

Manuel d'Utilisation
Fascicule U4.7- : Opérations sur les résultats et les champs
Document : U4.73.02

Macro-commande `MACR_RECAL`

1 But

Recaler des résultats de calculs sur des résultats expérimentaux ou sur d'autres résultats de calculs.

Considérons d'une part un ou plusieurs résultats d'essais et d'autre part un ou plusieurs calculs Aster modélisant ces essais. `MACR_RECAL` permet de déterminer les paramètres de ces calculs (qui peuvent être des paramètres de loi de comportement, de chargement, etc ...) décrivant au mieux les essais.

Pour plus de précisions sur l'algorithmie mise en oeuvre, se reporter à [R4.03.06].

2 Syntaxe

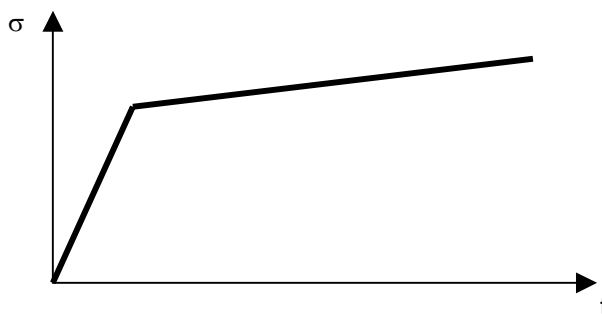
```
lr = MACR_RECAL                                [listr8]
(
  ♦ UNITE_ESCL = uni                            [I]
  ♦ RESU_EXP   = resu_exp                       [assd]
  ◇ POIDS      = poids                         [assd]
  ♦ RESU_CALC  = resu_calc                     [assd]
  ♦ LIST_PARA  = list_para                    [assd]
  ◇ UNITE_RESU = /91                          [défaut]
                                     /uni_r
  ◇ ITER_MAXI  = /10                          [défaut]
                                     /it
  ◇ RESI_GLOB_RELA = /1.E-3                  [défaut]
                                     / resi
  ◇ PARA_DIFF_FINI = /1.E-3                  [défaut]
                                     / coef
  ◇ GRAPHIQUE
    ◇ UNITE      = / 90                      [défaut]
                                     / uni_g
    ◇ INTERACTIF = / 'NON'                  [défaut]
                                     / 'OUI'
);
```

3 Présentation générale

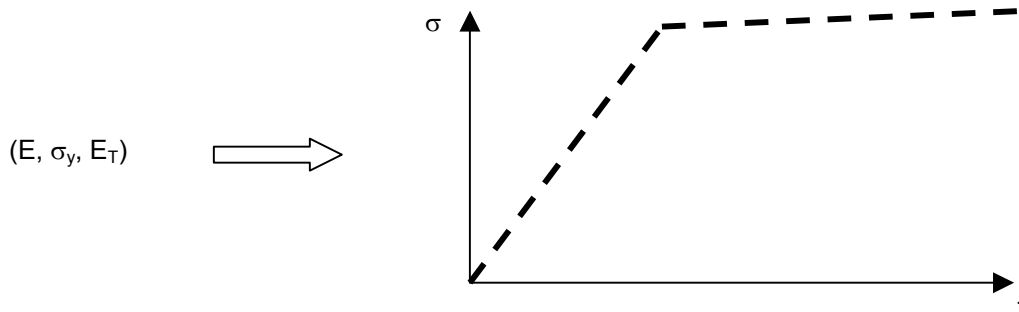
3.1 Principe du recalage

Considérons le problème modèle d'identification des caractéristiques élastoplastiques E , σ_y , E_T (respectivement module d'Young, limite d'élasticité et module d'écrouissage) d'un matériau sur un essai de traction uniaxiale.

On a d'une part la courbe de traction expérimentale donnant l'évolution de la contrainte en fonction du temps et qui est une donnée :



On a d'autre part une **fonction** des 3 paramètres qui pour chaque valeur du triplet E , σ_y , E_T renvoie une courbe de traction calculée :



L'objectif du recalage est alors de répondre à la question :

Quelles sont les valeurs de (E, σ_y, E_T) décrivant au mieux mon expérience ?

3.2 Organisation du recalage

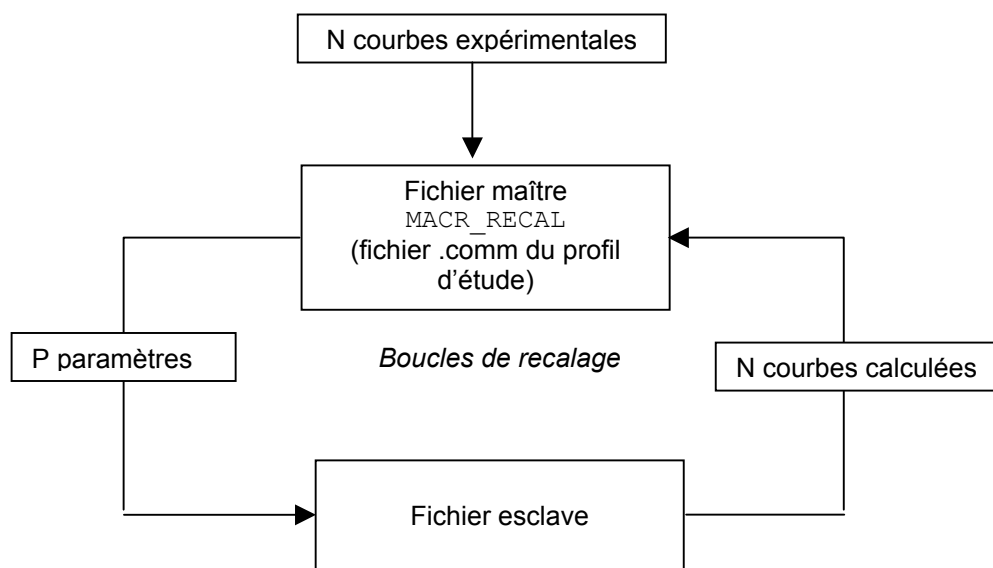
Pour mener à bien un recalage, il est nécessaire de disposer de l'ensemble des informations suivantes :

- les N courbes expérimentales (à chacune de ces courbes peut être attribué un poids arbitraire),
- les P paramètres à recaler ainsi que, pour chacun, une estimation de sa valeur initiale, sa valeur minimale et sa valeur maximale,
- le fichier de commandes modélisant les N essais que l'on veut recaler,
- les noms des N grandeurs à extraire du fichier de commandes ci-dessus et qui seront recalées sur les N courbes expérimentales. Ces grandeurs doivent être contenues dans une table issue de `POST_RELEVÉ_T`.

La mise en données de ces informations nécessite alors l'organisation suivante :

- un fichier de commandes dit **maître** contenant les N courbes expérimentales, les P paramètres, les noms des grandeurs à recalcr ainsi que d'autres informations propres au recalage, le tout renseigné dans `MACR_RECAL`. Les différents formats utilisés sont précisés dans ce qui suit,
- un fichier de commandes dit **esclave** modélisant les essais expérimentaux.

En effet, le recalage est un processus **itératif** : le fichier maître exécute le fichier esclave, il récupère les N courbes calculées avec les valeurs courantes des P paramètres, il compare les valeurs des courbes calculées à celles des courbes expérimentales, il en déduit de nouvelles valeurs pour les P paramètres et relance le fichier esclave. Ce processus continue jusqu'à obtention de la convergence.



Dans la partie suivante sont décrites les opérandes de `MACR_RECAL`. On y fait référence à quelques notions du langage Python. Il n'est cependant nullement nécessaire de connaître Python pour utiliser cette macro commande. La partie « Exemple d'utilisation » est là pour éclairer l'utilisateur.

La structure de données produite est une liste de réels contenant les valeurs des paramètres à convergence en cas de convergence ou à la dernière itération dans le cas contraire.

4 Opérandes

4.1 Opérande **UNITE_ESCL**

◆ **UNITE_ESCL**

Numéro d'unité logique du fichier esclave, attribué dans l'interface ASTERIX (colonne UL). L'extension de ce fichier peut être quelconque.

4.2 Opérande **RESU_EXP**

◆ **RESU_EXP**

Nom de la liste Python de N tableaux Numeric Python contenant les N courbes expérimentales. La liste est définie préalablement sous la forme :

```
resu_exp=[ Numeric.array([ [x0,y0],  
                           [x1,y1],  
                           ...  
                           [xn,yn] ]),  
          .....  
          Numeric.array([ [u0,v0],  
                           [u1,v1],  
                           ...  
                           [un,vn] ])  
        ]
```

4.3 Opérande **POIDS**

◆ **POIDS**

Nom du tableau Numeric Python contenant les N poids à affecter aux N courbes expérimentales. Si non renseigné, alors la table est composée de 1. La liste est définie préalablement sous la forme :

```
POIDS= Numeric.array(p0,p1,...)
```

4.4 Opérande **RESU_CALC**

◆ **RESU_CALC**

Nom de la liste Python de N listes Python contenant la définition des réponses numériques correspondant aux mesures expérimentales sur lesquelles on va effectuer le recalage.

4.5 Opérande **LIST_PARA**

◆ **LIST_PARA**

Nom de la liste Python de P listes Python contenant les noms des variables, leurs valeurs initiales, leurs valeurs minimales et leurs valeurs maximales. Cette liste est définie préalablement sous la forme :

```
List_para=[ [ 'PARA1__',INI_1,MIN_1,MAX_1 ],  
            [ 'PARA2__',INI_2,MIN_2,MAX_2 ],  
            ...  
            [ 'PARAP__',INI_P,MIN_P,MAX_P ] ]
```

Attention :

On demande que les noms des variables se terminent par deux blancs soulignés (par exemple : *YOUN__*).

4.6 Opérande ITER_MAXI

◇ ITER_MAXI

Nombre d'itérations maximales de recalage.

4.7 Opérande RESI_GLOB_RELA

◇ RESI_GLOB_RELA

Résidu global relatif du recalage.

Cette valeur est disjointe de celle renseignée pour les solveurs non linéaires STAT_NON_LINE et DYNA_NON_LINE.

4.8 Opérande UNITE_RESU

◇ UNITE_RESU

Numéro d'unité logique du fichier de résultat du recalage (évolution des paramètres au cours des itérations, critères de convergence).

4.9 Opérande COEF_DIFF_FINIES

◇ PARA_DIFF_FINI

Le recalage nécessite le calcul des dérivées des réponses par rapport aux paramètres. Ce calcul est réalisé par différences finies. COEF_DIFF_FINIES correspond à α dans la formule suivante :

$$\frac{\partial f}{\partial x} \approx \frac{f(x + \alpha x) - f(x)}{\alpha x}$$

4.10 Opérande GRAPHIQUE

◇ UNITE

Numéro d'unité logique des graphiques produits au cours du recalage. A chaque itération, MACR_RECAL produit N graphiques POSTSCRIPT représentant les N courbes expérimentales et calculées.

◇ INTERACTIF

Affichage des graphiques de manière interactive.

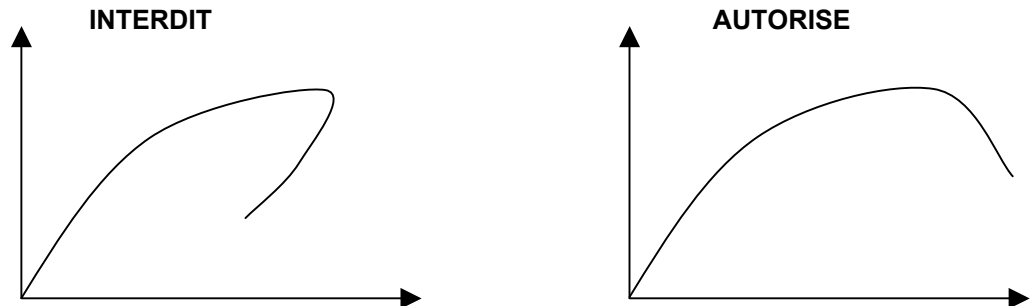
Attention :

| Ceci n'est possible que lorsque le recalage tourne en interactif et non en batch.

5 Précautions d'emploi

Voici un ensemble de conseils **indispensables** à la bonne utilisation du recalage.

- Les courbes expérimentales sont définies comme des tableaux à deux colonnes : une pour les abscisses et une pour les ordonnées.
- Les courbes expérimentales doivent être des fonctions : à une abscisse ne doit correspondre qu'une ordonnée.



- On doit recalcr N courbes calculées sur N courbes expérimentales.
- La première courbe calculée sera recalée sur la première courbe expérimentale, la deuxième courbe calculée sera recalée sur la deuxième courbe expérimentale, et ainsi de suite dans l'ordre renseigné pour les opérandes `RESU_EXP` et `RESU_CALC`.
- Les grandeurs calculées renseignées sous l'opérande `RESU_CALC` doivent être issues de `POST_RELEVE_T`.
- Les paramètres du recalage doivent être déclarés en bloc au début du fichier de commandes esclave. Par exemple :

```
DEBUT ( ) ;  
DSDE__ = 200. ;  
YOUN__ = 8.E4 ;  
SIGY__ = 10. ;  
.....
```
- Les valeurs initiales des paramètres du recalage sont celles renseignées pour l'opérande `LIST_PARA` et non celles présentes dans le fichier esclave de l'utilisateur.
- A chaque itération de recalage, les calculs définis dans le fichier esclave doivent converger. Dans le cadre de recalage de calculs non linéaires, il est donc fortement recommandé d'utiliser la découpe automatique du pas de temps.
- Dans le cadre de recalage de calculs non linéaires avec découpe automatique du pas de temps, il est **indispensable** de définir une liste d'archivage sous l'opérande `LIST_ARCH`.
- Le recalage est un moyen puissant d'obtenir des valeurs de paramètres à partir d'essais. Il n'est cependant pas miraculeux : les courbes **expérimentales** doivent contenir suffisamment d'informations pour identifier les paramètres. Il est par exemple impossible d'identifier des paramètres élastoplastiques avec un essai restant dans le domaine élastique. Les essais expérimentaux doivent donc exciter les paramètres à identifier.
- Dans le même logique, il est souhaitable que les courbes **expérimentales** contiennent des points en nombre suffisant pour bien décrire l'action des paramètres à identifier.
- Enfin, dans le cas de l'utilisation de plusieurs courbes expérimentales, le fait qu'elles aient le même nombre de points équilibre l'information qu'elles apportent.

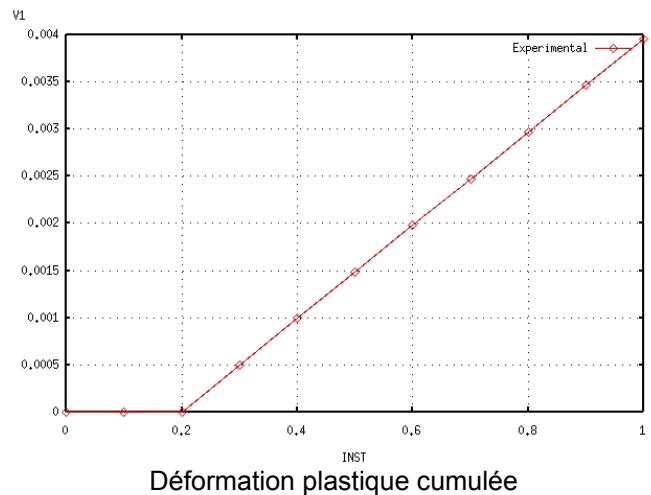
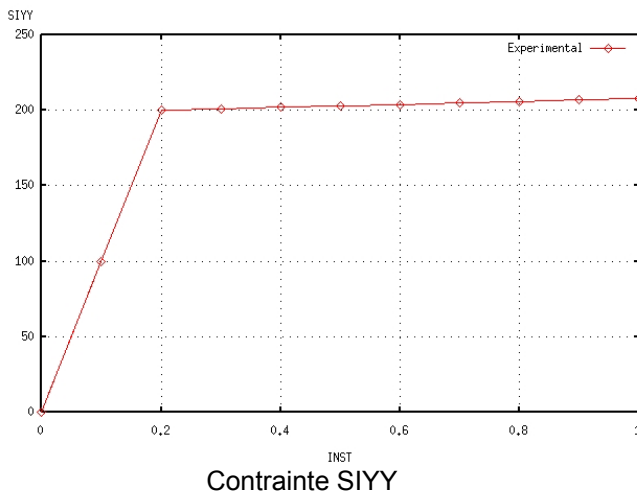
6 Exemple d'utilisation

6.1 Identification des paramètres d'une loi de comportement élastoplastique sur un essai de traction

Cet exemple est traité par le test ZZZZ159A [V1.01.159].

6.1.1 Position du problème

On dispose des résultats d'un essai de traction. Il s'agit de l'évolution de la contrainte SIYY au cours du temps ainsi que de l'évolution de la déformation plastique cumulée au cours du temps.



On désire recalculer sur ces essais le module d'Young, la limite d'élasticité et la pente d'écrouissage d'une loi de comportement élastoplastique à écrouissage isotrope linéaire.

6.1.2 Mise en données

6.1.2.1 Expérience

On commence tout d'abord par définir nos résultats d'essais. Ils consistent en deux courbes que l'on définit comme suit.

```
experience=[ Numeric.array([ [0.00000E+00 , 0.00000E+00 ],
                             [5.00000E-02 , 5.00000E+01 ],
                             .....
                             [9.50000E-01 , 2.07500E+02 ],
                             [1.00000E+00 , 2.08000E+02 ] ]),
              Numeric.array([ [0.00000E+00 , 0.00000E+00 ],
                             [5.00000E-02 , 0.00000E+00 ],
                             .....
                             [9.50000E-01 , 3.71250E-03 ],
                             [1.00000E+00 , 3.96000E-03 ] ] ) ]
```

`experience` est donc le nom d'une liste Python (définie entre crochets) de 2 tableaux Numeric Python.

6.1.2.2 Calcul

On écrit ensuite le fichier de commandes Aster esclave modélisant cet essai de traction où vont apparaître nos 3 paramètres ainsi que les deux courbes à recaler.

```
DEBUT();
# AFFECTATION DES VALEURS DES PARAMETRES A RECALER
# LES VALEURS RENSEIGNEES ICI SONT SANS IMPORTANCE
# SEULES COMPTENT LES VALEURS RENSEIGNEES DANS LE FICHIER MAITRE
DSDE__ = 200.;

YOUN__ = 8.E4;

SIGY__ = 1.;

.....

ACIER=DEFI_MATERIAU(ECRO_LINE=_F(D_SIGM_EPSI=DSDE__,
                                SY=SIGY__,),
                    ELAS=_F(NU=0.3,
                           E=YOUN__,),);

.....

EVOL=STAT_NON_LINE(CHAM MATER=CHMAT,
                   MODELE=MO,
                   ARCHIVAGE=_F(LIST_INST=INSTANTS,
                                ARCH_ETAT_INIT='OUI',),
                   CONVERGENCE=_F(ITER_GLOB_MAXI=10,
                                  RESI_GLOB_RELA=1.E-05,),
                   COMP_INCR=_F(RELATION='VMIS_ISOT_LINE',),
                   INCREMENT=_F(LIST_INST=INSTANTS,
                                SUBD_PAS=4,
                                COEF_SUBD_PAS_1=1.0,
                                SUBD_PAS_MINI=1.E-05,),
                   NEWTON=_F(REAC_ITER=1,
                             REAC_INCR=1,),
                   EXCIT=_F(CHARGE=TRACTION,
                           FONC_MULT=RAMPE,),);

.....

# EXTRACTION DE LA REPONSE SIGMAYY(T)
REPOSE1=POST_RELEVE_T(ACTION=_F(OPERATION='EXTRACTION',
                                INTITULE='SIGYY',
                                RESULTAT =EVOL,
                                NOM_CHAM = 'SIEF_ELNO_ELGA',
                                NOM_CMP  = 'SIYY',
                                GROUP_NO  = 'A',),);

# EXTRACTION DE LA REPONSE EPSP(T)
REPOSE2=POST_RELEVE_T(ACTION=_F(OPERATION='EXTRACTION',
                                INTITULE='V1',
                                RESULTAT =EVOL,
                                NOM_CHAM = 'VARI_ELNO_ELGA',
                                NOM_CMP  = 'V1',
                                GROUP_NO  = 'A',),);

FIN();
```

Il nous faut maintenant définir dans le fichier maître les valeurs initiales et les plages de variations de nos paramètres. On désire :

```
1.E5 < Module d'Young initial = 1.E5 < 5.E5
5. < Limite d'élasticité initiale = 30. < 500.
1.E3 < Module d'écrouissage initial = 1.E3 < 1.E4
```

Ce que l'on écrit sous la forme=

```
Parametres
=[['YOUNG__',100000.,50000.,500000.],['DSDE__',1000.,500.,10000.],
 ['SIGY__',30.,5.,500.]]
```

Enfin il nous reste à définir dans le fichier maître les grandeurs à extraire du fichier de commandes esclave ci-dessus. Nous désirons d'une part extraire la colonne INST et la colonne SIYY de la table REPONSE1 et d'autre part la colonne INST et la colonne V1 de la table REPONSE2. Nous l'écrivons :

```
calcul = [['REPONSE1','INST','SIYY'], ['REPONSE2','INST','V1']]
```

6.1.2.3 MACR_RECAL

Nous renseignons maintenant ces informations dans le corps de MACR_RECAL :

```
RESU=MACR_RECAL(
    UNITE_ESCL =3,
    RESU_EXP =experience,
    LIST_PARA =parametres,
    RESU_CALC =calcul,);
```

6.1.2.4 ASTERIX

On définit enfin le profil d'étude suivant :

DREC	Type	Fichier (à partir de \$HOME)	User	Machine	UL
D E	comm	Anomalies/Restit6/Test/zzzz159a.comm	*****	*****	1
D E	comm	Anomalies/Restit6/Test/zzzz159a.3	*****	*****	3
D E	mail	Anomalies/Restit6/Test/zzzz159a.mail	*****	*****	20
RE	resu	Anomalies/Restit6/Test/zzzz159a.resu	*****	*****	8
RE	mess	Anomalies/Restit6/Test/zzzz159a.mess	*****	*****	6
RE	ps	Anomalies/Restit6/Test/zzzz159a.ps	*****	*****	90
RE	libr	Anomalies/Restit6/Test/zzzz159_opti.resu	*****	*****	91

6.1.3 Résultats

Une fois l'étude réalisée, le fichier de résultat du recalage ZZZZ159_opti.resu contient les informations suivantes :

```
Calcul de la sensibilité par rapport à = YOUN__ DSDE__ SIGY__  
=====
```

```
Iteration 0 =
```

```
=> Fonctionnelle = 1.0  
=> Résidu        = 1.0  
=> Paramètres    =  
    YOUN__ = 100000.0  
    DSDE__ = 1000.0  
    SIGY__ = 30.0  
=====
```

```
Calcul de la sensibilité par rapport à = YOUN__ DSDE__ SIGY__  
=====
```

```
Iteration 1 =
```

```
=> Fonctionnelle = 0.259742161795  
=> Résidu        = 0.30865397471  
=> Paramètres    =  
    YOUN__ = 300857.888503  
    DSDE__ = 9135.12770111  
    SIGY__ = 152.548047532  
=====
```

```
Calcul de la sensibilité par rapport à = YOUN__ DSDE__ SIGY__  
=====
```

```
Iteration 2 =
```

```
=> Fonctionnelle = 0.0757636994765  
=> Résidu        = 0.473053125246  
=> Paramètres    =  
    YOUN__ = 157723.378846  
    DSDE__ = 2022.7431335  
    SIGY__ = 213.155325073  
=====
```

```
Calcul de la sensibilité par rapport à = YOUN__ DSDE__ SIGY__  
=====
```

```
Iteration 3 =
```

```
=> Fonctionnelle = 0.00190706595529  
=> Résidu        = 0.0520849911718  
=> Paramètres    =  
    YOUN__ = 192302.166747  
    DSDE__ = 895.845518907  
    SIGY__ = 203.753909707  
=====
```

```
Calcul de la sensibilité par rapport à = YOUN__ DSDE__ SIGY__  
=====
```

```
Iteration 4 =
```

```
=> Fonctionnelle = 2.70165453323e-06  
=> Résidu        = 0.00172172540305  
=> Paramètres    =
```

Titre : Macro-commande MACR_RECAL
Auteur(s) : N. TARDIEU, O. NICOLAS

Date : 17/06/04
Clé : U4.73.02-B1 Page : 12/12

YOUN__ = 199801.572817
DSDE__ = 1928.08902726
SIGY__ = 200.274590793

Calcul de la sensibilité par rapport à = YOUN__ DSDE__ SIGY__

Iteration 5 =

=> Fonctionnelle = 2.65431115925e-12
=> Résidu = 1.83121468206e-06
=> Paramètres =
YOUN__ = 199999.975047
DSDE__ = 1999.86955101
SIGY__ = 200.000462987

CONVERGENCE ATTEINTE

Hessien adimensionnel de la fonctionnelle cout=
[[6.91329664e+00 -3.37051575e-02 -1.29686389e+00]
[-3.37051575e-02 2.60794725e-03 3.25552912e-02]
[-1.29686389e+00 3.25552912e-02 6.22343397e-01]]

Valeurs propres du Hessien=
[7.17037083e+00 6.25201433e-04 3.67251955e-01]

Vecteurs propres associés=
[[-0.98092602 0.00549514 0.19430377]
[0.00810935 -0.99757318 0.06915192]
[-0.19421223 -0.0694086 -0.97850092]]

Et le fichier POSTSCRIPT ZZZZ159.ps contient :

