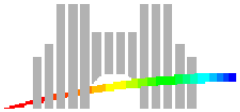
Projet Arénaire
Arithmétique des ordinateurs

CNRS - ENSL - INRIA - LIP - UCBL

Gilles Villard

Visite scientifique de J. Vuillemin - 7 mars 2006



Plan

I - Thématiques générales

Exemples de réalisation

II.1 - Évaluation de fonctions en matériel

II.2 - Efficacité et fiabilité en calcul flottant

II.3 - Complexité algorithmique en algèbre linéaire

III- Perspectives

Plan

15 personnes + S. Torres (30%) + S. Boyer (20%)

Permanents :

N. Brisebarre (U. St-Étienne)
F. de Dinechin (ENSL)
C.-P. Jeannerod (INRIA)
J.-M. Muller (CNRS)
N. Revol (INRIA)
G. Villard (CNRS)

Départs 2005 :

M. Daumas (CNRS, Montpellier-Perpignan)
A. Tisserand (CNRS, Montpellier)

J.-L. Beuchat (FNSRS 2001-2005)

Postdoc : I. Toli (INRIA)

Ingénieur associé : E. Bechetoille (INRIA)

Doctorants : F. Cháves, J. Detrey, C. Lauter, G. Melquiond, R. Michard, S.K. Raina, N. Veyrat-Charvillon

Opérateurs arithmétiques : algorithmes et propriétés

- **Briques de base** : systèmes de représentation, $+, -, \times, \div, \sqrt{\ }, \cdot/cst, \cdot \times cst$, entiers, nombres réels et complexes, intervalles, multi-précision, corps finis
- **Fonctions algébriques, élémentaires et spéciales** : $\sqrt[3]{\ }$, $\log$, $\cos$, erf, . . .
- **Opérateurs composites** : $1/\sqrt{x^2+y^2}$, . . .
- **Spécifiques** : filtres FIR, transformées DCT, opérateurs DSP, cryptographie
- **Propriétés** des opérateurs : représentabilité de résultats, d'erreurs, . . .
- Actions de **normalisation** : IEEE 754, C++

Calculer mieux

Performances	**Qualité**	**Algorithmique**
surface, mémoire	précision	représentations
vitesse	arrondi correct	approximation
énergie	portabilité	méthodes (tables, . . .)
testabilité	fiabilité	analyse numérique
		algèbre

Calculer mieux

↙	↓	↘
Performances	**Qualité**	**Algorithmique**
surface, mémoire	précision	représentations
vitesse	arrondi correct	approximation
énergie	portabilité	méthodes (tables, . . .)
testabilité	fiabilité	analyse numérique
		algèbre

⇝ Outils de conception et méthodologie

Approximation de fonctions : polynomiale, rationnelle, coefficients contraints

Qualité et automatisation : approximation, estimation d'erreur, fiabilité

Preuve formelle : preuve de théorèmes, assistants à la preuve

$\rightsquigarrow$ Utilisation des arithmétiques

Complexité algorithmique : calcul flottant, algébrique/binaire

Algèbre linéaire : matrices polynomiales, calcul mathématique

Optimisation globale : intervalles, réduction de l'intervalle de recherche

Cryptographie : clef publique, signature numérique

Traitement du signal, multimédia

$\rightsquigarrow$ Utilisation des arithmétiques

Complexité algorithmique : calcul flottant, algébrique/binaire

Algèbre linéaire : matrices polynomiales, calcul mathématique

Optimisation globale : intervalles, réduction de l'intervalle de recherche

Cryptographie : clef publique, signature numérique

Traitement du signal, multimédia

Supports d'implantation

Matériel : ASIC, FPGA, asynchrone, basse consommation (Verilog, VHDL . . .)

Couches de base : processeurs dédiés, bibliothèques optimisées, compilation

Logiciel : bibliothèques d'opérateurs, programmes (C, C++, Coq, PVS, Maple . . .)

`http://www.ens-lyon.fr/LIP/Arenaire/Ware`

Plan

I - Thématiques générales

Exemples de réalisation

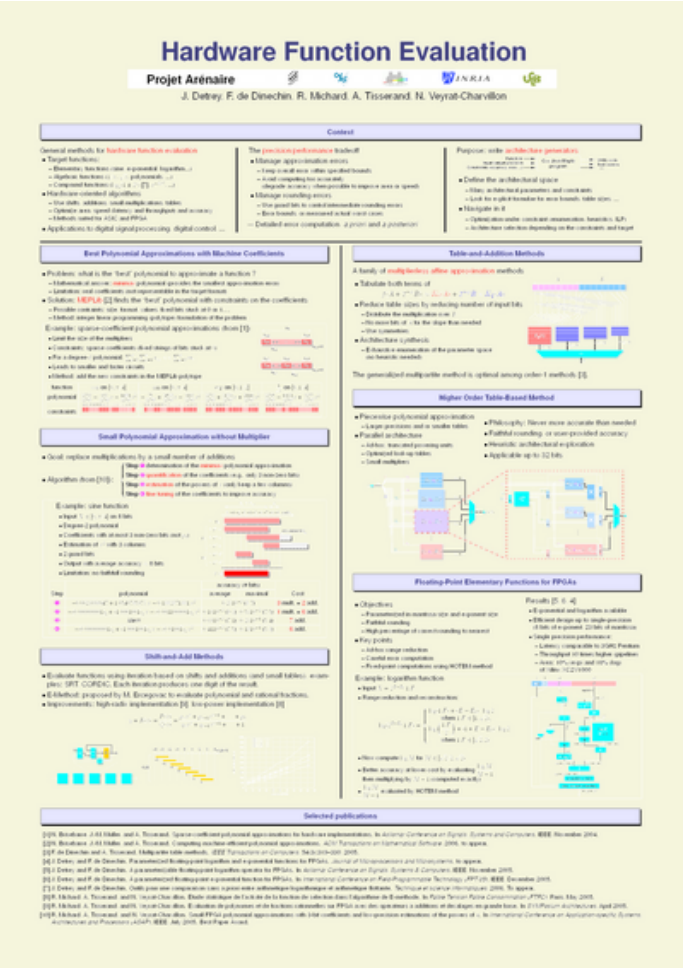
II.1 - Évaluation de fonctions en matériel

II.2 - Efficacité et fiabilité en calcul flottant

II.3 - Complexité algorithmique en algèbre linéaire

III- Perspectives

[Detrey, de Dinechin, Michard, Tisserand, Veyrat-Charvillon]



Fonctions algébriques et élémentaires

- ⟶ génération automatique d'architectures
- ⟶ gestion et estimation des erreurs d'approximation et d'arrondi
- ⟶ méthodes pour ASIC et FPGA
- ⟶ applications en traitement du signal

Expertise et résultats :

Compromis précision/performance
Approximation polynomiale économique/meilleure
Tables multi-partites
Additions/décalages, additions/tables
Fonctions élémentaires en flottant

Plan

I - Thématiques générales

Exemples de réalisation

II.1 - Évaluation de fonctions en matériel

II.2 - Efficacité et fiabilité en calcul flottant

II.3 - Complexité algorithmique en algèbre linéaire

III- Perspectives

Calculs en nombres flottants $\Longleftarrow$

Réduire les **temps d'exécution**

Maîtriser la **précision**

Assurer la **fiabilité**

Accroître la **portabilité**

Participer à la **normalisation**

Calculs en nombres flottants $\Longleftarrow$

Réduire les **temps d'exécution**

Maîtriser la **précision**

Assurer la **fiabilité**

Accroître la **portabilité**

Participer à la **normalisation**

$\Downarrow$

Bibliothèques, algorithmique sophistiquée

Approximations et erreurs d'arrondi

Outils d'aide à la conception et à la validation

↪ Bibliothèque `flip`

[Jeannerod, Raina, Tisserand, Rhône-Alpes/STMicroelectronics]

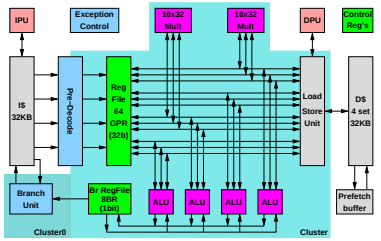
Couche flottante pour processeurs entiers/virgule fixe

Opérateurs arithmétiques de base

$\pm, \times, \div, \sqrt{}$, fma, $1/x, 1/\sqrt{}$

Adéquation algorithmes/architecture

Processeur embarqué ST200 multimédia, traitement du signal sans support flottant matériel



↪ Bibliothèque `flip`

[Jeannerod, Raina, Tisserand, Rhône-Alpes/STMicroelectronics]

Couche flottante pour processeurs entiers/virgule fixe

Opérateurs arithmétiques de base

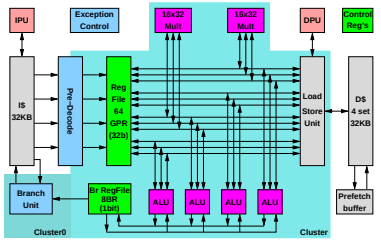
$\pm, \times, \div, \sqrt{\ }$, fma, $1/x$, $1/\sqrt{\ }$

Adéquation algorithmes/architecture

$\Downarrow$

gain en vitesse d'un facteur 1.25 à 4

Processeur embarqué ST200 multimédia, traitement du signal sans support flottant matériel



Répercussions :

- diffusion de l'arithmétique flottante dans le monde de l'embarqué
- architectures et style de développement simples et peu coûteux
- la question du portage en virgule fixe d'un code flottant ne se pose plus

État courant : C LGPL, intégrée dans les compilateurs de production des ST200

Futur : extension à d'autres cibles (DSP TI, Motorola, Analog Device, . . .) [Bechetoille]

↪ Bibliothèque `crlibm` [de Dinechin, Defour, Lauter, Loirat, Muller]

Arrondi correct pour fonctions élémentaires

Peut nécessiter plus de bits que les 53 de la double précision IEEE-754

ex : $\cos(0.837482955647818762922279 49) \approx 0.74296072936219620000000032$

$\Rightarrow$ représentait un obstacle sérieux à la performance

Absence de spécification $\Rightarrow$ erreurs de quelques fois à plusieurs centaines de fois la précision avec les bibliothèques mathématiques courantes [acos `Pentium 4/libc`, cosh `libsunmath`]

↪ Bibliothèque `crlibm`

[de Dinechin, Defour, Lauter, Loirat, Muller]

Arrondi correct pour fonctions élémentaires

Peut nécessiter plus de bits que les 53 de la double précision IEEE-754

ex : $\cos(0.837482955647818762922279 49) \approx 0.74296072936219620000000032$

$\Rightarrow$ représentait un obstacle sérieux à la performance

Absence de spécification $\Rightarrow$ erreurs de quelques fois à plusieurs centaines de fois la précision avec les bibliothèques mathématiques courantes [acos `Pentium 4/libc`, cosh `libsunmath`]

Équipe Arénaire 1997-2006 :

analyse théorique de la précision requise	méthodes à base de tables
calculs exhaustifs de pires cas	optimisation de caches
précision adaptative	précision double étendue, fma
réduction d'argument	techniques pour l'arrondi correct

Répercussions : efficacité, transfert, normalisation

Quatre modes d'arrondis corrects
11 fonctions

	`crlibm/libm` moyenne	max
log, Pentium 4 DLM05	**1.05**	0.57
exp, Itanium 2 Zimmermann 05	**1.06**	1.81

max `libultim` $\approx$ **100**$\times$ max `crlibm`

Utilisateurs

CERN/`LHC@home` : reproductibilité des simulations
plus de 36000 machines
processeurs AMD et Intel (Windows, Linux)

Objectif à moyen terme : GNU `glibc`

L'arrondi correct de certaines fonctions élémentaires est inclus dans l'ébauche actuelle de la révision de la norme IEEE 754 (été 2005)

Mise en œuvre algorithmique

approximation automatisée de fonctions

[Brisebarre, Muller, Tisserand, Torres]

Aide à la conception et fiabilité

estimation automatique des erreurs d'arrondi

preuve formelle

[Boldo, Daumas, Chávez, Melquiond, Revol]

Plan

I - Thématiques générales

Exemples de réalisation

II.1 - Évaluation de fonctions en matériel

II.2 - Efficacité et fiabilité en calcul flottant

II.3 - Complexité algorithmique en algèbre linéaire

III- Perspectives

Matrices polynomiales, entières et par intervalles

[Jeannerod, Revol, Villard]

Comparaison modèle algébrique/modèle binaire

Complexité du calcul certifié

- **Réductions au produit de matrices autres que sur un corps abstrait**
- **Algorithme quasi-optimal d'inversion d'un polynôme matriciel**
- **Meilleure complexité pour le déterminant sans division**
- **Algorithmique exacte pour les très grandes matrices creuses**

Applications et Bibliothèques :

- Contrôle linéaire, optimisation globale, calcul mathématique
- Programmation générique, par composants
- MPFI : `http://perso.ens-lyon.fr/nathalie.revol/software.html`
- LinBox : `http://www.ens-lyon.fr/LIP/Arenaire/Ware`

Plan

I - Thématiques générales

Exemples de réalisation

II.1 - Évaluation de fonctions en matériel

II.2 - Efficacité et fiabilité en calcul flottant

II.3 - Complexité algorithmique en algèbre linéaire

III - Perspectives

Forte demande en **qualité du calcul** :

- coûts réduits
- appréciation des erreurs, augmentation de la précision
- portabilité
- confiance/sûreté des codes

[CEA, IBM, Airbus, Intel]

Côté Arénaire :

- conception des opérateurs arithmétiques
- propriétés du calcul
- calcul scientifique
- maîtrise ad hoc des étapes de conception
- potentiel en transfert de connaissances

Un défi pour le **calcul intensif efficace et fiable** :

la maîtrise combinée {
- des coûts de développement
- des aspects matériels
- des bibliothèques de programmes
- des aspects mathématiques et algorithmiques
- de méthodes de validation

Un défi pour le **calcul intensif efficace et fiable** :

la maîtrise combinée { des coûts de développement
des aspects matériels
des bibliothèques de programmes
des aspects mathématiques et algorithmiques
de méthodes de validation

"Nouveaux outils automatiques pour le calcul flottant validé"

Comment

↪ évaluer une formule mathématique (voire un programme) en calcul flottant ?

↪ automatiser la génération de code pour cette évaluation ?

↪ assurer la qualité des résultats ?

Axe 1. Opérateurs arithmétiques et évaluation de fonctions

- Applications embarquées
- Approximation et évaluation, évaluation d'expression (compilation)
- Opérateurs matériels : virgules fixe et flottante, FPGA
- Opérateurs logiciels : arrondi correct, processeurs entiers, augmentation de la précision

Axe 2. Propriétés, validation, calcul certifié

- Propriétés des opérateurs, fma, opérateurs composites
- Bornes d'erreurs, erreur de méthode/d'arrondi
- Liens avec la preuve formelle
- Complexité algorithmique, modèles de nombres et d'intervalles
- Algorithmes en calcul certifié

Axe 3. Nouveaux outils automatiques

- Méthodologie de développement
- Adaptabilité au matériel
- Génération de code
- Génération de preuve formelle
- Interopérabilité