This supplementary material is composed of four parts. The first part presents the approach to solve the linear system of inequalities for global interlocking test (Paper Section 3.1). The second part formulates the optimization on the 3D surface tessellation (Paper Section 5). The third part describes the optimization to find the vertices of the target feasible section (Paper Section 6.1). The last part presents the gradient-based approach to solve the structural optimization on the topological interlocking (TI) assemblies (Paper Section 6.2).

1 Global Interlocking Test

An assembly is global interlocking if the following linear system of inequalities (paper Eq. 5) does not have any non-zero solution:

\[
\begin{bmatrix}
B_{1,1} & B_{1,2} & \cdots & B_{1,n} \\
\vdots & \vdots & \ddots & \vdots \\
B_{m,1} & B_{m,2} & \cdots & B_{m,n}
\end{bmatrix}
\begin{bmatrix}
Y_1 \\
\vdots \\
Y_n
\end{bmatrix}
\geq
\begin{bmatrix}
0 \\
\vdots \\
0
\end{bmatrix}
\text{s.t. } \begin{bmatrix}
Y_1 \\
\vdots \\
Y_n
\end{bmatrix} \neq 0, (1)
\]

where

\[
B_{i,j} = \begin{bmatrix}
-n^T_{i} - (r^c_{i} \times n_i)T \\ 
\vdots \\ 
-n^T_{i} - (r^c_{i} \times n_i)T
\end{bmatrix}
\]

\[
B_{i,k} = 0, \quad \text{for } k \neq i, k \neq j
\]

\[
i, j \text{ satisfies: } C_3 = P_i \cap P_j, \quad i < j
\]

with boundary conditions:

\[
Y_n = 0, \quad \text{part } P_n \text{ is the boundary frame}
\]

We solve this linear system by formulating it as a linear program [Wang et al. 2018]:

\[
\max \sum_{i=1}^{m} t_i
\]

s.t.

\[
\begin{bmatrix}
B_{1,1} & B_{1,2} & \cdots & B_{1,n} \\
\vdots & \vdots & \ddots & \vdots \\
B_{m,1} & B_{m,2} & \cdots & B_{m,n}
\end{bmatrix}
\begin{bmatrix}
Y_1 \\
\vdots \\
Y_n
\end{bmatrix}
\geq
\begin{bmatrix}
t_1 \\
\vdots \\
t_m
\end{bmatrix}
\]

\[
Y_n = 0
\]

We solve above linear program to find the optimal solution. If the optimal solution of Eq. 2 is non-zero, the assembly is not globally interlocking. Otherwise, the assembly is globally interlocking except for one special case in which the optimal solution of Eq. 2 is zero yet \(B_n \cdot Y = 0\) actually has non-zero solutions. The inset shows such an example. In this assembly, each component part can move vertically (i.e., not interlocking) without any collision among the parts (i.e., the optimal solution of Eq. 2 is zero) since the translation vector of each component part is always perpendicular to the contact normals. To identify this special case, we compute the rank of first \(n-1\) columns of \(B_n\) after solving the linear program. The equation \(B_n \cdot Y = 0\), where \(Y = 0\) has non-zero solutions if and only if rank of the first \(n-1\) columns of \(B_n\) is smaller than \(n-1\) according to the theorem of homogeneous linear equations.

2 Optimization of 3D Surface Tessellation

Taking a surface tessellation \(T\) using the conformal maps as a good initialization, we further optimize its vertices while fixing their connectivity for desirable properties described in the paper (i.e., planarity and regularity of each face, and small dihedral angles between adjacent faces). We formulate this optimization as an energy minimization problem:

\[
E = \omega_1 E_{surf} + \omega_2 E_{bound} + \omega_3 E_{planar} + \omega_4 E_{regular} + \omega_5 E_{coplanar}
\]

where \(E_{surf}\) aims to make the tessellation \(T\) resemble the reference surface \(S\); \(E_{bound}\) aims to preserve boundary of the tessellation \(T\); \(E_{planar}\) and \(E_{regular}\) aim to make faces in \(T\) planar and regular, respectively; and \(E_{coplanar}\) aims to ensure sufficient dihedral angles among adjacent faces in the tessellation \(T\).

In particular, the surface term \(E_{surf}\) is a summation of the distance from each vertex \(v_i\) of the tessellation \(T\) to the surface \(S\):

\[
E_{surf} = \sum_{v_i \in T} d^2(v_i, S)
\]

The boundary term \(E_{bound}\) evaluates the distance from each boundary vertex \(v_{b_i}\) of \(T\) to its initial position \(v_{b_i}^{init}\):

\[
E_{bound} = \sum ||v_{b_i} - v_{b_i}^{init}||^2
\]

The \(E_{planar}\) and \(E_{regular}\) terms for each face and \(E_{coplanar}\) term for each pair of adjacent faces in the tessellation \(T\) are computed following the projection-based approach in [Bouaziz et al. 2012].

Figure 1 shows an example surface tessellation \(T\) before and after our optimization, as well as the corresponding TI assemblies. Thanks to the desirable properties of the optimized tessellation, the resulting TI assembly have more coherent arrangement of the blocks; compare Figure 1-\(d&e\).

Figure 1: (a) Input reference surface and 2D tessellation. (b) Initial surface tessellation using conformal maps. (c) Optimized surface tessellation. (d&e) Corresponding TI assemblies with the same \(\alpha\).
3 Compute Target Force Directions

Given a feasible section $S(P)$ of an assembly $P$, the radius $r_0$ of its largest inner circle centered at the origin is equal to $\tan(\Phi)$, where $\Phi$ is the stability measure of the assembly $P$. The distance between a 2D point $v_k$ and $S(P)$ is denoted as:

$$\text{dist}(v_k, S(P)) = \begin{cases} \min_{v \in S} \|v_k - s\| & v \notin S \\ 0 & v \in S \end{cases}$$

(6)

where $\partial S$ denotes all vertices on the boundary of $S(P)$.

Our goal is to compute a $K$-gon with vertices $\{v_k\}$ that contains the target circle with $r_1 = \tan(\frac{\Phi}{2})$ and is close to the feasible section $S(P)$. We formulate this as an optimization problem:

$$\min_{\{v_k\}} \sum_{k=1}^{K} \text{dist}(v_k, S(P))$$

(7)

s.t. $\text{dist}(v_k, v_{k+1}, 0) = r_1, \forall k \in 1, \ldots, K$

where the line segment $v_k v_{k+1}$ is $k$th edge of the $K$-gon and $v_K+1 = v_0$. The constraints ensure that the target polygon always contains the target circle. Since this optimization has non-convex constraints, we solve it by using the interior conjugate gradient algorithm from knitro.

4 Gradient-based Structural Optimization

In the paper, structural optimization on the TI assemblies is formulated as:

$$\{\alpha^*_{ij}\} = \arg\min_{\{\alpha_{ij}\}} \sum_{k=1}^{K} f_k^* H f_k$$

(8)

s.t. $\text{Area}(C_i) \geq A_{\text{thres}}, \forall \text{contact } C_i$

$$\max(\alpha_{\text{min}}, \alpha^{\text{eq}}_{ij}) \leq \alpha_{ij} \leq \min(\alpha^{\text{max}}, \alpha^{\text{eq}}_{ij})$$

$$A_{\text{eq}} \cdot F_k = W_k$$

We solve above optimization following the gradient-based approach in [Whiting et al. 2012]. Given the optimization problem formulated as a quadratic program (QP) with inequality constraints, it can be transformed into QP with only equality constraints by considering active constraints at a local intermediate solution $f_k^*$, leading to a closed-form force solution:

$$f_k^* = H^{-1} C^T (CH^{-1} C^T)^{-1} b_k$$

(9)

where

$$C = \begin{bmatrix} A_{\text{eq}} \\ I_{lb} \end{bmatrix}$$

and $b_k = \begin{bmatrix} W_k \\ 0 \end{bmatrix}$

and the active lower bounds are denoted as $\tilde{I}_{lb} \cdot f_k^* = 0$, which contains the contact forces in $f_k^*$ that are exactly at zero.

We assume that the topology of each block and that of the block contact polygons remain the same for the differential movement of $\{\alpha_{ij}\}$, and derive an analytic gradient for $f_k^*$ with respect to the vector rotation angles $\{\alpha_{ij}\}$. Among the components of $f_k^*$ in Equation 9, only $A_{\text{eq}}$ and $W_k$ are the functions of $\{\alpha_{ij}\}$ while $H$ and $I_{lb}$ are constants. Hence, we compute the gradient of each element in $A_{\text{eq}}$, $W_k$, as well as $C_i$ with respect to $\{\alpha_{ij}\}$ using chain rule. In particular, the variables in $A_{\text{eq}}$ include block centroid, block contact normal and contact vertex, while the variables in $W_k$ are only about block volume. In the followings, we describe the derivations of each variable’s gradient with respect to the vector rotation angles $\{\alpha_{ij}\}$.

4.1 Definition

Following the constructive procedure to generate TI blocks, our differentiation pipeline for the variables in $A_{\text{eq}}$, $W_k$, and $C_i$ is shown as follows:

$$\alpha_{ij} \rightarrow \nabla n_{ij}^F \rightarrow \nabla F_i \rightarrow \nabla V_i \rightarrow \nabla e_{ij}^g \rightarrow -\nabla A_{ij}$$

The meaning of each variable in the pipeline is described as below and our goal is to calculate the derivatives of each variable with respect to $\alpha_{ij}$:

1. $n_{ij}^F$: normal of a face-face contact between $P_i$ and $P_j$ (pointing towards $P_j$).
2. $n_{ij}^F$: normal of an edge-edge contact between $P_i$ and $P_j$ (pointing towards $P_j$).
3. $\psi_k^i$: $k$th edge of the contact interface between $P_i$ and $P_j$.
4. $e_{ij}^g$: $g$th vertex of the contact interface between $P_i$ and $P_j$.
5. $A_{ij}$: area of contact interface between $P_i$ and $P_j$.
6. $o_i$: center of mass of $P_i$.
7. $V_i$: volume of $P_i$.

In our differentiation, most elements of gradients $\nabla X$ ($X = n_{ij}, v_{k}, \ldots$) are zero due to the fact that $\nabla X$ only depends on a few vector angles $\{\alpha_{ij}\}$. Hence, we only need to compute the potentially non-zero partial derivatives. We use the notation $\frac{\partial X}{\partial \alpha}$ to represent one of the non-zero derivatives $\frac{\partial X}{\partial \alpha}$ for writing simplicity.

4.2 Face-Face Contact Normal $n_{ij}^F$

The initial face-face contact normal $n_{ij}^{\text{init}} (\alpha_{ij} = 0)$ between $P_i$ and $P_j$ is:

$$n_{ij}^{\text{init}} = \text{Normalized}(e_{ij} \times (N_i + N_j))$$

(10)

While Normalized is a function which normalizes the input vector and $e_{ij}$ is the halfedge vector between $P_i$ and $P_j$. Rotating initial contact normal by $x_{ij} \alpha_{ij}$ around the halfedge vector $e_{ij}$ gives the expression of $n_{ij}^F$:

$$E_{ij} = \begin{bmatrix} 0 & -e_{ij}^y & e_{ij}^x \\ e_{ij}^y & 0 & -e_{ij}^x \\ -e_{ij}^x & e_{ij}^y & 0 \end{bmatrix}$$

(11)

$$n_{ij}^F = [\cos(x_{ij} \alpha_{ij}) E_{ij} + \sin(x_{ij} \alpha_{ij}) I_{3 \times 3}] n_{ij}^{\text{init}}$$

(12)

Thus, its derivative is:

$$\frac{\partial n_{ij}^F}{\partial \alpha_{ij}} = [x_{ij} \sin(x_{ij} \alpha_{ij}) E_{ij} + x_{ij} \cos(x_{ij} \alpha_{ij}) I_{3 \times 3}] n_{ij}^{\text{init}}$$

(13)

4.3 Edge-Edge Contact Normal $n_{ij}^E$

The normal of edge-edge contact is involved four face normals associated to the contact point, the two normals $n_k, n_i$ from $P_i$ and
the two normals \( \mathbf{n}_2, \mathbf{n}_3 \) from \( P_2 \). Then the normal of its edge-edge contact is:

\[
\mathbf{n}_{ij}^E = \frac{(\mathbf{n}_0 \times \mathbf{n}_1) \times (\mathbf{n}_2 \times \mathbf{n}_3)}{\left \| (\mathbf{n}_0 \times \mathbf{n}_1) \times (\mathbf{n}_2 \times \mathbf{n}_3) \right \|}
\]  
(14)

In case of the \( \mathbf{n}_{ij}^E \) pointing toward \( P_i \) instead of \( P_j \), we will reverse the direction of \( \mathbf{n}_{ij}^E \). Its derivative can be computed by applying the chain rules.

### 4.4 Block Vertex \( v_i^h \)

In section 5 of the paper, the geometry of \( P_i \) is constructed by intersecting several 3D half spaces (planes). A valid part vertex \( v_i^h \) are on at least three of these half spaces/planes:

\[
(\mathbf{n}_{ij}^h, d_{ij}^h), j = 0, 1, 2
\]  
(15)

The normal vector \( \mathbf{n}_{ij}^h \), a.k.a \( \mathbf{n}_j \), is one of the face-face contact normals associated to \( P_i \), vertex \( v_i^h \). The \( d_{ij}^h \), a.k.a \( d_j \), is the inner product of any point on \( j \)th plane in Eq. 15 and the normal \( \mathbf{n}_j \). Solving the following linear system gives the vertex \( v_i^h \):

\[
\begin{aligned}
\begin{bmatrix}
\mathbf{n}_{j1}^h \\
\mathbf{n}_{j2}^h
\end{bmatrix}
&= (N_j^{-1})^{-1} D_j^h \\
&= (N_j^{-1})^{-1} D_j^h
\end{aligned}
\]

According to the derivative formulation of inverse matrix:

\[
\frac{\partial N^{-1}}{\partial \alpha} = -N^{-1} \frac{\partial N}{\partial \alpha} N^{-1}
\]

substitute Eq. 17 into Eq. 16 gives:

\[
\frac{\partial v_i^h}{\partial \alpha} = -(N_i^h)^{-1} \frac{\partial N_i}{\partial \alpha} (N_i^h)^{-1} \frac{\partial D_i^h}{\partial \alpha}
\]

### 4.5 Block Volume \( V_i \)

Triangulating the faces of \( P_i \) gives a set of triangles \( A_t(\mathbf{a}_t, \mathbf{b}_t, \mathbf{c}_t), t = 0, ..., T - 1 \) which is ordered counter clockwise on its corresponding face of \( P_i \). Suppose

\[
\hat{\mathbf{n}}_t = (\mathbf{b}_t - \mathbf{a}_t) \times (\mathbf{c}_t - \mathbf{a}_t)
\]  
(19)

The formulation of block volume is:

\[
V_i = \frac{1}{6} \sum_{t=0}^{T-1} \mathbf{a}_t \cdot \hat{\mathbf{n}}_t
\]

(20)

Its derivative can be computed by applying the chain rules. Please refer to [Nurnberg 2013] for more details.

### 4.6 Block Centroid \( \mathbf{o}_i \)

Denoting the standard basis in \( \mathbb{R}^3 \) by \( \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3 \), the centroid \( \mathbf{o}_i \) of \( P_i \) is:

\[
\mathbf{e}_u \cdot \mathbf{o}_i = \frac{1}{48V_i} \sum_{t=0}^{T-1} (\hat{\mathbf{n}}_t \cdot \mathbf{e}_u) \left[ (\mathbf{a}_t + \mathbf{b}_t) \cdot \mathbf{e}_u \right]^2 - \left[ (\mathbf{c}_t + \mathbf{b}_t) \cdot \mathbf{e}_u \right] \\
+ \left[ (\mathbf{c}_t + \mathbf{a}_t) \cdot \mathbf{e}_u \right]^2 + \left[ (\mathbf{c}_t + \mathbf{a}_t) \cdot \mathbf{e}_u \right]^2, u = 1, 2, 3
\]

(21)

Its derivative can be computed by applying the chain rules. Please refer to [Nurnberg 2013] for more details.

### 4.7 Contact Vertex \( c_{ij} \)

The paper lists two kinds of contact vertices, we only give the derivatives expression for the face-face contact case while the vertex of an edge-edge contact is a constant point of the surface tessellation.

The contact polygon \( C_i \) are the result of boolean intersection between two overlap faces of neighboring \( P_i \) and \( P_j \). The vertices of this polygon are either 1) a vertex of the two neighboring part (\( v_i^h \) or \( v_j^h \)); or 2) the intersection of two non-parallel edges of \( P_i \) and \( P_j \), which called \( e_{ij}^0 \) (\( j \)th point of the \( C_i \)). Suppose the two lines are:

\[
\begin{aligned}
(v_{0j}^h, v_{1j}^h), (v_{2j}^h, v_{3j}^h)
\end{aligned}
\]

(22)

a.k.a as:

\[
(v_0, v_1), (v_2, v_3)
\]

(23)

The contact vertex \( c_{ij}^g \) satisfies the following equations

\[
\begin{aligned}
e_{ij}^g &= v_0 + (v_1 - v_0)s = v_2 + (v_3 - v_2)t
\end{aligned}
\]

(24)

Denote the \( \mathbf{v}_{a,b} = \mathbf{v}_a - \mathbf{v}_b \). The solution of linear equation \( s, t \) is:

\[
\begin{aligned}
\begin{bmatrix}
\mathbf{v}_{1,0} \cdot \mathbf{v}_{1,0} & -\mathbf{v}_{1,0} \cdot \mathbf{v}_{2,3} \\
-\mathbf{v}_{1,0} \cdot \mathbf{v}_{2,3} & \mathbf{v}_{2,3} \cdot \mathbf{v}_{2,3}
\end{bmatrix}
&= \begin{bmatrix}
\mathbf{v}_{1,0} \cdot \mathbf{v}_{2} - \mathbf{v}_{1,0} \cdot \mathbf{v}_{0} \\
\mathbf{v}_{2,3} \cdot \mathbf{v}_{2} - \mathbf{v}_{3,2} \cdot \mathbf{v}_{0}
\end{bmatrix}
\end{aligned}
\]

(25)

Substituting the solution into Equation 24, its derivative can be computed by applying the chain rules.

### 4.8 Contact Area \( A_{ij} \)

The contact polygon \( C_i \) is a list of points \( c_{ij}^g \) (\( g = 0, ..., |C_i| - 1 \)) which are in the same orientation of contact normal \( \mathbf{n}_{ij} \). The area of the polygon is:

\[
A_{ij} = \frac{1}{2} \sum_{g=1}^{|C_i|-2} \left[ (c_{ij}^g - c_{ij}^{g+1}) \times (c_{ij}^{g+1} - c_{ij}^g) \right] \cdot \mathbf{n}_{ij}
\]

(26)

Its derivative can be computed by applying the chain rules.

### References


