True*Grid*® User's Guide For AUTODYN®

by

Robert Rainsberger

Version 2.3.0

XYZ Scientific Applications, Inc.

October 15, 2006

AUTODYN® is an explicit three-dimensional simulation code to model solids, fluids, and gases and their interaction. The focus in this manual will be on those features in ***TrueGrid***® that are specific to creating an AUTODYN® input file. The ***TrueGrid***® User's Manual covers the creation of a mesh and will not be covered in this manual. This manual is incomplete in another sense because it cannot be used as a substitute for a working knowledge of AUTODYN®.

Different fonts are used through out this manual to indicate their meaning. A literal is highlighted in **bold**. A symbol to be substituted with a literal or a number is *italicized*. A computer example uses the `Courier font`.

The following is a list of ***TrueGrid***® commands that can be used to produce features that are unique to an AUTODYN® file with the block structured format.

AUTODYN® feature	***TrueGrid***® commands
choose AUTODYN® format	**autodyn**
optimum block structure	**supblk**
void a block	**mt**, **mti**, **mtv**
delete a block	**de**, **dei**
transition between blocks	**bb**, **trbb**

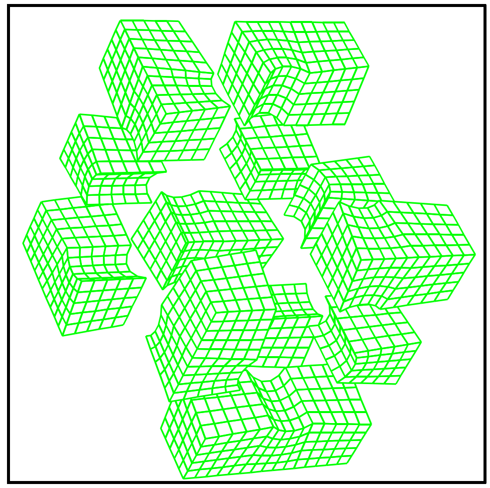
Boundary conditions, loads, properties, and analysis options are not created in ***TrueGrid***®. These properties are assigned within AUTODYN®.

It is essential that the **autodyn** command be issued before any parts are created. This is needed because the default data structure within ***TrueGrid***® is for finite elements. This version of the AUTODYN® format requires block structured meshes and once a part is formed, it is too late to convert to the block structured format.

When you create a multi-block part, many blocks are used to form one large grid for AUTODYN®. A grid is an array of nodes where each row has the same number of nodes. In a solids grid, each layer, formed by rows and columns, has the same number of columns in each layer. A grid cannot have holes in it due to deleted blocks. The method that forms these large grids is constrained by these holes in the mesh. Also, this method that forms these large grids may not form the ideal grouping of blocks in that there may be a different arrangement of blocks forming the grids that has few grids. You can prescribe how this grid formation algorithm forms large grids by using the **supblk** command. For every multi-block region you select with the **supblk** command, you will see a corresponding grid in the output file. When you use this command, be sure not to over lap these regions. Also, you do not have to cover the entire mesh with **supblk** commands. Any blocks not covered will be applied to grids automatically with the grid forming method. It is quite common to only specify a few grids with the **supblk** command. The automatic grid formation algorithm will form the remaining grids almost optimally. It is easy to check in the merge phase because each grid becomes a part. If you do not like the results, rerun the session file with additional **supblk** commands.

For example, three intersecting holes are formed by deleting the appropriate blocks. The automatic grid formation for 12 grids is shown in an exploded view so that you can see the boundaries of each grid.

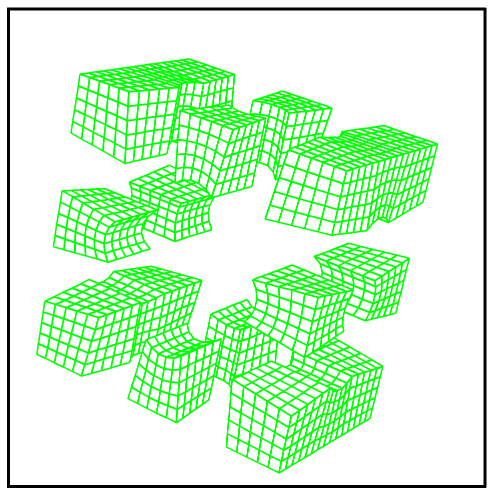
```
autodyn
sd 1 cy 4 4 4 1 0 0 1
sd 2 cy 4 4 4 0 1 0 1
sd 3 cy 4 4 4 0 0 1 1
block 1 6 11 16;1 6 11 16;1 6 11 16;
      1 3 5 7;1 3 5 7;1 3 5 7;
de 2 2 1 3 3 4
de 1 2 2 4 3 3
de 2 1 2 3 4 3
sfi 1 2 0 3 4; -2 -3; -2 -3;sd 1
sfi -2 -3; 1 2 0 3 4; -2 -3;sd 2
sfi -2 -3; -2 -3; 1 2 0 3 4;sd 3
endpart
merge
```



Automatic grids - Exploded view

Although there are no grid formations that have less than 12 grids for this mesh, there may be advantages to choose a different formation of grids. If the following **supblk** commands are added to the above commands, you would get a different set of grids.

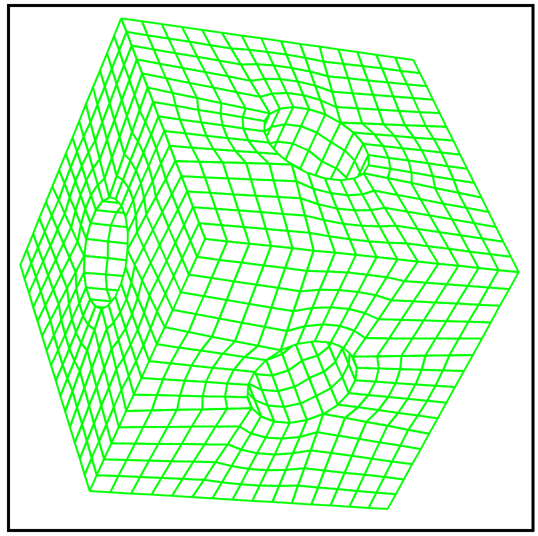
```
supblk 1 3 3 4 4 4
supblk 1 1 3 4 2 4
supblk 1 3 1 4 4 2
supblk 1 1 1 4 2 2
```



Supblk grids - Exploded view

There is an alternative to deleting blocks and creating holes in the mesh. When a block is assigned the material number 2, it will disappear from the graphics but not from the mesh. Instead, when the grids are formed, the elements that have been assigned the material number 2 will be flagged internally and in the AUTODYN® file as void elements. This technique can have a dramatic effect on the number of grids that are formed.

```
autodyn
sd 1 cy 4 4 4 1 0 0 1
sd 2 cy 4 4 4 0 1 0 1
sd 3 cy 4 4 4 0 0 1 1
block 1 6 11 16;1 6 11 16;1 6 11 16;
      1 3 5 7;1 3 5 7;1 3 5 7;
mt 2 2 1 3 3 4 2
mt 1 2 2 4 3 3 2
mt 2 1 2 3 4 3 2
sfi 1 2 0 3 4; -2 -3; -2 -3;sd 1
sfi -2 -3; 1 2 0 3 4; -2 -3;sd 2
sfi -2 -3; -2 -3; 1 2 0 3 4;sd 3
endpart
merge
```



Single grid with void elements

Two parts can be joined that have a different number of elements. This is done with the block transition feature. For complete instructions on the use of the **bb** and **trbb** commands, see the **True*Grid*®** User's Manual. The following is an example of the use of the **bb** and **trbb** commands to form a transition between two parts. This is not the only application of the **bb** command for AUTODYN® applications. It is common to use the **bb** command to glue parts together where ever it is convenient.

The transitions are always found in the part with the slave side of the interface. The roles of the lower and higher density parts can be switched from the example above, so that the master side has the higher density mesh.

```
autodyn
block 1 13;1 13;1 13;
 0 1 0 1 0 1
bb 1 1 2 2 2 2 1;
endpart
block 1 5;1 7;1 6;
 0 1 0 1 1 2
trbb 1 1 1 2 2 1 1 ;
endpart
merge
stp .0001
write
```

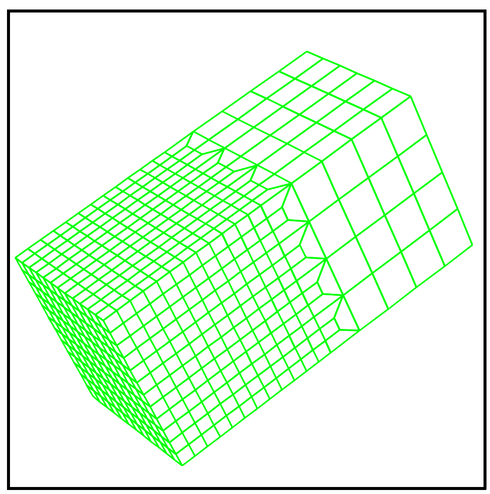


Figure 4 Two Way Transition