Some underground data structures

Abstract:
One describes some “underground” data structures here. It is those which are accessible in the code without passing in argument because their name is known a priori. This is possible only if because these `SD` are “single” (a kind of aggregate variable).

Note:

> *For better understanding the notions evoked in `the SD` underground related on parallelism and the linear solvers, one will be able to consult the documents:*
>
> *•D4.01.03 (Data format distributed and parallelism),*
> *•U2.08.06 (Note of use of parallelism),*
> *•U4.50.01 (Key word `solver`).*

Contents

1 Lists SD today

underground the indexed underground data structures aujourd” are:

“&CATASTROPHES ”	sd_cata_elem (cf D4.04.01)
“&MUMPS ”	Objects allowing L” use of MUMPS and increase of its diagnoses
“&FETI ”	Idem with FETI (cf D4.06.21)
“&CALCUL.PARALLELE ”	For the parallelization of elementary computations (routine CALCUL)
“&SSR”	Stiffness of the static macro-elements in STAT_NON_LINE or DYNA_NON_LINE
“&&FETI ”	Objects allowing L” use of FETI (D 4.06.21/D9.03.01)

2 SD “&MUMPS. ****”

Objects related on the description of the treated linear system and the monitoring of the performances of MUMPS (only if SOLVEUR/METHODE=” MUMPS” and INFO=2). They are created in CRESOL/CRSVMU.f, reset after each resolution (those concerning times and the memory) via AMUMPT . F and destroyed at the end of the operator Aster *(mechanism* automatic because of “&” initial). JEVEUX OBJECT

WKVECT	DESCRIPTION	&MUMPS.NB.MAILLE
V V I nbproc number of meshes		per processor. &MUMPS.INFO.MAILLE V V I nbproc
many terms of	the matrix by	processor (nnz local) &MUMPS.INFO.MEMOIRE V V I nbproc many *terms*
of factorized by	processor (	INFO (9) MUMPS) &MUMPS.INFO.CPU.FACS V V R nbproc TEMPS CPU + system of the phase
of factorization symbolic system	of	Code_Aster , by processor (measured via TEMPS (5) +TEMPS (6) of *UTTCPU in* NUMERO.f) &MUMPS.INFO.CPU.CAEL V V R nbproc idem for elementary computations (CALCUL.f) &MUMPS.INFO.CPU.
ASSE V V R nbproc idem	for the assemblies	(ASSMAM/VEC/MIV.f) &MUMPS.INFO.CPU.ANAL V V R nbproc
idem for the phase	of analysis of	MUMPS (AMUMPT.F) &MUMPS.INFO.CPU.FACN V V R nbproc idem *for*
the phase of numerical factorization		(AMUMPT.F) &MUMPS.INFO.CPU.SOLV V V R nbproc idem for the phase
of resolution (	AMUMPT.F) &MUMPS.INFO.MEM.EIC	V V I nbproc MUMPS Estimate (after the phase of *analysis*
) of consumption	RAM in	In-Core by processor (INFO (15)) &MUMPS.INFO.MEM. EOC V V *I*
nbproc Idem in Out-	Of-Core by processor	(INFO (17)) &MUMPS.INFO.MEM.USE V V I nbproc real Consumption (after numerical factorization) with the approach

chosen by	user	(INFO (16)) SD "&CALCUL .PARALLELE" This object is
present when	for the parallelization	of elementary computations is asked (scenarios 1b or 3of a documentation [U2.08.06] "Note of use

3 of parallelism

"). It is created and destroyed in the routine calcul.f. It is used only in routines called by CALCUL. It is a vector of Boolean . Its length is the number of elements of the "current" GREL. V (iel): .true. → the element iel must be calculated by the processor. This object is not made up with: FETI (via object "&FETI.MAILLE.NUMSD `) (scenario) the data structure sd_partition

(Doc. D4.05 .01) generated upstream and representing the distribution of

meshes by processor. In the routine

- *computation* this sd_partition underlies the parallel scenario 3a
- . One then uses L" sd_partition.NUPROC.MAILLE JEVEUX object. Thanks to this second approach, the parallelization of CALCUL is thus dissociated from that of the linear solvers who follow . Note: This object 1b is also created with FETI into sequential . SD "&&.FETI. ***" One counts

the underground objects here accompanying a resolution with the solver multi-field FETI (cf sd_FETI Doc. D4.06.21 and "Implemented

of L"

algorithm FETI' D9.03.01). These temporary objects

4 of the volatile base

can exist during a good part of a resolution FETI (i.e apart from the routine leader ALFETI.f). For the needs for monitoring "&FETI.FINF " S V K24 dim= 1 Character string to refine the monitoring of FETI [U4.50.01]. "&FETI.INFO.STOCKAGE.FIDD" S V I dim= 2 auxiliary Vector for the filling of .FVAF and .FVAL. - V (1) = subdomain

running, - V (2) = many		
subdomains	"&FETI.INFO.STOCKAGE.FVAF "	S V I dim= nb_sd+1 Numbers of components of factorized local, therefore
by subdomain. "&FETI.INFO.STOCKAGE.FVAL	" S V I dim	= nb_sd+1 Numbers of components of local matrixes "&FETI.INFO.STOCKAGE.FNBN " S V I dim= nb_sd+1 Many nodes of subdomains "
&FETI.INFO.CPU.FACN" S V R dim= nb_sd+	1 Time (obtained via	routine UTTCPU, which is thus lower than the true consumed time
(elapsed)) CPU + SYS of local		numerical factorizations. "&FETI.INFO.CPU.FACS" S V R dim= nb_sd+1

TEMPS CPU + SYS of factorizations	local symbolic systems	. “&FETI.INFO.CPU.ASSE” S V R dim= Nb
_sd+1 TEMPS CPU + SYS of	the local assemblies.	“&FETI.INFO.CPU.ELEM” S V R dim= nb_ proc1 TEMPS CPU + SYS of elementary computations by processor. Note: nb_proc 1: Many processors used in
ring MPI of FETI	(not those possibly	reserved for the OpenMP of MULT_FRONT). For
the routines of assembly	“&FETI.MAILLE.NUMSD” S V I dim=	nb_ma_tot Indicates the number of subdomain
to which a mesh of	the model belongs. Value	initialized with , that makes it possible the belonging to test

all

meshes mesh with only one subdomain (only in sequential mode . In parallel, each processor does not reach qu “ partial

information and thus these checks		
are invalid) and	D″ to assemble the local	matrixes and vectors. Note: The existence of this object is often tested to decide if -999 one uses linear solver FETI or not. For the routines of assembly in the presence of LIGREL to meshes late or of contact continuous method. LIGREL_DE_CHARGE (K19). "FEL1" S V K24 dim= nb_sd Names of projections of the ligrel on the subdomains concerned. LIGREL_DE_CHARGE (K19). "

FEL2" S

> *V Idim= 2 * number of meshes of the ligrel For the mesh: V (2 (i-1) +1) = new number in the ligrel project*

, So V (2 (i-1) +2) >0 then number of the subdomain concerned , if not - V (2 (i-1) +2) = multiplicity of the mesh		
late (DDL_IMPO on the interface	e.g.) and associated	with a .FEL4. LIGREL_DE _CHARGE (K19). "FEL3" Only so meshes
late with late nodes S	V I dim= 2 * many nodes of the ligrel For	the node $i^{ème}$ V (2 (i-1) +1) = new number in the ligrel project, So V (2 (i-1) +2) >0 then number of the subdomain concerned, if not - V (2 (i-1) +2) = multiplicity of the node late (DDL_IMPO on the interface e.g.) and associated with a .FEL5. LIGREL_
DE_CHARGE (K19). ‟FEL4″ S V I dim= 3 * number of meshes of interface potential	V (1) = last index used of the vecteurPour	$i^{ème}$ the mesh multipleV (3 (i-1) +2) = new number in the ligrel project, V (3 (I - 1) +3) = number of the subdomain concerned, - V (3 (i-1) +4) = old number. LIGREL_DE_CHARGE (K19). ‟FEL5″ Only so meshes late with late nodes ‟S
V I dim= 3 * many nodes D	″ interface potential V (1) = last index used of the vecteurPour	the node multipleV (3 (i-1) +2) = new number $i^{ème}$ in the ligrel project, V (3 (i-1) +3) = number of the subdomain concerned , - V (3 (i-1) +4) = old number. Rem arches: At the beginning of computation , this LIGREL
comprises only the meshes physical ones with physical nodes, then it understands nothing any more	but the meshes late ones (always with physical nodes).	For parallelism MPI "& FETI.LISTE.SD.MP I" $i^{ème}$ S V I dim= nb_sd+1 Indicates in the loops on the subdomains

		, if the processor running is concerned with the subdomain : V (i+1) = 1

⇒ **the loop**

on this subdomain is carried out , V (i+1) = 0 ⇒ it is jumped. With convention of the loops, V (1) concerns the total field and is always worth. Into sequential, V (I) =

1 for all. “& FETI.LISTE.SD.MP		
IB″ S V I dim	= nb_sd opposite Object	of the precedent V (I) = J ⇒ subdomain I am concerned with the processor. Into sequential $i^{ème}$, V (I) = 0 for all. For reorthogonalisations “&FETI.PAS.TEMPS” S V I dim=3 Object for the procedure d'acceleration: V (1) =NB_REORTHO _ INST (number of time step retained for 1 acceleration) V (2) = index of time step i
V (3) = not activated for	time ‶	&FETI.MULTIPLE.SM.K24” “& FETI.MULTIPLE.SM.IN ″ S V K24 S V I dim= NB_REORTHO_INST+1 Objects of management j of the directions of descents for i

accelerations enters time step		
. Various (except exception	, the life duration	of these objects is restricted with routine RESOUD/ALFETI) “&&FETI.MULT” S V I dim=neq (size of the total problem) It is during FETI of the .CONL. It gives the geometrical
multiplicity mult_j degrees of freedom of the total	problem . That serves to save time	during the reconstruction as the field solution. This vector is used during all the operator

(of NUMERO with RESOUD). “&FETI.ITERATION.RESIDU” V V R dim=niter_max (number iterations max) Vector storing		
the residue with	each iteration for monitoring and writing	file (if INFO_FETI (13 : 13) = ' T `). “&&FETI.ALPHA.MCR” V V R dim=nbi (nodes number of interface) Vector of amplitude of the rigid modes (cf R6.01.03 chap4.2). “&&FETI.COLLECTION* “XC V I/R dim=nbsd (many subdomains) Collections making it possible to make the joint enters
the local numbers to each	SD and their total classification. “&FETI.CRITER.CRTI”	V V I dim=1 SD interns with STAT_NON_LINE to trace the column of iterations FETI in table D” evolution
of computation. Its use	lasts all the STAT_NON_LINE. “&&FETI.LAGR.INTERFACE” “&&	FETI.RESIDU.R” “&&FETI.REPROJ.G” “&&FETI.REPCPJ.H” “&&FETI.FIDDZ” “ &&FETI.VECNBI . *”
V V R dim=nbi auxiliary	Vectors of computation (linear algebra)	. “&&FETI.FET*.AUX” V V I Vectors containing of addresses JEVEUX to avoid expensive JEVEUO in routines
FET*. “& &FETI.GGT.V*”	“& &FETI.GI.R”	“ &&FETI.GITGI.R” “&& FETINL.E.R” V V R auxiliary Objects to project on the coarse problem. “&&FETI.LAPACK.IPIV” V V S dim=dimGI (dimension of the coarse problem) Vector of
swivelling for routines LAPACK carrying out the resolutions of systems linear of the coarse problem. “ &&FETI.MMA.R*” V V R dim= nbi auxiliary	Objects	for accelerations between successive resolutions
(key word ACCELERATION_SM		). “&&FETI.TEST*” V V R /L auxiliary Objets to test the definite one - positivity of L”
op. of interface FETI (cf key word INFO_FETI). Note: During a parallel	execution	, these temporary objects are declined by processor
. However, according to the distribution	of load, each processor is concerned	only by certain subdomains (cf objects “ &FETI.LISTE…”). Therefore, put except for these the last two JEVEUX objects,
the other related objects	contain	only information relating to the subdomains which interest them. For example , object SDFETI
(1:8)/“.MAILLE .NUMSD	”	will comprise initialized values with

		-999 for meshes of the subdomains concerning the other processors

.