

Manuel d'Utilisation
Fascicule U4.3- : Fonction
Document : U4.36.06

Opérateur *GENE_MATR_ALEA*

1 But

Générer des réalisations de matrices généralisées considérées comme aléatoires pour des structures ou des sous-structures. La loi de probabilité des matrices est construite selon le principe du maximum d'entropie en considérant l'information disponible (moyenne et coefficient de variation) et leurs propriétés algébriques (symétrie définie positivité) [R4.03.05].

Produit une structure de données `matr_asse_gene_R` ou `macr_elem_dyna` selon le type de données d'entrée.

2 Syntaxe

```
[macr_elem_dyna]      = GENE_MATR_ALEA

( ♦ / ♦ MATR_MOYEN    = moyenne                [matr_asse_gene_R]
    ♦ COEF_VAR        = / δ                    [R]
                                / 0.1          [DEFAULT]

    / ♦ MATR_MOYEN    = moyenne                [macr_elem_dyna]
    ♦ COEF_VAR_RIGI   = / δR                  [R]
                                / 0.1          [DEFAULT]

    ♦ COEF_VAR_MASS   = / δM                  [R]
                                / 0.           [DEFAULT]

    ♦ COEF_VAR_AMOR   = / δC                  [R]
                                / 0.           [DEFAULT]

    ♦ INIT_ALEA       = ni                      [I]

);

Si moyenne = [matr_asse_gene_R]      alors fr = [matr_asse_gene_R]
Si moyenne = [macr_elem_dyna]        alors fr = [macr_elem_dyna]
```

3 Opérandes

Avec ou sans sous-structuration, cet opérateur consiste in fine à générer des réalisations d'une ou plusieurs matrices aléatoires notées de façon générique $[A]$. $[A]$ est un variable aléatoire à valeur dans l'ensemble des matrices réelles définies positives de dimension (n, n) dont la loi est paramétrée par sa valeur moyenne $[A]$ et son coefficient de dispersion [R4.03.05].

3.1 Mot clé MATR_MOYEN

♦ MATR_MOYEN = moyenne

moyenne désigne la matrice moyenne $[A]$ de la matrice aléatoire $[A]$.

Si moyenne est de type [matr_asse_gene_R], alors $[A]$ est obtenue par projection d'une matrice assemblée moyenne du modèle aux éléments finis moyen sur un nombre donné de modes propres du système dynamique (opérateur MACRO_PROJ_BASE par exemple). $[A]$ et les réalisations de $[A]$ générées par GENE_MATR_ALEA peuvent ainsi être des matrices de masses, raideur ou amortissement généralisées.

Attention :

La matrice moyenne ($[A]$) doit être stockée en mode de stockage plein (opérateur NUME_DDL_GENE, mot-clé STOCKAGE='PLEIN' ou opérateur MACRO_PROJ_BASE, mot-clé PROFIL='PLEIN').

Si moyenne est de type [macr_elem_dyna] (sous-structuration), alors $[A]$ est un concept contenant les matrices de rigidité, de masse et éventuellement d'amortissement projetées sur la base modale de la sous-structure complétées par les matrices de liaison des interfaces, du modèle moyen.

3.2 Mot clé COEF_VAR

♦ COEF_VAR = $\frac{\delta}{0.1}$ [DEFAULT]

Ce mot clé renseigne le paramètre δ de contrôle de la dispersion de la matrice généralisée aléatoire $[A]$ qui peut être de masse, de raideur ou de dissipation. Ce coefficient de variation δ est défini par :

$$\delta = \sqrt{\frac{(n+1) * \|A\|_F^2}{\text{tr}(A)^2 + \text{tr}(A^2)}} * \sqrt{\frac{E\{\| [A] - [A]\|_F^2\}}{\| [A]\|_F^2}}^{1/2}$$

avec :

- $\| [A]\|_F = \left(\text{tr}\{ [A][A]^T \} \right)^{1/2}$
- n la dimension de $[A]$
- $\sqrt{\frac{E\{\| [A] - [A]\|_F^2\}}{\| [A]\|_F^2}}^{1/2}$ le coefficient de dispersion de la matrice $[A]$

δ peut aussi être écrit :

$$\delta = \left\{ \frac{E\left\{\|G_A\|_F^2\right\}}{\|G_A\|_F^2} \right\}$$

avec $[L_A]$ la matrice triangulaire inférieure issue de la factorisation de Cholesky $[A] = [L_A]^T [G_A] [L_A]$ de la matrice moyenne $E\{[A]\} = [A]$.

On doit avoir (cf. [R4.03.05]) :

$$0 < \delta_A < \sqrt{\frac{n_0 + 1}{n_0 + 5}},$$

où $n_0 \in \mathbb{N}$ est une constante du modèle probabiliste choisie de sorte que $n_0 < n$.

◇ COEF_VAR_RIGI = / δ_R [R]
/ 0.1 [DEFAULT]

Ce mot clé renseigne le paramètre δ_R de contrôle de la dispersion de la matrice aléatoire de rigidité d'une sous-structure. Ce coefficient de variation est défini de manière identique à la définition donnée pour le mot clé COEF_VAR.

◇ COEF_VAR_MASS = / δ_R [R]
/ 0. [DEFAULT]

Ce mot clé renseigne le paramètre δ_R de contrôle de la dispersion de la matrice aléatoire de masse d'une sous-structure. Ce coefficient de variation est défini de manière identique à la définition donnée pour le mot clé COEF_VAR.

◇ COEF_VAR_AMOR = / δ_R [R]
/ 0. [DEFAULT]

Ce mot clé renseigne le paramètre δ_R de contrôle de la dispersion de la matrice aléatoire de dissipation d'une sous-structure. Ce coefficient de variation est défini de manière identique à la définition donnée pour le mot clé COEF_VAR.

3.3 Opérande INIT_ALEA

◇ INIT_ALEA = ni [I]

Provoque l'initialisation à son ni-ième terme de la suite de nombres pseudo-aléatoires utilisés pour la génération des matrices.

Si le mot-clé INIT_ALEA est absent, les termes utilisés de la suite sont ceux immédiatement consécutifs à ceux déjà utilisés. Si aucun terme n'a encore été utilisé, la suite est initialisée à son premier terme.

Recommandation:

A moins d'un usage particulier, il est conseillé de ne pas renseigner le mot-clé INIT_ALEA dans les opérateurs suivant : GENE_FONC_ALEA, GENE_VARI_ALEA et GENE_MATR_ALEA. Dans ce cas, au premier appel à l'un de ces opérateurs, la suite de nombres pseudo-aléatoires est initialisée à son premier terme. L'omission du mot-clé INIT_ALEA à chacun des appels de ces opérateurs dans le fichier de commande garantit l'indépendance statistique des nombres pseudo-aléatoires utilisés.

Remarque :

Le germe de la suite reste identique d'une exécution à l'autre de Code_Aster ; les résultats restent donc rigoureusement identiques (on peut ainsi tester la non régression de résultats statistiques non convergés). Si l'on souhaite générer des résultats statistiquement indépendants d'une exécution à l'autre, alors il faut utiliser le mot-clé `INIT_ALEA` avec des valeurs majorant le nombre de termes utilisés dans les exécutions antérieures.

Attention :

Le générateur de variable aléatoire utilisé est celui du module "random" de Python. Il dépend de la version de Python exploité par Code_Aster. Des résultats non convergés statistiquement peuvent donc varier d'une version à l'autre de Code_Aster ou d'une plateforme à l'autre, si la version de Python n'est pas la même et qu'entre les deux versions le module random a évolué (cas entre Python 2.1 et 2.3).

Remarque :

En version Python 2.3, la période du générateur est de $2^{19937-1}$ [bib1].

4 Exemple

Par appel, la commande ne génère qu'une seule réalisation de la matrice aléatoire à simuler. Pour générer plusieurs réalisations d'une même matrice aléatoire, il faut répéter la commande sans changer ses paramètres ou bien placer la commande dans une boucle du langage de commande de Code_Aster - le langage python.

Dans l'exemple suivant, on génère ns réalisations d'une matrice aléatoire de valeur moyenne `MATR_MOYEN` avec un $\delta = 0.1$. Ces réalisations sont ensuite utilisées comme valeurs de matrice de masse.

```
ns=100

for k in range(1,ns+1):

# Génération
  MAT_ALEA=GENE_MATR_ALEA(
                                MATR_MOYEN=MATR_MOY,
                                COEF_VAR=0.1,
                                )

  DYN=DYNA_TRAN_MODAL(
                        ...MASS_GENE= MAT_ALEA,
                        )
# Ici par exemple, traitement statistique de DYN

  DETRUIRE(CONCEPT=_F(NOM=(DYN, MAT_ALEA)))

# Fin de la boucle (indentation)
```

Pour des exemples plus complets, consulter les cas test SDNS01 [V5.06.001], SDNL105d [V5.02.105] et SHLS200a [V2.06.200], ainsi que [U2.08.05].

5 Bibliographie

- [1] M. Matsumoto and T. Nishimura, Mersenne Twister : A 623-dimensionally equidistributed uniform pseudorandom number generator, ACM Transactions on Modeling and Computer Simulation Vol. 8, No. 1, January pp.3-30 1998.