Organisme(s) : EDF-R&D/SINETICS

Manuel de Descriptif Informatique
Fascicule D9.03 : -
Document : D9.03.01

Mise en œuvre de l'algorithme FETI

Résumé :

On décrit ici la mise en œuvre informatique de l'algorithme de résolution de systèmes linéaires FETI. On reprend les notations des notes de Référence [R6.01.03] [bib1] et on s'appui aussi sur celles d'Utilisation [U4.50.01] et de Développement [D4.06.05], [D4.06.07], [D4.06.10], [D4.06.11] et [D4.06.21]. On trouvera dans ce document l'organigramme simplifié du processus de résolution, en mode séquentiel comme en parallèle, permettant de discerner ses principales articulations logiques, son arbre d'appel, les principales variables incriminées ainsi que leurs contenus. Les spécificités et la philosophie de parallélisme mis en place sont particulièrement détaillées.
On a par contre choisit de ne pas alourdir l'exposé en mentionnant les appels aux routines superviseur, à celles de gestion d'objets JEVEUX, de manipulations de bas niveau (`VTDEFS, VTCREB...`) et les détails des routines préliminaires au déroulement de l'algorithme FETI (`ASSMAM`, `MEACMV`, `MERESO...`).

1 Organigramme simplifié

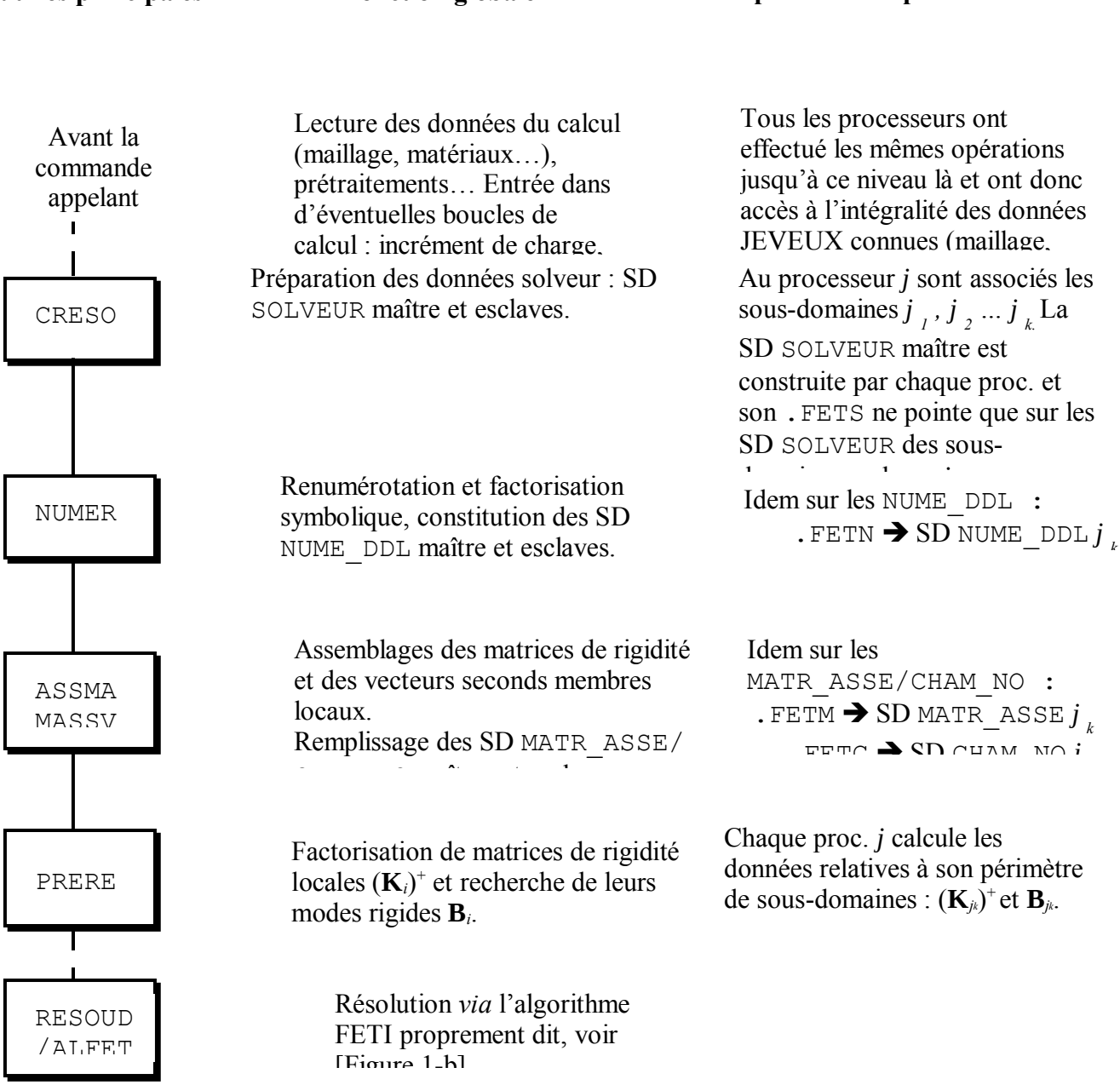

Figure 1-a : Organigramme simplifié avant ALFETI

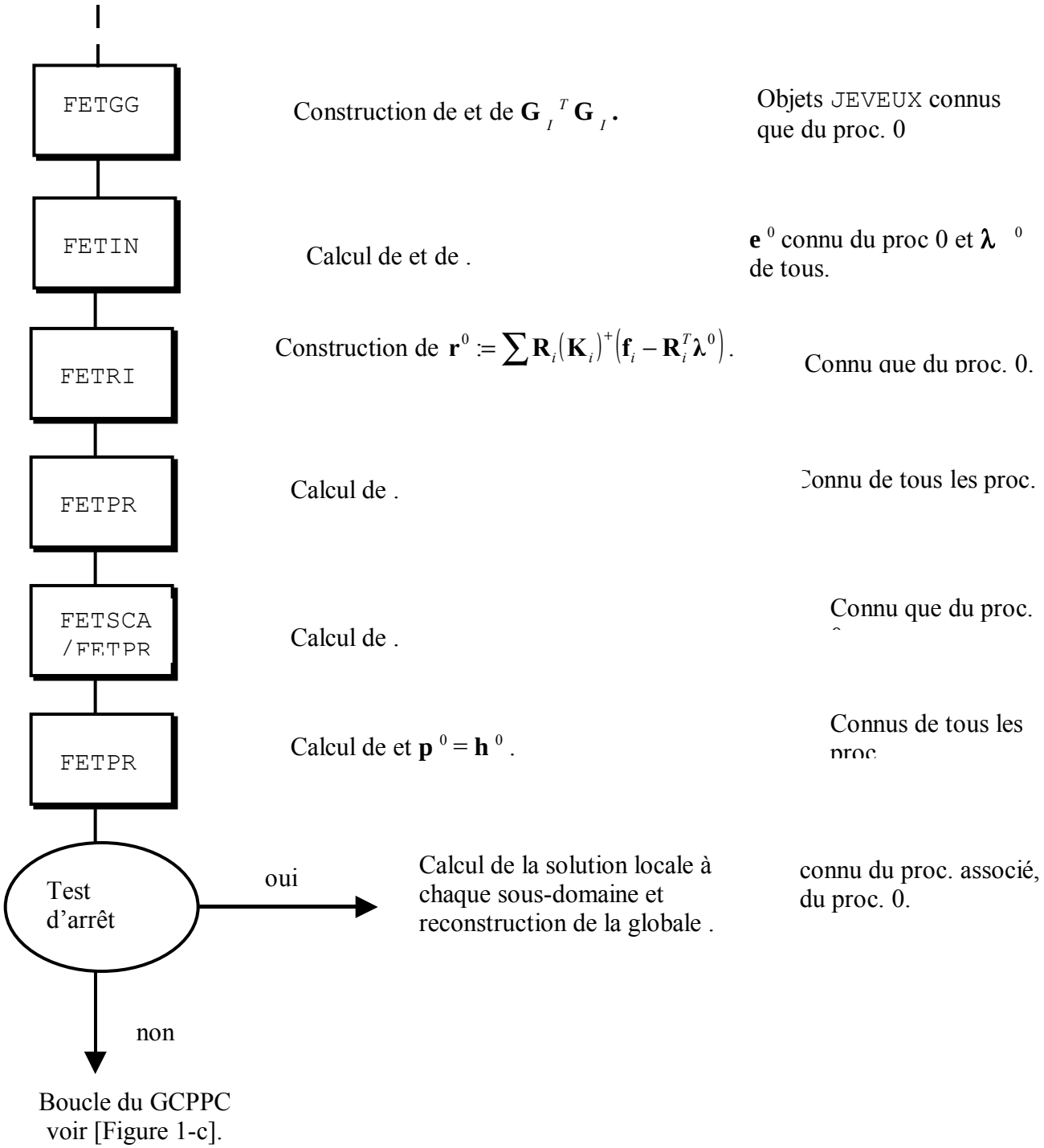

Figure 1-b : Organigramme simplifié dans ALFETI (niveau 1)

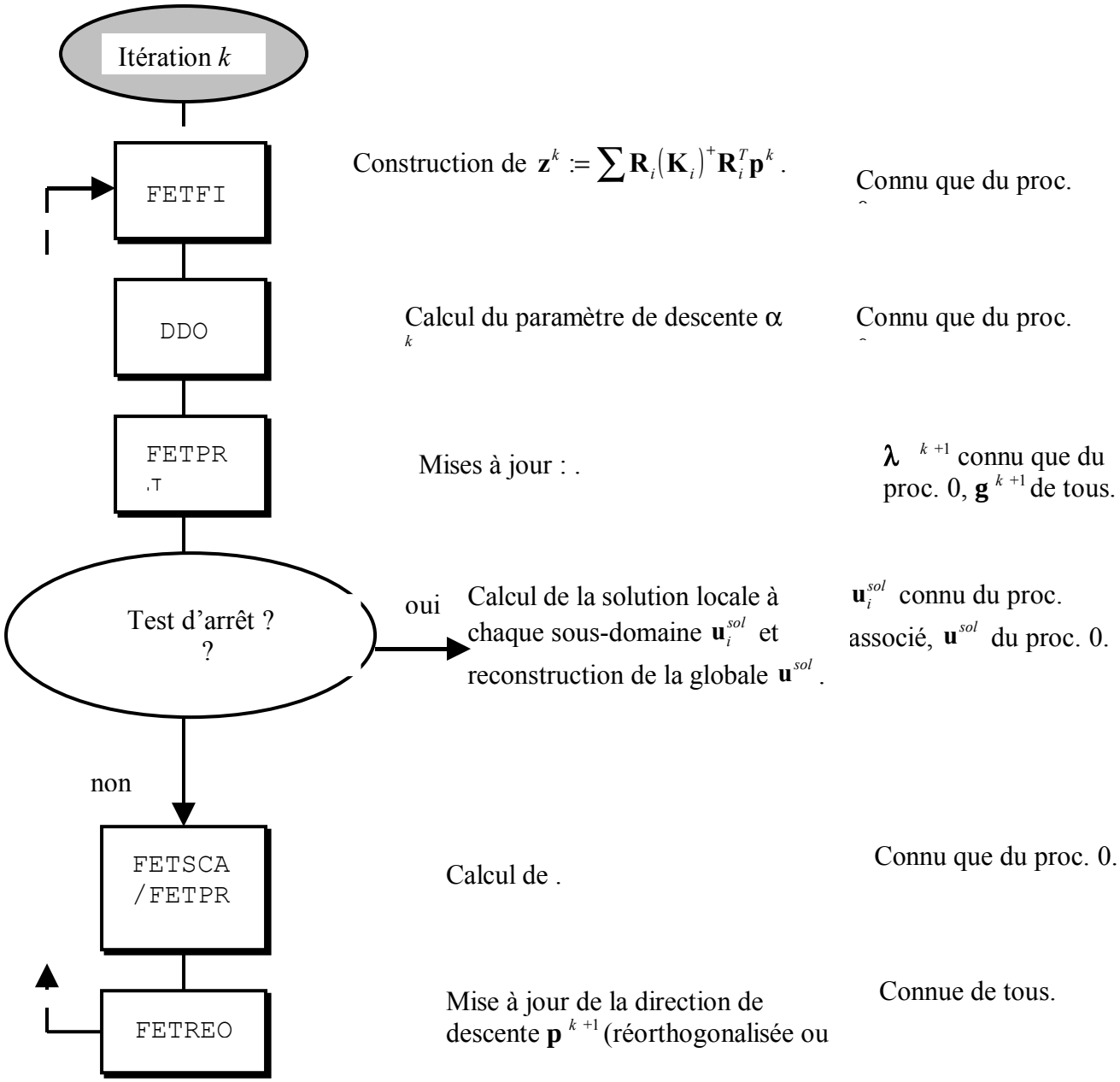

Figure 1-c : Organigramme simplifié dans ALFETI (niveau 2)

2 Organigramme détaillé et arbre d'appel

Routine appelante	Routines appelées niveau 1	Routines appelées niveau 2	Détails	Informations relatives aux processeurs
Avant la commande appelant FETI (MECA_STATIQUE …)			Lecture des données du calcul (maillage, matériaux, SD_FETI [D4.06.21]…), prétraitements… Entrée dans d'éventuelles boucles de calcul : incrément de charge, pas de Newton…	Tous les proc. ont effectué les mêmes opérations jusqu'à ce niveau et ont donc accès à l'intégralité des données JEVEUX connues (maillage, champs issus de prétraitements…).
CRESOL			• Lecture des paramètres inhérents aux mot-clé facteur SOLVEUR [U4.50.01], • Création et initialisation des objets connexes '&FETI.INFO…' (pour le monitoring [D4.06.21 §4]) et SDFETI(1:8)//'.MAILLE.NUMSD' (pour les routines d'assemblage [D4.06.21 §4]), • Création du pointeur SOLVEUR_maître.FETS vers les SOLVEUR esclaves.	O*/O P/P P/P La SD SOLVEUR maître est construite par chaque proc. et son .FETS ne pointera que sur les SD SOLVEUR des sous-domaines esclaves j_k
Remarque : * Le code O/N signifie que l'opération est réalisée, ou que la variable est connue, seulement du proc. maître (O pour Oui) mais pas des autres proc. (N pour Non). On utilise aussi P (Partiellement) pour notifier que le réalisé ne concerne que les sous-domaines du périmètre du proc. courant.				
	FETMPI		• Répartition des sous-domaines par proc. et détermination du nombre de proc.	MPI_COMM_SIZE MPI_COMM_RANK
Boucle sur les sous-domaines concernés**	CRESO1		• Constitution des SD SOLVEUR [D4.06.11] des sous-domaines (« esclaves »), • Vérifications mailles du	P/P En séquentiel

Routine appelante	Routines appelées niveau 1	Routines appelées niveau 2	Détails	Informations relatives aux processeurs
par le proc. courant Fin de boucle			modèle/mailles des sous-domaines.	
	CRESO1		• Constitution de la SD SOLVEUR du domaine global (« maître »), • Vérifications mailles du modèle/mailles des sous-domaines.	O/O En séquentiel
Remarque : ** En séquentiel, tous les sous-domaines sont concernés par le processeur courant qui est aussi le processeur maître ou proc. 0. En parallèle, le proc. courant *j* se voit attribué un ensemble de sous-domaines contiguës : $j_1, j_2 \ldots j_k$. Dans les boucles sur les sous-domaines, cette information est formulée *via* l'objet JEVEUX '&FETI.LISTE.SD.MPI' ([D4.06.21 §4]) qui filtre les indices de boucle.				
NUMERO			• Constitution de la liste de charges globales à tout le modèle.	O/O
	NUMER2		• Constitution du NUME_DDL [D4.06.07] maître et de son pointeur .FETN.	O/O Le .FETN ne pointera que sur les SD des sous-domaines esclaves j_k
	NUMER2	NUEFFE	• Création du NUME_EQUA maître.	O/O
	NUMER2	PROFMA	• On ne créé pas la SD STOCKAGE maître.	O/O
			• Vérification de la cohérence de la SD_FETI vis-à-vis du modèle et des charges (piloté par le mot-clé VERIF_SDFETI).	O/O
	FETMPI		• Détermination du nombre de proc. et du rang du proc. courant.	MPI_COMM_SIZE MPI_COMM_RANK
Boucle sur les sous-domaines concernés par le proc. courant	EXLIM1		• Création du LIGREL des mailles physiques du sous-domaine.	P/P
	EXLIM2		• Constitution de la liste des LIGREL de charge (à mailles tardives) impactant le sous-domaine, • Leurs projections éventuelles sur plusieurs sous-domaines. Remplissage *ad hoc* des SD .FEL*i* associées à ces projections.	P/P P/P
	NUMER2		• Constitution du NUME_DDL esclave.	P/P
	NUMER2	NUEFFE	• Création du NUME_EQUA esclave.	P/P

Routine appelante	Routines appelées niveau 1	Routines appelées niveau 2	Détails	Informations relatives aux processeurs
Fin de boucle	NUMER2	PROFMA	• On créé la SD STOCKAGE esclave.	P/P
			• Test de l'identité du PROF_CHNO.NUEQ ([D4.06.07 §5.3])	O/N
ASSMAM/ ASSVEC			• Constitution des MATR_ASSE/CHAM_NO [D4.06.10], [D4.06.05] maîtres et de leurs pointeurs .FETM/.FETC. On ne construit pas la MATR_ASSE.VALE inutile.	O/O Leurs .FETM/.FETC ne pointeront que sur les SD des sous-domaines esclaves j_k
Boucle sur les sous-domaines concernés par le proc. courant Fin de boucle			Constitution des MATR_ASSE/CHAM_NO esclaves en s'appuyant sur les objets auxiliaires : • SDFETI(1:8)//'.MAILLE.NUMSD' qui détermine dans les boucles sur les données globales si la maille considérée intéresse le sous-domaine, • Les .FEL*i* pour pouvoir faire la jointure entre la numérotation de mailles ou de nœuds tardifs et leur numérotation locale au sous-domaine (celle du NUME_DDL esclave).	P/P
PRERES			• Mise à jour du champ 'DOCU' du MATR_ASSE maître.REFA pour franchir RESOUD et la recopie du second membre en vecteur solution.	O/O
Boucle sur les sous-domaines concernés par le proc. courant	FETFAC	TLDLG2	• Remplissage des MATR_ASSE esclaves.VALF des sous-domaines associés au proc. courant : $(\mathbf{K}_i)^+$, • Calcul des modes rigides et remplissage des objets temporaires '&&FETFAC.FETI.MOCR' (modes rigides) et '&&FETFAC.FETI.INPN' (indice des pivots quasi-nulles), • Vérifications des modes rigides et d'une des conditions de Moore-Penrose (si INFO_FETI(6:6)='T' activé [U4.50.01 §3.5])	P/P P/P P/P
Fin de boucle	FETFAC		• Remplissage des objets MATR_ASSE.FETF, .FETP et .FETR ([D4.06.10 §3]) : $\mathbf{B}_i$.	P/P
RESOUD	Boucle sur		• Vérifie que le PROF_CHNO de la	P/P

Routine appelante	Routines appelées niveau 1	Routines appelées niveau 2	Détails	Informations relatives aux processeurs
	les sous-domaines concernés par le proc. courant Fin de boucle		MATR_ASSE est identique à celui du second membre (pour le maître et les SD esclaves), • Vérifie que la MATR_ASSE globale et ses MATR_ASSE esclaves ont bien été factorisées.	P/P
	RESFET	Boucle sur les sous-domaines concernés par le proc. courant Fin de boucle	• Mise à jour des objets temporaires &INT des MATR_ASSE locales.	P/P
		ALFETI	• Algorithme FETI proprement dit, voir tableaux suivants [Tableau 2-2] et [Tableau 2-3].	

Tableau 2-1 : Organigramme détaillé et arbre d'appel avant ALFETI

Routine appelante	Routines appelées niveau 1	Routines appelées niveau 2	Détails	Informations relatives aux processeurs
ALFETI	FETMPI		• Détermination du nombre de proc. et du rang du proc. courant.	MPI_COMM_SIZE MPI_COMM_RANK
	Boucle sur les sous-domaines concernés par le proc. courant Fin de boucle		• Initialisation des collections de vecteurs dimensionnés au nombre de DDLs (physiques et tardifs) de chaque sous-domaines : '&&FETI.COLLECTIONR' et '&&FETI.COLLECTIONL'. Elles serviront aux opérations matricielles de la taille des problèmes locaux (la seconde est limitée au produit matrice-vecteur du préconditionnement).	P/P
	FETING	Boucle sur les sous-domaines concernés par le proc. courant Fin de boucle	• Constitution de la collection de vecteurs dimensionnés au nombre de Lagranges d'interface des sous-domaines '&&FETI.COLLECTIONI'. Elle sert à faire la jointure entre la numérotation d'un Lagrange d'interface dans la liste des nœuds d'un sous-domaines (SDFETI.FETB) et celle du	P/P

Routine appelante	Routines appelées niveau 1	Routines appelées niveau 2	Détails	Informations relatives aux processeurs
			PROF_CHNO local issu de la factorisation symbolique.	
		Boucle sur les sous-domaines concernés par le proc. courant Fin de boucle	• Initialisation d'objets temporaires liés à la réorthogonalisation (REORTH, NBREOR…), de vecteurs temporaires (K24IRR, K24LAI…), • Création d'objets temporaires pour économiser plus tard des appels JEVEUO : K24REX, K24FIV....	O/O P/P
	FETMPI		• Réduction puis diffusion globale de l'objet MATR_ASSE maître.FETF pour que tous les proc. sachent combien de modes rigides possède un sous-domaine donné.	En parallèle (testé par la présence d'au moins 2 processeurs) O/O MPI_ALLREDUCE+ MPI_SUM
	FETMPI		• Idem pour l'objet de monitoring '&FETI.INFO.STOCKAGE.FVAL', • Tant qu'à synchroniser on fait la même chose, sans diffusion globale, pour les autres objets de monitoring '&FETI.INFO…'.	En parallèle O/O MPI_ALLREDUCE+ MPI_SUM O/N MPI_REDUCE+ MPI_SUM
	FETGGT	Boucle sur les sous-domaines concernés par le proc. courant Fin de boucle FETREX	• Construction de la matrice rectangulaire $\mathbf{G}_I := \left[\mathbf{R}_1\mathbf{B}_1 \quad \ldots \quad \mathbf{R}_Q\mathbf{B}_Q\right]$ (NOMGI='&&FETI.GI.R').	P/P
	FETGGT	FETMPI	• Construction du $\mathbf{G}_I$ complet par collecte sélective vers le proc. 0. **ATTENTION**, c'est ici qu'interviennent les contraintes : STOCKAGE_GI='OUI' obligatoire en parallèle et distribution des sous-domaines de manière contiguë par proc.	En parallèle O/N MPI_GATHERV
	FETGGT	BLAS DDOT	• Construction de la matrice carrée $\mathbf{G}_I^T\mathbf{G}_I$ (NOMGGT='&&FETI.GITGI.R').	Si proc 0 O/N
	FETMON		• Monitoring si INFOFE(9:9)='T' : tailles des sous-domaines, profiling de leurs temps CPU d'assemblage, de	Si proc 0 O/N

Routine appelante	Routines appelées niveau 1	Routines appelées niveau 2	Détails	Informations relatives aux processeurs
			factorisation…	
	FETINL	Boucle sur les sous-domaines concernés par le proc. courant Fin de boucle	• Construction du vecteur $\mathbf{e}^0 := \left[\mathbf{f}_1^T\mathbf{B}_1 \quad \ldots \quad \mathbf{f}_Q^T\mathbf{B}_Q\right]^T$ (K24ER='&&FETINL.E.R').	P/P
	FETINL	FETMPI	• Construction du $\mathbf{e}^0$ complet par réduction vers le proc. 0.	En parallèle O/N MPI_REDUCE+ MPI_SUM
	FETINL	FETREX et BLAS DAXPY (si STOCKAGE_GI ='OUI'), LAPACK DSPTRF/S	• Calcul du vecteur Lagrange d'interface initial $\boldsymbol{\lambda}^0 := \mathbf{G}_I\left[\left(\mathbf{G}_I{}^T\mathbf{G}_I\right)^{-1}\mathbf{e}\right]$ (VLAGI/K24LAI/ZR(IVLAGI)).	Si proc 0 O/N
	FETINL		• Distribution de $\boldsymbol{\lambda}^0$ à tous les proc.	En parallèle O/O MPI_BCAST
	FETRIN OPTION=1	Boucle sur les sous-domaines concernés par le proc. courant, BLAS DAXPY, FETREX, RLTFR8, Fin de boucle	• Calcul du résidu initial $\mathbf{r}^0 := \sum \mathbf{R}_i\left(\mathbf{K}_i\right)^+\left(\mathbf{f}_i - \mathbf{R}_i^T\boldsymbol{\lambda}^0\right)$ (K24IRR='&&FETI.RESIDU.R'/ ZR(IRR))	P/P
	FETRIN OPTION=1	FETMPI	• Construction du $\mathbf{r}^0$ complet par réduction vers le proc. 0.	En parallèle O/N MPI_REDUCE+ MPI_SUM
	FETPRJ OPTION=1	BLAS DGEMV/ DCOPY, LAPACK DSPTRS, FETREX et BLAS DAXPY (si STOCKAGE_GI ='OUI').	• Calcul du résidu projeté initial $\mathbf{g}^0 := \mathbf{Pr}^0 = \left(\mathbf{I} - \mathbf{G}_I\left[\left(\mathbf{G}_I\right)^T\mathbf{G}_I\right]^{-1}\left(\mathbf{G}_I\right)^T\right)\mathbf{r}^0$ (K24IRG='&&FETI.REPROJ.G'/ ZR(IRG)).	Si proc 0 O/N
	FETPRJ	FETMPI	• Distribution de $\mathbf{g}^0$ à tous les proc.	En parallèle O/O MPI_BCAST
	FETSCA		• Mise à l'échelle du résidu projeté	O/O

Routine appelante	Routines appelées niveau 1	Routines appelées niveau 2	Détails	Informations relatives aux processeurs
			initial $\widetilde{\mathbf{g}}^0 = \mathbf{A}\mathbf{g}^0$ (K24IR1/ZR(IR1)).	
	FETPRC	Boucle sur les sous-domaines concernés par le proc. courant, BLAS DAXPY, FETREX, MRMULT, Fin de boucle	• Calcul de résidu projeté préconditionné initial $\overline{\mathbf{g}}^0 := \mathbf{M}^{-1}\widetilde{\mathbf{g}}^0$ (K24IR2='&&FETI.VECNBI.AUX 2'/ ZR(IR2)).	P/P
	FETPRC	FETMPI	• Construction du $\overline{\mathbf{g}}^0$ complet par réduction vers le proc. 0.	En parallèle O/N MPI_REDUCE+ MPI_SUM
	FETSCA		• Mise à l'échelle du résidu projeté préconditionné initial $\widetilde{\mathbf{h}}^0 = \mathbf{A}\overline{\mathbf{g}}^0$ (K24IR3='&&FETI.VECNBI.AUX3'/ ZR(IR3)).	Si proc. 0 O/N
	FETPRJ OPTION= 1	BLAS DGEMV/ DCOPY, LAPACK DSPTRS, FETREX et BLAS DAXPY (si STOCKAGE_GI ='OUI').	• Calcul de $\mathbf{h}^0 := \mathbf{P}\widetilde{\mathbf{h}}^0$ (K24IRH='&&FETI.REPCPJ.H'/ ZR(IRH)).	Si proc. 0 O/N
	FETPRJ	FETMPI	• Distribution de $\mathbf{h}^0$ à tous les proc.	En parallèle O/O MPI_BCAST
	BLAS DCOPY		• La variable $\mathbf{p}^0$ reçoit $\mathbf{h}^0$ (K24IRP='&&FETI.DD.P'/ZR(IRP))	O/O
	BLAS DDOT BLAS DNRM2		• Calcul du numérateur du paramètre de descente $\alpha_N^0 := \mathbf{g}^0.\mathbf{p}^0$ (ALPHAN), • Calcul de la norme initiale du résidu projeté $\left\|\mathbf{g}^0\right\|$ (ANORM) et du critère d'arrêt $\varepsilon_K := \text{RESI\_RELA}\left\|\mathbf{g}^0\right\|$ (EPSIK).	Si proc 0 O/N Si proc 0 O/N
	FETMPI		• Distribution de ANORM à tous les procs et calcul de EPSIK.	En parallèle O/O MPI_BCAST

Routine appelante	Routines appelées niveau 1	Routines appelées niveau 2	Détails	Informations relatives aux processeurs
			• Préparation de l'objet JEVEUX CRITER.	O/O
	FETRIN OPTION=2	FETPRJ OPTION=2 FETPRJ FETMPI Boucle sur les sous-domaines concernés par le proc. courant, BLAS DAXPY, FETREX, RLTFR8, Sous-boucles sur les LIGRELs physiques et tardifs du CHAM_NO local Fin des boucles FETMPI	Test d'arrêt si résidu $\|\mathbf{g}^0\|$ quasi-nul (c'est-à-dire inférieur à R8MIEM()**(2.D0/3.D0)) : • Calcul de $\gamma^{sol} := [\mathbf{G}_I^T \mathbf{G}_I]^{-1} \mathbf{G}_I^T \mathbf{r}^0$ (K24ALP='&&FETI.ALPHA.MCR'), • Distribution de γ^0 à tous les procs (variables K24ALP/ZR(IALPHA)). • Calcul de la solution locale $\mathbf{u}_i^{sol} := (\mathbf{K}_i)^+ (\mathbf{f}_i - \mathbf{R}_i^T \boldsymbol{\lambda}^0) - \gamma^{sol} \mathbf{B}_i$ (K24IRR/ZR(IRR)). • Reconstruction du CHAM_NO $\mathbf{u}_i^{sol}$ solution esclave propre au sous-domaine (CHAMLS/ZR(IVALCS)), • Reconstruction du CHAM_NO $\mathbf{u}^{sol}$ solution maître associé au proc. Pour les nœuds physique, on rajoute leur contribution préalablement divisée par la multiplicité géométrique dudit nœud (K24VAL/ZR(IVALS)). • Construction du $\mathbf{u}^{sol}$ complet par réduction vers le proc. 0.	Si proc. 0 O/N En parallèle O/O MPI_BCAST P/P P/P P/P En parallèle O/N MPI_REDUCE+ MPI_SUM
	FETARP	FETPRJ FETFIV ARPACK DNAUPD/DNEUPD BLAS DCOPY	• Test de la définie-positivité de l'opérateur d'interface $\mathbf{PF}_I\mathbf{P}$ si INFO_FETI(7:7)='T' ([U4.50.01 §3.5])	En séquentiel
			• Allocation des gros objets liés à la réorthogonalisation : K24PSR='&&FETI.PS.REORTHO.R', K24DDR='&&FETI.DD.REORTHO.R', K24FIR='&&FETI.FIDD.REORTHO.R' .	Si proc 0 O/N
	Boucle sur les		Algorithme FETI niveau 2 voir tableau [Tableau 2-3].	

Routine appelante	Routines appelées niveau 1	Routines appelées niveau 2	Détails	Informations relatives aux processeurs
	itérations du GCPPC			

Tableau 2-2 : Organigramme détaillé et arbre d'appel dans ALFETI (niveau 1)

Routine appelante	Routines appelées niveau 1	Routines appelées niveau 2	Détails	Informations relatives aux processeurs
ALFETI	BLAS DCOPY		• Si réorthogonalisation, stockage de la direction de descente $\mathbf{p}^k$ dans K24DDR.	Si proc 0 O/N
Boucle sur les itérations du GCPPC	FETFIV	Boucle sur les sous-domaines concernés par le proc. courant, BLAS DAXPY, FETREX, RLTFR8, Fin de boucle	• Calcul du résultat de l'opérateur FETI d'interface sur la direction de descente $\mathbf{z}^k := \sum \mathbf{R}_i (\mathbf{K}_i)^+ \mathbf{R}_i^T \mathbf{p}^k$ (K24IRZ='&&FETI.FIDD.Z'/ZR (IRZ))	P/P
	FETFIV	FETMPI	• Construction du $\mathbf{z}^k$ complet par réduction vers le proc. 0.	En parallèle O/N MPI_REDUCE+ MPI_SUM
	BLAS DCOPY		• Si réorthogonalisation, stockage de $\mathbf{z}^k$ dans K24FIR.	Si proc 0 O/N
	BLAS DDOT		• Calcul du dénominateur du paramètre de descente courant $\alpha_D^k := \mathbf{z}^k.\mathbf{p}^k$ (ALPHAD), • Calcul du paramètre de descente courant $\alpha^k := \frac{\alpha_N^k}{\alpha_D^k}$ (ALPHA).	Si proc 0 O/N Idem
			• Si réorthogonalisation, stockage de ALPHAD dans K24PSR.	Si proc 0 O/N
	FETTOR	FETPRJ BLAS DDOT, DCOPY	• Test des orthogonalités du GCPPC si INFO_FETI(8:8)='T' ([U4.50.01 §3.5]).	Si proc 0 O/N
	BLAS DAXPY FETPRJ OPTION= 1 DAXPY		• Mise à jour du vecteur Lagrange d'interface courant $\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \alpha^k \mathbf{p}^k$ (K24LAI/ZR(IVLAGI)), • Calcul du projeté intermédiaire $\mathbf{r}_1^k = \mathbf{P}\mathbf{z}^k$ (K24IR1='&&FETI.VECNBI.AU X1'/ ZR(IR1)), • Mise à jour du vecteur résidu projeté $\mathbf{g}^{k+1} = \mathbf{g}^k - \alpha^k \mathbf{r}_1^k$ (ZR(IRG)).	Si proc 0 O/N Si proc 0 O/N Si proc 0 O/N
	FETMPI		• Distribution de $\mathbf{g}^{k+1}$ à tous les proc.	En parallèle O/O MPI_BCAST
	BLAS		• Calcul de la norme du résidu projeté	O/O

Routine appelante	Routines appelées niveau 1	Routines appelées niveau 2	Détails	Informations relatives aux processeurs
	DNRM2		$\left\|\mathbf{g}^{k+1}\right\|$ (ANORM).	
	FETRIN OPTION=1 FETRIN OPTION=1 FETRIN OPTION=2	Boucle sur les sous-domaines concernés par le proc. courant, BLAS DAXPY, FETREX, RLTFR8, Fin de boucle FETMPI FETPRJ OPTION=2	Test d'arrêt si $\left\|\mathbf{g}^{k+1}\right\| < \varepsilon_K$: • Recalcul du résidu avec le vecteur d'interface solution $\mathbf{r}^{sol} := \sum \mathbf{R}_i (\mathbf{K}_i)^+ (\mathbf{f}_i - \mathbf{R}_i^T \boldsymbol{\lambda}^{k+1})$ (K24IRR/ZR(IRR)), • Construction du $\mathbf{r}^{sol}$ complet par réduction vers le proc. 0, Calcul de $\boldsymbol{\gamma}^{sol}$ et reconstruction des CHAM_NO solutions maître et esclaves comme dans le test d'arrêt du tableau [Tableau 2-2].	O/O P/P En parallèle O/N MPI_REDUCE+ MPI_SUM Cf. test d'arrêt du 2.b.
	FETSCA		• Mise à l'échelle du résidu projeté courant $\tilde{\mathbf{g}}^{k+1} = \mathbf{A}\mathbf{g}^{k+1}$ (K24IR1/ZR(IR1)).	O/O
	FETPRC	Boucle sur les sous-domaines concernés par le proc. courant, BLAS DAXPY, FETREX, MRMULT, Fin de boucle	• Calcul de résidu projeté préconditionné courant $\bar{\mathbf{g}}^{k+1} := \mathbf{M}^{-1}\tilde{\mathbf{g}}^{k+1}$ (K24IR2='&&FETI.VECNBI.AUX2'/ ZR(IR2)).	P/P
	FETPRC	FETMPI	• Construction du $\bar{\mathbf{g}}^{k+1}$ complet par réduction vers le proc. 0.	En parallèle O/N MPI_REDUCE+ MPI_SUM
	FETSCA		• Mise à l'échelle du résidu projeté préconditionné courant $\tilde{\mathbf{h}}^{k+1} = \mathbf{A}\bar{\mathbf{g}}^{k+1}$ (K24IR3='&&FETI.VECNBI.AUX3'/ ZR(IR3)),.	Si proc. 0 O/N
	FETPRJ OPTION=1		• Calcul du projeté courant $\mathbf{h}^{k+1} = \mathbf{P}\tilde{\mathbf{h}}^{k+1}$ (K24IRH	Si proc. 0 O/N

Routine appelante	Routines appelées niveau 1	Routines appelées niveau 2	Détails	Informations relatives aux processeurs
			/ZR(IRH)).	
	FETREO	BLAS DDOT, DAXPY, DCOPY.	• Mise à jour de la direction de descente courante $\mathbf{p}^{k+1}$ (ZR(IRP)) en réorthogonalisant, ou pas, par rapport aux précédentes directions, • Calcul du numérateur du paramètre de descente courant $\alpha_N^{k+1} := \mathbf{g}^{k+1}.\mathbf{p}^{k+1}$ (ALPHAN).	Si proc. 0 O/N Si proc. 0 O/N
	FETREO	FETMPI	• Distribution de ZR(IRP) à tous les proc.	En parallèle O/O MPI_BCAST
Fin boucle du GCPPC			Nettoyage objets JEVEUX suivant option et numéro de proc.	

Tableau 2-3 : Organigramme détaillé et arbre d'appel dans ALFETI (niveau 2)

3 Mise en place du parallélisme

Tout d'abord, l'algorithme FETI a été codé en séquentiel puis cette implantation a été adaptée pour supporter un parallélisme par envoi de message en MPI-1. En effet, la priorité était de mesurer l'impact d'un tel solveur multidomaine sur l'architecture et les SD du code, d'en limiter les conséquences (lisibilité, efficacité, maintenabilité) et de s'assurer de son bon fonctionnement sur des cas standards. D'ailleurs, pour de nombreux auteurs, un tel solveur se révèle souvent très efficace (en CPU et en occupation mémoire), même en séquentiel, lorsqu'on monte en DDL (cf. [bib2]).
Ensuite, la stratégie de parallélisation a été la suivante :

- Jusqu'à l'opérateur principal appelant le solveur FETI (MECA_STATIQUE…), tous les processeurs exécute la même séquence d'opérations et connaissent donc les mêmes objets JEVEUX : maillage, matériaux, champs issus de prétraitements, SD FETI… C'est relativement sous-optimal, mais compte-tenu de l'architecture du code et de son utilisation courante, c'est la seule option envisageable. Elle a cependant le mérite de ne pas impacter le code séquentiel et, lorsque les prétraitements sont comparativement au solveur, peu gourmands en CPU et en mémoire, c'est aussi souvent la stratégie retenue par les développeurs de codes parallèles.
- Une fois dans l'opérateur principal, on va orienter les opérations effectuées concurremment par les processeurs en tarissant le volume de données qui leur est affecté. Et ce, de la préparation des données solveur, aux factorisations symboliques et numériques, en passant par les phases d'assemblages (de la matrice et des contributions aux seconds membres) et bien sûr l'algorithme de résolution proprement dit. Cela va se faire très simplement, sans envoi de message particulier, *via* l'objet '&&FETI.LISTE.SD.MPI' qui va filtrer les boucles sur les sous-domaines :

```
CALL JEVEUO('FETI.LISTE.SD.MPI','L',ILIMPI)
DO 50 I=1,NBSD                                  <boucle sur les sous-domaines>
   IF (ZI(ILIMPI+I).EQ.1) THEN
      ….                                        <on effectue la suite d'instructions prévues que si le
                                                sous-domaine est contenu dans le périmètre du proc. courant>
   ENDIF
50 CONTINUE
```

Concernant les gros objets JEVEUX usuels, chaque proc. ne va donc construire que les données dont il a besoin : SD SOLVEUR maître et celles esclaves dépendant du périmètre du processeur courant et la même chose pour les NUME_DDL, les MATR_ASSE, les CHAM_NO. Par contre, les données de petit volume, sont elles calculées par tous les proc. car elles permettent souvent d'orienter le calcul et il est bien sûr important que tous les procs. fassent le même cheminement logiciel.

Par contre, le NUME_DDL de chaque sous-domaine n'étant connu que de son processeur d'accueil, les envois de message se font avec des vecteurs de tailles homogène : le nombre de DDL d'interface au cours de l'algorithme ou celui du nombre de DDL total lors de la phase finale de reconstruction.

Le processeur maître gère les étapes de réorthogonalisation et de projection et leurs (potentiellement) gros objets JEVEUX associés.

Le coût de communication est *grosso modo* :

Initialisation : 3 MPI_REDUCE (taille nbi(nbi est le nombre de Lagrange d'interface, c'est-à-dire la taille du problème FETI à résoudre)) + 4 MPI_BCAST (taille nbi) + MPI_GATHERV
A chaque itération du GCPPC : 2 MPI_REDUCE (taille nbi) + 2 MPI_BCAST (taille nbi)
Reconstruction finale : 2 MPI_REDUCE (taille nbi) + 2 MPI_BCAST (taille nbi) + MPI_REDUCE (taille nbddl (nbddl est le nombre total d'inconnus (physiques et tardives) du problème))

Plutôt que des boucles de communications points à points entre les processeurs esclaves et le maître (MPI_SEND/RECV), on a retenu dans une première approche des communications collectives (MPI_REDUCE…) qui encapsulent les premières et gèrent de manière transparentes les problèmes de synchronisation et de buffering. Cela assure une meilleure lisibilité, maintenabilité et portabilité mais, *a contrario*, on ne peut les optimiser en chevauchant les calculs et les communications, en limitant les temps de latence ou le buffering.
Cependant, au vu de l'architecture logicielle actuelle, il semble que ces optimisations ne sont pas si prometteuses que cela, car elles sont très dépendantes de la configuration machine, de celle du réseau, de la carte réseau, de l'implémentation MPI et type de problème. Des gains seraient sans doute plutôt à rechercher du coté d'une implémentation purement parallèle de l'algorithme (sans la vision proc. maître/proc. esclaves) où les échanges de messages se limiteraient aux voisins des sous-domaines et sur des flots de données plus réduits.

4 Bibliographie

[1] O. BOITEAU : Décomposition de domaine et parallélisme en mécanique des structures : Etat de l'art et benchmark pour une implantation raisonnée dans *Code_Aster*. Note interne EDF-R&D HI-23/03/009 (2003).

[2] O. BOITEAU : Rapport d'activité 2004 pour l'UMR EDF-CNRS-LaMSID. Compte rendu interne EDF-R&D CR-I23/2005/006 (2005).