



HAL
open science

Utilisation de la technologie des composants pour la résolution de problèmes d'interaction fluide-structure

Christophe Kassiotis, Adnan Ibrahimbegovic, Hermann Matthies

► To cite this version:

Christophe Kassiotis, Adnan Ibrahimbegovic, Hermann Matthies. Utilisation de la technologie des composants pour la résolution de problèmes d'interaction fluide-structure. 9e colloque national en calcul des structures, CSMA, May 2009, Giens, France. hal-01499927

HAL Id: hal-01499927

<https://hal.science/hal-01499927>

Submitted on 1 Apr 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Public Domain

Utilisation de la technologie des composants pour la résolution de problèmes d'interaction fluide-structure

C. Kassiotis^{1,2,3}, A. Ibrahimbegović¹ et H. G. Matthies²

¹ LMT-Cachan (ENS Cachan/CNRS/UPMC/PRES UniverSud Paris)
61 av. du Président Wilson
F-94230 Cachan, France
{kassiotis,ai}@lmt.ens-cachan.fr

² Institut für Wissenschaftliches Rechnen (TU-Braunschweig)
D-38092 Braunschweig
wire@tu-bs.de

³ UR Navier (École des ponts/Université Paris-Est/LCPC/CNRS)
6/8 av Blaise Pascal – Cité Descartes – Champs-sur-Marne
77455 Marne-la-Vallée cedex 2

Résumé — Dans notre exposé, nous montrerons le potentiel de l'utilisation de la technologie des composants pour la modélisation et la simulation de problème multi-physique. Cette approche permet de coupler les codes existants par tout, diminue le coût de développement tout en étant efficace d'un point de vue temps de calcul.

Mots clés — Interaction fluide-structure, approche partitionnée, couplage de codes.

1 Introduction

Si dans un premier temps, la résolution de problèmes multi-modèles a pu passer par le développement de nouveaux codes, il apparaît que le coût de implémentation de tels outils numériques pour prendre en compte des phénomènes fins à l'échelle d'une des physiques couplées rend peu viable ce type solution.

D'un autre côté, l'approche partitionnée permettant de conserver à la fois les outils et stratégies développés pour chacune des physiques couplées pour résoudre ces problèmes est longtemps restée limitée par la lenteur des algorithmes de couplages, lenteur accentuée par de dispendieux échanges de fichiers utilisés pour le couplage des codes.

Nous proposons dans nos travaux une approche basée sur la technologie des composants permettant une communication directe entre les codes, via `thread` ou protocoles réseaux comme le `tcp`, autorisant ainsi la résolution de problèmes multi-modèles en couplant des codes de manière performante. Nous illustrons nos propos par l'exemple de structures côtières impactées par des vagues. Pour envisager ce type de phénomènes nous considérons trois types de composants :

- Propagation de la vague. A une échelle suffisamment grande, une modélisation utilisant un fluide potentiel et des équations non-linéaires de surface libre donne de bon résultat. La discrétisation du problème est assurée par une méthode aux Éléments de Frontières [1].

- Écoulement du fluide à proximité de la structure. Nous utilisons ici la méthode des Volumes Finis [2] basé sur une formulation ALE, souvent utilisé dans les codes industriels de mécanique des fluides.
- Structure (de Génie Civil). Nous avons développé le composant `coFeap` [3] permettant de réutiliser nos outils numériques développés pour le code Éléments Finis `Feap` dans un contexte plus général.

Toutefois dans cet article nous nous intéresserons exclusivement au couplage des deux derniers composants pour la résolution de problème d'interaction fluide-structure académique.

2 La stratégie de couplage

L'idée directrice de nos travaux est de coupler des codes existants dans un contexte plus général en utilisant le *middleware* CTL [4]. Cet outil donnée, reste la question des algorithmes de couplages.

Nous considérons ici des algorithmes de couplage forts de type point fixe dont la stabilité n'est assuré que sous certaines conditions – même si chacun des sous-problèmes est assuré de converger [5]. De plus, nous cherchons à conserver une indépendance maximum des schémas d'intégration temporelle, permettant à la fois de conserver les schémas d'intégration et la taille des pas de temps naturellement utilisés par chacun de nos modèles. Notons que, dans ce contexte, ce ne sont pas les variables à certains instants qui sont échangées, mais leurs évolutions sur un interval de temps $[T_N, T_{N+1}]$. Les discrétisations des sous-problèmes sont indépendantes, tant du point de vue de l'algorithme que du pas de temps choisis (voir Fig. 1).

Pour être plus précis, considérons deux sous-problèmes (a priori non-linéaires et évoluant en temps) fluide et structure (respectivement d'indices f et s). Pour chacun des sous-problèmes on peut définir un opérateur de Steklov-Poincaré. Par exemple pour le problème de structure :

$$\begin{aligned} \mathcal{S}_s : \mathcal{H}^{\frac{1}{2}}(\Gamma_{fs}) &\rightarrow \mathcal{H}^{-\frac{1}{2}}(\Gamma_{fs}) \\ \mathbf{u} &\rightarrow \boldsymbol{\sigma}_s \mathbf{n} \end{aligned} \quad (1)$$

est un opérateur qui associe aux évolutions de déplacements à l'interface les forces généralisés à cette même interface. Le même opérateur peut bien évidemment être définie pour le problème fluide. Ces opérateurs se décompose en trois phases :

- chargement (imposer la condition au limite) – peu coûteux en temps de calcul.
- résolution (calculer la solution du problème avec les conditions aux limites données) – coûteux en temps de calcul.
- restriction (restreindre la solution sur l'ensemble du domaine à l'interface)– peu coûteux en temps de calcul.

La résolution du problème d'interaction fluide-structure impose, outre la résolution des sous-problèmes que nous ne détaillerons pas ici, l'équilibre des forces et l'égalité des vitesses ou déplacements à l'interface. En utilisant les opérateurs de Steklov-Poincaré définis ci-dessus, on montre que :

$$\begin{cases} \mathbf{u}_s - \mathbf{u}_f = 0 \\ \boldsymbol{\sigma}_s \mathbf{n} - \boldsymbol{\sigma}_f \mathbf{n} = 0 \end{cases} \Leftrightarrow \begin{cases} \mathbf{u}_s = \mathbf{u}_f = \mathbf{u} \\ \mathcal{S}_s(\mathbf{u}) - \mathcal{S}_f(\mathbf{u}) = 0 \end{cases} \Leftrightarrow \mathbf{u} - \mathcal{S}_s^{-1}(\mathcal{S}_f(\mathbf{u})) = 0 \quad (2)$$

Le problème couplé peut donc se ramener à la résolution du problème de type point-fixe suivant :

$$\mathcal{R}(\mathbf{u}) = \mathcal{S}_s^{-1}(\mathcal{S}_f(\mathbf{u})) - \mathbf{u} = 0 \quad (3)$$

La résolution de ce problème non-linéaire peut-être abordée par des algorithmes du type :

1. Prédiction initiale $\mathbf{u}_{N+1}^{(0)}$ (en fonction des valeurs \mathbf{u}_N convergées) et calcul du résidu initial $\mathcal{R}(\mathbf{u}_{N+1}^{(0)})$

2. Si $\|\mathcal{R}(\mathbf{u}_{N+1}^{(k)})\| < \text{TOL}$ fin.
3. Résoudre $\mathcal{J}_{N+1}^{(k)} \cdot \delta \mathbf{u}_{N+1}^{(k)} = -\mathcal{R}(\mathbf{u}_{N+1}^{(k)})$
4. Actualiser $\mathbf{u}_{N+1}^{(k+1)} = \mathbf{u}_{N+1}^{(k)} + \delta \mathbf{u}_{N+1}^{(k)}$

Où $\mathcal{J}_{N+1}^{(k)}$ donne une “direction” de recherche calculé de manière plus ou moins précise au prix d’efforts de calculs et de développements informatiques souvent importants. Nous utilisons ici une approche de type Block-Gauss Seidel (voir Fig. 1) avec relaxation de type Aitken (équivalente à considérer $\mathcal{J}_{N+1}^{(k)}$ scalaire) qui a montrée son efficacité [6] pour un coût d’implantation faible.

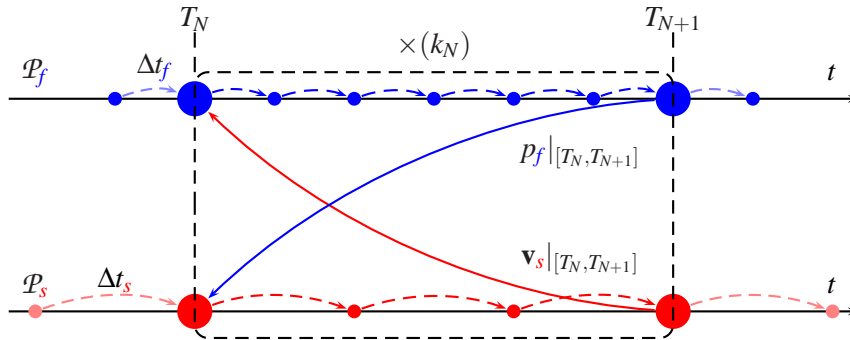


Figure 1 – Block Gauß-Seidel coupling algorithm for fluid (\mathcal{P}_f) structure (\mathcal{P}_s) interaction problem ; this iterative scheme is applied (k_N) times until convergence on a window $[T_N, T_{N+1}]$.

Remarque : l’opérateur de Steklov-Poincaré est *a priori* aussi couteux que l’opérateur de Poincaré-Steklov.

3 Technologie des composants

L’idée de la technologie des composants est de généraliser la programmation orientée objet. Un composant est un logiciel (ou une partie de logiciel) qui réalise certaines tâches (services) définie par une interface pour un client. Il nous paraît important de souligner les points suivants :

- L’interface peut-être vue comme un contrat liant les deux parties clients/services.
- La communication entre composants/clients est assurées par un *middleware* (intergiciel). Nous utilisons CTL.
- L’implantation d’un composant est libre du point de vue du langage de programmation ainsi que des paradigmes utilisés pour leur développement. Ainsi, dans nos travaux, nous couplons des codes C++, Fortran et C orienté objets ou non, et développés par différentes équipes.

L’architecture d’un composant résolvant des problèmes mécaniques (coFeap) est détaillé en Fig. 2. Ce composant est suffisamment général pour pouvoir être utilisé pour la résolution de différents problèmes mécaniques évolués (multi-physiques, décomposition de domaine, multi-échelles...).

Dans la Partie 2 nous avons montrés qu’un opérateur de type Steklov-Poincaré pour la partie solide nécessitait les services suivant (appel en C++) :

```
Solid.set_load(f[solid]);
Solid.solve();
Solid.get_state(u[solid]);
```

CTL nécessite la définition des interfaces en C-preprocessor est défini comme suit pour le service set_load :

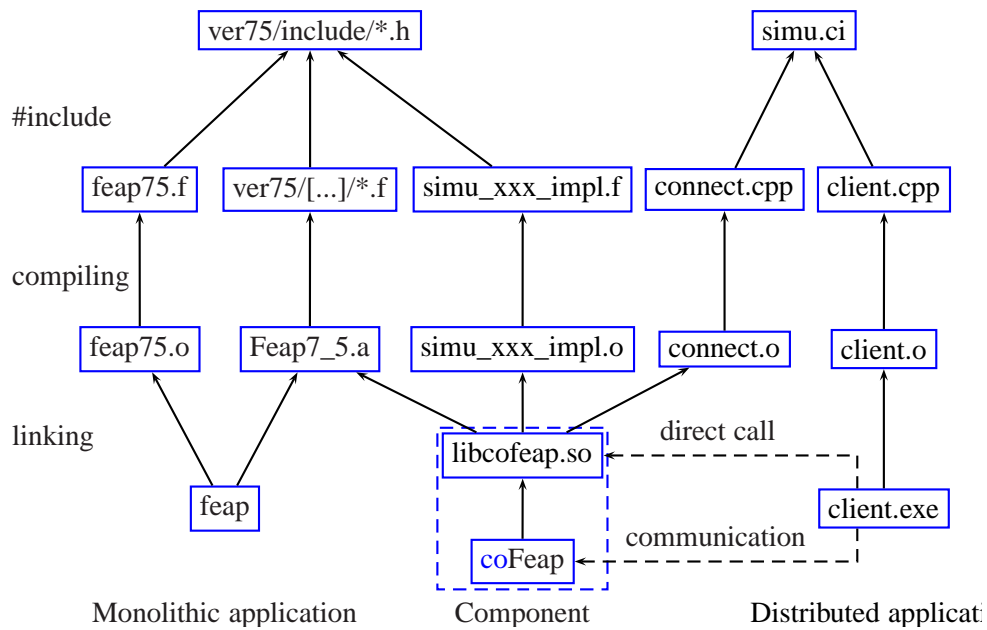


Figure 2 – Graphique de dépendance et de *linkage* pour *coFeap*

```
# define CTL_Method6 void, set_load, (const array<scalar1> /*load*/), 1
```

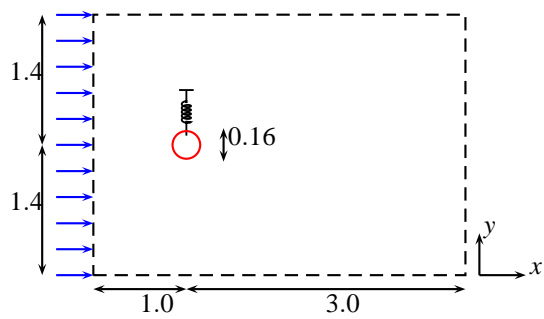
Le service en lui-même est implémenté en Fortran, puisque notre composant mécanique (*coFeap*) est basé sur le code élément finis *Feap* programmé en Fortran 75 :

```
subroutine simu_set_load_impl(s0,x0)
  implicit none
[...]
```

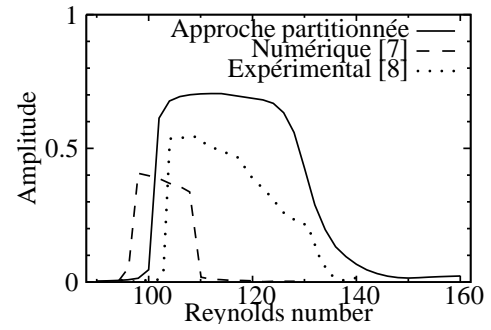
```
include 'control.h'
integer s0, i, j, k
real*8 x0(s0)
[...]
```

```
  k = 0
  do i = 0,neq-1
    j = mr(np(31)+i+k)
    do while(j.lt.0)
      k = k + 1
      j = mr(np(31)+i+k)
    enddo
    hr(np(27)+i+k) = x0(i+1)
  enddo
end
```

Il n'y a pas d'échange de fichiers pour transmettre les données entre composants : les données *u[solid]* sont directement envoyées dans la mémoire du composant mécanique (utilisation du pointeur *hr(np(27))*). De plus, les composants peuvent être appelés grâce à CTL en utilisant différents protocoles, sur une même machine ou à travers des réseaux, ainsi qu'en parallèle.

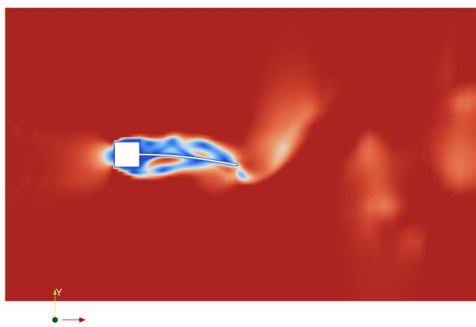


(a) Benchmark unidimensionnel

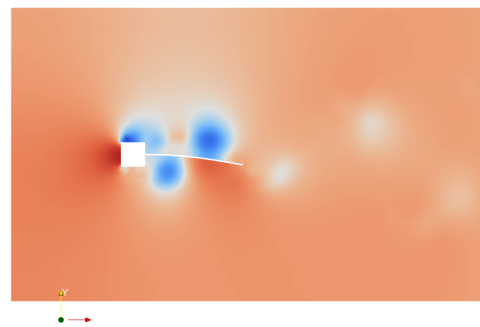


(b) Comparaison avec d'autres résultats

Figure 3 – Test 1-D : disque oscillant

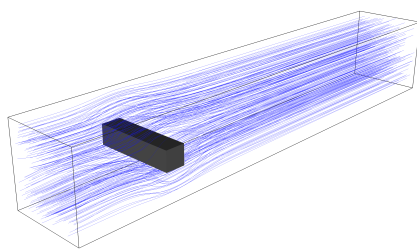


(a) Champ de vitesse à $t = 7,8s$

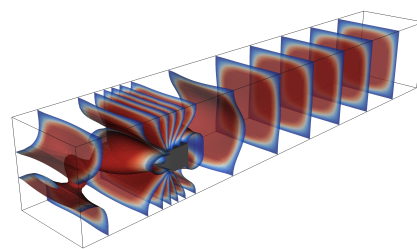


(b) Champ de pression à $t = 7,8s$

Figure 4 – Test 2-D : poutre excitée sur un de ses modes



(a) Ligne de courants



(b) Isosurface de pression

Figure 5 – Test 3-D : cylindre oscillant

4 Résultats et validation

Les résultats présentés au cours de notre exposé concerneront la validation de notre approche pour des problèmes d'interaction fluide-structure. Les benchmarks de la littérature [7] (Fig. 3 et 4) et la proposition d'un benchmark 3D (Fig. 5) sont basés sur l'idée suivante : construire un problème d'écoulement instationnaire autour d'un solide pour lequel la fréquence propre est celle de lâché des tourbillons qu'il génère.

La comparaison des résultats, obtenus par notre code basé sur une approche partitionnée (Fig. 3, 4 et 5), avec les observations tant numériques [6, 7] qu'expérimentales [8] donne des résultats encourageants.

5 Conclusion

La résolution de problème multi-modèles fortement couplés par une approche partitionnée est ici implémentée dans le contexte de la technologie des composants permettant notamment la réutilisation de codes dans un contexte plus général, ainsi qu'un couplage performant (car fait sans échange de fichiers).

Cette approche est validée par la résolution de Benchmark, notamment tri-dimensionnels. La prise en compte des écoulements surfaces libres fait partie des extensions possibles de ce travail.

Remerciements

Le financement de ces travaux est en partie assuré par l'Université Franco-Allemande (UFA-DFH).

Références

- [1] C. Fochesato, S. Grilli, and F. Dias. Numerical modeling of extreme rogue waves generated by directional energy focusing. *Wave Motion*, 44 :395–416, 2007.
- [2] H. Jasak. *Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows*. Phd Thesis, Department of Mechanical Engineering, Imperial College of Science, Technology and Medicine, 1996.
- [3] C. Kassiotis and M. Hautefeuille. *coFeap's Manual*. École Normale Supérieure de Cachan, Laboratoire de Mécanique et Technologies, Secteur Génie Civil, 2008. <http://www.lmt.ens-cachan.fr/cofeap>.
- [4] H. G. Matthies, R. Niekamp, and J. Steindorf. Algorithms for strong coupling procedures. *Computer Methods in Applied Mechanics and Engineering*, 195 :2028–2049, 2006.
- [5] M. Arnold and M. Gunther. Preconditioned dynamic iteration for coupled differential-algebraic systems. *BIT Numer. Math.*, 41 :1–25, 2001.
- [6] U. Küttler and W. A. Wall. Fixed-point fluid-structure interaction solvers with dynamic relaxation. *Computational Mechanics*, 43(1) :61–72, 2008.
- [7] W. G. Dettmer and D. Perić. A fully implicit computational strategy for strongly coupled fluid-solid interaction. *Archives of Computational Methods in Engineering*, 14 :205–247, 2007.
- [8] P. Anagnostopoulos and P.W. Bearman. Response characteristics of a vortex-excited cylinder at low Reynolds numbers. *Journal of Fluids and Structures*, 6(1) :39 – 50, 1992.