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 three 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, PETSC are **paralleled**. The first in shared memory (OpenMP), others in distributed memory (MPI). But all the solveurs are compatible with a parallel treatment of elementary calculations and assemblies [U4.41.01].
In addition, only the three direct solveurs are compatible with modal calculation and the studies of buckling.

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

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)

Contents

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

2 Syntax

```
◊   SOLVEUR = _F (

# Parameter common to all the solveurs (direct, iterative)
    ◊   SYME=                          /   'NOT',                      [DEFECT]
                                       /   'YES',
    ◊   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',                                         [DEFECT]
            ◊   RENUM=                 /   'MONGREL',                      [DEFECT]
                                       /   'MANDELEVIUM',
                                       /   'MDA'

direct #Solvor "classical" of Gauss type: cf §3.4.
    /   METHODE=' LDLT',
            ◊   RENUM=                 /   'RCMK',                     [DEFECT]
                                       /   'WITHOUT'

direct #Solvor of multifrontal type based on library MUMPS: cf §3.5.
    /   METHODE=' MUMPS',
            ◊   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',
                                       /   'SCOTCH TAPE'
            ◊   POSTTRAITEMENTS=       /   'WITHOUT',
                                       /   'FORCE',
                                       /   'CAR'                       [DEFECT]
```

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

```
◊   MATR_DISTRIBUEE=    /   'YES',          (perimeter limited cf §3.5)
                        /   'NOT'                       [DEFECT]

◊   GESTION_MEMOIRE=    /   'CAR',                  [DEFECT]
                        /   'IN_CORE',
                        /   'OUT_OF_CORE',
                        /   'EVAL'
```

# 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

                 ◊   RENUM=                 /   'WITHOUT',
                                            /   'RCMK'        [DEFECT]

             /   PRE_COND=           /   'LDLT_SP',

                 ◊   PCENT_PIVOT =          /   20,           [DEFECT]
                                            /   pcent

                 ◊   REAC_PRECOND=          /   30,           [DEFECT]
                                            /   reactionary
                 ◊   RENUM=                 /   'WITHOUT',        [DEFECT]

         ◊   NMAX_ITER=              /   0,                   [DEFECT]
                                     /   niter                [I]

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

iterative #Solveurs based on the external library PETSc: Cf. § 3.7.

```
/   METHODE=' PETSC',
         ◊   ALGORITHME=             /   'GMRES',             [DEFECT]
                                     /   'CG',
                                     /   'CR',
                                     /   'GCR'

         ◊   /   PRE_COND=           /   'LDLT_SP',           [DEFECT]

                 ◊   PCENT_PIVOT =          /   20,           [DEFECT]
                                            /   pcent

                 ◊   REAC_PRECOND=          /   30,           [DEFECT]
                                            /   reactionary

                 ◊   RENUM=                 /   'WITHOUT',        [DEFECT]

             /   PRE_COND=           /   'LDLT_INC',

                 ◊   NIVE_REMPLISSAGE=      /   0,            [DEFECT]
                                            /   niv

                 ◊   REMPLISSAGE=           /   1.0,          [DEFECT]
                                            /   rem

                 ◊   RENUM=                 /   'WITHOUT',
                                            /   'RCMK'        [DEFECT]

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

                 ◊   RENUM=                 /   'WITHOUT',        [DEFECT]

             /   PRE_COND=           /   'JACOBI',
                                     /   'SOR',
```

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

```
                                      /   'WITHOUT'
                        ◊   RENUM=             /   'WITHOUT',        [DEFECT]
                                               /   'RCMK'
                    ◊   NMAX_ITER=            /   0,                    [DEFECT]
                                              /   niter                 [I]
                    ◊   RESI_RELA=            /   10-6,                 [DEFECT]
                                              /   resi                  [R]
                    ◊   MATR_DISTRIBUEE=    /   'YES',          (perimeter limited cf §3.7)
                                            /   'NOT'                   [DEFECT]

          ),
```

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

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' [**DEFECT**]	**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'	**Direct Solvor of multifrontale type** with swivelling. This solvor calls the library MUMPS developed by CERFACS/IRIT/INRIA/CNRS. Allows to treat the models leading to positive nondefinite matrices (except boundary conditions). For example, “mixed” elements having ddls of the type “Lagrange” (incompressible elements…). This method is paralleled in distributed memory (MPI) and can be carried out on several processors (*via* small the Astk interface `Options/Options of launching/mpi_nbcpu & mpi_nbnoeud`).

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 (*via* small the Astk interface `Options/Options of launching/mpi_nbcpu & mpi_nbnoeud`). Caution: solveurs `PETSC` and `MUMPS` being incompatible into sequential, one generally privileges `MUMPS`. To use `PETSC`, it is thus necessary often to launch a parallel version of *Code_Aster* (since it is necessary to requesting one processor).

The Council:

The method by default remains the internal multifrontale MULT_FRONT. But for fully profiting from the savings of time and memory which gets it **parallelism** or to solve one **numerically difficult problem** (X-FEM, incompressibility, THM), one recommends the use of **MUMPS**.
To solve a problem of **big size** (> 10^6 ddls), the solvor `PETSC` with its adjustments by default can allow to pass a calculation in spite of limitations memory of the machine of calculation.
For more details and advices on the employment of the linear solveurs one will be able to consult the specific note of use [U2.08.03].

3.2 Parameters common to several solveurs

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

This parameter is common to all the linear solveurs (direct, iterative). It is appealable only for the nonlinear operators.

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

If the matrix of the linear system $\mathbf{K}$ is not-symmetrical, the keyword `SYME=' OUI'` allows to symmetrize this matrix before the resolution of the system. The matrix is then replaced by $\mathbf{K}'=\frac{1}{2}\left(\mathbf{K}+\mathbf{K}^T\right)$.

Caution:

> *The interest of this keyword is to save time at the time of the resolution of the linear systems because the direct solveurs are generally more expensive in nonsymmetrical.*
> *The symmetrization of the matrix $\mathbf{K}$ conduit however to solve a different linear system. By doing this, it can degrade the convergence of the non-linear algorithms.*

```
◊    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 has 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.

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)

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): boundary conditions insufficient of blocking of the structure, linear relations redundant, 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 good 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]. Attention, this technique does not function in parallel.

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.

Historically, direct linear solveurs of *Code_Aster* (`'MULT_FRONT'` and `'LDLT'`) did not have algorithm of swivelling. To circumvent this problem, the taking into account of the limiting conditions by of Lagranges was modified by introducing doubles Lagranges at the cost of a overcost report and calculation. Like `MUMPS` have faculties of swivelling, this choice of dualisation of the limiting conditions can be called into question.
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`).

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

3.3 METHODE=' MULT_FRONT'

Perimeter of use:
Universal Solvor usable everywhere. With to disadvise for modelings requiring of the swivelling (mixed, incompressible finite elements…) like for the matrices generalized with connections (operators `ASSE_ELEM/MATR_SSD` ...).
In these cases, use the solvor rather `MUMPS` (cf. § 3.5).

◊ `RENUM =`

This argument allows to renumber 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`.

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)

3.4 METHODE=' LDLT'

Perimeter of use:
Universal Solvor (but very slow on large case). With to disadvise for modelings requiring of the swivelling (mixed finite elements, incompressible…).

◊ RENUM =

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

/'WITHOUT'	One keeps the initial order given in the file of grid.
/'RCMK' [**DEFECT**]	'Reverse Cuthill-MacKee', this algorithm of renumerotation is often effective to reduce the place necessary (in storage SKYLINE) assembled matrix and to reduce the time necessary to the factorization of the matrix.

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

3.5 METHODE=' MUMPS'

Perimeter of use:
Universal Solvor.

LE solvor MUMPS currently developed by CERFACS/CNRS/ENSEEIHT-INPT/INRIA (Copyright with §3 of [U2.08.03]) is a direct solvor of multifrontal type, paralleled (in MPI) and robust, because it makes it possible to swivel the lines and columns of the matrix during digital factorization.
Starting from the MUMPS v4.10.0, the perimeter of the product in *Code_Aster* is complete. For the previous versions (in fact only the 4.9.2 is temporarily accepted), calculations of determinant necessary to the operators CALC_MODES (with OPTION=‘SEPARATE’/‘ADJUSTS’) and INFO_MODE (method APM) are not possible. Calculation stops then in ERREUR_F by advising the rise of version.

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 parameter SYME). It is the same type of tricks 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).
If the relative error 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 of which the objective is to improve the solution obtained. This postprocessing profits from a particular parameter setting

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

(keyword POSTTRAITEMENTS). It is the solution resulting from this process of iterative improvement which is tested by RESI_RELA.

3.5.2 Digital parameters

```
◊   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

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

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 *VI*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.*
> • *But to gain place memory without being likely to lose in precision of calculation, one can also be interested in the elements according to: parallel calculation [U2.08.03], parameters GESTION_MEMOIRE, MATR_DISTRIBUEE even RENUM.*
> • *Up this idea was taken with the preconditionnor LDLT_SP of GCPC/PETSC.*

◊ 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 preprocessing.
/'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: the tools "frustrate" dedicated to a use and

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

provided with MUMPS (`'AMD'`, `'MFA'`, `'QAMD'`, `'PORD'`), and, "rich" libraries more and more "sophisticated" that it is necessary to install separately (`'MONGREL'`, `'SCOTCH TAPE'`).

/`'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.
/`'SCOTCH TAPE'`	Outil external of renumerotation which tends to supplant the tool of reference in the field (MONGREL).
/`'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 external tools of renumerotation have by way of broader features: partitioning of graphs and grids, organization of course of tasks… For example, in Code_Aster, MONGRELS and SCOTCH TAPE are also used in the operator of partitioning of grid `DEFI_PARTITION`.*
- *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.*
- *In particular, during parallel modal calculations, one sometimes observed of speed-UPS disappointing because of an inappropriate choice of renumerotor. In this case there one noted that the choice of a "sophisticated" renumerotor was against-performing. It is to better impose on MUMPS simple 'MFA'or'QAMD', rather than 'MONGREL' (often taken automatically in mode 'AUTO').*

◊ `POSTTRAITEMENTS =`

This parameter is used only if `RESI_RELA` is activated. 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; E.g. `TO FACTORIZE` and `INFO_MODE`).

This keyword makes it possible to control the procedure of iterative refinement of which the objective is to improve quality of the solution (cf keyword `RESI_RELA`).

/`'WITHOUT'`	Desactivation.
/`'FORCE'`	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.
/`'CAR'` [**DEFECT**]	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.

Note:

- *This process consumes mainly descent-increase whose cost in term of performances is weak in In-Core. On the other hand, they can be expensive into Out-Of-Core even not very useful into nonlinear (the algorithm of Newton corrects).*
- *To limit any against-productive drift, this process 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.*

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)

• On certain badly conditioned CAS-tests (for example perf001e), the forcing of this process made it possible to reach the desired precision.

3.5.3 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):

- **With constant digital precision and with savings of time of calculation:** parallelism (finely `Options/mpi_ ** of Astk`) coupled, or not, with the activation of the keyword `MATR_DISTRIBUEE` in distributed parallelism mode (by default mode).
- **With constant digital precision but with potentially of the wastes of time of calculation:** activation clarifies Out-Of-Core faculties of MUMPS (cf keyword `GESTION_MEMOIRE` Ci below) or 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 your 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`) even those related to the nonlinear process itself (elastic tangent matrix, space of projection in modal calculation…).

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
                    /  2 0%       [ 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

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

operator. It is supposed that the digital difficulty does not go to decrease and one thus preserve 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.

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

```
       -  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 discharges on disc the 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`)…

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

• 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).

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

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 solvor provided that one activates, as a preliminary, the keyword SYME .

◊ 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 *via* external tool MUMPS

This second approach is more expensive in CPU/RAM but more robust. 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.

◊ RENUM =

This argument makes it possible to renumber the nodes of the model to decrease the size of factorized (and thus consumption CPU and memory of the resolution):

/'WITHOUT'	One keeps the initial order given in the file of grid.
/'RCMK'	'Reverse Cuthill-MacKee', this algorithm of renumerotation is often effective to reduce the place necessary (in storage SKYLINE) assembled matrix and to reduce the time necessary to the factorization of the matrix.

The value by default of this keyword depends on the preconditionnor. Only LDLT_INC use a renumerotation RCMK by default.

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

This parameter concerns only the preconditionnor LDLT_SP .
This preconditionnor is much more expensive than the incomplete preconditionnor but it is more robust because nearer to the exact solution. 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, one reactualizes the preconditionnor by remaking a factorization single precision.

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

This parameter concerns only the preconditionnor `LDLT_SP` .
It is the same keyword as for the solvor `MUMPS`, *cf.* §3.5.3.

```
◊    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.

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

3.7 METHODE=' PETSC'

Perimeter of use:
All types of problèmes real except those requiring obligatorily a detection of singularity (modal calculation).

Caution: solveurs PETSC and MUMPS being incompatible into sequential, one generally privileges MUMPS. 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* :

/'GMRES' [**DEFECT**]	'Minimal Generalised RESidual'
/'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.

◊ PRE_COND =

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

/'LDLT_INC'	Incomplete factorization by level
/'LDLT_SP' [**DEFECT**]	Factorization single precision *via* external tool MUMPS
/'Ml'	Algebraic Multigrille " *multilevel* " (library ml)
/'BOOMER'	Algebraic Multigrille "BoomerAMG" (library HYPRE)
/'GAMG'	Algebraic Multigrille (PETSc library)
/'JACOBI'	standard diagonal Pre-conditioner
/'SOR'	'Successive Over Relieving'
/'WITHOUT'	Pas de preconditionnor

In parallel mode, only LDLT_SP, Ml, BOOMER and JACOBI offer a pre-packaging equivalent to the sequential mode. 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).

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

The preconditionnor `'LDLT_SP'` is *a priori* most robust but it is also most expensive. However, and contrary to other préconditionneurs, it is not rebuilt with each linear resolution, which makes it finally competitive (*cf.* keyword `REAC_PRECOND`).

```
◊   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.

```
◊    RENUM =   /   'WITHOUT'
               /   'RCMK'
```

Renumerotor of the type 'Transfers Cuthill-Mackee' (like `'LDLT'`) to limit the filling of factorized incomplete.
The value by default of this keyword depends on the type of preconditionnor used. Only the preconditioner `LDLT_INC` carry out a renumerotation by default (`RCMK`).

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

This parameter concerns only the preconditionnor `LDLT_SP`.
This preconditionnor is much more expensive than the incomplete preconditionnor but it is more robust because nearer to the exact solution. 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.
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]
```

This parameter concerns only the preconditionnor `LDLT_SP` .
It is the same keyword as for the solvor `MUMPS`, *cf.* §3.5.3.

```
◊    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

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

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
                     /   1   [DEFECT]
```

Maximum iteration count of the iterative algorithm of resolution. If $niter \leq 0$ then it is fixed automatically by PETSc (10^5).

```
◊    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 pre-conditioner
$\mathbf{r}_m$ is the residue with the iteration m
$\mathbf{f}$ is the second member and the standard $\|\ \|$ is the usual euclidian norm.

Note:

- *Convergence criteria for PETSC is evaluated differently from GCPC ;*
- *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 $\sqrt{resi}$;*
- *The algorithm 'GCR' be based on a prepacking on the right and thus checks the convergence criteria normalizes some nonpackaged.*

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

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