Operator DYNA_VIBRA

1 Goal

DYNA_VIBRA is the single operator allowing the launching of all calculations of vibratory dynamics with Code_Aster:

- transients and harmonics
- on physical basis and modal basis

It is an macro-order which calls the historical operators DYNA_TRAN_MODAL, DYNA_LINE_TRAN and DYNA_LINE_HARM according to the choice that the user made on two keywords:

- TYPE_CALCUL, to choose between the transient and the harmonic,
- BASE_CALCUL, to choose between the physical base and the modal base.

The produced concepts are, according to these choices, of type tran_gene, dyna_trans, harm_gene, dyna_harmo and acou_harmo.

This document presents the catalogue of the operator and the two new keywords making it possible to direct the execution towards a historical operator. For description of the keywords and the operands, the reader is directed towards the handbooks of the operators subjacent with the macro-order:

DYNA_TRAN_MODAL	[u4.53.21]
DYNA_LINE_TRAN	[u4.53.02]
DYNA_LINE_HARM	[u4.53.11]

2 Syntax

```
nom_concept [dyna_vibra_prod] = DYNA_VIBRA (

      ◊  reuse =   nom_concept,
      ♦ BASE_CALCUL = (   | 'PHYS',
                          | 'GENE',
                        ),
      ♦ TYPE_CALCUL = (   | 'TRAN',
                          | 'HARM',
                        ),
# Keywords concerning the setting in data if harmonic or transitory calculation on physical basis :
      ◊  MODEL   = Mo,                                  [model]
      ◊  CHAM_MATER = chmat,                               [cham_mater]
      ◊  CARA_ELEM  = carac,                               [cara_elem]

# Keywords informing the assembled matrices :
      ♦  MATR_MASS =         my  ,                      /[matr_asse_gene_R]
                                                        /[matr_asse_depl_R]
                                                        /[matr_asse_pres_C]

      ♦  MATR_RIGI =         laughed   ,                      /
[matr_asse_gene_R]
                                                        /[matr_asse_depl_R]
                                                        /[matr_asse_pres_C]
                                                        /[matr_asse_depl_C]
                                                        /[matr_asse_gene_C]

      ◊  MATR_AMOR =         amndt  ,                      /[matr_asse_gene_R]
                                                        /[matr_asse_depl_R]
                                                        /[matr_asse_pres_C]

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

# if harmonic calculation with D-returning concept:
      ◊  RESULT =  harm,                             /[dyna_harmo]
                                                        /[harm_gene]
# introduction of modal damping:
      ◊  AMOR_MODAL = _F (
                        /  AMOR_REDUIT =       ,      [l_R]
                        /  LIST_AMOR   =    l_amor ,  [listr8]
                        /  MODE_MECA   =    mode,     [mode_meca]
                        /  NB_MODE     = / nbmode,    [I]
                                         / 9999,      [DEFECT ]
                        ),
# parameters for harmonic calculation:
      ♦  /  FREQ                  =  lf,             [l_R]
         /  LIST_FREQ             =  cf,             [listr8]

      ◊  /  TOUT_CHAM             =  'YES',          [DEFECT]
         /  NOM_CHAM              =  |   'DEPL',
                                     |   'QUICKLY',
                                     |   'ACCE',
# parameters of the diagrams of integration

       ◊ SCHEMA_TEMPS =_F (
```

```
                        ♦ DIAGRAM = (| 'NEWMARK',      [DEFECT]
                                     | 'EULER',
                                     | 'WILSON',
                                     | 'DEVOGE',
                                     | 'ADAPT_ORDRE1',
                                     | 'ADAPT_ORDRE2',
                                     | 'DIFF_CENTRE',
                                     | 'ITMI',
                                     | 'RUNGE_KUTTA_54',
                                     | 'RUNGE_KUTTA_32',

                          ),
# Keywords only associated with the diagram 'NEWMARK':
                        ◊ BETA    =/0.25,             [DEFECT]
                                   /beta,             [R]
                        ◊ GAMMA   =/0.5,              [DEFECT]
                                   /gamma,            [R]
 # Keywords only associated with the diagram 'ITMI':
                        ◊  BASE_ELAS_FLUI=   mix,  [melasflu]
                        ◊  Digital_QUICKLY_FLUI=   Nvitf,   [I]
                        ◊  STATE_STAT       =  /'NOT',   [DEFECT]
                                              /'YES',
                        ◊  PREC_DUREE    =   /1.E-2,    [DEFECT]
                                              /prec,     [R]
                        ◊  CHOC_FLUI     =   /'NOT',    [DEFECT]
                              /  'YES',
                        ◊  NB_MODE       =   Nmode,     [I]
                        ◊  NB_MODE_FLUI  =   Nmodef,    [I]
                        ◊  TS_REG_ETAB   =   tsimu,     [R]
# Keyword only associated with the diagram 'WILSON':
                        ◊ THETA   =/1.4,              [DEFECT]
                                   /th,               [R]
# Keywords only associated with the diagrams 'RUNGE_KUTTA_*':
                        ◊ TOLERANCE  =/1.E-3,         [DEFECT]
                                      /tol,           [R]
                        ◊ ALPHA      =/1.E-3,         [DEFECT]
                                      /alpha,         [R]

      ♦   INCREMENT =_F ( ♦   /  LIST_INST = litps,     [listr8]
                          /   NOT       = dt,         [R]
                        ◊   INST_INIT  = Ti,          [R]
                        ◊   /   INST_FIN= tf,         [R]
                            /   NUME_FIN= nufin,      [I]

                        ◊   VERI_NOT =    / 'YES',    [DEFECT]
                                          / 'NOT',

# Operands specific to an integration by step of adaptive times
                        ◊   QUICKLY_MIN =    / 'NORM',    [DEFECT]
                                             / 'MAXIMUM',
                        ◊   COEFF_MULT_NOT = / 1.1  ,     [DEFECT]
                                             / cmp  ,     [R]
                        ◊   COEFF_DIVI_PAS = / 1.33333334,[DEFECT]
                                             /   cdp,     [R]
                        ◊   NOT_LIMI_RELA =  /   1.E-6,   [DEFECT]
                                             /   per,     [R]
                        ◊   NB_POIN_PERIOD = 50,         [DEFECT]
                                             /   NR,      [I]
                        ◊   NMAX_ITER_NOT =  /   16,     [DEFECT]
                                             /   NR,      [I]
```

```
                          ◊  NOT_MAXIMUM =       dtmax,     [R]
                          ◊  NOT_MINIS =         dtmin,     [R]  ),

        ◊ ETAT_INIT = _F ( ♦  /  =res RESULT,           [tran_gene]
                          . If RESULT
                                ◊  /INST_INIT = to,       [R]
                                   /NUME_ORDR =  No,      [I]
                                ◊  / CRITERION = 'RELATIVE', [DEFECT]
                                ◊ PRECISION = / 1.E-06,   [DEFECT]
                                              / prec,     [R]
                                   / CRITERION = 'ABSOLUTE',
                                ♦ PRECISION = prec,       [R]

                             /    | DEPL = C,            /[vect_asse_gene]
                                                        /[cham_no]
                                  | QUICKLY = vo,          /
[vect_asse_gene]
                                                        /[cham_no]
                                  | ACCE = acc,         [cham_no]
                    ),
        ◊ EXCIT =   _F (  ◊  / VECT_ASSE      =  v,       [cham_no]
                          / VECT_ASSE_GENE =  v,       [vect_asse_gene]
                          / LOAD =  chi,            [char_meca]
                       ◊  Digital_ORDER =  nmordr,        [I]
                       ◊     /  FONC_MULT =  F,         /[function]
                                                        /[tablecloth]
                                                        /[formula]
                             /  COEFF_MULT = has,          [R]
                              /  FONC_MULT_C  =  hci,   /[fonction_C]
                                                        /[formule_C]
                             /  COEF_MULT_C  =  aci,   [C]

                             /  ◊  ACCE = ac,          [function]
                                                        /[tablecloth]
                                                        /[formula]
                                ◊  QUICKLY = VI,           [function]
                                                        /[tablecloth]
                                                        /[formula]
                                ◊  DEPL = dp,           [function]
                                                        /[tablecloth]
                                                        /[formula]
                       ◊  PHAS_DEG        =  /  0. ,         [DEFECT]
                                             /  phi,      [R]
                       ◊  PUIS_PULS       =  /  0,         [DEFECT]
                                             /  nor,          [Is]
# Operands and keywords specific to the seismic analysis
                       ◊  MULT_SUPPORT =   /   'NOT',          [DEFECT]
                                         /   'YES',
                       ◊  DIRECTION = (dx, Dy, dz, drx, dry Martini, drz),
   [l_R]
                       ◊  /   NODE =      lno,              [l_noeud]
                          /   GROUP_NO  = lgrno,           [l_groupe_no]

                       ◊  ♦ CORR_STAT  =   'YES'
                          ♦ D_FONC_DT  =   dfdt,           [function]
                          ♦ D_FONC_DT2 =   dfdt2,          [function]
                       ),

        ◊     / MODE_STAT = psi,                           [mode_meca]
              / MODE_CORR = modcor,                 [mult_elas, mode_meca]
```

```
        ◊ EXCIT_RESU    =
          _F (      ♦  RESULT    =  resuforc,              /  [dyna_harmo]
                                                           /  [harm_gene]
                                                           /  [dyna_trans]
                                                           /  [tran_gene]

                  ◊  /COEF_MULT   =  have,                      [R]
                     /COEF_MULT_C =  aci,                    [C]
              ),
```

End of the operands and keywords specific to the seismic analysis

```
      ◊  SHOCK = _F (
                  ◊  ENTITLE = int,                        [l_Kn]

                  /  ♦  /  NODE_1       =  no1,             [node]
                        /  GROUP_NO_1   =  grno1,           [group_no]
                     ◊  /  NODE_2       =  no2,             [node]
                        /  GROUP_NO_2   =  grno2,           [group_no]
                  /  ♦  /  MESH      =  my,                [mesh]
                        /  GROUP_MA     =  grma,            [group_ma]

                  ♦  OBSTACLE =   obs,                      [obstacle]
                  ♦  NORM_OBST =  NOR,                      [listr8]
                  ◊  ORIG_OBST =  ori,                      [listr8]
                  ◊  GAME =       / 1. ,                    [DEFECT]
                                  / game,                      [R]

                  ◊  ENG_VRIL =   gamma,                    [R]

                  ◊  DIST_1 =  dist1,                       [R]
                  ◊  DIST_2 =  dist2,                       [R]

                  ◊  UNDER_STRUC_1 =  ss1,                  [K8]
                  ◊  UNDER_STRUC_2 =  ss2,                  [K8]
                  ◊  REFERENCE MARK =    / 'TOTAL',
   [DEFECT]
                                  / nom_sst,                [K8]

                  ◊  RIGI_NOR =   kN,                       [R]
                  ◊  AMOR_NOR =   / 0. ,                    [DEFECT]
                                  / Cn,                     [R]
                  ◊  RIGI_TAN =   / 0. ,                    [DEFECT]
                                  / kt,                     [R]
                  ◊  AMOR_TAN =   / ct,                     [R]
                  ◊  FRICTION =
                                  / 'NOT'                   [DEFECT]
                                  / 'COULOMB'
                                     ♦ COULOMB = driven     [R]
                                  / 'COULOMB_STAT_DYNA'
                                     ♦ COULOMB_STAT = driven[R]
                                     ♦ COULOMB_DYNA = mud   [R]


# Operands specific to the taking into account of a transient speed
  for the rotors (number of revolutions variable)
    ◊ VITESSE_VARIABLE = 'NOT',                             [DEFECT]
                       / 'YES',
      # if VITESSE_VARIABLE=' OUI':
            ♦  VITE_ROTA =  vrota,                  [function]
            ♦  MATR_GYRO =  gyro,                   [matr_asse_gene_R]
            ◊  ACCE_ROTA =  arota,                  [function]
            ◊  MATR_RIGY =  gyro,                   [matr_asse_gene_R]
      # if VITESSE_VARIABLE=' NON':
            ♦  VITE_ROTA =  / 0.0,                  [DEFECT]
                            / vrota,                [R]
```

```
# Keyword specific to the taking into account of a crack in a rotor
   ◊  ROTOR_FISS=_F (
                  /  ♦  /  NOEUD_G       =  nog,            [node]
                        /  GROUP_NO_G    =  grnog,          [group_no]
                     ♦  /  NOEUD_D       =  nod,            [node]
                        /  GROUP_NO_D    =  grnod,          [group_no]
                  ♦  ANGL_INIT =  0.0,                      [DEFECT]
                  ◊  ANGL_ROTA =  0.0,                      [function]
                  ♦  K_PHI     =  kphi                      [function]
                  ♦  DK_DPHI   =  dkdphi                    [function]
                   )



      ◊  VERI_SHOCK =  _F (
                  ◊  STOP_CRITERE  = /  'YES',              [DEFECT]
                                    /  'NOT',
                  ◊  THRESHOLD =  /  0.5,                      [DEFECT]
                                 /  S,                      [R]
                  ),

      ◊  ANTI_SISM =   _F (
                   ♦ /  NODE_1    =  no1,               [node]
                     /  GROUP_NO_1   =  grno1,              [group_no]
                   ♦ /  NODE_2    =  no2,               [node]
                     /  GROUP_NO_2   =  grno2,              [group_no]
                   ◊ RIGI_K1 =    / 0. ,                   [DEFECT]
                                  / kN,                    [R]
                   ◊ RIGI_K2 =    / 0. ,                   [DEFECT]
                                  / kN,                    [R]
                   ◊ THRESHOLD_FX =  / 0. ,                   [DEFECT]
                                  / Py,                    [R]
                   ◊ C =          / 0. ,                   [DEFECT]
                                  / C,                     [R]
                   ◊ THEN_ALPHA = / 0. ,                   [DEFECT]
                                  / alpha,                 [R]
                   ◊ DX_MAX =     / 1. ,                   [DEFECT]
                                  / dx,                    [R]
                     ),

     ◊ DIS_VISC = _F (
         ♦   / NOEUD_1          = no1,                          [node]
             /GROUP_NO_1        = grno1,                     [group_no]
         ♦   /NOEUD_2           = no2,                          [node]
             /GROUP_NO_2        = grno2,                     [group_no]

         ♦   / K1               = k1,                             [R]
             / UNSUR_K1         = usk1,                           [R]
         ♦   / K2               = k2,                             [R]
             / UNSUR_K2         = usk2,                           [R]
         ♦   / K3               = k3,                             [R]
             / UNSUR_K3         = usk3,                           [R]
         ♦   C                  = C,                              [R]
         ♦   PUIS_ALPHA         =/0.5                          [defect]
                                  /alpha,                          [R]

         ◊   ITER_INTE_MAXI     =/20                           [defect]
                                  /iter                            [I]
         ◊   RESI_INTE_RELA     =/1.0E-06                      [defect]
                                  /resi                            [R]
      ),

      ◊ BUCKLING =    _F (
```

```
                ♦  /  NODE_1    =  no1,                    [node]
                   /  GROUP_NO_1   =  grno1,                  [group_no]
                ◊  /  NODE_2    =  no2,                    [node]
                   /  GROUP_NO_2   =  grno2,                  [group_no]
                ♦  OBSTACLE = obs,                         [obstacle]
                ◊  ORIG_OBST = ori,                        [listr8]
                ♦  NORM_OBST = NOR,                        [listr8]
                ◊  ENG_VRIL =   / 0,                       [DEFECT]
                                / gamma,                   [R]
                ◊  GAME = / 1. ,                           [DEFECT]
                          /jeu,                            [R]
                ◊  DIST_1 = dist1,                         [R]
                ◊  DIST_2 = dist2,                         [R]
                ◊  REFERENCE MARK =    /'TOTAL',
[DEFECT]
                                / nom_sst ,                [K8]
                ◊  RIGI_NOR = kN,                          [R]
                ◊  FNOR_CRIT = film,                       [R]
                ◊  FNOR_POST_FL = fseuil,                  [R]
                ◊  RIGI_NOR_POST_FL = k2,                  [R]
                ),

   ◊  RELA_EFFO_DEPL = _F (
                   ♦   NODE =      Noah,                   [node]
                   ◊  SOUS_STRUC = ss,                     [K8]
                   ◊  NOM_CMP =    nomcmp,                 [K8]
                   ♦  RELATION =   F,                      [function]
                      ),

   ◊  RELA_EFFO_QUICKLY = _F (
                   ♦   NODE =      Noah,                   [node]
                   ◊  SOUS_STRUC = ss,                     [K8]
                   ◊  NOM_CMP=     nomcmp,                 [K8]
                   ♦  RELATION =   F,                      [function]
                      ),
```

Keywords factors only associated with the coupling with code EDYOS

```
   ◊  COUPLAGE_EDYOS   =   _F (
                   ♦  VITE_ROTA        =     vrota,          [R]
                   ♦  PAS_TPS_EDYOS    =  dtedyos,           [R]
                      ),

   ◊  STAGE_EDYOS  =   _F (
                   ♦  /  UNIT      =  uled,                [I]
                      /  GROUP_NO  =  grnoed,              [group_no]
                       / NODE      =  noed,                [node]
                   ◊ TYPE_EDYOS    =  /   'PAPANL',
                                      /   'PAFINL',
                                      /   'PACONL',
                                      /   'PAHYNL',
                      ),
```

End of the keywords factors only associated with the coupling with code EDYOS

Keywords concerning the setting in data if transitory calculation on physical basis

```
      ◊ ENERGY =_F ()
```

End of the mots key concerning the setting in data if transitory calculation on physical basis

```
      ◊ FILING =_F (  ◊  / LIST_INST  = list      [listr8]
                           / INST       = in        [R]
                           / PAS_ARCH   = ipa       [I]
                        ◊/ CRITERION =  'RELATIVE', [DEFECT]
                          ◊ PRECISION = / 1.E-06,[  DEFECT]
                                        / prec,     [R]
```

```
                                / CRITERION =  'ABSOLUTE',
                                ♦ PRECISION = prec,        [R]
                          ),

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

    ◊   INFORMATION      =   /   1,
[DEFECT]
                         /   2,

    ◊   IMPRESSION =  _F (
              ◊  /   ALL = 'YES',                          [DEFECT]
                 /   LEVEL =    |   'DEPL_LOC',
                               |   'QUICKLY_LOC',
                               |   'FORC_LOC',
                               |   'RATE_CHOC',
              ◊   INST_INIT = Ti,                          [R]
              ◊   INST_END =  tf,                          [R]
               ◊  UNITE_DIS_VISC = links                   [I]
                 ),

    ◊   TITLE     =   title,                               [l_Kn]
)
```

Structure of data produced:

if BASE_CALCUL == 'PHYS'	and	TYPE_CALCUL == 'TRAN'	dyna_trans
if BASE_CALCUL == 'PHYS'	and	TYPE_CALCUL == 'HARM'	dyna_harmo
if BASE_CALCUL == 'GENE'	and	TYPE_CALCUL == 'HARM'	harm_gene
if AsType (MATR_RIGI) == matr_asse_pres_c			acou_harmo
if BASE_CALCUL == 'GENE'	and	TYPE_CALCUL == 'TRAN'	tran_gene

3 Operands specific to the order DYNA_VIBRA

3.1 TYPE_CALCUL

This keyword which makes it possible to make the choice between transitory calculation (TYPE_CALCUL=' TRAN') and harmonic calculation (TYPE_CALCUL=' HARM).

3.2 BASE_CALCUL

This keyword makes it possible to make the choice between a calculation on physical basis (BASE_CALCUL=' PHYS') and a calculation on modal basis (BASE_CALCUL=' GENE').

4 References towards the description of the other keywords and operands

The user who has made the choice TYPE_CALCUL=' TRAN' and BASE_CALCUL=' PHYS' will find the description of the keywords and operands specific to transitory calculation on physical basis in [U4.53.02], the user's manual of the operator DYNA_LINE_TRAN.

The user who has made the choice TYPE_CALCUL=' TRAN' and BASE_CALCUL=' GENE' will find the description of the keywords and operands specific to transitory calculation on modal basis in [U4.53.21], the user's manual of the operator DYNA_TRAN_MODAL.

The user who has made the choice TYPE_CALCUL=' HARM' and BASE_CALCUL=' GENE' or 'PHYS' will find the description of the keywords and operands specific to harmonic calculation in [U4.53.11], the user's manual of the operator DYNA_LINE_HARM