Proceedings of the 2nd International Conference on Mathematical Physics and Computational Simulation
DOI: 10.54254/2753-8818/39/20240638

Effective resistance, random walk and algorithms

Ruihan Zhang
The Athenian School, Danville, CA, USA

25rzhang@athenian.org

Abstract. Efficiently determining the effective resistance of large-scale electrical networks is crucial for optimizing energy distribution and assessing network robustness. In this paper, we propose a novel approach that combines random walk simulations and advanced linear algebra techniques to compute the effective resistance of complex electrical networks. Our method leverages the concept of random walks to simulate the flow of current through the network, capturing its behavior under various conditions. Subsequently, we employ graph Laplacian-based linear algebra algorithms to analyze the resulting data, enabling accurate computation of the effective resistance. In our model, we aim to deploy these combined algorithm to develop a novel method to deal with such methods.

Keywords: Linear Algebra, Algorithms, Electrical Networks, Random Walk.

1. Introduction

Consider a large network of electrical resistors. Each resistor might be wired (in series or parallel or both) onto other resistors forming a complicated device. Now choose any two points on the network. What would be the effective resistance between them? Is there an algorithm to find out? If so, how efficient is this algorithm?

This seemingly complicated and intractable problem admits a surprisingly clean and beautiful solution. It not only involves graph theory, which is used to model the network mathematically, but also linear algebra and algorithms for matrix factorization. Hidden in the middle is some deep ideas about probability and random walk. This article explains the process of reaching this beautiful solution. All necessary background is provided to ensure a self-contained account.

Section 2 defines a graph and its laplacian. This gives the first glimpse of our problem's connection to linear algebra. Important concepts such as symmetric and positive semidefinite matrices are also introduced.

Section 3 delves into the world of algorithm. It starts with the familiar Gaussian elimination. Then concrete algorihtm for LU decomposition is explained. Finally this section culminates with the famous Cholesky factorization algorithm.

Section 4 takes a rather different approach to electrical network. It focuses on random walk which is a topic in probability theory. Surprisingly, random walk and its related topics is crucial to understanding electrical networks and its effective resistance.

Section 5 puts all the ingredients together and produce a satisfying and elegant solution to this industrial problem. To illustrate the theory, examples are worked out in detail.

Section 6 concludes the article and provide directions for further investigation.

2. Graphs and their Laplacian

2.1. Graphs

Definition 1. A graph $G = (V, E)$ is a finite set V of vertices, and a set $\mathcal{E}$ of edges defined as pairs of distinct vertices. When there is no distinction between the pairs of vertices (x_i, x_j) and (x_j, x_i), the edges are represented by unordered pairs, and the graph is **undirected**. In an undirected graph G with vertices $\{x_1, x_2, \cdots, x_n\}$, $\{x_i, x_j\}$ is an edge if and only if there is a line in G that connects the vertices x_i and x_j. If, however, the pairs are ordered, the graph is a **directed** graph.

Definition 2. Define $G^{'}$ to be a **subgraph** of another graph G if the vertex set and edge set of $G^{'}$ are subsets of the vertex set and edge set of G, respectively.

Definition 3. A sequence of k edges in an undirected graph G

$$u_0 \leftrightarrow u_1 \leftrightarrow u_2 \leftrightarrow \cdots \leftrightarrow u_{k-1} \leftrightarrow u_k$$

is defined to be a **walk** of length k.

Definition 4. Define a **path** to be a non-empty graph $G_p = (V, E)$ in the form of:

$$V = \{x_0, x_1, \cdots, x_n\}, E = \{\{x_0, x_1\}, \{x_1, x_2\}, \cdots, \{x_{n-1}, x_n\}\}$$

where all vertices x_i are distinct from $i \in [0, n]$.

The definition of a path is similar to that of the walk, those two are similar, except for the fact that a path is a walk with no repeating vertices.

Definition 5. A non-empty graph G is **connected** if any two of its vertices $x_0, x_1, \cdots, x_n$ are contained in a path G_p in G.

Definition 6. The **connectivity** of a graph is defined to be the minimum amount of elements (edges or vertices) that need to be removed from the graph to separate the remaining vertices into two or more separate subgraphs.

2.2. Adjacency Matrices of Graphs

Definition 7. In a graph $G = (V, E)$, define two vertices $x_i, x_j \in V$ to be **adjacent** if $\{x_i, x_j\} \in E$.

Definition 8. Define the **adjacency matrix** A_G of a graph G to be defined by

$$a_{ij} = \begin{cases} 1 & \text{if } (i,j) \in E \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where a_{ij} are the entries of the matrix.

One use of the adjacency matrix of a graph is to evaluate the number of walks of different length connecting two vertices in the graph.

2.3. Laplacian Matrices of Graphs

Definition 9. Define the **orientation** of an undirected graph G to be the assignment of direction to each edge connecting two vertices x_i and x_j; note that we may reference undirected graphs with orientations to be **oriented graphs**.

It turns out that, the Laplacian matrix of an undirected graph is independent of the graph's orientation.

Definition 10. Define the **laplacian matrix** L_G of a graph G to be defined by

$$l_{ij} = \begin{cases} -1 & \text{if } (i,j) \in E \\ d(i) & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where l_{ij} are the entries of the matrix, and $d(i)$ is the degree of the vertex x_i.

Note that $L_G = D_G - A_G$, where D_G is the diagonal matrix of vertex degrees and A_G is the previously defined adjacency matrix.

We may also reference the Laplacian matrix by its quadratic form:

$$xL(G)x^t = \Sigma(x_i - x_j)^2 \tag{3}$$

where, $x \in \{x_1, x_2, \cdots, x_n\}$; and the summation is done over all ordered pairs of (i, j) where $i < j$ and $\{v_i, v_j\} \in E$ [4].

This definition of the Laplacian matrix is indeed, equivalent to the one preceding it. However, we find that many elementary properties of the Laplacian matrix follow easily from the new definition. One of which being that $L(G)$ is symmetric, and positive semi-definite. Recall the following definitions of symmetric and positive semi-definite:

Definition 11. For a graph $G = (V, E)$, $L_G = \sum_{\{u,v\}\in E} L_{G_{\{u,v\}}}$.

Definition 12. A **symmetric** matrix A satisfies the condition $A_{ij} = A_{ji}$ for all values of i and j.

Definition 13. A symmetric matrix A with real entries is **positive semidefinite** if the scalars of $\vec{z}^T M\vec{z}$ and $\vec{z}^* M\vec{z}$ are nonnegative, for every nonzero real column vector $\vec{z} \in \mathbf{R}^n$. [7]

Theorem 14. *Every eigenvalue in a Laplacian matrix L_G is nonnegative.*

Proof. Suppose λ is an eigenvalue and $\vec{z} \in \mathbf{R}^n$ is a nonzero eigenvector of λ. Then we reach the following:

$$\begin{aligned}\vec{z}^T L_G \vec{z} &= \vec{z}^T \lambda \vec{z} \\ &= (\vec{z}^T \vec{z})\lambda\end{aligned}$$

And because we are given that $(\vec{z}^T \vec{z}) \geq 0$ and $\vec{z}^T L_G \vec{z} \geq 0$, it is obvious that we can derive $\lambda = 0$. [1]

From now, we could use the eigenvalues of a Laplacian matrix L_G to state several useful characteristics of any graph G, we will denote the eigenvalues to be $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$.

Lemma 15. *For any graph G, $\lambda_1 = 0$*

Proof. As previously mentioned, L_G is a semidefinite matrix, which we thus know that $\lambda \geq 0$. Applying the quadratic form definition, we know that:

$$xL(G)x^t = \Sigma(x_i - x_j)^2 \geq 0$$

and from the characterization of eigenvalues, we get:

$$\lambda_1 = \min_{||x||=1} xL(G)x^t = \min_{||x||=1} \textstyle\sum_{i\sim j}(x_i - x_j)^2 \geq 0$$

Now, since $\sum_{i\sim j}(x_i - x_j)^2 \geq 0$, all we have to do is to construct vectors x such that $\sum_{i\sim j}(x_i - x_j)^2 = 0$. This could be done if we simply let $x = 1$ and the resulting vector to be a vector with all 1s, hence $(1, 1, 1, \cdots, 1, 1)$. From that, we get $L_G x = (0, 0, 0, \cdots, 0, 0)$, thus proving that 0 is an eigenvalue of L_G. In fact, this construction also demonstrates that the all 1s vector is always the first eigenvector of a Laplacian L_G. [2]

The second smallest eigenvalue λ_2 of a Laplacian matrix L_G of a graph G corresponds to the connectivity of the graph G, and is hence used to partition the graph. Now, we will analyze that specific eigenvalue and prove its usability.

Lemma 16. *$\lambda_2 > 0$, if and only if graph G is connected.*

Proof. We will first prove that $\lambda_2 = 0$, if and only if graph G is disconnected. Again, we start by using its quadratic definition:

$$\lambda_2 = 0, \text{ then } \exists x \perp 1 \text{ such that}$$

$$xL(G)x^t = \textstyle\sum_{i \sim j}(x_i - x_j)^2 = 0 \leftrightarrow (x_i - x_j)^2 = 0 \, \forall i \sim j.$$

And now, use proof by contradiction to complete this proof.

Suppose that G is connected under this circumstance. Then, $(x_i - x_j)^2 = 0 \, \forall i \sim j$ is true if and only if $x_i = x_j \; \forall i, j$. But, this means that i, j, which are in the same connected section of G, we have the above equality $x_i = x_j$. However, since this is the second eigenvector, we know that:

$$x \perp 1 \; : \langle x, 1\rangle = \textstyle\sum_i x_i = 0.$$

So it shows that x is actually not a constant vector, so $\exists i, j$ such that $x_i \neq x_j$. So, there is a contradiction and G must be disconnected.

Now, we will prove $\sigma_2 > 0$ if graph G is connected.

Since 0 is an eigenvalue of L_G, let $\vec{z}$ be a nonzero eigenvector of 0. Then we have the following:

$$\vec{z}^T L_G \vec{z} = \vec{z}^T \cdot 0 = 0$$

Therefore, we have:

$$\vec{z}^T L_G \vec{z} = \textstyle\sum_{\{u,v\} \in E}(z_u - z_v)^2 = 0$$

This equation implies that for any u, v such that $\{u, v\} \in E$, $z_u - z_v = 0$ and thus $z_u = z_v$. And since we are performing these operations under a connected graph G, this means that for all $i, j \in V$, $z_i = z_j$. So, the vector $\vec{z}$ can be expressed as the all 1s vector multiplied by a constant scalar. For such, we can write:

$$\vec{z} = \textstyle\sum_{i=1}^{k} a_i u^i$$

This implies that every eigenvector of L corresponding to the eigenvalue 0 is contained in the subspace spanned by $\{u^1, u^2, \cdots, u^k\}$. Therefore, the multiplicity of eigenvalue 0 is 1, and it thus follows that $\lambda_2 \neq 0$, thus $\lambda_2 > 0$.

It is also important to remark that the multiplicity of the eigenvalue 0 is the number of connected components of graph G.

3. LU Factorization and Cholesky Factorization

This section introduces matrix factorization.

3.1. Gaussian Elimination

First, there are three types of triangular systems to consider:

- Define a **diagonal matrix** to be a matrix where $a_{ij} = 0$ for all $i \neq j$. We often denote this matrix as D.
- An **upper triangular matrix** is where $a_{ij} = 0$ for all $i > j$. We often denote this matrix as U.
- A **lower triangular matrix** is where $a_{ij} = 0$ for all $i < j$. We often denote this matrix as L.

The process of eliminating variables from the equations, or, equivalently, zeroing entries of the corresponding matrix, in order to reduce the system to upper triangular form is called **Gaussian Elimination**.

The conventional way of describing Gaussian elimination is based on the systematic column-by-column elimination of the entries in the lower triangular part of a matrix A. We yield the following elimination sequence from the column elimination matrices:

$$A = A^{(1)}, A^{(2)}, \cdots, A^{(n)}$$

These matrices $A^{(n)}$ could be generalized as partially eliminated matrices as follows:

$$A^{(1)} \to A^{(2)} = C_1 A^{(1)} \to A^{(3)} = C_2 C_1 A^{(1)} \to \cdots \to A^{(n)} = C_{n-1} C_{n-2} \cdots C_2 C_1 A^{(1)}$$

Recall that the kth partially eliminated matrix is $A^{(k)}$. For all **active entries** in $A^{(k)}$, we denote them to be $a_{ij}^{(k)}$, for $1 \leq k \leq i, j \leq n$, and the submatrix with dimensions $(n-k+1) \times (n-k+1)$ submatrix is defined to be the matrix's **active submatrix**.

The inverse of each C_k is the unit lower triangular matrix that is obtained by changing the sign of all the off-diagonal entries, and because the product of unit lower triangular matrices is a unit lower triangular matrix, it is quite obvious that:

$$A = A^{(1)} = C_1^{-1} C_2^{-1} \cdots C_{n-1}^{-1} A^{(n)} = LU$$

Where, the unit lower triangular matrix $L = C_1^{-1} C_2^{-1} \cdots C_{n-1}^{-1}$, and the upper triangular matrix $U = A^{(n)}$. If A is a symmetric positive definite matrix, then, by setting $U = DL^T$, we get the following definition of LU Factorization:

Definition 17. LU Factorization expression:

$$A = LDL^T \tag{4}$$

Alternatively, this could be written as the Cholesky factorization.

Definition 18. Cholesky Factorization expression:

$$A = (LD^{\frac{1}{2}})(LD^{\frac{1}{2}})^T \tag{5}$$

[5] Where the lower triangular matrix, $LD^{\frac{1}{2}}$, has all positive diagonal entries.

3.2. LU Factorization and Algorithms

In this section, the derivation of the LU Factorization will be discussed along with further implications of it.

3.2.1. Using Gaussian Elimination First, we will aim to derive LU factorization's expression using Gaussian Elimination.

Recall from the previous section, for each matrix C_k from $k \in [1, n-1]$, it is also a **unit lower triangular matrix**, as all of its diagonal entries are equal to 1. Note two commonly known properties for all unit lower triangular matrices:

- The product of two unit lower/upper triangular matrices is unit lower/upper triangular.
- The inverse of an unit lower/upper triangular matrix is also unit lower/upper triangular.

The inverse of each C_k is easily computed, we have:

$$L^{(k)} = (C_k)^{-1} = \begin{bmatrix} 1 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & 1 & 0 & & & & & & 0 \\ \vdots & \ddots & 1 & \ddots & & & & & \vdots \\ \vdots & & 0 & \ddots & \ddots & & & & \vdots \\ \vdots & & \vdots & c_{k+1,k} & \ddots & \ddots & & & \vdots \\ \vdots & & \vdots & \vdots & 0 & \ddots & \ddots & & \vdots \\ \vdots & & \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \vdots & \vdots & \vdots & & \ddots & 1 & 0 \\ 0 & \cdots & 0 & c_{nk} & 0 & \cdots & \cdots & 0 & 1 \end{bmatrix}$$

It follows from this, that, if we express C as $C = C_{n-1} \cdots C_2 C_1$, then C is unit lower triangular, and $CA = U$, where the matrix U is upper triangular. From that, we could derive $A = C^{-1}U = LU$, where we express LU as:

$$L = L^{(1)} \cdots L^{(n-1)} = (C_1)^{-1}(C_2)^{-1} \cdots (C_{n-1})^{-1}$$

which is also unit triangular. Rearranging:

$$(C_1)^{-1}(C_2)^{-1} \cdots (C_{n-1})^{-1}U = (C_1)^{-1}(C_2)^{-1} \cdots (C_{n-1})^{-1}C_{(n-1)}C_{(n-2)} \cdots C_1 A = A$$

Simplifying, we derive:

$$L^{(1)}L^{(2)} \cdots L^{(n-1)}U = LU = A.$$

And this is the LU decomposition, or the LU factorization of a matrix A.

3.2.2. Bordering The generic LU factorization scheme does not cover all possible approaches. An alternative method is factorization by **bordering**.

The algorithm works as the following:

First, we set all diagonal entries of L to be equal to 1, and assume the first $k-1$ rows of L and the first $k-1$ columns of U where $1 < k \leq n$, have been computed (which are $L_{1:k-1,1:k-1}$ and $U_{1:k-1,1:k-1}$).

At step k, we use the following derivation for each term and factor:

$$A_{1:k,1:k} = \begin{pmatrix} A_{1:k-1,1:k-1} & A_{1:k-1,k} \\ A_{k,1:k-1} & a_{kk} \end{pmatrix} = \begin{pmatrix} L_{1:k-1,1:k-1} & 0 \\ L_{k,1:k-1} & 1 \end{pmatrix} * \begin{pmatrix} U_{1:k-1,1:k-1} & U_{1:k-1,k} \\ 0 & u_{kk} \end{pmatrix}$$

Equating terms, the lower triangular part of row k of L and the upper triangular part of column k of U are obtained by solving the following equations:

$$L_{k,1:k-1}U_{1:k-1,1:k-1} = A_{k,1:k-1},$$
$$L_{1:k-1,1:k-1}U_{1:k-1,k} = A_{1:k-1,k}$$

Where, the diagonal entry $u_k k$ is given by:

$$u_{kk} = a_{kk} - L_{k,1:k-1}U_{1:k-1,k}$$

In this case obviously $u_{11} = a_{11}$. [6]

3.2.3. Summary To summarize, there are many algorithms that would result in the final expression of LU decomposition; the Gaussian Elimination method and the bordering method are two of which. Technically speaking, anything that factors a matrix into LU, with L being lower triangular and U being upper triangular passes as LU decomposition.

Being more specific, to identify the variants, names that derive from the order in which the indices are assigned to the loops can be used. The kij and kji variants are called **submatrix LU factorizations**. The schemes jik and jki compute the factors by columns and are called **column factorizations**. The final two are **row factorizations** because they proceed by rows. A row factorization can be considered as a column LU factorization applied to A^T . These algorithms are not discussed in this work but is also widely applicable.

3.3. Cholesky Factorization

Sometimes, it is preferrable to use the Cholesky factorization of a positive semidefinite matrix A, which can be expressed as:

$$A = UU^T$$

Where, U is a lower triangular matrix with all positive diagonal entries and could be written as $LD^{\frac{1}{2}}$. For mostly the rest of this section, we will be referring to the lower triangular matrices in all Cholesky factorizations to be U.

Generally, because A is factored into two matrices that are the transpose of one another, the process of computing the Cholesky Factorization requires about half as many floating-point operations as that of the LU Decomposition.

The very general algorithm for all Cholesky factorization can be derived by matching every entry of the UU^T matrices with the corresponding ones of A. The specific relation between the two is as follows:

$$a_{ik} = \sum_{j=1}^{k} u_{ij}u_{kj}$$

For all $i, k \in (1, 2, 3, \cdots, n)$ such that $k \leq i$.

This relation can be used to obtain formulas for the entries of U, when entries of A are examined in the right order.

3.3.1. Algorithm for Cholesky Factorization If A is a positive definite matrix, then its Cholesky factorization $A = UU^T$ can be obtained as follows:

(i) Carry A to an upper triangular matrix U_1 with positive diagonal entries using row operations, where each of which adds a multiple of a row to a lower row.

(ii) Obtain U from U_1 by dividing each row of U_1 by the square root of the diagonal entry in that row.

3.3.2. Proof of the Algorithm Per usual, let $A = UU^T$, and let $D = diag(d_1, d_2, \cdots, d_n)$ be the common diagonal of U and U^T. Then, $U^T D^{-1}$ is lower triangular with its diagonal entries all equal to 1. Hence, $(U^T D^{-1})^{-1}$ must also be a matrix with all its diagonal entries equal to 1, and thus, $I_n \to L$ By a sequence of row operations each of which adds a multiple of a row to a lower row. Then, $A \to LA$ by the same sequence of row operations. Since we have $LA = [D(U^T)^{-1}][U^T U] = DU$ is upper triangular with positive entries on the diagonal, and thus this shows that the first step of the algorithm is possible.

Now we turn to proving step two, we will now prove the conciseness of this proof. First, let $A \to U_1$ so that $U_1 = L_1 A$. Since A is symmetric, we get:

$$L_1 U_1^T = L_1(L_1 A)^T = L_1 A L_1^T = U_1 L_1^T$$

And again, let D_1 denote the diagonal of U_1. Then the above equation yields $L_1(U_1^T D_1^{-1}) = U_1 L_1^T D_1^{-1}$. This satisfy the characteristics of being upper triangular and all diagonal entries equal to 1, so it is the same as I_n. And now, set new diagonal $D_2 = diag(\sqrt{e_1}, \sqrt{e_2}, \cdots, \sqrt{e_n}$, so that $D_2^2 = D_1$. And if we set $U = D_2^{-1} U_1$, we have

$$U^T U = (U_1^T D_2^{-1})(D_2^{-1} U_1) = U_1^T (D_2^2)^{-1} U_1 = (L_1^{-1}) U_1 = A \text{ [9]}$$

This essentially proves the second step as $U = D_2^{-1} U_1$ is derived by dividing each row of U_1 by the square root of its respective diagonal entry.

4. Electrical network

Electrical network is another name for a weighted graph. An electrical network consists of an ordinary graph $G = (V, E)$ together with a conductance (or weight) function $c : E \to (0, \infty)$. It is customary to define resistance $r : E \to (0, \infty)$ as the reciprocal of the conductance. An unweighted graph can be viewed as an electrical network where $c = 1$ for all edges. One encounters electrical networks all the time. They could be literal networks with different resistors on edges such as the power grid. Or they may model complex network of social relations. In fact, all Markov processes can be represented as electrical networks. One classic problem for an electrical network is finding the effective conductance (or resistance) between two given vertices on the network.

4.1. Random walk

Now, let S_n represent the position of the random walk at time n. Formally, a simple random walk in one dimension can be defined as follows:

(i) The random walk starts at a specific initial position, S_0.

(ii) At each time step, the walker takes a step to the left with probability p or a step to the right with probability $1 - p$

(iii) The size of each step is usually fixed to 1 (i.e., the walker moves one unit to the left or one unit to the right).

The position S_n at time n is the sum of the individual steps taken by the walker up to that point:

$$S_n = S_0 + X_1 + X_2 + \cdots + X_n$$

where X_i represents the size of the ith step taken by the walker at time i.

A random walk on a graph is a process that begins at some vertex $v \in V$, and at each time step moves to another vertex. The vertex the walk moves to is chosen uniformly at random among the neighbors of the present vertex. Similarly for an electrical network, it moves to a neighbor with probability proportional to the conductance of the corresponding edge.

Note the starting vertex v is important. An event can have very different probability depending on the starting vertex of the random walk. Therefore, we indicate it with a subscript when discussing probability. For example $\mathbb{P}_v(E)$ denotes the probability some event E happens during a random walking starting at v. Observe that for a fixed event E, $\mathbb{P}_v(E)$ can be viewed as a function from V to $[0, 1]$. The most notable such function is the so-called unit voltage function:

$$F(v) := \mathbb{P}_v(\tau_B < \tau A),$$

where A, B are two vertices and τ_u is the time the random walk first reach some vertex u. In plain English, $F(v)$ is the probability that a random walk starting at v reaches some vertex B before it reaches A. Clearly, $F(B) = 1$ and $F(A) = 0$. This function will be important later.

4.2. Functions and 1-forms

This subsection starts by defining two useful concepts on a network. Then two operations are given to relate them. Surprisingly, laplacian of network (defined similarly as graph laplacian but replacing degrees with sums of conductance) appears as their composition.

A function on some network is simply a function $F : V \to \mathbb{R}$ on its vertices. We have encountered a class of functions $\mathbb{P}_v(E)$ from the previous section.

4.3. Effective resistance

The Laplacian matrices of graphs arise when one models electrical flow in networks of resistors. The vertices of a graph correspond to points at which we may inject or remove current and at which we will measure potentials. The edges correspond to resistors, with the weight of an edge being the reciprocal of its resistance. If $p \in R^V$ denotes the vector of potentials and i_{ext} the vectors of currents entering and leaving vertices, then these satisfy the relation:

$$Lp = i_{ext} \tag{6}$$

We exploit this formula to compute the effective resistance between pairs of vertices. The effective resistance between vertices u and v is the difference in potential one must impose between u and v to flow one unit of current from u to v. To measure, we compute the vector p for which the above equation holds true, where:

$$i_{ext}(x) = \begin{cases} 1 & \text{for } x = u \\ -1 & \text{for } x = v \text{ [3]} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

4.4. Harmonic function and laplacian

In this section, we introduce harmonic functions and the mathematical model for the heat equation.

We define the linear operator Q on functions by the following:

$$QF(x) = \tfrac{1}{2d}, \textstyle\sum_{y \in Z^d, |x-y|=1} F(y),$$

where the expression $|x - y| = 1$ shows that the vertices are one edge away, and the linear operator L can be the following:

$$LF(x) = (Q - I)\tfrac{1}{2d}, \textstyle\sum_{y \in Z^d, |x-y|=1} F(y) - F(x)$$

As a fact, we often regard this linear operator L to be the laplacian. Hence, if we let S_n be a sample random walk in the space Z^d, the we can write the function:

$$LF(x) = E[F(S_1) - F(S_0)|S_0 = x]$$

where $\mathbb{E}(P)$ denotes the expectation of a certain event.

Definition 19. We say that F is harmonic at x if $F(x)$ = $\frac{1}{deg(x)} \sum_y F(y)$

Theorem 20. *F is harmonic at x if and only if $LF(x) = 0$; this is an example of a mean-value property.*

Proof. First, we discuss in the context of an unweighted graph. Where let's consider a function F defined on the vertices of the graph. The Laplacian operator $LF(x)$ could be interpreted more simply as:

$$LF(x) = F(x) - \frac{1}{deg(x)} \sum_y F(y), \tag{8}$$

where $F(x)$ represents the value of the function F and vertex x, and $deg(x)$ denotes the degree of vertex x (which is the number of neighbors that x has), and the summation $\sum_y F(y)$ is taken over all neighboring vertices y of x.

To continue this demonstration, we need to examine the specific case where $LF(x) = 0$. To do this, we simply let $LF(x)$ in the equation above to be equal to 0:

$$0 = F(x) - \frac{1}{deg(x)} \sum_y F(y) \tag{9}$$

Rearranging the equation, we yield:

$$F(x) = \tfrac{1}{deg(x)} \sum_y F(y)$$

This equation implies that the value of the function F at vertex F is equal to the average value of F over its neighboring vertices, as our expression $\frac{1}{deg(x)} \sum_y F(y)$ denotes the average. Recalling our definition of a harmonic graph in the realms of graph theory, a function is said to be harmonic at a vertex x if it satisfies the above condition. Therefore, we have shown that, when $LF(x) = 0$, it signifies that the function F is harmonic at the point.

[8]

4.5. Generalization to electrical network

Now, we will examine this situation for weighted graphs. In the context of any weighted graphs, the Laplacian operator $LF(x)$ can be defined to be the following:

$$LF(x) = F(x) - \frac{\sum_y w_{xy} F(y)}{\sum_y w_{xy}}, \tag{10}$$

where, in the above definition:

(i) $F(x)$ represents the value of the function F at vertex x

(ii) w_{xy} denotes the weight of an edge that connects vertices x and y

(iii) The summation $\sum_y$ is taken over all neighboring vertices y of all x,(i.e., all vertices y connected to x by an edge)

In this weighted graph setting, $LF(x)$ calculates the difference between the value of the function F at vertex x and the weighted average of F over all of its neighboring vertices. The weights determine the significance or influence of each neighboring vertex y.

The condition $LF(x)$ indicates that the value of F at vertex x is equal to the weighted average of F. which characterizes the harmonic behavior of the F at that vertex.

5. Finding effective resistance

From the last section, finding effective resistance boils down to solving a linear equation involving Laplacian L. Therefore, given any graph, we are able to use the algorithm provided in section 3. The process of doing so as follows:

(i) First we construct the graph for the resistor network.

(ii) Then, we evaluate the individual resistance, which are the edges connecting the nodes. Indicate a starting node and an ending note, mark them as node 1 and node N, where N is the number of nodes in the network.

(iii) Then, we can construct the laplacian matrix for this set of resistor network. For each entry L_{ij} for $i = j$ and $i, j, \leq N$, that value is the sum of all resistance at node i. And for each entry L_{ij} for $i \neq j$, L_{ij} is equal to the resistance of the edge connecting the two nodes i and j.

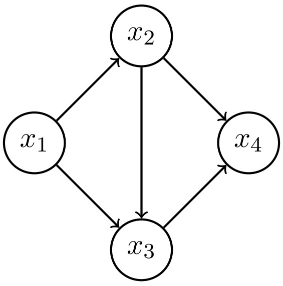
(iv) And it is obvious that $i_{ext} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ -1 \end{bmatrix}$ where there are N rows, depending on the number of nodes the electrical network.

(v) Now we set up the following linear equation of matrices to calculate the effective resistance: $\begin{bmatrix} L_{11} & L_{12} & \cdots & L_{1j} \\ L_{21} & L_{22} & \cdots & L_{2j} \\ \vdots & \vdots & & \vdots \\ L_{i1} & L_{i2} & \cdots & L_{ij} \end{bmatrix} \cdot \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_N \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ -1 \end{bmatrix}$, where we aim to find this matrix $\begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_N \end{bmatrix}$ using our algorithms.

(vi) After we determine that matrix, subtracting P_N from P_1 will yield our answer for the effective resistance.

Now, we will provide two examples of this.

First, we will have the following graph:



Where the edge connecting x_1 and x_2 is set to be $\frac{1}{2}$, the edge connecting x_1 and x_3 is set to be 1. the edge connecting x_2 and x_3 is set to be 1. The edge connecting x_2 and x_4 is set to be 1, and the edge connecting x_3 and x_4 is $\frac{1}{2}$. Given this construction, we first set up the Laplacian matrix L to be the following:

$$\begin{bmatrix} 1 & -\frac{1}{2} & -1 & 0 \\ -\frac{1}{2} & 2 & -1 & -1 \\ -1 & -1 & \frac{5}{2} & -\frac{1}{2} \\ 0 & -1 & -\frac{1}{2} & \frac{3}{2} \end{bmatrix}$$

And denote the matrix that we aim to find to be the following:

$$\begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix}$$

And obviously we know that $i_{ext} =$

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ -1 \end{bmatrix}$$

So now we can set up the linear system to solve for our desired matrix, which is:

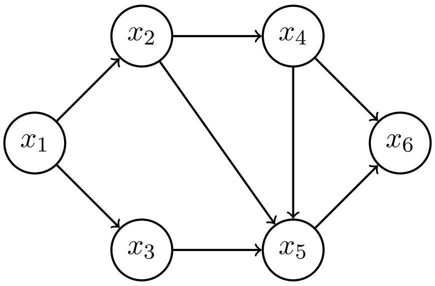
$$\begin{bmatrix} 1 & -\frac{1}{2} & -1 & 0 \\ -\frac{1}{2} & 2 & -1 & -1 \\ -1 & -1 & \frac{5}{2} & -\frac{1}{2} \\ 0 & -1 & -\frac{1}{2} & \frac{3}{2} \end{bmatrix} \cdot \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ -1 \end{bmatrix}$$

After solving this linear system usig our algorithm, we can find that:

$$\begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix} = \begin{bmatrix} \frac{12}{23} \\ -\frac{12}{23} \\ -\frac{5}{23} \\ \frac{25}{23} \end{bmatrix}$$

From this final matrix that we solved, we find that our effective resistance is $\frac{12}{23} - (-\frac{25}{23}) = \frac{37}{23}$.

Now, we will provide a slightly more complicated model to demonstrate the same process and result.



Where the edge connecting x_1 and x_2 is set to be 1, the edge connecting x_1 and x_3 is set to be 1. the edge connecting x_2 and x_4 is set to be $\frac{1}{2}$. The edge connecting x_2 and x_5 is set to be $\frac{1}{2}$, the edge connecting x_3 and x_5 is set to be $\frac{1}{2}$ and the edge connecting x_4 and x_5 is 2. Finally, the edges connecting x_4 and x_5 to x_6 are both set to be 1.

From this, we are once again able to set up our Laplacian matrix to be the following:

$$\text{L} = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 & 0 \\ -1 & 2 & 0 & -\frac{1}{2} & -\frac{1}{2} & 0 \\ -1 & 0 & \frac{3}{2} & 0 & -\frac{1}{2} & 0 \\ 0 & -\frac{1}{2} & 0 & 3 & -2 & -1 \\ 0 & -\frac{1}{2} & -\frac{1}{2} & -2 & 4 & -1 \\ 0 & 0 & 0 & -1 & -1 & 2 \end{bmatrix}$$

Following the same process, we can now set up our linear system:

$$\begin{bmatrix} 2 & -1 & -1 & 0 & 0 & 0 \\ -1 & 2 & 0 & -\frac{1}{2} & -\frac{1}{2} & 0 \\ -1 & 0 & \frac{3}{2} & 0 & -\frac{1}{2} & 0 \\ 0 & -\frac{1}{2} & 0 & 3 & -2 & -1 \\ 0 & -\frac{1}{2} & -\frac{1}{2} & -2 & 4 & -1 \\ 0 & 0 & 0 & -1 & -1 & 2 \end{bmatrix} \cdot \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \end{bmatrix}$$

Solving this linear system, we get that our desired matrix is

$$\begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \end{bmatrix} = \begin{bmatrix} -\frac{17}{100} \\ -\frac{1}{2} \\ -\frac{17}{50} \\ -\frac{49}{50} \\ -\frac{17}{25} \\ -\frac{133}{100} \end{bmatrix}$$

Again, from this final matrix that we determined, we can find by subtracting our P_6 from P_1, our effective resistance is $-\frac{17}{100} - (-\frac{133}{100}) = \frac{116}{100} = \frac{59}{50}$

6. Conclusion and discussions

In conclusion, this article introduced an important problem in the area of electrical networks (modeled as weight graphs). Then a novel method was used to connect this graph-theoretic problem with linear algebra through the bridge of harmonic functions. This linear algebra problem was then tackled successfully using effective algorithms such as LU decomposition. It exemplifies the power of abstract mathematics combined with the efficiency of computer science.

Although our method proves to be effective and universal in dealing with all electrical networks, most networks encountered in industry are actually ‘sparse’. This means even though the number of vertices might be huge, each vertex is connected to only a small number of edges. Therefore, a less universal approach specializing on sparse networks would further improve the efficiency and speed of the computation. This would be great for a further topic of research.

Acknowledgement

I would like to thank Dr. Bowen Yang from California Institutes of Technology for assisting me throughout the process of writing this paper. I would also like to thank my parents for supporting my research and work.

References

[1] Spielman. *The Laplacian,* Yale, 2009.
[2] Kothari. *Spectral Graph Theory,* Carnegie Mellon, 2015.
[3] Spielman. *Algorithms, Graph Theory, and Linear Equa- tions in Laplacian Matrices,* Proceedings of the International Congress of Mathematicians, 2010.
[4] Merris. *Laplacian Matrices of Graphs: A Survey,* California State University, 1994.
[5] Lambers. *The LU Decomposition,* Graduate Texts in Mathematics 201, Springer, 2000.
[6] Scott and Tuma. *Algorithms for Sparse Linear System,* Birkhauser, 2022.
[7] Jiang. *AN INTRODUCTION TO SPECTRAL GRAPH THEORY,* 2019.
[8] Lawler. *Random Walk and the Heat Equation,* STUDENT MATHEMATICAL LIBRARY Volume 55, 2010.
[9] Nicholson. *Linear Algebra with Applications,* Emory, 2014.