

# Numerical Simulation Components for Open Python Environments

Marc Poinot (marc.poinot@onera.fr) - Dépt. Simulation Numérique et Aéroacoustique  
 Benoit Rodriguez (benoit.rodriguez@onera.fr) - Dépt. Aérodynamique Appliquée  
 Xavier Juvigny (xavier.juvigny@onera.fr) - Dépt. Techniques de l'Information et Modélisation  
 Jean-Didier Garaud (jean-didier.garaud@onera.fr) - Dépt. Matériaux et Structures Métalliques



## CONTEXTE

De nombreux codes de calculs utilisent désormais des assemblages de composants Python HPC, avec des bibliothèques Fortran, C, C++, du MPI et la plupart du temps une interface Python/Numpy générée par SWIG, F2PY ou réalisée en Cython.

Malgré ces techniques communes, il reste souvent impossible d'assembler les codes en des applications de simulation multi-physique, voire de réutiliser des applications ou des parties d'applications pour en faire de nouveaux composants. L'effort est porté sur la technologie, pas sur le modèle de données, le modèle de contrôle et le cycle de vie.

NSCOPE propose des pratiques communes par des guides, des formations et des canevas de code source. Aucune méthode, application ou bibliothèque propriétaire n'est requise, seules les ressources Open Source très largement répandues sont considérées.

**NSCOPE** est né de l'expérience autour du solver CFD **elsa** de l'ONERA. De nombreux outils, extensions, assemblages applicatifs, couplages de codes sont réalisés autour d'elsa, en milieu industriel ou en milieu recherche.

Lors des intégrations de composants dans **elsa** ou bien lors d'intégration dans les ateliers applicatifs divers (MpCCI, Calcium, Salomé, OpenPalm, FSDM, Cannelle, Ganesh...) l'ONERA propose une interface de type Open System basée sur les standards de facto et les normes.

**NSCOPE** est un Axe de Recherche Fédérateur (ARF), projet interne ONERA regroupant 8 départements (mécanique des fluides, énergétique, électromagnétisme, aérodynamique appliquée, informatique HPC, structures, matériaux, physique).

**Aujourd'hui:** après 18 mois, NSCOPE est à même de prioriser ses thèmes et de définir les livrables (docs, formations, canevas, composants). En un an, NSCOPE a formé 50 ingénieurs/chercheurs de l'ONERA aux techniques de mise en composant avec interface Python/Numpy.

**Demain:** NSCOPE se présente aux avis d'un groupe d'experts reconnus en simulation multi-physique & logiciel, industriels, organismes de recherche et éditeurs de logiciel puis ouvre un site web public.

## METHODOLOGIE

### Les 3 thèmes NSCOPE

#### GL: Génie logiciel

Regroupe des méthodes et des outils relatifs au cycle de vie d'un composant (spécification, conception, réalisation, diffusion, maintenance...). Le thème GL correspond donc à la constitution d'une boîte à outils publics mais comprend aussi des méthodes de construction de logiciel.

#### BC: Bibliothèque de composants

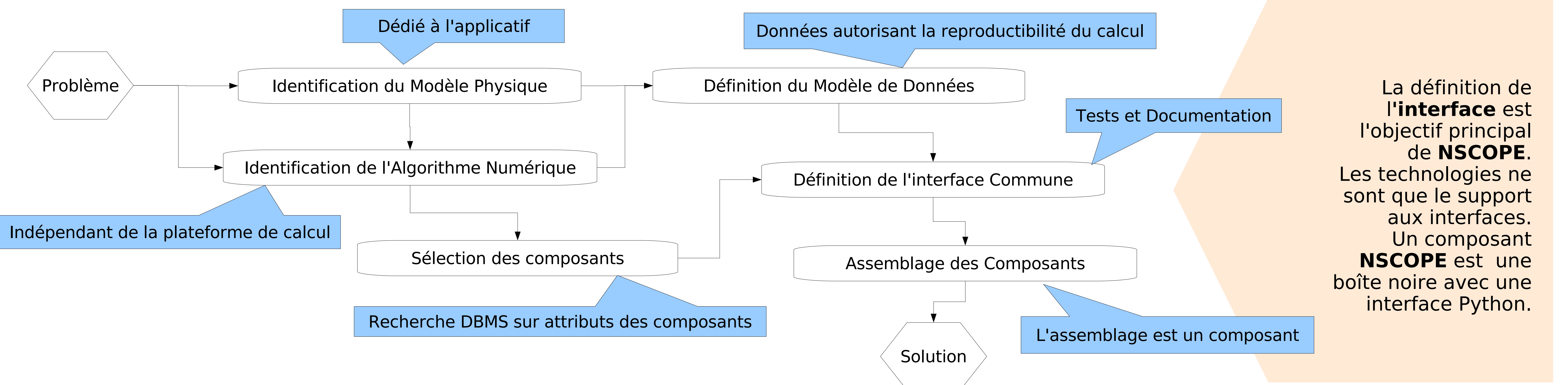
Exemples de composants NSCOPE et des techniques de mise en composants associées. Le thème BC montre des mises en oeuvre des outils et méthodes logiciels identifiés dans GL.

#### AP: Assemblages applicatifs

Illustration des assemblages de composants, identifie les critères de sélection des composants, propose des techniques de communication et de contrôle pour les simulations. Un important volet algorithmique propose diverses techniques synchrones et asynchrones d'échange de données et de pilotage de systèmes répartis.

Le composant **NSCOPE** est décrit par son **interface**:

- Signatures de fonctions, modèles de données, constantes...
- Protocole d'utilisation, erreurs, limites...
- Dépendances, performances, licence, disponibilité...



## TECHNOLOGIES & SERVICES

Formation MPI

Formation Python

Formation Cython

Formation CGNS

Guide basique *inno* setup et livraison Windows

Canevas MPI avec communicateur externe

Formation Mise en composant Python

Canevas Numpy avec partage Fortran en mémoire

Support pour mise en projet Mercurial

Guide basique intégration librairie Fortran dans un module Python livrable

Canevas de modèle de données CGNS pour FSI

Canevas d'algorithmes de couplage FSI

Composant CGNS/HDF5

Guide des contraintes clusters sur les accès réseaux, file system et signaux

Support pour extension du modèle CGNS

Composant Interpolation maillage

Guide sur les stratégies de distribution des fichiers HDF5 sur cluster

Composant sauvegarde disque asynchrone

Guide du code Python performant

Formation Numpy

Canevas de simulation mixte HPC/station hors batch avec serveur asynchrone Python

Guide basique SCM branche de développement/branche de livraison

Canevas d'interface d'un module Python HPC embarquable

Support pour mise en production scons

Composant Interface ré-entrante

Canevas doc Sphinx pour utilisateur et développeur

Guide de définition des interfaces et tests associés

Canevas de distribution Python pour le HPC

Composant Déformation maillage

### Exemple

Un code Fortran de CSM est transformé en composant.

L'utilisation de Cython et le partage par tableaux Numpy en mémoire sont mis en oeuvre en suivant les conseils **NSCOPE**, sur les contraintes d'allocation de la mémoire utilisée pour partager les tableaux Fortran/Python, ou encore sur le contenu du *main* et l'initialisation de MPI, les accès au *file system* ou la gestion de signaux. Une interface est définie, le nouveau code est capable d'introspection et répond aux demandes des applications intégratrices sur son statut MPI, sur la disposition mémoire de ses tableaux (entrelacés, Fortran/C, contigus...).

Pour le test et la documentation l'approche 'le test avant le code' est appliquée, la doc et les tests décrivent l'interface avant le codage. Les tests sont aussi utilisés pour vérifier le comportement du module après intégration dans une application, ils sont livrés et auto-suffisants.

L'application utilisatrice est un couplage CFD/CSM avec fluide sur cluster HPC avec MPI et structure sur une machine déportée sans file d'attente. Le canevas du système est pris dans **NSCOPE** mais l'utilisateur n'est pas contraint par un atelier propriétaire.

Toutes les technologies sont disponibles hors **NSCOPE** qui accompagne le module de la spécification à la livraison.