

Procédure IMPR_RESU au format 'MED'

1 But

Écrire le résultat d'un calcul dans un fichier au format MED. On décrit l'ensemble des mots clés de la commande `IMPR_RESU` [U4.91.01] pour ce qui concerne ce format de sortie uniquement.

On peut écrire au choix dans un fichier au format MED :

- un maillage,
- des champs aux nœuds,
- des champs aux éléments.

Lors de l'écriture des champs par éléments aux points de Gauss, on écrit également la localisation des éléments de référence (coordonnées et poids des points de Gauss).

MED (Modélisation et Échanges de Données) est un format de données neutre développé par EDF R&D et le CEA pour les échanges de données entre codes de calcul. Les données que l'on peut échanger selon ce format sont les maillages et les champs de résultats aux nœuds et par éléments. Les fichiers MED sont des fichiers binaires et portables (s'appuyant sur la bibliothèque HDF5, Hierarchical Data Format). L'écriture de résultats dans un fichier MED permet à tout autre code de calcul interfacé avec MED, de lire les résultats produits par *Code_Aster* via la commande `IMPR_RESU`.

2 Syntaxe

```
IMPR_RESU      (
#  Syntaxe de la procédure IMPR_RESU au format 'MED'
◇  UNITE = unit,
◇  FORMAT = / 'MED',                                [DEFAULT]

◇  PROC0 = / 'OUI',                                [DEFAULT]
           / 'NON',
◇  IMPR_NOM_VARI = / 'OUI',                          [DEFAULT]
                  / 'NON',
◇  INFO = / 1,                                     [DEFAULT]
          / 2,

/  RESU = _F(
  ◆ |  MAILLAGE = ma,                                [maillage]
    |  /  RESULTAT = resu,                           [sd_resultat]

        ◇ /  NOM_CHAM = l_nomsymb,                    [l_K16]

        ◇ /  NUME_ORDRE = lordre,                    [l_I]
          /  NUME_MODE = lmode,                       [l_I]
          /  NOEUD_CMP = lnoecmp,                    [l_K16]
          /  NOM_CAS = ncas,                          [l_K16]
          /  ANGLE = langl,                           [l_K16]
          / /  FREQ = lfreq,                          [l_R]
          /  INST = linst,                            [l_R]
        ◇ |  PRECISION = / prec,                      [R]
          / 1.0D-3,                                   [DEFAULT]
          |  CRITERE = / 'RELATIF',                  [DEFAULT]
          / 'ABSOLU',

        /  CHAM_GD = chgd,                            [cham_gd]

        ◇ /  NOM_CHAM_MED = l_nomcham,                [l_K64]
          /  NOM_RESU_MED = l_nomresu,                [K8]

        ◇  CARA_ELEM = carele,                        [cara_elem]

        ◇  PARTIE = / 'REEL',
                  / 'IMAG',
◇  INFO_MAILLAGE = / 'OUI'
                  / 'NON'                            [DEFAULT]
),

# Pour imprimer quelques champs de « données » :
/  CONCEPT = _F(
  /  CHAM_MATER = chmat,                              [cham_mater]
  /  CARA_ELEM = carele,                              [cara_elem]
  ◇  REPERE_LOCAL = / 'NON',                          [DEFAULT]
                  / 'ELEM',
                  / 'ELNO',

  # si REPERE_LOCAL = 'ELEM' ou 'ELNO',
  ◆  MODELE = mo                                     [modele]
/  CHARGE = charg,                                   [charge]
)
```

3 Opérandes `FORMAT` et `UNITE`, `PROC0` et `INFO`

3.1 Opérande `FORMAT`

L'opérande `FORMAT` permet de spécifier le format du fichier où écrire le résultat.

Le format 'MED' signifie à la procédure `IMPR_RESU` que le résultat doit s'écrire dans un fichier au format MED. C'est le format d'écriture par défaut.

3.2 Opérande `UNITE`

Définit dans quelle unité on écrit le fichier med. Par défaut, `UNITE` = 80 et correspond à l'unité par défaut du type `rmed` dans `astk`.

3.3 Opérande `PROC0`

L'opérande `PROC0` dont la valeur par défaut est 'OUI', permet de restreindre l'impression sur le processeur de rang 0. Si on lui affecte la valeur 'NON', les impressions seront effectuées sur tous les processeurs.

3.4 Opérande `INFO`

Le mot-clé `INFO` lorsqu'il est égal à 2 permet d'obtenir des informations sur les impressions réalisées par la commande.

4 Opérande `IMPR_NOM_VARI`

Ce mot clé est utile dans le cas des variables internes. Lorsqu'il est utilisé et que l'impression d'un champ `VARI_*` a été demandée, c'est en fait un champ `VARI_*_NOMME` qui sera imprimé. Ce champ aura des composantes dont le nom sera basé sur le catalogue des lois de comportement utilisées dans le calcul. Si deux lois de comportement ont des variables internes communes, celles-ci seront fusionnées dans une unique composante.

5 Mot-clé facteur `RESU`

Ce mot clé facteur permet de spécifier les résultats et les champs à imprimer.

5.1 Opérande `MAILLAGE`

Si le résultat est un maillage (opérande `maillage` [U4.91.01]), les données reportées dans le fichier résultat au format MED sont :

- | | |
|-----------------------------------|---|
| • la liste des nœuds | numéro, nom, coordonnées, |
| • la liste des mailles | numéro, nom, type, nom des nœuds, |
| • la liste des groupes de nœuds | numéro, nom, nombre de nœuds, noms des nœuds, |
| • la liste des groupes de mailles | numéro, nom, nombre de mailles, noms des mailles. |

Nota :

Dans un fichier MED, il y a partition des nœuds et des mailles en fonction des groupes. Une partition correspond à une famille MED. Dans un fichier MED, les groupes sont répartis au sein des familles : on y trouve donc des familles de nœuds et des familles d'éléments.

5.2 Opérande `RESULTAT`

L'opérande `RESULTAT` permet d'imprimer dans un fichier MED, des champs contenus dans un concept `resultat`.

On écrit dans le fichier 'MESSAGE' les informations suivantes :

- opérande 'RESULTAT',
- opérande 'NOM_CHAM',
- opérande 'NUME_ORDRE',
- nom du champ stocké dans le fichier MED : concaténation des trois précédentes opérandes.

Si `INFO_MALLAGE = 'OUI'`, des informations plus détaillées sont imprimées dans le fichier 'MESSAGE' lors de l'écriture du maillage MED. On pourra par exemple obtenir les types de mailles imprimées, les noms des familles MED qui sont créées, etc.

5.3 Opérande CARA_ELEM

L'opérande `CARA_ELEM` sert pour l'impression des champs à sous-points. Lorsque `CARA_ELEM` est fourni, les champs à sous-points sont imprimés en ajoutant des informations dans le fichier MED permettant de positionner les sous-points en tenant compte des informations contenues dans la `sd_cara_elem` (épaisseur d'une coque, angle de vrille d'une poutre multi-fibre, ...).

Remarque :

Il n'est actuellement pas possible de visualiser des champs aux sous-points sur des éléments TUYAU dans le module ParaViS de Salome-Meca. On pourra donc pour l'instant utiliser l'opérateur IMPR_RESU_SP [U7.05.41] pour ce faire.

5.4 Opérande CHAM_GD

L'opérande `CHAM_GD` permet d'imprimer dans le fichier une structure de donnée de type `cham_gd`. Concrètement, on peut donc imprimer avec ce mot-clé une carte, un champ par éléments ou un champ aux nœuds.

5.5 Opérande NOM_CHAM_MED

L'opérande `NOM_CHAM_MED` permet de définir le nom du champ MED. C'est une chaîne de 64 caractères. Ceci peut servir en particulier lorsque l'on souhaite imprimer certaines composantes du champ comme plusieurs champs dans le même fichier MED (par exemple pour la visualisation de `SIRO_ELEM`).

5.6 Opérande NOM_RESU_MED

L'opérande `NOM_RESU_MED` est une alternative à `NOM_CHAM_MED` concernant la terminologie des champs MED. Son utilisation va permettre de ne plus nommer explicitement les champs MED, ce qui signifie que tous les champs contenus dans le résultat seront imprimés. Chaque nom de champs MED sera construit à partir :

- de la chaîne de caractères fournie à `NOM_RESU_MED` (chaîne d'au plus 8 caractères),
- du nom symbolique du champ Aster.

Par exemple :

```
IMPR_RESU = (  
  FORMAT = 'MED',  
  RESU = _F ( RESULTAT = U,  
             NOM_RESU_MED = 'U_HAUT',  
             GROUP_MA = 'HAUT',  
             NUME_ORDRE = 1, )  
)
```

Si le résultat `U` contient les champs `DEPL` et `SIEF_ELGA`, alors la commande ci-dessus produira les champs MED :

- 'U_HAUT_DEPL',

- 'U_HAUT__SIEF_ELGA' ,

Ceci peut servir en particulier lorsque l'on souhaite imprimer dans le même fichier MED le même champ sur des groupes de mailles différents.

5.7 Opérande PARTIE

Il n'est pas possible d'écrire des champs complexes. C'est pourquoi il faut choisir entre la partie réelle (PARTIE='REEL') et la partie complexe (PARTIE='IMAG').

5.8 Opérandes NOM_CHAM / NUME_ORDRE / NUME_MODE / NOEUD_CMP / NOM_CAS / ANGLE / FREQ / INST / PRECISION / CRITERE / FICHER

Cf. document [U4.91.01].

6 Mot-clé facteur CONCEPT

Ce mot clé facteur permet d'imprimer sous une forme visualisable les quantités affectées par l'utilisateur avec les commandes AFFE_MATERIAU , AFFE_CARA_ELEM et AFFE_CHAR_MECA . Cela permet de vérifier que les règles de surcharge du code conduisent aux affectations souhaitées.

Par exemple :

```
IMPR_RESU( FORMAT='MED' ,  
           CONCEPT=(  
             _F(CHAM_MATER = CHAMPMAT) ,  
             _F(CARA_ELEM  = CARA_ELE) ,  
             _F(CHARGE     = CHARG1) ,  
           ))
```

Les champs contenus dans ces structures de données sont imprimées sous deux formes très différentes :

- La forme "A" (la plus facile à interpréter graphiquement) : chaque composante est imprimée séparément comme un nombre réel. Par exemple, l'épaisseur des éléments de coque est visualisable comme un champ scalaire (scalar map dans Salomé).
- La forme "B" : on affecte à chaque maille un code entier : 1, 2, 3, ...n. Les mailles affectées par le même code ont alors TOUTES leurs composantes identiques. La "définition" des codes, c'est à dire les valeurs de ces composantes est imprimée dans le fichier message. On peut visualiser les "codes" comme un champ scalaire, ce qui permet de "voir" les zones où "tout est constant".

La forme "A" est utilisée systématiquement pour le format "MED", sauf pour le champ de matériaux car ce champ contient le nom du matériau affecté sur les mailles et ce nom n'est pas un nombre. La forme "A" n'est pas programmée au format "RESULTAT".

La forme "B" est systématiquement utilisée pour le format "RESULTAT" et pour le champ de matériaux.

Pour la forme "B", la correspondance entre les quantités affectées et les code est donnée dans le fichier .mess.

Par exemple, pour le champ de matériaux :

```
IMPRESSION D'UN CHAMP DE CONCEPT : Champ de MATERIAUX  
NOM DU CHAMP : CHAMPMAT_CHAMP_MAT  
CORRESPONDANCE VALEUR <-> CONTENU :  
VALEUR = 1.  
X1      = MAT_1  
VALEUR = 2.
```

X1	=	MAT_2
VALEUR =		3.
X1	=	MAT_3

6.1.1 Opérande REPERE_LOCAL

Si REPERE_LOCAL a la valeur 'ELEM', les 3 vecteurs constituant le repère local de chaque élément sont imprimés, pour pouvoir être visualisés à l'aide de ParaVis.

Si REPERE_LOCAL a la valeur 'ELNO', les 3 vecteurs constituant le repère local de chaque élément sont enregistrés sous forme de champ aux nœuds, pour pouvoir être ensuite utilisés dans la calculatrice de ParaVis en combinaison avec d'autres champs aux nœuds. Dans ce cas aucune autre information venant du concept CARA_ELEM n'est enregistré dans le fichier.

Lorsque REPERE_LOCAL='ELNO' il est donc possible de combiner les vecteurs locaux avec les composantes des champs d'efforts internes. Cela permet de visualiser les vecteurs efforts en 3D, ainsi que de réaliser une animation de leur évolution au cours des instants de calcul. Pour réaliser cette action plusieurs opérations élémentaires sont à réaliser dans ParaVis :

- Ouvrir le fichier MED contenant les repères :
sur les REPE -> Filtre "ELNO Points" -> Filtre "Merge blocs"
- Ouvrir le fichier MED contenant les champs :
sur les EFGE_ELNO -> Filtre "ELNO Points" -> Filtre "Merge blocs"
- sélection des 2 "Merge blocs" puis "Append Attributes"

Dans "Calculator" on a accès aux vecteurs des REPE et aux composantes des champs.

Le calcul du vecteur effort (N, Vy, Vz) est fait de la façon suivante :

$$Fint = CAREL_REPLC_1 * xxxxxxxx EFGE_ELNO_N + \\ CAREL_REPLC_2 * xxxxxxxx EFGE_ELNO_VY + \\ CAREL_REPLC_3 * xxxxxxxx EFGE_ELNO_VZ$$

où 'xxxxxxx' désigne le nom du concept résultat, produit par Code_Aster.

Le calcul du vecteur Moment (Mt, My, Mz) est fait de la façon suivante :

$$Mint = CAREL_REPLC_1 * xxxxxxxx EFGE_ELNO_MT + \\ CAREL_REPLC_2 * xxxxxxxx EFGE_ELNO_MFY + \\ CAREL_REPLC_3 * xxxxxxxx EFGE_ELNO_MFZ$$

où 'xxxxxxx' désigne le nom du concept résultat, produit par Code_Aster.

6.2 Champs pouvant être visualisés

CHAM_MATER :

Champ de matériaux

CARA_ELEM :

Caractéristiques générales des barres
Caractéristiques géom. des barres
Caractéristiques générales des poutres
Caractéristiques géom. des poutres
Caractéristiques des cables
Caractéristiques des poutres courbes
Caractéristiques des poutres "fluides"
Caractéristiques des éléments discrets K_*
Caractéristiques des éléments discrets M_*
Caractéristiques des éléments discrets A_*
Caractéristiques géom. des coques
Orientation des éléments 2D et 3D
Orientation des coques et des poutres

CHARGE :

- Chargement de PESANTEUR
- Chargement de ROTATION
- Chargement de PRES_REP
- Chargement de forces volumiques en 3D
- Chargement de forces surfaciques en 3D
- Chargement de forces linéiques en 3D
- Chargement de forces surfaciques en 2D
- Chargement de forces linéiques en 2D
- Chargement de forces réparties pour les coques
- Chargement de PRE_EPSI
- Chargement de FORCE_ELEC
- Chargement de FLUX_THM_REP
- Chargement d'IMPE_FACE
- Chargement d'ONDE_FLUI

7 Exemple

```
IMPR_RESU = (  
  FORMAT = 'MED' ,  
  RESU = _F ( RESULTAT = REMEZERO ,  
             NOM_CHAM = 'ERME_ELEM' ,  
             NUME_ORDRE = 3 , )  
)
```

L'exécution de la commande IMPR_RESU provoquera l'affichage suivant dans le fichier 'MESSAGE' :

```
RESULTAT          : REMEZERO  
CHAMP             : ERME_ELEM  
NUME_ORDRE        : 3  
==> NOM MED      : REMEZEROERME_ELEM
```

Exemple d'utilisation de NOM_CHAM_MED pour l'impression de SIRO_ELEM :

```
IMPR_RESU (FORMAT='MED' ,  
           RESU=(  
             _F (RESULTAT=RESUNL ,  
                 NOM_CHAM=('SIRO_ELEM' , ) ,  
                 NOM_CHAM_MED=('RESUNL_SIRO_ELEM_NORMAL' ) ,  
                 NOM_CMP=('SIG_NX' , 'SIG_NY' , 'SIG_NZ' , 'SIG_N' , ) ,  
                 GROUP_MA='PRES' , ) ,  
             _F (RESULTAT=RESUNL ,  
                 NOM_CHAM=('SIRO_ELEM' , ) ,  
                 NOM_CHAM_MED=('RESUNL_SIRO_ELEM_TANGENT' ) ,  
                 NOM_CMP=('SIG_TX' , 'SIG_TY' , 'SIG_TZ' , ) ,  
                 GROUP_MA='PRES' , ) ,  
             _F (RESULTAT=RESUNL ,  
                 NOM_CHAM=('SIRO_ELEM' , ) ,  
                 NOM_CHAM_MED=('RESUNL_SIRO_ELEM_T1' ) ,  
                 NOM_CMP=('SIG_T1X' , 'SIG_T1Y' , 'SIG_T1Z' , 'SIG_T1' , ) ,  
                 GROUP_MA='PRES' , ) ,  
             _F (RESULTAT=RESUNL ,  
                 NOM_CHAM=('SIRO_ELEM' , ) ,  
                 NOM_CHAM_MED=('RESUNL_SIRO_ELEM_T2' ) ,  
                 NOM_CMP=('SIG_T2X' , 'SIG_T2Y' , 'SIG_T2Z' , 'SIG_T2' , ) ,  
                 GROUP_MA='PRES' , ) ,  
             ) , ) ;
```