
Operator DYNA_VIBRA

1 Drank

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

- transients and harmonics
- on physical base and modal base

It is a macro-command which calls historical operators DYNA_TRAN_MODAL, DYNA_LINE_TRAN and DYNA_LINE_HARM following the choice that the user made on two key words:

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

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

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

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",
                    ),
# key Keys concerning the setting in data if harmonic or transitory computation on physical base :
    ◊ MODELS =mo , [model]
    ◊ CHAM_MATER =chmat , [cham_mater]
    ◊ CARA_ELEM =carac , [cara_elem]

# key Keys informing the assembled matrixes :
    ◆ MATR_MASS =ma , /[matr_asse_gene_R]
                    /[matr_asse_depl_R]
                    /[matr_asse_pres_C]

    ◆ MATR_RIGI =ri , /[matr_asse_gene_R]
                    /[matr_asse_depl_R]
                    /[matr_asse_pres_C]
                    /[matr_asse_depl_C]
                    /[matr_asse_gene_C]

    ◊ MATR_AMOR =am , /[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 computation with D-returning concept:
    ◊RESULTAT=harm , /[dyna_harmo]
                    /[harm_gene]

# introduction of modal damping:
    ◊ AMOR_MODAL =_F (
                    /AMOR_REDUIT =la , [l_R]
                    /LIST_AMOR =l_amor , [listr8]
                    /MODE_MECA = mode, [mode_meca]
                    / NB_MODE = nbmode, [I]
                    /9999, [DEFAULT ]
                    ),

# parameters for harmonic computation:
    ◆ /FREQ =lf , [l_R]
      /LIST_FREQ =cf , [listr8]

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

# parameters of the diagrams of integration

    ◊ SCHEMA_TEMPS =_F (
                    ◆ SCHEMA = ( | "NEWMARK", [DEFAULT]

```

```

| "EULER",
| "WILSON",
| "DEVOGE",
| "ADAPT_ORDRE1",
| "ADAPT_ORDRE2",
| "DIFF_CENTRE",
| "ITMI",
| "RUNGE_KUTTA_54",
| "RUNGE_KUTTA_32",
),
# key Keys only associated with diagram "NEWMARK":
  ◇ BETA      =/0.25,                [DEFAULT]
    /beta,                [R]
  ◇ GAMMA     =/0.5,                [DEFAULT]
    /gamma,                [R]
# key Keys only associated with diagram "ITMI":
  ◇BASE ELAS FLUI=      mix,      [melasflu]
  ◇NUMÉRIQUE_VITE_FLUI=      Nvitf,      [I]
  ◇ETAT_STAT      =      "NON",      [DEFAULT]
    /"OUI",
  ◇ PREC_DUREE=/1.E-2,                [DEFAULT]
    /prec,                [R]
  ◇ CHOC FLUI=/      "NON",      [DEFAULT]
    /"OUI",
  ◇ NB_MODE=Nmode,                [I]
  ◇ NB_MODE_FLUI=Nmodef,                [I]
  ◇ TS_REG_ÉTAB=tsimu,                [R]
# key Key only associated with diagram "WILSON":
  ◇ THETA      =/1.4,                [DEFAULT]
    /th,                [R]
# key Key only associated with diagrams "RUNGE_KUTTA_*":
  ◇ TOLERANCE  =/1.E-3,                [DEFAULT]
    /tol,                [R]
  ◆ INCREMENT =_F ( ◆ /LIST_INST      = litps,      [listr8]
    /PAS          = dt,                [R]
  ◇INST_INIT      =ti,                [R]
  ◇ /INST_FIN     = tf,                [R]
    /NUME_FIN     = nufin,            [I]
  ◇VERI_PAS      = "OUI",                [DEFAULT]
    /"NON",
# Operands specific to an integration by time step adaptive
  ◇VITE_MIN      =      "NORM",      [DEFAULT]
    /"MAXI",
  ◇COEF_MULT_PAS = 1.1,                [DEFAULT]
    /cmp,                [R]
  ◇COEF_DIVI_PAS = 1.33333334,        [DEFAULT]
    /cdp,                [R]
  ◇PAS_LIMI_RELA =/1.E-6,                [DEFAULT]
    /per,                [R]
  ◇ NB_POIN_PERIODE =/50,                [DEFAULT]
    /N,                [I]
  ◇NMAX_ITER_PAS = 16,                [DEFAULT]
    / N,                [I]
  ◇PAS_MAXI      =dtmax,                [R]
  ◇PAS_MINI      =dtmin,                [R] ),

```

```

    ◇ ETAT_INIT = _F ( ◇ / RESULTAT =res, [tran_gene]
        . If RESULTAT
            ◇ /INST_INIT =to , [R]
            /NUME_ORDR =no , [I]
            ◇ / CRITERE = ' RELATIF', [DEFAULT]
            ◇ accuracy = /1.E-06, [DEFAULT]
            /prec, [R]
            / CRITERE = "ABSOLU",
            ◆ accuracy = prec, [R]

            / | DEPL =do , /

[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]
        / CHARGE=chi , [char_meca]
        ◇NUMÉRIQUE_ORDRE =nmordr , [I]
        ◇/FONC_MULT =f , / [function]
        / [three-dimensions]

function]
        / [formula]
        / COEF_MULT =a , [R]
        /FONC_MULT_C =hci , / [fonction_C]
        / [formule_C]
        /COEF_MULT_C =aci , [C]
        /◇ACCE =ac , [function]
        / [three-dimensions]

function]
        / [formula]
        ◇VITE =vi , [function]
        / [three-dimensions]

function]
        / [formula]
        ◇DEPL =dp , [function]
        / [three-dimensions]

function]
        / [formula]
        ◇PHAS_DEG=/0 , , [DEFAULT]
        /phi , [R]
        ◇PUIS_PULS=/0 , , [DEFAULT]
        /ni , [Is]

# Operands and key keys specific to the seismic analysis
    ◇MULT_APPUI = "NON", [DEFAULT]
        / "OUI",
    ◇DIRECTION = (dx, Dy, dz, drx, dry, drz), [l_R]
    ◇ / NOEUD =lno , [l_noeud]
        / GROUP_NO =lgrno ,

[l_groupe_no]
        ◆◆ CORR_STAT = ' OUI'
        ◆ 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 ( ◆RESULTAT=resuforc , /
    [dyna_harmo] , / [harm_gene]
    / [dyna_trans]
    / [tran_gene]

    ◇/COEF_MULT=ai , [R]
    /COEF_MULT_C =aci , [C]
    ),

# End of the operands and key keys specific to the seismic analysis

◇CHOC = _F (
    ◇INTITULE =int , [l_Kn]

    /◆/NOEUD_1 =no1 , [node]
    /GROUP_NO_1 =grno1 , [group_no]
    ◇/NOEUD_2 =no2 , [node]
    /GROUP_NO_2 =grno2 , [group_no]
    /◆/MAILLE =ma , [mesh]
    /GROUP_MA =grma , [group_ma]

    ◆OBSTACLE =obs , [obstacle]
    ◆ NORM_OBST =nor , [listr8]
    ◇ORIG_OBST =ori , [listr8]
    ◇JEU = 1. , [DEFAULT]
    /clearance, [R]

    ◇ ANGL_VRIL =gamma , [R]

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

    ◇SOUS_STRUC_1 =ss1 , [K8]
    ◇SOUS_STRUC_2 =ss2 , [K8]
    ◇ coordinate = "GLOBAL",
    [DEFAULT] /nom_sst, [K8]

    ◇ RIGI_NOR =kn , [R]
    ◇ AMOR_NOR = 0. , [DEFAULT]
    /Cn, [R]
    ◇RIGI_TAN = 0. , [DEFAULT]
    /kt, [R]
    ◇AMOR_TAN = ct, [R]
    ◇ FROTTEMENT =
    /"NON" [DEFAULT]
    /"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 velocity
for the rotors (variable rotational speed)
◇VITESSE_VARIABLE = "NON", [DEFAULT]
/ "OUI",
# 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, [DEFAULT]
                        /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 , [DEFAULT]
    ◇ ANGL_ROTA =0.0 , [function]
    ◆K_PHI =kphi [function]
    ◆ DK_DPHI =dkdphi [function]
)

# Operands and key keys specific to the taking into account of a fluid blade
◇LAME_FLUIDE = "NON", [DEFAULT]
                        /"OUI",
# if LAME_FLUIDE=' OUI':
    ◇ALPHA = 0. , [DEFAULT]
                        /alpha, [R]
    ◇BETA = 0. , [DEFAULT]
                        /beta, [R]
    ◇CHI = 0. , [DEFAULT]
                        /chi, [R]
    ◇ DELTA = 0. , [DEFAULT]
                        /delta, [R]

# End of the operands and key keys specific to the taking into account of a fluid blade
),
◇ PARA_LAME_FLUI = _F (
    ◇ NMAX_ITER =/20 , [DEFAULT]
                        /niter, [I]
    ◇ RESI_RELA =/1.E-3 , [DEFAULT]
                        /residu , [R]
    ◇ LAMBDA =/10 , [DEFAULT]
                        /lambda , [R]
    ),
◇ VERI_CHOC = _F (
    ◇STOP_CRITERE = "OUI", [DEFAULT]
                        / "NON",
    ◇SEUIL =/0.5 , [DEFAULT]
                        /s, [R]
    ),
◇ ANTI_SISM = _F (
    ◆/NOEUD_1=no1 , [node]
        /GROUP_NO_1 =grno1 , [group_no]
    ◆/NOEUD_2=no2 , [node]
        /GROUP_NO_2 =grno2 , [group_no]
    ◇ RIGI_K1 = 0. , [DEFAULT]
                        /kN, [R]
    ◇ RIGI_K2 = 0. , [DEFAULT]
                        /kN, [R]
    ◇SEUIL_FX = 0. , [DEFAULT]
                        /Py, [R]
    ◇C = 0. , [DEFAULT]
                        /C, [R]
    ◇PUIS_ALPHA = 0. , [DEFAULT]
                        /alpha, [R]
    ◇DX_MAX = 1. , [DEFAULT]
)
    
```

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.

```

), /dx, [R]
),
◇ FLAMBAGE =_F (
    ◇/NOEUD_1=no1 [node]
      /GROUP_NO_1 =grno1 ,
[group_no]
    ◇/NOEUD_2=no2 [node]
      /GROUP_NO_2 =grno2 ,
[group_no]
    ◇OBSTACLE = obs, [obstacle]
    ◇ORIG OBST = ori, [listr8]
    ◇ NORM OBST = NOR, [listr8]
    ◇ANGL_VRIL = 0, [DEFAULT]
      /gamma, [R]
    ◇JEU = 1. , [DEFAULT]
      /jeu, [R]
    ◇DIST_1 = dist1, [R]
    ◇ DIST_2 = dist2, [R]
    ◇ coordinate = /"GLOBAL", [R]
[DEFAULT]
      /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 (
    ◇ NOEUD =noe [node]
    ◇SOUS_STRUC =ss [K8]
    ◇NOM_CMP =nomcmp [K8]
    ◇RELATION =f, [function]
),
◇RELA_TRANSIS =_F (
    ◇ NOEUD =noe [node]
    ◇SOUS_STRUC =ss [K8]
    ◇NOM_CMP =nomcmp [K8]
    ◇RELATION =f, [function]
),
◇RELA_EFFO_VITE =_F (
    ◇ NOEUD =noe [node]
    ◇SOUS_STRUC =ss [K8]
    ◇NOM_CMP=nomcmp [K8]
    ◇RELATION =f, [function]
),
# key keys factors only associated with the coupling with code EDYOS
◇COUPLAGE_EDYOS=_F (
    ◇ VITE_ROTA =vrota [R]
    ◇PAS_TPS_EDYOS =dtedyos [R]
),
◇PALIER_EDYOS=_F (
    ◇ /UNITE =uled [I]
    ◇ /GROUP_NO =grnoed [group_no]
    ◇ /NOEUD =noed [node]
    ◇ TYPE_EDYOS = "PAPANL",
      / "PAFINL",
      / "PACONL",
      / "PAHYNL",
),
# End key keys factors associated only with the coupling with code EDYOS

```

key Keys concerning the setting in data if transient computation on physical base

◇ ENERGIE =_F ()

End of the key keys concerning the setting in data if transient computation on physical base

```

◇ ARCHIVAGE =_F ( ◇ /LIST_INST = list [listr8]
                  /INST = in [R]
                  /PAS_ARCH = ipa [I]
◇/ CRITERE = ' RELATIF', [DEFAULT]
◇ accuracy = /1.E-06, [ DEFAULT]
                  /prec, [R]
                  / CRITERE = "ABSOLU",
                  ◆ accuracy = prec, [R]
                  ),
◇ solver = _F (see [U4.50.01]),
◇INFO=/1 ,
[DEFAULT] /2 ,
◇IMPRESSION =_F (
◇/TOUT = "OUI", [DEFAULT]
/NIVEAU = | ' DEPL_LOC',
          | ' VITE_LOC',
          | ' FORC_LOC',
          | ' TAUX_CHOC',
◇INST_INIT = Ti, [R]
◇INST_FIN = tf, [R]
          ),
◇TITER=titer , [l_Kn]
)
    
```

Data format 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 command DYNA_VIBRA

3.1 TYPE_CALCUL

This key word which makes it possible to make the choice between transient computation (TYPE_CALCUL=' TRAN') and harmonic computation (TYPE_CALCUL=' HARM') .

3.2 BASE_CALCUL

This key word makes it possible to make the choice between a computation on physical base (BASE_CALCUL=' PHYS') and a computation on modal base (BASE_CALCUL=' GENE') .

4 References towards the description of the other key words and operands

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

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

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