Exemple 1 - Implantation EF du modèle: Z-ABA

å Mode utilisateur Z-MAT

Modèle EVP classique

Surface de charge $f = J(\underline{\sigma} - \underline{X}) - R - R_0$

loi d'écoulement $\underline{\dot{\varepsilon}}_p = \frac{3}{2}\frac{\underline{\sigma} - \underline{X}}{J(\underline{\sigma} - \underline{X})}\dot{p}; \quad \dot{p} = \left\langle \frac{f}{K} \right\rangle^n$

lois d'état $\underline{X} = C\underline{\alpha}; \quad R = bQr$

q. ° d'évolution $\underline{\dot{\alpha}} = \underline{\dot{\varepsilon}}_p - \frac{D}{C}\underline{X}\dot{p}; \quad \dot{r} = 1 - \frac{R}{Q}\sqrt{\dot{p}}$

Librairie Z- MAT: acier

```
*** material
  * integration

***behavior gen_evp
  **elasticity isotropic
     young, poisson [temp]
 [** thermal strain
     alpha, temp_ref]
  ** potential gen_evp ev
   * criterion mises
   * flow norton
      K, n  [temp]
   *kinematic nonlinear
      C, D [temp]
   *isotropic non linear
      Q, b, R0 [temp]
***  return
```

Fichier ABAQUS: modele.inp

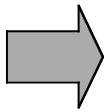
```
.....
* SOLID SECTION, ELSET = ..., MATERIAL = acier
* MATERIAL, NAME = acier
* DEPVAR
...
* USER MATERIAL, CONSTANTS = 1
0.0
* USER SUBROUTINES,INPUT=umat.f
....
```

UMAT

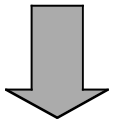
Exemple 2 - Implantation EF du modèle: Z-ABA

å Mode Développeur Z-FRONT

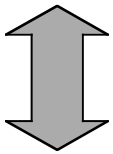
```
@Class NEW : BASIC_NL_BEHAVIOR {
 @Name new;
 @SubClass ELASTICITY elasticity;
 @Coefs    K, n, R0, ... ;
 @sVarInt  evcum ... ;
 @tVarInt  eel, evi, alpha1 ... ;
 @sVarAux  R1 ... ;
 @tVarAux  X1 ... ;
[@Implicit]
};
@StrainPart {
  evi = eto - eel;
  sig  = *elasticity*eel;
+ choix de la méthode d'intégration:
C++ code
};
@Derivative {
Définition des lois d'état et des
équations d'évolution: C++ code
}
[@CalcGradF {
Définition de la matrice jacobienne
pour une méthode implicite: C++ code
}]
```



Compilation



```
***material
  *integration
***behavior new
 **elasticity isotropic
young, poisson, [alpha,temp]
 **model_coef
coefs, [temp]
***return
```



Mode utilisateur

Implantation EF du modèle: Z-ABA

å Méthodes d'intégration

Ø Explicite (Runge-Kutta avec pas de temps adaptatif)
- plus facile à mettre en oeuvre
- moins robuste (calcul à grand nombre de cycles)

Ø Implicite (θ-méthode)
- plus robuste
- saut de cycles (durée de vie)
- nécessite le calcul de la matrice jacobienne

Equations d'évolution des variables internes du modèle

$$\begin{Bmatrix} \dot{s}_i \\ \dot{\underline{T}}_i \end{Bmatrix} = \begin{Bmatrix} F_1(s_j, \underline{T}_j) \\ F_2(\underline{T}_j, s_j) \end{Bmatrix}$$

$$t \cap t + \theta \Delta t$$

$$s_i \cap s_i(t) + \theta \Delta s_i$$

$$\underline{T}_i \cap \underline{T}_i(t) + \theta \Delta \underline{T}_i$$

$$R_{s_i} = \Delta s_i - F_1(\Delta s_j, \Delta \underline{T}_j, \Delta t)$$

$$\underline{R}_{T_i} = \Delta \underline{T}_i - F_2(\Delta \underline{T}_j, \Delta s_j, \Delta t)$$

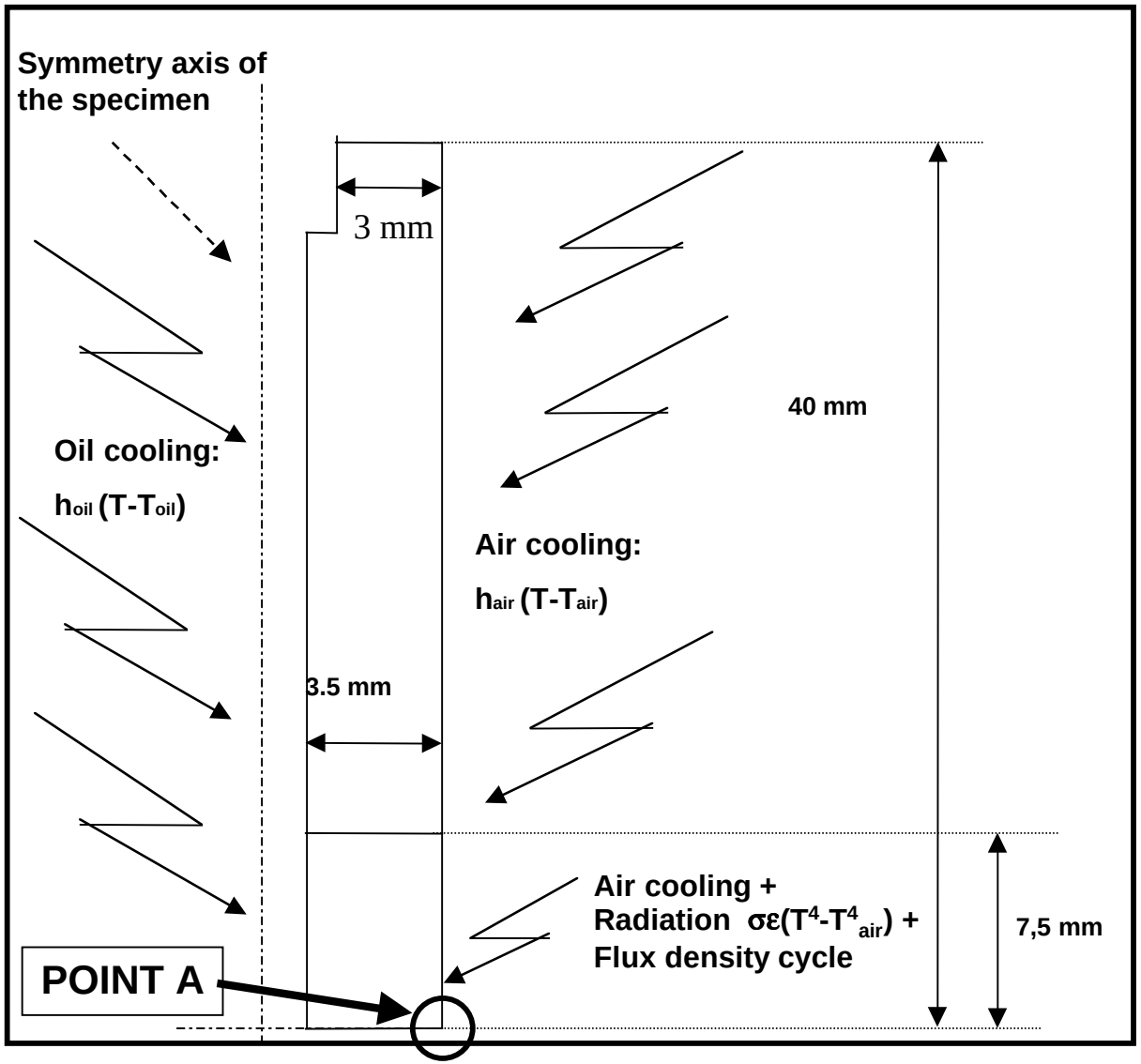
Matrice jacobienne

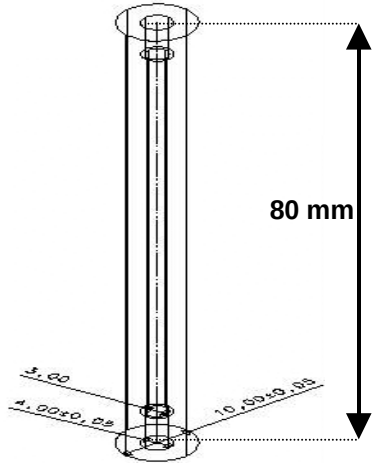
$$\begin{Bmatrix} R_{s_i} \\ \underline{R}_{T_i} \end{Bmatrix} = \begin{bmatrix} \frac{fR_{s_i}}{fs_k} & \frac{fR_{s_i}}{f\underline{T}_k} \\ \frac{f\underline{R}_{T_i}}{fs_k} & \frac{f\underline{R}_{T_i}}{f\underline{T}_k} \end{bmatrix} \begin{Bmatrix} \Delta s_i \\ \Delta \underline{T}_i \end{Bmatrix} = [0]$$

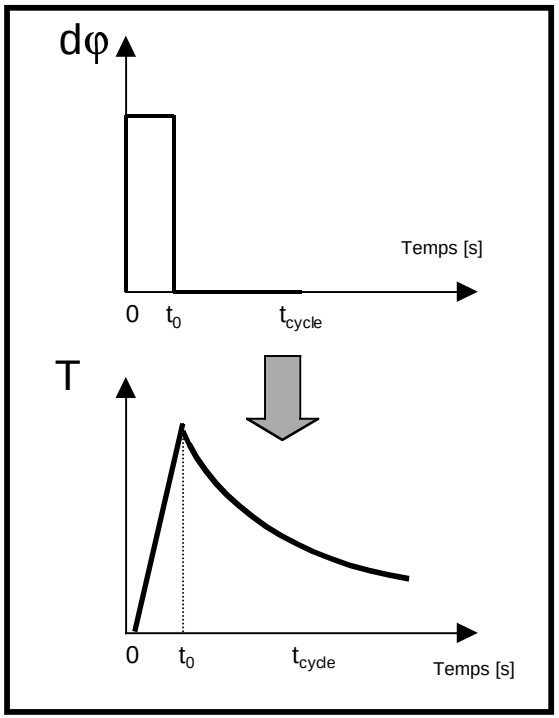
Résolue par l'algorithme de Newton-Raphson

Exemple 2 - détermination de CL thermiques

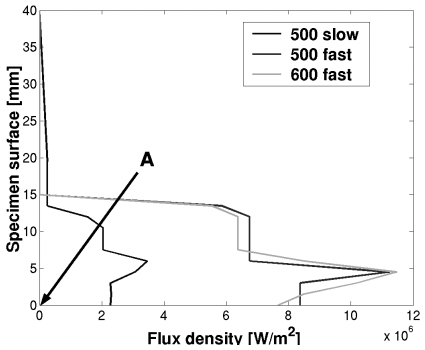
Ø Résumé des conditions aux limites







Exemple 2 - détermination de CL thermiques



Validation anisotherme: méthodologie du calcul EF