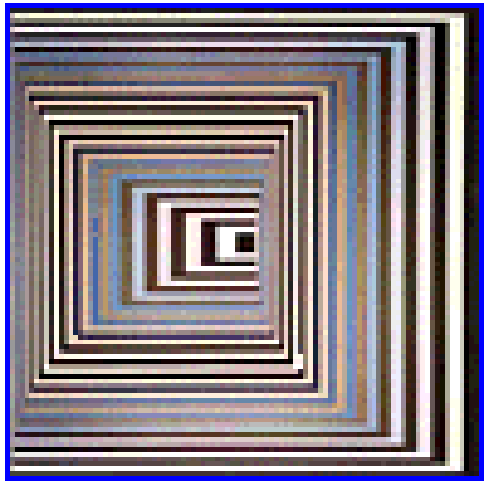
LinBox Lab – University of Delaware

D. Saunders, Z. Wan, D. Roche, C. Devore

(A. Duran, E. Schrag, R. Seagraves, B. Hovinen, ...).



Thanks to the National Science Foundation

Mirror sites are maintained at linalg.org (North America) and linalg.net (Europe). Local links: **org**, **net**.

Project LinBox: Exact computational linear algebra

LinBox is a C++ template library for exact, high-performance linear algebra computation with sparse and structured matrices over the integers and over finite fields.

No stable releases available at this time

Current development version: 0.1.3

Comments? Bug reports? Please contact us at linbox@yahoogroups.com

- **Overview**
- **News**
- **People**
- **Download**
- **Documentation**
- **Developer resources**
- **Links**
- **Support**

We offer related packages: (1) A gap share package for Simplicial Homology computation and for Smith normal forms, (2) A package for access to linbox computation from Maple.

- **GAP homology package**
- **Maple-LinBox package**

We offer a server which provides linear algebra computations including the Smith normal form of a matrix. A second server computes the full homology of simplicial complexes. Use our compute cycles gratis.

- **Online computing servers**

Comments? Bug reports? Please contact us at
linbox@yahoogroups.com
Page prepared by the LinBox team <*linbox@yahoogroups.com*>
This page's URL:
http://www.linalg.org/ (US), **http://www.linalg.net/** (Europe)
Page major version change: 4 August 2002
Page last updated: 7 March 2003
This material is based upon work supported by the National Science Foundation under grants 9726763, 9712362, 0098284, and 0112807. Any opinions, findings and conclusions or

Problems solved by LinBox

- Do exact rank, Smith form, determinant, system solve, min-poly, charpoly of integer matrices (via modular computation plus Chinese Remainder Algorithm or Hensel lifting).

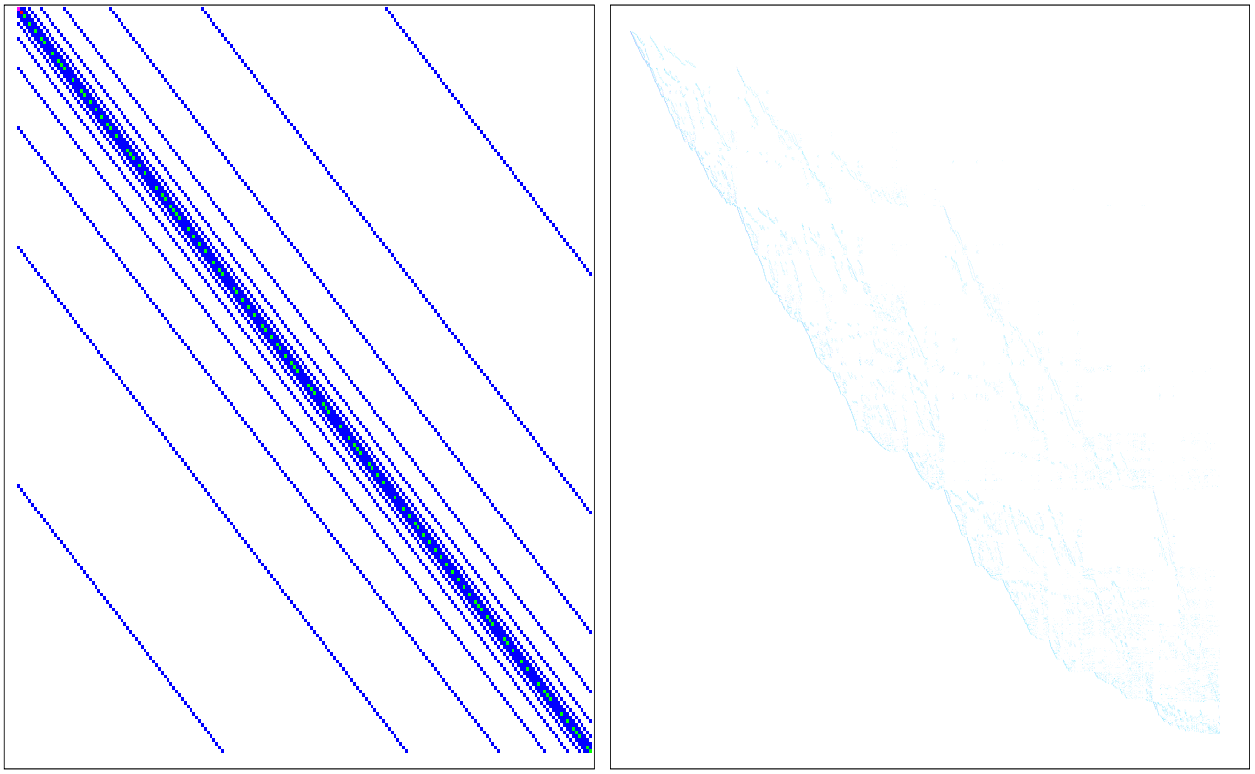
- Particularly, use rank and Smith form of $\{0, 1\}$ or $\{0, 1, -1\}$ matrices for Homology and other incidence matrix situations.

  – Homology of simplicial complexes.

  – multivariate polynomial equation system solving.

- Problems may be huge (100,000 equations, millions of nonzero entries.)

Picture of Trefethen and TF class matrices



Very sparse matrices, about $2 \log n$ non-zero entries per row in Trefethen matrices.

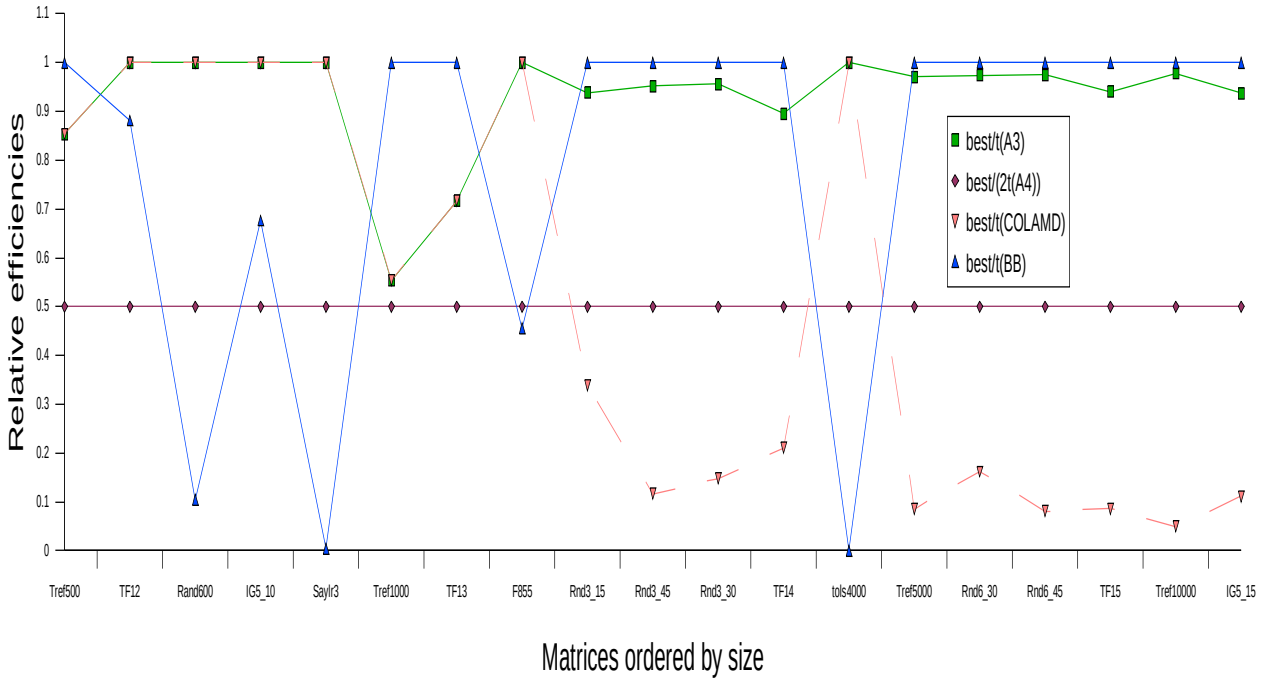
Methods

- Blackbox (BB) methods are excellent for large sparse matrices over finite fields. Wiedemann, Kaltofen-Saunders, Dumas-Saunders-Villard...

- Sparse elimination (such as SuperLU of Demmel, et al) is excellent on matrices which are small, or slow to fill in. Duran adapted it to work over finite fields.

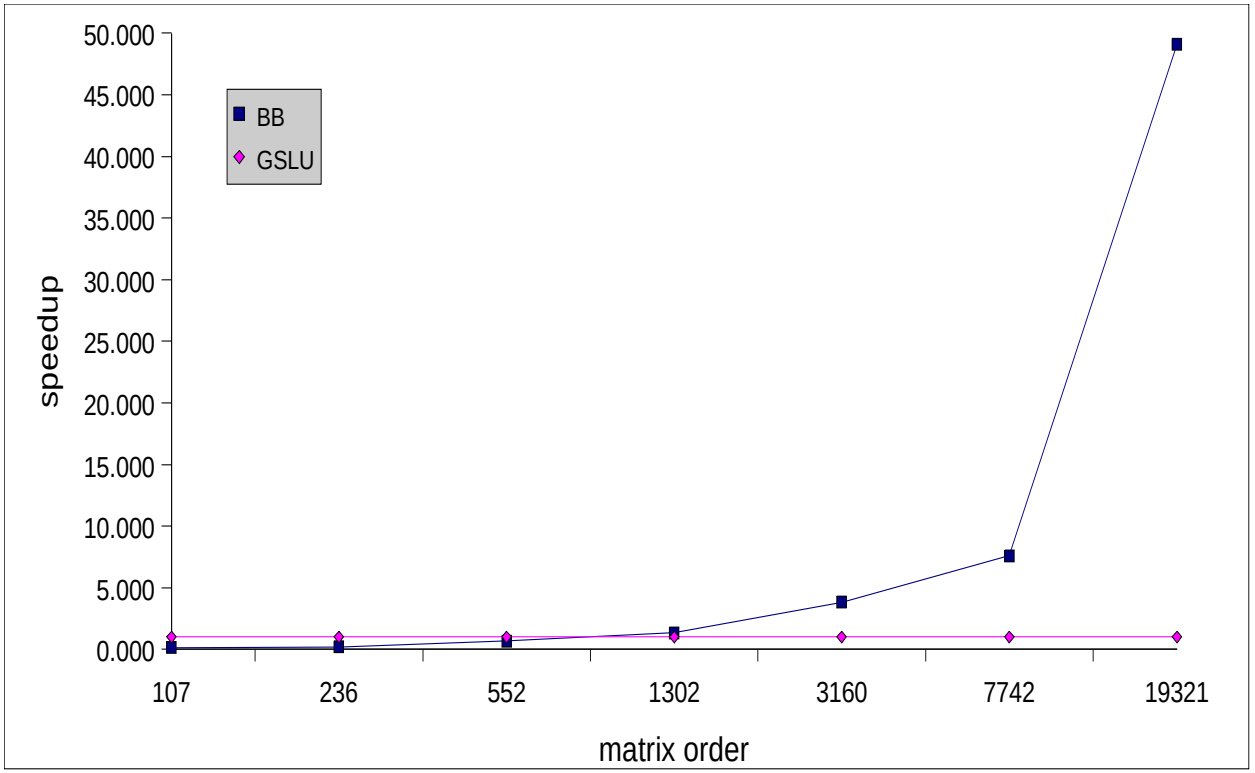
- Other eliminations are fast by using floating point BLAS.

Example 1. Engineered algorithm for rank



- Blackbox method
- Generalized SuperLU
- racing - guaranteed 1/2 efficiency of best of BB, GSLU
- hybrid - elim until BB estimate is faster

TF family



The crossover is near order 1000

(slide from Williamsburg report)

Conclusions

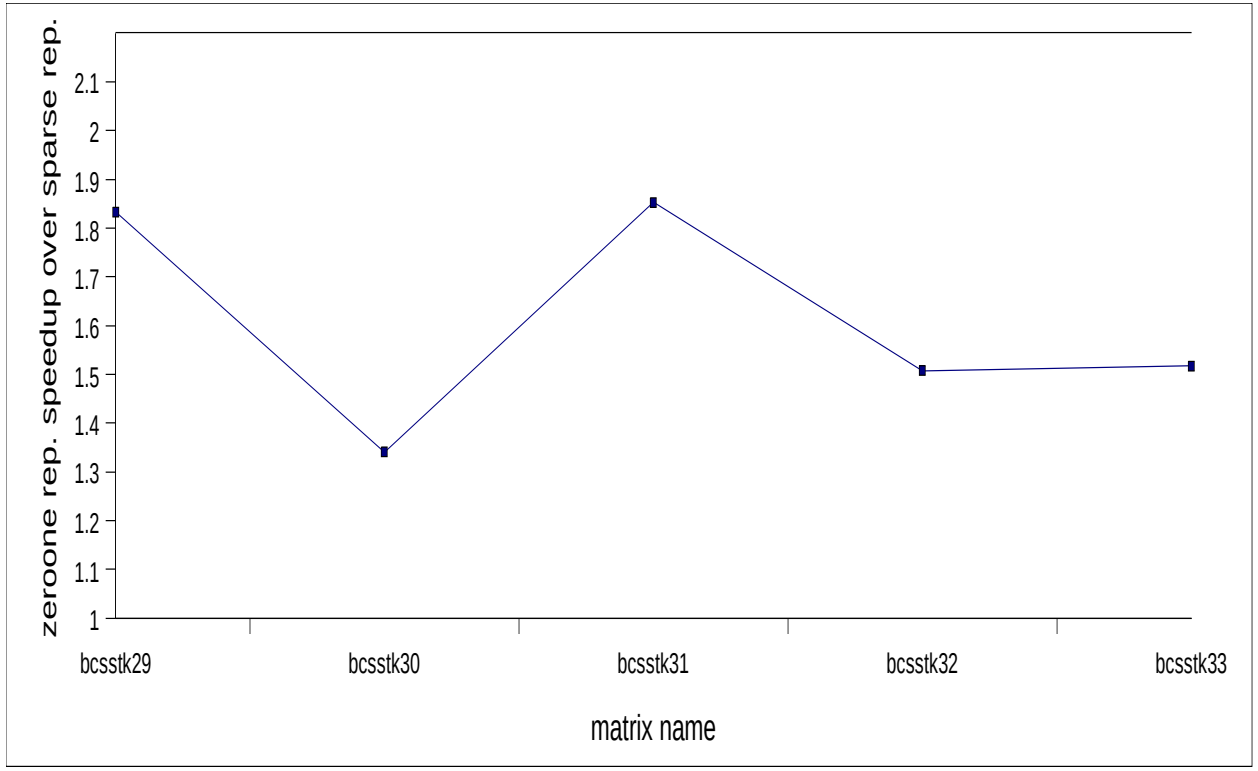
An adaptive hybrid of elimination and blackbox methods is advisable and effective for exact linear algebra over finite fields (and over the integers).

A left looking elimination such as SuperLU lends itself to early determination of excess fill-in and switch to an indirect (blackbox) method.

High performance exact linear algebra is implemented in LinBox, available at linalg.org.

Example 3: The Generic Design methodology

Speedup of ZeroOne over SparseMatrix for 32 bit prime



ZeroOne takes 2/3 as long as SparseMatrix for matrix-vector products.

Example 2. Rank of matrices of rational functions with rational number coefficients.

$$\begin{bmatrix} \frac{2x^2+7}{23x-5} & 33x^5+x+2 & \frac{x}{x^{100}-3} \\ \frac{x}{x^{100}-5} & \frac{3x^2+4}{23x-5} & 94x^4+x^3+10 \\ 3x^7+x^2-x & \frac{x}{x^{100}-8} & \frac{5x^2+1}{23x-5} \end{bmatrix}$$

...evaluated at a random point (in this example x = 1).

$$\begin{bmatrix} 1/2 & 36 & -1/2 \\ -1/4 & 7/18 & 105 \\ 3 & -1/7 & 1/3 \end{bmatrix}$$

...mod a random prime (in this example p = 11).

$$\begin{bmatrix} 6 & 3 & 5 \\ 8 & 1 & 6 \\ 3 & 3 & 4 \end{bmatrix}$$

- This is a very fast *heuristic* when p is a wordsize prime and the evaluation point is random from a sufficiently large set.

- It becomes a slower Monte Carlo algorithm with a proven upper bound on the probability of error, if sufficiently many primes and points are used.

- It becomes a very sloowww deterministic algorithm, if a really large number of points and primes are used (as calculated using formulas for bounds on determinants).

- This work won Carl Devore and me the Computer Algebra Nederland Foundation Prize - 1000 Euros.

Example 4: Quickly and exactly solve a challenge problem

In 2002, Prof. L. N. Trefethen posted "The SIAM 100-Dollar, 100-Digit Challenge".* Here is problem 7 (of 10):

> Let A be the $20,000 \times 20,000$ matrix whose entries are zero everywhere except for the primes $2, 3, 5, 7, ..., 224737$ along the main diagonal and the number 1 in all the positions a_{ij} with $|i-j| = 1, 2, 4, 8, ..., 16384$. What is the $(1,1)$ entry of A^{-1}?

*http://web.comlab.ox.ac.uk/oucl/work/nick.trefethen/hundred.html.

The 20000 by 20000 matrix has over half a million nonzero entries. The exact answer is a fraction whose numerator and denominator each has 97,389 decimal digits.

Our solutions of two years ago:

- Parallel solution by LinBoxer Jean-Guillaume Dumas (Grenoble, France): Solve mod 32 bit primes (use 12 thousand of them because of the size of the answer). Use Chinese Remainder Algorithm to combine the results. He ran 182 processors for four days using LinBox software (80 of them were the NSFRI cluster, the rest were PC's in France). This method runs in $O^{\sim}(n^4)$ time.

- A couple of months later, Zhendong Wan (Newark, Delaware) Recomputed the result on strauss using Dixon lifting. Strauss was called 'spare' then - it was in a test period before going public. Its huge memory was necessary. The method needed 8GB. This method runs in $O^{\sim}(n^3)$ time.

Zhendong's solution two years later:

- The exact answer can now be computed in 25 minutes on a cheap PC running Linux on a 1.9GHZ Pentium processor with 1GB memory (or in 12 minutes on a 3.2GHZ Intel Xeon processor). Only a few MB of memory is required. The method is a mixture of numeric approximation and symbolic exact computation. It runs in $O^{\sim}(n^2)$ time.

Methods	Complexity	Memory	Run time
Quotient of two determinants Wiedemann's algorithm Chinese remainder theorem	$O^{\sim}(n^4)$	a few MB	Four days in parallel using 182 processors, 96 Intel 735 MHZ PIII, 6 1G 20 4 $\times$ 250MHZ sun ultra-45
Solve $Ax = e_1 = (1, 0, \cdot, 0)$ by plain Dixon lifting for the dense case Rational reconstruction	$O^{\sim}(n^3)$	3.2 GB	12.5 days sequentially in a Sun Sun-Fire with 750 MHZ Ultrasparcs and 8GB for each processors
Solve $Ax = e_1 = (1, 0, \cdot, 0)$ by our methods above Rational reconstruction	$O^{\sim}(n^2)$	a few MB	25 minutes in a pc with 1.9GHZ Intel P processor, and 1 GB memory

The original work earned Zhendong a nice writeup in Trefethen's report on the contest. The new fast method earned him a place the website of a followup book about the contest. `http://www-m3.ma.tum.de/m3/bornemann/challengebook/Updates/index.html`

Future work for the LinBox team

- Theory: For the run time, best asymptotic lower bounds (problem complexity) $\neq$ best asymptotic upper bounds (algorithm complexity).
  - Design fast algorithms for general case.
  - Design fast algorithms for special matrix classes.
  - Prove *any* non-trivial lower bound.
- Practice: Best practical algorithm is determined problem size and shape, by hardware properties, by the available tools.
  - Implement and test the best algorithms.
  - Improve the library design for genericity and performance.
  - *Engineer the hybrid algorithms*.
  - Continue to provide the best performing integer matrix computation package in the world.
- Application:

- Homology - what is the geometry of huge, high dimensional, combinatorial objects?
- Graphics and medical imaging - quickly get the right shape.
- Cryptology - for instance, the RSA challenge problems.