## RESEARCH HIGHLIGHTS

The results of my research have been published in 3 books and over 230 research articles in journals and refereed proceedings of conferences and have been surveyed in my 18 surveys in journals and book chapters, including three articles in SIAM Review, of 1984, 1992, and 1997 (see the list of my publications). In the description below I use some abbreviations listed at the end.

**1. Polynomial evaluation 1959-1966.**

a) In 1962 I introduced the active operation/linear substitution method to prove the optimality of Horner's classical algorithm for polynomial evaluation. This solved A.M. Ostrowski's problem of 1955. The method was used by V. Strassen and S. Winograd for proving various other fundamental lower bounds in algebraic computing.

b) I have accelerated polynomial evaluation by using preconditioning.

The work is surveyed in my paper in Russian Math. Surveys, 1966, and in the book by D.E. Knuth, The Art of Computer Programming, volume 2, Addison-Wesley, 1998 (third edition).

**2. Fast matrix multiplication (MM).**

In 1978 I decreased Strassen's MM exponent 2.807 of 1968-69 based on my techniques of trilinear aggregating of 1972. Decreasing the exponent was a challenge that defied a decade of worldwide effort. The effort was due to the fundamental role of MM in the theory and practice of computing. Currently trilinear aggregating is the basis for the fastest known algorithms for n-by-n MM for all reasonable n>19. The algorithms from my papers in SIAM Review of 1984 and (with Laderman and Sha) in LAA 1992 were implemented by Igor Kaporin (see his paper in TCS '04). Trilinear aggregating is also a part of the fastest known algorithms for n-by-n MM for immense n (see JSC'90). The work (up to 1984) is surveyed in my article in SIAM Review of 1984 and my book in Springer Lecture Notes in Computer Science, volume 179, Berlin, 1984.

The current record computational complexity estimates for fast rectangular matrix multiplication were reached in my papers with X. Huang (Proc. PASCO '97 and J. of Complexity of 1998), where the estimates were also extended to similar record estimates for polynomial factoring in finite fields and parallel evaluation of the determinant of a matrix, based on my work with Z. Galil (IPL '89).

Trilinear aggregating was a novel approach to algorithm design. It begins with restating the two-dimensional matrix problem as the problem of decomposing a three-dimensional tensor. Then proper manipulation with the indices of this tensor leads to lower rank decompositions and effective algorithms. This was the first example where such a back and forth transition between matrix and tensors defined effective algorithms for some fundamental matrix computations. By now this became a highly important tool of the algorithm design in matrix computations, far beyond the area of matrix multiplication.

**3. Multigrid and Algebraic multigrid methods.** This popular field evolved from my paper with W. L. Miranker in LAA '80, which introduced linear aggregation/disaggregation processes. Then I left the field of multigrid methods until 1990-93 when I published four papers with J.H. Reif in Proc. SPAA'90, SIAM J. on Scientific and Stat. Computing in 1992 and in CAMWA '90 and 93, where a simple but novel acceleration technique was proposed.

**4. Parallel computation.**

a) Processor efficient NC and RNC algorithms for matrix computations with extensions. My paper with

J. H. Reif (Proc. STOC ‘85) on parallel solution of linear systems of equations was reviewed in the Science, Science News, and Byte magazines in 1985/86. It focused on processor efficiency of fast parallel algorithms in NC and RNC, which was an innovation in theoretical computer science. The resulting algorithm for sparse linear systems of equations (SICOMP’93) was implemented on the supercomputers of NASA and Thinking Machines. Independently, I introduced the techniques of variable diagonal, which enabled the fast and processor efficient solution of linear systems of equations and of some other fundamental problems of matrix computations with integer input (Proc. FST & TCS ‘85, TCS ‘87, IPL’89, and SICOMP ‘00). I also proposed some distinct techniques (by myself in SPAA’90, with D. Bini and L. Gemignani in ICALP’91, and with E. Kaltofen, in Proc. SPAA ’91 and Proc. FOCS ’92) for solving general and structured linear systems of equations over abstract fields; this has resolved a long standing theoretical challenge. I also extended this approach to fast and processor efficient computation of polynomial gcds, lcms and Pade approximations (TCS’96).

b) Graph algorithms. I obtained processor efficient and fast parallel algorithms for the computation of matchings and paths in graphs based on improving the known non-trivial reductions to matrix computations and on the extension of the known methods for matrix computations (Proc. FOCS ‘85 and Combinatorica of 1988 with Z. Galil, JCSS ‘89 and IPL ‘91 with J.H. Reif, my Chapter in the Handbook on Computer Science 1993, Proc. SPAA ‘92 and SICOMP ‘95 with F. P. Preparata, and Proc. SPAA ‘92 and Algorithmica of 1997 with Y. Han and J.H. Reif).

c) Polynomial division. I obtained the fastest processor efficient algorithms with D. Bini (Proc. FOCS ‘92 and SICOMP ‘93).

d) Hard theoretical problem of the NC-equivalence of the integer gcd and planar integer linear programming problems was solved by using novel techniques in Proc. SODA ‘92 (with Y. Lin-Kriz), Proc. FOCS ‘93 and SICOMP ‘98 (with D. Shallcross and Y. Lin-Kriz).

**5. Symbolic-Numerical Computations (SNC).**

In this large and highly important area, numerical techniques are applied to problems of symbolic computation and vice versa symbolic techniques are applied to problems of numerical computation, to the benefit of both areas. In my research and survey papers since 1991 and in two books of 1994 (with D. Bini) and of 2001, I consistently demonstrated the advantages of this kind in computations with structured matrices and in their applications to polynomial computations. In this way the highly developed numerical techniques of matrix computations can be effectively applied to the problems of polynomial computations, traditionally considered symbolic. Furthermore, both numerical and symbolic techniques can be combined for computations with structured matrices. Besides the books, see my survey articles in SIAM Review 1992 and 1997, in NATO ASI Series published by Springer 1991, Academic Press 1992, and Kluwer 1998, and in electronic proceeding of IMACS/ACA 1998, also the relevant papers in parts 6-12 below.

**6. Root-finding for a polynomial and a system of polynomials. Extensions to approximate polynomial gcd.**

a) The celebrated problem of univariate polynomial root-finding was central in mathematics and applied mathematics for about four millennia. It is still highly important to computer algebra. My algorithms (Proc. STOC ‘95/CAMWA ‘96, and Proc. ISSAC ‘01/JSC ‘02) improved the record estimates for arithmetic and Boolean complexity of the solution. The improvement (by the order of magnitude) relied on combining various known methods, in particular by A. Schönhage, P. Kirrinnis, J. Renegar, C.A. Neff and J.H. Reif, with new recursive techniques of lifting/descending and exploiting Padé approximation for numerical stabilization. The algorithms yield a nearly optimal solution (with both arithmetic and Boolean time-cost within polylogarithmic factors from the information lower bounds) and allowed processor efficient NC parallelization to run in polylogarithmic arithmetic and Boolean time. The novel techniques of utilizing Graeffe's iteration had several further applications. This work (up to 1997) is surveyed in SIAM Review of 1997. Furthermore, in 1996, 2000, 2005 and 2007, by myself and with coauthors, I presented other works on polynomial root-finding in J. of Complexity 1996 (two papers) and 2000 (two papers), in CAMWA 2005 (two papers), and in the ACM Proc. of SNC'2007.

b) With D.A. Bini and L. Gemignani, first in CAMWA '04 and then in ETNA'04 and Numerische Mathematiks '05, I proposed a novel distinct approach based on exploiting the structure of the associated companion and generalized companion matrices. Using matrix methods enabled numerically stable solution, while keeping arithmetic cost at a nearly optimum level due to using matrix structure. This work is surveyed in my chapter (with coauthors from CUNY) in the SNC volume of 2007 (published by Birkhauser). I further extended these algorithms to solving the eigen-problem for general matrix in Proc. SODA'05 and in CAMWA'07. I extended this work in 2009 by incorporating my novel preprocessing techniques into the inverse iteration for eigen-solving applied to companion and generalized companion matrices.

c) Polynomial root-finding algorithms were combined in Proc. SODA '98 and Information and Computation of 2001 with some graph algorithms in application to computing <u>approximate polynomial gcd</u>; also a new approach based on using Padé approximation was proposed in these papers;

d) A series of papers with B. Mourrain and with I.Z. Emiris have introduced and developed several novel techniques and algorithms for approximating <u>the roots of dense and sparse systems of multivariate polynomials</u>. The algorithms exploited the structure of the associated matrices and enabled substantial improvement of the known upper bounds on the time and memory space complexity. Sources: B. Mourrain, V.Y. Pan, J. of Complexity of 2000 (awarded by this Journal as one of its two best papers for 2000); I.Z. Emiris, V.Y. Pan, JSC '02 , Proc. CASC '03, J. of Complexity '05, and the references therein.

**7. <u>Algorithms for structured matrices</u>.** This is a highly important area for both theory and practice of computations for sciences, engineering, and signal and image processing. The subject was studied already in the 19th century and with increased intensity in the recent years. Here is a partial list of my contributions (also see the Reference below, the bibliography therein, and the list of my publications).

a) My general <u>method of displacement transformation</u>, (Proc. ISSAC '89 and MC '90) readily extends successful algorithms from one class of structured matrices to many other classes (e.g., the extension is readily obtained in all directions among Cauchy-like, Toeplitz-like, and Vandermonde-like matrices); this method was the basis of the GKO practical algorithm by Gohberg, Kailath and Olshevsky (MC '95) for Toeplitz and Toeplitz-like linear systems, which in turn has become the basis for a number of further
Advanced algorithms..

b) I supported <u>Newton-structured matrix iteration</u> and other structured residual correction methods with non-trivial techniques of recursive compression (also called recompression) of structured matrices (see part d below) and novel initialization techniques. The resulting superfast algorithms, allowing effective parallelization, have been unified over various classes of structured matrices.

c) With V. Olshevsky in Proc. FOCS '98, I proposed a "superfast" (that is running in nearly linear time) <u>divide-and-conquer algorithm</u> that was unified over various classes of structured matrices (also see my book of 2001 and the references therein)

d) <u>Effective algorithms for compression of the displacements of structured matrices were presented in CAMWA'92, SIMAX'93, and Section 4.6 of the book of 2001. These algorithms are required to</u>

supports fast algorithms for the compression of the displacements to their rank level. This operation is required in parts b) and c).

e) New, improved algorithms for various fundamental computations with structured matrices (solving linear systems of equations and the eigenproblem, computing the rank, a basis for the null space, and the characteristic and minimum polynomials).
f) The extension of successful methods for computations with structured matrices to some fundamental polynomial and rational computations.
g) Inversion of displacement operators (with X. Wang) as a basis for decompression of structured matrices from their displacements.

5. **References:** V.Y. Pan, Structured Matrices and Polynomials: Unified Superfast Algorithms, Birkhäuser/Springer, Boston/New York, 2001, and the bibliography therein. On Newton's iteration, also see my coauthored papers in LAA'02, TCS'04, Numerical Algorithms '04, and MC '06, and my Paper of 2008 in MATRIX METHODS: THEORY, ALGORITHMS AND APPLICATIONS and the bibliography therein. See part 6 above on using matrix structure on root-finding for polynomials and systems of polynomials. See CAMWA'93 and Annals of Numerical Math. 1997 on application to polynomial evaluation.

**8, Computation of the sign and the value of the determinant of a matrix.** (Also see part 9 below.) The problem is important for geometric computations.

6. a) Two novel algorithms appeared in Proc. ACM Annual Conference on Computational Geometry of 1997 and TCS '99 (with M. Bronnimann, I.Z. Emiris and S. Pion) and in Proc. SODA '99 and Algorithmica of 2001 (with Y. Yu). They perform with single or double precision (algebraically in TCS and numerically in Algorithmica), use small arithmetic time, and certify the output. The TCS paper also proposes some output sensitive technique and randomization to accelerate the computations.

7. b) My algorithms in TCS'87 (Appendix) and IPL'88 reduce the computation of the determinant to the solution of linear systems of equations and then apply p-adic lifting to yield the solution rapidly. These techniques have eventually evolved into one of the fastest practical symbolic algorithms for the computation of the determinant of a matrix (see J. Abbott, M. Bronstein, T. Mulders in Proc. ISSAC'99 and W. Eberly, M. Giesbrecht, G. Villard in Proc. FOCS '00).

**9. p-adic lifting for linear systems and determinants.** I proposed and analyzed nearly optimal exact solution algorithms for sparse and structured (e.g., Toeplitz and Toeplitz-like) systems of linear equations with integer coefficients. My approach relies on p-adic lifting combined with the divide-and-conquer algorithms (Proc. ISSAC '02 and CASC '02). I further extended these algorithms to rings instead of fields to initialize lifting modulo a power of two instead of modulo a random prime. This allowed more effective implementation. In SNC'09 I initialized Lifting by using numerical iterative refinement.
I also applied p-adic lifting to accelerate the bottleneck stage of block Hankel computations in the bloc Wiedemann algorithm for computing determinants. See Zapiski Nauchnykh Seminarov POMI 2004 (in English).

10. **Error-free Computations via Floating-Point Operations.** In CAMWA'09, with coauthors from CUNY, I contributed to the error-free computations performed with floating-point operations. I proposed a novel summation algorithm of this class based on modular reduction of finite and infinite binary numbers. The latter idea came from the algorithms in my technical report in ICSI '92, further refined and tested with I.Z. Emiris and Y.Yu (see JSC '98).

11. **<u>The symmetric tridiagonal eigenproblem and real polynomial roots</u>.** The first algorithm that computed all eigenvalues of a real symmetric tridiagonal matrix using nearly linear arithmetic time was proposed and elaborated upon in 1990 (with D. Bini) and published in some detail in January 1991 in Proc. SODA'91. It has journal versions in Computing 1992 and SICOMP '98. In SICOMP'98 the algorithm was also extended to nearly optimal root-finding for polynomials with only real roots.

**<u>Randomized preprocessing</u>.** This was my main subject since 2005. Multiplicative preconditioning is an important, popular and highly developed tool for solving linear systems of equations, but only for special classes of input matrices readily factorized or having certain properties of their singular values. I proposed novel techniques of randomized preprocessing that largely relax this restriction fo. The techniques handle both degeneracy and ill conditioning and have led me to new effective algorithms for computing vectors in the null space of a matrix, approximating a matrix by a nearby matrix having a smaller rank or a smaller displacement rank, and approximating eigenvectors, eigenspaces and singular spaces of a matrix, with further applications to root-finding for polynomial and secular equations and to solving linear systems of equations. According to extensive analysis and experiments, the approach substantially accelerate some most fundamental computations with matrices and polynomials. In the case of structured (e.g., Toeplitz) input the acceleration is dramatic both in terms of estimated arithmetic time and observed CPU time. So far this work is presented in 8 papers (in LAA (2), CAMWA (2), Proceedings of SNC 2007 (2) , SCR 08 (1) and SNC'09 (1 invited) and in a number of Technical Reports of the Computer Science Doctoral Program of the Graduate Center of CUNY in 2005-09