Solving sparse systems of linear equations
(with symbolic entries)

ERICH KALTOFEN

Rensselaer Polytechnic Institute
Department of Computer Science

Joint work with: B. DAVID SAUNDERS
University of Delaware
Department of Computer & Inform. Sciences

AUSTIN LOBO (graduate student, RPI)
Outline

• The non-singular case
  ○ what is a sparse matrix?
  ○ Wiedemann’s method

• The singular case
  ○ making principal sub-matrices non-singular
    — by Toeplitz matrix perturbation
    — by Benefeš permutation networks
  ○ computing the rank
  ○ picking a random solution

• Implementation efforts
  ○ on Sparc 2 workstations

• Open problems
What is a sparse matrix?

- **matrices with “few” non-zero entries**
  - a band matrix from a finite element method
  - a matrix over GF(2) from integer factoring by the NFS: $52250 \times 50001$ with 1095532 entries $\neq 0$ ($\approx 21$/row)

- **matrices with special structure**
  - the Sylvester matrix corresponding to a polynomial resultant

$$R = \begin{pmatrix}
    a_n & a_{n-1} & \cdots & a_0 \\
    a_n & \cdots & a_1 & a_0 & 0 \\
    0 & \cdots & \cdots & \cdots \\
    0 & \cdots & \cdots & \cdots \\
    b_n & b_{n-1} & \cdots & b_0 \\
    b_n & \cdots & b_1 & b_0 & 0 \\
    0 & \cdots & \cdots & \cdots \\
    b_n & \cdots & \cdots & b_0
\end{pmatrix}$$
• a “black box” matrix
  an efficient program with the specifications

\[
y \in K^n \rightarrow A \times y \in K^n
\]

\[A \in K^{n \times n}\]

\[K \text{ an arbitrary field}\]

e.g., for the Sylvester matrix \( R \), \( R \times y \) costs

\[O(n \log(n) \log\log(n))\]

arithmetic operations using fast polynomial multiplication
Symbolic objects given by black box representation are known for many problems:

- symbolic determinants using Gaussian elimination

- the polynomial remainder sequence of \( f_0(x) \) and \( f_1(x) \) using continued fraction approximations

\[
\{ q_i(x) \}_{i \geq 2} \text{ such that } f_i(x) = f_{i-2}(x) - q_i(x)f_{i-1}(x)
\]

- \( A^{-1} = P^{-1}U^{-1}L^{-1} \), the \( LUP \) factorization of \( A \in \mathbb{K}^{n \times n} \).

- streams for infinite objects, such as a program for the \( i \)-th order coefficient of a power series
Linear system solution with a black box matrix

Given a black box

\[ y \in K^n \quad \rightarrow \quad \begin{array}{c} A \in K^{n \times n} \text{ non-singular} \\ K \text{ an arbitrary field} \end{array} \quad \rightarrow \quad A \times y \in K^n \]

compute \( A^{-1}b \) “efficiently.”

D. Wiedemann (1986) constructs a Las Vegas randomized algorithm that computes \( A^{-1}b \) in at most

\[ 3n \text{ “} A \times y \text{ steps”} \]

and

\[ O(n^2) \text{ additional arithmetic operations in } K. \]

The algorithm needs \( O(n) \) space.
The Krylov subspace

Consider the minimum linear dependency of the sequence of vectors \( \{A^i b\}_{i \geq 0} \),

\[
\begin{align*}
   f_0(b)b + f_1(b)Ab + f_2(b)A^2b + f_3(b)A^3b + \cdots + f_k(b)A^kb &= 0, \quad f_k(b) \neq 0, \\
   f(b)(\lambda) &= f_0(b) + f_1(b)\lambda + \cdots + f_k(b)\lambda^k \in K[\lambda]
\end{align*}
\]

As a consequence of the Cayley/Hamilton Theorem,

\[ f(b)(\lambda) \quad \text{divides} \quad \text{Det}(\lambda I - A), \quad \text{thus} \ k \leq n. \]

Hence: If \( f_0(b) = 0 \), then \( \text{Det}(A) = 0 \);

otherwise \( A^{-1}b = x \leftarrow -\frac{1}{f_0(b)} \left( f_1(b)b + f_2(b)Ab + \cdots + f_k(b)A^{k-1}b \right) \).
Idea for finding $f^{(b)}(\lambda)$ given $A$ and $b$

Let $u \in K^n$ and consider the sequence of field elements

$$a_0 = u^T b, \ a_1 = u^T Ab, \ a_2 = u^T A^2 b, \ a_3 = u^T A^3 b, \ldots$$

Since $u^T A^j f^{(b)}(A)b = 0$, we have

$$\forall \ j \geq 0: f_0^{(b)} a_{0+j} + f_1^{(b)} a_{1+j} + \cdots + f_k^{(b)} a_{k+j} = 0$$

that is $\{a_i\}_{i=0,1,...}$ satisfies a linear recurrence.

By the Berlekamp/Massey (1969) or the extended Euclidean algorithm we can compute in $O(n l)$ steps a minimal recurrence polynomial

$$f^{(b,u)}(\lambda) = f_0^{(b,u)} + f_1^{(b,u)} \lambda + \cdots + f_{l-1}^{(b,u)} \lambda^{l-1} - \lambda^l$$

that generates $\{a_i\}_{i=0,1,...}$

$$\forall \ j \geq 0: a_{l+j} = f_{l-1}^{(b,u)} a_{l-1+j} + f_{l-2}^{(b,u)} a_{l-2+j} + \cdots + f_0^{(b,u)} a_{0+j}.$$ 

**Important fact:** For “random” $u$ with high probability

$$f^{(b,u)}(\lambda) = f^{(b)}(\lambda).$$
Making leading principal sub-matrices non-singular
a) our method using Toeplitz multipliers

Let $A \in K^{n \times n}$,

$$
\tilde{A} = \begin{pmatrix}
1 & t_2 & t_3 & \ldots & t_n \\
1 & t_2 & \ldots & t_{n-1} \\
1 & \ddots & \ddots & \ddots \\
0 & \ddots & t_2 & 1
\end{pmatrix}
\begin{pmatrix}
1 \\
l_2 \\
l_3 \\
\vdots
\end{pmatrix}
\begin{pmatrix}
l_2 & 1 & 0 \\
l_3 & l_2 & 1 \\
\vdots & \ddots & \ddots \\
l_n & l_{n-1} & \ldots & l_2 & 1
\end{pmatrix}
$$

If $t_i, l_i \in S \subset K$ are randomly and uniformly selected, the probability

$$\text{Prob}(\text{Det}(\tilde{A}_{1\ldots s,1\ldots s}) \neq 0) \geq 1 - \frac{2s}{\text{card}(S)}, \text{ for } s \leq \text{rank}(A)$$

$s$'th leading principal minor

After an idea by Borodin, von zur Gathen, Hopcroft (1982).
b) Wiedemann’s method using Beneš networks

The generic row/column exchange matrix

\[ E(t) = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ -1 - 2t & 1 \end{pmatrix} \]

\[ = \begin{pmatrix} 1 - t - 2t^2 & t \\ -3t - 2t^2 & 1 + t \end{pmatrix} = \begin{cases} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} & \text{for } t = 0 \\ \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} & \text{for } t = -1 \end{cases} \]

Use randomized network exchanges

\[ \tilde{A} = \prod_{i=1}^{2\log_2(n) - 1} E_i(t_{i,1}, \ldots, t_{i,n/2}) \quad A = \prod_{j=1}^{2\log_2(n) - 1} E_j(l_{j,1}, \ldots, l_{j,n/2}) \]

Note that \( V \) and \( W \) are black box matrices with

\[ V \times y \text{ and } W \times y \text{ costing } O(n \log(n)) \text{ field operations.} \]
Computing the rank (without binary search)

Suppose perturbed $\tilde{A}$ has rank $< n$; then for random $d_i$, the minimum polynomial of

$$\tilde{A} \begin{pmatrix} d_1 & & \\ & d_2 & \\ & & \ddots \\ & & & d_n \end{pmatrix}$$

has with high probability degree $= \text{rank}(\tilde{A}) + 1$

Also, with high probability, for random vectors $u$ and $v$,

$$f^{(u,v)}(\lambda) = \text{minimum polynomial}$$
Picking a random solution of a singular system

Let \( \tilde{A} \in \mathbb{K}^{n \times n} \) be of rank \( r \) with the leading principal \( r \times r \) submatrix non-singular; suppose \( \tilde{A}x = b \) is solvable; then for

\[
\tilde{A} \begin{pmatrix}
y' \\
0 \\
\vdots \\
0
\end{pmatrix}
\begin{aligned}
\{&n - r \} = b + \tilde{A}v, \\
v &\text{ random in } \mathbb{K}^n,
\end{aligned}
\]

\( y - v \) uniformly samples the solution manifold of \( \tilde{A}x = b \).
Our current implementation efforts

Ausin Lobo has implemented in C

- the general case using Beneš networks for $K = GF(2^m)$ on Sun4/Sparc2’s

- a special method for finding a non-zero solution of homogenous problems

Comparison with

- LaMacchia and Odlyzko’s conjugate gradient method

- Coppersmith’s blocked Wiedemann method
ODLYZKO’s example over GF(2)

Row nr. Columns with non-zero entries
1 1 2 11 107 118 158 240 305 761 888 6842 12779 26995 44350 47385
2 1 2 11 12 14 20 22 115 247 249 657 1303 5844 7979 20425 24113 26984
3 1 2 3 5 7 42 53 128 173 202 349 371 406 619 4410 6351 30534 50001
4 10 13 50 178 480 843 1153 3557 3619 8042 8754 14355
5 16309 25417 28976 29051 33269 35446 37117

We found one non-zero linear dependence in 113.5 hours on a Sun4, namely the rows

1 6 7 9 12 14 16 17 19 20 21 22 24 ... 49995 49996 49997 49999 50000

(23587 rows are chosen).
Open problems

• **Compute the characteristic polynomial**
  → multi-polynomial resultant computation

• **Reduce cardinality of field in probability estimates**

• **Compute entire right null space**

• **Numerical error analysis**
  → general sparse linear system solver

• **Implement in distribute fashion**
  → Coppersmith’s blocked Wiedemann method on our DSC system