CAD-Surface Creation by Carpeting Method for Engineering Analysis

Aziz B. Patavegar
ME (CAD) student, Fr.CRCE,
University of Mumbai
aaziz_p@yahoo.co.in

Vinod B. Suryavanshi
Asst. Professor, Fr.CRCE,
University of Mumbai
vinodb@yahoo.com

Sandip N. Jadhav
Director, Centre for Computational
Technologies (CCTech), Pune
sandip.cctech@gmail.com

Abstract

Project aims at developing a CAD functionality which can create single monolithic surface from set of disjoint surfaces. Currently used methods are tedious and time consuming and therefore cannot be used for the analysis purpose. This report presents a method that is used to create a monolithic surface known as carpeting technique for surface projection on set of triangulated surface patches. Carpeting method applied in our approach can be described as projection of surface on input geometry and enhancement of surface.

1. Introduction

Due to reliable mathematical models, increase in available computing power and geometric modeling tools; large number of design and analysis problem in wide range of areas are being solved numerically. Domain discritization plays an important role in the numerical solution of set of partial differential equations using finite volume or finite element methods. Good quality grid and ability to generate grids with minimum user interaction are the two most desirable properties of a grid generator.

In this context, successful integration of Computer Aided Design (CAD) with grid generator plays a vital role. Various studies have shown that transformation of a CAD model to discretized geometry consumes significant amount of time. Thereby emphasis lies on creating quick and robust geometry-grid processing system.

Present work aims at developing a CAD functionality which can create single monolithic surface from set of disjoint surface for engineering analysis.

2. Computer Aided Design

CAD is an essential component of Computer Aided Engineering (CAE). After CAD, grid generation is a next step in CAE. Interface between CAD and grid generation systems interface is not well defined. Standard file format such as IGES, STEP, DXF or STL have enabled translation of CAD data to grid generator. Even now automatic grid generation from CAD is a distant dream. The challenges are entity conversion, representation transformation, interpretation of imported data to complex CAD clean up and repair of complete geometry. Problems arise due to errors in data translation, incorrect design, and non-adherence to standards. Error in CAD ranges from incorrect entity definition, missing surface to gaped, overlapping or intersecting surfaces.

For analysis purpose, CAD cleanup includes discarding geometric features that are not pertinent to problems in hand. Large CAD files often contain information down to machining details of part and a lot of construction entities. Repair of CAD file is daunting task because it requires an experienced user with set of tools to fix up problems manually.

3. Grid Generation

Domain discretization [1] is accomplished by covering the solution field with discrete points that can be connected in various manners to form networks of cells. Based on the distribution of the points and cells in space and their connectivity, grids are classified as structured and unstructured grids.

A structured grid of quadrilaterals and hexahedra consists of aset of co-ordinates and connectivity that is naturally embedded into element matrix. Every cell or vertex has definite number of neighbors. Structured grid is preferred choice of grid in CFD simulation. However it is considered to be difficult to generate.

An unstructured grid [2] uses triangle and tetrahedrons as the basis elements. Lack of any global structure gives great advantages to unstructured grid. It is therefore possible to add and delete nodes and element as the geometry requires or in a flow solution of adaptive problems as flow gradient indicated.

4. CAD Interface

CAD interface development involves deciphering of imported CAD data and converting it to appropriate entities of CAE geometric system. There are various representations, each supporting diverse set of entities. It is not possible to support all entities in local CAE geometric system. Thereby our focus was on converting all imported data into either standard base entities such as PBLS or triangulated surface. Because of various CAD errors, CAD surface directly can not be processed to grid generation [4][5]. Mainly three CAD errors such as gap, intersection and edge mismatch (fig 1) are common in CAD translation. These make grid generation task on CAD model with multiple surfaces difficult.

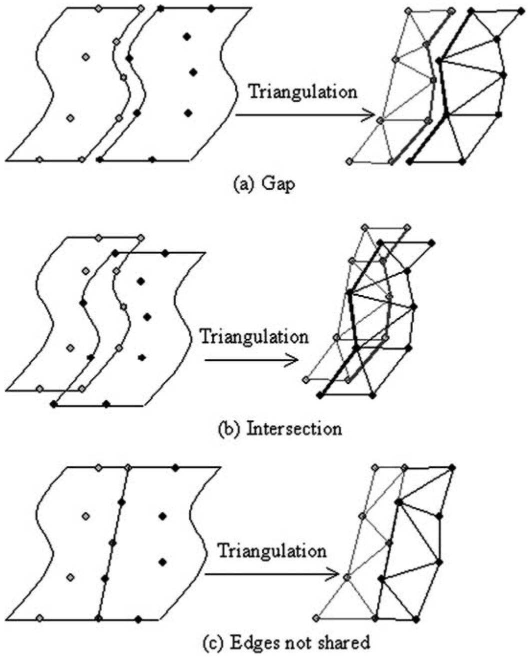

Figure 1, Error in CAD data for grid generation

4.1. DXF

The DXF (Data Exchange Format) [11] file format developed by AutoDesk Inc. supports Wire Frame and Surface Modeling entities. This format support large number of entities. DXF format is preferred for polygonal mesh surface data transfer. It is considered to be inferior to IGES or STEP because of verbose and at length description of entities. But writing DXF is easy.

4.2. STL

STL (Stereolithography Interface Specification) [10] file format developed by 3D Systems Inc., supports triangle as the only base entity. Geometry is defined with set of triangles, with Cartesian co-ordinate and normal vector from each vertex. Simplicity of specification and no modification with version has attracted other discipline to adopt STL format. Solid model can be translated to other systems in the form of triangle, using STL. Therefore DXF and STL formats are preferred in tessellated model transmission.

5. Problem Definition

Now a day many of the numerical simulations are performed on realistic CAD geometries. Complexity of CAD models has gone from few hundred surfaces to thousands of surfaces. For example, external and internal flow simulation of a car having assembly of the entire components under the bonnet can sum up to a total of 50,000 surfaces or more.

A system that can automatically create single monolithic surface or surface grid intern from a given CAD model is desired. It should process file format such as IGES, DXF, and STL. Entity and representation conversion routines are essential as these file format support lot of entities. Single surfaces creation technique are required to be fast as well as accurate to reduce the total response time of system.

We address these problems using carpeting method. In carpeting method, an initial surface of convex shape enclosing given geometry is dropped on input geometry. Process is similar to piece of cloth being dropped on physical object taking shape of physical object. We apply point smoothing as post processing operation on output carpet surface to remove any anomalies created in output carpet surface.

6. Algorithm of Carpeting

This paper presents the novel algorithm of “Surface Carpeting” towards solving the above problem of monolithic surface creation which is robust and efficient. Carpeting provides alternate methods to create monolithic surface quickly and efficiently. Initially surface of approximate shape is created. Then initial surface is dropped or projected on input geometry. In the next step, surface smoothing or surface repair algorithm would edit output surface. This would eliminate manual operation required otherwise.

In this method we make use of Piecewise Linear Curve (PLC) and Piecewise Bi-Linear surface (PBLS)[3][8] which are special cases of NURBS representation of unit degree.

6.1 Algorithm of 2D Carpeting

Problem of single monolithic curve creation is addressed by 2D carpeting. Object is defined as multiple curves in the form of Piecewise Linear Curve (PLC) is taken as input. Initial carpet is curve of PLC type with convex shape bounding the input object. Following is brief algorithm of 2D carpeting:

I. Read the PLC curve (termed carpet curve or carpet)
II. Compute curvature normal (fig 2) at all the points
III. Read segmented objects on which curve is to be dropped
IV. Project each point of carpet curve on segmented object

V. Find nearest point and assign this point as new carpet point

VI. Perform step IV and V for all carpet points

Above algorithm uses following two sub algorithms.

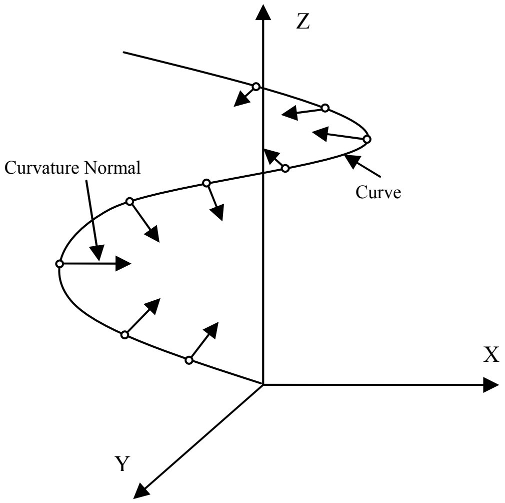

Figure 2. Curvature normal of curve

Algorithm for curvature normal computation:

I. For any internal point Centre, (with its index termed as location) of a PLC create two linear segments with points as Back (location-1), Center (location) and Front (location+1)

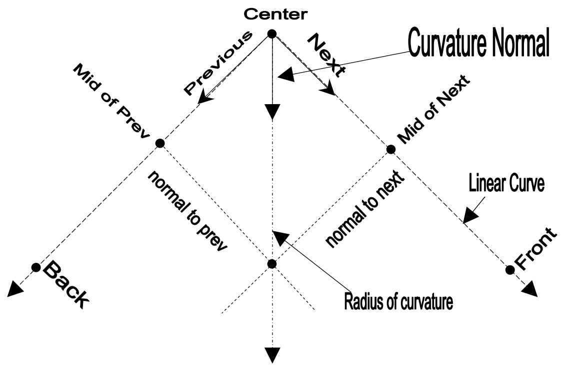

Figure 3. Curvature normal computation

II. From Centre form the two vectors Previous (Back – Center) and Next (Front – Center).

III. Calculate normal to the plane using vector cross product N_To_Plane = Next **X** Prev

IV. Calculate following vector cross products:
N_To_Next = N_To_Plane **X** Next
N_To_Prev = Prev **X** N_To_Plane

V. Compute the intersection of N_To_Next and N_T_Prev which is a Centre of Curvature

VI. Form Curvature Normal vector by connecting Centre and Centre of Curvature

Algorithm for ray segment intersection:

I. Create a ray using following equation[7]:
Ray = R.start point + t *R.direction.

II. Create a segment using following parametric equation
Segment = $P_0 + t_1 (P_1 - P_0)$

III. Equate the equation I and II and solve by Cramer rule

IV. Check for the condition
If t>0 and 0<t1<1, then intersecting
Else not intersecting or skewed

V. Find the intersection point by substituting value of t or t_1 in respective equation

Ray- t>0
Segment 0<t1<1

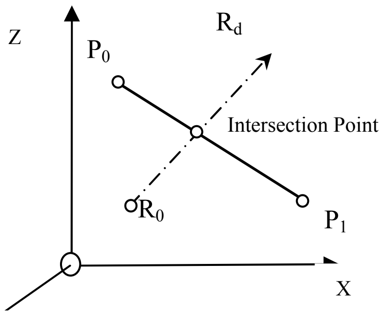

Figure 4. Ray-segment intersection

6.2. Algorithm of 3D Carpeting

Problem of single monolithic surface creation is addressed by 3D carpeting. In 3D desired geometric shape are defined using triangulated surfaces. Input objects can also be defined as a set of surfaces which are converted to triangulated surfaces for fast ray surface intersection. Initial carpet surface is Piecewise Bi-Linear Surface (PBLS) with convex shape enclosing the input object. Following is the brief algorithm for carpeting 3D:

I. Read the PBLS surface (termed as carpet)

II. Compute surface normal at all the carpet surface point

III. Read the objects on which surface is to dropped in triangulated form or set of disjoint surfaces which are converted to triangulated surface

IV. Project each point of carpet surface on triangulated object

V. Find nearest point and assign this point as new carpet point

VI. Perform step IV and V for all carpet points

VII. Repair bad surface points by point smoothing method

Above algorithm uses following two sub algorithm of surface normal and ray-triangle intersection.

Algorithm for surface normal

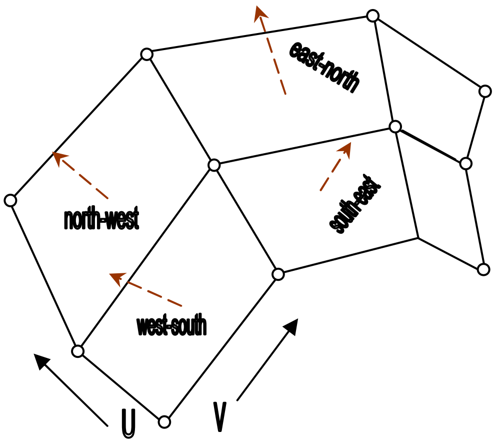

Figure 5a. Surface normal

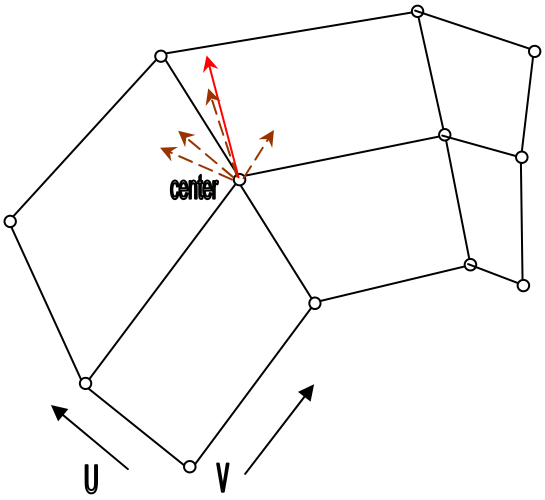

Figure 5b. Surface normal

I. For any given centre point (surface[u][v]) on surface, find neighboring points using
surface[u+1][v], surface[u][v-1],
surface[u][v+1], surface[u-1][v]

II. Calculate outward vectors from centre using
north = surface[u+1][v] – center
east = surface[u][v+1] – center
south = surface[u-1][v] – center
west = surface[u-1][v] – center

III. Calculate the normal (fig 5a) to the plane formed by neighboring points using cross product
normal_to_nw= north **X** west
normal_to_ws= west **X** south
normal_to_se= south **X** east
normal_to_en= east **X** north

IV. Find the average normal (fig 5b) at given point centre, by normal = (normal_nw+normal_ws+normal_se+normal_en)/4

Algorithm for 3D ray triangle intersection:

I. Create a ray using following equation:
Ray = R.carpet_point + t *R.normal_direction

II. Create the barycentric coordinate of a triangle using following equation
$P(\alpha,\beta,\gamma)= \alpha a + \beta b + \gamma c$ for $\alpha +\beta+\gamma=1$

III. Equate ray and barycentric coordinate equation for intersection point
$R_0 + t*R_d = a+ \beta(p_1-p_0)+ \gamma(p_2-p_0)$

IV. Solve above equation for β, γ,t by Cramer rule

V. Check the condition
If $\beta>0$,$\gamma>0$ and $\beta +\gamma<1$ and $t >0$ then
Intersecting
Else Nonintersecting

VI. Find the intersection point by substituting value of t in ray equation

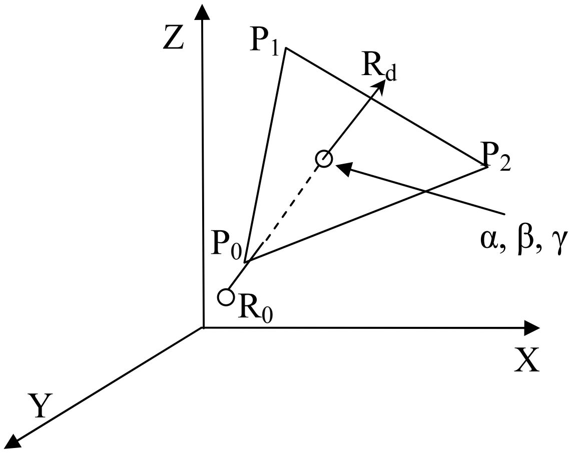

Figure 6. Ray-triangle intersection

Algorithm for surface point smoothing

There are various methods available in literature for surface point smoothing. Our smoothner uses point averaging method also knows as Laplacian smoothing. Following are the important steps:

I. Locate the highly perturbed point[u][v] on surface using surface quality checks (fig 7)

II. Modify above point using either four point stencil or eight point stencil as per following formulation
point[u][v] = (point[u+1][v]+ point[u][v+1]+
point[u-1][v]+ point[u][v-1])/4
point[u][v] = (point[u+1][v]+point[u+1][v+1]+
point[u+1][v-1]+point[u-1][v-1]+

point[u-1][v]+point[u-1][v+1]+
point[u][v-1]+ point[u][v+1])/8

III. Perform step I and II for all surface point

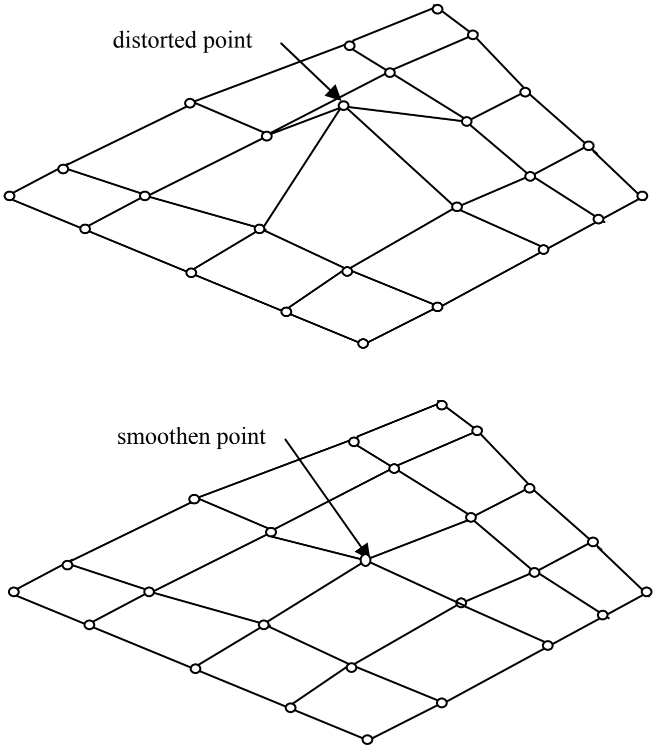

Figure 7. Surface smoothner

7. Implementation details

The code has been written using ANSI C[6] on Linux platform. CAD translator developed for DXF and STL reads all important geometric entities. Non-triangulated entities are converted to the triangulated surfaces.

Implementation of carpeting algorithm makes use of the geometric tool library developed to support the surface projection algorithm. It consist of vector algebric operations such as vector cross product, subtraction, dot product, vector normal and unit normal. Code uses the parametric as well as implicit form of equation for various geometric entities like segments, plane, triangles, curves and surfaces. To find the closest point of the surface of intersection is a tedious and time consuming process. Therefore we make use of spatial search method to increase the speed of ray tracing algorithm in surface intersection. For results display and 2D grid generation IITZeus [12] CFD preprocessing software is used.

7.1. Carpet 2D Test Case

Carpeting code for 2D was tested on India map consisting of 50 disconnected PLC curves. For analysis purpose, grid needs to be generated in this map. As shown in figure 8, initial bounding carpet was created using 400 points. Carpet was dropped in curvature normal direction. As shown in figure 9, 2D carpet code created a output PLC curve, closely following the original geometry.

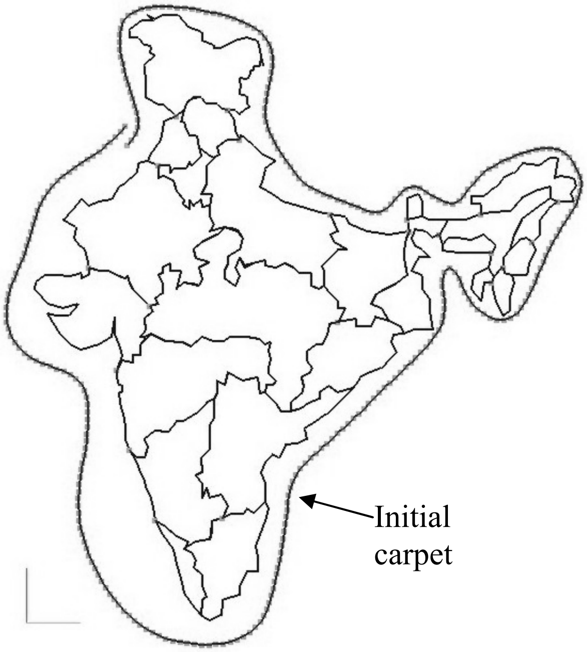

Figure 8. 2D carpeting test case – India Map

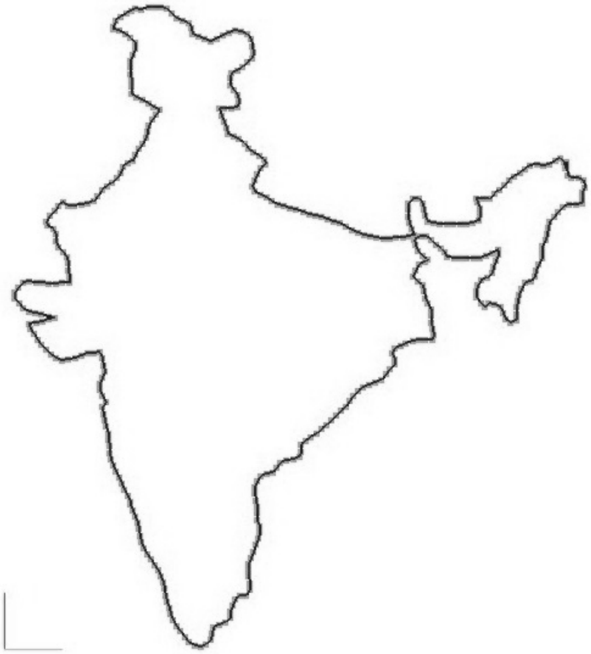

Figure 9. 2D carpet output close PLC curve

Figure 10 shows unstructured Delaunay triangulation generated in output carpet. Quality of grid generated is satisfactory which can be attributed to good quality of 1D grid created by carpeting.

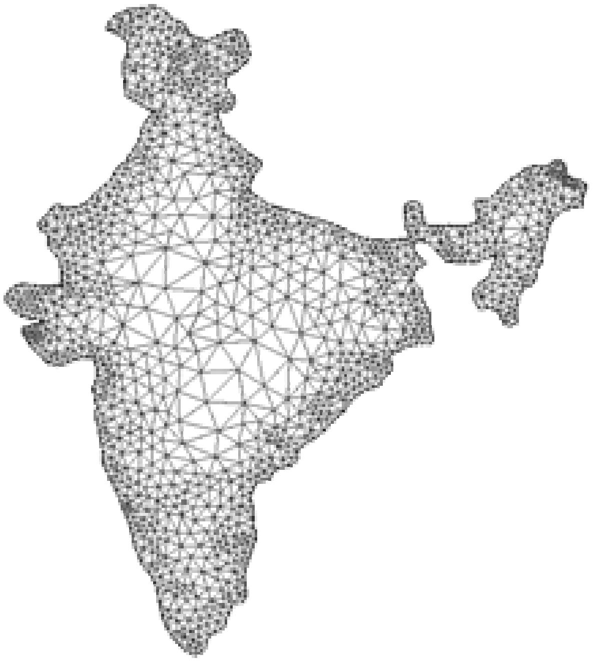

Figure 10. Unstructured grid inside output carpet curve

7.2. Carpet 3D Test Case

Carpeting code for 3D was tested for human head geometry imported using DXF reader. This geometry is created using CAD scan data converted to triangulated surface of 24236 triangles. It has many small and medium size holes since scanner can not scan all the concave regions of the geometry as shown in figure 11.

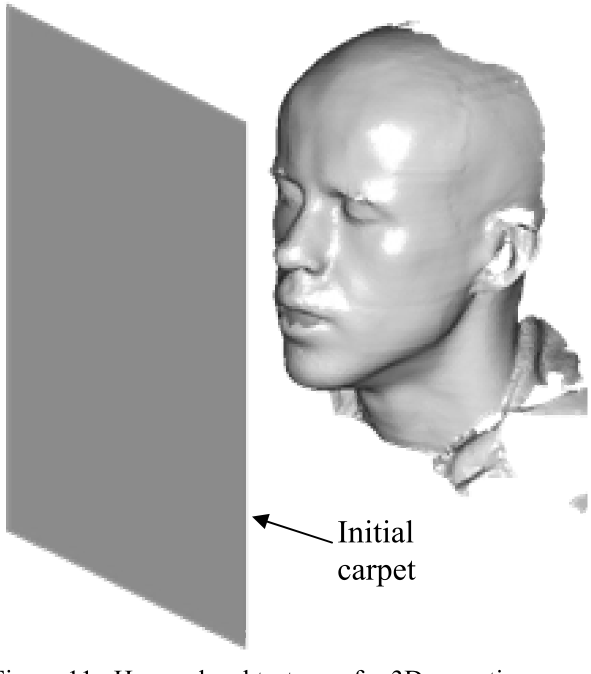

Figure 11. Human head test case for 3D carpeting

Initial carpet surface was created as plane surface consisting of 4900 control points. Output surface is shown in figure 12. Carpeting code generated single monolithic surface over set of triangles following their shape closely. Laplacian smoothner removed perturbed points due to the holes in original geometry. Very few manual corrections are required in output carpet surface to make its use in grid generation process.

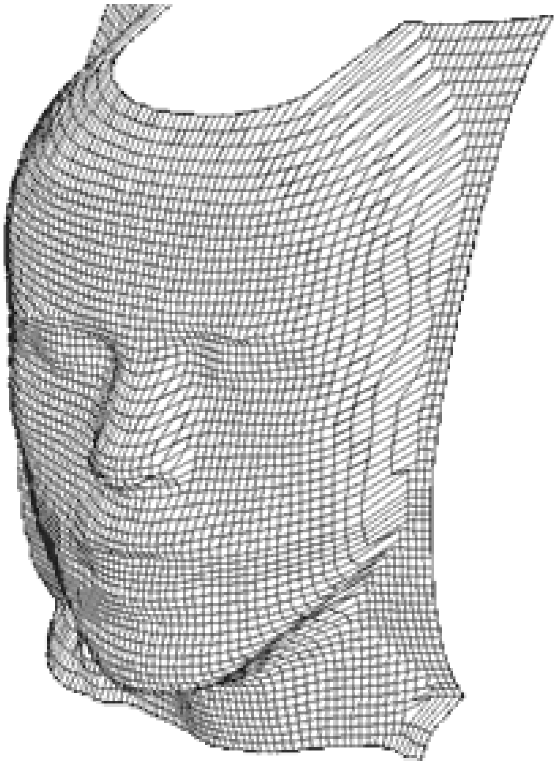

Figure 12. Output carpet surface for Human head test case

8. Conclusion

A system that can create continuous surface from discontinues data with minimum user interface has been attempted. Piecewise Bi-Linear Surface is created for given complex CAD model using these procedures. Piecewise Bi-Linear Surface created by carpeting can be directly used as surface grids in multiblock structured grid generation. Implementation of carpeting method takes care of floating point arithmetic problems. Laplacian smoothner helps to fix the problems of kink or wrinkle in output surface. Quality of output surface or grid is up to the mark. Spatial search method is implemented for efficient ray-triangle intersection and thereby reduction in computational time. Monolithic surface created helps in accelerating meshing process and hence reducing total time required for analysis cycle.

Further work would be to make automatic initial carpet geometry, surface smoothners for patches, and improvement in surface quality.

9. References

[1] Thompson, J.F., *Handbook of Grid Generation*, CRC Press, 1st Ed., 1998.

[2] O'Rourke, J., "*Computational Geometry in C",* Cambridge University Press,2nd Ed 2001.

[3] Farin, G.E., "*Curves and Surfaces in CAGD*", Academic Press Inc., 4th Ed.,1997.

[4] Steinbrenner, J.P., Wyman, N.J., And Chawner, T.C., "*Fast Surface Meshing on Imperfect CAD Models",* In Proceedings of the 9th International Meshing Roundtable, 2000, pp.333-31.

[5] Hamies, R., Aftosmis, M., "On *Generation High Quality Water-Tight Triangulation directly fro CAD"*, *"Numerical Grid Generation in Computational Field Simulations"*, The International Society of Grid Generation, 2002, pp. 27-46.

[6] Ritche, Kernigham, "*The C Programming Language*", Princeton-Hall Inc., 2nd Ed, 2000.

[7] Schneider, J.P., Eberly, H.D., "*Geometric Tools for Computer Graphic*", Morgan Kauffmann, 2003.

[8] Roger, Adams, "*Mathematical Computation for Computer Graphics*", 2nd Ed., 2001.

[9] Mezentsev, A., Woehler, T., "*Methods and Algorithm of Automated CAD Repair for Incremental Surface Meshing*", In Proceedings, *10th International Meshing Roundtable*, Sandia National Laboraories, 2001, pp.353 – 362.

[10] 3D Systems,Inc., CA, *Stereolithography Interface Specification (STL)*, http://www.3dsystems.com

[11] AutoCAD Reference Manual, AutoDesk, Inc., *Data Exchange file format (DXF),* http://www.autodesk.com.

[12] IITZeus Preprocessor, http://www.aero.iitb.ac.in/~iitzeus