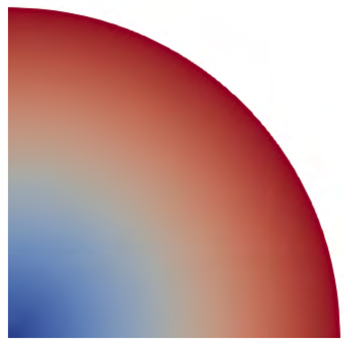
Implementation of rational Bézier cells into VTK

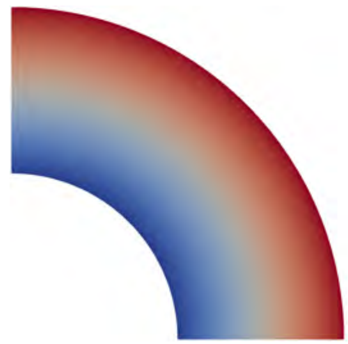
Coreform LLC

March 13, 2020

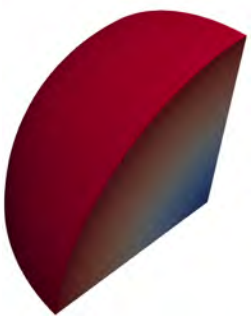
(a) VTK_BEZIER_CURVE (quadratic)



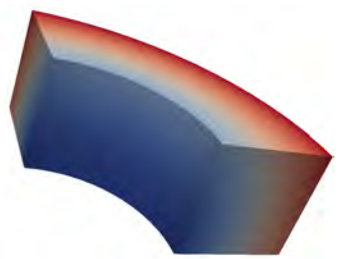
(b) VTK_BEZIER_TRIANGLE (quadratic)



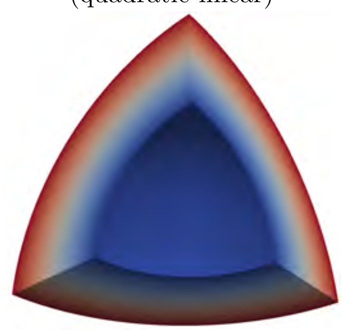
(c) VTK_BEZIER_QUADRILATERAL (quadratic-linear)



(d) VTK_BEZIER_TETRAHEDRON (quartic)



(e) VTK_BEZIER_HEXAHEDRON (quadratic-bi-linear)



(f) VTK_BEZIER_WEDGE (quadratic-linear)

Figure 1: Illustration of the six new Bézier cells that are now available in vtk, featuring capabilities for higher-order degree and rational weights. One benefit of rational Bézier cells is that they can exactly represent conic geometries.

1 Introduction

Spline-based simulation, sometimes called isogeometric analysis (IGA), is a method for running engineering simulations directly on the same spline basis functions used to define computer-aided design (CAD) models. Spline-based simulation is growing in popularity both in academia and industry. Since the method was first published in 2005, there have been over 2000 additional papers published, making this one of the fastest growing fields of research in finite element analysis. Commercial companies across industries such as automotive, defence, aerospace, nuclear energy, mining, and 3D printing are exploring these methods. A number of simulation companies are also investigating how to integrate spline-based simulation into their solvers, notably Ansys (with their LS-DYNA code) and Coreform LLC, a startup focused on developing a native spline-based simulation solver. Potential benefits of spline-based simulation include higher accuracy with fewer degrees of freedom, and increased simulation robustness, and reduced computation time. All commercial spline definitions, such as Non-Uniform Rational B-Splines (NURBS), T-splines, U-splines, etc. are comprised of Bèzier elements. Coreform and LS-DYNA have jointly defined an open source Bezier Extraction file format that any solver using splines can export for interoperability and to exchange data. Preprocessors such as Coreform Flex can export structured and unstructured spline models via Bèzier extraction for use in spline-compatible solvers.

In the most recent released of VTK, Coreform worked together with Kitware to introduce six new Bézier cells into VTK. This is significant in that now, for the first time, smooth spline simulation results can be accurately visualized in Paraview instead of sampled as a relatively coarse mesh. This is a significant improvement in the accurate visualization of the simulation results. Higher-order Lagrange elements had been previously added to Paraview, and it can be helpful to understand the technical considerations of the Bèzier cells by comparing them with these Lagrange counterparts.

Non-rational Bézier functions can be expressed in terms of non-rational Lagrange functions and visa versa. In other words, an object modeled with a non-rational Lagrange basis can be also modeled with a non-rational Bézier basis. An example is provided in Fig. 2 where Bézier and Lagrange quadratic curves are overlapping. Note that the parameterization (the position of the control points) is different for these two curves. For Lagrange, all the points lie on the curve whereas for the Bézier curves, only the points at the extremities are interpolatory. This is the first main difference between these both parameterizations, and we will see in Section 4 what the implications of non-interpolatory points are in terms of the implementation in VTK.

Control points that parametrize a Bézier or a Lagrange object can be associated to rational weights. These new rational functions are nonlinear and as a consequence, it is not possible to express rational Lagrange functions in terms of rational Bézier ones. Whereas rational Bézier functions are widely used to exactly represent conical shapes, rational Lagrange functions are not. NURBS functions, which can be exactly decomposed into Bézier objects, are the dominant representation used in CAD software. An example of an arc modeled with a quadratic Bézier curve is shown in Fig. 2 (this is one third of a circle, which is actually the maximum portion of a circle that can be represented with a single Bézier element).

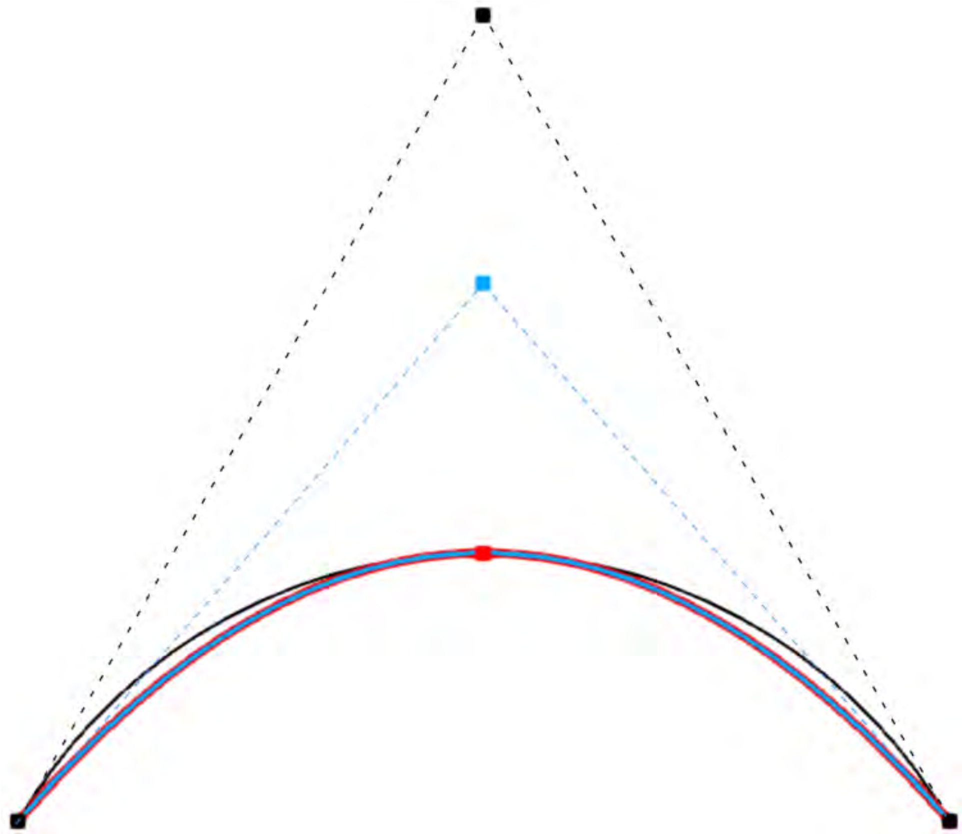


Figure 2: Comparison between a Lagrange curve (red), and Bézier curve (blue) and a rational Bézier curve (black). Only the rational Bézier curve can represent exactly the portion of the circle.

2 Lagrange and Bézier cells implemented in VTK

Table 2 present the different Lagrange and Bézier cells implemented in VTK.

Cell type	Cell Id	Rational support	Anisotropic degree support
VTK_LAGRANGE_CURVE	68	no	no
VTK_LAGRANGE_TRIANGLE	69	no	no
VTK_LAGRANGE_QUADRILATERAL	70	no	yes
VTK_LAGRANGE_TETRAHEDRON	71	no	no
VTK_LAGRANGE_HEXAHEDRON	72	no	yes
VTK_LAGRANGE_WEDGE	73	no	yes
VTK_BEZIER_CURVE	75	yes	no
VTK_BEZIER_TRIANGLE	76	yes	no
VTK_BEZIER_QUADRILATERAL	77	yes	yes
VTK_BEZIER_TETRAHEDRON	78	yes	no
VTK_BEZIER_HEXAHEDRON	79	yes	yes
VTK_BEZIER_WEDGE	80	yes	yes

While Lagrange elements were already implemented into VTK, we added the possibility for some cell types to have anisotropic degree, meaning that different directions of the cell can possess different degrees.

We added rational support only for Bézier cells. While rationality can be easily extended to Lagrange cells, we are not aware of applications or demand for this functionality.

3 How to define the rationality and anisotropic degree

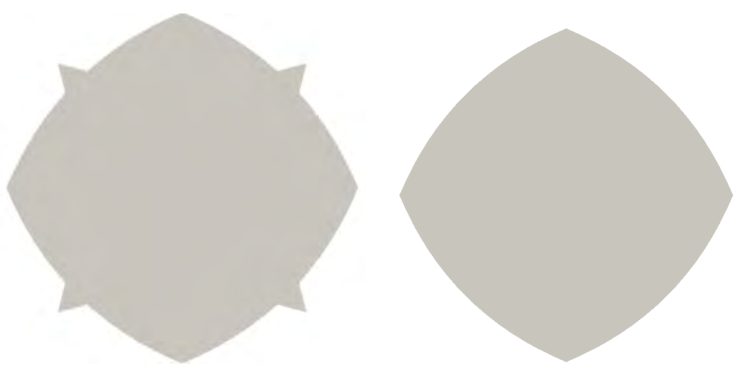
Both rationality and the anisotropic degree are optional attributes. If nothing is specified, the cell is assumed to be of uniform degree and non-rational, maintaining the compatibility with the already existing Lagrange cells.

For Bézier rational cells, a weight is defined at each control point, and so the data can be accessed via a *VTKPointData* with the attribute *RATIONALWEIGHTS*. To specify the anisotropic degree, a vector of size three for each cell must be used to define the degree in each direction (in the two-dimensional case, the third component is required but not used). The anisotropic degree can be accessed via *VTKCellData* with the attribute *HIGHERORDERDEGREES*. Several examples are provided in the VTK file *TestBezier.cxx*.

4 Implications of non-interpolatory control points

This is the first time that non-interpolatory quantities have been implemented into VTK. Non-interpolatory means that while the geometry or the field is defined by the control points, these control points do not lie on the geometry (see the Bézier curves in Fig. 2). Nonlinear cell representations in VTK can be refined using a tessellation filter or a non-linear subdivision level. To reduce the number of points to be computed, VTK reuses the intermediate control points of a cell. For Bézier cells, this results in the visualization of spurious peaks as shown in Fig. 3. This issue has been fixed in the implementation by overwriting the intermediate points for Bézier cells only.

There is currently an outstanding issue that occurs when the automatic *Rescale to Data Range* is used on a field displayed via a nonlinear subdivision level. The extremes are computed from the points data, but again, these points can be non-interpolatory, leaning to overestimated values. To overcome this small issue, the scale can be set using a custom range, or instead of using the nonlinear subdivision level for the representation, a tessellation filter can be used.



(a) (b)

Figure 3: Quadratic Bézier cell before (a) and after (b) we introduce fixes to support non-interpolatory control points.

5 Lagrange and Bézier node numbering

The node numbering for Lagrange and Bézier cell is the same. Because we noticed some mismatches between the original documentation (`https://blog.kitware.com/wp-content/uploads/2018/09/Source_Issue_`

43.pdf) and the implementation for hexahedra and tetraedra, for clarity we review the the node numbering that is used in the code.

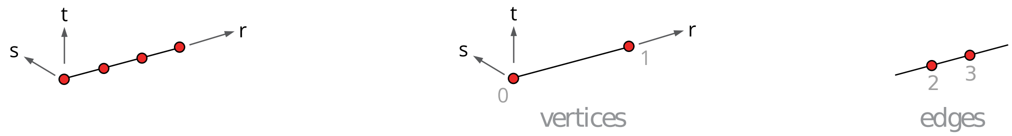


Figure 4: Lagrange or Bézier curve node numbering.

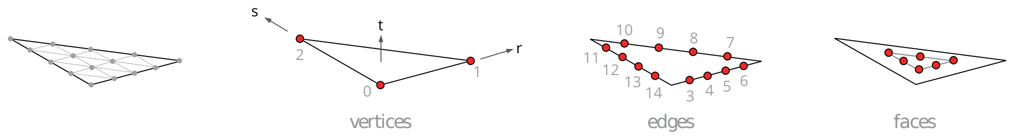


Figure 5: Lagrange or Bézier triangle node numbering.

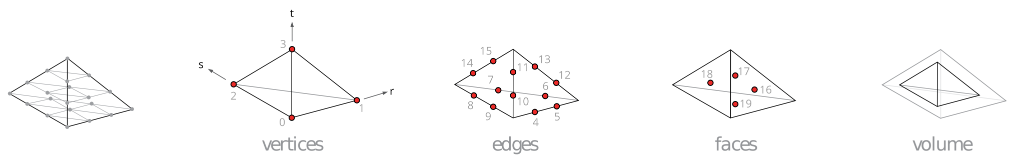


Figure 6: Lagrange or Bézier tetrahedron node numbering.

Figure 7: Lagrange or Bézier quadrilateral node numbering.

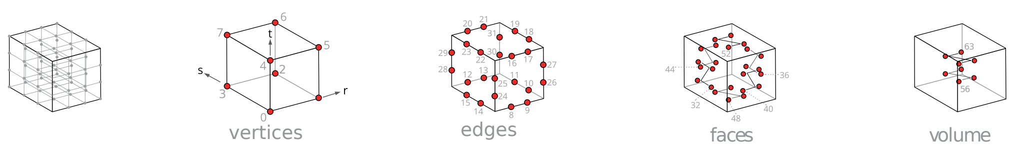


Figure 8: Lagrange or Bézier hexahedron node numbering.

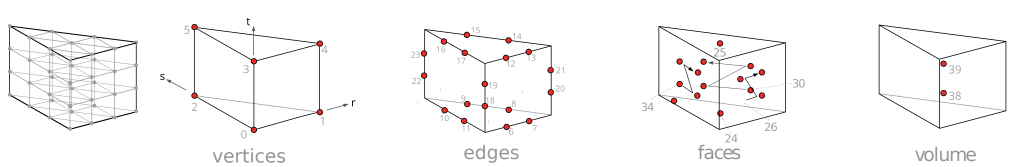


Figure 9: Lagrange or Bézier wedge node numbering.

6 Some examples of Bézier cells representing exact conic shapes

Figure 10: Quadratic VTK_BEZIER_TRIANGLE cell that represents exactly a full disk (this is actually the minimal parameterization of an exact full disk). Nonlinear subdivision level 0, 1 and 5.

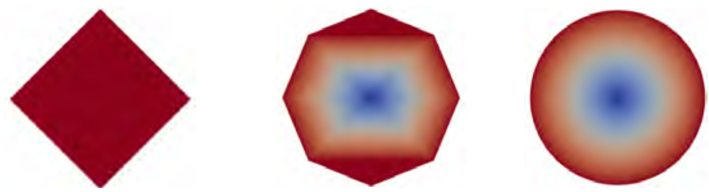


Figure 11: Quadratic VTK_BEZIER_QUADRILATERAL cell that represents exactly a full disk. Nonlinear subdivision level 0, 1 and 5.

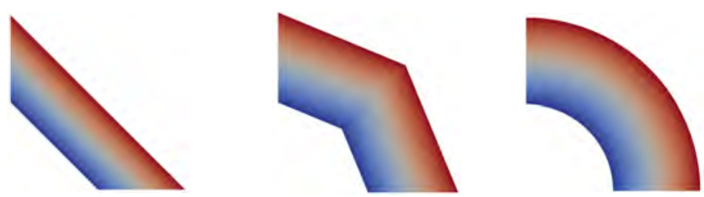


Figure 12: Multi-degrees linear-quadratic VTK_BEZIER_QUADRILATERAL cell that represents exactly a quarter of a disk. Nonlinear subdivision level 0, 1 and 5.

Figure 13: Quadratic VTK_BEZIER_TETRAHEDRON cell that represents a kind of cone. Nonlinear subdivision level 0, 1 and 5.

Figure 14: Quartic VTK_BEZIER_TETRAHEDRON cell that represents exactly a solid sphere octant. Nonlinear subdivision level 0, 1 and 5.

Figure 15: Quartic VTK_BEZIER_HEXAHEDRON cell that represents exactly a sphere. Nonlinear subdivision level 0, 1 and 5.

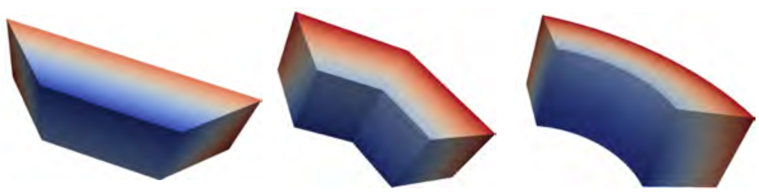


Figure 16: Multi-degrees quadratic-bi-linear VTK_BEZIER_HEXAHEDRON cell that represents exactly quarter ring with a square cross-section. Nonlinear subdivision level 0, 1 and 5.

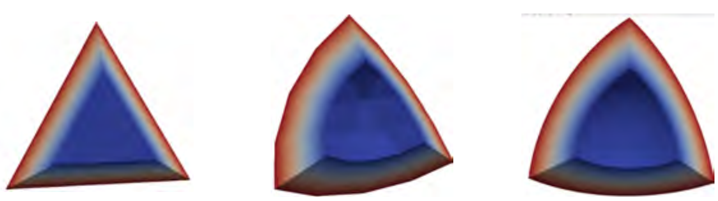


Figure 17: Multi-degrees bi-quartic-linear VTK_BEZIER_WEDGE cell that represents exactly a thick sphere octant. Nonlinear subdivision level 0, 1 and 5.

7 An example of application

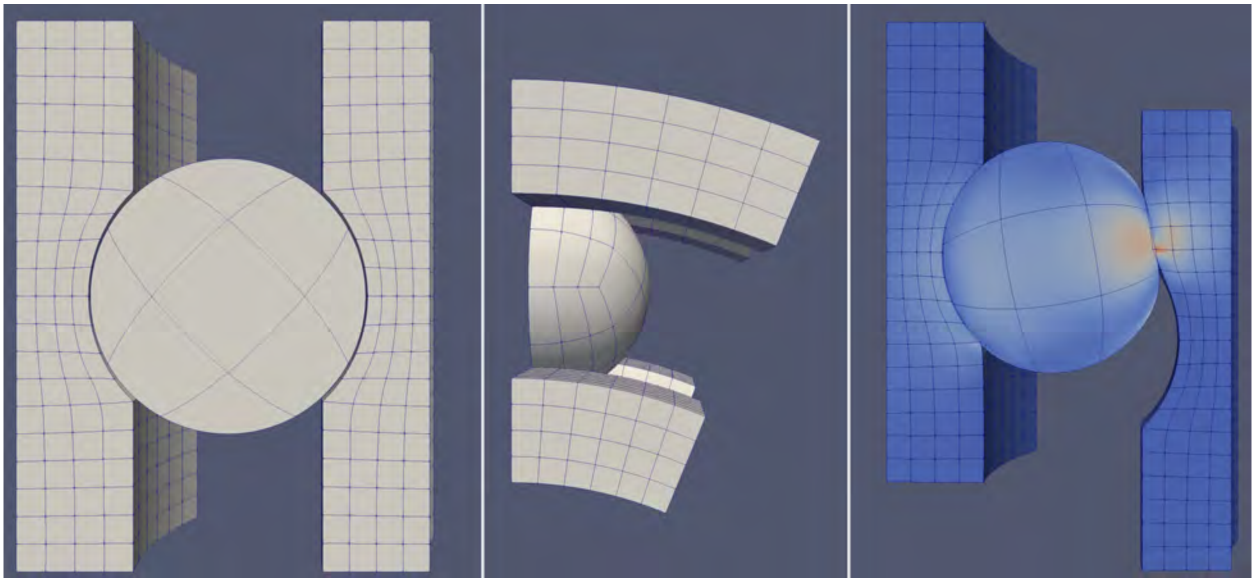


Figure 18: A ball bearing shearing simulation performed by Coreform. Accounting for the symmetries, only one sixteenth of the ball bearing is represented. The inner and geometries are represented exactly using tri-quadratic Bézier hexahedra, and the half sphere is also modeled exactly using tri-quadtric Bézier hexahedra (a single element could have been used without losing the geometry exactness, but several elements were required here for the simulation accuracy)

8 Acknowledgements

This work was found by the Office of Science, U.S. Department of Energy under the SBIR project number DE-SC0019617.

The ball bearing application was provided by Honeywell Federal Manufacturing & Technologies under Contract No. DE-NA0002839 with the U.S. Department of Energy.

T.J. Corona from Kitware is gratefully acknowledged for his help and review of the code.