Quasi-Symmetric Nets: A Constructive Approach to the Equimodular Elliptic Type of Kokotsakis Polyhedra

Abdukhomid Nurmatov[a], Mikhail Skopenkov[a], Florian Rist[a], Jonathan Klein[a], Dominik L. Michels[a,*]

[a] *KAUST Computational Sciences Group, Campus Building 1, Thuwal, 23955-6900, KSA*

APPENDIX

1. Coordinate Setup

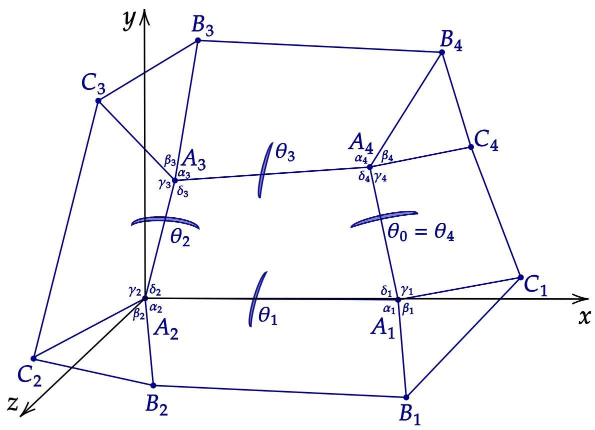

Figure 1: Figure 1: Illustration of a polyhedron with notation for the vertices, flat and dihedral angles in the A_2xyz frame.

Introduce a Cartesian coordinate system with the origin at A_2 such that the face $A_2A_3A_4A_1$ lies in the half-plane $y \geq 0$ of the xy-plane and the ray A_2A_1 is the x-axis as shown in Figure 1. Denote $a_1 := A_2A_1$, $a_2 := A_2A_3$. Then $a_3 := A_1A_4 = \frac{(a_2 \sin\delta_3 - a_1 \sin(\delta_2+\delta_3))}{\sin\delta_4}$, $a_4 := A_3A_4 = \frac{(a_1 \sin\delta_1 - a_2 \sin(\delta_1+\delta_2))}{\sin\delta_4}$, see Section 2. Thus we have

$$
\begin{aligned}
x_{A_1} &= x_{A_2} + a_1, & x_{A_3} &= x_{A_2} + a_2\cos\delta_2,\\
x_{A_2} &= 0, & x_{A_4} &= x_{A_1} - a_3\cos\delta_1,\\
y_{A_1} &= y_{A_2}, & y_{A_3} &= y_{A_2} + a_2\sin\delta_2,\\
y_{A_2} &= 0, & y_{A_4} &= y_{A_1} + a_3\sin\delta_1,\\
z_{A_1} &= z_{A_2}, & z_{A_3} &= z_{A_2},\\
z_{A_2} &= 0, & z_{A_4} &= z_{A_1}.
\end{aligned}
$$

*Corresponding author

Email addresses: abdukhomid.nurmatov@kaust.edu.sa (Abdukhomid Nurmatov), mikhail.skopenkov@gmail.com (Mikhail Skopenkov), florian.rist@kaust.edu.sa (Florian Rist), jonathan.klein@kaust.edu.sa (Jonathan Klein), dominik.michels@kaust.edu.sa (Dominik L. Michels)

Denote $b_1 := A_1B_1$, $b_2 := A_2B_2$, $b_3 := A_3B_3$, $b_4 := A_4B_4$. Then we have

$$\begin{aligned}
x_{B_1} &= x_{A_1} - b_1 \cos\alpha_1, & x_{B_3} &= x_{A_3} - b_3(\cos\alpha_3 \cos(\delta_2+\delta_3) + \sin\alpha_3 \cos\theta_3 \sin(\delta_2+\delta_3)),\\
x_{B_2} &= x_{A_2} + b_2 \cos\alpha_2, & x_{B_4} &= x_{A_4} + b_4(\cos\alpha_4 \cos(\delta_2+\delta_3) - \sin\alpha_4 \cos\theta_3 \sin(\delta_2+\delta_3)),\\
y_{B_1} &= y_{A_1} + b_1 \sin\alpha_1 \cos\theta_1, & y_{B_3} &= y_{A_3} - b_3(\cos\alpha_3 \sin(\delta_2+\delta_3) - \sin\alpha_3 \cos\theta_3 \cos(\delta_2+\delta_3)),\\
y_{B_2} &= y_{A_2} + b_2 \sin\alpha_2 \cos\theta_1, & y_{B_4} &= y_{A_4} + b_4(\cos\alpha_4 \sin(\delta_2+\delta_3) + \sin\alpha_4 \cos\theta_3 \cos(\delta_2+\delta_3)),\\
z_{B_1} &= z_{A_1} + b_1 \sin\alpha_1 \sin\theta_1, & z_{B_3} &= z_{A_3} + b_3 \sin\alpha_3 \sin\theta_3,\\
z_{B_2} &= z_{A_2} + b_2 \sin\alpha_2 \sin\theta_1, & z_{B_4} &= z_{A_4} + b_4 \sin\alpha_4 \sin\theta_3.
\end{aligned}$$

Now, denote $c_1 := A_1C_1$, $c_2 := A_2C_2$, $c_3 := A_3C_3$, $c_4 := A_4C_4$. Then we have

$$\begin{aligned}
x_{C_1} &= x_{A_1} - c_1(\cos\gamma_1 \cos\delta_1 + \sin\gamma_1 \cos\theta_4 \sin\delta_1), & x_{C_3} &= x_{A_3} - c_3(\cos\gamma_3 \cos\delta_2 - \sin\gamma_3 \cos\theta_2 \sin\delta_2),\\
x_{C_2} &= x_{A_2} + c_2(\cos\gamma_2 \cos\delta_2 + \sin\gamma_2 \cos\theta_2 \sin\delta_2), & x_{C_4} &= x_{A_4} + c_4(\cos\gamma_4 \cos\delta_1 - \sin\gamma_4 \cos\theta_4 \sin\delta_1),\\
y_{C_1} &= y_{A_1} + c_1(\cos\gamma_1 \sin\delta_1 - \sin\gamma_1 \cos\theta_4 \cos\delta_1), & y_{C_3} &= y_{A_3} - c_3(\cos\gamma_3 \sin\delta_2 + \sin\gamma_3 \cos\theta_2 \cos\delta_2),\\
y_{C_2} &= y_{A_2} + c_2(\cos\gamma_2 \sin\delta_2 - \sin\gamma_2 \cos\theta_2 \cos\delta_2), & y_{C_4} &= y_{A_4} - c_4(\cos\gamma_4 \sin\delta_1 + \sin\gamma_4 \cos\theta_4 \cos\delta_1),\\
z_{C_1} &= z_{A_1} + c_1 \sin\gamma_1 \sin\theta_4, & z_{C_3} &= z_{A_3} + c_3 \sin\gamma_3 \sin\theta_2,\\
z_{C_2} &= z_{A_2} + c_2 \sin\gamma_2 \sin\theta_2, & z_{C_4} &= z_{A_4} + c_4 \sin\gamma_4 \sin\theta_4.
\end{aligned}$$

2. Edge Generation

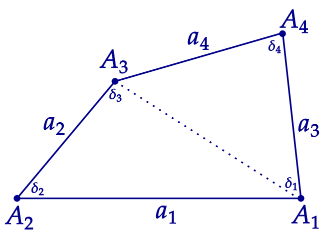

Figure 2: Figure 2: Illustration of a quadrilateral $A_1A_2A_3A_4$ with flat angles.

Given a_1 and angles $\delta_1 := \angle A_4A_1A_2$, $\delta_2 := \angle A_1A_2A_3$, $\delta_3 := \angle A_2A_3A_4 \in (0, \pi)$ we want to generate suitable a_2, a_3 and a_4 so that the quadrilateral $A_1A_2A_3A_4$ is convex. See Figure 2.

Consider $\triangle A_1A_2A_3$ and $\triangle A_1A_3A_4$. Denote $\angle A_2A_3A_1 := \delta_{31}$ and $A_1A_3 := x$. By law of sines for $\triangle A_1A_2A_3$ we get

$$\frac{a_1}{\sin\delta_{31}} = \frac{x}{\sin\delta_2} \Rightarrow \sin\delta_{31} = \frac{a_1 \sin\delta_2}{x}. \tag{B.1}$$

By law of cosines for $\triangle A_1A_2A_3$ we get

$$\begin{cases} a_1^2 = a_2^2 + x^2 - 2a_2x\cos\delta_{31} \\ x^2 = a_1^2 + a_2^2 - 2a_1a_2\cos\delta_2 \end{cases} \Rightarrow \cos\delta_{31} = \frac{a_2 - a_1\cos\delta_2}{x}. \tag{B.2}$$

By law of sines for $\triangle A_1A_3A_4$ we get

$$\frac{a_3}{\sin(\delta_3 - \delta_{31})} = \frac{x}{\sin\delta_4} \tag{B.3}$$

Using (B.1) and (B.3) we get

$$\sin\delta_3 \frac{\cos\delta_{31}}{\sin\delta_{31}} - \cos\delta_3 = \frac{a_3 \sin\delta_4}{a_1 \sin\delta_2}.$$

Using (B.1) and (B.2) we get

$$a_3 = \frac{a_2 \sin \delta_3 - a_1 \sin(\delta_2 + \delta_3)}{\sin \delta_4}.$$

Similarly, we get

$$a_4 = \frac{a_1 \sin \delta_1 - a_2 \sin(\delta_1 + \delta_2)}{\sin \delta_4}.$$

Now, we have to impose the following conditions:

$$a_1 > 0, \quad a_2 > 0, \quad a_3 > 0, \quad a_4 > 0.$$

Thus, we get the following cases:

1. If $\delta_2 + \delta_3 \geq \pi$ then condition $a_3 > 0$ is satisfied. Thus, we need to check only condition for a_4:

   (a) If $\delta_1 + \delta_2 \geq \pi$ then condition $a_4 > 0$ is satisfied. This means that we can choose a_2 and a_1 as we want. For example, $a_2 = a_1$.

   (b) If $\delta_1 + \delta_2 < \pi$ then from condition $a_4 > 0$ we get

   $$a_2 < a_1 \frac{\sin \delta_1}{\sin(\delta_1 + \delta_2)}. \tag{B.4}$$

   This means that given a_1 we can choose a_2 such that (B.4) holds true, i.e. $a_2 = ka_1 \frac{\sin \delta_1}{\sin(\delta_1 + \delta_2)}$, for some $0 < k < 1$. For example, $a_2 = 0.8a_1 \frac{\sin \delta_1}{\sin(\delta_1 + \delta_2)}$.

2. If $\delta_2 + \delta_3 < \pi$ then from condition $a_3 > 0$ we get

   $$a_2 > a_1 \frac{\sin(\delta_2 + \delta_3)}{\sin \delta_3}. \tag{B.5}$$

   Now, we need to check condition for a_4:

   (a) If $\delta_1 + \delta_2 \geq \pi$ then condition $a_4 > 0$ is satisfied. This means that given a_1 we can choose a_2 such that (B.5) holds true, i.e. $a_2 = ka_1 \frac{\sin(\delta_2 + \delta_3)}{\sin \delta_3}$, for some $k > 1$. For example, $a_2 = 1.2a_1 \frac{\sin(\delta_2 + \delta_3)}{\sin \delta_3}$.

   (b) If $\delta_1 + \delta_2 < \pi$ then from condition $a_4 > 0$ we get

   $$a_2 < a_1 \frac{\sin \delta_1}{\sin(\delta_1 + \delta_2)} \tag{B.6}$$

   This means that given a_1 we should choose a_2 such that (B.5) and (B.6) hold true. Introduce $l := \frac{\sin(\delta_2 + \delta_3)}{\sin \delta_3}$ and $r := \frac{\sin \delta_1}{\sin(\delta_1 + \delta_2)}$ then we should have $a_1 l < a_2 < a_1 r$. Note that $l \geq r$ is impossible for convex quadrilaterals when $\delta_1 + \delta_2 < \pi$ and we should raise error in such case. Indeed, since our quadrilateral is convex it follows that

   $$2 \sin \delta_2 \sin \delta_4 > 0,$$

and hence

$$\cos\delta_2\cos\delta_4 + \sin\delta_2\sin\delta_4 > \cos\delta_2\cos\delta_4 - \sin\delta_2\sin\delta_4.$$

Thus

$$\cos(\delta_2 - \delta_4) > \cos(\delta_2 + \delta_4).$$

Since $\delta_4 = 360 - \delta_1 - \delta_2 - \delta_3$ it follows that

$$\cos(\delta_1 + 2\delta_2 + \delta_3) > \cos(\delta_1 + \delta_3),$$

and

$$-\cos(\delta_1 + 2\delta_2 + \delta_3) < -\cos(\delta_1 + \delta_3).$$

Hence

$$\cos(\delta_2 + \delta_3 - \delta_1 - \delta_2) - \cos(\delta_1 + \delta_2 + \delta_2 + \delta_3) < \cos(\delta_3 - \delta_1) - \cos(\delta_1 + \delta_3).$$

Thus

$$2\sin(\delta_2 + \delta_3)\sin(\delta_1 + \delta_2) < 2\sin\delta_1\sin\delta_3.$$

Since $\delta_1 + \delta_2 < \pi$ it finally follows that

$$\frac{\sin(\delta_2 + \delta_3)}{\sin\delta_3} < \frac{\sin\delta_1}{\sin(\delta_1 + \delta_2)},$$

i.e., we always have $l < r$ when $\delta_1 + \delta_2 < \pi$.

Thus we should choose $a_2 = a_1(kl + (1-k)r)$, for some $0 < k < 1$. For example, $a_2 = a_1\dfrac{l+r}{2}$.

3. Angle Generation

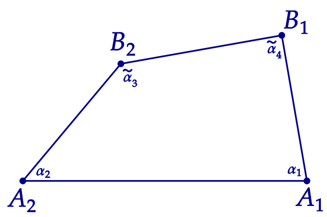

Figure 3: Figure 3: Illustration of a quadrilateral $A_1A_2B_2B_1$ with flat angles.

Given α_1, $\alpha_2 \in (0, \pi)$ we want to generate suitable $\widetilde{\alpha}_3$, $\widetilde{\alpha}_4 \in (0, \pi)$ so that quadrilateral $A_1A_2B_2B_1$ is convex. See Figure 3.

We have

$$\widetilde{\alpha}_3 + \widetilde{\alpha}_4 = 360 - \alpha_1 - \alpha_2 \tag{C.1}$$

From (C.1) we conclude that $\widetilde{\alpha}_3 < 360 - \alpha_1 - \alpha_2$. On the other hand $0 < \widetilde{\alpha}_3 < 180$. Thus, we have the following cases:

1. If $360 - \alpha_1 - \alpha_2 \geq 180$, i.e. $\alpha_1 + \alpha_2 \leq 180$ then condition on $\widetilde{\alpha}_3$ is such that $0 < \widetilde{\alpha}_3 < 180$. On the other hand, since $\widetilde{\alpha}_4 = 360 - \alpha_1 - \alpha_2 - \widetilde{\alpha}_3$ and $0 < \widetilde{\alpha}_4 < 180$ then we should choose $\widetilde{\alpha}_3$ such that

$$180 - \alpha_1 - \alpha_2 < \widetilde{\alpha}_3 < 180.$$

   Thus, $\widetilde{\alpha}_3 = k \cdot (180 - \alpha_1 - \alpha_2) + (1 - k) \cdot 180 = 180 - k \cdot (\alpha_1 + \alpha_2)$, for some $0 < k < 1$. For example, $\widetilde{\alpha}_3 = \dfrac{360 - \alpha_1 - \alpha_2}{2}$.

2. If $360 - \alpha_1 - \alpha_2 < 180$, i.e. $\alpha_1 + \alpha_2 > 180$ then condition on $\widetilde{\alpha}_3$ is such that $0 < \widetilde{\alpha}_3 < 360 - \alpha_1 - \alpha_2$. On the other hand, since $\widetilde{\alpha}_4 = 360 - \alpha_1 - \alpha_2 - \widetilde{\alpha}_3$ and $0 < \widetilde{\alpha}_4 < 180$ then we should choose $\widetilde{\alpha}_3$ such that

$$0 < \widetilde{\alpha}_3 < 360 - \alpha_1 - \alpha_2.$$

   Thus, $\widetilde{\alpha}_3 = k \cdot (360 - \alpha_1 - \alpha_2)$, for some $0 < k < 1$. For example, $\widetilde{\alpha}_3 = \dfrac{360 - \alpha_1 - \alpha_2}{2}$.

Figure 4 is a figure with denoted angles and edges which is used as a reference for easy construction of a polyhedron in our code (see visualization Python scripts in the example folders).

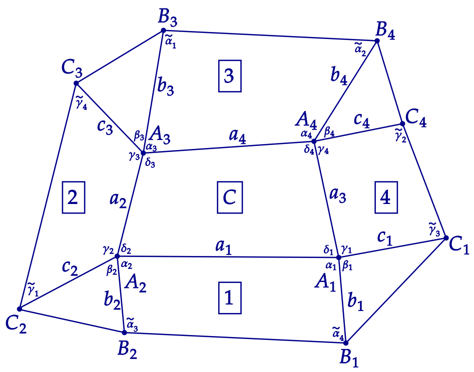

Figure 4: Figure 4: Illustration of a polyhedron with notation for vertices, edge lengths, flat angles, and quadrilateral labels.

4. Visualization

Having established the existence criterion and a numerical method to find solutions, we now describe a constructive and reproducible procedure to *build* and *visualize* the corresponding polyhedra. The input is the full set of flat and dihedral angles from any verified solution tuple.

Constructing a polyhedron from prescribed flat and dihedral angles is nontrivial. A naive approach – blindly guessing edge lengths until all faces become convex – is time-consuming and numerically fragile. Instead, we propose a deterministic pipeline that requires only the flat angles and a single user-specified base edge (e.g., A_2A_1) and guarantees convex faces; see Section 2. With this, the central quadrilateral $A_1A_2A_3A_4$ is constructed directly from $\{\delta_i\}_{i=1}^4$ and the base edge. For adjacent faces (e.g., $A_1A_2B_2B_1$), certain interior angles (such as $\tilde{\alpha}_3$, $\tilde{\alpha}_4$; cf. Fig. 3) must be chosen carefully to preserve convexity. Section 3 provides a principled selection rule that ensures a convex configuration without trial-and-error. In summary, Sections 2 and 3 replace ad hoc edge guessing by a reliable construction that, given flat angles, dihedral angles, and one base edge, produces a polyhedron with convex faces.

The algorithmic construction places the base quadrilateral $A_1A_2A_3A_4$ in the xy-plane and then attaches the four surrounding quadrilaterals in sequence. Once all face geometries are fixed, the 3D coordinates of the twelve vertices are computed from closed-form expressions (Section 1). Also, note that in the *static construction* we do not require the angles $\{\beta_i\}$ as inputs: once $\{\alpha_i\}$, $\{\gamma_i\}$, $\{\delta_i\}$ and $\{\theta_i\}$ are given, the $\{\beta_i\}$ are *uniquely determined*.

To demonstrate flexibility, we provide an interactive visualization (Matplotlib) that varies a primary dihedral angle (e.g., θ_1) continuously. For each slider value, the remaining dihedral angles are recomputed by solving the Bricard's equations. If Bricard's equations becomes inconsistent under perturbation, the configuration is rigid and the display remains unchanged. For a flexible configuration, solutions to Bricard's equations exist on an non-empty interval of θ_1; realizability in $\mathbb{R}^3$ further requires that all dihedral angles remain real. Our visualization tool implements this subsequent check. If the chosen value of a primary angle yields real solutions for all dependent angles, the new set of dihedral angles satisfies Bricard's equations. Consequently, the existence of a valid polyhedron is guaranteed for this new configuration, and the model is re-rendered. Critically, all edge lengths and flat angles are also recalculated at each step to ensure that the polyhedron's flexion preserves the congruence of all faces within a desired numerical tolerance. If the user moves the primary angle beyond this range of real flexion, the visualization remains at the last valid configuration, effectively demonstrating the boundaries of its motion in real space.