Incomplete Nested Dissection

Rasmus Kyng * Simons Institute
Richard Peng † Georgia Tech
Robert Schwieterman † Georgia Tech
Peng Zhang † Georgia Tech

May 25, 2018

Abstract

We present an asymptotically faster algorithm for solving linear systems in well-structured 3-dimensional truss stiffness matrices. These linear systems arise from linear elasticity problems, and can be viewed as extensions of graph Laplacians into higher dimensions. Faster solvers for the 2-D variants of such systems have been studied using generalizations of tools for solving graph Laplacians [Daitch-Spielman CSC'07, Shklarski-Toledo SIMAX'08].

Given a 3-dimensional truss over n vertices which is formed from a union of k convex structures (tetrahedral meshes) with bounded aspect ratios, whose individual tetrahedrons are also in some sense well-conditioned, our algorithm solves a linear system in the associated stiffness matrix up to accuracy ϵ in time $O(k^{1/3}n^{5/3}\log(1/\epsilon))$. This asymptotically improves the running time $O(n^2)$ by Nested Dissection for all $k \ll n$.

We also give a result that improves on Nested Dissection even when we allow any aspect ratio for each of the k convex structures (but we still require well-conditioned individual tetrahedrons). In this regime, we improve on Nested Dissection for $k \ll n^{1/44}$.

The key idea of our algorithm is to combine nested dissection and support theory. Both of these techniques for solving linear systems are well studied, but usually separately. Our algorithm decomposes a 3-dimensional truss into separate and balanced regions with small boundaries. We then bound the spectrum of each such region separately, and utilize such bounds to obtain improved algorithms by preconditioning with partial states of separator-based Gaussian elimination.

*email: rjkyng@gmail.com. This work was done in part while the author was visiting the Simons Institute for the Theory of Computing. It was partially supported by the DIMACS/Simons Collaboration on Bridging Continuous and Discrete Optimization through NSF grant #CCF-1740425.

†emails: rpeng@cc.gatech.edu, {rschwieterman3,pzhang60}@gatech.edu. This material is based on work supported by the National Science Foundation under Grant No. 1637566.

1 Introduction

Linear systems in truss stiffness matrices arise from linear elasticity problems for simulating the effect of forces on geometrically embedded objects [ST08, DS07]. A truss is an undirected weighted graph over n vertices which are points embedded in d dimensions. We refer to the weights as *stiffness coefficients*. The associated truss stiffness matrix is a $dn \times dn$ matrix, which can be written as a sum of rank-one positive semi-definite matrices such that each matrix corresponds to an edge in the truss graph.

Graph Laplacian matrices can be viewed as a special case of truss stiffness matrices, in which each vertex is embedded in 1 dimension. 2-dimensional variants have been studied in [ST08] and [DS07]. In this paper, we solve linear systems in truss stiffness matrices of 3-dimensional meshes. These meshes commonly arise from applying finite element methods [SF73] to simulating 3-dimensional physical space in scientific computing and numerical analysis.

Unlike linear systems in graph Laplacians for which nearly-linear time solvers exist (see for example [ST14, KMP10, KMP11, PS14, CKM$^+$14, KS16], etc.), solving linear systems in general truss stiffness matrices can be as hard as solving linear systems over the reals [KZ17], even for 2-dimensional non-planar trusses.

Throughout this paper, we are interested in 3-dimensional trusses with additional geometric structures. Specifically, we assume a truss is a tetrahedral mesh, which can be decomposed into several convex simplicial complexes. In addition, each edge has bounded length and stiffness coefficient, and each tetrahedron has bounded aspect ratio. The aspect ratio of a tetrahedron is the ratio between its volume and the cubic of its diameter. Bounded aspect ratio of simplices is a common criterion in mesh generation [BEG94, Che89, MV92, Rup93].

Existing works which use geometric structures to speed up algorithms are mostly for planar graphs (e.g., [Fre87, Goo95, HKRS97, BKM$^+$11a]). A common structure underlying these speedups is the r-division. An r-division divides a graph into balanced pieces in which only few edges are between any two pieces [Fre87, Goo95, KMS13]. Then divide-and-conquer can be applied. Usually one layer of r-division is sufficient for running time speedups [BKM$^+$11a, ŁS11, BENWN14], while recursive r-divisions lead to even faster algorithms [HKRS97, BKM$^+$11b, KMS13]. Designing efficient algorithms for 3-dimensional graphs is much harder than in 2-dimensions, even for graphs with bounded genus surface embedded graphs [EN11, BENWN14, BCFN16].

Existing linear system solvers for 3-dimensional tetrahedral meshes are mainly based on nested dissection, which produces a vertex ordering for sparse Gaussian elimination [Geo73, LRT79, MT90, AY10]. The main idea of nested dissection is to recursively compute small separators, which recursively partition a (sub)graph into separate and balanced pieces. Due to small overlaps between any two pieces, Gaussian elimination with this ordering introduces only a small fill-in size. The study of nested dissection, and the prevalence of linear systems in 3-D complexes in turn motivated the study of small separators for 3-dimensional meshes [MT90, MTTV98]. However, the existence of 3-D complexes with no sub-linear sized constant-fraction separators [MT90] means that such algorithms critically depend on the aspect ratios of the individual tetrahedrons.

In addition, support theory has been widely used for solving linear systems in graph Laplacian matrices [ST14, CKM$^+$14] and generalized Laplacian matrices (e.g., connection Laplacians [KLP$^+$16]). Instead of directly solving the original linear system, support theory seeks a sparse preconditioner and iteratively solves a sequence of linear systems (refer to a survey [Axe85]). Both nested dissection and support theory are well-studied techniques in the literature of solving linear systems, but usually they are applied separately.

We design a new approach for solving linear systems in 3-D truss stiffness matrices, by combining the ideas of nested dissection and support theory. We show that the $O(n^2)$-time[1] bound for solving systems on simplicial complexes with bounded aspect ratios [MT90] can be further improved for truss matrices on simplicial complexes formed as a union of k convex tetrahedral meshes.

Theorem 1.1 (Informal statement)**.** *Given a linear system in the stiffness matrix of a 3-D truss $\mathcal{T}$ over n vertices satisfying:*

1. *$\mathcal{T}$ is a mesh of tetrahedrons formed by a union of k convex simplicial complexes with constant aspect ratio each,*

2. *each edge of $\mathcal{T}$ has constant length and stiffness coefficient,*

3. *each tetrahedron of $\mathcal{T}$ has constant aspect ratio,*

and an error parameter $\epsilon > 0$, there is an algorithm which outputs a solution of the linear system up to accuracy ϵ in time $O(k^{1/3}n^{5/3}\log(1/\epsilon))$.

Theorem 1.1 improves on Nested Dissection for all $k \ll n$. We also show a second result that improves on Nested Dissection even when we allow any aspect ratio for each of the k convex structures (but we still require individual tetrahedrons to have bounded aspect ratio and size). In this regime, we improve on Nested Dissection provided $k \ll n^{1/44}$.

Our requirements for trusses have more restrictions than the previous nested dissection based algorithms [MT90], which only requires constant aspect ratios for individual tetrahedrons. Our additional restrictions are mainly due to the need to derive spectral bounds for the preconditioner. Assumptions such as bounded edge lengths and stiffness coefficients (in addition to aspect ratios) are also present in previous work by Daitch and Spielman [DS07], which forms the starting point for our key technical results on eigenvalues of 3-D simplicial complexes. Furthermore, the need to maintain null spaces as well as truss structures in intermediate steps of Gaussian elimination places additional requirements on the mesh structure.

Our algorithm can be viewed as incomplete nested dissection. Specifically, for each convex simplicial complex, we compute an r-division whose boundaries have nice structures. We then eliminate all interior vertices of the r-division, which gives a partial state of Gaussian elimination. To solve the remaining linear system, we simply use the r-division boundaries to precondition the intermediate submatrix of the Gaussian elimination as an incomplete Cholesky factorization. Our key technical result shows that the associated stiffness matrix of carefully chosen boundaries has bounded condition number, which may be of independent interest.

This algorithm only relies on the geometric structures of the underlying graphs and good eigenvalue bounds of the associated matrices. Our presentation only represents a basic instantiation of utilizing both geometric structures and spectral ideas to solve linear systems. It is likely that the same techniques can be extended to more linear systems.

Finally, due to reliance on numerical as well as combinatorial structures, our result, as well as previous works on faster linear system solvers for truss matrices [ST08, DS07], are limited to meshes with both bounded side lengths and aspect ratios. This is significantly more restrictive than nested dissection based algorithms that only depend on the aspect ratios of the truss elements. We believe in future works it would be useful to extend our approach to more general cases, and more systematically study the necessity of restricting to such special cases.

Algorithm	D	Global Req.	Local Req.	Runtime
Gaussian Elimination	any	any	none	$O(n^3)$
Fast Inversion [LG14]	any	any	none	$O(n^\omega)$
Nested Dissection [LRT79]	2	any	none	$O(n^{\omega/2})$
Nested Dissection [MT90]	3	any	aspect ratio (AR)	$O(n^{2\omega/3})$
Fretsaw Extension [ST08]	2	any	none	unspecified
Augmented Tree [DS07]	2	stiffly-connected	size, AR, stiffness coefs.	about $n^{5/4}\log(1/\epsilon)$
Theorem 3.1	3	k × (convex & AR)	size, AR, stiffness coefs.	$O(k^{1/3}n^{5/3}\log(1/\epsilon))$
Theorem 3.2	3	k × convex	size, AR, stiffness coefs.	$O(k^{22/3}n^{11/6}\log(1/\epsilon))$

Table 1: Comparisons of algorithms for solving linear systems in stiffness matrices of trusses. Global Req. / Local Req. refer to the restrictions / assumptions made by these algorithms on the overall truss complex and individual truss elements respectively. Here ω is the matrix multiplication exponent, which by [LG14] is < 2.3728639. The running time of [DS07] also ignores an overhead term of $O(\log^{3/2} n \log\log^{3/4} n)$ related to the qualities of tree embeddings.

1.1 Related Works

A comparison of related results and their geometric requirements are given in Table 1. Many geometric restrictions are crucial to designing fast linear system solvers. In addition to requirements on individual truss elements (e.g., bounded aspect ratios, edge lengths and stiffness coefficients), a truss being *stiffly connected*, defined in [DS07], is important as a global requirement. From a physical view, any deformation of a stiffly connected truss, except a translation and / or a rotation, requires energy. From an algebraic view, the associated stiffness matrix of a stiffly connected truss only has a trivial null space. In our algorithm, we always guarantee that our preconditioner of a convex truss is stiffly connected.

Both direct methods (mainly Gaussian elimination, refer to [Geo73, LRT79, MT90, GNP94, LG14]) and iterative methods (e.g., conjugate gradient, multigrid methods, etc., refer to [Fed64, Axe85, Saa03]) have been widely studied for solving linear systems with additional structures. Combining these two methods has led to fast linear system solvers in graph Laplacians and its generalizations (e.g., [ST14, KMP10, KMP11, PS14, KLP$^+$16, CKP$^+$17, KS16] etc.).

One key idea among these algorithms is incomplete Cholesky factorization. It was used in [KS16] to accelerate Gaussian elimination for graph Laplacian linear systems. Each time the algorithm eliminates a vertex, it sparsifies the partial Cholesky factorization by randomly picking a subset of its nonzeros. In addition, sparsified block Cholesky factorization and multigrid methods were applied in [KLP$^+$16] to solving connection Laplacian linear systems. In [KLP$^+$16], they eliminate a subset of vertices, sparsify its Schur complement (which is an intermediate submatrix of Gaussian elimination, refer to Definition 2.4 for a formal definition) without explicitly computing it, and then repeat this process recursively.

The analyses of both these two papers crucially rely on the fact that graph Laplacian matrices and connection Laplacian matrices are closed under taking the Schur complement. However, this fact does not hold for truss stiffness matrices, for which the Schur complements can be essentially any PSD matrices [KZ17]. The hard instance given in [KZ17] is a 2D non-planar truss. In this paper, instead, we utilize the geometric structures, balanced divisions, and tetrahedral meshes whose tetrahedrons form simplicial complexes and have bounded size and aspect ratio. In this

[1] Here we assume that multiplying two $n \times n$ matrices needs time $O(n^3)$, following the convention of nested dissection literature [Geo73, LRT79, GNP94]. We will discuss the running time by using fast matrix multiplication in Appendix B.

setting we can approximate Cholesky factorization.

One way of measuring the quality of a sparse approximated Cholesky factorization is the relative condition number of the sparse matrix and the original matrix, which determines the number of iterations of preconditioned iterative algorithms. Cheeger's inequality provides tight bounds on the smallest nonzero eigenvalues of graph Laplacian matrices through sparse cuts in graphs [Chu96]. Extensions to higher-order eigenvalues, connection Laplacian matrices, and simplicial complexes can be found in [LGT14, BSS13, SKM14, PRT16], etc. However, it is unclear whether a Cheeger-type inequality exists for truss stiffness matrices.

1.2 Organization of the Remaining Paper

In Section 2, we give definitions and notations on graphs and linear algebra, and formally define truss stiffness matrices. In Section 3, we present our algorithm for solving linear systems in 3-dimensional trusses and prove our main theorems. Sections 4 and 5 bound eigenvalues of a well-structured truss stiffness matrix which is the key lemma for proving our main results. In Section 6 we show our algorithm for constructing preconditioners, and in Section 7 we show our nested dissection based algorithms for solving linear systems in these preconditioners. Appendix A gives some useful facts on Schur complements, and Appendix B gives our running times parameterized by ω.

2 Preliminaries

2.1 Tetrahedral Meshes

For a subset $S \subset \mathbb{R}^3$, we define the *diameter* of S to be the maximum Euclidean distance between any pair of points in S. We define the *aspect ratio* of S to be the ratio between the radius of the smallest ball containing S and the radius of the largest ball inscribed in S. In mesh generation, aspect ratio is a common criterion for individual elements (see for example [BEG94, Che89, MT90, MV92, Rup93]).

A tetrahedron is the convex hull of four non-coplanar points in $\mathbb{R}^3$. We will specify tetrahedrons in terms of sets of four such points. We refer to a set of tetrahedrons as a tetrahedral mesh.

We say a tetrahedral mesh is a *simplicial complex* if the intersection of every two tetrahedrons is either empty or a face of both two tetrahedrons. A simplicial complex is *convex* if the union of the images of its simplices is convex, as defined in [CFM$^+$14].

Definition 2.1. A tetrahedral mesh is said to be *simple* iff it is a simplicial complex and every tetrahedron has bounded aspect ratio.

The following definition characterizes rigidity and stiffness of a tetrahedral mesh, which is an adaption of Definition 2.3 in [DS07].

Definition 2.2. The *rigidity graph* of a tetrahedral mesh is a graph whose vertices correspond to tetrahedrons and whose edges connect any two tetrahedrons sharing a triangle face. A tetrahedral mesh is *stiffly-connected* iff (1) its rigidity graph is connected, and (2) for any vertex in the mesh, the rigidity subgraph induced on the tetrahedrons containing this vertex is connected.

We define a *bounding box* of a convex 3D shape to be a 3D box such that: (1) the box contains all points of this shape, and (2) the volume of the box is same as the volume of this shape up to a constant factor. [BHP01] gives a linear time algorithm which computes a bounding box of a convex shape in 3 dimensions.

Lemma 2.3. *(Lemma 3.6 of [BHP01]) Given a 3D convex shape, one can compute in linear time a bounding box of this shape.*[2] *Moreover, the aspect ratio of the bounding box is same as the aspect ratio of the given shape up to a constant factor.*

2.2 Vectors and Matrices

Given a vector $\boldsymbol{x} \in \mathbb{R}^n$, for $1 \leq i < i+j \leq n$, we denote $\boldsymbol{x}_i$ the ith entry of $\boldsymbol{x}$, and we denote $\boldsymbol{x}_{i:i+j}$ the subvector whose entries are $\boldsymbol{x}_i, \boldsymbol{x}_{i+1}, \ldots, \boldsymbol{x}_{i+j}$. The Euclidean norm of $\boldsymbol{x}$ is defined as $\|\boldsymbol{x}\|_2 \stackrel{\text{def}}{=} \sum_{1 \leq i \leq n} \boldsymbol{x}_i^2$.

Given a matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$, for $1 \leq i, j \leq n$, we denote $\boldsymbol{A}_{ij}$ the (i,j)th entry of $\boldsymbol{A}$. A square matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is a *positive semi-definite matrix* (PSD) iff for every vector $\boldsymbol{x} \in \mathbb{R}^n$ we have $\boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{x} \geq 0$. We denote $\lambda_{\min}(\boldsymbol{A})$ the smallest *nonzero* eigenvalue of $\boldsymbol{A}$.

For two symmetric matrices $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{n \times n}$, we say $\boldsymbol{A} \succcurlyeq \boldsymbol{B}$ iff $\boldsymbol{A} - \boldsymbol{B}$ is PSD. We define the *condition number* of $\boldsymbol{A}$ relative to $\boldsymbol{B}$, denoted by $\kappa(\boldsymbol{A}, \boldsymbol{B})$, to be

$$\min \left\{ \frac{\lambda_{\max}}{\lambda_{\min}} : \lambda_{\min} \cdot \boldsymbol{B} \preccurlyeq \boldsymbol{A} \preccurlyeq \lambda_{\max} \cdot \boldsymbol{B} \right\}.$$

In addition, we define Schur complements which arise from the process of Gaussian elimination.

Definition 2.4 (Schur complement)**.** Let S, T be a partition of the indices of a square matrix $\boldsymbol{A}$ so that $\boldsymbol{A} = \begin{pmatrix} \boldsymbol{A}_{SS} & \boldsymbol{A}_{ST} \\ \boldsymbol{A}_{ST}^\top & \boldsymbol{A}_{TT} \end{pmatrix}$ where $\boldsymbol{A}_{SS}, \boldsymbol{A}_{ST}, \boldsymbol{A}_{TT}$ are block matrices, the *Schur complement* of $\boldsymbol{A}$ onto T is

$$\textsc{Sc}[\boldsymbol{A}]_T \stackrel{\text{def}}{=} \boldsymbol{A}_{TT} - \boldsymbol{A}_{ST}^\top \boldsymbol{A}_{SS}^{-1} \boldsymbol{A}_{ST}.$$

We will use the following fact of Schur complements.

Fact 2.5 (Lemma 4 of [RTL76])**.** *Let* $\boldsymbol{A} = \begin{pmatrix} \boldsymbol{A}_{SS} & \boldsymbol{A}_{ST} \\ \boldsymbol{A}_{ST}^\top & \boldsymbol{A}_{TT} \end{pmatrix}$ *be a symmetric matrix, the* (i,j)*th entry of the Schur complement* $\textsc{Sc}[\boldsymbol{A}]_T \neq 0$ *only if there exists a sequence of indices* $k_1, \ldots, k_l \in S$ *such that all* $\boldsymbol{A}_{ik_1}, \boldsymbol{A}_{k_1 k_2}, \ldots, \boldsymbol{A}_{k_{l-1} k_l}, \boldsymbol{A}_{k_l, j}$ *are nonzero.*

Fact 2.6. *Let* $\boldsymbol{A}$ *be a symmetric PSD matrix and* $\textsc{Sc}[\boldsymbol{A}]_T$ *be its Schur complement.*

1. $\textsc{Sc}[\boldsymbol{A}]_T$ *is a symmetric PSD matrix.*
2. $\lambda_{\max}(\textsc{Sc}[\boldsymbol{A}]_T) \leq \lambda_{\max}(\boldsymbol{A})$.

We prove Fact 2.6 in Appendix A.

2.3 Solving Linear Systems

Our algorithm combines two of the most important tools for solving linear systems: Nested dissection and preconditioning. Below, we give a brief introduction to some of the central results on these techniques.

Classic results due to Lipton, Rose, and Tarjan [LRT79], and Miller and Thurston [MT90] combine to show that linear systems arising from simple tetrahedral meshes (see Definition 2.1) can be solved in $O(n^2)$ time. These results concern linear equations in an $n \times n$ matrix $\boldsymbol{A}$ where the

[2] The lemma statement in [BHP01] gives a bound related to the minimum-volume bounding box B^* of the input shape, but their proof uses the volume of the shape as a lower bound of the volume of B^*.

indices $\{1, \ldots, n\}$ can be embedded as points $\{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n\}$ that form the vertices of an explicitly given, simple tetrahedral mesh, and $\boldsymbol{A}_{ij}$ is non-zero only if the vertices i and j share an edge in the tetrahedral mesh.

Theorem 2.7 (Nested dissection [MT90])**.** *Let $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ be a symmetric matrix defined on a simple tetrahedral mesh. A Cholesky factorization $\boldsymbol{A} = \boldsymbol{P}\boldsymbol{L}\boldsymbol{L}^\top\boldsymbol{P}^\top$ can be computed in time $O(n^2)$, in which $\boldsymbol{P}$ is a permutation matrix and $\boldsymbol{L}$ is a lower triangular matrix with $O(n^{4/3})$ nonzero entries. As a result, a linear system in $\boldsymbol{A}$ can be solved in time $O(n^2)$ by Gaussian elimination.*

Theorem 2.7 can be extended to a block matrix $\boldsymbol{A} \in \mathbb{R}^{cn\times cn}$ where c is a constant positive integer. Each vertex of the underlying graph corresponds to c indices of $\boldsymbol{A}$. In addition, the block corresponding to the column indices for vertex i and the row indices for j should be non-zero only if the vertices i and j share an edge in the tetrahedral mesh, or if $i = j$, i.e. when the block is on the diagonal.

The Nested Dissection algorithm relies on invoking *separators* recursively. A separator is a set of indices S such that the remaining indices $[n] \setminus S$ can be partitioned into two sets B and C such that every entry with $i \in B$ and $j \in C$ has $\boldsymbol{A}_{ij} = 0$. Furthermore, we guarantee that the partition is roughly balanced, for example, each of B and C contains no more than $\frac{3}{4} \cdot n$ indices. Nested Dissection recursively repeats the partitioning process on the union of each subset and the separator itself, that is, $B \cup S$ and $C \cup S$. Given such a recursive partition scheme, we reorder the indices of the matrix so that the indices in the separator S are eliminated *last*, and we then order the indices in B and C recursively in a similar way. We perform Gaussian elimination on the matrix according to this ordering, which only introduces a small fill-in size and few multiplication counts. This approach also works for eliminating a subset of the variables, resulting in a Schur complement on the rest.

Both the running time and representation cost of nested dissection algorithms are bottlenecked by the costs of the top-level separators. In Algorithm 1 TrussSolver, we will utilize improved running time bounds for nested dissection when better separators exist. The following lemma characterizes the performance of Nested Dissection given better top-level separators.

Lemma 2.8. *Suppose we have a recursive separator decomposition of a simplicial complex with n bounded aspect ratio tetrahedrons such that:*

1. *the number of leaves, and hence total number of recursive calls, is at most n^α.*
2. *each leaf (bottom layer partition) has at most n^β tetrahedrons.*
3. *each top separator has size at most n^γ.*

Then we can find an exact Cholesky factorization of the associated stiffness matrix in time $O(n^{\alpha+2\beta} + n^{\alpha+3\gamma})$, and the total resulting fill-in is $O(n^{\alpha+\frac{4}{3}\beta} + n^{\alpha+2\gamma})$.

We give a proof of the above lemma in Section 7.1, which is an adaption of the analysis in [LRT79] We remark that the algorithmic realization of this can be viewed as utilizing the nested dissection algorithm 2.7 to complete this structure into a full separator tree.

Last but not least, we state the following theorem for preconditioned conjugate gradient, which will be used in bounding the running time of our algorithm.

Theorem 2.9 (Preconditioned conjugate gradient [Axe85])**.** *Let $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{n\times n}$ be two symmetric positive semidefinite matrices and let $\boldsymbol{b} \in \mathbb{R}^n$. Each iteration of the preconditioned conjugate gradient multiplies one vector by $\boldsymbol{A}$, solves one linear system in $\boldsymbol{B}$, and performs a constant number of vector additions. For any $\epsilon > 0$, the algorithm outputs an $\boldsymbol{x}$ satisfying $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2 \leq \epsilon \|\boldsymbol{b}\|_2$ in $O(\sqrt{\kappa(\boldsymbol{A}, \boldsymbol{B})} \log(1/\epsilon))$ such iterations.*

We remark that while there are settings where the convergence of preconditioned conjugate gradient is numerically unstable, the eigenvalue-based bound that we utilize here is stable once the solves involving $\boldsymbol{B}$ have polynomially small errors.

2.4 Truss Stiffness Matrices

We extend the definition of 2-dimensional truss stiffness matrices from [DS07] (see Definition 2.1 and 2.2) to 3 dimensions.

Definition 2.10 (3-dimensional truss)**.** *3-dimensional truss* $\mathcal{T} = \langle V, \{\boldsymbol{p}_i\}_{i\in V}, T, E, \gamma\rangle$ is given by

- A set of n vertices V embedded at distinct points $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n \in \mathbb{R}^3$
- A mesh (i.e. set) of tetrahedrons $T = \{t_1, t_2, \ldots\}$, each specified in terms of four vertices, i.e. we identify tetrahedron t_i with both four vertices $\{a_i, b_i, c_i, d_i\} \subseteq V$ and the convex hull of $\boldsymbol{p}_{a_i}, \boldsymbol{p}_{b_i}, \boldsymbol{p}_{c_i}, \boldsymbol{p}_{d_i}$.
- A set of edges E which is exactly the set of pairs of vertices that appear in some tetrahedron together. Each edge $e = (i, j) \in E$ represents a straight idealized bar between vertex points $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$.
- A function $\gamma : E \to \mathbb{R}_+$, which assigns a stiffness coefficient $\gamma(e)$ to each edge e. The stiffness coefficient represents the stiffness of the idealized bar corresponding to edge e.

Definition 2.11 (Truss stiffness matrix)**.** Let $\mathcal{T} = \langle V, \{\boldsymbol{p}_i\}_{i\in V}, T, E, \gamma\rangle$ be a 3-dimensional truss. For each edge $e = (i, j) \in E$, we define an edge vector $\boldsymbol{b}^{(e)} \in \mathbb{R}^{3n}$ with 6 nonzero entries:

$$\boldsymbol{b}^{(e)}_{3i-2:3i} = -\boldsymbol{b}^{(e)}_{3j-2:3j} = \frac{\boldsymbol{p}_i - \boldsymbol{p}_j}{\left\|\boldsymbol{p}_i - \boldsymbol{p}_j\right\|_2}.$$

The *stiffness matrix* of the truss $\mathcal{T}$ is defined as

$$\boldsymbol{A}_{\mathcal{T}} \stackrel{\text{def}}{=} \sum_{e=(i,j)\in E} \frac{\gamma(e)}{\left\|\boldsymbol{p}_i - \boldsymbol{p}_j\right\|_2} \boldsymbol{b}^{(e)} \boldsymbol{b}^{(e)\top}.$$

In general, solving linear systems in truss stiffness matrices can be as hard as solving linear systems in real matrices [KZ17].

In this paper, we study 3D trusses with some additional geometric structures. These structures enable us to design linear system solvers that run much faster than solvers for general simple tetrahedral meshes.

Definition 2.12. We say a 3D truss is *edge-simple* if its tetrahedral mesh is simple and every tetrahedron has bounded edge lengths and stiffness coefficients (i.e. both are bounded above and below by constants).

Definition 2.13. We say a 3D truss is *convex edge-simple* if it is edge-simple, and its tetrahedral mesh is convex.

3 Algorithm Overview

In this section, we present our algorithm for solving linear systems in stiffness matrices of edge-simple 3D trusses. Our first main result concerns trusses formed by combining k convex edge-simple trusses, each with constant aspect ratio upper bounded by some arbitrarily large but fixed constant.

Theorem 3.1. *Given an edge-simple 3-D truss with n vertices, formed from a union of k convex edge-simple trusses each with aspect ratio at most $O(1)$[3], and an error parameter $\epsilon > 0$, there is an algorithm which solves a linear system in the corresponding stiffness matrix up to accuracy ϵ in time $O(k^{1/3}n^{5/3}\log(1/\epsilon))$.*

This theorem is appealing because in many modeling applications, only large constant aspect ratios are needed for individual convex parts that are being combined. In Theorem 3.1, we see that the performance degrades smoothly towards the $O(n^2)$ running time of Nested Dissection as k approaches n.

Our second main result deals with the case when we allow a truss formed from k convex edge-simple trusses, each of which may have arbitrarily large aspect ratio.

Theorem 3.2. *Given an edge-simple 3-D truss with n vertices, formed from a union of k convex edge-simple trusses, and an error parameter $\epsilon > 0$, there is an algorithm which solves a linear system in the corresponding stiffness matrix up to accuracy ϵ in time $O(n^{11/6}k^{22/3}\log(1/\epsilon))$.*

We remark that the geometric assumptions of Theorem 3.2 are in many ways fairly weak. We need the individual truss tetrahedrons to have small aspect ratio, but each of the k convex edge-simple trusses may overall have a wide range of shapes: It can form a ball, a pancake, or even a very long beam with arbitrarily large aspect ratio. The dependence on k is fairly bad and has not been carefully optimized, meaning currently that only about $k \ll n^{1/44} \approx n^{0.0227}$ convex edge-simple trusses can be combined while still achieving a speed-up over nested dissection. Even so, this allows the construction of some shapes with genus up to $n^{1/44}$.

Fast matrix multiplication can be used in our algorithm, as well as earlier routines. In accordance with previous works on nested dissection, and to simplify presentation, we assume $\omega = 3$ (the matrix multiplication constant) throughout our calculations. However, in Appendix B, we also give the ω-dependent bounds. Assuming $\omega = 2.3728639$ as in [LG14], the bounded aspect ratio case from Theorem 3.1 takes time $O(k^{0.1210452}n^{1.4608641}\log(1/\epsilon))$, while the arbitrary aspect ratio case from Theorem 3.2 takes time $O(k^{5.7115596}n^{1.5175803}\log(1/\epsilon))$. In both cases the running times are less than the $O(n^{1.5819093})$ bound obtained by plugging $\omega = 2.3728639$ into 3-D nested dissection.

Our algorithm pseudocode is stated in Algorithm 1. Note this algorithm proves both Theorem 3.1 and 3.2. Theorem 3.1 is a special case where $\mathcal{I} = [k]$ in line 4.

3.1 Main Ideas

Both our main theorems are based on speeding up Nested Dissection by combining it with preconditioning and iterative solvers. In Section 2.3 we gave a brief outline of Nested Dissection. In classical Nested Dissection, the main bottleneck that constrains the running time of the algorithm is the process of applying Gaussian elimination to the few separators in the top levels of the separator tree, after having eliminated all the matrix indices at lower levels. The intermediate matrix that arises during Nested Dissection after eliminating the indices at lower levels is in fact the Schur complement onto the separators at top levels.

[3] A slightly modified analysis extends this result to allow each individual truss has aspect ratio $O(n_i^{1/4})$, where n_i is the number of vertices of the ith individual truss.

Our central idea that gives us an advantage over Nested Dissection is that: the outer boundary of tetrahedrons of a single convex edge-simple truss[4] with small aspect ratio is a good preconditioner for the Schur complement of the whole truss onto the boundary.

By forcing Nested Dissection to use these outer boundaries of individual convex edge-simple trusses as the top-level separator, we get a separator which has a good sparse preconditioner. Fortunately, we can also ensure that this top-level separator has a small size.

The phenomenon that the boundary itself is a good preconditioner for the Schur complement onto the boundary has a natural interpretation based on structural mechanics. The quadratic form associated with the Schur complement corresponds to the energy associated with deforming the whole truss by squishing or stretching the boundary vertices while leaving the interior intact and finding the positions of the interior vertices that minimize the overall energy. We show that this energy is not much more than the energy that arises from applying the same deformation to just the boundary tetrahedrons after deleting the interior vertices.

This means that we can speed up the process of applying the inverse of the Schur complement onto the boundary, which is the bottleneck of nested dissection. We avoid directly inverting the Schur complement by instead solving a linear system in the Schur complement using the boundary itself as a preconditioner in preconditioned conjugate gradient. Because the boundary is much sparser than the Schur complement, and has much fewer tetrahedrons than the initial mesh, we significantly reduce the running time.

How good a preconditioner the boundary is for the Schur complement depends on the number of tetrahedrons in the initial convex mesh. If the mesh is large, then the preconditioner is worse in the sense that solving an associated linear system is time-consuming, but the gain from hollowing out the mesh is relatively larger, because the number of tetrahedrons on the surface is relatively smaller. Ideally, we want to balance these two phenomena against each other. We can ensure a good trade-off between these two effects by first dividing very large convex trusses into smaller chunks before hollowing out these chunks and using the resulting boundaries, which now look somewhat like a Swiss cheese, as a preconditioner.

All together, this approach of partitioning, hollowing out, preconditioning, and using Nested Dissection gives us Theorem 3.1.

If the aspect ratios of individual convex trusses are allowed to be extremely large, so that the trusses can be very thin, then we cannot gain much by hollowing out the trusses. However, if an individual convex truss has large aspect ratio, then we can get good separators for Nested Dissection by slicing the truss along its longest dimension. Combining this observation with our preconditioning approach, we are able to obtain Theorem 3.2, which has no requirements on the aspect ratios of each of the k trusses we are combining. Unfortunately, leveraging both the preconditioning behavior and the existence of good separators for individual large aspect ratio trusses requires fairly technical work, which currently introduces a bad dependence on k in this version of our main result.

3.2 Bounding Eigenvalues of an Edge-Simple and Stiffly-Connected Truss

Our main structural results are bounds on the condition number of the stiffness matrix of a edge-simple and stiffly connected (see Definition 2.2) simplicial complex. As each vertex is involved in at most a constant number of tetrahedrons, we can easily obtain bounds on the maximum eigenvalue.

Thus, our main technical contribution is a lower bound on the minimum non-zero eigenvalue of the stiffness matrix of a 3D edge-simple truss. We state the explicit bound in Lemma 3.3. Such a bound is analogous to the bound on minimum eigenvalues of a path in a graph.

[4] It can be a subset of one of the k input individual convex edge-simpletrusses.

Algorithm 1 TrussSolver($\mathcal{T} = \langle V, \{\boldsymbol{p}_i\}_{i \in V}, T, E, \gamma \rangle, \boldsymbol{f}, \epsilon, c_\alpha, c_r$)

Input: a 3D truss $\mathcal{T} = \langle V, \{\boldsymbol{p}_i\}_{i \in V}, T, E, \gamma \rangle$ with n vertices, which is a union of k convex edge-simpletrusses $\mathcal{T}_1, \ldots, \mathcal{T}_k$, a vector $\boldsymbol{f} \in \mathbb{R}^{3n}$, an error parameter $\epsilon > 0$.
Constants for aspect ratio threshold $0 < c_\alpha < 1$, and hollowing rate $0 < c_r < 1$.

Output: an approximate solution $\boldsymbol{x}$ such that $\|\boldsymbol{A}_{\mathcal{T}}\boldsymbol{x} - \boldsymbol{f}\|_2 \le \epsilon \|\boldsymbol{f}\|_2$.

```
for each i do
    Compute a bounding box B_i of T_i, via Lemma 2.3.
end for
Let I = {1 ≤ i ≤ k : α(T_i) ≤ n_i^{c_α}}.
for each i ∈ I do
    Hollow out the interior vertices of T_i with parameter r_i = n_i^{c_r} to form H_i.
end for
Run nested dissection on the preconditioner (possibly with a specific set of separators).
Run preconditioned conjugate gradient with this preconditioner to solve the overall system.
return the solution x.
```

Lemma 3.3. *Let $\mathcal{T}$ be an edge-simple and stiffly-connected 3D truss. Let n be the number of vertices of $\mathcal{T}$ and Δ be the diameter. Let $\boldsymbol{M}$ denote the associated stiffness matrix. Then, $\lambda_{\min}(\boldsymbol{M}) = \Omega(n^{-1}\Delta^{-4})$ and rank$(\boldsymbol{M}) = 3n - 6$.*

Our proof is heavily motivated by the Path Lemma by Daitch and Spielman [DS07], which bounds the minimum non-zero eigenvalue of the stiffness matrix of a path of triangles in 2D. We start by shifting all the tetrahedrons by the normals w.r.t. a particular centering tetrahedron, which in effect projects away the coordinates from the null space. Then we lower bound the minimum dot-product of a quadratic form in terms of the pairwise ℓ_2 differences among these tetrahedrons. However, our calculations result in an exponential factor loss depending on the "hop distance" between the first and last tetrahedrons. This is due to the accumulation of rotational operators, which we need to treat as matrices instead of simple rotations.

We are not sure whether this exponential increase is simply due to an algebraic artifact in our proof. To circumvent it, we instead show that there exists a particular centering where a pair of close-by tetrahedrons are placed $1/\operatorname{poly}(n)$ apart. This proof relies on analyzing the "average behavior" of all centerings globally. It once again relies on treating the initial rotations and projections as linear operators, and working directly with the singular values of these projection operators.

3.3 Proving the Main Result for Small Aspect Ratio Truss Unions

To accelerate Nested Dissection, we need to find a set of balanced separators that are small and whose Schur complements have good sparse preconditioners.

To build these good separators, motivated by r-divisions, we divide each of the small-aspect ratio trusses in our union of k trusses into smaller chunks, such that the boundary of each chunk is a good preconditioner of the Schur complement onto that boundary. The union of all these boundaries is called a *hollowing*. To create a hollowing, we fix two parameters: a bounding box B that determines the directions of each smaller chunks of the hollowing, and a size parameter r that controls the size of the smaller chunks. We call each smaller chunk as a *region*.

Definition 3.4 ((B, r)-hollowing)**.** Given a convex edge-simple3D truss $\mathcal{T} = \langle V, \{\boldsymbol{p}_i\}_{i \in V}, T, E, \gamma \rangle$, a bounding box B of $\mathcal{T}$, and a parameter $r \le n/\alpha^2$ where α is the aspect ratio of $\mathcal{T}$, a (B, r)-*hollowing* of $\mathcal{T}$ is another edge-simple 3D truss $\mathcal{H} = \langle U, \{\boldsymbol{p}_i\}_{i \in U}, S, F, \gamma' \rangle$ such that, $U \subseteq V$, $S \subseteq T$, and F is the subset of E that arises from edges in S, while γ' is just the restriction of γ to F. I.e. edges maintain the same stiffness factors as in $\mathcal{T}$. Also

1. $\mathcal{H}$ contains $O(nr^{-1/3})$ points. $\mathcal{T} \setminus \mathcal{H}$ consists of $O(nr^{-1})$ disjoint chunks, each of which has $O(r)$ vertices and is incident to $O(r^{2/3})$ vertices of $\mathcal{H}$.
2. for *every* plane P whose normal vector has angle $\theta \in (0, \pi/2)$ with the longest direction of B, the number of tetrahedrons in $\mathcal{H}$ intersected by P is
$$O\left(n^{2/3}\alpha^{-1/3}r^{-1/3}\cos^{-2}\theta\right).$$
3. $\boldsymbol{A}_{\mathcal{H}} \preceq \textsc{Sc}[\boldsymbol{A}_{\mathcal{T}}]_U \preceq O(r^2)\boldsymbol{A}_{\mathcal{H}}$.

The next lemma describes the performance of algorithm Hollow, Algorithm 2 in Section 6, that we use to compute a (B, r)-hollowing of a convex edge-simple truss.

Lemma 3.5. *Given a convex edge-simple 3-dimensional truss $\mathcal{T} = \langle V, \{\boldsymbol{p}_i\}_{i\in V}, T, E, \gamma\rangle$ with n vertices, a bounding box B of $\mathcal{T}$, and a positive integer r such that the aspect ratio of $\mathcal{T}$ is at most $\sqrt{n/r}$, the algorithm* $\textsc{Hollow}(\mathcal{T}, B, r)$ *returns a (B, r)-hollowing $\mathcal{H}$ of $\mathcal{T}$, and runs in time $O(n)$.*

Proof of Theorem 3.1. Let $\mathcal{T}$ be a edge-simple 3-D truss with n vertices, formed from a union of k convex edge-simple trusses, say $\mathcal{T}_1, \ldots, \mathcal{T}_k$, each with aspect ratio at most $O(1)$. For each $1 \leq i \leq k$, let n_i be the number of vertices of $\mathcal{T}_i$, and define $r_i \stackrel{\text{def}}{=} n_i^{1/2}$. In Algorithm 1, for each $\mathcal{T}_i$, we compute a (B_i, r_i)-hollowing, where B_i is a bounding box of $\mathcal{T}_i$. By Lemma 2.3 and 3.5, the total running time here is $O(n)$. In each (B_i, r_i)-hollowing region, we eliminate its interior vertices in total time
$$O\left(\sum_i n_i r_i^{-1} \cdot r_i^2\right) = O\left(n^{3/2}\right).$$
The Schur complement onto the boundaries has
$$O\left(\sum_i n_i r_i^{-1} \cdot (r_i^{2/3})^2\right) = O\left(n^{7/6}\right)$$
nonzeros. We then run preconditioned conjugate gradient (PCG) to solve the linear system in the Schur complement by preconditioning it via the union of the (B_i, r_i)-hollowings, say $\mathcal{T}'$. Note by Jensen's inequality, $\mathcal{T}'$ has size
$$O\left(\sum_i n_i r_i^{-1/3}\right) = O\left(\sum_i n_i^{5/6}\right) = O\left(k^{1/6}n^{5/6}\right).$$

Before running PCG, we compute a Cholesky factorization of $\boldsymbol{A}_{\mathcal{T}'}$ by nested dissection. According to Theorem 2.7, the running time is $O(k^{1/3}n^{5/3})$, and the fill-in size is $O(k^{2/9}n^{10/9})$. By Definition 3.4, the condition number is $O(\max_i r_i^2) = O(n)$. According to Theorem 2.9, the number of PCG iterations is at most $O(n^{1/2}\log(1/\epsilon))$ to output a solution up to accuracy ϵ. In each PCG iteration, we do a matrix-vector multiplication with the Schur complement in time $O(n^{7/6})$, and solve a linear system in $\boldsymbol{A}_{\mathcal{T}'}$ in time $O(k^{2/9}n^{10/9})$. Thus the total running time is $O(k^{1/3}n^{5/3}\log(1/\epsilon))$. $\square$

3.4 Proving the Main Result for All-Aspect Ratio Truss Unions

We extend our result to cover the case when the union of convex edge-simple trusses also include trusses with arbitrarily large aspect ratios. The lemma below shows that large aspect ratio implies the existence of good plane separators, which is proven in Section 7.

Lemma 3.6. *Given a convex edge-simple 3D truss with aspect ratio at least $\alpha > 0$, say, $\mathcal{T} = \langle V, \{\boldsymbol{p}_i\}_{i \in V}, T, E, \gamma \rangle$, and its bounding box B. Let $\boldsymbol{d} \in \mathbb{R}^3$ be a unit vector along the longest direction of B, and let $\boldsymbol{g} \in \mathbb{R}^3$ be a unit vector with $\boldsymbol{d} \cdot \boldsymbol{g} > 0$. Then every plane orthogonal to $\boldsymbol{g}$ intersects at most $O(n^{2/3}\alpha^{-1/3}(\boldsymbol{d} \cdot \boldsymbol{g})^{-1})$ tetrahedrons.*

The above lemma tells us that a single convex edge-simple truss with large aspect ratio has a good plane separator. It turns out that even if we have many such trusses whose longest dimension may point in different directions, and we have hollowed-out trusses from small aspect ratio parts, we can still find a single plane that acts as a reasonably good separator for all of these trusses at the same time. This is captured by the following lemma, which is obtained by instantiating Lemma 7.3 with $c_r = 1/3$ and $l = n^{1/6}$.

Lemma 3.7 (Combining Separators)**.** *Given a edge-simple 3D truss $\mathcal{T} = \langle V, \{\boldsymbol{p}_i\}_{i \in V}, T, E, \gamma \rangle$, which is a union of k convex edge-simple trusses with up to n vertices in total. Let $\mathcal{T}' \subset \mathcal{T}$ be a truss by selectively computing (B_i, r_i)-hollowings of some of the pieces with parameter*

$$r_i \leq n_i^{1/3}.$$

There exists a randomized algorithm which with high probability returns a vertex ordering so that a complete elimination of $\mathcal{T}'$ has size $O(n^{23/18}k^{44/9})$, and takes time $O(n^{11/6}k^{22/3})$ to compute.

The algorithm that achieves Lemma 3.7 is Algorithm 4 ConvexTrussUnionND in Section 7. Given these lemmas, we can now sketch a proof of Theorem 3.2.

Proof of Theorem 3.2. We bound the running time of Algorithm 1 TrussSolver with the preconditioner and nested dissection constructed as per Lemma 3.7.

Since all the hollowings involve pieces with $r_i \leq n_i^{1/3}$, Definition 3.4 gives a bound of $O(n^{2/3})$ on the condition number, and in turn a bound of $O(n^{1/3}\log(1/\epsilon))$ on the number of PCG iterations via Theorem 2.9. Furthermore, similar to the proof of Theorem 3.1, the Schur complement of $\mathcal{T}$ onto the elements of $\mathcal{T}'$ has size $O(n^{10/9})$, and computing them by eliminating all interior vertices of our hollowings takes time $O(n^{4/3})$.

Thus, the total running time of Algorithm 1 is

$$O\left(n^{4/3} + n^{11/6}k^{22/3} + n^{1/3}\log(1/\epsilon) \cdot \left(n^{10/9} + n^{23/18}k^{44/9}\right)\right) = O\left(n^{11/6}k^{22/3}\log(1/\epsilon)\right).$$

□

4 Bounding the Smallest Nonzero Eigenvalue of a edge-simple Truss

In this section, we prove Lemma 3.3, which lower bounds the smallest nonzero eigenvalue of a edge-simple 3D truss. We restate Lemma 3.3 in the following.

Lemma 3.3. *Let $\mathcal{T}$ be an edge-simple and stiffly-connected 3D truss. Let n be the number of vertices of $\mathcal{T}$ and Δ be the diameter. Let $\boldsymbol{M}$ denote the associated stiffness matrix. Then, $\lambda_{\min}(\boldsymbol{M}) = \Omega(n^{-1}\Delta^{-4})$ and rank$(\boldsymbol{M}) = 3n - 6$.*

4.1 Main Ideas

The proof of Lemma 3.3 is an extension of the path support lemma by Daitch and Spielman [DS07]. That proof relies on recentering a vector $\boldsymbol{q}$, which is a unit vector orthogonal to the null space, with respect to the a single face by transforming it along the null space of $\boldsymbol{M}$.

Let $\mathcal{T} = \langle V, \{\boldsymbol{p}_i\}_{i \in V}, T, E, \gamma \rangle$ be a 3D stiffly-connected truss over n vertices. The null space of the stiffness matrix of $\mathcal{T}$ can be characterized as:

1. $\boldsymbol{p}^x$, $\boldsymbol{p}^y$, $\boldsymbol{p}^z \in \mathbb{R}^{3n}$: for each $1 \leq i \leq n$, the corresponding 3-dimensional vector $\boldsymbol{p}_i^x$ ($\boldsymbol{p}_i^y$ and $\boldsymbol{p}_i^z$) has 1 for its x-coordinate (y-coordinate, and z-coordinate, respectively) and 0 for the other two coordinates.

2. $\boldsymbol{p}^{\perp xy}$, $\boldsymbol{p}^{\perp xz}$, $\boldsymbol{p}^{\perp yz} \in \mathbb{R}^{3n}$: fix an arbitrary index $1 \leq c \leq n$, for each $1 \leq i \leq n$:

$$\begin{aligned} \boldsymbol{p}_i^{\perp xy} &= \left[-\left(\boldsymbol{p}_i - \boldsymbol{p}_c\right)_y, \left(\boldsymbol{p}_i - \boldsymbol{p}_c\right)_x, 0 \right]^\top, \\ \boldsymbol{p}_i^{\perp xz} &= \left[-\left(\boldsymbol{p}_i - \boldsymbol{p}_c\right)_z, 0, \left(\boldsymbol{p}_i - \boldsymbol{p}_c\right)_x \right]^\top, \\ \boldsymbol{p}_i^{\perp yz} &= \left[0, -\left(\boldsymbol{p}_i - \boldsymbol{p}_c\right)_z, \left(\boldsymbol{p}_i - \boldsymbol{p}_c\right)_y \right]^\top. \end{aligned}$$

Also, as many of our arguments are symmetric across dimensions, we will use d, d_1 and d_2 to represent symmetric indexing over the dimensions, or pairs of dimensions respectively. Finally, as centering and exploring a simplicial complex from a particular triangle introduces an ordering on the tetrahedrons, faces, and edges, we will define our edges, triangles, and tetrahedrons as ordered tuples:

1. Edges: $e = \langle e_1, e_2 \rangle$,

2. Triangles: we denote these as $s = \langle s_1, s_2, s_3 \rangle$. Here $e(s)$ means the edge $\langle s_1, s_2 \rangle$.

3. Tetrahedrons: an ordered 4-tuples of pairwise adjacent points, $t = \langle t_1, t_2, t_3, t_4 \rangle$. Here $s(t)$ means the (triangle) surface $\langle t_1, t_2, t_3 \rangle$, and $e(t)$ means the edge $e(s) = \langle t_1, t_2 \rangle$.

With these notations in mind, we can center a vector $\boldsymbol{q}$ w.r.t. a particular (oriented) triangle surface s. This can be viewed as an extension of the centering lemma in [DS07]. We prove the following Lemma in Section 4.2.

Lemma 4.1. *Given a stiffly-connected, edge-simple truss $\mathcal{T} = \langle V, \{\boldsymbol{p}_i\}_{i \in V}, T, E, \gamma \rangle$, let $\boldsymbol{q}$ be a vector orthogonal to the null space of its stiffness matrix. For each oriented triangle s, there exists a (unique) vector $\overline{\boldsymbol{q}}^{\langle s \rangle}$ with scalar shift parameters $c^{\langle s \rangle \perp xy}$, $c^{\langle s \rangle \perp xz}$, $c^{\langle s \rangle \perp yz}$:*

$$\overline{\boldsymbol{q}}^{\langle s \rangle} = \boldsymbol{q} + \sum_{d1d2} c^{\langle s \rangle \perp d1d2} \boldsymbol{p}^{\perp d1d2} \tag{1}$$

satisfying:

1. *the plane containing the points $\overline{\boldsymbol{q}}_s^{\langle s \rangle}$ is parallel to the plane containing $\boldsymbol{p}_s$.*

2. *The edge $\overline{\boldsymbol{q}}_{e(s)}^{\langle s \rangle}$ is parallel to the edge $\boldsymbol{p}_{e(s)}$.*

Daitch and Spielman then showed that with any centering, the value of $\boldsymbol{q}^T \boldsymbol{M} \boldsymbol{q}$ is lower bounded by the sum of a series of shifted values, or in simpler terms, the norm of $\overline{\boldsymbol{q}}_i^{\langle s \rangle} - \overline{\boldsymbol{q}}_j^{\langle s \rangle}$ for some edge ij. However, our extension of this bound (which we will describe next in Lemma 4.4) to the 3-D case has an exponential dependency on the distance in tetrahedrons between ij and s. As a result, we first show the existence of a good centering, namely one where there exist an edge close to s whose endpoints are far apart. This notion of distance can be defined in terms of 'hop count' of tetrahedrons.

Definition 4.2. The *tetrahedron-distance* between a pair of objects x and y in a simplicial complex is the shortest sequence of tetrahedrons

$$t^{(0)}, t^{(1)}, \dots t^{(d)}$$

such that $x \subseteq t^{(0)}$, $y \subseteq t^{(d)}$, and for all $1 \leq i \leq d$, $t^{(i-1)}$ and $t^{(i)}$ share a triangle face.

We remark that because all edges and angles are within some constant range, this combinatorial distance is within constant factors of the Euclidean distance of the associated points. However, we will not make use of this connection.

Lemma 4.3. *Given a stiffly-connected, edge-simple truss $\mathcal{T}$ with n vertices and diameter Δ, let $\boldsymbol{q}$ be a unit vector orthogonal to the null space of the stiffness matrix of $\mathcal{T}$. There exists an oriented triangle s and a pair of points i, j within tetrahedron-distance $O(1)$ of s satisfying:*

$$\left\| \overline{\boldsymbol{q}}_i^{\langle s \rangle} - \overline{\boldsymbol{q}}_j^{\langle s \rangle} \right\|_2^2 = \Omega\left(\frac{1}{\Delta^4 n}\right).$$

We prove this lemma in Section 4.3.

We can then check, via an argument similar to [DS07], that such a centering and distance pair implies a large quadratic form. The following lemma will be proved in Section 5.

Lemma 4.4. *Given a stiffly-connected, edge-simple truss with stiffness matrix $\boldsymbol{M}$, an oriented triangle s, and a pair of points i, j within tetrahedron-distance h of s, we have*

$$\left\| \overline{\boldsymbol{q}}^{\langle s \rangle} \right\|_{\boldsymbol{M}}^2 \geq 2^{-\Theta(h)} \left\| \overline{\boldsymbol{q}}_i^{\langle s \rangle} - \overline{\boldsymbol{q}}_j^{\langle s \rangle} \right\|_2^2.$$

Proof of Lemma 3.3. Consider an arbitrarily fixed unit vector $\boldsymbol{q}$ that's orthogonal to the null space of $\boldsymbol{M}$. Let s be the centering given by Lemma 4.3, and let i, j be a pair of points within a constant tetrahedron-distance of s. Lemma 4.4 gives

$$\left\| \overline{\boldsymbol{q}}^{\langle s \rangle} \right\|_{\boldsymbol{M}}^2 \geq \Omega(1) \left\| \overline{\boldsymbol{q}}_i^{\langle s \rangle} - \overline{\boldsymbol{q}}_j^{\langle s \rangle} \right\|_2^2 \geq \Omega\left(\frac{1}{\Delta^4 n}\right). \tag{2}$$

On the other hand, by Equation (1), $\left\| \overline{\boldsymbol{q}}^{\langle s \rangle} \right\|_2 \geq \|\boldsymbol{q}\|_2 = 1$. Thus, $\lambda_{\min}(\boldsymbol{M}) \geq \Omega\left(\frac{1}{\Delta^4 n}\right)$. □

4.2 Centering a Vector (Proof of Lemma 4.1)

To prove Lemma 4.1, we define the following operation. Let P be any fixed plane in $\mathbb{R}^3$ and let $\boldsymbol{y} \in \mathbb{R}^3$. We define $\boldsymbol{y}^{\perp_P}$ to be the vector obtained by first projecting $\boldsymbol{y}$ onto plane P and then rotating the projected vector on the plane counterclockwise by $\pi/2$. The following claim shows that $\boldsymbol{y}^{\perp_P}$ can be written as a linear combination of $\boldsymbol{y}^{\perp_{xy}}, \boldsymbol{y}^{\perp_{yz}}, \boldsymbol{y}^{\perp_{xz}}$.

Claim 4.5 (Rotation matrix)**.** *Let P be any fixed plane in $\mathbb{R}^3$, and let $\boldsymbol{w}$ be its normal vector. Then,*

$$\forall \boldsymbol{y} \in \mathbb{R}^3, \quad \boldsymbol{y}^{\perp_P} = \boldsymbol{w}_z \boldsymbol{y}^{\perp_{xy}} - \boldsymbol{w}_y \boldsymbol{y}^{\perp_{xz}} + \boldsymbol{w}_x \boldsymbol{y}^{\perp_{yz}}.$$

Proof. Let

$$\boldsymbol{R} \stackrel{\text{def}}{=} \begin{pmatrix} 0 & -\boldsymbol{w}_z & \boldsymbol{w}_y \\ \boldsymbol{w}_z & 0 & -\boldsymbol{w}_x \\ -\boldsymbol{w}_y & \boldsymbol{w}_x & 0 \end{pmatrix}$$

be the rotation matrix which rotates a vector on the plane P counterclockwise by $\pi/2$. Then,

$$\boldsymbol{y}^{\perp_P} = \boldsymbol{R}\left(\boldsymbol{y} - (\boldsymbol{y}^\top \boldsymbol{w})\boldsymbol{w}\right) = \boldsymbol{R}\left(\boldsymbol{I} - \boldsymbol{w}^\top \boldsymbol{w}\right)\boldsymbol{y}.$$

We can check that $\boldsymbol{R}\boldsymbol{w} = \boldsymbol{0}$. It implies that $\boldsymbol{y}^{\perp_P} = \boldsymbol{R}\boldsymbol{y}$.

On the other hand,

$$\boldsymbol{w}_z \boldsymbol{y}^{\perp_{xy}} - \boldsymbol{w}_y \boldsymbol{y}^{\perp_{xz}} + \boldsymbol{w}_x \boldsymbol{y}^{\perp_{yz}} = \begin{pmatrix} 0 & -\boldsymbol{w}_z & \boldsymbol{w}_y \\ \boldsymbol{w}_z & 0 & -\boldsymbol{w}_x \\ -\boldsymbol{w}_y & \boldsymbol{w}_x & 0 \end{pmatrix} \boldsymbol{y} = \boldsymbol{R}\boldsymbol{y}.$$

Thus, the claim holds. □

Claim 4.6. *Let $\boldsymbol{h} \in \mathbb{R}^3$ be a nonzero vector. Then matrix*

$$\boldsymbol{H} \stackrel{\text{def}}{=} \begin{pmatrix} 0 & -\boldsymbol{h}_z & \boldsymbol{h}_y \\ \boldsymbol{h}_z & 0 & -\boldsymbol{h}_x \\ -\boldsymbol{h}_y & \boldsymbol{h}_x & 0 \\ \boldsymbol{h}_x & \boldsymbol{h}_y & \boldsymbol{h}_z \end{pmatrix}.$$

has rank 3.

Proof. The 2-by-2 bottom left submatrix has determinant $-(\boldsymbol{h}_y^2 + \boldsymbol{h}_x^2)$. If $\boldsymbol{h}_y^2 + \boldsymbol{h}_x^2 = 0$, then clearly $\boldsymbol{H}$ has rank 3 and we have done; otherwise, the 3rd and the 4th rows are independent.

Now it suffices to show that the 2nd row is independent of the 3rd and the 4th rows. Assume by contradiction, suppose

$$\begin{aligned} \boldsymbol{h}_z &= -\alpha \boldsymbol{h}_y + \beta \boldsymbol{h}_x, \\ 0 &= \alpha \boldsymbol{h}_x + \beta \boldsymbol{h}_y, \\ -\boldsymbol{h}_x &= \beta \boldsymbol{h}_z. \end{aligned}$$

By solving the last two equations, we get $\beta = -\boldsymbol{h}_x/\boldsymbol{h}_z$ and $\alpha = \boldsymbol{h}_y/\boldsymbol{h}_z$. Plugging these values into the 1st equation, we have $\boldsymbol{h}_x^2 + \boldsymbol{h}_y^2 + \boldsymbol{h}_z^2 = 0$, which contradicts that $\boldsymbol{h} \neq \boldsymbol{0}$. □

Proof of Lemma 4.1. Let $\boldsymbol{w}$ be the normal vector of the plane containing $s = \langle \boldsymbol{p}_{i_1}, \boldsymbol{p}_{i_2}, \boldsymbol{p}_{i_3} \rangle$, that is,

$$\boldsymbol{w}^\top(\boldsymbol{p}_{i_2} - \boldsymbol{p}_{i_1}) = 0, \boldsymbol{w}^\top(\boldsymbol{p}_{i_3} - \boldsymbol{p}_{i_1}) = 0, \text{ and } \|\boldsymbol{w}\|_2 = 1.$$

We first show that there exist $\alpha_{xy}, \alpha_{xz}, \alpha_{yz} \in \mathbb{R}$ such that the vector

$$\boldsymbol{z} \stackrel{\text{def}}{=} \boldsymbol{q} + \sum_{d1d2} \alpha_{d1d2} \boldsymbol{p}^{\perp_{d1d2}}$$

satisfies the first condition. It suffices to show that the following linear system has a solution for real numbers α_{d1d2}'s:

$$\boldsymbol{w}^\top(\boldsymbol{q}_{i_2} - \boldsymbol{q}_{i_1} + \sum_{d1d2} \alpha_{d1d2}(\boldsymbol{p}_{i_2} - \boldsymbol{p}_{i_1})^{\perp_{d1d2}}) = 0,$$
$$\boldsymbol{w}^\top(\boldsymbol{q}_{i_3} - \boldsymbol{q}_{i_1} + \sum_{d1d2} \alpha_{d1d2}(\boldsymbol{p}_{i_3} - \boldsymbol{p}_{i_1})^{\perp_{d1d2}}) = 0.$$

Rearrange it and write it in matrix form:

$$\begin{pmatrix} \boldsymbol{w}^\top(\boldsymbol{p}_{i_2} - \boldsymbol{p}_{i_1})^{\perp_{xy}} & \boldsymbol{w}^\top(\boldsymbol{p}_{i_2} - \boldsymbol{p}_{i_1})^{\perp_{yz}} & \boldsymbol{w}^\top(\boldsymbol{p}_{i_2} - \boldsymbol{p}_{i_1})^{\perp_{xz}} \\ \boldsymbol{w}^\top(\boldsymbol{p}_{i_3} - \boldsymbol{p}_{i_1})^{\perp_{xy}} & \boldsymbol{w}^\top(\boldsymbol{p}_{i_3} - \boldsymbol{p}_{i_1})^{\perp_{yz}} & \boldsymbol{w}^\top(\boldsymbol{p}_{i_3} - \boldsymbol{p}_{i_1})^{\perp_{xz}} \end{pmatrix} \begin{pmatrix} \alpha_{xy} \\ \alpha_{yz} \\ \alpha_{xz} \end{pmatrix} = \begin{pmatrix} -\boldsymbol{w}^\top(\boldsymbol{x}_{i_2} - \boldsymbol{x}_{i_1}) \\ -\boldsymbol{w}^\top(\boldsymbol{x}_{i_3} - \boldsymbol{x}_{i_1}) \end{pmatrix}.$$

It suffices to show that the coefficient matrix has rank 2.

Assume by contradiction, there is some $k \in \mathbb{R}$ such that

$$\boldsymbol{w}^\top(\boldsymbol{p}_{i_2} - \boldsymbol{p}_{i_1})^{\perp_{d1d2}} = k\boldsymbol{w}^\top(\boldsymbol{p}_{i_3} - \boldsymbol{p}_{i_1})^{\perp_{d1d2}}, \quad \forall d1, d2$$

It equals to

$$\boldsymbol{w}^\top \left((\boldsymbol{p}_{i_2} - \boldsymbol{p}_{i_1}) - k(\boldsymbol{p}_{i_3} - \boldsymbol{p}_{i_1})\right)^{\perp_{d1d2}} = 0, \quad \forall d1, d2$$

Let $\boldsymbol{h} \overset{\text{def}}{=} (\boldsymbol{p}_{i_2} - \boldsymbol{p}_{i_1}) - k(\boldsymbol{p}_{i_3} - \boldsymbol{p}_{i_1})$. Since vectors $\boldsymbol{p}_{i_2} - \boldsymbol{p}_{i_1}, \boldsymbol{p}_{i_3} - \boldsymbol{p}_{i_1}$ are not parallel, we have $\boldsymbol{h} \neq \boldsymbol{0}$. Besides, $\boldsymbol{h} \perp \boldsymbol{w}$. Write the above equation in matrix form:

$$\begin{pmatrix} 0 & -\boldsymbol{h}_z & \boldsymbol{h}_y \\ \boldsymbol{h}_z & 0 & -\boldsymbol{h}_x \\ -\boldsymbol{h}_y & \boldsymbol{h}_x & 0 \\ \boldsymbol{h}_x & \boldsymbol{h}_y & \boldsymbol{h}_z \end{pmatrix} \boldsymbol{w} = \boldsymbol{0}.$$

By Claim 4.6, the coefficient matrix has rank 3, which implies that $\boldsymbol{w} = \boldsymbol{0}$. It contradicts that $\|\boldsymbol{w}\|_2 = 1$.

Then we show that there exist $\beta_{xy}, \beta_{xz}, \beta_{yz} \in \mathbb{R}$ such that

$$\boldsymbol{q}^{\langle s \rangle} = \boldsymbol{z} + \sum_{d1d2} \beta_{d1d2} \boldsymbol{p}^{\perp_{d1d2}}$$

satisfies both conditions.

Let P be the plane containing $\boldsymbol{p}_{i_1}, \boldsymbol{p}_{i_2}, \boldsymbol{p}_{i_3}$. Let $\boldsymbol{g} \in \mathbb{R}^{3n}$ satisfy

$$\boldsymbol{g}_i = (\boldsymbol{p}_i - \boldsymbol{p}_1)^{\perp_P}, \quad \forall 1 \leq i \leq n$$

Then, there exists an appropriate multiplier $\gamma \in \mathbb{R}$ such that the vector

$$(\boldsymbol{z}_{i_2} - \boldsymbol{z}_{i_1}) + \gamma(\boldsymbol{g}_{i_2} - \boldsymbol{g}_{i_1}) = (\boldsymbol{z}_{i_2} - \boldsymbol{z}_{i_1}) + \gamma(\boldsymbol{p}_{i_2} - \boldsymbol{p}_{i_1})^{\perp_P}$$

is parallel to $\boldsymbol{p}_{i_2} - \boldsymbol{p}_{i_1}$. Besides, since both $\boldsymbol{z}$ and $\boldsymbol{g}$ are parallel to the plane P, the vector $\boldsymbol{q}^{\langle s \rangle} = \boldsymbol{z} + \gamma\boldsymbol{g}$ is parallel to the plane P.

By Claim 4.5, there exists real numbers $\beta_{xy}, \beta_{xz}, \beta_{yz}$ such that

$$\gamma\boldsymbol{g} = \sum_{d1d2} \beta_{d1d2} \boldsymbol{p}^{\perp_{d1d2}}.$$

Let

$$c^{\langle s \rangle \perp_{d1d2}} = \alpha_{d1d2} + \beta_{d1d2}, \quad \forall d1, d2$$

This completes the proof. $\square$

4.3 Existence of Good Centering (Proof of Lemma 4.3)

In this section we prove Lemma 4.3.

The proof is by contradiction. We assume that for *every* centering at a triangle s, for every pair $\langle i, j\rangle$ within tetrahedron-distance 3 of $\langle s\rangle$ satisfies

$$\left\| \overline{\boldsymbol{q}}_i^{\langle s\rangle} - \overline{\boldsymbol{q}}_j^{\langle s\rangle} \right\|_2 \le \epsilon, \tag{3}$$

where $\epsilon \overset{\text{def}}{=} \sqrt{\frac{1}{\beta\Delta^4 n}}$ for a sufficiently large constant β.

Recall that in Lemma 4.1, for each centering at $\langle s\rangle$, we define 3 scalar coefficients

$$c^{\langle s\rangle \perp xy}, c^{\langle s\rangle \perp xz}, c^{\langle s\rangle \perp yz} \in \mathbb{R}$$

for the 3 null space vectors in Equation (1). We will write the vector containing these 3 coefficients as $\boldsymbol{c}^{\langle s\rangle}$.

To prove Lemma 4.3, we need the following lemma. It says that, under the assumption in Equation (3), the difference between the coefficient vectors w.r.t. to two close centering triangles is small.

Lemma 4.7. *Assume that for every centering triangle s and every pair of points i, j within distance 3 of $\langle s\rangle$ satisfies Equation (3). Then for every pair of centering triangles s_1 and s_2 within constant tetrahedron distance to each other, we have*

$$\left\| \boldsymbol{c}^{\langle s_1\rangle} - \boldsymbol{c}^{\langle s_2\rangle} \right\|_2 = O(\epsilon).$$

The next lemma implies that: for any two vertices such that each is centered w.r.t. a triangle close to itself, the difference between the two centered vertices is small.

Lemma 4.8. *Let u, w be two arbitrary vertices of $\mathcal{T}$. Let s_u (s_w) be a triangle containing u (and w, respectively). Under the assumption in Equation (3), we have*

$$\left\| \overline{\boldsymbol{q}}_u^{\langle s_u\rangle} - \overline{\boldsymbol{q}}_w^{\langle s_w\rangle} \right\|_2 = O\left(\Delta^2 \epsilon\right).$$

Proof. Let $u = v_1, v_2, \ldots, v_f = w$ be a shortest path from u to w. Note $f \le \Delta$. Let s_i be a triangle next to v_i for $2 \le i \le f-1$. The path from vertex u, centered at $\langle s_u\rangle$, to vertex w, centered at $\langle s_w\rangle$, can be expressed as the following:

$$\overline{\boldsymbol{q}}_u^{\langle s_u\rangle} - \overline{\boldsymbol{q}}_w^{\langle s_w\rangle} = \overline{\boldsymbol{q}}_u^{\langle s_u\rangle} - \overline{\boldsymbol{q}}_{v_2}^{\langle s_u\rangle} + \sum_{2 \le i \le f-1} \left(\overline{\boldsymbol{q}}_{v_i}^{\langle s_i\rangle} - \overline{\boldsymbol{q}}_{v_i}^{\langle s_{i+1}\rangle} \right) + \sum_{2 \le i \le f-1} \left(\overline{\boldsymbol{q}}_{v_i}^{\langle s_{i+1}\rangle} - \overline{\boldsymbol{q}}_{v_{i+1}}^{\langle s_{i+1}\rangle} \right).$$

Taking ℓ_2 norm on both sides and applying the triangle inequality, we an bound the norm of the LHS by the sum of ℓ_2 norm of each term in the RHS.

By Equation (1) and the triangle inequality,

$$\left\| \overline{\boldsymbol{q}}_{v_i}^{\langle s_i\rangle} - \overline{\boldsymbol{q}}_{v_i}^{\langle s_{i+1}\rangle} \right\|_2 \le \sum_{d_1 d_2} \left| \boldsymbol{c}^{\langle s_i\rangle \perp d1d2} - \boldsymbol{c}^{\langle s_{i+1}\rangle \perp d1d2} \right| \left\| (\boldsymbol{p}_{v_i} - \boldsymbol{p}_c)^{\perp d1d2} \right\|_2.$$

Apply Lemma 4.7:

$$\left\| \overline{\boldsymbol{q}}_{v_i}^{\langle s_i\rangle} - \overline{\boldsymbol{q}}_{v_i}^{\langle s_{i+1}\rangle} \right\|_2 = O(\Delta\epsilon). \tag{4}$$

Together with our assumption in Equation (3), we have

$$\left\|\overline{\boldsymbol{q}}_u^{\langle s_u\rangle} - \overline{\boldsymbol{q}}_w^{\langle s_w\rangle}\right\|_2 \leq O(\Delta^2\epsilon).$$

$\square$

Now we prove Lemma 4.3.

Proof of Lemma 4.3. For each vertex i, let s_i denote an arbitrary triangle next to vertex i. We can write vector $\boldsymbol{q}$ as

$$\boldsymbol{q} = \widehat{\boldsymbol{q}} + \tilde{\boldsymbol{q}} - \sum_{d_1d_2} c^{\langle s_1\rangle \perp d_1d_2}\boldsymbol{p}^{\perp d_1d_2} + \boldsymbol{e}, \tag{5}$$

where

$$\widehat{\boldsymbol{q}}_i = \overline{\boldsymbol{q}}_i^{\langle s_i\rangle} - \overline{\boldsymbol{q}}_1^{\langle s_1\rangle}, \tilde{\boldsymbol{q}}_i = \overline{\boldsymbol{q}}_i^{\langle s_1\rangle} - \overline{\boldsymbol{q}}_i^{\langle s_i\rangle}, \boldsymbol{e}_i = \overline{\boldsymbol{q}}_1^{\langle s_1\rangle}.$$

Note that the last two terms of Equation (5) are in the null space of $\boldsymbol{M}$. Thus,

$$\|\widehat{\boldsymbol{q}} + \tilde{\boldsymbol{q}}\|_2 \geq \|\boldsymbol{q}\|_2 = 1.$$

On the other hand, by the triangle inequality,

$$\|\widehat{\boldsymbol{q}} + \tilde{\boldsymbol{q}}\|_2 \leq \|\widehat{\boldsymbol{q}}\|_2 + \|\tilde{\boldsymbol{q}}\|_2 = \left(\sum_i \left\|\overline{\boldsymbol{q}}_i^{\langle s_i\rangle} - \overline{\boldsymbol{q}}_1^{\langle s_1\rangle}\right\|_2^2\right)^{1/2} + \left(\sum_i \left\|\overline{\boldsymbol{q}}_i^{\langle s_1\rangle} - \overline{\boldsymbol{q}}_i^{\langle s_i\rangle}\right\|_2^2\right)^{1/2}.$$

We apply Lemma 4.8 to the first term, and apply Equation (4) (which is true for any two close centering triangles) Δ times for the second term:

$$\|\widehat{\boldsymbol{q}} + \tilde{\boldsymbol{q}}\|_2 \leq \|\widehat{\boldsymbol{q}}\|_2 + \|\tilde{\boldsymbol{q}}\|_2 = O(\sqrt{n}\Delta^2\epsilon).$$

By our choice of $\epsilon \leftarrow \frac{1}{\beta\sqrt{n}\Delta^2}$ for a sufficiently large constant β, we get a contradiction. $\square$

It remains to prove Lemma 4.7. For it, we define a matrix w.r.t. a given vector $\boldsymbol{v} \in \mathbb{R}^3$:

$$\boldsymbol{Q}_{\boldsymbol{v}} \stackrel{\text{def}}{=} \left(\boldsymbol{v}^{\perp yz}, \boldsymbol{v}^{\perp xz}, \boldsymbol{v}^{\perp xy}\right) = \begin{pmatrix} 0 & -\boldsymbol{v}_z & \boldsymbol{v}_y \\ \boldsymbol{v}_z & 0 & -\boldsymbol{v}_x \\ -\boldsymbol{v}_y & \boldsymbol{v}_x & 0 \end{pmatrix}. \tag{6}$$

We will use the following properties of $\boldsymbol{Q}_{\boldsymbol{v}}$.

Lemma 4.9. *For the matrix $\boldsymbol{Q}_{\boldsymbol{v}}$ as defined in Equation (6), its singular values are $0, 1, 1$, and its null space is multiples of the vector $\boldsymbol{v}$.*

Proof. For any vector $\boldsymbol{p} \in \mathbb{R}^3$, the cross product $\boldsymbol{v} \times \boldsymbol{p} = \boldsymbol{Q}_{\boldsymbol{v}}\boldsymbol{p}$. That is, $\boldsymbol{Q}_{\boldsymbol{v}}\boldsymbol{p}$ is the vector obtained by:

1. First projecting $\boldsymbol{p}$ onto the plane with normal vector $\boldsymbol{v}$, say $P_{\perp \boldsymbol{v}}$, and

2. then rotating the projected vector on the plane $P_{\perp \boldsymbol{v}}$ by $\frac{\pi}{2}$ counterclockwise.

From this description, we can infer that

$$\mathrm{null}(\boldsymbol{Q}_{\boldsymbol{v}}) = \mathbf{Span}\,(\boldsymbol{v})$$

and any vector orthogonal to $\mathrm{null}(\boldsymbol{Q}_{\boldsymbol{v}})$ is on this plane $P_{\perp \boldsymbol{v}}$. Such vectors are not affected by the first step projection, and their lengths are not changed by the subsequent rotation. This means for such vectors $\boldsymbol{u} \perp \mathrm{null}(\boldsymbol{Q}_{\boldsymbol{v}})$ we have $\boldsymbol{u}^\top \boldsymbol{Q}_{\boldsymbol{v}}^\top \boldsymbol{Q}_{\boldsymbol{v}} \boldsymbol{u} = \|\boldsymbol{u}\|_2^2$. Thus, the singular values of $\boldsymbol{Q}_{\boldsymbol{v}}$ are 1,1,0. □

Lemma 4.10. *If $\boldsymbol{v}$ and $\boldsymbol{u}$ are vectors with length at least 1 such that the angle between them is $\theta \in (0, \pi)$, then the matrix*

$$\boldsymbol{Q}_{\boldsymbol{v}}^\top \boldsymbol{Q}_{\boldsymbol{v}} + \boldsymbol{Q}_{\boldsymbol{u}}^\top \boldsymbol{Q}_{\boldsymbol{u}}$$

is full rank, and has minimum eigenvalue at least $2\sin^2 \frac{\theta}{2}$.

Proof. Let $\boldsymbol{h} \in \mathbb{R}^3$ be a unit vector. Decompose $\boldsymbol{h}$:

$$\boldsymbol{h} = \alpha_v \boldsymbol{v} + \beta_v \hat{\boldsymbol{v}} = \alpha_u \boldsymbol{u} + \beta_u \hat{\boldsymbol{u}},$$

where $\alpha_v, \alpha_u, \beta_v, \beta_u \in \mathbb{R}$, $\hat{\boldsymbol{v}}$ is a unit vector orthogonal to $\boldsymbol{v}$, and $\hat{\boldsymbol{u}}$ is a unit vector orthogonal to $\boldsymbol{u}$. Then,

$$\boldsymbol{h}^\top (\boldsymbol{Q}_v^\top \boldsymbol{Q}_v + \boldsymbol{Q}_u^\top \boldsymbol{Q}_u) \boldsymbol{h} = \beta_v^2 \, \|\boldsymbol{Q}_v \hat{\boldsymbol{v}}\|_2^2 + \beta_u^2 \, \|\boldsymbol{Q}_u \hat{\boldsymbol{u}}\|_2^2 .$$

By Lemma 4.9, $\|\boldsymbol{Q}_v \hat{\boldsymbol{v}}\|_2 = 1$ and $\|\boldsymbol{Q}_u \hat{\boldsymbol{u}}\|_2 = 1$. Thus,

$$\boldsymbol{h}^\top (\boldsymbol{Q}_v^\top \boldsymbol{Q}_v + \boldsymbol{Q}_u^\top \boldsymbol{Q}_u) \boldsymbol{h} = \beta_v^2 + \beta_u^2 = 2 - (\alpha_v^2 + \alpha_u^2).$$

The second equality is due to that $\boldsymbol{h}$ is a unit vector.

$$\alpha_v^2 + \alpha_u^2 = \left\|\boldsymbol{h}^\top \boldsymbol{v}\right\|_2^2 + \left\|\boldsymbol{h}^\top \boldsymbol{u}\right\|_2^2 \leq 2\cos^2 \frac{\theta}{2}.$$

Therefore,

$$\lambda_{\min}(\boldsymbol{Q}_v^\top \boldsymbol{Q}_v + \boldsymbol{Q}_u^\top \boldsymbol{Q}_u) \geq 2\sin^2 \frac{\theta}{2}.$$

$\theta < \pi$ implies that $\lambda_{\min}(\boldsymbol{Q}_v^\top \boldsymbol{Q}_v + \boldsymbol{Q}_u^\top \boldsymbol{Q}_u) > 0$, and thus the matrix $\boldsymbol{Q}_v^\top \boldsymbol{Q}_v + \boldsymbol{Q}_u^\top \boldsymbol{Q}_u$ has full rank. □

Equipped with the above lemmas, we prove Lemma 4.7.

Proof of Lemma 4.7. Let s_1 and s_2 be two triangle centerings for which there exist two edges belong to same tetrahedron, say $(i, j), (j, k)$, such that vertices i, j, k are all within tetrahedron-distance constant h of both s_1 and s_2.

Recall $\overline{\boldsymbol{q}}^{\langle s_1 \rangle}$ is defined in Equation (1). Subtracting

$$\overline{\boldsymbol{q}}_i^{\langle s_1 \rangle} = \boldsymbol{q}_i + \sum_{d_1 d_2} c^{\langle s_1 \rangle \perp d_1 d_2} \left(\boldsymbol{p}_i^{\perp d_1 d_2} - \boldsymbol{p}_c^{\perp d_1 d_2}\right)$$

from the corresponding equation for j gives:

$$\overline{\boldsymbol{q}}_i^{\langle s_1 \rangle} - \overline{\boldsymbol{q}}_j^{\langle s_1 \rangle} = \boldsymbol{q}_i - \boldsymbol{q}_j + \sum_{d_1 d_2} c^{\langle s_1 \rangle \perp d_1 d_2} \left(\boldsymbol{p}_i^{\perp d_1 d_2} - \boldsymbol{p}_j^{\perp d_1 d_2}\right).$$

Plugging in the definition of $\boldsymbol{Q}_{(\boldsymbol{p}_i-\boldsymbol{p}_j)}$:

$$\overline{\boldsymbol{q}}_i^{\langle s_1\rangle}-\overline{\boldsymbol{q}}_j^{\langle s_1\rangle}=\boldsymbol{q}_i-\boldsymbol{q}_j+\boldsymbol{Q}_{(\boldsymbol{p}_i-\boldsymbol{p}_j)}\boldsymbol{c}^{\langle s_1\rangle}.$$

Subtracting this equation from centering triangle s_2 in turn cancels the $\boldsymbol{q}_i-\boldsymbol{q}_j$ term on the RHS, giving:

$$\boldsymbol{Q}_{(\boldsymbol{p}_i-\boldsymbol{p}_j)}\left(\boldsymbol{c}^{\langle s_1\rangle}-\boldsymbol{c}^{\langle s_2\rangle}\right)=\left(\overline{\boldsymbol{q}}_i^{\langle s_1\rangle}-\overline{\boldsymbol{q}}_j^{\langle s_1\rangle}\right)-\left(\overline{\boldsymbol{q}}_i^{\langle s_2\rangle}-\overline{\boldsymbol{q}}_j^{\langle s_2\rangle}\right).$$

Together with the corresponding equation for j,k,

$$\begin{pmatrix}\boldsymbol{Q}_{(\boldsymbol{p}_i-\boldsymbol{p}_j)}\\ \boldsymbol{Q}_{(\boldsymbol{p}_j-\boldsymbol{p}_k)}\end{pmatrix}\left(\boldsymbol{c}^{\langle s_1\rangle}-\boldsymbol{c}^{\langle s_2\rangle}\right)=\begin{pmatrix}\left(\overline{\boldsymbol{q}}_i^{\langle s_1\rangle}-\overline{\boldsymbol{q}}_j^{\langle s_1\rangle}\right)-\left(\overline{\boldsymbol{q}}_i^{\langle s_2\rangle}-\overline{\boldsymbol{q}}_j^{\langle s_2\rangle}\right)\\ \left(\overline{\boldsymbol{q}}_j^{\langle s_1\rangle}-\overline{\boldsymbol{q}}_k^{\langle s_1\rangle}\right)-\left(\overline{\boldsymbol{q}}_j^{\langle s_2\rangle}-\overline{\boldsymbol{q}}_k^{\langle s_2\rangle}\right)\end{pmatrix}.$$

Multiplying $\begin{pmatrix}\boldsymbol{Q}_{(\boldsymbol{p}_i-\boldsymbol{p}_j)}^\top & \boldsymbol{Q}_{(\boldsymbol{p}_j-\boldsymbol{p}_k)}^\top\end{pmatrix}$ on both sides gives:

$$\begin{aligned}&\left(\boldsymbol{Q}_{(\boldsymbol{p}_i-\boldsymbol{p}_j)}^\top\boldsymbol{Q}_{(\boldsymbol{p}_i-\boldsymbol{p}_j)}+\boldsymbol{Q}_{(\boldsymbol{p}_j-\boldsymbol{p}_k)}^\top\boldsymbol{Q}_{(\boldsymbol{p}_j-\boldsymbol{p}_k)}\right)\left(\boldsymbol{c}^{\langle s_1\rangle}-\boldsymbol{c}^{\langle s_2\rangle}\right)\\ &\qquad=\begin{pmatrix}\boldsymbol{Q}_{(\boldsymbol{p}_i-\boldsymbol{p}_j)}^\top & \boldsymbol{Q}_{(\boldsymbol{p}_j-\boldsymbol{p}_k)}^\top\end{pmatrix}\begin{pmatrix}\left(\overline{\boldsymbol{q}}_i^{\langle s_1\rangle}-\overline{\boldsymbol{q}}_j^{\langle s_1\rangle}\right)-\left(\overline{\boldsymbol{q}}_i^{\langle s_2\rangle}-\overline{\boldsymbol{q}}_j^{\langle s_2\rangle}\right)\\ \left(\overline{\boldsymbol{q}}_j^{\langle s_1\rangle}-\overline{\boldsymbol{q}}_k^{\langle s_1\rangle}\right)-\left(\overline{\boldsymbol{q}}_j^{\langle s_2\rangle}-\overline{\boldsymbol{q}}_k^{\langle s_2\rangle}\right)\end{pmatrix}.\end{aligned}$$

Solving the above linear equations and taking norm on both sides:

$$\begin{aligned}\left\|\boldsymbol{c}^{\langle s_1\rangle}-\boldsymbol{c}^{\langle s_2\rangle}\right\|_2&\le\left\|\left(\boldsymbol{Q}_{(\boldsymbol{p}_i-\boldsymbol{p}_j)}^\top\boldsymbol{Q}_{(\boldsymbol{p}_i-\boldsymbol{p}_j)}+\boldsymbol{Q}_{(\boldsymbol{p}_j-\boldsymbol{p}_k)}^\top\boldsymbol{Q}_{(\boldsymbol{p}_j-\boldsymbol{p}_k)}\right)^{-1}\right\|_2\\ &\qquad\cdot\left\|\begin{pmatrix}\boldsymbol{Q}_{(\boldsymbol{p}_i-\boldsymbol{p}_j)}^\top & \boldsymbol{Q}_{(\boldsymbol{p}_j-\boldsymbol{p}_k)}^\top\end{pmatrix}\right\|_2\left\|\begin{pmatrix}\left(\overline{\boldsymbol{q}}_i^{\langle s_1\rangle}-\overline{\boldsymbol{q}}_j^{\langle s_1\rangle}\right)-\left(\overline{\boldsymbol{q}}_i^{\langle s_2\rangle}-\overline{\boldsymbol{q}}_j^{\langle s_2\rangle}\right)\\ \left(\overline{\boldsymbol{q}}_j^{\langle s_1\rangle}-\overline{\boldsymbol{q}}_k^{\langle s_1\rangle}\right)-\left(\overline{\boldsymbol{q}}_j^{\langle s_2\rangle}-\overline{\boldsymbol{q}}_k^{\langle s_2\rangle}\right)\end{pmatrix}\right\|_2.\end{aligned}$$

Applying Lemma 4.9 and 4.10 on the first two terms, and applying the triangle inequality and the assumption in Equation (3) give:

$$\left\|\boldsymbol{c}^{\langle s_1\rangle}-\boldsymbol{c}^{\langle s_2\rangle}\right\|_2\le O(\epsilon).$$

□

5 Proof of Path Lemma

We now prove Lemma 4.4, which lower bounds the quadratic form of an edge-simple and stiffly-connected truss stiffness matrix by the distance between two centered points. As we now only deal with a single centering in this section, we will drop this superscription for simplicity and relabel all indices w.r.t. this centering. Our relabeling is similar to that in [DS07].

5.1 Relabeling Tetrahedrons and Vertices

Fix an arbitrary tetrahedron, say t_1, and we center the vector $\boldsymbol{q}$ w.r.t. one of the triangle faces of t_1 as in Lemma 4.1. Use t_1 as root, we run breadth-first-search (BFS) in the rigidity graph (refer to Definition 2.2) of $\mathcal{T}$, and relabel the tetrahedrons of $\mathcal{T}$ according to this BFS ordering. For example, the neighbor tetrahedrons of t_1 are labeled as $t_2, t_3, \ldots$. Let T_{BFS} be the corresponding BFS tree in which each node represents a tetrahedron in $\mathcal{T}$.

Based on T_{BFS}, we relabel the vertices of $\mathcal{T}$ as follows. We label the vertices of t_1 by $-2, -1, 0, 1$ in an arbitrary order. Each child of t_1 (and subsequent recursions) shares a triangle face with their respective parent tetrahedron and thus only requires us to label one vertex per child tetrahedron, which can be labeled according to the BFS ordering.

For each newly labeled vertex j, we use t_j to denote the tetrahedron encompassing the t_j and its parental face. Besides, we use a 3-dimensional vector σ_j to consist of the indexes of the parental face sorted in ascending order, and is assumed that indexing this vector is implicitly modulo 3. Then, $t_j = \{j, \sigma_j(1), \sigma_j(2), \sigma_j(3)\}$. For completeness, we define $\sigma_1 \stackrel{\text{def}}{=} \{-2, -1, 0\}$. See Figure 1 for an example.

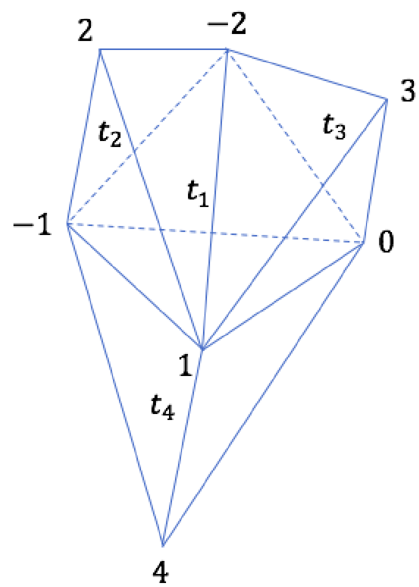

Figure 1: An example of the relabeling of tetrahedrons and vertices: $t_1 = \{-2, -1, 0, 1\}, t_2 = \{-2, -1, 1, 2\}, t_3 = \{-2, 0, 1, 3\}, t_4 = \{-1, 0, 1, 4\}$ and $\sigma_1 = \{-2, -1, 0\}, \sigma_2 = \{-2, -1, 1\}, \sigma_3 = \{-2, 1, 0\}, \sigma_4 = \{0, -1, 1\}$.

5.2 Distance between Local Minimizers and the Centered Vectors

Let $\boldsymbol{q} \in \mathbb{R}^{3n}$ be a unit vector which is orthogonal to $\mathbf{Span}\left(\boldsymbol{p}^x, \boldsymbol{p}^y, \boldsymbol{p}^z, \boldsymbol{p}^{\perp xy}, \boldsymbol{p}^{\perp yz}, \boldsymbol{p}^{\perp xz}\right)$ defined at the beginning of Section 4.1. By Lemma 4.1, there exist scalars $c^{\langle t_1 \rangle \perp xy}, c^{\langle t_1 \rangle \perp xz}, c^{\langle t_1 \rangle \perp yz} \in \mathbb{R}$ such that the vector

$$\overline{\boldsymbol{q}}^{\langle t_1 \rangle} \stackrel{\text{def}}{=} \boldsymbol{q} + \sum_{d_1 d_2} c^{\langle t_1 \rangle \perp d_1 d_2} \boldsymbol{p}^{\perp d_1 d_2}$$

satisfies:

1. the plane containing $\overline{\boldsymbol{q}}_{-2}^{\langle t_1 \rangle}, \overline{\boldsymbol{q}}_{-1}^{\langle t_1 \rangle}, \overline{\boldsymbol{q}}_0^{\langle t_1 \rangle}$ is parallel to the plane σ_{t_1}, and

2. $\overline{\boldsymbol{q}}_{-2}^{\langle t_1 \rangle} - \overline{\boldsymbol{q}}_{-1}^{\langle t_1 \rangle}$ is parallel to $(\sigma_{t_1}(1) - \sigma_{t_1}(2))$.

We drop the superscription $\langle t_1 \rangle$ when the context is clear.

For each $0 \le i \le n-3$, we define a 3-dimensional vector $\boldsymbol{y}_i$. $\boldsymbol{y}_0$ is a vector on the plane of $\sigma_{t_1} = \{\boldsymbol{p}_{-2}, \boldsymbol{p}_{-1}, \boldsymbol{p}_0\}$ and minimizes the energy / quadratic form, suppose both $\overline{\boldsymbol{q}}_{-2}^{\langle t_1 \rangle}, \overline{\boldsymbol{q}}_{-1}^{\langle t_1 \rangle}$ are fixed. That is,

$$\begin{array}{l} (\boldsymbol{p}_0 - \boldsymbol{p}_j)^\top (\boldsymbol{y}_0 - \overline{\boldsymbol{q}}_j) = 0, \quad \forall j \in \{-2, -1\} \\ \boldsymbol{w}^\top (\boldsymbol{y}_0 - \overline{\boldsymbol{q}}_{-2}) = 0 \end{array}. \tag{7}$$

For each $1 \le i \le n-3$, $\boldsymbol{y}_i$ is a vector which minimizes the energy / quadratic form, suppose all the three points of σ_j are fixed. That is,

$$(\boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i(j)})^\top (\boldsymbol{y}_i - \overline{\boldsymbol{q}}_{\sigma_i(j)}) = 0, \quad \forall j \in \{1, 2, 3\} \tag{8}$$

We then define the distance between each local minimizer $\boldsymbol{y}_i$ and the centered vector $\overline{\boldsymbol{q}}_i$. Specifically,

$$\boldsymbol{d}_i \stackrel{\text{def}}{=} \begin{cases} \overline{\boldsymbol{q}}_{-1} - \overline{\boldsymbol{q}}_{-2}, & i = -1 \\ \overline{\boldsymbol{q}}_i - \boldsymbol{y}_i, & 0 \le i \le n-3 \end{cases} \tag{9}$$

This definition intermediately gives that $\forall -1 \le i \le n-3, 1 \le j \le \min\{i+2, 3\}$,

$$\boldsymbol{d}_i^\top (\boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i(j)}) = (\overline{\boldsymbol{q}}_i - \boldsymbol{y}_i)^\top (\boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i(j)}) = (\overline{\boldsymbol{q}}_i - \overline{\boldsymbol{q}}_{\sigma_i(j)})^\top (\boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i(j)}). \tag{10}$$

Note that equation (10) describes the net stress on an edge.

We can explicitly express these $\boldsymbol{d}_i$'s by solving Equations (7) and (8).

Let

$$\boldsymbol{F}_0 \stackrel{\text{def}}{=} \begin{pmatrix} (\boldsymbol{p}_0 - \boldsymbol{p}_{-2})^\top \\ (\boldsymbol{p}_0 - \boldsymbol{p}_{-1})^\top \\ \boldsymbol{w}^\top \end{pmatrix}.$$

Let $\boldsymbol{h}_{-2}, \boldsymbol{h}_{-1}, \boldsymbol{h}_{\boldsymbol{w}} \in \mathbb{R}^3$ such that for each $1 \le k \le 3$,

$$\begin{aligned} \boldsymbol{h}_{-2}(k) &\stackrel{\text{def}}{=} \det \begin{pmatrix} (\boldsymbol{p}_0 - \boldsymbol{p}_{-1})^{(k+1)_3} & (\boldsymbol{p}_0 - \boldsymbol{p}_{-1})^{(k+2)_3} \\ \boldsymbol{w}^{(k+1)_3} & \boldsymbol{w}^{(k+2)_3} \end{pmatrix}, \\ \boldsymbol{h}_{-1}(k) &\stackrel{\text{def}}{=} \det \begin{pmatrix} (\boldsymbol{p}_0 - \boldsymbol{p}_{-2})^{(k+1)_3} & (\boldsymbol{p}_0 - \boldsymbol{p}_{-2})^{(k+2)_3} \\ \boldsymbol{w}^{(k+1)_3} & \boldsymbol{w}^{(k+2)_3} \end{pmatrix}, \\ \boldsymbol{h}_{\boldsymbol{w}}(k) &\stackrel{\text{def}}{=} \det \begin{pmatrix} (\boldsymbol{p}_0 - \boldsymbol{p}_{-1})^{(k+1)_3} & (\boldsymbol{p}_0 - \boldsymbol{p}_{-1})^{(k+2)_3} \\ (\boldsymbol{p}_0 - \boldsymbol{p}_{-2})^{(k+1)_3} & (\boldsymbol{p}_0 - \boldsymbol{p}_{-2})^{(k+2)_3} \end{pmatrix}. \end{aligned}$$

where $(k+1)_3$ and $(k+2)_3$ denotes accessing specific dimensions of vectors $\in \mathbb{R}^3$ modulo 3. Then we define

$$\boldsymbol{H}_{0,2} \stackrel{\text{def}}{=} \boldsymbol{h}_{-2}(\boldsymbol{p}_0 - \boldsymbol{p}_{-2})^\top, \quad \boldsymbol{H}_{0,1} \stackrel{\text{def}}{=} \boldsymbol{h}_{-1}(\boldsymbol{p}_0 - \boldsymbol{p}_{-1})^\top, \quad \text{and } \boldsymbol{H}_{\boldsymbol{w}} \stackrel{\text{def}}{=} \boldsymbol{h}_{\boldsymbol{w}} \boldsymbol{w}^\top.$$

Similarly, for $1 \le i \le n-3$, let

$$\boldsymbol{F}_i \stackrel{\text{def}}{=} \begin{pmatrix} (\boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i(1)})^\top \\ (\boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i(2)})^\top \\ (\boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i(3)})^\top \end{pmatrix},$$

and

$$\boldsymbol{H}_{i,j} \stackrel{\text{def}}{=} \boldsymbol{h}_{i,j}(\boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i(j)})^\top, \quad \forall 1 \le j \le 3, \tag{11}$$

where $\boldsymbol{h}_{i,j} \in \mathbb{R}^3$ with

$$\boldsymbol{h}_{i,j}(k) \stackrel{\text{def}}{=} \det \begin{pmatrix} (\boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i((j+1)_3)})^{(k+1)_3} & (\boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i((j+1)_3)})^{(k+2)_3} \\ (\boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i((j+2)_3)})^{(k+1)_3} & (\boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i((j+2)_3)})^{(k+2)_3} \end{pmatrix}.$$

We can check that the following $\boldsymbol{y}_i$'s satisfy Equations (7) and (8):

$$\boldsymbol{y}_i = \begin{cases} \frac{1}{\det(\boldsymbol{F}_0)} \left((\boldsymbol{H}_{0,2} + \boldsymbol{H}_{\boldsymbol{w}}) \overline{\boldsymbol{q}}_{-2} + \boldsymbol{H}_{0,1} \overline{\boldsymbol{q}}_{-1} \right), & i = 0 \\ \frac{1}{\det(\boldsymbol{F}_i)} \sum_{1 \le j \le 3} \boldsymbol{H}_{i,j} \overline{\boldsymbol{q}}_{\sigma_i(j)}, & \forall 1 \le i \le n \end{cases}$$

Claim 5.1.

$$\det(\boldsymbol{F}_i) = (\boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i(j)})^\top \boldsymbol{h}_{i,j}. \tag{12}$$

Proof. Let

$$\boldsymbol{Q} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$$

be a 3×3 matrix. Then, the determinant of $\boldsymbol{Q}$ is

$$\det(\boldsymbol{Q}) = a \cdot \det\left(\begin{pmatrix} e & f \\ h & i \end{pmatrix} \right) - b \cdot \det\left(\begin{pmatrix} d & f \\ g & i \end{pmatrix} \right) + c \cdot \det\left(\begin{pmatrix} d & e \\ g & h \end{pmatrix} \right).$$

Applying the above rule to $\boldsymbol{F}_i$ gives Eqaution (12). □

Claim 5.2.

$$\det(\boldsymbol{F}_i)\boldsymbol{I} = \sum_{1 \le j \le 3} \boldsymbol{H}_{i,j}, \quad \forall i \ge 0$$

Proof. We first show that the diagonals of $\sum_{j \in [3]} \boldsymbol{H}_{i,j}$ are equal to $\det(\boldsymbol{F}_i)$.

$$\begin{aligned} &\sum_{j \in [3]} \boldsymbol{H}_{i,j}(1,1) \\ =& \sum_{j \in [3]} (\boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i(j)})(1) \cdot \det \begin{pmatrix} (\boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i((j+1)_3)})(2) & (\boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i((j+1)_3)})(3) \\ (\boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i((j+2)_3)})(2) & (\boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i((j+2)_3)})(3) \end{pmatrix} \\ =& \det \begin{pmatrix} (\boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i(1)})^\top \\ (\boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i(2)})^\top \\ (\boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i(3)})^\top \end{pmatrix} \\ =& \det(\boldsymbol{F}_i). \end{aligned}$$

Similarly, we have

$$\sum_{j \in [3]} \boldsymbol{H}_{i,j}(2,2) = \sum_{j \in [3]} \boldsymbol{H}_{i,j}(3,3) = \det(\boldsymbol{F}_i).$$

Then, we show that the off-diagonals of $\sum_{j\in[3]} \boldsymbol{H}_{i,j}$ are 0.

$$\begin{aligned}
&\sum_{j\in[3]} \boldsymbol{H}_{i,j}(1,2)\\
=&\sum_{j\in[3]} (\boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i((j+1)_3)})(2)\cdot \det\begin{pmatrix} (\boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i((j+1)_3)})(2) & (\boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i((j+1)_3)})(3) \\ (\boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i((j+2)_3)})(2) & (\boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i((j+2)_3)})(3) \end{pmatrix}\\
=&\det\begin{pmatrix} (\boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i(1)})(2) & (\boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i(1)})(2) & (\boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i(1)})(3) \\ (\boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i(2)})(2) & (\boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i(2)})(2) & (\boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i(2)})(3) \\ (\boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i(3)})(2) & (\boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i(3)})(2) & (\boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i(3)})(3) \end{pmatrix}\\
=&0.
\end{aligned}$$

The last equation is due to the 3 columns of the matrix are linearly dependent. Similarly,

$$\sum_{j\in[3]} \boldsymbol{H}_{i,j}(s,t) = 0, \quad \forall 1 \le s \ne t \le 3.$$

This completes the proof. □

Plugging the above equations into the definition of $\boldsymbol{d}_i$'s gives:

Claim 5.3. *For each $-1 \le i \le n-3$,*

$$\boldsymbol{d}_i = \overline{\boldsymbol{q}}_i - \overline{\boldsymbol{q}}_{\sigma_i(j)} + \frac{1}{\det(\boldsymbol{F}_i)} \sum_{\substack{1\le j'\le \min\{i+2,3\}\\ j'\ne j}} \boldsymbol{H}_{i,j'}\left(\overline{\boldsymbol{q}}_{\sigma_i(j)} - \overline{\boldsymbol{q}}_{\sigma_i(j')}\right).$$

5.3 Bounding the Norm of $\overline{\boldsymbol{q}}_i - \overline{\boldsymbol{q}}_{\sigma_i(j)}$ in terms of $\boldsymbol{d}_i$'s

Rearranging the above equation gives that for each $-1 \le i \le n-3, 1 \le j \le \min\{i+2,3\}$,

$$\overline{\boldsymbol{q}}_i - \overline{\boldsymbol{q}}_{\sigma_i(j)} = \boldsymbol{d}_i - \frac{1}{\det(\boldsymbol{F}_i)} \sum_{\substack{1\le j'\le \min\{i+2,3\}\\ j'\ne j}} \boldsymbol{H}_{i,j'}\left(\overline{\boldsymbol{q}}_{\sigma_i(j)} - \overline{\boldsymbol{q}}_{\sigma_i(j')}\right). \tag{13}$$

Our goal is to express each $\overline{\boldsymbol{q}}_i - \overline{\boldsymbol{q}}_{\sigma_i(j)}$ as a function of $\boldsymbol{d}_i, \boldsymbol{d}_{i-1}, \ldots, \boldsymbol{d}_{-1}$. If each term $\overline{\boldsymbol{q}}_{\sigma_i(j)} - \overline{\boldsymbol{q}}_{\sigma_i(j')}$ in the right hand side of Equation (13) satisfies

$$\{\sigma_i(j), \sigma_i(j')\} = \{i_1, \sigma_{i_1}(j_1)\}$$

for some $i_1 < i$ and $1 \le j_1 \le \min\{i_1+2, 3\}$, then we can substitute it by Equation (13) with the left hand side being $\overline{\boldsymbol{q}}_{i_1} - \overline{\boldsymbol{q}}_{\sigma_{i_1}(j_1)}$. The substitution terminates when the right hand side term becomes $\pm\left(\overline{\boldsymbol{q}}_{-1} - \overline{\boldsymbol{q}}_{-2}\right) = \pm \boldsymbol{d}_{-1}$.

Claim 5.4. *Let $\overline{\boldsymbol{q}}_{\sigma_{i_1}(j_1)} - \overline{\boldsymbol{q}}_{\sigma_{i_1}(j_2)}$ ($0 \le i_1 \le n-3, 1 \le j_1 \ne j_2 \le \min\{i+2,3\}$) be a term appearing in the right hand side of Equation* (13)*. There exist $-1 \le i_2 < i_1, 1 \le j_3 \le \min\{i_2+2,3\}$ satisfying*

$$\{\sigma_{i_1}(j_1), \sigma_{i_1}(j_2)\} = \{i_2, \sigma_{i_2}(j_3)\}.$$

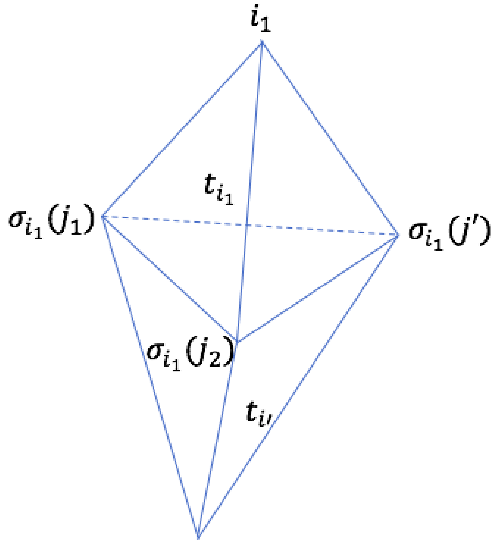

Figure 2: Tetrahedrons t_{i_1} and $t_{i'}$

Proof. Without the loss of generality, assume $\sigma_{i_1}(j_1) > \sigma_{i_1}(j_2)$. If $i_1 = 0$, then $\{\sigma_{i_1}(j_1), \sigma_{i_1}(j_2)\} = \{-1, \sigma_{-1}(1)\}$; if $i_1 = 1$, then $\{\sigma_{i_1}(j_1), \sigma_{i_1}(j_2)\} \in \{\{0, \sigma_0(1)\}, \{0, \sigma_0(2)\}, \{-1, \sigma_0(1)\}\}$. The remaining proof focuses on $i_1 \geq 2$.

Let $\sigma_{i_1}(j')$ be the other vertex in tetrahedron t_{i_1}. Let $t_{i'}$ be the parent of t_{i_1} in the BFS tree T_{BFS}. $i' < i_1$, and the two tetrahedrons t_{i_1} and $t_{i'}$ share a triangle face containing vertices $\sigma_{i_1}(j_1), \sigma_{i_1}(j_2), \sigma_{i_1}(j')$. See Figure 2. If $i' = 1$, then

$$\{\sigma_{i_1}(j_1), \sigma_{i_1}(j_2)\} \in \{\{i_2, \sigma_{i_2}(j_3)\} : -1 \leq i_2 \leq 1, 1 \leq j_3 \leq \min\{i_2 + 2, 3\}\}.$$

Otherwise $i' \geq 2$, we prove the statement by case analysis.
Case 1. $\sigma_{i_1}(j_1) = \max_{1 \leq k \leq 3}\{\sigma_{i_1}(k)\}$. By our labeling rules, vertex i' is the one with the maximum index among all the vertices of $t_{i'}$ and it is contained in the triangle shared by t_{i_1} and $t_{i'}$. Thus, $\sigma_{i_1}(j_1) = i'$ and

$$\{\sigma_{i_1}(j_1), \sigma_{i_1}(j_2)\} = \{i', \sigma_{i'}(j_3)\}$$

for some $1 \leq j_3 \leq 3$.
Case 2. $\sigma_{i_1}(j') = \max_{1 \leq k \leq 3}\{\sigma_{i_1}(k)\}$. Then, $\sigma_{i_1}(j') = i'$ and $\sigma_{i_1}(j_1), \sigma_{i_1}(j_2) \in \{\sigma_{i'}(k) : 1 \leq k \leq 3\}$. Note $i' < i$. By induction on the tetrahedron index i', we can see that there exist i_2, j_3 satisfying $\{\sigma_{i_1}(j_1), \sigma_{i_1}(j_2)\} = \{i_2, \sigma_{i_2}(j_3)\}$. ☐

We use a recursion tree $T_{\text{rec}}^{(i,j)}$ to express the process of recursively substituting $\overline{\boldsymbol{q}}_{i'} - \overline{\boldsymbol{q}}_{\sigma_{i'}(j')}$ via Equation (13). The root of $T_{\text{rec}}^{(i,j)}$ is $\overline{\boldsymbol{q}}_i - \overline{\boldsymbol{q}}_{\sigma_i(j)}$, and each node u of $T_{\text{rec}}^{(i,j)}$ represents a term $\overline{\boldsymbol{q}}_{i_u} - \overline{\boldsymbol{q}}_{\sigma_{i_u}(j_u)}$ which is substituted at that point. The leaves are $\pm(\overline{\boldsymbol{q}}_{-1} - \overline{\boldsymbol{q}}_{-2})$, which equals $\pm\boldsymbol{d}_{-1}$ by Equation (9).

Claim 5.5. *The number of nodes in the recursion tree $T_{\text{rec}}^{(i,j)}$ is at most 2^i.*

Proof. Since $T_{\text{rec}}^{(i,j)}$ is a binary tree, it suffices to prove that the height of $T_{\text{rec}}^{(i,j)}$ is at most i. For each non-leaf node u of $T_{\text{rec}}^{(i,j)}$, let u_1 be a child of u. According to Equation (13) and our labeling rules,

i_{u_1} is a vertex in tetrahedron t_{i_u} and $i_{u_1} < i_u$. Let t_k be the tetrahedron such that t_k and t_{i_u} share a triangle face containing vertices $\sigma_{i_u}(1), \sigma_{i_u}(2), \sigma_{i_u}(3)$. Then in T_{BFS}, t_k is the parent of t_{i_u}. Since $i_{u_1} < i_u$, in T_{BFS}, the depth of $t_{i_{u_1}}$ is smaller than the depth of t_{i_u}. See Figure 3. It implies that the height of $T_{\mathrm{rec}}^{(i,j)}$ is at most the height of T_{BFS}, which is at most i. $\square$

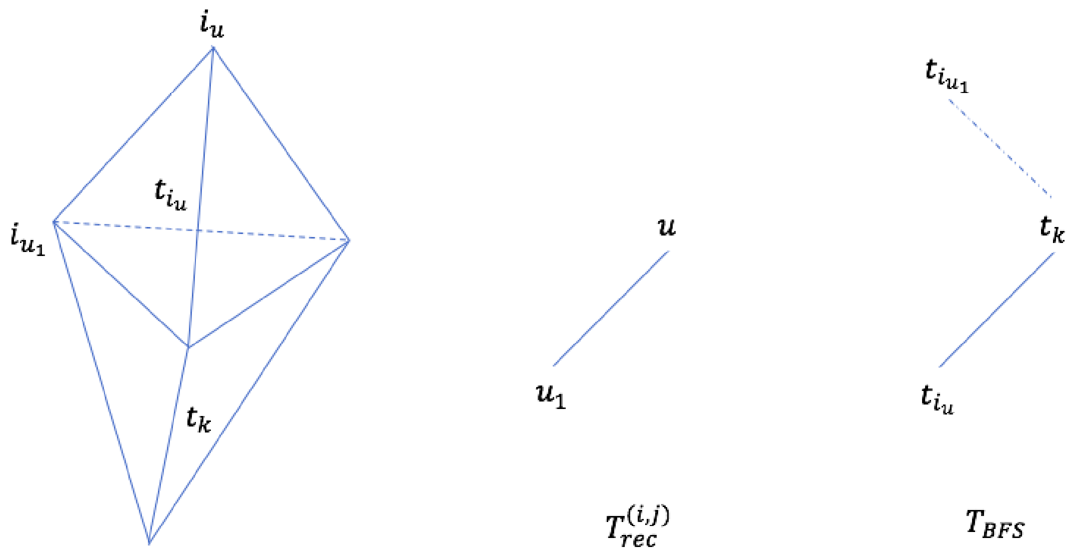

Figure 3: If u_1 is a child of u in $T_{\mathrm{rec}}^{(i,j)}$, then the depth of $t_{i_{u_1}}$ is smaller than the depth of t_{i_u} in T_{BFS}.

At each node u of the recursion tree $T_{\mathrm{rec}}^{(i,j)}$, by applying Equation (13) with the left hand side being $\overline{\boldsymbol{q}}_{i_u} - \overline{\boldsymbol{q}}_{\sigma_{i_u}(j_u)}$, we introduce a term related to $\boldsymbol{d}_{i_u}$. For each non-root node u, denote $\mathrm{pa}(u)$ the parent of u in $T_{\mathrm{rec}}^{(i,j)}$. Let P_u be the path from the root node to node u in $T_{\mathrm{rec}}^{(i,j)}$. For each non-root node u, define

$$\boldsymbol{M}_u \stackrel{\mathrm{def}}{=} \prod_{v \in P_{pa(u)}} \frac{\boldsymbol{H}_{i_v, j_v}}{\det(\boldsymbol{F}_{i_v})}. \tag{14}$$

For root node r, define $\boldsymbol{M}_r \stackrel{\mathrm{def}}{=} \boldsymbol{I}$. By recursively applying Equation (13), we get

$$\overline{\boldsymbol{q}}_i - \overline{\boldsymbol{q}}_{\sigma_i(j)} = \sum_{u \in T_{\mathrm{rec}}^{(i,j)}} \mathrm{sgn}(u) \boldsymbol{M}_u \boldsymbol{d}_{i_u},$$

where $\mathrm{sgn}(u) \in \{+1, -1\}$. By the triangle inequality and the multiplicative inequality of 2-norm,

$$\left\| \overline{\boldsymbol{q}}_i - \overline{\boldsymbol{q}}_{\sigma_i(j)} \right\|_2 \leq \sum_{u \in T_{\mathrm{rec}}^{(i,j)}} \|\boldsymbol{M}_u\|_2 \, \|\boldsymbol{d}_{i_u}\|_2 \, . \tag{15}$$

We bound $\|\boldsymbol{M}_u\|_2$ for each non-root node u by the following claim.

Claim 5.6. *Let $l_{\max}$ be the maximum edge length, and let $V_{\min}$ be the minimum tetrahedron volume. Define $c \stackrel{\mathrm{def}}{=} \frac{l_{\max}^3}{6V_{\min}}$, which is a constant by our assumption that all tetrahedrons have constant aspect ratios and all edge lengths are constant. For each non-root u in $T_{rec}^{(i,j)}$, $\|\boldsymbol{M}_u\|_2 \leq c^i$.*

Proof. Let $i = u_1 > u_2 > \ldots > u_t > u_{t+1} = u$ be the vertices along the path P_u. By the proof of Claim 5.5, the length of P_u is at most i. Let $i_k = i_{u_k}$ and $j_k = j_{u_k}$ for each $1 \le k \le t$. By Equations (14), (11), and (12),

$$\begin{aligned}\boldsymbol{M}_u &= \prod_{i_1 \ge i_l \ge i_t} \frac{\boldsymbol{h}_{i_l,j_l}(\boldsymbol{p}_{i_l} - \boldsymbol{p}_{\sigma_{i_l}(j_l)})^\top}{(\boldsymbol{p}_{i_l} - \boldsymbol{p}_{\sigma_{i_l}(j_l)})^\top \boldsymbol{h}_{i_l,j_l}} \\ &= \boldsymbol{h}_{i_1,j_1}\left(\prod_{i_1 \ge i_l \ge i_{t-1}} \frac{(\boldsymbol{p}_{i_l} - \boldsymbol{p}_{\sigma_{i_l}(j_l)})^\top \boldsymbol{h}_{(i_{l+1},j_{l+1})}}{(\boldsymbol{p}_{i_l} - \boldsymbol{p}_{\sigma_{i_l}(j_l)})^\top \boldsymbol{h}_{(i_l,j_l)}}\right) \frac{(\boldsymbol{p}_{i_t} - \boldsymbol{p}_{\sigma_{i_l}(j_t)})^\top}{(\boldsymbol{p}_{i_t} - \boldsymbol{p}_{\sigma_{i_l}(j_t)})^\top \boldsymbol{h}_{(i_t,j_t)}}.\end{aligned}$$

Recall that from Claim 5.1,

$$(\boldsymbol{p}_{i_l} - \boldsymbol{p}_{\sigma_{i_l}(j_l)})^\top \boldsymbol{h}_{(i_l,j_l)} = \det(\boldsymbol{F}_{i_l}),$$

which is equal to the volume of the tetrahedron generated by vectors $\boldsymbol{p}_{i_l} - \boldsymbol{p}_{\sigma_{i_l}(1)}, \boldsymbol{p}_{i_l} - \boldsymbol{p}_{\sigma_{i_l}(2)}$, and $\boldsymbol{p}_{i_l} - \boldsymbol{p}_{\sigma_{i_l}(3)}$. Similarly, $(\boldsymbol{p}_{i_l} - \boldsymbol{p}_{\sigma_{i_l}(j_l)})^\top \boldsymbol{h}_{(i_{l+1},j_{l+1})}$ equals to the volume of the tetrahedron generated by vectors $\boldsymbol{p}_{i_l} - \boldsymbol{p}_{\sigma_{i_l}(j_l)}, \boldsymbol{p}_{i_{l+1}} - \boldsymbol{p}_{\sigma_{i_{l+1}}((j_{l+1}+1)_3)}$, and $\boldsymbol{p}_{i_{l+1}} - \boldsymbol{p}_{\sigma_{i_{l+1}}((j_{l+1}+2)_3)}$. Thus, by our definition $c = l_{\max}^3/(6V_{\min}) \ge V_{\max}/V_{\min}$,

$$\left|\frac{(\boldsymbol{p}_{i_l} - \boldsymbol{p}_{\sigma_{i_l}(j_l)})^\top \boldsymbol{h}_{(i_{l+1},j_{l+1})}}{(\boldsymbol{p}_{i_l} - \boldsymbol{p}_{\sigma_{i_l}(j_l)})^\top \boldsymbol{h}^{(j_l)}_{(i_l,j_l)}}\right| \le c.$$

Thus,

$$\|\boldsymbol{M}_u\|_2 \le \left(\prod_{i_1 \ge i_l \ge i_{t-1}} \left|\frac{(\boldsymbol{p}_{i_l} - \boldsymbol{p}_{\sigma_{i_l}(j_l)})^\top \boldsymbol{h}_{(i_{l+1},j_{l+1})}}{(\boldsymbol{p}_{i_l} - \boldsymbol{p}_{\sigma_{i_l}(j_l)})^\top \boldsymbol{h}_{(i_l,j_l)}}\right|\right) \left\|\frac{\boldsymbol{h}_{(i_1,j_1)}(\boldsymbol{p}_{i_t} - \boldsymbol{p}_{\sigma_{i_t}(j_t)})^\top}{(\boldsymbol{p}_{i_t} - \boldsymbol{p}_{\sigma_{i_t}(j_t)})^\top \boldsymbol{h}_{(i_t,j_t)}}\right\|_2 = O(c^i).$$

$\square$

Applying Claim 5.6 to Equation (15),

$$\left\|\overline{\boldsymbol{q}}_i - \overline{\boldsymbol{q}}_{\sigma_i(j)}\right\|_2 = O\left(c^i \sum_{u \in T_{\text{rec}}^{(i,j)}} \|\boldsymbol{d}_{i_u}\|_2\right).$$

By Claim 5.5 and the fact that $-1 \le i_u \le i$ for each $u \in T_{\text{rec}}^{(i,j)}$,

$$\left\|\overline{\boldsymbol{q}}_i - \overline{\boldsymbol{q}}_{\sigma_i(j)}\right\|_2 = O\left((2c)^i \sum_{-1 \le i' \le i} \|\boldsymbol{d}_{i'}\|_2\right). \tag{16}$$

5.4 Bounding the Quadratic Form $(\overline{\boldsymbol{q}})^\top \boldsymbol{M}\overline{\boldsymbol{q}}$

Note

$$(\overline{\boldsymbol{q}})^\top \boldsymbol{M}\overline{\boldsymbol{q}} \ge \sum_{\substack{-1 \le i \le n-3 \\ 1 \le j \le \min\{i+2,3\}}} \left((\overline{\boldsymbol{q}}_i - \overline{\boldsymbol{q}}_{\sigma_i(j)})^\top (\boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i(j)})\right)^2.$$

By Equation (10),

$$(\overline{\boldsymbol{q}})^\top \boldsymbol{M}\overline{\boldsymbol{q}} \ge \sum_{\substack{-1 \le i \le n-3 \\ 1 \le j \le \min\{i+2,3\}}} \left(\boldsymbol{d}_i^\top (\boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i(j)})\right)^2.$$

By the fact that $\boldsymbol{d}_{-1} = \overline{\boldsymbol{q}}_{-1} - \overline{\boldsymbol{q}}_{-2}$ is parallel to $\boldsymbol{p}_{-1} - \boldsymbol{p}_{-2}$,

$$\left(\boldsymbol{d}_{-1}^\top(\boldsymbol{p}_{-1} - \boldsymbol{p}_{-2})\right)^2 = \Omega\left(\|\boldsymbol{d}_{-1}\|_2^2\right).$$

Claim 5.7 (Lemma 3.7 of [DS07])**.** *Under the assumption: the angle between $\boldsymbol{p}_0 - \boldsymbol{p}_{-1}$ and $\boldsymbol{p}_0 - \boldsymbol{p}_{-2}$ is in the range $[\theta, \pi - \theta]$ for some constant θ. For any fixed unit vector $\boldsymbol{y} \in \mathbb{R}^3$,*

$$\sum_{j \in \{-2,-1\}} \left(\boldsymbol{y}^\top(\boldsymbol{p}_0 - \boldsymbol{p}_j)\right)^2 = \Theta(1).$$

Claim 5.8. *Under the assumption of constant edge lengths: for each $1 \leq i \leq n-3$, the determinant of the matrix:*

$$\det\left(\begin{pmatrix} \boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i(1)} & \boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i(2)} & \boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i(3)} \end{pmatrix}\right) = \Theta(1).$$

Furthermore, for any fixed unit vector $\boldsymbol{y} \in \mathbb{R}^3$, for each $1 \leq i \leq n-3$,

$$\sum_{1 \leq j \leq 3} \left(\boldsymbol{y}^\top\left(\boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i(j)}\right)\right)^2 = \Theta(1).$$

Proof. The determinant of matrix $\begin{pmatrix} \boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i(1)} & \boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i(2)} & \boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i(3)} \end{pmatrix}$ is the signed volume of tetrahedron $t_i = \{i, \sigma_i(1), \sigma_i(2), \sigma_i(3)\}$, which is constant by our assumption.

We claim that for any unit vector $\boldsymbol{y}$, there exists some $j \in \{1,2,3\}$ such that $\boldsymbol{y}^\top(\boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i(j)}) = \Theta(1)$. Assume by contradiction, for every $j \in \{1,2,3\}$ we have $\boldsymbol{y}^\top(\boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i(j)}) = o(1)$. This means that all three vectors $\boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i(1)}, \boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i(2)}, \boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i(3)}$ are between two 2D planes with distance $o(1)$ and orthogonal to $\boldsymbol{y}$. This contradicts the assumption that the volume of tetrahedron $t_i = \{i, \sigma_i(1), \sigma_i(2), \sigma_i(3)\}$ is constant.

Thus, we have

$$\sum_{1 \leq j \leq 3} \left(\boldsymbol{y}^\top\left(\boldsymbol{p}_i - \boldsymbol{p}_{\sigma_i(j)}\right)\right)^2 = \Theta(1).$$

$\square$

Therefore,

$$(\overline{\boldsymbol{q}})^\top \boldsymbol{M} \overline{\boldsymbol{q}} = \Omega\left(\sum_{-1 \leq i \leq n} \|\boldsymbol{d}_i\|_2^2\right).$$

Applying the Cauchy-Schwarz inequality on the above equation and Equation (16) gives the path lemma.

6 Computing a (B, r)-Hollowing of a Convex Edge-simple Truss

In this section, we present Algorithm 2 Hollow, which is used as a subroutine for line 6 of Algorithm 1 TrussSolver. The input of Algorithm 2 consists of a convex edge-simple 3D truss $\mathcal{T} = \langle \{\boldsymbol{p}_i\}_{i \in V}, T, E, \gamma \rangle$, a bounding box B and an integer parameter r. The algorithm outputs a (B, r)-hollowing $\mathcal{H}$ of $\mathcal{T}$. We can check that the algorithm terminates in time $O(|V|)$. These together prove Lemma 3.5.

Given that each individual tetrahedron has constant volume and constant aspect ratio, the following observation converts counting the number of tetrahedrons in a truss intersecting a 2D plane into the intersection area of this truss and the plane.

Algorithm 2 $\textsc{Hollow}(\mathcal{T} = \langle \{\boldsymbol{p}_i\}_{i\in V}, T, E, \gamma\rangle, B, r)$

Input: a convex edge-simple 3D truss $\mathcal{T} = \langle \{\boldsymbol{p}_i\}_{i\in V}, T, E, \gamma\rangle$, a bounding box B, a positive integer $r \le n/\alpha(\mathcal{T})^2$

Output: a (B, r)-hollowing of $\mathcal{T}$

1: Compute r-division planes, which are planes orthogonal to each of the three directions of B such that these planes divide B into $O(n/r)$ small cubes of side length $O(r^{1/3})$ each.

2: $\mathcal{H} \leftarrow$ tetrahedrons intersecting an r-division plane.

3: $\mathcal{H} \leftarrow \mathcal{H} \cup$ tetrahedrons on the boundary of $\mathcal{T}$.

4: Make $\mathcal{H}$ stiffly-connected by adding a minimal number of tetrahedrons in $\mathcal{T}$.

5: **return** $\mathcal{H}$.

Observation 6.1. *Let $\mathcal{T}$ be a convex edge-simple 3D truss, and let P be a 2D plane. Let $A(\mathcal{T} \cap P)$ be the intersection area of $\mathcal{T}$ and P. Then the number of tetrahedrons in $\mathcal{T}$ intersecting P is upper bounded by $O(\max\{A(\mathcal{T} \cap P), 1\})$.*

Proof. Since every individual tetrahedron in $\mathcal{T}$ has constant volume and constant aspect ratio, a tetrahedron of $\mathcal{T}$ intersects P only if all points of this tetrahedron is within some constant distance of P. The number of tetrahedrons of $\mathcal{T}$ intersecting P can be upper bounded by the volume within some constant distance to $\mathcal{T} \cap P$. Thus, the number of tetrahedrons of $\mathcal{T}$ intersecting P is at most $O(\max\{A(\mathcal{T} \cap P), 1\})$. □

6.1 Bounding the Size of $\mathcal{H}$

In this section, we show that $\mathcal{H} = \textsc{Hollow}(\mathcal{T}, B, r)$, computed by Algorithm 2, has a small size. That is, $\mathcal{H}$ satisfies the first condition of the (B, r)-hollowing definition in Definition 3.4.

Note in Algorithm 2 line 1, the bounding box B is divided into $O(n/r)$ small cubes of volume $O(r)$ each. We call a small cube as a *region*. The tetrahedrons in a region are the tetrahedrons of $\mathcal{H}$ which intersect a single small cube. A tetrahedron can appear in at most eight regions.

The following lemma upper bounds the number of tetrahedrons of $\mathcal{H}$ in each region.

Lemma 6.2. *Given a convex edge-simple 3D truss $\mathcal{T}$ of n vertices, a bounding box B and a positive integer $r \le n/\alpha(\mathcal{T})^2$, let $\mathcal{H} = \textsc{Hollow}(\mathcal{T}, B, r)$ returned by Algorithm 2. Then, $\mathcal{H}$ has at most $O(nr^{-1/3})$ tetrahedrons.*

Note the shortest side length of the bounding box of $\mathcal{T}$ is at least $n^{1/3}\alpha^{-2/3}$. The requirement $r \le n/\alpha(\mathcal{T})^2$ guarantees that the shortest side length is at least $r^{1/3}$ so that an r-division exists.

Proof. Note $\mathcal{H}$ has $O(n/r)$ regions. It suffices to show that each region of $\mathcal{H}$ has at most $O(r^{2/3})$ tetrahedrons.

Let R be a region of $\mathcal{H}$. A tetrahedron of $\mathcal{H}$ belongs to region R if either this tetrahedron is within constant distance to the boundary of R, or this tetrahedron is within constant distance to the part of the boundary of $\mathcal{T}$ that's contained in R. Since every tetrahedron of $\mathcal{H}$ has constant volume and aspect ratio, the number of tetrahedrons within constant distance to the boundary of R is $O(r^{2/3})$. It remains to bound the number of tetrahedrons within constant distance to the boundary of $\mathcal{T}$ that's contained in R.

Define $S \stackrel{\text{def}}{=} \mathcal{T} \cap R$. Since both $\mathcal{T}$ and R are convex, S is convex. Let $\mathrm{Surf}(\cdot)$ the surface area of a shape. By Observation 6.1, the number of boundary tetrahedrons of $\mathcal{T}$ contained in R is $O(\mathrm{Surf}(S))$.

Let B_1 be the smallest ball containing S. Since R is a cube, we have

$$\mathrm{Surf}(B_1) = O(\mathrm{Surf}(R)).$$

So it suffices to show $\mathrm{Surf}(S) \leq \mathrm{Surf}(B_1)$. We do so by giving a one-to-one mapping of every point from the surface of S onto B_1.

Let $\phi : S \to B_1$ which maps each face of S to a subset of the surface of B_1, defined as follows. Consider a face of S, say f, with vertices $v_1, \ldots, v_k$ in a clockwise order. Let P_f be the plane containing f. P_f cuts B_1 into two parts, let B_1' be the part of the smaller volume (break a tie arbitrarily), aka the sphere cap generated by the plane p_f.

For each $1 \leq i \leq k$, let P_i be the plane orthogonal to P_f that passes through v_i and v_{i+1}) (if $i = k$, then the intersection line is (v_k, v_1)). Let u_i be the point of intersection of the surface of B_1 with the planes P_{i-1} and P_i (if $i = 1$, then u_1 is the intersect vertex of P_1, P_k and the surface of B_1').

We define $\phi(f)$ to be the surface of B_1' enclosed by $(u_1, \ldots, u_k, u_1)$. See Figure 4 for an example.

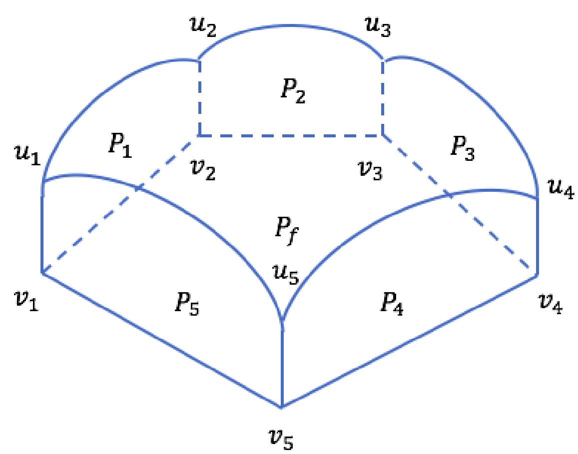

Figure 4: An example of $\phi(f)$. Face f has five vertices $v_1, \ldots, v_5$. $\phi(f)$ is the surface of B_1' enclosed by $(u_1, \ldots, u_5, u_1)$.

For each face f of S, the orthogonal projection of $\phi(f)$ onto the plane P_f is f. Thus

$$\textsc{Area}(\phi(f)) \geq \textsc{Area}(f).$$

In addition, since S is convex, for any two distinct faces f_1 and f_2, $\phi(f_1)$ and $\phi(f_2)$ are also disjoint. So we have

$$\mathrm{Surf}\left(S\right) = \sum_{f \in S} \textsc{Area}\left(f\right) \leq \sum_{f \in S} \textsc{Area}\left(\phi\left(f\right)\right) \leq \mathrm{Surf}\left(B_1\right).$$

Combining this with $\mathrm{Surf}(B_1) \leq O(\mathrm{Surf}(S)) \leq O(r^{2/3})$ then completes the proof. $\square$

6.2 Bounding the Number of Tetrahedrons in $\mathcal{H}$ Intersecting with a Plane

In this section, we show that $\mathcal{H} = \textsc{Hollow}(\mathcal{T}, B, r)$, computed by Algorithm 2, has a small overlap with any plane whose normal vector has an angle between $(0, \pi/2)$ with the longest direction of B. That is, $\mathcal{H}$ satisfies the second condition of the (B, r)-hollowing definition in Definition 3.4.

Lemma 6.3. *Given a convex edge-simple 3D truss $\mathcal{T}$ of n vertices, a bounding box B and a positive integer $r \le n/\alpha(\mathcal{T})^2$, let $\mathcal{H} = $ HOLLOW$(\mathcal{T}, B, r)$ returned by Algorithm 2. Let $\boldsymbol{d} \in \mathbb{R}^3$ be a unit vector such that the angle between $\boldsymbol{d}$ and let the angles with the three directions of the box (normals to its faces) be $\theta_x, \theta_y, \theta_z \in (\theta, \pi/2)$, for some $\theta > 0$. Then, the number of tetrahedrons in $\mathcal{H}$ which intersect any plane P orthogonal to $\boldsymbol{d}$ is at most*

$$O\left(\frac{n^{2/3}}{\alpha(\mathcal{T})^{1/3} r^{1/3} \cos^2\theta}\right).$$

Without loss of generality, we assume that the bounding box B is *axis-parallel*, that is, the sides of B are parallel to the three axes: the x axis, the y axis and the z-axis. We say an axis-paralleled box has side lengths $a, b, c > 0$, if the sides parallel to the x-axis have length a, the sides parallel to the y-axis have length b, and the sides parallel to the z-axis have length c, respectively.

To prove Lemma 6.3, we need the following claim, which bounds the intersection area of a 2D plane and a 3D box.

Claim 6.4. *Let B be a 3D axis-parallel box of side lengths $a, b, c > 0$. Let $\boldsymbol{d} \in \mathbb{R}^3$ be a unit vector such that the angle between $\boldsymbol{d}$ and the x-axis (the y-axis, and the z-axis) is θ_x (θ_y, θ_z, respectively). Suppose $\theta_x, \theta_y, \theta_z \in (0, \pi/2)$. Then the intersection area S of any 2D plane P orthogonal to $\boldsymbol{d}$ and the 3D box B satisfies*

$$S \le \min\left\{\frac{bc}{\cos\theta_x}, \frac{ac}{\cos\theta_y}, \frac{ab}{\cos\theta_z}\right\}.$$

Proof. Since the three terms in the right hand side are symmetric, we only prove S can be upper bounded by the first term and the other two follow in a similar way.

Let B_{x1}, B_{x2} be the two faces of B which are orthogonal to the x-axis, without loss of generality, assume B_{x1} has a smaller x-coordinate.

If P intersects neither B_{x1} nor B_{x2}, then the volume of B is equal to $Sa\cos\theta_x$.

If P intersects B_{x1} say with line (K, L), see Figure 5, then we draw a line going through point K and parallel to the x-axis, which intersects face B_{x2} at point M, similarly we draw a line going through point L and parallel to the x-axis, which intersects face B_{x2} at point N. We cut the box B by the plane $KMLN$, see Figure 5. Note that the volume of the convex hull of (K, G, E, F, G, H, I, J), the right one in Figure 5, equals to $Sa\cos\theta_x$, which is smaller than the volume of B.

Similarly, if P intersects B_{x2}, then we can draw two lines parallel to the x-axis and going through the two intersection points respectively and get a shape of volume $Sa\cos\theta_x$ smaller than the volume of B. Note that the intersection between P and B_{x2} cannot coincide with (M, N), otherwise the angle $\theta_x = \pi/2$. We can check that the shape we get must contain $P \cap B$, and the two faces of the shape which are orthogonal to the x-axis are congruent. If we cut the shape along the plane P, then we can shift the left part along the x-axis and glue the two faces which are orthogonal to the x-axis, and get a parallelepiped which has $P \cap B$ as a face. Figure 6 shows the parallelepiped we get from the example of Figure 5. Thus,

$$S \cdot a \cdot \cos\theta_x \le abc.$$

That is, $S \le bc/\cos\theta_x$. This completes the proof. □

Now we prove Lemma 6.3.

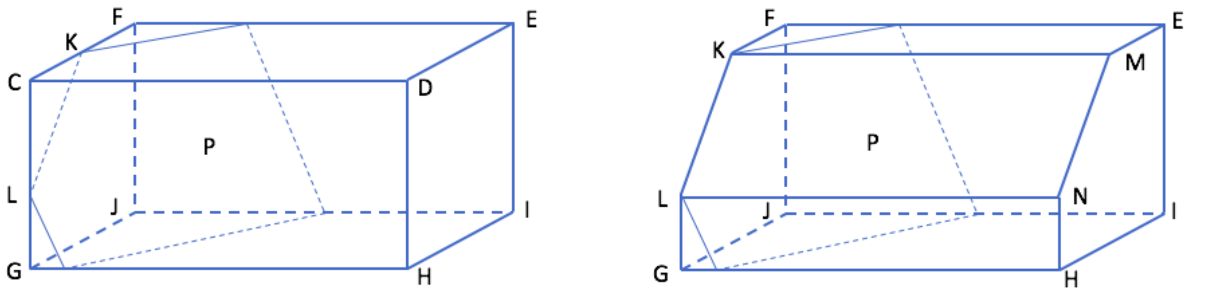

Figure 5: Plane P intersects face B_{x1} with line (K, L). We draw a line going through point K and parallel to the x-axis, which intersects face B_{x2} at point M, similarly we draw a line going through point L and parallel to the x-axis, which intersects face B_{x2} at point N.

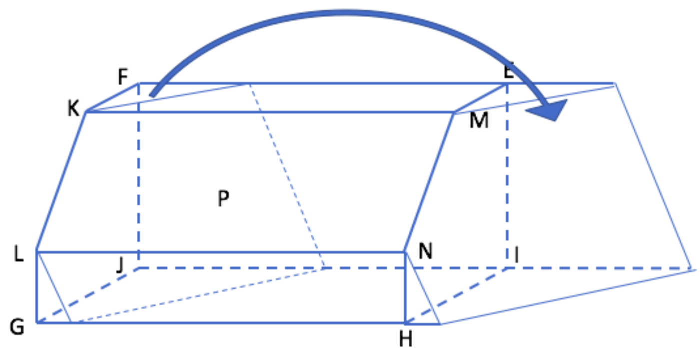

Figure 6: Cut the shape along the plane P, and shift the left part along the x-axis and glue the two faces $KFJGL$ and $MEIHN$.

Proof of Lemma 6.3. Let t be the number of regions (that is, small cubes of the hollowing) of $\mathcal{H}$ which intersect a 2D plane P, and let m be the maximum number of tetrahedrons of a single region of $\mathcal{H}$ which intersect the plane P. The number of tetrahedrons in the hollowing $\mathcal{H}$ which intersect P can be upper bounded by tm.

We first bound t, the number of regions of $\mathcal{H}$ which intersect the plane P. A cube region intersects the plane P only if all its points are within a distance $\sqrt{3}r^{1/3}$ of P. We put two planes, say P_1, P_2, which are parallel to the plane P, above and below P with distance $\sqrt{3}r^{1/3}$ to P. All cube regions which intersect the plane P must be within the two planes P_1 and P_2. By Claim 6.4, the volume of the box B between the two planes P_1, P_2 is at most

$$\frac{bc}{\cos\theta} \cdot 2\sqrt{3}r^{1/3}.$$

Since each cube has volume at most r, we can bound the number of cube regions which intersect the plane P

$$t \leq \frac{2\sqrt{3}bc}{r^{2/3}\cos\theta}.$$

We then bound m, the maximum number of tetrahedrons of a single region of $\mathcal{H}$ which intersect the plane P. Consider the tetrahedrons of $\mathcal{H}$ in this single cube region which intersect a some hollowing plane. These tetrahedrons are within constant distance of an $r^{1/3} \times r^{1/3}$ square, which is on a plane orthogonal to one of the direction of the bonding box. By Observation 6.1 and Claim 6.4, the number of these tetrahedrons which intersect the plane P is at most $O(r^{1/3}/\cos\theta)$. Thus, we can bound the maximum number of tetrahedrons of a single cube region which intersect the plane P

$$m \leq O\left(\frac{r^{1/3}}{\cos\theta}\right).$$

Therefore, the number of tetrahedrons of the hollowing $\mathcal{H}$ which intersect the plane P is at most

$$tm \leq O\left(\frac{bc}{r^{1/3}\cos^2\theta}\right).$$

Note that we have $a = c \cdot \alpha(\mathcal{T})$ and $a \geq b \geq c$. The above number is upper bounded by

$$O\left(\frac{n^{2/3}}{\alpha(\mathcal{T})^{1/3} r^{1/3}\cos^2\theta}\right).$$

□

6.3 Bounding the Relative Condition Number of $\mathcal{H}$

In this section, we show that the condition number of $\boldsymbol{A}_{\mathcal{H}}$ and $\textsc{Sc}[\boldsymbol{A}_{\mathcal{T}}]_U$ is small. That is, $\mathcal{H}$ satisfies the third condition of the (B, r)-hollowing definition in Definition 3.4.

Since $\mathcal{T}$ is convex, each region of $\mathcal{H}$ is connected. Lemma 3.3 implies that $\boldsymbol{A}_{\mathcal{H}}$ and $\textsc{Sc}[\boldsymbol{A}_{\mathcal{T}}]_U$ has the same null space. By Lemma 3.3, the smallest nonzero eigenvalue of $\boldsymbol{A}_{\mathcal{H}}$ is lower bounded by the diameter and the size of each region of $\mathcal{H}$.

Lemma 6.5. *Let $\boldsymbol{A}_{\mathcal{T}}$ be the truss stiffness matrix of a convex edge-simple 3D truss $\mathcal{T}$. Given a positive integer r and a bounding box B, let $\mathcal{H} = \textsc{Hollow}(\mathcal{T}, B, r)$ be returned by Algorithm 2. Let $\boldsymbol{A}_{\mathcal{H}}$ be the associated truss stiffness matrix of $\mathcal{H}$. Then,*

$$\boldsymbol{A}_{\mathcal{H}} \preccurlyeq \textsc{Sc}[\boldsymbol{A}_{\mathcal{T}}]_U \preccurlyeq O(r^2)\boldsymbol{A}_{\mathcal{H}},$$

where U consists of all vertices in $\mathcal{H}$.

Proof. The first inequality is equivalent to: $\textsc{Sc}[\boldsymbol{A}_{\mathcal{T}}]_U - \boldsymbol{A}_{\mathcal{H}}$ is a symmetric PSD matrix. Let $\mathcal{T}'$ be the truss obtained by removing all the edges of $\mathcal{T}$ whose two endpoints are both in U. Let $\boldsymbol{A}_{\mathcal{T}'}$ be the associated truss stiffness matrix of $\mathcal{T}'$. We can check that

$$\textsc{Sc}[\boldsymbol{A}_{\mathcal{T}}]_U - \boldsymbol{A}_{\mathcal{H}} = \textsc{Sc}[\boldsymbol{A}_{\mathcal{T}'}]_U.$$

By Fact 2.6, $\textsc{Sc}[\boldsymbol{A}_{\mathcal{T}}]_U - \boldsymbol{A}_{\mathcal{H}}$ is a symmetric PSD matrix.

It remains to prove the second inequality. Note the planes of Algorithm 2 line 1 divide $\mathcal{T}$ into small regions. Let $\mathcal{T}_i$ denote the subgraph induced by $\mathcal{T}$ on the ith region, and let $\mathcal{T}_{C_i}$ denote the subgraph induced by $\mathcal{T}$ on the boundary of the ith region. Let $\boldsymbol{A}_{\mathcal{T}_i}, \boldsymbol{A}_{\mathcal{T}_{C_i}}$ be the associated truss stiffness matrices of $\mathcal{T}_i$ and $\mathcal{T}_{C_i}$ respectively. Let U_i denote the boundary vertices of the ith region. Let $\textsc{Sc}[\boldsymbol{A}_{\mathcal{T}_i}]_{U_i}$ be the Schur complement of $\boldsymbol{A}_{\mathcal{T}_i}$ w.r.t. to U_i.

Since $\mathcal{T}$ is a convex edge-simple 3D truss, each vertex in $\mathcal{T}_i$ has constant degree and each edge has constant length and elasticity parameter. It implies $\lambda_{\max}(\boldsymbol{A}_{\mathcal{T}_i}) = O(1)$. By Fact 2.6, $\lambda_{\max}(\textsc{Sc}[\boldsymbol{A}_{\mathcal{T}_i}]_{U_i}) = O(1)$.

According to Algorithm 2 from line 2 to line 4, in each $\mathcal{T}_{C_i}$, the tetrahedrons are arranged in simplicial complex and $\mathcal{T}_{C_i}$ is connected. By Lemma 3.3, the null spaces of $\boldsymbol{A}_{\mathcal{T}_{C_i}}$ and the null space of $\textsc{Sc}[\boldsymbol{A}_{\mathcal{T}_i}]_{U_i}$ are the same. Besides, each $\mathcal{T}_{C_i}$ has $O(r^{2/3})$ vertices and diameter $O(r^{1/3})$, by Lemma 6.2. Applying Lemma 3.3 gives:

$$\lambda_{\min}(\boldsymbol{A}_{\mathcal{T}_{C_i}}) = \Omega\left(\frac{1}{r^2}\right).$$

This implies:

$$\textsc{Sc}[\boldsymbol{A}_{\mathcal{T}_i}]_{U_i} \preccurlyeq O(r^2)\boldsymbol{A}_{\mathcal{T}_{C_i}}.$$

Note each edge of $\mathcal{T}$ only appears in a constant number of regions. Thus,

$$\textsc{Sc}[\boldsymbol{A}_{\mathcal{T}}]_U \preccurlyeq \sum_i \textsc{Sc}[\boldsymbol{A}_{\mathcal{T}_i}]_{U_i} \preccurlyeq O(r^2)\sum_i \boldsymbol{A}_{\mathcal{T}_{C_i}} \preccurlyeq O(r^2)\boldsymbol{A}_{\mathcal{H}}.$$

This completes the proof. □

7 Nested Dissection

In this section, we present our solver for the constructed preconditioners, Algorithm 4 ConvexTrussUnionND, which is used as a subroutine in line 8 of Algorithm 1 TrussSolver.

We first prove Lemma 2.8 in Section 7.1. Then, in Section 7.2, we use Lemma 2.8 to analyze the performance of Algorithm 4, which proves Lemma 3.7.

7.1 Proof of Lemma 2.8

We restate Lemma 2.8 with running time phrased in terms of the matrix multiplication exponent ω.

Lemma 2.8. *Suppose we have a recursive separator decomposition of a simplicial complex with n bounded aspect ratio tetrahedrons such that:*

1. *the number of leaves, and hence total number of recursive calls, is at most n^α.*
2. *each leaf (bottom layer partition) has at most n^β tetrahedrons.*
3. *each top separator has size at most n^γ.*

Then we can find an exact Cholesky factorization of the associated stiffness matrix with multiplication count $O(n^{\alpha+2\omega\beta/3} + n^{\alpha+\gamma\omega}\log n)$ and fill-in size $O(n^{\alpha+\frac{4}{3}\beta} + n^{\alpha+2\gamma})$.

Nested dissection according to the separator decomposition stated in Lemma 2.8 has three parts of cost:

1. Inverting leaf components:
   (a) the cost only associated with vertices not belonging to top-level separators;
   (b) the cost associated with top-level separators.

2. Inverting top-level separators.

We first analyze the cost of inverting top-level separators. Note that top-level separators are disjoint. After eliminating all leaf components, we get a *layered graph*. That is, its vertices can be partitioned into $V_1 \cup \ldots \cup V_l$ for some positive integer l such that there is no edge between V_i and V_j for $|i-j| > 1$. Algorithm 3 gives a numbering for a layered graph, and Claim 7.1 analyzes its performance.

Algorithm 3 LayeredGraphND($G = (V_1 \cup \ldots \cup V_l, E)$)

Input: a layered graph G with l layers
Output: an elimination ordering for vertices in G

```
if l = 1 then
    number the vertices in V_1 arbitrarily and return this numbering.
end if
Label V_{⌊l/2⌋} with the highest possible numbers.
return the numbering of LayeredGraphND(G[V_1 ∪ ... ∪ V_{⌊l/2⌋−1}] and
    LayeredGraphND(G[V_{⌊l/2⌋+1} ∪ ... ∪ V_l], and V_{⌊l/2⌋}.
```

Claim 7.1. *Let G be a layered graph with l layers of at most s vertices each. Algorithm 3* LayeredGraphND *returns a numbering such that, Gaussian elimination according to this order has fill-in size $O(s^2 l)$ and multiplication count $O(s^\omega l)$.*

Proof. We follow the proofs of [LRT79]. We first bound the fill-in size. Consider a recursion of Algorithm 3 on a graph with n vertices, let $f(n)$ denote the maximum number of fill-in edges whose lower numbered endpoint is numbered by this recursion. Suppose this recursion deals with layers $V_s, V_{s+1} \ldots, V_t$. The algorithm numbers the vertices in the middle layer, that is, $V_{\lfloor (s+t)/2 \rfloor}$, which can be viewed as a separator whose removal separates the graph into two disjoint and balanced parts. The fill-in edges whose lower numbered endpoint is in $V_{\lfloor (s+t)/2 \rfloor}$ consists of the following edges: (1) edges whose both endpoints are in $V_{\lfloor (s+t)/2 \rfloor}$; and (2) edges whose one endpoint is in $V_{\lfloor (s+t)/2 \rfloor}$ and the other is in V_{s-1} (if exists) or V_{t+1} (if exists). Since each layer has at most s vertices,

$$f(n) \leq \begin{cases} O(s^2), & \text{if } n = O(s) \\ f(n_1) + f(n_2) + 3s^2, & \text{otherwise} \end{cases}$$

Note $n_2 \leq n_1 \leq n_2 + s$. Thus, the total fill-in size is $O(s^2 l)$.

We then bound the multiplication count. For a recursion of Algorithm 3 on a graph with n vertices, let $g(n)$ denote the maximum multiplication count associated with vertices in this graph which are going to be numbered in this recursion. Similar to the analysis of the fill-in, we have

$$g(n) \leq \begin{cases} O(s^\omega), & \text{if } n = O(s) \\ g(n_1) + g(n_2) + 3s^\omega, & \text{otherwise} \end{cases}$$

Thus, the total multiplication count is $O(s^\omega l)$. □

Using Algorithm 3 as a subroutine, we prove Lemma 2.8.

Proof of Lemma 2.8. We first bound the fill-in size. Note each leaf component has $O(n^\beta)$ vertices in which $O(n^\gamma)$ vertices are in some top-level separators and are labeled by higher numbers. By the result in [LRT79] and [MT90], the fill-in introduced by inverting each leaf component is

$$O\left(n^{4\beta/3} + n^{2\beta/3+\gamma}\right).$$

There are totally $O(n^{\alpha})$ leaf components. Thus, the fill-in size introduced by inverting all leaf components is $O\left(n^{\alpha+4\beta/3}+n^{\alpha+2\beta/3+\gamma}\right)$.

By Claim 7.1, the fill-in size of inverting top-level separators is $O(n^{\alpha+2\gamma})$. Thus, the total fill-in size is

$$O\left(\alpha+n^{4\beta/3}+n^{\alpha+2\beta/3+\gamma}+n^{\alpha+2\gamma}\right)=O\left(n^{\alpha+4\beta/3}+n^{\alpha+2\gamma}\right).$$

Here, we use Cauchy-Schwarz inequality to drop the second term.

We then bound the multiplication count. By [MT90], when inverting each leaf component, the multiplication count only associated with vertices not belonging to top-level separators is $O(n^{2\omega\beta/3})$. By Claim 7.1, the multiplication count of inverting top-level separators is $O(n^{\alpha+\omega\gamma})$.

It remains to upper bound the multiplication count associated with top-level separators when inverting a leaf component. We adapt the analysis in [LRT79].

We prove a more general result. Consider a tetrahedral mesh G with n vertices, in which s vertices are in a top-level separator and have been labeled with higher numbers. Let $g(n,s)$ be the multiplication count associated these s top-level vertices when running nested dissection on G.

Nested dissection finds a separator of size $n^{2/3}$ for G. The multiplication count associated with the s top-level vertices when inverting this separator can be upper bounded by $O((n^{2/3}+s)^{\omega})$. This gives the following recursion:

$$\begin{aligned} g(n,s) &\leq O((n^{2/3}+s)^{\omega})+\max_{n_i,s_i}\{\sum_{1\leq i\leq 2} g(n_i,s_i)\} \\ &\leq O(2^{\omega}n^{2\omega/3}+2^{\omega}s^{\omega})+\max_{n_i,s_i}\{\sum_{1\leq i\leq 2} g(n_i,s_i)\}. \end{aligned}$$

Here, the maximum is taken over

$$\begin{aligned} s_1+s_2 &\leq s+n^{2/3} \\ n_1+n_2 &\leq n+n^{2/3} \\ n_1,n_2 &\leq \frac{4}{5}n+n^{2/3}. \end{aligned}$$

We can compute that

$$g(n,s)=O\left(n^{2\omega/3}+s^{\omega}\log n\right).$$

Note each leaf component is incident to at most two top-level separators. Thus, the multiplication count associated with the top-level vertices when inverting this leaf component is

$$O\left(n^{2\omega\beta/3}+n^{\gamma\omega}\log n\right).$$

Combining the other two parts of cost, and the fact that there are totally n^{α} leaf components, the total multiplication count is

$$O\left(n^{\alpha+2\omega\beta/3}+n^{\alpha+\gamma\omega}\log n\right).$$

If we replace ω by 3, then we can drop the $\log n$ factor in the above equation, which gives the result in the statement. $\square$

Algorithm 4 ConvexTrussUnionND($\mathcal{T} = \langle \{\boldsymbol{p}_i\}_{i\in V}, T, E, \gamma\rangle, \mathcal{B}, \mathcal{I}, \mathcal{H}, l$)

Input: a edge-simple 3D truss $\mathcal{T}$ which is the union of k convex edge-simple trusses $\mathcal{T}_1, \ldots, \mathcal{T}_k$,
$\mathcal{B} = \{B_1, \ldots, B_k\}$ in which B_i is a bounding box of $\mathcal{T}_i$ constructed from Line 2 of Algorithm 1,
Index set $\mathcal{I}$ of large aspect ratio complexes constructed from Line 4 of Algorithm 1,
Hollowings of each $\mathcal{T}_i$ with $i \in \mathcal{I}$ constructed from Line 6 of Algorithm 1,
l, the number of top-level partitions.

Output: an elimination ordering for vertices in $(\cup_{i\in\mathcal{I}}\mathcal{H}_i)\bigcup(\cup_{i\notin\mathcal{I}}\mathcal{T}_i)$

For each $1 \le i \le k$, $\boldsymbol{d}_i \leftarrow$ the direction of the longest sides of B_i.
Compute a unit vector $\boldsymbol{d}$ such that $\frac{1}{10k} \le \left|\boldsymbol{d}^\top \boldsymbol{d}_i\right| \le 1 - \frac{1}{10k}, \forall 1 \le i \le k$.
Compute separator planes $P_1, \ldots, P_l$, which are planes orthogonal to $\boldsymbol{d}$ and dividing $\mathcal{T}$ into $l+1$ parts $Q_1, \ldots, Q_{l+1}$ of $O(nl^{-1})$ tetrahedrons each.
For each $1 \le j \le l$, $S_j \leftarrow$ tetrahedrons in $(\cup_{i\in\mathcal{I}}\mathcal{H}_i)\bigcup(\cup_{i\notin\mathcal{I}}\mathcal{T}_i)$ which intersect plane P_j.
$Q_1 \leftarrow Q_1 \cup S_1, Q_{l+1} \leftarrow Q_{l+1} \cup S_l$.
For each $2 \le j \le l, Q_j \leftarrow Q_j \cup S_{j-1} \cup S_j$.
LayeredGraphND($\mathcal{T}[S_1 \cup \ldots \cup S_l]$) with highest numbers.
Run nested dissection with MT-separators[5] for each Q_j to number its unnumbered vertices.
return the elimination ordering of vertices in $(\cup_{i\in\mathcal{I}}\mathcal{H}_i)\bigcup(\cup_{i\notin\mathcal{I}}\mathcal{T}_i)$.

7.2 Proof of Lemma 3.7

We now present Algorithm 4 ConvexTrussUnionND. The input consists of a edge-simple 3D truss $\mathcal{T}$ which is a union of k convex edge-simple trusses, a bounding box for each convex edge-simple truss, the index subset of small-aspect-ratio trusses and (B_i, r_i)-hollowings for each small-aspect-ratio truss. The output is an elimination ordering for the union of the hollowings of small-aspect-ratio trusses and the large-aspect-ratio trusses. This proves Lemma 3.7.

We first prove that there exists a good direction $\boldsymbol{d}$ such that: the angle between $\boldsymbol{d}$ and the longest direction of each bounding box is in a proper range.

Lemma 7.2. *Let $k \ge 2$ and $\boldsymbol{d}_1, \ldots, \boldsymbol{d}_k \in \mathbb{R}^3$ be k unit vectors. Then there exists a unit vector $\boldsymbol{d} \in \mathbb{R}^3$ such that*

$$\frac{1}{10k} \le \left|\boldsymbol{d}^\top \boldsymbol{d}_i\right| \le 1 - \frac{1}{10k}, \forall 1 \le i \le k. \tag{17}$$

Proof. We pick a unit vector $\boldsymbol{d}$ uniformly at random. For any fixed i,

$$\Pr\left(\frac{1}{10k} \le \left|\boldsymbol{d}^\top \boldsymbol{d}_i\right| \le 1 - \frac{1}{10k}\right) = 2 \cdot \frac{\mathrm{vol}(1/10k) - \mathrm{vol}(1 - 1/10k)}{V}.$$

Here, $\mathrm{vol}(x)$ is the volume of a cap of a 3D unit ball with height $1-x$, and V is the volume of a 3D unit ball. We have $\mathrm{vol}(x) = \frac{1}{6}\pi(1-x^2)(3(1-x^4)+(1-x^2)^2)$ and $V = \frac{4}{3}\pi$. Plugging these volumes into the above equation,

$$\Pr\left(\frac{1}{10k} \le \left|\boldsymbol{d}^\top \boldsymbol{d}_i\right| \le 1 - \frac{1}{10k}\right) = \frac{1}{2}\left(\left(1 - \frac{1}{10k}\right)^2\left(2 + \frac{1}{10k}\right) - \left(\frac{1}{10k}\right)^2\left(3 - \frac{1}{10k}\right)\right).$$

Take the opposite:

$$\Pr\left(\left|\boldsymbol{d}^\top \boldsymbol{d}_i\right| < \frac{1}{10k} \text{ or } \left|\boldsymbol{d}^\top \boldsymbol{d}_i\right| > 1 - \frac{1}{10k}\right) \le \frac{3}{20k} - \frac{1}{2000k^3} + \frac{3}{200k^2}.$$

By union bound,

$$\Pr\left(\exists i\ s.t.\ \left|\boldsymbol{d}^\top \boldsymbol{d}_i\right| < \frac{1}{10k} \text{ or } \left|\boldsymbol{d}^\top \boldsymbol{d}_i\right| > 1 - \frac{1}{10k}\right) \leq k\left(\frac{3}{20k} - \frac{1}{2000k^3} + \frac{3}{200k^2}\right) \leq \frac{1}{5}.$$

Thus,

$$\Pr\left(\forall i, \frac{1}{10k} \leq \left|\boldsymbol{d}^\top \boldsymbol{d}_i\right| \leq 1 - \frac{1}{10k}\right) \geq \frac{4}{5}.$$

This implies there exists a $\boldsymbol{d}$ as desired. $\square$

We independently pick $O(\log n)$ unit vectors uniformly at random. By a Chernoff bound, we can find a direction $\boldsymbol{d}$ satisfying Equation (17) with high probability.

Recall that Lemma 6.3 states that: any 2D plane P orthogonal to $\boldsymbol{d}$ intersects a small number of tetrahedrons in a (B, r)-hollowing of a convex edge-simple 3D truss. Lemma 3.6 bounds the number of tetrahedrons in a convex edge-simple 3D truss which intersect a single 2D plane P orthogonal to $\boldsymbol{d}$. We restate it in the following, which can be proved by combining Observation 6.1 and Claim 6.4.

Lemma 3.6. *Let $\mathcal{T}$ be a convex edge-simple 3D truss of n vertices, and let B be a bounding box of $\mathcal{T}$. Let $\boldsymbol{d} \in \mathbb{R}^3$ be a unit vector such that the angle between $\boldsymbol{d}$ and the longest direction of B is $\theta \neq \pi/2$. Then any plane P orthogonal to $\boldsymbol{d}$ intersects $\mathcal{T}$ in at most $O(n^{2/3}\alpha(\mathcal{T})^{-1/3}\cos^{-1}\theta)$ tetrahedrons.*

Lemma 6.3 and Lemma 3.6 together imply that: the top-level separator $S_1 \cup \ldots \cup S_l$ computed in Algorithm 4 has a small size. This, together with nested dissection in [MT90], lets us prove that Algorithm 4 outputs an elimination ordering with a small fill-in size and multiplication count.

Lemma 7.3. *Given a edge-simple 3D truss $\mathcal{T}$ of n vertices which is a union of k convex edge-simple trusses with n_i vertices each, running Algorithm 1,* TrussSolver$(\mathcal{T}, \boldsymbol{f}, \epsilon, c_\alpha, c_r)$*, with Line 8 replaced by Algorithm 4,* ConvexTrussUnionND$(\mathcal{T}, \mathcal{B}, \mathcal{I}, \mathcal{H}, l)$*, leads to performance in terms of n that is optimized by setting*

$$c_\alpha = c_r \leq \frac{1}{3}$$

in Line 4 of Algorithm 1, TrussSolver*. In terms of c_r, the hollowing parameter, and l, the number of top-level separators, this gives an elimination ordering with fill-in size at most*

$$O(n^{4/3}l^{-1/3} + k^{14/3+2c_r/3}n^{4/3-2c_r/3}l),$$

that can be computed in time

$$O(n^{2\omega/3}l^{-2\omega/3+1} + k^{7\omega/3+\omega c_r/3}n^{2\omega/3-c_r\omega/3}l\log n),$$

where ω is the matrix multiplication exponent.

Proof. We apply Lemma 2.8. According to Algorithm 4 line 2, for each i, the angle between the longest direction of the ith bounding box and $\boldsymbol{d}$ has cosine value in $[1/10k, 1 - 1/10k]$.

We first upper bound the number of vertices in each top-level separators, that is, the number of tetrahedrons in $(\cup_{i\in\mathcal{I}}\mathcal{H}_i)\bigcup(\cup_{i\notin\mathcal{I}}\mathcal{T}_i)$ which intersects a plane P_j, see Algorithm 4 line 4. For each $i \in \mathcal{I} \stackrel{\text{def}}{=} \{i \in [k] : \alpha(\mathcal{T}_i) \leq n_i^{c_\alpha}\}$, by Lemma 6.3, the number of tetrahedrons in $\mathcal{H}_i$ intersect a plane P_j is at most

$$O\left(k^2 n_i^{2/3}\alpha_i^{-1/3}r_i^{-1/3}\right) = O\left(k^2 n_i^{2/3-c_r/3}\right),$$

since $\alpha_i \geq 1$. For each $i \notin \mathcal{I}$, by Lemma 3.6, the number of tetrahedrons in $\mathcal{T}_i$ intersect a plane P_j is at most

$$O\left(kn_i^{2/3}\alpha_i^{-1/3}\right) = O\left(kn_i^{2/3-c_\alpha/3}\right)$$

since $\alpha_i > n_i^{c_\alpha}$. The two terms have same exponent for n when we set $c_r = c_\alpha$. Note Algorithm 2 requires that $c_r + 2c_\alpha \leq 1$. Thus, here we need $c_r \leq 1/3$.

Thus, the total number of tetrahedrons in $(\cup_{i\in\mathcal{I}}\mathcal{H}_i)\bigcup(\cup_{i\notin\mathcal{I}}\mathcal{T}_i)$ which intersect a single separator plane P_j is then at most:

$$s \stackrel{\text{def}}{=} O\left(\sum_{1\leq i\leq k} k^2 n_i^{2/3-c_r/3}\right) = O\left(k^{7/3+c_r/3}n^{2/3-c_r/3}\right).$$

The last inequality is by Jensen's inequality.

There are totally l top-level separators, which separates the whole truss into $l+1$ separate components and each component has $O(n/l)$ vertices, according to Algorithm 4 line 3.

We plug these parameters into Lemma 2.8, the total fill-in size is

$$O\left(\left(\frac{n}{l}\right)^{4/3} l + s^2 l\right) = O(n^{4/3}l^{-1/3} + k^{14/3+2c_r/3}n^{4/3-2c_r/3}l),$$

and the multiplication count is

$$O\left(\left(\frac{n}{l}\right)^{2\omega/3} l + s^\omega l \log n\right) = O(n^{2\omega/3}l^{-2\omega/3+1} + k^{7\omega/3+\omega c_r/3}n^{2\omega/3-c_r\omega/3}l\log n).$$

This completes the proof. □

References

[Axe85] Owe Axelsson. A survey of preconditioned iterative methods for linear systems of algebraic equations. *BIT Numerical Mathematics*, 25(1):165–187, 1985.

[AY10] Noga Alon and Raphael Yuster. Solving linear systems through nested dissection. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 225–234. IEEE, 2010.

[BCFN16] Glencora Borradaile, Erin Wolf Chambers, Kyle Fox, and Amir Nayyeri. Minimum cycle and homology bases of surface embedded graphs. *arXiv preprint arXiv:1607.05112*, 2016.

[BEG94] Marshall Bern, David Eppstein, and John Gilbert. Provably good mesh generation. *Journal of Computer and System Sciences*, 48(3):384–409, 1994.

[BENWN14] Glencora Borradaile, David Eppstein, Amir Nayyeri, and Christian Wulff-Nilsen. All-pairs minimum cuts in near-linear time for surface-embedded graphs. *arXiv preprint arXiv:1411.7055*, 2014.

[BHP01] Gill Barequet and Sariel Har-Peled. Efficiently approximating the minimum-volume bounding box of a point set in three dimensions. *J. Algorithms*, 38(1):91–109, January 2001.

[BKM+11a] Glencora Borradaile, Philip N Klein, Shay Mozes, Yahav Nussbaum, and Christian Wulff-Nilsen. Multiple-source multiple-sink maximum flow in directed planar graphs in near-linear time. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 170–179. IEEE, 2011.

[BKM+11b] Glencora Borradaile, Philip N Klein, Shay Mozes, Yahav Nussbaum, and Christian Wulff-Nilsen. Multiple-source multiple-sink maximum flow in directed planar graphs in near-linear time. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 170–179. IEEE, 2011. Available at: https://arxiv.org/abs/1105.2228.

[BSS13] Afonso S Bandeira, Amit Singer, and Daniel A Spielman. A cheeger inequality for the graph connection laplacian. *SIAM Journal on Matrix Analysis and Applications*, 34(4):1611–1630, 2013.

[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, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 204–216, 2014. Available at: https://www.cs.cmu.edu/~glmiller/Publications/Papers/CoFMNPW14.pdf.

[Che89] L Paul Chew. Guaranteed-quality triangular meshes. Technical report, Cornell University, 1989.

[Chu96] Fan RK Chung. Laplacians of graphs and cheegerâĂŹs inequalities. *Combinatorics, Paul Erdos is Eighty*, 2(157-172):13–2, 1996.

[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.

[CKP+17] Michael B. Cohen, Jonathan Kelner, John Peebles, Richard Peng, Anup B. Rao, Aaron Sidford, and Adrian Vladu. Almost-linear-time algorithms for markov chains and new spectral primitives for directed graphs. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, pages 410–419, New York, NY, USA, 2017. ACM.

[DS07] Samuel I Daitch and Daniel A Spielman. Support-graph preconditioners for 2-dimensional trusses. *arXiv preprint cs/0703119*, 2007.

[EN11] Jeff Erickson and Amir Nayyeri. Computing replacement paths in surface embedded graphs. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 1347–1354. Society for Industrial and Applied Mathematics, 2011.

[Fed64] Radii Petrovich Fedorenko. The speed of convergence of one iterative process. *USSR Computational Mathematics and Mathematical Physics*, 4(3):227–235, 1964.

[Fre87] Greg N. Frederickson. Fast algorithms for shortest paths in planar graphs, with applications. *SIAM J. Comput.*, 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.

[GNP94] John R Gilbert, Esmond G Ng, and Barry W Peyton. An efficient algorithm to compute row and column counts for sparse cholesky factorization. *SIAM Journal on Matrix Analysis and Applications*, 15(4):1075–1091, 1994.

[Goo95] Michael T Goodrich. Planar separators and parallel polygon triangulation. *Journal of Computer and System Sciences*, 51(3):374–389, 1995.

[HKRS97] Monika R Henzinger, Philip Klein, Satish Rao, and Sairam Subramanian. Faster shortest-path algorithms for planar graphs. *journal of computer and system sciences*, 55(1):3–23, 1997.

[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*, STOC '16, pages 842–850, New York, NY, USA, 2016. ACM.

[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. Available at http://arxiv.org/abs/1003.2958.

[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. Available at http://arxiv.org/abs/1102.4842.

[KMS13] Philip N Klein, Shay Mozes, and Christian Sommer. Structured recursive separator decompositions for planar graphs in linear time. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 505–514. ACM, 2013.

[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.

[KZ17] Rasmus Kyng and Peng Zhang. Hardness results for structured linear systems. *arXiv preprint arXiv:1705.02944*, 2017.

[LG14] François Le Gall. Powers of tensors and fast matrix multiplication. In *Proceedings of the 39th international symposium on symbolic and algebraic computation*, pages 296–303. ACM, 2014. Available at: https://arxiv.org/abs/1401.7714.

[LGT14] James R Lee, Shayan Oveis Gharan, and Luca Trevisan. Multiway spectral partitioning and higher-order cheeger inequalities. *Journal of the ACM (JACM)*, 61(6):37, 2014.

[LRT79] Richard J Lipton, Donald J Rose, and Robert Endre Tarjan. Generalized nested dissection. *SIAM journal on numerical analysis*, 16(2):346–358, 1979.

[ŁS11] Jakub Łącki and Piotr Sankowski. Min-cuts and shortest cycles in planar graphs in o (n loglogn) time. In *European Symposium on Algorithms*, pages 155–166. Springer, 2011.

[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. ACM, 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.

[MV92] Scott A Mitchell and Stephen A Vavasis. Quality mesh generation in three dimensions. In *Proceedings of the eighth annual symposium on Computational geometry*, pages 212–221. ACM, 1992.

[PRT16] Ori Parzanchevski, Ron Rosenthal, and Ran J Tessler. Isoperimetric inequalities in simplicial complexes. *Combinatorica*, 36(2):195–227, 2016.

[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.

[RTL76] Donald J Rose, R Endre Tarjan, and George S Lueker. Algorithmic aspects of vertex elimination on graphs. *SIAM Journal on computing*, 5(2):266–283, 1976.

[Rup93] Jim Ruppert. A new and simple algorithm for quality 2-dimensional mesh generation. In *Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms*, pages 83–92. Society for Industrial and Applied Mathematics, 1993.

[Saa03] Yousef Saad. *Iterative methods for sparse linear systems*. SIAM, 2003.

[SF73] Gilbert Strang and George J Fix. *An analysis of the finite element method*, volume 212. Prentice-hall Englewood Cliffs, NJ, 1973.

[SKM14] John Steenbergen, Caroline Klivans, and Sayan Mukherjee. A cheeger-type inequality on simplicial complexes. *Advances in Applied Mathematics*, 56:56–77, 2014.

[ST08] Gil Shklarski and Sivan Toledo. Rigidity in finite-element matrices: Sufficient conditions for the rigidity of structures and substructures. *SIAM Journal on Matrix Analysis and Applications*, 30(1):7–40, 2008.

[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.

Appendices

A Schur Complements

In this section, we prove Fact 2.6 on Schur complements. Throughout this subsection, let

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{A}_{SS} & \boldsymbol{A}_{ST} \\ \boldsymbol{A}_{ST}^{\top} & \boldsymbol{A}_{TT} \end{pmatrix}$$

be a symmetric matrix. Let n_1 be the size of S and n_2 be the size of T. Recall that the Schur complement

$$\textsc{Sc}[\boldsymbol{A}]_T = \boldsymbol{A}_{TT} - \boldsymbol{A}_{ST}^\top \boldsymbol{A}_{SS}^\dagger \boldsymbol{A}_{ST}.$$

We restate Fact 2.6 below.

Fact 2.6. *Let $\boldsymbol{A}$ be a symmetric PSD matrix defined as above, and let $\textsc{Sc}[\boldsymbol{A}]_T$ be its Schur complement. Then,*

1. $\textsc{Sc}[\boldsymbol{A}]_T$ *is a symmetric PSD matrix.*
2. $\lambda_{\max}(\textsc{Sc}[\boldsymbol{A}]_T) \leq \lambda_{\max}(\boldsymbol{A})$.

To prove this fact, we need the following fact and a special case of Weyl inequalities.

Fact A.0.1. *For any fixed vector $\boldsymbol{y} \in \mathbb{R}^{n_2}$,*

$$\min_{\boldsymbol{x} \in \mathbb{R}^{n_1}} \begin{pmatrix} \boldsymbol{x}^\top & \boldsymbol{y}^\top \end{pmatrix} \boldsymbol{A} \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{pmatrix} = \boldsymbol{y}^\top \textsc{Sc}[\boldsymbol{A}]_T \boldsymbol{y}.$$

Proof. We expand the left hand side,

$$\begin{pmatrix} \boldsymbol{x}^\top & \boldsymbol{y}^\top \end{pmatrix} \boldsymbol{A} \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{pmatrix} = \boldsymbol{x}^\top \boldsymbol{A}_{SS} \boldsymbol{x} + 2\boldsymbol{x}^\top \boldsymbol{A}_{ST} \boldsymbol{y} + \boldsymbol{y}^\top \boldsymbol{A}_{TT} \boldsymbol{y}. \tag{18}$$

Taking derivative w.r.t. $\boldsymbol{x}$ and setting it to be 0 give that

$$2\boldsymbol{A}_{SS}\boldsymbol{x} + 2\boldsymbol{A}_{ST}\boldsymbol{y} = \boldsymbol{0}.$$

Plugging $\boldsymbol{x} = -\boldsymbol{A}_{SS}^\dagger \boldsymbol{A}_{ST} \boldsymbol{y}$ into (18),

$$\min_{\boldsymbol{x} \in \mathbb{R}^{n_1}} \begin{pmatrix} \boldsymbol{x}^\top & \boldsymbol{y}^\top \end{pmatrix} \boldsymbol{A} \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{pmatrix} = -\boldsymbol{y}^\top \boldsymbol{A}_{ST}^\top \boldsymbol{A}_{SS}^\dagger \boldsymbol{A}_{ST} \boldsymbol{y} + \boldsymbol{y}^\top \boldsymbol{A}_{TT} \boldsymbol{y} = \boldsymbol{y}^\top \textsc{Sc}[\boldsymbol{A}]_T \boldsymbol{y}.$$

This completes the proof. □

Theorem A.0.2 (A special case of Weyl inequalities)**.** *Let $\boldsymbol{H} \in \mathbb{R}^n$ and $\boldsymbol{H} = \boldsymbol{H}_1 + \boldsymbol{H}_2$ where $\boldsymbol{H}_1, \boldsymbol{H}_2$ are symmetric matrices and $\boldsymbol{H}_2$ is a PSD matrix. Let $\lambda_1(\cdot) \geq \ldots \geq \lambda_n(\cdot)$ be eigenvalues of a matrix. Then, $\lambda_i(\boldsymbol{H}) \geq \lambda_i(\boldsymbol{H}_1), i = 1, 2, \ldots, n$.*

Proof of Fact 2.6. The first statement immediately follows Fact A.0.1.

To prove the second statement, we decompose $\boldsymbol{A}$:

$$\begin{aligned} \boldsymbol{A} &= \begin{pmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \textsc{Sc}[\boldsymbol{A}]_T \end{pmatrix} + \begin{pmatrix} \boldsymbol{A}_{SS} & \boldsymbol{A}_{ST} \\ \boldsymbol{A}_{ST}^\top & \boldsymbol{A}_{ST}^\top \boldsymbol{A}_{SS}^{-1} \boldsymbol{A}_{ST} \end{pmatrix} \\ &= \begin{pmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \textsc{Sc}[\boldsymbol{A}]_T \end{pmatrix} + \begin{pmatrix} \boldsymbol{A}_{SS}^{1/2} \\ \boldsymbol{A}_{ST}^\top \boldsymbol{A}_{SS}^{-1/2} \end{pmatrix}^\top \begin{pmatrix} \boldsymbol{A}_{SS}^{1/2} \\ \boldsymbol{A}_{ST}^\top \boldsymbol{A}_{SS}^{-1/2} \end{pmatrix}. \end{aligned}$$

Here we assume that $\boldsymbol{A}_{SS}$ is invertible, otherwise we can use pseudo-inverse. The first matrix is symmetric, and the second matrix is symmetric and PSD. By Theorem A.0.2, $\lambda_{\max}(\boldsymbol{A}) \geq \lambda_{\max}(\textsc{Sc}[\boldsymbol{A}]_T)$. □

B Running Times In Terms of Fast Matrix Multiplication

We now restate the running times of our algorithms in terms of faster matrix multiplication / inversion routines. Specifically, we assume inverting an $n \times n$ matrix takes time $O(n^{\omega})$, where $\omega < 2.3728639$ [LG14].

We first examine purely nested dissection based algorithms. The running time of these algorithms are dominated by the cost of inverting the matrix at the top-most level. Thus, the running time of 3-D nested dissection from [MT90] as given in Theorem 2.7 is

$$O\left(n^{2\omega/3}\right),$$

while the performance of Lemma 2.8 becomes

$$O\left(n^{2\omega\beta/3+\alpha} + n^{\omega\gamma+\alpha}\right).$$

We now propagate these different costs for constructing the nested dissection partial states into our running time analyses.

For the bounded aspect ratio case described in Theorem 3.1, recall that the input truss is a union of k convex pieces of n_i vertices each. Hollowing with parameter

$$r_i = n_i^{c_r}$$

now takes time

$$O\left(\sum_i \frac{n_i}{r_i} \cdot r_i^{2\omega/3}\right) = O\left(n^{1+c_r(2\omega/3-1)}\right),$$

while still giving a Schur complement of size

$$O\left(n^{1+c_r/3}\right)$$

on the boundaries, and total boundary size of

$$O\left(k^{c_r/3} n^{1-c_r/3}\right).$$

Theorem 2.7 then gives that solving this problem on just the boundary elements takes time

$$O\left(k^{2c_r\omega/9} n^{2(1-c_r/3)\omega/3}\right) = O\left(k^{2c_r\omega/9} n^{2\omega/3-2c_r\omega/9}\right),$$

and results in a total fill-in of

$$O\left(k^{4c_r/9} n^{4/3-4c_r/9}\right).$$

Putting these parameters back into the condition number bound of $O(n^{2c_r})$ gives an iteration count of $O(n^{c_r}\log(1/\epsilon))$, which in turn gives a total cost of

$$O\left(n^{1+c_r(2\omega/3-1)}\right) + O\left(k^{2c_r\omega/9} n^{2\omega/3-2c_r\omega/9}\right) + O\left(n^{c_r}\log(1/\epsilon)\right) \cdot \left[O\left(n^{1+c_r/3}\right) + O\left(k^{c_r/3} n^{4/3-4c_r/9}\right)\right]$$
$$= O\left(n^{1+c_r(2\omega/3-1)} + k^{2c_r\omega/9} n^{2\omega/3-2c_r\omega/9} + n^{1+4c_r/3}\log(1/\epsilon) + k^{c_r/3} n^{4/3+5c_r/9}\log(1/\epsilon)\right).$$

We can (slightly) simplify this using the fact that $\omega \le 3$ to drop the first term: it is always upper bounded by the third. Also, since $k \le n$, we will focus on optimizing the exponent on n, that is, we want to pick c_r to minimize the maximum of

$$\begin{aligned}
&2\omega/3 - 2c_r\omega/9\\
&1 + 4c_r/3\\
&4/3 + 5c_r/9
\end{aligned}$$

By running an LP solver, we get:

- when $\omega = 3$ this is optimized at $c_r = 1/2$, which gives a total cost of $O(k^{1/3}n^{5/3}\log(1/\epsilon))$.
- when $\omega = 2.3728639$, this is optimized at $c_r = 0.2295553$. Here the exponents on n the three terms are 1.4608640, 1.3060737 and 1.4608640 respectively, and we have $2\omega/9 > 1/3$, so so the total cost is bounded by $O(k^{0.1210452}n^{1.4608641}\log(1/\epsilon))$.

For the more general case from Theorem 3.2, combining the bounds from Lemma 7.3 with the

- $O(n^{1+c_r(2\omega/3-1)})$ cost of computing the Schur complement of eliminating the innards of the hollowings, and
- the $O(n^{1+c_r/3})$ size of the these Schur complements, and
- $O(n^{c_r}\log(1/\epsilon))$ iteration count of PCG

gives a total cost of:[6]

$$O\Big(n^{1+c_r(2\omega/3-1)} + n^{2\omega/3}l^{-2\omega/3+1} + k^{7\omega/3+c_r\omega/3}n^{2\omega/3-c_r\omega/3}l \\ + n^{c_r}\log(1/\epsilon)\Big(n^{1+c_r/3} + n^{4/3}l^{-1/3} + k^{14/3+2c_r/3}n^{4/3-2c_r/3}l\Big)\Big).$$

Since $\omega \le 3$, we drop the first term. We can simplify this by moving k and $\log(1/\epsilon)$ to the outermost, and only optimizing the remaining terms:

$$O\left(k^{7\omega/3+c_r\omega/3}\log(1/\epsilon)\right)\cdot\Big(n^{2\omega/3}l^{-2\omega/3+1} + n^{2\omega/3-c_r\omega/3}l + \Big(n^{1+4c_r/3} + n^{4/3+c_r}l^{-1/3} + n^{4/3+c_r/3}l\Big)\Big).$$

Let $l = n^{c_l}$. Since Algorithm 2 requires that $c_r + 2c_\alpha \le 1$ and in Lemma 7.3 we set $c_r = c_\alpha$, we have $0 \le c_r \le 1/3$. Subject to this requirement, we minimize the maximum of the following terms:

$$\begin{aligned}
&1 + \left(\frac{2\omega}{3} - 1\right)c_r\\
&\frac{2\omega}{3} + \left(-\frac{2\omega}{3} + 1\right)c_l,\\
&\frac{2\omega}{3} - \frac{\omega}{3}c_r + c_l,\\
&1 + \frac{4}{3}c_r,\\
&\frac{4}{3} + c_r - \frac{1}{3}c_l,\\
&\frac{4}{3} + \frac{1}{3}c_r + c_l.
\end{aligned}$$

By running an LP solver, we get:

[6] We drop the $\log n$ factor here, given $\log n \ll n^c$ for any constant $c > 0$.

- when $\omega = 3$ this is optimized at $c_r = 1/3$ and $c_l = 1/6$, which gives a total cost of $O(k^{22/3} n^{11/6} \log(1/\epsilon))$ $(11/6 \approx 1.8333)$
- when $\omega = 2.3728639$, an optimum solution is $c_r = 0.2210963$ and $c_l = 0.1105482$ for a total cost of $O(k^{5.7115596} n^{1.5175803} \log(1/\epsilon))$.