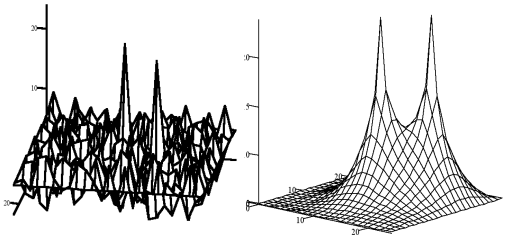
# Re-usable Implicit Functions

Alyn Rockwood
Mathematical and Computer Sciences, Colorado School of Mines, Golden, CO

Implicit functions play a major role in modeling shapes. [Bloo] lists many implicit models such as super-quadrics, blended surfaces, convolution surfaces, metaballs, alpha patches and others. The *zero-set* of an implicit function $f$ (where $f(\mathbf{x}) = 0$) is typically the set of points employed as the model. The non-zero set is, however, also important as many operations on implicit models depend on the non-zero part. Operations on implicit models such as booleans, blending, shape texturing, or local parameterizations rely on smoothly varying space in the non-zero region to produce reasonable results. Many of the operations tend to degrade the space: thus booleans and blending create non-differentiabilities [Rock]. Some implicit functions like the Weierstrass function (a fractal) begin with undesirable characteristics. We present a method that redefines the non-zero space, without affecting the zero set; thus making operations more successful. It makes implicit modeling operations reusable and in some cases makes difficult implicit models usable.

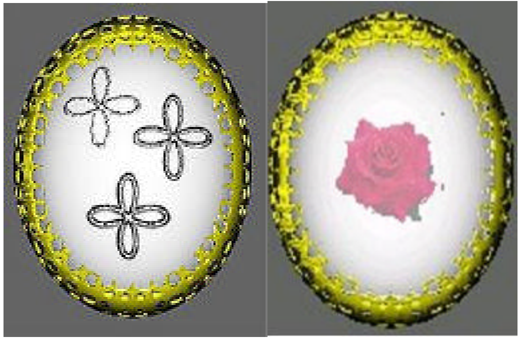
We modify space by finding the minimal energy function off the zero set, one that leaves the zero set unchanged. We illustrate in 2D. Consider the bi-variate function shown in the following figure on the left.



The two peaks are defined as the zero set. The non-zero region is defined randomly. This is as bad a non-zero region as could be imagined. We fix the two zero set points and the boundary, and then find the minimal energy (soap film) surface that connects the fixed parts (right). Such surfaces are the same as minimal surface area. Our algorithm uses the sum of the areas of the rectangles that tessellate the surface as a cost function. By varying a function value on an arbitrary rectangle vertex a new cost is computed.

This is applied to 3D by minimizing the energy of the volume, i.e. changing the value of the function on a vertex of the cuboids that tessellate space. Using a simple greedy algorithm that accepted lower energy states and ignored others, the convergence was rapid and robust.

The broaches below required this in several steps such the union of the filigree to the super-ellipsoid and the parameterization via streamlines to imprint the petals with local morphs (left), or a bitmap (right)



Each of the above models is very concise; requiring less than a dozen lines of C++ code to define the functions, which is characteristic of this kind of implicit model.

## Bibliography

[Bloo] Bloomenthal, J., *Introduction to Implicit Surfaces*, Morgan Kaufmann Publishers, 1997.

[Rock] Rockwood, A., “The Displacement Method for Blending Surfaces in Solid Modeling,” *ACM Transactions on Graphics*, Special Issue on CAD/CAM, v.8, No.4, Oct. 1989, pp. 279-292.