Theory and Practice for Solving Sparse Rational Linear Systems

Pascal Giorgi

University of Perpignan, DALI team

joint work with

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

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

séminaire LIRMM - département informatique,

jeudi 8 mars 2007

Problem

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

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

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

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

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

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

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

Main difficulty : expression swell and take advantage of sparsity

Motivations

Large linear systems are involved in many mathematical applications

Over finite fields : 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]

Outline

I. a small guide to rational linear system solving

II. a quest to improve the cost of rational sparse solver

III. what are benefits in practice ?

IV. conclusion and future work

Outline

I. a small guide to rational linear system solving

II. a quest to improve the cost of rational sparse solver

III. what are benefits in practice ?

IV. conclusion and future work

Some notations in this talk

We will use :

- $\tilde{O}(n^{\lambda_1})$ to describe a complexity of $O(n^{\lambda_1} \log^{\lambda_2} n)$ for any $\lambda_2 > 0$.
- ω to refer to the exponent in the algebraic complexity of matrix multiplication $O(n^{\omega})$.
- $||...||$ to refer to the maximal entries in a matrix or vector.
- $\mathbb{F}$ to refer to a field (e.g. finite fields).

Rational solution for non-singular system

Dense matrices :

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

Sparse matrices :

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

Rational solution for non-singular system

Dense matrices :

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

Sparse matrices :

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

Remark [Giesbrecht 1997 ;Mulder, Storjohann 2003]

Diophantine solutions with an extra $\log n$ from rational solutions.

P-adic lifting with matrix inversion

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

$$
\begin{aligned}
&(1\text{-}1) \quad B := A^{-1} \bmod p \\
&(1\text{-}2) \quad r := b \\
&\qquad\quad\ \ \text{for } i := 0 \text{ to } k \\
&(2\text{-}1) \quad\quad x_i := B.r \bmod p \\
&(2\text{-}2) \quad\quad r := (1/p)(r - A.x_i) \\
&(3\text{-}1) \quad x := \textstyle\sum_{i=0}^{k} x_i.p^i \\
&(3\text{-}2) \quad \textit{rational reconstruction on } x
\end{aligned}
$$

P-adic lifting with matrix inversion

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

(1-1)	$B := A^{-1} \bmod p$	$\tilde{O}(n^3 \log \|\|A\|\|)$
(1-2)	$r := b$	
	for $i := 0$ to k	$k = \tilde{O}(n)$
(2-1)	$\quad x_i := B.r \bmod p$	$\tilde{O}(n^2 \log \|\|A\|\|)$
(2-2)	$\quad r := (1/p)(r - A.x_i)$	$\tilde{O}(n^2 \log \|\|A\|\|)$
(3-1)	$x := \sum_{i=0}^{k} x_i.p^i$	
(3-2)	*rational reconstruction on x*	

P-adic lifting with matrix inversion

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

(1-1)	$B := A^{-1} \bmod p$	$O\tilde{}(n^3 \log \|\|A\|\|)$
(1-2)	$r := b$	
	for $i := 0$ to k	$k = O\tilde{}(n)$
(2-1)	$\quad x_i := B.r \bmod p$	$O\tilde{}(n^2 \log \|\|A\|\|)$
(2-2)	$\quad r := (1/p)(r - A.x_i)$	$O\tilde{}(n^2 \log \|\|A\|\|)$
(3-1)	$x := \sum_{i=0}^{k} x_i.p^i$	
(3-2)	*rational reconstruction on x*	

Main operations : matrix inversion and matrix-vector products

Dense linear system solving into practice

Efficient implementations are available : LinBox 1.1 [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]

Dense linear system solving into practice

use of LinBox library on Pentium 4 - 3.4Ghz, 2Gb RAM

random dense linear system with 3 bits entries

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

random dense linear system with 20 bits entries

n	500	1000	2000	3000	4000	5000
time	1.8s	12.9s	91.5s	299.7s	706.4s	MT

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

$\hookrightarrow$ 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] :
$\hookrightarrow$ 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] :
$\hookrightarrow$ use modular minimal polynomial instead of inverse

Wiedemann approach (1986)

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

Let $f^A(\lambda) = f_0 + f_1\lambda + ... + f_d\lambda^d \in \mathbb{F}[\lambda]$ 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 $A \in \mathbb{F}^{n\times n}$ non-singular and $b \in \mathbb{F}^{n}$. Then $x = A^{-1}b$ can be expressed as a linear combination of the Krylov subspace $\{b, Ab, ..., A^n b\}$

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

$$A^{-1}b \quad = \quad \frac{-1}{f_0}(f_1 b + f_2 Ab + ... + f_d A^{d-1} b)$$

Sparse linear system and P-adic lifting

Computing the modular inverse is proscribed due to fill-in

Solution [Kaltofen, Saunders 1991] :
$\hookrightarrow$ use modular minimal polynomial instead of inverse

Wiedemann approach (1986)

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

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

$$A^{-1}b \quad = \quad \underbrace{\frac{-1}{f_0}(f_1 b + f_2 Ab + \ldots + f_d A^{d-1} b)}_{x}$$

P-adic algorithm for sparse systems

Scheme to compute $A^{-1}b$ [Kaltofen, Saunders 1991] :

$$f^A := \mathrm{minpoly}(A) \bmod p \tag{1-1}$$

$$r := b \tag{1-2}$$

$$\text{for } i := 0 \text{ to } k$$

$$x_i := \frac{-1}{f_0} \sum_{i=1}^{\deg f^A} f_i A^{i-1} r \bmod p \tag{2-1}$$

$$r := (1/p)(r - A.x_i) \tag{2-2}$$

$$x := \sum_{i=0}^{k} x_i . p^i \tag{3-1}$$

$$\textit{rational reconstruction on } x \tag{3-2}$$

P-adic algorithm for sparse systems

Scheme to compute $A^{-1}b$ [Kaltofen, Saunders 1991] :

(1-1)	$f^A := \mathrm{minpoly(A)} \bmod \mathrm{p}$	$O\tilde{}(n^2 \log \|\|A\|\|)$
(1-2)	$r := b$	
	for $i := 0$ to k	$k = O\tilde{}(n)$
(2-1)	$\quad x_i := \frac{-1}{f_0} \sum_{i=1}^{\deg f^A} f_i A^{i-1} r \bmod p$	$O\tilde{}(n \deg f^A \log \|\|A\|\|)$
(2-2)	$\quad r := (1/p)(r - A.x_i)$	$O\tilde{}(n \log \|\|A\|\|)$
(3-1)	$x := \sum_{i=0}^{k} x_i.p^i$	
(3-2)	*rational reconstruction on x*	

P-adic algorithm for sparse systems

Scheme to compute $A^{-1}b$ [Kaltofen, Saunders 1991] :

(1-1) $f^A := \text{minpoly}(\mathrm{A}) \bmod \mathrm{p}$

(1-2) $r := b$

for $i := 0$ to k $\qquad k = \tilde{O}(n)$

(2-1) $x_i := \frac{-1}{f_0} \sum_{i=1}^{\deg f^A} f_i A^{i-1} r \bmod p$ $\qquad \tilde{O}(n \deg f^A \log ||A||)$

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

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

(3-2) *rational reconstruction on x*

worst case $\deg f^A = n$ gives a complexity of $\tilde{O}(n^3 \log ||A||)$

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*	*1 600*	*2 500*	*3 600*
Maple	64.7s	849s	11 098s	–	–
CRA-Wied	14.8s	168s	1 017s	3 857s	11 452s
P-adic-Wied	10.2s	113s	693s	2 629s	8 034s
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*	*1 600*	*2 500*	*3 600*
Maple	64.7s	849s	11 098s	–	–
CRA-Wied	14.8s	168s	1 017s	3 857s	11 452s
P-adic-Wied	10.2s	113s	693s	2 629s	8 034s
Dixon	**0.9s**	**10s**	**42s**	**178s**	**429s**

main difference :

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

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

Remark :

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

Outline

I. a small guide to rational linear system solving

II. a quest to improve the cost of rational sparse solver

III. what are benefits in practice ?

IV. conclusion and future work

Our objectives

In practice :

Integrate level 2 and 3 BLAS in integer sparse solver

In theory :

Improve bit complexity of sparse linear system solving

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

Integration of BLAS in sparse solver

Our goals :

- minimize the number of sparse matrix-vector products.
- maximize the number of level 2 and 3 BLAS operations.

$\hookrightarrow$ Block Wiedemann algorithm seems to be a good candidate

Let s be the blocking factor of Block Wiedemann algorithm.
then

- the number of sparse matrix-vector product is divided by roughly s.
- order s matrix operations are integrated.

A good candidate : Block Wiedemann

- Replace vector projections by block of vectors projections

$$s\{\begin{pmatrix} & u & \end{pmatrix}\begin{pmatrix} & & \\ & A^i & \\ & & \end{pmatrix}\overbrace{\begin{pmatrix} \\ v \\ \\ \end{pmatrix}}^{s} \leftarrow b \text{ is 1st column of } v$$

Block Wiedemann approach [Coppersmith 1994]

Let $A \in \mathbb{F}^{n \times n}$ of full rank, $b \in \mathbb{F}^n$ and $n = m \times s$.

One can use a column of the minimal generating matrix polynomial $P \in \mathbb{F}[x]^{s \times s}$ of sequence $\{uA^i v\}$ to express $A^{-1}b$ as a linear combination of block krylov subspace $\{v, Av, \ldots, A^m v\}$

A good candidate : Block Wiedemann

- Replace vector projections by block of vectors projections

$$s\{\begin{pmatrix} \quad u \quad \end{pmatrix} \begin{pmatrix} & & \\ & A^i & \\ & & \end{pmatrix} \overbrace{\begin{pmatrix} v \end{pmatrix}}^{s} \leftarrow b \text{ is 1st column of } v$$

Block Wiedemann approach [Coppersmith 1994]

Let $A \in \mathbb{F}^{n\times n}$ of full rank, $b \in \mathbb{F}^{n}$ and $n = m \times s$.

One can use a column of the minimal generating matrix polynomial $P \in \mathbb{F}[x]^{s\times s}$ of sequence $\{uA^i v\}$ to express $A^{-1}b$ as a linear combination of block krylov subspace $\{v, Av, \ldots, A^m v\}$

the cost to compute P is :

- $O\tilde{}(s^3 m)$ field op. [Beckermann, Labahn 1994 ; Kaltofen 1995 ; Thomé 2002],
- $O\tilde{}(s^\omega m)$ field op. [Giorgi, Jeannerod, Villard 2003].

Block Wiedemann and P-adic

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

(1-1) $r := b$

for $i := 0$ to k

(2-1) $v_{*,1} := r$

(2-2) $P :=$ block minpoly $\{uA^i v\}$ mod p

(2-3) $x_i :=$ linear combi $(A^i v, P)$ mod p

(2-4) $r := (1/p)(r - A.x_i)$

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

(3-2) *rational reconstruction on x*

Block Wiedemann and P-adic

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

(1-1)	$r := b$	
	for $i := 0$ to k	$k = O\tilde{~}(n)$
(2-1)	$v_{*,1} := r$	
(2-2)	$P :=$ block minpoly $\{uA^i v\}$ mod p	$O\tilde{~}(s^2 n \log \|\|A\|\|)$
(2-3)	$x_i :=$ linear combi $(A^i v, P)$ mod p	$O\tilde{~}(n^2 \log \|\|A\|\|)$
(2-4)	$r := (1/p)(r - A.x_i)$	$O\tilde{~}(n \log \|\|A\|\|)$
(3-1)	$x := \sum_{i=0}^{k} x_i.p^i$	
(3-2)	*rational reconstruction on x*	

Block Wiedemann and P-adic

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

(1-1) $r := b$

for $i := 0$ to k — $k = O\tilde{}(n)$

(2-1) $v_{*,1} := r$

(2-2) $P :=$ block minpoly $\{uA^i v\}$ mod p — $O\tilde{}(s^2 n \log ||A||)$

(2-3) $x_i :=$ linear combi $(A^i v, P)$ mod p — $O\tilde{}(n^2 \log ||A||)$

(2-4) $r := (1/p)(r - A.x_i)$

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

(3-2) *rational reconstruction on x*

Not satisfying : computation of block minpoly. at each steps

How to avoid the computation of the block minimal polynomial ?

Our alternative to Block Wiedemann

Express the inverse of the sparse matrix through a structured form
$\hookrightarrow$ block Hankel/Toeplitz structures

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

$$U = \begin{pmatrix} u \\ uA \\ \vdots \\ uA^{m-1} \end{pmatrix}, \; V = \left(\begin{array}{c|c|c|c} v & Av & \dots & A^{m-1}v \end{array}\right) \in \mathbb{F}^{\mathrm{n}\times\mathrm{n}}$$

Our alternative to Block Wiedemann

Express the inverse of the sparse matrix through a structured form
$\hookrightarrow$ 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

$$U = \begin{pmatrix} u \\ uA \\ \vdots \\ uA^{m-1} \end{pmatrix}, \; V = \left(\begin{array}{c|c|c|c} v & Av & \dots & A^{m-1}v \end{array}\right) \in \mathbb{F}^{n\times n}$$

then we can define the block Hankel matrix

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

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

Block-Hankel matrix inversion

Nice property on block Hankel matrix inverse

[Gohberg, Krupnik 1972, Labahn, Choi, Cabay 1990]

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

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

Block-Hankel matrix inversion

Nice property on block Hankel matrix inverse

[Gohberg, Krupnik 1972, Labahn, Choi, Cabay 1990]

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

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

- Computing inverse formula of H^{-1} reduces to matrix-polynomial multiplication : $O\tilde{}(s^3 m)$ [Giorgi, Jeannerod, Villard 2003].
- Computing $H^{-1}v$ for any vector v reduces to matrix-polynomial/vector-polynomial multiplication : $O\tilde{}(s^2 m)$

On the way to a better algorithm

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

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

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

(1-3) $r := b$

for $i := 0$ to k

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

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

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

(3-2) *rational reconstruction on x*

On the way to a better algorithm

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

(1-1)	$H(z) := \sum_{i=1}^{2m-1} uA^i v.z^{i-1} \bmod p$	$O\tilde{}(sn^2 \log \|\|A\|\|)$
(1-2)	compute $H^{-1} \bmod p$ 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)	$x_i := VH^{-1}U.r \bmod p$	$O\tilde{}((n^2 + sn) \log \|\|A\|\|)$
(2-2)	$r := (1/p)(r - A.x_i)$	$O\tilde{}(n \log \|\|A\|\|)$
(3-1)	$x := \sum_{i=0}^{k} x_i.p^i$	
(3-2)	*rational reconstruction on x*	

On the way to a better algorithm

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

(1-1) $H(z) := \sum_{i=1}^{2m-1} uA^i v.z^{i-1} \bmod p$ $\qquad O\tilde{}(sn^2 \log ||A||)$

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

(1-3) $r := b$

for $i := 0$ to k $\qquad k = O\tilde{}(n)$

(2-1) $\quad x_i := V H^{-1} U.r \bmod p$ $\qquad O\tilde{}((n^2 + sn) \log ||A||)$

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

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

(3-2) *rational reconstruction on x*

<u>Not yet satisfying</u> : applying matrices U and V is too costly

Applying block Krylov subspaces

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

can be rewrite as

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

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

$$V = \left(v \middle| Av \middle| \ldots \middle| A^{m-1}v \right) \in \mathrm{I\!F}^{\mathrm{n\times n}} \text{ and } \mathrm{v} \in \mathrm{I\!F}^{\mathrm{n\times s}}$$

can be rewrite as

$$V = \left(v \middle| \quad \right) + A \left(\quad \middle| v \middle| \quad \right) + \ldots + A^{m-1} \left(\quad \middle| v \right)$$

Therefore, applying V to a vector corresponds to :

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

Applying block Krylov subspaces

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

can be rewrite as

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

Therefore, applying V to a vector corresponds to :

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

How to improve the complexity ?

Applying block Krylov subspaces

$$V = \left(v \middle| Av \middle| \ldots \middle| A^{m-1}v \right) \in \mathrm{I\!F}^{\mathrm{n}\times\mathrm{n}} \text{ and } \mathrm{v} \in \mathrm{I\!F}^{\mathrm{n}\times\mathrm{s}}$$

can be rewrite as

$$V = \left(v \middle| \quad \right) + A \left(\quad \middle| v \middle| \quad \right) + \ldots + A^{m-1} \left(\quad \middle| v \right)$$

Therefore, applying V to a vector corresponds to :

- $m-1$ linear combinations of columns of v $\quad O\tilde{~}(m \times sn \log ||A||)$
- $m-1$ applications of A

How to improve the complexity ?

$\Rightarrow$ try to use special block projections u and v

Definition of suitable block projections

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

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

Definition :

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

such that :

1. $\mathcal{K}(RA, v)$ and $\mathcal{K}((RA)^T, u^T)$ are non-singular,

Definition of suitable block projections

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

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

Definition :

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

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 $\tilde{O}(n)$ operations,

Definition of suitable block projections

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

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

Definition :

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

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 $O\tilde{~}(n)$ operations,
3. u, u^T, v and v^T can be applied to a vector with $O\tilde{~}(n)$ operations.

A suitable sparse block projection

Theorem [Eberly, Giesbrecht, Giorgi, Storjohann, Villard - ISSAC'07 submission] :

Let $v^T = \begin{pmatrix} \mathrm{I}_s & \dots & \mathrm{I}_s \end{pmatrix} \in \mathbb{F}^{n \times s}$ (*m copies of $s \times s$ identity*) and let $\mathcal{D} = \mathrm{diag}(\delta_1, \dots, \delta_1, \delta_2, \dots, \delta_2, \dots, \delta_m, \dots, \delta_m)$ be an $n \times n$ diagonal matrix with m distinct indeterminates δ_i, each occurring s times.

If the leading $ks \times ks$ minor of A is non-zero for $1 \leq k \leq m$, then $\mathcal{K}(\mathcal{D}A\mathcal{D}, v) \in \mathbb{F}^{n \times n}$ is invertible.

A suitable sparse block projection

Theorem [Eberly, Giesbrecht, Giorgi, Storjohann, Villard - ISSAC'07 submission] :

Let $v^T = \begin{pmatrix} \mathrm{I}_s & \dots & \mathrm{I}_s \end{pmatrix} \in \mathbb{F}^{n \times s}$ (*m copies of $s \times s$ identity*) and let $\mathcal{D} = \mathrm{diag}(\delta_1, \dots, \delta_1, \delta_2, \dots, \delta_2, \dots, \delta_m, \dots, \delta_m)$ be an $n \times n$ diagonal matrix with m distinct indeterminates δ_i, each occurring s times.

If the leading $ks \times ks$ minor of A is non-zero for $1 \leq k \leq m$, then $\mathcal{K}(\mathcal{D}A\mathcal{D}, v) \in \mathbb{F}^{n \times n}$ is invertible.

Assuming that $\#\mathbb{F} > n(n+1)$

Let $A \in \mathbb{F}^{n \times n}$ a non-singular matrix with all leading minors being non zero and $D \in \mathbb{F}^{n \times n}$ a diagonal matrix. Then the triple $(R, \hat{u}, \hat{v})$ such that $R = D^2$, $\hat{u}^T = D^{-1}v$ and $\hat{v} = Dv$ define a suitable block projection.

A suitable sparse block projection

Theorem [Eberly, Giesbrecht, Giorgi, Storjohann, Villard - ISSAC'07 submission] :

Let $v^T = \begin{pmatrix} \mathrm{I}_s & \dots & \mathrm{I}_s \end{pmatrix} \in \mathrm{I\!F}^{n \times s}$ (*m copies of $s \times s$ identity*) and let $\mathcal{D} = \mathrm{diag}(\delta_1, \dots, \delta_1, \delta_2, \dots, \delta_2, \dots, \delta_m, \dots, \delta_m)$ be an $n \times n$ diagonal matrix with m distinct indeterminates δ_i, each occurring s times.

If the leading $ks \times ks$ minor of A is non-zero for $1 \leq k \leq m$, then $\mathcal{K}(\mathcal{D}A\mathcal{D}, v) \in \mathrm{I\!F}^{n \times n}$ is invertible.

Assuming that $\#\mathrm{I\!F} > n(n+1)$

Let $A \in \mathrm{I\!F}^{n \times n}$ a non-singular matrix with all leading minors being non zero and $D \in \mathrm{I\!F}^{n \times n}$ a diagonal matrix. Then the triple $(R, \hat{u}, \hat{v})$ such that $R = D^2$, $\hat{u}^T = D^{-1}v$ and $\hat{v} = Dv$ define a suitable block projection.

Remark : The same result holds for arbitrary non-singular matrices (Toeplitz preconditioners achieve generic rank profile [Kaltofen, Saunders 1991].)

Our new algorithm

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

(1-1) choose R and blocks $\hat{\mathrm{u}}, \hat{\mathrm{v}}$

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

(1-3) $$H(z) := \sum_{i=1}^{2m-1} \hat{\mathrm{u}} A^i \hat{\mathrm{v}}.z^{i-1} \bmod p$$

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

(1-5) $r := b$

for $i := 0$ to k

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

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

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

(3-2) *rational reconstruction on x*

Our new algorithm

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

(1-1)	choose R and blocks $\hat{u}, \hat{v}$	
(1-2)	set $A := R.A$ and $b := R.b$	
(1-3)	$H(z) := \sum_{i=1}^{2m-1} \hat{u}A^i\hat{v}.z^{i-1} \bmod p$	$\tilde{O}(n^2 \log \lVert A \rVert)$
(1-4)	compute $H^{-1} \bmod p$ from $H(z)$	$\tilde{O}(s^2 n \log \lVert A \rVert)$
(1-5)	$r := b$	
	for $i := 0$ to k	$k = \tilde{O}(n)$
(2-1)	$x_i := VH^{-1}U.r \bmod p$	$\tilde{O}((mn + sn) \log \lVert A \rVert)$
(2-2)	$r := (1/p)(r - A.x_i)$	$\tilde{O}(n \log \lVert A \rVert)$
(3-1)	$x := \sum_{i=0}^{k} x_i.p^i$	
(3-2)	*rational reconstruction on x*	

Our new algorithm

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

(1-1)	choose R and blocks $\hat{u}, \hat{v}$	
(1-2)	set $A := R.A$ and $b := R.b$	
(1-3)	$H(z) := \sum_{i=1}^{2m-1} \hat{u}A^i\hat{v}.z^{i-1} \bmod p$	$\tilde{O}(n^2 \log \|\|A\|\|)$
(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)	$x_i := VH^{-1}U.r \bmod p$	$\tilde{O}((mn + sn) \log \|\|A\|\|)$
(2-2)	$r := (1/p)(r - A.x_i)$	$\tilde{O}(n \log \|\|A\|\|)$
(3-1)	$x := \sum_{i=0}^{k} x_i.p^i$	
(3-2)	*rational reconstruction on x*	

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

Outline

I. a small guide to rational linear system solving

II. a quest to improve the cost of rational sparse solver

III. what are benefits in practice ?

IV. conclusion and future work

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)
  $\hookrightarrow$ *FFT, Karatsuba, middle product*
- fast application of H^{-1} is needed to get $O\tilde{}(n^{2.5}\log||A||)$

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

Is our new algorithm efficient in practice?

Comparing performances

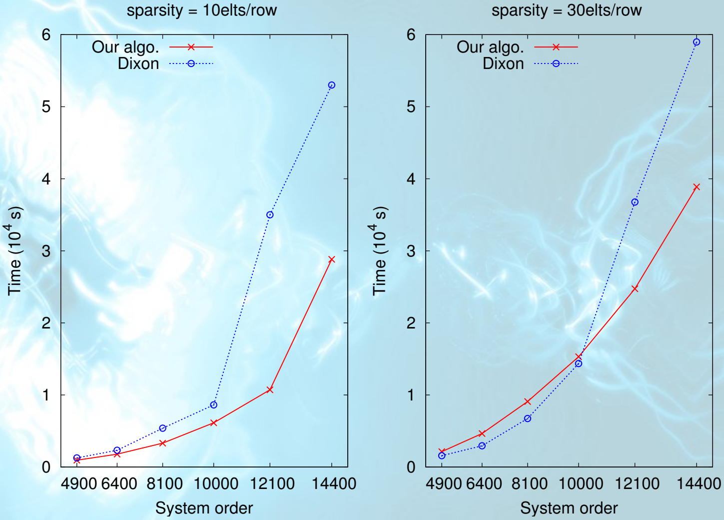
use of LinBox library on Itanium II - 1.3Ghz, 128Gb RAM

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

	system order				
	900	*1600*	*2500*	*3600*	*4900*
Maple 10	849s	11 098s	–	–	–
CRA-Wied	168s	1 017s	3 857s	11 452s	≈ 28 000s
P-adic-Wied	113s	693s	2 629s	8 034s	≈ 20 000s
Dixon	**10s**	**42s**	178s	429s	1 257s
Our algo.	15s	61s	**175s**	**426s**	**937s**

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 ≈ 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 ≈ 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

Outline

I. a small guide to rational linear system solving

II. a quest to improve the cost of rational sparse solver

III. what are benefits in practice ?

IV. conclusion and future work

Conclusions

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

- improve the best known complexity by a factor $\sqrt{n}$.
- improve efficiency by minimizing sparse matrix operations and maximizing dense block operations.

minor drawback : not taking advantage of low degree minimal polynomial

Our sparse block projections yield other improvement for sparse linear algebra [Eberly, Giesbrecht, Giorgi, Storjohann, Villard - ISSAC'07 submission] :

- sparse matrix inversion over a field in $O\tilde{}(n^{2.27})$ field op.
- integer sparse matrix determinant & Smith form in $O\tilde{}(n^{2.66})$ bit op.

Future work

- provide an automatic choice of block dimension (non square ?)
- handle the case of singular matrix
- optimize code (minimize the constant)
- introduce fast matrix multiplication in the complexity
- asymptotic implications in exact linear algebra

Future work

- provide an automatic choice of block dimension (non square ?)
- handle the case of singular matrix
- optimize code (minimize the constant)
- introduce fast matrix multiplication in the complexity
- asymptotic implications in exact linear algebra

our LinBox library is available at

`www.linalg.org`

Future work

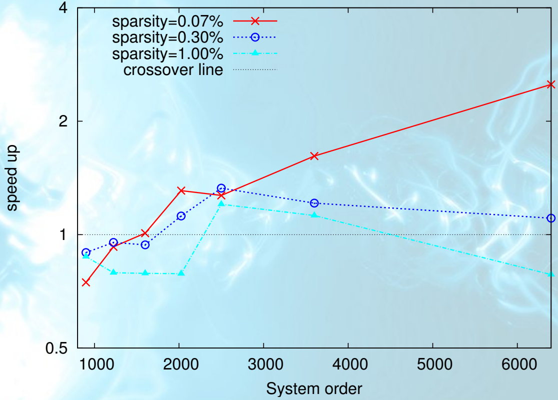
- provide an automatic choice of block dimension (non square ?)
- handle the case of singular matrix
- optimize code (minimize the constant)
- introduce fast matrix multiplication in the complexity
- asymptotic implications in exact linear algebra

our LinBox library is available at

`www.linalg.org`

Questions ?

Sparse solver vs Dixon's algorithm



The sparser the matrices are, the earlier the crossover appears