

MakeCSE

Version 5.1

by

Stephanie Wimmer

1. Introduction

The purpose of the code MakeCSE is to insert Cohesive Surface Elements (CSE's) within or around a preexisting mesh. The code used the programming libraries of Tahoe and Exodus. The code works for any type of meshing pattern, texture, or scheme and for any dimension. At present, the only limitation of the code is its library of element types. The code was developed as a stand alone preprocessor, but could be included within a finite element program to adaptively insert CSE's as needed or within a mesh generator to create initial CSE's.

The stand alone code operates by reading in the traditional finite element connectivities and coordinate list. The user supplies a short input file, describing what method is used to prescribe where CSE's are to be inserted, if any nodes are not allowed to split, and how the node sets are to be mapped. The resulting connectivities and coordinate list is then written. Side sets or node sets are transferred or mapped. The input and output data is typically in an Exodus II file, but other formats are also allowed.

2. What is a CSE?

A CSE is not a typical finite element. In reality it is two surfaces that lie between traditional finite elements. The surfaces are used to model decohesion in fracture simulations. Decohesion modeling developed from the theories of Barenblatt and Dugdale. They envisioned the existence of a small region (cohesive zone) ahead of the crack over which the material behavior can be characterized by atomic interaction.

In finite elements the cohesive constitutive relation determines the separation distance between the two surfaces for the CSE. The amount of separation can be divided into normal and tangential crack opening displacements or an effective opening displacement. When the crack opening displacement reaches a critical value, the traction vectors are set to zero. At this point the cohesive surface describes "new" free surface.

In the past, CSE's were created for only a certain type of 2D finite element geometry. When a CSE is first created, the element has zero volume and the two surface lie on top of each other. Existing mesh generators (like Patran) do not like creating zero volume elements. However

it is possible to create CSE's by first creating non-zero volume elements and then translating the nodes on one surface of the element to lie on top of the other surface. This involves a lot of time and limits the geometry and meshing pattern that is creatable.

There are only a couple of methods currently used to insert CSE's, all of which use a pre-existing 2D mesh of triangle elements. A typical method for creating CSE's is to use an orthogonal or mapped meshing scheme, where the nodes are split upon element creation for a certain zone. The pattern of the node numbering is known a priori. Then CSE's are put on all surfaces within the zone.

3. Implementation

The code works by inserting the CSE, finding neighbors, renumbering, and mapping facet nodes that appear in node sets for each element facet specified by the user. This code is only limited by the library of finite element types. The current list of library elements is shown in Table 1.

The user has three methods to specify the element facets: facet, zone, and boundary. The facet method allows the user to list the element facets in side sets. A list element groups is provided for either the zone or boundary method. The zone method will determine all internal element facets, while the boundary method will determine all the element facets between the specified element groups. For the zone method, facets on the boundary surfaces between element groups are treated as internal facets. The default, for the zone method, is not to split any of the nodes on the exterior surface of the zone. The user is allowed to specify if some or all the exterior surface nodes are allowed to split. Table 2 lists the options for specifying the splitting of exterior surface nodes.

Table 1: Element Types Supported by MakeCSE

Original Finite Element	Cohesive Surface Element
Tria 3	Quad 4
Tria 6	Quad 8
Quad 4	Quad 4
Quad 8	Quad 8
Tet 4	Penta 6
Tet 10	Penta 15
Hex 8	Hex 8
Hex 20	Hex 20

Table 2: Exterior Zone Node Splitting Options

Option	Description
1	Split None
2	Split All
3	Split None of the Nodes in the Node Sets
4	Split Only the Nodes in the Node Sets

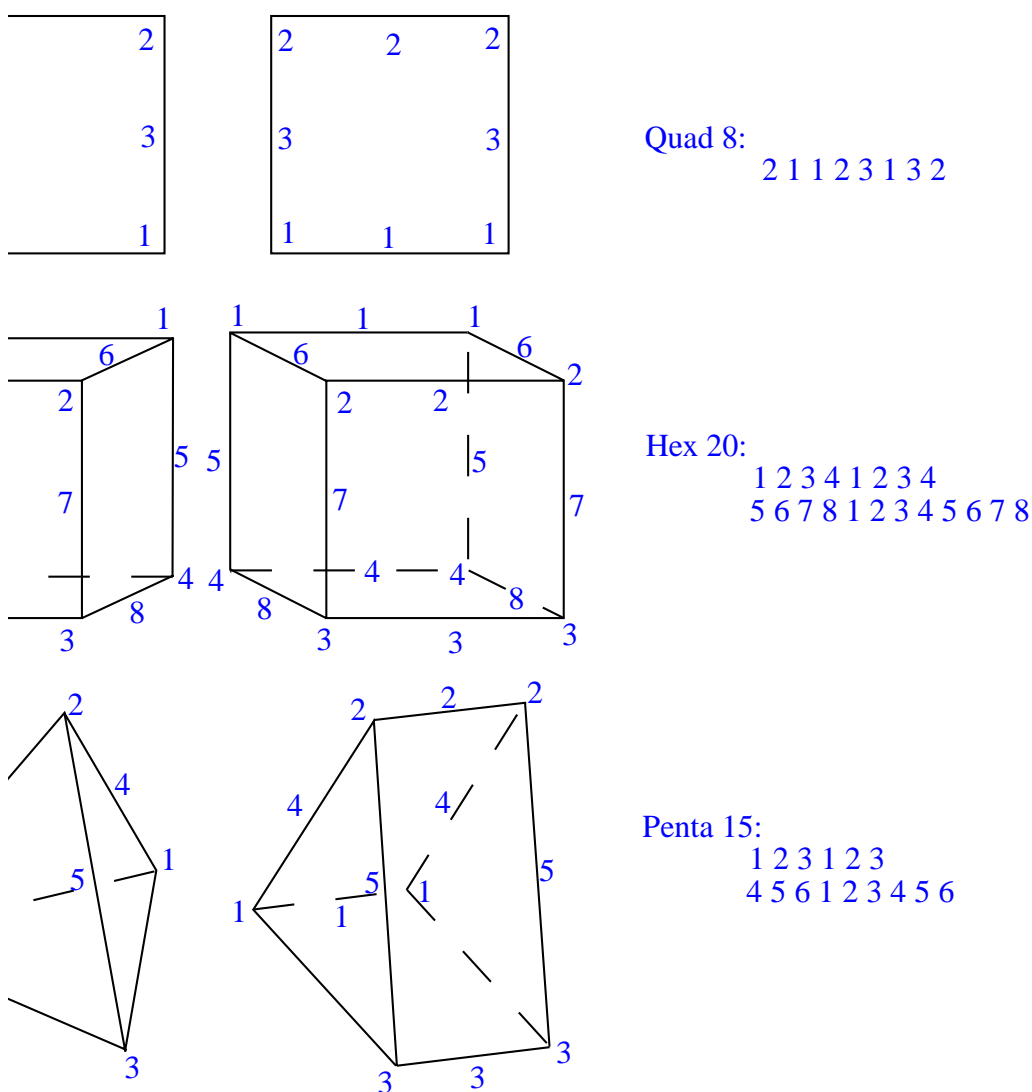


Figure 1. Initial Cohesive Surface Element Numbering. The node numbers are locally numbered for the facet of the regular element shown. Note: CSE's are drawn with width for easy viewing.

CSE's are inserted at the element facets specified. The specified facet is called surface 1, and the neighboring element's corresponding facet is called surface 2. If there is no neighboring element (i.e., the specified element facet was external), then surface 2 is a body exterior surface. Initially the newly inserted CSE is numbered based on the existing node numbers on the specified element facet. Figure 1 shows how a CSE is initially numbered based on the numbering of the original finite elements facet numbers.

After the CSE is initially numbered, a list of element facets using each node on surface 1 is compiled. Each list is divided into groups. The group boundaries are the CSE's in the list. Each group is then assigned a node number, starting with the existing node number. Then all element facets in the list are renumbered.

Table 3: Cohesive Surface Numbered by Element Facets

CSE Type	Facet Numbering
Quad 8 1 2 3 4 5 6 7 8	Facet 1 1 2 5 Facet 3 4 3 7
Hex 20 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20	Facet 1 1 2 3 4 9 10 11 12 Facet 2 5 6 7 8 13 14 15 16
Penta 15 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	Facet 1 1 2 3 7 8 9 Facet 2 4 5 6 10 11 12

Table 4: Renumbering Methods

Input Value	Name
0	None
1	Added Nodes
2	All Nodes

The extraneous mid-side nodes for higher order elements are numbered using preexisting nodes and are not given unique nodes, as they are not used in actual calculations. They are just used to represent a CSE that is valid for graphically previewing. In two dimensions, the user's analysis program will need to read the connectivity for the first and third facets of the CSE quad-

ilateral using the Tahoe facet numbering convention. In three dimensions, the cohesive surfaces are the first and second facets. Table 3 lists the node numbering of the two facets that should be used in analysis calculations to designate the two cohesive surfaces.

3.1. Renumbering

The program allows the user the option of having the nodes renumbered after all the cohesive surfaces have been inserted (see options listed in Table 4). If *RENUMBER does not appear in the input file, then no renumbering is done. For option 1, the added nodes are inserted into the coordinate list following the node that they were split from. Option 2 renumbers all nodes using a simple scheme.

Table 5: Node Set Transfer Methods

Input Value	Name
0	Surface 1
1	Surface 2
2	Mapping
3	Splitting

3.2. Mapping Node Sets

If renumbering occurs, the node sets are examined to determine if their numbering also changes. The user is allowed to choose from three node mapping methods listed in Table 5. For Method 0 and 1, if the node was split, the node number on the designated surface is used. Method 2 forces the code to determine on which surface the node in the node set lies and renumber the node set node. For CSE's created along an exterior surface, Method 2 will use the nodes on surface 2, the exterior surface. An exterior surface is the boundary of the area in which CSE's are inserted and is not necessarily the body's exterior surface. Methods 0-2 work best when there is a clear surface 1 and surface 2 at the node point. Method 3 just adds all the newly created node for that coordinate point to the node set list, allowing the node set to grow by one.

3.3. Courtesy Output Data

For the user's convenience, the option of outputting contact surface data has been added. For this option, the user specifies the CSE block ID's, then the program writes to the output database surface 1 and surface 2 facets to side sets and surface 1 and surface 2 nodes to node sets.

Also for the user's convenience, the program outputs the massless nodes created in a node set. Massless nodes are nodes on surface 2, when surface 2 is not attached to a material element. The user can use this node set as a boundary condition in an analysis program or use the node set to determine if there are any problems with the original mesh creation (like nodal equivalency).

Finally for initial boundary conditions and other purposes, the code will output all the nodes used by an element block to a node set.

3.4. Element Splitting

For the user's convenience, the program has the ability to split elements before inserting CSE's as prescribed. The side sets are transferred to appropriate facets. None of the existing nodes are renumbered at the time triangles are created, so the user may wish to optimize the node numbering after running this program. The splitting methods are listed in Table 6. This is not a quick process and the user is advised to use a mesh generator designed for this action before using MakeCSE.

4. User's Interface

The user's interface consists of a short input file or the user may choose to specify the input interactively and save a input file for future reference. The input file specifies where CSE's are to be created, the type of output file desired, what geometry should be transferred to the output file, and how node sets should be mapped. A database can be an Exodus II file, a set of text files, or inline data. Only the element groups, node sets, and side sets specified in the input file will be transferred to the output file. For a listing of databases supported, see Table 7.

Inline text data (Tahoe I) is not longer accepted. All geometry is read from the input database. All keywords are preceded by a *. Keywords are shown in all capital letters in the following sections, however, the user is allowed to used lower or mixed cases. The input file is white space delimited. All text following the symbol '#' to the end of the line is considered a comment and is skipped. Comments should not appear in the middle of an array list. The keywords may appear in almost any order. The user must specify at least one facet to insert a CSE at (*FACET, *ZONE, or *BOUNDARY).

The user must specify the CSE block for each set of element facet data supplied (*FACET, *ZONE, *BOUNDARY). This allows the user to do more than one data set at a time and to have the resulting CSE's be outputted to different element blocks. Anytime a [/] is used in the following keywords, more than one line of data can be used to specify the desired results. Any list of ID's can contain the short hand '-' to designate a continuous list.

Table 6: Element Splitting Methods

Input Value	Element Splitting Method
0	X - Method for Quad 4 to Tria 3
1	Slash Method for Quad 4 to Tria 3
2	Back Slash Method for Quad 4 to Tria 3
3	Star Method for Quad 8 to Tria 3

Table 7: Databases supported by MakeCSE.

Flag	Database Name
0	Tahoe I ^a
1	Tahoe II
2	TecPlot 7 ^a
3	EnSight 6 Gold ASCII ^a
4	EnSight 6 Gold Binary ^a
5	Exodus II
a. Output Only	

4.1. Input File Keywords

*BLOCKTONODESET

list_of_element_block_ids

For the *list_of_element_block_ids*, the program will output the nodes used as node sets.

*BOUNDARY

[list_of_element_block_ids 1] [list_of_element_block_ids 2] [output element block id]

This keyword consists of three columns of data. *list_of_element_block_ids* 1 and 2 defines the surface where the CSE's are to be put. The *output element block ID* designates where the newly created CSE's belong. Repeat this row of data as needed.

***CONTACT**

list_of_element_block_ids

For the *list_of_element_block_ids*, which are CSE block ID's, the program will output the facets on surface 1 as side sets, and the nodes on surface 2 as a node sets.

***COPYSIDASET**

[Sideset ID] [Element Block ID]

This *keyword* will consist of two columns listing *side set ID's* and *element block ID's* in columns. Repeat this row of data as needed.

***EDGETYPE**

option

This is an optional keyword, if this is not found, then the program assumes that all exterior surface nodes of the zone are not to be split.

***EDGENODESETS**

[NodeSet_ID_list]

This is an option keyword and only need to be used if the user wants to specify certain exterior surface nodes of the zone not to be split or to be split. It must be used in conjunction with *EDGETYPE.

***EOF**

Anything found after this in the input file is ignored. This keyword is optional and not needed, but provided for user to make lots of comments.

***FACET**

[sideset ID] [Element Block ID] [Output Element Block ID]

This *keyword* will consist of two columns listing *side set ID's* and *element block ID's* in columns. The *output element block ID* designates where the newly created CSE's belong. Repeat this row of data as needed.

***INPUT**

database_flag
filename (if Tahoe II or Exodus)

This keyword is not needed if reading all data from Tahoe I.

***MAPNODESET**

[Nodeset ID] [Transfer Method]

This *keyword* will consist of two columns listing *node set ids* and *methods* in columns. Repeat this row of data as needed.

***OUTPUT**

database_flag
write_external_files_switch (if Tahoe II)

The user can specify one main input database and one output database type. The entry *write_external_files_switch* only applies to Tahoe II databases.

***RENUMBER**

option

***SINGLE**

list_of_node_set_ids

This keyword is used to identify which nodes are not allowed to split.

***SPLITELEMENT**

[list_of_element_block_ids] [Creation Method]

This keyword will consist of two columns listing *element block ids* and *methods* in columns. Repeat this row of data as needed.

***TITLE**

title

The title is limited to 255 characters. The end of the title is denoted by a return/enter or end line character.

VERBOSE *on/off_switch

This keyword gives the user the option to see extra information in the log file pertaining to the mapping of the node sets. Not including this keyword, results in the suppression of the extra information.

***ZONE**

[list_of_element_block_ids] [Output Element Block ID]

This *keyword* will consist of two columns of data. The output *element block ID* designates where the newly created CSE's belong. Repeat this row of data as needed.

5. Example Problems

Table 8 lists a short summary of the example problems currently used to verify the programs abilities. Most of these problems are described in this manual. These examples are provided to the user to examine. Some examples are described in the following pages of this manual. The input files for these examples and others are provided to the user in the examples directory that accompanies the program executable.

Table 8: Comparison of Test Problems

Name	Method	CSE's Created	Nodes Added	Total Nodes
quad4sidesn	Facet	16	18	54
quad4sides2	Boundary X Split	6	7	68
quad8tria6	Zone	13	27	68
hex8simple.4	Facet	4	9	27
hex8simple	Zone	12	7	34
tet4	Zone	56	31	58
tet10	Boundary	4	15	86
analyticalsym	Facet	400	603	4221
3grain	Boundary	88	120	1035
3grainb	Boundary	32	40	955
55grain	Boundary	642	696	3627
beamDCB	Boundary	20	21	252
hex20	Zone	12	25	106
quad4sides3	Zone Slash Split	45	50	86
4point3	Zone	2927	4254	8263
quad4sidesn2	Boundary X Split	16	18	54
indenttri	Zone Edge Method 4	239400	397806	558507

5.1. Facet Method using 4-noded Quadrilaterals

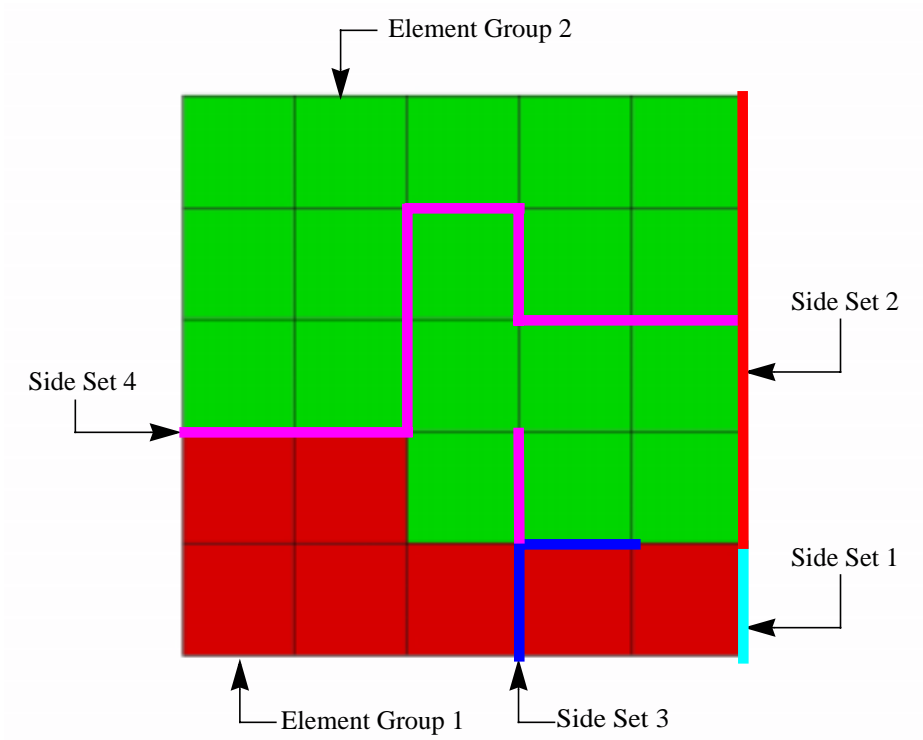


Figure 2. Example problem 5.1. using a quadrilateral mesh.

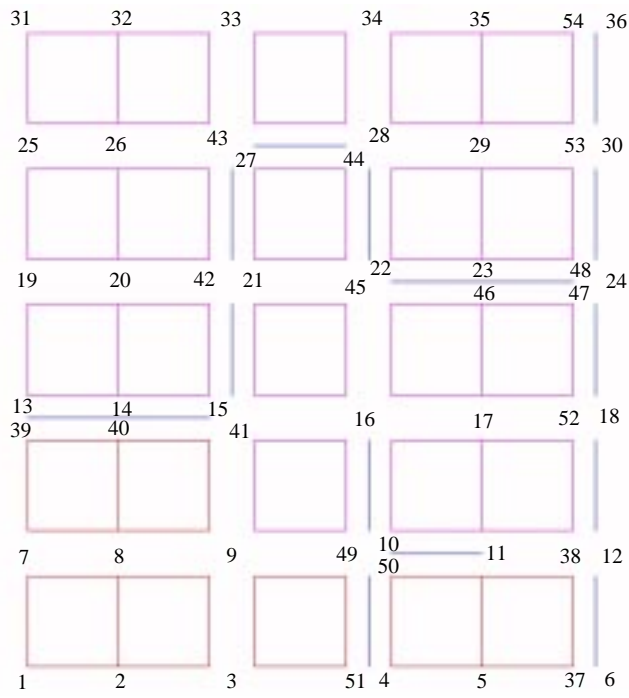


Figure 3. Example problem 5.1. after MakeCSE using the facet method.

5.2. Zone Method using 8-noded Quadrilaterals and 6-noded Triangles

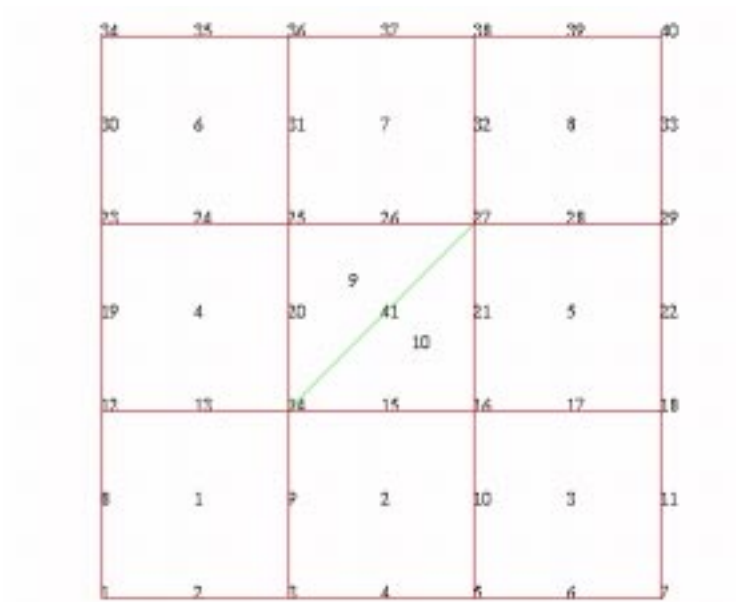


Figure 4. Example problem 5.2. using quadrilaterals and triangles.

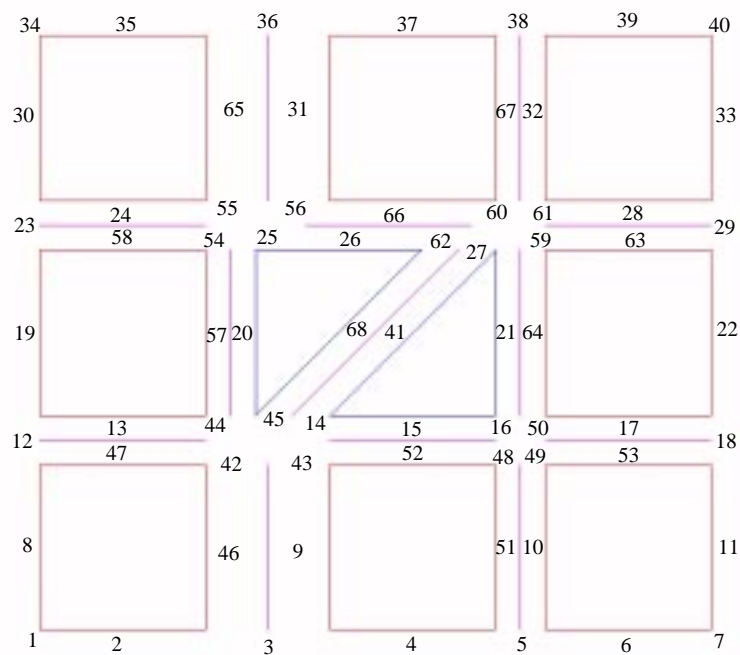


Figure 5. Example problem 5.2. after MakeCSE using the zone method.

5.3. Boundary Method using 4-noded Quadrilaterals

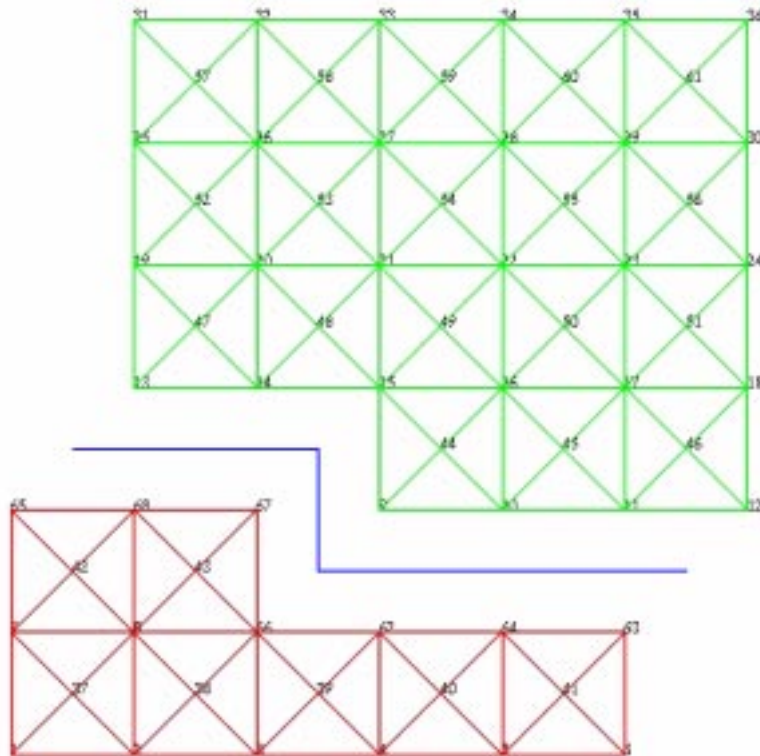


Figure 6. Example problem 5.3. after MakeCSE using the boundary method.

5.4. Facet Method using 8-noded Hexahedrons

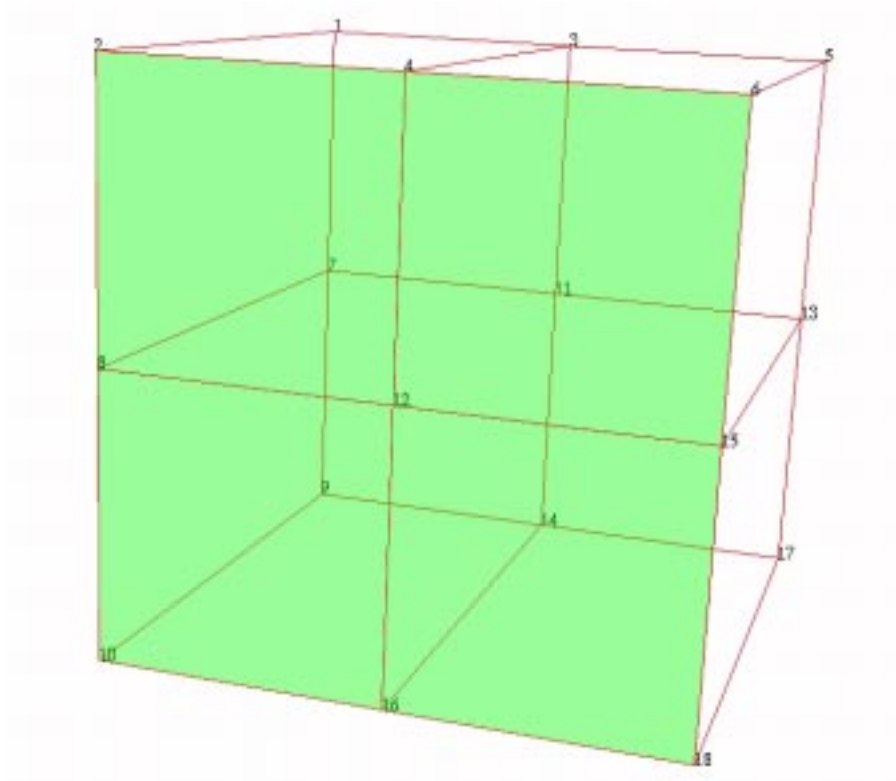


Figure 7. Example problem 5.4. using hexahedrons.

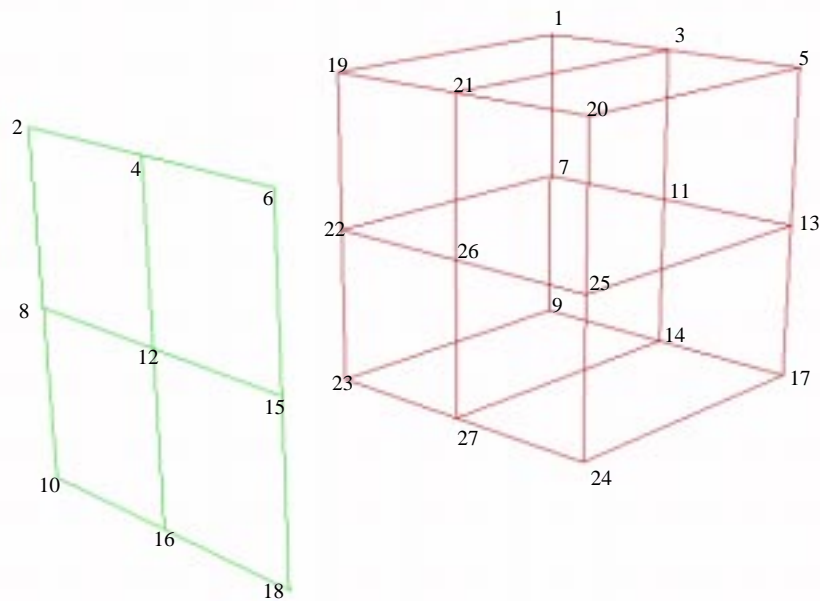


Figure 8. Example problem 5.4. after MakeCSE using the facet method.

5.5. Boundary Method using 10-noded Tetrahedrons

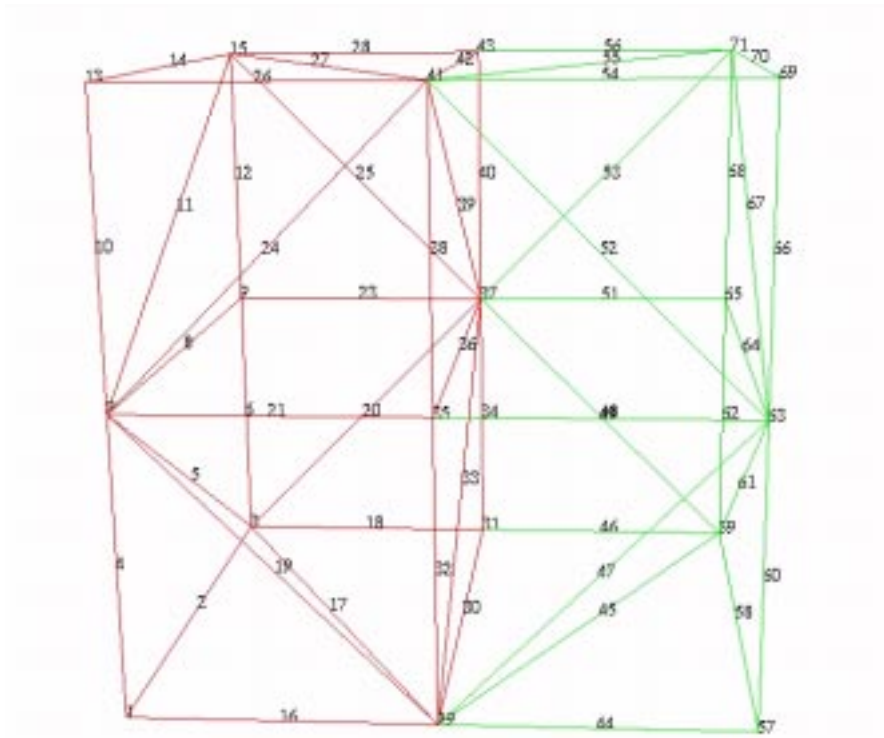


Figure 9. Example problem 5.5. using tetrahedrons.

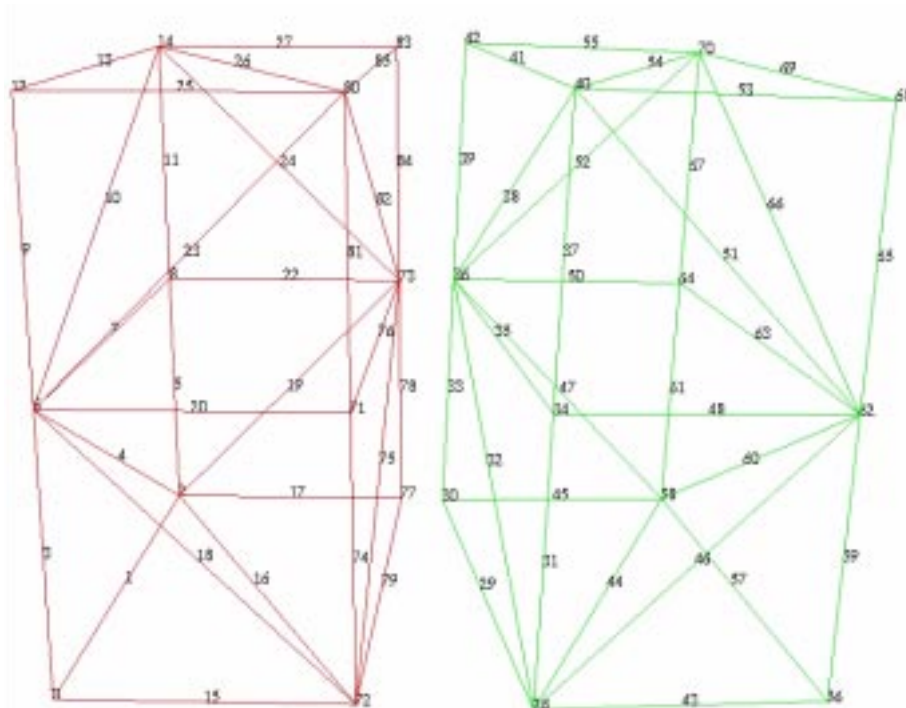


Figure 10. Example problem 5.5. after MakeCSE using the boundary method.

5.6. Zone method with Edge Method 4, using 3-noded Triangles

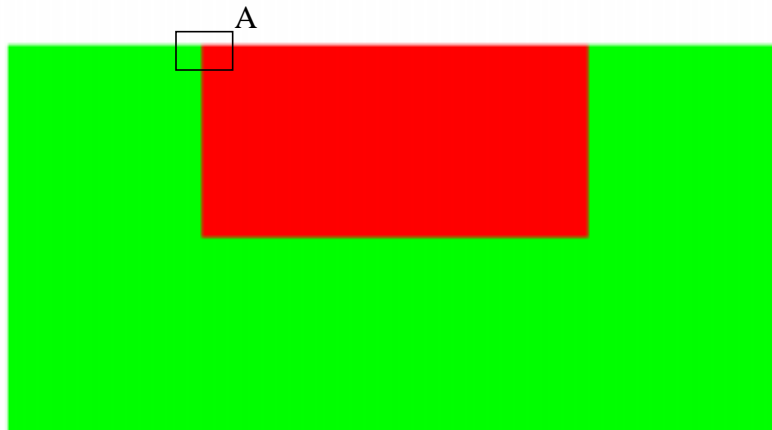


Figure 11. Example Problem 5.6 using triangles after MakeCSE.



Figure 12. Enlargement of Figure 11, Box A.

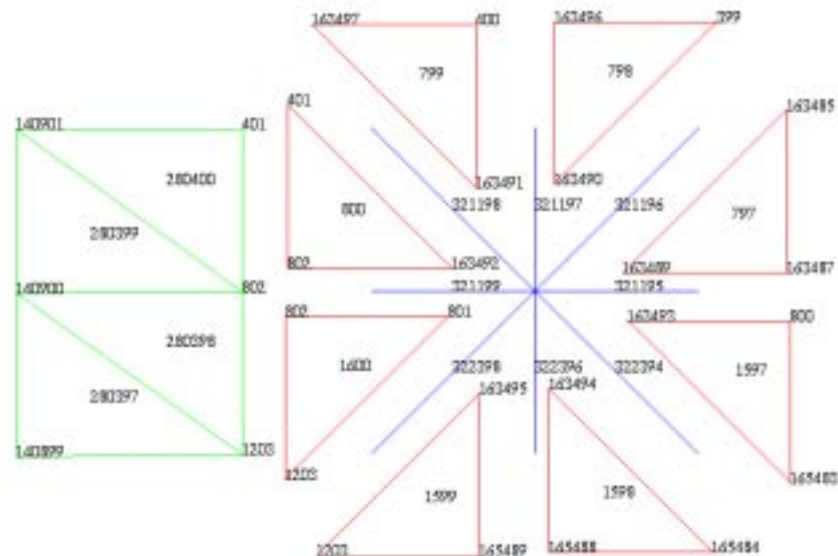


Figure 13. Exploded view of Figure 12. Top or free edge of element block 1 (red) is “double-noded”, while the side or interface edge is single-noded.

6. Version Notes

- Version 1.0.** Program created to insert CSE's with zone method.
- Version 2.0.** Program modified to insert CSE's with facet method, but only one side set is allowed to be specified.
- Version 3.0.** Program allows for multiple side sets to specify the facet method.
- Version 3.3.** Program modified to insert CSE's with boundary method.
- Version 4.0.** Separated data storage, and used more Tahoe-like classes for data storage. This allows multiple element groups. Also updated connectivity numbering and node set numbering within the CSE creation loop. Thereby eliminating the initial negative numbering.
- Version 4.2.** Input Manager creation for Exodus input.
- Version 4.3.** Input Manager modified to allow text input, and a mixture of text and Exodus input.
- Version 4.3.1.** Resize/allocation bug fix.
- Version 4.3.2.** String memory leak bug fix.
- Version 4.3.3.** Cleaned up code. Aligned internal numbering to coincide with Tahoe 2.16.
- Version 4.3.4.** Cleaned up code. Added the option of converting quads to triangles.
- Version 4.3.5.** Aligned code to agree with Tahoe 3.0. Now transfer set and block ID values instead of renumbering them. Added node transfer method 0. Made keyword EOF an option. Added Tahoe II Output.
- Version 4.4.** Reorganized input file keywords. Added Tahoe II Input.
- Version 4.5.** Added more parameters to input file keywords *FACET, *ZONE, *BOUNDARY for the user to specify which output element group the newly created CSE's belong to. Also changed name of *MAKETRI to *SPLITELEMENT and added an option of which method is to be used. Added option *CONTACT. Added massless node set generation. Added option *RENUMBER.
- Version 5.0.** Reorganized program to store neighbor data globally. Added option to *MAPNODESET. Added more output data to *CONTACT.
- Version 5.1.** Updated program to Tahoe 3.0 libraries. Removed ability to read Tahoe I input data. Removed keywords *COORDINATES, *CONNECTIVITY, *NODESET, *SIDESET. Added Edge Type options to Zone Method.