Solving Sparse Integer Linear Systems

Pascal Giorgi

Dali team - University of Perpignan (France)

in collaboration with

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

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

LMC - MOSAIC Seminar, June 21, 2006

Motivations

Large linear systems are involved
in many mathematical applications

over a 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]

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

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

$$x = A^{-1}b = \begin{pmatrix} \frac{-9591197817}{95078} \\ \frac{131244}{47539} \\ \frac{2909895}{665546} \\ \frac{2909895}{665546} \end{pmatrix}$$

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

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

$$x = A^{-1}b = \begin{pmatrix} \frac{-378283}{1076295} \\ \frac{-107}{11} \\ \frac{155201}{1174140} \\ \frac{934024}{1076295} \end{pmatrix}$$

Main difficulty : expression swell and take advantage of sparsity

Interest in linear algebra

Integer linear systems are central in recent linear algebra algorithms

- Determinant
  [Abbott, Bronstein, Mulders 1999 ; Storjohann 2005]
- Smith Form
  [Eberly, Giesbrecht, Villard 2000]
- Nullspace, Kernel
  [Chen, Storjohann 2005]
- Diophantine solutions
  [Giesbrecht 1997 ; Giesbrecht, Lobo, Saunders 1998 ; Mulders, Storjohann 2003 ; Mulders 2004]

Algorithms for non-singular system solving

Dense matrices :

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

Sparse matrices :

- P-adic lifting or CRA [Wiedemann 1986 ; Kaltofen, Saunders 1991]
  $\hookrightarrow$ $O(\gamma n^2(\log(n) + \log ||A||))$ bit operations with γ non-zero elts.

P-adic algorithm with matrix inversion

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

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

(1-2) $r := b$

for $i := 0$ to k

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

(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*

P-adic algorithm with matrix inversion

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

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

P-adic algorithm with matrix inversion

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

(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 in practice

Efficient implementations are available :
LinBox 1.0 [www.linalg.org]
IML library [www.uwaterloo.ca/~z4chen/iml]

Details :

- level 3 BLAS-based matrix inversion over prime field
  - with LQUP factorization [Dumas, Giorgi, Pernet 2004]
  - with Echelon form [Chen, Storjohann 2005]
- level 2 BLAS-based matrix-vector product
  - use of CRT over the integers
- rational number reconstruction
  - half GCD [Schönage 1971]
  - heuristic using integer multiplication [NTL library]

Timing for dense linear system solving

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

- random dense linear system with coefficients over 3 bits :

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 coefficients over 20 bits :

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

performances improvement by a factor 10
compare to 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.

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

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

Sparse linear system and P-adic lifting

P-adic lifting doesn't improve complexity as in dense case.
↪ computing the modular inverse is proscribed due to fill-in

Solution [Wiedemann 1986 ; Kaltofen, Saunders 1991] :
↪ use modular minimal polynomial instead of inverse

Sparse linear system and P-adic lifting

P-adic lifting doesn't improve complexity as in dense case.
↪ computing the modular inverse is proscribed due to fill-in

Solution [Wiedemann 1986 ; Kaltofen, Saunders 1991] :
↪ use modular minimal polynomial instead of inverse

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

Let $\Pi(\lambda) = c_0 + c_1\lambda + ... + \lambda^d \in \mathbb{Z}_p[\lambda]$ be the minimal polynomial of A

Sparse linear system and P-adic lifting

P-adic lifting doesn't improve complexity as in dense case.
$\hookrightarrow$ computing the modular inverse is proscribed due to fill-in

Solution [Wiedemann 1986 ; Kaltofen, Saunders 1991] :
$\hookrightarrow$ use modular minimal polynomial instead of inverse

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

Let $\Pi(\lambda) = c_0 + c_1\lambda + ... + \lambda^d \in \mathbb{Z}_p[\lambda]$ be the minimal polynomial of A

$$A^{-1}b \quad = \quad \frac{-1}{c_0}(c_1 b + c_2 Ab + ... + A^{d-1}b)$$

Sparse linear system and P-adic lifting

P-adic lifting doesn't improve complexity as in dense case.
↪ computing the modular inverse is proscribed due to fill-in

Solution [Wiedemann 1986 ; Kaltofen, Saunders 1991] :
↪ use modular minimal polynomial instead of inverse

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

Let $\Pi(\lambda) = c_0 + c_1\lambda + \ldots + \lambda^d \in \mathbb{Z}_p[\lambda]$ be the minimal polynomial of A

$$A^{-1}b \quad = \quad \underbrace{\frac{-1}{c_0}(c_1 b + c_2 Ab + \ldots + A^{d-1}b)}_{x}$$

P-adic algorithm for sparse systems

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

(1-1) $\Pi := minpoly(A) \bmod p$

(1-2) $r := b$

for $i := 0$ to k

(2-1) $$x_i := \frac{-1}{\Pi_{[0]}} \sum_{i=1}^{\deg \Pi} \Pi_{[i]} A^{i-1} r \bmod p$$

(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*

P-adic algorithm for sparse systems

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

(1-1)	$\Pi := minpoly(A) \bmod p$	$O\tilde{}(n^2 \log \|\|A\|\|)$
(1-2)	$r := b$	
	for $i := 0$ to k	$k = O\tilde{}(n)$
(2-1)	$x_i := \frac{-1}{\Pi_{[0]}} \sum_{i=1}^{\deg \Pi} \Pi_{[i]} A^{i-1} r \bmod p$	$O\tilde{}(n^2 \log \|\|A\|\|)$
(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*	

P-adic algorithm for sparse systems

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

(1-1)	$\Pi := minpoly(A) \bmod p$	
(1-2)	$r := b$	
	for $i := 0$ to k	$k = \tilde{O}(n)$
(2-1)	$x_i := \frac{-1}{\Pi_{[0]}} \sum_{i=1}^{\deg \Pi} \Pi_{[i]} A^{i-1} r \bmod p$	$\tilde{O}(n^2 \log \lvert\lvert A \rvert\rvert)$
(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*	

Integer sparse linear system in practice

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

- random non-singular sparse linear system with coefficients over 3 bits and 10 non zero elements per row.

	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**	**429**s

Integer sparse linear system in practice

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

- random non-singular sparse linear system with coefficients over 3 bits and 10 non zero elements per row.

	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**	**429**s

main difference :

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

$$x_i := \frac{-1}{\Pi_{[0]}} \sum_{i=1}^{\deg \Pi} \Pi_{[i]} A^{i-1} r \bmod p \quad \text{(2-1)} \qquad (P\text{-}adic\text{-}Wied)$$

Remark :

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

Our objectives

In pratice :

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.

↪ 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.

Block Wiedemann and P-adic

- Replace vector projections by block of vectors projections

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

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

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

Block Wiedemann and P-adic

- Replace vector projections by block of vectors projections

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

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

One can use a column of the minimal generating matrix polynomial $P \in \mathbb{Z}_p[x]^{s\times s}$ of sequence $\{uA^iv\}$ 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 — $k = \tilde{O}(n)$

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

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

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

(2-4) $r := (1/p)(r - A.x_i)$ — $\tilde{O}(n \log ||A||)$

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

(3-2) *rational reconstruction on x*

Block Wiedemann and P-adic

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

(1-1)	$r := b$	
	for $i := 0$ to k	$k = \tilde{O}(n)$
(2-1)	$v_{*,1} := r$	
(2-2)	$P :=$ block minpoly $\{uA^i v\}$ mod p	$\tilde{O}(s^2 n \log \lVert A \rVert)$
(2-3)	$x_i :=$ linear combi $(A^i v, P)$ mod p	$\tilde{O}(n^2 \log \lVert A \rVert)$
(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 ?

Alternative to Block Wiedemann

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

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

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

Alternative to Block Wiedemann

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

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

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

then we can define the block Hankel matrix

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

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]

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

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]

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

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 Pade approximants of $H(z) = \alpha_1 + \alpha_2 z + \ldots + \alpha_{2m-1} z^{2m-2}$ [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) $H(z) := \sum_{i=1}^{2m-1} uA^i v.z^{i-1} \bmod p$

(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

Alternative to Block Wiedemann

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*	

Alternative to Block Wiedemann

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

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

Applying block Krylov subspaces

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

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
- $m-1$ applications of A

Applying block Krylov subspaces

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

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 — $O(m \times sn \log ||A||)$
- $m-1$ applications of A — $O(mn \log ||A||)$

Applying block Krylov subspaces

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

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 $\qquad O(m \times sn \log ||A||)$
- $m-1$ applications of A

How to improve the complexity ?

Applying block Krylov subspaces

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

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 $\quad O(m \times sn \log ||A||)$
- $m-1$ applications of A

How to improve the complexity ?

$\Rightarrow$ using special block projections u and v

Candidates as suitable block projections

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

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

A suitable block projection is defined through the triple

$$(R, u, v) \in \mathbb{Z}_p^{n \times n} \times \mathbb{Z}_p^{s \times n} \times \mathbb{Z}_p^{n \times s}$$

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

Candidates as suitable block projections

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

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

A suitable block projection is defined through the triple

$$(R, u, v) \in \mathbb{Z}_p^{n\times n} \times \mathbb{Z}_p^{s\times n} \times \mathbb{Z}_p^{n\times s}$$

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

Conjecture :

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

A structured block projection

Let u and v be defined as follow

$$u = \begin{pmatrix} u_1 \ \ldots \ u_m & & & \\ & u_{m+1} \ \ldots \ u_{2m} & & \\ & & \ddots & \\ & & & u_{n-m+1} \ \ldots \ u_n \end{pmatrix} \in \mathbb{Z}_p^{s \times n}$$

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

where u_i and v_i are chosen randomly from a sufficient large set.

A structured block projection

Let u and v be defined as follow

$$u = \begin{pmatrix} u_1 \dots u_m & & & \\ & u_{m+1} \dots u_{2m} & & \\ & & \ddots & \\ & & & u_{n-m+1} \dots u_n \end{pmatrix} \in \mathbb{Z}_p^{s \times n}$$

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

where u_i and v_i are chosen randomly from a sufficient large set.

<u>open question</u> : Let R diagonal and v as defined above,

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

We prooved 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 block projection u and v

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

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

(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 block projection u and v	
(1-2)	choose R and $A := R.A$, $b := R.b$	
(1-3)	$H(z) := \sum_{i=1}^{2m-1} uA^i v.z^{i-1} \bmod p$	$\tilde{O}(n^2 \log \|\|A\|\|)$
(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*	

Our new algorithm

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

(1-1)	choose block projection u and v	
(1-2)	choose R and $A := R.A$, $b := R.b$	
(1-3)	$H(z) := \sum_{i=1}^{2m-1} uA^i v.z^{i-1} \bmod p$	$\tilde{O}(n^2 \log \|\|A\|\|)$
(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 $\tilde{O}(n^{2.5} \log ||A||)$

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)
  ↪ *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
    ↪ *use Vandermonde matrix to have dense matrix operations*

Is our new algorithm efficient in practice ?

Performances

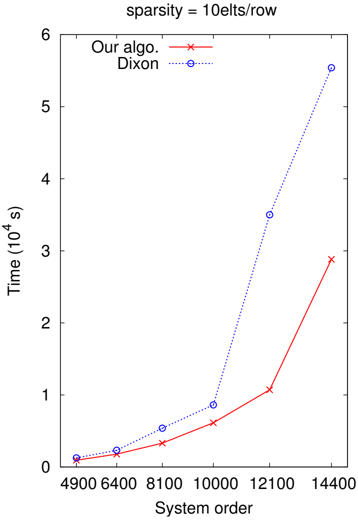
use of LinBox library on Itanium II - 1.3Ghz, 128Go RAM

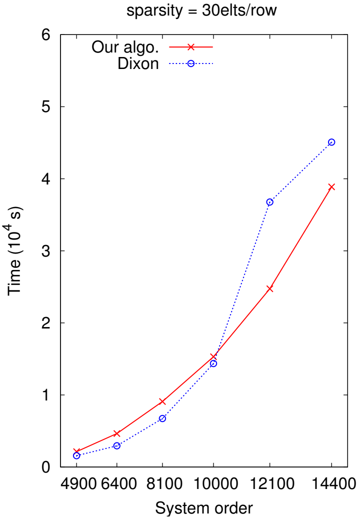
- random non-singular sparse linear system with coefficients over 3 bits and 10 non zero elements per row.

	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
Our algo.	2.4s	15s	61s	**175s**	**426s**

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

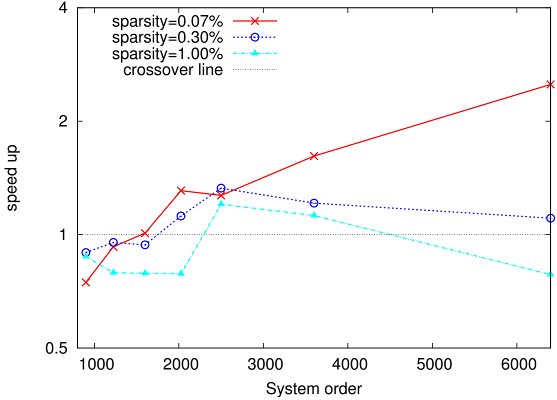
Sparse solver vs Dixon's algorithm





Our algorithm performances are depending on matrix sparsity

Sparse solver vs Dixon's algorithm



The sparser the matrices are, the earlier the crossover appears

Practical effect on blocking factors

$\sqrt{n}$ blocking factor value is theoretically optimal

Is this still true in practice ?

Practical effect on 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 on 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 certainly depending on the ratio of sparse matrix/dense matrix operations efficiency

Conclusions

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

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

We introduce special block projections for sparse linear algebra
$\hookrightarrow$ inverse of sparse matrix in $O(n^{2.5})$ field op.

drawback : not taking advantage of low degree minimal polynomial

Conclusions

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

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

We introduce special block projections for sparse linear algebra
$\hookrightarrow$ inverse of sparse matrix in $O(n^{2.5})$ field op.

drawback : not taking advantage of low degree minimal polynomial

On going work :

- provide an automatic choice of block dimension (non square ?)
- proove conjecture for special block projections
- how to handle the case of singular matrix ?
- how to introduce fast matrix multiplication in the complexity ?