

Club utilisateur MFRONT

Utilisation de MFRONT pour
l'implémentation des lois de
comportement mécaniques dans
le code CYRANO3

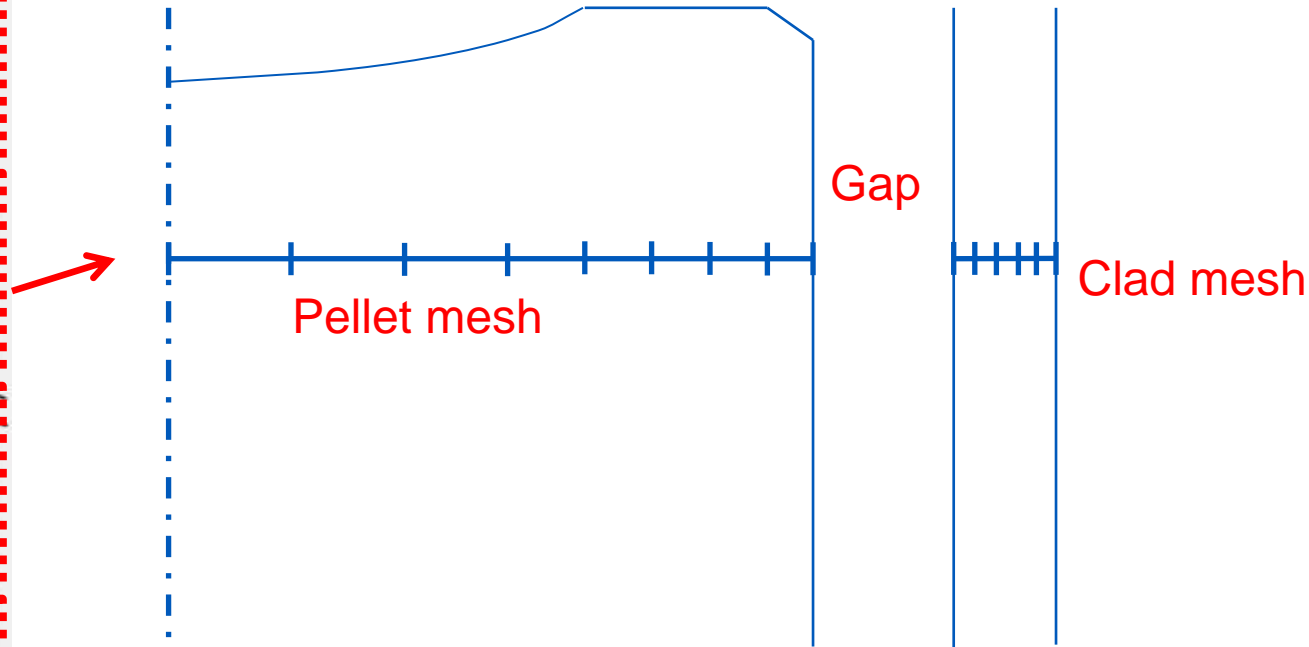
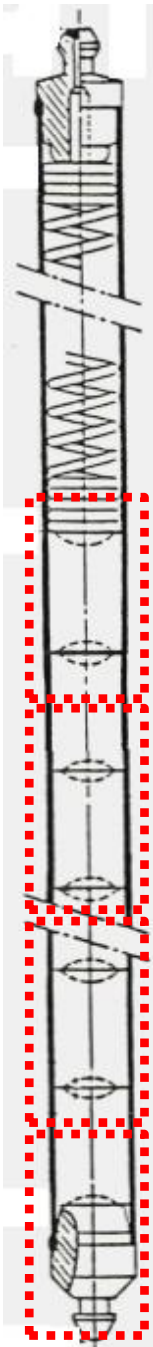
Charles Pétry – EDF R&D



Le code CYRANO3

Principe : un solveur EF "1,5D"

- Discrétisation axiale en « tranches »
- Transmission des efforts axiaux moyens d'une tranche à l'autre
- Maillage radial 1D + résolution axisymétrique + hyp. de contrainte plane généralisée



Le code CYRANO3 (2)

Problèmes multi-physiques fortement couplés

MECHANICAL ANALYSIS

Fuel :

- poro-elasticity & visco-plasticity
- μ and macro cracking
- fragments relocation
- dishing
- hourglass
- hooking conditions (axial friction)
- PCI
- cladding : elasto-visco-plasticity
- external corrosion effect

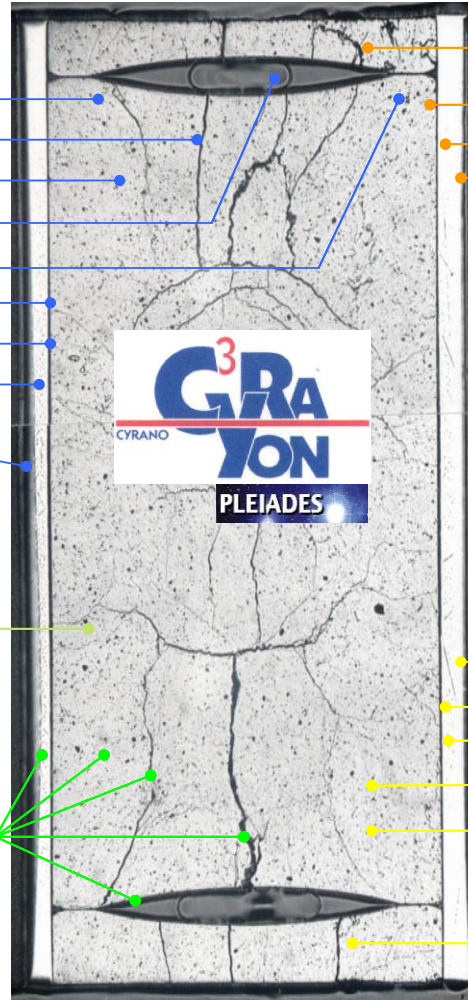
NEUTRONIC

Radial power distribution
Fission products evaluation

INTERNAL PRESSURE

Free volume areas :

- Pellet-cladding gap
- plenum
- macro cracking
- open porosity
- chamfer
- dishing



THERMAL ANALYSIS

- Radial temperature distribution
- Gap conductivity
- Cladding conductivity
- Corroded alloy conductivity

THERMIC-HYDRAULIC

- Heat transfert
- Oxydation effects
- Cooling's temperature

PHYSICAL-CHEMICAL

- Corrosion
- Hydruration
- Irradiated cladding behavior
- Pellet : irradiated local properties
- density, solid and fission gas swelling
- fission gas released
- thermal conductivity
- stoichiometry variation
- grain size

Le code CYRANO3 (3)

Résolution mécanique : principes

- **L'équilibre mécanique global** est résolu par un algorithme de Newton implicite
- **L'équilibre mécanique local** aussi -> la résolution locale fournit l'opérateur tangent cohérent nécessaire au calcul de l'équilibre global
- **L'hypothèse de contrainte plane généralisée** peut être traitée de 2 façons différentes
 1. Prise en compte directement **dans les équations** de la loi de comportement
 2. Vérification de l'hypothèse **a posteriori** via une boucle de convergence supplémentaire

Comparaison des temps d'exécution sur la base de plusieurs lois gaine

Temps CPU (s)	LEMAITRE IRRA	CARASSOU	STANDARD
Solution 1	11,95	14,89	14,10
Solution 2	62,50	70,07	60,61

Utilisation des lois MFRONT dans CYRANO3

Avantages

- L'implémentation d'une nouvelle loi demande **très peu de développements**
- L'implémentation est réalisée **une fois pour toutes** (identification, validation etc.)
- Les algorithmes de résolutions sont **optimisés**
- **Mutualisation** possible avec :
 1. Castem (applications PLEIADES)
 2. Code_Aster
 3.
- MFRONT passé en **open source** en 2014

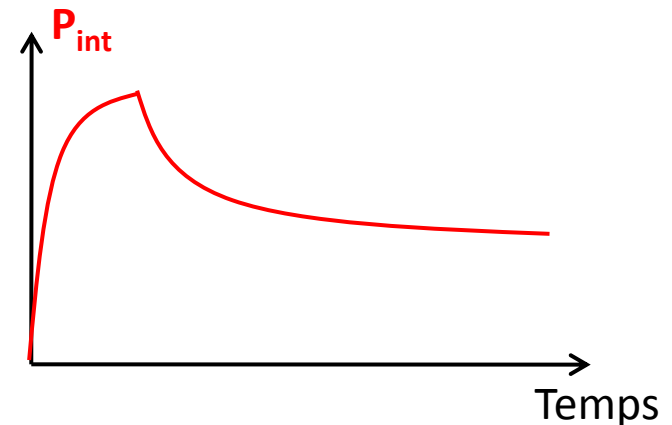
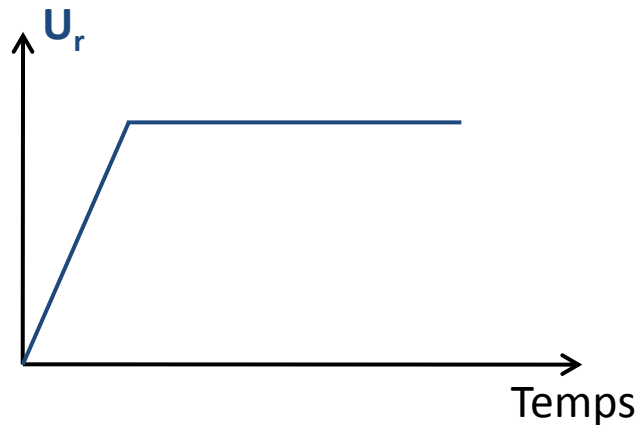
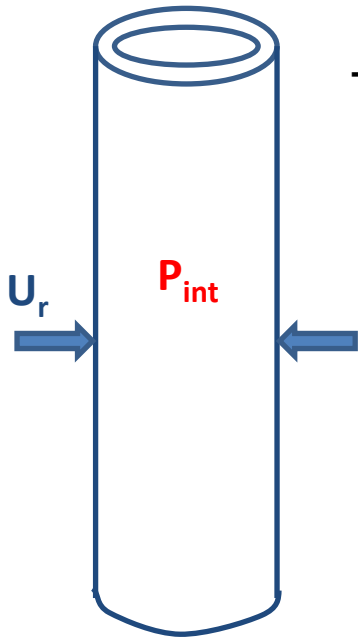
Quelques adaptations nécessaires

- Développement d'une **interface** spécifique pour CYRANO3
- Renvoi de **l'opérateur tangent cohérent** (nécessaire pour C3)
- Prise en compte de l'hypothèse de **contrainte plane généralisée** pour éviter la boucle externe qui ralentit la convergence générale

Quelques réalisations

Identification d'une loi d'écrouissage/relaxation pour le matériau de gainage

- **Principe** : chargement mécanique en pression interne (avec effet de fond)
- **Déplacement radial piloté** via une boucle de rétro-contrôle



L'état de contrainte est imposé par la géométrie...

$$\sigma_r \approx 0$$

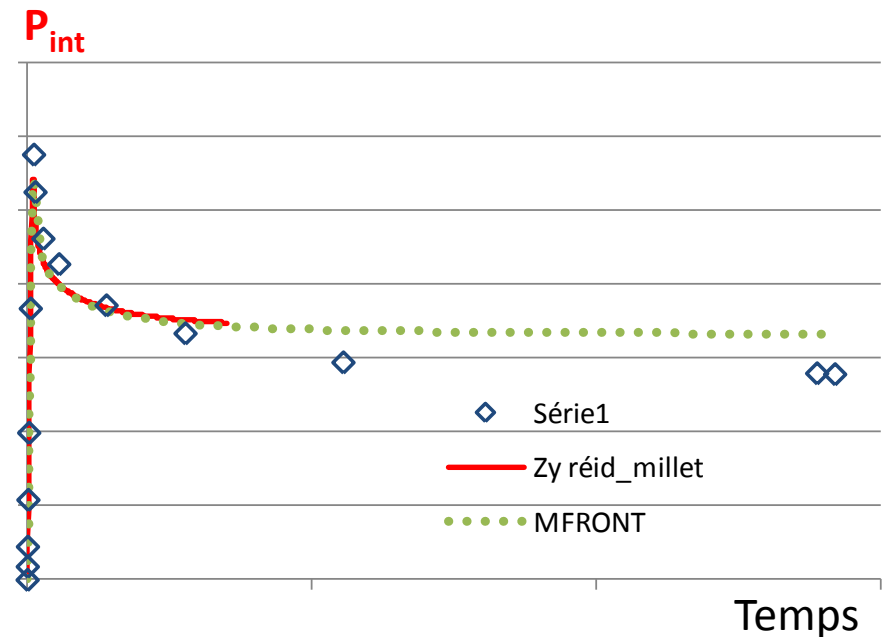
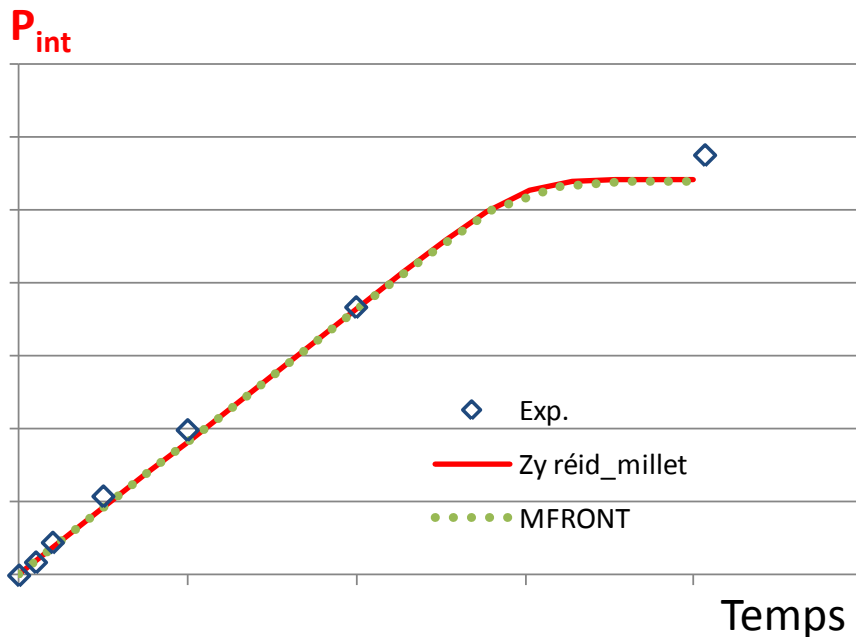
$$\sigma_\theta \approx \frac{PR}{e}$$

$$\sigma_z \approx \frac{\sigma_\theta}{2}$$

Quelques réalisations (2)

Identification d'une loi d'érouissage/relaxation pour le matériau de gainage (2)

- **loi élasto-viscoplastique** implémentée de façon indépendante dans MFRONT
- état de contrainte imposé via **MTEST**
- **surcouche python** pour déterminer l'incrément de pression interne compatible avec l'évolution du déplacement radial souhaitée (dichotomie) et effectuer l'optimisation des paramètres de la loi (simplex)



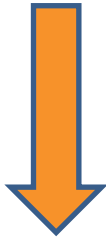
Quelques réalisations (3)

Implémentation dans CYRANO3 de lois de fluage pour le matériau de gainage

- lois de type **élasto-viscoplastique**
- élasticité **isotrope** / (visco)plasticité **isotrope** ou **orthotrope**
- **schéma d'intégration implicite**

Plasticité isotrope

- formalisme de type **Von Mises**
- mécanismes de déformation **additifs**



Utilisation des analyseurs spécifiques
(MultipleIsotropicMisesFlows...)

Plasticité orthotrope

- formalisme de type **Hill**
- mécanismes de déformation **additifs**



Utilisation de l'analyseur générique
(Implicit)

Quelques réalisations (4)

Exemple de loi isotrope à écoulements multiples

$$\varepsilon^{totale} = \varepsilon^{thermique} + \varepsilon^{élastique} + \varepsilon^{viscoplastique}$$

$$\varepsilon^{viscoplastique} = \varepsilon_{irradiation}^{fluage} + \varepsilon_{thermique}^{fluage} + \varepsilon_{instantanée}^{plastique}$$

Dépendance à la température, flux de neutrons rapides, fluence rapide...

```
@Parser      MultipleIsotropicMisesFlows;  
@ExternalStateVariable real flux;  
flux.setGlossaryName("FastNeutronFlux");  
@ExternalStateVariable real fluence;  
fluence.setGlossaryName("FastNeutronFluence");  
  
@FlowRule StrainHardeningCreep{ // irradiation  
@FlowRule StrainHardeningCreep{ // thermique  
@FlowRule Plasticity{           // plastique  
.....
```



Ecriture très simplifiée et “naturelle”

Quelques réalisations (5)

Exemple de loi orthotrope

$$\varepsilon^{totale} = \varepsilon^{thermique} + \varepsilon^{élastique} + \varepsilon^{viscoplastique}$$

$$\varepsilon^{viscoplastique} = \dot{p} \frac{H : \sigma}{\sigma_{Hill}}$$

Dépendance à la température, flux de neutrons rapides, fluence rapide...

```
@Parser      Implicit;  
@OrthotropicBehaviour;  
H = hillTensor<N,real>(Hz,Hr,Ht,Hrt,Hrz,Htz);  
// Hill Stress  
  real sigeq = sqrt(sig|H*sig);  
@TangentOperator{  
  Stensor4 Je;  
  getPartialJacobianInvert(Je);  
  Dt = D*Je;
```



Ecriture très simplifiée par l'utilisation de bibliothèques dédiées

Quelques réalisations (6)

Exemple de loi orthotrope (2)

Prise en compte de l'hypothèse de contrainte plane généralisée

```
@Integrator<AxisymmetricalGeneralisedPlaneStress, Append, AtEnd>{
    const stress szz=(lambda+2*mu)*(eel(1)+deel(1))+lambda*(eel(0)
        +deel(0)+eel(2)+deel(2));
    fetozz = (szz-sigzz-dsigzz)/young;
    feel(1) -= detozz ;

    // jacobian
    dfeel_ddetozz(1) = -1 ;
    dfetozz_ddetozz = real(0) ;
    dfetozz_ddeel(1) = (lambda+2*mu)/young ;
    dfetozz_ddeel(0) = lambda/young ;
    dfetozz_ddeel(2) = lambda/young ;
}
```



Surcharge de quelques termes spécifiques

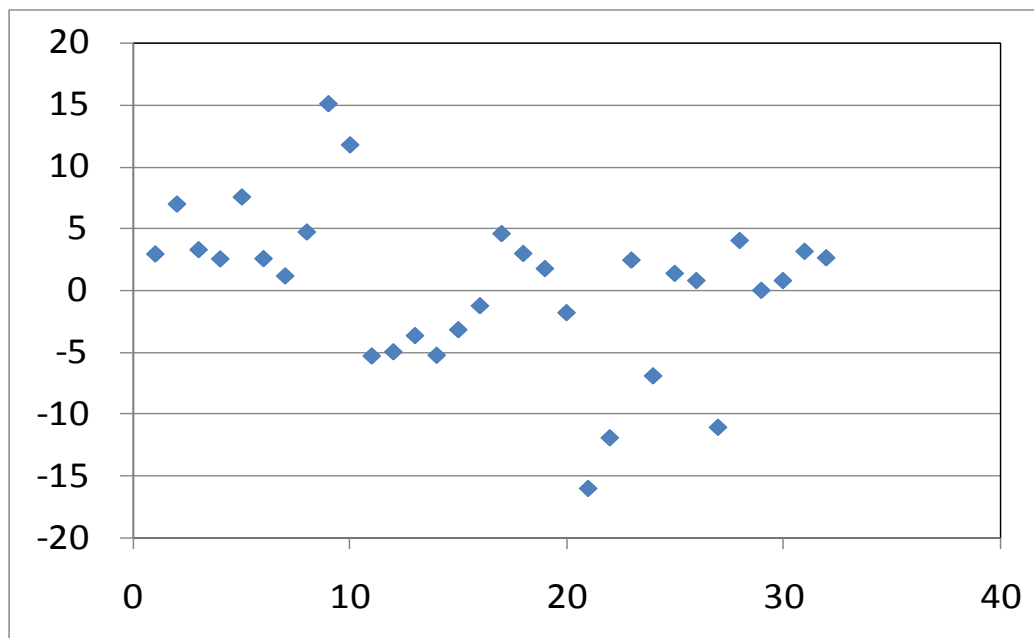
Le formalisme reste valide quelle que soit l'hypothèse choisie par l'utilisateur final...

Quelques réalisations (7)

Exemple de loi orthotrope (3)

Benchmark entre l'implémentation native de CYRANO3 et celle utilisant MFRONT
(32 cas tests = base de validation réduite CYRANO3)

Gain de temps CPU
MFRONT (%)



Temps CPU comparable voire meilleur dans 60% des cas

Conclusions / Perspectives

- Le **couplage CYRANO3 / MFRONT** est aujourd'hui effectif et permet de mener des calculs isofonctionnels avec ceux de la version native (opérateur tangent, contrainte plane généralisée)
- La **mutualisation** des lois de comportement entre les différentes applications est une réalité
- Les travaux en cours dans CYRANO3 concernent la mise en place d'une **interface spécifique** permettant le branchement générique des lois (UMAT++)
- Une **généralisation** de l'utilisation de MFRONT est envisagée à terme