

Manuel d'Utilisation
Fascicule U2.01 : Notions générales
Document : U2.01.02

Notice d'utilisation des conditions aux limites traitées par élimination

Résumé

Le traitement des conditions aux limites du type Dirichlet valeur de ddl imposée par élimination n'offre pas la même souplesse que par dualisation. Ce traitement est à utiliser lorsque l'on recherche à améliorer les temps d'exécution d'un calcul (bien qu'à l'heure actuelle, le gain ne soit pas garanti) ou si l'on souhaite utiliser des matrices de rigidité définies positives.

Notons que tous les types de conditions aux limites disponibles dans `AFPE_CHAR_*` (* = `meca/ther/acou`) ne peuvent être éliminés et traités par `AFPE_CHAR_CINE`.

1 Principe d'élimination utilisé

On cherche à résoudre dans \mathbb{R}^n le problème de minimisation sous contrainte (Pb1) suivant :

$$\begin{aligned} \text{Min} \quad & \left\{ \frac{1}{2} u^T K u - u^T f \right\} \text{ avec : } U_G = \{ u \in \mathbb{R}^n, u|_G = u_0 \} \\ & u \in U_G \end{aligned}$$

où

- $u_0 \in \mathbb{R}^p$ est connu ($1 \leq p \leq n$)
- G est le sous ensemble de $N = \{1, \dots, n\}$, de cardinal $p : G = g_1 \dots g_p$
- $u|_G$ est la projection de u sur le sous espace engendré par $\{u_i\}_{i \in G}$
- où $(u_i)_j = \delta_{ij} \forall j \in N$
- K est une matrice symétrique $n \times n$,
- $f \in \mathbb{R}^n$ est fixé.

La contrainte $u|_G = u_0$ représente des conditions aux limites de type Dirichlet homogène ou non.

Si on note $L = C_N^G$, le complémentaire de G dans N , on peut, à l'aide des u_i définis précédemment, décomposer \mathbb{R}^n en somme directe de $V_G =$ espace vectoriel engendré par $\{u_i\}_{i \in G}$ et de $V_L =$ espace vectoriel engendré par $\{u_i\}_{i \in L}$;

Dès lors, nous avons $\mathbb{R}^n = V_G \oplus V_L$
et l'on note $u = u_G \oplus u_L$ où $u_G = u|_G$ et $u_L = u|_L$

soit encore en notation vectorielle $u = \begin{pmatrix} u_G \\ u_L \end{pmatrix}$

Le problème (Pb1) peut donc s'écrire sous la forme du problème (Pb2) :

$$\left\{ \begin{array}{l} \text{Min} \quad \left\{ \frac{1}{2} u_G^T K_{GG} u_G + \frac{1}{2} u_L^T K_{LL} u_L + u_L^T K_{LG} u_G - u_L^T f_L - u_G^T f_G \right\} \\ u_G \in V_G \\ u_L \in V_L \\ u_G = u_0 \end{array} \right.$$

Ce qui revient à écrire (en supposant que K_{GG} est définie positive)

$$(Pb1) \Leftrightarrow (Pb3) \left\{ \begin{array}{l} \text{Min} \quad \frac{1}{2} u_L^T K_{LL} u_L + u_L^T K_{LG} u_0 - u_L^T f_L \\ u_L \in V_L \\ u = u_0 \oplus u_L \end{array} \right.$$

On a alors éliminé u_G du problème de minimisation.

Nous allons maintenant rechercher le problème matriciel associé à (Pb3).

On recherche u_L minimisant

$$\frac{1}{2} u_L^T K_{LL} u_L + u_L^T K_{LG} u_0 - u_L^T f_L$$

ce qui revient à résoudre le problème matriciel suivant :

$$K_{LL} u_L = F_L - K_{LG} u_0$$

On peut donc écrire :

$$(Pb1) \Leftrightarrow (Pb2) \Leftrightarrow (Pb3) \Leftrightarrow \begin{bmatrix} K_{LL} & 0 \\ 0 & I_G \end{bmatrix} \begin{bmatrix} u_L \\ u_G \end{bmatrix} = \begin{bmatrix} f_L & - K_{LG} u_0 \\ u_0 \end{bmatrix}, \text{ soit } K' \begin{bmatrix} u_L \\ u_G \end{bmatrix} = f'$$

2 Traitement dans Aster

2.1 Les charges cinématiques

Une charge cinématique (type *Aster* : `char_cine_*` [* = `meca/ther/acou`]) permet de caractériser l'ensemble G des ddl imposés et les $(u_0)_i$ pour $i \in G$ qui sont les valeurs affectées à ces ddl.

La définition d'une charge cinématique se fait par l'intermédiaire de l'opérateur `AFFE_CHAR_CINE` pour les $(u_0)_i$ constants ou fonctions de la géométrie ou du temps.

2.2 Les vecteurs cinématiques

Les vecteurs cinématiques sont des `cham_no_*` qui représentent les vecteurs $\begin{bmatrix} 0 \\ u_0 \end{bmatrix}$ stockés dans l'ordre de la numérotation sous jacente au problème.

A chaque charge cinématique correspond un vecteur cinématique.

Cette opération est effectuée par l'opérateur `CALC_CHAR_CINE`.

2.3 Calcul de K'

K' est directement calculée au moment de l'assemblage par l'opérateur `ASSE_MATRICE` sous réserve naturellement que l'on fournisse en argument une liste de charges cinématiques.

La structure de données `MATR_ASSE_*` a été modifiée de façon à pouvoir stocker K' quand cela est nécessaire.

2.4 Calcul de f'

L'opérateur `FACT_LDLT` n'étant pas modifié, le concept de type `matr_asse_*` produit, contient la factorisée de K' et la matrice K_{LG} inchangée.

Le calcul de f' s'effectue au moment de la résolution : il faut fournir à l'opérateur `RESO_LDLT` en

argument le vecteur cinématique correspondant à $\begin{bmatrix} 0 \\ u_0 \end{bmatrix}$ par l'intermédiaire du mot clé `CHAM_CINE`.

Cet opérateur calcule alors f' avant de résoudre $\text{fact}(K') \begin{bmatrix} u_L \\ u_G \end{bmatrix} = f'$.

3 Exemples de fichiers de commandes

3.1 En utilisant FACT_LDLT et RESO_LDLT :

```
DEBUT( CODE:( NOM:'SSLV101AC' )      CALCUL: 'GROS' );

MA =LIRE_MALLAGE ( ) ;

MO =AFFE_MODELE ( MAILLAGE: MA ,
                  AFFE: ( TOUT: 'OUI' , PHENOMENE: 'MECANIQUE' ,
                        MODELISATION: '3D' ) ) ;

MAT =DEFI_MATERIAU ( ELAS: ( E: 202702.7 , NU: 0.3 )
                    THER: ( LAMBDA: 2.7          CP: 0.3 ) );

CHMAT =AFFE_MATERIAU ( MAILLAGE: MA ,
                      AFFE: ( TOUT: 'OUI' , MATER: MAT ) );

%
=====
CHCINE = AFFE_CHAR_CINE( MODELE: MO
                        MECA_IMPO:(GROUP_NO: GRNO7, DX: 0.0, DY: 0.01)
                                (GROUP_NO: GRNO1, DZ: 0.0)
                                (GROUP_NO: GRNO8, DY: 0.0) );

%
=====
MEL =CALC_MATR_ELEM ( MODELE: MO CHAM_MATER: CHMAT OPTION: 'RIGI_MECA' );

NU = NUME_DDL ( MATR_RIGI: MEL ) ;

MATASS = ASSE_MATRICE( MATR_ELEM: MEL , NUME_DDL: NU
                      CHAR_CINE : CHCINE ) ;

VECASS = AFFE_CHAM_NO ( MAILLAGE: MA GRANDEUR : 'DEPL_R'
                       NUME_DDL : NU
                       AFFE      : ( TOUT: 'OUI' ,
                                     NOM_CMP :('DX','DY','DZ')
                                     VALE_R:( 0. , 0. , 0. ) ) ) ;

%
=====
VCINE = CALC_CHAR_CINE ( NUME_DDL: NU CHAR_CINE : CHCINE );

%
=====
&MATASS = FACT_LDLT ( MATR_ASSE: MATASS );

&VECASS = RESO_LDLT ( MATR_FACT: MATASS CHAM_NO: VECASS CHAM_CINE: VCINE );

IMPR_RESU ( MODELE: MO
            RESU: ( FICHIER:'RESULTAT'
                  CHAM_GD : VECASS ) ) ;

FIN();
```

3.2 En utilisant MECA_STATIQUE

```
DEBUT( code : (NOM:'SSLX100A') );

ma = LIRE_MALLAGE();

mo = AFFE_MODELE ( Maillage:ma,
                   Affe:( Maille:HE1,
                          Phenomene:'MECANIQUE', Modelisation:'3D' )
                   ( Group_ma:GRMA1,
                     Phenomene:'MECANIQUE', Modelisation:'DKT' )
                   ( Group_ma:GRMA2,
                     Phenomene:'MECANIQUE', Modelisation:'POU_D_E' ) );

mat = DEFI_MATERIAU ( Elas:(E:2.E5 Nu:0.3 Alpha:0.));

chma = AFFE_MATERIAU ( Maillage:ma Affe:(Tout:'OUI' Mater:mat));

cara = AFFE_CARA_ELEM ( Modele:mo,
                        Coque:( Group_ma:GRMA1,Epais:1.)
                        Poutre:( Group_ma:GRMA2,
                                Section:'RECTANGLE',
                                CARA : ('HZ', 'HY'),VALE:(3.,1.)) );

chci = AFFE_CHAR_cine ( Modele:mo,
                        meca_imo:(Group_no:GRNO1 Dx:0. Dy:0. Dz:0.)
                                (Noeud:(N10,N11,N26,N23) Dz:0.) );

chme = AFFE_CHAR_MECA ( Modele:mo,
                        Liaison_ddl:
                        (Noeud:(N4,N21) Ddl:('DX','DX')
                        Coef_mult:(1.,-1.) Coef_imo:0.)
                        (Noeud:(N4,N21) Ddl:('DY','DY')
                        Coef_mult:(1.,-1.) Coef_imo:0.)
                        (Noeud:(N4,N21) Ddl:('DZ','DZ')
                        Coef_mult:(1.,-1.) Coef_imo:0.)
                        (Noeud:(N16,N25) Ddl:('DX','DX')
                        Coef_mult:(1.,-1.) Coef_imo:0.)
                        (Noeud:(N16,N25) Ddl:('DY','DY')
                        Coef_mult:(1.,-1.) Coef_imo:0.)
                        (Noeud:(N16,N25),Ddl:('DZ','DZ')
                        Coef_mult:(1.,-1.) Coef_imo:0.)
                        (Noeud:(N5,N21) Ddl:('DX','DRZ')
                        Coef_mult:(1.,0.5) Coef_imo:0.)
                        (Noeud:(N17,N25) Ddl:('DX','DRZ')
                        Coef_mult:(1.,0.5) Coef_imo:0.)
                        (Noeud:(N11,N26) Ddl:('DX','DRZ')
                        Coef_mult:(1.,0.5) Coef_imo:0.)
                        (Noeud:(N3,N21) Ddl:('DX','DRZ')
                        Coef_mult:(1.,-0.5) Coef_imo:0.)
                        (Noeud:(N15,N25) Ddl:('DX','DRZ')
                        Coef_mult:(1.,-0.5) Coef_imo:0.)
                        (Noeud:(N10,N26) Ddl:('DX','DRZ')
                        Coef_mult:(1.,-0.5) Coef_imo:0.)
                        (Noeud:(N22,N23) Ddl:('DRZ','DRZ')
                        Coef_mult:(1.,-1.) Coef_imo:0.)
                        (Noeud:(N23,N24) Ddl:('DRZ','DRZ')
                        Coef_mult:(1.,-1.) Coef_imo:0.)
Force_nodale:(Noeud:N29,Fy:-1.)
Impr:0 );
```

```
dep = MECA_STATIQUE ( Modele:mo
                      Cham_mater : chma
                      Cara_elem  : cara
                      EXCIT       : ( Charge : chme )
                      EXCIT       : ( Charge : chci ) ) ;

&dep = CALC_ELEM ( Modele:mo
                  Resultat:dep
                  Cham_mater: chma
                  Option:'EFGE_ELNO_DEPL'
                  Cara_elem: cara
                  Tout_ordre:'OUI' );

&dep = CALC_ELEM ( Modele:mo
                  Resultat:dep
                  Cham_mater: chma
                  Option:'SIGM_ELNO_DEPL'
                  Cara_elem: cara
                  Tout_ordre:'OUI' );

dep1 = RECU_CHAMP ( Resultat: dep  Nom_cham: 'DEPL' NUME_ORDRE:1 );

IMPR_RESU ( Modele:mo Resu:(Resultat:dep));

FIN();
```