Data format Summarized

sd_solvor:

This document describes the data structure solver. This one defines the method of resolution of the linear systems as well as the parameters contiguous to the selected solver.

Contents

1 General information

This object of the type solver have as a function to store and convey between the various routines of the code (in-house of a command or between commands), information related to the parameter settings of the linear solvers. In particular, it defines the method of resolution of the linear systems mono-field (LDLT, MULT_FRONT, PETSC, MUMPS or PCG) or multi-fields (FETI). This object is created on the volatile basis (the most frequent case) or on the global database (splitted commands).

Generally, it is created and fills out *via* CRESOL/CRSV ** (the first two letters of the solver: MU for MUMPS direct solver and CRSMSP for MUMPS pre-conditioner, FE for FETI, EP for PETSC, LD for LDLT, GC for PCG and MF for MULT_FRONT).
In the cas particulier of FETI, taking into account the loop on the subdomains (which conceptually could accept a distinct SD_SOLVEUR by subdomain), one finishes the filling with CRESO1 (thus called by CRSVFE).
To answer some typical cases, other dedicated routines create and fill a SD_SOLVEUR : CRSOLV [1], OP0014 [2] and CRSINT [3]. One tries to limit the number and to privilege the use of the principal routine of it: CRESOL.
Whatever the routine which creates the objects of the data structure solver , the design is done exclusively in a routine hat: SDSOLV .

During a modification of this data structure it is thus necessary to take care of:
- *to put in coherence, if necessary, sources mentioned above,*
- *to update, if necessary, the catalog sd_solveur.py, to update*
- *documentations (this Doc. D and if necessary Doc. U4.50.01), to enrich*
- *or modify, if necessary, some benchmarks. Tree structure*

2 solver

(K19):: =record ♦
". SLVK": OJB S V K24 long =12 (initialized with " ") ♦
". SLVR": OJB S V R long =4 (initialized with 0. d0) ♦
". SLVI": OJB S V I long =8 (initialized to -9999) #

so FETI solver (SLVK (1) = ' FETI'): ◊
". FETS": OJB S V K24 dim=nbsd (many subdomains) Contents

3 of the basic objects Case

3.1 FETI One

speaks here about the contents of the solver " total" or "main". SLVK
: SLVK
(1): method of resolution (" FETI") ("MULT_FRONT" for the SD), SLVK
(2): preconditionning of the matrix of work (PRE _COND=' LUMPE'/"SANS"), SLVK
(3): value of VERIF_SDFETI ("OUI'/"NON"), SLVK
(4): value of RENUM (" MD"/"MONGREL" "MDA"/), SLVK
(5): value of SYME ("OUI'/"NON"), SLVK
(6): name of data structure of SD_FETI type , SLVK
(7): value of TYPE_REORTHO_DD ("GS"/"GSM"/"IGSM"/"SANS"), SLVK

1 CALC_CORR_SSD, MODE_STATIQUE, NUME_DDL_GENE, NUME_DDL, MACR_ELEM_STAT.
2 TO FACTORIZE.
3 a call in writing pad of MUMPS in MODE_STATIQUE and CALC_CORR_SSD

(8): value of SCALING ("MULT"/"SANS"), SLVK
(9): value of STOCKAGE_GI ("CAL"/"OUI'/"NON"), SLVK
(10): unutilised, SLVK
(11): value of ACCELERATION_SM (" OUI'/"NON"), SLVK
(12): unutilised. SLVR

: SLVR
(1): unutilised, SLVR
(2): value of RESI_RELA , SLVR
(3): unutilised, SLVR
(4): value of TEST_CONTINU . SLVI

: SLVI
(1): value of NPREC , SLVI
(2): value of NMAX_ITER , SLVI
(3): istop test
of singularity during factorization (takes
value 1 if STOP_SINGULIER = "NON", 0 if not), SLVI
(4): unutilised, SLVI
(5): value of NB_REORTHO_DD , SLVI
(6): value of NB_REORTHO_INST , SLVI
(7): value of REAC_RESI , SLVI
(8): unutilised. FETS

: S V K24 dim=nbsd (many subdomains) FETS
(I) = name of the SD solver of the ième subdomain . Then

for each one of SD _SOLVEUR (pointed by .FETS) and associated with a particular subdomain, there are SLVK

: SLVK
(1): method of resolution (" MULT_FRONT"), SLVK
(2): unutilised, SLVK
(3): unutilised, SLVK
(4): value of RENUM (" MD"/"MONGREL" "MDA"/), SLVK
(5): value of SYME ("OUI'/"NON"), SLVK
(6): name of data structure of SD_FETI type , SLVK
(7): unutilised, SLVK
(8): unutilised, SLVK
(9): value of STOCKAGE_GI ("CAL"/"OUI'/"NON"), SLVK
(10) with SLVK (12): unutilised. SLVR

: SLVR
(1) with SLVR (4): unutilised. SLVI

: SLVI
(1): value of NPREC , SLVI
(2): unutilised, SLVI
(3): istop test
of singularity during factorization (takes
value 1 if STOP_SINGULIER = "NON", 0 if not). SLVI
(4) with SLVI (8): unutilised, Case

3.2 LDLT or MULT_FRONT SLVK

: SLVK
(1): method of resolution (" LDLT" or "MULT_FRONT"), SLVK
(2): unutilised, SLVK
(3): unutilised, SLVK
(4): method of renumbering (key word RENUM). The possible values are: "RCMK
" or "SANS " (if LDLT) "MD
" or "MDA " or "METIS " (if MULT_FRONT), SLVK
(5): value of SYME ("OUI'/"NON"), SLVK
(6) with SLVK (12): unutilised. SLVR

: SLVR
(1): unutilised, SLVR
(2): unutilised, SLVR
(3): cut block (if LDLT) it
is the size of the blocks of object .UALF of a storage on sky line, SLVR
(4): unutilised. SLVI

: SLVI
(1): value of NPREC , SLVI
(2): unutilised, SLVI
(3): istop Behavior
desired in the event of singularity during factorization 0
: <F error > in the event of singularity or of quasi-singularity 1
: <F error > in the event of singularity alarms
<A > in the event of quasi-singularity 2
: No message in the event of singularity or of quasi-singularity SLVI
(4) with SLVI (8): unutilised. Case

3.3 PCG SLVK

: SLVK
(1): method of resolution "PCG ", SLVK
(2): preconditionning of the matrix of work (PRECOND
= ' LDLT_INC', "LDLT_SP" or "SANS "), SLVK
(3): name of the SD Mumps solver in the case PRECOND = ' LDLT_SP', SLVK
(4): value of RENUM ("RCMK" or "SANS "), SLVK
(5): value of SYME ("OUI'/"NON"), SLVK
(6) with SLVK (12): unutilised. SLVR

: SLVR
(1): value of RESI_RELA (copy for NEWTON_KRYLOV), SLVR
(2): value of RESI_RELA , SLVR
(3) with SLVR (4): unutilised. SLVI

: SLVI
(1): unutilised, SLVI
(2): value of NMAX_ITER , SLVI
(3): unutilised, SLVI
(4): value of NIVE_REMPLISSAGE , SLVI
(5): nombre of iterations to reach convergence in the case PRECOND = ' LDLT_SP', SLVI
(6): value of REAC_PRECOND in the case PRECOND = ' LDLT_SP', SLVI
(7): value of PCENT_PIVOT in the case PRECOND = ' LDLT_SP', SLVI
(8): istop Behavior
wished in the event of iterative error during the resolution linear 0
: <F error > in the event of failure 2

: no message in the event of failure, non-zero return code Case

3.4 PETSc SLVK

: SLVK
(1): method of resolution "PETSC ", SLVK
(2): preconditionning of the matrix of work (PRECOND
= ' LDLT_INC', "LDLT_SP", "JACOBI" or "SOR "), SLVK
(3): name of the SD Mumps solver in the case PRECOND = ' LDLT_SP', SLVK
(4): value of RENUM ("RCMK" or "SANS"), SLVK
(5): value of SYME ("OUI' or "NON"), SLVK
(6): name of the iterative method used ("
CG", "BCGS", "BICG", "CR", "GMRES", TFQMR'), SLVK
(6) with SLVK (12): unutilised. SLVR

: SLVR
(1): value of RESI_RELA (copy for NEWTON_KRYLOV), SLVR
(2): value of RESI_RELA , SLVR
(3): value of REMPLISSAGE , SLVR
(4): value of RESI _RELA_PC (hidden key word). SLVI

: SLVI
(1): unutilised, SLVI
(2): value of NMAX_ITER , SLVI
(3): unutilised, SLVI
(4): value of NIVE_REMPLISSAGE , SLVI
(5): nombre of iterations to reach convergence in the case PRECOND = ' LDLT_SP', SLVI
(6): value of REAC_PRECOND in the case PRECOND = ' LDLT_SP', SLVI
(7): value of PCENT_PIVOT in the case PRECOND = ' LDLT_SP', SLVI
(8): istop Behavior
wished in the event of iterative error during the resolution linear 0
: <F error > in the event of failure 2
: no message in the event of failure, non-zero return code Case

3.5 MUMPS SLVK

: SLVK
(1): method of resolution (" MUMPS"), SLVK
(2): pretreatments (PRETRAITEMENTS = ' AUTO' or "SANS "), SLVK
(3): algorithm of resolution wished (TYPE _RESOL=) "NONSYM
": nonsymmetrical matrix (factorization READ) "SYMGEN
": symmetric matrix "general" "SYMDEF
": symmetric matrix "definite positive" "AUTO
": automatic choice made within sight of the characteristics of matrix SLVK
(4): desired renumerotor (RENUM = ' AUTO', "AMD", "AMF", "QAMD", "PORD" and
"METIS "), SLVK
(5): forced symmetrization (SYME = ' OUI'/"NON"), SLVK
(6): "virtual" elimination of the 2nd family of Lagranges when one transmits the matrix Aster to MUMPS (ELIM_LAGR2=' OUI'/"NON"), SLVK
(7): mixed accuracy (TO MIX _PRECISION=' OUI'/"NON"), SLVK
(8): use out of preconditioner single precision for the PCG ("OUI'/"NON"), SLVK
(9): management of the memory allocated by MUMPS (GESTION
_MEMOIRE=' IN_CORE'/"OUT_OF_CORE"/"AUTO"/"EVAL"), SLVK
(10): in parallel distributed, to recut with just the pieces of matrixes Aster in accordance with the perimeter of meshes for which each processor has responsibility (MATR

_DISTRIBUEE=' OUI'/"NON"), SVLK
(11): management of postprocessings (POSTTRAITEMENTS = ' SANS', "AUTO", "FORCE"), SVLK
(12): number of version of MUMPS (for example: "4.9.2" or "4.10.0"). It is about a number licit version, i.e. tested and approved by the version of Code_Aster *considered* . In the contrary case one stops in ERREUR_F before the filling of this field. Value just filled after the initialization of the occurrence MUMPS only via the routines amump */mumpu. F. SLVR

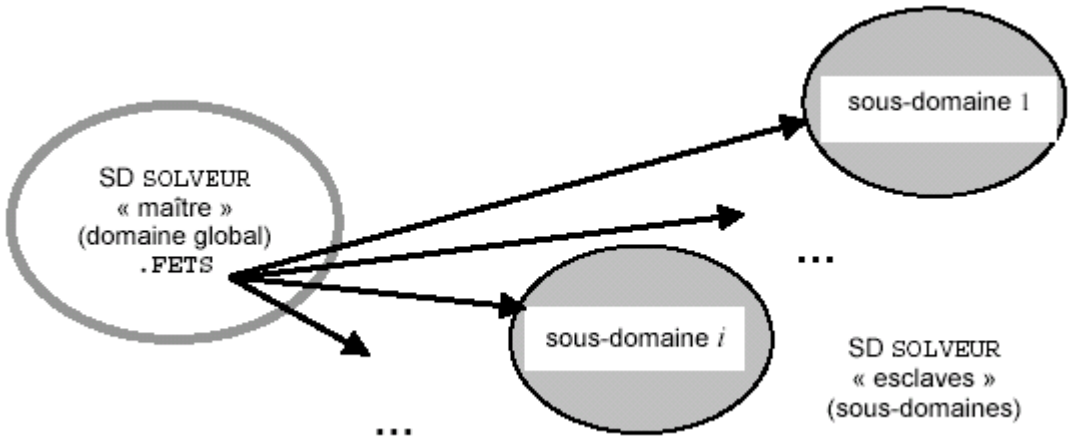
: SLVR
(1): value of FILTRAGE_MATRICE , SLVR
(2): value of RESI_RELA (computation and quality control of the solution, release of postprocessings according to the value of POSTTRAITEMENTS), SLVR
(3) with SLVR (4): unutilised. SLVI

: SLVI
(1): value of NPREC (like LDLT and MULT_FRONT), SLVI
(2): percentage of additional memory necessary to late swivellings (value of PCENT_PIVOT) SLVI
(3): value of ISTOP (like LDLT and MULT_FRONT), SLVI
(4): indicator to say to MUMPS not to store the terms of its factorized (interesting if for example, one has right need for result a total computation type for determinant or criterion for Sturm type). If it is worth 1, one does not store this not factorized (large gain report), if not one keeps it according to the standard mode. This functionality is activated only from MUMPS 4.10.0 (for the versions in on this side one does not do anything and one transmits a message UTMESS _I (if INF 0=2). Temporary value (one gives, if necessary, after use its value initial) filled in the routines vpstur.f and apchar.f . SLVI (5)
: indicator to say to MUMPS to calculate in more the determinant of the matrix. If it is worth 1, it is calculated and one stores it in temporary object "&&AMUMP. DETERMINANT" (cf amumpu. F), if not it is not calculated. This functionality is activated only from MUMPS 4.10 .0 (for the versions in on this side one stops in ERREUR_F . Temporary value (one gives, if necessary, after use its value initial) filled in the routines vpstur.f and apchar.f. SLVI (6) with SLVI
(8): unutilised . Complements in the case of

4 FETI solver Data format

4.1 the solver recursive In the case of

method FETI, the data structure solver is recursive on two levels. A SD solver "Master ", concerning the total field (SLVK (1) = ' FETI "), gathers the usual JEVEUX objects with in more one object .FETS. This last is a pointer indicating the SD solver" slaves "associated with each subdomains. These SD solver local are made up by same JEVEUX objects qu" a linear solver usual mono-field (such MULT_FRONT or LDLT) and are concerned with the same parameter setting. On the level of a subdomain, one has to be worried only inversions of local matrixes and not, for example, name of SD SD_FETI or type of reorthogonalisation installation by the total solver of interface (of which the parameter setting is contained in the SD solver" Master ") . For time, the implementation of FETI in Code_Aster presupposes that these subdomains use all the same linear solver mono-field (SLVK (1) = "MULT_FRONT" imposed by default) and with same parameter setting (RENUM and STOP_SINGULIER /NPREC). This homogeneity facilitates handling of the matrixes and the second local members. Appear 4.1-a:



Data format 4.1-asolver recursive so FETI solver Rule of naming

4.2 In the case of

one FETI solver, one chose the following rule of naming for the SD solver slaves `dependant` each on a subdomain: nom_de_la_SD_SOLVEUR

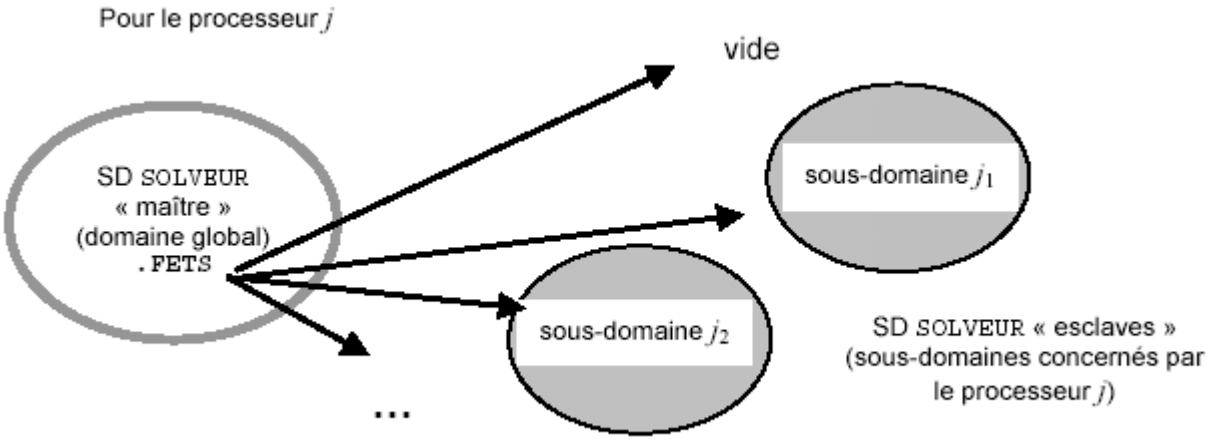
`_maître (1:11) "F" character string _de_caractères _libre (2:8) the free character string`

is generated by a call to routine GCNCON. Typical case

4.3 of parallelism MPI During an execution

in parallel mode MPI, a processor is seen allotting a certain number of subdomains (additional cf objects "&FETI.LISTE ... " `of data structure` SD_FETI [D4.06 `.21]).` The SD solver

"Master " is always built, but its pointer. FETS will indicate only the subdomains concerned with the processor running: .FETS (J K) will be `one K24 valid` that `if` the subdomain J K is in `the perimeter` of the processor J. Figure 4.3-a :



Data format 4.3-asolver recursive so FETI solver and parallelism MPI Example In

4.4 the case test

FETI002A, partitioning in four subdomains lead to the SD solver following : Built

a SD solver Master "&&OP 0046.SOLVEUR" ====> IMPR_CO

```
OF Data structure: &&OP0046.SOLVEUR????? ATTRIBUT   : F
 CONTENU:  T  BASE: >V<  MANY  OBJECTS
 (OR COLLECTIONS) FIND: 4 ============                       ==
================================================================ PRINTING OF
  THE CONTENU OF THE OBJECTS FIND:
--------------------------------------------------------------------------------
PRINTING SEGMENT
OF VALUES >&&OP0046.SOLVEUR .FETS 1 - >&&OP   0046.
      S.SOF0000000 <>&&OP0046.SOF   0000001 < 3 - >&&OP0046    .
      S.SOF0000002 <>&&OP0046.SOF   0000003 <
--------------------------------------------------------------------------------
PRINTING SEGMENT
OF VALUES >&&OP0046.SOLVEUR .SLVI < 1 -   8 100        0
      0           100
--------------------------------------------------------------------------------
PRINTING SEGMENT
OF VALUES >&&OP0046.SOLVEUR .SLVK < 1 - >   FETI <        >LUMPE
      < 3 -                    >OUI MONGREL                  <>
      < 5 -                     >NON <>                   SDFETI
      < 7                      - >GSM <>                  MULT
      < 9 -                     >CAL
--------------------------------------------------------------------------------

PRINTING SEGMENT
OF VALUES >&&OP0046.SOLVEUR .SLVR < 1 -   0.00000        D+
      00 1.00000  D-08 8.00000  D+02 1.00000  D-08 and  of the SD solver

slaves "&&OP0046.SOF0000000" ====> IMPR_CO

OF Data structure: &&OP0046.SOF0000000????? ATTRIBUT: F
 CONTENU:  T  BASE: >V<  MANY  OBJECTS
 (OR COLLECTIONS) FIND: 3 ============                       ==
================================================================ PRINTING OF
  THE CONTENU OF THE OBJECTS FIND:
--------------------------------------------------------------------------------
PRINTING SEGMENT
OF VALUES >&&OP0046.SOF0000000.SLVI 1 - 8 100 0
      0           100
--------------------------------------------------------------------------------
PRINTING SEGMENT
OF VALUES >&&OP0046.SOF0000000.SLVK < 1 - >MULT_FRO
      <>LUMPE < 3 -               >OUI MONGREL                  <>
      < 5 -                    >NON <>
      < 7 -                    >GSM <>                    MULT
      < 9 -                    >CAL
--------------------------------------------------------------------------------

PRINTING SEGMENT
OF VALUES >&&OP0046.SOF0000000.SLVR < 1 - 0.00000        D+
      00 1.00000  D-08 8.00000  D+02 1.00000  D-08 ==  ==> FIN IMPR
 _CO OF DATA STRUCTURE: &&OP0046.SOF0000000????? …
```