
Key word solver

1 Drank

the key word factor `solver` is common to all the commands which solve **linear systems of equations** (`STAT_NON_LINE`, `MODE_ITER_SIMULT`,...). To solve these systems of equations, particular algorithms called “linear solvers” are used. The key word `solver` makes it possible to choose the linear solver to be used among 3 categories: direct solvers, iterative solvers and the hybrid solvers.

Concerning the direct solvers, one has the classical algorithm of “Gauss” (`METHODE=' LDLT '`), of a factorization multifrontal (`"MULT_FRONT"`) and of an external resolution (`"MUMPS"`).

For the iterative solvers, it is possible to call on a conjugate gradient (`"PCG"`) or with certain algorithms of the PETSc library (`"PETSC"`).

When the first two approaches are mixed, hybrid resolution is made. It is the case of the solver multi-fields FETI (`"FETI"`).

Only `MULT_FRONT`, `MUMPS`, `PETSC` and `FETI` **are paralleled**. The first in shared memory (OpenMP), others in distributed memory (MPI). But all the solvers are compatible with a parallel processing of elementary computations and assemblies [U4.41.01].

In addition, only the three direct solvers are compatible with modal computation and the studies of buckling.

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

Contents

1 But	1
2 Syntaxe	3
3 Opérandes	6
3.1 Operand METHODE	6
3.2 Parameters common to several solveurs	7
3.3 METHODE=' MULT_FRONT'	9
3.4 METHODE=' LDLT'	10
3.5 METHODE=' MUMPS'	11
3.5.1 Parameters fonctionnels	11
3.5.2 Parameters numériques	12
3.5.3 Parameters for management mémoire	14
3.6 METHODE=' GCPC'	18
3.7 METHODE=' PETSC'	20
3.8 METHODE=' FETI'	23

2 Syntax

```

◇SOLVEUR = _F (

# Parameter common to all the solvers (direct, iterative and hybrid)
  ◇SYME=/ "NON",
[DEFAULT]
  / "OUI".

# Parameters common to the direct solvers and hybrid ("MULT_FRONT", "LDLT", "MUMPS" and
"FEI")
  ◇NPREC=/8 ,
[DEFAULT]
  /nprec . [I]
  ◇STOP_SINGULIER=/ "OUI",
[DEFAULT]
  / "NON",

direct #Solver "interns" of multifrontal type: cf §3.33.3.
  /METHODE = ' MULT_FRONT',
[DEFAULT]
  " MONGREL" ◇RENUM=/,
[DEFAULT]
  / "MD",
  / "MDA".

direct #Solver "classical" of Gauss type: cf §3.43.4.
  /METHODE = ' LDLT',
  ◇RENUM=/"RCMK",
[DEFAULT]
  / "SANS".

direct #Solver of multifrontal type based on the external product MUMPS: cf §3.53.5.
  /METHODE = ' MUMPS',
  ◇TYPE_RESOL=/ "AUTO", [DEFAULT]
  / "NONSYM",
  / "SYMGEN",
  / "SYMDEF".
  ◇PCENT_PIVOT=/20 ,
[DEFAULT]
  /pcent . [R]
  ◇ELIM_LAGR2=/ "OUI", [DEFAULT]
  / "NON".
  ◇RESI_RELA=/-1.0 , (not flax. /modal)
[DEFAULT]
  /+1.e-6 , (linear) [DEFAULT]
  /resi . [R]
  ◇FILTRAGE_MATRICE=-1.d0 ,
[DEFAULT]
  /filtma . (restricted perimeter cf
§3.53.5)
  ◇MIXER_PRECISION=/ "OUI", (restricted perimeter cf
§3.53.5)
  / "NON". [DEFAULT]
  ◇PRETRAITEMENTS=/ "SANS",
  / "AUTO". [DEFAULT]

```

Warning : The translation process used on this website is a "Machine Translation". It may be imprecise and inaccurate in whole or in part and is provided as a convenience.

```

[DEFAULT]          ◇RENUM=/                "AUTO",
                    / "AMD",
                    / "AMF",
                    / "QAMD",
                    / "PORD",
                    / "MONGREL",
                    / "SCOTCH".
                    ◇POSTTRAITEMENTS=/      "SANS",
                    / "FORCE",
                    / "AUTO".                [DEFAULT]
§3.53.5)          ◇MATR_DISTRIBUEE=/        "OUI",      (restricted perimeter of
                    / "NON".                [DEFAULT]
[DEFAULT]          ◇GESTION_MEMOIRE=/        "AUTO",
                    / "IN_CORE",
                    / "OUT_OF_CORE",
                    / "EVAL".

```

iterative Solver "interns" of preconditioned PCG type by Incomplete Cholesky $ILU(k)$ or one factorized single precision (via MUMPS). C F. § 3.6 .

```

/METHODE          = ' GCPC ',
[DEFAULT]          ◇/PRE_COND=/            "LDLT_INC",
                    ◇NIVE_REEMPLISSAGE=/0  ,
[DEFAULT]          /niv .
                    ◇RENUM=/              "SANS",
                    / "RCMK".            [DEFAULT]
                    /PRE_COND =          "LDLT_SP",
[DEFAULT]          ◇PCENT_PIVOT  =/20      ,
                    /pcent .
[DEFAULT]          ◇REAC_PRECOND=/30      ,
                    /reac .
[DEFAULT]          ◇RENUM=/              "SANS",
                    ◇NMAX_ITER=/0        ,                [DEFAULT]
                    /niter .            [I]
                    ◇RESI_RELA=/10-6     ,                [DEFAULT]
                    /resi .             [R]

```

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

```

/METHODE          = ' PETSC ',
[DEFAULT]          ◇ALGORITHME=/          "GMRES",      [DEFAULT]
                    / "CG",
                    / "CR",
                    / "GCR".
[DEFAULT]          ◇/PRE_COND=/          "LDLT_SP",
[DEFAULT]          ◇PCENT_PIVOT  =/20      ,

```

```

                                /pcent .
                                ,
[DEFAULT]      ◊REAC_PRECOND=/30
                                ,
                                /reac .
                                ,
[DEFAULT]      ◊RENUM=/
                                "SANS",
                                ,
                                /PRE_COND =
                                "LDLT_INC",
                                ,
[DEFAULT]      ◊NIVE_REPLISSAGE=/0
                                ,
                                /niv .
                                ,
[DEFAULT]      ◊REPLISSAGE=/1.0
                                ,
                                /rem .
                                ,
                                ◊RENUM=/
                                "SANS",
                                / "RCMK".      [DEFAULT]
                                ,
                                /PRE_COND =
                                "ML",
                                / "BOOMER",
                                ,
                                ◊RENUM=/
                                "SANS",
[DEFAULT]
                                ,
                                /PRE_COND =
                                "JACOBI",
                                / "SOR",
                                / "SANS".
                                ,
                                ◊RENUM=/
                                "SANS",
[DEFAULT]
                                ,
                                / "RCMK".
                                ,
                                ◊NMAX_ITER=/0
                                ,      [DEFAULT]
                                /niter .      [I]
                                ,
                                ◊RESI_RELA=/10-6
                                ,      [DEFAULT]
                                /resi .      [R]
                                ,
                                ◊MATR_DISTRIBUEE=/
                                "OUI",      (restricted perimeter cf
§3.73.7)
                                / "NON".      [DEFAULT]

```

hybrid #Solvor multidomaines of the type FETI: cf §3.83.8.

```

/METHODE = ' FETI ',
◊PARTITION=sdfeti
◊NMAX_ITER=/0
                                ,      [DEFAULT]
                                /niter .      [I]
◊REAC_RESI=/0
                                ,      [DEFAULT]
                                /nreac .      [I]
◊RESI_RELA=/10-6
                                ,      [DEFAULT]
                                /resi .      [R]
◊PRE_COND=/
                                "SANS",
                                / "LUMPE".      [DEFAULT]
◊SCALING=/
                                "SANS",
                                / "MULT".      [DEFAULT]
◊TYPE_REORTHO_DD=/
                                "SANS",
                                / "GSM",      [DEFAULT]
                                / "GS",
                                / "IGSM".

```

```

    ◇NB_REORTHO_DD=/0
[DEFAULT]
    /nb_reortho . [I]
    "
MONGREL" ◇RENUM=/,
[DEFAULT]
    / "MD",
    / "MDA".
    ◇VERIF_SDFETI=/
[DEFAULT]
    "OUI",
    / "NON".
    ◇TEST_CONTINU=/10-8
[DEFAULT]
    /test_continu . [R]
    ◇STOCKAGE_GI=/
[DEFAULT]
    "CAL",
    / "OUI",
    / "NON".
    ◇INFO_FETI=/
[DEFAULT]
    "FFFFFFFFFFFFFFFF", [DEFAULT]
    /info_feti . [K15]
    ◇NB_SD_PROC0=/0
[DEFAULT]
    /nb_sdproc0 . [I]
    ◇ACCELERATION_SM=/
[DEFAULT]
    "OUI",
    / "NON".
    ◇NB_REORTHO_INST=/0
[DEFAULT]
    /nb_reortho_inst . [I]
    ),
```

3 Operands

3.1 Operand METHODE

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

```
direct
#Solveurs
/"MULT_FRONT" [DEFAULT]
/"LDLT"
/"MUMPS"

iterative
#Solveurs
/"PCG"
/"PETSC"

hybrid
#Solvor
/"FETI"
```

direct Solver of standard multifrontal (without swivelling during factorization). This method is paralleled in shared memory (OpenMP) and can be carried out on several processors (*via* the Astk interface small Options/Options of launching/ncpus).

direct Solver with factorization of Crout per blocks (without swivelling). This solver is paginated, it can thus be carried out with little memory.

direct Solver of standard multifrontal with swivelling. This solver calls the library MUMPS developed by CERFACS/IRIT/INRIA/CNRS. Allows to treat the models leading to positive nondefinite matrixes (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* the Astk interface small Options/Options of launching/mpi_nbcpu & mpi_nbnoeud).

iterative Solver of standard conjugate gradient with preconditioning by an incomplete with *K levels* or complete factorization in single precision.

iterative Solvers resulting from the library PETSc (*Argonne National Laboratory*) with various preconditioners. This method is paralleled in distributed memory (MPI) and can be carried out on several processors (*via* the Astk interface small Options/Options of launching/mpi_nbcpu & mpi_nbnoeud).

Caution: the solvers PETSC and MUMPS being incompatible into sequential, MUMPS is generally privileged. To use PETSC, it is thus necessary often to launch a parallel version of *Code_Aster* (since it is necessary to requesting one processor).

hybrid Solver by decomposition of fields of the type FETI. Preconditioned conjugate gradient (GCPPC) project for the problem of interface and direct solver multifrontal for the local resolutions. This method is paralleled in distributed memory (MPI) and can be carried out on several processors (*via* the Astk interface small Options/Options of launching/mpi_nbcpu & mpi_nbnoeud).

Advice:

The method by default remains internal multifrontal MULT_FRONT. But for fully profiting from the savings of time and memory which parallelism gets or to solve a numerically difficult problem (X-FEM, incompressibility, THM), one recommends the use of MUMPS. To solve a problem of big size (> 106 ddls), solver PETSC with his adjustments by default can allow to pass a computation in spite of limitations memory of the machine of computation. **For more details and advice** on employment of the linear solvers one will be able to consult the specific note of use [U2.08.03].

3.2 Parameters common to several solvers

```
◇SYME=/ "OUI"  
/ "NON" [DEFAULT]
```

This parameter is common to all the linear solvers (direct, iterative and hybrid). It is applicable only for the quasi-static nonlinear operators.

If the matrix of the linear system \mathbf{K} is NON-symmetric, key word `SYME=' OUI '` makes it possible to symmetrize this matrix before the resolution of the system. The matrix is then replaced par.

$$\mathbf{K}' = \frac{1}{2}(\mathbf{K} + \mathbf{K}^T)$$

Attention:

The interest of this key word is to gain time linear systems during the resolution because the direct solvers are generally more expensive into asymmetric.

The symmetrization of the matrix \mathbf{K} however results in solving a different linear system. By doing this, it can degrade the convergence of the nonlinear algorithms.

```
◇ NPREC =/nprec  
/8 [DEFAULT]  
  
◇STOP_SINGULIER=/ "OUI" [DEFAULT]  
/ "NON"
```

These two parameters are common to all the direct linear solvers and hybrid (`LDLT`, `MULT_FRONT`, `MUMPS` and `FETI`).

They are used to control the course of numerical factorization and the quality of the solution of the linear system. The numerical factorization of a matrix can fail in two cases: problem of construction of factorized (structurally or numerically singular matrix) and numerical detection of a singularity (solution of the unstable linear system).

Key word `NPREC` and `STOP_SINGULIER` make it possible 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 $nprec < 0$, one disables the detection of singularity, if not one activates it.

In all the cases, if the value `nprec` is left to zero one 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` and `FETI` :

When at the end of factorization, one notes that term diagonal d' is become very small (compared to what it was before factorization d), it is that the matrix is (probably) almost singular. Maybe

$n = \log \left| \frac{d}{d'} \right|$, this ratio of magnitude indicates that on an equation (at least) one lost n significant figures.

If $n > nprec$, it is considered that the matrix is singular. If the user indicated:

STOP_SINGULIER=' OUI '

the code stops then in error.

STOP_SINGULIER='NON '

the execution continues after emission of an ALARME. The quality of the solution is then not guaranteed. This parameter setting is not advised. In 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 norm of line (or column) is lower than the threshold 10^{-nprec} the matrix then 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): insufficient boundary conditions of blocking of structure, redundant linear relations, 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 a negative nprec).*
- *By default, with the direct solver MUMPS, one thus have a double quality control of the solution: into linear, RESI_RELA and NPREC, in nonlinear, the criterion of Newton and NPREC. It is possible to disconnect them, but it is not advised without good reason.*

3.3 METHODE=' MULT_FRONT '

Perimeter of use:

Universal solver usable everywhere. To disadvise for the modelizations requiring of the swivelling (mixed finite elements, incompressible...).

◇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):

/"MONGREL" [DEFAULT]	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 TEMPS CPU and memory).
/"MD"	("Minimum Degree") these numbers of the nodes minimize the filling of the matrix during its factorization.
/"MDA"	("Minimum Approximate Degree") this classification is in theory less optimal than "MD" with regard to the filling but it is more economic to calculate. It is however preferable with "MD" for the large models ($\geq 50\,000$ degrees of freedom).

Note:

In the case of matrixes généralisées¹ Générées¹ comprising stresses of connection, MULT_FRONT does not apply a renumbering. This strategy is not prejudicial because these matrixes are often quasi-full and with smalls. The choice of the renumberator operated by the user is thus ignored. An informational message announces this case in the message file.

¹ 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.

3.4 METHODE=' LDLT '

Perimeter of use:

Universal solver (but very slow on large case). To disadvise for the modelizations requiring of the swivelling (mixed finite elements, incompressible...).

◇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):

/"SANS"	One keeps the initial order given in mesh file.
/"RCMK" [DEFAULT]	"Reverse Cuthill-MacKee", this algorithm of renumbering is often effective to reduce core necessary (in storage SKYLINE) of the assembled matrix and to reduce time necessary to the factorization of the matrix.

3.5 METHODE=' MUMPS '

Perimeter of use: Universal solver.

The MUMPS solver currently developed by CERFACS/CNRS/ENSEEIH-INPT/INRIA (Copyright with the §3 of [U2.08.03]) is a direct solver of multifrontal type, paralleled (in MPI) and robust, because it makes it possible to swivel the lines and columns of the matrix during numerical factorization. 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), computations of determinant necessary to operators `MODE_ITER_INV` (option "SEPARATE"/"ADJUSTS") and `INFO_MODE` (method APM) are not possible. Computation stops then in `ERREUR_F` by advising the rise of version.

3.5.1 Functional parameters

◇ `TYPE_RESOL` =

This key word makes it possible to choose the type of MUMPS resolution:

<code>/"NONSYM"</code>	Must be selected for the asymmetric matrixes.
<code>/"SYMGEN"</code>	Must be selected for the positive nondefinite symmetric matrixes. It is the most general case in <i>Code_Aster</i> because of dualisation of the boundary conditions by coefficients of Lagrange.
<code>/"SYMDEF"</code>	Can be selected for the positive definite symmetric matrixes. There is no swivelling. The algorithm is faster and less expensive in memory.
<code>/"AUTO" [DEFAULT]</code>	the code will choose "NONSYM" for the asymmetric matrixes and "SYMGEN" for the symmetric matrixes.

It is not interdict to choose "NONSYM" for a symmetric matrix. That will probably double the cost of computation but this option gives to MUMPS more algorithmic possibilities (swivelling, *scaling*...). *A contrario*, it can be interesting, in nonlinear, to symmetrize its asymmetric problem (cf parameter `SYME`). It is the same type of tricks as for the parameters of relaxation `FILTRAGE_MATRICE` and `MIXER_PRECISION`.

◇ `ELIM_LAGR2=`

This key word makes it possible to eliminate the Lagrange doubles from the taking into account of the boundary conditions:

<code>/"OUI" [DEFAULT]</code>	One does not take any more account that of a Lagrange, the other being witness
<code>/"NON"</code>	One preserves the usual dualized matrixes.

Historically, the direct linear solvers 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 computation. As 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 computation of the determinant of the matrix. This functionality is mainly necessary by operators `MODE_ITER_INV` and `INFO_MODE`. The user is then informed of this automatic change of parameter setting *via* a dedicated message (only in `INFO=2`).

◇ `RESI_RELA=/resi`
/1.d-6 [**DEFAULT**] in linear

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.

/-1.d0 [**DEFAULT**] in nonlinear and in modal computation.

This parameter is disabled by a negative value. It is appealing 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 key word (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 linear operators THER_LINEAIRE and MECA_STATIQUE. In nonlinear or in modal computation, the criterion of detection of singularity and the correction of the including algorithm (Newton or modal solver) are sufficient parapets. One can thus disconnect this process of control (it is what is made by default via value -1).

If the relative error on the solution estimated by MUMPS is higher than `resi` the code stops in ERREUR_FATALA, by specifying the nature of the problem and the accused values.

The activation of this key word initiates also a process of iterative refinement of which the purpose is to improve the solution obtained. This postprocessing profits from a particular parameter setting (key word POSTTRAITEMENTS). It is the solution resulting from this process of iterative improvement which is tested by RESI_RELA.

3.5.2 Numerical parameters

```
◇ FILTRAGE_MATRICE=/filtma  
                        /-1.d0 [ DEFAULT ]  
◇ MIXER_PRECISION=/ "OUI"  
                        / "NON" [ DEFAULT ]
```

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

These functionalities make it possible "to release" the resolutions carried out with MUMPS in order to gain in performance. The idea is simple. In nonlinear, the computation of the tangent matrix can be sullied with error. That probably will slow down the process of Newton (of nombre of iterations), but if the handling of this approximated matrix is less expensive, one can gain overall in time (less floating operations), in memory consumption (RAM even disc if the OOC is activated) and as a band-width (effect hides, volume of I/O).

Thus, the activation of functionality FILTRAGE_MATRICE, with a value of `filtma`>0, led Code_Aster has not to provide to MUMPS that the matrix terms checking

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

the filter is thus based on a threshold relative compared to the absolute values of the diagonal terms corresponding

By initializing MIXER_PRECISION with "OUI", one uses the version single precision of MUMPS by providing him a matrix Aster double precision (possibly filtered via FILTRAGE_MATRICE). From where potentially of the gains 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 relaxation of the resolutions of systems linear via MUMPS are in the line of those which already exist for the nonlinear solvers (key word NEWTON/REAC_ITER, MATRICE ...). 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 computations thereafter.*
- *But to gain core memory without being likely to lose in accuracy of computation, one can also be interested in the elements according to: parallel computation [U2.08.03], parameters GESTION_MEMOIRE, MATR_DISTRIBUEE even RENUM.*
- *Up this idea was taken with preconditioner LDLT_SP of GCPC/PETSC.*

◇PRETRAITEMENTS=

This key word 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):

/ "SANS"	No preprocessing.
/ "AUTO" [DEFAULT]	MUMPS chooses the best combination of parameters according to the situation.

◇RENUM=

This key word makes it possible to control the renumbering 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 provided with MUMPS ("AMD", "AMF", "QAMD", "PORD"), and, the "rich" libraries more and more "sophisticated" that it is necessary to install separately ("MONGREL", "SCOTCH").

/ "AMD"	"Minimum Approximate Dismantles" (Minimum Approximate Degree)
/ "AMF"	"Minimum Approximate Wire" (Approximate Minimum Filling)
/ "QAMD"	Alternative of "AMD" (automatic control of line quasi-dense)
/ "PORD"	external Tools of renumbering distributed with MUMPS.
/ "MONGREL"	external Tools of renumbering (available also with METHODE='MULT_FRONT'). It is the renumberator of reference since the end of the Nineties. He universally is recognized and used.
/ "SCOTCH"	external Tools of renumbering which tends to supplant the tools of reference in the field (MONGREL).
/ "AUTO" [DEFAULT]	MUMPS chooses the best combination of parameters according to the problem and the packages available. If the user specifies a particular renumberator and that this last is not available, the solver chooses most adequate in the list of available and an ALARME is emitted.

Note:

- *The external tools of renumbering have by way of broader features: partitioning of graphs and meshes, organization of course of tasks... For example, in Code_Aster, METIS and SCOTCH are also used in the operator of partitioning of mesh `DEFI_PART_FETI`.*
- *The choice of the renumberator has a great importance over memory consumptions and time of the linear solver. If one seeks has to optimize/regulate the numerical parameters related to the linear solver, this parameter must be one of the first to be tested.*
- *In particular, during parallel modal computations, one sometimes observed of speed-UPS disappointing because of an inappropriate choice of renumberator. In this case there one noted that the choice of a "sophisticated" renumberator was against-performing. It is to better impose on MUMPS a simple "AMF" 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 key word makes it possible to control the procedure of iterative refinement of which the purpose is to improve quality of the solution (cf key word `RESI_RELA`).

/ "SANS"	Deactivation.
----------	---------------

`/"FORCE"` MUMPS carries out at least an iteration of iterative refinement because its stopping criteria are initialized with a very low value. The nombre of iterations is limited to 10.

`/"AUTO" [DEFAULT]` MUMPS often carries out an iteration of iterative refinement. Its stopping criteria are close to the accuracy machine and the nombre of iterations 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 in Out-Of-Core even not very useful in 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 gain of at least a factor 5, the process stops. The generally noted nombre of iterations (by posing `INFO=2`) is 1 or 2.*
- *On certain badly conditioned benchmarks (for example `perf001e`), the forcing of this process made it possible to reach the desired accuracy.*

3.5.3 Parameters for management memory

to gain in RAM memory without changing linear solver (and modelization or data-processing platform), several strategies are available (and often combinable):

•With constant numerical accuracy and with savings of time of computation:

parallelism (finely `Options/mpi_**` of `Astk`) coupled, or not, with the activation of key word `MATR_DISTRIBUEE` in distributed parallelism mode (mode by default).

•With constant numerical accuracy but with potentially of the wastes of time of computation: activation clarifies Out-Of-Core faculties of MUMPS (cf key word `GESTION_MEMOIRE Ci` below) or the change of renumerator (cf key word `RENUM` considering previously). It is also necessary to take care to fund an additional space reserved for the swivelling of reasonable size: key word `PCENT_PIVOT`.

Often the values by default of these parameters (`GESTION_MEMOIRE=' AUTO'`, `RENUM=' AUTO'` and `PCENT_PIVOT=20`) get the best compromises to adapt this part of the parameter setting to your case.

•By accepting a loss of accuracy within a nonlinear process (e.g. `STAT` or `DYNA_NON_LINE`, `MODE_ITER_SIMULT...`): all parameters of relaxation related to solver (`FILTRAGE_MATRICE`, `MIXER_PRECISION`) even those related to the nonlinear process itself (elastic tangent matrix, space of projection in modal computation...).

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

```
◇ PCENT_PIVOT=/pcent  
/2 0% [ DEFAULT ]
```

This key word makes it possible to choose a percentage of memory that MUMPS will hold at the beginning of computation 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 tools require 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 Aster computations . If for example MUMPS estimate at 100 the core necessary to a factorization without swivelling, it will allocate *in fine* 120 to manage computation with swivelling. A value exceeding the 50% must remain exceptional.

Thereafter, if the memory capacity required by the swivellings proves more important, the allocated core 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 key word `GESTION_MEMOIRE`):

- If it is about a precise mode, "IN-CORE" or "OUT_OF_CORE", computation stops in `ERREUR_FATALE` and suggests various palliative solutions.
- If it is the automatic mode, "AUTO", computation 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-assignments 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 of `PCENT_PIVOT` automatically, it is this new value which is used until the end of the operator. It is supposed that the numerical difficulty will not decrease and one thus preserves this value of swivelling in order to more not waste time in fallen through attempts at factorization.
- In mode "AUTO", jointly with the doubling of the additional space of swivelling, one can also have to pass automatically in management memory Out-Of-Core MUMPS (as if one had parameterized `GESTION_MEMOIRE='OUT_OF_CORE'` explicitly). That occurs according to certain return code MUMPS or with the third (and last) attempt.

◇`GESTION_MEMOIRE=`

This key word makes it possible to choose the mode of management memory of the 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 3rd mode, the automatic mode, which will do everything (in some limiting!) so that computation does not stumble for core memory reasons. 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 key word `PCENT_PIVOT`).

<code>/"IN_CORE"</code>	One privileges to the maximum the velocity of computation. It is the option which requires the most memory, because here one allows MUMPS to preserve in RAM all the objects which it needs.
<code>/"OUT_OF_CORE"</code>	One privileges to the maximum the economies in memory consumption. It is the option which requires less memory, because here one forces MUMPS to discharge on disc his objects more encombrants ² Les ² .
<code>/"AUTO" [DEFAULT]</code>	One decides management memory automatically to impose on MUMPS (cf In-Core or preceding Out-Of-Core) according to the memory sizes available at this time precise of computation. One activates also a mechanism of pre-assignment 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 assignments to ensure the swivelling. Two mechanisms of autocorrection can be also set up so necessary (increase in the <code>PCENT_PIVOT</code> , bifurcation pre-assignments memory).
<code>/"EVAL"</code>	Helps with the calibration memory of computation. One provides a synthetic display (cf figure 3.1) of the resources memories required by computation <i>Code_Aster</i> + MUMPS ³ Les ³ according to the selected

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 MUMPS can be slightly overestimated. They recapitulate the figures necessary for a use in "stand-alone" of the 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 executable MUMPS.

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.

type of management: In-Core or Out-Of-Core. Then computation stops in `ERREUR_FATALE` in order to allow the user to start again his computation by choosing a parameter setting report leaning 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 computation, one thus needs a quantity of RAM memory at least from
      - 3500 Mo if GESTION_MEMOIRE=' IN_CORE',
      - 500 Mo if GESTION_MEMOIRE=' OUT_OF_CORE'.
In case of doubt, use GESTION_MEMOIRE='AUTO'.
*****
```

Figure 3.1. *Display in the message file in mode "AUTO".*

The activation of Out-Of-Core contributes to reduce the RAM memory required by processor, but that can slow down computation (cost of the I/O RAM/disque). This overcost can be notable when many descent-increase are carried out (nonlinear e.g. computation with much of time step or iterations of Newton, search for many eigen modes in modal computation...). 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 computation. For this reason, one privileges to the maximum the In-Core mode (especially in parallel).

In mode "EVAL", the pre-estimates of consumed the memory are much faster and less expensive in memory than complete computation. 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. A anecdotic title, this mode can be also used to coarsely test the setting in data and/or the executable one used. If all functions until this evaluating it is rather good sign for later computation!

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* provides him an estimate of RAM available. These pre-assignments often make it possible not to more have to adjust, often "with the wet finger", parameter "PCENT_PIVOT". From where an unquestionable time-saver for the clarification of the studies.

In addition, in mode "AUTO", a *Code_Aster*+MUMPS *computation* 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 do 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 evaluating of the memory disponible⁴Cela⁴.

Another mechanism also makes it possible to mitigate this kind of nuisance: if MUMPS seek 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 parameter `PCENT_PIVOT`, is activated only with mode "AUTO".

Note:

- In the standard modes ("IN_CORE" and "OUT_OF_CORE") *Code_Aster* discharges on disc the largest objects ⁵ related to the linear system. And this, in order to leave with MUMPS a maximum of core in memory RAM. So thereafter MUMPS do not have sufficient core to allocate its data, an alarm is emitted and computation continues its unfolding. According to the cases, computation

⁴ can arrive on certain platforms (e.g. `clpaster-Rocks`)

⁵ (MATR ASSE), description of unknowns (NUME_DDL)...

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.

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 "AUTO".

- In mode "AUTO", 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 déchargeables JEVEUX objects. 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` (+ advice).
- Massive unloadings of JEVEUX objects evoked previously can, in exceptional cases, to largely slow down the execution. That can happen for example in the event of clogging of the access disc in parallel mode or when one discharges much from data (fields to different time step, 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 computation in several stages.
- In mode "EVAL", the evaluating then the stop are carried out with the first matrix factorization via MUMPS. Maybe, for example, in the phase of prediction for `STAT_NON_LINE`, or in the test of Sturm type for `MODE_ITER_SIMULT`. It is often sufficient to have a good order of magnitude of the requirements in memory. For if required pushing back this evaluating, it is necessary to cut out its computation and to use another linear solver (for example "MULT_FRONT") for the operators whom one wishes to preserve.

```
◇ MATR_DISTRIBUEE=/ "OUI"  
/ "NON" [DEFAULT]
```

This parameter is for the time limited to operators `MECA_STATIQUE`, `STAT_NON_LINE` and `DYNA_NON_LINE` and it active only in parallel is not distributed (`AFFE_MODELE/PARTITION/PARALLELISME!` = 'CENTRALIZE').

By activating this key word, one causes cutting with just of the object stamps in Aster (one does not store any more useless values belonging to the other processors). That makes it possible to save memory in parallel distributed (this key word does not have any influence into sequential or parallel centralized) without overcost in time, nor loss of accuracy.

3.6 METHODE=' GCPC '

Perimeter of use:

Real symmetric problems except those requiring obligatorily a detection of singularity (modal computation). In nonlinear, if the problem is real asymmetric, one can use this solver provided that one activates, au préalable, key word SYME .

◇PRE_COND=

This key word makes it possible to choose the method of preconditioning:

/"LDLT_INC" [DEFAULT] incomplete **Decomposition** LDLT (by level) of the assembled matrix
/"LDLT_SP" Factorization single precision via the external tools MUMPS

This second approach is more expensive in CPU/RAM but more robust. Its interest lies especially in its mutualisation (cf key word REAC_PRECOND) during several resolutions if one seeks to solve problems of the multiple type second members (e.g. STAT_NON_LINE or thermomechanical sequence with MECA_STATIQUE).

◇NIVE_REPLISSAGE=/niv /0 [DEFAULT]

This parameter relates to only preconditioner LDLT_INC. The matrix of preconditioning (**P**) used to accelerate the convergence of the conjugate gradient is obtained by factorizing in a more or less complete way the initial matrix (**K**).

The larger niv is, the closer **P** the matrix is to \mathbf{K}^{-1} and thus more the conjugate gradient converges quickly (of nombre of iterations). On the other hand, more niv is the great more storage of **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 conjugate gradient to converge, one will test successively the values niv=1,2,3... .

In the same way if the nombre of iterations of the conjugate 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):

/"SANS" One keeps the initial order given in mesh file.
/"RCMK" "Transfers Cuthill-MacKee", this algorithm of renumbering is often effective to reduce core necessary (in storage SKYLINE) of the assembled matrix and to reduce time necessary to the factorization of the matrix.

The value by default of this key word depends on the preconditioner. Only LDLT_INC uses a renumbering RCMK by default.

◇REAC_PRECOND=/reac /30 [DEFAULT]

This parameter relates to only preconditioner LDLT_SP .

This preconditioner is much more expensive than the incomplete preconditioner but it is more robust because nearer to the exact solution. To make it really competitive compared to the classical direct solvers (MULT_FRONT or MUMPS double precision), it should be preserved during several successive resolutions. One thus exploits the "relative proximity" of these reiterated successive. For this making, parameter REAC_PRECOND conditions the number of times where the same preconditioner is kept whereas the matrix of the problem changed. As long as the iterative method PCG takes less `reac` iterations to converge, the unchanged preconditioner is preserved; if it exceeds this number, one reactualizes the preconditioner by remaking a factorization single precision.

```
/20  
[ DEFAULT ]  $\diamond$ PCENT_PIVOT=/pcent
```

This parameter relates to only preconditioner LDLT_SP .
It is the same key word as for the MUMPS solver , cf §3.5.33.5.3.

```
 $\diamond$  NMAX_ITER=/niter  
/0 [DEFAULT]
```

maximum Nombre of iterations of the iterative algorithm of resolution. So `niter=0` then the maximum number of iterations is calculated as follows:

$$niter = nequ/2 \text{ where } nequ \text{ is the number of equations of the system.}$$

```
 $\diamond$  RESI_RELA=/resi  
/10-6 [DEFAULT]
```

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

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

\mathbf{r}_m is the residue NON-prepacked with the iteration m
 \mathbf{f} is the second member and the norm $\|\ \|\$ is the usual euclidian norm.

3.7 METHODE=' PETSC '

Perimeter of use:

All types of real problems except those requiring obligatorily a detection of singularity (modal computation).

Caution: the solvers PETSC and MUMPS being incompatible into sequential, MUMPS is generally privileged. To use PETSC, it is thus necessary often to launch a parallel version of Code_Aster (since it is necessary to requesting one processor).

◇ALGORITHME=

Name of the iterative solvers (of Krylov type) of PETSc accessible since Code_Aster :

/`GMRES`	“Minimal Generalised RESidual”
[DEFAULT]	
/`CG`	Conjugate gradient
/`CR`	combined Residue
/`GCR`	“Generalised Conjugate Residual”

the method by default ensures the best relationship between robustness and cost of computation. The methods `CG` and `CR` are to be held for the modelizations leading to symmetric matrixes. Into asymmetric, in addition to `GMRES`, one can call on `GCR` which treats unspecified matrixes.

◇PRE_COND=

Name of the accessible preconditioners of PETSc since Code_Aster:

/`LDLT_INC`	Factorization incomplete by level
/`LDLT_SP` [DEFAULT]	Factorization single precision via the external tools MUMPS
/algebraic` Multigrid	“ml <i>multilevel</i> ” (library ml)
/algebraic` Multigrid	“BOOMER BoomerAMG” (library HYPRE)
/`JACOBI`	diagonal Pre-conditioner standard
/`SOR`	“Successive Over Relaxation”
/`SANS`	No preconditioner

In parallel mode, alone LDLT_SP, ml, BOOMER and JACOBI offer a preconditioning equivalent to the sequential mode. Two others, `LDLT_INC` and `SOR`, a little computation by means of local diagonal blocks with the processors modify. They are simpler to implement but less effective. `SANS` makes it possible not to apply a preconditioner what is of an interest only at the time of the settling of a computation.

Multigrid preconditioners algebraic ml and BOOMER have a very restricted perimeter of application (computation without Lagrange multipliers and with a constant number of degrees of freedom per node). They are nevertheless very effective in parallel.

Preconditioner `LDLT_SP` is a *priori* most robust but it is also most expensive. However, and contrary to other preconditioners, it is not rebuilt with each linear resolution, which makes it finally competitive (cf key word REAC_PRECOND).

◇NIVE_REPLISSAGE=/niv /0 [**DEFAULT**]

This parameter relates to only preconditioner LDLT_INC.

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.

Level of filling of the preconditioner of Incomplete Cholesky.

```
◇REPLISSAGE=/          α  
                /1.0  [ DEFAULT ]
```

This parameter relates to only preconditioner `LDLT_INC`.

Growth factor in the size of the preconditioner according to the level of filling (cf §3.63.6). 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 preconditioner. If this estimate is too weak, PETSc increases the objects with the flight, but this operation is more expensive.

```
◇  RENUM=/          "SANS"  
      / "RCMK"
```

Renombrateur of the type "Transfers Cuthill-Mackee" (like "LDLT") to limit the filling of factorized incomplete.

The value by default of this key word depends on the type of preconditioner used. Only pre-conditioner `LDLT_INC` carries out a renumbering by default (RCMK).

```
◇REAC_PRECOND=/reac  
                /30  [ DEFAULT ]
```

This parameter relates to only preconditioner `LDLT_SP`.

This preconditioner is much more expensive than the incomplete preconditioner but it is more robust because nearer to the exact solution. To make it really competitive compared to the classical direct solvers (`MULT_FRONT` or `MUMPS` double precision), it should be preserved during several successive resolutions.

Parameter `REAC_PRECOND` conditions the number of times where the same preconditioner is kept whereas the matrix of the problem changed. As long as the iterative solver (`ALGORITHME`) called by `PETSC` takes less `reac` iterations to converge, the unchanged preconditioner is preserved; if it exceeds this number, one reactualizes the preconditioner by remaking a factorization single precision.

```
/20  
                [ DEFAULT ] ◇PCENT_PIVOT=/pcent
```

This parameter relates to only preconditioner `LDLT_SP`.

It is the same key word as for the `MUMPS` solver , cf §3.5.33.5.3.

```
◇  MATR_DISTRIBUEE=/          "OUI"  
      / "NON" [ DEFAULT ]
```

This parameter is for the time limited to operators `MECA_STATIQUE`, `STAT_NON_LINE` and `DYNA_NON_LINE` and it active only in parallel is not distributed (`AFFE_MODELE/PARTITION/PARALLELISME! = ' CENTRALIZE'`).

By activating this key word, one causes cutting with just of the object stamps in *Aster* (one does not store any more useless values belonging to the other processors). That makes it possible to save memory in parallel distributed (this key word does not have any influence into sequential or parallel centralized) without overcost in time, nor loss of accuracy.

```
◇  NMAX_ITER=/niter  
                /1  [ DEFAULT ]
```

maximum Nombre of iterations of the iterative algorithm of resolution. So $niter \leq 0$ then it is built-in automatically by PETSc (105).

```
◇  RESI_RELA=/resi  
                /10-6  [ DEFAULT ]
```

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

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

M^{-1} is the preconditioner

\mathbf{r}_m is the residue with the iteration m

\mathbf{f} is the second member and the norm $\| \cdot \|$ is the usual euclidian norm.

Note:

- The convergence criterion for *PETSC* is differently evaluated from *PCG* ;
- When the preconditioner is of bad quality (for example because of a bad conditioning of the problem), the convergence criterion used by *PETSC* can cause bad solutions; this is why in the operators of linear computation, an additional checking on the NON-prepacked residue is carried out. The tolerance chosen for this additional criterion is \sqrt{resi} .

3.8 METHODE=' FETI '

Perimeter of use:

This method is restricted to operators MECA_STATIQUE and STAT_NON_LINE. The limitations are:

- No AFFE_CHAR_CINE,
- No macro-element,
- No Dirichlet generalized between subdomains (AFFE_CHAR_MECA/LIAISON_ ***),
- No fluid force,
- No contact-friction,
- reserves on the modelizations and the algorithms likely to impact the interfaces (except the loadings d'AFFE_CHAR_MECA for which all the cases are planned): joined elements, crack... It is to better try to fix quotas for these complicated zones inside the subdomains.
- Local Solvers with each subdomain all homogeneous, based on MULT_FRONT.

◆PARTITION=sdfeti

Name user of object SD_FETI describing partitioning in subdomains. It is generated by a call preliminary to operators DEFI_PART_FETI/OPS [U4.23.05].

◆NMAX_ITER=/niter
/0 [DEFAULT]

maximum Nombre of iterations of the GCPPC which solves the problem of interface. If niter = the 0 then number maximum of iterations is calculated as follows:

niter=max (nbi/100,10) where nbi the number of unknowns of the problem of interface.

◆REAC_RESI=/nreac
/0 [DEFAULT]

Frequency of reactualization of the computation of the residue. Advised values: 10 or 20.

◆RESI_RELA=/resi
/10-6 [DEFAULT]

Convergence criterion of the algorithm: it is a relative criterion on the residue project of the problem of interface:

$$\frac{\|\mathbf{P} \mathbf{r}_m\|}{\|\mathbf{b}\|} \leq resi$$

\mathbf{r}_m with the iteration the operator of m

\mathbf{P} projection is the residue

\mathbf{b} is the second member and $\|\ \|\$ is the usual euclidian norm.

◆ PRE_COND=

This argument makes it possible to choose the type of preconditioner for the GCPPC:

/"SANS" No preconditionning.
/"LUMPE" lumped Preconditionning.
[DEFAULT]

Normally the lumped preconditioner led to a gain in iterations and CPU, without overcost report.

◆ SCALING=

This argument makes it possible to choose the type of scaling (put on the scale) adopted for the preconditioner. It is thus taken into account only if `PRE_COND` is different from "SANS".

`/"SANS"` No phase of scaling.
`/"MULT" [DEFAULT]` Put on the scale by the multiplicity of the nodes of interface.

Normally the phase of scaling leads to a gain in iterations and CPU, without overcost report. Especially when partitioning produces many junction points (points belonging to more than two subdomains).

◇ `TYPE_REORTHO_DD=`

This argument makes it possible to choose the type of reorthogonalisation of the directions of descent (within a resolution of system linear or between various resolutions of `ACCELERATION_SM`). It is related to parameter `NB_REORTHO_DD`.

`/"SANS"` No reorthogonalisation of the methods of descent.
`/"GS"` Reorthogonalisation of Gram-Schmidt.
`/"GSM" [DEFAULT]` Reorthogonalisation of Gram-Schmidt Modified.
`/"IGSM"` Iterative Reorthogonalisation of Gram-Schmidt Modified.

This phase makes it possible to fight against the propensity of the directions of descent of the GCPPC to lose their orthogonality. In theory, "IGSM" is better than "GSM" which is him even higher than "GS". In practice, the best compromise "overcost computation/quality of orthogonality" is often carried out by "GSM".

◇ `NB_REORTHO_DD =/nb_reortho`
`/0 [DEFAULT]`

Many initial directions of descent used in the phase of reorthogonalisation. In theory, more it is large, better is convergence, but more deep east also the overcost computation and memory. It is thus necessary to find a compromise between these elements.

If `nb_reortho = 0` then this number is calculated as follows: `nb_reortho=max (niter/10,5)` where `niter` is the maximum number of iterations defined above.

◇ `RENUM` See [§3.33.3].

◇ `STOP_SINGULIER` See [§3.23.2].

◇ `NPREC` See [§3.23.2].

◇ `VERIF_SDFETI=/'OUI' [DEFAULT]`
`/'NON"`

One enters the inconsistencies in term of name of model and of names of loading, between the parameter setting of the operator calling the key word `solver` and that provides the operator of partitioning which remains stored in the `SD_FETI`. It is necessary that the names of models are identical and that the list of the loadings of the appealing operator is equal to that of `DEFI_PART_OPS`. If it is not the case and if `VERIF_SDFETI=' OUI '`, one stops in `ERREUR_FATALE`, if not one emits an `ALARME`.

◇ `TEST_CONTINU=/test_continu`
`/10-8 [DEFAULT]`

Criterion of the test of continuity to the interface: it is a criterion relative on the values (non-zero) of the unknowns to the interfaces. If one is in top of the criterion, there is emission of an `ALARME` . This criterion is for disabled time.

◇STOCKAGE_GI=

When the number of subdomains increases, an object becomes prominent, it is G_I , the matrix of the traces of the modes of rigid bodies on the interface. It is used in the phase of projection, i.e $10 + \text{nombre_itérations_FETI} \times 4$. To allow the user to adapt the compromise "cuts memory/TEMPS CPU", its storage is skeletal:

/"OUI" [DEFAULT] It is calculated and stored once for all. That requires more memory but less time computation when one makes use of it.
/"NON" It is the reverse, it is recomputed with each time that is necessary.
/"CAL" the choice "OUI" or "NON" will be calculated automatically. If the size of the matrix is lower than the average size of the local stiffness matrixes, one stores ("OUI"), if not, one recomputes ("NON").

◇INFO_FETI=/info_feti
/ "FFFFFFFFFFFFFFFF" [DEFAULT]

Key word of *monitoring* and *debugging*. Its parameter setting is detailed in documentation [U2.08.03].

◇NB_SD_PROC0=/nb_sdproc0
/0 [DEFAULT]

Parameter used in parallel mode MPI, allowing to allot an arbitrary number of subdomains to processor 0 (the "Master"). This number can thus be lower than that which would be allotted to him by the procedure of automatic distribution "subdomains/processors". That makes it possible to relieve it, in CPU and memory capacity, compared to the other processors because it must manage additional stages and potentially bulky JEVEUX objects (phase of reorthogonalisation, projections of the coarse problem...).

It is active only if it is licit: $\text{nb_sdproc0} > 0$ and $\text{nb_sdproc0} < \text{nbsd} - \text{nbproc} + 1$ (nbproc , many processors and nbsd , many subdomains).

◇ ACCELERATION_SM=

This argument makes it possible to activate the phase of acceleration of a problem with multiple second members (for example, a computation of elasticity with thermal loadings depending on time).

/"OUI" [DEFAULT] activated Acceleration if the conditions are met (see low).
/"NON" Acceleration deactivated.

When it is activated, the GCPPC of FETI solver will not start its process on the basis of zero, but on the contrary will lean on information *a priori*, that of the directions of descent stored with the nb_reortho_inst time step preceding. One will thus gain much of nombre of iterations and thus in CPU, by conceding little in memory enough (if the interface is weak in front of the size of the problem).

This argument is related to parameter NB_REORTHO_INST and is activated only if the problem is a succession of linear systems to second different members and if the reorthogonalisation of the directions of descent, within each time step, is activated (TYPE_REORTHO_DD different from "SANS").

◇ NB_REORTHO_INST =/nb_reortho_inst
/0 [DEFAULT]

With one time step given, it is the number of time step preceding which one will use the directions of descent for the procedure of acceleration. In theory, more it is large, better is convergence, but more deep east also the overcost computation and memory. It is thus necessary to find a compromise between these elements.

So $\text{nb_reortho_inst} = 0$ then this number is calculated as follows:

$$\text{nb_reortho_inst} = \max(\text{nb_pas_temps}/5, 5)$$

where `nb_pas_temps` is the number of time step problem.

And if, with time step `num_pas_temps`, `nb_reortho_inst` is higher than many time step preceding available (e.g. with 5th time step one can use only 4 the time step former ones) one dynamically fixes it at this value:

$$\text{nb_reortho_inst} = \text{num_pas_temps} - 1$$