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:

<table>
<thead>
<tr>
<th>Operator</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>DYNA_TRAN_MODAL</td>
<td>[u4.53.21]</td>
</tr>
<tr>
<td>DYNA_LINE_TRAN</td>
<td>[u4.53.02]</td>
</tr>
<tr>
<td>DYNA_LINE_HARM</td>
<td>[u4.53.11]</td>
</tr>
</tbody>
</table>
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,
    ◊ CHAM_MATER = chmat,
    ◊ CARA_ELEM = carac,

  ◊ 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 ,
          [lstr8]
      / MODE_MECA = mode ,
          [mode_meca]
      / NB_MODE = nbmode ,
          [I]
      / 9999 ,
          [DEFECT ]
    },

  # parameters for harmonic calculation:
    ◊ / FREQ = lf,
      /
      [l_R]
    ◊ / LIST_FREQ = cf,
      /
      [lstr8]
    ◊ / TOUT_CHAM = 'YES',
      /
      [DEFECT]
    ◊ / NOM_CHAM
      |  'DEPL',
      | 'QUICKLY',
      | 'ACCE',

  # parameters of the diagrams of integration
    ◊ SCHEMA_TEMPS = _F {
`DIAGRAM` = (`'NEWMARK'`, [DEFECT]
  | `'EULER'`,
  | `'WILSON'`,
  | `'DEVOGÉ'`,
  | `'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,
  ◊ GAMMA = /0.5, [DEFECT]
       /gamma,

# 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,
       /th,

# 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_PON 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,
   . 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]
   | QUICKLY = vo, [cham_no]
   }
   [vect_asse_gene]
   | 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]
   ◊ 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’, [l_R]
◊ 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,  
   ◊ /COEF_MULT = have,  
   ◊ /COEF_MULT_C = aci,  
   ◊ /COL_S = [R],  
   ◊ /COL_C = [C]  
),

# End of the operands and keywords specific to the seismic analysis

◊ SHOCK = _F (  
   ◊ ENTITLE = int,  
   ◊ / / NODE_1 = no1,  
   ◊ / / GROUP_NO_1 = grno1,  
   ◊ / / NODE_2 = no2,  
   ◊ / / GROUP_NO_2 = grno2,  
   ◊ / / MESH = my,  
   ◊ / / GROUP_MA = grma,  
   ◊ / / OBSTACLE = obs,  
   ◊ / / NORM_OBST = NOR,  
   ◊ / / ORIG_OBST = ori,  
   ◊ / / GAME = / 1.,  
   ◊ / / ENG_VRIL = gamma,  
   ◊ / / DIST_1 = dist1,  
   ◊ / / DIST_2 = dist2,  
   ◊ / / UNDER_STRUCT_1 = ss1,  
   ◊ / / UNDER_STRUCT_2 = ss2,  
   ◊ / / REFERENCE_MAR = / 'TOTAL',  
   ◊ / / nom_sst,  
   ◊ / / RIGI_NOR = kn,  
   ◊ / / AMOR_NOR = / 0.,  
   ◊ / / RIGI_TAN = / 0.,  
   ◊ / / AMOR_TAN = / ct,  
   ◊ / / FRICTION = / 'NOT',  
   ◊ / / 'COULOMB = driven'  
# Operands specific to the taking into account of a transient speed  
# for the rotors (number of revolutions variable)
   ◊ VITESSE_VARIABLE = 'NOT',  
   ◊ / / 'OUI',  
   # if VITESSE_VARIABLE='OUI':  
      ◊ VITE_ROTA = vrota,  
      ◊ MATR_GYRO = gyro,  
      ◊ ACCE_ROTA = arota,  
      ◊ MATR_RIGY = gyro,  
   # if VITESSE_VARIABLE='NON':  
      ◊ VITE_ROTA = / 0.0,  
),

# 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.

Copyright 2017 EDF R&D - Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)
# Keyword specific to the taking into account of a crack in a rotor

◊ Rotor fissure = F (
  ◊ / NOEUD G = noq,
  ◊ / GROUP_NO G = grnog,
  ◊ / NOEUD D = nod,
  ◊ / GROUP_NO D = grnod,
  ◊ ANGL_INIT = 0.0,
  ◊ ANGL_ROTA = 0.0,
  ◊ K_PHI = kphi
  ◊ DK_DPHI = dkdphi
)

◊ VERI_SHOCK = F (  
  ◊ STOP_CRITERE = / 'YES',
  ◊ THRESHOLD = / 0.5, S,
)

◊ ANTI_SISM = F (  
  ◊ / NODE 1 = no1,
  ◊ / GROUP_NO 1 = grno1,
  ◊ / NODE 2 = no2,
  ◊ / GROUP_NO 2 = grno2,
  ◊ RIGI_K1 = / 0., kN,
  ◊ RIGI_K2 = / 0., kN,
  ◊ THRESHOLD_FX = / 0., Py,
  ◊ C = / 0., C,
  ◊ THEN_ALPHA = / 0., alpha,
  ◊ DX_MAX = / 1., dx,
)

◊ DIS_VISC = F (  
  ◊ / NOEUD 1 = no1,
  ◊ /GROUP_NO 1 = grno1,
  ◊ /NOEUD 2 = no2,
  ◊ /GROUP_NO 2 = grno2,
  ◊ K1 = k1,
  ◊ UNSUR_K1 = usk1,
  ◊ K2 = k2,
  ◊ UNSUR_K2 = usk2,
  ◊ K3 = k3,
  ◊ UNSUR_K3 = usk3,
  ◊ C = C,
  ◊ PUIS_ALPHA = /0.5
  ◊ /alpha,
)

◊ BUCKLING = F (  
  ◊ ITER_INTE_MAXI =/20
  ◊ RESI_INTE_RELA =/1.0E-06
)
Code_Aster

Titre : Opérateur DYNA_VIBRA
Responsable : ALARCON Albert

Date : 10/07/2015 Page : 7/9
Clé : U4.53.03 Révision : 8dac6d9676af

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.
Copyright 2017 EDF R&D - Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)

♣ / NODE 1 = no1,
  / GROUP NO 1 = grn01,
♣ / NODE 2 = no2,
  / GROUP NO 2 = grn02,
♣ OBSTACLE = obs,
♣ ORIG OBST = ori,
♣ NORM OBST = NOR,
♣ ENG VRIL = / 0,
  / gamma,
♣ GAME = / 1,
♣ DIST 1 = dist1,
♣ DIST 2 = dist2,
♣ REFERENCE MARK = / 'TOTAL',
[DEFECT] / nom_sst,
♣ RIGI NOR = kN,
♣ FNOR CRIT = film,
♣ FNOR POST_FL = fseuil,
♣ RIGI NOR POST_FL = k2,
)

♣ RELA_EFFO_DEPL = F (♣ NODE = Noah,
  ♣ SOUS STRUC = ss,
  ♣ NOM_CMP = nomcmp,
  ♣ RELATION = F,
)

♣ RELA_EFFO_QUICKLY = F (♣ NODE = Noah,
  ♣ SOUS STRUC = ss,
  ♣ NOM_CMP = nomcmp,
  ♣ RELATION = F,
)

# Keywords factors only associated with the coupling with code EDYOS
♣ COUPLAGE_EDYOS = F (♣ VITE ROTA = vrota,
  ♣ PAS TPS_EDYOS = dtedyos,
)

♣ STAGE_EDYOS = F (♣ UNIT = uled,
  ♣ / GROUP NO = grn0ed,
  ♣ NODE = noed,
  ♣ 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]
Structure of data produced:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{BASE_CALCUL} = &quot;\text{PHYS}&quot; ) and ( \text{TYPE_CALCUL} = &quot;\text{TRAN}&quot; )</td>
<td>dyna_trans</td>
</tr>
<tr>
<td>( \text{BASE_CALCUL} = &quot;\text{PHYS}&quot; ) and ( \text{TYPE_CALCUL} = &quot;\text{HARM}&quot; )</td>
<td>dyna_harmo</td>
</tr>
<tr>
<td>( \text{BASE_CALCUL} = &quot;\text{GENE}&quot; ) and ( \text{TYPE_CALCUL} = &quot;\text{HARM}&quot; )</td>
<td>harm_gene</td>
</tr>
<tr>
<td>( \text{AsType (MATR_RIGI)} = \text{matr_asse_pres_c} )</td>
<td>acou_harmo</td>
</tr>
<tr>
<td>( \text{BASE_CALCUL} = &quot;\text{GENE}&quot; ) and ( \text{TYPE_CALCUL} = &quot;\text{TRAN}&quot; )</td>
<td>tran_gene</td>
</tr>
</tbody>
</table>

**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.
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.