

Contents under Creative Commons BY 4.0 license and code under MIT license. © Julian Parra 2019. This material is part of the Master of Engineering program by Julian Parra at Universidad EAFIT.

Input parameters, Output and General concepts.

3.1. Input files

A problem is defined in **NLDYNA** in terms of 7 input data files written in plain text format. Input files associated to different problems can be found in the examples section of this REPO.

3.1.1. General input parameters

The `parameters` file contains global data required for the problem solution.

```

----- INITIAL GLOBAL PARAMETERS FILE -----
Notes:
- 1 = compute, 0 = don't compute
- If NLSTA = 1 and NLDYNA = 1, only computes NLDYNA.
-----
0.02 ..... DeltaT
20.0 ..... Total-time
0.001 ..... Tolerance
100 ..... Max # iterations
0 .....
0 ..... Compute incremental nonlinear static analysis (NLSTA)
1 ..... Compute time dependent nonlinear dynamic analysis (NLDYNA)
1 ..... Accel - UX
0 ..... Accel - UY
0 ..... Accel - UZ
0 ..... Accel - RX
0 ..... Accel - RY
0 ..... Accel - RZ
0 .....
1 ..... Storage state variables history
1 ..... Storage internal forces history
0 ..... Data intentionally left as 0

```

The problem parameters are:

- **DeltaT:** Time step defining the input excitation when performing dynamic analysis (NLDYNA) or incremental pseudo-time step when conducting non-linear static analysis (NLSTA). This time step is different from the solution time step which is directly computed by the program.
- **Total-Time:** Total extension of the analysis time window. (This parameter may be different from the total duration of the input excitation).
- **Tolerance:** Precision employed to control the Newton-Raphson iterations (dimensionless).
- **Maximum number of iterations:** Maximum number of Newton-Raphson iterations within a time or deformation increment.
- **NLSTA:** Integer flag defining the analysis type (1 = Perform nonlinear static analysis).
- **NLDYNA:** Integer flag defining the analysis type (1 = Perform nonlinear dynamic analysis).

A value of 1 for any of the following set of parameters defines if inertial effects are active along the associated degree of freedom during a dynamic analysis step. These parameters are active only for `NLDYNA = 1`.

- **Accel-UX:** Activate inertial effects along the "X" direction.
- **Accel-UY:** Activate inertial effects along the "Y" direction.
- **Accel-UZ:** Activate inertial effects along the "Z" direction .
- **Accel-RX:** Activate inertial effects along the rotational mass along the "X" direction.
- **Accel-RY:** Activate inertial effects along the rotational mass along the "Y" direction.
- **Accel-RZ:** Activate inertial effects along the rotational mass along the "Z" direction.

This file also contains two general storage options.

- **Storage state variables history** (1 = Storage)
- **Storage internal forces history** (1 = Storage)

3.1.2. Material profiles

Material properties are defined in terms of material **profiles**. Since the code is not specific to any particular physical context the material properties are always user defined. In this context a **material profile** is a set of parameters defining material behavior and to be used by the different elements in the model. Geometric parameters, like those involved in the definition of structural elements are treated as material properties. The number of parameters may be different in each independent profile.

The sample profile shown in the figure below is defined by a 7-parameters set. This specific material profile is used by structural elements. (see Example 02 for reference)

MATERIAL/SECTION PARAMETERS INFORMATION FILE						
Global material's properties definitions. Nonlinear properties should be specified according to each particular elastoplastic model.						
e.g: 7 parameters for 2D frame: A / Iz / E / Density / Alpha-Damp / Betha-Damp / Gravity Value						
PARAMETERS INFORMATION						
0.25	0.00521	2000000	2.4	0.0	0.0	9.806
0.16	0.00213	2000000	2.4	0.0	0.0	9.806

The set of parameters required in the definition of the material profile used in sample problem 02 (Element type = 2) are

$$Profile = [A, Iz, E, \rho, \alpha_{damp}, \beta_{damp}, g]$$

and where

- **A:** Cross sectional area.
- **Iz:** Moment of inertia along Z axis (out of the window).
- **E:** Young's modulus.
- ρ : Material's density
- α_{damp} : Coefficient to calculate the element's damping matrix
- β_{damp} : Coefficient to calculate the element's damping matrix
- **g:** Value of gravity

Review the module `uelutil.py` for a definition of the different parameters that must be defined for the remaining elements currently available in NLDYNA

3.1.3. Nodal coordinates and boundary conditions

This files contains nodal coordinates (CX, CY and CZ). At each node the user must also define if the degrees of freedom (TrasX, TrasY, TrasZ, RotX, RotY and RotZ) are active or restrained. By default all the degrees of freedom are assumed active (a value of 0) while a value of -1 defines a restrained degree of freedom.

The user must restrain all those degrees of freedom associated to a non existing space dimension. For instance in a two-dimensional elasticity problem values of -1 must be assigned to those degrees of freedom associated to TrasZ, RotX, RotY and RotZ (see examples 02 for reference).

NODES INFORMATION FILE										
ID	CX	CY	CZ	TrasX	TrasY	TrasZ	RotX	RotY	RotZ	
0	0	0	0	-1	-1	-1	-1	-1	-1	
1	4	0	0	-1	-1	-1	-1	-1	-1	
2	4	4	0	-1	-1	-1	-1	-1	-1	
3	0	4	0	-1	-1	-1	-1	-1	-1	
4	0	0	3	0	0	0	0	0	0	
5	4	0	3	0	0	0	0	0	0	
6	4	4	3	0	0	0	0	0	0	
7	0	4	3	0	0	0	0	0	0	
8	0	0	6	0	0	0	0	0	0	
9	4	0	6	0	0	0	0	0	0	
10	4	4	6	0	0	0	0	0	0	
11	0	4	6	0	0	0	0	0	0	
12	0	0	9	0	0	0	0	0	0	
13	4	0	9	0	0	0	0	0	0	
14	4	4	9	0	0	0	0	0	0	
15	0	4	9	0	0	0	0	0	0	

3.1.4. Elements definition file

This file contains data required to define the different elements within the model. The number of parameters varies depending upon the element type. Each element is defined by a line with the following parameters:

3.1.4.1. Element identifier: Integer defining the element ID.

3.1.4.2. Element type: Integer defining the element type.

The following elements are currently available in NLDYNA:

- Type = 0. Linear - 1D Spring.
- Type = 1. Linear - 2D Simple frame.
- Type = 2. Linear - 2D Full frame.
- Type = 3. Linear - 2D Truss.
- Type = 4. Nonlinear - 4 noded plane strain.
- Type = 5. Nonlinear - 1D Spring.
- Type = 6. Linear - 2D Shear-Rotational spring.
- Type = 7. Linear - 1D Rotational spring.
- Type = 8. Nonlinear - 1D Rotational spring.
- Type = 9. Nonlinear - 1D soil spring with a bilinear P-Y curve.
- Type = 10. Linear - 3D Full frame considering shear effects

3.1.4.3. Material/Section profile for the current element.

Material/Section profile identifier assigned to the current element.

ELEMENT INFORMATION FILE				

Element's connectivity information. Element type and connectivity should be specified.				
e.g: 5 parameters for frame elements: ID / TYPE / MATERIAL / N1 / Nf				
7 parameters for 4-noded-solids: ID / TYPE / MATERIAL / N1 / N2 / N3 / N4				

ELEMENTS				
0	10	0	0	4
1	10	0	1	5
2	10	0	2	6
3	10	0	3	7
4	10	0	4	5
5	10	0	5	6
6	10	0	6	7
7	10	0	7	4
8	10	0	4	8
9	10	0	5	9
10	10	0	6	10

3.1.5. Nodal (static) loads

Nodal loads are specified in this file. If $NLSTA = 1$, these loads will be treated as pseudo-static loads and are applied incrementally from 0 up to its maximum value; if $NLDYNA = 1$, nodal loads will be treated as permanent loads and will be added to each time interval. A static load is specified by its nodal ID and the load magnitude along a given degree of freedom.

NODAL STATIC LOADS INFORMATION FILE						
NODE	Fx	Fy	Fz	Mx	My	Mz
5	100.0	235.0	-452.0	0.0	0.0	0.0
3	23.0	-12.5	245.30	0.0	-478.6	0.68
17	123.0	784.0	0.0	0.0	0.0	0.0
0	48.0	-7.5	287.30	0.0	58.6	0.0

3.1.6. Ground acceleration:

A value of $NLDYNA = 1$ implies that the program is expected to conduct nonlinear dynamic analysis requiring as input excitation a ground motion time history. Only the acceleration history must be specified at this file. The format of the time history file is shown in the figure below.

SEISMO GROUND ACCELERATION SIGNAL
Note: Signal's time step should be the same that the specified in the initial global parameters file.
0.0063
0.00364
0.00099
0.00428
0.00758
0.01087
0.00682
0.00277
-0.00128
0.00368
0.00864
0.0136
0.00727
0.00094
0.0042
0.00221
0.00021
0.00444
0.00867

3.1.7. General constraints:

As many automated structural analysis computer programs, NLDYNA admits to use master-slave general constraints options. Two general types of constraints are defined:

- **Floor diaphragms:** This is activated when Type = 0 to n, and MSTND = 0. For this case, NLDYNA automatically calculates the location of the master node based on the center of mass of the constraint nodes.
- **Rigid constraints:** This is activated when Type = -1 and MSTND = *Node ID*.

For general rigid constraints, master-slave degrees of freedom should be specified. For example, in rigid bodies, the most general three-dimensional constraint in which all degrees of freedom are related to the master node rigid body displacements, **TrasX, TrasY, TrasZ, RotX, RotY, RotZ** should be equal to 1.

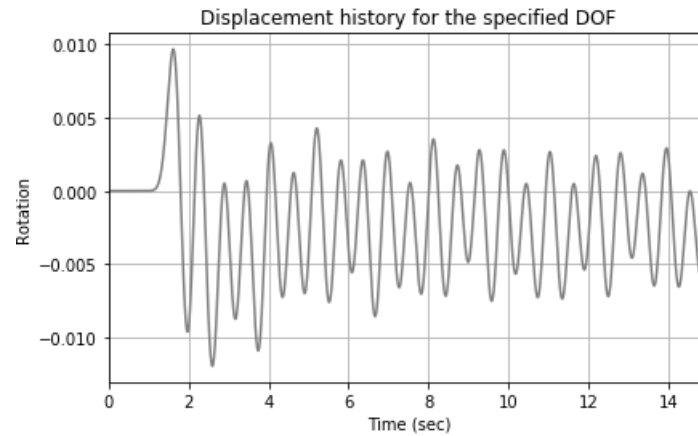
NODES GENERAL CONSTRAINTS INFORMATION FILE								
Note: - Type 0 to n = Floor diaphragm constraints. - Type -1 = Any other rigid constraint. - If Type = -1, master node for analysis solution should be specified, e.g. MSTND = 1. - If MSTND = 1 then it is the master node.								
NODE_ID	Type	MSTND	TrasX	TrasY	TrasZ	RotX	RotY	RotZ
4	0	0	1	1	0	0	0	1
5	0	0	1	1	0	0	0	1
6	0	0	1	1	0	0	0	1
7	0	0	1	1	0	0	0	1
8	1	0	1	1	0	0	0	1
9	1	0	1	1	0	0	0	1
10	1	0	1	1	0	0	0	1
11	1	0	1	1	0	0	0	1
12	-1	0	1	1	1	1	1	1
13	-1	13	1	1	1	1	1	1
14	-1	0	1	1	1	1	1	1
15	-1	0	1	1	1	1	1	1

3.2. Output

By default, the program returns the following results:

- The system's natural periods.
- Nodal displacements for each time step.
- Internal forces for each element at each time step.
- State variables at each time step. (This concept will be explained in section 3.3.1).

In the figure below, it is shown the displacement history obtained in the solution of Example 05, see notebook 10 for reference.



3.3. General concepts

To allow for a truly general multiphysics environment the nonlinear finite element code NLDYNA uses the concept of **User elements**. In a user element the physics per se is directly handle by the user which provides NLDYNA with the Jacobian matrix, residual vector and those variables which history are updated in each Newton iteration. Since the main program is unaware of the specific physics of the problem the variables are termed **State Variables**.

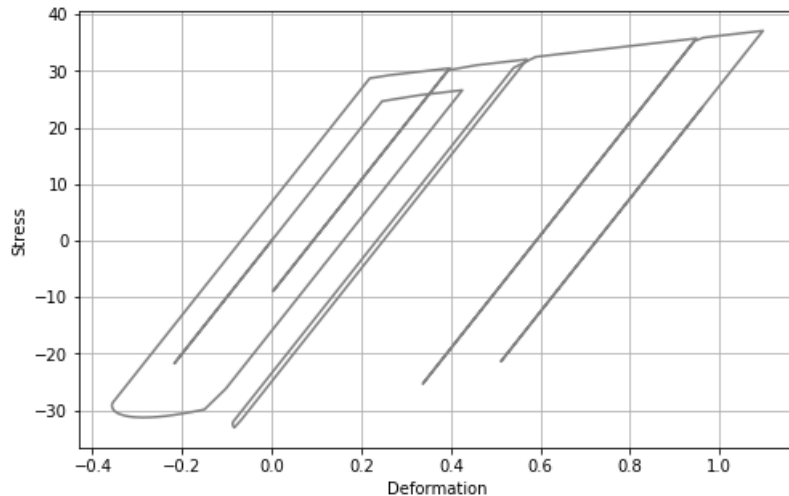
3.3.1. State variables

The state variables is the set of problem specific variables which are of interest in the physical problem being solved and those variables whose time history is required for the computation of the Jacobian matrix and residual vector. During program execution the last converged value of the state variables vector is passed to the element subroutine where it is updated depending upon the increment of the problem primary variable.

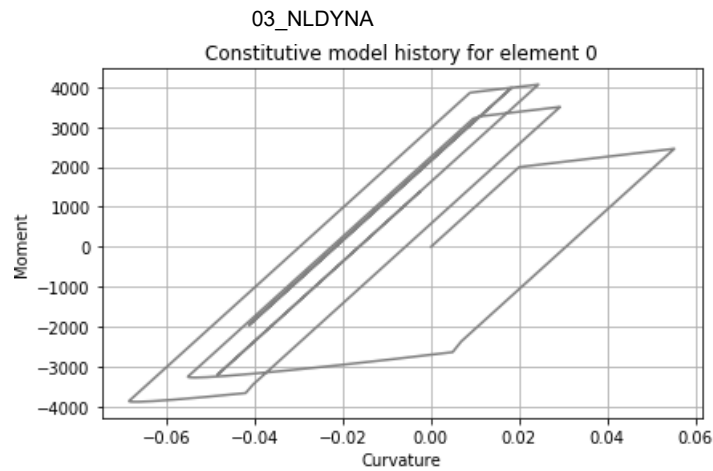
The concept of **State Variables** is briefly explained in terms of the simple example corresponding to a **type = 5** element corresponding to a Nonlinear -1D Spring along the horizontal direction. The constitutive response of the spring is described by a bilinear isotropic hardening elasto-plastic model (Simo & Hughes, 1991.). In this problem the state variables are:

$$svar = [\sigma_i, \delta_i, \delta_{elasi}, \delta_{plas_i}, \alpha_i]$$

- σ_i : Stress at increment i
- δ_i : Spring deformation at increment i
- δ_{elasi} : Elastic deformation at increment i
- δ_{plas_i} : Plastic deformation at increment i
- α_i : Isotropic hardening variable at increment i



In the figure below, it is shown the Moment-curvature history for one of the rotational springs used in the solution of Example 05 (see notebook 10 for reference).



3.3.1. User elements

A `user element` subroutine is the set of functions required for the computation of the element contribution to the problem Jacobian matrix and the corresponding residual vector. Since this concept is strongly tied to the Newton-Raphson algorithm it is important that the user reviews **Notebook 04** [UEL \(04_UEL_subroutine.ipynb\)](#), and **Notebook 05** [UMAT \(05_UMAT_subroutine.ipynb\)](#).

```
In [1]: from IPython.core.display import HTML
def css_styling():
    styles = open('./nb_style.css', 'r').read()
    return HTML(styles)
css_styling()
```

Out[1]:

In []: