Grasping Fragile Objects Using A Stress-Minimization Metric

Zherong Pan[1], Xifeng Gao[2], and Dinesh Manocha[3]

***Abstract*— We present a new method to generate optimal grasps for brittle and fragile objects using a novel stress-minimization (SM) metric. Our approach is designed for objects that are composed of homogeneous isotopic materials. Our SM metric measures the maximal resistible external wrenches that would not result in fractures in the target objects. In this paper, we propose methods to compute our new metric. We also use our SM metric to design optimal grasp planning algorithms. Finally, we compare the performance of our metric and conventional grasp metrics, including $Q_1, Q_\infty, Q_{G11}, Q_{MSV}, Q_{VEW}$. Our experiments show that our SM metric takes into account the material characteristics and object shapes to indicate the fragile regions, where prior methods may not work well. We also show that the computational cost of our SM metric is on par with prior methods. Finally, we show that grasp planners guided by our metric can lower the probability of breaking target objects.**

I. Introduction

Grasp quality metrics are scalar functions defined on the set of possible grasps, which is used to compare the quality of different grasps. The performance and properties of (asymptotically) optimal grasp planning algorithms depend heavily on the type of grasp quality metrics used. A summary of these metrics can be found in [24]. The usual requirements for high-quality grasps include force closure, small contact force magnitudes, the preference of normal forces over frictional forces, higher resilience to external wrenches, and force resilience along all directions. For example, the Q_1, Q_∞ metrics [8] take all these factors into consideration, while Q_{VEW} [17] is direction-dependent. The choice of a metric is typically based on the requirements of an application, but may also affect the choice of the resulting grasp planning algorithm. For example, the Q_1, Q_∞ metrics are submodular, which allow fast discrete grasp points selection [25]. The Q_1 metric has an optimizable lower-bound, which allows an optimization-based grasp planning algorithm [7] to jointly search for grasp points and grasp poses.

However, all the metrics considered so far make a common assumption that the target object is rigid and stiff, and will not crack or break due to to the forces exerted by the grasp. As a result, prior methods assume that the target object is a rigid body and all the forces and torques are applied on the center-of-mass, which greatly simplifies the computation and analysis of metrics. However, this assumption does not hold when a robot is grasping fragile or brittle objects, and certain weak locations on the objects should be avoided to prevent fractures. Grasping of fragile objects has been considered in prior works [22], [1], which attempt to avoid breaking objects by developing safe grippers. Our goal is complimentary, as we want to formulate new grasp metrics that reduce the stresses in fragile objects.

Main Results: We present a novel stress-minimization metric that takes into account the probability of the target object being broken into pieces. Based on the theory of brittle fracture [10], we formulate the set of external wrenches that can be resisted without causing fractures in the object. Similar to the Q_1 metric [8], our metric measures the maximal size of the sphere-shaped subset of resistible wrenches. We refer to this new metric as the stress-minimization (SM) metric Q_{SM}. We show that Q_{SM} can be computed efficiently using the boundary element method (BEM) [6], given a surface triangle mesh of the object and a set of contact points. The cost of computing Q_{SM} is similar to that of computing Q_1. Using a simple scenario, we also show that most previous grasp metrics, including Q_1, Q_∞ [8], Q_{G11} [4], Q_{VEW} [17], and Q_{MSV} [4], do not reflect the probability of the target object being broken, while Q_{SM} does. Finally, we show that conventional optimization-based grasp planning algorithms [5], [25], [14] can be modified to search for globally optimal grasps by maximizing Q_{SM}. Based on our algorithm, we can compute optimal grasps for a row of complex, high-genus objects with the lowest probability of breaking them. Computing Q_{SM} takes $4-8$s and finding optimal grasps under Q_{SM} takes $1-5$hr on average using a global optimization algorithm [14] or less than 10min using a stochastic optimization algorithm [27].

II. Related Work

We review related work in grasp quality metrics, material and fracture modeling, and grasp planning.

Grasp Quality Metrics: Although some grasp planners only consider external wrenches along some certain directions [21], more pertinent characterization of a robust grasp requires resilience to external wrenches along all directions, which is known as force closure [23]. However, infinitely many grasps can have force closure and some of them are classified as good grasps based on different quality metrics [24]. Most of the metrics Q are designed such that $Q > 0$ implies force closure. Closely related metrics to our Q_{SM} are Q_1, Q_∞ [8], which measures the maximal radius of an origin-centered wrench-space sphere contained in the convex set of resistible wrenches under contact force magnitude constraints.

Material and Fracture Modeling: Real world solid objects may result in brittle or ductile fractures under external forces, depending on the underlying materials. However, modeling the physics during fractures is difficult because they can undergo large deformations during fractures [10]. Fortunately, for grasp planning, we do not need to model these deformations precisely, but only need to detect where fractures might occur. In this case, the theory of linear elasticity suffices [10] and the stress induced by external wrenches

[1]Zherong is with Department of Computer Science, University of North Carolina at Chapel Hill. {zherong@cs.unc.edu} [2]Xifeng Gao is with Department of Computer Science, Florida State University. {gao@cs.fsu.edu} [3]Dinesh Manocha is with Department of Computer Science and Electrical & Computer Engineering, University of Maryland at College Park. {dm@cs.umd.edu}

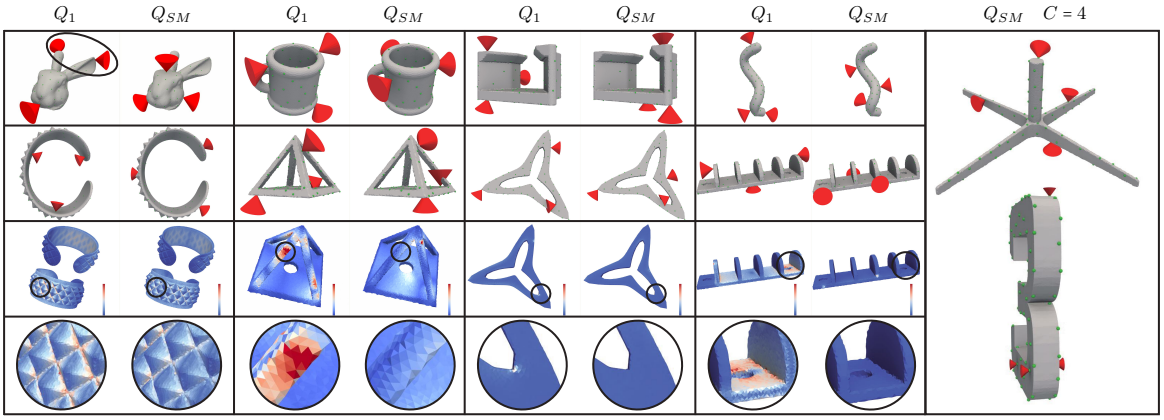

Fig. 1: First and second rows: We show globally optimal grasps for 8 different target objects based on Q_1 (left) and Q_{SM} (right) in each box. These grasps are generated by choosing $C = 3$ contact points (frictional cones in red) from $N = 100$ potential contact points using BB, which takes 1.7hr to compute. The optimal grasp for the bunny head avoids the ears of the bunny under the Q_{SM} metric (black circle). Third and fourth rows: We plot the maximal stress configuration (color coded). Using Q_{SM} can drastically reduce the maximal stress as indicated in the black circle. Finally, we show two optimal grasps with $C = 4$ and Q_{SM} on the right, which takes 5.4hr to compute.

on target objects can be computed efficient using the finite element method (FEM) [16] or the boundary element method (BEM) [6]. We use BEM as our computational tool because it can take the materials of the target object into consideration while only requires a surface triangle mesh, which is more amenable to robot grasp applications.

Grasp Planning: Given a grasp quality metric, an (asymptotically) optimal grasp planning algorithm finds a grasp that maximizes the grasp quality metric. Early algorithms [27] use sampling-based methods for planning. These algorithms are very general and they are agnostic to grasp quality metric types. However, more efficient algorithms such as [7], [25] can be designed if quality metrics have certain properties such as monotonicity and submodularity. A good grasp metric can also be used in learning based grasp planners such as [19], [18], [20] that use precomputed metrics to train a grasp quality function represented by a deep neural network and then use the function as a guidance.

III. Problem Statement

In this section, we formulate the problem of stress-minimization grasp planning. Throughout the paper, we assume that the 3D target object is in its reference space, where the origin coincides with its center-of-mass. The object takes up a volume that is a closed subset $\Omega \subset R^3$. In addition, the object is under an external 6D wrench $\mathbf{w}$ and a set of N external contact forces $\mathbf{f}_{1,\cdots,N}$ at contact points $\mathbf{x}_{1,\cdots,N}$ with unit contact normals $\mathbf{n}_{1,\cdots,N}$. If a grasp is valid, we have the following wrench balance condition:

$$\mathbf{w} = -\sum_{i=1}^{N} \begin{pmatrix} \mathbf{f}_i \\ \mathbf{x}_i \times \mathbf{f}_i \end{pmatrix} \quad \mathbf{s.t.}\ \|(\mathbf{I} - \mathbf{n}_i\mathbf{n}_i^T)\mathbf{f}_i\| \le \theta \mathbf{n}_i^T \mathbf{f}_i, \qquad (1)$$

where θ is the frictional coefficient. A well-known method for comparing the quality of two different grasps is to compare their Q_1 metric [8], which is the maximal radius of the origin-centered inscribed sphere in the convex hull of all possible resistible external wrenches when the magnitude of $\mathbf{f}_i$ is bounded. Mathematically, this can be expressed as:

$$Q_1 = \mathbf{max}\ r \quad \mathbf{s.t.}\{\mathbf{w} | \mathbf{w}^T \mathbb{W} \mathbf{w} \le r^2\} \subseteq$$
$$\{\mathbf{w} | \exists \mathbf{f}_{1,\cdots,N},\ \mathbf{s.t.} Equation\ 1, \sum_{i=1}^{N} \mathbf{n}_i^T \mathbf{f}_i \le 1\},$$

where $\mathbb{W}$ is the 6×6 positive semi-definite metric tensor in the wrench space.

However, the formulation of Q_1 metric assumes that the object remains rigid and will never be broken or decomposed into parts, no matter how large the external forces are. To relax this condition, we have to make use of the numerical models of brittle fracture, e.g., [10]. In these formulations, we assume that the object is made of homogeneous isotropic elastic material with λ, μ being its Lamé material parameters [10]. This material model covers most target objects encountered in our daily lives, including objects made of copper, rubber, glass, and porcelain. When under external force fields $\mathbf{g}(\mathbf{x}) : R^3 \to R^3$, an infinitesimal displacement $\mathbf{u}(\mathbf{x}) : R^3 \to R^3$ and a stress field $\boldsymbol{\sigma}(\mathbf{x}) : R^3 \to R^{3\times3}$ will occur $\forall \mathbf{x} \in \Omega$. $\mathbf{u}(\mathbf{x}), \boldsymbol{\sigma}(\mathbf{x})$ can be computed from $\mathbf{g}(\mathbf{x})$ using the force balance condition:

$$\begin{aligned} \forall \mathbf{x} \in \Omega: \quad & \boldsymbol{\epsilon} \triangleq (\nabla \mathbf{u} + \nabla \mathbf{u}^T)/2 \\ & \boldsymbol{\sigma} \triangleq 2\mu\boldsymbol{\epsilon} + \lambda \mathbf{tr}(\boldsymbol{\epsilon})\mathbf{I} \\ & \nabla \cdot \boldsymbol{\sigma} + \mathbf{g} = 0 \qquad (2) \\ \forall \mathbf{x} \in \partial\Omega: \quad & \mathbf{n}(\mathbf{x}) \cdot \boldsymbol{\sigma} + \sum_{i=1}^{N} \delta(\mathbf{x} - \mathbf{x}_i)\mathbf{f}_i = 0, \end{aligned}$$

where we assume the boundary of Ω is smooth almost everywhere with the unit outward normal defined as $\mathbf{n}(\mathbf{x})$ and δ is the Dirac's delta operator. Classical theory of brittle fracture [10] further assumes that there exists a tensile stress $\boldsymbol{\sigma}_{max}$ and brittle fractures will not happen if the following condition holds:

$$\forall \|\mathbf{d}\| = 1, \mathbf{x} \in \Omega: \ \mathbf{d}^T \boldsymbol{\sigma}(\mathbf{x}) \mathbf{d} \le \boldsymbol{\sigma}_{max}.$$

In order to make sure that the grasp metric always takes bounded value, we propose to slightly modify the above condition and limit $\boldsymbol{\sigma}(\mathbf{x})$ on both sides, leading to the following condition:

$$\forall \|\mathbf{d}\| = 1, \mathbf{x} \in \Omega: \ -\boldsymbol{\sigma}_{max} \le \mathbf{d}^T \boldsymbol{\sigma}(\mathbf{x}) \mathbf{d} \le \boldsymbol{\sigma}_{max}. \qquad (3)$$

Note that the stress tensor must be symmetric so that its singular values coincide with its eigenvalues. Given the theory of brittle fracture, the goal of our work is to propose a grasp metric Q_{SM} that measures grasp qualities with Equation 3 as a precondition, present algorithms to compute

Q_{SM}, and analyze grasp planning algorithms that are based on our Q_{SM} metric.

IV. The Stress-Minimization Metric Q_{SM}

Our construction of Q_{SM} is illustrated in Figure 2. The basic idea behind the construction of Q_{SM} is very similar to that of the Q_1 metric. Intuitively, we first define a convex subset $\mathbf{W}$ of the 6D wrench space, which contains resistible wrenches that does not violate Equation 3. We then define Q_{SM} as the maximal radius of the origin-centered sphere contained in $\mathbf{W}$.

A. Definition of $\mathbf{W}$ and Q_{SM}

Given a certain wrench $\mathbf{w}$, we need to determine whether $\mathbf{w} \in \mathbf{W}$. This can be performed by first computing $\boldsymbol{\sigma}$ and then testing whether Equation 3 holds. However, Equation 2 is a relationship between $\boldsymbol{\sigma}$ and $\mathbf{g}(\mathbf{x})$ but not $\mathbf{w}$, so we need to find a relationship between $\mathbf{g}$ and $\mathbf{w}$. In other words, we need to find a body force distribution such that the net effect of $\mathbf{g}$ is equivalent to applying $\mathbf{w}$ on the center-of-mass. Obviously, infinitely many formulations of $\mathbf{g}$s will satisfy this relationship and different choices of $\mathbf{g}(\mathbf{x})$ will lead to different variants of Q_{SM} metrics. In this paper, we propose choosing $\mathbf{g}$ as a linear function in $\mathbf{x}$. The most important reason behind this choice is that the computation of $\boldsymbol{\sigma}$ can be performed using BEM [6] if $\mathbf{g}$ is a harmonic function of $\mathbf{x}$, and BEM can be applied to a surface mesh representing the target object while FEM [16] requires a volume mesh. Under this choice, we have: $\mathbf{g}(\mathbf{x}) = \mathbf{g}_0 + \nabla\mathbf{g}\mathbf{x}$, where $\mathbf{g}_0$ is the constant term, and $\nabla\mathbf{g}$ is the constant spatial derivative tensor. Clearly $\mathbf{g}(\mathbf{x})$ has 12 degrees of freedom (3 in $\mathbf{g}_0$ and 9 in $\nabla\mathbf{g}$) and we can solve for $\mathbf{g}_0$ and $\nabla\mathbf{g}$ to equate the effect of $\mathbf{g}$ and $\mathbf{w}$ as follows:

$$\mathbf{w} = \int_\Omega \begin{pmatrix} \mathbf{g} \\ \mathbf{x} \times \mathbf{g} \end{pmatrix} d\mathbf{x} = \begin{pmatrix} |\Omega|\mathbf{g}_0 \\ \mathcal{T} \begin{pmatrix} [\nabla\mathbf{g}]_x \\ [\nabla\mathbf{g}]_y \\ [\nabla\mathbf{g}]_z \end{pmatrix} \end{pmatrix},$$

where $\mathcal{T} \triangleq \int_\Omega \left(x\mathbf{x}\times, y\mathbf{x}\times, z\mathbf{x}\times \right) d\mathbf{x}$. Also, $[\nabla\mathbf{g}]_{x,y,z}$ are the first, second, and third column of $\nabla\mathbf{g}$, respectively. However, there are 9 degrees of freedom in $\nabla\mathbf{g}$ but only 3 constraints so we have to solve for $\nabla\mathbf{g}$ in a least square sense:

$$\nabla\mathbf{g} = \underset{\nabla\mathbf{g}}{\mathbf{argmin}} \int_\Omega \|\nabla\mathbf{g}\mathbf{x}\|^2 d\mathbf{x} \quad \mathbf{s.t.}\ \mathcal{T} \begin{pmatrix} [\nabla\mathbf{g}]_x \\ [\nabla\mathbf{g}]_y \\ [\nabla\mathbf{g}]_z \end{pmatrix} = \begin{pmatrix} \mathbf{w}_4 \\ \mathbf{w}_5 \\ \mathbf{w}_6 \end{pmatrix},$$

the solution of which can be computed analytically. In summary, we have:

$$\begin{pmatrix} \mathbf{g}_0 \\ [\nabla\mathbf{g}]_x \\ [\nabla\mathbf{g}]_y \\ [\nabla\mathbf{g}]_z \end{pmatrix} = \begin{pmatrix} \frac{\mathbf{I}}{|\Omega|} & \\ & \mathcal{M}^{-1}\mathcal{T}^T \left[\mathcal{T}\mathcal{M}^{-1}\mathcal{T}^T \right]^{-1} \end{pmatrix} \mathbf{w}, \qquad (4)$$

where $\mathcal{M} \triangleq \left[\int_\Omega \mathbf{x}\mathbf{x}^T d\mathbf{x} \right] \otimes \mathbf{I}$ and $\otimes$ denotes Kronecker product. The matrices $\mathcal{T}, \mathcal{M}$ are constants and can be precomputed from the shape of the target object or, more specifically, from the inertia tensor. Given these definitions, we can now define $\mathbf{W}$ as follows:

$$\mathbf{W} = \{\mathbf{w} | \exists \mathbf{g}, \mathbf{f}_{1,\cdots,N}, \mathbf{u}, \boldsymbol{\epsilon}, \boldsymbol{\sigma},\ \mathbf{s.t.}\ Equation\ 1,2,3,4\}. \qquad (5)$$

Finally, we are ready to give a mathematical definition of Q_{SM} using the following optimization function:

$$Q_{SM} = \mathbf{max}\ r \quad \mathbf{s.t.}\ \{\mathbf{w} | \mathbf{w}^T \mathbb{W} \mathbf{w} \le r^2\} \subseteq \mathbf{W}.$$

From the mathematical definition of Q_{SM}, we immediately have the following properties of $\mathbf{W}$:

Lemma 4.1: $\mathbf{W}$ is a convex set.

Proof: Equation 1 is a set of quadratic cone constraints, which defines a convex set. Equation 2 is a set of infinite-dimensional linear constraints, which defines a convex set. Equation 3 is an infinite-dimensional PSD-cone constraint, which defines a convex set. Finally, Equation 4 is a linear constraint, which also defines a convex set. As the intersection of convex sets, $\mathbf{W}$ is convex. ■

Lemma 4.2: $\mathbf{W}$ is a compact set so that Q_{SM} is finite.

Proof: For any $\mathbf{w} \neq 0$, $\boldsymbol{\sigma}$ that satisfies all three conditions (Equation 1, Equation 2 and Equation 4) cannot be zero for all $\mathbf{x} \in \Omega$, otherwise the last equation in Equation 2 will be violated. In other words, there exist $\mathbf{d}$ and $\mathbf{x}$ such that $|\mathbf{d}^T \boldsymbol{\sigma}(\mathbf{x})\mathbf{d}| > \epsilon > 0$. If we multiply $\mathbf{w}$ by $\alpha > \boldsymbol{\sigma}_{max}/\epsilon$, Equation 3 will be violated so that $\alpha\mathbf{w} \notin \mathbf{W}$. Therefore, $\mathbf{W}$ is bounded and is obviously closed, so that $\mathbf{W}$ is compact and Q_{SM} is finite. ■

In addition, the following property of Q_{SM} is obvious:

Lemma 4.3: $Q_{SM} > 0$ implies force closure.

The following property has been proved in [25] for Q_1 and also holds for Q_{SM} by a similar argument:

Lemma 4.4: $Q_{SM} = \min\limits_{\mathbf{d}, \|\mathbf{d}\|=1} \max\limits_{\mathbf{w} \in \sqrt{\mathbb{W}}\mathbf{W}} \mathbf{w}^T \mathbf{d}$.

B. Discretization of Q_{SM}

The computation of an exact Q_{SM} is impractical because it involves infinite dimensional tensor fields: $\boldsymbol{\sigma}, \boldsymbol{\epsilon}$, so we have to discretize them using conventional techniques such as FEM [16] or BEM [6]. We provide the detailed derivation of BEM in Appendix and summarize the main results here. Our BEM algorithm approximates the stress field $\boldsymbol{\sigma}(\mathbf{x})$ to be piecewise constant on each triangular patch of the surface. Assuming that the target object has K surface triangles whose centroids are: $\mathbf{x}_{1,\cdots,K}$, we have K different stress values:

$$\begin{pmatrix} \boldsymbol{\sigma}_x(\mathbf{x}_j) \\ \boldsymbol{\sigma}_y(\mathbf{x}_j) \\ \boldsymbol{\sigma}_z(\mathbf{x}_j) \end{pmatrix} = \mathcal{A}_j \begin{pmatrix} \mathbf{g}_0 \\ [\nabla\mathbf{g}]_x \\ [\nabla\mathbf{g}]_y \\ [\nabla\mathbf{g}]_z \end{pmatrix} + \mathcal{B}_j \begin{pmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_N \end{pmatrix} \quad \forall j = 1, \cdots, K, \quad (6)$$

where $\mathcal{A}, \mathcal{B}$ are dense coefficient matrices defined from BEM discretization and the detailed formulations are given in Appendix . Note that computing the coefficients of these two matrices is computationally costly, for which a naive implementation of BEM requires $\mathcal{O}(K^3)$ operations and acceleration techniques such as the H-matrix [13] can reduce this cost to $\mathcal{O}(K\mathbf{log}^2 K)$ operations. However, these two matrices are constant and can be precomputed for a given target object shape, so the cost of BEM computation is not a part of grasp planning. After discretization, we arrive at the finite-dimensional version of the fracture condition:

$$\forall \|\mathbf{d}\| = 1, j = 1, \cdots, K: \ -\boldsymbol{\sigma}_{max} \le \mathbf{d}^T \boldsymbol{\sigma}(\mathbf{x}_j)\mathbf{d} \le \boldsymbol{\sigma}_{max}. \quad (7)$$

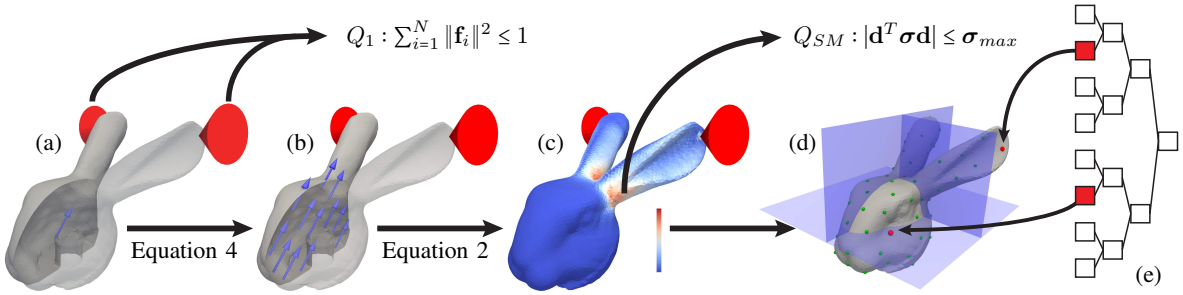

Fig. 2: A toy example where we grasp a bunny head represented as a triangular surface mesh, with grasp points on two ears (frictional cones in red). (a): When the bunny head is under an external wrench (blue arrow), the Q_1 metric assumes that the wrench is applied on the center-of-mass. (b): Our Q_{SM} metric assumes that the external wrench is applied as a force field all over the volume of bunny head (blue arrows). (c): We use BEM to solve for a surface stress field (color coded on surface with high stress in red around connections between the ears and the head). Q_{SM} assumes that the stress along any direction $\mathbf{d}$ is smaller than the tensile stress, $\boldsymbol{\sigma}_{max}$. (d): When performing grasp planning, we first construct a KD-tree (transparent blue planes) for the set of N potential contact points. (e): We select C contact points (red points) by descending the tree.

The finite-dimensional version of $\bar{\mathbf{W}}$:

$$\bar{\mathbf{W}} \triangleq \{\mathbf{w} | \exists \mathbf{g}, \mathbf{f}_{1,\cdots,N}, \mathbf{u}, \boldsymbol{\epsilon}, \boldsymbol{\sigma}, \ \mathbf{s.t.}\ \textit{Equation}\ 1,6,7,4\},$$

and the finite-dimensional version of $\bar{Q}_{SM}$ defined as:

$$\bar{Q}_{SM} = \mathbf{argmax}\ r \quad \mathbf{s.t.}\ \{\mathbf{w} | \mathbf{w}^T \mathbb{W} \mathbf{w} \leq r^2\} \subseteq \bar{\mathbf{W}}.$$

All the properties of the infinite-dimensional $\mathbf{W}$ and Q_{SM} hold for the finite-dimensional version $\bar{\mathbf{W}}$ and $\bar{Q}_{SM}$ by a similar argument.

C. Computation of $\bar{Q}_{SM}$

Computing $\bar{Q}_{SM}$ amounts to a non-trivial global optimization. According to Lemma 4.4, the equivalent optimization problem for $\bar{Q}_{SM}$ is:

$$\bar{Q}_{SM} = \min_{\mathbf{d}, \|\mathbf{d}\|=1} \max_{\mathbf{w} \in \sqrt{\mathbb{W}}\bar{\mathbf{W}}} \mathbf{w}^T \mathbf{d},$$

for which direct optimization does not guarantee a global solution. In this section, we modify two existing algorithms to (approximately) compute $\bar{Q}_{SM}$, which were originally proposed to compute Q_1. In [25], the space of the unit vectors is discretized into a finite set of D directions: $\mathbf{d}_{1,\cdots,D}$. As a result, we can compute an upper bound for $\bar{Q}_{SM}$ as:

$$\bar{Q}_{SM} \leq \min_{j=1,\cdots,D} \max_{\mathbf{w} \in \sqrt{\mathbb{W}}\bar{\mathbf{W}}} \mathbf{w}^T \mathbf{d}_j.$$

We can make this upper bound arbitrarily tight by increasing D. In another algorithm [28], a convex polytope $\mathcal{C} \subseteq \bar{\mathbf{W}}$ is maintained using H-representation [12] and we can compute a lower bound for $\bar{Q}_{SM}$ as:

$$\bar{Q}_{SM} \geq \min_{\mathbf{d}, \|\mathbf{d}\|=1} \max_{\mathbf{w} \in \sqrt{\mathbb{W}}\mathcal{C}} \mathbf{w}^T \mathbf{d}. \tag{8}$$

The global optimum of the optimization function in Problem 8 is easy to compute from an H-representation of $\mathcal{C}$ by computing the distance between the origin and each face of $\mathcal{C}$. This lower bound can be iteratively tightened by first computing the blocking face normal $\mathbf{d}$ of $\mathcal{C}$ and then expanding $\mathcal{C}$ via:

$$\mathcal{C} \leftarrow \mathbf{ConvexHull}(\mathcal{C} \cup \{\mathbf{argmax}_{\mathbf{w} \in \sqrt{\mathbb{W}}\bar{\mathbf{W}}} \mathbf{w}^T \mathbf{d}\}).$$

These two algorithms can be implemented if we can find the supporting point of $\sqrt{\mathbb{W}}\bar{\mathbf{W}}$, which amounts to the following conic programming problem:

$$\begin{aligned} &\mathbf{argmax}_{\mathbf{w}, \mathbf{f}_i, \boldsymbol{\sigma}(\mathbf{x}_j)}\ \mathbf{w}^T \sqrt{\mathbb{W}} \mathbf{d} \\ &\quad \mathbf{s.t.}\ \textit{Equation}\ 1,6,4 \\ &\quad -\boldsymbol{\sigma}_{max}\mathbf{I} \preceq \boldsymbol{\sigma}(\mathbf{x}_j) \preceq \boldsymbol{\sigma}_{max}\mathbf{I} \quad \forall j = 1, \cdots, K. \end{aligned} \tag{9}$$

The conic programming reformulation in Problem 9 can be solved using an interior point method [3]. Given this solution procedure, we summarize the modified version of [25] in Algorithm 1 and the modified version of [28] in Algorithm 2. Note that Algorithm 2 is computationally costlier but it can approximate $\bar{Q}_{SM}$ up to an arbitrary precision ϵ, so we always use Algorithm 2 in the rest of the paper.

Compared with the Q_1 metric, a major limitation of using the $\bar{Q}_{SM}$ metric is that the computational cost is much higher. Note that the computational cost of solving Problem 9 is at least linear in K and can be superlinear depending on the type of conic programming solver used, such as [3]. This K is the number of surface triangles on the target object, which can be several thousands for complex objects. Fortunately, we can drastically reduce this cost by progressively adding constraints.

D. Performance Optimization

The naive execution of Algorithm 2 can be prohibitively slow due to the repeatedly solving of Problem 9. The conic programming problem has K PSD-cone constraints with K being several thousands. Solving Problem 9 using interior point method [3] involves repeatedly solving a sparse linear system with a size proportional to K. We propose a method that can greatly improve the performance when solving Problem 9. Our idea is that when the global optimum of Problem 9 is reached, most PSD-cone constraints are inactive, so removing these constraints does not alter the solution. This idea is inspired by [29] which shows that, empirically, maximal stress only happens on a few sparse points on the surface of the target object. However, we do not know the active constraints as a prior. Therefore, we propose to progressively detect these active constraints.

In order to perform these computations, we first select a subset $\mathbb{K} \subset \{1, \cdots, K\}$ such that $|\mathbb{K}| \ll K$ and $\{\boldsymbol{\sigma}(\mathbf{x}_i) | i \in \mathbb{K}\}$ are the stress constraints that are most likely to be violated. In other words, $\mathbb{K}$ is the initial guess of the active constraints. To select this set $\mathbb{K}$, we use a precomputation step and solve Problem 9 for 1000 times using random $\mathbf{d}$ and record which PSD-cones are active. For each PSD-cone, we maintain how many times they become active during the 1000 solves of Problem 9. We then select the most frequent $|\mathbb{K}|$ PSD-cones to form $\mathbb{K}$. After selecting $\mathbb{K}$, we maintain an active set $\mathcal{S}$ that initializes to $\mathbb{K}$ and we solve Problem 9 using constraints

only in $\mathcal{S}$, which is denoted by:

$$\begin{aligned} &\underset{\mathbf{w},\mathbf{f}_i,\boldsymbol{\sigma}(\mathbf{x}_j)}{\mathbf{argmax}}\ \mathbf{w}^T\sqrt{\mathbb{W}}\mathbf{d} \\ &\quad \mathbf{s.t.}\ \textit{Equation}\ 1,6,4 \\ &\quad -\boldsymbol{\sigma}_{max}\mathbf{I} \preceq \boldsymbol{\sigma}(\mathbf{x}_j) \preceq \boldsymbol{\sigma}_{max}\mathbf{I} \quad \forall j \in \mathcal{S}. \end{aligned} \tag{10}$$

Problem 10 is convex and we can solve for its global optimum, after which we check the stresses on the remaining constraints and we pick the most violated constraint:

$$j^* = \underset{j\in\{1,\cdots,K\}/\mathcal{S}}{\mathbf{argmax}} \sqrt{\|\boldsymbol{\sigma}(\mathbf{x}_j)\boldsymbol{\sigma}(\mathbf{x}_j)\|_2}. \tag{11}$$

If we have $\sqrt{\|\boldsymbol{\sigma}(\mathbf{x}_{j^*})\boldsymbol{\sigma}(\mathbf{x}_{j^*})\|_2} < \boldsymbol{\sigma}_{max}$, then the global optimum of Problem 10 and Problem 9 will coincide. Otherwise, we add j^* to $\mathcal{S}$. This method is summarized in Algorithm 3 and is guaranteed to return the same global optimum of Problem 9. The complexity of Algorithm 3 is hard to analyze, but in practice it is orders of magnitude more efficient than considering all constraints at once.

V. Grasp Planning Under The SM Metric

By grasp planning, our goal is to select C points that tend to maximize $\bar{Q}_{SM}$ given a set of N potential grasp points sampled on $\partial\Omega$. There are two types of algorithms that can be used for grasp planning, which are based on branch-and-bound (BB) [5], [25], [14] and sub-modular coverage (SMC) [25]. However, SMC requires the formulation in Problem 9 to be a sub-modular function in the set of contact points. Although this property holds for Q_1, Q_∞, whether this property holds for Q_{SM} is still an open problem. However, we can use BB that only requires Problem 9 to be a monotonic function in the set of contact points, which is obvious. We follow [14] and build a KD-tree for the set of N potential grasp points, as illustrated in Figure 2d and Figure 2e. We maintain a pointer to one KD-tree node for each of the C selected points. Next, we descend the tree until all the C pointers reach leaf nodes and keep track of the best set of C leaf nodes. BB can find the globally optimal set of C grasp points that maximize any monotonic grasp metric (see [14] for more details).

Algorithm 1 A modified algorithm of [25] to compute $\bar{Q}_{SM}$

```
sample directions d_{1,...,D} in SO(3)
for i = 1, ..., D do
    ▷ Using Algorithm 3
    Solve Problem 9 with d ← d_i for w_i
Return min_i {w_i^T √W d_i}
```

VI. Evaluations

We have implemented our algorithms for computing Q_{SM} and perform grasp planning using C++. The accuracy of BEM relies on the quality of the surface triangle mesh, so we first optimize the mesh quality to maximize the minimal internal angles of each surface mesh triangle using CGAL [2]. We implement the BEM using a kernel independent numerical integration scheme [9]. The most computationally costly step in BEM is the inversion of system matrices, for which we use LU-factorization accelerated by H-matrices [13]. Finally, we use CGAL [15] to construct convex hulls with exact arithmetic [11] to avoid degenerate cases. All the experiments are performed on a single desktop machine with two Xeon E5-2697 CPUs and 256Gb memory. In the rest of this section, we evaluate the properties of Q_{SM} and compare it with other metrics, including Q_1, Q_∞ [8], Q_{G11}, Q_{MSV} [4], and Q_{VEW} [17].

Algorithm 2 A modified algorithm of [28] to compute $\bar{Q}_{SM}$

```
sample initial directions d_{1,...,D} in SO(3)
for i = 1, ..., D do
    ▷ Using Algorithm 3
    Solve Problem 9 with d ← d_i for w_i.
C_0 ← ConvexHull(w_{1,...,D})
Solve Problem 8 with C ← C_0 for Q̄^0_SM
Store the blocking face normal on C_0 as d_0
while k = 1, ... do
    ▷ Using Algorithm 3
    Solve Problem 9 with d ← d_{k-1} for w_k
    C_k ← ConvexHull(C_{k-1} ∪ {w_k})
    Solve Problem 8 with C ← C_k for Q̄^k_SM
    Store the blocking face normal on C_k as d_k
    if |Q̄^k_SM − Q̄^{k-1}_SM| < ε then
        return Q̄^k_SM
```

Algorithm 3 Progressive solve of Problem 9

```
S ← K
while S ≠ {1, ..., K} do
    Solve Problem 10 for w, f_i, σ(x_j)
    Pick j* using Equation 11
    if √(‖σ(x_{j*})σ(x_{j*})‖_2) < 1 then
        Return w, f_i, σ(x_j)
    else
        S ← S ∪ {j*}
Return w, f_i, σ(x_j)
```

Parameter Choices: Computing Q_{SM} requires more parameters than are necessary for computing Q_1. Specifically, there are three additional variables: tensile stress $\boldsymbol{\sigma}_{max}$ and Lamé material parameters: μ, λ. However, if we transform μ, λ to an equivalent set of parameters: Young's modulus E and Poisson ratio ν [16], it is obvious that Q_{SM} is proportional to $\boldsymbol{\sigma}_{max}$ and inversely proportional to E. Since the absolute value of a grasp metric is meaningless for grasp planning and only the relative value matters, we can always set $\boldsymbol{\sigma}_{max} = E = 1$ and choose only ν according to the material type of the target object, and then set:

$$\mu = 1/2(1+\nu) \quad \lambda = \nu/\left[(1+\nu)(1-2\nu)\right].$$

In our experiments, we assume that objects are made of copper with $\nu = 0.33$. Finally, when running Algorithm 2, we set $\epsilon = 0.001$.

Shape-Awareness: The most remarkable advantage of Q_{SM} over previous metrics, such as Q_1, Q_∞, Q_{MSV}, Q_{VEW}, and Q_{G11}, is shape awareness. In Figure 3a, our

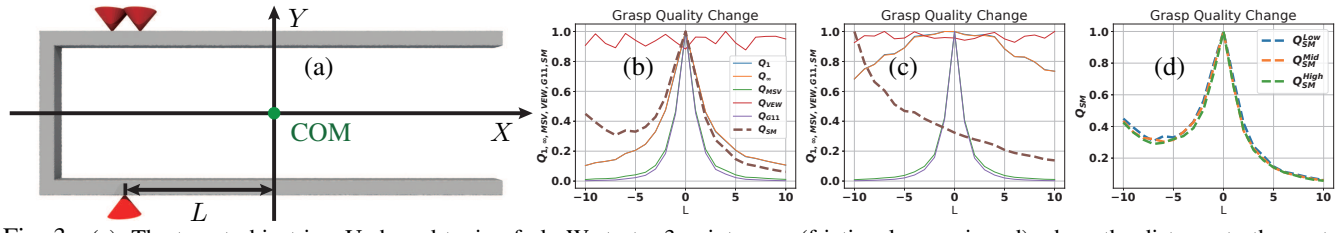

Fig. 3: (a): The target object is a U-shaped tuning fork. We test a 3-point grasp (frictional cones in red) where the distance to the center of mass (green) is L. (b): We plot the change of $Q_1, Q_\infty, Q_{MSV}, Q_{VEW}, Q_{G11}$, and Q_{SM} against L under 4 different conditions, where Q_{VEW} is not stable and $Q_1, Q_\infty, Q_{MSV}, Q_{G11}$ do not reflect asymmetric. Here, we apply equal weights to the forces and torques with $\mathbb{W}_{bd} = \mathbf{Id}$. (c): We use lower weights for torques with $\mathbb{W}_c = \mathbf{diag}(1, 1, 1, 0.01, 0.01, 0.01)$, in which case the best grasp under Q_{SM} is different from that under the four other metrics. (d): We plot Q_{SM} using $\mathbb{W}_{bd}$ and meshes of different resolutions (Low: $K = 5730$, Mid: $K = 24204$, High: $K = 94398$).

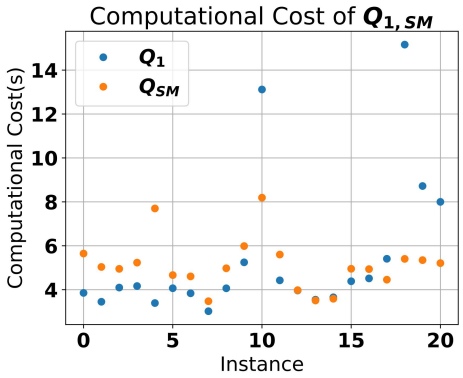

Fig. 4: We compare the computational cost of computing Q_1 and Q_{SM} for 20 random target objects and grasps.

target object is a U-shaped tuning fork and we use a 3-point grasp. The shape of the tuning fork is asymmetric along the X-axis. According to Figure 3b, the best grasp under all metrics are the same, i.e., grasping the centroid point. However, previous metrics are not aware of the asymmetry, while Q_{SM} correctly reflects the fact that grasping the leftmost point is better than grasping the rightmost point because it is less likely to break the object. If we change the metric $\mathbb{W}$ and emphasize force resistance over torque resistance, then the difference between Q_{SM} and previous metrics is more visible and even the best grasp changes as shown in Figure 3c.

Robustness to Mesh Resolution: The change of Q_{SM} is not sensitive to the resolution of the surface meshes, as shown in Figure 3d, which makes Q_{SM} robust to target objects discretized using small, low-resolution meshes. As we increase K from 5730 to 24204 and finally to 94398, the change of Q_{SM} against L is almost intact, with very small fluctuations around $L = -5$.

Computational Cost: Q_{SM} has higher computational cost than Q_1. Most of the computational overhead lies in the assembly of matrices $\mathcal{A}, \mathcal{B}$, which involves the direct factorization of a large, dense matrix. However, this assembly is precomputation and is required only once for each target object before grasp planning. In Figure 3d, this step takes 112s when $K = 5730$, 1425s when $K = 24204$, and 3892s when $K = 94398$. After precomputation, the costs of evaluating Q_{SM} and Q_1 are very similar, as shown in Figure 4. This implies that using Q_{SM} does not incur a higher cost in grasp planning. This is largely due to the progressive Algorithm 3, which greatly reduces the number of constraints in solving Problem 9. Without this method, solving Problem 9 is prohibitively costly as it requires the solve of a sparse linear system of a size proportional to K.

Grasp Planning: In Figure 1, we show globally optimal grasps for 8 different target objects under both the Q_{SM} and Q_1 metric. To generate these results, we choose $C = 3$ contact points from $N = 100$ potential contact points using branch-and-bound. These contact points are generated using Poisson disk sampling. The computational cost of BB, on average, is 1.7hr under Q_{SM} and 0.6hr under Q_1. When we choose $C = 4$, the average computational cost under Q_{SM} increases to 5.4hr. Although the cost of computing Q_{SM} is comparable to that of computing Q_1, we found that Q_{SM} tends to create more local minima so BB needs to create a larger search tree under Q_{SM}. In the third row of Figure 1, we show the maximal stress configuration in $\mathbf{W}$ and the corresponding stress configuration under Q_1 next to each other. The advantage of Q_{SM}, which suppresses the stress to resist the same external wrench, is quite clear. For some target objects, the high stress is concentrated in a very small region and we indicate these regions using black circles. Finally, for applications where globally optimal grasps are too costly to compute and only sub-optimal grasps are needed, users can choose stochastic grasp planning algorithms such as [27], which can return a sub-optimal grasp under Q_{SM} within 10min of computation.

VII. Conclusion and Limitation

We present the SM metric, which reflects the tendency of a target object to be broken during grasping. As a result, a grasp maximizing Q_{SM} will minimize the probability of breaking a fragile object. We show that Q_{SM} can be computed using previous methods and its computational cost can be drastically reduced by progressively detecting the active set. Finally, we show that grasp planning under Q_{SM} can be performed using BB algorithms. Our experiments show that Q_{SM} is aware of geometric fragility while $Q_{1,\infty,MSV,VEW,G11}$ are not. We also show that using Q_{SM} does not increase the computational cost of the resulting grasp planning algorithm.

The major limitation of our work is that computing Q_{SM} requires a costly precomputation step to solve the BEM problem. In addition, the BEM problem requires high-quality, watertight surface meshes of target objects, while in many applications we only have objects represented using point clouds. An avenue of future research is to infer the value of Q_{SM} for given unknown objects using machine learning,

as is done in [19]. Finally, Q_{SM} requires user to input the Poisson ratio, which is a material parameter. We are considering detecting these parameters from casual inputs such as RGBD images.

APPENDIX
THE BOUNDARY ELEMENT METHOD

In this section, we summarize the boundary element discretization of Equation 2 and the definition of $\mathcal{A}_i, \mathcal{B}_i$. In addition, we derive the special form of BEM with our body force and external traction distribution. We follow [26] with minor changes. First, we define a set of notations and useful theorems. For any 3×3 matrix such as $\boldsymbol{\sigma}$, we have:

$$\nabla \cdot \boldsymbol{\sigma} = \left(\nabla \cdot \boldsymbol{\sigma}_x{}^T, \nabla \cdot \boldsymbol{\sigma}_y{}^T, \nabla \cdot \boldsymbol{\sigma}_z{}^T \right)^T.$$

For any 3 vector such as $\mathbf{u}$, we have:

$$\Delta \mathbf{u} = \nabla \cdot (\nabla \mathbf{u}^T) = \left(\Delta \mathbf{u}_x{}^T, \Delta \mathbf{u}_y{}^T, \Delta \mathbf{u}_z{}^T \right)^T$$
$$\nabla \cdot \nabla \mathbf{u} = \nabla \nabla \cdot \mathbf{u} = \nabla \cdot (\mathbf{tr}(\nabla \mathbf{u})\mathbf{I}).$$

A. Elastostatic Equation in Operator Form

We first derive the operator form of the elastostatic problem. By combining the three equations in Equation 2, we have:

$$\begin{aligned} 0 &= \nabla \cdot (\mu \nabla \mathbf{u} + \mu \nabla \mathbf{u}^T + \lambda \mathbf{tr}(\nabla \mathbf{u})\mathbf{I}) + \mathbf{g} \\ &= -\mathcal{L}[\mathbf{u}] + \mathbf{g} = \nabla \cdot \boldsymbol{\sigma} + \mathbf{g} \\ \mathcal{L}[] &\triangleq -(\mu + \lambda)\nabla \nabla \cdot [] - \mu \Delta [] . \end{aligned} \tag{12}$$

B. Boundary Integral Equation (BIE)

Next, we derive BIE via the divergence theorem:

$$\begin{aligned} \int_\Omega \mathbf{v}^T \mathbf{g} d\mathbf{s} &= \int_\Omega \mathbf{v}^T \mathcal{L}[\mathbf{u}] d\mathbf{x} = -\int_\Omega \mathbf{v}^T \nabla \cdot \boldsymbol{\sigma} d\mathbf{x} \\ &= -\int_{\partial\Omega} \mathbf{v}^T \boldsymbol{\sigma} \mathbf{n} d\mathbf{s} + \mathbf{Sym}(\mathbf{u}, \mathbf{v}), \end{aligned}$$

where $\mathbf{Sym}(\mathbf{u}, \mathbf{v})$ denotes a symmetric term in $\mathbf{u}, \mathbf{v}$ satisfying: $\mathbf{Sym}(\mathbf{u}, \mathbf{v}) = \mathbf{Sym}(\mathbf{v}, \mathbf{u})$. We then swap $\mathbf{u}, \mathbf{v}$ and subtract the two equations to get:

$$\begin{aligned} \int_\Omega (\mathbf{u}^T \mathcal{L}[\mathbf{v}] - \mathbf{v}^T \mathcal{L}[\mathbf{u}]) d\mathbf{x} &= \int_{\partial\Omega} (\mathbf{v}^T \mathcal{N}[\mathbf{u}] - \mathbf{u}^T \mathcal{N}[\mathbf{v}]) d\mathbf{s} \\ \mathcal{N}[] \triangleq \boldsymbol{\sigma}[]\mathbf{n} &= \mu \nabla [] \mathbf{n} + \mu \nabla []^T \mathbf{n} + \lambda \nabla \cdot [] \mathbf{n}. \end{aligned} \tag{13}$$

C. Fundamental Solution

If $\mathbf{v}$ is the fundamental solution centered at $\tilde{\mathbf{x}}$, which is denoted by $U(\mathbf{x} - \tilde{\mathbf{x}})$, then we get the boundary integral equation by plugging U into Equation 13:

$$\int_\Omega U^T \mathbf{g} d\mathbf{x} + \int_{\partial\Omega} (U^T \mathcal{N}[\mathbf{u}] - \mathcal{N}[U]^T \mathbf{u}) d\mathbf{s} = \mathbf{u}(\tilde{\mathbf{x}}), \tag{14}$$

where the fundamental solution satisfying:

$$\int_\Omega \mathcal{L}[U(\mathbf{x} - \tilde{\mathbf{x}})] \mathbf{u}(\mathbf{x}) d\mathbf{x} = \mathbf{u}(\tilde{\mathbf{x}}),$$

has the following analytic form:

$$\begin{aligned} U(\mathbf{x} - \tilde{\mathbf{x}}) &= \frac{1}{8\pi\mu} \left[\Delta r \mathbf{I} - \frac{\lambda + \mu}{\lambda + 2\mu} \nabla^2 r \right] \\ &= \frac{1}{16\pi\mu(1-\nu)r} \left[(3 - 4\nu)\mathbf{I} + \nabla r \nabla r^T \right] \\ r &\triangleq |\mathbf{x} - \tilde{\mathbf{x}}| \quad \nu = \frac{\lambda}{2(\lambda + \mu)}. \end{aligned} \tag{15}$$

D. Body Force Term

Equation 14 still involves a volume integral, but we can reduce that to a surface integral by using the special form of body force: $\mathbf{g}(\mathbf{x}) = \mathbf{g}_0 + \nabla \mathbf{g} \mathbf{x}$ and the Galerkin vector form of the fundamental solution (Equation 15). The body force term involves two basic terms. The first one is:

$$\begin{aligned} \int_\Omega \Delta r \mathbf{g} d\mathbf{x} &= \int_\Omega \nabla \cdot (\nabla r \mathbf{g}^T - r \nabla \mathbf{g}^T) d\mathbf{x} \\ &= \int_{\partial\Omega} \left[\mathbf{g} \nabla r^T - r \nabla \mathbf{g} \right] \mathbf{n} d\mathbf{s}. \end{aligned}$$

The second one is:

$$\begin{aligned} \int_\Omega \nabla^2 r \mathbf{g} d\mathbf{x} &= \int_\Omega \nabla \cdot (\mathbf{g} \nabla r^T - \mathbf{tr}(\nabla \mathbf{g}) r \mathbf{I}) d\mathbf{x} \\ &= \int_{\partial\Omega} \left[\nabla r \mathbf{g}^T - \mathbf{tr}(\nabla \mathbf{g}) r \mathbf{I} \right] \mathbf{n} d\mathbf{s}. \end{aligned}$$

Plugging these two terms into Equation 14, we get:

$$\begin{aligned} \int_\Omega U^T \mathbf{g} d\mathbf{x} &= \int_{\partial\Omega} \mathbf{G} \mathbf{n} d\mathbf{s} \\ \mathbf{G} &\triangleq \frac{1}{8\pi\mu} \left[(\mathbf{g} \nabla r^T - r \nabla \mathbf{g}) - \frac{\lambda + \mu}{\lambda + 2\mu} (\nabla r \mathbf{g}^T - \mathbf{tr}(\nabla \mathbf{g}) r \mathbf{I}) \right]. \end{aligned} \tag{16}$$

E. Singular Integrals

At this step all the terms in Equation 13 have been transformed into boundary integrals. However, $\mathbf{x}$ in this form must be interior to Ω. In this section, we take the limit of $\mathbf{x}$ to $\partial\Omega$ and derive the Cauchy principle value of singular integral terms.

The first integral in Equation 13, or the body force term in Equation 16, has removable singularity so that we can use numerical techniques to integrate them directly. The second term in Equation 13 takes a special form due to our Dirac external force distribution in Equation 2:

$$\begin{aligned} \int_{\partial\Omega} U^T \mathcal{N}[\mathbf{u}] d\mathbf{s} &= -\int_{\partial\Omega} U^T \sum_{i=1}^N \delta(\mathbf{x} - \mathbf{x}_i) \mathbf{f}_i d\mathbf{s} \\ &= -\sum_{i=1}^N U(\mathbf{x}_i - \tilde{\mathbf{x}}^*) \mathbf{f}_i. \end{aligned}$$

which is also non-singular. The third term in Equation 13 is singular and the value must be determined as follows:

$$\begin{aligned} &\lim_{\epsilon \to 0} \int_{\partial\Omega} \mathcal{N}[U]^T \mathbf{u} d\mathbf{s} \\ &= \lim_{\epsilon \to 0} \int_{\partial\Omega - B(\epsilon)} \mathcal{N}[U]^T \mathbf{u} d\mathbf{s} + \lim_{\epsilon \to 0} \int_{\partial\Omega \cap B(\epsilon)} \mathcal{N}[U]^T \mathbf{u} d\mathbf{s}, \end{aligned}$$

where we assume that $\|\tilde{\mathbf{x}} - \tilde{\mathbf{x}}^*\| = \epsilon$, $\tilde{\mathbf{x}}^* \in \partial\Omega$, and $B(\epsilon)$ is the sphere centered at $\tilde{\mathbf{x}}^*$ with radius ϵ. We evaluate the two terms separately. For the first term, we have:

$$\begin{aligned} &\lim_{\epsilon \to 0} \int_{\partial\Omega - B(\epsilon)} \mathcal{N}[U]^T \mathbf{u} d\mathbf{s} \triangleq \mathcal{D}[\mathbf{u}] \\ &= \int_{\partial\Omega} [2\mu (\mathbb{M}[U])^T \mathbf{u} + (\mathbb{M}[\frac{1}{4\pi r}])\mathbf{u} + \mathbf{n}^T \nabla [\frac{1}{4\pi r}] \mathbf{u}] d\mathbf{s} \\ &= \int_{\partial\Omega} [2\mu U \mathbb{M}[\mathbf{u}] - \frac{1}{4\pi r} \mathbb{M}[\mathbf{u}] + \mathbf{n}^T \nabla [\frac{1}{4\pi r}] \mathbf{u}] d\mathbf{s} \\ &\mathbb{M}[] \triangleq \nabla [] \mathbf{n}^T - \mathbf{n} \nabla []^T, \end{aligned}$$

which is known as double layer potential and only has removable singularities. To evaluate the second term, we use the following identity:

$$\begin{aligned} &\lim_{\epsilon \to 0} \int_{\partial\Omega \cap B(\epsilon)} \mathcal{N}[U]^T \mathbf{u} d\mathbf{s} \\ &= \lim_{\epsilon \to 0} \int_{\partial(\Omega \cap B(\epsilon))} \mathcal{N}[U]^T \mathbf{u} d\mathbf{s} - \lim_{\epsilon \to 0} \int_{\partial B(\epsilon) \cap \Omega} \mathcal{N}[U]^T \mathbf{u} d\mathbf{s}. \end{aligned} \tag{17}$$

Again, we break this into two terms. The first term in Equation 17 is easy to evaluate using the divergence theorem:

$$\begin{aligned}&\lim_{\epsilon\to 0}\int_{\partial(\Omega\cap B(\epsilon))}\mathcal{N}[U_x]^T\mathbf{u}d\mathbf{s}\\ =&\lim_{\epsilon\to 0}\int_{\partial(\Omega\cap B(\epsilon))}\mathbf{n}^T\boldsymbol{\sigma}(U_x)^T\mathbf{u}(\tilde{\mathbf{x}}^*)d\mathbf{s}\\ =&-\mathbf{u}(\tilde{\mathbf{x}}^*)^T\lim_{\epsilon\to 0}\int_{\Omega\cap B(\epsilon)}\mathcal{L}[U_x]d\mathbf{s}=-\mathbf{u}_x(\tilde{\mathbf{x}}^*).\end{aligned}$$

The second term in Equation 17 is called the integral free term, which evaluates to:

$$\lim_{\epsilon\to 0}\int_{\partial B(\epsilon)\cap\Omega}\mathcal{N}[U]^T\mathbf{u}d\mathbf{s}\triangleq-\mathbf{C}\mathbf{u}$$

$$\mathbf{C}\triangleq\frac{\phi\mathbf{I}}{4\pi}-\int_{\partial(B(\epsilon)\cap\partial\Omega)}(\mathbf{x}-\tilde{\mathbf{x}}^*)\mathbf{n}^T dl,$$

where ϕ is the internal solid angle at $\tilde{\mathbf{x}}^*$.

F. Putting Everything Together

Plugging all the integrals into Equation 14, we have:

$$\int_{\partial\Omega}\mathbf{G}\mathbf{n}d\mathbf{s}-\sum_{i=1}^{N}U(\mathbf{x}_i-\tilde{\mathbf{x}}^*)\mathbf{f}_i-\mathcal{D}[\mathbf{u}]=\mathbf{C}\mathbf{u}(\tilde{\mathbf{x}}^*),$$

which is a dense system allowing us to solve for $\mathbf{u}$ every-

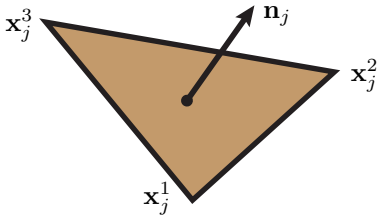

Fig. 5: The jth triangle.

where on $\partial\Omega$. This system is discretized using Galerkin's method with piecewise linear $\mathbf{u}$ and piecewise constant $\mathbf{f}$. All the integrals are evaluated using variable-order Gauss Quadratures. This linear system is denoted by:

$$(\mathbf{D}+\mathbf{C})\mathbf{u}=\mathbf{A}\begin{pmatrix}\mathbf{g}_0\\ [\nabla\mathbf{g}]_x\\ [\nabla\mathbf{g}]_y\\ [\nabla\mathbf{g}]_z\end{pmatrix}+\mathbf{B}\begin{pmatrix}\mathbf{f}_1\\ \vdots\\ \mathbf{f}_N\end{pmatrix},$$

where $\mathbf{D}$ is the coefficient matrix of $\mathcal{D}$, $\mathbf{A}$ is the coefficient matrix of body force terms, and $\mathbf{B}$ is the coefficient matrix of external force terms. After the displacements $\mathbf{u}$ have been computed, we can recover the stress on the jth surface triangle by solving the following linear system:

$$\begin{aligned}&\nabla\mathbf{u}_j(\mathbf{x}_j^2-\mathbf{x}_j^1)=\mathbf{u}(\mathbf{x}_j^2)-\mathbf{u}(\mathbf{x}_j^1)\\ &\nabla\mathbf{u}_j(\mathbf{x}_j^3-\mathbf{x}_j^1)=\mathbf{u}(\mathbf{x}_j^3)-\mathbf{u}(\mathbf{x}_j^1)\\ &\mu(\nabla\mathbf{u}_j+\nabla\mathbf{u}_j^T)\mathbf{n}_j+\lambda\mathbf{tr}(\nabla\mathbf{u}_j)\mathbf{n}_j=\mathbf{f}_i\\ &\boldsymbol{\sigma}(\mathbf{x}_j)=\mu(\nabla\mathbf{u}_j+\nabla\mathbf{u}_j^T)+\lambda\mathbf{tr}(\nabla\mathbf{u}_j)\mathbf{I},\end{aligned}\tag{18}$$

which is 18 linear equations that can be solved for $\nabla\mathbf{u}_j$ and $\boldsymbol{\sigma}(\mathbf{x}_j)$. This linear system is denoted by:

$$\begin{pmatrix}\boldsymbol{\sigma}_x(\mathbf{x}_j)\\ \boldsymbol{\sigma}_y(\mathbf{x}_j)\\ \boldsymbol{\sigma}_z(\mathbf{x}_j)\end{pmatrix}=\mathbf{N}_j\mathbf{u}+\mathbf{M}_j\begin{pmatrix}\mathbf{f}_1\\ \vdots\\ \mathbf{f}_N\end{pmatrix},$$

where $\mathbf{N},\mathbf{M}$ are corressponding coefficient matrices in Equation 18. Combining these two systems, we can define $\mathcal{A}_j,\mathcal{B}_j$ as:

$$\begin{aligned}\mathcal{A}_j\triangleq&\mathbf{N}_j(\mathbf{D}+\mathbf{C})^{-1}\mathbf{A}\\ \mathcal{B}_j\triangleq&\mathbf{N}_j(\mathbf{D}+\mathbf{C})^{-1}\mathbf{B}+\mathbf{M}_j.\end{aligned}$$

REFERENCES

[1] A. Al-Ibadi, S. Nefti-Meziani, and S. Davis, "Active soft end effectors for efficient grasping and safe handling," *IEEE Access*, vol. 6, pp. 23 591–23 601, 2018.

[2] P. Alliez, C. Jamin, L. Rineau, S. Tayeb, J. Tournois, and M. Yvinec, "3D mesh generation," in *CGAL User and Reference Manual*. CGAL Editorial Board, 2019.

[3] E. D. Andersen and K. D. Andersen, "The mosek interior point optimizer for linear programming: an implementation of the homogeneous algorithm," in *High performance optimization*. Springer, 2000, pp. 197–232.

[4] Byoung-Ho Kim, Sang-Rok Oh, Byung-Ju Yi, and Il Hong Suh, "Optimal grasping based on non-dimensionalized performance indices," in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the Next Millennium (Cat. No.01CH37180)*, vol. 2, Oct 2001, pp. 949–956 vol.2.

[5] J. Clausen, *Branch and Bound Algorithms-Principles and Examples*. Citeseer, 1999.

[6] T. A. Cruse, D. Snow, and R. Wilson, "Numerical solutions in axisymmetric elasticity," *Computers & Structures*, vol. 7, no. 3, pp. 445–451, 1977.

[7] H. Dai, A. Majumdar, and R. Tedrake, *Synthesis and Optimization of Force Closure Grasps via Sequential Semidefinite Programming*. Cham: Springer International Publishing, 2018, pp. 285–305.

[8] C. Ferrari and J. Canny, "Planning optimal grasps," in *Proceedings 1992 IEEE International Conference on Robotics and Automation*, May 1992, pp. 2290–2295 vol.3.

[9] P. Fiala and P. Rucz, "Nihu: An open source c++ bem library." *Advances in Engineering Software*, vol. 75, pp. 101–112, 2014.

[10] G. A. Francfort and J.-J. Marigo, "Revisiting brittle fracture as an energy minimization problem," *Journal of the Mechanics and Physics of Solids*, vol. 46, no. 8, pp. 1319–1342, 1998.

[11] T. Granlund and G. D. Team, *GNU MP 6.0 Multiple Precision Arithmetic Library*. United Kingdom: Samurai Media Limited, 2015.

[12] B. Grünbaum and G. C. Shephard, "Convex polytopes," *Bulletin of the London Mathematical Society*, vol. 1, no. 3, pp. 257–300, 1969.

[13] W. Hackbusch, "A sparse matrix arithmetic based on h-matrices. part i: Introduction to h-matrices," *Computing*, vol. 62, no. 2, pp. 89–108, 1999.

[14] K. Hang, J. A. Stork, N. S. Pollard, and D. Kragic, "A framework for optimal grasp contact planning," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 704–711, April 2017.

[15] S. Hert and M. Seel, "dD convex hulls and delaunay triangulations," in *CGAL User and Reference Manual*. CGAL Editorial Board, 2019.

[16] T. J. Hughes, *The finite element method: linear static and dynamic finite element analysis*. Courier Corporation, 2012.

[17] Z. Li and S. S. Sastry, "Task-oriented optimal grasping by multifingered robot hands," *IEEE Journal on Robotics and Automation*, vol. 4, no. 1, pp. 32–44, Feb 1988.

[18] M. Liu, Z. Pan, K. Xu, K. Ganguly, and D. Manocha, "Generating grasp poses for a high-dof gripper using neural networks," *arXiv preprint arXiv:1903.00425*, 2019.

[19] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. Aparicio, and K. Goldberg, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," in *Proceedings of Robotics: Science and Systems*, 07 2017.

[20] J. Mahler, F. T. Pokorny, B. Hou, M. Roderick, M. Laskey, M. Aubry, K. Kohlhoff, T. Kröger, J. Kuffner, and K. Goldberg, "Dex-net 1.0: A cloud-based network of 3d objects for robust grasp planning using a multi-armed bandit model with correlated rewards," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 1957–1964.

[21] J. Mahler, F. T. Pokorny, Z. McCarthy, A. F. van der Stappen, and K. Goldberg, "Energy-bounded caging: Formal definition and 2-d energy lower bound algorithm based on weighted alpha shapes," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 508–515, 2016.

[22] B. Matulevich, G. E. Loeb, and J. A. Fishel, "Utility of contact detection reflexes in prosthetic hand control," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov 2013, pp. 4741–4746.

[23] V. . Nguyen, "Constructing force-closure grasps," in *Proceedings. 1986 IEEE International Conference on Robotics and Automation*, vol. 3, April 1986, pp. 1368–1373.

[24] M. A. Roa and R. Suárez, "Grasp quality measures: review and performance," *Autonomous robots*, vol. 38, no. 1, pp. 65–88, 2015.

[25] J. D. Schulman, K. Goldberg, and P. Abbeel, "Grasping and fixturing as submodular coverage problems," in *Robotics Research*. Springer, 2017, pp. 571–583.

[26] O. Steinbach, *Numerical approximation methods for elliptic boundary value problems: finite and boundary elements*. Springer Science & Business Media, 2007.

[27] Z. Zhang, J. Gu, and J. Luo, "Evaluation of genetic algorithm on grasp planning optimization for 3d object: A comparison with simulated annealing algorithm," in *2013 IEEE International Symposium on Industrial Electronics*, May 2013, pp. 1–8.

[28] Y. Zheng, "An efficient algorithm for a grasp quality measure," *IEEE Transactions on Robotics*, vol. 29, no. 2, pp. 579–585, 2012.

[29] Q. Zhou, J. Panetta, and D. Zorin, "Worst-case structural analysis," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 137:1–137:12, July 2013.