Structure de données sd_solveur

Résumé :

Ce document décrit la structure de données SOLVEUR. Celle-ci définit la méthode de résolution des systèmes linéaires ainsi que les paramètres attenants au solveur choisi.

Table des Matières

1 Généralités

Cet objet de type SOLVEUR a pour fonction de stocker et de véhiculer entre les différentes routines du code (en interne d'une commande ou entre commandes), les informations liées aux paramétrages des solveurs linéaires. En particulier, il définit la méthode de résolution des systèmes linéaires mono-domaine (LDLT, MULT_FRONT, PETSC, MUMPS ou GCPC) ou multi-domaines (FETI). Cet objet est créé sur la base volatile (le cas le plus fréquent) ou sur la base globale (commandes éclatées).

Le plus souvent, il est créé et remplit *via* CRESOL/CRSV** (deux premières lettres du solveur : MU pour MUMPS solveur direct et CRSMSP pour MUMPS pré-conditionneur, FE pour FETI, PE pour PETSC, LD pour LDLT, GC pour GCPC et MF pour MULT_FRONT).
Dans le cas particulier de FETI, compte-tenu de la boucle sur les sous-domaines (qui conceptuellement pourrait accepter une SD_SOLVEUR distinctes par sous-domaine), on finit le remplissage avec CRESO1 (appelée donc par CRSVFE).
Pour répondre à quelques cas particuliers, d'autres routines dédiées créent et remplissent une SD_SOLVEUR : CRSOLV [1], OP0014 [2] et CRSINT [3]. On essaie d'en limiter le nombre et de privilégier l'usage de la routine principale : CRESOL.
Quelle que soit la routine qui crée les objets de la structure de données SOLVEUR , le dimensionnement se fait exclusivement dans une routine chapeau : SDSOLV .

Lors d'une modification de cette structure de données il faut donc veiller à :
- *mettre en cohérence, si nécessaire, les sources mentionnées ci-dessus,*
- *mettre à jour, si nécessaire, le catalogue sd_solveur.py ,*
- *mettre à jour les documentations (cette doc. D et si nécessaire la doc. U4.50.01),*
- *enrichir ou modifier, si nécessaire, quelques cas-tests.*

2 Arborescence

```
SOLVEUR (K19)   ::=record
  ♦     '.SLVK'   :   OJB   S V K24  long=12  (initialisé à 'XXXX')
  ♦     '.SLVR'   :   OJB   S V R    long=4   (initialisé à 0.d0)
  ♦     '.SLVI'   :   OJB   S V I    long=8   (initialisé à -9999)

# si solveur FETI (SLVK(1)='FETI') :
  ◊     '.FETS'   :   OJB   S V K24 dim=nbsd (nombre de sous-domaines)
```

3 Contenu des objets de base

3.1 Cas FETI

On parle ici du contenu du SOLVEUR « global » ou « maître ».
SLVK :
- SLVK(1) : méthode de résolution ('FETI') ('MULT_FRONT' pour les SD),
- SLVK(2) : préconditionnement de la matrice de travail (PRE_COND='LUMPE'/'SANS'),
- SLVK(3) : valeur de VERIF_SDFETI ('OUI'/'NON'),
- SLVK(4) : valeur de RENUM ('MD'/'MDA'/'METIS'),
- SLVK(5) : valeur de SYME ('OUI'/'NON'),
- SLVK(6) : nom de la structure de données de type SD_FETI,
- SLVK(7) : valeur de TYPE_REORTHO_DD ('GS'/'GSM'/'IGSM'/'SANS'),

1 Pour CALC_CORR_SSD, MODE_STATIQUE, NUME_DDL_GENE, NUME_DDL, MACR_ELEM_STAT.
2 Pour FACTORISER.
3 Pour un appel en sous-main de MUMPS dans MODE_STATIQUE et CALC_CORR_SSD.

SLVK(8) : valeur de SCALING ('MULT'/'SANS'),
SLVK(9) : valeur de STOCKAGE_GI ('CAL'/'OUI'/'NON'),
SLVK(10) : inutilisé,
SLVK(11) : valeur de ACCELERATION_SM ('OUI'/'NON'),
SLVK(12) : inutilisé.

SLVR :
SLVR(1) : inutilisé,
SLVR(2) : valeur de RESI_RELA,
SLVR(3) : inutilisé,
SLVR(4) : valeur de TEST_CONTINU.

SLVI :
SLVI(1): valeur de NPREC,
SLVI(2): valeur de NMAX_ITER,
SLVI(3): istop
test de singularité lors de la factorisation
(prend la valeur 1 si STOP_SINGULIER = 'NON', 0 sinon),
SLVI(4): inutilisé,
SLVI(5): valeur de NB_REORTHO_DD,
SLVI(6): valeur de NB_REORTHO_INST,
SLVI(7): valeur de REAC_RESI,
SLVI(8): inutilisé.

FETS : S V K24 dim=nbsd (nombre de sous-domaines)
FETS(i)= nom de la SD SOLVEUR du i^{ème} sous-domaine.

Ensuite pour chacune des SD_SOLVEUR (pointée par .FETS) et associée à un sous-domaine particulier, on a

SLVK :
SLVK(1) : méthode de résolution ('MULT_FRONT'),
SLVK(2) : inutilisé,
SLVK(3) : inutilisé,
SLVK(4) : valeur de RENUM ('MD'/'MDA'/'METIS'),
SLVK(5) : valeur de SYME ('OUI'/'NON'),
SLVK(6) : nom de la structure de données de type SD_FETI,
SLVK(7) : inutilisé,
SLVK(8) : inutilisé,
SLVK(9) : valeur de STOCKAGE_GI ('CAL'/'OUI'/'NON'),
SLVK(10) à SLVK(12) : inutilisés.

SLVR :
SLVR(1) à SLVR(4) : inutilisés.

SLVI :
SLVI(1): valeur de NPREC,
SLVI(2): inutilisé,
SLVI(3): istop
test de singularité lors de la factorisation
(prend la valeur 1 si STOP_SINGULIER = 'NON', 0 sinon).
SLVI(4) à SLVI(8) : inutilisés,

3.2 Cas LDLT ou MULT_FRONT

SLVK :
SLVK(1) : méthode de résolution ('LDLT' ou 'MULT_FRONT'),
SLVK(2) : inutilisé,
SLVK(3) : inutilisé,
SLVK(4) : méthode de renumérotation (mot-clé RENUM). Les valeurs possibles sont :
'RCMK' ou 'SANS' (si LDLT)
'MD' ou 'MDA' ou 'METIS' (si MULT_FRONT),
SLVK(5) : valeur de SYME ('OUI'/'NON'),
SLVK(6) à SLVK(12) : inutilisés.

SLVR :
SLVR(1) : inutilisé,
SLVR(2) : inutilisé,
SLVR(3) : taille bloc (si LDLT)
c'est la taille des blocs de l'objet .UALF d'un stockage en ligne de ciel,
SLVR(4) : inutilisé.

SLVI :
SLVI(1) : valeur de NPREC,
SLVI(2) : inutilisé,
SLVI(3) : istop
Comportement voulu en cas de singularité lors de la factorisation
0 : erreur <F> en cas de singularité ou de quasi-singularité
1 : erreur <F> en cas de singularité
alarme <A> en cas de quasi-singularité
2 : Aucun message en cas de singularité ou de quasi-singularité
SLVI(4) à SLVI(8) : inutilisés.

3.3 Cas GCPC

SLVK :
SLVK(1) : méthode de résolution 'GCPC',
SLVK(2) : préconditionnement de la matrice de travail
(PRECOND='LDLT_INC', 'LDLT_SP' ou 'SANS'),
SLVK(3) : nom de la SD solveur Mumps dans le cas PRECOND='LDLT_SP',
SLVK(4) : valeur de RENUM ('RCMK' ou 'SANS'),
SLVK(5) : valeur de SYME ('OUI'/'NON'),
SLVK(6) à SLVK(12) : inutilisés.

SLVR :
SLVR(1) : valeur de RESI_RELA (copie pour NEWTON_KRYLOV),
SLVR(2) : valeur de RESI_RELA,
SLVR(3) à SLVR(4) : inutilisés.

SLVI :
SLVI(1) : inutilisé,
SLVI(2) : valeur de NMAX_ITER,
SLVI(3) : inutilisé,
SLVI(4) : valeur de NIVE_REMPLISSAGE,
SLVI(5) : nombre d'itérations pour atteindre la convergence dans le cas PRECOND='LDLT_SP',
SLVI(6) : valeur de REAC_PRECOND dans le cas PRECOND='LDLT_SP',
SLVI(7) : valeur de PCENT_PIVOT dans le cas PRECOND='LDLT_SP',
SLVI(8) : istop
Comportement souhaité en cas d'erreur lors de la résolution linéaire itérative
0 : erreur <F> en cas d'échec
2 : aucun message en cas d'échec, code retour non nul

3.4 Cas PETSc

SLVK :
SLVK(1) : méthode de résolution 'PETSC',
SLVK(2) : préconditionnement de la matrice de travail
('LDLT_INC', 'LDLT_SP', 'JACOBI' ou 'SOR'),
SLVK(3) : nom de la SD solveur Mumps dans le cas PRECOND='LDLT_SP',
SLVK(4) : valeur de RENUM ('RCMK' ou 'SANS'),
SLVK(5) : valeur de SYME ('OUI' ou 'NON'),
SLVK(6) : nom de la méthode itérative utilisée
('CG', 'BCGS', 'BICG', 'CR', 'GMRES', TFQMR'),
SLVK(6) à SLVK(12) : inutilisés.

SLVR :
SLVR(1) : valeur de RESI_RELA (copie pour NEWTON_KRYLOV),
SLVR(2) : valeur de RESI_RELA,
SLVR(3) : valeur de REMPLISSAGE,
SLVR(4) : valeur de RESI_RELA_PC (mot-clé caché).

SLVI :
SLVI(1) : inutilisé,
SLVI(2) : valeur de NMAX_ITER,
SLVI(3) : inutilisé,
SLVI(4) : valeur de NIVE_REMPLISSAGE,
SLVI(5) : nombre d'itérations pour atteindre la convergence dans le cas PRECOND='LDLT_SP',
SLVI(6) : valeur de REAC_PRECOND dans le cas PRECOND='LDLT_SP',
SLVI(7) : valeur de PCENT_PIVOT dans le cas PRECOND='LDLT_SP',
SLVI(8) : istop
Comportement souhaité en cas d'erreur lors de la résolution linéaire itérative
0 : erreur <F> en cas d'échec
2 : aucun message en cas d'échec, code retour non nul

3.5 Cas MUMPS

SLVK :
SLVK(1) : méthode de résolution ('MUMPS'),
SLVK(2) : prétraitements (PRETRAITEMENTS='AUTO' ou 'SANS'),
SLVK(3) : algorithme de résolution souhaité (TYPE_RESOL=)
'NONSYM' : matrice non-symétrique (factorisation LU)
'SYMGEN' : matrice symétrique "générale"
'SYMDEF' : matrice symétrique "définie positive"
'AUTO' : choix automatique fait au vu des caractéristiques de la matrice
SLVK(4) : renuméroteur souhaité (RENUM='AUTO','AMD','AMF','QAMD','PORD'
et 'METIS'),
SLVK(5) : symétrisation forcée (SYME='OUI'/'NON'),
SLVK(6) : élimination «virtuelle» de la 2ieme famille de Lagranges lorsqu'on transmet la matrice Aster à MUMPS (ELIM_LAGR2='OUI'/'NON'),
SLVK(7) : précision mixte (MIXER_PRECISION='OUI'/'NON'),
SLVK(8) : utilisation en préconditionneur simple précision pour le GCPC ('OUI'/'NON'),
SLVK(9) : gestion de la mémoire allouée par MUMPS
(GESTION_MEMOIRE='IN_CORE'/'OUT_OF_CORE'/'AUTO'/'EVAL'),
SLVK(10) : en parallèle distribué, retailler au plus juste les morceaux de matrices Aster conformément au périmètre de mailles dont chaque processeur a la responsabilité (MATR_DISTRIBUEE='OUI'/'NON'),
SVLK(11) : gestion des post-traitements (POSTTRAITEMENTS='SANS','AUTO','FORCE'),

SVLK(12) : numéro de version de MUMPS (par exemple: '4.9.2' ou '4.10.0'). Il s'agit d'un numéro version licite, c'est-à-dire testé et approuvé par la version de *Code_Aster* considérée. Dans la cas contraire on s'arrête en ERREUR_F avant le remplissage de ce champ. Valeur remplie juste après l'initialisation de l'occurence MUMPS uniquement *via* les routines amump*/mumpu.F.

SLVR :
- SLVR(1) : valeur de FILTRAGE_MATRICE,
- SLVR(2) : valeur de RESI_RELA (calcul et contrôle de la qualité de la solution, déclenchement des post-traitements suivant la valeur de POSTTRAITEMENTS),
- SLVR(3) à SLVR(4) : inutilisés.

SLVI :
- SLVI(1) : valeur de NPREC (comme LDLT et MULT_FRONT),
- SLVI(2) : pourcentage de mémoire supplémentaire nécessaire aux pivotages tardifs (valeur de PCENT_PIVOT)
- SLVI(3) : valeur de ISTOP (comme LDLT et MULT_FRONT),
- SLVI(4) : indicateur pour dire à MUMPS de ne pas stocker les termes de sa factorisée (intéressant si par exemple, on a juste besoin d'un résultat global type calcul de déterminant ou critère de Sturm). Si il vaut 1, on ne stocke pas cette factorisée (gros gain mémoire), sinon on la garde suivant le mode standard.
  Cette fonctionnalité n'est activée qu'à partir de MUMPS 4.10.0 (pour les versions en deçà on ne fait rien et on émet un message UTMESS_I (si INFO=2). Valeur temporaire (on remet, si nécessaire, après usage sa valeur initiale) remplie dans les routines vpstur.f et apchar.f .
- SLVI(5) : indicateur pour dire à MUMPS de calculer en plus le déterminant de la matrice. Si il vaut 1, on le calcule et on le stocke dans l'objet temporaire ' &&AMUMP.DETERMINANT ' (cf. amumpu.F), sinon on ne le calcule pas.
  Cette fonctionnalité n'est activée qu'à partir de MUMPS 4.10.0 (pour les versions en deçà on s'arrête en ERREUR_F . Valeur temporaire (on remet, si nécessaire, après usage sa valeur initiale) remplie dans les routines vpstur.f et apchar.f .
- SLVI(6) à SLVI(8) : inutilisés.

4 Compléments dans le cas du solveur FETI

4.1 Structure de données SOLVEUR récursive

Dans le cas de la méthode FETI, la structure de données SOLVEUR est récursive à deux niveaux. Une SD SOLVEUR " maître ", concernant le domaine global (SLVK(1)='FETI'), regroupe les objets JEVEUX habituels avec en plus un objet .FETS. Ce dernier est un pointeur désignant les SD SOLVEUR " esclaves " associées à chaque sous-domaines. Ces SD SOLVEUR locales sont constituées des mêmes objets JEVEUX qu'un solveur linéaire mono-domaine usuel (tel MULT_FRONT ou LDLT) et sont concernées par le même paramétrage. Au niveau d'un sous-domaine, on n'a à se préoccuper que d'inversions de matrices locales et non, par exemple, du nom de la SD SD_FETI ou du type de réorthogonalisation mise en place par le solveur d'interface global (dont le paramétrage est contenu dans la SD SOLVEUR " maître "). Pour l'instant, l'implémentation de FETI dans Code_Aster présuppose que ces sous-domaines utilisent tous le même solveur linéaire mono-domaine (SLVK(1) = 'MULT_FRONT' imposé par défaut) et avec le même paramétrage (RENUM et STOP_SINGULIER/NPREC). Cette homogénéité facilite les manipulations des matrices et des seconds membres locaux.

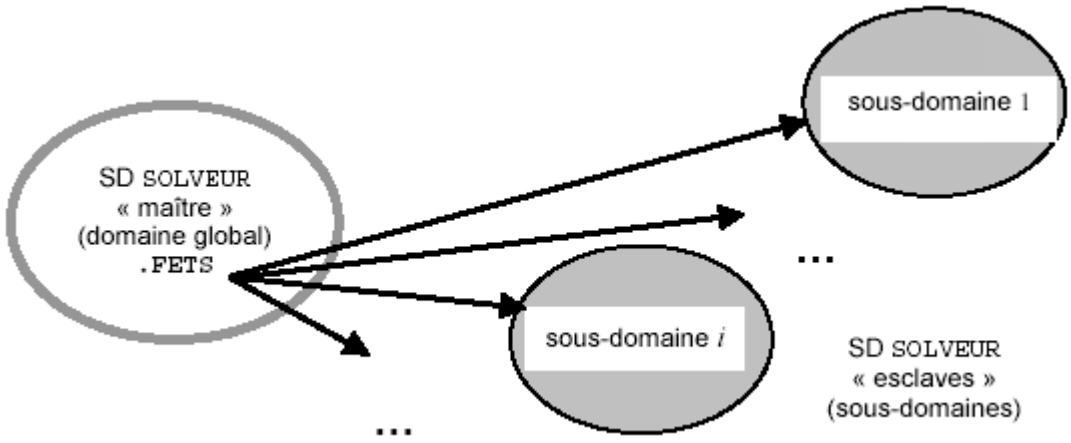

Figure 4.1-a: Structure de données SOLVEUR récursive si solveur FETI

4.2 Règle de nommage

Dans le cas d'un solveur FETI , on a choisi la règle de nommage suivante pour les SD SOLVEUR esclaves liées chaque à un sous-domaine :

nom_de_la_SD_SOLVEUR_maître(1:11) 'F' chaîne_de_caractères_libre(2:8)

La chaîne de caractères libre est engendrée par un appel à la routine GCNCON .

4.3 Cas particulier du parallélisme MPI

Lors d'une exécution en mode parallèle MPI, un processeur se voit attribuer un certain nombre de sous-domaines (cf. objets annexes ' &FETI.LISTE... ' de la structure de données SD_FETI [D4.06.21]).

La SD SOLVEUR "maître" est toujours construite, mais son pointeur . FETS ne va désigner que les sous-domaines concernés par le processeur courant : .FETS(j $_k$) sera un K24 valide que si le sous-domaine j $_k$ est dans le périmètre du processeur j .

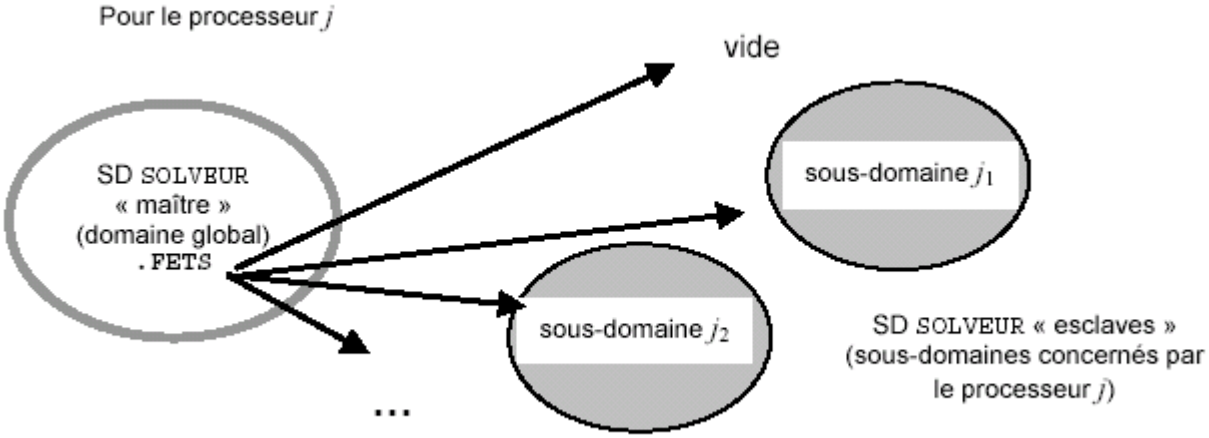

Figure 4.3-a: Structure de données SOLVEUR récursive si solveur FETI et parallélisme MPI

4.4 Exemple

Dans le cas test FETI002A, le partitionnement en quatre sous-domaines conduit aux SD SOLVEUR suivantes :

Construit une SD SOLVEUR maître '&&OP0046.SOLVEUR'

```
====> IMPR_CO DE LA STRUCTURE DE DONNEE : &&OP0046.SOLVEUR   ?????
 ATTRIBUT :  F  CONTENU :  T  BASE : >V<
 NOMBRE D'OBJETS (OU COLLECTIONS) TROUVES :                    4
================================================================================
  IMPRESSION DU CONTENU DES OBJETS TROUVES :
--------------------------------------------------------------------------------
IMPRESSION SEGMENT DE VALEURS >&&OP0046.SOLVEUR   .FETS
      1 - >&&OP0046.S.SOF0000000   <>&&OP0046.SOF0000001    <
      3 - >&&OP0046.S.SOF0000002   <>&&OP0046.SOF0000003     <
--------------------------------------------------------------------------------
IMPRESSION SEGMENT DE VALEURS >&&OP0046.SOLVEUR   .SLVI        <
      1 -            8          100            0            0          100
--------------------------------------------------------------------------------
IMPRESSION SEGMENT DE VALEURS >&&OP0046.SOLVEUR   .SLVK        <
      1 - >FETI                   <>LUMPE                  <
      3 - >OUI                    <>METIS                  <
      5 - >NON                    <>SDFETI                 <
      7 - >GSM                    <>MULT                   <
      9 - >CAL
--------------------------------------------------------------------------------
IMPRESSION SEGMENT DE VALEURS >&&OP0046.SOLVEUR   .SLVR        <
      1 -  0.00000D+00  1.00000D-08  8.00000D+02  1.00000D-08

et des SD SOLVEUR esclaves '&&OP0046.SOF0000000'

====> IMPR_CO DE LA STRUCTURE DE DONNEE : &&OP0046.SOF0000000 ?????
 ATTRIBUT :  F  CONTENU :  T  BASE : >V<
 NOMBRE D'OBJETS (OU COLLECTIONS) TROUVES :                     3
================================================================================
  IMPRESSION DU CONTENU DES OBJETS TROUVES :
--------------------------------------------------------------------------------
IMPRESSION SEGMENT DE VALEURS >&&OP0046.SOF0000000.SLVI
      1 -            8          100            0            0          100
--------------------------------------------------------------------------------
IMPRESSION SEGMENT DE VALEURS >&&OP0046.SOF0000000.SLVK        <
      1 - >MULT_FRO               <>LUMPE                  <
      3 - >OUI                    <>METIS                  <
      5 - >NON                    <>XXXX                   <
      7 - >GSM                    <>MULT                   <
      9 - >CAL
--------------------------------------------------------------------------------
IMPRESSION SEGMENT DE VALEURS >&&OP0046.SOF0000000.SLVR        <
      1 -  0.00000D+00  1.00000D-08  8.00000D+02  1.00000D-08
 ====> FIN IMPR_CO DE DE STRUCTURE DE DONNEE : &&OP0046.SOF0000000?????
...
```