# Keyword SOLVEUR

## 1 Goal

keyword factor SOLVEUR is common to all the orders which solve **linear systems of equations** (STAT_NON_LINE, CALC_MODES,...). To solve these systems of equations, one uses called particular algorithms "linear solveurs". The keyword SOLVEUR allows to choose the linear solvor to use among two categories: direct solveurs and iterative solveurs.

Among the solveurs directS, one has the classical algorithm of "Gauss" (METHODE=' LDLT'), of a factorization multifrontale ('MULT_FRONT') and of library MUMPS ('MUMPS').
For the iterative solveurs, it is possible to call on a combined gradient ('GCPC') or with certain algorithms of the PETSc library ('PETSC').

Only MULT_FRONT, MUMPS and PETSC are **paralleled**. The first in shared memory (OpenMP), two others mainly in distributed memory (MPI). Their parallel diagrams can even be chained (PETSc + MUMPS as a preconditionnor) or draw part of a second level from parallelism (OpenMP in MUMPS and library BLAS).
On the other hand, some is their level of parallelism, all the solveurs remain compatible with a parallel treatment of elementary calculations and assemblies.

In addition, only the three direct solveurs are compatible with modal calculation, the studies of buckling and certain procedures of refinement of step of time into nonlinear.

**For more details and advices** on the employment of the linear solveurs one will be able to consult the specific notes of use [U2.08.03]/[U2.08.06].

# Contents

# 2 Syntax

```
◊  SOLVEUR = _F (

# Parameter common to all the solveurs (direct, iterative)
    ◊ ELIM_LAGR=          /  'NOT',                          [DEFECT]
                          /  'YES',
                          /  'LAGR2',            (for MUMPS only)
# Parameters common to the direct solveurs ('MULT_FRONT', 'LDLT', 'MUMPS' )
    ◊  NPREC=             /  8,                              [DEFECT]
                          /  nprec                           [I]
    ◊  STOP_SINGULIER=    /  'YES',                          [DEFECT]
                          /  'NOT'

direct #Solvor "interns" of multifrontal type: cf. §3.3.
    /  METHODE=' MULT_FRONT'
            ◊  RENUM=     /  'MONGREL',                        [DEFECT]
                          /  'MANDELEVIUM',
                          /  'MDA'


direct #Solvor of multifrontal type based on library MUMPS: cf §3.5.
    /  METHODE=' MUMPS',                                       [DEFECT]
            ◊  TYPE_RESOL=/  'CAR',                  [DEFECT]
                          /  'NONSYM',
                          /  'SYMGEN',
                          /  'SYMDEF'
            ◊  PCENT_PIVOT=/ 20,                             [DEFECT]
                          /  pcent                           [R]
            ◊  RESI_RELA= /  -1.0,  (not flax. /modal)  [DEFECT]
                          /  +1.e-6, (linear)           [DEFECT]
                          /  resi                            [R]
            ◊  FILTRAGE_MATRICE= / -1.d0,                    [DEFECT]
                          /  filtma        (perimeter limited cf. §3.5)
            ◊  MIXER_PRECISION=  / 'ORI',    (perimeter limited cf. §3.5)
                          /  'NOT'                           [DEFECT]
            ◊  PRETRAITEMENTS=   / 'WITHOUT',
                          /  'CAR'                           [DEFECT]
            ◊  RENUM=     /  'CAR',                  [DEFECT]
                          /  'AMD',
                          /  'MFA',
                          /  'QAMD',
                          /  'PORD',
                          /  'MONGREL',
                          /  'PARMETIS' (only with version MPI)
                          /  'SCOTCH TAPE'
                          /  'PTSCOTCH' (only with version MPI)
            ◊  POSTTRAITEMENTS=  / 'WITHOUT',
                          /  'FORCE',
                          /  'MINI',
                          /  'CAR'                           [DEFECT]
            ◊  MATR_DISTRIBUEE=  / 'YES',    (perimeter limited cf §3.5)
                          /  'NOT'                           [DEFECT]
            ◊  GESTION_MEMOIRE=  / 'CAR',                [DEFECT]
```

```
                                    /   'IN_CORE',
                                    /   'OUT_OF_CORE',
                                    /   'EVAL'
```

*#Parameters limited to certain versions of MUMPS (cf §3.5)*

```
              ◊  ACCELERATION     =     /'CAR',                  [DEFECT]
                                    /   /'FR',
                                    /   /'FR+',
                                    /   /'FR++',
                                    /   /'LR',
                                    /   /'LR+'
                                    /   /'LR++',

              ◊  LOW_RANK_SEUIL=    /   0.0                      [DEFECT]
                                    /   lr_seuil                 [R]

              ◊  REDUCTION_MPI=     /   1,                       [DEFECT]
                                    /   redmpi                   [I]
```

*# iterative Solveur "interns" of type GCPC prepacked by Incomplete Cholesky $ILU(k)$ or by one factorized single precision (via MUMPS). Cf. § 3.6.*

```
    /   METHODE=' GCPC',
              ◊  /  PRE_COND=           /   'LDLT_INC',          [DEFECT]
                    ◊  NIVE_REMPLISSAGE=    /   0,               [DEFECT]
                                            /   niv              [I]

                 /  PRE_COND=           /   'LDLT_SP',
                                        /   'LDLT_DP',

                    ◊  PCENT_PIVOT =        /   20,              [DEFECT]
                                            /   pcent            [I]
                    ◊  REAC_PRECOND=        /   30,              [DEFECT]
                                            /   reactionary          [I]
                    ◊  GESTION_MEMOIRE=     /   'CAR',           [DEFECT]
                                            /   'IN_CORE',
                    ◊  RENUM=               /   'WITHOUT',
[DEFECT]
                                            /   'PARMETIS',
                                            /   'MONGREL',
                    ◊  LOW_RANK_SEUIL=      /   1.E-08,              [DEFECT]
                                            /   lr_seuil,            [R]

              ◊  NMAX_ITER=             /   0,                   [DEFECT]
                                        /   niter                [I]
              ◊  RESI_RELA=             /   10-6,                [DEFECT]
                                        /   resi                 [R]
```

*iterative #Solveurs resulting of the external library PETSc: cf § 3.7.*

```
    /   METHODE=' PETSC',
              ◊  ALGORITHME=            /   'FGMRES',            [DEFECT]
                                        /   'GMRES',
                                        /   'GMRES_LMP',
                                        /   'CG',
                                        /   'CR',
                                        /   'GCR'
              ◊  /  PRE_COND=           /   'LDLT_SP',           [DEFECT]
                                        /   'LDLT_DP',
```

```
                    ◊  PCENT_PIVOT =          /  20,           [DEFECT]
                                              /  pcent         [I]
                    ◊  REAC_PRECOND=          /  30,           [DEFECT]
                                              /  reactionary         [I]
                    ◊  GESTION_MEMOIRE=       /  'CAR',    [DEFECT]
                                              /  'IN_CORE',
                    ◊  RENUM=                 /  'WITHOUT',       [DEFECT]
                                              /  'PARMETIS',
                                              /  'MONGREL',
                    ◊  LOW_RANK_SEUIL=        /  1.E-08,       [DEFECT]
                                              /  lr_seuil,     [R]


                  /  PRE_COND=           /  'LDLT_INC',
                    ◊  NIVE_REMPLISSAGE=      /  0,            [DEFECT]
                                              /  niv           [I]
                    ◊  REMPLISSAGE=           /  1.0,          [DEFECT]
                                              /  rem           [R]


                  /  PRE_COND=           /  'Ml',
                                         /  'BOOMER',
                                         /  'GAMG',
                                         /  'BLOC_LAGR'


                  /  PRE_COND=           /  'JACOBI',
                                         /  'SOR',
                                         /  'WITHOUT'


                  /  PRE_COND=           /  'FIELDSPLIT',
                    ♦  NOM_CMP=          /  (),                [listK]
                    ♦  PARTITION_CMP=       (),                [listI]


                ◊  NMAX_ITER=            /  0,                 [DEFECT]
                                         /  niter              [I]
                ◊  RESI_RELA=            /  10^-6,             [DEFECT]
                                         /  resi               [R]
                ◊  MATR_DISTRIBUEE=      /  'YES',     (perimeter limited cf §3.7)
                                         /  'NOT'              [DEFECT]


                ◊  OPTION_PETSC=         /  '',                [K]
    ),
```

# 3 Operands

## 3.1 Operand METHOD

This keyword makes it possible to choose the method of resolution of the linear systems:

**#Solveurs**
**direct**

/'MULT_FRONT' **Direct Solvor of multifrontale type** (without swivelling during factorization). This method is paralleled in shared memory (OpenMP) and can be carried out on several processors (*via* small the Astk interface Options/Options of launching/ncpus).

/'LDLT' **Direct Solvor with factorization of Crout** by blocks (without swivelling). This solvor is paginated, it can thus be carried out with little memory.

/'MUMPS' [**DEFECT**] **Direct Solvor of multifrontale type** with swivelling. This solvor calls the library MUMPS developed by CERFACS/CNRS/ENS Lyon/INPT/INRIA/Université of Bordeaux.
Allows to treat the models leading to positive nondefinite matrices (except boundary conditions). For example, "mixed" elements having degrees of freedom of type "Lagrange" (incompressible elements, etc).
This method is paralleled mainly in distributed memory (MPI) but some of its stages can also profit from a parallelism in shared memory (OpenMP). It can be carried out on several hearts themselves possibly distributed on several nodes.
MUMPS is provided with external rénuméroteurs in 64 bits. That makes it possible to extend the perimeter of use of code_aster to the very large models finite elements (> $10^7$ degrees of freedom). Attention however, the resolution of the linear systems associated with these models requires to have great resources of calculation and to activate the various levels of parallelism even one of the options of acceleration.

**iterative**
**#Solveurs**

/'GCPC' **Iterative Solvor of standard gradient combined with pre-packaging** by an incomplete factorization with *K* levels or complete in single precision.

/'PETSC' **Iterative Solveurs resulting from the PETSc library** (*National Argonne Laboratory*) with various préconditionneurs.
This method is paralleled in distributed memory (MPI) and can be carried out on several processors.
Caution: solveurs PETSC and MUMPS being incompatible into sequential, PETSC is not available in the sequential versions of code_aster. To use PETSC, it is thus necessary to launch a parallel version of code_aster (since it is necessary to requesting one processor).

**The Council:**

**The method by default is MUMPS.** It allows, at the same time, fully to profit from the savings of time and memory which gets it **parallelism,** and, to solve **numerically difficult problems** (X-FEM, incompressibility, THM…).
Pour to solve a problem more effectively of **big size** (> $10^6$ degrees of freedom), one can resort to compressions 'low-rank' of MUMPS (cf keywords ACCELERATION/LOW_RANK_SEUIL ) or, if the functional perimeter allows it, with the iterative solveurs PETSC or GCPC.

**For more details and advices** on the employment of the linear solveurs one will be able to consult Lbe noteS of specific use [U2.08.03] and [U2.08.06].

## 3.2 Parameters common to several solveurs

```
◊   NPREC =            /  nprec
                       /  8      [DEFECT]

◊  STOP_SINGULIER  =   /  'YES'  [DEFECT]
                       /  'NOT'
```

These two parameters are common to all the direct linear solveurs (LDLT, MULT_FRONT, MUMPS).

They are used to control the course of digital factorization and the quality of the solution of the linear system. The digital factorization of a matrix can fail in two cases: problem of construction of factorized (structurally or numerically singular matrix) and digital detection of a singularity (solution of the unstable linear system).
The keyword NPREC and STOP_SINGULIER allow to fix the threshold of detection of the singularities and the behavior to be adopted in the event of failure during factorization.

nprec is used to gauge the process of detection as singularity of the matrix of the system to be solved. With LDLT and MULT_FRONT, one takes the absolute value of nprec, with MUMPS, one takes nprec because its sign has an importance: if $nprec<0$ , the detection of singularity is disabled, if not it is activated.
In all the cases, if the value nprec to zero one is left initializes it with the value by default (8).
By initializing this parameter to a rather low value (1 or 2) (respectively strong, for example, 20), the detection of singularity will very often start (respectively seldom).

For LDLT , MULT_FRONT :
When at the end of factorization, one notes that a diagonal term $d'$ became very small (compared to what it was before factorization $d$ ), it is that the matrix is (probably) almost singular. That is to say $n=\log\left|\frac{d}{d'}\right|$ , this report magnitude indicates that on an equation (at least) one lost $n$ significant figures.

If $n>nprec$ , one idiotsidère that the matrix is singular. If the user indicated:

| | |
|---|---|
| STOP_SINGULIER=' OUI' | The code stops then in error. |
| STOP_SINGULIER='NOT' | The execution continues after emission of one ALARM. The quality of the solution is then not guaranteed. This parameter setting is not advised. Into nonlinear, it is not inevitably too prejudicial with the quality of the results because those “are corrected” by the process of Newton. |

For MUMPS :

So for at least a pivot, the infinite standard of the line (or column) is lower than the threshold $10^{-nprec}$ then the matrix is regarded as singular.

One compares some aspects of the two types of criteria of detection of singularity in documentation [U2.08.03].

***Note:***

- *Any important loss of significant figures during a factorization is an indicator of a badly posed problem. Several causes are possible (nonexhaustive list): conditions with insufficient limits of blocking of the structure, the redundant linear relations, the very heterogeneous numerical data (too large terms of penalization)…*

*• For LDLT and MULT_FRONT, the detection of singularity is made all the time because it is far from expensive.*
*• Concerning MUMPS, a mechanism makes it possible to check the quality of the solution in addition (RESI_RELA). One thus left freedom disable this criterion (by choosing one nprec negative).*
*• By default, with the direct solvor MUMPS, one thus has a double quality control of the solution: into linear, RESI_RELA and NPREC, into nonlinear, the criterion of Newton and NPREC. It is possible to disconnect them, but it is not advised without valid reason.*

◊ ELIM_LAGR = 'NOT'/'YES'/'LAGR2'

This keyword makes it possible to eliminate the equations from Lagrange corresponding to the conditions dualized kinematics.

By default (except for MUMPS), these equations are not eliminated ('NOT').

Technique of elimination used for ELIM_LAGR= 'YES'is described in [R3.03.05].
To use ELIM_LAGR=' OUI' on several processors, it is imperative to have chosen as a preliminary DISTRIBUTION=' CENTRALISE' in AFFE_MODELE. The phase of elimination is then retorted on all the processors (without involving saving of time). The resolution of the resulting linear system is carried out then in parallel on all the processors (this time with a saving of time, which depends on the selected linear solvor).
One cannot use ELIM_LAGR=' OUI' with the direct linear solvor ' MULT_FRONT'.

In the case of the solvor MUMPS, a third value is possible: 'LAGR2'. The objective is then to remove the second equation of Lagrange, but to preserve the first.
The value 'LAGR2'is the defect for solvor MUMPS.

This parameter can be temporarily disabled by the code not to distort the calculation of the determinant of the matrix. This functionality is mainly necessary by the operators CALC_MODES with OPTION among ['NEAR', 'ADJUSTS', 'SEPARATE'] and INFO_MODE. The user is then informed of this automatic change of parameter setting *via* a message dedicated (only in INFO=2).

## 3.3 METHODE=' MULT_FRONT'

**Perimeter of use:**
Solvor robust, with to disadvise however for modelings requiring of the swivelling (mixed, incompressible finite elements…), for the matrices generalized with connections (operators ASSE_ELEM/MATR_SSD …) as on the large models finite elements (> $10^6$ degrees of freedom).
In these cases, use the method rather MUMPS (cf. § 3.5) or, into nonlinear, PETSC + PRE_COND=' LDLT_SP' (cf. § 3.7).

◊ RENUM =

This argument allows to number the nodes of the model pour to decrease the size of factorized (and thus consumption CPU and memory of the resolution):

| | |
|---|---|
| /'MONGREL' [**DEFECT**] | Method of classification based on an encased dissection. One uses the external product of the same name which is a world standard in the field. It is, in general, the most effective method (in time CPU and memory). |
| /'MANDELEVIUM' | ('Minimum Degree') this classification of the nodes minimizes the filling of the matrix during its factorization. |
| /'MDA' | ('Minimum Approximate Degree') this classification is in theory less optimal than 'MANDELEVIUM' with regard to the filling but it is more economic to calculate. It is however preferable with 'MANDELEVIUM' for the large models ( $\geq 50\,000$ degrees of freedom). |

**Note:**

- *In the case of generalized matrices[1] comprising constraints of connection, MULT_FRONT does not apply a renumerotation. This strategy is not prejudicial because these matrices are often quasi-full and with smalls. The choice of the renumerotor operated by the user is thus ignored. An informational message announces this case in the file message.*

1 Generated by the operator NUME_DDL_GENE.

### 3.4 METHODE=' LDLT'

**Perimeter of use:**
Universal Solvor but only on small sizes finite elements (< $10^5$ ddls) . Beyond that, the method is very slow.
With to disadvise for modelings requiring of the swivelling (mixed finite elements, incompressible…).

**Note:**

- *The matrix is systematically renumbered using the algorithm Reverse-Cuthill-Mackee. The user cannot modify this choice.*

## 3.5 METHODE=' MUMPS'

**Perimeter of use:**
Universal performing and robust Solvor. With to use on all types of problems and in particular the large models finite elements (>$10^6$ degrees of freedom), even the very large models (>$10^7$ degrees of freedom). Especially S'they mix modelingS, types of finite elements or if the associated linear systems require swivelling (finite elements mixed, incompressible, modeling X-FEM, condition with limitS dualizedS, connectionS between group of meshs, etc).
Into nonlinear, it is often preferable to use MUMPS as a preconditionnor, *via* the option PETSC + PRE_COND=' LDLT_SP'/ 'LDLT_DP'(cf. § 3.7).

LE solvor MUMPS, currently developed by CERFACS/CNRS/ENS Lyon/INPT/INRIA/Université of Bordeaux, is a direct solvor of paralleled multifrontal type (in MPI and OpenMP). It is robust because it makes it possible to swivel the lines and columns of the matrix during digital factorization.
Only public versions of MUMPS v5.2.1 and v5.2.1 are accepted in the coupling with *code_aster* .
In addition, their versions in access restricts MUMPS v5. 2 . 1 consortium/v5. 4 . 1 consortium are also accepted but only for EDF uses. They get an access, in phase lead compared to the public version, with exploratory features.
All are from now on compiled with external renuméroteurs 64 bits ((BY) MONGREL, (Pt) SCOTCH TAPE and PORD), which allows MUMPS, by activating them (cf keyword RENUM), to treat very large models finite elements (million meshs).
Attention however, the resolution of the linear systems associated with these models requires to have great resources of calculation and to activate the various levels of parallelism even one of the options of acceleration (cf keyword ACCELERATION).
For more information one will be able to consult it §4 of [U2.08.03] and [U2.08.06].

### 3.5.1 Functional parameters

◊ TYPE_RESOL =

CE keyword makes it possible to choose the type of resolution MUMPS:

| | |
|---|---|
| /'NONSYM' | Must be selected for the nonsymmetrical matrices. |
| /'SYMGEN' | Must be selected for the positive nondefinite symmetrical matrices. It is the case more the general in *code_aster* because of dualisation of the boundary conditions by coefficients of Lagrange. |
| /'SYMDEF' | Can be selected for the positive definite symmetrical matrices. There is no swivelling. The algorithm is faster and less expensive in memory. |
| /'CAR' [**DEFECT**] | The code will choose ' NONSYM ' for the nonsymmetrical matrices and ' SYMGEN ' for the symmetrical matrices. |

It is not interdict to choose 'NONSYM' for a symmetrical matrix. That will probably double the cost of calculation but this option gives to MUMPS more algorithmic possibilities (swivelling, *scaling*…). *A contrario*, it can be interesting, into nonlinear, to symmetrize its nonsymmetrical problem (cf. [U4.51.03], keyword MATR_RIGI_SYME). It is the same type of trick as for the parameters of relieving FILTRAGE_MATRICE and MIXER_PRECISION.

◊ RESI_RELA = / resi
/ 1.d-6 [ **DEFECT** ] into linear
/ -1.d0 [ **DEFECT** ] into nonlinear and modal calculation.

**This parameter is disabled by a negative value.** It is appealable in the operators who can need to control the quality of a complete resolution of linear system (thus not the burst operators carrying out factorizations just; For example TO FACTORIZE and INFO_MODE).

By specifying a strictly positive value with this keyword (for example $10^{-6}$), the user indicates that it wishes to test the validity of the solution of each linear system solved by MUMPS (into relative compared to the exact solution).
This careful approach is advised when the solution is not it not even corrected by another algorithmic process (algorithm of Newton, detection of singularity…) in short in the linear operators THER_LINEAIRE and MECA_STATIQUE. Into nonlinear or modal calculation, the criterion of detection of singularity and the correction of the including algorithm (Newton or modal solvor) are sufficient parapets. One can thus disconnect this process of control (it is what is made by default *via* the value -1) and this, more especially as it has a cost in considerable time and that this one is more important (into relative) in parallel and/or management memory with unloading of the large objects on disc (cf keyword GESTION_MEMOIRE). It is an additional functionality which the other direct solveurs do not offer of *code_aster*.

If the relative error (based on conditionings and the opposite errors of the treated linear system) on the solution estimated by MUMPS is higher than resi the code stops in ERREUR_FATALE, by specifying the nature of the problem and the values accused.
The activation of this keyword initiates also a process of iterative refinement (except if POSTTRAITEMENTS=' SANS') the objective is to improve the solution obtained. This posttraitement profits from a particular parameter setting (keyword POSTTRAITEMENTS). It is the solution resulting from this process of iterative improvement which is tested by RESI_RELA.

***Note:***

*• In the typical case where POSTTRAITEMENTS=' MINI' and RESI_RELA>0 , the estimate of the quality carried out by MUMPS only partial (is solely based on the opposite errors) and thus code_aster does not stop calculation if this value is higher than resi . This combination of keywords has interest only with INFO=2 to appraise qualities of the solutions.*

### 3.5.2 Parameters of relieving

```
◊   FILTRAGE_MATRICE=   /   filtma
                        /   -1.d0   [ DEFECT ]
◊   MIXER_PRECISION =   /   'YES'
                        /   'NOT'   [DEFECT]
```

These parameters are reserved for nonlinear quasi-static. **A negative value of `filtma` disable the functionality.**

These features make it possible “to release” the resolutions carried out with MUMPS in order to gain in performance. The idea is simple. Into nonlinear, the calculation of the tangent matrix can be sullied with error. That probably will slow down the process of Newton (of iteration count), but if the handling of this approximated matrix is less expensive, one can gain overall in time (less floating operations), in consumption memory (RAM even disc if the OOC is activated) and in band-width (effect hides, volume of I/O).
Thus, the activation of the functionality FILTRAGE_MATRICE, with a value of filtma>0, led *code_aster* has to provide to checking MUMPS only the matric terms

$$|\mathbf{K}_{ij}| > \texttt{filtma} \cdot \left( |\mathbf{K}_{ii}| + |\mathbf{K}_{jj}| \right)$$

The filter is thus based on a threshold relative compared to the absolute values of the diagonal terms corresponding.
While initializing MIXER_PRECISION with 'YES', one uses the version single precision of MUMPS by providing him a matrix *Aster* double precision (possibly filtered *Vl*has FILTRAGE_MATRICE). From where potentially of the profits in memory (often 50%) and time on the level of the resolution. However, this trick is really paying only if the tangent matrix is well conditioned ( $\eta(\mathbf{K}) < 10^{+6}$ ). If not the resolution of the linear system is too vague and the nonlinear algorithm is likely not to converge more.

***Note:***

*• These parameters of relieving of the resolutions of systems linear via MUMPS are in the line of those which already exist for the nonlinear solveurs (keyword NEWTON/REAC_ITER, MATRIX …). These families of parameters are clearly complementary and they can make it possible to gain tens of percent in consumption CPU and RAM. To spend a little time gauging them, on a first data file, can be paying when one must carry out many similar calculations thereafter.*
*• Up this idea was taken with the preconditionnor LDLT_SP of GCPC/PETSC.*
*• But to gain place memory without being likely to lose in precision of calculation, one can also be interested in the following elements: parallel calculation [U2.08.03], parameters GESTION_MEMOIRE, MATR_DISTRIBUEE* even *RENUM.*

### 3.5.3 Digital parameters

◊ PRETREATMENTS =

This keyword makes it possible to control the type of preprocessing to be operated with the system to improve its resolution (various strategies of balancing of the terms of the matrix and permutation of its lines and its columns):

| | |
|---|---|
| /'WITHOUT' | Pas de pretreatment. |
| /'CAR' [**DEFECT**] | MUMPS chooses the best combination of parameters according to the situation. |

◊ RENUM =

This keyword makes it possible to control the renumerotation and the order of elimination. The various tools proposed are not inevitably all available. That depends on the installation of MUMPS/ code_aster. These tools break up into two categories: tools " basic " dedicated to a use and provided with MUMPS ('AMD', 'MFA', 'QAMD', 'PORD'), and, "rich" libraries more and more "sophisticated" that it is necessary to install separately ('METIS/PARMETIS', 'SCOTCH/PTSCOTCH').

| | |
|---|---|
| /'AMD' | 'Minimum Approximate Dismantles' (Minimum Approximate Degree) |
| /'MFA' | 'Minimum Approximate Wire' (Approximate Minimum Filling) |
| /'QAMD' | Alternative of 'AMD' (automatic control of quasi-dense line) |
| /'PORD' | External tool of renumerotation distributed with MUMPS. |
| /'MONGREL' | Outil external of renumerotation (available also with METHODE=' MULT_FRONT'). It is the renumerotor of reference since the end of the Nineties. He universally is recognized and used. |
| /'BYMONGREL' | Parallel version MPI of the preceding tool. Only available with Lhas version MPI of code_aster. Cf [U2.08.03].<br>With to privilege on the very large models finite elements (million meshs). |
| /'SCOTCH TAPE' | Outil external of renumerotation which tends to supplant the tool of reference in the field (MONGREL). |
| /'PTSCOTCH' | Parallel version MPI of the preceding tool. Only available with Lhas version MPI of code_aster. Cf [U2.08.03].<br>With to privilege on the very large models finite elements (million meshs). |
| /'CAR'[**DEFECT**] | MUMPS chooses the best combination of parameters according to the problem and the packages available. If the user specifies a particular renumerotor and that this last is not available, the solvor chooses most adequate in the list of available and one ALARM is emitted. |

***Note:***

• *The choice of the renumerotor has a great importance on consumption memory and time of the linear solvor. If one seeks has to optimize/regulate the digital parameters related to the linear solvor, this parameter must be one of the first to be tested.*
• *Lbe renuméroteurs (BY) MONGREL, (PT) SCOTCH TAPE and PORD are compiled in 64 bits in order to allow MUMPS to solve linear systems several tens of million unknown factors.*

◊ POSTTRAITEMENTS =

This paramêtre is appealable in the operators who can need to control the quality of a complete resolution of linear system (thus not the burst operators carrying out factorizations just, e.g. TO FACTORIZE and INFO_MODE ). Its interest is less with the non-linear operators ( STAT_NON_LINE …) except in the event of difficult resolutions.

This keyword makes it possible to control two things:
– procedure of iterative refinement of which the objective is to improve quality of the solution (cf keyword RESI_RELA),
– the partial estimate (value 'MINIS') or supplements (other values) by MUMPS of the quality of the solution. The stop of calculation by *code_aster* will be carried out only if this estimate is thus complete (not with 'MINIS') and that if its value is higher than the criterion indicated in RESI_RELA.

| | |
|---|---|
| /'WITHOUT' | Desactivation. |
| /'FORCE' | **This parameter is used only if RESI_RELA is activated.** MUMPS carries out at least an iteration of iterative refinement because its criterion of stop is initialized with a very low value. The iteration count is limited to 10.<br>It carries out a complete diagnosis of the quality of the solution *via* the rather expensive calculation of estimate of conditionings and the opposite errors of the treated hollow linear system. |
| /'CAR' [**DEFECT**] | **This parameter is used only if RESI_RELA is activated.** MUMPS often carries out an iteration of iterative refinement. Its criterion of stop is close to the precision machine and the iteration count is limited to 4.<br>It carries out a complete diagnosis of the quality of the solution *via* the rather expensive calculation of estimate of conditionings and the opposite errors of the treated hollow linear system. |
| /'MINI' | MUMPS carries out two iterations of iterative refinement exactly. With use one notes that it is often sufficient to solve many the linear systems effectively. This option is to be privileged in the event of search for performance, without loss of precision, during the activation of compressions low-rank (cf keyword ACCELERATION).<br>By default here MUMPS does not carry out a diagnosis of the quality of the solution. One carries out only one partial diagnosis, *via* the only calculation (not very expensive) of the opposite errors, if RESI_RELA>0 (useful only for expert testimony). |

***Note:***
• *This process consumes mainly descent-increase whose overcost in time is still reasonable in In-Core sequential. On the other hand, it can be important into Out-Of-Core and parallel, even not very useful into nonlinear (the algorithm of Newton corrects).*
• *To limit any against-productive drift, with the values 'CAR'and'FORCE'the procedure of iterative refinement is attached in-house MUMPS: as soon as an iteration does not get a profit of at least a factor 5, the process stops. The iteration count generally noted (while posing INFO=2) is 1 or 2.*
• *On certain badly conditioned CAS-tests (for example perf001e), the forcing of this process made it possible to reach the precision desired (via the values 'FORCE'or'MINI').*

### 3.5.4 Parameters for management memory

To gain in RAM memory without changing linear solvor (and modeling or data-processing platform), several strategies are available (and often combinable). One lists them here by order of importance.

- **With constant digital precision and with savings of time of calculation:**
  LE parallelism (finely Options/mpi_ ** of Astk) coupled, or not, with the activation of the keyword MATR_DISTRIBUEE in distributed parallelism mode (by default mode);
  activation of the pretreatments (made by default, cf. PRETREATMENTS);
  The redistribution of parallelism enters level MPI and that OpenMP (cf. REDUCTION_MPI).

- **With constant digital precision but with, potentially, of the wastes of time of calculation:**
  L'activation clarifies faculties of unloading on disc MUMPS (cf. GESTION_MEMOIRE),
  LE change of renumerotor (cf keyword RENUM considering previously).

  It is also necessary to take care to fund an additional space reserved for the swivelling of reasonable size: keyword PCENT_PIVOT. **Often the values by default of these parameters (GESTION_MEMOIRE=' AUTO', RENUM=' AUTO' and PCENT_PIVOT=20) the best compromises get** to adapt this part of the parameter setting to the case.

- **By accepting a loss of precision** within a nonlinear process (e.g. STAT or DYNA_NON_LINE, CALC_MODES, …): all parameters of relieving related to the solvor (FILTRAGE_MATRICE, MIXER_PRECISION seen previously) even those related to the nonlinear process itself (elastic tangent matrix, space of projection in modal calculation…).

- By not using more MUMPS as direct solvor but as preconditionnor (cf. GCPC or PETSC with REAC_PRECOND=' LDLT_SP/DP' cf. §3.6/3.7).

For more further information one will be able to consult documentations [U2.08.03] (Note of use of the linear solveurs) and [U2.08.06] (Note of use of parallelism).

```
◊   PCENT_PIVOT =   /   pcent  [I]
                    /   20%    [ DEFECT ]
```

This keyword allows Cto hoisir a percentage of memory that MUMPS will hold at the beginning of calculation for the swivelling. Indeed, to factorize a matrix *Aster*, it is often preferable to permute two of its lines and/or its columns (cf [R6.02.03] §2.3.1). However the data-processing objects managing this swivelling are difficult to dimension *a priori*. Therefore the tool requires of the user a preliminary and arbitrary estimate of this additional space.
The value by default is of 20%. It corresponds to a reasonable number of swivellings which is sufficient for most calculations *Aster*. If for example MUMPS estimates at 100 the place necessary to a factorization without swivelling, it will allocate *in fine* 120 to manage calculation with swivelling. A value exceeding the 50% must remain exceptional.

Thereafter, if the memory capacity required by the swivellings proves more important, the allocated place will be insufficient and the code will require to increase this criterion. Two cases will arise then according to the selected type of management memory (*via* the keyword GESTION_MEMOIRE):

- If it is about a precise mode, 'IN-CORE'or'OUT_OF_CORE', calculation stops in ERREUR_FATALE and suggests various palliative solutions.
- If it is the automatic mode, 'CAR', calculation will continue and retenter to factorize with a value of PCENT_PIVOT doubled. Up to three attempts at this type will be carried out front, in the event of repeated failures, a stop in ERREUR_FATALE + proposal for various palliative solutions.

***Note:***

*• For the small problems (<1000 ddls), MUMPS can underestimate its requirements in pre-allowances for memory capacity. A great value of PCENT_PIVOT (>100) is then not surprising.*

*• Process of auto--training: if, in the process describes previously, one is brought to modify the value automatically of PCENT_PIVOT , it is this new value which is used until the end of the operator. It is supposed that the digital difficulty does not go to decrease and one thus preserves this value of swivelling in order to more not waste time in fallen through attempts at factorization.*
*• In mode ' CAR ', jointly with the doubling of the additional space of swivelling, one can also have to pass automatically in management Out-Of-Core memory MUMPS (as if one had parameterized explicitly GESTION_MEMOIRE=' OUT_OF_CORE' ). That occurs according to certain codes return MUMPS or to the third (and last) attempt.*

◊ GESTION_MEMOIRE=

This keyword makes it possible to choose the memory way of managing of external product MUMPS, even in last spring, certain objects managed directly by *code_aster*.
The first two modes are "without net": no correction of parameter setting will be operated "with the flight" in the event of problem. Contrary to the 3 ieme mode, the automatic mode, which will do everything (in some limiting!) so that calculation does not stumble for reasons of place memory. In particular, according to the memory which it will manage to release in addition, it will exploit the modes In-Core and Out-Of-Core of MUMPS even on space required for its swivelling (cf keyword PCENT_PIVOT ).

/'IN_CORE' One privileges to the maximum the speed of calculation. It is the option which requires the most memory, because here one allows MUMPS to preserve in RAM all the objects which it needs.

/'OUT_OF_CORE' One privileges to the maximum the economies in consumption memory. It is the option which requires less memory, because here one imposes on MUMPS of to discharge on disc its most cumbersome objects[2].

/'CAR' [**DEFECT**] One decides management memory automatically to impose on MUMPS (cf In-Core or Out-Of-Core precedent) according to the memory sizes available at this time precise of calculation. One activates also a mechanism of pre-allowance memory so that MUMPS can benefit from the maximum of the memory available (cf paragraph below). That makes it possible to limit the problems of late allowances to ensure the swivelling. Two mechanisms of autocorrection can be also set up so necessary (increase in PCENT_PIVOT, disconnection pre-allowances memory).

/'EVAL' Assistance with the calibration memory of calculation. One provides a synthetic posting (cf figure 3.1) of the resources memories required by calculation *code_aster* + MUMPS[3] according to the selected type of management: In-Core or Out-Of-Core. Then calculation stops in ERREUR_FATALE in order to allow the user to start again his calculation by choosing a parameter setting report being pressed on these elements.

```
*******************************************************************************
 - Size of the linear system: 500000

 - Minimal RAM memory consumed by code_aster                              : 200 Mo
 - Estimate of the Mumps memory with GESTION_MEMOIRE=' IN_CORE'             : 3500 Mo
 - Estimate of the Mumps memory with GESTION_MEMOIRE=' OUT_OF_CORE'         : 500 Mo
 - Estimate of the disk space for Mumps with GESTION_MEMOIRE=' OUT_OF_CORE': 2000 Mo

  ===> For this calculation, one thus needs a quantity of RAM memory at least of
       - 3500 Mo if GESTION_MEMOIRE=' IN_CORE',
```

2 Blocks of factorized (real or complex) managed by the current processor. The vectors of indices (whole) describing these blocks remain them in RAM.

3 Estimated the MUMPS can be slightly overestimated. They recapitulate the figures necessary for a use in "stand-alone" of product MUMPS on the problem resulting from *code_aster*. These estimates thus integrate, not only the objects which will need MUMPS to build its factorized, but also the preliminary objects of data storage (matrix, RHS) and one 30 contractual Mo for the achievable MUMPS.

```
        -  500 Mo if GESTION_MEMOIRE=' OUT_OF_CORE'.
 In case of doubt, use GESTION_MEMOIRE=' AUTO'.
*****************************************************************************
```

*Figure 3.1. _Posting in the file message in mode 'CAR'.*

Activation the Outone contributes to reduce the RAM memory required by processor, but that can slow down calculation (cost of the I/O RAM/disque). This overcost can be notable when many descent-increase are carried out (nonlinear e.g. calculation with much of step of time or iterations of Newton, search for many clean modes in modal calculation…). Because in these algorithmic stages, one spends as much time to handle the data (in RAM) that with going to seek them (on disc). This is all the more true as the disc is common to several hearts of calculation. For this reason, one privileges to the maximum the mode In-Core (especially in parallel).

In mode 'EVAL', the pre-estimates of consumed the memory are much faster and less expensive in memory than complete calculation. They can make it possible to gauge its study on local machine or an interactive node of a machine centralized before launching in batch mode the study itself.
On a purely anecdotic basis, this mode can be also used to coarsely test the setting in data and/or the achievable one used. If all functions until this evaluation it is rather good sign for later calculation!

In mode 'AUTO', one allows MUMPS "to be spread out in RAM" to gain in time and to limit the late requirements in memory related to the swivelling. MUMPS will thus be able to take all the memory which he considers necessary, even possibly beyond its estimated initial. That enables him to mitigate possible futures needs. With this intention, *code_aster* he provides an estimate of RAM available.
These pre-allowances often make it possible not to more have to adjust, often "with the wet finger", the parameter 'PCENT_PIVOT'. From where an unquestionable time-saver for the clarification of the studies.
In addition, in mode 'CAR', a calculation *code_aster*+MUMPS benefits thus really from all the memory available: Vmpeak is close to the figure parameterized in Astk.
On the other hand, in the two other modes ('IN_CORE' and 'OUT_OF_CORE'), MUMPS does not have the right "to be spread out" in RAM. It pre-does not allocate any additional space beyond its estimated memory initial. That makes it possible to preserve a sure operating process in the event of bad evaluation of the memory available[4].
Another mechanism also makes it possible to mitigate this kind of nuisance: if MUMPS seeks to allocate an object of size higher than the memory capacity really available, one retente a new factorization while allowing him more pre-to allocate of additional space. This strategy corrective, similar to that used for the parameter PCENT_PIVOT, is activated only with the mode 'CAR'.

***Note:***

*• In the standard modes ('IN_CORE'and'OUT_OF_CORE') code_aster discharge on disc largest objects [5] dependent on the linear system. And this, in order to leave with MUMPS a maximum of place in RAM memory. So thereafter MUMPS does not have sufficient place to allocate its data, an alarm is emitted and calculation continues its unfolding. According to the cases, calculation can be completed without encumbers but at the cost of a large overcost in time (swap system) or stop in ERREUR_FATALE . The user then sees himself proposing various alternatives of which the parameter setting in mode ' CAR '.*

*• In mode ' CAR ', if this released space is insufficient to make it possible MUMPS to fully function in In-Core, one discharges on disc all the remainder from objects JEVEUX déchargeables. Then, according to the memory capacity thus released, one activates the mode In-Core or Out-Of-Core of MUMPS or one stops in ERREUR_FATALE (+ advices).*

*• Massive unloadings of objects JEVEUX evoked previously can, in exceptional cases, to largely slow down the execution. That can arrive for example in the event of clogging of the access disc in parallel mode or when one discharges much from data (fields to the various steps of time, projected fields…). The solution can then be to occupy less processors by node, to consume less memory (to increase the number of processors, Out-Of-Core mode.) or to cut out its calculation in several stages.*

4 That can arrive on certain platforms (e.g. Clpaster-rock'n'rolls).
5 Matrix (MATR_ASSE), description of the unknown factors (NUME_DDL)…

*• In mode 'EVAL', the evaluation then the stop are carried out with the first matric factorization via MUMPS. Maybe, for example, in the phase of prediction for STAT_NON_LINE , or in the test of Sturm for CALC_MODES. It is often sufficient to have a good order of magnitude of the requirements in memory. For if required pushing back this evaluation, it is necessary to cut out its calculation and to use another linear solvor (for example ' MULT_FRONT ') for the operators whom one wishes to preserve.*

◊ MATR_DISTRIBUEE = / 'YES'
/ 'NOT' [**DEFECT**]

**This parameter is for the moment limited to the operators MECA_STATIQUE, STAT_NON_LINE and DYNA_NON_LINE and it active only in parallel is not distributed (AFFE_MODELE/PARTITION/PARALLELISME! = ' CENTRALISE').**
By activating this keyword, the storage of the assembled matrix is done in a way distributed on all the processors (one does not store any more useless values belonging to the other processors). That makes it possible to save memory in parallel without overcost in time, nor loss of precision (this keyword does not have any influence into sequential or parallel centralized).

### 3.5.5 Parameters to reduce time calculation

To reduce time calculation without changing linear solvor (and modeling or data-processing platform), several strategies are available (and combinable). One lists them here by order of importance.

- **With constant digital precision:**
  - parallelism MPI and/or OpenMP or, possibly, an optimized combination of the two (cf keyword REDUCTION_MPI);
  - the activation of various accelerations and/or compressions (cf keyword ACCELERATION);
  - change of renumerotor (keyword RENUM considering previously, often the choice makes by default is optimal);
  - the activation of the pretreatments (made by default, keyword PRETREATMENTS);
  - the use of MUMPS, one not as a direct solvor but as a preconditionnor. Perhaps very performing, especially into nonlinear, with the mutualisation on several steps of Newton of the same preconditionnor (cf. PETSC or GCPC with PRE_COND=' LDLT_SP/DP' §3.6/3.7)

- **By accepting a loss of precision,** what is often not prejudicial (the precision is sufficient) or compensated by a nonlinear process including (e.g. STAT or DYNA_NON_LINE, CALC_MODES…):
  - for problems of big size ($NR$ at least > 2.10$^6$ ddls), the activation of compressions low-rank (cf keywords ACCELERATION/LOW_RANK_SEUIL below);
  - parameters of relieving related to the solvor (FILTRAGE_MATRICE, MIXER_PRECISION seen previously) even those related to the nonlinear process itself (elastic tangent matrix, space of projection in modal calculation…);
  - the reduction of postprocessings (keyword POSTTRAITEMENTS).

For more further information one will be able to consult documentations [U2.08.03] (Note of use of the linear solveurs) and [U2.08.06] (Note of use of parallelism).

◊ ACCELERATION= / 'CAR' [DEFECT]
/ 'FR' (available for all the versions of MUMPS)
/ 'FR+' or 'FR++' (only with versions " consortium ")

/ 'LR' (available for all the versions of MUMPS)
/ 'LR+'or 'LR++' (only with versions " consortium ")

**type of acceleration:** often effective values, 'FR++'or'LR++'.

If ACCELERATION=' LR'/'LR+' or 'LR++' :
◊ LOW_RANK_SEUIL= / 0.0 [DEFECT]
/ lr_seuil [R]
**threshold of compression:** often effective values between $10^{-12}$ and $10^{-9}$ (cf explanations below).

These keywords define the type of acceleration implemented to reduce time calculation. These accelerations can significantly to reduce the computing time of large studies, and this, without restriction of perimeter of use and with little or not of impact on the precision, the robustness and the total behavior of simulation. Their availability depends on the versions of MUMPS coupled with code_aster.
They are especially interesting on problems of big sizes (NR at least > 2.10 [6] ddls). The profits noted on some studies code_aster vary from 20% to 80%. They increase with the size problem, its massive character and they are complementary to those gotten by parallelism and the renumerotor.
Various values of the parameter ACCELERATION are:

- The value ' CAR ' (catch by default) chooses the best parameter setting according to the version of MUMPS available, the treated case and the configuration of calculation. From MUMPS 5.4.1, for the public version 'AUTO'=' FR' and, for that consortium, 'AUTO'=' FR+'.
- The value ' FR ' the implementation of a standard resolution MUMPS allows (known as 'Full-Rank'), i.e. without compression 'low-rank' and "aggressive optimizations" of the internal options with MUMPS. It is that used by default starting from 5.4.1public MUMPS.
- The value ' FR+ ' activating but the implementation of a resolution without compression 'low-rank allows' "aggressive optimizations" of internal options MUMPS. It is that used by default starting from MUMPS 5.4.1consortium.
- The value ' FR++ ' add with 'FR+ 'a particular option of MUMPS which reduces the time of its stage D' analyzes.

- The value ' LR ' activate a resolution MUMPS with compression 'Low-Rank'. The compression ratio is fixed by the parameter provided by the keyword LOW_RANK_SEUIL. *Roughly speaking* , more this figure is large, for example $10^{-12}$ or $10^{-9}$ , more compression will be important and thus more the savings of time can be interesting. From a certain threshold of compression (thus "of approximation"), it is advised to activate, in complement, the procedure of iterative refinement (for example *via* POSTTRAITEMENTS=' MINI' ). This one makes it possible to find, with often a weak overcost, an error on the solution close to that which one would have obtained with standard calculation, ' FR '.
- Lbe valueS ' LR+'/ 'LR++' activeNT the same option as previously (' LR ') but while adding Lbe "aggressive optimizations" of ' FR+ ' (resp. ' FR++') as well as a methodology of more sophisticated compression.

For more details on these two keywords one will be able to consult [U2.08.03] § 7.2.7.

◊ **REDUCTION_MPI** = / redmpi [I]
/ 0 [**DEFECT**]
CE keyword is usable only on one parallel version MPI of the code. It is active only for one whole value strictly positive and compatible with the number of activated MPI (cf example below). It makes it possible to reduce, just on the level of MUMPS, the parallelism of first level (MPI) and of to redistribute on that of second level (OpenMP). At exit of MUMPS, calculation begins again the fixed parallel configuration have launching of code_aster.
Thus, with one calculation code_aster distributed on 4 process MPI[6] each one using 9 OpenMP[7]
(occupant for example 1 node of the cluster of calculation gaia, that is to say 4x9=36 Cœthe USSR):

1. the matrices and vectors are built in the elementary procedures of calculations and assemblies of code_aster and only in parallel MPI; those thus use only 4 C œ the USSR on the 36 allocated; the acceleration of these stages of calculation is thus to the maximum of **X4** ;

6 Parameter mpi_nbcpu tool for launching.
7 Resp. ncpus. This parameter setting is normally automatically determined by the tool for launching.

2. the resolutions of systems linear MUMPS entirely use they 48 C œ the USSR thanks to on the 2 levels of parallelism MPI and OpenMP. Their capacities of acceleration of those are grosso-modo similar and with effectiveness parallel of about 30 to 50 %. From where a potential acceleration of this stage of calculation enters **X15 and X24** ;

LE made to activate the keyword REDUCTION_MPI, goes to allow:
- to use more process MPI in order to better accelerate part 1;
- T out by preserving the level of acceleration of part 2 and without impacting its consumption total memory ;

Thus, with **REDUCTION_MPI=3** on the preceding example and while distributing this time calculation code_aster, not on 4 but on 4x3=12MPI, each one using 3 OpenMP (to continue to remain on 1 node with 36 Cœthe USSR):
- part S construction of matrices and vectors little wind to be accelerated this time until **X12** by occupying 12 C œ the USSR ;
- L part MUMPS has remains quite as effective while always occupying 36 C œ the USSR; while distributing this time calculation on 12/3=4 MPI, each one using 3x3=9 OpenMP. **One finds the point of initial operation ement quoted** .
  Part MUMPS being most consuming in memory, distribution initial 12MPI X 3 OpenMP is undoubtedly not possible or difficult on 1 only node. From where this need for reorganization of parallelism right on the level of MUMPS.

This redistribution of parallelism within MUMPS allows as follows:
- **To accelerate overall calculations whose laws of behavior are expensive** , and this, without impacting the other important part of calculation, that of resolution of system linear; N otamment in terms of consumption memory.
- **If required to reduce the peak total report of a calculation** (nap of the peak report of processes MPI) by reducing the number of processes MPI in the stage of calculation MUMPS; C being often that requiring the most memory to it-here.

***Note:***

*• This parameter is available only with the version consortium of MUMPS and it is not activable with the operator of modal calculation CALC_MODES . This one having its own redistribution of parallelism (2 levels MPI and 1 OpenMP level).*

## 3.6 METHODE=' GCPC'

**Perimeter of use:**
Real symmetrical problems except those requiring obligatorily a detection of singularity (modal calculation). Into nonlinear, if the problem is real nonsymmetrical, one can use this provided solvor that the matrix was made symmetrical .

◊ PRE_COND =

This keyword makes it possible to choose the method of pre-packaging:

| | |
|---|---|
| /'LDLT_INC' [**DEFECT**] | Decomposition **LDL**$^T$ incomplete (by level) of the assembled matrix |
| /'LDLT_SP' | Factorization single precision calculated by MUMPS |
| /'LDLT_DP' | Factorization double precision calculated by MUMPS |

The use of 'LDLT_SP' or 'LDLT_DP' is more expensive in CPU/RAM but more robust. These préconditionneurs are obtained by calculating with library MUMPS an approximate factorization of the matrix of the linear system. This factorization can be calculated into simple (LDLT_SP) or in double (LDLT_DP) precision. Its interest lies especially in its mutualisation (cf keyword REAC_PRECOND) during several resolutions if one seeks to solve problems of the multiple type second members (e.g. STAT_NON_LINE or thermomechanical chaining with MECA_STATIQUE).

◊ NIVE_REMPLISSAGE = / niv
/ 0 [**DEFECT**]

This parameter concerns only the preconditionnor LDLT_INC. The matrix of pre-packaging ( $\mathbf{P}$ ) used to accelerate the convergence of the combined gradient is obtained by factorizing in a more or less complete way the initial matrix ( $\mathbf{K}$ ).

More niv is large, more the matrix $\mathbf{P}$ is close to $\mathbf{K}^{-1}$ and thus more the combined gradient converges quickly (of iteration count). On the other hand, more niv is the great more storage of $\mathbf{P}$ becomes bulky (in memory and on disc) and more the iterations are expensive in CPU.

It is advised to use the value by default ( $niv=0$ ). If $niv=0$ does not allow the gradient combined to converge, one will test successively the values $niv=1,2,3\ldots$ .
In the same way if the iteration count of the combined gradient is considered to be too important, it is often beneficial to increase the level of filling.

◊ REAC_PRECOND = / reactionary
/ 30 [**DEFECT**]

This parameter relates to only itS preconditionnorS LDLT_SP and LDLT_DP
ItS preconditionnorS are much more expensive to build that the incomplete preconditionnor but itS are more robustS. To make it really competitive compared to the classical direct solveurs (MULT_FRONT or MUMPS double precision), it should be preserved during several successive resolutions. One thus exploits the “relative proximity” of these reiterated successive. With this intention, the parameter REAC_PRECOND condition the number of times where the same preconditionnor is kept whereas the matrix of the problem changed. As long as iterative method GCPC takes less reactionary iterations to converge, the unchanged preconditionnor is preserved; if it exceeds this number, the Re preconditionnor is reactualizedcalculating a factorization.

◊ PCENT_PIVOT = / pcent
/ 20 [**DEFECT**]
◊ GESTION_MEMOIRE = / 'CAR', [DEFECT]

```
                        /   'IN_CORE'
◊   LOW_RANK_SEUIL   =   /   1.E-08 ,              [DEFECT]
                        /    lr_seuil           [R]
◊  RENUM  =             /   'WITHOUT',             [DEFECT]
                        /   ' PARMETIS '
                        /   ' MONGREL ',
```

These parameters relate to only it S preconditionnorS LDLT_SP and LDLT_DP. The keyword ' LOW_RANK_SEUIL ' allows to regulate the precision of compression block-low-rank used by MUMPS. The larger this parameter is, the more severe compression is and the less faithful factorization is. They is the same keywords as for the solvor MUMPS, *cf*. §3.5.4.

```
◊   NMAX_ITER    =  /  niter
                    /  0   [DEFECT]
```

Maximum iteration count of the iterative algorithm of resolution. If $niter=0$ then the maximum number of iterations is calculated as follows:

$niter=nequ/2$ where $nequ$ is the number of equations of the system.

```
◊   RESI_RELA    =  /  resi
                    /  10^-6        [DEFECT]
```

Convergence criteria of the algorithm. It is a relative criterion on the not-prepacked residue:

$$\frac{\|\mathbf{r}_m\|}{\|\mathbf{f}\|}\leq resi$$

$\mathbf{r}_m$ is the residue not prepacked with the iteration $m$
$\mathbf{f}$ is the second member and the standard $\|\ \|$ is the usual euclidian norm.

**Note:**

• *When the preconditionnor is used LDLT_INC , the matrix is systematically renumbered using the algorithm Reverse-Cuthill-Mackee. The user cannot modify this choice.*

## 3.7 METHODE=' PETSC'

**Perimeter of use:**
All types of problèmes real except those requiring obligatorily a detection of singularity (modal calculation). With to use in priority on the nonlinear problems (with PRE_COND=' LDLT_SP') or on the problems "borders" (> $5.10^7$ degrees of freedom).

**Caution:** solveurs PETSC and MUMPS being incompatible into sequential, only MUMPS is available in the sequential versions of code_aster. To use PETSC, it is thus necessary always to launch a parallel version of code_aster (since it is necessary to requesting one processor).

◊ ALGORITHM =

Name of the iterative solveurs (of Krylov type) of PETSc accessible since *code_aster* :

| | |
|---|---|
| /'FGMRES' [**DEFECT**] | 'Flexible Minimal Generalised RESidual' |
| /'GMRES' | 'Minimal Generalised RESidual' |
| /'GMRES_LMP' | 'Minimal Generalised RESidual', with preconditionnor of second level to limited memory (Limited Memory To prepack) |
| /'CG' | Combined gradient |
| /'CR' | Combined residue |
| /'GCR' | 'Generalised Conjugate Residual' |

The method by default ensures the best relationship between robustness and cost of calculation.
Methods 'CG' and 'CR' are to be reserved for modelings leading to symmetrical matrices. In nonsymmetrical, in addition to 'GMRES', one can call on 'GCR' who treats unspecified matrices.
The algorithm ' GMRES_LMP' be pressed on the iterative solvor 'GMRES'. It must be obligatorily used with the preconditionnor of first level 'LDLT_SP'. Its use is interesting in a non-linear calculation: indeed, the preconditionnor of second level improves the pre-packaging of a system starting from spectral information resulting from the preceding linear resolutions (see [R6.01.02])

◊ PRE_COND =

Name of the accessible préconditionneurs of PETSc since *code_aster:*

| | |
|---|---|
| /'LDLT_INC' | Incomplete factorization by level |
| /'LDLT_SP' [ **DEFECT** ] | Factorization single precision calculated with MUMPS |
| /'LDLT_ D P' | Factorization double precision calculated with MUMPS |
| /'Ml' | Algebraic Multigrille " *multilevel* " (library ml) |
| /'BOOMER' | Algebraic Multigrille "BoomerAMG" (library HYPRE) |
| /'GAMG' | Algebraic Multigrille (PETSc library) |
| /'BLOC_LAGR' | Preconditionnor per blocks of the Lagrangian type Increased |
| /'JACOBI' | standard diagonal Pre-conditioner |
| /'SOR' | 'Successive Over Relieving' |
| /'FIELDSPLIT' | Preconditionnor per blocks (advanced functionality) |
| /'WITHOUT' | Pas de preconditionnor |

Only LDLT_SP, LDLT_DP, Ml, BOOMER and JACOBI have exactly same operation into sequential and parallel. Two others, 'LDLT_INC' and 'SOR', modify a little calculation by using local diagonal blocks with the processors. They are simpler to implement but less effective. 'WITHOUT' allows not to apply of preconditionnor what is of an interest only at the time of the clarification of a calculation.
The algebraic préconditionneurs multigrilles ML, BOOMER and GAMG have a very restricted perimeter of application:

- calculation without multipliers of Lagrange (to use AFFE_CHAR_CINE to impose the loadings),
- with a constant number of degrees of freedom per node.

They are nevertheless very effective in parallel. It will be noted that the pre-conditioner Ml be based during its algorithm on a random pulling, which can involve a slightly different convergence between two identical resolutions. These préconditionneurs multi-grids are to be used with the solveurs CG or GCR (which can function whereas CG fail).

preconditionnor 'BLOC_LAGR' is a preconditionnor per blocks designed for calculations with multipliers of Lagrange. It must be used with METHODE=' PETSC'.

S preconditionnorS 'LDLT_SP' and 'LDLT_DP' are *a priori* S more robustS, but also itS more expensive to build. However, and contrary to other préconditionneurs, itS NE are not rebuiltS with each linear resolution, which itS returnsent finally competitiveS (*cf.* keyword REAC_PRECOND). ItS preconditionnorS are to use with the solvor FGMRES by default (or CG or GCR if the matrix is symmetrical). It is preferable to avoid GMRES (or its symmetrical equivalent CR) combined to a preconditionnor in single precision (risk to obtain a vague solution, because the criterion of stop of the solvor is disturbed by the mixture of arithmetic).
In a non-linear calculation, one can finally use 'LDLT_SP' with the algorithm 'GMRES_LMP' : a preconditionnor of second-level (LMP) then improves the pre-packaging of a linear resolution starting from spectral information exits of the preceding linear resolutions (see [R6.01.02]).

preconditionnor 'FIELDSPLIT' offer a framework very general to define préconditionneurs per blocks. It is about a very powerful functionality which requires a very good knowledge of the iterative methods and bookstore PETSc. It is primarily used in research actions.

◊ NIVE_REMPLISSAGE = / niv
/ 0 [**DEFECT**]

This parameter concerns only the preconditionnor LDLT_INC.
Level of filling of the preconditionnor of Incomplete Cholesky.

◊ FILLING = / α
/ 1.0 [ **DEFECT** ]

This parameter concerns only the preconditionnor LDLT_INC.
Growth factor in the size of the preconditionnor according to the level of filling (cf §3.6). The reference is fixed at $niv=0$ for which $\alpha=1$. This parameter is taken into account only if PRE_COND=' LDLT_INC'. This figure makes it possible PETSc to coarsely envisage the size necessary to store the preconditionnor. If this estimate is too weak, PETSc increases the objects with the flight, but this operation is more expensive.

◊ REAC_PRECOND = / reactionary
/ 30 [**DEFECT**]

This parameter relates to only itS preconditionnorS LDLT_SP and LDLT_DP.

ItS preconditionnorS are much more expensive than the incomplete preconditionnor but itS are more robustS. ForS to make really competitiveS compared to the classical direct solveurs (MULT_FRONT or MUMPS double precision), one needs itS to preserve during several successive resolutions.
The parameter REAC_PRECOND condition the number of times where the same preconditionnor is kept whereas the matrix of the problem changed. As long as the iterative solvor (ALGORITHM) called by

PETSC takes less reactionary iterations to converge, the unchanged preconditionnor is preserved; if it exceeds this number, one reactualizes the preconditionnor by remaking a factorization single precision.

```
◊  PCENT_PIVOT   =  /  pcent
                   /  20  [DEFECT]
◊  GESTION_MEMOIRE =  /  'CAR',              [DEFECT]
                   /  'IN_CORE'
◊   LOW_RANK_SEUIL   =  /  1.E-08 ,             [DEFECT]
                   /   lr_seuil           [R]
◊  RENUM  =            /  'WITHOUT',            [DEFECT]
                   /  ' PARMETIS '
                   /  ' MONGREL ',
```

These parameters relate to only it S preconditionnor S LDLT_SP and LDLT_DP. The keyword ' LOW_RANK_SEUIL ' allows to regulate the precision of compression block-low-rank used by MUMPS. The larger this parameter is, the more severe compression is and the less faithful factorization is. They is the same keywords as for the solvor MUMPS, *cf*. §3.5.4.

```
◊   MATR_DISTRIBUEE =  /  'YES'
                   /  'NOT' [DEFECT]
```

**This parameter is for the moment limited to the operators MECA_STATIQUE, STAT_NON_LINE and DYNA_NON_LINE and it active only in parallel is not distributed (AFFE_MODELE/PARTITION/PARALLELISME! = ' CENTRALISE').**
By activating this keyword, the storage of the assembled matrix is done in a way distributed on all the processors (one does not store any more useless values belonging to the other processors). That makes it possible to save memory in parallel without overcost in time, nor loss of precision (this keyword does not have any influence into sequential or parallel centralized). It is it should be noted that it is recommended to use partitioning SOUS_DOMAINE in AFFE_MODELE in order to avoid potential problems of conditioning involved in the renumerotation of the assembled matrix.

```
◊   NMAX_ITER   =  /  niter
                   /  0   [DEFECT]
```

Maximum iteration count of the iterative algorithm of resolution. If $niter \leq 0$ , the maximum iteration count is fixed by default at 10000. This value is reduced to 100 if the preconditionnor is used LDLT_SP.

```
◊   RESI_RELA   =  /  resi
                   /  10-6    [DEFECT]
```

Convergence criteria of the algorithm. It is a relative criterion on the prepacked residue:

$$\frac{\|M^{-1}.\mathbf{r}_m\|}{\|M^{-1}.\mathbf{f}\|} \leq resi$$

$M^{-1}$ is the preconditionnor
$\mathbf{r}_m$ is the residue with the iteration $m$
$\mathbf{f}$ is the second member and the standard $\|\ \|$ is the usual euclidian norm.

***Note:***

1 *Convergence criteria for PETSC is evaluated differently from GCPC ;*
2 *When the preconditionnor is of poor quality (for example because of a bad conditioning of the problem), convergence criteria used by PETSC can cause bad solutions; this is why in the operators of linear calculation, an additional checking on the not-prepacked residue is carried out. The tolerance chosen for this additional criterion is* $\sqrt{resi}$ *;*

3 *The algorithm 'GCR' be based on a prepacking on the right and thus checks the convergence criteria normalizes some nonpackaged.*
4 *When the preconditionnor is used LDLT_INC , the matrix is systematically renumbered using the algorithm Reverse-Cuthill-Mackee. The user cannot modify this choice.*
5 The preconditionnor *LDLT_INC is incompatible with MATR_DISTRIBUEE=' OUI'.*

### 3.7.1 Keywords specific to the preconditionnor FIELDSPLIT

◊ NOM_CMP

List of the components of the fields which appear in modeling, for example NOM_CMP= ('DX', 'DY', 'DZ', 'NEAR') for a mixed modeling in displacement and pressure for an incompressible problem of elasticity.

◊ PARTITION_CMP

Way in which one will group the components of the fields. In the example NOM_CMP= ('DX', 'DY', 'DZ', 'NEAR'), one uses PARTITION_CMP=(3.1), which means that one will treat them separately three components of displacement and the component of pressure.

◊ OPTION_PETSC

Character string which defines the preconditionnor per blocks. This chain must be on a line Uscrew (without return to the line).

### 3.7.2 Keyword OPTION_PETSC

In addition to defining the preconditionnor precisely FIELDSPLIT (as evoked above), the keyword OPTION_PETSC also allows to pass to PETSc of the additional options, not-catches in load by the interface described here. This bookstore is indeed designed to be highly configurable with the execution what allows to realize this keyword. **It is about a advanced use of PETSc, to hold exclusively to the connoisseur at end of research.**

◊ OPTION_PETSC
Character string which allows to pass a set of additional options to PETSc

It is important to note that the last additional options with PETSc are preserved of its initialization at its closing, generally with the execution of the order END (). Thus if a user passes from the additional options in STAT_NON_LINE, they will be used for all the resolutions later even with other operators of resolution like DYNA_VIBRA.
For contouRner that, it possible to stop PETSc what causes to reset all the options. That is done using the orders:

```
from code_aster importation LinearAlgebra # importation of the LinearAlgebra module
LinearAlgebra.petscInitialize ()      # call to the method which initializes PETSc. It
                                      # can take as arguments of news
                                      # options
LinearAlgebra.petscFinalize ()            # call to the method which closes PETSc and
resets
                                      # options
```