U-splines for Unstructured IGA Meshes in LS-DYNA®

Dr. Michael Scott
Brigham Young University, Coreform LLC

Abstract

The isogeometric analysis (IGA) paradigm [5] eliminates the CAD/CAE data translation problem by using the CAD geometry directly as a basis for analysis. IGA was introduced by Dr. Thomas J.R. Hughes et. all in 2005 (Dr. Hughes is now a senior advisor and co-founder of Coreform), and has produced over 1500 academic papers to date, multiple annual conferences dedicated to IGA, and numerous eye-popping results [7]. In other problems, it gives an accurate answer when FEA gives an inaccurate answer. IGA especially shines in nonlinear structural simulations like contact, highly nonlinear deformations, and fracture, with examples showing dramatic increases in efficiency and robustness over traditional FEA [4, 6]. Despite this success, IGA has yet to be successfully commercialized. The reason is that there has not existed a suitable, watertight CAD geometry capable of representing arbitrarily complex, industrial-grade shapes in a way that is suitable for direct simulation. The closest CAD description to achieving this is T-splines, introduced into IGA in Coreform co-founder Dr. Michael Scott's PhD dissertation [8]. T-splines are a pioneering watertight CAD technology which overcome many of the decades old limitations of standard CAD descriptions based on Non-Uniform Rational B-splines (NURBS) [10]. T-splines can now be found in several major commercial computer-aided design (CAD) products [1, 2]. However, important limitations remain in the T-spline definition which prevent a full instantiation of the isogeometric paradigm [9]. Over that last ten years, the IGA community has distilled a concise set of properties that a CAD description should possess to be analysis-suitable or useable as an optimal basis for IGA. These properties are shown in Table 1 where T-splines, Boundary Representation (BREP) CAD models (the CAD industry standard), FEA meshes, and a new CAD description called U-splines (to be described subsequently) are compared.

Unstructured splines or U-splines maintain and improve upon the design advantages of T-splines but are also analysis-suitable as shown in Table 1. This means that U-splines can be used as a completely interchangeable and integrated geometric representation for CAD and simulation. U-splines can be represented as unstructured meshes, as is common in FEA, or as CAD BREP geometry, the current de-facto industry standard, as is common in mechanical CAD. In particular, a U-spline is characterized by blending functions that are smooth, higher-order, refineable, linearly independent, positive, form a partition of unity, and are complete through some parametric polynomial degree. To achieve this level of flexibility and precision in the basis, the standard process used to construct splines like NURBS and T-splines is inverted.

U-splines can be used to convert unstructured meshes into smooth models for use in LS-DYNA through exporting to Bezier elements.

U-spline Basis Construction

In the standard approach to constructing a NURBS or T-spline, a control mesh is specified. The control mesh defines the position of control points, their connectivity, and the relationship between blending functions. Additionally, parametric lengths, called knot intervals, are assigned to each edge of the control mesh and a global parametric degree for the blending functions is specified. Once the control mesh is specified, an algorithm is executed which infers a set of B-spline blending functions from the topology of the control mesh, interval assignment, and degree of the control mesh. An algorithm for NURBS and T-splines is described in [10]. Once the set of blending functions is determined, the final element mesh can be inferred, which then becomes the computational finite element mesh used in IGA. This mesh is referred to as the Bézier mesh as each element represents a single 20lstc18 patch. The technique used to determine a Bézier mesh for a spline is called Bézier extraction [3].

In contrast, U-splines invert this process. The basic steps followed in the construction of a U-spline are shown in Figure 2. First, a mesh is specified. A parametric length is assigned to each edge (or face in 3d). In step 1 of the figure, the red lines are half the length of the black lines. Next, the polynomial degree of each element (which can be different from element to element and in independent directions within each element) is specified and smoothness levels are assigned to each edge in the mesh. In step 2 of the figure, the orange boxes mark cells with polynomial degree 3 in both directions. The blue lines indicate curvature-continuous edges while the black lines mark discontinuous edges. An algorithm is then executed, which builds each basis function in turn by determining the elements required for a single function and enforcing the smoothness and degree requirements on each element and edge. The result of this algorithm is shown in step 3 of the figure. A full U-spline surface produced by building all possible basis functions and assigning each a unique value is also shown.

A novel property of U-splines is shown in Step 3 of Figure 2 — a contour plot of one of the basis functions constructed in the neighborhood of two T-junctions in close proximity. This example is a local refinement of a simple square mesh. The local nature of the U-spline algorithm produces an L-shaped basis function that is adapted to the local element structure. Other technologies, like T-splines, are unable to product analysis-suitable geometry for this topology. Another basis function is shown in Figure 3. This basis function is defined over a mesh with varying levels of continuity. The blue lines are curvature continuous while the green lines are value continuous. This variation is observable in the basis function. The sharp features are due to the reduced levels of continuity in localized areas. Similar results can be obtained for mixed degree cases but are more difficult to observe visually. The optimal locality of the U-spline algorithm permits multiple (hpk) types of adaptive refinement for both design and analysis while preserving the mathematical properties necessary for analysis. The ability of the U-spline algorithm to construct splines of maximal smoothness provides increased robustness and efficiency as compared to standard FEA techniques while still enabling the representation of complex CAD geometry.

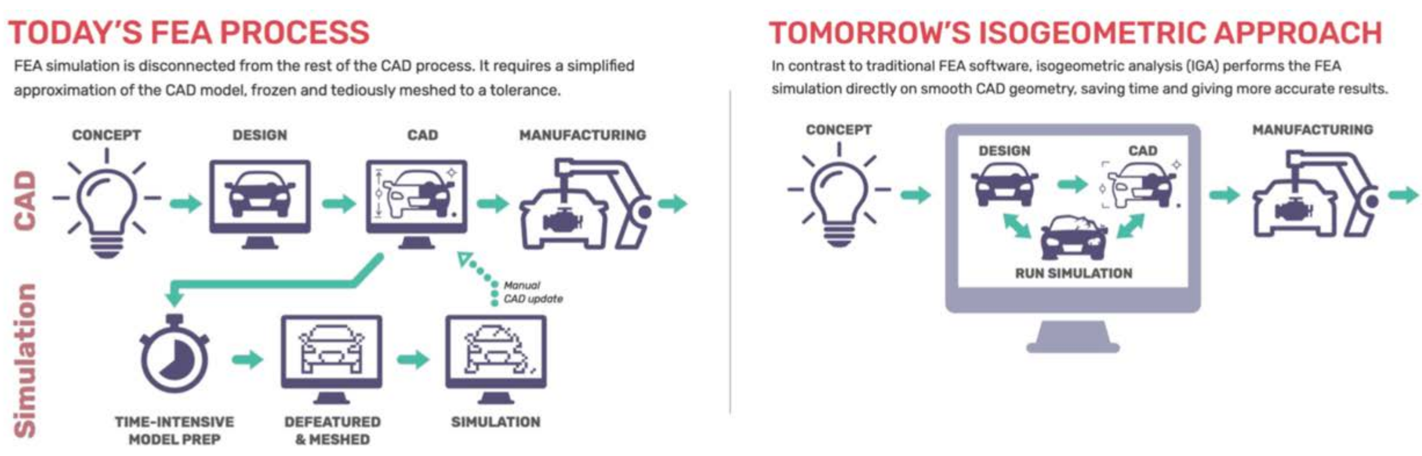

Figure 1: An isogeometric CAD-CAE approach will enable more accurate and robust simulation and better integration between CAD and CAE. For example, in automotive crash simulation, today's FEA process takes many millions of dollars of manual labor and several months to prepare the model for simulation, time that would be nearly eliminated by the isogeometric approach.

Property	U-splines	BREPs	T-Splines	FEA
NURBS compatibility	**Yes**	Yes	Yes	No
Exact Geometry	**Yes**	Yes	Yes	No
No dirty geometry	**Yes**	No	Yes	No
Smooth basis	**Yes**	Yes	Yes	No
Arbitrary degree	**Yes**	Yes	No	Yes
Local exact h, p, k adaptivity	**Yes**	No	No	No
Watertight	**Yes**	No	Yes	Yes
Linear independence	**Yes**	Yes	No	Yes
Optimal approximation	**Yes**	No	No	Yes
Volumes	**Yes**	No	No	Yes

Table 1: A comparison of the analysis-suitability of U-splines, BREPs, T-splines (patent owned by Autodesk), and FEA meshes.

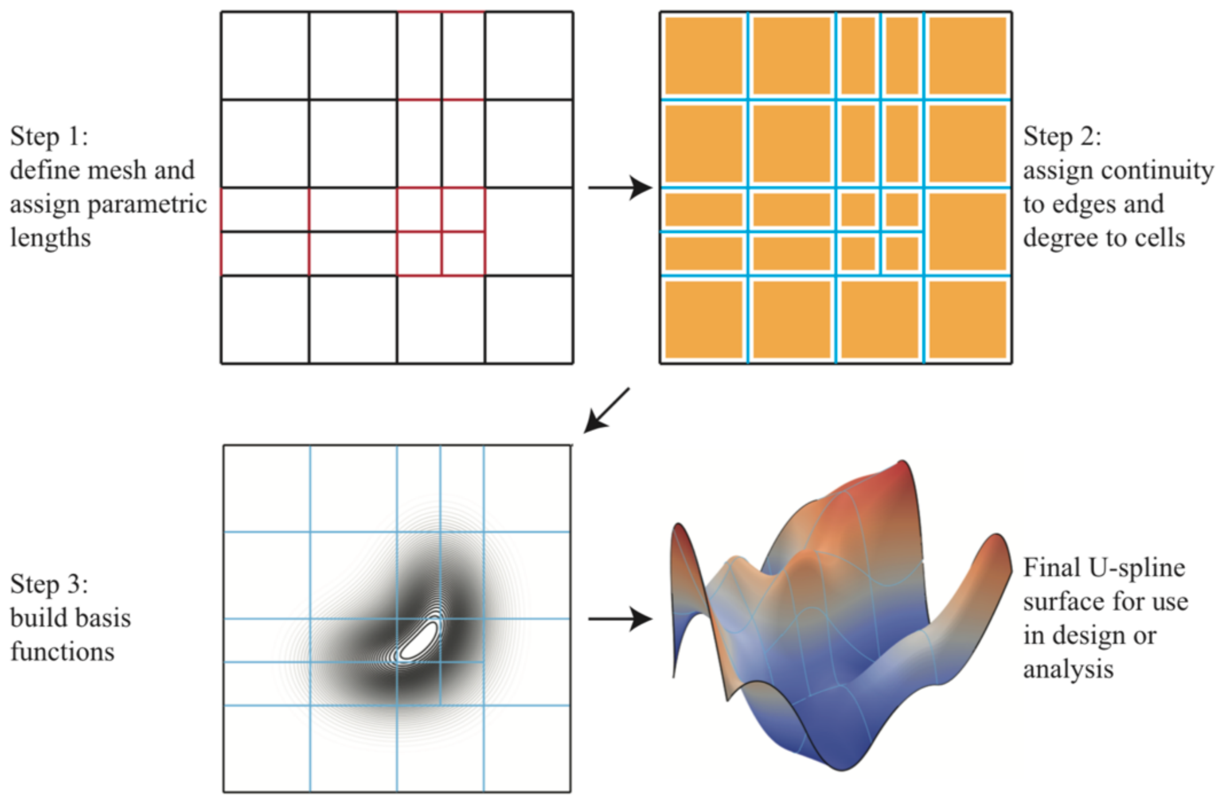

Figure 2: Graphical representation of the U-spline process. A mesh is defined with parametric lengths (red line segments are half the length of black line segments). Continuity and polynomial degree values are then assigned to edges and faces respectively. The U-spline algorithm is then used to construct each basis function. The basis can then be used to define a surface.

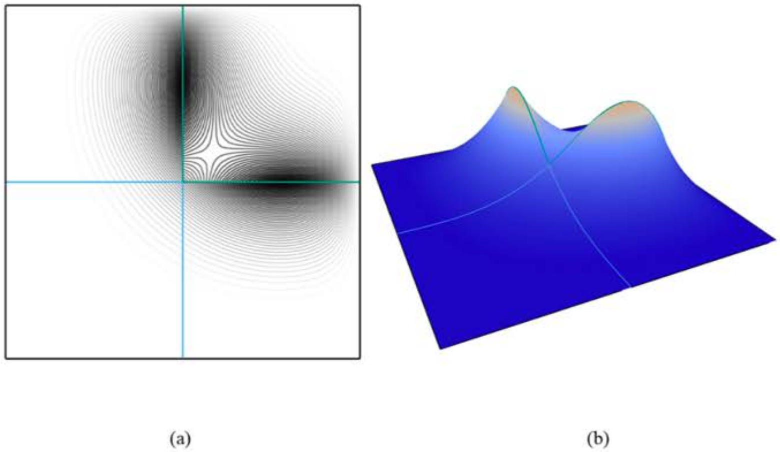

Figure 3: Example U-spline basis function exhibiting nonuniform continuity. The blue lines are curvature continuous while the green lines are value continuous. A contour plot is shown in part (a). The creased surface corresponding to this basis function is shown in part (b).

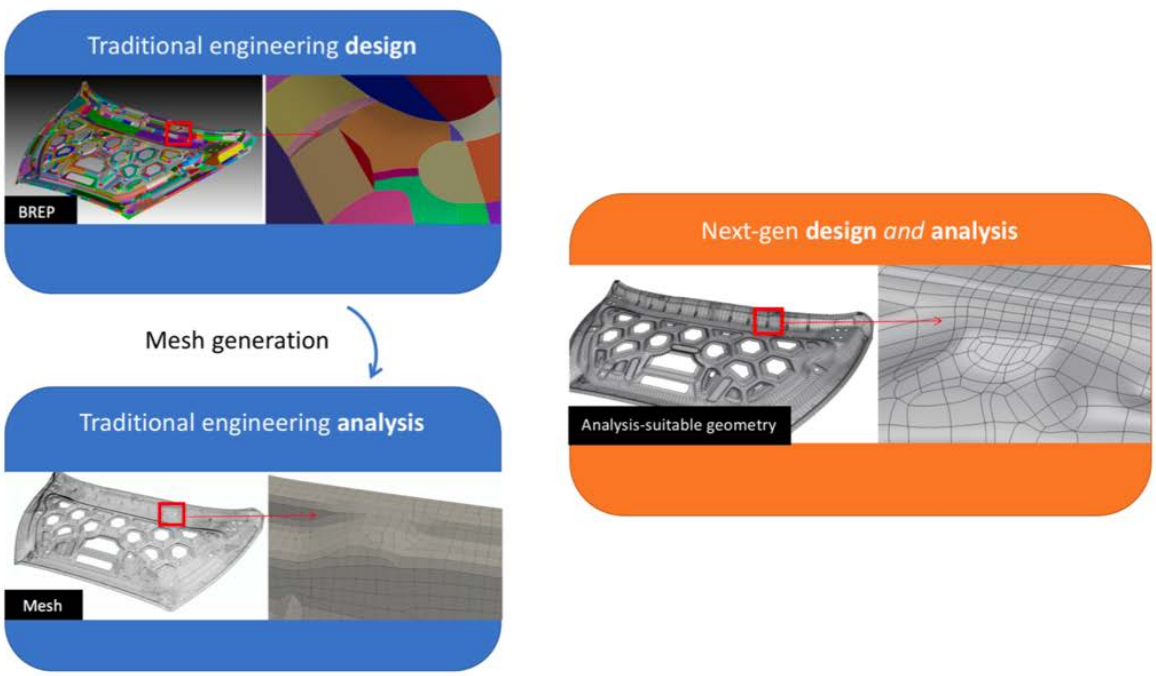

Figure 4: Unlike CAD BREPS, which are smooth but lack the watertight requirements of FEA, and FEA meshes, which are watertight but lack the smooth precision of CAD, U-splines (and analysis-suitable CAD) are watertight and smooth and are suitable for both CAD and FEA.

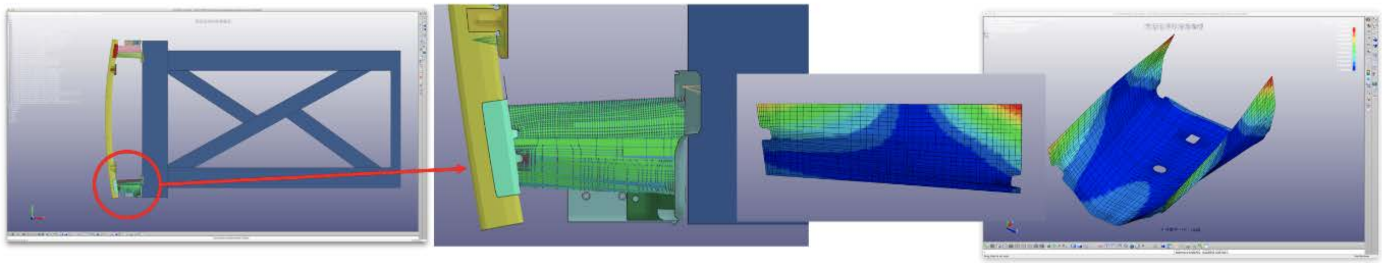

Figure 5: An FEA bumper assembly in LS-DYNA with crash can (circled) converted to smooth U-splines. Simulation of the crash can in LS-DYNA is shown in the right image. Data courtesy Ford Motors.

References

1. Autodesk. Autodesk T-Splines Plug-in for Rhino user manual. Autodesk, 2012.
2. Autodesk, Inc. Autodesk Fusion 360. Autodesk, Inc., 2014.
3. M. J. Borden, M. A. Scott, J. A. Evans, and T. J. R. Hughes. Isogeometric finite element data structures based on B´ezier extraction of NURBS. International Journal for Numerical Methods in Engineering,, 87:15 – 47, 2011.
4. J. A. Cottrell, T. J. R. Hughes, and A. Reali. Studies of refinement and continuity in isogeometric analysis. Computer Methods in Applied Mechanics and Engineering, 196:4160–4183, 2007.
5. T. J. R. Hughes, J. A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement. Computer Methods in Applied Mechanics and Engineering, 194:4135–4195, 2005.
6. S. Lipton, J. A. Evans, Y. Bazilevs, T. Elguedj, and T. J. R. Hughes. Robustness of isogeometric structural discretizations under severe mesh distortion. Computer Methods in Applied Mechanics and Engineering, 199(5-8):357–373, 2010.
7. S Morganti, F Auricchio, DJ Benson, FI Gambarin, S Hartmann, TJR Hughes, and A Reali. Patient-specific isogeometric structural analysis of aortic valve closure. Computer Methods in Applied Mechanics and Engineering, 284:508–520, 2015.
8. M. A. Scott. T-splines as a Design-Through-Analysis technology. PhD thesis, The University of Texas at Austin, 2011.
9. M. A. Scott, X. Li, T. W. Sederberg, and T. J. R. Hughes. Local refinement of analysis-suitable T-splines. Computer Methods in Applied Mechanics and Engineering, 213:206 – 222, 2012.
10. T. W. Sederberg, J. Zheng, A. Bakenov, and A. Nasri. T-splines and T-NURCCs. ACM Trans. Graph., 22:477–484, July 2003.