

**Projet ANR-06-BLAN-0257**

**EVA-Flo**

**Évaluation et Validation Automatique pour le calcul Flottant**

Programme Blanc 2006

<b>A IDENTIFICATION</b>	<b>2</b>
<b>B RÉSUMÉ CONSOLIDÉ PUBLIC</b>	<b>2</b>
B.1 EVA-Flo : maîtriser les erreurs d'arrondi commises par votre ordinateur	2
B.2 EVA-Flo : taming roundoff errors done by your computer.....	5
<b>C MÉMOIRE SCIENTIFIQUE</b>	<b>7</b>
C.1 Résumé du mémoire	7
C.2 Enjeux et problématique	7
C.3 Approche scientifique et technique, état de l'art	8
C.4 Résultats obtenus	10
C.5 Exploitation des résultats	11
C.6 Discussion	11
C.7 Conclusions	12
C.8 Références	12
<b>D LISTE DES LIVRABLES</b>	<b>13</b>
<b>E IMPACT DU PROJET</b>	<b>14</b>
E.1 Indicateurs d'impact	14
E.2 Liste des publications et communications	15
E.3 Liste des éléments de valorisation	22
E.4 Bilan et suivi des personnels recrutés en CDD (hors stagiaires)	24

## A IDENTIFICATION

Acronyme du projet	EVA-Flo
Titre du projet	Évaluation et Validation Automatique pour le calcul Flottant
Coordinateur du projet (société/organisme)	INRIA, centre de recherche de Grenoble - Rhône-Alpes
Période du projet (date de début – date de fin)	06-11-2006 - 05-11-2010
Site web du projet, le cas échéant	<a href="http://www.ens-lyon.fr/LIP/Arenaire/EVA-Flo/index.html">http://www.ens-lyon.fr/LIP/Arenaire/EVA-Flo/index.html</a>

Rédacteur de ce rapport	
Civilité, prénom, nom	Mme Nathalie Revol
Téléphone	04 72 72 85 83
Adresse électronique	Nathalie.Revol@ens-lyon.fr
Date de rédaction	24-06-2011

Liste des partenaires présents à la fin du projet (société/organisme et responsable scientifique)	<ul style="list-style-type: none"><li>• INRIA Grenoble – Rhône-Alpes, responsable scientifique pour EVA-Flo : Nathalie Revol</li><li>• DALI (qui a changé d'affiliation : ELIAUS, Université de Perpignan Via Domitia au début de ce projet, LIRMM, Université Montpellier 2 en fin), responsable scientifique pour EVA-Flo : Marc Daumas</li><li>• CEA-LIST, équipe MeaSI, responsable scientifique en début de ce projet : Matthieu Martel (parti pour Perpignan) puis Éric Goubault</li><li>• INRIA Sophia-Antipolis – Méditerranée, équipe Tropics, responsable scientifique pour EVA-Flo : Laurent Hascoët</li></ul>
---	---

## B RÉSUMÉ CONSOLIDÉ PUBLIC

### B.1 EVA-FLO : MAÎTRISER LES ERREURS D'ARRONDI COMMISES PAR VOTRE ORDINATEUR

#### *Une exigence de qualité pour les calculs numériques en arithmétique flottante*

Lors du passage d'une formulation mathématique à un calcul numérique sur ordinateur, on aimerait que les résultats calculés soient proches des valeurs exactes. Or les ordinateurs calculent le plus souvent en arithmétique flottante : ils utilisent une représentation de taille finie et fixée pour les nombres et commettent donc, entre autres, des erreurs d'arrondi. Le premier objectif du projet EVA-Flo est d'évaluer numériquement une formule mathématique, de façon rapide et précise. On veut pouvoir spécifier la qualité du résultat calculé, par exemple une erreur absolue ou relative entre le résultat calculé et le résultat exact, ou la garantie qu'aucun débordement ne se produit (nombres trop grands, arrondis en , ou trop petits)...

Le deuxième objectif est que cette qualité soit quantifiée (par exemple « erreur relative  $10^{-14}$  ») et garantie, le dernier objectif étant que ce processus d'évaluation et de validation soit automatisé.

Les formules mathématiques visées s'expriment à l'aide d'opérations arithmétiques ou algébriques et de fonctions mathématiques ( $\exp$ ,  $\sin$ ,  $\operatorname{atanh}$ ...), ainsi que de branchements conditionnels et de boucles. Il s'agit typiquement de petites portions critiques de plus gros codes numériques.

### ***Automatiser, automatiser, automatiser l'expertise acquise***

De nombreux exemples ont été traités à la main par le passé, le stade actuel est d'automatiser l'expertise ainsi acquise pour chacune des étapes. Tout d'abord, il s'agit de définir précisément le résultat mathématique cherché et de construire de bons approximants (faible erreur relative) qui conduisent à une implantation adaptée à un calcul sur ordinateur, typiquement des polynômes à coefficients flottants. Ensuite on détermine des schémas d'évaluation rapides et précis pour ces approximants, par exploration exhaustive ; il faut ici tenir compte des contraintes architecturales. Pour atteindre une précision satisfaisante, différentes techniques sont envisageables : utilisation de double-double, de schémas compensés... La mise au point de la preuve de la qualité du résultat ainsi calculé fait appel à des propriétés fines de l'arithmétique flottante, à des calculs en arithmétique par intervalles et en arithmétique à précision étendue... Ces preuves sont établies de façon à pouvoir être vérifiées par un assistant de preuve (Coq dans notre cas) : en effet ces preuves, impliquant beaucoup de cas particuliers, sont sujettes aux oublis et erreurs, d'où l'importance de les vérifier automatiquement.

### ***Résultats majeurs du projet***

Notre objectif d'automatisation s'est concrétisé par le développement de logiciels nouveaux (voir en section C.4) : Sollya, Gappa, CRLibm, CGPE, Flip, ou pré-existants : Fluctuat, Tapenade. Ces travaux vont de la recherche de bons approximants et de bons schémas d'évaluation pour ces approximants, jusqu'à l'implantation sur des architectures spécifiques (DSP sans unité flottante, GPU) en passant par la preuve formelle.

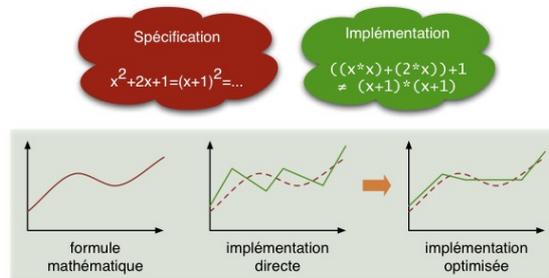
Un autre résultat est l'élaboration de LEMA, un langage de représentation étendant MathML pour l'arithmétique flottante.

Enfin, les membres du projet EVA-Flo ont mis au point d'autres projets après EVA-Flo, tels que Sardanes, TaMaDi et HPAC.

### ***Production scientifique et brevets depuis le début du projet***

Outre les logiciels déjà cités, neuf thèses ou habilitation ont été soutenues pendant la période et sur des thématiques en lien avec le projet. Une vingtaine d'articles de revues et une soixantaine de présentations lors de conférences internationales ont été produits (voir la liste en section E.2). Enfin, l'expertise du groupe sur l'arithmétique flottante a donné lieu à un livre collectif, *Handbook of Floating-Point Arithmetic*, paru en 2010. À noter aussi, la participation active aux groupes de normalisation IEEE 754 (arithmétique flottante) et IEEE 1788 (arithmétique par intervalles), avec présidence de ce dernier groupe.

### ***Illustration***



Légende : La formulation mathématique, par exemple une fonction (à gauche) peut donner lieu à un implantation qui en est une mauvaise approximation (au centre). On aimerait obtenir une meilleure implantation (à droite) et de façon la plus automatisée possible.

### **Informations factuelles**

Le projet EVA-Flo est un projet de recherche fondamentale du programme Blanc de l'ANR, coordonné par Nathalie Revol. Les équipes partenaires sont Arénaire (équipe-projet INRIA, LIP, École Normale Supérieure de Lyon), DALI (laboratoire Eliaus, Université de Perpignan

Via Domitia), MeaSI (laboratoire CEA-LIST à Saclay), Tropics (équipe-projet INRIA, INRIA Sophia-Antipolis – Méditerranée). Le projet s'est déroulé de novembre 2006 à novembre 2010. Il a bénéficié d'une aide ANR de 130 500 € pour un coût global de 1 570 577 €.

## **B.2 EVA-Flo : TAMING ROUND-OFF ERRORS DONE BY YOUR COMPUTER**

### ***Demanding quality for numerical computations using floating-point arithmetic***

When going from a mathematical formula to a numerical calculation on a computer, one aims at getting a computed result which is close to the exact (unknown) value. However, this is not straightforward, as computers usually employ floating-point arithmetic: numbers are represented using a finite storage, of fixed size, and thus every operation entails a rounding error or another error (overflow...).

The first goal of the EVA-Flo project is to be able to evaluate numerically a mathematical formula, fast and accurately. It should be possible to specify the quality of the result, for instance the – relative or absolute – error between the computed result and the exact result, or the guarantee that no overflow nor underflows occurs (numbers too large to be represented and rounded into  $\infty$ , or too small)... The second goal is that this quality is quantified (for instance « relative error  $< 10^{-14}$  ») and certified. The last goal of the project is that this process of evaluation and validation is automated.

The target mathematical formulas are given by arithmetic or algebraic operations and by mathematical functions (exp, sin, atanh...), and also by conditional branches and loops. Typically, small but critical portions of larger numerical codes are singled out.

### ***Automate, automate, automate the human expertise***

Numerous examples have been handled « manually » in the past, we now wish to automate the human expertise acquired in handling these cases, and each step of this process. First, the sought mathematical result must be precisely defined, and good approximants (with small relative error) must be built, that must be suited to an implementation on a computer: typically, polynomials with fixed- or floating-point coefficients are the right choice. Then, evaluation schemes that yield accurate results and performances in terms of execution resources (time and space) are determined, via exhaustive exploration; architectural constraints have to be taken into account. To get a satisfying accuracy, several techniques are considered: use of double-double arithmetic, of compensated algorithms... Establishing the proof that the computed result matches the quality requirements implies to master the features and properties of floating-point arithmetic and to resort to interval arithmetic, to extended precision... Such a proof is elaborated in such a way that it can be checked by a proof assistant (mainly Coq in our work). Indeed, these proofs usually contain many cases and it is thus difficult to ensure that no case has been omitted, except if a proof-checker has been applied.

### ***Main achievements of the EVA-Flo project***

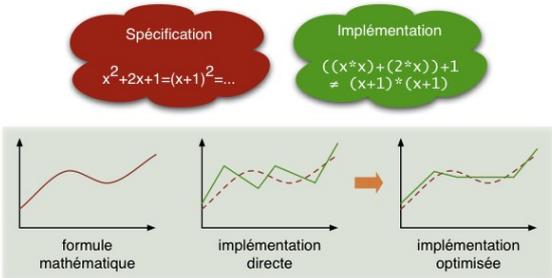
Our goal of automation has been reached through the development of new pieces of software (see Section C.4) : Sollya, Gappa, CRLibm, CGPE, Flip, or pre-existing ones : Fluctuat, Tapenade. These developments range from the determination of good approximants and of good evaluation schemes for these approximants, to the implementation on specific architectures (DSPs without floating-point unit, GPUs), via formal proof.

Another result is the elaboration of LEMA, a representation language that extends MathML to encompass floating-point arithmetic.

Eventually, members of the EVA-Flo project are at the origin of new projects, such as Sardanes, TaMaDi and HPAC.

### ***Scientific production***

Apart from the software development mentioned above, nine PhD theses or habilitation, on topics related to EVA-Flo, have been defended during this period. Around twenty journal articles and sixty presentations at international conferences have been produced (see Section E.2). The expertise of the group on floating-point arithmetic has led to the collective book, *Handbook of Floating-Point Arithmetic*, published in 2010. Let us also note the active part taken in standardization groups: IEEE 754 for floating-point arithmetic and IEEE 1788 for interval arithmetic, with the chair of this group belonging to EVA-Flo.



**Figure**

Caption: The mathematical formula, such as a function (on the left) can lead to an implementation of poor approximating quality (center). The goal is to get the better implementation (shown on the right) and as automatically as possible.

### **Relevant facts**

The EVA-Flo project is a project for fundamental research from the « Blanc » program of ANR. Its coordinator is Nathalie Revol. Participants to this project are Arenaire (INRIA team-project, LIP, École Normale Supérieure de Lyon), DALI (Eliaus laboratory, Université de Perpignan Via Domitia), MeaSI (CEA-LIST laboratory in Saclay), Tropics (INRIA team-project, INRIA Sophia-Antipolis – Méditerranée). The project EVA-Flo started in November 2006 and ended in November 2010. The total cost was 1 570 577 € and the subsidy from ANR was 130 500 €.

## **C MÉMOIRE SCIENTIFIQUE**

**Mémoire scientifique confidentiel** : non

### **C.1 RÉSUMÉ DU MÉMOIRE**

Lors du passage d'une formulation mathématique à un calcul numérique sur ordinateur, on aimerait que les résultats calculés soient proches des valeurs exactes. Or les ordinateurs calculent le plus souvent en arithmétique flottante : ils utilisent une représentation de taille finie et fixée pour les nombres et commettent donc, entre autres, des erreurs d'arrondi. Le premier objectif du projet EVA-Flo est d'évaluer numériquement une formule mathématique, de façon rapide et précise. On veut pouvoir spécifier la qualité du résultat calculé, par exemple une erreur absolue ou relative entre le résultat calculé et le résultat exact, ou la garantie qu'aucun débordement ne se produit (nombres trop grands, arrondis en , ou trop petits)...

Le deuxième objectif est que cette qualité soit quantifiée (par exemple « erreur relative  $10^{-14}$  ») et garantie, le dernier objectif étant que ce processus d'évaluation et de validation soit automatisé.

Les formules mathématiques visées s'expriment à l'aide d'opérations arithmétiques ou algébriques et de fonctions mathématiques (exp, sin, atanh...), ainsi que de branchements conditionnels et de boucles. Il s'agit typiquement de petites portions critiques de plus gros codes numériques.

### **C.2 ENJEUX ET PROBLÉMATIQUE**

Le projet EVA-Flo s'intéresse à la manière dont une formule mathématique (incluant des fonctions transcendantes et peut-être des itérations) se retrouve évaluée en virgule flottante dans un calculateur. Ce projet attaque cette question sous trois angles: comment évaluer une telle formule, comment automatiser la génération de code pour cette évaluation, comment valider la précision numérique du code obtenu.

Il s'agit d'améliorer la situation suivante : un programmeur expert sait écrire un code de qualité tenant compte par exemple des propriétés mathématiques des fonctions utilisées, des ordres de grandeurs et de l'accumulation des erreurs d'arrondis, et de propriétés fines de

l'arithmétique virgule flottante utilisée, mais ne dispose pas des outils permettant de transmettre cette expertise au compilateur.

Le compilateur se contente d'assembler des opérateurs de base et des appels de fonctions de bibliothèques sans s'intéresser à la précision numérique du code produit. Quant à obtenir une garantie sur cette précision, c'était envisageable uniquement *a posteriori*, sur de tout petits codes, et de manière essentiellement manuelle.

Notre approche a été de travailler en parallèle sur trois aspects : l'étude d'algorithmes d'évaluation de formules mathématiques (du point de vue précision et performance), la validation de tels algorithmes (notamment des aspects numériques) au cours de leur développement, afin de guider les choix algorithmiques, enfin l'automatisation de l'ensemble, point pour lequel la difficulté essentielle consiste à faire coopérer les composants utilisés pour chaque étape.

### **C.3 APPROCHE SCIENTIFIQUE ET TECHNIQUE, ÉTAT DE L'ART**

Du point de vue algorithmique, on a travaillé par exemple sur l'approximation d'une formule par un polynôme à une variable ou, dans une moindre mesure, à plusieurs variables. Si les mathématiques sont bien établies dans ce domaine (« algorithme » de Remez qui fournit une suite de polynômes convergeant vers le polynôme mathématique de meilleure approximation), leur adaptation au monde du calcul flottant est souvent non triviale. Par exemple, on peut construire le « meilleur polynôme » théorique d'approximation d'une fonction, dans un domaine et pour une norme donnés, mais obtenir le meilleur polynôme avec la contrainte que les coefficients sont exactement représentables en virgule flottante et en tenant compte de l'erreur d'arrondi due à l'évaluation du polynôme (et non plus seulement de l'écart entre la fonction et la valeur exacte que prend le polynôme) reste un problème ouvert, que l'on a attaqué sous divers angles. Mentionnons ici la programmation linéaire ainsi que l'expression de ce problème comme la recherche d'un plus proche vecteur dans un réseau euclidien.

De plus, il existe une grande variété de schémas d'évaluation d'un polynôme donné, certains minimisant les erreurs d'arrondis, d'autres maximisant le parallélisme, etc. Nous avons mis au point des méthodes d'exploration exhaustive des schémas d'évaluation de polynômes ou, plus récemment avec l'émergence du projet Sardanes, d'expressions SCADÉ, ainsi que des heuristiques permettant de limiter la taille de l'espace exploré. Cette exploration emprunte, au calcul formel, des techniques de manipulation symbolique ainsi que la problématique de la complexité des algorithmes mis en œuvre. Les critères pris en compte lors de cette exploration peuvent être l'erreur numérique finale ou des contraintes architecturales. En effet, nos processeurs cibles sont aussi bien des processeurs usuels que des GPU, des FPGA ou des DSP ne disposant pas d'unité flottante (dans le cadre d'une collaboration avec STMicroelectronics). En particulier, l'exploitation de l'ILP (Instruction Level Parallelism) a été étudiée et optimisée. Cependant notre objectif majeur reste la qualité numérique, à la différence de projets tels que Atlas [Atlas], Spiral [Spiral] ou FFTW [FFTW] qui privilégient le temps d'exécution.

Outre cet aspect algorithmique fondamental, il s'est agi essentiellement de généraliser et mécaniser une expertise acquise par les membres du projet, dans le cadre notamment de l'évaluation des fonctions élémentaires, avec la mise au point de la bibliothèque de fonctions mathématiques avec arrondi correct CRLibm, et de la géométrie algorithmique, en collaboration notamment avec D. Michelucci (U. Bourgogne). Avec la mise au point de la norme IEEE 754 sur l'arithmétique flottante, les nombres et les opérations flottants sont devenus des objets mathématiques avec une spécification précise. Soulignons ici l'expertise des membres du projet en arithmétique flottante, comme l'illustre leur très forte participation à la révision de la norme IEEE 754, publiée en 2008, avec 7 références sur 22 issues de membres du projet. Cette norme a permis de définir en quel sens les nombres flottants sont une approximation des nombres réels, ce qu'est une erreur d'arrondi, et par conséquent en quel sens un programme numérique sur les flottants est l'approximation d'un algorithme sur

les réels. Par exemple il est possible de calculer, en n'utilisant que des opérations flottantes, l'erreur d'arrondi – qui est représentable par un nombre flottant – d'une opération flottante. De tels calculs sont au cœur des algorithmes dits « compensés », initialement proposés par Kahan en 1965 ou Babuška [Bab69] et actuellement développés par Rump, Ogita et Oishi [ROO08, ORO05] et des membres du projet : les erreurs d'arrondi sont explicitement calculées et utilisées pour corriger les résultats, afin d'améliorer la précision numérique. Il est également possible d'analyser très finement un code, et même de mettre au point une preuve formelle, vérifiable par un assistant de preuve, des propriétés sur les bits manipulés par ce code. Ceci peut être comparé avec l'approche du projet ANR FoST, qui considère des programmes C annotés, les annotations sur les valeurs flottantes offrant moins d'expressivité et de niveau de détail que dans notre travail. Cependant, pour y parvenir, il faut être très soigneux et attentif au moindre détail, ce qui explique que cela n'avait été fait que pour de tout petits programmes, comportant quelques opérations flottantes. Ces analyses étaient « manuelles », au cas par cas pour chaque programme, ce qui les rendait à la fois sujettes à caution (du fait que ces études comportent typiquement de très nombreux cas particuliers et qu'il est facile à un humain d'en oublier un), chronophages et non réutilisables, à refaire entièrement à chaque modification du code étudié. Les outils dont nous disposons et que nous maîtrisons au sein du projet sont divers et complémentaires. Un premier levier pour obtenir une précision donnée est le choix de la précision de calcul : précision fixe et calculs très rapides de l'arithmétique flottante, précision double-double basée sur l'arithmétique flottante, précision étendue mais qui reste limitée, précision arbitraire quand le besoin s'en fait vraiment sentir. Un autre outil qui permet de borner les erreurs est l'arithmétique par intervalles, sur GPU ou sur processeur usuel, et ses variantes, telles que l'arithmétique affine utilisée dans Fluctuat ou les modèles polynomiaux (modèles de Taylor et leur implantation en arithmétique flottante comme nous l'avons étudiée en collaboration avec l'équipe de COSY [RMB05] et preuve formelle en Coq de ceci [CNR10], ou bien modèles de Chebyshev pour la recherche de bons approximaux polynomiaux). Enfin, la détermination des conditionnements, qui indiquent la sensibilité d'un problème à des perturbations de ses entrées, peut être abordée à l'aide de la différentiation automatique du code résolvant ce problème. L'objectif d'EVA-Flo était donc d'automatiser l'expertise acquise par les membres du projet pour ces analyses et de la généraliser à de plus gros codes flottants. À la différence des travaux autour de Cadna [Vig04] qui estiment la précision de façon probabiliste, notre approche est essentiellement déterministe. Une analyse statique du code telle que celle effectuée par Fluctuat (dont les développeurs participent à EVA-Flo) permet de traiter de plus gros codes flottants, à nouveau au détriment de la finesse de l'analyse.

Une partie de l'automatisation était acquise, puisque des outils comme Coq, PVS ou HOL-light savent vérifier des preuves formelles incluant des aspects numériques. Harrison chez Intel [Har99], Russinoff chez AMD [Rus98] ont déjà mis en œuvre des assistants de preuve pour vérifier formellement le comportement numérique de certaines implémentations, après avoir établi manuellement ce comportement. Toutefois ces outils ne rédigent pas la preuve, et notre projet était de rechercher une classe de formules aussi large que possible pour laquelle cette rédaction d'une preuve de propriété numérique sera automatisable. Plus précisément, il s'agit de produire en parallèle le code évaluant la formule, et une preuve formelle d'une propriété de ce code, typiquement une borne de l'erreur totale entre le résultat du code et le résultat mathématique. En effet, notre objectif premier est de majorer l'erreur dans le pire des cas. Cependant, cette approche peut être trop conservatrice pour certaines applications, pour lesquelles une approche probabiliste établissant la probabilité que l'erreur soit inférieure à un seuil jugé acceptable, assortie de sa preuve formelle, peut convenir.

En général, l'automatisation complète avec garantie de solution optimale n'a pas de sens sans information supplémentaire sur le contexte de l'application, et il faut formaliser ce qu'un outil a besoin de connaître en plus de la formule à implémenter. On pense naturellement au domaine d'entrée des valeurs et à la précision finale demandée, et autres données héritées du contexte de la formule. Il faut bien plus dans le cas général : par exemple, s'il est envisageable qu'un outil vérifie un invariant de boucle, il ne pourra en général pas l'inventer. De même, il faut souvent suggérer à l'outil une identité mathématique à utiliser, etc.

Outre un prototype intégré dans un compilateur, ce projet avait pour objectif de produire une meilleure compréhension de la notion de formule et de preuve en arithmétique virgule flottante.

#### C.4 RÉSULTATS OBTENUS

Notre objectif d'automatisation s'est concrétisé par le développement de logiciels, chacun réalisant une partie du processus global allant de la formule mathématique à une implantation de qualité. Il s'agit soit de nouveaux logiciels (CRLibm, Sollya, Gappa, CGPE, FLIP, FloPoCo, Barra), soit de la poursuite du développement de logiciels pré-existants (Fluctuat, Tapenade). Parmi les nouveaux logiciels, citons :

- CRLibm (*Correctly Rounded mathematical library*, <http://lipforge.ens-lyon.fr/www/crlibm/>) : cette bibliothèque implante les fonctions mathématiques (log, tan, acosh...) avec arrondi correct. Initialement, chaque fonction était étudiée et implantée manuellement, et ceci a été de plus en plus automatisé grâce aux logiciels ci-dessous. Cette bibliothèque est à l'origine de la recommandation de la norme IEEE 754-2008 d'inclure les fonctions mathématiques avec arrondi correct. Elle est désormais utilisée par Fedora.
- Sollya (<http://sollya.gforge.inria.fr/>): ce logiciel s'utilise à la fois de façon interactive et comme une bibliothèque. Sollya automatise la recherche de bons approximaux polynomiaux à coefficients flottants, avec certaines contraintes qui peuvent être spécifiées sur les coefficients, et détermine la précision requise pour chaque coefficient, de la double précision à la triple-double.
- Gappa (Génération Automatique de Preuves de Propriétés Arithmétiques, <http://gappa.gforge.inria.fr/>) : cette bibliothèque permet de générer des preuves, vérifiables par l'assistant de preuve Coq, de la qualité numérique de formules flottantes, en regard des formules mathématiques correspondantes. Son auteur a poursuivi le développement de Gappa en dehors du projet après sa thèse.
- CGPE (Code Generation for Polynomial Evaluation, <http://cgpe.gforge.inria.fr/>) : ce logiciel explore l'espace des schémas d'évaluation possible pour un polynôme donné en entrée, de façon à assurer une qualité numérique imposée tout en exploitant au mieux un parallélisme donné par l'architecture (ILP: Instruction Level Parallelism).
- FLIP (*Floating-point Library for Integer Processors*, <http://flip.gforge.inria.fr/>) : cette bibliothèque d'arithmétique flottante pour processeurs ne disposant que d'une unité arithmétique entière (tels que certains DSP : Digital Signal Processors, ou VLIW : Very Large Instruction Word, comme ceux de STMicroelectronics) a bénéficié de Sollya et CGPE pour sa mise au point et plus de la moitié de son code a été prouvé correct, formellement, à l'aide de Gappa.
- FloPoCo (*Floating-Point Cores*, <http://flopoco.gforge.inria.fr/>) : ce logiciel génère du code pour des cœurs flottants pour des FPGA (Field-Programmable Gate Arrays), en tenant compte des contraintes architecturales de ces architectures.
- Barra (<http://gpgpu.univ-perp.fr/index.php/Barra>) : il s'agit d'un simulateur de l'exécution de calculs numériques pour une autre famille architecturale particulière, à savoir les GPU.

Ces logiciels n'ont pas été créés uniquement à l'instigation du projet EVA-Flo, cependant ils relèvent de l'optique du projet et ont été réalisés essentiellement pendant la période 2006-2010.

D'autres logiciels pré-existaient à EVA-Flo et continuent leur existence de façon indépendante. Certains de leurs développements ont cependant bénéficié du travail en commun :

- Fluctuat (<http://www-list.cea.fr/labos/fr/LSL/fluctuat/>) : ce logiciel a pour objectif l'analyse de la qualité numérique de codes scientifiques, en utilisant l'analyse statique. Il s'est enrichi d'une approche probabiliste, via les *P-boxes*, pendant le projet EVA-Flo.
- Tapenade (*On-line Automatic Differentiation Engine*, <http://tapenade.inria.fr:8080/tapenade/>) : cet outil de différentiation automatique est désormais capable de traiter des programmes C, à la demande des membres du projet, l'objectif étant de l'utiliser pour calculer les conditionnements (indicateurs de la sensibilité d'un problème).

Le projet EVA-Flo s'est également traduit par l'élaboration de LEMA, un langage de représentation étendant MathML pour l'arithmétique flottante. LEMA permet de représenter les différents objets manipulés au cours du processus allant de la formule mathématique au code généré en passant par la preuve. Il a donc pour objectif de permettre aux différents logiciels mentionnés ci-dessus de communiquer.

Ces développements logiciels correspondent à des avancées scientifiques, qui portent aussi bien sur l'approximation dans les flottants que la notion et l'amélioration de la régularité des codes numériques s'exécutant sur GPU. Ces avancées scientifiques se retrouvent détaillées dans les neuf thèses ou habilitation soutenues pendant la période et sur des thématiques en lien avec le projet, ainsi que dans la vingtaine d'articles de revues et une soixantaine de présentations lors de conférences internationales produits par les membres du projet (voir la liste en section E.2). Enfin, l'expertise du groupe sur l'arithmétique flottante a donné lieu à un livre collectif, *Handbook of Floating-Point Arithmetic*, paru en 2010. À noter aussi, la participation active aux groupes de travail IEEE 754 pour la révision de la norme pour l'arithmétique flottante (sur les 22 références bibliographiques citées par la norme, 7 proviennent des membres du projet) et IEEE 1788 pour la normalisation de l'arithmétique par intervalles (avec présidence de ce dernier groupe).

Enfin, les membres du projet EVA-Flo ont élaboré d'autres projets après EVA-Flo : citons le projet Sardanes financé par la Fondation de Recherche pour l'Aéronautique et l'Espace (2009-2012) pour l'amélioration de la qualité numérique de programmes SCADE, le projet ANR blanc 2010-2013 TaMaDi sur le dilemme du fabricant de tables et le projet ANR blanc 2011-2014 HPAC sur le calcul algébrique haute performance.

## C.5 EXPLOITATION DES RÉSULTATS

L'impact des résultats obtenus pendant ces quatre années peut se mesurer par les utilisateurs des logiciels mentionnés ci-dessus, qui débordent largement le cadre des membres présents ou passés d'EVA-Flo, par l'impact des publications, par la fertilisation de la collaboration avec STMicroelectronics sur la génération de codes flottants, par les activités de normalisation qui attestent de notre expertise et enfin par l'émergence de nouveaux projets impliquant de nouvelles collaborations.

## C.6 DISCUSSION

Le projet initial avait pour ambition l'automatisation d'une chaîne complète de traitement, allant d'une formule mathématique à son implantation précise et efficace sur machine. Les résultats obtenus – avec des moyens financiers bien inférieurs à ceux demandés, puisqu'il n'a été possible de financer qu'une année d'ingénieur et non les trois ans demandés, et il n'a pas

été possible de financer une thèse – constituent une avancée importante dans cette direction, puisque la majorité des étapes a été traitée. Il resterait à mettre au point le « chef d'orchestre » qui gèrerait l'enchaînement de ces étapes. Un autre point encore en cours, pour lequel une conclusion proche est en vue, est l'utilisation de la différentiation automatique pour l'étude de la sensibilité numérique d'un problème, via la détermination de son conditionnement.

Les projets qui succèdent à EVA-Flo portent soit sur la détermination des cas les plus difficiles à évaluer avec arrondi correct, pour les fonctions élémentaires, comme dans TaMaDi, soit sur une classe différente d'expressions à traiter, à savoir les programmes SCADÉ visés par le projet Sardanes, soit sur l'évaluation numérique ou exacte de certains problèmes sur des architectures émergentes telles que les multicœurs, comme dans le projet HPAC. Ces projets couvrent bien les perspectives ouvertes par les résultats du projet EVA-Flo.

Un aspect important du projet EVA-Flo, qui ne transparaît pas de la liste de publications mais s'est traduit par des responsabilités scientifiques communes et plus encore par des échanges de personnel, est le travail mené en commun sur nombre de sujets. Mentionnons par exemple le travail sur l'exploration de l'espace des schémas d'évaluation mené à Lyon et à Perpignan qui a conduit au logiciel CGPE et au projet Sardanes, au travail sur les algorithmes compensés (Perpignan et Lyon), à l'étude et à l'optimisation de l'ILP (Lyon et Perpignan), à l'utilisation de la différentiation automatique qui a conduit à l'ajout du C dans les langages compris par Tapenade (Sophia et Lyon), à l'utilisation de l'analyse statique pour étudier la qualité numérique d'un programme (Saclay et Perpignan), à l'utilisation de la preuve formelle (Lyon et Perpignan), aux choix qui ont conduit à LEMA et qui ont été discutés par tous les partenaires... Ces échanges et cette culture commune s'expriment également par la participation à des conférences, voire à des comités de programme pour des conférences, telles que RNC8, RAIM 2009, NSV3, SCAN 2010, PARA 2010 et PASCO 2010. De façon plus marquante, ces échanges scientifiques se traduisent par des mobilités de personnes : Matthieu Martel est allé de Saclay à Perpignan, Sylvain Collange, Guillaume Révy et Christophe Mouilleron de Lyon à Perpignan, Nicolas Louvet et Sylvain Collange de Perpignan à Lyon (Sylvain Collange, après son mastère lyonnais, a effectué son doctorat à Perpignan avant de revenir à Lyon en post-doctorat), sans compter l'accueil pour de courts séjours de stagiaires (Laurent Thévenoux de Perpignan à Lyon par exemple).

Un regret est qu'il n'a pas été possible de prendre en compte notre participation au pôle de compétitivité mondial Minalogic, puisqu'elle a été notifiée officiellement deux jours après la date limite fixée par l'ANR pour renvoyer les justificatifs (20 mai 2006).

## C.7 CONCLUSIONS

Le projet initial avait pour ambition l'automatisation d'une chaîne complète de traitement, allant d'une formule mathématique à son implantation précise et efficace sur machine. Les résultats obtenus constituent une avancée importante dans cette direction, puisque la majorité des étapes a été traitée.

Un aspect important du projet EVA-Flo, qui s'est traduit par des responsabilités scientifiques communes et plus encore par des échanges de personnel, est le travail mené en commun sur nombre de sujets.

## C.8 RÉFÉRENCES

- [Atlas] *Automatically Tuned Linear Algebra Software (ATLAS)* : <http://math-atlas.sourceforge.net/>
- [Bab69] I. Babuška. *Numerical stability in mathematical analysis*. Proceedings of the 1968 IFIP Congress, vol. 1, pages 11-23, 1969.

- [CNR10] P. Collins, M. Niqui, and N. Revol. *A Taylor function calculus for hybrid systems: validation in Coq*. NSV-3: 3<sup>rd</sup> International Workshop on Numerical Software Validation, 2010.
- [FFTW] M. Frigo, and S. G. Johnson. *The Design and Implementation of FFTW3*. Proceedings of the IEEE, 93(2):216-231, 2005.
- [Har99] J. Harrison. *A machine-checked theory of floating-point arithmetic*. Proc. Theorem Proving in Higher Order Logics, Lecture Notes in Computer Science vol. 1690, pages 113-130, 1999.
- [ORO05] T. Ogita, S. M. Rump, and S. Oishi. *Accurate sum and dot product*. SIAM Journal on Scientific Computing, 26(6):1955-1988, 2005.
- [RMB05] N. Revol, K. Makino, and M. Berz. *Taylor models and floating-point arithmetic: proof that arithmetic operations are validated in COSY*. Journal of Logic and Algebraic Programming, 64:135-154, 2005.
- [ROO08] S. M. Rump, T. Ogita, and S. Oishi. *Accurate floating-point summation. Part I: faithful rounding*. SIAM Journal on Scientific Computing, 31(1):189-224, 2008.
- [Rus98] D. M. Russinoff. *A mechanically checked proof of IEEE compliance of a register-transfer-level specification of the AMD-K7 floating-point multiplication, division and square root instructions*, LMS Journal of Computation and Mathematics, 1:148-200, 1998.
- [Spiral] *Spiral. Software/Hardware Generation for DSP Algorithms* : <http://www.spiral.net/>
- [Vig04] J. Vignes. *Discrete Stochastic Arithmetic for Validating Results of Numerical Software*, Numerical Algorithms, 37(1):377-390, 2004.

## D LISTE DES LIVRABLES

La proposition de projet EVA-Flo ne comportait pas de livrables. Voici un compte-rendu du fonctionnement du projet EVA-Flo et des jalons, année par année, qui ont marqué sa progression.

Les partenaires du projet EVA-Flo se sont réunis deux fois par an (à l'exception de la période de congé maternité de la coordinatrice et à l'exception de la fin 2010 puisque nous nous sommes retrouvés à Lyon fin septembre pour la conférence SCAN 2010). Les dates et programmes de ces réunions sont disponibles sur le site Web du projet : <http://www.ens-lyon.fr/LIP/Arenaire/EVA-Flo/>. Nous avons initialement prévu quatre réunions par an, mais le budget alloué ne l'a pas permis.

De plus, le projet EVA-Flo a servi de moteur à un travail interne à une partie lyonnaise du projet, qui a fait avancer de façon régulière (réunions bi-mensuelles) la réflexion sur le langage LEMA.

La première année devait servir à identifier les besoins en validation, les briques de base pour y répondre et les techniques nécessaires. Cela a été réalisé avec le lancement de la réflexion sur LEMA, incluant la discussion d'un langage SSA ou non, fonctionnel ou impératif... Les travaux sur la recherche des meilleurs algorithmes d'évaluation, en arithmétique flottante, d'un polynôme donné, avec exploration exhaustive des schémas possibles et mise en place d'heuristiques pour limiter l'espace exploré, ont commencé.

La deuxième année devait voir le début de la thèse, qui n'a pas eu lieu faute de financement suffisant. Elle devait aussi voir le début de l'implantation : comme une seule année

d'ingénieur n'était possible et non les trois ans demandés, cela a été reporté d'un an. Les avancées de cette année ont porté sur l'automatisation des travaux sur l'approximation et l'évaluation (développement de Sollya...), sur la prise en compte des spécificités architecturales pour produire du code performant en temps d'exécution (DSP, FPGA, GPU), sur la qualité numérique des calculs flottants (division, algorithmes compensés, arithmétique par intervalles, calcul numérique certifié, sur- et sous-approximations avec l'analyse statique), sur l'utilisation de la différentiation automatique pour le calcul de conditionnements, sur la preuve formelle avec une approche pire cas ou probabiliste.

La troisième année a vu l'embauche de deux ingénieurs. Andrew Novocin (janvier 2010 - avril 2010) a approfondi les techniques de recherche des meilleurs approximaux polynomiaux à coefficients flottants reposant sur la recherche de plus proche vecteur dans un réseau euclidien, en lien avec l'algorithme LLL. Philippe Théveny (janvier 2010 – août 2010) a essentiellement développé LEMA, langage de représentation des données flottantes (et de toute information utile indiquant le lien entre les données mathématiques et les valeurs flottantes) qui étend MathML, ainsi que les interfaces vis-à-vis de Sollya et Gappa en particulier. Il a également contribué à la bibliothèque MPFI d'arithmétique par intervalles en précision arbitraire : tests couvrant plus de 99% du code, découpage modulaire des sources, amélioration de la portabilité, ce qui a conduit à une nouvelle version en août 2010. Les partenaires ont poursuivi leurs travaux sur les aspects architecturaux (prise en compte et exploitation de l'ILP, FPGA), sur l'évaluation (CGPE, algorithmes compensés, conception du projet Sardanes), sur la vérification de calculs numériques (solutions de systèmes linéaires ou factorisation QR, utilisant ou non l'arithmétique par intervalles), sur la différentiation automatique (prise en compte du langage C et des surcharges dans Tapenade), la preuve formelle. La quatrième année a vu la poursuite de ces travaux, sans hélas aller jusqu'à une phase d'expérimentation sur de plus gros exemples et de tests plus poussés, faute de main d'œuvre.

D'une part, le projet EVA-Flo a permis une progression commune des différents partenaires sur chacun des points composant le projet, avec la construction d'une culture commune et des échanges constructifs, grâce à des réunions régulières. D'autre part, il a donné lieu à des avancées pratiques concrétisées par des développements logiciels, réalisés par les ingénieurs embauchés pour le projet et pour une large part par les partenaires d'EVA-Flo.

## E IMPACT DU PROJET

### E.1 INDICATEURS D'IMPACT

#### *Nombre de publications et de communications (à détailler en E.2)*

		Publications multipartenaires	Publications monopartenaires
<b>International</b>	<b>Revue à comité de lecture</b>		22
	<b>Ouvrages ou chapitres d'ouvrage</b>		2
	<b>Communications (conférence)</b>	3	62
<b>France</b>	<b>Revue à comité de lecture</b>		2
	<b>Ouvrages ou chapitres d'ouvrage</b>		
	<b>Communications (conférence)</b>		2
<b>Actions de diffusion</b>	<b>Articles vulgarisation</b>		
	<b>Conférences vulgarisation</b>		
	<b>Autres</b>		

Certains auteurs ont changé d'équipe en cours de projet, leurs publications ne sont pas nécessairement comptabilisées comme « multipartenaires » pour autant.

Les actions de diffusion réalisées par les membres du projet n'étant pas liées uniquement à EVA-Flo, elles ne sont pas comptabilisées dans ce bilan.

### **Autres valorisations scientifiques (à détailler en E.3)**

	<b>Nombre, années et commentaires (valorisations avérées ou probables)</b>
<b>Brevets internationaux obtenus</b>	
<b>Brevet internationaux en cours d'obtention</b>	
<b>Brevets nationaux obtenus</b>	
<b>Brevet nationaux en cours d'obtention</b>	
<b>Licences d'exploitation (obtention / cession)</b>	
<b>Créations d'entreprises ou essaimage</b>	
<b>Nouveaux projets collaboratifs</b>	
<b>Colloques scientifiques : organisation</b>	Rencontre RAIM 2009 (70 participants) Conférence SCAN 2010 (123 participants venant de 20 pays)
<b>Autres (préciser)</b>	

## **E.2 LISTE DES PUBLICATIONS ET COMMUNICATIONS**

Les auteurs participant au projet EVA-Flo sont indiqués en gras, les publications multi-partenaires en gris-bleuté.

### **International : revues à comité de lecture**

1. Sylvie Boldo, **Marc Dumas**, and Ren-Cang Li. *Formally verified argument reduction with a fused-multiply-add*. IEEE Transactions on Computers, 58(8) :1139–1145, 2009.
2. **Nicolas Brisebarre** and **Jean-Michel Muller**. *Correct rounding of algebraic functions*. Theoretical Informatics and Applications, 41 :71–83, 2007.
3. **Nicolas Brisebarre** and **Jean-Michel Muller**. *Correctly rounded multiplication by arbitrary precision constants*. IEEE Transactions on Computers, 57(2) :165–174, 2008.
4. **Nicolas Brisebarre**, **Jean-Michel Muller**, and Arnaud Tisserand. *Computing machine-efficient polynomial approximations*. ACM Transactions on Mathematical Software, 32(2) :236–256, 2006.
5. **Nicolas Brisebarre**, **Jean-Michel Muller**, Arnaud Tisserand, and Serge Torres. *Hardware operators for function evaluation using sparse-coefficient polynomials*. IEEE Electronics Letters, 42(25) :1441–1442, 2006.
6. **Sylvain Collange**, **Marc Dumas**, and **David Defour**. *Line-by-line spectroscopic simulations on graphics processing units*. Computer Physics Communications, 178(2) : 135– 143, 2008.
7. Octavian Creț, Ionut Trestian, Radu Tudoran, Laura Darabant, Lucia Vačariu, and **Florent de Dinechin**. *Accelerating the computation of the physical parameters involved in transcranial magnetic stimulation using FPGA devices*. Romanian Journal of Information, Science and Technology, 10(4) :361–379, 2008.
8. **Marc Dumas**, David Lester, and César Muñoz. *Verified real number calculations : A library for interval arithmetic*. IEEE Transactions on Computers, 58(2) :226–237, 2009.

9. **Florent de Dinechin, Christoph Quirin Lauter, and Jean-Michel Muller.** *Fast and correctly rounded logarithms in double-precision.* Theoretical Informatics and Applications, 41 :85–102, 2007.
10. **Florent de Dinechin, Bogdan Pasca, Octavian Creț, and Radu Tudoran.** *An FPGA-specific approach to floating-point accumulation and sum-of-products.* In Field-Programmable Technologies, pages 33–40. IEEE, 2008.
11. **Florent de Dinechin and Gilles Villard.** *High precision numerical accuracy in physics research.* Nuclear Inst. and Methods in Physics Research, A, 559(1) :207–210, 2006.
12. Milos Ercegovac and **Jean-Michel Muller.** *An Efficient Method for Evaluating Complex Polynomials.* Journal of Signal Processing Systems, 58 :17–27, 2010.
13. Laurent Fousse, Guillaume Hanrot, **Vincent Lefèvre, Patrick Pélessier, and Paul Zimmermann.** *MPFR : A multiple-precision binary floating-point library with correct rounding.* ACM Transactions on Mathematical Software, 33(2), 2007.
14. Stef Graillat, **Philippe Langlois, and Nicolas Louvet.** *Algorithms for accurate, validated and fast computations with polynomials.* Japan Journal of Industrial and Applied Mathematics, Special issue on Verified Numerical Computation, 26(2–3):191–214, 2009.
15. **Claude-Pierre Jeannerod, Hervé Knochel, Christophe Monat, and Guillaume Revy.** *Computing floating-point square roots via bivariate polynomial evaluation.* Computers, IEEE Transactions on, 60(2) :214–227, 2011.
16. **Claude-Pierre Jeannerod, Nicolas Louvet, Jean-Michel Muller, and Adrien Panhaleux.** *Midpoints and exact points of some algebraic functions in floating-point arithmetic.* IEEE Transactions on Computers, 60 :228–241, 2011.
17. Peter Kornerup, **Christoph Quirin Lauter, Vincent Lefèvre, Nicolas Louvet, and Jean-Michel Muller.** *Computing correctly rounded integer powers in floating-point arithmetic.* ACM Transactions on Mathematical Software, 37(1), 2009.
18. **Christoph Quirin Lauter and Vincent Lefèvre.** *An efficient rounding boundary test for  $\text{pow}(x,y)$  in double precision.* IEEE Transactions on Computers, 58(2) :197–207, 2009.
19. **Matthieu Martel.** *Enhancing the implementation of mathematical formulas for fixed-point and floating-point arithmetics.* Formal Methods in System Design, 35 :265–278, 2009.
20. **Guillaume Melquiond and Sylvain Pion.** *Formally certified floating-point filters for homogeneous geometric predicates.* RAIRO - Theoretical Informatics and Applications, 41(1) :57–69, 2007.
21. **Hong Diep Nguyen and Nathalie Revol.** *Solving and Certifying the Solution of a Linear System.* Reliable Computing, 15(2) :120–131, 2011.
22. Ionuț Trestian, Octavian Creț, Laura Creț, Lucia Vacariu, Radu Tudoran, and **Florent de Dinechin.** *FPGA-based computation of the inductance of coils used for the magnetic stimulation of the nervous system.* In Biomedical Electronics and Devices, volume 1, pages 151–155, 2008.

### **International : ouvrages ou chapitres d'ouvrage**

1. **Florent de Dinechin, Milos Ercegovac, Jean-Michel Muller, and Nathalie Revol.** Encyclopedia of Computer Science and Engineering (Benjamin W. Wah ed.), chapter *Digital Arithmetic*, pages 935–948. Wiley, 2009.
2. **Jean-Michel Muller, Nicolas Brisebarre, Florent de Dinechin, Claude-Pierre Jeannerod, Vincent Lefèvre, Guillaume Melquiond, Nathalie Revol, Damien Stehlé, and Serge Torres.** *Handbook of Floating-Point Arithmetic.* 600 pages. Birkhäuser Boston, 2010.

### **International : communications (conférences)**

1. Jean-Claude Bajard, **Philippe Langlois**, Dominique Michelucci, Géraldine Morin, and **Nathalie Revol**. *Towards guaranteed geometric computations with approximate arithmetics*. In Advanced Signal Processing Algorithms, Architectures, and Implementations XVIII, part of the SPIE Optics & Photonics 2008 Symposium, volume 7074, 12 pages, 2008.
2. Christian Bertin, **Claude-Pierre Jeannerod**, Jingyan Jourdan-Lu, Hervé Knochel, Christophe Monat, **Christophe Moulleron**, **Jean-Michel Muller**, and **Guillaume Revy**. *Techniques and tools for implementing IEEE 754 floating-point arithmetic on VLIW integer processors* (invited conference). In 4th International Workshop on Parallel and Symbolic Computation (PASCO'10), pages 1–9. ACM, 2010.
3. Sylvie Boldo, **Marc Daumas**, and Pascal Giorgi. *Formal proof for delayed finite field arithmetic using floating point operators*. In Javier D. Bruguera and Marc Daumas, editors, 8th Conference on Real Numbers and Computers (RNC8), pages 113–122, 2008.
4. **Nicolas Brisebarre** and **Sylvain Chevillard**. Efficient polynomial  $L_\infty$ -approximations. In 18th IEEE Symposium on Computer Arithmetic (ARITH-18), pages 169–176. IEEE Computer Society, 2007.
5. **Nicolas Brisebarre**, **Sylvain Chevillard**, Milos Ercegovic, **Jean-Michel Muller**, and Serge Torres. *An efficient method for evaluating polynomial and rational function approximations*. In Application-specific Systems, Architectures and Processors (ASAP 2008), pages 245–250. IEEE, 2008.
6. **Nicolas Brisebarre**, **Florent de Dinechin**, and **Jean-Michel Muller**. *Integer and floating-point constant multipliers for FPGAs*. In Application-specific Systems, Architectures and Processors (ASAP 2008), pages 239–244. IEEE, 2008.
7. **Nicolas Brisebarre** and Guillaume Hanrot. *Floating-point  $L_2$  approximations to functions*. In 18th IEEE Symposium on Computer Arithmetic (ARITH-18), pages 177–184. IEEE Computer Society, 2007.
8. **Nicolas Brisebarre** and **Mioara Joldes**. *Chebyshev interpolation polynomial-based tools for rigorous computing*. In 2010 International Symposium on Symbolic and Algebraic Computation (ISSAC'10), pages 147–154, 2010. ACM.
9. **Nicolas Brisebarre**, **Nicolas Louvet**, Erik Martin-Dorel, **Jean-Michel Muller**, Adrien Panhaleux, and Milos D. Ercegovic. *Implementing decimal floating-point arithmetic through binary : Some suggestions* (poster). In Application-specific Systems Architectures and Processors (ASAP 2010), pages 317–320, 2010.
10. **Francisco José Cháves Alonso**, **Marc Daumas**, César Muñoz, and **Nathalie Revol**. *Automatic strategies to evaluate formulas on Taylor models and generate proofs in PVS*. In 6th International Congress on Industrial and Applied Mathematics (ICIAM'07), July 2007.
11. **Sylvain Chevillard**, **Mioara Joldes**, and **Christoph Lauter**. *Certified and fast computation of supremum norms of approximation errors*. In 19th IEEE Symposium on Computer Arithmetic (ARITH-19), Portland, Oregon, U.S.A., 2009.
12. **Sylvain Chevillard** and **Christoph Quirin Lauter**. *A certified infinite norm for the implementation of elementary functions*. In A. Mathur, W. E. Wong, and M. F. Lau, editors, Seventh International Conference on Quality Software (QSIC 2007), pages 153–160. IEEE, 2007.
13. **Sylvain Chevillard** and **Nathalie Revol**. *Computation of the error function erf in arbitrary precision with correct rounding*. In Javier D. Bruguera and Marc Daumas, editors, 8th Conference on Real Numbers and Computers (RNC8), pages 27–36, 2008.
14. Pieter Collins, Milad Niqui, and **Nathalie Revol**. *A Taylor Function Calculus for Hybrid System Analysis : Validation in Coq*. In Georgios Fainekos, Eric Goubault, and Sylvie

- Putot, editors, NSV-3: Third International Workshop on Numerical Software Verification, Edinburgh, United Kingdom, 2010.
15. **Marc Daumas**, David Lester, Érik Martin-Dorel, and Annick Truffert. *Stochastic formal correctness of numerical algorithms*. In NASA Formal Methods Symposium, pages 136–145, 2009.
  16. **Florent de Dinechin**, Jérémie Detrey, Ionuț Trestian, Octavian Creț, and Radu Tudoran. *When FPGAs are better at floating-point than microprocessors*. Poster at FPGA'2008, 2008.
  17. **Florent de Dinechin**, Cristian Klein, and Bogdan Pasca. *Generating high-performance custom floating-point pipelines*. In Proceedings of the 19th International Conference on Field Programmable Logic and Applications. IEEE, 2009.
  18. **Florent de Dinechin** and Bogdan Pasca. *Large multipliers with less DSP blocks*. In Proceedings of the 19th International Conference on Field Programmable Logic and Applications. IEEE, 2009.
  19. **Jéréemie Detrey** and **Florent de Dinechin**. *Floating-point trigonometric functions for FPGAs*. In International Conference on Field-Programmable Logic and Applications, pages 29–34. IEEE, 2007.
  20. **Jéréemie Detrey**, **Florent de Dinechin**, and Xavier Pujol. *Return of the hardware floating-point elementary function*. In 18th IEEE Symposium on Computer Arithmetic (ARITH-18), pages 161–168. IEEE, 2007.
  21. Pouya Dormiani, Milos Ercegovac, and **Jean-Michel Muller**. *Design and implementation of a radix-4 complex division unit with prescaling*. In Application-specific Systems, Architectures and Processors (ASAP 2009), Boston, U.S.A., IEEE Computer Society, 2009.
  22. Milos Ercegovac and **Jean-Michel Muller**. *Complex multiply-add and other related operators*. In Proceedings of SPIE Conf. Advanced Signal Processing Algorithms, Architectures and Implementation XVII, 2007.
  23. Milos Ercegovac and **Jean-Michel Muller**. *A hardware-oriented method for evaluating complex polynomials*. In Proceedings of 18th IEEE Conference on Application-Specific Systems, Architectures and Processors (ASAP 2007). IEEE, 2007.
  24. **Bernard Goossens**, **Philippe Langlois**, and David Parello. *Processor simulation : a new way for the performance analysis of numerical algorithms*. In Dagstuhl Seminar 09471 on Computer-assisted proofs - tools, methods and applications, 2009.
  25. **Éric Goubault** and **Sylvie Putot**. *Inner and outer approximation of functionals using generalized affine forms*. In SCAN 2008 - 13th GAMM - IMACS International Symposium on Scientific Computing, Computer Arithmetic and Validated Numerics, El Paso, Texas, 2008.
  26. Stef Graillat, **Philippe Langlois**, and **Nicolas Louvet**. *Improving the compensated Horner scheme with a fused multiply and add*. In Proceedings of the 21st Annual ACM Symposium on Applied Computing, volume 2, pages 1323–1327. Association for Computing Machinery, 2006.
  27. **Claude-Pierre Jeannerod**, Hervé Knochel, Christophe Monat, and **Guillaume Revy**. *Faster floating-point square root for integer processors*. In IEEE Symposium on Industrial Embedded Systems (SIES'07), 2007.
  28. **Claude-Pierre Jeannerod**, Hervé Knochel, Christophe Monat, **Guillaume Revy**, and **Gilles Villard**. *A new binary floating-point division algorithm and its software implementation on the ST231 processor*. In J.D. Bruguera, M. Cornea, D. DasSarma, and J. Harrison, editors, 19th IEEE Symposium on Computer Arithmetic (ARITH-19), pages 95–103, Portland, Oregon, USA, 2009. IEEE Computer Society.

29. **Claude-Pierre Jeannerod, Nicolas Louvet, Nathalie Revol, and Gilles Villard.** *Computing condition numbers with automatic differentiation.* In SCAN 2008 - 13th GAMM - IMACS International Symposium on Scientific Computing, Computer Arithmetic and Validated Numerics, El Paso, Texas, 2008.
30. **Claude-Pierre Jeannerod, Nicolas Louvet, Nathalie Revol, and Gilles Villard.** *On the computation of some componentwise condition numbers.* In SNSC'08 - 4th International Conference on Symbolic and Numerical Scientific Computing, Hagenberg, Austria, 2008.
31. **Claude-Pierre Jeannerod and Guillaume Revy.** *Optimizing correctly-rounded reciprocal square roots for embedded VLIW cores.* In Asilomar'09 : Proceedings of the 43rd Asilomar Conference on Signals, Systems, and Computers, Washington, DC, USA, 2009. IEEE Computer Society.
32. R. Baker Kearfott, John D. Pryce, and **Nathalie Revol.** *Discussions on an interval arithmetic standard at Dagstuhl seminar 08021.* In Dagstuhl Seminar on Numerical Validation in Current Hardware Architectures, volume 5492 of Lecture Notes in Computer Science, pages 1–6. Annie Cuyt and Walter Krämer and Wolfram Luther and Peter Markstein editors, 2009.
33. Peter Kornerup, **Vincent Lefèvre, Nicolas Louvet, and Jean-Michel Muller.** *On the computation of correctly-rounded sums.* In 19th IEEE Symposium on Computer Arithmetic (ARITH-19), Portland, Oregon, U.S.A., 2009.
34. Peter Kornerup, **Vincent Lefèvre, and Jean-Michel Muller.** *Computing integer powers in floating-point arithmetic.* In Proceedings of 41th Conference on signals, systems and computers. IEEE Conference Publishing Services, 2007.
35. **Philippe Langlois.** *Compensated algorithms and validated bounds.* In SWIM 08, 2008.
36. **Philippe Langlois.** *Performance analysis of some accurate and validated algorithms (conférence invitée).* In International Workshop on Verified Numerical Computations and its Applications, Miyako, Japan, 2008.
37. **Philippe Langlois, Stef Graillat, and Nicolas Louvet.** *Compensated Horner scheme.* In Bruno Buchberger, Shin'ichi Oishi, Michael Plum, and Siegfried M. Rump, editor, Algebraic and Numerical Algorithms and Computer-assisted Proofs, Dagstuhl seminar number 05391, 2006.
38. **Philippe Langlois and Nicolas Louvet.** *Operator dependent compensated algorithms.* In Proceedings of the 12th GAMM - IMACS International Symposium on Scientific Computing, Computer Arithmetic, and Validated Numerics, Duisburg, Germany, pages 1–11, 2006.
39. **Philippe Langlois and Nicolas Louvet.** *Faithful Horner algorithm.* In 6th International Congress on Industrial and Applied Mathematics, Zurich, Switzerland, page 25. SIAM, 2007.
40. **Philippe Langlois and Nicolas Louvet.** *How to ensure a faithful polynomial evaluation with the compensated Horner algorithm?* In 18th IEEE International Symposium on Computer Arithmetic (ARITH-18), pages 141–149. IEEE Computer Society, 2007.
41. **Philippe Langlois and Nicolas Louvet.** *Accurate solution of triangular linear system.* In SCAN 2008 - 13th GAMM - IMACS International Symposium on Scientific Computing, Computer Arithmetic and Validated Numerics, El Paso, Texas, 2008.
42. **Philippe Langlois and Nicolas Louvet.** *Compensated Horner algorithm in  $K$  times the working precision.* In Javier D. Bruguera and Marc Daumas, editors, 8th Conference on Real Numbers and Computers (RNC8), pages 157–166, 2008.
43. **Philippe Langlois, Matthieu Martel, and Laurent Thévenoux.** *Accuracy Versus Time : A Case Study with Summation Algorithms.* In 4th International Workshop on Parallel and Symbolic Computation (PASCO'10), pages 121–130. ACM, 2010.

44. **Christoph Quirin Lauter** and **Florent de Dinechin**. *Optimising polynomials for floating-point implementation*. In Javier D. Bruguera and Marc Daumas, editors, 8th Conference on Real Numbers and Computers (RNC8), pages 7–16, 2008.
45. **Vincent Lefèvre**, Damien Stehlé, and Paul Zimmermann. *Worst cases for the exponential function in the IEEE 754r decimal64 format*. In *Reliable Implementation of Real Number Algorithms : Theory and Practice*, volume 5045 of *Lecture Notes in Computer Science*, pages 114–126, Dagstuhl, Germany, 2008.
46. **Vincent Lefèvre**, **Philippe Théveny**, **Florent de Dinechin**, **Claude-Pierre Jeannerod**, **Christophe Moulleron**, David Pfannholzer, and **Nathalie Revol**. *LEMA : towards a language for reliable arithmetic*. *ACM Commun. Comput. Algebra*, 44 :41–52, 2010.
47. **Matthieu Martel**. *Semantics-based transformation of arithmetic expressions*. In *Static Analysis Symposium, SAS'07*, number 4634 in *Lecture Notes in Computer Science*, pages 298–314. Springer-Verlag, 2007.
48. **Matthieu Martel**. *Program transformation for numerical precision*. In *ACM Workshop on Partial Evaluation and Program Manipulation, PEPM'09*, pages 101–109. ACM Press, 2009.
49. **Christophe Moulleron** and **Guillaume Revy**. *Automatic Generation of Fast and Certified Code for Polynomial Evaluation*. In *20 IEEE Symposium on Computer Arithmetic (ARITH-20)*, 2011.
50. **Jean-Michel Muller**. *Exact computations with an arithmetic known to be approximate*. Invited lectures (3 hours) at the conference *Numeration : Mathematics and Computer Science*, CIRM, Marseille, 2009.
51. **Jean-Michel Muller**. *Use of FMAs for accurate floating-point arithmetic*. Invited seminar, University of Malaga, Spain, 2006.
52. **Jean-Michel Muller**. *Some algorithmic improvements due to the availability of an FMA*. In *Dagstuhl seminar 08021, Numerical Validation in Current Hardware Architectures*, 2008.
53. **Hong Diep Nguyen**. *Efficient implementation of interval matrix multiplication*. In *PARA 2010 : State of the Art in Scientific and Parallel Computing*, 2010.
54. **Hong-Diep Nguyen** and **Nathalie Revol**. *Solving and certifying the solution of a linear system*. In *SCAN 2008 - 13th GAMM - IMACS International Symposium on Scientific Computing, Computer Arithmetic and Validated Numerics*, El Paso, Texas, 2008.
55. **Hong Diep Nguyen** and **Nathalie Revol**. *An interval iterative refinement method to certify the solution of a linear system*. In *Dagstuhl Seminar 09471 on Computer-assisted proofs, tools, methods and applications*, 2009.
56. **Hong Diep Nguyen** and **Nathalie Revol**. *Accuracy issues in linear algebra using interval arithmetic*. In *SCAN 2010 : 14th GAMM-IMACS International Symposium on Scientific Computing, Computer Arithmetic and Validated Numerics*, 2010.
57. **Hong Diep Nguyen** and **Nathalie Revol**. *Certification of a Numerical Result : Use of Interval Arithmetic and Multiple Precision*. In Georgios Fainekos, Eric Goubault, and Sylvie Putot, editors, *NSV-3 : Third International Workshop on Numerical Software Verification*, 2010.
58. **Hong Diep Nguyen** and **Nathalie Revol**. *High performance linear algebra using interval arithmetic*. In *4th International Workshop on Parallel and Symbolic Computation (PASCO'10)*, pages 171–172. ACM Digital Library, 2010.
59. **Andrew Novocin**, Damien Stehlé, and **Gilles Villard**. *An LLL-reduction algorithm with quasi-linear time complexity*, *Symposium on Theory of Computing*, 2011.
60. **Nathalie Revol**. *Implementing Taylor models arithmetic with floating-point arithmetic*. In *6th International Congress on Industrial and Applied Mathematics (ICIAM'07)*, 2007.

61. **Nathalie Revol.** *Automatic adaptation of the computing precision.* In Dagstuhl seminar 08021 on Numerical Validation in Current Hardware Architectures, 2008.
62. **Nathalie Revol.** *Automatic adaptation of the computing precision.* In V Taylor Models Workshop, 2008.
63. **Nathalie Revol.** *Introduction to interval analysis and to some interval-based software systems and libraries* (invited conference). In ECMI 2008, The European Consortium For Mathematics In Industry, 2008.
64. **Nathalie Revol.** *Survey of proposals for the standardization of interval arithmetic.* In SWIM 08 : Small Workshop on Interval Methods, 2008.
65. **Nathalie Revol.** *Standardized Interval Arithmetic and Interval Arithmetic Used in Libraries.* In ICMS 2010 - Third International Congress on Mathematical Software, volume 6327, pages 337–341. Springer, 2010.
66. **Gilles Villard.** *Certification of the QR factor R, and of lattice basis reducedness.* In International Symposium on Symbolic and Algebraic Computation (ISSAC 2007), pages 361–368, Waterloo, Canada, August 2007. ACM Press.

#### **France : revues à comité de lecture**

1. **Sylvain Collange, Marc Daumas, and David Defour.** *État de l'intégration de la virgule flottante dans les processeurs graphiques.* *Technique et Science Informatiques*, 27(6) :719–733, 2008.
2. **Jérémie Detrey and Florent de Dinechin.** *Fonctions élémentaires en virgule flottante pour les accélérateurs reconfigurables.* *Technique et Science Informatiques*, 27(6) :673–698, 2008.

#### **France : communications (conférences)**

1. **Nicolas Brisebarre, Florent de Dinechin, and Jean-Michel Muller.** *Multiplieurs et diviseurs constants en virgule flottante avec arrondi correct.* In RenPar'18, SympA'2008, CFSE'6, 2008.
2. **Sylvain Collange, Marc Daumas, David Defour, and Régis Olivès.** *Fonctions élémentaires sur GPU exploitant la localité de valeurs.* In Symposium en Architectures nouvelles de machines, (12 pages), Fribourg, Switzerland, 2008.

#### **Thèses et habilitation sur les thèmes d'EVA-Flo, soutenues pendant le projet**

1. **Francisco José Cháves Alonso.** *Utilisation et certification de l'arithmétique d'intervalles dans un assistant de preuves.* PhD thesis, Université de Lyon - École Normale Supérieure de Lyon, France, September 2007.
2. **Sylvain Chevillard.** *Évaluation efficace de fonctions numériques - Outils et exemples.* PhD thesis, Université de Lyon - École Normale Supérieure de Lyon, France, July 2009.
3. **Sylvain Collange.** *Enjeux de conception des architectures GPGPU : unités arithmétiques spécialisées et exploitation de la régularité.* PhD thesis, Université de Perpignan, November 2010.
4. **Florent de Dinechin.** *Matériel et logiciel pour l'évaluation de fonctions numériques. Précision, performance et validation.* Habilitation à diriger des recherches, Université Claude Bernard - Lyon 1, June 2007.
5. **Jérémie Detrey.** *Arithmétiques réelles sur FPGA : virgule fixe, virgule flottante et système logarithmique.* PhD thesis, École Normale Supérieure de Lyon, France, January 2007.
6. **Christoph Quirin Lauter.** *Arrondi correct de fonctions mathématiques - fonctions univariées et bivariées, certification et automatisation.* PhD thesis, Université de Lyon - École Normale Supérieure de Lyon, France, October 2008.

7. **Nicolas Louvet.** *Algorithmes compensés en arithmétique flottante : précision, validation, performances.* PhD thesis, Université de Perpignan, November 2007.
8. **Hong Diep Nguyen.** *Efficient algorithms for verified scientific computing : numerical linear algebra using interval arithmetic.* PhD thesis, Université de Lyon - École Normale Supérieure de Lyon, France, January 2011.
9. **Guillaume Revy.** *Implementation of binary floating-point arithmetic on embedded integer processors : Polynomial evaluation-based algorithms and certified code generation.* PhD thesis, Université de Lyon - École Normale Supérieure de Lyon, France, December 2009.

La liste complète des publications relevant des thèmes du projet EVA-Flo se trouve en fichier PDF joint.

Aucune action de vulgarisation relevant uniquement du projet EVA-Flo n'est à mentionner, cependant plusieurs membres du projet ont mené des actions de vulgarisation (articles dans « Images des maths » du CNRS, interventions en collèges et lycées...).

### **E.3 LISTE DES ÉLÉMENTS DE VALORISATION**

#### **Logiciels**

- CRlibm : bibliothèque de fonctions mathématiques avec arrondi correct ; à l'origine de la recommandation de la norme IEEE 754-2008 d'inclure les fonctions mathématiques avec arrondi correct ;
- Sollya : recherche d'un bon approximant ;
- Gappa : preuve de qualité numérique, vérifiable par Coq ;
- CGPE : recherche d'un schéma d'évaluation exploitant le parallélisme architectural ;
- FLIP : arithmétique flottante sur certains DSP ;
- FloPoCo : cœurs flottants sur FPGA ;
- Barra : calcul numérique sur GPU ;
- Fluctuat : analyse de la qualité numérique de codes scientifiques (pré-existant au projet) ;
- Tapenade : différentiation automatique (pré-existant au projet).

#### **Organisation et soutien de rencontres et conférences**

- RAIM 2009 (<http://www.ens-lyon.fr/LIP/Arenaire/RAIM09/>) : 3es Rencontres Arithmétique de l'Informatique Mathématique, 26-28 octobre 2009, LIP, ENS de Lyon.

Rencontres francophones regroupant plus de 70 participants et, parmi les orateurs, des membres du projet. Les organisateurs sont des membres lyonnais du projet.

- SCAN 2010 (<http://scan2010.ens-lyon.fr/>) : 14<sup>th</sup> GAMM-IMACS International Symposium on Scientific Computing, Computer Arithmetic and Validated Numerics, 27-30 septembre 2010, ENS de Lyon.

Conférence internationale regroupant 123 participants issus de 20 pays et, parmi les intervenants, des membres du projet. Les organisateurs à la fois pour la partie scientifique et pour la partie logistique sont des membres lyonnais du projet.

Cette organisation avait été mentionnée lors de la soumission du projet EVA-Flo. Comme cette conférence a lieu tous les deux ans et que les organisateurs sont choisis 4 ans à l'avance, il ne nous a pas été possible d'organiser SCAN 2008 comme prévu, mais SCAN 2010.

#### ***Participation à des actions de normalisation***

- Révision de la norme IEEE 754 sur l'arithmétique flottante en 2008.
- Norme IEEE 1788 sur l'arithmétique par intervalles depuis 2008 : création et présidence du groupe de travail.

#### ***Nouveaux projets***

- Sardanes : pour l'amélioration de la qualité numérique de programmes SCADE, financé par la Fondation de Recherche pour l'Aéronautique et l'Espace (2009-2012) ;
- TaMaDi : projet ANR blanc 2010-2013 sur le dilemme du fabricant de tables ;
- HPAC : projet ANR blanc 2011-2014 HPAC sur le calcul algébrique haute performance.

## E.4 BILAN ET SUIVI DES PERSONNELS RECRUTÉS EN CDD (HORS STAGIAIRES)

Ce tableau dresse le bilan du projet en termes de recrutement de personnels non permanents sur CDD ou assimilé. Renseigner une ligne par personne embauchée sur le projet quand l'embauche a été financée partiellement ou en totalité par l'aide de l'ANR et quand la contribution au projet a été d'une durée au moins égale à 3 mois, tous contrats confondus, l'aide de l'ANR pouvant ne représenter qu'une partie de la rémunération de la personne sur la durée de sa participation au projet.

Les stagiaires bénéficiant d'une convention de stage avec un établissement d'enseignement ne doivent pas être mentionnés.

Les données recueillies pourront faire l'objet d'une demande de mise à jour par l'ANR jusqu'à 5 ans après la fin du projet.

Identification				Avant le recrutement sur le projet			Recrutement sur le projet				Après le projet				
Nom et prénom	Sexe H/F	Adresse email (1)	Date des dernières nouvelles	Dernier diplôme obtenu au moment du recrutement	Lieu d'études (France, UE, hors UE)	Expérience prof. Antérieure, y compris post-docs (ans)	Partenaire ayant embauché la personne	Poste dans le projet (2)	Durée missions (mois) (3)	Date de fin de mission sur le projet	Devenir professionnel (4)	Type d'employeur (5)	Type d'emploi (6)	Lien au projet ANR (7)	Valorisation expérience (8)
NOVOCIN Andrew	H	andy@novocin.com	Juin 2011	doctorat	Hors UE (Floride)	1 an post-doc au LIRMM	INRIA Grenoble, LIP, ENS de Lyon	ingénieur	24 mois	Avril 2010	CDD	Enseignement et recherche publique	post-doctorat	Non partenaire (Canada)	oui
THÉVENY Philippe	H	Philippe.Thieveny@laposte.net	Juin 2011	IUP Génie Mathématique et Informatique	France	3 ans comme ingénieur	INRIA Grenoble, LIP, ENS de Lyon	ingénieur	11 mois	Août 2010	Reprise d'études				oui

### Aide pour le remplissage

(1) **Adresse email** : indiquer une adresse email la plus pérenne possible

(2) **Poste dans le projet** : post-doc, doctorant, ingénieur ou niveau ingénieur, technicien, vacataire, autre (préciser)

(3) **Durée missions** : indiquer en mois la durée totale des missions (y compris celles non financées par l'ANR) effectuées sur le projet

(4) **Devenir professionnel** : CDI, CDD, chef d'entreprise, encore sur le projet, post-doc France, post-doc étranger, étudiant, recherche d'emploi, sans nouvelles

(5) **Type d'employeur** : enseignement et recherche publique, EPIC de recherche, grande entreprise, PME/TPE, création d'entreprise, autre public, autre privé, libéral, autre (préciser)

(6) **Type d'emploi** : ingénieur, chercheur, enseignant-chercheur, cadre, technicien, autre (préciser)

(7) **Lien au projet ANR** : préciser si l'employeur est ou non un partenaire du projet

(8) **Valorisation expérience** : préciser si le poste occupé valorise l'expérience acquise pendant le projet.

Les informations personnelles recueillies feront l'objet d'un traitement de données informatisées pour les seuls besoins de l'étude anonymisée sur le devenir professionnel des personnes recrutées sur les projets ANR. Elles ne feront l'objet d'aucune cession et seront conservées par l'ANR pendant une durée maximale de 5 ans après la fin du projet concerné. Conformément à la loi n° 78-17 du 6 janvier 1978 modifiée, relative à l'Informatique, aux Fichiers et aux Libertés, les personnes concernées disposent d'un droit d'accès, de rectification et de suppression des données personnelles les concernant. Les personnes concernées seront informées directement de ce droit lorsque leurs coordonnées sont renseignées. Elles peuvent exercer ce droit en s'adressant l'ANR (<http://www.agence-nationale-recherche.fr/Contact>).