Propagation des incertitudes et calcul de courbes de fragilité

Résumé :

Ce document donne les éléments pour la mise en œuvre de simulations numériques de Monte Carlo via la distribution de calculs avec Code_Aster.

Les cinq principaux ingrédients sont :

- la création du plan d'expérience: tirage des variables aléatoires des Ns cas d'études ;
- la mise sous format Code_Aster pour le lancement de Ns études mécaniques (fichier `.distr`) ;
- les calculs Code_Aster ;
- les lectures des résultats de calcul et écriture dans un fichier unique ;
- le post-traitement statistique: calcul d'estimateurs statistiques classiques comme la moyenne, l'écart-type, etc ; l'estimation des paramètres d'une courbes de fragilité log-normale (`POST_DYNA_ALEA`) ; autres post-traitements ...

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

Table des Matières

1 Introduction....3
2 Simulation de Monte Carlo avec Code_Aster....3
2.1 Réalisation du plan d'expérience....3
2.2 Définition et lancement des études distribuées avec Code_Aster....4
2.3 Récupérer les résultats des calculs dans un fichier....4
3 Post-traitements statistiques....5
3.1 Calcul des grandeurs statistiques....5
3.2 Calcul de courbes de fragilité....5
3.2.1 Définitions....5
3.2.2 Évaluation des paramètres par maximum de vraisemblance....6
3.2.3 Évaluation par régression linéaire....7
4 Bibliographie....9

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

1 Introduction

La méthode numérique de Monte Carlo consiste à faire Ns simulations numériques du modèle mécanique afin d'évaluer les grandeurs statistiques des résultats de calcul. Les 3 principales étapes d'une propagation des incertitudes par simulation de Monte Carlo sont :

- La modélisation des incertitudes : Génération d'un échantillon de Ns réalisations des données aléatoires en entrée du modèle mécanique (paramètres aléatoires et processus stochastique),
- La propagation des incertitudes : Calcul des Ns grandeurs résultats correspondant à ces données,
- Le calcul des estimateurs statistiques des grandeurs recherchées : moyenne, écart-type, fractiles … ; évaluation de courbes de fragilité sismique.

Dans le cadre d'une étude mécanique avec Code_Aster, on peut faire appel aux fonctionnalités de distribution d'études paramétriques afin de réaliser une telle analyse probabiliste. En effet, dans le cas courant où les Ns tirages et simulations mécaniques sont indépendants, on peut au préalable établir un plan d'expériences des cas d'étude à lancer. On lance ensuite les Ns études correspondantes, et les grandeurs statistiques sont calculées en post-traitement des résultats.

La démarche et sa mise en œuvre avec *Code_Aster* sont décrites dans ce document.

2 Simulation de Monte Carlo avec *Code_Aster*

Dans le cadre de la simulation de Monte Carlo, la modélisation des incertitudes passe par la génération de variables aléatoires et éventuellement de processus stochastiques. Dans ce qui suit, on suppose que l'utilisateur ait choisi des lois caractérisant bien les m paramètres incertains. Pour ce faire, on peut avoir recours à des résultats d'essais, le jugement d'expert ou encore le principe du maximum d'entropie (voir [R4.03.05]). Très souvent, on peut supposer que les différentes sources d'incertitudes sont indépendantes. On peut alors effectuer des tirages aléatoires selon les m lois marginales des variables aléatoires. Dans certains cas, on peut aussi inclure des grandeurs plus complexes (non scalaires) comme les matrices aléatoires ou encore les processus stochastiques.

On parle aussi de plan d'expérience pour désigner la méthode de tirage et l'échantillon qui en résulte. Dans *Code_Aster,* la simulation de Monte Carlo selon un plan d'expériences est possible à l'aide de l'option calculs distribués [U2.08.07].

2.1 Réalisation du plan d'expérience

Toutes les méthodes dont les tirages sont indépendants peuvent être mises en œuvre via la distribution des calculs avec *Code_Aster*. En particulier, on peut faire appel aux tirages aléatoires selon la méthode de l'hypercube Latin (Latin Hypercube Sampling – LHS) ou le quasi Monte Carlo (suites quasi-aléatoires comme la suite de Sobol, …) [bib4, bib5].

- Variables aléatoires: les incertitudes sur les paramètres du modèle éléments finis (paramètres matériaux, valeurs d'un jeu, d'une raideur de butée élastiques, d'un module d'Young, etc).
- Matrices aléatoires: En dynamique des structures sur base modale, les variables aléatoires peuvent être les valeurs des matrices généralisées de masse, de raideur et d'amortissement du modèle aux éléments finis. Dans cette approche, on fait des tirage de ces matrices aléatoires à l'aide de l'opérateur GENE_MATR_ALEA [U4.36.06]. Dans cette approche, on modélise à la fois les incertitudes de modèle et de modélisation par une approche probabiliste dite « non-paramétrique ».
- Excitations aléatoires: l'opérateur GENE_FONC_ALEA permet de générer des signaux aléatoires stationnaire à partir de la densité spectrale de puissance. L'opérateur GENE_ACCE_SEISME permet de générer des signaux sismiques non stationnaire en amplitude et en contenu fréquentiel. Dans d'autres cas, on peut disposer d'une base de données contenant des signaux

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

qui modélisent un phénomène physique aléatoire. Cela peut être le cas en analyse sismique où un ensemble d'accélérogrammes mesurés in situ peut être fourni par des sismologues.

Il existe deux possibilités pour réaliser le plan d'expérience: 1) soit on crée l'échantillon des jeux de paramètres au préalable selon la méthode de tirage souhaitée et on le stocke dans un fichier de données, pour cela on peut utiliser un logiciel dédié comme OpenTURNS ou encore les toolbox dédiés de Python et Matlab2) soit on utilise l'un des opérateurs du *Code_Aster* pour générer une réalisation d'un processus stochastique ou d'une matrice aléatoire. Dans le cas d'un calcul dynamique transitoire, on peut ainsi utiliser des chargements temporels aléatoires générés par le *Code_Aster*. On peut bien sûr utiliser les deux options conjointement.

Remarque: Si, dans le cadre d'une analyse sismique, on dispose d'un échantillon de Ns accélérogrammes, on peut se ramener, pour le plan d'expérience, à une variable aléatoire discrète prenant des valeurs de 1 à Ns. Chaque entier désigne l'une des réalisations du processus aléatoire (c'est-à-dire un accélérogramme). Le tirage par Hypercube Latin consiste alors à parcourir les entiers de 1 à Ns.

Attention: *Les commandes* `GENE_MATR_ALEA` *et* `GENE_FONC_ALEA` *génèrent la même suite de variables aléatoires à l'intérieur d'une même exécution de Code_Aster. Ainsi, d'une exécution à l'autre, un fichier de commande strictement identique (mêmes appels aux commandes dans le même ordre avec les mêmes arguments) fournira exactement les mêmes résultats. Ceci est dû au fait que le générateur de variables aléatoires utilisé par Code_Aster est toujours initialisé à la même valeur. Si l'on souhaite générer des résultats différents d'une exécution à l'autre, alors il faut utiliser le mot-clé* `INIT_ALEA` *. En revanche,* `GENE_ACCE_SEISME` *génère des signaux sismiques indépendants aléatoires.*

2.2 Définition et lancement des études distribuées avec Code_Aster

Il faut, dans la fenêtre astk :

1	définir l'étude déterministe (fichier `.comm`, maillage, etc.) nominale
2	définir, dans le `.comm`, les valeurs nominales des paramètres (incertains) qui seront modifiés ensuite selon le plan d'expérience
3	définir, toujours dans le `.comm`, la ou les variables d'intérêt et écrire ces variables dans un fichier résultat (sous format python)
4	définir un répertoire output (type `repe`) où seront écrit ces résultats
5	définir le plan d'expérience dans le `.distr` (jeux de paramètres pour les études à lancer)
6	aller dans l'onglet « Option », choisir `distrib = yes` (cf. [U1.04.00] §8)
7	dans l'onglet « Option », choisir `classe = distr` si calculs sur le serveur centralisé
8	si utile, ajouter un fichier de type `hostfile` définissant les machines disponibles (cf. [U1.04.00] §9)

Pour plus de détail, on peut consulter le tutoriel [U2.08.07].

2.3 Récupérer les résultats des calculs dans un fichier

Les résultats des calculs distribués avec *Code_Aster* sont écrites dans des fichiers de sortie qu'on retrouve dans les Ns répertoires nommés $CALC_i$, $i=1,Ns$ (cf. [U2.08.07]). Afin de procéder au post-traitement par un logiciel de statistique (comme OpenTURNS ou encore les toolbox dédiés de Python et Matlab, R), il faut rassembler l'échantillon de sorties dans un fichier de résultats unique. Ceci peut se réaliser de manière très aisée via un script python.

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

3 Post-traitements statistiques

3.1 Calcul des grandeurs statistiques

Dans le cadre d'une approche par simulation de Monte Carlo, on s'intéresse non seulement à un résultat de calcul déterministe, mais on souhaite évaluer moyenne, médiane, écart-type, fractile et d'autres statistiques décrivant la variable de sortie. Ceci permet notamment d'associer des intervalles de confiance aux résultats de calcul.

Après avoir récupéré l'échantillon des sorties, on peut faire appel au logiciel préféré (comme OpenTURNS, les toolbox dédiés de Python et Matlab, R) pour réaliser ces post-traitements statistiques. En outre, des courbes de fragilité log-normales peuvent être évaluées à l'aide de l'opérateur POST_DYNA_ALEA de *Code_Aster*. Ceci est décrit plus en détail dans le paragraphe suivant.

3.2 Calcul de courbes de fragilité

3.2.1 Définitions

Les courbes de fragilité donnent la probabilité conditionnelle de défaillance d'une structure ou d'un composant en fonction du niveau d'excitation sismique. La modélisation et la propagation des incertitudes décrites ci-dessus permettent également la détermination des courbes de fragilité. Il suffit d'introduire un critère de défaillance et de vérifier à chaque calcul de Monte Carlo si la défaillance est atteinte ou non.

Les ingrédients principaux pour l'établissement de courbes de fragilité par simulation numérique sont les suivants :

- Détermination de l'excitation sismique à considérer (base de données d'accélérogrammes),
- Définition de critères de défaillance,
- Modélisation des incertitudes et propagation par simulation numérique (Monte Carlo)
- Estimation des paramètres (médiane et écart-type logarithmique) de la courbe de fragilité log-normale.

La courbe de fragilité d'un composant peut être définie à partir de la notion de "capacité". La capacité d'un composant est la valeur du paramètre représentatif de l'action sismique à partir de laquelle le composant est défaillant. L'approche courante consiste à modéliser la capacité par une variable aléatoire suivant une loi log-normale, telle que $A = A_m \varepsilon$, où A_m est la capacité médiane et ε désigne une variable aléatoire log-normale de médiane unité et d'écart-type logarithmique β . Aussi , la capacité A d'un composant (et A_m ainsi sa courbe de fragilité), est caractérisée par deux paramètres qui sont la médiane (« capacité médiane ») et l'écart-type β .
Ainsi, la probabilité de ruine pour un niveau d'accélération a donné peut s'écrire [bib3, bib2]:

$$P_f(a) = \int_0^a \underbrace{p(x)}_{\text{loi log-normale}} dx = \Phi\left(\frac{\ln(a/A_m)}{\beta}\right)$$

où $\Phi(.)$ désigne la fonction de répartition d'une variable aléatoire gaussienne centrée réduite. L'utilisation du modèle lognormal possède l'avantage de nécessiter un nombre réduit de simulations par rapport à un calcul direct des probabilités de défaillance. En effet, dans un calcul direct, sans hypothèse de loi a priori, il faut déterminer des probabilités d'évènements rares (les probabilités en queue de distribution), ce qui demande un nombre de simulations de Monte Carlo très importants. Dans ce qui suit, nous présentons deux méthodes, l'estimation par maximum de vraisemblance et la régression linéaire, pour évaluer ces deux paramètres.

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

Si l'endommagement ou la défaillance d'une structure sont caractérisés par une variable d'intérêt Y continue, assortie d'un seuil Y_s, on peut exprimer la probabilité de défaillance (du niveau de dommage) pour un séisme de niveau a comme

$$P_f(a)=P(Y>Y_s|a)$$

Des variables d'endommagement Y classiques pour une structure en béton sont le drift (déplacement différentiel entre deux étages d'un bâtiment) ou encore la baisse des premières fréquences propres. Les notations introduites ici seront utilisés pour l'évaluation d'une courbe de fragilité par régression linaire (§3.2.2).

3.2.2 Évaluation des paramètres par maximum de vraisemblance

Dans ce qui suit, on considère que l'accélération maximale a été choisie pour caractériser le niveau d'excitation sismique et donc la capacité. La démarche suivie consiste alors à modéliser le résultat des expériences numériques par une variable aléatoire de Bernoulli X. En effet, pour chaque simulation numérique i, on a deux issues possibles : soit on a atteint le niveau critique et on a défaillance ($x_i=1$) soit on n'a pas défaillance ($x_i=0$). De même pour chaque simulation, on peut déterminer la valeur de l'accélération maximale a_i . L'estimation des paramètres d'une courbe de fragilité peut se faire par la méthode du maximum de vraisemblance. La fonction de vraisemblance à maximiser pour ce problème s'écrit :

$$L=\prod_{i=1}^{N}\left(P_{f|a}(a_i)\right)^{x_i}\left(1-P_{f|a}(a_i)\right)^{1-x_i}$$

Dans cette expression, la réalisation x_i de X prend donc la valeur 1 si on a défaillance ou 0 s'il n'y a pas défaillance pour le chargement (l'accélération maximale) a_i . Ces événements arrivent avec la probabilité $P_{f|a}$ donnée par l'expression (1). Les estimations des paramètres β et A_m sont ceux qui minimisent $-\ln(L)$:

$$(\beta_C^e, A_m^e)=\arg\min_{\beta,\, A_m}\left[-\ln(L)\right] .$$

Cette étape peut être mise en œuvre en post-traitement des résultats de calculs. Dans le fichier de commande *Code_Aster,* il faut, pour chaque simulation, vérifier si on a défaillance ou non. À titre d'exemple, si le critère de défaillance consiste en une contrainte admissible, on vérifie si la contrainte maximale calculée dépasse cette contrainte.

Puis on fait écrire, à laide de CREA_TABLE, les résultats des simulations dans une table de résultats. Cette table doit contenir deux colonnes, PARA_NOCI et DEFA, qui renseignent respectivement la valeur a_i (la valeur PGA ou tout autre indicateur) et la valeur x_i (1 ou 0).

```
TAB1=CREA_TABLE(LISTE=(
                        _F(PARA='PARA_NOCI',LISTE_R = PGA,),
                        _F(PARA='DEFA',LISTE_I =xi ,),
                        ),);
```

Remarque : Si on renseigne une colonne DEMANDE au lieu de DEFA, alors cette colonne doit contenir les valeurs de la demande sismique (variable d'intérêt décrivant la défaillance ou l'endommagement, par exemple une contrainte ou déplacement maximale).

```
TAB2=CREA_TABLE(LISTE=(
                        _F(PARA='PARA_NOCI',LISTE_R = PGA,),
                        _F(PARA='DEMANDE',LISTE_I =demande ,),
                        ),);
```

Le mot-clé SEUIL doit alors être renseigné dans POST_DYNA_ALEA pour déterminer la suite des x_i .

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

L'opérateur POST_DYNA_ALEA [U4.84.04] permet d'estimer les paramètres de la courbe de fragilité lognormale à partir de la table TAB1 via le mot-clé FRAGILITE:

```
TAB_POST=POST_DYNA_ALEA( FRAGILITE=(_F(TABL_RESU=TAB1,
                                       AM_INI  =0.3 ,
                                       BETA_INI=0.1 ,
                                       METHODE= 'EMV',
                                       FRACTILE = (0.0,0.05,0.5,0.95,1.0),
                                       NB_TIRAGE =Ns,
                                     ),),
                         TITRE = 'courbe 1',
                         INFO=2,);
```

On donne des valeurs initiales des paramètres A_m et β à estimer (point de démarrage pour l'algorithme d'optimisation) via AM_INI et BETA_INI. Si l'on renseigne le mot-clé FRACTILE, on détermine également les fractiles (intervalles de confiance) de la courbe de fragilité par une méthode de ré-échantillonnage (dite méthode de « bootstrap »). Le ré-échantillonnage consiste à tirer de nouveaux échantillons à partir des valeurs de l'échantillon original. Ces tirages s'effectuent avec remise, voir [bib1] pour plus de détails. Ensuite, on détermine les paramètres de la courbe de fragilité pour chaque échantillon « bootstrap », ce qui donne un échantillon de courbes de fragilités. On détermine alors les fractiles pour l'échantillon de courbes de fragilités obtenu. En général, on tire autant d'échantillons « bootstrap » qu'on dispose de valeurs dans l'échantillon original. Il est néanmoins possible de travailler avec un nombre de tirages inférieur en renseignant NB_TIRAGE (par défaut le nombre de tirages correspond à la taille de l'échantillon original, ici Ns). Dans l'exemple ci-dessus, on détermine donc les courbes enveloppes (fractiles 1.0 et 0.0) ainsi que la médiane (0.5) et les fractiles à 0.05 et 0.95 .

Dans le graphique ci-dessous, on donne un exemple de courbe de fragilité pour $A_m=0.68$ et $\beta=0.25$:

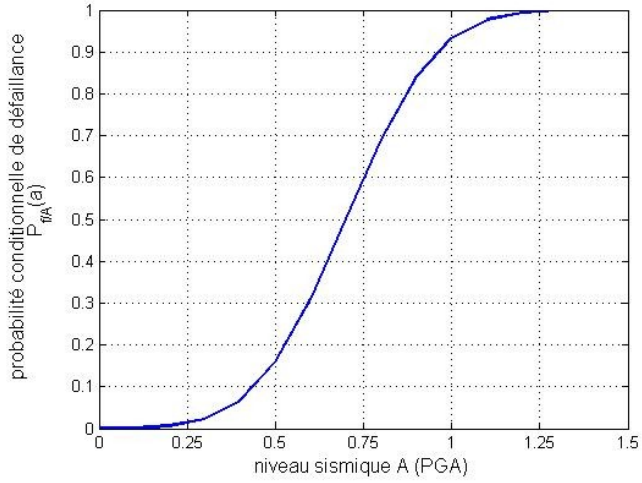


Figure 3.2.2-a : Probabilité conditionnelle de défaillance en fonction du PGA

3.2.3 Évaluation par régression linéaire

La régression est une méthode très répandue pour déterminer les paramètres d'une courbe de fragilité log-normale. Elle est pratiquée surtout aux États-Unis pour évaluer la fragilité des structures civiles comme les routes, les ouvrages d'art et bâtiments publics.

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

Dans cette approche, on relie la variable d'intérêt Y (le drift par exemple), à la valeur de l'indicateur sismique a (par exemple le PGA) via une relation du type :

$$Y = b\, a^c\, \eta$$

Dans l'expression ci-dessus, η est une variable aléatoire log-normale qu'on peut définir à partir de la variable normale centrée réduite U : $\eta = \exp(\beta U)$, et les variables b et c sont les paramètres du modèle. Ces derniers sont obtenus par une régression linéaire à l'aide de l'expression suivante:

$$\ln(Y) = \ln(b) + c \ln(a) + \epsilon$$

où ϵ est une variable aléatoire normale (Gaussienne) centrée d'écart-type σ telle que $\epsilon = \sigma U$. Avec ces notations, la probabilité de défaillance s'écrit comme une probabilité conditionnelle, en fonction du niveau sismique a et pour un seuil Y_s :

$$P_f(a) = P(Y > Y_s | a) = 1 - P(Y < Y_s | a)\ .$$

En tenant compte de l'expression log-normale de la variable Y, on obtient l'expression analytique de la courbe de fragilité:

$$P_f(a) = \Phi\left(\frac{\ln(b\, a^c / Y_s)}{\sigma}\right)$$

A l'aide de la droite de régression, on peut exprimer le seuil médian Y_s à partir de la capacité médiane comme $Y_s = b\, a_s^c\, \eta$. Si on considère un seuil critique « best-estimate » qu'on assimile à la valeur médiane dans la démarche incertitudes, on peut noter $A_m = a_s$. On aboutit ainsi à l'expression de la courbe de fragilité log-normale:

$$P_f(a) = \Phi\left(\frac{\ln(a / A_m)}{\beta}\right)$$

où $\beta = \sigma / c$ avec l'écart-type de l'erreur de régression σ. En pratique, la valeur de la capacité médiane peut être déduite directement de la courbe de régression et s'exprime à partir des paramètres de cette dernière comme

$$A_m = \exp\left[\frac{\ln(Y_s / b)}{c}\right]$$

Il est important de souligner que ce modèle permet de tenir compte du comportement hétéroscédastique (la variance de Y dépend des paramètres du modèle et n'est pas constante).

L'opérateur `POST_DYNA_ALEA` [U4.84.04] permet d'estimer les paramètres de la courbe de fragilité lognormale par régression à partir de la table `TAB2` du §3.2.2 via le mot-clé `FRAGILITE`:

```
TAB_POST=POST_DYNA_ALEA( FRAGILITE=(_F(TABL_RESU=TAB2,
                                       METHODE= 'REGRESSION',
                                       SEUIL=0.05,
                                      ),),
                         TITRE = 'courbe 2',
                         INFO=2,);
```

L'expérience montre que la dispersion (variabilité) de Y augmente avec le niveau sismique. En revanche, au contraire de la méthode de maximum de vraisemblance, l'écart-type logarithmique ne dépend pas du seuil.

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

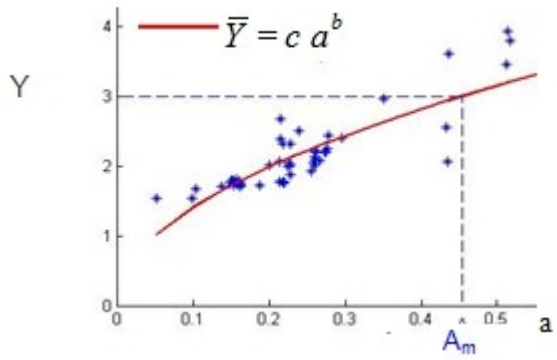


Figure 3.2.3-a : Exemple de nuage de points et courbe de régression: variable Y en fonction du PGA.

4 Bibliographie

[bib1] Saporta G., *Probabilités, analyse de données et statistique.* Editions Technip, 2006.

[bib2] Zentner I., Numerical computation of fragility curves for NPP equipment. *Nuclear Eng. Design* **240:6** (2010) 1614-1621.

[bib3] EPRI, Seismic Probabilistic Risk Assessment Implementation Guide, Final Report 1002989, 2003.

[bib4] Lefebvre Y., de Rocquigny E., Dutfoy A., Delcoigne F., Sudret B., Cagnac A., *Guide Méthodologique pour le traitement des incertitudes*. Note EDF R&D HT-56-2007-01798, 2007.

[bib5] De Rocquigny et al. *Uncertainty in Industrial Practice*. Wiley & Sons Ltd., 2008.

[bib6] Ellingwood B.R., Kinali K. Quantifying and communicating uncertainty in seismic risk assessment. *Structural Safety* **31(2)** , 2009.

Documentation Code_Aster

[R4.05.05] Génération de signaux sismiques

[U1.03.02] Méthodes d'accès aux objets Aster

[U2.08.07] Calculs paramétriques – Distribution de calculs

[U1.04.00] Interface d'accès à Code_Aster: astk

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