Operator TO SOLVE

1 Goal

To solve a system of linear equations (direct or iterative method)

Methods of resolutions established in *Code_Aster* and usable by this order are:

1) method MULT_FRONT (direct method),
2) method LDLT (direct method),
3) method MUMPS (direct method),
4) method GCPC (iterative method),
5) method PETSC (iterative method).

The effective choice of the method is done through the order TO FACTORIZE [U4.55.01].

For the direct methods, the matrix must be factorized beforehand by the order TO FACTORIZE [U4.55.01]. In the case of the iterative methods with prepacking, the matrix of pre conditioning is provided it-also by the operator TO FACTORIZE [U4.55.01].

The operator allows complex resolutions for the “direct” methods (not for the iterative methods).

Product a structure of data of the type cham_no.

2 Syntax

```
U   [cham_no_*]   =   TO SOLVE
                  (   ◊   reuse  =  U,
                      ♦   MATR   =  With,
                      #   If method LDLT, MULT_FRONT, MUMPS:
                                                     /   [matr_asse_DEPL_R]
                                                     /   [matr_asse_DEPL_C]
                                                     /   [matr_asse_TEMP_R]
                                                     /   [matr_asse_TEMP_C]
                                                     /   [matr_asse_PRES_R]
                                                     /   [matr_asse_PRES_C]
                      #   If method GCPC or PETSC:
                                                     /   [matr_asse_DEPL_R]
                                                     /   [matr_asse_TEMP_R]
                                                     /   [matr_asse_PRES_R]

                      ♦   CHAM_NO    =  B,        /          [cham_no]
                      ◊   CHAM_CINE =  vcine,    /          [cham_no]

                      #   if method PETSC :
                      ◊   ALGORITHM =   /   'GMRES',      [DEFECT]
                                            /   'CG',
                                            /   'CR',
                                            /   'GCR',

                      #   if method MUMPS, GCPC, PETSC :
                      ◊   RESI_RELA =   /   1.e-6,           [DEFECT]
                                        /   eps,             [R]

                      #   if method GCPC or PETSC :
                      ◊   ♦   MATR_PREC =  precond,  /      [matr_asse_DEPL_R]
                                                     /      [matr_asse_TEMP_R]
                                                     /      [matr_asse_PRES_R]
                          ◊   NMAX_ITER =  /  niter,        [I]
                                           /  0,            [DEFECT]

                      #   if method MUMPS :
                      ◊   POSTTRAITEMENTS  =   … (see keyword SOLVEUR [U4.50.01])

                      ◊   TITLE  =  titr   ,                [l_K80]
                      ◊   INFORMATION   =  /  1   ,              [DEFECT]
                                   /  2   ,
                  )
```

If `CHAM_NO`:

[cham_no_DEPL_R]	then (*)	→	DEPL_R
[cham_no_TEMP_R]		→	TEMP_R
[cham_no_PRES_C]		→	PRES_C

3 General information

This order makes it possible to solve:

- by a direct method, the linear system $\mathbf{AX=B}$, where $\mathbf{A}$ is a matrix "factorized beforehand" by the order TO FACTORIZE [U4.51.01.],
- by an iterative method (GCPC or PETSC), the linear system $\mathbf{P^{-1}AX=P^{-1}B}$, where $\mathbf{P^{-1}}$ is a matrix of prepacking determined by the order TO FACTORIZE [U4.51.01] and $\mathbf{A}$ the initial assembled matrix.

The resolution is possible for boundary conditions of Dirichlet (boundary conditions kinematics) dualized or eliminated [U2.01.02]. In this last case, if the loading $\mathbf{X=X_0}$ on the "edge" Γ_0 is applied with a kinematic load (operator AFFE_CHAR_CINE [U4.44.03]) taken into account in the assembled matrix (operator ASSE_MATRICE [U4.61.22]), the "value" of this loading $(\mathbf{X_0})$, calculated by the operator CALC_CHAR_CINE [U4.61.03] must be provided by the keyword CHAM_CINE.

4 Operands

4.1 Operand MATR

♦ MATR = With,

Name of the matrix of the system to be solved:

- For the direct methods, one provides to MATR the concept modified by the operator TO FACTORIZE ; this matrix can be real or complex, symmetrical or not.
- For the iterative methods, one provides to MATR the initial matrix. The matrix of prepacking is to be provided with the keyword MATR_PREC.

4.2 Operand CHAM_NO

♦ CHAM_NO = B,

Name of the vector second member (in general obtained by the order ASSE_VECTEUR).

4.3 Operand CHAM_CINE

◊ CHAM_CINE = vcine,

Name of the vector representing the "value" of the boundary conditions of Dirichlet eliminated (i.e. applied with one from the orders AFFE_CHAR_CINE or AFFE_CHAR_CINE_F).

It cham_no comes from the execution of the operator CALC_CHAR_CINE on the list of char_cine (loadings kinematics) associated with the assembled matrix With [U2.01.02].

4.4 Operand ALGORITHM

```
◊   ALGORITHM =/'GMRES'    [DEFECT]
              / 'CG'
              / 'CR'
              / 'GCR'
```

This keyword is used to choose the algorithm of the iterative method PETSC. The various algorithms available are documented in the keyword SOLVEUR[U4.50.01].

4.5 Operand MATR_PREC

```
◊   MATR_PREC =   precond
```

Matrix of prepacking, obtained by the operator TO FACTORIZE [U4.55.01].
Prepacking is necessary in the iterative methods to obtain a good convergence with a minimum of iterations.
With the method GCPC, the matrix of prepacking is a matrix distinct from the matrix of the problem (keyword MATR).
On the other hand, with the method PETSC, one uses the same matrix for MATR_PREC and MATR, which wants to say that the order TO FACTORIZE must be made "places from there" (with the keyword reuse). See example below.

4.6 Operand RESI_RELA

```
◊   RESI_RELA =  /   1.e-6,            [DEFECT]
                 /   eps ,             [R]
```

This keyword is described in [U4.50.01]

For the iterative methods GCPC and PETSC, it is the convergence criteria of the algorithm.
For the method MUMPS , this keyword makes it possible to check the quality of the solution .

4.7 Operand NMAX_ITER

```
◊   NMAX_ITER =   niter
```

Maximum iteration count of the iterative algorithm.

If niter = 0 then the algorithm chooses an iteration count by default.

4.8 Operand TITLE

```
◊   TITLE  =   titr,
```

Title which one wants to give to the produced result [U4.03.01].

4.9 Operand INFORMATION

```
◊   INFORMATION   =
```

1 : no impression.
2 : impressions

5 Examples

5.1 Resolution by the direct method MULT_FRONT

- Constitution of the assembled matrix and the second member:

  The elementary terms before were calculated KEL, FEL.

```
NAKED  =NUMÉRIQUE_DDL (MATR_RIGI=KEL)
K   =ASSE_MATRICE (MATR_ELEM=KEL, NUME_DDL=NU,)
F   =ASSE_VECTEUR (MATR_ELEM=FEL, NUME_DDL=NU,)
```

- Factorization:

```
K   =FACTORISER (reuse=K,  MATR_ASSE=K, METHODE=' MULT_FRONT',)
```

- Resolution:

```
U   =RESOUDRE (MATR=K,  CHAM_NO=F,)
```

- for the use of the loads kinematics (with elimination of the imposed degrees of freedom), to see the example set in the order AFFE_CHAR_CINE [U4.44.03].

5.2 Resolution by method MUMPS

```
NAKED     =  NUME_DDL (MATR_RIGI= KEL)

K      =  ASSE_MATRICE (  MATR_ELEM= KEL, NAKED NUME_DDL=)
F      =  ASSE_VECTEUR (  VECT_ELEM= FEL, NAKED NUME_DDL=)
K      =  TO FACTORIZE (  reuse= K, MATR_ASSE= K, METHODE= ‘MUMPS' )
EPD.      =  TO SOLVE     (  CHAM_NO  = F ,  MATR= K )
```

5.3 Resolution by the method of the conditioned pre combined gradient

```
NAKED     =  NUME_DDL (MATR_RIGI= KEL)

K      =  ASSE_MATRICE (  MATR_ELEM= KEL, NAKED NUME_DDL=)
F      =  ASSE_VECTEUR (  VECT_ELEM= FEL, NAKED NUME_DDL=)
KPREC  =  TO FACTORIZE (  MATR_ASSE= K, METHODE= ‘GCPC’,
                          PRE_COND=' LDLT_INC')
EPD.      =  TO SOLVE     (  CHAM_NO  = F ,  MATR= K, MATR_PREC= KPREC,
                          NMAX_ITER= 1000  ,  RESI_RELA= 1e-07
                        )
```

5.4 Resolution by method PETSC

```
NAKED     =  NUME_DDL (MATR_RIGI= KEL)

K      =  ASSE_MATRICE (  MATR_ELEM= KEL, NAKED NUME_DDL=)
F      =  ASSE_VECTEUR (  VECT_ELEM= FEL, NAKED NUME_DDL=)
K      =  TO FACTORIZE (  reuse=K, MATR_ASSE= K, METHODE= ‘PETSC’)
EPD.      =  TO SOLVE     (  CHAM_NO  = F ,  MATR= K, MATR_PREC= K,
                          ALGORITHME=' GMRES',
                          NMAX_ITER= 1000  ,  RESI_RELA= 1e-07 )
```