

Automatic Hexahedral Mesh of Concrete Box Girder with the Vertical Section Method

Wenpu Zhao, Qingyun Zhao, Zhiqiang Lu, Shihua Yu
College of Construction Engineering, Jilin University, Changchun, 130026, China
E-mail: zhaowenpu@yahoo.cn

Abstract—According to the geometric characteristics of concrete box girder structure, the author proposed a way to automatically complete the concrete box girder with automatic hexahedral mesh subdivision algorithm—the vertical section method. After introducing its computing ideas, it will be applied to engineering examples to verify its feasibility.

Keywords- box girder, the vertical section method, hexahedral mesh

I. INTRODUCTION

Concrete box beam is now widely used for the box girder bridge structure, and there is no highly efficient method of mesh generating for complex bridge structure. In this paper, on the basis of many years' CAD software development, according to the geometric characteristics of concrete box girder structure, the author proposed a way to automatically complete the concrete box girder with automatic hexahedral mesh subdivision algorithm - the vertical section method.

II. BASIC IDEAS OF THE VERTICAL SECTION METHOD

A. Basic Parameters.

The coordinate system used in this method is three-dimensional, and the three axes are respectively L-axis along the center line direction, S-axis along the cross-sectional direction, and Z-axis along the vertical direction. So the basic length of the mesh along these three axes are setting respectively L_1 , L_s , and L_z (Figure 1) as well as L-axis as basic cross-section.

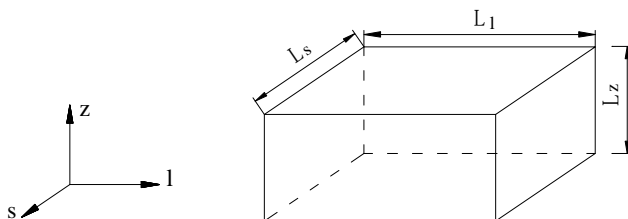


Figure 1. Three-dimensional coordinate system

B. Plane Mesh of the Basic Cross Section.

When we select fundamentals, we mainly consider whether the cross-sectional form is representative. If both ends of the box girder have diaphragms and the section is not typical box girder cross-section, they cannot take as a fundamental of the box girder mesh. So midspan of the box girder is selected as the fundamental, shown in Figure 2.

After deciding the fundamental, we mesh the fundamental in accordance with the following steps.

1) *Computing the Position of the Fundamental.* Because L-axis is along the center line direction, we can compute the position of the fundamental on the basis of the L-axis coordinate.

2) *Solving the Section in the Fundamental Aimed at Geometric Model.* The fundamental is only a cross-section determined by L-axis coordinate and have no thickness, so we need to obtain a section which can be meshed during the subdivision. Take the geometric form of the fundamental as the section form and take the mesh length L_1 along the direction of the L-axis as the section thickness, shown in Figure 3.

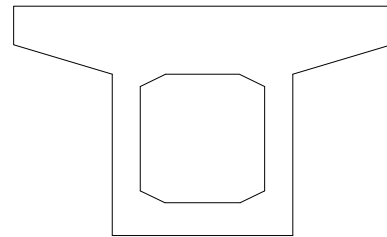


Figure 2. Fundamental

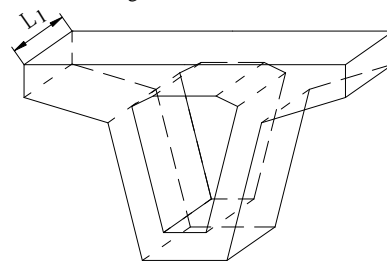


Figure 3. Section

3) *Section Coordinate System Transformation.* Section length along the L-axis has been defined as the mesh length, so in this direction subdivision is not needed, thus coordinate system of the section is transformed into two-dimensional coordinate system for further computing, shown in Figure 4.

4) *Defining the Outer Ring and Inner Ring.* Search algorithm should be used for various segments of the cross section to judging whether there is a coincidence point, and take the lines which have coincidence points to merge into

rings, then compute area of each ring, and the largest one is the outer ring and the others are inner rings.

5) *Inserting Node at the End of the Segment.*

6) *Computing the Bounding Box of the Outer Ring.* The bounding box of the outer ring divided by the nodes distance, and are inserted the nodes to form a two-dimensional array, shown in Figure 5.

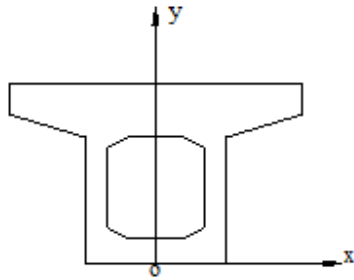


Figure 4. Two-dimensional coordinate system of the section

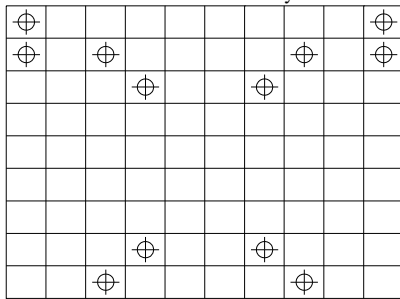


Figure 5. Two-dimensional array comprised by codes

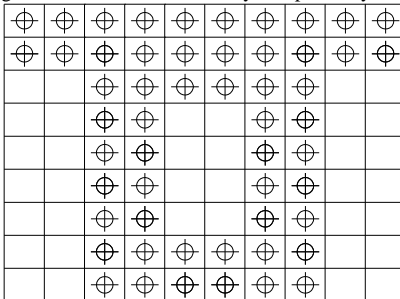


Figure 6. Two-dimensional array after inserting the codes

7) *Using Judgemental Algorithm to Adjusting the Oblique Webs Haunched.*

8) *Inserting Nodes.* Compute the start and end point of each segment in the above two-dimensional array. If they are not adjacent, we choose to have a linear interpolation. After inserting the nodes, two-dimensional array is formed, shown in Figure 6.

9) *Computing the Quantity of the Section of the Full-bridge.* According to the distance of the L-axis, two-dimensional array is extended to three-dimensional array, and two-dimensional array is put into the corresponding location of cross-section.

C. *Loop Computation of the Remaining Cross-section.*

After meshing the fundamental we can use the same way (loop computation) to mesh the others. The methods are as follows.

- 1) *Splitting the cross section in the geometric model.*
- 2) *Analysing the outer and inner ring of the cross-section.*
- 3) *Determining the correspondence of the segments.*
- 4) *Mapping the finite nodes of the corresponding segments.*
- 5) *Assembling the nodes into the three-dimensional array.*

D. *Handling the Section with Only Outer Ring.*

In the diaphragm beam, there is only outer ring without inner ring. After mapping the nodes of the outer ring, the nodes can be generated by directly inserting nodes at the internal face of the outer ring.

E. *Generating the Hexahedral Grid.*

By the previous steps, an array contained three-dimensional nodes can be got. We first choose a non-empty Node $A(z_i, s_i, l_i)$ and then after plusing 1 at the array subscript search the other 7 points b, c, d, e, f, g, h. If these eight nodes are non-empty, we can determine a hexahedral unit here and they are made up of a hexahedral unit, as shown in Figure 7. Then by using the same methods we can search all the nodes to generate the hexahedral mesh of the full-bridge.

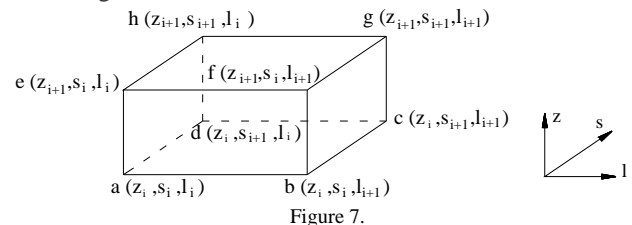


Figure 7.

III. IMPLEMENTATION OF THE PROGRAM

For achieving automatic hexahedral mesh, the second development is adopted by the the python language and combining with functions and modules in the Salome Computer-Aided Modeling Platform in this paper

A. *the Main Functions Used in the Procedure.*

1) *Geompy.SubShapeAll.* A parent element is broken down into a series of child elements by command. The parent element contains edge, face, line, shell, solid and so on. Child element can be vertex, edge, line and so on. The two functions are respectively the type of the parent and child element, and their return values are a list of the child elements.

2) *Geompy.PointCoordinates.* It can get the coordinates of the nodes. The return value is the coordinates of the target node.

3) *Geompy.MakeMarkerPntTwoVec(theOrigin ,theXVec, theYVec).* A new coordinate system can be created by this

command. The return value is the object under the new coordinate system.

4) *Geompy.MakeTranslation(theObject, theDX, theDY, theDZ)*. It is translation operation, and the return value is a translation unit.

5) *Geompy.MakePosition(theObject, theStartLCS, theEndLCS)*. It can move an object, and the return value is the new object moved.

6) *GetMesh(self)*.

7) *SetNodeOnVertex*. It can bind a node in the vertex. If the operation is successful, the return value is true, otherwise not.

8) *SetNodeOnEdge (self , NodeID , Edge , paramonEdge)*. It can build a node in the edge. If the operation is successful, the return value is true, otherwise not.

9) *AddNode (def AddNode (self, x, y, z)*. A node can be created in the mesh by the coordinate. The return value is the number of the new one.

10) *AddFace (self, IDsofNodes)*. According to a given node, a linear quadratic face is created, and the return value is the number of the new one.

IV. EXAMPLE

In order to verify the feasibility of the vertical section method, a straight bridge and a section of curved bridge are selected to be meshed.

Shown in Figure 8 and 9, a straight bridge using the vertical section method gets a high quality hexahedral mesh. Shown in Figure 10, a curved bridge also gets a satisfying result. So the vertical section method is reliable in the practical project, while it can be applied to many fields.



Figure 8

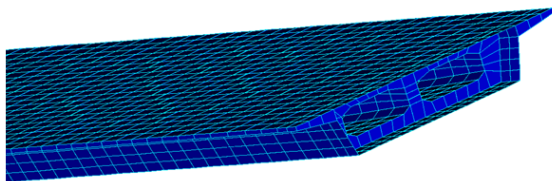


Figure 9

V. CONCLUSION

This paper introduces the vertical section method which can be used in the automatic hexahedral mesh subdivision. After explaining the ideas about it, the procedure is made by the salome and python language, and then we get a good result by selecting a section of box girder to mesh. All in all, this method not only stay in the theoretical stage, but also can be applied to the practical engineering.

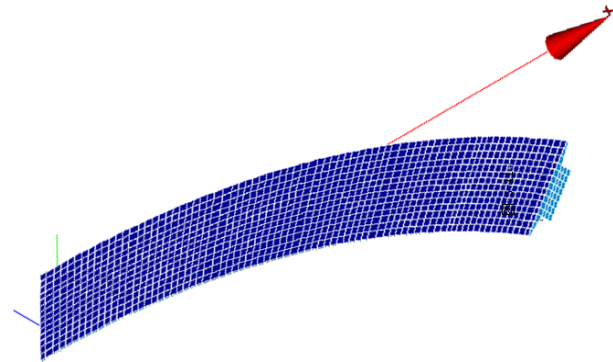


Figure 10

REFERENCES

- [1] Raphael, Krishnamoorthy C S, Automating finiting element development using object oriented techniques, Engineering Computations, 1993.
- [2] Patrick Knupp, Atanly Steinberg, Fundamentals Grid Generation, CRC Press, 1993.
- [3] Schneiders R, Weiler F, Oct ree, Based Generation of Hexahedral Element Mesher, 5th International Meshing Roundtable, Sandia National Laboratories, 1996.
- [4] Blacker Ted D, Paving, a New Approach to Automated Quadrilateral Mesh Generation, International Journal for Numerical Methods in Engineering, 1991.
- [5] Lei Wang, A Research on Crucial Technology to Three-dimension Bridge CAD, Jilin University, 2005.