Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 00 (2015) 000–000

Procedia Engineering

www.elsevier.com/locate/procedia

24th International Meshing Roundtable (IMR24)

# A comparison of $C^0$ and $G^1$ continuous curved meshes on high-order finite element simulations

Daniel W. Zaide*, Qiukai Lu, Mark S. Shephard

*Scientific Computation Research Center, Rensselaer Polytechnic Institute, Troy, NY, 12180-3590, USA*

## Abstract

This paper examines the effect of curved boundary continuity on finite element simulations. Quartic $C_0$ continuous and $G_1$ continuous meshes with curved boundaries are created from CAD geometry. These $C_0$ continuous and $G_1$ continuous surface triangles are constructed using Bézier triangles and triangular Gregory patches respectively. Curved tetrahedra are constructed using a shape blending with tetrahedral mapping. The curved meshes are solved on using a high-order finite element method. The effect of surface continuity on solution accuracy is quantified for several fluid problems in three dimensions. While $G^1$ continuous boundary patches can improve solution accuracy, there are still open questions and further investigation is needed.

*Keywords:* Curved Meshes; Bézier Surfaces; High-order Meshing;

## 1. Introduction

In order to fully realize the benefits of high-order methods, the curved portions of the geometric domain must be properly represented by the computational mesh. Previous analyses using the relation of approximation theory to the convergence of the error in the energy norm have indicated that numerical solutions will converge as the geometric approximation of the mesh is within one order of that used in the finite element basis [1]. Thus, for proper solution accuracy, the mesh edges and faces representing curved domain geometry must be curved and provide a sufficient order of geometry approximation to prevent the loss of convergence and accuracy due to the geometric approximation [2–5].

Early attempts to accurately represent curved domains for finite element simulations date back to the 1970s when the isoparametric element approach was introduced to solid mechanics applications [6]. With the rapid development of the Computer-Aided Geometric Design (CAGD) technology, researchers started to work on integration of CAD technology for geometrical representation with finite element analysis methods. Schramm and Pilkey [7] used Non-Uniform Rational B-Splines (NURBS), the industry standard for geometric modeling, for geometry to implement transfinite elements and applied it to shape optimization. A number of different techniques have been proposed in

[*] Corresponding author.
*E-mail address:* dan.zaide@gmail.com

recent years to generate curvilinear meshes based on high-order polynomial mappings and optimal nodal placement [8–11]. More recent work on tetrahedral meshes by Sevilla et al [12] have used NURBS to create a NURBS-enhanced FEM method with NURBS based elements near the CAD model boundary and standard polynomial based finite elements for the interior, demonstrating its improvement on several problems with Maxwell's equations.

Alongside this research, in the past ten years extensive research has been on isogeometric analysis, integrating CAD and finite element methods using the same representation for both the geometry and the finite element solution. First introduced by Hughes et al in 2005 [3,13], these approaches were initially applied with NURBS but have since been extended to other tensor-product geometric representations [14,15] and demonstrated on a wide range of physical problems from structural mechanics to fluid-structure interaction problems. Isogeometric approaches have shown improvements for representing quadrilateral and hexahedral meshes, however their application to triangular meshes is only beginning to be explored. In 2012, Speleers et al [16] used quadratic Powell-Sabin splines (NURPS) for advection-diffusion problems on triangular domains and then in 2014, Jaxon and Qian [17] used rational triangular Bézier splines to represent their two dimensional geometries. With the challenges of isogeometric analysis on triangles, we focus on an alternative approach for tetrahedral meshes.

On most curved tetrahedral meshes, representations of triangular boundary faces focus on treating each face as an individual curved element, resulting in $C^0$ continuous surface patches. By considering neighboring boundary faces, a smoother representation of the geometry can be derived. $G^1$ geometrically continuous surfaces have been around since 1974 when Gregory [18] introduced the idea for rectangular surfaces. Walton and Meek [19] later developed the triangular version used in this work, and it has since been extended and applied [20,21] to higher orders for computer graphics applications. Geometrically continuous surfaces have been used in meshing applications as well, with Owen et al [22] using them to create a smooth geometry representation from a faceted input in the common geometry module and Frey [23] using them to support curved surface meshes and provided enhanced geometric information in the absence of underlying CAD. It has also been previously determined that certain types of physical applications, such as electromagnetic scattering and compressible flow applications, are sensitive to the accuracy and smoothness of the computational mesh that approximates the curved domain boundaries [3,4,12].

In this paper, we investigate the effects of surface triangle continuity on finite element simulations, comparing a $C^0$ continuous Bézier surface with a $G^1$ continuous surface representation. Our meshing software [24] is integrated into Nektar++ [25], a high-order finite element solver. Several curved meshes are then used for problems in fluid dynamics and the accuracy is quantified. This work will provide insight into future directions for more in-depth studies in node placement and curved surface representation as we continue to improve high-order meshes.

**Nomenclature**

? Fill in later!

## 2. Curved Surfaces

We compare the effect of surface patch continuity using $C^0$ and $G^1$ triangles. While $C^1$ continuity would be ideal between connected triangles, the constraints for constructing triangles of this nature make geometric continuity a suitable alternative. As the $G^1$ triangle is based on a Bézier triangle, we first provide a brief overview of Bézier triangles.

### 2.1. Bézier Curves

In computer aided geometric design, Bézier polynomials are frequently used to construct curves and surface patches in parametric forms [26]. An $n^{th}$ order Bézier curve can be expressed as:

$$\mathbf{b}(t) = \sum_{i=0}^{n} b_i^{(n)}(t)\mathbf{P}_i^{(n)}, \qquad t \in [0,1], \tag{1}$$

where $b_i^{(n)}(t)$ are the $i^{\text{th}}$ Bernstein functions of degree $n$,

$$b_i^{(n)}(t) = \binom{n}{i} t^i (1-t)^{n-i}, i = 0, ..., n, \qquad \binom{n}{i} = \frac{n!}{i!(n-i)!}. \tag{2}$$

and $\mathbf{P}_i^{(n)}$ is the $i^{\text{th}}$ control point of an $m^{\text{th}}$ order Bézier curve. The derivative of the curve is also a Bézier curve, of $(n-1)^{\text{th}}$ order. Higher-order derivatives are also Bézier curves, and can be obtained in a similar manner. Bézier polynomials have several other properties that lend themselves to curved meshing [26]:

1. The convex hull property. The max and min values of a Bézier polynomial evaluated within the domain are bounded by the max and min values of its corresponding control points.
2. Variation diminishing. The curve has no more intersections with any plane than polygon of its control points, and converges to the curve.
3. Affine invariance. Curves form an affine map, allowing for reparametrization.
4. Degree elevation and subdivision. Curves can be increased to higher-orders without changing behavior of the curve.

For representations of surfaces, Bézier triangles are analogous to their curves and of the form

$$\mathbf{B}(u, v, w) = \sum_{i,j,k \geq 0}^{i+j+k=n} B_{ijk}^{(n)}(u, v, w) \mathbf{P}_{ijk} \tag{3}$$

with Bernstein polynomials for a triangle,

$$B_{ijk}^{(n)}(u, v, w) = \frac{n!}{i!j!k!} u^i v^j w^k \tag{4}$$

and $u, v, w = 1 - u - v$ are barycentric coordinates, and $\mathbf{P}_{ijk}$ are the $\frac{1}{2}(n+1)(n+2)$ control points. Each edge of the Bézier triangle is itself a Bézier curve, allowing for a the triangle to be deconstructed into its edge control points and its $\frac{1}{2}(n-1)(n-2)$ internal control points. As with the curve, the derivatives can also be written in terms of Bernstein polynomials [27].

To determine node placement on the geometry, we use interpolating Bézier triangles, first determining interpolation points using points in the surface parametrization from Chen and Babuška [28] and then determining the control points by solving the system of equations

$$\begin{aligned}
\mathbf{L}_1 &= \sum_{i,j,k \geq 0}^{i+j+k=n} B_{ijk}^{(n)}(u_0, v_0, w_0) \mathbf{P}_{ijk} \\
&\vdots \\
\mathbf{L}_N &= \sum_{i,j,k \geq 0}^{i+j+k=n} B_{ijk}^{(n)}(u_N, v_N, w_N) \mathbf{P}_{ijk}
\end{aligned} \tag{5}$$

set at the $N = \frac{(n+1)(n+2)}{2}$ interpolating points. This leads to a linear system of equations with a matrix of Bernstein polynomials, evaluated in parametric space. This matrix can be inverted once, and reused to determine control points for each curved triangle. This approach allows us to use arbitrary order interpolating Bézier triangles, making use of information provided from our geometric representation.

### 2.2. $G^1$ Surfaces

To construct $G^1$ surfaces, we follow the work of Walton and Meek [19]. Their approach results in a $G^1$ continuous quartic surface, represented by 18 control points, three more than required for a similar Bézier triangle. Geometric continuity is obtained by constraining tangential derivatives across boundary edges. In order to obtain the required $G^1$ continuity, the cross-boundary tangent fields associated with the three mesh edges have to be satisfied simultaneously, thus requiring more degrees of freedom than a typical cubic triangular Bézier patch. As a result, the order of the polynomials representing the surface patch is increased from cubic to at least quartic $B^{(4)}(\xi)$, leading to the additional three control points. Had $C^1$ continuity been required, not only would the direction of the internal points be constrained, but also their distance to their edge. This is particularly problematic if adjacent triangles are of significantly different sizes.

Unlike the interpolating Bézier triangles, the edges of the $G^1$ triangle are determined using the the point locations and normals at each vertex to determine the cubic control points. The internal points, two associated with each edge, are then determined such that the any two adjoining triangles have a common tangent plane. We refer the reader to [19] for more details. While cubic boundaries could be used, a more consistent approach is to elevate these boundaries to quartics, such that we can evaluate it as if it were a 4th order Bézier, with

$$\mathbf{B}^{(4)}(\xi) = \sum_{i,j,k\geq 0}^{i+j+k=4} P_{ijk} b^4_{ijk}(\xi). \tag{6}$$

and the surface control points $P_{112}, P_{121}, P_{211}$ decomposed into affine combinations of the $G^1$ control points $G_{ij}$ and are calculated using

$$P_{112} = \frac{1}{\xi_0 + \xi_1}(\xi_0 G_{22} + \xi_1 G_{01}) \tag{7}$$

$$P_{121} = \frac{1}{\xi_2 + \xi_0}(\xi_2 G_{02} + \xi_0 G_{11}) \tag{8}$$

$$P_{211} = \frac{1}{\xi_1 + \xi_2}(\xi_1 G_{12} + \xi_2 G_{21}) \tag{9}$$

As this approach is 4th order, we note that Farin and Hansford [21] and Tong and Kim [20] have extended this to higher dimensions.

## 3. Construction of Curved Meshes

The construction of curved meshes relies on a type of mesh modification operation, referred to as entity geometry modification, which modifies the geometry of a given mesh entity. We begin with an initially linear mesh of the CAD model along with the model itself. Our algorithms for curved mesh construction are as shown in algorithms 1 and 2.

```
for each boundary edge and face in the mesh do
    determine internal face interpolation points;
    Calculate internal face control points by solving Eq 5;
end
```

**Algorithm 1:** Creating $C^0$ continuous curved meshes.

### 3.1. Tetrahedral Blending

In our curved mesh, for tetrahedra with curved faces, we use the blending approach of Dey et al [1]. In the conventional isoparametric approach with $C^0$ meshes, the volumetric mapping between a standard parametric space and the physical space is constructed based on the same polynomial basis functions used for the finite element space. However, the basis functions used to represent the rational $G^1$ curved mesh are generally not the same as the finite element shape functions used for analysis. Therefore, a more general approach is adopted to construct the volumetric mapping in order to account for the $G^1$ surface geometry. The shapes of lower dimensional mesh entities bounding

```
for each boundary edge in the mesh do
    if the edge represents a model edge then
        determine edge control points using model edge tangents;
    end
    if the edge represents a model face then
        determine edge control points using model face normals;
    end
end
for each boundary face in the mesh do
    determine face control points $G_{ij}$ using edge control points;
end
```

**Algorithm 2:** Creating $G^1$ continuous curved meshes.

the element volume are multiplied with polynomial blending functions, and the contributions are summed together to get the complete volume mapping.

The equation for the mapping is given in Eq 10 as

$$\begin{aligned} x_i(\boldsymbol{\xi}) = {} & (1-\xi_1)^{k+1}F_1(\xi'_{F_1}) + (1-\xi_2)^{k+1}F_2(\xi') + (1-\xi_3)^{k+1}F_3(\xi') + (1-\xi_4)^{k+1}F_4(\xi') - (1-\xi_1-\xi_2)^{k+1}E_1(\xi') \\ & -(1-\xi_1-\xi_3)^{k+1}E_2(\xi') - (1-\xi_1-\xi_4)^{k+1}E_3(\xi') - (1-\xi_2-\xi_3)^{k+1}E_4(\xi') - (1-\xi_2-\xi_4)^{k+1}E_5(\xi') \\ & -(1-\xi_3-\xi_4)^{k+1}E_6(\xi') + \xi_1^{k+1}V_1 + \xi_2^{k+1}V_2 + \xi_3^{k+1}V_3 + \xi_4^{k+1}V_4 \end{aligned} \tag{10}$$

where $\boldsymbol{\xi} = (\xi_1, \xi_2, \xi_3, \xi_4)$ are barycentric coordinates and $F_i$, $E_i$, and $V_i$ are the positions corresponding to the bounding faces, edges, and vertices. $\xi'$ defines a normalized parametric coordinate on the edge or face, for example on edge $E_1$, $\xi_1 = \xi_2 = 0$, and $\xi' = (\xi'_1, \xi'_2) = \left(\frac{\xi_3}{\xi_3+\xi_4}, \frac{\xi_4}{\xi_3+\xi_4}\right)$. The blending approach is independent of the chosen face and edge parametrization and thus can be used with both types of curved faces. The choice of $k$ in the blending functions determines the smoothness of the blending functions, and it can be shown that this particular volumetric mapping is $C^k$ continuous. In this work, we use $k = 1$ for a $C^1$ continuous mapping, though in the future this can be investigated in more detail.

## 4. Implementation

The curving procedure and mesh is implemented in the Parallel Unstructured Mesh Infrastructure (PUMI) [24]. Developed as part of the set of tools within the DOE SciDAC FASTMath institute, PUMI is a set of C and C++ libraries that implement data structures to represent parallel finite element meshes, and provides functions for querying and modifying parallel meshes. PUMI also provides an interface with CAD libraries, in this work, we use Parasolid for geometric information. Control points are stored with their respective mesh entities, with vertices, edges, and faces all storing control point information. This avoids duplication of control points, at the cost of additional work to collect and organize control points for surface location and gradient evaluation. In future work, PUMI will allow us to easily perform mesh adaptation, load balancing, and other operations efficiently in parallel.

### 4.1. Finite Element Solver

For the finite element solver, we use Nektar++ [25], a high-order finite element code for solving partial differential equations. Nektar++ has a wide range of physics implemented, and is a well validated solver to test the effects of curved meshes. Nektar++ has already been used with curved meshes [8,29] and supports curved meshes defined by standard interpolation functions, describing curved entities by sets of interpolating points.

To integrate PUMI within Nektar++, only several small changes are required to allow our curved meshes to be used. First, the XML based input format is adjusted to contain information about the PUMI mesh and CAD model files, rather than contain the entire mesh and its connectivity. After the input file is read, the PUMI mesh and CAD model are loaded, and we overload the edge, triangular face, and tetrahedron geometry classes to store our mesh entities and query them for geometric information. The existing curved framework is used with support for deformed elements. This occurs in two places: first, the coordinates and second, the spatial derivatives. Both are set to call the PUMI API for any edge, face, or tetrahedral element rather than the original geometric functions.

## 5. Numerical Results

Several test problems are used to compare the effects of surface continuity. For each example, solution accuracy in the $L_2$ and $L_\infty$ norms is compared. As it has been previously suggested that the geometric approximation be within one order of the finite element basis, quartic $C^0$ and $G^1$ approximations are solved on with 4th, 5th, and 6th order finite element methods. Geometric accuracy is measured in the $L_\infty$ norm of the Hausdorff distance for each boundary triangle,

$$L_{\infty,geom} = \max_{M_i^2 \sqsubset G_j^2} H(M_i^2, G_j^2), \qquad H(A,B) = \max_{a\in A} \min_{b\in B} \|a-b\| \tag{11}$$

a measure of the furthest distance of any point on the boundary to its closest point on the geometric representation, the maximum interpolation error.

### 5.1. Poisson's Equation on a Sphere

As a first example, we compare the effects of geometric continuity for a simple problem on a sphere. Spherical geometry is particularly interesting, as Bézier polynomials cannot exactly represent the spherical geometry illustrated by the geometric interpolation error, plotted in Figure 1 for the curved mesh obtained from the illustrated linear mesh. In this example, the $G^1$ continuous triangle is a better approximation to the surface than the interpolating Bézier triangles, though the interpolation error asymptotes as the triangle order is increased.

On the 98 tetrahedra mesh shown, Poisson's equation is solved in cartesian coordinates as

$$-\Delta u(\mathbf{x}) = f(\mathbf{x}), \qquad \mathbf{x} \in \Omega_G \tag{12}$$

on a sphere of unit radius with $f(\mathbf{x})$ as a polynomial such that

$$u(\mathbf{x}) = \frac{5539}{371}(1-r)\exp(r^5), \qquad u(\mathbf{x}) = 0 \in \Gamma_G \tag{13}$$

scaled such that $\max u(r) = 1.0$, where $r = (x^2 + y^2 + z^2)^{1/2}$. The boundary condition is $u(\mathbf{x}) = 0$ and is enforced on all boundary faces. As the surface triangles increase in geometric accuracy, the boundary conditions become more accurate.

To measure solution error, the $L_2$ and $L_\infty$ norms of the approximate solution, $u^{h,p}$, within the domain are computed, defined as

$$\begin{aligned} L_2(u^{h,p}) &= \left(\int_{\Omega_M} |u^{h,p} - u_{exact}| \mathrm{d}\Omega_M\right)^{1/2} \\ L_\infty(u^{h,p}) &= \max_{\Omega_M} \left|u^{h,p} - u_{exact}\right| \end{aligned} \tag{14}$$

and are plotted in Figures 2 and 3 for second to seventh order finite element methods. As expected, the original linear mesh does not accurately represent the problem, resulting in a large error in both norms. As the order of the curved surface triangles is increased, the accuracy is in the solution levels off due to the finite interpolation error in the curving. This is expected, as the order of geometric approximation should be within one order of the solution [1]. Of particular interest, looking at only results for the quartic meshes in Figure 4 is in the quartic $C^0$ mesh has a lower $L_2$ norm than the $G^1$ mesh, but in the $L_\infty$ norm, this is the opposite.

### 5.2. Hagen-Poiseuille Flow

Hagen-Poiseuille flow describes a fully developed, laminar viscous flow through a circular pipe. In this example, the geometry is a cylinder with unit radius and length 10 in the $z$-direction. The exact solution for the $z$-velocity is

$$u_z(\mathbf{x}) = -\frac{1}{4\nu}\left(\frac{\mathrm{d}p}{\mathrm{d}z}\right)(R^2 - r^2), \qquad r^2 = x^2 + y^2 \tag{15}$$

with the $u_x = u_y = 0$ for a specified constant pressure gradient across the pipe. A pressure drop of one is chosen here such that

$$p(\mathbf{x}) = 1 - \frac{z}{10}. \tag{16}$$

The kinematic viscosity is chosen to be 0.025 such that the velocity profile is $u_z = 1 - r^2$. The inlet to the pipe consists of the velocity flow profile with the outlet specified by standard Neumann boundary conditions on velocity, and a pressure of zero. The initial linear mesh consists of 2134 tetrahedra is shown on the left of Figure 5. Results for quartic $C^0$ and $G^1$ curved meshes are shown on the right of Figure 5 with the geometric errors are in Table 5.2. On the cylindrical geometry, the quartic $C^0$ mesh has three orders of magnitude smaller interpolation error than the $G^1$ mesh. Looking at the error in $u_z$ in Figure 5, for the third and fourth order methods, the $G^1$ mesh does a better job than the $C^0$ mesh, however the error remains constant as $p$ is increased. For the fifth and sixth order methods, the $C^0$ mesh has a significantly better solution than the $G^1$ mesh.

| | $L_{\infty,geom}$ |
|---|---|
| $C^0_4$ | $6.231 \times 10^{-7}$ |
| $G^1_4$ | $6.242 \times 10^{-4}$ |

Table 1. Geometric error of the cylindrical mesh.

## 6. Conclusions and Future Work

In this paper, the effects of $C^0$ continuity compared to $G^1$ surface triangle continuity for curved tetrahedral meshes are examined. On several fluid problems, meshes with 4th order $G^1$ continuous surface patches do not significantly improve solution accuracy compared to similar fourth order $C^0$ continuous patches for finite element methods. Geometric continuity is far from the only factor at play, and further work is needed on optimal control point placement for interpolating Bézier triangles. In the absence of underlying CAD geometry, representing curved surfaces by geometrically continuous surface patches may prove beneficial, however if the full geometric representation is known, placing control points to best represent the geometry and minimize geometric interpolation error seems to be a better approach.

The next step for this research is to extend existing techniques for curved mesh refinement to these surface patches to allow for mesh refinement and coarsening, leveraging the subdivision and elevation properties of Bézier curves and triangles. This will allow for investigation of their effects on highly anisotropic problems, such as boundary layer problems. The extension of $G^1$ continuous patches to 5th order and higher will also be investigated, as multiple approach. As part of going to higher order geometry, optimal control point placement will be examined, focusing on both geometric mesh quality measures and their effect on solution accuracy.

## Acknowledgements

This work was supported by [INSERT GRANT NAME?]. The first author would like to thank Daniel A. Ibanez for assistance with software implementation and discussions on mesh reordering.

## References

[1] S. Dey, M. S. Shephard, J. E. Flaherty, Geometry representation issues associated with p-version finite element computations, Computer Methods in Applied Mechanics and Engineering 150 (1997) 39 – 55. Symposium on Advances in Computational Mechanics.

[2] X. Luo, M. S. Shephard, J.-F. Remacle, R. M. O'Bara, M. W. Beall, B. Szabo, R. Actis, P-version mesh generation issues, in: Proceedings of the 11th International Meshing Roundtable, Ithaca, NY., 2002, pp. 343–354.

[3] T. J. R. Hughes, J. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement., Computer Methods in Applied Mechanics and Engineering. 194 (2005) 4135–4195.

[4] L. Demkowicz, P. Gatto, W. Qiu, A. Joplin, G1-interpolation and geometry reconstruction for higher order finite elements, Computer Methods in Applied Mechanics and Engineering 198 (2009) 1198–1212.

[5] Q. Lu, M. S. Shephard, S. Tendulkar, M. W. Beall, Parallel mesh adaptation for high-order finite element methods with curved element geometry, Engineering with Computers 30 (2014) 271–286.

[6] O. C. Zienkiewicz, The Finite Element Method in Engineering Science, second ed. ed., London: McGraw-Hill, 1971.

[7] U. Schramm, W. Pilkey, The coupling of geometric descriptions and finite elements using NURBs – a study in shape optimization, Finite Elements in Analysis and Design. 15 (1993) 11–34.

[8] S. J. Sherwin, J. Peiro, Mesh generation in curvilinear domains using high-order elements, International Journal for Numerical Methods in Engineering 53 (2002) 207–223.

[9] P.-O. Persson, J. Peraire, Curved mesh generation and mesh refinement using lagrangian solid mechanics, in: Proceedings of the 47th AIAA Aerospace Sciences Meeting and Exhibit., 2009.

[10] A. Johnen, J.-F. Remacle, C. Geuzaine, Geometrical validity of curvilinear finite elements, in: Proceedings of the 20th International Meshing Roundtable, 2011.

[11] A. Gargallo-Peiró, X. Roca, J. Sarrate, A surface mesh smoothing and untangling method independent of the cad parameterization, Computational Mechanics 53 (2014) 587–609.

[12] R. Sevilla, S. Fernández-Méndez, A. Huerta, 3d nurbs-enhanced finite element method (nefem), International Journal for Numerical Methods in Engineering 88 (2011) 103–125.

[13] J. A. Cottrell, T. J. Hughes, Y. Bazilevs, Isogeometric analysis: toward integration of CAD and FEA, John Wiley & Sons, 2009.

[14] Y. Bazilevs, V. M. Calo, J. A. Cottrell, J. A. Evans, T. Hughes, S. Lipton, M. Scott, T. Sederberg, Isogeometric analysis using t-splines, Computer Methods in Applied Mechanics and Engineering 199 (2010) 229–263.

[15] A. Buffa, G. Sangalli, R. Vázquez, Isogeometric analysis in electromagnetics: B-splines approximation, Computer Methods in Applied Mechanics and Engineering 199 (2010) 1143–1152.

[16] H. Speleers, C. Manni, F. Pelosi, M. L. Sampoli, Isogeometric analysis with powell–sabin splines for advection–diffusion–reaction problems, Computer methods in applied mechanics and engineering 221 (2012) 132–148.

[17] N. Jaxon, X. Qian, Isogeometric analysis on triangulations, Computer-Aided Design 46 (2014) 45–57.

[18] J. A. Gregory, Smooth interpolation without twist constraints, Brunel University Mathematics Technical Papers collection; (1974).

[19] D. J. Walton, D. S. Meek, A triangular g 1 patch from boundary curves, Computer-Aided Design 28 (1996) 113–123.

[20] W.-h. Tong, T.-w. Kim, High-order approximation of implicit surfaces by g1 triangular spline surfaces, Computer-Aided Design 41 (2009) 441–455.

[21] G. Farin, D. Hansford, Agnostic g 1 gregory surfaces, Graphical Models 74 (2012) 346–350.

[22] S. J. Owen, D. R. White, T. J. Tautges, Facet-based surfaces for 3d mesh generation., in: Proceedings of the 11th International Meshing Roundtable, 2002, pp. 297–311.

[23] P. J. Frey, About surface remeshing (2000).

[24] D. A. Ibanez, E. S. Seol, C. W. Smith, M. S. Shephard, Pumi: Parallel unstructured mesh infrastructure, ACM Transactions on Mathematical Software, submitted (2014).

[25] C. Cantwell, D. Moxey, A. Comerford, A. Bolis, G. Rocco, G. Mengaldo, D. De Grazia, S. Yakovlev, J.-E. Lombard, D. Ekelschot, et al., Nektar++: An open-source spectral/hp element framework, Computer Physics Communications 192 (2015) 205–219.

[26] G. Farin, Curves and surfaces for computer-aided geometric design: a practical guide, Elsevier, 2014.

[27] G. Farin, Triangular bernstein-bézier patches, Computer Aided Geometric Design 3 (1986) 83–127.

[28] Q. Chen, I. Babuška, Approximate optimal points for polynomial interpolation of real functions in an interval and in a triangle, Computer Methods in Applied Mechanics and Engineering 128 (1995) 405–417.

[29] D. Moxey, D. Ekelschot, Ü. Keskin, S. Sherwin, J. Peiró, A thermo-elastic analogy for high-order curvilinear meshing with control of mesh validity and quality, Proceedings of the 23rd International Meshing Roundtable 82 (2014) 127–135.

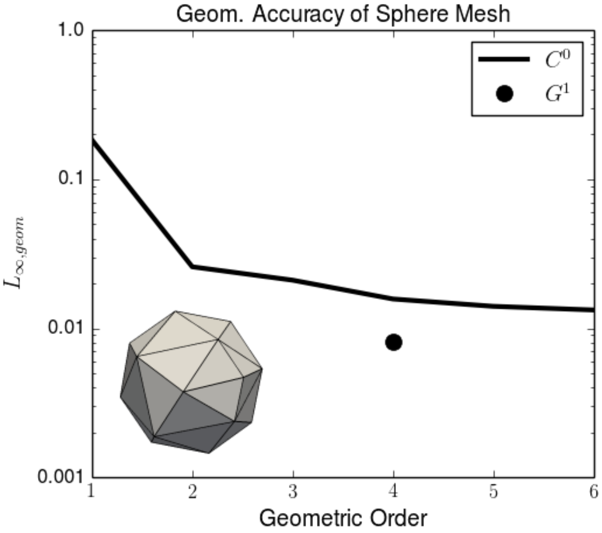

Fig. 1. Initial linear mesh (inset) and interpolation error for curved meshes.

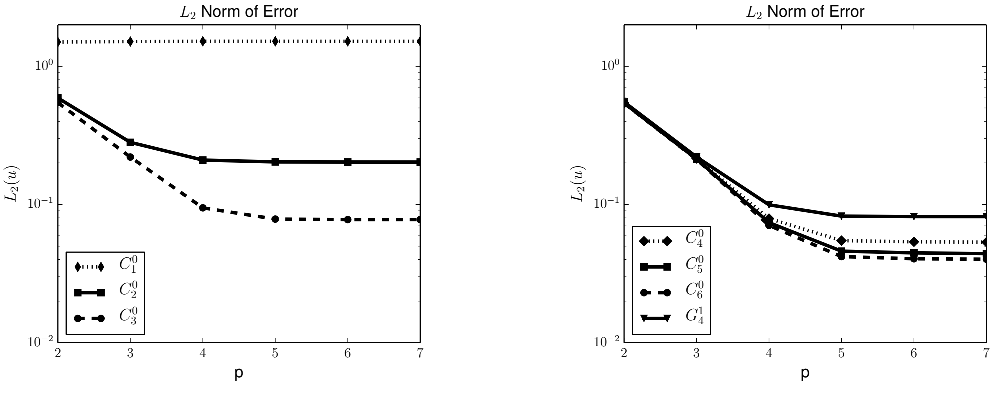

Fig. 2. $L_2$ norm of error for Poisson's Equation.

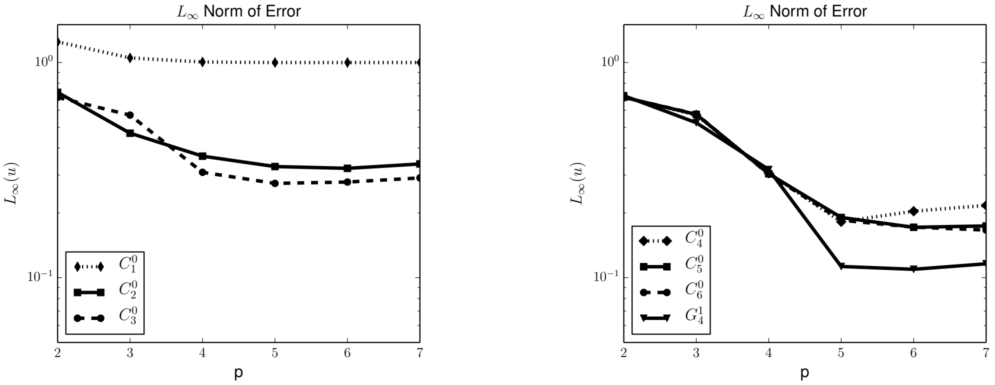

Fig. 3. $L_\infty$ norm of error for Poisson's Equation.

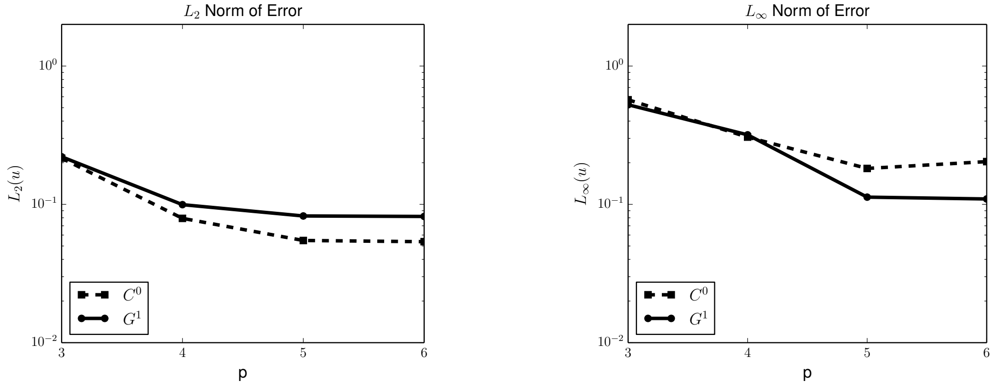

Fig. 4. Error norms for Poisson's Equation for both quartic curved meshes.

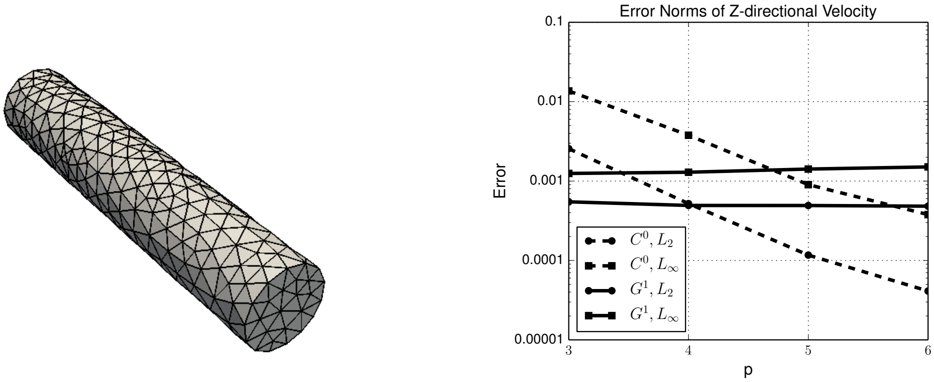

Fig. 5. Linear mesh used for Hagen-Poisseuille Flow.