

Macro commande MACR_FIABILITE

1 But

Calculer des probabilités de dépassement de seuil, également appelée dans le domaine de la fiabilité « probabilité de défaillance ». Actuellement, en ce qui concerne la partie fiabiliste, la macro-commande utilise le logiciel MEFISTO (Méthode d'Etude Fiabiliste Incluant une Série de Tests d'Optimalité) qui est développé à EDF R&D. Le logiciel MEFISTO utilise le principe des méthodes analytiques FORM (First Order Reliability Method) et SORM (Second Order Reliability Method). La macro-commande retourne un concept de type `listr8` qui est une liste contenant une seule valeur réelle : la probabilité de défaillance.

Les principales étapes de la macro commande sont :

- elle crée les fichiers de données nécessaires à MEFISTO à partir des données fournies par l'utilisateur, puis lance MEFISTO,
- afin de calculer la probabilité de défaillance MEFISTO lance une série de calculs déterministes via *Code_Aster* pour obtenir les gradients des variables définies comme sensibles dans la macro-commande ; si les gradients ne sont pas disponibles dans *Code_Aster*, MEFISTO les calcule par différence finie.
- après convergence, MEFISTO imprime les résultats dans le fichier de message.

Table des Matières

1But.....	1
2Syntaxe.....	4
3Opérandes.....	7
3.1Mot clé LOGICIEL.....	7
3.2Mot clé VERSION.....	7
3.3Mot clé UNITE_ESCL.....	7
3.4Mot clé MESS_ASTER.....	7
3.5Mot clé facteur VARIABLE.....	7
3.5.1Mot clé NOM.....	7
3.5.2Mot clé LOI.....	8
3.5.3Mot clé VALE_MIN.....	8
3.5.4Mot clé VALE_MAX.....	8
3.5.5La valeur moyenne.....	8
3.5.6L'écart-type.....	9
3.5.7Les points de calcul.....	9
3.5.8Le gradient.....	9
3.6Mot clé MATRICE.....	10
3.7Le seuil de défaillance.....	10
3.7.1Mot clé SEUIL.....	10
3.7.2Mot clé SEUIL_TYPE.....	10
3.8Recherche du point de conception.....	11
3.8.1Mot clé RECH_PT_CONCEPT.....	11
3.8.2Mot clé EPSILON_U.....	11
3.8.3Mot clé EPSILON_G.....	11
3.8.4Mot clé TAU.....	11
3.8.5Mot clé OMEGA.....	11
3.8.6Mot clé ITER_MAX.....	11
3.9Mot clé METHODE_FORM.....	11
3.10Mot clé METHODE_SORM.....	12
3.11Tirage d'importance.....	12
3.11.1Mot clé TIRAGE_IMPORTANCE.....	12
3.11.2Mot clé NB_SIMULATION.....	12
3.12Mot clé POLYNOME_TAYLOR.....	12
3.13Mot clé HGRAD.....	12
3.14Mot clé HHESS.....	12
3.15Recherche de plan d'expérience.....	12
3.15.1Mot clé PLAN_EXPERIENCE.....	12
3.15.2Mot clé ALPHA.....	13

3.15.3Mot clé BETA.....	13
3.16Test de la sphère.....	13
3.16.1Mot clé T_SPHERE.....	13
3.16.2Mot clé METHODE_TEST.....	13
3.16.3Mot clé NB_POINT.....	13
3.17Test du maximum fort.....	13
3.17.1Mot clé T_MAXIMUM_FORT.....	13
3.17.2Mot clé COS_LIM.....	13
3.17.3Mot clé DPROB.....	14
3.18Mot clé T_HESSIEN.....	14
3.19Opérande INFO.....	14
4Exemples.....	15
4.1Fichier de commandes contenant la macro-commande (.comm).....	15
4.2Fichier de commandes contenant le calcul physique (.com1).....	17
5Bibliographie.....	21

2 Syntaxe

```
lr = MACR_FIABILITE (

# Choix du logiciel probabiliste :

    ◇ LOGICIEL = / 'MEFISTO' [DEFAULT]
    ◇ VERSION  = / 'V3_2', [DEFAULT]
                  / 'V3_N', [TXM]

# Caractérisation du calcul déterministe :

    ◆ UNITE_ESCL = num_unite_escl, [I]
    ◇ MESS_ASTER = / 'DERNIER', [DEFAULT]
                  / 'AUCUN', [TXM]
                  / 'TOUS', [TXM]

# Caractérisation des variables aléatoires, paramètres sensibles
# de la simulation :

    ◆ VARIABLE =_F (
        ◆ NOM = nom_vari, [TXM]
        ◆ LOI = / 'UNIFORME', [TXM]
                / 'NORMALE', [TXM]
                / 'LOGNORMALE', [TXM]
                / 'NORMALE_TRONQUEE', [TXM]

# Si la variable suit la loi uniforme :
    ◆ VALE_MIN = val_min, [R]
    ◆ VALE_MAX = val_max, [R]

# Ou si la variable suit la loi normale :
    ◆ VALE_MOY = val_moy, [R]
    ◆ ECART_TYPE = ecart_type, [R]

# Ou si la variable suit la loi log-normale :
    ◆ VALE_MIN = val_min, [R]
    # Soit le couple :
    ◆ VALE_MOY = val_moy, [R]
    ◆ ECART_TYPE = ecart_type, [R]
    # Soit le couple :
    ◆ VALE_MOY_PHY = val_moy_physique, [R]
    ◆ ECART_TYPE_PHY = ecart_type_physique, [R]

# Ou si la variable suit la loi normale tronquée :
    ◆ VALE_MOY = val_moy, [R]
    ◆ ECART_TYPE = ecart_type, [R]
    ◆ VALE_MIN = val_min, [R]
    ◆ VALE_MAX = val_max, [R]
# Finsi

    ◇ POINT_INI = pt_init, [R]
    ◇ POINT_REF = pt_refe, [R]

# Si on ne cherche pas le point de conception :
    ◆ POINT_CONCEPT = pt_concept, [R]
# Finsi
```

```
# Méthode de calcul des gradients :
♦ GRADIENT = / 'OUI',                                [TXM]
              / 'NON',                                [TXM]
# Si le gradient n'est pas calculé automatiquement
♦ INCREMENT = val_increment,                          [R]
# Finsi

),

♦ SEUIL = val_seuil,                                  [R]
♦ SEUIL_TYPE = / 'MINIMUM',                            [TXM]
                / 'MAXIMUM',                            [TXM]

# Matrice de corrélation enter les variables :
♦ MATRICE = mat_correlation,                          [l_R]

# Recherche du point de conception :
♦ RECH_PT_CONCEPT = / 'OUI',                        [TXM]
                      / 'NON',                        [TXM]

# Si on recherche le point de conception
♦ EPSILON_U = / epsi_u,                                [R]
               / 1.0E-2,                                [DEFAULT]

♦ EPSILON_G = / epsi_g,                                [R]
               / 1.0E-2,                                [DEFAULT]

♦ TAU = / tau,                                          [R]
        / 5.0E-1,                                       [DEFAULT]

♦ OMEGA = / omega,                                     [R]
          / 1.0E-4,                                     [DEFAULT]

♦ ITER_MAX = / iter_max,                               [I]
              / 50,                                     [DEFAULT]

# Finsi

# Choix de la méthode de recherche de la probabilité de défaillance :
♦ METHODE_FORM = / 'OUI',                              [DEFAULT]
                  / 'NON',                              [TXM]

♦ METHODE_SORM = / 'OUI',                              [TXM]
                  / 'NON',                              [DEFAULT]

♦ TIRAGE_IMPORTANCE = / 'OUI',                          [TXM]
                      / 'NON',                          [DEFAULT]

# Si on veut une surface de défaillance : TIRAGE_IMPORTANCE == 'OUI'
♦ NB_SIMULATION = / nb_simu,                            [I]
                  / 3,                                    [DEFAULT]

# Finsi

♦ POLYNOME_TAYLOR = / 'OUI',                            [TXM]
                    / 'NON',                            [DEFAULT]

♦ HGRAD = / h_grad,                                     [R]
          / 1.0E-2,                                     [DEFAULT]

♦ HHES = / h_hess,                                      [R]
          / 1.0E-2,                                     [DEFAULT]
```

```

    ◇ PLAN_EXPERIENCE = / 'OUI', [TXM]
                        / 'NON', [DEFAULT]

# Si on cherche un plan d'expérience : PLAN_EXPERIENCE == 'OUI'
    ◇ ALPHA = / alpha, [R]
              / 2.0E-1, [DEFAULT]

    ◇ BETA = / beta, [R]
            / 4.0E-1, [DEFAULT]
# Finsi

# Tests de convergence :
    ◇ T_SPHERE = / 'OUI', [TXM]
                / 'NON', [DEFAULT]

# Si on active le test de la sphère :
    ◇ METHODE = / 'GAUSSIENNE', [DEFAULT]
              / 'PARAMETRIQUE', [TXM]
              / 'REJECTION', [TXM]

    ◇ NB_POINT = / nb_point, [I]
                 / 60, [DEFAULT]
# Finsi

    ◇ T_MAXIMUM_FORT = / 'OUI', [TXM]
                      / 'NON', [DEFAULT]

# Si on active le test du maximum fort : T_MAXIMUM_FORT == 'OUI'
    ◇ COS_LIM = / cos_limit, [R]
               / 1.0, [DEFAULT]

    ◇ DPROB = / d_proba, [R]
              / 4.3E-1, [DEFAULT]
# Finsi

    ◇ T_HESSIEN = / 'OUI', [TXM]
                 / 'NON', [DEFAULT]

    ◇ INFO = / 1, [DEFAULT]
            / 2, [I]

)
```

3 Opérandes

3.1 Mot clé LOGICIEL

◇ LOGICIEL = / 'MEFISTO' , [DEFAULT]

Ce mot clé permet de sélectionner le logiciel de fiabilité. Actuellement, seul le logiciel MEFISTO est disponible.

En ce qui concerne le logiciel MEFISTO, le lecteur se reportera aux références [bib1], [bib2], [bib3] pour connaître la signification des différents paramètres des méthodes employées.

3.2 Mot clé VERSION

◇ VERSION = / 'V3_2', [DEFAULT]
/ 'V3_N', [TXM]

Permet de sélectionner la version du logiciel de fiabilité.

3.3 Mot clé UNITE_ESCL

◆ UNITE_ESCL = num_unite_escl [I]

C'est le numéro de l'unité au sens fortran du fichier de commandes qui permet de réaliser le calcul déterministe.

3.4 Mot clé MESS_ASTER

◇ MESS_ASTER = / 'DERNIER', [DEFAULT]
/ 'AUCUN', [TXM]
/ 'TOUS', [TXM]

Permet :

- d'imprimer le compte rendu du DERNIER calcul déterministe dans le fichier de message ;
- de n'imprimer AUCUN compte rendu des calculs déterministes dans le fichier de message ;
- d'imprimer le compte rendu de TOUS les calculs déterministes dans le fichier de message.

3.5 Mot clé facteur VARIABLE

Pour chaque variable, il est nécessaire de donner ses caractéristiques.

3.5.1 Mot clé NOM

◆ NOM = nom_vari, [TXM]

On précise ici le nom de la variable aléatoire (paramètre sensible). Le nom donné ici doit être identique à celui donné au concept para_sensi défini dans le fichier de commandes correspondant au calcul déterministe ; voir l'exemple du paragraphe [§4].

3.5.2 Mot clé LOI

```
♦ LOI = / 'UNIFORME', [TXM]
        / 'NORMALE', [TXM]
        / 'LOG NORMALE', [TXM]
        / 'NORMALE_TRONQUEE', [TXM]
```

On indique ici le nom de la loi de probabilité qui gouvernera la variable aléatoire.

Selon les lois, des valeurs minimales, maximales, etc. sont à donner. Le tableau suivant récapitule les besoins.

Mot-clé \ Loi	Uniforme	Normale	Log normale	Normale tronquée
VALE_MIN	Obligatoire	Sans objet	Obligatoire	Obligatoire
VALE_MAX	Obligatoire	Sans objet	Sans objet	Obligatoire
VALE_MOY	Sans objet	Obligatoire	L'un des deux est	Obligatoire
VALE_MOY_PHY	Sans objet	Sans objet	obligatoire.	Sans objet
ECART_TYPE	Sans objet	Obligatoire	L'un des deux est	Obligatoire
ECART_TYPE_PHY	Sans objet	Sans objet	obligatoire.	Sans objet

3.5.3 Mot clé VALE_MIN

```
♦ VALE_MIN = val_min, [R]
```

On précise ici la borne inférieure de la loi de probabilité. Il est obligatoire de renseigner le mot clé VALE_MIN si l'utilisateur choisit une des lois de probabilité suivantes : 'UNIFORME', 'LOG NORMALE' ou 'NORMALE_TRONQUEE'.

3.5.4 Mot clé VALE_MAX

```
♦ VALE_MAX = val_max, [R]
```

On précise ici la borne supérieure de la loi de probabilité. Il est obligatoire de renseigner le mot clé VALE_MAX si l'utilisateur choisit une des lois de probabilité suivantes : 'UNIFORME' ou 'NORMALE_TRONQUEE'.

3.5.5 La valeur moyenne

On précise ici la moyenne de la loi de probabilité. Dans le cas de la loi 'LOG NORMALE', l'utilisateur a le choix entre VALE_MOY s'il souhaite travailler dans l'espace standard et VALE_MOY_PHY s'il souhaite travailler dans l'espace physique.

3.5.5.1 Mot clé VALE_MOY

```
♦ VALE_MOY = val_moy, [R]
```

On précise ici la moyenne de la loi de probabilité. Il est obligatoire de renseigner le mot clé VALE_MOY si l'utilisateur choisit une des lois de probabilité suivantes : 'NORMALE' ou 'NORMALE_TRONQUEE'. Dans le cas de la loi 'LOG NORMALE', l'utilisateur a le choix entre VALE_MOY s'il souhaite travailler dans l'espace standard et VALE_MOY_PHY s'il souhaite travailler dans l'espace physique.

3.5.5.2 Mot clé VALE_MOY_PHY

```
♦ VALE_MOY_PHY = val_moy_physique, [R]
```

On précise ici la moyenne de la loi de probabilité dans l'espace physique. Il est obligatoire de renseigner le mot clé VALE_MOY_PHY si l'utilisateur choisit la loi de probabilité : 'LOG NORMALE' et n'a pas renseigné le mot clé VALE_MOY.

3.5.6 L'écart-type

On précise ici l'écart type de la loi de probabilité. Dans le cas de la loi 'LOG NORMALE', l'utilisateur a le choix entre `ECART_TYPE` s'il souhaite travailler dans l'espace standard et `ECART_TYPE_PHY` s'il souhaite travailler dans l'espace physique.

3.5.6.1 Mot clé `ECART_TYPE`

◇ `ECART_TYPE = ecart_type,` [R]

On précise ici l'écart type de la loi de probabilité. Il est obligatoire de renseigner le mot clé `ECART_TYPE` si l'utilisateur choisit une des lois de probabilité suivantes : 'NORMALE' et 'NORMALE_TRONQUEE'. Dans le cas de la loi 'LOG NORMALE', l'utilisateur a le choix entre `ECART_TYPE` s'il souhaite travailler dans l'espace standard et `ECART_TYPE_PHY` s'il souhaite travailler dans l'espace physique

3.5.6.2 Mot clé `ECART_TYPE_PHY`

◇ `ECART_TYPE_PHY = ecart_type_physique,` [R]

On précise ici l'écart type de la loi de probabilité dans l'espace physique. Il est obligatoire de renseigner le mot clé `ECART_TYPE_PHY` si l'utilisateur choisit la loi de probabilité : 'LOG NORMALE' et n'a pas renseigné le mot clé `ECART_TYPE`.

3.5.7 Les points de calcul

3.5.7.1 Mot clé `POINT_INITIAL`

◇ `POINT_INITIAL = pt_init,` [R]

On indique ici la valeur du paramètre pour définir le point initial à partir duquel le logiciel probabiliste va chercher le point de conception. Si ce mot clé n'est pas renseigné, on prendra par défaut la valeur moyenne.

3.5.7.2 Mot clé `POINT_DE_REFERENCE`

◇ `POINT_DE_REFERENCE = pt_refe,` [R]

On indique ici la valeur du paramètre pour définir le point de référence qui sert à rendre adimensionnels les résultats de *Code_Aster* au sein du logiciel probabiliste. Si ce mot clé n'est pas renseigné, on prendra par défaut la valeur moyenne.

3.5.7.3 Mot clé `POINT_DE_CONCEPTION`

◇ `POINT_DE CONCEPTION = pt_concept,` [R]

Si la recherche du point de conception n'est pas demandé par l'utilisateur (i.e. `RECH_PT_CONCEPT = 'NON'`), il est obligatoire de donner la valeur correspondant au point de conception.

3.5.8 Le gradient

On précise ici comment est calcul le gradient de la valeur cible par rapport à la variable.

3.5.8.1 Mot clé GRADIENT

♦ GRADIENT = / 'OUI', [TXM]
/ 'NON', [TXM]

Pour la variable aléatoire (paramètre sensible) on indique si le gradient est fourni directement par un calcul de sensibilité dans *Code_Aster* ou s'il ne l'est pas. Dans le cas où le gradient n'est pas fourni par le code physique (GRADIENT = 'NON'), le logiciel probabiliste le calculera par différence finie. Il faut alors préciser l'incrément.

3.5.8.2 Mot clé INCREMENT

♦ INCREMENT = val_increment, [R]

Si le gradient n'est pas fourni par un calcul de sensibilité dans *Code_Aster*® (GRADIENT = 'NON') il est obligatoire de donner une valeur au mot clé INCREMENT, de manière à ce que le logiciel probabiliste puisse le calculer par différence finie.

3.6 Mot clé MATRICE

♦ MATRICE = mat_correlation, [1_R]

Permet de définir la matrice de corrélation entre les variables aléatoires (paramètres sensibles). Les coefficients de corrélation appartiennent à l'intervalle [-1, 1]. S'il n'y a pas de corrélation entre les variables aléatoires, la matrice de corrélation est égale à la matrice identité.

La taille de cette matrice est donc égale au carré du nombre de variables aléatoires définies.

Si le mot-clé est absent, le calcul se fait avec la matrice identité.

3.7 Le seuil de défaillance

Le calcul de la probabilité de défaillance se fait sur le dépassement d'un seuil. Il faut donner deux informations : la valeur de ce seuil et comment il est franchi.

3.7.1 Mot clé SEUIL

♦ SEUIL = val_seuil, [R]

On indique ici la valeur du seuil dont on veut connaître la probabilité qu'il soit dépassé.

3.7.2 Mot clé SEUIL_TYPE

♦ SEUIL_TYPE = / 'MINIMUM', [TXM]
/ 'MAXIMUM', [TXM]

Permet de préciser le type de seuil à ne pas dépasser.

L'option SEUIL_TYPE = 'MINIMUM' signifie que la zone sûre est celle où la cible est supérieure au seuil ; on calcule la probabilité que la cible passe en-dessous de ce seuil minimum.

Symétriquement, l'option SEUIL_TYPE = 'MAXIMUM' signifie que la zone sûre est celle où la cible est inférieure au seuil ; on calcule la probabilité que la cible passe au-dessus de ce seuil maximum.

3.8 Recherche du point de conception

3.8.1 Mot clé RECH_PT_CONCEPT

♦ RECH_PT_CONCEPT = / 'OUI', [TXM]
/ 'NON', [TXM]

On précise ici si le logiciel probabiliste **doit** ou **ne doit pas** chercher le point de conception. Le point de conception est le point situé sur la frontière entre domaine de sécurité et domaine de défaillance et qui est à la distance minimale du centre de l'espace standard.

3.8.2 Mot clé EPSILON_U

◇ EPSILON_U = / epsi_u, [R]
/ 1.0E-2, [DEFAULT]

On indique ici la précision du test d'arrêt sur les points itératifs de l'espace standard.

3.8.3 Mot clé EPSILON_G

◇ EPSILON_G = / epsi_g, [R]
/ 1.0E-2, [DEFAULT]

On indique ici la précision du test d'arrêt sur la proximité de la surface d'état limite.

3.8.4 Mot clé TAU

◇ TAU = / tau, [R]
/ 5.0E-1, [DEFAULT]

On précise ici la valeur du premier paramètre qui sert à optimiser la direction de descente.

3.8.5 Mot clé OMEGA

◇ OMEGA = / omega, [R]
/ 1.0E-4, [DEFAULT]

On précise ici la valeur du second paramètre qui sert à optimiser la direction de descente.

3.8.6 Mot clé ITER_MAX

◇ ITER_MAX = / iter_max, [I]
/ 50, [DEFAULT]

On précise ici le nombre maximum d'itérations, pour l'algorithme de minimisation sous contrainte de MEFISTO, au delà duquel l'algorithme s'arrête.

3.9 Mot clé METHODE_FORM

◇ METHODE_FORM = / 'OUI', [DEFAULT]
/ 'NON', [TXM]

Permet d'activer l'option du logiciel probabiliste qui servira à calculer la probabilité de défaillance par la méthode FORM. FORM remplace la surface d'état limite par un hyperplan au voisinage du point de conception.

3.10 Mot clé METHODE_SORM

```
◇ METHODE_SORM = / 'OUI', [TXM]
                  / 'NON', [DEFAULT]
```

Permet d'activer l'option du logiciel probabiliste qui servira à calculer la probabilité de défaillance par la méthode SORM. SORM remplace la surface d'état limite par un polynôme de degré 2 au voisinage du point de conception.

3.11 Tirage d'importance

3.11.1 Mot clé TIRAGE_IMPORTANCE

```
◇ TIRAGE_IMPORTANCE = / 'OUI', [TXM]
                      / 'NON', [DEFAULT]
```

Permet de rechercher la probabilité de défaillance avec le tirage d'importance.

3.11.2 Mot clé NB_SIMULATION

```
◇ NB_SIMULATION = / nb_simu, [I]
                  / 3, [DEFAULT]
```

On spécifie ici le nombre de simulations pour le tirage d'importance.

3.12 Mot clé POLYNOME_TAYLOR

```
◇ POLYNOME_TAYLOR = / 'OUI', [TXM]
                    / 'NON', [DEFAULT]
```

Permet de préciser si l'on veut créer une surface de réponse polynomiale.

3.13 Mot clé HGRAD

```
◇ HGRAD = / h_grad, [R]
           / 1.0E-2, [DEFAULT]
```

On précise ici la valeur du pas incrémental pour le calcul des gradients par différence finie. Cette valeur est exprimée dans le repère transformé.

3.14 Mot clé HHES

```
◇ HHES = / h_hess, [R]
          / 1.0E-2, [DEFAULT]
```

On précise ici la valeur du pas incrémental pour le calcul des dérivées secondes. Cette valeur est exprimée dans le repère transformé.

3.15 Recherche de plan d'expérience

3.15.1 Mot clé PLAN_EXPERIENCE

```
◇ PLAN_EXPERIENCE = / 'OUI', [TXM]
                    / 'NON', [DEFAULT]
```

Calcule ou ne calcule pas le plan d'expériences centré sur le point de conception.

3.15.2 Mot clé ALPHA

```
◇ ALPHA = / alpha, [R]
           / 2.0E-1, [DEFAULT]
```

On précise ici la valeur de la maille du plan de type composite centré.

3.15.3 Mot clé BETA

```
◇ BETA = / beta, [R]
          / 4.0E-1, [DEFAULT]
```

On précise ici la valeur de la maille du plan de type factoriel.

3.16 Test de la sphère

On peut tester la fonction de performance au point de conception avec le test de la sphère (voir les références [bib1], [bib2], [bib3]).

3.16.1 Mot clé T_SPHERE

```
◇ T_SPHERE = / 'OUI', [TXM]
              / 'NON', [DEFAULT]
```

Active le test de la sphère.

3.16.2 Mot clé METHODE_TEST

```
◇ METHODE_TEST = / 'GAUSSIENNE', [DEFAULT]
                  / 'PARAMETRIQUE', [TXM]
                  / 'REJECTION', [TXM]
```

On précise ici la méthode qui sera utilisée pour exécuter le test de la sphère.

3.16.3 Mot clé NB_POINT

```
◇ NB_POINT = / nb_point, [I]
              / 60, [DEFAULT]
```

On précise le nombre points qui seront utilisés pour exécuter le test de la sphère

3.17 Test du maximum fort

On peut tester la fonction de performance au point de conception avec le test du maximum fort (voir les références [R1, R2, R3]).

3.17.1 Mot clé T_MAXIMUM_FORT

```
◇ T_MAXIMUM_FORT = / 'OUI', [TXM]
                   / 'NON', [DEFAULT]
```

Active le test du maximum fort.

3.17.2 Mot clé COS_LIM

```
◇ COS_LIM = / cos_lim, [R]
             / 1.0, [DEFAULT]
```

On précise ici la valeur du cosinus limite dont l'angle définit le voisinage du point de conception.

3.17.3 Mot clé DPROB

◇ DPROB = / d_proba, [R]
/ 4.3E-1, [DEFAULT]

On précise ici la valeur du rapport entre la densité de probabilité des points sur la sphère de rayon β et ceux de la sphère de rayon $\beta + d\beta$.

3.18 Mot clé T_HESSIEN

◇ T_HESSIEN = / 'OUI', [TXM]
/ 'NON', [DEFAULT]

On peut tester la fonction de performance au point de conception avec le test du Hessian.

3.19 Opérande INFO

◇ INFO =

Indique le niveau d'impression des résultats de l'opérateur :

- 1 : aucune impression,
- 2 : impression d'informations relatives au maillage.

4 Exemples

L'exemple décrit ici correspond au cas test **fiab001a** . On a noté en caractères gras la similitude de noms entre la description d'une variable aléatoire dans la macro-commande et le concept paramètre sensible dans le jeu de commandes du calcul déterministe.

4.1 Fichier de commandes contenant la macro-commande (.comm)

```
# Préliminaire :
# Ce cas-test est un cas de validation du couplage entre Code_ASTER et
# le logiciel fiabiliste MEFISTO. Il peut servir d'exemple pour
# la réalisation d'une étude fiabiliste mais n'est en aucun cas une
# évaluation de cette technique.
#
# Description du cas :
# Le fonctionnement nominal correspond au lancement du calcul
# déterministe avec les valeurs moyennes pour les 5 paramètres :
#     PA moyen = 1000.
#     PB moyen = 8000.
#     E1 moyen = 430000.
#     E2 moyen = 380000.
#     E3 moyen = 130000.
# Ces conditions nominales entraînent dans le coin bas-gauche de la
# structure une contrainte de composante SIXX = -2.6795397166E+05
#
# Pour les besoins du cas-test, on déclare que la structure est en mode
# de défaillance quand la valeur absolue de cette contrainte est
# supérieure a la valeur nominale. Par exemple, quand on dépasse une
# valeur absolue de 2.71E5. Autrement dit, il faut déclarer un seuil de
# -2.71E+5 et dire que c'est un minimum a ne pas franchir.
#
# Quelques remarques :
# . Le jeu de commandes déterministe associe (dans le .38) a été
#   crée en donnant aux paramètres sensibles leurs valeurs moyennes. En
#   fait, on peut mettre n'importe quoi ! En effet, les calculs seront
#   effectués a partir de valeurs déduites des lois des paramètres
#   définis ci-après. On a mis les valeurs moyennes car c'est ainsi
#   que le calcul déterministe avait été mis au point et rien n'a été
#   change.
# . Le seuil de défaillance est ici très proche du fonctionnement
#   nominal. C'est volontaire pour avoir très peu d'itération dans la
#   recherche du point de conception par MEFISTO et avoir ainsi un
#   cas-test rapide. Mais cela n'a aucun intérêt physique.
#
DEBUT(CODE=_F(NOM='FIAB001A',NIV_PUB_WEB='INTRANET'),PAR_LOT='OUI');
#
RESUFIAB = MACR_FIABILITE(
    INFO=1,
    MESS_ASTER = 'DERNIER',
    LOGICIEL = 'MEFISTO',
    VERSION='V3_2',
    UNITE_ESCL = 38,
    RECH_PT_CONCEPT='OUI',
    SEUIL=-2.71E+05,
    SEUIL_TYPE = 'MINIMUM',
    VARIABLE=(
        _F(NOM=' E1 ',
            LOI='NORMALE',
```

```
GRADIENT='NON',
INCREMENT=0.5,
VALE_MOY=430000.0,
ECART_TYPE=2000.0,
POINT_REF=431000.0,)),
_F (NOM=' E2 ',
    LOI='NORMALE',
    GRADIENT='NON',
    INCREMENT=0.5,
    VALE_MOY=380000.0,
    ECART_TYPE=2000.0,
    POINT_INI=381400.0,)),
_F (NOM=' E3 ',
    LOI='NORMALE',
    GRADIENT='OUI',
    VALE_MOY=130000.0,
    ECART_TYPE=1000.0,)),
_F (NOM=' PA ',
    LOI='NORMALE',
    GRADIENT='OUI',
    VALE_MOY=1000.0,
    ECART_TYPE=100.0,
    POINT_INI=1050.0,)),
_F (NOM=' PB ',
    LOI='NORMALE',
    GRADIENT='OUI',
    VALE_MOY=8000.0,
    ECART_TYPE=150.0,
    POINT_INI=8110.0,
    POINT_REF=8080.0,)),
),
T_SPHERE = 'NON',
);

#
# Le retour de la macro est un concept de type LISTR8. Il contient
# une seule valeur : la probabilité de défaillance.
# On transforme cette liste en une fonction pour le test de non régression.
#
LR_BIDON = DEFI_LIST_REEL ( VALE = ( 1. ) , );
#
R_FIAB = DEFI_FONCTION ( NOM_PARA = 'INST',
                        VALE_PARA = LR_BIDON,
                        VALE_FONC = RESUFIAB,
                        INFO = 1 );
#
TEST_FONCTION ( VALEUR = _F( FONCTION = R_FIAB,
                              PRECISION = 1.E-6,
                              VALE_PARA = 1.,
                              REFERENCE = 'ANALYTIQUE',
                              VALE_REFE = 0.278,
                              CRITERE = 'ABSOLU',
                              ),
                );
#
FIN();
```


4.2 Fichier de commandes contenant le calcul physique (.com1)

```
DEBUT(CODE=_F(NOM='FIAB001A',NIV_PUB_WEB='INTRANET'),PAR_LOT='OUI');
#
# Ce cas-test est identique au cas-test sensm07a, qui sert d'exemple a
# la notice d'utilisation des sensibilités, U2.08.02.
# C'est un domaine rectangulaire 2D, compose de 3 matériaux distincts.
# La structure est encastre sur la gauche. On applique des pressions
# sur la face supérieure.
# On s'intéresse aux sensibilités par rapport aux 2 pressions imposées
# et aux trois modules d'Young.
#
# Voici le schéma simplifie du domaine de calcul.
# Les groupes y sont désignés par leurs noms.
#
#   'COIN_HG'      'BORD_H_1'      'BORD_H_2'      'COIN_HD'
#   y=6 *-----*
#   !               !               !
#   y=5 !           !               !
#   !               !               !
#   y=4 !   'ZONE_1' !   'ZONE_2'   !
#   !               !               !
#   y=3 !           !               !
#   !               !               !
#   y=2 !           !               !
#   !               !               !
#   y=1 !           !---!           !
#   !               ! <---! 'ZONE_3' !
#   y=0 *-----*
#   'COIN_BG'      'COIN_BD'
#   x = 0  1  2  3  4  5  6  7  8  9 10 11 12
#
# 1. Définitions de fonctions
# 1.1. Définition des paramètres sensibles
#
PA = DEFI_PARA_SENSI (VALE=1000.);
PB = DEFI_PARA_SENSI (VALE=8000.);
E1 = DEFI_PARA_SENSI (VALE=430000.);
E2 = DEFI_PARA_SENSI (VALE=380000.);
E3 = DEFI_PARA_SENSI (VALE=130000.);
NU3 = DEFI_PARA_SENSI (VALE=0.27);
#
# 1.2. Définition des constantes
#
NU1=DEFI_CONSTANTE (VALE=0.33);
NU2=DEFI_CONSTANTE (VALE=0.38);
#
# 2. Définition des matériaux
#
mater_1=DEFI_MATERIAU ( ELAS_FO=_F(E= E1 ,NU=NU1) );
#
mater_2=DEFI_MATERIAU ( ELAS_FO=_F(E= E2 ,NU=NU2) );
#
mater_3=DEFI_MATERIAU ( ELAS_FO=_F(E= E3 ,NU=NU3) );
#
# 3. Le maillage
# 3.1. Lecture du maillage
#
```

```
PRE_GMSH();
maill_d1=LIRE_MALLAGE();
maill_d2=CREA_MALLAGE( MALLAGE = maill_d1,LINE_QUAD = _F(TOUT='OUI'))
#

# 3.2. Nommage des groupes
#
maill_d2=DEFI_GROUP(reuse =maill_d2,
    MALLAGE=maill_d2,
    CREA_GROUP_MA=( _F(GROUP_MA='GM11',NOM='BORD_H_1'),
        _F(GROUP_MA='GM12',NOM='BORD_H_2'),
        _F(GROUP_MA='GM13',NOM='BORD_GAU'),
        _F(GROUP_MA='GM21',NOM='ZONE_1'),
        _F(GROUP_MA='GM22',NOM='ZONE_2'),
        _F(GROUP_MA='GM23',NOM='ZONE_3')),
    CREA_GROUP_NO=_F(GROUP_MA=('GM1' , 'GM2' , 'GM3' , 'GM4' ),
        NOM=('COIN_BG', 'COIN_BD', 'COIN_HD', 'COIN_HG')) )
;
#
# 4. Le modèle
#
modele=AFFE_MODELE(MALLAGE=maill_d2,
    AFFE=_F(TOUT='OUI',
        PHENOMENE='MECANIQUE',
        MODELISATION='D_PLAN')) ;
#
maill_d2=MODI_MALLAGE(reuse =maill_d2,
    MALLAGE=maill_d2,
    ORIE_PEAU_2D=_F(GROUP_MA=('BORD_H_1', 'BORD_H_2', 'BORD_GAU')),
    MODELE=modele);
#
# 5. Les chargements
#
encastre=AFFE_CHAR_MECA(MODELE=modele,
    DDL_IMPO=_F(GROUP_NO='COIN_BG',DY=0.0),
    FACE_IMPO=_F(GROUP_MA='BORD_GAU',DNOR=0.0) );
#
pression=AFFE_CHAR_MECA_F(MODELE=modele,
    PRES_REP=( _F(GROUP_MA='BORD_H_1',PRES= PA ),
        _F(GROUP_MA='BORD_H_2',PRES= PB ) ) );
#
# 6. Mise en place des matériaux
#
ch_mater=AFFE_MATERIAU(MALLAGE=maill_d2,
    MODELE=modele,
    AFFE=( _F(GROUP_MA='ZONE_1',MATER=mater_1),
        _F(GROUP_MA='ZONE_2',MATER=mater_2),
        _F(GROUP_MA='ZONE_3',MATER=mater_3)) );
#
# 7. Calcul avec dérivations
#
résultat=MECA_STATIQUE(MODELE=modele,
    CHAM_MATER=ch_mater,
    EXCIT=( _F(CHARGE=encastre),
        _F(CHARGE=pression)),
    SENSIBILITE=( E3 , PA , PB ),
    SOLVEUR=_F(NPREC=8,
        METHODE='MULT_FRONT',
        STOP_SINGULIER='OUI',
        RENUM='MDA') ,
```

```
);  
  
#  
# 8. Autres champs  
# 8.1. Les contraintes standard  
#  
résultat=CALC_ELEM(reuse =resultat, RESULTAT=resultat,  
MODELE=modele,  
CHAM_MATER=ch_mater,  
EXCIT=( _F(CHARGE=encastre),  
_F(CHARGE=pression)),  
OPTION=('SIGM_ELNO','ERRE_ELGA_NORE'));  
  
resultat=CALC_NO(reuse =resultat,  
RESULTAT=resultat,  
EXCIT=( _F(CHARGE=encastre),  
_F(CHARGE=pression)),  
OPTION='SIGM_NOEU');  
  
#  
# 8.2. Les derivees des contraintes aux points de Gauss  
#  
resultat=CALC_ELEM(reuse =resultat,  
RESULTAT=resultat,  
SENSIBILITE=( E3 , PA , PB ),  
MODELE=modele,  
CHAM_MATER=ch_mater,  
EXCIT=( _F(CHARGE=encastre),  
_F(CHARGE=pression)),  
OPTION=('SIEF_ELGA','SIGM_ELNO'));  
  
#  
# 8.3. Les dérivées des contraintes aux nœuds  
#  
resultat=CALC_NO(reuse =resultat,  
RESULTAT=resultat,  
SENSIBILITE=( E3 , PA , PB ),  
EXCIT=( _F(CHARGE=encastre),  
_F(CHARGE=pression)),  
OPTION='SIGM_NOEU');  
  
#  
# 9. On crée des tables contenant une seule valeur : la composante SIXX  
# de la contrainte dans le coin en bas à gauche, ou de ses dérivées.  
#  
Cible = POST_RELEVE_T( ACTION=_F( GROUP_NO = 'COIN_BG',  
INTITULE = 'SIGXX COIN BAS A GAUCHE',  
RESULTAT = resultat,  
NOM_CHAM = 'SIGM_NOEU',  
NOM_CMP = ( 'SIXX',),  
OPERATION = 'EXTRACTION',),  
_F( GROUP_NO = 'COIN_BG',  
INTITULE = 'GRADIENT E3 COIN BAS A GAUCHE',  
RESULTAT = resultat,  
SENSIBILITE = ( E3 ),  
NOM_CHAM = 'SIGM_NOEU',  
NOM_CMP = ( 'SIXX',),  
OPERATION = 'EXTRACTION',),  
_F( GROUP_NO = 'COIN_BG',  
INTITULE = 'GRADIENT PA COIN BAS A GAUCHE',  
RESULTAT = resultat,  
SENSIBILITE = ( PA ),
```

```

                                NOM_CHAM = 'SIGM_NOEU',
                                NOM_CMP = ( 'SIXX', ),
                                OPERATION = 'EXTRACTION', ),
                                _F( GROUP_NO = 'COIN_BG',
INTITULE = 'GRADIENT PB COIN BAS A GAUCHE',
                                RESULTAT = resultat,
                                SENSIBILITE = (PB),
                                NOM_CHAM = 'SIGM_NOEU',
                                NOM_CMP = ( 'SIXX', ),
                                OPERATION = 'EXTRACTION', ),
                                ), );

#

#
# 10. Impression des résultats avec le format attendu par le logiciel
#   fiabiliste
#
MACR_FIAB_IMPR( INFO = 1,
                TABLE_CIBLE = Cible, NOM_PARA_CIBLE = 'SIXX',
GRADIENTS=( _F(TABLE = Cible, PARA_SENSI = E3, NOM_PARA = 'SIXX'),
            _F(TABLE = Cible, PARA_SENSI = PA, NOM_PARA = 'SIXX'),
            _F(TABLE = Cible, PARA_SENSI = PB, NOM_PARA = 'SIXX'),
                ), );

#
FIN();
```

5 Bibliographie

- (1)DUTFOY A. ; Dossier de Conception et de Validation de MEFISTO version 2.2 ; Rapport HT-52/01/021/A ; Août 2001.
- (2)DUTFOY A. ; Dossier de Conception et de Validation de MEFISTO : nouveautés de la version 2.4 ; Rapport HT-52/02/008/A ; Avril 2002.
- (3)DUTFOY A. ; Manuel théorique de Conception et de Validation ; Rapport HT-52/03/001/A ; Janvier 2003.