# Opérateur GENE_VARI_ALEA

## 1 But

Générer une réalisation d'une variable aléatoire réelle de loi de probabilité donnée (lois gamma ou exponentielle issues de l'application du maximum d’entropie, [R4.03.05]).

En sortie, on obtient un réel.

# 2 Syntaxe

```
[réel]    =  GENE_VARI_ALEA   (

    ◊  / TYPE   =   ' GAMMA'                             [DEFAUT]
           ◊  VALE_MOY  =  /   vale_moy                  [R]
                             / 1.0                       [DEFAUT]
           ◊  BORNE_INF =  /   a                          [R]
                           /   0.                        [DEFAUT]
            ◊  COEF_VAR =  /   delta                      [R]
                           /   0.1                        [DEFAUT]

       /   TYPE   =   'EXPONENTIELLE'
           ◊  VALE_MOY  =  /   vale_moy                  [R]
                           /   0.                        [DEFAUT]
           ◊  BORNE_INF =  /   a                          [R]
                           /   -1.0                      [DEFAUT]

       /   TYPE   =   'EXP_TRONQUEE'
           ◊  VALE_MOY  =  /   vale_moy                  [R]
                           /   0.                        [DEFAUT]
           ◊  BORNE_INF =  /   a                          [R]
                           /   -1.0                      [DEFAUT]
           ◊  BORNE_SUP =  /   b                          [R]
                           /   1.0                       [DEFAUT]
    ◊  INIT_ALEA =  ni                                   [I]
    );
```

# 3 Opérandes

## 3.1 Mot clés TYPE

Suivant l'information utilisable sur la variable aléatoire à simuler, trois types de loi de probabilité sont disponibles. Si l'information disponible est un support non borné $[a,+\infty[$, une moyenne $\underline{w}$, et un coefficient de dispersion $\delta$, la loi est une gamma. Si l'information disponible est un support non borné $[a,+\infty[$ et une moyenne $\underline{w}$, la loi est une exponentielle. Si l'information disponible est un support compact $[a,b]$ et une moyenne $\underline{w}$, la loi est une exponentielle tronquée.

```
/   TYPE   =  ' GAMMA'                                       [DEFAUT]
```

La variable aléatoire suit une loi de probabilité de type « gamma » dont la distribution de probabilité $P_W(dw)$ est définie par :

$$P_W(dw)=I_{[a,+\infty[}(w)\frac{\left(\underline{w}\delta^2-a\delta^2\right)^{-\frac{1}{\delta^2}}}{\Gamma\left(\frac{1}{\delta^2}\right)}(w-a)^{\frac{1-\delta^2}{\delta^2}}\exp\{-\frac{w-a}{(\underline{w}-a)\delta^2}\}dw$$

avec $\Gamma(z)=\int_0^{+\infty}t^{z-1}e^{-t}dt$

```
/   TYPE   =  'EXPONENTIELLE'
```

La variable aléatoire suit une loi de probabilité de type « exponentielle » dont la distribution de probabilité $P_W(dw)$ est définie par :

$$P_W(dw)=I_{[a,+\infty[}(w)\frac{1}{\underline{w}-a}\exp\{-\frac{w-a}{\underline{w}-a}\}dw$$

avec $I_{[a,+\infty[}(w)=1$ si $w\in[a,+\infty[$ et $I_{[a,+\infty[}(w)=0$ si $w\notin[a,+\infty[$

```
/   TYPE   =  'EXP_TRONQUEE'
```

La variable aléatoire suit une loi de probabilité de type « exponentielle tronquée » dont la distribution de probabilité $P_W(dw)$ est définie par :

$$P_W(dw)=I_{[a,b]}(w)\frac{k}{\alpha(k)}e^{-kw}dw$$

avec $I_{[a,b]}(w)=1$ si $w\in[a,b]$ et $I_{[a,b]}(w)=0$ si $w\notin[a,b]$ et où $k$ est tel que $(\underline{w}k-1)\alpha(k)-k\beta(k)=0$ , avec $\alpha(k)=e^{-ak}-e^{bk}$ et $\beta(k)=ae^{-ak}-be^{-bk}$ .

## 3.2 Mot clé VALE_MOY

```
◊     VALE_MOY   =   /   w                           [R]
                     /   0. ou 1.0                   [DEFAUT]
```

Désigne la valeur moyenne de la variable aléatoire à simuler.

## 3.3 Mot clé COEF_VAR

```
◊   COEF_VAR  =  /  δ                                  [R]
                 /  0.1                                [DEFAUT]
```

Ce mot clé renseigne le coefficient de dispersion (rapport écart-type sur valeur absolue de la moyenne). La valeur prise par défaut est 0.1.

## 3.4 Mots clé BORNE_INF ET BORNE_SUP

```
◊   BORNE_INF =  /  a                                  [R]
                 /  -1.0 ou 0.                         [DEFAUT]
◊   BORNE_SUP =  /  b                                  [R]
                 /  +1.0                               [DEFAUT]
```

Ces mots clé renseignent la borne inférieure et la borne supérieure (lorsqu'elles existent) du support $[a,b]$ ou $[a,+\infty[$ des lois.

## 3.5 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 variables.

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.

**Suggestion :**

*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 plate-forme 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.

# 4 Exemple

Par appel, la commande ne génère qu'une seule réalisation de la variable aléatoire à simuler. Pour générer plusieurs réalisations d'une même variable 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. Chaque réalisation est statistiquement indépendante des autres réalisations.

Dans l'exemple suivant, on génère $ns$ réalisations d'une variable aléatoire gamma de valeur moyenne 25000, de coefficient de dispersion 0.1 et de support les réels positifs. Ces réalisations sont ensuite utilisées comme valeurs de raideur de choc.

```
ns=100

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

# Génération
   KN=GENE_VARI_ALEA(  TYPE      =  'GAMMA',
                       BORNE_INF =  0.,
                       VALE_MOY  =  25000.,
                       COEF_VAR  =  0.1,
                       )

   DYN=DYNA_TRAN_MODAL(
                       ...
                       CHOC=_F(
                                 ...
                                 RIGI_NOR  =  KN,
                                 ...
                                 )
                       )

# Ici par exemple, traitement statistique de DYN

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

# Fin de la boucle (indentation)
```

Pour des exemples plus complets, on peut consulter la documentation « Simulation numérique de Monte Carlo » [U2.08.05] ou le cas test SDNS01 [V5.06.001].