} A_I^{NB} \Phi_{NB} = B_I$$  \hspace{1cm} (28)

Then the $A_I$, $A_I^{NB}$ and $B_I$ come directly from Equation (27) as

$$A_I = \sum_{i \text{ in } I} a_i - \sum_{i \text{ in } I} a_i^{nb}$$  \hspace{1cm} (29)

$$A_I^{NB} = \sum_{i \text{ in } I} a_i^{nb}$$  \hspace{1cm} (30)

$$B_I = \sum_{i \text{ in } I} \tilde{r}_i$$  \hspace{1cm} (31)

The algebraic multi-grid (AMG) method contains restriction and prolongation processes. In the restriction process, the equation coefficients and source terms on coarse mesh are generated based on those on fine mesh from Equations (29-31). While in the prolongation process, variable on fine mesh is modified by that on coarse mesh
according to Equation (25). It can be seen that there is no actual coarse grid mesh in algebraic multi-grid method, but only the coarse grid equations followed completely from the fine grid equations. This solver strongly based on physical conservative concept rather than pure mathematical property. It requires the governing equation on fine grid to be in conservation form.

The coarsening process is merging the adjacent cells to form larger finite volume with the limitation on minimum and maximum number of cells to group together. The merging cell picked is the one that has strong connection with the cells in the group. The neighbor cell $n$ is considered as strongly connected with cell $i$ if

$$ a_i^n > c \cdot a_i^{\text{max}} $$

where $a_i^{\text{max}}$ presents the largest neighboring coefficient among $a_i^{nb}$, $c$ is a constant and the value of 1/3 works well. The minimum and maximum fine cell number in one coarse volume is set to 5 and 9 for 2-D, and 9 and 13 for 3-D, respectively.

There are two types of fixed cycle multi-grid sequences, V cycle and W cycle. W cycle is more efficient because each grid level has the chance to pass its residual down to the coarse grid level twice and receive the corrections twice. The paths of V cycle and W cycle are shown in Figure 4, where the restriction process goes down at “d”, reach the bottom at “b” and then the prolongation process goes up at “u”. In this research, W cycle is used and the coarse grid level is set to no more than six. The bottom level is reached when the mesh cell number is less than 100 or the level is number 6 whichever comes first. The Incomplete Lower Upper (ILU) factorization scheme is used to solve the linear equations. It takes two sweeps in the down processes and 3 sweeps in the up processes. At the bottom, GMRES method is used to ensure the accurate solution at the coarsest grid level. Several cycles may be needed to reduce the residual by two orders of magnitude in each time step. For all the calculation in this study, the cycle number is less than 10 and mostly around 5.

![Figure 4. AMG solver V and W cycle diagram.](image-url)
Mathematical Formulations For Radiative Transfer Equation (RTE)

Consider the RTE in a Cartesian coordinate system as shown in Fig. 5a. The balance of energy passing in a specified direction $\Omega$ through a small differential volume in an absorbing-emitting and scattering medium can be written as:

$$
(\Omega \cdot \nabla) I_\lambda (r, \Omega) = - (\kappa_\lambda + \sigma_\lambda) I_\lambda (r, \Omega) + \kappa_\lambda I_{b,\lambda} (r) + \frac{\sigma_\lambda}{4\pi} \int_{\Omega'} I_\lambda (r, \Omega') \Phi_\lambda (\Omega' \rightarrow \Omega) d' (33)
$$

where the subscript $\lambda$ represents the wave-number; $I_\lambda (r, \Omega)$ is the spectral radiative intensity, which is a function of position and direction; $I_{b,\lambda} (r)$ is the blackbody radiative intensity at the temperature of the medium; $\kappa_\lambda$ and $\sigma_\lambda$ are the spectral absorption and scattering coefficients, respectively; and $\Phi_\lambda (\Omega' \rightarrow \Omega)$ is the scattering phase function from the incoming $\Omega'$ direction to the outgoing direction $\Omega$. The term on the left hand side represents the gradient of the intensity in the direction $\Omega$. The three terms on the right hand side represent the changes in intensity due to absorption and out-scattering, emission and in-scattering, respectively.

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

$$
I_\lambda (r_w, \Omega^+) = \varepsilon_\lambda I_{b,\lambda} (r_w) + \frac{1 - \varepsilon_\lambda}{\pi} \int_{\Omega} I_\lambda (r_w, \Omega^-) |n \cdot \Omega^-| d'
$$

(34)

where $\Omega^+$ and $\Omega^-$ denote the leaving and arriving radiative intensity directions, respectively; $\varepsilon_\lambda$ is the spectral wall emissivity; $n$ represents the unit normal vector on the wall.

Equation (33) is a complex integro-differential equation whose exact analytical solution is only possible for very simple and specific settings. This intrinsic difficulty has resulted in the development of several approximated models. In this project, the finite volume method (FVM) is used to solve the RTE and their numerical analysis procedures are briefly described here. In the following analysis, subscripts $\lambda$ are dropped for the sake of brevity and they are added whenever necessary.

In the FVM, the spatial and angular domains are divided into a finite number of control volumes and control angles, respectively. Then, Eqs. (33) and (34) are integrated over each control volume and control angle. Since the FVM shares the same computational grid as the CFD approach, the considered spatial domain is divided into MA control volumes and surfaces by a grid generator. For numerical analysis of the FVM, a representative control volume resulted from the spatial domain division is shown in Fig. 5b. By referring to the division practice for the spatial domain, the angular domain (see Fig. 5c) at a node centered in a control volume is divided into $N_\theta \times N_\phi = MB$ control
angles with $N_\theta$ and $N_\phi$ representing numbers of control angle in polar angle $\theta$ and azimuthal angle $\phi$ directions, respectively. These MB discrete solid angles are nonoverlapping and their sum is $4\pi$. Unlike the selection of a quadrature scheme in the discrete ordinates method (DOM), there is no specific restriction in selecting control angles in the FVM. However, the control angles are usually chosen in a manner that best capture the physics of a given problem. This is analogous to the selection of control volumes.

Figure 5. (a) Coordinate system for radiative transfer equation, (b) a representative Control-volume, and (c) a representative control angle.

Multiplying Eq. (33) by a representative control volume $\Delta V$ (Fig. 5b) and a control angle $\Delta \Omega^m$ (Fig. 5c), carrying out the integration, and transforming the left-hand side of the equation from the volume integral to a surface integral by the divergence theorem, Eq. (33) then becomes

$$\sum \int_I I_{i} A_i \int_{\Delta \Omega^m} (\Omega^m \cdot n_i) d\Omega^m - \sum \int_I I_{i} A_i \int_{\Delta \Omega^m} (\Omega^m \cdot n_i) d\Omega^m$$

$$=\left[-(\kappa + \sigma)I + \kappa \right]_P + \sum_{m=1}^{M} \Delta \Omega^m \Phi^m I^m \Delta V \Delta \Omega^m$$

where the subscripts e, w, etc. indicate the values on the eastern, western, etc. surfaces of the volume; The subscript P represents the value at the central node of the control volume and A represents the control volume surface area; $\Phi^m$ is the averaged scattering phase function from the control angle $\Delta \Omega^m$ to the control angle $\Delta \Omega^m$. Dividing Eq. (35) by $\Delta \Omega^m$, we have
\[
\sum_{i=a,b,c} I_{m}^{n} A_{i}\overrightarrow{D}_{i} - \sum_{i=a,b,c} I_{m}^{n} A_{i}\overrightarrow{D}_{i} = \left[-\left(\kappa + \sigma\right)I_{m}^{n} + \kappa I_{b} + \frac{\sigma}{4\pi} \sum_{n=1}^{M} \Delta \Omega_{n}^{m} \Phi_{n}^{m} I_{n}^{m}\right]_{p} \Delta V \quad (36)
\]

where
\[
\overrightarrow{D}_{i} = n_{i} \cdot \overrightarrow{\Omega}_{i} = \frac{1}{\Delta \Omega_{i}} \int (\Omega_{i} \cdot n_{i}) d\Omega_{i} \quad (37)
\]

In Eq. (37), \( \overrightarrow{D}_{i}^{m} \) is the product of a surface unit normal vector and the averaged intensity direction \( \overrightarrow{\Omega}^{m} \).

To close the above equation, relations are needed between the intensities on the control volume surfaces and the nodal intensities. One appropriate closure relation for complicated geometries is based on the step scheme, which sets the downstream surface intensities equal to the upstream nodal intensities. Use of the step scheme avoids the negative intensities, overshoots, and undershoots which may occur in other radiation schemes such as diamond scheme, positive scheme, etc. Furthermore it has much less connection with the neighboring nodes, and thus it is particularly suitable for parallel computation. The effect of the step scheme on communication costs is not significant. The final discretized equation for the FVM can be written as
\[
a_{m}^{I_{p}} I_{n} = a_{E}^{I_{p}} I_{E} + a_{W}^{I_{p}} I_{W} + a_{N}^{I_{p}} I_{N} + a_{S}^{I_{p}} I_{S} + a_{T}^{I_{p}} I_{T} + a_{B}^{I_{p}} I_{B} + b^{m} \quad (38)
\]

where the intensities with the subscripts E, W, ... denote the eastern, western, etc. nodal intensities, and
\[
a_{p}^{m} = \sum_{i=a,b,c} \max(A_{i}\overrightarrow{D}_{i}, 0) - \sum_{i=a,b,c} \min(A_{i}\overrightarrow{D}_{i}, 0) + \Delta V(\kappa + \sigma) \quad (39a)
\]
\[
a_{i}^{m} = -\min(A_{i}\overrightarrow{D}_{i}, 0) i = e,n,t and I = E,N,T \quad (39b)
\]
\[
a_{i}^{m} = \max(A_{i}\overrightarrow{D}_{i}, 0) i = w,s,b and I = W,S,B \quad (39c)
\]
\[
b^{m} = \Delta V[\kappa I_{b} + \frac{\sigma}{4\pi} \sum_{n=1}^{M} \Delta \Omega_{n}^{m} \Phi_{n}^{m} I_{n}^{m}]_{p} \quad (39d)
\]

The preceding discretization is carried out along only one control angle at a node. The same procedure should be applied to all of the MB control angles at all of the MA nodes. This forms MA×MB systems of non-symmetric algebraic equations. A solution of these equations only represents radiative contribution at a single wave-number. The radiative divergence is the quantity used in the energy equation and it should consist of radiative contributions from all wave-numbers. The radiative divergence is expressed in terms of the radiative intensities as
\[
\nabla \cdot q_{r} = \int_{0}^{4\pi} \int_{\lambda_{b}}^{\lambda_{a}} 4\pi \kappa_{\lambda} I_{\lambda,b,\lambda} d\omega d\lambda = \int_{0}^{4\pi} \int_{\lambda_{b}}^{\lambda_{a}} 4\pi \kappa_{\lambda} I_{\lambda,b,\lambda} d\omega d\lambda = \int_{0}^{\infty} \left[4\pi \kappa_{\lambda} I_{\lambda,b,\lambda} - \kappa_{\lambda} \sum_{m=1}^{MB} I_{\lambda}^{m} \Delta \Omega_{m}\right] d\lambda \quad (40)
\]
A typical radiatively participating gas consists of many lines whose absorption coefficients vary rapidly with wave number. Thus, it becomes a very difficult and time-consuming to evaluate the radiative properties over the actual band contour and include them into the RTE. To avoid this difficulty, the spectrum can be divided into MC bands and the radiative properties are assumed constant over each band. The integrated quantity in Eq. (40) is found as the summation over all bands of the individual contribution for each band, that is

\[
\nabla \cdot q_r = \sum_{j=1}^{MC} [4\pi \kappa_j I_{bj} - \kappa_j \sum_{m=1}^{MB} I_j^m \Delta \Omega^m] \Delta \lambda_j
\]

The above approach to the spectral problem essentially corresponds to the spectral discretization and it represents a good compromise between accuracy and computational time. The number of MC can change from one (gray gas model) to several hundred (narrow band model). Obviously, the use of higher MC number provides more accurate results.
MESH ADAPTATION REFINEMENT

Unstructured meshes are particularly flexible for local grid refinement and coarsening. The fundamental idea behind adaptation is to modify the grids to better resolve the features in the flow fields and hence to achieve accurate numerical solutions. In this study, the modification of the grids is carried out by a hanging node approach. In this approach [18], a new node in the grid refinement process is produced by subdividing an edge. This new node is a hanging node if it is not a vertex of all the cells sharing that edge. The present face based flux evaluation method provides an ideal environment for implementing a hanging node adaptation scheme - solver simply visits each of new child faces instead of original parent face. In addition, hanging node grid adaptation provides the ability to efficiently operate on grids with a variety of cell shapes, including hybrid grids. The basic structure of an adaptive solution procedure consists of: (a) solving the governing equations on the current grid; (b) identifying cells for refinement or coarsening; (c) subdividing the cells identified for refinement; (d) coalescing the cells identified for coarsening; and (e) refining additional cells to maintain a smooth grid density variation required for solver to guarantee the stability and accuracy.

Identification of Cells for Refinement or Coarsening

The grid adaptation requires a formulation to detect and locate features of interest of the flow field, which is generally called as adaptation function. An error indicator or sensor is used in the adaptation criterion to identify regions of higher solution gradients, and it is the key to the success of grid adaptation. Such indicator can be specified based on the physics of the solution fields. It is very important for the error indicator to detect a variety of flow features yet sensitive enough to detect weak features. Therefore, the formulation of the error indicator must be flexible to identify the flow features based on density, pressure, velocity magnitude, vorticity, turbulence kinetic energy or their combinations.

By assuming $\phi$ as a suitable flow property, the adaptation parameter at cell $i$ is evaluated by [18]

$$E_i = \max_{j \in \partial i} \left| \phi_i - \phi_j \right|$$

(42)

where $j$ represents the cells adjacent to cell $i$. The standard deviation of error indicator is then calculated as [18]:

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^{N} E_i^2}$$

(43)
The summation is only conducted for cells, which satisfy $E_i > \varepsilon E_{\text{max}}$, therefore the undisturbed fields are not included to determine the standard deviation, where $\varepsilon$ is a specified small value and is equal to 0.05 in the present study.

To better resolve the multiple flow field features, a combined adaptation function can be obtained by summation each normalized error indicator ($E_i/\sigma_i$) multiplied by a weighting factor ($\omega_i$) as follows [18]:

$$A_i = \sum \omega_i E_i/\sigma_i$$ (44)

Then, the minimum and maximum thresholds for adaptation function can be assigned, and cells are subsequently marked for refinement or coarsening based on the value of adaptation function whether the cells fall into the specified region or not.

The effectiveness of adaptive schemes, as it has been pointed out in the open literature, depends very much on the selection of flowfield functions, from which gradients are linked to the strength for mesh refinement. Naturally, this selection again depends on what types of flow problems are under investigation. For low-speed (or incompressible) flows, what interest the researcher or the engineering designer most would be the resolutions of the boundary layers and/or the shear layers. Conversely, for high-speed flows, we would like to also include density, pressure and temperature in the formulations of the function to be used to guide the mesh adaptation. Presently, seven functions are provided as options selectable from the input data file, unic.inp, of the UNIC-UNS code. These functions can be selected by setting the adaptive-function parameter, IADPF, as given in the following:

<table>
<thead>
<tr>
<th>IADPF</th>
<th>Flow Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Combination of Density, Pressure and Velocity Magnitude</td>
</tr>
<tr>
<td>2</td>
<td>Density</td>
</tr>
<tr>
<td>3</td>
<td>Pressure</td>
</tr>
<tr>
<td>4</td>
<td>Temperature</td>
</tr>
<tr>
<td>5</td>
<td>Combination of Density and Mach Number</td>
</tr>
<tr>
<td>6</td>
<td>Turbulence Kinetic Energy</td>
</tr>
<tr>
<td>7</td>
<td>Vorticity Magnitude</td>
</tr>
</tbody>
</table>

These functions have been tested and their detailed formulation may need further adjustment for different types of flow problems to reveal their effectiveness. For rocket engine performance and base heating problems, the calculated thrust and heat fluxes are used to guide the sensitivities of the functions.
Grid Refinement

The cells are refined by either isotropically or non-isotropically subdividing each cell identified for refinement. The sub-division of the supported cell shapes is described below (see Figure 6):

- A triangle is split into 4 triangles;
- A quadrilateral is split into two or four quadrilaterals according to anisotropic refinement functionality;
- A tetrahedron is split into eight tetrahedra. The subdivision consists of trimming each corner of the tetrahedron, and then subdividing the enclosed octahedron by introducing the shortest diagonal;
- A hexahedron is split into 2, 4 or 8 hexahedra in terms of anisotropic refinement functionality;
- A prism is split into 8 prisms;
- A pyramid is split into 6 pyramids and 4 tetrahedra.

The basic steps of the subdivision process include orienting the geometric entities (nodes, edges and faces) of cell marked for subdivision in a consistent way. To maintain accuracy, neighboring cells are not allowed to differ by more than one level of refinement. This prevents the adaptation from producing excessive cell volume changes and ensures the positions of the parent and child cell centroids are similar, thus keeping the accuracy of the flux evaluations.

Figure 6. Cell sub-dividing strategy in the grid refinement method.
Cell Coarsening

The mesh is coarsened by reintroducing inactive parent cells. This process is equivalent to coalescing the child cells of the previously subdivided parent cell. An inactive parent cell is recovered if all its children are marked for coarsening. The basic procedure is to visit each inactive cell, locate its children using the face sibling information, remove the appropriate nodes, edges, faces and cells, reintroduce the parent cell with its associated entities into the active data structure, and assign the cell variables to the parent using the volume weighted averaging of children’s cell variables. The original grid is ultimately reclaimed. This is a particularly attractive feature for transient computations, especially transient solutions with limiting cycles that the maximum cell number is bounded.

Smoothing Strategy

To assure a smooth variation of cell volume, additional cells are refined based on the number and/or relative position of neighboring cells that have been subdivided. A triangle cell is refined if it has more than one refined neighboring cells. A quadrilateral cell is refined if it has two refined opposing cells or more than two refined neighboring cells. A tetrahedral cell is refined if it has more than two refined neighboring cells. A hexahedral cell is refined if it has two refined opposing cells, or more than three refined neighboring cells. And finally a pyramid or prism cell is refined if it has more than two refined neighboring cells. The smoothing process is effective that a cell is divided mandatorily due to the refining state of its neighboring cells, even though its value of adaptation function is less than the value specified. When a cell is coarsened, the same rule is followed to ensure that no excessive cell volume variations occur.

Parallelization

Compared with a structured grid approach, the unstructured grid algorithm is more memory and CPU intensive because “links” between nodes, faces, cells, needs to be established explicitly, and many efficient solution methods developed for structured grids such as approximate factorization, line relaxation, SIS, etc. cannot be used for unstructured methods.

As a result, numerical simulation of three-dimensional flow fields remains very expensive even with today’s high-speed computers. As it is becoming more and more difficult to increase the speed and storage of conventional supercomputers, a parallel architecture wherein many processors are put together to work on the same problem seems to be the only alternative. In theory, the power of parallel computing is unlimited. It is reasonable to claim that parallel computing can provide the ultimate throughput for large-scale scientific and engineering applications. It has been demonstrated that performance that rivals or even surpasses supercomputers can be achieved on parallel computers.
Domain Decomposition

To implement a parallel CFD computation strategy, the computational domain needs to be partitioned into many sub-domains. Each sub-domain then occupies one processor of a parallel computer. Many partitioning algorithms have been developed to partition an unstructured grid. These algorithms include Recursive Coordinate Bisection (RCB), Recursive Spectral Bisection (RSB), and Recursive Graph Bisection (RGB) methods. A publicly available package developed at University of Minnesota, METIS [28], can partition high quality unstructured meshes efficiently. Grids with 1 million vertices can be partitioned in 256 parts in less than 20 second on a Pentium Pro personal computer. In this work, both of RCB and RGB are employed and implemented in the solver.

In performing parallel computation, domain decomposition method is currently used in the present code. By default, METIS’ recursive graph bisection (RGB) method is employed for efficient decomposition. In some cases (e.g. internal flow problems with long domain), however, the recursive coordinate bisection (RCB) method seems to provide better overall convergence for the flowfield solutions. To make these two methods readily accessible in the present CFD code, an input control parameter, IPART, is introduced to control the selection of domain decomposition methods. The RCB method is also activated as a built-in option for domain decomposition so that it can be used when the METIS library is not available on a computer system. The selections are shown below.

<table>
<thead>
<tr>
<th>IPART</th>
<th>Decomposition Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Built-in Recursive Coordinate Bisection (RCB)</td>
</tr>
<tr>
<td>2</td>
<td>METIS’ Recursive Graph Bisection (RGB)</td>
</tr>
</tbody>
</table>

When the IPART = 2 option is selected, the METIS module must be executed to generate the RGB database before running the flow solver in parallel mode.

Parallel Implementation

In a parallel computation, the governing equations are solved in all sub-domains, which are assigned to different computers’ processor [29]. Exchange of data between processors is necessary to enforce the boundary conditions at the divided interfaces. The communication overhead must be kept well below the computational time. Currently, many communication software packages, such as PVM and MPI, have been developed for distributed computing.

The Parallel Virtual Machine (PVM) software system [30] is 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 massively parallel processors (MPPs). It 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. PVM is implemented for data communication among processors in this work.

MPI stands for Message Passing Interface [31]. The goal of MPI, simply stated, is to develop a widely used standard for writing message-passing programs. As such, the interface attempts to establish a practical, portable, efficient, and flexible standard for message passing. The main advantages of establishing a message-passing standard are portability and ease-of-use. In a distributed memory communication environment in which the higher level routines and/or abstractions are built upon lower level message passing routines the benefits of standardization are clear. Furthermore, the definition of a message passing standard provides vendors with a clearly defined base set of routines that they can implement efficiently or in some cases provide hardware support for, thereby enhancing scalability. The MPI implementation of the present CFD program is described in Appendix A.

Implementation of Parallel Computing Algorithm

Generally, there are two schemes for parallelization, explicit and implicit. Explicit schemes are relatively easy to parallelized, since all operations are performed on data from preceding time steps. It is only necessary to exchange the data at the interface regions between neighboring sub-domains after each step is completed. The sequence of operations and results are identical on one and many processors. The most difficult part of the problem is usually the solution of the elliptic Poisson-like equation for the pressure or pressure-correction equation. Implicit methods are more difficult to parallelize. While calculation of the coefficient matrix and source vectors uses only ‘old’ data and can be efficiently performed in parallel, solution of the linear equation systems needs special attention to parallelize.

With the explicit block coupling in parallel computing, the solution may not converge in some cases. Thus in this solver the implicit block coupling is implemented in the AMG method, which greatly enhance the stability and capability of the solver to handle complex problems.
VALIDATIONS

To validate the accuracy of the methodologies used in the present flow solver and demonstrate its capabilities, several representative test cases for inviscid and viscous flow are presented below.

Flow Solver Accuracy

Flows over an arc bump are studied. Different types of flow cases (subsonic, transonic and supersonic) in a channel with a circular arc bump were chosen as validation examples for inviscid flow calculations. The width of the channel is the same as the length of the bump and the total length of the channel is three times the axial length of the bump.

First, a subsonic flow case was considered. At the inlet, it is assumed that the flow has uniform properties and the upstream far-field variable values (except for pressure) are specified. At the outlet, all variables are extrapolated except for pressure, which is prescribed. At the upper and lower wall, the flow tangency and zero mass flux through the boundary are prescribed. Fig. 7 shows the pressure contours for a given Mach number at the inlet, $M_{in} = 0.5$. This result compares favorably with those found in other publication [12].

Next, a transonic flow case was considered. The mesh and the treatment of boundary conditions are identical to those described for subsonic flow. For the given inlet Mach number $M_{in} = 0.675$, the pressure contours are shown in Fig. 8. The normal shock location and strength predicted agree well with the results in Ref. 12.

Finally, a supersonic flow was analyzed. The inlet Mach number is uniform and equals to 1.65. The flow is also supersonic at the outlet. Thus, all variables are prescribed at inlet and are extrapolated at the outlet. Fig. 9 shows the shock pattern predicted of this case. The presentation solution also compares well with the results given in Ref. 12.

Figure 7. Pressure contours with subsonic flow.
Next, low speed flows past a circular cylinder is considered. The flow feature of this case depends on the Reynolds numbers. For Reynolds numbers below 50, steady wake flow has been observed experimentally. When the Reynolds number increased beyond 100, unsteady wake flow with cyclic vortex shedding patterns emerges. In the present study, cases with Reynolds numbers of 40 and 300 were considered. Figure 10 shows the streamline plot of the predicted wake flow at Reynolds number 40. The predicted wake length (defined as the length of the recirculation zone behind the cylinder) is 2.22 times the cylinder diameter, which compares well with the measured data of 2.21. For the second case with Reynolds number 300, the wake pattern is initially symmetric for some period of time. The bifurcation started later, due to the accumulated numerical noise, with growing amplitude in oscillations, and finally reaches a periodic vortex shedding pattern as shown in Figure 11. The predicted Strouhal number (St) is 0.205, which is in good agreement with the measured value of 0.2. The predicted mean drag coefficient is 1.385, which compares well with other predictions.
Figure 11. Unsteady wake flow behind a circular cylinder at Reynolds number 300.
Solution Adaptation Mesh Refinement

To demonstrate the capabilities of the adaptation with the solver, several cases with different flow conditions and geometries were investigated and their results are presented below.

The first test case is an incompressible laminar flow past three circular cylinders. The flow Reynolds number 40 (based on the free stream velocity and the diameter of the cylinder) is used and the flow is believed to be stable at this flow condition. The initial hybrid unstructured mesh is shown in Fig. 12. Quadrilateral cells are generated at the vicinity of the cylinders to better resolve the flow boundary layers and achieve high grid quality.

The initial solution was obtained from the initial mesh when it was converged. Then the level-2 adaptive mesh was generated based on the current solution and is shown in Fig. 13. It can be observed that the grids closed to cylinders and at the wake region are enriched. The level-3 mesh adaptation is achieved based on the solution of the level-2 adaptation mesh, and is illustrated in Fig. 14. The final solutions of velocity contours and vectors are presented in Figs. 15 and 16. The flow fields are symmetric about the centerline due to the symmetric geometry and stable flow conditions. The results obtained indicate the ability to achieve higher resolution flow fields through automatic mesh adaptation.

An inviscid transonic (M = 0.799) flow around a NACA 0012 profile at an angle of attack (2.27 degree) is considered in the second test case. Fig. 17 illustrates a series of adaptive meshes and the corresponding solutions from the beginning to the end of an adaptation loop. In all the following cases, the color of meshes represents the contours of pressure. Comparing the pressure contours between the initial grid (Fig. 17(a)) and the final adapted grid (Fig. 17(b)), one can see a dramatic improvement in the resolution of the shock wave on the upper surface of the airfoil.

The third test case is a supersonic flow over a bump as described in previous validation cases. Here, we would like to look at the effect of mesh refinement on solution improvements. Fig. 18 illustrates the effects of adaptive meshes from the initial to the end of an adaptation loop. The numerical solution on the initial mesh as shown in Fig. 18(a) is very diffusive and less accurate at capturing the complicated interactions of shock-to-shock, and shock-to-boundary. However, adapting the grid dramatically improves the accuracy of the numerical simulation as shown in Fig. 18(b). It can also be observed that two oblique shocks are formed at both corners of the bump. The leading edge shock reflects from the top wall and intersects with the shock leaving the trailing edge. It is interesting to note that the shock position changes only slightly with grid refinement. Both solutions satsisfie the conservation of the equations. Refined grid gives better shock resolutions.

The fourth test case is an inviscid, subsonic flow past a three-element airfoil that has been undergone extensive testing in the Low Turbulence Pressure Tunnel located
at NASA Langley Research Center [32]. Adaptive simulations were performed for a free-stream Mach number of $M_\infty = 0.2$ and an angle of attack (16.2 degree). Fig. 19 demonstrates a series of meshes from the initial to the end of an adaptation loop. The initial coarse grid contains 10,403 node and 20,294 triangles. After three levels of grid adaptation the numbers of final grid nodes and cells increase to 17,611 and 32,957, respectively. Comparing to the pressure contours of the level-2 refined grid (Fig. 19(a)), the solution from the level-3 adapted mesh is greatly improved as shown in Fig. 19(b).

The last case considered is a transient turbulent flow past a half cylinder disc. For this case, the solution adaptive mesh refinement procedure is called for every time step. Thus, the refined mesh follows the development of the flowfield. Figure 20 shows a series of adaptive mesh development due the unsteady feature of the wake flow. When the flow reaches its cyclic solutions, the maximum number of cells vary within a bounded value. Based on this test, for this kind of flow problems, a maximum cell number limit to 3 times the original grid size is a good number to be specified in the input data file.

Figure 12. Mesh of level-1 (initial) for flow past multiple cylinders.

Figure 13. Mesh of level-2 adaptation.
Figure 14. Mesh of level-3 adaptation.

Figure 15. Velocity contours.

Figure 16. Velocity vectors.

Figure 17. Transonic flow around NACA 0012 with adaptation.
Figure 18. Supersonic flow over a bump with adaptation.
Figure 19. Subsonic flow around a three-element airfoil with mesh adaptation.
Figure 20. Test of transient adaptive mesh refinement for a wake flow behind a half cylinder.
Hypersonic Heat Transfer Benchmark Test Case

This test case investigates the interaction between an impinging shock wave generated by a wedge and the bow shock around a circular cylinder. This is a type IV shock-chock interaction in Edney’s classification. The geometry is obtained from the drawing in the test case specification of the Houston High Speed Flow Database (Chapter 11). Umesh is used for mesh generation. The grid size used in computation is 5,248. The following figure shows the mesh system.

Figure 21. Mesh system used in the T11-97 test case CFD computation.

The free stream conditions are:

\[
\begin{align*}
M_\infty &= 9.2 \\
T_\infty &= 797 \text{ K} \\
\rho_\infty &= 2.19 \times 10^{-3} \text{ kg/m}^3 \\
Y_{N_2} &= 0.74 \\
Y_N &= 0.00 \\
Y_{O_2} &= 0.05 \\
Y_O &= 0.17 \\
Y_{NO} &= 0.04 \\
T_{\text{wall}} &= 300 \text{ K}
\end{align*}
\]

Finite-rate chemistry model with 9 species and 20 reactions is used in the computation. The calculated Mach number contour, impinging jet pattern and data comparisons for the wall pressure and heat flux data are shown in the following figures. Good correlation between the data and the current model is revealed in Figures 24 & 25.
Figure 22. Mach number contours predicted for the T11-97 test case.

Figure 23. Velocity Vectors Near the Blunt Body Leading Edge

Impinging jet due to shock-shock interaction, which causes high heating at blunt-body surface.
Figure 24. Blunt body nose surface pressure data comparisons.

Figure 25. Blunt body nose wall heat flux comparisons.
UNIC-UNS INPUT GUIDE

List of the README file:

=================================================================================================
INSTRUCTION FOR UNIC UNSTRUCTURED-GRID CFD CODE

---- Unified Finite Volume Method for Fluid Flow Calculations ------

Revised 10/08/2002

=================================================================================================
If you have any question, please contact:
=================================================================================================

UNIC Department                      Voice:  (256) 883-6233 (O)
Engineering Sciences, Inc.                (256) 883-6445 (O)
1900 Golf Rd, Suite D                Fax:     (256) 883-6267
Huntsville, AL 35802               E-Mail: unic@esi-al.com

=================================================================================================

Introduction

The unstructured grid flow solver has been developed for all-speed flow calculation using pressure-based method at Engineering Sciences Inc. (ESI). The unstructured grid method has the advantages of automated grid generation in very complex domain and flexible mesh adaptation in high gradient region over the structured grid method. The general grid topology allows the use of even the traditional structured grids and optimized grids in viscous boundary layers. A high-order upwind scheme with flux limiter has been incorporated for convection terms. The convergence of the linear algebraic equations is accelerated through a preconditioned conjugate gradient matrix solver. Numerical calculations have been conducted for compressible/incompressible, steady/transient, laminar/turbulent, spray combustion and chemical reacting flows.

A parallel implementation using MPICH-1.2.1 is also developed.

Model Data Files

unic.inp General input data.
curve.bc Used for 2-D unstructured grids to specify boundary condition.
unic.sgb Used for 2/3-D structured grids to specify boundary condition.
fastg.unf unformatted, single block, unstructured FAST grid file.
fastg.fmt formatted, single block, unstructured FAST grid file.
unicg.unf unformatted, multi block, unstructured UNIC-UNS grid file for general grid topology.
unicg.fmt formatted, multi block, unstructured UNIC-UNS grid file for general grid topology.
plot3dg.unf unformatted, multi block, structured PLOT3D grid file.
plot3dg.fmt formatted, multi block, structured PLOT3D grid file.
patrang.fmt formatted, PATRAN unstructured grid file. Boundary index is assigned through pressure load.
fieldvw.fmt formatted, Gridgen unstructured grid file saved in Fieldview format.

Important Pre-/Post-Processing and User Files

xcont Connects multi-block grids into a single block mesh (PLOT3D grids need unic.sgb boundary index file to proceed with this tool).
xprep Generates model data base files: unic.con, unic.xyz, and unic.per (if periodic bc involved).
xconv Converts unic.flo for plotting.
xunic Main program for UNIC-UNS.

Test Cases

# cyln Laminar 2-D incompressible flow past a cylinder.
# mfoil Inviscid 2-D compressible flow past a multi-element air foil.
# naca12 Turbulent 2-D compressible flow past NACA0012 air foil.
    Uses hybrid grids.
# moon Laminar 2-D flow past a half cylinder.
    Uses 2-block structured grids.
# cascade 2-D cascade case with periodic bc.
# port3 3-D portflow case.

Database and Solution Data Files

unic.con unformatted, direct access grid connection data file Generated by xprep.
unic.xyz unformatted, direct access grid xyz location data file Generated by xprep.
unic.per formatted data file for periodic bc. Generated by xprep.
unic.flo Output unformatted, direct access flow solution file.
unic.ini Input unformatted, direct access flow solution file.
    * Copy unic.flo to unic.ini for restart runs.
fieldview.unf  unformatted FIELDVIEW unstructured data file.
fieldview.fmt  formatted FIELDVIEW unstructured data file.
plot3dq1.unf  unformatted, multi-block,unstructured PLOT3D solution file
     (rho, rho*u, rho*v, rho*w, rho*e0).
plot3dq2.unf  unformatted, multi-block,unstructured PLOT3D solution file
     (pressure,temperature,turbulent k,eddy viscosity, Ma/vof).
plot3dq3.unf  unformatted, multi-block,unstructured PLOT3D solution file
     (mass fractions defined by user).
*Note:  Solution files are generated by xconv from unic.flo. They are the same for
structured and unstructured grids. Model data base files include: unic.con and unic.xyz,
and unic.per if periodic bc is involved

To generate model data base files

Method 1: (only for structured multi-block mesh from Umesh or other grid generators)

Enter the command "uinit" in an appropriate directory. Follow the instructions for
running uinit to finish input data and flowfield specifications. Then, select option #4 to
write out UNIC-UNS data files (unic.sgb, unic.unf, unic.con, unic.xyz, unic.001 and
unic_ini). unic.001 is to be copied to unic.inp and unic_ini is to be copied to unic.ini
before start running xunic.

Method 2:

Type the command "xprep" in an appropriate directory. It will use file curve.bc for 2-D
unstructured grids or unic.sgb for structured grids.

xprep  convert the following files:

= 3: FAST UNFORMATTED fastg.unf
= 4: FAST FORMATTED fastg.fmt
= 5: UNIC-UNS UNFORMATTED unicg.unf
= 6: UNIC-UNS FORMATTED unicg.fmt
= 7: PLOT3D UNFORMATTED plot3dg.unf
= 8: PLOT3D FORMATTED plot3dg.fmt
=10: PATRAN FORMATTED patrang.fmt
=12: FIELDVIEW FORMATTED fieldvw.fmt (Gridgen Output)

to

= 1: UNIC-UNS MODEL DATA BASE FILES, UNFORMATTED, DIRECT
     ACCESS unic.con, unic.xyz, unic.per(for periodic bc)
To run UNIC-UNS

The following is for versions using the PVM method. For MPI version, running script is simply: `mpirun –np # xunic > out1 &`. (where # stands for number processors)

1. Modify your `.cshrc` file by inserting the following lines for all parallel computers:
   (Skip steps 1, 2 & 4 for PC Windows systems)

   ```
   #set for PVM
   setenv PVM_ROOT /home8/pvm3
   if ($?PVM_ROOT) then
       setenv PVM_ARCH `$PVM_ROOT/lib/pvmgetarch`
       set path=($path $PVM_ROOT/lib)
   endif
   ```

2. Prepare a `.rhosts` file to include the hosts to be used.
   
   cpc1.esi.inc
   cpc2.esi.inc
   cpc3.esi.inc
   cpc4.esi.inc
   cpc5.esi.inc
   cpc6.esi.inc

3. Create executable `xunic` file in your source directory:

4. Prepare a hostfile contains the host names. Type `pvmd hostfile &` to launch the PVM. (This step is not need for sequential runs.)

5. In the working directory, type `xunic` for sequential runs and `xunic -p` for parallel runs. (Note: Parallel model may not work properly on PC Windows environment due to PVM basic functions that does not allow assignment of specific working directories)

To restart UNIC-UNS

Simply copy `unic.flo` to `unic.ini`, then resubmit the job for succeeding runs. These two files are direct access files. The same files are used for serial and parallel runs.

To convert `unic.flo` for plotting

Type the command "`xconv`" in appropriate directory. `xconv` reads input from `conv.inp`.
Or, without converting, you can just use the `tecplot.dat` output file for TECPLOT.
Data structure for FAST grids

C.....NODE: NUMBER OF NODES; NBFACE: NUMBER OF BOUNDARY FACES
C.....NELE: NUMBER OF ELEMENTS OR CELLS
C.....X, Y AND Z(): NODE LOCATIONS
C.....IBFACE(3:5): BOUNDARY FACE NODE IDENTIFICATIONS
C.....IBFACE(2): BOUNDARY FACE SURFACE GROUP
C.....IDNODE(): CELL NODE IDENTIFICATIONS
OPEN(8,FILE='fastg.unf',FORM='UNFORMATTED')
READ(8) NODE,NBFACE,NELE
KNODE=4           ! 3D TETRAHEDRON, 4 NODES PER CELL
IF(NELE.EQ.0) THEN
  KNODE=3         ! 2D TRIANGULAR, 3 NODES PER CELL
  NELE=NBFACE
  NBFACE=0
ENDIF
READ(8) (X(I),I=1,NODE),(Y(I),I=1,NODE),(Z(I),I=1,NODE),
&    ((IBFACE(J,IF),J=3,5),IF=1,NBFACE),
&    (IBFACE(2,IF),IF=1,NBFACE),
&    (IDNODE(I),I=1,NELE*KNODE)

Data structure for UNIC-UNS grids

OPEN(8,FILE='unicg.unf',FORM='UNFORMATTED')
READ(8) NZON,IDIM ! NUMBER OF ZONE AND DIMENSION (2D OR 3D)
READ(8) NELE,NODE,NBFACE,NINODE ! NINODE: MAXIMUM INDEX FOR IDNODE()
READ(8) (X(I),Y(I),Z(I),I=1,NODE) ! NODE LOCATION
READ(8) (INODE(I),I=1,NELE+1)   ! INODE(I:I+1): CELL NODE NUMBER
READ(8) (IDNODE(I),I=1,NINODE)  ! CELL NODE ID
READ(8) ((IBFACE(J,I),J=1,6),I=1,NBFACE) ! BOUNDARY FACES
READ(8) (IPID(I),I=1,NELE) ! MATERIAL INDEX (0:Fluid; >0:Solid)

Sample Input Data File

# A sample of input data file
# TITLE: FLOW PAST NACA0012 AIR FOIL (M=0.799 ALFA=2.26 DEGREE)
# IDIM: DIMENSION OF PROBLEM (=2: 2D; -3: 3D)
# IGRID: GRID DATA FORMAT (UNIC-UNS ONLY READ ITS MODEL DATA
# BASE FILES)
# -1: UNIC-UNS DATA BASE FILES, UNFORMATTED, DIRECT ACCESS
# (Default)
# unic.con, unic.xyz, unic.per (for periodic bc)
# IAX: =1: PLAIN; =2: AXISYMMETRIC
# ICYC: TYPE OF CYCLIC BC (NOT IN USE)
# LCONG: GRID CONNECTION (NOT IN USE)
# IDCASE: CASE IDENTIFICATION
# LDBG: =0: NO OUTPUT OF DEBUG INFORMATION; =1, 2: YES
#  IDIM IGRID IAX ICYC LCONG IDCASE LDBG
#  2  1  1  0  0  2  0
### #NSSTEP: NUMBER OF TIME STEP OR GLOBAL ITERATION
### #ITPNT: SOLUTION OUTPUT FREQUENCY
# >0: UPDATE unic.flo EACH ITPNT STEPS (also TECPLT output file)
# <0: OUTPUT unic.flo.LSTEP EACH ITPNT STEPS (also TECPLT output file)
# (MAY USE UP LARGE DISK SPACE)
### #IREC: CHAKRAVARTHY-OSHER SCHEME FOR CONVECTION TERM
# = 1: BARTH's SCHEM, 2ND ORDER (Recommended)
# =-2: CENTRAL, 2ND ORDER
### #REC: 0<= REC <=1: FOR UPWIND DAMPING (=1: FULLY FIRST ORDER UPWIND)
### #EREXT: RUN STOP WHEN MAXIMUM RESIDUAL < EREXT (<0: STOP AT NSTEP)
### #IMON: MONITOR POINT
### #THETA: TIME MARCHING SCHEME PARAMETER
# =1.0: FOR STEADY-STATE APPLICATIONS (FIRST-ORDER TIME MARCHING)
# =0.5: FOR SECOND-ORDER TRANSIENT APPLICATIONS (SUBITERATION METHOD)
### #BETAP: PRESSURE UPDATING UNDER RELAXATION PARAMETER (0 < BETAP <= 1)
# (TYPICALLY 1.0)

<table>
<thead>
<tr>
<th>NSTEP</th>
<th>ITPNT</th>
<th>IREC</th>
<th>REC</th>
<th>EREXT</th>
<th>IMON</th>
<th>THETA</th>
<th>BETAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>5000</td>
<td>10</td>
<td>1</td>
<td>0.00</td>
<td>-1.0E-7</td>
<td>1</td>
<td>1.000</td>
<td>1.00</td>
</tr>
</tbody>
</table>

### #DTT: TIME STEP SIZE (IN SEC)
### #URELX: VELOCITY DAMPING FACTOR (AP=SP+ANB+AMAX1(RHO*VOL/DTT, URELX*ANB))
### #PRELX: PRESSURE DAMPING FACTOR (AP=SP+ANB+AMAX1(RHO*VOL/DTT, PRELX*ANB))
### #TMRELX: TEMPERATURE DAMPING FACTOR (AP=SP+ANB+AMAX1(RHO*VOL/DTT, TMRELX*ANB))
### #DKRELX: K ABD E DAMPING FACTOR (AP=SP+ANB+ANB(RHO*VOL/DTT, DKRELX*ANB))
### #FMRELX: MASS-FRACT. DAMPING FACTOR (AP=SP+ANB+ANB(RHO*VOL/DTT, FMRELX*ANB))
### #FACU: FOURTH ORDER PRESSURE DAMPING TERM (0<= FACU <=1)

<table>
<thead>
<tr>
<th>DTT</th>
<th>URELX</th>
<th>PRELX</th>
<th>TMRELX</th>
<th>DKRELX</th>
<th>FMRELX</th>
<th>FACU</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0E-4</td>
<td>0.50</td>
<td>0.005</td>
<td>0.50</td>
<td>1.00</td>
<td>0.50</td>
<td>0.00</td>
</tr>
</tbody>
</table>

### #GRVX: GRAVITY FORCE ABOUT X-AXIS (M/S**2 OR FT/S**2)
### #GRVY: GRAVITY FORCE ABOUT Y-AXIS (M/S**2 OR FT/S**2)
### #GRVZ: GRAVITY FORCE ABOUT Z-AXIS (M/S**2 OR FT/S**2)
### #OMGX: FRAME ROTATIONAL SPEED ABOUT X-AXIS (1/S)
### #OMGY: FRAME ROTATIONAL SPEED ABOUT Y-AXIS (1/S)
### #OMGZ: FRAME ROTATIONAL SPEED ABOUT Z-AXIS (1/S)

<table>
<thead>
<tr>
<th>GRVX</th>
<th>GRVY</th>
<th>GRVZ</th>
<th>OMGX</th>
<th>OMGY</th>
<th>OMGZ</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.0</td>
<td>0.00</td>
</tr>
</tbody>
</table>

### #DENG: GAS PHASE DENSITY (KG/M**3 OR LBM/M**3)
### #DENF: LIQUID PHASE DENSITY (KG/M**3 OR LBM/M**3)
### #SFTN: SURFACE TENSION COEFFICIENT (N/M OR LBF/M)
### #VISF: LIQUID PHASE VISCOSITY (KG/(M*S) OR LBM/(FT*S))

<table>
<thead>
<tr>
<th>DENG</th>
<th>DENF</th>
<th>SFTN</th>
<th>VISF</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

### #GAMA: GAS SPECIFIC HEAT RATIO (DEFAULT VALUE FOR AIR)
### #RMXBAR: GAS CONSTANT (DEFAULT VALUE FOR AIR)
### #SIGT: GAS LAMINAR PRANDTL NUMBER (DEFAULT VALUE FOR AIR)
### #T_POWER: GAS VISCOSITY TEMPERATURE POWER LAW PARAMETER (DEFAULT VALUE FOR AIR)

<table>
<thead>
<tr>
<th>GAMA</th>
<th>RMXBAR</th>
<th>SIGT</th>
<th>T_POWER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.3896</td>
<td>287.04</td>
<td>0.72</td>
<td>0.67</td>
</tr>
</tbody>
</table>

---

40
#XREF: REFERENCE LENGTH (METERS)
#VISC: GAS PHASE REFERENCE VISCOSITY (KG/(M*S)
#AMC: REFERENCE MACH NUMBER (TO INDICATE COMPRESSIBLE OR INCOMPRESSIBLE)

# XREF  VISC  AMC
1.0000  1.84E-5  0.7990

#SPECIFY EQUATIONS TO BE SOLVED AND CG SOLVER (-0: DEACTIVE, >-1: ACTIVE)
#U,V,W: U,V, AND W VELOCITY EQUATION (0 OR 1)
#P: NUMBER OF PRESSURE CORRECTIONS (=0: NOCORRECTION)
#TM: ENERGY EQUATION (0 OR 1)
#DK: TURBULENT KINETIC ENERGY EQUATION
#   =1: STANDARD K-E MODEL
#   =2: EXTENDED K-E MODEL
#   =3: TWO-LAYER MODEL (NOT YET AVAILABLE)
#   =4: LOW-REYNOLDS MODEL (NOT YET AVAILABLE)
#   =5: ALGEBRA STRESS MODEL (NOT YET AVAILABLE)
#   =6: REYNOLDS STRESS MODEL (NOT YET AVAILABLE)
#DE: TURBULENT KINETIC ENERGY DISSIPATION EQUATION (0 OR 1)
#FM: MASS-FRACTION EQUATION (0 OR 1)
#VF: VOF EQUATION (0 OR 1)
#10: RESERVED FOR RADIATIVE TRANSFER MODEL
#11: RESERVED FOR VIBRATIONAL ENERGY EQUATION
#12: RESERVER FOR ELETRONE ENERGY EQUATION
#IERCG: CG MATRIX SOLVER CONTROL
#   = 1: ALL EQUATIONS SOLVED BY BiCGSTAB SOLVER
#   = 2: PRESSURE EQUATION SOLVED BY GMRES SOLVER, OTHERS BY BiCGSTAB
#   = 3: PRESSURE EQUATION SOLVED BY MULTI-GRID SOLVER, OTHERS BY BiCGSTAB
#   =-3: ALL EQUATIONS SOLVED BY MULTI-GRID SOLVER

# SOLVE: U  V  W  P  TM  DK  DE  FM  VF  10  11  12  IERCG
1  1  0  1   1   1   1   0   0   0   0      2

#BOUNDARY CONDITION ++++++++++++++++++++++++++++++++++++++++++++++++++++++++
#NDBC: NUMBER OF BOUNDARIES TO BE DEFINED
#IDBC: BOUNDARY INDEX
#IBTY: TYPE OF BOYNDARY CONDITIONS
#   =< 1: INLET OR FREESTREAM; 2: OUTLET;
#   3: SYMMETRY OR INVISCID WALL; 4: WALL
#-----INLET B.C. SPECIFICATION
#   = 0: INLET FIX EVERYTHING (SUPERSONIC INLET);
#   =-1: FOR SUBSONIC INLET FIX TOTAL P AND T, EXTRAPOLATE P
#       FOR INCOMPRESSIBLE FLO, EXTRAPOLATE P
#   =-1: INLET FIX MASS FLOW RATE, EXTRAPOLATE P
#   =-2: INLET FIX EVERYTHING EXCEPT P AND DENSITY
#   =-3: FOR FREESTREAM FAR-FIELD FIX TOTAL P AND T, SOLVE FOR U,V,W
#PRAT(or PEXIT): SPECIFIES FREESTREAM/OUTLET PRESSURE CONDITION OPTION (in ATM)
#   = 0.0: FOR OUTLET MASS CONSERVATION
#   =-1.0: FOR SUPERSONIC OUTLET BC
#       OR = 9999.0 FOR FIXED UNIFORM PRESSURE AT OUTLET
#TRAT: FREESTREAM TOTAL TEMPERATURE CONDITION (in K)
#XPFIX: X-COORDINATE FOR PRESSURE ANCHORING LOCATION (in grid unit)
#       (Active when IBTY = 2 and PRAT > 0.0)
#   OR = 9999.0 FOR FIXED UNIFORM PRESSURE AT OUTLET
#YPFIX: Y-COORDINATE FOR PRESSURE ANCHORING LOCATION (in grid unit)
#ZPFIX: Z-COORDINATE FOR PRESSURE ANCHORING LOCATION (in grid unit)
#TMEX: ENERGY EQUATION BOUNDARY CONDITIONS
#   =-1.0 FOR FIXED TEMPERATURE CONDITION
# = 0.0 FOR ADIABATIC CONDITION (Zero Gradient)--DOES NOT APPLY TO INLET
# IBTYRD: = 0: NON-REFLECTIVE SURFACE FOR RADIATION
# > 0: REFLECTIVE SURFACE FOR RADIATION
# EMITB: BOUNDARY EMISSIVITY FOR RADIATION (0.0 TO 1.0)

## NDBC
3

<table>
<thead>
<tr>
<th>IDBC</th>
<th>IBTY</th>
<th>PRAT</th>
<th>TRAT</th>
<th>XPFIX</th>
<th>YPFIX</th>
<th>ZPFIX</th>
<th>TMEX</th>
<th>IBTYRD</th>
<th>EMITB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-1</td>
<td>0.0</td>
<td>300.0</td>
<td>0.0</td>
<td>0.0</td>
<td>-1.00</td>
<td>0</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1.0</td>
<td>300.0</td>
<td>0.0</td>
<td>9999.0</td>
<td>0.0</td>
<td>-1.00</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>0.0</td>
<td>300.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>-1.00</td>
<td>1.0</td>
<td></td>
</tr>
</tbody>
</table>

## INITIAL FIELD & BOUNDARY VALUE +++++++++++++++++++++++++++++++++++++++++

<table>
<thead>
<tr>
<th>NZON</th>
<th>NDBC</th>
<th>ISETBC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IZON</th>
<th>UIN</th>
<th>VIN</th>
<th>WIN</th>
<th>PIN</th>
<th>Tmin</th>
<th>DKIN</th>
<th>DEIN</th>
<th>VOFIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>271.45</td>
<td>10.71</td>
<td>0.00</td>
<td>1.00</td>
<td>300.00</td>
<td>1.0000</td>
<td>-0.10</td>
<td>0.00</td>
</tr>
<tr>
<td>2</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>300.00</td>
<td>1.0000</td>
<td>-0.10</td>
<td>0.00</td>
</tr>
<tr>
<td>3</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>300.00</td>
<td>1.0000</td>
<td>-0.10</td>
<td>0.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IDBC</th>
<th>UBC</th>
<th>VBC</th>
<th>WBC</th>
<th>PBC</th>
<th>TMBC</th>
<th>DKBC</th>
<th>DEBC</th>
<th>VOFBC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>271.45</td>
<td>10.71</td>
<td>0.00</td>
<td>1.00</td>
<td>300.00</td>
<td>1.0000</td>
<td>-0.10</td>
<td>0.00</td>
</tr>
<tr>
<td>2</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>300.00</td>
<td>1.0000</td>
<td>-0.10</td>
<td>0.00</td>
</tr>
<tr>
<td>3</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>300.00</td>
<td>1.0000</td>
<td>-0.10</td>
<td>0.00</td>
</tr>
</tbody>
</table>

## FOR SPECIES MASS FRACTION (IF NGAS>8, ADD MORE LINES)

<table>
<thead>
<tr>
<th>IZON</th>
<th>FM01</th>
<th>FM02</th>
<th>FM03</th>
<th>FM04</th>
<th>FM05</th>
<th>FM06</th>
<th>FM07</th>
<th>FM08</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0399</td>
<td>0.2234</td>
<td>0.00</td>
<td>0.00</td>
<td>0.7367</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>2</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>3</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

## SPECIES PROPERTIES ++++++++++++++++++++++++++++++++++++++++++++++++++++++++

NGAS: NUMBER OF CHEMICAL SPECIES
ICEC: 1: FOR CEC DATA; 2: FOR CEA DATA
(If $NGAS TITLE IS READ INSTEAD OF $NGAS_ICEC, ICEC = 2 IS ASSUMED)
(Then, followed by species data list)

$NGAS_ICEC ! (CEA format in this case; ICEC = 1 for CEC format)

6 2

H2O  1  0.0399  0.2234  0.00  0.00  0.7367  0.00  0.00  0.00
#CHEMICAL REACTIONS

**NREACT:** NUMBER OF CHEMICAL REACTIONS

**N3BE:** NUMBER OF THIRD-BODY EFFICIENCY INPUT (100% IS ASSUMED W/O INPUTS)

#CHEMICAL REACTIONS

$NREACT

9

<table>
<thead>
<tr>
<th>#I</th>
<th>A</th>
<th>N</th>
<th>E/R</th>
<th>I3R</th>
<th>IGL</th>
<th>ITME</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.700E13</td>
<td>0.00</td>
<td>24233.0</td>
<td>0</td>
<td>0</td>
<td>0 / O₂ + H₂ = 2 OH</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2.190E13</td>
<td>0.00</td>
<td>2590.0</td>
<td>0</td>
<td>0</td>
<td>0 / H₂ + OH = H₂O + H</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>6.023E12</td>
<td>0.00</td>
<td>550.0</td>
<td>0</td>
<td>0</td>
<td>0 / 2 OH = H₂O + O</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1.800E10</td>
<td>1.00</td>
<td>4480.0</td>
<td>0</td>
<td>0</td>
<td>0 / H₂ + O = H + OH</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1.220E17</td>
<td>-0.91</td>
<td>8369.0</td>
<td>0</td>
<td>0</td>
<td>0 / O₂ + H = O + OH</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1.000E16</td>
<td>0.00</td>
<td>0.0</td>
<td>999</td>
<td>0</td>
<td>0 / O + H = OH</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>2.550E18</td>
<td>-1.00</td>
<td>59390.0</td>
<td>0</td>
<td>0</td>
<td>0 / 2 O = O₂</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>5.000E15</td>
<td>0.00</td>
<td>0.0</td>
<td>999</td>
<td>0</td>
<td>0 / 2 H = H₂</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>8.400E21</td>
<td>-2.00</td>
<td>0.0</td>
<td>999</td>
<td>0</td>
<td>0 / H + OH = H₂O</td>
<td></td>
</tr>
</tbody>
</table>

$N3BE

4

#3RDBE: H₂O, O₂, H₂, O, H, OH, N₂

6, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.0

7, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.0

8, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.0

9, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.0

#GRID REFINEMENT ADAPTATION CONTROL

$LREF: <= 1: NO REFINEMENT; >= 2: INDICATES LEVEL OF REFINEMENT

$MEM_ES: ESTIMATED MEMORY SIZE LIMIT (3 IS A GOOD STARTING NUMBER)

$INPNT_ADP: INDICATES FREQUENCY OF PERFORMING ADAPTIVE MESH REFINEMENT

$IADPF: MESH REFINEMENT FORMULATION INDEX

1: ALGEBRAIC AVERAGE OF DENSITY, PRESSURE AND VELOCITY MAGNITUDE

2: DENSITY

3: PRESSURE

4: TEMPERATURE

5: ALGEBRAIC AVERAGE OF DENSITY AND MACH NUMBER

6: TURBULENCE KINETIC ENERGY

7: VORTICITY MAGNITUDE

#GRID REFINEMENT ADAPTATION CONTROL

$ADAPTIVE GRID

$LREF  MEM_ES  INPNT_ADP  IADPF

0  3  200  5

#PERFORMANCE DATA INTEGRATION

$BODY-FORCE INTEGRATION

$IDBODY IDUNIT

2  1

$(IDBODY SETS OF THE FOLLOWING, KK=1,IDBODY)

$IDNUMB IINLET

1  1

$IDFORC(KK,J), J=1,IDNUMB(KK)

3

$IDMONT(KK,J), J=1,IINLET(KK)

2

$IDNUMB IINLET
1 0
# IDFORC(KK,J),J=1,IDNUMB(KK)
1
# IDMONT(KK,J),J=1,IINLET(KK)
1

#USER DATA OUTPUT FOR XY- PLOT PRESENTATIONS ----------------------------------------

#XY- PLOT SPECIFICATION:
# $XY- PLOT: TITLE LINE FOR THIS SECTION
# N: 
# (Followed by N sets of input lines): Curve-Data-File-Name, up to 5 Data Names
# Data Names Available: DENSITY (in kg/m**3)
# PRESSURE (in ATM)
# TEMPERATURE (in K)
# MACH_NUMBER
# U_VELOCITY (in m/s)
# V_VELOCITY (in m/s)
# W_VELOCITY (in m/s)
# SHEAR_STRESS (in kg/m-s**2)
# HEAT_FLUX (in watts/m**2)
# Y_PLUS
#CURVE LINE DATA FORMAT (in Grid Coordinate Units):
# x1  y1  z1
# x2  y2  z2
# x3  y3  z3
# ..  ..  ..
#NOTE: Default output file names are xyout-#.dat, where # represents the
# sequence number of the input data lines of this group.
# Output coordinates, (X,Y,Z and S) are in meters.

#USER DATA OUTPUT FOR XY- PLOT PRESENTATIONS ----------------------------------------

$XY- PLOT
2
line1.dat  DENSITY  PRESSURE  TEMPERATURE  MACH_NUMBER  SHEAR_STRESS
line2.dat  U_VELOCITY  V_VELOCITY  W_VELOCITY  HEAT_FLUX  Y_PLUS
#

#USER DATA OUTPUT FOR TIME-HISTORY PRESENTATIONS --------------------------------------

#XY- TIME SPECIFICATION:
# $XY- TIME: TITLE LINE FOR THIS SECTION
# (Followed by 1 input line): Point-Data-File-Name, up to 5 Data Names
# Data Names Available: Same as in the XY- PLOT data group
# Note: Default output file names are xytime-#.dat, where # represents the
# sequence number of the coordinate data points contained in the
# Point-Data-File-Name of this group (e.g. point-time.dat). Therefore,
# do not include too many points in this file to avoid too many output
# files created for monitoring time history data.

#USER DATA OUTPUT FOR TIME-HISTORY PRESENTATIONS --------------------------------------

$XY- TIME
point-time.dat  DENSITY  PRESSURE  TEMPERATURE  MACH_NUMBER  SHEAR_STRESS
#

#PARALLEL COMPUTING CONTROL-------------------------------------------------------------

#IPART:  1: BUILT-IN RECURSIVE COORDINATE BISECTION (RCB) METHOD
# 2: METIS RECURSIVE GRAPH BISECTION (RGB) METHOD -- NEEDS TO RUN xmetis
#(FOLLOWED BY A LIST OF PROCESSES)

#PARALLEL COMPUTING CONTROL-------------------------------------------------------------

$PARALLEL
#IPART
1 2
host=cpc1.esi.inc
edir=/home8/ychen/UNIC/src
Conjugate Heat Transfer Model Setup

Run xprep to set up the Conjugate Heat Transfer Model
1. Read Grid Options

PLEASE INPUT IGRID FOR INPUT:
   IGRID= 3: fastg.unf; = 4: fastg.fmt
   IGRID= 5: unicg.unf; = 6: unicg.fmt
   IGRID= 7: plot3dg.unf; = 8: plot3dg.fmt
   IGRID ; =10: patrang.fmt
   IGRID=11: fieldvw.unf; =12: fieldvw.fmt

Multi-block PLOT3D grid file is read directly (you'll be asked for the file name, e.g. f12.fmt). For other grid formats, separate grid files for fluid and solid domains need to be converted to unicg.fmt format individually, then combined into a multi-zone unicg.fmt file manually, before running this option. Next, the following question is asked:

**************************************************************************
* DOES THE PROBLEM INVOLV CONJUG. HEAT TRANS.?? *
* INPUT 1 FOR YES, 0 FOR NO *
**************************************************************************

If YES, The xprep module allows you to set block IDs (IPID) for each block. Here, BLOCK ID = 0 is for fluid domains and BLOCK ID > 0 is for identifying solid blocks.
Then, xprep performs the block connection function and checks the data integrity. Database files, unic.xyz and unic.con, are written on the current working directory.

2. Setup Parameters in the Input File, unic.inp (unic.inpn is a temporary input file after this setup)

```
***(unic.inp Physical Model List)********************
CONJUGATE HEAT TRANSFER ==========================
NBLOCK
  2
IBLOCK   CTHCN   BLDEN   BLCP
  1 .200E+03  .300E+04  .350E+03
  2 .100E+03  .200E+04  .600E+03
Enter Line No. listed above to Modify:
(0 to end the first part of CH Modification)
```

Just type in Line Number then enter properties for each block. These parameters are defined as:

- **CTHCN**: Solid Thermal Conductivity, watts/m-K
- **BLDEN**: Solid Material Density, kg/m**3
- **BLCP**: Solid Heat Capacity, m**2/sec**2-K

Follow directions from the command line, you can setup various models to be included in the input file, unic.inpn. For example, let’s select option 3 for the conjugate heat transfer model, the following entry line appears.

```
********(unic.inp Physical Model List)*************
CONJUGATE HEAT TRANSFER ==========================
NBLOCK
  2
IBLOCK   CTHCN   BLDEN   BLCP
  1 .200E+03  .300E+04  .350E+03
  2 .100E+03  .200E+04  .600E+03
Enter Line No. listed above to Modify:
(0 to end the first part of CH Modification)
```

Just type in Line Number then enter properties for each block. These parameters are defined as:

- **CTHCN**: Solid Thermal Conductivity, watts/m-K
- **BLDEN**: Solid Material Density, kg/m**3
- **BLCP**: Solid Heat Capacity, m**2/sec**2-K
Then, enter 0 to proceed to the second part of this setup. The following command lines appear.

```
NDBC
  8
  IDBC   TFLD   COCH
  1   .000E+00  .000E+00
  2   .000E+00  .000E+00
  3   .000E+00  .000E+00
  4   .000E+00  .000E+00
  5   .000E+00  .000E+00
  6   .000E+00  .000E+00
  7   .000E+00  .000E+00
  8   .300E+02  .100E+02
```

These input lines provide option for heat transfer between ambient far field and the solid boundaries (effective only for IBTY=6 boundaries set in the NDBC boundary group). The above parameters are defined as:

- **TFLD**: Ambient Far Field Temperature, K
- **COCH**: Heat Transfer Coefficient,

Finally, enter 99 to save the input data in unic.inpn, then, enter 0 to finish the setup session.

3. Copy `unic.inpn` to `unic.inp` and modify if necessary. And, you are ready to run the conjugate heat transfer model.

### Wall Temperature Assignments in UNS

The UNS code comes with tools to do the operations of wall temperature assignments. First of all, you need to prepare a file, `walltm.dat`, in surface data format to map out the temperature distributions on those wall patches you want to assign temperature. The format is:

```
Npatch ----- number of surface patches
  do i=1,Npatch
    (Imax, Jmax) of each patch
  enddo
  do i=1,Npatch
    do j=1,Imax*Jmax
      (X, Y, Z, Temperature) of each point
    enddo
  enddo
```

The (X,Y,Z) coordinates have the same unit as the mesh and Temperature is in deg-K. The wall surface patches you need to prepare are those you want to assign temperature distributions only (not all wall surfaces).
Next, check um01.inc to make sure that the following two lines are used in the surface cells searching do loop:

\[
\begin{align*}
ybc &= ye(ie2) \\
zbc &= ze(ie2)
\end{align*}
\]

where, ybc and zbc represent the cell-center (y, z) coordinates of boundary cells. And, make sure that the next two lines are not in use (the following two lines are for axisymmetric geometry setup only):

\[
\begin{align*}
ybc &= \sqrt{ye(ie2)^2 + ze(ie2)^2} \\
zbc &= 0.0
\end{align*}
\]

For some coding in um01.inc, axisymmetric type surfaces with single-curve data input in walltm.dat are used. So, the above operation for um01.inc makes the code general for any surface shapes.

In um01.inc, "call bcdata(3,...)" reads walltm.dat and "call bcdata(4,...)" performs bilinear data interpolations.

Then, recompile the code in the local working directory to take effect of the above changes. The following commands are used to recompile the code:

```
> getuns
> vi um01.inc  ➔ To check and modify the coding describe above. If no change is needed (i.e. correct coding is already in use), then skip the make command next (no need to recompile the program).
> make
```

As a result, new executable, xunic.exe is created in the local working directory. The above recompilation operation assumes that the user has the Power FORTRAN 4.0 installed in the system.
DOMAIN DECOMPOSITION & COMMUNICATION EXAMPLE

DOMAIN DECOMPOSITION

For parallel computing, the computational area is subdivided to subblocks, and each subblock is assigned to each processor. The total grid size of each processor is equal so that the load is balanced. The domain decomposition methods used in the code are Recursive Coordinate Bisection (RCB) and Recursive Graph Bisection (RGB). For RGB a library, METIS-4.0 developed at University of Minnesota is used.

COMMUNICATION

The communication routines for parallel computing with PVM include the following sections:

1. Process Initialization
   All the processors are initialized by starting the code and reading their respective initial data, grid and flow field files.

2. Information Exchange
   Exchange the zonal boundary conditions for solution field variables, such as density, velocity and pressure etc., during the iterative time marching process.

3. Process Exit
   Collection of output files and termination of execution of each processor.

The information exchange between processors is carried by point-to-point communications and the global communications by collective communications. The routines used in this code are blocking send and receive (PVM-Send & PVM-Receive) for point-to-point communications. For faster communication for large arrays, PVM-Pack and PVM-Unpack calls have been used.

RESULTS & DISCUSSION

NACA-12

Pressure contours of sequential and parallel results of NACA-12 are presented. Fig.A.1 shows the sequential pressure contour, Fig.A.2 presents Recursive Graph Bisection (RGB) decomposition of the domain, and, Fig.A.3 presents Recursive Coordinate Bisection (RCB) decomposition. Each domain is allocated to one processor. From the results it can be clearly seen that parallel and sequential results are in agreement.
Fig. A.1 Sequential Pressure Contour

Fig. A.2 Parallel Pressure Contour with RGB Decomposition of the domain

Fig. A.3 Parallel Pressure Contour with RCB Decomposition of the domain
REFERENCES