Branches and Bifurcations

Building a framework for modeling with isosurfaces in Generative Components

Mirco Becker
Kohn Pedersen Fox Associates International (PA), England
http://www.kpf.com

An isosurface is a three-dimensional representation of a constant value of a field function within a given volume. They are normally used in computer graphics to visualize data in fluid dynamics, medical imaging, geophysics, and meteorology. The advantage of isosurfaces is that they can represent all sorts of topologies. That makes them a perfect tool for modeling, branching, forking, and bifurcating objects with smooth transitions. As they work of a field function, the surface is implicit, the polygonization an approximation. This is a good base for coupling performance with precision.

The task was to define a set of handles to change and model an isosurface. It had to happen through the modeling of the field function in a way that is rather intuitive but without giving up the precision one is used to have from standard NURBS/ BREP modeling. The paper shows how a modeling framework for isosurfaces is implemented as a plug-in for Bentley Systems Generative Components allowing an intuitive way of exploring design variations. The implementation is illustrated with a proof of concept showing a sketch design.

Keywords: *Isosurface; Polygonization; Scalar field; Marching Cube; Generative Components.*

Use for architecture and design

The use of isosurfaces in standard CAD applications is rather rare. They are more common in animation / visualization packages. In these applications they are often known as Metaballs. Through that sidetrack they even found their way into architectural design. One example is the "Bubble" pavilion for BMW by Bernhard Franken. Another potential application can be found in forms that one might be associated with the architecture of Santiago Calatrava.

Most 3D surface and solid geometry in CAD modeling packages works are based on BREP solids and/or NURBS surfaces. Both techniques have limitations regarding their capability to blend smoothly between objects. Especially, when implemented in an environment that supports dynamics, a great deal of computational effort is necessary to maintain consistency between boundary conditions and blends. These problems might occur when designing joints and knots with smooth transitions.

GC-IsoSurf was implemented as a plug-in for

Bentley Systems Generative Components (GC) utilizing a Marching Cube (MC) algorithm[1]. There are other alternative algorithms for filed polygonization but the MC gave the best result in terms of rapid implementation.

The main task was to define a control mechanism that generates a field that could than be polygonized and possibly match a certain design intent as closely as possible. There are algebraic means to define a filed but that limits the variety of possible forms to the availability of surface equations. As architectural design is already quite object / component based, it seemed to be a sensible choice to use seed objects as field emitters. The strength of the field that each object emits is defined by a parameter that corresponds to a global threshold value. That allows setting dimensional values for the surface in relation to its seed object. They also carry a flag on how to interact with other fields. That allows for various transitions and compensations.

Using geometry as field emitters

Taking the approach of object based emitters, the most obvious choice is to start with simple geometry. In that sense, a point can be seen as the most basic field emitter. We assume an emitter characteristic where the field strength decreases with the square of distance to the point.

isoValue = 1 / distance * distance

There are other forms of field function; but this very simple one will do and can be swapped later on against more sophisticated functions, if necessary. An arbitrary iso-value within the filed describes a spherical surface with the point emitter as a centre of the sphere. It would be more practical to have the iso-value for a certain distance to the point so one can retrieve a sphere with a specific radius. We find the required iso-value through solving the formula by inserting the required distance. We call this value the threshold value. Obviously the metrically precise threshold value only works for emitters that don't interact with other emitters as that could change the absolute dimensions in Cartesian space. So we remember that some control over the metric articulation of the surface is lost as it comes into interaction with emitters.

As we introduce other geometry than points the field strength of any point in space would be the inverse square of the distance to the closest point on the emitter geometry. That makes it relatively easy to use any kind of geometry available in a CAD package as field emitter, i.e. a line emits a field that would result in a cylindrical isosurface with rounded ends

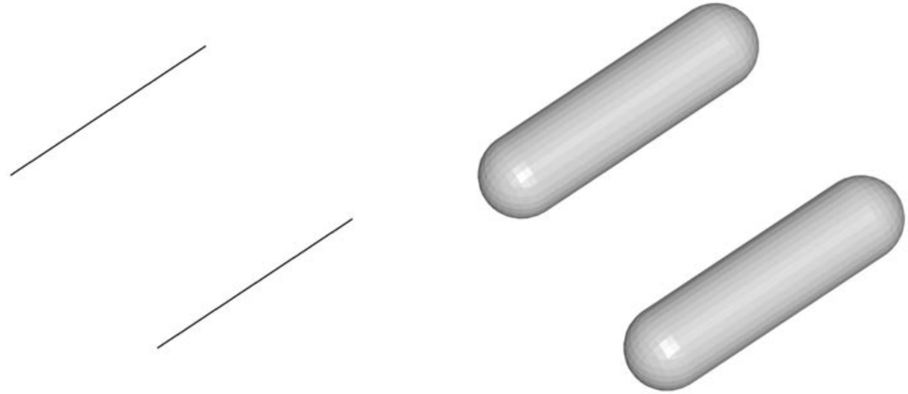


Figure 1
Two lines emitting fields without influencing each other

---

[1] http://astronomy.swin.edu.au/~pbourke/modelling/polygonise/

at any given threshold value (Figure 1).

Emitter sets and field interaction

Emitters that contribute to the same field are part of a set. Interaction is only possible within a set. The basic interaction is that all fields from all emitters accumulate to one filed per set. Each set has a maximum bounding box based on the contained geometry. Within that bounding box, the MC polygonization is performed for each set individually. That allows for the best flexibility and performance as one can apply different resolutions of polygonization to different sets and the volume analyzed is kept relatively small. No matter how many emitters happen to be in one set, there is only one field per set. Depending on the distance between emitters and the chosen threshold value, one might retrieve independent surfaces from that field (Figure 2). That is one of the main characteristics of isosurfaces. Based on that characteristic they are also known as blobs.

As two or more linear emitters of one set meet in one point they not only blend together but also result in a thickening of the node. That effect is also an important part of the isosurface characteristics but not always required or wanted (Figure 3).

There are basically two useful ways of gaining control over that effect. The first takes advantage of the way we use geometry as emitters. If we replace adjacent lines by polygons or line-strings, the thickening disappears as the iso-value is calculated on a closest point distance and only one emitter is contributing to that field (Figure 4). The second approach of emitter based dampening will be described in detail further down.

Using B-Spline curves as emitters proves how flexible the concept of geometry based emitters is (Figure 5). It is even possible to use B-Spline surfaces

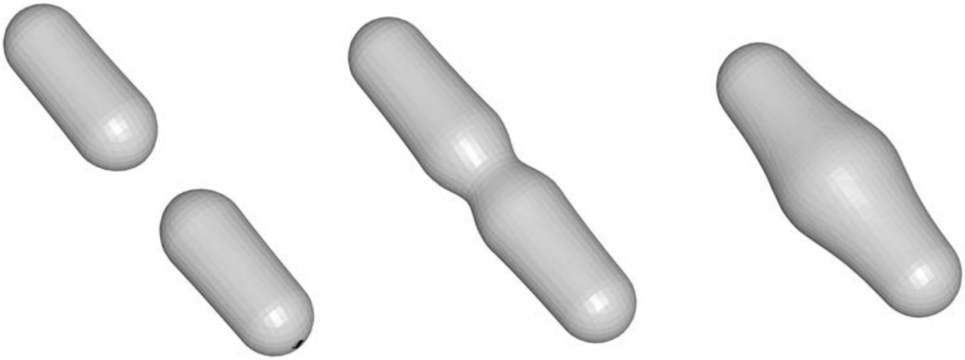


Figure 2
Two lines emitting fields with varying distance showing how the fields add up

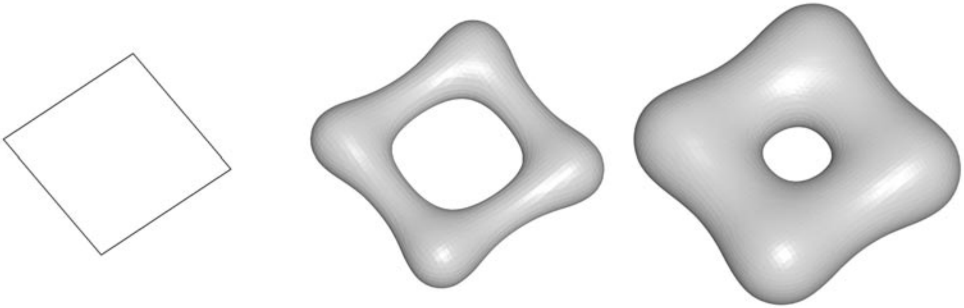


Figure 3
Four lines emitting fields adding up at the junctions and form characteristic thickening

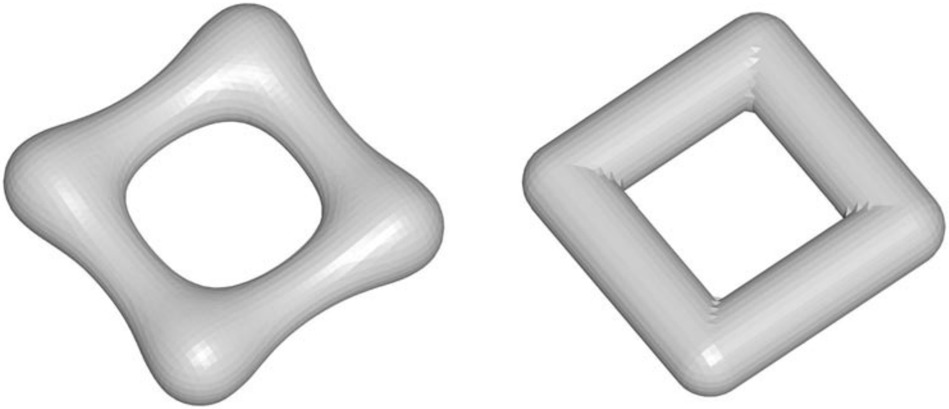
or solids as emitters. Even the contribution of algebraic embitters is possible.

Having arranged the emitter geometry in a certain way it is possible to retrieve multiple isosurfaces from the resulting field by solving for different threshold values. An inherent property of isosurfaces is that they layer perfectly like "onion skin". That carries a huge potential to explore objects as layered composites. These layers will result in off-set isosurfaces that have a constant iso-value off-set but not necessarily a constant Cartesian off-set (Figure 6). The great advantage over other surface / solid concepts is that this method never results in problematic self intersections. Nevertheless, the isosurfaces can undergo standard Boolean operations. This opens the possibility for thin-shelled objects and the combination and integration with standard BREP solids.

Emitter based damping

The second method of gaining control over the node thickening at the junctions of linear emitters is by introducing negative fields. By putting a negative point emitter in the junction, the thickening can be controlled quite precisely. (Figure 7). That carries the potential for optimizing such a nodal point in terms of strength and volume. As any geometry could emit a negative filed, that concept could be employed for more radical design intervention, i.e. generating

Figure 4
Left isosurface from four line emitters, right isosurface from one closed polygon emitter



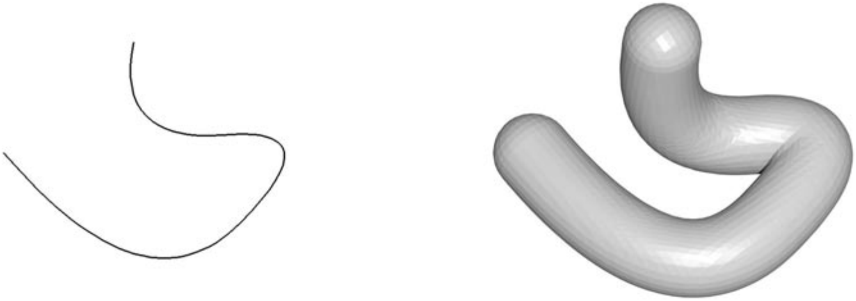


Figure 5
B-Spline emits field

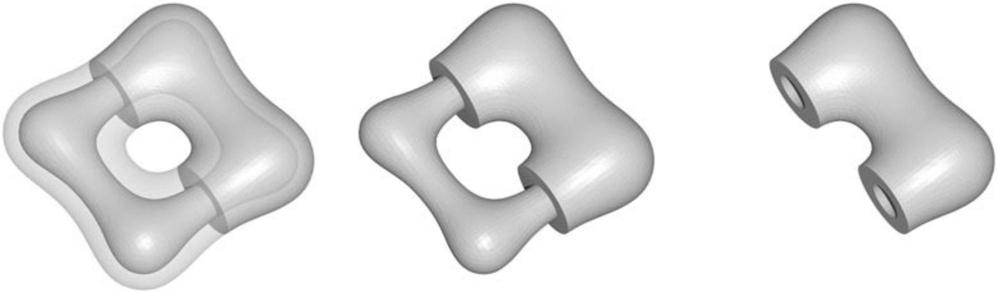


Figure 6
Two isosurfaces from the same field and Boolean subtraction

openings and holes.

Polygon mesh

As the MC subdivides the field in equal cubes, the resulting mesh has possibly some redundancies which might be overcome by employing a different polygonization algorithm. On the other hand the set based approach allows defining the mesh resolution individually. A low resolution gives a coarse polygonization but high dynamic performance. A higher resolution gives a finer surface approximation and can be carried forward for further CAD treatment or directly to prototyping.

Each solution comes as a mesh object which is a very generic 3D representation and integrates with almost every available CAD rapid-prototyping technique without further translation.

Implementing IsoSurf in Generative Components

Bentley Generative Components is a parametric design framework based on Bentley's MicroStation. It is a feature based dependency graph system where the user can build its own features on basically three possible levels: A graphic GUI level, a scripting level within the application and by pure code in C#.

For the implementation of GC-Isosurf, the C# method was the most sensitive choice as it allowed transferring existing algorithms quite easily and getting the best performance as the MC algorithm is already computationally quite expansive.

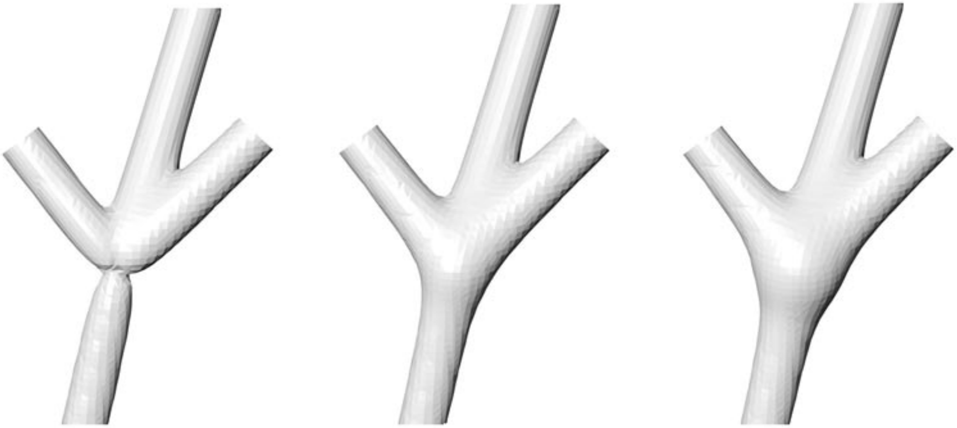


Figure 7
Four lines as emitters and one negative point emitter to dampen the node thickening

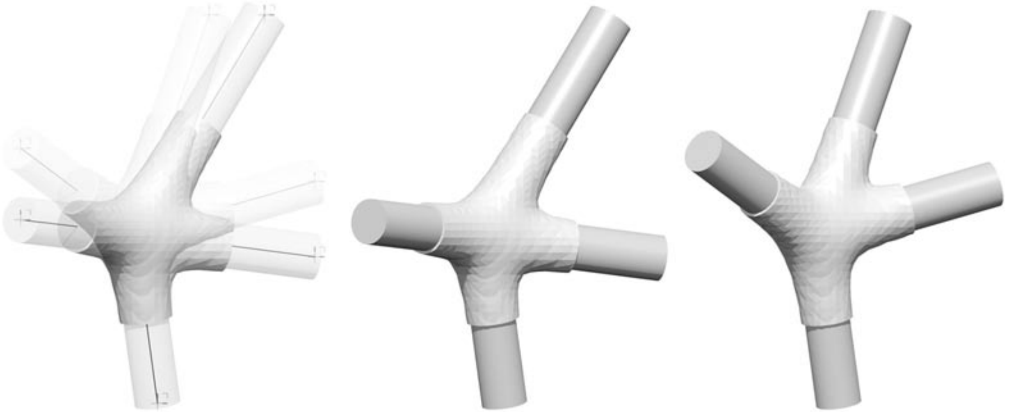


Figure 8
Study of node system using IsoSurf in Generative Components

Once the plug-in was written and loaded it nicely integrated in the dependency graph logic where it takes emitter geometry as inputs and outputs the polygonized solution based on the current parameter set.

Proof of concept

GC- Isosurf's potential was tested against a sketch design using flexural resistant joints. That provided a perfect base for testing various sorts of branching connectivity with standard profiles and customization of a large number of joints. The joints were supposed to be fed directly into a digital fabrication, in case of this example into rapid prototyping for a scale model.

One might argue that isosurfaces have some disadvantages in terms of metric control. But that is probably balanced by the great potential that lies in there other characteristics. The most interesting seems to be the notion that the inherent layered logic of isosurfaces follows a similar logic of emerging prototyping and manufacturing techniques that work with additive material deposition.

The missing link for making it a truly powerful solution would be a feedback connection to FEM analysis. That would open the possibility for optimizing the material distribution in respect to design performance and material properties and could thereby supersede metric control.

References

Franken B.: 2003, Real As Data, in Architecture in the Digital Age: Design and Manufacturing Branko Kolarevic (Editor) Spoon Press.

Lorensen W.E. & Cline H.E.: Marching Cubes: a high resolution 3D surface reconstruction algorithm, Computer Graphics, Vol. 21, No. 4, pp. 163-169, (Proc. of SIGGRAPH), 198

Aish R.: 2006, GenerativeComponents Parametric Design Software Development, in Architectural Design March/April 2006, p. 47.