Foundations and Trends® in
Theoretical Computer Science
Vol. 9, No. 2 (2013) 125–210

DOI: 10.1561/0400000065

now
the essence of knowledge

Faster Algorithms via Approximation Theory

Sushant Sachdeva
Yale University
sushant.sachdeva@yale.edu

Nisheeth K. Vishnoi
Microsoft Research
nisheeth.vishnoi@gmail.com

Contents

Abstract

This monograph presents ideas and techniques from approximation theory for approximating functions such as x^s, x^{-1} and e^{-x}, and demonstrates how these results play a crucial role in the design of fast algorithms for problems which are increasingly relevant. The key lies in the fact that such results imply faster ways to compute primitives such as $A^s v$, $A^{-1}v$, $\exp(-A)v$, eigenvalues, and eigenvectors, which are fundamental to many spectral algorithms. Indeed, many fast algorithms reduce to the computation of such primitives, which have proved useful for speeding up several fundamental computations such as random walk simulation, graph partitioning, and solving systems of linear equations.

S. Sachdeva and N. K. Vishnoi. *Faster Algorithms via Approximation Theory*. Foundations and Trends® in Theoretical Computer Science, vol. 9, no. 2, pp. 125–210, 2013.

DOI: 10.1561/0400000065.

Introduction

A Brief History of Approximation Theory

The area of approximation theory is concerned with the study of how well functions can be approximated by simpler ones. While there are several notions of *well* and *simpler*, arguably, the most natural notion is that of *uniform* approximations by *polynomials*: Given a function $f : \mathbb{R} \mapsto \mathbb{R}$ and an interval $\mathcal{I}$, what is the closest a degree d polynomial can remain to $f(x)$ throughout the entire interval? Formally, if Σ_d is the class of all univariate real polynomials of degree at most d, the goal is to understand

$$\varepsilon_{f,\mathcal{I}}(d) \stackrel{\text{def}}{=} \inf_{p \in \Sigma_d} \sup_{x \in \mathcal{I}} |f(x) - p(x)|.$$

This notion of approximation, called uniform approximation or Chebyshev approximation, is attributed to Pafnuty Chebyshev, who initiated this area in an attempt to improve upon the *parallel motion* invented by James Watt for his steam engine; see [13]. Chebyshev discovered the *alternation property* of the best approximating polynomial and found the best degree-$d-1$ polynomial approximating the monomial x^d; see [14]. Importantly, the study of this question led to the discovery of, what are now referred to as, Chebyshev polynomials (of the first kind). Chebyshev polynomials find applications in several different areas of science and mathematics and, indeed, repeatedly make an appearance in this monograph due to their extremal properties.[1]

[1] The Chebyshev polynomial of degree-d is the polynomial that arises when one writes $\cos(d\theta)$ as a polynomial in $\cos\theta$.

Despite Chebyshev's seminal results in approximation theory, including his work on best rational approximations, several foundational problems remained open. While it is obvious that $\varepsilon_{f,\mathcal{I}}(d)$ cannot increase as we increase d, it was Weierstrass [67] who later established that, for any continuous function f and a bounded interval $\mathcal{I}$, the error $\varepsilon_{f,\mathcal{I}}(d)$ tends to 0 as d goes to infinity. Further, it was Emile Borel [11] who proved that the best approximation is always *achieved* and is *unique*. Among other notable initial results in approximation theory, A. A. Markov [38], motivated by a question in chemistry posed by Mendeleev, proved that the absolute value of the derivative of a degree d polynomial that is bounded in absolute value by 1 in the interval $[-1,1]$ cannot exceed d^2. These among other results not only solved important problems motivated by science and engineering, but also significantly impacted theoretical areas such as mathematical analysis in the early 1900s.

With computers coming into the foray around the mid 1900s, there was a fresh flurry of activity in the area of approximation theory. The primary goal was to develop efficient ways to calculate mathematical functions arising in scientific computation and numerical analysis. For instance, to evaluate e^x for $x \in [-1,1]$, it is sufficient to store the coefficients of its best polynomial (or rational) approximation in this interval. For a fixed error, such approximations often provided a significantly more succinct representation of the function than the representation obtained by truncating the appropriate Taylor series.

Amongst this activity, an important development occurred in the 1960s when Donald Newman [43] showed that the best degree-d rational approximation to the function $|x|$ on $[-1,1]$ achieves an approximation error of $e^{-\Theta(\sqrt{d})}$, while the best degree-d polynomial approximation can only achieve an error of $\Theta(1/d)$. Though rational functions were also considered earlier, including by Chebyshev himself, it was Newman's result that revived the area of uniform approximation with rational functions and led to several rational approximation results where the degree-error trade-off was exponentially better than that achievable by polynomial approximations. Perhaps the problem that received the most attention, due to its implications to numerical methods for solving systems of partial differential equations (see [19]), was to understand the best rational approximation to e^{-x} over the interval $[0,\infty)$. Rational functions of degree d were shown to approximate e^{-x} on $[0,\infty)$ up to an error of c^d for some constant $c < 1$. This line of research culminated in a land-

mark result in this area by Gonchar and Rakhmanov [20] who determined the optimal c. Despite remarkable progress in the theory of approximation by rational functions, there seems to be no clear understanding as to why rational approximations are often significantly better than polynomial approximations of the same degree, and surprising results abound. Perhaps this is what makes the study of rational approximations promising and worth understanding.

Approximation Theory in Algorithms and Complexity

Two of the first applications of approximation theory in algorithms[2] were the Conjugate Gradient method (see [24, 31]) and the Lanczos method (see [36]), which are used to solve systems of linear equations $Ax = v$ where A is an $n \times n$ real, symmetric, and positive semi-definite (PSD) matrix. These results, which surfaced in the 1950s, resulted in what are called *Krylov subspace methods* and can also be used to speed up eigenvalue and eigenvector computations. These methods are iterative and reduce such computations to a small number of computations of the form Au for different vectors u. Thus, they are particularly suited for sparse matrices that are too large to handled by Gaussian elimination-based methods; see the survey [58] for a detailed discussion.

Until recently, the main applications of approximation theory in theoretical computer science have been in complexity theory. One of the most notable was by Beigel *et al.* [8] who used Newman's result to show that the complexity class PP is closed under intersections and unions.[3] Another important result where approximation theory, in particular Chebyshev polynomials, played a role is the *quadratic* speed-up for quantum search algorithms, initiated by a work by Grover [22]. The fact that one cannot speed up beyond Grover's result was shown by Beals *et al.* [7] which, in turn, relied on the use of Markov's theorem as inspired by Nisan and Szegedy's lower bound for the Boolean OR function [46]. For more on applications of approximation theory to complexity theory, communication complexity and computational learning theory, we refer the reader to [1, 33, 61, 65], and for applications to streaming algorithms to [23].

[2] More precisely, in the area of numerical linear algebra.

[3] PP is the complexity class that contains sets that are accepted by a polynomial-time bounded probabilistic Turing machine which accepts with probability strictly more than $1/2$.

Faster Algorithms via Approximation Theory

The goal of this monograph is to illustrate how classical and modern techniques from approximation theory play a crucial role in obtaining results that are relevant to the emerging theory of fast algorithms. For example, we show how to compute good approximations to matrix-vector products such as $A^s v$, $A^{-1}v$ and $\exp(-A)v$ for any matrix A and a vector v.[4] We also show how to speed up algorithms that compute the top few eigenvalues and eigenvectors of a symmetric matrix A. Such primitives are useful for performing several fundamental computations quickly, such as random walk simulation, graph partitioning, and solving linear system of equations. The algorithms for computing these primitives perform calculations of the form Bu where B is a matrix closely related to A (often A itself) and u is some vector. A key feature of these algorithms is that if the matrix-vector product for A can be computed quickly, e.g., when A is sparse, then Bu can also be computed in essentially the same time. This makes such algorithms particularly relevant for handling the problem of *big data*. Such matrices capture either numerical data or large graphs, and it is inconceivable to be able to compute much more than a few matrix-vector product on matrices of this size.

Roughly half of this monograph is devoted to the ideas and results from approximation theory that we think are central, elegant, and may have wider applicability in TCS. These include not only techniques relating to polynomial approximations but also those relating to approximations by rational functions and beyond. The remaining half illustrates a variety of ways we can use these results to design fast algorithms.

As a simple but important application, we show how to speed up the computation of $A^s v$ where A is a symmetric matrix with eigenvalues in $[-1,1]$, v is a vector and s is a large positive integer. The straightforward way to compute $A^s v$ takes time $O(ms)$ where m is the number of non-zero entries in A, *i.e.*, A's *sparsity*. We show how, appealing to a result from approximation theory, we can bring this running time down to essentially $O(m\sqrt{s})$. We start with a result on polynomial approximation for x^s over the interval $[-1,1]$. Using some of the earliest results proved by Chebyshev, it can be shown that

[4] Recall that the matrix exponential is defined to be $\exp(-A) \stackrel{\text{def}}{=} \sum_{k\geq 0} \frac{(-1)^k A^k}{k!}$.

there is a polynomial p of degree $d \approx \sqrt{s \log 1/\delta}$ that δ-approximates x^s over $[-1,1]$. Suppose $p(x)$ is $\sum_{i=0}^{d} a_i x^i$, then the candidate approximation to $A^s v$ is $\sum_{i=0}^{d} a_i A^i v$. The facts that all the eigenvalues of A lie in $[-1,1]$, and that p is close to x^s in the entire interval $[-1,1]$ imply that $\sum_{i=0}^{d} a_i A^i v$ is close to $A^s v$. Moreover, the time taken to compute $\sum_{i=0}^{d} a_i A^i v$ is easily seen to be $O(md) = O(m\sqrt{s \log 1/\delta})$, which gives us a saving of about $\sqrt{s}$.

When A is the random walk matrix of a graph and v is an initial distribution over the vertices, the result above implies that we can speed up the computation of the distribution after s steps by a quadratic factor. Note that this application also motivates why uniform approximation is the right notion for algorithmic applications, since all we know is the interval in which eigenvalues of A lie while v can be any vector and, hence, we would like the approximating polynomial to be close everywhere in that interval.

While the computation of $\exp(-A)v$ is of fundamental interest in several areas of mathematics, physics, and engineering, our interest stems from its recent applications in algorithms and optimization. Roughly, these latter applications are manifestations of the *multiplicative weights method* for designing fast algorithms, and its extension to solving semi-definite programs via the framework by Arora and Kale [6].[5] At the heart of all algorithms based on the matrix multiplicative weights update method is a procedure to quickly compute $\exp(-A)v$ for a symmetric, positive semi-definite matrix A and a vector v. Since exact computation of the matrix exponential is expensive, we seek an approximation. It suffices to approximate the function e^{-x} on a certain interval. A simple approach is to truncate the Taylor series expansion of e^{-x}. However, we can use a polynomial approximation result for e^{-x} to produce an algorithm that saves a quadratic factor (a saving similar to the application above). In fact, when A has more structure, we can go beyond the square-root.

For fast graph algorithms, often the quantity of interest is $\exp(-t\mathcal{L})v$, where $\mathcal{L}$ is the normalized Laplacian of a graph, $t \geq 0$ and v is a vector. The vector $\exp(-t\mathcal{L})v$ can also be interpreted as the resulting distribution of a t-length continuous-time random walk on the graph with starting distribution v. Appealing to a rational approximation to e^{-x} with some additional prop-

[5] See also [26, 27, 28, 29, 50, 51, 48, 66, 5].

erties, the computation of $\exp(-t\mathcal{L})v$ can be reduced to a small number of computations of the form $\mathcal{L}^{-1}u$. Thus, using the near-linear-time Laplacian solver[6] due to Spielman and Teng [62], this gives an $\widetilde{O}(m)$-time algorithm for approximating $\exp(-t\mathcal{L})v$ for graphs with m edges. In the language of random walks, continuous-time random walks on an undirected graph can be simulated essentially independent of time; such is the power of rational approximations.

A natural question which arises from our last application is whether the Spielman-Teng result (which allows us to perform computations of the form $\mathcal{L}^{-1}u$) is *necessary* in order to compute $\exp(-\mathcal{L})v$ in near-linear time. In our final application of approximation theory, we answer this question in the affirmative: We show that the inverse of a positive-definite matrix can be approximated by a weighted-sum of a small number of matrix exponentials. Roughly, we show that for a PSD matrix A, $A^{-1} \approx \sum_{i=1}^{k} w_i \exp(-t_i A)$ for a small k. Thus, if there happens to be an algorithm that performs computations of the form $\exp(-t_i A)v$ in time T (independent of t_i), then we can compute $A^{-1}v$ in essentially $O(Tk)$ time. Thus, we show that the disparate looking problems of inversion and exponentiation are really the same from a point of view of designing fast algorithms.

Organization

We first present the ideas and results from approximation theory and subsequently we present applications to the design of fast algorithms. While we have tried to keep the presentation self-contained, for the sake of clarity, we have sometimes sacrificed tedious details. This means that, on rare occasions, we do not present complete proofs or do not present theorems with optimal parameters.

In Section 1, we present some essential notations and results from approximation theory. We introduce Chebyshev polynomials in Section 2, and prove certain extremal properties of these polynomials which are used in this monograph. In Sections 3 and 4 we construct polynomial approximations to

[6] A Laplacian solver is an algorithm that (approximately) solves a given system of linear equations $\mathcal{L}x = v$, where $\mathcal{L}$ is a (normalized) graph Laplacian and $v \in \mathrm{Im}(\mathcal{L})$, *i.e.*, it (approximately) computes $\mathcal{L}^{-1}v$; see [66].

the monomial x^s over the interval $[-1,1]$ and e^{-x} over the interval $[0,b]$ respectively. Both results are based on Chebyshev polynomials. In Section 5 we prove a special case of Markov's theorem which is then used to show that these polynomial approximations are asymptotically optimal.

Sections 6–7 are devoted to introducing techniques for understanding rational approximations for the function e^{-x} over the interval $[0,\infty)$. In Section 6, we first show that degree d rational functions can achieve c^d error for some $0 < c < 1$. Subsequently we prove that this result is optimal up to the choice of constant c. In Section 7 we present a proof of the theorem that such geometrically decaying errors for the e^{-x} can be achieved by rational functions with an additional restriction that all its poles be real and negative. We also show how to bound and compute the coefficients involved in this rational approximation result; this is crucial for the application presented in Section 11.

Sections 8–11 contain the presentation of applications of the approximation theory results. In Section 8 we show how the results of Section 3 imply that we can quadratically speed up random walks in graphs. Here, we discuss the important issue of computing the coefficients of the polynomials in Section 3. In Section 9 we present the famous Conjugate Gradient method for iteratively solving symmetric PSD systems $Ax = v$, where the number of iterations depends on the square-root of the condition number of A. The square-root saving is shown to be due to the scalar approximation result for x^s from Section 2. In Section 10 we present the Lanczos method and show how it can be used to approximate the largest eigenvalue of a symmetric matrix. We show how the existence of a good approximation for x^s, yet again, allows a quadratic speedup over the *power method*.

In Section 11 we show how the polynomial and rational approximations to e^{-x} developed in Sections 6 and 7 imply the best known algorithms for computing $\exp(-A)v$. If A is a symmetric and diagonally dominant (SDD) matrix, then we show how to combine rational approximations to e^{-x} with negative poles with the powerful SDD (Laplacian) solvers of Spielman-Teng to obtain near-linear time algorithms for computing $\exp(-A)v$.

Finally, in 12, we show how x^{-1} can be approximated by a sparse sum of the form $\sum_i w_i e^{-t_i x}$ over the interval $(0,1]$. The proof relies on the Euler-

Maclaurin formula and certain bounds derived from the Riemann zeta function. Using this result, we show how to reduce computation of $A^{-1}v$ for a symmetric positive-definite (PD) matrix A to the computation of a small number of computations of the form $\exp(-tA)v$. Apart from suggesting a new approach to solving a PD system, this result shows that computing $\exp(-A)v$ inherently requires the ability to solve a system of equations involving A.

References

[1] S. Aaronson. The polynomial method in quantum and classical computing. In *Foundations of Computer Science, 2008. FOCS '08. IEEE 49th Annual IEEE Symposium on*, pages 3–3, 2008.

[2] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions*. Dover, New York, fifth edition, 1964.

[3] N. Alon and V. D. Milman. λ_1, isoperimetric inequalities for graphs, and superconcentrators. *J. Comb. Theory, Ser. B*, 38(1):73–88, 1985.

[4] J.-E. Andersson. Approximation of e^{-x} by rational functions with concentrated negative poles. *Journal of Approximation Theory*, 32(2):85 – 95, 1981.

[5] S. Arora, E. Hazan, and S. Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(6):121–164, 2012.

[6] S. Arora and S. Kale. A combinatorial, primal-dual approach to semidefinite programs. In *STOC*, pages 227–236, 2007.

[7] R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf. Quantum lower bounds by polynomials. *J. ACM*, 48(4):778–797, July 2001.

[8] R. Beigel, N. Reingold, and D. Spielman. PP is closed under intersection. *Journal of Computer and System Sciences*, 50(2):191 – 202, 1995.

[9] G. Beylkin and L. Monzón. On approximation of functions by exponential sums. *Applied and Computational Harmonic Analysis*, 19(1):17 – 48, 2005.

[10] G. Beylkin and L. Monzón. Approximation by exponential sums revisited. *Applied and Computational Harmonic Analysis*, 28(2):131 – 149, 2010.

[11] É. Borel. *Lecons sur les Fonctions de Variables Réelles et les Développements en Séries de Polynomes*. Gauthier-Villars, Paris (2nd edition, 1928), 1905.

[12] M. Bun and J. Thaler. Dual lower bounds for approximate degree and Markov-Bernstein inequalities. In *Automata, Languages, and Programming*, volume 7965 of *Lecture Notes in Computer Science*, pages 303–314. Springer Berlin Heidelberg, 2013.

[13] P. L. Chebyshev. Théorie des mécanismes connus sous le nom de parallélogrammes. *Mém. Acad. Sci. Pétersb.*, 7:539–568, 1854.

[14] P. L. Chebyshev. Sur les questions de minima qui se rattachent à la représentation approximative des fonctions. *Mém. Acad. Sci. Pétersb.*, 7:199–291, 1859.

[15] J. Cheeger. A lower bound for the smallest eigenvalue of the Laplacian. *Problems Anal.*, pages 195–199, 1970.

[16] E. W. Cheney. *Introduction to approximation theory*. McGraw-Hill, New York, 1966.

[17] F. Chung. *Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No. 92)*. American Mathematical Society, 1997.

[18] F. Chung and O. Simpson. Solving linear systems with boundary conditions using heat kernel pagerank. In *Algorithms and Models for the Web Graph*, volume 8305 of *Lecture Notes in Computer Science*, pages 203–219. Springer International Publishing, 2013.

[19] W. J. Cody, G. Meinardus, and R. S. Varga. Chebyshev rational approximations to e^{-x} in $[0,\infty)$ and applications to heat-conduction problems. *Journal of Approximation Theory*, 2(1):50 – 65, 1969.

[20] A. A. Gonchar and E. A. Rakhmanov. On convergence of simultaneous Padé approximants for systems of functions of Markov type. *Proc. Steklov Inst. Math.*, 157:31–50, 1983.

[21] R. L. Graham, D. E. Knuth, and O. Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 1994.

[22] L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, STOC '96, pages 212–219, New York, NY, USA, 1996. ACM.

[23] N. J. A. Harvey, J. Nelson, and K. Onak. Sketching and streaming entropy via approximation theory. In *Foundations of Computer Science, 2008. FOCS '08. IEEE 49th Annual IEEE Symposium on*, pages 489–498, Oct 2008.

[24] M. R. Hestenes and E. Stiefel. Methods of Conjugate Gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49:409–436, December 1952.

[25] M. Hochbruck and C. Lubich. On Krylov subspace approximations to the matrix exponential operator. *SIAM J. Numer. Anal.*, 34(5):1911–1925, October 1997.

[26] R. Jain, Z. Ji, S. Upadhyay, and J. Watrous. QIP=PSPACE. *J. ACM*, 58(6):30:1–30:27, December 2011.

[27] R. Jain, S. Upadhyay, and J. Watrous. Two-message quantum interactive proofs are in PSPACE. In *Proceedings of the 2009 50th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '09, pages 534–543, 2009.

[28] R. Jain and J. Watrous. Parallel approximation of non-interactive zero-sum quantum games. *2012 IEEE 27th Conference on Computational Complexity*, 0:243–253, 2009.

[29] S. Kale. Efficient algorithms using the multiplicative weights update method. Technical report, Princeton University, Department of Computer Science, 2007.

[30] N. N. Kalitkin and I. A. Panin. On the computation of the exponential integral. *Mathematical Models and Computer Simulations*, 1(1):88–90, 2009.

[31] S. Kaniel. Estimates for some computational techniques in linear algebra. *Math. Comp.*, 20:369–378, 1966.

[32] J. A. Kelner, L. Orecchia, A. Sidford, and Z. A. Zhu. A simple, combinatorial algorithm for solving SDD systems in nearly-linear time. In *Proceedings of the 45th annual ACM symposium on Symposium on theory of computing*, STOC '13, pages 911–920. ACM, 2013.

[33] A. R. Klivans and R. Servedio. Learning DNF in time $2^{\widetilde{O}(n^{1/3})}$. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 258–265. ACM, 2001.

[34] I. Koutis, G. L. Miller, and R. Peng. A nearly- $m \log n$ time solver for SDD linear systems. In *Proceedings of the 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, FOCS '11, pages 590–598. IEEE Computer Society, 2011.

[35] J. Kuczyński and H. Woźniakowski. Estimating the largest eigenvalues by the power and Lanczos algorithms with a random start. *SIAM J. Matrix Anal. Appl.*, 13(4):1094–1122, October 1992.

[36] C. Lanczos. Solution of systems of linear equations by minimized iterations. *J. Res. Natl. Bur. Stand*, 49:33–53, 1952.

[37] D. A. Levin, Y. Peres, and E. L. Wilmer. *Markov chains and mixing times*. American Mathematical Society, 2006.

[38] A. A. Markov. On a question by D.I. mendeleev. *Zapiski Imperatorskoi Akademii Nauk*, 62:1–24, 1890.

[39] V. A. Markov. On functions deviating least from zero in a given interval. *Izdat. Imp. Akad. Nauk, St. Petersburg*, pages 218–258, 1892.

[40] M. Mitzenmacher and E. Upfal. *Probability and computing: Randomized algorithms and probabilistic analysis*. Cambridge University Press, New York, NY, USA, 2005.

[41] Y. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. In *Soviet Mathematics Doklady*, volume 27, pages 372–376, 1983.

[42] Y. Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer, 2004.

[43] D. J. Newman. Rational approximation to $|x|$. *Michigan Math. J.*, 11:11–14, 1964.

[44] D. J. Newman. Rational approximation to e^{-x}. *Journal of Approximation Theory*, 10(4):301 – 303, 1974.

[45] D. J. Newman. Approximation to x^n by lower degree rational functions. *Journal of Approximation Theory*, 27(3):236 – 238, 1979.

[46] N. Nisan and M. Szegedy. On the degree of Boolean functions as real polynomials. *Computational Complexity*, 4(4):301–313, 1994.

[47] L. Orecchia. *Fast Approximation Algorithms for Graph Partitioning using Spectral and Semidefinite-Programming Techniques*. PhD thesis, EECS Department, University of California, Berkeley, May 2011.

[48] L. Orecchia, S. Sachdeva, and N. K. Vishnoi. Approximating the exponential, the Lanczos method and an $\widetilde{O}(m)$-time spectral algorithm for Balanced Separator. *CoRR*, abs/1111.1491, 2011.

[49] L. Orecchia, S. Sachdeva, and N. K. Vishnoi. Approximating the exponential, the Lanczos method and an $\widetilde{O}(m)$-time spectral algorithm for Balanced Separator. STOC '12, pages 1141–1160, 2012.

[50] L. Orecchia, L. J. Schulman, U. V. Vazirani, and N. K. Vishnoi. On partitioning graphs via single commodity flows. In *STOC '08: Proc. 40th Ann. ACM Symp. Theory of Computing*, pages 461–470, 2008.

[51] L. Orecchia and N. K. Vishnoi. Towards an SDP-based approach to spectral methods: A nearly-linear-time algorithm for graph partitioning and decomposition. In *SODA'11: Proc. 22nd Ann. ACM-SIAM Symp. Discrete Algorithms*, pages 532–545, 2011.

[52] V. Y. Pan and Z. Q. Chen. The complexity of the matrix eigenproblem. In *STOC'99*, pages 507–516, 1999.

[53] B. N. Parlett. *The symmetric eigenvalue problem*, volume 7. SIAM, 1980.

[54] T. J. Rivlin. *An introduction to the approximation of functions*. Blaisdell book in numerical analysis and computer science. Blaisdell Pub. Co., 1969.

[55] W. Rudin. *Principles of mathematical analysis*. McGraw-Hill Book Co., New York, third edition, 1976. International Series in Pure and Applied Mathematics.

[56] Y. Saad. On the rates of convergence of the Lanczos and the block-Lanczos methods. *SIAM Journal on Numerical Analysis*, 17(5):pp. 687–706, 1980.

[57] Y. Saad. *Numerical methods for large eigenvalue problems*. Society for Industrial and Applied Mathematics, 2011.

[58] Y. Saad and H. A. van der Vorst. Iterative solution of linear systems in the 20th century. *Journal of Computational and Applied Mathematics*, 123(1-2):1–33, 2000. Numerical Analysis 2000. Vol. III: Linear Algebra.

[59] E. B. Saff, A. Schönhage, and R. S. Varga. Geometric convergence to e^{-z} by rational functions with real poles. *Numerische Mathematik*, 25:307–322, 1975.

[60] A. Schönhage. Zur rationalen Approximierbarkeit von e^{-x} über $[0,\infty)$. *Journal of Approximation Theory*, 7(4):395 – 398, 1973.

[61] A. Sherstov. Lower bounds in communication complexity and learning theory via analytic methods. Technical report, University of Texas at Austin, 2009.

[62] D. A. Spielman and S.-H. Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *STOC*, pages 81–90, New York, NY, USA, 2004. ACM.

[63] G. Szegö. Über eine Eigenschaft der Exponentialreihe. *Sitzungsber. Berl. Math. Ges.*, 23:50–64, 1924.

[64] G. Szegö. *Orthogonal polynomials*. American Mathematical Society Providence, 4th ed. edition, 1939.

[65] G. Valiant. Finding correlations in subquadratic time, with applications to learning parities and juntas. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 11–20. IEEE, 2012.

[66] N. K. Vishnoi. *Lx = b*. *Foundations and Trends in Theoretical Computer Science*, 8(1-2):1–141, 2012.

[67] K. Weierstrass. Über die analytische Darstellbarkeit sogenannter willkürlicher Functionen einer reellen veränderlichen. *Sitzungsberichte der Königlich Preußischen Akademie der Wissenschaften zu Berlin*, 2:633–639, 1885.

[68] S. M. Zemyan. On the zeroes of the Nth partial sum of the exponential series. *The American Mathematical Monthly*, 112(10):pp. 891–909, 2005.