Why solve integer linear systems exactly?
... an enabling tool for discovery and innovation?

David Saunders - U of Delaware

LinBox library at linalg.org
LinBoxers

LinBox started in April, 1998 as an international collaboration seeded by NSF/CNRS $17,500 travel grant.
PIs Erich Kaltofen, David Saunders, and Gilles Villard.

| Li Chen             | U of D                      | Carl Devore             | U of D                      |
| Ahmet Duran        | U of D                      | Jean-Guillaume Dumas    | Université de Grenoble    |
| Pascal Giorgi      | ENS Lyon, U Waterloo       | Wayne Eberly            | University of Calgary     |
| Thierry Gautier    | INRIA Rhne-Alpes           | Mark Giesbrecht         | Waterloo                  |
| Bradford Hovinen   | U of D, U Waterloo, U Toronto | Erich Kaltofen        | North Carolina State University |
| Austin Lobo        | RPI, Washington College    | John P. May             | NCSU, Waterloo, U of D    |
| Clément Pernet     | INP Grenoble, U Waterloo   | David Pritchard         | ENS Lyon                  |
| Daniel Roche       | U of D, U Waterloo         | David Saunders          | U of D                    |
| Arne Storjohann    | U Waterloo                 | William J. Turner       | NCSU                      |
| Anna Urbanska      | Université de Grenoble     | Gilles Villard          | ENS Lyon                  |
| Zhendong Wan       | U of D                      | George Yuhasz           | NCSU                      |
| Undergrads         |                            |                          |                           |
| Frank Bellamy      | U of D                      | Brian Gold              | U of D                    |
| Ben Karel          | U of D                      | Dmitriy Morozov         | NCSU                      |
| Eric Schrag        | U of D                      | Richard Seagraves       | U of D                    |
| Peter Steijn       | U of D                      | Bryan Youse             | U of D                    |
Outline

• What are our basic methods and techniques?
• What problems have we solved for people?
• What might exact linear algebra do for you?
Blackbox algorithms

the minimal polynomial of a matrix is the lowest degree monic polynomial \( m(x) = \sum_i m_i x^i \), such that \( m(A) = \sum_i m_i A^i = 0 \).

Matrix sequence viewpoint (over \( n^2 \) dimensional vector space):

\[
I, \ A, \ A^2, \ A^3, \ \ldots, \ A^f, \ A^{f+1}, \ \ldots
\]

\( m_A(A) = 0 \). (and shifted: \( \sum_i m_i A^{i+k} = 0 \))

Lanczos: Vector sequence viewpoint (over \( n \) dimensional space)

\[
b, \ Ab, \ A^2b, \ A^3b, \ \ldots, \ A^e b, \ A^{e+1} b, \ \ldots
\]

\( m_{A,b}(A)b = \sum_i m_i A^i b = 0 \) (and shifted: \( \sum_i m_i A^{i+k} b = 0 \))

Wiedemann: Scalar sequence viewpoint (over 1 dimensional space)

\[
u^T b, \ u^T Ab, \ u^T A^2 b, \ u^T A^3 b, \ \ldots, \ u^T A^d b, \ u^T A^{d+1} b, \ \ldots
\]

\( u^T m_{u,A,b}(A)b = \sum_i m_i u^T A^i b = 0 \) (and shifted), \( d \leq e \leq f \)
Wiedemann’s method

Wiedemann’s method is to *sparse* matrices as Gaussian elimination is to *dense* matrices.

**Berlekamp/Massey:** if scalar sequence satisfies a linear recurrence of degree $n$ or less, the minimal polynomial may be found from the first $2n$ sequence entries in time $O(n^2)$.

[Wiedemann 88]:

1. If vector $u$ is chosen at random, $m_{A,b} = m_{u,A,b}$ with high probability.

2. If $m_0 \neq 0$ solve, else singular system.
**Consequences:** (a) Compute minimal polynomial $m_A \pmod{p}$ in $O(n^{1+e})$ for matrix with $O(n^e)$ nonzeros. ($e = \log_n(nnz)$, for $n =$ matrix order.

(b) Solve nonsingular system:

\[
0 = m_0b + m_1Ab + \ldots + m_eA^eb \\
= m_0b + A(m_1b + \ldots + m_eA^{e-1}b) \\
\downarrow \\
b = A \times (-1/m_0)(m_1b + \ldots + m_eA^{e-1}b)
\]

(c) With preconditioning: rank, determinant, solve singular systems, characteristic polynomial, invariant factors, etc
LinBox initial concept:

(1) High performance implementation of blackbox methods

(2) Genericity: Wrap any arithmetic mod p (any finite field implementation), wrap any matrix representation.

(3) Application: Build it, they will come (it’s linear algebra after all).

(40 Middleware:
LinBox uses: GMP, ATLAS, NTL, Givaro, ...
LinBox used by: Gap, Maple, Sage, ...

LinBox discovery: A matrix is a matrix. Consequence: Users demand to handle dense matrices, performance demands use BLAS.
Integer Matrix Elimination - extra dimension, Cost

During Gaussian elimination we compute about \((1/3)n^3\) quotients of minors. Sizes averaging \(O^\sim(n)\).

Thus: The size of intermediate storage is \(O^\sim(n^3)\). The time cost with standard integer arithmetic is \(O^\sim(n^5)\). The size of the rational solution vector is itself \(O^\sim(n^2)\).

For example:

10000 \(\times\) 10000 matrix of \(\{0, 1, -1\}\). Initial memory: 100 Megabytes. Intermediate memory need: \(10^{12}\) bytes = 1 Terabyte. Run time: \(10^{20}\) cycles = 300 years at 10 GHz speed.

Length 10000 solution vector is around 100 Megabytes. Entries are rational numbers with numerators and denoms are around 10000 digits long.
modular methods

Given a problem over $\mathbb{Z}$ with answer $a$:

- CRA.
  - project to a problem over $\mathbb{Z}_{p_i}$,
  - compute $a_i \mod p_i$ for many $i$ such that $\prod p_i > a$,
  - reconstruct $a$ using Chinese Remainder Algorithm

- Hensel lifting (variant on Newton iteration)
  - project to a problem over $\mathbb{Z}_p$ and solve to get $a_1$.
  - lift to a solution $a_2$ valid over $\mathbb{Z}_{p^2}$,
  - continue iteratively until $p^e > a$, so that $a_e = a$.
[Dixon 82]

Given $n \times n$ matrix $A$, vector $b$ over $\mathbb{Z}$, solve $Ax = b$. Choose a prime $p$ near $2^{27}$ ($p^2$ fits in fraction of a double precision float)

Compute $LU \ mod \ p$. $O(n^3)$, use BLAS. Set $x_0 = U^{-1}L^{-1}b \ (mod \ p)$. Set $r_1 = (Ax_0 - b)/p. \ O(n^2)$

[Hensel lifting - base p expansion of x ]

for $i = 1 \ to \ n\log_p(n) \ [\text{Hadamard bound}]$ do:

Set $x_i = U^{-1}L^{-1}r_i \ (mod \ p). \ O(n^2)$

Set $r_{i+1} = (Ax - b)/p. \ (\text{Can be done in } O(n^2))$

Thus: for integer $A, b$, obtain rational solution vector $x$ to $Ax = b$

in bit complexity of $O^\sim(n^3)$, memory $O^\sim(n^2)$. 
\[ Ax_0 = b (\text{mod } p), \quad A(x_0 + x_1 p) = b (\text{mod } p^2), \quad A(x_0 + x_1 p + x_2 p^2) = b (\text{mod } p^3), \quad \ldots \]

\[ n^3, \quad n^2, \quad \ldots \]

Dixon’s method Bit complexity of \( O^\sim(n^3) \), memory \( O^\sim(n^2) \): No worse than classic algebraic cost!

For example:

10000 \times 10000 matrix of \{0, 1, -1\}. Initial memory: 100 Megabytes.
Intermediate memory need: \( 10^8 \) bytes = 100 Megabytes.
Run time: \( 10^{12} \) cycles = 17 minutes at 1 GHz speed.

Further work: output sensitive method. Early termination.
Dense Matrix Rational Solving

<table>
<thead>
<tr>
<th>order</th>
<th>100</th>
<th>200</th>
<th>300</th>
<th>400</th>
<th>500</th>
<th>600</th>
<th>700</th>
<th>800</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bottom up</td>
<td>0.91</td>
<td>7.77</td>
<td>29.2</td>
<td>78.38</td>
<td>158.85</td>
<td>298.81</td>
<td>504.87</td>
<td>823.06</td>
</tr>
<tr>
<td>Bottom up (use BLAS)</td>
<td>0.11</td>
<td>0.60</td>
<td>1.61</td>
<td>3.40</td>
<td>6.12</td>
<td>10.09</td>
<td>15.15</td>
<td>21.49</td>
</tr>
<tr>
<td>Top down</td>
<td>0.03</td>
<td>0.20</td>
<td>0.74</td>
<td>1.84</td>
<td>3.6</td>
<td>6.03</td>
<td>9.64</td>
<td>14.31</td>
</tr>
</tbody>
</table>

All entries are randomly and independently chosen from $[-2^{20}, 2^{20}]$.

1. The **Bottom up** is implementation of Dixon lifting without calling BLAS in LinBox.

2. The **Bottom up (use BLAS)** is implemented by Zhuliang Chen the idea of FFLAS and mixture of Dixon lifting and the Chinese remainder algorithm.

3. The **Top down** is implemented by us using hybrid numeric/symbolic solver (Zhendong Wan, thesis 2004)
Trefethen’s matrix

\[ T_{i,j} = \begin{cases} 
  \text{i-th prime}, & \text{if } i = j, \text{[diagonal of primes]} \\
  1, & \text{if } i - j \text{ is a power of 2[bands of 1’s]} \\
  0, & \text{otherwise.[very sparse matrix]} 
\end{cases} \]

\[ T_9 = \begin{bmatrix}
  2 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\
  1 & 3 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\
  1 & 1 & 5 & 1 & 1 & 0 & 1 & 0 & 0 \\
  0 & 1 & 1 & 7 & 1 & 1 & 0 & 1 & 0 \\
  1 & 0 & 1 & 1 & 11 & 1 & 1 & 0 & 1 \\
  0 & 1 & 0 & 1 & 1 & 13 & 1 & 1 & 0 \\
  0 & 0 & 1 & 0 & 1 & 1 & 17 & 1 & 1 \\
  0 & 0 & 0 & 1 & 0 & 1 & 1 & 19 & 1 \\
  1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 23 
\end{bmatrix} \]

\[ T_{20000} X = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \]

What \( X_1 \)?
Rational solve on Trefethen’s matrix

Solution is quotient of integers having $10^5$ digits.

2002

- CRA - Jean-Guillaume Dumas, (4 days, 180 procs)
- CRA and Dixon, Hensel lifting - William Turner (est. similar effort)
- Dixon, Hensel lifting - Zhendong Wan (12.5 days, 1 proc, big mem)

2004

- Hybrid numeric-symbolic solver - Zhendong Wan (12.5 minutes, 1 proc, small mem)

Days to minutes: A factor of 1440 speedup!
Hybrid algorithms

Wan’s rational solver is an example of a numeric-symbolic hybrid applied to a symbolic problem (rational solution to linear system). Problem for integer systems is to get the numeric part to converge (numeric preconditioning issue). Poor luck so far.

The converse problem may be more important. Consider a (large, sparse) numeric linear system which is

- too large or too nasty (fill in) for sparse direct solvers.
- unresponsive to iterative methods (convergence failure).

It may prove useful to solve it exactly. Relative to numeric iterative methods the existing blackbox method is “slow but sure”.
Welker’s homology boundary matrices

Problem: Homology of simplicial complexes in dimensions up to about 10. Algebraic topology application, study of relations among matchings in graphs.

Specific computation: Smith Normal form of \{0, 1, −1\}-boundary matrices.

Solution: Jean-Guillaume Dumas code (Gap package[Dumas, Heckenback, S, Welker 2003]/LinBox) for sparse matrix Smith form. [Dumas, S, Villard 2001.]

Example: 135135 \times 270270 matrix, 5 entries per row (n^{1.14}). Smith form: 133991 one’s, 220 three’s, 924 zeroes. Time: 4 days.
Krattenthaler: "Combinatorialists love determinants"

Favorite theorem: the number of \(< \ldots >\) of size \(n\) is NICE\((n)\).

NICE formula is (roughly) hypergeometric. For many \(< \ldots >\) the \(n\)-th instance is a determinant. The nice formula arises if the determinant involves only small primes.

Example problem: Conjectured formula

\[
\det(q^{\text{maj}}_B(\sigma\pi^{-1})) = \prod_{i=1}^{n} (1 - q^{2i})^{e(i)} \prod_{i=2}^{n} (1 - q^i)^{f(i)}
\]

Done by Macsyma: \(n = 1, 2, 3, 4\) (hard) Done by LinBox: \(n = 5\), matrix size is \(2^n * n! = 3840\), entries are smallish powers of \(q\).

We then deduce \(e(i) = 2^{n-1}n!/i, f(i) = 2^nn!(i - 1)/i\)

Specific computation: Recent hybrid Smith form algorithm [Wan, S 2004]. (fastest way to get integer determinant in this case).
Krattenthaler π formula determinants

Another family of determinants generates formulas for π.
Interesting for LinBox: Matrix is very sparse with mostly small (9 digit) entries, but a few entries are very large (1000 digits).

Blackbox for $A = B + C$, where B entries are int, C entries are GMP integers $Ax = Bx + Cx$, where C has few nonzeroes and $Bx$ is fast to compute.

SNF of Krattenthaler’s 5797x5797 matrix in 6445.45sec: (1 2226, 2 430, 4 801, 8 410, 16 1590, 80 170, 160 70, 320 99, 6720 1, )

Determinant via CRA of Krattenthaler’s 5797x5797 matrix in 13050sec.
Royle’s graph adjacency matrices

Problem: Find two graphs with cospectral symmetric cubes.

Subproblem: Pairs of strongly-regular graphs with cospectral symmetric square have been found. Do any pairs of strongly-regular graphs have cospectral symmetric cubes?

Specific computation: Determinants of $A + \alpha I \mod p$, for 32548 matrices. Each of them is of order 7140 with 295680 nonzeros ($n^{1.4}$).

Ans: (Pernet and Dumas) No cospectral pairs for strongly-regular graphs with 36 or fewer vertices. Time cost: About one minute per determinant. Use blackbox determinant algorithm discussed above.
Chandler’s Toeplitz matrices

Problem: Smith forms of 0,1 Toeplitz matrices. Order about 10000. Incidence matrix of flats in projective spaces.

Richly structured Smith forms, but only one small prime occurring except in largest invariant factor. Easy for LinBox.
Lie Atlas operator signatures

General topic: understanding symmetry

Specific question: For Weyl group E8, given lie algebra operator $\alpha$ constructed in a certain way, is $\alpha$ positive (semi)definite in every irreducible representation?

There are 112 irreducible representations. The largest is as $7168 \times 7168$ rational matrices. The operator $\alpha$ maps to matrix $A$ which is dense and has entries of length about 100 digits. But also $A$ has a representation as a product of 121 very sparse matrices, each with quite small entries.

[Adams, Saunders, Wan 2005]: Obtained complete computation for an operator $\alpha_1$ of low rank (verifies a difficult recent theorem). Partial solution for an operator $\alpha_2$ of full rank. Estimation that the order 7168 representation will take 2 cpu years for this operator by current methods.
Qing Xiang’s p-rank problem

General topic: determine isomorphism (or not) of strongly regular graphs constructed in various ways.

Specific problem: Determine rank mod 3 of a matrix of order $3^e$ when rank is low (say near $2^e$). $3^{12} = 531441, 2^{12} = 4096$.

Solution: Project to a matrix of order only slightly larger than the rank, then use dense matrix methods.
Computer Algebra Niederland prize problem

“Let $A$ be a $m \times n$ matrix with coefficients in $Q(x)$, the field of rational functions in the variables $x = x_1, \ldots, x_l$. The question is how to determine efficiently the rank of the matrix $A$. ”
LinBox good at:

- rank (dimension reduction questions)
- determinant
- characteristic polynomial
- Frobenius canonical form (rational, primary forms)
- Smith normal form (invariant factors)
- rational linear system solution
- diophantine linear system solution

But needs more implementations/refinements of block blackbox methods, numeric/symbolic hybrids, attention to parallelism.
Summary

• So far, users are primarily mathematicians.

• So far, users need det, rank, characteristic polynomial, SNF, etc., not system solving specifically.

• But, it has turned out that efficient system solving is at the heart of the exact solution to other linear algebra problems (det, Smith normal form, etc.).

• LinBox aims to be the lapack for integer matrices. Can be linked into general purpose systems such as Maple, Mathematica, Matlab, Sage.

• Exact linear algebra applied to numeric problems(?)

• Move to massively parallel methods. Mod p in $O^\sim(n^{1+e})$ plus chinese remainder algorithm $\Rightarrow$ Integer problems in $O^\sim(n^{2+e})$ time. Better?
• Applications in biology (proteins are discrete sequences), in image processing (shape info via homology, inpainting), network analysis (topology again), ...