CFD Research Corporation
215 Wynn Dr. • Huntsville, Alabama 35805 • Tel.: (256) 726-4800 • FAX: (256) 726-4806 • info@cfdrc.com

AN AUTOMATED VISCOUS UNSTRUCTURED ADAPTIVE CARTESIAN GRID GENERATION METHOD FOR COMPLEX GEOMETRIES

Phase I Option
Final Report

by

Dr. Z.J. Wang

CFD Research Corporation
215 Wynn Drive
Huntsville, AL 35805

August 1998

CFDRC Project: 4870-O/1

for

Naval Air Warfare Center - Aircraft Division
22541 Millstone Rd - Bldg 2187
Patuxent River, MD 20670-5304

Technical Liaison: Mr. Darren Grove
Contracting Officer: Dr. Christopher Veith

Phase I Option Contract: N68335-97-C-0027

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	28 AUG 98	FINAL 08 MAY 98 - 06 AUG 98

4. TITLE AND SUBTITLE
An Automated Viscoius Unstructured Adaptive Cartesian Grid Generation Method for Complex Geometries

5. FUNDING NUMBERS
N68335-97-C-0027

6. AUTHOR(S)
Dr. Z.J. Wang

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
CFD Research Corporation
215 Wynn Drive
Huntsville, AL 35805

8. PERFORMING ORGANIZATION REPORT NUMBER
CFDRC Report Number:
4870-O/1

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Naval Air Warfare Center-Aircraft Division
22541 Millstone Road, Blud 2187
Patuxent River, MD 20670-5304

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
None.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This Phase I Option final report contains two major work items carried out during the Option period: redefinition of the Phase II Statement of Work and the preliminary design for the Grid Generator and Grid Adaptor to be produced in Phase II.

14. SUBJECT TERMS
Grid Generation, Viscous, Cartesian

15. NUMBER OF PAGES
13

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

1. INTRODUCTION

The basic Phase I project of this option was performed from January 1997 through June 1997. The feasibility of an automated viscous Cartesian grid generation method was successfully demonstrated in the basic Phase I study. Several simple to fairly complex geometries were successfully used to demonstrate this methodology, including a generic car body, wing-body combination, a missile geometry with fins, and the F-16 aircraft with stores. The overall methodology was proved very fast and robust. **For example, it took only about 8 minutes on a SGI O2 workstation to generate a viscous Cartesian grid around the complete F-16 geometry with over half a million cells.**

A detailed description of the technical approach was provided in the Phase I Final Report. Some of the key advantages and innovations of the developed approach are listed below.

- Automatic grid generation for complex geometries;
- High-resolution for viscous boundary layers through level-distance cutting;
- No cell cutting, Convex cells;
- Automatic feature detection and recovery;
- Near optimum search operations;
- Automatic flow-based grid adaptations; and
- Amenable for parallel processing.

The overall objective of the project (Phase I and Phase II) is to develop and demonstrate an automatic viscous Cartesian grid generator for arbitrarily complex geometries. In Phase I, the methodology is developed and demonstrated for relatively simple 3D geometries. Specific objectives for the Phase I study are listed below.

- Establish an Octree data structure for the Cartesian grid, and a suitable data structure for the directionally-stretched viscous layer grids;
- Develop efficient algorithms to block Cartesian cells inside or intersected by the body surface, and smooth the exposed Cartesian grid front;
- Develop robust and efficient algorithms to project nodes from the Cartesian grid to the body surface, while preserving critical geometric features;
- Adapt an existing viscous flow solver for 3D grids supporting arbitrary polyhedra; and
- Demonstrate the grid generation method with a wing-body and F-16, and carry out sample flow field calculations.

All the above objectives were successfully achieved in the basic Phase I program. In the original Phase I proposal, the following objective was set for the Phase I Option.

- Demonstrate the viscous Cartesian grid generation methodology for more complex geometries including the following cases:

  a. The combination of a wing and a fuselage; and
  b. Wing/Missile combination.

As a matter of fact, the objective of the Phase I Option was achieved in the basic Phase I stage due to the fierce competition. The most complex case tackled under the Phase I stage is the complete F-16 aircraft, which is far more complex than any geometry proposed. The sample viscous Cartesian grid for the F-16 geometry is shown in Figure 1, and a cutting plane through the mesh is shown in Figure 2. The objective of the Phase I Option is then to make preparations for the Phase II project, in particular, to design an object oriented architecture which is capable of being easily extendable and maintainable. Major work items are then described in the following sections.

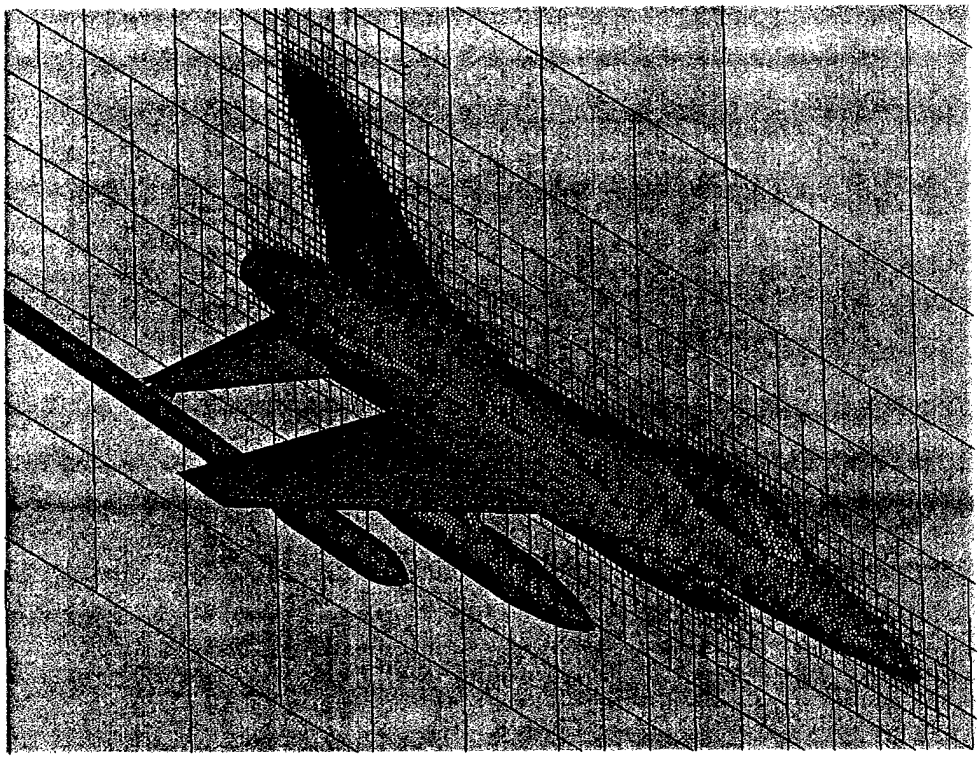

Figure 1. The Viscous Cartesian Grid for the Complete F-16 with Stores

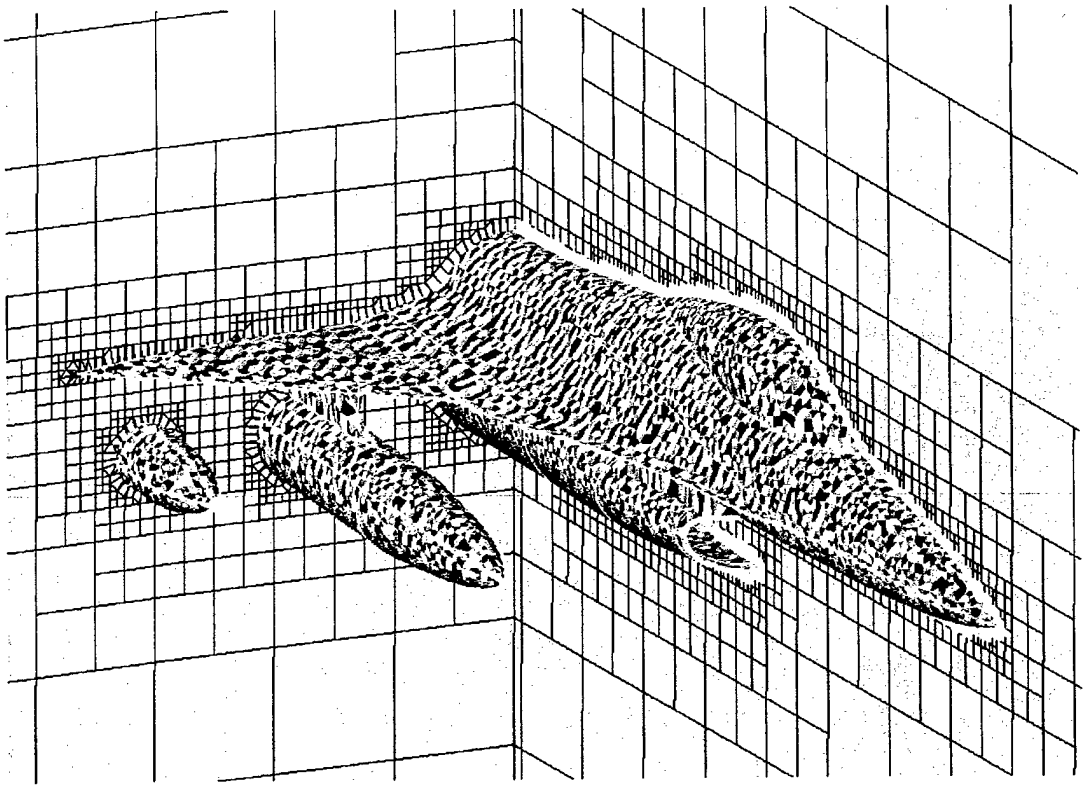

Figure 2. X-Cut Plane through the Viscous Cartesian Grid of the Complete F-16

2. REDEFINITION OF THE STATEMENT OF WORK

The original Phase II proposal was submitted in June 1997. In that proposal, CFDRC decided to subcontract part of the work to Lockheed Martin. However, Lockheed Martin started its own development work in the beginning of this year since CFDRC was not awarded the Phase II in the first round. In order to coordinate development effort and discuss a possible new Statement of Work (SOW), a joint meeting between the Navy, Lockhead Martin and CFDRC was called by the Navy program monitor, Mr. Darren Grove and held on June 11th at NAWC/AD, Patuxent River. Mr. Sami Habchi and Dr. Z.J. Wang from CFDRC attended the meeting. The purpose of the meeting was to re-define the statement of work based on possible progresses made by Lockhead Martin in implementing the viscous Cartesian grid methodology into their own Splitflow code. During the meeting, the three sides made presentations from their own perspectives. I presented a revised statement of work. We discussed the work items in detail, and decided two validation and demonstration cases: diamond wing and F/A-18 aircraft. We also discussed possible subcontract work from CFDRC to Lockhead Martin. The meeting turned out to be very productive. We achieved mutual understanding, and agreed that the following work items will be subcontracted to Lockhead Martin from CFDRC subject to agreeable sub-contract budget:

1. Consultation on adaptation criteria, and grid smoothing laws;

2. Consultation on the implementation of omni-tree into CFDRC's viscous Cartesian grid generator; and

3. Demonstration of CFDRC's grid generator and grid adaptor with F/A 18 using Splitflow.

After the meeting, CFDRC and Lockhead Martin discussed the budget on the sub-contract work. Several telephone conversations and face-to-face meetings were held between CFDRC and Lockhead Martin to discuss the issue based upon the agreed work scope for the subcontract work. It turned out that the two sides had very different expectations of the budget, and the gap was too wide to be bridged. After trying its best, CFDRC decided to proceed alone. A revised budget and statement of work was then produced by CFDRC, and discussed with the Technical Monitor, Mr. Darren Grove.

2.1 Statement of Work

The following is the Statement of Work for Navy SBIR Phase II entitled, "An Automated Viscous Unstructured Adaptive Cartesian Grid Generation Method for Complex Geometries."

2.1.1. Technical Objectives

The overall objective is to develop an automated viscous grid generation, and flow-based grid adaptation capability for arbitrarily complex geometries. Specific objectives for the Phase II are the followings:

1. To develop a stand-alone automatic viscous Cartesian grid generator which can handle complex geometries such as complete F/A-18 aircraft with stores;

2. To develop a stand-alone grid adaptor which can adapt the viscous Cartesian grid according to the computed flow solution;

3. To interface developed tools with selected CFD codes such as Navy's PUMA code and commercial CFD codes such as CFD-FASTRAN;

4. To validate and demonstrate the grid generator and adaptor for a complete F/A-18 geometry including fuel tank and stores, and the diamond wing configuration.

2.1.2. Task Description

The Phase II project will be divided into the following Tasks:

Task 1. Support "water tight" STL, PLOT3D, FAST and ACAD Geometry Formats

- STL, FAST, PLOT3D and ACAD readers will be developed;
- Possible geometric problems will be reported;
- All the critical geometric features will be detected and will be preserved in the viscous Cartesian grid generation process.
- A module to project an arbitrary point in a 3D space to the surface will also be developed;

Task 2. Test and Refine the Viscous Cartesian Grid Methodology

- Point, line, plane and box sources will be added to control Cartesian cell distributions.
- The Cartesian grid will be adapted based on local curvatures;
- The viscous layer spacing will be determined according to the Reynolds number and y+.
- The Cartesian front will be further smoothed.
- Narrow gaps will be detected and handled automatically.

Task 3. Develop Direct Interfaces to Selected CFD Codes

- Write the grid in selected CFD solver formats including PUMA, CFD-FASTRAN, GUST and USM3D;
- Develop a mechanism to specify boundary conditions;
- Dump boundary conditions to interface with selected codes.

Task 4. Implement 2^nTree (Omnitree) Data Structure to Support Directional Grid Adaptations

- Develop a 2^n tree adaptive Cartesian grid generator;
- Enforce grid smoothness;
- Adapt the Cartesian grid according to local curvature;
- Generate the Cartesian grid front;
- Smooth Cartesian grid front.

Task 5. Develop a Stand-Alone Grid Adaptor Supporting both Octree and 2^nTree Data Structure

- Evaluate several popular grid adaptation criteria;
- Use directional criteria for 2^n tree based grid adaptations;
- Adapt viscous layer grids based on the Reynolds number and y+ values;
- Interpolate field solutions from the old to the new grid;
- Dump the restart file and the new grid file.

Task 6. Develop a Grid Quality Checking Conversion Module

- Evaluate several popular grid quality criteria;
- Convert the final viscous Cartesian grid into all tetrahedral or mixed grids (prisms, pyramids, hex, tests etc.);

Task 7. Develop a Graphical User Interface for the Viscous Cartesian Grid Generator

- Implement x and OpenGL for viewing of geometry (triangulated surface), and x-, y-, z-cutting planes through both the adaptive Cartesian and viscous layer grids;
- Define control parameters;
- Define boundary conditions;
- Perform visual debugging.

Task 8. Verification and Demonstration Studies

- Verify each development with relatively simple geometries;
- Demonstrate the viscous Cartesian grid generator, grid adaptor with the case of diamond wing geometry (to be supplied by the Navy) and flow solver PUMA or CFD-FASTRAN;

Task 9. Ensure Post-Processing Compatibility with FAST, Ensight, Fieldview and CFD-VIEW

Phase II Option Validation and Demonstration of Developed Technology with Full F-18

The Phase II option will be divided into the following sub-tasks

a. Validation with CFD-FASTRAN

- Generate a viscous Cartesian grid for the full F-18 geometry with fuel tank and JDAM store;
- CFDRC's commercial CFD solver CFD-FASTRAN will be used first to validate the grid generator and adaptor. CFD-FASTRAN has very similar numerical algorithms to Navy's PUMA code, and has been extensively validated for a variety of flow problems. Recently, Chimera overlapped grids were used simulate flow around the F-18 with fuel tank and JDAM store.
- Run the grid adaptor to adapt the grid according to flow features, and iterate for several times;
- Compare the computational results with experimental data and results with Chimera grid method.

b. Demonstration with Navy's PUMA Code

- The same initial viscous Cartesian grid will be decomposed into a mixed grid (if necessary) and interface with PUMA;
- Appropriate boundary conditions will be defined for PUMA;
- Run PUMA to compute the flow fields;
- Run the grid adaptor to adapt the grid according to flow features, and iterate for several times;
- Compare the computational results with experimental data and CFD-FASTRAN simulations.

c. Demonstration with USM3D

- Interface generated viscous Cartesian F-18 grid with USM3D using appropriate grid format;
- Define appropriate boundary conditions for USM3D;
- Run USM3D to compute the flow fields;
- Run the grid adaptor to adapt the grid according to flow features, and iterate for several times;

- Compare the computational results with experimental data, results from CFD-FASTRAN and PUMA.

2.1.3. Deliverables

1. CFDRC will submit quarterly progress reports;
2. CFDRC will update the source code delivered to the Navy every six months;
3. A software user's manual with tutorials including one simple geometry and one complex geometry such as the diamond wing;
4. Final source code (both batch and GUI) in C++ with make files for SGI (R4000, R8000, R10000) and IBM (generic)
5. A final report

3. DATA STRUCTURE AND ARCHITECTURE DESIGN

Several internal project meetings were called to discuss various challenges of the project. CFDRC has several Cartesian grid related projects going on in several different groups (e.g., CFD-GEOM, SuperMesh, etc). By working together and sharing developed technologies, we can produce better products for out clients. It is planned that engineers from different groups will jointly work together on the Navy Phase II. It is also decided that an object-oriented approach will be taken and the C++ programming language will be used for this Phase II project. This approach will allow parallel development from different groups to be carried out. Furthermore, the developed technology will find many more applications than otherwise possible.

We have started the preliminary design process. During our several design meetings, several critical design issues were raised. One of them was concerning Octree versus Omnitree. After extensive discussions and debates, we decided to support only Omnitree with Octree as the subset. We believe a single data structure will eliminate many potential duplications. Another design issue was whether we should use the Standard Template Library (STL) embedded in C++. There are advantages and disadvantages in using STL. One major criticism was that the code may become very large if we use STL. The major advantages, of coarse, include the generality and efficiency built into a standard library. Again, we decided to use STL after extensive discussions.

Some of the major classes have also been defined. For example, the geometry class and Omnitree class, etc. Several of them are shown below.

```
class CAGeometry

{
public:

    virtual void            Scale (Real fact) {}
    virtual void            Translate (Real dx, Real dy, Real dz) {}
    virtual void            Rotate (Real x, Real y, Real z, Real lx, Real ly, Real lz,
                            Real angle) {}
    virtual void            BoundingBox () {}
    virtual CAPoint3D       *Project (Real x, Real y, Real z) { return 0;}
    virtual Real            *GetMin () {return 0;}
    virtual Real            *GetMax () {return 0;}

};


class CATriSurface : public CAGeometry
{
private:
    int     nTris, nNodes;          // no of triangles
```

```
    Real     *x, *y, *z;              // the coordinates
    int      *faceNode[3];            // face node link
    Real     pMin[3], pMax[3];        // bounding box
    char     *type;                   // type of the face, or patch number

    Real     dsT, dsN;                // the tangential and normal length scale

public:
    CATriSurface ();
    CATriSurface (int nt, int nn, Real *x, Real *y, Real *z, int **faceNode);
    ~CATriSurface ();

    void                    Scale (Real fact) {}
    void                    Translate (Real dx, Real dy, Real dz) {}
    void                    Rotate (Real x, Real y, Real z, Real lx, real ly, Real lz,
                            Real angle) {}
    CAPoint3D               *Project (Real x, Real y, Real z) {return 0;}
    inline                  Real X(int i) const {return x[i];}
    inline                  Real Y(int i) const {return y[i];}
    inline                  Real Z9int i) const {return z[i];}
    void                    BoundingBox ();
    Real                    *GetMin () {return pMin;}
    Real                    *GetMax () {return pMax;}
};


template <class T>

class TOmniNode

{
protected:
    TOmniNode<T>            *parent;        // no of triangles
    char                    split;          // how the cNode is split
    unsigned short          pCell;          // position of the cNode in parent's children
    char                    level [3];      // the levels in x, y, z directions
    signed char             type;           // the type of the cNode, such as INTERIOR
    TOmniNode<T>            *c;             // the children, if any
    int                     index;          // an index, for grid generation */

    T                       *p[2];          // the two corner nodes
    static TOmniNode<T>     ***neighbor;    // to support multiple roots,
                                               connectivity


public:

     TOmniNode ();
    ~TOmniNode ();
     TOmniNode<T>           *FindNeighbor (char) const;
     list<TOmniNode<T>*>    FindAllNeighbor (char) const {};
     void                   SplitOmni (char) {};
     void                   CoarsenOmni () {};

     enum {UN_DEFINED, INTERIOR, EXTERIOR, INTERSECTED, BLOCKED};
     enum {X_MIN_DIR, X_MAX_DIR, Y_MIN_DIR, Y_MAX_DIR, Z_MIN_DIR, Z_MAX_DIR};
     friend class           OmniTree;

};
```

```
Class OmniTree

{
private:
    Real            pMin[3], pMax[3];       // the bounding box
    Real            dxMin, dyMin, dzMin;    // the low bound of grid spacing
    Real            dxMax, dyMax, dzMax;    // the high bound of grid spacing

    Real            dstInt;                 // the minimum cell size in the
                                               tangential
                                            // direction for cells intersecting a
                                               body face
    Real            dsnInt;                 // the minimum cell size in the normal
                                            // direction for cells interecting a
                                               body face
    int             type;                   // Octree or true 2^n tree?
    int             nRoots;                 // no of root cells
    OmniNode        **roots;                // the root cells
    OmniNode        **neighbor[6];          // connectivities of the roots
    CAPointStore    *pStore;                // a place to hold the node list

public:
    enum {OCT_TREE, OMNI_TREE};

    OmniTree ();
    OmniTree (Real *pMin, Real *pMax, int id=2, int jd=2, int kd=2, int type=OCT_TREE);
    OmniTree (int id, int jd, int kd, Real *x,Real *y,Real *z, int type=OCT_TREE);
      // id: no of points in x-direction; jd: no of points in y-direction
      // kd: no of points in z-direction
    ~OmniTree ();


    void SmoothTree ();
    void AdaptOmniTreeToGeom (CAGeometry *geom);
    void RefineOmniTree (int level);

    void GridGeneration (CAGeometryList geomList, DataSafe cPara);

};
```

A newly developed C++ graphics library named FOX (Free Objects for X) based on X and Open GL will be used to build the Graphical User Interface (GUI) for the Phase II. Compared to Motif, FOX enables a GUI to be easily built in a fraction of time need with Motif. A preliminary experiment with FOX was carried out. A GUI for geometry viewing was built in a couple of days. The example window is shown in Figure 3, which displays a triangulated surface of F-16 aircraft.

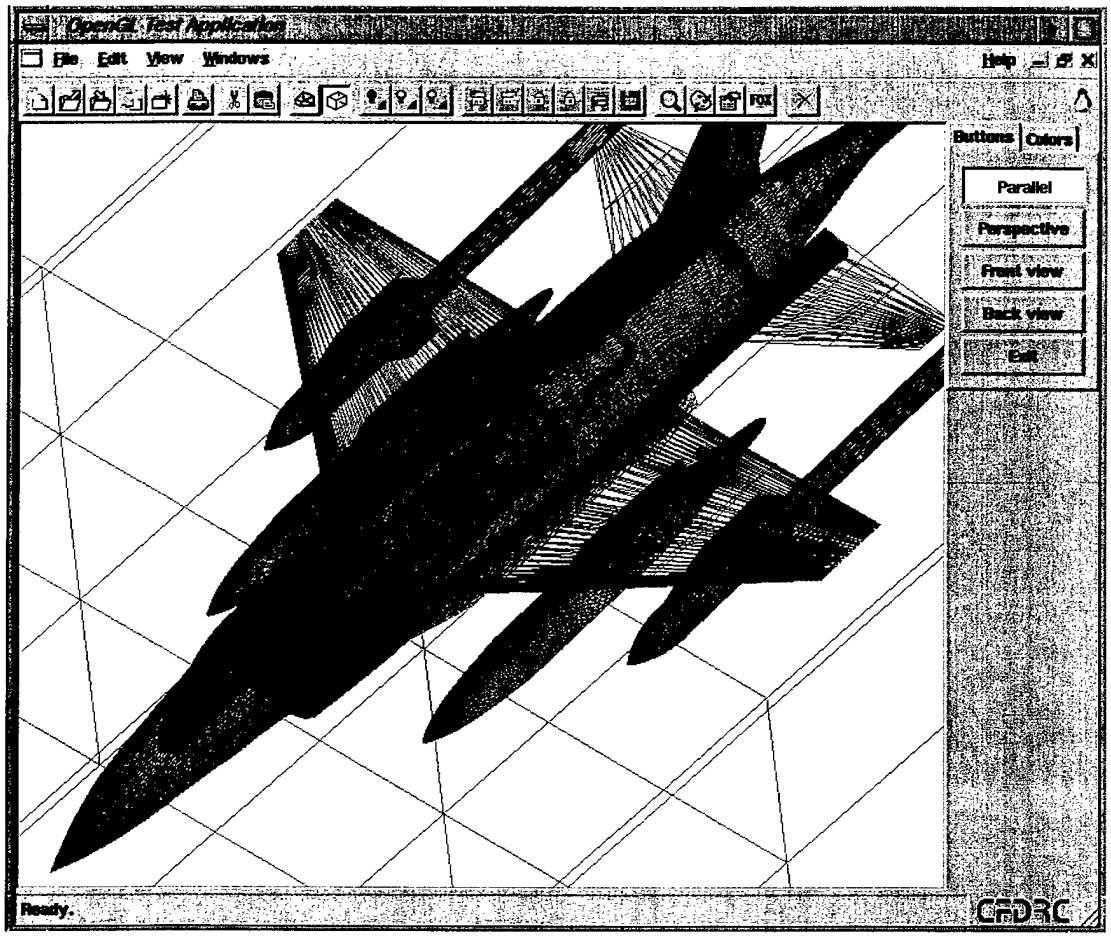

Figure 3. The Graphical User Interface for the Grid Generator

4. CONCLUSIONS

During the Phase I Option period, we redefined the Statement of Work for the Phase II and carried out preliminary design for the Phase II product. These efforts will lay a solid foundations for a successful Phase II project.