ACM Communications in Computer Algebra, Vol. 56, No. 2, Issue 220, June 2022

Extended Abstract for: Solving Rupert's problem algorithmically[*]

Jakob Steininger, Sergey Yurkevich

1 Introduction

Undoubtedly the following fact is surprising when being first encountered with:

It is possible to cut a hole in the unit cube such that another unit cube can pass through it.

Indeed, Prince Rupert of the Rhine won a wager in the 17th century by betting on the validity of this claim. An elegant and simple way to see why this assertion is true is presented in Figure 1: it is easy to verify that the projection of the unit cube in the direction of a main diagonal yields a regular hexagon of side length $\sqrt{2/3}$ and the unit square (a different projection of the cube) fits inside that hexagon. These two observations are already enough to win Rupert's bet; at the same time they also open a whole world of interesting questions, conjectures and studies.

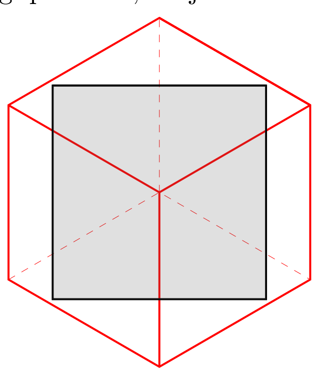

Figure 1: The unit square fits inside the regular hexagon of side length $\sqrt{2/3}$.

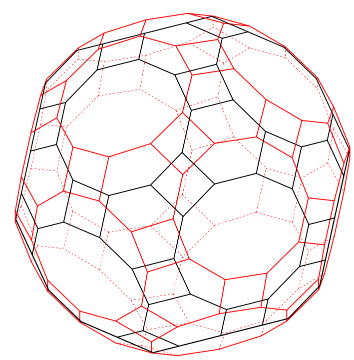

Figure 2: Two projections of the same truncated icosidodecahedron, one inside the other.

Analogously to the cube, Rupert's property can be defined for any polyhedron. A somewhat imprecise definition of this property is as follows: a (convex) polyhedron $\mathbf{P} \subset \mathbb{R}^3$ is *Rupert* (or has *Rupert's property*) if a hole (with the shape of a straight tunnel) can be cut into it such that a copy of $\mathbf{P}$ can be moved through this hole. *Rupert's problem* is the task to decide whether a given polyhedron has Rupert's property.

Scriba showed in 1968 that the tetrahedron and octahedron have Rupert's property [8]. Half a century later and hence already quite recently, Jerrard, Wetzel and Yuan built on Scriba's work and investigated Rupert's property of Platonic solids further: they could prove that all five of them are Rupert [5]. One year later Chai, Yuan and Zamfirescu looked at Archimedean solids from "Rupert's perspective" and showed that 8 out of 13 have Rupert's property [1]. In the same articles [5, 1] the possible non-existence of "non-Rupert" convex polyhedra in $\mathbb{R}^3$ is formulated as a conjecture. Another year later, Hoffmann [4] and Lavau [6] showed in 2019 Rupert's property for the truncated tetrahedron, thus enhancing the number to 9 out of 13. In 2021 the authors of the present text proposed an algorithmic approach for Rupert's problem:

[*]This extended abstract for a poster (short communication) at the ISSAC conference 2022 summarizes a selection of findings from the recent preprint "An algorithmic approach to Rupert's problem" [9] by the same authors.

they showed that it is algorithmically decidable for polyhedra with algebraic coordinates and also designed a probabilistic algorithm which can efficiently prove that a given polyhedron is Rupert [9]. They "solved" the truncated icosidodecahedron (Figure 2), pushing the number of settled Archimedean solids to 10. The current text summarizes and presents some of the findings of the latter work.

A natural extension of Rupert's problem concerns finding the optimal solution for given polyhedra. It was first investigated by Nieuwland who could prove that the cube with side length $3\sqrt{2}/4 \approx 1.06$ can be moved through a hole of the unit cube, and that this number is optimal. Conjecturally, the same number seems to be achieved by the optimal solution of the octahedron. Moreover, recent findings of the authors suggest that the dodecahedron and icosahedron both have solutions (Figure 3 and Figure 4) with optimal "scaling factor" ≈ 1.0108 which is a root of $P(x) = 2025x^8 - 11970x^6 + 17009x^4 - 9000x^2 + 2000$. The connection between duality and Rupert's property is evident but yet to be fully understood.

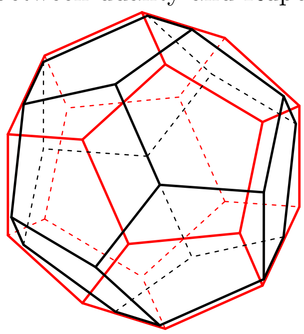

Figure 3: The dodecahedron is Rupert.

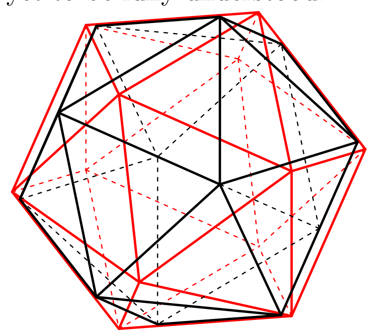

Figure 4: The icosahedron is Rupert.

2 Probabilistic algorithm

Contrary to the existing methods for proving that a polyhedron has Rupert's property, we present a new probabilistic and algorithmic approach to Rupert's problem in the recent preprint [9]. The main idea of the first and naive version of our algorithm is already clearly visible in Figure 1: a polyhedron $\mathbf{P}$ is Rupert if and only if there exist two projection matrices $M_{\theta_1,\varphi_1}, M_{\theta_2,\varphi_2} : \mathbb{R}^3 \to \mathbb{R}^2$, a rotation matrix $R_\alpha : \mathbb{R}^2 \to \mathbb{R}^2$ and a translation map $T_{x,y} : \mathbb{R}^2 \to \mathbb{R}^2$ such that the polygon $\mathcal{P} = (T_{x,y} \circ R_\alpha \circ M_{\theta_1,\varphi_1})(\mathbf{P})$ lies strictly inside the polygon $\mathcal{Q} = M_{\theta_2,\varphi_2}(\mathbf{P})$. It follows that any solution of Rupert's problem can be translated into the 7 parameters $(x, y, \alpha, \theta_1, \theta_2, \varphi_1, \varphi_2)$ and vice versa. Moreover, one can easily bound all the parameters: $\alpha, \theta_1, \theta_2 \in [0, 2\pi)$, $\varphi_1, \varphi_2 \in [0, \pi]$ and $x, y \in [-R, R]$, where $R \in \mathbb{R}$ is the circumference radius of $\mathbf{P}$. Finally, it is algorithmically not difficult to decide whether a polygon $\mathcal{P}$ lies inside a polygon $\mathcal{Q}$. We arrive at the *Las Vegas* type algorithm below. Note that this algorithm is *approximately complete* for Rupert polyhedra but *essentially incomplete* in general. In other words, the algorithm will find a solution *eventually* it it exists, but cannot prove the non-existence of a solution.

Algorithm 1
Input: A polyhedron $\mathbf{P} = \{P_1, \ldots, P_n\} \subset \mathbb{R}^3$.
Output: The solution encoded by $(x, y, \alpha, \theta_1, \theta_2, \varphi_1, \varphi_2) \in \mathbb{R}^7$ if $\mathbf{P}$ is Rupert.

(1) Find the circumference radius R of $\mathbf{P}$. Draw x and y uniformly in $[-R, R]$, θ_1, θ_2 and α uniformly in $[0, 2\pi)$, and φ_1, φ_2 uniformly in $[0, \pi]$.

(2) Construct two 3×2 matrices A and B corresponding to the linear maps $R_\alpha \circ M_{\theta_1,\varphi_1}$, M_{θ_2,φ_2}. Compute two projections of $\mathbf{P}$ given by $\mathcal{P}' := T_{x,y}(A \cdot \mathbf{P}) = A \cdot \mathbf{P} + (x, y)$ and $\mathcal{Q}' := B \cdot \mathbf{P}$.

(3) Find vertices on the convex hulls of $\mathcal{P}'$ and $\mathcal{Q}'$; denote them by $\mathcal{P}$ and $\mathcal{Q}$.

(4) Decide whether $\mathcal{P}$ lies inside of $\mathcal{Q}$ by checking each vertex of $\mathcal{P}$.

(5) If step (4) yields a True, return $(x, y, \alpha, \theta_1, \theta_2, \varphi_1, \varphi_2)$. Otherwise, repeat steps (1)-(5).

Already this very simple algorithm is able to find solutions for many polyhedra. However, in practice, it is quite slow, mostly because the 7-dimensional search space for $(x, y, \alpha, \theta_1, \theta_2, \varphi_1, \varphi_2)$ is large. The first and most significant improvement to it is to reduce the parameter search space from $\mathbb{R}^7$ to $\mathbb{R}^4$ by algorithmically finding x, y and α for given $\theta_1, \theta_2, \varphi_1, \varphi_2$. For that purpose we can use Chazelle's algorithm [2] for deciding the polygon containment under translation and rotation. Another natural practical improvement to the resulting algorithm is to discard pairs $(\mathcal{P}, \mathcal{Q})$ if it can be algorithmically easily seen that $\mathcal{P}$ cannot fit inside $\mathcal{Q}$. Indeed, we can do so by first computing elementary geometric invariants of the polygons like the perimeter, area and diameter. Moreover, these invariants can be computed for a large batch of polygons coming from randomly drawn projections; then we can discard most pairs and need to test only the remaining ones.

As an example, our algorithm can prove the following theorem in a few seconds on a regular PC, settling down a previously unsolved Archimedean solid:

Theorem 1. *The truncated icosidodecahedron has Rupert's property.*

A pictorial proof corresponding to the found parameters $x = y = 0, \alpha = 0.4358, \theta_1 = 2.7769, \theta_2 = 2.0941, \varphi_1 = 0.7906, \varphi_2 = 2.8967$ is presented in Figure 2. Running the probabilistic algorithm on a collection of famous solids and verifying afterwards rigorously, we obtain the following result, after a few minutes of computation on a regular laptop:

Theorem 2. *It holds that:*

1. *All 5 Platonic solids are Rupert.*
2. *At least 10 out of 13 Archimedean solids are Rupert.*
3. *At least 9 out of 13 Catalan solids are Rupert.*
4. *At least 82 out of 92 Johnson solids are Rupert.*

We suspect that the remaining polyhedra do not admit Rupert's property. Especially for the rhombicosidodecahedron we can argue statistically and heuristically to state the following conjecture with quite some confidence:

Conjecture 1. *The rhombicosidodecahedron does not have Rupert's property.*

3 Deterministic algorithm

Even though the probabilistic algorithm presented above works quite well in practice, it cannot disprove Rupert's property. For this purpose we designed a deterministic algorithm by showing that Rupert's problem can be reformulated in the decision problem of the emptiness of semi-algebraic sets. This classical problem is known to be solved [10] and we may rely on quite efficient algorithms for it [3, 7].

We call the cycle $(s_1, \dots, s_k)$ the *silhouette* of a polyhedron $\mathbf{P} = \{P_1, \dots, P_n\} \in \mathbb{R}^3$ under some projection $M_{\theta,\varphi}$ if $P_{s_1}, \dots, P_{s_k}$ is the boundary of $M_{\theta,\varphi}(\mathbf{P})$. By S_n we denote the set of all such non-empty cycles of the numbers from 1 to n.

The deterministic algorithm can decide whether there exists a solution to Rupert's problem

$$(T_{x,y} \circ R_\alpha \circ M_{\theta_1,\varphi_1})(\mathbf{P}) \subset M_{\theta_2,\varphi_2}(\mathbf{P})^\circ$$

for any possible silhouette $s \in S_n$ of the projection on the right-hand side. Then the full algorithm runs over all elements of S_n.

Let $x, y, \alpha, \theta_1, \varphi_1, \theta_2, \varphi_2$ be variables. Given a silhouette $s = (s_1, s_2, \ldots, s_k)$, let $Q_i \coloneqq M_{\theta_2,\varphi_2}(\mathbf{P}_{s_i})$ and $P_j \coloneqq (T_{x,y} \circ R_\alpha \circ M_{\theta_1,\varphi_1})(\mathbf{P}_j)$ for $i = 1, \ldots, k$ and $j = 1, \ldots, n$. We also set $Q_{k+1} \coloneqq Q_1$. In other words, $Q_1, \ldots, Q_k$ denote the vertices on the boundary of $M_{\theta_2,\varphi_2}(\mathbf{P})^\circ$ given a solution with silhouette s. We define the system of kn inequalities in the seven unknowns $x, y, \alpha, \theta_1, \varphi_1, \theta_2, \varphi_2$:

$$\det(Q_{s_{i+1}} - P_j, Q_{s_i} - P_j) > 0 \qquad j = 1, \ldots, n \text{ and } i = 1, \ldots, k. \tag{1}$$

The main observation is that (1) has a solution $(x, y, \alpha, \theta_1, \varphi_1, \theta_2, \varphi_2)$ if and only if there exists a solution to Rupert's problem with silhouette s. Moreover, the system (1) is algebraic in x, y and the trigonometric values of the other variables. Hence, the rational parametrization of the circle $t \mapsto \left(\frac{1-t^2}{1+t^2}, \frac{2t}{1+t^2}\right)$ and a (careful) clearance of denominators transforms (1) into a system of polynomial inequalities. This proves:

Theorem 3. *Let $\mathbf{P}$ be a convex polyhedron with algebraic coordinates. There exists a deterministic algorithm deciding whether $\mathbf{P}$ is Rupert and finding a solution if it exists. If moreover $\mathbf{P}$ has n vertices with integers coordinates bounded in absolute value by m, then the running time of this algorithm is at most $(\log(m) \cdot n)^{O(1)} \cdot n!$.*

In practice, it seems that the polynomial systems one (naively) obtains already from quite small polyhedra are too big to be solved with existing methods. For example, for the rhombicosidodecahedron rough estimates imply that we would need to prove emptiness of 50 semi-algebraic sets each defined by 3600 polynomial inequalities in 6 variables of total degree 22. Of course, the ongoing task is to reduce this number significantly by exploiting symmetries in the system coming from its definition and the symmetries of the polyhedron.

References

[1] Y. Chai, L. Yuan, and T. Zamfirescu. Rupert property of Archimedean solids. *Amer. Math. Monthly*, 125(6):497–504, 2018.

[2] B. Chazelle. The polygon containment problem. *Advances in Computing Research I*, pages 1–33, 1983.

[3] D. Y. Grigor'ev and N. Vorobjov. Solving systems of polynomial inequalities in subexponential time. *Journal of Symbolic Computation*, 5(1):37–64, 1988.

[4] B. Hoffmann. Rupert properties of polyhedra and the generalised Nieuwland constant. *J. Geom. Graph.*, 23(1):29–35, 2019.

[5] R. P. Jerrard, J. E. Wetzel, and L. Yuan. Platonic passages. *Math. Mag.*, 90(2):87–98, 2017.

[6] G. Lavau. The truncated tetrahedron is Rupert. *Amer. Math. Monthly*, 126(10):929–932, 2019.

[7] M. Safey El Din and E. Schost. Properness defects of projections and computation of at least one point in each connected component of a real algebraic set. *Discrete Comput. Geom.*, 32(3):417–430, 2004.

[8] C. J. Scriba. Das Problem des Prinzen Ruprecht von der Pfalz. *Praxis Math.*, 10(9):241–246, 1968.

[9] J. Steininger and S. Yurkevich. An algorithmic approach to Rupert's problem, 2021. Preprint, https://arxiv.org/abs/2112.13754.

[10] A. Tarski. *A decision method for elementary algebra and geometry.* University of California Press, Berkeley-Los Angeles, Calif., 1951. 2nd ed.