2026-2027
Sujet de stage M1/M2
Titre: Compilation et Transformation Automatique de Programmes via des Bibliothèques Numériques Spécialisées

Contexte

Les programmes scientifiques et les applications critiques reposent largement sur l'arithmétique à virgule flottante [1] (Floating-Point – FP), notamment les types `float` et `double`. Cependant, l'utilisation de ces formats ne garantit pas toujours la précision des résultats numériques. Selon les propriétés du programme et les données manipulées, les erreurs d'arrondi peuvent s'accumuler et conduire à des résultats inexacts, voire à des instabilités numériques [2]. Plusieurs bibliothèques proposent des mécanismes alternatifs pour améliorer la précision, analyser les erreurs numériques [3] ou détecter les instabilités. Parmi elles, on peut citer CADNA [4], qui repose sur l'arithmétique stochastique, ainsi que PROMISE [6], MPFR [5] et des implémentations d'arithmétique à précision étendue, telles que l'arithmétique double-simple (double-single), etc.
L'utilisation de ces bibliothèques nécessite généralement de modifier le code source pour remplacer les opérations arithmétiques classiques par des appels aux fonctions correspondantes. Cette adaptation manuelle peut être fastidieuse et difficile à maintenir, particulièrement pour des applications de grande taille. Les compilateurs modernes, notamment LLVM [7] et son infrastructure intermédiaire MLIR [8] (Multi-Level Intermediate Representation), offrent des possibilités intéressantes pour automatiser ces transformations. Il devient ainsi envisageable de détecter les opérations effectuées sur certains types numériques et de les remplacer automatiquement par des appels aux fonctions d'une bibliothèque spécialisée.
Une question importante concerne également les performances : faut-il conserver les appels de fonctions ou appliquer des transformations d'inlining afin d'intégrer directement leur implémentation dans le programme compilé ?

Objectifs

L'objectif principal du stage est de développer une chaîne de compilation permettant de transformer automatiquement des opérations en virgule flottante en appels à des fonctions de bibliothèques numériques spécialisées, puis d'étudier différentes stratégies d'optimisation de ces transformations. Les travaux du/de la stagiaire pourront s'articuler autour des objectifs suivants :

- **Détection des types numériques :** développer une passe de compilation capable d'identifier les variables, les expressions et les opérations utilisant des types flottants, notamment `double`.
- **Transformation des opérations :** remplacer automatiquement les opérations arithmétiques détectées par des appels aux fonctions correspondantes d'une bibliothèque spécialisée. Cette transformation pourra notamment cibler CADNA, PROMISE, MPFR ou une implémentation d'arithmétique double-simple, etc.
- **Inlining et optimisation :** étudier différentes stratégies de compilation, en comparant les appels explicites aux fonctions de bibliothèques avec leur expansion par inlining. L'objectif est notamment

d'évaluer les possibilités d'optimisation offertes par le compilateur après transformation. Le travail pourra également explorer la possibilité de rendre le processus configurable, afin de sélectionner la bibliothèque cible et la stratégie de transformation selon les besoins de l'application.

Les travaux du/de la stagiaire pourront donner lieu à des publications dans des conférences et revues nationales et internationales, ainsi qu'à des présentations lors des séminaires de l'équipe. La soutenance de stage aura lieu à l'institution d'origine de l'étudiant(e), avec une présence des encadrant(e)s de stage en visioconférence. Les encadrant(e)s américain(e)s délivreront un certificat de réussite de stage détaillé, ainsi que des lettres de recommandation pour la suite des études du/de la stagiaire. Les stages M2 peuvent donner lieu à des propositions de continuation en thèse, sous condition d'adéquation des compétences, des financements disponibles et des règles d'admission de l'École doctorale locale.

Etablissement d'accueil

L'Université du Texas à El Paso (UTEP) est une université publique américaine classée **R1** (à activité de recherche de haute catégorie) par la classification Carnegie, qui distingue les établissements ayant une très forte activité de recherche. Située au Texas, à la frontière entre les États-Unis et le Mexique, elle accueille plus de 26000 étudiants et propose 169 formations de licence, de master et de doctorat. Elle se distingue par son engagement en faveur de l'égalité des chances et de l'accès à un enseignement supérieur de qualité. L'UTEP figure parmi les 5% d'universités américaines les mieux classées en matière de recherche. Son département d'informatique, rattaché au *Miguel A. Loya College of Engineering*, propose des formations en informatique du niveau licence au doctorat, et collabore avec des partenaires académiques et industriels. Ses principaux axes de recherche couvrent le génie logiciel, la cybersécurité, l'intelligence artificielle et les fondements théoriques de l'informatique.

Visa et mobilité

La réalisation du stage à l'Université du Texas à El Paso (UTEP) nécessite l'obtention d'un visa J-1. L'établissement pourra accompagner le/la candidat(e) dans ses démarches administratives, mais les frais liés à la demande de visa et aux formalités associées resteront à sa charge. L'accueil en stage est conditionné à l'obtention du visa requis.

Durée, rémunération et frais de scolarité

Le stage, d'une durée de quatre à six mois, n'est pas rémunéré. Aucun frais de scolarité (tuition fees) ne sera demandé par l'UTEP au stagiaire dans le cadre de ce stage, et une attestation de stage sera délivrée à l'issue de ce dernier. Les encadrant(e)s aideront le/la stagiaire pour toute démarche pour obtenir une bourse de mobilité internationale.

Encadrement

Le (la) stagiaire sera encadré(e) par :

- Christoph Lauter – Maître de conférences en informatique (Associate Professor), HDR, à l'Université de l'UTEP (`https://www.christoph-lauter.org/`)
- Hanane Benmaghnia – Post-Doctorante en validation numérique au LIP6, Sorbonne Université – CNRS, future Chargée de recherche à l'UTEP, (`https://hanane-benmaghnia.github.io/`)
- Marek Felšöci – associate professor in high-performance computing at LI-PaRAD, Université de Versailles–St. Quentin en Yvelines (`https://felsoci.sk`)

Profil recherché

Le stage s'adresse à des étudiant(e)s (M1 ou M2) en informatique, mathématiques appliquées, électronique ou dans un domaine connexe en premier lieu, mais est ouvert aussi aux étudiant(e)s d'autres disciplines s'ils/elles peuvent démontrer une forte capacité de travail interdisciplinaire. Les compétences recherchées sont :

- Modélisation de problèmes découlant des différentes disciplines des science à l'aide d'outils mathématiques et informatiques.
- Intérêt pour les compilateurs et les représentations intermédiaires. Une première expérience avec LLVM ou MLIR serait appréciée.
- Connaissances de base en arithmétique flottante et en analyse numérique.
- Compétences en programmation C/C++ et en génie logiciel basique.
- Curiosité pour aborder une problématique de recherche dans un contexte international, et ouverture d'esprit pour la découverte de la langue anglaise, utilisée dans un contexte universitaire de recherche.

La curiosité scientifique, l'autonomie, la rigueur et la capacité à analyser des résultats expérimentaux seront particulièrement appréciées.

Modalités de candidature

Les candidatures doivent être envoyées à Christoph Lauter (cqlauter@utep.edu), Hanane Benmaghnia (hanane.benmaghnia@lip6.fr) et Marek Felšöci (marek.felsoci@uvsq.fr). Elles doivent inclure :

- un curriculum vitae,
- une lettre de motivation,
- des liens vers des projets personnels, professionnels ou universitaires représentatifs, par exemple des dépôts Git (si disponibles) ou publications antérieures.
- une ou deux personnes référentes à contacter.

References

[1] IEEE Standard for Floating-Point Arithmetic.

[2] Hanane Benmaghnia, Marek Felšoci, Fabienne Jézéquel, and Stef Graillat. How many digits can you trust? numerical validation and instability localization in a real-world application. In *Proceedings of the 30th International Conference on Engineering of Complex Computer Systems (ICECCS)*, Brisbane, Australia, November 2026.

[3] Sylvain Chevillard, Mioara Joldes, and Christoph Quirin Lauter. Sollya: An environment for the development of numerical codes. In Komei Fukuda, Joris van der Hoeven, Michael Joswig, and Nobuki Takayama, editors, *Mathematical Software - ICMS 2010, Third International Congress on Mathematical Software, Kobe, Japan, September 13-17, 2010. Proceedings*, volume 6327 of *Lecture Notes in Computer Science*, pages 28–31. Springer, 2010.

[4] Pacôme Eberhart, Julien Brajard, Pierre Fortin, and Fabienne Jézéquel. High performance numerical validation using stochastic arithmetic. *Reliable Computing*, 21:35–52, 2015.

[5] Laurent Fousse, Guillaume Hanrot, Vincent Lefevre, Patrick Pelissier, and Paul Zimmermann. MPFR: A multiple-precision binary floating-point library with correct rounding. *ACM Transactions on Mathematical Software*, 33(2), 2007.

[6] Stef Graillat, Fabienne Jezequel, Romain Picot, Francois Fevotte, and Bruno Lathuiliere. Auto-tuning for floating-point precision with discrete stochastic arithmetic. *Journal of Computational Science*, 36:101017, 2019.

[7] Chris Lattner and Vikram Adve. LLVM: A compilation framework for lifelong program analysis and transformation. In *Proceedings of the International Symposium on Code Generation and Optimization*, pages 75–86, 2004.

[8] Chris Lattner, Mehdi Amini, Uday Bondhugula, Albert Cohen, Andy Davis, Jacques Pienaar, River Riddle, Tatiana Shpeisman, Nicolas Vasilache, and Oleksandr Zinenko. MLIR: Scaling compiler infrastructure for domain specific computation. In *Proceedings of the IEEE/ACM International Symposium on Code Generation and Optimization*, pages 2–14, 2021.