Exact 2D Convex Hull for floating-point data

K. Ozaki[1)], **T. Ogita**[2,1)], and **S. Oishi**[1,3)]

[1)]Faculty of Science and Engineering, Waseda University,
3-4-1 Okubo, Shinjyuku-ku, Tokyo 169-0072, k_ozaki@aoni.waseda.jp

[2)]Department of Mathematical Sciences, Tokyo Woman's Christian University,
2-6-1 Zempukuji, Suginami-ku, Tokyo 167-8585, Japan, ogita@lab.twcu.ac.jp

[3)]CREST, Japan Science and Technology Agency

Keywords: *Computational Geometry; Floating-point Arithmetic; Accurate Computations.*

Abstract

In this paper, we discuss how to develop an algorithm which outputs an exact convex hull for a set of points on two-dimensional space. Assuming that all coordinates are expressed by a floating-point number defined by IEEE 754. Generating the convex hull, a two-dimensional orientation problem must be solved many times. Here is an explanation of this problem: Suppose that there are an oriented line and a point on two-dimensional space. It is required to know which the the point is left or right to the oriented line. This problem can be boiled down to a sign of a 3-by-3 matrix determinant. If floating-point arithmetic is used to solve this problem, then an incorrect sign may be obtained due to accumulation of rounding errors. By using the incorrect information, the algorithm has prospects of outputting an inexact convex hull. For example, the result is not a convex, or some points are not enclosed by the result (see Kettner et al. (2008)).

Our aim is to obtain the exact convex hull as fast as possible. If methods in Shewchuk (1997); Ozaki et al. (2009) are used, then the correct sign of the matrix determinant can be obtained. These are called 'verification methods' for two-dimensional orientation problem. By using these methods, the algorithms for the convex hull can output the correct result. In this paper, we specialize the verification method into the algorithm of the convex hull. Our algorithm takes heed of problems of overflow and underflow in floating-point arithmetic so that it always outputs the correct convex hull for any set of floating-point numbers.

Finally, we present numerical examples to illustrate the efficiency of the proposed method.

References

L. Kettner, K. Mehlhorn, S. Pion, S. Schirra, C. Yap. Classroom Examples of Robustness Problems in Geometric Computations. *Computational Geometry*, 40:61–78, 2008.

J. R. Shewchuk. Adaptive precision floating-point arithmetic and fast robust geometric predicates. *Discrete & Computational Geometry*, 18: 305–363, 1997 .

K. Ozaki, T. Ogita, S. M. Rump, S. Oishi. Adaptive and Efficient Algorithm for 2D Orientation Problem. *Japan Journal of Industrial and Applied Mathematics* , to appear.