H-Polytopity Mathematical Documentation

by Milo Moses and Lucien Werner

May 4, 2021

Contents

1 Introduction

When coding virtual environments, the standard way of representing polyhedra in $\mathbb{R}^n$ has always been to store just the locations of their verticies and do all computations with these as their starting point. In this project, we seek to create a virtual environment using the equivilant H-polytope definition of convex polyhedra given by Weyl and Minkowski, which is as the intersection of a finite number of closed half spaces. The classical definition of convex polygon and this alternative definition of Weyl Minkowski is well known to be equivilant, as shown in [1] for example.

Both the V polytope and H polytope data will be carried around for every polygon. This will be at an initial cost for every polygon created since computing the H polytope representation from the V polytope one is not always trivial and may require the Fourier-Motskin algorithm, but it will yield many savings in other places where the optimal algorithms run off of the H polytope data.

Figure 1 is a table of the different algorithms used in the project, and which representation of the data was used in the computation. Cited next to each conclusion is a reference to the section where further discussion is found.

Algorithm	Representation
Point in Polygon	H Polytope (2)

2 Point in Polygon algorithms

One of the first (and most basic) algorithms implemented is the classic Point in Polygon (PIP) algorithm which asks whether or not a given point is inside a given polygon. While many different algorithms are used, as of 2021 modern consensus seems to be that for many classes of polyhedra the Winding Number Algorithm is the fastest[2].

While the general Winding Number Algorithm for PIP is more complicated, in the case of convex polyhedra it reduces to checking on what side of each of its bounding half planes it is in and then returning true on the case of it being inside all of the half planes. As can be clearly seen from the statement of this algorithm, it is faster when working directly with H polyhedra.

References

[1] Jean Gallier. Polyhedra and polytopes. *Advanced Geometric Methods in Computer Science Course Notes*, 4:49–65, 2003.

[2] Anirudh Topiwala. Is the point inside the polygon? *towards data science*, 2020.