Research Accomplishments and Objectives

Sivan Toledo

March 2002

This document presents my research accomplishments and objectives as of Spring 2002. The document references and lists most of my publications, so it also serves as an annotated list of publications. In addition, Section 8 discusses my teaching and describes a textbook that I have written.

Most of my recent and current research focuses on discrete and computer-architecture-related issues in numerical linear algebra. Section 1 describes my work on combinatorial preconditioning; Section 2 describes my work on parallel algorithms in numerical linear algebra, and Section 3 describes work on cache-efficient and out-of-core algorithms. Section 4 describes work on sparse direct linear solvers. I plan to continue this research effort in the future.

My work on algorithms in numerical linear algebra is motivated both by a desire to find techniques and solutions that are relevant to computational scientists, and by a desire to understand fundamental issues. Consequently, some of my work is theoretical, some is experimental, and some combines both. Making solutions relevant to computational scientists often requires the implementation of algorithms in the form of robust and easy-to-use algorithmic libraries. I have produced several such codes: I participated in the production of a code for IBM's Parallel Engineering and Scientific Subroutine Library, and I produced a library of out-of-core dense matrix algorithms called SOLAR, a library of sparse linear solvers called TAUCS, as well as a number of smaller codes. Except for the IBM code, the codes a publicly available. TAUCS, the most recent library, includes combinatorial preconditioners, incomplete-Cholesky preconditioners, a number of high-performance sparse direct solvers, and out-of-core sparse direct solvers. In addition to producing codes that computational scientists can use on their own, I also collaborate directly with computational physicists and chemists. For example, I recently collaborated with a computational chemist on an electronic-structure calculation code that required an out-of-core SVD algorithm.

I have also conducted research that is not related to numerical linear algebra. This includes work on parallel algorithms and parallel-programming systems, described in Section 5, work on geometric optimization that Section 6 describes, and work on digital typography that Section 7 describes. My research on digital typography resulted in another widely-used and widely deployed piece of code, one that is installed and used on most Linux desktop systems.

1 Numerical Linear Algebra: Combinatorial Preconditioning

Preconditioners are easy-to-factor (sometimes easy to invert) approximations to the coefficient matrix of a linear system of equations. An appropriate preconditioner can dramatically accelerate the convergence of many iterative linear solvers.

In 1991 Vaidya proposed a way to construct and analyze preconditioners using mainly combinatorial and graph-theoretic algorithms and analyzes. In two of my papers [1, 2, 3]

I, together with my coauthors, laid out Vaidya's theory in detail and extended it (Vaidya never formally published his results). In two other papers my PhD student Doron Chen and I experimentally investigated the behavior of combinatorial preconditioners. One paper [4, 5] investigates Vaidya's original preconditioners. Another paper [6] investigates Gremban-and-Miller's combinatorial preconditioners, as well as a range of new combinatorial preconditioners. The work on this last paper is still ongoing.

Our implementations of these combinatorial preconditioners, which are fairly complex to implement, are the only publicly available implementations of these algorithms. The implementations are part of TAUCS.

Work on other kinds of preconditioners is reported in subsequent sections.

[1] Marshall Bern, John R. Gilbert, Bruce Hendrickson, Nhat Nguyen, and Sivan Toledo. Support-graph preconditioners. Submitted to the *SIAM Journal on Matrix Analysis and Applications*, 29 pages, January 2001.

[2] Marshall Bern, John R. Gilbert, Bruce Hendrickson, Nhat Nguyen, and Sivan Toledo. Support-graph preconditioners. In *Proceedings of the Copper Mountain Conference On Iterative Methods*, page 7 unnumbered pages, Copper Mountain, Colorado, 1998.

[3] Erik G. Boman, Doron Chen, Bruce Hendrickson, and Sivan Toledo. Maximum-weight-basis preconditioners. To appear in *Numerical Linear Algebra with Applications*, 29 pages, June 2001.

[4] Doron Chen and Sivan Toledo. Implementation and evaluation of Vaidya's preconditioners. Submitted to Preconditioning 2001 to be held in Tahoe, California, 3 pages, 2001.

[5] Doron Chen and Sivan Toledo. Vaidya's preconditioners: Implementation and experimental study. Submitted to *Electronic Transactions on Numerical Analysis*, 20 pages, August 2001.

[6] Doron Chen and Sivan Toledo. Multilevel support-graph preconditioners. In *The Book of Abstracts of Latsis 2002: Iterative Solvers for Large Linear Systems*, page 36, Zurich, Switzerland, February 2002.

2 Numerical Linear Algebra: Parallel Algorithms

Some of my work on parallel numerical linear algebra, though not all of it, addressed the issue of the communication complexity of algorithms. Communication between processors is expensive on parallel computers, since the communication bandwidth between processing nodes is typically significantly slower than the local memory bandwidth and the computational rate within processing nodes. Therefore, algorithms should perform as little interprocessor communication as possible. I made two fundamental contributions to this area. One contribution is the discovery that all existing FFT algorithms perform more communication than is required in order to compute the transform to within machine precision [7, 8, 9]. With Edelman and McCorquodale, I quantified the required amount of communication as a function of the required accuracy of the transform, designed an innovative algorithm that performs near-minimal communication, and implemented and tested the algorithm. Another contribution is the discovery of communication-optimal dense linear solvers. Together with my PhD student Dror Irony, I have developed so-called three-dimensional dense linear solvers, which perform asymptotically less communication than any existing distributed-memory parallel solver [10, 11]. We also proved [12] that these solvers are optimal, in the sense that no schedules of the conventional algorithms perform asymptotically less communication. Furthermore, we proved that any communication-optimal algorithm requires as much temporary storage as our new algorithms, and that existing algorithm are optimal

when no significant temporary storage is allowed. Both of these contributions are currently only of theoretical interest, since the constants involved are rather high, so the implementations do not outperform conventional algorithms on practical problem sizes and machine sizes. However, it is plausible that in both cases the constants could be reduced, hence making the algorithms practical.

Another significant contribution to parallel numerical linear algebra is the design and implementation, together with others, of a parallel linear solver for banded systems [13, 14]. The code that we have developed as part of this project is now part of an IBM product, the Parallel Engineering and Scientific Subroutine Library.

I also worked on multicolor orderings for distributed-memory parallel incomplete-Cholesky preconditioners [15] and on parallel shared-memory fill-reducing ordering algorithms for sparse elimination [16].

More recently, together with my PhD student Dror Irony and with my MSc student Gil Shklarski, I have been working on novel implementation techniques for parallel sparse direct factorization algorithms [17]. More specifically, we have been investigating the use of a randomized dynamic scheduler and the use of recursive data layouts.

[7] Alan Edelman, Peter McCorquodale, and Sivan Toledo. The future fast fourier transform? In *Proceedings of the 8th SIAM Conference on Parallel Processing for Scientific Computing*, March 1997.

[8] Alan Edelman, Peter McCorquodale, and Sivan Toledo. The future fast Fourier transform? *SIAM Journal on Scientific Computing*, 20(3):1094–1114, 1999.

[9] Sivan Toledo. On the communication complexity of the discrete fourier transform. *IEEE Signal Processing Letters*, 3:171–172, 1996.

[10] Dror Irony and Sivan Toledo. Communication-efficient parallel dense LU using a 3-dimensional approach. In *Proceedings of the 10th SIAM Conference on Parallel Processing for Scientific Computing*, Norfolk, Virginia, March 2001. 10 pages on CDROM.

[11] Dror Irony and Sivan Toledo. Trading replication for communication in parallel distributed-memory dense solvers. Submitted to *Parallel Processing Letters*, 16 pages, July 2001.

[12] Dror Irony and Sivan Toledo. Communication lower bounds for distributed-memory matrix multiplication. Submitted to the *Journal of Parallel and Distributed Computing*, 16 pages, April 2001.

[13] Anshul Gupta, Fred G. Gustavson, Mahesh Joshi, and Sivan Toledo. The design, implementation, and evaluation of a banded linear solver for distributed-memory parallel computers. In *Proceedings of the Third International Workshop on Parallel Computing (PARA '96)*, Lecture Notes in Computer Science volume 1184, Lyngby, Denmark, August 1996. Springer-Verlag.

[14] Anshul Gupta, Fred G. Gustavson, Mahesh Joshi, and Sivan Toledo. The design, implementation, and evaluation of a symmetric banded linear solver for distributed-memory parallel computers. *ACM Transactions on Mathematical Software*, 24(1):74–101, 1998.

[15] Sivan Toledo. Preconditioning with a decoupled rowwise ordering on the CM-5. In *Proceedings of the 7th SIAM Conference on Parallel Processing for Scientific Computing*, pages 484–489, San Francisco, California, 1995.

[16] Tzu-Yi Chen, John R. Gilbert, and Sivan Toledo. Toward an efficient column minimum degree code for symmetric multiprocessors. In *Proceedings of the 9th SIAM Conference on Parallel Processing for Scientific Computing*, San-Antonio, Texas, 1999. 11 pages on CDROM.

[17] Dror Irony, Gil Shklarski, and Sivan Toledo. Parallel and fully recursive multifrontal supernodal sparse cholesky. In *Proceedings of the International Conference on Computational Science (ICCS 2002)*, pages 335–344 of Part II, Amsterdam, April 2002.

3 Numerical Linear Algebra: Locality of Reference

Nearly all computers today have hierarchical memories, in which a small fraction of the memory is fast and the rest is slower. To achieve high performance, it is essential to reduce the number of accesses to slow memory (cache misses, which are accesses to main memory, and I/O, which accesses the even slower disks). I have investigated ways to reduce cache misses (and/or I/O) in several numerical-linear-algebra algorithms.

In [18, 19], Leiserson, Rao and I developed a technique to reduce the asymptotic number of cache misses in multilevel linear relaxation, such as multigrid V cycles. The constants in this technique are quite large, so it appears to be mainly of theoretical interest. In another paper, [20][21, Chapter 10], I proposed a technique to reduce the number of cache misses in the Conjugate Gradients algorithm. The technique, which is essentially a symbolic technique for deriving multistep methods, suffers from two numerical problems that prevent it from being widely deployed.

In [22], I showed that a recursive schedule for the LU factorization with partial pivoting of a dense matrix ensures an asymptotically optimal number of cache misses. The paper also shows that the cache-efficient schedules that were used up to that point perform more cache misses, both asymptotically and absolutely. This paper has been fairly influential: it led to the development and implementation of many other recursive dense-matrix algorithms, by Gustavson, Elmroth, Jonsson, Waśniewski, Whaley, and others. In particular, ATLAS, a widely-used library uses recursive formulations of several key algorithms. I have implemented a parallel out-of-core version of the pivoting LU algorithm in the publicly-available library SOLAR [23].

In [24, 25], I investigated ways to speed up a fundamental subroutine in many linear-algebra computations, the multiplication of a vector by a sparse matrix. The paper focused on both memory-system issues and on instruction-level parallelism.

In [26], my coauthors and I used loop transformation techniques to reduce cache misses and improve performance in a histogramming code.

More recently, I have been working on I/O efficient schedules for sparse matrix factorization algorithms. In [27], Gilbert and I designed an out-of-core sparse LU with partial pivoting algorithm. The implementation of this algorithm has just been incorporated into TAUCS and will be part of the next release of the library (the code is already used by the applied physics lab in the University of Washington). Together with my MSc student Vladimir Rotkin, I have designed and implemented an out-of-core sparse Cholesky algorithm. We are still working on the paper, but the code is already included in the released version of TAUCS.

I have recently designed and implemented an out-of-core singular value decomposition, which is now part of an electronic-structure calculation code [28, 29]. The code has been used to generate over 4000 electronic states (eigenvectors) represented on a mesh with over 2 million points. To the best of our knowledge, this is the largest computation of this kind ever carried out; the out-of-core SVD has been the enabling factor in this computation.

The paper [30] surveys I/O-efficient algorithms for many problems in numerical linear algebra. The paper also presents a simple proof of a fundamental but hard-to-follow result by Hong and Kung, an I/O lower bound for matrix multiplication.

[18] Charles E. Leiserson, Satish Rao, and Sivan Toledo. Efficient out-of-core algorithms for linear relaxation using blocking covers. In *Proceedings of the 34th Symposium on Foundations of Computer Science*, pages 704–713, 1993.

[19] Charles E. Leiserson, Satish Rao, and Sivan Toledo. Efficient out-of-core algorithms for linear relaxation using blocking covers. *Journal of Computer and System Sciences*, 54(2):332–344, 1997.

[20] Sivan Toledo. Out-of-core krylov-subspace methods. In *Proceedings of the 5th Annual MIT Student Workshop on Scalable Computing*, pages 53–0–53–1, 1995.

[21] Sivan A. Toledo. *Quantitative Performance Modeling of Scientific Computations and Creating Locality in Numerical Algorithms*. PhD thesis, Massachusetts Institute of Technology, 1995. Also published as MIT Laboratory for Computer Science Technical Report MIT-LCS-TR-656.

[22] Sivan Toledo. Locality of reference in LU decomposition with partial pivoting. *SIAM Journal on Matrix Analysis and Applications*, 18(4):1065–1081, 1997.

[23] Sivan Toledo and Fred G. Gustavson. The design and implementation of SOLAR, a portable library for scalable out-of-core linear algebra computations. In *Proceedings of the 4th Annual Workshop on I/O in Parallel and Distributed Systems*, pages 28–40, Philadelphia, May 1996.

[24] Sivan Toledo. Improving instruction-level parallelism in sparse matrix-vector multiplication using reordering, blocking, and prefetching. In *Proceedings of the 8th SIAM Conference on Parallel Processing for Scientific Computing*, March 1997.

[25] Sivan Toledo. Improving memory-system performance of sparse matrix-vector multiplication. *IBM Journal of Research and Development*, 41(6):771–725, 1997.

[26] Eran Toledo, Sivan Toledo, Yael Almog, and Solange Akselrod. A vectorized algorithm for correlation dimension estimation. *Physics Letters A*, 229(6):375–378, 1997.

[27] John R. Gilbert and Sivan Toledo. High-performance out-of-core sparse LU factorization. In *Proceedings of the 9th SIAM Conference on Parallel Processing for Scientific Computing*, San-Antonio, Texas, 1999. 10 pages on CDROM.

[28] Eran Rabani and Sivan Toledo. Out-of-core SVD and QR decompositions. In *Proceedings of the 10th SIAM Conference on Parallel Processing for Scientific Computing*, Norfolk, Virginia, March 2001. 10 pages on CDROM.

[29] Sivan Toledo and Eran Rabani. Very large electronic structure calculations using an out-of-core filter-diagonalization method. To appear in the *Journal of Computational Physics*, October 2001.

[30] Sivan Toledo. A survey of out-of-core algorithms in numerical linear algebra. In James M. Abello and Jeffrey Scott Vitter, editors, *External Memory Algorithms*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, pages 161–179. American Mathematical Society, 1999.

4 Numerical Linear Algebra: Sparse Direct Methods

Direct methods solve sparse linear systems of equations by factoring the coefficient matrix into triangular (and sometimes orthogonal) factors. The desire to represent explicitly only nonzero coefficient results in difficult discrete problems related to fill minimization, parallelism, cache efficiency, and data structures.

I have already mentioned three papers that focus on sparse direct methods: the paper on parallel minimum-degree ordering algorithms, the paper on out-of-core direct factorizations, and the paper on parallel sparse Cholesky factorization.

My MSc student Igor Brainman and I investigated the use of orderings based on separators (nested dissection) to reduce fill in sparse LU factorizations with partial pivoting. The paper [31, 32, 33] shows that on many large matrices, a wide-separator-based column ordering can significantly reduce fill, memory, and factorization times. The technique was proposed as a theoretical idea in 1980 by Gilbert and Schreiber but was never before implemented or tested. Our paper essentially shows that the method is effective and practical for today's large matrices (it certainly is ineffective on even the largest matrices that could be factored in 1980).

In [34], Gilbert and I compare experimentally a sparse direct solver with state-of-the art iterative solvers for unsymmetric linear systems. More specifically, the paper compares a sparse LU with partial pivoting solver to Krylov-subspace iterative solvers with pivoting incomplete-LU preconditioners. The paper shows that direct solvers are often more efficient on problems that are considered today large. (As problems grow larger, the situation may well change.) The paper also proposes an algorithm to prevent a certain kind of structural breakdown in incomplete-LU factorizations with no fill.

[31] Igor Brainman and Sivan Toledo. Nested-dissection orderings for sparse LU with partial pivoting. In Lubin Vulkuv, Jerzy Waśniewski, and Plamen Yalamov, editors, *Proceedings of the 2nd Conference on Numerical Analysis and Applications*, volume 1988 of *Lecture Notes in Computer Science*, pages 125–132, Rousse, Bulgaria, June 2000. Springer.

[32] Igor Brainman and Sivan Toledo. Nested-dissection orderings for sparse LU with partial pivoting. In *Proceedings of the 10th SIAM Conference on Parallel Processing for Scientific Computing*, Norfolk, Virginia, March 2001. 10 pages on CDROM.

[33] Igor Brainman and Sivan Toledo. Nested-dissection orderings for sparse LU with partial pivoting. To appear in the *SIAM Journal on Matrix Analysis and Applications*, 17 pages, February 2001.

[34] John R. Gilbert and Sivan Toledo. An assessment of incomplete-lu preconditioners for non-symmetric linear systems. *Informatica*, 24:409–425, 2000.

5 Parallel Systems and Algorithms

As a PhD student, I spent some of my time working on parallel architectures and parallel-programming systems. This work consisted of research on interconnection networks [35, 36] and of research on automatic performance prediction of parallel programs in data-parallel programming languages [37, 38].

More recently, I worked with an MSc student, Yaron Shoham, on a novel parallel algorithm for searching game trees (e.g., for Chess-playing programs) [39]. The algorithm uses randomization to improve an earlier deterministic algorithm and to generate parallelism.

[35] Sivan Toledo. Competitive fault tolerance in area-universal networks. In *Proceedings of the 4th ACM Symposium on Parallel Algorithms and Architectures*, pages 236–246, 1992.

[36] Sivan Toledo. Space sharing a scan network. In *Proceedings of the 1993 MIT Student Workshop on Supercomputing Technologies*, pages 44–0–44–1, 1993.

[37] Sivan Toledo. Perfsim: A tool for automatic performance analysis of data-parallel fortran programs. In *Proceedings of the 5th Symposium on the Frontiers of Massively Parallel Computation*, pages 396–405, 1995.

[38] Sivan Toledo. Performance prediction with benchmaps. In *Proceedings of the 10th International Parallel Processing Symposium*, pages 479–484, Honolulu, Hawaii, April 1996.

[39] Yaron Shoham and Sivan Toledo. Parallel randomized best-first search. *Artificial Inteligence*, 137:165–196, 2002.

6 Geometric Optimization

As an MSc and PhD student I worked on geometric-optimization algorithms. Some of my work in this area focused on applying a technique due to Megiddo to the solution of geometric-optimization problems [40, 41, 42, 43, 44, 45, 46]. A typical problem in this area is to find the largest copy of a polygon with a given shape that fits into another polygon. Another part of my work focused on extending Megiddo's technique in various ways, such as applying it to solve nonlinear problems and to solve hard problems approximately [47, 48, 49, 50].

[40] Sivan Toledo. Extremal polygon containment problems. In *Proceedings of the 7th ACM Symposium on Computational Geometry*, pages 176–185, 1991.

[41] Pankaj K. Agarwal, Micha Sharir, and Sivan Toledo. Applications of parametric searching in geometric optimization. In *Proceedings of the 3rd ACM-SIAM Symposium on Discrete Algorithms*, pages 72–82, 1992.

[42] Pankaj K. Agarwal, Alon Efrat, Micha Sharir, and Sivan Toledo. Computing a segment-center for a planar point set. *Journal of Algorithms*, 15:314–323, 1993.

[43] Klara Kedem, Micha Sharir, and Sivan Toledo. On critical orientations in the kedem-sharir motion planning algorithm for a convex polygon in the plane. In *Proceedings of the 5th Canadian Conference on Computational Geometry*, pages 204–209, 1993.

[44] Pankaj K. Agarwal, Micha Sharir, and Sivan Toledo. Applications of parametric searching in geometric optimization. *Journal of Algorithms*, 17:292–318, 1994.

[45] K. Kedem, M. Sharir, and S. Toledo. On critical orientations in the Kedem-Sharir motion planning algorithm. *Discrete and Compututational Geometry*, 17:227–239, 1997.

[46] Micha Sharir and Sivan Toledo. Extremal polygon containment problems. *Computational Geometry Theory and Applications*, 4:99–118, 1994.

[47] Sivan Toledo. Maximizing non-linear concave functions in fixed dimensions. In *Proceedings of the 33th Symposium on Foundations of Computer Science*, pages 676–685, 1992.

[48] Pankaj K. Agarwal, Micha Sharir, and Sivan Toledo. An efficient multi-dimensional searching technique and its applications. Technical Report CS-1993-20, Duke University, 1993.

[49] Sivan Toledo. Maximizing non-linear concave functions in fixed dimensions. In Panos M. Pardalos, editor, *Complexity in Numerical Computations*, pages 429–447. World Scientific, 1993.

[50] Sivan Toledo. Approximate parametric searching. *Information Processing Letters*, 47:1–4, 1993.

7 Digital Typography

I have some work on computer typesetting [51, 52, 53, 54]. The code that I produced in the context of this research [54] is now part of virtually all the major Linux distributions, and it is used every time a user prints a document containing text from a KDE application (e.g., Kmail) or from a Qt application (e.g., the commercial web browser Opera).

[51] Sivan Toledo. A simple technique for typesetting hebrew with vowel points. *TUGBoat*, 20(1):15–19, 1999.

[52] Sivan Toledo. Exploiting rich fonts. *TUGBoat*, 21(2):121–129, 2000.

[53] Sivan Toledo. Typesetting Hebrew with LaTeX. *Eutupon*, 6:39–56, April 2001.

[54] Sivan Toledo and Lars Knoll. Font subsetting and downloading in the PostScript printer driver of Qt/X11. In *Proceedings of the XFree86 Technical Conference*, Oakland, California, November 2001. USENIX.

8 A Few Words on Teaching and on a Textbook

Since I arrived in Tel-Aviv three and a half years ago I have taught one core CS course, Operating Systems, and two elective courses on parallel computing and high-performance numerical linear algebra. I have taught the operating-systems course and the parallel-computing course four times each, and I have taught the high-performance numerical linear algebra course twice.

The basic high-performance computing course focuses on three core issues: parallel architectures, parallel programming, and parallel and cache-efficient algorithms. The parallel architecture part of the course covers topics that are relevant to algorithm designers and implementors, primarily cache coherence protocols and synchronization mechanisms. The parallel programming part teaches distributed-memory programming using MPI and shared-memory programming using both threads and Cilk (an experimental language). The algorithmic part of the course focuses on dense-matrix algorithms and on sorting, with theoretical and experimental analyzes of both parallelism and cache efficiency. The students solve several programming assignments that focus on both distributed-memory parallel programming, on shared-memory parallel programming, and on improving cache-efficiency.

The advanced high-performance computing course focuses on advanced algorithms in numerical linear algebra. The algorithms that the course covers include high-performance formulations of the FFT, sparse direct methods (including multifrontal and supernodal methods), and fill-reducing orderings.

While teaching the Operating Systems undergraduate course I have discovered that students suffer from a lack of reference material. Although there are several excellent English-language textbooks on the subject, I have found that reading an English-language textbook is too difficult for many of the students; as a consequence, many of them do not consult a textbook, even when one is recommended. Therefore, many of them do not perform as well as they could. Since my mission is to teach them the subject matter (as opposed to ensuring fluency in technical English), I have decided to write a Hebrew-language textbook [55], which was published in Fall 2001.

[55] Sivan Toledo. *Operating Systems*. Akademon, Jerusalem, 2001. 209 page, In Hebrew.