Proceedings of IASS-IWSS 2026 – September 14-18, 2026, Turin, Italy
"R-Evolution of Shapes: Sustainability, Re-Use, and New Design Paradigms"
Annual Symposium of the International Association for Shell and Spatial Structures
3rd Italian Workshop on Shell and Spatial Structures
A. Manuello Bertetto, S. Gabriele, F. Marmo, A. Micheletti (eds.)

A reconfigurable hardware-geometry system for the assembly of multiple structures from finite panel kits and adaptable nodes

Sam WILCOCK*, Emil KORKIS, and Ornella IUORIO

Department of Architecture, Built Environment and Construction Engineering, Politecnico di Milano
Via Giuseppe Ponzio 31, 20133, Milano, MI, Italy
* sam.wilcock@polimi.it

Abstract

This paper presents an integrated hardware-geometry framework for approximating semi-freeform shell and spatial structures through a finite kit of rigid panels and a reconfigurable 3D-printed vertex joint. Connecting exactly three panels at a shared vertex, the joint combines a continuous dihedral fold range of $-50°$ (closing) to $+10°$ (hyperextension) about flat with $\pm 30°$ of in-plane rotation around a neutral $120°$ configuration, and its geometry requires the projected arm directions to sum to $360°$ in the connection plane, accommodating convex panel corners of 90–150°. These bounds define a constrained design space in which doubly curved geometries can be approximated without bespoke components. A complementary digital workflow rationalises freeform surfaces into this space: candidate panels harvested from planarised hex-dominant meshes are clustered into a finite kit of convex polygonal types by data-driven analysis of edge lengths and internal angles; valid three-panel vertex configurations (triads) are enumerated with every dihedral constrained to the joint's fold range; and an integer linear programme selects a non-overlapping subset from a redundant pool of surface-anchored candidate placements, maximising panel-to-panel edge bonding whilst steering the curvature-correct tile to each location. The method thereby turns a design intent into an enumerated family of buildable discrete structures assembled from one finite kit of parts.

Keywords: reconfigurable structures, kit of parts, panel systems, freeform rationalisation, vertex joints, computational design, doubly curved shells

1. Introduction

Contemporary structural design faces increasing pressure to reduce material waste and extend the useful life of building components. One response is the development of reconfigurable and reusable structural systems, in which a finite inventory of fabricated parts is assembled, disassembled, and reassembled into multiple distinct configurations over its lifespan [1] – the kit-of-parts principle, whereby a fixed palette of standardised components produces structural variety through intelligent combinatorics rather than bespoke fabrication [2].

The particular challenge addressed here is to extend this approach to doubly-curved shell and spatial structures. While reconfigurable systems for planar and single-curvature geometries are relatively well established [3], achieving non-trivial curvature from a finite discrete panel kit demands careful coordination between the physical joint mechanism, the geometry of the panel set, and the algorithmic workflow used to rationalise a design intent into that panel set [4]. Without such coordination, the combinatorial compatibility between adjacent panels is typically too sparse to tile a curved surface continuously.

This paper presents an integrated hardware-geometry system that addresses these difficulties simultaneously. At the hardware level, a 3D-printed reconfigurable vertex joint connects exactly three flat panels at a shared vertex, providing two bounded degrees of freedom per arm – a continuous dihedral fold and an in-plane rotation, with ranges detailed in Section 2.4 – whose limits define a constrained design space of physically realisable, geometrically consistent vertex configurations. At the geometry level, a computational pipeline rationalises arbitrary target surfaces into a panel kit compatible with those constraints, clustering candidate panels into a finite typed set, enumerating valid three-panel vertex configurations (*triads*), and tiling the surface by formulating placement as an integer linear programme.

The key contribution is the tight coupling between the joint's mechanical bounds and the panel combinatorics: encoding the continuous fold range as a hard constraint in the geometric search ensures every computed triad is directly realisable by the physical joint, and a parameter sweep over kit size and edge-length discretisation identifies configurations that maximise the non-planar triads available for curved assembly. Section 2 describes the four-stage methodology, Section 3 presents the parameter sweep and applications to test surfaces, and Section 4 draws conclusions.

2. Methodology

The pipeline proceeds in four stages (Figure 1): candidate panel generation, clustering into a finite typed kit with compatibility analysis, ILP-based surface tiling, and realisation of connections via the reconfigurable vertex joint.

2.1. Kit of panels generation in Grasshopper

A custom Grasshopper parametric script instantiates three baseline structural typologies – domes, hypars, and blobs – and parametrically deforms each to yield 100 distinct variations per typology. Each variation receives a hex-dominant tiling whose faces are planarised with the Kangaroo solver [5], producing a large population of discrete, flat convex polygons spanning a variety of curvature conditions; these are characterised by their edge-length and angle sequences and exported as the dataset feeding the clustering

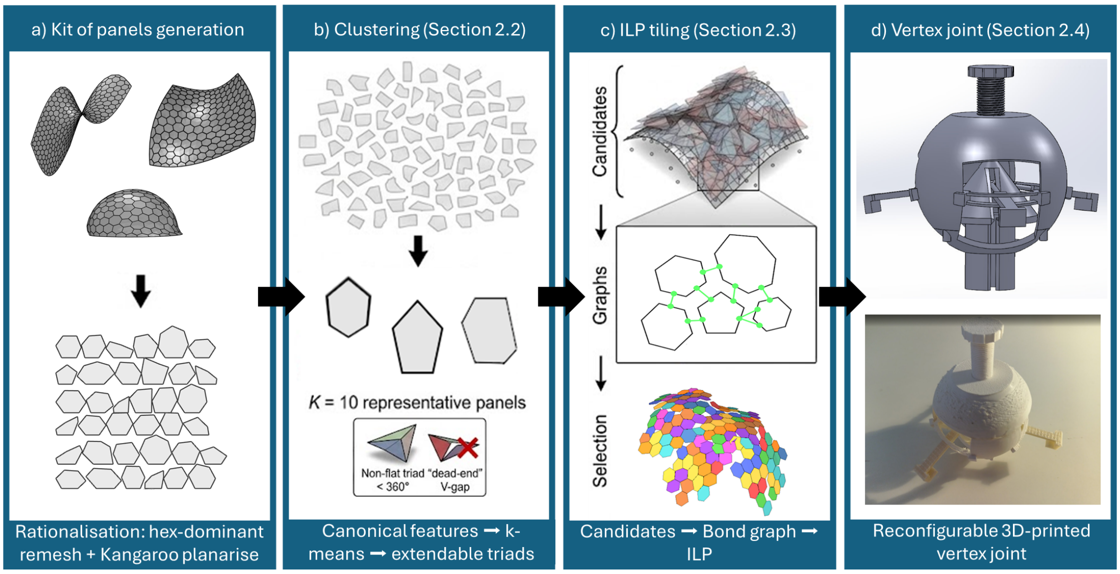


Figure 1: Overview of the four-stage hardware-geometry pipeline. (a) Mesh rationalisation: arbitrary surfaces are remeshed to a hex-dominant quad mesh and planarised in Kangaroo [5], filtering panels to the joint's fold range. (b) Panel clustering: the candidate population is reduced to K canonical types via k-means on invariant feature vectors, with valid triads enumerated and dead-ends removed. (c) ILP tiling: an integer linear programme extracts a non-overlapping, edge-bonded subset from a redundant pool of candidate placements. (d) Hardware realisation: a reconfigurable 3D-printed vertex joint accommodates the fold space via a continuous dihedral range of $-50°$ to $+10°$ and a $\pm 30°$ rotational collar.

phase. The base typologies serve only as a generator of realistic training data – planarisation couples edge lengths and corner angles as a buildable hex panel actually constrains them – and do not propagate into the final kit: clustering discards their global combinatorics, retaining only the recurring local panel shapes.

2.2. Clustering of panels

2.2.1. Invariant features

The remeshing process yields hundreds of candidate panels that must be reduced to a finite and manageable kit. The same physical panel shape can, however, be extracted with different vertex orderings – starting at any corner, traversed in either direction – producing up to $2n$ distinct raw sequences for a polygon with n edges, so each panel is encoded as a *canonical feature vector* invariant to these ambiguities.

Let a panel be described by its perimeter-normalised edge lengths $\hat{\ell}_0, \ldots, \hat{\ell}_{n-1}$ (with $\sum \hat{\ell}_i = 1$) and interior angles $\theta_0, \ldots, \theta_{n-1}$, where index i refers to the vertex between edges $i-1$ and i. These are interleaved into a flat sequence

$$\mathbf{f} = [\,\hat{\ell}_0,\ \theta_0,\ \hat{\ell}_1,\ \theta_1,\ \ldots,\ \hat{\ell}_{n-1},\ \theta_{n-1}\,]. \tag{1}$$

All $2n$ candidate representations of the same shape are then generated: n *forward* variants, each a cyclic shift of $\mathbf{f}$ by $2k$ positions for $k = 0, \ldots, n-1$ (each vertex occupies two consecutive entries), and n *reverse* variants, formed by reversing the edge and angle sequences to model the opposite traversal direction (equivalently, a mirror image) before applying the same shifts. The canonical representation is the lexicographically smallest element of this set:

$$\mathbf{f}^* = \min_{\text{lex}} \{\ \text{all } 2n \text{ variants of } \mathbf{f}\ \}. \tag{2}$$

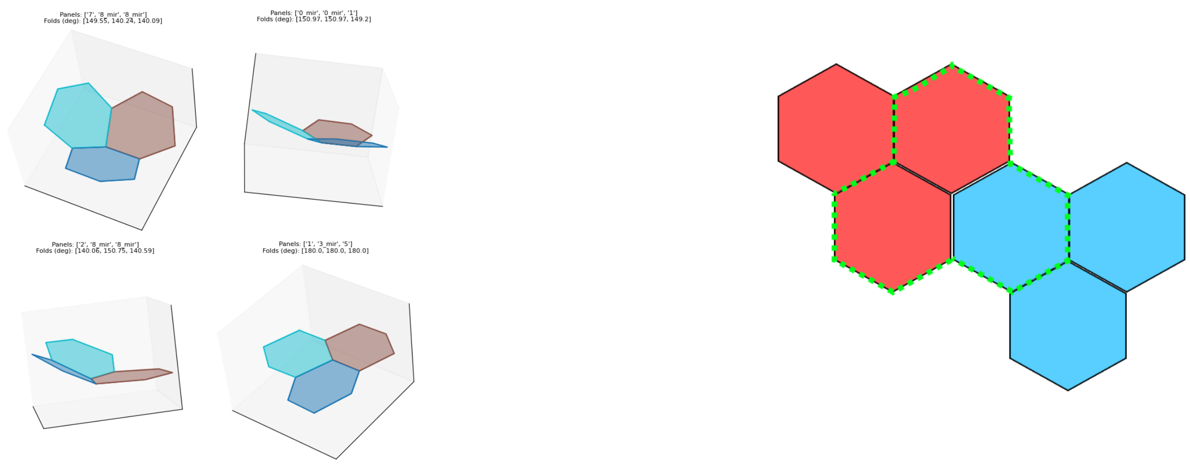
The lexicographic minimum picks a single deterministic representative, so the same physical polygon always maps to the same $\mathbf{f}^*$ regardless of extraction ordering, whilst perimeter normalisation provides scale invariance. A panel and its mirror image map to the *same* canonical vector (the forward variants of one are the reverse variants of the other), so clustering groups by abstract polygon type; mirror copies are reintroduced as distinct panel types during triad enumeration, ensuring chirality is respected without fragmenting the clustered population.

Edge lengths are additionally discretised onto a vocabulary of n_s standard values before computing $\mathbf{f}^*$, where n_s (denoted `num_std_lengths`) is a user-controlled parameter. The full panel population is then clustered into K types using k-means on $\mathbf{f}^*$ [6, 7]. The centroid of each cluster provides the representative geometry for that panel type.

2.2.2. Searching for high compatibility

A central requirement of the system is that the panel kit should support a rich set of three-panel vertex configurations, here called *triads*. A triad is a tuple of three panels (P_i, P_j, P_k) with chosen vertex indices (a, b, c) such that the interior angles at those vertices sum to at most 2π and every pairwise shared edge length matches within tolerance. The signed fold deviations implied by the spherical law of cosines at the shared vertex must additionally lie within the joint's continuous range, $-50° \leq \delta \leq +10°$ for each arm. Non-flat triads (angle sum $< 2\pi$) are of primary interest as they are the mechanism by which the system achieves curvature.

Not all triads are topologically extendable: a triad whose V-gaps (the junctions between adjacent panels) cannot be filled by any other valid triad creates a dead end that would halt surface coverage (Figure 2). A fixed-point iteration deletes triads with at least one unfillable gap until the set stabilises; the surviving set is called the *extendable* kit. To identify the clustering parameters maximising its size and curvature



(a) (b)

Figure 2: (a) 4 triad configurations found, with variable sets of fold angles, and (b) checking extended compatibility with other triads, projected into 2D, with colours showing two different triads. If two triads are compatible, the highlighted central region should also be a possible triad.

range, a grid search is performed over $K \in \{5, 10, 15, 20\}$ and $n_s \in \{2, 3, 4, 5, 6, 7\}$, recording for each pair the total valid triads, the number surviving dead-end filtering, and the number of non-planar extendable triads – the last directly indicating the system's capacity to tile curved surfaces.

2.3. Tiling meshes

Given the extendable triad kit $\mathcal{T}$ and a target mesh $\mathcal{M}$, the surface is tiled by panels drawn from the kit. The core difficulty is that a rigid flat panel cannot in general be laid tangent to a doubly curved surface: any greedy growth that commits to a single panel at every step accumulates closure error and self-intersects where two growing arms wrap around and meet. Candidate generation is therefore *decoupled* from selection: a deliberately redundant pool of overlapping placements is enumerated on the surface and sealed against loop-closure drift as it is embedded (Section 2.3.1), from which an edge-bonded, non-overlapping subset is extracted by an integer linear programme (ILP, Section 2.3.3). Light post-selection culls (surface lift, connectivity, residual overlap) then finish the assembly, and the surviving panels are snapped to their best-fit planes and dimensionally inset to leave mechanical clearance for the shared vertex joints.

Throughout, a candidate placement u is an embedded panel with ordered vertices $P_0^u, \dots, P_{n_u-1}^u \in \mathbb{R}^3$ and edges $e_i^u = (P_i^u, P_{i+1}^u)$ of length ℓ_i^u (indices modulo n_u). We write $c_u = \frac{1}{n_u} \sum_i P_i^u$ for the centroid and $\mathbf{n}_u$ for the unit face normal.

2.3.1. *Candidate generation*

Curvature budget. The amount of curvature a triad must absorb at a mesh vertex is fixed by the surface. For an interior vertex v with incident face angles $\{\phi_i\}$, the *angular deficit*

$$\kappa(v) = 2\pi - \sum_i \phi_i \tag{3}$$

is the discrete Gaussian curvature concentrated at v (the integrated curvature of the Gauss–Bonnet theorem) [8]. A triad seated at v can lie tangent to the surface only if the three interior angles it presents at v sum to $2\pi - \kappa(v)$; the residual angle is taken up by the dihedral folds δ, each constrained to the joint range $-50° \leq \delta \leq +10°$. A too-coarse mesh concentrates more deficit at each vertex than any triad in $\mathcal{T}$ can fold to, so for procedural targets the mesh resolution is chosen automatically to bring the 90^{th}-percentile vertex deficit below the median deficit the kit's combos can realise.

Breadth-first growth. Candidate placements are generated by breadth-first growth from a seed triad on a chosen face. At each open vertex gap the search instantiates the kit triads whose available edge lengths match the exposed edge and whose interior-angle sum best matches the local deficit $\kappa(v)$, so loops tend to close on the curved surface. The three-dimensional embedding propagates fold angles outward along the growth tree, each dihedral drawn from the joint range and signed so the panel normal aligns with the local mesh normal $\mathbf{n}_{\mathcal{M}}(c_u)$. Since each panel is positioned off its parent's edge alone, positional error would otherwise compound down every growth arm; each placed panel is therefore slid rigidly so its centroid returns to the surface, converting accumulated arm drift into a small per-step gap the joint clearance absorbs.

Ring welding. The tree embedding constrains only parent–child edges, yet the growth records a full bond graph whose lateral edges close loops between branches. Taking the tree pose as an initial guess, panels are ordered into rings by graph distance from the seed; with inner rings frozen, each frontier panel is re-fitted by the rigid transform – solved in closed form by the Kabsch algorithm [9] – minimising the summed endpoint mismatch of its *recorded* shared edges to their placed neighbours, then re-anchored, with Gauss–Seidel sweeps repeating until the ring stops moving. Only recorded bonds pull, so non-adjacent panels are never forced together and loop error redistributes into hairline per-edge gaps whilst panels remain dimensionally exact.

Redundancy. A single growth pass yields one topology; the solver is instead given a deliberately over-complete pool drawn from multiple seeds, per-seed filler variants, and in-plane seed rotations, with every alternative filler at a contested gap instantiated as a mutually overlapping "ghost" candidate – the ILP, not the growth heuristic, then decides which filler occupies each gap. The pool is de-duplicated (exact template-and-pose copies only inflate the conflict graph), clipped to the mesh outline, and augmented with a centre panel dropped into any near-closed ring the growth left open (a *pocket*), whose small angular warp the ring weld and joint clearance absorb.

2.3.2. *Bond and conflict graphs*

Two graphs over the candidate set encode the combinatorial structure of a valid assembly.

A *bond* represents a shared structural edge between two candidates. Because two panels meeting at an edge traverse it in opposite directions, edge (u, i) and edge (v, j) bond when their lengths agree and their endpoints coincide reversed, within a tolerance τ_b (by default 10% of the mean kit edge length):

$$|\ell_i^u - \ell_j^v| \leq \tau_b \quad \text{and} \quad \max\big(\|P_i^u - P_{j+1}^v\|,\ \|P_{i+1}^u - P_j^v\| \big) \leq \tau_b. \tag{4}$$

The ring welding of Section 2.3.1 pulls loop-closing edges within this tolerance, so the bond graph captures the full loop topology rather than the growth tree alone; only genuinely unclosable seams remain unbonded. Each candidate edge is a *bond slot* admitting at most one active bond; the set of bonds incident on slot (u, e) is denoted $S(u, e)$.

A *conflict* marks two unbonded candidates that cannot physically coexist. A broad phase over centroids and circumradii discards pairs whose circumscribed spheres are disjoint; a narrow phase then confirms a true interior overlap, either coplanar (via separating-axis projection [10]) or a transversal intersection in which an edge of one panel pierces the face of the other, as folded panels crossing at a dome rim present without being coplanar. Bonded pairs are excluded, since legitimate edge-sharing neighbours are governed by slot exclusivity instead; a final *seam* conflict marks growth-adjacent candidates whose shared edge is embedded too far apart ever to close, forcing the ILP to pick a clean side of an unclosable loop rather than a lifted edge.

2.3.3. *Integer programme*

Let $x_u \in \{0,1\}$ select candidate u and $y_b \in \{0,1\}$ activate bond b, which joins panels $p(b)$ and $q(b)$. The assembly is chosen by maximising a weighted count of selected panels and active bonds,

$$\max_{\mathbf{x},\,\mathbf{y}} \;\; \sum_u w_u\, x_u \; + \; w_b \sum_b y_b, \tag{5}$$

subject to the constraints below. The weights w_u and w_b are not free tuning knobs but follow from a single principle: every *interior* edge slot (one with at least one potential bond partner in the pool) left unmatched incurs a penalty p, discouraging fragmented arrays and isolated panels. Since each active bond covers exactly two slots, the summed penalty separates into a per-panel and a per-bond term, and folding these into (5) gives the effective weights

$$w_u = 1 - p\, s_u - w_f\,(1 - \tau_u), \qquad w_b = \lambda + 2p, \tag{6}$$

where s_u is the number of interior slots on panel u and λ is the base reward per bond. With $p = \frac{1}{4}$ an isolated hexagon carries weight $1 - \frac{6}{4} < 0$ and is selected only if enough of its edges bond to compensate. The term $w_f\,(1-\tau_u)$ is a curvature-matching bias built on the Goldberg-polyhedron principle that a sphere closes with flat hexagons and exactly twelve pentagons: $\tau_u \in [0,1]$ multiplies the candidate's tangency to a locally averaged surface normal by a sign-agreement factor $\frac{1}{2}\big(1+\tanh(d_u\,\kappa_u/\bar{\kappa})\big)$ between the panel defect $d_u = 6 - n_u$ (pentagon $+1$, hexagon 0, heptagon -1) and the local vertex deficit κ_u, so among the overlapping ghost fillers at a gap the ILP keeps the tile whose *shape* suits the local curvature whilst a curvature-wrong tile with no alternative still beats a hole.

The objective is subject to:

$$y_b \le x_{p(b)}, \quad y_b \le x_{q(b)}, \quad y_b \ge x_{p(b)} + x_{q(b)} - 1 \qquad \text{(bond activation)} \tag{7a}$$

$$\sum_{b\in S(u,e)} y_b \le 1 \qquad \text{(slot exclusivity)} \tag{7b}$$

$$x_u \le \sum_{b\,\ni\, u} y_b \qquad \text{(no floaters)} \tag{7c}$$

$$\sum_{u\in Q} x_u \le 1, \quad \forall\, Q \in \mathcal{Q} \qquad \text{(conflict cliques)} \tag{7d}$$

$$\sum_{u\in U_t} x_u \le N_t, \quad \forall\, t \in \mathcal{K} \qquad \text{(part inventory)} \tag{7e}$$

Constraint (7a) is the standard linearisation of the logical product $y_b = x_{p(b)} \wedge x_{q(b)}$; slot exclusivity (7b) forbids an edge from bonding two neighbours at once; and the no-floater constraint (7c) forces every selected panel to carry at least one active bond, stopping the solver from harvesting cheap isolated panels that would fragment the result. The inventory constraints (7e) make this a true kit-of-parts problem: each template $t \in \mathcal{K}$ has finite physical stock N_t (U_t being the candidates of that type), so once it is exhausted the remaining surface must be covered with other types or left open – exactly the decision a greedy construction cannot make; $N_t \to \infty$ recovers the unbounded case.

The conflict constraints (7d) govern solver performance. Stated edge by edge as $x_u + x_v \le 1$ the linear relaxation is weak (a triangle of mutual conflicts admits $x_u = x_v = x_w = \frac{1}{2}$), so the pairwise cuts are covered by greedily grown cliques $\mathcal{Q}$ with one constraint per clique, tightening the relaxation and shrinking the constraint count by an order of magnitude. The programme then reduces to a maximum-weight independent-set problem augmented by the bond rewards, and is solved with Gurobi [11] where licensed (else the open-source HiGHS [12]) under a wall-clock limit with fixed seeds, warm-started with a greedily grown bonded, conflict-free patch cover so branch-and-cut improves a good incumbent rather than searching from zero; on the dome, Gurobi certifies optimality in seconds where HiGHS exhausts the limit. Single-component connectivity is deliberately *not* enforced inside the ILP – the natural flow encoding gives a weak big-M relaxation – and is instead recovered cheaply as a post-filter on the final geometry.

2.4. Node joint design

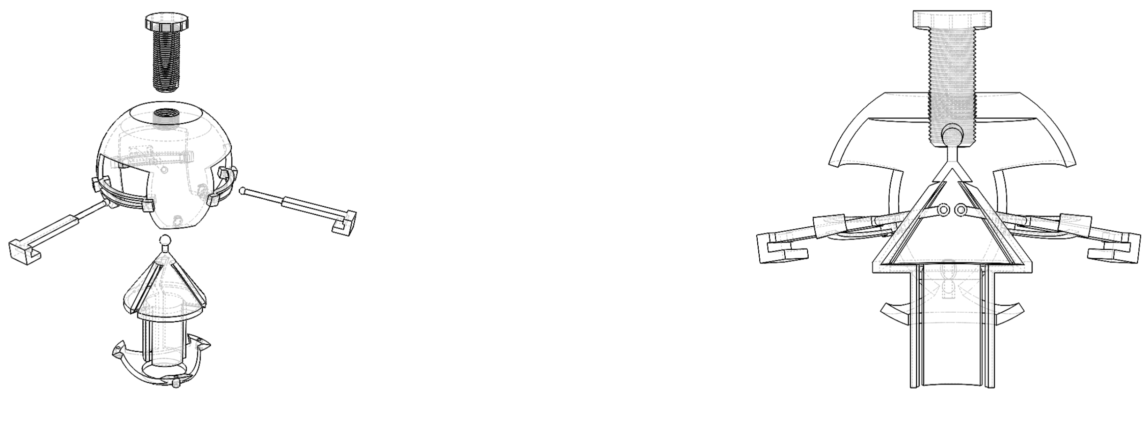
The vertex joint is a 3D-printed component with three identical arms arranged at approximately 120° to one another in plan (Figure 3). Each arm engages a single panel edge and provides two mechanical degrees of freedom. A continuous dihedral mechanism allows fold deviations from $-50°$ (closing direction) to $+10°$ (hyperextension) about the flat configuration, locked at the assembled angle; a rotational collar additionally allows ±30° of in-plane reorientation around the 120° neutral angle. The asymmetric fold range reflects the mechanical clearance between adjacent panels: deep folds approximate curved surfaces, whilst the small hyperextension margin tolerates assembly slack. A fundamental geometric constraint governs assembly: the projections of the three arm directions onto the local connection plane must sum to 360°, which restricts panel interior angles at the joint to the range 90°–150° and is enforced analytically during triad enumeration. Both freedoms are free only during placement: a screw drawn down the joint's axis clamps an internal conical wedge against all three arms, locking fold and rotation in one action (Figure 3b); the assembled node's stiffness comes from the screw preload rather than the printed sliding collar, which carries minimal load once locked.

Assembly transcribes this solved configuration directly: the tiling and welding stages fix, at each vertex, the panel types meeting there and every arm's fold and collar angle. Setting a triad is largely self-registering – matched edge endpoints and fixed corner angles admit essentially one seating – so the panel edges drop into the sliding cuffs and the screw locks with little on-site judgement, shared panels chaining outward with no bespoke parts.

3. Results

3.1. Clustering parameters search results

Figure 4 maps the extendable and non-planar-extendable triad counts across the parameter grid. The total count grows rapidly with K (a few dozen at $K = 5$; over 400 extendable triads at $K = 10$, $n_s = 2$), but the non-planar count – the metric governing curved coverage – is non-monotonic, peaking



(a) (b)

Figure 3: Reconfigurable vertex joint: (a) exploded view showing dihedral locking mechanism, sliding cuffs and rotational collar, (b) section view of the joint internals. Per arm, the dihedral mechanism spans $-50°$ to $+10°$ about flat and the collar allows $\pm 30°$ of in-plane rotation about the 120° neutral position.

at 81 for $K = 10$, $n_s = 2$ and falling at $K = 15$ and $K = 20$, where the extra panel types fragment the edge-length distribution and reduce cross-compatibility. The discretisation acts oppositely: fewer standard lengths pool more panels onto shared edges, raising compatibility at the cost of geometric fidelity. We therefore adopt $K = 10$, $n_s = 2$ throughout as a trade-off between kit diversity and extendability at a testable kit size.

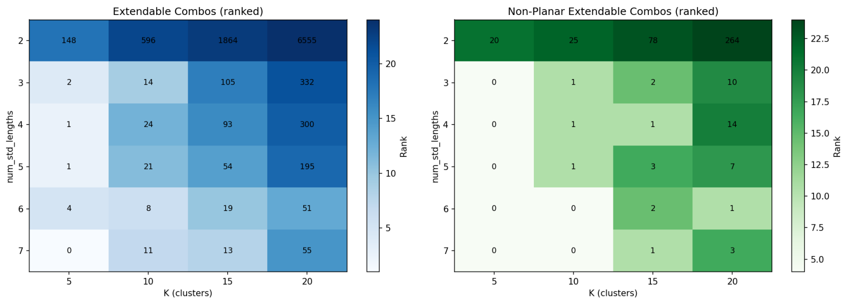


Figure 4: Parameter sweep heatmaps: (left) total extendable triad count; (right) non-planar extendable triad count. Cells show raw counts; colour encodes rank. Selected parameters ($K = 10$, $n_s = 2$) are highlighted.

3.2. Panel cluster designs

At $K = 10$, the clustering produces ten panel types plus their mirror images, giving 20 distinct tiles ranging from near-equilateral quadrilaterals to elongated pentagons and hexagons (Figure 5). The selected kit spans the joint's full closing-direction fold range (deviation magnitudes up to 50°) across its non-planar triads, with the majority of curved configurations concentrated in the 10°–30° band.

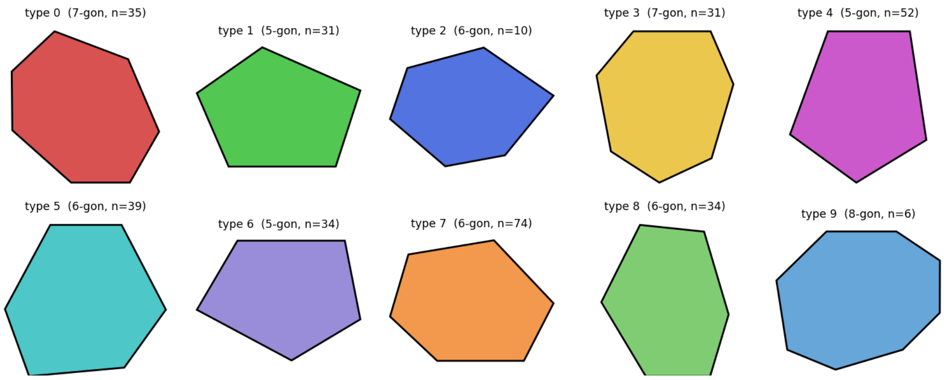


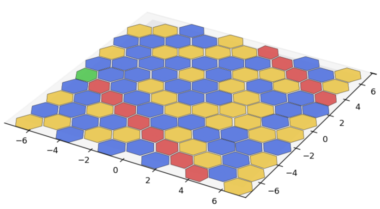
Figure 5: The ten base panel types in the selected kit ($K = 10$, $n_s = 2$), edges coloured by length.

3.3. Application to forms

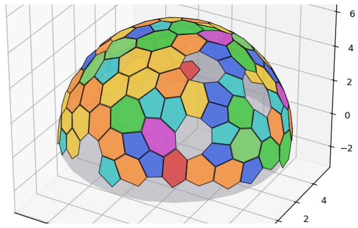
The ILP tiling pipeline is applied to three test surfaces: a flat plane, a hemispherical dome (high uniform curvature) and a hand-designed shell with double curvature. On the flat surface, the solver selects a regular planar tiling of triads with zero fold deviation. On the dome, triads with fold deviations of 20°–50° dominate at the crown, and the curvature-matching bias concentrates pentagons there – echoing the twelve a Goldberg sphere requires – yielding a single edge-connected shell with no interpenetration under a separating-axis audit, suggesting the system's capacity to accommodate high Gaussian curvature from the discrete kit.

For the double curved shell, the contraflexure region behaves as expected: the hex-dominant tiling cannot cross from positive to negative Gaussian curvature with convex tile perimeters. The non-convex hexagons that resolve sign-changing curvature in freeform honeycomb meshes [13] are deliberately excluded here

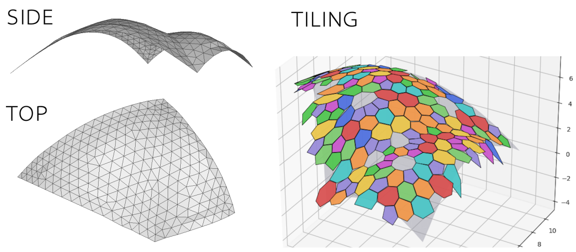
– a reflex corner violates the joint's 90°–150° convex-angle range and would also fragment the clustering – so the present kit trades contraflexure coverage for joint compatibility. Figure 6 illustrates the tiled outputs for the three cases, with panel faces coloured by panel type from the kit.



(a)



(b)



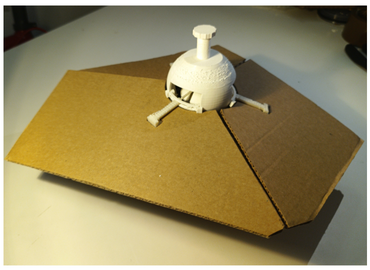
(c)

Figure 6: ILP tiling results: (a) Planar surface; (b) hemispherical dome; (c) designed shell, with base design on the left. Panel colour shows different panels from the kit, the same colour for standard or mirrored (upside down).

3.4. Panel assembly

A physical proof-of-concept assembly was produced using cardboard panels corresponding to three cluster types from the selected kit, joined by a PLA 3D-printed joint to the Section 2.4 design (Figure 7) with a semi-circular region removed at a vertex and a hole cut in for the arm ends to engage. Early assembly tests suggest that the dihedral locking mechanism engages reliably for initial structural validation.

(a)



(b)

Figure 7: (a) 3D printed joint, (b) joint pulling in 3 panels into locked angular configuration.

4. Conclusions

This paper has presented an integrated hardware-geometry system for constructing reconfigurable doubly-curved spatial structures from a finite kit of flat panels. The core innovation is the tight coupling between a mechanically bounded 3D-printed vertex joint and a computational pipeline that respects its fold bounds throughout panel clustering, triad enumeration, dead-end filtering, and ILP-based surface tiling, circumventing the need for bespoke connecting nodes. The parameter sweep indicated that $n_s = 2$ edge-length classes maximise non-planar triad availability whilst $K = 10$ panel types provide a sufficient catalogue of curved configurations, and physical prototyping demonstrated the functional viability of the dihedral locking mechanism. Ultimately, this framework advances kit-of-parts construction for non-standard surfaces, offering a pathway toward more sustainable, demountable, and adaptable architectural systems.

Several extensions are planned. The residual gaps that remain after welding are of two kinds: coverage slots the candidate pool never offered a tile for, and vertex wedges narrower than the kit's smallest corner angle, which no placement can close – a limit of the panel inventory rather than the optimisation. Enriching the kit at its acute end and freeing template assignment within the optimisation are therefore primary goals which may require further node development, alongside triangular or quadrilateral tiles for sign-changing curvature. The panel development strategy should be expanded to improve coverage, preventing the holes seen in the meshes in Figure 5. Structural performance must also be evaluated: a load-path analysis of the locked joint is needed to establish viability for larger spans, culminating in full-scale prototyping to measure assembly tolerances and reconfiguration effort.

Acknowledgements

Work part of the AS_Lab, Politecnico di Milano.

References

[1] K. Ostapska, P. Rüther, A. Loli, and K. Gradeci, "Design for disassembly: A systematic scoping review and analysis of built structures designed for disassembly," *Sustainable Production and Consumption*, vol. 48, pp. 377–395, 2024.

[2] J. Brütting, G. Senatore, and C. Fivet, "Design and fabrication of a reusable kit of parts for diverse structures," *Automation in Construction*, vol. 125, 2021.

[3] E. Korkis and O. Iuorio, "Robotic manipulators as advanced manufacturing agents for laser-cut construction systems," in *AM Perspectives 2: Research in advanced manufacturing for architecture and construction* (O. Iuorio, A. Wolf, B. Figueiredo, U. Knaack, and P. J. S. Cruz, eds.), pp. 184–191, Stichting OpenAccess, 2025.

[4] H. Pottmann, A. Asperl, M. Hofer, and A. Kilian, *Architekturgeometrie*. Springer & Bentley Institute Press, 1st ed., 2010.

[5] D. Piker, "Kangaroo: Form finding with computational physics," *Architectural Design*, vol. 83, no. 2, pp. 136–137, 2013.

[6] S. Lloyd, "Least squares quantization in pcm," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[7] T. Kimura and H. Takeuchi, "Form-finding for free-form reticulated shell structure considering structural performance and regularization of node connection patterns," *Journal of the International Association for Shell and Spatial Structures*, vol. 64, no. 4, pp. 306–315, 2023.

[8] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr, "Discrete differential-geometry operators for triangulated 2-manifolds," in *Visualization and Mathematics III* (H.-C. Hege and K. Polthier, eds.), Mathematics and Visualization, pp. 35–57, Berlin, Heidelberg: Springer, 2003.

[9] W. Kabsch, "A solution for the best rotation to relate two sets of vectors," *Acta Crystallographica Section A*, vol. 32, no. 5, pp. 922–923, 1976.

[10] C. Ericson, *Real-Time Collision Detection*. San Francisco, CA: Morgan Kaufmann, 2004.

[11] Gurobi Optimization, LLC, "Gurobi optimizer reference manual," 2025.

[12] Q. Huangfu and J. A. J. Hall, "Parallelizing the dual revised simplex method," *Mathematical Programming Computation*, vol. 10, pp. 119–142, Mar 2018.

[13] C. Jiang, J. Wang, J. Wallner, and H. Pottmann, "Freeform honeycomb structures," *Computer Graphics Forum*, vol. 33, no. 5, pp. 185–194, 2014.