Squaring the circle: polyhedral approximation of convex sets

Work-in-progress poster

Maxim Demenkov

Institute of Control Sciences, Moscow, Russia

JuliaCon
29—31 July 2020

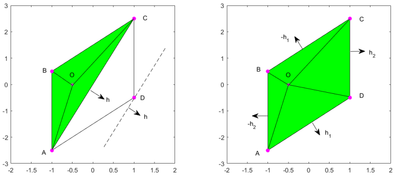
In 1982, a Soviet mathematician Mukhamediev[1] proposed the basic idea. Consider convex and compact set $S \in \mathbb{R}^n$. For an inner polytopic approximation $P_k \subseteq S$, its description in the form of collections of its vertices $\{x_1, ..., x_N\}$ and normals to its facets $\{h_1, ..., h_M\}$ is maintained. On each iteration, new points are given by the following *linear oracle*:

$$z_i = \arg\max_{x \in S} < h_i, x >, \ i = \overline{1, M}.$$

After that, a new point z_i that is outside of the current approximation is added to the convex hull of similar points:

$$P_{k+1} = \mathrm{conv}\{z_i, x_1, ..., x_N\}$$

and the collection of facet normals is changed accordingly.



The algorithm is universal for any convex and compact set, moreover, if the set is a polytope, it converges to the polytope itself (see in the picture ABCD polygon as S and ABC triangle as P_0):

$$P_0 \subseteq P_1 \subseteq ... \subseteq P_k \equiv S.$$

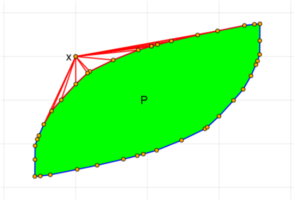
Using this fact, Bushenkov[2] applied the same method to compute an orthogonal projection of polytopes.

[1] Mukhamediev, B.M.: Approximate method of solving concave programming problems. USSR Comput. Math. Math. Phys. **22**(3), 238–245 (1982)

[2] Bushenkov, V.A.: An iteration method of constructing orthogonal projections of convex polyhedral sets. USSR Comput. Math. Math. Phys. **25**(5), 1–5 (1985)

Connections with computational geometry

The algorithm can be viewed as a variant of the beneath-beyond method developed for convex hull computations in computer geometry[3] during the 1980s in the West (or used e.g. for a proof in the theory of polytopes[4]). A subtle difference is that Soviet scientists had considered[5] *any* convex set, while their Western counterparts had tailored their algorithms for a collection of points.



- One of common issues is the *visibility cone* construction (red in the picture) while adding a new point x. We say that a facet of P is visible (from x) if its supporting hyperplane separates P and x. But, with increasing of the number of polytope facets, we have more and more facets with normals almost orthogonal to the "line of sight" (in red) from the point x, and also high number of almost coplanar facets.
- As a result, reliable computations using these algorithms were not possible beyond 6-7 or maximum 9 dimensions, with polytopes loosing even convexity and in need of special correction after attaching a point (e.g. "fat facets" in QuickHull).

[3] Seidel, R.: Convex hull computations. In: Handbook of Discrete and Computational Geometry, Goodman, J.E., O'Rourke, J. and Tóth, C.D. (eds.) 3rd edition, CRC Press, Boca Raton, FL, 2017

[4] Joswig, M.: Beneath-and-beyond revisited. In: Joswig, M., Takayama, N., (eds.) Algebra, Geometry, and Software Systems, pp. 1–21. Springer, Berlin, 2003

[5] Kamenev, G.K.: Methods for polyhedral approximation of convex bodies and their application for the construction and analysis of generalized reachable sets. PhD thesis (in Russian), 1986

- Our very preliminary program is available as PolyAnn.jl at `https://github.com/mdemenkov/polyann`. It requires the linear oracle implemented as a method in Julia for a specific convex set (e.g. linear programming for a system of linear inequalities). The main program is totally independent from the particularities of different sets. This is multiple dispatch at its best.
- The program name stems from *"polyhedral annexation"*[6], which is one possible name for our method given in the context of its application in *concave programming* [7]:

$$\min_{x \in S} -f(x),$$

where $f(x)$ is a convex function and S is a convex set. The solution of the concave programming problem (which is nonconvex) over a polyhedral approximation $P \subseteq S$ should exist on its vertices.

- Our approach for circumventing the dimensionality curse is to avoid the visibility cone construction. Instead, we add each time not only one, but all new unique points and maintain a collection of corresponding simplicial cones (i.e. every cone is given by exactly n rays in $\mathbb{R}^n$). This method have its own problems of course. But, the convex hull implementation available in Julia (and Python) is the wrapper for the same old QuickHull algorithm from 1996, which seems a little bit outdated novadays...

[6] Tuy, H.: Convex analysis and global optimization. 2nd edn. Springer Optimization and Its Applications, vol. 110, 2016

[7] Benson, H.P.: Concave minimization: theory, applications and algorithms. In: Horst, R., Pardalos, P.M. (eds.) Handbook of global optimization, pp. 43–148. Springer, Dordrecht, 1995