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).*

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Table des Matières

1 Liste des SD souterraines....3
2 SD '&MUMPS.****'....3
3 SD '&CALCUL.PARALLELE'....4

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

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
'&CALCUL.PARALLELE '	Pour la parallélisation des calculs élémentaires (routine CALCUL)
'&SSR'	Rigidité des macro-éléments statiques dans STAT_NON_LINE ou DYNA_NON_LINE

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))

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

3 SD '&CALCUL.PARALLELE'

Cet objet est présent lorsque l'on demande la parallélisation des calculs élémentaires. 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.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)