Quelques structures de données souterraines

Résumé :
On décrit ici quelques structures de données “souterraines”. Ce sont celles qui sont accessibles dans le code sans passer en argument parce que leur nom est connu a priori. Ceci n’est possible que parce que ces SD sont “uniques” (une sorte de variable globale).

Remarque :

Pour mieux comprendre les notions évoquées dans les SD souterraines liées au parallélisme et aux solveurs linéaires, on pourra consulter les documents:

- *D4.01.03 (Structure de données distribuées et parallélisme),*
- *U2.08.06 (Notice d’utilisation du parallélisme),*
- *U4.50.01 (Mot-clé SOLVEUR).*

Table des Matières

1 Liste des SD souterraines

Les structures de données souterraines répertoriées aujourd'hui sont :

'&CATA '	sd_cata_elem (cf. D4.04.01)
'&MUMPS '	Objets permettant l'utilisation de MUMPS et la remontée de ses diagnostics
'&FETI '	Idem avec FETI (cf. D4.06.21)
'&CALCUL.PARALLELE '	Pour la parallélisation des calculs élémentaires (routine CALCUL)
'&SSR'	Rigidité des macro-éléments statiques dans STAT_NON_LINE ou DYNA_NON_LINE
'&&FETI '	Objets permettant l'utilisation de FETI (D4.06.21/D9.03.01)

2 SD '&MUMPS.****'

Objets liés à la description du système linéaire traité et au monitoring des performances de MUMPS (uniquement si SOLVEUR/METHODE='MUMPS' et INFO=2). Ils sont créés dans CRESOL/CRSVMU.f, réinitialisés après chaque résolution (ceux concernant les temps et la mémoire) via AMUMPT.F et détruits en fin d'opérateur *Aster* (mécanisme automatique du fait du « & » initial).

OBJET JEVEUX	WKVECT	DESCRIPTION
&MUMPS.NB.MAILLE	V V I nbproc	nombre de mailles par processeur.
&MUMPS.INFO.MAILLE	V V I nbproc	nombre de termes de la matrice par processeur *(nnz local)*
&MUMPS.INFO.MEMOIRE	V V I nbproc	nombre de termes de la factorisée par processeur (INFO(9) MUMPS)
&MUMPS.INFO.CPU.FACS	V V R nbproc	temps CPU + système de la phase de factorisation symbolique de *Code_Aster* , par processeur (mesurés via TEMPS(5)+TEMPS(6) de UTTCPU dans NUMERO.f*)*
&MUMPS.INFO.CPU.CAEL	V V R nbproc	idem pour les calculs élémentaires (CALCUL.f*)*
&MUMPS.INFO.CPU.ASSE	V V R nbproc	idem pour les assemblages (ASSMAM/VEC/MIV.f *)*
&MUMPS.INFO.CPU.ANAL	V V R nbproc	idem pour la phase d'analyse de MUMPS (AMUMPT.F)
&MUMPS.INFO.CPU.FACN	V V R nbproc	idem pour la phase de factorisation numérique (AMUMPT.F*)*
&MUMPS.INFO.CPU.SOLV	V V R nbproc	idem pour la phase de résolution (AMUMPT.F *)*
&MUMPS.INFO.MEM.EIC	V V I nbproc	Estimation MUMPS (après la phase d'analyse) de la consommation RAM en In-Core par processeur (INFO(15))
&MUMPS.INFO.MEM.EOC	V V I nbproc	Idem en Out-Of-Core par processeur (INFO(17))
&MUMPS.INFO.MEM.USE	V V I nbproc	Consommation réelle (après le factorisation numérique) avec l'approche choisie par l'utilisateur (INFO(16))

3 SD '&CALCUL.PARALLELE'

Cet objet est présent lorsque l'on demande la parallélisation des calculs élémentaires (scénarios 1b ou 3a de la documentation [U2.08.06] "Notice d'utilisation du parallélisme"). Il est créé et détruit dans la routine calcul.f. Il n'est utilisé que dans des routines appelées par CALCUL. C'est un vecteur de booléens. Sa longueur est le nombre d'éléments du GREL "courant".

V(iel) : .true. → l'élément iel doit être calculé par le processeur.

Cet objet n'est constitué avec:

- FETI (*via* l'objet '&FETI.MAILLE.NUMSD') (scénario 3a)
- La structure de données sd_partition (doc D4.05.01) générées en amont et représentant la distribution des mailles par processeur. Dans la routine calcul cette sd_partition sous-tend le scénario parallèle 1b . On utilise alors l'objet JEVEUX sd_partition.NUPROC.MAILLE.

Grâce à cette deuxième approche, la parallélisation de CALCUL est donc dissociée de celle des solveurs linéaires qui suivent.

Remarque :

Cet objet est aussi créé avec FETI en séquentiel.

4 SD '&&.FETI.***'

On recense ici les objets souterrains accompagnant une résolution avec le solveur multi-domaine FETI (cf. sd_FETI doc. D4.06.21 et 'Mise en œuvre de l'algorithme FETI' D9.03.01). Ces objets temporaires de la base volatile peuvent exister durant une bonne partie d'une résolution FETI (i.e. en dehors de la routine chef d'orchestre ALFETI.f).

Pour les besoins du monitoring		
'&FETI.FINF'	S V K24 dim= 1	Chaîne de caractère pour affiner le monitoring de FETI [U4.50.01].
'&FETI.INFO.STOCKAGE.FIDD'	S V I dim= 2	Vecteur auxiliaire pour le remplissage de .FVAF et .FVAL. - V(1) = sous-domaine courant, - V(2) = nombre de sous-domaines
'&FETI.INFO.STOCKAGE.FVAF'	S V I dim= nb_sd+1	Nombres de composantes des factorisées locales, donc par sous-domaine.
'&FETI.INFO.STOCKAGE.FVAL'	S V I dim= nb_sd+1	Nombres de composantes des matrices locales
'&FETI.INFO.STOCKAGE.FNBN'	S V I dim= nb_sd+1	Nombres de nœuds des sous-domaines
'&FETI.INFO.CPU.FACN'	S V R dim= nb_sd+1	Temps (obtenu via la routine UTTCPU, qui est donc inférieur au véritable temps consommé (elapsed)) CPU + SYS des factorisations numériques locales.
'&FETI.INFO.CPU.FACS'	S V R dim= nb_sd+1	Temps CPU + SYS des factorisations symboliques locales.
'&FETI.INFO.CPU.ASSE'	S V R dim= nb_sd+1	Temps CPU + SYS des assemblages locaux.

'FETI.INFO.CPU.ELEM'	S V R dim= nb_proc1	Temps CPU + SYS des calculs élémentaires par processeur.

Remarque :

nb_proc1 : Nombre de processeurs utilisés dans l'anneau MPI de FETI (pas ceux éventuellement réservés pour l' OpenMP de MULT_FRONT).

Pour les routines d'assemblage		
'&FETI.MAILLE.NUMSD'	S V I dim= nb_ma_tot	Indique le numéro de sous-domaine auquel appartient une maille du modèle. Valeur initialisée à -999, cela permet de tester l'appartenance de toutes les mailles du maillage à un seul sous-domaine (uniquement en mode séquentiel. En parallèle, chaque processeur n'accède qu'à une information partielle et donc ces vérifications sont invalides) et d'assembler les matrices et vecteurs locaux.

Remarque :

L'existence de cet objet est souvent testée pour décider si on utilise le solveur linéaire FETI ou non.

Pour les routines d'assemblage en présence de LIGREL à mailles tardives ou de contact méthode continue.		
LIGREL_DE_CHARGE(K19).'FEL1'	S V K24 dim= nb_sd	Noms des projections du ligrel sur les sous-domaines concernés.
LIGREL_DE_CHARGE(K19).'FEL2'	S V I dim= 2 * nombre de mailles du ligrel	Pour la $i^{ème}$ maille : V(2(i-1)+1)= nouveau numéro dans le ligrel projeté, Si V(2(i-1)+2)>0 alors numéro du sous-domaine concerné, sinon -V(2(i-1)+2) = multiplicité de la maille tardive (DDL_IMPO sur l'interface par ex.) et associé à un .FEL4.
LIGREL_DE_CHARGE(K19).'FEL3' Uniquement si mailles tardives à nœuds tardifs	S V I dim= 2 * nombre de nœuds du ligrel	Pour le $i^{ème}$ nœud V(2(i-1)+1)= nouveau numéro dans le ligrel projeté, Si V(2(i-1)+2)>0 alors numéro du sous-domaine concerné, sinon -V(2(i-1)+2)= multiplicité du nœud tardif (DDL_IMPO sur l'interface par ex.) et associé à un .FEL5.
LIGREL_DE_CHARGE(K19).'FEL4'	S V I dim= 3 * nombre de mailles d'interface potentielles	V(1)= dernier indice utilisé du vecteur Pour la $i^{ème}$ maille multiple V(3(i-1)+2)= nouveau numéro dans le ligrel projeté, V(3(i-1)+3)= numéro du sous-domaine concerné, -V(3(i-1)+4)= ancien numéro.
LIGREL_DE_CHARGE(K19).'FEL5' Uniquement si mailles tardives à nœuds tardifs`	S V I dim= 3 * nombre de nœuds d'interface potentiels	V(1)= dernier indice utilisé du vecteur Pour le $i^{ème}$ nœud multiple V(3(i-1)+2)= nouveau numéro dans le ligrel projeté, V(3(i-1)+3)= numéro du sous-domaine concerné, -V(3(i-1)+4)= ancien numéro.

Rem arque :

En début de calcul, ce LIGREL ne comporte que des mailles physiques à nœuds physiques, puis il ne comprend plus que des mailles tardives (toujours à nœuds physiques).

Pour le parallélisme MPI		
'&FETI.LISTE.SD.MPI '	S V I dim= nb_sd+1	Indique dans les boucles sur les sous-domaines, si le processeur courant est concerné par le $i^{ème}$ sous-domaine : V(i+1)= 1 ⇒ la boucle sur ce sous-domaine est effectuée, V(i+1)= 0 ⇒ elle est sautée. Par convention des boucles, V(1) concerne le domaine global et vaut toujours 1 . En séquentiel, V(i)= 1 pour tout i .
'&FETI.LISTE.SD.MPIB '	S V I dim= nb_sd	Objet inverse du précédent V(i)= j ⇒ le sous-domaine i est concerné par le processeur j . En séquentiel, V(i)= 0 pour tout i .

Pour les réorthogonalisations		
'&FETI.PAS.TEMPS '	S V I dim=3	Objet pour la procédure d'accélération: V(1)=NB_REORTHO_INST (nombre de pas de temps retenu pour l'accélération) V(2)= indice du pas de temps V(3)= non activé pour l'instant
'&FETI.MULTIPLE.SM.K24' '&FETI.MULTIPLE.SM.IN '	S V K24 S V I dim= NB_REORTHO_INST+1	Objets de gestion des directions de descentes pour les accélérations entre pas de temps.

Divers (sauf exception, la durée de vie de ces objets est limitée à la routine RESOUD/ALFETI)		
'&&FETI.MULT '	S V I dim=neq (taille du problème global)	C'est le pendant FETI du .CONL. Il donne la multiplicité géométrique mult_j des degrés de liberté du problème global. Cela sert à gagner du temps lors de la reconstruction du champ solution. Ce vecteur est utilisé durant tout l'opérateur (de NUMERO à RESOUD).
`&FETI.ITERATION.RESIDU '	V V R dim=niter_max (nombre itérations max)	Vecteur stockant le résidu à chaque itération pour monitoring et écriture fichier (si INFO_FETI(13:13)='T').
`&&FETI.ALPHA.MCR '	V V R dim=nbi (nombre nœuds d'interface)	Vecteur d'amplitude des modes rigides (cf. R6.01.03 chap4.2).
`&&FETI.COLLECTION* '	XC V I/R dim=nbsd (nombre de sous-domaines)	Collections permettant de faire la jointure entre les numéros locaux à chaque SD et leur numérotation globale.
`&FETI.CRITER.CRTI '	V V I dim=1	SD interne à STAT_NON_LINE pour tracer la colonne des itérations FETI dans le tableau d'évolution du calcul. Son utilisation dure tout le STAT_NON_LINE.
`&&FETI.LAGR.INTERFACE' '&&FETI.RESIDU.R' '&&FETI.REPROJ.G' '&&FETI.REPCPJ.H' '&&FETI.FIDDZ' '&&FETI.VECNBI.* '	V V R dim=nbi	Vecteurs auxiliaires de calcul (algèbre linéaire).
'&&FETI.FET*.AUX '	V V I	Vecteurs contenant des adresses JEVEUX pour éviter de couteux JEVEUO dans les routines FET*.
'&&FETI.GGT.V*' '&&FETI.GI.R' '&&FETI.GITGI.R' '&&FETINL.E.R '	V V R	Objets auxiliaires pour projeter sur le problème grossier.
'&&FETI.LAPACK.IPIV '	V V S dim=dimGI (dimension du problème grossier)	Vecteur de pivotage pour les routines LAPACK effectuant les résolutions de systèmes linéaires du problème grossier.
'&&FETI.MMA.R* '	V V R dim=nbi	Objets auxiliaires pour les accélérations entre résolutions successives (mot-clé ACCELERATION_SM).
'&&FETI.TEST* '	V V R/L	Objets auxiliaires pour tester la définie-positivité de l'op. d'interface FETI (cf. mot-clé INFO_FETI).

Remarques :

Lors d'une exécution parallèle, ces objets temporaires sont déclinés par processeur. Or, suivant la répartition de charge, chaque processeur n'est concerné que par certains sous-domaines (cf. objets '&FETI.LISTE…'). Donc, mis à part ces deux derniers objets JEVEUX , les autres objets connexes ne contiennent que les informations relatives aux sous-domaines qui les intéressent.

Par exemple, l'objet SDFETI(1:8)//'.MAILLE.NUMSD ' comportera des valeurs initialisées à -999 pour les mailles des sous-domaines concernant les autres processeurs.