# Sujet de stage de M2 Informatique :
# Implantation fonctionnelle d'algorithmes géométriques en 3D

*Please check out page 2 for an english version of this document.*

Nicolas Magaud (magaud@unistra.fr)
https://dpt-info.u-strasbg.fr/~magaud/
Lab. ICube UMR 7357 CNRS Université de Strasbourg, France

Décrire formellement des algorithmes géométriques, puis en démontrer la correction dans un outil d'aide à la preuve tel que Coq [5, 1] nécessite en premier lieu d'en produire une implantation fonctionnelle. Il y a quelques années, nous avons décrit et prouvé formellement des algorithmes, implantés en programmation fonctionnelle, pour calculer l'enveloppe convexe d'un ensemble de points en 2D [4, 3].

L'objectif de ce travail est d'aborder le domaine de la preuve formelle en 3D. Nous proposons d'implanter, en programmation fonctionnelle, dans un premier temps en OCaml, puis ensuite en Coq, des algorithmes géométriques en 3D [2, 6]. Un premier exemple à considérer sera le calcul de de l'enveloppe convexe d'un ensemble de points en 3D. Il s'agira de choisir des structures de données bien adaptées au problème, en utilisant par exemple des structures combinatoires comme les cartes. On cherchera ensuite à implanter différents algorithmes de calcul de l'enveloppe convexe en 3D, notamment les versions *incrémental* et *quickhull*. Pour cela, on pourra s'appuyer sur les implantations disponibles dans la bibliothèque C++ de modélisation géométrique CGoGN [1] [7], développée à Strasbourg et fournissant une implantation efficace des cartes combinatoires. Afin de rendre ce travail plus concret, on étudiera comment relier le code fonctionnel produit avec un outil de visualisation des objets en 3D. Cela permettra, d'une part, de rendre le résultat présentable et, d'autre part, d'identifier d'éventuels bugs du programme. Si le temps le permet, ce travail de programmation pourra se prolonger en un travail de preuve formelle où la question de la correction de ces algorithmes sera étudiée.

## Références

[1] Yves Bertot and Pierre Castéran. *Interactive Theorem Proving and Program Development, Coq'Art : The Calculus of Inductive Constructions.* Springer, 2004.

[2] Jean-Daniel Boissonnat and Mariette Yvinec. *Algorithmic Geometry.* Cambridge University Press, 1998. 544 pages. Translated by Hervé Brönnimann.

[3] Christophe Brun, Jean-François Dufourd, and Nicolas Magaud. Designing and Proving Correct a Convex Hull Algorithm with Hypermaps in Coq. *Computational Geometry, Theory and Applications*, 45(8) :436–457, 2012.

[4] Christophe Brun, Jean-François Dufourd, and Nicolas Magaud. Formal Proof in Coq and Derivation of a Program in C++ to Compute Convex Hulls. In Tetsuo Ida and Jacques D. Fleuriot, editors, *Automated Deduction in Geometry (ADG'2012) - Revised Selected Papers*, volume 7993 of *LNCS*, pages 71–88. Springer, 2012. ISBN 978-3-642-40671-3.

[5] Coq development team. *The Coq Proof Assistant Reference Manual, Version 8.14.0*, 2021.

[6] Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. *Computational Geometry, Algorithms and Applications (Third Edition).* Springer-Verlag, Berlin/Heidelberg, March 2008. 386 pages.

[7] Pierre Kraemer, Lionel Untereiner, Thomas Jund, Sylvain Thery, and David Cazier. Cgogn : *n*-dimensional meshes with combinatorial maps. In Josep Sarrate and Matthew L. Staten, editors, *Proceedings of the 22nd International Meshing Roundtable, IMR 2013, October 13-16, 2013, Orlando, FL, USA*, pages 485–503. Springer, 2013.

1. https://cgogn.github.io/

# Master Internship in Computer Science
# Functional Implementation of Geometric Algorithms in 3D

Nicolas Magaud (magaud@unistra.fr)
https://dpt-info.u-strasbg.fr/~magaud/
Lab. ICube UMR 7357 CNRS Université de Strasbourg, France

Describing geometric algorithms in a formal setting and then proving their correctness in a proof assistant such as Coq [5, 1] first requires to implement them using the functional programming paradigm. A few years ago, we formally described and proved correct the implementation of geometric algorithms to compute the convex hull of a set of points in the plane [4, 3].

In this internship, we shall investigate how similar proofs can be carried out in a 3D setting. We aim at implementing, first using a mainstream functional programming language such as Ocaml, and then using the specification language Gallina of the Coq proof assistant [2, 6], some geometric algorithms in 3D. A first example could be the computation of the convex hull of a set of points in 3D. This would require choosing appropriate data structures, e.g. combinatorial structures such as maps. We shall then implement the *incremental* and *quickhull* variants of the algorithm. As a starting point, we could consider the implementations of such algorithms as they are provided in the C++ geometric modeling library CGoGN [1] [7], designed and implemented in our team in Strasbourg. Moreover, we shall also study how to link our development to some simple visualization tools. This would allow to check whether the algorithm seems to work correctly and this would also make the result easier to understand. If we have enough time, the programming task of implementing the algorithms in a functional manner could be extended by studying how to carry out a formal proof of correctness of such algorithms.

## Références

[1] Yves Bertot and Pierre Castéran. *Interactive Theorem Proving and Program Development, Coq'Art : The Calculus of Inductive Constructions.* Springer, 2004.

[2] Jean-Daniel Boissonnat and Mariette Yvinec. *Algorithmic Geometry.* Cambridge University Press, 1998. 544 pages. Translated by Hervé Brönnimann.

[3] Christophe Brun, Jean-François Dufourd, and Nicolas Magaud. Designing and Proving Correct a Convex Hull Algorithm with Hypermaps in Coq. *Computational Geometry, Theory and Applications*, 45(8) :436–457, 2012.

[4] Christophe Brun, Jean-François Dufourd, and Nicolas Magaud. Formal Proof in Coq and Derivation of a Program in C++ to Compute Convex Hulls. In Tetsuo Ida and Jacques D. Fleuriot, editors, *Automated Deduction in Geometry (ADG'2012) - Revised Selected Papers*, volume 7993 of *LNCS*, pages 71–88. Springer, 2012. ISBN 978-3-642-40671-3.

[5] Coq development team. *The Coq Proof Assistant Reference Manual, Version 8.14.0*, 2021.

[6] Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. *Computational Geometry, Algorithms and Applications (Third Edition).* Springer-Verlag, Berlin/Heidelberg, March 2008. 386 pages.

[7] Pierre Kraemer, Lionel Untereiner, Thomas Jund, Sylvain Thery, and David Cazier. Cgogn : *n*-dimensional meshes with combinatorial maps. In Josep Sarrate and Matthew L. Staten, editors, *Proceedings of the 22nd International Meshing Roundtable, IMR 2013, October 13-16, 2013, Orlando, FL, USA*, pages 485–503. Springer, 2013.

1. https://cgogn.github.io/