$$Lx = b$$

$Lx = b$

Nisheeth K. Vishnoi

Microsoft Research
India
nisheeth.vishnoi@gmail.com

the essence of knowledge

Boston – Delft

Foundations and Trends® in Theoretical Computer Science

Published, sold and distributed by:
now Publishers Inc.
PO Box 1024
Hanover, MA 02339
USA
Tel. +1-781-985-4510
www.nowpublishers.com
sales@nowpublishers.com

Outside North America:
now Publishers Inc.
PO Box 179
2600 AD Delft
The Netherlands
Tel. +31-6-51115274

The preferred citation for this publication is N. K. Vishnoi, $Lx = b$ — Laplacian Solvers and Their Algorithmic Applications, Foundation and Trends® in Theoretical Computer Science, vol 8, nos 1–2, pp 1–141, 2012

ISBN: 978-1-60198-946-8

Foundations and Trends® in Theoretical Computer Science
Volume 8 Issues 1–2, 2012

Editorial Board

Editorial Scope

Foundations and Trends® in Theoretical Computer Science will publish survey and tutorial articles in the following topics:

- Algorithmic game theory
- Computational algebra
- Computational aspects of combinatorics and graph theory
- Computational aspects of communication
- Computational biology
- Computational complexity
- Computational geometry
- Computational learning
- Computational Models and Complexity
- Computational Number Theory
- Cryptography and information security
- Data structures
- Database theory
- Design and analysis of algorithms
- Distributed computing
- Information retrieval
- Operations Research
- Parallel algorithms
- Quantum Computation
- Randomness in Computation

Information for Librarians
Foundations and Trends® in Theoretical Computer Science, 2012, Volume 8, 4 issues. ISSN paper version 1551-305X. ISSN online version 1551-3068. Also available as a combined paper and online subscription.

Foundations and Trends® in
Theoretical Computer Science
Vol. 8, Nos. 1–2 (2012) 1–141

DOI: 10.1561/0400000054

now
the essence of knowledge

$Lx = b$
Laplacian Solvers and Their Algorithmic Applications

Nisheeth K. Vishnoi

Microsoft Research, India, nisheeth.vishnoi@gmail.com

Abstract

The ability to solve a system of linear equations lies at the heart of areas such as optimization, scientific computing, and computer science, and has traditionally been a central topic of research in the area of numerical linear algebra. An important class of instances that arise in practice has the form $L\mathbf{x} = \mathbf{b}$, where L is the Laplacian of an undirected graph. After decades of sustained research and combining tools from disparate areas, we now have Laplacian solvers that run in time nearly-linear in the sparsity (that is, the number of edges in the associated graph) of the system, which is a distant goal for general systems. Surprisingly, and perhaps not the original motivation behind this line of research, Laplacian solvers are impacting the theory of fast algorithms for fundamental graph problems. In this monograph, the emerging paradigm of employing Laplacian solvers to design novel fast algorithms for graph problems is illustrated through a small but carefully chosen set of examples. A part of this monograph is also dedicated to developing the ideas that go into the construction of near-linear-time Laplacian solvers. An understanding of these methods, which marry techniques from linear algebra and graph theory, will not only enrich the tool-set of an algorithm designer but will also provide the ability to adapt these methods to design fast algorithms for other fundamental problems.

Contents

Preface

The ability to solve a system of linear equations lies at the heart of areas such as optimization, scientific computing, and computer science and, traditionally, has been a central topic of research in numerical linear algebra. Consider a system $A\mathbf{x} = \mathbf{b}$ with n equations in n variables. Broadly, solvers for such a system of equations fall into two categories. The first is Gaussian elimination-based methods which, essentially, can be made to run in the time it takes to multiply two $n \times n$ matrices, (currently $O(n^{2.3\ldots})$ time). The second consists of iterative methods, such as the conjugate gradient method. These reduce the problem to computing n matrix–vector products, and thus make the running time proportional to mn where m is the number of nonzero entries, or sparsity, of A.[1] While this bound of n in the number of iterations is tight in the worst case, it can often be improved if A has additional structure, thus, making iterative methods popular in practice.

An important class of such instances has the form $L\mathbf{x} = \mathbf{b}$, where L is the Laplacian of an undirected graph G with n vertices and m edges

[1] Strictly speaking, this bound on the running time assumes that the numbers have bounded precision.

with m (typically) much smaller than n^2. Perhaps the simplest setting in which such *Laplacian systems* arise is when one tries to compute currents and voltages in a resistive electrical network. Laplacian systems are also important in practice, e.g., in areas such as scientific computing and computer vision. The fact that the system of equations comes from an underlying undirected graph made the problem of designing solvers especially attractive to theoretical computer scientists who entered the fray with tools developed in the context of graph algorithms and with the goal of bringing the running time down to $O(m)$. This effort gained serious momentum in the last 15 years, perhaps in light of an explosive growth in instance sizes which means an algorithm that does not scale near-linearly is likely to be impractical.

After decades of sustained research, we now have a solver for Laplacian systems that runs in $O(m \log n)$ time. While many researchers have contributed to this line of work, Spielman and Teng spearheaded this endeavor and were the first to bring the running time down to $\tilde{O}(m)$ by combining tools from graph partitioning, random walks, and low-stretch spanning trees with numerical methods based on Gaussian elimination and the conjugate gradient. Surprisingly, and not the original motivation behind this line of research, Laplacian solvers are impacting the theory of fast algorithms for fundamental graph problems; giving back to an area that empowered this work in the first place.

That is the story this monograph aims to tell in a comprehensive manner to researchers and aspiring students who work in algorithms or numerical linear algebra. The emerging paradigm of employing Laplacian solvers to design novel fast algorithms for graph problems is illustrated through a small but carefully chosen set of problems such as graph partitioning, computing the matrix exponential, simulating random walks, graph sparsification, and single-commodity flows. A significant part of this monograph is also dedicated to developing the algorithms and ideas that go into the proof of the Spielman–Teng Laplacian solver. It is a belief of the author that an understanding of these methods, which marry techniques from linear algebra and graph theory, will not only enrich the tool-set of an algorithm designer, but will also provide the ability to adapt these methods to design fast algorithms for other fundamental problems.

How to use this monograph. This monograph can be used as the text for a graduate-level course or act as a supplement to a course on spectral graph theory or algorithms. The writing style, which deliberately emphasizes the presentation of key ideas over rigor, should even be accessible to advanced undergraduates. If one desires to teach a course based on this monograph, then the best order is to go through the sections linearly. Essential are Sections 1 and 2 that contain the basic linear algebra material necessary to follow this monograph and Section 3 which contains the statement and a discussion of the main theorem regarding Laplacian solvers. Parts of this monograph can also be read independently. For instance, Sections 5–7 contain the Cheeger inequality based spectral algorithm for graph partitioning. Sections 15 and 16 can be read in isolation to understand the conjugate gradient method. Section 19 looks ahead into computing more general functions than the inverse and presents the Lanczos method. A dependency diagram between sections appears in Figure 1. For someone solely interested in a near-linear-time algorithm for solving Laplacian systems, the quick path to Section 14, where the approach of a short and new proof is presented, should suffice. However, the author recommends going all

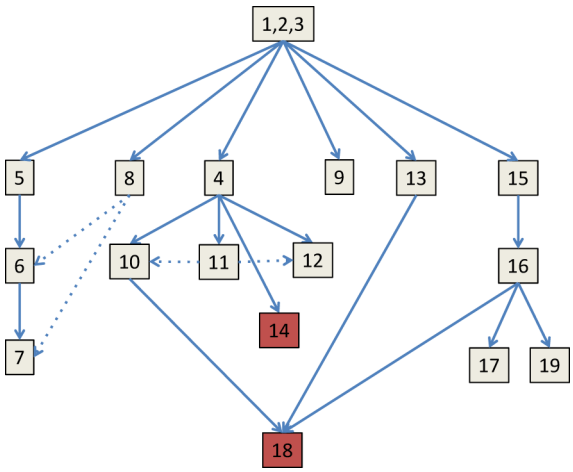

Fig. 1 The dependency diagram among the sections in this monograph. A dotted line from i to j means that the results of Section j use some results of Section i in a black-box manner and a full understanding is not required.

the way to Section 18 where multiple techniques developed earlier in the monograph come together to give an $\tilde{O}(m)$ Laplacian solver.

Acknowledgments. This monograph is partly based on lectures delivered by the author in a course at the Indian Institute of Science, Bangalore. Thanks to the scribes: Deeparnab Chakrabarty, Avishek Chatterjee, Jugal Garg, T. S. Jayaram, Swaprava Nath, and Deepak R. Special thanks to Elisa Celis, Deeparnab Chakrabarty, Lorenzo Orecchia, Nikhil Srivastava, and Sushant Sachdeva for reading through various parts of this monograph and providing valuable feedback. Finally, thanks to the reviewer(s) for several insightful comments which helped improve the presentation of the material in this monograph.

Bangalore
15 January 2013

Nisheeth K. Vishnoi
Microsoft Research India

Notation

- The set of real numbers is denoted by $\mathbb{R}$, and $\mathbb{R}_{\geq 0}$ denotes the set of nonnegative reals. We only consider real numbers in this monograph.
- The set of integers is denoted by $\mathbb{Z}$, and $\mathbb{Z}_{\geq 0}$ denotes the set of nonnegative integers.
- Vectors are denoted by boldface, e.g., $\mathbf{u}, \mathbf{v}$. A vector $\mathbf{v} \in \mathbb{R}^n$ is a column vector but often written as $\mathbf{v} = (v_1, \ldots, v_n)$. The transpose of a vector $\mathbf{v}$ is denoted by $\mathbf{v}^\top$.
- For vectors $\mathbf{u}, \mathbf{v}$, their inner product is denoted by $\langle \mathbf{u}, \mathbf{v} \rangle$ or $\mathbf{u}^\top \mathbf{v}$.
- For a vector $\mathbf{v}$, $\|\mathbf{v}\|$ denotes its ℓ_2 or Euclidean norm where $\|\mathbf{v}\| \stackrel{\text{def}}{=} \sqrt{\langle \mathbf{v}, \mathbf{v} \rangle}$. We sometimes also refer to the ℓ_1 or Manhattan distance norm $\|\mathbf{v}\|_1 \stackrel{\text{def}}{=} \sum_{i=1}^{n} |v_i|$.
- The outer product of a vector $\mathbf{v}$ with itself is denoted by $\mathbf{v}\mathbf{v}^\top$.
- Matrices are denoted by capitals, e.g., A, L. The transpose of A is denoted by $A^\top$.
- We use t_A to denote the time it takes to multiply the matrix A with a vector.

- The A-norm of a vector $\mathbf{v}$ is denoted by $\|\mathbf{v}\|_A \stackrel{\text{def}}{=} \sqrt{\mathbf{v}^\top A \mathbf{v}}$.
- For a real symmetric matrix A, its real eigenvalues are ordered $\lambda_1(A) \leq \lambda_2(A) \leq \cdots \leq \lambda_n(A)$. We let $\Lambda(A) \stackrel{\text{def}}{=} [\lambda_1(A), \lambda_n(A)]$.
- A positive-semidefinite (PSD) matrix is denoted by $A \succeq 0$ and a positive-definite matrix $A \succ 0$.
- The norm of a symmetric matrix A is denoted by $\|A\| \stackrel{\text{def}}{=} \max\{|\lambda_1(A)|, |\lambda_n(A)|\}$. For a symmetric PSD matrix A, $\|A\| = \lambda_n(A)$.
- Thinking of a matrix A as a linear operator, we denote the image of A by $\mathrm{Im}(A)$ and the rank of A by $\mathrm{rank}(A)$.
- A graph G has a vertex set V and an edge set E. All graphs are assumed to be undirected unless stated otherwise. If the graph is weighted, there is a weight function $w : E \mapsto \mathbb{R}_{\geq 0}$. Typically, n is reserved for the number of vertices $|V|$, and m for the number of edges $|E|$.
- $\mathbb{E}_{\mathcal{F}}[\cdot]$ denotes the expectation and $\mathbb{P}_{\mathcal{F}}[\cdot]$ denotes the probability over a distribution $\mathcal{F}$. The subscript is dropped when clear from context.
- The following acronyms are used liberally, with respect to (w.r.t.), without loss of generality (w.l.o.g.), with high probability (w.h.p.), if and only if (iff), right-hand side (r.h.s.), left-hand side (l.h.s.), and such that (s.t.).
- Standard big-o notation is used to describe the limiting behavior of a function. $\tilde{O}$ denotes potential logarithmic factors which are ignored, i.e., $f = \tilde{O}(g)$ is equivalent to $f = O(g \log^k(g))$ for some constant k.

Part I

Basics

1

Basic Linear Algebra

This section reviews basics from linear algebra, such as eigenvalues and eigenvectors, that are relevant to this monograph. The spectral theorem for symmetric matrices and min–max characterizations of eigenvalues are derived.

1.1 Spectral Decomposition of Symmetric Matrices

One way to think of an $m \times n$ matrix A with real entries is as a linear operator from $\mathbb{R}^n$ to $\mathbb{R}^m$ which maps a vector $\mathbf{v} \in \mathbb{R}^n$ to $A\mathbf{v} \in \mathbb{R}^m$. Let $\dim(S)$ be dimension of S, i.e., the maximum number of linearly independent vectors in S. The rank of A is defined to be the dimension of the image of this linear transformation. Formally, the image of A is defined to be $\mathrm{Im}(A) \stackrel{\text{def}}{=} \{\mathbf{u} \in \mathbb{R}^m : \mathbf{u} = A\mathbf{v} \text{ for some } \mathbf{v} \in \mathbb{R}^n\}$, and the rank is defined to be $\mathrm{rank}(A) \stackrel{\text{def}}{=} \dim(\mathrm{Im}(A))$ and is at most $\min\{m, n\}$.

We are primarily interested in the case when A is square, i.e., $m = n$, and symmetric, i.e., $A^\top = A$. Of interest are vectors $\mathbf{v}$ such that $A\mathbf{v} = \lambda\mathbf{v}$ for some λ. Such a vector is called an eigenvector of A with respect to (w.r.t.) the eigenvalue λ. It is a basic result in linear algebra that every real matrix has n eigenvalues, though some of them could

be complex. If A is symmetric, then one can show that the eigenvalues are real. For a complex number $z = a + ib$ with $a, b \in \mathbb{R}$, its conjugate is defined as $\bar{z} = a - ib$. For a vector $\mathbf{v}$, its conjugate transpose $\mathbf{v}^\star$ is the transpose of the vector whose entries are conjugates of those in $\mathbf{v}$. Thus, $\mathbf{v}^\star \mathbf{v} = \|\mathbf{v}\|^2$.

Lemma 1.1. If A is a real symmetric $n \times n$ matrix, then all of its eigenvalues are real.

Proof. Let λ be an eigenvalue of A, possibly complex, and $\mathbf{v}$ be the corresponding eigenvector. Then, $A\mathbf{v} = \lambda \mathbf{v}$. Conjugating both sides we obtain that $\mathbf{v}^\star A^\top = \overline{\lambda} \mathbf{v}^\star$, where $\mathbf{v}^\star$ is the conjugate transpose of $\mathbf{v}$. Hence, $\mathbf{v}^\star A \mathbf{v} = \overline{\lambda} \mathbf{v}^\star \mathbf{v}$, since A is symmetric. Thus, $\lambda \|\mathbf{v}\|^2 = \overline{\lambda} \|\mathbf{v}\|^2$ which implies that $\lambda = \overline{\lambda}$. Thus, $\lambda \in \mathbb{R}$. □

Let $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ be the n real eigenvalues, or the *spectrum*, of A with corresponding eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_n$. For a symmetric matrix, its *norm* is

$$\|A\| \stackrel{\text{def}}{=} \max\{|\lambda_1(A)|, |\lambda_n(A)|\}.$$

We now study eigenvectors that correspond to distinct eigenvalues.

Lemma 1.2. Let λ_i and λ_j be two eigenvalues of a symmetric matrix A, and $\mathbf{u}_i$, $\mathbf{u}_j$ be the corresponding eigenvectors. If $\lambda_i \neq \lambda_j$, then $\langle \mathbf{u}_i, \mathbf{u}_j \rangle = 0$.

Proof. Given $A\mathbf{u}_i = \lambda_i \mathbf{u}_i$ and $A\mathbf{u}_j = \lambda_j \mathbf{u}_j$, we have the following sequence of equalities. Since A is symmetric, $\mathbf{u}_i^\top A^\top = \mathbf{u}_i^\top A$. Thus, $\mathbf{u}_i^\top A \mathbf{u}_j = \lambda_i \mathbf{u}_i^\top \mathbf{u}_j$ on the one hand, and $\mathbf{u}_i^\top A \mathbf{u}_j = \lambda_j \mathbf{u}_i^\top \mathbf{u}_j$ on the other. Therefore, $\lambda_j \mathbf{u}_i^\top \mathbf{u}_j = \lambda_i \mathbf{u}_i^\top \mathbf{u}_j$. This implies that $\mathbf{u}_i^\top \mathbf{u}_j = 0$ since $\lambda_i \neq \lambda_j$. □

Hence, the eigenvectors corresponding to different eigenvalues are orthogonal. Moreover, if $\mathbf{u}_i$ and $\mathbf{u}_j$ correspond to the same eigenvalue λ, and are linearly independent, then any linear combination

is also an eigenvector corresponding to the same eigenvalue. The maximal eigenspace of an eigenvalue is the space spanned by all eigenvectors corresponding to that eigenvalue. Hence, the above lemma implies that one can decompose $\mathbb{R}^n$ into maximal eigenspaces U_i, each of which corresponds to an eigenvalue of A, and the eigenspaces corresponding to distinct eigenvalues are orthogonal. Thus, if $\lambda_1 < \lambda_2 < \cdots < \lambda_k$ are the set of distinct eigenvalues of a real symmetric matrix A, and U_i is the eigenspace associated with λ_i, then, from the discussion above,

$$\sum_{i=1}^{k} \dim(U_i) = n.$$

Hence, given that we can pick an orthonormal basis for each U_i, we may assume that the eigenvectors of A form an orthonormal basis for $\mathbb{R}^n$. Thus, we have the following spectral decomposition theorem.

Theorem 1.3. Let $\lambda_1 \leq \cdots \leq \lambda_n$ be the spectrum of A with corresponding eigenvalues $\mathbf{u}_1, \ldots, \mathbf{u}_n$. Then, $A = \sum_{i=1}^{n} \lambda_i \mathbf{u}_i \mathbf{u}_i^\top$.

Proof. Let $B \stackrel{\text{def}}{=} \sum_{i=1}^{n} \lambda_i \mathbf{u}_i \mathbf{u}_i^\top$. Then,

$$\begin{aligned} B\mathbf{u}_j &= \sum_{i=1}^{n} \lambda_i \mathbf{u}_i \mathbf{u}_i^\top \mathbf{u}_j \\ &= \lambda_j \mathbf{u}_j \\ &= A\mathbf{u}_j. \end{aligned}$$

The above is true for all j. Since $\mathbf{u}_j$s are orthonormal basis of $\mathbb{R}^n$, we have for all $\mathbf{v} \in \mathbb{R}^n$, $A\mathbf{v} = B\mathbf{v}$. This implies $A = B$. □

Thus, when A is a real and symmetric matrix, $\mathrm{Im}(A)$ is spanned by the eigenvectors with nonzero eigenvalues. From a computational perspective, such a decomposition can be computed in time polynomial in the bits needed to represent the entries of A.[1]

[1] To be very precise, one can only compute eigenvalues and eigenvectors to a high enough precision in polynomial time. We will ignore this distinction for this monograph as we do not need to know the *exact* values.

1.2 Min–Max Characterizations of Eigenvalues

Now we present a variational characterization of eigenvalues which is very useful.

Lemma 1.4. If A is an $n \times n$ real symmetric matrix, then the largest eigenvalue of A is

$$\lambda_n(A) = \max_{\mathbf{v} \in \mathbb{R}^n \setminus \{\mathbf{0}\}} \frac{\mathbf{v}^\top A \mathbf{v}}{\mathbf{v}^\top \mathbf{v}}.$$

Proof. Let $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ be the eigenvalues of A, and let $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n$ be the corresponding orthonormal eigenvectors which span $\mathbb{R}^n$. Hence, for all $\mathbf{v} \in \mathbb{R}^n$, there exist $c_1, \ldots, c_n \in \mathbb{R}$ such that $\mathbf{v} = \sum_i c_i \mathbf{u}_i$. Thus,

$$\begin{aligned} \langle \mathbf{v}, \mathbf{v} \rangle &= \left\langle \sum_i c_i \mathbf{u}_i, \sum_i c_i \mathbf{u}_i \right\rangle \\ &= \sum_i c_i^2. \end{aligned}$$

Moreover,

$$\begin{aligned} \mathbf{v}^\top A \mathbf{v} &= \left(\sum_i c_i \mathbf{u}_i \right)^\top \left(\sum_j \lambda_j \mathbf{u}_j \mathbf{u}_j^\top \right) \left(\sum_k c_k \mathbf{u}_k \right) \\ &= \sum_{i,j,k} c_i c_k \lambda_j (\mathbf{u}_i^\top \mathbf{u}_j) \cdot (\mathbf{u}_j^\top \mathbf{u}_k) \\ &= \sum_i c_i^2 \lambda_i \\ &\leq \lambda_n \sum_i c_i^2 = \lambda_n \langle \mathbf{v}, \mathbf{v} \rangle . \end{aligned}$$

Hence, $\forall\ \mathbf{v} \neq 0$, $\frac{\mathbf{v}^\top A \mathbf{v}}{\mathbf{v}^\top \mathbf{v}} \leq \lambda_n$. This implies,

$$\max_{\mathbf{v} \in \mathbb{R}^n \setminus \{\mathbf{0}\}} \frac{\mathbf{v}^\top A \mathbf{v}}{\mathbf{v}^\top \mathbf{v}} \leq \lambda_n.$$

Now note that setting $\mathbf{v} = \mathbf{u}_n$ achieves this maximum. Hence, the lemma follows. □

If one inspects the proof above, one can deduce the following lemma just as easily.

Lemma 1.5. If A is an $n \times n$ real symmetric matrix, then the smallest eigenvalue of A is

$$\lambda_1(A) = \min_{\mathbf{v} \in \mathbb{R}^n \setminus \{\mathbf{0}\}} \frac{\mathbf{v}^\top A \mathbf{v}}{\mathbf{v}^\top \mathbf{v}}.$$

More generally, one can extend the proof of the lemma above to the following. We leave it as a simple exercise.

Theorem 1.6. If A is an $n \times n$ real symmetric matrix, then for all $1 \leq k \leq n$, we have

$$\lambda_k(A) = \min_{\mathbf{v} \in \mathbb{R}^n \setminus \{\mathbf{0}\}, \mathbf{v}^\top \mathbf{u}_i = 0, \forall i \in \{1, \ldots, k-1\}} \frac{\mathbf{v}^\top A \mathbf{v}}{\mathbf{v}^\top \mathbf{v}},$$

and

$$\lambda_k(A) = \max_{\mathbf{v} \in \mathbb{R}^n \setminus \{\mathbf{0}\}, \mathbf{v}^\top \mathbf{u}_i = 0, \forall i \in \{k+1, \ldots, n\}} \frac{\mathbf{v}^\top A \mathbf{v}}{\mathbf{v}^\top \mathbf{v}}.$$

Notes

Some good texts to review basic linear algebra are [35, 82, 85]. Theorem 1.6 is also called the Courant–Fischer–Weyl min–max principle.

References

[1] I. Abraham, Y. Bartal, and O. Neiman, "Nearly tight low stretch spanning trees," in *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 781–790, 2008.

[2] I. Abraham and O. Neiman, "Using petal-decompositions to build a low stretch spanning tree," in *ACM Symposium on Theory of Computing (STOC)*, pp. 395–406, 2012.

[3] D. Achlioptas, "Database-friendly random projections: Johnson-Lindenstrauss with binary coins," *Journal of Computer and Systems Sciences*, vol. 66, no. 4, pp. 671–687, 2003.

[4] R. Ahlswede and A. Winter, "Addendum to "Strong converse for identification via quantum channels"," *IEEE Transactions on Information Theory*, vol. 49, no. 1, p. 346, 2003.

[5] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, February 1993.

[6] N. Alon, R. M. Karp, D. Peleg, and D. B. West, "A graph-theoretic game and its application to the k-server problem," *SIAM Journal on Computing*, vol. 24, no. 1, pp. 78–100, 1995.

[7] N. Alon and V. D. Milman, "λ_1, isoperimetric inequalities for graphs, and superconcentrators," *Journal of Combinational Theory, Series B*, vol. 38, no. 1, pp. 73–88, 1985.

[8] R. Andersen, F. R. K. Chung, and K. J. Lang, "Local graph partitioning using pagerank vectors," in *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 475–486, 2006.

[9] R. Andersen and Y. Peres, "Finding sparse cuts locally using evolving sets," in *ACM Symposium on Theory of Computing (STOC)*, pp. 235–244, 2009.

[10] S. Arora, E. Hazan, and S. Kale, "$O(\sqrt{\log n})$ approximation to sparsest cut in $\tilde{O}(n^2)$ time," in *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 238–247, 2004.

[11] S. Arora, E. Hazan, and S. Kale, "The multiplicative weights update method: A meta-algorithm and applications," *Theory of Computing*, vol. 8, no. 1, pp. 121–164, 2012.

[12] S. Arora and S. Kale, "A combinatorial, primal-dual approach to semidefinite programs," in *ACM Symposium on Theory of Computing (STOC)*, pp. 227–236, 2007.

[13] S. Arora, S. Rao, and U. V. Vazirani, "Expander flows, geometric embeddings and graph partitioning," *Journal of the ACM*, vol. 56, no. 2, 2009.

[14] J. D. Batson, D. A. Spielman, and N. Srivastava, "Twice-Ramanujan sparsifiers," in *ACM Symposium on Theory of Computing (STOC)*, pp. 255–262, 2009.

[15] M. Belkin, I. Matveeva, and P. Niyogi, "Regularization and semi-supervised learning on large graphs," in *Proceedings of the Workshop on Computational Learning Theory (COLT)*, pp. 624–638, 2004.

[16] A. A. Benczúr and D. R. Karger, "Approximating s–t minimum cuts in $\tilde{O}(n^2)$ time," in *ACM Symposium on Theory of Computing (STOC)*, pp. 47–55, 1996.

[17] R. Bhatia, *Matrix Analysis (Graduate Texts in Mathematics)*. Springer, 1996.

[18] E. G. Boman and B. Hendrickson, "Support theory for preconditioning," *SIAM Journal on Matrix Analysis Applications*, vol. 25, no. 3, pp. 694–717, 2003.

[19] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, March 2004.

[20] J. Cheeger, "A lower bound for the smallest eigenvalue of the Laplacian," *Problems in Analysis*, pp. 195–199, 1970.

[21] E. W. Cheney, *Introduction to Approximation Theory/E. W. Cheney*. New York: McGraw-Hill, 1966.

[22] P. Christiano, J. A. Kelner, A. Madry, D. A. Spielman, and S. Teng, "Electrical flows, Laplacian systems, and faster approximation of maximum flow in undirected graphs," in *ACM Symposium on Theory of Computing (STOC)*, pp. 273–282, 2011.

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

[24] S. I. Daitch and D. A. Spielman, "Faster approximate lossy generalized flow via interior point algorithms," in *ACM Symposium on Theory of Computing (STOC)*, pp. 451–460, 2008.

[25] P. G. Doyle and J. L. Snell, *Random Walks and Electric Networks*. Washington, DC: Mathematical Association of America, 1984.

[26] P. Elias, A. Feinstein, and C. Shannon, "A note on the maximum flow through a network," *IEEE Transactions on Information Theory*, vol. 2, no. 4, pp. 117–119, December 1956.

[27] M. Elkin, Y. Emek, D. A. Spielman, and S.-H. Teng, "Lower-stretch spanning trees," *SIAM Journal on Computing*, vol. 38, no. 2, pp. 608–628, 2008.

[28] M. Fiedler, "Algebraic connectivity of graphs," *Czechoslovak Mathematical Journal*, vol. 23, pp. 298–305, 1973.

[29] L. R. Ford and D. R. Fulkerson, "Maximal flow through a network," *Canadian Journal of Mathematics*, vol. 8, pp. 399–404, 1954.
[30] A. Frangioni and C. Gentile, "Prim-based support-graph preconditioners for min-cost flow problems," *Computational Optimization and Applications*, vol. 36, no. 2–3, pp. 271–287, April 2007.
[31] W. S. Fung, R. Hariharan, N. J. A. Harvey, and D. Panigrahi, "A general framework for graph sparsification," in *ACM Symposium on Theory of Computing (STOC)*, pp. 71–80, 2011.
[32] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman, 1979.
[33] A. George, "Nested dissection of a regular finite element mesh," *SIAM Journal on Numerical Analysis*, vol. 10, no. 2, pp. 345–363, 1973.
[34] C. D. Godsil and G. Royle, *Algebraic Graph Theory*. Springer, 2001.
[35] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Johns Hopkins Univ. Press, 1996.
[36] G. H. Golub and M. L. Overton, "The convergence of inexact Chebyshev and Richardson iterative methods for solving linear systems," Technical Report, Stanford University, Stanford, CA, USA, 1987.
[37] G. H. Golub and R. S. Varga, "Chebyshev semi-iterative methods, successive overrelaxation iterative methods, and second order Richardson iterative methods," *Numerische Mathematik*, vol. 3, pp. 157–168, 1961.
[38] K. Gremban, "Combinatorial preconditioners for sparse, symmetric, diagonally dominant linear systems," PhD Thesis, Carnegie Mellon University, Pittsburgh, CMU CS Tech Report CMU-CS-96-123, October 1996.
[39] M. R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *Journal of Research of the National Bureau of Standards*, vol. 49, pp. 409–436, December 1952.
[40] G. Iyengar, D. J. Phillips, and C. Stein, "Approximating semidefinite packing programs," *SIAM Journal on Optimization*, vol. 21, no. 1, pp. 231–268, 2011.
[41] G. Iyengar, D. J. Phillips, and C. Stein, "Approximation algorithms for semidefinite packing problems with applications to maxcut and graph coloring," in *IPCO'05: Proceedings of Conference on Integer Programming and Combinatorial Optimization*, pp. 152–166, 2005.
[42] A. Joshi, "Topics in optimization and sparse linear systems," PhD Thesis, University of Illinois at Urbana-Champaign, Champaign, IL, USA, UMI Order No. GAX97-17289, 1997.
[43] S. Kaczmarz, "Ängenäherte Auflösung von Systemen linearer Gleichungen," *Bulletin International del' Académie Polonaise Sciences et des Lettres*, pp. 355–356, 1937.
[44] S. Kale, "Efficient algorithms using the multiplicative weights update method," PhD Thesis, Princeton University, Department of Computer Science, 2007.
[45] J. A. Kelner and A. Madry, "Faster generation of random spanning trees," in *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 13–21, 2009.
[46] J. A. Kelner, G. L. Miller, and R. Peng, "Faster approximate multicommodity flow using quadratically coupled flows," in *ACM Symposium on Theory of Computing (STOC)*, pp. 1–18, 2012.

[47] J. A. Kelner, L. Orecchia, A. Sidford, and Z. A. Zhu, "A simple, combinatorial algorithm for solving SDD systems in nearly-linear time," in *ACM Symposium on Theory of Computing (STOC)*, 2013.

[48] R. Khandekar, S. Rao, and U. V. Vazirani, "Graph partitioning using single commodity flows," *Journal of the ACM*, vol. 56, no. 4, 2009.

[49] I. Koutis, G. L. Miller, and R. Peng, "Approaching optimality for solving SDD linear systems," in *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 235–244, 2010.

[50] I. Koutis, G. L. Miller, and R. Peng, "A nearly-$m \log n$ time solver for SDD linear systems," in *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 590–598, 2011.

[51] D. A. Levin, Y. Peres, and E. L. Wilmer, *Markov Chains and Mixing Times*. American Mathematical Society, 2006.

[52] R. J. Lipton, D. J. Rose, and R. E. Tarjan, "Generalized nested dissection," *SIAM Journal on Numerical Analysis*, vol. 16, no. 2, pp. 346–358, 1979.

[53] L. Lovász and M. Simonovits, "Random walks in a convex body and an improved volume algorithm," *Random Structures & Algorithms*, vol. 4, no. 4, pp. 359–412, 1993.

[54] R. Lyons and Y. Peres, *Probability on Trees and Networks*. To Appear in Cambridge University Press, 2012.

[55] A. Madry, "Fast approximation algorithms for cut-based problems in undirected graphs," in *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 245–254, 2010.

[56] M. W. Mahoney, L. Orecchia, and N. K. Vishnoi, "A spectral algorithm for improving graph partitions," *Journal of Machine Learning Research*, vol. 13, pp. 2339–2365, 2012.

[57] S. Maji, N. K. Vishnoi, and J. Malik, "Biased normalized cuts," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2057–2064, 2011.

[58] M. Mihail, "Conductance and convergence of markov chains — a combinatorial treatment of expanders," in *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 526–531, 1989.

[59] L. Orecchia, "Fast approximation algorithms for graph partitioning using spectral and semidefinite-programming techniques," PhD Thesis, EECS Department, University of California, Berkeley, May 2011.

[60] L. Orecchia, S. Sachdeva, and N. K. Vishnoi, "Approximating the exponential, the Lanczos method and an $\tilde{O}(m)$-time spectral algorithm for balanced separator," in *ACM Symposium on Theory of Computing (STOC)*, pp. 1141–1160, 2012.

[61] L. Orecchia, L. J. Schulman, U. V. Vazirani, and N. K. Vishnoi, "On partitioning graphs via single commodity flows," in *ACM Symposium on Theory of Computing (STOC)*, pp. 461–470, 2008.

[62] 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 *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 532–545, 2011.

[63] V. Y. Pan and Z. Q. Chen, "The complexity of the matrix eigenproblem," in *ACM Symposium on Theory of Computing (STOC)*, pp. 507–516, 1999.

[64] L. F. Richardson, "The approximate arithmetical solution by finite differences of physical problems involving differential equations with an application to the stresses in a masonry dam," *Transactions of the Royal Society of London*, vol. Series A, no. 210, pp. 307–357, 1910.

[65] T. J. Rivlin, *An Introduction to the Approximation of Functions*. Blaisdell book in numerical analysis and computer science. Blaisdell Pub. Co., 1969.

[66] M. Rudelson, "Random vectors in the isotropic position," *Journal of Functional Analysis*, vol. 164, no. 1, pp. 60–72, 1999.

[67] Y. Saad, "Analysis of some Krylov subspace approximations to the matrix exponential operator," *SIAM Journal on Numerical Analysis*, vol. 29, pp. 209–228, February 1992.

[68] Y. Saad, *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics. Philadelphia, PA, USA, 2nd Edition, 2003.

[69] S. Sachdeva and N. K. Vishnoi, "Inversion is as easy as exponentiation," *manuscript*, 2012.

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

[71] J. Sherman, "Breaking the multicommodity flow barrier for $O(\sqrt{\log n})$-approximations to sparsest cut," in *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS)*, 2009.

[72] J. R. Shewchuk, "An introduction to the conjugate gradient method without the agonizing pain," Technical Report, Carnegie Mellon University, Pittsburgh, PA, USA, 1994.

[73] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 888–905, 1997.

[74] D. B. Shmoys, *Cut Problems and Their Application to Divide-and-Conquer*. pp. 192–235, 1997.

[75] D. A. Spielman, "Algorithms, graph theory, and linear equations in Laplacian matrices," in *Proceedings of International Congress of Mathematicians (ICM'10)*, 2010.

[76] D. A. Spielman and N. Srivastava, "Graph sparsification by effective resistances," *SIAM Journal on Computing*, vol. 40, no. 6, pp. 1913–1926, 2011.

[77] D. A. Spielman and S.-H. Teng, "Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems," in *ACM Symposium on Theory of Computing (STOC)*, pp. 81–90, New York, NY, USA, 2004.

[78] D. A. Spielman and S.-H. Teng, "Nearly-linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems," CoRR, abs/cs/0607105, 2006.

[79] D. A. Spielman and S.-H. Teng, "A local clustering algorithm for massive graphs and its application to nearly-linear time graph partitioning," CoRR, abs/0809.3232, 2008.

[80] D. A. Spielman and S.-H. Teng, "Spectral sparsification of graphs," *SIAM Journal on Computing*, vol. 40, no. 4, pp. 981–1025, 2011.

[81] D. A. Spielman and J. Woo, "A note on preconditioning by low-stretch spanning trees," CoRR, abs/0903.2816, 2009.

[82] G. Strang, *Linear Algebra and Its Applications*. Harcourt Brace Jonanovich. San Diego, 3rd Edition, 1988.

[83] T. Strohmer and R. Vershynin, "A randomized Kaczmarz algorithm with exponential convergence," *Journal of Fourier Analysis and Applications*, vol. 15, pp. 262–278, 2009.

[84] S. Teng, "The Laplacian paradigm: Emerging algorithms for massive graphs," in *Proceedings of the Annual Conference on Theory and Applications of Models of Computation (TAMC)*, pp. 2–14, 2010.

[85] L. N. Trefethen and D. Bau, *Numerical Linear Algebra*. SIAM, 1997.

[86] P. M. Vaidya, "Solving linear equations with symmetric diagonally dominant matrices by constructing good preconditioners," Technical Report, Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, 1990.

[87] J. vanden Eshof and M. Hochbruck, "Preconditioning Lanczos approximations to the matrix exponential," *SIAM Journal on Scientific Computing*, vol. 27, pp. 1438–1457, November 2005.

[88] J. H. Wilkinson, *The Algebraic Eigenvalue Problem (Monographs on Numerical Analysis)*. USA: Oxford University Press, 1st Edition, April 1988.

[89] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," in *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 912–919, 2003.