CONF-980610--

LA-UR-97-4630

Approved for public release;
distribution is unlimited.

53

Title: Rigel: An Interactive Structured Grid Generation System

Author(s): Hachfeld, William-XCM
Khamayseh, Ahmed-XCM
Hansen, Glen-XCM

RECEIVED
MAR 2 5 1998
OSTI

Submitted to: ASME/AIAA Thermophysics Conference
June ~~April~~ 1998
Albuquerque, NM

**Los Alamos**
NATIONAL LABORATORY

# DISCLAIMER

# RIGEL: AN INTERACTIVE STRUCTURED GRID GENERATION SYSTEM

**William D. Hachfeld, Ahmed K. Khamayseh, Glen A. Hansen**
Los Alamos National Laboratory, Los Alamos, NM 87545, U.S.A.

## ABSTRACT

An interactive structured grid generation application that facilitates the construction of complex, discretized, simulation models directly from the original CAD geometry specifications is presented. The application, named Rigel, reads physical model descriptions generated by modern CAD packages. Rigel includes a suite of interactive geometry editing functions to assist the user in the construction of a topologically-correct geometry from the original CAD specification. Once a topologically-correct geometry is created, an interactively-steered grid generation capability is provided to facilitate the construction of an appropriate discretization for the simulation. Grid quality enhancement is supported with the application of user-directed elliptic smoothing, refinement, and coarsening operators. After a grid is completed, various output filters are supplied to write an input file for the target simulation code. This paper is intended to provide an overview of the mechanics of this process and to highlight some of the novel algorithms and techniques employed.

## INTRODUCTION

In many industries, there are typically two broad groups of people involved in the design and evaluation of a product. One group consists of engineers and technicians who are concerned with the details of product design and manufacturing; the second group consists of analysts that evaluate the expected performance of the design. The design group typically uses specification tools such as CAD packages to specify components in sufficient detail to allow manufacturing of the parts (including dimensions and tolerances for the components). The analysis group employs simulation codes and/or experimental means to assess the design.

The tools used by these two groups are not generally compatible. CAD systems produce detailed representations of a given design, usually in terms of parametric surfaces or solid bodies. Simulation codes generally require an input specification of a discrete grid and isotopic conditions based on the geometries in the CAD model. To generate a grid input specification based directly on the output of a CAD tool requires a sequence of general operations.

The process of defining a simulation (creating an input file) may be viewed as a forward process of generating a grid based on a geometrical description of the objects to be modeled. Typically, these objects are in the form of a solid model CAD description, created by a commercial package. It is necessary to output the geometric description from the CAD package in a "clean" form, or provide a post-processing mechanism to "clean" and simplify the CAD geometrical description in preparation for the grid generation step.

The emphasis on a "clean" geometry is due to two phenomena: 1) occasionally there are errors and flaws in the CAD geometrical data that remain as artifacts or are introduced during the output phase, and 2) there is excessive detail (such as screws, clips, etc.) that are irrelevant and complicate the grid generation and simulation processes.

Given a "clean" geometry, it is necessary to perform topological operations to prepare the description for the grid generation phase. In the case of structured planar grid generation, it is necessary to manipulate the geometrical entities to form four-sided logical regions that remain an accurate description of the original geometry. Furthermore, it is necessary to enforce that the local coordinate systems of the regions are consistent with a global coordinate system. At this point, it is possible to begin the process of generating a grid; the boundaries of which are determined based upon the description of the geometries contained within the domain. During the simulation setup process, it is also necessary to specify or obtain from the CAD description other information that fully specifies the simulation, such as material information and properties that describe boundary and initial conditions for the simulation.

Clearly, from a user's standpoint it is desirable to perform the simulation setup process in a consistent, automated fashion. Those functions that require user input should be accommodated through an intuitive user interface in a general but not complicated manner. This suggests that those functions that cannot be performed entirely within the CAD tool should be isolated within a single user interface. Given these characteristics, it is possible for an analyst to run a simulation of the design quickly, without considerable domain-specific knowledge of the setup process.

Several commercial grid generation tools are available that address these concerns for most applications requiring unstructured and multi-block structures meshes. Unfortunately, the need exists to generate single logical meshes on complex multi-component domains while providing specific grid quality features such as grid-cell boundary orthogonality and impedance matching across component interfaces.

Rigel was designed to be a fully-interactive grid generation package for creating a two-dimensional logical mesh on multi-component domains. Geometry files are read into the application, with the curves and surfaces that describe the geometry rendered in a display window. The user can interactively translate, scale, and rotate the view of the geometry as well as select individual geometric entities using the mouse. These selected entities can then be hidden, deleted, or otherwise manipulated. Topological tools provide the user a capability for removing unwanted features, repairing defects, and otherwise building a suitable geometry for grid generation. Grid generation tools include capabilities for grid quality analysis and enhancement. This overview concludes with a brief discussion of Rigel's future and the challenges that lie ahead.

## GEOMETRY MODIFICATION AND TOPOLOGY TOOLS

Modern CAD packages produce detailed information about the parts being designed. Internally, most have a custom database engine for storing the information and a specialized geometry engine for manipulating it. It is undesirable to require a user to re-enter the geometry information manually, both due to efficiency and model accuracy concerns. A method by which the CAD package's database could be converted into a form usable in Rigel was required. The conversion process that was developed has three major steps.

### CAD File Translation

Since the database files utilized by many CAD packages are proprietary, conversion to a more standardized format was required. Rigel obtains geometry information using an IGES (Initial Graphics Exchange Specifications) format input file, which is an output format supported by most CAD packages. Within the CAD package, the user writes the necessary geometry information to an IGES file, which contains descriptions of 3D surfaces, 3D solids, or 2D cross-section curves.

To support Rigel's current grid generation capability, this data consists of either planar surfaces or planar curves. Rigel is not limited to processing planar geometry; the database model and rendering engines are designed for 3D entities. However, 3D grid generation capability remains to be developed in a future version.

Reading an IGES file into Rigel requires selecting the appropriate file name from a file selection window. Multiple IGES files may be read in succession and appended together, facilitating support of CAD packages that write each model part as a separate IGES file.

Since IGES supports a range of geometric entities, different CAD packages write IGES files in different ways. To accommodate these potentially different structures, Rigel converts all IGES geometric entities into one of three basic forms during the input process; vertices, non-uniform rational B-spline (NURB) curves, or NURB surfaces.

NURBS [Piegl and Tiller, 1995] were selected as Rigel's preferred internal object model due to their ability to accurately represent a variety of curve and surface forms. Since nearly any surface or object can be converted into NURB form, Rigel can handle surface types ranging from simple spheres to complex (folded and twisted) surfaces. The NURB data model also has the advantage of presenting a consistently-parameterized representation of all surfaces to the grid generation algorithms.

### Geometry Repair and Surface Creation

Unfortunately, the conversion from CAD database to IGES file results in a topologically incomplete specification. A CAD model that can be converted into geometric entities directly suitable for the grid generation process is not often encountered. Typically, further processing is required to prepare the geometries for grid generation.

The CAD model to IGES conversion process often creates duplicate curves and surfaces. As the CAD package traverses its internal model, certain elements of the model geometry may be encountered multiple times. If the package does not specifically disallow the output of duplicate entities when writing the IGES file, these entities will be written to the file multiple times. Rigel automatically removes these duplicates by comparing entities of identical types against each other (vertices to vertices, curves to curves, and surfaces to surfaces). A distinct algorithm is used for the comparison in each case. Vertices require only a simple distance test. Curves and surfaces require a multi-level comparison that bypasses distinctly different entities immediately, saving more complex coincidence tests for the entities that differ only slightly. Any duplicates detected, within a given tolerance, are removed from Rigel's internal object model.

After the elimination of duplicate geometry entities, the data is not yet suitable for grid generation. In many cases, the output of planar cross-sections of solid models frequently results in cross-section curves containing gaps and mis-matches. Rigel provides capabilities to correct these defects, interactively.

Two curves separated by a gap can be interactively selected and blended together. Individual curves may be selected and cut at a specific point on the curve, or two curves may be split at an intersection point. Two or more curves may be selected and glued together to form a single curve. By using combinations of these operations, one can eliminate gaps from the original geometry and assemble consistent curves that accurately represent the original CAD model. These capabilities may also be used to remove small-scale details in the geometry that are not desirable in the final grid.

The above steps are required for any typical model-based grid generation process. The specific type of grid being generated by Rigel requires an additional geometry processing step. Planar material regions, as specified within the CAD model, are typically defined by boundary curves. The single logical grid generation capability requires that these boundary curves specify one or more NURB surfaces that overlay each planar material region.

Surface creation operations are provided to accommodate these requirements. Surfaces may be constructed by ruling, sweeping, or revolving existing curves. Coon's patches [Piegl and Tiller, 1995] may be constructed between four connected curves, or three connected curves and a vertex. In addition to the actual construction operators, additional operations allow the user to intersect, split, blend, and join the constructed surfaces, as necessary, to form the required surfaces.

### Topology Correction

The final step prior to proceeding with grid generation is to correct the topology of the NURB surfaces that define the material regions. Each NURB surface has an orientation consisting of the U and V parameterization vectors as well as a normal to the surface. The effective generation of a single (logically mapped) planar grid overlaying the material regions requires that the orientation of each surface be aligned with the other surfaces, consistent with a global coordinate system. The normal vectors of all surfaces must also be oriented in the same direction. Additionally, all surfaces representing material regions must have U and V parameterization vectors aligned in roughly the same direction (within approximately 30 degrees).

The user must perform these alignments manually, but Rigel provides several interactive diagnostics and operators to achieve the alignment. U, V, and normal vectors for each surface may be displayed simultaneously or individually. Operators that flip the direction of each of the three vectors, and swap the U and V vectors, are also provided.

These tools may be used to orient the surfaces correctly in almost all cases. Occasionally a situation arises where the UV vectors of a surface are such that alignment with the vectors of the other surfaces is not possible. In this situation, the user may employ the operations described in the previous section to cut the surface boundary curves and reform the surface to result in a better UV alignment.

The final step to be performed prior to grid generation is the creation of a "background surface" that encloses all the other regions representing material regions. This surface is used to define the extent of the single logical grid to be generated. In some cases, this surface will already exist naturally from the geometry. If this surface doesn't exist, the user can use an operation in Rigel's topology tool set to automatically create such a surface.

## THE GRID GENERATION PROCESS

Once the user has used Rigel's geometry modification and topology tools to construct a topologically-correct surface geometry, it is possible to interactively generate a grid. The actual process of grid generation is entirely user-driven. Most of the grid operators that Rigel provides can be applied at any time and in any order. However, in order to begin the process, the user typically starts by generating a surface grid on the topologically-required background surface.

### Surface Grid Generation

At this point, it is essential to note an important property of NURB surfaces. Since the (u, v) computational domain parameterizes these surfaces, they contain exactly four boundary curves. One curve is associated with each of the u_min, u_max, v_min, and v_max values. It is also possible for one or more of these boundary curves to degenerate to a point. This property leads to a consistent mapping of the structured grid's I coordinate axis onto the surface's 'u' parameterization coordinate axis, and the J coordinate axis onto the surface 'v' parameterization coordinate axis, respectively.

The user begins generating a surface grid by first placing grid points along each of the surface's four boundary curves. A parameter controlling the number of grid points to be placed is entered. Parameters controlling the spacing are also specified. Points may be placed uniformly in arc length, concentrated at areas of high curvature, or concentrated at the beginning or end of the boundary curve. These spacing parameters are not mutually exclusive; the mix of the allowed spacing options is user controllable.

Once user has selected the desired grid generation parameters, one or more boundary curves are selected and the grid generator is subsequently applied to those curves. The grid for each boundary curve is generated in a single pass by an algorithm described in [Khamayseh and Kuprat, 1997].

For each surface boundary curve selected by the user, Rigel also examines the geometry for topologically opposite curves (of which there is always at least one). The constraints of structured grid generation require that an identical number of grid points be placed on these opposite curves. Rigel enforces this constraint automatically by propagating the requested number of points to all topologically opposite curves.

Grids are generated on each of the four boundaries of a given surface. Once these curve grids exist, the user may then select the surface and generate a surface grid. Grid data from each boundary is copied into the surface grid structure. Interior points are then computed using trans-finite interpolation (TFI) [Thompson et al., 1985] and [Knupp and Steinberg, 1993].

Both curve and surface grids are visualized in Rigel's rendering window immediately upon generation. At any point the user may select grids and delete them. Curve grids may also be modified (regenerated) at any time. When a curve grid is regenerated, all surface grids that use this curve grid are automatically deleted and must be regenerated.

### Surface Grid Merge

Using the tools described above the user can generate a grid on each of the surfaces of the material region. The user may write the generated grids to a file at this point if multi-patch structured grids are desired. To generate a single structured grid spanning the entire geometry, additional steps are necessary.

To generate a single structured grid, the user successively applies a merge operator to the background grid and one of the grids representing a material region. First, the grid to merge into the background grid is selected. Then, an IJ rectangular merge region within the background grid is selected. Users may select this merge region manually or it will be selected automatically by Rigel. This merge region specifies that portion of the background that is to be occupied by the merged grid. Once the merge grid and the merge region have been selected, the merge operator is applied.

The operator begins by regenerating curve grids on the surface that is associated with the merge grid. The number of points to be generated on each of the four boundary curves is determined by the logical IJ size of the selected region on the background grid. The points in the regenerated curve grids are then copied into the background grid region. After copying these points, the merge operator uses the TFI algorithm to regenerate the interior points within the selected background grid region. Finally, a smoothing operator is applied to the background grid in order to eliminate any folding that may have resulted from the merge.

After the two grid regions have been successfully merged, the user is presented with a new background grid that now includes a region that has been boundary fitted to the merged grid. The grid edges making up the boundary curve have been marked to enforce that subsequent grid enhancement operations will preserve the boundary fitted quality at the curve. Visually these edges are also drawn in a highlighted color to allow the user to distinguish them from interior grid edges.

### Grid Diagnostics and Enhancement

At any point during the grid generation or merge process, a variety of grid diagnostics and enhancement operators may be applied to a grid. By allowing these operations to be applied dynamically, the user is able to steer the quality of the grid during the generation process.

A grid's quality can be assessed visually at any time using several different quality metrics. Individual grid cells may be shaded based on metrics including aspect ratios, skew angles, grid smoothness, and grid cell folding. Cells may also be selected to display a local value of a given quality metric. Grid cell shading is performed dynamically to illustrate the effects of various grid enhancement operators on mesh quality.

Grid quality enhancement is supported with the application of user-directed elliptic smoothing, boundary smoothing, refinement, coarsening, and thinning operations. The elliptic smoother [Khamayseh et al., 1997] moves non-boundary interior grid points to achieve a smoother spacing of interior points. Boundary smoothing moves boundary points (both interior and exterior) along the original geometry to achieve roughly equal physical spacing between those points.

The refinement and coarsening operations are used to insert or remove grid lines in the domain, allowing the user to manipulate the local grid density to place mesh points in areas of greater interest or detail. Refinement allows the insertion of additional mesh points into the logical domain, placed physically in the grid by either interpolating based on the original geometry (when the inserted point lies on a boundary) or with the use of TFI in the interior region. Coarsening removes grid lines from the logical domain of the grid without removing boundary constraints. Each of these operators affects a user-specified region of the grid.

Refinement in the context of a logically structured grid is limited by the requirement that grid lines must extend across the entire length or width of the grid. For those codes that support unstructured cells, Rigel provides an operator to thin grid lines. By sweeping across the selected region in the I or J direction by a user-specified stride, edges in the appropriate direction are marked as "unused". These unused edges are removed from the grid, resulting in a locally unstructured grid region within the globally structured mesh. Constraints are imposed on these unstructured cells enforcing that the composed edges always lie on a straight line.

## SOFTWARE MODEL

Rigel is implemented based on an object-oriented design in C++, targeted for the SGI IRIX 6.x operating system. Ports to additional Unix platforms are planned for the future, and a Windows NT version is also under consideration.

To reduce the amount of time necessary to generate the graphical user interface (GUI), a commercial GUI builder for X11/Motif was utilized. This tool allowed the development team to interactively assemble the GUI and automatically generate the required C++ code to implement the interface.

Rigel uses OpenGL to draw geometric objects and grids in the rendering window. OpenGL is a standard application programming interface (API) that allows for cross-platform rendering of 3D scenes on any machine that provides an OpenGL implementation. Using OpenGL also allows Rigel to take advantage of the hardware-assisted 3D rendering capability present on most high-end SGI workstations.

The Atlas geometry kernel written at the National Science Foundation Engineering Research Center for Computation Field Simulation (NSF/ERC) of Mississippi State University (MSU) has been equally valuable in the development effort. Atlas is an outgrowth of the National Grid Project (NGP) [Thompson, 1992] and provides all the necessary NURB evaluation and manipulation functions in Rigel. Also included with the Atlas geometry kernel is a library for reading and writing IGES CAD files. Using these libraries significantly reduced the time required to implement the Rigel software and allowed the team to focus on the grid generation aspects of the tool.

## CONCLUSIONS AND FUTURE CHALLENGES

An overview of the mechanics and methods used within Rigel to read, repair, modify, and generate grids based on CAD geometries was discussed in the preceding sections. The current version of Rigel is a complete grid generation application, but several challenges and improvements have yet to be addressed. In the future, the authors plan to further automate the geometry, topology, and grid generation facilities within the tool. Automatic correction of surface topology and the ability to merge multiple surfaces into the background grid simultaneously are important enhancements. Additional grid generation capability is under development, including the ability to impose orthogonality on the boundaries and interior of the grid, and impose an impedance matching capability to interior interfaces.

Longer-term enhancements will focus on adding additional types of grid generation to the current framework. This will include a capability for 3D structured and unstructured grid generation, and an enhanced linkage to CAD applications to allow geometrical modifications based on simulation results to propagate back to the CAD model. Additionally, a tighter coupling with the CAD package may allow use of the geometry functions of the CAD tool, hopefully circumventing much of the geometry repair operations that are currently required.

## REFERENCES

A. Khamayseh, A. Kuprat, "CRC Handbook of Grid Generation, Surface Grid Generation Systems", CRC Press, Boca Raton, Florida, 1997

A. Khamayseh, A. Kuprat, W. Mastin, "CRC Handbook of Grid Generation, Boundary Orthogonality in Elliptic Grid Generation", CRC Press, Boca Raton, Florida, 1997

P. Knupp, S. Steinberg, "Fundamentals of Grid Generation", CRC Press, Boca Raton, Florida, 1997

L. Piegl, W. Tiller, "The NURBS Book", Springer-Verlag, Berlin, Germany, 1995

J. F. Thompson, "The National Grid Project", *Computer Systems Engineering*, 1992, Vol. 3, Nos. 1-4, pp. 393-399

J.F. Thompson, Z.U.A., C.W. Mastin, "Numerical Grid Generation: Foundations and Applications", North-Holland, New York, New York, 1985

M98004661

Report Number (14) LA-UR--97-4630

CONF-980610--

Publ. Date (11) 199802

Sponsor Code (18) DOE/DP, XF

UC Category (19) UC-700, DOE/ER

DOE