Structures of data sd_resu_dyna, sd_dyna_phys and sd_dyna_gene

Summary:

This document describes the structures of data used for calculation of the transitory or harmonic answer using the operator DYNA_VIBRA.

Contents

1 General information

The concepts of the dynamic results, that they are expressed on a physical or generalized basis, have the structure of data in common sd_resu_dyna. This structure is supplemented by other objects making differentiation between the structures of data for the concepts of dynamics expressed on a physical basis (sd_dyna_phys) or on generalized basis (sd_dyna_gene).

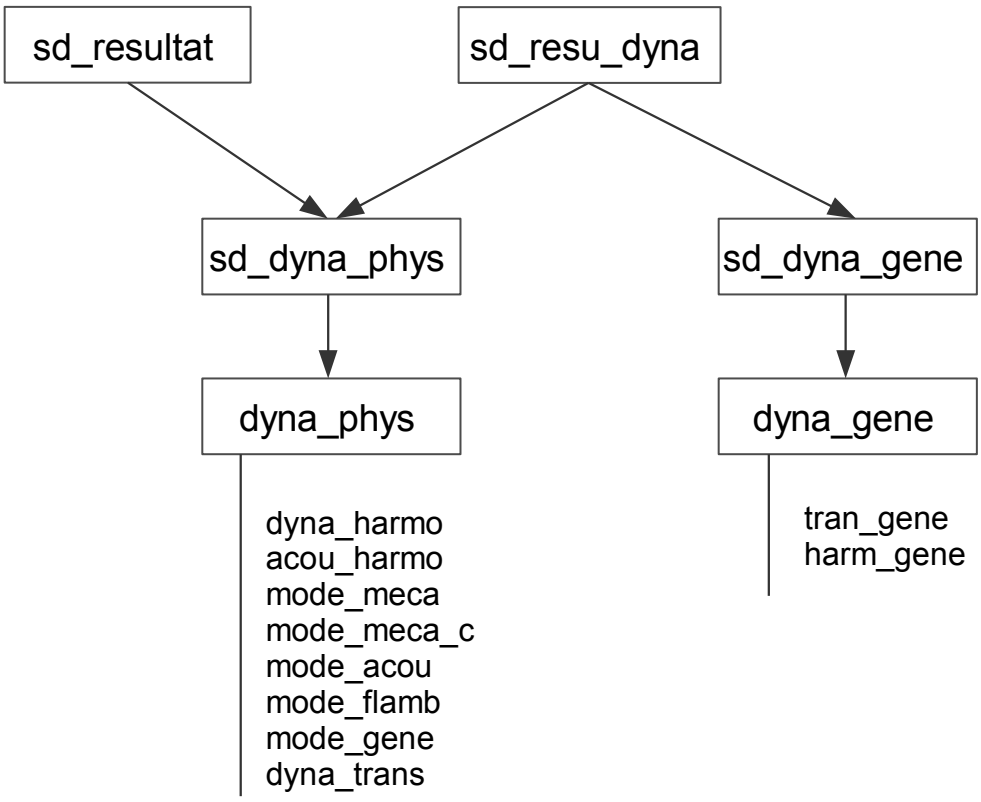
In this document are described the structure hat sd_resu_dyna as well as the construction of the structures of data sd_dyna_phys and sd_dyna_gene who result from this.

The concepts carried by these structures of data are:

- in physical base: acou_harmo, dyna_harmo, dyna_trans, mode_acou, mode_flamb, mode_meca, mode_meca_c.
- in generalized base: tran_gene and harm_gene.
- The concept mode_gene represent the only exception of a concept on generalized basis, carried by sd_dyna_phys.

2 Construction of the structures of data of the operators of dynamics

The figure below illustrates the organization of the structures of data of the unit of the types produced by operators of dynamics listed in the preceding section.



3 Tree structure of a Structure of Data sd_resu_dyna

Any structure of data of the type sd_resu_dyna must contain the objects obligatorily .REFD and .INDI whose tree structure is given below:

```
'sd_resu_dyna' '(K19)    :: == record
          (O)      '.REFD'    OJB  XC  V   K24
                   '.INDI'    OJB  S   V   I
```

3.1 Contents of objects JEVEUX of one sd_resu_dyna

3.1.1 Object '.REFD'

This object is a collection of vectors containing information on the matrices of mass, damping and rigidity but also of other information such as classification (NUME_DDL) or the list of dynamic and static interfaces having been used for calculation producing the concept.

3.1.2 Object '.INDI'

This object makes the indirection between the sequence number of the recorded fields and the objects recorded in the collection .REFD.

Notice : In a general way, the length of the object .INDI is equal to the number of occurrences of the collection .REFD. On the other hand, when the length of the object .INDI is equal to 1 and its value with -1, one considers that all the fields are in keeping with the first occurrence of the object .REFD.

3.2 Utility routines for the handling of sd_resu_dyna

The access to the contained information in objects JEVEUX .REFD and .INDI structures of data sd_dyna_resu can be done via utility DISMOI as described in documentation [D6.07.05].

In addition, following routines FORTRAN are available for the modification of the objects .REFD and .INDI sd_resu_dyna.

subroutine refdaj (stop, result, nbordr, numer, typre, conre, codret)

Goal: to add a new entry of reference to a dynamic result resting on one sd_resu-dyna. Objects .REFD and .INDI will be nouveau riches.

stop	K1	'F'/' ' 'F' : in the event of problem, one stops in fatal error ' ' : in the event of problem, one leaves the routine with codret =0
result	K8	Name of the result to enrich
nbordr	I	Many sequence numbers which will be filed with these references
numer	K*	Name of the classification of the concepts of reference (nume_ddl or, failing this, one prof_chno)
typre	K*	'DYNAMICS' / 'INTERF_DYNA'/'INTERF_STAT'/'MEASUREMENT' Type of reference to be added
conre	array [K24]	List of names of sd has to refer
codret	I	0 / 1 Code return 0: the operation failed 1: a modification was made

Examples:

```
cal refdaj ('F', modmec, nbordr, prchno, 'DYNAMICS', matric, iret)
cal refdaj ('F', tragen, nbordr, numddl, 'DYNAMICS' , matric, iret)
cal refdaj ('F', ritzba, nbordr, numddl, 'INTERF_DYNA', intdyn, iret)
```

subroutine refdcp (resin, solves)

Goal: to copy the contents of the dynamic references of the result `resin` in the result `solves`

`resin`	K8	Name of the dynamic result from where information of the objects will be copied.`REFD` and.`INDI`
`solves`	K8	Name of the dynamic result to enrich with the contents by `resin`

subroutine refdag (resin)

Goal: to double the size of the containers objects of the dynamic references.`REFD` and.`INDI`.
Note: this routine is automatically called in `refdaj` when there needs to increase the size of the containers objects of the dynamic references.

`resin`	K8	Name of the dynamic result of which size of the objects `.REFD` and `.INDI` will be doubled

4 Tree structure of the Structures of Data sd_dyna_phys

All the concepts of dynamics are expressed on a physical basis which is based on the structure of data `sd_dyna_phys`. This one inherits, like illustrates it the figure of section 2, at the same time of the properties of `sd_resu_dyna` described above but also of the properties of `sd_resultat`. The reader is invited to consult documentation [D4.06.08] to know the details of this last structure of data.

5 Tree structure of the Structures of Data sd_dyna_gene

All concepts of dynamics expressed in a base generalized (except for `mode_gene`) are based on the structure of data `sd_dyna_gene`. Following the example of `sd_dyna_phys`, this one inherits the properties of the structures sd_resu_dyna described above. In addition it is supplemented, according to the produced concept, by the tree structure below:

```
sd_dyna_gene (K8)
          '(11) .DESC'     :   OJB   S      V      I
          '(11) .REFD'     :   OJB   S      V      K24

          '(11) .DISC'     :   OJB   S      V      R
          '(11) .ORDR'     :   OJB   S      V      I
          '(11) .DEPL'     :   OJB   S      V      R or C
          '(11) .VITE'     :   OJB   S      V      R or C
          '(11) .ACCE'     :   OJB   S      V      R or C

   # if transitory calculation:

          '(11) .PTEM'     :   OJB   S      V      R

   # if EXCIT in the case of a transitory calculation:
          '(11) .FACC'     :   OJB   S      V      K8
          '(11) .FDEP'     :   OJB   S      V      K8
```

```
        '(11) .FVIT'     :   OJB    S       V      K8

    # if MULT_APPUI or CORR_STAT:

        '(11) .IPSD'     :   OJB    S       V      R

    # if produced by PROJ_MESU_MODAL   :
        '.PROJM'         :   sd_resu    (cf documentation [D4.08.05])

    # if BEHAVIOR:

        '(8) .NL.TYPE'       :   OJB    S       V      I
        '(8) .NL.VINT'       :   OJB    S       V      R
        '(8) .NL.VIND'       :   OJB    S       V      I
        '(8) .NL.INTI'       :   OJB    S       V      K24
```

5.1 Contents of objects JEVEUX of one sd_dyna_gene

5.1.1 Definition of some variables

nbmode : many modes (= dimension of the matrix of generalized mass)
nbnoli : number of non-linearity
nbsauv : many steps of saved times (keyword FILING)
nbexcit : many excitations (many occurrences of the keyword EXCIT)

5.1.2 Object .DESC

```
'(11) .DESC'     :   S   V   I      LONG=5
```

V (1): 1.2, 3, or 4.
2 if transitory calculation with nbnoli > 0
3 if transitory calculation with adaptive step
1 if transitory calculation other that in the preceding cases
4 if harmonic calculation

V (2): many basic vectors (nbmode)
V (3): many non-linearities (nbnoli)
V (4): many internal variables per moment of filing (nbvint)
V (5): static indicator of correction (0/1)

5.1.3 Object .DEPL

In the case of a transitory calculation, this object contains realities and is obligatory. For a harmonic calculation, on the other hand, it contains complexes and is an optional object.

```
'(11) .DEPL'     :   S   V   R/C    LONG=nbsauv*nbmode
```

V ((isauv-1) *nbmod+1 with isauv*nbmode): value of the displacement generalized for " isauv " ième pas de discretization (temporal or frequential according to the type of calculation)

5.1.4 Object .VITE

In the case of a transitory calculation, this object contains realities and is obligatory. For a harmonic calculation, on the other hand, it contains complexes and is an optional object.

```
'(11) .VITE'     :  S  V  R/C    LONG=nbsauv*nbmode
```

V ((isauv-1) *nbmod+1 with isauv*nbmode): value the speed generalized for " isauv " ième pas de discretization (temporal or frequential according to the type of calculation)

5.1.5 Object .ACCE

In the case of a transitory calculation, this object contains realities and is obligatory. For a harmonic calculation, on the other hand, it contains complexes and is an optional object.

```
'(11) .ACCE'     :  S  V  R      LONG=nbsauv*nbmode
```

V ((isauv-1) *nbmod+1 with isauv*nbmode): value of the acceleration generalized for " isauv " ième pas de discretization (temporal or frequential according to the type of calculation)

5.1.6 Object .ORDR

```
'(11) .ORDR'     :  S  V  I      LONG=nbsauv
```

V (I): I ème filed sequence number

5.1.7 Object .DISC

```
'(11) .DISC'     :  S  V  R      LONG=nbsauv
```

V (I): I ème value of the saved moment (for a transitory calculation) or frequency of calculation (for the harmonic case).

5.1.8 Object .PTEM

This object exists only for transitory calculations.

```
'(11) .PTEM'     :  S  V  R      LONG=nbsauv
```

V (I): step value of time to I ème moment of saved calculation.

5.1.9 Objects related to the presence of the keyword factor BEHAVIOR

Objects related to the presence of the keyword BEHAVIOR exist a transitory calculation only in the case of on modal basis.

```
'(8) .NL.TYPE'      :  OJB   S      V      I
'(8) .NL.VINT'      :  OJB   S      V      R
'(8) .NL.VIND'      :  OJB   S      V      I
'(8) .NL.INTI'      :  OJB   S      V      K24
```

5.1.9.1 Object .NL.TYPE

```
'(8) .NL.TYPE'      :  S  V  I      LONG=nbnoli
```

List of entireties giving the type of each linearity

5.1.9.2 Object .NL.VINT

```
'(8) .NL.VINT'      :  S  V  R      LONG=nbvint*nbsauv
```

Internal variables realities, stored in the order indicated by .NL.VIND, for each step of filing.

5.1.9.3 Object .NL.VIND

```
'(8) .NL.VIND'      :  S  V  I      LONG=nbnoli+1
```

List of the indices of indirection towards .NL.VINT for each non-linearity.

To illustrate it, the following example is taken:

If: VIND = [1, 5,12,25]

That corresponds to the presence of 3 non-linearities (nbnoli = 3)
The full number of internal variables for each step of filing is: nbvint = 24 = (25-1)

With each step of filing isauv, one has, in CAME, starting from the position pos = nbvint*isauv :
Of CAME (pos+1) with CAME (pos+4) : internal variables of non-linearity number 1
Of CAME (pos+5) and CAME (pos+11) : internal variables of non-linearity number 2
Of CAME (pos+12) and CAME (pos+24) : internal variables of non-linearity number 3

5.1.9.4 Object .NL.INTI

```
'(8) .NL.INTI'      :  S  V  K24    LONG=nbnoli*5
```

This vector contains the characteristics of each non-linearity.

For non-linearity number nli, one a:

V (5* (nli-1) +1): heading of non-linearity number nli
V (5* (nli-1) +2): name of the first node (node 1) non-linearity number nli
V (5* (nli-1) +3): name of the second node (node 2) of non-linearity number nli
V (5* (nli-1) +4): name of substructure of node 1
V (5* (nli-1) +5): name of the substructure of node 2

5.1.10 Objects related to the presence of the keyword factor EXCIT in the case of a transitory calculation

Objects .FACC , .FDEP and .FVIT exist a transitory calculation only in the case of.

5.1.10.1 Object .FACC

```
'(11) .FACC'     :  S  V  K8     LONG=2*nbexcit
```

V (1 with nbexcit): names of the functions of excitation of type acceleration
V (nbexcit+1 with 2*nbexcit): type of the function ('CONSTANT' , 'FUNCTION' , 'TABLECLOTH' ,...)

5.1.10.2 Object .FDEP

'(11) .FDEP' : S V K8 LONG=2*nbexcit

V (1 with nbexcit): names of the functions of excitation of type displacement
V (nbexcit+1 with 2*nbexcit): type of the function

5.1.10.3 Object .FVIT

'(11) .FVIT' : S V K8 LONG=2*nbexcit

V (1 with nbexcit): names of the functions of excitation of type speed
V (nbexcit+1 with 2*nbexcit): type of the function

5.1.11 Object related to the presence of the keywords MULT_APPUI or CORR_STAT

The object .IPSD can exist a transitory calculation only in the case of.

5.1.11.1 Object .IPSD

'(11) .IPSD' : S V I LONG=nbexcit*neq
neq corresponds to the numbers of equations (i.e the rank of the matrix of rigidity).

V (1+neq* (iexcit-1) with neq*iexcit): Component of the vector PSI*direction earthquake in the case of a multimedia seismic calculation (case multi-support) or component of the vector of the corrected clean modes (static case correction).