



# RENCONTRES ARCANE

## 3<sup>ÈME</sup> ÉDITION

---

**17 AVRIL 2023**





## ● Evolution d'Arcane au cours de l'avenant n°6 (2021-2024)

- Développer une **application Arcane**
- Développer une **application modulaire**
- Développer une **application performante**

## ● Feuille de route produits Open Source

- Alien
- Les collaborations
- Documentation

## ● Travaux futurs

- Deuxième moitié avenant Arcane 2023-2024
- Pistes de réflexion futur avenant
- PEPR NumPEX



## ● Evolution d'Arcane au cours de l'avenant n°6 (2021-2024)

### ● Développer une **application Arcane**

- Mise en Open Source : Septembre 2021
- Développement de mini-app/démonstrateurs
  - ShArc
  - QAMA (Quick-Silver)
  - FemTest1

### ● Développer une **application modulaire**

- Nouveaux maillages
- Utilisation directe d'Arcane

### ● Développer une **application performante**

- API GPU



### ● Développer une **application Arcane**

- Mise en Open Source : Septembre 2021 (ouverture publique du dépôt GitHub)
- Mise en place d'un CI complet pour toute la chaîne
  - Les *Pull requests* sont testées avant le merge
- Mise en place de docker représentatifs de nos environnements de travail
  - Centos 7, Rocky 8
  - Ubuntu 20 et 22
  - Windows 10
- Utilisation de ressources gratuites uniquement (pour l'instant ;- )
  - Ressources GitHub et CircleCI
  - L'accès aux GPU (payant) pourrait être intéressant
- Suivi de la qualité des développements
  - Codacy et Coverity (analyse statique de code)
  - Codecov (couverture de test)

# EVOLUTION D'ARCANÉ AU COURS DE L'AVENANT N°6



Rencontres Arcane

17 avril 2023



arcaneframework / framework Public

Edit Pins Watch 6 Fork 9 Starred 22

<> Code Issues 35 Pull requests 3 Discussions Actions Projects Wiki Security Insights Settings

Actions

New workflow

All workflows

Codecov Scan

compile-all-vcpkg

Coverity Scan

Create Release Arcane

Create Release Arcon

Create Release Arccore

Create Release Axlstar

Header/Encoding checker

IFPEN devenv 2021b

IFPEN el7 foss/2018b

Show more workflows...

All workflows

Showing runs from all workflows

Filter workflow runs

Tell us how to make GitHub Actions work better for you with three quick questions.

Give feedback

12,607 workflow runs

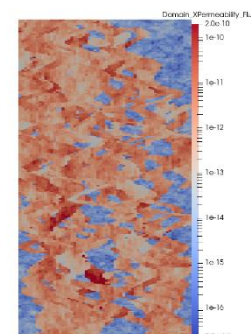
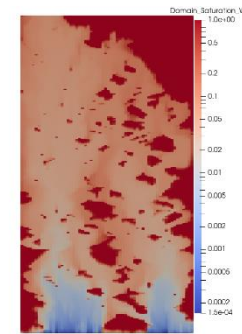
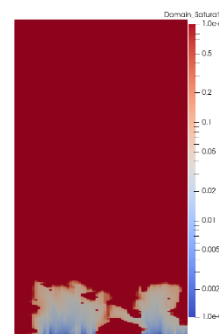
	Event	Status	Branch	Actor
✓ Merge pull request #613 from arcaneframework/dev/gg-add-helper-class-... IFPEN el8 foss/2018b #1056: Commit dda6434 pushed by grospellierrilles	main	3 hours ago 51m 37s		...
✓ Merge pull request #613 from arcaneframework/dev/gg-add-helper-class-... IFPEN el7 foss/2018b #1057: Commit dda6434 pushed by grospellierrilles	main	3 hours ago 48m 34s		...
✓ Merge pull request #613 from arcaneframework/dev/gg-add-helper-class-... IFPEN el8 gimkl/2018b #1910: Commit dda6434 pushed by grospellierrilles	main	3 hours ago 1h 8m 45s		...

5

| © 2017 IFPEN – Direction R11 – Troisième édition des Rencontres Arcane – 17 avril 2023



- Mini-Application/Démonstrateur : ShArc
- Qu'est-ce que ShArc ?
  - Une appli de simulation d'un écoulement en milieu poreux diphasique
  - Résolu par des outils numériques classiques :
    - Discrétisation Fully-implicit
    - Schémas volume finis 2-points
    - Solveur de Newton (ArcGeoSim)
    - Alien pour les solveurs linéaires
- Sur quel outillage s'appuie-t-il ?
  - Gump : un outil de génération de modèle de données d'un système physique
  - Law : un outil de génération pour l'évaluation des lois physiques
  - Audi : un outil de différenciation automatique
- Ces outils sont inclus dans la bibliothèque partagée ArcGeoSim







Outillage de ShArc – couplage des outils

Tous ces outils concourent au montage de la jacobienne du problème

```
const Law::Contribution grad_kl = T[iface] * (P[cell_k] - P[cell_l]);
ENUMERATE_PHASE(iphase, fluid.phases()) {
    const Law::Contribution mobility_k = rho[iphase][cell_k] * kr[iphase][cell_k] / mu[iphase][cell_k];
    const Law::Contribution mobility_l = rho[iphase][cell_l] * kr[iphase][cell_l] / mu[iphase][cell_l];

    const Law::Contribution flux_kl =
        ( audi::one(grad_kl >= 0) * mobility_k
        + audi::one(grad_kl < 0) * mobility_l ) * grad_kl;
```

Calcul du flux (Gump + Lois + Audi)

```
const Arcane::Integer iequation = iphase.index();
if (cell_k.isOwn()) {
    residual[iequation][cell_k] += flux_kl;
    jacobian[iequation][cell_k][stencil] += flux_kl;
}
if (cell_l.isOwn()) {
    residual[iequation][cell_l] -= flux_kl;
    jacobian[iequation][cell_l][stencil] -= flux_kl;
}
```

Ajout du flux au résidu et à la jacobienne

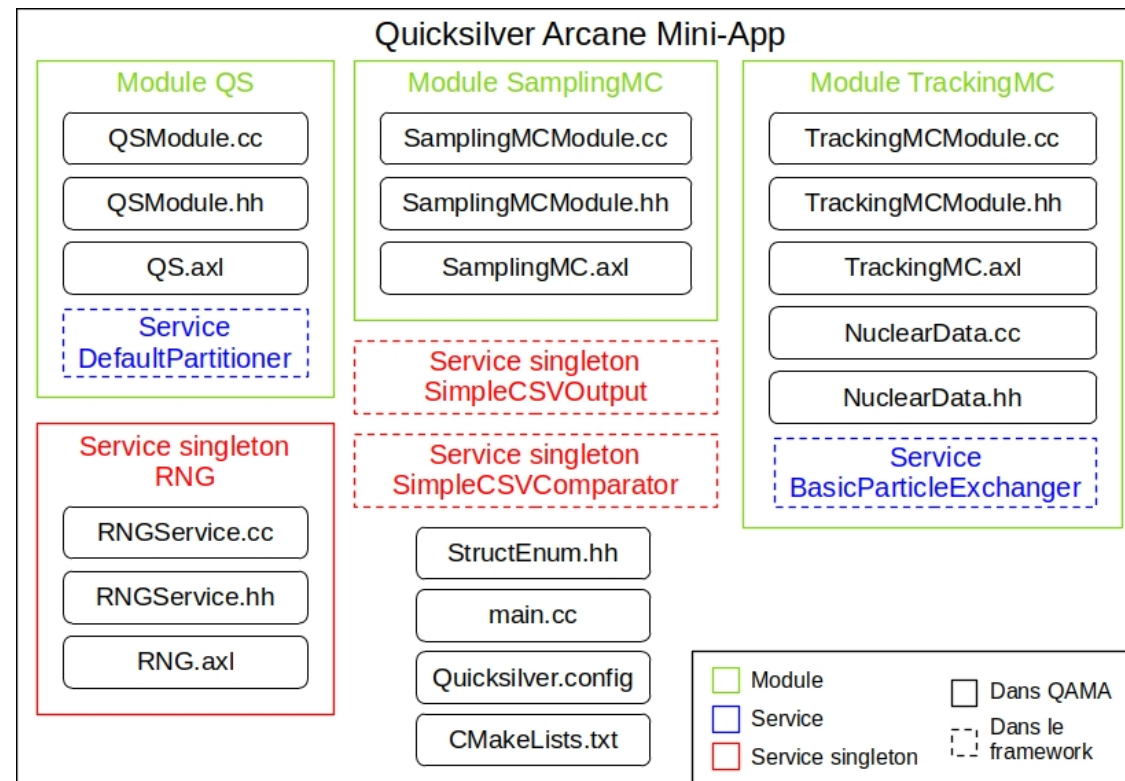


- Mini-Application/Démonstrateur : QuickSilver Arcane Mini-App (QAMA)
  - Simulation de transport de particule Monte-Carlo simplifiée
  - Portage de QuickSilver (LLNL) (<https://github.com/LLNL/Quicksilver>)
  - Représente les éléments principaux du code Mercury (gros code Monte-Carlo du LLNL)
- Disponible sur GitHub : <https://github.com/arcaneframework/arcane-benchs>



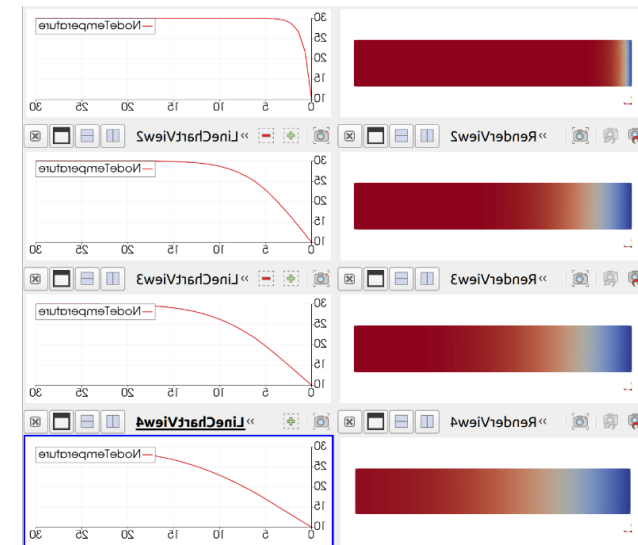
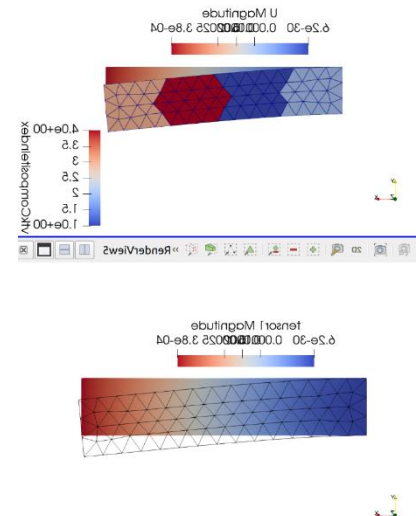
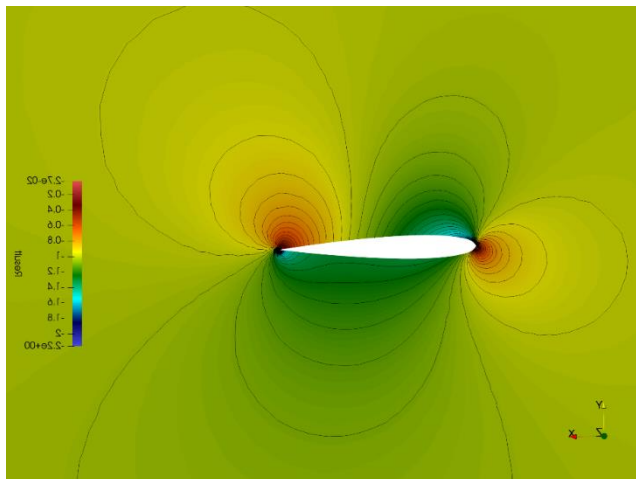


### ● Mini-Application/Démonstrateur : QuickSilver Arcane Mini-App (QAMA)





- Mini-Application/Démonstrateur : **FEMTEST1** (<https://github.com/arcaneframework/femtest1>)
- Collaboration avec le CEA/DES (Direction des Énergies)
  - Démonstrateur pour ajouter des mécanismes pour gérer les éléments finis dans Arcane
  - Plusieurs exemples: calcul de structures, aérodynamique, équation de la chaleur
  - A terme servira de base pour un code de calcul de structures pour simuler l'impact des séismes sur les bâtiments





- Mini-Application/Démonstrateur : **Athena**
- Code interne CEA/DAM de transport de particules
  - Neutronique déterministe
  - Implémentation MPI/TBB ou MPI/GPU
- P.Anguill, P.Cargo, C.Enaux, P.Hoch, E.Labourasse, G.Samba : **An asymptotic preserving method for the linear transport equation on general meshes**, Journal of Computational Physics 450 (2022) 110859

## EVOLUTION D'ARCANE AU COURS DE L'AVENANT N°6

- Développer une **application modulaire**
  - Nouveaux maillages
  - Utilisation directe d'Arcane



**Rencontres Arcane**

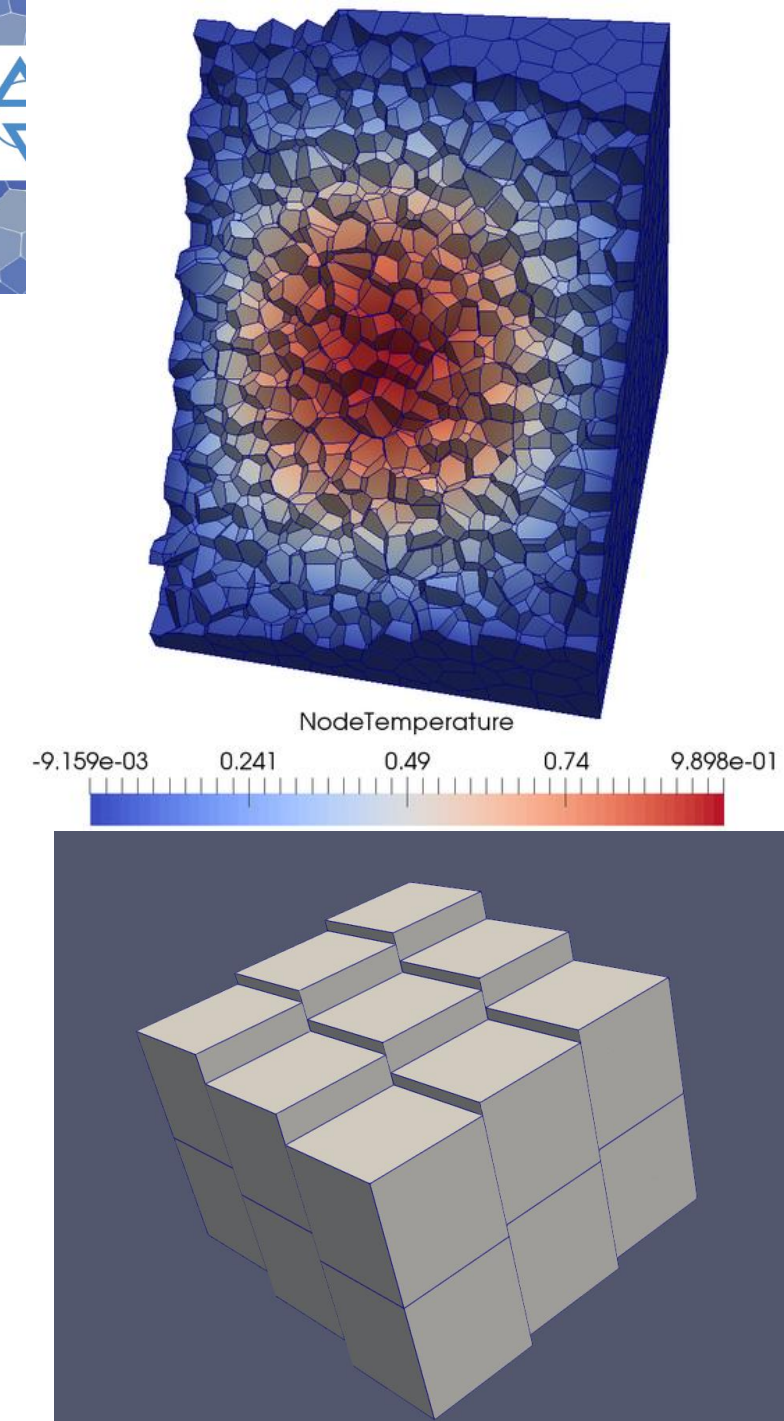
17 avril 2023



## EVOLUTION D'ARCANE AU COURS DE L'AVENANT N°6



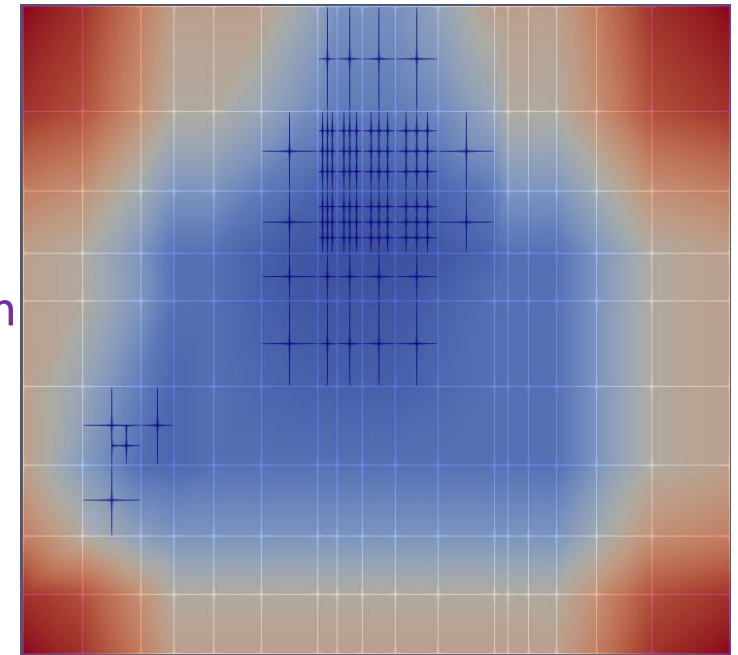
- Nouveaux maillages : **polyédrique**
- Import format Vtk, export Ensign, en cours de finalisation
- Quelques maillages de tests
  - « Vrai » polyédriques
  - Maillage mis en conformité
- Représentatifs des deux types de use case en géosciences
  - Génération pour mieux prendre en compte **géométries complexes**
    - Thèse Alexandre Marin (Laurent Astart– Alexandra Bac)
  - Mise en **conformité** de maillages CPG (raccord sur des failles)







- Nouveaux maillages : AMR cartésien par patch
- Enrichi la notion de maillage cartésien en ajoutant la notion de Patch
  - Un patch est une zone rectangulaire du maillage sur laquelle on raffine les mailles
  - Plusieurs patches sont possibles
- Utilise les fonctionnalités AMR disponibles dans Arcane
- Évolutions planifiées
  - Structure de données spécifique pour l'AMR cartésien par patch
  - Optimisation mémoire
  - Duplication des nœuds et faces dans les patches
  - Numérotation des entités optimisé pour le cartésien







- Utilisation directe d'Arcane
  - API GPU utilisable directement
  - Possibilité d'utiliser Arcane
    - sans passer par la boucle en temps
    - sans utiliser de jeu de données
- Plusieurs exemples dans le code source
  - mini\_nbody
  - direct\_cartesian
  - standalone\_accelerator



- Développer une **application performante**
  - Ajout d'une API pour cibler les accélérateurs
    - Implémentation actuelle avec CUDA
    - Boucle de calcul unifiée entre séquentiel, multi-thread et accélérateur
    - Un seul exécutable avec la possibilité de choisir lors de l'exécution le support d'exécution
  - Démonstrateur MaHyCo (<https://github.com/cea-hpc/MaHyCo>)
- Développements planifiés
  - Support des matériaux/milieus et des maillages cartésiens et AMR par patch
  - Variables multi-dimensionnelles avec agencement spécifique (layout)
  - Optimisations diverses
  - Support d'autres API (ROCm ou oneAPI par exemple)

# ROADMAP ET POINT SUR L'OPEN SOURCE



**Rencontres Arcane**  
17 avril 2023



## ● Feuille de route produits Open Source

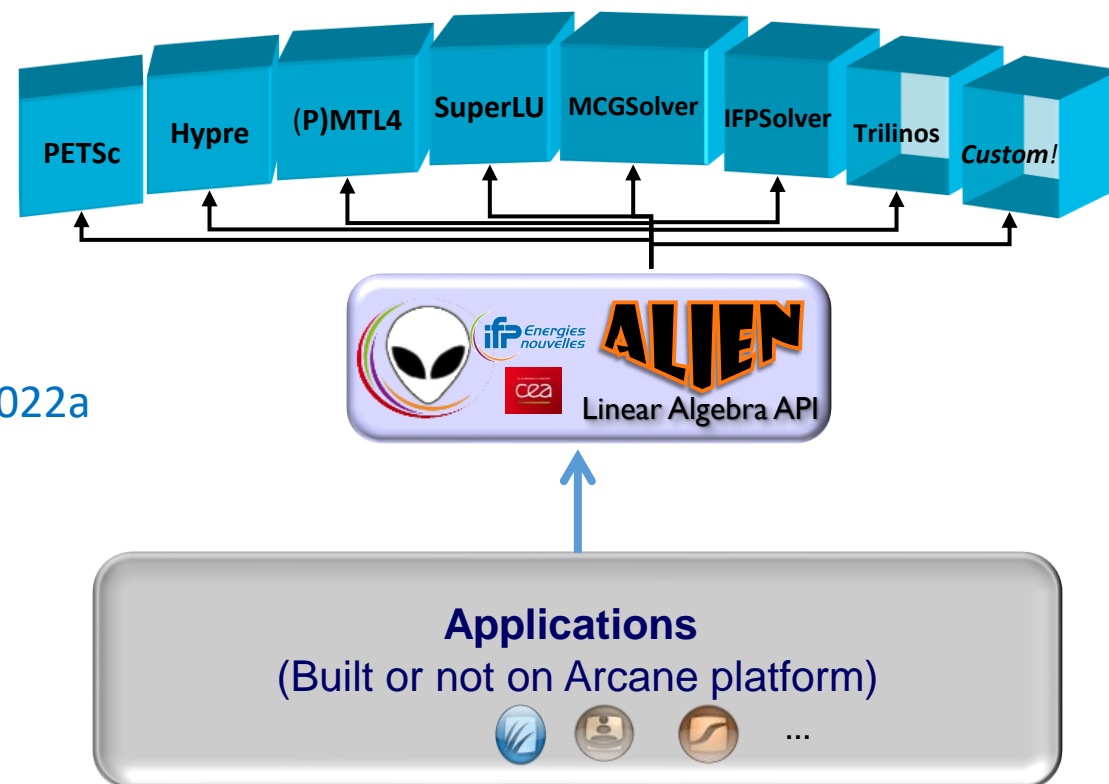
- Alien
- Les collaborations
- Documentation



## ● Alien

- Plusieurs tests de solveurs GPU
  - Branchements Ginkgo
  - Branchement Trilinos
  - Branchement Hypre GPU
- Communication et publication à Ecomas 2022
  - Présentation d'Alien et benchmark de solveurs
  - [https://www.scipedia.com/public/Gratien\\_et\\_al\\_2022a](https://www.scipedia.com/public/Gratien_et_al_2022a)
- Travaux d'amélioration de la redistribution

### Solver Librairies (including IFPEN IFPSolver & MCG HPC solver)





## ● Les collaborations

### ● Andra

- Alien : branchement dans le code **Traces** (transport radionucléides en milieu poreux)
- Couplage de codes (branchement de **preCICE** dans un code Arcane)

### ● CEA/DES

#### ● FemTest1

- Enrichir le support des éléments finis dans Arcane
- Réaliser une composante utilisable par d'autres applications : ArcaneFEM.

- Prototype ArcaneSnSolver (voir A. Callo) : neutronique déterministe sur maillage en nid d'abeille

### ● Présentation d'Arcane/Alien

- INRIA
- ONERA
- BRGM



## ● Documentation

- Grande mise à jour publiée en Octobre 2022
  - Toujours générée grâce à Doxygen
  - Mise à jour des pages et réorganisation des pages en chapitres
  - Passage en 100 % Markdown
  - Thème « Doxygen Awesome » (disponible sur GitHub) avec plusieurs ajouts
  - Ajout de la possibilité, pour les utilisateurs, de personnaliser la documentation via une page dédiée
- Disponible ici : <https://arcaneframework.github.io>





## Arcane 3.7.6.0

Page principale

Pages associées

Modules

Espaces de nommage

Classes

Fichiers

Recherche

Arcane

Nouvelles fonctionnalités

Vue d'ensemble de Arcane

Les types fondamentaux

Les modules

Les variables

Les points d'entrée

Les services

Le didacticiel ARCANE

Les règles de codage

Lancement d'un calcul

Configuration des options jeu de données (fichier AXL)

La configuration du code

Jeu de données

La boucle en temps

Prise en compte du parallélisme dans ARCANE

Concurrence et multi-threading

Vectorisation

Utilisation des traces

Utilisation des accélérateurs

Utilisation du C# avec Arcane

Le support des tests unitaires dans Arcane

Gestion des matériaux et des milieux.

Boucles sur les entités des matériaux et des milieux

Gestion des maillages cartésiens

Gestion de la géométrie.

Les types de mailles

Variables d'environnement

Détection des problèmes mémoire.

Comparaison bit à bit

Analyse de performances (<em>Profiling</em>)

Utilisation des tableaux

Les types fondamentaux

Il existe 4 types fondamentaux dans Arcane, qui correspondent aux notions de *Module*, *Variable*, *Point d'entrée* et *Service*, .

Pour une description sommaire de ces notions, se reporter à la section [vue d'ensemble d'ARCANE](#).

- Module
- Variable
- Point d'entrée
- Service

Les modules

Un module est un ensemble de *points d'entrée* et de *variables*. Il peut posséder des options de configuration qui permettent à l'utilisateur de paramétrer le module via le jeu de données de la simulation.

Un module est représenté par une classe et un fichier XML appelé *descripteur de module*.

Le descripteur de module

Le descripteur de module est un fichier XML ayant l'extension ".axl". Il présente les caractéristiques du module :

- ses variables,
- ses points d'entrée,
- ses options de configuration.

```
<?xml version="1.0"?>
<module name="Test" version="1.0">
  <name lang="fr">Test</name>

  <description>Descripteur du module Test</description>

  <variables/>

  <entry-points/>

  <options/>
</module>
```


Table des matières

- Les modules
  - Le descripteur de module
  - La classe représentant le module
  - Connexion du module à Arcane
- Les variables
  - Le type
  - Le support
  - La dimension
  - Classe C++
  - Déclaration
  - Utilisation
- Les points d'entrée
  - Construction
- Les services
  - Le descripteur de service
  - Connexion du service à Arcane

%Arcane

Généré le Lundi 12 Septembre 2022 04:37:52 pour Arcane par doxygen 1.9.1





**Arcane** v3.10.0.0  
Documentation utilisateur

Arcane

- Nouveautés et changements
- Personnalisation de la doc
- Débuter avec Arcane
- Types fondamentaux
  - Module**
  - Service
  - Descripteur de module/service (.AXL)
  - Jeu de données (.ARC)
  - Configuration du code (.CONFIG)
  - Boucle en temps
  - Tableaux
  - Utilisation de la classe NumArray
- Exécution, runtime
- Généralités
- Exemples et didacticiel
- Accélérer un code
- Matériaux et milieux
- Debug, performance et validation
- Utilisation d'autres langages
- Entités, maillages et connectivités
- Services et modules disponibles
- Liste des éléments obsolètes
- Modules
- Espaces de nommage

## Module

Un module est un ensemble de **points d'entrée** et de **variables**. Il peut posséder des options de configuration qui permettent à l'utilisateur de paramétrer le module via le jeu de données de la simulation.

Un module est représenté par une classe et un fichier XML appelé **descripteur de module**.

## Descripteur de module

Le descripteur de module est un fichier XML ayant l'extension ".axl". Il présente les caractéristiques du module :

- ses variables,
- ses points d'entrée,
- ses options de configuration.

```
1 <?xml version="1.0"?>
2 <module name="Hydro" version="1.0">
3   <name lang="fr">Hydro</name>
4
5   <description>Descripteur du module Hydro</description>
6
7   <variables>
8   </variables>
9
10  <entry-points>
11  </entry-points>
12
13  <options>
14    <!-- Service de type IEquationOfState. -->
15  </options>
16 </module>
```

Par exemple, le fichier `Hydro.axl` ci-dessus présente le module nommé **Hydro** dont la classe de base, est `HydroModule` (cas général). **TODO: ajouter doc référence sur les autres attributs du module.**

Les variables, les points d'entrée et les options seront décrits dans le chapitre [Descripteur de module/service \(.AXL\)](#).

## Classe représentant le module

### Table des matières

[Descripteur de module](#)

Classe représentant le module

Connexion du module à Arcane

# Feuille de route produits Open Source



Rencontres Arcane

17 avril 2023



```
File Edit Selection View Go Run ... 0_getting_started.md -- framework -- I'm not Vim, I'm Visual S...
SimpleTableWriterHelper.cc SimpleTableWriterHelper.h SimpleTableInternalMng.cc 0_getting_started.md x
arcane > doc > doc_user > chap_getting_started > 0_getting_started.md > # Débuter avec %Arcane (#arcanedoc_getting_started)
Hugo Taboada, 3 months ago | 2 authors (You and others)
1 # Débuter avec %Arcane (#arcanedoc_getting_started)
2
3 Vous débutez sur %Arcane ? Ce chapitre devrait vous donner les notions de base.
4
5 <br>
6
7 Sommaire de ce chapitre :
8
9 1. \subpage arcanedoc_getting_started_about <br>
10 | Description de la plateforme %Arcane.
11
12 2. \subpage arcanedoc_getting_started_basicstruct <br>
13 | Présente les structures et les types de base de %Arcane.
14
15 3. \subpage arcanedoc_getting_started_iteration <br>
16 | Présente les différentes façons d'itérer dans un code %Arcane.
17
18
19 ----
20
21
22 <div class="section_buttons">
23 <span class="next_section_button">
24 \ref arcanedoc_getting_started_about
25 </span>
26 </div>
27
```



## Débuter avec Arcane

Vous débutez sur Arcane ? Ce chapitre devrait vous donner les notions de base.

Sommaire de ce chapitre :

1. Qu'est-ce que Arcane ?  
Description de la plateforme Arcane.
2. Structures et types de base  
Présente les structures et les types de base de Arcane.
3. Itération  
Présente les différentes façons d'itérer dans un code Arcane.

Qu'est-ce que Arcane ? >

# Feuille de route produits Open Source



Rencontres Arcane

17 avril 2023



## Note

Les valeurs de l'instance ne sont pas initialisées lors de la construction. Il faut appeler la méthode `Arcane::NumArray` accessible depuis l'hôte) si on souhaite remplir le tableau avec une valeur donnée.

Il est possible de spécifier lors de la construction ou avec le méthode `Arcane::NumArray::resize()` le nombre d'éléments d'arguments correspond au nombre de dimensions dynamiques de l'instance :

```
1 // Tableau 2D avec 2 dimensions statiques (3x4)
2 Arcane::NumArray<double, Arcane::ExtentsV<3,4>> t4;
3 // Tableau 2D avec 2 dimensions dynamiques
4 Arcane::NumArray<double, Arcane::ExtentsV<2,2>> t5(8);
5 // Tableau 2D avec 2 dimensions dynamiques
6 Arcane::NumArray<double, Arcane::ExtentsV<2,2>> t6(3,9);
7 // Redimensionne t2 avec 6x5 valeurs
8 t2.resize(6);
9 // Redimensionne t3 avec 2x7x4 valeurs
10 t3.resize(2,4);
11 // Redimensionne a1 avec 12 valeurs
12 a1.resize(12);
13 // Redimensionne a2 avec 3x4 valeurs
14 a2.resize(3,4);
15 // Redimensionne a3 avec 2x7x6 valeurs
16 a3.resize(2,7,6);
17 // Redimensionne a1 avec 2x9x4x6 valeurs
18 a4.resize(2,9,4,6);
```

**Arcane::NumArray**  
Tableaux multi-dimensionnels pour les types numériques accessible sur accélérateurs.  
Definition: utils/NumArray.h:619

## Avertissement

Le redimensionnement ne conserve pas les valeurs actuelles du tableau


() (uniquement si la mémoire est

ue dimension. Dans ce cas le nombre



Copier le code dans le presse-papier





**Arcane v3.10.0.0**  
Documentation utilisateur

Recherche

- Variable
- Parallel
- Xml
- StandardService
- IO
- Math
- Vectorisation
- Matériaux et des milieux.
- Maillages cartésiens.
  - Arcane::CartesianConnectivity
  - Arcane::DirCell
  - Arcane::DirCellLocalId
  - Arcane::DirCellNode
  - Arcane::DirCellNodeLocalId
  - Arcane::DirCellFace
  - Arcane::DirCellFaceLocalId
  - Arcane::CellDirectionMng
  - Arcane::DirFace**
    - nextCell
    - nextCellId
    - previousCell
    - previousCellId
  - Arcane::DirFaceLocalId
  - Arcane::FaceDirectionMng
  - Arcane::ICartesianMesh
  - Arcane::ICartesianMeshPatch
  - Arcane::DirNode
  - Arcane::DirNodeLocalId
  - Arcane::NodeDirectionMng
- Eléments géométriques.
- Espaces de nommage
- Classes

## Référence de la classe Arcane::DirFace

Maillages cartésiens.

Infos sur maille avant et après une face suivant une direction. [Plus de détails...](#)

```
#include <arcane/cartesianmesh/FaceDirectionMng.h>
```

» Graphe de collaboration de Arcane::DirFace:

### Fonctions membres publiques

`Cell` `previousCell () const`  
Maille avant.

`CellLocalId` `previousCellId () const`  
Maille avant.

`Cell` `nextCell () const`  
Maille après.

`CellLocalId` `nextCellId () const`  
Maille après.

### Description détaillée

Infos sur maille avant et après une face suivant une direction.

Les instances de cette classe sont temporaires et construites via `FaceDirectionMng::face()`.

Définition à la ligne 40 du fichier `cartesianmesh/FaceDirectionMng.h`.

### Documentation des fonctions membres

`nextCell()`

`Cell` `Arcane::DirFace::nextCell () const`  
Maille après.  
Définition à la ligne 55 du fichier `cartesianmesh/FaceDirectionMng.h`.

Fonctions membres publiques | Liste de tous les membres

Généré le Lundi 10 Avril 2023 02:26:46 pour Arcane par [doxygen](#) 1.9.6



# Feuille de route produits Open Source



Rencontres Arcane  
17 avril 2023



**Arcane 3.7.12.0**

Page principale | Pages association | Modules | Espaces de nommage | Cléver | Fichiers | Recherche

**'arcane-post-traitement' module**

Options concernant les **sorties** pour le dépouillement.  
Les sorties peuvent être effectués en fonction du nombre d'itérations (options `periode-sortie`) ou en fonction du temps physique (option `frequence-sortie`). Par exemple:  
`<periode-sortie>5</periode-sortie>`

**Summary of options**

- **compression-sortie-historique**
- **dépouillement** (2 options)
  - **groupe**
  - **variable**
- **format-service**
- **frequence-sortie**
- **periode-sortie**
- **periode-sortie-historique**
- **sauvegarde-initiale**
- **sauvegarde-temps-final**
- **sortie-fin-execution**

**Detailed list of options**

**compression-sortie-historique** (bool=false)

`<compression-sortie-historique>value</compression-sortie-historique>`

Evite de sauvegarder tous les points d'abscisse entre les périodes de l'option `output-history-period`. Cela permet de faire baisser la consommation mémoire notamment lorsque le code effectue un grand nombre d'itérations.

**dépouillement**

Liste des variables et des groupes à dépouiller

**dépouillement/groupe** [0... undefined] (Arcane::ItemGroup)

`<groupe>value</groupe>`

Groupe d'entité de maillage sur lequel on souhaite exporter des variables. Il est possible de mettre plusieurs options de ce type pour exporter les variables sur plusieurs groupes. Si cette option n'est pas spécifiée, les variables sont sorties sur l'ensemble du maillage.

**dépouillement/variable** [0... undefined] (usttring)

`<variable>value</variable>`

Nom d'une variable à exporter. Il est possible de mettre plusieurs options de ce type pour exporter plusieurs variables. Si aucune variable n'est spécifiée, aucune sortie n'est effectuée.

Liste des modules et services du jeu de données

- 'arcane-equilibrage-charge' module
- 'arcane-post-traitement' module
- 'arcane-protections-reprises' module
- 'module-test-unitaire' module
- 'verificateur' module
- 'ArcaneCaseMeshMasterService' service
- 'ArcaneCaseMeshService' service
- 'ArcaneCasePartitioner' service
- 'ArcaneCaseSolverExecution' service
- 'ArcaneHdf5Checkpoint2' service
- 'ArcaneHdf5SmpCheckpoint2' service
- 'ArcaneMeshConverter' service
- 'ArcaneMeshExchange' service
- 'Cartesian2D' service
- 'Cartesian3D' service
- 'DefaultPartitioner' service
- 'EnlightPostProcessor' service
- 'Hdf5VariableReader' service
- 'Hdf5VariableWriter' service
- 'HoneyComb2D' service
- 'HoneyComb3D' service
- 'Mets' service
- 'Mets' service
- 'PDESRandomNumberGenerator' service
- 'SimpleCivComparator' service
- 'SimpleCivOutput' service

**ARCANE FRAMEWORK**

Arcane v3.7.12.0  
Documentation utilisateur

Recherche

Généralités  
Exemples et didacticiel  
Accélérer un code  
Matériaux et milieux  
Debug, performance et validation  
Utilisation d'autres langages  
Entités, maillages et connectivités  
Services et modules disponibles

Liste des modules et services du jeu de données

- Liste des modules
- Liste des services
- Module 'arcane-equilibrage-charge'
- **Module 'arcane-post-traitement'**
- Liste des options
- Documentation des options
- Module 'arcane-protections-reprises'
- Module 'module-test-unitaire'
- Module 'verificateur'
- Service 'ArcaneCaseMeshMasterService'
- Service 'ArcaneCaseMeshService'
- Service 'ArcaneCasePartitioner'
- Service 'ArcaneCaseSolverExecution'
- Service 'ArcaneHdf5Checkpoint2'
- Service 'ArcaneHdf5SmpCheckpoint2'
- Service 'ArcaneMeshConverter'
- Service 'BasicParticleExchanger'
- Service 'Cartesian2D'
- Service 'Cartesian3D'
- Service 'DefaultPartitioner'
- Service 'EnlightPostProcessor'
- Service 'Hdf5VariableReader'
- Service 'Hdf5VariableWriter'
- Service 'HoneyComb2D'
- Service 'HoneyComb3D'
- Service 'Mets'
- Service 'Mets'
- Service 'PDESRandomNumberGenerator'
- Service 'SimpleCivComparator'
- Service 'SimpleCivOutput'
- Service 'Sed3D'

Interface des services

**Module 'arcane-post-traitement'**

Options concernant les **sorties** pour le dépouillement.  
Les sorties peuvent être effectués en fonction du nombre d'itérations (options `periode-sortie`) ou en fonction du temps physique (option `frequence-sortie`). Par exemple:  
`<periode-sortie>5</periode-sortie>`

**Liste des options**

- **compression-sortie-historique**
- **dépouillement** (2 options)
  - **groupe**
  - **variable**
- **format-service**
- **frequence-sortie**
- **periode-sortie**
- **periode-sortie-historique**
- **sauvegarde-initiale**
- **sauvegarde-temps-final**
- **sortie-fin-execution**

**Documentation des options**

**compression-sortie-historique** (bool=false)

Evite de sauvegarder tous les points d'abscisse entre les périodes de l'option `output-history-period`. Cela permet de faire baisser la consommation mémoire notamment lorsque le code effectue un grand nombre d'itérations.

`<compression-sortie-historique>value</compression-sortie-historique>`

**dépouillement**

Liste des variables et des groupes à dépouiller.

**dépouillement/groupe** [0... undefined] (Arcane::ItemGroup)

Groupe d'entité de maillage sur lequel on souhaite exporter des variables. Il est possible de mettre plusieurs options de ce type pour exporter les variables sur plusieurs groupes. Si cette option n'est pas spécifiée, les variables sont sorties sur l'ensemble du maillage.

`<groupe>value</groupe>`

**dépouillement/variable** [0... undefined] (usttring)

Nom d'une variable à exporter. Il est possible de mettre plusieurs options de ce type pour exporter plusieurs variables. Si aucune variable n'est spécifiée, aucune sortie n'est effectuée.

`<variable>value</variable>`

Arcane Services et modules disponibles Liste des modules et services du jeu de données



# ROADMAP ET POINT SUR L'OPEN SOURCE



**Rencontres Arcane**

17 avril 2023



## ● Travaux futurs

- Deuxième moitié avenant Arcane 2023-2024
- Pistes de réflexion futur avenant
- PEPR NumPEX



### ● Deuxième moitié de l'avenant n°6 (2023-2024)

#### ● Développer une **application modulaire**

- Nouvelle brique de base gérant les variables
- API de création dynamique de service ou de module
- Multi-représentation pour le maillage

#### ● Développer une **application performante**

- Finalisation API GPU
- Profilage, optimisation et non régression en perf (ShArc)
- Déploiement multi-thread et SIMD (ShArc)



- Pistes de réflexion futur avenant
  - La réflexion démarrera à partir de mi-2023
  - Quelques pistes
    - Un volet IA (inférence efficace de réseaux de neurones)
    - Un volet R&D via NumPEX : post-doctorats commun IFPEN/CEA
    - Intégration des travaux NumPEX (ShArc)
    - Toutes vos propositions...!

### ● PEPR NumPEx (Marc Pérache)



*Innovater les énergies*

Retrouvez-nous sur :

 [www.ifpenergiesnouvelles.fr](http://www.ifpenergiesnouvelles.fr)

 @IFPENinnovation

