

SOLVING SPARSE RATIONAL LINEAR SYSTEMS

Pascal Giorgi

University of Waterloo (Canada) / University of Perpignan (France)



UPVD
Université de Perpignan Via Domitia

joint work with

A. Storjohann, **M. Giesbrecht** (University of Waterloo),

W. Eberly (University of Calgary), **G. Villard** (ENS Lyon)

ISSAC'2006, Genova - July 11, 2006

Problem

Let A a non-singular matrix and b a vector defined over \mathbb{Z} .

Problem : Compute $x = A^{-1}b$ over the rational numbers

$$A = \begin{pmatrix} 289 & 237 & 79 & -268 \\ 108 & -33 & -211 & 309 \\ -489 & 104 & -24 & -25 \\ 308 & 99 & -108 & 66 \end{pmatrix}, \quad b = \begin{pmatrix} -131 \\ 321 \\ 147 \\ 43 \end{pmatrix}.$$

$$x = A^{-1}b = \begin{pmatrix} \frac{-5795449}{32845073} \\ \frac{152262251}{98535219} \\ \frac{428820914}{229915511} \\ \frac{1523701534}{689746533} \end{pmatrix}$$

Main difficulty : **expression swell**

Problem

Let A a non-singular matrix and b a vector defined over \mathbb{Z} .

Problem : Compute $x = A^{-1}b$ over the rational numbers

$$A = \begin{pmatrix} -289 & 0 & 0 & -268 \\ 0 & -33 & 0 & 0 \\ -489 & 0 & -24 & -25 \\ 0 & 0 & -108 & 66 \end{pmatrix}, \quad b = \begin{pmatrix} -131 \\ 321 \\ 147 \\ 43 \end{pmatrix}.$$

$$x = A^{-1}b = \begin{pmatrix} \frac{-378283}{1282641} \\ \frac{-107}{11} \\ \frac{-4521895}{15391692} \\ \frac{219038}{1282641} \end{pmatrix}$$

Main difficulty : **expression swell** and **take advantage of sparsity**

Motivations

Large linear systems are involved in many mathematical applications

Over a finite field : integers factorization [Odlyzko 1999],
discrete logarithm [Odlyzko 1999 ; Thomé 2003].

Over the integers : number theory [Cohen 1993], group theory [Newman 1972],
integer programming [Aardal, Hurkens, Lenstra 1999].

Rational linear systems are central in recent linear algebra algorithms

- ▶ Determinant [Abbott, Bronstein, Mulders 1999 ; Storjohann 2005]
- ▶ Smith form [Giesbrecht 1995 ; Eberly, Giesbrecht, Villard 2000]
- ▶ Nullspace, Kernel [Chen, Storjohann 2005]

Algorithms for non-singular system solving

Dense matrices :

- ▶ Gaussian elimination and CRA
↪ $\tilde{O}(n^{\omega+1} \log \|A\|)$ bit operations
- ▶ P-adic lifting [Monck, Carter 1979 ; Dixon 1982]
↪ $\tilde{O}(n^3 \log \|A\|)$ bit operations
- ▶ High order lifting [Storjohann 2005]
↪ $\tilde{O}(n^{\omega} \log \|A\|)$ bit operations

Sparse matrices :

- ▶ P-adic lifting or CRA [Kaltofen, Saunders 1991]
↪ $\tilde{O}(\gamma n^2 \log \|A\|)$ bit operations with γ non-zero elts.

P-adic algorithm with matrix inversion

Scheme to compute $A^{-1}b$ [Dixon 1982] :

$$(1-1) \quad B := A^{-1} \bmod p$$

$$(1-2) \quad r := b$$

for $i := 0$ to k

$$(2-1) \quad x_i := B.r \bmod p$$

$$(2-2) \quad r := (1/p)(r - A.x_i)$$

$$(3-1) \quad x := \sum_{i=0}^k x_i \cdot p^i$$

(3-2) *rational reconstruction on x*

P-adic algorithm with matrix inversion

Scheme to compute $A^{-1}b$ [Dixon 1982] :

$$(1-1) \quad B := A^{-1} \bmod p$$

$$(1-2) \quad r := b$$

for $i := 0$ to k

$$(2-1) \quad x_i := B.r \bmod p$$

$$(2-2) \quad r := (1/p)(r - A.x_i)$$

$$(3-1) \quad x := \sum_{i=0}^k x_i \cdot p^i$$

(3-2) *rational reconstruction on x*

$$O^{\sim}(n^3 \log \|A\|)$$

$$k = O^{\sim}(n)$$

$$O^{\sim}(n^2 \log \|A\|)$$

$$O^{\sim}(n^2 \log \|A\|)$$

P-adic algorithm with matrix inversion

Scheme to compute $A^{-1}b$ [Dixon 1982] :

$$(1-1) \quad B := A^{-1} \bmod p$$

$$(1-2) \quad r := b$$

for $i := 0$ to k

$$(2-1) \quad x_i := B.r \bmod p$$

$$(2-2) \quad r := (1/p)(r - A.x_i)$$

$$(3-1) \quad x := \sum_{i=0}^k x_i \cdot p^i$$

(3-2) *rational reconstruction on x*

$$O^{\sim}(n^3 \log \|A\|)$$

$$k = O^{\sim}(n)$$

$$O^{\sim}(n^2 \log \|A\|)$$

$$O^{\sim}(n^2 \log \|A\|)$$

Main operations : **matrix inversion** and **matrix-vector products**

Dense linear system solving in practice

Efficient implementations are available : LinBox 1.0 [www.linalg.org]

- Use tuned BLAS floating-point library for exact arithmetic
 - matrix inversion over prime field [Dumas, Giorgi, Pernet 2004]
 - BLAS matrix-vector product with CRT over the integers
- Rational number reconstruction
 - half GCD [Schönage 1971]
 - heuristic using integer multiplication [NTL library]

random dense linear system with 3 bits entries (P4 - 3.4Ghz)

n	500	1000	2000	3000	4000	5000
Time	0.6s	4.3s	31.1s	99.6s	236.8s	449.2s

performances improvement of a factor 10
over NTL's tuned implementation

What does happen when matrices are sparse ?

We consider sparse matrices with $O(n)$ non zero elements
↔ matrix-vector product needs only $O(n)$ operations.

Sparse linear system and P-adic lifting

Computing the modular inverse is proscribed due to fill-in

Solution [Kaltofen, Saunders 1991] :

↔ use **modular minimal polynomial** instead of inverse

Sparse linear system and P-adic lifting

Computing the modular inverse is proscribed due to fill-in

Solution [Kaltofen, Saunders 1991] :

↪ use **modular minimal polynomial** instead of inverse

Wiedemann approach (1986)

Let $A \in \mathbb{F}^{n \times n}$ of full rank and $b \in \mathbb{F}^n$. Then $x = A^{-1}b$ can be expressed as a linear combination of the Krylov subspace $\{b, Ab, \dots, A^n b\}$

Let $f^A(\lambda) = f_0 + f_1\lambda + \dots + f_d\lambda^d \in \mathbb{F}[\lambda]$ be the minimal polynomial of A

Sparse linear system and P-adic lifting

Computing the modular inverse is proscribed due to fill-in

Solution [Kaltofen, Saunders 1991] :

↪ use **modular minimal polynomial** instead of inverse

Wiedemann approach (1986)

Let $A \in \mathbb{F}^{n \times n}$ of full rank and $b \in \mathbb{F}^n$. Then $x = A^{-1}b$ can be expressed as a linear combination of the Krylov subspace $\{b, Ab, \dots, A^n b\}$

Let $f^A(\lambda) = f_0 + f_1\lambda + \dots + f_d\lambda^d \in \mathbb{F}[\lambda]$ be the minimal polynomial of A

$$A^{-1}b = \frac{-1}{f_0}(f_1b + f_2Ab + \dots + f_dA^{d-1}b)$$

Sparse linear system and P-adic lifting

Computing the modular inverse is proscribed due to fill-in

Solution [Kaltofen, Saunders 1991] :

↪ use **modular minimal polynomial** instead of inverse

Wiedemann approach (1986)

Let $A \in \mathbb{F}^{n \times n}$ of full rank and $b \in \mathbb{F}^n$. Then $x = A^{-1}b$ can be expressed as a linear combination of the Krylov subspace $\{b, Ab, \dots, A^{n-1}b\}$

Let $f^A(\lambda) = f_0 + f_1\lambda + \dots + f_d\lambda^d \in \mathbb{F}[\lambda]$ be the minimal polynomial of A

$$A^{-1}b = \underbrace{\frac{-1}{f_0}(f_1b + f_2Ab + \dots + f_dA^{d-1}b)}_x$$

Applying minpoly in each lifting steps cost $\tilde{O}(nd)$ field operations, then giving a worst case complexity of $\tilde{O}(n^3 \log \|A\|)$ bit operations.

Sparse linear system solving in practice

use of LinBox library on Itanium II - 1.3Ghz, 128Gb RAM

- random systems with 3 bits entries and 10 elts/row (plus identity)

	system order				
	<i>400</i>	<i>900</i>	<i>1600</i>	<i>2500</i>	<i>3600</i>
Maple	64.7s	849s	11098s	—	—
CRA-Wied	14.8s	168s	1017s	3857s	11452s
P-adic-Wied	10.2s	113s	693s	2629s	8034s
Dixon	0.9s	10s	42s	178s	429s

Sparse linear system solving in practice

use of LinBox library on Itanium II - 1.3Ghz, 128Gb RAM

- random systems with 3 bits entries and 10 elts/row (plus identity)

	system order				
	400	900	1600	2500	3600
Maple	64.7s	849s	11098s	—	—
CRA-Wied	14.8s	168s	1017s	3857s	11452s
P-adic-Wied	10.2s	113s	693s	2629s	8034s
Dixon	0.9s	10s	42s	178s	429s

main difference :

$$(2-1) \quad x_i = B.r \bmod p \quad (\text{Dixon})$$

$$(2-1) \quad x_i := \frac{-1}{f_0} \sum_{i=1}^{\deg f^A} f_i A^{i-1} r \bmod p \quad (\text{P-adic-Wied})$$

Remark :

n sparse matrix applications is far from level 2 BLAS in practice.

Our objectives

In practice :

Integrate level 2 and 3 BLAS in integer sparse solver

In theory :

Improve bit complexity of sparse linear system solving

$\Rightarrow O^{\sim}(n^{\delta})$ bits operations with $\delta < 3$?

Our alternative to Block Wiedemann

Express the inverse of the sparse matrix through a structured form
↪ block Hankel/Toeplitz structures

Let $u \in \mathbb{F}^{s \times n}$ and $v \in \mathbb{F}^{n \times s}$ s.t. following matrices are non-singular

$$U = \begin{pmatrix} u \\ uA \\ \vdots \\ uA^{m-1} \end{pmatrix}, V = \begin{pmatrix} v & Av & \dots & A^{m-1}v \end{pmatrix} \in \mathbb{F}^{n \times n}$$

Our alternative to Block Wiedemann

Express the inverse of the sparse matrix through a structured form
↔ block Hankel/Toeplitz structures

Let $u \in \mathbb{F}^{s \times n}$ and $v \in \mathbb{F}^{n \times s}$ s.t. following matrices are non-singular

$$U = \begin{pmatrix} u \\ uA \\ \vdots \\ uA^{m-1} \end{pmatrix}, \quad V = \begin{pmatrix} v & Av & \dots & A^{m-1}v \end{pmatrix} \in \mathbb{F}^{n \times n}$$

then we can define the block Hankel matrix

$$H = UAV = \begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_m \\ \alpha_2 & \alpha_3 & \dots & \alpha_{m+1} \\ \vdots & & & \\ \alpha_m & \alpha_m & \dots & \alpha_{2m-1} \end{pmatrix}, \quad \alpha_i = uA^i v \in \mathbb{F}^{s \times s}$$

and thus we have $A^{-1} = VH^{-1}U$

Alternative to Block Wiedemann

- Nice property on block Hankel matrix inverse [Gohberg, Krupnik 1972, Labahn, Choi, Cabay 1990]

$$H^{-1} = \underbrace{\begin{pmatrix} * & \dots & * \\ \vdots & \ddots & \vdots \\ * & & \end{pmatrix}}_{H_1} \underbrace{\begin{pmatrix} * & \dots & * \\ & \ddots & \vdots \\ & & * \end{pmatrix}}_{T_1} - \underbrace{\begin{pmatrix} * & \dots & * \\ \vdots & \ddots & \vdots \\ * & & \end{pmatrix}}_{H_2} \underbrace{\begin{pmatrix} * & \dots & * \\ & \ddots & \vdots \\ & & * \end{pmatrix}}_{T_2}$$

where H_1, H_2 are block Hankel matrices and T_1, T_2 are block Toeplitz matrices

Alternative to Block Wiedemann

- Nice property on block Hankel matrix inverse [Gohberg, Krupnik 1972, Labahn, Choi, Cabay 1990]

$$H^{-1} = \underbrace{\begin{pmatrix} * & \dots & * \\ \vdots & \ddots & \vdots \\ * & & \end{pmatrix}}_{H_1} \underbrace{\begin{pmatrix} * & \dots & * \\ & \ddots & \vdots \\ & & * \end{pmatrix}}_{T_1} - \underbrace{\begin{pmatrix} * & \dots & * \\ \vdots & \ddots & \vdots \\ * & & \end{pmatrix}}_{H_2} \underbrace{\begin{pmatrix} * & \dots & * \\ & \ddots & \vdots \\ & & * \end{pmatrix}}_{T_2}$$

where H_1, H_2 are block Hankel matrices and T_1, T_2 are block Toeplitz matrices

- Block coefficients in H_1, H_2, T_1, T_2 come from Hermite Padé approximants of $H(z) = \alpha_1 + \alpha_2 z + \dots + \alpha_{2m-1} z^{2m-2}$ [Labahn, Choi, Cabay 1990].
- Complexity of H^{-1} reduces to polynomial matrix multiplication [Giorgi, Jeannerod, Villard 2003].

Alternative to Block Wiedemann

Scheme to compute $A^{-1}b$:

$$(1-1) \quad H(z) := \sum_{i=1}^{2m-1} uA^i v \cdot z^{i-1} \pmod{p}$$

(1-2) compute $H^{-1} \pmod{p}$ from $H(z)$

(1-3) $r := b$

for $i := 0$ to k

$$(2-1) \quad x_i := VH^{-1}U \cdot r \pmod{p}$$

$$(2-2) \quad r := (1/p)(r - A \cdot x_i)$$

$$(3-1) \quad x := \sum_{i=0}^k x_i \cdot p^i$$

(3-2) *rational reconstruction on x*

Alternative to Block Wiedemann

Scheme to compute $A^{-1}b$:

$$(1-1) \quad H(z) := \sum_{i=1}^{2m-1} uA^i v \cdot z^{i-1} \pmod{p}$$

$$O\tilde{~}(sn^2 \log \|A\|)$$

(1-2) compute $H^{-1} \pmod{p}$ from $H(z)$

$$O\tilde{~}(s^2 n \log \|A\|)$$

(1-3) $r := b$

for $i := 0$ to k

$$k = O\tilde{~}(n)$$

(2-1) $x_i := VH^{-1}U \cdot r \pmod{p}$

$$O\tilde{~}((n^2 + sn) \log \|A\|)$$

(2-2) $r := (1/p)(r - A \cdot x_i)$

$$O\tilde{~}(n \log \|A\|)$$

(3-1) $x := \sum_{i=0}^k x_i \cdot p^i$

(3-2) *rational reconstruction on x*

Alternative to Block Wiedemann

Scheme to compute $A^{-1}b$:

$$(1-1) \quad H(z) := \sum_{i=1}^{2m-1} uA^i v \cdot z^{i-1} \bmod p$$

$$O^{\sim}(sn^2 \log \|A\|)$$

(1-2) compute $H^{-1} \bmod p$ from $H(z)$

$$O^{\sim}(s^2 n \log \|A\|)$$

(1-3) $r := b$

for $i := 0$ to k

$$k = O^{\sim}(n)$$

(2-1) $x_i := V H^{-1} U \cdot r \bmod p$

$$O^{\sim}((n^2 + sn) \log \|A\|)$$

(2-2) $r := (1/p)(r - A \cdot x_i)$

$$O^{\sim}(n \log \|A\|)$$

(3-1) $x := \sum_{i=0}^k x_i \cdot p^i$

(3-2) *rational reconstruction on x*

Not yet satisfying : applying matrices U and V is too costly

Applying block Krylov subspaces

$$V = \left(v \mid Av \mid \dots \mid A^{m-1}v \right) \in \mathbb{F}^{n \times n} \text{ and } v \in \mathbb{F}^{n \times s}$$

can be rewrite as

$$V = \left(v \mid \quad \quad \quad \right) + A \left(\quad \quad \quad \mid v \quad \quad \quad \right) + \dots + A^{m-1} \left(\quad \quad \quad \mid \quad \quad \quad \mid v \quad \quad \quad \right)$$

Therefore, applying V to a vector corresponds to :

- $m - 1$ linear combinations of columns of v
- $m - 1$ applications of A

Applying block Krylov subspaces

$$V = \left(v \mid Av \mid \dots \mid A^{m-1}v \right) \in \mathbb{F}^{n \times n} \text{ and } v \in \mathbb{F}^{n \times s}$$

can be rewrite as

$$V = \left(v \mid \quad \quad \quad \right) + A \left(\quad \quad \quad \mid v \quad \quad \quad \right) + \dots + A^{m-1} \left(\quad \quad \quad \mid \quad \quad \quad \mid v \quad \quad \quad \right)$$

Therefore, applying V to a vector corresponds to :

- $m - 1$ linear combinations of columns of v $O(m \times sn \log \|A\|)$
- $m - 1$ applications of A $O(mn \log \|A\|)$

Applying block Krylov subspaces

$$V = \left(v \mid Av \mid \dots \mid A^{m-1}v \right) \in \mathbb{F}^{n \times n} \text{ and } v \in \mathbb{F}^{n \times s}$$

can be rewrite as

$$V = \left(v \mid \quad \quad \quad \right) + A \left(\quad \quad \quad \mid v \quad \quad \quad \right) + \dots + A^{m-1} \left(\quad \quad \quad \mid \quad \quad \quad \mid v \quad \quad \quad \right)$$

Therefore, applying V to a vector corresponds to :

- $m - 1$ linear combinations of columns of v $O(m \times sn \log \|A\|)$
- $m - 1$ applications of A

How to improve the complexity ?

Applying block Krylov subspaces

$$V = \left(v \mid Av \mid \dots \mid A^{m-1}v \right) \in \mathbb{F}^{n \times n} \text{ and } v \in \mathbb{F}^{n \times s}$$

can be rewrite as

$$V = \left(v \mid \quad \quad \quad \right) + A \left(\quad \quad \quad \mid v \quad \quad \quad \right) + \dots + A^{m-1} \left(\quad \quad \quad \mid \quad \quad \quad \mid v \quad \quad \quad \right)$$

Therefore, applying V to a vector corresponds to :

- $m - 1$ linear combinations of columns of v $O(m \times sn \log \|A\|)$
- $m - 1$ applications of A

How to improve the complexity ?

⇒ using special block projections u and v

Candidates as suitable block projections

Considering $A \in \mathbb{F}^{n \times n}$ non-singular and $n = m \times s$.

Let us denote $\mathcal{K}(A, v) := [v \mid Av \mid \cdots \mid A^{m-1}v] \in \mathbb{F}^{n \times n}$

Conjecture :

for any non-singular $A \in \mathbb{F}^{n \times n}$ and $s|n$ there exists a suitable block projection $(R, u, v) \in \mathbb{F}^{n \times n} \times \mathbb{F}^{s \times n} \times \mathbb{F}^{n \times s}$

such that :

1. $\mathcal{K}(RA, v)$ and $\mathcal{K}((RA)^T, u^T)$ are non-singular,
2. R can be applied to a vector with $O^\sim(n)$ operations,
3. u, u^T, v and v^T can be applied to a vector with $O^\sim(n)$ operations.

A structured block projection

Let v be defined as follow

$$v^T = \begin{pmatrix} v_1 & \dots & v_m & & & & & & & \\ & & & v_{m+1} & \dots & v_{2m} & & & & \\ & & & & & & \ddots & & & \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & v_{n-m+1} & \dots & v_n \end{pmatrix} \in \mathbb{F}^{s \times n}$$

where v_i 's are chosen randomly from a sufficient large set.

A structured block projection

Let v be defined as follow

$$v^T = \begin{pmatrix} v_1 & \cdots & v_m & & & & & & & \\ & & & v_{m+1} & \cdots & v_{2m} & & & & \\ & & & & & & \ddots & & & \\ & & & & & & & & & v_{n-m+1} \cdots v_n \end{pmatrix} \in \mathbb{F}^{s \times n}$$

where v_i 's are chosen randomly from a sufficient large set.

open question : Let R diagonal and v as defined above,

is $\mathcal{K}(RA, v)$ necessarily non-singular?

We proved it for case $s = 2$ but answer is still unknown for $s > 2$

Our new algorithm

Scheme to compute $A^{-1}b$:

(1-1) choose structured blocks u and v

(1-2) choose R and $A := R.A$, $b := R.b$

(1-3) $H(z) := \sum_{i=1}^{2m-1} uA^i v \cdot z^{i-1} \pmod{p}$

(1-4) compute $H^{-1} \pmod{p}$ from $H(z)$

(1-5) $r := b$

for $i := 0$ to k

(2-1) $x_i := V H^{-1} U \cdot r \pmod{p}$

(2-2) $r := (1/p)(r - A \cdot x_i)$

(3-1) $x := \sum_{i=0}^k x_i \cdot p^i$

(3-2) *rational reconstruction on x*

Our new algorithm

Scheme to compute $A^{-1}b$:

(1-1) choose structured blocks u and v

(1-2) choose R and $A := R.A$, $b := R.b$

(1-3) $H(z) := \sum_{i=1}^{2m-1} uA^i v \cdot z^{i-1} \bmod p$ $O\tilde{~}(n^2 \log \|A\|)$

(1-4) compute $H^{-1} \bmod p$ from $H(z)$ $O\tilde{~}(s^2 n \log \|A\|)$

(1-5) $r := b$

for $i := 0$ to k

$k = O\tilde{~}(n)$

(2-1) $x_i := V H^{-1} U \cdot r \bmod p$ $O\tilde{~}((mn + sn) \log \|A\|)$

(2-2) $r := (1/p)(r - A \cdot x_i)$ $O\tilde{~}(n \log \|A\|)$

(3-1) $x := \sum_{i=0}^k x_i \cdot p^i$

(3-2) *rational reconstruction on x*

Our new algorithm

Scheme to compute $A^{-1}b$:

(1-1) choose structured blocks u and v

(1-2) choose R and $A := R.A$, $b := R.b$

(1-3) $H(z) := \sum_{i=1}^{2m-1} uA^i v \cdot z^{i-1} \bmod p$

$$\tilde{O}(n^2 \log \|A\|)$$

(1-4) compute $H^{-1} \bmod p$ from $H(z)$

$$\tilde{O}(s^2 n \log \|A\|)$$

(1-5) $r := b$

for $i := 0$ to k

$$k = \tilde{O}(n)$$

(2-1) $x_i := V H^{-1} U \cdot r \bmod p$

$$\tilde{O}((mn + sn) \log \|A\|)$$

(2-2) $r := (1/p)(r - A \cdot x_i)$

$$\tilde{O}(n \log \|A\|)$$

(3-1) $x := \sum_{i=0}^k x_i \cdot p^i$

(3-2) *rational reconstruction on x*

taking the optimal $m = s = \sqrt{n}$ gives a complexity of $\tilde{O}(n^{2.5} \log \|A\|)$

High level implementation

LinBox project (Canada-France-USA) : www.linalg.org

Our tools :

- BLAS-based matrix multiplication and matrix-vector product
- polynomial matrix arithmetic (**block Hankel inversion**)
↪ *FFT, Karatsuba, middle product*
- fast application of H^{-1} is needed to get $\tilde{O}(n^{2.5} \log \|A\|)$

High level implementation

LinBox project (Canada-France-USA) : www.linalg.org

Our tools :

- BLAS-based matrix multiplication and matrix-vector product
- polynomial matrix arithmetic (**block Hankel inversion**)
↪ *FFT, Karatsuba, middle product*
- fast application of H^{-1} is needed to get $\tilde{O}(n^{2.5} \log \|A\|)$
 - ▶ Lagrange's representation of H^{-1} at the beginning (*Horner's scheme*)
 - ▶ use evaluation/interpolation on polynomial vectors
↪ *use Vandermonde matrix to have dense matrix operations*

Is our new algorithm efficient in practice?

Comparing performances

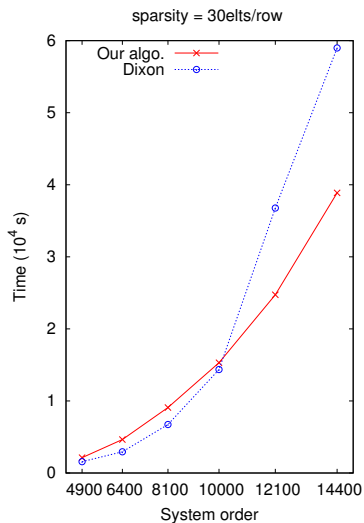
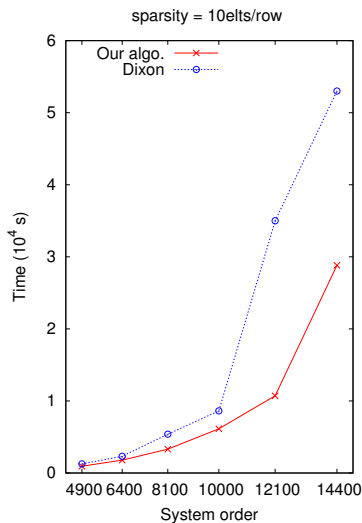
use of LinBox library on Itanium II - 1.3Ghz, 128Gb RAM

- random systems with 3 bits entries and 10 elts/row (plus identity)

	system order					extra memory
	900	1600	2500	3600	4900	
Maple 10	849s	11 098s	—	—	—	$O(1)$
CRA-Wied	168s	1 017s	3 857s	11 452s	$\approx 28\,000s$	$O(n)$
P-adic-Wied	113s	693s	2 629s	8 034s	$\approx 20\,000s$	$O(n)$
Dixon	10s	42s	178s	429s	1 257s	$O(n^2)$
Our algo.	15s	61s	175s	426s	937s	$O(n^{1.5})$

The expected \sqrt{n} improvement is unfortunately amortized by a high constant in the complexity.

Sparse solver vs Dixon's algorithm



Our algorithm performances are depending on matrix sparsity

Practical effect of blocking factors

\sqrt{n} blocking factor value is theoretically optimal

Is this still true in practice?

Practical effect of blocking factors

\sqrt{n} blocking factor value is theoretically optimal

Is this still true in practice?

system order = 10 000, optimal block = 100

block size	80	125	200	400	500
timing	7213s	5264s	4059s	3833s	4332s

system order = 20 000, optimal block \approx 140

block size	125	160	200	500	800
timing	44720s	35967s	30854s	28502s	37318s

Practical effect of blocking factors

\sqrt{n} blocking factor value is theoretically optimal

Is this still true in practice?

system order = 10 000, optimal block = 100

block size	80	125	200	400	500
timing	7213s	5264s	4059s	3833s	4332s

system order = 20 000, optimal block \approx 140

block size	125	160	200	500	800
timing	44720s	35967s	30854s	28502s	37318s

best practical blocking factor is dependent upon the ratio of
sparse matrix/dense matrix operations efficiency

Conclusions

We provide a new approach for solving sparse integer linear systems :

- ▶ improve the complexity by a factor \sqrt{n} (**heuristic**).
- ▶ improve efficiency by minimizing sparse matrix operations and maximizing BLAS use.

drawback : not taking advantage of low degree minimal polynomial

We propose special block projections for sparse linear algebra

↪ inverse of sparse matrix in $O(n^{2.5})$ field op.

Conclusions

We provide a new approach for solving sparse integer linear systems :

- ▶ improve the complexity by a factor \sqrt{n} (**heuristic**).
- ▶ improve efficiency by minimizing sparse matrix operations and maximizing BLAS use.

drawback : not taking advantage of low degree minimal polynomial

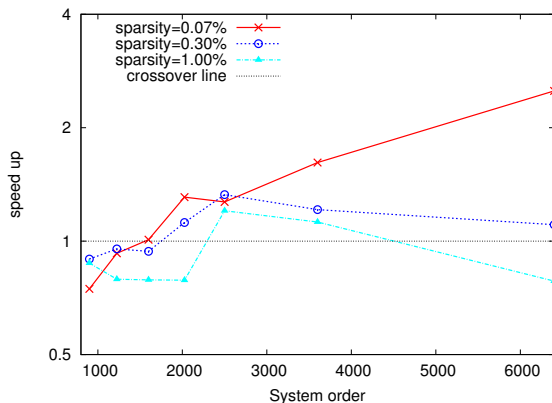
We propose special block projections for sparse linear algebra

↔ inverse of sparse matrix in $O(n^{2.5})$ field op.

Ongoing work :

- ▶ provide an automatic choice of block dimension (non square?)
- ▶ prove conjecture for our structured block projections
- ▶ handle the case of singular matrix
- ▶ introduce fast matrix multiplication in the complexity

Sparse solver vs Dixon's algorithm



The sparser the matrices are, the earlier the crossover appears