Operator MODI_MODELE

1 Goal

This operator allows to redefine the mode of distribution of the finite elements of a model for parallel calculation.

The partition of the finite elements is stored in SD_MODELE (preserved on the total basis). When one is in continuation, that implies to continue with the same number of processors. What is not inevitably desirable. To circumvent this difficulty, the order MODI_MODELE allows to redefine the partition of the model.

Modify the structure of data of the type model (D-entering operator).

2 Syntax

```
Mo [model] = MODI_MODELE  (

    ♦  reuse     =  Mo,
    ♦  MODEL     =  Mo,                                        [model]

    ◊  PARTITION =  _F (

          ◊ PARALLELISM =  /'GROUP_ELEM'                        [DEFECT]
                           /'MAIL_CONTIGU'
                                 ◊ CHARGE_PROC0_MA =/100  [DEFECT]
                                                   / pct
                           /'MAIL_DISPERSE'
                                 ◊ CHARGE_PROC0_MA =/100  [DEFECT]
                                                   / pct
                           /'SOUS_DOMAINE'
                                 ♦ PARTITION       =  share  [sd_partit]
                                 ◊ CHARGE_PROC0_SD =/0       [DEFECT]
                                                   / I
                           /'CENTRALIZES'

           )
     )
```

3 Operands

3.1 Operand MODEL

♦ reuse = Mo,
♦ MODEL = Mo

Name of the model which one seeks to modify (D-entering operator).

3.2 Keyword PARTITION

◊ PARTITION

This keyword as well as the operand PARALLELISM are described in the documentation of AFFE_MODELE [U4.41.01].

Notice important:

It is advised to begin the command files of the type CONTINUATION by MODI_MODELE (reuse=MO, MODELE=MO). This order will create a new partition adapted to the number of processors available.

4 Example

This example illustrates several modifications of the mode of partition of the model (extracted from mumps05a):

```
BEGINNING ()
…
# PARALLELISM CENTRALIZES (only the solvor Mumps will be treated in
parallel)
MODI_MODELE (reuse=MO, MODELE=MO,
             PARTITION=_F (PARALLELISME=' CENTRALISE'))

MECAC=MECA_STATIQUE (MODELE=MO,
                     SOLVEUR=_F (METHODE=' MUMPS',
                                 …
                                 )
                     …
                     )
…
# DISTRIBUTION OF MESHS DISPERSEES, AUTOMATIC BALANCING OF LOAD
MODI_MODELE (reuse=MO, MODELE=MO,
             PARTITION=_F (PARALLELISME=' MAIL_DISPERSE',
                           CHARGE_PROC0_MA=0))

MECAD1=MECA_STATIQUE (MODELE=MO,
                      SOLVEUR=_F (METHODE=' MUMPS',
                                  …
                                  )
                      …
               )
…
# DISTRIBUTION OF MESHS DISPERSEES, BALANCING OF LOAD Forced FOR
# TO RELIEVE PROCESSOR 0
MODI_MODELE (reuse=MO, MODELE=MO,
             PARTITION=_F (PARALLELISME=' MAIL_DISPERSE',
                           CHARGE_PROC0_MA=70))

MECAD2=MECA_STATIQUE (MODELE=MO,
                      SOLVEUR=_F (METHODE=' MUMPS',
                                  …
                                  )
                      …
               )
…
# PARALLELISM BY SOUS-DOMAINES, BALANCING OF LOAD FORCES TO RELIEVE # THE
PROC 0

sdpart = DEFI_PARTITION (MODELE=MO,…)

MODI_MODELE (reuse=MO, MODELE=MO,
             PARTITION=_F (PARALLELISME=' SOUS_DOMAINE',
                           PARTITION = sdpart,
                           CHARGE_PROC0_SD=2))

MECAD2=MECA_STATIQUE (MODELE=MO,
                      SOLVEUR=_F (METHODE=' MUMPS',
                                  …  )
                      …  )
```

```
...
END ()
```