

Release notes for TrueGrid[®] version 400

1. The **sph** option for the **PLAN** command did not function. This has been corrected.
2. The new command **TRIQUADRATIC** at the beginning of a session causes all brick elements to be formed using 27 nodes. This may be reduced automatically if the brick has some merged nodes. This command also causes all quad shell elements to be formed from 9 nodes. These elements are valid for the output options **NEK5000**, **LSDYNA**, **ABAQUS**, **MPACT**, and (a modified) **NASTRAN**.
3. The **NERL** command now works with the **QUADRATIC** and **TRIQUADRATIC** commands. The manual, dialogue box, and the help menu documented this command incorrectly. It should read:

NERL *flag*

where the *flag* can be

- | | |
|----|--|
| 0 | for default numbering method |
| 1 | for ordered by increasing i-index last |
| -1 | for ordered by decreasing i-index last |
| 2 | for ordered by increasing j-index last |
| -2 | for ordered by decreasing j-index last |
| 3 | for ordered by increasing k-index last |
| -3 | for ordered by decreasing k-index last |

4. For the command **MTV**, the option (number 3) to change element material numbers for all elements within a volume defined by a rotated 2D curve has been extended to all non-convex volumes formed from rotated 2D curves. Also a minor bug was fixed for an obscure case when the points defining the 2D curve are much larger than the hexahedron elements.
5. A recently introduced bug to the **Hermite** surface type for the **SD** (surface definition) command caused extreme abuse of memory and has now been fixed.
6. In some rare cases when colors are set for each material, the color for a material within one part may be incorrect. This graphics bug has been fixed.
7. The labeling of 3D curves and points on a 3D curve are now available in the Fill graphics mode. This applies only to non-fast graphics. Related to this is a bug fix. This bug was evident when objects, such as 3D curves, were labeled and then a window overlapped the physical window. At this point, the labels could not be selected using the mouse.
8. A bug was fixed with the **WHILE** statement. A problem (infinite loop) would occur if there was a sequence of **WHILE** statements within a **WHEN**, **FOR**, or **WHILE** statement.
9. A minor bug was fixed for one-way transition for one face of one element of the transition block, the proper orientation was flipped (e.g. pressure or flux was applied to a face of the transition region).

10. A very minor bug was fixed related to the projection to trimmed IGES surfaces. It required the very unlikely event when a node is projected exactly to the interface between the triangles and quads that represent the surface.

11. A bug was introduced in version 3.13 regarding the nodal distributions along edges using the `res`, `as`, `drs`, and `das` commands. This new bug affected such nodal distributions along 3D curves across multiple edges on the mesh. This new bug has been fixed.

12. The merge phase **Pick>Sets** display and the **LABELS** options for the display of face sets from quadratic elements had a bug that is now fixed.

13. A significant number of features were added to support face properties, face sets, element sets, and graphics for quadratic and triquadratic elements.

- i) face and element sets formed in the part phase now include faces from transition elements.
- ii) properties on faces such as inlets, outlets, periodic conditions can now be viewed using the **CONDITION (CO)** command in the merge phase.

14. There is a new output format for the simulation code called NEK5000. It requires either quadratic or tri-quadratic hex elements. There are several new commands that support features for NEK5000. **NEK5000** selects the output format for the same named code. **NEKTOL** sets the tolerance used to decide if an edge of the NEK5000 mesh is non-linear. This applies only to midside-nodes. **NEKOPTS** is used to set NEK5000 analysis switches and parameters. The **GEOP** and **GEOPI** commands identify edges of a NEK5000 mesh with special curvature.

Syntax for these new commands are as follows:

NEK5000 (no arguments)

NEKTOL *tolerance*

GEOP *i1 j1 k1 i2 j2 k2 type property*

where *type* can be

i for i-edges

j for j-edges

k for k-edges

where *property* can be

arc for on an arc of a circle

fsp for a face on a sphere

GEOPI *i_list; j_list; k_list; type property*

where *type* can be

i for i-edges

j for j-edges

k for k-edges

where *property* can be

arc for on an arc of a circle

fsp for a face on a sphere

NEKOPTS [*parameter value*] ;

where the *parameter* and *value* can be

ifflow t or f

ifheat t or f

iftran t or f

ifnav1 t or f

ifnav2 t or f

ifnav3 t or f

ifnav4 t or f

ifnav5 t or f

ifnav6 t or f

ifnav7 t or f

ifnav8 t or f

ifnav9 t or f

ifnav10 t or f

ifnav11 t or f

iftmsh1 t or f

iftmsh2 t or f

iftmsh3 t or f

iftmsh4 t or f

iftmsh5 t or f

iftmsh6 t or f

iftmsh7 t or f

iftmsh8 t or f

iftmsh9 t or f

iftmsh10 t or f

iftmsh11 t or f

iftmsh12 t or f

ifaxis t or f

ifstrs t or f

iflomach t or f

ifmgrid t or f

ifmvbd t or f

ifchar t or f

ifsync t or f

ifuservp t or f

aneden *value*

anevis *value*

anebet *value*

anegth *value*

anepgr *value*

p06 *value*

anerho *value*

anecon *value*

p09 *value*

anefin *value*
anenst *value*
anedt *value*
aneioc *value*
aneiot *value*
aneios *value*
anepss *value*
aneaxi *value*
anegri *value*
aneint *value*
anenor *value*
anediv *value*
anehel *value*
anenps *value*
anetlr *value*
anetla *value*
anecou *value*
anetor *value*
anenab *value*
anemhd *value*
aneuse *value*
anenpe *value*
anenbe *value*
p33 *value*
p34 *value*
p35 *value*
anexmg *value*
anengr *value*
anenr2 *value*
anenr3 *value*
p40 *value*
p41 *value*
p42 *value*
p43 *value*
p44 *value*
p45 *value*
p46 *value*
p47 *value*
p48 *value*
p49 *value*
p50 *value*
p51 *value*
anehip *value*
p53 *value*
p54 *value*

p55 *value*
p56 *value*
p57 *value*
p58 *value*
p59 *value*
p60 *value*
p61 *value*
p62 *value*
anewds *value*
p64 *value*
p65 *value*
p66 *value*
p67 *value*
p68 *value*
p69 *value*
p70 *value*
p71 *value*
p72 *value*
p73 *value*
p74 *value*
p75 *value*
p76 *value*
p77 *value*
p78 *value*
p79 *value*
p80 *value*
p81 *value*
p82 *value*
p83 *value*
p84 *value*
p85 *value*
p86 *value*
p87 *value*
p88 *value*
p89 *value*
p90 *value*
p91 *value*
p92 *value*
p93 *value*
p94 *value*
p95 *value*
p96 *value*
p97 *value*
p98 *value*
p99 *value*

p100 *value*
p101 *value*
p102 *value*
p103 *value*
p104 *value*
p105 *value*
p106 *value*
p107 *value*
p108 *value*
p109 *value*
p110 *value*
p111 *value*
p112 *value*
p113 *value*
p114 *value*
p115 *value*
anenlx *value*
anenly *value*
anenlz *value*

15. Trimmed surface (IGES geometry) can be used to form mid-plane surfaces. This is with the **intp** option of the **SD** (surface definition) command. This is typically the method used to form variable thickness shell elements that fit between two surfaces.

Also, the merge phase graphics display of variable thickness shells,

CO *thic*

now works in the wire, hide, and fill graphics mode but not fast graphics.

As a reminder, use the **SSF/SSFI** command to project to the interpolated surface and to automatically calculate the shell thickness.

16. Two bugs were fixed that involved some combinations of a nodal bias, edge attachment, and projection to a surface where:

- 1.) nodal bias means one of: **RES, DRS, AS, DAS, NDS**
- 2.) edge attachment means one of: **CUR, CURS, CURE, CURF, EDGE, SPLINT, PATCH**
- 3.) surface projection means one of: **SF, SFI, PATCH**

The problem was, in some cases, the bias was ignored.

17. An obscure bug was fixed that only occurred when zeros were in the index progression of the **BLOCK** or **CYLINDER** commands. It also required a projection (**SF** or **SFI** command) that crossed over the gap region caused by the zero in the **BLOCK** or **CYLINDER** command. In this case, only

the lower portion of the index progress of the projection command was performed. This only occurred with interior nodes of the face being projected.

18. Another obscure bug was fixed with the **INTERRUPT** command if the **INTERRUPT** command did not start in the first column of the text. It has various symptoms, but always ended with TG crashing.

19. A bug was fixed in the **CFX5** output option. The problem occurred when the **CFXSD** command was used in the merge phase.

20. Incorrect data was written to the **ABAQUS** output file if no materials were defined. An ***ORIENTATION** card was being written for each section card. This has been fixed. Also, a format bug for a point mass for **ABAQUS** has been fixed.

21. The **INTP** option of the **SD** command was improved. This command creates a mid-plane surface between 2 other surfaces. There are 2 improvements.

- i. the accuracy of the mid-plane surface has been improved
- ii. the allowed surfaces acting as the parent surface now includes composite surfaces.

22. The **SSF** and **SSFI** commands have been extended to include 8 and 9 noded shell elements. Also, the method has been improved so that all variable thickness shell elements that with a common node will have the same thickness at that node. Only **LSDYNA** output will be impacted by this last change and is required for consistency.

23. The **CO** command in the merge phase now has an improved **thic** option with the additional showing of the variable thickness in fill graphics. This applies to 3, 4, 6, 7, 8, and 9 node shell elements.

24. An obscure bug was fixed in the **CURD** command defining a cubic spline curve using the endpoint and end derivative of a previously defined curve to start the new cubic spline curve and is indicated by a first endpoint flag of 2 or 3. If the coordinates of the endpoint of the previous curve matches the first knot point of the new cubic spline curve, a division by zero occurred and **TrueGrid**[®] would crash.

25. A new command in the merge phase was added to specify distributed surface loads in an **ABAQUS** model. Syntax for this command is as follows:

DSLOAD *face_set load_case* [*options*] *type parameters*

where *options* can be any of the following:

CR to choose constant resultant
NF to choose no follower forces

OR *local_sys_#* for orientation

RN *node_#* for reference node

where *type* and associated *parameters* must be one of

P <i>ampl</i>	for uniform pressure
PNU <i>ampl</i>	for nonuniform pressure
SP <i>ampl</i>	for stagnation pressure
VP <i>ampl</i>	for viscous pressure
HP <i>ampl z0 zp</i>	for hydrostatic pressure
TRVEC <i>ampl xn yn zn</i>	for uniform general traction
TRSHR <i>ampl xn yn zn</i>	for uniform shear traction
EDLD <i>ampl xn yn zn</i>	for uniform general edge traction
EDMOM <i>ampl</i>	for uniform edge moment
EDNOR <i>ampl</i>	for uniform normal edge traction
EDSHR <i>ampl</i>	for uniform shear edge traction
EDTRA <i>ampl</i>	for uniform transverse edge traction

For the meaning of these *options*, *types*, and *parameters*, see the **ABAQUS** manual under the **DSLOAD** command.

A *SURFACE is formed in the **TrueGrid**[®] output file to **ABAQUS** for every face set built in **TrueGrid**[®]. The face set name can be used as an argument in this new command, which then creates a distributed load applied to the faces of the set. The load case in this command identifies the set of distributed loads so that they can be associated with a step (see **ABAQSTEP**). This separation of the **ABAQUS** step and the distributed loads makes it possible to use the same set of loads in different steps of the analysis. If a load curve is defined (see the **LCD** command) by the same number as the load case, then the set of distributed loads will be ramped according to the load curve.

For completeness, it is noted that the commands **PR** (pressure), **B** (boundary constraints), **FC** (force), **MOM** (moments), **FD** (fixed displacement), **FV** (fixed velocity), **ACC** (acceleration), and **TRACT** (traction) include a load case for **ABAQUS** loads so that they can also be associated with an **ABAQUS** step. If a load curve is defined using the same identification number, then the associated load will be ramped according to the load curve.

This command assigns distributed loads to both linear and quadratic elements.

Use the **TrueGrid**[®] **LSYS** command to create a local coordinate system and associate this system with the **DSLOAD** using the **OR** option.

The **ABAQSTEP** command, used to associate conditions to different analysis steps in the simulation, has a new option to specify an association of the general distributed surface command **DSLOAD** to an analysis step. That option is:

ABDLOAD BLC *load_case_#* **GE** ;

Initial velocities are defined using the following commands:

1. **VELOCITY** in the merge or control phase
2. **VELOCITY** in the part phase

3. **VE** in the merge phase
4. **VE** and **VEI** in the part phase

Item 1 is the global default velocity

Item 2 is the default velocity only for the part where it is used - it over rides the global default (item 1).

Items 3 and 4 above only define nodal initial velocities in regions of a part and they override both items 1 and 2.

The **CFD** process is now available in the **ABAQSTEP** command. If the first step invokes the **CFD** option, then the velocity initial conditions can be defined using the **VELOCITY** commands (items 1 and 2 above). In this case the velocity commands in items 3 and 4 will be ignored, because they only apply to nodes.

26. The **PR/PRI**, **CVT/CVTI** (**ANSYS** only), in the part phase and the **PR**, **CVT** (**ANSYS** only), and **TRACT** commands in the merge phase for the **ABAQUS**, **ANSYS**, and **NASTRAN** output options now includes quadratic elements.

27. A bug in the **PB** command with cylinder parts was fixed.

28. A new command finds the center and radius of a sphere from 4 points that are not co-linear nor co-planar.

SP4PT *x1 y1 z1 x2 y2 z2 x3 y3 z3 x4 y4 z4*

29. The command **LABELS PARTS** highlights parts in wire, hide, and fill graphics mode in both the part and merge phase.

30. The **NSETC**, **ESETC**, and **FSETC** commands can be used to set the title for node sets, element sets, and segment sets for **LSDYNA**. The **VFL** and **VFLI** commands now generate variable flux for **LSDYNA**. The amplitude is calculated for each node of each face of the selected region.

31. Use the **QUADRATIC** command before generating the mesh to get 20 node hexahedron elements for **LSDYNA**. Use the **TRIQUADRATIC** command to get 27 node hexahedron elements for **LSDYNA**. Note that this feature is only preliminary since these new types of elements are not yet generally available. These **TrueGrid**[®] features may have to change when the new version of **LSDYNA** is released. Also note, there are no facial properties in **LSDYNA** available at the time this feature was implemented for these elements.

32. Two bugs were fixed with the **COSURF** command. In some cases, it included other composite surfaces in the formation of a composite surface. Also, it worked for only a small list of surfaces.

33. An obscure bug was fixed when selecting a face set in the part phase. If the region that was selected started with a deleted region, it may have selected the wrong faces of the selected elements. This only affected **ABAQUS** and **ANSYS** outputs.

34. A pair of new commands locates the element that is closest to a point.

AJEL *element_type x y z*

SAJEL *element_type x y z*

where the *element_type* can be:

1d	for linear beams
2d	for linear shells
3d	for linear bricks
1dq	for quadratic beams
2dq	for quadratic shells
3dq	for quadratic bricks

AJEL prints the element number of the element that is closest to the point. Both **AJEL** and **SAJEL** (s for silent) assigns the automatic parameter %element to the element number that is closest to the point.

These commands are only available in the merge phase. The measurement is made from the point at (x,y,z) to the center of the approximate mass of the element. Both linear and quadratic elements are measured using the same number of nodes to calculate the center of mass. For example, the 8 nodes of a linear hexahedron element are used to calculate the center of mass. If the same element is measured as a quadratic brick element (i.e. the **QUADRATIC** or **TRIQUADRATIC** command was issued before the part was created) then only the 8 linear nodes at the corners of the hexa element are used to calculate the center of mass. This way, the same numbered element will be chosen no matter if the element is linear or quadratic.

35. The **VERBATIM/ENDVERBATIM** command data that is written to some of the output formats can include the '{' and '}' characters if they are preceded by the back slash character '\'. For example, if the following verbatim command were:

```
verbatim
This is a test of the \{ and \} characters
endverbatim
```

then the output file from **TrueGrid**[®] would contain the following line:

```
This is a test of the { and } characters
```

Also, the same three lines including the back slashes would appear in the tsave file.

36. A minor bug was fixed in the fill graphics within the part phase. On rare occasions, some of a previously generated part (i.e. not the present part) would have some of the front faces of the part clipped. This would only happen when the part had significant curvature. Another bug was fixed for the fill graphics in the part phase. The problem was that in some cases, when there was an automatic refresh of the physical window in the part phase, there would be a floating point error.

37. When creating a part using the **CYLINDER** part command, one can pick a node and then using the F7 function key to have the coordinates of the node printed. The default coordinate is cylindrical. However, one can choose the global button so that the coordinates are printed as Cartesian coordinates. In the past this all had to be done only when the **PICK Panel** was selected. Now, the same choice of coordinate system for printing the coordinates applies to the **DISPLAY List Panel** and the **LABELS Panel**, although one must first choose the **NODES** button and the desired coordinate system while the **PICK Panel** is displayed.

38. A parameter can be used as a name, such as the name of a node, face, or element set. The integer value will be used as the symbol. In other words, the character string will be the integer value of the parameter. As a reminder, the integer value of a parameter is the truncation of the floating point value of the parameter. For example:

```
para nb 2;  
nset 1 1 2 2 2 2 = %nb
```

means that a new node set is formed from the nodes in a face of the mesh and that set is named '2'.

39. The **READMESH** command for the **LSDYNA KEYWORD** format now reads the following cards and associated data:

```
*SET_NODE  
*SET_NODE_LIST  
*SET_NODE_TITLE  
*SET_NODE_TITLE_LIST  
*SET_SEGMENT  
*SET_SEGMENT_TITLE  
*SET_BEAM  
*SET_BEAM_TITLE  
*SET_SHELL  
*SET_SHELL_LIST  
*SET_SHELL_TITLE  
*SET_SOLID  
*SET_SOLID_TITLE  
*SET_TSHELL  
*SET_TSHELL_TITLE
```

The node sets numbers are offset depending on what numbered node sets already exist when the **READMESH** command is issued. The segment sets numbers are offset depending on what numbered segment sets already exist when the **READMESH** command is issued. The element sets are also offset and with a prefix to the number: BM for beam, SH for shell, BR for brick, and TS for thick shell.

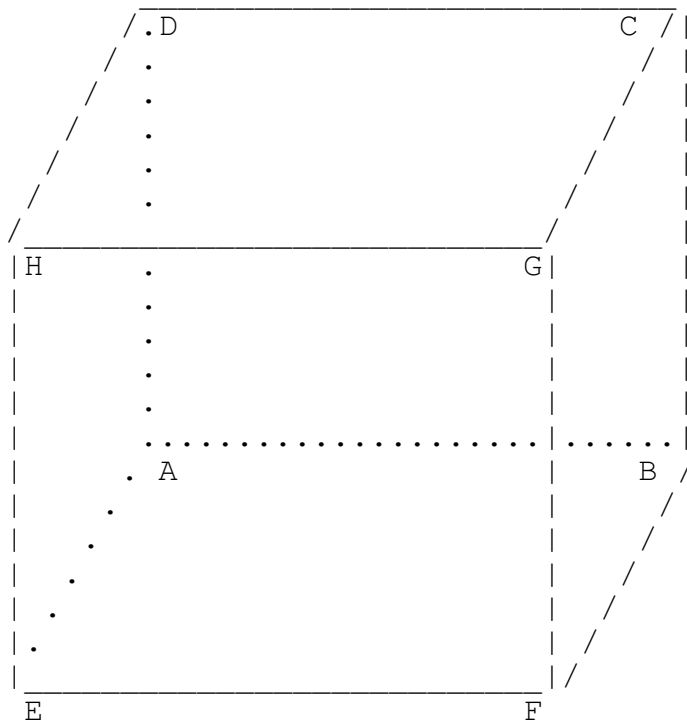
40. The default face IDs of elements depends on the element geometry after nodes have been properly merged so that **TrueGrid**[®] can determine the element geometry (i.e. hex, wedge, or tet). The

face IDs can be seen when in the merge phase by using the **LA FACESEL** command. The face IDs differ among the different simulation codes. You need to interpret what you see in **TrueGrid**[®] graphics to correspond with the simulation code. This interpretation of the face ID numbers is only needed when using the **LA FACESEL** command in the merge phase. When **TrueGrid**[®] writes an output file for a particular simulation code, it does the translation of the face ID numbers appropriate for that simulation code.

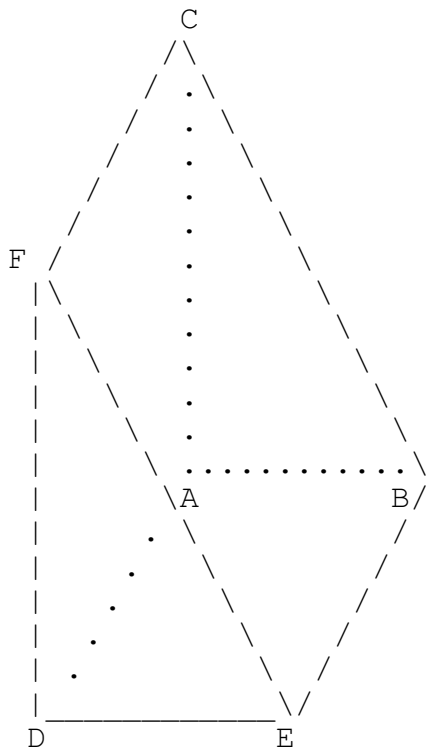
Almost all simulation codes that use face IDs agree on how to identify faces of a hex element. But there is significant variation among the different simulation codes on how to identify faces of a wedge or tet element. The diagrams below show how **TrueGrid**[®] identifies faces for all three element geometries. They are all based on the order of nodes that define the element. The sequence of nodes are identified on the left diagrams. The corresponding faces are identified in the table to the right of the diagram. These rules apply for **LINEAR**, **QUADRATIC**, and **TRIQUADRATIC** elements.

This is a change from the past versions of **TrueGrid**[®]. In the past, when a wedge or tet element was formed from a degenerate hex, the face ID numbers from the **LA FACESEL** command showed the face ID numbers that were inherited from the hex element that formed the wedge or tet.

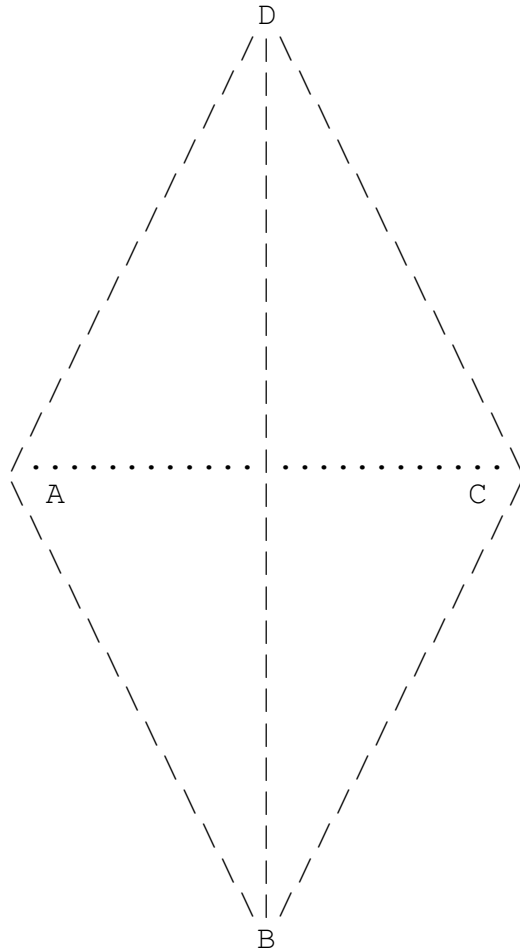
NOTE: Dotted line means it is hidden behind faces of the element.



- 1 A-B-C-D
- 2 A-E-F-B
- 3 B-F-G-C
- 4 D-C-G-H
- 5 A-D-H-E
- 6 E-H-G-F



- 1 A-B-C
- 2 D-E-F
- 3 A-D-E-B
- 4 A-C-F-D
- 5 B-E-F-C



- 1 A-B-C
- 2 A-D-B
- 3 B-D-C
- 4 C-D-A

Faces are oriented. If you use fill graphics to view the face set after issuing the **LA FACESEL** command, the faces in the face set will be drawn as red. A subtle feature in the graphics will indicate if the face is oriented in towards the center of the parent element or if it is pointing outward. An inward face will be darkened as if the light sources in the picture are not reflecting off of the face. If it is pointing outward, the face will look much brighter. You need to rotate the mesh so that the faces are facing you to see this difference.

The orientation of a face in a face set can be confirmed using the **LA FACESET** command. Arrows are placed at the center of each face of the face set to indicate orientation. This is true only in the **HIDE** graphics mode.

The face labels will only be visible if there is room for the label and if all corners of the face are visible.

41. A bug was fixed in the **KIVA4** output file. The **MTABLES** zero flag was missing. Also, a compiler optimization bug was discovered in the KIVA output. This caused the cell and face type table not to be written in the output file. This was not detected, evidently, because testing was done without optimization.

42. The transformation that can be applied to either the master or slave side or both of a block boundary interface (**BB** and **TRBB** commands) was not implemented when the master and slave were both a single vertex. This has been corrected.

43. A bug was fixed in the graphics when quadratic (2nd order) beams elements were generated using the **BM** command in the merge phase.

44. The precision and graphics description has been added to the version and date information.

Previously, the version information for all executables was

```
TRUEGRID VERSION 3.1.4  DATE  1/27/17
```

Now on Windows the following appears:

tg:

```
TRUEGRID VERSION 3.1.4 (32 bit, openGL graphics)  DATE  1/27/17
```

tgw:

```
TRUEGRID VERSION 3.1.4 (32 bit, Native WINDOWS graphics)  DATE  1/27/17
```

tgd:

```
TRUEGRID VERSION 3.1.4 (64 bit, openGL graphics)  DATE  1/27/17
```

tgdw:

```
TRUEGRID VERSION 3.1.4 (64 bit, Native WINDOWS graphics)  DATE  1/27/17
```

and on UNIX/LINUX/OSX the following appears:

tg:

```
TRUEGRID VERSION 3.1.4 (32 bit, openGL graphics)  DATE  1/27/17
```

tgx:

```
TRUEGRID VERSION 3.1.4 (32 bit, X-windows graphics)  DATE  1/27/17
```

tgd:

```
TRUEGRID VERSION 3.1.4 (64 bit, openGL graphics)  DATE  1/27/17
```

tgdx:

```
TRUEGRID VERSION 3.1.4 (64 bit, X-windows graphics)  DATE  1/27/17
```

45. One way and two way transitions has been added for 27 node hexahedron elements.

46. The **SW** and **ALT** switches in the **TRBB** command have been fixed for linear and quadratic (20 and 27 node) elements. Also, the **PRTRND** command found in the Interface menu now defaults to "on". This command determined if an exterior interpolated (second order) node as part of the slave side **TRBB** could be projected to the same surface(s) that the first order nodes are projected to. This was defaulted to "off" in the past because the only nodes in a transition were first order. With quadratic and triquadratic transitions, the situation is quite different and requires projection in most cases. It may slow things down just a little and it never hurts to have it on for all cases.

47. The new part phase command **SHOFF**, is available for moving the shell reference surface from the nodal points that define the shell. The syntax for this command is:

```
SHOFF i1 j1 k1 i2 j2 k2 offset
```

SHOFFI *i-list; j-list; k-list; offset*

where the *offset* value is a distance in the global coordinate system.

Use the **CO** command with the **THIC** option in the merge phase to view the shell element thickness, the location of the shell orthogonal to the reference surface of the shell, as well as the shell offset defined by this command. Since the material model shell location option (**SHLOC**) and the **SHOFF/SHOFFI** shell offset command have similar effects on the location of the shell w.r.t. the shell reference surface, one would usually only use **SHLOC** or **SHOFF**. If both are defined, both will affect the location of the shell in the graphics. It is unknown if the sum of the two options will both affect the results in the simulation code.

This is available for the following output options: **NASTRAN**, **LSDYNA**.

48. There was a bug involving **TRBB** and **SAVEPART** and when the master side of the block boundary interface (**BB** command) was defined in a previous part and when there is at least one master block boundary within the present part. This bug has been fixed.

49. A bug was fixed for **LS-DYNA** using the **VD** command to define a box. The 6 values that define the box (***DEFINE_BOX**) had been permuted.

50. Two bugs in the fast graphics were fixed. In some cases, when triangular shell elements were generated, a triangle would not be displayed in fast graphics. Also, the **ETD** command to control which element types were included in the display did not function for beam elements in fast graphics.

51. Stand-alone feature of **TrueGrid**[®] has been modified and now has the following restrictions:

- a. The machine must not be virtual.
- b. The operating system on the machine must be Windows
- c. **TrueGrid**[®] must be installed in c:\TrueGrid and the folder must allow user READ/WRITE permission.
- d. The licences must be for only one seat of **TrueGrid**[®].
- e. If one exits from **TrueGrid**[®] and restarts **TrueGrid**[®], it may take up to 10 seconds to restart.

52. The following command was introduced in version 234, but it was not added to the manuals. So it is being reintroduced as though it were new. **TRPRT** is a command in the merge phase which can transform a part. If nodes have been merged, they are unmerged. After the part is transformed, a new merge command must be issued to remerge nodes. This command can be found in the Misc. menu. The command is:

TRPRT *part_# trans ;*

where *trans* is a transformation from the product from left to right of the following operators

MX *x_offset*

translate in the x-direction

MY *y_offset*

translate in the y-direction

MZ <i>z_offset</i>	translate in the z-direction
V <i>x_offset y_offset z_offset</i>	translate in any direction
SCV <i>x1 y1 z1 distance</i>	translate in a direction for a given distance
DV <i>x1 y1 z1 x2 y2 z2</i>	translate by the difference vector
RX <i>theta</i>	rotate about the global x-axis
RY <i>theta</i>	rotate about the global y-axis
RZ <i>theta</i>	rotate about the global z-axis
RAXIS <i>angle x0 y0 z0 xn yn zn</i>	rotate about any axis in global coordinates
RXY	reflect about the global xy-plane
RYZ	reflect about the global yz-plane
RZX	reflect about the global zx-plane
TF <i>origin x-axis y-axis</i>	transform to a local sstem by 3 points
where each of the arguments consist of a coordinate type followed by coordinate information:	
RT <i>x y z</i>	Cartesian coordinates
CY <i>rho theta z</i>	cylindrical coordinates
SP <i>rho theta phi</i>	spherical coordinates
PT <i>c.i</i>	label of a labeled point from a 3D curve
PT <i>s.i.j</i>	label of a labeled point from a surface
FTF <i>1st_origin 1st_x-axis 1st_y-axis 2nd_origin 2nd_x-axis 2nd_y-axis</i>	transform from one local system to another
where each of the arguments consist of a coordinate type followed by coordinate information:	
RT <i>x y z</i>	Cartesian coordinates
CY <i>rho theta z</i>	cylindrical coordinates
SP <i>rho theta phi</i>	spherical coordinates
PT <i>c.i</i>	label of a labeled point from a 3D curve
PT <i>s.i.j</i>	label of a labeled point from a surface
INV	invert the present transformation
CSCA <i>scale_factor</i>	scale in all 3 coordinates
XSCA <i>scale_factor</i>	scale in the x-coordinate
YSCA <i>scale_factor</i>	scale in the y-coordinate
ZSCA <i>scale_factor</i>	Scale in the z-coordinate

For example:

```
sd 1 cy 0 0 0 0 1 0 1
block 1 11;1 11;1 11;-1 1 -1 1 -1 1
sfi -1 -2;; -1 -2;sd 1
block 1 11;1 11;1 11;-1 1 1 3 -1 1
sfi -1 -2;; -1 -2;sd 1
merge
stp .001
trprt 2 mx 1 my -1 rz 15 mx -1 my 1;
stp .001
```

In this example, 2 simple cylinder parts are generated, one on top of the other. In the merge phase, they are merged and 121 nodes get merged. Then the **TRPRT** command is applied. The nodes are no longer merged. A new merge command is issued to merge nodes and only one node gets merged.

Notice that the rotation of the second part uses the point at (-1,1,0) to apply a rotation of 15 degrees about the z-axis. This is done by first translating the part so that the point that was at (-1,1,0) is translated to (0,0,0). Then the rotation is applied. As a last step, the part is translated so that the point that is now at (0,0,0) will be translated back to (-1,1,0).

53. The new output command **STL** writes a STL file in ASCII or binary format. This is the data file sometimes used for StereoLithography (frequently called additive manufacturing). It can also be used to extract the exterior of a mesh and turn it into a surface. Only the visible faces of the mesh are placed into the STL file.

If the STL is to be used for a 3D printer, choose the option for 3D printers. This is needed because the surface in the STL file must have coordinates in all three directions starting at zero. On the other hand, if the surface is being extracted for other reasons and the part is not to be translated to zero, then choose the option for exterior mesh surface extraction.

STL mode

where *mode* can be

- | | |
|----------|---|
| 1 | for ASCII 3d Printing |
| 2 | for Binary 3D Printing |
| 3 | for ASCII exterior mesh surface extraction |
| 4 | for Binary exterior mesh surface extraction |

These files can get large and the binary format will be considerably shorter than the ASCII format file with the same data. The user is responsible for making the positive normals of shell elements all point outward. If this file is to be used with a 3D printer. All brick faces are automatically pointing outward. It is also the responsibility of the user to make the surface water tight by merging nodes.

Use the **ETD** command to select certain element types in the picture. Use the various ways of selecting parts and materials to be drawn in the graphics. Only the faces of the element types, materials, and parts shown in the picture will be recorded in the STP file.

It is possible to write multiple STL files, but each needs a unique name or only the last STL file will survive. Use the **MOF** command to change the STL file name between writes. After selected the parts, materials, element types, and the file name, issue the write command. As usual in **TrueGrid**[®], you must be in the merge phase to write an output file.

To read the STL file into **TrueGrid**[®], use the ASCII or binary **STL** option in the **SD** command.

54. A bug was fixed that would cause **TrueGrid**[®] to crash. This would occur if the **QUADRATIC** command was used to form 2nd order quadratic elements. If the **Fast Graphics** was invoked in the part phase, a floating point error might occur when the **Fast Graphics** was turned off. This was

random, so it did not occur every time these steps were taken. A second bug in **Fast Graphics** was fixed, which only effected the part phase. Sometimes the picture would be cropped.

55. A bug was fixed when a slave side of an intrapart **BB** command was issued where if the slave side was larger than a single region, the edges of the mesh that connect to the intermediate vertices of the slave side may not be interpolated properly. This would occur only if the slave side intermediate vertices were not ideally initialized. Also, a bug was fixed when an intrapart block boundary slave side had a normal offset transformation.

56. A bug was fixed having to do with facial boundary conditions and face sets of hexahedral elements. The conditions for this bug were either a selection of a face set or **CFX5** boundary condition on a k-face. Also, the part phase had to generate inverted elements (i.e. the determinant of the Jacobian had to be negative). The typical cause for negative determinant of the Jacobian is one or all three coordinate lists in the block or cylinder command are not monotonically increasing. In this case, if the minimum face of an element was selected (face ID 1 in **TrueGrid**[®]) would be incorrectly identified as face ID 6 (maximum k-face of the element) and vice versa.

This bug has other implications. If one is generating an **ABAQUS** or **EXODUSII** file, the face sets under the above conditions would be wrong.

57. The READMESH for NASTRAN files has been improved. The data for the following NASTRAN cards are entered into the **TrueGrid**[®] internal data base:

CHEXA, CTRETRA, CPENTA, CPTRAM, CPYRAM, CHEXA27, CQUAD4, CQUADR, CQUAD8, CTRIA4, CTRIAR, CTRIA8, CTRIA6, CTRIA3, CTRIAR, CBEAM, CBEND, CROD, CTUBE, CBAR, CELAS1, CELAS2, CSHEAR, FORCE, FORCE1, FORCE2, GRID, GRDSET, CORD, MAT1, MAT2, MAT4, MAT5, MAT8, MAT9, MAT10, MATT9, MOMENT, MOMENT1, MOMENT2, PLOAD, PSHEAR, PELAS, RBE2, PBEAM, PBAR, PTUBE, PROD, PBCOMP, PBEND, PCOMP, PSHELL, PSOLID.

58: There are additional cards generated by **TrueGrid**[®] for the **NASTRAN** output option. All generated features are listed here for completeness. Each keyword is followed by the **TrueGrid**[®] command(s) needed to generate the NASTRAN cards.

SOL: NASTRAN
CEND: NASTRAN
BEGIN BULK: NASTRAN
ENDDATA: NASTRAN
TITLE: TITLE
TEMP: TE/TEI
TEMPD: GTEMP
SUBCASE: PR/PRI, FC/FC, FCS/FCSI, FCC/FCCI, LCD/FLCD
PBAR: BSD, IBM/IBMI, JBM/JBMI, KBM/KBMI, BM
PROD: BSD, IBM/IBMI, JBM/JBMI, KBM/KBMI, BM
PTUBE: BSD, IBM/IBMI, JBM/JBMI, KBM/KBMI, BM

PBEAM: BSD, IBM/IBMI, JBM/JBMI, KBM/KBMI, BM
PBEAM3: BSD, IBM/IBMI, JBM/JBMI, KBM/KBMI, BM
PBCOMP: BSD, IBM/IBMI, JBM/JBMI, KBM/KBMI, BM
PBEND: BSD, IBM/IBMI, JBM/JBMI, KBM/KBMI, BM
CBARAO: BSD, IBM/IBMI, JBM/JBMI, KBM/KBMI, BM
GRID: BLOCK, CYLINDER, B/BI
CBEAM: IBM/IBMI, JBM/JBMI, KBM/KBMI, BM
CBEAM3: IBM/IBMI, JBM/JBMI, KBM/KBMI, BM
CBEND: IBM/IBMI, JBM/JBMI, KBM/KBMI, BM
CROD: IBM/IBMI, JBM/JBMI, KBM/KBMI, BM
CTUBE: IBM/IBMI, JBM/JBMI, KBM/KBMI, BM
CHEXA: BLOCK, CYINDER, LINEAR, QUADRATIC
CPENTA: BLOCK, CYINDER, LINEAR, QUADRATIC
CTRETRA: BLOCK, CYINDER, LINEAR, QUADRATIC
CHEXA27: BLOCK, CYINDER, TRIQUADRATIC
CPYRAM: BLOCK, CYLINDER, LINEAR, QUADRATIC
CQUAD4: BLOCK, CYLINDER, LINEAR, QUADRATIC
CQUAD8: BLOCK, CYLINDER, LINEAR, QUADRATIC
CQUADR: BLOCK, CYLINDER, LINEAR, QUADRATIC
CTRIA3: BLOCK, CYLINDER, LINEAR, QUADRATIC
CTRIAR: BLOCK, CYLINDER, LINEAR, QUADRATIC
CTRIA6: BLOCK, CYLINDER, LINEAR, QUADRATIC
CSHEAR: BLOCK, CYLINDER, LINEAR, QUADRATIC
FORCE: FC/FC, FCS/FCSI, FCC/FCCI, NDL, LL
PLOAD: PR/PRI
MAT1: NASTMATS
MAT2: NASTMATS
MAT3: NASTMATS
MAT4: NASTMATS
MAT5: NASTMATS
MAT8: NASTMATS
MAT9: NASTMATS
MATS1: NASTMATS
MATT1: NASTMATS
MATT2: NASTMATS
MATT4: NASTMATS
MATT5: NASTMATS
MATT9: NASTMATS
PSOLID: NASTMATS
PSHELL: NASTMATS
CORD2R: NASTMATS
PCOMP: NASTMATS
PSHEAR: NASTMATS
MOMENT: MOM
CMASS2: PM, NPM

MPC: JD, JT, MPC
PELAS: SPD
CELAS1: SPDP, SPRING
SET: NSET/NSETI, ESET/ESETI
TIC: VE, DIS
RBE2: RBE
RBE3: RBE
RROD: RBE
RBAR: RBE
RTRPLT: RBE
TAGDMP1: LCD/FLCD
TABLED1: LCD/FLCD
TABLED2: LCD/FLCD
TABLED3: LCD/FLCD
TABLEM1: LCD/FLCD
TABLEM2: LCD/FLCD
TABLEM3: LCD/FLCD
TABLES1: LCD/FLCD
TABLEST: LCD/FLCD
TABRND1: LCD/FLCD
TABLED4: LCD/FLCD
TAGLEM4: LCD/FLCD
DEFORM: DEFORM
GRAV: NASTOPTS
NLPARM: NASTOPTS
RANDPS: NASTOPTS
RANDT1: NASTOPTS
TSTEP: NASTOPTS
TSTEPNL: NASTOPTS
SPCD: FD/FDI, FDC/FDCI, FDS/FDSI, LCD/FLCD
SPC: FD/FDI, FDC/FDCI, FDS/FDSI, LCD/FLCD
SPC1: B/BI
SPCADD: B/BI
PLOAD4: TRACT

59. The integration rule, **VIGQ**, for the measure command has been changed so that one can choose from 1 to 5. The default is 2. This applies to the following measure options: **volume**, absolute volume (**avolume**), **Jacobian**, change in volume (**dvi**, **dvj**, **dvk**), stiffness (**stiffn**), and sub-element volume (**subvol**).

Also, in the past only the shape function for a hexahedron element was used to measure, since the hexahedron element was the primary element being generated. Since it is now easy to generate tetrahedron elements from the hexahedron element, the **MEASURE**, **MEA**, and **MEAI** commands now uses the proper shape function for each element type. The element types now generated by **TrueGrid**[®] has been extended to: 8 node hexahedron, 20 node hexahedron, 27 node hexahedron, 6

node prism, 15 node prism, 4 node tetrahedron, 10 node tetrahedron, 3 node shell, 4 node shell, 6 node shell, 8 node shell, 9 node shell, 2 node beam, and 3 node beam. Be sure to issue a merge nodes command, such as **STP**, before the first **MEASURE**, **MEA**, or **MEAI** command, if you are generating anything except hexahedron elements.

60. The new command **APAINFO** lists all of the automatic parameters and their values. Below is an example. Note that the automatic parameters associated with the part phase (the first 6 below) will only appear in the part phase. The same information can be gotten, one automatic parameter at a time, using the **DC** command (desk calculator).

```

List of full i-indices of the part
      idxlist =      1      3      5      7

List of full j-indices of the part
      jdxlist =      1      3      5      7      9      11      13

List of full k-indices of the part
      kdxlist =      1      3      5      7      9      11      13
15      17
          19      21      23      25      27      29      31
33      35
          37      39

Maximun reduced index in the i-direction
      maxrudi =      4
Maximun reduced index in the j-direction
      maxrudj =      7
Maximun reduced index in the k-direction
      maxrudk =      20
Next surface number
      nextsrf =      1
Next 3D curve number
      nextcrv =      1
Next load curve number
      nextlc  =      1
Next 2D curve number
      nextln  =      1
Next material number

```

```
nextmat = 2
Next Block Boundary ID number from BB command

nextbb = 1
Next part number

nextprt = 2
Node number result from ajnp/sajnp and PICK>NODE functions

node = 0
Element number result from ajel/sagel functions

element = 0
The functions ipil, trapt, tricent,
bulc, circent, project, ptcor, pptcd, and nodcor/snodcor
record their results in the following three parameters:
X-component of projection

xprj = 0.000000E+00
Y-component of projection

yprj = 0.000000E+00
Z-component of projection

zprj = 0.000000E+00
The functions circent, sp4pt, and project
record their results in the following three parameters:
X-component of a normal

xnrm = 0.000000E+00
Y-component of a normal

ynrm = 0.000000E+00
Z-component of a normal

znrm = 0.000000E+00
X-component from cross product crprod function

xcrprod = 0.000000E+00
Y-component from cross product crprod function

ycrprod = 0.000000E+00
Z-component from cross product crprod function

zcrprod = 0.000000E+00
```

Inner product from inprod function

inprod = 0.000000E+00
Distance result from distance, circent, and sp4pt functions
distance= 0.000000E+00
Subtended angle result from subang function

subang = 0.000000E+00
Minimum mesh quality measure

minmea = 0.000000E+00
Maximum mesh quality measure

maxmea = 0.000000E+00
Mean of mesh quality measure

mean = 0.000000E+00
Standard deviation of mesh quality measure

standev = 0.000000E+00
Irrational constant circle circumference/diameter

pi = 3.141593E+00
Minimum x-coordinate of the graphics containing box

xboxmin = 1.000000E+00
Minimum y-coordinate of the graphics containing box

yboxmin = 1.000000E+00
Minimum z-coordinate of the graphics containing box

zboxmin = 1.000000E+00
Maximum x-coordinate of the graphics containing box

xboxmax = 7.000000E+00
Maximum y-coordinate of the graphics containing box

yboxmax = 1.300000E+01
Maximum z-coordinate of the graphics containing box

zboxmax = 3.900000E+01
Next node number (befor merging)

nextnode= 1
Next linear brick

nextlbrick= 1
Next linear shell element

nextlshell= 1


```

Next linear beam

    nextlbeam =      1
Next quadratic brick element

    nextqbrick=      1
Next quadratic shell element

    nextqshell=      1
Next quadratic beam element

    nextqbeam =      1

```

61. There are 4 new element quality measures in the merge phase. They are node based. In each case the Jacobian is calculated at a node for every node in the element. The measure is applied to the Jacobian. All of the measures associated with a node are averaged. The keyword for these new measures (options to the **MEASURE** command) are:

ANJAC means that for each node, the determinant of the Jacobian is calculated for each element that contains this node, and the average is calculated.

ANNJAC means that for each node, the determinant of the relative Jacobian is calculated for each element that contains this node, and the average is calculated. The relative Jacobian is the Jacobian divided by the cube of the middle singular value of the Jacobian. In plane English, size of the element is factored out.

ANSTIF means that for each node, the stiffness of the Jacobian is calculated for each element that contains this node, and the average is calculated.

WANNJAC means that for each node, the determinant of the relative Jacobian is calculated for each element that contains this node, and the weighted average is calculated. The relative Jacobian is the Jacobian divided by the cube of the middle singular value of the Jacobian. In plane English, size of the element is factored out. The weight in the averaging is the volume of the contributing brick element or the area of the contributing shell element.

The **ELM** command can be used to highlight nodes that fall within a specified range, just as in the other options for the **MEASURE** command. The difference is that the node is identified by the three faces where that node is found within an element.

62. The **OFFSET** command syntax was not documented correctly in the help option from the menus in **TrueGrid**[®] and from the **TrueGrid**[®] manuals. So they are repeated here, but correctly.

OFFSET [*type offset*] ;
 where *type* can be
NODES *offset*
BRICKS *offset*

SHELLS *offset*
BEAMS *offset*
TSHELLS *offset*
NSETS *offset*
FSETS *offset*
ESETS *offset*
PARTS *offset*
VECIDS *offset*
LSYS *offset*

63. There was a minor bug in the expressions that are bracketed in the square brackets. Only in special cases when using the **RAND** or **NORM** function without arguments would this bug give an incorrect result. This is now fixed.

64. The set command with the thick option followed by an integer from 0-10 changed the line thickness in the postscript files produced by the postscript command. This option now also changes the line thickness for the wire and hide graphics (part and merge phase) of the mesh in both the physical and computational windows.

65. The **BBINT** command did not remove interior mesh lines from the picture for intra-part block boundary interfaces in the part phase. That has been corrected.

66. When the F4 function key is used to save the window environment to automatically recreate the same environment every time TG is run, it also saves the line thickness and the best choice of colors for the color deficient. These two parameters can be set using the **SET** command with the **thick** and **cbind** options, respectively.

67. There was a bug in the execution of the **PLAN** command specifying a symmetry plane. If the plane was not parallel to the x-y, y-z, or z-x planes, then the nodal constraints in the input files for **ABAQUS**, **ANSYS**, **DYNA3D**, **LSDYNA**, **NASTRAN**, and **NE/NASTRAN** would likely be incorrect. The problem was in the way the local frame of reference was constructed, which is different for each of the codes mentioned above. This bug has been fixed.

68. A minor bug was fixed in the **MEASURE** and **MEA/MEAI** command that caused a crash in rare cases.

69. The measure of aspect ratio for 20 node quadratic hexahedron elements was incorrect. This option is functioning properly now.

70. There are a number of new automatic parameters associated with spline curves. In order to be consistent in the use of these new automatic parameters with respect to Cubic Splines, B-Splines, and NURBS, the knots will refer to the monotonically increasing sequence of numbers in the 1D parameter space. Knot points will refer to the 3D coordinates that result when mapping the knots to the 3D curve. Control Points, which are not found in Cubic Splines, refer to the 3D coefficients that scale the basis functions. The 1D weights, which are only found in NURBS, scale the basis functions

to form the polynomial in the denominator of the rational function.

If the last 3D spline curve was formed using the `cps3` (Cubic Spline) option of the `curd` command, then the following parameters will be automatically set:

`splks` - 1 dimensional array containing all of the knots
`splnk` - scalar number of knots in the array `splks`
`splkps` - 2 dimensional array (3,*) of knot points on the 3D curve
`splord` - the order of the polynomials - always 4
`splcn` - **TrueGrid**[®] curve number that was created with this data
`spltyp` - type of spline curve - always 1

For example, if the following command were executed:

```
curd 11 csp3 00
-3.3954249e+00 1.2127594e-01 -3.5484806e-01
-3.3571637e+00 1.5977994e-01 -3.3923153e-01
-3.3138983e+00 2.0643232e-01 -3.1305539e-01
-3.2234428e+00 2.8636580e-01 -2.4210249e-01
-3.1053569e+00 3.5468611e-01 -1.2244769e-01
-3.0454067e+00 3.7062020e-01 -5.3555769e-02
-3.0135252e+00 3.7547471e-01 -1.6162919e-02
-3.0040656e+00 3.7634343e-01 -4.9400860e-03
-2.9906838e+00 3.7713331e-01 1.1124370e-02
-2.9789010e+00 3.9086247e-01 3.7436009e-02
-2.9678857e+00 4.1399955e-01 7.1353959e-02
-2.9503219e+00 4.4951856e-01 1.2442068e-01
-2.9295484e+00 4.8749987e-01 1.8401165e-01
-2.9095490e+00 5.1844508e-01 2.3694231e-01
-2.8932302e+00 5.3794400e-01 2.7557711e-01
-2.8758707e+00 5.5043960e-01 3.1012778e-01
-2.8553295e+00 5.4789609e-01 3.3721066e-01
-2.8470099e+00 5.3850897e-01 3.4150743e-01
-2.8389274e+00 5.2246647e-01 3.4014181e-01
-2.8261104e+00 4.7641905e-01 3.2144514e-01
-2.8157659e+00 4.1792813e-01 2.8925255e-01
-2.7706413e+00 2.9269938e-01 2.5559440e-01
-2.7520236e+00 2.6949270e-01 2.6488436e-01
-2.7330521e+00 2.5119662e-01 2.7857573e-01
-2.7025447e+00 2.2323597e-01 3.0129292e-01
-2.5951468e+00 1.1034914e-01 3.5839118e-01
-2.4999798e+00 0.0000000e+00 3.7500000e-01 ; ; ;
```

then the following automatic parameters would be set to have the values below:

```
%spltyp = 1
%splcn = 11
%splord = 4
```

%splnk = 27

%splks(*) = 0.000000D+00

3.260380D-02

7.231759D-02

1.531419D-01

2.578884D-01

3.113998D-01

3.399025D-01

3.483897D-01

3.604670D-01

3.788989D-01

4.034368D-01

4.416657D-01

4.841821D-01

5.214089D-01

5.481064D-01

5.715624D-01

5.912381D-01

5.988915D-01

6.092905D-01

6.389166D-01

6.779151D-01

7.571689D-01

7.751602D-01

7.923042D-01

8.195539D-01

9.153427D-01

1.000000D+00

%splkps(3,*) = -3.395425D+00 1.212759D-01 -3.548481D-01

-3.357164D+00 1.597799D-01 -3.392315D-01

-3.313898D+00 2.064323D-01 -3.130554D-01

-3.223443D+00 2.863658D-01 -2.421025D-01

-3.105357D+00 3.546861D-01 -1.224477D-01

-3.045407D+00 3.706202D-01 -5.355577D-02

-3.013525D+00 3.754747D-01 -1.616292D-02

-3.004066D+00 3.763434D-01 -4.940086D-03

-2.990684D+00 3.771333D-01 1.112437D-02

-2.978901D+00 3.908625D-01 3.743601D-02

-2.967886D+00 4.139995D-01 7.135396D-02

-2.950322D+00 4.495186D-01 1.244207D-01

-2.929548D+00 4.874999D-01 1.840116D-01

-2.909549D+00 5.184451D-01 2.369423D-01

-2.893230D+00 5.379440D-01 2.755771D-01

-2.875871D+00 5.504396D-01 3.101278D-01

-2.855329D+00 5.478961D-01 3.372107D-01

```

-2.847010D+00 5.385090D-01 3.415074D-01
-2.838927D+00 5.224665D-01 3.401418D-01
-2.826110D+00 4.764191D-01 3.214451D-01
-2.815766D+00 4.179281D-01 2.892525D-01
-2.770641D+00 2.926994D-01 2.555944D-01
-2.752024D+00 2.694927D-01 2.648844D-01
-2.733052D+00 2.511966D-01 2.785757D-01
-2.702545D+00 2.232360D-01 3.012929D-01
-2.595147D+00 1.103491D-01 3.583912D-01
-2.499980D+00 0.000000D+00 3.750000D-01

```

If the last 3D spline curve was formed using the bsp3 (B-Spline) option of the **CURD** command, then the following parameters will be automatically set:

```

splks - 1 dimensional array containing all of the knots
splnk - scalar number of knots in the array splks
splkps - 2 dimensional array (3,*) of knot points on the 3D curve
splcps - 2 dimensional array (3,*) of control points that form the curve
splord - the order of the polynomials
splcn - TrueGrid® curve number that was created with this data
spltyp - type of spline curve - always 2

```

For example, if the following command were executed:

```

curd 12 bsp3
c knots
0. 0. 0. 0. 0.0625 0.0625 0.125 0.125 0.25 0.25
0.375 .4875 .5 .5 0.625 0.625 0.75 0.75 0.875 0.875
1. 1. 1. 1.;
c control points
-3.395424947 0.121275936 -0.354848062
-3.374553416 0.140465722 -0.348289595
-3.355863722 0.161100044 -0.339381434
-3.319725144 0.200588894 -0.317643003
-3.302032345 0.219324773 -0.304963004
-3.246632986 0.2703958 -0.263503838
-3.208025634 0.2994159 -0.229995018
-3.133785624 0.344398736 -0.154716192
-3.096576553 0.360293935 -0.112937594
-3.02172923 0.376659885 -0.026342943
-2.985091236 0.377181147 0.017778912
-2.880315 6.026919D-01 3.450537D-01
-2.843035 5.878149D-01 3.868439D-01
-2.798396013 0.304679856 0.222954109
-2.760616573 0.276025848 0.257583401
-2.686456564 0.209309161 0.314211955
-2.648954363 0.170263866 0.336990097

```

```
-2.574723376 0.087828443 0.367203314
-2.537563881 0.044069575 0.375
-2.499979847 0.          0.375;;
```

then the following automatic parameters would be set to have the values below:

```
%spltyp = 2
%splcn = 12
%splord = 4
%splnk = 24
%splkns = 0.000000D+00
0.000000D+00
0.000000D+00
0.000000D+00
6.250000D-02
6.250000D-02
1.250000D-01
1.250000D-01
2.500000D-01
2.500000D-01
3.750000D-01
4.875000D-01
5.000000D-01
5.000000D-01
6.250000D-01
6.250000D-01
7.500000D-01
7.500000D-01
8.750000D-01
8.750000D-01
1.000000D+00
1.000000D+00
1.000000D+00
1.000000D+00
%splkps(3,*)= -3.395425D+00 1.212759D-01 -3.548481D-01
-3.337794D+00 1.808445D-01 -3.285122D-01
-3.337794D+00 1.808445D-01 -3.285122D-01
-3.283566D+00 2.363484D-01 -2.911433D-01
-3.283566D+00 2.363484D-01 -2.911433D-01
-3.170906D+00 3.219073D-01 -1.923556D-01
-3.170906D+00 3.219073D-01 -1.923556D-01
-3.085229D+00 3.610343D-01 -9.952367D-02
-2.992427D+00 3.770003D-01 8.962786D-03
-2.975566D+00 3.976821D-01 4.753117D-02
-2.975566D+00 3.976821D-01 4.753117D-02
-2.820715D+00 4.462474D-01 3.048990D-01
```

```

-2.820715D+00 4.462474D-01 3.048990D-01
-2.723537D+00 2.426675D-01 2.858977D-01
-2.723537D+00 2.426675D-01 2.858977D-01
-2.611839D+00 1.290462D-01 3.520967D-01
-2.611839D+00 1.290462D-01 3.520967D-01
-2.499980D+00 0.000000D+00 3.750000D-01
%splcps(3,*)= -3.395425D+00 1.212759D-01 -3.548481D-01
-3.374553D+00 1.404657D-01 -3.482896D-01
-3.355864D+00 1.611000D-01 -3.393814D-01
-3.319725D+00 2.005889D-01 -3.176430D-01
-3.302032D+00 2.193248D-01 -3.049630D-01
-3.246633D+00 2.703958D-01 -2.635038D-01
-3.208026D+00 2.994159D-01 -2.299950D-01
-3.133786D+00 3.443987D-01 -1.547162D-01
-3.096577D+00 3.602939D-01 -1.129376D-01
-3.021729D+00 3.766599D-01 -2.634294D-02
-2.985091D+00 3.771811D-01 1.777891D-02
-2.880315D+00 6.026919D-01 3.450537D-01
-2.843035D+00 5.878149D-01 3.868439D-01
-2.798396D+00 3.046799D-01 2.229541D-01
-2.760617D+00 2.760258D-01 2.575834D-01
-2.686457D+00 2.093092D-01 3.142120D-01
-2.648954D+00 1.702639D-01 3.369901D-01
-2.574723D+00 8.782844D-02 3.672033D-01
-2.537564D+00 4.406957D-02 3.750000D-01
-2.499980D+00 0.000000D+00 3.750000D-01

```

If the last 3D spline curve was formed using the `nrb3` (NURBS) option of the **CURD** command, then the following parameters will be automatically set:

```

splks - 1 dimensional array containing all of the knots
splnk - scalar number of knots in the array splks
splkps - 2 dimensional array (3,*) of knot points on the 3D curve
splcps - 2 dimensional array (3,*) of control points that form the curve
splwats - 1 dimensional array of weights
splord - the order of the polynomials
splcn - TrueGrid® curve number that was created with this data
spltyp - type of spline curve - always 3

```

For example, if the following command were executed:

```

curd 13 nrb3
c knots
0. 0. 0. 0. 0.0625 0.0625 0.125 0.125 0.25 0.25
0.375 .4875 .5 .5 0.625 0.625 0.75 0.75 0.875 0.875
1. 1. 1. 1.;

```

```

c weights
1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
1. 1. 1. 1. 1. 1. 1. 1. 1. 1. ;
c control points
-3.395424947 0.121275936 -0.354848062
-3.374553416 0.140465722 -0.348289595
-3.355863722 0.161100044 -0.339381434
-3.319725144 0.200588894 -0.317643003
-3.302032345 0.219324773 -0.304963004
-3.246632986 0.2703958 -0.263503838
-3.208025634 0.2994159 -0.229995018
-3.133785624 0.344398736 -0.154716192
-3.096576553 0.360293935 -0.112937594
-3.02172923 0.376659885 -0.026342943
-2.985091236 0.377181147 0.017778912
-2.880315 6.026919D-01 3.450537D-01
-2.843035 5.878149D-01 3.868439D-01
-2.798396013 0.304679856 0.222954109
-2.760616573 0.276025848 0.257583401
-2.686456564 0.209309161 0.314211955
-2.648954363 0.170263866 0.336990097
-2.574723376 0.087828443 0.367203314
-2.537563881 0.044069575 0.375
-2.499979847 0. 0.375;;

```

then the following automatic parameters would be set to have the values below:

```

%spltyp = 3
%splcn = 13
%splord = 4
%splnk = 24
%splkns = 0.000000D+00
0.000000D+00
0.000000D+00
0.000000D+00
6.250000D-02
6.250000D-02
1.250000D-01
1.250000D-01
2.500000D-01
2.500000D-01
3.750000D-01
4.875000D-01
5.000000D-01
5.000000D-01
6.250000D-01
6.250000D-01
7.500000D-01

```


7.500000D-01
8.750000D-01
8.750000D-01
1.000000D+00
1.000000D+00
1.000000D+00
1.000000D+00
%splkps(3,*)= -3.395425D+00 1.212759D-01 -3.548481D-01
-3.337794D+00 1.808445D-01 -3.285122D-01
-3.337794D+00 1.808445D-01 -3.285122D-01
-3.283566D+00 2.363484D-01 -2.911433D-01
-3.283566D+00 2.363484D-01 -2.911433D-01
-3.170906D+00 3.219073D-01 -1.923556D-01
-3.170906D+00 3.219073D-01 -1.923556D-01
-3.085229D+00 3.610343D-01 -9.952367D-02
-2.992427D+00 3.770003D-01 8.962786D-03
-2.975566D+00 3.976821D-01 4.753117D-02
-2.975566D+00 3.976821D-01 4.753117D-02
-2.820715D+00 4.462474D-01 3.048990D-01
-2.820715D+00 4.462474D-01 3.048990D-01
-2.723537D+00 2.426675D-01 2.858977D-01
-2.723537D+00 2.426675D-01 2.858977D-01
-2.611839D+00 1.290462D-01 3.520967D-01
-2.611839D+00 1.290462D-01 3.520967D-01
-2.499980D+00 0.000000D+00 3.750000D-01
%splcps(3,*)= -3.395425D+00 1.212759D-01 -3.548481D-01
-3.374553D+00 1.404657D-01 -3.482896D-01
-3.355864D+00 1.611000D-01 -3.393814D-01
-3.319725D+00 2.005889D-01 -3.176430D-01
-3.302032D+00 2.193248D-01 -3.049630D-01
-3.246633D+00 2.703958D-01 -2.635038D-01
-3.208026D+00 2.994159D-01 -2.299950D-01
-3.133786D+00 3.443987D-01 -1.547162D-01
-3.096577D+00 3.602939D-01 -1.129376D-01
-3.021729D+00 3.766599D-01 -2.634294D-02
-2.985091D+00 3.771811D-01 1.777891D-02
-2.880315D+00 6.026919D-01 3.450537D-01
-2.843035D+00 5.878149D-01 3.868439D-01
-2.798396D+00 3.046799D-01 2.229541D-01
-2.760617D+00 2.760258D-01 2.575834D-01
-2.686457D+00 2.093092D-01 3.142120D-01
-2.648954D+00 1.702639D-01 3.369901D-01
-2.574723D+00 8.782844D-02 3.672033D-01
-2.537564D+00 4.406957D-02 3.750000D-01
-2.499980D+00 0.000000D+00 3.750000D-01

stored in automatic parameters. The automatic parameter %N2dINT will be set to the number of intersections. The automatic array parameter %X2DCI will be set to the x-coordinate for every point of intersection. The automatic array parameter %Y2DCI will be set to the y-coordinate for every point of intersection.

The automatic array *RARCL1 will contain the relative position of the corresponding intersection point along the first curve. The automatic array *RARCL2 will contain the relative position of the corresponding intersection point along the second curve.

Alternatively, the new command

IT2DC *2Dcurve_1 2Dcurve_2*

will also create the list of intersections without the side effects of the LD command.

The order of the intersections in these arrays may be important, so the algorithm is explained. The first 2D curve in the argument list is search from start to finish for intersections with the second curve. This order is preserved in these new automatic array parameters.

78. The **SFB** command has a new option so that it can be applied to an edge or a single vertex of the mesh. When a face of the mesh is constrained by this command, The z-axis of the local coordinate system is taken from the normal of the mesh or surface (type). But when an edge or a vertex is selected as the region or index progression, there is no indication of which face or surface to use to get the normal. In this case, choose i, j, or k for the direction of the normal. For example, If the i-face is chosen, the neighboring nodes or the corresponding surface of projection, depending on the type, along the i-face passing through the edge or vertex will be used to form the normal. The flow face option must be used for edges and vertices and is ignored for faces of the mesh. The flow face argument is optional.

SFB *i1 j1 k1 i2 j2 k2 type flow_direction [flow_face] constraint_list ;*

or

SFBI *i_list; j_list; k_list; type flow_direction [flow_face] constraint_list ;*

where *type* can be

MESH

SURFACE

where *flow_direction* can be

NONE *option* if the constraints requires no flow direction

where an *option* must be one of

T for move only in the normal direction

N for move only in the tangent plane

I if the flow direction corresponds to the i-direction

J if the flow direction corresponds to the j-direction

K if the flow direction corresponds to the k-direction

COOR *x y z* to supply the flow vector

where the *flow_face* for edges/vertices is

I	for the neighboring i-face
J	for the neighboring j-face
K	for the neighboring k-face
where a local constraint can be any combination of	
DX value	for x-displacement
DY value	for y-displacement
DZ value	for z-displacement
RX value	for rotation about the x-axis
RY value	for rotation about the y-axis
RZ value	for rotation about the z-axis
	followed by a <i>value</i> of
0	for initialize to no constraint
1	for constraint

79. The **PCOMP** option for shell elements in the **NASTRAN** output incorrectly generated CSHEAR elements. That has been corrected. Also, additional options are now available. A bug was fixed in writing the NASTRAN file for boundary conditions in a local coordinate systems using the **LSYS** and **LB** commands. The **SFB** command now generates boundary conditions within a local coordinate system for NASTRAN. The **PR** command now creates the PLOAD4 cards for NASTRAN (instead of PLOAD).

A number of loads and boundary conditions have a new option to assign a set id. These are **MPC**, **LB/LBI**, **PR/PRI**, and **TE/TEI**. The creation of a NASTRAN material model has been improved, which includes the selection of the element type and associated properties as part of the material definition. A new "**TrueGrid**[®] to NASTRAN User's Manual" is available. It defines the **NASTMATS** command, as well as numerous other commands that have features uniquely designed to support NASTRAN.

80. A bug was fixed in the TRBB command. The problem occurred when the following conditions existed:

- 1.) A 2-way block boundary transition was formed.
- 2.) The slave side had the coarser mesh.
- 3.) Both transition ratios were 1:3.
- 4.) The slave side of the block boundary transition was (nearly) the size of the largest region in the part.

In addition, the transition algorithm was improved so that no merging is needed within the transition block. This was needed in some cases when the slave side of the transition was divided into several regions.

81. The new merge phase command **MPCE** (Multiple Point Constraint Equation) has been added. The equation has the following explicit form:

$$ND(0) = COEF(0) + \text{SUM} [ND(i) * COEF(i)] \text{ for } i \text{ from } 1 \text{ to } n$$

where

n is the number of independent nodes

COEF are coefficients

ND specifies the DOF of a node

where ND(0) is the DOF of the dependent nodes

and the rest are the DOF of independent nodes

In **ANSYS** it creates the CE cards , in **ABAQUS** it creates the *EQUATION card, in **NASTRAN** it creates the MPC cards, in the **NEUTRAL** file it creates card 14, and in **LSDYNA** it creates the *CONSTRAINT_INTERPOLATION cards. The syntax for this new command in **TrueGrid**[®] is:

MPCE *id depend_node dof option independ_nodes ;*

where *depend_node* can be

N *node_#*

PM *point_mass_#*

RT *x_coordinate y_coordinate z_coordinate constraints ;*

CY *radial_coordinate angular_coordinate z_coordinate constraints ;*

SP *radial_coordinate polar_angle azimuthal_angle constraints ;*

where *option* can be

C0 *constant_term*

for the constant term

where *independ_nodes* must be one of

NSETS [*node_set dof*]

for independent nodes in sets

NLISTS [*node_list ; dof weight*]

for independent nodes in lists

where *dof* can be any one of

DX

for x-displacement dependent

DY

for y-displacement dependent

DZ

for z-displacement dependent

RX

for x-rotation dependent

RY

for y-rotation dependent

RZ

for z-rotation dependent

When using the sets option, no weights can be specified. They are assumed to be 1. When using the lists option, one can be as specific as a unique DOF and weight for each node by choosing just 1 node for each list.

To check the **MPCEs**, use the **CO** command with the **MPCE** option to display the relation between the dependent node and the independent nodes. They are shown by a line connecting the dependent node and each of the independent nodes. The dependent node has a special mark. This works for wire, hide, fill, and Fast Graphics.

Related to this is the way in which the **MPC** and **JT** (Joint) commands were translated to the **MPCE** cards for the **NASTRAN** output file. This has been corrected. The option to generate a spotweld in the **JD/JT** command is deprecated, since there are much better ways in **TrueGrid**[®] to generate a spotweld. Also, a bug was fixed regarding the **MPC** generation in the **JD/JT** and **MPC** commands.

In all three commands (**MPCE**, **MPC**, and **JT**) that generate multiple point constraints, if any two nodes are in a **MPC**, they cannot be merged. All merge commands (**T**, **TP**, **ST**, and **STP**) will avoid merging such pairs.

82. A projection bug was fixed. This bug only applies to the surface formed by rotating an infinite parabolic curve.

83. A minor bug was fixed regarding the rotation of the mesh using the mouse. Rotation is selected in the **Environment window** in the **Move panel** while in the part phase. This failed only when the physical coordinates were very large (i.e. greater than about 1.0e+06).

84. A bug was fixed in the fast graphics for large models in the part phase.

85. There is a new output option to create the FEA, GEO, and DAT files for the FEA code called MPACT by Pedro Marcal. Issue the MPACT command at the start of using TG. This version can generate 8 node bricks and 4 node quad shells (use the **LINEAR** command at the start of TG) (default) or 27 node brick and 9 node quad shells (use the **TRIQUADRATIC** command at the start of TG). Nodes can be constrained using the **B/BI** commands in the part phase and the **B** command in the merge phase. Use the set id option to form separate conditions. Use the **PR** and **PRI** commands in the part phase and the **PR** command in the merge phase to assign pressure. Three files (FEA, GEO, and DAT) files will be written after issuing the write command in the merge phase. Be sure to merge the nodes first.

86. The **NASTRAN** quadratic beam property cards is now supported within the **BSD** command. Use the new option **bnA9** to define the properties as seen below:

BNA9 <i>t_options</i> ;	Tapered beam cross section
where <i>t_options</i> can be	
KY <i>factor</i>	Shear effectiveness y-directions
KZ <i>factor</i>	Shear effectiveness z-directions
STRESS <i>location</i>	Location for stress, strain, force
where <i>location</i> can be GRID or GAUSS	
SOB <i>option</i>	stress output request at node B
SOC <i>option</i>	stress output request at node C
where <i>option</i> can be YES or YESA	
CSDA <i>parameters</i> ;	cross section data at node A
CSDB <i>parameters</i> ;	cross section data at node B
CSDC <i>parameters</i> ;	cross section data at node C
where <i>parameters</i> can be any of	
A <i>area</i>	area of beam cross section
IZ <i>moment</i>	area moment of inertia about z-axis
IY <i>moment</i>	area moment of inertia about y-axis
IYZ <i>inertia</i>	area product of inertia
J <i>stiffness</i>	torsional stiffness
NSM <i>mass</i>	nonstructural mass

CY <i>y</i>	Stress output y-coordinate at point 1
CZ <i>z</i>	Stress output z-coordinate at point 1
DY <i>y</i>	Stress output y-coordinate at point 2
DZ <i>z</i>	Stress output z-coordinate at point 2
EY <i>y</i>	Stress output y-coordinate at point 3
EZ <i>z</i>	Stress output z-coordinate at point 3
FY <i>y</i>	Stress output y-coordinate at point 4
FZ <i>z</i>	Stress output z-coordinate at point 4
NY <i>y</i>	Neutral axis y-coordinate
NZ <i>z</i>	Neutral axis z-coordinate
MY <i>y</i>	Center of gravity y-coordinate
MZ <i>z</i>	Center of gravity z-coordinate
NSIY <i>moment</i>	nonstructural mass moments of inertia y-axis
NSIZ <i>moment</i>	nonstructural mass moments of inertia z-axis
NSIYZ <i>inertia</i>	nonstructural mass product of inertia
CW <i>coef</i>	warping coefficient
W1 <i>value</i>	value of warping function at point 1
W2 <i>value</i>	value of warping function at point 2
W3 <i>value</i>	value of warping function at point 3
W4 <i>value</i>	value of warping function at point 4
WY1 <i>dy</i>	y-gradient of warping function at point 1
WY2 <i>dy</i>	y-gradient of warping function at point 2
WY3 <i>dy</i>	y-gradient of warping function at point 3
WY4 <i>dy</i>	y-gradient of warping function at point 4
WZ1 <i>dz</i>	z-gradient of warping function at point 1
WZ2 <i>dz</i>	z-gradient of warping function at point 2
WZ3 <i>dz</i>	z-gradient of warping function at point 3
WZ4 <i>dz</i>	z-gradient of warping function at point 4

87. A bug was fixed where a space in the path or file name would result in an error. Double quotes are used in all cases when a path and file are specified and will show up in the tsave file.

88. Discrete Element Spheres can be generated using the new **DEMICLE** command prior to creating a part. This will cause all brick elements of the next part to be converted to Discrete Element Spheres. This is for **LSDYNA** output.

Each sphere will have a center coincident with the solid element. The radius will be the largest radius of a sphere that fits within the solid element.

The new **DEM/DEMI** commands in the part phase will assign DEM properties to the spheres. The syntax for the **DEM** and **DEMI** commands are:

DEM *i1 j1 k1 i2 j2 k2 inertia flag*
DEMI *i-list; j-list; k-list; inertia flag*
 where *flag* can be

0
1

to use mass in the definition
to use volume in the definition

The mass will be calculated from the volume and density found in the material definition (see **MATE**, **MT**, **MTI**).

Use the **CONDITION (CO)** command to view the Discrete Element numbers (**DEM**). The default length of the lines that cross to represent a discrete Element will be the radius of that element. The size can be scaled using the **CO SIZE** option.

89. The order of the brick element must be chosen as **LINEAR**, **QUADRATIC**, or **TRIQUADRATIC** before the first part is generated. **LINEAR** is the default.

90. A bug was fixed when 2nd order elements are generated (**QUADRATIC** or **TRIQUADRATIC** commands). If an edge of the mesh is attached to a 3D curve, if the same edge or a region containing the edge is projected to one surface, and if the 3D curve is not on the surface of projection, then it was possible that the 2nd order nodes (even nodes along and edge) would be interpolated between its two neighboring 1st order nodes and projected instead of being projected from its original position on the 3D curve.

91. There was a bug when displaying elements using the **ELM** command after using the **MEA/MEAI** command. This occurred only in the part phase with a part in cylindrical coordinates (**CYLINDER** command) using **Fast Graphics**. This has been fixed.

92. On WINDOWS, **TrueGrid**[®] licensing has changed and new license codes will be needed to run the new version of the license manager. Other changes to the license utilities includes version information. The information for GUI driven applications (e.g. "Registration", "**TrueGrid**[®] Control") are displayed in the main window. Console applications (i.e. run in a "Command Prompt" window) is available using the "-v" flag with the application (e.g. tgq.exe -v). Note, the "-v" flag also works with **TrueGrid**[®] (eg. tg.exe -v).

93. **TrueGrid**[®] now adds run information in the save file (i.e. tsave). This information includes **TrueGrid**[®] version information, login user name, machine name, time and date of run, and input file name if one is selected.