SAND2026-15741

SANDIA REPORT

Printed January 8, 2026

Sandia National Laboratories

Sierra/SD – User's Guide for NasGen – 5.28

Sierra Structural Dynamics Development Team

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185
Livermore, California 94550

Issued by Sandia National Laboratories, operated for the United States Department of Energy by National Technology & Engineering Solutions of Sandia, LLC.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone:	(865) 576-8401
Facsimile:	(865) 576-5728
E-Mail:	reports@osti.gov
Online ordering:	http://www.osti.gov/scitech

Available to the public from

U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Road
Alexandria, VA 22312

Telephone:	(800) 553-6847
Facsimile:	(703) 605-6900
E-Mail:	orders@ntis.gov
Online order:	https://classic.ntis.gov/help/order-methods

ABSTRACT

NasGen provides a path for migration of structural models from Nastran bulk data format (BDF) into both an Exodus mesh file and an ASCII input file for Sierra Structural Dynamics (Salinas) and Solid Mechanics (Adagio). Many tools at Sandia National Labs (SNL) use the Exodus format. This document describes capabilities and limitations of the NasGen translation software.

This page left blank

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

The **Sierra/SD** software package is the collective effort of many individuals and teams. A core Sandia National Laboratories based **Sierra/SD** development team is responsible for maintenance of documentation, testing, and support of code capabilities. This team includes Dagny Beale, Gregory Bunting, David Day, Clark Dohrmann, Payton Lindsay, Justin Pepe, Julia Plews, Jesse Thomas, and Ben Treweek.

The **Sierra/SD** team also works closely with the Sierra Inverse and Plato teams to jointly enhance and maintain several capabilities. This includes contributions from Wilkins Aquino, Mark Chen, Sean Hardesty, Elizabeth Livingston, Clay Sanders, Chandler Smith, Adam Sokolow, Timothy Walsh, and Ray Wildman.

The **Sierra/SD** team works closely with other Sierra teams on core libraries and shared tools. This includes the DevOps, Sierra Toolkit, Solid Mechanics, Fluid Thermal Teams. Additionally, analysts regularly provide code capabilities as well as help review and verify code capabilities, testing, and documentation. Other individuals not already mentioned directly contributing to the **Sierra/SD** documentation, testing, and code base during the last year include Victor Brunini, Jonathan Clausen, Nathan Crane, Jared Crean, David Glaze, Mark Hamilton, Jacob Healy, Andrew Kimler, Dong Lee, Kevin Manktelow, Scott Miller, Matthew Mosby, Tony Nguyen, Tolu Okusanya, Heather Pacella, Malachi Phillips, Kendall Pierson, Nick Reynolds, Philip Sakievich, Timothy Shelton, John Shimanek, Greg Sjaardema, Clinton Stimpson, Tyler Voskuilen, Ellen Wagman, Alan Williams, and Christopher Wilson.

Historically dozens of other Sandia staff, students, and external collaborators have also contributed to the **Sierra/SD** product and its documentation.

Many other individuals groups have contributed either directly or indirectly to the success of the **Sierra/SD** product. These include but are not limited to;

- Garth Reese implemented the original **Sierra/SD** code base. He served as principal investigator and product owner for **Sierra/SD** for over twenty years. His efforts and contributions led to much of the current success of **Sierra/SD**.
- The ASC program at the DOE which funded the initial **Sierra/SD** (Salinas) development as well as the ASC program which still provides the bulk of ongoing development support.
- Line managers at Sandia Labs who supported this effort. Special recognition is extended to David Martinez who helped establish the effort.
- Charbel Farhat and the University of Colorado at Boulder. They have provided incredible support in the area of finite elements, and especially in development of linear solvers.

- Carlos Felippa of U. Colorado at Boulder. His consultation has been invaluable, and includes the summer of 2001 where he visited at Sandia and developed the Hexshell element for us.
- Danny Sorensen, Rich Lehoucq and other developers of ARPACK, which is used for eigenvalue problems.
- Esmond Ng who wrote *sparspak* for us. This sparse solver package is responsible for much of the performance in **Sierra/SD** linear solvers.
- The *metis* team at the University of Minnesota. *Metis* is an important part of the graph partitioning schemes used by several of our linear solvers. These are copyright 1997 from the University of Minnesota.
- Padma Raghaven for development of a parallel direct solver that is a part of the linear solvers.
- The developers of the SuperLU Dist parallel sparse direct linear solver. It is used through GDSW for a variety of problems.
- Leszek Demkowicz at the University of Texas at Austin who provided the HP3D[2] library and has worked with the **Sierra/SD** team on several initiatives. The HP3D library is used to calculate shape functions for higher order elements.

This work was supported by the Laboratory Directed Research and Development (LDRD) program.

1. INTRODUCTION

NasGen provides a path for migration of structural models from NASTRAN bulk data format (BDF) into both an Exodus mesh file and an ASCII input file for Sierra Structural Dynamics (Salinas) and Solid Mechanics (Presto). Many tools at Sandia National Labs use the Exodus format. NasGen was written specifically for Salinas and Presto but should be usable with a number of these packages. *However, if the translation of a NASTRAN card type to Exodus, Salinas, or Presto files is not discussed in this document then that translation is either non-existent or not verified for use.*

1.1. NasGen Usage

`Nasgen` is a preprocessor tool for generation of the geometry files distributed with Sierra Mechanics Applications. `Nasgen` translates most of the `NASTRAN` bulk data file constructs to a geometry file format, Exodus, that works with the Sierra Mechanics Applications.

nasgen [options] *example.bdf example.exo*

The *example.bdf* file is a NASTRAN bulk data deck, and the *example.exo* file is the resulting Exodus file.

1.1.1. Options

-S Outputs a Salinas input file. The file is a text description of the contents of the Exodus file which can be used as a starting point to run Salinas (structural dynamics). The file name of the Salinas input is identical to the Exodus file name, with the "`.inp`" extension.

-P Outputs a Presto input file. This text file can be used as a starting point to run Presto (solid mechanics). The file name of the Presto input is identical to the Exodus file name, with the "`.i`" extension. The output Exodus file is sensitive to this option if rigid elements are specified in NASTRAN. See Sections 2.2 and 2.9.2 for more details.

-N Outputs a Presto input file (with auxiliary input files) that can be used as a starting point to run a coupled Presto-NEMO analysis.

-i input_nastran Specify the file to read input from.

-o output_exodus Specify the Exodus file to receive the data.

-i4 Instructs the program to write Exodus files in a format so that floating-point data is written in 4 bytes. This may result in a loss of accuracy. (Note: do NOT put a space between the characters, i.e. '-i 4' will generate an error).

-u Generates a log file of the cards NOT referenced in the translation process. The file name of the log file is identical to the Exodus file name, with the "`.log`" extension.

-e Echo all cards in the input bulk data deck to the log file. Note that this turns off the "-u" option. Cards are parsed and written to the log file. This is useful primarily for debugging.

-n Establishes Navy specific translation options. For Salinas and if "-P" is not specified, this includes defaulting to the Nbeam and some specialization for mount elements.

-rbe2blk Creates a new exodus element block for each RBE2 element in the NASTRAN input. This option is not compatible with Presto output "-P". Without this option, the RBE2 elements are put in the same block as RBARs.

-q "Quiet" running. Only warnings are output. No informational messages are printed.

-d "Debug" running. All debugging output is provided. When -d is specified, -q is ignored.

> Note: while the options may occur in any order, the options must all occur *before* the input NASTRAN or output Exodus file is specified, i.e. they must be specified last on the command line unless the -i and -o flags are used.

2. NASGEN TRANSLATION

2.1. Node and Element Numbering

The Exodus format uses an implied node numbering of 1:N, where N is the number of nodes (or GRID points) in the model. Exodus also supports a node number map. NasGen converts existing node numbering into the implicit node numbering of Exodus and provides the appropriate map. However, most tools for manipulation of Exodus files ignore the map. For example, `blot`, `p3/exodus`, and `ensight7` all ignore the node number map in output of the results data. In addition, it is common to use SEACAS tools such as `gjoin` to combine and manipulate multiple Exodus files. These processes destroy the node map.

It is therefore often advantageous (but not required) to renumber the GRID data before translation. This provides the clearest path from post processed node numbers to the original geometry. Additional methods to keep track of specific nodes in the FE model are discussed in Section 2.3.

The Exodus format also implicitly numbers elements. An element number map is provided back to the original element numbering from the Exodus file. Since element numbering in Exodus files is implicit by block, it is next to impossible to force a preprocessor to renumber consistent with Exodus numbering. However, it is advantageous to keep the element number range relatively small.

The element number map is used within NasGen to properly assign sideset type loads to Exodus elements. Memory for this reverse map is allocated based on the maximum difference of element numbers, i.e. an array of length (max_id-min_id) is allocated to do this reverse map. If there are no sidesets in the model, the reverse map will not be constructed.

2.2. Block Numbering

Blocks in an Exodus file correspond loosely to NASTRAN properties. However, Exodus element blocks require that all elements in a block have the same element topology. Thus, a collection of `CTRIA3` and `CQUAD4` elements could share a single NASTRAN property card but would require at least two Exodus blocks.

Block numbering in NasGen is based on the original NASTRAN property PID. The PID is multiplied by 10, and an integer added to ensure uniqueness. The integers are shown for solid and shell elements in the Table 2-1. Note that we do NOT translate elements with other numbers of nodes, e.g. there is no translation of a hex element with 15 nodes.

Table 2-1. – Block Numbering.

Element	# nodes	Increment	Comment
CHEXA	8	0	
CHEXA	20	1	
CPENTA	6	2	
CPENTA	15	3	
CTETRA	4	4	
CTETRA	10	5	
CTRIA3	3	0	
CTRIA6	6	1	
CQUAD4	4	2	
CQUAD8	8	3	
CBAR	2	0	
CBEAM	2	0	Equiv to bar
CDAMP	2	0	attributes not translated
CBUSH	2	0	
CGAP	2	0	attributes not translated
CELAS2	2	0	
CROD	2	0	
CONM2	1	7	see comments
Rigid Links	2+		see comments

Concentrated masses are treated differently from all other elements. All `CONM2` are in block 17, and the mass properties are written as attributes to the Exodus file.

Rigid links have limited support. The RBE1 element is recognized, but is not translated. All RROD, RTRPLT and RBAR elements are put into the same element block. Within the block, each link of the element is defined as a two node Exodus "BEAM-R" element. Block IDs for rigid elements start immediately above all other block IDs and are incremented by one. Each element within the block is assigned an attribute identifying its source element. In addition, two attributes identify the active degrees of freedom in the element. See section 2.9.2.1).

In Salinas, RBE3 elements are assigned a unique block ID, one for each RBE3. If the -rbe2blk option is specified, each RBE2 may be assigned to a unique block (as for RBE3). By default, the constraints for the RBE2 are added to the block with RBAR elements.

For Presto all RBAR, RROD, RTRPLT and fully constrained RBE2 elements are translated together. In Presto partially constrained RBE2s are converted into MPCs similar to RBE3s. In NasGen these NASTRAN elements are sorted to determine which are connected into rigid bodies but are always written to a single element block. The individual rigid bodies are indicated by an attribute written on the rigid Exodus elements, which Presto can read and used to separate rigid bodies.

2.2.1. Block Names

NASTRAN contains only integers to describe the different material blocks. To facilitate block identification as names in the Exodus format, the nasgen translator can sometimes read comments associated with the NASTRAN property card. This capability is enabled with the navy option (-n on the command line), and has been exercised only for FEMAP output.

2.3. Nodeset Numbering

Nodesets are a critical part of Exodus files. Each nodeset consists of a group of nodes with an associated distribution factor. There is exactly one distribution factor per node in a set. Nodesets are used to apply boundary conditions and to apply loads. Distribution factors are used to provide spatial variation. There are two issues that influence nodeset numbering:

- Because there is only one distribution factor per node, only a scalar can be represented by a single nodeset. Thus, if you wish to apply a force with spatially varying X, Y and Z components, three nodesets are required – one for each component.
- Nodeset numbering must be unique. In NASTRAN, load and BC cards have separate identifiers. Thus, having "FORCE,10,..." and "SPC,10,..." is valid in NASTRAN. To build a consistent Exodus file however, these two cards must be translated into two unique nodesets. In the event that two load or BC cards of the same type share a node the second instance on that node will be deleted and a warning generated.

translating nodesets is done as follows.

1. First the displacement nodesets are processed from `SPC` type cards. The nodeset ID is constructed from the SID by multiplying by 10 and adding component numbers. Thus,

   ```
   SPC1,30,12,...
   ```

   translates into nodeset ids 301 and 302.
2. Force nodesets are next translated. Each force will generate 3 nodesets. The IDs are computed similarly, but an offset is added so there is no conflict with any displacement nodesets.
3. Moment nodesets are processed exactly like force nodesets. An additional offset is applied.
4. Temperature nodesets are processed exactly like force nodesets. An additional offset is applied.
5. DAREA nodesets are processed exactly like force nodesets. An additional offset is applied.

Information about the nodeset numbering is output during the translation processes. In addition, if Salinas output is requested, further information can be found in the "`.inp`" file generated.

In addition, translation for Presto may not support some capabilities discussed above. It is advised that the user check the resulting Presto input file for the correct boundary conditions.

2.4. Sideset Numbering

Sidesets are collections of element faces that can be used for application of pressure loads. Many codes also use them for application of boundary conditions.

Sidesets are translated from `PLOAD1, PLOAD2,` and `PLOAD4` cards. Only the normal component of pressure is currently translated, i.e. fields greater than 9 on these cards are ignored.

Because Exodus applications use sidesets more extensively than NASTRAN, it may be necessary to "trick" the translator into generating appropriate sidesets. This is done by generating an unreferenced pressure (through a `PLOAD` card). If the pressure is not referenced in the Bulk Data deck, NASTRAN will ignore it. However, a sideset will be generated within Exodus.

The sideset ID is taken directly from the `SID` column of the `PLOAD*` card (column 2).

As with the nodeset capabilities, Presto translation should be checked for accuracy.

2.5. Parameters

Currently, the only `PARAM` card value used in Presto is `wtmass`. All material densities are multiplied by the value of `wtmass` before being written to the Presto input file. That is, the scaled values of densities are the values that will appear in the material parameters section of the Presto input file. If no `PARAM` card with `wtmass` is present in the NASTRAN Bulk Data deck, a default value of 1.0 is used.

2.6. Functions

NasGen has the ability to translate NASTRAN TABLED1 tables to sierra functions for use in Presto, but only for the LINEAR case. At this time, TABLED2 translation is not supported.

2.7. Ground

Some NASTRAN constructs allow attachments to GRID 0, which represents ground. If GRID 0 is referenced in the Bulk Data deck NasGen will generate an additional node at the origin and generate nodeset=1 for that node. This nodeset will need to be fixed in the Salinas and Presto input files.

2.8. Case Control

NasGen has limited case control capabilities. Loads and boundary conditions that are present in the NASTRAN database will be translated and used in both the Exodus and Salinas/.inp files. Case control that is recognized includes the following:

solution type (e.g. SOL), SPC, MPC, FREQUENCY, METHOD

This capability is currently not supported for translation to Presto input files.

2.9. Element Translation Table

There are two components in the translation of an element (and associated properties) from NASTRAN format to Salinas or Presto. Table 2-2 details the translation of element level data into the Exodus database. This database includes the topology, connectivity and element attributes. In addition, the properties and "behavior" of the element must be represented in the finite element code. Table 2-3 and Table 2-4 contain the element support for Salinas and Presto, respectively. Typically, the Exodus database can be translated with few difficulties. Differences in formulation of the finite elements can cause a significant issue for each element, and elements may lack support due to either NasGen or the finite element code. If an element is not listed in Tables 2-3 or 2-4 it is not supported by Salinas or Presto, respectively. Rigid element support is not reflected in these tables and is discussed in Section 2.9.2.

Notes from Exodus Translation Table

1. Bar and Beam Attributes are, "area", "I1", "I2", "J", "orientation(3)", "offset(3)". All bar and beam attributes are associated with the entire element, not with the individual ends. No "pin flags" are translated.
2. Dashpot coordinate components are rotated. No support for coupled directions.
3. No support for different component numbers, i.e. C1=C2.
4. Concentrated mass attributes: "mass", "I11", "I22", "I33", "I21", "I31", "I32", "offset(3)"
5. Coordinates are expanded in Salinas so that each coordinate references only the basic coordinate frame. Currently, this capability is not supported for Presto.
6. Only 8 or 20 node hex elements are supported.
7. Shell attributes include "thickness", "theta" and "offset".
8. No support of MCID, TFLAG or thickness variation within the element (an average is used).
9. Tets are supported for 4 and 10 node versions only.
10. Full 6×6 stiffness and damping matrices. No support for structural damping or offsets.
11. Limited support.

Table 2-2. – NASTRAN/Exodus Translation.

NASTRAN	Exodus	Attributes	Notes
CBAR	BEAM	10	1
CBEAM	BEAM	10	1
CDAMP	BEAM	3	2
CELAS1	TRUSS-SPR	0	3
CGAP	TRUSS-GAP	0	11
CHEXA	HEX8 OR HEX20	0	
CONM2	SPHERE-MASS	10	4
CORD	CORD	NA	5
CPENTA	WEDGE6 or 15	0	6
CROD	TRUSS	"Area"	
CQUAD4	SHELL4	3	7, 8
CQUAD8	SHELL8	3	7, 8
CTETRA	TETRA	0	9
CTRIA3	TRIANGLE	3	7, 8
CTRIA6	TRIANGLE	3	7, 8
CBUSH	TRUSS-BUS	25	10
CVISC	TRUS-VIS	0	11

Table 2-3. – NASTRAN/Salinas Element Support.

NASTRAN	Salinas	Issues and Notes	Status
CBAR	Beam2	b,c	very good
CBEAM	Beam2	d	good
CDAMP2	Dashpot	e, g	poor
CELAS	Spring	f, g	fair
CGAP	Gap	h, g	poor
CHEXA	Hex8 or 20		excellent
CONM2	Conmass	i	excellent
CORD	Coordinate	j	very good
CPENTA	Wedge 6 or 15		very good
CROD	Truss	g	very good
CQUAD4	QuadT	k, l	very good
CQUAD8	Quad8T	k, l	good
CTETRA	Tet4 or 10		excellent
CTRIA3	Tria3 or TriaShell	k, l	very good
CTRIA6	Tria6	k, l	very good
CBUSH	SpringDashpot		good
CVISC	—		none

Table 2-4. – NASTRAN/Presto Element Support.

NASTRAN	Presto	Issues and Notes	Status
CBAR	Beam	iii,iv	good
CBEAM	Beam	iii,iv,vi	good
CHEXA	Hex 8		good
CONM2	Point Mass	v	very good
CPENTA	Wedge 6		poor
CQUAD4	Quad 4	vii	good
CTETRA	Tet 4		fair
CTRIA3	Tri 3	viii	fair

Notes from Salinas Element Support

a. Only the topologies listed are supported. For example, there is no 19 node Hex.

b. There is no variation allowed through the length of the bar.

c. Torsion of bar elements may be an issue. Some versions of NASTRAN introduce no torsional mass, which allows some solvers to permit torsional degrees of freedom (dofs) to be unattached to the model. Salinas will treat these as zero energy modes.

d. Beam elements in Salinas are very nearly equivalent to NASTRAN CBAR elements. NASTRAN CBEAM elements are translated to the nearest approximation of these CBAR elements. They have no variation down the length of the bar.

e. The NASTRAN CDASH element is roughly equivalent to Salinas Dashpots. The scalar damper value, "B", is stored in the element attributes, but is correct only for an output coordinate system=0. Hand corrections may be needed to the Salinas input.

f. NASTRAN elements allow almost arbitrary flexibility in the degrees of freedom of CELAS type elements. In particular, stiffness in the X direction can couple to stiffness in the Y direction on the paired node. This flexibility is not fully supported. The Salinas spring elements couple X_1 with X_2.

g. One dimensional elements are problematic in parallel applications where it may be impossible to subdivide the model without generating dangling segments.

h. Gaps are not the same in Salinas and NASTRAN. The Salinas model supports the older NASTRAN gap specification only. Hand tuning is certainly required.

i. NasGen puts all concentrated masses into block 17 and assigns them a ConmassA element type. All properties are read from the Exodus file attributes.

j. Coordinates are expanded in Salinas so that each coordinate references only the basic coordinate frame.

k. The NASTRAN and Salinas shell formulations are radically different. Salinas uses a 6 dof/node formulation similar to the DKT membrane for the triangle. Many elements are composites, built of triangles.

l. TriaShell elements support multiple, orthotropic layers, while the Tria3 does not. If orthotropic layers are identified, then the TriaShell is selected. Likewise, those elements built on triangles (QuadT, Quad8T) select the TriaShell as underlying elements as needed.

Notes from Presto Element Support

i. Only the topologies listed are supported. For example, there is no 19 node Hex.

ii. In general Presto elements differ from NASTRAN elements in that Presto elements are nonlinear, which may cause differences in behavior.

iii. There is no variation allowed through the length of the bar.

iv. The moments of inertia and mass in Presto are computed directly from the cross section.

v. NasGen puts all concentrated masses into block 17 and translates them as point masses. All properties are read from the Exodus file attributes.

vi. Beam elements in Presto employ a nonlinear formulation and differ significantly from NASTRAN beams. NASTRAN bars and beams are translated to the closest Presto beams.

vii. The NASTRAN and Presto shell formulations are radically different. Presto employs a 6 dof/node nonlinear formulation.

viii. Presto does not support multiple orthotropic layers.

2.9.1. Property Cards (PCARDs)

Translation of properties is very specific to the application code. Tables 2-5 and 2-6 list the translated PCARDs and detail property fields that are partially or completely skipped in translation.

Table 2-5. – PCARD Translation for Salinas.

NASTRAN	partial fields	skipped fields
PBAR		C_i D_i E_i F_i K_i
PBARL		some types
PBEAM	A-end only	C_i D_i E_i F_i K_i S_i CW M_i N_i
PBEAML	A-end only	some types. SO_i
PCOMP		SB FT TREF GE LAM SOUTi
PELAS		GEi Si
PGAP		MU1 MU2 TMAX MAR TRMIN
PSHELL		$12I/T^3$ MID2 MID3 TS/T Z1 Z2 MID4
PSOLID		IN STRESS ISOP FCTN
PROD		J C

Table 2-6. – PCARD Translation for Presto.

NASTRAN	partial fields	skipped fields
PBARL	TYPE	some types GROUP NSM
PBEAML	TYPE A-end only	some types NSM_i SO_i X_i
PSHELL		$12I/T^3$ MID2 MID3 TS/T Z1 Z2 MID4 NSM
PSOLID		CORDM IN STRESS ISOP FCTN

2.9.2. Rigid Pseudo Element

NASTRAN contains a rich set of rigid elements that are essential to structural analysis. The main features of these rigid elements are translated by NasGen and are available within Salinas and, to a lesser extent, Presto. There are some significant differences that are outlined below. All such rigid elements depend on selection criteria for the dependent and independent degrees of freedom. In NASTRAN, these decisions are explicitly specified in the input. Salinas and Presto compute these sets as described below.

Salinas Rigid Element Limitations

RROD translates into a beam for visualization. Each beam constrains 1 DOF.

RBAR translates into a beam for visualization. The number of constraint equations depends on the coordinate mask (Section 2.9.2.1).

RTRPLT translates into 3 RBAR type elements.

RBE1 does not translate.

RBE2 translates into a collection of RBAR elements. The number of constraint equations depends on the coordinate mask (Section 2.9.2.1).

RBE3 translates into a collection of beams. Each RBE3 generates a new element block.

Presto Rigid Element Limitations

NasGen translation for Presto handles all rigid RBAR, RROD, RTRPLT and RBE2 elements together, sorting out which rigid elements form separate rigid bodies. The rigid elements are written to a single element block while the separate rigid bodies are distinguished by an attribute value written on the rigid elements in the Exodus file. Presto reads this attribute and treats the corresponding elements as rigid bodies. RBE3 elements are different and translate to MPCs for Presto (though with only partial support).

RBAR translates into a beam for visualization. Each beam constrains 6 DOFS.

RROD translates into a collection of RBAR elements. (Each RBAR constrains 6 DOFs.)

RTRPLT translates into a collection of RBAR elements. (Each RBAR constrains 6 DOFs.)

RBE2 Fully constrained RBE2 (those that constrain all six DOFS) translates into a collection of RBAR elements. (Each RBAR constrains 6 DOFs.) Partially constrained RBE translate into an MPC with one independent node and the remaining nodes dependent.

RBE3 translates to an MPC based on the first independent-dependent set of nodes in the RBE3. A collection of beams is also written to the Exodus file for visualization, but the block containing these bars is omitted in the Presto input file (since the MPC is the translation of the RBE3 element).

Presto PCOMP Limitations

Presto does not support a composite shell capability. The PCOMP card is translated to be a layered shell capability. Currently, there is no support for various fiber orientations per layer or any material other than elastic. This is a limited capability and should be used carefully since Presto and Salinas do not offer the same capabilities with respect to this card.

2.9.2.1. Coordinate Mask

NASTRAN input employs a coordinate mask on some fields. For example, the fourth field of an RBE2 describes the coordinate degrees of freedom which are active for the dependent nodes of that pseudo-element.

Early versions of NASGEN did not parse that field, but constrained all the structural degrees of freedom associated with the nodes. Current code [1] stores information in the exodus element attributes that are used to restrict the constraint equations applied. Because Exodus attributes are always stored as real numbers, the NASTRAN coordinate mask is mapped to a real number representative of the active dofs. The position in the exodus mask is significant. There are two attributes associated with each element which refer to the nodes on each end.

Defining $\gamma(x)$ to be 1 if dof x is active, and 0 otherwise, the exodus attribute is set as follows,

$$mask = 100000\gamma(x) + 20000\gamma(y) + 3000\gamma(z) + 400\gamma(R_x) + 50\gamma(R_z) + 6\gamma(R_z)$$

Thus, $mask = 123456$ when all dofs are active, and 123000 if only translations are active.

2.10. Recognized BDF cards

The following cards are at least partially recognized in the NasGen translator.

CBAR, CBEAM, CBUSH, CDAMP, CELAS1, CELAS2, CHEXA, CONM2, CORD2C, CORD2R, CORD2S, CPENTA, CQUAD4, CQUAD8, CROD, CTETRA, CTRIA3, CTRIA6, DAREA, FORCE, FORCE1, FORCE2, FREQ1, FREQ2 GRID, MPC, MPCADD, MAT1, MAT8, MOMENT, MOMENT1, MOMENT2, PBAR, PBARL*, PBEAM, PBUSH, PCOMP, PDAMP, PGAP, PARAM, PLOAD, PSHELL, PSOLID, RBAR, RBE1, RBE2, RBE3, RROD, RTRPLT, SPC, SPC1, SPCD, TABLE

[1] Revisions as of 2/2011 or later.

2.10.1. Additional Salinas BDF Card Support Details

CHEX Only CHEX elements with 8 or 20 grid points are supported.

CONM2 CONM2 cards with a CID not equal to 0 are not well tested. The values of the mass moments of inertia exported to Exodus should be examined to ensure that they were translated correctly.

CORDiC CORDiR CORDiS All coordinate systems are fully supported, but the frames are not nested. Each frame is redefined in terms of the basic frame.

CPENTA Only CPENTA elements with 6 or 15 grid points are supported.

CQUAD Only CQUAD elements with 4 or 8 grid points are supported.

CQUAD4 CQUAD8 The element thickness exported to Exodus is an average of T1-T4. If any thickness is blank, then it is replaced with T1. If thickness is not specified, then the value of the associated PSHELL card is used. THETA and MCID are ignored.

CQUADR CQUADX Not supported.

CROD Fully supported.

CTETRA Only CTETRA elements with 4 or 10 grid points are supported.

CTRIA3 CTRIA6 The element thickness exported to Exodus is an average of T1-T3. Blank thicknesses are replaced with T1. The default thickness is the thickness of the associated PSHELL card is used. THETA and MCID are ignored.

CTRIAR CTRIAX Not supported.

FORCE FORCE1 FORCE2 Fully supported

GRID Permanent single-point constraints are not supported. An SPC1 card must be used instead. If a GRID card with single-point constraints is present, then a warning will be generated.

MOMENT MOMENT1 MOMENT2 Fully supported.

PBAR A, I1, and I2 used. All other fields ignored.

PBARL A few of the more common cross sections are converted to equivalent PBAR terms (A, I1, I2).

PBEAM A, I1, and I2 used. All other fields ignored.

PLOAD1 PLOAD2 PLOAD4 PLOAD1 cards with moment load types are treated as force load types and a warning is generated. PLOAD1 cards are not well tested; they should be checked after translation. PLOAD2 and PLOAD4 cards fully supported, including cards defined with “Thru” field.

PROD A used, all other fields ignored.

PSHELL T used, all other fields ignored.

PSOLID Ignored.

RBAR RROD RTRPLT All elements are assigned to one Exodus element block. Limited support.

RBE1 Not supported.

RBE2 RBE3 Each element is assigned to its own Exodus element block. Limited support.

SPC SPC1 SPCD Fully supported. A nodeset per degree of freedom is generated for each constraint set.

3. NASGEN SUPPORT

3.1. Known Deficiencies

The following deficiencies exist in the code at the time of this writing.

Sidesets: Processing of sidesets on element edges has not yet been completed. At this point, it is recommended that these be added later by hand if necessary using a tool such as `p3/exodus`.

element offsets: Offsets for beam elements are translated from the values at one end, i.e. the offsets at each end are assumed to be identical and are taken from BDF fields 14-16 of the CBEAM card.

permanent constraints: Constraints introduced on the GRID BDF card are currently ignored.

TLoads: Time dependent loads are not currently translated.

parameters: Most NASTRAN parameters are not read or translated. The Exodus file would not normally depend on these.

uppercase: Translation of lowercase BDF files has not been thoroughly tested. Convert them to uppercase first.

Limited BDF format: Most test cases have been run using the small BDF format (i.e. 8 characters/field). In a few cases the large format has been used, but it is not extensively tested. The free form (comma delimited) field format is also not well tested.

Free form continuations: Continuation cards using the free form format (comma delimited) are not well tested. The blank delimited free field format is not supported.

Free form truncation: Free format fields are *truncated* at 8 characters, rather than rounding.

Replication: Replication, in either fixed or free field formats, is not supported.

arbitrary continuations Only continuation cards immediately following the parent cards are supported.

There are many BDF cards that are not translated by this translator. These can usually be determined using the "-u" option, which will attempt to list cards that were not translated to the Exodus or input files (or both). In addition, many cards are only partly translated. For example, pin information on beam cards is silently ignored. *It is critical that the analyst inspect the resulting translation to ensure that it is complete for his application.*

3.2. Support

Nasgen is provided to assist in the migration of structural models from NASTRAN to Exodus formats. The Sierra Structrual Dynamics and Solid Mechanics Team will try to fix bugs as they are found. Send bug reports and issues to sierra-help@sandia.gov. Clearly identify the program (NasGen) and if possible, include an example BDF file fragment.

3.3. Pedigree

NasGen grew out of a previous translator, `pat2exo`, which generated an Exodus file from a patran neutral file. Richard Naething wrote the core elements of NasGen during the summer of 2001. Additional work and functionality was added by Garth Reese in early 2002, and Richard Naething during the summer of 2002. Extensive additions were made in 2009 and 2010 by the Sierra Structural Dynamics and Solid Mechanics Team.[1]

REFERENCES

[1] Edward A. Bourcheron et al. *Sandia National Laboratories Advanced Simulation and Computing (ASC) Software Quality Plan Part 1: ASC Software Quality Engineering Practices Version 1.0*. Tech. rep. SAND2004-6602. PO Box 5800, Albuquerque, NM 87185-5800: Sandia National Laboratories, 2005 (cit. on p. 24).

[2] F. Fuentes et al. "Orientation embedded high order shape functions for the exact sequence elements of all shapes". In: *Computers and Mathematics with Applications* 70.1 (2015), pp. 353–458 (cit. on p. 8).

DISTRIBUTION

Email—Internal

Name	Org.	Sandia Email Address
Technical Library	1911	sanddocs@sandia.gov

Hardcopy—Internal

Number of Copies	Name	Org.	Mailstop
1	K. H. Pierson	1542	0845

Sandia
National
Laboratories

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.