# Operator `DYNA_LINE_HARM`

## 1 Drank

Compute the dynamic response complexes of a system to a harmonic excitation: direct computation for a structure in physical space, computation by harmonic substructuring for several under structures defined by their modal base in modal coordinates.

This dynamic response can be evaluated in several successive works (concept reentrant).

Product a data structure of the dyna_harmo `type` or `acou_harmo` or `harmo_gene`.

*Warning : The translation process used on this website is a "Machine Translation". It may be imprecise and inaccurate in whole or in part and is provided as a convenience.*

*Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)*

# 2 Syntax

```
harm    [*]    =DYNA_LINE_HARM
    (  ◊   reuse=harm                            ,
       ◊RESULTAT=harm                            ,       /   [dyna_harmo]
                                                      /   [harm_gene]
       ◊MODELE=mo                                ,              [model]
       ◊   CHAM_MATER=chmat                   ,           [cham_mater]
       ◊   CARA_ELEM=carac                       ,            [cara_elem]

       ♦MATR_MASS=m                           ,          /   [matr_asse_DEPL_R]
                                                      /   [matr_asse_PRES_C]
                                                      /   [matr_asse_GENE_R]

       ♦MATR_RIGI=k                           ,          /   [matr_asse_DEPL_R]
                                                      /   [matr_asse_DEPL_C]
                                                      /   [matr_asse_PRES_C]
                                                      /   [matr_asse_GENE_R]
                                                      /   [matr_asse_GENE_C]

       ◊MATR_AMOR=c                           ,          /   [matr_asse_DEPL_R]
                                                      /   [matr_asse_PRES_C]
                                                      /   [matr_asse_GENE_R]
       ◊AMOR_MODAL   = _F (
                        /AMOR_REDUIT      =l η  ,        [l_R]
                        /LIST_AMOR           =cformule    η  ,       [listr8]
                        )

       ◊MATR_IMPE_PHI=imp                        ,       /   [matr_asse_DEPL_R]
                                                      /   [matr_asse_GENE_R]

       ♦/FREQ=lf                              ,            [l_R]
          /LIST_FREQ                    =cf ,            [listr8]

                                  ◊/TOUT_CHAM=' OUI',         [DEFAULT]
          /NOM_CHAM                    = |   ' DEPL',
                                         |   ' VITE',
                                         |   ' ACCE',

       ◊SOLVEUR=                          (
                                         … to see [U4.50.01]…
                                         ),

       ♦   EXCIT=
          _F (      ♦/VECT_ASSE=vecti                   ,           [cham_no]
                    /VECT_ASSE_GENE   =vect_gene           [vect_asse_gene]
                    /CHARGE              =chi  ,           [char_meca]

                 ♦/FONC_MULT_C=hci               ,       /   [fonction_C]
                                                      /   [formule_C]
                    /COEF_MULT_C      =aci  ,           [C]
                    /FONC_MULT           =hi   ,          /   [function]
                                                      /   [formula]
                                                      /   [three-dimensions
function]
                    /COEF_MULT           =ai   ,              [R]
                 ◊PHAS_DEG=/0                    .,          [DEFAULT]
                                         /phi   ,        [R]
                 ◊PUIS_PULS=/0                   ,           [DEFAULT]
                                         /ni ,          [Is]
          ),
```



*Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)*

```
          ◊   EXCIT_RESU=
              _F (      ♦RESULTAT=resuforc             ,      /  [dyna_harmo]
                                                       /  [harm_gene]
                    ♦COEF_MULT_C=aci        ,               [C]
                                 ),
          ◊TITER=tx        ,                                  [l_Kn]
        );


    if MATR_RIGI=        [matr_asse_DEPL_R]  then   [*]       dyna_harmo
                        [matr_asse_DEPL_C]                  dyna_harmo
                        [matr_asse_PRES_C]                  acou_harmo
                        [matr_asse_GENE_R]                  harm_gene
                        [matr_asse_GENE_C]                  harm_gene
```



Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)

# 3 Recalls

## 3.1 Equation of dynamic behavior under harmonic excitation

The damping of structure can be viscous or hysteretic [U2.06.03] [R5.05.04].

This operator solves the equation:

$$(-j\omega^3 \mathrm{I}-\omega^2\mathrm{M}+j\omega\mathrm{C}+\mathrm{K})x=\left\{\sum_{i=1}^{k} h_i(f)\omega^{n_i}\mathrm{e}^{j\pi\frac{\varphi_j}{180}}g_i(\mathrm{P})\right\}$$

Where:

$K$ represent a stiffness matrix real or complex

$M$ a mass matrix represents

$C$ represents a damping matrix

$I$ represents a matrix of acoustic impedance resulting from a formulation in displacement-pressure-potential.

$P$ is a point running of structure.

$\omega=2\pi f$: pulsation of excitation

$x$ : response complexes

## 3.2 Hysteretic damping

This operator also makes it possible to calculate the harmonic response of a structure with hysteretic

$$(\mathrm{K}-\omega^2\mathrm{M})x=\left\{\sum_{i=1}^{k} h_i(f)\omega^{n_i}\mathrm{e}^{j\pi\frac{\varphi_j}{180}}g_i(\mathrm{P})\right\}$$

damping.

With $K$ : complex stiffness matrix.
For detailed examples, one will refer to the document [U2.06.03].
This operator is usable in imposed force and imposed (relative reference frame or absolute).

Warning : The translation process used on this website is a "Machine Translation". It may be imprecise and inaccurate in whole or in part and is provided as a convenience.

Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)

# 4 Operands

## 4.1 Operand RESULTAT

◊RESULTAT = harm

Name of data structure result to enriching. This key word is compulsory if one is in D-entering concept mode (reuse).

## 4.2 MODEL operand

◊MODELE = Mo

Name of the defining concept the model whose elements are the object of harmonic computation.

## 4.3 Operand CHAM_MATER

◊CHAM_MATER = chmat

Name of the concept defining the affected material field on the model Mo.

## 4.4 Operand CARA_ELEM

◊CARA_ELEM = carac

Name of the concept defining the characteristics of the beam elements, shells, etc…

## 4.5 Operand MATR_MASS

♦MATR_MASS = m

Name of the concept stamps assembled corresponding to the mass matrix of the system.

## 4.6 Operand MATR_RIGI

♦MATR_RIGI = K

Name of the concept stamps assembled corresponding to the stiffness matrix of the system. A hysterical damping is obtained with a complex stiffness matrix.

## 4.7 Operand MATR_AMOR

◊MATR_AMOR = C

Name of the concept stamps assembled corresponding to the viscous damping matrix of the system.

## 4.8 Key word AMOR_MODAL

Factor key word to inform damping in the shape of lists of reduced damping with the operands following.

### 4.8.1 Operands AMOR_REDUIT / LIST_AMOR

/AMOR_REDUIT = lformule $\eta$

List of all reduced dampings: $(\eta_{1,}\eta_{2,}\ldots,\eta_n)$ .

/LIST_AMOR = cformule $\eta$

Name of the concept of the listr8 type containing the list of reduced dampings.

Warning : The translation process used on this website is a "Machine Translation". It may be imprecise and inaccurate in whole or in part and is provided as a convenience.

Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)

## 4.9 Operand MATR_IMPE_PHI

◊MATR_IMPE_PHI = imp

Name of the concept stamps assembled corresponding to the matrix of impedance for a system fluid-structure whose formulation is in displacement-pressure-potential $(\mathrm{u}, p, \varphi)$ [R4.02.02].

## 4.10 Operands FREQ/LISTE_FREQ

♦/FREQ = lf

Liste of all the frequencies of computation: (f1, f2,…, fn).

/LIST_FREQ = cf

Name of the concept of the listr8 type containing the list of the frequencies of computation.

## 4.11 Operands TOUT_CHAM / NOM_CHAM

```
                    ◊/TOUT_CHAM=' /NOM_CHAM
OUI'           = |   ' DEPL'
                     |   ' VITE'
                     |   ' ACCE'
```

Choice of the fields to calculating to represent the response: displacement, velocity, acceleration or three.

## 4.12 Operands solver

◊SOLVEUR

This key word factor is optional. It makes it possible to define the method of resolution of the system. Syntax is described in the document [U4.50.01].
In the current version, method MULT_FRONT is not available for the resolution of the systems with generalized matrixes.

## 4.13 Key word EXCIT

♦EXCIT

Operand allowing to define several excitations. Either by indicating a vector assembled corresponding to a loading, or of the loads which will lead to the computation and the assembly of a second member. For each occurrence of the key word factor, one defines a component of the excitation in the form $(h(f), g(\mathrm{P}), \varphi)$.

### 4.13.1 Operands VECT_ASSE/VECT_ASSE_GENE/CHARGE

Make it possible to define $g(\mathrm{P})$ spatial discretization of the loading, in the form of a field at nodes corresponding to one or more loads of force or imposed motion.

♦/VECT_ASSE = vecti

Name of the product concept by:

- operator ASSE_VECTEUR in imposed force or imposed of displacement in an absolute reference frame. The amplitudes of the excitation can be defined in the concepts of the type charges corresponding. The expected field is a field at nodes of quantity DEPL_R, DEPL_C or PRES_C.

/VECT_ASSE_GENE = vect_gene

Name of the product concept by:


Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)

- the operator `PROJ_VECT_BASE` who allows to project a vector assembled on a modal base or a basis of Ritz.
- the operator `ASSE_VECT_GENE` who allows to project a loading on a basis defined on one modele generalized for computations of dynamic substructuring.

`/CHARGE = chi`

`chi` name of the concept of loading specified by the ième occurrence of `EXCIT`.
The MODEL `key word` must be indicated if key word CHARGE `is used`.

### 4.13.2 Operands FONC_MULT_C / COEF_MULT_C / FONC_MULT / COEF_MULT

Make it possible to define $h(f)$ law of evolution, complex or real, of the frequency, applied to all the components of the field to the node associated with this occurrence. Several opportunities are given:

`♦/FONC_MULT_C=hci`

Name of the concept of the fonction_C `type` or `formule_C` defining a function $h(f)$ complex frequency $f$,

`/COEF_MULT_C = aci`

Coefficient complexes multiplying loading, independent of the loading,

`/FONC_MULT = hi`

Concept of type `function`, `formula` or `three-dimensions function` defining a function $h(f)$ real frequency $f$,

`/COEF_MULT = have`

multiplying real Coefficient of the loading, independent of the loading.

### 4.13.3 Operand PUIS_PULS

`◊PUIS_PULS = nor`

Makes it possible to define the power of the pulsation when the loading is function of the frequency; by default `ni`= 0.

### 4.13.4 Operand PHAS_DEG

`◊PHAS_DEG = phi`

Makes it possible to define the phase of each component of the excitation in degrees compared to a single reference of phase; by default $\varphi_i = 0$ .

### 4.13.5 Notice

For a problem with imposed motion, one defines the blocked degrees of freedom (kinematical conditions preliminary to the construction of `the cham_no`); one can then choose an excitation:

- in imposed displacement $n=0$ , $\varphi=0$ degree
- of imposed velocity $n=1$ , $\varphi=90$ degrees
- in imposed acceleration $n=2$ , $\varphi=180$ degrees

## 4.14 Operand EXCIT_RESU

*Warning : The translation process used on this website is a "Machine Translation". It may be imprecise and inaccurate in whole or in part and is provided as a convenience.*

*Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)*

◊EXCIT_RESU

This factor key word makes it possible to define several complements of loading in the form of a harmonic evolution of dyna_harmo type of assembled vectors second members, calculated on physical base.

### 4.14.1 Operand RESULTAT

This key word makes it possible to define the second complementary members to extract for each frequency from computation from one result already calculated fields from nodal forces.

♦RESULTAT = resuforc

Name of the concept of evolution harmonic of second members produces by the sequence of operator CALC_FORC_NONL [U4.84.21] in order to produce a transitory evolution of second members, and of operator REST_SPEC_TEMP [U4.63.34] to transform this transitory evolution into harmonic evolution. An example of use is provided in the case test SDLS119A.

### 4.14.2 Operand COEF_MULT_C

♦COEF_MULT_C = aci

Coefficient result complexes multiplying second member vector extracted the resuforc for each frequency of computation.

## 4.15 Operand TITER

◊TITER=tx

Title attached to the product concept by this operator [U4.03.01].

Warning : The translation process used on this website is a "Machine Translation". It may be imprecise and inaccurate in whole or in part and is provided as a convenience.

Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)

# 5 Example of use in imposed force

```
# -------------------------------------------------------------
#REFERENCE   : SFM/VPCS SDLD21    COMMANDS
# -------------------------------------------------------------
#SYSTEME   MASSE-RESSORT A 8 DEGREES OF FREEDOM AVEC AMORTISSEMENT
#VISQUEUX  PROPORTIONAL (RESPONSE HARMONIC)
#9  SPRINGS  8 MASSES  -- K = 1.E+5    M = 10    C = 50--
# -------------------------------------------------------------
DEBUT (CODE = _F (NOM = "SDLD21A"),);
# -------------------------------------------------------------
MY = LIRE_MAILLAGE ();

MO = AFFE_MODELE (  MAILLAGE = MY,
                    AFFE =  ( _F (      TOUT=         "OUI',
                                    "      MECHANICAL" PHENOMENE=,
                                    MODELISATION= "DIS_T",),
                              _F (            GROUP_NO= MASSE,
                                    MODELISATION= "DIS_T",),
                       ),);

CARELEM = AFFE_CARA_ELEM ( MODELS = MO,
                          DISCRET =  ( _F (         GROUP_MA= "ARISES",
                                             CARA=      "K_T_D_L",
                                             VALE=      (1.E+5,1., 1.),),
                                       _F (         GROUP_NO= MASSE,
                                             CARA=      "M_T_D_N",
                                             VALE=      10. ,),
                                       _F (         DEADENED GROUP_MA=,
                                             CARA=      "A_T_D_L",
                                             VALE=      (50. , 1. , 1.),),
                          ),),);

CH = AFFE_CHAR_MECA (  MODELS = MO,
                       DDL_IMPO =    ( _F (   GROUP_NO = "A_ET_B",
                                          DX = 0. , DY = 0. , DZ = 0. ,),
                                     _F (   GROUP_NO = "MASSE",
                                          DY = 0. , DZ = 0. ,),),
                       FORCE_NODALE =_F  ( NOEUD = "P4", FX = 1. ,),
                       );

MELR = CALC_MATR_ELEM (MODELS = MO, CHARGE = CH,
                       OPTION = "RIGI_MECA",
                       CARA_ELEM = CARELEM,);

MELM = CALC_MATR_ELEM (MODELS = MO, CHARGE= CH,
                       OPTION = "MASS_MECA",
                       CARA_ELEM = CARELEM,);

MELC = CALC_MATR_ELEM (MODELS = MO, CHARGE = CH,
                       OPTION = "AMOR_MECA",
                       CARA_ELEM = CARELEM,);

VECT = CALC_VECT_ELEM (CHARGE = CH, OPTION = "CHAR_MECA",);
# -------------------------------------------------------------
NUM = NUME_DDL ( MATR_RIGI = MELR,);
MATASSR = ASSE_MATRICE (MATR_ELEM = MELR, NUME_DDL = NUM,);
MATASSM = ASSE_MATRICE (MATR_ELEM = MELM, NUME_DDL = NUM,);
```

*Warning : The translation process used on this website is a "Machine Translation". It may be imprecise and inaccurate in whole or in part and is provided as a convenience.*

*Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)*

```
MATASSC = ASSE_MATRICE (MATR_ELEM = MELC, NUME_DDL = NUM,);
VECTASS = ASSE_VECTEUR (VECT_ELEM = VECT, NUME_DDL = NUM,);
# ------------------------------------------------------------
# SINUSOIDAL EXCITATION FORCE Of AMPLITUDE CRETE FX = 1.N
# IN POINT P4    FREQUENCY BAND 5 Hz - 40 Hz
# (AMPLITUDE INDEPENDENT OF the FREQUENCY)
LIFREQ = DEFI_LIST_REEL (  debut = 5. ,
                           INTERVALLE =_F    (     JUSQU_A=40.,
                                                   NOMBRE=70,),
                              );

DYNAHARM = DYNA_LINE_HARM (   MATR_MASS = MATASSM,
                           MATR_RIGI = MATASSR,
                           MATR_AMOR = MATASSC,
                           LIST_FREQ = LIFREQ,
                           EXCIT=_F          (     VECT_ASSE = VECTASS,
                                                COEF_MULT = 1. ,) ,
                            );

IMPR_RESU (  MODELS = MO, RESU =_F   ( RESULTAT = DYNAHARM,),);

FIN ();
```

*Warning : The translation process used on this website is a "Machine Translation". It may be imprecise and inaccurate in whole or in part and is provided as a convenience.*

*Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)*