Organisme(s) : EDF-R&D/SINETICS

Manuel d’Utilisation
Fascicule U2.08 : Fonctions avancées et contrôle des calculs
Document : U2.08.02

Notice d’utilisation des calculs de sensibilité

Résumé :

Calculer la sensibilité d’un résultat à un paramètre donné suppose deux interventions :

- définir une donnée comme étant un paramètre sensible,
- activer le calcul effectif de la sensibilité.

Ce document présente l’ensemble des opérations à faire pour cela. Il détaille chacune des commandes concernées. Un exemple illustre les recommandations au fur et à mesure.

Table des matières

1 Introduction

Quel que soit le type de problème envisagé, thermique, mécanique, etc., le *Code_Aster* produit deux types de résultats : locaux ou globaux. Ce peut être un champ réparti sur le maillage, comme la température ou les contraintes, ou ce peut être une valeur globale, comme le taux de restitution d'énergie. Mais dans les deux cas, nous nous représentons ce résultat comme une fonction des données. Ces données sont d'origine variée. Nous trouvons ainsi :

- la géométrie du domaine de calcul,
- le mode de discrétisation à travers le choix du maillage,
- les conditions aux limites, comme les températures ou les déplacements imposés,
- les chargements, comme les sources d'énergie ou les pressions imposées,
- les propriétés de matériaux,
- les choix de calcul, comme les critères de convergence.

La liste n'est pas exhaustive. Evidemment, le résultat est sensible à chacune de ces données. Mais évidemment nous ne proposons pas de calcul automatique de toutes les sensibilités. Il est même de nombreux cas où une évaluation chiffrée n'a pas de sens. Comme par exemple chiffrer la sensibilité au choix de la méthode de résolution du système matriciel lié au calcul ? Les calculs de sensibilité disponibles avec le *Code_Aster* sont restreints aux cas où la donnée est un paramètre réel, clairement identifié dans le jeu de données, et où nous savons dériver la fonction qui lie cette donnée au résultat.

Prenons quelques exemples :

- choix du maillage : non, car ce n'est pas un paramètre réel,
- valeur de déplacement ou de pression imposée : oui,
- nombre de pas de temps : non, car c'est un entier,
- propriété des matériaux : oui et non ; oui si la valeur sensible est un module d'Young pur, non si on s'intéresse à une propriété donnée par une courbe point par point,
- critère de convergence : non, car nous ne savons pas dériver le résultat,
- etc.

Nous détaillerons les possibilités pour chaque type de problème. Il suffit de garder présent à l'esprit la règle énoncée plus haut : *Code_Aster* ne traite que les cas où le résultat est sous la forme $U(p)$, où p est un paramètre réel visible et où la dérivée partielle $\frac{\partial u}{\partial p}$ existe. Alors *Code_Aster* produira cette dérivée partielle, de même nature globale ou locale que le résultat, cette dérivée étant calculée au point nominal de fonctionnement.

Le sens physique attaché à la valeur de cette dérivée est loin d'être manifeste. Que dire d'une dérivée de contrainte par rapport à une valeur de pression imposée qui vaudrait 1,983 ? Sans même parler des unités… Comment interpréter ces résultats ? Comme nous venons de le voir, *Code_Aster* calcule une dérivée partielle. L'usage des dérivés est double à notre avis : une aide à la compréhension du phénomène étudié ou une insertion dans un processus plus global.

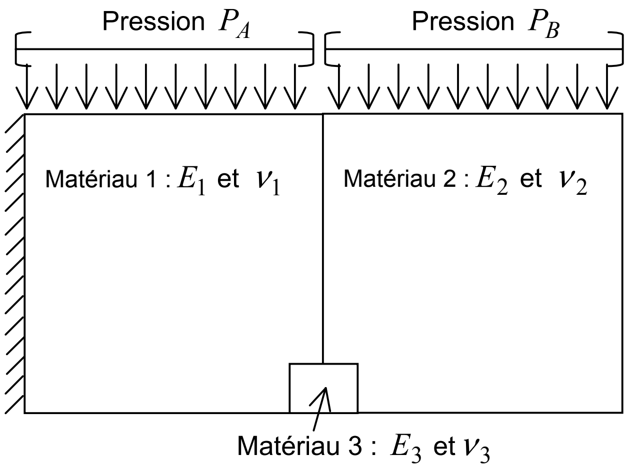
Dans un premier temps, la connaissance de la dérivée d'un résultat par rapport à des paramètres enrichit l'analyse du phénomène. Cela permet par exemple de localiser les zones où l'influence d'un changement est la plus grande. De même, on pourra comparer l'influence respective de deux données similaires. Si on doit faire une étude paramétrique, on pourra choisir de ne la faire que sur les paramètres les plus sensibles. Attention néanmoins à comparer des dérivées homogènes : les sensibilités à une pression externe et à une pression interne par exemple.

Dans un second temps, on pourra injecter les valeurs des dérivées obtenues dans un processus itératif. C'est le cas des algorithmes d'optimisation, de recalage, qui convergent en se basant sur la valeur de la fonction et de sa dérivée. C'est également le cas des calculs de mécanique fiabiliste utilisant la méthode FORM.

2 Un exemple emblématique

Nous allons illustrer les possibilités offertes par *Code_Aster* en examinant un exemple académique en mécanique. Cet exemple sera suivi jusqu'à la mise au point de son jeu de commandes.

Nous considérons une pièce formée de trois matériaux. Cette pièce est encastrée sur son bord gauche. Deux pressions sont appliquées sur les faces supérieures. Nous nous intéressons aux contraintes dans le troisième matériau. Plus particulièrement, nous aimerions connaître les sensibilités de ces contraintes aux différents modules d'Young et aux pressions imposées.



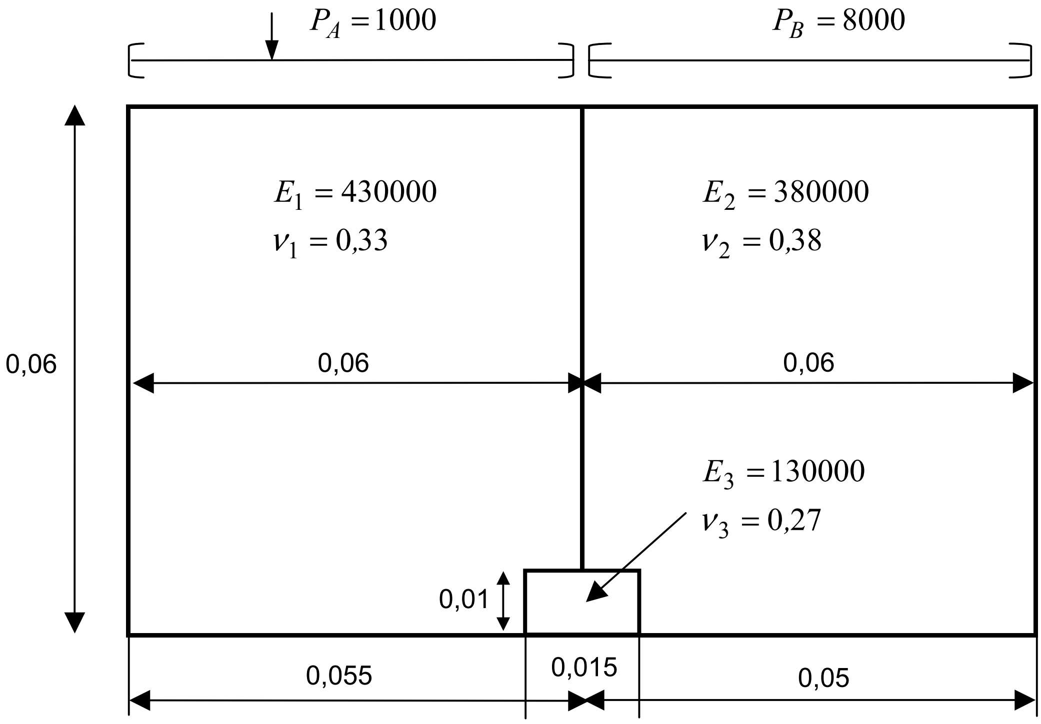
Comme nous le savons, le champ de contraintes est une fonction des données :

$$\sigma = \sigma(P_A, P_B, E_i, \nu_i, \text{géométrie, maillage, méthode, ...}).$$

Conformément aux règles énoncées plus haut, *Code_Aster* saura calculer chacune des dérivées partielles $\frac{\partial\sigma}{\partial P_A}, \frac{\partial\sigma}{\partial P_B}, \frac{\partial\sigma}{\partial E_1}, \frac{\partial\sigma}{\partial E_2}, \frac{\partial\sigma}{\partial E_3}$.

Le résultat σ est un champ exprimé aux points de Gauss de chaque élément ; c'est un tenseur de composantes σ_{xx}, σ_{yy}, etc. De la même manière, le résultat $\frac{\partial\sigma}{\partial P_A}$ sera un champ exprimé aux points de Gauss de chaque élément. Chacune de ses composantes sera la dérivée partielle de la composante correspondante de σ : $\frac{\partial\sigma_{xx}}{\partial P_A}, \frac{\partial\sigma_{yy}}{\partial P_A}$, etc. Nous obtiendrons ainsi automatiquement les dérivées partielles de toutes les composantes du tenseur des contraintes par rapport à chacun des paramètres mentionnés.

Avant d'aller plus loin dans la description du calcul de sensibilité, nous allons préciser les données numériques du problème, exprimées en système international.



En résolvant le problème de mécanique statique en déformation plane, nous obtenons les champs de déplacement et de contrainte suivants.

	Zone 1		Zone 2		Zone 3	
	Mini	**Maxi**	**Mini**	**Maxi**	**Mini**	**Maxi**
u_x	–0,0052	0,0066	–0,0072	0,0082	–0,0068	–0,0034
u_y	–0,0150	0	–0,0313	–0,0143	–0,0174	–0,0131
σ_{xx}	–189 468	68 192	–25 980	23 721	–20 213	–6 427
σ_{yy}	–280 144	15 453	–8 827	23 335	–3 826	204
σ_{zz}	–154 972	22 160	–11 165	77 065	–5 787	–1 935
σ_{xy}	–140 950	2 859	–11 182	149	–6 466	1 974

A ce stade de la description, le lecteur est invité à tester son sens physique et son appréciation des comportements mécaniques.

Question 1 : A laquelle des pressions P_A et P_B, le champ de contraintes dans la zone n° 3 est-il le plus sensible ?

Question 2 : Quel est l'ordre d'influence des trois modules d'Young E_1, E_2, E_3 sur ce même champ de contraintes ?

Si les réponses sont données au hasard, un rapide calcul montre que 8,3 % des lecteurs trouveront les deux bonnes réponses. Les utilisateurs de *Code_Aster* étant des experts, le taux de bonnes réponses sera très largement supérieur. Nous les départagerons avec la question suivante :

Question subsidiaire : Dans quel rapport sont les maximum des trois dérivés $\dfrac{\partial \sigma_{xx}}{\partial E_i}$ dans la zone n°3 ?

Dans le prochain chapitre, nous montrerons comment mettre en œuvre le calcul avec *Code_Aster* qui répondra à ces questions. Le lecteur verra dans le chapitre 4 si ses réponses étaient les bonnes …

3 Comment mettre en place un calcul de sensibilité

3.1 L'essentiel

Un calcul de sensibilité se fait grâce à l'introduction de la notion de "paramètre sensible". Autrement dit, si l'on veut dériver par rapport au module d'Young de l'un des matériaux du domaine, on définira un "paramètre sensible" qui représentera ce module d'Young. Ce paramètre sera vu sous deux aspects :

- comme une constante égale à la valeur nominale du module d'Young,
- comme un concept par rapport auquel on peut dériver.

Pour chaque dérivation souhaitée, on effectuera les opérations suivantes :

- définir le paramètre sensible avec sa valeur par la commande : `DEFI_PARA_SENSI`,
- utiliser ce paramètre sensible partout où sa valeur intervient dans les commandes (chargements, matériaux, …),
- demander à l'opérateur de résolution de dériver le résultat, avec le mot-clé : `SENSIBILITE=(…)`.

3.2 Définir les paramètres sensibles

Définir un paramètre sensible répond à ce double objectif : introduire dans le calcul un concept qui est égal à la valeur nominale de la donnée et qui soit reconnu comme "sensible". Pour cela, on utilise la commande `DEFI_PARA_SENSI` [U4.31.06]. Sa syntaxe est similaire à celle bien connue de `DEFI_CONSTANTE` :

```
PA = DEFI_PARA_SENSI (VALE = 1000.)
```

On doit donc définir ainsi tous les paramètres sensibles de la simulation.

Nous attirons l'attention sur ceci : la définition d'un paramètre sensible n'enclenche pas automatiquement le calcul de la dérivée. Le calcul sera fait uniquement pour les paramètres désignés ultérieurement. On peut donc en définir a priori beaucoup et, pour une simulation donnée, ne dériver que par rapport à quelques uns, voire aucun. Les données définies ainsi en excès seront utilisées comme de simples constantes.

Dans notre exemple, nous définirons a minima les cinq paramètres pour lesquels nous voulons obtenir des dérivés :

```
PA=DEFI_PARA_SENSI (VALE=1000.)
PB=DEFI_PARA_SENSI (VALE=8000.)
E1=DEFI_PARA_SENSI (VALE=430000.)
E2=DEFI_PARA_SENSI (VALE=380000.0)
E3=DEFI_PARA_SENSI (VALE=130000.0)
```

Conformément à la remarque précédente, rien ne nous interdit de définir d'autres paramètres sensibles, même si nous n'avons pas l'intention a priori de nous en servir.

```
NU3=DEFI_PARA_SENSI (VALE=0.27)
```

3.3 Utiliser les paramètres sensibles

La donnée associée à un paramètre sensible intervient en général dans un chargement ou dans la définition d'un matériau. Chacune de ces commandes sera activée en fournissant le paramètre sensible comme entrée. Cette entrée sera vue par la commande comme une fonction constante valant la valeur déclarée dans la définition du paramètre.

Dans notre exemple, les chargements en pression seront déclarés ainsi :

```
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) )    )
```

La définition des trois matériaux se passe de la même manière :

```
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)   )
```

On notera qu'utiliser un concept de type "paramètre sensible" en lieu et place d'une valeur numérique implique d'utiliser les définitions par fonctions des chargements ou matériaux. Toutefois, cela reste semblable aux cas où les valeurs sont définies par des concepts de type "constante", technique bien connue des utilisateurs de *Code_Aster*.

3.4 Lancer la dérivation du champ principal

Une fois que les paramètres sensibles ont été définis et utilisés, il ne reste qu'à lancer la dérivation. Cela se fait en insérant le mot-clé SENSIBILITE dans l'opérateur de calcul. Ce mot-clé est suivi de la liste des paramètres par rapport auxquels on souhaite dériver [U4.50.02]. Dans notre exemple, nous avons :

```
resultat=MECA_STATIQUE( MODELE=modele,
                        CHAM_MATER=ch_mater,
                        EXCIT=(_F(CHARGE=encastre  ),
                               _F(CHARGE=pression  )  ),
                        SENSIBILITE=(E1, E2, E3, PA, PB  )  )
```

Cette commande calculera simultanément le champ de déplacements et les cinq champ des dérivées de ce même déplacement par rapport à chacun des paramètres sensibles définis. Tous ces champs sont exprimés sur les nœuds du maillage.

Pour chaque type de problème, nous obtiendrons ainsi la dérivation du champ principal : la température en thermique, le déplacement en mécanique statique, etc.

3.5 Dériver les champs secondaires

Du champ principal, sont déduits des champs secondaires : flux de chaleur, déformations, contraintes, etc. Ces opérations sont activées par les commandes CALC_ELEM et CALC_NO. Ainsi le tenseur des contraintes est créé par :

```
resultat=CALC_ELEM( reuse =resultat,
                    RESULTAT=resultat,
                    MODELE=modele,
                    CHAM_MATER=ch_mater,
                    EXCIT=(_F (CHARGE=encastre),
                           _F (CHARGE=pression),
                    OPTION= ('SIEF_ELGA_DEPL', 'SIEF_ELNO_ELGA') )
```

Pour obtenir la dérivée des contraintes, il suffit d'insérer le mot-clé SENSIBILITE suivi de la liste des paramètres sensibles concernés.

```
resultat=CALC_ELEM( reuse =resultat,
                    RESULTAT=resultat,
                    SENSIBILITE=(E1, E2, E3, PA, PB),
                    MODELE=modele,
                    CHAM_MATER=ch_mater,
                    EXCIT=(_F (CHARGE=encastre),
                           _F (CHARGE=pression),
                    OPTION= ('SIEF_ELGA_DEPL', 'SIEF_ELNO_ELGA') )

resultat=CALC_NO (  reuse=resultat,
                    RESULTAT=resultat,
                    SENSIBILITE=(E1, E2, E3),
                    OPTION='SIGM_NOEU_DEPL')
```

Remarques :

- *Quand le mot-clé SENSIBILITE est inséré dans une commande CALC_ELEM ou CALC_NO, seul le champ dérivé est calculé.*
- *Pour calculer la dérivée d'un champ aux éléments, il faut au préalable avoir calculé le champ standard. En revanche, cela est inutile pour un champ aux nœuds car l'opérateur CALC_NO se contente de faire une moyenne aux nœuds d'un champ aux éléments.*

3.6 Post-traiter les résultats

Pour imprimer les champs de dérivés, il suffit d'insérer le mot-clé SENSIBILITE dans la commande IMPR_RESU. Ici encore, cela ne déclenchera que l'impression des champs dérivés par rapport aux paramètres concernés :

```
IMPR_RESU (RESU=_F( FORMAT='MED',
                    RESULTAT=resultat) )

IMPR_RESU (RESU=_F( FORMAT='MED',
                    RESULTAT=resultat,
                    SENSIBILITE= (E1, E2, E3, PA, PB) ) )
```

Toutes les options de la commande sont évidemment accessibles.

```
IMPR_RESU (RESU=_F( RESULTAT=resultat,
                    SENSIBILITE= (E1, E2, E3, PA, PB),
                    NOM_CHAM= 'SIEF_ELGA_DEPL',
                    GROUP_MA='ZONE_3',
                    VALE_MAX='OUI',
                    VALE_MIN='OUI') )
```

Au dela de l'impression, toutes les commandes qui manipulent les résultats ont été équipées du mot-clé SENSIBILITE : EXTR_RESU, POST_RELEVE_T etc. Le fonctionnement est similaire au standard : la commande réalise l'opération demandée mais sur les champs dérivés sélectionnés et exclusivement sur eux.

4 Exemple : calcul, commentaires et résultats

Voici le jeu de commandes complet associé à l'exemple décrit au chapitre 2.

```
DEBUT (CODE=_F(NOM='SENSM06A', NIV_PUB_WEB='INTERNET'))
#
# 1. Le maillage
# 1.1. Lecture du maillage
#
PRE_GMSH(MODI_QUAD='OUI')
maill_0=LIRE_MAILLAGE ( )
#
# 1.2. Nommage des groupes
#
maill_0= DEFI_GROUP (reuse =maill_0,
         MAILLAGE=maill_0,
         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')))
#
# 2. Definition des fonctions
# 2.1. Definition des parametres 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)
#
# 2.2 Definition des constantes
#
NU1=DEFI_CONSTANTE (VALE=0.33)
NU2=DEFI_CONSTANTE (VALE=0.38)
#
# 3. Definition des materiaux
#
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) )
#
# 4. Le modele
#
modele=AFFE_MODELE (MAILLAGE=maill_0,
                   AFFE=_F (  TOUT='OUI',
                              PHENOMENE='MECANIQUE',
                              MODELISATION='D_PLAN'))
#
```

```
# 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 materiaux
#
ch_mater=AFFE_MATERIAU(MAILLAGE=maill_0,
             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 derivations
#
resultat=MECA_STATIQUE(MODELE=modele,
                       CHAM_MATER=ch_mater,
                       EXCIT=(_F(CHARGE=encastre),
                              _F(CHARGE=pression)),
                       SENSIBILITE=(E1,E2,E3,PA,PB))
#
# 8. Autres champs
# 8.1. Les contraintes standard
#
resultat=CALC_ELEM( reuse =resultat,
                    RESULTAT=resultat,
                    MODELE=modele,
                    CHAM_MATER=ch_mater,
                    EXCIT=(_F(CHARGE=encastre),
                           _F(CHARGE=pression)),
                    OPTION=('SIEF_ELGA_DEPL', 'SIGM_ELNO_DEPL'))
#
# 8.2. Les derivees des contraintes aux points de Gauss
#
resultat=CALC_ELEM( reuse =resultat,
                    RESULTAT=resultat,
                    SENSIBILITE=(E1,E2,E3,PA,PB),
                    MODELE=modele,
                    CHAM_MATER=ch_mater,
                    EXCIT=(_F(CHARGE=encastre),
                           _F(CHARGE=pression)),
                    OPTION=('SIEF_ELGA_DEPL', 'SIGM_ELNO_DEPL'))
#
# 8.3. Les derivees des contraintes aux noeuds
#
resultat=CALC_NO( reuse =resultat,
                  RESULTAT=resultat,
                  SENSIBILITE=(E1,E2,E3),
                  EXCIT=(_F(CHARGE=encastre),
                         _F(CHARGE=pression)),
                  OPTION='SIGM_NOEU_DEPL')
#
```

```
# 9. Impressions des resultats
#
# 9.1. Le resultat standard
#
DEFUFI(IMPRESSION=_F(NOM='RESUGMSH', UNITE=37))
#
IMPR_RESU(RESU=_F(FORMAT='GMSH', RESULTAT=resultat,
                  FICHIER='RESUGMSH'))
#
IMPR_RESU(RESU=_F(FORMAT='MED',
                  RESULTAT=resultat))
#
# 9.2. Le resultat des derivees
#
IMPR_RESU(RESU=_F(FORMAT='MED',
                  RESULTAT=resultat,
                  SENSIBILITE=(E1, E2, E3, PA, PB)))
#
# 9.3. Les valeurs extremes du deplacement et des contraintes dans chaque
zone
#
IMPR_RESU(RESU=_F(RESULTAT=resultat,
                  NOM_CHAM=('DEPL', 'SIEF_ELGA_DEPL'),
                  GROUP_MA='ZONE_1',
                  VALE_MAX='OUI',
                  VALE_MIN='OUI',
                  FORMAT_R='1PE12.5'))
#
IMPR_RESU(RESU=_F(RESULTAT=resultat,
                  NOM_CHAM=('DEPL', 'SIEF_ELGA_DEPL'),
                  GROUP_MA='ZONE_2',
                  VALE_MAX='OUI',
                  VALE_MIN='OUI',
                  FORMAT_R='1PE12.5'))
#
IMPR_RESU(RESU=_F(RESULTAT=resultat,
                  NOM_CHAM=('DEPL', 'SIEF_ELGA_DEPL'),
                  GROUP_MA='ZONE_3',
                  VALE_MAX='OUI',
                  VALE_MIN='OUI',
                  FORMAT_R='1PE12.5'))
#
# 9.4. Les valeurs extremes des derivees des contraintes dans la zone 3
#
IMPR_RESU(RESU=_F(RESULTAT=resultat,
                  SENSIBILITE=(E1,E2,E3,PA,PB),
                  NOM_CHAM='SIEF_ELGA_DEPL',
                  GROUP_MA='ZONE_3',
                  VALE_MAX='OUI',
                  VALE_MIN='OUI',
                  FORMAT_R='1PE12.3'))
#
```

```
# 9.5. Test de non regression sur une composante de derivee de contrainte
#
TEST_RESU(RESU=_F(RESULTAT=resultat,
                  SENSIBILITE=E3,
                  NOM_CHAM='SIGM_NOEU_DEPL',NOM_CMP='SIXX',
                  NUME_ORDRE=1,GROUP_NO='COIN_BD',
                  VALE=3.160121E-5, CRITERE='RELATIF', PRECISION=1e-05,
                  REFERENCE='NON_REGRESSION'))
#
FIN( )
```

Il est temps d'aborder le résultat de notre concours du chapitre 2. Voici les valeurs extrêmes des dérivées des contraintes par rapport aux deux pressions P_A et P_B, dans la zone 3.

	Dérivée par rapport à P_A		**Dérivée par rapport à P_B**	
	Mini	**Maxi**	**Mini**	**Maxi**
σ_{xx}	–0,0068	0,0868	–2,537	–0,8063
σ_{yy}	–0,0107	0,0107	–0,4770	0,0256
σ_{zz}	–0,0046	0,0245	–0,7264	–0,2427
σ_{xy}	–0,0206	0,0050	–0,8057	0,0250

Nous constatons que le champ de contraintes est plus sensible à P_B qu'à P_A, le rapport maximal étant situé entre 30 et 50.

Pour la question 2 et la question subsidiaire, nous examinons dans la zone n° 3, les valeurs extrêmes des dérivées du champ de contraintes par rapport aux trois modules d'Young E_1, E_2 et E_3.

	Dérivée par rapport à E_1		**Dérivée par rapport à E_2**		**Dérivée par rapport à E_3**	
	Mini	**Maxi**	**Mini**	**Maxi**	**Mini**	**Maxi**
σ_{xx}	–0,0014	0,0127	–0,0023	0,0173	–0,0577	–0,0273
σ_{yy}	–0,0052	0,0043	–0,0083	0,0021	–0,0008	0,0161
σ_{zz}	–0,0009	0,0044	–0,0024	0,0049	–0,0157	–0,0043
σ_{xy}	–0,0028	0,0068	–0,0046	0,0048	–0,0182	0,0075

Sur ces résultats, nous constatons que les deux premiers modules d'Young ont grosso modo la même influence sur le champ de contraintes dans la zone n° 3, avec une légère prépondérance de E_2. Mais leur influence est surpassée par celle du troisième paramètre E_3. Si nous regardons les maximums de sensibilité en valeur absolue, nous avons les rapports suivants :

$$\left|\frac{\partial\sigma_{xx}}{\partial E_3}\right|_{\max} = 3{,}33\left|\frac{\partial\sigma_{xx}}{\partial E_2}\right|_{\max} = 4{,}54\left|\frac{\partial\sigma_{xx}}{\partial E_1}\right|_{\max}$$

$$\left|\frac{\partial\sigma_{yy}}{\partial E_3}\right|_{\max} = 1{,}94\left|\frac{\partial\sigma_{yy}}{\partial E_2}\right|_{\max} = 3{,}10\left|\frac{\partial\sigma_{yy}}{\partial E_1}\right|_{\max}$$

$$\left|\frac{\partial\sigma_{zz}}{\partial E_3}\right|_{\max} = 3{,}20\left|\frac{\partial\sigma_{zz}}{\partial E_2}\right|_{\max} = 3{,}57\left|\frac{\partial\sigma_{zz}}{\partial E_1}\right|_{\max}$$

$$\left|\frac{\partial\sigma_{xy}}{\partial E_3}\right|_{\max} = 3{,}79\left|\frac{\partial\sigma_{xy}}{\partial E_2}\right|_{\max} = 2{,}68\left|\frac{\partial\sigma_{xy}}{\partial E_1}\right|_{\max}$$

Félicitations aux lecteurs qui auront trouvé les bonnes solutions !

5 Commentaires généraux

5.1 Automatisation de l'analyse des commandes

Un utilisateur attentif qui consultera le fichier des messages produit par *Code_Aster* verra que, par rapport à ce qui avait été demandé, davantage de commandes ont été exécutées. C'est tout à fait normal. Le processus de calcul de sensibilité a besoin de dériver l'ensemble des commandes où interviennent les paramètres sensibles. Un pré-traitement du jeu de commandes va ainsi dupliquer chaque commande en remplaçant ses arguments par les arguments dérivés. De nouveaux concepts sont créés, dont les noms sont établis par un mécanisme automatique. Ils sont mémorisés en interne au calcul par la commande `MEMO_NOM_SENSI`. Leur connaissance n'a aucun intérêt pour l'utilisateur dans la mesure où toutes les informations sont accessibles par un couple (nom de concept standard, nom de paramètre sensible). En résumé, nous pouvons dire que le maximum a été fait pour simplifier la tâche de l'utilisateur.

Néanmoins, une réserve s'impose : ce mécanisme de pré-traitement n'est disponible que pour le traitement des commandes par lots. C'est d'ailleurs l'option par défaut de la commande `DEBUT`. Ainsi tout jeu de commandes produit par l'éditeur EFICAS en conservant le traitement par lot sera interprété correctement. Pour une utilisation avancée du jeu de commandes qui entraîne l'inactivation du traitement par lots, l'insertion automatique des commandes dérivées n'a pas lieu. C'est ce qui se passe quand on modifie à la main le jeu de commandes pour y insérer des instructions Python de base. Il faut alors faire à la main le travail de dérivation des commandes, les unes après les autres, en mémorisant les noms des concepts produits.

5.2 Performance

Le calcul d'une dérivée est toujours plus rapide que le calcul de la grandeur minimale.