LAPLACIAN LINEAR SOLVERS

Prathik Diwakar and
Sundararajan Srinivasan

TOPICS

PREREQUISITES

BASIC LINEAR ALGEBRA

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

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

Lemma 1.3: Let $\lambda_1 \leq \cdots \leq \lambda_n$ be the spectrum of A with the $\mathbf{u}_1, \ldots \mathbf{u}_n$ being the corresponding eigenvectors. Then $A = \sum_{i=1}^{n} \lambda_i \mathbf{u}_i \mathbf{u}_i^T$. Here $\mathbf{u}_1, \ldots \mathbf{u}_n$ are orthonormal.

Theorem 1.4: If A is a $n \times n$ real symmetric matrix, then for all $1 \leq k \leq n$,

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

And

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

THE GRAPH LAPLACIAN

We define the adjacency matrix A and the degree matrix D of a graph $G = (V, E)$ as follows:

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

$$D_{i,j} = \begin{cases} d_i & \text{if i = j} \\ 0 & \text{other}wise \end{cases}$$

The Graph Laplacian of G is then defined as $L = D - A$.

We define the $n \times n$ matrices L_e, where $e = ij \in E$ as follows: $L(i,i) = L(j,j) = 1, L(i,j) = L(j,i) = -1$.

PROPERTIES OF THE LAPLACIAN

Lemma 2.1: If L is the graph Laplacian of $G = (V, E)$, $L = \sum_{e \in E} L_e$.

Lemma 2.2: Let L be a graph Laplacian. Then L is positive semidefinite.

Lemma 2.3: Let L be a graph Laplacian. Then $\mathbf{1} = (1, \dots, 1)$ is an eigenvector of L with eigenvalue 0.

Theorem 2.4: Let L be the graph Laplacian of $G = (V, E)$. Then, $\lambda_2 > 0$ iff G is connected.

Theorem 2.5: Let $B \in \{-1,0,1\}^{m \times n}$ be any arbitrary incidence matrix of G. Then $B^T B = L$.

PROOF OF THEOREM 2.5

We have $(B^T B)_{i,j} = \sum_e B_{e,i} B_{e,j}$.

When $i = j$, the only nonzero terms are for those edges which are incident to i, and the product is 1. Hence, the sum is the degree of the vertex.

For other entries, the term is only nonzero when e is shared by i and j. We can see in that case, $B_{e,i} B_{e,j} = -1$. Therefore, $(B^T B)_{i,j} = -1, \forall i \neq j, ij \in E$.

WHAT ARE WE TRYING TO PROVE?

Theorem 3: There exists an algorithm which takes as input a graph Laplacian L, a vector $\mathbf{b}$, and an error value $\varepsilon > 0$ and returns $\mathbf{x}$ such that:

$$\|\mathbf{x} - L^{+}\mathbf{b}\|_L \leq \varepsilon \|L^{+}\mathbf{b}\|_L$$

Where $\|\mathbf{b}\|_L = \sqrt{\mathbf{b}^T L \mathbf{b}}$. This algorithm runs in $\tilde{O}\left(m \log \frac{1}{\varepsilon}\right)$ time, where m is the number of non-zero entries in L. Here L^{+} is the Moore-Penrose pseudoinverse.

GRAPHS AS ELECTRICAL NETWORKS

DEFINING THE CIRCUIT

We can associate every edge to be a resistor with a resistance of 1.

Consider a voltage vector $\mathbf{v} \in \mathbb{R}^n$, which represents the voltage at each vertex, a current vector $\mathbf{i} \in \mathbb{R}^m$, which represents the current through each edge, and $\mathbf{c} \in \mathbb{R}^n$ representing the external current supplied to each vertex. Note that $\langle \mathbf{c}, \mathbf{1} \rangle = 0$, as no charge can accumulate in the network.

From Kirchoff's current law, we have:

$$B^T \mathbf{i} = \mathbf{c}$$

From Ohm's Law, we have:

$$\mathbf{i} = B\mathbf{v}$$

This gives us:

$$B^T B\mathbf{v} = L\mathbf{v} = \mathbf{c} \Rightarrow \mathbf{v} = L^+\mathbf{c}$$

GRAPHS AS ELECTRICAL NETWORKS

Therefore, the current through an edge ij is $(\mathbf{e}_i - \mathbf{e}_j)L^+\mathbf{c}$, as the resistance is 1.

We use this to define the effective resistance of $\mathrm{e} = ij$:

$$R_{eff}(e) = (\mathbf{e}_i - \mathbf{e}_j)^T L^+ (\mathbf{e}_i - \mathbf{e}_j)$$

We can now consider the currents through any edge, when a unit current is passed through another edge. This would be given by (in matrix notation)

$$\Pi = BL^+B^T$$

PROPERTIES OF Π

Lemma 4.1: Π is symmetric.

- This is because L^+ is symmetric.

Lemma 4.2: $\Pi^2 = \Pi$

- Proof: $\Pi^2 = BL^+ \overbrace{B^TB}^{=L} L^+B^T = B \overbrace{L^+LL^+}^{=L^+} B^T = BL^+B^T = \Pi$

Lemma 4.3: The eigenvalues of Π are either 1 or 0.

- Proof: Let v be an eigenvector. We have $\lambda\mathbf{v} = \Pi\mathbf{v} = \Pi^2\mathbf{v} = \lambda^2\mathbf{v}$, which implies $\lambda = 0{,}1$.

Lemma 4.4: If G is connected, $\mathrm{rank}(\Pi) = \mathrm{n} - 1$

- Proof: B is full column rank and the rank of L^+ is $n - 1$ if G is connected. Therefore, the rank of Π is $n - 1$.

A FINAL NOTE ON EFFECTIVE RESISTANCES

Theorem 4.5[1]**:** Let T be a spanning tree chosen uniformly at random from all the spanning trees in G. Then the probability that an edge e belongs to the tree is:

$$\mathbb{P}[e \in T] = R_{eff}(e) = \Pi(e, e)$$

[1]: C. D. GODSIL AND G. ROYLE, ALGEBRAIC GRAPH THEORY. SPRINGER, 2001.

WEIGHTED GRAPHS

Let $W \in \mathbb{R}^{m \times m}$ such that $W(e,e) = w(e)$, where w is the weight vector.

We define:

$$L = B^T W B$$

$$\Pi = W^{\frac{1}{2}} B L^{+} B^T W^{\frac{1}{2}}$$

If we consider the resistance of an edge to be the inverse of its weight, then all the results proved before apply, with $\mathbf{i} = WB\mathbf{v}$, as according to Ohm's law. The definition of R_{eff} remains the same, but it used the new definition of the Laplacian.

GRAPH SPARSIFICATION

Sparsification via Effective Resistances

INTRODUCTION

As already introduced, graphs are connected to electrical networks.

This connection can be used to spectrally sparsify graphs.

The goal of cut sparsification is, for a given graph $G = (V, E)$ and parameter ε, to find a weighted graph $H = (V, E')$ such that for any cut $(S, \bar{S})$ of V, the weight of the edges in H that cross the cut is within a multiplicative factor $1 \pm \varepsilon$ of the number of edges in G that cross this cut, while keeping the number of edges in H small.

This must also be done quickly for various applications.

SPECTRAL SPARSIFICATION

Spectral sparsification is a stronger notion than cut sparsification, and plays an important role in the construction of Laplacian solvers.

Definition 5.1: Given an undirected graph $G = (V, E)$ and a parameter $\varepsilon > 0$, a weighted graph $H = (V, E')$ is said to be an ε-spectral sparsifier of G if

$$\frac{1}{(1+\varepsilon)} \leq \frac{\mathbf{x}^{\mathrm{T}} L_H \mathbf{x}}{\mathbf{x}^{\mathrm{T}} L_G \mathbf{x}} \leq (1+\varepsilon), \forall \mathbf{x} \in \mathbb{R}^n$$

Where L_G and L_H are the graph Laplacians for G and H.

SPECTRAL SPARSIFICATION (CONTD.)

The goal is then to minimize the number of edges in H, while constructing it as quickly as possible. In particular, we want to construct a spectral sparsifier with $\tilde{O}(n/poly(\varepsilon))$ edges in $\tilde{O}(m)$ time.

Note that an ε-spectral sparsifier is also a ε-cut sparsifier as for any cut, we can plug in $\mathbf{1}_S$, the indicator for the cut, into the equation.

We shall soon prove the following theorem:

Theorem 5.1: There exists a randomized algorithm that, given a graph $G = (V, E)$ and a parameter $\varepsilon > 0$, constructs a spectral sparsifier H of size $\mathrm{O}(n \log n / \varepsilon^2)$ (edges) with probability $1 - 1/n$

USING EFFECTIVE RESISTANCES

The algorithm we use is an edge sampling algorithm – we repeatedly sample (with replacement) edges from the graph G according to a carefully chosen probability distribution, and then weight these sampled edges proportionally to the inverse of their probability of their being selected.

Formally, let p_e be the probability that edge e is selected, and let Y be the random variable such that $\mathbb{P}[Y = e] = p_e$. Let T be the number of samples. Finally let $Y_1, Y_2, \dots, Y_T$ be i.i.d. copies of Y. Then the weighted multiset of edges is:

$$\left\{\left(Y_1, \frac{1}{T \cdot p_{Y_1}}\right), \left(Y_2, \frac{1}{T \cdot p_{Y_2}}\right), \dots, \left(Y_T, \frac{1}{T \cdot p_{Y_T}}\right)\right\}$$

PROPERTIES OF ITS LAPLACIAN

Let B be some incidence matrix for G and $\mathbf{b}_e$ be the column vector corresponding to edge e in B^{T}. Then $L_G = B^{\mathrm{T}}B$. Now define $\mathbf{u}_e \overset{\text{def}}{=} \mathbf{b}_e/\sqrt{p_e}$. Then, by the definition, we have:

$$L_H = \sum_{i=1}^{T} \frac{\mathbf{b}_{Y_i}\mathbf{b}_{Y_i}^{\mathrm{T}}}{T \cdot p_{Y_i}} = \frac{1}{T}\sum_{i=1}^{T} \mathbf{u}_{Y_i}\mathbf{u}_{Y_i}^{\mathrm{T}}$$

Now note that:

$$\mathbb{E}\left[\mathbf{u}_Y\mathbf{u}_Y^{\mathrm{T}}\right] = \sum_{e\in E} \mathbb{P}[Y = e] \cdot \mathbf{u}_e\mathbf{u}_e^{\mathrm{T}} = L_G$$

From this, it is obvious that $\mathbb{E}[L_H] = L_G$

CHOOSING THE PROBABILITY DISTRIBUTION

We have to now specify the probability distribution. To do this, we use the intuition from **Theorem 4.5** where we found that effective resistances are related to the probability of an edge being present in a randomly chosen minimum spanning tree.

We let $p_e \overset{\text{def}}{=} \frac{R_e}{n-1}$, where $R_e = R_{eff}(e)$. The $n-1$ normalization factor is present as $\sum_e R_e = n - 1$.

While this intuition behind choosing this distribution is not very clear, the idea is to choose edges proportional to R_e as picking a few random minimum spanning trees for G seems like a good strategy to help in building a spectral sparsifier for it.

PROOF OF **THEOREM 5.1**

Recall the matrix $\Pi \overset{\text{def}}{=} BL_G^+B^\mathrm{T}$, which satisfies:

- $\Pi^2 = \Pi$
- If Π_e is the column of Π corresponding to edge e, then $R_e = \|\Pi_e\|^2$
- $\sum_e R_e = n - 1$
- Π is unitarily equivalent to $\sum_{j=1}^{n-1} \mathbf{e}_j\mathbf{e}_j^\mathrm{T}$, where $\mathbf{e}_j$ is the j^{th} standard basis vector for $\mathbb{R}^m$. This is because Π is symmetric (thus normal), and has only two eigenvalues: 1 with multiplicity $n-1$ and 0 with multiplicity $m-n+1$, same as $\sum_{j=1}^{n-1} \mathbf{e}_j\mathbf{e}_j^\mathrm{T}$.

Next we shall state an important theorem (without proof).

MATRIX CHERNOFF BOUND

Theorem 5.2[1]**:** Let $\varepsilon > 0$ be a small constant. Let $M^{d\times d}$ be a random, symmetric PSD matrix such that $\mathbb{E}[M] = I_d$, where I_d is the d-dimensional identity matrix. Let $\rho = \sup_M \|M\|$. Let T be a non-negative integer and let $M_1, M_2, \ldots, M_T$ be i.i.d. copies of M. Then,

$$\mathbb{P}\left[\left\|\frac{1}{T}\sum_{i=1}^{T} M_i - \mathbb{E}[M]\right\| > \varepsilon\right] \leq 2d \cdot \exp\left(-\frac{T\varepsilon^2}{2\rho}\right)$$

This theorem also holds under various other conditions, the one we are interested in being when $\mathbb{E}[M]$ is unitarily equivalent to $\sum_{j=1}^{d'} \mathbf{e}_j \mathbf{e}_j^{\mathrm{T}}$, for some $0 < d' \leq d$, in which case d' replaces d in the bound.

[1] R. AHLSWEDE AND A. WINTER, "STRONG CONVERSE FOR IDENTIFICATION VIA QUANTUM CHANNELS," IN IEEE TRANSACTIONS ON INFORMATION THEORY, MARCH 2002

CONDITIONS FOR USING **THEOREM 5.2**

To use **Theorem 5.2**, we define $\mathbf{v}_e \overset{\text{def}}{=} \Pi_e/\sqrt{p_e}$, $M \overset{\text{def}}{=} \mathbf{v}_Y \mathbf{v}_Y^{\mathrm{T}}$ and $M_i \overset{\text{def}}{=} \mathbf{v}_{Y_i} \mathbf{v}_{Y_i}^{\mathrm{T}}$ for $i = 1,2, \dots, T$. Now note that:

$$\mathbb{E}[M] = \mathbb{E}[\mathbf{v}_e \mathbf{v}_e^{\mathrm{T}}] = \sum_{e \in E} \Pi_e \Pi_e^{\mathrm{T}} = \Pi$$

Thus $\mathbb{E}[M]$ is unitarily equivalent to $\sum_{j=1}^{n-1} \mathbf{e}_j \mathbf{e}_j^{\mathrm{T}}$. Also note that:

$$\|\mathbf{v}_e\|^2 = \frac{\|\Pi_e\|^2}{p_e} = \frac{R_e}{p_e} = n - 1$$

From this we have $\|M\| \leq n-1$, so we can apply **Theorem 5.2**

USING **THEOREM 5.2**

Define $\widetilde{\Pi} \stackrel{\text{def}}{=} \frac{1}{T}\sum_{i=1}^{T} M_i$

We now use **Theorem 5.2** to get:

$$\mathbb{P}\left[\left\|\widetilde{\Pi} - \Pi\right\| > \varepsilon\right] \leq 2(n-1) \cdot \exp\left(-\frac{T\varepsilon^2}{2(n-1)}\right)$$

By setting $T = O(n \log n / \varepsilon^2)$ we can ensure that this probability of failure is $n^{-\Omega(1)}$.
Since $\Pi = BL_G^+B^\mathrm{T}$, we have that, for any edge e:

$$\mathbf{v}_e = \frac{\Pi_e}{\sqrt{p_e}} = \frac{BL_G^+\mathbf{b}_e}{\sqrt{p_e}} = BL_G^+\mathbf{u}_e$$

PROOF OF **THEOREM 5.1** (CONTD.)

Thus, we have:

$$\widetilde{\Pi} = \frac{1}{T}\sum_{i=1}^{T} \mathbf{v}_{Y_i}\mathbf{v}_{Y_i}^{\mathrm{T}} = \frac{1}{T}\sum_{i=1}^{T} BL_G^{+}\mathbf{u}_{Y_i}\mathbf{u}_{Y_i}^{\mathrm{T}}L_G^{+}B^{\mathrm{T}} = BL_G^{+}L_H L_G^{+}B^{\mathrm{T}}$$

and,

$$\Pi = BL_G^{+}B^{\mathrm{T}} = BL_G^{+}L_G L_G^{+}B^{\mathrm{T}}$$

Thus,

$$\left\|\widetilde{\Pi} - \Pi\right\| = \sup_{\mathbf{x}\neq\mathbf{0}} \left|\frac{\mathbf{x}^{\mathrm{T}}\left(\widetilde{\Pi} - \Pi\right)\mathbf{x}}{\mathbf{x}^{\mathrm{T}}\mathbf{x}}\right| = \sup_{\mathbf{x}\neq\mathbf{0}} \left|\frac{\mathbf{x}^{\mathrm{T}}BL_G^{+}(L_H - L_G)L_G^{+}B^{\mathrm{T}}\mathbf{x}}{\mathbf{x}^{\mathrm{T}}\mathbf{x}}\right|$$

PROOF OF **THEOREM 5.1** (CONTD.)

Now note that as G is connected, for any $\mathbf{z}$, if $B\mathbf{z} = \mathbf{0}$, then $\mathbf{z}$ is parallel to $\mathbf{1}$. Thus is we consider only $\mathbf{z} \neq \mathbf{0}$ such that $\langle \mathbf{z}, \mathbf{1} \rangle = 0$, then $B\mathbf{z} \neq \mathbf{0}$.

So we substitute $\mathbf{x} = B\mathbf{z}$ in the equation to get:

$$\|\widetilde{\Pi} - \Pi\| \geq \sup_{\mathbf{z}\neq\mathbf{0}, \langle \mathbf{z}, \mathbf{1} \rangle = 0} \left| \frac{\mathbf{z}^{\mathrm{T}} B^{\mathrm{T}} B L_G^{+} (L_H - L_G) L_G^{+} B^{\mathrm{T}} B \mathbf{z}}{\mathbf{z}^{\mathrm{T}} B^{\mathrm{T}} B \mathbf{z}} \right|$$

$$\|\widetilde{\Pi} - \Pi\| \geq \sup_{\mathbf{z}\neq\mathbf{0}, \langle \mathbf{z}, \mathbf{1} \rangle = 0} \left| \frac{\mathbf{z}^{\mathrm{T}} L_G L_G^{+} (L_H - L_G) L_G^{+} L_G \mathbf{z}}{\mathbf{z}^{\mathrm{T}} L_G \mathbf{z}} \right|$$

$$\|\widetilde{\Pi} - \Pi\| \geq \sup_{\mathbf{z}\neq\mathbf{0}, \langle \mathbf{z}, \mathbf{1} \rangle = 0} \left| \frac{\mathbf{z}^{\mathrm{T}} L_H \mathbf{z}}{\mathbf{z}^{\mathrm{T}} L_G \mathbf{z}} - 1 \right|$$

PROOF OF **THEOREM 5.1** (CONTD.)

Thus we have

$$\mathbb{P}\left[\sup_{\mathbf{z}\neq\mathbf{0},\langle\mathbf{z},\mathbf{1}\rangle=\mathbf{0}}\left|\frac{\mathbf{z}^{\mathrm{T}}L_H\mathbf{z}}{\mathbf{z}^{\mathrm{T}}L_G\mathbf{z}}-1\right|>\varepsilon\right]\leq\mathbb{P}\left[\left\|\widetilde{\Pi}-\Pi\right\|>\varepsilon\right]=n^{-\Omega(1)}$$

This completes the proof of **Theorem 5.1**.

The theorem can be extended to include a running time bound of $\tilde{O}(m\log 1/\varepsilon)$, however we shall not be proving that here.

CRUDE SPECTRAL SPARSIFICATION

Instead of requiring knowledge of R_e, we can work with the knowledge of $q_e \geq R_e$ for all e. This is a crude spectral sparsifier, and can be easily shown to work with $W \stackrel{\text{def}}{=} \sum_e q_e$ random samples. Thus we have a new theorem:

Theorem 5.3: Consider a graph $G = (V, E)$ with edge weights w_G, $\gamma > 0$, and numbers $q_e \geq w_G(e)R_e$ for all e. If $W \stackrel{\text{def}}{=} \sum_e q_e$, then the spectral sparsifier in **Theorem 5.1** upon taking $\mathrm{O}(W \log W \log 1/\gamma)$ samples from the probability distribution induced by the q_es produces a graph H that satisfies

$$G \preccurlyeq 2H \preccurlyeq 3G$$

With a probability of at least $1 - \gamma$.

CHOLESKY DECOMPOSITION

Solving Laplacians for Trees

INTRODUCTION

Cholesky Decomposition is a way to solve $A\mathbf{x} = \mathbf{b}$ where A is symmetric and PD. However, the same method still works for the Laplacian of a connected graph, as $\langle \mathbf{b}, \mathbf{1} \rangle = 0$, i.e., we are working in the subspace orthogonal to $\mathbf{1}$. Here, $L^+ \succ 0$.

Lemma 6.1: *Schur's Lemma*: $A = \begin{pmatrix} d_1 & \mathbf{u}_1^T \\ \mathbf{u}_1 & B_1 \end{pmatrix} \succ 0$ iff

$d_1 > 0$ and $B_1 - \frac{\mathbf{u}_1 \mathbf{u}_1^T}{d_1} \succ 0$.

PROOF OF SCHUR'S LEMMA

As $A \succ 0$, $d_1 > 0$ (consider $\mathbf{e}_1^T A \mathbf{e}_1 = d_1$). Now, consider minimising the quadratic expression $z^2 d + 2z\mathbf{u}_1^T\mathbf{y} + \mathbf{y}^T B_1 \mathbf{y}$ over z, for any (fixed) $\mathbf{y}$. The minima is at $z = -{}^{\mathbf{u}_1^T\mathbf{y}}/_{d_1}$. Thus, the minima is $\mathbf{y}^T \left(B_1 - {}^{\mathbf{u}_1\mathbf{u}_1^T}/_{d_1}\right)\mathbf{y}$. As this is true for all $\mathbf{y}$, $B_1 - {}^{\mathbf{u}_1\mathbf{u}_1^T}/_{d_1}$ must be PD.

The other direction is trivial.

CHOLESKY DECOMPOSITION

Theorem 6.2: *Cholesky Decomposition*: If $A \succ 0$ and symmetric, then there exists a lower triangular matrix Λ, such that $A = \Lambda\Lambda^T$.

Proof:

The theorem is trivially true for a 1×1 matrix. Now, say it was true for all $(n-1) \times (n-1)$ matrices.

Since $A \succ 0$, it is sufficient to express $A = \Lambda\Delta\Lambda^T$. This is because the positive definiteness implies that $\Delta_{ii} > 0$, which means we can write $A = \left(\Lambda\Delta^{1/2}\right)\left(\Lambda\Delta^{1/2}\right)^T$. We can now write A as

$$\begin{pmatrix} d_1 & \mathbf{u}_1^T \\ \mathbf{u}_1 & B_1 \end{pmatrix} = \begin{pmatrix} 1 & \mathbf{0}^T \\ \mathbf{u}_1/d_1 & I_{n-1} \end{pmatrix} \begin{pmatrix} d_1 & \mathbf{0}^T \\ \mathbf{0} & B_1 - \mathbf{u}_1\mathbf{u}_1^T/d_1 \end{pmatrix} \begin{pmatrix} 1 & \mathbf{u}_1^T/d_1 \\ \mathbf{0} & I_{n-1} \end{pmatrix}$$

PROOF OF CHOLESKY DECOMPOSITION

Let the matrix in the middle in the middle be A_1. Now let $B = B_1 - {\mathbf{u}_1\mathbf{u}_1^T}/{d_1}$. Now, by Schur's Lemma, $B \succ 0$.

Using the induction hypothesis, we have $B = \Lambda'\Delta'\Lambda'^T$, and

$$A_1 = \begin{pmatrix} 1 & \mathbf{0}^T \\ \mathbf{0} & \Lambda' \end{pmatrix} \begin{pmatrix} d_1 & \mathbf{0}^T \\ \mathbf{0} & \Delta' \end{pmatrix} \begin{pmatrix} 1 & \mathbf{0}^T \\ \mathbf{0} & \Lambda'^T \end{pmatrix} \stackrel{\text{def}}{=} \Lambda''\Delta\Lambda''^T$$

Thus, we have $A = \Lambda_1\Lambda''\Delta\Lambda''^T\Lambda_1^T$. Now, as the product of lower triangular matrices is lower triangular, we are done.

USING CHOLESKY DECOMPOSITION

Given a decomposition, we can solve $A\mathbf{x} = \mathbf{b}$ quickly.

We can evaluate $\mathbf{b}' = \Lambda^{+}\mathbf{b}$ and then $\mathbf{x} = \left(\Lambda^{\mathrm{T}}\right)^{+}\mathbf{b}'$. Due to the triangular nature of Λ, the time taken to calculate the pseudoinverse is of the order of the number of non-zero elements of Λ.

If we first permute the rows by a permutation matrix Q and then find the decomposition, we might find a decomposition with fewer non-zero elements(known as the *fill-in*). However, finding the minimum fill-in is NP-hard.

FAST SOLVERS FOR TREES

We try to find a fast solver for $L_T\mathbf{x} = \mathbf{b}$, where T is a tree.

We can associate every symmetric matrix A with a weighted graph (potentially with self loops), where $A(i, j)$ is the weight of the edge connecting ij.

- Note: This is not the Laplacian of the graph.

Theorem 6.3: Given a symmetric, PSD matrix A and a vector $\mathbf{b}$ such that the graph of A corresponds to a tree, one can find in $O(n)$ time a permutation matrix Q such that the Cholesky decomposition of $Q^T AQ$ has at most $O(n)$ nonzero entries.

PROOF OF THEOREM 6.3

We can view the proof of 6.2 as one that modifies the graph. When we recursively process row i, the resulting graph (corresponding to A_1) has the following changes:

a. All edges $ij, i \neq j$ are deleted. This corresponds to setting $A_1(i,j) = 0$.
b. For every pair jk neighbouring to i, a (potentially new) edge is modified. This corresponds to setting $A_1(j,k) = B_1(j,k) - \frac{\overbrace{A(i,j)A(i,k)}^{\mathbf{u}_1\mathbf{u}_1^T}}{\underbrace{A(i,i)}_{d_1}}$.

Suppose the graph corresponds to the system is a tree, potentially with self loops. Then in each iteration, we can choose a leaf node, by choosing an appropriate permutation matrix.

PROOF OF THEOREM 6.3

Since there is a single node adjacent to i, the graph associated with A_1 is a tree.

This implies, we can write $A = \Lambda\Delta\Lambda^T$ where $\Lambda = \Lambda_1\Lambda_2 \dots \Lambda_n$, where each Λ_i is lower triangular, and has at most one nonzero off-diagonal element.

This gives a Cholesky decomposition with at most $O(n)$ nonzero entries, where in each iteration $O(1)$ operations are done, which implies the process takes $O(n)$ time.

AN IMPORTANT COROLLARY

Corollary 6.4: If L_T is the Laplacian of a tree T and $\mathbf{b}$ is a vector such that $\langle \mathbf{b}, \mathbf{1} \rangle = 0$, then the solution of $L_T\mathbf{x} = \mathbf{b}$ can found in $O(n)$ time.

Proof: We can see the graph associated with the Laplacian of a tree is a tree. Hence, by 6.3, we can find the Cholesky Decomposition of the permuted Laplacian to get $\Lambda\Lambda^T Q^T \mathbf{x} = Q^T \mathbf{b}$ in $O(n)$ time.

This is possible despite Λ not being full rank (as L_T) is not full rank, as $\langle \mathbf{b}, \mathbf{1} \rangle = 0$, and thus $\mathbf{b}$ is in the column space of $L_T Q$.

Therefore, this solution can be calculated in the number of nonzero entries of L_T, i.e., in $O(n)$ itime

ITERATIVE LINEAR SOLVERS

The Gradient Method

THE OPTIMIZATION VIEW

We shall formulate solving the equation $A\mathbf{x} = \mathbf{b}$ as a convex optimization problem. For this, we assume that A is symmetric and positive-definite.

Solving $A\mathbf{x} = \mathbf{b}$ is equivalent to finding the minimum of $f(\mathbf{x})$ –

$$f(\boldsymbol{x}) \overset{\text{def}}{=} \frac{1}{2}\mathbf{x}^{\mathrm{T}}A\mathbf{x} - \mathbf{b}^{\mathrm{T}}\mathbf{x}$$

Since A is PD, $\nabla^2 f = A \succ 0$, so f is strictly convex and thus has a unique minimum $\mathbf{x}^*$. As we already know, this $\mathbf{x}^*$ must satisfy –

$$\nabla f(\mathbf{x}^*) = A\mathbf{x}^* - \mathbf{b} = \mathbf{0}$$

GRADIENT DESCENT-BASED SOLVER

Since f is convex, we can use the well known gradient descent algorithm to solve for $\mathbf{x}^*$. Typically in gradient descent, we start at $\mathbf{x}_0$ and iterative move from $\mathbf{x}_t$ to $\mathbf{x}_{t+1}$ by moving opposite the direction of the gradient of f, which can (here) be calculated with a single multiplication of a matrix and a vector, which takes time t_A (say).

Theorem 7.1: There is an algorithm GDSOLVE that, given an $n \times n$ matrix $A \succ 0$, a vector $\mathbf{b}$ and $\varepsilon > 0$, finds a vector $\mathbf{x}$ such that

$$\|\mathbf{x} - A^+\mathbf{b}\|_A \leq \varepsilon \|A^+\mathbf{b}\|_A$$

in time $O(t_A \cdot \kappa(A) \log 1/\varepsilon)$, where the condition number of A is defined as $\kappa(A) \overset{\text{def}}{=} \lambda_n(A)/\lambda_1(A)$. For a vector $\mathbf{v}$, $\|\mathbf{v}\|_A \overset{\text{def}}{=} \sqrt{\mathbf{v}^{\mathrm{T}} A \mathbf{v}}$.

THE STEP SIZE

First we define $\mathbf{d}_t \stackrel{\text{def}}{=} \mathbf{x}^* - \mathbf{x}_t$ and $\mathbf{r}_t \stackrel{\text{def}}{=} -\nabla f(\mathbf{x}_t) = \mathbf{b} - A\mathbf{x}_t = A\mathbf{d}_t$.

The step size η_t is a parameter which determines how much to move towards $\mathbf{r}_t$, and we have $\mathbf{x}_{t+1} \stackrel{\text{def}}{=} \mathbf{x}_t + \eta_t \mathbf{r}_t$.

We can choose η_t greedily, to minimize $f(\mathbf{x}_{t+1})$. Define g as:

$$g(\eta) \stackrel{\text{def}}{=} f(\mathbf{x}_t + \eta \mathbf{r}_t) = \frac{1}{2}(\mathbf{x}_t + \eta \mathbf{r}_t)^T A(\mathbf{x}_t + \eta \mathbf{r}_t) - \mathbf{b}^{\mathrm{T}}(\mathbf{x}_t - \eta \mathbf{r}_t)$$

It is easy to see that g attains its minimum at $\eta_t = \frac{\mathbf{r}_t^{\mathrm{T}} \mathbf{r}_t}{\mathbf{r}_t^{\mathrm{T}} A \mathbf{r}_t}$

ALGORITHM 7.1: GDSolve

Input: Symmetric, PD matrix $A \in \mathbb{R}^{n \times n}$, vector $\mathbf{b} \in \mathbb{R}^n$ and T.

Output: $\mathbf{x}_T \in \mathbb{R}^n$

```
1. $\mathbf{x}_0 \leftarrow \mathbf{0}$
2. $\mathbf{for}\ t = 0 \to T - 1\ \mathbf{do}$
    1. Set $\mathbf{r}_t = \mathbf{b} - A\mathbf{x}_t$
    2. Set $\eta_t = \frac{\mathbf{r}_t^{\mathrm{T}}\mathbf{r}_t}{\mathbf{r}_t^{\mathrm{T}} A\mathbf{r}_t}$
    3. Set $\mathbf{x}_{t+1} = \mathbf{x}_t + \eta_t \mathbf{r}_t$
3. $\mathbf{end\ for}$
4. $\mathbf{return}\ \mathbf{x}_T$
```

LEMMA 7.2: $\|\mathbf{d}_{t+1}\|_A^2 \leq \left(1 - \frac{1}{\kappa(A)}\right) \cdot \|\mathbf{d}_t\|_A^2$

Note two things: $\mathbf{d}_{t+1} = \mathbf{d}_t - \eta_t \mathbf{r}_t$ and $\mathbf{r}_{t+1} = \mathbf{r}_t - \eta_t A \mathbf{r}_t$. This gives $\mathbf{r}_t^{\mathrm{T}} \mathbf{r}_{t+1} = \mathbf{r}_t^{\mathrm{T}} \mathbf{r}_t - \frac{\mathbf{r}_t^{\mathrm{T}} \mathbf{r}_t}{\mathbf{r}_t^{\mathrm{T}} A \mathbf{r}_t} \mathbf{r}_t^{\mathrm{T}} A \mathbf{r}_t = 0$, so $\mathbf{r}_t$ and $\mathbf{r}_{t+1}$ are orthogonal.

Thus, $\|\mathbf{d}_{t+1}\|_A^2 = \mathbf{d}_{t+1}^{\mathrm{T}} A \mathbf{d}_{t+1} = (\mathbf{d}_t - \eta_t \mathbf{r}_t)^{\mathrm{T}} \mathbf{r}_{t+1} = \mathbf{d}_t^{\mathrm{T}} \mathbf{r}_{t+1}$ since $\mathbf{r}_t$ and $\mathbf{r}_{t+1}$ are orthogonal. Thus, $\|\mathbf{d}_{t+1}\|_A^2 = \mathbf{d}_t^{\mathrm{T}} \mathbf{r}_{t+1} = \mathbf{d}_t^{\mathrm{T}} A(\mathbf{d}_t - \eta_t \mathbf{r}_t)$

Now we factor out the $\|\mathbf{d}_t\|_A^2$ to get

$$\|\mathbf{d}_t\|_A^2 \cdot \left(1 - \eta_t \frac{\mathbf{d}_t^{\mathrm{T}} A \mathbf{r}_t}{\mathbf{d}_t^{\mathrm{T}} A \mathbf{d}_t}\right) = \|\mathbf{d}_t\|_A^2 \cdot \left(1 - \frac{\mathbf{r}_t^{\mathrm{T}} \mathbf{r}_t}{\mathbf{r}_t^{\mathrm{T}} A \mathbf{r}_t} \cdot \frac{\mathbf{d}_t^{\mathrm{T}} A^2 \mathbf{d}_t}{\mathbf{d}_t^{\mathrm{T}} A \mathbf{d}_t}\right)$$

PROOF OF **LEMMA 7.2** AND **THEOREM 7.1**

Now recall the min-max characterization of eigenvalues (**Theorem 1.5**). Thus, the first factor (in the product inside the brackets) is at most $1/\lambda_n(A)$ while the second factor is at least $\lambda_1(A)$ [since $A \succ 0$, $A^{1/2}\mathbf{d}_t \neq 0$ so we can shift one of the As around in the product]. Thus, $\|\mathbf{d}_{t+1}\|_A^2 \leq (1 - 1/\kappa(A))\|\mathbf{d}_t\|_A^2$

Simply take $T = 2\kappa(A) \log 1/\varepsilon$ to get $\|\mathbf{d}_T\|_A \leq \varepsilon\|\mathbf{d}_0\|_A$, which completes the proof of **Theorem 7.1**

ITERATIVE LINEAR SOLVERS

The Conjugate Gradient Method

THE KRYLOV SUBSPACE

Consider the same problem as before, i.e.,

$$\min_{\mathbf{x}} f(\mathbf{x}) \stackrel{\text{def}}{=} \frac{1}{2}\mathbf{x}^T A\mathbf{x} - \mathbf{b}^T\mathbf{x}$$

In the previous algorithm, we generated, which generated:

$$\mathbf{x}_1 = \mathbf{x}_0 + \eta_0 \mathbf{r}_0$$

$$\mathbf{x}_2 = \mathbf{x}_1 + \eta_1 \mathbf{r}_1 = \mathbf{x}_1 + \eta_1(\mathbf{r}_0 - \eta_0 A\mathbf{r}_0) = \mathbf{x}_0 + \eta_1 \mathbf{r}_0 - \eta_1\eta_0 A\mathbf{r}_0,$$

and so on, with $\mathbf{r}_0 = \mathbf{b}$ and $\mathbf{r}_i = \mathbf{b} - A\mathbf{x}_i$.

This implies $\mathbf{x}_t \in \mathbf{x}_0 + \mathcal{K}_t$, where $\mathcal{K}_t$ is the subspace spanned by $\left\{A^i\mathbf{b} : i \in \{0,1,\dots,n-1\}\right\}$. This is known as the Krylov Subspace of order t generated by A and $\mathbf{b}$.

IDEA BEHIND CONJUGATE GRADIENT

In the previous algorithm, the point we move to might not be the minimizer of f over the affine space $\mathbf{x}_0 + \mathcal{K}_t$. This is what we will do in this algorithm.

Theorem 8.1: There is an algorithm that, given an $n \times n$ symmetric matrix $A \succ 0$, a vector $\mathbf{b}$, and $\varepsilon > 0$, finds a vector $\mathbf{x}$ such that

$$\|\mathbf{x} - A^{+}\mathbf{b}\|_A \leq \varepsilon \|A^{+}\mathbf{b}\|_A$$

in time $O\left(t\sqrt{\kappa(A)} \log \frac{1}{\varepsilon}\right)$.

We shall find this algorithm and prove the theorem in the following section.

MOTIVATION FOR PROOF

We could find the minimizer of f over $\mathbf{x} + \mathcal{K}_t$ quickly, if we had a basis of $\mathcal{K}_t$, $\{\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_{t-1}\}$ such that:

$$f\left(\mathbf{x}_0 + \sum_{i=0}^{t-1} \beta_i \mathbf{p}_i\right) - f(\mathbf{x}_0) = \sum_{i=0}^{t-1} \big(f(\mathbf{x}_0 + \beta_i \mathbf{p}_i) - f(\mathbf{x}_0)\big)$$

We could find the minimiser over each β_i separately.

If f is linear, this is trivially true. Thus, we only consider the quadratic portion. For brevity, let $\mathbf{v}_i = \beta_i \mathbf{p}_i$. Then, evaluating the LHS:

$$\frac{1}{2}\left(\mathbf{x} + \sum_i \boldsymbol{v}_i\right)^T A \left(\mathbf{x} + \sum_i \mathbf{v}_i\right) - \frac{1}{2}\mathbf{x}^T A \mathbf{x} = \mathbf{x}^T A \sum_i \mathbf{v}_i + \frac{1}{2}\left(\sum_i \mathbf{v}_i\right)^T A \left(\sum_i \mathbf{v}_i\right)$$

The RHS evaluates to $\sum_i \left(\mathbf{x}^T A \mathbf{v}_i + \frac{1}{2}\mathbf{v}_i^T A \mathbf{v}_i\right)$. The two are equal iff the cross terms $\mathbf{v}_i^T A \mathbf{v}_j = 0$, whenever $i \neq j$.

A-ORTHOGONAL VECTORS

Definition 8.1: Given a symmetric matrix A, two vectors $\mathbf{x}, \mathbf{y}$ are A-orthogonal iff $\mathbf{x}^T A \mathbf{y} = 0$.

Thus, we can see that our choices for $\{\mathbf{p}_0, \dots, \mathbf{p}_{t-1}\}$ must be A-orthogonal.

With this orthogonal basis, we can calculate α_t to be the vectors that minimise $f(\mathbf{x}_0 + \alpha \mathbf{p}_t) - f(\mathbf{x}_0)$. Then, $\alpha_t = \frac{\mathbf{p}_t^T \mathbf{r}_0}{\mathbf{p}_t^T A \mathbf{r}_0}$.

COMPUTING THE A-ORTHONORMAL BASIS

Gram-Schmidt orthogonalization would take $O(t)$ matrix-vector computations to calculate $\mathbf{p}_{t+1}$.

We use the symmetry of A to reduce this to $O(1)$ computations.

We start with $\mathbf{p}_0 = \mathbf{r}_0$. Suppose that $\{\mathbf{p}_0, \dots, \mathbf{p}_i\}$ spans $\mathcal{K}_{i+1}$, and $A\mathbf{p}_i \in \mathcal{K}_{i+2}$, for some i. This is true for $i = 0$.

Consider $A\mathbf{p}_i$. If $A\mathbf{p}_i \in \mathcal{K}_{i+1}$, $\mathcal{K}_j = \mathcal{K}_{i+1}, \forall j \geq i+1$, and we would be done, as this would span the entire space. Now assume that $A\mathbf{p}_i \notin \mathcal{K}_{i+1}$. Now, we construct $\mathbf{p}_{i+1}$ as follows:

$$\mathbf{p}_{i+1} \stackrel{\text{def}}{=} A\mathbf{p}_i - \sum_{j \leq i} \frac{\left(A\mathbf{p}_j\right)^T A\mathbf{p}_j}{\mathbf{p}_j^T A\mathbf{p}_j} \mathbf{p}_j$$

COMPUTING THE A-ORTHONORMAL BASIS

This implies that

$$\mathcal{K}_{i+2} = \text{span}\{\mathbf{r}_0, A\mathbf{r}_0, \dots, A^{i+1}\mathbf{r}_0\} = \text{span}\{\mathbf{p}_0, \dots, \mathbf{p}_{i+1}\}$$

This completes the induction.

Now, this implies that $A\mathbf{p}_i$ can be written as a linear combination of $\mathbf{p}_j$, for $j \leq i+1$. Thus, for all $j \leq i$,

$$(A\mathbf{p}_i)^T(A\mathbf{p}_j) = \mathbf{p}_i^T A(A\mathbf{p}_j) = \sum_{k \leq j+1} c_j \mathbf{p}_i^T A\, \mathbf{p}_k$$

Thus, $\forall j < i-1, (A\mathbf{p}_i)^T(A\mathbf{p}_j) = 0$. Thus, we can write $\mathbf{p}_{t+1}$ as:

$$\mathbf{p}_{t+1} = A\mathbf{p}_t - \frac{\mathbf{p}_t^T A^2 \mathbf{p}_t}{\mathbf{p}_t^T A \mathbf{p}_t} \boldsymbol{p}_t - \frac{\boldsymbol{p}_t^T A^2 \boldsymbol{p}_{t-1}}{\boldsymbol{p}_{t-1}^T A \boldsymbol{p}_{t-1}} \boldsymbol{p}_{t-1}$$

THE COMPLETE CONJUGATE GRADIENT ALGORITHM

Algorithm 8.1: CGSolve

Input: Symmetric, PD matrix $A \in \mathbb{R}^{n\times n}$, $\mathbf{b} \in \mathbb{R}^n$, and T

Output: $\mathbf{x}_T \in \mathbb{R}^n$

```
1.  $\mathbf{x}_0 \leftarrow \mathbf{0}, \mathbf{r}_0 \leftarrow \mathbf{b}, \mathbf{p}_0 \leftarrow \mathbf{r}_0$
2.  for $t = 0 \to T - 1$ do:
    1.  $\alpha_t \leftarrow \frac{\mathbf{p}_t \mathbf{r}_0}{\mathbf{p}_t A \mathbf{p}_t}$
    2.  $\mathbf{r}_t \leftarrow \mathbf{b} - A\mathbf{x}_t$
    3.  $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t + \alpha_t \mathbf{p}_t$
    4.  $\mathbf{p}_{t+1} \leftarrow A\mathbf{p}_t - \frac{\mathbf{p}_t^T A^2 \mathbf{p}_t}{\mathbf{p}_t^T A \mathbf{p}_t}\mathbf{p}_t - \frac{\mathbf{p}_t^T A^2 \mathbf{p}_{t-1}}{\mathbf{p}_{t-1}^T A \mathbf{p}_{t-1}}\mathbf{p}_{t-1}$
3.  end for
4.  return $\mathbf{x}_T$
```

ANALYSIS OF THE ALGORITHM

We can see in n steps, $\mathbf{x}_n = \mathbf{x}^\star$, as $\mathbf{x}^\star \in \mathbf{x}_0 + \mathcal{K}_n$.

What if we want an ε-approximate solution?

Since $\mathbf{x}_t \in \mathbf{x}_0 + \mathcal{K}_t$, $\mathbf{x}_t = \mathbf{x}_0 + \sum_{i=0}^{t-1} \gamma_i A^i \mathbf{r}_0$. This motivates the definition of $p(x) = \sum_{i=0}^{t-1} \gamma_i x^i$. Thus, $\mathbf{x}_t = \mathbf{x}_0 + p(A)\mathbf{r}_0$.

Now, as $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0 = A(\mathbf{x}^\star - \mathbf{x}_0), \mathbf{x}_t = \mathbf{x}_0 + p(A)A(\mathbf{x}^\star - \mathbf{x}_0)$. Therefore,

$$\mathbf{x}_t - \mathbf{x}^\star = (I - p(A)A)(\mathbf{x}_0 - \mathbf{x}^\star) = q(A)(\mathbf{x}_0 - \mathbf{x}^\star)$$

Where $q(x) = 1 - xp(x)$.

Now, there is a one-to-one correspondence between points in $\mathbf{x}_0 + \mathcal{K}_n$ and degree $t - 1$ polynomials, which has a one-to-one correspondence to degree t polynomials that evaluate to 1 at 0. Let this set of polynomials be $\mathcal{Q}_t$.

ANALYSIS OF THE ALGORITHM

Since $\mathbf{x}_t$ minimises $\|\mathbf{x}_t - \mathbf{x}^\star\|_A^2$ over $\mathcal{K}_t$, we get:

$$f(\mathbf{x}_t) - f(\mathbf{x}^\star) = \frac{1}{2}\|\mathbf{x}_t - \mathbf{x}^\star\|_A^2 = \min_{q \in \mathcal{Q}_t} \big(q(A)(\mathbf{x}_t - \mathbf{x}^\star)\big)^T A \left(q(A)(\mathbf{x}_t - \mathbf{x}^\star)\right)$$

Lemma 8.2: Let A be a symmetric matrix with eigenvalues $\lambda_1, \dots, \lambda_n$. Then, for a polynomial $p(\cdot)$ and vector $\mathbf{v}$:

$$(p(A)\mathbf{v})^T A (p(A)\mathbf{v}) \le \mathbf{v}^T A \mathbf{v} \cdot \max_{i \in [n]} |p(\lambda_i)|^2$$

PROOF OF 8.2

We can write $A = U\Gamma U^T$, which is the eigendecomposition of A. Now $p(A) = Up(\Gamma)U^T$, giving us:

$$p(\Gamma)^T A p(\Gamma) = Up(\Gamma)\Gamma p(\Gamma)U^T = U\Gamma p^2(\Gamma)U^T$$

Now, we can write any vector $\mathbf{v} = \sum_i \zeta_i \mathbf{u}_i$. Therefore, $\mathbf{v}^T p(\Gamma)^T A p(\Gamma)\mathbf{v} = \sum_i \zeta_i^2 \lambda_i p^2(\lambda_i)$, and $\mathbf{v}^T A\mathbf{v} = \sum_i \zeta_i^2 \lambda_i$. The lemma follows trivially.

USING LEMMA 8.2

Using Lemma 8.2, we have:

$$f(\mathbf{x}_t) - f(\mathbf{x}^\star) \le \min_{q \in \mathcal{Q}_t} \max_{i \in [n]} |q(\lambda_i)|^2 f(\mathbf{x}_0) - f(\mathbf{x}^\star)$$

Now, as $\lambda_1 \le \cdots \le \lambda_n$,

$$f(\mathbf{x}_t) - f(\mathbf{x}^\star) \le \min_{q \in \mathcal{Q}_t} \max_{x \in [\lambda_1, \lambda_n]} |q(x)|^2 f(\mathbf{x}_0) - f(\mathbf{x}^\star)$$

As $f(\mathbf{x}^\star) = -\frac{1}{2}\|\mathbf{x}^\star\|_A^2$, and $f(\mathbf{0}) = 0$, we have proved:

Lemma 8.3: Let $A \succ 0$, λ_1, λ_n be the smallest and largest eigenvalues of A, and $\mathcal{Q}_t$ be the set of polynomials of degree at most t which take value 1 at 0. Then:

$$\|\mathbf{x}_t - \mathbf{x}^\star\|_A^2 \le \|\mathbf{x}^\star\|_A^2 \min_{q \in \mathcal{Q}_t} \max_{x \in [\lambda_1, \lambda_n]} |q(x)|^2$$

Thus, any polynomial $q \in \mathcal{Q}_t$ can be used to give an upper bound.

CHEBYSHEV POLYNOMIALS

We recursively define the Chebyshev polynomials (of the first kind) as follows:

$$T_0(x) \overset{\text{def}}{=} 1, T_1(x) \overset{\text{def}}{=} x$$
$$T_t(x) \overset{\text{def}}{=} 2xT_{t-1}(x) - T_{t-2}(x)$$

Lemma 8.5: $T_t(\cos\theta) = \cos t\theta$. Specifically, $T_t([-1,1]) \subseteq [-1,1]$.

- Proof: This is true for the base case. Now, $\cos(t+1)\,\theta - \cos(t-1)\,\theta = 2\cos\theta\cos t\theta$, and thus we are done.

Now, for $0 < a < b$, we define:

$$Q_{a,b,t}(x) \overset{\text{def}}{=} \frac{T_t\left(\frac{a+b-2x}{b-a}\right)}{T_t\left(\frac{a+b}{b-a}\right)}$$

PROOF OF 8.1

We can see that $Q_{a,b,t} \in \mathcal{Q}_t$, and for $x \in [a,b]$, the numerator is at most 1, by lemma 8.5. Now, taking $a = \lambda_1$, $b = \lambda_n$ we have, $\forall x \in [\lambda_1, \lambda_n]$

$$Q_{\lambda_1,\lambda_n,t}(x) \le T_t\left(\frac{\lambda_n + \lambda_1}{\lambda_n - \lambda_1}\right)^{-1} = T_t\left(\frac{\kappa(A)+1}{\kappa(A)-1}\right)^{-1}$$

One can show, using $\cosh$ that

$$T_t\left(\frac{\kappa(A)+1}{\kappa(A)-1}\right) = \frac{1}{2}\left(\left(\frac{\sqrt{\kappa(A)}+1}{\sqrt{\kappa(A)}-1}\right)^t + \left(\frac{\sqrt{\kappa(A)}-1}{\sqrt{\kappa(A)}+1}\right)^t\right)$$

This gives us:

$$Q_{\lambda_1,\lambda_n,t}(x) \le 2\left(\frac{\sqrt{\kappa(A)}-1}{\sqrt{\kappa(A)}+1}\right)^t$$

Thus, by Lemma 8.3, for any $t > \Omega\left(\sqrt{\kappa(A)}\log \varepsilon^{-1}\right)$, after t steps of CGSolve, we get:

$$f(\mathbf{x}_t) - f(\mathbf{x}^\star) \le \varepsilon^2\big(f(\mathbf{x}_0) - f(\mathbf{x}^\star)\big) \Rightarrow \|\mathbf{x} - A^+\mathbf{b}\|_A \le \varepsilon\|A^+\mathbf{b}\|_A$$

CHEBYSHEV ITERATION

In CGSolve, the output is not linear in the input, i.e., we want sequence of polynomials such that $\mathbf{x}_t = p_t(A)\mathbf{b}$.

As in the proof of 8.3, it is sufficient to define p_ts such that:

$$\max_{x\in[\lambda_1,\lambda_n]} |xp_t(x) - 1| \leq O\left(1 - \sqrt{\lambda_1/\lambda_2}\right)^t$$

Which gives us:

$$\|\mathbf{x}_t - A^+\mathbf{b}\|_A \leq O\left(1 - \sqrt{\lambda_1/\lambda_2}\right)^t \|A^+\mathbf{b}\|_A^2$$

We can set $p_t = Q_{\lambda_1,\lambda_n,t}$ to get this bound. Further, we can see that if $0 < \lambda_l \leq \lambda_1$ is used in place of λ_1 and $\lambda_u \geq \lambda_n$ is used in place of λ_n, we get a similar bound.

CHEBYSHEV ITERATION

We can calculate the polynomial $Q_{\lambda_1,\lambda_n,t}$ using recursion, using the definitions shown before.

The iteration proceeds as follows:

$$\mathbf{x}_0 = \mathbf{0}, \mathbf{x}_1 = \mathbf{b}$$
$$\mathbf{x}_t = \alpha_2 A\mathbf{x}_{t-1} + \alpha_1 \mathbf{x}_{t-2} + \alpha_0 \mathbf{b}$$

The values of α_i depend on λ_l and λ_u.

Theorem 8.6[1]: There is an algorithm, which takes a $n \times n$ symmetric PD matrix A, a vector $\mathbf{b}$, numbers $0 < \lambda_l \leq \lambda_1$ and $\lambda_u \geq \lambda_n$, and an error parameter $\varepsilon > 0$ and returns $\mathbf{x}$ such that:

a) $\|\mathbf{x} - A^+\mathbf{b}\|_A \leq \varepsilon \|A^+\mathbf{b}\|_A$

b) $\mathbf{x} = Z\mathbf{b}$, where Z only depends on A and ε.

c) $\|Z - A^+\| \leq \varepsilon$

This algorithm runs in $O\left(t_A\sqrt{\lambda_u/\lambda_l}\log\left(\varepsilon^{-1}\lambda_l^{-1}\right)\right)$ time.

[1] BARRETT, RICHARD; MICHAEL, BERRY; TONY, CHAN; DEMMEL, JAMES; DONATO, JUNE; DONGARRA, JACK; EIJKHOUT, VICTOR; POZO, ROLDAN; ROMINE, CHARLES; VAN DER VORST, HENK (1993). "TEMPLATES FOR THE SOLUTION OF LINEAR SYSTEMS: BUILDING BLOCKS FOR ITERATIVE METHODS"