

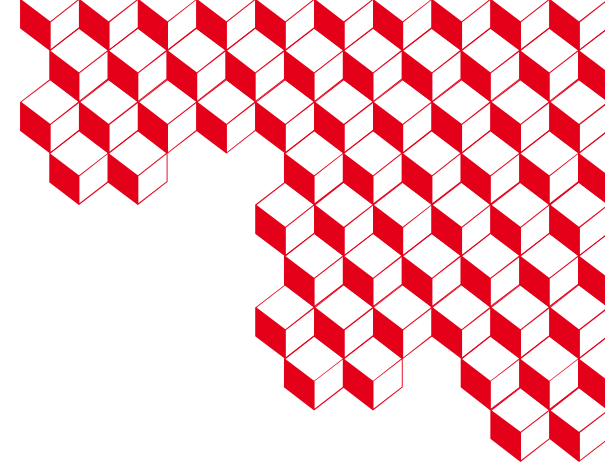


Rencontres Arcane 2023

IFPEN, 17 avril 2023

Phénix : un code multi-physiques sous Arcane

Thierry Pougeard Dulimbert



Phénix in a nut shell

Code multi-physique de dynamique rapide 2D-3D massivement parallèle :

- Des physiques intimement couplées
 - Hydrodynamique compressible (équations d'Euler), élasto-plasticité, détonique, rupture et endommagement, ...
 - Couplage aux bibliothèques de lois physiques (EOS, rhéologie)
- Des méthodes numériques adaptées
 - Volumes finis, Contexte multi-matériaux
 - Lagrangienne, ALE sur maillage non structuré : schéma VNR + projection multi-dimensionnel
 - Eulérienne sur maillage structuré : schéma lagrange (prédicteur-correcteur) + projection par directions alternées
- Parallélisme par décomposition de domaine (MPI)

Développé depuis 2012

- Codé en C++ sur la plateforme Arcane, 1 500 000 lignes (+ 370 000 lignes de python)
- Equipe de développement d'une quinzaine d'ingénieurs, répartis entre la DIF et le CESTA (Aquitaine)
- Remplacement de deux codes historiques (fin 90')
 - Processus de vérification intégré dans le PDL, ayant permis une mise en production en 6 ans



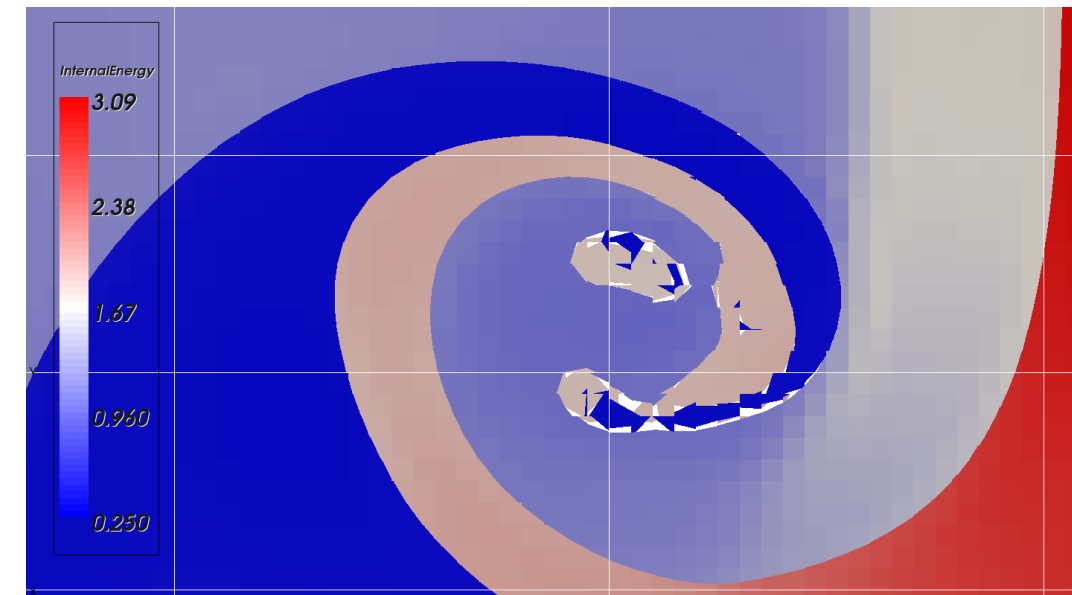
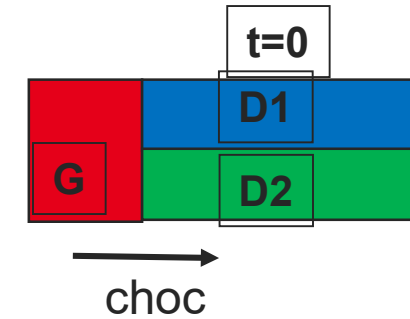
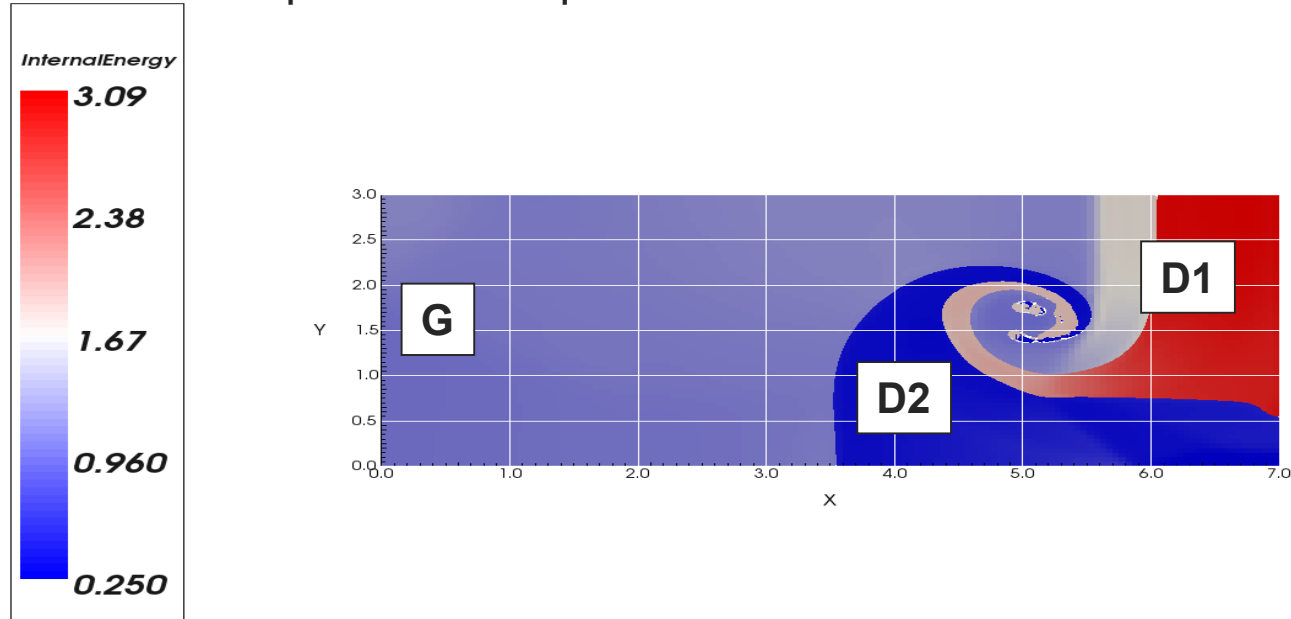
En production depuis 2018

- 90 utilisateurs sur trois centres, pour 550 millions d'heures de calculs en 2022

Exemples d'applications

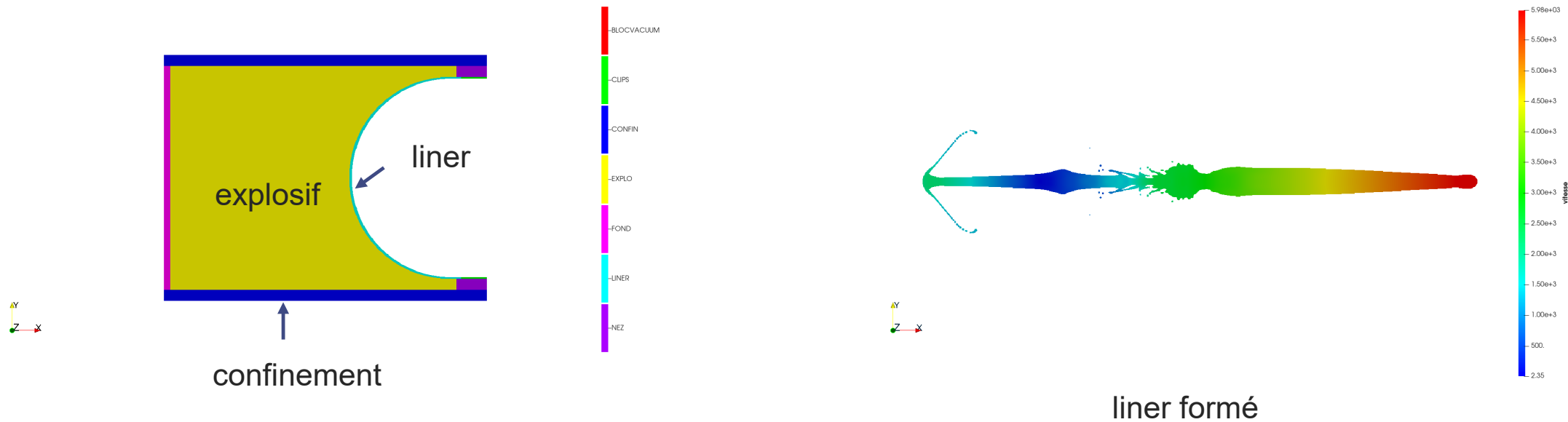
Ecoulements de fluides

- Exemple académique avec cas test d'instabilités de Richtmyer Meskov



Exemples d'applications

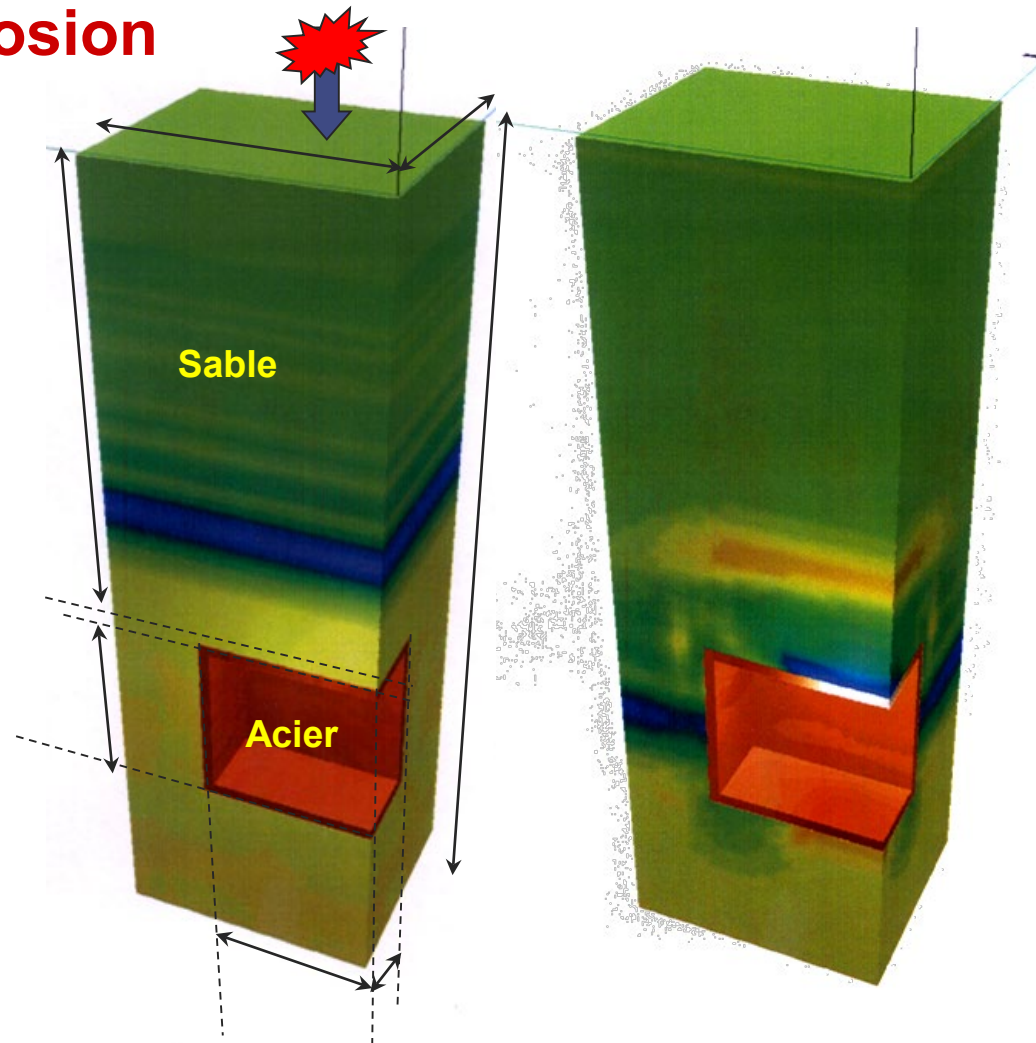
Simulation de charge creuse :



Exemples d'applications

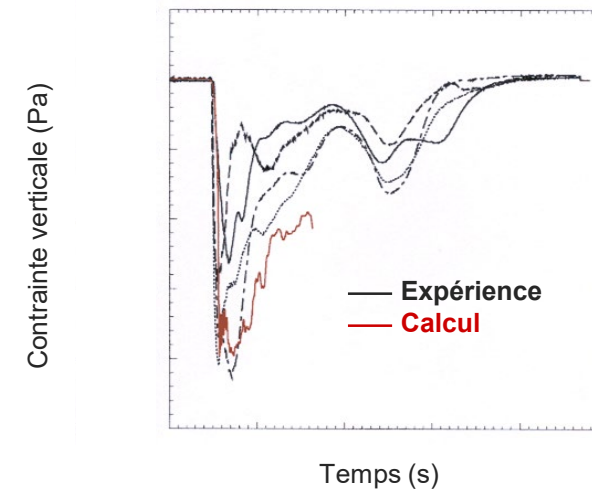
Résistance de structure

Explosion



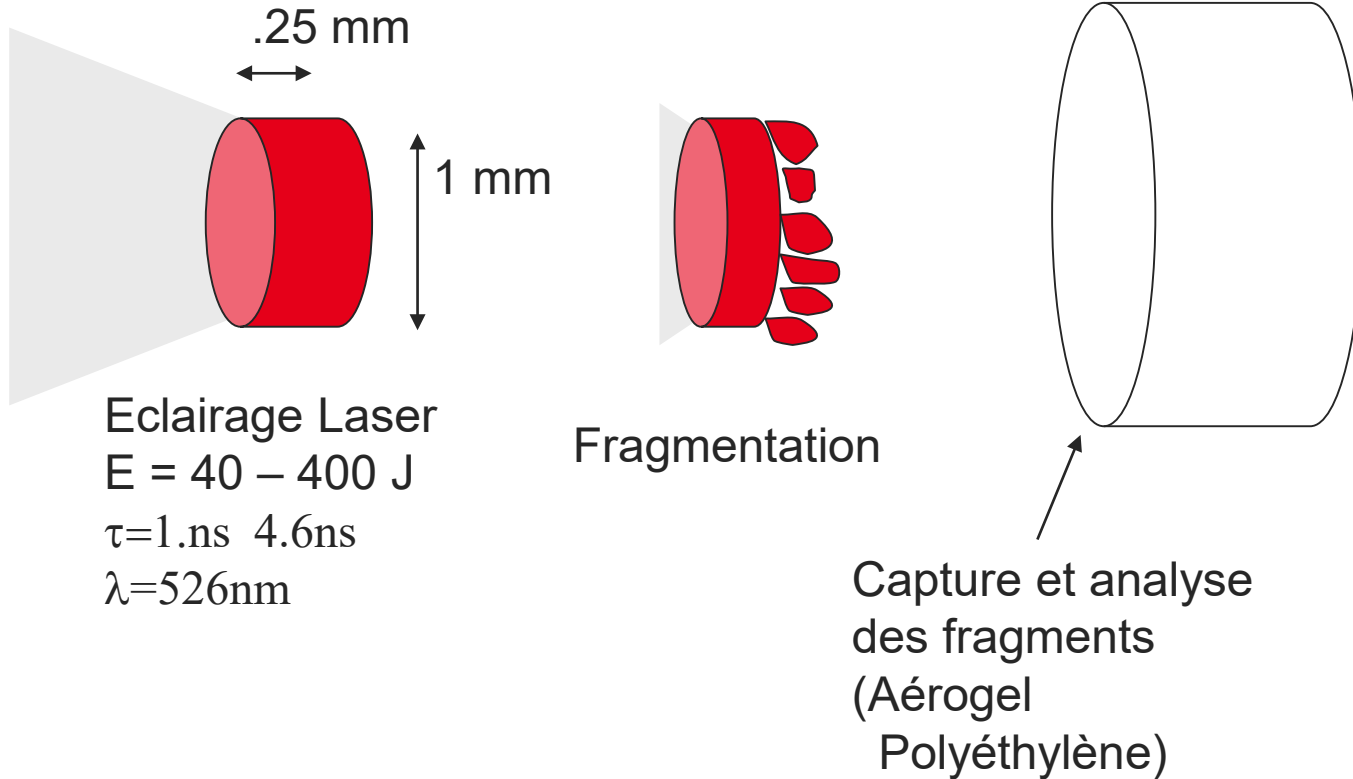
Comparaison avec une expérience réalisée au CEG :

Contrainte verticale dans le sable sur le toit de la structure

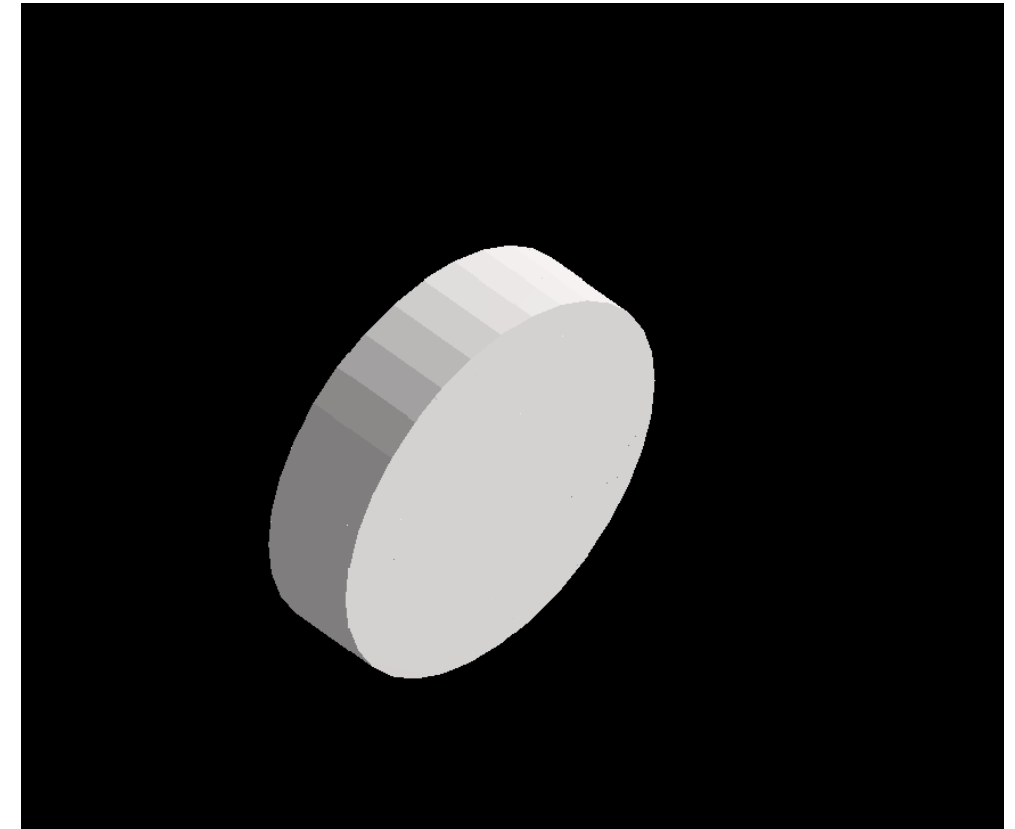


Exemples d'applications

Endommagement par laser:

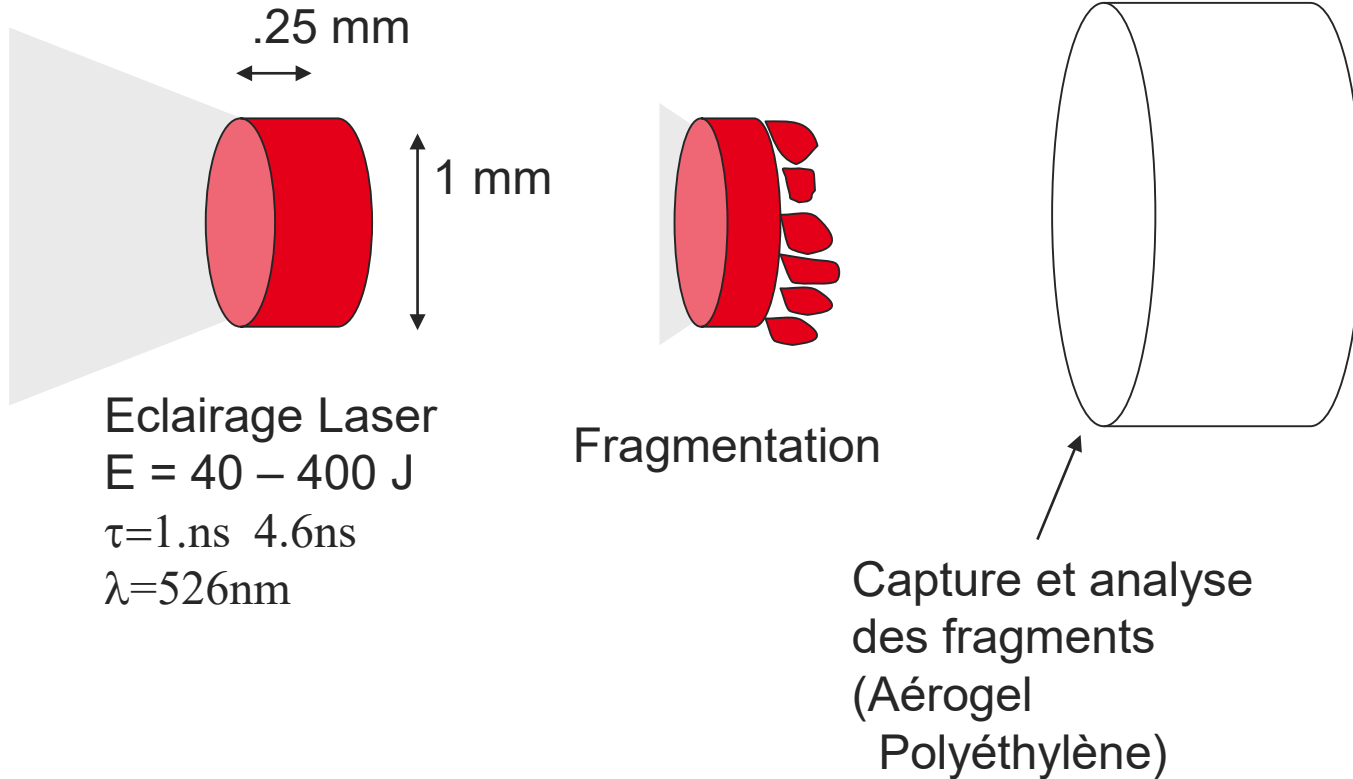


Simulation sur 850 ns
Visualisation : Vitesse V_y

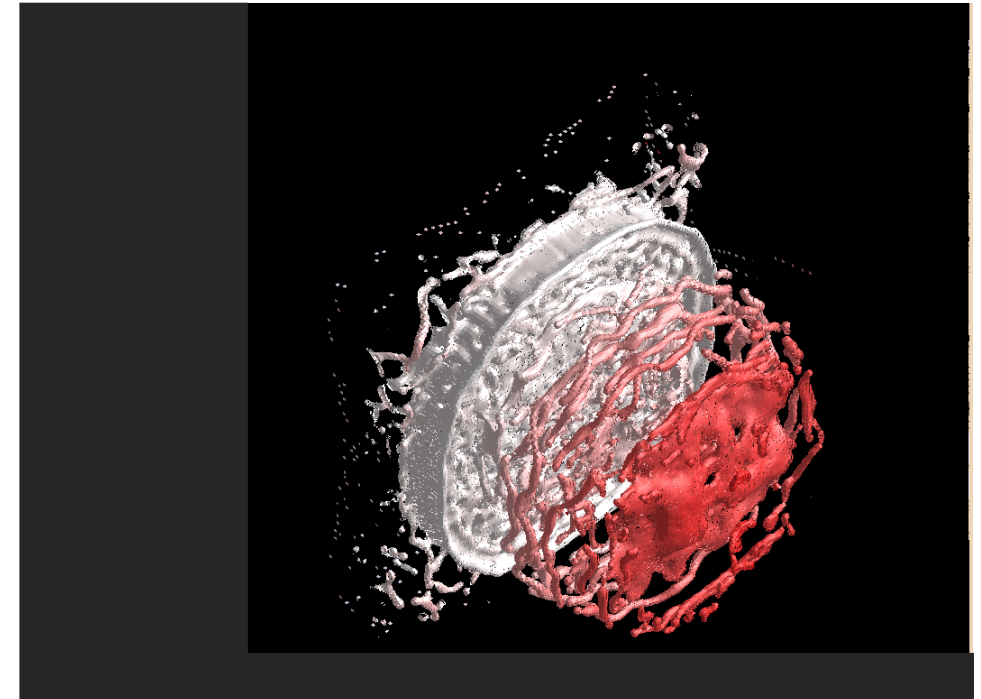


Exemples d'applications

Endommagement par laser:

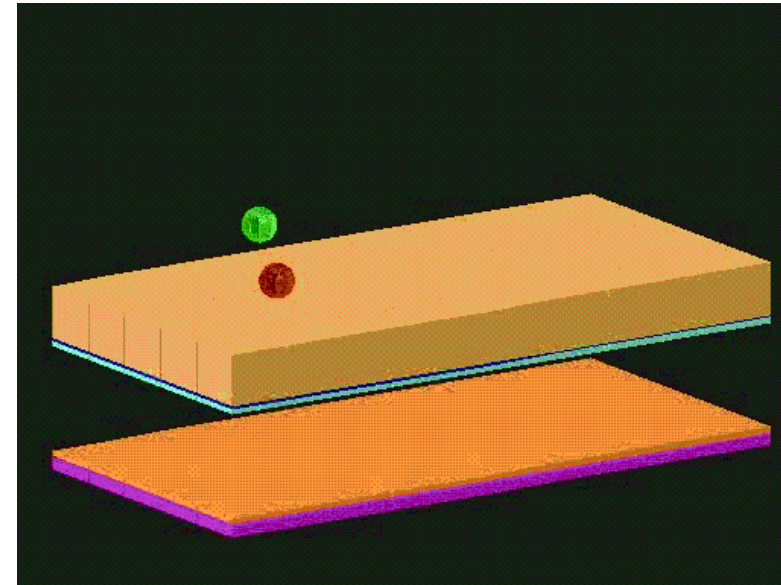
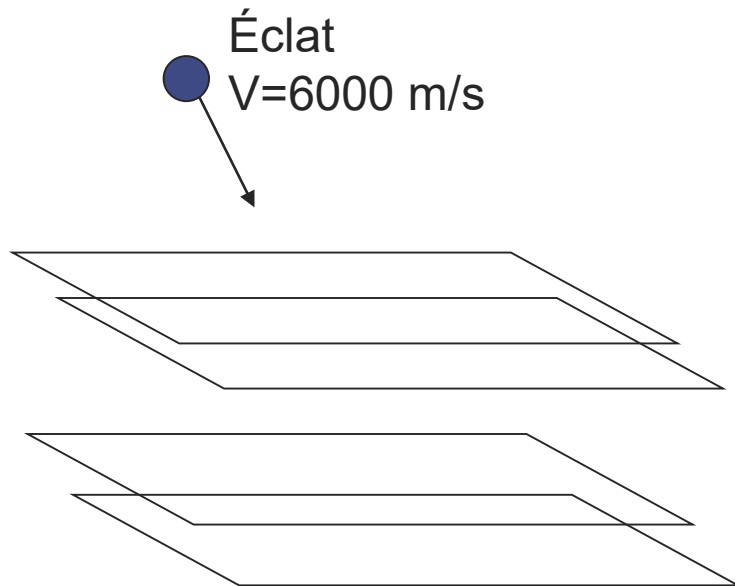


Simulation sur 850 ns
Visualisation : Vitesse V_y



Exemples d'applications

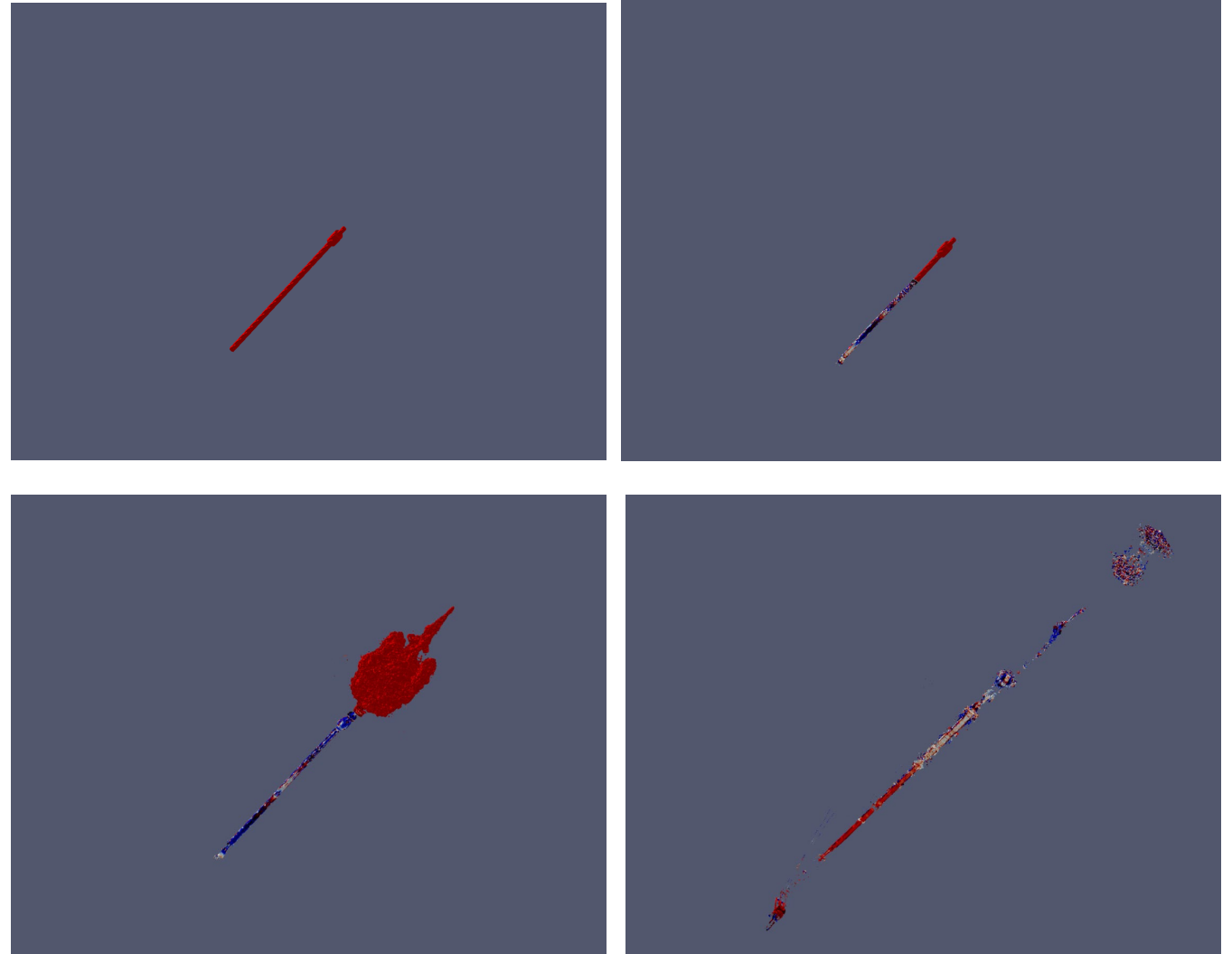
Impact hyper-vélocé :



Exemples d'applications

Calcul Grand challenge 2023 :

- Calcul de dynamique rapide
- Simulation Euler sur maillage structuré, en décomposition de domaine
- 23 milliards de mailles
- 65 500 sous-domaines MPI sur 1024 nœuds
- 172 millions d'heures consommées, 55 jours
- 3600 To de données produites



Focus sur un détail technologique de la simulation

Fonctionnalités Arcane utilisées

Fonctionnalités de bases

- Concepts de modules et services ; variables globales
- Lecture de maillage, décomposition de domaine, parallélisme

Fonctionnalités spécifiques

- Accesseurs directionnels pour les maillages cartésiens
- Tests unitaires
- AMR « patch-based » (voir présentation de Maxime Stauffert)
- Points d'entrées utilisateurs en C# (voir présentation de Nicolas Peton)

```
// Le maillage cartésien du calcul
ICartesianMesh* m_cartesian_mesh =
    ICartesianMesh::getReference(subDomain()->defaultMesh());

// Directions du maillage cartésien
CellDirectionMng cell_dm(m_cartesian_mesh->cellDirection(dir));

//-----
// Boucle sur les AllEnvCell
ENUMERATE_ALLENCVCELL (iallencvcell, m_mesh_material_mng, cell_dm.allCells()) {
    //
    // Donor Cell correspondante à l'AllEnvCell courante
    cell_d = (*iallencvcell).globalCell();

    // Pour recuperer les mailles avant/apres dans la direction courante
    DirCell dir_cell(cell_dm[cell_d]);

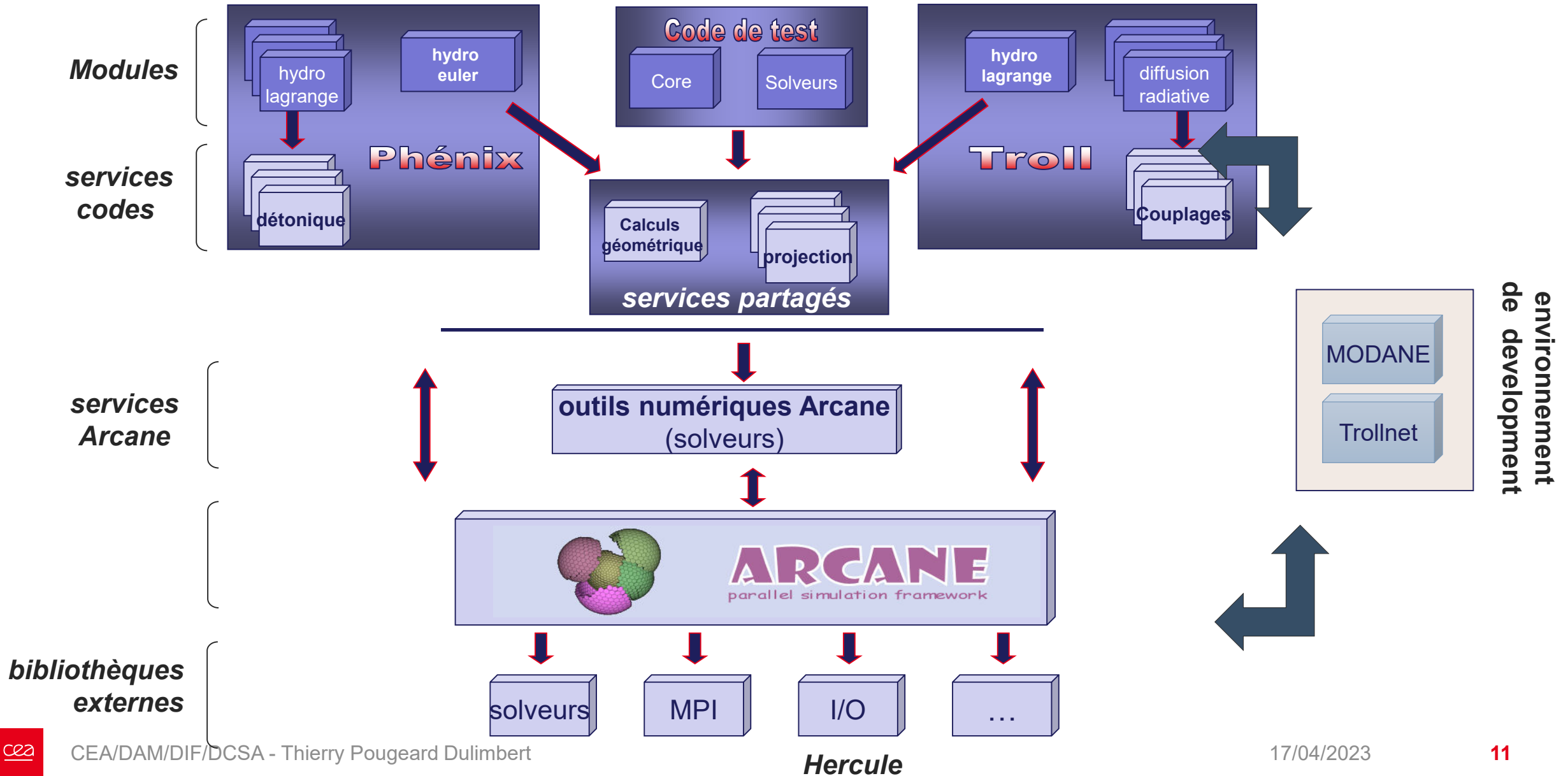
    cell_a = dir_cell.next();           // maille suivante dans la direction courante
    cell_n = dir_cell.previous();       // maille précédente dans la direction courante

    ...
}
```

Particularités

- Génération de sources via MODANE
- Jeu de données utilisateurs en surcouche du fichier de données Arcane + transcripteur
- Partage de sources avec le code Troll

Fonctionnalités Arcane utilisées



MODANE : Modélisation et génération de source

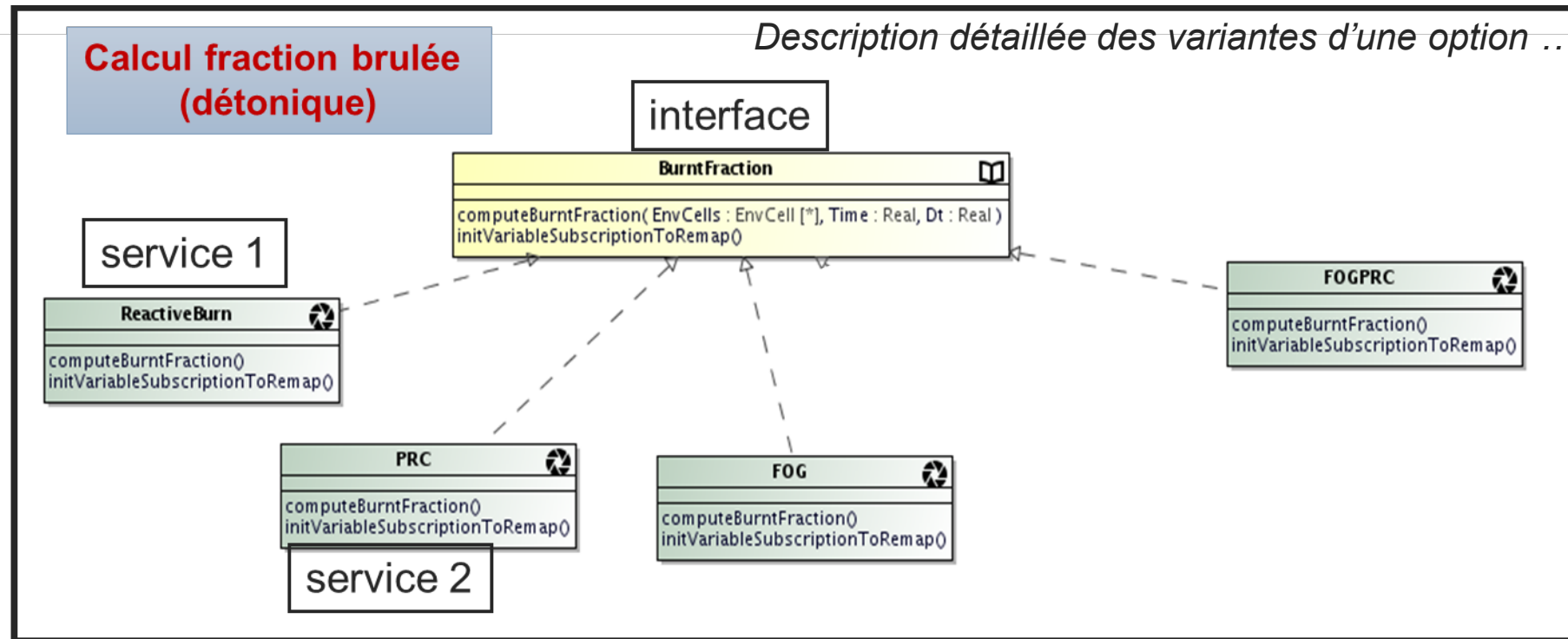
Retour d'expérience sur les précédents codes sous Arcane:

- Difficile d'identifier les entrées-sorties d'une fonction
- Confusion dans les différentes façon de déclarer les variables Arcane (dans l'axl, dans le source)
- En parallèle, expériences de modélisation UML a posteriori de composants

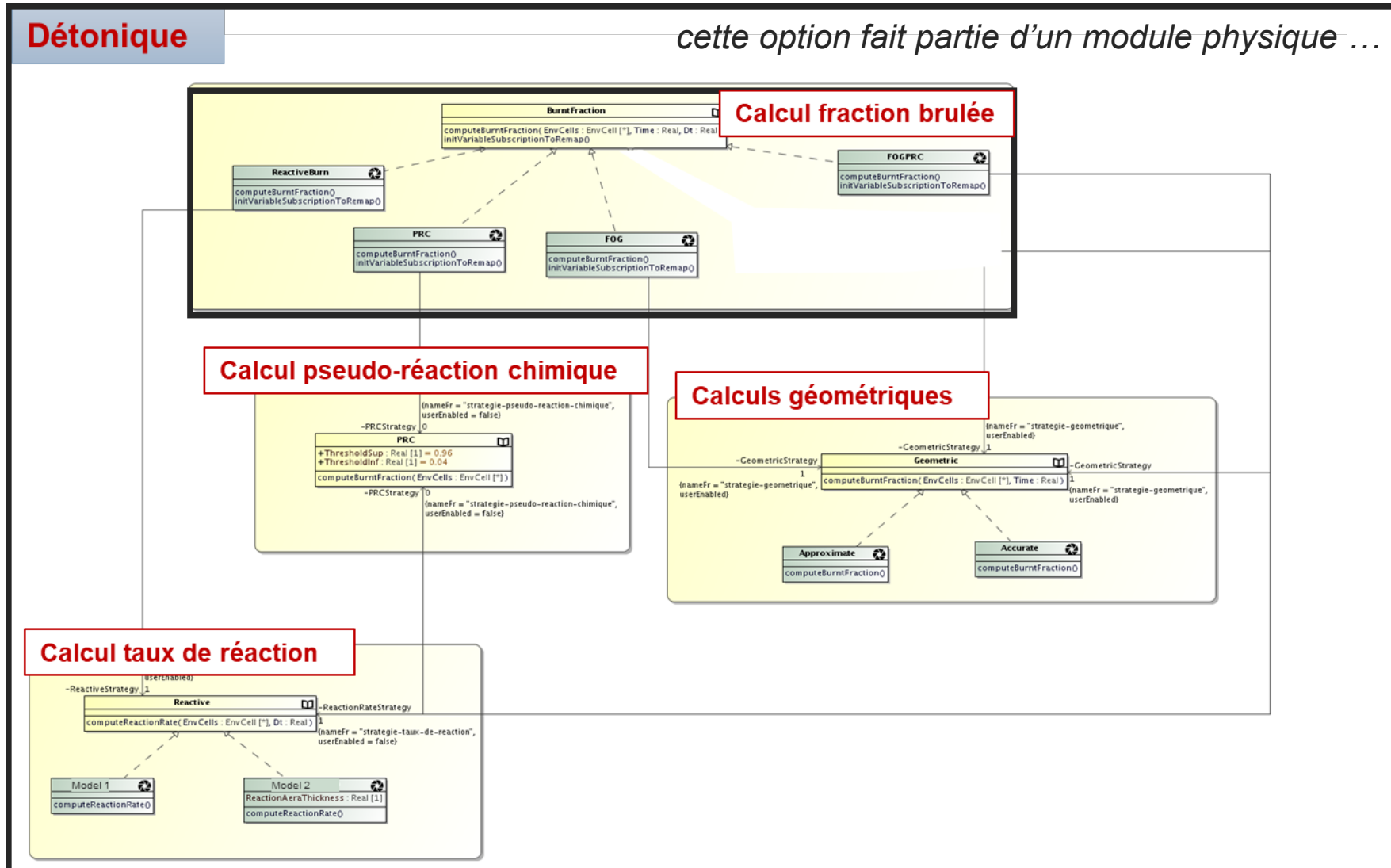
Proposition de DSSI : MODANE

- Modélisation informatique intégrant les concepts Arcane (modules, services)
- Intégré dans *MagicDraw*
- Spécification des entrées-sorties des fonctions
- Représentation graphique
- Génération du squelette des .cc et .h

MODANE : Modélisation et génération de source

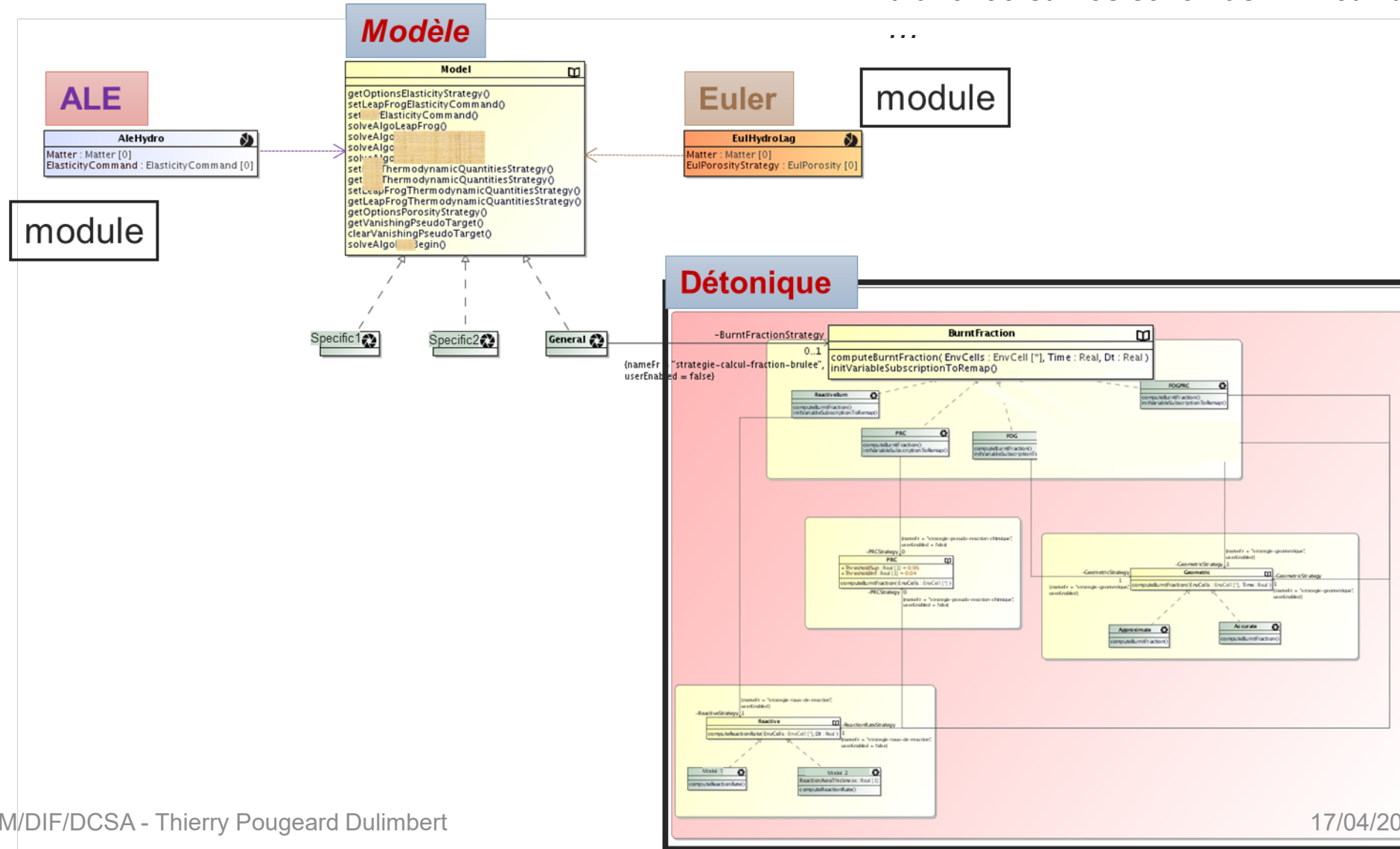


MODANE : Modélisation et génération de source

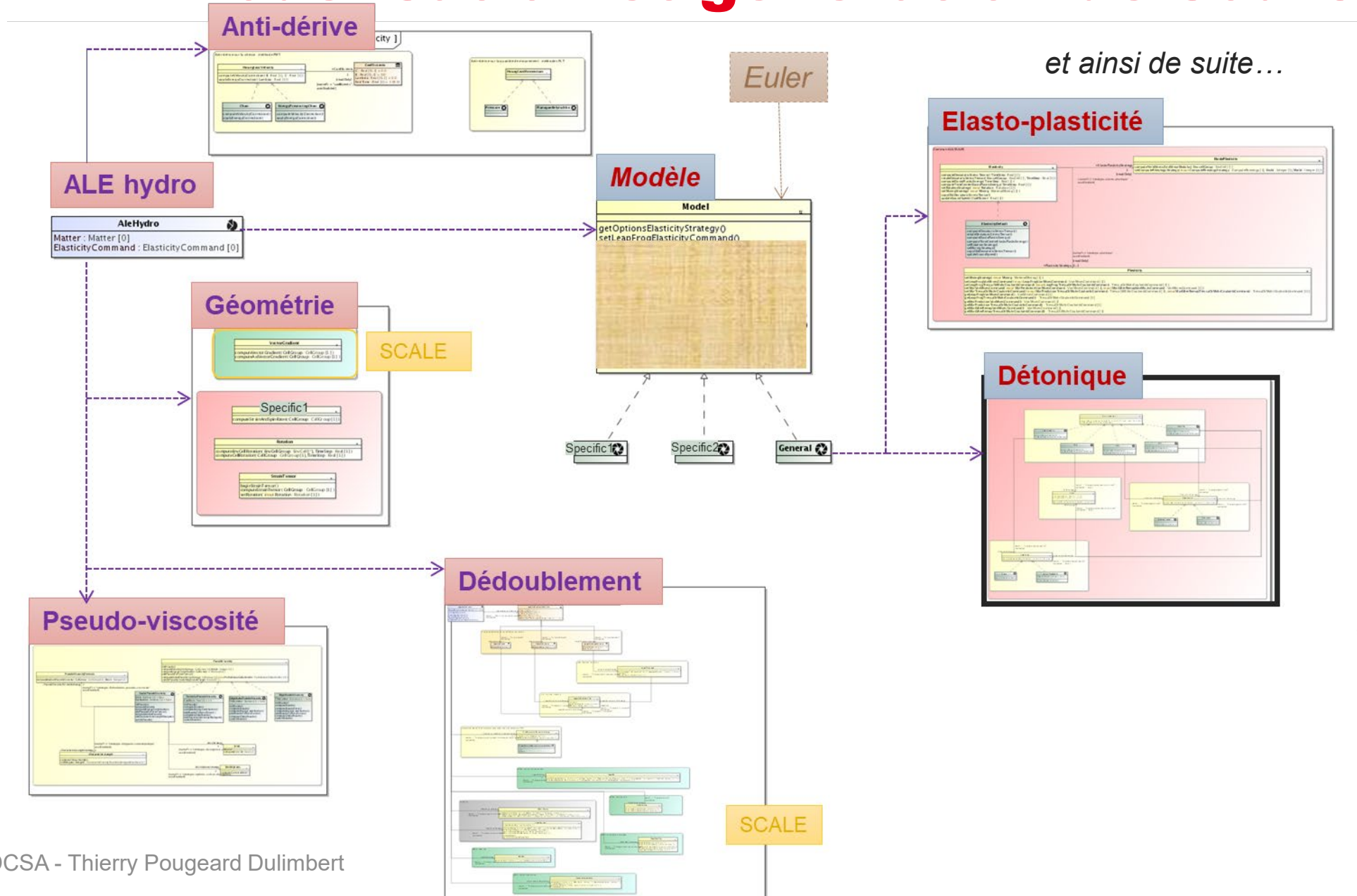


MODANE : Modélisation et génération de source

branchée sur les schémas ALE et Euler

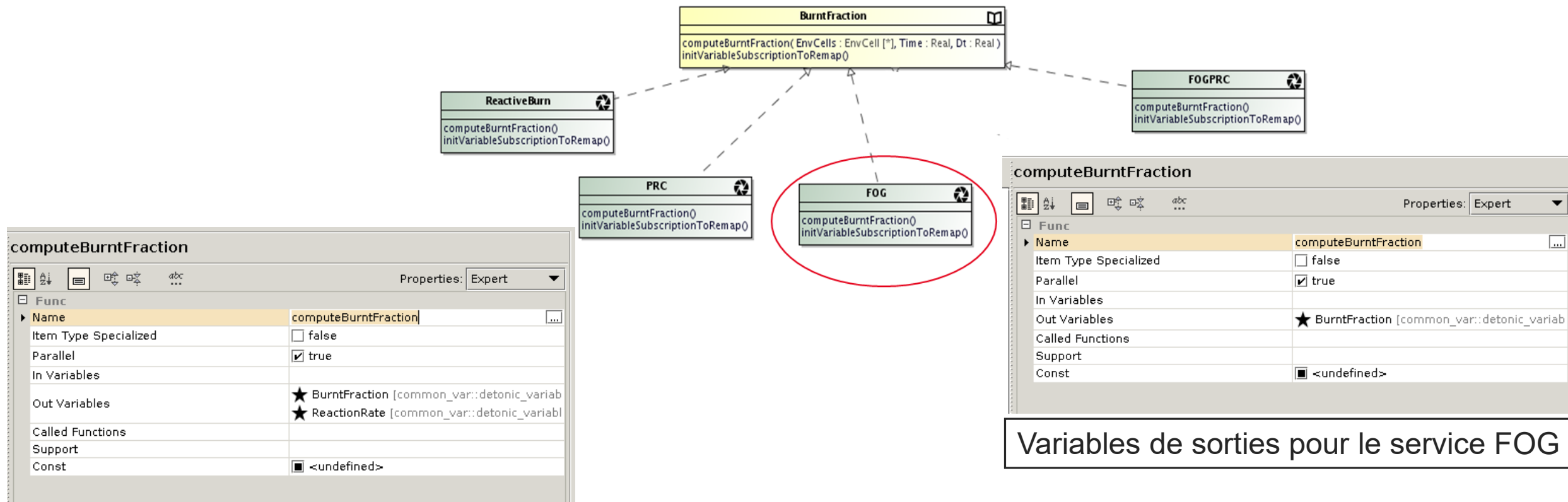


MODANE : Modélisation et génération de source



MODANE : Modélisation et génération de source

Génération de source



Variables de sorties pour le service PRC

Variables de sorties pour le service FOG

MODANE : Modélisation et génération de source

```
#ifndef DETONIC_BURNTFRACTION_FOGSERVICE_H
#define DETONIC_BURNTFRACTION_FOGSERVICE_H
```

```
/*-----*/
/*-----*/
#include "detonic_burntfraction/___FOGServiceBase.h"
#include "ProjectIncludes.h"
/*-----*/
/*-----*/
```

```
using namespace Arcane;
using namespace Arcane::Materials;
namespace Detonic_burntfraction {
```

```
/*-----*/
/*-----*/
/*!
```

```
* \brief
* Service detonic_burntfraction.FOG : implémentation
* Service calcul d'une fraction brûlée par une méthode géométrique.
*/
```

```
class FOGService
: public FOGServiceBase<FOGService>
{
public:
explicit FOGService(const ServiceBuildInfo& bi);
~FOGService();
```

```
public:
/*-----*/
/*-----*/
void computeBurntFraction(FOGComputeBurntFractionVars& vars, EnvCellVectorView env_cells,
const Real time, const Real dt);
/*-----*/
/*-----*/
void initVariableSubscriptionToRemap(FOGInitVariableSubscriptionToRemapVars& vars);
/*-----*/
/*-----*/
} /* namespace Detonic_burntfraction */
/*-----*/
/*-----*/
#endif /* DETONIC_BURNTFRACTION_FOGSERVICE_H */
```

FOGService.h

```
#include "detonic_burntfraction/FOGService.h"
```

```
using namespace Arcane;
using namespace Arcane::Materials;
namespace Detonic_burntfraction {
/*-----*/
/*-----*/
```

```
FOGService::
FOGService(const ServiceBuildInfo& bi)
: FOGServiceBase<FOGService>(bi)
{
```

```
/*-----*/
/*-----*/
```

```
FOGService::
~FOGService()
{
}
/*-----*/
void FOGService::
computeBurntFraction(FOGComputeBurntFractionVars& vars, EnvCellVectorView env_cells, const Real time,
[[maybe_unused]] const Real dt)
{
}
```

```
/*-----*/
/*-----*/
```

```
void FOGService::
initVariableSubscriptionToRemap([[maybe_unused]] FOGInitVariableSubscriptionToRemapVars& vars)
{
}
```

```
/*-----*/
/*-----*/
```

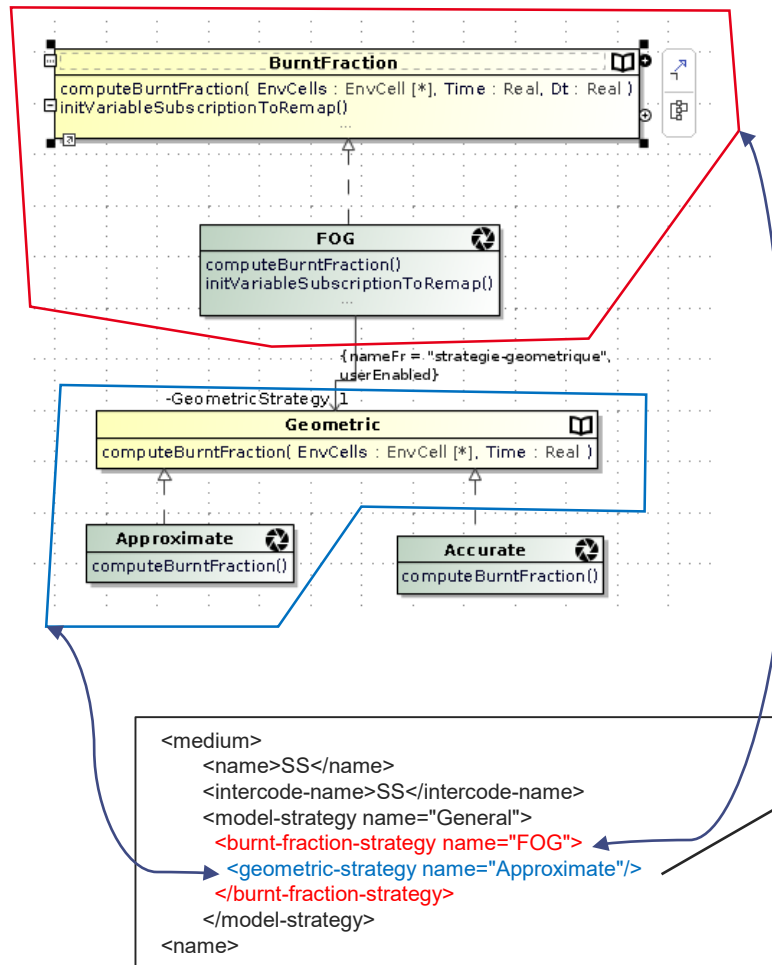
```
ARCANE_REGISTER_SERVICE_FOG(FOG, FOGService);
```

```
/*-----*/
/*-----*/
```

```
} /* namespace Detonic_burntfraction */
```

FOGService.cc

MODANE : Modélisation et génération de source



```
#include "detonic_burntfraction/FOGService.h"
```

```

using namespace Arcane;
using namespace Arcane::Materials;
namespace Detonic_burntfraction {
/*-----*/
/*-----*/
FOGService::
FOGService(const ServiceBuildInfo& bi)
: FOGServiceBase<FOGService>(bi)
{}

/*-----*/
/*-----*/
FOGService::
~FOGService()
{
}
/*-----*/
void FOGService::
computeBurntFraction(FOGComputeBurntFractionVars& vars, EnvCellVectorView env_cells, const Real time,
[[maybe_unused]] const Real dt)
{
...
  getGeometricStrategy()->computeBurntFraction(vars.m_abstract_node_coord, env_cells, time);
...
}
/*-----*/
/*-----*/
void FOGService::
initVariableSubscriptionToRemap([[maybe_unused]] FOGInitVariableSubscriptionToRemapVars& vars)
{
}

/*-----*/
/*-----*/

ARCANE_REGISTER_SERVICE_FOG(FOG, FOGService);

/*-----*/
/*-----*/

} /* namespace Detonic_burntfraction */
  
```

MODANE : Modélisation et génération de source

Bilan au bout de 10 ans

- Les +
 - Simplifie des révisions d'architecture, les modifications transverses à tout le code (ajout d'includes pour le GPU)
 - Aide à la compréhension (représentation graphique, documentation intégrée, visibilité I/O des fonctions)
 - Toujours cohérent avec les sources bien au-delà des phases de spécification du code
- Les -
 - Fichiers formatés, pas de synchronisation automatique en gestion de source, et *merge* magicDraw limité
 - Lourdeur du processus : nécessaire de passer par le modèle pour rajouter des variables en entrée sortie
 - Limite l'utilisation du C++ (pas d'héritage)
 - Besoin d'un support

Les perspectives

- MODANE V2, en cours d'expérimentation avec DSSI
 - Modélisation texte, avec représentation graphique a posteriori
 - Inclus dans *vscode*
- Réflexion sur l'utilisation à long terme

ODIN : Fichier de données



Contexte :

- Fichier de donnée Arcane reflète l'architecture du code
 - Comment changer l'architecture, sans impacter les jeux de données utilisateurs, et inversement ?
 - Choix de ne pas se brider sur le jeu de données, pour permettre des études numériques, tester différents couplages d'options → jdd très complet
 - Réticences des utilisateurs devant le format xml

Choix

- Jeu de donnée utilisateurs, format YAML
 - Formatage fort ; coloration syntaxique.
- Développement de ODIN, transcripteur de données
 - Vérification du jeu de données, de la compatibilité des options
 - Utilisation systématique de tests unitaires
 - ~ 350 000 lignes de python

ODIN : Fichier de données

```
60 DONNEES EULER:
61 ..Schema: .XXX V1
62 ..PasDeTemps:
63 ....CFL: .0.34
64 ..PseudoViscosite:
65 ....LongueurCaracteristique: .RacineCarreeDuVolume
66 ....CoeffGlobaux:
67 .....Lineaire: .0.3
68 .....Quadratique: .0.5
69 ..ConditionsBords:
70 ....ConditionBord:
71 .....Bord: .XMin
72 .....Nature: .Mur
73 .....Bord: .XMax
74 .....Nature: .Mur
75 .....Bord: .YMin
76 .....Nature: .Mur
77 .....Bord: .YMax
78 .....Nature: .Mur
79 ..Projection:
80 ....DirectionAlternee:
81 .....Methode: .Youngs
82 .....Ordre1: {}
```

propriétés générales
du schémas

Pas de temps

viscosité artificielle

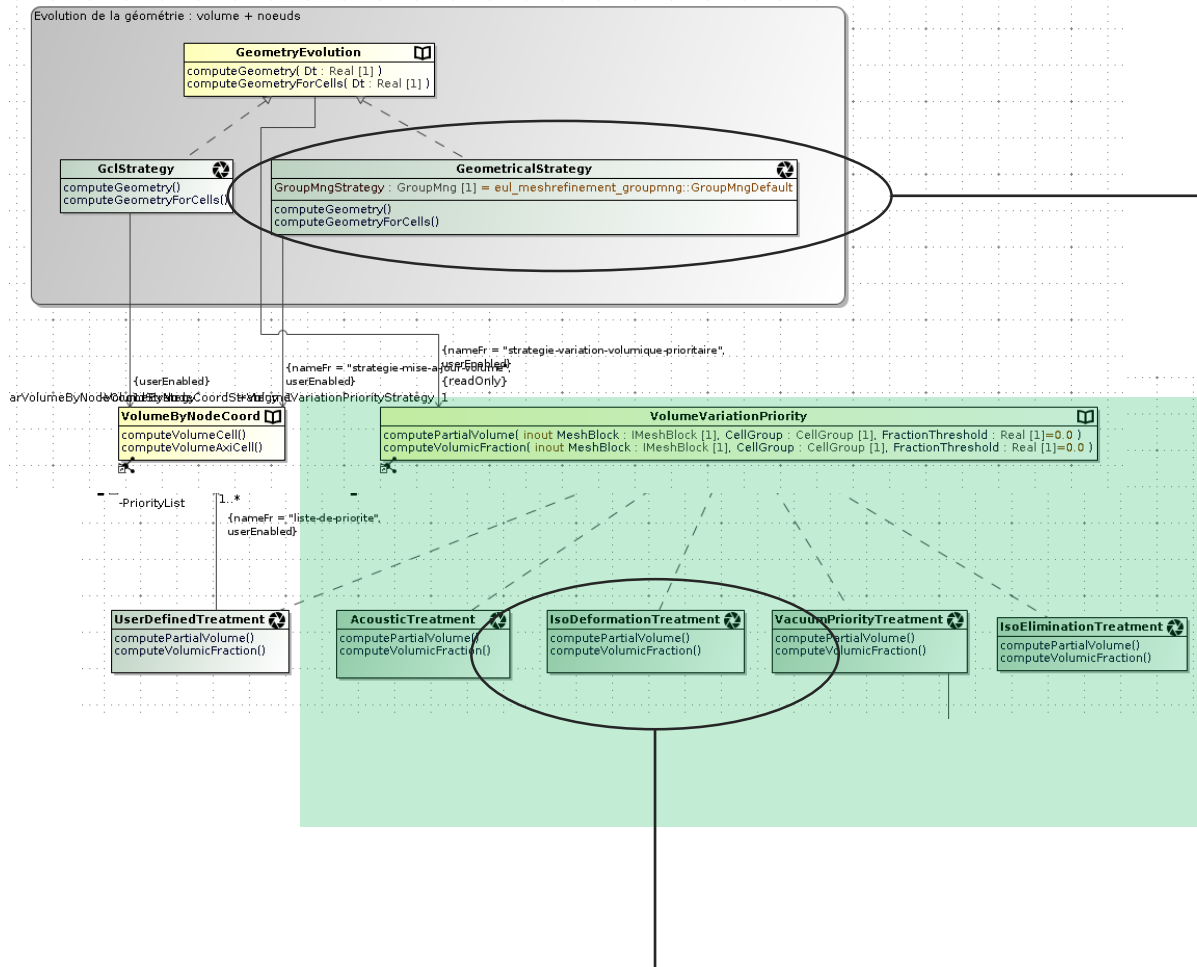
même variante de calcul
de la divergence de la vitesse

```
<eul-hydro-lag>
  <div-ufor-internal-energy-strategy name="StokesPlanarIntegrationPlanarDiv"/>
  <hydro-time-step-strategy name="Method1T">
    <cfl>0.34</cfl>
    <dt-max-rise-rate>101.0</dt-max-rise-rate>
  </hydro-time-step-strategy>
  <pseudo-viscosity-strategy name="ScalarPseudoViscosity">
    <div-ufor-pseudo-strategy name="StokesPlanarIntegrationPlanarDiv"/>
    <characteristic-length-strategy name="SqrtVolume"/>
    <pseudo-viscosity-formula-strategy name="GlobalPseudoViscosity"/>
    <coef-pseudo-viscosity>
      <linear-coefficient>0.3</linear-coefficient>
      <quadratic-coefficient>0.5</quadratic-coefficient>
    </coef-pseudo-viscosity>
    <partial-value-globalization-strategy name="PartialValueGlobalizationDefault">
      <sound-speed-ponderation-strategy name="MaximumValue"/>
      <partial-value-ponderation-strategy name="IsoDeformation"/>
    </partial-value-globalization-strategy>
    <cell-with-void-value-correction-strategy name="Global">
      <void-influence-threshold>1.0</void-influence-threshold>
    </cell-with-void-value-correction-strategy>
  </pseudo-viscosity-strategy>
  <geometry-evolution-strategy name="GeometricalStrategy">
    <volume-variation-priority-strategy name="IsoDeformationTreatment"/>
  </geometry-evolution-strategy>
  <partial-value-globalization-strategy name="PartialValueGlobalizationDefault">
    <sound-speed-ponderation-strategy name="MaximumValue"/>
    <partial-value-ponderation-strategy name="IsoDeformation"/>
  </partial-value-globalization-strategy>
  <cell-with-void-value-correction-strategy name="Global">
    <void-influence-threshold>1.0</void-influence-threshold>
  </cell-with-void-value-correction-strategy>
</eul-hydro-lag>
```

Fichier de données utilisateurs

Fichier de données Arcane

ODIN : Fichier de données



```

<eul-hydro-lag>
<div-ufor-internal-energy-strategy name="StokesPlanarIntegrationPlanarDiv"/>
<hydro-time-step-strategy name="Method1T">
  <cfl>0.34</cfl>
  <dt-max-rise-rate>101.0</dt-max-rise-rate>
</hydro-time-step-strategy>
<pseudo-viscosity-strategy name="ScalarPseudoViscosity">
  <div-ufor-pseudo-strategy name="StokesPlanarIntegrationPlanarDiv"/>
  <characteristic-length-strategy name="SqrtVolume"/>
  <pseudo-viscosity-formula-strategy name="GlobalPseudoViscosity"/>
  <coef-pseudo-viscosity>
    <linear-coefficient>0.3</linear-coefficient>
    <quadratic-coefficient>0.5</quadratic-coefficient>
  </coef-pseudo-viscosity>
  <partial-value-globalization-strategy name="PartialValueGlobalizationDefault">
    <sound-speed-ponderation-strategy name="MaximumValue"/>
    <partial-value-ponderation-strategy name="IsoDeformation"/>
  </partial-value-globalization-strategy>
  <cell-with-void-value-correction-strategy name="Global">
    <void-influence-threshold>1.0</void-influence-threshold>
  </cell-with-void-value-correction-strategy>
</pseudo-viscosity-strategy>
<geometry-evolution-strategy name="GeometricalStrategy">
  <volume-variation-priority-strategy name="IsoDeformationTreatment"/>
</geometry-evolution-strategy>
<partial-value-globalization-strategy name="PartialValueGlobalizationDefault">
  <sound-speed-ponderation-strategy name="MaximumValue"/>
  <partial-value-ponderation-strategy name="IsoDeformation"/>
</partial-value-globalization-strategy>
<cell-with-void-value-correction-strategy name="Global">
  <void-influence-threshold>1.0</void-influence-threshold>
</cell-with-void-value-correction-strategy>
</eul-hydro-lag>
  
```

Processus de validation

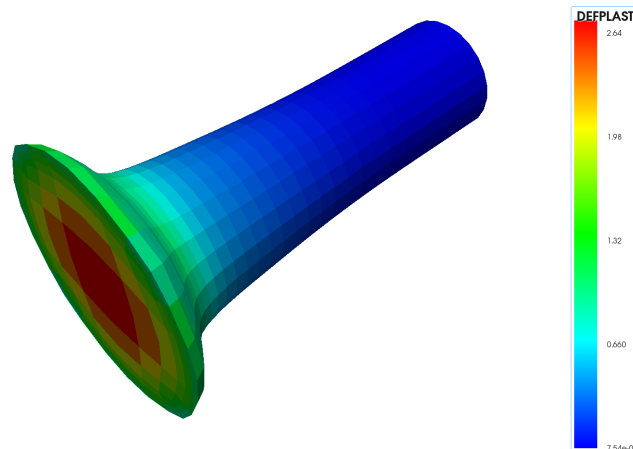
Réécriture de code : retrouver les résultats de l'ancien code est non négociable !

cas test standard : impact de Taylor

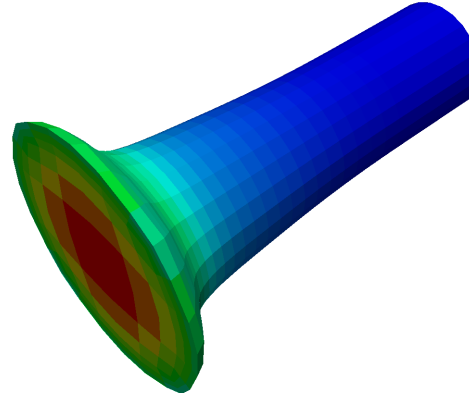
- Vitesse d'impact : 1000 m/s

Première étape : comparaisons visuelles

Ancien code



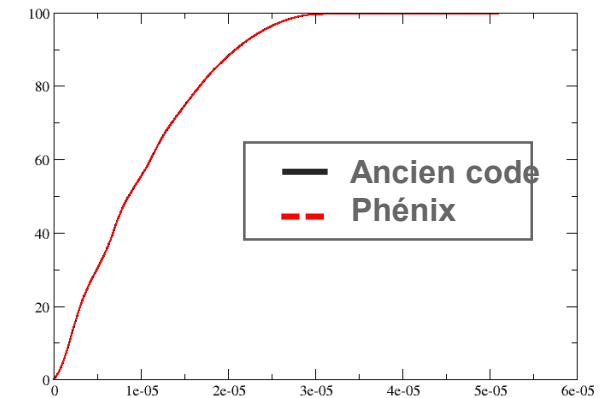
Phénix



Déformation plastique équivalente

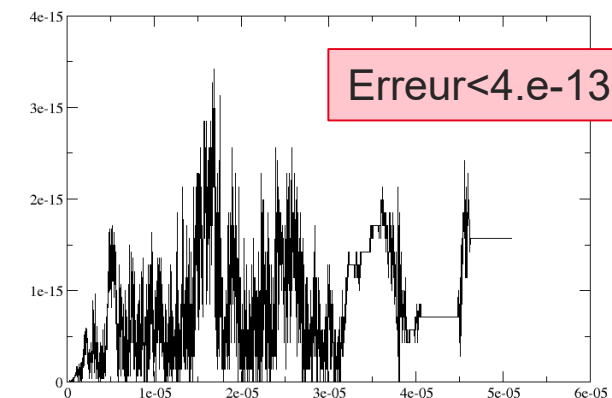
Deuxième étape : comparaisons de bilans

Déformation plastique équivalente



Troisième étape : mesure d'erreurs

Erreur sur la déformation plastique



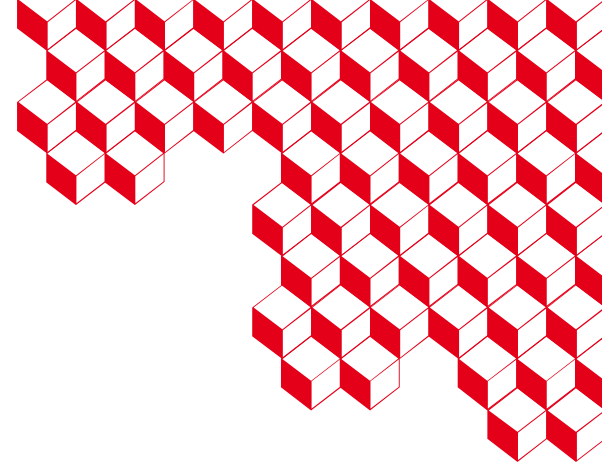
Conclusion et perspectives

HPC

- Portage GPU
 - Comment le concilier avec l'utilisation en production ?
 - Test de l'API GPU
- Exploration du parallélisme hybride proposé par Arcane
- Evaluation de l'équilibrage de charge
 - Avec contrainte de découpage cartésien du domaine dans le partitionneur Zoltan (DSSI)
- Poursuite du développement de la version AMR patch-based

Développement physico-numériques

- Poursuite de l'enrichissement du code pour répondre aux besoins des utilisateurs
- Prise en compte via Arcane de stencils élargis pour les nouveaux schémas numériques
- Développements de maquettes sous Arcane pour faciliter l'intégration d'expérimentations



Merci de votre attention

**Commissariat à l'énergie atomique et aux
énergies alternatives**

Centre DAM Ile de France Bruyères-le-Châtel

91297 Arpajon Cedex, France

thierry.pougear-dulimbert@cea.fr

Standard. + 33 1 69 26 40 00