Hermite mean value interpolation on polygons

Rick Beatson[*] Michael S. Floater[†] Carl Emil Kåshagen[‡]

January 29, 2018

Abstract

Hermite mean value interpolation is a method for interpolating function values and derivatives on the boundary of a domain, using boundary integrals. In this paper we specialize the interpolation to polygonal domains and show that if the boundary data is piecewise quadratic, the integrals can be found explicitly and evaluated easily.

Keywords: Transfinite interpolation, Hermite interpolation, mean value coordinates.

1 Introduction

In [3], a method was derived for constructing, in a domain $\Omega \subset \mathbb{R}^2$, a function $g : \Omega \to \mathbb{R}$ that matches the values and normal derivatives of some function f on the boundary $\partial\Omega$. The method comes from minimizing a kind of energy functional, similar to thin-plate energy, but which allows an explicit solution in terms of boundary integrals. Since the method is a natural extension of mean value interpolation to function values, we might call it Hermite mean value interpolation.

In analogy to the fact that mean value interpolation has linear precision, Hermite mean value interpolation has cubic precision, due to the energy

[*] Department of Mathematics and Statistics, University of Canterbury, Christchurch, New Zealand, *email: rick.beatson@canterbury.ac.nz*

[†] Department of Mathematics, University of Oslo, Moltke Moes vei 35, 0851 Oslo, Norway, *email: michaelf@math.uio.no*

[‡] Same address as second author, *email: carlek@student.matnat.uio.no*

functional being minimized. For this reason, it is not surprising that the method appears to perform very well in practice, generating visually smooth and well-behaved interpolants, even when the domain is irregularly shaped. In the current paper we specialize the method to polygonal domains and show that if the boundary data is piecewise quadratic, the integrals can be found explicitly and evaluated easily. Cubic precision for the continuous version of the method was proved in [3]. This implies quadratic precision for the discrete version with piecewise quadratic values, and piecewise linear first derivatives on the boundary.

In [7], the same method was derived, independently, based on a mean value property of biharmonic functions. The authors also specialized the method to polygons. They argued that the boundary integrals have a closed form if one makes g cubic along each edge, and the normal derivative of g linear. Due to the choice of normal derivative, the resulting method loses cubic precision but retains quadratic precision. They used the method for image deformation by applying it to vector-valued data. The result is a generalization of mean value coordinates, with more deformation parameters to tweak. Some nice examples of this were shown in [7].

On the other hand, the 'closed formulas' for the integrals in [7] are by no means simple to derive, nor to implement, and involve inverse trigonometric, log, and complex-valued functions. The goal of this paper is to propose a considerable simplification of the integrals by taking g to be piecewise quadratic along each edge, and the normal derivative of g piecewise linear. The resulting method still has quadratic precision, similar to [7], but the integrals can be computed using only square roots in addition to the usual arithmetic operations. We also focus on the idea of using the interpolation to generate n-sided patches for computer aided design, and compare it with the 'ribbon'-interpolants developed by Varady, Rockwood, and Salvi [10].

2 Hermite interpolation on convex domains

We start by recalling the construction of [3], specialized to planar domains, and for value and first derivative data only. We will consider both convex and non-convex domains, but it is simpler to explain the mathematical definitions for convex domains. The extension to non-convex ones is very simple, and parallels the generalization of mean value interpolation from convex domains to non-convex domains as in [4] and [1]. So, let $\Omega \subset \mathbb{R}^2$ be a bounded, open,

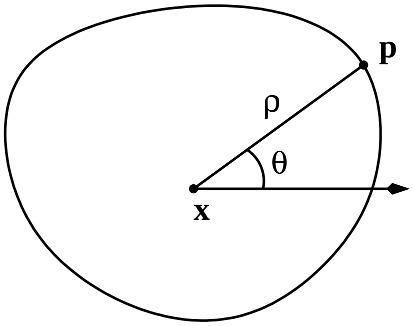

Figure 1: Definition of $\mathbf{p}$ and ρ.

convex domain. Given the values and first order partial derivatives of a function f on the boundary $\partial\Omega$, the approach to constructing an interpolant $g : \Omega \to \mathbb{R}$ in [3] was as follows. First, for all boundary points $\mathbf{p} \in \partial\Omega$, we set $g(\mathbf{p}) = f(\mathbf{p})$ and $D_{\mathbf{n}}g(\mathbf{p}) = D_{\mathbf{n}}f(\mathbf{p})$, where $D_{\mathbf{n}}$ denotes the normal derivative in the outward unit normal direction $\mathbf{n}$. Then, we determine $g(\mathbf{x})$ for each point $\mathbf{x} = (x_1, x_2)$ in Ω. With $\mathbf{x} \in \Omega$ fixed, let τ be a linear polynomial in $\mathbb{R}^2$, which we can represent as

$$\tau(\mathbf{y}) = a + \mathbf{b} \cdot (\mathbf{y} - \mathbf{x}), \qquad \mathbf{y} \in \mathbb{R}^2,$$

for some scalar $a \in \mathbb{R}$ and vector $\mathbf{b} \in \mathbb{R}^2$. Then

$$a = \tau(\mathbf{x}) \qquad \text{and} \qquad \mathbf{b} = \nabla\tau(\mathbf{x}). \tag{1}$$

For each unit vector $\mathbf{v} = (\cos\theta, \sin\theta)$ in $\mathbb{R}^2$, $0 \le \theta < 2\pi$, the ray $\{\mathbf{x} + r\mathbf{v} : r \ge 0\}$ intersects the boundary $\partial\Omega$ at the unique point (because Ω is convex)

$$\mathbf{p} = \mathbf{x} + \rho\mathbf{v}, \tag{2}$$

where $\rho = |\mathbf{p} - \mathbf{x}|$, the Euclidean distance between $\mathbf{p}$ and $\mathbf{x}$, see Figure 1. Let $q_{\mathbf{v}}$ be the Hermite cubic polynomial such that

$$q_{\mathbf{v}}^{(i)}(0) = D_{\mathbf{v}}^i \tau(\mathbf{x}), \qquad q_{\mathbf{v}}^{(i)}(\rho) = D_{\mathbf{v}}^i g(\mathbf{p}), \qquad i = 0, 1, \tag{3}$$

where $D_{\mathbf{v}}g$ denotes the directional derivative of g in the direction $\mathbf{v}$. Then we find τ_* that minimizes

$$E(\tau) := \int_0^{2\pi} \int_0^{\rho} \left(q_{\mathbf{v}}''(r)\right)^2 dr\, d\theta,$$

and set

$$g(\mathbf{x}) = \tau_*(\mathbf{x}) = a_*. \tag{4}$$

It was shown in [3] that τ_* is unique and numerical examples in [3] showed convincingly that g is C^1, although a proof was not found. However, it was proven that if f is a cubic polynomial then $g = f$, i.e., that the method reproduces polynomials of degree ≤ 3, and in this case g is clearly C^1.

How do we find τ_* and subsequently a_*? One approach is as follows. One can first show [2] that $E(\tau)$ is minimized if and only if

$$\int_0^{2\pi} q_{\mathbf{v}}'''(0)\, d\theta = 0 \quad \text{and} \quad \int_0^{2\pi} q_{\mathbf{v}}''(0)\mathbf{v}\, d\theta = \mathbf{0}. \tag{5}$$

Then it remains to solve these two equations in the unknowns a and $\mathbf{b}$ that define τ. Since $q_{\mathbf{v}}$ is a cubic polynomial, some simple calculations show that

$$\begin{aligned} q_{\mathbf{v}}'''(0) &= \frac{6}{\rho^2}\left(q_{\mathbf{v}}'(0) + q_{\mathbf{v}}'(\rho) - 2\left(\frac{q_{\mathbf{v}}(\rho) - q_{\mathbf{v}}(0)}{\rho}\right)\right), \\ q_{\mathbf{v}}''(0) &= \frac{2}{\rho}\left(3\left(\frac{q_{\mathbf{v}}(\rho) - q_{\mathbf{v}}(0)}{\rho}\right) - 2q_{\mathbf{v}}'(0) - q_{\mathbf{v}}'(\rho)\right). \end{aligned}$$

Applying these to (5) and using the definitions in (1) and (3) we obtain the two equations

$$\begin{aligned} \int_0^{2\pi} \frac{1}{\rho^2}\left(\mathbf{v}\cdot\mathbf{b} + D_{\mathbf{v}}g(\mathbf{p}) - 2\left(\frac{g(\mathbf{p}) - a}{\rho}\right)\right) d\theta = 0, \\ \int_0^{2\pi} \frac{1}{\rho}\left(3\left(\frac{g(\mathbf{p}) - a}{\rho}\right) - 2\mathbf{v}\cdot\mathbf{b} - D_{\mathbf{v}}g(\mathbf{p})\right)\mathbf{v}\, d\theta = \mathbf{0}. \end{aligned}$$

This is a linear system in the two unknowns a and $\mathbf{b}$. Defining $v_0 = \rho^{-1}$, so that

$$(v_0, v_1, v_2) = (\rho^{-1}, \cos\theta, \sin\theta),$$

we can write this system as

$$M\begin{bmatrix} a \\ \mathbf{b} \end{bmatrix} = \mathbf{c},$$

where

$$M = \begin{bmatrix} 6I_{00} & 3I_{01} & 3I_{02} \\ 3I_{10} & 2I_{11} & 2I_{12} \\ 3I_{20} & 2I_{21} & 2I_{22} \end{bmatrix}, \qquad \mathbf{c} = \begin{bmatrix} 6J_0 - 3K_0 \\ 3J_1 - K_1 \\ 3J_2 - K_2 \end{bmatrix}, \tag{6}$$

$$I_{jk} = \int_0^{2\pi} v_0 v_j v_k \, d\theta, \qquad j,k = 0,1,2,$$

and

$$J_j = \int_0^{2\pi} v_0^2 g(\mathbf{p}) v_j \, d\theta, \qquad K_j = \int_0^{2\pi} v_0 D_{\mathbf{v}} g(\mathbf{p}) v_j \, d\theta, \qquad j = 0,1,2.$$

The matrix M is symmetric and positive definite and the system is easy to solve,

$$\begin{bmatrix} a \\ \mathbf{b} \end{bmatrix} = M^{-1}\mathbf{c}, \tag{7}$$

and we now set $g(\mathbf{x}) = a = a_*$. We do not need the vector $\mathbf{b}$ to define $g(\mathbf{x})$, but $\mathbf{b}$ might be used as an approximation to $\nabla g(\mathbf{x})$ (see Section 5).

3 Polygons

Suppose now that the domain Ω is a convex polygon in $\mathbb{R}^2$ with $n \geq 3$ vertices $\mathbf{p}_1, \ldots, \mathbf{p}_n$ ordered in anti-clockwise sequence. When necessary we use cyclic notation, $\mathbf{p}_{n+1} := \mathbf{p}_1$, and so on. We will now construct g using only the data

$$f(\mathbf{p}_i),\ \nabla f(\mathbf{p}_i),\ D_{\mathbf{n}_i} f(\mathbf{q}_i), \qquad i = 1, \ldots, n, \tag{8}$$

where $\mathbf{q}_i$ is the midpoint of the edge $[\mathbf{p}_i, \mathbf{p}_{i+1}]$ and $\mathbf{n}_i$ is the outward unit normal on that edge; see Figure 2. If the polygon is a triangle, this is the same data used in the Clough-Tocher elements and Powell-Sabin 12-split

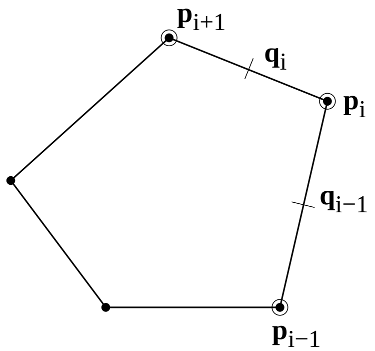

Figure 2: Data for the interpolant in a polygon.

elements; see [5], [8]. In Clough-Tocher elements, on each edge, g is a cubic polynomial and the normal derivative $D_{\mathbf{n}}g$ is quadratic. In Powell-Sabin 12-split elements, on each edge, g is a C^1 piecewise quadratic with the breakpoint at the midpoint of the edge, while $D_{\mathbf{n}}g$ is C^0 piecewise linear.

We will use the boundary data used in Powell-Sabin 12-split elements, extended to the polygon, because, as we will see, this reduces all the integrals defining M and $\mathbf{c}$ in (6) to simple closed formulas. Cubic precision is now lost, but we retain quadratic precision: if f is quadratic, then $g = f$.

Thus we start by representing g and $D_{\mathbf{n}_i}g$ on the edge $[\mathbf{p}_i, \mathbf{p}_{i+1}]$ in piecewise quadratic and piecewise linear Bernstein form, respectively (see for example [9] or [6]). Consider the edge points

$$\mathbf{p} = (1-t)\mathbf{p}_i + t\mathbf{q}_i, \quad \text{and} \quad \mathbf{q} = (1-u)\mathbf{q}_i + u\mathbf{p}_{i+1}, \qquad t, u \in [0,1].$$

Then

$$\begin{aligned} g(\mathbf{p}) &= (1-t)^2 c_0 + 2t(1-t)c_1 + t^2 c_2, \\ g(\mathbf{q}) &= (1-u)^2 c_2 + 2u(1-u)c_3 + u^2 c_4, \end{aligned}$$

where

$$\begin{aligned} c_0 &= f(\mathbf{p}_i), \qquad c_1 = f(\mathbf{p}_i) + (\mathbf{p}_{i+1} - \mathbf{p}_i) \cdot \nabla f(\mathbf{p}_i)/4, \\ c_4 &= f(\mathbf{p}_{i+1}), \qquad c_3 = f(\mathbf{p}_{i+1}) - (\mathbf{p}_{i+1} - \mathbf{p}_i) \cdot \nabla f(\mathbf{p}_{i+1})/4, \\ & \qquad c_2 = (c_1 + c_3)/2, \end{aligned}$$

and

$$\begin{aligned} D_{\mathbf{n}_i} g(\mathbf{p}) &= (1-t)d_0 + td_1, \\ D_{\mathbf{n}_i} g(\mathbf{q}) &= (1-u)d_1 + ud_2, \end{aligned}$$

where

$$d_0 = \mathbf{n}_i \cdot \nabla f(\mathbf{p}_i), \qquad d_1 = D_{\mathbf{n}_i} f(\mathbf{q}_i), \qquad d_2 = \mathbf{n}_i \cdot \nabla f(\mathbf{p}_{i+1}). \tag{9}$$

4 Computing the integrals

We will show that the three kinds of integral defining the matrix M and the right-hand side $\mathbf{c}$ in (6) can by found by simple formulas. Keeping $\mathbf{x} \in \Omega$

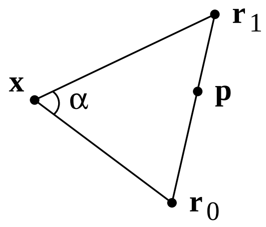

Figure 3: Notation for one line segment.

fixed, we first of all split the integrals into pieces, with each piece corresponding to a half-edge of the polygon, either of the form $[\mathbf{p}_i, \mathbf{q}_i]$ or $[\mathbf{q}_i, \mathbf{p}_{i+1}]$. For simplicity let us call this generic half-edge $[\mathbf{r}_0, \mathbf{r}_1]$ see, Figure 3. On this half-edge, g is a quadratic polynomial and the outward unit normal derivative $D_{\mathbf{n}}g$ is linear. Thus, there are coefficients c_0, c_1, c_2, and d_0, d_1 such that, at the point

$$\mathbf{p} = (1-t)\mathbf{r}_0 + t\mathbf{r}_1, \qquad t \in [0,1], \tag{10}$$

we have

$$g(\mathbf{p}) = (1-t)^2 c_0 + 2t(1-t)c_1 + t^2 c_2, \tag{11}$$

and

$$D_{\mathbf{n}}g(\mathbf{p}) = (1-t)d_0 + td_1, \tag{12}$$

Defining $\rho_i = |\mathbf{r}_i - \mathbf{x}|$, $i = 0, 1$, there are angles θ_i, $i = 0, 1$, such that

$$\mathbf{r}_i = \mathbf{x} + \rho_i \mathbf{v}_i,$$

where $\mathbf{v}_i = (\cos\theta_i, \sin\theta_i)$. We set

$$(v_{i,0}, v_{i,1}, v_{i,2}) = (\rho_i^{-1}, \cos\theta_i, \sin\theta_i), \qquad i = 0, 1.$$

4.1 Matrix elements

Let us start with the assembly of the elements of M since they depend only on the geometry of Ω, not on the function f. For a fixed half-edge $[\mathbf{r}_0, \mathbf{r}_1]$, let $\alpha = \theta_1 - \theta_0$ and

$$L_0 = L_3 = \frac{(1-\cos\alpha)^2(2+\cos\alpha)}{3\sin^3\alpha}, \qquad L_1 = L_2 = \frac{(1-\cos\alpha)^2}{3\sin^3\alpha}. \tag{13}$$

We can simplify notation using the sets of multi-indices

$$\mathcal{P}_{r,n} = \{(i_1, \ldots, i_n) : i_1, \ldots, i_n \in \{0,1\},\, i_1 + \cdots + i_n = r\},$$

for $n \geq 1$ and $0 \leq r \leq n$. In particular,

$$\begin{aligned}
&\mathcal{P}_{0,3} = \{(0,0,0)\}, \quad \mathcal{P}_{1,3} = \{(1,0,0),(0,1,0),(0,0,1)\}, \\
&\mathcal{P}_{2,3} = \{(1,1,0),(1,0,1),(0,1,1)\}, \quad \mathcal{P}_{3,3} = \{(1,1,1)\}.
\end{aligned}$$

Theorem 1

$$\int_{\theta_0}^{\theta_1} v_0 v_j v_k \, d\theta = \sum_{r=0}^{3} L_r \sum_{(i_1,i_2,i_3)\in\mathcal{P}_{r,3}} v_{i_1,0} v_{i_2,j} v_{i_3,k}, \quad j,k = 0,1,2.$$

Proof. From the representation of $\mathbf{p}$ in (2) and using trigonometry, similar to [3], we find

$$\rho^{-1} = \lambda \rho_0^{-1} + \mu \rho_1^{-1} \qquad \text{and} \qquad \mathbf{v} = \lambda \mathbf{v}_0 + \mu \mathbf{v}_1, \tag{14}$$

where

$$\lambda = \frac{\sin(\theta_1 - \theta)}{\sin\alpha}, \qquad \mu = \frac{\sin(\theta - \theta_0)}{\sin\alpha},$$

and therefore,

$$v_j = \lambda v_{0,j} + \mu v_{1,j}, \qquad j = 0,1,2. \tag{15}$$

Then we expand

$$v_0 v_j v_k = \sum_{r=0}^{3} \mu^r \lambda^{3-r} \sum_{(i_1,i_2,i_3)\in\mathcal{P}_{r,3}} v_{i_1,0} v_{i_2,j} v_{i_3,k}, \qquad j,k = 0,1,2,$$

and integrating this over $\theta \in [\theta_0, \theta_1]$ proves the theorem, with

$$L_r := \int_{\theta_0}^{\theta_1} \mu^r \lambda^{3-r} \, d\theta, \qquad r = 0,1,2,3.$$

It remains to show that the integrals L_r are as in (13). With the substitution $\beta = \theta - \theta_0$, we find

$$L_r = \frac{1}{\sin^3\alpha} \widehat{L}_r,$$

where

$$\widehat{L}_r = \int_0^\alpha \sin^r\beta \sin^{3-r}(\alpha - \beta) \, d\beta.$$

Since by symmetry, $\widehat{L}_0 = \widehat{L}_3$ and $\widehat{L}_1 = \widehat{L}_2$, it is enough to find $\widehat{L}_2$ and $\widehat{L}_3$. Using the fact that

$$\sin^3\beta = \sin\beta - \sin\beta\cos^2\beta,$$

we find

$$\widehat{L}_3 = \frac{1}{3}(2 - 3\cos\alpha + \cos^3\alpha) = \frac{1}{3}(1-\cos\alpha)^2(2+\cos\alpha).$$

To find $\widehat{L}_2$, we use the fact that

$$\sin(\alpha-\beta)\sin^2\beta = \sin\alpha\cos\beta\sin^2\beta - \cos\alpha\sin^3\beta,$$

which leads to

$$\widehat{L}_2 = \frac{1}{3}\sin\alpha(\sin^3\alpha) - \cos\alpha\widehat{L}_3,$$

which, after expressing $\sin^2\alpha$ as $1-\cos^2\alpha$, simplifies to

$$\widehat{L}_2 = \frac{1}{3}(1-\cos\alpha)^2.$$

This shows that the L_r are as in (13). □

4.2 Right-hand side

Next, consider the assembly of the elements of the right-hand side $\mathbf{c}$ in (6). These depend on the values and first derivatives of g on $\partial\Omega$. Firstly, consider the integrals J_j.

Theorem 2 *With g as in (11),*

$$\int_{\theta_0}^{\theta_1} v_0^2 g(\mathbf{p}) v_j = \sum_{r=0}^{3} L_r \sum_{(i_1,i_2,i_3)\in\mathcal{P}_{r,3}} c_{i_1+i_2} v_{i_1,0} v_{i_2,0} v_{i_3,j}, \qquad j = 0,1,2.$$

Proof. From the two representations of $\mathbf{p}$ in (2) and (10), we find

$$1 - t = \rho\lambda\rho_0^{-1}, \qquad t = \rho\mu\rho_1^{-1}, \tag{16}$$

and substituting these into (11) and multiplying by $v_0^2 = \rho^{-2}$ gives

$$v_0^2 g(\mathbf{p}) = \sum_{r=0}^{2} \mu^r \lambda^{2-r} c_r \sum_{(i_1,i_2)\in\mathcal{P}_{r,2}} v_{i_1,0} v_{i_2,0}.$$

Multiplying this by v_j, $j = 0, 1, 2$, and using the expansion (15) gives

$$v_0^2 g(\mathbf{p}) v_j = \sum_{r=0}^{3} \mu^r \lambda^{3-r} \sum_{(i_1,i_2,i_3)\in\mathcal{P}_{r,3}} c_{i_1+i_2} v_{i_1,0} v_{i_2,0} v_{i_3,j}, \qquad j = 0, 1, 2.$$

Integrating this over $\theta \in [\theta_0, \theta_1]$, gives the result. □

Secondly, consider the integrals K_j. The piece of K_j corresponding to the half-edge $[\mathbf{r}_0, \mathbf{r}_1]$ involves both g and the directional derivative $D_{\mathbf{v}} g$. Differentiating g in (11) along this half-edge gives, for the point $\mathbf{p}$ in (10),

$$D_{\mathbf{e}} g(\mathbf{p}) = (1-t)\widehat{c}_0 + t\widehat{c}_1, \tag{17}$$

where $\mathbf{e}$ is the unit vector

$$\mathbf{e} = (\mathbf{r}_1 - \mathbf{r}_0)/|\mathbf{r}_1 - \mathbf{r}_0|,$$

and

$$\widehat{c}_0 = 2(c_1 - c_0)/|\mathbf{r}_1 - \mathbf{r}_0|, \qquad \widehat{c}_1 = 2(c_2 - c_1)/|\mathbf{r}_1 - \mathbf{r}_0|.$$

Theorem 3 *With g, $D_{\mathbf{n}} g$, and $D_{\mathbf{e}} g$ given by (11), (12), and (17) respectively,*

$$\int_{\theta_0}^{\theta_1} v_0 D_{\mathbf{v}} g(\mathbf{p}) v_j = \sum_{r=0}^{3} L_r \sum_{(i_1,i_2,i_3)\in\mathcal{P}_{r,3}} (\mathbf{v}_{i_1} \cdot \widehat{\mathbf{d}}_{i_2}) v_{i_2,0} v_{i_3,j}, \quad j = 0, 1, 2,$$

where

$$\widehat{\mathbf{d}}_i = d_i \mathbf{n} + \widehat{c}_i \mathbf{e}, \qquad i = 0, 1. \tag{18}$$

Proof. The directional derivative of g at $\mathbf{p}$ in the direction $\mathbf{v}$ is

$$D_{\mathbf{v}} g(\mathbf{p}) = (\mathbf{v} \cdot \mathbf{n}) D_{\mathbf{n}} g(\mathbf{p}) + (\mathbf{v} \cdot \mathbf{e}) D_{\mathbf{e}} f(\mathbf{p}).$$

Substituting (16) into (12) and (17) gives, respectively,

$$\begin{aligned} D_{\mathbf{n}} g(\mathbf{p}) &= \rho(d_0 \rho_0^{-1} \lambda + d_1 \rho_1^{-1} \mu), \\ D_{\mathbf{e}} g(\mathbf{p}) &= \rho(\widehat{c}_0 \rho_0^{-1} \lambda + \widehat{c}_1 \rho_1^{-1} \mu), \end{aligned}$$

and expanding $\mathbf{v}$ from (14), gives

$$\begin{aligned}\mathbf{v}\cdot\mathbf{n} &= \lambda(\mathbf{v}_0\cdot\mathbf{n}) + \mu(\mathbf{v}_1\cdot\mathbf{n}),\\ \mathbf{v}\cdot\mathbf{e} &= \lambda(\mathbf{v}_0\cdot\mathbf{e}) + \mu(\mathbf{v}_1\cdot\mathbf{e}).\end{aligned}$$

Using these substitutions and multiplying $D_{\mathbf{v}}g(\mathbf{p})$ by $v_0 = \rho^{-1}$ gives

$$v_0 D_{\mathbf{v}}g(\mathbf{p}) = \sum_{r=0}^{2} \mu^r \lambda^{2-r} \sum_{(i_1,i_2)\in\mathcal{P}_{r,2}} (\mathbf{v}_{i_1}\cdot\widehat{\mathbf{d}}_{i_2}) v_{i_2,0},$$

with $\widehat{\mathbf{d}}_i$, $i = 0, 1$, as in (18). We now multiply this by v_j, $j = 0, 1, 2$, and the remainder of the proof is like that of Theorem 2. □

5 Numerical examples

Recall that the cross, or wedge, product of two vectors $\mathbf{a} = (a_1, a_2)$ and $\mathbf{b} = (b_1, b_2)$ is defined as $\mathbf{a}\times\mathbf{b} = a_1b_2 - b_1a_2 = |\mathbf{a}||\mathbf{b}|\sin\alpha$, where α is the angle counter clockwise between $\mathbf{a}$ and $\mathbf{b}$. Therefore, in implementations it is not necessary to compute the angle α in (13). Instead the required quantities $\cos(\alpha)$ and $\sin(\alpha)$ can be found from the scalar and cross products,

$$\cos\alpha = \mathbf{v}_0\cdot\mathbf{v}_1, \qquad \sin\alpha = \mathbf{v}_0\times\mathbf{v}_1. \tag{19}$$

We also note that if the normal derivative $d_1 = D_{\mathbf{n}_i}f(\mathbf{q}_i)$ in (9) is not available, a natural alternative is to use the average of adjacent normals,

$$d_1 = (\mathbf{n}_i\cdot\nabla f(\mathbf{p}_i) + \mathbf{n}_i\cdot\nabla f(\mathbf{p}_{i+1}))/2, \tag{20}$$

as proposed in [7]. Taking this average will retain quadratic precision.

Figure 4 shows two of the nodal functions for the interpolant on an irregular quadrilateral with vertices $(0,0)$, $(1.5,0)$, $(1.5,1)$ and $(0.5,1.5)$. The nodal elements were constructed by setting f to have value 1 at one vertex and setting all other data to 0. Figure 5 shows two derivative nodal functions on the same quadrilateral, scaled by a factor of 2 for better visualization, by setting a partial derivative of f to equal 2 at one vertex and setting all other data to 0.

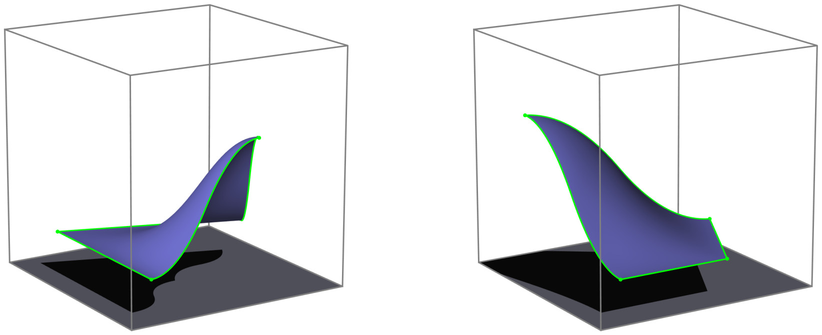

Figure 4: Value nodal elements on a quadrilateral

5.1 Experiments with convex and non convex tilings

In this subsection we apply the interpolant piecewise to construct a C^1 approximation $s(\mathbf{x})$ to the function

$$f(\mathbf{x}) = \cos(x) + 2\sin(y) + 3\cos(x + xy) + 4\sin(y + xy), \quad \mathbf{x} = (x, y),$$

on the domain $\mathcal{T} = [-1, 1] \times [-1, 1]$. As mentioned in the introduction it follows from the cubic precision of the continuous version [3] that the current method will have quadratic precision.

We initially tiled $\mathcal{T}$ into subsquares P of side length h. The tiling with $h = 1/2$ is shown on the left of Figure 6.

In our first experiment, we compute a single Hermite mean value approximation g_P to f in each subsquare P, to construct s piecewise. The approximation errors to function values f and gradients ∇f in each subsquare P are estimated by the values $a_{\mathbf{x},P} = a$ and $\mathbf{b}_{\mathbf{x},P} = \mathbf{b}$ in (7) at points $\mathbf{x}$ of a yet finer grid, consisting of the vertices, $\mathcal{X}_{h/8}$, of a grid of squares of side length $h/8$. The errors in function values and gradients are taken as

$$\max_{\mathbf{x}\in\mathcal{X}_{h/8}} |f(\mathbf{x}) - a_{\mathbf{x},P}| \quad \text{and} \quad \max_{\mathbf{x}\in\mathcal{X}_{h/8}} |\nabla f(\mathbf{x}) - \mathbf{b}_{\mathbf{x},P}|.$$

The results obtained by applying this procedure for various values of h are given in Table 1. The results are consistent with the piecewise interpolant

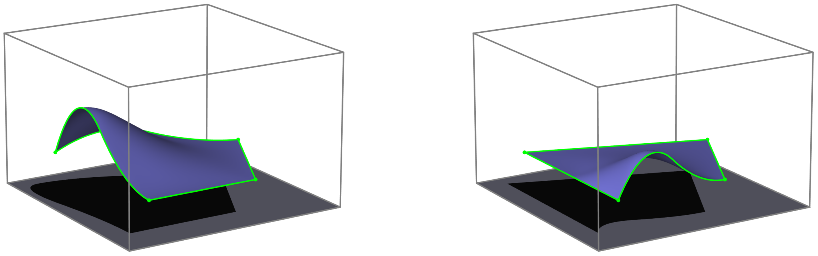

Figure 5: Derivative nodal elements on a quadrilateral.

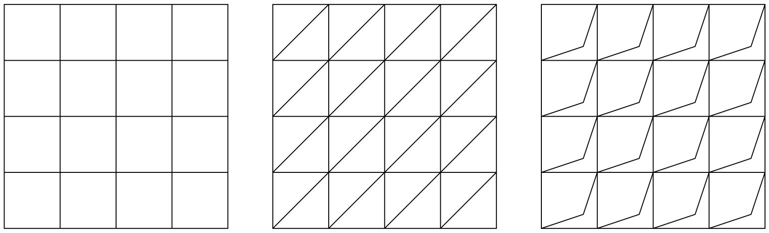

Figure 6: Three tilings of the unit square $[-1, 1]$ with $h = 1/2$.

showing $\mathcal{O}(h^3)$ convergence to function values and $\mathcal{O}(h^2)$ convergence to gradient values, when the function f being approximated is sufficiently smooth, and the polygonal region has diameter h.

In our second experiment, we further divide each subsquare into two triangles, as in the middle of Figure 6. The piecewise approximation now consists of Hermite mean value interpolants over each triangle. The error in the piecewise approximation is estimated using the finer square grid as before. The results are shown in Table 2.

Our third experiment tests the interpolation method when some of the polygons are nonconvex. In the polygonal case, the algorithm for g in a nonconvex polygon is the same as that for a convex polygon, except that the angle α must be taken to be a signed angle. This is simple to implement, since we do not use α explicitly, and $\sin\alpha$ will have the same sign as α if we simply use the same formulas (19) for $\cos\alpha$ and $\sin\alpha$ as previously.

h	function error	ratio to previous	gradient error	ratio to previous
1/2	1.4021e-2		2.8714e-1	
1/4	1.5253e-3	9.192	7.9161e-2	3.627
1/8	1.8675e-4	8.168	2.0348e-2	3.890
1/16	2.3399e-5	7.981	5.1181e-3	3.976
1/32	2.9340e-6	7.975	1.2810e-3	3.996
1/64	3.6752e-7	7.983	3.2026e-4	4.000
1/128	4.5994e-8	7.991	8.0072e-5	4.000

Table 1: Errors on squares

h	function error	ratio to previous	gradient error	ratio to previous
1/2	4.8435e-2		5.3881e-1	
1/4	4.8923e-3	9.900	1.5881e-1	3.393
1/8	5.7027e-4	8.579	4.1957e-2	3.785
1/16	6.9184e-5	8.243	1.0717e-2	3.195
1/32	8.5571e-6	8.085	2.7088e-3	3.956
1/64	1.0653e-6	8.033	6.8041e-4	3.981
1/128	1.3293e-7	8.014	1.7049e-4	3.991

Table 2: Errors on triangles

h	function error	ratio to previous	gradient error	ratio to previous
1/2	1.1257e-2		1.3203e-1	
1/4	1.3146e-3	8.563	4.7901e-2	2.756
1/8	1.6120e-4	8.155	1.5026e-2	3.188
1/16	2.0048e-5	8.040	4.1944e-3	3.583
1/32	2.5027e-6	8.011	1.1056e-3	3.794
1/64	3.1690e-7	7.898	2.8380e-4	3.896
1/128	4.0050e-8	7.913	7.1872e-5	3.949

Table 3: Errors on quads

For the non-convex test, we divided each subsquare into two quadrilaterals, one convex, the other non-convex, as shown on the right of Figure 6, and used one Hermite mean value interpolant in each quad. The results are shown in Table 3.

5.2 Parametric patches

Finally, we used the interpolation method to generate a parametric n-sided patch $\mathbf{g} : \Omega \to \mathbb{R}^3$, by matching vector-valued data

$$\mathbf{f}(\mathbf{p}_i),\ \nabla\mathbf{f}(\mathbf{p}_i),\ D_{\mathbf{n}_i}\mathbf{f}(\mathbf{q}_i), \qquad i = 1, \ldots, n, \tag{21}$$

for some vector-valued function $\mathbf{f} : \Omega \to \mathbb{R}^3$.

We made an example, in which there is a five-sided region, a 'hole', within a network of biquadratic Bézier surfaces, as seen in Figure 7. For this example, we chose the parameter domain Ω to be a regular pentagon, with vertices on the unit circle. For the domain edge $[\mathbf{p}_i, \mathbf{p}_{i+1}]$, we took the two tangential directional derivatives

$$D_{\mathbf{d}_i}\mathbf{g}(\mathbf{p}_i), \quad D_{\mathbf{d}_i}\mathbf{g}(\mathbf{p}_{i+1}),$$

where $\mathbf{d}_i = \mathbf{p}_{i+1} - \mathbf{p}_i$, and the three normal derivatives

$$D_{\mathbf{n}_i}\mathbf{g}(\mathbf{p}_i), \quad D_{\mathbf{n}_i}\mathbf{g}(\mathbf{q}_i), \quad D_{\mathbf{n}_i}\mathbf{g}(\mathbf{p}_{i+1})$$

directly from the surface adjacent to that edge.

A reflection line, or zebra stripe, plot of the resulting surface is shown on the left of Figure 7. We compared the result with the ribbon-based interpolation method of Varady, Rickwood, and Salvi [10], which is a generalization of (four-sided) Coons patches. Specifically, we used their 'direct' generalization of Coons patches, the first of the methods they proposed in Section 3.3 of [10]. A reflection line plot of the resulting surface is shown on the right of Figure 7. The plots show that the results of the two methods are very similar for this example. Also, the continuity of the curves at the edges of the zebra stripes visually confirms the G^1 continuity. We found the computation time for the two methods in this example to be similar.

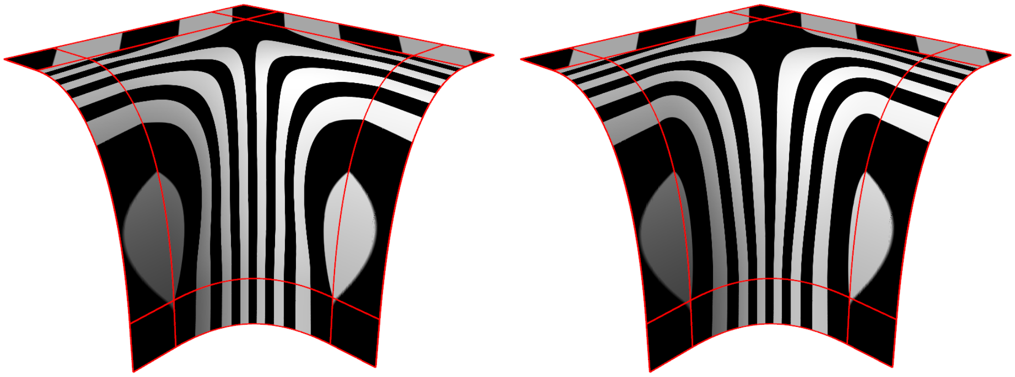

Figure 7: A 5-sided patch matching surrounding patches with G^1 continuity. Left: Hermite MV, Right: 'ribbon' method of Varady et al.

For both methods, the shape of the parameter domain Ω will affect the shape of the patch. Based on experience of curve-fitting, one might, in general, try to choose a parameter domain that better reflects the shape of the boundary of the n-sided region. Various parameterization methods are discussed in [10].

The shape of the Hermite MV patch also depends on the scaling of the vector-valued data in (21). How to make a good choice is a topic for future research.

References

[1] C. Dyken and M. S. Floater, *Transfinite mean value interpolation*, Comput. Aided Geom. Design **26** (2009), 117–134.

[2] M. S. Floater, *Generalized barycentric coordinates and applications*, Acta Numerica **24** (2015), 161–214.

[3] M. S. Floater and C. Schulz, *Pointwise radial minimization: Hermite interpolation on arbitrary domains*, Comput. Graph. Forum **27** (2008), 1505–1512, Proceedings of SGP 2008.

[4] K. Hormann and M. S. Floater, *Mean value coordinates for arbitrary planar polygons*, ACM Trans. on Graph. **25** (2006), 1424–1441.

[5] J. Kosinka and T. J. Cashman, *Watertight conversion of trimmed CAD surfaces to Clough-Tocher splines*, Comput. Aided Geom. Design **37** (2015), 2–41.

[6] M.-J. Lai and L. L. Schumaker, *Spline functions on triangulations*, Cambridge University Press, Cambridge, 2007.

[7] X.-Y. Li, T. Ju, and S.-M. Hu, *Cubic mean value coordinates*, ACM Trans. on Graph. **32** (2013), 126:1–126:10.

[8] M. J. D. Powell and M. A. Sabin, *Piecewise quadratic approximations on triangles*, ACM Trans. Math. Software **3** (1977), 316–325.

[9] H. Prautzsch, W. Boehm, and M. Paluszny, *Bézier and B-spline techniques*, Springer, 2002.

[10] T. Várady, A. Rockwood, and P. Salvi, *Transfinite surface interpolation over irregular n-sided domains*, Computer-Aided Design **43** (2011), 1330–1340.