

HIGHER-ORDER DGFEM TRANSPORT CALCULATIONS ON POLYTOPE
MESHES FOR MASSIVELY-PARALLEL ARCHITECTURES

A Dissertation
by
MICHAEL WAYNE HACKEMACK

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY

Chair of Committee, Jean Ragusa
Committee Members, Marvin Adams
 Jim Morel
 Nancy Amato
 Troy Becker
Head of Department, Yassin Hassan

August 2016

Major Subject: Nuclear Engineering

Copyright 2016 Michael Wayne Hackemack

1. FEM BASIS FUNCTIONS FOR UNSTRUCTURED POLYTOPES

In Section ??, we detailed the spatial discretization of the transport equation. We then proceeded to give the functional forms for the various elementary matrices needed to form the full set of spatially-discretized PDEs. These included the mass, streaming, and surface matrices where the integrations on the element's domain and boundary require combinations of the basis functions' values and gradients. From FEM theory [1], the basis functions act as interpolation functions with local measure on some subset of elements on a discretized mesh, \mathbb{T}_h . To achieve the maximum possible solution convergence rate from Section ??+1, the interpolation functions must have polynomial completeness of at least order p . For 2D interpolants, the basis functions are linearly-complete ($p = 1$) if they can exactly interpolate the $\{1, x, y\}$ span of functions. Likewise, 2D basis functions are said to be quadratically-complete if they can exactly interpolate the $\{1, x, y, x^2, xy, y^2\}$ span of functions. This work seeks to analyze the use of the different linearly-complete polygonal basis functions with the DGFEM S_N transport equation. We then seek to extend this analysis to quadratically-complete basis functions, thus achieving higher convergence rates (third-order). We will do this by utilizing the methodology developed by Rand et al. [2] on the different linearly-complete polygonal coordinates.

The remainder of this chapter is organized as follows. In Section 1.1, we present the 2D, linearly-complete, barycentric, polygonal basis functions that we will analyze in this dissertation. We then present in Section 1.2 the methodology to convert the barycentric polygonal basis functions presented in Section 1.1 into a serendipity space of basis functions with quadratic-completeness. Section 1.3 provides the methodology that will be employed to generate spatial quadrature sets on 2D arbitrary polygons.

Section 1.4 then presents the 3D, linearly-complete, polyhedral basis functions that will be exclusively used in Chapter ?? for 3D DSA calculations. We then present numerical results pertaining to our linear and quadratic 2D basis functions in Section 1.5. Section 1.6 concludes with some closing remarks.

1.1 Linear Basis Functions on 2D Polygons

Figure 1.1, gives an image of a reference polygon along with the geometric notations we will use to define the different linear polygonal coordinates. An element, $K \in \mathbb{R}^2$, is defined by a closed set of N_K points (vertices) in \mathbb{R}^2 . The vertices are ordered $(1, \dots, N_K)$ in a counter-clockwise manner without restriction on their convexity. Face j on the polygon, e_j , is defined as the line segment between vertices j and $j + 1$. The vertex $j + 1$ is determined in general as $j + 1 = \mod(j, N_K) + 1$, which gives a wrap-around definition of vertex $N_K + 1 = 1$.

We complete our geometric description for the polygonal coordinate system by analyzing a point \vec{x} inside the polygon's domain, as also seen in Figure 1.1. α_j is the angle between the points $(\vec{x}_j, \vec{x}, \vec{x}_{j+1})$. Since element K is defined by a closed set of \mathbb{R}^2 points, α_j is strongly bounded: (). We conclude by defining $|\vec{u}|$ as the Euclidean distance of the vector \vec{u} . This means that $|\vec{x} - \vec{x}_j|$ is the distance between the points \vec{x} and \vec{x}_j and $|e_j|$ is the length of face j between points \vec{x}_j and \vec{x}_{j+1} .

In this dissertation, all linearly-complete, 2D basis functions for an element K will obey the properties for barycentric coordinates. If the element K is composed of N_K vertices, then it contains N_K barycentric coordinates, where each one is located at a vertex. These barycentric coordinates will form a *partition of unity*,

$$\sum_{j=1}^{N_K} b_j(\vec{x}) = 1; \quad (1.1)$$

coordinate interpolation will result from an *affine combination* of the vertices,

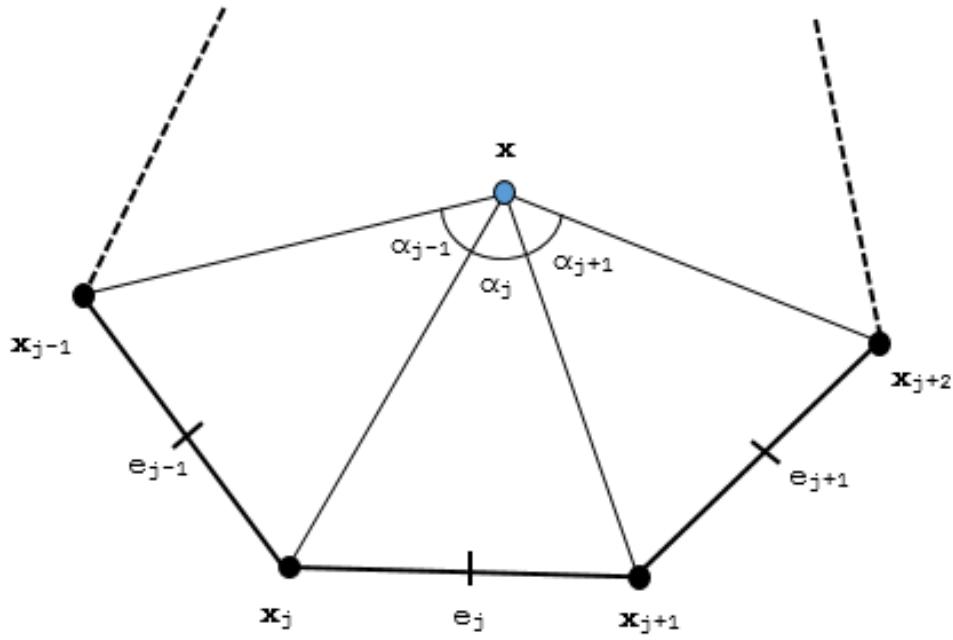


Figure 1.1: Arbitrary polygon with geometric properties used for 2D basis function generation.

$$\sum_{j=1}^{N_K} b_j(\vec{x}) \vec{x}_j = \vec{x}; \quad (1.2)$$

and they will satisfy the *Lagrange property*,

$$b_j(\vec{x}_i) = \delta_{ij}. \quad (1.3)$$

They also have piecewise linearity on faces adjacent to their vertex. As an example of this, consider the coordinates at vertex j ,  along face e_j . Then the piecewise linearity of the coordinate on the face means that it can interpolate as

$$b_j((1 - \mu)\vec{x}_j + \mu\vec{x}_{j+1}) = (1 - \mu)b_j(\vec{x}_j) + \mu b_j(\vec{x}_{j+1}), \quad \mu \in [0, 1]. \quad (1.4)$$

Using the *partition of unity* of Eq. (1.1), we can rewrite Eqs. (1.1-1.2) into a

separate, compact, vectorized form for completeness

$$\sum_{j=1}^{N_K} b_j(\vec{x}) \vec{c}_{j,1}(\vec{x}) = \vec{q}_1, \quad (1.5)$$

where $\vec{c}_{j,1}(\vec{x})$ and \vec{q}_1 are the linearly-complete constraint and equivalence terms, respectively. These terms are simply:

$$\vec{c}_{j,1}(\vec{x}) = \begin{bmatrix} 1 \\ \text{---} \\ y_j - y \end{bmatrix} \quad \text{and} \quad \vec{q}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad (1.6)$$

respectively. Equation (1.5) states that our interpolation functions (the basis functions) can exactly reproduce polynomial functions up to order 1. This is why we state that our basis functions are linearly-complete. However, we will not restrict our N_K basis functions to be polynomials. In fact, of the basis functions that we will use, only the PWL coordinates are formed by combinations of polynomial functions.

1.1.1 Wachspress Rational Basis Functions

The first linearly-complete polygonal coordinates that we will consider are the Wachspress rational functions [3]. These rational functions were the first derived for 2D polygons and possess all the properties of the barycentric coordinates previously detailed. However, they are only valid interpolants over strictly-convex polygons. They have zero measure and blow up for weakly-convex and concave polygons, respectively. Also, their values and gradients cannot be directly evaluated on the polygonal boundary. However, they do have a valid limit which we show in Appendix ???. The Wachspress coordinates (which we denote as b^W) have the following form

$$b_j^W(\vec{x}) = \frac{w_j(\vec{x})}{\sum_i w_i(\vec{x})}, \quad (1.7)$$

where the Wachspress weight function for vertex j , w_j , has the following definition:

$$w_j(\vec{x}) = \frac{A(\vec{x}_{j-1}, \vec{x}_j, \vec{x}_{j+1})}{A(\vec{x}, \vec{x}_{j-1}, \vec{x}_j) A(\vec{x}, \vec{x}_j, \vec{x}_{j+1})}. \quad (1.8)$$

In Eq. (1.8), the terms $A(\vec{a}, \vec{b}, \vec{c})$ denote the signed area of the triangle with vertices \vec{a} , \vec{b} , and \vec{c} . Each of these signed areas can be computed by

$$A(\vec{a}, \vec{b}, \vec{c}) = \frac{1}{2} \begin{vmatrix} 1 & 1 & 1 \\ x_a & x_b & x_c \\ y_a & y_b & y_c \end{vmatrix}. \quad (1.9)$$

There is an alternative method of expressing the Wachspress weight functions. Warren et al. [4] proposed weight functions that are defined in terms of the perpendicular distance of the point \vec{x} to the polygon's faces. Using the reference polygon of Figure 1.1, the perpendicular distance of the point \vec{x} to the face j is denoted as $h_j(\vec{x})$ and is given by

$$h_j(\vec{x}) = (\vec{x}_j - \vec{x}) \cdot \vec{n}_j = (\vec{x}_{j+1} - \vec{x}) \cdot \vec{n}_j, \quad (1.10)$$

where \vec{n}_j is the outward normal direction of face j . Using these perpendicular distance, the Wachspress coordinates can be calculated using Eq. (1.7) with new function definitions of

$$w_j(\vec{x}) = \frac{\vec{n}_{j-1} \times \vec{n}_j}{h_{j-1}(\vec{x}) h_j(\vec{x})}, \quad (1.11)$$

where

$$\vec{x}_1 \times \vec{x}_2 = \begin{vmatrix} x_1 & x_2 \\ y_1 & y_2 \end{vmatrix}. \quad (1.12)$$

For FEM theory, the basis function gradients are also necessary to compute some of the elementary matrices. The gradients of the Wachspress rational functions are straightforward to calculate by simply taking the partial derivatives of Eq. (1.7). Then, using derivative rules along with some algebra, the Wachspress gradients are given by,

$$\vec{\nabla} b_j^W(\vec{x}) = b_j^W(\vec{x}) \left(\vec{R}_j(\vec{x}) - \sum_i b_i^W(\vec{x}) \vec{R}_i(\vec{x}) \right), \quad (1.13)$$

where the reduced gradient, \vec{R}_i , is defined as

$$\vec{R}_i(\vec{x}) = \frac{1}{w_i} \vec{\nabla} w_i. \quad (1.14)$$

This means that the gradients of the Wachspress coordinates can be calculated by combinations of the all the weight functions and their gradients. The weight function gradients are easy to compute using the perpendicular form. The gradient of the j weight functions is given by

$$\vec{\nabla} w_j(\vec{x}) = w_j(\vec{x}) \left(\frac{\vec{n}_{j-1}}{h_{j-1}(\vec{x})} \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \frac{\vec{n}_j}{h_j(\vec{x})} \right). \quad (1.15)$$

This lets us immediately see that \vec{R}_j is simply

$$\begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \quad \vec{R}_j(\vec{x}) = \frac{\vec{n}_{j-1}}{h_{j-1}(\vec{x})} + \frac{\vec{n}_j}{h_j(\vec{x})}. \quad (1.16)$$

We now give a pair of contour plots of the Wachspress coordinates. First, Figure

1.2 provides the contour plots of the four Wachspress functions on the unit square. We see that the functions are smoothly varying within the square with at least C^1 continuity. Then in Figure 1.3, we give the contour plots for a degenerate pentagon which is simply the unit square with a vertex added at point $(1/2, 1)$. We see how the functions fail for this weakly-convex case. The function located at the degenerate vertex has zero measure everywhere within the polygon. Also, the functions located at the vertices adjacent to the degenerate vertex no longer maintain linearity on their adjacent faces. We will not show it here for brevity, but the Wachspress functions on concave polygons will have points in the interior that will result in divide-by-zero operations.

1.1.2 Piecewise Linear (PWL) Basis Functions

The second linearly-complete 2D polygonal coordinates that we will analyze are the Piecewise Linear (PWL) coordinates proposed by Stone and Adams [5, 6]. They originally introduced the PWL coordinates to work specifically for the DGFEM transport equation on unstructured quadrilateral and polygonal grids. These coordinates share some similarities with the Wachspress rational functions, but also contain some key differences. The properties of the PWL coordinates that are different from the Wachspress rational functions can be summarized with the following:

1. PWL works with concave polygons;
2. PWL cannot interpolate on curved surfaces;
3. points on the boundary can be directly evaluated;
4. the PWL integrals can be computed analytically;
5. the PWL functions are only C^0 continuous: their gradients are discontinuous within the element.

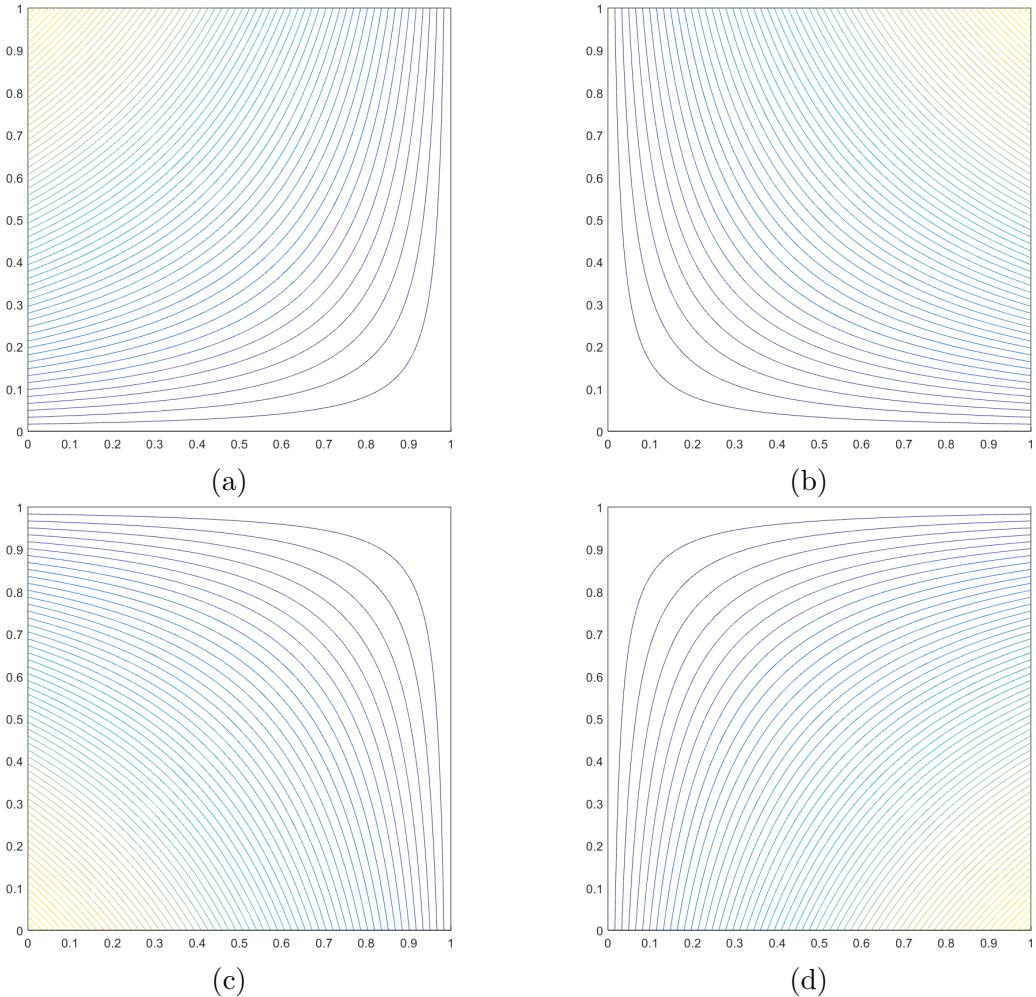


Figure 1.2: Contour plots of the linear Wachspress basis functions on the unit square for the vertices located at: (a) $(0,1)$, (b) $(1,1)$, (c) $(0,0)$, and (d) $(1,0)$.

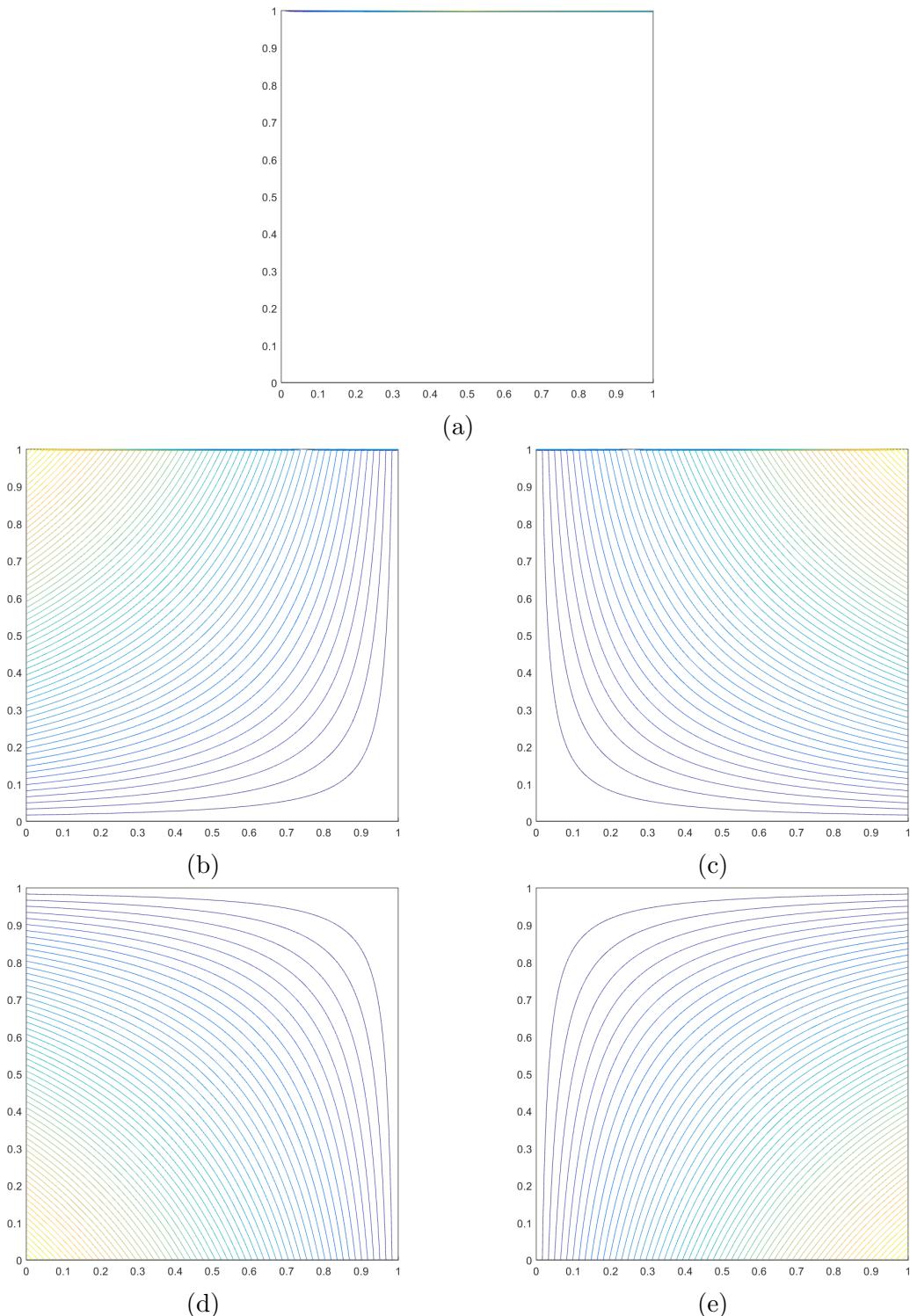


Figure 1.3: Contour plots of the linear Wachspress basis functions on the degenerate pentagon for the vertices located at: (a) $(1/2, 1)$, (b) $(0, 1)$, (c) $(1, 1)$, (d) $(0, 0)$, and (e) $(1, 0)$.

The 2D PWL functions are defined as combinations of linear triangular functions, with some of them only having measure within a subregion of a polygon. These subregions are formed by triangulating the arbitrary 2D polygon into a set of sub-triangles. Each sub-triangle is defined by two adjacent vertices (taken in a counter-clockwise ordering to maintain consistency) and the polygon's centroid. Looking at Figure 1.1 as an example, sub-triangle j is defined by the points $\{\vec{x}_j, \vec{x}_{j+1}, \vec{r}_c\}$, which are the polygon's vertices j and $j + 1$ and the polygon's centroid. If a polygon K has N_K vertices, then its centroid can be defined by

$$\vec{r}_c = \sum_{j=1}^{N_K} \alpha_j^K \vec{x}_j, \quad (1.17)$$

where α_j^K are the vertex weights functions and,

$$\sum_{j=1}^{N_K} \alpha_j^K = 1. \quad (1.18)$$

For this work, we continue to use the definition for the vertex weight functions from previous works [5, 6, 7],

$$\alpha_j^K = \frac{1}{N_K}. \quad (1.19)$$

This means that the weight functions are equal for every vertex and the cell centroid simply becomes the average position of all the vertices. However, we note that care must be taken so that the centroid does not lie on the polygon's boundary. This will cause the PWL functions to no longer have piecewise linearity along the boundary. Using these vertex weight functions, the PWL basis function for vertex j , b_j^{PWL} , is defined as



$$b_j^{PWL}(x, y) = t_j(x, y) + \alpha_j^K t_c(x, y). \quad (1.20)$$

In Eq. (1.20), t_j is the standard 2D linear function with unity at vertex j that linearly decreases to zero to the cell center and each adjoining vertex. t_c is the 2D cell “tent” function located at \vec{r}_c which is unity at the cell center and linearly decreases to zero to each cell vertex. α_j^K is the weight parameter for vertex j in cell K . The functional form of Eq. (1.20) with constant vertex weights means that the PWL function for vertex j , within the domain of K , linearly decreases to a value of $1/N_K$ at the polygonal center. From there, the function linearly decreases to zero on all faces that are not connected to vertex j . In Appendix ??, we detail how the 2D PWL coordinates can be analytically integrated using the reference triangle and affine mapping. The gradients of the PWL functions are easy to compute term-by-term in a straightforward manner:

$$\vec{\nabla} b_j^{PWL}(x, y) = \vec{\nabla} t_j(x, y) + \alpha_j^K \vec{\nabla} t_c(x, y). \quad (1.21)$$

We now give some example contour plots of the PWL coordinates over different polygons. First, we provide the contour plots for the four PWL functions on the unit square in Figure 1.4. In this example it is easy to discern the functional form of Eq. (1.20) with the use of constant vertex weights. We clearly see each function linearly decrease from its vertex to the cell center (with a value of $1/N_K$) and then linearly decrease to all non-adjoining faces. Next, Figure 1.5 provides the contour plots for the PWL functions on a degenerate (weakly-convex) pentagon where a fifth vertex was added to the unit square at $(1/2, 1)$. Unlike the Wachspress coordinates, the PWL functions work on weakly-convex polygons. The final example we give in Figure 1.6 is a favorite in the applied mathematics community: the “L-shaped”

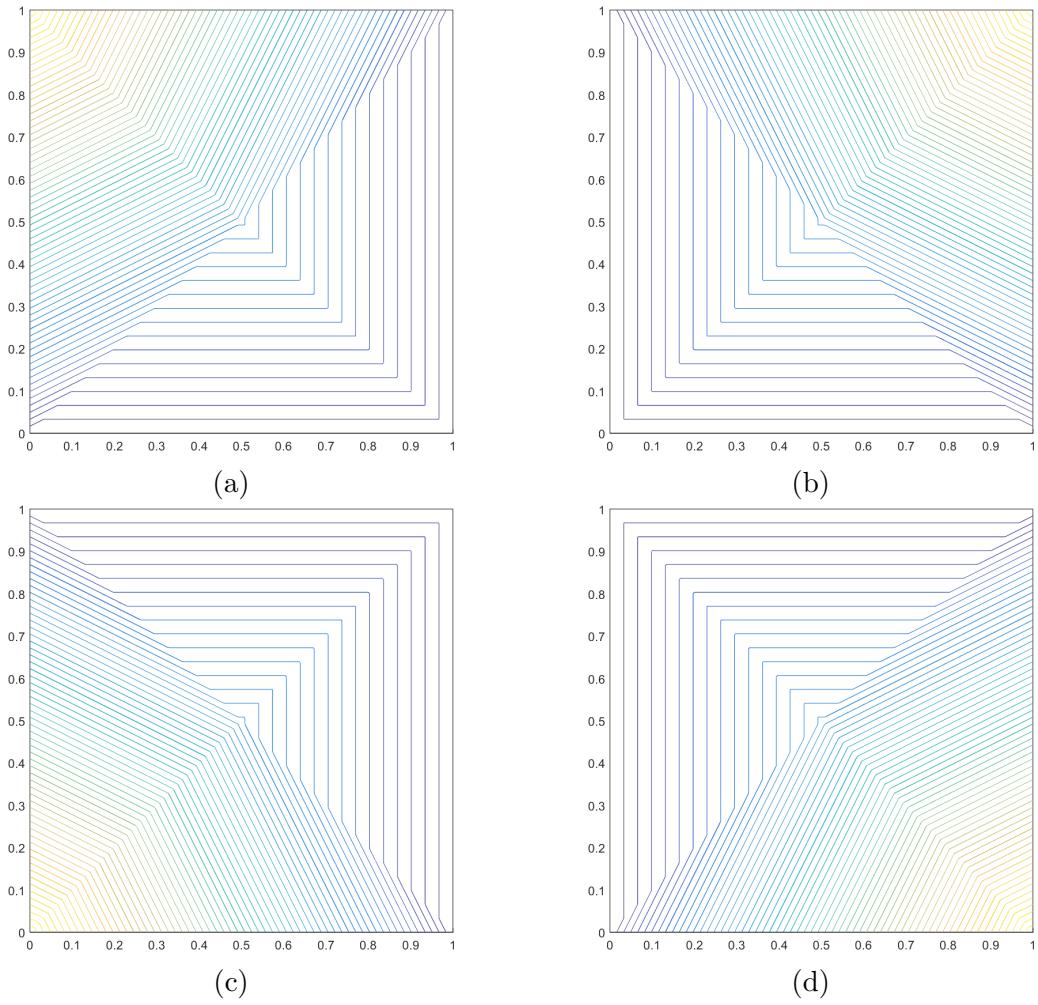


Figure 1.4: Contour plots of the linear PWL basis functions on the unit square for the vertices located at: (a) $(0,1)$, (b) $(1,1)$, (c) $(0,0)$, and (d) $(1,0)$.

domain. It provides an example of PWL's ability to still be linearly-complete on concave polygons. In this example, the cell centroid was forced to be at the point $(1/3, 1/3)$ so that it would be inside of the polygon.

1.1.3 Mean Value Basis Functions

At this point, we now introduce the first  polygonal basis set for use with the transport equation: the *mean value coordinates* (MV) developed by Floater [8]. The

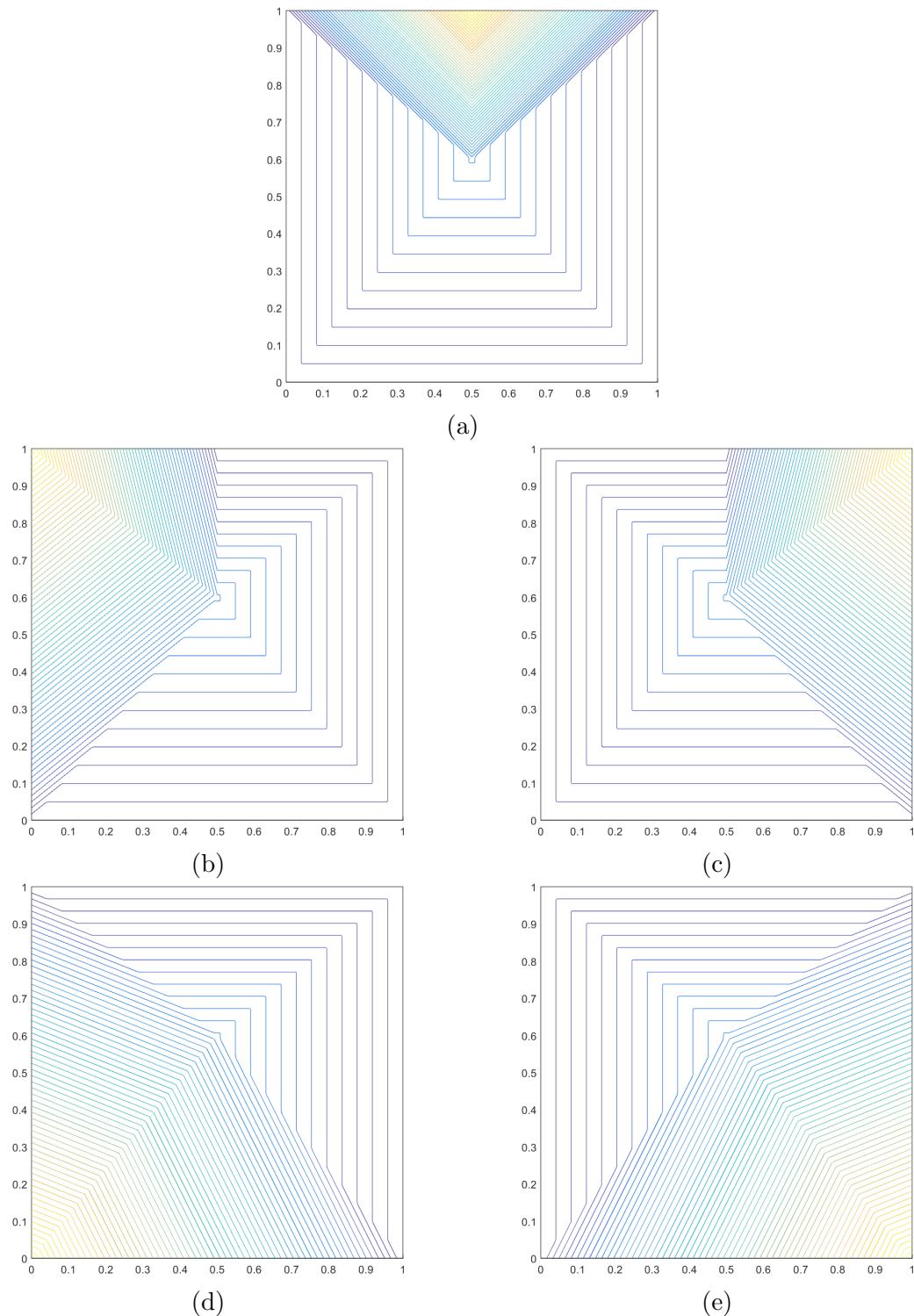


Figure 1.5: Contour plots of the linear PWL basis functions on the degenerate pentagon for the vertices located at: (a) $(1/2, 1)$, (b) $(0, 1)$, (c) $(1, 1)$, (d) $(0, 0)$, and (e) $(1, 0)$.

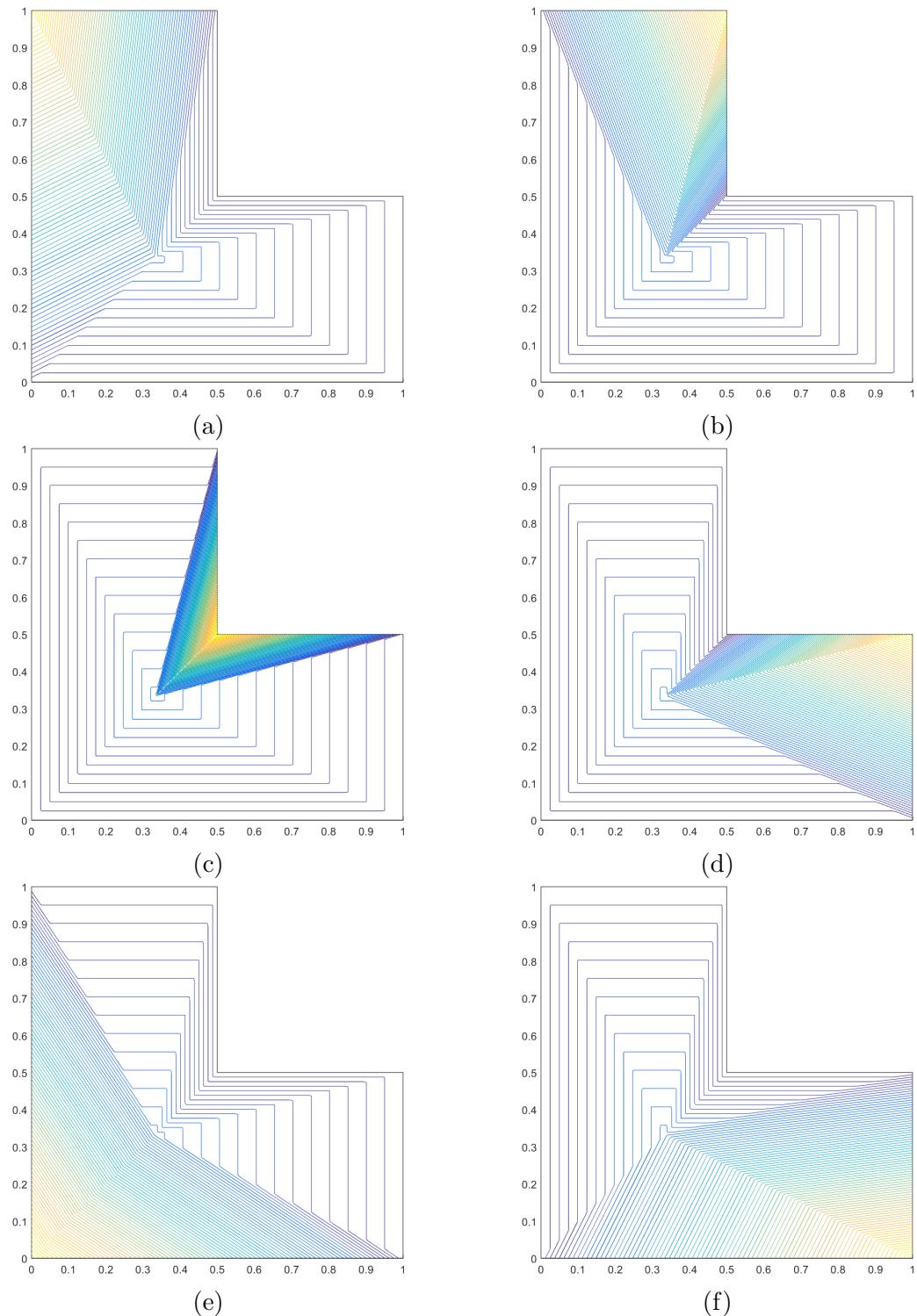


Figure 1.6: Contour plots of the linear PWL basis functions on the L-shaped domain for the vertices located at: (a) $(0,1)$, (b) $(1/2,1)$, (c) $(1/2,1/2)$, (d) $(1,1/2)$, (e) $(0,0)$, and (f) $(1,0)$.

original motivation behind the MV coordinates was to approximate harmonic maps on a polygon by a set of piecewise linear maps over a triangulation of the polygon for use in computer aided graphic design. Injectivity is preserved if the interpolatory function is *harmonic* over the piecewise linear maps. This can be shown by expressing a C^2 function u over each sub-triangle, \mathcal{T} , of the triangulated polygon and have it satisfy the Laplace equation,

$$\nabla^2 u = 0, \quad (1.22)$$

where $u(\vec{r}_b) = u_0$ consists of a piecewise linear Dirichlet boundary condition ($\vec{r}_b \in \partial\mathcal{T}$) for each triangulation. Then, by use of the mean value theorem (where these coordinates got their name), the mean value function at vertex j , b_j^{MV} , for a polygon K with N_K vertices can be given by

$$b_j^{MV}(\vec{x}) = \frac{w_j(\vec{x})}{\sum_i w_i(\vec{x})}, \quad (1.23)$$

where the mean value weight function for vertex j , w_j , has the following definition:

$$w_j(\vec{x}) = \frac{\tan(\alpha_{j-1}/2) + \tan(\alpha_j/2)}{|\vec{x}_j - \vec{x}|}. \quad (1.24)$$

This weight simply consists of the addition of the two tangent functions (where the angles are given in Figure 1.1) that is then divided through by the distance between the vertex j and the point of interest, \vec{x} . Through careful observation of Eq. (1.24), we can see that these MV weights are undefined on certain portions of the polygon's boundary, ∂K , in a similar way to the Wachspress coordinates. However, the limits of the coordinates are bounded on the polygon's faces and vertices, and we rigorously show this in Appendix ??.

We now give the form of the mean value gradients. Since the mean value coordinates given by Eq. (1.23) have the same form as the Wachspress coordinates, their gradients can be expressed in an **identical** manner. The gradients of the mean value coordinates can be expressed as

$$\vec{\nabla} b_j^{MV}(\vec{x}) = b_j^{MV}(\vec{x}) \left(\vec{R}_j(\vec{x}) - \sum_i b_i^{MV}(\vec{x}) \vec{R}_i(\vec{x}) \right), \quad (1.25)$$

where the reduced gradient, \vec{R}_j , still has the same definition from Eq. (1.14). If we define $t_j = \tan(\alpha_j/2)$ and $t_{j-1} = \tan(\alpha_{j-1}/2)$, then after extensive algebra, the mean value reduced gradients are

$$\vec{R}_j = \left(\frac{t_{j-1}}{t_{j-1} + t_j} \right) \frac{\{\vec{c}_{j-1}\}}{\sin(\alpha_{j-1})} + \left(\frac{t_j}{t_{j-1} + t_j} \right) \frac{\{\vec{c}_j\}}{\sin(\alpha_j)} + \frac{\vec{g}_j}{|\vec{x}_i - \vec{x}|}, \quad (1.26)$$

where

$$\vec{c}_j = \frac{\vec{g}_j}{|\vec{x}_j - \vec{x}|} - \frac{\vec{g}_{j+1}}{|\vec{x}_{j+1} - \vec{x}|}, \quad (1.27)$$

and

$$\vec{g}_j = \frac{\vec{x}_j - \vec{x}}{|\vec{x}_j - \vec{x}|}, \quad (1.28)$$

and

$$\{\vec{u}\} = (-u_2, u_1). \quad (1.29)$$

While this direct form for the MV coordinates is more complicated than the last two coordinates presented, it is still easily programmable. The interested reader can

look in the appendix of [9] for MATLAB code to compute these gradients, though we stress that it does not contain any logic for boundary detection.

We again provide example contour plots of the MV coordinates, and we use the same polygonal shapes that we showed for the PWL coordinates. Figure 1.7 gives the MV coordinates on the unit square. Like the Wachspress functions, the MV coordinates are smoothly varying within the domain of the polygon and possess continuous derivatives. Next in Figure 1.8, we give the example contour plots for the degenerate pentagon which is formed by inserting a vertex into the unit square at $(1/2, 1)$. Finally, Figure 1.9 gives the contour plots for the linear mean value coordinates on the L-shaped domain. We see that the linear mean value coordinates are applicable to interpolation on concave polygons.

1.1.4 Maximum Entropy Basis Functions

The final linearly-complete 2D basis functions that we will analyze in this work are generated by use of the *maximum entropy coordinates* (ME) [10, 11, 12]. These coordinates have garnered much interest in different fields because their functional forms can be tailored based on the application. The principle of maximum entropy stems from the concept of *Shannon entropy* [13]. If we define a functional for the Shannon entropy as H , then its maximum will lead to the least-biased statistical inference for some set of testable constraints [14]. For the application of FEM analysis, these testable constraints correspond to those given in Eq. (1.5). For N_K discrete probability functions (corresponding to the N_K vertex functions on polygon K), the functional form for the Shannon entropy can be given by

$$H(b, m) = - \sum_{j=1}^{N_K} b_j \log \left(\frac{b_j}{m_j} \right), \quad (1.30)$$

where m_j is called the prior distribution. These prior distributions are a key com-

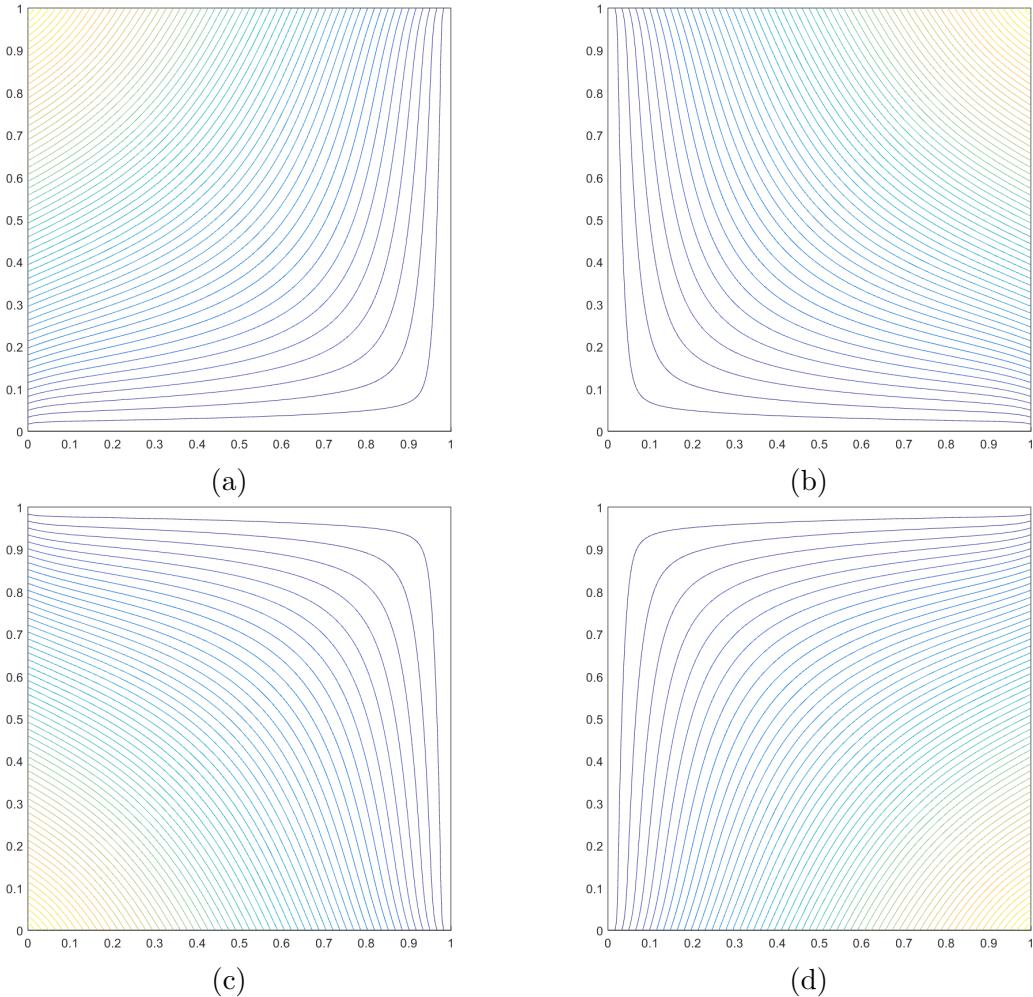


Figure 1.7: Contour plots of the linear mean value basis functions on the unit square for the vertices located at: (a) $(0,1)$, (b) $(1,1)$, (c) $(0,0)$, and (d) $(1,0)$.

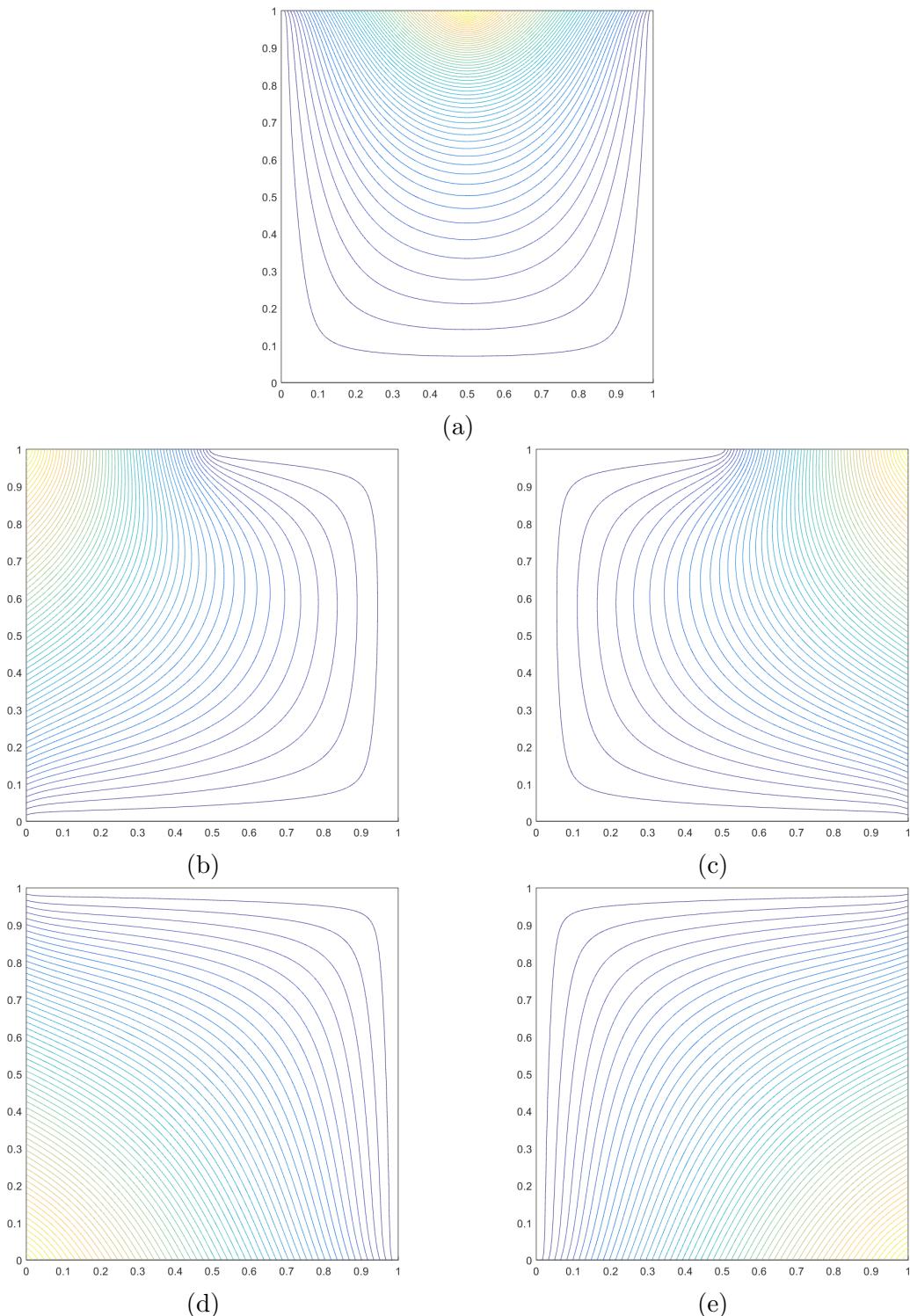


Figure 1.8: Contour plots of the linear mean value basis functions on the degenerate pentagon for the vertices located at: (a) $(1/2, 1)$, (b) $(0, 1)$, (c) $(1, 1)$, (d) $(0, 0)$, and (e) $(1, 0)$.

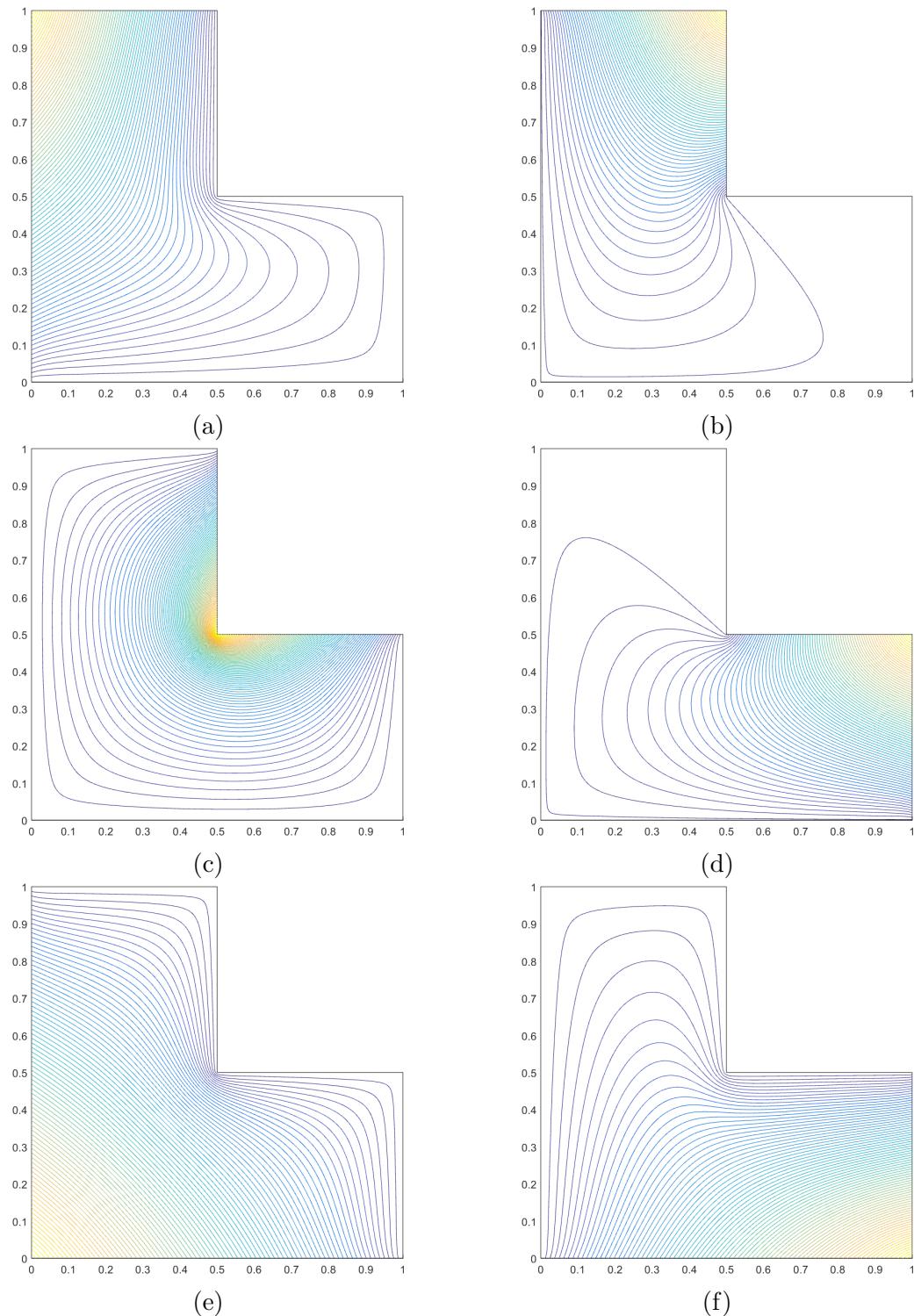


Figure 1.9: Contour plots of the linear mean value basis functions on the L-shaped domain for the vertices located at: (a) $(0,1)$, (b) $(1/2,1)$, (c) $(1/2,1/2)$, (d) $(1,1/2)$, (e) $(0,0)$, and (f) $(1,0)$.

ment of Bayesian inference [15, 16]. If $(b_j^{ME}(\vec{x}), j = 1, \dots, N_K)$ is the solution of the following constrained optimization problem

$$\boxed{\max_{b(\vec{x})}} H(b, m, \vec{x}), \quad (1.31)$$

then the maximum entropy coordinates can be given by

$$b_j^{ME}(\vec{x}) = \frac{w_j(\vec{x})}{\sum_i w_i(\vec{x})}. \quad (1.32)$$

In Eq. (1.32), the maximum entropy weight function for vertex j , w_j , has the following definition,

$$w_j(\vec{x}) = m_j(\vec{x}) \exp(-\vec{\kappa} \cdot (\vec{x}_j - \vec{x})), \quad (1.33)$$

where $\vec{\kappa}$ is a vector value of dimension d that will be explained shortly. In the context of Eq. (1.33), the prior distribution, m_j , can be viewed as a weight function associated with vertex j . This means that there is variability that one can employ for these weight functions. These weight functions can then be tailored depending on the application and the numerical scheme employed. For FEM applications, an optimal function form for the prior distribution is given by

$$m_j(\vec{x}) = \frac{\pi_j(\vec{x})}{\sum_k \pi_k(\vec{x})}, \quad (1.34)$$

where

$$\pi_j(\vec{x}) = \prod_{k \neq j-1, j} \rho_k(\vec{x}), \quad (1.35)$$

and

$$\rho_j(\vec{x}) = ||\vec{x} - \vec{x}_j|| + ||\vec{x} - \vec{x}_{j+1}|| - ||\vec{x}_{j+1} - \vec{x}_j||. \quad (1.36)$$

In Eqs. (1.35) and (1.36), we have defined a new weight function, ρ_j , that corresponds to face j between vertices j and $j + 1$. These face functions are zero along face j , but strictly positive elsewhere due to the triangle inequality. This means that the vertex function π_j is also non-negative and vanishes on all faces that are not adjacent to vertex j . This is important to ensure that each of the ME vertex functions are strictly zero on all non-adjacent faces. However, this also means that once again, the ME coordinates cannot be directly evaluated on the polygon's boundary. We show that the limits of these functions are bounded on the polygon's and vertices in Appendix ??.

Now that we have provided sufficient details for the basis functions and their weight functions, we can explain how the $\vec{\kappa}$ vector in Eq. (1.33) is computed. We can see that once $\vec{\kappa}$ is known, the ME functions can be directly calculated. To do this, we solve the constrained optimization problem of Eq. (1.31) through the use of Lagrange multipliers with a Newton's method. If we define κ_0 as the Lagrange multiplier for the constant constraint, then the Lagrangian for the problem of Eq. (1.31) is given by

$$\begin{aligned} \mathcal{L}(\vec{b}, \vec{\kappa}) = & - \sum_{j=1}^{N_K} b_j \log \left(\frac{b_j}{m_j} \right) - \kappa_0 \left(\sum_{j=1}^{N_K} b_j - 1 \right) \\ & - \vec{\kappa} \cdot \left(\sum_{j=1}^{N_K} b_j (\vec{x}_j - \vec{x}) \right) \end{aligned} \quad (1.37)$$

where we ~~referred~~ used the spatial parameter \vec{x} for brevity. If we take the first variation of \mathcal{L} to zero ($\delta \mathcal{L}(\vec{b}; \kappa_0, \vec{\kappa}) = 0$), then we obtain

$$\left[-1 - \log\left(\frac{b_j}{m_j}\right) - \kappa_0 - \vec{\kappa} \cdot (\vec{x}_j - \vec{x}) \right] \delta b_j = 0. \quad (1.38)$$

Since δb_j is arbitrary, everything within the bracket of Eq. (1.38) must be equal to zero. If we use the substitution $\log W = 1 + \kappa_0$ and extensive algebra, then we can rewrite Eq. (1.38) as

$$b_j(\vec{x}) = \frac{m_j(\vec{x}) \exp(-\vec{\kappa} \cdot (\vec{x}_j - \vec{x}))}{W}, \quad (1.39)$$

where $W = \prod_j w_j(\vec{x})$. We can immediately see that this satisfies the form for our maximum entropy coordinates given in Eq. (1.32). This means that we are just left with describing the non-linear numerical procedure to compute $\vec{\kappa}$. The solution of Eq. (1.39) is equivalent to solving the following dual unconstrained optimization problem [17]:

$$\vec{\kappa}^* = \arg \max F(\vec{\kappa}), \quad F(\vec{\kappa}) = \log W(\vec{\kappa}) \quad (1.40)$$

In Eq. (1.40), $F(\vec{\kappa})$ is the non-linear function to be solved with Newton's method. We summarize all the steps needed for computation as follows:

1. Compute and store $(\vec{x}_j - \vec{x})$ and the prior functions $m_j(\vec{x})$;
2. Start with iteration counter at $k = 0$ and initialize the Lagrange multiplier:
 $\vec{\kappa} = \vec{0}$;
3. Compute $\vec{g}_k = \vec{\nabla}_{\vec{\kappa}} F(\vec{\kappa}_k)$ and $\vec{H}_k = \vec{\nabla}_{\vec{\kappa}} \vec{\nabla}_{\vec{\kappa}} F(\vec{\kappa}_k)$;
4. Determine the Newton search direction: $\Delta \vec{\kappa}_k = -H_k^{-1} g_k$;
5. Update the multiplier: $\vec{\kappa}_{k+1} = \vec{\kappa}_k + \alpha \Delta \vec{\kappa}_k$;

6. Check convergence by testing if $\|g_{k+1}\| > \epsilon$.

7. Set $\vec{\kappa}^* = \vec{\kappa}_{k+1}$ and compute $\vec{b}(\vec{x})$

In these computational procedures, α is the  size. If the error at iteration k ,  is greater than 10^{-4} , then a line search algorithm is used [18]. Otherwise, α can be set to unity as the error decreases. We note that a line search algorithm must be used for certain classes of polygonal shapes. For extremely-distorted concave polygons this Newton iteration procedure can be unstable without it. Due to the quadratic convergence of Newton's method in the vicinity of the final solution, only 3-7 Newton iterations should be required to obtain accuracies of at least 10^{-10} .

We conclude our discussion of the maximum entropy coordinates, by providing example plots over the same polygonal domains that we showed for the PWL and MV coordinates. First, we give the contour plots of the ME coordinates over the unit square in Figure 1.10. Similar to the Wachspress and MV coordinates, the maximum entropy coordinates are smoothly varying within the domain of the polygon and possess continuous derivatives (in contrast to the PWL coordinates). Next in Figure 1.11, we give the contour plots of the degenerate polygon. Finally, Figure 1.12 gives the contour plots for the linear maximum entropy coordinates on the L-shaped domain. We can see that these coordinates can appropriately interpolate functions on concave polygons.

1.1.5 Summary of 2D Linear Basis Functions on Polygons

In Sections 1.1.1, 1.1.2, 1.1.3, and 1.1.4, we presented the linear Wachspress, PWL, mean value, and maximum entropy coordinates, respectively. We gave the functional forms for their values and gradients along with example contour plots for different polygonal elements. Table 1.1 provides a summary of the properties for the different coordinates. The Wachspress, PWL, and ME functions have natural

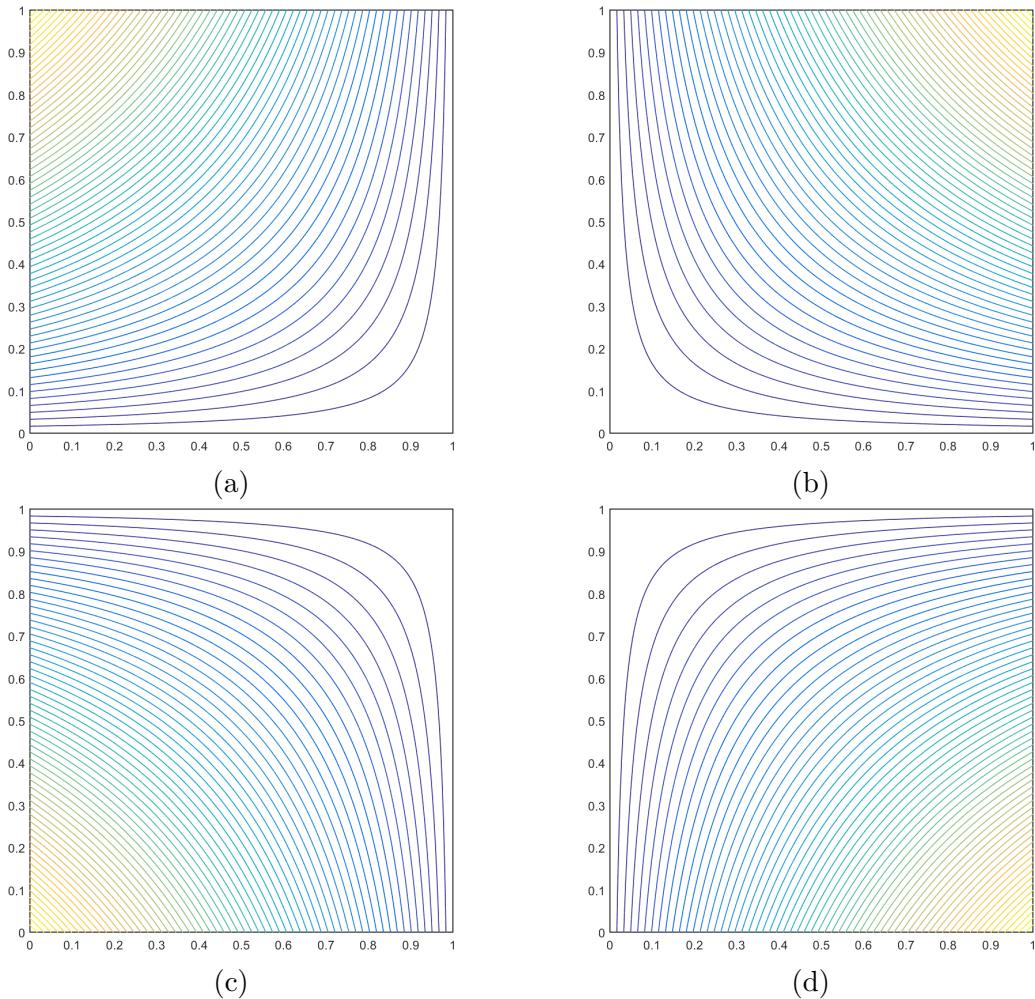


Figure 1.10: Contour plots of the linear maximum entropy basis functions on the unit square for the vertices located at: (a) $(0,1)$, (b) $(1,1)$, (c) $(0,0)$, and (d) $(1,0)$.

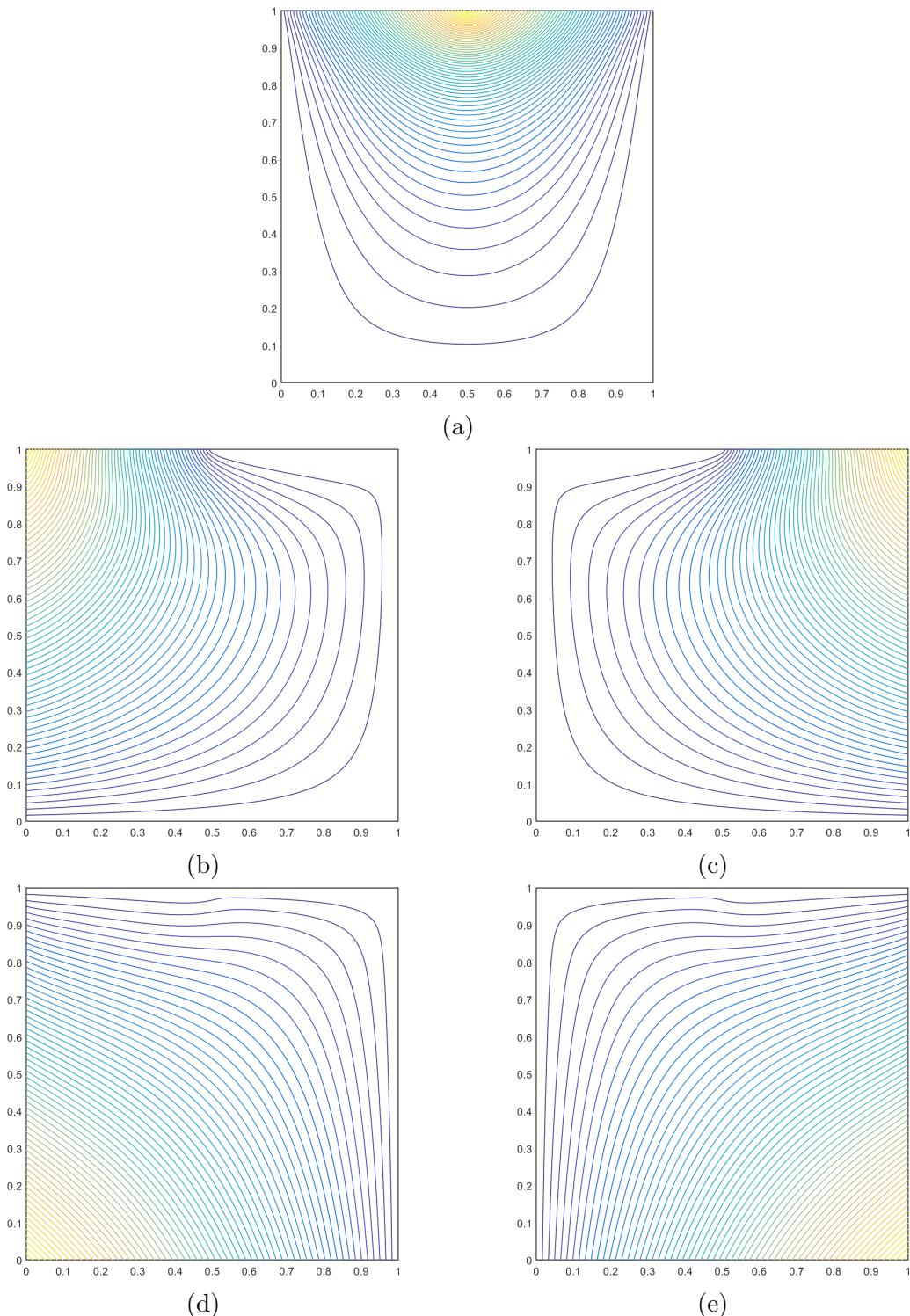


Figure 1.11: Contour plots of the linear maximum entropy basis functions on the degenerate pentagon for the vertices located at: (a) $(1/2, 1)$, (b) $(0, 1)$, (c) $(1, 1)$, (d) $(0, 0)$, and (e) $(1, 0)$.

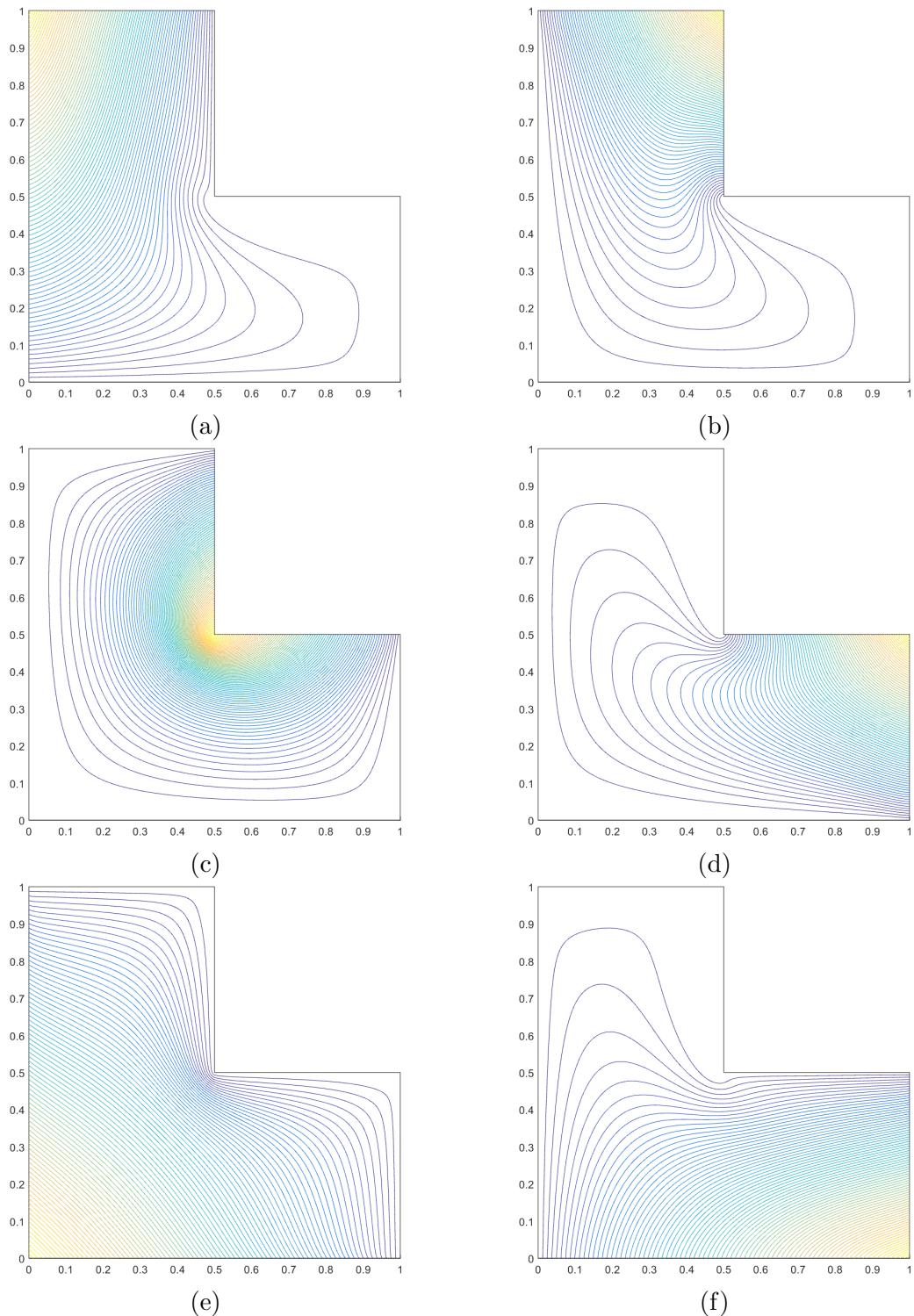


Figure 1.12: Contour plots of the linear maximum entropy basis functions on the L-shaped domain for the vertices located at: (a) $(0,1)$, (b) $(1/2,1)$, (c) $(1/2,1/2)$, (d) $(1,1/2)$, (e) $(0,0)$, and (f) $(1,0)$.

extensions to 3D polyhedra, but the MV functions can only interpolate on polyhedra with triangular facets [19, 20]. The PWL, MV, and ME coordinates can interpolate degenerate-convex and concave polygons while the Wachspress functions can only interpolate strictly-convex polygons. This means that the Wachspress coordinates are not suited for AMR calculations that form degenerate polygons. PWL is the only functional form that can analytically integrate the elementary matrices and directly evaluate its values and gradients on the boundary of the polygon. Finally, every point within the domain can be directly evaluated using the Wachspress, PWL, and MV coordinates. However, the ME coordinates use an iterative approach with Newton’s method since their functional form constitutes a non-linear minimization problem.

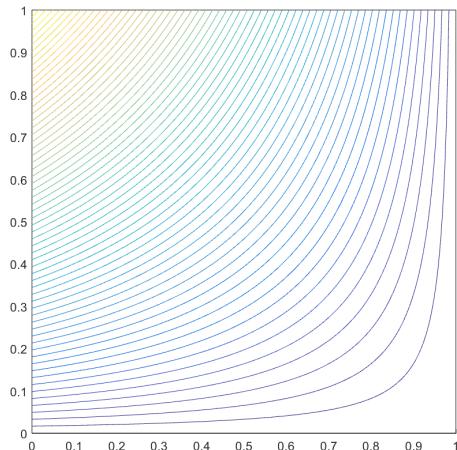
We conclude this summary discussion on the different linearly-complete 2D polygonal coordinates by again presenting examples of their functional forms. Figure 1.13 provides the contour plots of the different coordinates located at vertex $(0, 1)$ on the unit square. It is easy to see that the Wachspress, MV, and ME coordinates are smoothly varying within the polygon’s domain while the PWL coordinates only have C^0 continuity. Figure 1.14 provides the contour plots on the degenerate pentagon that is formed by inserting a vertex at $(1/2, 1)$ into the unit square. This re-emphasizes that the Wachspress coordinates are only valid interpolatory functions on strongly-convex polygons. Finally, Figure 1.15 provides the contour plots of the PWL, MV, and ME functions on the L-shaped domain at the $(0, 1)$ and $(1/2, 1/2)$ vertices. These coordinates can successfully interpolate on concave polygons.

1.2 Converting the Linear Polygonal Basis Functions to the Quadratic Serendipity Space of Functions

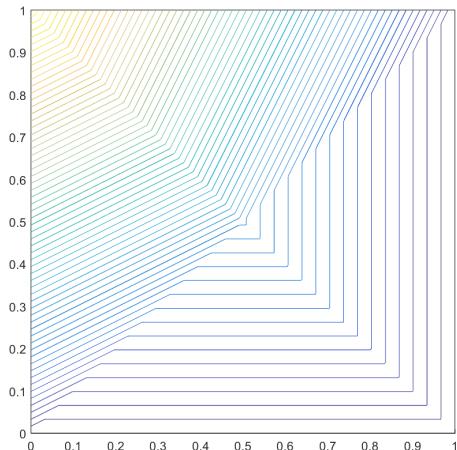
We have given complete details on the linearly-complete generalized barycentric coordinates that we will investigate for this work. Now we describe how to con-

Table 1.1: Summary of the properties of the 2D coordinate systems used on polygons.

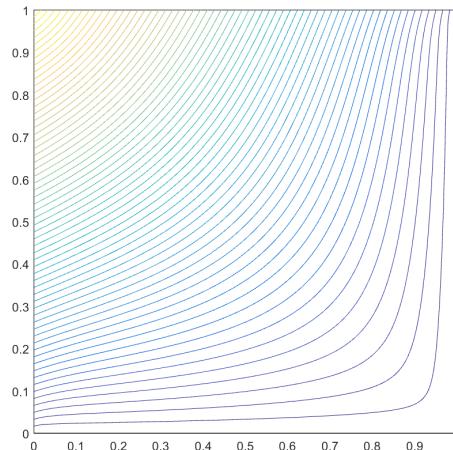
Basis Function	Dimension	Polygon Type	Integration	Evaluation
Wachspress	2D/3D	Convex	Numerical	Direct
PWL	1D/2D/3D	Convex/Concave	Analytical	Direct
Mean Value	2D	Convex/Concave	Numerical	Direct
Max Entropy	1D/2D/3D	Convex/Concave	Numerical	Iterative



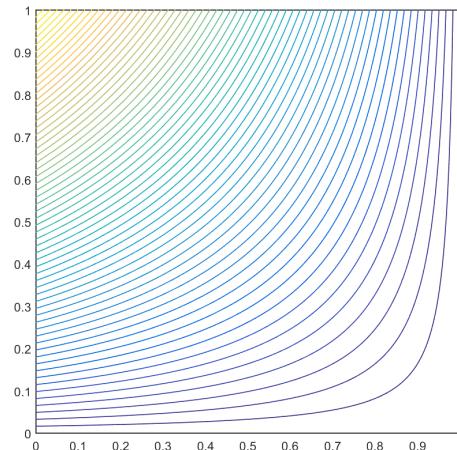
(a) Wachspress



(b) PWL



(c) Mean Value



(d) Maximum Entropy

Figure 1.13: Contour plots of the different linear basis function on the unit square located at vertex $(0,1)$.

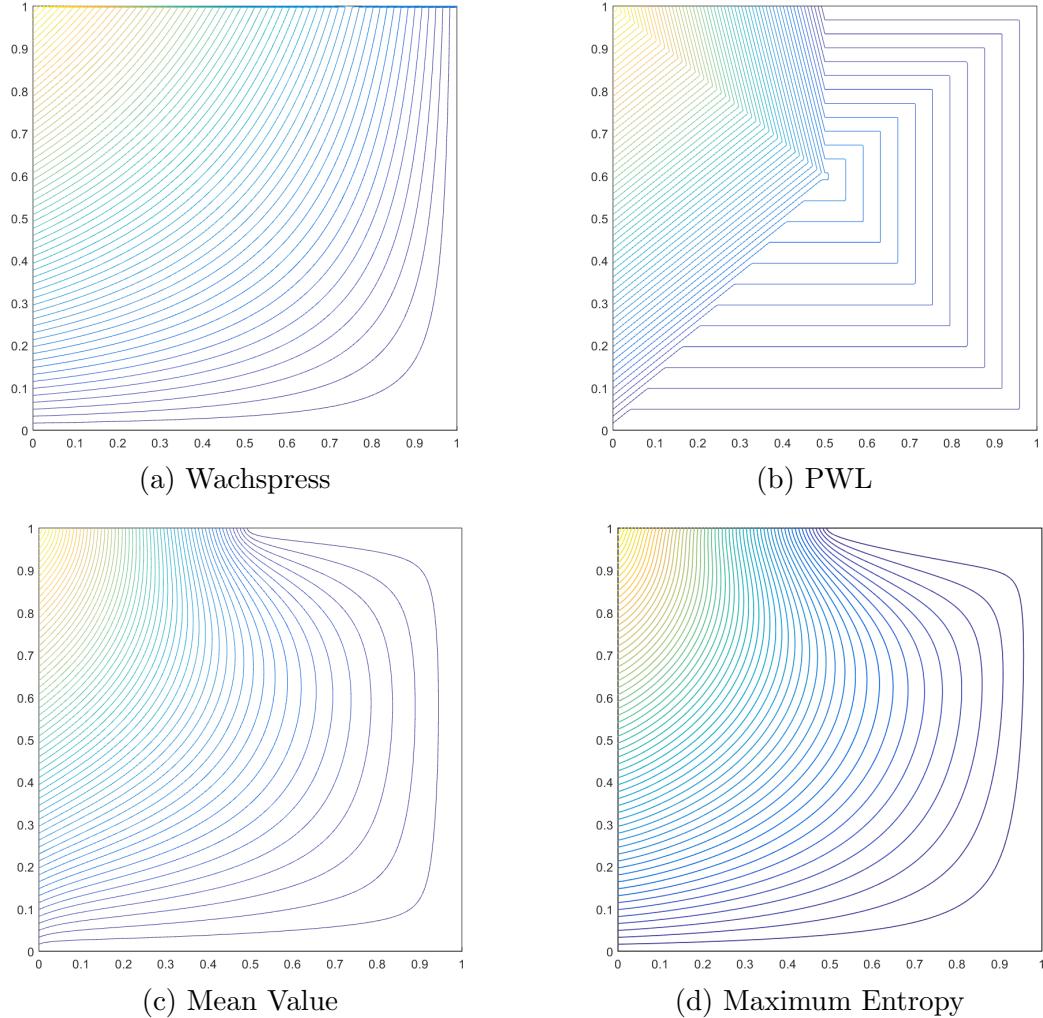


Figure 1.14: Contour plots of the different linear basis function on the degenerate pentagon located at vertex $(0,1)$. It is clear that the Wachspress coordinates fail for the weakly convex case.

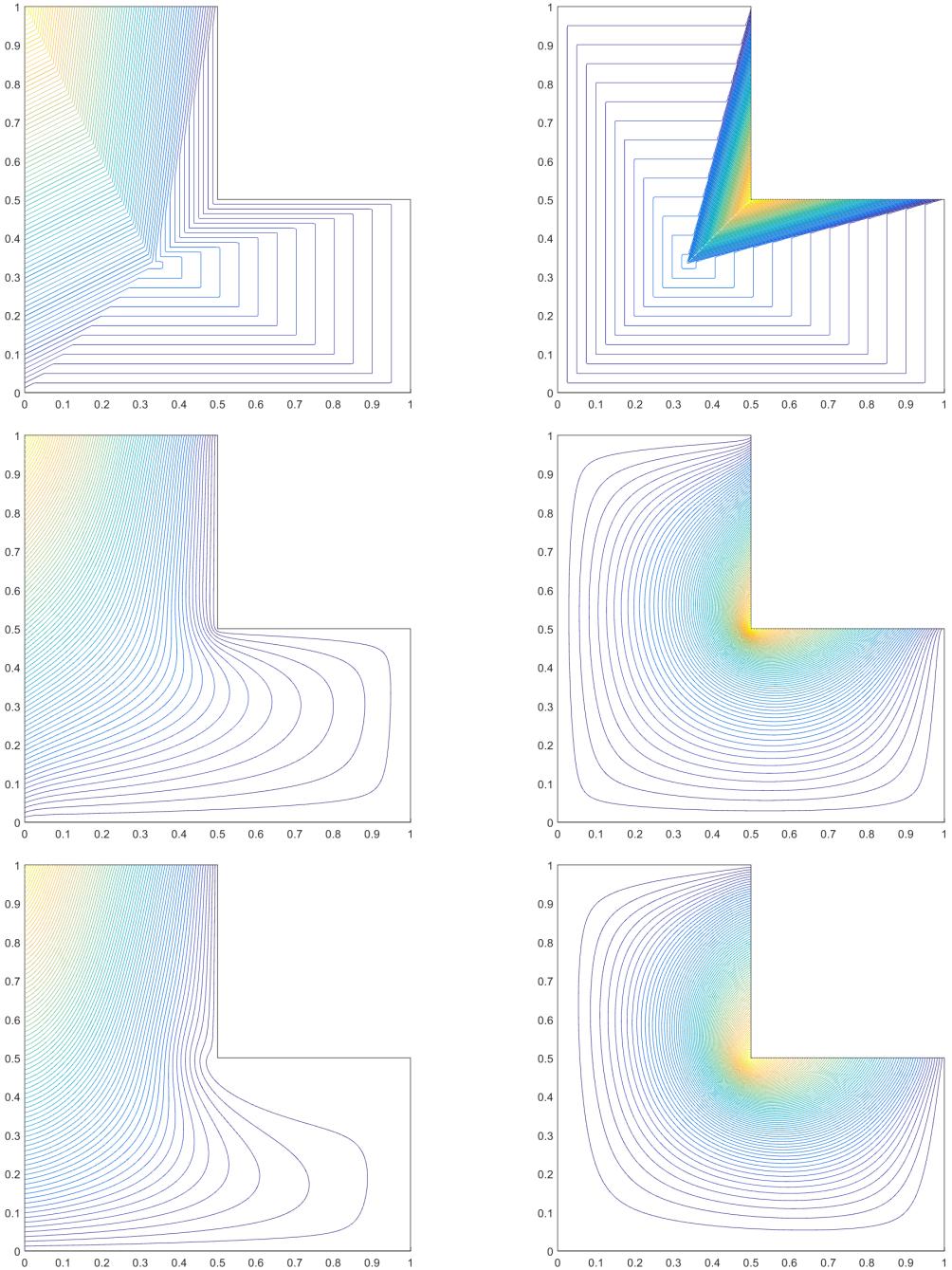


Figure 1.15: Contour plots of the different linear basis functions on the L-shaped domain. The PWL (top), mean value (middle), and maximum entropy (bottom) functions are plotted at vertices $(0, 1)$ (left) and $(1/2, 1/2)$ (right).

vert any generalized barycentric coordinate into the quadratic serendipity space of functions to yield quadratic (not linear) precision based on the work of Rand et al. [2]. The maximum entropy coordinates were independently converted into the quadratic space [21, 22]. These 2D serendipity coordinates can exactly interpolate the $\{1, x, y, x^2, xy, y^2\}$ span of functions, which can be shown with the first three levels of Pascal's triangle,

$$\begin{array}{ccc} & & 1 \\ & x & y \\ x^2 & xy & y^2 \end{array} \quad (1.41)$$

Between the linear and quadratic precision of the generalized barycentric coordinates and the quadratic serendipity extension, a relation can be formed on the functional space of their precision. If we seek a functional space up to order p precision, then for $k = 0, \dots, p$, the interpolatory functions must exactly span the following monomial functions,

$$f_{\sigma,\tau}^k(x, y) = x^\sigma y^\tau, \quad (1.42)$$

where $\sigma + \tau = k$. From the first three levels of Pascal's triangle given in Eq. (1.41), we see that $k = 0$ gives the constant function $\{1\}$, that $k = 1$ gives the linear functions $\{x, y\}$, and that $k = 2$ gives the quadratic functions $\{x^2, xy, y^2\}$.

Now, we give the full details on converting the linear generalized barycentric coordinates to the quadratic serendipity space of functions. Figure 1.16 gives a visual depiction of this conversion process. For a polygon with N_K vertices, we can summarize this procedure as the following:

1. For a point \vec{x} , compute the N_K linear barycentric functions $\boxed{\text{B}}(\vec{x})\}$;

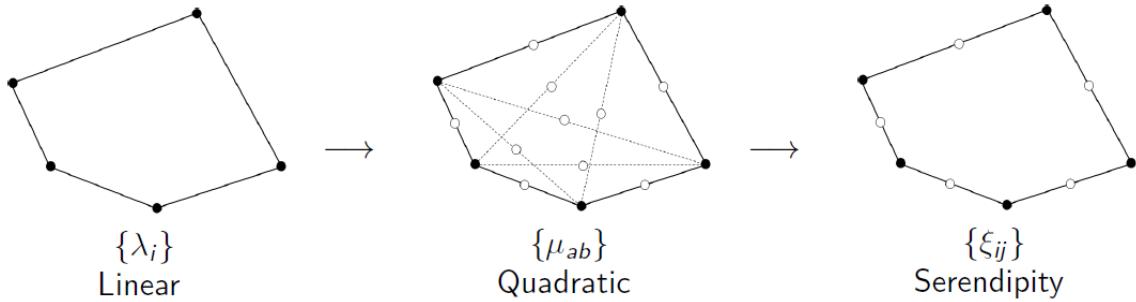


Figure 1.16: Overview of the process to construct the quadratic serendipity basis functions on polygons. The filled dots correspond to basis functions that maintain the Lagrange property while empty dots do not.

2. Take all non-repeating pairwise products of the linear functions to obtain $\frac{N_K(N_K+1)}{2}$ quadratic functions $\{\mu_{ab}\}$;
3. Form the linear transformation matrix \mathbb{A} through the use of monomial constraint equations;
4. Use the \mathbb{A} matrix to reduce the $\{\mu_{ab}\}$ function set to the $2N_K$ serendipity basis set $\{\xi_{ij}\}$.

We begin by computing the ($i = 1, \dots, N_K$) linear barycentric functions, $\{\lambda_i(\vec{x})\}$, and their gradients, $\{\vec{\nabla} \lambda_i(\vec{x})\}$, for a point \vec{x} . These linearly-complete barycentric functions can be converted immediately to barycentric-like functions with quadratic precision. Taking all non-repeating pairwise products of the linear functions yields a total of $N_Q = \frac{N_K(N_K+1)}{2}$ quadratic functions: $\mu_{ab} = \lambda_a \lambda_b$. Doing this generates functions that either live on the polygon's vertices, mid-face points, or midpoints of the polygon's diagonals between two vertices as seen in Figure 1.16. The N_K vertex functions are denoted as $\boxed{\text{---}}$, the N_K mid-edge functions are denoted as $a \boxed{\text{---}} E$, and the $\frac{N_K(N_K-3)}{2}$ mid-diagonal functions are denoted as $a \boxed{\text{---}} D$. We also define the abbreviated notation of $\vec{x}_{ab} = \frac{\vec{x}_a + \vec{x}_b}{2}$, so that \vec{x}_{aa} corresponds to a vertex function at

\vec{x}_a . Using these various notations, we can write the precision properties of the μ_{ab} functions for the constant constraint,

$$\sum_{aa \in V} \mu_{aa}(\vec{x}) + \sum_{ab \in E \cup D} \text{[highlighted function]} = 1, \quad (1.43)$$

for the linear constraint,

$$\sum_{aa \in V} \mu_{aa}(\vec{x}) \vec{x}_{aa} + \sum_{ab \in E \cup D} 2\mu_{ab}(\vec{x}) \vec{x}_{ab} = \vec{x}, \quad (1.44)$$

and for the quadratic constraint,

$$\sum_{aa \in V} \mu_{aa}(\vec{x}) (\vec{x} \text{[highlighted function]} \vec{x}_a) + \sum_{ab \in E \cup D} \mu_{ab}(\vec{x}) (\vec{x}_a \otimes \vec{x}_b + \vec{x}_b \otimes \vec{x}_a) = \vec{x} \otimes \vec{x}. \quad (1.45)$$

We can immediately see that quadratic precision is ensured. However, the set of these pairwise quadratic functions grows quadratically. This means that as N_K grows large, the number of interpolatory grows as order $O(N_K^2)$, but still only maintains precision of the $\{1, x, y, x^2, xy, y^2\}$ span of functions. Therefore, the computational work required to utilize these quadratic barycentric functions can become prohibitive for polygons with large vertex counts.

To minimize the number of interpolatory functions but still maintain the precision of the $\{1, x, y, x^2, xy, y^2\}$ span of functions, we seek to convert the quadratic barycentric functions into the quadratic serendipity space of functions $\{\xi_{ij}\}$. This quadratic serendipity space only contains the vertex and mid-face functions (total of $2N_K$) and has been extensively studied for tensor-based elements in the past [23, 24]. This means that we seek to reduce the $\{\mu_{ab}\}$ set of functions by removing the diagonal functions ($ab \in D$), but still maintaining quadratic precision. If we define ξ_{ii}

and $\xi_{i(i+1)}$ as the serendipity functions that live at vertex i and the mid-face point between vertices i and $i+1$, respectively, then we can write the serendipity precision properties for the constant constraint,

$$\sum_{\substack{\text{---} \\ \text{---}}} \xi_{ii}(\vec{x}) + \sum_{\substack{\text{---} \\ i \in E}} 2\xi_{i(i+1)}(\vec{x}) = 1, \quad (1.46)$$

for the linear constraint,

$$\sum_{ii \in V} \xi_{ii}(\vec{x}) \vec{x}_{ii} + \sum_{i(i+1) \in E} 2\xi_{i(i+1)}(\vec{x}) \vec{x}_{i(i+1)} = \vec{x}, \quad (1.47)$$

and for the quadratic constraint,

$$\sum_{ii \in V} \xi_{ii}(\vec{x}) (\vec{x}_i \otimes \vec{x}_i) + \sum_{i(i+1) \in E} \xi_{i(i+1)}(\vec{x}) (\vec{x}_i \otimes \vec{x}_{i+1} + \vec{x}_{i+1} \otimes \vec{x}_i) = \vec{x} \otimes \vec{x}. \quad (1.48)$$

To remove the diagonal functions, we formalize this procedure by recasting it as a linear algebra problem. We seek a matrix \mathbb{A} such that the linear transformation,

$$\{\xi\} = \mathbb{A} \{\mu\}, \quad (1.49)$$

will satisfy the precision properties of Eqs. (1.46 - 1.48). It is easy to see that \mathbb{A} has dimension $(2N \times N_Q)$. We wish for this matrix to have constant entries for any point within the polygon's interior so that it does not have to be recalculated for each interpolatory point of interest. To ease the notation, we will assign specific basis orderings for the quadratic and quadratic serendipity functions. The serendipity basis is ordered such that all vertex functions ($ii \in V$) in a counter-clockwise ordering are first, followed by a counter-clockwise ordering of the mid-face nodes ($ij \in E$) starting

with the node between vertices 1 and 2. This ordering can be succinctly stated by

$$\{\xi_{ij}\} = \left\{ [\xi_{11}, \xi_{22}, \dots, \xi_{N_K N_K}], [\xi_{12}, \xi_{23}, \dots, \xi_{(N_K-1)N_K}, \xi_{N_K}(\text{[} \text{])}_1] \right\}. \quad (1.50)$$

The quadratic basis begins identically to the serendipity basis by first listing the vertex and mid-face functions. Then the diagonal functions are indexed in lexicographical order. This gives the following ordering for the quadratic functions

$$\{\mu_{ab}\} = \left\{ [\mu_{11}, \mu_{22}, \dots, \mu_{N_K N_K}], [\mu_{12}, \mu_{23}, \dots, \mu_{(N_K-1)N_K}, \mu_{N_K(N_K+1)}], \right. \\ \left. [\mu_{13}, \dots, (\text{lexicographical}), \dots, \mu_{(N_K-2)N_K}] \right\}. \quad (1.51)$$

With these basis orderings, we can now denote the entries of \mathbb{A} (given by c_{ab}^{ij}) as the following:

$$\mathbb{A} = \begin{bmatrix} c_{11}^{11} & \dots & c_{ab}^{11} & \dots & c_{(\text{[} \text{])}_n}^{11} \\ \dots & \ddots & \vdots & \ddots & \vdots \\ c_{11}^{ij} & \dots & c_{ab}^{ij} & \dots & c_{(n-2)n}^{ij} \\ \dots & \ddots & \vdots & \ddots & \vdots \\ c_{11}^{n(n+1)} & \dots & c_{ab}^{n(n+1)} & \dots & c_{(n-2)n}^{n(n+1)} \end{bmatrix}. \quad (1.52)$$

A sufficient set of constraints for the entries in \mathbb{A} to ensure the precision properties of Eqs. (1.46 - 1.48) can be written for the constant constraint,

$$\sum_{ii \in V} c_{aa}^{ii} + \sum_{i(i+1) \in E} 2c_{(\text{[} \text{])}_1}^{11} = 1, \quad \forall aa \in V \\ \sum_{ii \in V} c_{ab}^{ii} + \sum_{i(i+1) \in E} 2c_{ab}^{i(i+1)} = 2, \quad \forall ab \in E \cup D \quad (1.53)$$

for the linear constraint,

$$\begin{aligned} \sum_{ii \in V} c_{aa}^{ii} \vec{x}_{ii} + \sum_{i(i+1) \in E} 2c_{aa}^{i(i+1)} \vec{x}_{i(i+1)} &= \vec{x}_{aa}, \quad \forall aa \in V \\ \sum_{ii \in V} c_{ab}^{ii} \vec{x}_{ii} + \sum_{i(i+1) \in E} 2c_{ab}^{i(i+1)} \vec{x}_{i(i+1)} &= 2\vec{x}_{ab}, \quad \forall ab \in E \cup D \end{aligned} \quad (1.54)$$

for the $a \in V$ vertex quadratic constraints

$$\sum_{ii \in V} c_{aa}^{ii} \vec{x}_{ii} \otimes \vec{x}_{ii} + \sum_{i(i+1) \in E} c_{aa}^{i(i+1)} (\vec{x}_i \otimes \vec{x}_{i+1} + \vec{x}_{i+1} \otimes \vec{x}_i) = \vec{x}_{aa} \otimes \vec{x}_{aa}, \quad (1.55)$$

and for the $ab \in E \cup D$ mid-face and diagonal quadratic constraints,

$$\sum_{ii \in V} c_{ab}^{ii} \vec{x}_{ii} \otimes \vec{x}_{ii} + \sum_{i(i+1) \in E} c_{ab}^{i(i+1)} (\vec{x}_i \otimes \vec{x}_{i+1} + \vec{x}_{i+1} \otimes \vec{x}_i) = (\vec{x}_a \otimes \vec{x}_b + \vec{x}_b \otimes \vec{x}_a). \quad (1.56)$$

For $N_K > 3$, there are more coefficients than the six constraint equations. This means that there is flexibility in the construction of the solution to the constraint equations. Therefore, we choose a simple structure for \mathbb{A} that consists of the following,

$$\mathbb{A} = [\mathbb{I} | \mathbb{A}'], \quad (1.57)$$

where \mathbb{I} is the $(2N_K \times 2N_K)$ identity matrix, and \mathbb{A}' is a full $(2N_K \times \frac{N_K(N_K-3)}{2})$ matrix. This means that the vertex and face midpoint serendipity functions, ξ_{ij} , are formed by taking their corresponding quadratic function, μ_{ij} , and adding some linear combination of the interior functions. Therefore, we only need to determine the $\frac{N_K(N_K-3)}{2}$ columns of the \mathbb{A}' matrix to complete this linear transformation.

In their work, Rand et al. proposed a methodology where only six coefficients are

chosen to be non-zero and can be directly calculated through geometric expressions. However, their approach is only valid for strictly-convex polygons. This will not work for our analysis since we wish to also analyze degenerate polygons that will arise in our AMR calculations. Therefore, we will use a least squares method to calculate each of the columns of \mathbb{A}' . We note that the coefficients calculated with this least squares method and that of Rand will be identical **only** for **orthogonal quadrilaterals**. If we isolate the column (ab) from \mathbb{A}' , then we can form the following system of equations,

$$\mathbb{B}\vec{c}_{ab} = \vec{q}_{ab}, \quad (1.58)$$

where \mathbb{B} is a matrix of dimension $(6 \times 2N_K)$, \vec{c}_{ab} is vector of length $2N_K$, and \vec{q}_{ab} is a vector of length 6. The entries of \mathbb{B} correspond to coefficients given in the left-hand-side terms of Eqs. (1.53), (1.54), and (1.56). The values of \vec{q}_{ab} are given by the right-hand-side constants of these same equations. To invert \mathbb{B} , we use the Moore-Penrose pseudoinverse (denoted as \mathbb{B}^*) [25]. For an under-determined system of equations, the pseudoinverse is given by,

$$\mathbb{B}^* = \mathbb{B}^T (\mathbb{B}\mathbb{B}^T)^{-1}. \quad (1.59)$$

Once all of the coefficients for the \mathbb{A}' matrix are known, each of the quadratic serendipity functions can be computed by

$$\begin{aligned} \text{[yellow box]} &= \mu_{i,j} + \sum_{ab \in D} c_{ab}^{ij} \mu_{ab} \\ &= \lambda_i \lambda_j + \sum_{ab \in D} c_{ab}^{ij} \lambda_a \lambda_b \end{aligned} . \quad (1.60)$$

The gradients of the serendipity basis are simple to compute with the chain rule

of Calculus. If we take the gradient of Eq. (1.60) and use appropriate derivative rules, then the gradients of the different serendipity functions can be given by

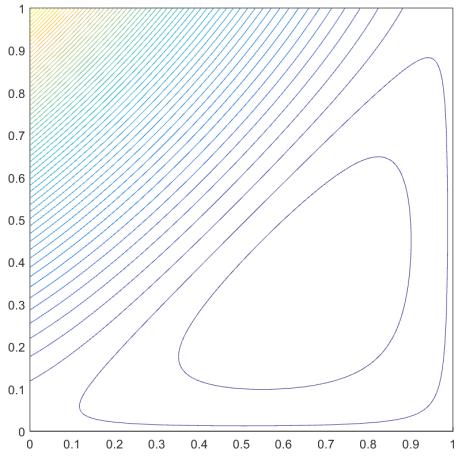
$$\vec{\nabla} \xi_{ij} = \lambda_j \vec{\nabla} \lambda_i + \lambda_i \vec{\nabla} \lambda_j + \sum_{ab \in D} c_{ab}^{ij} \left(\lambda_b \vec{\nabla} \lambda_a + \lambda_a \vec{\nabla} \lambda_b \right). \quad (1.61)$$

This means that all of the serendipity basis function gradients can be computed from the appropriate values and gradients of the linear barycentric basis functions using the linear transformation of the \mathbb{A} matrix.

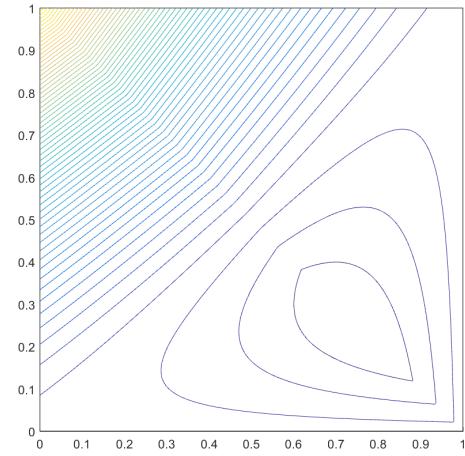
We now present some example contour plots for the conversion of the linear barycentric coordinates into the quadratic serendipity space. Figures 1.17 and 1.18 provide the contour plots of the different quadratic serendipity functions located at the upper-left vertex and left side-node, respectively. Then, Figure 1.19 provides some of the contour plots of the PWL, MV, and ME basis functions on the degenerate square that is formed by inserting a vertex at $(1/2, 1)$ onto the unit square.

1.3 Integrating the Arbitrary 2D Polygonal Elements

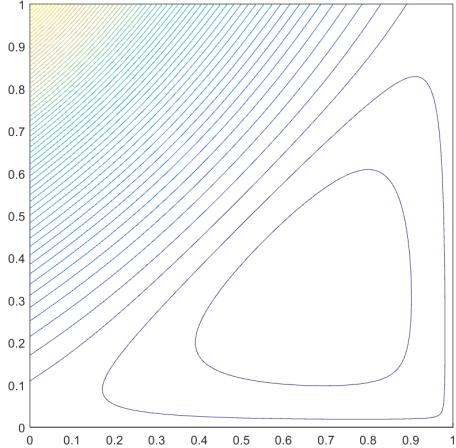
Sections 1.1 and 1.2  tell how the basis functions and their gradients can be computed at different points on a 2D polygonal element. These basis functions and gradients can then be used to calculate the integrals of the elementary matrices for a given element K as described in Section ???. Because the elementary matrix integrals using the Wachspress, mean value, and maximum entropy coordinates cannot be performed analytically, we need to define a numerical quadrature scheme. The spatial quadrature sets need to be amenable to arbitrary polygons and also integrate polynomials exactly (the different polynomial orders of the basis functions). Efficient quadrature schemes exist for both triangles and quadrilaterals [26, 27, 28, 29, 30]. These include symmetric rules on triangles and cubature and tensor-product rules on triangles and quadrilaterals, respectively. However, polygons have an infinite num-



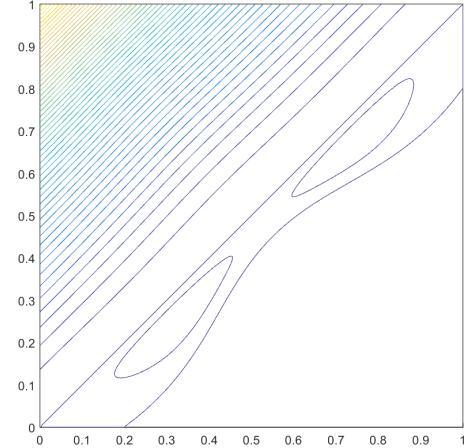
(a) Wachspress



(b) PWL

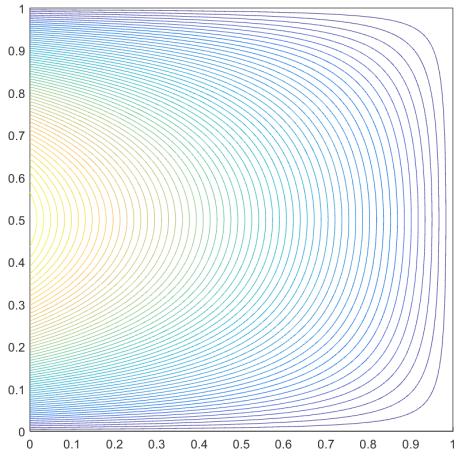


(c) Mean Value

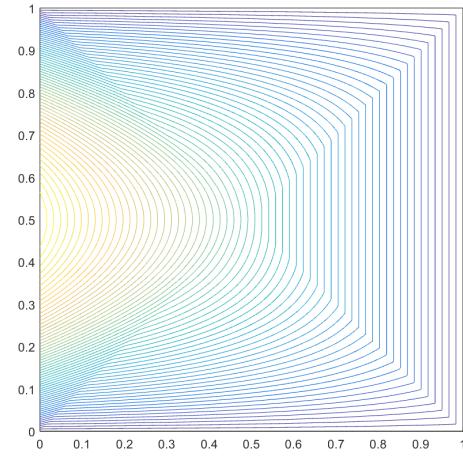


(d) Maximum Entropy

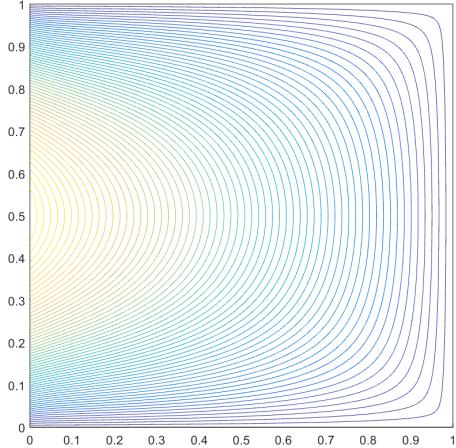
Figure 1.17: Contour plots of the different quadratic serendipity basis function on the unit square located at vertex $(0,1)$.



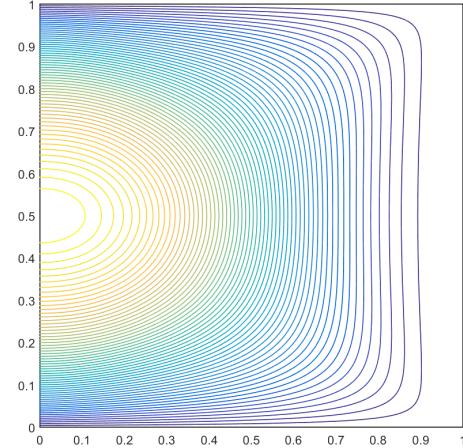
(a) Wachspress



(b) PWL



(c) Mean Value



(d) Maximum Entropy

Figure 1.18: Contour plots of the different quadratic serendipity basis function on the unit square at a mid-face node located at $(0, 1/2)$.

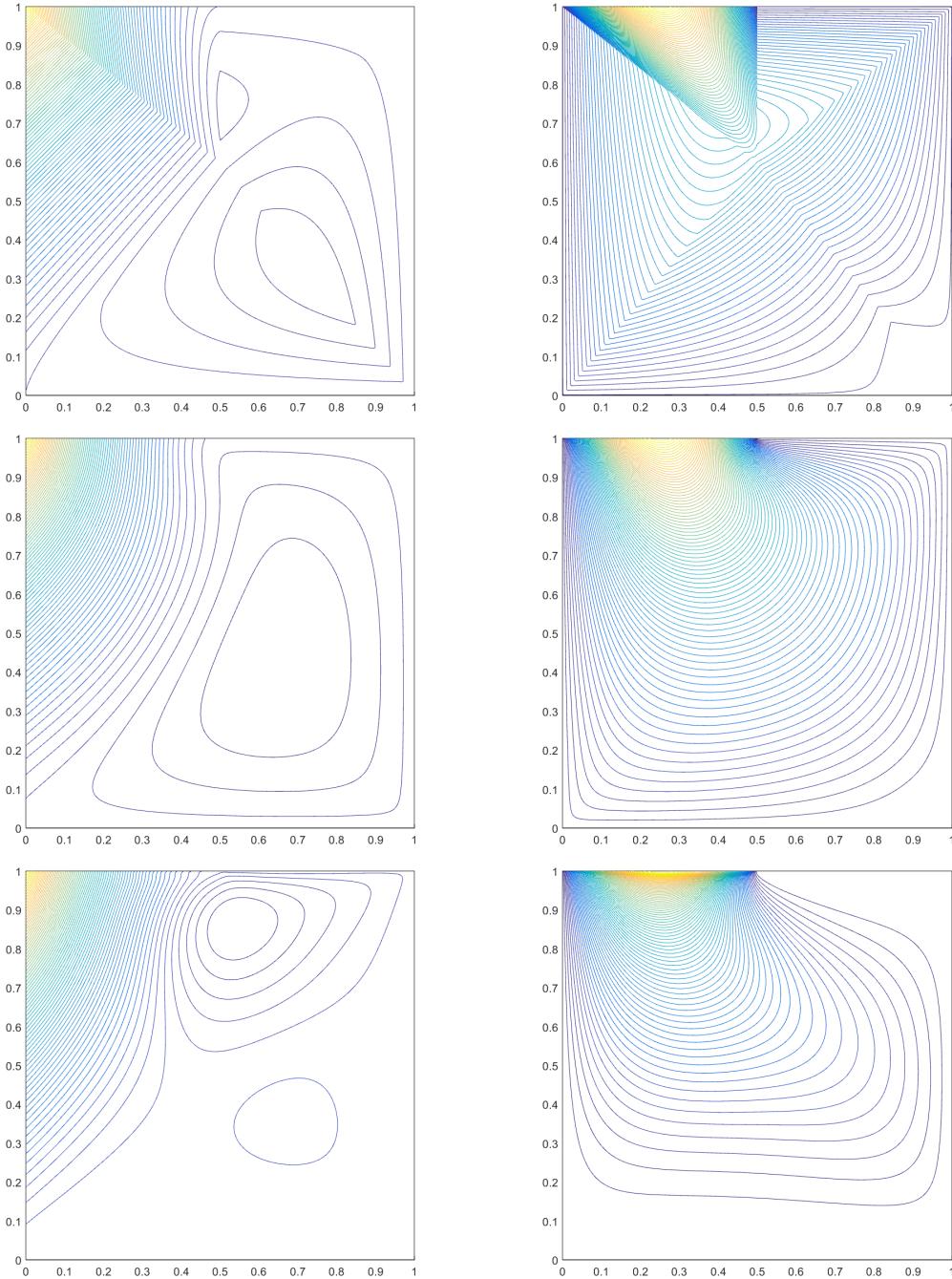


Figure 1.19: Contour plots of the different quadratic serendipity basis functions on the degenerate square. The PWL (top), mean value (middle), and maximum entropy (bottom) functions are plotted at vertices $(0, 1)$ (left) and $(1/2, 1)$ (right)

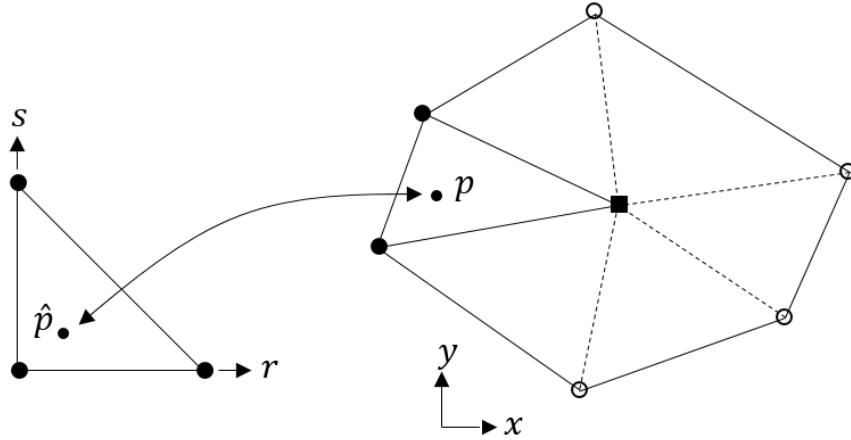


Figure 1.20: Mapping a point on the reference triangle onto a sub-triangle of an arbitrary polygon.

ber of topological shapes and explicit quadrature rules cannot be defined. Because of this, the development of efficient quadrature rules for arbitrary polytopes is an ongoing field of research [31, 32, 33].

At this time, we were only interested in the accuracy and not the efficiency of the integration of the elementary matrices. Therefore, we simply choose to use a simple triangulation-based scheme. The global polygonal element K with N_K vertices is sub-divided into N_K separate triangles. Each of these triangles is formed from two adjacent vertices and the polygon's centroid, \vec{c} . For convex and degenerate (not concave) polygons the centroid can be the average of the vertex coordinates, which is simply given by,

$$\vec{c} = \frac{1}{N_K} \sum_{i=1}^{N_K} \vec{x}_i. \quad (1.62)$$

Then for each sub-triangle, a quadrature rule with N_q nodes is employed (we do not vary the number of nodes between sub-triangles). This quadrature rule is specified in the reference space of $\{r, s\}$ on the unit triangle with vertices of $(0,0)$, $(1,0)$, and

$(0,1)$. We have chosen a symmetric reference quadrature set that is well documented in the literature [27]. We denote this reference quadrature rule by $\{\hat{x}_q, \hat{w}_q\}_{q=1}^{N_q}$, where the symbol $\hat{\cdot}$ denotes any quantity that lives in the reference space. We note that the sum of the reference weights equals the reference triangle area of $1/2$. This reference quadrature is mapped into the global space of the sub-triangle by an affine transformation. The mapping of a point from the reference space, $\hat{\mathbf{p}}$, to its corresponding point in global space, \mathbf{p} , is done with,

$$\mathbf{p} = \mathbf{x}_0 + J\hat{\mathbf{p}}, \quad (1.63)$$

where \mathbf{x}_0 is the global position of one of the sub-triangle vertices and J is the Jacobian matrix of the transformation. This mapping is presented graphically in Figure 1.20. If the global positions of the sub-triangle vertices are given by \vec{x}_0 , \vec{x}_1 , and \vec{x}_2 , then the Jacobian is given by the following,

$$\boxed{\begin{bmatrix} x_1 - x_0 & y_1 - y_0 \\ x_2 - x_0 & y_2 - y_0 \end{bmatrix}}. \quad (1.64)$$

The Jacobian matrix can also be used to map the gradients between the reference and global spaces. The gradient of the reference space can be computed in terms of the global space by,

$$\nabla_{\hat{x}} = \boxed{J}\nabla_x, \quad (1.65)$$

and the gradient of the global space can be computed in terms of the reference space by,

$$\nabla_x = J^{-1}\nabla_{\hat{x}} \quad (1.66)$$

With the positions of the nodes mapped to the global space, that just leaves the weights. The value of the global weight q on sub-triangle i within polygon K (given by $w_{i,q}^K$) is mapped from the corresponding reference weight, \hat{w}_q , by

$$w_{i,q}^K = \hat{w}_q |J_i|. \quad (1.67)$$

In Eq. (1.67), $|J_i|$ is the determinant of the Jacobian matrix corresponding the transformation to sub-triangle i , and it is equal to 2 times the area of the sub-triangle i . This means that the determinant acts to normalize the weights so that their sum is equal to the sub-triangle's area. Therefore, summing all the weights of all of the sub-triangles will equal the total area of the polygon K .

To this point, we have provided the means to generate the quadrature nodes and weights within the global space of a polygon K . Next, the values and gradients of the basis functions at these quadrature nodes can be calculated by the procedures outlined in Sections 1.1 and 1.2. Then, the function f can be integrated over the polygon K by the double sum,

$$\int_K f = \sum_{i=1}^{N_K} \sum_{q=1}^{N_q} w_{i,q}^K f(\vec{x}_{i,q}), \quad (1.68)$$

where $w_{i,q}^K$ and $\vec{x}_{i,q}$ correspond to the quadrature weights and global positions for node q within sub-triangle i , respectively. In this case, the function f can either be some scalar quantity or an elementary matrix. Thus, the elementary matrices needed for the DGFEM formulation of the transport equation can be computed in a logical manner for any arbitrary polygon. In a similar manner, the integral of f over the entire mesh, \mathbb{T}_h , is simply the sum of integrals over all elements. This integration of f over the entire domain is simply given by

$$\int_{\mathbb{T}_h} f = \sum_{K \in \mathbb{T}_h} \left(\sum_{i=1}^{N_K} \sum_{q=1}^{N_q} w_{i,q}^K f(\vec{x}_{i,q}) \right), \quad (1.69)$$

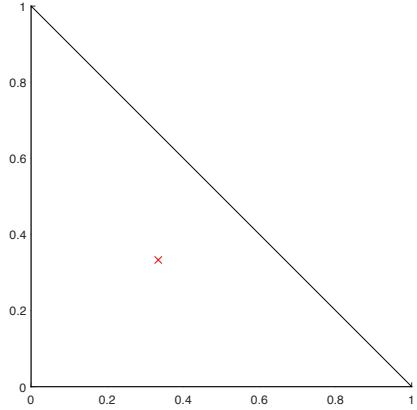
which is clearly just an element-wise sum of Eq. (1.68).

We conclude this section by providing some visual examples of quadrature sets for polygonal elements. Figure 1.21 gives quadrature sets on the reference triangle for orders 1-6. We can see that our reference quadrature is symmetric about any of the three vertices, though we note that true isoparametric symmetry is only obtained with equilateral triangles. Then Figures 1.22 and 1.23 provide examples of the mapping of the reference quadrature into the global space for a regular pentagon and hexagon, respectively.

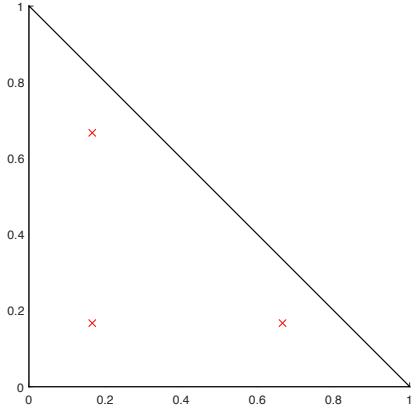
1.4 Linear Basis Functions on 3D Polyhedra

We have defined linearly-complete and quadratically-complete 2D polygonal basis functions for use in FEM analysis of the DGFEM transport equation. Now, we present an efficient coordinate system for arbitrary 3D polyhedra that is linearly-complete. At the time of this work and to the best of our knowledge, no analogous methodology to convert linear coordinates on 3D polyhedra to their serendipity basis exists. Therefore, we will utilize only a single linearly-complete 3D coordinate system for some of the analysis to be performed in Chapter ??, but we include it here for completeness with the 2D coordinates.

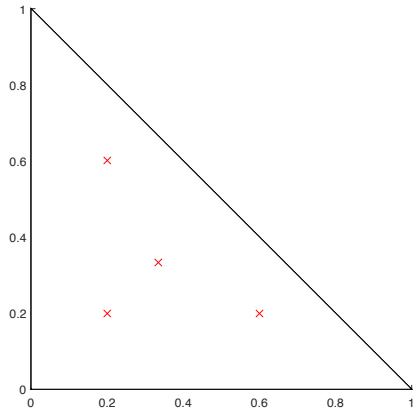
For this work, we will utilize the 3D version of the Piecewise Linear (PWL) coordinates that is suitable for x-y-z geometries [7]. From Table 1.1, we can see some of the properties of the different 2D polygonal coordinates. The PWL functions are the only coordinates with a 3D analogue that allow for direct, analytical integration of the elementary matrices. This means that no spatial quadrature sets are



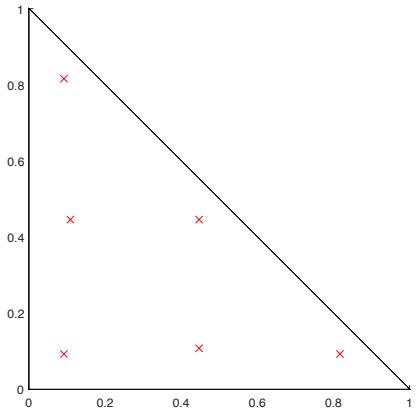
(a) Order 1



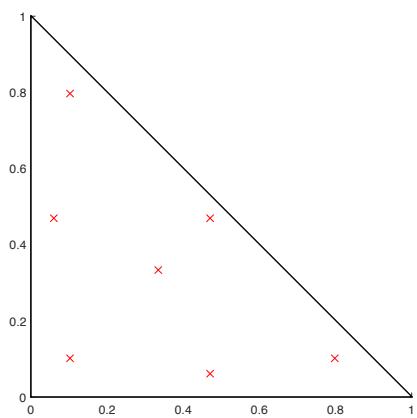
(b) Order 2



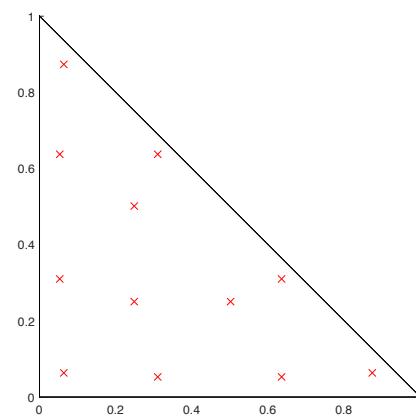
(c) Order 3



(d) Order 4



(e) Order 5



(f) Order 6

Figure 1.21: Quadrature sets on the reference triangle of varying order.

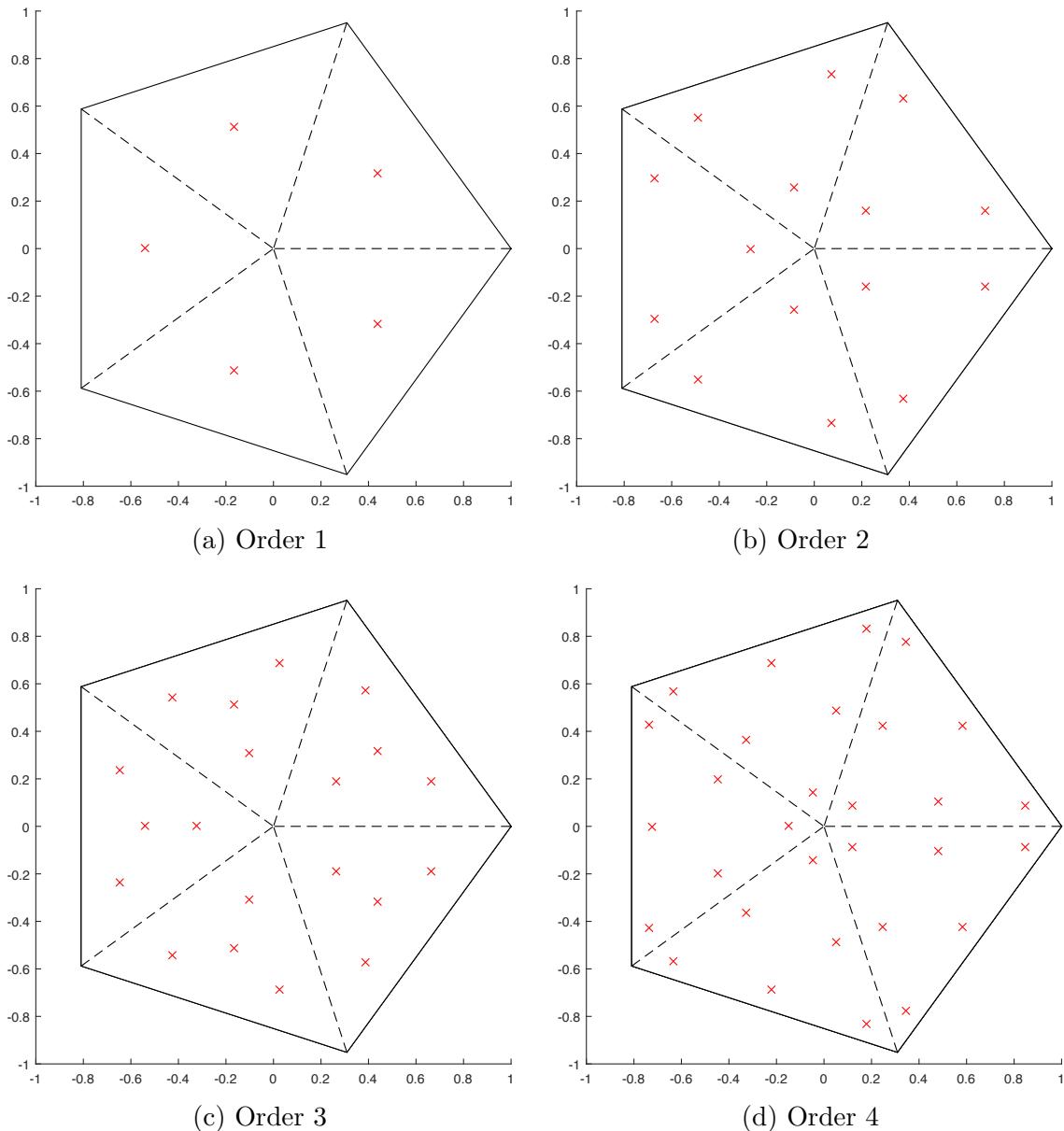


Figure 1.22: Examples of spatial quadrature sets of varying order on a regular pentagon.

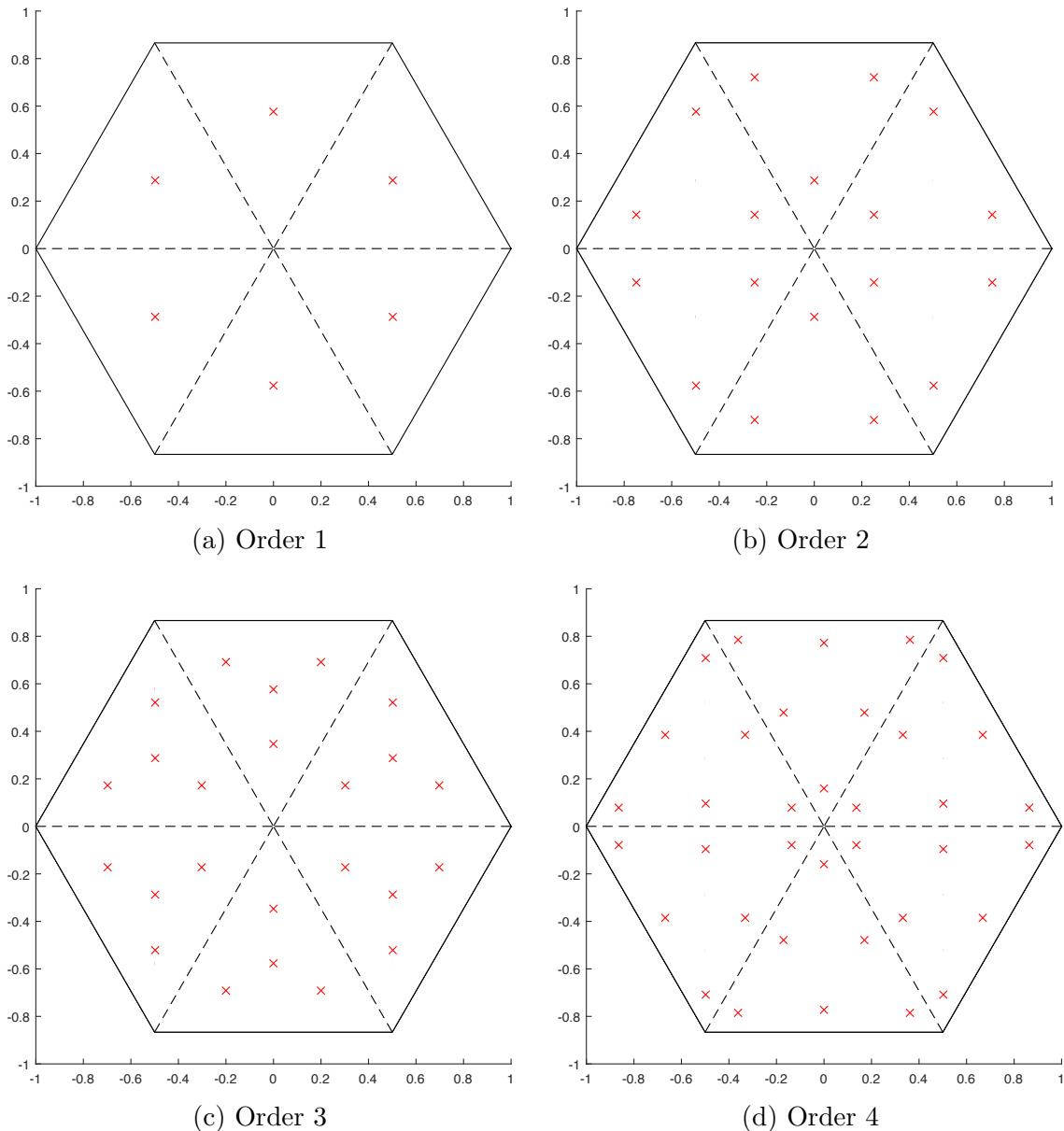


Figure 1.23: Examples of spatial quadrature sets of varying order on a regular hexagon.

required, though an analogous procedure to Section 1.3 could be employed using sub-tetrahedra instead of sub-triangles. In Appendix ??, we detail how these 3D PWL coordinates can be analytically integrated using the reference tetrahedron and affine mapping. The 3D PWL coordinates also allow for extremely-distorted concave polyhedra, though we will not analyze those in this work.

The 3D PWL coordinates have an analogous form to their 2D version from Eq. (1.20). In fact, the 2D PWL coordinates are identical to their 3D version along a polyhedral face (the 3D face centroid acts like the 2D cell centroid). If we define N_K as the number of vertices for cell K and N_f as the number of vertices composing face f , then the cell and face centroids for a strongly-convex polyhedra can be given by

$$\vec{r}_c = \frac{1}{N_K} \sum_{i=1}^{N_K} \vec{x}_i, \quad (1.70)$$

and

$$\vec{r}_f = \frac{1}{N_f} \sum_{i=1}^{N_f} \vec{x}_i, \quad (1.71)$$

respectively. We see that these centroids have the simple definition of the average positions of the cell and face vertices. Using these centroid definitions, the functional form for the 3D PWL coordinates is given by:

$$b_j(x, y, z) = t_j(x, y, z) + \sum_{f=1}^{F_j} \beta_f^j t_f(x, y, z) + \alpha_j^K t_c(x, y, z). \quad (1.72)$$

In Eq. (1.72), t_j is the standard 3D linear function with unity at vertex j that linearly decreases to zero to the cell center, the face center for each face that includes vertex j , and each vertex that shares an edge with vertex j . t_c is the 3D cell “tent” function located at \vec{r}_c which is unity at the cell center and linearly decreases to zero to each

cell vertex and face center. t_f is the face "tent" function for face f located at \vec{r}_f which is unity at the face center and linearly decreases to zero at each vertex on that face and the cell center. $\beta_{f,j}$ is the weight parameter for face f touching cell vertex j , and F_j is the number of faces touching vertex j . Like the previous work defining the PWLD method [7], we also choose to assume the cell and face weighting parameters are

$$\alpha_{K,j} = \frac{1}{N_K}, \quad (1.73)$$

and

$$\beta_{f,j} = \frac{1}{N_f}, \quad (1.74)$$

respectively, which leads to constant values of α and β for each cell and face, respectively. This assumption of the cell weight functions remains from the 2D PWL form.

1.5 Numerical Results

Now that we have presented several linear polygonal finite element basis sets along with the methodology to convert them to quadratic serendipity-like basis, we present several numerical problems to demonstrate our methodology. First, we demonstrate that all of the 2D linear and quadratic serendipity basis functions can capture the correct transport solution in the thick diffusion limit. Then, we demonstrate that the presented linear basis functions can capture an exactly-linear transport solution in Section 1.5.2. We follow that by demonstrating that the quadratic serendipity basis functions can capture an exactly-quadratic transport solution in Section 1.5.3. Next, we present some convergence properties of the basis sets using the method of man-

ufactured solutions (MMS) in Section 1.5.4. Then in Section 1.5.5, we demonstrate how the solution regularity can limit the convergence of our numerical transport solutions to the $H^{1/2}$ and $H^{3/2}$ Hilbert spaces. We conclude with a searchlight problem and observe how the basis sets react with adaptive mesh refinement (AMR) to mitigate numerical dispersion through a vacuum in Section 1.5.6.

1.5.1 Transport Solutions in the Thick Diffusive Limit

We present our first numerical example by demonstrating that the various 2D polygonal finite element basis functions provided in Chapter 1 satisfy the thick diffusion limit. We investigate the transport problem with isotropic scattering and an isotropic distributed source given by the following:

$$\vec{\Omega} \cdot \vec{\nabla} \Psi + \sigma_t \Psi = \frac{\sigma_s}{4\pi} \Phi + \frac{Q_0}{4\pi}. \quad (1.75)$$

As the transport problem becomes more optically thick, the total mean free paths of the neutrons increases. In the thick diffusion limit, the domain mean free path approaches infinity. If we fix the physical dimensions of the problem to some finite value, then we can scale the cross sections and the source term to reflect the properties of the thick diffusion limit. In the thick diffusion limit the total and scattering cross sections tend to infinity while the absorption cross section and the source term tend to zero. If we introduce a scaling parameter, ϵ , then we can write the scaled terms as,

$$\begin{aligned}
\sigma_t &\rightarrow \frac{\sigma_t}{\epsilon} \\
\sigma_a &\rightarrow \epsilon\sigma_t \\
\sigma_s &\rightarrow \left(\frac{1}{\epsilon} - \epsilon\right)\sigma_t \\
\frac{Q_0}{4\pi} &\rightarrow \epsilon\frac{Q_0}{4\pi}
\end{aligned} \tag{1.76}$$

Inserting these scaled cross sections and source term into Eq. (1.75) leads to the following scaled transport equation:

$$\vec{\Omega} \cdot \vec{\nabla}\Psi + \frac{\sigma_t}{\epsilon}\Psi = \sigma_t \left(\frac{1}{\epsilon} - \epsilon\right) \frac{\Phi}{4\pi} + \epsilon \frac{Q_0}{4\pi}. \tag{1.77}$$

We can also use the scaled terms of Eq. (1.76) to give the corresponding scaled diffusion equation. If we take the 0th and 1st moments of Eq. (1.77) and assume that the P1 terms obey Fick's Law, then the scaled diffusion equation is

$$\vec{\nabla} \cdot \frac{1}{3\sigma_t} \vec{\nabla}\Phi + \epsilon\sigma_t\Phi = \epsilon Q_0. \tag{1.78}$$

One can immediately see that Eq. (1.78) does not truly scale because there is an ϵ for each term. This is the desired behavior we want to see from the diffusion equation because, as $\epsilon \rightarrow 0$, the transport equation will converge to its diffusive limit and satisfy a diffusion equation.

For the sake of analysis, we seek to simplify Eqs. (1.77) and (1.78) by normalization. We choose to set σ_t and Q_0 to unity which gives the final transport and diffusion equations as

$$\vec{\Omega} \cdot \vec{\nabla}\Psi + \frac{1}{\epsilon}\Psi = \left(\frac{1}{\epsilon} - \epsilon\right) \frac{\Phi}{4\pi} + \frac{\epsilon}{4\pi}, \tag{1.79}$$

and

$$\frac{\epsilon}{3} \nabla^2 \Phi + \epsilon \Phi = \epsilon, \quad (1.80)$$

respectively.

From previous work [34], it is already known that linear interpolants with properties corresponding to barycentric coordinates satisfy the thick diffusion limit. However, it needs to be demonstrated that the quadratic serendipity extensions will also satisfy the limit. We will demonstrate this both qualitatively and quantitatively. The transport and diffusion equations to be solved are Eqs. (1.79) and (1.80), respectively. Vacuum boundary conditions are applied for the transport equations and homogeneous dirichlet conditions are applied for the diffusion equations. With this choice of boundary conditions, the transport and diffusion solutions will converge only as ϵ gets small. Specifically, they converge at a rate of $O(\epsilon)$ with an L_2 -norm. The diffusion equation is discretized using the same DGFEM functional space as the transport equations, and we leave these details until Section ??.

We first provide an example of diffusion solution in Figure 1.24 on a polygonal grid for both the linear and quadratic mean value basis functions. Next, Figures 1.25, 1.26, 1.27, and 1.28 provide the transport solutions with varying ϵ values using the Wachspress, PWL, mean value, and maximum entropy coordinates, respectively. We see that as we reduce ϵ from 10^{-1} to 10^{-5} , our transport solutions converge to our diffusion solutions. Finally, we show the convergence rates of the discretized transport, Φ_T , and diffusion, Φ_D , solutions under the L_2 -norm. As ϵ decreases, the L_2 -norm of the difference between the transport and diffusion solutions, $\|\Phi_T - \Phi_D\|_{L_2}$, should decrease as $O(\epsilon)$. We demonstrate this as true in Figures 1.29 and 1.30 for a Cartesian and polygonal mesh, respectively. For both meshes, all of the linear and quadratic basis functions show convergence rates of $O(\epsilon)$.

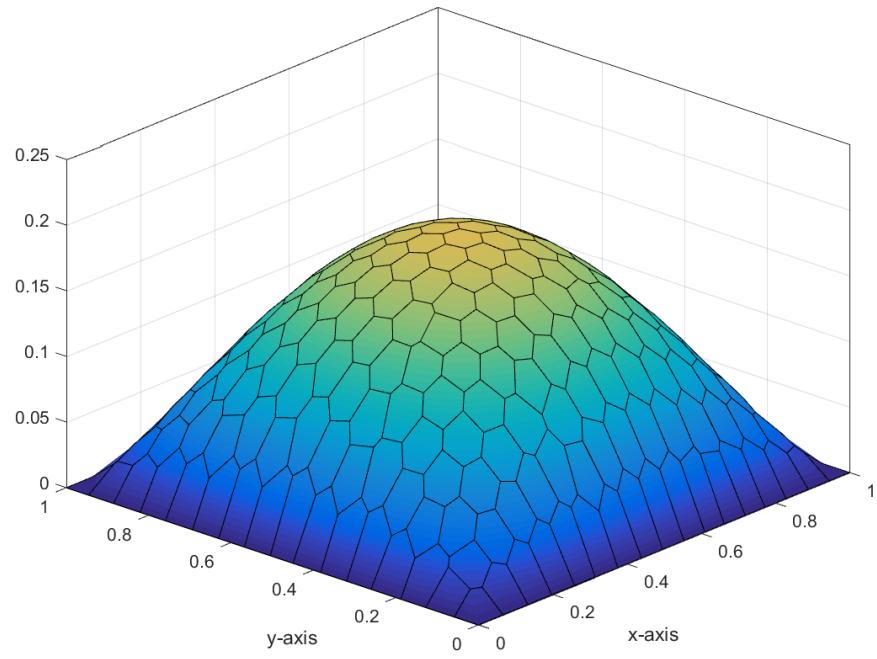
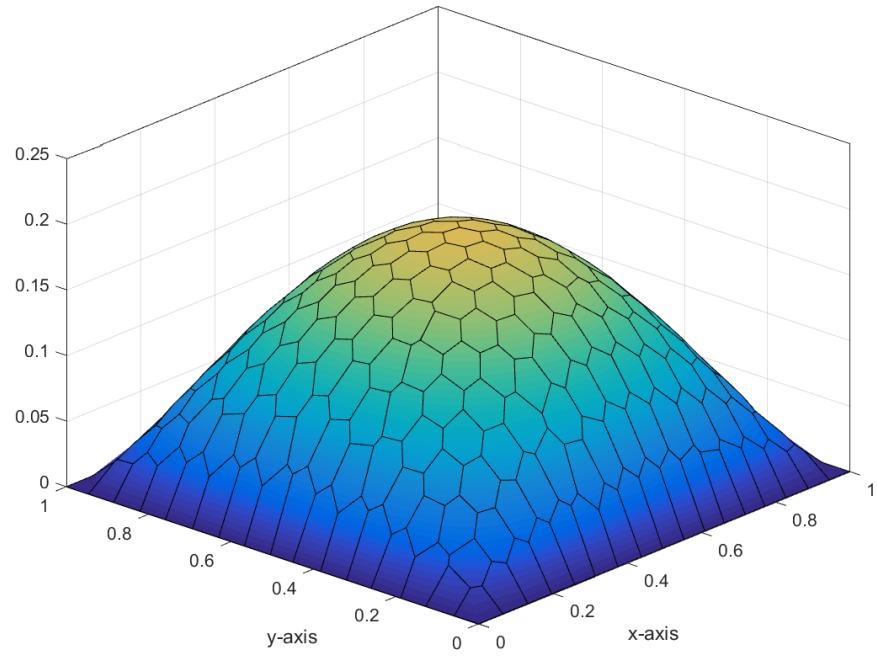


Figure 1.24: Diffusion solution representing the thick diffusion limit problem with linear (top) and quadratic (bottom) mean value basis functions.

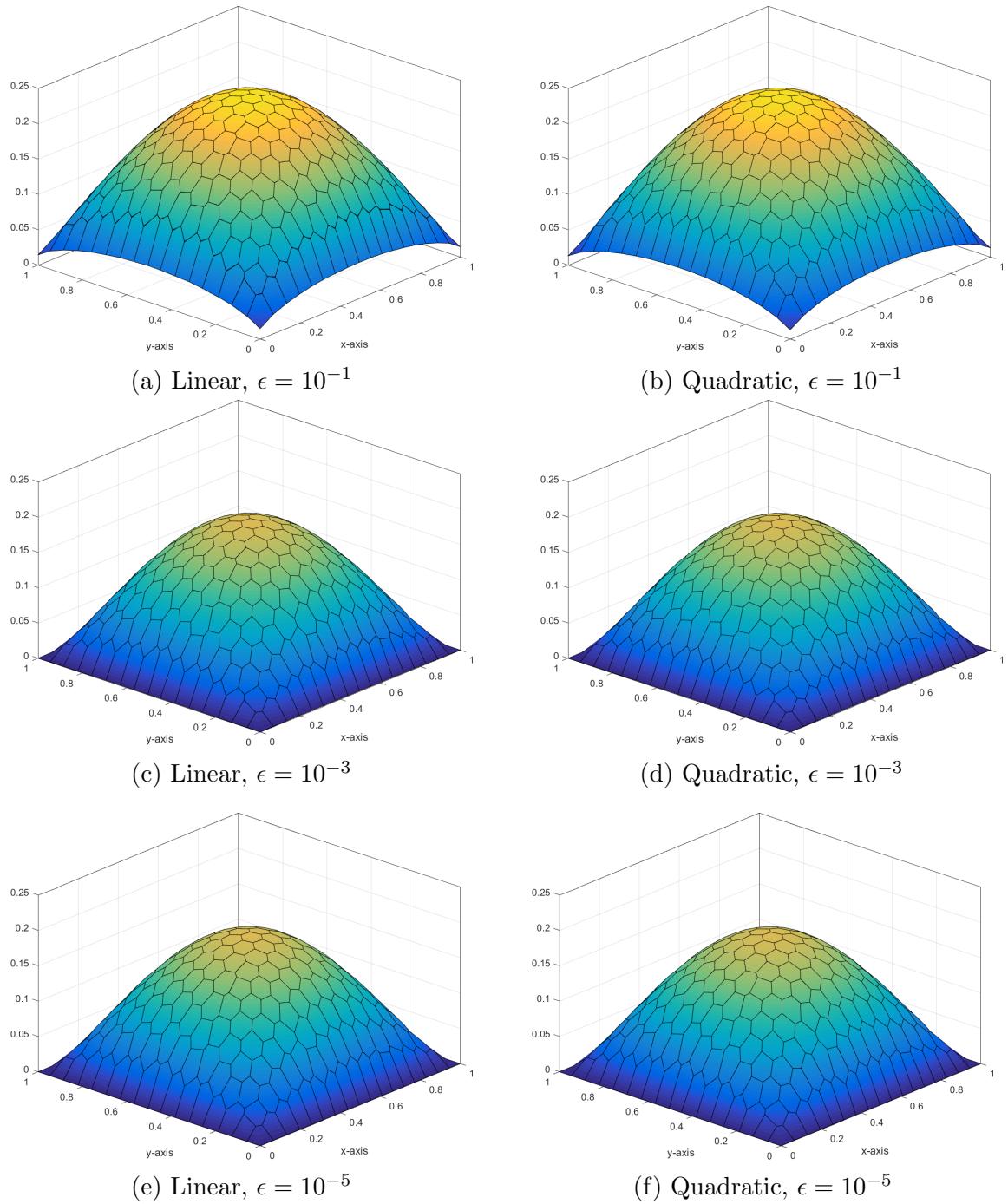


Figure 1.25: Transport solutions of the thick diffusion limit problem using the Wachspress basis functions for varying values of the scaling parameter, ϵ .

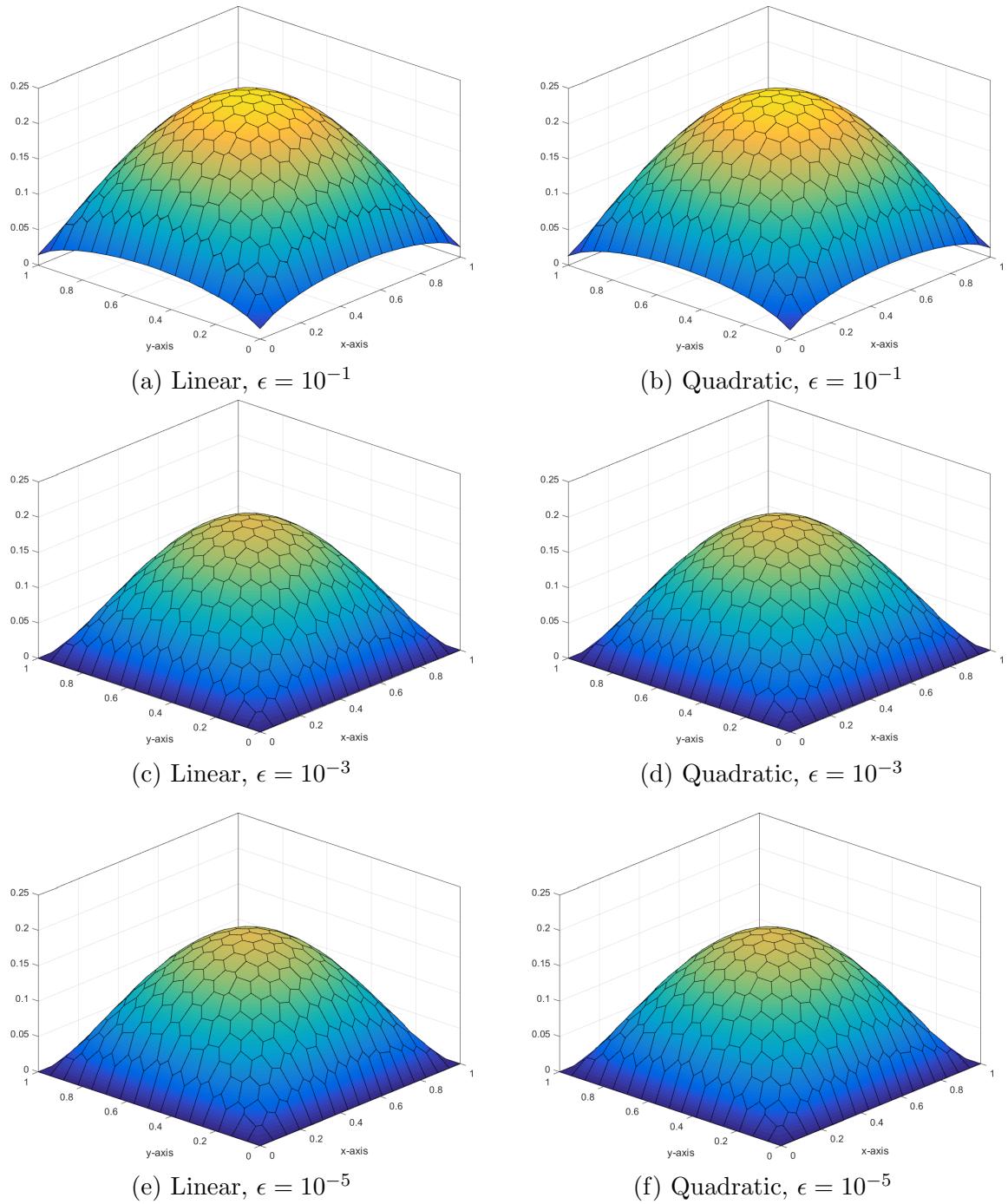


Figure 1.26: Transport solutions of the thick diffusion limit problem using the PWL basis functions for varying values of the scaling parameter, ϵ .

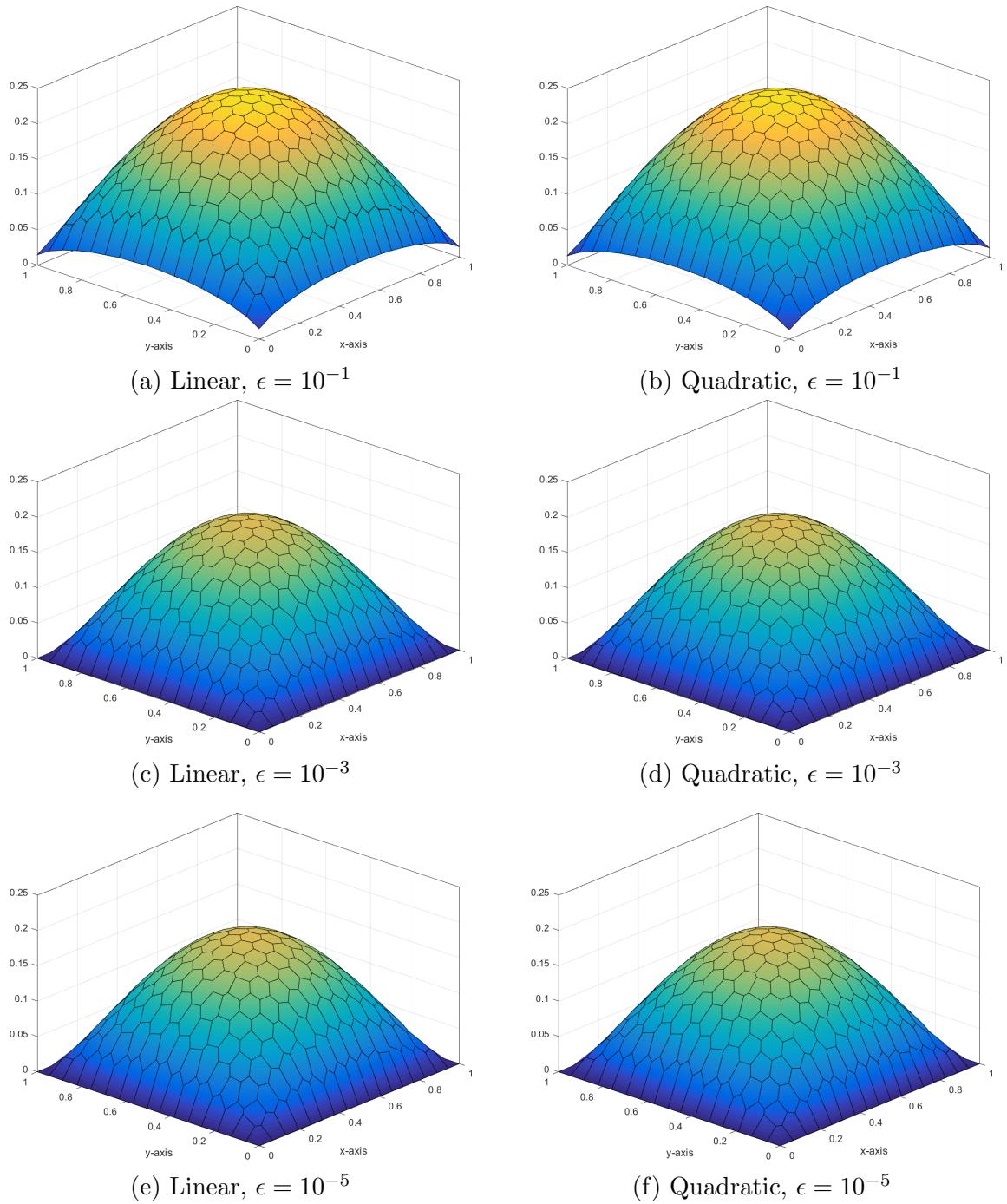


Figure 1.27: Transport solutions of the thick diffusion limit problem using the mean value basis functions for varying values of the scaling parameter, ϵ .

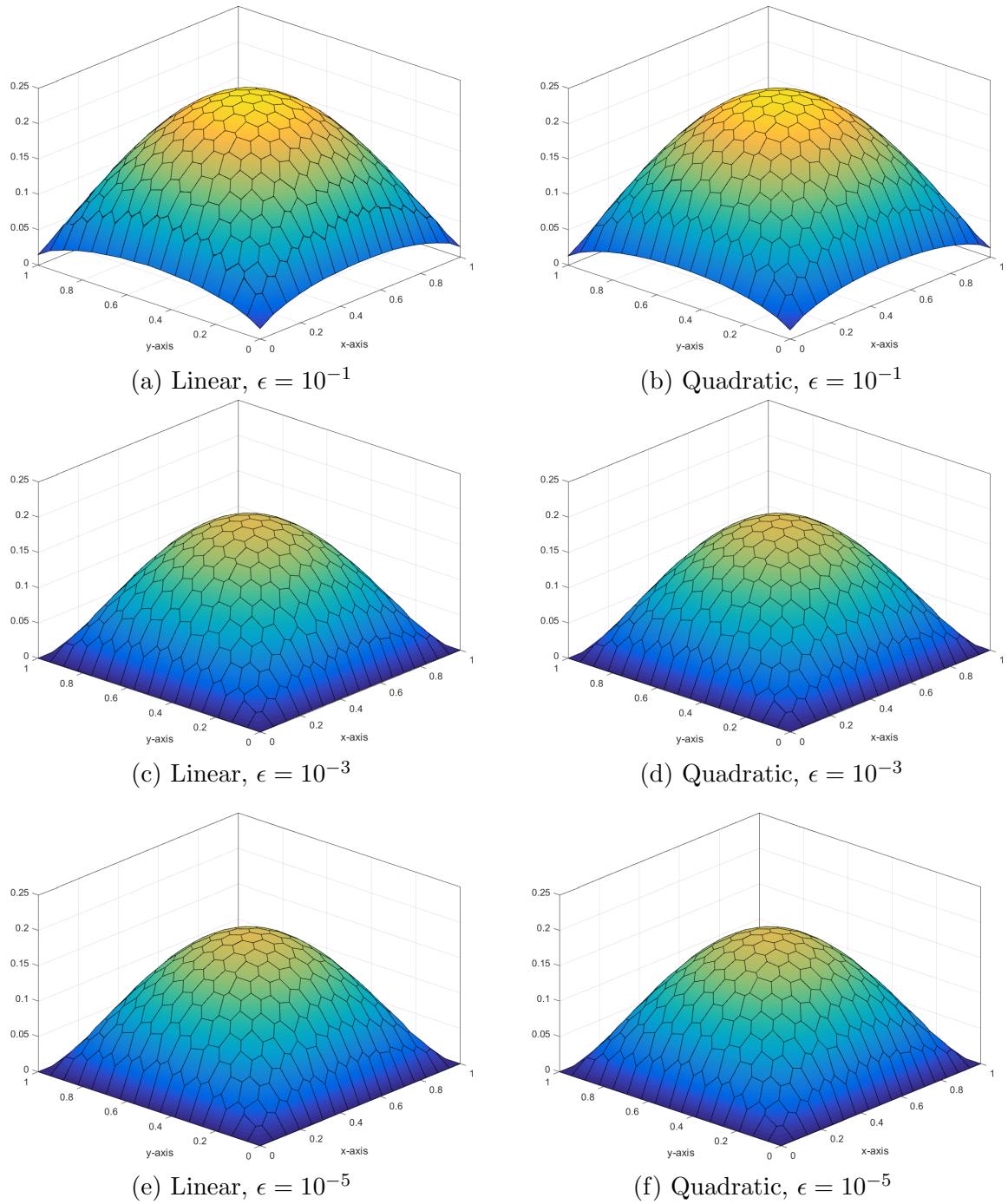


Figure 1.28: Transport solutions of the thick diffusion limit problem using the maximum entropy basis functions for varying values of the scaling parameter, ϵ .

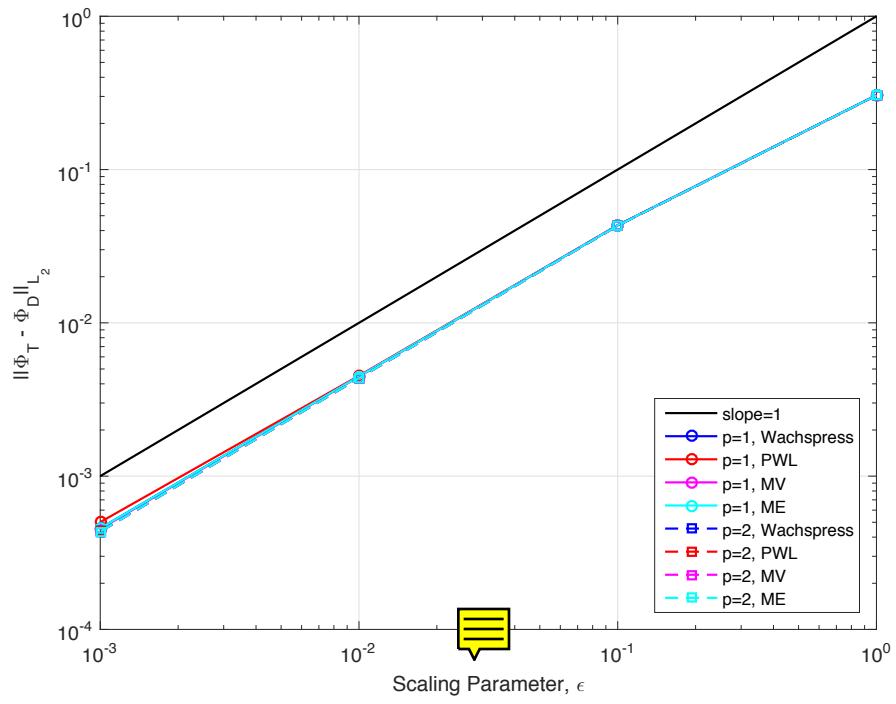


Figure 1.29: Convergence rates for the diffusion limit problem on a Cartesian mesh.

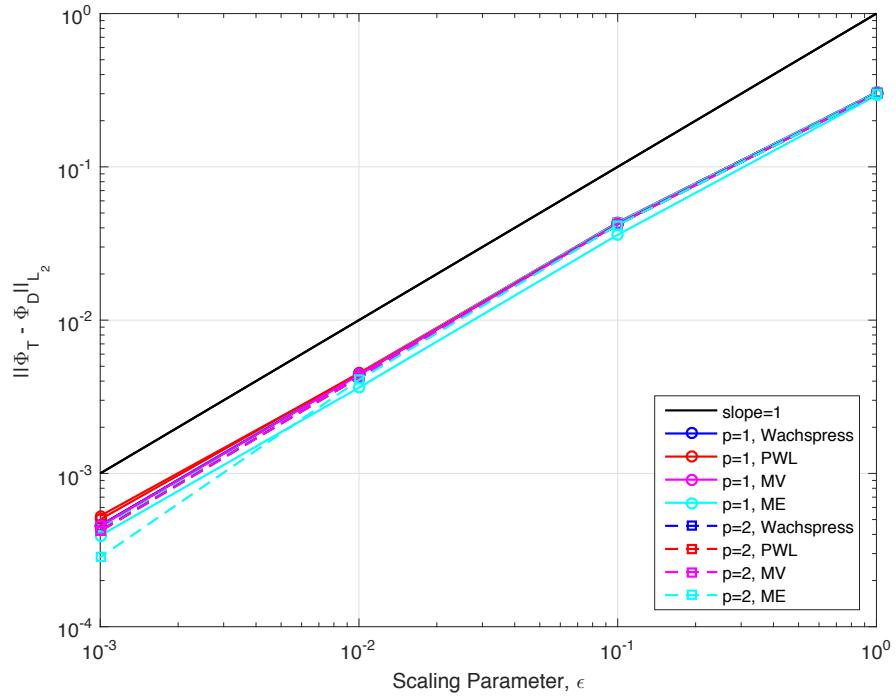


Figure 1.30: Convergence rates for the diffusion limit problem on a polygonal mesh.

1.5.2 Two-Dimensional Exactly-Linear Transport Solutions

Our next numerical example demonstrates that the linear and quadratic polygonal finite element basis functions capture an exactly-linear solution space. We will show this by the method of exact solutions (MES). Since the coordinate interpolation of the basis functions for the linear basis functions requires exact linear interpolation (Eq. (1.2)), then an exactly-linear solution space can be captured, even on highly distorted polygonal meshes. This also applies to the quadratic serendipity space since it is formed by the product-wise pairings of the linear basis functions. We build our exact solution by investigating the 2D, 1 energy group transport problem with no scattering and an angle-dependent distributed source,

$$\mu \frac{\partial \Psi}{\partial x} + \eta \frac{\partial \Psi}{\partial y} + \sigma_t \Psi = Q(x, y, \mu, \eta), \quad (1.81)$$

where the streaming term was separated into the corresponding two-dimensional terms. We chose to drop the scattering term for this example so that the error arising from iteratively converging our solution would have no impact.

We then define an angular flux solution that is linear in both space and angle along with the corresponding 0th moment scalar flux ($\Phi_{0,0} \rightarrow \Phi$) solution:

$$\begin{aligned} \Psi(x, y, \mu, \eta) &= ax + by + c\mu + d\eta + e \\ \Phi(x, y) &= 2\pi(ax + by + e) \end{aligned} . \quad (1.82)$$

One can immediately notice that our 0th moment solution is not dependent on angle. We arrive at this solution by enforcing our 2D angular quadrature set to have the following properties:

$$\sum_q w_q = 2\pi \quad \text{and} \quad \sum_q w_q \begin{bmatrix} \mu_q \\ \eta_q \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \quad (1.83)$$

The sum of the quadrature weights is handled by simply re-normalizing those that are generated in Section ?? to 2π .

Our boundary conditions for all inflow boundaries are then uniquely determined by the angular flux solution of Eq. (1.82). Inserting the angular flux solution of Eq. (1.82) into Eq. (1.81), we obtain the distributed source that will produce our exactly-linear solution space:

$$Q(x, y, \mu, \eta) = a\mu + b\eta + \sigma_t(c\mu + d\eta) + \sigma_t(ax + by + e). \quad (1.84)$$

It is noted that the angular dependence of the source can be removed (which can ease the code development burden) if one sets

$$\begin{aligned} a &= -c\sigma_t, \\ b &= -d\sigma_t. \end{aligned} \quad (1.85)$$

For this example, we test the various 2D polygonal finite element basis functions on six different mesh types. These mesh types include triangular, quadrilateral, and polygonal meshes:

1. Orthogonal Cartesian mesh formed by the intersection of 11 equally-spaced vertices in both the x and y dimensions. This forms a 10x10 array of quadrilateral mesh cells.
2. Ordered-triangular mesh formed by the bisection of the previous orthogonal Cartesian mesh (forming 200 triangles all of the same size/shape).

3. Quadrilateral Shestakov grid formed by the randomization of vertices based on a skewness parameter [35, 36]. With a certain range of this skewness parameter, highly distorted meshes can be generated.
4. Sinusoidal polygonal grid that is generated by the transformation of a uniform orthogonal grid based on a sinusoid functional. The transformed vertices are then converted into a polygonal grid by computing a bounded Voronoi diagram.
5. Kershaw's quadrilateral z-mesh [37]. This mesh is formed by taking an orthogonal quadrilateral grid and displacing certain interior vertices only in the y dimension.
6. A polygonal variant of the quadrilateral z-mesh. The polygonal grid is formed in a similar manner to the sinusoidal polygonal mesh with a Voronoi diagram.

We also wish that both the angular flux solution as well as the 0th moment solution are strictly positive everywhere. Therefore, we set the function parameters in Eq. (1.82) to $\sigma_t = a = c = d = e = 1.0$ and $b = 1.5$.  gave the solution the 40% tilt in space ($a \neq b$) so that it would not align with the triangular mesh. Using an S8 LS quadrature set, we ran all combinations of the polygonal basis functions and the mesh types. The linear solutions for the Wachspress, PWL, mean value, and linear maximum entropy are presented in Figures 1.31, 1.32, 1.33, 1.34, respectively. We can see that for all the polygonal basis functions, an exactly-linear solution is captured as shown by the unbroken nature of the contour lines. This even holds on the highly distorted quadrilateral Shestakov mesh.

1.5.3 Two-Dimensional Exactly-Quadratic Transport Solutions

In Section 1.2, we stated that the quadratic serendipity basis can only interpolate the $\{1, x, y, x^2, xy, y^2\}$ span of functions. Unlike the Q9 elements, the quadratic

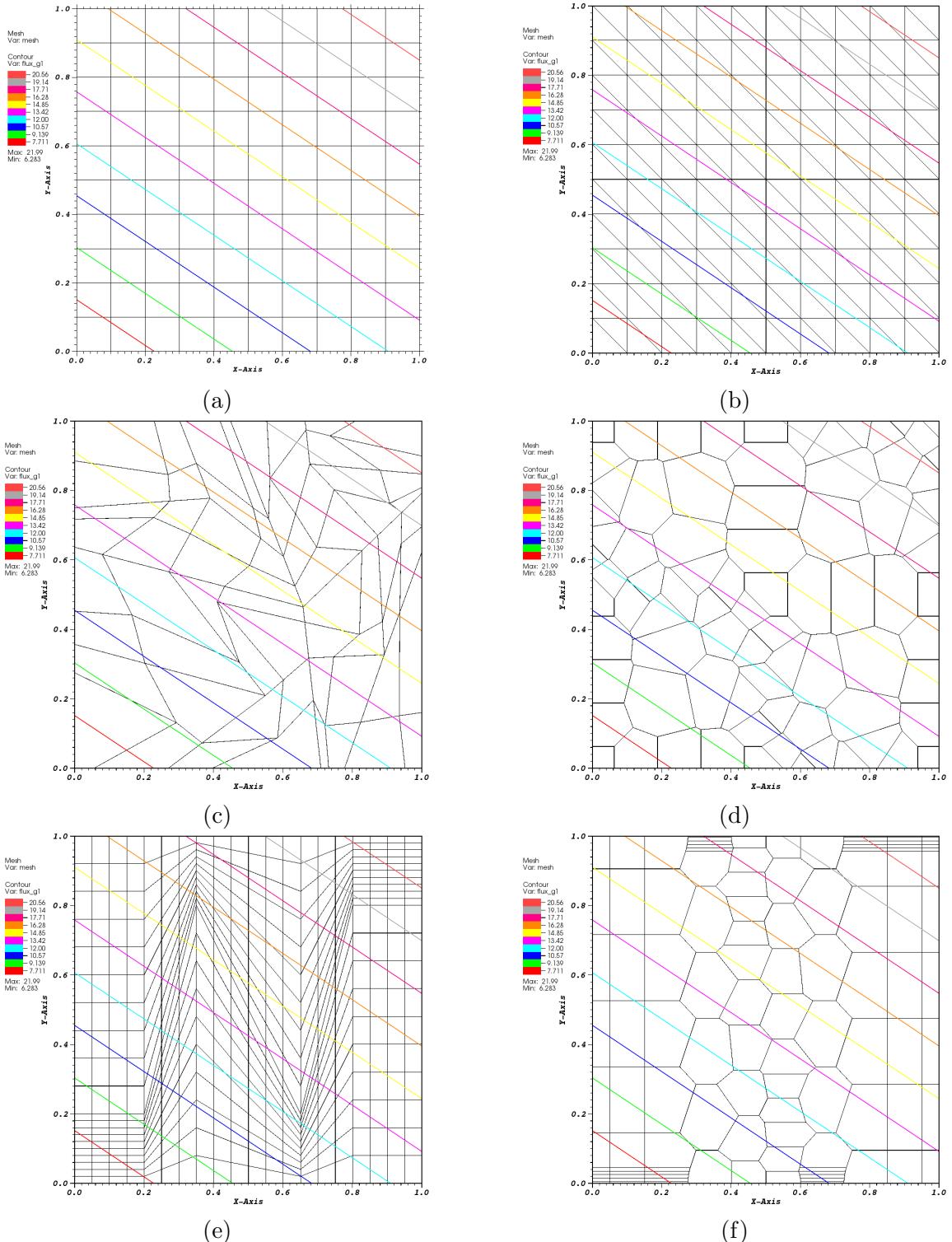


Figure 1.31: Contour plots of the exactly-linear solution with the Wachspress basis functions on (a) cartesian mesh, (b) ordered-triangular mesh, (c) quadrilateral Shestakov mesh, (d) sinusoidal polygonal mesh, (e) quadrilateral z-mesh, and (f) polygonal z-mesh.

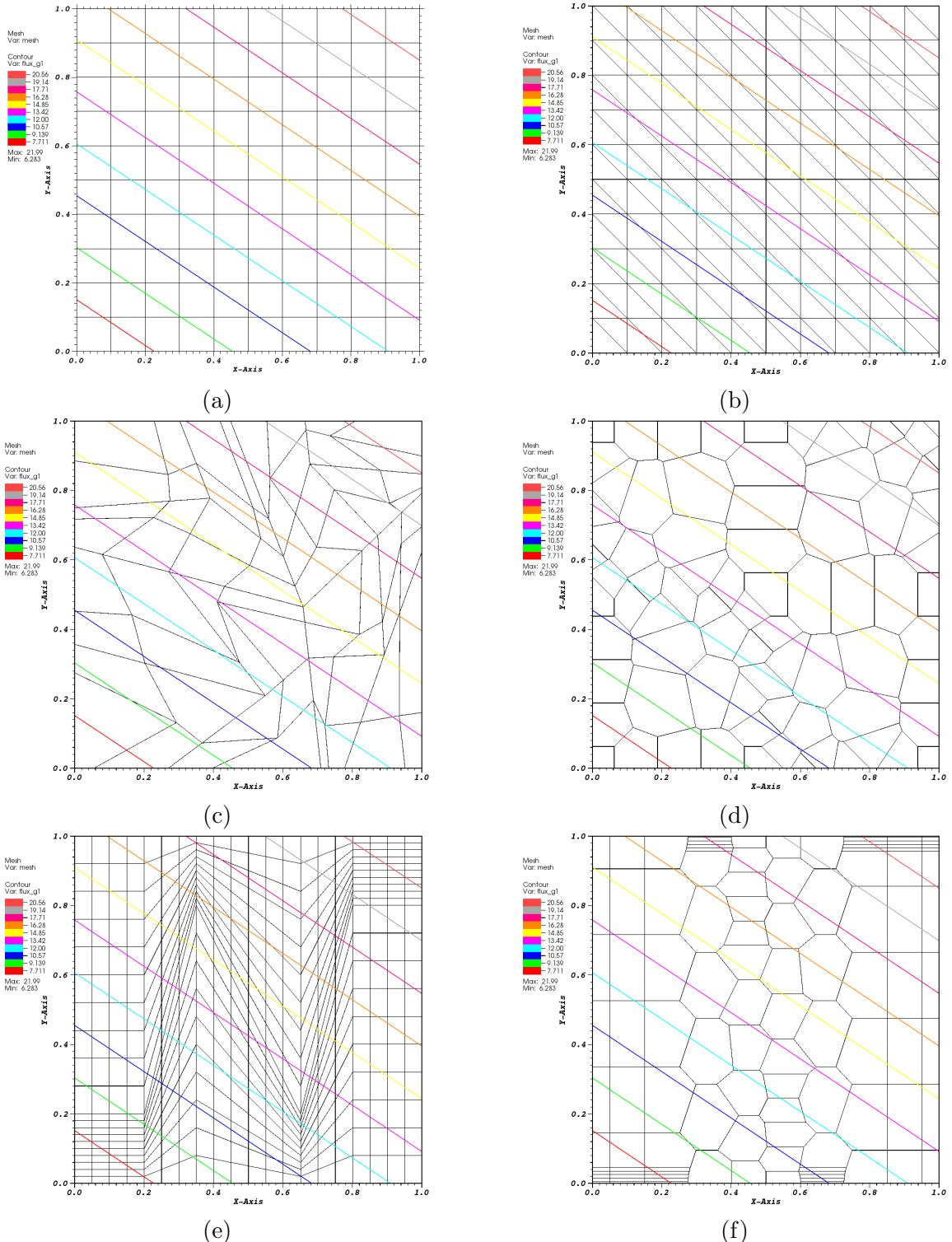


Figure 1.32: Contour plots of the exactly-linear solution with the PWL basis functions on (a) cartesian mesh, (b) ordered-triangular mesh, (c) quadrilateral Shestakov mesh, (d) sinusoidal polygonal mesh, (e) quadrilateral z-mesh, and (f) polygonal z-mesh.

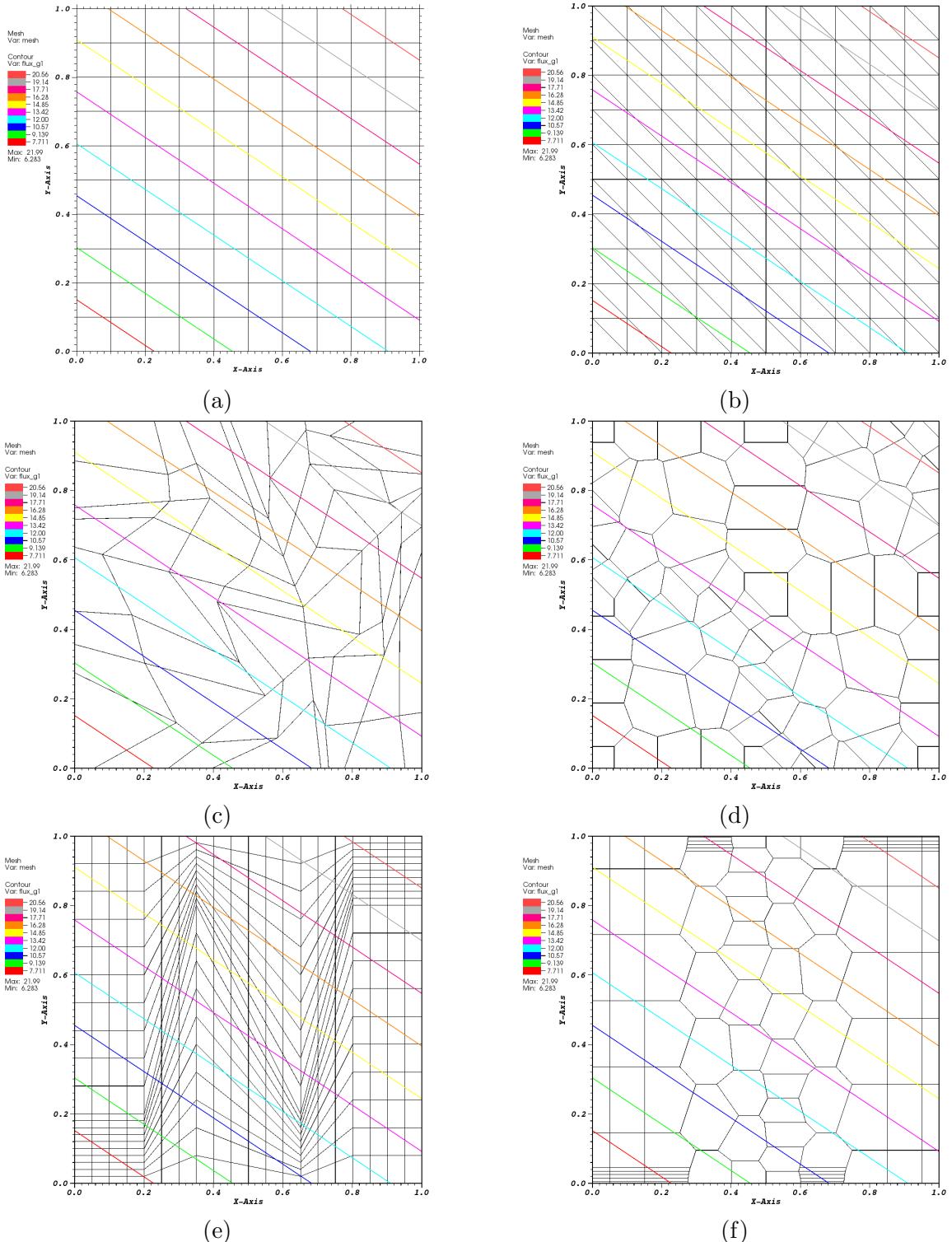


Figure 1.33: Contour plots of the exactly-linear solution with the mean value basis functions on (a) cartesian mesh, (b) ordered-triangular mesh, (c) quadrilateral Shestakov mesh, (d) sinusoidal polygonal mesh, (e) quadrilateral z-mesh, and (f) polygonal z-mesh.

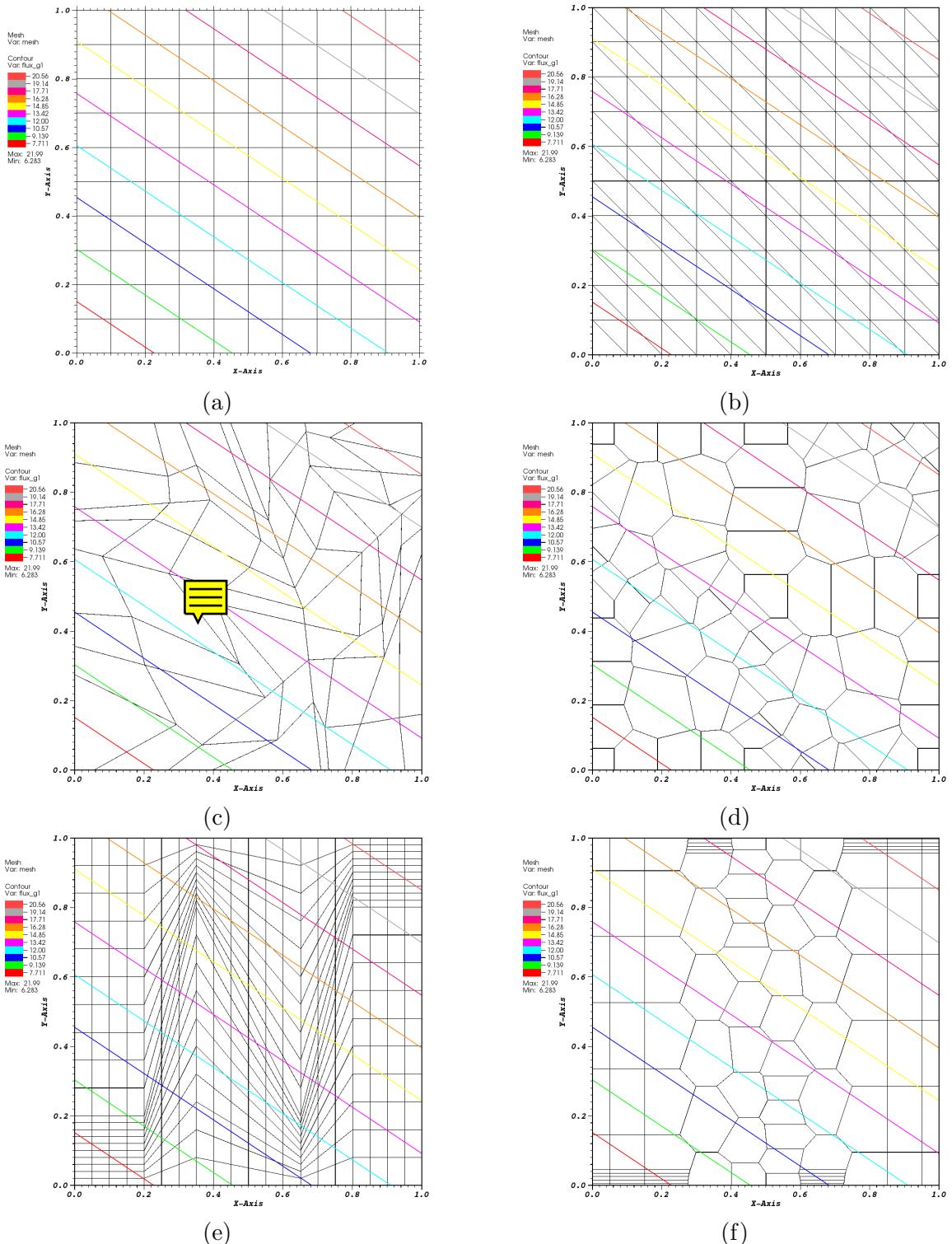


Figure 1.34: Contour plots of the exactly-linear solution with the linear maximum entropy basis functions on (a) cartesian mesh, (b) ordered-triangular mesh, (c) quadrilateral Shestakov mesh, (d) sinusoidal polygonal mesh, (e) quadrilateral z-mesh, and (f) polygonal z-mesh.

serendipity basis cannot interpolate the $\{x^2y, xy^2x^2y^2\}$ span of functions. In this example we show the limit of the interpolatory properties of the quadratic serendipity space by again using MES. We will test two exact solution spaces. The first spans the functions that are captured by the quadratic serendipity space, and the second spans the additional functions captured by the Q9 elements (but is not captured by the quadratic serendipity space).

The first problem interpolates the $\{1, x, y, x^2, xy, y^2\}$ span of functions, which we denote with  following general and exactly-quadratic solution, $\{\Psi_q, \Phi_q\}$:

$$\begin{aligned}\Psi_q(x, y, \mu, \eta) &= a + bx + cy + dxy + ex^2 + fy^2 \\ \Phi_q(x, y) &= 2\pi(a + bx + cy + dxy + ex^2 + fy^2)\end{aligned}. \quad (1.86)$$

Inserting the angular flux solution of Eq. (1.86) into the previously-defined Eq. (1.81) yields the following right-hand-side distributed source,

$$\begin{aligned}Q_q(x, y, \mu, \eta) &= (b + dy + 2ex)\mu + (c + dx + 2fy)\eta \\ &\quad + \sigma_t(a + bx + cy + dxy + ex^2 + fy^2)\end{aligned}. \quad (1.87)$$

We clearly see that any combination of positive or negative (non-zero) values for the coefficients $a - f$ will span the quadratic serendipity space. The second problem contains terms up to x^2y^2 . This higher-order functional form has a solution, $\{\Psi_{x2y2}, \Phi_{x2y2}\}$, given by the following

$$\begin{aligned}\Psi_{x2y2}(x, y, \mu, \eta) &= x(L_x - x)y(L_y - y) \\ \Phi_{x2y2}(x, y) &= 2\pi x(L_x - x)y(L_y - y)\end{aligned}, \quad (1.88)$$

where L_x and L_y are the dimensions of the domain in x and y . Again, inserting this solution into Eq. (1.86) yields the following right-hand-side distributed source,

$$\begin{aligned}
Q_{x2y2}(x, y, \mu, \eta) = & \left[y(L_x - x)(L_y - y) - xy(L_y - y) \right] \mu \\
& + \left[x(L_x - x)(L_y - y) - xy(L_x - x) \right] \eta . \quad (1.89) \\
& + \sigma_t x(L_x - x)y(L_y - y)
\end{aligned}$$

For both problems, L_x and L_y are set to 1.0 to form the unit square. For the exactly-quadratic solutions, $\{\Psi_q, \Phi_q\}$, we set the function parameters in Eqs. (1.86) and (1.87) all to be 1.0: $\sigma_t = a = b = c = d = e = f = 1.0$. The meshes used for both problems are the same used for the exactly-linear problem. We give an example of the numerical solutions for the exactly-quadratic solution in Figure 1.35 using the quadratic serendipity maximum entropy coordinates for all the different meshes. Next, we give the spatial distributions of the error for the exactly-quadratic problem in Figures 1.36, 1.37, 1.38, and 1.39 for the Wachspress, PWL, MV, and ME basis functions, respectively. The magnitudes of these spatial distributions are around $10^{-14} - 10^{-15}$ showing that the quadratic serendipity basis functions can capture exactly-quadratic solutions to machine precision. We next present results for the quadratic solution that includes the x^2y^2 term. From the theory for the quadratic serendipity functional space, we should not be able to capture solutions of higher order than x^2 and y^2 . Figure 1.40 gives an example of this solution using the quadratic serendipity maximum entropy coordinates for all the different meshes. Table 1.2 then gives the L_2 -norm of the MMS error for the different basis functions and mesh types. We clearly see errors on the order of $O(10^{-6})$ to $O(10^{-4})$ which means that we do not capture the functional space as predicted.

1.5.4 Convergence Rate Analysis by the Method of Manufactured Solutions

The next numerical example we investigate involves calculating the convergence rate of the solution error via the method of manufactured solutions (MMS). Like

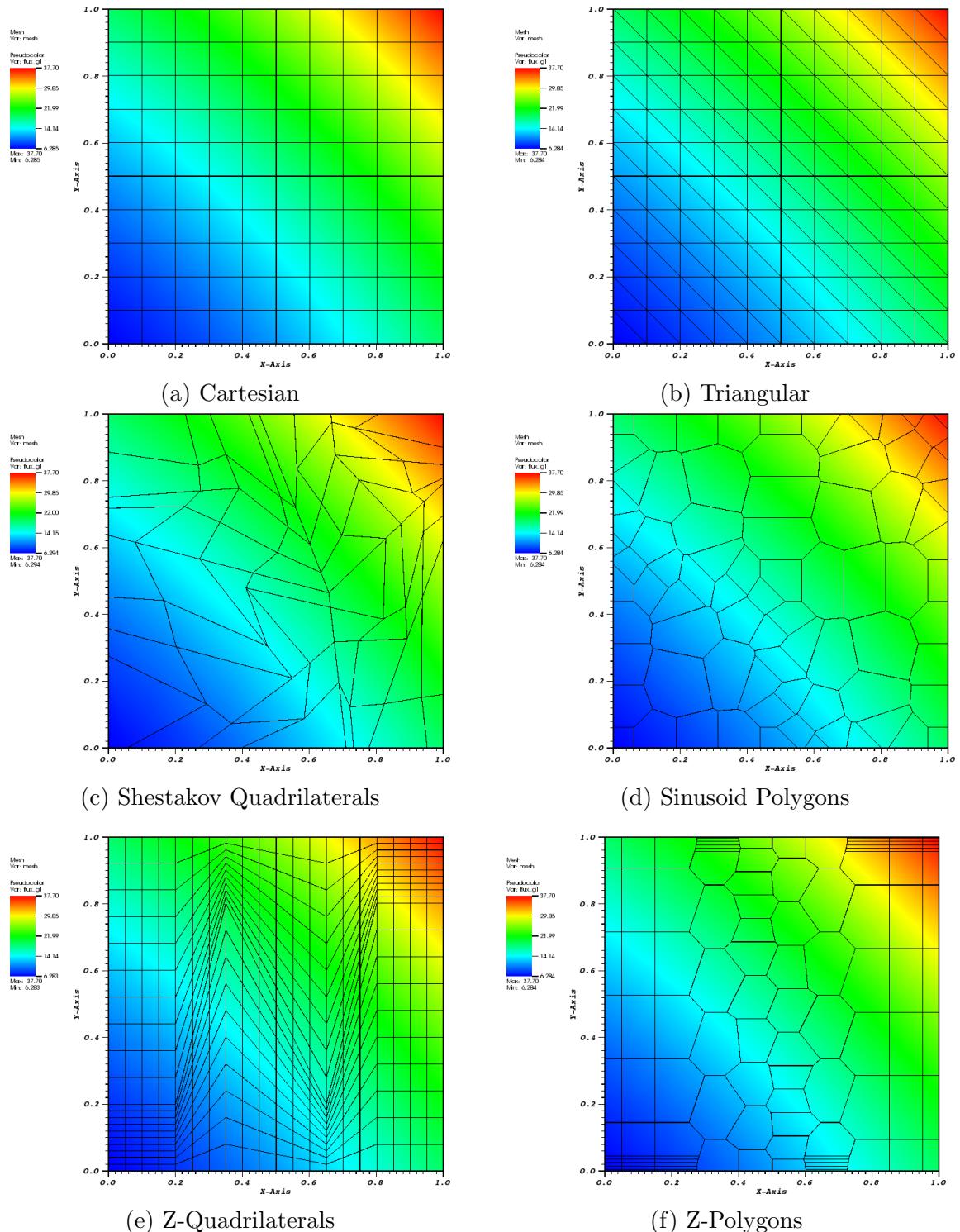


Figure 1.35: Plots of the exactly-quadratic solution with the quadratic serendipity maximum entropy basis functions.

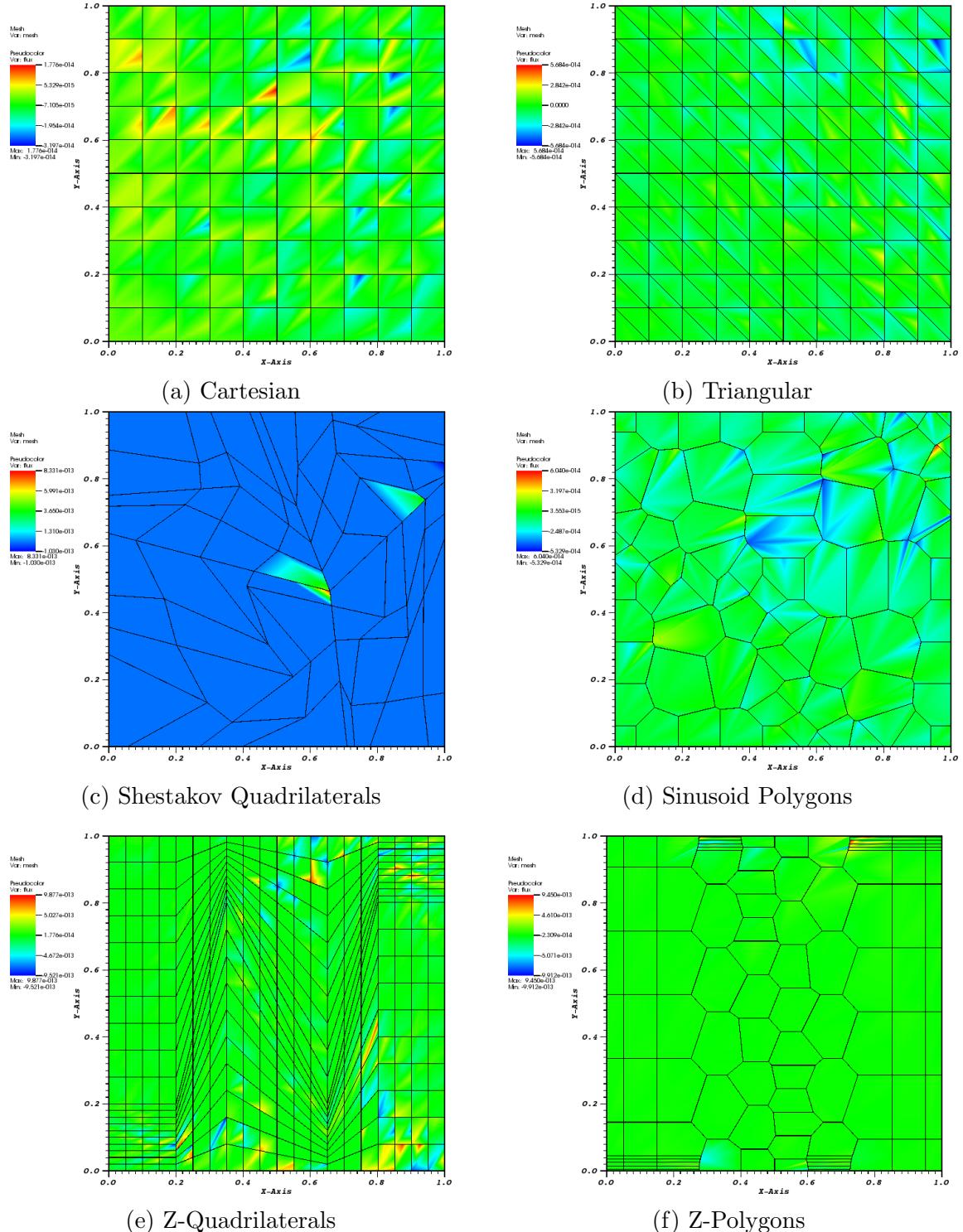


Figure 1.36: Plots of the error of the exactly-quadratic solution with the quadratic serendipity Wachspress basis functions.

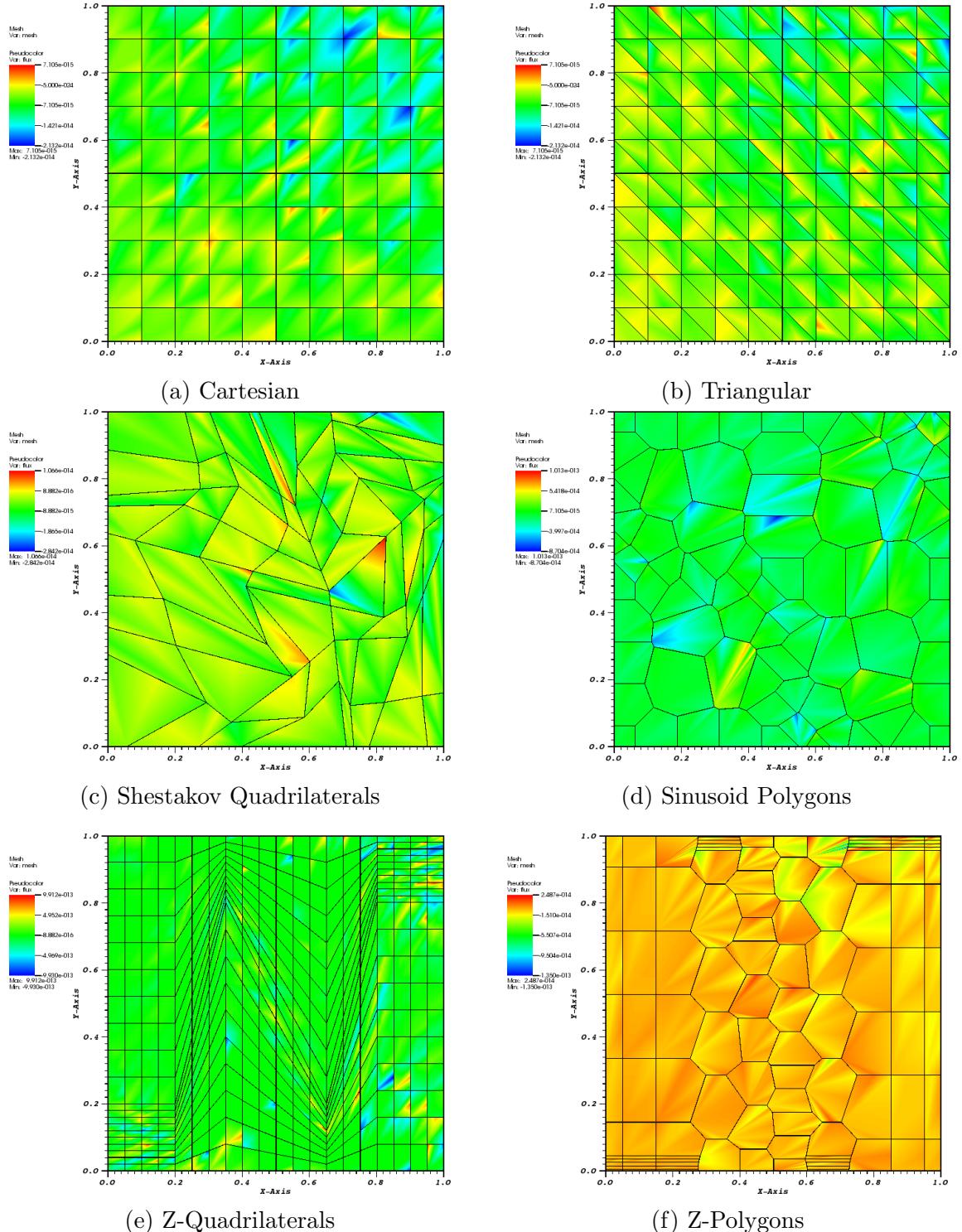


Figure 1.37: Plots of the error of the exactly-quadratic solution with the quadratic serendipity PWL basis functions.

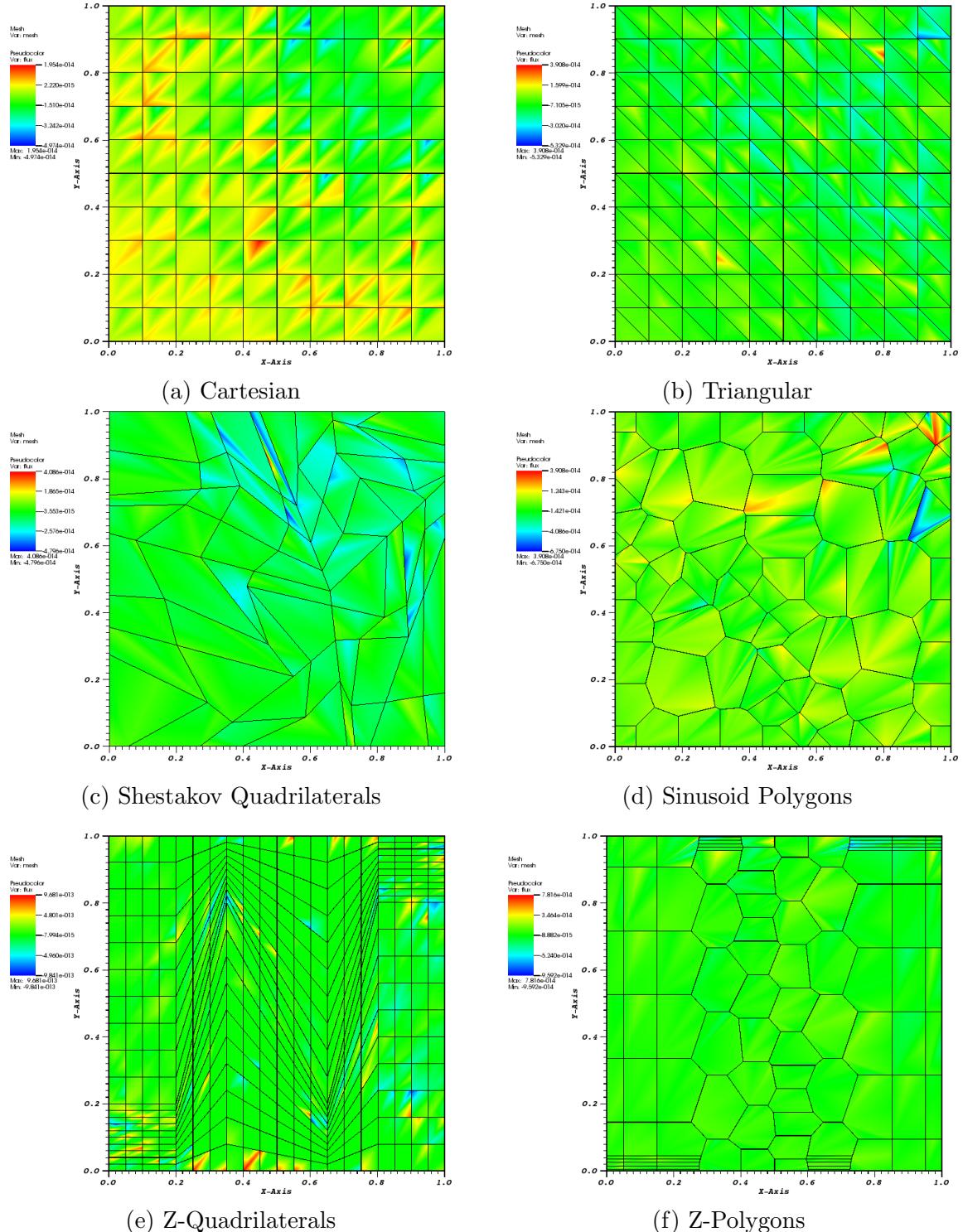


Figure 1.38: Plots of the error of the exactly-quadratic solution with the quadratic serendipity mean value basis functions.

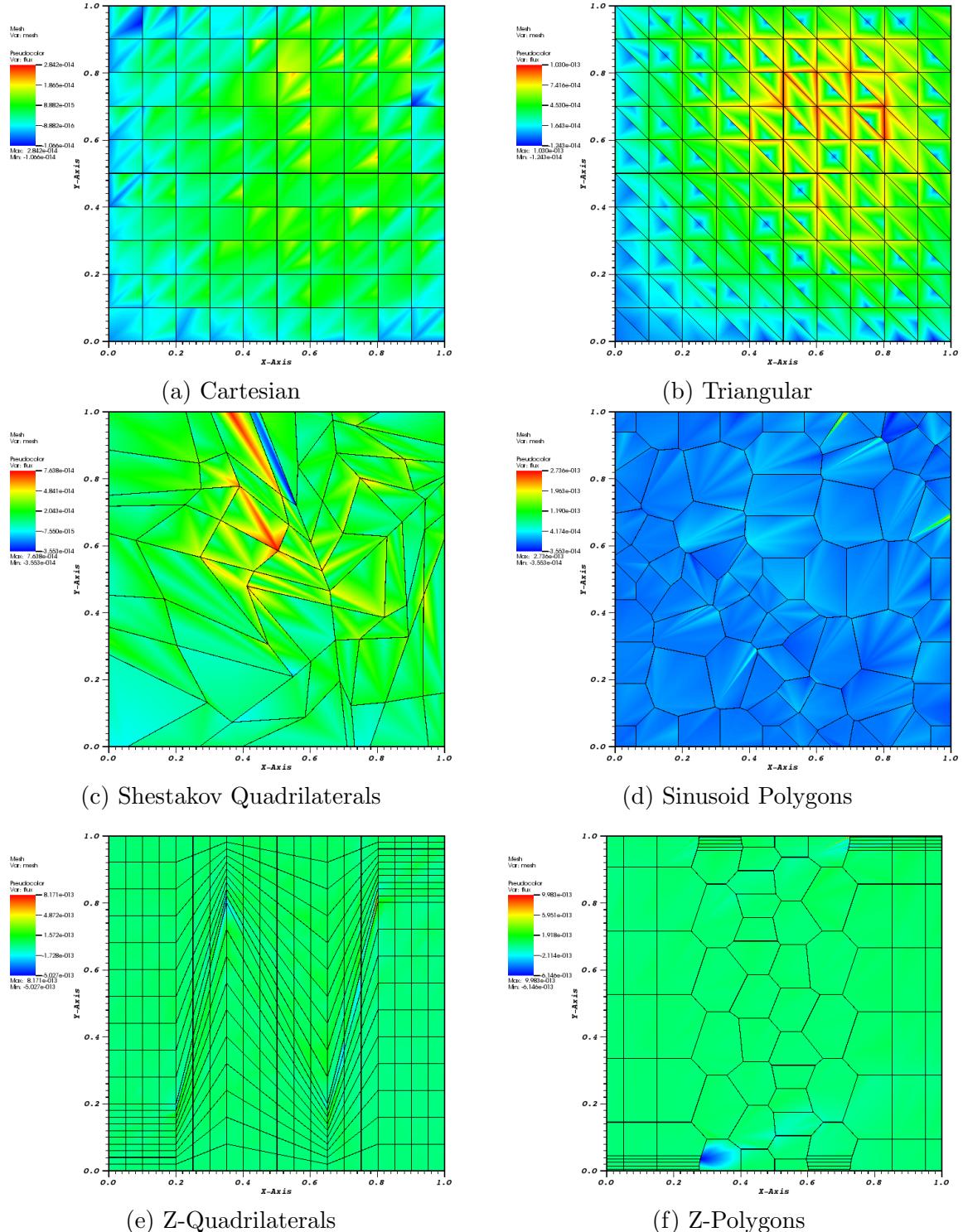


Figure 1.39: Plots of the error of the exactly-quadratic solution with the quadratic serendipity maximum entropy basis functions.

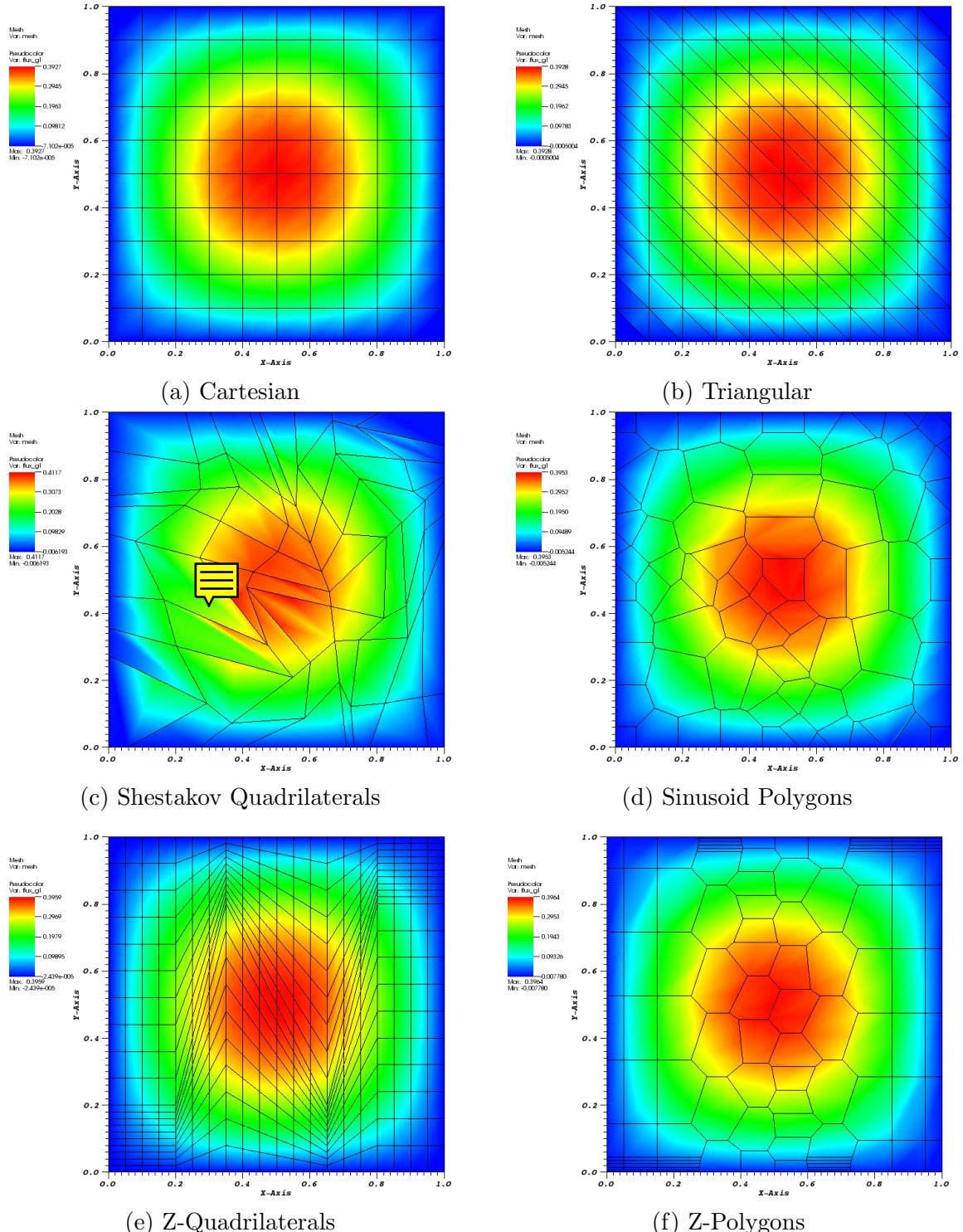


Figure 1.40: Quadratic solutions containing the x^2y^2 term for different meshes with the quadratic maximum entropy basis functions.

Table 1.2: L_2 -norm of the error in the quadratic solution containing the x^2y^2 term for the different quadratic serendipity basis functions on different mesh types.

Mesh Type	Basis Functions			
	Wachspress	PWL	Mean Value	Max. Entropy
Cartesian	3.5e-06	2.72e-05	8.29e-06	3.86e-05
Triangular	5.13e-05	5.13e-05	5.13e-05	5.13e-05
Shes. Quad	3.97e-04	3.37e-04	2.81e-04	3.91e-04
Sine Poly	3.75e-05	1.22e-04	7.62e-05	1.39e-04
Z-Poly	2.93e-05	3.07e-05	2.6e-05	3.46e-05
Z-Quad	2.98e-05	8.73e-05	5.08e-05	1.17e-04

MES, MMS enforces a given solution by use of a derived functional form for the driving source of the problem (Q_{ext}). However, unlike MES, we enforce a spatial solution that cannot be captured by the interpolation of the finite element space.

For this example, we choose the following solution and problem parameters and characteristics:

1. Constant total cross section so that parameterized material properties are not necessary;
2. No scattering to avoid solution discontinuities from the S_N discretization;
3. No solution dependence in angle to avoid introducing angular discretization error;
4. Analytical solutions that are C^∞ continuous in space for both the angular flux and 0th order flux moment;;

5. The angular flux solution is zero on the boundary for all incident directions - this is identical to vacuum boundaries which can ease code development.

To satisfy these characteristics, we choose to analyze two different solution spaces. The first is a smoothly varying sinusoid solution with no extreme local maxima. The second solution is a product of the quadratic function from Eq. (1.88) and a gaussian which yields a significant local maximum.

The sinusoid flux solutions, $\{\Psi_s, \Phi_s\}$, have the following parameterized form,

$$\begin{aligned}\Psi_s(x, y) &= \sin(\nu \frac{\pi x}{L_x}) \sin(\nu \frac{\pi y}{L_y}), \\ \Phi_s(x, y) &= 2\pi \sin(\nu \frac{\pi x}{L_x}) \sin(\nu \frac{\pi y}{L_y}),\end{aligned}\tag{1.90}$$

where ν is a frequency parameter. We restrict this parameter to positive integers ($\nu = 1, 2, 3, \dots$) to maintain characteristic 5 of the solution and problem space. The gaussian solution space, $\{\Psi_g, \Phi_g\}$, that has its local maximum centered at (x_0, y_0) has the parameterized form,

$$\begin{aligned}\Psi_g(x, y) &= C_M x(L_x - x)y(L_y - y) \exp\left(-\frac{(x - x_0)^2 + (y - y_0)^2}{\gamma}\right), \\ \Phi_g(x, y) &= 2\pi C_M x(L_x - x)y(L_y - y) \exp\left(-\frac{(x - x_0)^2 + (y - y_0)^2}{\gamma}\right),\end{aligned}\tag{1.91}$$

where the constants in the equations are:

$$C_M = \frac{100}{L_x^2 L_y^2} \quad \gamma = \frac{L_x L_y}{100}.\tag{1.92}$$

For this example, we choose the dimensionality of our problem to be $[0, 1]^2$ which makes $L_x = L_y = 1$ for both the sinusoid and gaussian solutions. For the sinusoid solution, we select the frequency parameter, ν , to be 3 and for the gaussian solution

we set the local maximum: $x_0 = y_0 = 0.75$. With these parameters, the sinusoid solution will have local minima and maxima of -2π and 2π , respectively, and the gaussian solution will have a global maximum of $\frac{225}{32}\pi \approx 22.1$.

For these two examples, we will use a different meshing strategy between them. The convergence rate analysis for the sinusoid problem will be performed on three different mesh types: 1) Cartesian meshes, 2) ordered-triangular meshes, and 3) polygonal meshes. The Cartesian and triangular meshes are the same as those used for the exactly-linear and exactly-quadratic problems. The polygonal meshes are the same as those used for the diffusion limit problem. We give an example of how the mesh will be refined for these different types in Figure 1.41. For the gaussian problem, we will utilize AMR to form our polygonal meshes. The numbers of polygons and their generation history will depend on the basis functions used and the mesh irregularity that is imposed.

The convergence rates of the sinusoid solution are given in Figures 1.44, 1.45, and 1.44 for the Cartesian, triangular, and polygonal meshes, respectively. The total cross section was set to 1.0 and S_8 level-symmetric quadrature was used. First, we provide some examples of the sinusoid solutions. Figures 1.42 and 1.43 give the contour plots for the sinusoid solutions using the linear and quadratic PWL basis functions, respectively. The meshes used are identical (in the same order) as those given in Figure 1.41. Next, we plot the L_2 -norm of the error against the number of spatial degrees of freedom in the problem. From the theoretical analysis of Section ??, we expected and obtained slope values of the convergence rate of -1 and $-3/2$ for the linear and quadratic basis functions, respectively.

Because we are using AMR for the gaussian problem, we will only analyze the PWL, MV, and ME basis functions since the Wachspress functions cannot handle degenerate polygons. For both the linear and quadratic AMR runsets, the refine-

ment criterion, α , was set to 0.1. For each linear and quadratic basis function, we performed mesh refinement where the maximum irregularity was set to either 1, 2, or 3. Figure 1.47, 1.48, and 1.49 give the convergence history of the 2D gaussian error in the L_2 -norm using the PWL, mean value, and maximum entropy basis functions, respectively. We plot the different AMR cases with the corresponding uniform refinement case ($\alpha = 0$) for comparison. We can clearly see that we obtain the proper convergence rates for all cases, and the AMR sets provide better solution accuracy with significantly less spatial degrees of freedom. Finally, we provide an example of a pair of AMR meshes and solutions in Figure 1.50 using the ME basis functions. We can see a clear difference in how the linear and quadratic basis functions will refine, with the quadratic functions causing much smoother refinement to occur.

1.5.5 Convergence Rate Analysis in a Purely-Absorbing Medium

Our next numerical examples involve studying the convergence rates of transport solutions in a purely-absorbing medium. Specifically, we seek to analyze the effects of mesh alignment along the solution discontinuities. From the theory presented in Section ??, the convergence rate of the discretized transport solution, Φ_h , to the exact solution, Φ , is given by

$$\|\Phi - \Phi_h\|_{L_2} \leq C \frac{h^{q+1/2}}{(p+1)^q}, \quad (1.93)$$

where $\min(p + 1/2, r - 1/2)$ and r is the regularity of the transport solution. If the mesh does not align with the discontinuities of the transport solution, then the solution convergence is restricted by r . However, if the mesh aligns with all of the solution discontinuities, then the maximum $p + 1$ converge rates can be observed. To study this behavior, we will analyze cases with aligned and non-aligned meshes.

For this example, two different transport problems will be evaluated. Both prob-

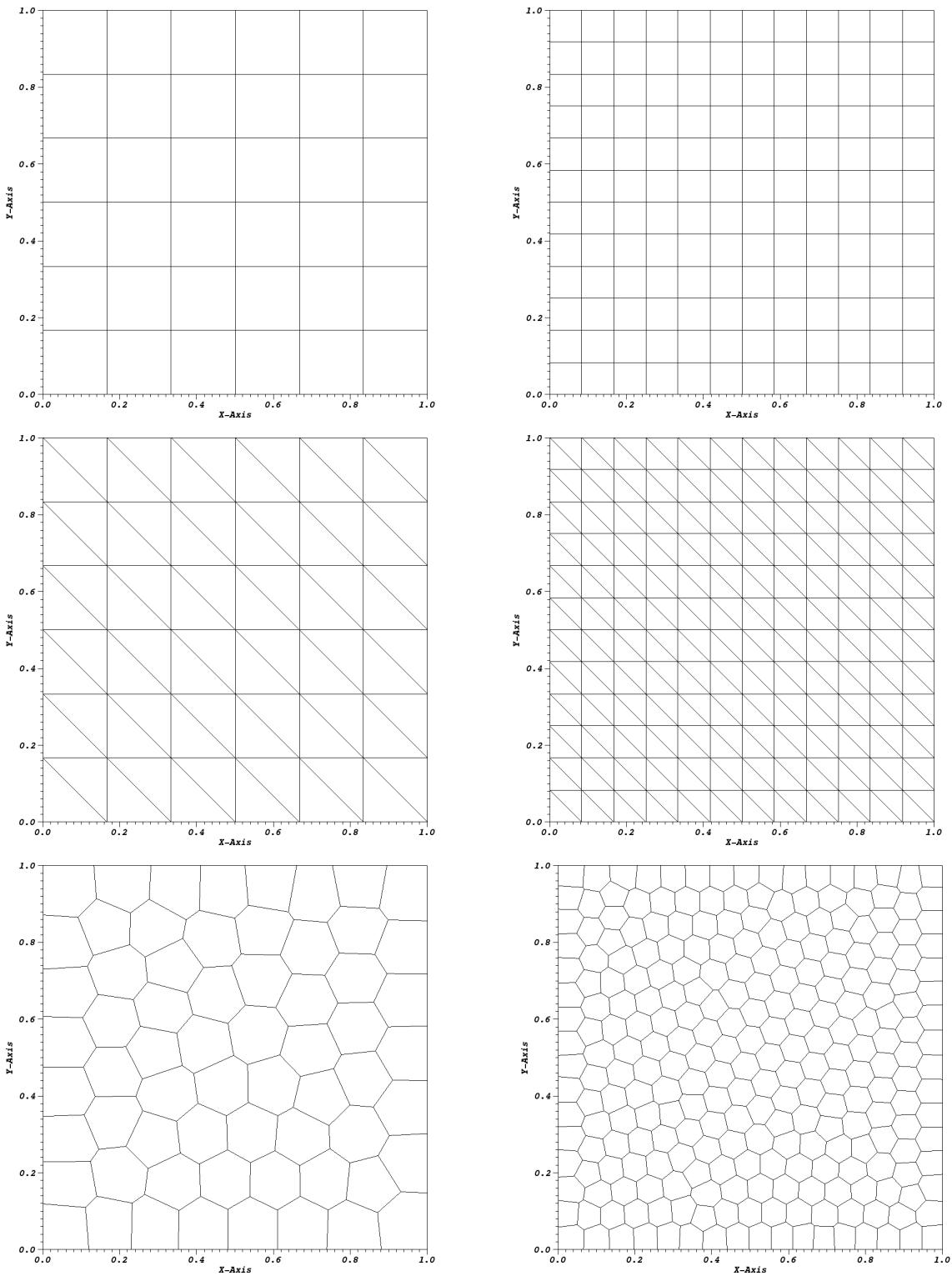


Figure 1.41: Examples of the mesh refinement for the sinusoid MMS transport problem for Cartesian (top), triangular (middle), and polygonal (bottom) meshes.

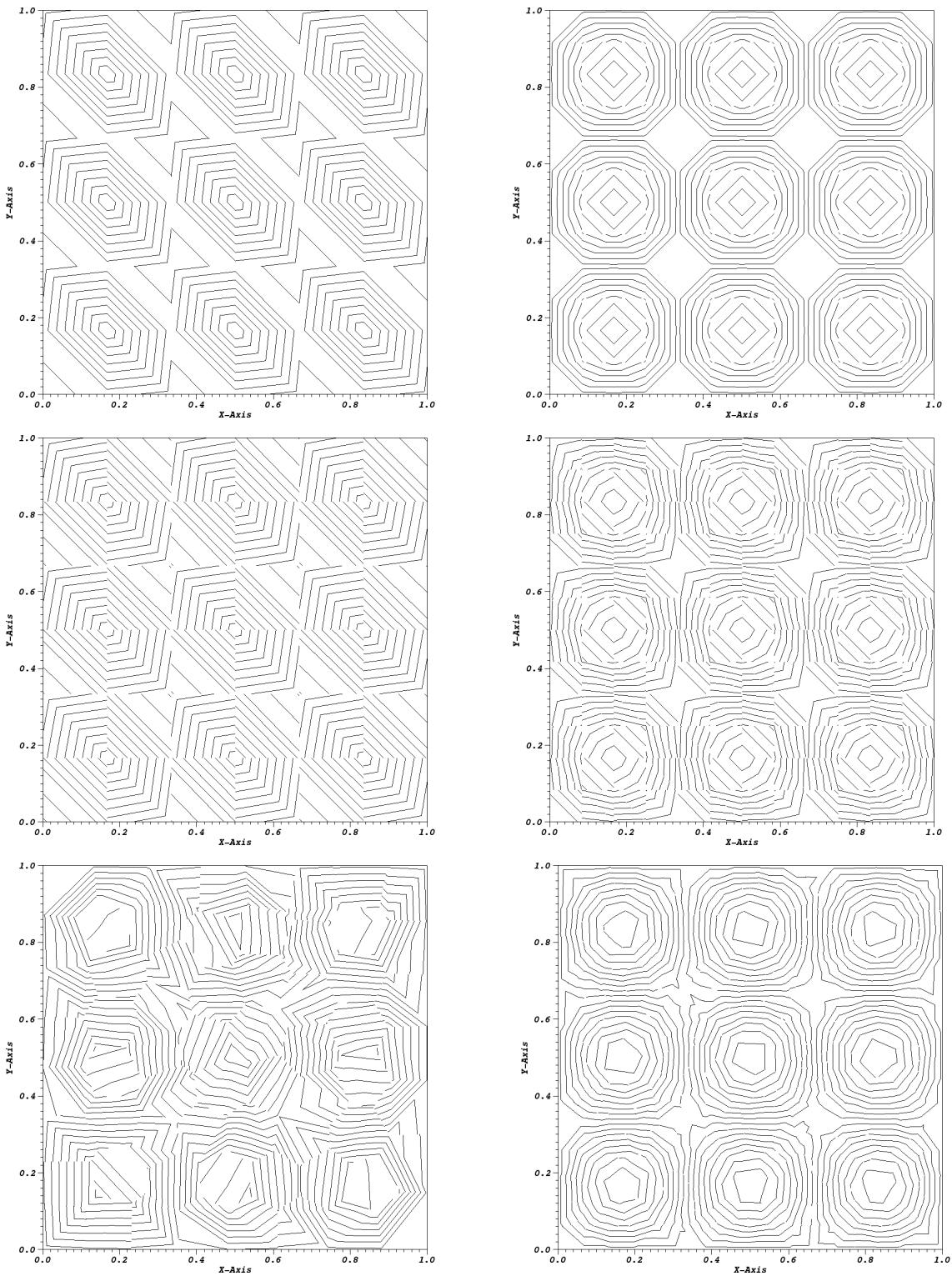


Figure 1.42: Example contour plots for the sinusoid problem using the linear PWL basis functions. The meshes used are those from Figure 1.41.

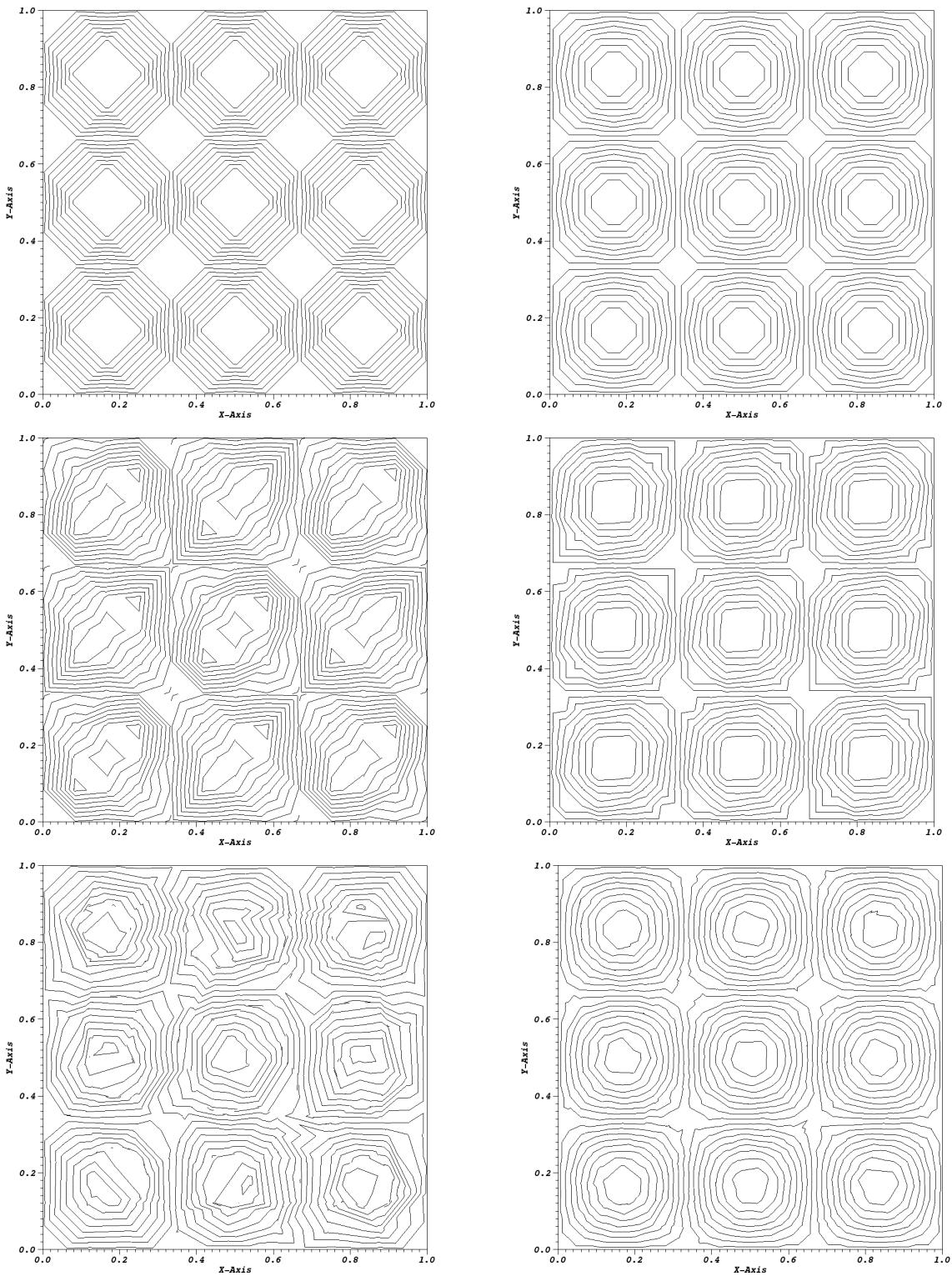


Figure 1.43: Example contour plots for the sinusoid problem using the quadratic PWL basis functions. The meshes used are those from Figure 1.41.

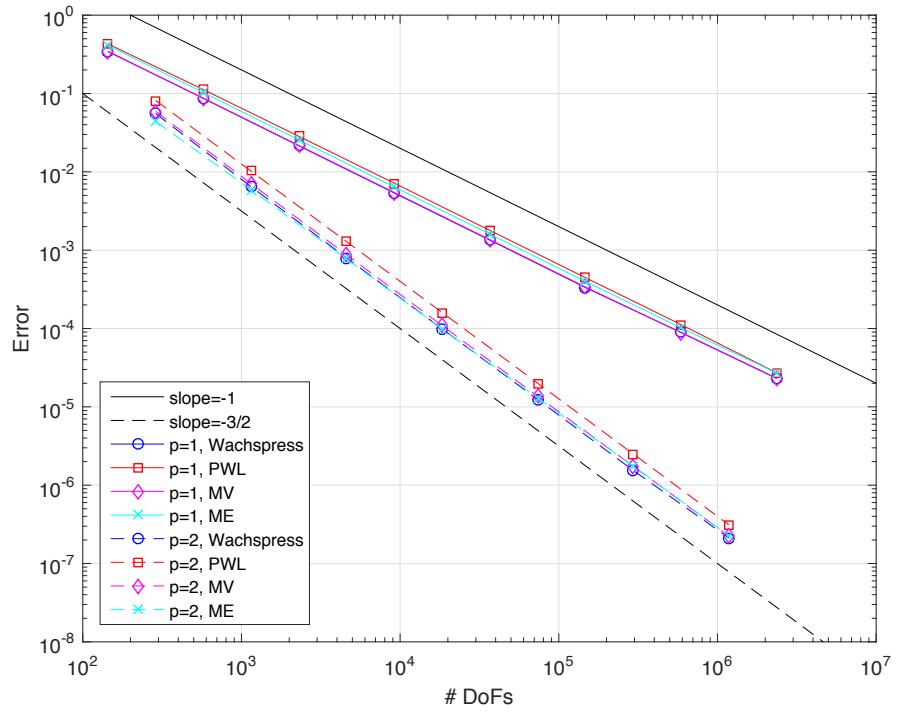


Figure 1.44: Convergence rates for the sinusoid MMS problem on a Cartesian mesh.

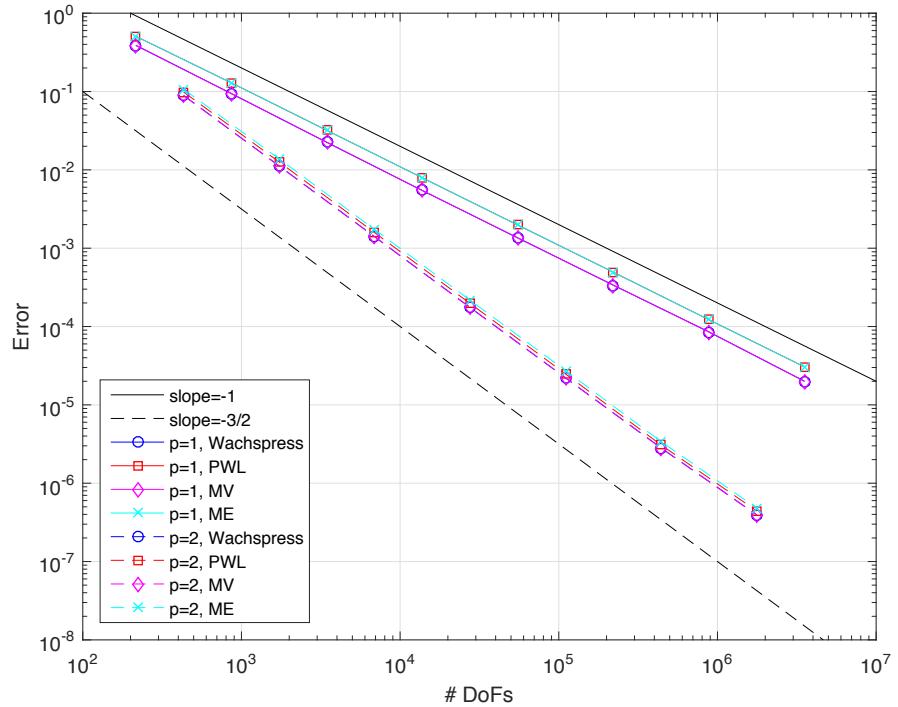


Figure 1.45: Convergence rates for the sinusoid MMS problem on an ordered-triangular mesh.

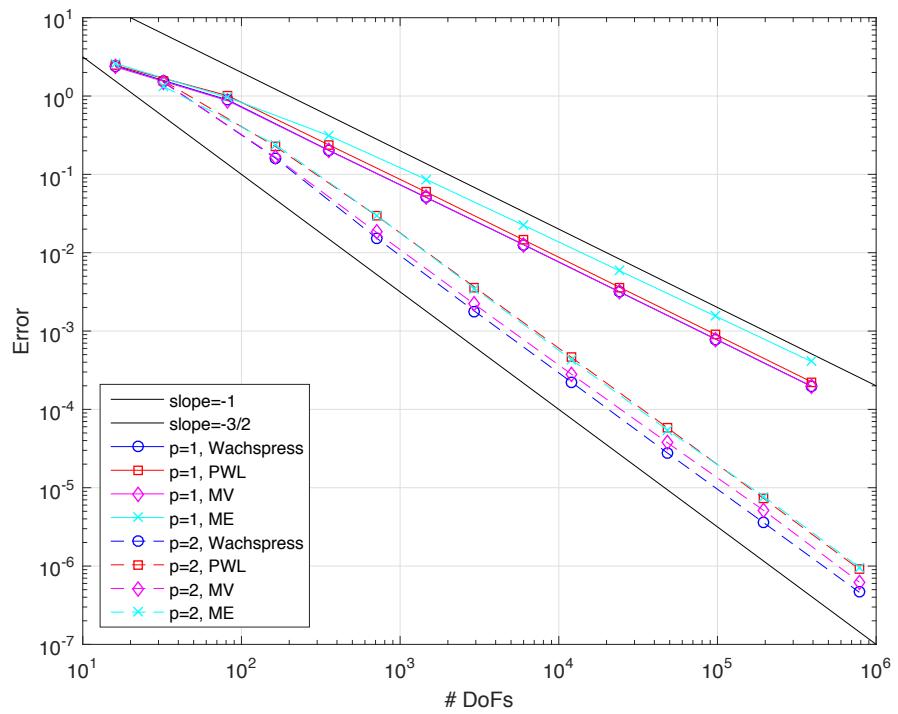


Figure 1.46: Convergence rates for the sinusoid MMS problem on a regular polygonal mesh.

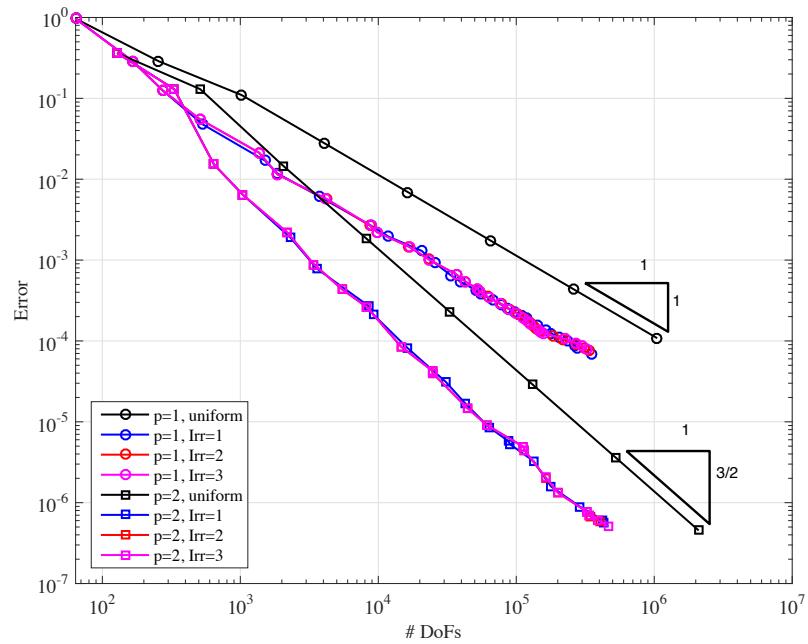


Figure 1.47: Convergence rates for the 2D Gaussian MMS problem using the PWL basis functions.

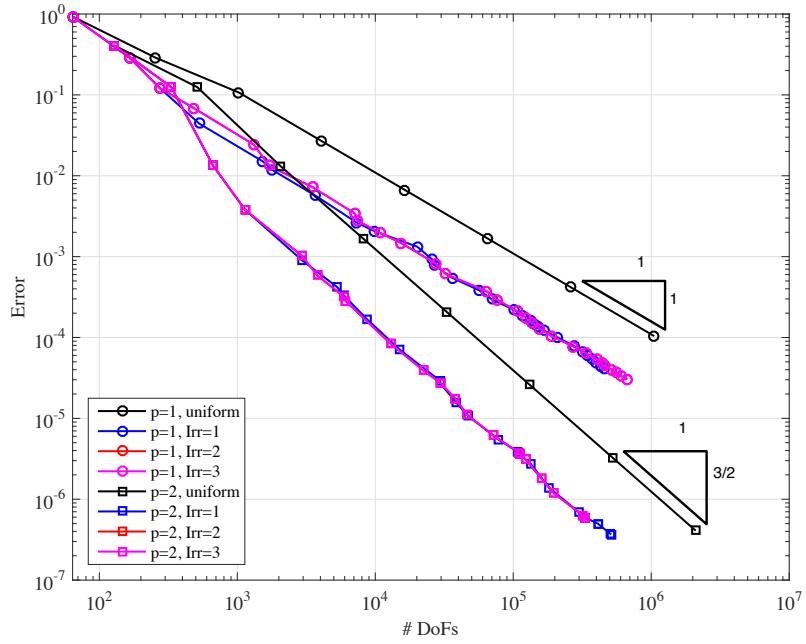


Figure 1.48: Convergence rates for the 2D Gaussian MMS problem using the mean value basis functions.

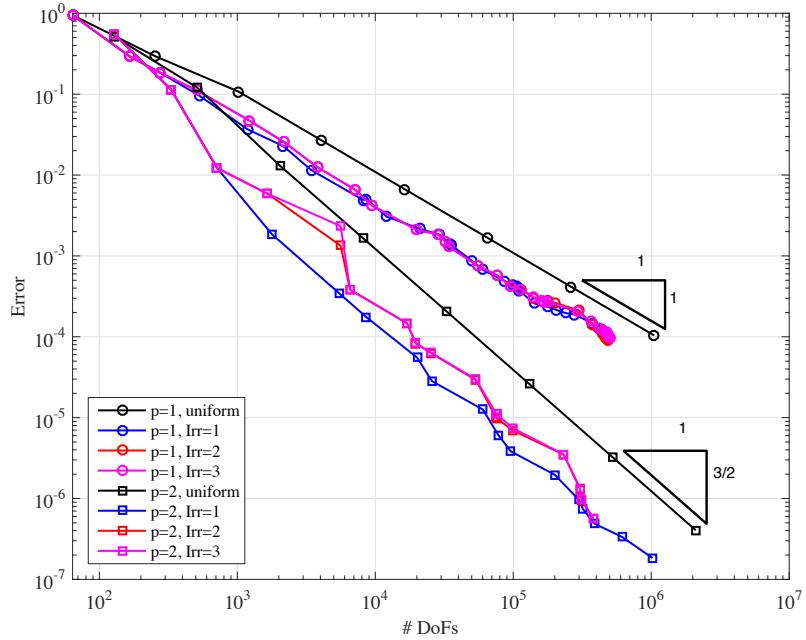


Figure 1.49: Convergence rates for the 2D Gaussian MMS problem using the maximum entropy basis functions.

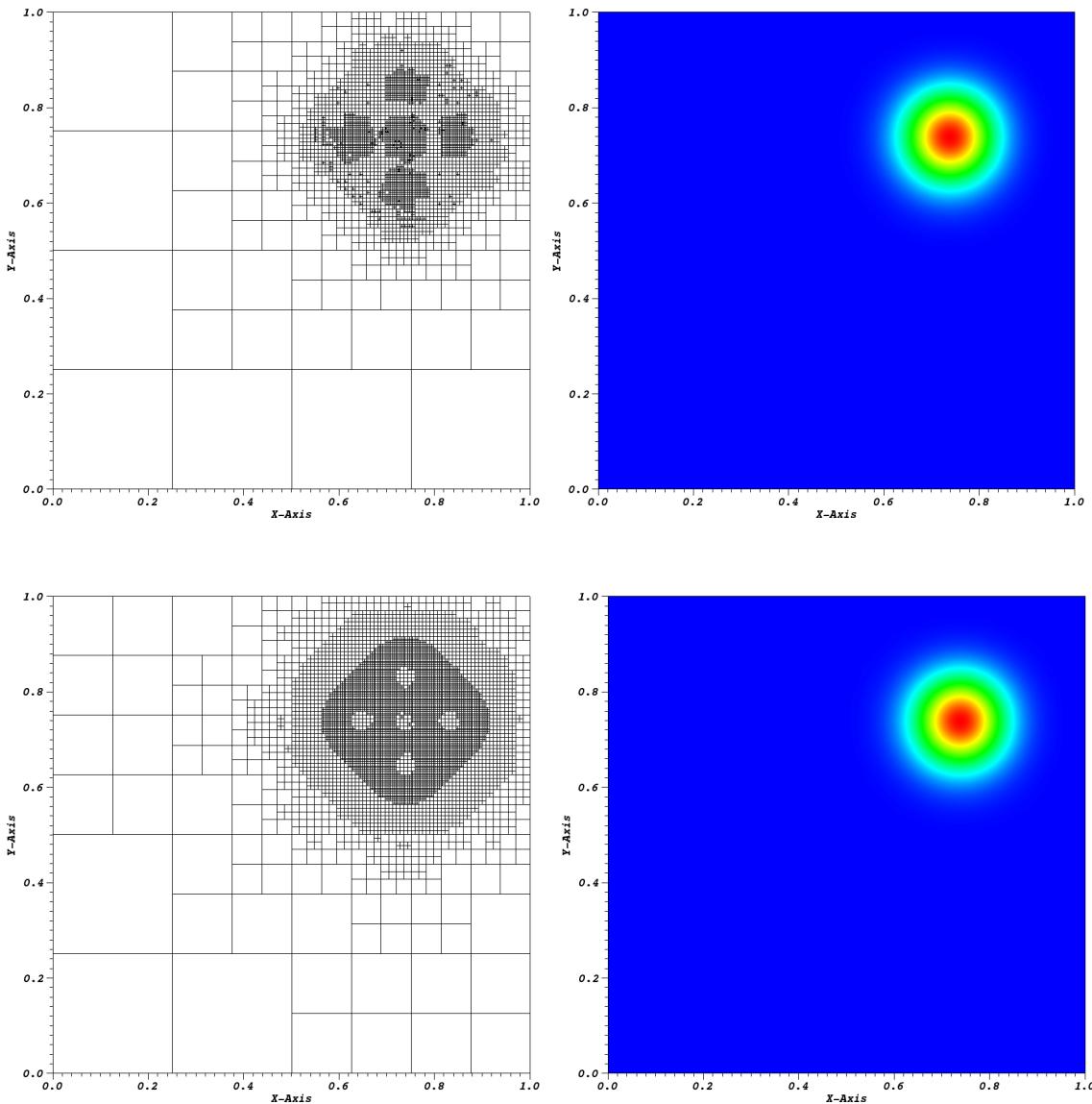


Figure 1.50: AMR meshes and solutions for the gaussian MMS problem using the maximum entropy coordinates: (top) linear basis functions at cycle 15 and (bottom) quadratic serendipity basis functions at cycle 08.

lems will consist of a purely-absorbing medium ($\sigma_s = 0$) and there will be no distributed source within the domain. All meshes for both problems will be contained in the unit square: $[0, 1]^2$ and S_4 level-symmetric quadrature is used. The first problem has incident angular flux on the left face at a downward 45° angle. The second problem has the same incident flux on the left face, but also has additional incident angular flux on the top face at a 45° angle aligned with that of the left face. This means that there is a solution discontinuity for both problems along the line from the vertex $(0, 1)$ to $(1, 0)$. Furthermore, the first problem has a void in the upper-right portion of the domain. Therefore, the problems have solution regularities of $1/2$ and $3/2$ for the first and second problem, respectively.

We will analyze four different mesh types. Two of them will align with solution discontinuity and other two will not. The meshes used are an ordered triangular mesh (aligns with the discontinuity), a regular Cartesian mesh (does not align with the discontinuity), a polygonal mesh (does not align with the discontinuity), and a split-polygonal mesh (aligns with the discontinuity). The split-polygonal meshes are formed by bisecting the polygonal meshes along the solution discontinuity. These meshes are shown in Figure 1.51. Next, in Figures 1.52 and 1.53, we provide example solutions for the problems with left-face incidence and then left-face and top-face incidence, respectively. It is easy to see in Figure 1.52 how the alignment of the triangular and split-polygonal meshes allow us to capture the void in the upper-right portion of the meshes. The numerical dispersion of the beam due to the nature of the upwind scheme forces bleeding of the flux into the void region for the Cartesian and polygonal meshes.

Next, we provide the convergence plots of the solution histories for both problems, for all meshes, for all basis functions, and for several values of σ_t . For each combination of problem and mesh type, convergence rate analysis was performed

using σ_t values of 1, 10, 50, and 100. First we analyze the problem with only left-face incidence. For the non-aligned meshes, we expect convergence rates of $1/2$ since the transport solutions only live in the $H^{1/2-\epsilon}$ space. The convergence rates are measured in the L_2 -norm and are plotted in Figures 1.54, 1.55, 1.56, and 1.57 for the triangular, Cartesian, polygonal, and split-polygonal meshes, respectively. With the aligned meshes, we can clearly see that convergence rates of $p + 1$ are obtained. The DGFEM solution “does not see” the lack of regularity in the transport solution. For the Cartesian and polygonal meshes, we see that convergence rates of about $1/2$ are obtained for optically thin meshes. However, for thicker domains in the pre-asymptotic range, higher convergence rates (without quite obtaining $p + 1$ values) are observed.

Finally, we analyze the problem with both left-face and top-face incidence of the angular flux. For the non-aligned meshes, we expect convergence rates of $3/2$ since the transport solutions only live in the $H^{3/2-\epsilon}$ space. Again, the convergence rates are measured in the L_2 -norm and are plotted in Figures 1.58, 1.59, 1.60, and 1.61 for the triangular, Cartesian, polygonal, and split-polygonal meshes, respectively. For the triangular and split-polygonal meshes, we again see convergence rates of order $p + 1$ in an identical manner to the left-face incidence problem. The non-aligned Cartesian and polygonal meshes give convergence rates of $3/2$ for the optically thin meshes. This higher convergence rate is achieved because the transport solution is now C^0 continuous yielding a higher regularity. Again, we also see for the thicker domains in the pre-asymptotic range higher convergence rates that tend to $p + 1$.

1.5.6 Searchlight Problem

Our final numerical example models a beam or searchlight through a vacuum. Similar problems were investigated in Dedner and Vollmöller [38] and Wang and

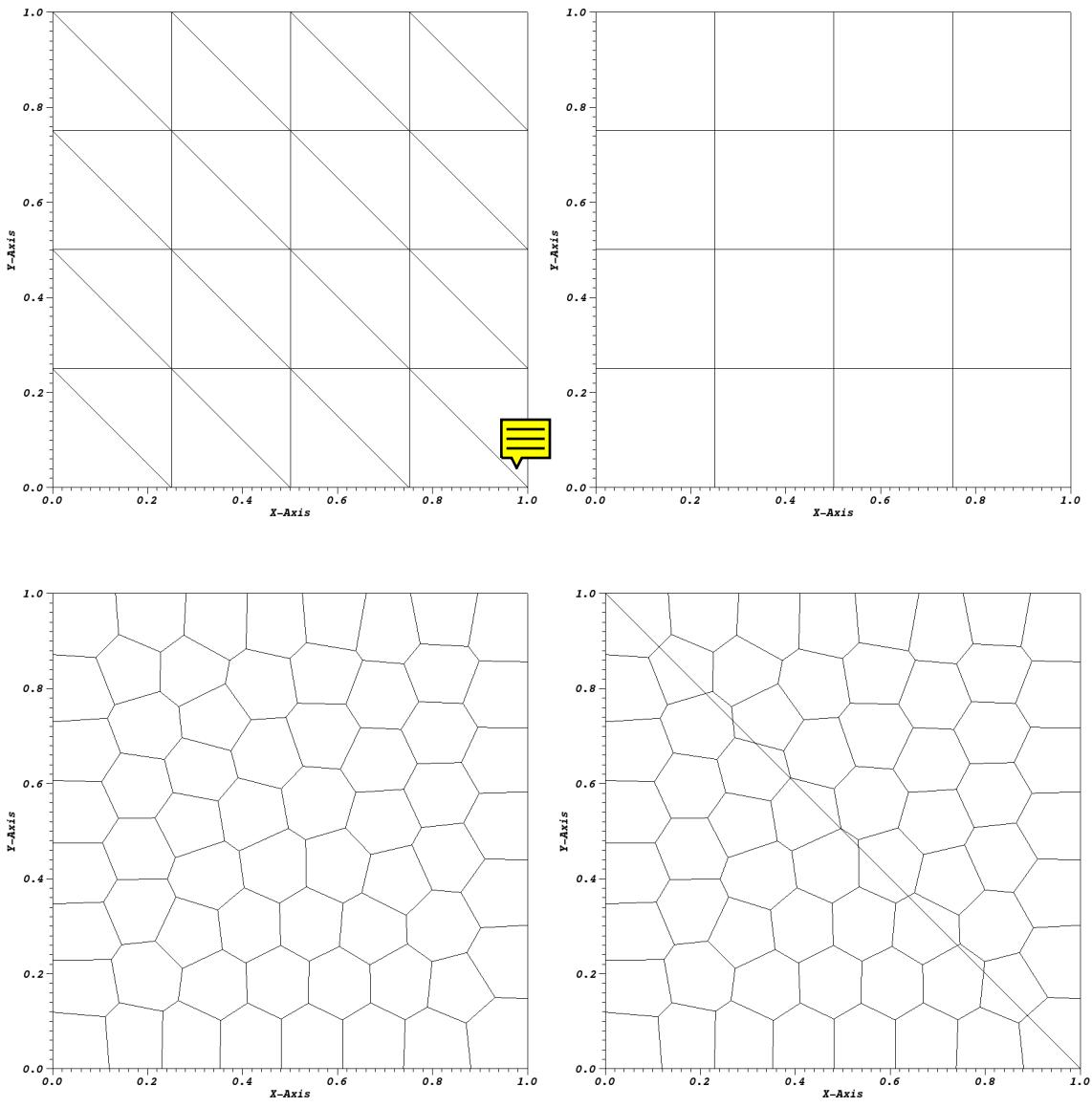


Figure 1.51: Mesh types used for the purely-absorbing medium problem.

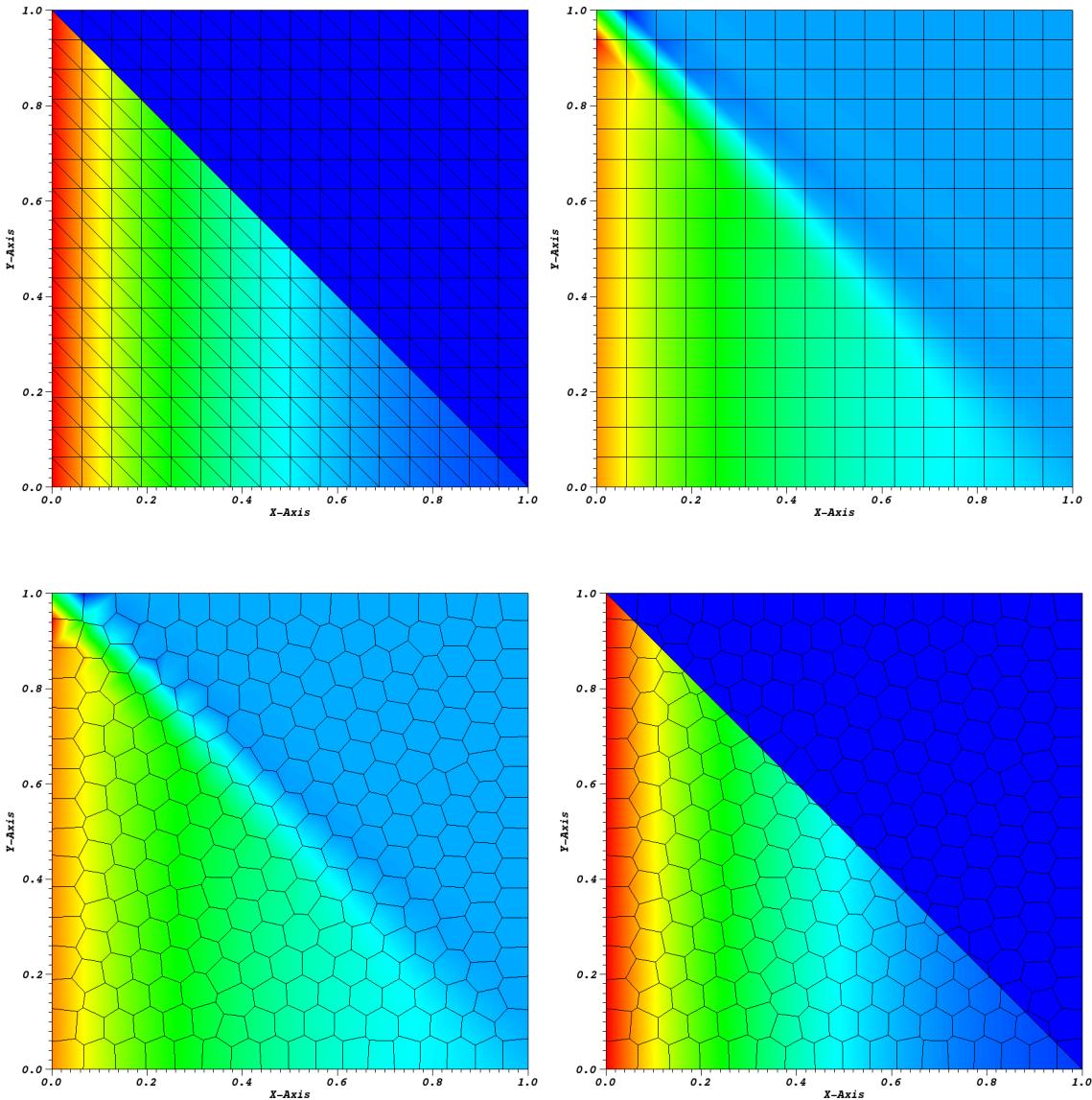


Figure 1.52: Example Solution of the purely-absorbing medium case with left-face incidence and $\sigma_t = 1$ using the linear PWL basis functions.

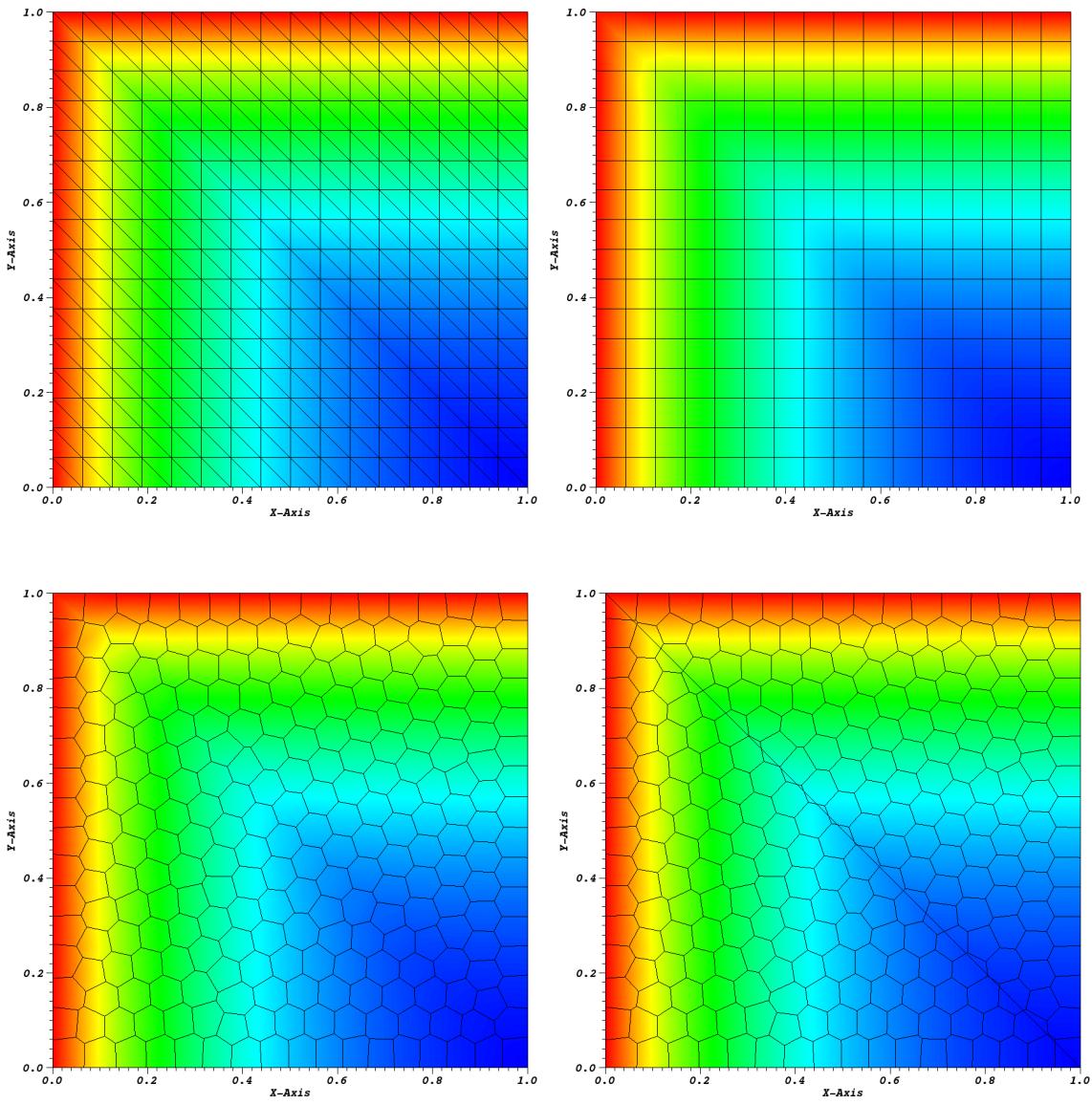


Figure 1.53: Example Solution of the purely-absorbing medium case with left-face and top-face incidence incidence and $\sigma_t = 1$ using the linear PWL basis functions.

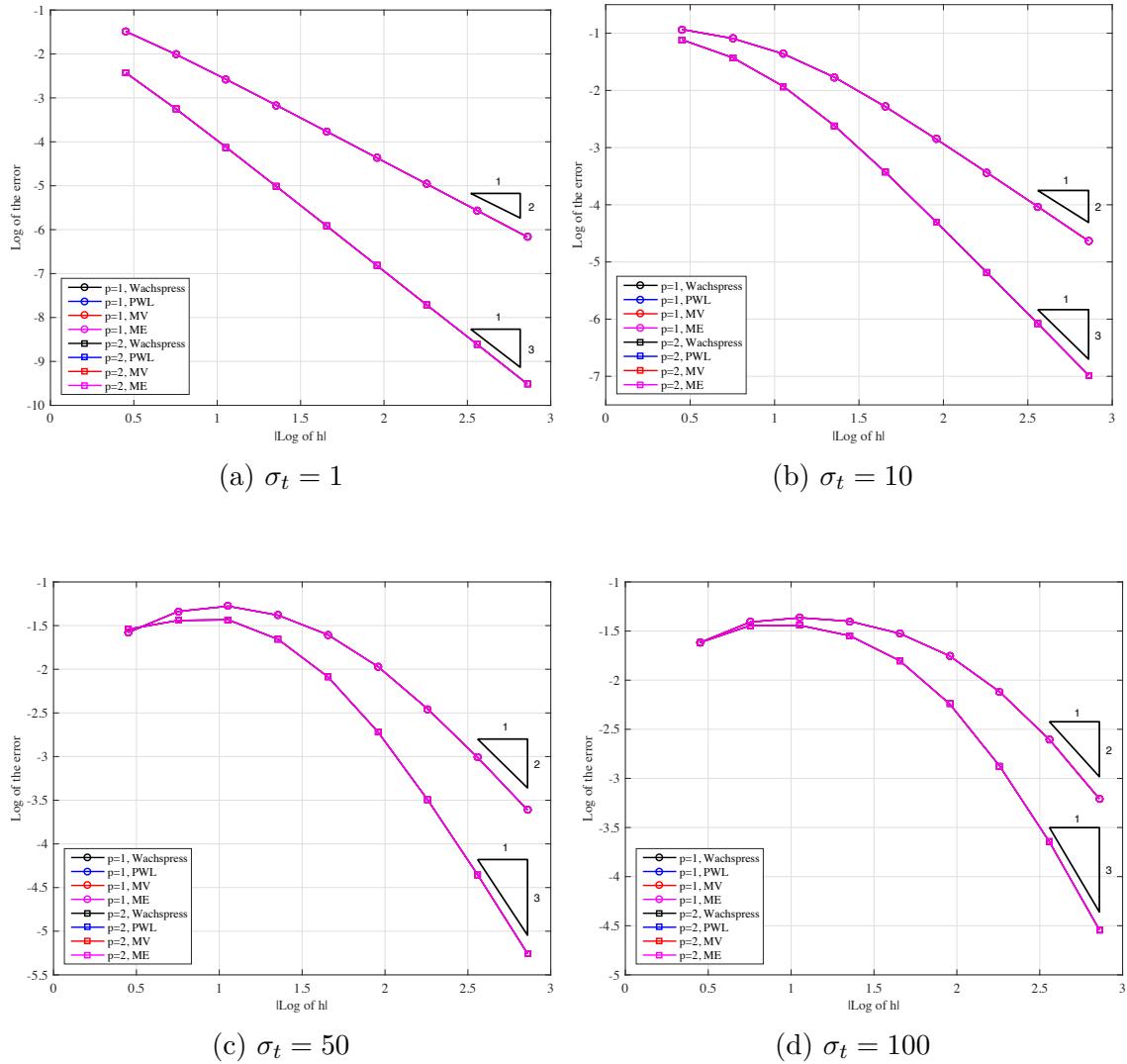


Figure 1.54: Convergence rates for the pure absorber problem with left-face incidence on triangular meshes with different values of σ_t .

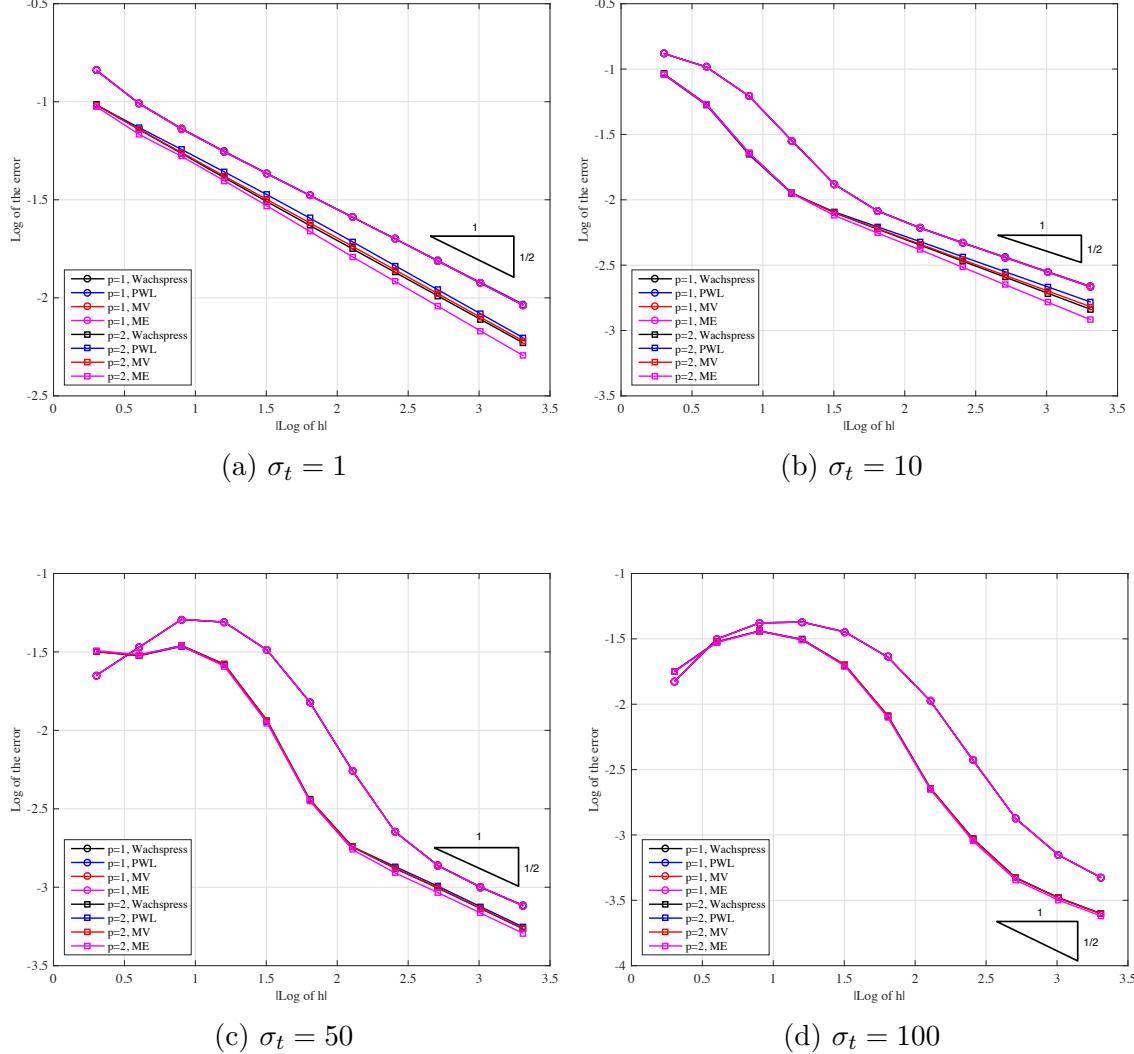


Figure 1.55: Convergence rates for the pure absorber problem with left-face incidence on Cartesian meshes with different values of σ_t .

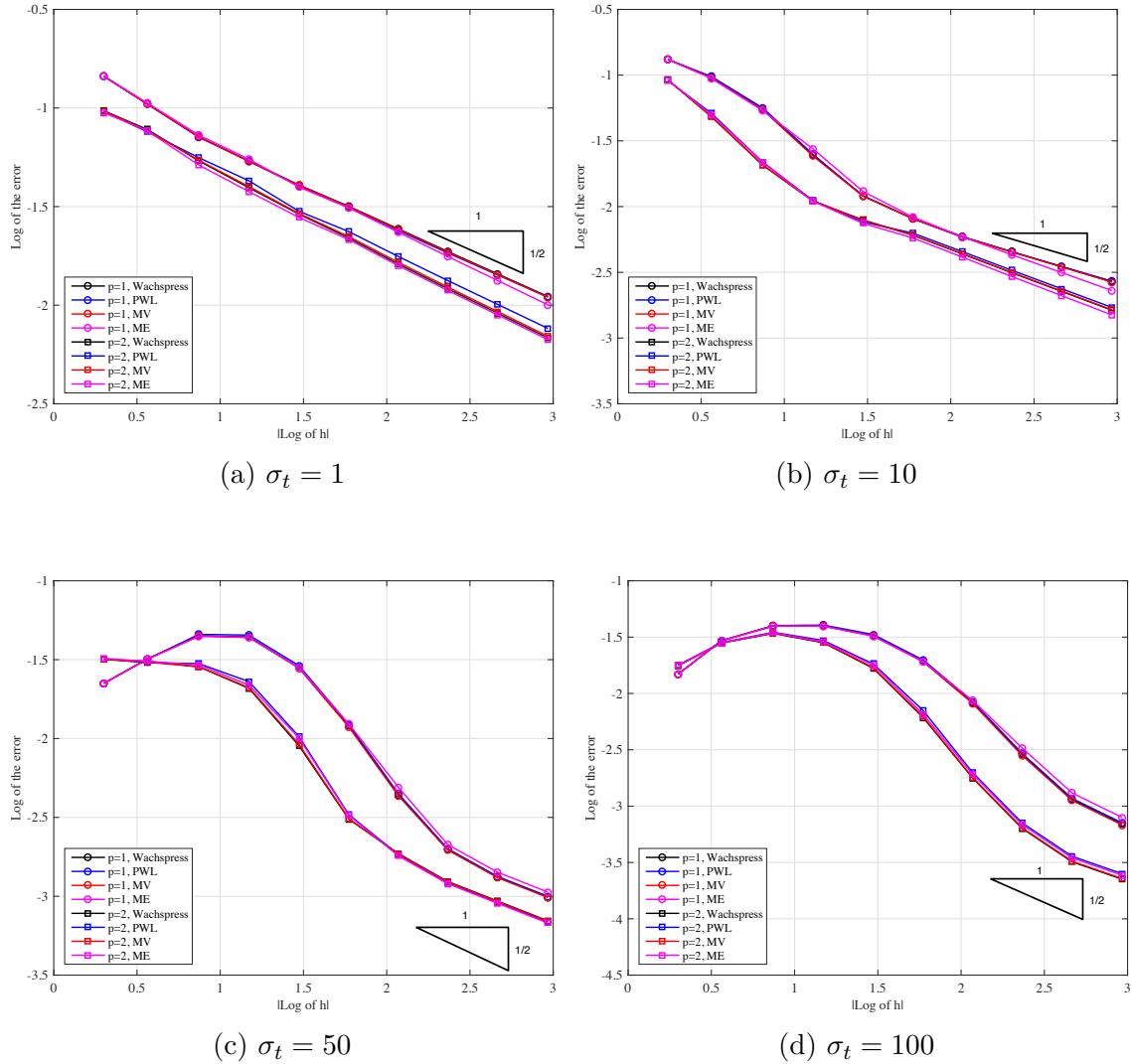


Figure 1.56: Convergence rates for the pure absorber problem with left-face incidence on polygonal meshes with different values of σ_t .

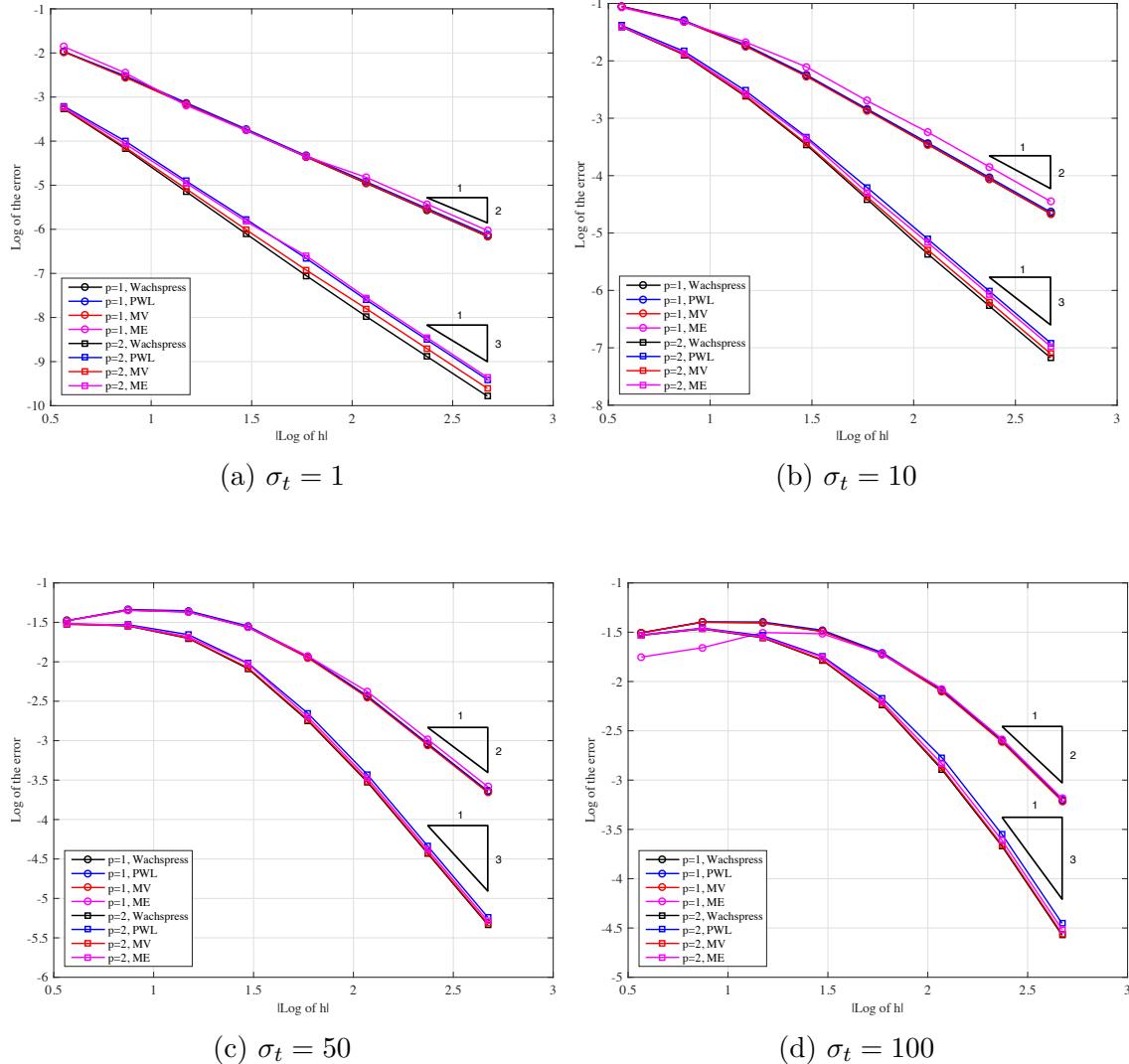


Figure 1.57: Convergence rates for the pure absorber problem with left-face incidence on split-polygonal meshes with different values of σ_t .

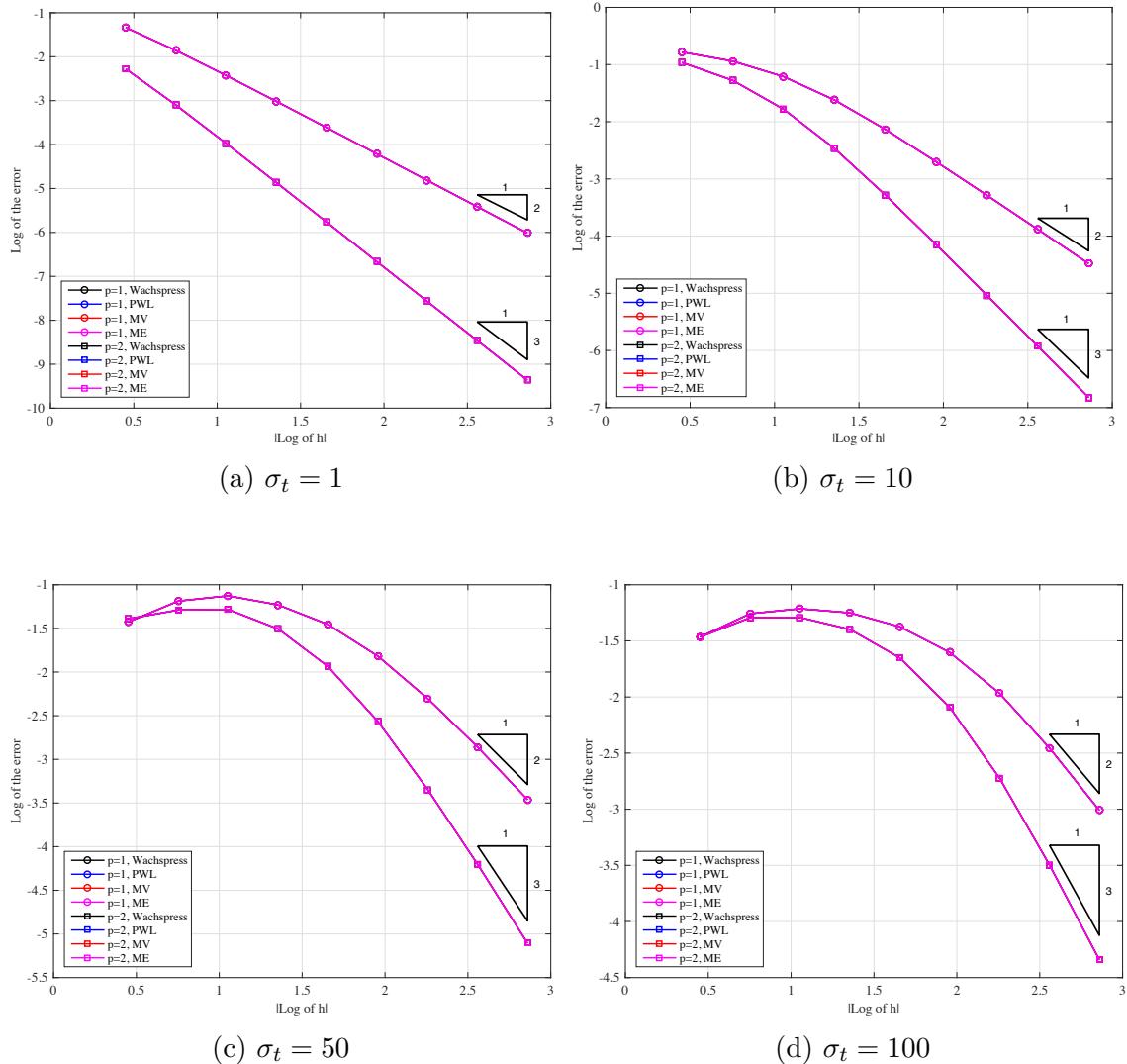


Figure 1.58: Convergence rates for the pure absorber problem with left-face and top-face incidence on triangular meshes with different values of σ_t .

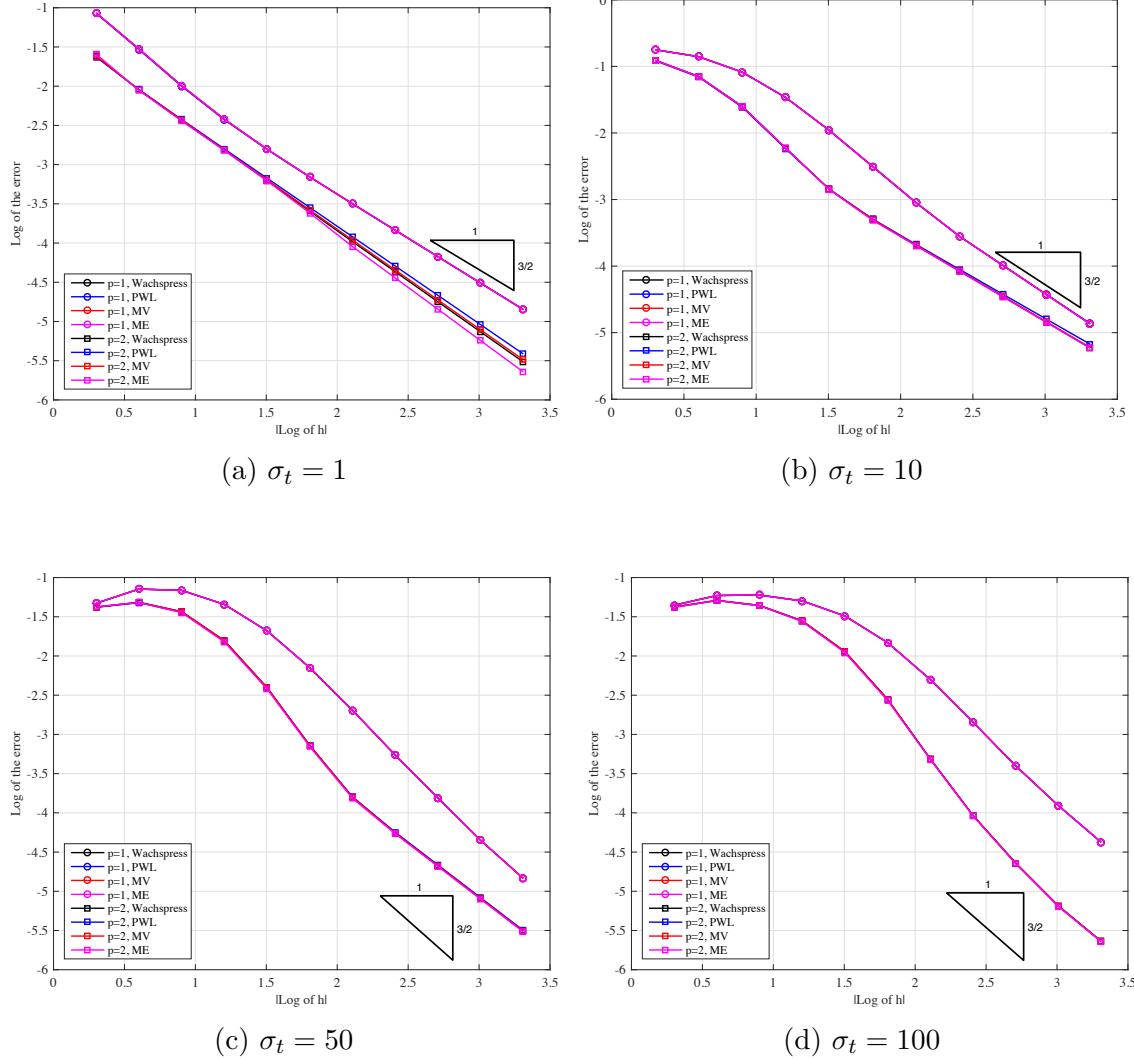


Figure 1.59: Convergence rates for the pure absorber problem with left-face and top-face incidence incidence on Cartesian meshes with different values of σ_t .

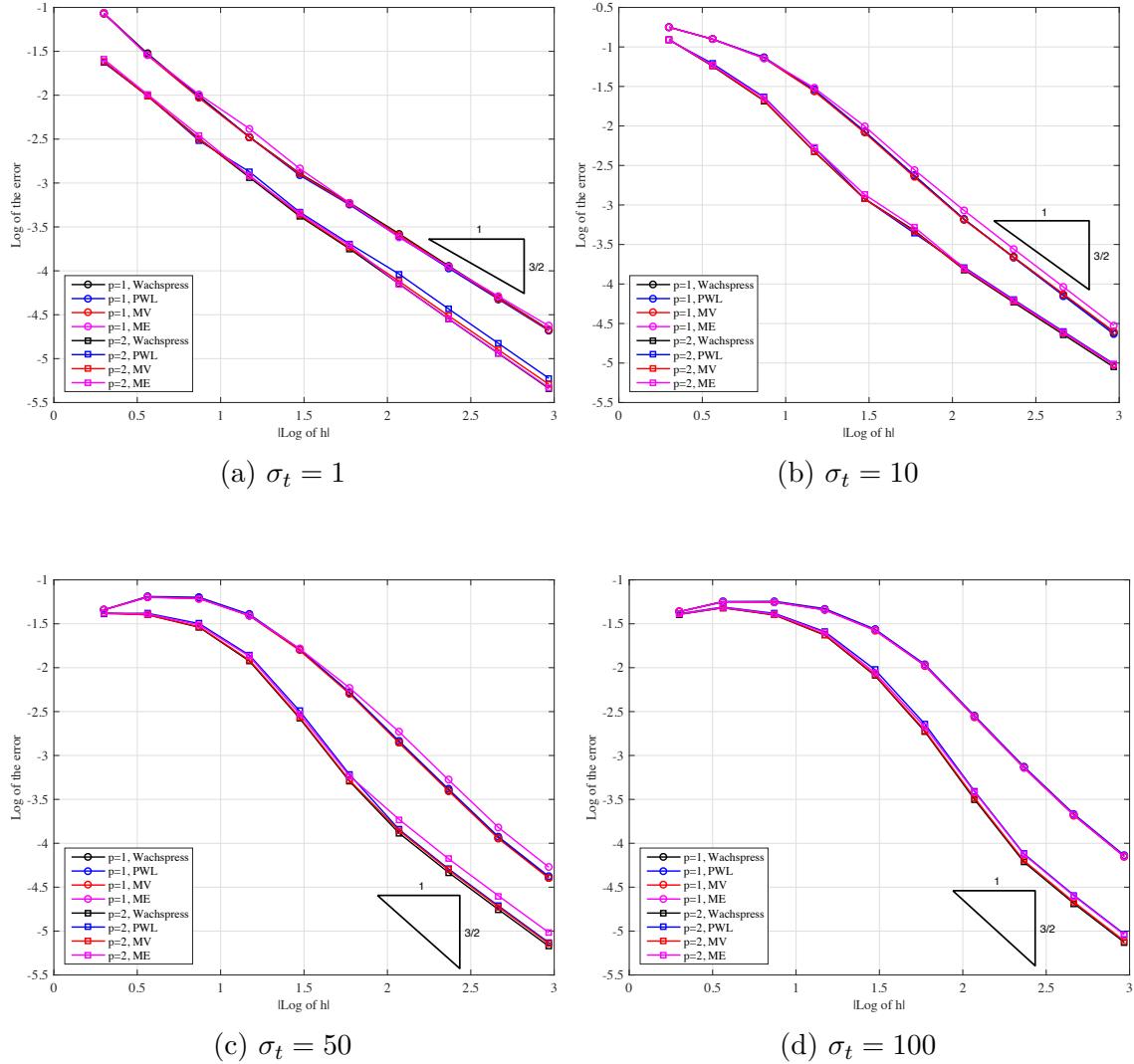


Figure 1.60: Convergence rates for the pure absorber problem with left-face and top-face incidence incidence on polygonal meshes with different values of σ_t .

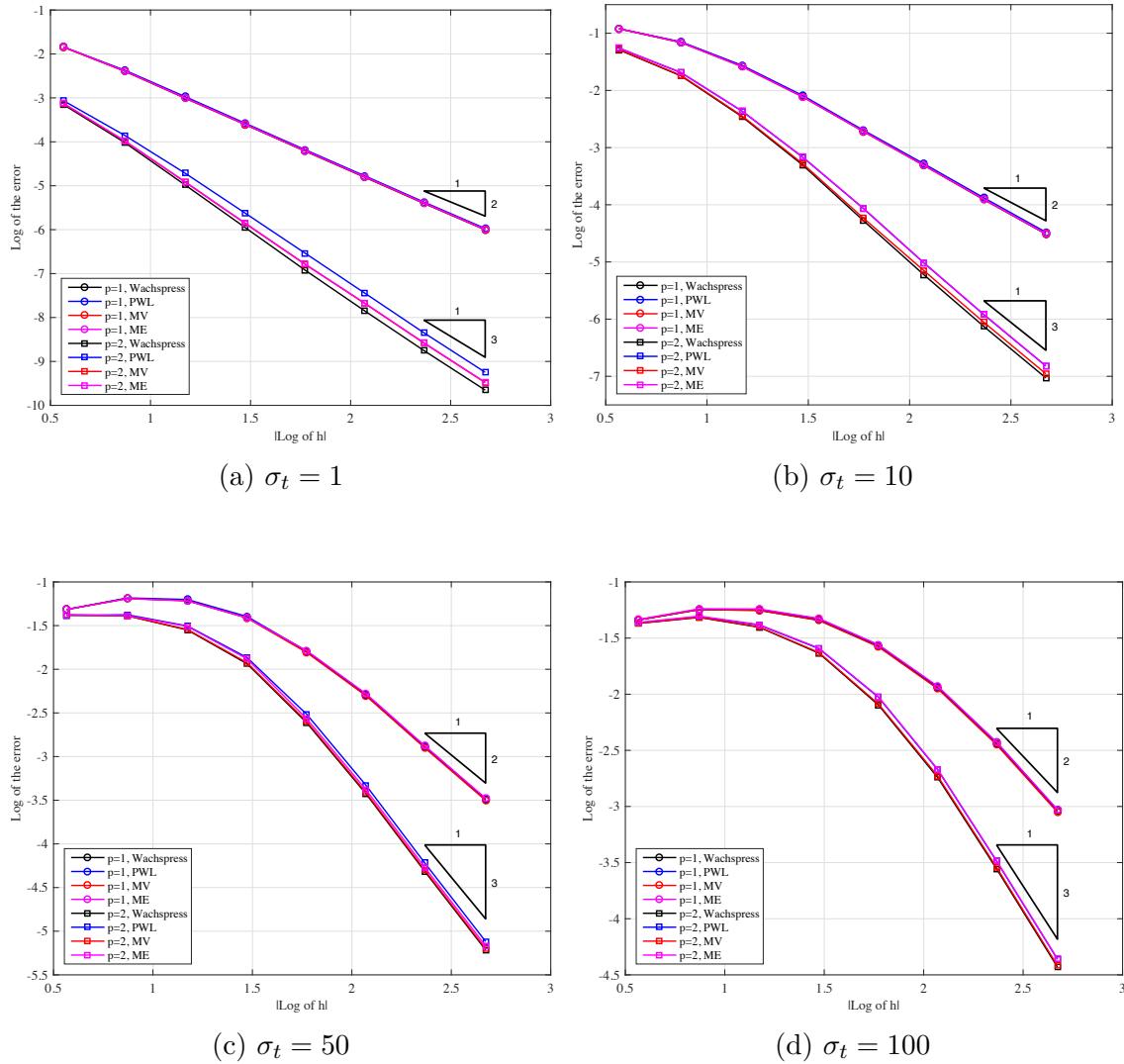


Figure 1.61: Convergence rates for the pure absorber problem with left-face and top-face incidence incidence on split-polygonal meshes with different values of σ_t .

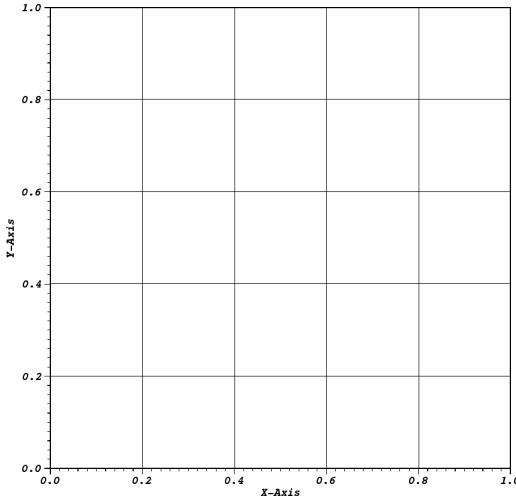


Figure 1.62: Initial mesh configuration for the searchlight problem before any refinement cycles.

Ragusa [39]. In this problem, an incident beam of neutrons is shined onto a small portion of a boundary, propagates through a vacuum, and then exits through a small portion of a different boundary. As the beam propagates through the vacuum, the spatial discretization causes radiation outflow through all downwind cell faces. This leads to numerical dispersion and will cause to beam to artificially broaden.

In this problem, we investigate an \mathbb{R}^2 domain of size $[0, 1]^2$ cm. The radiation enters the left boundary between $0.2 \leq y \leq 0.4$ with an un-normalized angular direction of $[1, 0.4]$. For this chosen direction, the radiation beam would analytically leave the right boundary between $0.6 \leq y \leq 0.8$. This means that any radiation leaving the right boundary for all other y values is due to the numerical dispersion of the beam. The initial mesh is a uniform 5x5 Cartesian grid as shown in Figure 1.62. We note that with this choice of initial grid, the analytical path of the beam is fully-contained within a spatial cell (a mesh cell does not bisect the analytical beam at the incoming and outgoing faces).

Since we are again using AMR for this problem, we will analyze the linear and

quadratic PWL, mean value, and maximum entropy basis functions (not the Wachspress basis functions). The refinement criterion, α , was set to 0.2 for all the linear elements and 0.1 for all the quadratic elements. The relative error of the leakage (in %) along the right face from $0.6 \leq y \leq 0.8$ is given in Figures 1.63, 1.64, and 1.65 for the PWL, mean value, and maximum entropy coordinates, respectively. For each of the figures we plotted the convergences histories for uniform refinement ($\alpha = 0.0$) as well as AMR runs where the maximum mesh irregularities were set to 1, 2, or 3. Like the purely-absorbing media cases without meshes aligned to the discontinuities, the convergence of the transport solution is limited by its regularity. This is apparent in that the AMR solutions give significantly better results than those obtained with uniform refinement.

Next, we plot the outgoing angular flux along the right face. We wish to see how the numerical dispersion is reduced with refinement. Figure 1.66 shows the outgoing fluxes for all of the linear and quadratic basis functions under uniform refinement. For all of the linear and quadratic basis functions we see the solution converging to the proper exit flux solution (looks like a step function along the face). However, there is still some over-shooting and under-shooting occurring, even for the most refined case. We then plot the outgoing angular fluxes for the AMR cases in Figures 1.67, 1.68, and 1.69 using the PWL, MV, and ME basis functions, respectively. For all of these cases, we can see better convergence to the analytical outgoing flux for the later AMR cycles. However, the over-shooting and under-shooting still remain.

1.6 Conclusions

In this chapter, we presented four different linearly-complete, barycentric, 2D polygonal basis functions to be used with the DGFEM transport equation: the Wachspress rational functions, the PWL coordinates, the mean value coordinates,

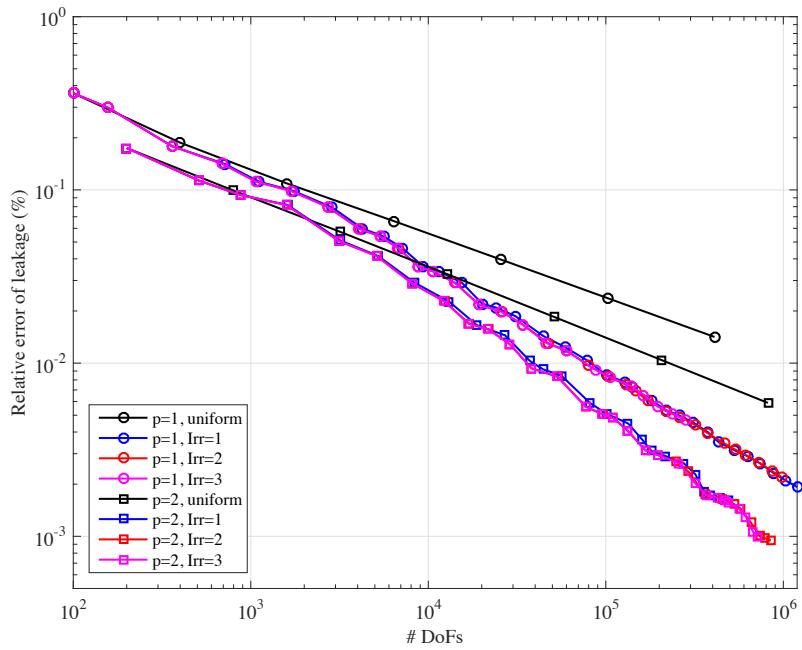


Figure 1.63: Convergence history for the searchlight problem using the PWL basis functions.

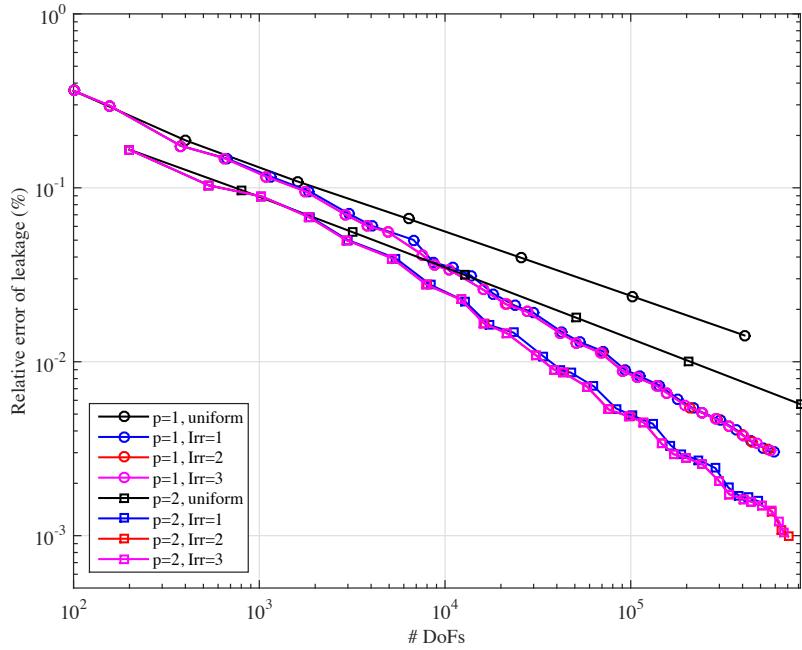


Figure 1.64: Convergence history for the searchlight problem using the mean value basis functions.

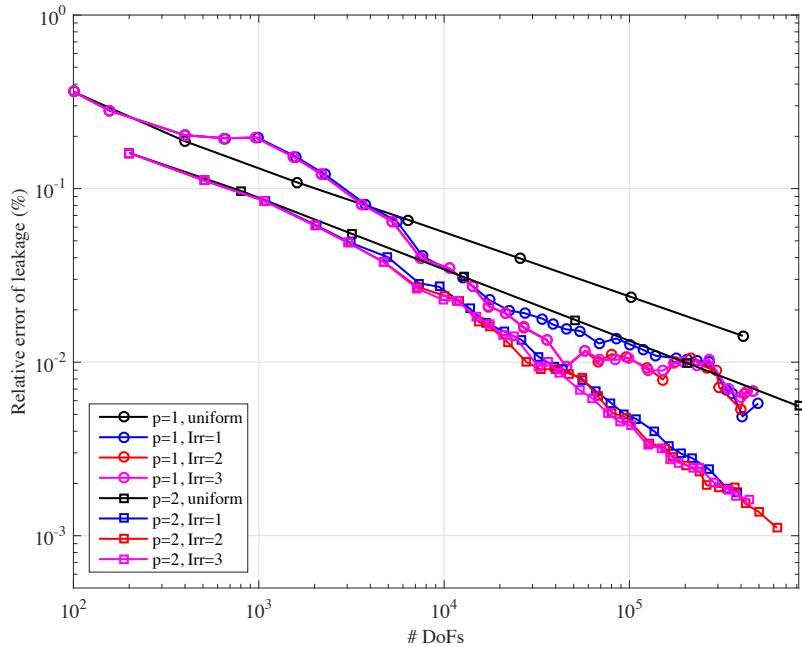


Figure 1.65: Convergence history for the searchlight problem using the maximum entropy basis functions.

and the maximum entropy coordinates. The Wachspress and the PWL coordinates had been previously utilized for DGFEM transport calculations, and we have extended the analysis to include the other two. Next, the procedure for converting these barycentric coordinates into the quadratic serendipity space of functions was given. For both the linear and quadratic coordinates, a simple quadrature rule for arbitrary polygons based on triangulation was used. We also provided some details of the 3D PWL coordinates that will be used in Chapter ?? for completeness.

Numerical results were obtained to demonstrate the completeness and convergence properties of these coordinates. The 2D linear and quadratic basis functions capture the thick diffusion limit, which is a necessary property for TRT calculations. The linear and quadratic basis functions can capture exactly-linear and exactly-quadratic solution spaces, respectively. Next, a series of convergence studies were

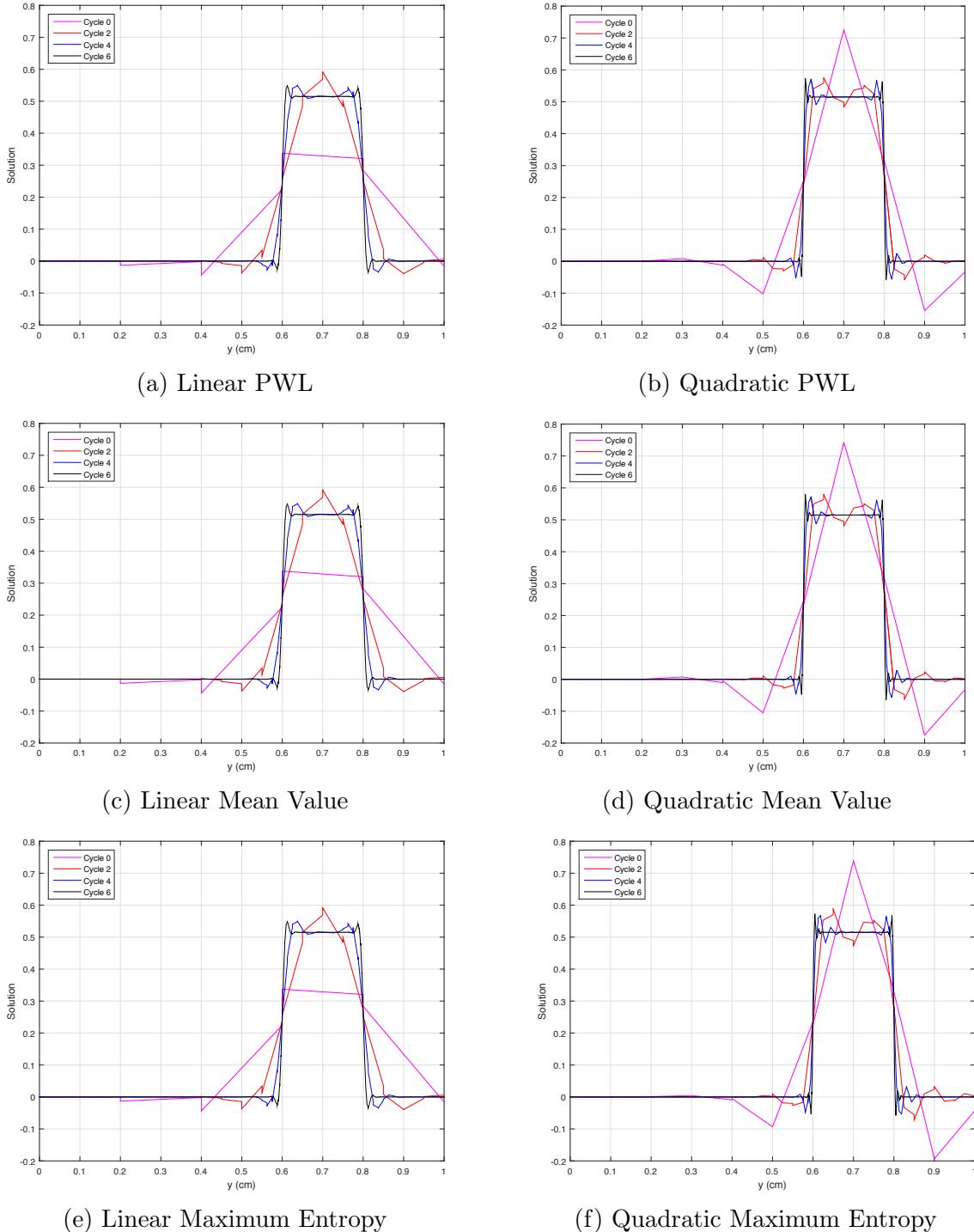


Figure 1.66: Exiting angular flux on the right boundary with uniform refinement.

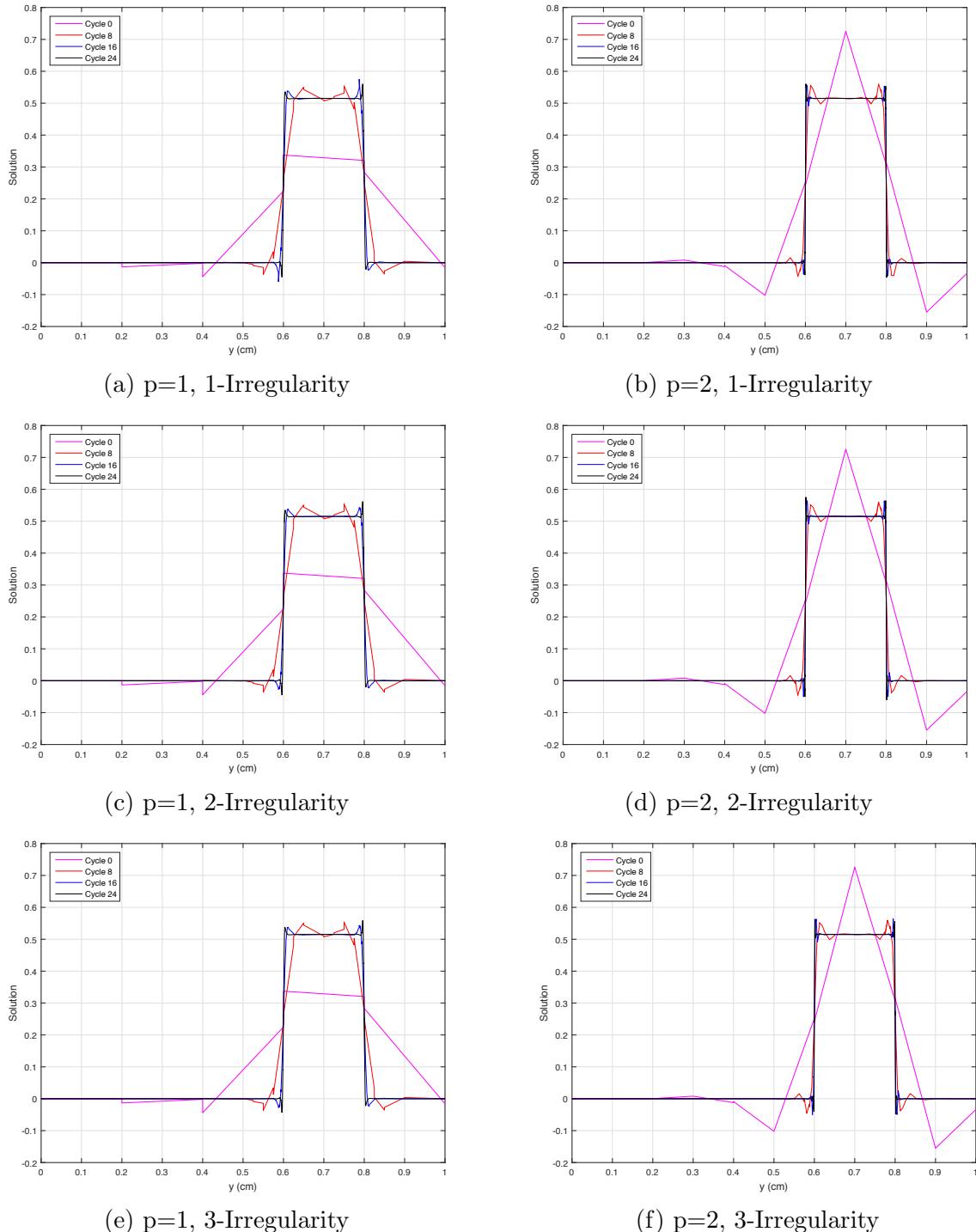


Figure 1.67: Exiting angular flux on the right boundary with AMR and the PWL basis functions with different mesh irregularities.

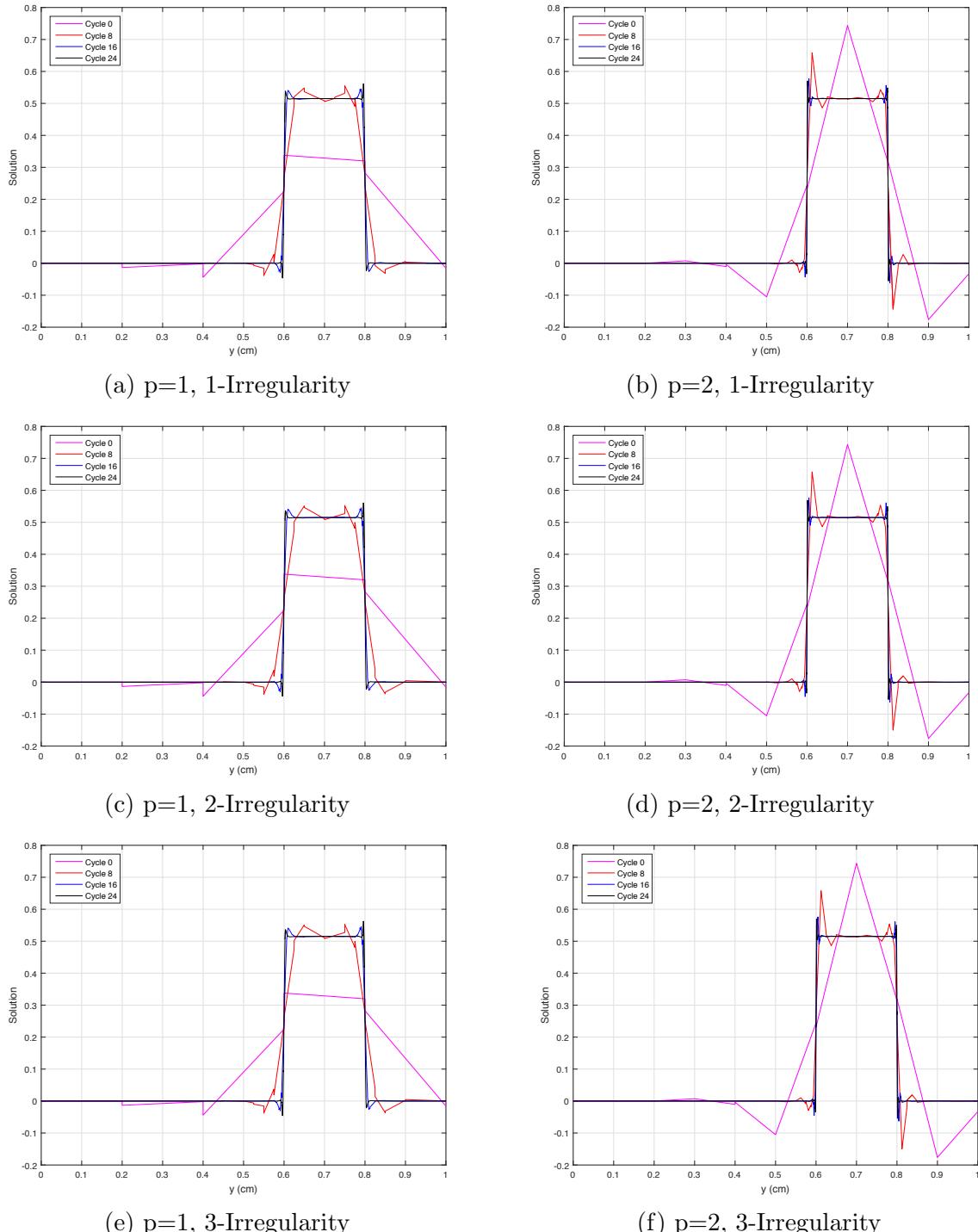


Figure 1.68: Exiting angular flux on the right boundary with AMR and the mean value basis functions with different mesh irregularities.

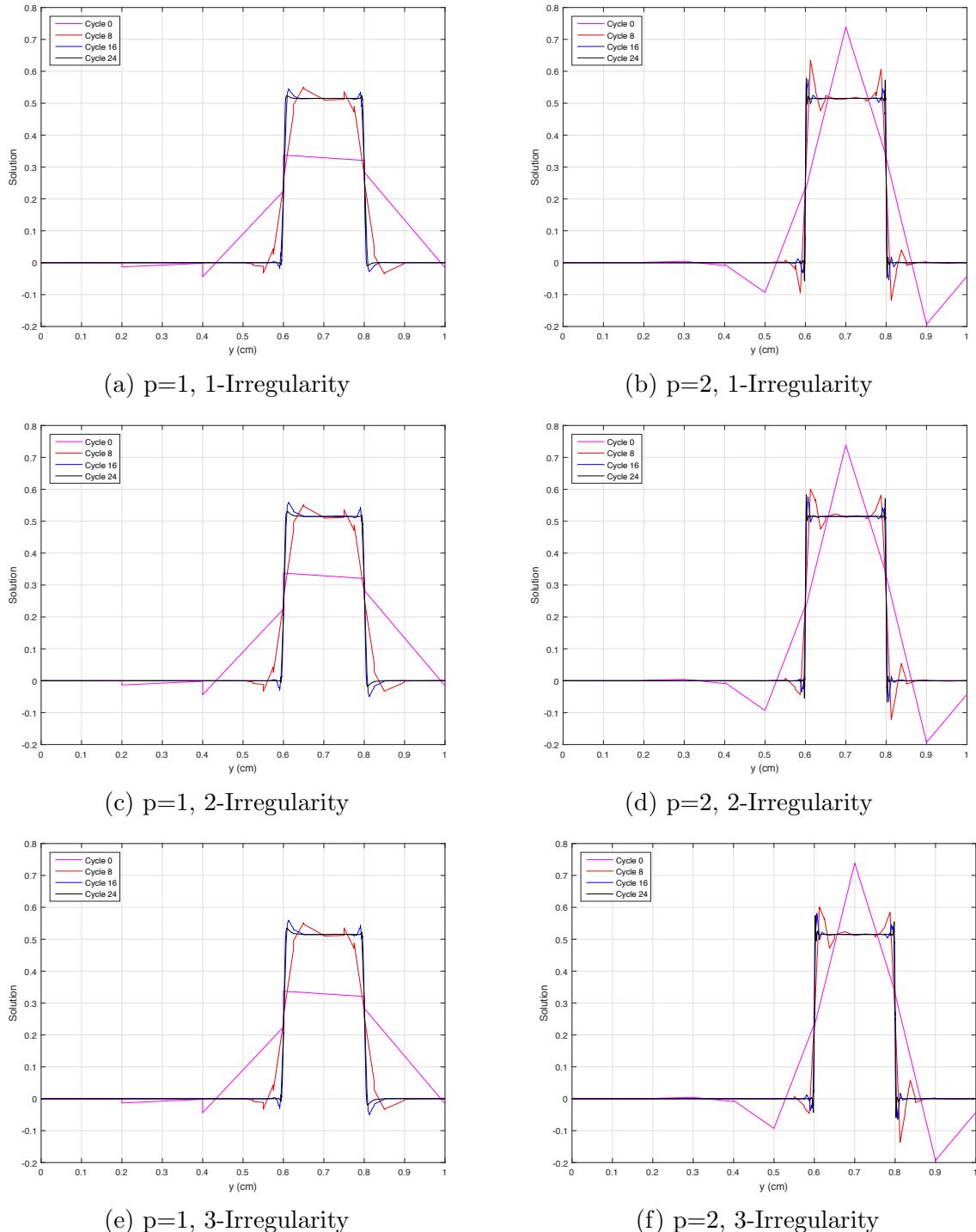


Figure 1.69: Exiting angular flux on the right boundary with AMR and the maximum entropy basis functions with different mesh irregularities.

performed to test the convergence behavior of the basis functions under different conditions. Using MMS, the transport solutions converged at a rate of $p + 1$ under uniform mesh refinement, which is perfectly in alignment with FEM theory. AMR was also used on MMS problems to achieve more accurate solutions with the same $p+1$ convergence rates using less degrees of freedom. Finally, we concluded with some numerical problems involving purely absorbing media containing a solution discontinuity. For these problems, the solutions converge at a rate of $\min(p + 1, r)$ depending on the regularity of the alignment of the transport solution. If the spatial mesh is aligned with the discontinuity, then the $p + 1$ convergence are still observed. If the meshes are not aligned, then convergence rates imposed by the regularity ($r = 1/2$ or $3/2$) are observed for optically thin meshes. However, for meshes that are still optically thick convergence rates of $p + 1$ are observed in the pre-asymptotic region before being restricted by r as the mesh gets optically thin.

REFERENCES

- [1] A. ERN and J.-L. GUERMOND, *Theory and practice of finite elements*, vol. 159, Springer Science & Business Media (2013).
- [2] A. RAND, A. GILLETTE, and C. BAJAJ, “Quadratic serendipity finite elements on polygons using generalized barycentric coordinates,” *Mathematics of computation*, **83**, 290, 2691–2716 (2014).
- [3] E. L. WACHSPRESS, “A Rational Finite Element Basis,” *Mathematics in science and engineering* (1975).
- [4] J. WARREN, S. SCHAEFER, A. N. HIRANI, and M. DESBRUN, “Barycentric coordinates for convex sets,” *Advances in computational mathematics*, **27**, 3, 319–338 (2007).
- [5] H. G. STONE and M. L. ADAMS, “A piecewise linear finite element basis with application to particle transport,” in “Proc. ANS Topical Meeting Nuclear Mathematical and Computational Sciences Meeting,” (2003).
- [6] H. G. STONE and M. L. ADAMS, “New Spatial Discretization Methods for Transport on Unstructured Grids,” in “Proc. ANS Topical Meeting Mathematics and Computation, Supercomputing, Reactor Physics and Biological Applications,” (2005).
- [7] T. S. BAILEY, *The piecewise linear discontinuous finite element method applied to the RZ and XYZ transport equations*, Ph.D. thesis, Texas A&M University (2008).
- [8] M. S. FLOATER, “Mean value coordinates,” *Computer aided geometric design*, **20**, 1, 19–27 (2003).

- [9] M. S. FLOATER, “Generalized barycentric coordinates and applications,” *Acta Numerica*, **24**, 161–214 (2015).
- [10] N. SUKUMAR, “Construction of polygonal interpolants: a maximum entropy approach,” *International Journal for Numerical Methods in Engineering*, **61**, 12, 2159–2181 (2004).
- [11] M. ARROYO and M. ORTIZ, “Local maximum-entropy approximation schemes: a seamless bridge between finite elements and meshfree methods,” *International journal for numerical methods in engineering*, **65**, 13, 2167–2202 (2006).
- [12] K. HORMANN and N. SUKUMAR, “Maximum entropy coordinates for arbitrary polytopes,” in “Computer Graphics Forum,” Wiley Online Library (2008), vol. 27, pp. 1513–1520.
- [13] C. E. SHANNON, “A mathematical theory of communication,” *Bell Systems Technical Journal*, **27**, 379–423 (1948).
- [14] E. T. JAYNES, “Information theory and statistical mechanics,” *Physical review*, **106**, 4, 620–630 (1957).
- [15] S. KULLBACK and R. A. LEIBLER, “On information and sufficiency,” *The annals of mathematical statistics*, pp. 79–86 (1951).
- [16] E. T. JAYNES, “Information theory and statistical mechanics,” in “Statistical Physics,” (1963), vol. 3, pp. 181–218.
- [17] S. BOYD and L. VANDENBERGHE, *Convex optimization*, Cambridge university press (2004).
- [18] R. L. BURDEN and J. D. FAIRES, *Numerical analysis*, Brooks/Cole, 8 ed. (2004).

- [19] M. S. FLOATER, G. KÓS, and M. REIMERS, “Mean value coordinates in 3D,” *Computer Aided Geometric Design*, **22**, 7, 623–631 (2005).
- [20] M. WICKE, M. BOTSCHE, and M. GROSS, “A finite element method on convex polyhedra,” in “Computer Graphics Forum,” Wiley Online Library (2007), vol. 26, pp. 355–364.
- [21] D. GONZÁLEZ, E. CUETO, and M. DOBLARÉ, “A higher order method based on local maximum entropy approximation,” *International journal for numerical methods in engineering*, **83**, 6, 741–764 (2010).
- [22] N. SUKUMAR, “Quadratic maximum-entropy serendipity shape functions for arbitrary planar polygons,” *Computer Methods in Applied Mechanics and Engineering*, **263**, 27–41 (2013).
- [23] R. H. MACNEAL and R. L. HARDER, “Eight nodes or nine?” *International journal for numerical methods in engineering*, **33**, 5, 1049–1058 (1992).
- [24] D. N. ARNOLD and G. AWANOU, “The serendipity family of finite elements,” *Foundations of Computational Mathematics*, **11**, 3, 337–344 (2011).
- [25] R. PENROSE, “A generalized inverse for matrices,” in “Mathematical proceedings of the Cambridge philosophical society,” Cambridge Univ Press (1955), vol. 51, pp. 406–413.
- [26] P. SILVESTER, “Symmetric quadrature formulae for simplexes,” *Mathematics of Computation*, **24**, 109, 95–100 (1970).
- [27] D. DUNAVANT, “High degree efficient symmetrical Gaussian quadrature rules for the triangle,” *International journal for numerical methods in engineering*, **21**, 6, 1129–1148 (1985).

- [28] S. WANDZURAT and H. XIAO, “Symmetric quadrature rules on a triangle,” *Computers & Mathematics with Applications*, **45**, 12, 1829–1840 (2003).
- [29] J. N. LYNESS and R. COOLS, “A survey of numerical cubature over triangles,” in “Proceedings of Symposia in Applied Mathematics,” (1994), vol. 48, pp. 127–150.
- [30] R. COOLS and A. HAEGEMANS, “Construction of minimal cubature formulae for the square and the triangle, using invariant theory,” (1987).
- [31] M. NOOIJEN, G. TE VELDE, and E. BAERENDS, “Symmetric numerical integration formulas for regular polygons,” *SIAM journal on numerical analysis*, **27**, 1, 198–218 (1990).
- [32] G. DASGUPTA, “Integration within polygonal finite elements,” *Journal of Aerospace Engineering*, **16**, 1, 9–18 (2003).
- [33] S. MOUSAVI, H. XIAO, and N. SUKUMAR, “Generalized Gaussian quadrature rules on arbitrary polygons,” *International Journal for Numerical Methods in Engineering*, **82**, 1, 99–113 (2010).
- [34] M. L. ADAMS, “Discontinuous finite element transport solutions in thick diffusive problems,” *Nuclear science and engineering*, **137**, 3, 298–333 (2001).
- [35] A. SHESTAKOV, J. HARTE, and D. KERSHAW, “Solution of the diffusion equation by finite elements in lagrangian hydrodynamic codes,” *Journal of Computational Physics*, **76**, 2, 385–413 (1988).
- [36] A. SHESTAKOV, D. KERSHAW, and G. ZIMMERMAN, “Test problems in radiative transfer calculations,” *Nuclear science and engineering*, **105**, 1, 88–104 (1990).

- [37] D. S. KERSHAW, “Differencing of the diffusion equation in Lagrangian hydrodynamic codes,” *Journal of Computational Physics*, **39**, 2, 375–395 (1981).
- [38] A. DEDNER and P. VOLLMÖLLER, “An adaptive higher order method for solving the radiation transport equation on unstructured grids,” *Journal of Computational Physics*, **178**, 2, 263–289 (2002).
- [39] Y. WANG and J. C. RAGUSA, “Standard and goal-oriented adaptive mesh refinement applied to radiation transport on 2D unstructured triangular meshes,” *Journal of Computational Physics*, **230**, 3, 763–788 (2011).