Assembling Algebraic Surfaces

Caden D. Joergens, Dr. Silviana Amethyst| Mathematics, University of Wisconsin - Eau Claire

University of Wisconsin Eau Claire

Introduction

Algebraic surfaces are beautiful objects that describe sets of space where a system of polynomials vanishes. Their surface is smooth, but can have corners, cusps, self-intersections, and sharp points. When 3D printing such surfaces, the smooth places are simple. Singularities, where the surface meets at a single point of no dimension, occur at all corners and crossings. This lack of dimensions at singularities presents inherent problems for accurately representing a 3D model. Past methods for creating a 3D printed model rely on global solidification, applying a thickness to the entire model, forcing the singularity to appear as a smooth surface.

Goals

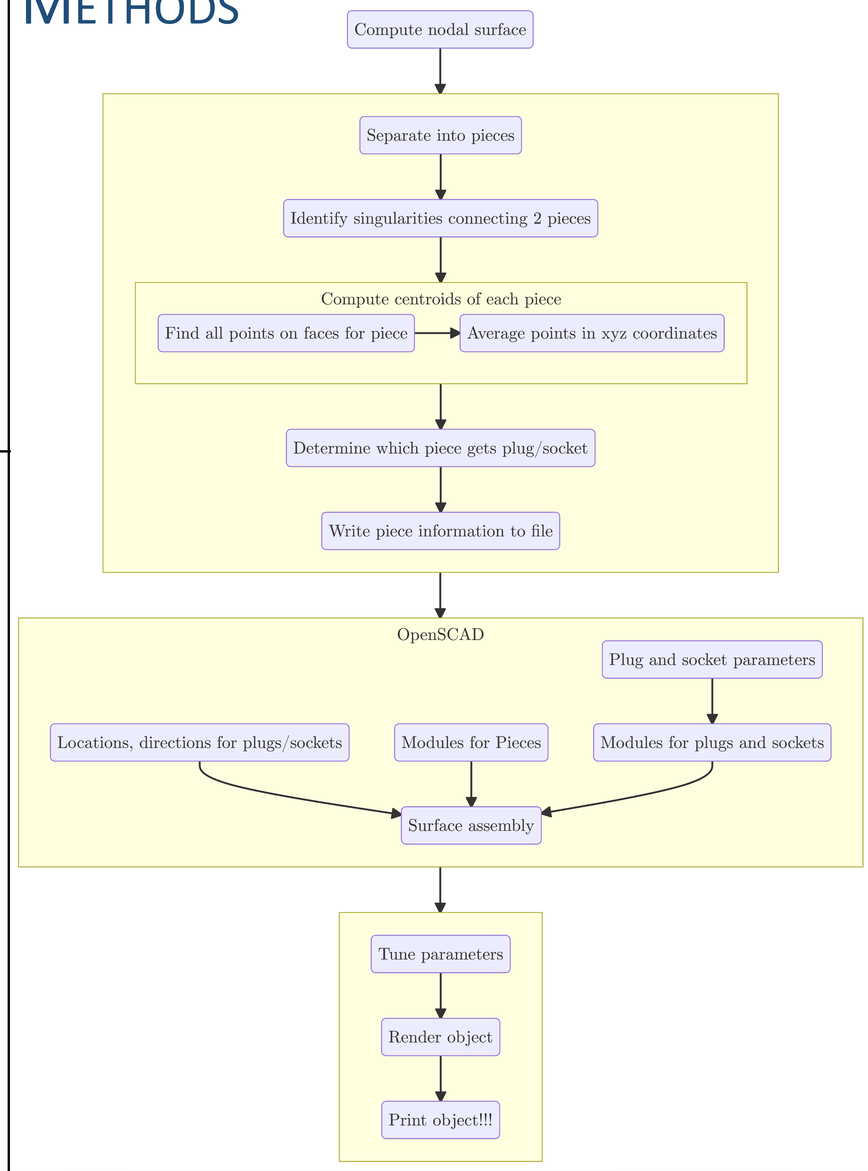
Implement a program that automatically places a plug and socket at singularities where two pieces connect, allowing pieces to snap together.

Tools

This work builds upon the bertini_real software, in development since 2013. bertini_real allows us to deconstruct algebraic surfaces and export for 3D printing
Other tools and technologies include:

OpenScad python blender

Methods



Results

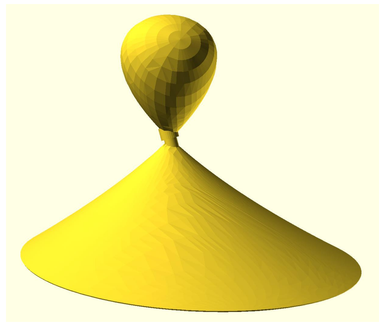


Fig.1 - Rendering of complete Dingdong surface.

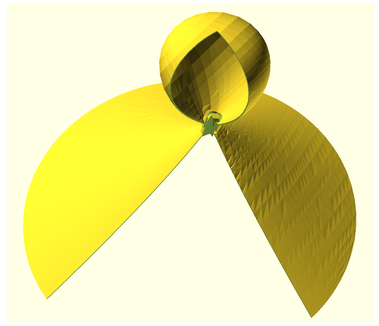


Fig.2 - Rendering of Dingdong surface. Section removed for internal view

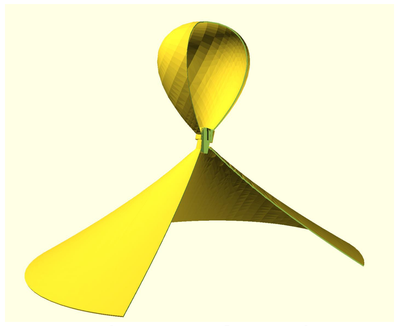


Fig.3 - Side view of Dingdong surface. Section removed for internal view

Issues

Because of the heuristic technique used to compute the directions the direction vectors are off-axis from what would be ideal.

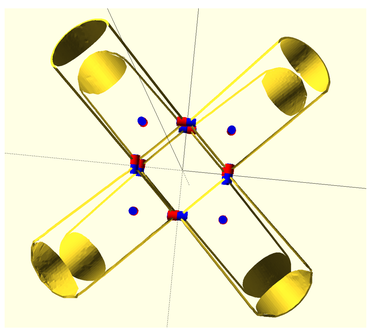


Fig.4 – Preview of Kummer Surface

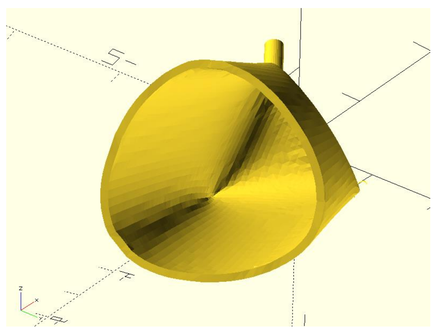


Fig.5 - Rendering of a Kummer Piece

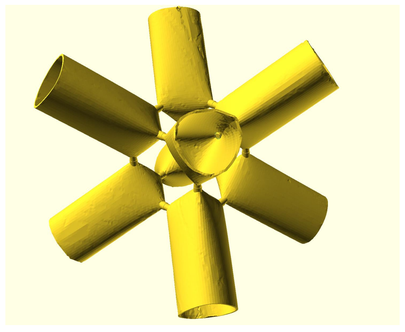


Fig.6 - attempt at fixing Kummer Surface

Issues

When rendering the Kummer surface a CGAL error is returned, these errors are returned when an assumption made by OpenSCAD is broken. This prevents the surface from rendering and when individual pieces are rendered holes for sockets are not cut. Attempts to resolve this issue involve importing each piece's .stl file to external software to repair the mesh. The most successful solution has been using Blender to solidify and remesh each piece. Fig.6 shows the rendered surface, but two sockets still fail to cut holes, returning a CGAL error and mesh not closed error.

Next Steps

Other methods of calculating the centroid will be investigated to calculate the centroid of each piece. This may include determining a weighted average or calculating the area under the curve.
The issues encountered with rendering are fundamental and have caused a stalemate with OpenSCAD. In the last several years Blender has undergone updates, making it a more viable candidate for rendering and printing.

Future Work

Building upon snap assembly of surfaces will be the implementation of a program to print the wireframes of these surfaces

References

•bertinireal.com
•S. Amethyst, D.J. Bates, W. Hao, J.D. Hauenstein, A.J. Sommese and C.W. Wampler. Bertini_real: software for real algebraic sets. Available online at bertinireal.com. Amethyst, S. C., Bates, D. J., Hao, W., Hauenstein, J. D., Sommese, A. J., & Wampler, C. W. (2017). Algorithm 976: Bertini_real: Numerical decomposition of real algebraic curves and surfaces. ACM Transactions on Mathematical Software (TOMS), 44(1), 1-30.

Acknowledgments

•This material is based upon work supported by the National Science Foundation under Grant No.~DMS-1929284 while the first author was in residence at the Institute for Computational and Experimental Research in Mathematics in Providence, RI, during the Illustrating Mathematics program.
•Support for this project came from Student Blugold Commitment Differential Tuition funds through the University of Wisconsin-Eau Claire Summer Research Experiences for Undergraduates program.

We thank the Office of Research and Sponsored Programs for supporting this research, and Learning & Technology Services for printing this poster.