SIEMENS

NX Nastran 11.0.2 Release Guide

Contents

Proprietary & Restricted Rights Notice

TAUCS Copyright and License

TAUCS Version 2.0, November 29, 2001. Copyright (c) 2001, 2002, 2003 by Sivan Toledo, Tel-Aviv University, stoledo@tau.ac.il. All Rights Reserved.

TAUCS License:

Your use or distribution of TAUCS or any derivative code implies that you agree to this License.

THIS MATERIAL IS PROVIDED AS IS, WITH ABSOLUTELY NO WARRANTY EXPRESSED OR IMPLIED. ANY USE IS AT YOUR OWN RISK.

Permission is hereby granted to use or copy this program, provided that the Copyright, this License, and the Availability of the original version is retained on all copies. User documentation of any code that uses this code or any derivative code must cite the Copyright, this License, the Availability note, and "Used by permission." If this code or any derivative code is accessible from within MATLAB, then typing "help taucs" must cite the Copyright, and "type taucs" must also cite this License and the Availability note. Permission to modify the code and to distribute modified code is granted, provided the Copyright, this License, and the Availability note are retained, and a notice that the code was modified is included. This software is provided to you free of charge.

Availability (TAUCS)

As of version 2.1, we distribute the code in 4 formats: zip and tarred-gzipped (tgz), with or without binaries for external libraries. The bundled external libraries should allow you to build the test programs on Linux, Windows, and MacOS X without installing additional software. We recommend that you download the full distributions, and then perhaps replace the bundled libraries by higher performance ones (e.g., with a BLAS library that is specifically optimized for your machine). If you want to conserve bandwidth and you want to install the required libraries yourself, download the lean distributions. The zip and tgz files are identical, except that on Linux, Unix, and MacOS, unpacking the tgz file ensures that the configure script is marked as executable (unpack with tar zxvpf), otherwise you will have to change its permissions manually.

Chapter 1: SOL 401 Enhancements

Distributed force to a surface or edge

You can now request that the software use forces and moments that you define at a single point to automatically compute and apply a traction to a surface or edge region. The software computed traction has a specific form and is described in Traction computation details.

You use the new FORCDST bulk entry to define the point location, the forces and moments at the point, and to select the element or edge regions for the traction location. The element face and edge regions that receive the traction are selected as follows.

- You use the BSURF entry to create face regions on the shell elements CTRIA3, CTRIA6, CTRIAR, CQUAD4, CQUAD8, and CQUADR.
- You use the BSURFS entry to create face regions on the solid elements CHEXA, CPENTA, CPYRAM, and CTETRA.
- You use the BEDGE entry to create edge regions on the following elements,

  Axisymmetric elements CTRAX3, CQUADX4, CTRAX6, CQUADX8.

  Plane stress elements CPLSTS3, CPLSTS4, CPLSTS6, CPLSTS8.

  Chocking elements CCHOCK3, CCHOCK4, CCHOCK6, CCHOCK8

You have the option on the FORCDST bulk entry to define the single point location for your forces and moments at a grid point, at a coordinate system origin, or at the centroid of the surface or edge region you have selected to receive the traction. The software automatically computes the centroid location.

For a time-unassigned load, you reference the SID of the FORCDST bulk entry and any other mechanical loads with the LOAD=n case control command either in a subcase or above the subcases (globally).

For example,

```
...
SUBCASE 5
LOAD = 150
BEGIN BULK
...
$ TIME-UNASSIGNED LOADS FOR SUBCASE 5
FORCDST,150,10,,CNTRD,,,,,+
+,1.0,23.4,428.6,84,2,49.7,7846.4,5876.5
...
RFORCE,150,5,,-6.4,0.0,0.0,1.0,2
...
```

For a time-assigned load, you must select the FORCDST bulk entry with the EXCITEID on the new TLOAD3 bulk entry. The TLOAD3 entry includes six TID fields that you can use to independently scale the three force and the three moment components with time by referencing a unique TABLEDi

entry for each. You can also reference the same TABLEDi entry for any or all TID fields to define the same load variations. The FORCDST entry is the only load you can select with the TLOAD3 entry. All other time-assigned mechanical loads are selected by the EXCITEID of the TLOAD1 bulk entry, which scales all referenced loads consistently with time. You can combine TLOAD1 and TLOAD3 entries into a single load set with the DLOAD bulk entry.

For example,

```
...
SUBCASE 5
$DLOAD CASE CONTROL SELECTS THE DLOAD BULK ENTRY
DLOAD = 250
BEGIN BULK
...
$DLOAD BULK ENTRY COMBINES TLOAD3 (ID=302) and TLOAD1 (ID=305)
DLOAD,250,1.,1.,302,1.,305
...
$TIME-ASSIGNED FORCDST, EXCITEID=125, TIME FUNCTION FOR EACH COMPONENT
TLOAD3,302,125,,,,,,,+
+,,401,402,403,,501,502,503
...
FORCDST,125,10,,CNTRD,,,,,+
+,1.0,23.4,428.6,84.2,49.7,78.4,56.5
...
$SOLID ELEMENT REGION TO RECEIVE TRACTION
BSURFS,10,,,,8,10,15,40,+
+,12,15,20,45,16,20,25,50,+
+,24,35,40,65,28,40,45,70,+
+,32,45,50,75,40,60,65,90
...
$TIME FUNCTIONS FOR EACH COMPONENT DEFINED ON FORCDST
TABLED2,401,0.,+
+,0.,0.,1.,100.,2.,0.,ENDT
TABLED2,402,0.,+
+,0.,0.,1.,120.,2.,0.,ENDT
TABLED2,403,0.,+
+,0.,0.,1.,140.,2.,0.,ENDT
TABLED2,501,0.,+
+,0.,0.,1.,80.,2.,0.,ENDT
TABLED2,502,0.,+
+,0.,0.,1.,90.,2.,0.,ENDT
TABLED2,503,0.,+
+,0.,0.,1.,100.,2.,0.,ENDT
...
$TIME-ASSIGNED RFORCE, EXCITEID=150, TIME FUNCTION TID=13
TLOAD1,305,125,13
$TIME FUNCTION 13 USED FOR FORCE LOAD
TABLED2 13 0. + 0. 0. 1. 100. 2. 0. ENDT
RFORCE,150,5,,-6.4,0.0,0.0,1.0,2
...
```

The FORCDST bulk entry includes the optional point location offset. This offset is simply added to the point location defined on the FORCDST bulk entry. The software does not scale the point location or the offset with time. The TABLEDi entry you reference from a TID field only scales the associated force or moment value. As a result, the location of the forces and moments remain constant for a solution.

The software always computes the traction based on the undeformed configuration. It does not take into account large displacements or rotations when recomputing the traction at a time step.

Traction computation details

Figure 1-1 shows an example element patch to illustrate the procedure. You define the FORCDST bulk entry to apply the forces and moments at point D or optionally at the centroid (C), and to select the element patch where the software computes and applies the traction.

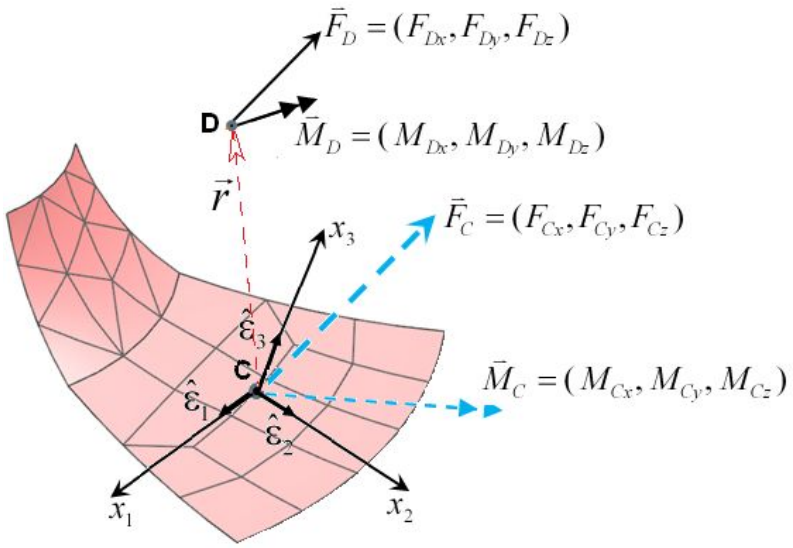


Figure 1-1. Element patch and load locations

The software computes the centroid (C) of the element patch. If you defined your forces and moments at D, the software transfers them to an equivalent system at C using a rigid link transformation,

$$\vec{F}_C = \vec{F}_D$$

$$\vec{M}_C = \vec{M}_D + \vec{r} \times \vec{F}_D$$

where $\vec{r}$ is the vector from C to D.

A local coordinate system is computed on the patch using the eigenvectors of the patch area inertia matrix. The eigenvector associated with the largest eigenvalue is the x_3 direction, which is generally perpendicular to the patch. The next two eigenvectors become the x_1 and x_2 directions, respectively.

The software will compute the following traction on the patch:

$$\vec{T} = (C_4 + C_6 x_2^L)\hat{\varepsilon}_1 + (C_5 - C_6 x_1^L)\hat{\varepsilon}_2 + (C_1 + C_2 x_1^L + C_3 x_2^L)\hat{\varepsilon}_3$$

where x_1^L, x_2^L are the coordinates of the patch grid points relative to the local patch coordinates $(\hat{\varepsilon}_1, \hat{\varepsilon}_2, \hat{\varepsilon}_3)$.

The assumption is that the computed traction must satisfy the following requirements.

- The resultant force computed using the forces from all grid points on the patch must equal the force resultant at C.

- The resultant moment computed using the moments from all grid points on the patch must equal the moment resultant at C.

Rearranging the terms in the traction gives:

$$\vec{T} = C_1\left[(1)\hat{\varepsilon}_3\right] + C_2\left[(x_1^L)\hat{\varepsilon}_3\right] + C_3\left[(x_2^L)\hat{\varepsilon}_3\right] + C_4\left[(1)\hat{\varepsilon}_1\right] + C_5\left[(x_1^L)\hat{\varepsilon}_2\right] + C_6\left[(x_2^L)\hat{\varepsilon}_1 + (-x_1^L)\hat{\varepsilon}_2\right]$$

The traction vector is written as a superposition of six traction loads:

$$\vec{T} = C_1\{\vec{t}_1\} + C_2\{\vec{t}_2\} + C_3\{\vec{t}_3\} + C_4\{\vec{t}_4\} + C_5\{\vec{t}_5\} + C_6\{\vec{t}_6\}$$

The six traction load cases are as follows (per unit surface area or per unit length).

$\{\vec{t}_1\} = (1)\hat{\varepsilon}_3$ is a uniformly distributed load of unit intensity in the $\hat{\varepsilon}_3$ direction.

$\{\vec{t}_2\} = (x_1^L)\hat{\varepsilon}_3$ is a linearly varying load in the $\hat{\varepsilon}_3$ direction.

$\{\vec{t}_3\} = (x_2^L)\hat{\varepsilon}_3$ is a linearly varying load in the $\hat{\varepsilon}_3$ direction.

$\{\vec{t}_4\} = (1)\hat{\varepsilon}_1$ is a uniformly distributed load of unit intensity in the $\hat{\varepsilon}_1$ direction.

$\{\vec{t}_5\} = (1)\hat{\varepsilon}_2$ is a uniformly distributed load of unit intensity in the $\hat{\varepsilon}_2$ direction.

$\{\vec{t}_6\} = (x_2^L)\hat{\varepsilon}_1 + (-x_1^L)\hat{\varepsilon}_2$ is a linearly varying load in the $\hat{\varepsilon}_1$ and $\hat{\varepsilon}_2$ directions.

The grid point force vector is computed by integrating $\vec{T}$ over the entire patch:

$$\{F\} = \int_S [N]^T \{\vec{T}\}\, dA$$

Substituting $\vec{T} = C_1\{\vec{t}_1\} + C_2\{\vec{t}_2\} + C_3\{\vec{t}_3\} + C_4\{\vec{t}_4\} + C_5\{\vec{t}_5\} + C_6\{\vec{t}_6\}$ into the equation above gives:

$$\{F\} = C_1\int_S [N]^T \{\vec{t}_1\}\, dA + C_2\int_S [N]^T \{\vec{t}_2\}\, dA + C_3\int_S [N]^T \{\vec{t}_3\}\, dA + C_4\int_S [N]^T \{\vec{t}_4\}\, dA + C_5\int_S [N]^T \{\vec{t}_5\}\, dA + C_6\int_S [N]^T \{\vec{t}_6\}\, dA$$

Each term can be written as a grid point force vector:

$$\int_S [N]^T \{\vec{t}_j\}\, dA = \{\vec{f}^{\,j}\} \quad j = 1,2,3,4,5,6$$

The grid point force vector for the patch is a linear combination of the grid point force vectors from each load case:

$$\{F\} = C_1\{\vec{f}^{\,1}\} + C_2\{\vec{f}^{\,2}\} + C_3\{\vec{f}^{\,3}\} + C_4\{\vec{f}^{\,4}\} + C_5\{\vec{f}^{\,5}\} + C_6\{\vec{f}^{\,6}\}$$

The original total load requirements are applied here. This results in a 6x6 system of equations that the software uses to solve for C1, C2, …C6.

- The resultant force of all grid point forces on the patch must be the same as the forces at point C. The resultant force is computed by:

$$\vec{F}_c = \sum_{j=1}^{6} \sum_{i=1}^{N} C_j \vec{f}_i^{\,j} = (F_{cx}, F_{cy}, F_{cz})$$

- The resultant moment of all grid point forces on the patch must be the same as the moments at point C. The resultant moment is computed by:

$$\vec{M}_c = \sum_{j=1}^{6} \sum_{i=1}^{N} \vec{r}_i \times (C_j \vec{f}_i^{\,j}) = (M_{cx}, M_{cy}, M_{cz})$$

  where $\vec{r}$ is the grid point position with respect to the centroid (C).

There are now six equations to solve for the six unknowns C_1, C_2, ….,C_6.

$$\vec{F}_c = \left(\sum_{i=1}^{N} \vec{f}_i^{\,1}\right) C_1 + \left(\sum_{i=1}^{N} \vec{f}_i^{\,2}\right) C_2 + \left(\sum_{i=1}^{N} \vec{f}_i^{\,3}\right) C_3 + \left(\sum_{i=1}^{N} \vec{f}_i^{\,4}\right) C_4 + \left(\sum_{i=1}^{N} \vec{f}_i^{\,5}\right) C_5 + \left(\sum_{i=1}^{N} \vec{f}_i^{\,6}\right) C_6$$

$$\vec{M}_c = \left(\sum_{i=1}^{N} \vec{r}_i \times \vec{f}_i^{\,1}\right) C_1 + \left(\sum_{i=1}^{N} \vec{r}_i \times \vec{f}_i^{\,2}\right) C_2 + \left(\sum_{i=1}^{N} \vec{r}_i \times \vec{f}_i^{\,3}\right) C_3 + \left(\sum_{i=1}^{N} \vec{r}_i \times \vec{f}_i^{\,4}\right) C_4 + \left(\sum_{i=1}^{N} \vec{r}_i \times \vec{f}_i^{\,5}\right) C_5 + \left(\sum_{i=1}^{N} \vec{r}_i \times \vec{f}_i^{\,6}\right) C_6$$

These equations are resolved into scalar equations.

$$\vec{F}^{\,j} = F_x^j \hat{i} + F_y^j \hat{j} + F_z^j \hat{k}$$

$$\vec{M}^{\,j} = M_x^j \hat{i} + M_y^j \hat{j} + M_z^j \hat{k}$$

The software uses the following 6x6 matrix to compute C_1, C_2, ….,C_6.

$$\begin{bmatrix} F_x^1 & F_x^2 & F_x^3 & F_x^4 & F_x^5 & F_x^6 \\ F_y^1 & F_y^2 & F_y^3 & F_y^4 & F_y^5 & F_y^6 \\ F_z^1 & F_z^2 & F_z^3 & F_z^4 & F_z^5 & F_z^6 \\ M_x^1 & M_x^2 & M_x^3 & M_x^4 & M_x^5 & M_x^6 \\ M_y^1 & M_y^2 & M_y^3 & M_y^4 & M_y^5 & M_y^6 \\ M_z^1 & M_z^2 & M_z^3 & M_z^4 & M_z^5 & M_z^6 \end{bmatrix} \begin{Bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \end{Bmatrix} = \begin{Bmatrix} F_{cx} \\ F_{cy} \\ F_{cz} \\ M_{cx} \\ M_{cy} \\ M_{cz} \end{Bmatrix}$$

For time-unassigned loads, this matrix is used once to compute C_1, C_2, ….,C_6.

For time-assigned loads, this matrix is reused to recompute C_1, C_2, ….,C_6 at each time step. The right-hand side is updated with the updated force and moment values at each time step.

FORCDST

Forces and moments to be distributed

Defines forces and moments at a point that the software will automatically distribute to element faces or edges in the form of a specific traction. Only valid for SOL 401.

FORMAT:

1	2	3	4	5	6	7	8	9	10
FORCDST	SID	RID	CID	PLOC	G	O1	O2	O3	
	F	FN1	FN2	FN3	M	MN1	MN2	MN3	

EXAMPLE:

FORCDST	3	6	100	ORIG	21.3				
		322.4	219.3	530.5		14.5	25.4	12.4	

FIELDS:

Field	Contents
SID	Set identification number. See Remark 2. (Integer>0)
RID	Region identification number. See Remark 3. (Integer>0)
CID	Coordinate system identification number used to define the components FNi and MNi, the location of FNi and MNi when PLOC="ORIG", and the components of Oi. See Remark 4. (Integer ≥ 0; Default=0 selects the basic CS)
PLOC	Reference location of the point load. See Remark 5. (Character: ORIG, CNTRD, or GRID)
G	Grid identification number when PLOC=”GRID”. See Remark 5. (Integer>0)
Oi	Components of an offset vector relative to the coordinate system selected by CID. It moves the load point from the location defined by PLOC to a new offset location. (Real; Default=0.0.)
F	Scale factor of force components. See Remark 6. (Real; No default)
FNi	Components of a force vector relative to the coordinate system selected by CID. (Real; at least one FNi ≠ 0.0.)
M	Scale factor of moment components. See Remark 6. (Real; No default)
MNi	Components of a moment vector relative to the coordinate system selected by CID. (Real; at least one MNi ≠ 0.0.)

REMARKS:

1. The software uses the forces and moments that you define at the single point on the FORCDST bulk entry to automatically compute and apply a traction to a surface or edge patch. The traction variation relative to the software computed patch coordinate system is:

$$\vec{T} = T_1\hat{\varepsilon}_1 + T_2\hat{\varepsilon}_2 + T_3\hat{\varepsilon}_3$$
$$T_3 = C_1 + C_2 x_1^L + C_3 x_2^L$$
$$T_1 = C_4 + C_6 x_2^L$$
$$T_2 = C_5 - C_6 x_1^L$$

   where $(\hat{\varepsilon}_1, \hat{\varepsilon}_2, \hat{\varepsilon}_3)$ are the software computed unit vectors which define the patch coordinate system, and x_1^L, x_2^L are the coordinates of the grid points on the patch with respect to the patch coordinate system.

2. For a time-unassigned load, you reference the SID of one or multiple FORCDST bulk entry and any other mechanical loads with the LOAD=n case control command either in a subcase or above the subcases (globally). Tme-unassigned loads with different SID can also be combined and optionally scaled with the LOAD bulk entry.

   For a time-assigned load, you reference the SID of one or multiple FORCDST bulk entry with the EXCITEID on the TLOAD3 bulk entry. The TLOAD3 entry includes six TID fields that you can use to independently scale the three force and the three moment components with time by referencing a unique TABLEDi entry for each. You can also reference the same TABLEDi entry for any or all TID fields to define consistent load variations. The FORCDST entry is the only load you can select with the TLOAD3 entry. All other time-assigned mechanical loads are selected by the EXCITEID of the TLOAD1 bulk entry, which scales all referenced loads consistently with time. You can combine TLOAD1 and TLOAD3 entries into a single load set with the DLOAD bulk entry.

   Note that the SID of a FORCDST bulk entry and the SID of any other mechanical load such as the FORCE entry can only be the same when the SID is selected as a time-unassigned load. For a time-assigned load, the SID of FORCDST bulk entries must be unique to the SID of all other mechanical loads.

3. The RID field selects a region of element edges or faces where the software computes and applies the traction. The regions are defined with the BSURFS, BSURF, and BEDGE bulk entries. The BSURFS entry creates a face region on the solid elements CHEXA, CPENTA, CPYRAM, and CTETRA. The BSURF entry creates a face region on the shell elements CTRIA3, CTRIA6, CTRIAR, CQUAD4, CQUAD8, and CQUADR. The BEDGE entry creates a edge region on the axisymmetric elements CTRAX3, CQUADX4, CTRAX6, CQUADX8, the plane stress elements CPLSTS3, CPLSTS4, CPLSTS6, CPLSTS8, and the chocking elements CCHOCK3, CCHOCK4, CCHOCK6, CCHOCK8.

4. CID=0 or blank (the default) references the basic coordinate system.

5. The location options for PLOC are “ORIG” for the origin of the coordinate system selected by CID (field 4), “CNTRD” for the centroid of the surface or edge region selected by RID (field 3), or “GRID” for the location of the grid point selected in G (field 6). The software automatically computes the centroid location. The PLOC field does not have a default, so it must be defined.

6. The software multiplies the scale factor F by the components FN1, FN2, FN3. The scale factor F must be defined, and at least one of FNi must be defined.

   The software multiplies the scale factor M by the components MN1, MN2, MN3. The scale factor M must be defined, and at least one of MNi must be defined.

7. For axisymmetric element edges, the forces and moments defined on the FORCDST entry are the total of the forces and moments applied to the full theoretical 360 degree area. For plane stress element edges, the forces and moments defined on the FORCDST entry are the total of the forces and moments applied to the full theoretical area of thickness t. For example, the total forces and moments defined on the FORCDST entry would be the same whether you want the traction computed and applied to solid or shell element faces, or to element edges.

TLOAD3

Time-dependent load defined with the FORCDST entry in SOL 401

References a time-dependent load defined with the FORCDST entry in SOL 401. The load has the form:

$$\left\{\overline{P(t)}\right\} = \begin{Bmatrix} \vec{F}_D(t) \\ \vec{M}_D(t) \end{Bmatrix} = \begin{Bmatrix} F * \begin{Bmatrix} FN1 * TFN1(t-\tau) \\ FN2 * TFN2(t-\tau) \\ FN3 * TFN3(t-\tau) \end{Bmatrix} \\ M * \begin{Bmatrix} MN1 * TMN1(t-\tau) \\ MN2 * TMN2(t-\tau) \\ MN3 * TMN3(t-\tau) \end{Bmatrix} \end{Bmatrix}$$

where F, FNi, M, and MNi are defined on the FORCDST entry, and TFNi, TMNi, and τ are defined on the TLOAD3 entry. The time steps (t) are defined with the TSTEP1 entry.

FORMAT:

1	2	3	4	5	6	7	8	9	10
TLOAD3	SID	EXCITEID	DELAY						
		TFN1	TFN2	TFN3		TMN1	TMN1	TMN1	

EXAMPLE:

TLOAD3	5	7	0.01		13				
		100	101	102		103	104	105	

FIELDS:

Field	Contents
SID	Load set identification number. (Integer>0)
EXCITEID	Identification number of a FORCDST bulk entry. (Integer>0)
DELAY	Time delay τ for all degree-of-freedom referenced by the FORCDST entry. (Real≥0.0 or blank; Default=0.0).
TFNi	Identification number of TABLEDi entry that defines the time dependent scale factors for FNi force components defined on the FORCDST entry. (Blank or Integer>0; Default=1.0). See Remark 2.
TMNi	Identification number of TABLEDi entry that defines the time dependent scale factors for MNi moment components defined on the FORCDST entry. (Blank or Integer>0; Default=1.0). See Remark 2.

REMARKS:

1. A single TLOAD3 entry can be selected directly by the DLOAD=SID case control command, or by the DLOAD bulk entry to combine and optionally scaled with other TLOAD3 or TLOAD1 bulk entries.

2. When a blank value is used for TFNi or TMNi, the software applies a unit scale factor for all times.

3. SID must be unique on all TLOAD1 and TLOAD3 entries.

Contact enhancement

The new JOINT=5 parameter option is available on the BCTPARM bulk entry to select a small sliding and frictionless contact option. The new option helps to reduce the solution time when you are confident that your contact pairs will only experience very small sliding, and that tangential contact stiffness is not required. It can be used with the large displacement capability on (LGDISP = +1) or off (LGDISP = -1). For either case, the contact elements are only formed once at the beginning of the solution.

When the small sliding option is requested (JOINT=5), for each load step, the software iterates to determine the following:

- For the contact elements which are formed at the beginning of the solution, the software determines if each contact element is active or inactive.

- The software determines the resulting normal contact forces required to prevent the regions from penetrating.

The standard option is used by default (JOINT=1). With the standard option and the large displacement capability on, if significant sliding occurs, the software does recreate new contact elements for the updated deformed state. There is a computation expense for this, but it is important for an accurate solution when significant sliding occurs.

When you use the standard option (JOINT=1), the software always uses the Full Newton stiffness matrix update method regardless of how you define the KUPDATE parameter on the NLCNTL bulk entry. When you use the small sliding option (JOINT=5), the software uses the stiffness update method selected with the KUPDATE parameter. For the best performance with the small sliding option (JOINT=5), it is recommended that you use the default, automatic stiffness update method (KUPDATE=0).

Table 1-1. Contact parameter for SOL 401:

Name	Description
JOINT	Selects the interface behavior between source and target region. = 1 (Default) Standard Standard segment-segment contact algorithm with large sliding (pairing updates) and friction. = 5 Small Sliding Simplified contact algorithm with no pairing updates or friction.

Convergence functions

A nonlinear solution iterates until the software computed convergence error functions are less than the convergence criteria you have defined for each function. A convergence error function is calculated with a ratio which compares the value of displacement, force, or work for the current increment to the total value of value of displacement, force, or work.

The SOL 401 displacement and force convergence error functions now include the following options. The work convergence error function has not changed.

You can now define the new NORMU and NORMP parameters on the NLCNTL bulk entry as L1, L2, or LINF to select a norm criteria of 1, 2, or infinity, respectively. The NORMU parameter applies to the displacement error function and the NORMP parameter applies to the force error function. The defaults NORMU=L2 and NORMP=L2 are consistent with the previous software behaviour.

In general, the norm computation results in a strictly positive value from a vector space v_n such that:

$$NORMU\ or\ NORMP = L1 \quad \rightarrow \quad \|v\|_1 = |v_1| + |v_2| + ... + |v_n|$$

$$NORMU\ or\ NORMP = L2 \quad \rightarrow \quad \|v\|_2 = \sqrt{{v_1}^2 + {v_2}^2 + ... + {v_n}^2}$$

$$NORMU\ or\ NORMP = \text{LINF} \rightarrow \quad \|v\|_\infty = \max\left(|v_1|, |v_2|, ..., |v_n|\right)$$

- The updated displacement convergence error function with the new norm criteria is as follows.

$$\varepsilon_U^i = \frac{\left\| U_n^i - U_n^{i-1} \right\|_{normu}}{\left\| U_n^i - U_{n-1} \right\|_{normu}}$$

  where,

  U_n^i is the displacement vector for iteration i of time step n,

  U_n^{i-1} is the displacement vector for iteration i-1 of time step n, and

  U_{n-1} is the displacement vector at convergence for time step n-1.

  In addition, when you define the new REFU parameter on the NLCNTL bulk entry, the form of the displacement error function is as follows.

$$\varepsilon_U^i = \frac{\left\| U_n^i - U_n^{i-1} \right\|_{normu}}{REFU}$$

- The updated force convergence error function with the new norm criteria is as follows.

$$\varepsilon_p^i = \frac{\left\| R_n^i \right\|_{normp}}{\max\left(\left\| P_n^i \right\|_{normp}, \left\| F_n^i \right\|_{normp}, \left\| R_n^1 \right\|_{normp} \right)}$$

  where,

  P_n^i is the external force vector for iteration i of time step n,

  F_n^i is the internal force vector for iteration i of time step n,

  R_n^i is the residual force vector for iteration i of time step n, and

R_n^1 is the residual force vector for iteration 1 of time step n.

In addition, when you define the new REFP parameter on the NLCNTL bulk entry, the form of the force error function is as follows.

$$\varepsilon_p^i = \frac{\left\| R_n^i \right\|_{normp}}{REFP}$$

- The following work convergence error function has not changed.

$$\varepsilon_W^i = \frac{ABS\left(\Delta U_n^i . R_n^i\right)}{\max\left(ABS\left(U_n^i . P_n^i\right), ABS\left(U_n^i . F_n^i\right), ABS\left(U_n^1 . R_n^1\right)\right)}$$

Where,

ΔU_n^i is the incremental displacement vector computed for iteration i of time step n,

R_n^i is the residual force vector for iteration i of time step n,

U_n^i is the total displacement vector for iteration i of time step n,

P_n^i is the external force vector for iteration i of time step n,

F_n^i is the internal force vector for iteration i of time step n,

R_n^1 is the residual force vector for iteration 1 of time step n, and

U_n^1 is the total displacement vector for iteration 1 of time step n.

Fourier harmonic solution update

In NX Nastran 11.0, the Fourier normal modes subcase became available for SOL 401 with models which include axisymmetric elements. The subcase is designated with the ANALYSIS=FOURIER and HARMONICS=N case control commands in the subcase.

Now in NX Nastran 11.0.2, for grid points which are defined on the rotation axis, in addition to any conditions that you defined, NX Nastran automatically applies the following SPC and MPC conditions during the solution.

- For the harmonic index k=0, NX Nastran fixes the radial (U_r) and azimuth (U_θ) degrees-of-freedom.
- For the harmonic index k=1, NX Nastran fixes the axial degree-of-freedom, and it creates the MPC condition $U_r = U_\theta$ for the cosine terms, and the MPC condition $U_r = -U_\theta$ for the sine terms.
- Now beginning in NX Nastran 11.0.2 for a harmonic index k>1, NX Nastran fixes all translational degrees-of-freedom.

Note:

When axisymmetric elements are defined on the XZ-plane of the basic coordinate system, the X-axis of the basic system is the radial direction, the Z-axis of the basic system is the axial direction, and the Y-axis of the basic system is the azimuth direction. The grid points defining these elements must have $X \geq 0$.

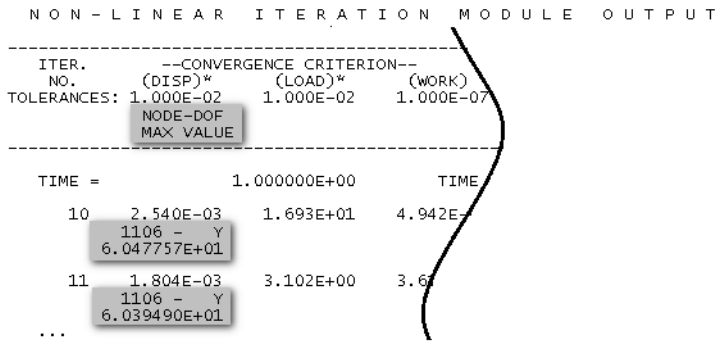
When axisymmetric elements are defined on the XY-plane of the basic coordinate system, the Y-axis of the basic system is the radial direction, the X-axis of the basic system is the axial direction, and the Z-axis of the basic system is the azimuth direction. The grid points defining these elements must have Y ≥ 0.

Message level enhancement

You can use the existing MSGLVL=1 parameter option on the NLCNTL bulk entry to request diagnostic output from your SOL 401 run.

Now you can use the new MSGLVL=2 option to request the same diagnostic output as MSGLVL=1 plus the location and value of the maximum translational displacement. The additional data printed with MSGLVL=2 includes the grid point ID, the translational component number (1, 2, or 3), and the dIsplacement value.

The highlighted text in the example output below shows the additional output when you define MSGLVL=2 on the NLCNTL entry.



The updated MSGLVL parameter is as follows.

MSGLVL Diagnostic level. (Integer = 0 or 1; Default = 1)

0: No additional diagnostic output

1: Convergence information is output for each iteration

2: Convergence information is output for each iteration plus the location and value of the maximum translational displacement.

Chapter 2: Problem Report (PR) fixes

Problem Report (PR) fixes

The NX Nastran 11.0.2 maintenance release includes the following fixes.

PR#	Problem Originated	Problem Description
7931781	8.0	In a SOL 200 run which has designed frequency dependent properties defined, updated TABLED1 bulk entries are written into the punch (pch) file. If the user requests the entire updated bulk data, these same TABLED1 entries are written in both the updated form in the "updated data" part of the punch file, as well as in the un-updated form in the "un-updated data" part of the punch file. The second (un-updated form) should not have been written by the software. This punch file bug has been fixed, such that if a TABLED1 is updated during optimization, it shows up only in the updated part of the punch file bulk data.
7931648	11.0	In a SOL 200 run for a topometry optimization (DVEREL1 bulk data directed optimization), the intention was to write the original PSHELL IDs into the connectivity data in the punch file. Instead, the internally assigned PSHELL IDs are written. The bug that allowed internally generated PSHELL IDs to be written into the punch file for DVEREL1 designed shell elements has been fixed.
8314984 and 8324262	10.0	When a model has a bolt preload defined on both beam element bolts and on solid element bolts, and if the bolt ID of a beam bolt is greater than all the solid bolt IDs, the beam element bolts do not produce the correct axial load. The issue has been fixed. The workaround for versions before NX Nastran 11.0.2 is to modify all solid bolt IDs to be greater than the largest beam bolt ID.

7974016	11.0.1	For the SOL 401 axisymmetric Fourier solution and cyclic modal solutions, when the AF normalization option is selected, the Eigen vectors are renormalized using a special algorithm outside the Eigen solver. After renormalization, the LAMA table needs to be recomputed to get the correct values for modal mass and generalized stiffness corresponding to the renormalzied Eigen vectors. Once the generalized mass and generalized stiffness gets recomputed, the LAMA table is rewritten. The problem with this PR occured while this table was getting written. The routine writing this table (LAMX) was incorrectly writing the header record in the old format and wasn't accounting for the updated record that identified the subcase ID, cyclic/Fourier solution type and the harmonic index. (In the header record for the LAMA table, the 4th word identifies subcase ID, 23rd word identifies solution type (0=default,2=Axi-Fourier,3=cyclic) and the 26th word identifies harmonic index. These values were not being written to the header. The correct header record is now written.
7955760 and 7914076	10.0	In a SOL 401 run, incorrect OTEMP output occured when a time-varying thermal load was applied using a DTEMP bulk entry which referenced a mixture of both TEMP and TEMPD entries. A DTEMP bulk entry defines a series of temperature intervals. The start and the end for any temperature-time interval on DTEMP can be defined using a combination of TEMP and/or TEMPD entries. In a few cases, if the start of an interval was defined by a TEMPD and the end interval using a TEMP (or vice-versa), incorrect nodal temperature values were applied. In such cases, the OTEMP output demonstrated the incorrect nodal temperature values. For versions before NX Nastran 11.0.2, there is no practical work around for this problem. However, defining all temperatures using only TEMP entries (including nodal temperatures not referenced by DTEMP) will avoid the problem.
8324060	11.0	An NX Nastran 11.0 topology optimization job running SOL 103 converged, but the mass density remained constant. The problem was a result of a poor sensitivity analysis. It has been resolved for NX Nastran 11.0.2, and the mass density results for this model demonstate an improved mass density optimization result. In addition, beginning with NX Nastran 11.0.2, the NX Nastran 11 topology optimization now supports multiple frequency (eigenvalue) constraints with multiple frequency calculation. However because of a mode tracking issue, we still recommend to use a single frequency calculation for a single constraint or objective. To use multiple frequencies for multiple frequency

		constraints, the frequency values should not be close to each other.

7926799	9.0	A SOL 103 analysis with STATSUB using an RLOAD fails with the fatal error 6704 which suggested increasing memory.
		
		We found that there are some trailer differences between the KGGG generated in phase1e.dat, and the KGGG imported through output4/input4, even the KGGG should be the same since we used the same version with identical options.
		
		It turns out that, there are some zeros packed in KGGG originally. After output4/input4, the zeros are removed from the sparse matrix, so the trailer of KGGG counts less nonzero terms with smaller density.
		
		With the KGGG imported from input4, the solution completed fine without the fatal in DFMFRD.
		
		The fix is to use "ADD KGGT,KGGG/KGGGL" to replace "ADD5 KGGT,KGGG,,,/KGGGL" in phase1e.dat. Note that ADD will remove zeros from the sparse matrix.
		
		You can use the following alter to avoid the fatal:
		
		`compile phase1e, list $`
		`alter 'ADD5', '' $`
		`ADD KGGT,KGGG/KGGGL/// $`

7937557	1.0	For a SOL 111 run, a model demonstated that stresses for CROD elements where incorrect when the magnitude/phase option was requested. The problem was descovered since the stress magnitudes were reported as negative values. Magnitude/phase output of forces and stresses for the CROD element were never calculated. What was being printed to the .f06 were real/imaginary values, even though the header reported magnitude/phase. This bug has been corrected in the NX Nastran 11.0.2 release.
7782458	11.0	For a pure acoustic fluid model running SOL108 which included a FEM AML, the GPAC module was taking a very long time to complete while processing. This performance issue has been resolved in NX Nastran 11.0.2.
7960192	11.0	For a SOL 401 Fourier modal analysis which includes grid points on the rotation axis, because of the MPC constraints in harmonic 1, the stiffness matrix for harmonic 1 is assembled in a different manner. This is due to the fact that for this case only, the symmetric and anti-symmetric part of the solution has a phase difference of 90 degress. This means that the stiffness matrix corresponding to the symmetric and anti-symmetric part are different in this case. For models with grid points on the rotation axis with a harmonic index greater than 1, and for models with no grid points on the rotation axis and all harmonic indices greater than 0 (including

		harmonic 1), the algorithm for stiffness matrix assembly is identical. This PR reported an error in which the softare incorrectly formed the same stiffness matrices for harmonic 1 for models with grid points on the axis and for models without grid points on the axis. This was causing wrong results for Eigen vectors and modal mass. This issue has been fixed in NX Nastran 11.0.2. In addition, for models with grid points on the rotation axis, all displacements (axial, radial and azimuthal) are now constrained to 0.0 for all harmonics greater than 1. This constraint was missing in NX Nastran 11.0.x. See "Fourier harmonic solution update" in the NX Nastran 11.0.2 Release Guide.
na	11.0	For a SOL 401 Fourier modal analysis, there was a bug related to writing the LAMA table to the OP2 file for Fourier modes when AF normalization was requested. For the AF normalization, the LAMA table is updated after the normalization. The updated table was getting correctly printed to the f06, but the OP2 was still receiving the older LAMA table. Now in NX Nastran 11.0.2, the updated LAMA table is written to the OP2 file.
7912597	1.0	An input file was used in a cold start with SOL103, then in a restart with SOL 111. The solution returned the following error when running NX Nastran 10: BIOMSG: ERROR 31 HAS OCCURRED IN ROUTINE GETSTR , FILE INDEX = 0. STATUS = 0 The solution returned the following error when running NX Nastran 11.0: ATTEMPT TO MULTIPLY INCOMPATIBLE MATRICES. The problem was that the required data block for the MEFFMASS calculations in the restart run were not properly saved in the initial run. When the data block was recreated in the restart run, it was of the wrong size, resulting in the error. The problem has been corrected in NX Nastran 11.0.2.
7885782	9.0	The customer runs a SOL 111 random response analysis using multiple input files using four subcases and the RMS von Mises stress is not calculated properly. If the contents of all four input files are combined into one input file with one subcase, the RMS von Mises stress seemed to be correct. The RMS von Mises results for multiple loadings has been fixed in the NX Nastran 11.0.2 release.

<table>
<tr>
<td>na</td>
<td>10.0</td>
<td>Incorrect sign of Coriolis terms for a rotordynamics rotating system analysis.

Before NX Nastran 11.0.0, a rotordynamics analysis in the fixed system used an incorrect sign for the Coriolis terms. Although, the rotating system used the correct sign.

In NX Nastran 11.0.0, the issue above was reported, and a change in the software occurred such that the sign of the Coriolis terms for the fixed system were corrected, although the sign of the Coriolis terms for the rotating system was mistakingly also changed, thus introducing a new problem for the rotating system.

Now in NX Nastran11.0.2, the Coriolis terms for the rotating system have been corrected. Now the Coriolis terms for both the fixed and the rotating systems are correct.

In summary,
<table>
<tr><th></th><th>Fixed</th><th>Rotating</th></tr>
<tr><td>NX Nastran10 and previous releases</td><td>sign flipped</td><td>sign correct</td></tr>
<tr><td>NX Nastran11.0.0 and 11.0.1</td><td>sign correct</td><td>sign flipped</td></tr>
<tr><td>NX Nastran11.0.2 +</td><td>sign correct</td><td>sign correct</td></tr>
</table>
</td>
</tr>
<tr>
<td>7743230</td>
<td>11.0</td>
<td>In SOL 101, thermal strain on the CQUAD4/CQUADR elements was not correct when a MAT2 bulk entry was used.

The problem was a result of an incorrect interpretation of the material coordinate system (theta). This problem is fixed in NX Nastran 11.0.2.</td>
</tr>
<tr>
<td>7907558</td>
<td>11.0</td>
<td>A model was run in SOL 101 which had shell composite elements (PCOMP) and offsets defined (ZOFFS), and the case control included the STRAIN, ELSTRN and THSTRN output requests. No thermal strain output occurred for this model, and as a result, the STRAIN output should have been equal to the ELSTRN output. However, it did not. If the offsets were removed, the STRAIN and ELSTRN results did match.

The problem was for the homogeneous elastic strains on CTRIA3/CQUAD4/CTRIA6/CQUAD8 when the offsets were defined. The composite ply strains were correct. The issue is fixed in NX Nastran 11.0.2.</td>
</tr>
<tr>
<td>7901064</td>
<td>5.0</td>
<td>A fatal error occured when including a superelement with the virtual mass fluid option.

This problem is fixed in NX Nastran 11.0.2. A DMAP alter was sent to allow the user to correct the bug in their local NX Nastran 11.0 run.</td>
</tr>
</table>

7894265	1.0	The output4 module using formatted option was writing an incorrect trailer for the final "end of data" record. The trailer indicates that the final record is 2 words but only 1 word is written. The OP4 trailer record has been corrected in the NX Nastran 11.0.2 release.
7880653	3.0	The relative displacement wasn't computed in SOL109 for grid points used on an RBE2 element. The absolute displacement was calculated on this grid point. These DOF will now have the correct relative displacement values beginning with the NX Nastran 11.0.2 release.
7625852	1.0	A software crash occured during a SOL 200 sensitivity analysis when an external structural damping matrix was used. The problem with this job was that there were several design variables which affect the same (non-damped) element. The issue is fixed in the NX Nastran 11.0.2 release.
na	11.0	A performance issue was found with the Math Kernel Library (MKL) in NX Nastran 11.0. The AVX2 version scgemm was tuned incorrectly. This performance issue only occured running on machines with AVX2 support (Haswell, Broadwell and Skylake). In NX Nastran 11.0.2, NX Nastran reverted back to MKL 11.2_initial to avoid the issue.

Siemens Industry Software

Headquarters
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 972 987 3000

Americas
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 314 264 8499

Europe
Stephenson House
Sir William Siemens Square
Frimley, Camberley
Surrey, GU16 8QD
+44 (0) 1276 413200

Asia-Pacific
Suites 4301-4302, 43/F
AIA Kowloon Tower, Landmark East
100 How Ming Street
Kwun Tong, Kowloon
Hong Kong
+852 2230 3308

About Siemens PLM Software

Siemens PLM Software, a business unit of the Siemens Industry Automation Division, is a leading global provider of product lifecycle management (PLM) software and services with 7 million licensed seats and 71,000 customers worldwide. Headquartered in Plano, Texas, Siemens PLM Software works collaboratively with companies to deliver open solutions that help them turn more ideas into successful products. For more information on Siemens PLM Software products and services, visit www.siemens.com/plm.