Delaunay Refinement by Corner Lopping

Steven E. Pav\textsuperscript{1} and Noel J. Walkington\textsuperscript{2}

\textsuperscript{1} University of California at San Diego, La Jolla, CA. \texttt{spav@ucsd.edu}
\textsuperscript{2} Carnegie Mellon University, Pittsburgh, PA. \texttt{noels@andrew.cmu.edu}

\textbf{Summary.} An algorithm for quality Delaunay meshing of 2D domains with curved boundaries is presented. The algorithm uses Ruppert’s “corner lopping” heuristic [MR96b:65137]. In addition to admitting a simple termination proof, the algorithm can accept curved input without any bound on the tangent angle between adjoining curves. In the limit case, where all curves are straight line segments, the algorithm returns a mesh with a minimum angle of \(\arcsin \left( \frac{1}{2\sqrt{2}} \right)\), except “near” input corners. Some loss of output quality is experienced with the use of curved input, but this loss is diminished for smaller input curvature.

\textbf{Key words:} unstructured, simplicial, planar, curved boundary, Delaunay, mesh.

1 Introduction

The Delaunay Refinement method is used for quality simplicial mesh generation in two and three dimensions. A Delaunay Refinement algorithm takes an input of points and segments (or curves) and adds Steiner Points to guarantee that the output Delaunay Triangulation conforms to the input and has high quality simplices, as measured by the circumsphere to shortest edge length ratio. A Steiner Point is added to “split” an input segment into subsegments if a mesh vertex forms an obtuse angle with the segment. A Steiner Point is added at the circumcenter of a poor triangle in the mesh. Termination of the algorithm is had by proving a lower bound on the distance between Steiner Points, and applying compactness arguments [MR96b:65137, Pse2003].

Ruppert was a pioneer of the Delaunay Refinement method. Ruppert’s Algorithm accepts a planar straight line graph, and outputs a Delaunay mesh where no output angle is smaller than a user-chosen parameter, which can be as large as 20.7\(^\circ\). In Ruppert’s analysis input segments have to meet at nonacute angles, otherwise his naïve algorithm might not terminate [MR96b:65137].

Ruppert offered two heuristic solutions to this problem. The first, “concentric shell splitting,” has been adapted to a working algorithm, and allows better output quality guarantees [Sjr1997, Pse2003, MglPseWnj2005]. In this solution, segments sharing a common endpoint are split at the same distance from the endpoint, \textit{i.e.}, on
the same “shell.” This simple fix gives a good lower bound, and an input-independent upper bound, on output angles. However, its analysis is involved, and does not generalize naturally to higher dimensions or curved input (due to its reliance on “power of two” arguments). Ruppert’s second solution, “corner lopping,” is analyzed herein, and admits a simple proof.

Delaunay Refinement for curved input was considered by Boivin and Ollivier-Gooch [BcOGc:2002]. Their analysis requires that input segments meet at an angle of at least $\pi/3$. Concentric shell splitting to deal with smaller input angles was mentioned in this context, but not shown to give a working algorithm; this fix clearly would require further modification to the output quality guarantee.

\textbf{Fig. 1.} The outline of a mock air foil and output from the meshing algorithm. Solving a fluid dynamics PDE would probably require further mesh refinement.

The Delaunay Refinement Algorithm has also been generalized to three dimensions. Early analysis required input segments and faces to meet at nonacute angles [MgIPseWnj2002c]. As a fix, later work used protective regions around input points and segments [338236, CoVeYv01, Cswetal2004, PseWnj2004]. In that way these algorithms resemble corner lopping, which places a protective ball around acute corners in the input. Reverse from what is usually seen, these three-dimensional algorithms do not appear to be the natural generalization of any known two-dimensional algorithm.

The motivations for the present work, then, are: