

Année 2009–2010
Proposition de stage de Licence 3

TITRE : PREUVE FORMELLE SUR L'ARITHMÉTIQUE FLOTTANTE EN DOUBLE
ARRONDI

LIEU : INRIA Saclay – Île-de-France
Équipe-Projet Proval
Parc Orsay Université, Bâtiment N
4 rue Jacques Monod
91893 Orsay Cedex

PERSONNES ENCADRANT LE STAGE LOCALEMENT :

Sylvie Boldo
Tél: 01 74 85 42 26
Email : sylvie.boldo@inria.fr

Guillaume Melquiond
Tél: 01 74 85 42 86
Email : guillaume.melquiond@inria.fr

CONTEXTE :

L'équipe-projet ProVal propose des méthodes et outils de développement logiciel faisant une large place à la preuve de programmes assistée par ordinateurs. L'un des axes de recherche concerne la vérification de programmes utilisant l'arithmétique à virgule flottante.

Cette arithmétique est une approximation de l'arithmétique usuelle sur les nombres réels. Les calculs y sont effectués en précision finie ; les résultats intermédiaires ne sont pas représentés exactement. Cela cause des erreurs dites d'arrondi qui peuvent se propager, s'amplifier et potentiellement faire diverger les résultats.

L'une des approches pour certifier un programme est la vérification déductive, c'est-à-dire la preuve de propriétés logiques extraites du programme. Cela s'appuie sur une base de théorèmes formellement prouvés avec le système Coq.

OBJET DU STAGE :

Lors de la conception d'un algorithme numérique, le choix de la précision de calcul peut jouer un rôle déterminant. Mais cette précision n'est pas forcément disponible dans tous les environnements où l'algorithme sera implanté. Par exemple, un algorithme en simple précision (`float`) peut se retrouver exécuté sur une architecture qui ne calcule qu'avec des nombres double précision (certains PowerPC) ou de précision plus grande encore (certains x86). Contrairement à l'intuition, l'utilisation d'une précision trop élevée peut produire des résultats d'une moins bonne qualité [1]. Il existe cependant des mé-

thodes, en particulier celles à base de double arrondi, qui garantissent que le résultat sera celui attendu même en cas de précision trop élevée.

L'objet du stage sera d'étudier la littérature sur ce sujet, notamment [2], et d'en tirer des théorèmes prouvés formellement en Coq. Le but est de trouver des conditions dans lesquelles les résultats sont inchangés et ainsi de justifier certaines transformations de code dans des compilateurs [3].

Les développements de ce stage seront intégrés dans la bibliothèque Pff¹ et ainsi accessibles aux personnes prouvant formellement des programmes numériques.

Références

- [1] Sylvie Boldo and Guillaume Melquiond. Emulation of a FMA and correctly-rounded sums: Proved algorithms using rounding to odd. *Transactions on Computers*, 57(4):462–471, 2008.
- [2] Samuel A. Figueroa. When is double rounding innocuous? *SIGNUM Newsletter*, 30(3):21–26, 1995.
- [3] Xavier Leroy. Formal verification of a realistic compiler. *Communications of the ACM*, 52(7):107–115, 2009.

¹<http://lipforge.ens-lyon.fr/www/pff/>