Introduction and Agenda
Agenda:

➢ Dynamics Analysis Enhancements
  • Frequency as a function of temperature for frequency dependent materials
  • PEM enhancements

➢ Nonlinear Enhancements
  • Geometric imperfection in SOL 400
  • Nonlinear buckling in SOL 400
  • New MATVE format

➢ Rotor-dynamics Enhancements

➢ Topology Optimization Enhancements

➢ Elements

➢ Fatigue Enhancements

➢ Numerical methods / HPC
  • Performance and efficiency improvements of the AVL Excite interface
  • CASI iterative solver support for inertia relief
  • GPU out of core for large problems

➢ Q & A
Frequency as a Function of Temperature for Frequency Dependent Materials
Frequency Dependent Materials: Frequency as a Function of Temperature

➢ Since v 2018, MSC Nastran allows its material specification to be frequency dependent.
➢ Also, in the specification of structural damping for anisotropic materials, the restriction that damping must be proportional to stiffness was removed and the damping coefficients can be a function of frequency.
➢ In v. 2020 these features are extended further to allow frequency itself to be a function of temperature.

➢ Benefits
  • Mechanical properties of constituent fiber and matrix materials often exhibit significant frequency-dependency sensitive to temperature
  • The MAT1F, MAT2F, MAT8F, MAT9F and PBUSHT, PFASTT entries referencing a TABLED5 entry provides a method to include the influence of local temperature variation on frequency dependent material properties
Frequency Dependent Materials: Frequency as a Function of Temperature

Usage

- Use TABLED5 to reference TABLEDi, i=1 to 4
- Solution Sequence SOL108, SOL111, or SO200, SOL400 with ANALYSIS=DFREQ or MFREQ
- In V2020, MAT1, MAT2, MAT8, and MAT9 material entries with associated MAT1F, MAT2F, MAT8F, and MAT9F entries may have frequency associated with temperature by allowing the above MATiF entries to point to a TABLED5 entry.

Example of TABLED5 use using MAT1

<table>
<thead>
<tr>
<th>ID</th>
<th>E</th>
<th>G</th>
<th>NU</th>
<th>GE</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAT1</td>
<td>33</td>
<td>7.2+10</td>
<td>.3</td>
<td>.02</td>
</tr>
<tr>
<td>MAT1F</td>
<td>33</td>
<td>110</td>
<td>111</td>
<td>112</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID</th>
<th>E</th>
<th>G</th>
<th>NU</th>
<th>GE</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLED5</td>
<td>200</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.</td>
<td>3</td>
<td>40.</td>
<td>4</td>
</tr>
</tbody>
</table>

Example of TABLED5 use using MAT1
## Frequency Dependent Materials: Frequency as a Function of Temperature

### Usage Example
- E is dependent only on frequency, therefore points to a TABLED1 entry.
- GE is both temperature and frequency dependent and therefore points to a TABLED5 entry

<table>
<thead>
<tr>
<th>Material ID</th>
<th>Frequency</th>
<th>Temperature</th>
<th>E Value</th>
<th>G Value</th>
<th>Nu Value</th>
<th>GE Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAT1</td>
<td>1</td>
<td>7.2 x 10^10</td>
<td>2.8 x 10^10</td>
<td>.3</td>
<td>2.22-5</td>
<td>0.02</td>
</tr>
<tr>
<td>MAT1F</td>
<td>1</td>
<td>110</td>
<td>111</td>
<td>112</td>
<td></td>
<td>200</td>
</tr>
<tr>
<td>TABLED1</td>
<td>110</td>
<td>10.</td>
<td>7.2 x 10^10</td>
<td>200.</td>
<td>7.1 x 10^10</td>
<td>6.9 x 10^10</td>
</tr>
<tr>
<td>TABLED1</td>
<td>111</td>
<td>10.</td>
<td>2.8 x 10^10</td>
<td>200.</td>
<td>2.7 x 10^10</td>
<td>2.6 x 10^10</td>
</tr>
<tr>
<td>TABLED1</td>
<td>111</td>
<td>10.</td>
<td>.3</td>
<td>200.</td>
<td>.3</td>
<td>.3</td>
</tr>
</tbody>
</table>

**Behavior**: GE has frequency as a function of temperature

### TABLED5

<table>
<thead>
<tr>
<th>Frequency</th>
<th>Temperature</th>
<th>G Value</th>
<th>Nu Value</th>
<th>GE Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>100.</td>
</tr>
<tr>
<td>10.</td>
<td>0.02</td>
<td>200.</td>
<td>0.03</td>
<td>300.</td>
</tr>
<tr>
<td>10.</td>
<td>0.025</td>
<td>200.</td>
<td>0.035</td>
<td>300.</td>
</tr>
<tr>
<td>10.</td>
<td>0.03</td>
<td>200.</td>
<td>0.04</td>
<td>300.</td>
</tr>
</tbody>
</table>

**Behavior**: TABLED1 units are x=frequency, y=material value

**Note**: TABLED5 input is numerical temperature – frequency table ID

- For an element with average temperature of 15.0 degrees the GE value will be selected from TABLED1 ID=3;
- For an element with average temperature of 30.0 degrees the GE value will be selected from TABLED1 ID=4;
- For an element with average temperature of 20.0 degrees the GE value will be selected from TABLED1 ID=3;
Frequency Dependent Materials: Frequency as a Function of Temperature

Limitations and Guidelines
- MATiF entries and MATTi entries that point to the same MATi entry are incompatible – a fatal message will be issued
- **Frequency as a function of temperature is a spatial feature** and TEMP, TEMPD entries are required to specify the spatial temperature of the model
- No advantage of Subcase case control structure reusing existing stiffness, mass, and loads as all are recomputed for any change in TEMP(INIT)
- Frequency as a function of temperature is a spatial feature and requires a double interpolation
  - For large models, analysis time can be significantly longer
Other Dynamics Enhancements

• Modal Damping in the MSC Nastran / AVL EXCITE Interface
• EXTSE Based Data Recovery for Adams MNF Interface
Modal Damping in the MSC Nastran / AVL EXCITE Interface

- Modal damping is added to the EXB file using the usual MSC Nastran mechanism, i.e. using SDAMP case control and TABDMP1 bulk data entries

```plaintext
SOL 103
CEND
$ Export EXB Flexbody
AVLEXB EXBBODY=YES
$ Case Control
$ Modal Damping, in EXB file: SDAMP case control
SDAMP = 200
$ Bulk Data
BEGIN BULK
$ Modal Damping in EXB file: TABDMP1 bulk data entry
TABDMP1  200   CRIT
         0.    0.00    300.   0.02  1000.   0.05  10000.   0.10
         1.E5   1.0    1.E6   1.0    ENDT
```
EXTSE Based Data Recovery for Adams MNF Interface

- To minimize the data storage and enable efficient data recovery in MSC Nastran, the MSC Nastran/Adams interface introduces the EXTSEOUT based data recovery in SOL 112.
- To use this method employ EXTSEOUT in SOL 103 MNF generation run:

```bash
$ Sample Nastran SOL 103 input file for Exporting MNF with EXTSEOUT
ASSIGN OUTPUT2='extse100.op2' UNIT=25 DELETE
SOL 103
CEND
...
$ Export MNF Flexbody
ADAMSMNF FLEXBODY=YES
$ EXTSEOUT feature is leveraged to minimize data storage requirements and enable $ efficient data recovery. To minimize the size of External Superelement .op2 the $ user should only request outputs for sets of physical quantities that of interest. $ For, e.g., displacement and velocities on surface nodes
EXTSEOUT(DMIGOP2=25, EXTID=100, ASMBULK)
...
```
EXTSE Based Data Recovery for Adams MNF Interface

• Following an Adams simulation, efficient EXTSE based data recovery in SOL 112 is achieved as follows:

```plaintext
$ Sample Nastran SOL 112 input file for conducting EXTSE based Data Recovery
$ after doing ADAMS simulation.
$ Assign the Superelement databases that have been stored on the .op2 files
ASSIGN INPUTT2='extse100.op2' UNIT=25
$ Assign the ADAMS modal coordinates
ASSIGN INPUTT2='crrod_coords.mdf' UNIT=31
...
DLOAD = 31
...
SOL 112
CEND
...
$ Following PARAM entries are also required for conducting EXTSE based Data Recovery
PARAM, ADMEXTU, 25
PARAM, ADMPOST, 1
...
$ These files are used to attach the external superlement
INCLUDE 'crrod.asm'
INCLUDE 'crrod.pch'
```
Porous-Elastic Material (PEM) Enhancements

- TRMC interior grid data recovery
- TRMC coupling definition with EID
- PEM job restart
- PEM support in SOL 108
PEM Enhancements: TRMC Data Recovery

➢ **Introduction**
  - TRMC data recovery was only available for surface nodes previously
  - Expanded to include interior nodes in MSC Nastran v2020

• **Benefits**
  - More complete data recovery is now possible
  - Easier control of desired data recovery
    - New Case Control entries for trim component data recovery
      - TDISPLACEMENT
      - TVELOCITY
      - TACCELERATION
    - Define a separate data recovery set for each trim component
PEM Enhancements: TRMC Data Recovery

➢ **Usage**
  - New Case Control commands, TDISP/TVELO/TACCE, for TRMC data recovery
    - Each TRMC can have an individual SET definition – new set definition type
    - SET id = trmcid1/set1, trmcid2/set2, …

➢ **Example**
  - TDISP = 17
  - SET 17 = 3/103, 5/0, 12/all

➢ **Notes**
  - TDISP=all can generate large amount of output data should be used with caution
  - Old TRMC data recovery request (via ‘TRMC=‘ subcommand of DISP/VELP/ACCE) is no longer supported and will cause UFM if used
PEM Enhancements: TRMC Coupling Definition with Element ID

➢ Introduction
  • Viewing a trim component coupling definition based on just grid IDs can be difficult to visualize
  • New PLTSURF elements can describe the coupling surface mesh

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PLTSURF</td>
<td>EID</td>
<td>GID1</td>
<td>GID2</td>
<td>GID3</td>
<td>GID4</td>
<td>GID5</td>
<td>GID6</td>
<td>GID7</td>
<td>GID8</td>
</tr>
</tbody>
</table>

➢ Benefits
  • Easier to view the coupling definition using PLTSURF elements
  • Easier to spot any potential coupling surface definition errors

Grids ➔ Elements
PEM Enhancements: TRMC Coupling Definition with Element ID

➢ **Usage**
  - Connects 3, 4, 6 or 8 grids mirroring TRIA3, QUAD4, TRIA6 and QUAD8 surface elements
  - PLTSURF ID can be used to describe of trim component surface coupling on ACPEMCP
  - PLTSURF ID must be referenced on SET3 with ELEM descriptor – element ID set definition
    - Do not use SET1 (grid ID set definition) – will result in incorrect coupling or fatal error
  - PLTSURF elements are similar to PLOTELs – nonstructural, visualization only elements
  - To create PLTSURF
    - Surface wrapper elements can be created over TRMC using TRIA3/QUAD4/TRIA6/QUAD8
    - Change TRIA3/QUAD4/TRIA6/QUAD8 to PLTSURF and remove PID field
    - For QUAD8 with mid-side nodes, further editing is needed
PEM Enhancements: TRMC Coupling Definition with Element ID

• Results comparison of GID vs. EID based coupling definitions
PEM Enhancements: PEM Restart

➢ Introduction
  • Restarts from cold start databases are supported for MSC Nastran PEM analyses

➢ Benefits
  • Reduced run time for loading, forcing frequency and data recovery changes.

➢ Notes
  • For efficient restarts, the following should remain unchanged:
    • TRIMGRP – no addition or deletion of trim components
    • Individual trim components including coupling nodes
    • TDISP/TVELO/TACCE case control entries
    • Coupling nodes of structural and/or cavity
  • Restart job with DMP>1 is not supported if cold start uses DMP>1
PEM Enhancements: PEM Restart

➢ **Usage**
- Use ‘scr=no’ during submittal for cold start
- For restart, insert following line at the top of the input deck
  - ASSIGN CSTART='cold_start_job_name.MASTER'
  - RESTART logical=CSTART
- Use ‘scr=yes’ for restart job submittal
- DMP support

<table>
<thead>
<tr>
<th>Cold Start</th>
<th>Restart</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMP=1</td>
<td>DMP=1</td>
<td>OK</td>
</tr>
<tr>
<td>DMP=1</td>
<td>DMP&gt;1</td>
<td>OK</td>
</tr>
<tr>
<td>DMP&gt;1</td>
<td>DMP=1 (attach master node DB or ‘t0’)</td>
<td>OK</td>
</tr>
<tr>
<td>DMP&gt;1</td>
<td>DMP&gt;1</td>
<td>Not supported</td>
</tr>
</tbody>
</table>
PEM Enhancements: PEM Restart

- **Performance**: normalized performance for a subcase / load case restart
PEM Enhancements: PEM support in SOL 108

➢ Introduction
  • PEM support is extended to SOL 108

➢ Benefits
  • Extends PEM analysis method to include a direct as well as modal approach
  • Method to verify SOL 111 results

➢ Notes
  • SOL 108 PEM analysis will be always much slower than SOL 111 PEM analysis
    • Much bigger problem size - uses physical vs. modal coordinates
PEM Enhancements: PEM support of SOL 108

Usage
• Steps to turn SOL 111 PEM deck to SOL 108 PEM deck
  • change SOL 111 to SOL 108
  • remove or comment out DOMAINSOLVER command, if any
  • remove or comment out METHOD case control commands
  • Recommend significantly reducing number of forcing frequencies to <5% of original
  • Change master frequencies of each TRMC to match forcing frequencies to improve performance and reduce disk space demand
PEM Enhancements: PEM support of SOL 108

- Performance

NASCAR normalized performance

<table>
<thead>
<tr>
<th>0</th>
<th>0.5</th>
<th>1</th>
<th>1.5</th>
<th>2</th>
<th>2.5</th>
<th>3</th>
<th>3.5</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOL 111 400 Freqs</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>SOL 108 1 Freq</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

NASCAR normalized performance
Nonlinear Enhancements
Geometric Imperfection in SOL 400

- Introduction and benefit
- Input user Interface
- Output
- Examples
Geometry Imperfection in SOL 400

➢ Introduction
  • Most FEM analyses are based on “perfect” geometry
  • Geometric imperfection is unavoidable in reality due to manufacturing processes
    • Imperfection may have significant effects on unstable structures
    • Post buckling analysis with geometric imperfection is important

➢ Benefits
  • This capability provides an easy way to take geometric imperfection effects into account
  • V2020 also includes nonlinear buckling capability, which can be used with geometric imperfection capability
Geometry Imperfection in SOL 400

➢ Approach

1. Pre-run, analyze model with “perfect” geometry in static, buckling, or normal modes solution and save mode shapes or displacements
   • Solutions of SOL 101, 103, 105 or SOL 400 (ANALYSIS=NLSTATIC, STATIC, MODES, BUCKL or NLTRAN)
   • Output to OP2 or HDF5 format

2. Scale and/or combine the saved displacements/mode shapes or user supplied file, add to the original geometry to form the “imperfect” geometry
   • Import pre-run imperfection data in OP2 or HDF5 format
   • Import user supplied IMPF imperfection text file

3. Analyze model with “imperfect” geometry in any analysis type of SOL 400
Geometry Imperfection in SOL 400

➢ Usage

• FMS ASSIGN statement to import pre-run imperfection data
  • ASSIGN INPUTT2=<a.op2> unit=32 (Existing capability)
  • ASSIGN HDF5IN=<a.h5> unit=33 (New capability)
  • ASSIGN IMPFIN=<a.impf> unit=41 (New capability)

• Case Control
  • IMPERFECT=impfid (Above subcase level, to activate imperfection)
  • OIMPERFECT (Above subcase level, for imperfection output)

• Bulk Data
  • IMPGEOM (Required to define single imperfection case)
  • IMPCASE (Optional for multiple imperfection cases)
Geometry Imperfection in SOL 400

➢ Usage: case control commands
  • IMPERFECT=n
    • n is an identification number of IMPGEOM or IMPCASE
    • Must be above all subcases

  • OIMPERFECT( [PRINT, PUNCH, PLOT], GEOM)= \{ ALL, n, NONE \}
    • Output request of imperfection shape. GEOM option is to output GRID entries with imperfection to punch file.
Geometry Imperfection in SOL 400

➢ IMPGEOM bulk data entries

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMPGEOM</td>
<td>ID</td>
<td>SETID</td>
<td>SCALE</td>
<td>UNIT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUBCASE1</td>
<td>STEP1</td>
<td>MODE1</td>
<td>SETID1</td>
<td>S1</td>
<td>UNIT1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUBCASE2</td>
<td>STEP2</td>
<td>MODE2</td>
<td>SETID2</td>
<td>S2</td>
<td>UNIT2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>etc</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- ID is referred by IMPCASE bulk data entry or IMPERFECT case control command
- SETID: default value of SETIDi
- SCALE default value of Si
- UNIT default value of UNITi
- SUBCASEi specifies which subcase solution is going to be used, ignored for text file input
- STEP ID is for SOL 400 ONLY and specifies which step solution is to be used, ignored for text file input (SOL 400 ONLY)
- MODEi specifies which mode solution to be used, ignored for text file input
- SETIDi is a grid point set (SET1 or GRID type of SET3), if it is non-zero, only the grid ids in the set have geometric imperfection effect
- Si is scale factor on all shapes specified by Si/Ci (default=0.0)
- UNITi is given when suing ASSIGN to specify the input file of op2, hdf5, or the text file. The default is the maximum value of units appeared in all ASSIGN statements
Geometry Imperfection in SOL 400

IMPCASE

➢ IMPCASE bulk data entries

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMPCASE</td>
<td>ID</td>
<td>IMPFID1</td>
<td>IMPFID2</td>
<td>etc</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Entry to collect IMPGEOM entries and referred to by IMPERFECT case control
  - ID: imperfect id referred by IMPERFECT case control
  - IMPFIDi: the ID of IMPGEOM entries, “THRU” is allowed.
Geometry Imperfection in SOL 400

IMPF File

➢ Format of impf file
  • A csv like text file
    • Disp or Geom (optional, default is DISP)
    • For DISP, coordinates x, y, z are assumed in output coordinate system (MSC Nastran global system)
    • For GEOM, coordinates x, y, z are assumed in input coordinate systems (CP field of Grid)
    • Delimiters can be a comma, spaces or a tab
    • GID is required, default is zero for other fields
    • A line starting with “$” or “#” is a comment line, a blank line is ignored also
    • The first line may indicate that the file is new geometry or displacement
  • Example
    # GEOM (geometry, not displacement)
    1, 1.e-3, 2.1e-5, 3.1e-6
    2, 1.e-3, 1.2e-5
    3, 1.0+1, 3.2-6, 2.0
    $ Delimited by spaces
    5 1.2-4 2.4e-5
Example Cylinder Buckling with Imperfection
Cylinder Buckling with Geometric Imperfection

➢ Pre-run TPL File: buckcy20r.dat

➢ Pre-run SOL 105 input deck (buckcy20r.dat)
  • ASSIGN HDF5='buckcy20r.h5'
  • DISPLACEMENT=ALL (Case control to request displacements and eigenvectors)
  • mdlprm,hdf5,1 (Bulk data to request HDF5 output)
Cylinder Buckling with Geometric Imperfection

- Imperfection Run TPL File: impf_buckcy20r.dat

```
ASSIGN HDF5IN='buckcy20r.h5' unit=32
SOL 400
CEND
IMPERFECT=11
OIMPERFECT(GEOM)=ALL
SUBCASE 1
   analysis=static
   ...
SUBCASE 2
   analysis=buckl
   statsub=1
BEGIN BULK
IMPGEOM, 12,
   , 2, , 1, ,.01, , 32 $ SUBCASE=2, MODE=1 and SCALE=0.01, unit is 32, buckcy20r.h5
IMPGEOM, 21,
   , 2, , 2, ,,009, , 32 $ SUBCASE=2, MODE=2 and SCALE=0.009, unit is 32, buckcy20r.h5
IMPCASE, 11,12,21
```
Cylinder Buckling with Geometric Imperfection

- F06 Output: Imperfection Vector and Titles with Imperfect ID

<table>
<thead>
<tr>
<th>POINT ID.</th>
<th>TYPE</th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>G</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>G</td>
<td>3.090169E-03</td>
<td>-5.486675E-13</td>
<td>-4.635254E-05</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>21</td>
<td>G</td>
<td>3.078410E-03</td>
<td>2.693260E-04</td>
<td>-4.635254E-05</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**EIGENVALUE CALCULATION**

**EIGENVALUE** = 2.485866E-01

- **REAL EIGENVECTOR NO. 1**
  - **IMPERFECT 12**
  - **SUBCASE 2**

<table>
<thead>
<tr>
<th>POINT ID.</th>
<th>TYPE</th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>G</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>-2.525371E-04</td>
<td>-1.363822E-04</td>
<td>-2.139241E-03</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>G</td>
<td>-4.595786E-06</td>
<td>4.908956E-04</td>
<td>7.099237E-06</td>
<td>2.070287E-04</td>
<td>1.353015E-04</td>
<td>-2.098460E-03</td>
</tr>
</tbody>
</table>
Cylinder Buckling with Geometric Imperfection

➢ Punch file output:
  • New grid locations with imperfection
  • Enabled by OIMPERFECT(GEOM)=n
  • Multiple imperfection cases are in one file with begin and end markers
  • Small or large field is automatically determined to keep precision

```plaintext
$ GRID for IMPERFECT ID  12
GRID  1  1  10.  0.  0.  0  0  0
......
GRID* 20  1  10.00901693808-5.4867569470-13
* .9999536474582010  0
GRID* 21  1  10.003078410353 5.0002693260738
* .9999536474587440  0
......
GRID* 1595  1  10.  -5.
* 20.000000000000  0
GRID* 5000  1  0.  0.
* 20.000000000000  0
$ END of GRID for IMPERFECT ID  12
......
$ GRID for IMPERFECT ID  21
GRID  1  1  10.  0.  0.  0  0  0
......
GRID* 20  1  9.998592090354363.34250898193-12
* 1.000021118645860  0
GRID* 21  1  9.998597447938374.99987729259606
* 1.000021118644740  0
......
GRID* 1595  1  10.  -5.
* 20.00002124945770  0
GRID* 5000  1  0.  0.
* 20.00002124945930  0
$ END of GRID for IMPERFECT ID  21
```

Cylinder Buckling with Geometric Imperfection

- Output in NH5RDB (HDF5 result)
  - A new column IMPFID is added to RESULT/DOMAINS dataset imperfect id in IMPGEOM
  - A new dataset IMPERFECT is added to RESULT/NODAL group
  - The dataset to express imperfection input as displacement type of output
Cylinder Buckling with Geometric Imperfection
Patran Postprocessing for Imperfection: Imperfect ID and Imperfection Shape

IMPF12: imperfect 12
IMPF21: imperfect 21
Cylinder Buckling with Geometric Imperfection

Patran Postprocessing for Imperfection: Imperfect ID and Imperfection Shape

Factor = 0.248587, less than "perfect" geometry (0.342617)
Cylinder buckling with geometric imperfection

Effect of Imperfection: Buckling Load Factors

- Buckling load factor of the pre-run is 0.343362, table shows the factor of each imperfection case

<table>
<thead>
<tr>
<th>Imperfection case</th>
<th>Buckling load factor</th>
<th>Factor to the initial run</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>0.2486</td>
<td>0.7294</td>
</tr>
<tr>
<td>21</td>
<td>0.2567</td>
<td>0.7476</td>
</tr>
</tbody>
</table>
Nonlinear Buckling Analysis

- Introduction
- Benefits
- User Interface
- Usage
- Guidelines and Limitations
Nonlinear Buckling Analysis

- Buckling analysis used to determine the critical load a structure
- Since v2014, SOL 400 can perform buckling analysis in linear domain only, similar to SOL 105
- In v2020, SOL 400 can now consider material and geometric nonlinearities in buckling analysis
  - User may request nonlinear buckling analysis in any STEP or SUBCASE at
    - Last converged load increment
    - All converged load increments
    - The equation to be solved is:

\[
[K_n + \lambda \Delta K] \{\Phi\} = 0
\]

where
- \(K_n\) and \(K_{n-1}\) are the stiffness matrices evaluated at the known solution points in the vicinity of the instability
- \(\Delta K = K_n - K_{n-1}\)
- \(\lambda\) is the eigenvalue
- \(\Phi\) is the eigenvector matrix
After the eigenvalues are computed then the critical buckling factor, $\alpha$, and the critical buckling load, $P_{cr}$ are computed where:

$$P_{cr} = P_n + \alpha \Delta P$$

where,

$P_n$ is the total applied load at the point of estimating buckling behavior

$$\Delta P = P_n - P_{n-1}$$

where,

$P_{n-1}$ is the applied load of the previous increment before $P_n$
Nonlinear Buckling Analysis

Introduction (cont.)

• $\alpha$ is computed accordingly:

$$\alpha = \frac{\lambda \{\Delta U\}^T \left[ K_n + \frac{1}{2} \lambda \Delta K \right] \Delta U}{\{\Delta U\}^T \{\Delta P\}}$$

Where:

• $\{\Delta U\} = \{U_n\} - \{U_{n-1}\}$
• $U_n$ are the critical displacements at the point of estimating buckling behavior
• $U_{n-1}$ are the displacements just prior to the point of estimating buckling behavior
• $K_n, \lambda, \Delta K, \text{ and } \Delta P$ are described in prior slides.
Nonlinear Buckling Analysis

Benefits:

• Large structures like an airplane, automobile, etc. with many parts and different materials can no longer be assumed to behave linearly under all loading conditions

• Contact, geometric and material nonlinearities must be considered, and a nonlinear buckling analysis may provide more accurate results than linear buckling

• V2020 also includes imperfection analysis and when combined with nonlinear buckling analysis gives the engineer a powerful tool to analyze structures which are more likely to buckle at a lower load if not built perfectly to specifications
Nonlinear Buckling Analysis

Usage:

• A new Case Control command called NLBUCK requests buckling analysis:

  SUBCASE 1
  STEP 1
  LOAD=10
  NLSTEP=10
  NLBUCK
  METHOD=30

• NLBUCK will have three options:
  • NLLOAD
  • NLBUCK=END
    • At the end of each step an eigenvalue projection is made to predict the buckled load (default)
  • NLBUCK=ALL
    • An eigenvalue projection after each converged load increment within the step in which it is defined

• A NLSTEP Case Control command must be specified in the same STEP as the NLBUCK command.
• If NLPARM is specified, then a fatal message will be issued.
Nonlinear Buckling Analysis

➢ Usage (cont.):

- There are three methods of eigenvalue extraction available for nonlinear buckling – Lanczos (EIGRL or EIGR entry with METHOD=LAN), enhanced inverse power method (EIGB entry with METHOD=SINV), and complex (EIGC entry) for unsymmetric stiffness due to follower stiffness.
- If no METHOD or CMETHOD command is specified, then the program will automatically attempt to compute two modes (ND=2) with an unspecified values for the eigenvalue range (F1 and F2) using the Lanczos method.
- The Lanczos method is recommended in most cases especially in finding the lowest mode.
- If no modes can be found with no eigenvalue range was specified, then it is highly recommended that a range (L1 and L2 on EIGB, F1 and F2 on EIGR, and V1 and V2 on EIGRL) be specified.
- If higher modes are desired, then the enhanced inverse power method is recommended with a narrow eigenvalue range specified for L1 and L2 on the EIGB entry.
- If a METHOD command is specified but the stiffness is unsymmetric then User Warning Message 9430 will be issued.
Nonlinear Buckling Analysis

➤ Usage: Flat Plate Example

- Cantilevered flat plate with 80 shell elements
- Load applied via RBE2 on free end
- Analyzed with:
  - Traditional and advanced nonlinear elements
  - Lanczos and enhanced inverse power eigenvalue extraction methods with and without a specified range
  - Geometric nonlinearity (PARAM, LGDISP, 1)
  - RIGID=LINEAR and RIGID=LAGRANGE
  - Various values for NINC on NLSTEP entry

First and Second Mode
Nonlinear Buckling Analysis

Flat Plate Example – f06 Output

1. Eigenvalue summary with LOAD STEP and ALPHA:

```
LOAD STEP = 1.00000E+00

REAL EIGENVALUES
MODE  EXTRUCTION  EIGENVALUE  RADIAN  CYCLES  GENERALIZED  GENERALIZED
NO.   ORDER
1     1          9.472063E+00  3.077672E+00  4.898267E-01  1.401756E-02  1.237754E-01
2     2          9.492261E+00  3.080951E+00  4.903486E-01  1.399872E-02  1.238056E-01
3     3          4.037811E+02  2.009430E+01  3.198108E+00  3.068562E-02  1.239027E-01
4     4          4.081096E+02  2.020172E+01  3.215204E+00  3.079828E-02  1.250458E-01
5     5          1.170210E+03  3.434618E+01  5.465104E+00  8.088118E-02  9.432496E-01
6     6          1.213184E+03  3.483046E+01  5.543464E+00  8.143678E-02  9.879609E-01
7     7          2.294178E+03  4.789823E+01  7.623482E+00  1.668264E-01  3.685575E-02
8     8          2.419752E+03  4.919137E+01  7.829505E+00  1.485164E-01  3.937866E-02
9     9          3.676824E+03  6.056700E+01  9.653906E+00  2.232124E-01  8.214983E-02
10   10         3.919417E+03  6.324956E+01  1.009511E+01  2.278202E-01  9.199538E-02
```

Critical buckling factor

2. Standard buckling eigenvector output, if requested, with LOAD STEP and STEP labeling

```
EIGENVALUE = 3.034956E+00

REAL EIGENVECTOR NO. 1
POINT ID.  TYPE
1     1.089747E-12 -1.25208E-02  -0.036925E-03  -0.131130E-05  -0.131093E-05  -0.131312E-05  -1.729182E-15  3.168215E-16
2     1.089747E-12 -1.25208E-02  -0.036925E-03  -0.131130E-05  -0.131093E-05  -0.131312E-05  -1.729182E-15  3.168215E-16
3     1.089747E-12 -1.25208E-02  -0.036925E-03  -0.131130E-05  -0.131093E-05  -0.131312E-05  -1.729182E-15  3.168215E-16
4     1.089747E-12 -1.25208E-02  -0.036925E-03  -0.131130E-05  -0.131093E-05  -0.131312E-05  -1.729182E-15  3.168215E-16
5     1.089747E-12 -1.25208E-02  -0.036925E-03  -0.131130E-05  -0.131093E-05  -0.131312E-05  -1.729182E-15  3.168215E-16
6     1.089747E-12 -1.25208E-02  -0.036925E-03  -0.131130E-05  -0.131093E-05  -0.131312E-05  -1.729182E-15  3.168215E-16
```

Critical buckling factor
Nonlinear Buckling Analysis

Flat Plate – f06 Output (cont.)

3. Critical displacements and loads, if DISPLACEMENT and OLOAD commands specified

4. Standard nonlinear static output, if requested:
Nonlinear Buckling Analysis
Flat Plate Example Results with RIGID=LAGRANGE and Lanczos

<table>
<thead>
<tr>
<th>NINC</th>
<th>$P_n$</th>
<th>$\Delta P$</th>
<th>Traditional elements</th>
<th>Advanced elements</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Range 0.01 - 100.</td>
<td>No range</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$\alpha$ $P_{cr}$</td>
<td>$\alpha$ $P_{cr}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>5</td>
<td>0.125</td>
<td>9.4721 6.1840</td>
<td>9.4721 6.1840</td>
</tr>
<tr>
<td>20</td>
<td>5</td>
<td>0.25</td>
<td>4.7361 6.1840</td>
<td>4.7361 6.1840</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>0.5</td>
<td>2.3682 6.1841</td>
<td>2.3682 6.1841</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>1</td>
<td>1.1842 6.1842</td>
<td>1.1842 6.1842</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>2.5</td>
<td>0.4738 6.1845</td>
<td>0.4738 6.1845</td>
</tr>
</tbody>
</table>
Example Imperfection with nonlinear buckling

tpl/imperf/imp_s buck12a_nlb.dat
Imperfection with nonlinear buckling

Patran shows LAMDA and critical load/displacement

assign INPUTT2='sbuckl2a_2.op2' unit=32
IMPERFECT=11
OIMPERF(GEOM)=ALL
SUBCASE 1
  step 1
  ...
  step 2
    anal=nlst
    NLstep=10
    NLBUCK = ALL
BEGIN BULK
  impcase, 11, 1, 2
  impgeom, 1, , 1, 1, 1, ,1., ,32
  impgeom, 2, , 1, 1, 1, ,2., ,32

![Patran screenshot showing LAMDA and critical load/displacement](image-url)
Nonlinear Buckling Analysis

Guidelines and Limitations:

- NLPARM Case Control command is not permitted in a nonlinear buckling step

- NLSTEP must be specified in the nonlinear buckling step
  - KMETHOD=PFNT (default) is strongly recommended along with NO=1 for FIXED time stepping or INTOUT=YES for ADAPT time stepping
  - If KMETHOD=ITER then KSTEP=1 is strongly recommended

- It is strongly recommended that PARAM,LGDISP,1 is specified

- RIGID=LAGRANGE or LGELIM with advanced nonlinear elements is not supported

- It is strongly recommended to specify an eigenvalue range on the EIGR, EIGRL, EIGB, and EIGC Bulk Data entries, but with NLBUCK=ALL it may be difficult to define a range for all load increments

- Node-to-segment is not recommended except in the case of permanent glued contact
New MATVE Format
New Format of MATVE

➢ When modelling viscoelastic material in SOL 400, MATVE allows Prony series up to 5 terms.

\[ g_r(t) = 1 - \sum_{i=1}^{N} w_i (1 - e^{-t/\tau_i}) \]

Existing MATVE entry
for Model types **ISO, MOONEY, OGDEN** and **FOAM**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATVE</td>
<td>MID</td>
<td>Model</td>
<td>Alphas</td>
<td>Alpha</td>
<td>( Wd1 )</td>
<td>( Td1 )</td>
<td>( Wv1 )</td>
<td>( Tv1 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( Wd2 )</td>
<td>( Td2 )</td>
<td>( Wd3 )</td>
<td>( Td3 )</td>
<td>( Wd4 )</td>
<td>( Td4 )</td>
<td>( Wd5 )</td>
<td>( Td5 )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( Wv2 )</td>
<td>( Tv2 )</td>
<td>( Wv3 )</td>
<td>( Tv3 )</td>
<td>( Wv4 )</td>
<td>( Tv4 )</td>
<td>( Wv5 )</td>
<td>( Tv5 )</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

for Model type **ORTHO**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATVE</td>
<td>MID</td>
<td>Ortho</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>( Td1 )</td>
<td>( Exx1 )</td>
<td>( Eyy1 )</td>
<td>Ezz1</td>
<td>Vyx1</td>
<td>Vyz1</td>
<td>Vzx1</td>
<td>Vxy1</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>( Gxy2 )</td>
<td>( Gyz2 )</td>
<td>( Gzx2 )</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>( Gyz3 )</td>
<td>( Gzy3 )</td>
<td>( Gzx3 )</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>( Td3 )</td>
<td>( Exx3 )</td>
<td>( Eyy3 )</td>
<td>Ezz3</td>
<td>Vyx3</td>
<td>Vyz3</td>
<td>Vzx3</td>
<td>Vxy3</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>( Gxy4 )</td>
<td>( Gyz4 )</td>
<td>( Gzx4 )</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>( Gyz5 )</td>
<td>( Gzy5 )</td>
<td>( Gzx5 )</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>
New Format of MATVE

➢ Usage and Benefits
  • Extend Prony series to indefinite terms, to remove the previous limitation of max 5
  • User can input weighting factor and relaxation time pairs as many as desired
  • More accurate viscoelastic material model can be simulated

New Format

Model = ISO1, MOONEY1, OGDEN1, FOAM1  Model = ORTHO

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATVE</td>
<td>MID</td>
<td>Model</td>
<td>Alphas</td>
<td>Alphal</td>
<td></td>
</tr>
<tr>
<td>Wd1</td>
<td>Td1</td>
<td>Wv1</td>
<td>Tv1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wd2</td>
<td>Td2</td>
<td>Wv2</td>
<td>Tv2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wd3</td>
<td>Td3</td>
<td>Wv3</td>
<td>Tv3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wd4</td>
<td>Td4</td>
<td>Wv4</td>
<td>Tv4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wd5</td>
<td>Td5</td>
<td>Wv5</td>
<td>Tv5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wd6</td>
<td>Td6</td>
<td>Wv6</td>
<td>Tv6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATVE</td>
<td>MID</td>
<td>ORTHO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Td1</td>
<td>Exx1</td>
<td>Eyy1</td>
<td>Ez1</td>
<td>Vxy1</td>
<td>Vy1</td>
<td>Vz1</td>
<td>Vx1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gxy1</td>
<td>Gyz1</td>
<td>Gox1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Td2</td>
<td>Exx2</td>
<td>Eyy2</td>
<td>Ez2</td>
<td>Vxy2</td>
<td>Vy2</td>
<td>Vz2</td>
<td>Vx2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gxy2</td>
<td>Gyz2</td>
<td>Gox2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Td3</td>
<td>Exx3</td>
<td>Eyy3</td>
<td>Ez3</td>
<td>Vxy3</td>
<td>Vy3</td>
<td>Vz3</td>
<td>Vx3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gxy3</td>
<td>Gyz3</td>
<td>Gox3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Td4</td>
<td>Exx4</td>
<td>Eyy4</td>
<td>Ez4</td>
<td>Vxy4</td>
<td>Vy4</td>
<td>Vz4</td>
<td>Vx4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gxy4</td>
<td>Gyz4</td>
<td>Gox4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Td5</td>
<td>Exx5</td>
<td>Eyy5</td>
<td>Ez5</td>
<td>Vxy5</td>
<td>Vy5</td>
<td>Vz5</td>
<td>Vx5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gxy5</td>
<td>Gyz5</td>
<td>Gox5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Td6</td>
<td>Exx6</td>
<td>Eyy6</td>
<td>Ez6</td>
<td>Vxy6</td>
<td>Vy6</td>
<td>Vz6</td>
<td>Vx6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gxy6</td>
<td>Gyz6</td>
<td>Gox6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

hexagonmi.com | mscsoftware.com
**New Format of MATVE**

Example: tpl/matveqa/matve1.dat

![Initial Approach velocity = 10 mm/s](image)

**MATVE1 1**

<table>
<thead>
<tr>
<th>MOONEY1</th>
<th>Nastran</th>
<th>Marc</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.677037e-1 1.0e-5 1.0e-3 1.0e-5</td>
<td>7.426267e-5 1.0e-4 1.0e-6</td>
<td>6.863759e-2 3.593814e-2 1.0e-4 1.0e-6</td>
</tr>
<tr>
<td>9.329138e-2 5.994843e-4 1.0e-4 1.0e-6</td>
<td>4.641589e-3 1.0e-4 1.0e-6</td>
<td>4.043750e-2 2.782559e-1 1.0e-4 1.0e-6</td>
</tr>
<tr>
<td>7.852159e-2 4.641589e-3 1.0e-4 1.0e-6</td>
<td>6.863759e-2 3.593814e-2 1.0e-4 1.0e-6</td>
<td>5.405654e-2 2.154435e+0 1.0e-4 1.0e-6</td>
</tr>
<tr>
<td>6.043750e-2 2.782559e-1 1.0e-4 1.0e-6</td>
<td>5.028579e-2 1.668101e+1 1.0e-4 1.0e-6</td>
<td>4.418413e-2 1.291550e+2 1.0e-4 1.0e-6</td>
</tr>
<tr>
<td>4.418413e-2 1.291550e+2 1.0e-4 1.0e-6</td>
<td>5.221412e-2 1.000000e+3 1.0e-4 1.0e-6</td>
<td>5.221412e-2 1.000000e+3 1.0e-4 1.0e-6</td>
</tr>
</tbody>
</table>

**VM-STRESS at NODE#96**

Nastran Max VM Stress= 7.14 MPa

Marc Max VM Stress= 7.15 MPa
Rotordynamics Enhancements
Rotordynamics: Enhancements to Bearing Modeling

➢ There is increasing interest in rotordynamics, and bearing modeling, from the OEMs and this enhancement serves to improved bearing and damper representation as function of speed
   • Achieved through user-defined subroutines (UDS) to accurately represent linear and nonlinear behavior using customized routines

➢ Linear 2D bush element (CBUSH2D) now allows specification of 2x2 K, B, M terms
   • For nominal conditions using PBUSH2D
   • For frequency dependent conditions using PBSH2DT and/or ELEMUDS
   • Available in SOLs 107, 108, 110, 111, 128 and 400

➢ Nonlinear squeeze film damper (NLRSFD) element can now take acceleration terms into the new user-defined subroutines
   • Available in SOL 128 and SOL 400
Rotordynamics: Enhancements to Bearing Modeling

- Benefits
  - Traditionally, 2D bush elements are modeled to represent lateral, vertical and cross-coupled K and B properties of a bearing as 2x2 terms
  - Bearing properties vary as a function of rotor speed – accuracy depends on modeling bearing properties at all operating scenarios
  - New capability allows bearing properties (2x2 terms) to be represented as a function of speed for linear and nonlinear solutions
  - Squeeze film damper representation is also enhanced to accurately capture the properties as a function of rotor speed
  - Externally supported routine can accept displacement, velocity and acceleration at connected grids
  - New UDS enable users to represent both linear and nonlinear dampers as a function of speed
  - UDS also support external bearing codes
Rotordynamics Enhancements to Bearing Modeling

**Usage:** CBUSH2DA service within ELEMUDS card

- Enabled through following definition as first line in main run file,
  - `CONNECT SERVICE TESTF 'SCA.MDSolver.Obj.Uds.Elements.cbush2da'`
- ELEMUDS card should have TESTF & cbush2da defined in 4<sup>th</sup> & 5<sup>th</sup> field
- 7<sup>th</sup> field specifies speed used for nominal bearing properties
- Supported in SOL107, 108, 110, 111, 128, 400

<table>
<thead>
<tr>
<th>ELEMUDS</th>
<th>PBUSH2D</th>
<th>TESTF</th>
<th>cbush2da</th>
<th>FREQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>1001</td>
<td>1002</td>
<td>1003</td>
<td>2001</td>
</tr>
<tr>
<td>char</td>
<td>freq</td>
<td>K11</td>
<td>K12</td>
<td>K21</td>
</tr>
<tr>
<td>real</td>
<td>10.</td>
<td>4.445+8</td>
<td>6.00+6</td>
<td>5.00+5</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>0.15</td>
<td>0.15</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>30.0</td>
<td>10.0</td>
<td>10.0</td>
<td>60.0</td>
</tr>
<tr>
<td></td>
<td>50.</td>
<td>5.455+8</td>
<td>7.00+6</td>
<td>7.00+6</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>0.23</td>
<td>0.28</td>
<td>0.3</td>
</tr>
<tr>
<td></td>
<td>85.0</td>
<td>35.0</td>
<td>55.0</td>
<td>65.0</td>
</tr>
</tbody>
</table>

**Output:** SOL 110 f06 summary

- Supported in SOL107, 108, 110, 111, 128, 400

---

**Format:**
- **ELEMUDS**
- **PBUSH2D**
- **TESTF**
- **cbush2da**
- **FREQ**

---

**Usage:**
- enabled through following definition as first line in main run file,
- `CONNECT SERVICE TESTF 'SCA.MDSolver.Obj.Uds.Elements.cbush2da'`
- ELEMUDS card should have TESTF & cbush2da defined in 4<sup>th</sup> & 5<sup>th</sup> field
- 7<sup>th</sup> field specifies speed used for nominal bearing properties
- Supported in SOL107, 108, 110, 111, 128, 400

**Output:**
- SOL 110 f06 summary
Rotordynamics Enhancements to Bearing Modeling

Usage: External Routine

---

### NEW GBUSH2Dmr format (with MASS) ---

```plaintext
subroutine ext_GBUSH2Dmr (flag, nomlkvm, mmlspd, freqva, ref_speed,
  + larray, rarray, callarray,
  + len_larray, len_rarray, len_callarray,
  + elid, +
  + kxx, kxy, kyy, +
  + cxx, cyx, cyy, +
  + mxx, mxy, myy, +
  + error_code)
```

**C INPUT ARGUMENT LIST:**

C | FLAG : Flag to identify type of analysis.  
C | SOL107/110 -> APP4=CEIG -> FLAG=1  
C | SOL108/111/126 -> APP4=FRQ -> FLAG=2  
C | SOL400 -> APP4=TRAN -> FLAG=3  
C | NLMSDM : Flag to identify nominal or freq dependent cond.  
C | = 0 (Nominal condition)  
C | = 1 (Freq dependent condition)  
C | NMLSFDO : Nominal Speed set in ELEMDUS card (7th field)  
C | Each ELEMDUS card has separate NMLSFDO entry  
C | in CPS & it is used only for that ELEMDUS/FRUSHD combination  
C | FREQVA : Frequency of Excitation in SOL 108 & SOL 111  
C | (always) Same for Rotor SPEED for FREQ analysis with SYNC option  
C | in CPS & not used in Complex Eigenvalue & Transient analysis  
C | REF_SPEED : Reference Rotor Speed in RGYRO card  
C | (always) Used in FREQ analysis with ASYNC, Complex Eigenvalue  
C | in CPS & Transient analysis  

**C OUTPUT ARGUMENT LIST:**

C | KXX, KXY, KYY : 2x2 ELEMDUS OUTPUT STIFFNESS MATRIX  
C | KXX, KYY, KXY, KYY : 2x2 ELEMDUS OUTPUT STIFFNESS MATRIX  
C | KXX, KYY, KXY, KYY : 2x2 ELEMDUS OUTPUT STIFFNESS MATRIX

---

### BELOW TABLE SHOWS SPEED/FREQUENCY USED UNDER DIFFERENT ROTOR RGYRO SETTINGS

<table>
<thead>
<tr>
<th>ANALYSIS</th>
<th>RGYRO SYNFLAG</th>
<th>WHICH FREQUENCY</th>
</tr>
</thead>
<tbody>
<tr>
<td>TYPE</td>
<td>SETTING</td>
<td>/SPEED USED</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FREQ RESP</th>
<th>SYNC</th>
<th>FREQ</th>
<th>SPEED</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOL108,111,126</td>
<td>ASYNC</td>
<td>SPEED</td>
<td>Units in CPS</td>
</tr>
<tr>
<td>SOL400</td>
<td>ASYNC</td>
<td>SPEED</td>
<td>Units in CPS</td>
</tr>
</tbody>
</table>

1. For Frequency response analysis with synchronous excitation, excitation frequency is equal to spin rate of reference rotor.
2. For Frequency response analysis with asynchronous excitation, spin rate is constant and equal to rotation speed of the reference rotor.

---

### CEIG | SYNC | SPEED | Units in CPS
| SOL107,110 | ASYNC | SPEED | Units in CPS |

1. For Complex modes analysis with synchronous excitation,
2. For Complex modes analysis with asynchronous excitation,
3. If CAMPBELL call is used, SPEED is set from DVVAL table.

### TRANSIENT | SYNC | NOT ALLOWED
| SOL400 | ASYNC | SPEED | Units in CPS |

1. For Transient analysis with asynchronous excitation,
2. For Transient analysis with synchronous excitation,
3. Spin rate is constant and equal to rotation speed of the reference rotor.

---

- Rotor speed, Frequency of excitation to external routine is always CPS
- SOL108 with SYNC uses frequency of excitation as reference
- For other setups, Rotor Speed from RGYRO card is used
Rotordynamics Enhancements to Bearing Modeling

Usage: PBSH2DT card

- A PBUSH2D with the same PID must exist
- Supports only TABLED1 id for K, B, M entry
- Values from selected TABLED1 entries will be used in any frequency-dependent loop.
- Values will be obtained by interpolation or extrapolation, if frequency of interest does not match with TABLED1 entry
- Any field left blank indicates that associated stiffness, damping, or mass is not frequency-dependent & nominal values will be used for that term in the solution

Format:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBUSH2DT PID</td>
<td>“K“</td>
<td>K11</td>
<td>K22</td>
<td>K12</td>
<td>K21</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>“B“</td>
<td>B11</td>
<td>B22</td>
<td>B12</td>
<td>B21</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>“M“</td>
<td>M11</td>
<td>M22</td>
<td>M12</td>
<td>M21</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PBUSH2DT 311 K 601 602 616 612
B 603 604 613 614
M 605 606 615 616

Output: Complex Eigenvalue, Frequency resp

SOL110
Rotordynamics Enhancements to Bearing Modeling

Usage: NLRSFDA service within NLRSFD card

- Enabled through following definition as first line in main run file,
- NAME2 (2\textsuperscript{nd} row, 5\textsuperscript{th} field) takes new ‘nlrsfda’ definition
- External routine takes acceleration in addition to displacement & velocity of previous time steps
- Applicable only in SOL128 & SOL400

Format:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>NLRSD</td>
<td>SID</td>
<td>GA</td>
<td>NB</td>
<td>PLAN</td>
<td>BDA</td>
<td>BLEN</td>
<td>BCLR</td>
<td>SOLN</td>
<td></td>
</tr>
<tr>
<td>VISCO</td>
<td>PVPACO</td>
<td>NBPORT</td>
<td>PSR1</td>
<td>THETA1</td>
<td>PSR2</td>
<td>THETA2</td>
<td>NPNT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OFFSET1</td>
<td>OFFSET2</td>
<td>GROUP NAME</td>
<td>NAME2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RDATA1</td>
<td>RDATA2</td>
<td>RDATA3</td>
<td>RDATA4</td>
<td>RDATA5</td>
<td>RDATA6</td>
<td>RDATA7</td>
<td>RDATA8</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NLRSD</th>
<th>40</th>
<th>911</th>
<th>912</th>
<th>XY</th>
<th>11.0</th>
<th>5.0</th>
<th>0.02</th>
<th>long</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>14.0</td>
<td>15.0</td>
<td>2</td>
<td>150.</td>
<td>10.0</td>
<td>350.</td>
<td>135.0</td>
<td></td>
</tr>
<tr>
<td>0.05</td>
<td>0.1</td>
<td>testnlr</td>
<td>nlrsfda</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100.0</td>
<td>200.0</td>
<td>300.0</td>
<td>400.0</td>
<td>500.0</td>
<td>600.0</td>
<td>700.0</td>
<td>800.0</td>
<td></td>
</tr>
</tbody>
</table>

External routine

```latex
\text{subroutine} \text{ext\_nlrsfda}(\text{sid, ga, gb, plane,} \ldots)
\begin{align*}
&\text{& bdia, blien, bclr,} \\
&\text{& soln, visco, pvpaco, nport,} \\
&\text{& pres1, theta1,} \\
&\text{& pres2, theta2,} \\
&\text{& npnt, offset1, offset2,} \\
&\text{& evalname, time,} \\
&\text{& xx, yy, xdt, ydt, xdtt, ydtt,} \\
&\text{& xb, yb, xbt, ybt, xbt, ybt,} \\
&\text{& fx, fy, fuseit, bisect,} \\
&\text{& parm1, parm2, parm3, parm4,} \\
&\text{& parm5, parm6, parm7, parm8,} \\
&\text{& omega})
\end{align*}
```

C INPUT PARAMETERS
- XX REAL/SCALAR Rotor Joint (I-END) Horizontal Displ.
- YY REAL/SCALAR Rotor Joint (I-END) Vertical Displ.
- KDT REAL/SCALAR Rotor Joint (I-END) Horizontal Velocity.
- VDT REAL/SCALAR Rotor Joint (I-END) Vertical Velocity.
- KDTT REAL/SCALAR Rotor Joint (I-END) Horizontal Acceleration.
- YDTT REAL/SCALAR Rotor Joint (I-END) Vertical Acceleration.
- XBD REAL/SCALAR Stator Joint (J-END) Horizontal Displ.
- YBD REAL/SCALAR Stator Joint (J-END) Vertical Displ.
- XBTT REAL/SCALAR Stator Joint (J-END) Vertical Acceleration.

C OUTPUT PARAMETERS
- FX : Force in X
- FY : Force in Y
Rotordynamics Enhancements to Bearing Modeling

- **Limitations**
  - PBSH2DT bulk data requires the existence of a PBUSH2D card of same PID
  
  - User cannot use both old CBUSH2D CONNECT SERVICE and new CBUSH2DA CONNECT SERVICE in the same analysis
  
  - User can combine new CBUSH2DA CONNECT SERVICE with PBSH2DT bulk data definition only if the PIDs are different
  
  - User cannot use both the old NLRSFD CONNECT SERVICE and the new NLRSFDA CONNECT SERVICE in the same analysis
  
  - Both PBSH2DT & CBUSH2DA SERVICE works only for rotordynamics analysis – ignored if there is no rotor in the model
  
  - For running user defined external CONNECT SERVICE, correct SDK (Software Development Kit) need to be installed
Optimization Enhancements
Optimization Enhancements

- Overhang constraints
- Anisotropic solid elements in a topological design group
- Element iterative solver CASI for segment to segment permanent glued contact
- Easier DRESP2 interface to maximize structural stiffness and fundamental frequency
Optimization: Overhang Constraints

➢ Introduction
  • Additively manufactured components often require temporary support material during the 3D printing process
    • Longer time to build
    • More material usage – increased cost
    • Extensive work to remove supports – increased cost
  • 45-degree rule (based on best practice)

➢ Benefits
  • Optimize to remove or minimize need for supports
  • Study influence of the print direction
Optimization: Overhang Constraints

- **Usage**
  - Simply add overhang constraints to TOPVAR entry
  - Coordinate system ID
  - Print direction

- **Example**

<table>
<thead>
<tr>
<th>Format:</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOPVAR</td>
<td>ID</td>
<td>LABEL</td>
<td>PTYPE</td>
<td>XINIT</td>
<td>XLB</td>
<td>DELXV</td>
<td>POWER</td>
<td>PID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>“SYM”</td>
<td>CID</td>
<td>MS1</td>
<td>MS2</td>
<td>MS3</td>
<td>CS</td>
<td>NCS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>“CAST”</td>
<td>CID</td>
<td>DD</td>
<td>DIE</td>
<td>ALIGN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>“EXT”</td>
<td>CID</td>
<td>ED</td>
<td>ALIGN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>“PRINT”</td>
<td>CID</td>
<td>PD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Example Images:
  - a) Without Overhang Constraint
  - b) With Overhang Constraint
Optimization: Support Anisotropic Solid Elements in a Topological Design Group

➢ Introduction
  • Design anisotropic solid elements referencing MAT9

➢ Benefits
  • Generate a light weight and cost-efficient design when the strength is along the given direction
  • Additive manufacturing

➢ Usage
  • A PSOLID ID on TOPVAR references a MAT9 entry
Optimization: Support CASI for Segment to Segment (S2S) Permanent Glued Contact

➢ Introduction
• Large solid models in SOL 200 and SOL 101 would benefit from the availability of the CASI solver
• Previously only supported node to segment permanent glued contact

➢ Benefits
• Significant performance improvement ~ 10x

➢ Usage
• Simply add SMETHOD=ELEMENT to use the CASI iterative solver
Optimization: Maximize Structural Stiffness and Frequency

➢ Introduction
  • Easier to use when the objective is to maximize structural stiffness (i.e. minimize compliance) and fundamental frequency

➢ Benefits
  • Ease of use - no complex DEQATN required
    • New DRESP2 function SFMAX
  • Typically needed in topology, topometry and topography optimization

➢ Usage
  • Analysis=MODE placed last
  • DRESP1 ID for Mode placed last in DRESP2
  • Set DRESP2 FUNC=SFMAX
Thermal Loading for CBUSH and CFAST Elements
Thermal Loading for CBUSH and CFAST

➢ Introduction
  • Thermal loading for CBUSH and CFAST elements. – useful when connectors or bushings are subject to temperature change

➢ Benefits
  • The PBUSH and PFAST entries has been expanded to allow the user to apply thermal loading to the CBUSH element in linear, dynamic and nonlinear structural analysis solution sequences

➢ Usage
  • Request Temperature in Case Control using combinations such as TEMP(INIT) and TEMP(LOAD) or TEMP(LOAD) with initial temperature specified on the PBUSH or PFAST entry
  • New PBUSH keyword “T”
    • Specify ALPHA for thermal expansion
    • Specify TREF if no TEMP(INIT)
    • Specify COINL for coincident grids
  • For PFAST similar fields on continuation entry
Layered Solids and Solid Shells
Layered Solids and Solid Shells

Solid Composite Elements and Solid-Shell Composite Elements are available in:

- Linear statics (SOL 101)
- Normal modes (SOL 103)
- Linear buckling (SOL 105)
- Direct complex eigenvalue (SOL 107)
- Direct frequency response (SOL 108)
- Direct transient response (SOL 109)
- Modal complex eigenvalue (SOL 110)
- Modal frequency response (SOL 111)
- Modal transient response (SOL 112)
- Analysis only in SOL 200
- Already available in SOL 400
Layered Solids and Solid Shells

➢ Benefits:

- Modelling of thick composite beams, thick composite shells, composite solids
  - Modelling of large composite parts such as gas turbine blades, stringers for pressure vessels, etc.
  - Modelling of thick laminates, laminates subjected to three-dimensional state of stress
  - Cases with loads in direction of laminate thickness
  - Layered solid shells useful in cases where bending is dominant and model has fewer layers through thickness
  - All above-mentioned scenarios can be simulated in all linear solution sequences in MSC Nastran
Layered Solids and Solid Shells

➢ Usage:
  • PCOMPLS referenced by CHEXA

<table>
<thead>
<tr>
<th>CHEXA</th>
<th>10</th>
<th>20</th>
<th>1</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>5</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>7</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PCOMPLS</th>
<th>20</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>C8</td>
<td>SLCOMP</td>
</tr>
<tr>
<td></td>
<td>1001</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>1002</td>
<td>32</td>
</tr>
</tbody>
</table>

• Layered Solid (linear): In PCOMPLS C8 line, BEH8=SLCOMP, INT8=L
• Layered Solid (quadratic): In PCOMPLS C20 line, BEH8=SLCOMP, INT8=Q
• Layered Solid Shell (assumed strain option): In PCOMPLS C8 line, BEH8=SLCOMP, INT8=ASTN
Layered Solids and Solid Shells

Output

- Example: wrapped thick cylinder under pressure loading

- Results output to OP2 and HDF5
- Postprocess in Patran
- Results more closely match NAFEMS benchmark

<table>
<thead>
<tr>
<th>Hoop Stress (N/mm²)</th>
<th>NAFEMS</th>
<th>CQUAD4/PCOMP/PSHLN1</th>
<th>CHEXA/PCOMPLS</th>
</tr>
</thead>
<tbody>
<tr>
<td>R=24mm (layer2 mid)</td>
<td>1483</td>
<td>1414</td>
<td>1466</td>
</tr>
<tr>
<td>R=26mm (layer1 mid)</td>
<td>822</td>
<td>875</td>
<td>831</td>
</tr>
</tbody>
</table>
Nastran Embedded Fatigue Enhancements
NEF Fatigue Enhancements: Reduction of Output

➢ **Introduction**
  • Fatigue analysis can generate a lot of output
  • Previously the only method to limit output was with TOPSTR or TOPDMG on FTGDEF
  • New NENTS option allows output to be limited by:
    • Most damage
    • Smallest safety factor
    • Maximum stress/strain range

➢ **Benefits**
  • Minimizing output reduces time to locate critical points of interest

➢ **Usage**
  • Specify NENTS on the FTGDEF entry

➢ **Output**
  • The output is reduced to return only the number of entities requested
NEF Fatigue Enhancements: Stress/Strain Range Vector

➢ Introduction
  • If a critical plane analysis is requested the critical angle reported is in the element coordinate system – this option allows for the stress/strain range to be transformed and printed in the basic coordinate system for 2D elements

➢ Benefits
  • Allows users to obtain stress/strain range vectors in the Nastran basic coordinate system for visualization purposes

➢ Usage
  • STROUT=2 on the FATIGUE case control entry

➢ Output

<table>
<thead>
<tr>
<th>ELEMENT</th>
<th>LAYER</th>
<th>LIFE</th>
<th>LOG-LIFE</th>
<th>LIFE</th>
<th>LOG-LIFE</th>
<th>DAMAGE</th>
<th>LOG</th>
<th>MAXIMUM/MINIMUM STRESS</th>
<th>SAFETY</th>
<th>CRIT.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>171</td>
<td>22</td>
<td>0.3672E+17</td>
<td>16.56</td>
<td>0.3672E+17</td>
<td>16.56</td>
<td>0.2724E-16</td>
<td>-16.56</td>
<td>0.4992E+04</td>
<td>-0.4992E+04</td>
<td>0.500E+01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.9976E+04</td>
<td>0.4007E+03</td>
<td>0.0000E+00</td>
<td>175.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
NEF Fatigue Enhancements: Scalar Stress Response

➢ Introduction

- Fatigue life is determined from a scalar stress time history for pseudo-static and transient dynamic analysis – the actual stress time history used to calculate fatigue life can now be output and accessed

➢ Usage

- STROUT=4 on the FATIGUE case control entry (used in conjunction with NENTS)

➢ Benefits

- Allows users to obtain actual scalar stress response history computed and used by the fatigue analysis

<table>
<thead>
<tr>
<th>TIME</th>
<th>ELEMENT ID</th>
<th>GRID ID</th>
<th>SN (ABSM)</th>
<th>EVENT ID</th>
<th>ALL EVENTS FATIGUE ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0</td>
<td>1</td>
<td>0.000E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td></td>
<td></td>
<td>0.889E+02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.0</td>
<td></td>
<td></td>
<td>0.119E+03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.0</td>
<td></td>
<td></td>
<td>0.201E+02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.0</td>
<td></td>
<td></td>
<td>0.257E+03</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
NEF Fatigue Enhancements: Stress Tensor History

➢ Introduction
• For SOL 112, the stress tensor history output can be requested for only those entities of the fatigue analysis

➢ Usage
• STROUT=8 on the FATIGUE case control entry

➢ Benefits
• Allows users to obtain stress tensor history used by the fatigue analysis
• STRESS case control is not necessary with this option
NEF Fatigue Enhancements: FATIGUE Case Control (STROUT recap)

➢ Introduction
  • STROUT used to request stress output as returned from or provided to a fatigue analysis

➢ Usage
  • STROUT=1, 2, 4, or 8 on the FATIGUE Case Control Entry
  • Combinations are summed: 1+2+8 = 11
    • STROUT = 0 – no stress output
    • STROUT = 1 – prints physical or modal stress tensor passed to fatigue analysis
    • STROUT = 2 – prints stress range vector as returned from the fatigue analysis for critical plane analysis only (2D elements)
    • STROUT = 4 – prints scalar response stress history as computed by the fatigue analysis (for SOL 101, 103, & 112)
    • STROUT = 8 – prints the tensor stress response (for SOL 112 only)

➢ Benefits
  • Allows users to obtain stresses associated with, used, or returned from the fatigue analysis at only the locations computed by or filtered based on the fatigue analysis
NEF Fatigue Enhancements: NANGLE

- **Introduction**
  - Previously CRITICAL plane analysis of 2D elements was done with a 10-degree fixed incremental angle.
  - New option allows user-specified incremental angles as small as 1 degree.

- **Benefits**
  - Allowing for a more accurate critical angle to be determined.

- **Usage**
  - NANGLE on the FTGPARM entry.

- **Output**

<table>
<thead>
<tr>
<th>ID</th>
<th>ID</th>
<th>ID</th>
<th>LOG-LIFE</th>
<th>LIFE</th>
<th>LOG-LIFE</th>
<th>DAMAGE</th>
<th>LOG</th>
<th>MAXIMUM</th>
<th>MINIMUM</th>
<th>SAFETY</th>
<th>CRIT.</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>14</td>
<td>21</td>
<td>0.4006E+05</td>
<td>4.60</td>
<td>0.4006E+05</td>
<td>4.60</td>
<td>0.2496E-04</td>
<td>-4.60</td>
<td>0.3031E+03</td>
<td>-0.3031E+03</td>
<td>12.0</td>
</tr>
<tr>
<td>19</td>
<td>21</td>
<td>0.2862E+10</td>
<td>9.46</td>
<td>0.2862E+10</td>
<td>9.46</td>
<td>0.3494E-09</td>
<td>-9.46</td>
<td>0.1113E+03</td>
<td>-0.1113E+03</td>
<td>30.0</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>21</td>
<td>0.1009E+11</td>
<td>10.00</td>
<td>0.1009E+11</td>
<td>10.00</td>
<td>0.9911E-10</td>
<td>-10.00</td>
<td>0.1017E+03</td>
<td>-0.1017E+03</td>
<td>20.0</td>
<td></td>
</tr>
</tbody>
</table>
NEF Fatigue Enhancements: Duty Cycle Performance

➢ **Introduction**
  • SOL 112 with Duty cycle performance has been enhanced to use version 2 of the DCY file

➢ **Benefits**
  • Significant Performance Gains
  • Number of temporary participation factor files significantly reduced

➢ **Usage**
  • Automatic for SOL 112 and Duty Cycle

Elapsed times SOL 112 with Duty Cycle

<table>
<thead>
<tr>
<th>Nastran Version</th>
<th>Elapsed Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019fp1</td>
<td>35000</td>
</tr>
<tr>
<td>2020</td>
<td>5000</td>
</tr>
</tbody>
</table>
NEF Fatigue Enhancements: Multi-channel File Support

➢ Introduction
• RPC, S3T are multi-channel time series files
• DAC files are single channel time series files
• Use these files to define dynamic loading directly without converting to TABLED1 entries

➢ Benefits
• Directly point dynamic loads to an RPC, S3T, or DAC file channel

➢ Usage
• TABLRPC entry points to a UDNAME entry identifying a file and channel number
• TABLRPC IDs can be referenced by any entry that accepts a TABLED1 entry
• Internally the TABLRPC entries convert the channel data to TABLED1 entries, which can be output to the PUNCH file
• Require NEF license, even though it is not necessary to run a fatigue analysis
Numerical Methods / HPC
Performance and Efficiency Improvements for the Nastran AVL EXCITE Interface

SOL 103 with ACMS and Reduction to A-set Boundary Points
AVL EXCITE Interface Performance Improvements

➢ Introduction
- MSC Nastran – AVL EXCITE interface was streamlined for version 2018.0 (July 2018)
- Additional performance issues are now addressed in version 2020

➢ Benefits
- Faster job turnaround with reduced resource requirements (I/O and disk space)
- No change to user interface – existing jobs simply run faster
- Target use case: automotive engine block or power train with large number of a-set DOF

➢ Usage (no change)
- Case Control AVLEXB command
- EXTSEOUT may or may not be specified
- DOMAINSOLVER ACMS – ACMS is used to reduce the model to the a-set boundary
AVL EXCITE Interface Performance Improvements

➢ Performance
• Solid model (auto engine)
• 6.3 million grid points; 11.6 million DOF; O-size 18.5 million DOF
• Number of O-set eigenvalues = 26 (300 Hz)
• Number of A-set DOF = 3,500
• memorymax=230GB smp=16

<table>
<thead>
<tr>
<th>Version</th>
<th>Elapsed Time</th>
<th>Disk I/O</th>
<th>Max Disk</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019 FP1</td>
<td>722:57</td>
<td>9.9 TB</td>
<td>1.34 TB</td>
</tr>
<tr>
<td></td>
<td>(12h 2m)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2020</td>
<td>245:07</td>
<td>4.7 TB</td>
<td>970 GB</td>
</tr>
<tr>
<td></td>
<td>(4h 5m)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

➢ Test Environment
OS:        Red Hat Enterprise Linux Server release 7.1 (Maipo)
Model:     Intel(R) Xeon(R) CPU E5-2660 v3 @ 2.60GHz
Nsocket:   2
Ncore:     20 ( 2 X 10 )
Cache:     25600 KB
Special:   avx2
Ram:       257680 Mb
CASI Iterative Solver Support for Inertia Relief (INREL=-1)

Linear Statics SOL 101
CASI Iterative Solver Support for Inertia Relief

PARAM, INREL, -1

➢ Introduction
  • The CASI solver is an efficient and robust iterative solver that uses element geometry to form an effective preconditioner
  • The interface to this solver was never attempted for inertia relief analysis
  • Large solid models would benefit from the availability of the CASI solver

➢ Benefits
  • Faster job turnaround with reduced resource requirements (memory, I/O and disk space)
  • Simple and pre-existing user interface
  • Target use case: large automotive engine block models, static analysis

➢ Usage
  • Simply add SMETHOD=ELEMENT to use the CASI iterative solver
  • Can also set SMETHOD=sid and use ITER bulk data entry with SID=sid to customize options
    • PRECOND=CASI
  • PARAM, INREL, -1 and SUPORT entry are required
  • PARAM, INREL, -2 support in future version depending on user response
CASI Iterative Solver Support for Inertia Relie- PARAM, INREL, -1

➢ Performance
- Solid model (auto engine)
- 8 million grid points; 48 million DOF (a-size 24M dof)
- 1 load case
- memorymax=200GB smp=8

<table>
<thead>
<tr>
<th>Solver</th>
<th>Elapsed Time</th>
<th>Disk I/O</th>
<th>Max Disk</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSCLDL</td>
<td>240:04 (6h 0m 4s)</td>
<td>1.2 TB</td>
<td>471 GB</td>
</tr>
<tr>
<td>CASI</td>
<td>30:26 (30m 26s)</td>
<td>73 GB</td>
<td>60 GB</td>
</tr>
</tbody>
</table>

➢ Test Environment
OS: Red Hat Enterprise Linux Server release 7.1 (Maipo)
Model: Intel(R) Xeon(R) CPU E5-2660 v3 @ 2.60GHz
Nsocket: 2
Ncore: 20 (2 X 10)
Cache: 25600 KB
Special: avx2
Ram: 257680 Mb
Improved GPU Acceleration
Improved GPU acceleration for FastFR and MPYAD

➢ Introduction
• v2019 FP1: limited device memory on the GPU (16GB on average), extra large models in frequency response analysis were unable to utilize GPU’s for acceleration as they could not fit in GPU memory
• v2020: out-of-core implementations are deployed in FastFR (Fast Frequency) and MPYAD (Multiply-Add) modules
  • SOL 111 – FastFR, MPYAD.
  • Other solution sequences – MPYAD.

➢ Benefits
• Take advantage of GPU acceleration without an upper limit on the model size
• No GPU minimum memory requirement
• Older as well as newer GPU architectures can benefit

➢ Further details
• Only Nvidia CUDA compatible GPU’s (Kepler, Maxwell, Pascal, Volta) are supported in v2020
Improved GPU acceleration for FastFR and MPYAD

➢ Usage
  • No additional change to command line inputs, request specific gpu’s with “gpuid=0” or “gpuid=1” or “gpuid=0:1” for example

➢ Details and Improvement
  • Speed up of 2x-3x on overall elapsed times
  • Example shown is a high frequency automotive interior acoustic analysis of 44M dof containing total of 67,172 modes (structure + fluid) up to a range of 2,300Hz through SOL 111
  • Executed on 1 Kepler K40M Nvidia GPU which was launched in 2013
Other Updates
Feature Deprecation List

➢ Notice of features to be removed from MSC Nastran in 2020:
  • In an effort to streamline the MSC Nastran program and simplify ongoing maintenance activity, some obsolete capabilities have been identified and tagged for removal in a future release of the program in late 2020, allowing for a notice period of about twelve months. Please review the list of features marked for deprecation below to ensure that there will be no disruption to your use of MSC Nastran. If you see a feature that you currently use and do not wish to lose, contact MSC Technical Support to report it.

➢ Features tagged for removal:
  • P-elements
  • SOL 600 nonlinear solution sequence
  • Unstructured one- and two-digit solution sequences (e.g. SOL 3, SOL 24)
  • SOL 190 (DBTRANS)
  • TAUCS solver
  • MSGMESH
  • Obsolete DMAP modules
  • SSSALTERS
List of MSC Nastran Books

A list of some of the MSC Nastran documents is as follows:

Installation and Release Guides
- Installation and Operations Guide
- Release Guide

Reference Guides
- Quick Reference Guide
- DMAP Programmer’s Guide
- Reference Guide
- Utilities Guide

Demonstration Guides
- Linear Analysis
- Implicit Nonlinear (SOL 400)
- Explicit Nonlinear (SOL 700)

User’s Guides
- Getting Started
- Linear Static Analysis
- Dynamic Analysis
- Embedded Fatigue
- Embedded Vibration Fatigue
- Thermal Analysis
- Superelements and Modules
- Design Sensitivity and Optimization
- Rotordynamics
- Implicit Nonlinear (SOL 400)
- Explicit Nonlinear (SOL 700)
- Aeroelastic Analysis
- User Defined Services
- Non Linear (SOL 600)
- High Performance Computing
- DEMATD
Thank You!