INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Projet ARENAIRE

Arithmétique des Ordinateurs

ENS-Lyon, INRIA Rhône-Alpes

THÈME 2B

Rapport d'Activité

2000

Table des matières

Le projet ARENAIRE est un projet commun au CNRS, à l'École Normale Supérieure de Lyon et à l'INRIA. Il fait partie du Laboratoire de l'Informatique du Parallélisme (LIP, UMR CNRS-ENS Lyon-INRIA 5668) de l'École Normale Supérieure de Lyon. Ce projet est localisé à Lyon dans les locaux de l'ENS Lyon.

1 Composition de l'équipe

Responsable scientifique

Jean-Michel Muller [Directeur de Recherche au CNRS]

Assistante de projet

Anne-Pascale Botonnet [Contractuelle, 20% sur le projet]

Personnel INRIA

Arnaud Tisserand [Chargé de Recherche]

Philippe Langlois [en délégation sur un poste de Chargé de Recherche]

Nathalie Revol [en délégation sur un poste de Chargé de Recherche, dans le projet depuis le 1/10/2000]

Personnel CNRS

Marc Daumas [Chargé de Recherche]

Gilles Villard [Chargé de Recherche, dans le projet depuis le 1/12/2000]

Personnel ENS Lyon

Florent Dupont de Dinechin [Maître de conférences à l'ENS Lyon]

Chercheurs doctorants

Claire Finot-Moreau [Allocataire et Moniteur MENRT]

David Defour [Allocataire MENRT. Dans le projet depuis le 1/10/2000]

Vincent Lefèvre [Allocation couplée. A quitté le projet le 1/10/2000 suite à son recrutement comme CR2 à l'INRIA Lorraine]

2 Présentation et objectifs généraux

Mots clés : arithmétique des ordinateurs, virgule flottante, fiabilité numérique, fonctions élémentaires, opérateurs asynchrones, circuits intégrés numériques, FPGA, calcul formel.

Résumé : *L'objectif du projet* ARÉNAIRE *est de contribuer à l'élaboration et à la consolidation des connaissances dans le domaine de l'arithmétique des ordinateurs. Nos travaux se situent sur deux axes privilégiés : d'une part l'utilisation avertie des normes existantes pour construire des algorithmes fiables et les prouver ; d'autre part la mise au point, plus en amont, des unités arithmétiques du futur. Fiabilité, précision, rapidité et (à plus long terme) faible consommation sont les principaux objectifs du projet.*

Le but scientifique du projet est de participer à l'amélioration de l'arithmétique disponible sur les microprocesseurs et les calculateurs dédiés. Cette amélioration peut porter aussi bien sur la rapidité des calculs que sur leur fiabilité et leur précision, ainsi que sur des paramètres plus technologiques tels que la consommation d'énergie dans les circuits intégrés.

L'arithmétique des ordinateurs a connu un changement majeur en 1985 avec l'apparition de la norme IEEE-754, qui spécifie les formats de représentation des nombres et les opérations arithmétiques en virgule flottante. Cette norme s'est rapidement imposée. Une des principales exigences de la norme est liée aux modes d'arrondi. La somme, le produit, le quotient de deux nombres qui s'écrivent exactement en virgule flottante (on dira des « nombres machine » pour simplifier) ne sont en général pas des nombres machine. Il faut donc les *arrondir*. La norme IEEE-754 demande que l'utilisateur puisse choisir entre 4 modes d'arrondi possibles : au plus près — c'est le mode par défaut — , vers le haut, vers le bas ou vers zéro. Si a et b sont des nombres machine, le résultat d'une des quatre opérations arithmétiques $a \star b$ doit être le nombre machine qui serait obtenu si on avait d'abord fait le calcul avec une *précision infinie* avant d'arrondir avec le mode d'arrondi demandé par l'utilisateur. Cette exigence est appelée « arrondi exact ».

La plupart des micro-ordinateurs et des stations de travail actuels suivent la norme IEEE et les conséquences de cette adhésion sont importantes :

- on peut développer des *algorithmes* et des *preuves* de correction qui utilisent la spécification des unités de calcul comme cela a été fait par William Kahan et ses élèves à l'*University of California* de Berkeley [Kah99] ;

- en jouant sur les différents modes d'arrondi, il est possible pour l'utilisateur d'obtenir un minorant ou un majorant certifié du résultat et de faire ainsi de l'*arithmétique d'intervalles*.

[Kah99] W. KAHAN, « Lecture Notes on the Status of the IEEE-754 Floating-Point Standard », 1999, `http://www.cs.berkeley.edu/~~wkahan/`.

L'exemple suivant illustre les avantages que l'on peut retirer de cette démarche. On prouve [CHGM99] que (sauf dépassements de capacité), si x est exactement représentable en flottant et si z est une approximation au poids du dernier bit près de $1/x$, alors la suite d'opérations

$$\begin{aligned} \epsilon &= \circ_n(1 - xz) \\ z' &= \circ_n(z + \epsilon z) \end{aligned}$$

où $\circ_n(a + bc)$ signifie que le calcul $a + bc$ est fait en arrondi au plus près, donne une valeur z' égale à l'arrondi au plus près de $1/x$. Ainsi, en utilisant l'arrondi correct de la multiplication-accumulation, on peut fournir des algorithmes de division fournissant un arrondi correct sans utiliser d'arithmétique à plus grande précision pour les calculs intermédiaires. C'est cette méthode qui sera utilisée pour effectuer des divisions sur l'architecture IA64, développée conjointement par Intel et HP.

La plupart des travaux actuels en arithmétique des ordinateurs se situent autour des deux thèmes suivants :

- **Thème 1** : Les opérateurs arithmétiques étant donnés, les utiliser « au mieux ». Par exemple, les « axiomes » qui constituent les spécifications de la norme IEEE (ou d'autres spécifications) étant donnés, il faut construire des algorithmes et les preuves de correction de ces algorithmes qui utilisent au mieux chaque opportunité de performance. Nous mettons au point une arithmétique en précision multiple qui utilise, au lieu d'entiers, des flottants IEEE comme « briques de base ». Nous démarrons une activité d'arithmétique en calcul formel. Dans le cadre de l'ARC INRIA « AOC » (Arithmétique des Ordinateurs Certifiée), nous travaillons avec des membres des projets LEMME (Sophia) et POLKA (Lorraine) à la preuve de propriétés et d'algorithmes en virgule flottante. Dans le cadre d'un contrat entre l'INRIA et ST Microelectronics, nous avons proposé des algorithmes de division par une constante utilisant les opérateurs disponibles sur un circuit DSP construit par ST.

- **Thème 2** : Concevoir et/ou valider des opérateurs, ou de manière peut-être plus générale, concevoir d'autres « briques de base » que celles de l'arithmétique IEEE. La norme IEEE ne spécifie que les quatre opérations arithmétiques et la racine carrée, et nous travaillons à une implantation rigoureuse et intelligente d'autres fonctions en virgule flottante. Nous travaillons également sur d'autres systèmes de numération. Il n'est pas utopique d'étudier ces systèmes : si pour des besoins « généralistes » comme pour un microprocesseur, l'arithmétique virgule flottante semble incontournable (même si on peut certainement en améliorer les implantations existantes), sur des systèmes dédiés à des applications particulières, d'autres modes de représentation des nombres peuvent s'avérer mieux adaptés. Un exemple classique est celui des systèmes de numération *redondants* (carry-save, borrow-save, etc.) qui sont utilisés à l'intérieur de nombreux multiplieurs et diviseurs. Cette utilisation est transparente : en entrée comme en sortie de ces opérateurs, les nombres sont représentés dans un système usuel.

[CHGM99] M. Cornea-Hasegan, R. Golliver, P. Markstein, « Correctness Proofs Outline for Newton-Raphson Based Floating-Point Divide and Square-Root Algorithms », *in : Proceedings of the 14th IEEE Symposium on Computer Arithmetic*, IEEE Computer Society Press, Los Alamitos, CA, Adelaide (Australie), Avril 1999.

La recherche sur les algorithmes et les architectures pour la multiplication, pour la division, et pour le calcul des fonctions élémentaires est toujours très active[1]. Les membres du projet ont acquis une solide réputation dans ce domaine et ont l'intention de poursuivre leurs travaux dans cette voie.

La maîtrise des erreurs d'arrondi dans les calculs (et de manière plus générale, la fiabilité numérique des systèmes) est un sujet de plus en plus important. On effectue des calculs considérablement plus volumineux que dans les années 70, alors que les formats de représentation des nombres (et par conséquent la précision de chaque opération prise individuellement) n'ont presque pas changé. Une conséquence est que les problèmes de précision se posent de plus en plus. Dans de nombreux domaines d'application, l'imprécision de l'arithmétique flottante peut avoir des conséquences tragiques. La maîtrise de la précision est un thème important. Notre équipe travaille déjà activement sur ce thème : nous avons participé jusqu'en 1999 à l'ARC INRIA Fiable. Nous avons également collaboré en 1998-99 avec la société Aérospatiale dans ce domaine. Nous nous intéressons tout particulièrement à la précision du calcul des fonctions élémentaires. Notre avancée récente sur le problème du « Dilemme du Fabricant de Tables » se situe dans ce domaine.

L'amélioration automatique de la précision des résultats complète la maîtrise de la fiabilité numérique. Les utilisateurs qui rencontrent des problèmes de précision numérique n'ont souvent ni le temps ni le métier pour répondre à ces problèmes difficiles. Ce constat incite à proposer des approches automatiques qui permettent de contrôler et d'améliorer la qualité numérique d'un programme vu comme une « boîte noire ».

3 Fondements scientifiques

3.0.1 Algorithmes orientés matériel de calcul

On s'intéresse depuis la création du projet à la mise au point d'algorithmes arithmétiques « orientés matériel », c'est-à-dire destinés à une implantation matérielle des opérations arithmétiques et des fonctions usuelles. Suivant l'application visée, les solutions retenues peuvent être très différentes selon qu'on privilégie la précision (processeurs généralistes et de calcul), la rapidité (par exemple pour des processeurs de traitement du signal), la consommation (systèmes embarqués ou portables) ou la taille des circuits (idem). Nous sommes une des très rares équipes universitaires à travailler en France sur ce sujet, ce qui nous conduit à avoir de nombreuses collaborations étrangères (en ce moment, le Centre Suisse d'Electronique et de Microtechniques de Neuchâtel, ainsi que les Universités de Californie d'Irvine et Los Angeles), et des collaborations industrielles (ST Microelectronics). Nous avons en particulier travaillé ces dernières années sur l'implantation de la division, de la racine carrée et des fonctions élémentaires.

1. Juste pour donner un exemple : les algorithmes des diviseurs des processeurs HP PA-7100, HP PA-8000, Amd 29050, UltraSparc, Ibm RS/6000 sont *tous* différents. Ceci montre à l'évidence que les solutions sont encore très loin d'être figées dans ce domaine.

3.0.2 Arithmétique virgule flottante et preuves formelles

Un opérateur arithmétique manipulant des nombres à virgule flottante est plus complexe que le même opérateur restreint aux seuls nombres entiers. Il faut en effet gérer correctement les divers modes d'arrondi, manipuler à la fois les mantisses et les exposants des opérandes, traiter les divers cas d'exception (infinis, nombres « dénormalisés », etc.). Sur de nombreux microprocesseurs actuels, l'unité de calcul à virgule flottante est plus puissante que l'unité de calcul entier (car les constructeurs y mettent le prix). Il devient important de bien en maîtriser le fonctionnement. Une bonne connaissance des spécificités des standards actuels peut permettre d'obtenir des programmes très efficaces.

Des «plantages» retentissants ont montré que l'arithmétique flottante est parfois difficile à manier. Peu d'outils permettent de faire des preuves rigoureuses sur ce type de données. Pourtant, le fait que, grâce à la norme IEEE-754, les opérations arithmétiques soient complètement spécifiées devrait permettre de construire des preuves d'algorithmes et de propriétés.

Pour formaliser nos preuves, et dans le cadre de notre collaboration avec les projets LEMME et POLKA (ARC AOC), nous utilisons l'assistant de preuve Coq. Les sytèmes tels Coq permettent de définir de nouveaux objets et de dériver des conséquences formelles de ces définitions. Le langage Coq est basé sur une logique d'un ordre supérieur. Avec une logique aussi puissante, on peut établir des propriétés sous une forme très générale. Par exemple, nous avons utilisé des quantificateurs universels pour établir des propriétés indépendamment de la base d'écriture des nombres flottants ou pour un mode d'arrondi arbitraire. Les preuves sont construites de façon interactive en utilisant des tactiques de haut niveau. A la fin de chaque preuve, Coq construit un objet interne qui contient tous les détails des dérivations et garantit que le théorème est valide.

3.0.3 Correction linéaire des erreurs d'arrondi et arithmétique à précision garantie

Les erreurs d'arrondi introduites par chaque opération arithmétique modifient le comportement numérique de certains algorithmes. Elles altèrent la précision du résultat final allant même jusqu'à parfois l'invalider. Elles peuvent dégrader la stabilité de certains algorithmes pourtant démontrés stables en arithmétique exacte. Elles peuvent également faire prendre un « mauvais branchement » lors de l'éxecution d'un test.

L'erreur globale sur le résultat final peut être approchée en linéarisant l'influence des erreurs d'arrondi élémentaires qui la génère. Cette technique de linéarisation avait jusqu'à présent été utilisée pour majorer cette erreur globale. La correction linéaire automatique étend cette technique en *calculant* cette linéarisation par différentiation automatique et évaluation des erreurs d'arrondi élémentaires. On dispose ainsi d'un terme correctif qui, ajouté au résultat final, en améliore la précision. Cette amélioration n'est pas équivalente à celle qui résulterait d'un passage en précision supérieure. Cette correction est appliquée automatiquement au résultat final de certains algorithmes mais aussi à certaines variables intermédiaires. Dans ce dernier cas, la stabilité de l'algorithme peut être améliorée.

Un autre outil que nous commençons à étudier (depuis l'arrivée de Nathalie Revol dans le projet) est le «couplage» de l'arithmétique d'intervalles et de la multi-précision. Utiliser l'arithmétique par intervalles permet de garantir l'encadrement des résultats d'un calcul. Cependant cette arithmétique fournit des encadrements parfois trop larges. Une solution pour pallier ce problème consiste à effectuer les calculs en précision plus grande que la précision machine, autrement dit à combiner arithmétique par intervalles et arithmétique multi-précision.

Le remplacement de l'arithmétique flottante par une arithmétique multi-précision au cœur de l'arithmétique par intervalles n'est pas chose évidente. En effet il faut disposer d'une arithmétique multi-précision qui offre la possibilité de diriger les arrondis afin de retourner un intervalle résultat qui contienne à coup sûr le résultat scalaire cherché. Il faut également remplacer la manipulation de scalaires par celle d'intervalles dans l'arithmétique multi-précision et gérer correctement les cas particuliers, comme les divisions par un intervalle contenant zéro par exemple.

Une nouvelle approche arithmétique doit être testée et validée à l'aide d'une application réaliste. Actuellement le sujet d'étude de Nathalie Revol est le problème de l'optimisation globale d'une fonction continue et sa résolution par l'algorithme de Hansen, qui utilise intensivement les propriétés de l'arithmétique par intervalles. Cette application pourra donc servir de programme de test pour les réalisations à venir.

Bien évidemment, cet aspect de notre travail se fera en liaison étroite avec le projet POLKA de l'UR Lorraine, qui s'intéresse lui aussi de près à ce problème (en particulier à l'occasion du développement de la bibliothèque mpfr).

3.0.4 Le dilemme du fabricant de tables

Dans la norme IEEE 754, l'exigence d'« arrondi exact» n'apparaît que pour les 4 opérations arithmétiques et la racine carrée. Il n'y a aucune exigence de ce type pour les fonctions élémentaires, car on a longtemps cru qu'une telle exigence était impossible. Lorsque l'on cherche à évaluer une fonction élémentaire (sinus, cosinus, logarithme, exponentielle, etc.), on calcule en fait une approximation intermédiaire de la valeur exacte sur une arithmétique à précision finie mais supérieure à la précision qui correspond au format dans lequel le résultat final sera retourné, appelé « format cible ». Ensuite, on arrondit l'approximation, suivant le mode d'arrondi désiré, vers le format cible. Tout le problème est de savoir si la précision du calcul intermédiaire suffit pour que cet arrondi de l'approximation coïncide avec l'arrondi du résultat exact. Résoudre ce problème permettrait de fournir des environnements où toutes les primitives numériques (i.e. les opérations et les fonctions élémentaires) seraient complètement spécifiées. C'est un des objectifs majeurs de notre projet, et nous avons réalisé une « percée » cette année, en obtenant (pour de nombreuses fonctions), les pires cas pour de nombreuses fonctions en double précision (un « pire cas» pour la fonction f est un nombre x tel que $f(x)$ est le plus proche du milieu de deux nombres machine consécutifs (en arrondi au plus près) ou d'un nombre machine (dans les autres modes d'arrondi). Donnons un exemple. Supposons des mantisses de 24 bits (c'est le cas de la simple précision de la norme IEEE). Considérons le nombre x qui s'écrit 1.11001100111000110011001. Son exponentielle s'écrit en base 2

$$\overbrace{110.000011010011110100110}^{24\text{ bits}}\,0\,\overbrace{111111111111111111111111}^{24\text{ uns}}\,01100100001\ldots$$

ce qui est extrêmement proche du milieu de deux nombres machine consécutifs. Si l'on désire calculer l'exponentielle de x en arrondi au plus près, il faudra le faire en calculant tout d'abord une approximation très précise de cette exponentielle (sur au moins 50 bits), afin de savoir si elle est au-dessous ou au-dessus du milieu des nombres machine

$$A = 110.000011010011110100110$$

et

$$B = 110.000011010011110100111.$$

Si elle est en dessous (ce qui est ici le cas), le résultat fourni devra être A, et dans le cas contraire ce sera B.

3.0.5 Calcul formel : algèbre linéaire et arithmétique

Ce thème doit démarrer avec l'arrivé de Gilles Villard dans le projet. En calcul formel comme en calcul numérique la bonne qualité des outils d'algèbre linéaire est souvent un point clé pour la résolution des problèmes. Le progrès considérable des algorithmes formels (rapides et rigoureusement analysés) ces quinze dernières années fait que tout le domaine devient abordable en pratique. Cependant, beaucoup reste à faire pour la mise à disposition de bibliothèques de programmes performants, capables de compléter les systèmes généralistes qui dominent. La conception de bibliothèques spécialisées et particulièrement efficaces est un mouvement largement suivi actuellement dans la communauté. Un des points cruciaux en pratique est le développement et l'utilisation d'arithmétiques efficaces.

Nos recherches seront menées dans trois directions, en *algorithmique*, pour le développement d'une *bibliothèque générique* de calcul et sur ces derniers aspects d'*arithmétique*.

- Le travail algorithmique pour de nouveaux algorithmes concerne les matrices structurées et creuses, les matrices qui dépendent de paramètres.

- Pour diffuser les algorithmes, le moyen que l'on se donne est l'écriture d'une bibliothèque générique, LinBox, dans un cadre de collaborations internationales Cette bibliothèque est basée sur le *branchement à la demande* de *composants* qui peuvent être d'autres bibliothèques ou réaliser des appels à des systèmes externes comme Maple ou Matlab.

- La généricité de LinBox se veut notamment relativement à l'arithmétique utilisée. Les types algébriques des nombres manipulés dépassent largement les nombres à précision fixée. Nous proposons – par le biais de l'algèbre linéaire – l'interfaçage et le test d'arithmétiques (précision fixée arbitraire, corps finis, précision infinie). Ces études se feront de manière complémentaire avec le projet Polka (INRIA Lorraine).

4 Résultats nouveaux

4.1 Algorithmes de division et algorithmes orientés matériel

Participants : Jean-Michel Muller, Arnaud Tisserand.

Mots clés : division, méthode de Newton, méthode de Goldschmidt, division par une constante, racine carrée.

Résumé : *Nous avons étudié des algorithmes de division de style « méthode de Newton » ainsi que des algorithmes de division par une constante.*

L'an dernier, Jean-Michel Muller avait collaboré avec Laurent Imbert (Laboratoire d'Informatique de Marseille), Milos Ercegovac (University of California at Los Angeles) et David Matula (Southern Methodist University de Dallas) sur l'accélération de l'algorithme itératif de division et racine carrée de Goldschmidt sur un multiplieur pipe-line (voir rapport d'activité 1999). Une version revue et augmentée de notre étude a été publiée dans la revue IEEE Transactions on Computers [9].

Dans le cadre d'un contrat entre l'INRIA et ST Microelectronics, Arnaud Tisserand et Jean-Michel Muller ont étudié des algorithmes de division par une constante (voir Section 5.1). Nous envisageons de publier une partie des résultats obtenus lorsque les caractéristiques du circuit de ST auquel ces algorithmes sont destinés seront rendues publiques (courant 2001).

En collaboration avec Milos Ercegovac (Univ. of California at Los Angeles) et T. Lang (Univ. of California at Irvine), Arnaud Tisserand et Jean-Michel Muller ont proposé et publié [10] une architecture permettant d'effectuer des divisions et des racines carrées et de calculer certaines fonctions en n'utilisant que des lectures dans des tables de taille raisonnable et des multiplications « rectangulaires » (c'est-à-dire dont l'une des opérandes est de petite taille, ce qui autorise des multiplieurs à la fois petits et rapides).

En collaboration avec Fabien Rico (LIRMM, Montpellier) et Laurent Imbert (LIM, Marseille), Jean-Michel Muller avait proposé l'an dernier un algorithme de calcul de l'exponentielle et du logarithme complexe (et donc de nombreuses fonctions élémentaires) utilisant la base 10, et donc adapté au calcul sur calculatrices de poche (voir le rapport d'activité 1999 du projet). Nous avons publié une version revue et augmentée de notre travail [11].

4.2 Calcul des fonctions élémentaires

Participants : Marc Daumas, Claire Finot-Moreau, Vincent Lefèvre, Jean-Michel Muller.

Mots clés : virgule flottante, erreurs d'arrondi, dilemme du fabricant de tables, fonctions élémentaires.

Résumé : *Les algorithmes et des programmes permettant de construire les pires cas pour le dilemme du fabricant de tables que nous avions conçus nous ont permis d'obtenir ces pires cas pour la « double précision ». Ces pires cas vont nous permettre de construire une bibliothèque calculant les fonctions élémentaires avec « arrondi exact ». Nous avons proposé de nouveaux programmes pour le calcul de certaines fonctions élémentaires avec une précision de l'ordre de la centaine de bits.*

Vincent Lefèvre avait mis au point un algorithme et une « batterie » de programmes permettant de construire les pires cas, pour le format virgule flottante « double précision » du dilemme du fabricant de tables. Cet algorithme a été présenté dans sa thèse soutenue cette année [5, 12], et nous avons travaillé à son amélioration dans plusieurs cas particuliers. Nous avons publié les résultats obtenus dans un rapport de recherches [35], soumis à la conférence IEEE Arith-15. Nous avons présenté les conséquences de notre étude à l'occasion d'une communication à la *34th Asilomar Conference on Circuits and Systems* [29] (où J.M. Muller était conférencier invité).

Nous rendons publique une liste restreinte de pire cas à l'URL

`http://www.ens-lyon.fr/~jmmuller/TMD.html`

Cette liste ne contient pas tous les pires cas trouvés, car nous désirons d'une part prendre un peu d'avance dans l'utilisation de nos pires cas pour la conception de fonctions, et d'autre part valoriser les résultats trouvés en ne les diffusant qu'à un « club de partenaires » de notre projet.

Par exemple, la table suivante donne les pires cas trouvés pour le logarithme népérien.

Intervalle	pire cas (en binaire)
$[2^{-1074}, 1/8]$	$\log(1.1001010001110110111000110000010011001101011111000111 \times 2^{-384})$ $= -100001001.10110110000011001010111101000111101100110101 \quad 1 \quad 0^{60}1010...$
	$\log(1.1110101001110001110110000101110011101110000000100000 \times 2^{-509})$ $= -101100000.00101001011010100110011010110100001011111111 \quad 1 \quad 1^{60}0000...$
	$\log(1.0010011011101001110001001101001100100111100101100000 \times 2^{-232})$ $= -10100000.101010110010110000100101111001101000010000100 \quad 0 \quad 0^{60}1001...$
$[1/8, 1)$	$\log(1.11001000111111010100001101010100000000010010000110011 \times 2^{-3})$ $= -1.01111111111111010100111111111000101010010001111011111 \quad 1 \quad 0^{54}1101...$
	$\log(0.110111010110111101101001100001100101111110001110001 00)$ $= -1.00101001000011101010000001001111000110110010001111000 \quad 1 \quad 1^{54}0110... \times 2^{-3}$
$(1, 8]$	$\log(110.010001101010001101111111110101010000000111111110111)$ $= 1.1101011000110011011010101000100000000111011110101001 \quad 1 \quad 1^{54}0000...$
$[8, 2^{1024}]$	$\log(1.0110001010101000100001100001001101100010100110110110 \times 2^{678})$ $= 111010110.01000111100111101011101001111100100101110001 \quad 0 \quad 0^{64}1110...$

Tomas Lang (University of California at Irvine) et Jean-Michel Muller ont regardé s'il était possible d'obtenir directement des pires cas pour le dilemme du fabricant de tables lorsque la fonction calculée est algébrique. Ils ont obtenu une borne permettant de savoir avec quelle précision une approximation de la fonction doit être évaluée pour que l'on soit certain qu'arrondir l'approximation est équivalent à arrondir la fonction. Ils ont montré que pour certaines fonctions courantes, la borne était atteinte (ce qui n'est en général pas le cas). Les résultats sont en cours de publication [32].

Pour certains problèmes de calcul, la « double précision » de la norme IEEE-754 est insuffisante même si une précision multiple (i.e., plusieurs centaines de bits) n'est pas nécessaire et serait souvent trop lente. Les codes correspondants ne sont pas instables, mais les puissances de calcul disponibles permettent d'envisager de travailler avec des échelles très différentes et dans ce cas, une précision quadruple s'avère nécessaire.

Une bibliothèque de calcul est actuellement disponible et distribuée sur le réseau par des laboratoires américains dépendant du *Department of Energy*. Cette bibliothèque utilise le format double précision comme « brique de base » pour atteindre une précision proche de la quadruple précision. Ces développements permettent ainsi de fournir aujourd'hui la quadruple précision, et quand cette dernière sera disponible au niveau matériel, ils se transposeront pour atteindre sans travail supplémentaire l'octuple précision. Nous travaillons avec David Bailey le responsable du développement de cette bibliothèque dans le cadre d'un financement France-Berkeley que nous avons obtenu l'an dernier. Suite à une étude systématique des choix possibles, nous avons proposé de nouvelles routines pour certaines fonctions élémentaires (sinus, cosinus, exponentielle) [18, 17].

L'étude systématique a été menée à l'aide du logiciel Maple et les fichiers sources seront bientôt disponibles sur Internet pour permettre à d'autres équipes d'explorer de nouvelles possibilités. Dans cette optique nous avons mené des travaux avec le LIRMM et le LIM pour étudier la possibilité d'utiliser l'algorithme CORDIC ou une de ses variantes [23].

Ces travaux pourront être utilisés dans le cadre de l'implantation de fonctions élémentaires avec arrondi correct. Les développements sous Maple pourront éventuellement être transcrits sous un environnement de preuve tel COQ.

Une phase critique (du point de vue de la précision) du calcul des fonctions élémentaires est la *réduction d'argument*, qui consiste à se ramener à l'intervalle dans lequel l'approximation choisie converge. Vincent Lefèvre et Jean-Michel Muller ont proposé un algorithme effectuant cette réduction d'argument « au vol » [34, 28].

4.3 Arithmétique virgule flottante : spécification et algorithmes de manipulation d'expansions

Participants : Marc Daumas, Claire Finot-Moreau.

Mots clés : virgule flottante, preuve d'algorithmes, erreurs d'arrondi, arithmétique exacte, précision multiple, expansion, division, géométrie algorithmique.

Résumé : *Nous travaillons à la preuve formelle de propriétés et d'algorithmes en arithmétique virgule flottante. Ceci est rendu possible grâce aux spécifications fournies par la norme* IEEE *754. Une expansion de nombres flottants est un nombre en précision multiple représenté par la somme d'un petit nombre de flottants. Manipuler des expansions permet parfois de rendre plus précise une portion critique d'un programme.*

Grâce à notre collaboration dans le cadre de l'action de recherches coopératives INRIA AOC, et en particulier au travail de Laurent Thery (INRIA Sophia), nous avons commencé à transférer certaines de nos preuves et spécifications sous l'outil Coq. Cette approche est particulièrement utile de plusieurs points de vue :

- Coq est un formalisme moins permissif que les formalismes utilisés par le passé pour des preuves de ce type.
- Cette preuve part d'une spécification naturelle et suit un chemin de preuve passant par des résultats intermédiaires connus et documentés dans la littérature. Cette approche nous garantit qu'un lecteur extérieur à notre développement pourra en extraire du savoir sans devoir d'abord ingurgiter tout notre formalisme.

– Notre travail sur les expansions nous a permis d'isoler des résultats intéressants et de difficulté «raisonnable». Nous avons déjà un ensemble significatif de résultats. Un exemple simple est le suivant: soient deux nombres flottants x et y tels que $\frac{y}{2} \leq x \leq 2y$, le rationnel $x-y$ est exactement représentable (il est donc calculé exactement sur toute arithmétique avec arrondi correct). Ce fait est vrai indépendamment de la base. Par contraposée, si $x+y \neq \circ(x+y)$, alors $|\circ(x+y)| \geq \max(|x|,|y|)/2$. La fonction $\circ$ est l'arrondi au plus près.

  L'état actuel de nos preuves peut être consulté à l'adresse

  `http://www-sop.inria.fr/lemme/AOC/coq/`

  Une prochaine extension de ces travaux est l'écriture d'une preuve validée des limites de la correction flottante pour la méthode CENA [25].

  Parallèlement à ce travail sur les preuves formelles, nous continuons à améliorer les algorithmes sur les « expansions» de nombres flottants. Une expansion de nombres flottants est un nombre en précision multiple représenté par la somme d'un petit nombre de flottants. Pour manipuler ces expansions, on utilise l'unité de calcul en virgule flottante du processeur pour les calculs internes à la place de l'unité de calcul entier.

  Nos résultats algorithmiques ne sont pas encore publiés, mais certains programmes commencent à être diffusés. La bibliothèque actuelle a été modifiée pour qu'elle ne travaille plus en précision arbitraire (calculs exacts aussi longs que nécessaires) mais en précision fixée (l'utilisateur indique *a priori* la précision de chaque opération). Sylvie Boldo, élève de l'ENS Lyon, a effectué un stage à Berkeley dans l'équipe de Schewchuck, dans le cadre du fonds France-Berkeley. Elle a développé des algorithmes d'addition en octuple précision. Ces algorithmes ont été intégrés dans la bibliothèque octuple précision développée et distribuée sous la direction de David Bailey.

4.4 La méthode CENA

Participant : Philippe Langlois.

Mots clés : Correction linéaire des erreurs d'arrondi, virgule flottante.

Résumé : *La méthode CENA automatise la correction linéaire des erreurs d'arrondi qui permet d'améliorer la précision du résultat et la stabilité de certains algorithmes numériques. Les résultats obtenus cette année concernent sa validation théorique, son interfaçage avec une autre méthode automatique et son utilisation de la différentiation automatique.*

Les résultats obtenus l'an passé concernaient l'amélioration de l'efficacité et du spectre d'application de la méthode de correction linéaire CENA [36]. Cette année 2000 a d'abord permis de valider théoriquement certaines limitations du principe de correction linéaire [26]. Les résultats obtenus vont être validés formellement avec l'outil COQ dans le cadre de l'ARC AOC.

Nous avons aussi étudié les contraintes engendrées par la précision courante et l'approximation des erreurs élémentaires [25]. Ces travaux ont permis d'identifier clairement les différences entre la correction automatique et le passage en précision supérieure [33]. Durant son stage de première année d'IUP, Yann Mokwinski a vérifié expérimentalement ces résultats dans le cas de la détermination de racines polynomiales mal conditionnées, ainsi que l'intérêt des critères d'arrêt en *backward error*.

Nous avons ensuite montré l'intérêt complémentaire de la méthode stochastique CESTAC (J.M. Chesneaux et J. Vignes, LIP6) et de la méthode CENA. Grace à l'approche stochastique, nous localisons dynamiquement les parties de code responsables de l'instabilité numérique que nous corrigeons ensuite automatiquement en appliquant la méthode CENA [27].

L'automatisation de la correction linéaire s'appuie sur la différentiation automatique (des parties) du code à corriger. L'effet des erreurs d'arrondi sur ce procédé est étudié dans [15]. En collaboration avec Thierry Braconnier (University of Iowa), nous proposons une stratégie de parallélisation adaptée à la correction automatique [16] et qui est actuellement en cours d'évaluation.

4.5 Arithmétique et FPGAs, Conception de circuits arithmétiques

Participants : Florent de Dinechin, Vincent Lefèvre, Arnaud Tisserand.

Mots clés : FPGAs, circuits reconfigurables, conception de circuits, tables.

Résumé : *Les circuits intégrés reconfigurables (FPGAs) permettent la réalisation d'unités de calcul spécialisées à faible coût. On s'intéresse à l'implantation d'opérateurs arithmétiques sur de tels circuits. On s'est en particulier cette année intéressé à l'implantation de fonctions à l'aide de méthodes utilisant des tables. Nous entamons le développement d'un système de conception de circuits arithmétiques.*

Arnaud Tisserand et Florent de Dinechin se sont intéressés à l'implantation sur FPGAs de méthodes d'évaluation de fonctions n'utilisant que des lectures de tables et un tout petit nombre d'opérations arithmétiques. Un premier travail comparant les performances des méthodes existantes lorsqu'elles sont implantées sur des FPGAs a été présenté à la conférence SPIE 2000 à San Diego [22]. Une étude généralisant les méthodes «multipartites» a été soumise à la conférence IEEE ARITH 15.

Florent de Dinechin et Vincent Lefèvre ont étudié l'implantation sur FPGAs d'algorithmes de multiplication par une constante [21], qui avaient été étudiés de manière générale par Vincent Lefèvre l'an dernier. Ils ont montré qu'une utilisation dans les FPGA de techniques anciennes de recodage en chiffres signés conduisait à un multiplieur par une constante plus petit que le multiplieur habituellement utilisé, dit multiplieur KCM.

Florent de Dinechin a terminé et publié [8] une étude sur le prix du routage dans les FPGAs qu'il avait commencée en 1999. Il travaille actuellement avec Jean-Claude Bajard (LIRMM, Montpellier) à exprimer des algorithmes utilisés en cryptographie de manière à remplacer le plus possible de multiplications par des multiplications par des constantes. Cela permet d'implanter ces algorithmes dans les FPGA de manière plus compacte et plus rapide qu'en utilisant des multiplications normales. Grâce à la reconfigurabilité, la constante peut être changée avec les clés de l'algorithme. Dans les systèmes de représentation des nombres sous forme de résidus (déjà utilisés en cryptographie), la difficulté est de jouer sur les choix des résidus de manière à conduire à de petits multiplieurs.

Arnaud Tisserand commence le développement d'un système de conception de circuits arithmétiques. Ce projet devrait permettre d'insérer les contraintes spécifiques des opérateurs arithmétiques (représentation des nombres, algorithmes de calcul particuliers) à haut niveau pour concevoir des circuits FPGA dans un premier temps et ASIC dans un deuxième temps optimisés. Les cibles sont à la fois les hautes performances et la basse consommation. Il s'agit d'un projet de longue haleine, qui s'étalera sur plusieurs années.

5 Contrats industriels (nationaux, européens et internationaux)

5.1 ST Microelectronics

Participants : Marc Daumas (10%), Jean-Michel Muller (45%), Arnaud Tisserand (45%).

Mots clés : algorithmes de division, arithmétique asynchrone..

La compagnie ST Microelectronics (division compilation) nous a confié une étude confidentielle portant sur la mise au point d'algorithmes de division par une constante spécialement adaptés à l'architecture d'un de leurs prochains processeurs DSP (contrat entre l'INRIA et ST Microelectronics, 60 KF). Nous envisageons de continuer à travailler avec eux.

De plus, nous collaborons avec la dicvision *Advanced Designs and Tools* de ST sur la conception d'opérateurs arithmétiques asynchrones. Nous travaillons actuellement sur la conception d'additionneurs et de multiplieurs totalement asychrones avec comme cible la basse consommation. Les opérateurs réalisés seront intégrés dans la version asychrone du processeur ST20 en début d'année 2001. Une publication commune des résultats est prévue pour le début de l'année 2001.

5.2 CSEM

Participant : Arnaud Tisserand.

Dans le cadre d'un contrat entre l'INRIA et le Centre Suisse d'Electronique et de Microtechnique (CSEM, Neuchâtel, Suisse), nous avons effectué une étude proposant des opérateurs arithmétiques optimisés pour le traitement d'image et la basse consommation. Dans le cadre de ce contrat, nous avons obtenu du CSEM une carte FPGA avec imageur (coût : entre 20 et 25 kF). La version ASIC de notre travail a été intégrée dans un circuit qui est actuellement en cours de fabrication (retour prévu pour la fin d'année 2000).

5.3 Xilinx

Participants : Florent de Dinechin, Arnaud Tisserand.

La compagnie Xilinx (qui est leader dans le domaine de la conception de FPGA) à donné à l'ENS Lyon, pour la réalisation des travaux de recherche du projet Arénaire, un circuit FPGA d'une valeur d'environ 15kF. En contrepartie, notre projet doit présenter sur une page web les réalisations effectuées sur ce circuit.

6 Actions régionales, nationales et internationales

6.1 ACI Jeunes Chercheurs

Participants : Florent de Dinechin, Arnaud Tisserand.

Florent de Dinechin et Arnaud Tisserand ont rédigé une demande de financement « ACI jeunes chercheurs» auprès du MENRT, sur le thème : arithmétique sur FPGA. Cette demande a été acceptée en septembre 2000. Le financement de 200 kF servira essentiellement à l'achat d'une carte FPGA très performante et de logiciels de CAO.

6.2 Collaboration avec l'UC Berkeley

Participants : Marc Daumas, Claire Finot, Jean-Michel Muller.

Mots clés : multiprécision, virgule flottante, manipulation d'expansions.

Résumé : *Nous avons répondu à l'appel d'offres France-Berkeley 1999 et notre proposition a été acceptée. Nous collaborons dans ce cadre avec J. Shewchuk et D. Bailey.*

Nous avons développé, en utilisant la notion d'expansion flottante, un ensemble de programmes efficaces pour le calcul en multiprécision. Ces travaux ont été appliqués pour le développement d'une bibliothèque «octuple-double» dans l'action du Fonds de collaboration avec l'University of California at Berkeley (`http://www.ias.berkeley.edu:80/cwes/fbf/`) entre Marc Daumas et Jonathan Shewchuk (`http://www.cs.berkeley.edu/~jrs/`). Jean Michel Muller et David Bailey (`http://www.nersc.gov/~dhb/`) participent également à cette action. Le fonds a permis le déplacement de Marc Daumas (visite), Claire Finot-Moreau (visite et séminaire), Sylvie Boldo (stage de longue durée) et Jonathan Shewchuck (visite, invitation au workshop AOC de Novembre 2000, séminaire).

Nous envisageons de prolonger ce travail par un autre financement avec quatre pôles : Berkeley (UCB ou LBL), Intel, ENS Lyon et INRIA Sophia Antipolis.

7 Diffusion de résultats

7.1 Organisation de conférences, édition de numéros spéciaux de journaux

Participants : Marc Daumas, Jean-Michel Muller.

Mots clés : organisation de conférences, édition de revues, numéros spéciaux.

- Marc Daumas et Jean-Michel Muller ont été membres du comité d'organisation de la conférence nationale SYMPA'6 (Besançon, Juin 2000).
- Jean-Michel Muller est éditeur associé de la revue IEEE Transactions on Computers (son double mandat de 4 ans se termine cette année).
- Jean-Michel Muller a fait partie du comité d'organisation et du comité de programme de la conférence RNC4 (*Real Numbers and Computers 4*), qui s'est tenue à Dagstuhl (Allemagne) en avril 2000. Il a également fait partie du comité de programme de la conférence IEEE ASAP'2000 (Boston, juillet 2000);

7.2 Enseignement de 3ème cycle

Participants : Marc Daumas, Philippe Langlois.

Depuis Octobre 1999, Marc Daumas enseigne l'Arithmétique des Ordinateurs au DEA d'Informatique Fondamentale de Lyon. Depuis octobre 2000, Philippe Langlois donne un cours intitulé *Algorithmique numérique et précision finie*, qui est commun aux DEA d'Informatique Fondamentale et d'Analyse Numérique à l'Ecole Normale Supérieure de Lyon.

7.3 Animation de la communauté

Participants : Marc Daumas, Jean-Michel Muller, Arnaud Tisserand.

- Jean-Michel Muller est directeur adjoint du LIP (UMR CNRS-INRIA-ENS Lyon numéro 5668);
- Arnaud Tisserand a animé le séminaire du LIP jusqu'en juin 2000. Il est maintenant responsable de la commission des moyens informatiques du laboratoire;
- Jean-Michel Muller est responsable du thème «Arithmétique» du PRC-GDR ARP. Dans le cadre de ce PRC-GDR, Marc Daumas, Arnaud Tisserand et Jean-Michel Muller ont participé à l'organisation de journées de travail (2 jours chaque fois) à Brest en janvier 2000, à Lyon en mai 2000 et à Montpellier en décembre 2000.

7.4 Participations à des jurys

Jean-Michel Muller est membre élu de la Commission d'Evaluation de l'INRIA. À ce titre il a entre autres participé en 2000 aux Sections d'audition de Rocquencourt et de Grenoble.

Il a participé aux jurys de thèse de Vincent Lefèvre (en tant que directeur de thèse) à Lyon en janvier 2000 et de Laurent Imbert (en tant que président du jury) à Marseille en mai 2000.

7.5 Participation à des colloques, séminaires, invitations

– Marc Daumas a présenté ses travaux lors des Journées Simulation et Pétaflops du Club des Utilisateurs de l'Informatique du CEA, Orsay (oct. 2000). Il a de plus donné un cours intitulé « Les normes de représentation des nombres et leurs implantations », à l'École thématique du CNRS « Conception d'architectures de systèmes informatiques dédiés à des applications spécifiques», Seix (nov. 2000).

– Philippe Langlois a présenté ses travaux à l'Université de Perpignan (en février 2000), au Séminaire de Mathématiques Appliquées de l'Université Lyon 1 (en mai 2000),et au Colloquium de Mathématiques Appliquées de Grenoble (en mars 2000).

– Jean-Michel Muller a été conférencier invité à la *34th Asilomar Conference on Circuits and Systems*. Il a donné des séminaires à Dassault Systèmes (séminaire "CGM", mai 2000), à l'Université d'Evry (Lab. LAMI, juin 2000) et au LIRMM (Montpellier, novembre 2000). Il a également été invité à la "table ronde" de la conférence RenPar12, en juin 2000.

– Arnaud Tisserand a présenté ses travaux lors de séminaires à l'École Polytechnique Fédérale de Lausanne (déc. 2000) et à l'Université de Blois (déc. 2000).

8 Bibliographie

Ouvrages et articles de référence de l'équipe

[1] M. Daumas, J.-M. Muller (éditeurs), *Qualité des calculs sur ordinateur : vers des arithmétiques plus fiables*, Masson, 1997.

[2] T. Lang, J. Muller, N. Takagi (éditeurs), *Proceedings of the 13-th IEEE Symposium on Computer Arithmetic (ARITH-13)*, Asilomar, USA, IEEE Computer Society Press, Los Alamitos, CA, juillet 1997.

[3] J.-M. Muller, *Elementary functions, algorithms and implementation*, Birkhauser, 1997.

Livres et monographies

[4] A. Barraud *et al.*, *Outils d'Analyse Numérique*, Hermès, Paris, 2001, (Auteur du chapitre 1, à paraître).

Thèses et habilitations à diriger des recherches

[5] V. Lefèvre, *Moyens arithmétiques pour un calcul fiable*, thèse de doctorat, École Normale Supérieure de Lyon, Lyon, France, janvier 2000, ftp://ftp.ens-lyon.fr/pub/LIP/Rapports/PhD/PhD2000/PhD2000-02.ps.Z.

Articles et chapitres de livre

[6] T. Braconnier, P. Langlois, J.-C. Rioual, « The influence of orthogonality on the Arnoldi method », *Linear Algebra and Appl. 309*, 1–3, 2000, p. 307–323.

[7] M. Daumas, P. Evripidou, « Parallel implementation of the selection problem: a case study », *International Journal of Parallel Programming 28*, 1, 2000, p. 103–131, http://calliope.inrialpes.fr/htbin/CALLIOPE/docsearch?cn=All+Documents&global=sc005277.

[8] F. de Dinechin, « The Price of Routing in FPGAs », *Journal of Universal Computer Science 6*, 2, février 2000, p. 227–239.

[9] M. Ercegovac, L. Imbert, D. Matula, J. Muller, G. Wei, « Improving Goldschmidt Division, Square Root and Square Root Reciprocal », *IEEE Transactions on Computers 49*, 7, juillet 2000.

[10] M. Ercegovac, T. Lang, J. Muller, A. Tisserand, « Reciprocation, Square Root, Inverse Square Root and Some Elementary Functions Using Small Multipliers », *IEEE Transactions on Computers 49*, 7, juillet 2000.

[11] L. Imbert, J. Muller, F. Rico, « Radix-10 BKM Algorithm for Computing Transcendentals on a Pocket Computer », *Journal of VLSI Signal Processing 25*, 2, juin 2000, p. 179–186.

[12] V. Lefèvre, J.-M. Muller, « L'erreur En Arithmétique Des Ordinateurs », *Le Temps des Savoirs*, 2, October 2000, p. 147–157, in French – en Français.

[13] A. Mignotte, J. Muller, O. Peyran, «Synthesis for Mixed Arithmetic», *Design Automation for Embedded Systems 5*, 1, février 2000.

[14] J. Muller, « Vers Des Primitives Propres En Arithmétique Des Ordinateurs », *Technique et Science Informatiques 19*, janvier 2000.

Communications à des congrès, colloques, etc.

[15] T. Braconnier, P. Langlois, « From rounding error estimation to automatic correction with automatic differentiation », *in: Proceedings of the Third International Conference on Automatic Differentiation: from Simulation to Optimization, Nice, June 2000*, Springer-Verlag, 2000. (A paraître, disponible à http://www.ens-lyon.fr/~~plangloi, RR INRIA numéro 3967, Juillet 2000)., http://www.inria.fr/rrrt/rr-3967.html.

[16] T. Braconnier, P. Langlois, « Parallelism in automatic differentiation for automatic correction of round-off errors », *in: The Third International Conference on Automatic Differentiation: from Simulation to Optimization*, June 2000.

[17] M. Daumas, C. Finot, J. M. Muller, « Table based implementation of elementary functions for hundred-bit precision », *in: Proceedings of the 16th IMACS World Congress on Computational and Applied Mathematics*, Lausanne, Switzerland, 2000, http://imacs2000.epfl.ch/Documents/Sessions/121-2.pdf.

[18] M. Daumas, C. Finot, « Exponential: implementation trade-offs for hundred bit precision », *in : Real Numbers and Computers 4*, p. 61–74, Dagstuhl, Germany, 2000.

[19] M. Daumas, P. Langlois, « Choix d'un support architectural pour le produit scalaire », *in : 6ème Symposium sur les Architectures Nouvelles de Machines*, p. 15–23, Besançon, France, 2000, `http://www.ens-lyon.fr/LIP/Sympa/Sympa6/01.pdf`.

[20] M. Daumas, D. W. Matula, « A Booth multiplier accepting both a redundant or a non-redundant input with no additional delay », *in : IEEE International Conference on Application-specific Systems, Architectures and Processors*, p. 205–214, Boston, Massachusetts, 2000, `ftp://www.ens-lyon.fr/~~daumas/HardArith/DauMat2Kb.ps.gz`.

[21] F. de Dinechin, V. Lefèvre, « Constant Multipliers for FPGAs », *in : Second International Workshop on Engineering of Reconfigurable Hardware/Software Objects (ENREGLE 2000)*, Las Vegas, USA, juin 2000. Also available as LIP research report 2000-18, `ftp://ftp.ens-lyon.fr/pub/LIP/Rapports/RR/RR2000/RR2000-18.ps.Z`.

[22] F. de Dinechin, A. Tisserand, « Table-based methods comparison for low-precision evaluation of the sine and cosine functions on FPGAs », *in : Advanced Signal Processing Algorithms, Architectures, and Implementations X (SPIE'2000)*, F. Luk (éditeur), *4116*, San Diego, California, aug 2000.

[23] L. Imbert, C. Moreau, F. Rico, « Comparison of different techniques to compute the complex exponential for hundred bit precision », *in : IMACS-GAMMM International Symposium on Scientific Computing, Computer Arithmetic and Validated Numerics*, p. 124–125, Karlsruhe, Germany, 2000, `http://iamm22.mathematik.uni-karlsruhe.de/scan/PAGES/abstracts/rico_f.ps`.

[24] P. Langlois, « Applications of the CENA method to automatic accuracy enhancement », *in : SCAN 2000 International Conference on Scientific Computing, Computer Arithmetic and Validated Numerics, Karlsruhe*, GAMM–IMACS, September 2000. (Slides available at URL :, `http://www.ens-lyon.fr/~~plangloi`.

[25] P. Langlois, « Limitations of finite precision correction », *in : RNC-4, Real Numbers and Computer Conference, Open session, Dagsthul*, J.-C. Bajard *et al.* (éditeurs), April 2000. (Slides available at URL :, `http://www.ens-lyon.fr/~~plangloi`.

[26] P. Langlois, « New Results for the CENA Method », *in : The journal BIT 40th anniversary conference, Lund*, August 2000. (Slides available at URL :, `http://www.ens-lyon.fr/~~plangloi`.

[27] P. Langlois, « Stochastic localization of instability and deterministic enhancement of accuracy for iterative algorithms », *in : Proceedings of the 16th IMACS World Congress, Lausanne*, August 2000. (Également disponible à `http://www.ens-lyon.fr/~plangloi`), RR INRIA numéro 3966, Juillet 2000, `http://www.inria.fr/rrrt/rr-3966.html`.

[28] V. Lefèvre, J. Muller, « On-the-Fly Range Reduction », *in : Advanced Signal Processing Algorithms, Architectures, and Implementations X (SPIE'2000)*, F. Luk (éditeur), *4116*, San Diego, California, août 2000.

[29] V. Lefèvre, J. Muller, « Correctly Rounded Functions for Better Arithmetic (conférence invitée) », *in : 34th Asilomar Conference on Circuits and Systems*, IEEE Computer Society Press, Los Alamitos, CA, Pacific Grove (USA), October 2000.

Rapports de recherche et publications internes

[30] M. Daumas, D. W. Matula, «Further reducing the redundancy of a notation over a minimally redundant digit set», *Research report n° 3886*, INRIA, Le Chesnay, France, 2000, `http://www.inria.fr/rrrt/rr-3886.html`.

[31] F. de Dinechin, «Constant Multipliers for FPGAs», *rapport de recherche n° 1999-50*, Laboratoire de l'Informatique du Parallélisme, mai 2000, `ftp://ftp.ens-lyon.fr/pub/LIP/Rapports/RR/RR2000/RR2000-18.ps.Z`.

[32] T. Lang, J. Muller, «Bound on Run of Zeros and Ones for Images of Floating-Point Numbers by Algebraic Functions», *Rapport de Recherche n° RR-4045*, INRIA, Le Chesnay, France, 2000, `http://www.inria.fr/rrrt/rr-4044.html`.

[33] P. Langlois, «A Revised Presentation of the CENA Method», *Research Report n° RR-4025*, INRIA, Rocquencourt, France, October 2000, `http://www.inria.fr/rrrt/rr-4025.html`.

[34] V. Lefèvre, J. Muller, «On-the-fly Range Reduction», *Rapport de Recherche n° RR-4043*, INRIA, Le Chesnay, France, 2000, `http://www.inria.fr/rrrt/rr-4043.html`.

[35] V. Lefèvre, J. Muller, «Worst Cases for Correct Rounding of the Elementary Functions», *Rapport de Recherche n° RR-4044*, INRIA, Le Chesnay, France, 2000, `http://www.inria.fr/rrrt/rr-4044.html`.

Divers

[36] P. Langlois, «Principles and Application of the CENA method», SEA'2000 Workshop, Toulouse, France, june 2000, (Invited Talk).