

**Manuel d'Utilisation**  
**Fascicule U4.5- : Méthode de résolution**  
**Document : U4.51.12**

# MACRO COMMANDE *SIMU\_POINT\_MAT*

---

## 1 But

---

Calculer l'évolution mécanique d'un point matériel, en quasi-statique non linéaire.

Tous les comportements disponibles dans *STAT\_NON\_LINE* [U4.51.03] le sont également ici.

Le but de cette macro-commande est de simplifier au maximum les données : il suffit de fournir :

- Le comportement et le matériau ;
- Les fonctions définissant l'évolution des composantes de contraintes ou de déformations choisies ;
- La discrétisation en temps.

Ceci permet en particulier de calculer l'évolutions du tenseur des contraintes dans le cas de déformations imposées, ou l'inverse (cas courants en identification de paramètres matériau)

Produit une structure de données de type `table` contenant, en fonction du temps, l'évolution de toutes les composantes des tenseurs de contraintes et de déformations, ainsi que les variables internes.

## Table des matières

1 But	
2 Syntaxe .....	3
3 Opérandes .....	4
3.1 Opérande <i>MATER</i> .....	4
3.2 Mots clés <i>SIGM_IMPOSE</i> / <i>EPSI_IMPOSE</i> .....	4
3.2.1 Opérandes <i>SIXX</i> , <i>SIYY</i> , <i>SIZZ</i> , <i>SIXY</i> , <i>SIXZ</i> , <i>SIYZ</i> .....	4
3.2.2 Opérandes <i>EPXX</i> , <i>EPYY</i> , <i>EPZZ</i> , <i>EPXY</i> , <i>EPXZ</i> , <i>EPYZ</i> .....	4
3.2.3 Compatibilité des opérandes .....	5
3.3 Mot-clé <i>COMP_INCR</i> .....	4
3.4 Mot clé <i>INCREMENT</i> / <i>NEWTON</i> / <i>CONVERGENCE</i> / <i>SUIVI_DDL</i> .....	4
3.5 Opérande <i>INFO</i> .....	5
4 Fonctionnement de la macro_commande .....	5
5 Exemple d'utilisation .....	5

## 2 Syntaxe

```

statnl [evol_noli] = STAT_NON_LINE
(
  ♦ MATER =mater,                                     [mater]
  ♦ COMP_INCR = _F (voir le document [U4.51.11] ),
  ♦ COMP_ELAS = _F (voir le document [U4.51.11] ),
  ♦ INCREMENT = _F (voir le document [U4.51.03]),
  ♦ NEWTON = _F ( voir le document [U4.51.03]),
  ♦ RECH_LINEAIRE = _F ( voir le document [U4.51.03]),
  ♦ ARCHIVAGE = _F ( voir le document [U4.51.03]),
  ♦ CONVERGENCE = _F ( voir le document [U4.51.03]),
  ♦ SUIVI_DDL = _F ( voir le document [U4.51.03]),
  ♦ MODELISATION = / '3D'                               [ DEFAULT ]
                  / 'C_PLAN'
                  / 'D_PLAN'
  ♦ SIGM_IMPOSE=_F(
      ♦ SIXX = sigxx                                     [fonction]
      ♦ SIYY = sigyy                                     [fonction]
      ♦ SIZZ = sigzz                                     [fonction]
      ♦ SIXY = sigxy                                     [fonction]
      ♦ SIXZ = sigxz                                     [fonction]
      ♦ SIYZ = sigyz                                     [fonction]
      )
  ♦ EPSI_IMPOSE=_F(
      ♦ EPXX = epsxx                                     [fonction]
      ♦ EPHY = epsyy                                     [fonction]
      ♦ EPZZ = epszz                                     [fonction]
      ♦ EPXY = epsxy                                     [fonction]
      ♦ EPXZ = epsxz                                     [fonction]
      ♦ EPHY = epsyz                                     [fonction]
      ),
  ♦ SIGM_INIT=_F(
      ♦ SIXX = sigxx                                     [R]
      ♦ SIYY = sigyy                                     [R]
      ♦ SIZZ = sigzz                                     [R]
      ♦ SIXY = sigxy                                     [R]
      ♦ SIXZ = sigxz                                     [R]
      ♦ SIYZ = sigyz                                     [R]
      )
  ♦ EPSI_INIT=_F(
      ♦ EPXX = epsxx                                     [R]
      ♦ EPHY = epsyy                                     [R]
      ♦ EPZZ = epszz                                     [R]
      ♦ EPXY = epsxy                                     [R]
      ♦ EPXZ = epsxz                                     [R]

```

```
    ◇ EPYZ = epsyz                                [R]
      ),

    ◇ VARI_INIT=_F(
      ◇ VALE = vari                                [R]
    ◇ INFO =      / 1,                             [DEFAULT]
                  / 2,

    );
```

## 3 Opérandes

### 3.1 Opérande **MATER**

◆ **MATER** = mater,

- Ce mot-clé permet de renseigner le nom du matériau (*mater*) défini par *DEFI\_MATERIAU* [U4.43.01], où sont fournis les paramètres nécessaires au comportement choisi.

### 3.2 Mots-clés **COMP\_INCR** / **COMP\_ELAS**

La syntaxe de ces mots clés est décrite dans le document [U4.51.11].

### 3.3 Mot clé **INCREMENT** / **NEWTON** / **CONVERGENCE** / **SUIVI\_DDL**

La syntaxe de ces mots-clés est décrite dans le document [U4.51.03].

Le mot-clé **INCREMENT** définit les intervalles de temps pris dans la méthode incrémentale.

Les mots-clés **NEWTON**, **CONVERGENCE**, **RECH\_LINEAIRE**, facultatifs, permettent de modifier les valeurs par défaut des paramètres de convergence de la méthode de Newton.

Le mot-clé **SUIVI\_DDL** permet de suivre l'évolution en un point de un ou plusieurs champs au cours des itérations de Newton.

Le mot clé **ARCHIVAGE** permet de définir la stratégie d'archivage des résultats au cours du temps.

### 3.4 Mot clé **MODELISATION**

Ce mot-clé permet de définir la dimension du problème traité : 3D (par défaut) ou bien 2D : déformation plane ou contrainte plane. Dans le cas 2D, les composantes des tenseurs fournis sous les mots-clés **SIGM\_IMPOSE**, **EPSI\_IMPOSE**, **SIGM\_INIT**, **EPSI\_INIT** sont au nombre de 4 : XX, YY, ZZ et XY.

### 3.5 Mots clés **SIGM\_IMPOSE** / **EPSI\_IMPOSE**

#### 3.5.1 Opérandes **SIXX**, **SIYY**, **SIZZ**, **SIXY**, **SIXZ**, **SIYZ**

Ces mot-clés permettent de définir des composantes du tenseur de contraintes imposées au point matériel, par l'intermédiaire de fonctions du temps. Ces fonctions peuvent être définies à l'aide de **DEFI\_FONCTION** [U4.31.02] ou à l'aide de **FORMULE** [U4.31.05] .

Par défaut, les composantes non affectées sont identiquement nulles.

#### 3.5.2 Opérandes **EPXX**, **EPYY**, **EPZZ**, **EPXY**, **EPXZ**, **EPYZ**

Ces mot-clés permettent de définir des composantes du tenseur de déformation imposées au point matériel, par l'intermédiaire de fonctions du temps. Ces fonctions peuvent être définies à l'aide de `DEFI_FONCTION` [U4.31.02] ou à l'aide de `FORMULE` [U4.31.05] .

Par défaut, les composantes non affectées sont laissées sans valeur ( pas de déformation imposée).

### 3.5.3 Compatibilité des opérandes

Pour une composante donnée, on ne peut définir à la fois une contrainte imposée et une déformation imposée.

## 3.6 Mots clés `SIGM_INIT` / `EPSI_INIT` / `VARI_INIT`

Ces mots clés permettent de définir un état initial par la donnée :

- des composantes des contraintes initiales (toutes les composantes ne sont pas nécessaires, par défaut on prend la valeur 0),
- des composantes des déformations initiales (si le mot clé `EPSI_INIT` est présent, il faut fournir toutes les composantes des déformations initiales : 4 en 2D, et 6 en 3D)
- l'ensemble des variables internes initiales pour le comportement utilisé.

Cette fonctionnalité est illustrée dans le test `SSNV160E` :

## 3.7 Opérande `INFO`

◇ `INFO = inf`

Permet d'effectuer dans le fichier message diverses impressions intermédiaires.

# 4 Fonctionnement de la macro\_commande

Cette macro\_commande a pour but de restreindre au strict nécessaire les données relatives à une simulation sur un point matériel pour un modèle de comportement incrémental.

Le fonctionnement interne allège donc le fichier de commandes de l'utilisateur, en réalisant des opérations répétitives pour ce genre de situations :

- vérifications préliminaires ;
- création d'un maillage d'un seul élément à un seul point de Gauss (un tétraèdre à 4 noeuds en 3D, un triangle à 3 noeuds en 2D)
- affectation d'un modèle 3D ou `C_PLAN` ou `D_PLAN`
- affectation du matériau sur ce maillage ;
- affectation des chargements :
  - en ce qui concerne les déformations imposées, pour chaque composante affectée via un des mots-clés de `EPSI_IMPOSE`, création d'un chargement unitaire en déformation qui sera multiplié par la fonction du temps fournie pour cette composante par l'utilisateur ;
  - en ce qui concerne les contraintes imposées, pour chaque composante affectée via un des mots-clés de `SIGM_IMPOSE`, création d'un chargement unitaire en contraintes qui sera multiplié par la fonction du temps fournie pour cette composante par l'utilisateur ;
- récupération des mots-clés de `STAT_NON_LINE`. Tous les mots-clés ayant des valeurs par défaut sont utilisés, sauf s'ils sont surchargés par l'utilisateur (`NEWTON`, `CONVERGENCE`, `SUIVI_DDL`, `ARCHIVAGE`, `RECH_LINEAIRE`) et de l'état initial.

Après l'appel à `STAT_NON_LINE`, l'ensemble des résultats (6 ou 4 composantes de contraintes et de déformations, variables internes) sont stockés dans une table. Pour chaque composante (colonne de la table) figure l'évolution en fonction du temps.

# 5 Exemple d'utilisation

Titre : Macro-commande SIMU\_POINT\_MAT  
Auteur(s) : J.M.PROIX

Date : 26-12/07  
Clé : U4.51.12-B1 Page : 6/6

Cet exemple est issu du test COMP001A :

#### # définition du matériau

```
ACIER=DEFI_MATERIAU(ELAS=_F(E=150000.0, NU=0.3, ),  
                    VENDOCHAB=_F(S_VP=0.0,  
                                  SEDVP1=0.0,  
                                  SEDVP2=0.0,  
                                  N_VP=12.0,  
                                  M_VP=9.0,  
                                  K_VP=2110.0,  
                                  R_D=6.3,  
                                  A_D=3191.0,  
                                  K_D=6.3, ), ), );
```

#### # discrétisation en temps

```
INSTANTS=DEFI_LIST_REEL(DEBUT=0.0,  
                        INTERVALLE=( _F(JUSQU_A=0.2, NOMBRE=10, ),  
                                       _F(JUSQU_A=2.0, NOMBRE=10, ),  
                                       ... ))
```

#### # chargement : histoire de contrainte (on traite ici un essai de fluage)

```
CHARGE=DEFI_FONCTION(NOM_PARA='INST', VALE=  
  (0.0, 0.0,  
   0.1, -200.0,  
   3000000.0, -200.0, ), ), );
```

#### # résolution

```
RESUXX=SIMU_POINT_MAT(  
  COMP_INCR=_F( RELATION='VENDOCHAB', ITER_INTE_MAXI=60 ),  
  CONVERGENCE=_F( ITER_GLOB_MAXI=60, ),  
  MATER = ACIER,  
  INCREMENT=_F( LIST_INST =INSTANTS,  
                SUBD_METHODE = 'UNIFORME',  
                SUBD_PAS = 4,  
                SUBD_PAS_MINI = 0.00000001,  
                ),  
  SIGM_IMPOSE=_F( SIXX = CHARGE, ),  
  )  
  
IMPR_TABLE(TABLE = RESUXX)
```