Algorithmique et arithmétique pour l'algèbre linéaire exacte à partir de la bibliothèque LinBox

Pascal Giorgi

sous la direction de
Gilles Villard

LIP, UMR 5668, CNRS, ENS Lyon, INRIA, UCB Lyon

20 décembre 2004

Motivations

L'algèbre linéaire exacte est un outil de calcul répandu.

Applications en calcul formel :

- bases de Gröbner [Faugère LIP6],
  rang, triangularisation
- cryptographie [Thomé 2003],
  systèmes linéaires creux ($1.033.593 \times 766.150$)
- combinatoire, topologie algébrique [Dumas 2000],
  forme de Smith (376.320×117.600)
- programmation linéaire [contact EDF],
  systèmes linéaires diophantiens (50.000×50.000)
- ...

Diversité des problèmes

$$A = \begin{bmatrix} -289 & 236 & 79 & -268 \\ 108 & -33 & -211 & 309 \\ -489 & 104 & -24 & -25 \\ 308 & 99 & -108 & 66 \end{bmatrix}, \; b = \begin{bmatrix} -131 \\ 321 \\ 147 \\ 43 \end{bmatrix}.$$

Diversité des problèmes

$$A = \begin{bmatrix} -289 & 236 & 79 & -268 \\ 108 & -33 & -211 & 309 \\ -489 & 104 & -24 & -25 \\ 308 & 99 & -108 & 66 \end{bmatrix}, \; b = \begin{bmatrix} -131 \\ 321 \\ 147 \\ 43 \end{bmatrix}.$$

solution sur $\mathbb{Z}_{1009}$

$$x = \begin{bmatrix} 593 \\ 313 \\ 130 \\ 187 \end{bmatrix},$$

Diversité des problèmes

$$A = \begin{bmatrix} -289 & 236 & 79 & -268 \\ 108 & -33 & -211 & 309 \\ -489 & 104 & -24 & -25 \\ 308 & 99 & -108 & 66 \end{bmatrix}, \; b = \begin{bmatrix} -131 \\ 321 \\ 147 \\ 43 \end{bmatrix}.$$

solution sur $\mathbb{Z}_{1009}$

$$x = \begin{bmatrix} 593 \\ 313 \\ 130 \\ 187 \end{bmatrix},$$

solution sur $\mathbb{Q}$

$$x = \begin{bmatrix} \frac{-9591197817}{95078} \\ \frac{131244}{47539} \\ \frac{2909895}{665546} \\ \frac{2909895}{665546} \end{bmatrix},$$

Diversité des problèmes

$$A = \begin{bmatrix} -289 & 236 & 79 & -268 \\ 108 & -33 & -211 & 309 \\ -489 & 104 & -24 & -25 \\ 308 & 99 & -108 & 66 \end{bmatrix}, \; b = \begin{bmatrix} -131 \\ 321 \\ 147 \\ 43 \end{bmatrix}.$$

solution sur $\mathbb{Z}_{1009}$

$$x = \begin{bmatrix} 593 \\ 313 \\ 130 \\ 187 \end{bmatrix},$$

solution sur $\mathbb{Q}$

$$x = \begin{bmatrix} \frac{-9591197817}{95078} \\ \frac{131244}{47539} \\ \frac{2909895}{665546} \\ \frac{2909895}{665546} \end{bmatrix},$$

solution sur $\mathbb{Z}$

$$x = \begin{bmatrix} -106495695463 \\ -2208888459779 \\ -4204431397194 \\ -3069666048124 \end{bmatrix}$$

Diversité des problèmes

$$A = \begin{bmatrix} -289 & 236 & 79 & -268 \\ 108 & -33 & -211 & 309 \\ -489 & 104 & -24 & -25 \\ 308 & 99 & -108 & 66 \end{bmatrix}, \; b = \begin{bmatrix} -131 \\ 321 \\ 147 \\ 43 \end{bmatrix}.$$

solution sur $\mathbb{Z}_{1009}$

$$x = \begin{bmatrix} 593 \\ 313 \\ 130 \\ 187 \end{bmatrix},$$

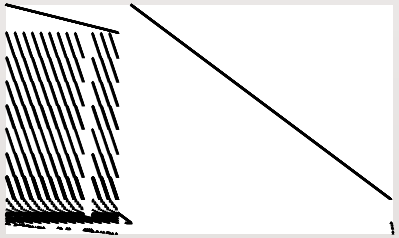
solution sur $\mathbb{Q}$

$$x = \begin{bmatrix} \frac{-9591197817}{95078} \\ \frac{131244}{47539} \\ \frac{2909895}{665546} \\ \frac{2909895}{665546} \end{bmatrix},$$

solution sur $\mathbb{Z}$

$$x = \begin{bmatrix} -106495695463 \\ -2208888459779 \\ -4204431397194 \\ -3069666048124 \end{bmatrix}$$

A peut être creuse (seulement $O(n)$ éléments non nuls)



problème d'optimisation (emploi du temps)
3202×4048, ≈ 19000 coeff. non nuls.

Diversité des problèmes

$$A = \begin{bmatrix} -289 & 236 & 79 & -268 \\ 108 & -33 & -211 & 309 \\ -489 & 104 & -24 & -25 \\ 308 & 99 & -108 & 66 \end{bmatrix}, \quad b = \begin{bmatrix} -131 \\ 321 \\ 147 \\ 43 \end{bmatrix}.$$

solution sur $\mathbb{Z}_{1009}$

$$x = \begin{bmatrix} 593 \\ 313 \\ 130 \\ 187 \end{bmatrix},$$

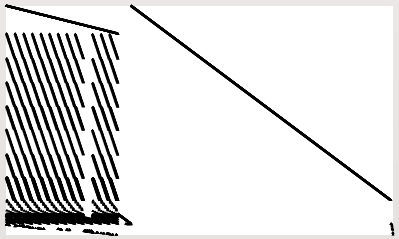
solution sur $\mathbb{Q}$

$$x = \begin{bmatrix} \frac{-9591197817}{95078} \\ \frac{131244}{47539} \\ \frac{2909895}{665546} \\ \frac{2909895}{665546} \end{bmatrix},$$

solution sur $\mathbb{Z}$

$$x = \begin{bmatrix} -106495695463 \\ -2208888459779 \\ -4204431397194 \\ -3069666048124 \end{bmatrix}$$

A peut être creuse (seulement $O(n)$ éléments non nuls)



problème d'optimisation (emploi du temps)
3202×4048, ≈ 19000 coeff. non nuls.

Aujourd'hui, il existe des algorithmes efficaces pour résoudre certains problèmes d'algèbre linéaire exacte

De réelles attentes...

Les gains algorithmiques récents sont importants (gains linéaires, algorithmes optimaux).

Des logiciels généralistes comme MAPLE ou MATHEMATICA ne sont plus dominants.

Existence de bibliothèques spécialisées très performantes :

- GMP : arithmétiques multiprécisions (entiers, rationnels, flottants).
- NTL : arithmétiques des polynômes, des corps finis.
- BLAS/LAPACK : algèbre linéaire numérique.

De réelles attentes...

Les gains algorithmiques récents sont importants (gains linéaires, algorithmes optimaux).

Des logiciels généralistes comme MAPLE ou MATHEMATICA ne sont plus dominants.

Existence de bibliothèques spécialisées très performantes :

- GMP : arithmétiques multiprécisions (entiers, rationnels, flottants).
- NTL : arithmétiques des polynômes, des corps finis.
- BLAS/LAPACK : algèbre linéaire numérique.

algorithmes récents + bibliothèques de calcul spécialisées
$\Rightarrow$ calcul en algèbre linéaire exacte de hautes performances.

Développement logiciel

Points clés :

- Diversité des composantes pour le calcul exact.

Développement logiciel

Points clés :

- Diversité des composantes pour le calcul exact.
  - arithmétiques :

    corps finis, grands entiers, polynômes, corps de fractions, extensions algébriques, ...

Développement logiciel

Points clés :

- Diversité des composantes pour le calcul exact.
  - arithmétiques :

    corps finis, grands entiers, polynômes, corps de fractions, extensions algébriques, ...
  - algorithmiques :

    méthodes d'élimination, méthodes itératives, restes chinois, développements p-adiques, ...

Développement logiciel

Points clés :

- Diversité des composantes pour le calcul exact.
  - arithmétiques :

    corps finis, grands entiers, polynômes, corps de fractions, extensions algébriques, ...
  - algorithmiques :

    méthodes d'élimination, méthodes itératives, restes chinois, développements p-adiques, ...
- Meilleurs algorithmes souvent probabilistes (Monte Carlo, Las Vegas).

Développement logiciel

Points clés :

- Diversité des composantes pour le calcul exact.
  - arithmétiques :

    corps finis, grands entiers, polynômes, corps de fractions, extensions algébriques, ...
  - algorithmiques :

    méthodes d'élimination, méthodes itératives, restes chinois, développements p-adiques, ...
- Meilleurs algorithmes souvent probabilistes (Monte Carlo, Las Vegas).
- Spécifications informatiques : représentation et typage des données.

Développement logiciel

Points clés :

- Diversité des composantes pour le calcul exact.
  - arithmétiques :

    corps finis, grands entiers, polynômes, corps de fractions, extensions algébriques, ...
  - algorithmiques :

    méthodes d'élimination, méthodes itératives, restes chinois, développements p-adiques, ...
- Meilleurs algorithmes souvent probabilistes (Monte Carlo, Las Vegas).
- Spécifications informatiques : représentation et typage des données.
- Interaction avec des bibliothèques spécialisées.

Questions

Comment développer une bibliothèque de calcul de hautes performances en algèbre linéaire exacte ?

Comment bénéficier des performances de bibliothèques spécialisées comme GMP, NTL ou BLAS ?

Quelle stratégie adopter pour profiter au mieux des évolutions futures dans le domaine ?

Plan de l'exposé

Projet LINBOX

Implantation générique des corps finis

- modèle de base et interface
- un exemple de *"plug-in"*

Algèbre linéaire dense sur un corps fini

- réduction au produit de matrices
- paquetages FFLAS-FFPACK
- performances

Systèmes linéaires diophantiens

- interface LINBOX pour la résolution
- performances pour des systèmes denses.

Conclusion et perspectives

LINBOX en détails

Projet international Canada-France-USA (NSF/CNRS).

- depuis 1997, 33 chercheurs $\Longrightarrow$ algèbre linéaire exacte
- bibliothèque C++, licence GPL, 100.000 lignes de code, version développement 0.2.0 (juin 2004)
- site web : *www.linalg.org*

Principaux développements :

- algorithmes (systèmes linéaires, formes normales, ...),
- matrices (boîtes noires, conteneurs),
- domaines de calcul (corps finis, entiers, rationnels),
- généricité (plug&play).

Nos travaux :

recherche et implantation d'algorithmes,
développement, validation et maintenance de la bibliothèque.

Solutions algorithmiques dans LINBOX

Sur les corps finis,

- Gauss par blocs (déterministe) :

  triangularisation, déterminant, rang, inverse, systèmes linéaires, polynômes minimal et caractéristique

- Élimination creuse (déterministe) :

  triangularisation, rang

- Krylov/Wiedemann [blocs], Lanczos [blocs] (probabliliste) :

  systèmes linéaires, polynôme minimal, déterminant, rang

Sur les entiers,

- Théorème des restes chinois
- Forme de Smith
- Systèmes linéaires : solutions rationnelles, diophantiennes ; certificat d'inconsistance, minimalité.

Solutions algorithmiques dans LINBOX

Nos contributions

Sur les corps finis,

- Gauss par blocs (déterministe) :

  triangularisation, déterminant, rang, inverse, systèmes linéaires, polynômes minimal et caractéristique

- Élimination creuse (déterministe) :

  triangularisation, rang

- Krylov/Wiedemann [blocs], Lanczos [blocs] (probabliliste) :

  systèmes linéaires, polynôme minimal, déterminant, rang

Sur les entiers,

- Théorème des restes chinois
- Forme de Smith
- Systèmes linéaires : solutions rationnelles, diophantiennes ; certificat d'inconsistance, minimalité.

LINBOX en pratique...

- Challenge de Trefethen "*100-Dollars, 100-Digit Challenge*" [SIAM 2002] [Bornemann-Laurie-Wagon-Waldvogel].
calculer $A^{-1}(1,1)$ avec A une matrice creuse 20000×20000.

$$A_{i,j} = \begin{cases} i\text{ème nb. premier si } i = j \\ 1 \text{ si } |i-j| \text{ est une puissance de 2} \\ 0 \text{ sinon} \end{cases}$$

$\Rightarrow$ solution numérique avec un maximum de précision.

LINBOX en pratique...

- Challenge de Trefethen "*100-Dollars, 100-Digit Challenge*" [SIAM 2002] [Bornemann-Laurie-Wagon-Waldvogel].

calculer $A^{-1}(1,1)$ avec A une matrice creuse 20000×20000.

$$A_{i,j} = \begin{cases} i\text{ème nb. premier si } i = j \\ 1 \text{ si } |i-j| \text{ est une puissance de 2} \\ 0 \text{ sinon} \end{cases}$$

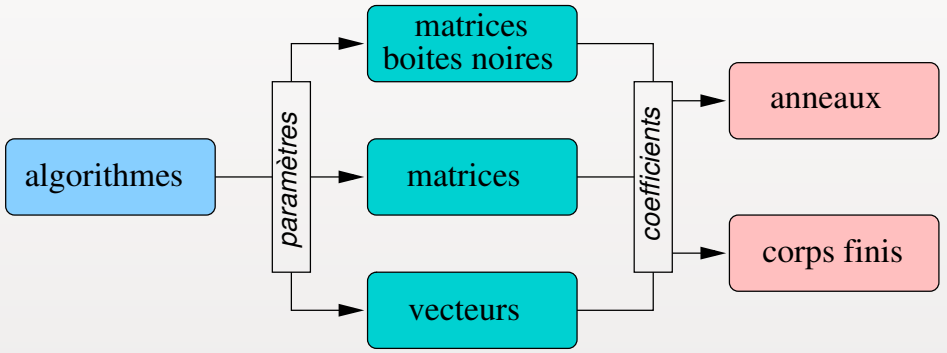
$\Rightarrow$ solution numérique avec un maximum de précision.

Solution exacte avec LINBOX (nb. rationnel de 100000 chiffres) [Dumas, Turner, Wan 2002-2004]

quotient de deux déterminants Wiedemann + CRT		4 jours (parallèle) 180 processeurs $\approx$ 2 ans CPU 96 PIII-735Mhz, 6 PIII-1Ghz 20 4 $\times$ 250Mhz Sun Ultra-450
solution $Ax = [1, 0, \ldots, 0]^T$ p-adique (dense) + reconstruction		12.5 jours (séquentiel) Sun Sunfire 750Mhz Ultrasparc
solution $Ax = [1, 0, \ldots, 0]^T$ semi-numérique (BLAS/LAPACK)		25 minutes (séquentiel) Pentium 1.9Ghz

La bibliothèque LINBOX : principes de généricité

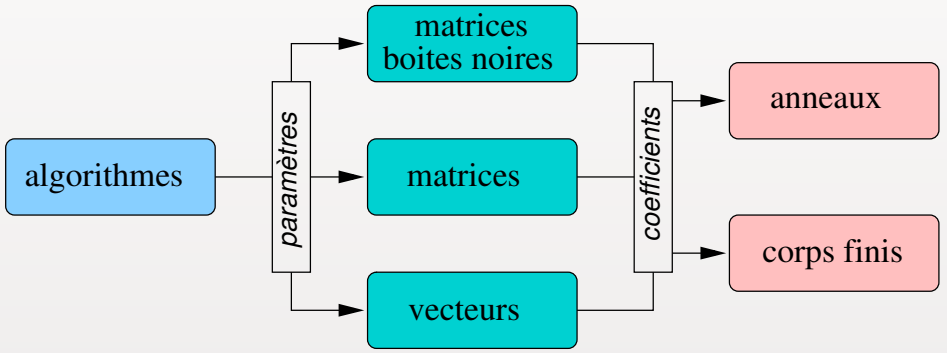
Trois niveaux d'implantation (réutilisation et reconfiguration)



Définition par des classes (dissociation données/manipulation)

La bibliothèque LINBOX : principes de généricité

Trois niveaux d'implantation (réutilisation et reconfiguration)



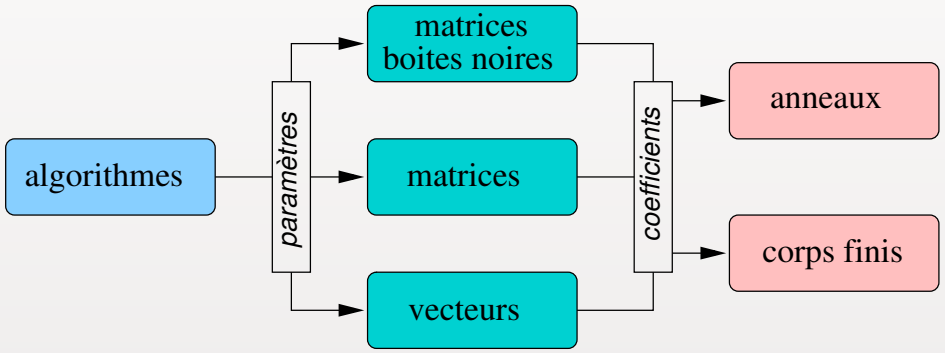
Définition par des classes (dissociation données/manipulation)

Généricité : modèles de base et *template*.
⇒ intégration de codes externes via des adaptateurs (*wrapper*)

Alternative au polymorphisme statique

La bibliothèque LINBOX : principes de généricité

Trois niveaux d'implantation (réutilisation et reconfiguration)



Définition par des classes (dissociation données/manipulation)

Généricité : modèles de base et *template*.
⇒ intégration de codes externes via des adaptateurs (*wrapper*)

Alternative au polymorphisme statique (archétype ≈ interface Java) :

- fournit une instance de code compilé,
- contrôle l'explosion de code,
- utilisation optionnelle.

Projet LINBOX

Implantation générique des corps finis

- modèle de base et interface
- un exemple de *"plug-in"*

Algèbre linéaire dense sur un corps fini

- réduction au produit de matrices
- paquetages FFLAS-FFPACK
- performances

Systèmes linéaires diophantiens

- interface LINBOX pour la résolution
- performances pour des systèmes denses.

Conclusion et perspectives

Structure des corps finis

Corps finis $\Leftrightarrow$ Domaines de calcul.

- élément : représentation (structure des données),
- domaine : méthode de calcul (lois mathématiques).

Éléments : aucune information sur le corps (gain mémoire).

Le domaine encapsule les types (*éléments*, *générateurs aléatoires*).

Méthodes du domaine : *affectation*, *égalité*, *arithmétique*, *IO*.

x = y	:	F.assign(x,y)
x == y	:	F.areEqual(x,y)
x = y + z	:	F.add(x,y,z)
cout<< x	:	F.write(cout,x)

La plupart des implantations sont externes à la bibliothèque
$\rightarrow$ intégration par des wrappers

LINBOX : implantations de corps finis.

Diverses bibliothèques :

- GIVARO (`www-apache.imag.fr/software/givaro/`),
- LIDIA (`www.informatik.tu-darmstadt.de/TI/LiDIA/`),
- NTL (`www.shoup.net/ntl/`),
- GMP (`www.swox.com/gmp/`),

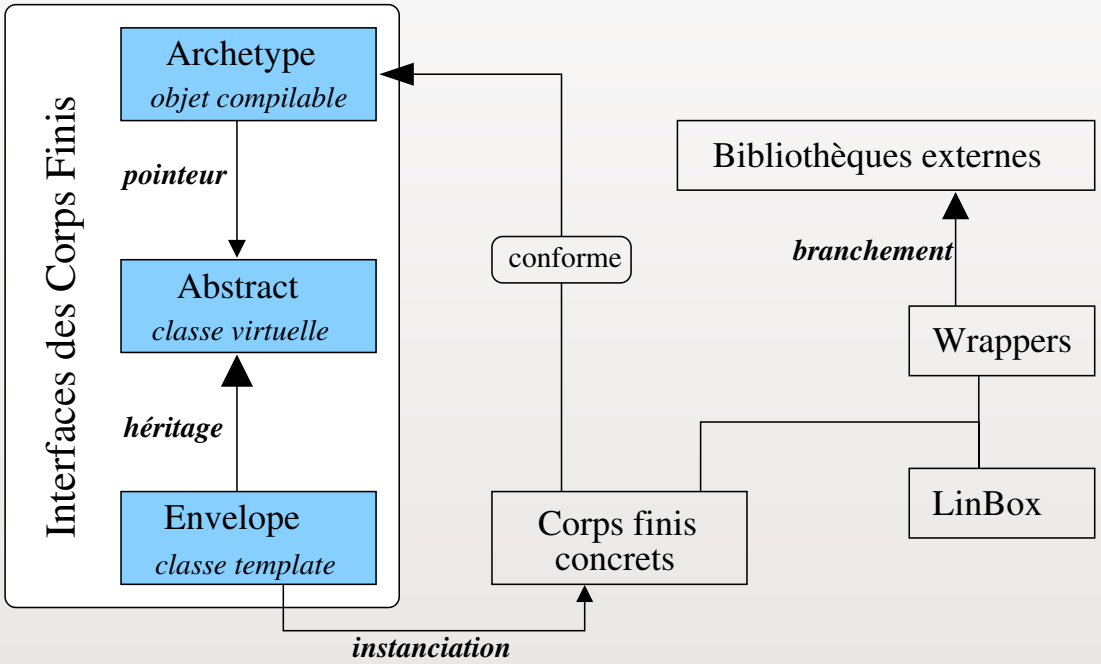
ou des implantations propres à LINBOX.

Divers corps finis :

- corps premiers $\mathbb{Z}_p$,
- extensions algébriques $\texttt{GF}(2^k)$, $\texttt{GF}(p^k)$.

Utilisation maximale des précisions machines
$\Rightarrow$ meilleures performances.

Abstraction du domaine de calcul via une interface générique : l'archétype [Kaltofen, LINBOX]



Avantages :

- dissociation de l'interface dans les implantations
- gestion automatique des allocations de données
- réutilisation des codes génériques *template*
- généricité des codes compilés *"plug-in"*

Projet LINBOX

Implantation générique des corps finis

- modèle de base et interface
- un exemple de *"plug-in"*

Algèbre linéaire dense sur un corps fini

- réduction au produit de matrices
- paquetages FFLAS-FFPACK
- performances

Systèmes linéaires diophantiens

- interface LINBOX pour la résolution
- performances pour des systèmes denses.

Conclusion et perspectives

Des performances non sacrifiées

	Produit scalaire (ordre 1000)				
implantation	corps	boucle	**direct**	**wrapper**	**archétype**
GivaroZpz<Std32>	$\mathbb{Z}_{1009}$	10.000 100.000	0.41s 4.00s	0.42s 4.01s	0.79s 7.41s
NTL_zz_p	$\mathbb{Z}_{1009}$	10.000 100.000	1.97s 19.35s	2.11s 20.97s	2.30s 22.77s
NTL_ZZ_pE	$\mathtt{GF}(3^7)$	10.000	272.11s	279.23s	280.31s

Coût de l'archétype quasi constant
$\Rightarrow$ impact faible sur des calculs coûteux (e.g. extension algébrique)

Concept archétype, domaine, wrapper valide pour les autres composantes de LINBOX (boîtes noires, matrices,...).

LINBOX : compromis performance/généricité très acceptable...

Projet LINBOX

Implantation générique des corps finis

- modèle de base et interface
- un exemple de *"plug-in"*

Algèbre linéaire dense sur un corps fini

- réduction au produit de matrices
- paquetages FFLAS-FFPACK
- performances

Systèmes linéaires diophantiens

- interface LINBOX pour la résolution
- performances pour des systèmes denses.

Conclusion et perspectives

Algorithmique

Depuis 1969, multiplication de matrices d'ordre n en moins de $O(n^3)$ opérations arithmétiques.

[Strassen 1969] : $O(n^{2.81})$.

...

[Coppersmith-Winograd 1990] : $O(n^{2.37})$

Les meilleurs algorithmes se réduisent à la multiplication de matrices (complexité $O(n^\omega)$).

- inversion, systèmes linéaires, déterminant [Strassen 1969]
- extension aux matrices non génériques [Bunch-Hopcroft 1974]
- matrices singulières : LQUP, rang, noyau [Ibarra-Moran-Hui 1982]

Algorithmique

Depuis 1969, multiplication de matrices d'ordre n en moins de $O(n^3)$ opérations arithmétiques.

[Strassen 1969] : $O(n^{2.81})$.

...

[Coppersmith-Winograd 1990] : $O(n^{2.37})$

Les meilleurs algorithmes se réduisent à la multiplication de matrices (complexité $O(n^\omega)$).

- inversion, systèmes linéaires, déterminant [Strassen 1969]
- extension aux matrices non génériques [Bunch-Hopcroft 1974]
- matrices singulières : LQUP, rang, noyau [Ibarra-Moran-Hui 1982]

la multiplication de matrices est une opération centrale

Bibliothèques BLAS

Calculs numériques pour les opérations de base en algèbre linéaire

- produit matrice-vecteur,
- résolution de systèmes linéaires triangulaires,
- produit de matrices,
- ...

Collection de routines Fortran/C optimisées.
tire partie de la hierarchisation mémoire des processeurs.

$\longrightarrow$ implantations très performantes

Bibliothèques BLAS

Calculs numériques pour les opérations de base en algèbre linéaire

- produit matrice-vecteur,
- résolution de systèmes linéaires triangulaires,
- produit de matrices,
- ...

Collection de routines Fortran/C optimisées.
tire partie de la hierarchisation mémoire des processeurs.

$\longrightarrow$ implantations très performantes

Est-il possible de réutiliser efficacement les routines BLAS pour des calculs sur les corps finis ?

Multiplication de matrices sur un corps finis

Multiplication de matrices sur $\mathbb{Z}_p$ [Dumas-Gautier-Pernet 2002]
le calcul numérique doit rester exact : $n(p-1)^2 < 2^{53}$

- conversion corps premier $\Rightarrow$ nombres flottants (`double`)
- multiplication BLAS (routine `dgemm`)
- conversion nombres flottants $\Rightarrow$ corps premier

Multiplication de matrices sur un corps finis

Multiplication de matrices sur $\mathbb{Z}_p$ [Dumas-Gautier-Pernet 2002]
le calcul numérique doit rester exact : $n(p-1)^2 < 2^{53}$

- conversion corps premier $\Rightarrow$ nombres flottants (`double`)
- multiplication BLAS (routine `dgemm`)
- réduction modulo

Amélioration :
corps premiers en nombres flottants (`Modular<double>`)
$\Rightarrow$ 69s pour des matrices d'ordre 5000 sur $\mathbb{Z}_{101}$

temps de calcul très proches de ceux de la bibliothèque BLAS.

Systèmes linéaires triangulaires matriciels

Étant données $A, B \in \mathbb{Z}_p^{n\times n}$, A triangulaire.
Calculer $X \in \mathbb{Z}_p^{n\times n}$ tel que $AX = B$.

Conditions d'utilision des BLAS sur $\mathbb{Z}_p$:

- les divisions doivent être exactes.
- aucun dépassement de capacité (max$=$ 53 bits).
- utilisation limitée (e.g. $p = 2 \rightarrow n \leq 55$).

Systèmes linéaires triangulaires matriciels

Étant données $A, B \in \mathbb{Z}_p^{n \times n}$, A triangulaire.
Calculer $X \in \mathbb{Z}_p^{n \times n}$ tel que $AX = B$.

Conditions d'utilision des BLAS sur $\mathbb{Z}_p$:

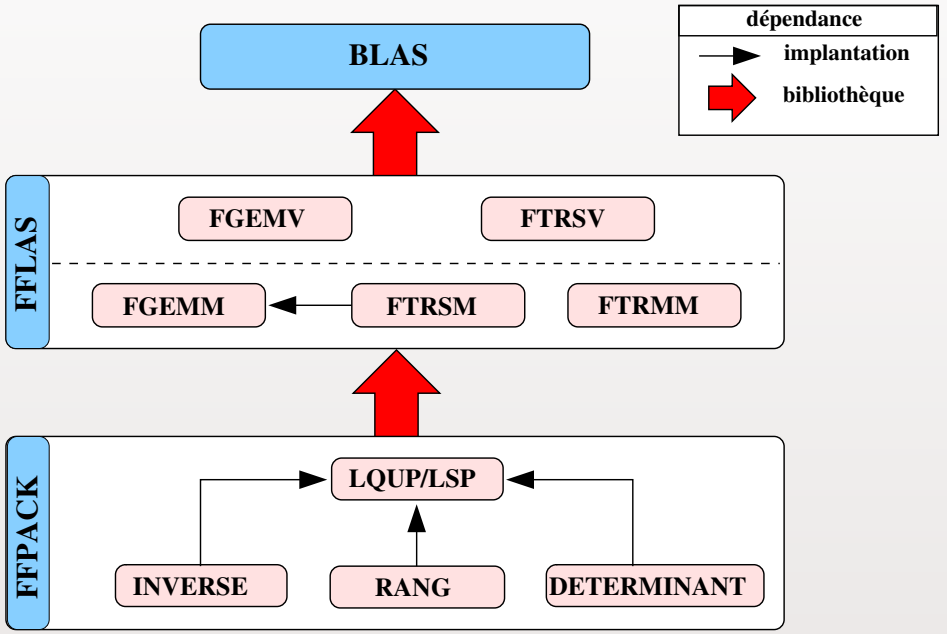
- les divisions doivent être exactes.
- aucun dépassement de capacité (max= 53 bits).
- utilisation limitée (e.g. $p = 2 \longrightarrow n \leq 55$).

Solution :
algorithme récursif par blocs + seuil de changement
$\Rightarrow$ diminution des dimensions , produit de matrices

Deux bénéfices apportés par la bibliothèque BLAS :
produit de matrices $\longrightarrow$ appels récursifs
résolution numérique $\longrightarrow$ derniers niveaux récursifs

Implantations FFLAS-FFPACK[1]

Collaboration avec J-G. Dumas et C. Pernet (LMC-IMAG)

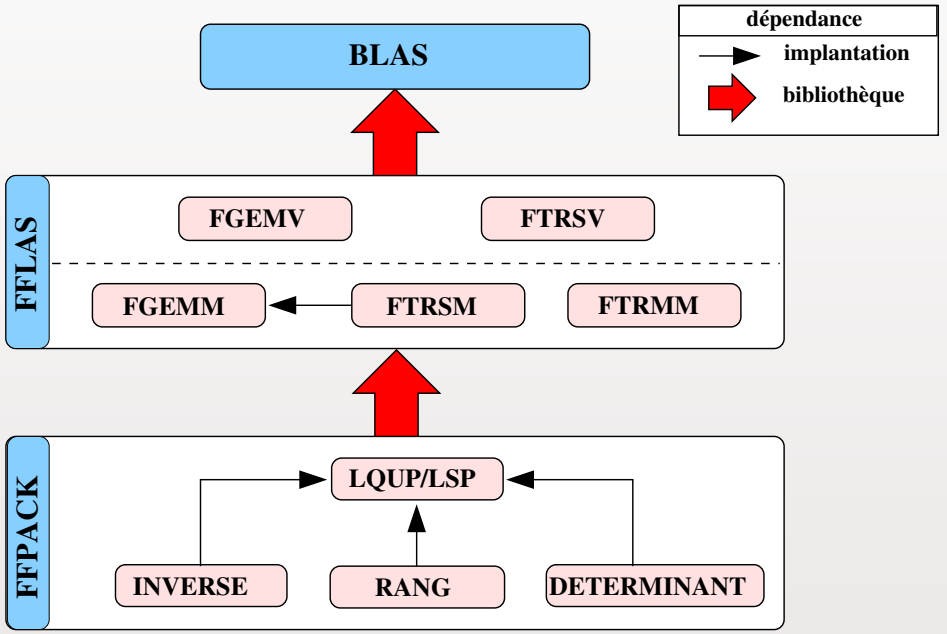


- routines C++ génériques (corps finis).
- interface à la BLAS (tableau de données, *stride*).
- minimisation espace mémoire (calcul en place).
- dépendances aux BLAS et au produit de matrices `fgemm`.

[1] J.-G. Dumas, P. Giorgi and C. Pernet. FFPACK : Finite Field Linear Algebra Package. In *Proceedings of the 2004 International Symposium on Symbolic and Algebraic Computation*, ACM Press New York.

Implantations FFLAS-FFPACK[1]

Collaboration avec J-G. Dumas et C. Pernet (LMC-IMAG)



- routines C++ génériques (corps finis).
- interface à la BLAS (tableau de données, *stride*).
- minimisation espace mémoire (calcul en place).
- dépendances aux BLAS et au produit de matrices `fgemm`.

  améliorations BLAS et `fgemm` ⇒ meilleures routines

[1] J.-G. Dumas, P. Giorgi and C. Pernet. FFPACK : Finite Field Linear Algebra Package. In *Proceedings of the 2004 International Symposium on Symbolic and Algebraic Computation*, ACM Press New York.

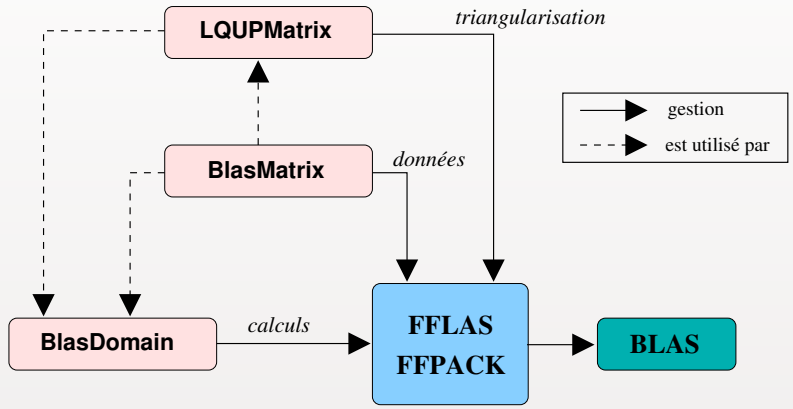
Exemple : code pour l'inversion

Inverse(A) :

décomposer $A = LUP$;
calculer $T = L^{-1}$;
résoudre $UX = T$;
retourner XP^t;

```
 template <class Field>
static typename Field::Element*
Invert(const Field &F, const size_t N,
       typename Field::Element * A, const size_t lda,
       typename Field::Element * X, const size_t ldx,
       int &nullity)
{
   typename Field::Element one ;
   F.init(one,1) ;
   size_t *P = new size_t[N] ;
   size_t *Q = new size_t[N] ;
   nullity = N - LUdivine(F, FflasNonUnit, N, N, A, lda, P, FflapackLQUP, Q) ;
   delete[] Q ;
   if (nullity > 0)
       return NULL ;
   else {
       invL(F, N, A, lda, X, ldx) ;
       ftrsm(F, FflasLeft, FflasUpper, FflasNoTrans, FflasNonUnit, N, N, one, A, lda , X, ldx) ;
       applyP(F, FflasLeft, FflasTrans, N, 0, N, X, ldx, P) ;
       delete[] P ;
       return X ;
   }
}
```

Une interface dédiée aux non experts...




```
Field F(p) ;
BlasMatrix<Field::Element> A(n,n), B(n,n), X(n,n) ;
TriangularBlasMatrix<Field::Element> L(n,n,BlasTag::low,BlasTag::unit) ;
std::vector<Field::Element> b(n), x(n) ;

...

BlasMatrixDomain<Field> BMD(F) ;

BMD.left_solve(X,A,B) ; // AX=B              BMD.right_solve(X,A,B) ; // XA=B
BMD.left_solve(x,A,b) ; // Ax=b              BMD.right_solve(x,A,b) ; // xA=b
BMD.left_solve(X,L,B) ; // LX=B              BMD.right_solve(X,L,B) ; // XL=B
BMD.left_solve(x,L,b) ; // Lx=b              BMD.right_solve(x,L,b) ; // xL=b

BMD.mul(X,A,B) ; // X=A*B
BMD.mul(x,A,b) ; // x=A*b
BMD.mul(X,L,B) ; // X=L*B

int r     = BMD.rank(A) ;    // r=rang(A)
int r     = BMD.rankin(A) ;  // r=rang(A), calcul en place
Element d = BMD.det(A) ;     // d=déterminant(A)
Element d = BMD.detin(A) ;   // d=déterminant(A), calcul en place
```

Projet LINBOX

Implantation générique des corps finis

- modèle de base et interface
- un exemple de *"plug-in"*

Algèbre linéaire dense sur un corps fini

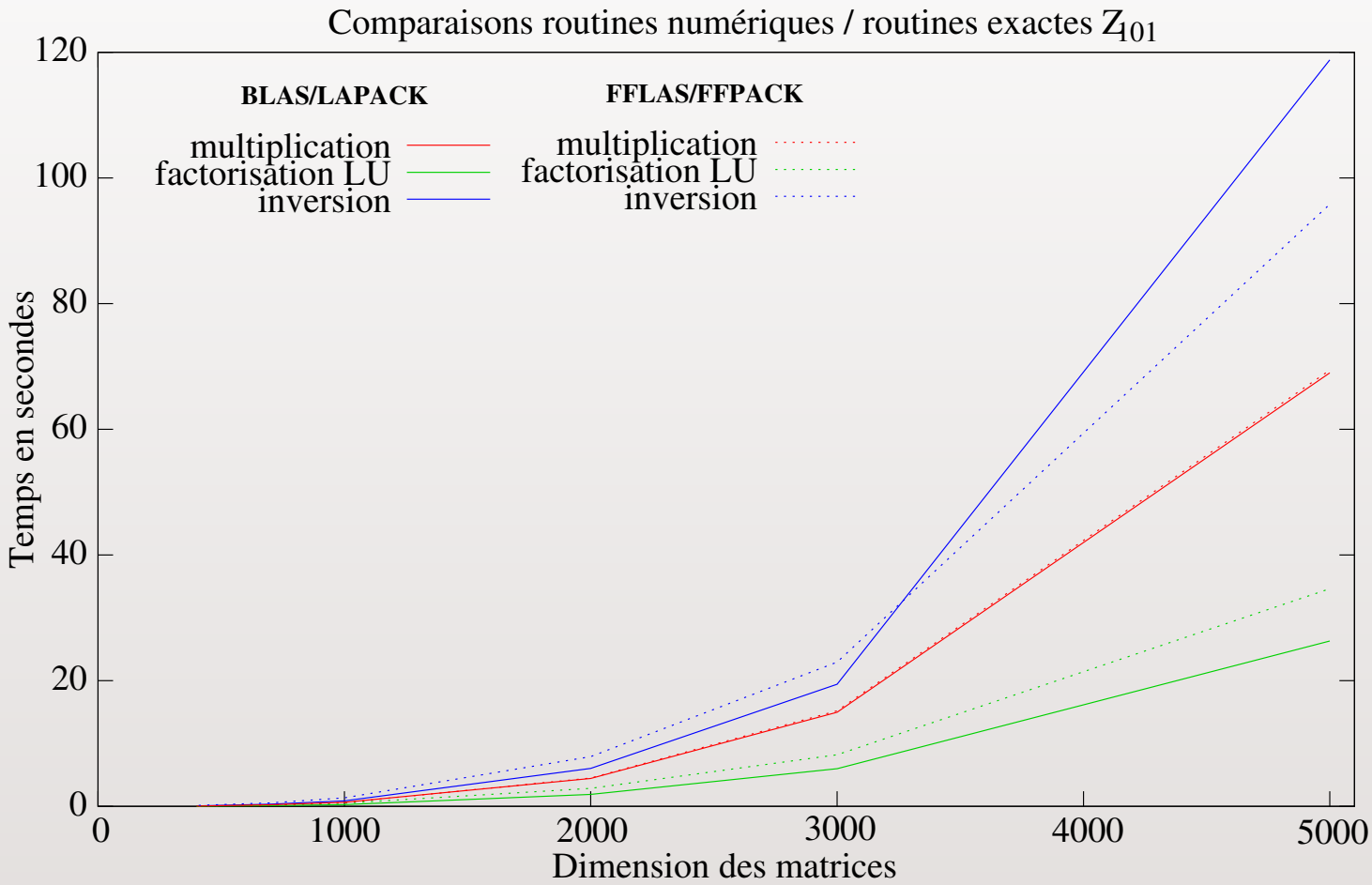
- réduction au produit de matrices
- paquetages FFLAS-FFPACK
- performances

Systèmes linéaires diophantiens

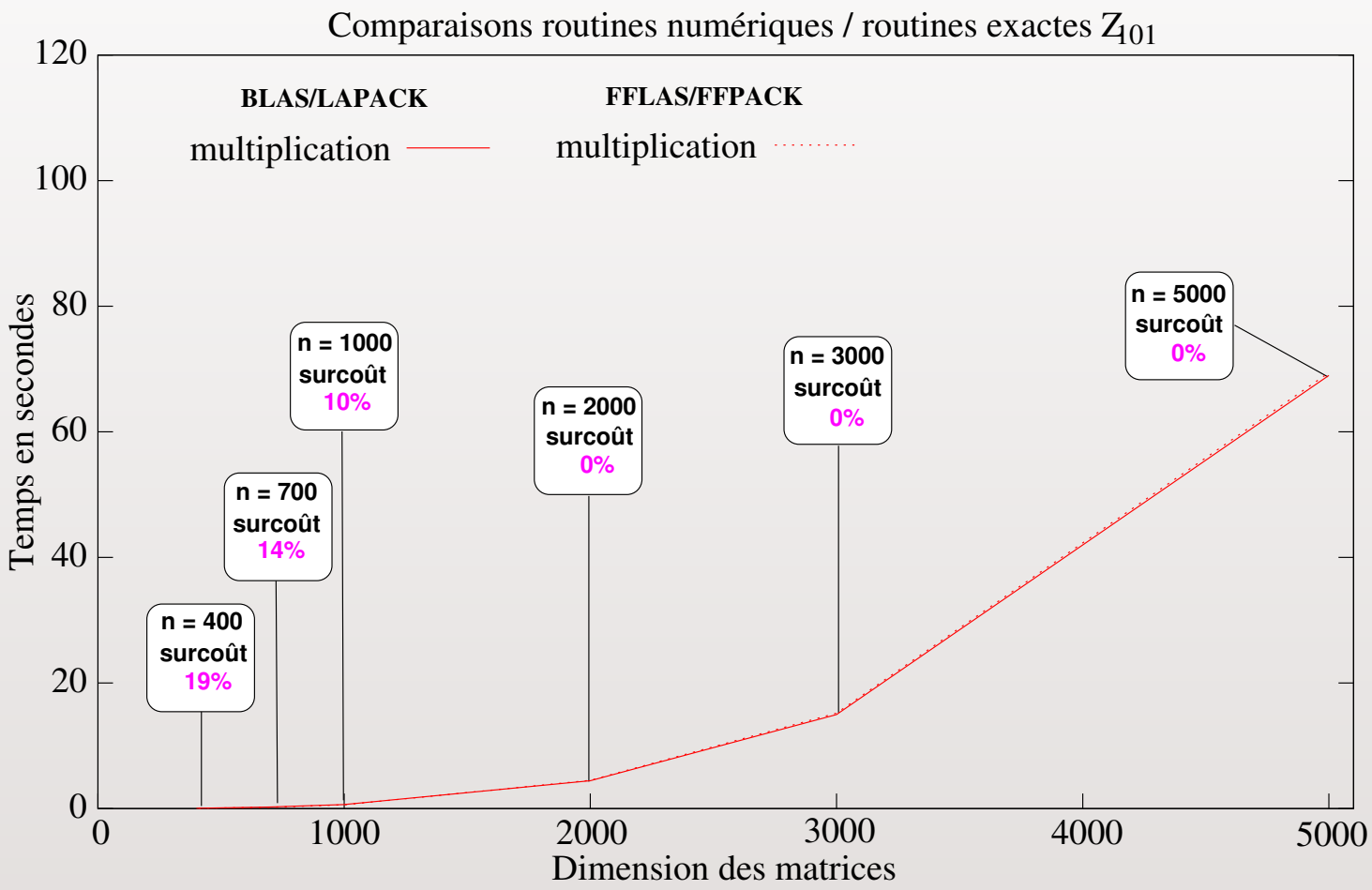
- interface LINBOX pour la résolution
- performances pour des systèmes denses.

Conclusion et perspectives

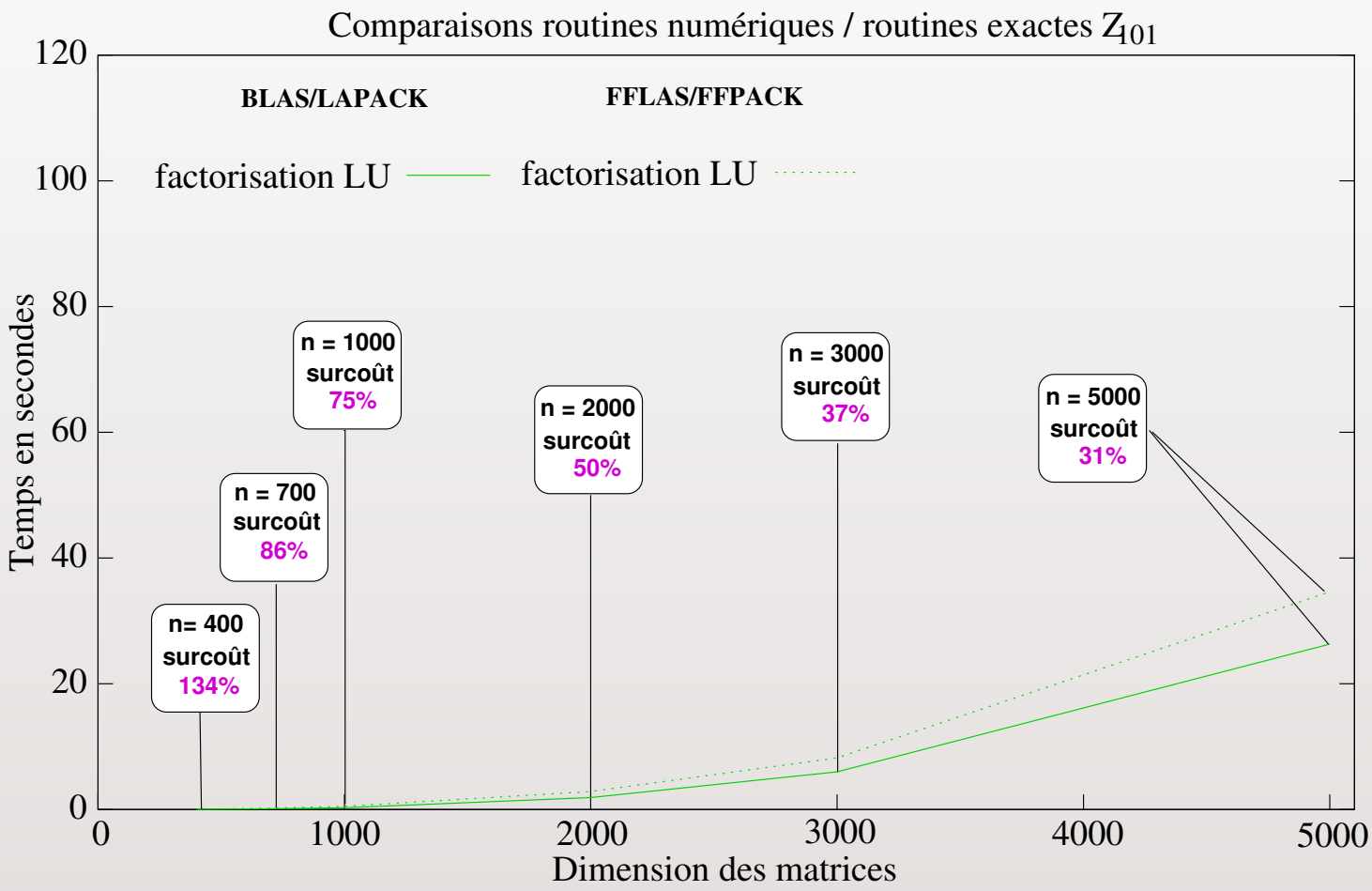
Performances (Pentium Xeon-2.66 Ghz, 1Go RAM)



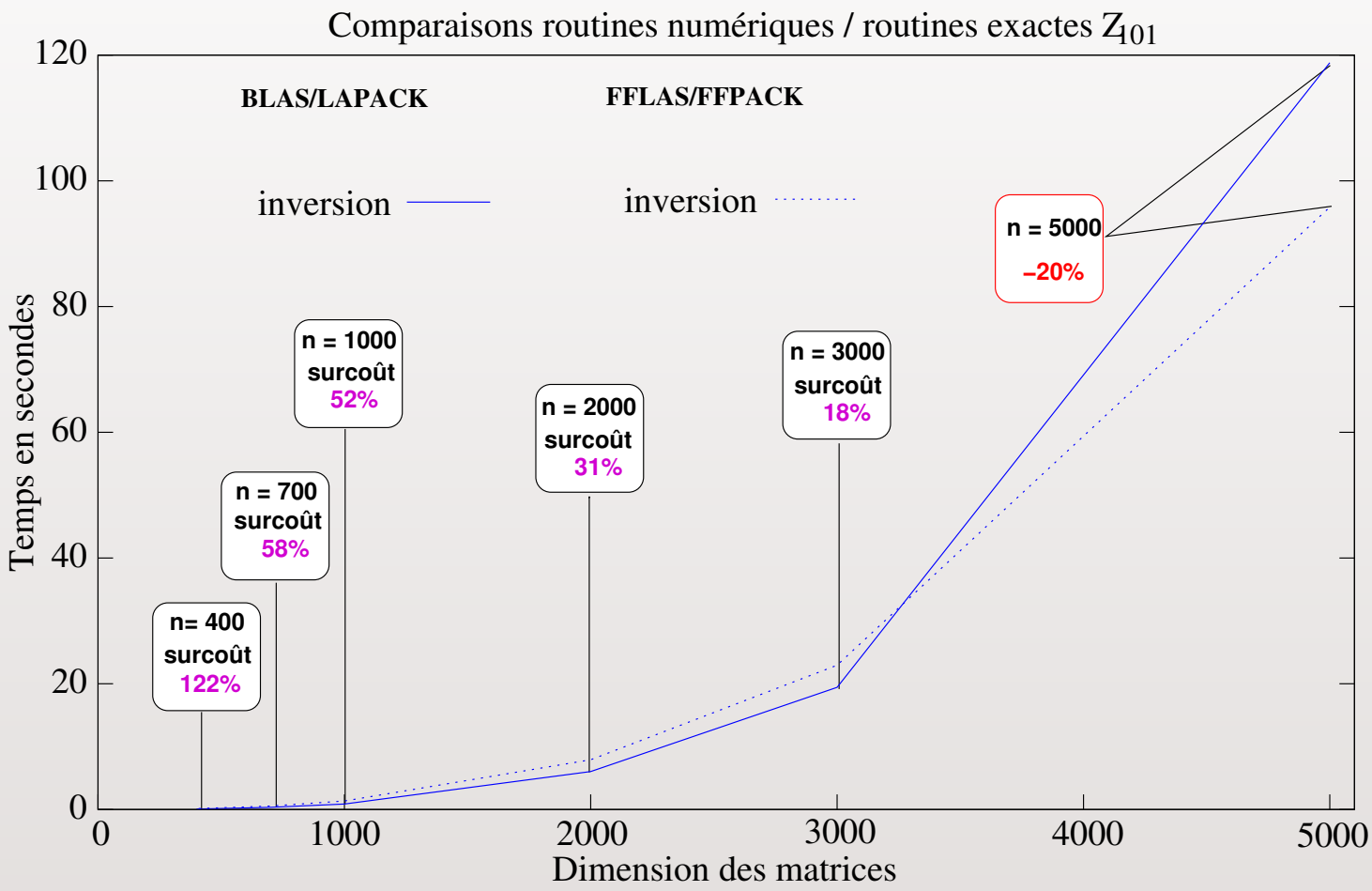
Performances (Pentium Xeon-2.66 Ghz, 1Go RAM)



Performances (Pentium Xeon-2.66 Ghz, 1Go RAM)



Performances (Pentium Xeon-2.66 Ghz, 1Go RAM)



Comportement pratique des réductions théoriques

- coût théorique LQUP/produit matrices ($\omega = 3$)$\Rightarrow$ 1/3

n	*400*	*700*	*1000*	*2000*	*3000*	*5000*
`LQUP`	0.04s	0.20s	0.49s	2.85s	8.22s	34.13s
`fgemm`	0.04s	0.24s	0.66s	4.44s	14.96s	69.19s
Ratio	**1**	**0.83**	**0.74**	**0.64**	**0.55**	**0.49**

- coût théorique inversion/produit matrices ($\omega = 3$) $\Rightarrow$ 1

n	*400*	*700*	*1000*	*2000*	*3000*	*5000*
`inv.`	0.14s	0.53s	1.34s	7.93s	22.94s	95.19s
`fgemm`	0.04s	0.24s	0.66s	4.44s	14.96s	69.19s
Ratio	**3.5**	**2.2**	**2.03**	**1.78**	**1.53**	**1.37**

Les ratios pratiques tendent vers les ratios théoriques.

Comportement pratique des réductions théoriques

- coût théorique LQUP/produit matrices ($\omega = 3$)$\Rightarrow$ 1/3

n	*400*	*700*	*1000*	*2000*	*3000*	*5000*
`LQUP`	0.04s	0.20s	0.49s	2.85s	8.22s	34.13s
`fgemm`	0.04s	0.24s	0.66s	4.44s	14.96s	69.19s
Ratio	**1**	**0.83**	**0.74**	**0.64**	**0.55**	**0.49**

- coût théorique inversion/produit matrices ($\omega = 3$) $\Rightarrow$ 1

n	*400*	*700*	*1000*	*2000*	*3000*	*5000*
`inv.`	0.14s	0.53s	1.34s	7.93s	22.94s	95.19s
`fgemm`	0.04s	0.24s	0.66s	4.44s	14.96s	69.19s
Ratio	**3.5**	**2.2**	**2.03**	**1.78**	**1.53**	**1.37**

Les ratios pratiques tendent vers les ratios théoriques.

LINBOX : calcul de hautes performances...

Projet LINBOX

Implantation générique des corps finis
- modèle de base et interface
- un exemple de *"plug-in"*

Algèbre linéaire dense sur un corps fini
- réduction au produit de matrices
- paquetages FFLAS-FFPACK
- performances

Systèmes linéaires diophantiens
- interface LINBOX pour la résolution
- performances pour des systèmes denses.

Conclusion et perspectives

Motivations

Problème représentatif de l'algèbre linéaire exacte :

- plusieurs niveaux de calcul (corps finis, entiers, rationnels),
- plusieurs méthodes suivant le type du système (dense, structuré, creux),
- gestion de la singularité des matrices, test d'inconsistance,
- approches probabilistes et nombreuses heuristiques de calcul.

Motivations

Problème représentatif de l'algèbre linéaire exacte :

- plusieurs niveaux de calcul (corps finis, entiers, rationnels),
- plusieurs méthodes suivant le type du système (dense, structuré, creux),
- gestion de la singularité des matrices, test d'inconsistance,
- approches probabilistes et nombreuses heuristiques de calcul.

$\Rightarrow$ Validation des briques de base développées dans LINBOX.

Problème

Étant donnés $A \in \mathbb{Z}^{m \times n}$, $b \in \mathbb{Z}^m$.

- trouver $x \in \mathbb{Z}^n$ tel que $Ax = b$,
- certifier qu'il n'existe pas de solution diophantienne.

- Approche classique : formes normales (Smith ou Hermite)

Problème

Étant donnés $A \in \mathbb{Z}^{m \times n}$, $b \in \mathbb{Z}^m$.

- trouver $x \in \mathbb{Z}^n$ tel que $Ax = b$,
- certifier qu'il n'existe pas de solution diophantienne.

- Approche classique : formes normales (Smith ou Hermite)
  pas satisfaisante $\approx n^4$ opérations binaires.

solutions rationnelles en $\approx n^3 \log n$ [Dixon 1982, Wiedemann 1986]

Problème

Étant donnés $A \in \mathbb{Z}^{m \times n}$, $b \in \mathbb{Z}^m$.

- trouver $x \in \mathbb{Z}^n$ tel que $Ax = b$,
- certifier qu'il n'existe pas de solution diophantienne.

• Approche classique : formes normales (Smith ou Hermite)
pas satisfaisante $\approx n^4$ opérations binaires.

solutions rationnelles en $\approx n^3 \log n$ [Dixon 1982, Wiedemann 1986]

• Approche probabiliste $\approx (n^3 \log n) \times \log n$ [Giesbrecht 1997]
$\Rightarrow$ combiner des solutions rationnelles

Problème

Étant donnés $A \in \mathbb{Z}^{m \times n}$, $b \in \mathbb{Z}^m$.

- trouver $x \in \mathbb{Z}^n$ tel que $Ax = b$,
- certifier qu'il n'existe pas de solution diophantienne.

• Approche classique : formes normales (Smith ou Hermite)
pas satisfaisante $\approx n^4$ opérations binaires.

solutions rationnelles en $\approx n^3 \log n$ [Dixon 1982, Wiedemann 1986]

• Approche probabiliste $\approx (n^3 \log n) \times \log n$ [Giesbrecht 1997]
$\Rightarrow$ combiner des solutions rationnelles

De nombreuses études algorithmiques :
[Giesbrecht, Lobo, Mulders, Saunders, Storjohann 1997-2004]
$\hookrightarrow$ systèmes creux/denses, inconsistance sur $\mathbb{Z}$, solution minimale

Combinaisons des solutions rationnelles

$$A = \begin{bmatrix} 11 & 13 & 4 \\ 5 & 7 & 9 \end{bmatrix}, \; b = \begin{bmatrix} 7 \\ 10 \end{bmatrix}.$$

Combinaisons des solutions rationnelles

$$A = \begin{bmatrix} 11 & 13 & 4 \\ 5 & 7 & 9 \end{bmatrix}, \; b = \begin{bmatrix} 7 \\ 10 \end{bmatrix}.$$

$$y_1 = \begin{bmatrix} -27/4 \\ 25/4 \\ 0 \end{bmatrix}, \; y_2 = \begin{bmatrix} 2/3 \\ -1/3 \\ 1 \end{bmatrix} \text{ deux solutions rationnelles.}$$

Combinaisons des solutions rationnelles

$$A = \begin{bmatrix} 11 & 13 & 4 \\ 5 & 7 & 9 \end{bmatrix},\ b = \begin{bmatrix} 7 \\ 10 \end{bmatrix}.$$

$$y_1 = \begin{bmatrix} -27/4 \\ 25/4 \\ 0 \end{bmatrix},\ y_2 = \begin{bmatrix} 2/3 \\ -1/3 \\ 1 \end{bmatrix}$$ deux solutions rationnelles.

$$x = 4y_1 - 3y_2 = \begin{bmatrix} -29 \\ 26 \\ -3 \end{bmatrix}$$ est une solution diophantienne.

Combinaisons des solutions rationnelles

$$A = \begin{bmatrix} 11 & 13 & 4 \\ 5 & 7 & 9 \end{bmatrix},\ b = \begin{bmatrix} 7 \\ 10 \end{bmatrix}.$$

$$y_1 = \begin{bmatrix} -27/4 \\ 25/4 \\ 0 \end{bmatrix},\ y_2 = \begin{bmatrix} 2/3 \\ -1/3 \\ 1 \end{bmatrix} \text{ deux solutions rationnelles.}$$

$$x = 4y_1 - 3y_2 = \begin{bmatrix} -29 \\ 26 \\ -3 \end{bmatrix} \text{ est une solution diophantienne.}$$

Utilisation de pgcd sur les dénominateurs des solutions.

Soient $y_1 = \frac{\bar{y}_1}{d_1}$, $y_2 = \frac{\bar{y}_2}{d_2}$ *avec* $\bar{y}_1, \bar{y}_2 \in \mathbb{Z}^n$ *et* $d_1, d_2 \in \mathbb{Z}$.
Soit $g = pgcd(d_1, d_2) = sd_1 + td_2$.
Alors
$y = \frac{sd_1\bar{y}_1 + td_2\bar{y}_2}{g}$ *est une solution avec un dénominateur* $\leq g$.

Combinaisons des solutions rationnelles

$$A = \begin{bmatrix} 11 & 13 & 4 \\ 5 & 7 & 9 \end{bmatrix},\ b = \begin{bmatrix} 7 \\ 10 \end{bmatrix}.$$

$$y_1 = \begin{bmatrix} -27/4 \\ 25/4 \\ 0 \end{bmatrix},\ y_2 = \begin{bmatrix} 2/3 \\ -1/3 \\ 1 \end{bmatrix} \text{ deux solutions rationnelles.}$$

$$x = 4y_1 - 3y_2 = \begin{bmatrix} -29 \\ 26 \\ -3 \end{bmatrix} \text{ est une solution diophantienne.}$$

Utilisation de pgcd sur les dénominateurs des solutions.

Soient $y_1 = \frac{\bar{y}_1}{d_1}$, $y_2 = \frac{\bar{y}_2}{d_2}$ *avec* $\bar{y}_1, \bar{y}_2 \in \mathbb{Z}^n$ *et* $d_1, d_2 \in \mathbb{Z}$.
Soit $g = pgcd(d_1, d_2) = sd_1 + td_2$.
Alors
$y = \frac{sd_1\bar{y}_1 + td_2\bar{y}_2}{g}$ *est une solution avec un dénominateur* $\leq g$.

$\longrightarrow$ nécessite des solutions avec des dénominateurs différents

Dénominateurs et préconditionneurs

Les dénominateurs dépendent du mineur utilisé pour résoudre le système (on se ramène à des systèmes non singuliers).

Dénominateurs et préconditionneurs

Les dénominateurs dépendent du mineur utilisé pour résoudre le système (on se ramène à des systèmes non singuliers).

Utilisation de préconditionneurs aléatoires
$\Rightarrow$ changement de sous matrice pour déterminer le mineur

Étant données P, Q deux matrices à coefficients entiers aléatoires

$$PAQx = Pb \longrightarrow Ay = b \text{ avec } y = Qx$$

Richesse des préconditionneurs [Chen et al. 2002] :
approche développée par les membres du projet LINBOX.

$\longrightarrow$ application à un vecteur $\approx n \log n$

Toeplitz, réseaux de permutation, diagonales, creuses

Projet LINBOX

Implantation générique des corps finis

- modèle de base et interface
- un exemple de *"plug-in"*

Algèbre linéaire dense sur un corps fini

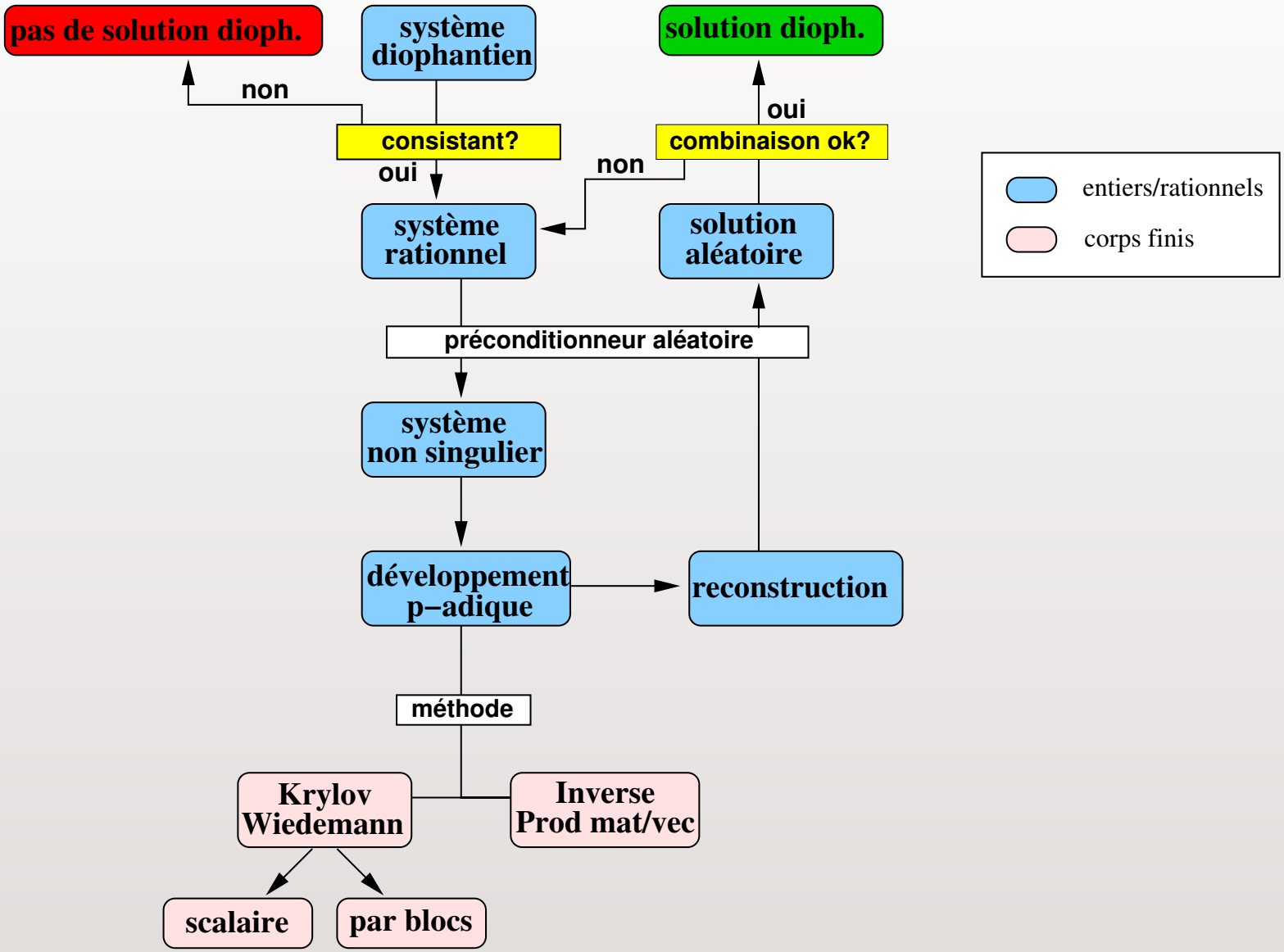
- réduction au produit de matrices
- paquetages FFLAS-FFPACK
- performances

Systèmes linéaires diophantiens

- interface LINBOX pour la résolution
- performances pour des systèmes denses.

Conclusion et perspectives

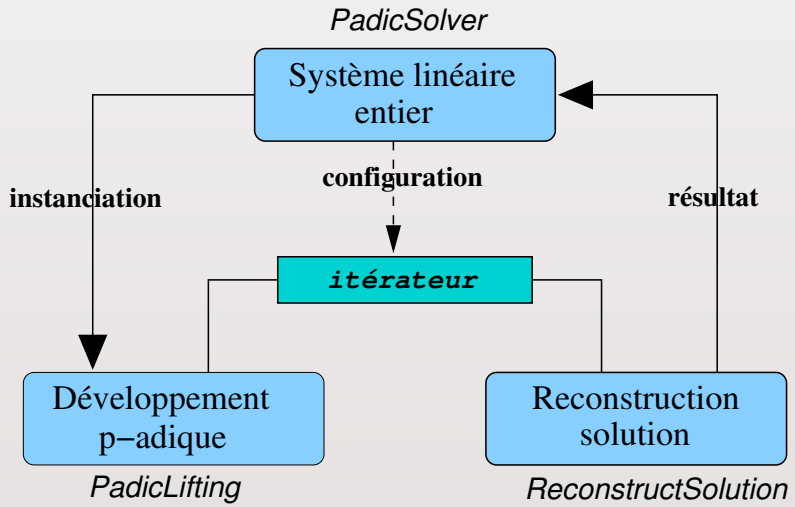
Notre contribution



Interface générique

Fournir une interface de calcul générique :

- gestion de la singularité, des préconditionneurs,
- abstraction de la méthode de calcul des chiffres p-adiques,
- stratégies algorithmiques (Monte Carlo, Las Vegas).



calcul des solutions rationnelles

Opérations clés :

- préconditionnements :
  produit matrice-vecteur (creux, structuré), produit de matrices (dense)
- calcul d'une sous matrice maximale inversible :
  calcul du rang probabiliste (modulo p)
- développement p-adique [Dixon 1982]
  $A^{-1}b \mod p^k$ $(k \approx n \log n)$
- reconstruction de la solution rationnelle [Wang 1981]
  $x = A^{-1}b \in \mathbb{Q}^n$ à partir de $A^{-1}b \mod p^k$.

Développement *p*-adique et reconstruction

Calcul itératif des chiffres *p*-adiques par correction du résidu.
$\longrightarrow$ système linéaire modulo *p*, produit matrice-vecteur

nos apports :

- calcul hybride "exact/numérique" BLAS,
- représentation *q*-adique des matrices,
- résolution itérative Krylov/Wiedemann par blocs (Padé matriciel[2])

[2] P. Giorgi, C-.P. Jeannerod and G. Villard. On the complexity of polynomial matrix computations. In *Proceedings of the 2003 International Symposium on Symbolic and Algebraic Computation*, ACM Press New York.

Développement p-adique et reconstruction

Calcul itératif des chiffres p-adiques par correction du résidu.
$\longrightarrow$ système linéaire modulo p, produit matrice-vecteur

nos apports :

- calcul hybride "exact/numérique" BLAS,
- représentation q-adique des matrices,
- résolution itérative Krylov/Wiedemann par blocs (Padé matriciel[2])

Construction du développement à partir des chiffres p-adiques.
$\longrightarrow$ évaluation de polynômes en p

nos apports :

- Horner, "pas de bébé/pas de géant", "diviser pour régner".

[2] P. Giorgi, C-.P. Jeannerod and G. Villard. On the complexity of polynomial matrix computations. In *Proceedings of the 2003 International Symposium on Symbolic and Algebraic Computation*, ACM Press New York.

Développement p-adique et reconstruction

Calcul itératif des chiffres p-adiques par correction du résidu.
$\longrightarrow$ système linéaire modulo p, produit matrice-vecteur

nos apports :

- calcul hybride "exact/numérique" BLAS,
- représentation q-adique des matrices,
- résolution itérative Krylov/Wiedemann par blocs (Padé matriciel[2])

Construction du développement à partir des chiffres p-adiques.
$\longrightarrow$ évaluation de polynômes en p

nos apports :

- Horner, "pas de bébé/pas de géant", "diviser pour régner".

Reconstruction de la solution (Euclide étendu).

nos apports :

- Heuristique : Euclide sur une seule composante, produits modulaires.

[2] P. Giorgi, C-.P. Jeannerod and G. Villard. On the complexity of polynomial matrix computations. In *Proceedings of the 2003 International Symposium on Symbolic and Algebraic Computation*, ACM Press New York.

Projet LINBOX

Implantation générique des corps finis

- modèle de base et interface
- un exemple de *"plug-in"*

Algèbre linéaire dense sur un corps fini

- réduction au produit de matrices
- paquetages FFLAS-FFPACK
- performances

Systèmes linéaires diophantiens

- interface LINBOX pour la résolution
- performances pour des systèmes denses.

Conclusion et perspectives

Rapport solution diophantiennes/solutions rationnelles

Systèmes linéaires entiers aléatoires :

Pentium Xeon-2.66Ghz, 1Go RAM, 1Go swap

	systèmes linéaires entiers à coeff. sur 3 bits		
	sol. rationnelle	sol. diophantienne	dioph./ratio.
400×800	3.82s	5.72s	1.49
800×1000	18.15s	27.49s	1.51
1200×1400	51.97s	78.57s	1.51
1200×2000	58.73s	86.17s	1.46

Deux solutions rationnelles $\Rightarrow$ solution diophantienne

Pas de préconditionnement pour la première solution (meilleure performance)

Systèmes diophantiens en pratique...

Comparaison avec la bibliothèque IML [Chen, Storjohann 2004]
$\Rightarrow$ heuristique différente (blocs de solutions)

Pentium Xeon-2.66Ghz, 1Go RAM, 1Go swap

	systèmes linéaires diophantiens			
	coeff. 3 bits		coeff. 100 bits	
	LINBOX	IML	LINBOX	IML
400×800	5.72s	7.18s	95.23s	252.06s
600×1000	14.42s	20.59s	(3) 402.92s	675.96s
800×1000	27.49s	42.49s	(3) 886.98s	1387.24s
1200×1400	78.57s	124.79s	-	-
1200×2000	86.17s	128.66s	-	-

(..) $\rightarrow$ nb. combinaisons si $\neq 2$
\- $\rightarrow$ mémoire insuffisante

notre heuristique est plus adaptée lorsque peu de solutions rationnelles sont nécessaires.

Remarque : la comparaison théorique/pratique du nombre d'itérations est à compléter.

Projet LINBOX

Implantation générique des corps finis
modèle de base et interface
un exemple de *"plug-in"*

Algèbre linéaire dense sur un corps fini
réduction au produit de matrices
paquetages FFLAS-FFPACK
performances

Systèmes linéaires diophantiens
interface LINBOX pour la résolution
performances pour des systèmes denses.

Conclusion et perspectives

Nos contributions au calcul formel

- LINBOX : une bibliothèque d'algèbre linéaire exacte de hautes performances.
  - interface dédiée aux non experts,
  - généricité des implantations (mécanisme de "plug-in"),
  - large choix d'algorithmes et de solutions,
  - réutilisation de bibliothèques (BLAS, GMP, ...).

  pas loin de 20.000 lignes de codes...

- Interaction avec d'autres logiciels (e.g. MAPLE).

- Un nouvel algorithme pour les approximants de matrices polynomiales (réduction au produit de matrices polynomiales).

- Une implantation d'heuristique efficace pour la résolution de systèmes linéaires diophantiens.

Extension de nos travaux

- Finaliser le cas creux pour la résolution de systèmes linéaires diophantiens.
  $\Rightarrow$ gestion des calculs probabilistes de Krylov/Wiedemann, choix des préconditionneurs.

- Nouvelles approches utilisant des réductions au produit de matrices entières/polynomiales.
  [Storjohann 2002-2004], [Giorgi, Jeannerod, Villard 2003-2004]
  $\Rightarrow$ approche FFLAS-FFPACK au cas entier/polynomial.

- Interaction automatique de LINBOX avec MAPLE : archétypes, protocole d'échange standard de données
  $\Rightarrow$ Postdoc à Waterloo (Canada), janvier 2005.

- Interopération de bibliothèques en calcul formel : projet ROXANE $\longrightarrow$ correspondant pour LINBOX .

Merci de votre attention...