

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.

The remainder of the 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 1.4 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: $([0, \pi])$. 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} .

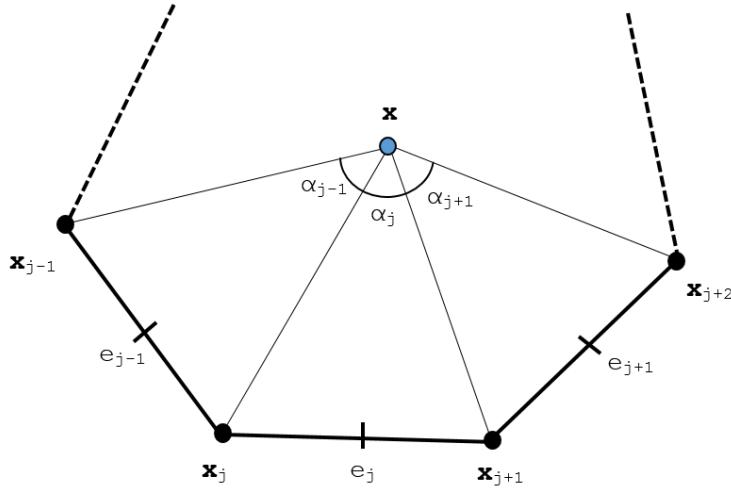


Figure 1.1: Arbitrary polygon with geometric properties used for 2D basis function generation.

In this dissertation, all 1st-order, two-dimensional basis functions for an element K will obey the properties for barycentric coordinates. They will form a *partition of unity*,

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

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

$$\sum_{i=1}^{N_K} b_i(\vec{x}) \vec{x}_i = \vec{x}; \quad (1.2)$$

and they will satisfy the *Lagrange property*,

$$b_i(\vec{x}_j) = \delta_{ij}. \quad (1.3)$$

N_K is again the number of spatial degrees with measure in element K . 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

$$\sum_{i=1}^{N_K} b_i(\vec{x}) \vec{c}_{i,1}(\vec{x}) = \vec{q}_1, \quad (1.4)$$

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

$$\vec{c}_{i,1}(\vec{x}) = \begin{bmatrix} 1 \\ x_i - x \\ y_i - y \end{bmatrix} \quad \text{and} \quad \vec{q}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad (1.5)$$

respectively. Equation (1.4) 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 [1]. These rational functions were the first derived for

2D polygons and possess all the properties of the barycentric coordinates previously detailed.

$$b_j^W(\vec{x}) = \frac{w_j(\vec{x})}{\sum_i w_i(\vec{x})} \quad (1.6)$$

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

$$w_j(\vec{x}) = \frac{A(\vec{x}_{j-1}, \vec{x}_j, \vec{x}_{j+1})}{A(\vec{x}, \vec{x}_{j-1}, \vec{x}_j) A(\vec{x}, \vec{x}_j, \vec{x}_{j+1})} \quad (1.7)$$

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 [2, 3]. The originally introduced the work specifically for

$$b_j^{PWL}(x, y) = t_j(x, y) + \alpha_j^K t_c(x, y) \quad (1.8)$$

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 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 .

1.1.3 Mean Value Basis Functions

At this point, we now introduce the first new polygonal basis set for use with the transport equation: the *mean value coordinates* (MV) developed by Floater [4, 5]. The 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.

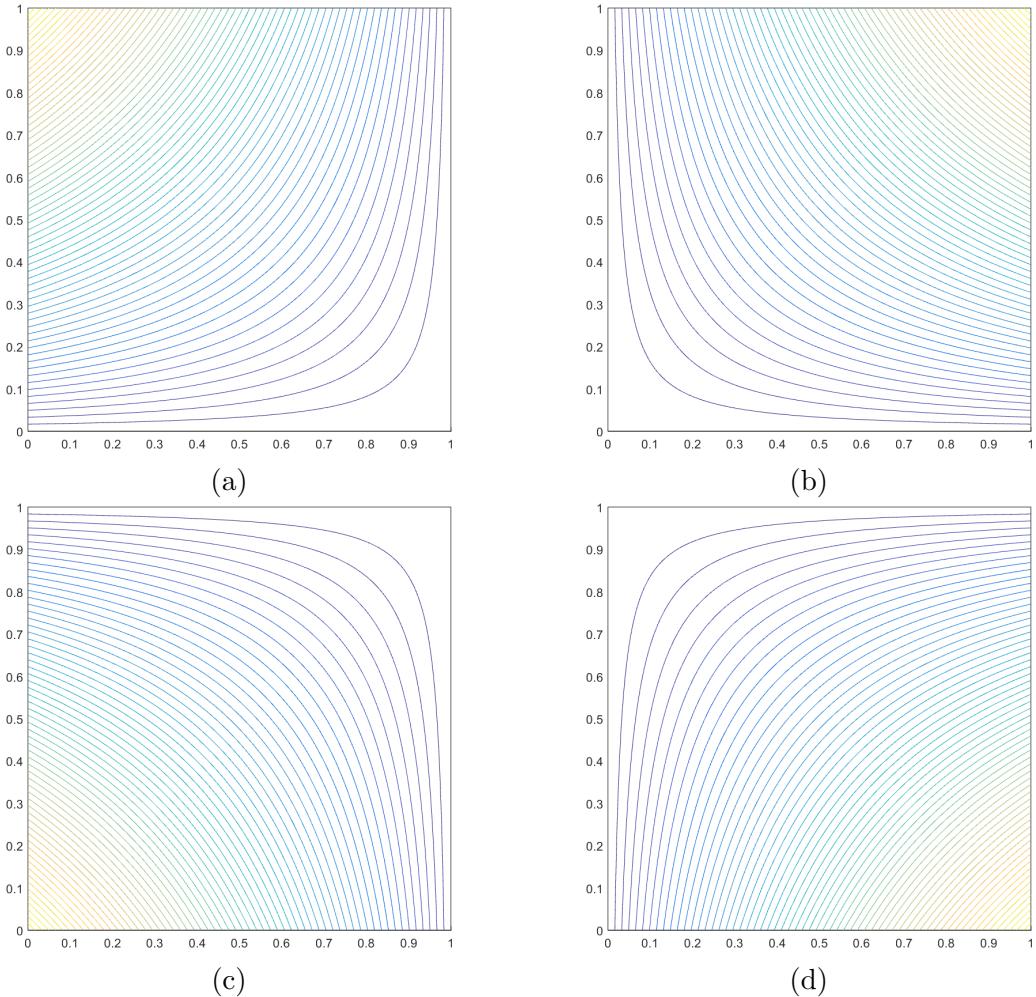


Figure 1.2: Contour plots of the linear Wachspress basis functions on the unit square for the vertices located at: (a) $(0,1)$, (b) $(1,1)$, (c) $(0,0)$, and (d) $(1,0)$.

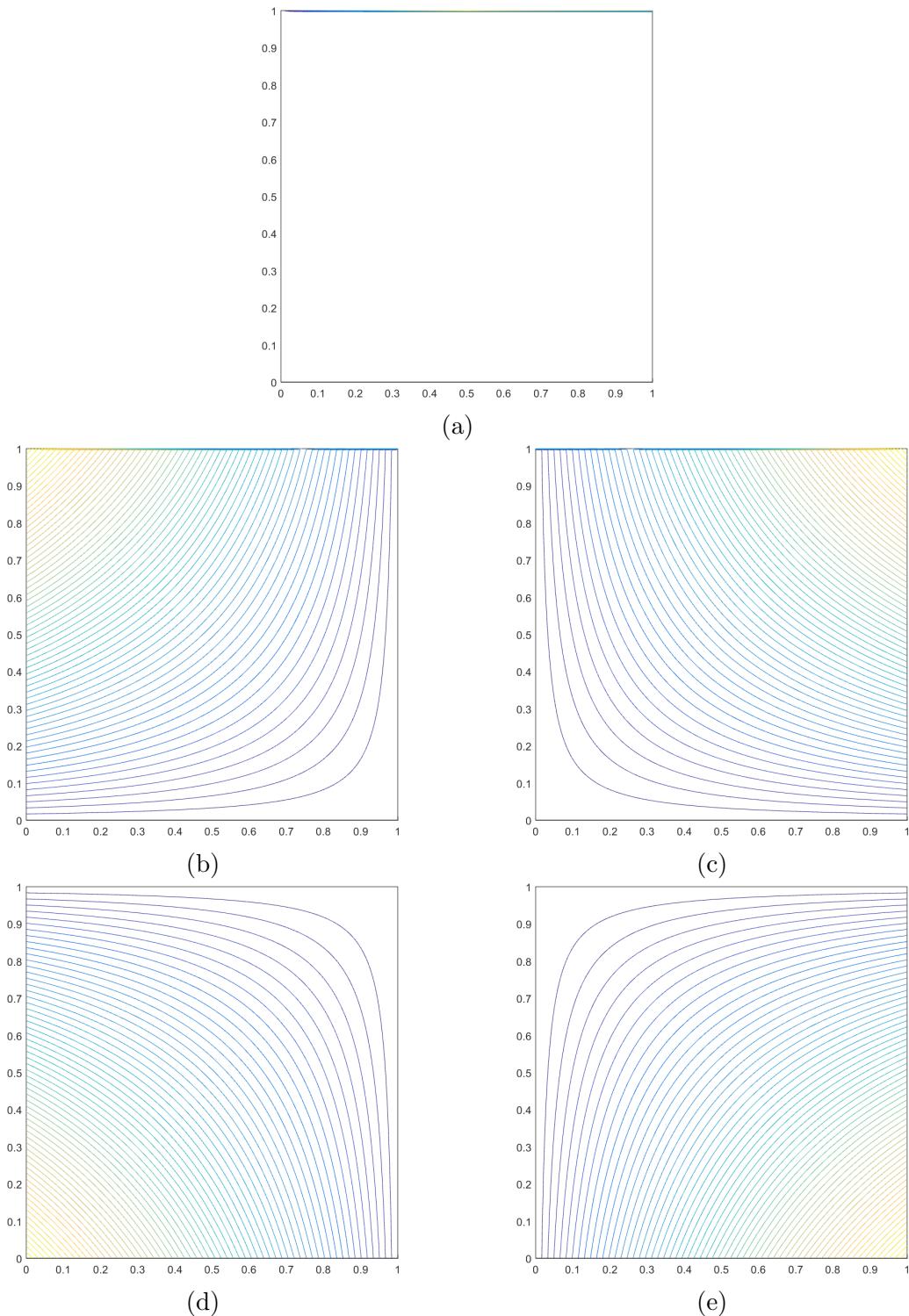


Figure 1.3: Contour plots of the linear Wachspress basis functions on the degenerate pentagon for the vertices located at: (a) $(1/2, 1)$, (b) $(0, 1)$, (c) $(1, 1)$, (d) $(0, 0)$, and (e) $(1, 0)$.

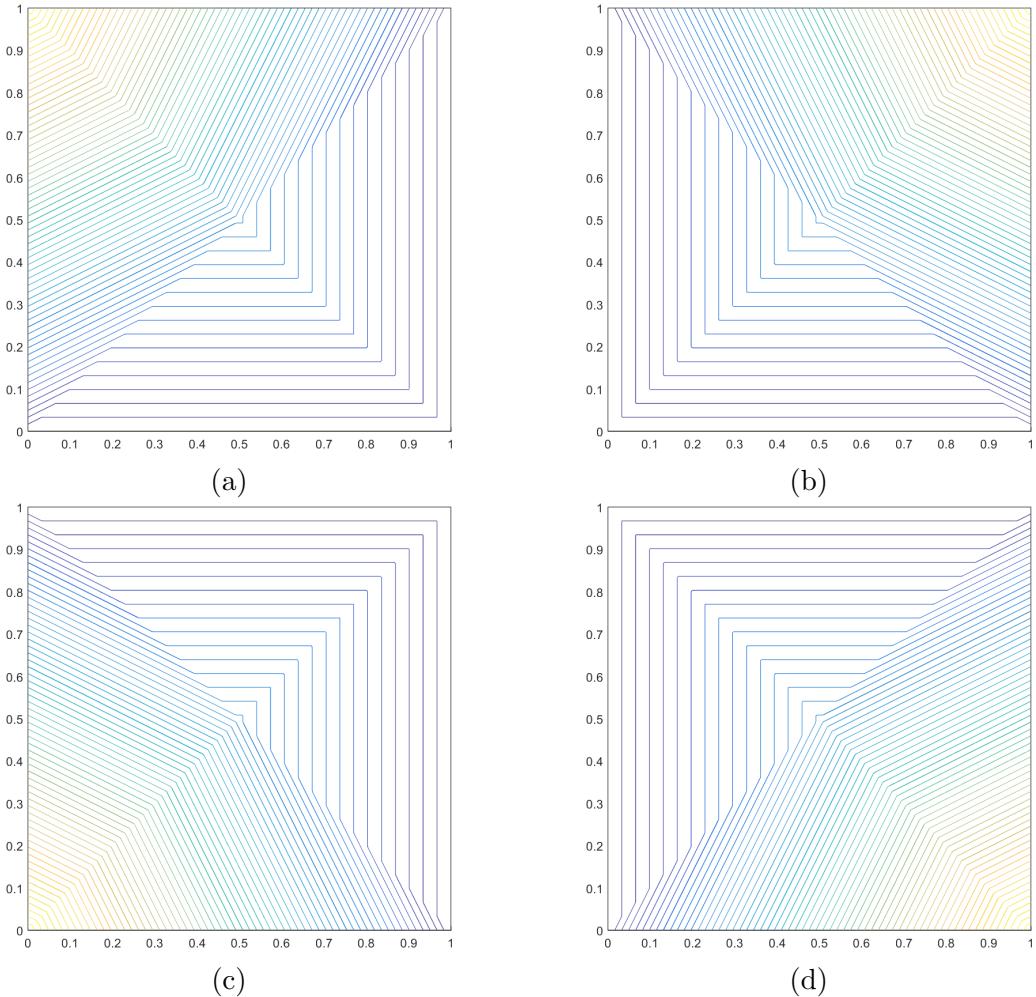


Figure 1.4: Contour plots of the linear PWL basis functions on the unit square for the vertices located at: (a) $(0,1)$, (b) $(1,1)$, (c) $(0,0)$, and (d) $(1,0)$.

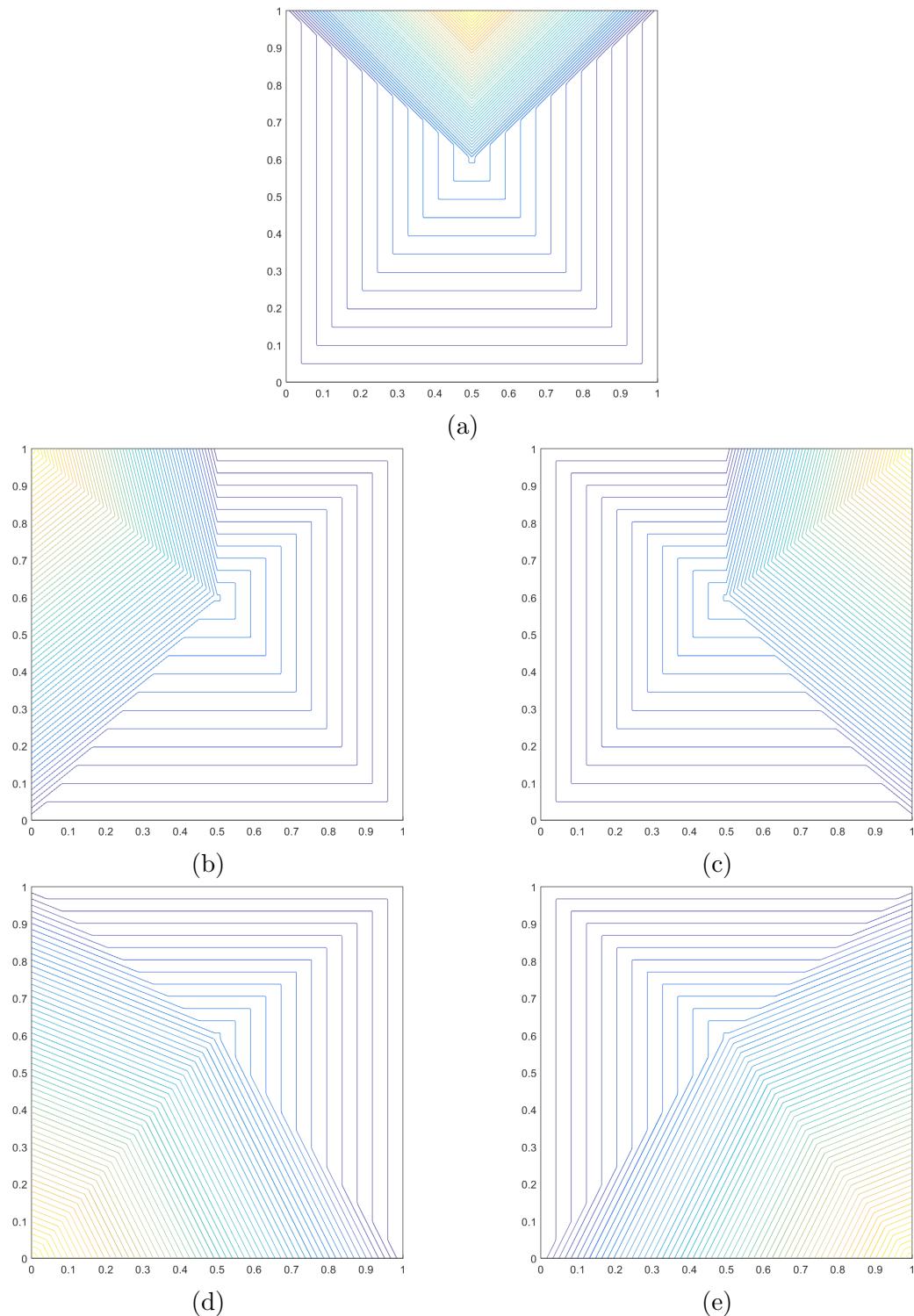


Figure 1.5: Contour plots of the linear PWL basis functions on the degenerate pentagon for the vertices located at: (a) $(1/2, 1)$, (b) $(0, 1)$, (c) $(1, 1)$, (d) $(0, 0)$, and (e) $(1, 0)$.

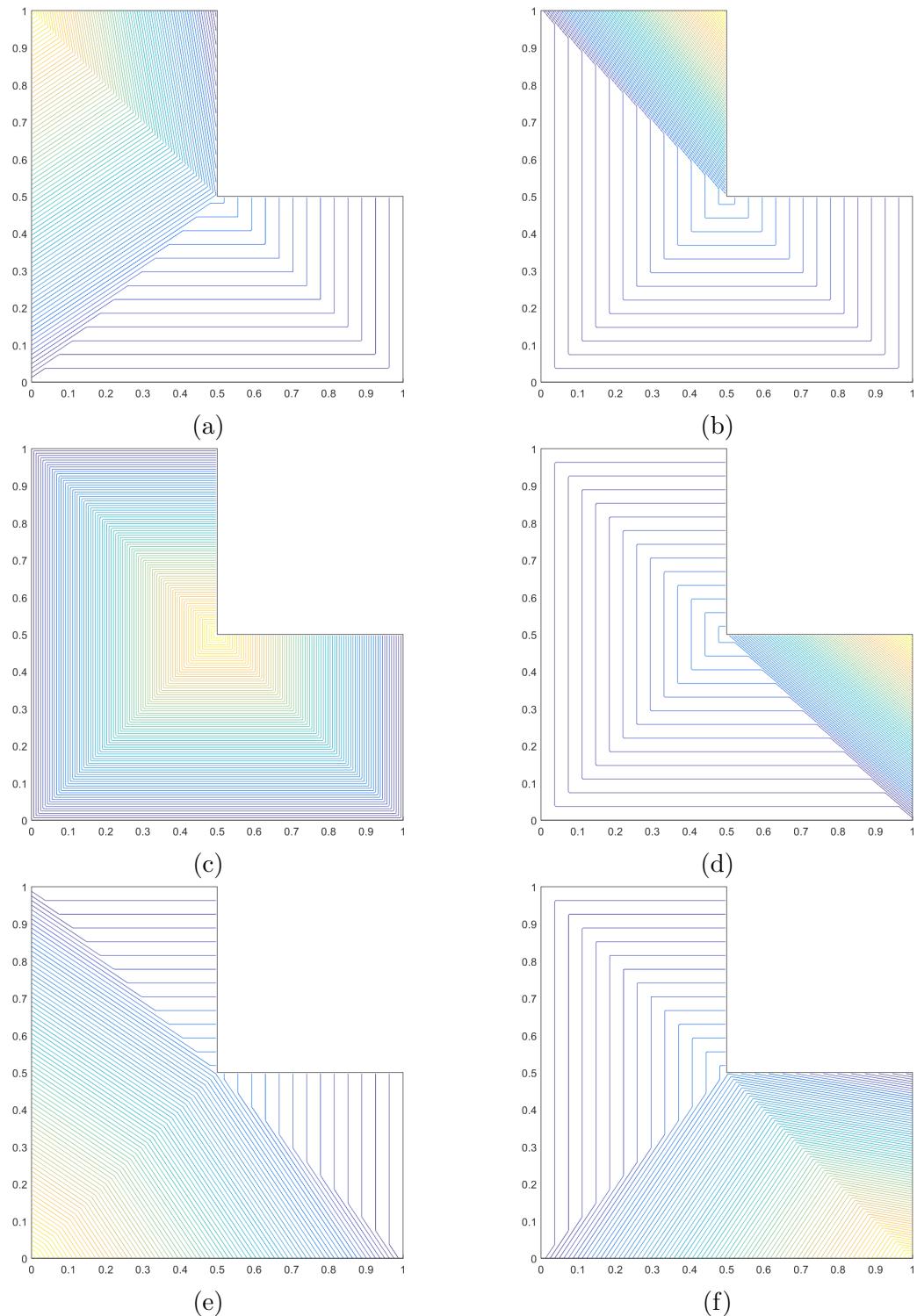


Figure 1.6: Contour plots of the linear PWL basis functions on the L-shaped domain for the vertices located at: (a) $(0,1)$, (b) $(1/2,1)$, (c) $(1/2,1/2)$, (d) $(1,1/2)$, (e) $(0,0)$, and (f) $(1,0)$.

$$\nabla^2 u = 0, \quad (1.9)$$

with $u(\vec{r}) = u_0$ constituting a piecewise linear function

$$b_j^{MV}(\vec{x}) = \frac{w_j(\vec{x})}{\sum_i w_i(\vec{x})} \quad (1.10)$$

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

$$w_j(\vec{x}) = \frac{\tan(\alpha_{j-1}/2) + \tan(\alpha_j/2)}{|\vec{x}_j - \vec{x}|} \quad (1.11)$$

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) [6, 7, 8].

$$b_j^{ME}(\vec{x}) = \frac{w_j(\vec{x})}{\sum_i w_i(\vec{x})}. \quad (1.12)$$

where the maximum entropy weight function for vertex j , w_j , has the following definition,

$$w_j(\vec{x}) = m_j(\vec{x}) \exp(-\kappa \cdot (\vec{x}_j - \vec{x})), \quad (1.13)$$

where κ is a vector value of dimension d that will be explained shortly. In Eq. (1.13), m_j is called the prior distribution and is a key component of Bayesian inference [9, 10]. In the context of Eq. (1.13), 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.

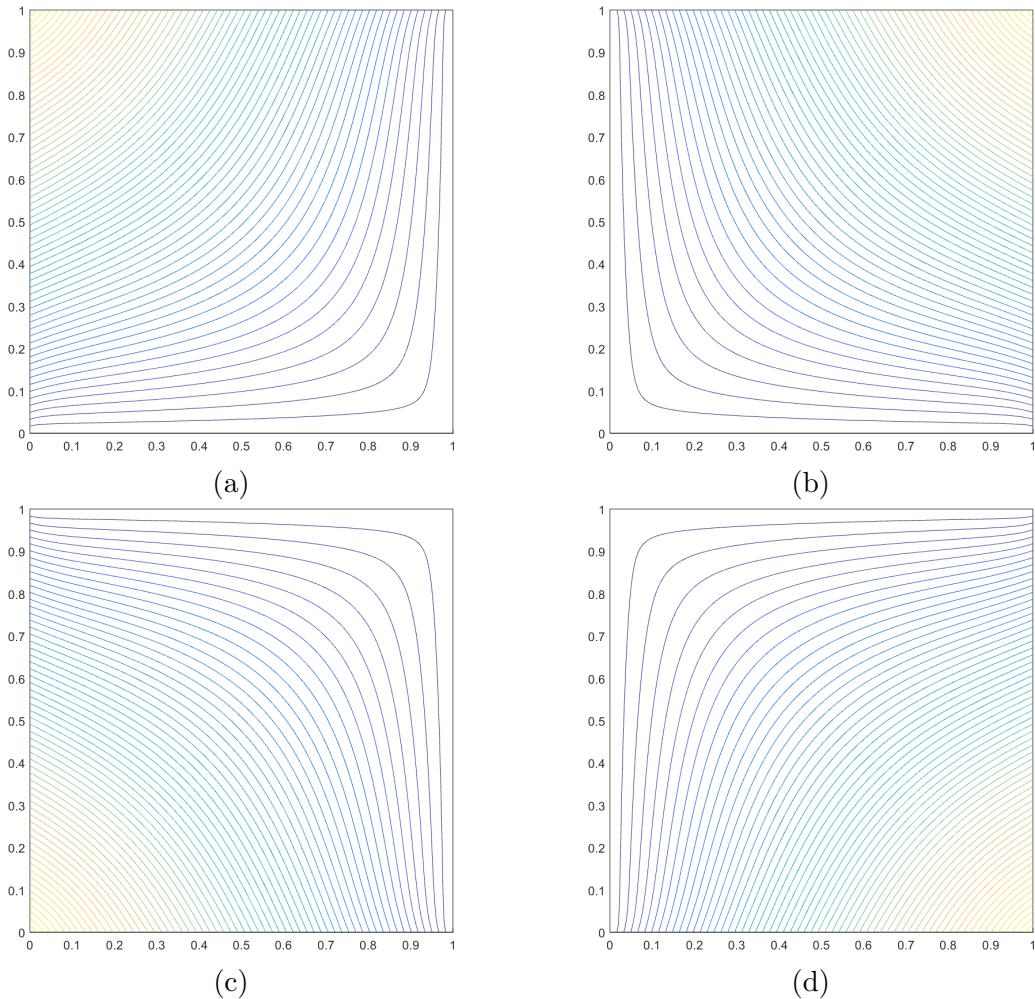


Figure 1.7: Contour plots of the linear mean value basis functions on the unit square for the vertices located at: (a) $(0,1)$, (b) $(1,1)$, (c) $(0,0)$, and (d) $(1,0)$.

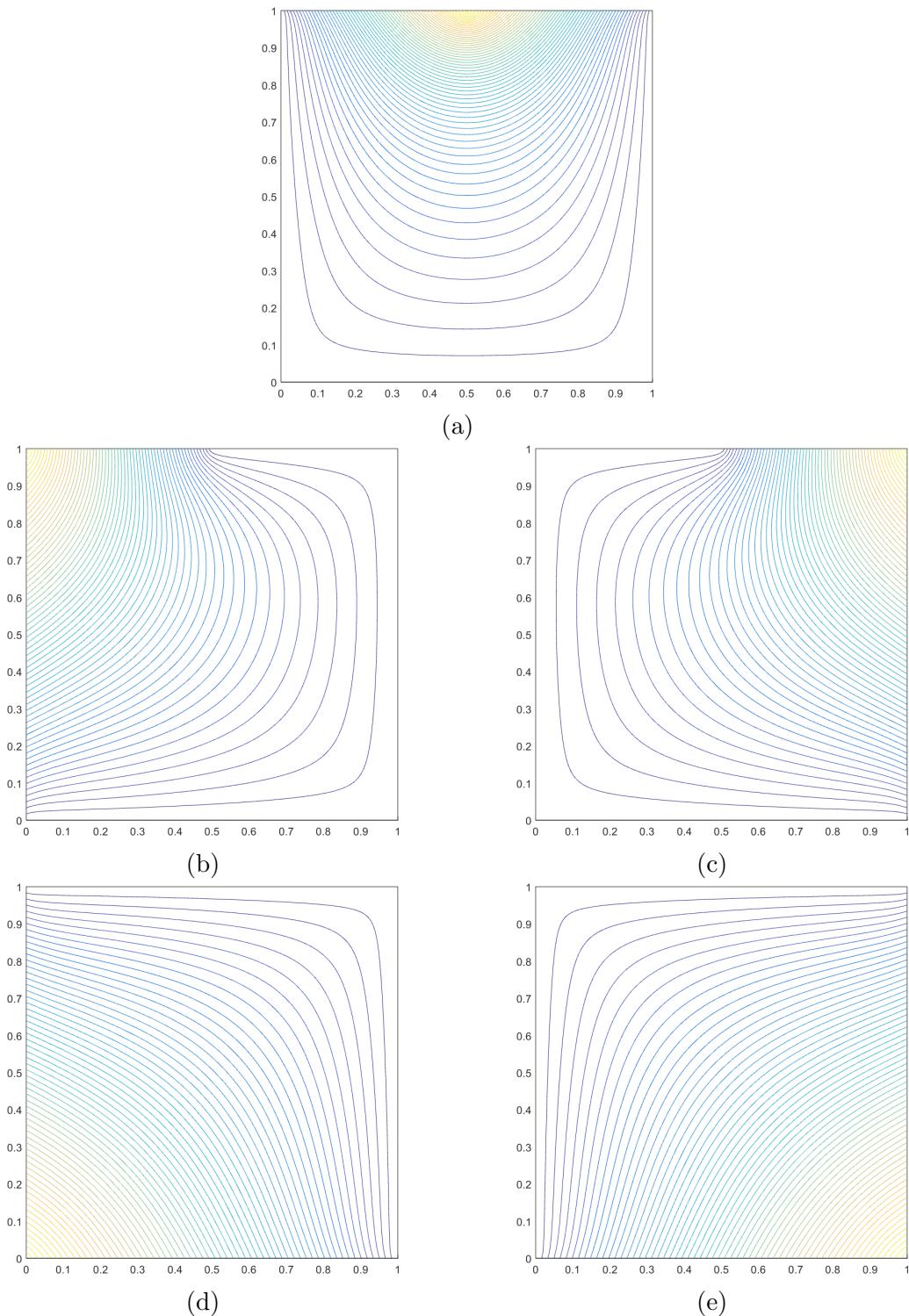


Figure 1.8: Contour plots of the linear mean value basis functions on the degenerate pentagon for the vertices located at: (a) $(1/2, 1)$, (b) $(0, 1)$, (c) $(1, 1)$, (d) $(0, 0)$, and (e) $(1, 0)$.

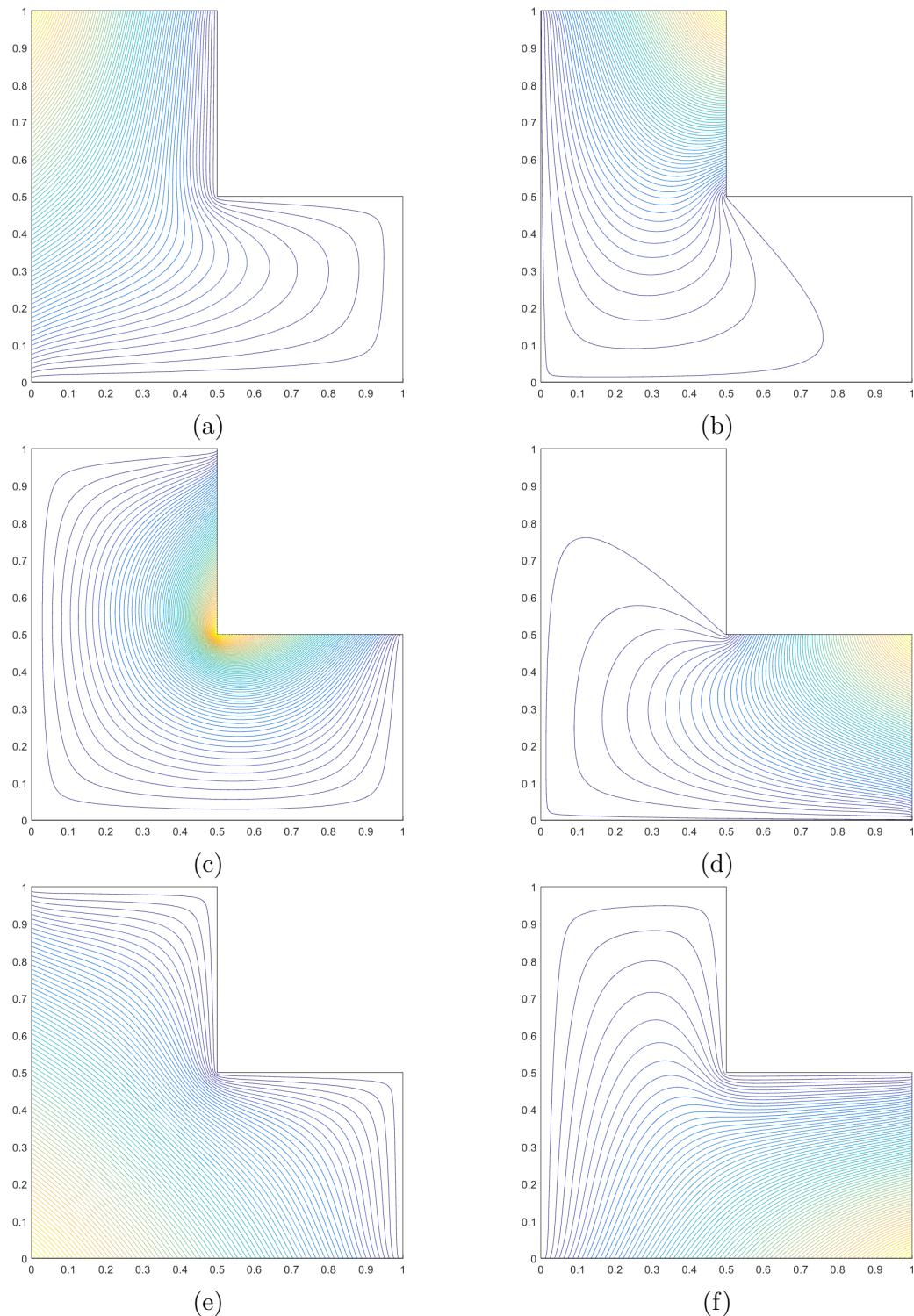


Figure 1.9: Contour plots of the linear mean value basis functions on the L-shaped domain for the vertices located at: (a) $(0,1)$, (b) $(1/2,1)$, (c) $(1/2,1/2)$, (d) $(1,1/2)$, (e) $(0,0)$, and (f) $(1,0)$.

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

Basis Function	Dimension	Polytope Types	Integration	Direct/Iterative
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

$$m_j(\vec{x}) = \frac{\pi_j(\vec{x})}{\sum_k \pi_k(\vec{x})} \quad (1.14)$$

where

$$\pi_j(\vec{x}) = \prod_{i \neq j-1, j} \rho_j(\vec{x}) \quad (1.15)$$

where

$$\rho_j(\vec{x}) = ||\vec{x} - \vec{x}_j|| + ||\vec{x} - \vec{x}_{j+1}|| - ||\vec{x}_{j+1} - \vec{x}_j|| \quad (1.16)$$

1.1.5 Summary of 2D Linear Basis Functions on Polygons

1.2 Converting the Linear Polygonal Basis Functions to the Quadratic Serendipity

Space of Functions

Now that we have given complete details on the linearly-complete generalized barycentric coordinates that we will investigate for this work, we

for the constant constraint,

$$\sum_{a=1}^{N_K} \sum_{b=1}^{N_K} \mu_{ab}(\vec{x}) = 1, \quad (1.17)$$

for the linear constraint,

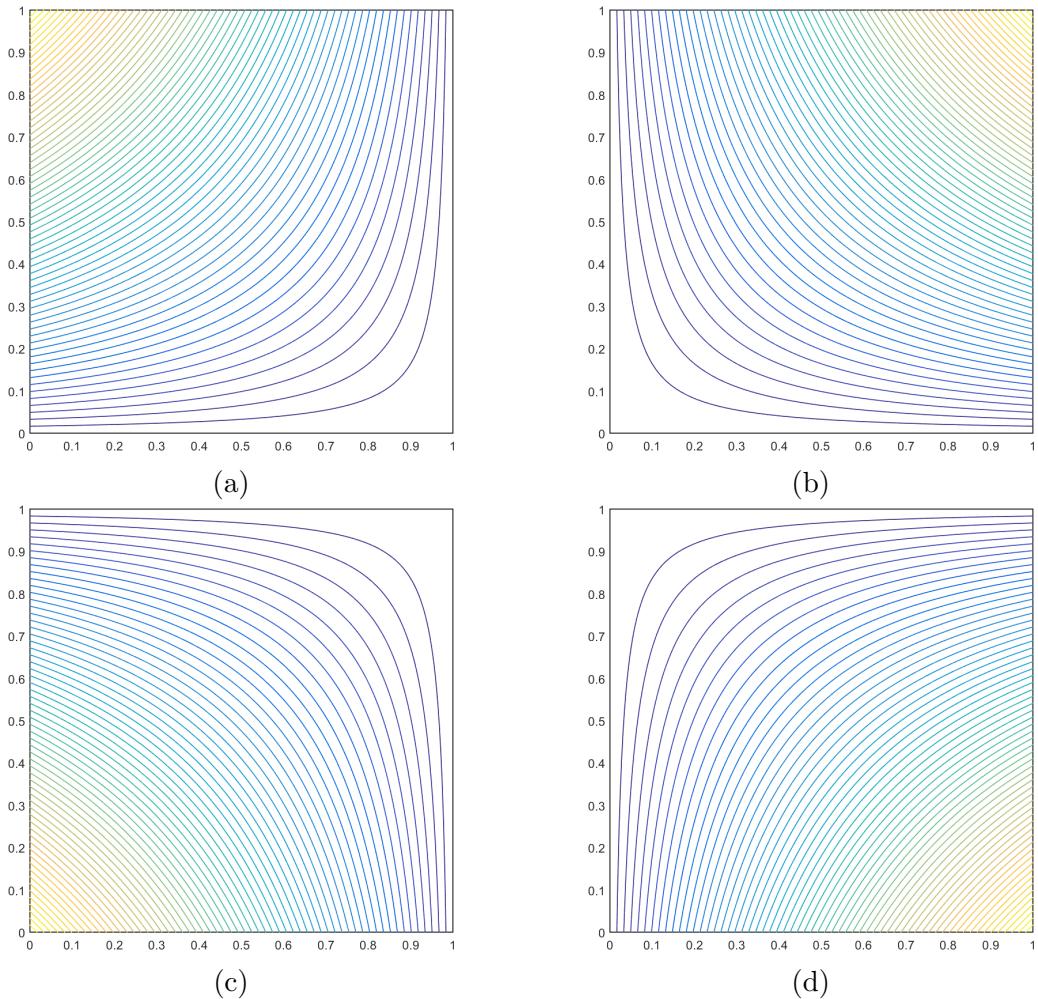


Figure 1.10: Contour plots of the linear maximum entropy basis functions on the unit square for the vertices located at: (a) $(0,1)$, (b) $(1,1)$, (c) $(0,0)$, and (d) $(1,0)$.

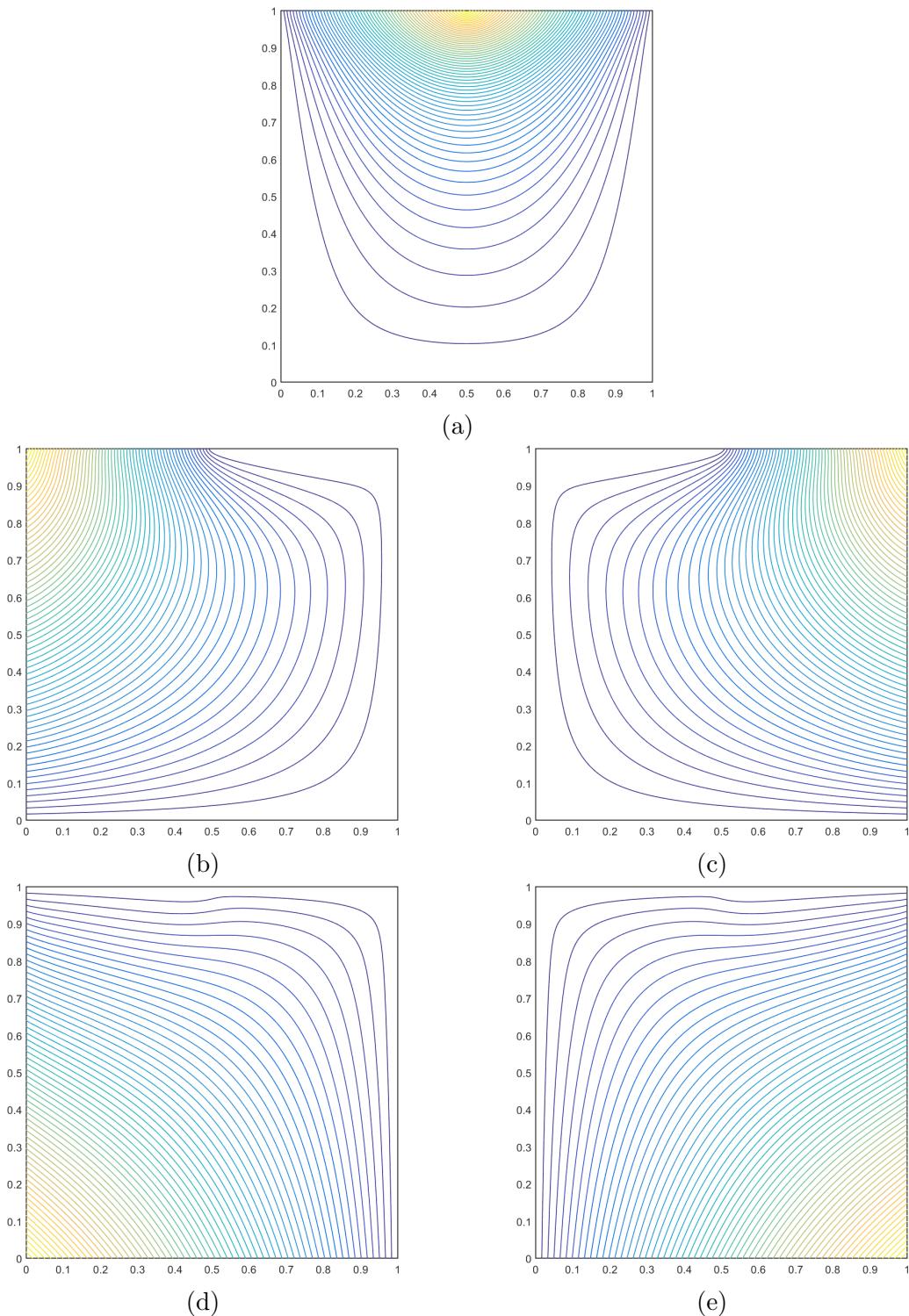


Figure 1.11: Contour plots of the linear maximum entropy basis functions on the degenerate pentagon for the vertices located at: (a) $(1/2, 1)$, (b) $(0, 1)$, (c) $(1, 1)$, (d) $(0, 0)$, and (e) $(1, 0)$.

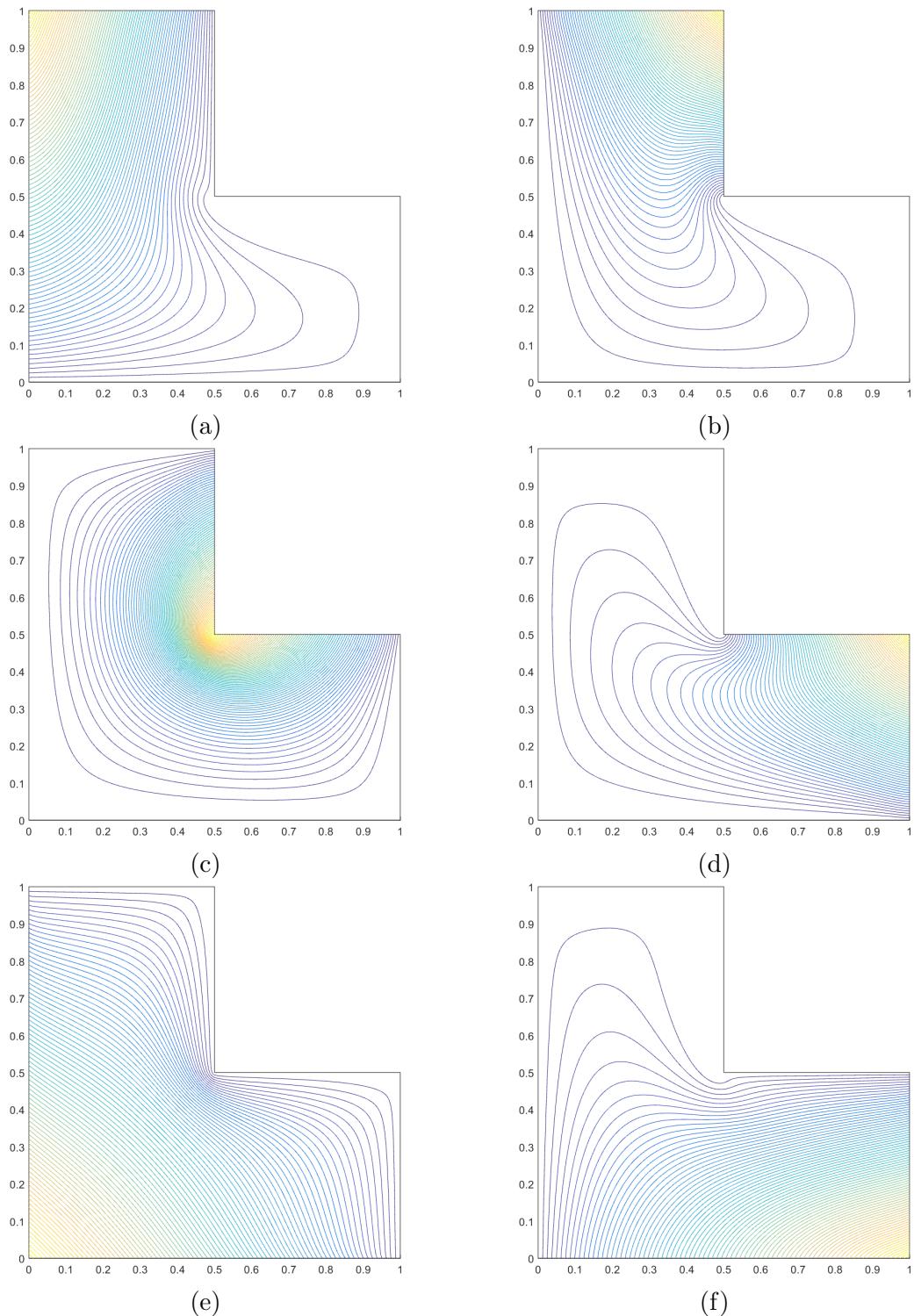
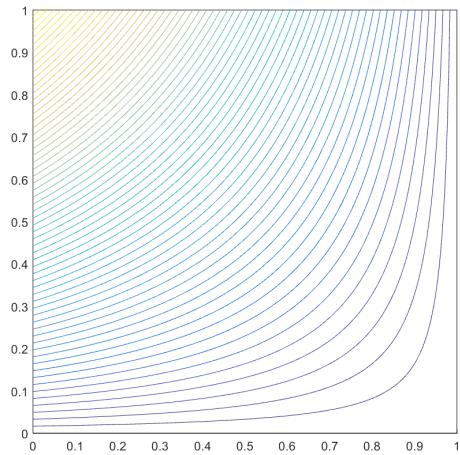
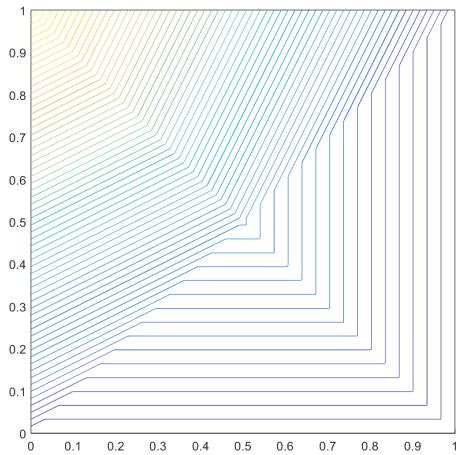


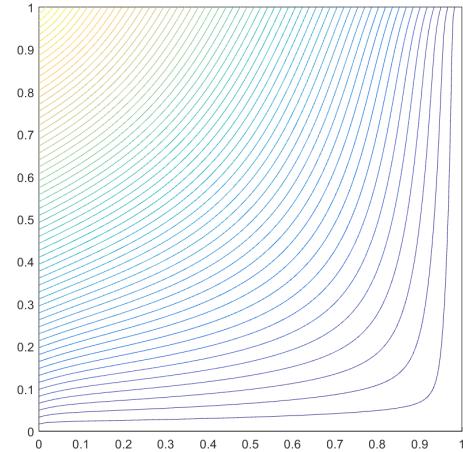
Figure 1.12: Contour plots of the linear maximum entropy basis functions on the L-shaped domain for the vertices located at: (a) $(0,1)$, (b) $(1/2,1)$, (c) $(1/2,1/2)$, (d) $(1,1/2)$, (e) $(0,0)$, and (f) $(1,0)$.



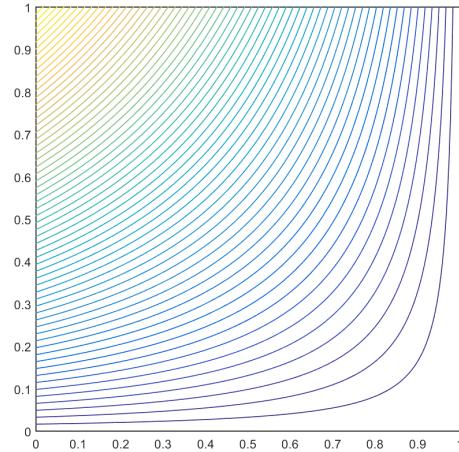
(a) Wachspress



(b) PWL

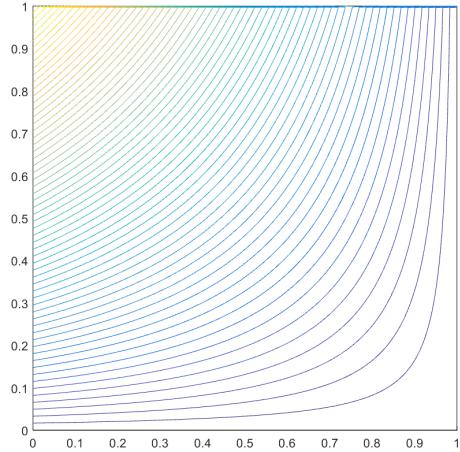


(c) mean value

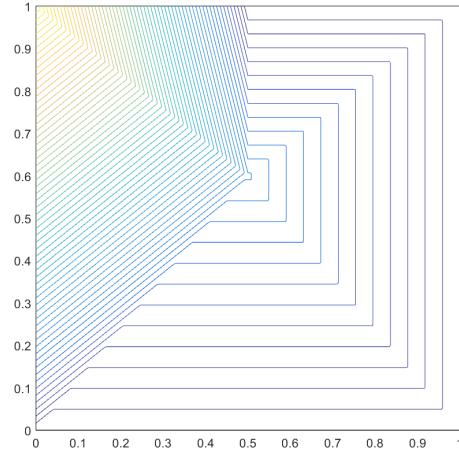


(d) maximum entropy

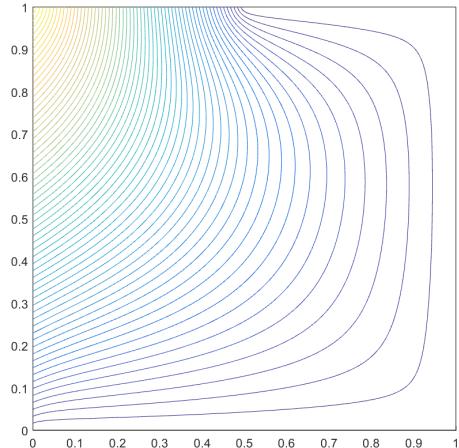
Figure 1.13: Contour plots of the different linear basis function on the unit square located at vertex $(0,1)$.



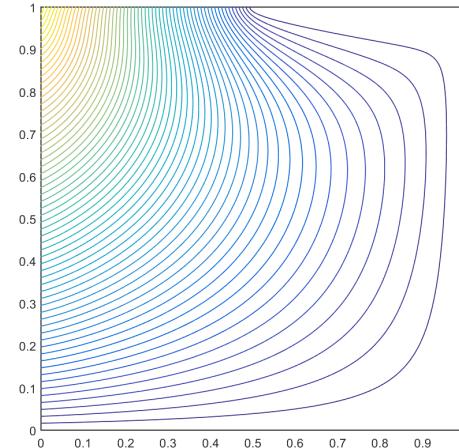
(a) Wachspress



(b) PWL



(c) Mean Value



(d) Maximum Entropy

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.

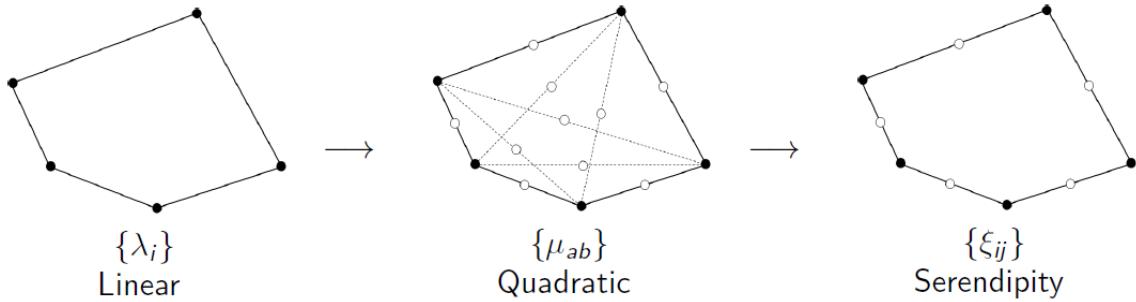


Figure 1.15: Overview of the process to construct the quadratic serendipity basis functions on polygons. The filled dots correspond to basis functions that maintain the Lagrange property while empty dots do not.

$$\sum_{a=1}^{N_K} \sum_{b=1}^{N_K} \mu_{ab}(\vec{x}) \vec{x}_a = \vec{x}, \quad (1.18)$$

and for the quadratic constraint,

$$\sum_{a=1}^{N_K} \sum_{b=1}^{N_K} \mu_{ab}(\vec{x}) (\vec{x}_a \otimes \vec{x}_b) = \vec{x} \otimes \vec{x}. \quad (1.19)$$

$$\vec{x}_{ab} = \frac{\vec{x}_a + \vec{x}_b}{2} \quad (1.20)$$

With this definition

for the constant constraint,

$$\sum_{aa \in V} \mu_{aa}(\vec{x}) + \sum_{ab \in E \cup D} 2\mu_{ab}(\vec{x}) = 1, \quad (1.21)$$

for the linear constraint,

$$\sum_{aa \in V} \mu_{aa}(\vec{x}) \vec{x}_{aa} + \sum_{ab \in E \cup D} 2\mu_{ab}(\vec{x}) \vec{x}_{ab} = \vec{x}, \quad (1.22)$$

and for the quadratic constraint,

$$\sum_{aa \in V} \mu_{aa}(\vec{x}) (\vec{x}_a \otimes \vec{x}_a) + \sum_{ab \in E \cup D} \mu_{ab}(\vec{x}) (\vec{x}_a \otimes \vec{x}_b + \vec{x}_b \otimes \vec{x}_a) = \vec{x} \otimes \vec{x}. \quad (1.23)$$

for the constant constraint,

$$\sum_{ii \in V} \xi_{ii}(\vec{x}) + \sum_{i(i+1) \in E} 2\xi_{i(i+1)}(\vec{x}) = 1, \quad (1.24)$$

for the linear constraint,

$$\sum_{ii \in V} \xi_{ii}(\vec{x}) \vec{x}_{ii} + \sum_{i(i+1) \in E} 2\xi_{i(i+1)}(\vec{x}) \vec{x}_{i(i+1)} = \vec{x}, \quad (1.25)$$

and for the quadratic constraint,

$$\sum_{ii \in V} \xi_{ii}(\vec{x}) (\vec{x}_i \otimes \vec{x}_i) + \sum_{i(i+1) \in E} \xi_{i(i+1)}(\vec{x}) (\vec{x}_i \otimes \vec{x}_{i+1} + \vec{x}_{i+1} \otimes \vec{x}_i) = \vec{x} \otimes \vec{x}. \quad (1.26)$$

$$\{\xi\} = \mathbb{A}\{\mu\} \quad (1.27)$$

$$\mathbb{A} = \begin{bmatrix} c_{11}^{11} & \dots & c_{ab}^{11} & \dots & c_{(n-2)n}^{11} \\ \dots & \ddots & \vdots & \ddots & \vdots \\ c_{11}^{ij} & \dots & c_{ab}^{ij} & \dots & c_{(n-2)n}^{ij} \\ \dots & \ddots & \vdots & \ddots & \vdots \\ c_{11}^{n(n+1)} & \dots & c_{ab}^{n(n+1)} & \dots & c_{(n-2)n}^{n(n+1)} \end{bmatrix} \quad (1.28)$$

$$\mathbb{A} = [\mathbb{I} | \mathbb{A}'] \quad (1.29)$$

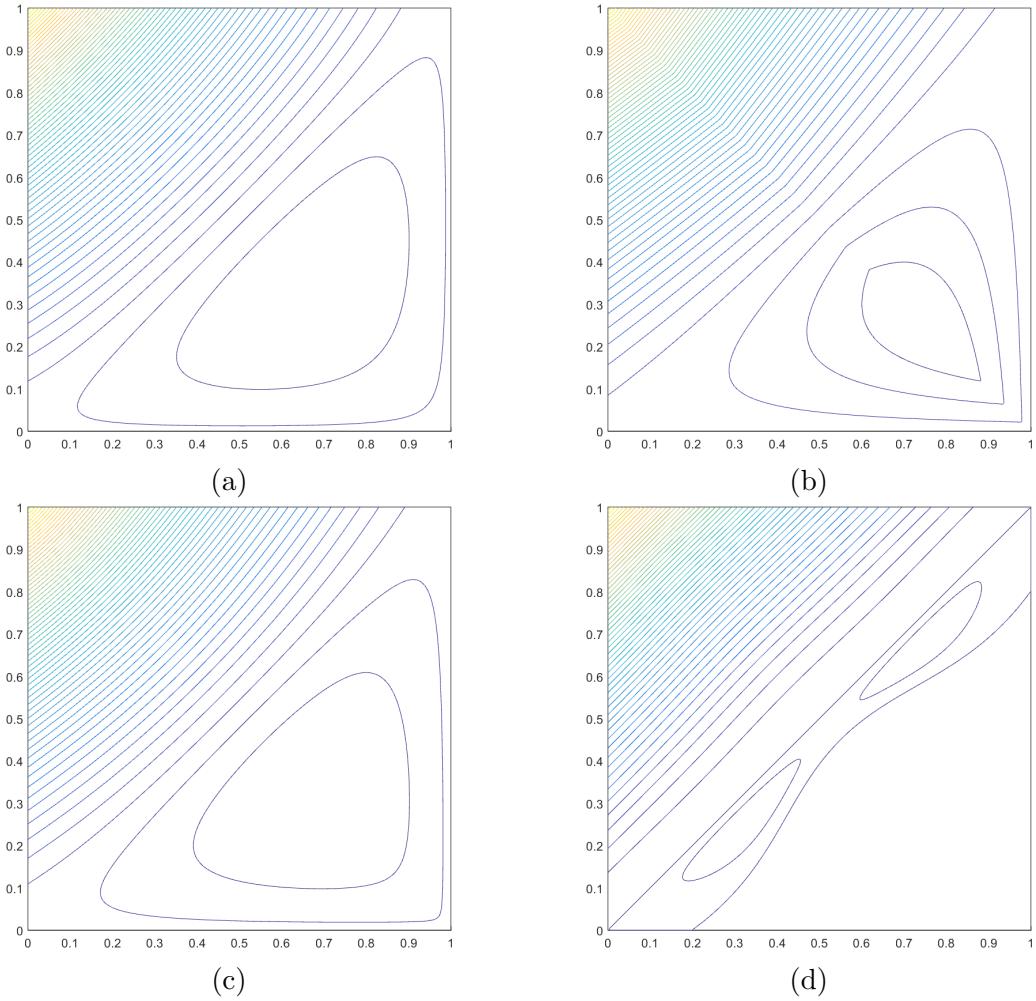


Figure 1.16: Contour plots of the different quadratic serendipity basis function on the unit square located at vertex $(0,1)$: (a) Wachspress, (b) PWL, (c) mean value, and (d) maximum entropy.

where \mathbb{I} is the $(2N_K \times 2N_K)$ identity matrix, and \mathbb{A}' is a full $(2N_K \times N_Q)$ 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

$$B^* = B^T (BB^T)^{-1} \quad (1.30)$$

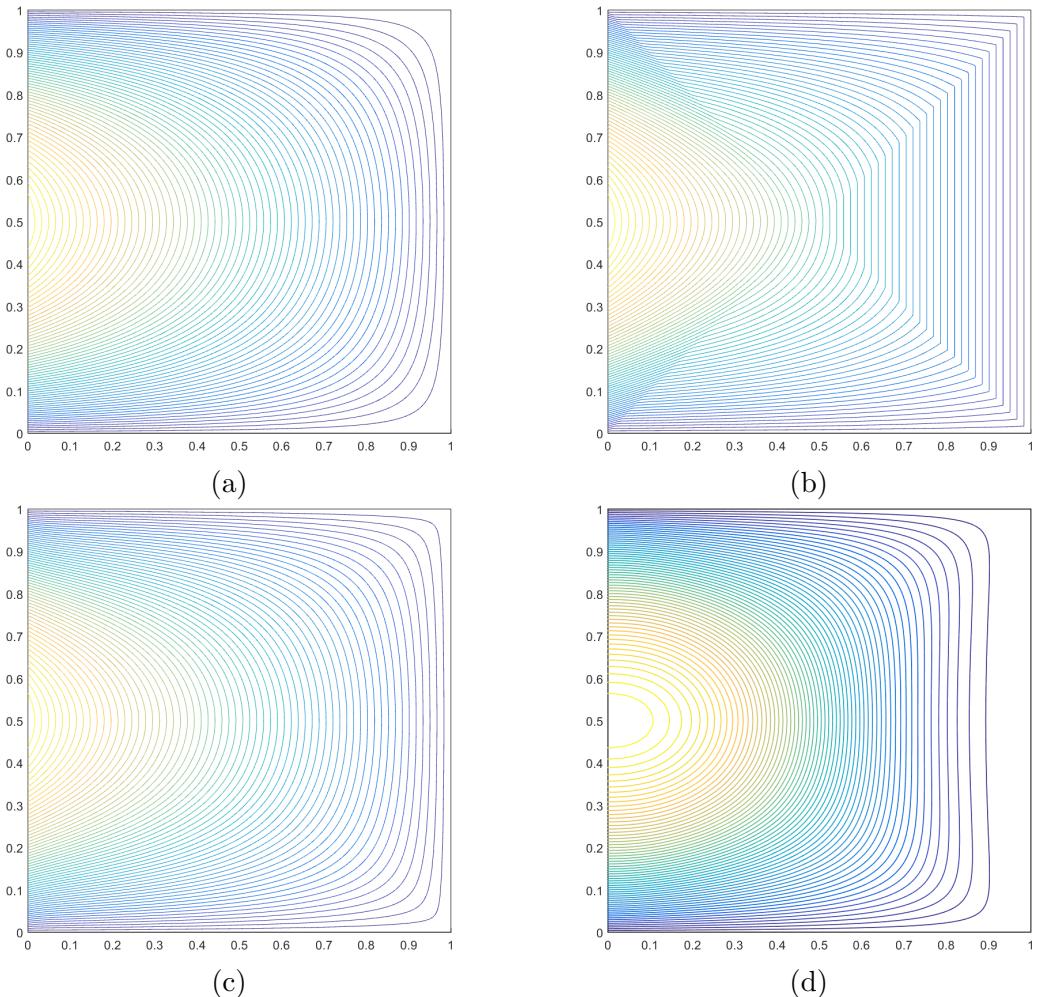


Figure 1.17: Contour plots of the different quadratic serendipity basis function on the unit square at a mid-face node located at $(0,1/2)$: (a) Wachspress, (b) PWL, (c) mean value, and (d) maximum entropy.

1.3 Integrating the Arbitrary 2D Polygonal Elements

Sections 1.1 and 1.2 detail 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 integrated 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 [11, 12, 13, 14, 15]. These include symmetric rules on triangles and cubature and tensor-product rules on triangles and quadrilaterals, respectively. However, polygons have an infinite number 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 [16, 17, 18].

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,

$$\vec{c} = \frac{1}{N_K} \sum_{i=1}^{N_K} \vec{x}_i. \quad (1.31)$$

Then for each sub-triangle, a quadrature rule with N_q nodes is employed (we do not

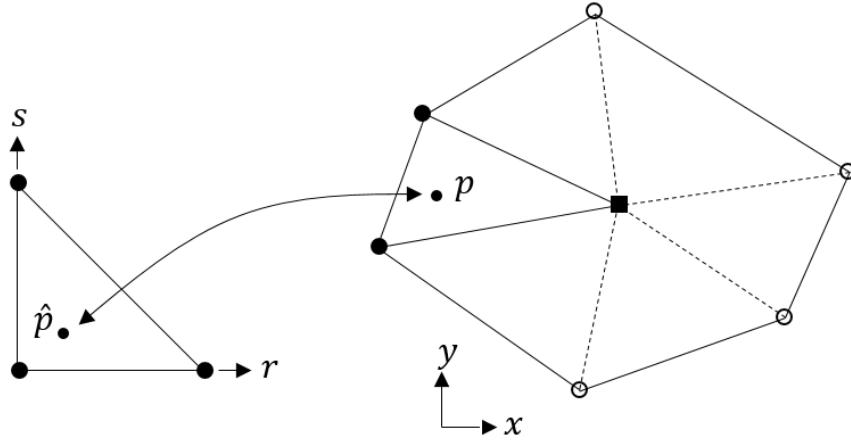


Figure 1.18: blah.

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 [12]. We denote this reference quadrature rule by $\left\{\hat{x}_q, \hat{w}_q\right\}_{q=1}^{N_q}$, 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{p} , to its corresponding point in global space, p , is done with,

$$\mathbf{p} = \mathbf{x}_0 + J\hat{\mathbf{p}}, \quad (1.32)$$

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 1.18. 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,

$$J = \begin{bmatrix} x_1 - x_0 & y_1 - y_0 \\ x_2 - x_0 & y_2 - y_0 \end{bmatrix}. \quad (1.33)$$

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,

$$\nabla_{\hat{x}} = J \nabla_x, \quad (1.34)$$

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

$$\nabla_x = J^{-1} \nabla_{\hat{x}} \quad (1.35)$$

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

$$w_{i,q}^K = \hat{w}_q |J_i|. \quad (1.36)$$

In Eq. (1.36), $|J_i|$ 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,

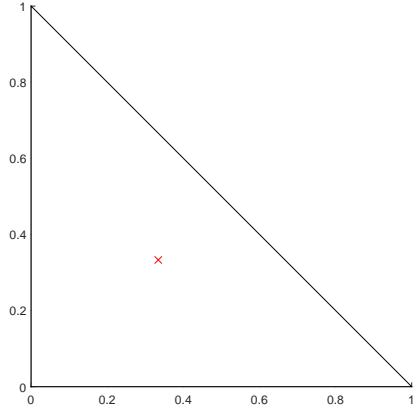
$$\int_K f = \sum_{i=1}^{N_K} \sum_{q=1}^{N_q} w_{i,q}^K f(\vec{x}_{i,q}), \quad (1.37)$$

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

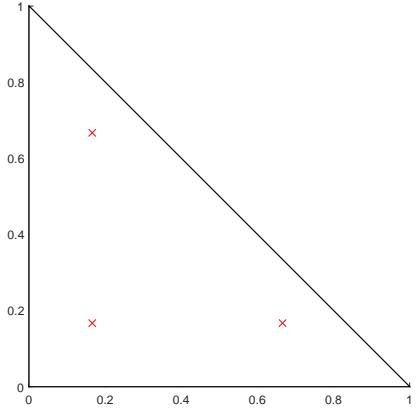
$$\int_{\mathbb{T}_h} f = \sum_{K \in \mathbb{T}_h} \left(\sum_{i=1}^{N_K} \sum_{q=1}^{N_q} w_{i,q}^K f(\vec{x}_{i,q}) \right), \quad (1.38)$$

which is clearly just an element-wise sum of Eq. (1.37).

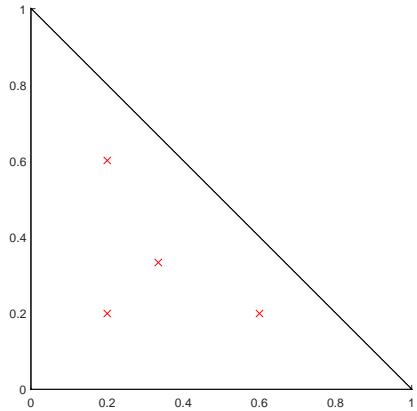
We conclude this section by providing some visual examples of quadrature sets for polygonal elements. Figure 1.19 gives quadrature sets on the reference triangle for orders 1-6. We can see that our reference quadrature is symmetric about any of the three vertices, though we note that true isoparametric symmetry is obtained with equilateral triangles. Then Figures 1.20 and 1.21 provide examples of the mapping of the reference quadrature into the global space for a regular pentagon and hexagon, respectively.



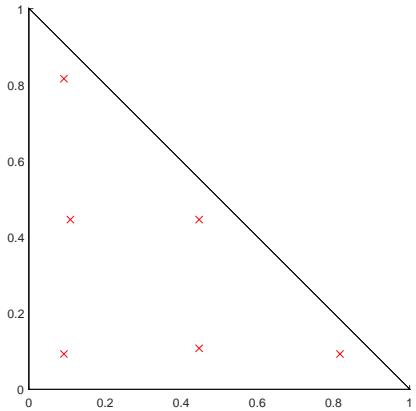
(a) Order 1



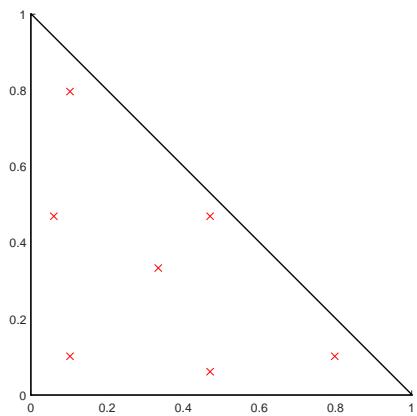
(b) Order 2



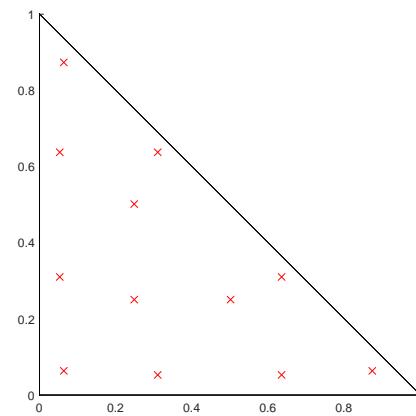
(c) Order 3



(d) Order 4



(e) Order 5



(f) Order 6

Figure 1.19: Quadrature sets on the reference triangle of varying order.

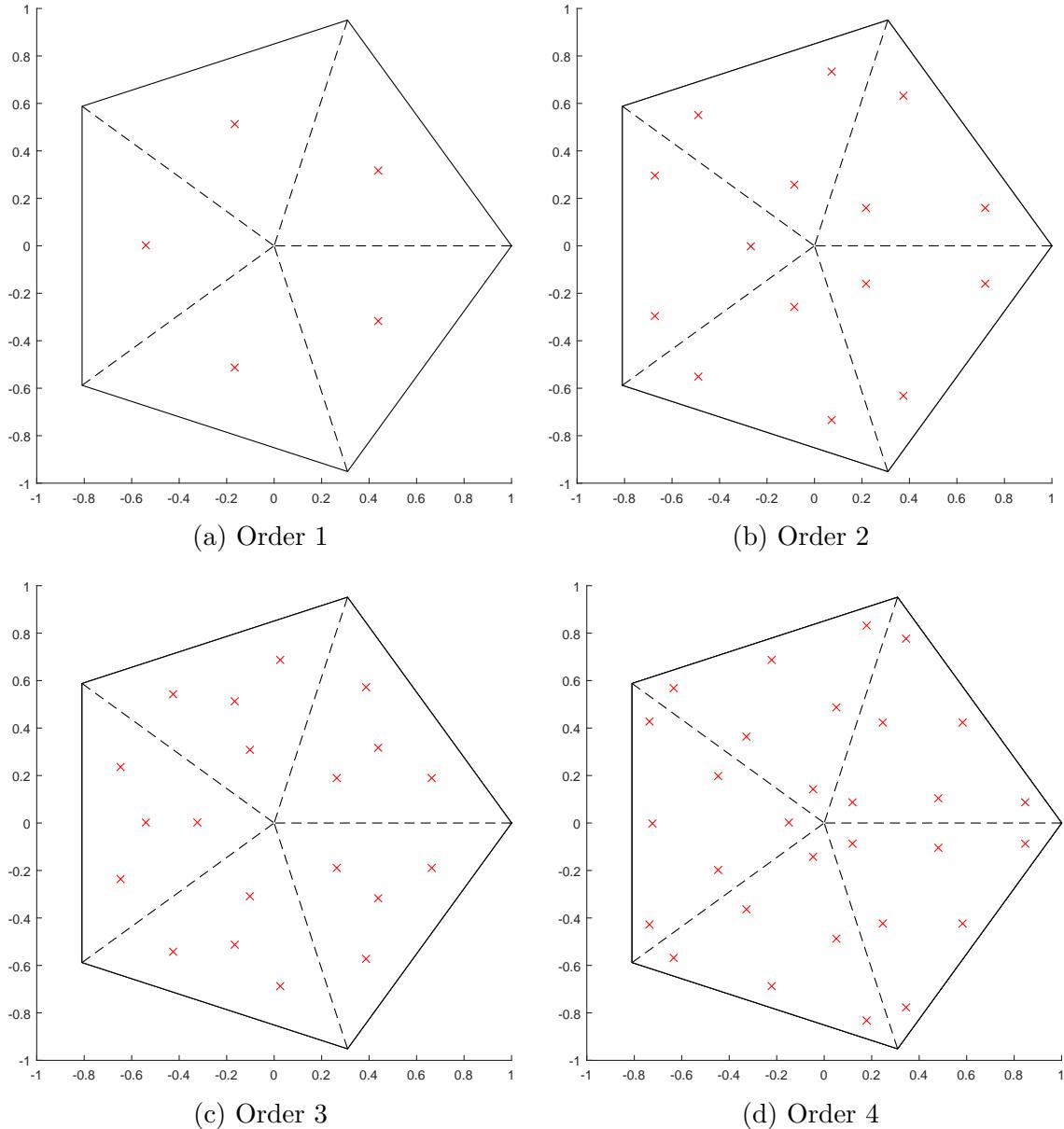


Figure 1.20: blah

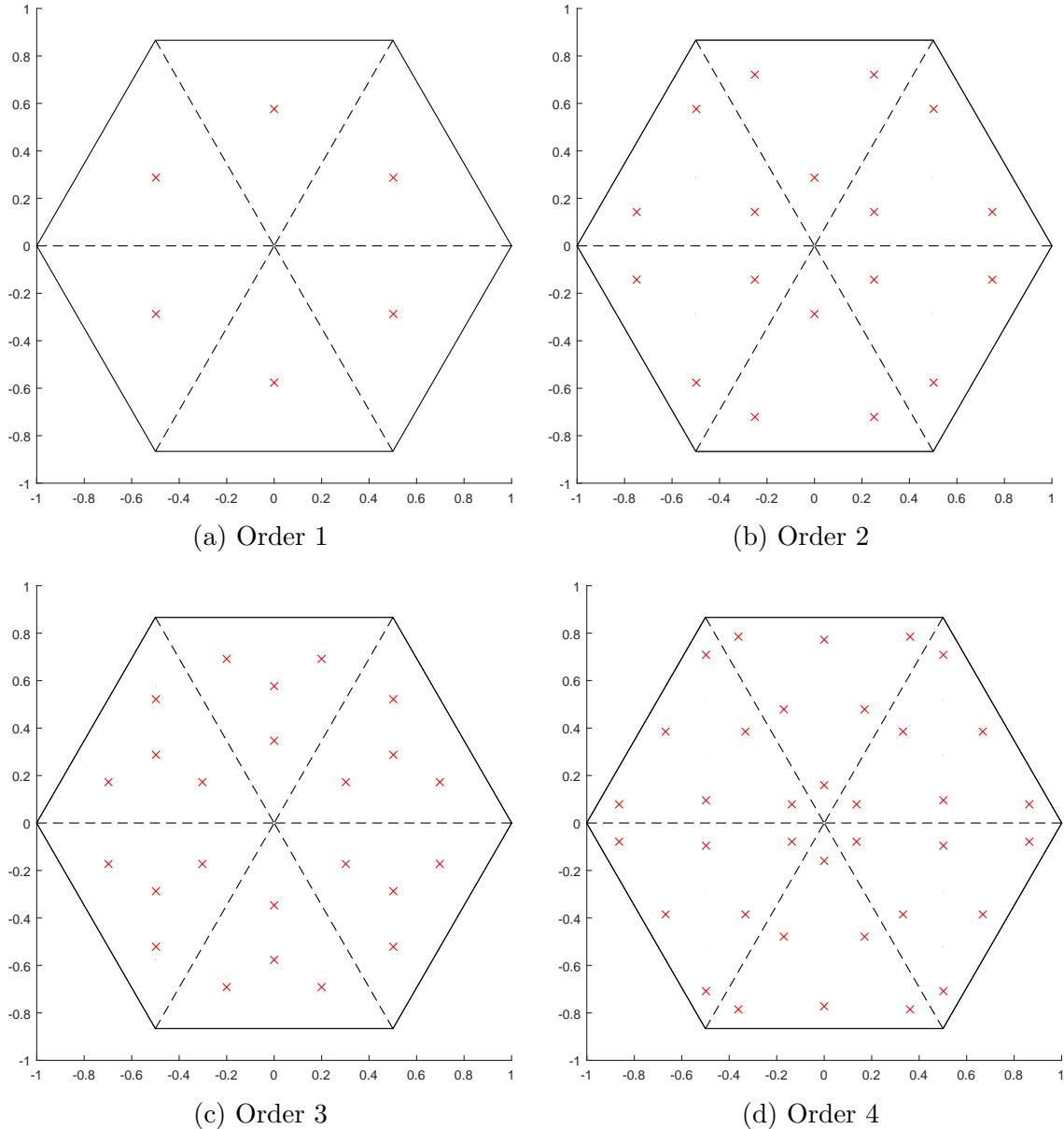


Figure 1.21: blah

1.4 Linear Basis Functions on 3D Polyhedra

We have defined linearly-complete and quadratically-complete 2D polygonal basis functions. Next, we present an efficient coordinate system for arbitrary 3D polyhedra that is linearly-complete. However, at the time of this work, no analogous methodology to convert 3D linear coordinates 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 ??.

For this work, we will utilize the 3D version of the Piecewise Linear (PWL) coordinates. 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. They also allow for extremely-distorted concave polyhedra, though we will not analyze those in this work.

$$b_j(x, y, z) = t_j(x, y, z) + \sum_{f=1}^{F_j} \beta_f^j t_f(x, y, z) + \alpha_j^K t_c(x, y, z) \quad (1.39)$$

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 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 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 [19], we also choose to assume the cell and face weighting parameters are

$$\alpha_{K,j} = \frac{1}{N_K} \quad \text{and} \quad \beta_{f,j} = \frac{1}{N_f}, \quad (1.40)$$

respectively, where N_K is the number of vertices in cell K and N_f is the number of vertices on face f , which leads to constant values of α and β for each cell and face, respectively. This assumption of the cell weight function holds for both 2D and 3D.

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 the presented basis sets can capture an exactly-linear transport solution in Section 1.5.1. Next, we present some convergence properties of the basis sets using the method of manufactured solutions (MMS) in Section 1.5.3. We then present a searchlight problem and observe how the basis sets react with adaptive mesh refinement (AMR) to mitigate numerical dispersion through a vacuum in Section 1.5.5.

1.5.1 Two-Dimensional Exactly-Linear Transport Solutions

We present our first numerical example by demonstrating 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,

$$\mu \frac{\partial \Psi}{\partial x} + \eta \frac{\partial \Psi}{\partial y} + \sigma_t \Psi = Q(x, y, \mu, \eta), \quad (1.41)$$

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:

$$\begin{aligned} \Psi(x, y, \mu, \eta) &= ax + by + c\mu + d\eta + e \\ \Phi(x, y) &= 2\pi(ax + by + e) \end{aligned} . \quad (1.42)$$

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:

$$\sum_q w_q = 2\pi \quad \text{and} \quad \sum_q w_q \begin{bmatrix} \mu_q \\ \eta_q \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} . \quad (1.43)$$

The sum of the quadrature weights is handled by simply renormalizing 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. (1.42). Inserting the angular flux solution of Eq. (1.42) into Eq. (1.41), we obtain the distributed source that will produce our exactly-linear solution space:

$$Q(x, y, \mu, \eta) = a\mu + b\eta + \sigma_t(c\mu + d\eta) + \sigma_t(ax + by + e) . \quad (1.44)$$

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

$$\begin{aligned} a &= -c \sigma_t, \\ b &= -d \sigma_t. \end{aligned} \tag{1.45}$$

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 [20, 21]. 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 [22]. 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. (1.42) to $\sigma_t = a = c = d = e = 1.0$ and $b = 1.5$. We 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, linear maximum entropy, and quadratic serendipity maximum entropy basis functions are presented in Figures 1.22, ??, ??, ??, and ??, respectively. We can see that for all the polygonal basis functions, an exact linear solution is captured as shown by the unbroken nature of the contour lines. This even holds on the highly distorted quadrilateral shestakov mesh.

1.5.2 Two-Dimensional Exactly-Quadratic Transport Solutions

$$\begin{aligned}\Psi(x, y, \mu, \eta) &= a + bx + cy + dxy + ex^2 + fy^2 \\ \Phi(x, y) &= 2\pi(a + bx + cy + dxy + ex^2 + fy^2)\end{aligned}. \quad (1.46)$$

1.5.3 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 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

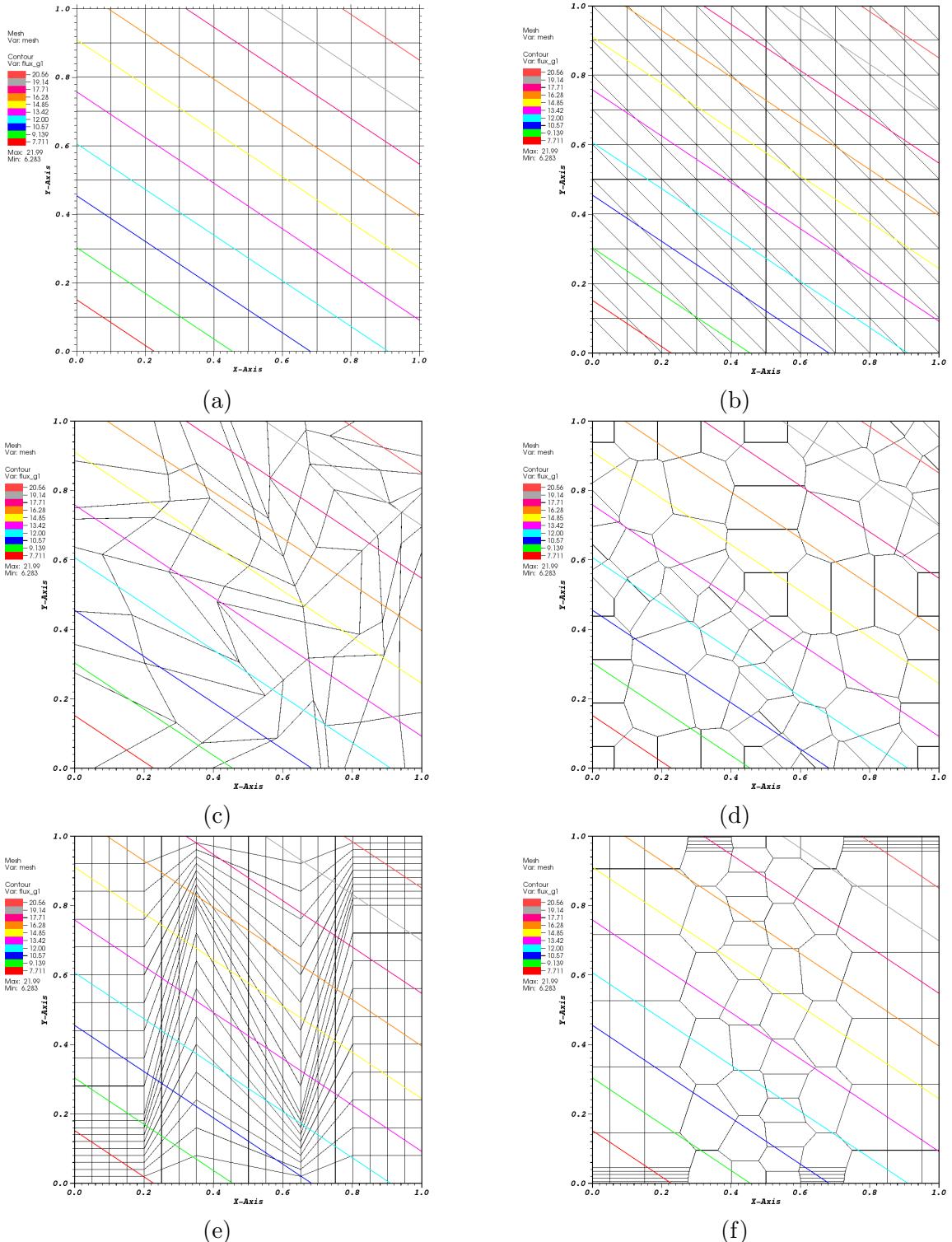


Figure 1.22: Contour plots of the exactly-linear solution with the Wachspress basis functions on (a) cartesian mesh, (b) ordered-triangular mesh, (c) quadrilateral shestakov mesh, (d) sinusoidal polygonal mesh, (e) quadrilateral z-mesh, and (f) polygonal z-mesh.

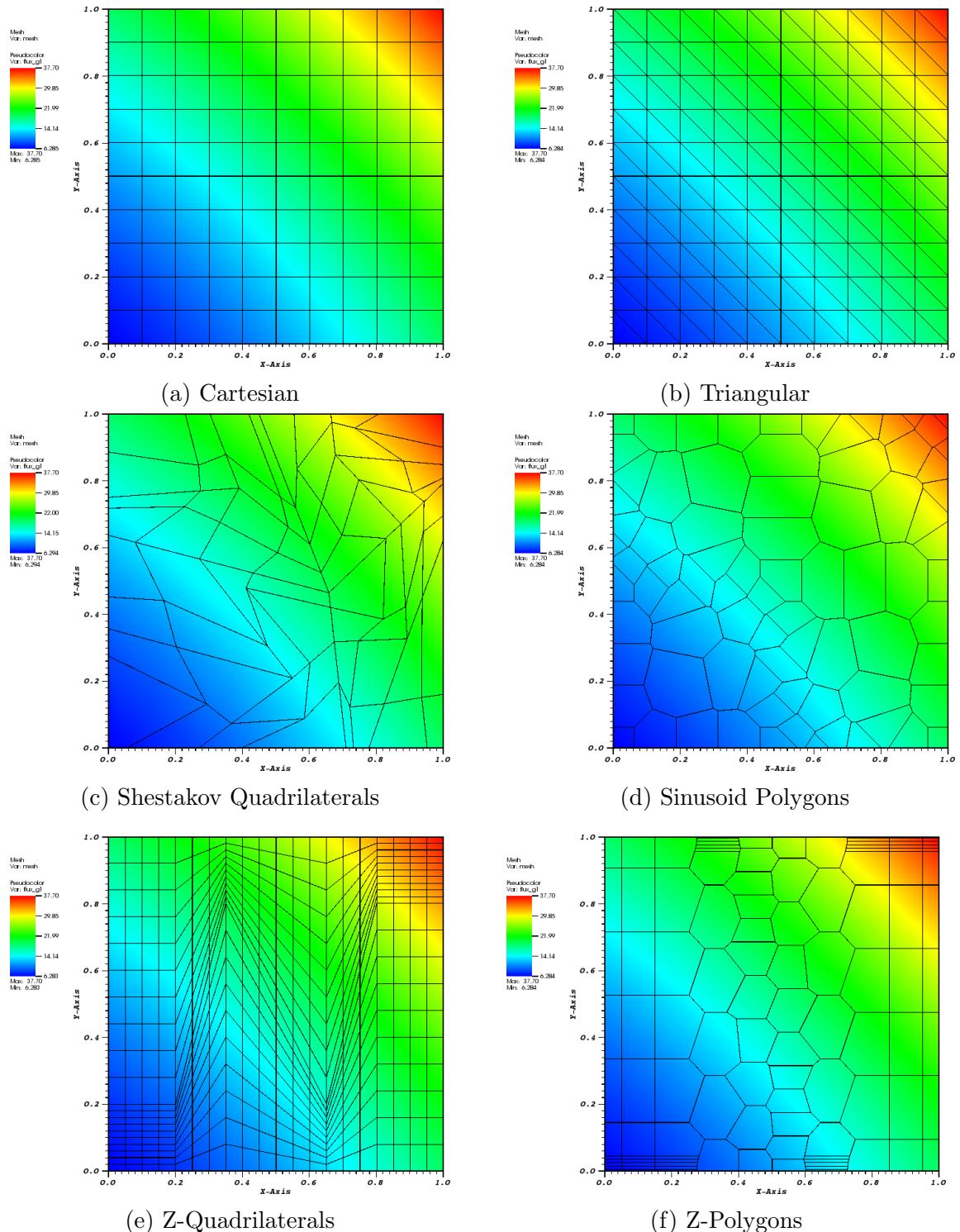


Figure 1.23: Plots of the exactly-quadratic solution with the quadratic serendipity maximum entropy basis functions.

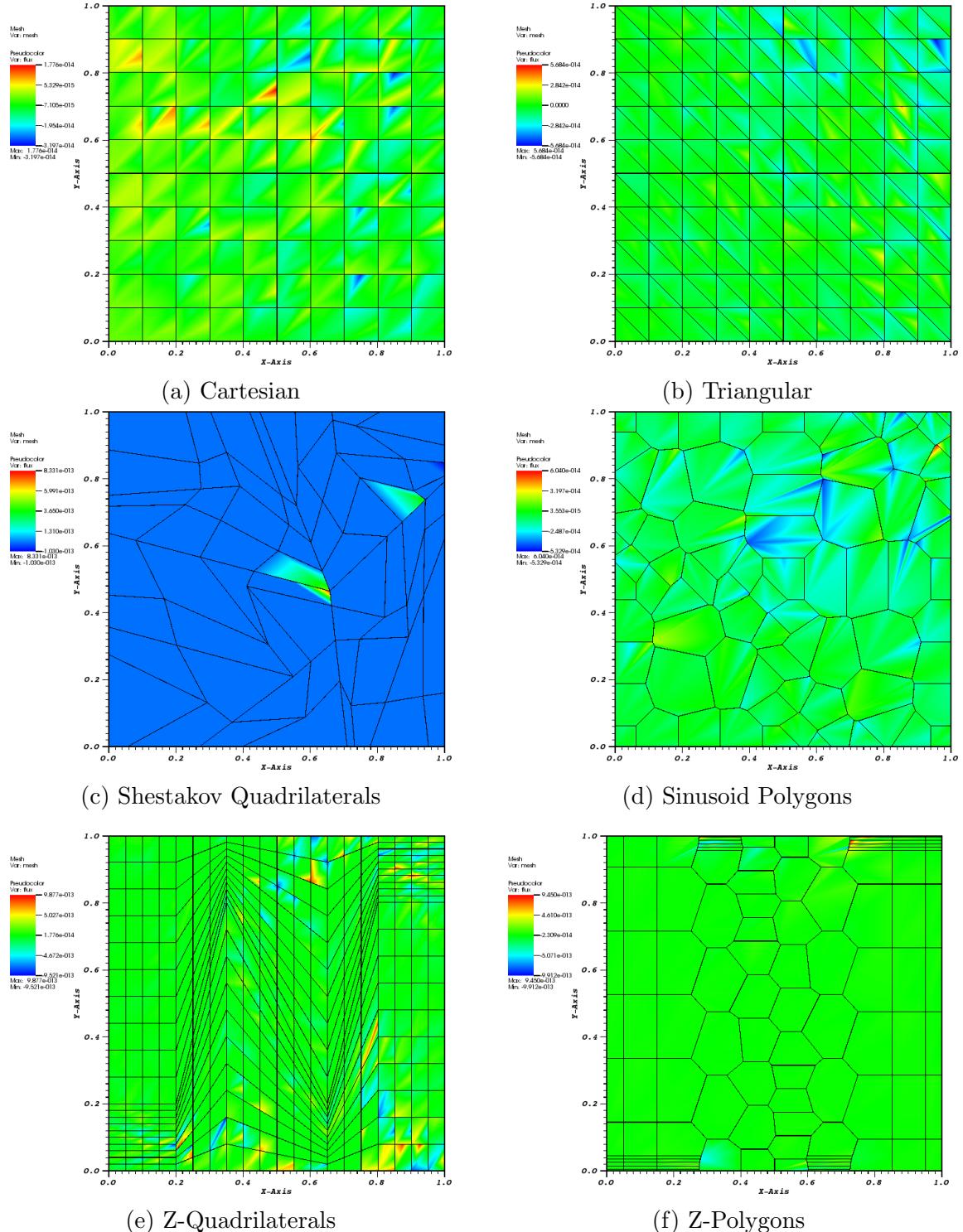
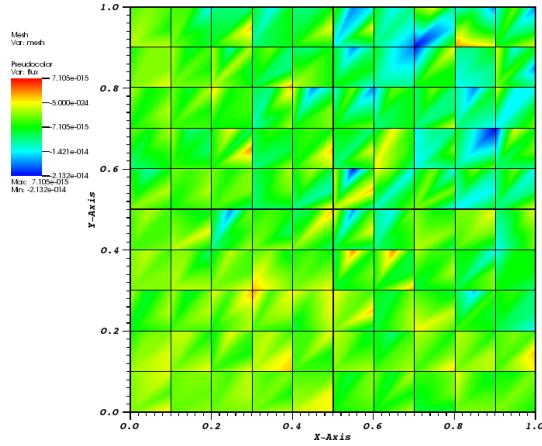
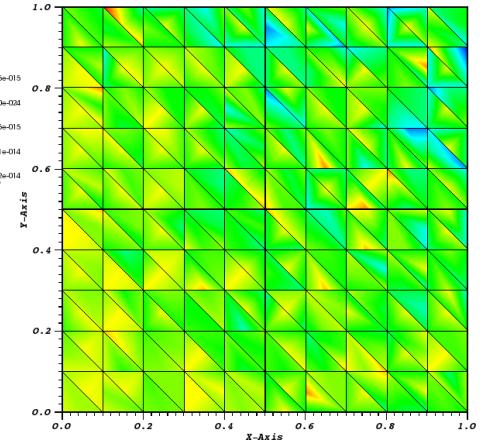


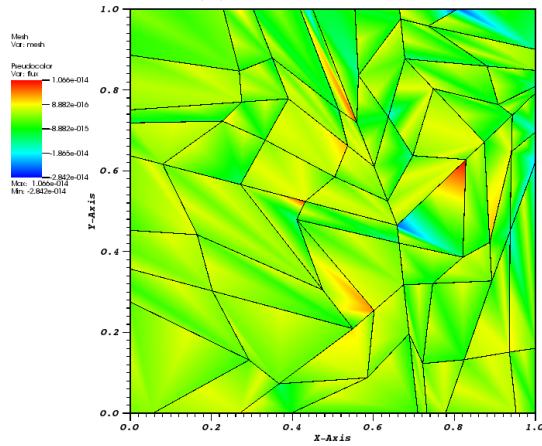
Figure 1.24: Plots of the error of the exactly-quadratic solution with the quadratic serendipity Wachspress basis functions.



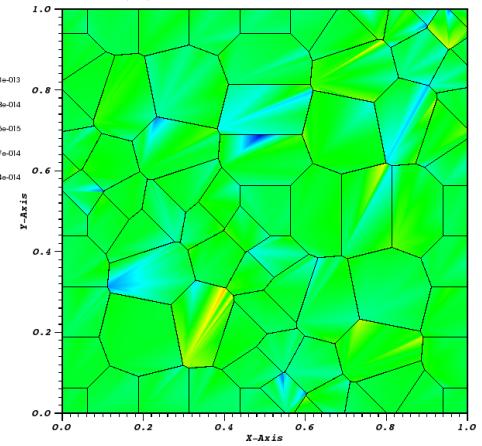
(a) Cartesian



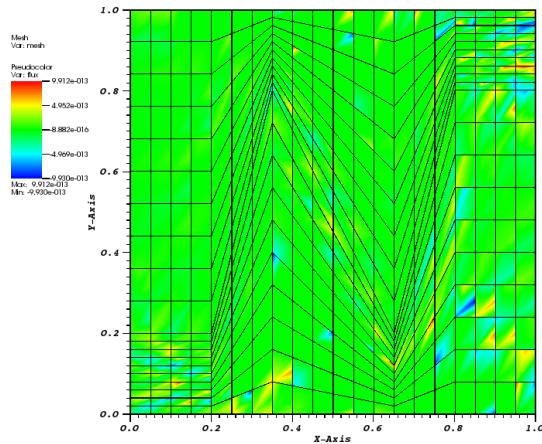
(b) Triangular



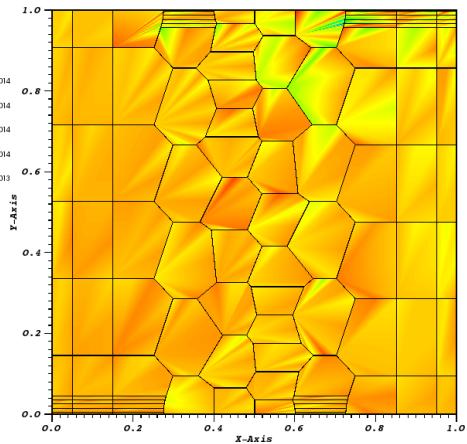
(c) Shestakov Quadrilaterals



(d) Sinusoid Polygons



(e) Z-Quadrilaterals



(f) Z-Polygons

Figure 1.25: Plots of the error of the exactly-quadratic solution with the quadratic serendipity PWL basis functions.

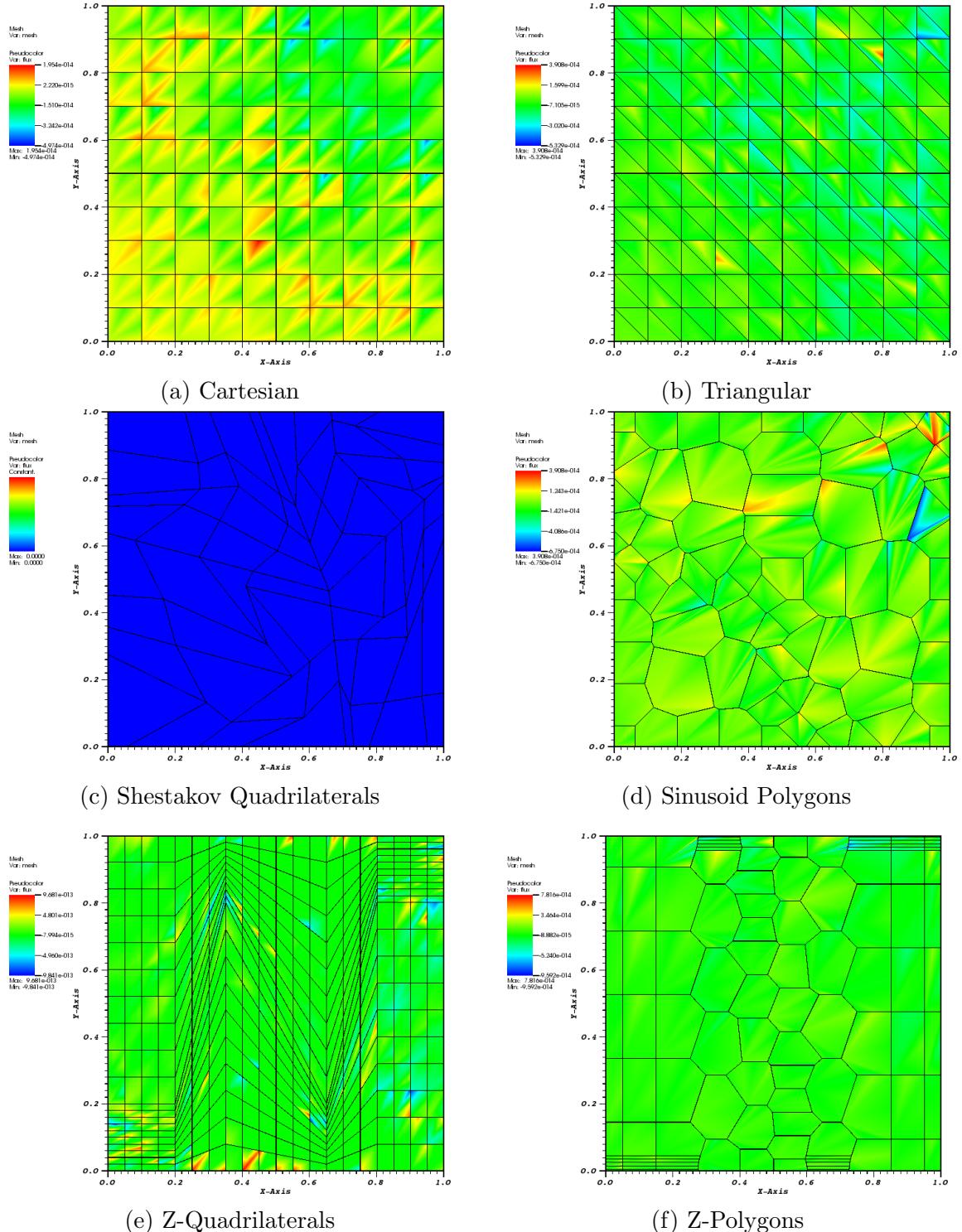


Figure 1.26: Plots of the error of the exactly-quadratic solution with the quadratic serendipity mean value basis functions.

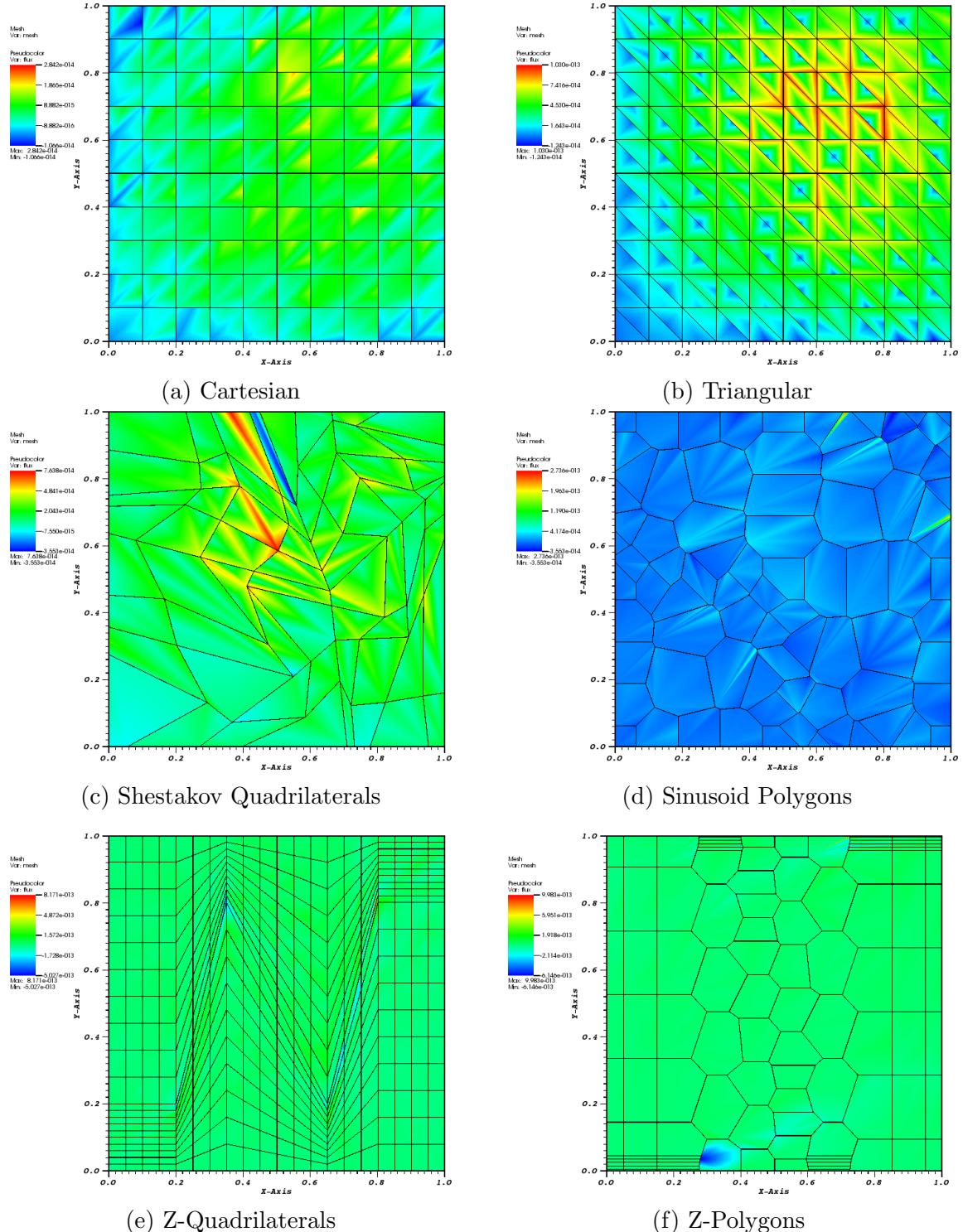


Figure 1.27: Plots of the error of the exactly-quadratic solution with the quadratic serendipity maximum entropy basis functions.

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 a quadratic function and a gaussian which yields a significant local maximum.

The sinusoid flux solutions, $\{\Psi^s, \Phi^s\}$, have the following parameterized form,

$$\begin{aligned}\Psi^s(x, y) &= \sin(\nu \frac{\pi x}{L_x}) \sin(\nu \frac{\pi y}{L_y}), \\ \Phi^s(x, y) &= 2\pi \sin(\nu \frac{\pi x}{L_x}) \sin(\nu \frac{\pi y}{L_y}),\end{aligned}\tag{1.47}$$

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,

$$\begin{aligned}\Psi^g(x, y) &= C_M x(L_x - x)y(L_y - y) \exp\left(-\frac{(x - x_0)^2 + (y - y_0)^2}{\gamma}\right), \\ \Phi^g(x, y) &= 2\pi C_M x(L_x - x)y(L_y - y) \exp\left(-\frac{(x - x_0)^2 + (y - y_0)^2}{\gamma}\right),\end{aligned}\tag{1.48}$$

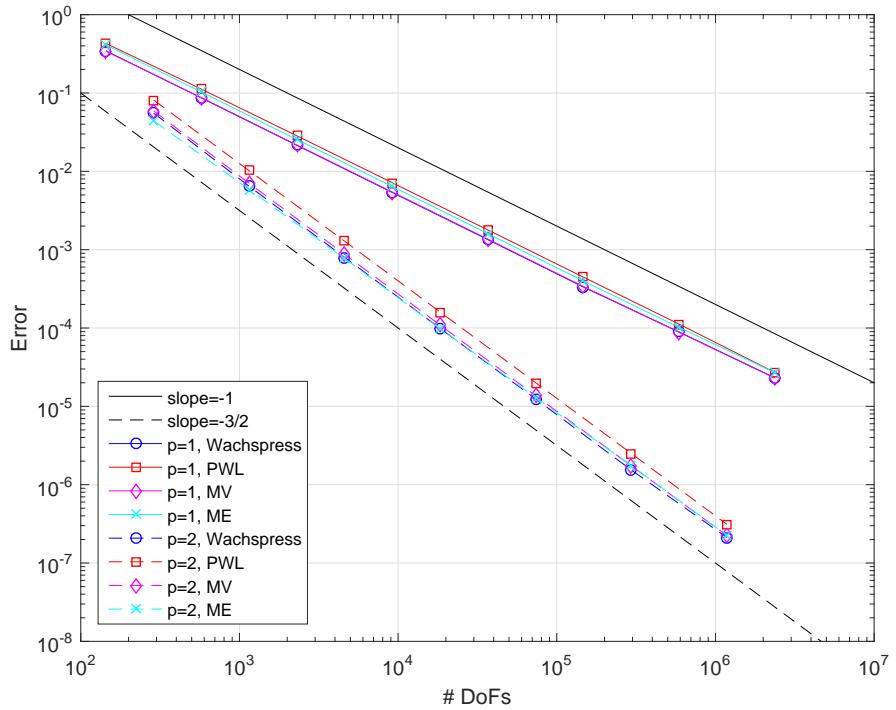


Figure 1.28: Convergence rates for the sinusoid MMS problem on a Cartesian mesh.

where the constants in the equations are:

$$C_M = \frac{100}{L_x^2 L_y^2} \quad \gamma = \frac{L_x L_y}{100}. \quad (1.49)$$

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 $\frac{225}{32}\pi \approx 22.1$.

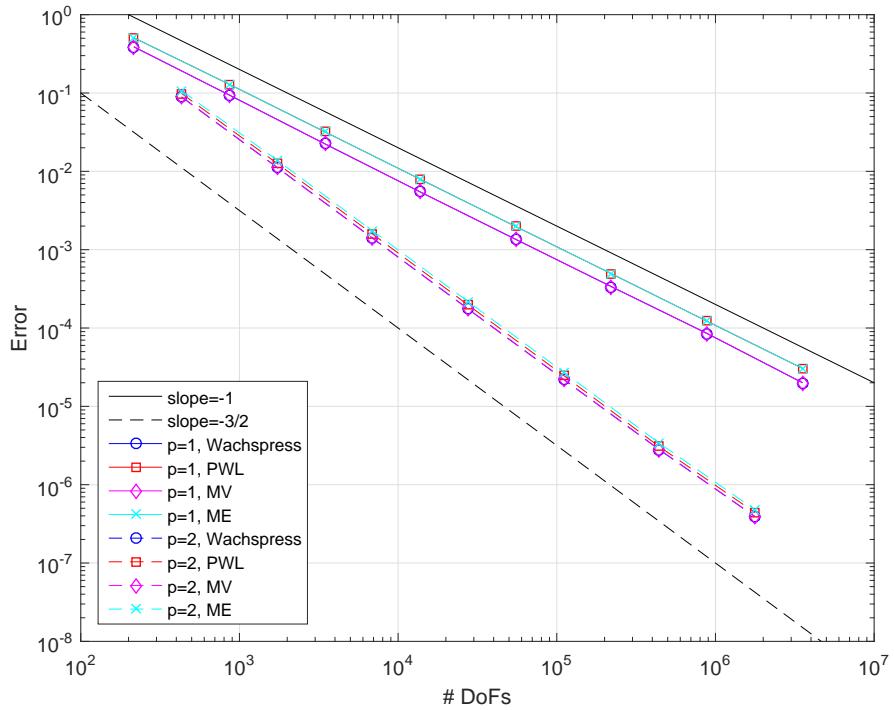


Figure 1.29: Convergence rates for the sinusoid MMS problem on an ordered triangular mesh.

1.5.4 Convergence Rate Analysis in a Purely-Absorbing Medium

1.5.5 Searchlight Problem

The next example models a beam or searchlight. Similar problems were investigated in Dedner and Vollmöller [23] and Wang and Ragusa [24]. 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 the 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

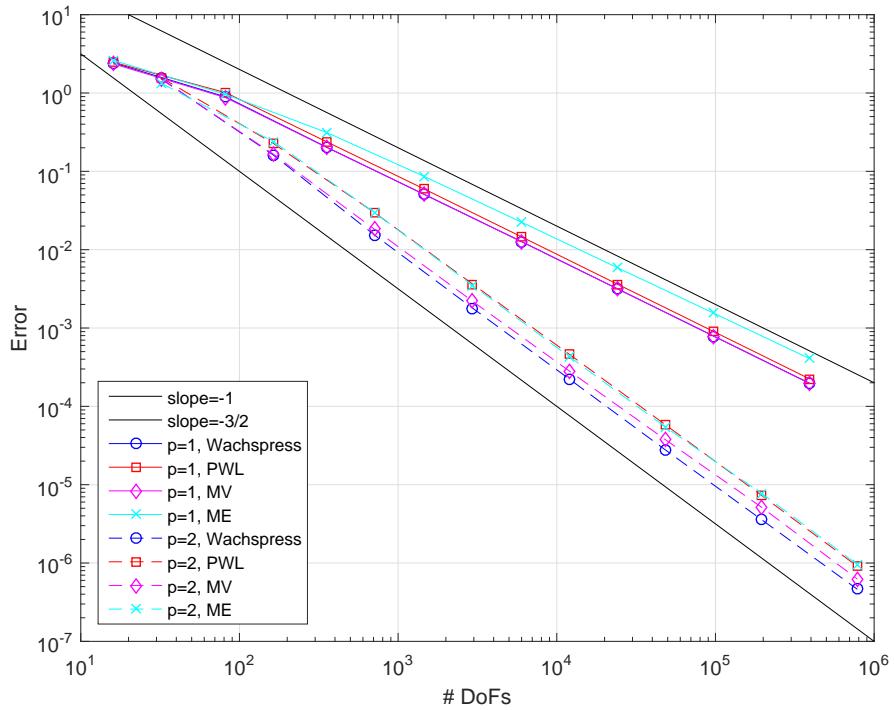


Figure 1.30: Convergence rates for the sinusoid MMS problem on a regular polygonal mesh.

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.

We investigated this problem using several of the 2D polygonal basis functions as outlined in Sections 1.1 and 1.2 as well as

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, and maximum entropy coordinates. The Wachspress and the PWL coordinates had

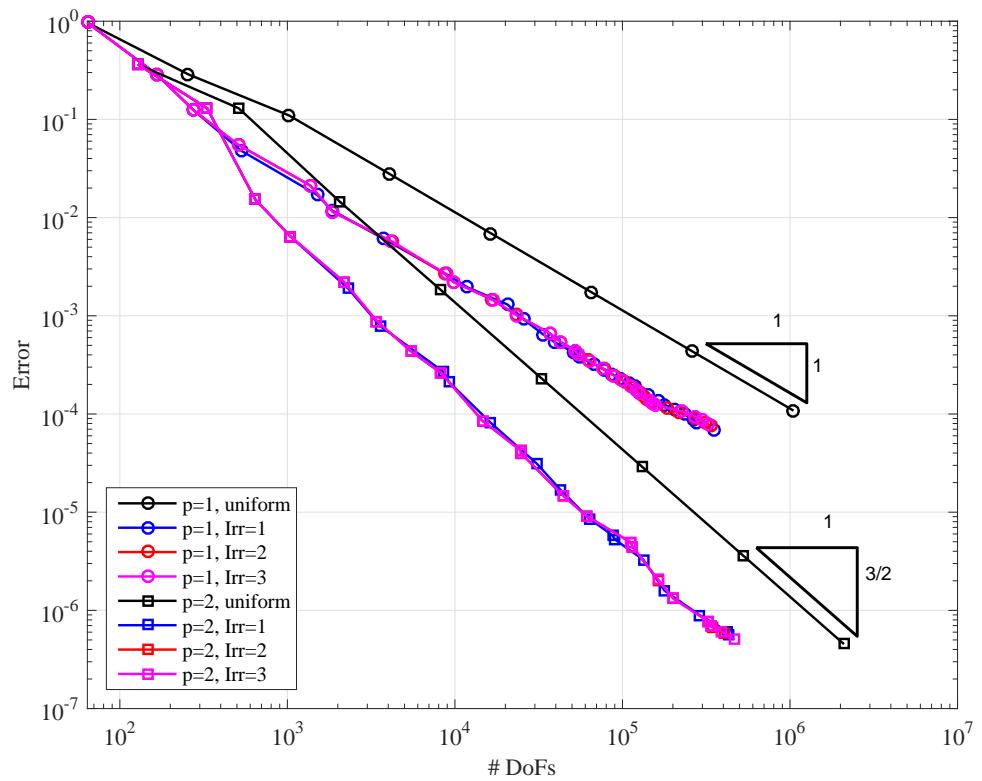


Figure 1.31: Convergence rates for the 2D Gaussian MMS problem using the PWL basis functions.

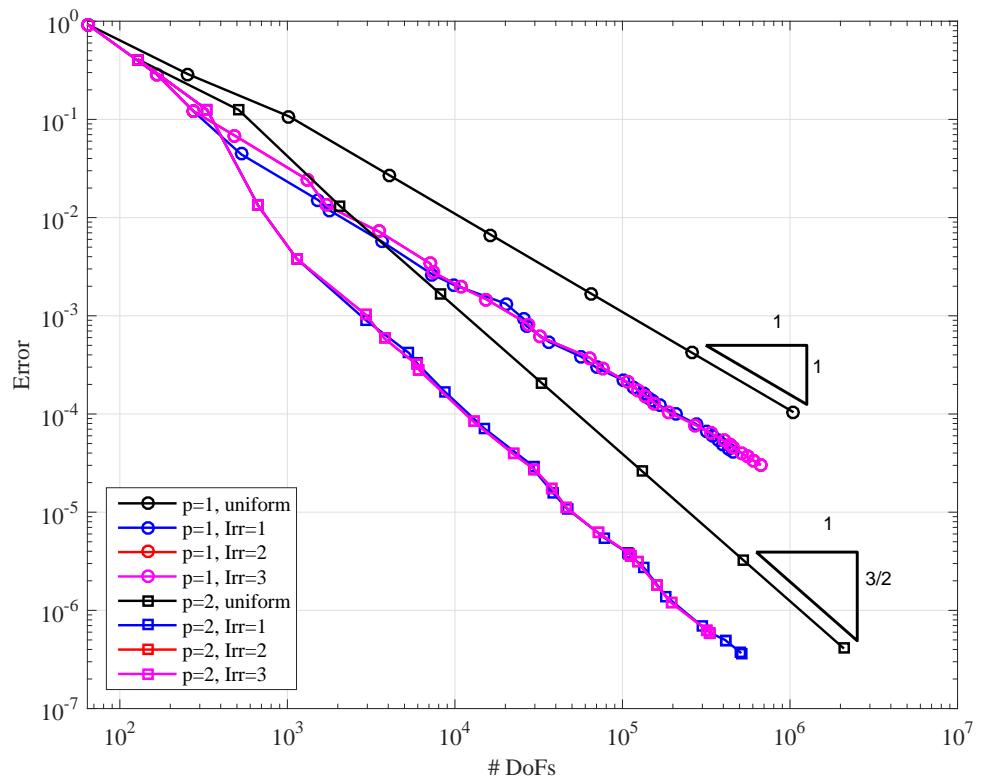


Figure 1.32: Convergence rates for the 2D Gaussian MMS problem using the mean value basis functions.

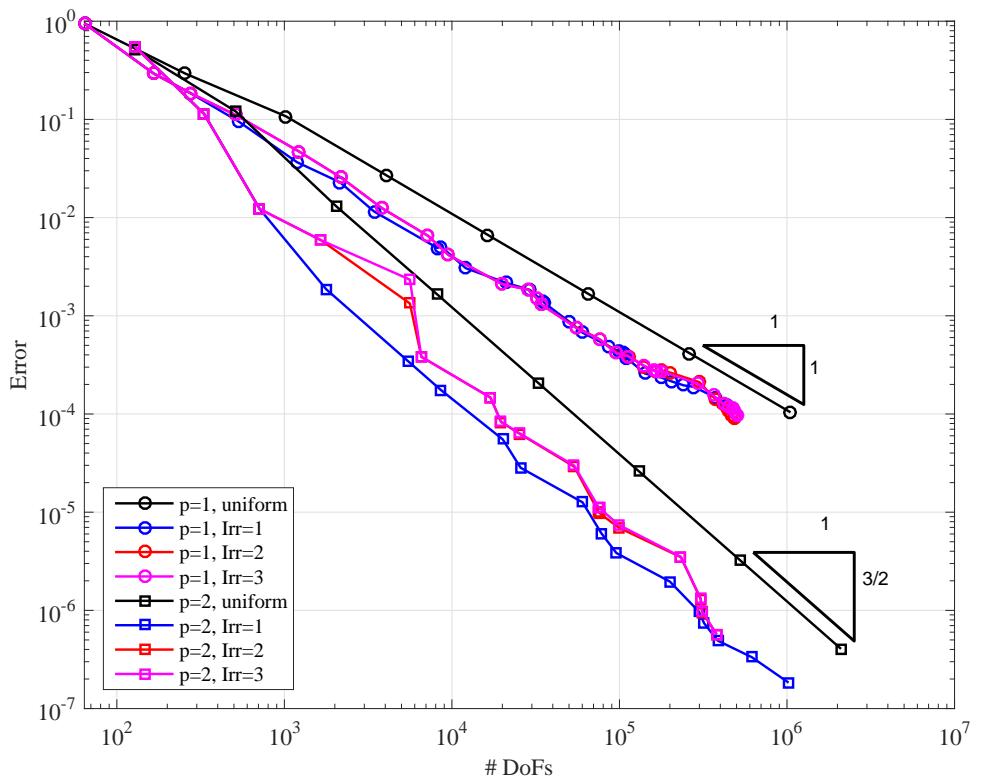


Figure 1.33: Convergence rates for the 2D Gaussian MMS problem using the maximum entropy basis functions.

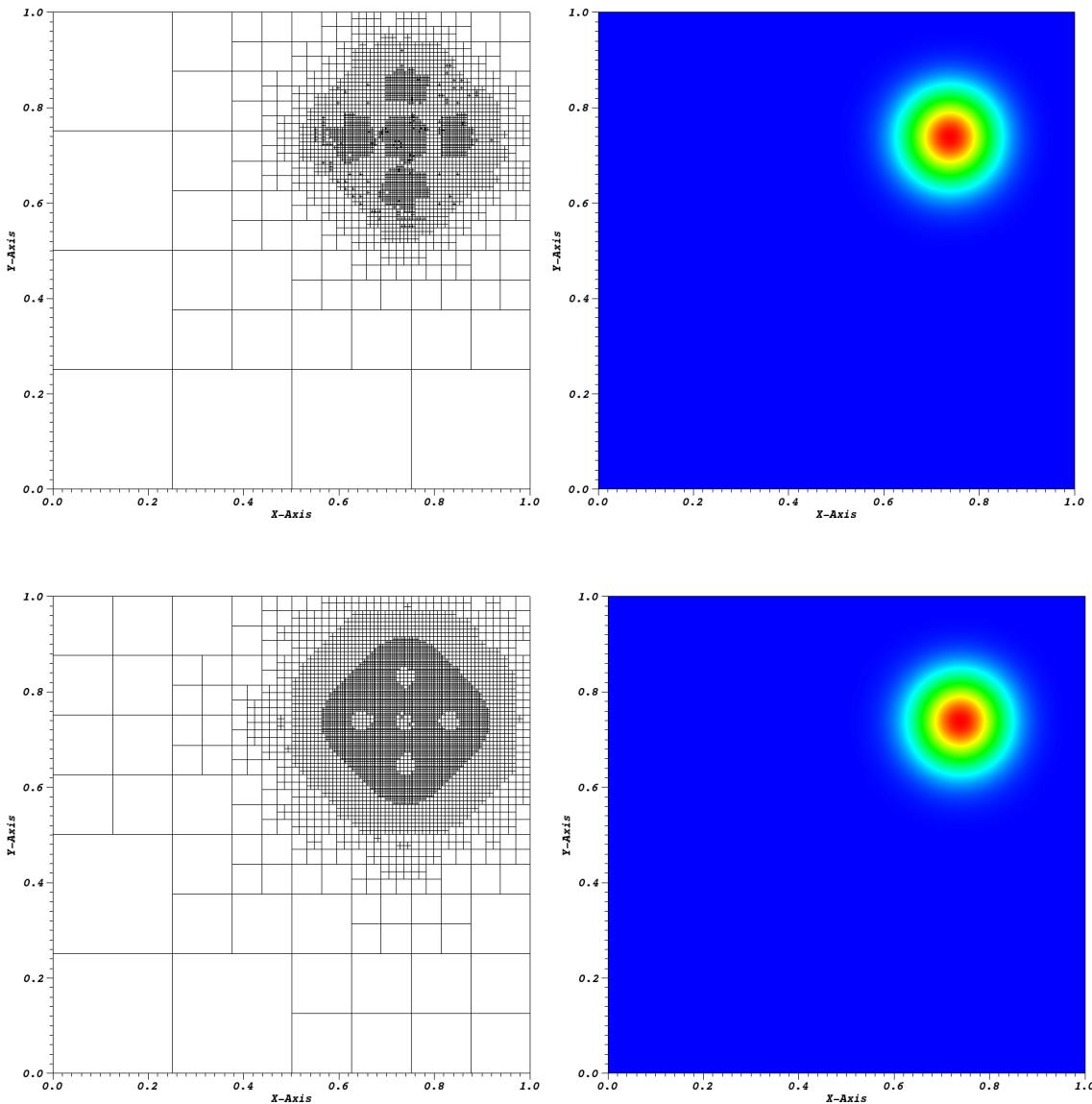


Figure 1.34: AMR meshes and solutions for the gaussian MMS problem using the maximum entropy coordinates: (top) linear basis functions at cycle 15 and (bottom) quadratic serendipity basis functions at cycle 08.

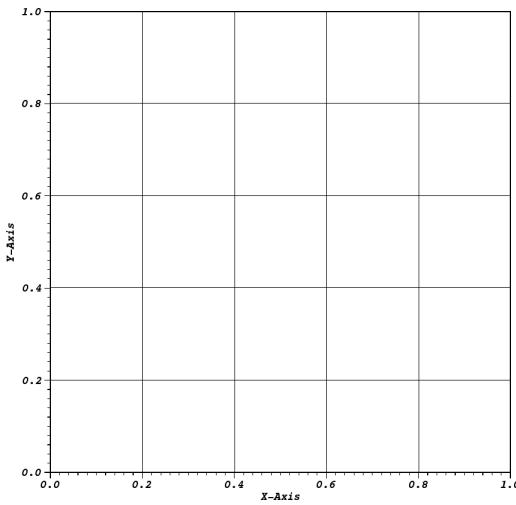


Figure 1.35: Initial mesh configuration for the searchlight problem before any refinement cycles.

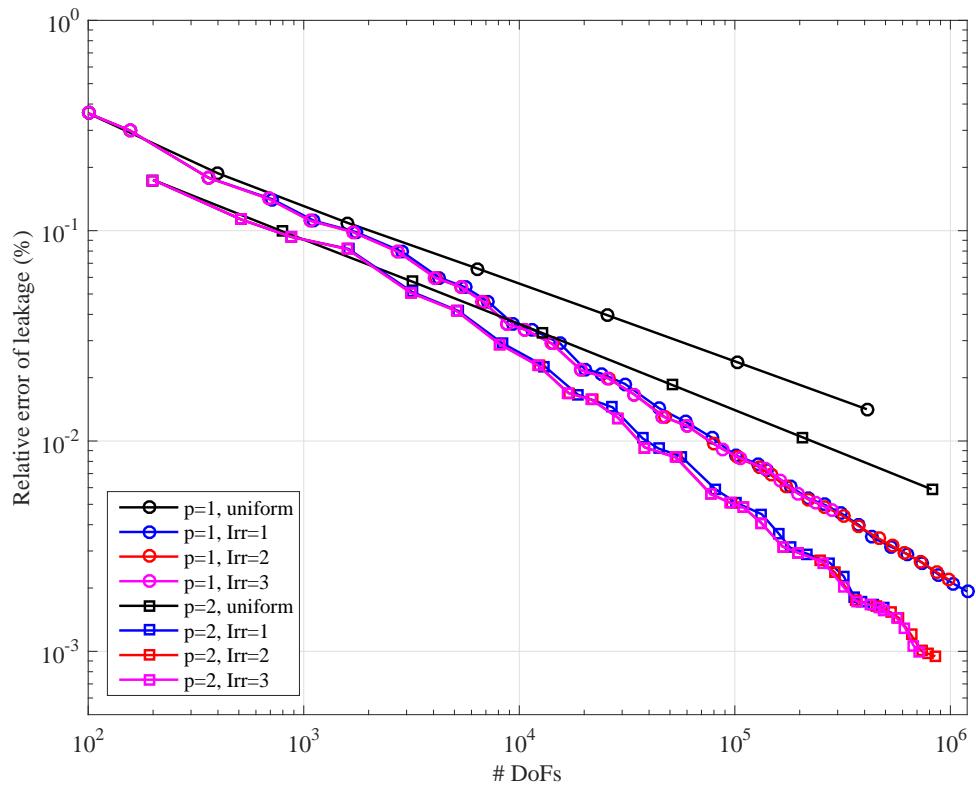


Figure 1.36: blah.

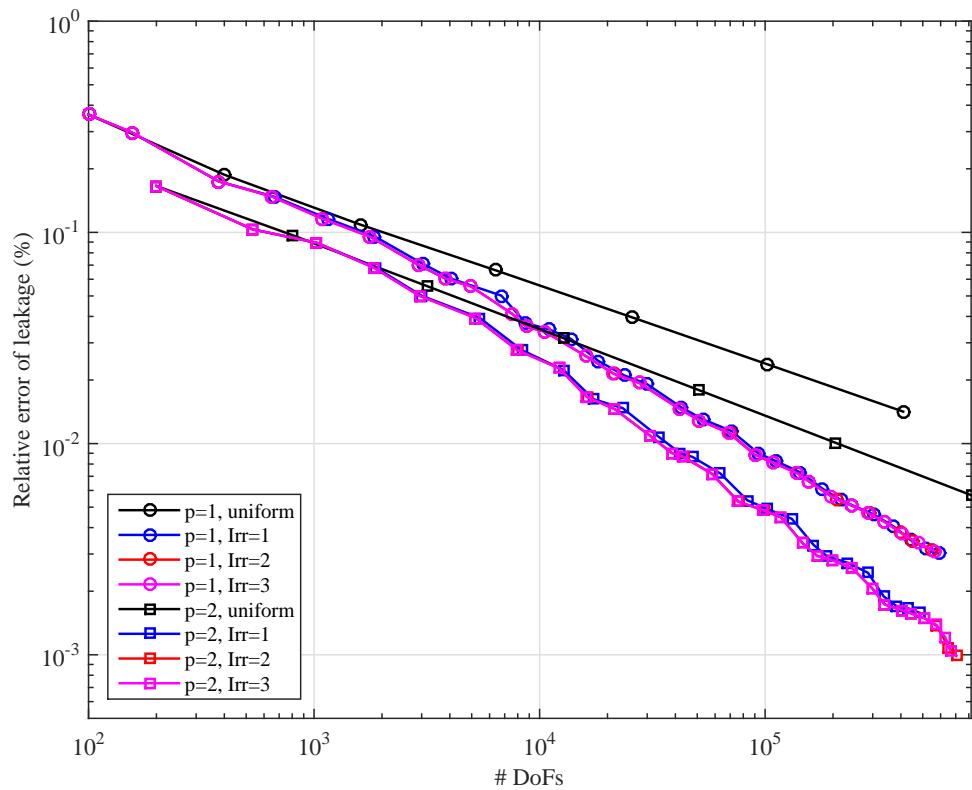


Figure 1.37: blah.

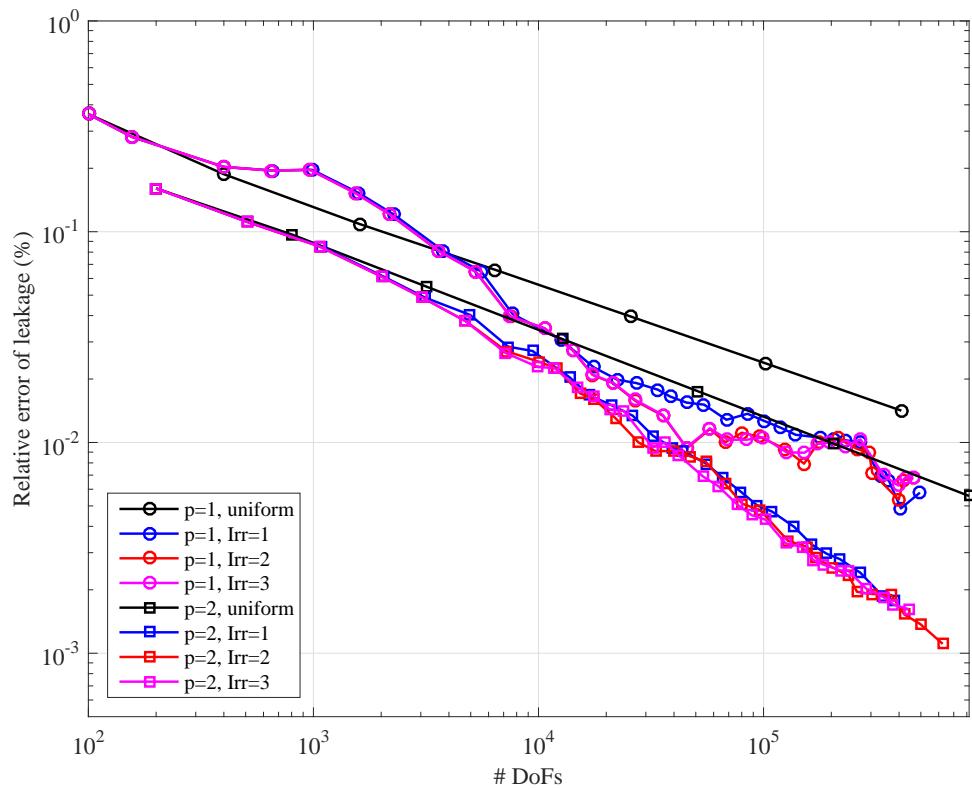


Figure 1.38: blah.

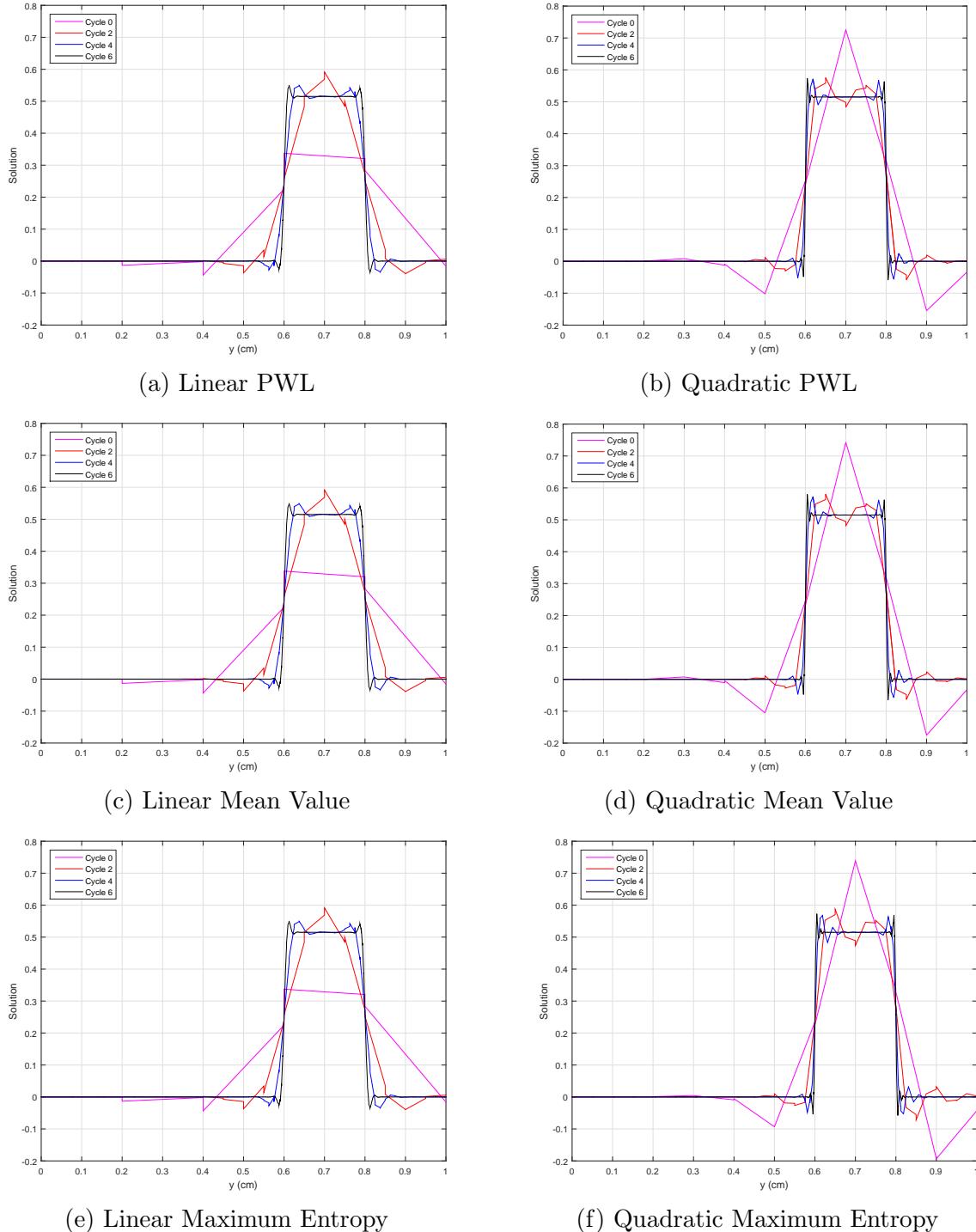


Figure 1.39: Exiting angular flux on the right boundary with uniform refinement.

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 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 preasymptotic region before being restricted by r as the mesh gets optically thin.

REFERENCES

- [1] E. L. WACHSPRESS, “A Rational Finite Element Basis,” *Mathematics in science and engineering* (1975).
- [2] 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).
- [3] 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).
- [4] M. S. FLOATER, “Mean value coordinates,” *Computer aided geometric design*, **20**, 1, 19–27 (2003).
- [5] K. HORMANN and M. S. FLOATER, “Mean value coordinates for arbitrary planar polygons,” *ACM Transactions on Graphics (TOG)*, **25**, 4, 1424–1441 (2006).
- [6] N. SUKUMAR, “Construction of polygonal interpolants: a maximum entropy approach,” *International Journal for Numerical Methods in Engineering*, **61**, 12, 2159–2181 (2004).
- [7] 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).

- [8] K. HORMANN and N. SUKUMAR, “Maximum entropy coordinates for arbitrary polytopes,” in “Computer Graphics Forum,” Wiley Online Library (2008), vol. 27, pp. 1513–1520.
- [9] S. KULLBACK and R. A. LEIBLER, “On information and sufficiency,” *The annals of mathematical statistics*, pp. 79–86 (1951).
- [10] E. T. JAYNES, “Information theory and statistical mechanics,” in “Statistical Physics,” (1963), vol. 3, pp. 181–218.
- [11] P. SILVESTER, “Symmetric quadrature formulae for simplexes,” *Mathematics of Computation*, **24**, 109, 95–100 (1970).
- [12] D. DUNAVANT, “High degree efficient symmetrical Gaussian quadrature rules for the triangle,” *International journal for numerical methods in engineering*, **21**, 6, 1129–1148 (1985).
- [13] S. WANDZURAT and H. XIAO, “Symmetric quadrature rules on a triangle,” *Computers & Mathematics with Applications*, **45**, 12, 1829–1840 (2003).
- [14] 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.
- [15] R. COOLS and A. HAEGEMANS, “Construction of minimal cubature formulae for the square and the triangle, using invariant theory,” (1987).
- [16] M. NOOLJEN, G. TE VELDE, and E. BAERENDS, “Symmetric numerical integration formulas for regular polygons,” *SIAM journal on numerical analysis*, **27**, 1, 198–218 (1990).
- [17] G. DASGUPTA, “Integration within polygonal finite elements,” *Journal of Aerospace Engineering*, **16**, 1, 9–18 (2003).

- [18] 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).
- [19] 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).
- [20] 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).
- [21] A. SHESTAKOV, D. KERSHAW, and G. ZIMMERMAN, “Test problems in radiative transfer calculations,” *Nuclear science and engineering*, **105**, 1, 88–104 (1990).
- [22] D. S. KERSHAW, “Differencing of the diffusion equation in Lagrangian hydrodynamic codes,” *Journal of Computational Physics*, **39**, 2, 375–395 (1981).
- [23] 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).
- [24] 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).