

## ED STIC - Proposition de Sujets de Thèse pour la campagne d'Allocation de thèses 2015

**Axe Sophi@Stic :**

**Titre du sujet :**

**Mention de thèse :**

**HDR Directeur de thèse inscrit à l'ED STIC :**

---

### Co-encadrant de thèse éventuel :

**Nom :**

**Prénom :**

**Email :**

**Téléphone :**

---

**Email de contact pour ce sujet :**

**Laboratoire d'accueil :**

---

### Description du sujet :

Dans une société où la part des applications embarquées ne cesse de grandir et leur caractère critique d'augmenter, la capacité à en vérifier la correction et la robustesse devient un enjeu majeur. C'est un enjeu particulièrement crucial pour le calcul basé sur l'arithmétique des nombres à virgule flottante. En effet, cette arithmétique qui a des comportements assez inhabituels [1, 2] est de plus en plus utilisée dans les logiciels embarqués. Par exemple, pour certaines valeurs d'entrée, le flot de contrôle lié aux réels peut passer dans une branche de la conditionnelle tandis qu'il passe par l'autre pour les flottants [3].

Le projet CeV a développé un ensemble d'outils et de techniques dédiées à la résolution de ces problèmes grâce à des approches basées sur la programmation par contraintes

[2,5,6,7]. Ces contributions ont montré l'intérêt des approches par contraintes pour vérifier la conformité d'un programme vis à vis de sa spécification, mais aussi les limites de ces techniques. L'objectif de cette thèse est d'améliorer la compréhension de ces techniques pour améliorer leur efficacité et repousser certaines de ces limites de pouvoir traiter des problèmes de taille réaliste.

Concrètement, le travail du thésard pourra porter sur des thèmes suivants:

- □ Stratégies de recherche de solutions: Les stratégies de recherche de solutions sont un thème récurrent de la programmation par contrainte où elles jouent un rôle clef dans la résolution de ces problèmes. Le domaine des nombres flottants qui tient une place particulière située à la croisée des domaines finis (entiers) et continus (réels) n'a pas encore fait l'objet de recherches rigoureuses sur ce thème. Ce domaine se caractérise aussi par une répartition des solutions liée à la représentation même des flottants qui, selon la dérivée, passe d'un espace où les solutions sont singulières à un espace où les solutions sont « continues ». Il s'agirait donc de sélectionner les stratégies de recherches les plus appropriées pour les flottants, qu'elles s'inspirent des contraintes sur le fini ou sur le continu, et de les adapter aux particularités de ces mêmes flottants. Le développement de stratégies spécifiques, déduites de propriétés de l'arithmétique des flottants, ou même, capables de prendre en compte la structure du problème, pourraient compléter l'approche proposée.
- □ Traitement des boucles: l'un des freins à l'utilisation des contraintes en vérification du logiciel est le traitement des boucles. Pour l'instant, l'approche proposée repose sur un dépliage borné des boucles avec comme conséquence, l'impossibilité de déduire la conformité du programme dans certains cas. Or, si des limites bien connues comme l'indécidabilité de la terminaison restreignent les déductions qu'il est possible d'effectuer sur les boucles, des techniques comme l'interprétation abstraite ont montré leurs capacités à déterminer l'ensemble des états atteignables au sein d'une boucle. Des travaux préliminaires proposent [8] une première approche sur les entiers et orientée vers la génération de cas de tests. Il s'agirait ici d'étendre ces résultats aux flottants et de les consolider pour améliorer les méthodes de vérifications.
- □ Accélération du processus de résolution par des tests d'exclusion: lors de la résolution d'un système de contraintes, le solveur s'appuie généralement sur la combinaison d'un algorithme de recherche avec des procédures de filtrage. Alors que le filtrage tente de réduire la taille des domaines, il est parfois possible d'effectuer un simple test sur un domaine afin de prouver qu'elle ne contient aucune solution. Lorsque ce test, dit « test d'exclusion » est possible et peu coûteux, il permet une accélération notable du processus de résolution. Ces approches ont plutôt vu le jour dans les domaines continus. Il s'agirait donc de définir des procédures propres aux contraintes sur les flottants en cherchant à exploiter les propriétés spécifiques de l'arithmétique des flottants.

Références:

[1] D. Goldberg. What every computer scientist should know about floating-point arithmetic. ACM

Comput. Surv., 23(1) :5-48, 1991.

- [2] D. Monniaux. The pitfalls of verifying floating-point computations. ACM TOPLAS Trans. Program. Lang. Syst., 30(3), 2008.
- [3] E. Goubault and S. Putot. Robustness analysis of finite precision implementations. In Proc. of the 11th Asian Symposium on Programming Languages and Systems (APLAS'13), volume 8301 of LNCS, pages 50-57. Springer, 2013.
- [4] H. Collavizza and M. Rueher. Exploring different constraint-based modelings for program verification. In Proc. of the 13th Int. Conf. on Principles and Practice of Constraint Programming (CP'07), pages 49-63, 2007.
- [5] C. Michel. Exact projection functions for floating point number constraints. In Proc. Int.l Symp. on Artificial Intelligence and Mathematics (AI&M 2002), 2002.
- [6] B. Marre and C. Michel. Improving the floating point addition and subtraction constraints. In Proc. of the 16th Int. Conf. on Principles and Practice of Constraint Programming (CP'10), pages 260-267, 2010.
- [7] B. Botella, A. Gotlieb, and C. Michel. Symbolic execution of floating-point computations. Software Testing, Verification and Reliability, 16(2) :97-121, June 2006.
- [8] T. Denmat, A. Gotlieb, and M. Ducasse. An Abstract Interpretation Based Combinator for Modeling While Loops in Constraint Programming. In Principles and Practices of Constraint Programming (CP'07), pages 241-255, Sep. 2007.

**URL :** [/http://users.polytech.unice.fr/~rueher/sujet.pdf](http://users.polytech.unice.fr/~rueher/sujet.pdf)

**English version:**