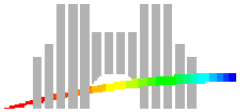
Projet Arénaire

Arithmétique des ordinateurs

CNRS - ENSL - INRIA - LIP - UCBL

Gilles Villard

Comité d'évaluation du LIP - 10 janvier 2006



Plan

I - Présentation et objectifs généraux

II - Un exemple de réalisation :
« efficacité et fiabilité en calcul flottant »

III- Perspectives

Plan

- Création, octobre 1998, **5** personnes

  Permanents :
  M. Daumas, CNRS
  F. de Dinechin, ENSL
  J.-M. Muller, CNRS

  Doctorants :
  V. Lefèvre, C. Finot-Moreau

- Début 2002, **12** personnes

- Janvier 2006, **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

Thématiques de l'équipe

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`

Principales actions académiques 2002-2005

- Coordination de deux actions spécifiques STIC : arithmétique des ordinateurs et calcul formel
- Animations de groupes de travail GDR ARP, GDR MACS
- Animation de l'action AriNews GDR ALP/ARP
- ACI Sécurité, ACI Cryptologie, ACI Jeunes chercheurs, ACI Nouv. interfaces des maths, ANR
- Action INRIA de standardisation IEEE 754, et C++
- Actions intégrées Alliance Cardiff et Oxford
- Action CNRS/NSF Canada France USA LinBox
- PICS CNRS/NASA USA
- Présidences de *steering committees* : ARITH, ISSAC

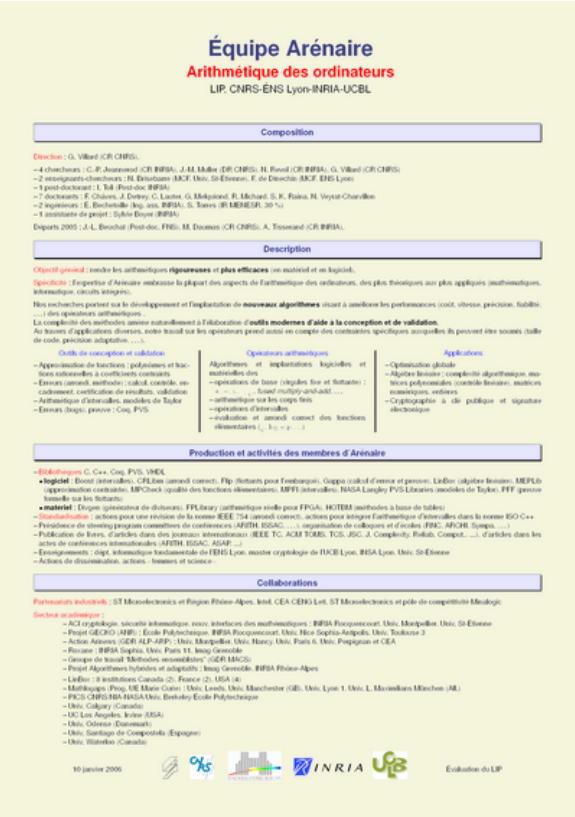
Édition de quatre numéros spéciaux de revues internationales

Organisation de quatre écoles et trois conférences

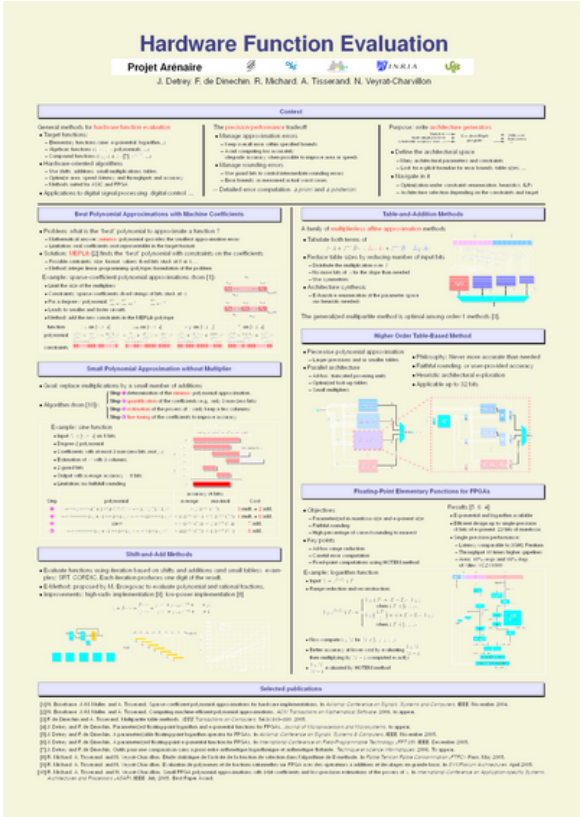
Partenariats industriels

Industrie : Posic S.A., Intel, Région Rhône-Alpes/STMicroelectronics
STMicroelectronics/pôle de compétitivité Minalogic

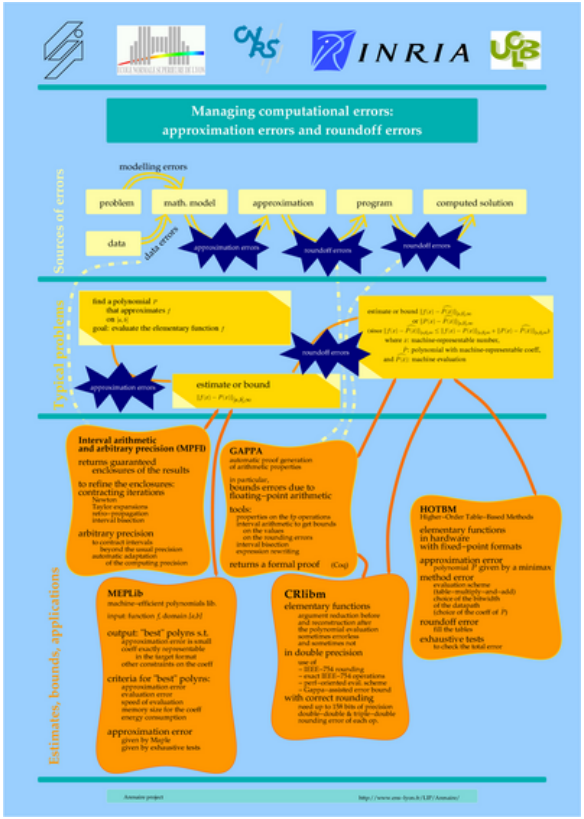
Posters de l'équipe Arénaire



"Équipe Arénaire"



"Hardware function evaluation"



"Managing computational errors"

Plan

I - Présentation et objectifs généraux

II - Un exemple de réalisation :
« efficacité et fiabilité en calcul flottant »

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

II.1 - Bibliothèques logicielles

Exemple 1. Bibliothèque `flip`

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

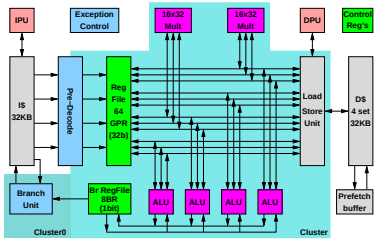
Couche flottante pour processeurs entiers/virgule fixe

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



Exemple 1. Bibliothèque `flip`

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

Couche flottante pour processeurs entiers/virgule fixe

Opérateurs arithmétiques de base

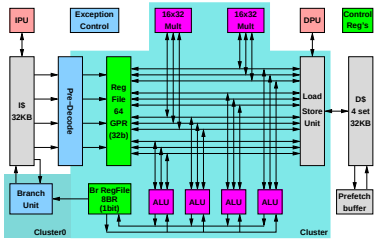
$\pm, \times, \div, \sqrt{\ }$, fma, $1/x$, $1/\sqrt{\ }$

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]

Exemple 2. 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`]

Exemple 2. 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)

II.2 - Mise en œuvre algorithmique
Approximation de fonctions

Approximation polynomiale

Ex. $\cos x$ sur $[0, \pi/4]$ $\approx 0.063\,x^3 - 0.53\,x^2 + 0.005\,x + 0.99$ **Minimax**

$\approx x^3/16 - 17x^2/32 + 5x/1024 + 1$ **Minimax arrondi**

$\approx x^3/16 - 17x^2/32 + 3x/512 + 4095/4096$ **Efficace : `meplib`**

Expertise :

- contraintes de précision
- contraintes sur la représentation
- contraintes de coût à l'évaluation
- approximation par morceaux (tables)

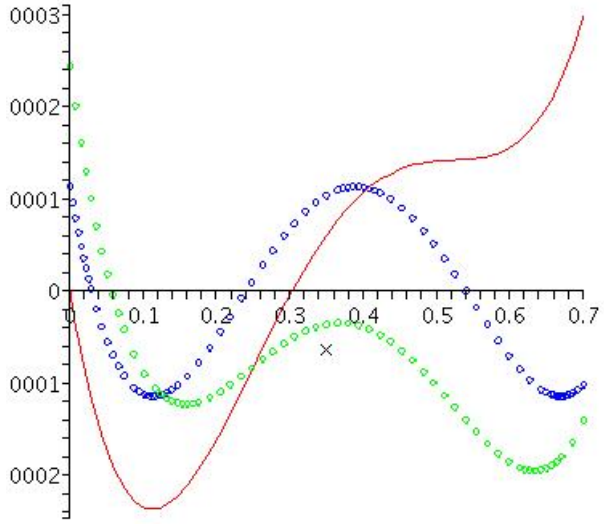


Fig. Courbes des erreurs

Meilleure approximation machine

Contraintes sur le codage des coefficients

Modélisation à base de polytopes

[Brisebarre, Muller, Tisserand, Torres]

II.3 - Aide à la conception et fiabilité
Estimation automatique de la précision

Estimation automatique des erreurs d'arrondi

Ex : $[1/3]_{32} \approx 1/3 + 10^{-8}$

Évaluer ou vérifier la qualité d'une approximation

$\downarrow$

Estimer voire borner l'erreur d'arrondi du programme

Pour un résultat précis et garanti :

Humain

- intelligence pour la preuve
- expertise (analyse numérique)

- difficile (propriétés, volume d'op.)
- fastidieux
- de longue haleine
- pour chaque modification d'algorithme

Estimation automatique des erreurs d'arrondi

Ex : $[1/3]_{32} \approx 1/3 + 10^{-8}$

Évaluer ou vérifier la qualité d'une approximation

$\downarrow$

Estimer voire borner l'erreur d'arrondi du programme

Pour un résultat précis et garanti :

Humain

- intelligence pour la preuve
- expertise (analyse numérique)

$\longrightarrow$

Automatisation

[Chávez, Melquiond, Revol]

calcul d'erreur
arithmétique d'intervalles
propriétés des flottants

logiciel `Gappa`

II.4 - Aide à la conception et fiabilité
Preuve formelle

Preuve formelle

[Boldo, Daumas, Chávez, Melquiond]

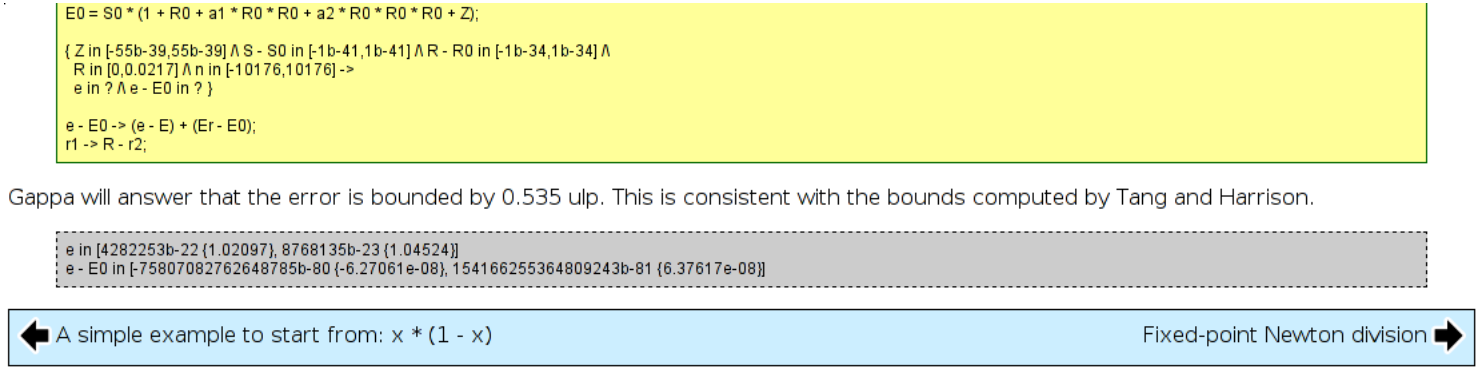
Formalisation (flottants, modèles de Taylor)

Théorèmes de base et propriétés (flottants, intervalles)

Preuve d'algorithmes

Logiciel Gappa → Calcul d'erreur
→ **Assistant à la preuve**
→ **Générateur de preuve en Coq**

Ex : preuve de la fonction exponentielle de Tang



Différents aspects qui interagissent

Recherches fondamentales
\+ Algorithmique
\+ Programmation optimisée

\+ Preuve par recherche exhaustive (pires cas)
\+ Preuve Maple (erreur de méthode)
\+ Preuve « à la main »
\+ Preuve Coq (erreur d'arrondi)

$\Rightarrow$ **Efficacité et fiabilité**

Thématiques non abordées aujourd'hui :

- **Opérateurs matériels et applications**

[Beuchat, De Dinechin, Detrey, Michard, Tisserand, Veyrat-Charvillon, Villard]

- **Complexité algorithmique, arithmétiques et algorithmes**

[Jeannerod, Revol, Villard]

Plan

I - Présentation et objectifs généraux

II - Un exemple de réalisation :
« efficacité et fiabilité en calcul flottant »

III - Perspectives

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

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

Autour du cœur de métier sur les opérateurs, **développer les interactions** :

- adéquation matériels / programmes
- adéquation opérateurs / algorithmes qui les utilisent
- développement, mutualisation et diffusion de bibliothèques de programmes
- bornes d'erreur, preuve

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

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

Autour du cœur de métier sur les opérateurs, **développer les interactions** :

- adéquation matériels / programmes
- adéquation opérateurs / algorithmes qui les utilisent
- développement, mutualisation et diffusion de bibliothèques de programmes
- bornes d'erreur, preuve

Utiliser l'ordinateur comme instrument :

$\Rightarrow$ automatisation, génération de code, outils de validation

Opérateurs arithmétiques

- Applications embarquées
- Approximation et évaluation
- Opérateurs matériels : virgules fixe et flottante, FPGA
- Opérateurs logiciels : arrondi correct, processeurs entiers, augmentation de la précision

Propriétés, validation, calcul certifié

- Propriétés des opérateurs, fma, opérateurs composites
- Bornes d'erreurs, erreur de méthode / d'arrondi
- Preuve formelle
- Complexité algorithmique, modèles de flottants
- Algorithmes en calcul certifié