Index of Notation and Definitions

CS 292F: Graph Laplacians and Spectra

Version of May 11, 2021

There is a lot of variation in terminology and notation in the field of Laplacian matrix computation and spectral graph theory. Indeed, even "Laplacian matrix" is defined differently by different authors!

This list gives the versions of notation, terminology, and definitions that we will use in CS 292F. I mostly follow the conventions of Dan Spielman's notes, though I prefer not to use greek letters for vectors. I will keep adding to this list during the quarter.

1. Unless otherwise stated, a *graph* $G = (V, E)$ is always an undirected graph whose n vertices are the integers 1 through n, with no multiple edges or loops.

2. The *degree* of a vertex is the number of edges incident on it, or equivalently (because we don't allow multiple edges or loops) the number of its neighboring vertices.

3. A graph is said to be *regular* if every vertex has the same degree.

4. A graph is said to be *connected* if, for every choice of two vertices a and b, there is a *path* of edges from a to b. The *connected components* of a graph are its maximal connected subgraphs.

5. K_n is the *complete graph*, which has n vertices and all $n(n-1)/2$ possible edges.

6. P_n is the *path graph*, which has n vertices and $n-1$ edges in a single path.

7. S_n is the *star graph*, which has n vertices, one with degree $n-1$ and $n-1$ with degree 1.

8. H_k is the *hypercube graph*, which has $n = 2^k$ vertices, all of degree k. Vertices i and j have an edge between them if i and j differ by a power of 2. Equivalently, we can identify each vertex with a subset of $\{1, \ldots, k\}$, with edges to just those subsets formed by adding or deleting one element.

9. G_e or $G_{(a,b)}$ is the graph with n vertices and only one edge $e = (a, b)$.

10. We will write a *vector* as a lower-case latin letter, possibly with a subscript, like x or w_2. We often think of an n-vector as a set of labels for the n vertices of a graph; in that case element i of vector x is written as $x(i)$, and we may write $x \in \mathbb{R}^V$ instead of $x \in \mathbb{R}^n$. In linear algebraic expressions, vectors are column vectors.

11. Two special vectors are $\mathbf{0}$, the vector of all zeros, and $\mathbf{1}$, the vector of all ones.

12. If a is a vertex then $\mathbf{1}_a$ is the *characteristic vector* of a, which is zero except for $\mathbf{1}_a(a) = 1$. Similarly if S is a set of vertices, then $\mathbf{1}_S$ is the vector that is equal to one on the elements of S and zero elsewhere.

13. If x and y are vectors of the same dimension,

$$x^T y = y^T x = \sum_{i=0}^{n} x(i)y(i)$$

is their inner product (or dot product). Thus $\mathbf{1}^T x$ is the sum of the elements of x, and $x^T x$ is the square of the 2-norm (Euclidean length) of x. If $x^T y = 0$, we call x and y *orthogonal*, and they are in fact perpendicular as vectors in $\mathbb{R}^n$.

14. If d is an n-vector, $\mathrm{Diag}(d)$ is the n-by-n diagonal matrix with the elements of d on the diagonal. If A is any n-by-n matrix, $\mathrm{diag}(A)$ is the n-vector of the diagonal elements of A.

15. **Laplacian matrix**. The Laplacian of graph G is the n-by-n matrix L whose diagonal element $L(a,a)$ is the degree of vertex a, and whose off-diagonal element $L(a,b)$ is -1 if $(a,b) \in E$ and 0 if $(a,b) \notin E$. This matrix, which we (and Spielman) just call the Laplacian, is sometimes called the *combinatorial Laplacian* to distinguish it from the normalized Laplacian below (36). Note that $L\mathbf{1} = \mathbf{0}$.

16. L_e or $L_{(a,b)}$ is the n-by-n Laplacian matrix of the graph with n vertices and only one edge $e = (a,b)$. This matrix has only four nonzero elements, two 1's on the diagonal and two -1's in positions (a,b) and (b,a); thus

$$L_{(a,b)} = (\mathbf{1}_a - \mathbf{1}_b)(\mathbf{1}_a - \mathbf{1}_b)^T.$$

The Laplacian of any graph $G = (V,E)$ is the sum of the Laplacians of its edges,

$$L_G = \sum_{e \in E} L_e.$$

17. **Laplacian quadratic form**. The *Laplacian quadratic form* (or just LQF) is $x^T L x$, where L is a particular graph's Laplacian and x is a variable n-vector. Its value for a particular vector x is

$$x^T L x = \sum_{(a,b) \in E} (x(a) - x(b))^2.$$

18. **Cut vector**. A cut vector is a vector each of whose elements is $+1$ or -1. We can think of a cut vector x as representing a *cut* that partitions the vertices of graph into two sets $S = \{\, a : x(a) = 1 \,\}$ and $V - S = \{\, a : x(a) = -1 \,\}$; then $x = \mathbf{1}_S - \mathbf{1}_{V-S}$. The LQF evaluated at a cut vector is easily seen to be four times the number of edges that cross the cut:

$$x^T L x = 4 \cdot |\{\, (a,b) \in E : a \in S \wedge b \in V - S \,\}|.$$

19. **Eigenvalues and eigenvectors**. If $Aw = \lambda w$ for any square matrix A, nonzero vector w, and scalar λ, then λ is an *eigenvalue* of A and w is an *eigenvector* associated with λ.

20. If A is square and B is nonsingular, then the eigenvalues of BAB^{-1} are the same as those of A, and the eigenvectors of BAB^{-1} are B times the eigenvectors of A.

21. Every Laplacian L is *positive semidefinite*, which (along with symmetry) implies that its n eigenvalues are nonnegative and real. Zero is an eigenvalue of L with multiplicity equal to the number of connected components of the graph G. Therefore, if G is connected, we have $0 = \lambda_1 < \lambda_2 \leq \cdots \leq \lambda_n$. In that case the eigenvector w_1 is the constant vector $\mathbf{1}/\sqrt{n}$.

22. **Fiedler value and Fiedler vector**. The Fiedler value of a graph is λ_2, its second-smallest eigenvalue, and the Fiedler vector is w_2, the associated eigenvector. The Fiedler value of a graph is also called its *algebraic connectivity.* Note that $\lambda_2 = 0$ iff the graph is not connected.

23. **Orthogonal matrix**. A square matrix Q is orthogonal if $Q^TQ = I$, that is, its inverse is its transpose. As vectors, the columns of Q have unit length and are pairwise perpendicular; the same is true of the rows of Q.

24. **Symmetric eigenvalue factorization**. If the n-by-n matrix A is symmetric, then it possesses n real eigenvalues $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ (possibly including duplicates) associated with n mutually orthogonal unit-length eigenvectors $w_1, w_2, \ldots, w_n$. If W is the matrix $[w_1\ w_2\ \ldots\ w_n]$ and Λ is the matrix $\mathrm{Diag}(\lambda_1, \ldots, \lambda_n)$ then we can summarize this as $AW = W\Lambda$ and $W^TW = I$. We also have $A = W\Lambda W^T$, whence

$$A = \sum_{i=1}^{n} \lambda_i w_i w_i^T.$$

25. **Eigenvector basis**. If symmetric A and its eigenvalues and eigenvectors are as in (24), any vector x can be written as a linear combination of eigenvectors,

$$x = \sum_{i=1}^{n} \alpha_i w_i,$$

where $\alpha_i = w_i^T x$. Multiplication by A acts termwise on such a sum:

$$A^k x = \sum_{i=1}^{n} \alpha_i \lambda_i^k w_i.$$

26. **Pseudoinverse**. If A and its eigenvalues and eigenvectors are as in (24), the pseudoinverse of A is

$$A^\dagger = \sum_{\lambda_i \neq 0} \frac{1}{\lambda_i} w_i w_i^T,$$

where the sum is taken over the nonzero eigenvalues of A. If A is nonsingular, $A^\dagger = A^{-1}$. If x is orthogonal to the null space of A (i.e. $x^T w_i = 0$ whenever $\lambda_i = 0$), then

$$A^\dagger A x = A A^\dagger x = x.$$

27. **Square root**. The positive semidefinite square root of a positive semidefinite matrix A with eigenvalues and eigenvectors as in (24) is the matrix

$$A^{1/2} = \sum_{i=1}^{n} \lambda_i^{1/2} w_i w_i^T.$$

We write the psd square root of $A^\dagger$ as

$$A^{\dagger/2} = \sum_{\lambda_i \neq 0} \lambda_i^{-1/2} w_i w_i^T.$$

28. **Rayleigh quotient**. The Rayleigh quotient of a nonzero vector x and a matrix A is

$$\frac{x^T A x}{x^T x}.$$

If $Ax = \lambda x$, then the Rayleigh quotient of x and A is λ.

29. **Rayleigh quotient theorem**. The eigenvectors of a symmetric matrix A are critical points of its Rayleigh quotient (considered as a real-valued function of an n-vector). Specifically,

$$\lambda_k = \min_{x \perp w_1, \ldots, w_{k-1}} \frac{x^T A x}{x^T x} = \max_{x \perp w_{k+1}, \ldots, w_n} \frac{x^T A x}{x^T x},$$

and the extreme values are attained at $x = w_k$. In particular, therefore, for a Laplacian L the Fiedler value is

$$\lambda_2 = \min_{\mathbf{1}^T x = 0} \frac{x^T A x}{x^T x},$$

attained at the Fiedler vector w_2.

30. **Courant-Fischer theorem** (another version of the Rayleigh quotient theorem). The eigenvalues $\lambda_1 \leq \cdots \leq \lambda_n$ of a symmetric matrix A are characterized by

$$\lambda_k = \max_{\dim \mathbb{S} = n-k+1} \min_{x \in \mathbb{S}} \frac{x^T A x}{x^T x} = \min_{\dim \mathbb{S} = k} \max_{x \in \mathbb{S}} \frac{x^T A x}{x^T x},$$

where $\mathbb{S}$ ranges over subspaces of $\mathbb{R}^n$. The extreme values are attained at $x = w_k$.

31. **Test vector**. A test vector for λ_2 is an n-vector that is orthogonal to $\mathbf{1}$. By the Raleigh quotient theorem, if v is any test vector then $\lambda_2 \leq v^T L v / v^T v$. Note that any vector x can be converted to a test vector $v = x - (\mathbf{1}^T x/n)\mathbf{1}$; in words, subtracting off the mean of any vector orthogonalizes it against the constant vector.

32. The *boundary* of a set $S \subseteq V$ of vertices, written ∂S, is the set of edges with just one endpoint in S. Formally, $\partial S = \{ (a,b) \in E : a \in S \wedge b \in V - S \}$. The number of edges in ∂S is $|\partial S|$.

33. **Isoperimetric ratio**. The isoperimetric ratio of a set $S \subseteq V$ of vertices, written $\theta(S)$, is the ratio

$$\theta(S) = \frac{|\partial S|}{|S|}.$$

This is one sort of "surface-to-volume ratio"; see the definition of conductance (35) for another. The isoperimetric ratio of a graph G, written θ_G, is the smallest isoperimetric ratio over all sets with at most half the vertices,

$$\theta_G = \min_{|S| \leq n/2} \theta(S).$$

Note that $\theta_G = 0$ if and only if G is not connected.

34. **Isoperimetric theorem**. For any set S of vertices,

$$\theta(S) \geq \lambda_2(1 - |S|/n).$$

It follows that the isoperimetric ratio of the graph is bounded in terms of the Fiedler value,

$$\theta_G \geq \lambda_2/2.$$

This says that the larger λ_2 is, the larger the surface-to-volume ratio of any relatively small set of vertices must be.

35. **Conductance**. The conductance of a set $S \subseteq V$ of vertices, written $\phi(S)$, is the ratio

$$\phi(S) = \frac{|\partial S|}{\min(d(S), d(V - S))},$$

where $d(S)$ is the sum of the degrees of the vertices in S. This is another sort of "surface-to-volume ratio"; isoperimetric number (33) measures volume just by counting vertices, while conductance measures volume by counting vertices weighted by their degrees. In class we defined conductance for unweighted graphs, but the definition extends to weighted graphs (43) with a suitable interpretation of $d(S)$. The conductance of a graph G, written ϕ_G, is the smallest conductance of any nonempty proper subset of vertices,

$$\phi_G = \min_{S \subset V} \phi(S).$$

Note that $\phi_G = 0$ iff G is not connected. ("Conductance" has a different meaning in resistive networks, as we'll see later.)

36. **Normalized Laplacian**. The normalized Laplacian of graph G is the n-by-n matrix N whose diagonal element $N(a,a)$ is equal to 1, and whose off-diagonal element $N(a,b)$ is $-1/\sqrt{d(a)d(b)}$ if $(a,b) \in E$ and 0 if $(a,b) \notin E$, where we define d to be the vector of vertex degrees of G. Another way to say it is that the normalized Laplacian is the (ordinary) Laplacian with rows and columns scaled symmetrically to make the diagonal elements equal to 1. If $D = \text{diag}(d)$ is the diagonal matrix of degrees, then

$$N = D^{-1/2} L D^{-1/2}.$$

Some authors, including notably Fan Chung in her wonderful book *Spectral Graph Theory*, use the name "Laplacian" for this matrix N instead of for our L.

37. The normalized Laplacian N is symmetric and positive semidefinite, and like the Laplacian it has 0 as an eigenvalue with multiplicity equal to the number of connected components of G. In general however N's eigenvalues and eigenvectors are different from L's. We write $0 = \nu_1 \leq \nu_2 \leq \cdots \leq \nu_n$ for the eigenvalues of N. The eigenvector corresponding to ν_1 is not the constant vector, but the vector $d^{1/2}$ of the square roots of the vertex degrees:

$$Nd^{1/2} = D^{-1/2} L D^{-1/2} d^{1/2} = D^{-1/2} L \mathbf{1} = D^{-1/2} \mathbf{0} = \mathbf{0}.$$

38. The Rayleigh quotient for the normalized Laplacian N, whose critical points determine the eigenvalues, is related to a "generalized Rayleigh quotient" for the Laplacian L. Specifically, we have

$$\frac{x^T N x}{x^T x} = \frac{y^T L y}{y^T D y},$$

where $D = \text{Diag}(d)$ is the diagonal matrix of vertex degrees and $y = D^{-1/2} x$. Thus the eigenvalues of $Nx = \nu x$ come from the generalized eigenvalue problem $Ly = \nu Dy$.

39. **Gershgorin's theorem**. If A is any square matrix (real or complex), its n eigenvalues are all contained in the union of the n disks $D_1, \ldots, D_n$ in the complex plane defined by

$$D_a = \{\alpha : |\alpha - A(a,a)| \leq \sum_{b \neq a} |A(a,b)|\}.$$

This implies, for example, that the largest eigenvalue λ_n of a Laplacian is at most twice the maximum vertex degree.

40. It follows from Gershgorin's theorem (39) that the eigenvalues of the normalized Laplacian N are always bounded by 0 and 2,

$$0 = \nu_1 \leq \nu_2 \leq \cdots \leq \nu_n \leq 2.$$

41. **Cheeger's inequalities**. The normalized Laplacian can be used to give both upper and lower bounds on the conductance,

$$\nu_2/2 \leq \phi_G \leq \sqrt{2\nu_2}.$$

Equivalently,

$$\phi_G^2/2 \le \nu_2 \le 2\phi_G.$$

The upper bound on ν_2 is analogous to the isoperimetric inequality (34). The lower bound on ν_2 is Cheeger's inequality, one of the most significant theorems of spectral graph theory. In class we stated (and mostly proved) these inequalities for unweighted graphs, but they hold for weighted graphs (43) as well; the Spielman book proves the weighted version.

42. **Cauchy-Schwarz inequality**. Just for reference, because it comes up in several of the proofs we're looking at. If x and y are n-vectors, then

$$|x^T y| \le \|x\| \, \|y\|.$$

Equivalently,

$$\Big(\sum_i x(i)y(i)\Big)^2 \le \Big(\sum_i x(i)^2\Big)\Big(\sum_i y(i)^2\Big).$$

43. **Weighted graph**. A weighted graph is an undirected graph that comes with *positive* weights on the edges, which we write $c(e)$ or $c(a, b)$. We take $c(a, b) = 0$ if (a, b) is not an edge; weights on edges are required to be strictly positive. Note that $c(a, b) = c(b, a)$. We can think of all graphs as weighted graphs; an "unweighted" graph just has edge weights all equal to 1.

44. In a weighted graph, we often interpret $d(a)$, for a vertex a, not as the number of incident edges but as the sum of the weights of the incident edges:

$$d(a) = \sum_{b \ne a} c(a, b),$$

and we often define the diagonal matrix $D = \mathrm{Diag}(d)$ as the matrix whose entries are those sums. We also (as before) write $d(S)$, where S is a set of vertices, to mean $\sum_{a \in S} d(a)$.

45. **Weighted Laplacian**. The Laplacian matrix of a weighted graph is the n-by-n matrix L whose off-diagonal element $L(a, b)$ is $-c(a, b)$ if $(a, b) \in E$ and 0 if $(a, b) \notin E$, and whose diagonal element $L(a, a) = d(a) = \sum_{b \ne a} c(a, b)$ is chosen to make the row sums zero. Like the Laplacian of an unweighted graph, we have $L\mathbf{1} = \mathbf{0}$, and indeed 0 is an eigenvalue of L with multiplicity equal to the number of connected components of the graph. For an unweighted graph, this is equivalent to our previous definition, with all edge weights equal to 1.

46. **Normalized weighted Laplacian**. The normalized Laplacian matrix of a weighted graph is the matrix N whose diagonal element $N(a, a)$ is equal to 1, and which for each edge (a, b) has symmetric off-diagonal elements $N(a, b) = N(b, a) = -c(a, b)/\sqrt{d(a)d(b)}$. Here $d(a)$ is the sum of the weights of edges incident on a. If $D = \mathrm{Diag}(d)$ is the diagonal matrix of those sums and L is the Laplacian of the weighted graph, then

$$N = D^{-1/2} L D^{-1/2}.$$

Like the normalized Laplacian of an unweighted graph, we have $Nd^{1/2} = \mathbf{0}$, and 0 remains an eigenvalue of N with multiplicity equal to the number of connected components of the graph. Again this is equivalent to our previous definition for an unweighted graph if all edge weights are equal to 1.

47. **Multiple of a graph**. If G is a (weighted) graph and $\alpha > 0$ is a constant, αG is the graph whose edge weights are all multiplied by α. The ordinary Laplacian of αG is α times the Laplacian of G,
$$L_{\alpha G} = \alpha L_G.$$
On the other hand, the normalized Laplacian of αG is the same as the normalized Laplacian of G,
$$N_{\alpha G} = N_G,$$
since the normalization wipes out the factor of α.

48. **Semidefinite ordering**. If A is a matrix, $A \succeq 0$ means that A is symmetric and positive semidefinite. Thus $L \succeq 0$ for any Laplacian L. If A and B are matrices, $A \succeq B$ means $A - B \succeq 0$. If G and H are graphs or weighted graphs, $G \succeq H$ means $L_G \succeq L_H$ (note that we are using the ordinary, un-normalized Laplacian here). Then $G \succeq H$ if and only if $x^T L_G x \geq x^T L_H x$ for all vectors x. For matrices $A \succeq 0$ and $B \succeq 0$, $A \succeq B$ implies $\lambda_k(A) \geq \lambda_k(B)$ for all k, but the converse is false. Also, $A \succeq B$ implies $B^\dagger \succeq A^\dagger$.

49. **Graph approximation**. For any constant $\alpha \geq 1$, (weighted) graph H is an α-*approximation* of (weighted) graph G if $\alpha H \succeq G \succeq H/\alpha$. This definition actually applies to all symmetric matrices, not just graph Laplacians.

50. **Krylov subspace**. The t-dimensional Krylov subspace based on a square matrix A and a vector b is
$$\mathcal{K}_t(A, b) = \operatorname{span}(b, Ab, A^2b, \ldots, A^{t-1}b).$$

51. **Symmetric QR algorithm**. This algorithm computes all the eigenvalues and all the eigenvectors of a symmetric matrix A in two phases. The first phase applies a sequence of $n-2$ elementary orthogonal transformations called *Householder reflections* to A, symmetrically from the left and right. The reflections are chosen to zero out each column in turn below its first subdiagonal element (and, because of symmetry, each row after its first superdiagonal element), giving the factorization
$$Q^T A Q = T,$$
where Q is orthogonal and T is tridiagonal (and symmetric). The second phase is iterative, and converts T to a diagonal matrix by applying a sequence of elementary orthogonal transformations called *Givens rotations* to reduce the magnitude of the off-diagonal elements by a process called "bulge-chasing" that is reminiscent of squeezing the last bit of toothpaste

from a tube. Iterations continue until the off-diagonal elements are sufficiently small to be neglible. This gives the factorization

$$V^T T V = \Lambda,$$

where V is orthogonal and Λ is diagonal. Taking $W = QV$, we then have the eigenvalue factorization

$$A = W \Lambda W^T.$$

The first phase does $O(n^3)$ work, and we can think of the second phase (modulo details about convergence and floating-point arithmetic) as doing $O(n^2)$ work. This is the workhorse method for computing all eigenvalues and eigenvectors of dense matrices, but it can't be used for very large sparse matrices both because of the n^3 work and because the first phase needs n^2 memory to store intermediate fill. See the Demmel or Trefethen/Bau textbooks for details.

52. **Lanczos iteration**. The Lanczos iteration computes the matrices Q and T above, one column at a time. It begins with an arbitrary (typically random) unit vector q_1 as the first column of Q, and at step t it computes the vector $v = Aq_t$, orthogonalizes v against columns q_t and q_{t-1}, and scales the orthogonalized vector to unit length. Because T is symmetric and tridiagonal, the resulting vector is actually orthogonal to all columns q_1 through q_t, and it becomes column q_{t+1}. The coefficients of the orthogonalization become the entries of T. See the class notes (or Demmel or Trefethen/Bau) for the details of the formulas. The Lanczos iteration can be viewed as building orthogonal bases for the Krylov subspaces $K_t(A, q_1)$. Unlike the QR algorithm (51), the only thing Lanczos needs to do with the matrix A is to multiply it by vectors, which is useful when A is sparse.

53. **Lanczos algorithm**. The Lanczos algorithm computes approximations to some of the eigenvalues of A. It first performs some number t (typically much less than n) of Lanczos iterations to get an n-by-t matrix Q_t with orthonormal columns (i.e. $Q_t^T Q_t = I_t$) and a t-by-t symmetric tridiagonal matrix $T_t = Q_t^T A Q_t$. It then uses the second ("toothpaste-squeezing") phase of the symmetric QR algorithm (51) to diagonalize $T_t = V_t \Theta V_t^T$, where V_t is t-by-t and orthogonal, and Θ is diagonal. Then the numbers $\theta_1, \theta_2 \ldots, \theta_t$ on the diagonal of Θ (the eigenvalues of T_k) are *Ritz values* for A, and the columns of $Q_k V_k$ are the corresponding *Ritz vectors.* (There is a lot of numerical-algorithm engineering involved in a practical implementation of Lanczos; see Demmel or the 1994 Grimes/Lewis/Simon SIMAX paper for details.)

54. **Ritz values**. The Ritz values of a matrix approximate some of its eigenvalues. The whole story is rather involved; see Demmel for more of it. In exact real arithmetic, if A has no multiple eigenvalues, the Ritz values after $t = n$ steps are equal to the n eigenvalues of A. Multiple (or very close) eigenvalues are tricky; block versions of Lanczos can be used here. In exact arithmetic, at stage t, every Ritz value is guaranteed to be within β_t of some eigenvalue, where β_t is the subdiagonal element in column t of T. Generally speaking, the extreme Ritz values (those closest to $+\infty$ and $-\infty$) tend to converge quickly to the extreme eigenvalues.

55. **Schur complement**. If matrix M is partitioned into a 2-by-2 block form

$$M = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$$

such that A is square and nonsingular, then the Schur complement of A in M (sometimes also called the "Schur complement on D") is the matrix

$$D - CA^{-1}B.$$

56. **Incidence matrix**. If G is a graph with n vertices and m edges, an incidence matrix of G is an n-by-m matrix U with a column for each edge of G. The column for edge (a, b) contains two nonzeros, a 1 and a -1, one in row a and one in row b. The incidence matrix is not unique; permuting its columns or negating some of its columns produces another incidence matrix. Any incidence matrix U is related to the Laplacian L_G by

$$L_G = UU^T.$$

57. If G is a weighted graph, its incidence matrix is the same as above (56), without weights. If $C = \mathrm{diag}(c)$ is the diagonal matrix of edge weights, in the same order as the columns of U, then the weighted Laplacian L_G satisfies

$$L_G = UCU^T.$$

58. **Augmented matrix**. An augmented matrix of an unweighted graph with n vertices and m edges is a symmetric $(n + m)$-by-$(n + m)$ matrix defined in block form as

$$\begin{pmatrix} I & U^T \\ U & 0 \end{pmatrix},$$

where U is an incidence matrix and I is the m-by-m identity matrix. An augmented matrix of a weighted graph is

$$\begin{pmatrix} R & U^T \\ U & 0 \end{pmatrix},$$

where $R = \mathrm{diag}(1/c) = C^{-1}$ is the diagonal matrix of inverse edge weights. The Schur complement of R is then $-UR^{-1}U^T = -L$, the negative of the weighted Laplacian.

59. **Resistive network**. A resistive network is a weighted graph with n vertices interpreted as nodes of an electrical circuit and m edges interpreted as resistors joining pairs of nodes. If the resistor at edge e has resistance $r(e)$, the edge's weight is the inverse resistance $c(e) = 1/r(e)$. (Inverse resistance is often called "conductance," whence the letter c, but we will not use the term in this context to avoid confusion with the unrelated notion of graph conductance in (35) above.)

60. In a resistive network G, suppose a current $i(a)$ is injected at each vertex a, where $\mathbf{1}^T i = 0$ so the same total current is injected and removed from the network as a whole. For each edge $(a, b) \in E$ with $a < b$, let $f(a, b)$ be the *current* or *flow* along edge (a, b). Then *Ohm's law*, or current times resistance equals voltage, says

$$v(a) - v(b) = f(a, b)r(a, b) \quad \text{for all } (a, b) \in E.$$

Kirchoff's current law, or current entering a node equals current leaving the node, says

$$\sum_{b:(a,b)\in E} f(a, b) = i(a) \quad \text{for all } a \in V.$$

The two laws can be combined in one augmented system (58) as

$$\begin{pmatrix} R & U^T \\ U & 0 \end{pmatrix} \begin{pmatrix} f \\ -v \end{pmatrix} = \begin{pmatrix} 0 \\ i \end{pmatrix},$$

where U is the incidence matrix of G with appropriate signs, and R is the m-by-m diagonal matrix of edge resistances. The Schur complement of R is $-UR^{-1}U^T$, which is the negative Laplacian $-L$. A step of block Gaussian elimination on the augmented system thus leads to the Laplacian linear system

$$Lv = i$$

relating the node voltages to the externally injected currents.

61. **Effective resistance**. In a resistive network, the effective resistance between two vertices a and b, written $R^{\text{eff}}(a, b)$, is the positive difference in voltage between a and b when one unit of current is injected at a and extracted at b. That is, if $Lv = \mathbf{1}_a - \mathbf{1}_b$ for $a < b$, then

$$R^{\text{eff}}(a, b) = (\mathbf{1}_a - \mathbf{1}_b)^T v = (\mathbf{1}_a - \mathbf{1}_b)^T L^\dagger (\mathbf{1}_a - \mathbf{1}_b) = v^T Lv,$$

where $\mathbf{1}_a - \mathbf{1}_b$ is the vector whose k'th element is equal to 1 when $k = a$, equal to -1 when $k = b$, and equal to zero elsewhere. We write $R^{\text{eff}}_G(a, b)$ if the graph is not clear from context.

62. Let G and H be resistive networks on the same number of vertices. If $H \succeq G$, then for every pair of vertices a, b we have $R^{\text{eff}}_G(a, b) \geq R^{\text{eff}}_H(a, b)$.

63. **Symmetric Gaussian elimination**. If L is the Laplacian of a connected weighted graph G with $n > 1$ vertices, then the result of one step of Gaussian elimination on L is the $n-1$-by-$n-1$ matrix

$$L_B = L(2:n, 2:n) - \frac{1}{L(1,1)} L(2:n, 1)L(1, 2:n),$$

which is a rank-1 modification to the submatrix $L(2:n, 2:n)$. Note that L_B is the Schur complement of $L(1, 1)$ in L. Also, L_B is the Laplacian matrix of the (suitably weighted) graph obtained from G by adding edges between all neighbors of vertex 1, and then deleting vertex 1 and all its incident edges.

If B is any set of vertices of G and $A = V - B$, and Gaussian elimination is used to eliminate all the vertices of A in any order, the result is the matrix

$$L_B = L(B, B) - L(B, A)L(A, A)^{-1}L(A, B),$$

which is the Schur complement of $L(A, A)$ (or on $L(B, B)$) in L. Then L_B is the Laplacian matrix of the (suitably weighted) graph obtained from G by adding an edge between every pair of vertices in B that are connected in G by a path of vertices in A.

64. **Equivalent networks**. If G is a connected weighted graph whose vertices are partitioned into a set B of "boundary" vertices and a set $V - B$ of "interior" vertices, and if L_B is the Schur complement on $L_G(B, B)$ as above (63), then the effective resistances between vertices of B are the same in L_G and in L_B:

$$R^{\text{eff}}_{L_B}(a, b) = R^{\text{eff}}_{L_G}(a, b) \text{ for all } a, b \in B.$$

Note that L_B is *not* a submatrix of L_G, and the graph of L_B generally has more edges than the induced subgraph of G on vertices B.

65. **Cholesky factorization**. If A is any positive definite matrix (or, with some care, a positive semidefinite matrix), the Cholesky factorization is $A = R^T R$, where R is an upper triangular matrix with positive diagonal elements (non-negative in the semidefinite case). The Cholesky factorization of any n-by-n matrix can be computed in $O(n^3)$ time and $O(n^2)$ memory; some but by no means all sparse matrices have better bounds.

66. **Cholesky graph game**. Given positive (semi)definite A with undirected graph $G(A)$, the undirected graph $G^+(A) = G(R + R^T)$ of the Cholesky factors of A is obtained as follows:

```
for a = 1 : n
  mark vertex a;
  add "fill" edges between the umarked neighbors of vertex a;
end for
```

(This gives the nonzero structure of R but not the nonzero values.) We are free to mark the vertices in any order; choosing a different order corresponds to applying a permutation symmetrically to the rows and columns of A.

67. **Parter's theorem**. If the graph $G(A)$ is a tree, a vertex ordering exists for which the Cholesky factorization adds no fill and solving $Ax = b$ takes only $O(n)$ time and memory.

68. **Nested dissection**. If the graph $G(A)$ is the $\sqrt{n}$-by-$\sqrt{n}$ grid graph, the best possible elimination ordering has $O(n \log n)$ fill, for which Cholesky takes $O(n^{3/2})$ time. The same upper bounds hold for any planar graph. For the three-dimensional grid graph, the best possible fill is $O(n^{4/3})$ and Cholesky takes $O(n^2)$ time.

69. **Matrix polynomials**. If $p(z) = \sum_{k=0}^{t} \beta_k z^k$ is a polynomial in a scalar variable z, and A is a matrix, we write $p(A) = \sum_{k=0}^{t} \beta_k A^k$. Note that every vector in the Krylov subspace $\mathcal{K}_{t+1}(A, b)$ is $p(A)b$ for some polynomial $p(z)$ of degree t.

70. If A is a symmetric matrix, $Ax = b$, and $q(z) = 1 - zp(z)$ is a polynomial with $q(0) = 1$ and $q(\lambda) = 0$ for every eigenvalue λ of A, then $p(A)b = x$. Therefore x is in the Krylov subspace $\mathcal{K}_t(A, b)$, where t is the number of *distinct* eigenvalues of A.

71. The *condition number* of a square matrix A is $\kappa(A) = \|A\| \, \|A^{-1}\|$, interpreted as ∞ if A is singular. If A is symmetric and positive definite, $\kappa(A) = \lambda_n/\lambda_1$ is the ratio of the extreme eigenvalues.

72. **Conjugate gradient**. The conjugate gradient algorithm (or CG) solves $Ax = b$, where $A \succ 0$ is a symmetric, positive definite matrix (see Shewchuk for details). Each iteration performs one matrix-vector multiplication with A and some vector arithmetic, taking $O(n+m)$ time per iteration if A has m nonzeros. The relative error in the approximate solution x_j is bounded by

$$\frac{||x_j - x||_A}{||x||_A} < \epsilon$$

after

$$j = O(\sqrt{\kappa(A)} \, \log(1/\epsilon))$$

iterations (in exact arithmetic), where $\kappa(A) = \lambda_n/\lambda_1$ is the condition number of A and $||v||_A = (v^T A v)^{1/2}$ is the A-norm. With some care, CG can also be used for a positive semidefinite matrix whose null space is known, e.g. a weighted graph Laplacian.

73. **Preconditioned conjugate gradient**. The preconditioned conjugate gradient algorithm (or PCG) solves $Ax = b$ by applying CG to the linear system

$$(B^{-1/2} A B^{-1/2})(B^{1/2} x) = B^{-1/2} b,$$

where A and B are symmetric positive definite. Each iteration of PCG performs one matrix-vector multiplication with A, one linear system solve with B, and some vector arithmetic. Matrix B is called a *preconditioner* for A, and may or may not be formed explicitly. A good preconditioner satisfies two criteria:

- It should be “easy” to solve the linear system $By = z$ for y.
- The condition number $\kappa(B^{-1/2} A B^{-1/2}) = \kappa(AB^{-1})$ should be smaller than $\kappa(A)$.

With some care, PCG can also be used with positive semidefinite matrices A and B if they have the same null space.

74. For a symmetric positive semidefinite matrix A, the *finite condition number* is $\kappa_f(A) = \lambda_n/\lambda_k$, where λ_k is the smallest nonzero eigenvalue. For example, if L is the Laplacian of a connected graph, $\kappa_f(L) = \lambda_n/\lambda_2$ is the relevant condition number for the convergence of conjugate gradient.

75. Let A and B be symmetric positive semidefinite matrices with the same null space (e.g., weighted Laplacians of two connected graphs on the same vertices). The *finite condition number* $\kappa_f(A, B)$ is $\kappa_f(AB^\dagger)$, which is the relevant condition number for the convergence of conjugate gradient on $Ax = b$ with preconditioner B. Note that $\kappa_f(A, B) = \kappa_f(B, A)$.

76. Let A and B be two symmetric positive semidefinite matrices with the same null space. If $\alpha B \preceq A \preceq \beta B$, then $\alpha \leq \lambda \leq \beta$ for every nonzero eigenvalue λ of $AB^\dagger$, and therefore $\kappa_f(A, B) \leq \beta/\alpha$.