DOI: 10.1111/cgf.14490
EUROGRAPHICS 2022 / R. Chaine and M. H. Kim
(Guest Editors)

*Volume 41 (2022), Number 2*

# Worst-Case Rigidity Analysis and Optimization for Assemblies with Mechanical Joints

Zhenyuan Liu[1,2] Jingyu Hu[1] Hao Xu[†1] Peng Song[3] Ran Zhang[4] Bernd Bickel[2] Chi-Wing Fu[1]

[1]The Chinese University of Hong Kong [2]IST Austria [3]Singapore University of Technology and Design [4]Hasso Plattner Institute

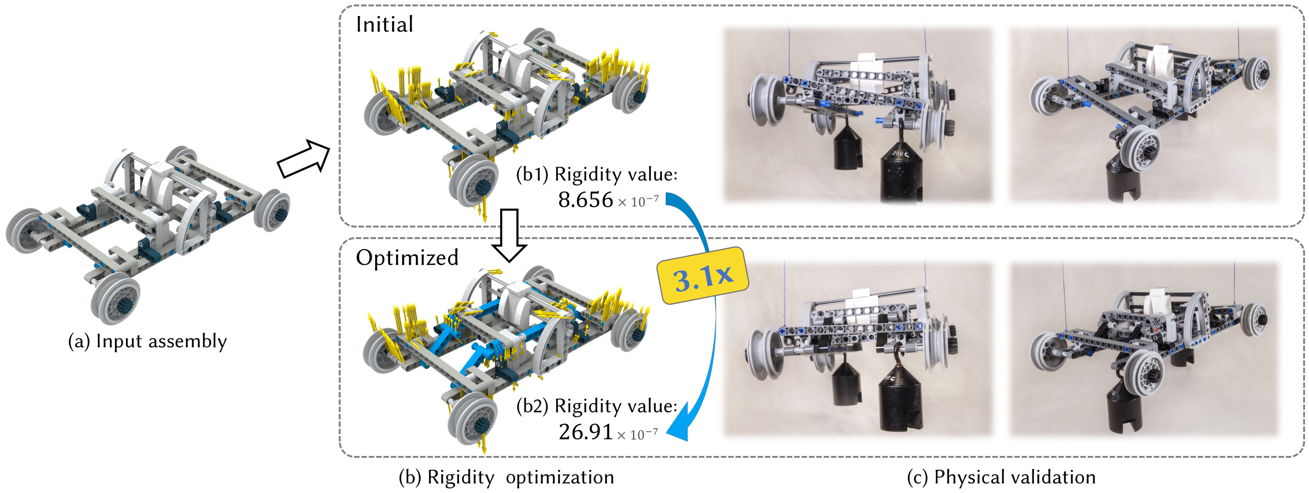


**Figure 1:** *Given the assembly structure of the 153-brick LEGO Technic* ROLLING CHASSIS *(a), our method quantifies the rigidity of the structure (b1) and finds the worst-case external load configuration (yellow arrows) that maximally deforms it (c). A larger rigidity value indicates a more rigid assembly and thus greater resistance to external forces. After we strengthen the initial model as per the recommendation given by our method, the reinforced model (b2) has its rigidity value tripled, and undergoes less deformation than the initial one under the same loads that twist the model. The physical experiments (c) also show results that align with the analysis of our method.*

**Abstract**

*We study structural rigidity for assemblies with mechanical joints. Existing methods identify whether an assembly is structurally rigid by assuming parts are perfectly rigid. Yet, an assembly identified as rigid may not be that "rigid" in practice, and existing methods cannot quantify how rigid an assembly is. We address this limitation by developing a new measure,* worst-case rigidity, *to quantify the rigidity of an assembly as the largest possible deformation that the assembly undergoes for arbitrary external loads of fixed magnitude. Computing worst-case rigidity is non-trivial due to non-rigid parts and different joint types. We thus formulate a new computational approach by encoding parts and their connections into a stiffness matrix, in which parts are modeled as deformable objects and joints as soft constraints. Based on this, we formulate worst-case rigidity analysis as an optimization that seeks the worst-case deformation of an assembly for arbitrary external loads, and solve the optimization problem via an eigenanalysis. Furthermore, we present methods to optimize the geometry and topology of various assemblies to enhance their rigidity, as guided by our rigidity measure. In the end, we validate our method on a variety of assembly structures with physical experiments and demonstrate its effectiveness by designing and fabricating several structurally rigid assemblies.*

**CCS Concepts**

• ***Computing methodologies*** → ***Modeling and simulation;***

## 1. Introduction

Assembly-based objects are ubiquitous. They can be built as dynamic mechanisms that transfer motions or as static structures that

[†] corresponding author

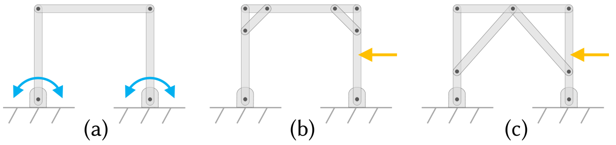


**Figure 2:** *Illustrations of 2D assemblies connected by mechanical joints, e.g., linkages. The dynamic mechanism in (a) can transfer motions (blue arrows) between the left and right poles, whereas the two static structures in (b) and (c) can bear external loads such as lateral loads (orange arrows), with different degrees of rigidity.*

bear loads; see, e.g., the illustrations in Figure 2. As a dynamic mechanism, the relative movements between parts can transfer the driving motion from a motor/crank to the end effectors. For an assembly intended to be a static structure, its parts should have minimum movement relative to one another to resist the external forces and to make the assembly *structurally rigid*.

In this work, *we study the structural rigidity of static assemblies with mechanical joints*. Such assemblies are commonly seen in architecture, furniture, or machine frameworks. To evaluate the rigidity of an assembly, one may employ existing approaches from rigidity theory [TD02]. The core idea behind these approaches is to model an assembly as a graph with fixed edge length embedded in Euclidean space. An assembly is said to be *rigid* if the only continuous motions of the graph nodes that preserve the edge lengths are rigid-body motions of the entire assembly; otherwise, the assembly is said to be *flexible*.

One limitation of these approaches is that they can only classify an assembly as rigid or non-rigid since they assume perfectly-rigid parts and firmly connected joints. However, real materials are rarely perfectly rigid and joints can be loose, so rigid assemblies identified by these approaches may not be that “rigid” in reality. For example, existing approaches would identify both structures in Figure 2(b) and (c) as rigid, yet fail to distinguish how rigid each structure is. As a result, we cannot rely on these approaches to guide the assembly design. The rigidity of an assembly is a result of the physical interaction between all parts and joints, so a small local modification on a part or joint could change the overall mechanical property of the assembly. Hence, analyzing the structural rigidity for assembly is challenging, requiring global considerations over part shapes, material, arrangements, and their joint connections.

We approach the problem by developing a *worst-case rigidity* measure for quantifying the structural rigidity of assemblies, as inspired by a similar concept in structural optimization [ZPZ13]. Our key idea is to estimate the largest possible deformation of the assembly for arbitrary external loads of fixed magnitude, by modeling parts as non-rigid objects and also the effect of joint jitter. An intuitive way for this task is to enumerate a large number of possible load configurations, apply forward finite element method (FEM) simulation to each of them, and then choose the one that leads to the largest deformation. This naïve approach, however, induces huge computation, with trials and errors.

We formulate a new and unified framework, in which parts are treated as non-rigid and modeled using finite elements, and joints are modeled as soft constraints that restrict the movements of the parts. By this means, we represent each part by a local stiffness matrix and aggregate these matrices with the joints to form a global stiffness matrix that represents the entire assembly. Furthermore, we cast the worst-case rigidity problem as an optimization that seeks the least deformation on the assembly for arbitrary external loads. We solve the optimization by using eigenanalysis, where the smallest positive eigenvalue gives the worst-case rigidity measure, and the associated eigenvector gives the worst-case external load.

Also, the rigidity analysis is differentiable, so we can readily adopt it to guide an optimization on the geometry and topology of the assembly to improve its structural rigidity. Therefore, the only input to our rigidity analysis and optimization is the assembly itself; unlike existing methods, prescribed loads are not required.

Overall, this work has the following contributions.

- First, we formulate worst-case rigidity analysis for static assemblies with mechanical joints and develop a unified and efficient framework for computing worst-case rigidity and finding the corresponding load configuration.
- Second, we develop methods to optimize the geometry and topology of assemblies to maximize the worst-case rigidity, while minimizing necessary modifications to the structure.

We validate our approach on a variety of assemblies, including bar-and-joint frameworks, 2D mechanisms, spatial linkages, 3D body-and-joint framework, LEGO Technic assemblies, and several everyday objects. Also, we conduct physical experiments to verify our rigidity measure and demonstrate its effectiveness. Lastly, we employ our method to guide the design of several structurally rigid assemblies with mechanical joints.

## 2. Related Work

**Assemblies with Mechanical Joints.** A mechanical joint such as a hinge or ball joint connects parts while allowing the parts to exhibit relative movement. Assemblies with mechanical joints can form dynamic mechanisms for transferring motions [MYY*10], e.g., a linkage-based mechanical toy that converts an input motion from a crank or motor to generate intriguing motions [CTN*13, TCG*14]. Assemblies with mechanical joints can also form objects whose poses are manually adjustable [KLY*14, UTZ16, NBA19]. By the frictional contacts at joints, these articulated models [BBJP12, CCA*12] can be stable at desired poses. In contrast, the pose of robots can be adjusted automatically, e.g., by using motors to control the joint angles for performing locomotion [MTN*15].

Also, assemblies with mechanical joints can be employed to create transformable designs, e.g., foldable assemblies [ZSMS14, LHAZ15, MEKM17], deployable structures [ZWC*16, ZSC16], and reconfigurable assemblies [GJG16, LMaH*18, YZC18]. Transformable designs focus on functionalities to be delivered by different configurations of the assembly, in which the mechanical joints define the transformations between the configurations. So far, little attention has been paid to the rigidity of each assembly configuration, which, however, is critical for practical usage. Our work addresses this limitation by proposing an efficient way to quantify and evaluate the rigidity of an assembly, and automatically suggesting how parts can be modified and/or added to an assembly configuration to enhance its rigidity.

**Structural Rigidity.** Structural rigidity was initially studied by mathematicians and engineers [TD02]. To analyze the rigidity properties of a framework, the framework is often first represented as a graph embedded in Euclidean space. Laman [Lam70] developed the first complete combinatorial characterization of bar-and-joint frameworks in 2D, enabling one to test the rigidity of a framework by counting the vertices and edges in a graph and its subgraphs. Tay [Tay84] addressed a generalized rigidity problem for body-and-bar frameworks in arbitrary dimensions, in which the rigid bodies are linked by bars with ball joints at the ends. Besides bars, hinges can also be used to connect rigid bodies. Tay [Tay89] studied the rigidity of body-and-hinge frameworks composed of rigid bodies articulated by hinges, while Jackson and Jordán [JJ10] studied the rigidity of body-bar-and-hinge frameworks, in which the rigid bodies are connected by bars or hinges. Instead of assuming rigid bodies connected by bars and/or hinges, Haller et al. [HLSS*12] studied a more general body-and-cad framework composed of rigid bodies three-dimensionally constrained by pairwise *c*oincidence, *a*ngular and *d*istance constraints.

Following this line of research, this work focuses on studying the rigidity of assembly structures connected by mechanical joints. Compared to the state-of-the-art, our work is unique in two aspects. First, we propose a *worst-case rigidity analysis* approach that is able not only to test the rigidity of a given assembly but also to quantify how rigid the assembly is under unknown external forces. Second, we take our rigidity analysis to *guide the design and optimization* of assemblies with several types of mechanical joints.

**Structural Stability.** A concept closely related to structural rigidity is *structural stability*. Unlike assemblies with mechanical joints, structurally stable assemblies often consist of rigid parts connected by integral joints (e.g., mortise and tenon joints) and/or planar/curved contacts, and are deployed mainly as static structures.

Fundamentally, an assembly is structurally stable, if it is in static equilibrium under external forces and torques. The equilibrium method [WOD09, WSP21a] is the current state-of-the-art for static analysis of 3D assemblies in computer graphics research, and the method has been employed for designing structurally sound masonry [WSW*12], LEGO sculptures [LYH*15], and furniture with decorative joints [YKGA17]. An assembly is globally interlocking, if it is in equilibrium under arbitrary external forces and torques [WSIP19]. Due to the static-kinematic duality, global interlocking can be tested efficiently by using a kinematic method [WSIP19]. Several computational methods have been developed to construct interlocking assemblies for different applications, including puzzles [SFCO12] and 3D printed objects [YCXW17]. Please refer to a recent survey [WSP21b] for more details on stability analysis methods.

**Structural Analysis.** Structural analysis is to determine the effect of loads on a physical object. When the external loads are known, finite element method is a widely-used technique in structural analysis [SB12]. Taking the structural analysis as a foundation, shape optimization can be further performed to minimize an object's material cost while preserving its structural strength [MHR*16]. Recently, several computational methods were developed for structural analysis *without prescribed loads*, including worst-case structural analysis [ZPZ13, ZCT16, SZB18], stochastic structural analysis [LSD*16], and critical instant structural analysis under force location uncertainty [UMK17]. Our approach shares the same spirit as the methods for worst-case structural analysis, in which the basic idea is to perform eigenanalysis to identify the weakest areas and the worst load configuration on an object. Instead of analyzing a single solid object, our method performs worst-case rigidity for assemblies, where the biggest challenge is to model the complex interactions among the parts, which are connected through different kinds of mechanical joints. Also, we cannot simply assume the adjacent parts are fixed relative to one another (like being glued or welded together as a single solid object), because mechanical joints allow various kinds of relative movement between adjacent parts. Therefore, we have to model these properties of parts and joints and consider them in our worst-case rigidity analysis.

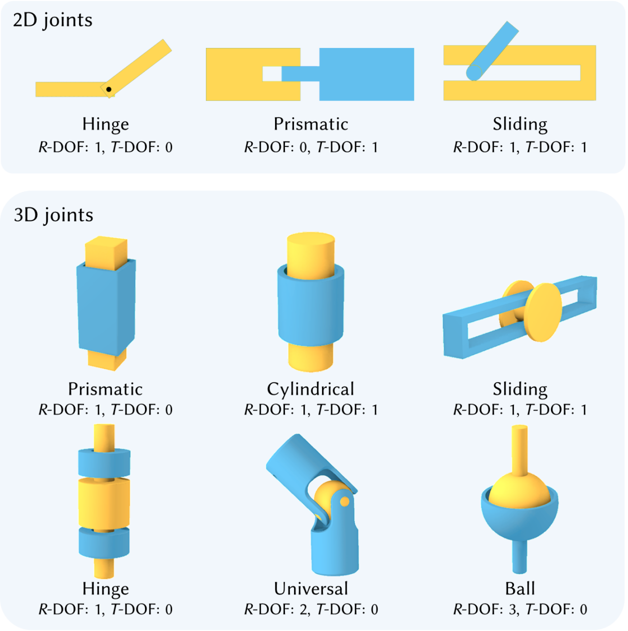
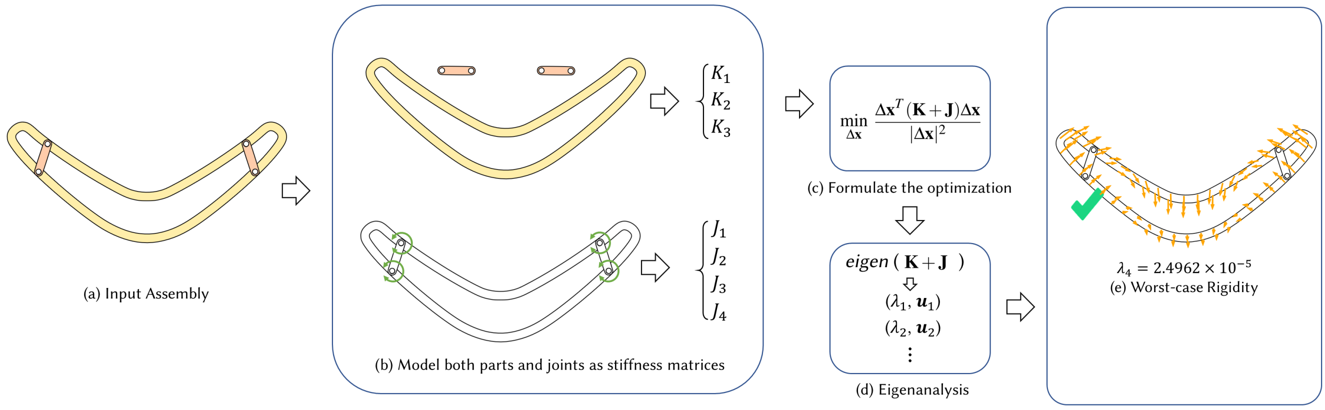


**Figure 3:** *Common mechanical joints supported by our computational approach. R-DOF and T-DOF below each joint model denote the associated degrees of freedom for joint rotation and translation, respectively.*

## 3. Overview

Our goal is to design mechanical assemblies that can stay rigid at certain configurations, i.e., not easily deform/break, under (unknown) external forces. Example assemblies that we attempt to work with include LEGO Technic models (e.g., Figure 1), linkage-based models (e.g., Figure 4), and reconfigurable assemblies (e.g., Figure 14 (a)). In contrast to the existing methods of rigidity analysis that assume perfectly rigid parts, our approach only requires nearly rigid parts in the assembly. Also, our approach is able to model various common mechanical joints in the analysis; see Figure 3 for a list of the joints and the associated motion constraints.

To enable the design of structurally rigid assemblies, our computational approach should support the following four tasks:


$K_1$
$K_2$
$K_3$
$J_1$
$J_2$
$J_3$
$J_4$
$\min_{\Delta\mathbf{x}} \frac{\Delta\mathbf{x}^T(\mathbf{K}+\mathbf{J})\Delta\mathbf{x}}{|\Delta\mathbf{x}|^2}$
eigen ( $\mathbf{K}+\mathbf{J}$ )
$(\lambda_1, \boldsymbol{u}_1)$
$(\lambda_2, \boldsymbol{u}_2)$
$\lambda_4 = 2.4962 \times 10^{-5}$

(a) Input Assembly

(b) Model both parts and joints as stiffness matrices

(c) Formulate the optimization

(d) Eigenanalysis

(e) Worst-case Rigidity

**Figure 4:** *Overview of our worst-case rigidity analysis approach. From an input assembly (a), our method first computes the stiffness matrix of each part, as well as each connecting joint (b). Using this representation, worst-case rigidity analysis can be formulated as an optimization (c) and solved by an eigenanalysis (d). In our framework, the smallest non-zero eigenvalue and the associated eigenvector correspond to the worst-case rigidity measure and the associated load configuration, respectively (e). Throughout this paper, we use orange arrows (as in (e)) to indicate the worst-case load configuration.*

(i) *Test the rigidity.* Analyze the given assembly and determine whether it is structurally rigid or flexible.

(ii) *Identify potential motions.* If the assembly is flexible, find the movable parts, as well as their infinitesimal motion.

(iii) *Estimate the degree of rigidity.* Estimate how rigid the given assembly is, subject to the worst-case load configuration.

(iv) *Enhance the rigidity.* Automatically modify the geometry of the existing parts, as well as make suggestions on how to introduce new part(s), to enhance the rigidity.

While tasks (i) and (ii) can possibly be handled by a combination of existing methods, e.g., [HLSS*12, PTV*17], we aim for a computational approach that addresses all the four tasks with a *unified and efficient computational framework*.

Supporting these tasks involves several technical challenges. First, we do not assume perfectly rigid parts, so we need to consider the influence of the physical material of every individual part on the rigidity of the whole assembly. Second, we aim to analyze the rigidity without assuming prescribed external loads, and simply trying out all possible load configurations is computationally intractable. Lastly, we need to be able to represent and analyze both the static properties (tasks (i) and (iii)) and dynamics (task (ii)) of the assemblies in a unified manner.

**Overview of our approach.** To meet these challenges, we first formulate an optimization model for analyzing the worst-case rigidity, in which we represent parts as stiffness matrices and joints as soft constraints on the constrained motions (Figure 4(b)). Then, we formulate the optimization to seek the structural deformation that produces the least elastic potential energy in the assembly structure (Figure 4(c)). Our optimization model can be solved efficiently by finding the eigenvectors of a matrix derived from our parts and joints representations (Figure 4(d&e)). Ignoring the joint friction, our analysis can also identify movable parts (tasks (i) and (ii)) in an assembly, which correspond to structural deformations with zero energy. Thanks to the eigenvalue, our rigidity measure is differentiable with respect to the geometric parameters of the input assembly. Hence, to enhance the assembly's rigidity, we can derive a gradient-based method to automatically optimize the geometric parameters of the assembly. Likewise, we can take the analysis results to guide the modification of the assembly's topology, e.g., by adding reinforcement parts; see, e.g., Figure 1(b).

## 4. Worst-Case Rigidity Analysis

In this section, we first present our formulation of worst-case rigidity analysis as an optimization problem (Section 4.1) and then describe how to solve the problem to obtain the rigidity measure (Section 4.2). For simplicity, our method is illustrated with 2D examples, and can be naturally extended to 3D.

### 4.1. Problem Formulation

Given an assembly with $n$ parts and $m$ joints, we denote each part as $P_i, i \in \{1,...,n\}$ and each joint as $J_k, k \in \{1,...,m\}$. In this work, we consider each part as an individual component that builds up the assembly and each joint as a set of soft constraints (not a physical object) on relative part movements. If more than two parts joined together, we define pair-wise constraints among the connecting parts.

**Modeling parts.** We represent parts using FEM, so each part $P_i$ is composed of a set of elements and nodes. Elements can be bars (see Figure 2), 2D triangles (see Figure 8(b)), or 3D tetrahedra (see Figure 14), and we create them using existing triangulation and tetrahedralization tools, such as [HZG*18]. Denoting $\mathbf{x}_i = \{\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, ...\}^T$ as the position of $P_i$'s nodes, the deformation of $P_i$ can be captured by $\Delta\mathbf{x_i}$, a column vector storing the displacements of nodes in $P_i$. Furthermore, for each part $P_i$, we compute stiffness matrix $\mathbf{K}_i$ based on the standard FEM theory [SB12]. The stiffness matrix $\mathbf{K}_i$ maps displacements $\Delta\mathbf{x_i}$ to external forces $\mathbf{f_i}$ exerted on $P_i$'s nodes, represented as linear equation $\mathbf{f}_i = \mathbf{K}_i\Delta\mathbf{x_i}$. By concatenating the displacement vector $\Delta\mathbf{x}_i$ of all parts, we can form

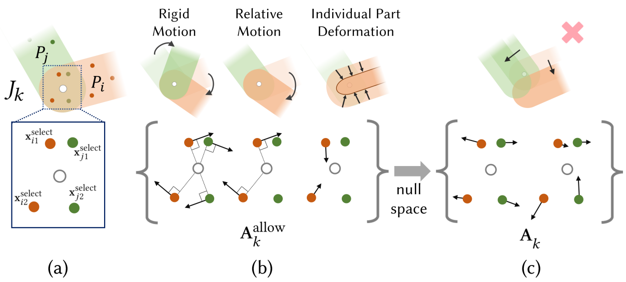

Rigid Motion
Relative Motion
Individual Part Deformation
$P_j$
$J_k$
$P_i$
$\mathbf{x}_{i1}^{\text{select}}$
$\mathbf{x}_{j1}^{\text{select}}$
$\mathbf{x}_{i2}^{\text{select}}$
$\mathbf{x}_{j2}^{\text{select}}$
null space
$\mathbf{A}_k^{\text{allow}}$
$\mathbf{A}_k$
(a)
(b)
(c)

**Figure 5:** *Modeling a 2D hinge joint. We select nearby nodes in each connecting part (a) to build matrix $\mathbf{A}_k^{allow}$ whose span represents the feasible movements of the selected nodes allowed by the joint (b). The null space of $\mathbf{A}_k^{allow}$, i.e., $\mathbf{A}_k$, represents all nodes movements that are constrained by the joint (c).*

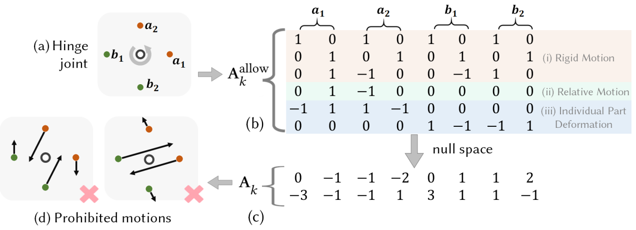


**Figure 6:** *A running example that illustrates how to formulate the linear equations for a hinge joint (a): for each node, $a_i$ and $b_i$, we concatenate their allowed instantaneous movements into row vectors of $\mathbf{A}_k^{allow}$ (b) and find the basis of the null space spanned by $\mathbf{A}_k^{allow}$, denoted by $\mathbf{A}_k$ (c). The row vectors of matrix $\mathbf{A}_k$ represents the constrained motions by the joint (d). To represent node translations is in fact easier, as we can directly use the allowed translation vector for all related nodes in composing $\mathbf{A}_k^{allow}$.*

the overall displacement vector $\Delta\mathbf{x}$ of the entire assembly. Similarly, we aggregate the stiffness matrix $\mathbf{K}_i$ of all parts, and arranging them on the block diagonal, we can build the overall stiffness matrix $\mathbf{K}$ for the entire assembly. By doing so, we can measure the elastic potential energy induced by part deformation of the whole assembly using

$$E_P = \Delta\mathbf{x}^T \mathbf{K} \Delta\mathbf{x}. \tag{1}$$

**Modeling joints.** To model a joint, say joint $J_k$ that connects parts $P_i$ and $P_j$, for each part we sample two nodes from the set of nodes used for computing the FEM stiffness matrix of that part. More specifically, we choose the two nodes that are the closest to the contact surface between the two parts. Then, we aggregate the displacement vectors of the selected nodes in $P_i$ and $P_j$ into column vectors $\Delta\mathbf{x}_i^{\text{select}}$ and $\Delta\mathbf{x}_j^{\text{select}}$, respectively; see Figure 5(a). Hence, our goal of modeling joint $J_k$ is to compose matrix $\mathbf{A}_k$ based on $J_k$'s properties, where the row vectors of $\mathbf{A}_k$ span the space of part movements constrained by $J_k$. By doing so, we can formulate $\Delta\mathbf{n}_k = \mathbf{A}_k[(\Delta\mathbf{x}_i^{\text{select}})^T, (\Delta\mathbf{x}_j^{\text{select}})^T]^T$ to quantify the nodes displacement that is constrained by $J_k$, where the dimension of $\mathbf{n}_k$ is the DOFs of the constrained motion of $J_k$.

Directly constructing the matrix elements in $\mathbf{A}_k$ is not easy, since it is not intuitive to locate all possible part movements prohibited by joint $J_k$. Hence, we circumvent the problem by first composing matrix $\mathbf{A}_k^{\text{allow}}$ whose row vectors span different feasible $\Delta\mathbf{x}_i^{\text{select}}$ and $\Delta\mathbf{x}_j^{\text{select}}$ allowed by $J_k$. In other words, matrix $\mathbf{A}_k^{\text{allow}}$ represents the space of all feasible nodes movements allowed by joint $J_k$. Here, we can obtain $\mathbf{A}_k$ by computing a basis of the null space of $\mathbf{A}_k^{\text{allow}}$; see Figure 5(b) and (c). Note also that displacement vector $\Delta\mathbf{x}$ in our formulation represents infinitesimal movements of nodes in the given assembly pose, so that we can use linear equations to model different types of joint and part motions in a unified manner.

In detail, matrix $\mathbf{A}_k^{\text{allow}}$ can be composed row-by-row by considering three types of feasible node displacements (see Figure 6(b) for a running example on how to compose $\mathbf{A}_k^{\text{allow}}$ and use its null space to find the prohibited movements for a hinge joint):

- *Rigid motion.* First, the parts can move all together as a single rigid body. Since rigid bodies have three DOFs in 2D, we create three rows in $\mathbf{A}_k^{\text{allow}}$, two for translations (2D) and one for rotation (1D); see the top red area (i) in Figure 6(b).
- *Relative part motion allowed by joint.* Second, the parts can perform relative rigid motions allowed by the joint. For each DOF allowed by joint $J_k$, we create one row vector in $\mathbf{A}_k^{\text{allow}}$; see Figure 3 for translation and rotation DOFs allowed by different joints. To do so, we fix part $P_j$ and translate/rotate nodes in $\mathbf{x}_i^{\text{select}}$ according to the associated DOF allowed by the joint; see the middle green area (ii) in Figure 6(b).
- *Deformation of parts.* Third, each part can undergo deformation without affecting others. Since it is not easy to directly enumerate all possible node displacements associated with part deformations, we again circumvent the problem by using null space. If the node displacements of a part are not caused by rigid motion or relative part motions, they must be caused by deformation. Hence, for each part, we first find a basis of its rigid motion and relative part motions, and then compute the basis of its null space as the space of node displacements of its part deformation; e.g., see area (iii) in Figure 6(b), $\{-1,1,1,-1\}$ is the null space of the four-by-four submatrix above it and we pad zeros for the remaining row elements not related to the part.

Next, by finding the null space of the row vectors of $\mathbf{A}_k^{\text{allow}}$, we can obtain $\mathbf{A}_k$ and establish the following elastic energy term to quantify the deviation of the nodes isplacement from the allowed motion of joint $J_k$:

$$e_k = \Delta\mathbf{n}_k^T \mathbf{S}_k \Delta\mathbf{n}_k, k \in \{1...m\}.$$

Here, $e_k$ measures the elastic potential energy induced by the constrained joint movement, and $\mathbf{S}_k$ is a diagonal matrix that measures the "stiffness" or "jitter" of the joint along different degrees of freedom. We determine the value of coefficient parameters in $\mathbf{S}_k$ by physical experiments, where our method searches for the paramater value that best matches the physical behavior of the real joint; see Section 6 for details. Recall that $\Delta\mathbf{n}_k = \mathbf{A}_k[(\Delta\mathbf{x}_i^{\text{select}})^T, (\Delta\mathbf{x}_j^{\text{select}})^T]^T$ and $e_k = \Delta\mathbf{n}_k^T \mathbf{S}_k \Delta\mathbf{n}_k$, and by combining all energy terms $\{e_k\}_{k=1}^m$ into a single term, we obtain the deviation from the allowed movements of all joints in the assembly:

$$E_J = \Delta\mathbf{x}^T \mathbf{J} \Delta\mathbf{x}, \tag{2}$$

where $\mathbf{J} = \mathbf{A}^T\mathbf{S}\mathbf{A}$ represents the stiffness matrix for all joints, $\mathbf{A}$ represents the matrix aggregated from $\{\mathbf{A}_k\}_{k=1}^m$ for all joints, and $\mathbf{S}$ represents the stiffness coefficients aggregated from $\{\mathbf{S}_k\}_{k=1}^m$ of all joints. Note that $\mathbf{A} \in \mathbb{R}^{D\times 2N}$ and $\Delta\mathbf{x} \in \mathbb{R}^{2\text{N}}$ for 2D assemblies, where $N$ is the total number of nodes and $D$ is the total number of rows of all $\{\mathbf{A}_k\}_{k=1}^m$.

**Our worst-case rigidity model.** We model *worst-case rigidity* as an optimization problem, for finding node displacements $\Delta\mathbf{x}$ that minimizes the elastic potential energy induced by part and joint deformation ($\Delta\mathbf{x}$).

$$\min_{\Delta\mathbf{x}} \quad \frac{E(\Delta\mathbf{x})}{|\Delta\mathbf{x}|^2} \tag{3}$$

$$\text{with} \quad E(\Delta\mathbf{x}) = E_P + E_J = \Delta\mathbf{x}^T(\mathbf{K}+\mathbf{J})\Delta\mathbf{x},$$

where $|\cdot|$ denotes vector norm, $E_P$ is the potential energy of part deformation, and $E_J$ is the potential energy of joint deformation.

The solution space of Eq. (3) contains two kinds of part movements: (i) rigid part movements (if any), which are allowed by the joints, and (ii) part and joint deformations. First of all, if $E(\Delta\mathbf{x})$ is zero for a non-zero $\Delta\mathbf{x}$, it means that one or more parts in the structure are movable at some joints without inducing energy (i.e., without deformation of parts and violating joint constraints). On the other hand, if $E(\Delta\mathbf{x})$ is non-zero, it means that the structure is under deformation. By fixing the squared length of displacement vector $|\Delta\mathbf{x}|^2$, the relationship between $E(\Delta\mathbf{x})$ and $\Delta\mathbf{x}$ can reflect the stiffness of an assembly structure. The stiffer the structure is, the more energy is required to cause the same amount of deformation, i.e., $|\Delta\mathbf{x}|^2$. Therefore, we can measure the rigidity of a structure as the ratio between $E(\Delta\mathbf{x})$ and $|\Delta\mathbf{x}|^2$; see again Eq. (3). This objective value is invariant under scaling $|\Delta\mathbf{x}|$, since $\frac{E(\Delta\mathbf{x})}{|\Delta\mathbf{x}|^2} = \frac{E(c\Delta\mathbf{x})}{|c\Delta\mathbf{x}|^2}$ for any non-zero constant $c$. Thus, by solving Eq. (3), we can obtain *the direction of* $\Delta\mathbf{x}$, which indicates how the nodes move to induce the largest deformation with least energy on the structure.

**Modeling fixed nodes.** In case one or more nodes in an assembly are fixed to the ground or fixed to some other structures, we adopt a common implementation in FEM by simply removing corresponding rows and columns in $\mathbf{K}+\mathbf{J}$ and also the corresponding dimension in $\Delta x$.

**Visualization of our optimization problem.** To study the relationship between the optimization problem (Eq. (3)) and worst-case rigidity, we create two simple structures with only one movable node, as shown in Figure 7. For simplicity, we fix node(s) to the ground and set the coefficients of joint constraints to very large values, such that we have $\Delta\mathbf{x} \in \mathbb{R}^2$ and can explore how $E(\Delta\mathbf{x})$ changes with $\Delta\mathbf{x}$. From the optimization landscape, Eq. (3) finds the direction $\Delta\mathbf{x}$ that induces the least increase in elastic potential energy (vertical axis in the plots in Figure 7); It points in the direction of the smallest second-order derivative of the energy. in particular, this $\Delta\mathbf{x}$ is the instantaneous node displacement from the current configuration of the structure, and the second-order derivative in this direction indicates the amount of energy introduced by an infinitesimal $\Delta\mathbf{x}$. If the structure is not rigid, we can identify the movable parts and $\Delta\mathbf{x}$ indicates the part movement directions.

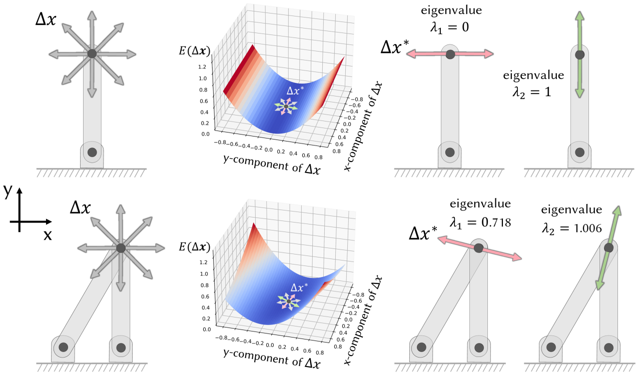


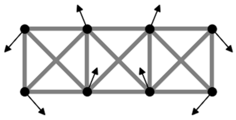
**Figure 7:** *Visualizing how our formulation works on two simple 2D assemblies (top-left & bottom-left), both with only one movable node (marked by* $\Delta\mathbf{x}$*); all other nodes near the ground are fixed. For each assembly, we plot how the elastic potential energy* $E(\Delta\mathbf{x})$ *changes with node displacement* $\Delta\mathbf{x}$*. The plot shows that the optimal solution of Eq. (3) is the direction of the smallest second-order derivative from the origin (i.e.,* $\Delta\mathbf{x}^*$*), marked by the red arrows in the plots. This direction can be found by computing the eigenvector associated with the smallest eigenvalue (right). Note also that the assembly on top is flexible; our method can identify that it is not rigid (*$\lambda_1 = 0$*) and find the movable direction (red arrow).*

## 4.2. Solve for the worst-case rigidity

Based on our formulation, Eq. (3) can be directly converted into a eigenproblem by the Courant-Fischer theorem [HJ12], where the optimal value corresponds to the smallest eigenvalue of the matrix $\mathbf{K}+\mathbf{J}$ and the optimal solution is the associated eigenvector.

**Interpretation of the eigenvalues and eigenvectors.** We denote the eigenvalues and eigenvectors computed for solving Eq. (3) by $(\lambda_1, \mathbf{u}_1)$, $(\lambda_2, \mathbf{u}_2)$, $\cdots$ $(\lambda_r, \mathbf{u}_r)$, where $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_r$ and $r$ is dimension the corresponding matrix. The optimal value of Eq. (3) is $\lambda_1$ and the corresponding optimal solution is $\Delta\mathbf{x}^* = \mathbf{u}_1$. As illustrated in Figure 7, the smallest eigenvalue $\lambda_1$ indicates the smallest second-order derivative of $\mathbf{E}$ over all the directions from the origin, revealing the amount of potential energy in deforming the assembly under the worst-case external load, and $\Delta\mathbf{x}^*$ is the direction of the smallest second-order derivative. It indicates how the assembly is deformed by the worst-case node displacement. Furthermore, we compute the worst-case load configuration by $\mathbf{f} = (\mathbf{K}+\mathbf{J})\Delta\mathbf{x}^*$, where $\mathbf{K}+\mathbf{J}$ is the stiffness matrix of the assembly and $\mathbf{f}$ is the corresponding external load that causes the node displacements.

If $\lambda_1$ is zero, the given assembly structure is flexible, not rigid. In this case, we can locate the movable parts by examining the associated eigenvector $\Delta\mathbf{x}^*$. The underlying rationale is that if the structure contains movable parts, our method will find a way to "deform" the whole structure with zero energy. Figure 11 shows two examples, where $\Delta\mathbf{x}^*$ indicates the potential motion of the movable parts. In the case of flexible structures, our method can still estimate the worst-case rigidity of their largest rigid substructures by taking the smallest positive eigenvalue. Also, if we interpret the eigenvectors as external loads, the loads together will automatically keep the structure in equilibrium; see the inset on the right. This phenomenon occurs only if the assembly has no fixed points.



## 5. Optimizing Rigidity of Assemblies

Our worst-case rigidity analysis model can further guide us to modify a given assembly's geometry or topology to improve its rigidity. We present a gradient-based method for optimizing an assembly's geometry in Section 5.1 and a heuristic-based method for optimizing an assembly's topology in Section 5.2. Similar to Section 4, we illustrate our methods with 2D examples.

### 5.1. Continuous Optimization on Assembly Geometry

Given an assembly, we can parameterize and represent its geometry as vector $\mathbf{q}$. We take a 2D linkage model with vertex set $\{\mathbf{p}_i\}$ and edge set $\{\mathbf{e}_{ij}\}$, where $\mathbf{p}_i$ locates at the endpoints of the parts in the model. Essentially, the geometry can be parameterized by vertex positions $\{\mathbf{p}_i\}$, which can be aggregated to form $\mathbf{q}$. Note that $\{\mathbf{p}_i\}$ and $\{\mathbf{x}_i\}$ are different; $\{\mathbf{p}_i\}$ is a high-level geometric representation of the overall assembly, whereas $\{\mathbf{x}_i\}$ is a set of sampled FEM nodes, which is larger in quantity and harder to optimize.

Our goal is to improve the worst-case rigidity of assemblies with least modification on the geometry. To this end, we formulate the geometric optimization as a minimization problem:

$$\mathcal{E}(\mathbf{q}) = w_{\text{rigid}}\mathcal{E}_{\text{rigid}}(\mathbf{q}) + w_{\text{geom}}\mathcal{E}_{\text{geom}}(\mathbf{q}) \ , \tag{4}$$

where $\mathcal{E}_{\text{rigid}}$ is our worst-case rigidity measure Eq. (5) and $\mathcal{E}_{\text{geom}}$ measures the amount of modification on the assembly's geometry. To handle both rigid and flexible assemblies, we choose the worst-case rigidity measure $\mathcal{E}_{\text{rigid}}$ as the smallest non-zero eigenvalue, and model $\mathcal{E}_{\text{geom}}$ as the change in the assembly's total edge length:

$$\mathcal{E}_{\text{geom}}(\mathbf{q}) = \left[ \sum_{\mathbf{e}_{ij}} |\mathbf{p}_i - \mathbf{p}_j| - \sum_{e_{ij}} |\bar{\mathbf{p}}_i - \bar{\mathbf{p}}_j| \right]^2 , \tag{5}$$

where $\bar{\mathbf{p}}_i$ is the initial position of $\mathbf{p}_i$.

Our worst-case rigidity measure $\mathcal{E}_{\text{rigid}}$ is differentiable with respect to the input assembly's geometric parameter $\mathbf{q}$. This is because the computation of eigenvalues and eigenvectors is differentiable with respect to input matrix $\mathbf{K}+\mathbf{J}$. On the other hand, $\mathcal{E}_{\text{geom}}$ is also differentiable with respect to $\mathbf{q}$ according to Eq. (5). Hence, our objective function $\mathcal{E}$ is differentiable with respect to $\mathbf{q}$.

We compute the gradient of $\mathcal{E}$ with respect to $\mathbf{q}$ and employ gradient descent to iteratively update $\mathbf{q}$ for minimizing $\mathcal{E}$. In our implementation, we use the automatic differentiation package in PyTorch [PGM*19] for computing the gradients of the eigenvalues, which suffices for our use case since our objective for the gradient-based optimization only depends on the eigenvalues. Note that automatic differentiation of eigenvalues is numerically stable [Gil08] and the gradients may also be computed using an analytical method [vdAMM07]. In our experiment, we empirically set $w_{\text{rigid}}$ and $w_{\text{geom}}$ as $-1$ and 0.002, respectively, such that we can maximize the rigidity of the assembly while preserving its original shape. Figure 8 shows two examples, demonstrating how our method works. In Figure 8(a), the middle part of BRIDGE is "lifted" after the optimization to resist the bending force associated with the worst-case deformation of the initial structure. In Figure 8(b), the whole assembly of EXCAVATOR is not rigid. Yet, our method can still optimize its geometry to improve the rigidity of its substructure. After the optimization, the lever arm of the EXCAVATOR becomes shorter, such that the external torque exerted on the substructure (marked by the red triangle in Figure 8(b)) is reduced.

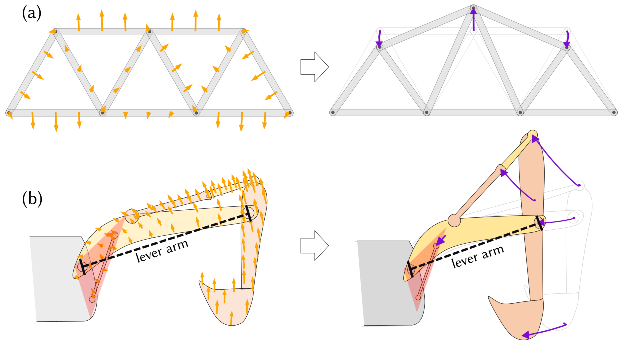


**Figure 8:** *Our method improves the worst-case rigidity of a rigid* BRIDGE FRAME *(a) and a flexible* CRAWLER EXCAVATOR *(b) by continuous optimization of their geometry. The orange arrows (left) indicate the worst-case external load configuration, whereas the purple arrows (right) indicate the path of vertices during the optimization with our method.*

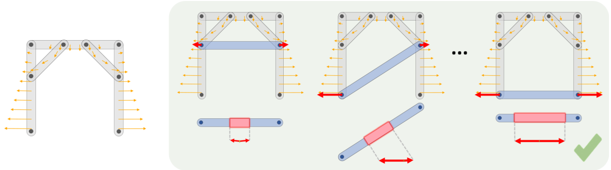


**Figure 9:** *Given an assembly (left), our method can recommend where to add a new part to reinforce the assembly's rigidity against the worst-case load (orange arrows). In detail, we examine each new part candidate and recommend the one that gives the most resistence to the worst-case load.*

### 5.2. Discrete Optimization on Assembly Topology

Besides optimizing the geometry, we can optimize an assembly's topology to improve its rigidity, e.g., by introducing new parts. A naïve way to form a new part is to exhaustively connect different pairs of nodes in existing assembly parts, estimate the change in rigidity, and choose the new part(s) that bring the most improvement. Clearly, this approach is not efficient, since it requires running the worst-case rigidity analysis on every new part candidate.

To allow controllability in the introduction of new parts, we work around the issue by developing an interactive solution, in which the user selects a pair of parts in the assembly and the new part will only be added to connect nodes between the selected parts. In

**Table 1:** *Statistics of our results. From left to right: name of the assembly model, corresponding figure, number of parts, number of joints, number of nodes, time to compute* **K** *(see Section 4), time to compute eigenvalues and eigenvectors, total computation time, whether the input assembly is rigid, and worst-case rigidity value (smallest eigenvalue) before and after rigidity optimization. Note that N/A in the last column indicates that we did not perform optimization on the assembly in the paper.*

| Name | Figure | # parts | # joints | # nodes ($N$) | Time for computing **K** (s) | Time for eigenanalysis (s) | Total time (s) | Rigid or not? | Rigidity value before opt. ($10^{-7}$) | Rigidity value after opt. ($10^{-7}$) |
|---|---|---|---|---|---|---|---|---|---|---|
| ROLLING CHASSIS | Fig. 1 | 153 | 190 | 4827 | 5.607 | 16.526 | 22.133 | Yes | 8.656 | 26.91 |
| BRIDGE FRAME | Fig. 8 | 11 | 7 | 88 | 0.062 | 0.023 | 0.085 | Yes | 63000 | 113513 |
| CRAWLER EXCAVATOR | Fig. 8 | 6 | 7 | 109 | 0.018 | 0.023 | 0.041 | No | 0 | 0 |
| - | Fig. 9 | 4 | 6 | 210 | 0.015 | 0.012 | 0.027 | Yes | 16.2 | 44.3 |
| - | Fig. 10 (a) | 8 | 6 | 96 | 0.018 | 0.005 | 0.023 | Yes | 1707 | N/A |
| - | Fig. 10 (b) | 16 | 18 | 200 | 0.032 | 0.014 | 0.046 | Yes | 3.74 | N/A |
| - | Fig. 10 (c) | 3 | 4 | 108 | 0.012 | 0.006 | 0.018 | Yes | 9.386 | N/A |
| - | Fig. 10 (d) | 12 | 6 | 468 | 0.591 | 0.210 | 0.801 | Yes | 137 | N/A |
| - | Fig. 10 (e) | 4 | 4 | 108 | 0.050 | 0.053 | 0.103 | Yes | 3.26 | N/A |
| - | Fig. 10 (f) | 35 | 44 | 956 | 0.445 | 0.413 | 0.858 | Yes | 7.189 | N/A |
| - | Fig. 10 (g) | 25 | 32 | 422 | 0.1187 | 0.3608 | 0.4795 | Yes | 2103.14 | N/A |
| - | Fig. 11 (top) | 6 | 6 | 72 | 0.013 | 0.004 | 0.017 | No | 0 | 1658 |
| - | Fig. 11 (bot) | 25 | 32 | 502 | 0.205 | 0.263 | 0.468 | No | 0 | 3918 |
| BOOMERANG | Fig. 14 (a) | 3 | 4 | 188 | 0.055 | 0.008 | 0.063 | Yes | 249.62 | 1097.59 |
| BUNNY | Fig. 14 (b) | 22 | 32 | 215 | 0.062 | 0.007 | 0.069 | Yes | 80.95 | 216.31 |
| DRYING RACK | Fig. 14 (c) | 10 | 10 | 638 | 0.131 | 0.616 | 0.747 | Yes | 5.11 | 64.6 |
| TECHNIC BIRD | Fig. 14 (d) | 84 | 109 | 1900 | 1.105 | 1.809 | 2.914 | Yes | 65.31 | 97.87 |
| RODS BOAT | Fig. 14 (e) | 88 | 30 | 968 | 2.508 | 2.235 | 4.743 | No | 0 | 2595 |

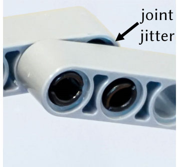
detail, our method first forms a set of new part candidates, each connecting a unique pair of nodes on the two selected parts. For each pair of nodes, we compute the difference between the worst-case node displacement vectors for each node, and project these vectors onto the line that connects the two nodes. The length of the projected vector reveals the new part's ability to resist the deformation of the worst-case load, so we recommend the node pair with the longest projected vectors to the user; see, e.g., Figure 9. Experimentally, enumerating all node pairs may yield as many as 100 candidates, but it takes less than 0.01s to compute. Interested readers may further adopt other heuristics for speed-up.

## 6. Results and Experiments

We implemented our method in Python 3.8 on a computer with a dual-core 2.5GHz Intel i5-7200U CPU and 8GB RAM. We used NumPy [HMvdW*20] and SciPy [VGO*20] for numerical computing, e.g., computing the null space and eigenvectors. We conducted experiments to evaluate our rigidity analysis and optimization in various aspects; see Section 6.1. We showed the usefulness of our approach by designing and fabricating several structurally rigid assemblies for use in different applications; see Section 6.2.

Table 1 summarizes the statistics of all results presented in the paper, in which we report the complexity of each input assembly (i.e., the number of parts, joints, and nodes), the computing time (next three columns), whether the input assembly is rigid or not, and the rigidity value before and after our optimization (last three columns). From Table 1, we can see that our approach can well handle assemblies with over 100 parts and nearly 200 mechanical joints; see also Figure 1. The total run time for most assemblies is less than one second, making it possible to provide nearly interactive feedback to users. As expected, the run time mainly increases with the number of nodes, and the computation for eigenvalues and eigenvectors dominates the overall run time. In practice, if the user only requires the result for worst-case rigidity (the smallest eigenvalue and eigenvector), we can accelarate the computation by applying methods tailored to sparse matrix such as [LSY98]. Also, we only use a laptop computer and do not optimize our computation in our experiments, e.g., harnessing parallel computation on the GPU to speed up the matrix construction and eigenanalysis, which will further accelerate the computation.

**Measuring stiffness coefficients for joints.** We conduct physical experiments to measure the stiffness coefficients to form matrix **S** (see Section 4.1). We introduce our method of measuring the coefficients of the joint between the LEGO Technic beam and LEGO Technic pin, as shown in the inset figure. In this case, relative rotation about the pin axis is allowed, and five other DOFs are constrained, including three for relative translation that cause shearing of the pin and two for relative rotation that cause bending of the pin. Among them, the DOF for translation along the pin axis is restricted by the blocking of the material of the parts, so we model it by setting the stiffness coefficients to a very large value ($10^8$ N/m in our experiments). Then, we conduct physical experiments to measure the coefficients of two DOFs for pin shearing and two DOFs for pin bending. Since both the LEGO Technic beam and LEGO Technic pin are symmetric about their rotation axis, we use the same coefficient value for two DOFs of pin bending, and also the same value for two DOFs of pin shearing. In our experiment, we build a simple assembly composed of two LEGO Technic beams connected by two pins, and fix one beam, and hang a 200g standard weight on the other beam. Then, we place the assembly under different poses, and for each pose, we measure the deviation of the endpoint of the beam compared to its rest position. After we collect the data about the weight and the corresponding deviation, we use a simple search procedure to search for the correct



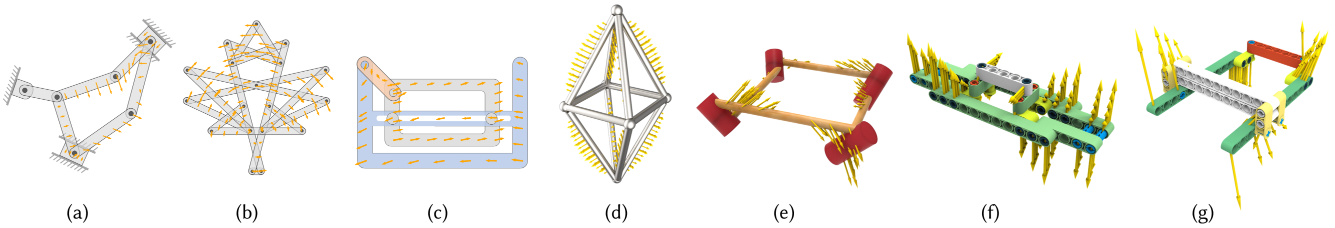


**Figure 10:** *Our method can successfully identify all the above models as rigid, fully automatically. (a), (b) and (c) are 2D mechanical assemblies made up of hinges and sliding joints; (d) is a body-and-joint framework; (e) is a spatial linkage model with hinge joints (none of the rotation axes are parallel); and (f) & (g) are mechanical assemblies (LEGO Technic models) with 32 and 30 joints, respectively. The orange (or yellow) arrows reveal the worst-case load configurations.*

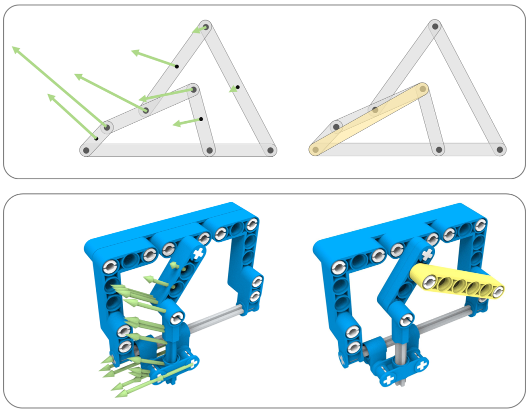


**Figure 11:** *Our method can detect potential (infinitesimal) parts motions, as demonstrated on the six-bar linkage shown on top and the LEGO Technic assembly (28 bricks) shown at the bottom. The green arrows indicate the detected parts' movement directions. Based on the motions of the detected parts, users can introduce additional parts (in yellow) to make the assembly rigid.*

coefficients by running a series of forward simulations over a range of stiffness values and select the one that produces the closest deviation effect with the physical experiment. The coefficient values measured by our method are 65625 N/m for pin shearing and 14992 N/m for pin bending. We use the values for stiffness coefficients of the joint in experiments shown in Figure 10 (e-g), Figure 11(bottom), ROLLING CHASSIS (Figure 1), BOOMERANG, BUNNY and TECHNIC BIRD (Figure 14(a-c, e)). For other experiments, we use 10000 N/m as the default value.

### 6.1. Validations

Next, we evaluate and validate our method in terms of its capability of performing the four tasks introduced in Section 3.

**Evaluation of rigidity test.** First, we employ our method to test the rigidity of a wide variety of assemblies. Figure 10 shows the models employed in this evaluation. They cover different kinds of mechanical objects and also different types of joints, e.g., hinges, sliding, and ball joints. All assemblies (except the one in Figure 10(a)) do not have fixed nodes. Some of them are rather complex, making it a challenging task to decide if the assembly is rigid purely by visual observation.

We also apply our method to compute the worst-case load configuration, as visualized by the orange arrows. The test results show that our method can successfully identify all assemblies as rigid. An important strength of our method is that it is a unified computational framework that can test the rigidity of a wide variety of assemblies. These assemblies can have very different forms, and they can be composed of different kinds of parts and joints, including complex mechanical assemblies that have rarely been explored before, e.g., spatial linkages and LEGO Technic assemblies.

**Evaluation of potential part motions.** For flexible assemblies, our method can find potential part motions from just the input assembly, without requiring other forms of input such as the prescribed external loads or power sources. Figure 11 shows the potential part motions identified by our method on a six-bar linkage model and a LEGO Technic assembly. The detected part motions are all correct and the run times for both models are less than 0.4 seconds. Based on the visualization of the detected motions, users can add more parts to make the assembly rigid. Specifically, users just need to mark a pair of connectable points among all parts (while considering the aesthetics and fabricability), such that the new part(s) can prevent the detected part motion; see Figure 11 (right) for the results.

**Verification of our worst-case rigidity value.** Next, we conducted experiments to verify the worst-case rigidity measure, i.e., $\lambda_1$ (see Section 4.2) computed by our method. Starting from the flexible four-bar linkage assembly shown in Figure 12(a1), we incrementally add a part or change the endpoint position of an existing part to improve the rigidity of the assembly; see Figure 12(a2-a4). For each linkage assembly, we apply our method to compute the worst-case rigidity value and report it in the figure. The result shows that our method is able to analyze both flexible (Figure 12(a1)) and rigid assemblies (Figure 12(a2-a4)). Moreover, our computed worst-case rigidity value is consistent with human intuition. The assembly in Figure 12(a2) is rigid since a single short bar is added onto the input four-bar linkage. After adding one more short bar to Figure 12(a2), the assembly (Figure 12(a3)) becomes structurally more rigid, corresponding to a larger worst-case rigidity value. Furthermore, elongating the two new bars can make the overall assembly (Figure 12(a4)) even more structurally rigid, due to the two large three-bar triangles formed in the assembly.

Also, we apply our method to the two collinear bars shown in Figure 12(b). Here, we actually know that increasing the overlap

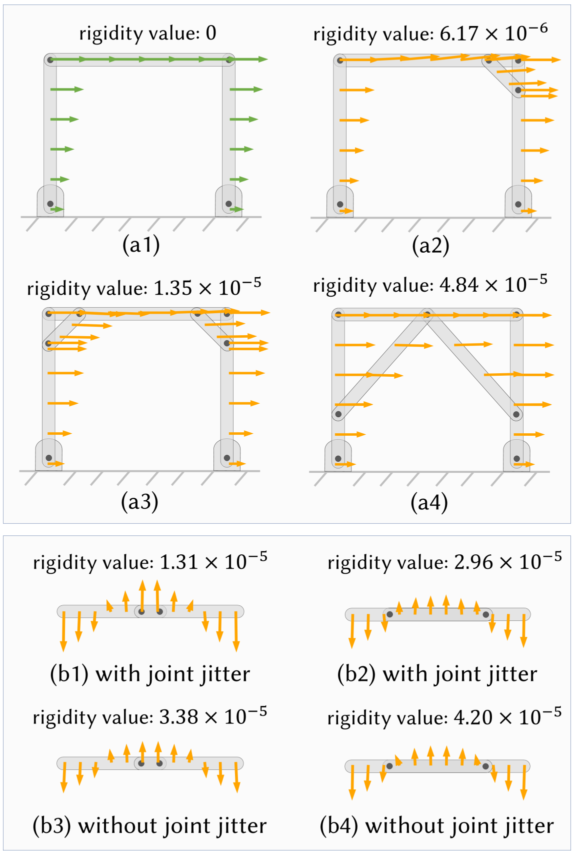

rigidity value: 0
rigidity value: 6.17 × 10⁻⁶
(a1)
(a2)
rigidity value: 1.35 × 10⁻⁵
rigidity value: 4.84 × 10⁻⁵
(a3)
(a4)
rigidity value: 1.31 × 10⁻⁵
rigidity value: 2.96 × 10⁻⁵
(b1) with joint jitter
(b2) with joint jitter
rigidity value: 3.38 × 10⁻⁵
rigidity value: 4.20 × 10⁻⁵
(b3) without joint jitter
(b4) without joint jitter

**Figure 12:** *Our method quantifies the worst-case rigidity value for each assembly in (a) and (b), and shows the corresponding force (motion) in yellow (green) when the assembly is rigid (flexible). From (a1) to (a3), two short bars are incrementally added to a three-bar linkage and their lengths and positions are changed in (a4). In (b), we show four different collinear bars with different overlaps and with or without they have joint jitter effect. The predicted motion and the relative magnitudes of rigidity values in groups (a) and (b) are consistent with human intuition.*

between the two bars will increase the rigidity of the assembly. From Figure 12 (b1&b2), we can see that our method is able to produce results that are consistent with our expectations. It is worth noticing that the phenomenon of increasing the overlap is observed when we apply our optimization method to larger-scale examples, e.g., BUNNY in Figure 14. In practice, we also find joint jitters common in manufacturing methods such as 3D printing and laser cutting, and usually have a negative impact on rigidity. Comparing Figure 12 (b1) with (b3) and (b2) with (b4), we find that our analysis results are aligned with the expected effects of joint jittering.

**Evaluation of our geometry optimization.** Lastly, we evaluate our geometry optimization by conducting a physical experiment on two assemblies and analyzing their rigidity before and after the optimization. The first one is a three-bar linkage where the length of its bottom edge is fixed (see Figure 13(a)), whereas the second one is a five-bar linkage where the lengths of three bars on the rectangular boundary is fixed (see Figure 13(e)). For each assembly, we perform our gradient-based optimization (see again Section 5.1) to automatically adjust the position of the connecting points in the assembly structure, i.e., vertices.

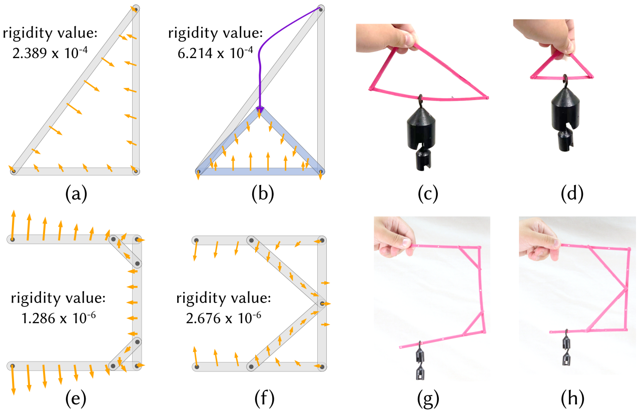

rigidity value: 2.389 x 10⁻⁴
rigidity value: 6.214 x 10⁻⁴
(a)
(b)
(c)
(d)
rigidity value: 1.286 x 10⁻⁶
rigidity value: 2.676 x 10⁻⁶
(e)
(f)
(g)
(h)

**Figure 13:** *Verification of our geometry optimization. Given input assemblies (a&e), our optimization automatically adjusts the position of endpoints of parts (b&f) to improve the structural rigidity. We apply loads on the corresponding physical assemblies that are before (c&g) and after (d&h) our optimization, showing that the optimized assemblies have less deformation under the same amount of external load.*

Figure 13 shows the optimized assemblies and the physical validation results. First, we can see from Figure 13(b) that the initial tall triangle becomes an isosceles triangle after our optimization; on the other hand, comparing Figure 13(e&f), we can see that the two supporting bars of the five-bar linkage become longer and join together in the middle of the vertical bar. We fabricate the before- and after-optimization versions of both assembly models using 3D printing and apply the same amount of load on each of them. Comparing Figure 13 (c) with (d) and (g) with (h), we can see that the optimized assembly models can better resist the (same) external load with less deformation. Please also refer to Figure 8 for another two results on geometry optimization.

## 6.2. Applications

To demonstrate the usefulness of our rigidity analysis and optimization approach, we employ it to design and fabricate four different kinds of structurally rigid assemblies; see Figure 14.

**Linkage structure.** Figure 14(a&b) show the BOOMERANG and BUNNY models, which consist of both 3D-printed parts and LEGO Technic pins. Since the forces exerted on them are often hard to predict during the usage, it is important to optimize their structures for their worst-case rigidity. We employ our gradient-based optimization method to improve their rigidity, and it took 125 and 272 seconds for the two models, respectively.

For the BOOMERANG shown in Figure 14(a), our method adjusts the position of supporting bars, such that the boundary is less deformed under the same 800g weights. We measure the vertical position changes of the lowest point of the BOOMERANG, and observe a 13mm difference between two designs.

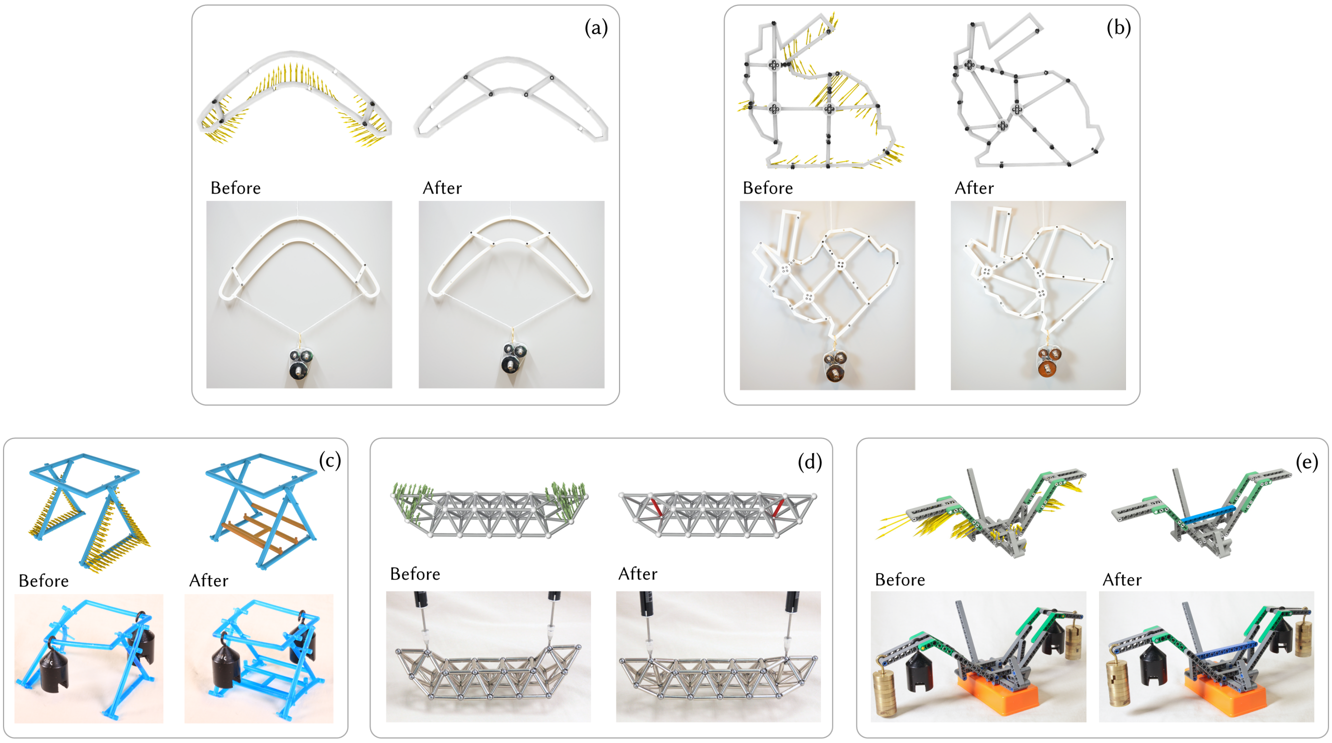


**Figure 14:** *Five different kinds of structurally rigid assemblies designed with our approach. We apply continuous optimization method as presented in Section 5.1 to improve the rigidity for linkage models* BOOMERANG *(a), and* BUNNY *(b). We also apply discrete optimization to introduce new part(s) to improve rigidity for reconfigurable* DRYING RACK *(c), rod-and-ball-joint structure* RODS BOAT *(d), and LEGO* TECHNIC BIRD *(e). For each result, we show the assembly before (left) and after (right) the improvement on structural rigidity and also show the worst-case loads (potential motions) with orange (green) arrows. We show physical experiments on fabricated prototypes to validate the improvement in rigidity.*

For the BUNNY in Figure 14(b), we first manually add internal supporting structure to prevent the boundary from deforming, as shown in Figure 14(b) left. Yet, the manually-added structure may not be perfect, so we apply our method to adjust the position of the internal nodes, making the BUNNY more rigid without introducing new parts and joints. Similarly, we measure the vertical position changes of the lowest point when weights are hung, and observe a 7mm difference between the initial and optimized designs.

**Reconfigurable assembly.** Our approach can help users to design reconfigurable assemblies that are structurally rigid for user-desired assembly forms. Figure 14(c) shows a DRYING RACK designed by our approach. Our system can predict and visualize the worst-case deformation of the rack in its unfolded form and suggest additional parts to improve structural rigidity. The physical experiment shows the input model fails under the torsional deformation as predicted by our rigidity analysis. With the new reinforced parts, the assembly resists from the same external force.

**Rod-and-ball-joint structure.** As a natural extension, our system can also support the design of assemblable rod-and-ball-joint structures, which may benefit the prototyping process of the industrial design. Figure 14(d) shows a RODS BOAT structure designed by our approach. Our system successfully detects the two flexible regions around the two ends of the initial assembly and visualizes the corresponding motion with green arrows. By adding two new rods (in red) to the structure, the assembly resists a pulling force at the two ends, under which the initial one easily fails.

**LEGO Technic assembly.** We demonstrate our algorithm for predicting and enhancing the structural rigidity of static 3D LEGO Technic assemblies. Figure 14(e) shows a TECHNIC BIRD assembly made of 84 LEGO Technic bricks, where our system detects the vulnerability at the wings of the initial assembly. The worst-case external load configuration shown as yellow arrows can guide the user to add additional beams to reinforce the assembly, among which we choose a long horizontal beam that connects the two wings. We built the physical prototype with LEGO Technic parts and applied the same external force to the input and the optimized prototypes. The result shows that the reinforced assembly on the right undergoes less deformation than the initial one.

Figure 1 shows another LEGO Technic example, a ROLLING CHASSIS composed of beams in different orientations. Our method predicts a twisting deformation under the worst-case external loads, and guides users to add new beams to reinforce the assembly. We perform physical experiments to validate the worst-case load: we first lift the two opposite corners of the assembly, then hang up the two weights (200g for each) on the other two corners. The resulting reinforced assembly also exhibits less deformation.

## 7. Conclusion

In this paper, we present a new approach with a unified framework for computing, analyzing, and optimizing the structural rigidity of assemblies with mechanical joints. Our contributions in this work are threefold. First, we account for the fact that real materials are generally not perfectly rigid, so we consider the impact of part deformation and joint jitters in rigidity analysis by modeling parts as deformable objects of linear elasticity and joints as soft constraints. Then, we represent both parts and joints as stiffness matrices, such that the problem can be formulated as an optimization problem that can be solved by an eigenanalysis, such that the eigenvalue gives the worst-case rigidity and the eigenvector gives the worst-case deformation. Lastly, we also develop methods to make use of our rigidity measure to guide the optimization of the geometry and topology of the assembly for greater rigidity. We validate the worst-case rigidity measure through physical experiments on several prototypes and demonstrate the effectiveness of our optimization approach to aid the design and creation of structurally rigid assemblies for a rich variety of forms.

**Limitations and Future Work.** This work has several limitations that open up interesting new directions for future research. First, our rigidity analysis and optimization approach can only handle assemblies with mechanical joints. Extending it to support other types of joints, such as woodworking joints, would be interesting future work that may enable us to efficiently analyze the rigidity of wooden structures related to furniture and architecture. Second, we model the imperfection of mechanical joints by allowing deformation at the contact surfaces of the parts and measuring the joint stiffness using a diagonal matrix. Simulating more accurately the actual physics of the mechanical joints would lead to a more precise worst-case rigidity analysis. Third, our optimization-based design approach focuses on structural rigidity. Taking other design goals, e.g., desired motion and/or functionality, into consideration would be very helpful to enable the design and creation of more practical assemblies. Lastly, our current continuous optimization on assembly geometry can only handle simple linkage-based assemblies. Generalizing it to handle complex and general 3D assemblies is another interesting and challenging future direction; in particular, doing so can help us to better optimize continuous geometry, e.g., assemblies with 3D-printed parts.

## Acknowledgments

This work was supported by the Research Grants Council of the Hong Kong Special Administrative Region, China [Project No.: CUHK 14201921] and the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 715767 – MATERIALIZABLE). We thank the anonymous reviewers for their insightful feedback; Christian Hafner for proofreading and discussions; Ziqi Wang, Haisen Zhao, and Martin Hafskjold Thoresen for the helpful discussions; and the Miba Machine Shop at IST Austria for 3D printing the BUNNY and BOOMERANG models.

## References

[BBJP12] BÄCHER M., BICKEL B., JAMES D. L., PFISTER H.: Fabricating articulated characters from skinned meshes. *ACM Trans. on Graph. (SIGGRAPH) 31*, 4 (2012), 47:1–47:9. doi:10.1145/2185520.2185543.

[CCA*12] CALÌ J., CALIAN D. A., AMATI C., KLEINBERGER R., STEED A., KAUTZ J., WEYRICH T.: 3D-printing of non-assembly, articulated models. *ACM Trans. on Graph. (SIGGRAPH Asia) 31*, 6 (2012), 130:1–130:8. doi:10.1145/2366145.2366149.

[CTN*13] COROS S., THOMASZEWSKI B., NORIS G., SUEDA S., FORBERG M., SUMNER R. W., MATUSIK W., BICKEL B.: Computational design of mechanical characters. *ACM Trans. on Graph. (SIGGRAPH) 32*, 4 (2013), 83:1–83:12. doi:10.1145/2461912.2461953.

[Gil08] GILES M.: *An extended collection of matrix derivative results for forward and reverse mode algorithmic differentiation*. Tech. rep., Oxford University Computing Laboratory, 2008.

[GJG16] GARG A., JACOBSON A., GRINSPUN E.: Computational design of reconfigurables. *ACM Trans. on Graph. (SIGGRAPH) 35*, 4 (2016), 90:1–90:14. doi:10.1145/2897824.2925900.

[HJ12] HORN R. A., JOHNSON C. R.: *Matrix analysis*. Cambridge University Press, 2012. doi:10.1017/CBO9780511810817.

[HLSS*12] HALLER K., LEE-ST.JOHN A., SITHARAM M., STREINU I., WHITE N.: Body-and-cad geometric constraint systems. *Computational Geometry 45*, 8 (2012), 385–405. doi:10.1145/1529282.1529530.

[HMvdW*20] HARRIS C. R., MILLMAN K. J., VAN DER WALT S. J., GOMMERS R., VIRTANEN P., COURNAPEAU D., WIESER E., TAYLOR J., BERG S., SMITH N. J., KERN R., PICUS M., HOYER S., VAN KERKWIJK M. H., BRETT M., HALDANE A., DEL R'IO J. F., WIEBE M., PETERSON P., G'ERARD-MARCHANT P., SHEPPARD K., REDDY T., WECKESSER W., ABBASI H., GOHLKE C., OLIPHANT T. E.: Array programming with NumPy. *Nature 585*, 7825 (Sept. 2020), 357–362. doi:10.1038/s41586-020-2649-2.

[HZG*18] HU Y., ZHOU Q., GAO X., JACOBSON A., ZORIN D., PANOZZO D.: Tetrahedral meshing in the wild. *ACM Trans. Graph. 37*, 4 (July 2018), 60:1–60:14. doi:10.1145/3197517.3201353.

[JJ10] JACKSON B., JORDÁN T.: The generic rank of body-bar-and-hinge frameworks. *European Journal of Combinatorics 31*, 2 (2010), 574–588. doi:10.1016/j.ejc.2009.03.030.

[KLY*14] KOO B., LI W., YAO J., AGRAWALA M., MITRA N. J.: Creating works-like prototypes of mechanical objects. *ACM Trans. on Graph. (SIGGRAPH Asia) 33*, 6 (2014), 217:1–217:9. doi:10.1145/2661229.2661289.

[Lam70] LAMAN G.: On graphs and rigidity of plane skeletal structures. *Journal of Engineering Mathematics 4*, 4 (1970), 331–340. doi:10.1007/BF01534980.

[LHAZ15] LI H., HU R., ALHASHIM I., ZHANG H.: Foldabilizing furniture. *ACM Trans. on Graph. (SIGGRAPH) 34*, 4 (2015), 90:1–90:12. doi:10.1145/2766912.

[LMaH*18] LI S., MAHDAVI-AMIRI A., HU R., LIU H., ZOU C., KAICK O. V., LIU X., HUANG H., ZHANG H.: Construction and fabrication of reversible shape transforms. *ACM Trans. on Graph. (SIGGRAPH Asia) 37*, 6 (2018), 190:1–190:14. doi:10.1145/3272127.3275061.

[LSD*16] LANGLOIS T., SHAMIR A., DROR D., MATUSIK W., LEVIN D. I.: Stochastic structural analysis for context-aware design and fabrication. *ACM Trans. on Graph. (SIGGRAPH Asia) 35*, 6 (2016), 226:1–226:13. doi:10.1145/2980179.2982436.

[LSY98] LEHOUCQ R. B., SORENSEN D. C., YANG C.: *ARPACK users' guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods*. SIAM, 1998. doi:10.1137/1.9780898719628.

[LYH*15] LUO S.-J., YUE Y., HUANG C.-K., CHUNG Y.-H., IMAI S., NISHITA T., CHEN B.-Y.: Legolization: Optimizing LEGO designs. *ACM Trans. on Graph. (SIGGRAPH Asia) 34*, 6 (2015), 222:1–222:12. doi:10.1145/2816795.2818091.

[MEKM17] Miyamoto E., Endo Y., Kanamori Y., Mitani J.: Semi-automatic conversion of 3D shape into flat-foldable polygonal model. *Comp. Graph. Forum (Pacific Graphics) 36*, 7 (2017), 41–50. doi:10.1111/cgf.13270.

[MHR*16] Musialski P., Hafner C., Rist F., Birsak M., Wimmer M., Kobbelt L.: Non-linear Shape Optimization using Local Subspace Projections. *ACM Trans. on Graph. (SIGGRAPH) 35*, 4 (2016), 87:1–87:13. doi:10.1145/2897824.2925886.

[MTN*15] Megaro V., Thomaszewski B., Nitti M., Hilliges O., Gross M., Coros S.: Interactive design of 3d-printable robotic creatures. *ACM Trans. on Graph. (SIGGRAPH Asia) 34*, 6 (2015), 216:1–216:9. doi:10.1145/2816795.2818137.

[MYY*10] Mitra N. J., Yang Y.-L., Yan D.-M., Li W., Agrawala M.: Illustrating how mechanical assemblies work. *ACM Trans. on Graph. (SIGGRAPH) 29*, 4 (2010), 58:1–58:11. doi:10.1145/1778765.1778795.

[NBA19] Nishida G., Bousseau A., Aliaga D. G.: Multi-pose interactive linkage design. *Comp. Graph. Forum (Eurographics) 38*, 2 (2019), 277–289. doi:10.1111/cgf.13637.

[PGM*19] Paszke A., Gross S., Massa F., Lerer A., Bradbury J., Chanan G., Killeen T., Lin Z., Gimelshein N., Antiga L., Desmaison A., Kopf A., Yang E., DeVito Z., Raison M., Tejani A., Chilamkurthy S., Steiner B., Fang L., Bai J., Chintala S.: Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035. URL: https://proceedings.neurips.cc/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf.

[PTV*17] Pietroni N., Tarini M., Vaxman A., Panozzo D., Cignoni P.: Position-based tensegrity design. *ACM Trans. on Graph. (SIGGRAPH Asia) 36*, 6 (2017), 172:1–172:14. doi:10.1145/3130800.3130809.

[SB12] Sifakis E., Barbic J.: Fem simulation of 3D deformable solids: A practitioner's guide to theory, discretization and model reduction. In *ACM SIGGRAPH Courses*. 2012, pp. 20:1–20:50. doi:10.1145/2343483.2343501.

[SFCO12] Song P., Fu C.-W., Cohen-Or D.: Recursive interlocking puzzles. *ACM Trans. on Graph. (SIGGRAPH Asia) 31*, 6 (2012), 128:1–128:10. doi:10.1145/2366145.2366147.

[SZB18] Schumacher C., Zehnder J., Bächer M.: Set-In-Stone: Worst-case optimization of structures weak in tension. *ACM Trans. on Graph. (SIGGRAPH Asia) 37*, 6 (2018), 252:1–252:13. doi:10.1145/3272127.3275085.

[Tay84] Tay T.-S.: Rigidity of multi-graphs. i. linking rigid bodies in n-space. *Journal of Combinatorial Theory 36*, 1 (1984), 95–112. doi:10.1016/0095-8956(84)90016-9.

[Tay89] Tay T.-S.: Linking (n- 2)-dimensional panels in n-space ii: (n-2, 2)-frameworks and body and hinge structures. *Graphs and Combinatorics 5* (1989), 245–273. doi:10.1007/BF01788678.

[TCG*14] Thomaszewski B., Coros S., Gauge D., Megaro V., Grinspun E., Gross M.: Computational design of linkage-based characters. *ACM Trans. on Graph. (SIGGRAPH) 33*, 4 (2014), 64:1–64:9. doi:10.1145/2601097.2601143.

[TD02] Thorpe M. F., Duxbury P. M.: *Rigidity Theory and Applications*. Kluwer Academic Publishers, 2002. doi:10.1007/b115749.

[UMK17] Ulu E., Mccann J., Kara L. B.: Lightweight structure design under force location uncertainty. *ACM Trans. on Graph. (SIGGRAPH) 36*, 4 (2017), 158:1–158:13. doi:10.1145/3072959.3073626.

[UTZ16] Ureta F. G., Tymms C., Zorin D.: Interactive modeling of mechanical objects. *Comp. Graph. Forum (SGP) 35*, 5 (2016), 145–155. doi:10.1111/cgf.12971.

[vdAMM07] van der Aa N., Morsche H., Mattheij R.: Computation of eigenvalue and eigenvector derivatives for a general complex-valued eigensystem. *ELA. The Electronic Journal of Linear Algebra [electronic only] 16* (09 2007), 300–314. doi:10.13001/1081-3810.1203.

[VGO*20] Virtanen P., Gommers R., Oliphant T. E., Haberland M., Reddy T., Cournapeau D., Burovski E., Peterson P., Weckesser W., Bright J., van der Walt S. J., Brett M., Wilson J., Millman K. J., Mayorov N., Nelson A. R. J., Jones E., Kern R., Larson E., Carey C. J., Polat İ., Feng Y., Moore E. W., VanderPlas J., Laxalde D., Perktold J., Cimrman R., Henriksen I., Quintero E. A., Harris C. R., Archibald A. M., Ribeiro A. H., Pedregosa F., van Mulbregt P., SciPy 1.0 Contributors: SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods 17* (2020), 261–272. doi:10.1038/s41592-019-0686-2.

[WOD09] Whiting E., Ochsendorf J., Durand F.: Procedural modeling of structurally-sound masonry buildings. *ACM Trans. on Graph. (SIGGRAPH Asia) 28*, 5 (2009), 112:1–112:9. doi:10.1145/1618452.1618458.

[WSIP19] Wang Z., Song P., Isvoranu F., Pauly M.: Design and structural optimization of topological interlocking assemblies. *ACM Trans. on Graph. (SIGGRAPH Asia) 38*, 6 (2019), 193:1–193:13. doi:10.1145/3355089.3356489.

[WSP21a] Wang Z., Song P., Pauly M.: MOCCA: Modeling and optimizing cone-joints for complex assemblies. *ACM Trans. on Graph. (SIGGRAPH) 40*, 4 (2021), 181:1–181:14. doi:10.1145/3450626.3459680.

[WSP21b] Wang Z., Song P., Pauly M.: State of the art on computational design of assemblies with rigid parts. *Comp. Graph. Forum (Eurographics) 40*, 2 (2021), 633–657. doi:10.1111/cgf.142660.

[WSW*12] Whiting E., Shin H., Wang R., Ochsendorf J., Durand F.: Structural optimization of 3d masonry buildings. *ACM Trans. on Graph. (SIGGRAPH Asia) 31*, 6 (2012), 159:1–159:11. doi:10.1145/2366145.2366178.

[YCXW17] Yao M., Chen Z., Xu W., Wang H.: Modeling, evaluation and optimization of interlocking shell pieces. *Comp. Graph. Forum (Pacific Graphics) 36*, 7 (2017), 1–13. doi:10.1111/cgf.13267.

[YKGA17] Yao J., Kaufman D. M., Gingold Y., Agrawala M.: Interactive design and stability analysis of decorative joinery for furniture. *ACM Trans. on Graph. 36*, 2 (2017), 20:1–20:16. doi:10.1145/3054740.

[YZC18] Yuan Y., Zheng C., Coros S.: Computational design of transformables. *Comp. Graph. Forum (SCA) 37*, 8 (2018), 103–113. doi:10.1111/cgf.13516.

[ZCT16] Zehnder J., Coros S., Thomaszewski B.: Designing Structurally-Sound Ornamental Curve Networks. *ACM Trans. on Graph. (SIGGRAPH) 35*, 4 (2016), 99:1–99:10. doi:10.1145/2897824.2925888.

[ZPZ13] Zhou Q., Panetta J., Zorin D.: Worst-case structural analysis. *ACM Trans. on Graph. (SIGGRAPH) 32*, 4 (2013), 137:1–137:11. doi:10.1145/2461912.2461967.

[ZSC16] Zheng C., Sun T., Chen X.: Deployable 3D linkages with collision avoidance. In *Proc. Eurographics/ ACM SIGGRAPH Symposium on Computer Animation* (2016), pp. 179–188. doi:10.2312/sca.20161235.

[ZSMS14] Zhou Y., Sueda S., Matusik W., Shamir A.: Boxelization: Folding 3D objects into boxes. *ACM Trans. on Graph. (SIGGRAPH) 33*, 4 (2014), 71:1–71:8. doi:10.1145/2601097.2601173.

[ZWC*16] Zhang R., Wang S., Chen X., Ding C., Jiang L., Zhou J., Liu L.: Designing planar deployable objects via scissor structures. *IEEE Trans. Vis. & Comp. Graphics 22*, 2 (2016), 1051–1062. doi:10.1109/TVCG.2015.2430322.