Dynamic Particle Systems for Efficient and Accurate Finite Element Visualization

Miriah Meyer and Ross Whitaker

www.sci.utah.edu

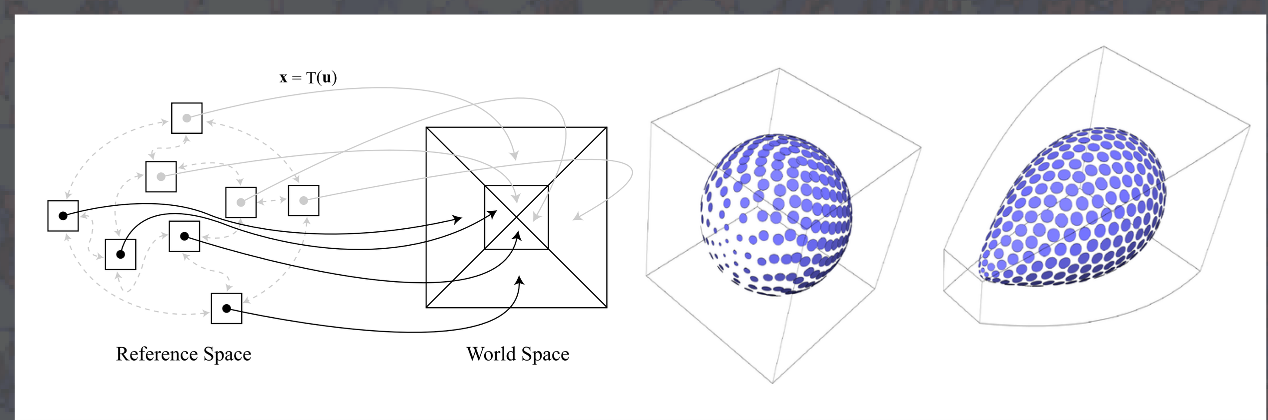
The University of Utah

Visualization has become an important component of the simulation pipeline, providing scientists and engineers a visual intuition of their models. However, simulations that make use of the finite element method for spatial subdivision present a challenge to conventional isosurface visualization techniques. Finite element isosurfaces are defined by basis functions in *reference space*, which give rise to a *world space* solution through a coordinate transformation, which does not necessarily have a closed-form inverse. Therefore isosurface rendering methods such as marching cubes and raytracing require a nested root finding approach which is computationally expensive. To this end, we propose visualizing these isosurfaces with a dynamic particle system. We present a framework that allows particles to sample an isosurface in reference space, avoiding the costly inverse mapping of positions from world space when evaluating the basis functions. The distribution of particles across the reference space isosurface is controlled by geometric information from the world space isosurface, such as the surface gradient and curvature. The resulting particle distributions can be distributed evenly or adapted to accommodate world-space surface features. This provides compact, efficient, and accurate isosurface representations of these challenging data sets.

High-Order Finite Elements

The method of finite elements is a common spatial subdivision scheme used by scientists and engineers to reduce large simulation domains to sets of small subdomains over which physical simulations can be computed robustly and efficiently. While traditional finite element methods utilize only low-order, linear basis functions for representing data over the elements, they provide considerable flexibility for handling complex geometries. The geometric flexibility is aided by the transformation of individual elements constructed as identical cubes in *reference space* into unique *world space* elements, which can have not only rectangular faces, but also triangular faces. In world space, the spatial extent of each element is defined by characteristics of the domain and simulation, such as boundary conditions and features of interest. The mapping functions responsible for the transformations can distort the elements by stretching, skewing, or even collapsing the faces of the reference space cubes, as illustrated in the following images.

A number of researchers have developed methods to improve the convergence properties of finite elements through the use of high-order functions for the representation of the data, as well as the element transformations. Today, high-order finite element techniques have reached a level of sophistication such that they are commonly applied to a broad range of engineering problems. Conventional approaches to finite element isosurface visualization assume that linear data representations can be adapted to accommodate low-order finite elements. This strategy, however, faces a number of challenges when considering high-order data sets. First, the data must be finely subsampled to ensure that features are adequately captured with linear approximation schemes. Second, there is, in general, no closed form expression for the inverse of high-order mapping functions. Numerical inversion schemes are required to transform world space locations into the reference space when sampling the data. This results in a nested root-finding problem when locating an isosurface, because valuating the solution requires a numerical inverse of the coordinate transformation. Furthermore, determining which reference-space element in which to invert a particular point in world space adds to the computation.



The left image is a schematic of how elements defined in reference space are mapped into world space, with the effects of the mapping function illustrated as the center image is mapped to the right image with a curvalinear function.

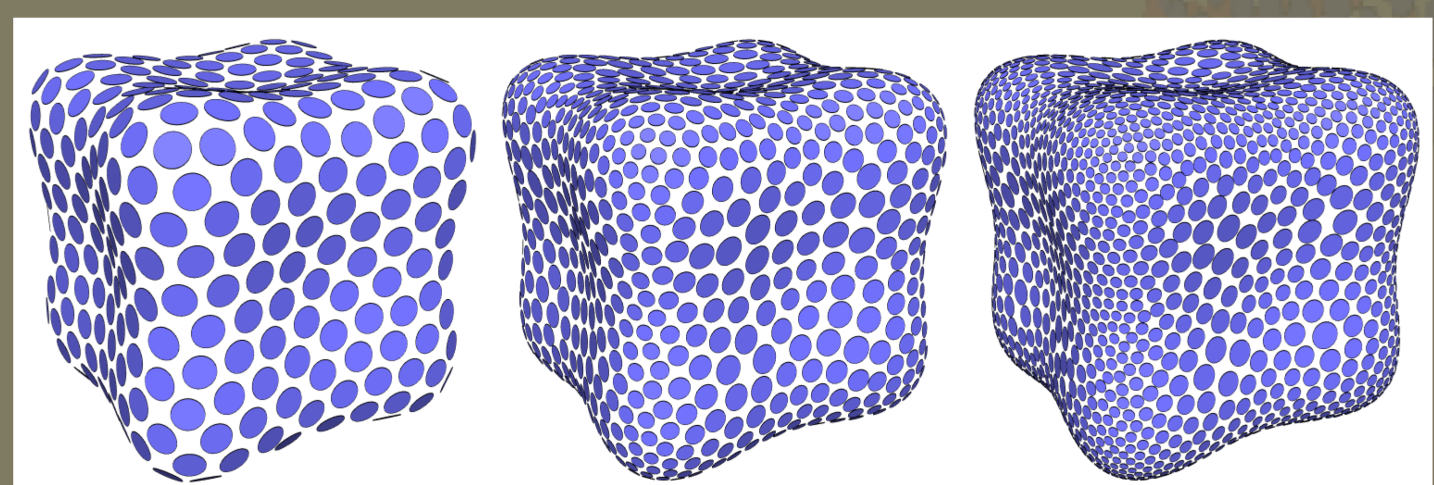
Computational scientists who wish to visualize high-order finite element solutions will require visualization algorithms that are flexible enough to accommodate these constraints. These algorithms will need to have variable degrees of freedom so that users can easily control the trade-off between visualization quality and speed. For efficiency, these computations must be locally adaptive, allowing computational power to be applied to regions of the solutions that exhibit the most complexity. Furthermore, these algorithms will need to achieve the appropriate balance of computations in world space, where the metrics for adaptivity are defined, and reference space, where there are closed-form expressions for the associated geometric quantities. To address these issues we are proposing an isosurface visualization technique that relies on a particle system.

Particle System Framework

To distribute the particles across an isosurface, we use the framework proposed in [2], which builds on the work of [3]. In this framework, the particles are constrained to an isosurface using a Newton-Rhapson root finding scheme. For each particle on the surface we associate a compact, potential energy kernel, which decreases monotonically with the distance from the particle. These kernels give rise to a computation of the force at a particle based on the distances to the neighboring particles, which is used to move the particle a lower energy state. By iteratively moving the particles along their energy gradients the system converges to an even distribution across the surface.

To increase the efficiency of the system we adapt the distributions to allow higher densities of particles around complex surface features. This is accomplished by scaling the distance between neighboring particles base on the local geometry of the surface. For instance, if we allow surface curvature to increase the effective distance between particles, areas of high curvature will have a higher density of particles. The following images illustrating varying degrees of adaptivity.

Applying the particle system framework to higher order finite elements requires several modifications. First, distribution adaptivity to computed using the curvature of the surface. For higher order finite elements, the curvature calculation is a complex formulation of second derivatives of not only the basis functions but also the mapping functions, which results in a series of vector-matrix-tensor products. We have reduced this formulation into a series of standard vector-matrix computations by using Einstein notation and Feynman diagrams. Second, to avoid the costly compuations of numerically inverting the mapping function we have developed a strategy that maintains particle positions in reference space, while controlling the particle distributions with geometric information from the world space. We ensure that the particle distributions are even and adaptive in the world space by using world space positions when computing the repulsive forces.

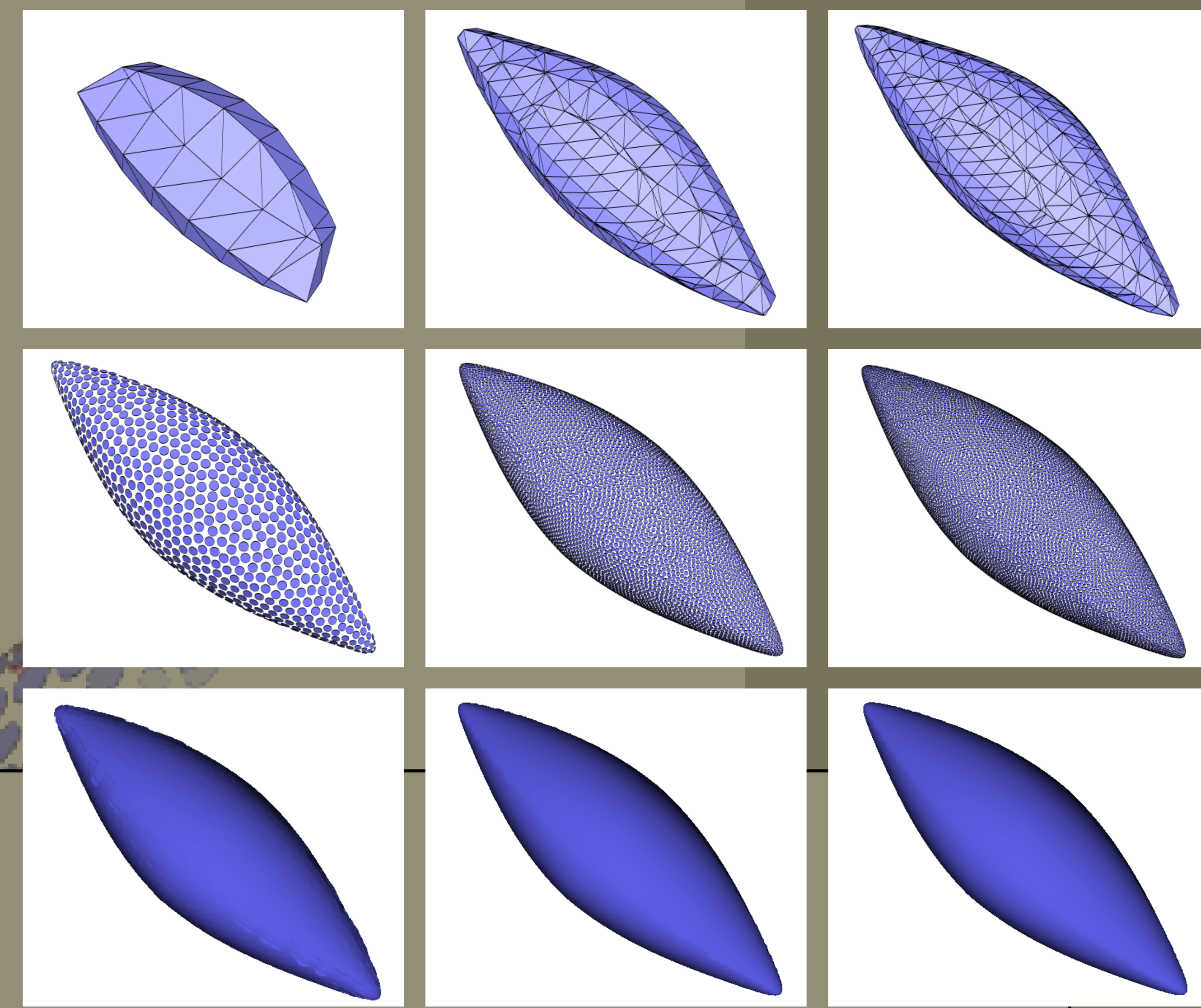


The results of varying the adaptivity of the particle system.

Comparison with Marching Cubes

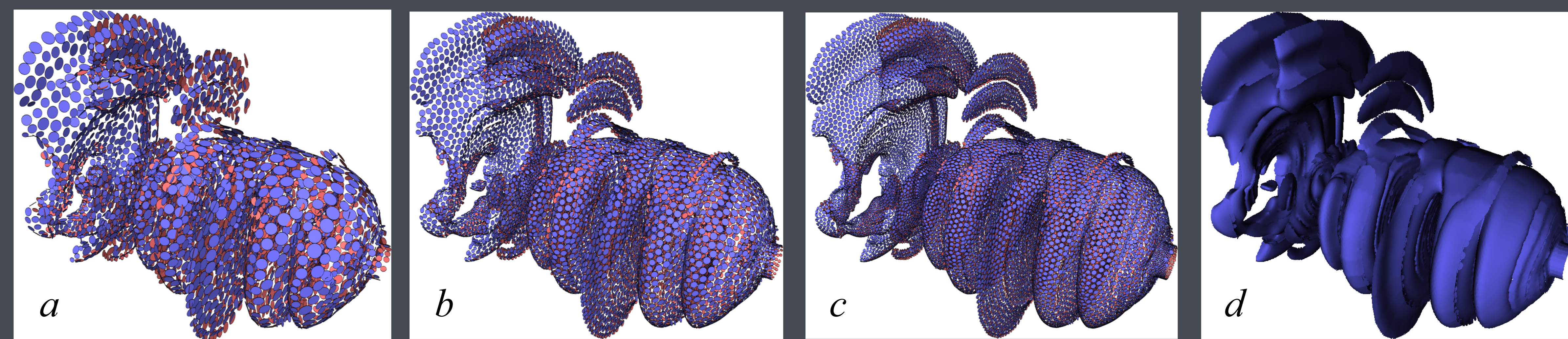
Applying standard linear-based visualization methods to high-order finite elements presents several challenges. First, high-order basis functions represent features in the data with far fewer grid elements than an equivalent low-order representation. The second problem stems from the need to numerically compute an inverse of the mapping function to evaluate the data in the world space. And third, the data and coordinate transformations are valid for only a single element, and in practice another layer of computation is required to determine which reference element contains the point under consideration.

Adapting marching cubes[1] to accommodate high-order finite elements elucidates these three challenges. In the following images we present the results of several different resolutions of the isosurface extraction technique applied to a sphere that is transformed through a $2\times2\times2$ grid of quadratic B-Spline functions. We compare the results with what our particle system achieves in approximately the same amount of time.

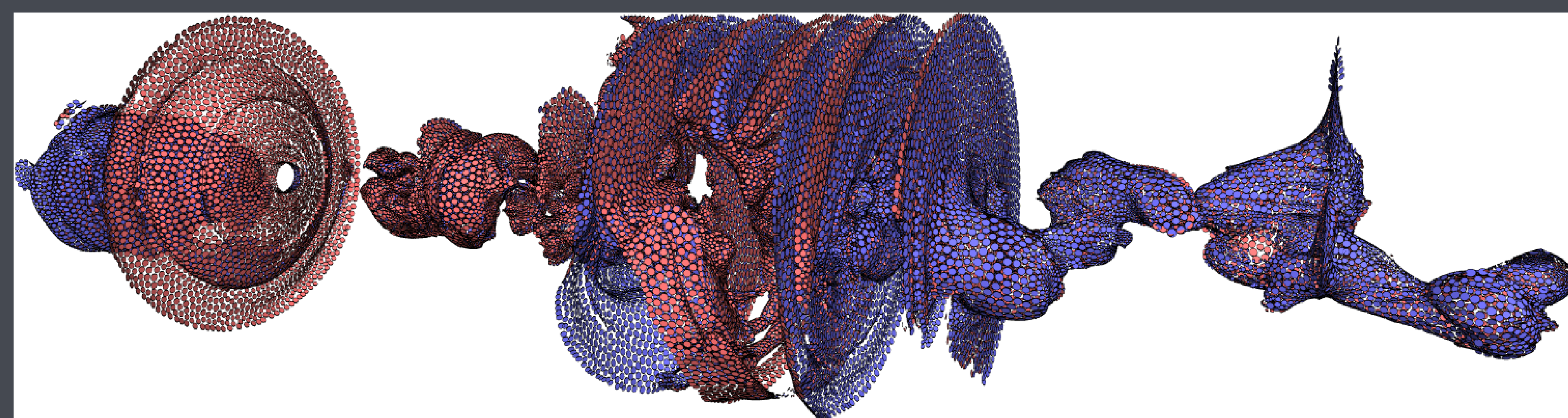


Comparison of marching cubes with particles distributed in the same amount of time. The bottom row is a water tight surface generated by a splatting algorithm on the GPU. (left column) 13 seconds; (middle column) 70 seconds; (right column) 105 seconds.

Results



The isosurface of pressure C = 0 for a CFD simulation over 5736 elements with a third-order polynomial implicit function in each element. (a) 5000 particles, 55 seconds. (b) 13,000 particles, 3.4 minutes. (c) 28,000 particles, 15 minutes. (d) 59,000 particles, 39 minutes.



The same data set as above, visualizing the pressure value of C = -0.05.

References

[1] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 163–169.

[2] Miriah D. Meyer, Pierre Georgel, and Ross T. Whitaker. Robust particle systems for curvature dependent sampling of implicit surfaces. In *Proceedings of the International Conference on Shape Modeling and Applications (SMI)*, pages 124–133, June 2005.

[3] Andrew P. Witkin and Paul S. Heckbert. Using particles to sample and control implicit surfaces. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 269–277.

This research was supported by the NSF grant CCR-0310705, the ARO grant DAAD19-01-10013, and the ARO grant W911NF-05-1-0395.