A 2D Moving Boundary Cartesian Grid Solver for Internal Flow Fields of SPRM’s

KEREM PEKKAN*  
Post-Doctoral Fellow  
Purdue School of Engineering and Technology  
Mechanical Engineering Department, IUPUI 723  
West Michigan Street, Suite SL260  
Indianapolis, IN 46202  
USA  
kpekkan@iupui.edu

AHMET S. ÜCER  
Professor  
Middle East Technical University (ODTU)  
Makina Muh. Bolumu  
Inonu Bulvari  
06531 Ankara  
TURKEY  
ucer@metu.edu.tr

SUMMARY
In this report a two-dimensional Cartesian grid solver will be described that is originally designed to be used in solid propellant internal flow modeling. The aim is to obtain a fast, practical time-dependent solution by eliminating the re-meshing requirement and connectivity load of classical unstructured meshes. Moreover by the methods described, the solver can be used in “continuous” transient motor instability calculations where the frequency or phase of the oscillations changes during the course of motor operation. In the previous quasi one-dimensional studies, the effect of grain movement has been studied. Here an extension to multi-dimensions will be presented with test cases and burnback applications in Cartesian grids.

INTRODUCTION

Computational Internal Flow Modeling
For the complete motor operation, the three-dimensional unsteady compressible flow equations can be solved as it was done both for liquid [1] and solid [2] propellant motors. Navier-Stokes [3] and Euler [4,5] formulations can predict associated transient phenomena leading to motor instability. Other attempts to solve “cold flow” inside solid propellant motors are published [6-9]. Two-phase flow sources can be included to the formulation [10]. Full motor simulations with reaction is demonstrated by applying homogeneous three-step solid propellant gas/solid phase chemical kinetics mechanisms [11], and with eddy break-up turbulent reaction models [12]. In all of these studies internal flow fields are obtained for fixed grain geometry at the choked steady state.

Moving Boundaries in SPRM’s
Efforts to predict full motor internal solution with the grain boundaries modeled as moving surfaces are relatively rare. To the author’s knowledge, there are two recent full motor studies. In Ref. [13], with a

* Research Engineer at Roketsan Missiles Industries Inc., Ankara/TURKIYE, during the course of study.

body-fitted unstructured grid, performance of a cylindrical grain motor is predicted. Euler equations are solved on a moving grid employing the flux-difference scheme of Roe. Numerical modeling details of the internal flow prediction are not presented but a pressure-time history of a single motor is compared with the applied steady erosive burning rate model. More complex moving grain shapes are also studied [14], where one method is to update the combustion chamber geometry at each time step by a separate surface burnback code and renew the tetrahedral mesh. Multi-block and hybrid grid options are utilized to generate a high quality nozzle grid independently.

Recently [15], for an end-burning grain configuration effect of boundary movement to instability and internal ballistics is studied for different motor lengths, chamber pressures, burning rate models and propellant response functions. A continuous time-accurate computational solution method is described for cases where the oscillation frequency changes during the motor operation. At each time step the forcing frequency is estimated real-time, from a fixed number of filtered backward samples of the pressure data. In order to calculate the amplitude of pressure oscillations for a given phase lag that can appear during the non-stationary regions of the pressure history (pressurization and tail-off), a cubic least square polynomial is fitted to the backward data samples. In terms of grain movement the traditional stationary grain assumption is found to be valid, even for ultra fast propellants that are used in high-acceleration interceptor vehicles. Due to the high-energy source term the rarefaction effect of the regressing interface cannot be observed. This result may be valid only for an end burning configuration since according to another recent study, investigating the effect of *radial* wall regression via the analytical injection-driven, incompressible, inviscid flow approach [16] has predicted significant differences in the resulting streamlines for low gas injection velocities and propellant regression rates of around 0.1 m/s. By this study it will be possible to extend these previous studies, make comparisons and include grain movement in multi-dimensions.

**Cartesian Grids**

For computational gas dynamics applications, involving complex 2D bodies, Cartesian Grid approach is an alternative to unstructured grid solvers. Quirk [17] defines the building blocks and solver related requirements for such an approach. External flows around stationary bodies were considered. Unlike [18] where various in cell boundary approximations are compared Quirk has kept the exact orientations of the cut portions of solid walls relative to Cartesian control volumes. Cartesian grid generation algorithms are readily applicable to modern combined grid methods. Some examples are given in [19-21]. Adaptive mesh refinement [17,22,23], three-dimensional [18,23] and viscous [22,24] extensions of the Cartesian cell-based approach are studied in the literature for stationary boundaries.

Moving boundary problems can be handled by Fixed Grid, Boundary Conforming [25] or Level Set Methods [26]. Each method has its own advantages. The fixed grid finite volume approach is especially suitable for solvers utilizing Cartesian Grids and favored in this study, since the correct representation of characteristic waves generated by the moving boundaries requires the accurate orientation of moving boundary relative to the control volume [42]. Wall boundary conditions are applied with respect to the moving boundary. Moreover satisfying an extra geometric conservation law [27,28] is not required for every cell in the solution domain.

The core part is a finite volume solver. Fluxes from cell faces are calculated by the Godunov method, which utilizes an exact Riemann solver that is generic in terms of the code structure and its functions. In this paper, the main focus is on boundary movements and geometry handling procedures. Assembling and performance comparisons of variations of flux difference/vector splitting procedures will not be presented. Such studies are available in literature [29-32]. The exact Riemann core solver can easily be replaced by higher order flux
calculation schemes (Variations of TVD and WAF schemes, MUSCL-Hancock methods) that are available from the solver database [33], coded in the form of C language classes.

GOVERNING EQUATIONS

Two Dimensional Euler equations are presented as follows:

\[ \vec{U}_t + \vec{F}(\vec{U})_x + \vec{G}(\vec{U})_y = \vec{S}(\vec{U}) \]  

where,

\[
\begin{bmatrix}
\rho \\
\rho u \\
\rho v \\
E \\
\rho q_i
\end{bmatrix}
, \quad
\vec{F} = 
\begin{bmatrix}
\rho u \\
\rho u^2 + P \\
\rho uv \\
u(E + P) \\
\rho u q_i
\end{bmatrix}
, \quad
\vec{G} = 
\begin{bmatrix}
\rho v \\
\rho v^2 + P \\
\rho uv \\
v(E + P) \\
\rho v q_i
\end{bmatrix}
, \quad
\vec{S}(\vec{U}) = 
\begin{bmatrix}
\rho v/r - S_p \\
\rho v/r \\
\rho v^2/r - u v S_p \\
v(E + P)/r - H_{in} S_p \\
\rho v q_i/r - q_{(inj)} S_p
\end{bmatrix}
\]

\( E \) is the total energy per unit volume \( E = \rho (0.5 \vec{V}^2 + e) \). \( \vec{V}^2 = (u^2 + v^2 + w^2) \) is specific kinetic energy and \( e \) is specific internal energy. The perfect gas equation of state is \( P = \rho RT \) and \( e = P/(\gamma - 1) \rho \).

The advected species terms \( q_i \) are included in the formulation. The source term contains both the energetic mass injection and axisymmetric contributions. \( H_{in} \) is the total enthalpy per unit mass of the injected gas and \( S_p \) is its mass per unit volume. Species \( q_i \), equations are also included although these terms play no role and treated as passively advected quantities.

SOLVER

Godunov’ s solution scheme is applied, based on an exact Riemann solver [34]. The exact Riemann solver computes the arbitrary Riemann problem for 1D, x-split, time dependent Euler equations which is verified via the test cases suggested in literature [34,35]. The details of the Godunov solver will not be given but the moving boundary issues. The homogeneous part of Equation (1) can be written in integral form as:

\[
\frac{D}{Dt} \int_U d\mathcal{V} = \frac{d}{dt} \int_U d\mathcal{V} + \int_S (\vec{U} - \vec{V}_S) \cdot \vec{n} dS
\]

(2)

In moving grid finite volume solvers, flux term on the right hand side of (2) is computed using the same flux stencils for fixed grids by taking normal velocity vector relative to the moving interface [36], where \( \vec{V}_S \) is the cell boundary velocity. The properties vector \( \vec{U} \) is advanced to the next time step by discretization of the total derivative as follows:

\[
\vec{U}^{n+1}_{\mathcal{V}^{n+1}} - \vec{U}^n_{\mathcal{V}^n} = -\Delta t \sum_{N \text{ sides}} \vec{F}_M^{n+1} (\vec{U})
\]

(3)
A 2D Moving Boundary Cartesian Grid Solver for Internal Flow Fields of SPRM's

Where \( \bar{F}^M \) is any selected flux stencil for moving cell sides, which is compatible with the flux term in (1). For practical use, fluxes are applied to the initial control volume and then scaled according to Equation (4) after the new cell size is calculated.

\[
\begin{align*}
\bar{U}^{n+1} = \frac{\bar{V}}{\partial n} \left\{ \bar{U}^n + \frac{\Delta t}{\bar{V}^n} \left[ - \sum_{N \text{ sides}} \bar{F}^M (\bar{U}) \right] \right\}.
\end{align*}
\]

(4)

The update in (4) may not be practical for solution algorithms that are developed for fixed Cartesian grids. Since, for Cartesian cells that are initially solid and becoming cut cells, by the sweep of a moving boundary, initial volume \( \bar{V}^n \) is zero, \( \bar{U}^n_L \) with respect to neighbor cells should be suitably defined and approximated.

For these cells, property vector \( \bar{U}^{n+1} \) is completely generated by the accumulation of flux due to the moving boundary.

Since in the Cartesian case only cut sides are moving and this appears as a contact surface, same solver both for fixed and moving grids can be used. The total derivative on the right-hand side of (2), by Leibnitz rule can be represented in two terms:

\[
\frac{d}{dt} \int_\Omega \bar{U} d\Omega = \frac{\partial \bar{U}}{\partial t} d\Omega + \int_\partial \bar{U} \bar{V}_s \cdot \hat{n} dS.
\]

(5)

The first change term on the right-hand side of (5), which is the local part, provides the solution for constant cell volume. The state vector for this undeformed control volume can be calculated via the basic balance equation (6) by the procedure given below. Where fluxes for the moving cell boundaries are calculated using the left and right states without any transformation. Standard flux stencils are used.

\[
\begin{align*}
\bar{U}^{n+1}_i = \bar{U}^n_i + \frac{\Delta t}{\bar{V}^n_i} \left[ - \sum_{N \text{ sides}} \bar{F} (\bar{U}) \right].
\end{align*}
\]

(6)

Riemann solution at the cell side, \( RP(\bar{U}^n_i, \bar{U}^{n+1}_i) \) provides \( \tilde{F}_i \). The state variables are rotated to the direction of x-splitting. Calculated Riemann flux is back rotated to the original direction before it is used in the balance equation (1).

Time step is selected in order to prevent wave interaction between different cells.

\[
\Delta t = \frac{CFL \Delta x}{S^n_{\text{max}}}
\]

(7)

CFL number acts as a factor of safety. An estimate for maximum wave speed is given as:

\[
S^n_{\text{max}} = \max \left\{ u^n_{i,j} + a^n_{i,j} \right\}
\]

(8)

The second change term in (5), when multiplied by the time step, corresponds to the solution in the deformed part. It is calculated exactly by averaging discrete Riemann solutions over the retarding region. The initial left
and right states are again obtained from the wall boundary conditions. In this study, these solutions are termed as “extra solutions” and calculated for each solid/moving cut and kink cells.

The source term is handled by the method of splitting [34]; at each time-step, the intermediate computational solution of the homogeneous part of (1) is updated by solving the ordinary differential equation that contains the source term. For time integration forth ordered Runge-Kutta scheme is employed.

Inflow/Outflow, Transmissive and Moving Wall boundary conditions are applied to specify the fictitious boundary cell states. For supersonic outflow all of the three fictitious flow variables are extrapolated from the solution domain. For subsonic outflow the pressure is the only variable that is specified. Single neighbor cell states are used in the extrapolation routine, for geometries involving high curvatures procedures given in [43] are applied. The moving wall boundary condition is derived with respect to the observer moving with the quasi-steady boundary velocity as given below in the direction of x-splitting.

\[
\begin{align*}
\rho_R &= \rho_L \\
u_R &= -u_L + 2 \left( \bar{V}_s \right)_n \\
P_R &= P_L \\
v_R &= v_L \\
q_{1R} &= q_{1L}
\end{align*}
\]

\[(9)\]

**CARTESIAN GRID GENERATION ROUTINES**

Cartesian Grid Solvers does not require a separate grid generator program. However, when the aim is arbitrary complex geometries, generation of the template grids and finding local intersections becomes a complicated geometric task. An addition to this is the complexity brought by the boundary movement in the solution domain.

Unlike boundary-conforming grids where global remeshing, at each time step alters the positions of grid points [13,14], Cartesian grid points are stationary. Thus for Cartesian control volumes that are not cut by a moving boundary, any geometric conservation law or a Jacobian transformation that cares for the time dependent grid movement is not applied.

The solution procedure is given in Figure 1. The important routines for moving boundary handling and Cartesian grid generation will be discussed next.
Template and Cartesian Grid Geometry Definitions

The solution domain is rectangular. Given its length and width, square grids are generated inside. The solid and moving walls are defined as line segments in this solution template. Since the domain is rectangular and the cells are square, the number of grid lines in x- and y-directions are dependent on each other. In the code, the minimum number of grids in each direction are calculated, if an extra fineness is required the number of grids are increased in both directions with the same ratio. Integer arithmetic is performed during these calculations.

Moving and Stationary Boundaries

A stream is a combination of solid and moving curves, Figure 2. If there are no moving walls and all the geometry that defines fluid boundaries are solid, then each solid wall is assigned as a new stream. If there are moving walls, new streams are generated by tracing moving/solid curves. “A stream can form a closed loop or start and end at a template boundary” is the basic rule. Until this rule is satisfied, each trimmed/extended moving curve is traced first in its start direction and then towards its end. During this trace, intersections with other curves will be detected. Each new detected curve during this trace is kept in the order as a member of the generated stream.
A 2D Moving Boundary Cartesian Grid Solver for Internal Flow Fields of SPRM's

Cell Types and Intersections

The basic geometry, which defines the variables used in the algorithm, is presented in Figure 2. For each stream their line segments are searched for an intersection with vertical and horizontal grid lines. If an intersection is found, intersection point and its type (moving or solid) is stored in the cell structure relative to cell coordinates. The coordinates should be specified relative to square cells because of the accuracy considerations. This algorithm is different than the one proposed in [17]. In that study streams were traced and cell coordinates defined as integer variables taking discrete values to overcome the accuracy problem.

The intersection routine considers the sense of each line segment and covers different cases including specific orientations of segments that are parallel to Cartesian grid lines. Two intersections are allowed and typical for each Cartesian cell. If more than two intersections are found, their positions are stored for degenerate cell considerations.

For geometries involving moving and solid boundaries, there are five basic cell types. These are: full flow, solid, cut-solid, cut-move and kink cells, Figure 3. Cut cells contain a single curve segment, which may be moving or stationary. Kink cells are cut cells where a solid curve ends and a moving curve starts or vice versa. The same convention also holds for segments inside cells: the solid part is on left, in the direction of curve parameter increase.
A 2D Moving Boundary Cartesian Grid Solver for Internal Flow Fields of SPRM's

Figure 3: Basic Cell Types (SOLID, FULFLOW, CUT-M/S, KINK).

In the solution domain, if arbitrary intersections with the input geometry are allowed, some cells that are not covered by the Cartesian solver may appear. These cells are named as Degenerate Cells in the Cartesian literature [17]. (A simple example is a cell with more than two intersections, Figure 2) By increasing grid size or slightly shifting the input geometry some of the problematic cases can be corrected, however such remedies work only for bodies that are not moving. For applications involving continuously changing shapes and offsetting, these problematic geometries must be identified in a systematic way and suitably approximated. Depending on the number of intersections in the degenerate Cartesian cell, geometrically possible problem types can be grouped into four. These problem topologies when the rotations and symmetries of the basic geometry are taken into account make a total of 208 cases that the solver distinguishes and corrects. Without considering these topologies an efficient moving body Cartesian solver is not possible.

In Cartesian grid generation there are important issues related to sub-cell types, solid/fluid cell distinguishing/marking, cell combinations however these will not be covered in this report. The focus will be on moving boundary handling approach.

State Interpolation/Assignment for New Domains Created by Retarding Moving Boundaries

When the boundary geometry of the problem is updated at each time step due to retarding boundary movement, “new domains” are created inside Cartesian cut cells. These domains whose solution is not yet known, does not exactly overlap with the “extra solutions” that are calculated during the solver phase. These “extra solutions” are valid for rectangular regions that are generated by the cut segments sweeping during the time step.

New cell center and area of the cell that is under focus determines if there is a created domain without a solution. The center position of this new solution domain is calculated. Nearest four rectangular extra solutions are then found, their state vectors are inverse-distance approximated and assigned as a solution to the new solution domain of the investigated cell. This procedure is exact for purely translating moving boundaries without rotation. Offset boundary movement is a translating motion in the surface normal direction, which occurs in surface reactions, evaporation or agglomeration.

Burnback

To demonstrate the geometry handling and moving wall-offsetting functions two burnback examples are presented, Figure 4. Geometric problems encountered during normal offsetting are discussed in [37]. Wall offset velocity is constant for both examples and one tenth of the Cylindrical Shell is plotted. For the case with 50x60 Cartesian grids the template of length ratio is 5/6. Due to its high convex curvatures this geometry may cause some problems in geometry dependent burnback codes [38]. With the current method it is observed that sharp grain corners are tolerably radiused.
TEST CASES

2D Advancing and Retarding Piston

Gas dynamics of the simple piston problem is a good test case to verify the moving boundary handling routines. The analytic solution obtained via characteristic methods for different piston velocities are available in the literature [39].

In this test case the chamber length is 10 and width is 4 meters. The moving piston head is placed at the left-hand side (at t=0. s piston is at x=1 m). Open-end boundary condition is specified at right end, all other template boundaries are reflective simulating closed chamber walls. The CFL number is 0.1. The plotted results are for 40x16 grids, also the calculations are repeated for a 80x32 grid size and the error in density is found to be 0.15% for the coarse grid. A relatively high piston velocity, 40 m/s is selected. The density contours are plotted when the generated wave is approximately at one-third of the solution domain, i.e. before the whole domain reaches to the piston velocity and to piston head state, Figure 5.
Figure 5: Density Flood Plots of Advancing/Retarding Piston for Different Orientations.

For comparison density flood plot for an 45° inclined wall retarding with the same velocity is also presented in Figure 5. Although template boundaries are transmissive, generated waves are not parallel to the inclined wall because of the unsymmetrical decrease of retarding wall length. In Figure 6, velocity vectors are plotted for an advancing 45° inclined wall.

Figure 6: Velocity Vectors for 45 degree Inclined Advancing Piston.
2D Motor Test Case (Steady End-Burning Motor of ONERA [40])

Grain geometry of the test case is end-burning. Propellant in the gas phase is injected from the head end. The test case is suggested by Ref [41] for solid propellant internal flow modeling studies. Results that are obtained by the Cartesian grid solver will be compared, with the converged results of the code SIERRA.

The SIERRA (ONERA/France) is a research code dedicated to internal flows inside solid rocket motors, especially to unsteady regimes. It is based on a cell centered, finite volume, explicit predictor-corrector MacCormack’s scheme, in its original unsplit form (1969). Artificial viscosity can be added in the form of second and forth order terms that are adapted to the flow by sensors reacting to the pressure and/or the velocity fields (Jameson’s type). The code is multi-block and can be applied to a variety of geometries. Several boundary conditions are available.

Besides the differences in the solver and numerical scheme, for this particular problem the grid methodology is SIERRA is different compared to the present work. The Cartesian grid results are obtained for two different grid sizes. For comparison these grids are plotted in Figure 7. The baseline solution is obtained using a high quality grid, which is continuously refined near the boundaries. Whereas in the Cartesian grid method nozzle wall region is composed of irregular cut cells. Besides, since many of them are combined cells, the effective grid size near the wall is lower.
In Table 1, grid sizes of the three solutions are compared. The number and density of grids at the throat of the nozzle is important for the quality of the solution. For fine grid case, Cartesian method generates 12 of 32 radial cells at the throat, whereas SIERRA solution utilizes all 16 grids at the throat.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Grid Size</th>
<th>Number of Radial Grids</th>
<th>Number of Grids at the Throat</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structured, Body-Fitted</td>
<td>99x16</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>Cartesian, Fine Mesh</td>
<td>192x32</td>
<td>32</td>
<td>12</td>
</tr>
<tr>
<td>Cartesian, Coarse Mesh</td>
<td>96x16</td>
<td>16</td>
<td>6</td>
</tr>
</tbody>
</table>

The head-end injection boundary condition used in SIERRA solution is different than the propellant injection boundary condition used in this study, which depends on the source terms of the governing equations, and discussed in [15]. In the current solution method, the injected propellant is introduced to the solution domain as a mass source from the cell center, whereas in SIERRA, injecting gas is specified as a characteristic subsonic inflow boundary condition. In both approaches, energy equation is used to calculate injection density, which depends on injecting mass flux, flame temperature of the propellant, specific heat ratio, gas constant and pressure. Pressure of the injecting gas is assumed to be same with the static pressure of the boundary cell. The injection velocity is obtained by dividing the injecting mass flux by the injection density. Supersonic outflow boundary conditions are applied for the fictitious cells at the nozzle exit.

Contour plots for Mach number and pressure are presented in Figure 8 and Figure 9 with the corresponding solutions of the SIERRA code. The difference in the Mach contours at the throat entrance is associated to the small differences in the nozzle shape modeled. Cartesian grid code generates a grid having a short straight throat section with straight wall elements and thus the wave phenomena at these discrete corner points are being modeled exactly. For a throat without a straight element the results approached each other, but the effect of discrete corners, due to cut grids, remained in the solution. This effect is found to be disappearing for very fine meshes.
Effect of approximating nozzle as discrete line elements is more evident from the contour plot of Pressure, Figure 9. The nozzle inlet stagnation region is affected most, since this region is directly related to the nozzle wall flow, where all its streamlines pass. As the grid size decreases the region of irregular cells that neighbors the nozzle wall boundaries also decrease which results a corresponding decrease in the size of the upstream region affected.
The stagnation pressure and density, inlet and exit Mach numbers can be calculated for comparison, Table 2. The results obtained by the IBSE2D code are also included which uses a cell vertex, finite difference, explicit multistage Runge-Kutta solution scheme based on the work of Jameson, Schmidt, and Turkel [41]. As seen in Table 2, although the stagnation values are predicted higher than the two axisymmetric codes, both parameters approach to their isentropic flow values as the grid size decreased.

Figure 9: Pressure Contours.
Table 2: Comparisons of Selected Quantities with the Analytical Results of Different Axisymmetric Codes

<table>
<thead>
<tr>
<th>Feature</th>
<th>Inlet Mach Number (center)</th>
<th>Exit Mach Number (center)</th>
<th>Stagnation Pressure (Pa)</th>
<th>Stagnation Density (kg/m$^3$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Isentropic</td>
<td>0.08314</td>
<td>2.460</td>
<td>129561.</td>
<td>0.12772</td>
</tr>
<tr>
<td>SIERRA [49]</td>
<td>0.08280</td>
<td>2.456</td>
<td>129048.</td>
<td>0.12728</td>
</tr>
<tr>
<td>IBSE2D [50]</td>
<td>0.08290</td>
<td>2.476</td>
<td>129470.</td>
<td>0.12770</td>
</tr>
<tr>
<td>2D Cartesian (Fine Grid)</td>
<td>0.08070</td>
<td>2.465</td>
<td>131443.</td>
<td>0.12954</td>
</tr>
<tr>
<td>2D Cartesian (Coarse Grid)</td>
<td>0.07920</td>
<td>2.385</td>
<td>136331.</td>
<td>0.13440</td>
</tr>
</tbody>
</table>

Both codes IBSE2D and SIERRA, that their results are compared in Table 2, are specially designed for internal motor applications. Moreover, exactly the same grid is used in the solutions obtained by them, with a high quality nozzle grid.

CONCLUSIONS AND FUTURE WORK

A practical 2D Euler code is put to use that is free from any classical meshing considerations. Usage of Cartesian grids is a growing popularity, however studies involving moving boundary problems in Cartesian grids are relatively few and limited in complexity. To accomplish this task, a systematic degenerate cell classification is implemented.

The governing equations are derived for deforming control volumes that is suitable for the fixed grid method of solution. This derivation only allows problems with translating and offsetting boundary movement.

Apart from the Euler solution, the code also generates burnback data as an alternative to existing burnback tools that may be geometry dependent for certain problems. Generation of burnback data is a special topic by itself. In this study its possibility is demonstrated using the algorithms that are written originally for Cartesian grid methodology. The generated burnback data should be compared quantitatively with the results from the existing codes. Another related future study is the three dimensional burnback. Consecutive 2D grain sections may offer a practical approach.

Adaptive mesh refinement at the boundaries will be worth the fairly involved coding work, which is currently being studied. Boundary mesh refinement is required if a viscous solution is desired. Usage of hybrid grids is a popular approach to this problem.

Another problem that is discussed is the approximation of curved solid wall boundaries as straight-line segments. These line segments produce non-smooth solution boundaries where the exact Riemann solver captures the exact wave phenomena at these corner points, which results a different flow field unless the mesh size is decreased considerably. Whereas in conventional structured grid solvers boundary geometries are continuous up to their second derivative. One remedy to this problem is to store cut cell curvatures besides the other cell attributes.
REFERENCES


SYMPOSIA DISCUSSION – PAPER NO: 30

Discusser's Name: Ron Derr

Question:
Will your flow model be coupled with a combustion instability analysis that assesses acoustic energy gains and losses?

Author's Name: H. Tugrul Tinaztepe

Author's Response:
Yes, but our colleague Dr. Keren Pekkar is on leave for his post-graduate studies. We intend to continue this work when he returns.
This page has been deliberately left blank

Page intentionnellement blanche