Verified Numerical Computations for Computational Geometry

Katsuhisa Ozaki[†,‡]

† College of Systems Engineering and Sciences,
Shibaura Institute of Technology
307 Fukasaku, Minuma-ku, Saitama-shi, Saitama 337-8570, Japan

‡ JST, CREST
ozaki@sic.shibaura-it.ac.jp

Keywords: sparse linear systems, floating-point arithmetic, verified numerical computations

We propose new semi-static floating-point filters for computational geometry. We focus on a two-dimensional orientation problem which is one of the basic problems in computational geometry. Suppose that an oriented line and a point $C = (c_x, c_y)$ in the two-dimensional Euclidean space are given. The oriented line passes from a point $A = (a_x, a_y)$ to a point $B = (b_x, b_y)$ for $A \neq B$. The aim is to judge whether the point C is located on the left or the right of the oriented line, or on the line. This problem can be boiled down to determining the sign of a 3-by-3 matrix determinant as follows:

$$\mathtt{sgn}(\det(G)),\ G := \begin{pmatrix} a_x & a_y & 1 \\ b_x & b_y & 1 \\ c_x & c_y & 1 \end{pmatrix}. \tag{1}$$

If the sign of the determinant is positive / negative, then the point is left / right of the oriented line. If the sign is zero, then the point is on the line.

If all coordinates are represented by floating-point numbers and if the determinant (1) is evaluated by floating-point arithmetic, then an incorrect sign may be obtained due to accumulation of rounding errors. Once the incorrect sign is computed, algorithms for computational geometry have prospects of producing an inexact result. Such problems are called 'robustness problems'. Plentiful topics of robustness problems in computational geometry are introduced in [1].

If we use multi-precision arithmetic with sufficient precision or symbolic computations, it is possible to obtain the correct sign of the determinant (1). However, it is basically rare to handle ill-conditioned problems. In addition, the cost for using multi-precision arithmetic or symbolic computations is very expensive. To overcome these problems, it is preferred that so-called 'floating-point filters' are applied first. Such a filter quickly checks whether a sufficient condition for the correctness of the sign of the determinant is satisfied or not. Therefore, the filter answers 'the sign of the computed result is correct' or 'the correctness of the computed result is unknown'. If the filter cannot guarantee correctness of the computed sign, more accurate algorithms can be applied. Therefore, it is possible to develop an 'adaptive algorithm' which does as much work as possible to guarantee the sign of the determinant. There are several kinds of filters, fully-static filters [2], semi-static filters [2,3,4], dynamic filters [3,5,6].

We develop an efficient semi-static filter and an improved fully-static filter. As a result, even if overflow or underflow occur, our filters work correctly with only one branch. Finally, we show the application of verified numerical computations for Graham's algorithm.

This is joint work with Florian Bünger, Takeshi Ogita and Shin'ichi Oishi.

References:

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

[2] C. Burnikel, S. Funke, M. Seel, Exact Geometric Computation Using Cascading, International Journal of Computational Geometry & Applications, 11(3):245-266, 2001.

[3] J. R. Shewchuk, Adaptive Precision Floating-point Arithmetic and Fast Robust Geometric Predicates, Discrete & Computational Geometry, 18: 305–363, 1997.

[4] G. Melquiond, S. Pion, Formally certified floating-point filters for homogenous geometric predicates, Theoretical Informatics and Applications, Special issue on Real Numbers, 41:57–69, 2007.

[5] H. Brönnimann, C. Burnikel, S. Pion, Interval Arithmetic Yields Efficient Dynamic Filters for Computational Geometry, Discrete Applied Mathematics, 109:25–47, 2001.

[6] V. Y. Pan, Y. Yu, Certified Computation of the Sign of a Matrix Determinant, Proc. 10th Annual ACM-SIAM Symposium on Discrete Algorithms, 715-724, ACM Press, New York, and SIAM Publications, Philadelphia, 1999.