

Weighted 1-Laplacian Solvers for Well-Shaped Simplicial Complexes

Ming Ding

ming.ding@inf.ethz.ch

Department of Computer Science
ETH Zurich

Peng Zhang

pz149@cs.rutgers.edu

Department of Computer Science
Rutgers University

Abstract

We present efficient algorithms for solving systems of linear equations in weighted 1-Laplacians of well-shaped simplicial complexes. 1-Laplacians or higher-dimensional Laplacians generalize graph Laplacians to higher-dimensional simplicial complexes and are crucial in computational topology and topological data analysis. Previously, nearly-linear time solvers were designed for *unweighted* simplicial complexes that triangulate a three-ball in \mathbb{R}^3 (Cohen, Fasy, Miller, Nayyeri, Peng, and Walkington [SODA'2013]) and their sub-complexes (Black, Maxwell, Nayyeri, and Winkelman [SODA'2022], Black and Nayyeri [ICALP'2022]). Additionally, quadratic time solvers by Nested Dissection exist for more general systems whose nonzero structures encode well-shaped simplicial complexes embedded in \mathbb{R}^3 .

We generalize the specialized solvers for 1-Laplacians to *weighted* simplicial complexes with additional geometric structures and improve the runtime of Nested Dissection. Specifically, we consider simplicial complexes embedded in \mathbb{R}^3 such that: (1) the complex triangulates a convex ball in \mathbb{R}^3 , (2) the underlying topological space of the complex is convex and has a bounded aspect ratio, and (3) each tetrahedron has a bounded aspect ratio and volume. We say such a simplicial complex is *stable*. We can approximately solve weighted 1-Laplacian systems in a stable simplicial complex with n simplexes up to high precision in time $\tilde{O}(n^{3/2})$ ¹ if the ratio between the maximum and minimum simplex weights is $\tilde{O}(n^{1/6})$. In addition, we generalize this solver to *a union of stable simplicial complex chunks*. As a result, our solver has a comparable runtime, parameterized by the number of chunks and the number of simplexes shared by more than one chunk. Our solvers are inspired by the Incomplete Nested Dissection designed by Kyng, Peng, Schwieterman, and Zhang [STOC'2018] for stiffness matrices of well-shaped trusses.

¹In this paper, we let $\tilde{O}(\cdot)$ hide poly-logarithmic factors on the number of simplexes, the ratio between the maximum and the minimum weights, and the inverse of the error parameter. We say an algorithm has a nearly-linear runtime if for any input of size n the algorithm runs in time $\tilde{O}(n)$.

1 Introduction

Combinatorial Laplacians generalize Graph Laplacians to higher dimensional simplicial complexes – a collection of 0-simplexes (vertices), 1-simplexes (edges), 2-simplexes (triangles), and their higher dimensional counterparts. Given an oriented d -dimensional simplicial complex \mathcal{K} , for each $0 \leq i \leq d$, let \mathcal{C}_i be the vector space generated by the i -simplexes in \mathcal{K} with coefficients in \mathbb{R} . We can define a sequence of boundary operators:

$$\mathcal{C}_d \xrightarrow{\partial_d} \mathcal{C}_{d-1} \xrightarrow{\partial_{d-1}} \cdots \xrightarrow{\partial_2} \mathcal{C}_1 \xrightarrow{\partial_1} \mathcal{C}_0,$$

where each ∂_i is a linear map that maps every i -simplex to a signed sum of its boundary $(i-1)$ -faces. In addition, each simplex in \mathcal{K} is assigned a positive weight, which encodes additional combinatorial and geometric information about the complex [HJ13, OPW20]. For each $0 \leq i \leq d$, let \mathbf{W}_i be a diagonal matrix whose diagonals are the weights of the i -simplexes in \mathcal{K} . We define the *weighted* i -Laplacian $\mathbf{L}_i : \mathcal{C}_i \rightarrow \mathcal{C}_i$ to be

$$\mathbf{L}_i = \partial_{i+1} \mathbf{W}_{i+1} \partial_{i+1}^\top + \partial_i^\top \mathbf{W}_{i-1} \partial_i. \quad (1)$$

In particular, ∂_1 is the vertex-edge incidence matrix, and \mathbf{L}_0 is the graph Laplacian (following the convention, we define $\partial_0 = \mathbf{0}$). We say an i -Laplacian is unweighted if both \mathbf{W}_{i+1} and \mathbf{W}_{i-1} are multiples of the identity matrices.

Combinatorial Laplacians play a key role in computational topology and topological data analysis [Chu93, Zom05, Ghr08, Car09, EH10, CdSGO16, Lim20]. One can encode the relations between data points in a metric space as a simplicial complex and study its topological properties, which capture higher-order features beyond connectivity and clustering. Among these applications, computing homology groups and their ranks, known as the Betti numbers, are central problems. By the discrete Hodge decomposition [Eck44], the i th homology group of \mathcal{K} is isomorphic to the kernel of the i -Laplacians \mathbf{L}_i . Thus, we can compute the i th Betti number of \mathcal{K} by solving a logarithmic number of systems of linear equations in \mathbf{L}_i [BV21].

It is well-known that systems of linear equations in graph Laplacians can be approximately solved in nearly-linear time in the number of nonzeros of the system [ST14, KMP10, KMP11, KOSZ13, LS13, PS14, CKM⁺14, KS16, KLP⁺16, JS21]. However, nearly-linear time solvers for 1-Laplacian systems are only known for very restricted classes of *unweighted* simplicial complexes, such as simplicial complexes with a known collapsing sequence [CFM⁺14] and their subcomplexes with bounded first Betti numbers [BMNW22, BN22]. One concrete example studied in the above papers is convex simplicial complexes that piecewise linearly triangulate a convex ball in \mathbb{R}^3 (also called a three-ball), for which a collapsing sequence exists and can be computed in linear time [Chi67, Chi80]. Since a simplicial complex embedded in \mathbb{R}^3 can always be extended to a three-ball in quadratic time, one can solve 1-Laplacian systems for unweighted simplicial complexes in \mathbb{R}^3 with bounded first Betti numbers in quadratic time [BMNW22, BN22].

On the contrary, 1-Laplacian systems for general simplicial complexes embedded in \mathbb{R}^4 are as hard to solve as general sparse linear equations [DKGZ22], for which the best-known algorithms need super-quadratic time [PV21, Nie22]. Moreover, deciding whether a simplicial complex has a collapsing sequence is NP-hard [Tan16]. All the above motivates the following questions²:

1. Can we efficiently solve 1-Laplacian systems for *weighted* convex simplicial complexes that triangulate a three-ball?

²Both questions were mentioned as a natural generalization or an interesting question in [CFM⁺14, BN22].

2. Can we efficiently solve 1-Laplacian systems for other classes of simplicial complexes *without collapsing sequences*?

Besides the specialized solvers for 1-Laplacian systems mentioned above, Nested Dissection can solve 1-Laplacian systems in quadratic time for *weighted* simplicial complexes in \mathbb{R}^3 with additional geometric structures [Geo73, LRT79, MT90]. It requires that each tetrahedron has a bounded aspect ratio³, and there are not many boundary simplexes.

Inspired by these solvers that utilize geometric structures, we design efficient 1-Laplacian solvers for weighted and well-shaped simplicial complexes. We say a simplicial complex \mathcal{K} linearly embedded in \mathbb{R}^3 is *stable* if: (1) the underlying topological space of \mathcal{K} is convex and \mathcal{K} triangulates a three-ball, (2) the aspect ratio of each tetrahedron of \mathcal{K} is bounded, (3) the aspect ratio of the underlying topological space of \mathcal{K} is bounded, and (4) the volume of each tetrahedron of \mathcal{K} is bounded from below and above. Our assumptions about \mathcal{K} are mild. The first assumption is required for the nearly-linear time solvers for unweighted 1-Laplacians in [CFM⁺14, BMNW22, BN22]. The second assumption is necessary for Nested Dissection [MT90, MTTV98]. In addition, Nested Dissection requires that \mathcal{K} does not have too many boundary simplexes. We do not explicitly have this assumption⁴.

We informally state our results below.

Theorem 1.1 (Informal statement). *Let \mathcal{K} be a weighted stable simplicial complex with n simplexes. For any $\epsilon > 0$, we can approximately solve a system in the 1-Laplacian of \mathcal{K} within error ϵ in time $O(n^{3/2} + U^{1/2}n^{17/12} \log(nU/\epsilon))$, where U is the ratio between the maximum and the minimum triangle weights in \mathcal{K} .*

If $U = \tilde{O}(n^{1/6})$, the solver in Theorem 1.1 has runtime $\tilde{O}(n^{3/2})$. If $U = o(n^{7/6})$, the solver has runtime $o(n^2)$, asymptotically faster than Nested Dissection.

We also examine unions of weighted stable simplicial complexes glued together by identifying certain subsets of simplexes on their boundaries. Such a simplicial complex, called \mathcal{U} , *may not be embeddable in \mathbb{R}^3 , may not have a known collapsing sequence, and may have large Betti numbers*. So, the previously established methods from [CFM⁺14, BMNW22, BN22] and Nested Dissection are unsuitable for this scenario. Building on our algorithm for single stable simplicial complexes, we design an efficient algorithm for \mathcal{U} whose runtime depends sub-quadratically on the size of \mathcal{U} and polynomially on the number of chunks and the number of simplexes shared by more than one chunk.

Theorem 1.2 (Informal statement). *Let \mathcal{U} be a union of h weighted stable 3-complexes that are glued together by identifying certain subsets of their boundary simplexes. For any $\epsilon > 0$, we can solve a system in the 1-Laplacian of \mathcal{U} within error ϵ in time $O(n^{3/2}h^{3/2} + nk + h^3k^3 + n^{1/4}U^{1/2}(nh + h^2k^2 + n^{7/6}) \log(nU/\epsilon))$, where n is the number of simplexes in \mathcal{U} , k is the number of boundary simplexes shared by more than one complex chunk, and U is the ratio between the maximum and the minimum triangle weights in \mathcal{U} .*

When $h = \tilde{O}(1)$ and $k = \tilde{O}(n^{1/2})$, the solver in Theorem 1.2 has the same runtime as Theorem 1.1. When $h = o(n^{1/3})$, $k = o(n^{1/3})$ and $U = o(n^{5/6})$, the runtime is $o(n^2)$, asymptotically faster than Nested Dissection.

³The aspect ratio of a geometric shape S is the radius of the smallest ball containing S divided by the radius of the largest ball contained in S .

⁴We suspect our assumptions together imply \mathcal{K} does not have many boundary simplexes. But we will not particularly need it for our algorithm design.

By a minor modification, we can relax the third assumption for stable simplicial complexes to that the aspect ratio of the underlying topological space of \mathcal{K} is $O(n^{1/4})$, and Theorem 1.1 and 1.2 still hold.

1.1 Main Approach

1.1.1 Solver for 1-Laplacian Systems

Cohen, Fasy, Miller, Nayyeri, Peng, and Walkington [CFM⁺14] observed that the pseudo-inverse of the 1-Laplacian \mathbf{L}_1 is

$$\mathbf{L}_1^\dagger = (\mathbf{L}_1^{\text{up}})^\dagger + (\mathbf{L}_1^{\text{down}})^\dagger,$$

where $\mathbf{L}_1^{\text{up}} = \partial_2 \mathbf{W}_2 \partial_2^\top$ is called the *up-Laplacian* operator and $\mathbf{L}_1^{\text{down}} = \partial_1^\top \mathbf{W}_0 \partial_1$ the *down-Laplacian* operator. So, given any vector \mathbf{b} in the image of \mathbf{L}_1 , vector $\mathbf{x} = (\mathbf{L}_1^{\text{up}})^\dagger \mathbf{b} + (\mathbf{L}_1^{\text{down}})^\dagger \mathbf{b}$ is a solution to the system $\mathbf{L}_1 \mathbf{x} = \mathbf{b}$. One can approximate $(\mathbf{L}_1^{\text{down}})^\dagger \mathbf{b}$ via the nearly-linear time graph Laplacian solvers. So, the main difficulty is approximating $(\mathbf{L}_1^{\text{up}})^\dagger \mathbf{b}$.

1.1.2 The Approach in [CFM⁺14] for Approximating $(\mathbf{L}_1^{\text{up}})^\dagger \mathbf{b}$

Let \mathcal{K} be a convex simplicial complex that triangulates a three-ball. Since the first Betti number of \mathcal{K} is zero, we can compute an approximate projection of \mathbf{b} onto the image of ∂_2 in nearly-linear time by graph Laplacian solvers.

Given any \mathbf{b}^{up} in the image of ∂_2 , the algorithm in [CFM⁺14] approximately finds a vector \mathbf{x}^{up} such that $\partial_2 \partial_2^\top \mathbf{x}^{\text{up}} = \mathbf{b}^{\text{up}}$ by three steps: computes \mathbf{y} such that $\partial_2 \mathbf{y} = \mathbf{b}^{\text{up}}$, then approximately projects \mathbf{y} onto the image of ∂_2^\top and gets a new vector \mathbf{y}' , finally computes \mathbf{x}^{up} such that $\partial_2^\top \mathbf{x}^{\text{up}} = \mathbf{y}'$. Since the second Betti number of \mathcal{K} is zero, one can approximately project a vector onto the image of ∂_2^\top via the dual graph of \mathcal{K} and the nearly-linear time graph Laplacian solvers. In addition, the vectors \mathbf{y} and \mathbf{x}^{up} in the first and third steps can be computed in linear time by utilizing the collapsing sequence of \mathcal{K} .

Can we generalize this approach to weighted \mathcal{K} ? We discuss two simple attempts and explain their issues. One attempt of solving $\partial_2 \mathbf{W}_2 \partial_2^\top \mathbf{x}^{\text{up}} = \mathbf{b}^{\text{up}}$ is to first solve $\partial_2 \mathbf{W}_2^{1/2} \mathbf{y} = \mathbf{b}^{\text{up}}$, then project \mathbf{y} onto the image of $\mathbf{W}_2^{1/2} \partial_2^\top$ and get \mathbf{y}' , and finally solve $\mathbf{W}_2^{1/2} \partial_2^\top \mathbf{x}^{\text{up}} = \mathbf{y}'$. However, the image of $\mathbf{W}_2^{1/2} \partial_2^\top$ can be quite different from the image of ∂_2^\top , and thus, the graph Laplacian solvers for computing the projection are no longer helpful; in addition, the orthogonal projection onto the image of $\mathbf{W}_2^{1/2} \partial_2^\top$ is $\mathbf{W}_2^{1/2} \partial_2^\top (\partial_2 \mathbf{W}_2 \partial_2^\top)^\dagger \partial_2 \mathbf{W}_2^{1/2}$, without knowing more structures about $\mathbf{W}_2^{1/2} \partial_2^\top$, applying this projection operator is almost equally hard as solving a system in $\partial_2 \mathbf{W}_2 \partial_2^\top$. Another attempt is to first solve $\partial_2 \mathbf{y} = \mathbf{b}^{\text{up}}$, then solve $\mathbf{W}_2 \mathbf{z} = \mathbf{y}$, approximately project \mathbf{z} onto the image of ∂_2^\top and get \mathbf{z}' , and finally solve $\partial_2^\top \mathbf{x}^{\text{up}} = \mathbf{z}'$. However, the above \mathbf{x}^{up} may not satisfy $\partial_2 \mathbf{W}_2 \partial_2^\top \mathbf{x}^{\text{up}} = \mathbf{b}^{\text{up}}$ since we cannot find the “correct” \mathbf{y} in the first step. A correct \mathbf{y} needs to be in the image of $\mathbf{W}_2 \partial_2^\top$ so that there exists an \mathbf{x}^{up} satisfying $\partial_2 \mathbf{W}_2 \partial_2^\top \mathbf{x}^{\text{up}} = \partial_2 \mathbf{y} = \mathbf{b}^{\text{up}}$. Since the image of $\mathbf{W}_2 \partial_2^\top$ and the kernel of ∂_2 are not orthogonal unless \mathbf{W}_2 is a multiple of the identity matrix, such a \mathbf{y} must have a component in the kernel of ∂_2 , which we cannot find by solving $\partial_2 \mathbf{y} = \mathbf{b}^{\text{up}}$ only.

1.1.3 Our Approach for Approximating $(\mathbf{L}_1^{\text{up}})^\dagger \mathbf{b}$

Let \mathcal{K} be a *stable* simplicial complex with positive simplex weights as in Theorem 1.1, and let $F \cup C$ be a partition of the edges of \mathcal{K} . We will describe how to find a good partition shortly. We write

the matrix $\mathbf{L}_1^{\text{up}} = \partial_2 \mathbf{W}_2 \partial_2^\top$ as a block matrix:

$$\mathbf{L}_1^{\text{up}} = \begin{pmatrix} \mathbf{L}_1^{\text{up}}[F, F] & \mathbf{L}_1^{\text{up}}[F, C] \\ \mathbf{L}_1^{\text{up}}[C, F] & \mathbf{L}_1^{\text{up}}[C, C] \end{pmatrix}.$$

We have the following identity:

$$\mathbf{L}_1^{\text{up}} = \begin{pmatrix} \mathbf{I} & \\ \mathbf{L}_1^{\text{up}}[C, F] \mathbf{L}_1^{\text{up}}[F, F]^\dagger & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{L}_1^{\text{up}}[F, F] & \\ & \text{Sc}[\mathbf{L}_1^{\text{up}}]_C \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{L}_1^{\text{up}}[F, F]^\dagger \mathbf{L}_1^{\text{up}}[F, C] \\ & \mathbf{I} \end{pmatrix}, \quad (2)$$

where

$$\text{Sc}[\mathbf{L}_1^{\text{up}}]_C = \mathbf{L}_1^{\text{up}}[C, C] - \mathbf{L}_1^{\text{up}}[C, F] \mathbf{L}_1^{\text{up}}[F, F]^\dagger \mathbf{L}_1^{\text{up}}[F, C]$$

is the Schur complement onto C . We can solve a system in \mathbf{L}_1^{up} by solving two systems in $\mathbf{L}_1^{\text{up}}[F, F]$ and one system in $\text{Sc}[\mathbf{L}_1^{\text{up}}]_C$ plus a constant number of matrix-vector multiplications (see Lemma 4.7 for more details). The partition $F \cup C$ determines how fast we can solve systems in $\mathbf{L}_1^{\text{up}}[F, F]$ and $\text{Sc}[\mathbf{L}_1^{\text{up}}]_C$ and thus the system in \mathbf{L}_1^{up} .

Our solver is inspired by the “Incomplete Nested Dissection” algorithm designed by Kyng, Peng, Schwieterman, and Zhang [KPSZ18], which solves systems in stiffness matrices of 3-dimensional well-shaped trusses in sub-quadratic time. Nested Dissection accelerates Gaussian elimination when the nonzero structure of the coefficient matrix encodes a graph whose subgraphs can be separated into balanced pieces after removing a few vertices. A carefully chosen F can guarantee the Schur complement obtained by “partial” Gaussian elimination does not have many nonzeros. In particular, Nested Dissection can solve a system in \mathbf{L}_1^{up} for a stable \mathcal{K} in quadratic time⁵. Incomplete Nested Dissection improves Nested Dissection by further sparsifying the Schur complement. Concretely, we first “cut” \mathcal{K} into well-shaped and “interior-disjoint” pieces of small size each. Then, we let F contain all the “interior” edges of each piece and C the “boundary” edges. Matrix $\mathbf{L}_1^{\text{up}}[F, F]$ is block diagonal, and we can solve a system in $\mathbf{L}_1^{\text{up}}[F, F]$ by applying Nested Dissection to each diagonal block. Then, we solve the system in the Schur complement onto C by Preconditioned Conjugate Gradient (PCG), with the preconditioner being the union of the boundary of each piece. In each PCG iteration, we solve a system in the preconditioner by Nested Dissection, which is much faster than solving a system in the Schur complement. To guarantee the quality of the preconditioner, we need each piece to be well-shaped, which requires additional assumptions on \mathcal{K} compared to Nested Dissection.

The main challenge of adapting the Incomplete Nested Dissection algorithm for 3-dimensional well-shaped truss stiffness matrices to up-Laplacians \mathbf{L}_1^{up} is that the image of the former has a clear characterization, but the latter’s image does not. This requires constructing (approximate) projection operators onto the image of ∂_2 . Although the projection operator from [CFM⁺14] is applicable when \mathcal{K} is a stable complex, a different operator will be necessary when \mathcal{K} is a union of stable simplicial complexes because its first Betti number is no longer zero. Similar challenges arise when solving a system $\mathbf{L}_1^{\text{up}} \mathbf{x}^{\text{up}} = \mathbf{b}^{\text{up}}$ where \mathbf{b}^{up} is in the image of \mathbf{L}_1^{up} . We solve this system by breaking it down into smaller sub-systems, so care must be taken to ensure that the right-hand side of each sub-system remains in the image of the coefficient matrix. Furthermore, constructing a PCG preconditioner whose image is the same as the Schur complement requires additional work. Overcoming these difficulties, we can adapt the Incomplete Nested Dissection algorithm to solve a system in \mathbf{L}_1^{up} of a stable simplicial complex in sub-quadratic time, slightly *worse* than the runtime stated in Theorem 1.1.

⁵We remark that Nested Dissection only needs (1) each tetrahedron of \mathcal{K} has bounded aspect ratio, and (2) the number of boundary vertices of \mathcal{K} is at most $O(n^{2/3})$, where n is the number of simplexes in \mathcal{K} .

In this paper, we further improve the Incomplete Nested Dissection by accelerating the solver for the PCG preconditioner. Different from [KPSZ18], we do not solve the preconditioner systems by Nested Dissection. Instead, we apply the matrix identity in Equation (2) again and solve the “ F ” part in linear time. For this, we cut \mathcal{K} more carefully so that the union of the boundary of each piece (our preconditioner) is a set of triangulated discs glued together via their boundary edges. We let C_F contain the interior edges of each disc and C_C the boundary edges. Note that edges in C_F are incident to exactly two triangles and edges in C_C are incident to more than two triangles. Denote the up-Laplacian operator of the preconditioner by $\mathbf{L}_{1,\mathcal{T}}^{\text{up}}$. Systems in $\mathbf{L}_{1,\mathcal{T}}^{\text{up}}[C_F, C_F]$ can be solved in linear time. To further speed up solvers for the Schur complement, we observe that most of the rows for disc boundary edges are linearly dependent, and thus removing them does not affect the solution of the system. This reduces the system in the Schur complement to a system with much smaller dimensions, which can be inverted efficiently.

Approximating $(\mathbf{L}_1^{\text{up}})^\dagger \mathbf{b}$ when \mathbf{L}_1^{up} is the up-Laplacian of a union of stable simplicial complexes, denoted by \mathcal{U} , is more challenging. It poses two main difficulties: (1) \mathcal{U} may not be embeddable in \mathbb{R}^3 , so Nested Dissection or Incomplete Nested Dissection cannot be used; (2) the nontrivial Betti numbers of \mathcal{U} means the approximate projection operators from [CFM⁺14] based on trivial Betti numbers cannot be used. To approximately solve the system $\mathbf{L}_1^{\text{up}} \mathbf{x} = \mathbf{b}^{\text{up}}$ for any \mathbf{b}^{up} in the image of \mathbf{L}_1^{up} , we generalize our ideas for a single stable complex by including the edges “near” the simplexes shared by multiple complex chunks into set C . If there are not many such simplexes, we can efficiently solve a system in the Schur complement onto C . For approximating the projection operator, note that the projection of \mathbf{b} onto the image of \mathbf{L}_1^{up} is $\partial_2(\partial_2^\top \partial_2)^\dagger \partial_2^\top \mathbf{b}$. We approximate this operator by implicitly solving a system in $\partial_2^\top \partial_2$, which is the second down-Laplacian operator assuming unit edge weight. We employ the same idea again. We partition the *triangles* of \mathcal{U} into $F \cup C$, where C contains all the triangles “near” the simplexes shared by more than one chunk and F contains all the other triangles. The subsystem for the F part corresponds to unweighted 3-complexes and can be efficiently solved by [CFM⁺14, BMNW22]. The Schur complement onto C can be inverted efficiently if only a few simplexes are shared by more than one chunk.

Organization of the Remaining Paper. In Section 2, we provide background knowledge related to linear algebra and topology in the paper. In Section 3, we formally state our main theorems. In Section 4, we overview our algorithm ideas. Then, we prove our main theorem for stable simplicial complexes in Section 5 6, 7, and 8. Finally, we prove our main theorem for unions of stable simplicial complexes in Section 9.

2 Preliminaries

2.1 Background of Linear Algebra

Given a vector $\mathbf{x} \in \mathbb{R}^n$, for $1 \leq i \leq n$, we let $\mathbf{x}[i]$ be the i th entry of \mathbf{x} ; for $1 \leq i < j \leq n$, let $\mathbf{x}[i : j]$ be $(\mathbf{x}[i], \mathbf{x}[i+1], \dots, \mathbf{x}[j])^\top$. The Euclidean norm of \mathbf{x} is $\|\mathbf{x}\|_2 \stackrel{\text{def}}{=} \sqrt{\sum_{i=1}^n \mathbf{x}[i]^2}$. Given a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, for $1 \leq i \leq m, 1 \leq j \leq n$, we let $\mathbf{A}[i, j]$ be the (i, j) th entry of \mathbf{A} ; for $S_1 \subseteq \{1, \dots, m\}, S_2 \subseteq \{1, \dots, n\}$, let $\mathbf{A}[S_1, S_2]$ be the submatrix with row indices in S_1 and column indices in S_2 . Furthermore, we let $\mathbf{A}[S_1, :] = \mathbf{A}[S_1, \{1, \dots, n\}]$ and $\mathbf{A}[:, S_2] = \mathbf{A}[\{1, \dots, m\}, S_2]$. The operator norm of \mathbf{A} (induced by the Euclidean norm) is $\|\mathbf{A}\|_2 \stackrel{\text{def}}{=} \max_{\mathbf{v} \in \mathbb{R}^n} \frac{\|\mathbf{A}\mathbf{v}\|_2}{\|\mathbf{v}\|_2}$. The image of \mathbf{A} is the linear span of the columns of \mathbf{A} , denoted by $\text{Im}(\mathbf{A})$, and the kernel of \mathbf{A} to be $\{\mathbf{x} \in \mathbb{R}^n : \mathbf{A}\mathbf{x} = \mathbf{0}\}$, denoted by $\text{Ker}(\mathbf{A})$. A fundamental theorem of Linear Algebra states $\mathbb{R}^m = \text{Im}(\mathbf{A}) \oplus \text{Ker}(\mathbf{A}^\top)$.

Fact 2.1. ⁶ For any matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\text{Im}(\mathbf{A}) = \text{Im}(\mathbf{A}\mathbf{A}^\top)$.

Pseudo-inverse and projection matrix. The pseudo-inverse of \mathbf{A} is defined to be a matrix \mathbf{A}^\dagger that satisfies all the following four criteria: (1) $\mathbf{A}\mathbf{A}^\dagger\mathbf{A} = \mathbf{A}$, (2) $\mathbf{A}^\dagger\mathbf{A}\mathbf{A}^\dagger = \mathbf{A}^\dagger$, (3) $(\mathbf{A}\mathbf{A}^\dagger)^\top = \mathbf{A}\mathbf{A}^\dagger$, (4) $(\mathbf{A}^\dagger\mathbf{A})^\top = \mathbf{A}^\dagger\mathbf{A}$. The orthogonal projection matrix onto $\text{Im}(\mathbf{A})$ is $\Pi_{\text{Im}(\mathbf{A})} = \mathbf{A}(\mathbf{A}^\top\mathbf{A})^\dagger\mathbf{A}^\top$.

Eigenvalues and condition numbers. Given a square matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, let $\lambda_{\max}(\mathbf{A})$ be the maximum eigenvalue of \mathbf{A} and $\lambda_{\min}(\mathbf{A})$ the minimum *nonzero* eigenvalue of \mathbf{A} . The condition number of \mathbf{A} , denoted by $\kappa(\mathbf{A})$, is the ratio between $\lambda_{\max}(\mathbf{A})$ and $\lambda_{\min}(\mathbf{A})$. A symmetric matrix \mathbf{A} is *positive semi-definite (PSD)* if all eigenvalues of \mathbf{A} are non-negative. Let $\mathbf{B} \in \mathbb{R}^{n \times n}$ be another square matrix. We say $\mathbf{A} \succcurlyeq \mathbf{B}$ if $\mathbf{A} - \mathbf{B}$ is PSD. The *condition number of \mathbf{A} relative to \mathbf{B}* is

$$\kappa(\mathbf{A}, \mathbf{B}) \stackrel{\text{def}}{=} \min \left\{ \frac{\alpha}{\beta} : \beta \mathbf{B} \preccurlyeq \mathbf{A} \preccurlyeq \alpha \mathbf{B} \right\}.$$

Fact 2.2. Let $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$ be symmetric matrices such that $\mathbf{A} \preccurlyeq \mathbf{B}$. Then, for any $\mathbf{V} \in \mathbb{R}^{m \times n}$, $\mathbf{V}\mathbf{A}\mathbf{V}^\top \preccurlyeq \mathbf{V}\mathbf{B}\mathbf{V}^\top$.

Schur Complement. Let $\mathbf{A} \in \mathbb{R}^{n \times n}$, and let $F \cup C$ be a partition of $\{1, \dots, n\}$. We write \mathbf{A} as a block matrix:

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}[F, F] & \mathbf{A}[F, C] \\ \mathbf{A}[C, F] & \mathbf{A}[C, C] \end{pmatrix}. \quad (3)$$

We define the (generalized) Schur complement of \mathbf{A} onto C to be

$$\text{Sc}[\mathbf{A}]_C = \mathbf{A}[C, C] - \mathbf{A}[C, F]\mathbf{A}[F, F]^\dagger\mathbf{A}[F, C].$$

The Schur complement appears in performing a block Gaussian elimination on matrix \mathbf{A} to eliminate the indices in F .

Fact 2.3. Let \mathbf{A} be a PSD matrix defined in Equation (3). Then,

$$\mathbf{A} = \begin{pmatrix} \mathbf{I} & \\ \mathbf{A}[C, F]\mathbf{A}[F, F]^\dagger & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{A}[F, F] & \\ & \text{Sc}[\mathbf{A}]_C \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{A}[F, F]^\dagger\mathbf{A}[F, C] \\ & \mathbf{I} \end{pmatrix}.$$

Fact 2.4. Let \mathbf{A} be a PSD matrix defined in Equation (3). Let $\mathbf{A} = \mathbf{B}\mathbf{B}^\top$, and we decompose $\mathbf{B} = \begin{pmatrix} \mathbf{B}_F \\ \mathbf{B}_C \end{pmatrix}$ accordingly. Then, $\text{Sc}[\mathbf{A}]_C = \mathbf{B}_C\Pi_{\text{Ker}(\mathbf{B}_F)}\mathbf{B}_C^\top$, where $\Pi_{\text{Ker}(\mathbf{B}_F)}$ is the projection onto the kernel of \mathbf{B}_F .

Solving Linear Equations. We will need Fact 2.5 for relations between different error notations for linear equations and Theorem 2.6 for Preconditioned Conjugate Gradient.

Fact 2.5. Let $\mathbf{A}, \mathbf{Z} \in \mathbb{R}^{n \times n}$ be two symmetric PSD matrices, and let Π be the orthogonal projection onto $\text{Im}(\mathbf{A})$.

1. If $(1 - \epsilon)\mathbf{A}^\dagger \preccurlyeq \mathbf{Z} \preccurlyeq (1 + \epsilon)\mathbf{A}^\dagger$, then $\|\mathbf{A}\mathbf{Z}\mathbf{b} - \mathbf{b}\|_2 \leq \epsilon\sqrt{\kappa(\mathbf{A})}\|\mathbf{b}\|_2$ for any $\mathbf{b} \in \text{Im}(\mathbf{A})$.

⁶All the facts in this section are well-known. For completeness, we include their proofs in Appendix A.1.

2. If $\|\mathbf{A}\mathbf{Z}\mathbf{b} - \mathbf{b}\|_2 \leq \epsilon \|\mathbf{b}\|_2$ for any $\mathbf{b} \in \text{Im}(\mathbf{A})$, then $(1 - \epsilon)\mathbf{A}^\dagger \preceq \mathbf{\Pi}\mathbf{Z}\mathbf{\Pi} \preceq (1 + \epsilon)\mathbf{A}^\dagger$.

Theorem 2.6 (Preconditioned Conjugate Gradient [Axe85]). *Let $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$ be two symmetric PSD matrices, and let $\mathbf{b} \in \mathbb{R}^n$. Each iteration of Preconditioned Conjugate Gradient multiplies one vector with \mathbf{A} , solves one system of linear equations in \mathbf{B} , and performs a constant number of vector operations. For any $\epsilon > 0$, the algorithm outputs an \mathbf{x} satisfying $\|\mathbf{A}\mathbf{x} - \mathbf{\Pi}_\mathbf{A}\mathbf{b}\|_2 \leq \epsilon \|\mathbf{\Pi}_\mathbf{A}\mathbf{b}\|_2$ in $O(\sqrt{\kappa} \log(\kappa/\epsilon))$ such iterations, where $\mathbf{\Pi}_\mathbf{A}$ is the orthogonal projection matrix onto the image of \mathbf{A} and $\kappa = \kappa(\mathbf{A}, \mathbf{B})$.*

2.2 Background of Topology

Simplicial Complexes and Geometric Assumptions. We consider a d -simplex (or d -dimensional simplex) σ as an ordered set of $d + 1$ vertices, denoted by $\sigma = [v_0, \dots, v_d]$. A *face* of σ is a simplex obtained by removing a subset of vertices from σ . A *simplicial complex* \mathcal{K} is a finite collection of simplexes such that (1) for every $\sigma \in \mathcal{K}$ if $\tau \subset \sigma$ then $\tau \in \mathcal{K}$, and (2) for every $\sigma_1, \sigma_2 \in \mathcal{K}$, $\sigma_1 \cap \sigma_2$ is either empty or a face of both σ_1, σ_2 . The *dimension* of \mathcal{K} is the maximum dimension of any simplex in \mathcal{K} . A d -complex is a d -dimensional simplicial complex. For $1 \leq i \leq d$, the i -skeleton of a d -complex \mathcal{K} is the subcomplex consisting of all the simplexes of \mathcal{K} of dimensions at most i . In particular, the 1-skeleton of \mathcal{K} is a graph.

A *piecewise linear embedding* of a 3-complex in \mathbb{R}^3 maps a 0-simplex to a point, a 1-simplex to a line segment, a 2-simplex to a triangle, and a 3-simplex to a tetrahedron. Such an embedding of a simplicial complex \mathcal{K} defines an *underlying topological space* \mathbb{K} – the union of the images of all the simplexes of \mathcal{K} . We say \mathcal{K} is *convex* if \mathbb{K} is convex. We say \mathcal{K} *triangulates* a topological space \mathbb{X} if \mathbb{K} is homeomorphic to \mathbb{X} . A simplex σ of \mathcal{K} is a *boundary* simplex if σ is contained in the boundary of \mathbb{K} , and σ is an *interior* simplex otherwise.

In this paper, we are interested in simplicial complexes with additional geometric structures. The *aspect ratio* of a set $S \subset \mathbb{R}^3$ is the radius of the smallest ball containing S divided by the radius of the largest ball contained in S . The aspect ratio of S is always greater than or equal to 1. We say a simplicial complex \mathcal{K} embedded in \mathbb{R}^3 *stable* if (1) its underlying topological space \mathbb{K} is convex and \mathcal{K} triangulates a three-ball $\{x \in \mathbb{R}^3 : \|\mathbf{x}\|_2 \leq 1\}$, (2) the aspect ratios of individual tetrahedrons of \mathcal{K} are $O(1)$, (3) the aspect ratio of \mathbb{K} is $O(1)$, and (4) the volume of each tetrahedron of \mathcal{K} is $\Theta(1)$. Miller and Thurston proved the following lemma. As a corollary, the numbers of the vertices, the edges, the triangles, and the tetrahedrons of a stable 3-complex \mathcal{K} are all equal up to a constant factor.

Lemma 2.7 (Lemma 4.1 of [MT90]). *Let \mathcal{K} be a 3-complex in \mathbb{R}^3 in which each tetrahedron has $O(1)$ aspect ratio. Then, each vertex of \mathcal{K} is contained in at most $O(1)$ tetrahedrons.*

Boundary Operators and Combinatorial Laplacians. An i -chain is a weighted sum of the oriented i -simplexes in \mathcal{K} with the coefficients in \mathbb{R} . Let \mathcal{C}_i denote the i th chain space. The *boundary operator* is a linear map $\partial_i : \mathcal{C}_i \rightarrow \mathcal{C}_{i-1}$ such that for an oriented i -simplex $\sigma = [v_0, v_1, \dots, v_i]$,

$$\partial_i(\sigma) = \sum_{j=0}^i (-1)^j [v_0, \dots, \hat{v}_j, \dots, v_i],$$

where $[v_0, \dots, \hat{v}_j, \dots, v_i]$ is the oriented $(i - 1)$ -simplex obtained by removing v_j from σ . The operator ∂_i can be written as a matrix in $|\mathcal{C}_{i-1}| \times |\mathcal{C}_i|$ dimensions, where the (r, l) th entry of ∂_i is ± 1 if the r th $(i - 1)$ -simplex is a face of the l th i -simplex and 0 otherwise.

An important property of the boundary operators is $\partial_i \partial_{i+1} = \mathbf{0}$, which implies $\text{Im}(\partial_{i+1}) \subseteq \text{Ker}(\partial_i)$. So, we can define the quotient space $H_i = \text{Ker}(\partial_i) \setminus \text{Im}(\partial_{i+1})$, called the *ith homology space* of \mathcal{K} . The rank of H_i is called the *ith Betti number* of \mathcal{K} . If the *ith Betti number* of \mathcal{K} is 0, then $\text{Im}(\partial_i^\top) \oplus \text{Im}(\partial_{i+1}) = \mathbb{R}^{|\mathcal{C}_i|}$. The first and second Betti numbers of a triangulation of a three-ball are both 0.

For each $1 \leq i \leq d$, we assign each i -simplex of \mathcal{K} with a *positive weight*, and let $\mathbf{W}_i : \mathcal{C}_i \rightarrow \mathcal{C}_i$ be a diagonal matrix where $\mathbf{W}_i[\sigma, \sigma]$ is the weighted of the i -simplex σ . The (weighted) *i-Laplacian* of \mathcal{K} is a linear operator $\mathbf{L}_i : \mathcal{C}_i \rightarrow \mathcal{C}_i$ defined as

$$\mathbf{L}_i \stackrel{\text{def}}{=} \partial_i^\top \mathbf{W}_{i-1} \partial_i + \partial_{i+1} \mathbf{W}_{i+1} \partial_{i+1}^\top.$$

We call $\mathbf{L}_i^{\text{down}} \stackrel{\text{def}}{=} \partial_i^\top \mathbf{W}_{i-1} \partial_i$ the *ith down-Laplacian* operator and $\mathbf{L}_i^{\text{up}} \stackrel{\text{def}}{=} \partial_{i+1} \mathbf{W}_{i+1} \partial_{i+1}^\top$ the *ith up-Laplacian* operator. Sometimes, we use subscripts to specify the complex on which these operators are defined: $\partial_{i,\mathcal{K}}, \mathbf{W}_{i,\mathcal{K}}, \mathbf{L}_{i,\mathcal{K}}^{\text{down}}, \mathbf{L}_{i,\mathcal{K}}^{\text{up}}$.

3 Main Theorems

We formally state our main results as follows.

Theorem 3.1. *Let \mathcal{K} be a stable 3-complex whose 1-Laplacian operator is \mathbf{L}_1 . Let $\mathbf{\Pi}_1$ be the orthogonal projection matrix onto the image of \mathbf{L}_1 . For any $\mathbf{b} \in \mathbb{R}^n$ and $\epsilon > 0$, we can find a solution $\tilde{\mathbf{x}}$ such that*

$$\|\mathbf{L}_1 \tilde{\mathbf{x}} - \mathbf{\Pi}_1 \mathbf{b}\|_2 \leq \epsilon \|\mathbf{\Pi}_1 \mathbf{b}\|_2$$

in time $O(n^{3/2} + U^{1/2} n^{17/12} \log(nU/\epsilon))$, where n is the number of simplexes in \mathcal{K} and U is the ratio between the maximum and the minimum triangle weights in \mathcal{K} .

We will overview our algorithm for Theorem 3.1 in Section 4 and provide detailed proofs in Section 5, 6, 7, and 8.

Theorem 3.2. *Let \mathcal{U} be a union of h stable 3-complexes that are glued together by identifying certain subsets of their boundary simplexes. Let \mathbf{L}_1 be the 1-Laplacian operator of \mathcal{U} , and let $\mathbf{\Pi}_1$ be the orthogonal projection matrix onto the image of \mathbf{L}_1 . For any $\mathbf{b} \in \mathbb{R}^n$ and $\epsilon > 0$, we can find a solution $\tilde{\mathbf{x}}$ such that*

$$\|\mathbf{L}_1 \tilde{\mathbf{x}} - \mathbf{\Pi}_1 \mathbf{b}\|_2 \leq \epsilon \|\mathbf{\Pi}_1 \mathbf{b}\|_2$$

in time $O(n^{3/2} h^{3/2} + nk + h^3 k^3 + n^{1/4} U^{1/2} (nh + h^2 k^2 + n^{7/6}) \log(nU/\epsilon))$, where n is the number of simplexes in \mathcal{U} , k is the number of simplexes shared by more than one chunk, and U is the ratio between the maximum and the minimum triangle weights in \mathcal{U} .

We will defer our proof of Theorem 3.2 to Section 9.

4 Algorithm Overview for Stable Simplicial Complexes

We follow the framework of solving systems of linear equations in 1-Laplacians by Cohen, Fasy, Miller, Nayyeri, Peng, and Walkington [CFM⁺14]. Recall $\mathbf{L}_1^{\text{down}} = \partial_1^\top \mathbf{W}_0 \partial_1$ and $\mathbf{L}_1^{\text{up}} = \partial_2 \mathbf{W}_2 \partial_2^\top$. Since $\partial_1 \partial_2 = \mathbf{0}$, we have $\text{Im}(\mathbf{L}_1^{\text{down}}) \perp \text{Im}(\mathbf{L}_1^{\text{up}})$ and

$$\mathbf{L}_1^\dagger = \left(\mathbf{L}_1^{\text{down}}\right)^\dagger + \left(\mathbf{L}_1^{\text{up}}\right)^\dagger.$$

The orthogonal projection matrices onto $\text{Im}(\partial_1^\top)$ and $\text{Im}(\partial_2)$ are:

$$\Pi_1^{\text{down}} \stackrel{\text{def}}{=} \partial_1^\top (\partial_1 \partial_1^\top)^\dagger \partial_1, \quad \Pi_1^{\text{up}} \stackrel{\text{def}}{=} \partial_2 (\partial_2^\top \partial_2)^\dagger \partial_2^\top.$$

Lemma 4.1 (Lemma 4.1 of [CFM⁺14]). *Let $\mathbf{b} \in \text{Im}(\mathbf{L}_1)$. Consider the systems of linear equations: $\mathbf{L}_1 \mathbf{x} = \mathbf{b}$, $\mathbf{L}_1^{\text{up}} \mathbf{x}^{\text{up}} = \Pi_1^{\text{up}} \mathbf{b}$, $\mathbf{L}_1^{\text{down}} \mathbf{x}^{\text{down}} = \Pi_1^{\text{down}} \mathbf{b}$. Then, $\mathbf{x} = \Pi_1^{\text{up}} \mathbf{x}^{\text{up}} + \Pi_1^{\text{down}} \mathbf{x}^{\text{down}}$.*

Lemma 4.1 implies that four operators are needed to approximate \mathbf{L}_1^\dagger : (1) an approximate projection operator $\tilde{\Pi}_1^{\text{down}} \approx \Pi_1^{\text{down}}$, (2) an approximate projection operator $\tilde{\Pi}_1^{\text{up}} \approx \Pi_1^{\text{up}}$, (3) a down-Laplacian solver $\mathbf{Z}_1^{\text{down}}$ such that $\mathbf{L}_1^{\text{down}} \mathbf{Z}_1^{\text{down}} \mathbf{b} \approx \mathbf{b}$ for any $\mathbf{b}^{\text{down}} \in \text{Im}(\mathbf{L}_1^{\text{up}})$, and (4) an up-Laplacian solver \mathbf{Z}_1^{up} such that $\mathbf{L}_1^{\text{up}} \mathbf{Z}_1^{\text{up}} \mathbf{b} \approx \mathbf{b}$ for any $\mathbf{b} \in \text{Im}(\mathbf{L}_1^{\text{up}})$.

The orthogonal projections Π_1^{down} and Π_1^{up} are not affected by simplex weights. So, the approximate projection operators $\tilde{\Pi}_1^{\text{down}}$ and $\tilde{\Pi}_1^{\text{up}}$ from [CFM⁺14] work for weighted stable 3-complexes.

Lemma 4.2 (Approximate projection operators, Lemma 3.2 of [CFM⁺14]). *Let \mathcal{K} be a 3-complex with n simplexes. For any $\epsilon > 0$, there exists a linear operator $\tilde{\Pi}_1^{\text{down}}$ such that*

$$(1 - \epsilon) \Pi_1^{\text{down}} \preceq \tilde{\Pi}_1^{\text{down}}(\epsilon) \preceq \Pi_1^{\text{down}}.$$

If the first Betti number of \mathcal{K} is 0, there exists another operator $\tilde{\Pi}_1^{\text{up}}$ such that

$$(1 - \epsilon) \Pi_1^{\text{up}} \preceq \tilde{\Pi}_1^{\text{up}}(\epsilon) \preceq \Pi_1^{\text{up}}.$$

In addition, for any $\mathbf{b} \in \mathbb{R}^n$, we can compute $\tilde{\Pi}_1^{\text{down}}(\epsilon) \cdot \mathbf{b}$ and $\tilde{\Pi}_1^{\text{up}}(\epsilon) \mathbf{b}$ in time $\tilde{O}(n \log(1/\epsilon))$.

4.1 Solver for Down-Laplacian

This section shows how to solve $\mathbf{L}_1^{\text{down}} \mathbf{x}^{\text{down}} = \mathbf{b}$ for any $\mathbf{b} \in \text{Im}(\mathbf{L}_1^{\text{down}})$ in linear time. Recall that $\mathbf{L}_1^{\text{down}} = \partial_1^\top \mathbf{W}_0 \partial_1$. Our down-Laplacian solver works for *any* simplicial complexes and returns a solution *without error*. Our approach is a slight modification of the down-Laplacian solver in [CFM⁺14] to incorporate the vertex weights \mathbf{W}_0 . Specifically, we compute \mathbf{x}^{down} by three steps: solve $\partial_1^\top \mathbf{W}_0^{1/2} \mathbf{y} = \mathbf{b}$, project \mathbf{y} onto $\text{Im}(\mathbf{W}_0^{1/2} \partial_1)$ and get a new vector \mathbf{y}' , then solve $\mathbf{W}_0^{1/2} \partial_1 \mathbf{x}^{\text{down}} = \mathbf{y}'$. The first and the last steps can be solved by the approach in Lemma 4.2 of [CFM⁺14] (stated below), and the second step can be explicitly solved by utilizing that ∂_1 is a vertex-edge incidence matrix of an oriented graph.

Lemma 4.3 (A restatement of Lemma 4.2 of [CFM⁺14]). *Given any $\mathbf{b}_1 \in \text{Im}(\partial_1^\top)$ (respectively, $\mathbf{b}_0 \in \text{Im}(\partial_1)$), there is a linear operator $\partial_1^{+\top}$ such that $\partial_1^\top \partial_1^{+\top} \mathbf{b}_1 = \mathbf{b}_1$ (respectively, $\partial_1^+ = (\partial_1^{+\top})^\top$ such that $\partial_1 \partial_1^+ \mathbf{b}_0 = \mathbf{b}_0$). In addition, we can compute $\partial_1^{+\top} \mathbf{b}_1$ (respectively, $\partial_1^+ \mathbf{b}_0$) in linear time.*

We remark that the operators ∂_1^+ and $\partial_1^{+\top}$ in Lemma 4.3 are not necessary to be the pseudo-inverses of ∂_1 and ∂_1^\top .

Claim 4.4 explicitly characterizes the image of $\mathbf{W}_0^{1/2} \partial_1$.

Claim 4.4. *Let \mathcal{K} be a simplicial complex whose 1-skeleton is connected, and let v be the number of vertices in \mathcal{K} . Let $\mathbf{W}_0 = \text{diag}(w_1, \dots, w_v)$ where $w_1, \dots, w_v > 0$. Then,*

$$\text{Ker}(\partial_1^\top \mathbf{W}_0^{1/2}) = \text{span}\{\mathbf{u}\} \text{ where } \mathbf{u} = \left(\frac{1}{\sqrt{w_1}}, \dots, \frac{1}{\sqrt{w_v}} \right)^\top.$$

Proof. Let $\dim(\mathcal{V})$ be the dimension of a space \mathcal{V} . Since all the diagonals of \mathbf{W}_0 are positive,

$$\dim(\text{Ker}(\partial_1^\top \mathbf{W}_0^{1/2})) = \dim(\text{Ker}(\partial_1^\top)) = v - \dim(\text{Im}(\partial_1)) = 1,$$

where the last equality holds since the 1-skeleton of \mathcal{K} is connected. We can check that

$$\partial_1^\top \mathbf{W}_0^{1/2} \mathbf{u} = \partial_1^\top \mathbf{1} = \mathbf{0}.$$

Here, $\mathbf{1}$ is the all-one vector, and the second equality holds since ∂_1 is the vertex-edge incidence matrix of an oriented (weakly) connected graph. Thus, the statement holds. \square

We construct our down-Laplacian solver by combining Lemma 4.3 and Claim 4.4. We remark this down-Laplacian solver applies to any simplicial complex.

Lemma 4.5 (Down-Laplacian solver). *Let \mathcal{K} be a weighted simplicial complex, and let $\mathbf{b} \in \text{Im}(\mathbf{L}_1^{\text{down}})$. There exists an operator $\mathbf{Z}_1^{\text{down}}$ such that $\mathbf{L}_1^{\text{down}} \mathbf{Z}_1^{\text{down}} \mathbf{b} = \mathbf{b}$. In addition, we can compute $\mathbf{Z}_1^{\text{down}} \mathbf{b}$ in linear time.*

Proof. Without loss of generality, we assume the 1-skeleton of \mathcal{K} is connected. Otherwise, we can write $\mathbf{L}_1^{\text{down}}$ as a block diagonal matrix, each corresponding to a connected component of the 1-skeleton, and we reduce solving a system in $\mathbf{L}_1^{\text{down}}$ into solving several smaller down-Laplacian systems.

Let $\partial_1^+, \partial_1^{+\top}$ be the linear operators in Lemma 4.3 for \mathcal{K} , and let \mathbf{u} be the vector in Claim 4.4. We define

$$\mathbf{Z}_1^{\text{down}} \stackrel{\text{def}}{=} \partial_1^+ \mathbf{W}_0^{-1/2} \left(\mathbf{I} - \frac{\mathbf{u} \mathbf{u}^\top}{\|\mathbf{u}\|_2^2} \right) \mathbf{W}_0^{-1/2} \partial_1^{+\top}.$$

We can compute $\mathbf{Z}_1^{\text{down}} \mathbf{b}$ in linear time. In addition, we define

$$\mathbf{y} = \partial_1^{+\top} \mathbf{b}, \mathbf{z} = \mathbf{W}_0^{-1/2} \mathbf{y}, \mathbf{z}_1 = \left(\mathbf{I} - \frac{\mathbf{u} \mathbf{u}^\top}{\|\mathbf{u}\|_2^2} \right) \mathbf{z}, \text{ and } \mathbf{x} = \partial_1^+ \mathbf{W}_0^{-1/2} \mathbf{z}_1.$$

Then, $\mathbf{z}_1 \in \text{Im}(\mathbf{W}_0^{1/2} \partial_1)$ and $\mathbf{z} - \mathbf{z}_1 \in \text{Ker}(\partial_1^\top \mathbf{W}_0^{1/2})$.

$$\begin{aligned} \mathbf{L}_1^{\text{down}} \mathbf{Z}_1^{\text{down}} \mathbf{b} &= \partial_1^\top \mathbf{W}_0 \partial_1 \mathbf{x} = \partial_1^\top \mathbf{W}_0 \partial_1 \partial_1^+ \mathbf{W}_0^{-1/2} \mathbf{z}_1 \\ &= \partial_1^\top \mathbf{W}_0 \mathbf{W}_0^{-1/2} \mathbf{z}_1 && (\text{since } \mathbf{W}_0^{-1/2} \mathbf{z}_1 \in \text{Im}(\partial_1)) \\ &= \partial_1^\top \mathbf{W}_0^{1/2} \mathbf{z} && (\text{since } \mathbf{z} - \mathbf{z}_1 \in \text{Ker}(\partial_1^\top \mathbf{W}_0^{1/2})) \\ &= \partial_1^\top \mathbf{y} = \mathbf{b}. \end{aligned}$$

\square

4.2 Solver for Up-Laplacian

Our primary technical contribution in this work is the development of an efficient solver for the up-Laplacian system, stated in Lemma 4.6. We will describe the key idea behind our solver in this section.

Lemma 4.6 (Up-Laplacian solver for a stable 3-complex). *Let \mathcal{K} be a stable 3-complex with n simplexes. Then for any $\epsilon > 0$, there exists an operator \mathbf{Z}_1^{up} such that*

$$\forall \mathbf{b} \in \text{Im}(\mathbf{L}_1^{up}), \quad \|\mathbf{L}_1^{up} \mathbf{Z}_1^{up} \mathbf{b} - \mathbf{b}\|_2 \leq \epsilon \|\mathbf{b}\|_2.$$

In addition, $\mathbf{Z}_1^{up} \mathbf{b}$ can be computed in time $O(n^{3/2} + U^{1/2} n^{17/12} \log(nU/\epsilon))$, where U is the ratio between the maximum and the minimum triangle weights.

Suppose we partition the edges in \mathcal{K} into $F \cup C$. Recall the matrix identity in Equation (2):

$$\mathbf{L}_1^{up} = \begin{pmatrix} \mathbf{I} & \\ \mathbf{L}_1^{up}[C, F] \mathbf{L}_1^{up}[F, F]^\dagger & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{L}_1^{up}[F, F] & \\ & \text{Sc}[\mathbf{L}_1^{up}]_C \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{L}_1^{up}[F, F]^\dagger \mathbf{L}_1^{up}[F, C] \\ & \mathbf{I} \end{pmatrix},$$

where

$$\text{Sc}[\mathbf{L}_1^{up}]_C = \mathbf{L}_1^{up}[C, C] - \mathbf{L}_1^{up}[C, F] \mathbf{L}_1^{up}[F, F]^\dagger \mathbf{L}_1^{up}[F, C].$$

Lemma 4.7 reduces (approximately) solving a system in \mathbf{L}_1^{up} to (approximately) solving two systems in $\mathbf{L}_1^{up}[F, F]$ and one system in $\text{Sc}[\mathbf{L}_1^{up}]_C$. The statement of the lemma or its variants is not original and can be proved by manipulating matrices and vectors. Our proof is in Appendix A.2. It is worth noting that Lemma 4.7 holds if we replace \mathbf{L}_1^{up} with an arbitrary symmetric PSD matrix, and we will apply it or a variant of it for different PSD matrices in our solvers. To avoid introducing additional notations, we state the lemma below in terms of \mathbf{L}_1^{up} .

Lemma 4.7. *Suppose we have two operators (1) $\text{UPLAPFSOLVER}(\cdot)$ such that given any $\mathbf{b} \in \text{Im}(\mathbf{L}_1^{up}[F, F])$, $\text{UPLAPFSOLVER}(\mathbf{b})$ returns a vector \mathbf{x} satisfying $\mathbf{L}_1^{up}[F, F] \mathbf{x} = \mathbf{b}$, and (2) $\text{SCHURSOLVER}(\cdot, \cdot)$ such that for any $\mathbf{h} \in \text{Im}(\text{Sc}[\mathbf{L}_1^{up}]_C)$ and $\delta > 0$, $\text{SCHURSOLVER}(\mathbf{h}, \delta)$ returns $\tilde{\mathbf{x}}$ satisfying*

$$\|\text{Sc}[\mathbf{L}_1^{up}]_C \tilde{\mathbf{x}} - \mathbf{h}\|_2 \leq \delta \|\mathbf{h}\|_2.$$

Given any $\mathbf{b} = \begin{pmatrix} \mathbf{b}_F \\ \mathbf{b}_C \end{pmatrix} \in \text{Im}(\mathbf{L}_1^{up})$ and any $\epsilon > 0$, let

$$\begin{aligned} \mathbf{h} &= \mathbf{b}_C - \mathbf{L}_1^{up}[C, F] \cdot \text{UPLAPFSOLVER}(\mathbf{b}_F), \\ \tilde{\mathbf{x}}_C &= \text{SCHURSOLVER}(\mathbf{h}, \delta), \\ \tilde{\mathbf{x}}_F &= \text{UPLAPFSOLVER}(\mathbf{b}_F - \mathbf{L}_1^{up}[F, C] \tilde{\mathbf{x}}_C), \end{aligned} \tag{4}$$

where $\delta \leq \frac{\epsilon}{1 + \|\mathbf{L}_1^{up}[C, F] \mathbf{L}_1^{up}[F, F]^\dagger\|_2}$. Then,

$$\|\mathbf{L}_1^{up} \tilde{\mathbf{x}} - \mathbf{b}\|_2 \leq \epsilon \|\mathbf{b}\|_2,$$

where $\tilde{\mathbf{x}} = \begin{pmatrix} \tilde{\mathbf{x}}_F \\ \tilde{\mathbf{x}}_C \end{pmatrix}$. Let $m_F = |F|$ and $m_C = |C|$, and let UPLAPFSOLVER have runtime $t_1(m_F)$ and SCHURSOLVER have runtime $t_2(m_C)$. Then, we can compute $\tilde{\mathbf{x}}$ in time $O(t_1(m_F) + t_2(m_C) + m_F + m_C)$.

4.2.1 Partitioning the Edges

Suggested by Lemma 4.7, we want to partition the edges of \mathcal{K} into $F \cup C$ so that both systems in $\mathbf{L}_1^{up}[F, F]$ and the Schur complement $\text{Sc}[\mathbf{L}_1^{up}]_C$ can be efficiently solved. One idea is to divide \mathcal{K} into “disjoint”, “small”, and “balanced” regions. We then let F be the set of the “interior” edges of these regions and C be the set of the “boundary” edges. Our idea of dividing \mathcal{K} is motivated by a

well-studied notion called *r-division*, where r is a parameter controlling the size of each small piece [Fed87]. An r -division of a graph G over n vertices is a subdivision of G 's vertices into $O(n/r)$ regions each of size $O(r)$. Each region has two types of vertices: interior vertices appearing in exactly one region and only adjacent to vertices in this region, and boundary vertices appearing in at least two regions. One usually wants an r -division with only a few boundary vertices.

We want an analog of r -division for the *edges* of \mathcal{K} so that the edges are divided into balanced regions with “disjoint” interiors and small boundaries. Choosing F to be the set of all the interior edges, we can write $\mathbf{L}_1^{\text{up}}[F, F]$ as a block diagonal matrix where each block has a small size. Then, we can run Nested Dissection for each diagonal block to speed up Gaussian elimination for $\mathbf{L}_1^{\text{up}}[F, F]$. To efficiently solve systems in $\text{Sc}[\mathbf{L}_1^{\text{up}}]_C$, the Schur complement onto the edges in C , we need the boundary to be well-shaped so that we can “sparsify” the Schur complement onto the boundary by the boundary itself and run the Preconditioned Conjugate Gradient. Similar to [KPSZ18], we define r -hollowings, which are *well-shaped* r -divisions for the edges of a 2-complex. We remark that our r -hollowing defined below has *more restrictions* on the shape of the hollowing compared to [KPSZ18].

Definition 4.8 (r -hollowing). Let \mathcal{K} be a stable 3-complex with n simplexes, and let $r = o(n)$ be a positive integer. We divide \mathcal{K} into $O(n/r)$ *regions*. Each region is a 3-complex that triangulates a three-ball, and the underlying topological space of the region has diameter ⁷ $O(r^{1/3})$. Each region has $O(r)$ simplexes and $O(r^{2/3})$ boundary simplexes. Only boundary simplexes can belong to more than one region. In addition, each region intersects with at most $O(1)$ other regions. If two regions intersect, their intersection is either a triangulated disc or a path on the boundary of the two regions. The union of all boundary simplexes of each region is referred to as an r -hollowing of \mathcal{K} , which uniquely determines the division of \mathcal{K} .

Let us consider a concrete example. There are n unit cubes centered at (i, j, k) where $i, j, k \in \{1, 2, \dots, n^{1/3}\}$, where we assume $n^{1/3}$ is an integer. Each unit cube is subdivided into $O(1)$ tetrahedrons. Let r be an integer such that $r^{1/3}$ and $(n/r)^{1/3}$ are integers. We divide the n cubes into n/r subsets: the (i, j, k) th subset contains all the cubes centered at $\{(i-1)r^{1/3} + l_i, (j-1)r^{1/3} + l_j, (k-1)r^{1/3} + l_k) : 1 \leq l_i, l_j, l_k \leq r^{1/3}\}$. The tetrahedrons in the cubes in the same subset form a region. The union of the boundary simplexes of the regions is an r -hollowing.

By Definition 4.8, since different regions can only share boundary simplexes, different intersection discs and paths can only share boundary edges. So, an r -hollowing is a union of discs such that only a boundary edge of a disc can be shared by more than one disc.

Let us examine why Definition 4.8 satisfies our requirements for a well-shaped r -division. We first show the interiors of different regions are “disjoint” in the sense that $\mathbf{L}_1^{\text{up}}[F, F]$ is a block diagonal matrix where each diagonal block corresponds to the interior of a region. We can write \mathbf{L}_1^{up} as the sum of rank-1 matrices that each corresponds to a triangle in \mathcal{K} :

$$\mathbf{L}_1^{\text{up}} = \partial_2 \mathbf{W}_2 \partial_2^\top = \sum_{\sigma: \text{triangle in } \mathcal{K}} \mathbf{W}_2[\sigma, \sigma] \cdot \partial_2[:, \sigma] \partial_2[:, \sigma]^\top. \quad (5)$$

For any two edges e_1, e_2 , $\mathbf{L}_1^{\text{up}}[e_1, e_2] = 0$ if and only if no triangle in \mathcal{K} contains both e_1, e_2 . We say such e_1 and e_2 are Δ -disjoint. The following claim shows that interior edges from different regions are Δ -disjoint.

Claim 4.9. *Let \mathcal{K} be a stable 3-complex and \mathcal{T} be an r -hollowing of \mathcal{K} . Let R_1, R_2 be two different regions of \mathcal{K} w.r.t. \mathcal{T} , and let e_1 be an interior edge in R_1 and e_2 an interior edge in R_2 . Then, e_1 and e_2 are Δ -disjoint.*

⁷The diameter of a 3-complex \mathcal{K} is the maximum distance between any pair of vertices in \mathcal{K} .

Proof. Assume by contradiction there exists a triangle $\sigma \in \mathcal{K}$ contains both e_1, e_2 . Let e_3 be the third edge in σ . Since e_1, e_2 are interior edges of different regions R_1, R_2 , e_3 must cross the boundary of R_1, R_2 . This contradicts the fact that regions can only intersect on their boundaries. Thus, e_1, e_2 must be \triangle -disjoint. \square

In addition, the following lemma shows that the boundaries of the regions well approximate the Schur complement onto the boundaries. The proof of Lemma 4.10 is slightly more involved, and we defer it to Section 7.1.

Lemma 4.10 (Spectral bounds for r -hollowing). *Let \mathcal{K} be a stable 3-complex. Let \mathcal{T} be an r -hollowing of \mathcal{K} , and let C be the edges of \mathcal{T} . Then,*

$$\mathbf{L}_{1,\mathcal{T}}^{up} \preceq \text{Sc}[\mathbf{L}_1^{up}]_C \preceq O(rU)\mathbf{L}_{1,\mathcal{T}}^{up},$$

where U is the ratio between the maximum and the minimum triangle weights in \mathcal{K} .

4.2.2 Proof of Lemma 4.6 for Up-Laplacian Solver

Algorithm 1 sketches a pseudo-code for our up-Laplacian solver.

Algorithm 1: UPLAPSOLVER($\mathcal{K}, \mathbf{b}, \epsilon$)

Input: A stable 3-complex \mathcal{K} with up-Laplacian \mathbf{L}_1^{up} , a vector $\mathbf{b} \in \text{Im}(\mathbf{L}_1^{up})$, an error parameter $\epsilon > 0$

Output: An approximate solution $\tilde{\mathbf{x}}$ such that $\|\mathbf{L}_1^{up}\tilde{\mathbf{x}} - \mathbf{b}\|_2 \leq \epsilon \|\mathbf{b}\|_2$

- 1 $r \leftarrow \lfloor n^{1/2} \rfloor$, where n is the number of simplexes in \mathcal{K} .
- 2 $\mathcal{T} \leftarrow \text{HOLLOWING}(\mathcal{K}, r)$, via Algorithm 2. \mathcal{T} divides \mathcal{K} into regions.
- 3 $F \leftarrow$ the interior edges of each region of \mathcal{K} and $C \leftarrow$ the boundary edges of each region.
- 4 UPLAPFSOLVER(\cdot) \leftarrow a solver by Nested Dissection that satisfies the requirement in Lemma 4.7.
- 5 SCHURSOLVER(\cdot, \cdot) \leftarrow a solver by Preconditioned Conjugate Gradient with preconditioner being the up-Laplacian of \mathcal{T} that satisfies the requirement in Lemma 4.7.
- 6 $\tilde{\mathbf{x}} \leftarrow$ computed by Equation (4)
- 7 **return** solution $\tilde{\mathbf{x}}$

By Lemma 4.7, the $\tilde{\mathbf{x}}$ returned by Algorithm 1 satisfies $\|\mathbf{L}_1^{up}\tilde{\mathbf{x}} - \mathbf{b}\|_2 \leq \epsilon \|\mathbf{b}\|_2$. To bound the runtime of Algorithm 1, we need the following lemmas for lines 2, 4, and 5.

Lemma 4.11 (Find an r -hollowing). *Given a stable 3-complex \mathcal{K} with n simplexes and a positive integer $r = o(n)$, we can find an r -hollowing of \mathcal{K} in linear time.*

Algorithm 2 describes how to find an r -hollowing in linear time. The algorithm first finds a bounding box of \mathcal{K} , then “cuts” the box into $O(n/r)$ smaller boxes of equal volume, and finally turns each smaller box into a region of an r -hollowing. We prove Lemma 4.11 in Section 5.

Lemma 4.12 (Solver for the “ F ” part). *Let \mathcal{K} be a stable 3-complex with n simplexes. Let \mathcal{T} be an r -hollowing of \mathcal{K} , and let F be the set of interior edges in each region of \mathcal{K} w.r.t. \mathcal{T} . Then, with a pre-processing time $O(nr)$, there exists a solver UPLAPFSOLVER(\cdot) such that given any $\mathbf{b}_F \in \text{im}(\mathbf{L}_1^{up}[F, F])$, UPLAPFSOLVER(\mathbf{b}_F) returns an \mathbf{x}_F such that $\mathbf{L}_1^{up}[F, F]\mathbf{x}_F = \mathbf{b}_F$ in time $O(nr^{1/3})$.*

By our choice of F , the matrix $\mathbf{L}_1^{\text{up}}[F, F]$ can be written as a block diagonal matrix where each block corresponds to a region of \mathcal{K} w.r.t. the r -hollowing \mathcal{T} . Since each region is a 3-complex in which every tetrahedron has an aspect ratio $O(1)$, we can construct the solver UPLAPFSOLVER by Nested Dissection [MT90]. However, since each row or column of $\mathbf{L}_1^{\text{up}}[F, F]$ corresponds to an edge in \mathcal{K} , we need to turn the good vertex separators in [MT90] into good edge separators for regions of \mathcal{K} . We prove Lemma 4.12 in Section 6.

Lemma 4.13 (Solver for the Schur complement). *Let \mathcal{K} be a stable 3-complex with n simplexes. Let \mathcal{T} be an r -hollowing of \mathcal{K} , and let C be the set of boundary edges of each region of \mathcal{K} w.r.t. \mathcal{T} . Then, there exists a solver SCHURSOLVER(\cdot, \cdot) such that for any $\mathbf{h} \in \text{Im}(\text{Sc}[\mathbf{L}_1^{\text{up}}]_C)$ and $\delta > 0$, SCHURSOLVER(\mathbf{h}, δ) returns an $\tilde{\mathbf{x}}_C$ such that $\|\text{Sc}[\mathbf{L}_1^{\text{up}}]_C \tilde{\mathbf{x}}_C - \mathbf{h}\|_2 \leq \delta \|\mathbf{h}\|_2$ in time $O(r^{1/2}U^{1/2} \log(rU/\delta) (n^2r^{-2} + nr^{1/3}) + n^2r^{-4/3} + n^3r^{-3} + nr)$, where U is the ratio between the maximum and the minimum triangle weights in \mathcal{K} .*

Our solver SCHURSOLVER is based on the Preconditioned Conjugate Gradient (PCG) with the preconditioner $\mathbf{L}_{1,\mathcal{T}}^{\text{up}}$, the up-Laplacian operator of \mathcal{T} . By Theorem 2.6 and Lemma 4.10, the number of PCG iterations is $O(\sqrt{rU} \log(rU/\epsilon))$. In each PCG iteration, we solve the system in $\mathbf{L}_{1,\mathcal{T}}^{\text{up}}$ via Lemma 4.7 again. Observe \mathcal{T} is a union of discs glued together by identifying certain sets of their boundary edges. We let F contain all the interior edges of discs and let C contain all the boundary edges. Each edge in F is shared by exactly two triangles. So, $\mathbf{L}_{1,\mathcal{T}}^{\text{up}}[F, F]$ is the down-Laplacian operator of the dual graph of \mathcal{T} restricted to interior simplexes, and a system in $\mathbf{L}_{1,\mathcal{T}}^{\text{up}}[F, F]$ can be solved efficiently via the linear-time down-Laplacian solvers. The size of C can be reduced to $O(n/r)$ at the beginning so that we can efficiently invert the Schur complement in time $O(n^3r^{-3})$. We prove Lemma 4.13 in Section 7.

Given the above lemmas, we prove Lemma 4.6.

Proof of Lemma 4.6. The correctness of Algorithm 1 is by Lemma 4.7. By Lemma 4.11, 4.12, and 4.13, the total runtime of the algorithm is

$$O\left(r^{1/2}U^{1/2} \log(rU/\epsilon) \left(n^2r^{-2} + nr^{1/3}\right) + n^2r^{-4/3} + n^3r^{-3} + nr\right),$$

where r is the parameter for the r -hollowing. We choose $r = \lfloor n^{1/2} \rfloor$. The runtime is

$$O\left(n^{3/2} + U^{1/2}n^{17/12} \log(nU/\epsilon)\right).$$

□

5 Computing an r -Hollowing

In this section, we describe a linear time algorithm (Algorithm 2) that finds an r -hollowing of a stable 3-complex \mathcal{K} , and we prove 4.11.

Let \mathbb{K} be the underlying topological space of \mathcal{K} . Algorithm 2 first finds a *nice bounding box*, which encompasses \mathbb{K} and whose volume and aspect ratio is within constant factors of those of \mathbb{K} . Lemma 5.2 provides a linear time algorithm for finding a nice bounding box for \mathcal{K} when the aspect ratio of the underlying topological space of \mathcal{K} is $O(1)$. Then, Algorithm 2 “cuts” the bounding box into $O(n/r)$ smaller boxes of equal volume using 2-dimensional planes and turns these cutting planes into an r -hollowing.

We need the following lemma from [BHP01] to construct a nice bounding box.

Lemma 5.1 (Lemma 3.4 of [BHP01]). *Given a set X of points in \mathbb{R}^3 , we can compute in linear time a bounding box \mathcal{B} with $\text{vol}(\mathcal{B}) \leq 6\sqrt{6}\text{vol}(\mathcal{B}^*)$, where $\text{vol}(\cdot)$ is the volume and \mathcal{B}^* is a bounding box of X with the minimum volume.*

Corollary 5.2 (Nice bounding box). *Let \mathcal{K} be a 3-complex embedded in \mathbb{R}^3 whose underlying topological space has aspect ratio $O(1)$. We can compute a nice bounding box of \mathcal{K} in linear time.*

Proof. Let \mathcal{B} be the bounding box computed by Lemma 5.1 with X being the set of points in the underlying topological space of \mathcal{K} . We will show that \mathcal{B} is a nice bounding box of \mathcal{K} . Let \mathcal{O}_1 be the minimum-volume ball containing \mathcal{K} , and let \mathcal{O}_2 be the maximum-volume ball contained in \mathcal{K} . Since \mathcal{K} has $O(1)$ aspect ratio, we know

$$\text{vol}(\mathcal{O}_2) \leq \text{vol}(\mathcal{K}) \leq \text{vol}(\mathcal{O}_1) = O(\text{vol}(\mathcal{O}_2)). \quad (6)$$

Let \mathcal{B}^* be a bounding box of \mathcal{K} with the minimum volume, and let \mathcal{B}_1 be a bounding box of \mathcal{O}_1 with the minimum volume. Since \mathcal{B}_1 contains \mathcal{K} and \mathcal{O}_1 is a Euclidean ball in \mathbb{R}^3 , we have

$$\text{vol}(\mathcal{B}^*) \leq \text{vol}(\mathcal{B}_1) = O(\text{vol}(\mathcal{O}_1)).$$

By Lemma 5.1 and Equation (6), we have

$$\text{vol}(\mathcal{B}) \leq 6\sqrt{6}\text{vol}(\mathcal{B}^*) = O(\text{vol}(\mathcal{O}_1)) = O(\text{vol}(\mathcal{O}_2)) = O(\text{vol}(\mathcal{K})).$$

The aspect ratio of the box \mathcal{B} is $O(1)$ times the ratio between the maximum and the minimum lengths of its edges. Since \mathcal{B} contains \mathcal{O}_2 and $\text{vol}(\mathcal{B}) = O(\text{vol}(\mathcal{O}_2))$, the aspect ratio of \mathcal{B} is $O(1)$. \square

Algorithm 2: HOLLOWING(\mathcal{K}, r)

Input: A stable 3-complex \mathcal{K} with n simplexes and a positive integer $r \ll n$

Output: An r -hollowing \mathcal{T}

- 1 Find a nice bounding box \mathcal{B} for \mathcal{K} by Corollary 5.2.
 - 2 $\mathcal{T} \leftarrow$ all the boundary simplexes of \mathcal{K} .
 - 3 For each pair of parallel faces of \mathcal{B} , find $\lfloor n^{1/3}r^{-1/3} \rfloor$ evenly-spaced 2-dimensional planes parallel to the face which divide \mathcal{B} into equal-volume pieces. We can slightly perturb the planes so that no plane intersects with any vertex of \mathcal{K} .
 - 4 **for** each 2-dimensional plane P **do**
 - 5 $\mathcal{Q} \leftarrow$ all the tetrahedrons of \mathcal{K} that intersect P .
 - 6 Let \mathbb{K} be the underlying space of \mathcal{K} and \mathbb{Q} be that of \mathcal{Q} . Then, $\mathbb{K} \setminus \mathbb{Q}$ has two disjoint spaces $\mathbb{K}_1, \mathbb{K}_2$.
 - 7 $\mathcal{T} \leftarrow \mathcal{T} \cup$ all the simplexes of \mathcal{K} on the boundary of \mathbb{K}_1 .
 - 8 **end**
 - 9 **return** \mathcal{T}
-

Proof of Lemma 4.11. We can check that Algorithm 2 has linear runtime. In the rest of the proof, we show \mathcal{T} returned by Algorithm 2 is an r -hollowing of \mathcal{K} .

Since each tetrahedron in \mathcal{K} has volume $\Theta(1)$, the volume of \mathcal{K} is $\Theta(n)$. By Lemma 5.2, we have $\text{vol}(\mathcal{B}) = \Theta(\mathcal{K}) = \Theta(n)$. In Algorithm 2, the 2-dimensional planes divide the box \mathcal{B} into $O(n/r)$ smaller boxes each of volume $O(r)$ and surface area $O(r^{2/3})$. By our construction of \mathcal{T} , each smaller box corresponds to a region, which triangulates a three-ball. For each 2-dimensional plane P , let

T_P be the newly added triangles to \mathcal{T} related to P on line 7 of Algorithm 2. Since each triangle in T_P has $O(1)$ Euclidean distance to the plane P and \mathcal{K} is convex, there are $O(n/r)$ regions, and each region of \mathcal{T} has $O(r)$ simplexes and $O(r^{2/3})$ boundary simplexes. Additionally, the underlying topological space of each region has diameter $O(r^{1/3})$.

Consider a box containing a region R , and the box's volume is 1.1 times the volume of R . All the regions intersecting with R must intersect with this box. Since only $O(1)$ regions can intersect with the box, R can only intersect with $O(1)$ regions.

By our construction, two regions R_1, R_2 have a non-empty intersection if and only if both of their boundaries have a non-empty intersection with a common T_P and R_1 and R_2 are on different sides of T_P . The intersection of R_i and T_P is a triangulated disc or a path for $i \in \{1, 2\}$, and thus the intersection of R_1, R_2 and T_P is a triangulated disc \mathcal{D} or a path. Since T_P separates R_1 and R_2 , there is no other $T_{P'}$ that intersects with both R_1 and R_2 and the triangulated disc \mathcal{D} or the path is exactly the intersection of R_1 and R_2 .

So, \mathcal{T} satisfies all the conditions in Definition 4.8 and is an r -hollowing of \mathcal{K} . \square

6 Solver for $\mathbf{L}_1^{\text{up}}[F, F]$

In this section, we build an efficient solver $\text{UPLAPFSOLVER}(\mathbf{b})$ that returns an \mathbf{x} such that $\mathbf{L}_1^{\text{up}}[F, F]\mathbf{x} = \mathbf{b}$ for any $\mathbf{b} \in \text{Im}(\mathbf{L}_1^{\text{up}}[F, F])$ and prove Lemma 4.12. Since any two interior edges from different regions are \triangle -disjoint (Claim 4.9), we can write $\mathbf{L}_1^{\text{up}}[F, F]$ as a block diagonal matrix where each block diagonal corresponds to a region. Then, computing an \mathbf{x} such that $\mathbf{L}_1^{\text{up}}[F, F]\mathbf{x} = \mathbf{b}$ reduces to computing a solution for each diagonal block submatrix per region. For this reason, Lemma 4.12 is a corollary of the following Lemma 6.1.

Lemma 6.1. *Let \mathcal{X} be a 3-complex with $O(r)$ simplexes embedded in \mathbb{R}^3 such that (1) each tetrahedron of \mathcal{X} has $O(1)$ aspect ratio and (2) \mathcal{X} has $O(r^{2/3})$ boundary simplexes. Let F be the set of interior edges of \mathcal{X} , and let $\mathbf{L}_{1,\mathcal{X}}^{\text{up}}$ be the up-Laplacian of \mathcal{X} and $\mathbf{M} \stackrel{\text{def}}{=} \mathbf{L}_{1,\mathcal{X}}^{\text{up}}[F, F]$. Then, there is a permutation matrix \mathbf{P} and a lower triangular matrix \mathbf{L} with $O(r^{4/3})$ nonzeros such that*

$$\mathbf{M} = \mathbf{P}\mathbf{L}\mathbf{L}^\top\mathbf{P}^\top. \quad (7)$$

In addition, we can find such \mathbf{P} and \mathbf{L} in time $O(r^2)$. Given the above factorization, for any $\mathbf{b} \in \text{Im}(\mathbf{M})$, we can compute an \mathbf{x} such that $\mathbf{M}\mathbf{x} = \mathbf{b}$ in $O(r^{4/3})$ time.

The factorization in Equation (7) is called Cholesky factorization. We will utilize the geometric structures of \mathcal{X} to find a sparse Cholesky factorization efficiently. Miller and Thurston [MT90] studied vertex separators of the 1-skeleton of a 3-complex \mathcal{X} in which each tetrahedron has $O(1)$ aspect ratio. A subset of vertices S of a graph over v vertices δ -separates if the remaining vertices can be partitioned into two sets A, B such that there are no edges between A and B , and $|A|, |B| \leq \delta v$. The set S is an $f(v)$ -separator if there exists a constant $\delta < 1$ such that S δ -separates and $|S| \leq f(v)$. These separators can be incorporated with Nested Dissection to efficiently compute a sparse Cholesky factorization of a matrix in which the nonzero structure encodes the 1-skeleton of \mathcal{X} .

Theorem 6.2 (Vertex separator for a 3-complex, Theorem 1.5 of [MT90]). *Let \mathcal{X} be a 3-complex in \mathbb{R}^3 in which each tetrahedron has $O(1)$ aspect ratio, and suppose \mathcal{X} has t tetrahedrons and \bar{v} boundary vertices. Then, the 1-skeleton of \mathcal{X} has a $O(t^{2/3} + \bar{v})$ -separator that $4/5$ -separates \mathcal{X} . In addition, a randomized algorithm can find such a separator in linear time.*

We need a slightly modified version of Theorem 6.2 to apply to our matrix \mathbf{M} .

Corollary 6.3 (Edge separator for a 3-complex). *Let \mathcal{X} be a 3-complex that satisfies the requirements in Theorem 6.2. In addition, the 1-skeleton of \mathcal{X} is connected. Let m be the number of edges in \mathcal{X} . Then, there exists a randomized algorithm that removes $O(t^{2/3} + \bar{v})$ edges in linear time so that the remaining edges can be partitioned into two Δ -disjoint sets each of size at most cm for some constant $c < 1$.*

Proof. Let $v = O(r)$ be the number of vertices in \mathcal{X} . By Lemma 2.7, $v = \Theta(m)$. Let S be the set of $O(t^{2/3} + \bar{v})$ vertices that 4/5-separates \mathcal{X} (by Theorem 6.2), and let A, B be two disjoint sets of the remaining vertices after removing S such that $|A|, |B| \leq 4v/5$. Let E be the set of edges in \mathcal{X} incident to some vertex in S . Since each vertex has $O(1)$ degree by Lemma 2.7, $|E| = O(|S|) = O(t^{2/3} + \bar{v})$. Now, we remove edges in E from the edges of \mathcal{X} . Let E_A be the set of the remaining edges incident to a vertex in A , and let E_B be the set of the remaining edges incident to a vertex in B . Since A, B are disjoint, E_A and E_B are Δ -disjoint. We then show that $|E_A|, |E_B| \leq cm$ for some constant $c < 1$. Let E'_A be the set of edges in \mathcal{X} that are incident to some vertex in A . Since the 1-skeleton of \mathcal{X} is a connected graph, $|E'_A| \geq |A| \geq (1 - o(1))\frac{v}{5}$. Besides, $|E'_A| = |E_A| + |E''_A|$, where E''_A contains edges with one endpoint in A and the other in S . Since $|E''_A| = O(t^{2/3} + \bar{v})$, we know $|E_A| \geq (1 - o(1))\frac{v}{5} > c'm$ for some constant c' . Since E_A and E_B are disjoint, we know $|E_B| \leq m - |E_A| \leq (1 - c')m$. By symmetry, we have $|E_A| \leq (1 - c')m$. \square

We need the following theorem about Nested Dissection from [LRT79].

Theorem 6.4 (Nested Dissection, Theorem 6 of [LRT79]). *Let \mathcal{G} be any class of graphs closed under subgraph on which an v^α -separator exists for $\alpha > 1/2$. Let $\mathbf{A} \in \mathbb{R}^{v \times v}$ be symmetric and positive definite (that is, all eigenvalues of \mathbf{A} are positive). Let G_A be a graph over vertices $\{1, \dots, v\}$ where vertices i, j are connected if and only if $\mathbf{A}[i, j] \neq 0$. If $G_A \in \mathcal{G}$, then we can find a permutation matrix \mathbf{P} and a lower triangular matrix \mathbf{L} with $O(v^{2\alpha})$ nonzeros in $O(v^{3\alpha})$ time such that $\mathbf{A} = \mathbf{P}\mathbf{L}\mathbf{L}^\top\mathbf{P}^\top$.*

Combining Corollary 6.3 and Theorem 6.4, we prove Lemma 6.1.

Proof of Lemma 6.1. Our approach for finding a sparse Cholesky factorization is the same as Section 4 of [MT90]. Specifically, we find a family of separators for \mathcal{X} by recursively applying Corollary 6.3 to the remaining sets of edges A, B . Since the separator size grows as a function of the boundary vertices, at the top level of the recursion, we include all the boundary edges $O(r^{2/3})$ in the root separator, which only increases the root separator by a constant factor. In each remaining recursion step, suppose we want to separate \mathcal{Y} , a sub-complex of \mathcal{X} ; we let $\bar{\mathcal{Y}}$ be the complex consisting \mathcal{Y} and all the simplexes in \mathcal{X} that contain a vertex in \mathcal{Y} . Then we apply a slightly modified version of Corollary 6.3 (obtained by a slightly modified Theorem 6.2) to $\bar{\mathcal{Y}}$ in which we only separate \mathcal{Y} . Here, the boundary-vertex term $O(\bar{v})$ in the size of the separator can be ignored since all the boundary edges of $\bar{\mathcal{Y}}$ have already been included in upper-level separators; we also use the fact that the number of boundary vertices and the number of boundary edges are equal up to a constant factor. This separator family provides an elimination ordering for the edges of \mathcal{K} , which is the permutation matrix \mathbf{P} in Equation (7), and the ordering uniquely determines the matrix \mathbf{L} . By Theorem 6.4, \mathbf{L} has $O(r^{4/3})$ nonzeros, and \mathbf{P}, \mathbf{L} can be found in time $O(r^2)$.

One issue left is that \mathbf{M} in Lemma 6.1 is positive semidefinite but *not* positive definite. During the process of numeric factorization, the first row and column of some Schur complements are all-zero. We simply ignore these zeros and proceed (Ref: Chapter 4.2.8 of [VLG96]). This produces a Cholesky factorization of $\mathbf{P}^\top \mathbf{M} \mathbf{P} = \mathbf{L}\mathbf{L}^\top$ such that only k columns of \mathbf{L} are nonzero, where k is

the rank of \mathbf{M} . We permute the rows and the columns of \mathbf{L} by multiplying permutation matrices $\mathbf{P}_1, \mathbf{P}_2$ so that

$$\mathbf{P}_1 \mathbf{L} \mathbf{P}_2 = \begin{pmatrix} \mathbf{T} & \mathbf{0} \end{pmatrix} \stackrel{\text{def}}{=} \begin{pmatrix} \mathbf{H}_1 & \mathbf{0} \\ \mathbf{H}_2 & \mathbf{0} \end{pmatrix},$$

where \mathbf{H}_1 is a $k \times k$ non-singular lower-triangular matrix. Then, solving $\mathbf{M}\mathbf{x} = \mathbf{b}$ is equivalent to solving

$$\mathbf{T} \mathbf{T}^\top \mathbf{z} = \mathbf{P}_1 \mathbf{P}^\top \mathbf{b} \stackrel{\text{def}}{=} \mathbf{f}, \quad \mathbf{P}_1 \mathbf{P}^\top \mathbf{x} = \mathbf{z}.$$

We solve the first system by solving $\mathbf{T}\mathbf{y} = \mathbf{f}$ and $\mathbf{T}^\top \mathbf{z} = \mathbf{y}$. We let \mathbf{y} satisfy $\mathbf{H}_1 \mathbf{y} = \mathbf{f}[1:k]$. Since \mathbf{H}_1 has full rank and $O(r^{4/3})$ nonzeros, such a \mathbf{y} exists and can be found in $O(r^{4/3})$ time. Since $\mathbf{y} \in \text{Im}(\mathbf{T})$, we know $\mathbf{T}\mathbf{y} = \mathbf{f}$. Then we let $\mathbf{z}[k+1:v] = \mathbf{0}$, where v is the number of vertices in \mathcal{X} , and we let $\mathbf{H}_1^\top \mathbf{z}[1:k] = \mathbf{y}$. Again, such \mathbf{z} exists and can be found in $O(r^{4/3})$ time. Finally, we compute $\mathbf{x} = \mathbf{P} \mathbf{P}_1^\top \mathbf{z}$ in linear time. \square

Proof of Lemma 4.12. We apply Lemma 6.1 to each diagonal block matrix of $\mathbf{L}_1^{\text{up}}[F, F]$ to compute a sparse Cholesky factorization as in Equation (7) in time

$$O\left(\frac{n}{r} \cdot r^2\right) = O(nr).$$

Given this Cholesky factorization, we can solve a system in $\mathbf{L}_1^{\text{up}}[F, F]$ in time

$$O\left(\frac{n}{r} \cdot r^{4/3}\right) = O(nr^{1/3}).$$

\square

7 Solver for the Schur Complement

In this section, we establish a fast approximate solver for $\text{Sc}[\mathbf{L}_1^{\text{up}}]_C$ and prove Lemma 4.13. Recall that C contains all the edges in \mathcal{T} (an r -hollowing of \mathcal{K}), and $\text{Sc}[\mathbf{L}_1^{\text{up}}]_C$ is the Schur complement of the up-Laplacian operator of \mathcal{K} onto C . The idea is to run the Preconditioned Conjugate Gradient (PCG) for systems in $\text{Sc}[\mathbf{L}_1^{\text{up}}]_C$ with preconditioner $\mathbf{L}_{1,\mathcal{T}}^{\text{up}}$, which is the first up-Laplacian operator of \mathcal{T} . By Theorem 2.6, the number of PCG iterations is $O(\sqrt{\kappa} \log(\kappa/\epsilon))$ where $\kappa = \kappa(\text{Sc}[\mathbf{L}_1^{\text{up}}]_C, \mathbf{L}_{1,\mathcal{T}}^{\text{up}})$ is the relative condition number and ϵ is the error parameter; in each PCG iteration, we need to solve a system in $\mathbf{L}_{1,\mathcal{T}}^{\text{up}}$, multiply $\text{Sc}[\mathbf{L}_1^{\text{up}}]_C$ with $O(1)$ vectors, and implement $O(1)$ vector operations. In Section 7.1, we upper bound the relative condition number. In Section 7.2, we show how to efficiently solve a system in $\mathbf{L}_{1,\mathcal{T}}^{\text{up}}$.

We will need the following observation.

Claim 7.1. *Let \mathcal{X} be a simplicial complex. Changing the orientations of the triangles in \mathcal{X} does not change its first up-Laplacian operator.*

Proof. Let $\partial_{2,\mathcal{X}}$ be the second boundary operator of \mathcal{X} , $\mathbf{W}_{2,\mathcal{X}}$ the diagonal matrix for the triangle weights, and $\mathbf{L}_{1,\mathcal{X}}^{\text{up}}$ the first up-Laplacian. Changing the orientations of the triangles in \mathcal{X} corresponds to multiplying a ± 1 diagonal matrix \mathbf{X} to the right of $\partial_{2,\mathcal{X}}$. Observe

$$\partial_{2,\mathcal{X}}(\mathbf{X} \mathbf{W}_{2,\mathcal{X}} \mathbf{X}) \partial_{2,\mathcal{X}}^\top = \partial_{2,\mathcal{X}} \mathbf{W}_{2,\mathcal{X}} \partial_{2,\mathcal{X}}^\top = \mathbf{L}_{1,\mathcal{X}}^{\text{up}}.$$

Thus the statement holds. \square

7.1 Preconditioning the Schur Complement

We will prove Lemma 4.10, which upper bounds the relative condition number of $\text{Sc}[\mathbf{L}_1^{\text{up}}]_C$ and $\mathbf{L}_{1,\mathcal{T}}^{\text{up}}$ by $O(rU)$, where U is the ratio between the maximum and the minimum triangles weights in \mathcal{K} . We will need the following well-known fact about Schur complements. One can find its proof in Fact 4.6 of [KZ20].

Fact 7.2. *Let \mathbf{A} be a symmetric and PSD matrix:*

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}[F, F] & \mathbf{A}[F, C] \\ \mathbf{A}[C, F] & \mathbf{A}[C, C] \end{pmatrix}.$$

Let $\text{Sc}[\mathbf{A}]_C = \mathbf{A}[C, C] - \mathbf{A}[C, F]\mathbf{A}[F, F]^\dagger\mathbf{A}[F, C]$ be the Schur complement of \mathbf{A} onto C . Then, for any $\mathbf{x} \in \mathbb{R}^{|C|}$,

$$\min_{\mathbf{y} \in \mathbb{R}^{|F|}} \begin{pmatrix} \mathbf{y}^\top & \mathbf{x}^\top \end{pmatrix} \mathbf{A} \begin{pmatrix} \mathbf{y} \\ \mathbf{x} \end{pmatrix} = \mathbf{x}^\top \text{Sc}[\mathbf{A}]_C \mathbf{x}.$$

As a corollary, $\text{Sc}[\mathbf{A}]_C$ is symmetric and PSD.

The first inequality in Lemma 4.10 follows immediately from Fact 7.2, and we restate it as the following claim

Claim 7.3. $\mathbf{L}_{1,\mathcal{T}}^{\text{up}} \preceq \text{Sc}[\mathbf{L}_1^{\text{up}}]_C$.

Proof. Let \mathcal{K}' be the 2-complex that contains all the interior triangles in each region of \mathcal{K} w.r.t. \mathcal{T} and their subsimplexes, and let $\mathbf{L}_{1,\mathcal{K}'}^{\text{up}}$ be the up-Laplacian operator of \mathcal{K}' . We extend $\mathbf{L}_{1,\mathcal{K}'}^{\text{up}}$ and $\mathbf{L}_{1,\mathcal{T}}^{\text{up}}$ to larger matrices $\widehat{\mathbf{L}}_{1,\mathcal{K}'}^{\text{up}}$ and $\widehat{\mathbf{L}}_{1,\mathcal{T}}^{\text{up}}$ in dimensions same as the up-Laplacian of \mathcal{K} by putting zeros to the entries for edges not in \mathcal{K}' and \mathcal{T} , respectively. Then,

$$\mathbf{L}_1^{\text{up}} = \widehat{\mathbf{L}}_{1,\mathcal{K}'}^{\text{up}} + \widehat{\mathbf{L}}_{1,\mathcal{T}}^{\text{up}}.$$

Since all the edges of \mathcal{T} are in C , we know

$$\text{Sc}[\mathbf{L}_1^{\text{up}}]_C = \text{Sc}[\widehat{\mathbf{L}}_{1,\mathcal{K}'}^{\text{up}}]_C + \mathbf{L}_{1,\mathcal{T}}^{\text{up}} \succcurlyeq \mathbf{L}_{1,\mathcal{T}}^{\text{up}},$$

where the inequality holds due to Fact 7.2. □

We prove the second inequality in Lemma 4.10 in the rest of the section. Since each region of \mathcal{K} intersects with $O(1)$ other regions, we can first prove a similar inequality for a single region and then generalize it to \mathcal{K} , where we lose at most a constant factor in the second step.

Lemma 7.4. *Let \mathcal{X} be a region of \mathcal{K} w.r.t. to an r -division. Let \mathcal{B} be the 2-complex containing all the boundary simplexes of \mathcal{X} , let B be the set of edges in \mathcal{B} , and let $\mathbf{L}_{1,\mathcal{B}}^{\text{up}}$ be the first up-Laplacian operator of \mathcal{B} . Then,*

$$\text{Sc}[\mathbf{L}_{1,\mathcal{X}}^{\text{up}}]_B \preceq O(rU) \mathbf{L}_{1,\mathcal{B}}^{\text{up}},$$

where U is the ratio between the maximum and the minimum triangle weights in \mathcal{X} .

Lemma 4.10 follows immediately from Lemma 7.4.

Proof of Lemma 4.10. For each i , let \mathcal{X}_i be the 3-complex for the i th region of \mathcal{K} , and let \mathcal{B}_i be 2-complex that contains all the boundary simplexes of \mathcal{X}_i , and let B_i be the set of edges in \mathcal{B}_i . Then C is the union of B_i 's. We extend each $\mathbf{L}_{1,\mathcal{X}_i}^{\text{up}}$ to a larger matrix $\widehat{\mathbf{L}_{1,\mathcal{X}_i}^{\text{up}}}$ in dimensions same as \mathbf{L}_1^{up} by putting zeros for the entries not in \mathcal{X}_i . Similarly, we extend each $\mathbf{L}_{1,\mathcal{B}_i}^{\text{up}}$ to a larger matrix $\widehat{\mathbf{L}_{1,\mathcal{B}_i}^{\text{up}}}$ in dimensions same as $\mathbf{L}_{1,\mathcal{T}}^{\text{up}}$. Then,

$$\text{Sc}[\mathbf{L}_1^{\text{up}}]_C \preceq \sum_i \text{Sc} \left[\widehat{\mathbf{L}_{1,\mathcal{X}_i}^{\text{up}}} \right]_{B_i} \preceq O(rU) \sum_i \widehat{\mathbf{L}_{1,\mathcal{B}_i}^{\text{up}}} \preceq O(rU) \mathbf{L}_{1,\mathcal{T}}^{\text{up}},$$

where the second inequality is due to Lemma 7.4 and the third since each region intersects with $O(1)$ other regions (so that each boundary edge can be shared by at most $O(1)$ regions). \square

It remains to prove Lemma 7.4. We first show that the images of $\text{Sc}[\mathbf{L}_{1,\mathcal{X}}^{\text{up}}]_C$ and $\mathbf{L}_{1,\mathcal{B}}^{\text{up}}$ are equal.

Claim 7.5. $\text{Im}(\text{Sc}[\mathbf{L}_{1,\mathcal{X}}^{\text{up}}]_B) = \text{Im}(\mathbf{L}_{1,\mathcal{B}}^{\text{up}})$.

Proof. In the proof, we drop the subscript \mathcal{X} to simplify our notations by letting $\mathbf{L}_1^{\text{up}} = \mathbf{L}_{1,\mathcal{X}}^{\text{up}}$ and $\partial_2 = \partial_{2,\mathcal{X}}$. We let A be the set of interior edges in \mathcal{X} , and let V_A be the set of interior vertices in \mathcal{X} and V_B the set of boundary vertices.

Since both the two matrices $\text{Sc}[\mathbf{L}_1^{\text{up}}]_B, \mathbf{L}_{1,\mathcal{B}}^{\text{up}}$ are symmetric and PSD, the statement in the claim is equivalent to $\text{Ker}(\text{Sc}[\mathbf{L}_1^{\text{up}}]_B) = \text{Ker}(\mathbf{L}_{1,\mathcal{B}}^{\text{up}})$. By Claim 7.3 with $\mathcal{K} = \mathcal{X}$ and $\mathcal{T} = \mathcal{B}$, we know

$$\mathbf{L}_{1,\mathcal{B}}^{\text{up}} \preceq \text{Sc}[\mathbf{L}_1^{\text{up}}]_B \implies \text{Ker}(\text{Sc}[\mathbf{L}_1^{\text{up}}]_B) \subseteq \text{Ker}(\mathbf{L}_{1,\mathcal{B}}^{\text{up}}).$$

It remains to show $\text{Ker}(\text{Sc}[\mathbf{L}_1^{\text{up}}]_B) \supseteq \text{Ker}(\mathbf{L}_{1,\mathcal{B}}^{\text{up}})$.

Let \mathbf{x}_B be an arbitrary vector in $\text{Ker}(\mathbf{L}_{1,\mathcal{B}}^{\text{up}})$. We want to show $\mathbf{x}_B \in \text{Ker}(\text{Sc}[\mathbf{L}_1^{\text{up}}]_B)$, that is, $\mathbf{x}_B^\top \text{Sc}[\mathbf{L}_1^{\text{up}}]_B \mathbf{x}_B = 0$. By Fact 7.2, it suffices to show that there exists an $\mathbf{x} = \begin{pmatrix} \mathbf{x}_A \\ \mathbf{x}_B \end{pmatrix}$ such that $\mathbf{x}^\top \mathbf{L}_1^{\text{up}} \mathbf{x} = 0$. This is equivalent to

$$\mathbf{x} \in \text{Ker}(\partial_2^\top) \iff \mathbf{x} \perp \text{Im}(\partial_2) \iff \mathbf{x} \in \text{Im}(\partial_1^\top), \quad (8)$$

where the second “if and only if” holds since the first Betti number of \mathcal{X} is 0. We can write

$$\partial_1^\top = \begin{pmatrix} \partial_1^\top[A, V_A] & \partial_1^\top[A, V_B] \\ \mathbf{0} & \partial_1^\top[B, V_B] \end{pmatrix},$$

where $\partial_1[V_B, B] = \partial_{1,\mathcal{B}}$. Since $\mathbf{x}_B \in \text{Ker}(\partial_{2,\mathcal{B}}^\top)$ and the first Betti number of \mathcal{B} is 0, by an argument similar to Equation (8), we have $\mathbf{x}_B \in \text{Im}(\partial_{1,\mathcal{B}}^\top)$, that is, $\mathbf{x}_B = \partial_{1,\mathcal{B}}^\top \mathbf{y}_B$ for some \mathbf{y}_B . Setting $\mathbf{x}_A = \partial_1^\top[A, V_B] \mathbf{y}_B$, we have $\mathbf{x} \in \text{Im}(\partial_1^\top)$. \square

Claim 7.6. $\lambda_{\max}(\text{Sc}[\mathbf{L}_{1,\mathcal{X}}^{\text{up}}]_B) = O(w_{\max})$, where w_{\max} is the maximum triangle weight in \mathcal{X} .

Proof. In the proof, we drop the subscript \mathcal{X} to simplify our notations. The Courant-Fischer Minimax Theorem (Ref: Theorem 8.1.2 of [VLG96]) states that for any symmetric matrix \mathbf{A} ,

$$\lambda_{\max}(\mathbf{A}) = \max_{\mathbf{x}: \|\mathbf{x}\|_2=1} \mathbf{x}^\top \mathbf{A} \mathbf{x}.$$

Apply this theorem to $\text{Sc}[\mathbf{L}_1^{\text{up}}]_B$ and Fact 7.2,

$$\lambda_{\max}(\text{Sc}[\mathbf{L}_1^{\text{up}}]_B) = \max_{\mathbf{x}: \|\mathbf{x}\|_2=1} \mathbf{x}^\top \text{Sc}[\mathbf{L}_1^{\text{up}}]_B \mathbf{x} \leq \max_{\mathbf{x}: \|\mathbf{x}\|_2=1} \begin{pmatrix} \mathbf{0}^\top & \mathbf{x}^\top \end{pmatrix} \mathbf{L}_1^{\text{up}} \begin{pmatrix} \mathbf{0} \\ \mathbf{x} \end{pmatrix} \leq \lambda_{\max}(\mathbf{L}_1^{\text{up}}).$$

Below, we bound $\lambda_{\max}(\mathbf{L}_1^{\text{up}})$:

$$\begin{aligned} \lambda_{\max}(\mathbf{L}_1^{\text{up}}) &= \max_{\mathbf{x}: \|\mathbf{x}\|_2=1} \mathbf{x}^\top \mathbf{L}_1^{\text{up}} \mathbf{x} = \max_{\mathbf{x}: \|\mathbf{x}\|_2=1} \sum_{\sigma: \text{triangle in } \mathcal{X}} \mathbf{W}_2[\sigma, \sigma] (\partial_2[:, \sigma]^\top \mathbf{x})^2 \\ &\leq 3w_{\max} \cdot \max_{\mathbf{x}: \|\mathbf{x}\|_2=1} \sum_{\substack{\sigma=[i,j,k]: \\ \text{triangle in } \mathcal{X}}} (\mathbf{x}[i]^2 + \mathbf{x}[j]^2 + \mathbf{x}[k]^2) = O(w_{\max}). \end{aligned}$$

The last inequality holds since each edge appears in at most $O(1)$ triangles by Lemma 2.7. Thus, $\lambda_{\max}(\text{Sc}[\mathbf{L}_1^{\text{up}}]_B) = O(w_{\max})$. \square

One more piece we need is a lower bound for $\lambda_{\min}(\mathbf{L}_{1,\mathcal{B}}^{\text{up}})$, which will be established via graph Laplacian matrices.

Theorem 7.7 (Section 4.2 of [Moh91]). *Let G be an unweighted graph over n vertices with diameter D – the length of the longest path in G . Let \mathbf{L}_G be the graph Laplacian matrix of G . Then,*

$$\lambda_{\min}(\mathbf{L}_G) \geq \frac{4}{nD}.$$

Claim 7.8. $\lambda_{\min}(\mathbf{L}_{1,\mathcal{B}}^{\text{up}}) = \Omega(\frac{w_{\min}}{r})$, where w_{\min} is the minimum triangle weight in \mathcal{B} .

Proof. In the proof, we drop the subscript \mathcal{B} to simplify our notations. Again we decompose the matrix as a sum of rank-1 matrices for each triangle in \mathcal{B} :

$$\mathbf{L}_1^{\text{up}} = \sum_{\sigma: \text{triangle in } \mathcal{B}} \mathbf{W}_2[\sigma, \sigma] \partial_2[:, \sigma] \partial_2[:, \sigma]^\top \succeq w_{\min} \cdot \partial_2 \partial_2^\top.$$

Since $\text{Im}(\mathbf{L}_1^{\text{up}}) = \text{Im}(\partial_2 \partial_2^\top)$, we have

$$\lambda_{\min}(\mathbf{L}_1^{\text{up}}) \geq w_{\min} \cdot \lambda_{\min}(\partial_2 \partial_2^\top) = w_{\min} \cdot \lambda_{\min}(\partial_2^\top \partial_2).$$

Since \mathcal{B} is the boundary of a stable 3-complex, \mathcal{B} triangulates a two-sphere $\{x \in \mathbb{R}^3 : \|\mathbf{x}\|_2 = 1\}$ and each edge of \mathcal{B} appears in exactly two triangles. Since changing the orientations of the triangles in \mathcal{B} does not change its up-Laplacian operator (by Claim 7.1), we assume all the triangles in \mathcal{B} are oriented clockwise. Then, each column of ∂_2^\top has exactly one entry with value 1 and one entry -1 and all others 0. That is, $\partial_2^\top \partial_2$ is the Laplacian of the dual graph of \mathcal{B} : the vertices are the triangles in \mathcal{B} , and two vertices are adjacent if and only if the corresponding two triangles share a common edge. The dual graph has $O(r^{2/3})$ vertices and diameter $O(r^{1/3})$. By Theorem 7.7, $\lambda_{\min}(\partial_2^\top \partial_2) = \Omega(w_{\min} \cdot r^{-1})$. \square

Combining all the claims above, we prove Lemma 7.4.

Proof of Lemma 7.4. The first inequality follows from Claim 7.3. For the second inequality, let $\mathbf{\Pi}$ be the orthogonal projection matrix onto $\text{Im}(\mathbf{L}_{1,\mathcal{T}}^{\text{up}}) = \text{Im}(\text{Sc}[\mathbf{L}_{1,\mathcal{X}}^{\text{up}}]_B)$ (by Claim 7.5). By Claim 7.6, $\text{Sc}[\mathbf{L}_{1,\mathcal{X}}^{\text{up}}]_B \preceq O(w_{\max})\mathbf{\Pi}$. By Claim 7.8, $\mathbf{\Pi} \preceq O(\frac{r}{w_{\min}})\mathbf{L}_{1,\mathcal{B}}^{\text{up}}$. Combining all together, we have $\text{Sc}[\mathbf{L}_{1,\mathcal{X}}^{\text{up}}]_B \preceq O(rU)\mathbf{L}_{1,\mathcal{B}}^{\text{up}}$. \square

7.2 Solver for the Preconditioner

We develop a fast solver for the preconditioner $\mathbf{L}_{1,\mathcal{T}}^{\text{up}}$, the up-Laplacian operator for \mathcal{T} (an r -hollowing of \mathcal{K}).

Lemma 7.9 (Solver for the preconditioner). *Let \mathcal{K} be a stable 3-complex with n simplexes. Let \mathcal{T} be an r -hollowing of \mathcal{K} . With a pre-processing in time $O(n^2 r^{-4/3} + n^3 r^{-3})$, given any vector $\mathbf{b} \in \text{Im}(\mathbf{L}_1^{\text{up}})$, we can find an \mathbf{x} such that $\mathbf{L}_{1,\mathcal{T}}^{\text{up}} \mathbf{x} = \mathbf{b}$ in time $O(nr^{-1/3} + n^2 r^{-2})$.*

We describe our high-level idea for solving a system in $\mathbf{L}_{1,\mathcal{T}}^{\text{up}}$. By our definition of r -hollowing, \mathcal{T} is a union of discs where two discs only share boundary edges. We observe the first up-Laplacian operator of a disc can be (approximately) treated as a first down-Laplacian operator. Thus a linear system can be solved in linear time via the down-Laplacian solver in Section 4.1. So, we apply Lemma 4.7 to $\mathbf{L}_{1,\mathcal{T}}^{\text{up}}$ with F being all the interior edges of the discs and C the rest of the edges. We solve systems in $\mathbf{L}_{1,\mathcal{T}}^{\text{up}}[F, F]$ in linear time by the down-Laplacian solver, and we solve a system in $\text{Sc}[\mathbf{L}_{1,\mathcal{T}}^{\text{up}}]_C$ by directly inverting the matrix. To further reduce the runtime of the second step, we observe that most of the boundary edges of the discs are linearly dependent, whose removal from $\mathbf{L}_{1,\mathcal{T}}^{\text{up}}$ does not affect solving systems in $\mathbf{L}_{1,\mathcal{T}}^{\text{up}}$.

7.2.1 Reducing to a Smaller System

We partition the edges in \mathcal{T} into $E_1 \cup E_2$, where E_1 contains all the interior edges of the discs and E_2 the boundary edges. For each $e \in E_2$, let D_e be the set of discs that contain e .

Claim 7.10. *Let R be an arbitrary region of \mathcal{K} . There are $O(1)$ distinct sets D_e 's among all the boundary edges e in R .*

Proof. By our definition of r -hollowing in Definition 4.8, R intersects with $O(1)$ regions; each intersection is a disc on the boundary of R , and two discs only intersect on their boundaries. So, the boundary of R is divided into $O(1)$ discs. This also holds for any other region. Let $\mathcal{D}_1, \dots, \mathcal{D}_l$, where $l = O(1)$, be all the discs belonging to the regions that intersect with R . For any boundary edge e of R , D_e is a subset of $\{\mathcal{D}_1, \dots, \mathcal{D}_l\}$. Since $l = O(1)$, there are $O(1)$ subsets of $\{\mathcal{D}_1, \dots, \mathcal{D}_l\}$. \square

Let $\mathbf{B}_1 = \partial_{2,\mathcal{T}}[E_1, :]$ and $\mathbf{B}_2 = \partial_{2,\mathcal{T}}[E_2, :]$. Without loss of generality (Claim 7.1), we assume all the triangles in the same disc of \mathcal{T} have the same orientation. So then, each row of \mathbf{B}_1 has exactly one entry being 1 and one being -1 and all the others 0.

Claim 7.11. *Let $e_1, e_2 \in E_2$ such that $D_{e_1} = D_{e_2}$. Then, the row $\mathbf{B}_2[e_2, :]$ is a linear combination of the rows in \mathbf{B}_1 and $\mathbf{B}_2[e_1, :]$.*

Proof. Suppose $D_{e_1} = D_{e_2} = \{\mathcal{D}_1, \dots, \mathcal{D}_l\}$ where $l = O(1)$. Suppose e_1 belongs to triangles $\triangle_1 \in \mathcal{D}_1, \dots, \triangle_l \in \mathcal{D}_l$ and e_2 belongs to $\triangle'_1 \in \mathcal{D}_1, \dots, \triangle'_l \in \mathcal{D}_l$. Since all the triangles in the same disc of \mathcal{T} have the same orientation, there exists $s \in \{\pm 1\}$ such that

$$\mathbf{B}_2[e_1, \triangle_1] = s\mathbf{B}_2[e_2, \triangle'_1], \dots, \mathbf{B}_2[e_1, \triangle_l] = s\mathbf{B}_2[e_2, \triangle'_l].$$

For any $1 \leq i \leq l$, let T_i be the set of triangles in \mathcal{D}_i and $E_{1,i}$ be the set of interior edges in \mathcal{D}_i . We claim there exists a vector \mathbf{a}_i such that

$$\mathbf{B}_2[e_2, T_i] = s\mathbf{B}_2[e_1, T_i] + \mathbf{a}_i^\top \mathbf{B}_1[E_{1,i}, T_i]. \quad (9)$$

Since $\mathbf{B}_2[e_2, :] = (\mathbf{B}_2[e_2, T_1] \ \cdots \ \mathbf{B}_2[e_2, T_l] \ \mathbf{0})$ and $E_{1,1}, \dots, E_{1,l}$ are disjoint, Equation (9) implies there exists a vector \mathbf{a} such that

$$\mathbf{B}_2[e_2, :] = s\mathbf{B}_2[e_1, :] + \mathbf{a}^\top \mathbf{B}_1.$$

It remains to prove Equation (9). Since both $\triangle_i, \triangle'_i \in \mathcal{D}_i$, we can find a sequence of triangles in \mathcal{D}_i : $\hat{\triangle}_0 = \triangle_i, \hat{\triangle}_1, \hat{\triangle}_2, \dots, \hat{\triangle}_j = \triangle'_i$ such that for any $1 \leq h \leq j$, the two triangles $\hat{\triangle}_{h-1}$ and $\hat{\triangle}_h$ share an interior edge of \mathcal{D}_l , denoted by \hat{e}_h . For every $1 \leq h \leq j$, the row $\mathbf{B}_1[\hat{e}_h, T_i]$ has exactly one entry being 1 and one entry -1 indexed at $\hat{\triangle}_{h-1}, \hat{\triangle}_h$ and all others 0. Denote $t = s\mathbf{B}_2[e_1, \triangle_i] \in \{\pm 1\}$. Thus, there exists $s_1 \in \{\pm 1\}$ such that $s\mathbf{B}_2[e_1, T_i] + s_1\mathbf{B}_1[\hat{e}_1, T_i] = (0 \ \cdots \ 0 \ t \ 0 \ \cdots \ 0)$ where the entry t has index $\hat{\triangle}_1$. By induction on h , there exist $s_1, \dots, s_j \in \{\pm 1\}$ such that

$$s\mathbf{B}_2[e_1, T_i] + s_1\mathbf{B}_1[\hat{e}_1, T_i] + \cdots + s_j\mathbf{B}_1[\hat{e}_j, T_i] = (0 \ \cdots \ 0 \ t \ 0 \ \cdots \ 0),$$

where the index of entry t is \triangle'_i . This proves Equation (9). \square

Due to the linear dependence in Claim 7.11, we can remove most of the rows of \mathbf{B}_2 . Among all the edges with the same D_e , we pick an arbitrary edge and put it into a set \hat{E}_2 . Let $\hat{\mathbf{B}}_2 \stackrel{\text{def}}{=} \mathbf{B}_2[\hat{E}_2, :]$ and

$$\mathbf{B} \stackrel{\text{def}}{=} \begin{pmatrix} \mathbf{B}_1 \\ \hat{\mathbf{B}}_2 \end{pmatrix}, \quad \mathbf{M} \stackrel{\text{def}}{=} \mathbf{B} \mathbf{W}_{2,\mathcal{T}} \mathbf{B}^\top.$$

Claim 7.12. *Let $\mathbf{b} \in \text{Im}(\partial_{2,\mathcal{T}})$ and let $\mathbf{b}_1 = \mathbf{b}[E_1 \cup \hat{E}_2]$. If \mathbf{x}_1 satisfies $\mathbf{M}\mathbf{x}_1 = \mathbf{b}_1$, then $\mathbf{x} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{0} \end{pmatrix}$ satisfies $\mathbf{L}_{1,\mathcal{T}}^{\text{up}} \mathbf{x} = \mathbf{b}$.*

Proof. By Claim 7.11, there exists a matrix \mathbf{H} such that $\mathbf{B}_2[E_2 \setminus \hat{E}_2, :] = \mathbf{H}\mathbf{B}$. Thus, $\partial_{2,\mathcal{T}}$ can be written as $\begin{pmatrix} \mathbf{B} \\ \mathbf{H}\mathbf{B} \end{pmatrix}$ up to row permutation. Since $\mathbf{b} \in \text{Im}(\partial_{2,\mathcal{T}})$, we know $\mathbf{b}_1 \in \text{Im}(\mathbf{B})$ and thus $\mathbf{M}\mathbf{x}_1 = \mathbf{b}_1$ is feasible and $\mathbf{b}_2 = \mathbf{H}\mathbf{b}_1$. Then,

$$\mathbf{L}_{1,\mathcal{T}}^{\text{up}} \mathbf{x} = \begin{pmatrix} \mathbf{B} \\ \mathbf{H}\mathbf{B} \end{pmatrix} \mathbf{W}_{2,\mathcal{T}} \begin{pmatrix} \mathbf{B}^\top & \mathbf{B}^\top \mathbf{H}^\top \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{B} \mathbf{W}_{2,\mathcal{T}} \mathbf{B}^\top \mathbf{x}_1 \\ \mathbf{H}\mathbf{B} \mathbf{W}_{2,\mathcal{T}} \mathbf{B}^\top \mathbf{x}_1 \end{pmatrix} = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{pmatrix}.$$

\square

7.2.2 Solver for the Smaller System

Let $\mathbf{M}_{11} \stackrel{\text{def}}{=} \mathbf{B}_1 \mathbf{W}_{2,\mathcal{T}} \mathbf{B}_1^\top, \mathbf{M}_{12} \stackrel{\text{def}}{=} \mathbf{B}_1 \mathbf{W}_{2,\mathcal{T}} \hat{\mathbf{B}}_2^\top, \mathbf{M}_{22} \stackrel{\text{def}}{=} \hat{\mathbf{B}}_2 \mathbf{W}_{2,\mathcal{T}} \hat{\mathbf{B}}_2^\top$. Then, we can write \mathbf{M} as a block matrix:

$$\mathbf{M} = \begin{pmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{12}^\top & \mathbf{M}_{22} \end{pmatrix}.$$

By Lemma 4.7, it suffices to design efficient solvers for systems in \mathbf{M}_{11} and systems in the Schur complement:

$$\text{Sc}[\mathbf{M}]_{\hat{E}_2} = \mathbf{M}_{22} - \mathbf{M}_{12}^\top \mathbf{M}_{11}^\dagger \mathbf{M}_{12}. \quad (10)$$

Let $m_1 = |E_1|, m_2 = |\hat{E}_2|$.

Claim 7.13. *For any $\mathbf{b} \in \text{Im}(\mathbf{M}_{11})$, we can compute \mathbf{x} such that $\mathbf{M}_{11}\mathbf{x} = \mathbf{b}$ in time $O(m_1)$.*

Proof. Without loss of generality (Claim 7.1), we assume all the triangles in the same disc of \mathcal{T} have the same orientation. Each row of \mathbf{B}_1 has exactly one entry with value 1 and one with value -1 and all others 0. Thus, $\mathbf{M}_{11} = \mathbf{B}_1 \mathbf{W}_2 \mathbf{B}_1^\top$ can be viewed as a down-Laplacian operator. By Lemma 4.5, we can solve a system in \mathbf{M}_{11} in linear time. \square

Claim 7.14. *We can compute $(\text{Sc}[\mathbf{M}]_{\widehat{E}_2})^\dagger$ in time $O(m_1 m_2 + m_2^3)$.*

Proof. We first compute the Schur complement $\text{Sc}[\mathbf{M}]_{\widehat{E}_2}$ defined in Equation (10). Note that $\mathbf{M}_{22} = \widehat{\mathbf{B}}_2 \mathbf{W}_{2,\mathcal{T}} \widehat{\mathbf{B}}_2^\top$ can be written as a weighted sum of the outer products of each nonzero column of $\widehat{\mathbf{B}}_2$. Since $\widehat{\mathbf{B}}_2$ has $O(m_2)$ nonzero columns and each nonzero column of $\widehat{\mathbf{B}}_2$ has at most 3 nonzero entries, we can compute \mathbf{M}_{22} in time $O(m_2)$. Similarly, we can compute $\mathbf{M}_{12} = \mathbf{B}_1 \mathbf{W}_{2,\mathcal{T}} \widehat{\mathbf{B}}_2^\top$ in time $O(m_1 + m_2)$. Then, by Lemma 7.13, we can find a matrix $\mathbf{X} \in \mathbb{R}^{m_1 \times m_2}$ such that $\mathbf{M}_{11} \mathbf{X} = \mathbf{M}_{12}$ in time $O(m_1 m_2)$. We can write $\mathbf{X} = \mathbf{X}_1 + \mathbf{X}_2$ where $\mathbf{X}_1 = \mathbf{M}_{11}^\dagger \mathbf{M}_{12}$ and each column of \mathbf{X}_2 is orthogonal to $\text{Im}(\mathbf{M}_{11}) = \text{Im}(\mathbf{B}_1)$. Then,

$$\mathbf{M}_{12}^\top \mathbf{X} = \widehat{\mathbf{B}}_2 \mathbf{W}_{2,\mathcal{T}} \mathbf{B}_1^\top \mathbf{X}_1 = \mathbf{M}_{12}^\top \mathbf{M}_{11}^\dagger \mathbf{M}_{12}.$$

Multiplying $\mathbf{M}_{12}^\top \mathbf{X}$ runs in time $O(m_1 m_2 + m_2^2)$ since \mathbf{B}_1 has $O(m_1)$ nonzeros and $\widehat{\mathbf{B}}_2$ has $O(m_2)$ nonzeros. Thus, the total time of computing the Schur complement is $O(m_1 m_2 + m_2^2)$.

We then show how to compute the pseudo-inverse of the Schur complement in time $O(m_2^3)$. We can decompose

$$\text{Sc}[\mathbf{M}]_{\widehat{E}_2} = \mathbf{Q} \begin{pmatrix} \mathbf{T} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \mathbf{Z},$$

where $\mathbf{Q}, \mathbf{Z} \in \mathbb{R}^{m_2 \times m_2}$ are orthogonal matrices and $\mathbf{T} \in \mathbb{R}^{k \times k}$ is a full-rank matrix and $k = \text{rank}(\text{Sc}[\mathbf{M}]_{\widehat{E}_2})$. Such a decomposition can be obtained in time $O(m_2^3)$ by Householder QR factorization with column pivoting (Chapter 5.4 of [VLG96]). Then,

$$(\text{Sc}[\mathbf{M}]_{\widehat{E}_2})^\dagger = \mathbf{Z}^\top \begin{pmatrix} \mathbf{T}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \mathbf{Q}^\top.$$

The inverse of \mathbf{T} can be computed in time $O(m_2^3)$ by matrix multiplication (Theorem 28.2 of [CLRS22]). Thus, the total runtime is $O(m_1 m_2 + m_2^3)$. \square

Claim 7.13 and 7.14 and Lemma 4.7 give us the following claim.

Claim 7.15. *Given $\mathbf{b} \in \text{Im}(\mathbf{M})$, we can find \mathbf{x} such that $\mathbf{M}\mathbf{x} = \mathbf{b}$ in time $O(m_1 + m_2^2)$ with a pre-processing time $O(m_1 m_2 + m_2^3)$.*

Proof. In the pre-processing, we compute the pseudo-inverse of $\text{Sc}[\mathbf{M}]_{\widehat{E}_2}$. To solve the system $\mathbf{M}\mathbf{x} = \mathbf{b}$, we solve the system in the Schur complement by multiplying its pseudo-inverse with the right-hand side vector. \square

We are ready to prove Lemma 7.9, which bounds the runtime of solving a system in $\mathbf{L}_{1,\mathcal{T}}^{\text{up}}$.

Proof of Lemma 7.9. Recall that \mathcal{T} is an r -hollowing of \mathcal{K} with n simplexes. So,

$$m_1 = O\left(\frac{n}{r} \cdot r^{2/3}\right) = O(nr^{-1/3}), \quad m_2 = O(nr^{-1}).$$

By Claim 7.12, it suffices to solve the system in \mathbf{M} . By Claim 7.15, we can solve a system in \mathbf{M} in time $O(nr^{-1/3} + n^2 r^{-2})$ with a pre-processing time $O(n^2 r^{-4/3} + n^3 r^{-3})$. \square

At the end of this section, we prove Lemma 4.13.

Proof of Lemma 4.13. We run the Preconditioned Conjugate Gradient (PCG) to find $\tilde{\mathbf{x}}$ by preconditioning the Schur complement $\text{Sc}[\mathbf{L}_{1,\mathcal{K}}^{\text{up}}]_C$ by $\mathbf{L}_{1,\mathcal{T}}^{\text{up}}$. By Lemma 4.10, the relative condition number

$$\kappa(\text{Sc}[\mathbf{L}_{1,\mathcal{K}}^{\text{up}}]_C, \mathbf{L}_{1,\mathcal{T}}^{\text{up}}) = O(rU).$$

By Theorem 2.6, the number of PCG iterations is $O\left(\sqrt{rU} \log(rU/\epsilon)\right)$.

In each PCG iteration, we need to solve a system of linear equations in $\mathbf{L}_{1,\mathcal{T}}^{\text{up}}$ with the right-hand side vector being the residual, which is in $\text{Im}(\text{Sc}[\mathbf{L}_1^{\text{up}}]_C) = \text{Im}(\mathbf{L}_{1,\mathcal{T}}^{\text{up}})$ (by Claim 7.5), and perform $O(1)$ matrix-vector multiplications and vector operations. By Lemma 7.9, with a pre-processing time $O(n^2r^{-4/3} + n^3r^{-3})$, we can solve a system in $\mathbf{L}_{1,\mathcal{T}}^{\text{up}}$ in time $O(nr^{-1/3} + n^2r^{-2})$. In addition, for any $\mathbf{f} \in \mathbb{R}^{|C|}$, we can multiply $\text{Sc}[\mathbf{L}_1^{\text{up}}]_C$ with \mathbf{f} by

$$\text{Sc}[\mathbf{L}_1^{\text{up}}]_C \mathbf{f} = \mathbf{L}_1^{\text{up}}[C, C] \mathbf{f} - \mathbf{L}_1^{\text{up}}[C, F] (\mathbf{L}_1^{\text{up}}[F, F])^\dagger \mathbf{L}_1^{\text{up}}[F, C] \mathbf{f}.$$

Here, all the matrix-vector multiplication has time $O(|C|) = O(nr^{-1/3})$. We can find \mathbf{y} such that $\mathbf{L}_1^{\text{up}}[F, F] \mathbf{y} = \mathbf{L}_1^{\text{up}}[F, C] \mathbf{f}$ in time $O(\frac{n}{r} \times r^{4/3}) = O(nr^{1/3})$ with a pre-processing time $O(\frac{n}{r} \times r^2) = O(nr)$ (by Lemma 4.12). Note \mathbf{y} may have a component orthogonal to the image of $\mathbf{L}_1^{\text{up}}[C, F]$, but this component does not contribute to $\mathbf{L}_1^{\text{up}}[C, F] \mathbf{y}$. We can write $\mathbf{y} = \mathbf{y}_1 + \mathbf{y}_2$ where $\mathbf{y}_1 = (\mathbf{L}_1^{\text{up}}[F, F])^\dagger \mathbf{L}_1^{\text{up}}[F, C] \mathbf{f}$ and \mathbf{y}_2 is orthogonal to the image of $\mathbf{L}_1^{\text{up}}[F, F]$. Then,

$$\mathbf{L}_1^{\text{up}}[C, F] \mathbf{y} = \mathbf{L}_1^{\text{up}}[C, F] \mathbf{y}_1 = \mathbf{L}_1^{\text{up}}[C, F] (\mathbf{L}_1^{\text{up}}[F, F])^\dagger \mathbf{L}_1^{\text{up}}[F, C] \mathbf{f}.$$

The total runtime of computing $\text{Sc}[\mathbf{L}_1^{\text{up}}]_C \mathbf{f}$ is $O(nr^{1/3})$. Therefore, the total runtime of PCG is

$$O\left(\sqrt{rU} \log(rU/\epsilon) \left(n^2r^{-2} + nr^{1/3}\right) + n^2r^{-4/3} + n^3r^{-3} + nr\right).$$

□

8 Proof of Theorem 3.1

Given all the four operators in Lemma 4.2, 4.5 and 4.6, we prove Theorem 3.1.

Proof of Theorem 3.1. Let κ be the maximum of $\kappa(\mathbf{L}_1^{\text{down}})$ and $\kappa(\mathbf{L}_1^{\text{up}})$. Let $\delta > 0$ be a parameter to be determined later. Let $\tilde{\Pi}_1^{\text{up}} = \tilde{\Pi}_1^{\text{up}}(\delta)$, $\tilde{\Pi}_1^{\text{down}} = \tilde{\Pi}_1^{\text{down}}(\delta)$ be defined in Lemma 4.2, and let $\mathbf{Z}_1^{\text{down}}$ be the operator in Lemma 4.5 with no error and \mathbf{Z}_1^{up} in Lemma 4.6 with error δ . Let

$$\begin{aligned} \tilde{\mathbf{b}}^{\text{up}} &\stackrel{\text{def}}{=} \tilde{\Pi}_1^{\text{up}} \mathbf{b}, \quad \tilde{\mathbf{b}}^{\text{down}} \stackrel{\text{def}}{=} \tilde{\Pi}_1^{\text{down}} \mathbf{b}, \\ \tilde{\mathbf{x}}^{\text{up}} &\stackrel{\text{def}}{=} \mathbf{Z}_1^{\text{up}} \tilde{\mathbf{b}}^{\text{up}}, \quad \tilde{\mathbf{x}}^{\text{down}} \stackrel{\text{def}}{=} \mathbf{Z}_1^{\text{down}} \tilde{\mathbf{b}}^{\text{down}}, \\ \tilde{\mathbf{x}} &\stackrel{\text{def}}{=} \tilde{\Pi}_1^{\text{up}} \tilde{\mathbf{x}}^{\text{up}} + \tilde{\Pi}_1^{\text{down}} \tilde{\mathbf{x}}^{\text{down}}. \end{aligned}$$

Then,

$$\|\mathbf{L}_1 \tilde{\mathbf{x}} - \Pi_1 \mathbf{b}\|_2 \leq \left\| \mathbf{L}_1^{\text{up}} \tilde{\Pi}_1^{\text{up}} \tilde{\mathbf{x}}^{\text{up}} - \tilde{\mathbf{b}}^{\text{up}} \right\|_2 + \left\| \mathbf{L}_1^{\text{down}} \tilde{\Pi}_1^{\text{down}} \tilde{\mathbf{x}}^{\text{down}} - \tilde{\mathbf{b}}^{\text{down}} \right\|_2 + \left\| \tilde{\mathbf{b}}^{\text{up}} + \tilde{\mathbf{b}}^{\text{down}} - \Pi_1 \mathbf{b} \right\|_2.$$

- For the first term:

$$\left\| \mathbf{L}_1^{\text{up}} \tilde{\mathbf{\Pi}}_1^{\text{up}} \tilde{\mathbf{x}}^{\text{up}} - \tilde{\mathbf{b}}^{\text{up}} \right\|_2 \leq \left\| \mathbf{L}_1^{\text{up}} \tilde{\mathbf{\Pi}}_1^{\text{up}} \tilde{\mathbf{x}}^{\text{up}} - \mathbf{L}_1^{\text{up}} \mathbf{\Pi}_1^{\text{up}} \tilde{\mathbf{x}}^{\text{up}} \right\|_2 + \left\| \mathbf{L}_1^{\text{up}} \tilde{\mathbf{x}}^{\text{up}} - \tilde{\mathbf{b}}^{\text{up}} \right\|_2.$$

By Lemma 4.2,

$$\begin{aligned} \left\| \mathbf{L}_1^{\text{up}} \tilde{\mathbf{\Pi}}_1^{\text{up}} \tilde{\mathbf{x}}^{\text{up}} - \mathbf{L}_1^{\text{up}} \mathbf{\Pi}_1^{\text{up}} \tilde{\mathbf{x}}^{\text{up}} \right\|_2 &\leq \left\| \mathbf{L}_1^{\text{up}} \right\|_2 \left\| \tilde{\mathbf{\Pi}}_1^{\text{up}} - \mathbf{\Pi}_1^{\text{up}} \right\|_2 \left\| \mathbf{\Pi}_1^{\text{up}} \tilde{\mathbf{x}}^{\text{up}} \right\|_2 \\ &\leq \delta \left\| \mathbf{L}_1^{\text{up}} \right\|_2 \left\| \mathbf{\Pi}_1^{\text{up}} \tilde{\mathbf{x}}^{\text{up}} \right\|_2. \end{aligned}$$

Let $\mathbf{y} \stackrel{\text{def}}{=} (\mathbf{L}_1^{\text{up}})^\dagger \tilde{\mathbf{b}}^{\text{up}}$. By Lemma 4.6,

$$\left\| \mathbf{\Pi}_1^{\text{up}} \tilde{\mathbf{x}}^{\text{up}} - \mathbf{y} \right\|_2 \leq \left\| (\mathbf{L}_1^{\text{up}})^\dagger \right\|_2 \left\| \mathbf{L}_1^{\text{up}} \mathbf{\Pi}_1^{\text{up}} \tilde{\mathbf{x}}^{\text{up}} - \tilde{\mathbf{b}}^{\text{up}} \right\|_2 \leq \delta \left\| (\mathbf{L}_1^{\text{up}})^\dagger \right\|_2 \left\| \tilde{\mathbf{b}}^{\text{up}} \right\|_2.$$

By the triangle inequality,

$$\left\| \mathbf{\Pi}_1^{\text{up}} \tilde{\mathbf{x}}^{\text{up}} \right\|_2 \leq \left\| \mathbf{y} \right\|_2 + \delta \left\| (\mathbf{L}_1^{\text{up}})^\dagger \right\|_2 \left\| \tilde{\mathbf{b}}^{\text{up}} \right\|_2 \leq (1 + \delta) \left\| (\mathbf{L}_1^{\text{up}})^\dagger \right\|_2 \left\| \tilde{\mathbf{b}}^{\text{up}} \right\|_2.$$

So,

$$\left\| \mathbf{L}_1^{\text{up}} \tilde{\mathbf{\Pi}}_1^{\text{up}} \tilde{\mathbf{x}}^{\text{up}} - \mathbf{L}_1^{\text{up}} \mathbf{\Pi}_1^{\text{up}} \tilde{\mathbf{x}}^{\text{up}} \right\|_2 \leq \delta(1 + \delta) \kappa \left\| \tilde{\mathbf{b}}^{\text{up}} \right\|_2.$$

By Lemma 4.6,

$$\left\| \mathbf{L}_1^{\text{up}} \tilde{\mathbf{x}}^{\text{up}} - \tilde{\mathbf{b}}^{\text{up}} \right\|_2 \leq \delta \left\| \tilde{\mathbf{b}}^{\text{up}} \right\|_2.$$

So,

$$\left\| \mathbf{L}_1^{\text{up}} \tilde{\mathbf{\Pi}}_1^{\text{up}} \tilde{\mathbf{x}}^{\text{up}} - \tilde{\mathbf{b}}^{\text{up}} \right\|_2 \leq 3\delta\kappa \left\| \tilde{\mathbf{b}}^{\text{up}} \right\|_2.$$

- For the second term, the operator $\mathbf{Z}_1^{\text{down}}$ has no error, which means $\mathbf{L}_1^{\text{down}} \tilde{\mathbf{x}}^{\text{down}} = \tilde{\mathbf{b}}^{\text{down}}$. Then,

$$\left\| \mathbf{L}_1^{\text{down}} \tilde{\mathbf{\Pi}}_1^{\text{down}} \tilde{\mathbf{x}}^{\text{down}} - \tilde{\mathbf{b}}^{\text{down}} \right\|_2 = \left\| \mathbf{L}_1^{\text{down}} \tilde{\mathbf{\Pi}}_1^{\text{down}} \tilde{\mathbf{x}}^{\text{down}} - \mathbf{L}_1^{\text{down}} \tilde{\mathbf{x}}^{\text{down}} \right\|_2 \leq \delta(1 + \delta) \kappa \left\| \tilde{\mathbf{b}}^{\text{down}} \right\|_2.$$

- For the third term:

$$\begin{aligned} \left\| \tilde{\mathbf{b}}^{\text{up}} + \tilde{\mathbf{b}}^{\text{down}} - \mathbf{\Pi}_1 \mathbf{b} \right\|_2^2 &= \left\| (\tilde{\mathbf{\Pi}}^{\text{up}} - \mathbf{\Pi}^{\text{up}}) \mathbf{b} \right\|_2^2 + \left\| (\tilde{\mathbf{\Pi}}^{\text{down}} - \mathbf{\Pi}^{\text{down}}) \mathbf{b} \right\|_2^2 \\ &\leq \delta^2 \left(\left\| \mathbf{\Pi}^{\text{up}} \mathbf{b} \right\|_2^2 + \left\| \mathbf{\Pi}^{\text{down}} \mathbf{b} \right\|_2^2 \right) \quad (\text{by Lemma 4.2 and Fact 2.5}) \\ &= \delta^2 \left\| \mathbf{\Pi}_1 \mathbf{b} \right\|_2^2. \end{aligned}$$

Combining all the above inequalities,

$$\begin{aligned} \left\| \mathbf{L}_1 \tilde{\mathbf{x}} - \mathbf{\Pi}_1 \mathbf{b} \right\|_2 &\leq 3\delta\kappa \left\| \tilde{\mathbf{b}}^{\text{up}} \right\|_2 + 2\delta\kappa \left\| \tilde{\mathbf{b}}^{\text{down}} \right\|_2 + \delta \left\| \mathbf{\Pi}_1 \mathbf{b} \right\|_2 \\ &\leq 3\delta\kappa(1 + \delta) \left\| \mathbf{\Pi}_1^{\text{up}} \mathbf{b} \right\|_2 + 2\delta\kappa(1 + \delta) \left\| \mathbf{\Pi}_1^{\text{down}} \mathbf{b} \right\|_2 + \delta \left\| \mathbf{\Pi}_1 \mathbf{b} \right\|_2 \\ &\leq 11\delta\kappa \left\| \mathbf{\Pi}_1 \mathbf{b} \right\|_2. \end{aligned}$$

Choosing $\delta \leq \frac{\epsilon}{11\kappa}$, we have

$$\left\| \mathbf{L}_1 \tilde{\mathbf{x}} - \mathbf{\Pi}_1 \mathbf{b} \right\|_2 \leq \epsilon \left\| \mathbf{\Pi}_1 \mathbf{b} \right\|_2.$$

□

9 Union of Stable 3-Complexes

In this section, we consider a union \mathcal{U} of stable 3-complex chunks $\mathcal{K}_1, \dots, \mathcal{K}_h$, and these chunks are glued together by identifying certain subsets of their boundary simplexes. Let k be the number of simplexes shared by more than one chunk. We remark that \mathcal{U} may *not* be embeddable in \mathbb{R}^3 and the first and second Betti numbers of \mathcal{U} are *no longer zero*. This makes designing efficient solvers for 1-Laplacian systems of \mathcal{U} much harder. Edelsbrunner and Parsa [EP14] showed that computing the first Betti number of a simplicial complex linearly embedded in \mathbb{R}^4 with m simplexes is as hard as computing the rank of a 0-1 matrix with $\Theta(m)$ nonzeros; Ding, Kyng, Probst Gutenberg, and Zhang [DKGZ22] showed that (approximately) solving 1-Laplacian systems for simplicial complexes in \mathbb{R}^4 is as hard as (approximately) solving general sparse systems of linear equations.

We design a 1-Laplacian solver for \mathcal{U} whose runtime is comparable to that of the 1-Laplacians solver for a single stable 3-complex when both h and k are small. From Lemma 4.1, we need four (approximate) operators: operators that project a vector onto $\text{Im}(\partial_1^\top)$ and onto $\text{Im}(\partial_2)$, and solvers for linear equations in $\mathbf{L}_1^{\text{down}}$ and \mathbf{L}_1^{up} . The approximate projection operator for $\text{Im}(\partial_1^\top)$ in Lemma 4.2 and the approximate solver for $\mathbf{L}_1^{\text{down}}$ in Section 4.1 hold for any simplicial complexes. However, since the first Betti number of \mathcal{U} is nonzero, the approximate projection operator for $\text{Im}(\partial_2)$ in Lemma 4.2 no longer holds. We need to design an approximate projection operator for $\text{Im}(\partial_2)$ and an approximate solver for the first up-Laplacian \mathbf{L}_1^{up} .

9.1 Projection onto the Image of the Up-Laplacian

We will approximate $\mathbf{\Pi}_1^{\text{up}} = \partial_2 (\partial_2^\top \partial_2)^\dagger \partial_2^\top$ and prove the following lemma.

Lemma 9.1 (Approximate projection operator onto $\text{Im}(\partial_2)$). *Let \mathcal{U} be a union of stable 3-complex chunks that are glued together by identifying certain subsets of their boundary simplexes. For any $\epsilon > 0$, there exists an operator $\tilde{\mathbf{\Pi}}_1^{\text{up}}$ such that*

$$(1 - \epsilon)\mathbf{\Pi}_1^{\text{up}} \preceq \tilde{\mathbf{\Pi}}_1^{\text{up}} \preceq (1 + \epsilon)\mathbf{\Pi}_1^{\text{up}}.$$

In addition, for any \mathbf{b} , we can compute $\tilde{\mathbf{\Pi}}_1^{\text{up}} \mathbf{b}$ in time $\tilde{O}(nk \log(1/\epsilon) + k^3)$, where n is the number of simplexes of \mathcal{U} and k is the number of simplexes shared by more than one chunk.

Our approach is similar to Lemma 4.7. Note that $\partial_2^\top \partial_2$ is the second down-Laplacian operator assuming unit edge weight. We implicitly (and approximately) solve a system in $\mathbf{L}_2^{\text{down}} = \partial_2^\top \partial_2$ via a variant of Lemma 4.7. We partition the *triangles* in \mathcal{U} into $F \cup C$. The following lemma states that we can decompose $\mathbf{\Pi}_1^{\text{up}}$ into two parts, where one relates to the F part and the other one relates to the Schur complement onto the C part.

Lemma 9.2. *The orthogonal projection $\mathbf{\Pi}_1^{\text{up}}$ can be decomposed as*

$$\mathbf{\Pi}_1^{\text{up}} = \mathbf{\Pi}_{\text{Im}(\partial_2[\cdot, F])} + \mathbf{\Pi}_{\text{Ker}(\partial_2^\top[F, \cdot])} \partial_2[\cdot, C] (\text{Sc}[\mathbf{L}_2^{\text{down}}]_C)^\dagger \partial_2^\top[C, \cdot] \mathbf{\Pi}_{\text{Ker}(\partial_2^\top[F, \cdot])}.$$

In addition, the second matrix in the right-hand side of the above equation is the orthogonal projection matrix onto the image of $\mathbf{\Pi}_{\text{Ker}(\partial_2^\top[F, \cdot])} \partial_2[\cdot, C]$.

Proof. For any $\mathbf{b} \in \mathbb{R}^n$, we define

$$\begin{aligned} \mathbf{f}_F &= \partial_2^\top[F, \cdot] \mathbf{b}, \quad \mathbf{f}_C = \partial_2^\top[C, \cdot] \mathbf{b} \\ \mathbf{h} &= \mathbf{f}_C - \mathbf{L}_2^{\text{down}}[C, F] (\mathbf{L}_2^{\text{down}}[F, F])^\dagger \mathbf{f}_F \\ \mathbf{x}_C &= (\text{Sc}[\mathbf{L}_2^{\text{down}}]_C)^\dagger \mathbf{h} \\ \mathbf{x}_F &= (\mathbf{L}_2^{\text{down}}[F, F])^\dagger (\mathbf{f}_F - \mathbf{L}_2^{\text{down}}[F, C] \mathbf{x}_C). \end{aligned} \tag{11}$$

Let $\mathbf{x} = \begin{pmatrix} \mathbf{x}_C \\ \mathbf{x}_F \end{pmatrix}$. Applying Lemma 4.7 with $\delta = 0$, we have $\partial_2^\top \partial_2 \mathbf{x} = \partial_2^\top \mathbf{b}$. That is, \mathbf{x} can be written as $\mathbf{x} = \mathbf{x}_1 + \mathbf{x}_2$ where $\mathbf{x}_1 = (\partial_2^\top \partial_2)^\dagger \partial_2^\top \mathbf{b}$ and \mathbf{x}_2 is in $\text{Ker}(\partial_2)$. Then, $\partial_2 \mathbf{x} = \partial_2 \mathbf{x}_1 = \Pi_1^{\text{up}} \mathbf{b}$. By Equation (11),

$$\Pi_1^{\text{up}} \mathbf{b} = \partial_2 \mathbf{x} = \Pi_{\text{Im}(\partial_2[:, F])} \mathbf{b} + \Pi_{\text{Ker}(\partial_2^\top [F, :])} \partial_2[:, C] (\text{Sc}[\mathbf{L}_2^{\text{down}}]_C)^\dagger \partial_2^\top [C, :] \Pi_{\text{Ker}(\partial_2^\top [F, :])} \mathbf{b}.$$

Thus, the equation in the statement holds. By Fact 2.4,

$$\text{Sc}[\mathbf{L}_2^{\text{down}}]_C = \partial_2^\top [C, :] \Pi_{\text{Ker}(\partial_2^\top [F, :])} \partial_2[:, C]. \quad (12)$$

The second matrix on the right-hand side of the equation in the statement is the orthogonal projection onto the image of $\Pi_{\text{Ker}(\partial_2^\top [F, :])} \partial_2[:, C]$. \square

We will approximate Π_1^{up} by constructing operators to approximate each part of the right-hand side of the equation in Lemma 9.2. We “cut off” the $O(k)$ tetrahedrons with a subsimplex shared by more than one chunk from \mathcal{U} . The remaining subcomplex is a collection of h disjoint subcomplexes of $\mathcal{K}_1, \dots, \mathcal{K}_h$ and each has zero first Betti number. An approximate projection operator onto the image of their up-Laplacians can be efficiently built by [CFM⁺14, BMNW22]. The Schur complement onto the cut-off part has dimensions $O(k) \times O(k)$. Assuming k is small, we can compute its pseudo-inverse efficiently.

More formally, let T be the set of the $O(k)$ tetrahedrons with a subsimplex shared by more than one chunk. Let \mathcal{G} be the 3-complex containing all the tetrahedrons in T and their subsimplexes. For $1 \leq i \leq h$, let \mathcal{S}_i be a complex that contains all the tetrahedrons in \mathcal{K}_i but not in T ; we also let \mathcal{S}_i include all the subsimplexes of its tetrahedron. All $\mathcal{S}_1, \dots, \mathcal{S}_h$ are disjoint. We let \mathcal{S} be the union of $\mathcal{S}_1, \dots, \mathcal{S}_h$. We partition all the *triangles* in \mathcal{U} into $F \cup C$ where F contains all the triangles in \mathcal{S} and C the other triangles. The size of C is $O(k)$.

Claim 9.3. *For any $\epsilon > 0$, there exists a linear operator \mathbf{Z} such that*

$$(1 - \epsilon) \mathbf{L}_2^{\text{down}}[F, F]^\dagger \preceq \Pi_F \mathbf{Z} \Pi_F \preceq \mathbf{L}_2^{\text{down}}[F, F]^\dagger,$$

where Π_F is the projection onto $\text{Im}(\mathbf{L}_2^{\text{down}}[F, F])$. In addition, we can compute $\mathbf{Z} \mathbf{b}$ in nearly-linear time for any $\mathbf{b} \in \text{Im}(\mathbf{L}_2^{\text{down}}[F, F])$.

The proof of the above claim is a combination of the following lemma from [BMNW22] and Lemma 4.2.

Lemma 9.4 (A special case of Lemma 3.9 of [BMNW22]⁸). *Let \mathcal{K} be a 3-complex that triangulates a three-ball, and let \mathcal{X} be a subcomplex of \mathcal{K} . Let t be the number of triangles in \mathcal{K} . Let ∂_2 be the second boundary operator of \mathcal{X} . There is a linear operator ∂_2^+ such that $\partial_2 \partial_2^+ \mathbf{b}_1 = \mathbf{b}_1$ for any $\mathbf{b}_1 \in \text{Im}(\partial_2)$ and $\partial_2^\top \partial_2^{+\top} \mathbf{b}_2 = \mathbf{b}_2$ for any $\mathbf{b}_2 \in \text{Im}(\partial_2^\top)$. In addition, both $\partial_2^+ \mathbf{b}_1$ and $\partial_2^{+\top} \mathbf{b}_2$ can be computed in time $O(t)$.*⁹

Proof of Claim 9.3. Since $\partial_2^\top [F, :] = (\partial_{2, \mathcal{S}}^\top \quad \mathbf{0})$, we can write $\mathbf{L}_2^{\text{down}}[F, F] = \text{diag}(\mathbf{L}_{2, \mathcal{S}_1}^{\text{down}}, \dots, \mathbf{L}_{2, \mathcal{S}_h}^{\text{down}})$. For each $1 \leq i \leq h$, let $\partial_{2, \mathcal{S}_i}^+$ and $\partial_{2, \mathcal{S}_i}^{+\top}$ be the operators for \mathcal{S}_i defined in Theorem 9.4, let $\Pi_{1, \mathcal{S}_i}^{\text{up}}$ be the projection onto the image of $\partial_{2, \mathcal{S}_i}$, and let $\tilde{\Pi}_{1, \mathcal{S}_i}^{\text{up}}$ be the corresponding approximate operator

⁸Lemma 3.9 of [BMNW22] holds for any $(d+1)$ -complex embedded in \mathbb{R}^{d+1} that collapses into its $(d-1)$ -skeleton via a known collapsing sequence.

⁹We remark that the ∂_2^+ and $\partial_2^{+\top}$ in the above theorem are different from the pseudo-inverse ∂_2^\dagger and $\partial_2^{\dagger\top}$.

defined in Lemma 4.2 with error ϵ . We let $\mathbf{Z}_i = \partial_{2,S_i}^+ \tilde{\Pi}_{1,S_i}^{\text{up}} \partial_{2,S_i}^{+\top}$. Let Π_i be the projection onto $\text{Im}(\mathbf{L}_{2,S_i}^{\text{down}})$. By Fact 2.2,

$$\begin{aligned} (1 - \epsilon) \partial_{2,S_i}^+ \Pi_{1,S_i}^{\text{up}} \partial_{2,S_i}^{+\top} &\preceq \mathbf{Z}_i \preceq \partial_{2,S_i}^+ \Pi_{1,S_i}^{\text{up}} \partial_{2,S_i}^{+\top} \\ \implies (1 - \epsilon) (\mathbf{L}_{2,S_i}^{\text{down}})^\dagger &\preceq \Pi_i \mathbf{Z}_i \Pi_i \preceq (\mathbf{L}_{2,S_i}^{\text{down}})^\dagger. \end{aligned}$$

We let $\mathbf{Z} = \text{diag}(\mathbf{Z}_1, \dots, \mathbf{Z}_r)$, which satisfies the requirement in the statement. By Lemma 9.4 and Lemma 4.2, we can compute $\mathbf{Z}\mathbf{b}$ for any $\mathbf{b} \in \text{Im}(\mathbf{L}_2^{\text{down}}[F, F])$ in nearly-linear time. \square

Claim 9.5. *For any $\epsilon > 0$, there exists a linear operator $\tilde{\Pi}_{\text{Ker}(\partial_2^\top[F, :])}$ such that*

$$(1 - \epsilon) \Pi_{\text{Ker}(\partial_2^\top[F, :])} \preceq \tilde{\Pi}_{\text{Ker}(\partial_2^\top[F, :])} \preceq \Pi_{\text{Ker}(\partial_2^\top[F, :])},$$

where $\Pi_{\text{Ker}(\partial_2^\top[F, :])}$ is the orthogonal projection matrix onto the kernel of $\partial_2^\top[F, :]$. In addition, for any \mathbf{b} , we can compute $\tilde{\Pi}_{\text{Ker}(\partial_2^\top[F, :])} \mathbf{b}$ in nearly-linear time.

Proof. Let $\Pi_{\text{Im}(\partial_2[\cdot, F])}$ be the orthogonal projection matrix onto the image of $\partial_2[\cdot, F]$. Note $\Pi_{\text{Im}(\partial_2[\cdot, F])} = \text{diag}(\Pi_{1,S_1}^{\text{up}}, \dots, \Pi_{1,S_h}^{\text{up}}, \mathbf{0})$. Since the first Betti number of each S_i is zero, we have

$$\Pi_{\text{Ker}(\partial_2^\top[F, :])} = \text{diag}(\Pi_{1,S_1}^{\text{down}}, \dots, \Pi_{1,S_h}^{\text{down}}, \mathbf{I}).$$

By Lemma 4.2, there exists an operator $\tilde{\Pi}_{\text{Ker}(\partial_2^\top[F, :])}$ such that

$$(1 - \epsilon) \Pi_{\text{Ker}(\partial_2^\top[F, :])} \preceq \tilde{\Pi}_{\text{Ker}(\partial_2^\top[F, :])} \preceq \Pi_{\text{Ker}(\partial_2^\top[F, :])}.$$

In addition, for any \mathbf{b} , we can compute $\tilde{\Pi}_{\text{Ker}(\partial_2^\top[F, :])} \mathbf{b}$ in nearly-linear time. \square

Claim 9.6. *For any $\epsilon > 0$, there exists a linear operator \mathbf{S} such that*

$$(\text{Sc}[\mathbf{L}_2^{\text{down}}]_C)^\dagger \preceq \mathbf{S} \preceq (1 + \epsilon) (\text{Sc}[\mathbf{L}_2^{\text{down}}]_C)^\dagger.$$

In addition, for any \mathbf{b} , we can compute $\mathbf{S}\mathbf{b}$ in time $\tilde{O}(nk + k^3)$ where n is the number of simplexes in \mathcal{U} and k is the number of simplexes shared by more than one chunk.

Proof. By Fact 2.4,

$$\text{Sc}[\mathbf{L}_2^{\text{down}}]_C = \partial_2^\top[C, :] \Pi_{\text{Ker}(\partial_2^\top[F, :])} \partial_2[\cdot, C].$$

Let $\tilde{\Pi}_{\text{Ker}(\partial_2^\top[F, :])}$ be the operator in Claim 9.5 with error $\delta = 1 - (1 + \epsilon)^{-1}$. Let

$$\mathbf{S}^\dagger \stackrel{\text{def}}{=} \partial_2^\top[C, :] \tilde{\Pi}_{\text{Ker}(\partial_2^\top[F, :])} \partial_2[\cdot, C]$$

and \mathbf{S} be the pseduo-inverse of \mathbf{S}^\dagger . By Fact 2.2,

$$(1 - \delta) \text{Sc}[\mathbf{L}_2^{\text{down}}]_C \preceq \mathbf{S}^\dagger \preceq \text{Sc}[\mathbf{L}_2^{\text{down}}]_C.$$

Thus,

$$(\text{Sc}[\mathbf{L}_2^{\text{down}}]_C)^\dagger \preceq \mathbf{S} \preceq (1 + \epsilon) (\text{Sc}[\mathbf{L}_2^{\text{down}}]_C)^\dagger.$$

In addition, we can compute $\mathbf{Y} = \tilde{\Pi}_{\text{Ker}(\partial_2^\top[F, :])} \partial_2[\cdot, C]$ in time $\tilde{O}(nk \log(1/\epsilon))$ by Lemma 9.5 and $\mathbf{S}^\dagger = \partial_2^\top[C, :] \mathbf{Y}$ in time $O(k^2)$ (since $\partial_2^\top[C, :]$ has $O(k)$ nonzeros). Since \mathbf{S}^\dagger has dimensions $O(k) \times O(k)$, we can compute \mathbf{S} in time $O(k^3)$ and $\mathbf{S}\mathbf{b}$ in time $O(k^2)$. Therefore, the total time is $\tilde{O}(nk \log(1/\epsilon) + k^3)$. \square

Combining the above claims, we prove Lemma 9.1. In addition, we will need the following lemma from [CFM⁺14] to prove Theorem 3.1.

Lemma 9.7 (Lemma 4.5 of [CFM⁺14]). *Let $\mathbf{A} : \mathcal{C}_1 \rightarrow \mathcal{C}_1$ be a symmetric linear operator and $\mathbf{\Pi}$ be an orthogonal projection and $\tilde{\mathbf{\Pi}}$ be a linear operator such that $(1 - \epsilon)\mathbf{\Pi} \preceq \tilde{\mathbf{\Pi}} \preceq (1 + \epsilon)\mathbf{\Pi}$. Then,*

$$(1 - 3\epsilon\kappa^{\mathbf{\Pi}}(\mathbf{A}))\mathbf{\Pi}\mathbf{A}\mathbf{\Pi} \preceq \tilde{\mathbf{\Pi}}\mathbf{A}\tilde{\mathbf{\Pi}} \preceq (1 + 3\epsilon\kappa^{\mathbf{\Pi}}(\mathbf{A}))\mathbf{\Pi}\mathbf{A}\mathbf{\Pi},$$

where $\kappa^{\mathbf{\Pi}}(\mathbf{A})$ is the condition number of \mathbf{A} restricted to $\text{Im}(\mathbf{\Pi})$.

Proof of Lemma 9.1. Let $\delta > 0$ be an error parameter to be determined later. Let \mathbf{Z} be the operator in Claim 9.3, \mathbf{S} the operator in Claim 9.6 and $\tilde{\mathbf{\Pi}}_{\text{Ker}(\partial_2^\top[F, :])}$ the operator in Claim 9.5, all with error δ . We set

$$\tilde{\mathbf{\Pi}}_1^{\text{up}} \stackrel{\text{def}}{=} \partial_2[:, F]\mathbf{Z}\partial_2^\top[F, :] + \tilde{\mathbf{\Pi}}_{\text{Ker}(\partial_2^\top[F, :])}\partial_2[:, C]\mathbf{S}\partial_2^\top[C, :]\tilde{\mathbf{\Pi}}_{\text{Ker}(\partial_2^\top[F, :])}.$$

Note that $\mathbf{\Pi}_{\text{Im}(\partial_2[:, F])} = \partial_2[:, F](\partial_2^\top[F, :]\partial_2[:, F])^\dagger\partial_2^\top[F, :]$. By Claim 9.3 and Fact 2.2,

$$(1 - \delta)\mathbf{\Pi}_{\text{Im}(\partial_2[:, F])} \preceq \partial_2[:, F]\mathbf{Z}\partial_2^\top[F, :] \preceq \mathbf{\Pi}_{\text{Im}(\partial_2[:, F])}.$$

Similarly, by Claim 9.6 and Fact 2.2,

$$\partial_2[:, C](\text{Sc}[\mathbf{L}_2^{\text{down}}]_C)^\dagger\partial_2^\top[C, :] \preceq \partial_2[:, C]\mathbf{S}\partial_2^\top[C, :] \preceq (1 + \delta)\partial_2[:, C](\text{Sc}[\mathbf{L}_2^{\text{down}}]_C)^\dagger\partial_2^\top[C, :].$$

By Claim 9.5 and Lemma 9.7,

$$\begin{aligned} & (1 - 7\delta\kappa)\mathbf{\Pi}_{\text{Ker}(\partial_2^\top[F, :])}\partial_2[:, C](\text{Sc}[\mathbf{L}_2^{\text{down}}]_C)^\dagger\partial_2^\top[C, :]\mathbf{\Pi}_{\text{Ker}(\partial_2^\top[F, :])} \\ & \preceq \tilde{\mathbf{\Pi}}_{\text{Ker}(\partial_2^\top[F, :])}\partial_2[:, C]\mathbf{S}\partial_2^\top[C, :]\tilde{\mathbf{\Pi}}_{\text{Ker}(\partial_2^\top[F, :])} \\ & \preceq (1 + 7\delta\kappa)\mathbf{\Pi}_{\text{Ker}(\partial_2^\top[F, :])}\partial_2[:, C](\text{Sc}[\mathbf{L}_2^{\text{down}}]_C)^\dagger\partial_2^\top[C, :]\mathbf{\Pi}_{\text{Ker}(\partial_2^\top[F, :])}. \end{aligned}$$

Here, κ is the condition number of $\mathbf{\Pi}_{\text{Ker}(\partial_2^\top[F, :])}\partial_2[:, C](\text{Sc}[\mathbf{L}_2^{\text{down}}]_C)^\dagger\partial_2^\top[C, :]\mathbf{\Pi}_{\text{Ker}(\partial_2^\top[F, :])}$, which is the orthogonal projection of $\mathbf{\Pi}_{\text{Ker}(\partial_2^\top[F, :])}\partial_2[:, C]$ (by Lemma 9.2). Thus, $\kappa = 1$. We choose $\delta = \frac{\epsilon}{7}$. Then,

$$(1 - \epsilon)\mathbf{\Pi}_1^{\text{up}} \preceq \tilde{\mathbf{\Pi}}_1^{\text{up}} \preceq (1 + \epsilon)\mathbf{\Pi}_1^{\text{up}}.$$

To compute $\tilde{\mathbf{\Pi}}_1^{\text{up}}\mathbf{b}$ for a vector \mathbf{b} , we need to apply $\tilde{\mathbf{\Pi}}_{\text{Ker}(\partial_2^\top[F, :])}$ twice in nearly-linear time, apply \mathbf{Z} once in nearly-linear time, and apply \mathbf{S} once in $\tilde{O}(nk \log(\kappa/\epsilon) + k^3)$ time, and perform a constant number of matrix-vector multiplications in nearly-linear time. Thus, the total runtime is $\tilde{O}(nk \log(\kappa/\epsilon) + k^3)$. \square

9.2 Solver for the Up-Laplacian

In this section, we build an efficient solver for the up-Laplacian \mathbf{L}_1^{up} of \mathcal{U} and prove the following lemma.

Lemma 9.8 (Up-Laplacian solver for unions of stable complexes). *Let \mathcal{U} be a union of h stable 3-complex chunks that are glued together by identifying certain subsets of their boundary simplexes. For any $\mathbf{b} \in \text{Im}(\mathbf{L}_1^{\text{up}})$ and $\epsilon > 0$, we can compute an $\tilde{\mathbf{x}}$ such that $\|\mathbf{L}_1^{\text{up}}\tilde{\mathbf{x}} - \mathbf{b}\|_2 \leq \epsilon\|\mathbf{b}\|_2$ in time $\tilde{O}(n^{3/2}h^{3/2} + h^3k^3 + n^{1/4}U^{1/2}(nh + h^2k^2 + n^{7/6})\log(nU/\epsilon))$ where n is the number of simplexes in \mathcal{U} and k is the number of simplexes shared by more than one chunk. and U is the ratio between the maximum and the minimum triangle weights in \mathcal{U} .*

Proof. Our approach is similar to that for a single stable 3-complex. For each $1 \leq i \leq h$, let n_i be the number of simplexes in \mathcal{S}_i and $r_i = \lfloor n_i^{1/2} \rfloor$. We slightly modify Algorithm 2 `HOLLOWING`(\mathcal{K}_i, r_i) to compute an r_i -hollowing \mathcal{T}_i for \mathcal{S}_i : in Line 2, we initialize \mathcal{T} to contain all the boundary simplexes of \mathcal{S}_i , instead of \mathcal{K}_i ; in the for-loop from Line 4 to Line 8, our operations are on \mathcal{S}_i instead of \mathcal{K}_i . Since all simplexes in $\mathcal{K}_i \setminus \mathcal{S}_i$ have $O(1)$ Euclidean distances to the boundary of \mathcal{S}_i , only the regions containing boundary simplexes of \mathcal{S}_i are different from the corresponding regions of \mathcal{K}_i from the unmodified `HOLLOWING`(\mathcal{K}_i, r_i). We can check these regions have $O(r_i)$ simplexes and $O(r_i^{2/3})$ boundary simplexes. Thus, \mathcal{T}_i returned by the modified Algorithm 2 is an r_i -hollowing of \mathcal{S}_i .

We let F be the set of all the interior edges of the regions of \mathcal{S}_i w.r.t. \mathcal{T}_i , and let E be the set of all the other edges. Since \mathcal{S}_i 's are disjoint, we can write $\mathbf{L}_1^{\text{up}}[F, F] = \text{diag}(\mathbf{L}_1^{\text{up}}[F_1, F_1], \dots, \mathbf{L}_1^{\text{up}}[F_h, F_h])$ where F_i contains all the edges in the intersection of F and the edge set of \mathcal{S}_i . By Lemma 4.12, with a pre-processing time

$$O\left(\sum_{i=1}^h n_i r_i\right) = O\left(\sum_{i=1}^h n_i^{3/2}\right) = O\left(n^{3/2}\right),$$

for any $\mathbf{b} \in \text{Im}(\mathbf{L}_1^{\text{up}}[F, F])$, we can find \mathbf{x} such that $\mathbf{L}_1^{\text{up}}[F, F]\mathbf{x} = \mathbf{b}$ in time

$$O\left(\sum_{i=1}^h n_i r_i^{1/3}\right) = O\left(\sum_{i=1}^h n_i^{7/6}\right) = O\left(n^{7/6}\right).$$

To solve the system in the Schur complement $\text{Sc}[\mathbf{L}_1^{\text{up}}]_C$, we precondition it by the union of \mathcal{T}_i 's and \mathcal{G} (recall \mathcal{G} is the 3-complex that contains all the tetrahedrons with a subsimplex shared by more than one chunk and the subsimplexes of these tetrahedrons). We run Preconditioned Conjugate Gradient. We write $\mathbf{L}_1^{\text{up}} = \mathbf{L}_{1,\mathcal{S}}^{\text{up}} + \mathbf{L}_{1,\mathcal{G}'}^{\text{up}}$ where \mathcal{G}' is \mathcal{G} without the triangles in \mathcal{S} , and we put zeros so that all the three up-Laplacians have equal dimensions. Since edges in F are \triangle -disjoint from the edges in \mathcal{G}' , we know

$$\text{Sc}[\mathbf{L}_1^{\text{up}}]_C = \text{Sc}[\mathbf{L}_{1,\mathcal{S}}^{\text{up}}]_C + \mathbf{L}_{1,\mathcal{G}'}^{\text{up}}[C, C].$$

By Lemma 4.10 and the fact that all \mathcal{S}_i 's are disjoint, the union of the hollowing for each chunk $\mathcal{T} = \mathcal{T}_1 \cup \dots \cup \mathcal{T}_h$ satisfies

$$\mathbf{L}_{1,\mathcal{T}}^{\text{up}} \preceq \text{Sc}[\mathbf{L}_{1,\mathcal{S}}^{\text{up}}]_C \preceq O(r_{\max} U) \mathbf{L}_{1,\mathcal{T}}^{\text{up}},$$

where $r_{\max} = \max_{1 \leq i \leq h} r_i$. Let \mathcal{R} be the union of \mathcal{T} and \mathcal{G}' . Then,

$$\mathbf{L}_{1,\mathcal{R}}^{\text{up}} \preceq \text{Sc}[\mathbf{L}_1^{\text{up}}]_C \preceq O(r_{\max} U) \mathbf{L}_{1,\mathcal{R}}^{\text{up}}.$$

To solve a system in $\mathbf{L}_{1,\mathcal{R}}^{\text{up}}$, we partition the edges in \mathcal{R} into $E_1 \cup E_2$ where E_1 contains all the edges shared by exactly two triangles and E_2 the other edges. Similar to our solver for the preconditioner in Section 7.2, we choose a subset $\hat{E}_2 \subset E_2$ such that every row of ∂_2 corresponding to an edge in $E_2 \setminus \hat{E}_2$ can be written as a linear combination of the rows in $\partial_2[\hat{E}_2 \cup E_1, :]$. We first include all the edges in \mathcal{G} to \hat{E}_2 ; then, for each $1 \leq i \leq h$, among all the edges in $E_2 \setminus \hat{E}_2$ shared by the same set of discs in $\mathcal{T}_i \setminus \hat{E}_2$, we include an arbitrary edge to \hat{E}_2 . By Jensen's inequality,

$$\begin{aligned} |E_1| &= O\left(\sum_{i=1}^h n_i r_i^{-1/3}\right) = O\left(\sum_{i=1}^h n_i^{5/6}\right) = O\left(n^{5/6} h^{1/6}\right), \\ |\hat{E}_2| &= O\left(\sum_{i=1}^h n_i r_i^{-1} + (h+1)k\right) = O\left(\sum_{i=1}^h n_i^{1/2} + hk\right) = O\left(n^{1/2} h^{1/2} + hk\right). \end{aligned}$$

By the proof of Lemma 7.9, with a pre-processing time

$$O\left(n^{5/6}h^{1/6}\left(n^{1/2}h^{1/2}+hk\right)+\left(n^{1/2}h^{1/2}+hk\right)^3\right)=O\left(n^{3/2}h^{3/2}+h^3k^3\right),$$

we can solve a system in $\mathbf{L}_{1,\mathcal{R}}^{\text{up}}$ in time

$$O\left(n^{5/6}h^{1/6}+\left(n^{1/2}h^{1/2}+hk\right)^2\right)=O\left(nh+h^2k^2\right).$$

Therefore, the total runtime of approximately solving a system in \mathbf{L}_1^{up} is

$$O\left(n^{3/2}h^{3/2}+h^3k^3+n^{1/4}U^{1/2}\left(nh+h^2k^2+n^{7/6}\right)\log(nU/\epsilon)\right).$$

□

Combining Section 9.1 and 9.2, we have the following lemma.

Proof of Theorem 3.2. Combine Lemma 9.1 and 9.8, and Lemma 4.2 and 4.5.

□

References

- [Axe85] Owe Axelsson. A survey of preconditioned iterative methods for linear systems of algebraic equations. *BIT Numerical Mathematics*, 25(1):165–187, 1985.
- [BHP01] Gill Barequet and Sarel Har-Peled. Efficiently approximating the minimum-volume bounding box of a point set in three dimensions. *Journal of Algorithms*, 38(1):91–109, 2001.
- [BMNW22] Mitchell Black, William Maxwell, Amir Nayyeri, and Eli Winkelman. Computational topology in a collapsing universe: Laplacians, homology, cohomology*. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 226–251. SIAM, 2022.
- [BN22] Mitchell Black and Amir Nayyeri. Hodge decomposition and general laplacian solvers for embedded simplicial complexes. In *49th International Colloquium on Automata, Languages, and Programming (ICALP 2022)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2022.
- [BV21] Mitali Bafna and Nikhil Vyas. Optimal Fine-Grained Hardness of Approximation of Linear Equations. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming (ICALP 2021)*, volume 198 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 20:1–20:19, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.
- [Car09] Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009.
- [CdSGO16] Frédéric Chazal, Vin de Silva, Marc Glisse, and Steve Oudot. *The Structure and Stability of Persistence Modules*. Springer, October 2016.
- [CFM⁺14] Michael B Cohen, Brittany Terese Fasy, Gary L Miller, Amir Nayyeri, Richard Peng, and Noel Walkington. Solving 1-laplacians in nearly linear time: Collapsing and expanding a topological ball. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 204–216. SIAM, 2014.
- [Chi67] David RJ Chillingworth. Collapsing three-dimensional convex polyhedra. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 63, pages 353–357. Cambridge University Press, 1967.
- [Chi80] David RJ Chillingworth. Collapsing three-dimensional convex polyhedra: correction. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 88, pages 307–310. Cambridge University Press, 1980.
- [Chu93] Fan Chung. The laplacian of a hypergraph. *Expanding graphs (DIMACS series)*, pages 21–36, 1993.
- [CKM⁺14] Michael B Cohen, Rasmus Kyng, Gary L Miller, Jakub W Pachocki, Richard Peng, Anup B Rao, and Shen Chen Xu. Solving sdd linear systems in nearly $m \log 1/2 n$ time. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 343–352. ACM, 2014.

- [CLRS22] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2022.
- [DKGZ22] Ming Ding, Rasmus Kyng, Maximilian Probst Gutenberg, and Peng Zhang. Hardness results for laplacians of simplicial complexes via sparse-linear equation complete gadgets. In *49th International Colloquium on Automata, Languages, and Programming (ICALP 2022)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2022.
- [Eck44] Beno Eckmann. Harmonische funktionen und randwertaufgaben in einem komplex. *Commentarii Mathematici Helvetici*, 17(1):240–255, 1944.
- [EH10] Herbert Edelsbrunner and John Harer. *Computational Topology: An Introduction*. American Mathematical Soc., 2010.
- [EP14] Herbert Edelsbrunner and Salman Parsa. On the computational complexity of betti numbers: reductions from matrix rank. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on discrete algorithms*, pages 152–160. SIAM, 2014.
- [Fed87] Greg N Federickson. Fast algorithms for shortest paths in planar graphs, with applications. *SIAM Journal on computing*, 16(6):1004–1022, 1987.
- [Geo73] Alan George. Nested dissection of a regular finite element mesh. *SIAM Journal on Numerical Analysis*, 10(2):345–363, 1973.
- [Ghr08] Robert Ghrist. Barcodes: the persistent topology of data. *Bulletin of the American Mathematical Society*, 45(1):61–75, 2008.
- [HJ13] Danijela Horak and Jürgen Jost. Spectra of combinatorial laplace operators on simplicial complexes. *Advances in Mathematics*, 244:303–336, 2013.
- [JS21] Arun Jambulapati and Aaron Sidford. Ultrasparse ultrasparsifiers and faster laplacian system solvers. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 540–559. SIAM, 2021.
- [KLP⁺16] Rasmus Kyng, Yin Tat Lee, Richard Peng, Sushant Sachdeva, and Daniel A Spielman. Sparsified cholesky and multigrid solvers for connection laplacians. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 842–850, 2016.
- [KMP10] Ioannis Koutis, Gary L. Miller, and Richard Peng. Approaching optimality for solving SDD linear systems. In *Proceedings of the 2010 IEEE 51st Annual Symposium on Foundations of Computer Science, FOCS '10*, pages 235–244, Washington, DC, USA, 2010. IEEE Computer Society.
- [KMP11] Ioannis Koutis, Gary L. Miller, and Richard Peng. A nearly-m log n time solver for SDD linear systems. In *Proceedings of the 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS '11*, pages 590–598, Washington, DC, USA, 2011. IEEE Computer Society.
- [KOSZ13] Jonathan A Kelner, Lorenzo Orecchia, Aaron Sidford, and Zeyuan Allen Zhu. A simple, combinatorial algorithm for solving sdd systems in nearly-linear time. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 911–920, 2013.

- [KPSZ18] Rasmus Kyng, Richard Peng, Robert Schwieterman, and Peng Zhang. Incomplete nested dissection. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 404–417, 2018.
- [KS16] Rasmus Kyng and Sushant Sachdeva. Approximate gaussian elimination for laplacians-fast, sparse, and simple. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, pages 573–582. IEEE, 2016.
- [KZ20] Rasmus Kyng and Peng Zhang. Hardness results for structured linear systems. *SIAM Journal on Computing*, 49(4):FOCS17–280, 2020.
- [Lim20] Lek-Heng Lim. Hodge laplacians on graphs. *Siam Review*, 62(3):685–715, 2020.
- [LRT79] Richard J Lipton, Donald J Rose, and Robert Endre Tarjan. Generalized nested dissection. *SIAM journal on numerical analysis*, 16(2):346–358, 1979.
- [LS13] Yin Tat Lee and Aaron Sidford. Efficient accelerated coordinate descent methods and faster algorithms for solving linear systems. In *2013 IEEE 54th annual symposium on foundations of computer science*, pages 147–156. IEEE, 2013.
- [Moh91] Bojan Mohar. Eigenvalues, diameter, and mean distance in graphs. *Graphs and combinatorics*, 7(1):53–64, 1991.
- [MT90] Gary L Miller and William Thurston. Separators in two and three dimensions. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 300–309, 1990.
- [MTTV98] Gary L Miller, Shang-Hua Teng, William Thurston, and Stephen A Vavasis. Geometric separators for finite-element meshes. *SIAM Journal on Scientific Computing*, 19(2):364–386, 1998.
- [Nie22] Zipei Nie. Matrix anti-concentration inequalities with applications. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 568–581, 2022.
- [OPW20] Braxton Osting, Sourabh Palande, and Bei Wang. Spectral sparsification of simplicial complexes for clustering and label propagation. *J. Comput. Geom.*, 11(1):176–211, 2020.
- [PS14] Richard Peng and Daniel A Spielman. An efficient parallel solver for sdd linear systems. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 333–342. ACM, 2014.
- [PV21] Richard Peng and Santosh Vempala. Solving sparse linear systems faster than matrix multiplication. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 504–521. SIAM, 2021.
- [ST14] Daniel A Spielman and Shang-Hua Teng. Nearly linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems. *SIAM Journal on Matrix Analysis and Applications*, 35(3):835–885, 2014.
- [Tan16] Martin Tancer. Recognition of collapsible complexes is np-complete. *Discrete & Computational Geometry*, 55(1):21–38, 2016.

- [VLG96] Charles F Van Loan and G Golub. Matrix computations (johns hopkins studies in mathematical sciences). *Matrix Computations*, 1996.
- [Zom05] Afra J. Zomorodian. *Topology for Computing*, volume 16. Cambridge university press, 2005.

A Missing Linear Algebra Proofs

A.1 Missing Proofs in Section 2.1

Proof of Fact 2.1. By the definition of image, $\text{Im}(\mathbf{A}) \supseteq \text{Im}(\mathbf{A}\mathbf{A}^\top)$. It suffices to show $\text{Im}(\mathbf{A}) \subseteq \text{Im}(\mathbf{A}\mathbf{A}^\top)$. Let \mathbf{x} be an arbitrary vector in $\text{Im}(\mathbf{A})$. Then, $\mathbf{x} = \mathbf{A}\mathbf{y}$ for some $\mathbf{y} \in \mathbb{R}^n$. Write $\mathbf{y} = \mathbf{y}_1 + \mathbf{y}_2$ such that $\mathbf{y}_1 \in \text{Im}(\mathbf{A}^\top)$ and $\mathbf{y}_2 \in \text{Ker}(\mathbf{A})$. Let \mathbf{z} satisfy $\mathbf{A}^\top \mathbf{z} = \mathbf{y}_1$. Then,

$$\mathbf{A}\mathbf{A}^\top \mathbf{z} = \mathbf{A}\mathbf{y}_1 = \mathbf{A}\mathbf{y} = \mathbf{x}.$$

Thus, $\mathbf{x} \in \text{Im}(\mathbf{A}\mathbf{A}^\top)$. □

Proof of Fact 2.3. We multiply the right-hand side:

$$\begin{aligned} & \begin{pmatrix} \mathbf{I} & \\ \mathbf{A}[C, F]\mathbf{A}[F, F]^\dagger & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{A}[F, F] & \\ & \text{Sc}[\mathbf{A}]_C \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{A}[F, F]^\dagger \mathbf{A}[F, C] \\ & \mathbf{I} \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{I} & \\ \mathbf{A}[C, F]\mathbf{A}[F, F]^\dagger & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{A}[F, F] & \mathbf{A}[F, C] \\ & \text{Sc}[\mathbf{A}]_C \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{A}[F, F] & \mathbf{A}[F, C] \\ \mathbf{A}[C, F] & \mathbf{A}[C, F]\mathbf{A}[F, F]^\dagger \mathbf{A}[F, C] + \text{Sc}[\mathbf{A}]_C \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{A}[F, F] & \mathbf{A}[F, C] \\ \mathbf{A}[C, F] & \mathbf{A}[C, C] \end{pmatrix} = \mathbf{A}. \end{aligned}$$

□

Proof of Fact 2.4. By the definition of the Schur complement,

$$\begin{aligned} \text{Sc}[\mathbf{A}]_C &= \mathbf{B}_C \mathbf{B}_C^\top - \mathbf{B}_C \mathbf{B}_F^\top (\mathbf{B}_F \mathbf{B}_F^\top)^\dagger \mathbf{B}_F \mathbf{B}_C^\top \\ &= \mathbf{B}_C \left(\mathbf{I} - \mathbf{B}_F^\top (\mathbf{B}_F \mathbf{B}_F^\top)^\dagger \mathbf{B}_F \right) \mathbf{B}_C^\top = \mathbf{B}_C \mathbf{\Pi}_{\text{Ker}(\mathbf{B}_F)} \mathbf{B}_C^\top. \end{aligned}$$

□

Proof of Fact 2.5. For the first statement,

$$\begin{aligned} \|\mathbf{A}\mathbf{Z}\mathbf{b} - \mathbf{b}\|_2 &= \left\| \mathbf{A}^{1/2} \left(\mathbf{A}^{1/2} \mathbf{Z} \mathbf{A}^{1/2} - \mathbf{\Pi} \right) \mathbf{A}^{\dagger/2} \mathbf{b} \right\|_2 \\ &\leq \sqrt{\kappa(\mathbf{A})} \left\| \mathbf{A}^{1/2} \mathbf{Z} \mathbf{A}^{1/2} - \mathbf{\Pi} \right\|_2 \|\mathbf{b}\|_2. \end{aligned}$$

By the assumption, $(1 - \epsilon)\mathbf{\Pi} \preceq \mathbf{A}^{1/2} \mathbf{Z} \mathbf{A}^{1/2} \preceq (1 + \epsilon)\mathbf{\Pi}$. Thus,

$$\|\mathbf{A}\mathbf{Z}\mathbf{b} - \mathbf{b}\|_2 \leq \epsilon \sqrt{\kappa(\mathbf{A})} \|\mathbf{b}\|_2.$$

For the second statement, we know $\|\mathbf{A}\mathbf{\Pi}\mathbf{Z}\mathbf{\Pi} - \mathbf{\Pi}\|_2 \leq \epsilon$. So,

$$1 - \epsilon \leq \lambda_{\min}(\mathbf{A}\mathbf{\Pi}\mathbf{Z}\mathbf{\Pi}) \leq \lambda_{\max}(\mathbf{A}\mathbf{\Pi}\mathbf{Z}\mathbf{\Pi}) \leq 1 + \epsilon.$$

Since matrices $\mathbf{A}\mathbf{\Pi}\mathbf{Z}\mathbf{\Pi}$ and $\mathbf{A}^{1/2}\mathbf{\Pi}\mathbf{Z}\mathbf{\Pi}\mathbf{A}^{1/2}$ have the same eigenvalues, we have $(1-\epsilon)\mathbf{A}^\dagger \preceq \mathbf{\Pi}\mathbf{Z}\mathbf{\Pi} \preceq (1+\epsilon)\mathbf{A}^\dagger$. \square

A.2 Missing Proofs in Section 4

Proof of Lemma 4.7. Since $\mathbf{b} \in \text{Im}(\mathbf{L}_1^{\text{up}})$, we know $\mathbf{b}_F \in \text{Im}(\mathbf{L}_1^{\text{up}}[F, F])$ and $\mathbf{b}_F - \mathbf{L}_1^{\text{up}}[F, C]\tilde{\mathbf{x}}_C \in \text{Im}(\mathbf{L}_1^{\text{up}}[F, F])$. We can apply the solver UPLAPFSOLVER to these two vectors. By the statement assumption, we can write $\text{UPLAPFSOLVER}(\mathbf{b}_F) = \mathbf{L}_1^{\text{up}}[F, F]^\dagger \mathbf{b}_F + \mathbf{y}$, where $\partial_2[F, :]^\top \mathbf{y} = \mathbf{0}$. Then,

$$\mathbf{h} = \mathbf{b}_C - \mathbf{L}_1^{\text{up}}[C, F] \cdot \text{UPLAPFSOLVER}(\mathbf{b}_F) = \mathbf{b}_C - \mathbf{L}_1^{\text{up}}[C, F] \cdot \mathbf{L}_1^{\text{up}}[F, F]^\dagger \mathbf{b}_F.$$

We first show that $\mathbf{h} \in \text{Im}(\text{Sc}[\mathbf{L}_1^{\text{up}}]_C)$ so that we can apply the solver SCHURSOLVER to \mathbf{h} and obtain a vector $\tilde{\mathbf{x}}_C$ satisfying $\|\text{Sc}[\mathbf{L}_1^{\text{up}}]_C \tilde{\mathbf{x}}_C - \mathbf{h}\|_2 \leq \delta \|\mathbf{h}\|_2$. Since $\mathbf{b} \in \text{Im}(\mathbf{L}_1^{\text{up}})$, there exists $\mathbf{x} = \begin{pmatrix} \mathbf{x}_F \\ \mathbf{x}_C \end{pmatrix}$ such that

$$\begin{aligned} \begin{pmatrix} \mathbf{b}_F \\ \mathbf{b}_C \end{pmatrix} &= \begin{pmatrix} \mathbf{I} & \\ \mathbf{L}_1^{\text{up}}[C, F] \mathbf{L}_1^{\text{up}}[F, F]^\dagger & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{L}_1^{\text{up}}[F, F] & \\ & \text{Sc}[\mathbf{L}_1^{\text{up}}]_C \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{L}_1^{\text{up}}[F, F]^\dagger \mathbf{L}_1^{\text{up}}[F, C] \\ & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{x}_F \\ \mathbf{x}_C \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{I} & \\ \mathbf{L}_1^{\text{up}}[C, F] \mathbf{L}_1^{\text{up}}[F, F]^\dagger & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{L}_1^{\text{up}}[F, F] & \\ & \text{Sc}[\mathbf{L}_1^{\text{up}}]_C \end{pmatrix} \begin{pmatrix} \mathbf{x}_F + \mathbf{L}_1^{\text{up}}[F, F]^\dagger \mathbf{L}_1^{\text{up}}[F, C] \mathbf{x}_C \\ \mathbf{x}_C \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{I} & \\ \mathbf{L}_1^{\text{up}}[C, F] \mathbf{L}_1^{\text{up}}[F, F]^\dagger & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{L}_1^{\text{up}}[F, F] \mathbf{x}_F + \mathbf{L}_1^{\text{up}}[F, C] \mathbf{x}_C \\ \text{Sc}[\mathbf{L}_1^{\text{up}}]_C \mathbf{x}_C \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{L}_1^{\text{up}}[F, F] \mathbf{x}_F + \mathbf{L}_1^{\text{up}}[F, C] \mathbf{x}_C \\ \mathbf{L}_1^{\text{up}}[C, F] \mathbf{x}_F + \mathbf{L}_1^{\text{up}}[C, F] \mathbf{L}_1^{\text{up}}[F, F]^\dagger \mathbf{L}_1^{\text{up}}[F, C] \mathbf{x}_C + \text{Sc}[\mathbf{L}_1^{\text{up}}]_C \mathbf{x}_C \end{pmatrix}. \end{aligned} \tag{13}$$

Here, the third equality holds since

$$\mathbf{L}_1^{\text{up}}[F, F] \mathbf{L}_1^{\text{up}}[F, F]^\dagger \mathbf{L}_1^{\text{up}}[F, C] = \mathbf{\Pi}_{\text{Im}(\partial_2[F, :])} \mathbf{L}_1^{\text{up}}[F, C] = \mathbf{L}_1^{\text{up}}[F, C],$$

and the fourth equality holds similarly by using symmetry. Thus,

$$\begin{aligned} \mathbf{b}_C &= \mathbf{L}_1^{\text{up}}[C, F] \mathbf{x}_F + \mathbf{L}_1^{\text{up}}[C, F] \mathbf{L}_1^{\text{up}}[F, F]^\dagger (\mathbf{b}_F - \mathbf{L}_1^{\text{up}}[F, F] \mathbf{x}_F) + \text{Sc}[\mathbf{L}_1^{\text{up}}]_C \mathbf{x}_C \\ &= \mathbf{L}_1^{\text{up}}[C, F] \mathbf{x}_F + \mathbf{L}_1^{\text{up}}[C, F] \mathbf{L}_1^{\text{up}}[F, F]^\dagger \mathbf{b}_F - \mathbf{L}_1^{\text{up}}[C, F] \mathbf{x}_F + \text{Sc}[\mathbf{L}_1^{\text{up}}]_C \mathbf{x}_C \\ &= \mathbf{L}_1^{\text{up}}[C, F] \mathbf{L}_1^{\text{up}}[F, F]^\dagger \mathbf{b}_F + \text{Sc}[\mathbf{L}_1^{\text{up}}]_C \mathbf{x}_C. \end{aligned}$$

That is, $\mathbf{h} \in \text{Im}(\text{Sc}[\mathbf{L}_1^{\text{up}}]_C)$.

Next we look at $\mathbf{L}_1^{\text{up}} \tilde{\mathbf{x}}$. Let $\delta = \text{Sc}[\mathbf{L}_1^{\text{up}}]_C \tilde{\mathbf{x}}_C - \mathbf{h}$. We replace \mathbf{x}_F and \mathbf{x}_C in Equation (13) with $\tilde{\mathbf{x}}_F$ and $\tilde{\mathbf{x}}_C$:

$$\begin{aligned} \mathbf{L}_1^{\text{up}} \tilde{\mathbf{x}} &= \begin{pmatrix} \mathbf{I} & \\ \mathbf{L}_1^{\text{up}}[C, F] \mathbf{L}_1^{\text{up}}[F, F]^\dagger & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{L}_1^{\text{up}}[F, F] \tilde{\mathbf{x}}_F + \mathbf{L}_1^{\text{up}}[F, C] \tilde{\mathbf{x}}_C \\ \text{Sc}[\mathbf{L}_1^{\text{up}}]_C \tilde{\mathbf{x}}_C \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{b}_F \\ \mathbf{L}_1^{\text{up}}[C, F] \mathbf{L}_1^{\text{up}}[F, F]^\dagger \mathbf{b}_F + \text{Sc}[\mathbf{L}_1^{\text{up}}]_C \tilde{\mathbf{x}}_C \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{b}_F \\ \mathbf{b}_C + \delta \end{pmatrix}. \end{aligned}$$

Then,

$$\begin{aligned}\|\mathbf{L}_1^{\text{up}}\tilde{\mathbf{x}} - \mathbf{b}\|_2 &= \|\boldsymbol{\delta}\|_2 \leq \delta \|\mathbf{h}\|_2 \leq \delta \left(\|\mathbf{b}_C\|_2 + \left\| \mathbf{L}_1^{\text{up}}[C, F] \mathbf{L}_1^{\text{up}}[F, F]^\dagger \right\|_2 \|\mathbf{b}_F\|_2 \right) \\ &\leq \delta \left(1 + \left\| \mathbf{L}_1^{\text{up}}[C, F] \mathbf{L}_1^{\text{up}}[F, F]^\dagger \right\|_2 \right) \|\mathbf{b}\|_2 \leq \epsilon \|\mathbf{b}\|_2,\end{aligned}$$

where the last inequality is by our setting of δ .

We compute $\tilde{\mathbf{x}}$ by two calls of UPLAPFSOLVER and one call of SCHURSOLVER and $O(1)$ matrix-vector multiplications and vector-vector additions. Thus, the total runtime is $O(t_1(m_F) + t_2(m_C) + \text{the number of nonzeros in } \mathbf{L}_1^{\text{up}})$. \square