

# Cyrille Piacibello

Développeur

234 rue de Suzon

33400 Talence

☎ 06 82 21 11 76

✉ cyrille.piacibello@gmail.com

## Expériences

- Avril 2017 **Ingénieur d'étude**, *IAP-CNRS*, Paris, Participation au développement du simulateur de l'instrument VIS de la mission spatial EUCLID.  
Aujourd'hui Environnement technique : Python, Linux, git
- Octobre 2015 **Ingénieur d'étude**, *INRIA*, Bordeaux, Implémentation d'un solveur itératif.  
Mars 2017 Environnement technique : C, C++, OpenMP, Linux, git
- Octobre 2013 **Ingénieur d'étude**, *INRIA*, Bordeaux, Participation au projet ScalFMM.
- Octobre 2015 Environnement technique : C, C++, MPI, OpenMP, Linux, git

## Compétences

### Informatique

Langages : C, C++ (norme 2011)

Standards : BLAS, LAPACK, MPI, OpenMP

Outils : Shell,  $\LaTeX$ , Git, Svn

### Langues

Anglais lu, écrit, parlé

Rédaction documentaire. TOEIC 800 en 2012

## Diplômes et Études

- 2013 **Diplôme Ingénieur**, *ENSEIRB-MATMECA*, Bordeaux.  
Spécialisation P.R.C.D : Parallélisme, Régulation et Calcul Distribué
- 2010 **Classes Préparatoires**, *Lycée Massena*, Nice.

## Détails des expériences

IAP-CNRS **Ingénieur d'étude**, *Participation au développement du simulateur de l'instrument VIS de la mission spatiale EUCLID.*

- Étude et développement de nouveaux effets instrumentaux (réponse impulsionnelle, lumière parasite)
- Validation d'effets instrumentaux.
- Contribution au Challenge Scientifique.

Environnement technique: Python, Git, Jenkins

INRIA **Ingénieur d'étude**, *IB-BGMRes-DR : Implémentation d'un solveur itératif.*

- Implémentation d'un GMRes (multi second membre) par bloc avec produit matrice vecteur externe.
- Détection et prise en compte d'Inexact Breakdown (convergence partielle parmi les seconds membres).
- Deflation au restart (recyclage d'information au restart).

Environnement technique: C++, C, Git

INRIA **Ingénieur d'étude**, *Participation au projet ScalFMM, bibliothèque générique implémentant l'algorithme de Fast Multipole Method.*

- Mise en place d'un ordonnancement par tâche (Standard OpenMP 4.0)

Environnement technique: C++, C, MPI, OpenMP, Git