

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. INTRODUCTION

### 1.1 Motivation and Purpose of the Dissertation

1. Polytope mesh cells are now being employed in other physics communities - most notably computational fluid dynamics (CFD) [1];
2. They are believed to reduce the number of unknowns to solve with equivalent accuracy;
3. They can reduce cell/face counts which can reduce algorithm wallclock times depending on the solution method;
4. They can allow for transition elements between different portions of the domain (e.g., tetrahedral elements bordering hexahedral elements at the border of the boundary layer);
5. They can easily be split along cut planes - allowing the mesh to be partitioned into regular or irregular divisions as well as be generated by simplicial meshing techniques across processor sets in parallel;
6. Hanging nodes from non-conforming meshes, like those that naturally arise from locally refined/adapted meshes, are no longer necessary.

## 1.2 Current State of the Problem

### 1.2.1 Background on the Multigroup DGFEM $S_N$ Transport Equation

### 1.2.2 Diffusion Synthetic Acceleration

### 1.2.3 Polytope Grids Formed from Voronoi Mesh Generation

Since this dissertation work is on the solution of the transport equation on polytope meshes, we next describe how these grids can be generated. Traditionally, FEM calculations have been performed on simplicial (triangles and tetrahedra) and tensor based meshes (quadrilaterals and hexahedra). In fact, it is still a standard practice to refer to any type of mesh as a *triangulation* in some communities [2]. Many different mesh generation software has been developed to build these simple grids [3, 4, 5, 6]. However, multiple fields including *computational fluid dynamics* (CFD) and *solid mechanics* are now finding benefits to utilizing polygonal and polyhedral meshes for their calculations.

However, polytope mesh generation is still in its infancy.

### 1.2.4 Adaptive Mesh Refinement for the DGFEM $S_N$ Transport Equation

## 1.3 Organization of the Dissertation

In this introductory chapter, we have presented a summary of work performed. We also gave our motivation for choosing this work as well as a brief discussion of previous work that has directly influenced this dissertation. We conclude this introduction by briefly describing the remaining chapters of this dissertation.

In Chapter ??, we present the DGFEM formulation for the multigroup,  $S_N$  transport equation. We then describe the transport equation's discretization in energy, angle, and space. We have left the FEM spatial interpolation function as arbitrary at this point to be defined in detail in Chapter 2. For the spatial variable, we pro-

vide the theoretical convergence properties of the DGFEM form. We also detail the elementary assembly procedures to form the full set of spatial equations. We conclude by providing the methodology to be used to solve the full phase-space of the transport problem.

In Chapter 2, we present all the finite element basis functions that we will use in this work. In two dimensions, we present four different linearly-complete polygonal coordinate systems that we will use to generate our finite element basis functions. We then present the methodology that converts each of these linear coordinate systems into quadratically-complete coordinates for use as higher-order basis functions. We also present the single linearly-complete polyhedral coordinate system that we will use for the 3D transport problems.

In Chapter ??, we present the methodologies to be used for DSA preconditioning of the DGFEM transport equation for optically thick problems. We give a discontinuous form of the diffusion equation which can be used on 2D and 3D polytope grids. The theoretical limits of the DSA scheme are analyzed and we conclude with a real-world problem of accelerating the thermal neutron upscattering of a large multigroup, heterogeneous transport problem. In doing so, we demonstrate that our methodology will work on massively-parallel computer architectures.

We then finalize this dissertation work by drawing conclusions and discussing open topics of research stemming from this dissertation in Chapter ???. We note that our detailed literature reviews, numerical results, and conclusions pertaining to each topic are presented in their corresponding chapter.

Additional material that is not included in the main body of the dissertation for the sake of brevity is appended for completeness. The appendices are organized in a simple manner:

- Appendix ???: addendum to Section ??, corresponding to additional material relating to the multigroup  $S_N$  equations.
- Appendix ???: addendum to Section 2, corresponding to additional material relating to the various polytope coordinate systems to be utilized as finite element basis functions.
- Appendix ???: addendum to Section ??, corresponding to additional material relating to DSA preconditioning on polytope grids.

## 2. 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 2.1, we present the 2D, linearly-complete, barycentric, polygonal basis functions that we will analyze in this dissertation. We give their properties along with a comparison to historical basis functions that are exact on non-polygonal elements (triangles and quadrilaterals). We then present in Section 2.2 the methodology to convert the barycentric polygonal basis functions presented in Section 2.1 into a serendipity space of basis functions with quadratic-completeness. Section 2.4 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 2.5. We then present concluding remarks for the chapter in Section 2.6.

### 2.1 Linear Basis Functions on 2D Polygons

Figure 2.1, 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 2.1.  $\alpha_j$  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,  $\alpha_j$  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  $\vec{x}_{j+1}$ .

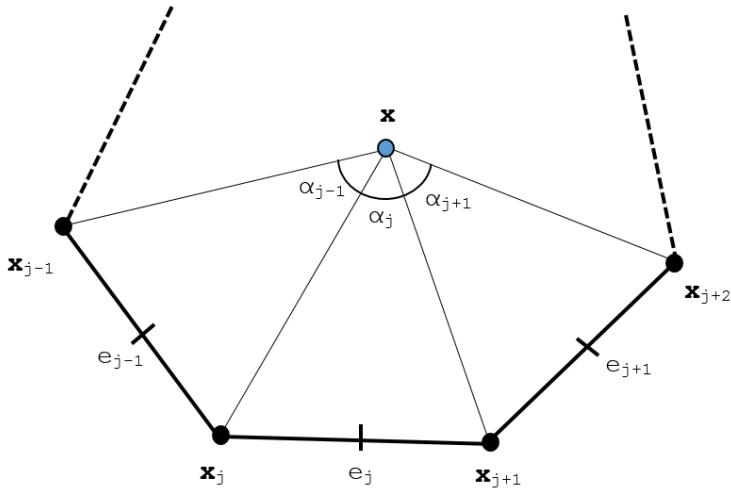


Figure 2.1: Arbitrary polygon with geometric properties used for 2D basis function generation.

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*,

$$\sum_{i=1}^{N_K} b_i(\vec{x}) = 1; \quad (2.1)$$

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

$$\sum_{i=1}^{N_K} b_i(\vec{x}) \vec{x}_i = \vec{x}; \quad (2.2)$$

and they will satisfy the *Lagrange property*,

$$b_i(\vec{x}_j) = \delta_{ij}. \quad (2.3)$$

$N_K$  is again the number of spatial degrees with measure in element  $K$ . Using the *partition of unity* of Eq. (2.1), we can rewrite Eqs. (2.1-2.2) into a separate, compact, vectorized form for completeness

$$\sum_{i=1}^{N_K} b_i(\vec{x}) \vec{c}_{i,1}(\vec{x}) = \vec{q}_1, \quad (2.4)$$

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

$$\vec{c}_{i,1}(\vec{x}) = \begin{bmatrix} 1 \\ x_i - x \\ y_i - y \end{bmatrix} \quad \text{and} \quad \vec{q}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad (2.5)$$

respectively. Equation (2.4) 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.

### 2.1.1 Traditional Linear Basis Functions - $\mathbb{P}_1$ and $\mathbb{Q}_1$ Spaces

Before presenting basis function sets applicable to polygonal finite elements, we first provide two common basis functions that are exact on triangles and convex

quadrilaterals: the  $\mathbb{P}_1$  and  $\mathbb{Q}_1$  spaces, respectively.

$$\begin{aligned} b_1(r, s) &= 1 - r - s \\ b_2(r, s) &= r \\ b_3(r, s) &= s \end{aligned} \tag{2.6}$$

and

$$\begin{aligned} b_1(r, s) &= (1 - r)(1 - s) \\ b_2(r, s) &= r(1 - s) \\ b_3(r, s) &= rs \\ b_4(r, s) &= (1 - r)s \end{aligned} \tag{2.7}$$

### 2.1.2 Wachspress Rational Basis Functions

$$b_j^W(\vec{x}) = \frac{w_j(\vec{x})}{\sum_i w_i(\vec{x})} \tag{2.8}$$

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

$$w_j(\vec{x}) = \frac{A(\vec{x}_{j-1}, \vec{x}_j, \vec{x}_{j+1})}{A(\vec{x}, \vec{x}_{j-1}, \vec{x}_j) A(\vec{x}, \vec{x}_j, \vec{x}_{j+1})} \tag{2.9}$$

### 2.1.3 Piecewise Linear (PWL) Basis Functions

$$b_j^{PWL}(x, y) = t_j(x, y) + \alpha_j^K t_c(x, y) \tag{2.10}$$

$t_j$  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.

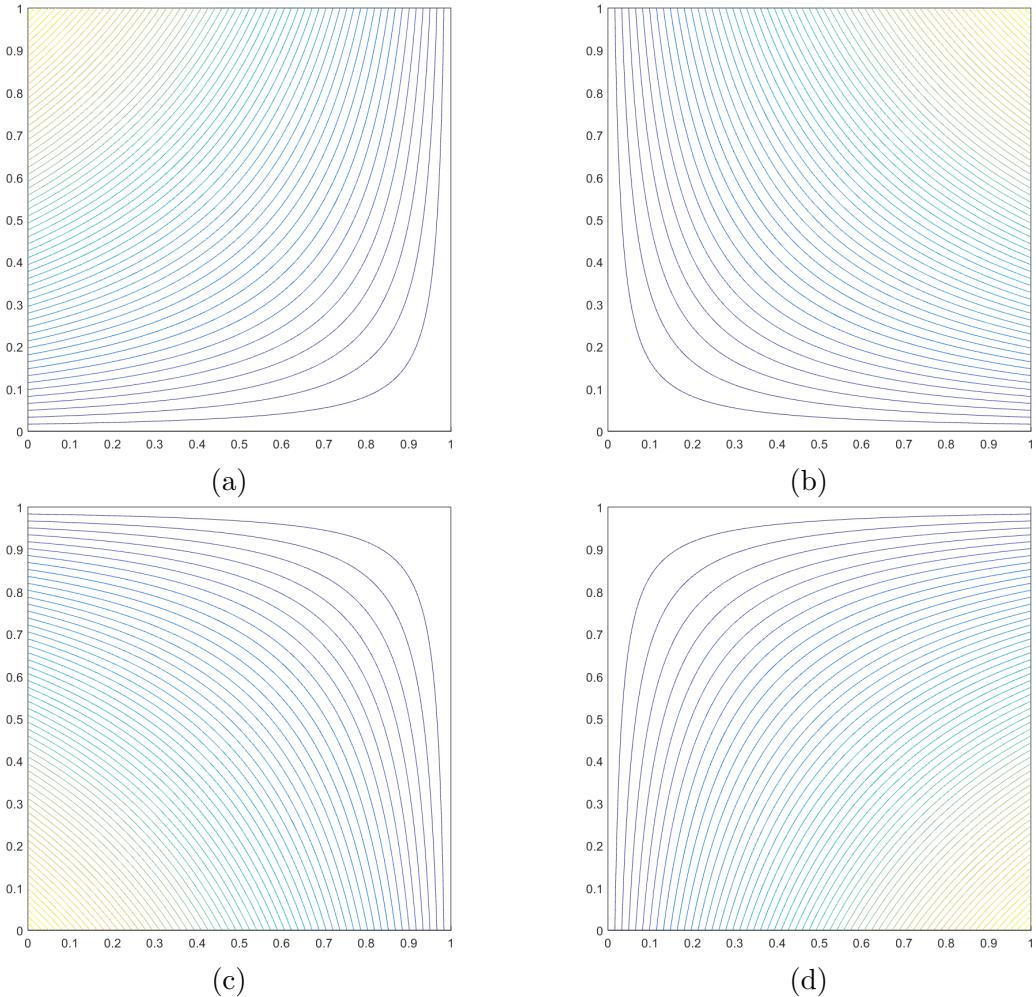


Figure 2.2: Contour plots of the linear Wachspress basis functions on the unit square for the vertices located at: (a)  $(0,1)$ , (b)  $(1,1)$ , (c)  $(0,0)$ , and (d)  $(1,0)$ .

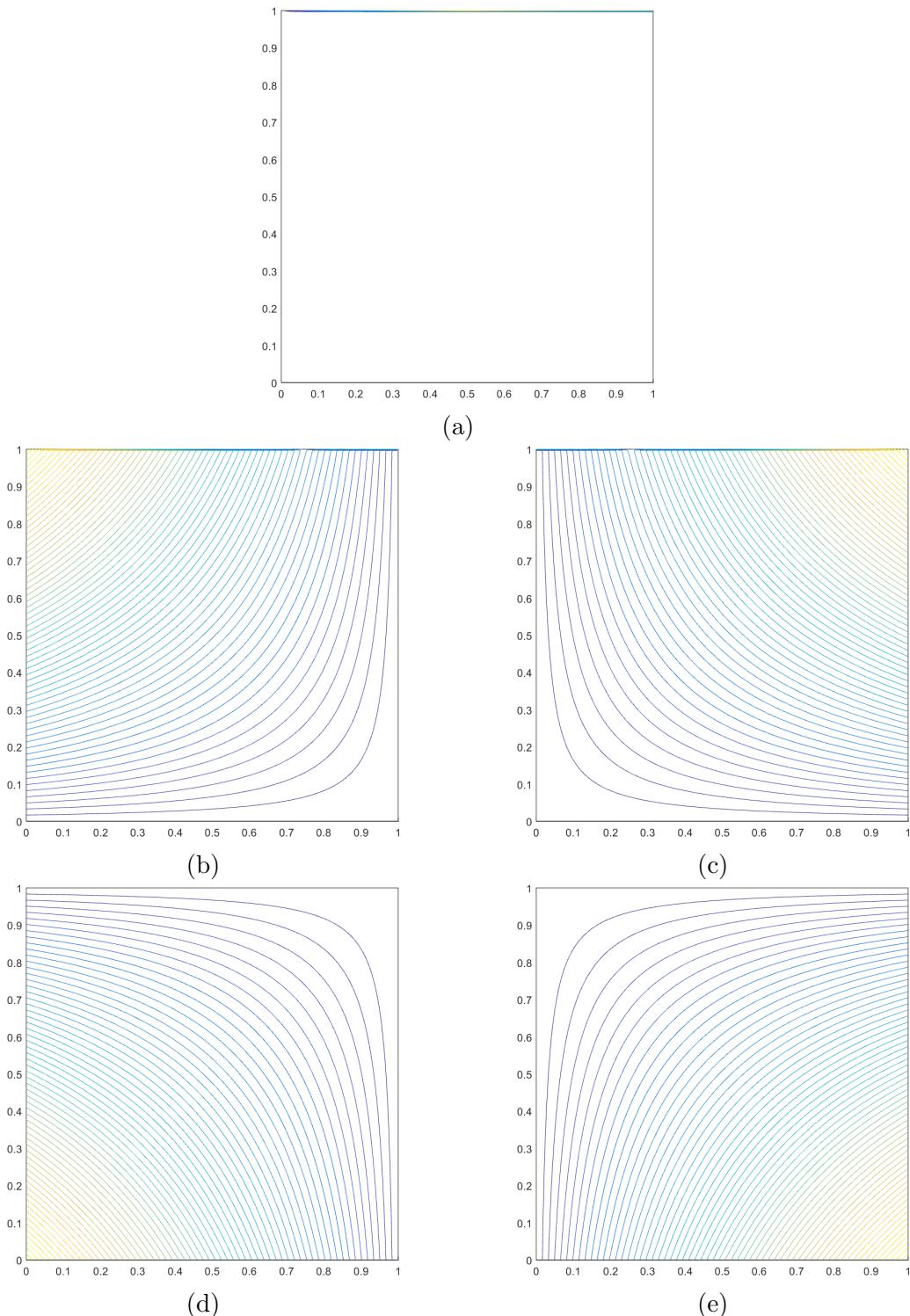


Figure 2.3: Contour plots of the linear Wachspress basis functions on the degenerate pentagon for the vertices located at: (a)  $(1/2, 1)$ , (b)  $(0, 1)$ , (c)  $(1, 1)$ , (d)  $(0, 0)$ , and (e)  $(1, 0)$ .

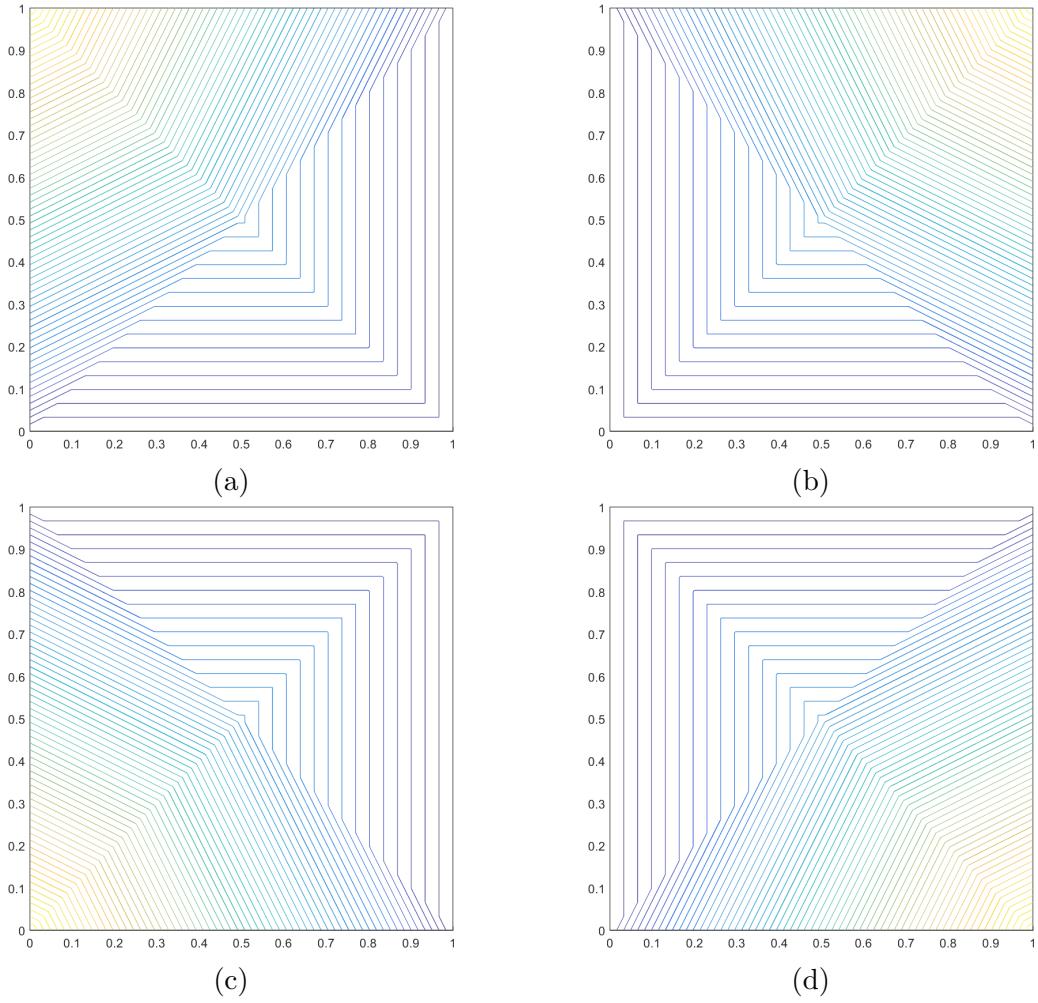


Figure 2.4: Contour plots of the linear PWL basis functions on the unit square for the vertices located at: (a)  $(0,1)$ , (b)  $(1,1)$ , (c)  $(0,0)$ , and (d)  $(1,0)$ .

$\alpha_j^K$  is the weight parameter for vertex  $j$  in cell  $K$ .

#### 2.1.4 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 [7, 8]. 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

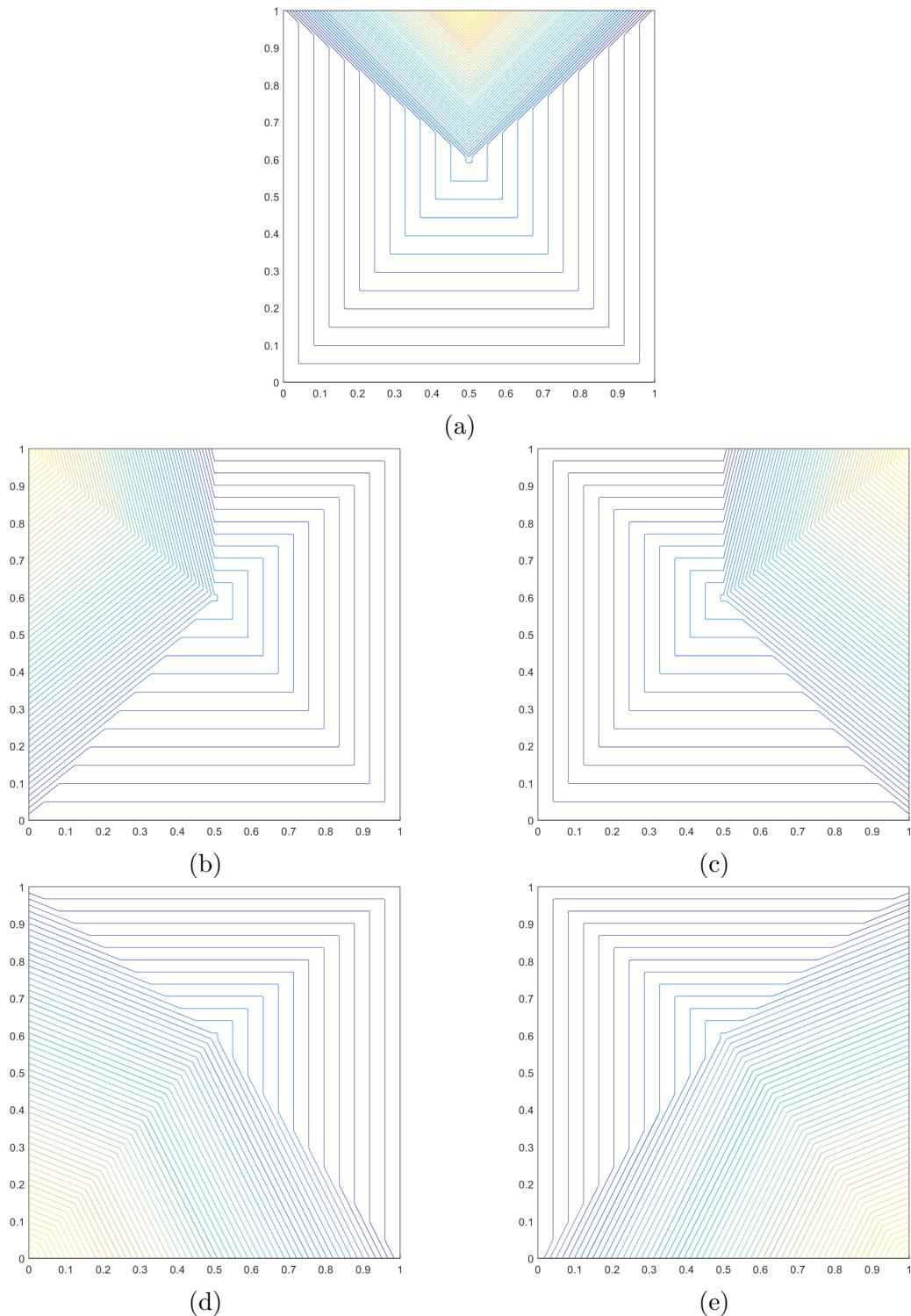


Figure 2.5: Contour plots of the linear PWL basis functions on the degenerate pentagon for the vertices located at: (a)  $(1/2, 1)$ , (b)  $(0, 1)$ , (c)  $(1, 1)$ , (d)  $(0, 0)$ , and (e)  $(1, 0)$ .

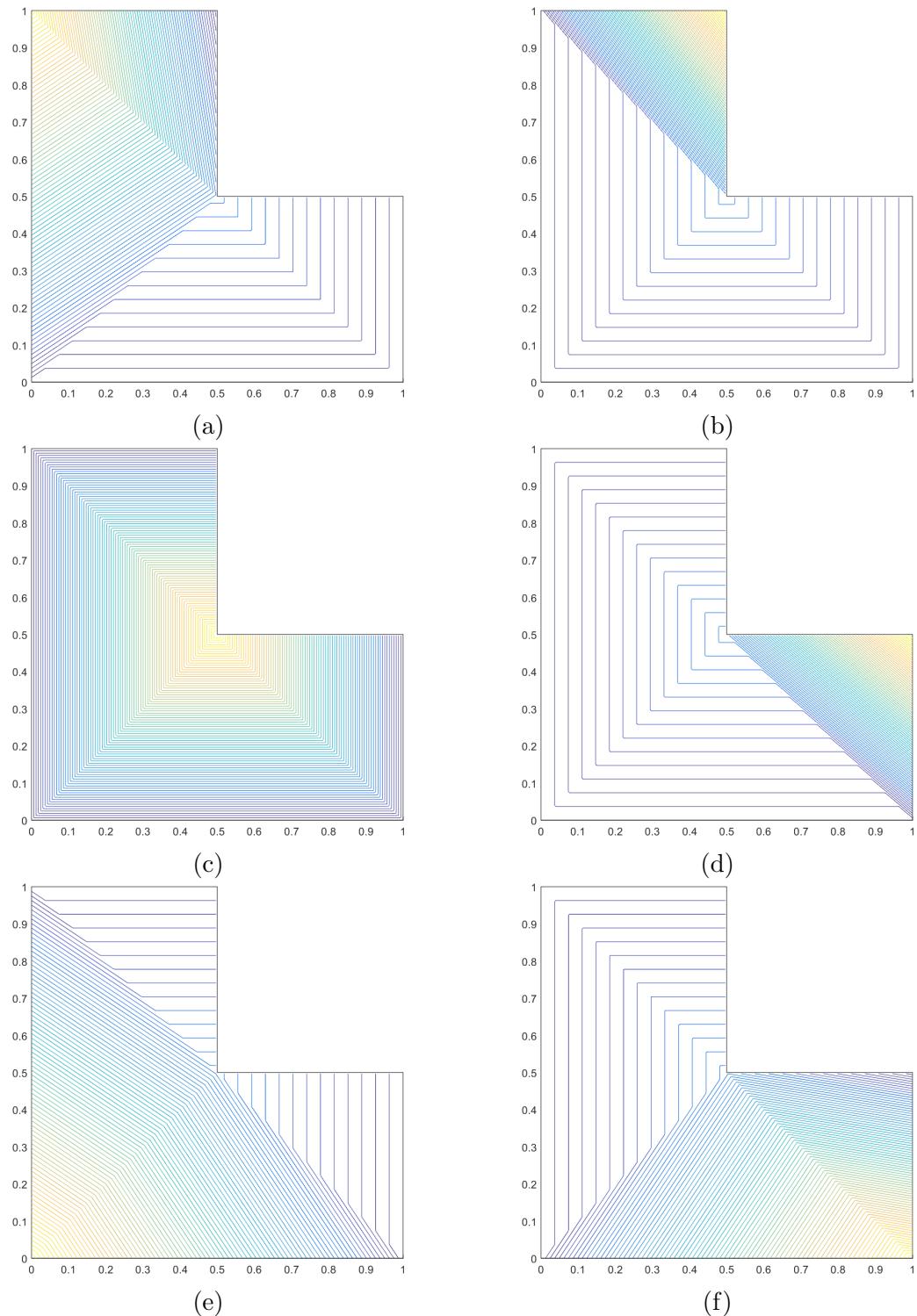


Figure 2.6: Contour plots of the linear PWL basis functions on the L-shaped domain for the vertices located at: (a)  $(0,1)$ , (b)  $(1/2,1)$ , (c)  $(1/2,1/2)$ , (d)  $(1,1/2)$ , (e)  $(0,0)$ , and (f)  $(1,0)$ .

polygon for use in computer aided graphic design.

$$\nabla^2 u = 0, \quad (2.11)$$

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

$$b_j^{MV}(\vec{x}) = \frac{w_j(\vec{x})}{\sum_i w_i(\vec{x})} \quad (2.12)$$

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

$$w_j(\vec{x}) = \frac{\tan(\alpha_{j-1}/2) + \tan(\alpha_j/2)}{|\vec{x}_j - \vec{x}|} \quad (2.13)$$

### 2.1.5 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) [9, 10, 11].

$$b_j^{ME}(\vec{x}) = \frac{w_j(\vec{x})}{\sum_i w_i(\vec{x})}. \quad (2.14)$$

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

$$w_j(\vec{x}) = m_j(\vec{x}) \exp(-\kappa \cdot (\vec{x}_j - \vec{x})), \quad (2.15)$$

where  $\kappa$  is a vector value of dimension  $d$  that will be explained shortly. In Eq. (2.15),  $m_j$  is called the prior distribution and is a key component of Bayesian inference [12, 13]. In the context of Eq. (2.15), 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

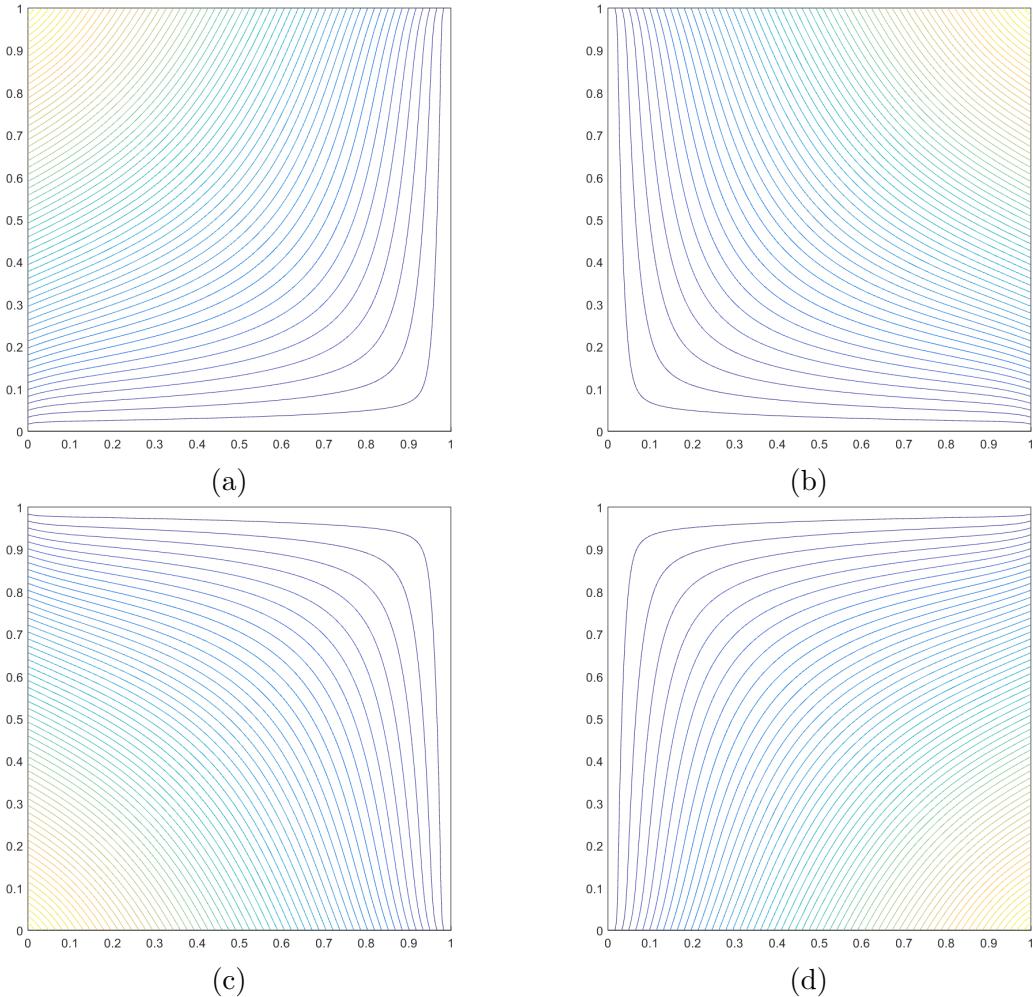


Figure 2.7: Contour plots of the linear mean value basis functions on the unit square for the vertices located at: (a)  $(0,1)$ , (b)  $(1,1)$ , (c)  $(0,0)$ , and (d)  $(1,0)$ .

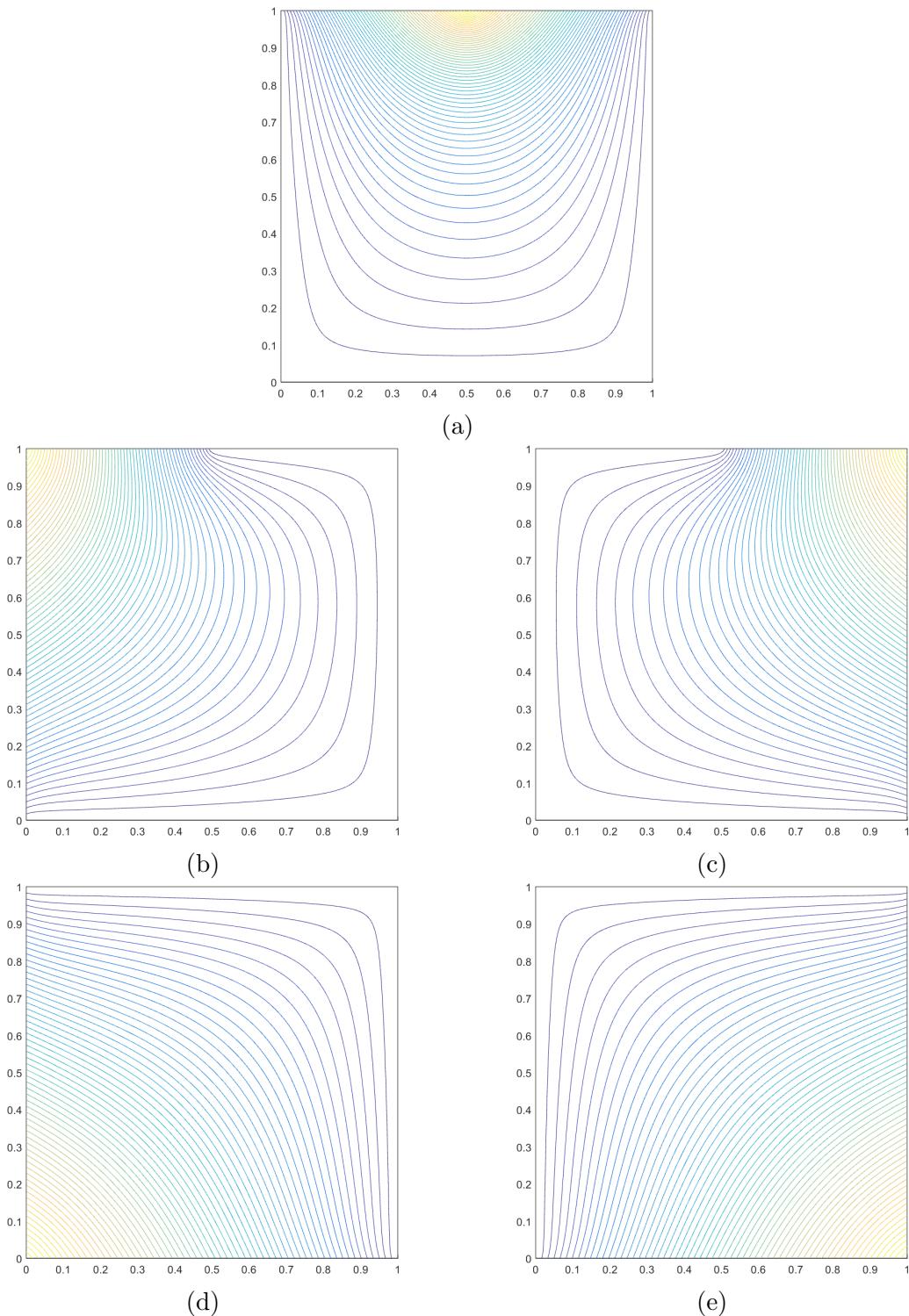


Figure 2.8: Contour plots of the linear mean value basis functions on the degenerate pentagon for the vertices located at: (a)  $(1/2, 1)$ , (b)  $(0, 1)$ , (c)  $(1, 1)$ , (d)  $(0, 0)$ , and (e)  $(1, 0)$ .

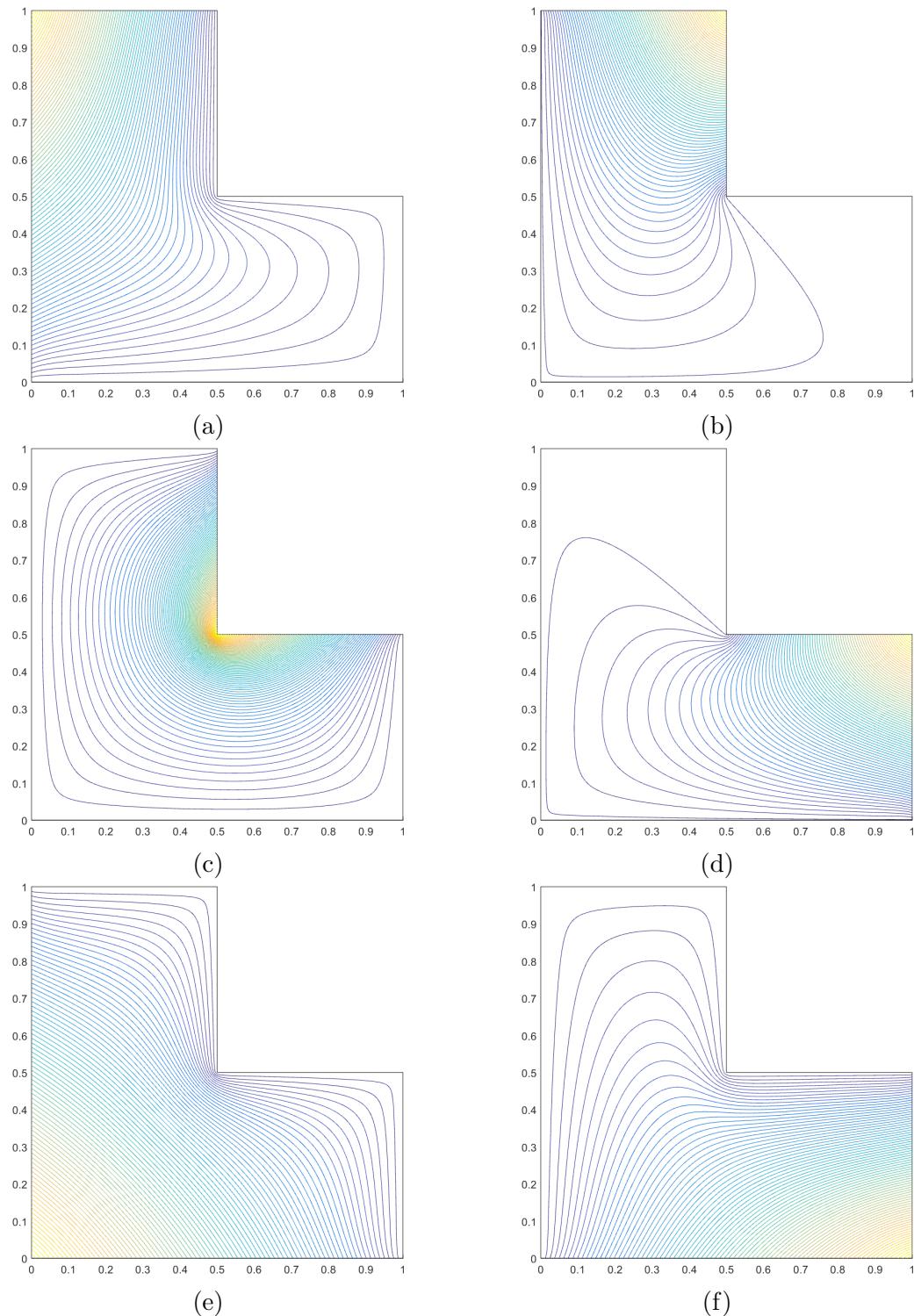


Figure 2.9: Contour plots of the linear mean value basis functions on the L-shaped domain for the vertices located at: (a)  $(0,1)$ , (b)  $(1/2,1)$ , (c)  $(1/2,1/2)$ , (d)  $(1,1/2)$ , (e)  $(0,0)$ , and (f)  $(1,0)$ .

Table 2.1: 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

tailored depending on the application and the numerical scheme employed.

$$m_j(\vec{x}) = \frac{\pi_j(\vec{x})}{\sum_k \pi_k(\vec{x})} \quad (2.16)$$

where

$$\pi_j(\vec{x}) = \prod_{i \neq j-1, j} \rho_j(\vec{x}) \quad (2.17)$$

where

$$\rho_j(\vec{x}) = \|\vec{x} - \vec{x}_j\| + \|\vec{x} - \vec{x}_{j+1}\| - \|\vec{x}_{j+1} - \vec{x}_j\| \quad (2.18)$$

### 2.1.6 Summary of 2D Linear Basis Functions on Polygons

## 2.2 Quadratic Serendipity Basis Functions on 2D Polygons

for the constant constraint,

$$\sum_{i=1}^{2N_K} b_i(\vec{x}) = 1, \quad (2.19)$$

for the linear constraint,

$$\sum_{i=1}^{2N_K} b_i(\vec{x}) (\vec{x}_i - \vec{x}) = \vec{0}, \quad (2.20)$$

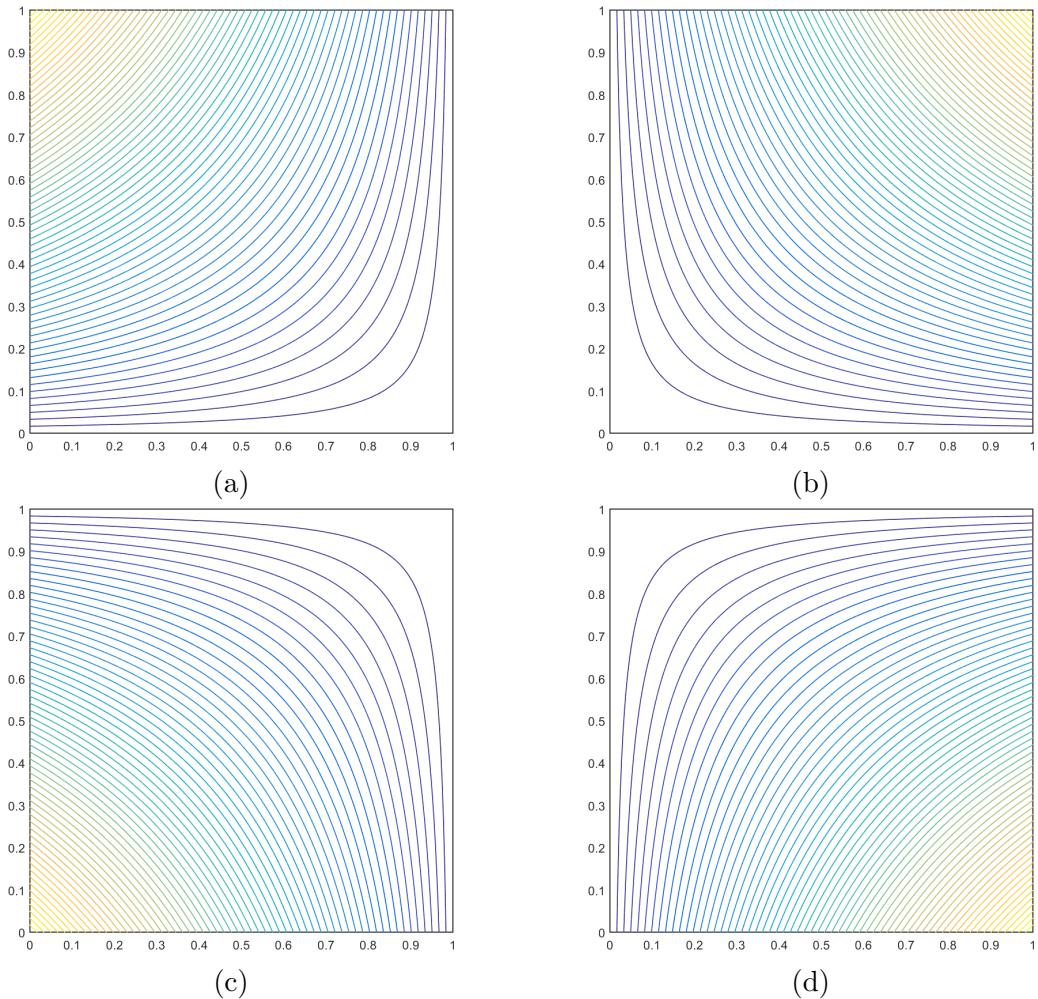


Figure 2.10: Contour plots of the linear maximum entropy basis functions on the unit square for the vertices located at: (a)  $(0,1)$ , (b)  $(1,1)$ , (c)  $(0,0)$ , and (d)  $(1,0)$ .

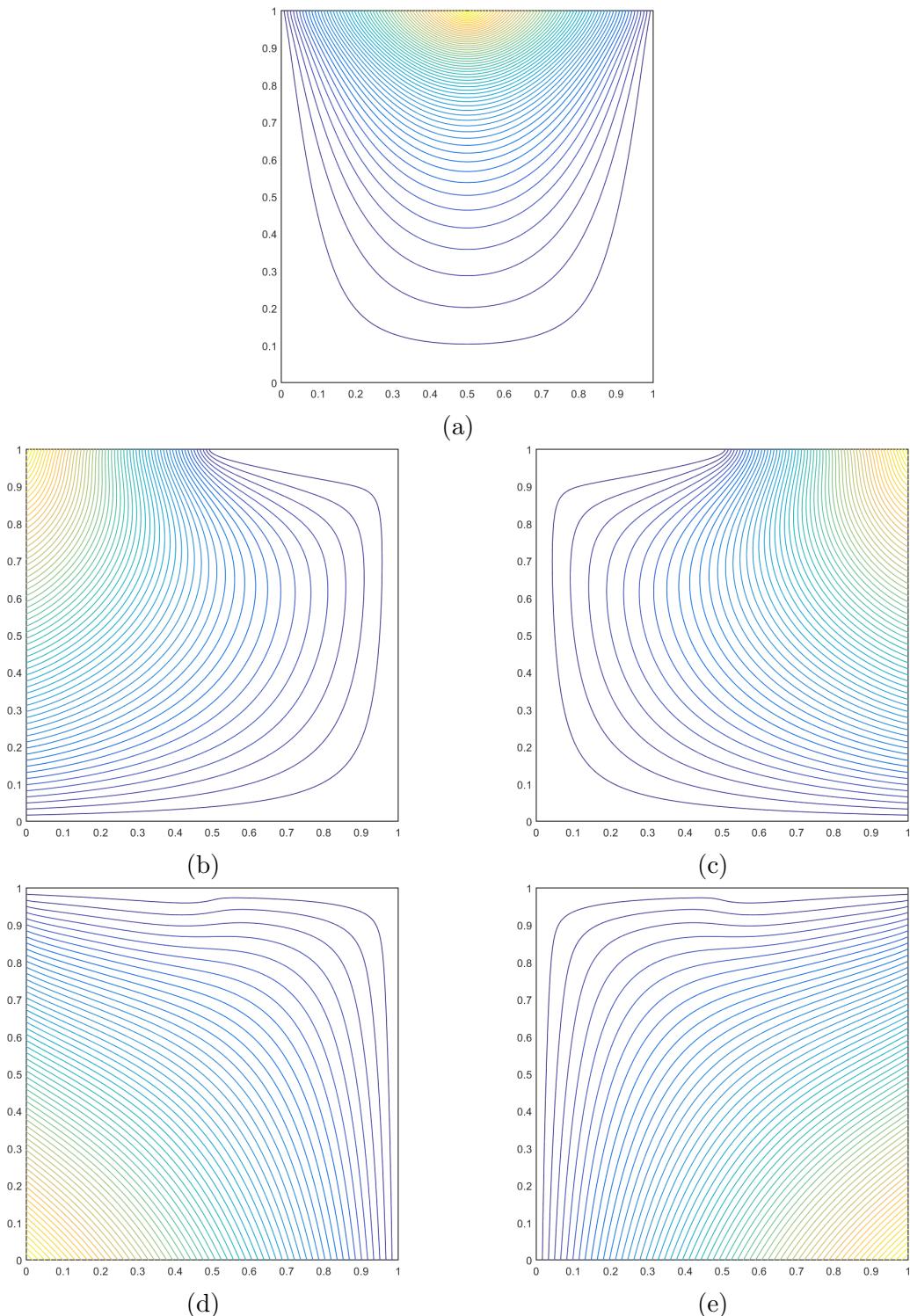


Figure 2.11: Contour plots of the linear maximum entropy basis functions on the degenerate pentagon for the vertices located at: (a)  $(1/2, 1)$ , (b)  $(0, 1)$ , (c)  $(1, 1)$ , (d)  $(0, 0)$ , and (e)  $(1, 0)$ .

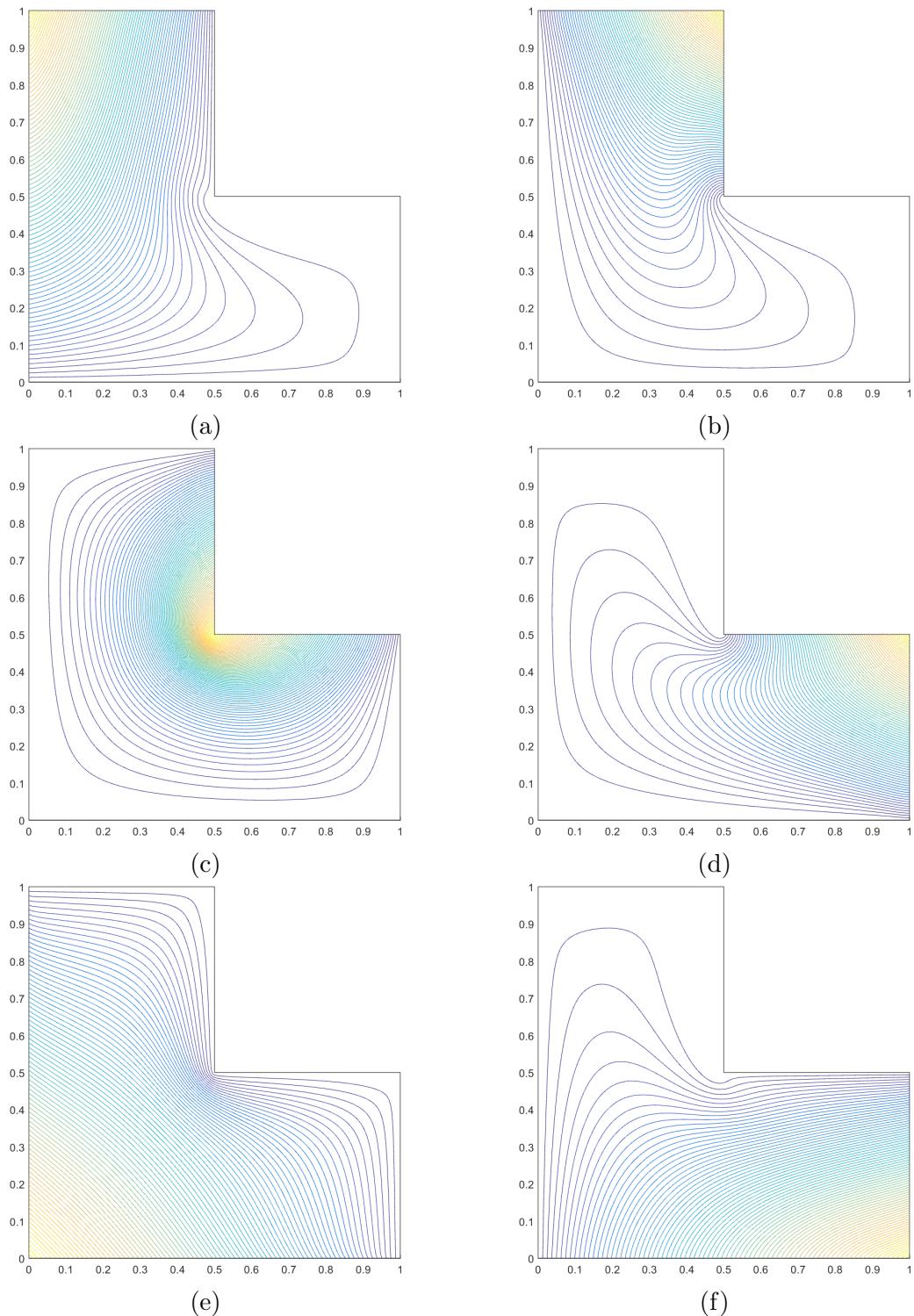


Figure 2.12: Contour plots of the linear maximum entropy basis functions on the L-shaped domain for the vertices located at: (a)  $(0,1)$ , (b)  $(1/2,1)$ , (c)  $(1/2,1/2)$ , (d)  $(1,1/2)$ , (e)  $(0,0)$ , and (f)  $(1,0)$ .

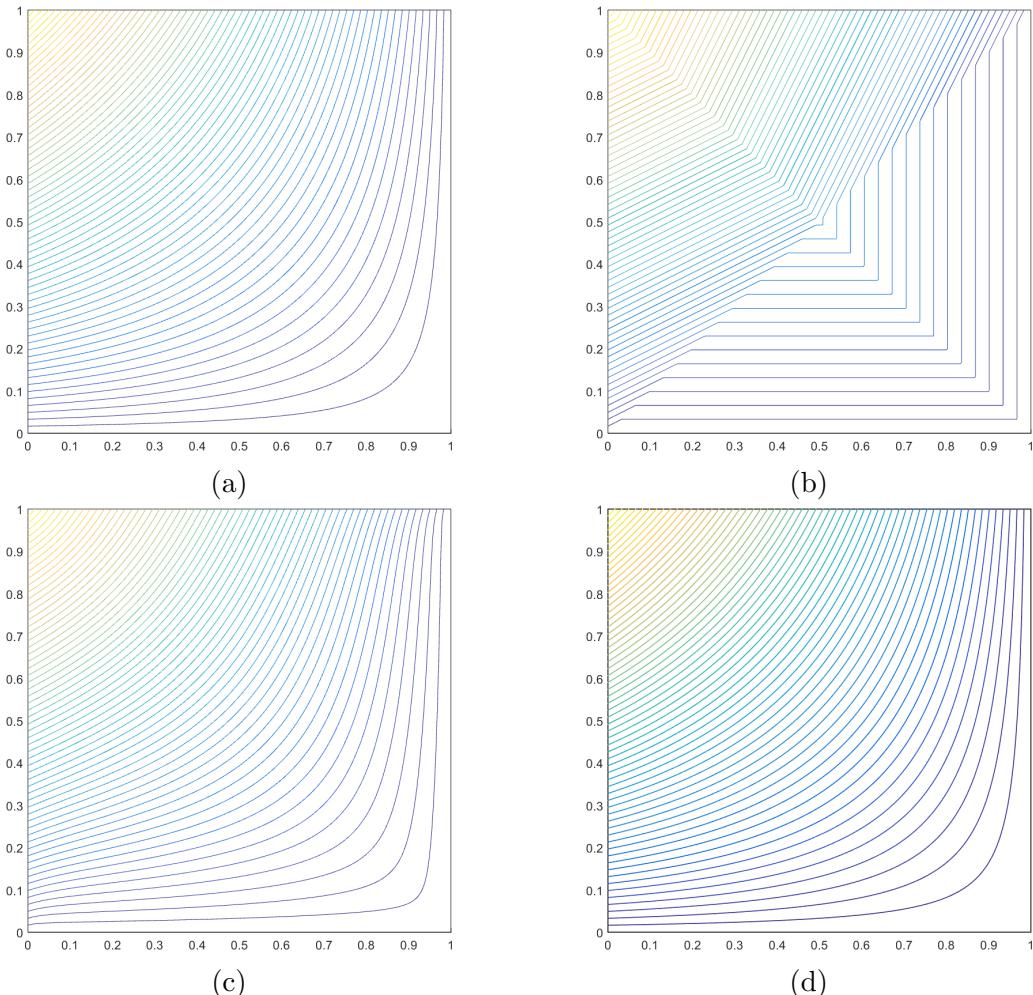


Figure 2.13: Contour plots of the different linear basis function on the unit square located at vertex  $(0,1)$ : (a) Wachspress, (b) PWL, (c) mean value, and (d) maximum entropy.

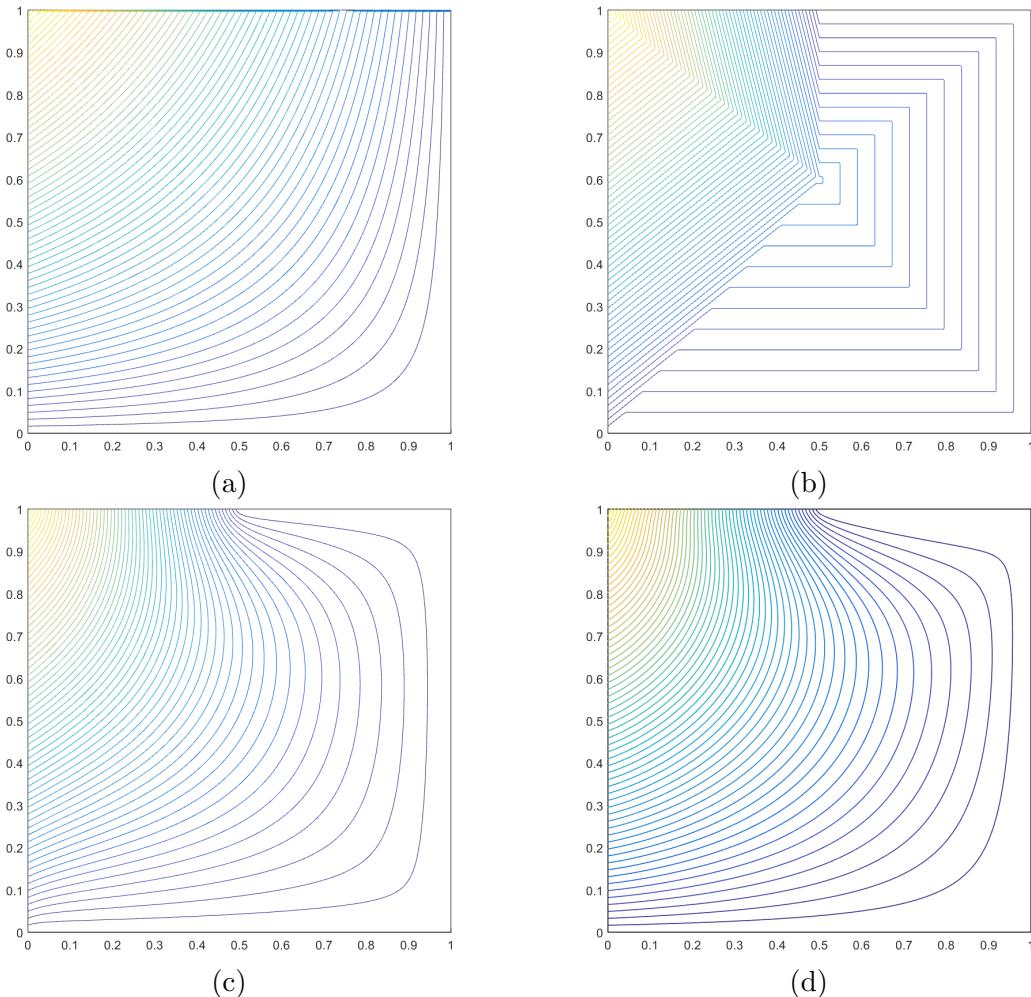


Figure 2.14: Contour plots of the different linear basis function on the degenerate pentagon located at vertex  $(0,1)$ : (a) Wachspress, (b) PWL, (c) mean value, and (d) maximum entropy.

and for the quadratic constraint,

$$\sum_{i=1}^{2N_K} b_i(\vec{x}) \left[ (\vec{x}_i - \vec{x}) \otimes (\vec{x}_i - \vec{x}) - \vec{d}_i \right] = \vec{0}. \quad (2.21)$$

In Eq. (2.21), the nodal coefficients,  $\vec{d}_i$ , have the interpretation of being the nodal gap functions [14]. This ensures the reproduction of our second-order convex approximation scheme. These nodal gap functions have the form for the vertex functions of,

$$\vec{d}_i = \vec{0}, \quad i \in (1, N_K), \quad (2.22)$$

and the form on the face midpoints of,

$$\vec{d}_{a+N_K} = \frac{(\vec{x}_{a+1} - \vec{x}_a) \otimes (\vec{x}_{a+1} - \vec{x}_a)}{4}, \quad 1 \leq a \leq N_K. \quad (2.23)$$

$$\sum_{i=1}^{N_K} b_i(\vec{x}) \vec{c}_{i,2}(\vec{x}) = \vec{q}_2, \quad (2.24)$$

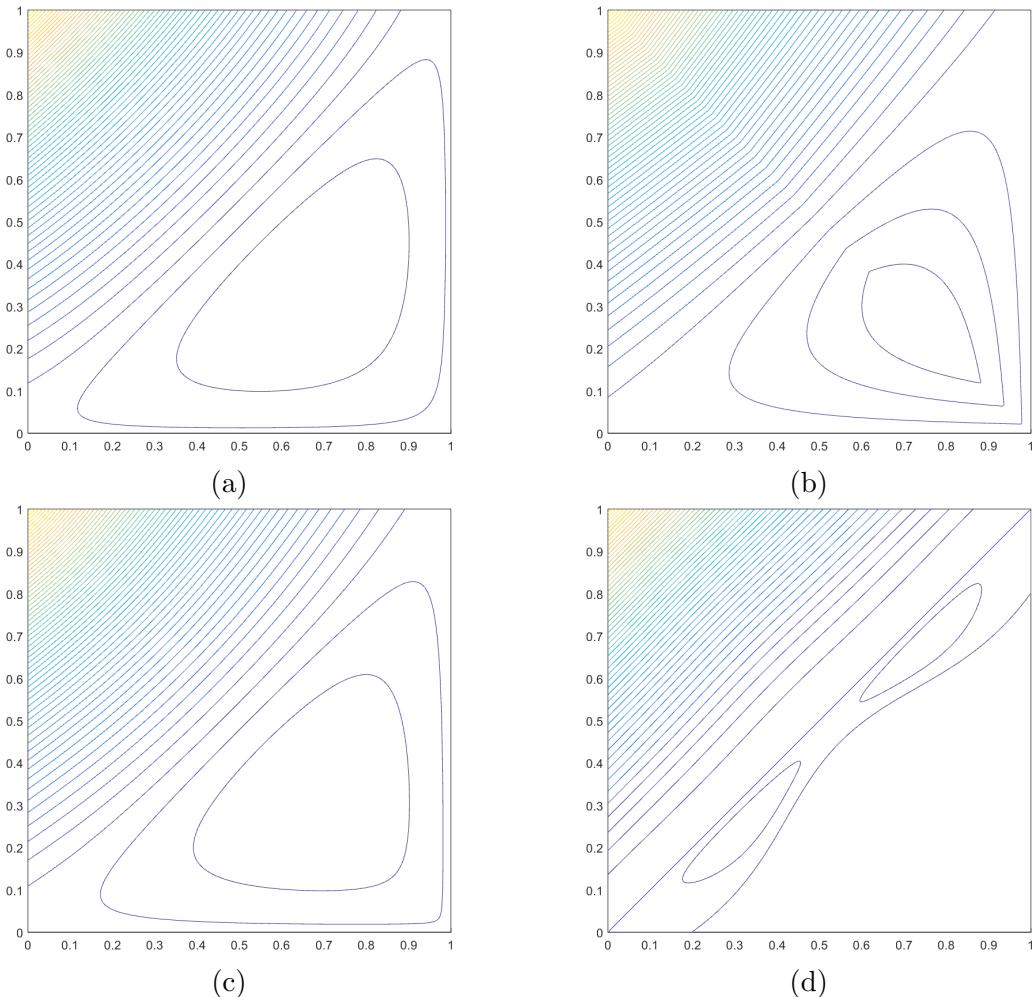


Figure 2.15: Contour plots of the different quadratic serendipity basis function on the unit square located at vertex  $(0,1)$ : (a) Wachspress, (b) PWL, (c) mean value, and (d) maximum entropy.

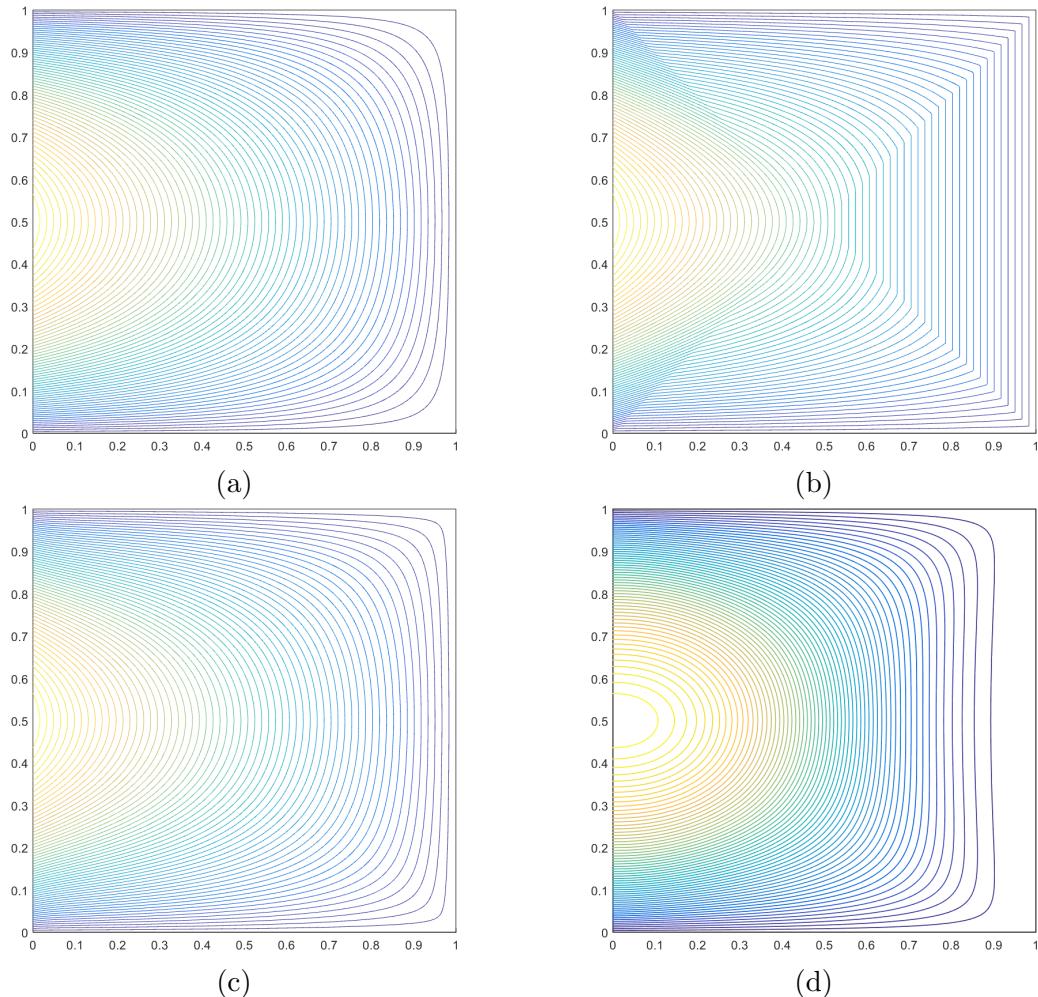


Figure 2.16: Contour plots of the different quadratic serendipity basis function on the unit square at a mid-face node located at  $(0,1/2)$ : (a) Wachspress, (b) PWL, (c) mean value, and (d) maximum entropy.

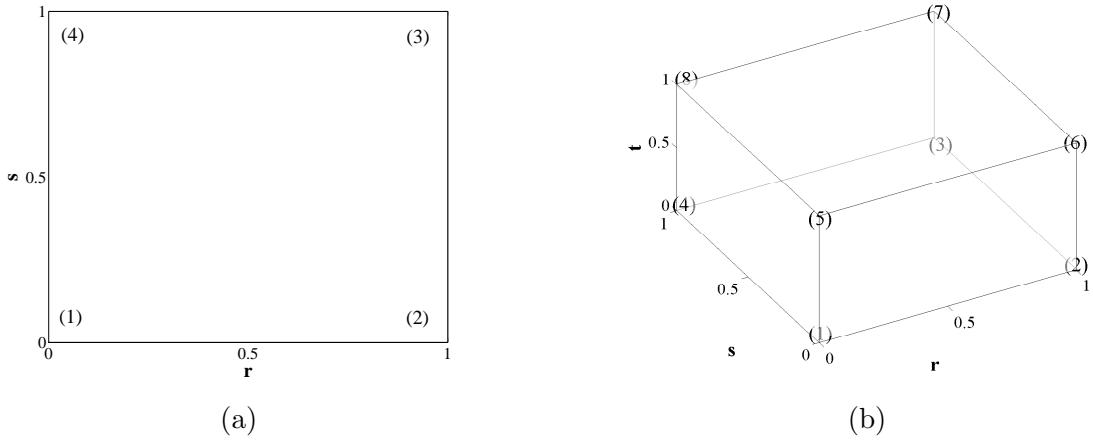


Figure 2.17: Vertex structure for the (a) unit square and (b) unit cube.

### 2.2.1 Analytical Form of the Quadratic Serendipity Extension of the PWL Functions

### 2.2.2 Summary of 2D Quadratic Serendipity Basis Functions on Polygons

## 2.3 Integrating the Arbitrary 2D Polygonal Elements

### 2.4 Linear Basis Functions on 3D Polyhedra

#### 2.4.1 3D Linear and TriLinear Basis Functions

$$b_1(r, s) = 1 - r - s - t$$

$$b_2(r, s) = r \tag{2.25}$$

$$b_3(r, s) = s$$

$$b_4(r, s) = t$$

and

$$\begin{aligned}
b_1(r, s, t) &= (1 - r)(1 - s)(1 - t) \\
b_2(r, s, t) &= r(1 - s)(1 - t) \\
b_3(r, s, t) &= rs(1 - t) \\
b_4(r, s, t) &= (1 - r)s(1 - t) \\
b_5(r, s, t) &= (1 - r)(1 - s)t \\
b_6(r, s, t) &= r(1 - s)t \\
b_7(r, s, t) &= rst \\
b_8(r, s, t) &= (1 - r)st
\end{aligned} \tag{2.26}$$

#### 2.4.2 3D Piecewise Linear (PWL) Basis Functions

The 3D PWL basis functions share a similar form to the 2D PWL basis functions.

$$b_j(x, y, z) = t_j(x, y, z) + \sum_{f=1}^{F_j} \beta_f^j t_f(x, y, z) + \alpha_j^K t_c(x, y, z) \tag{2.27}$$

$t_j$  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 [15], we also choose to assume the cell and face weighting parameters are

$$\alpha_{K,j} = \frac{1}{N_K} \quad \text{and} \quad \beta_{f,j} = \frac{1}{N_f}, \tag{2.28}$$

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  $\alpha$  and  $\beta$  for each cell and face, respectively. This assumption of the cell weight function holds for both 2D and 3D.

## 2.5 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 2.5.1. Next, we present some convergence properties of the basis sets using the method of manufactured solutions (MMS) in Section 2.5.3. 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 2.5.5.

### 2.5.1 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. (2.2)), 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,

$$\mu \frac{\partial \Psi}{\partial x} + \eta \frac{\partial \Psi}{\partial y} + \sigma_t \Psi = Q(x, y, \mu, \eta), \quad (2.29)$$

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:

$$\begin{aligned} \Psi(x, y, \mu, \eta) &= ax + by + c\mu + d\eta + e \\ \Phi(x, y) &= 2\pi(ax + by + e) \end{aligned} \quad . \quad (2.30)$$

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:

$$\sum_q w_q = 2\pi \quad \text{and} \quad \sum_q w_q \begin{bmatrix} \mu_q \\ \eta_q \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \quad (2.31)$$

The sum of the quadrature weights is handled by simply renormalizing those that are generated in Section ?? to  $2\pi$ .

Our boundary conditions for all inflow boundaries are then uniquely determined by the angular flux solution of Eq. (2.30). Inserting the angular flux solution of Eq. (2.30) into Eq. (2.29), we obtain the distributed source that will produce our exactly-linear solution space:

$$Q(x, y, \mu, \eta) = a\mu + b\eta + \sigma_t(c\mu + d\eta) + \sigma_t(ax + by + e). \quad (2.32)$$

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

$$\begin{aligned} a &= -c\sigma_t, \\ b &= -d\sigma_t. \end{aligned} \quad (2.33)$$

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 [16, 17]. 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 [18]. 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. (2.30) 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 2.18, ??, ??, ??, 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.

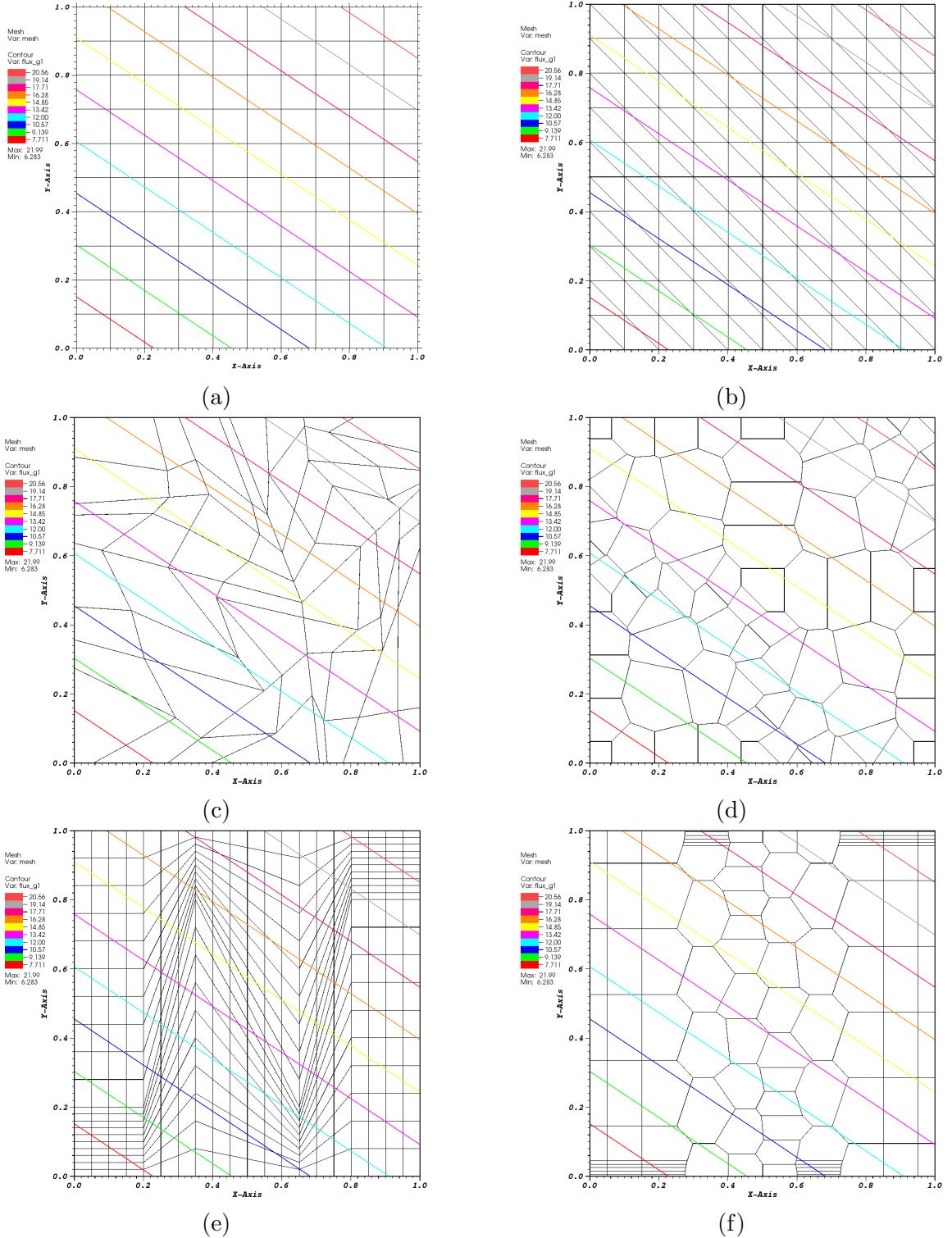


Figure 2.18: Contour plots of the exactly-linear solution with the Wachspress basis functions on (a) cartesian mesh, (b) ordered-triangular mesh, (c) quadrilateral shestakov mesh, (d) sinusoidal polygonal mesh, (e) quadrilateral z-mesh, and (f) polygonal z-mesh.

### 2.5.2 Two-Dimensional Exactly-Quadratic Transport Solutions

$$\begin{aligned}\Psi(x, y, \mu, \eta) &= a + bx + cy + dxy + ex^2 + fy^2 \\ \Phi(x, y) &= 2\pi(a + bx + cy + dxy + ex^2 + fy^2)\end{aligned}. \quad (2.34)$$

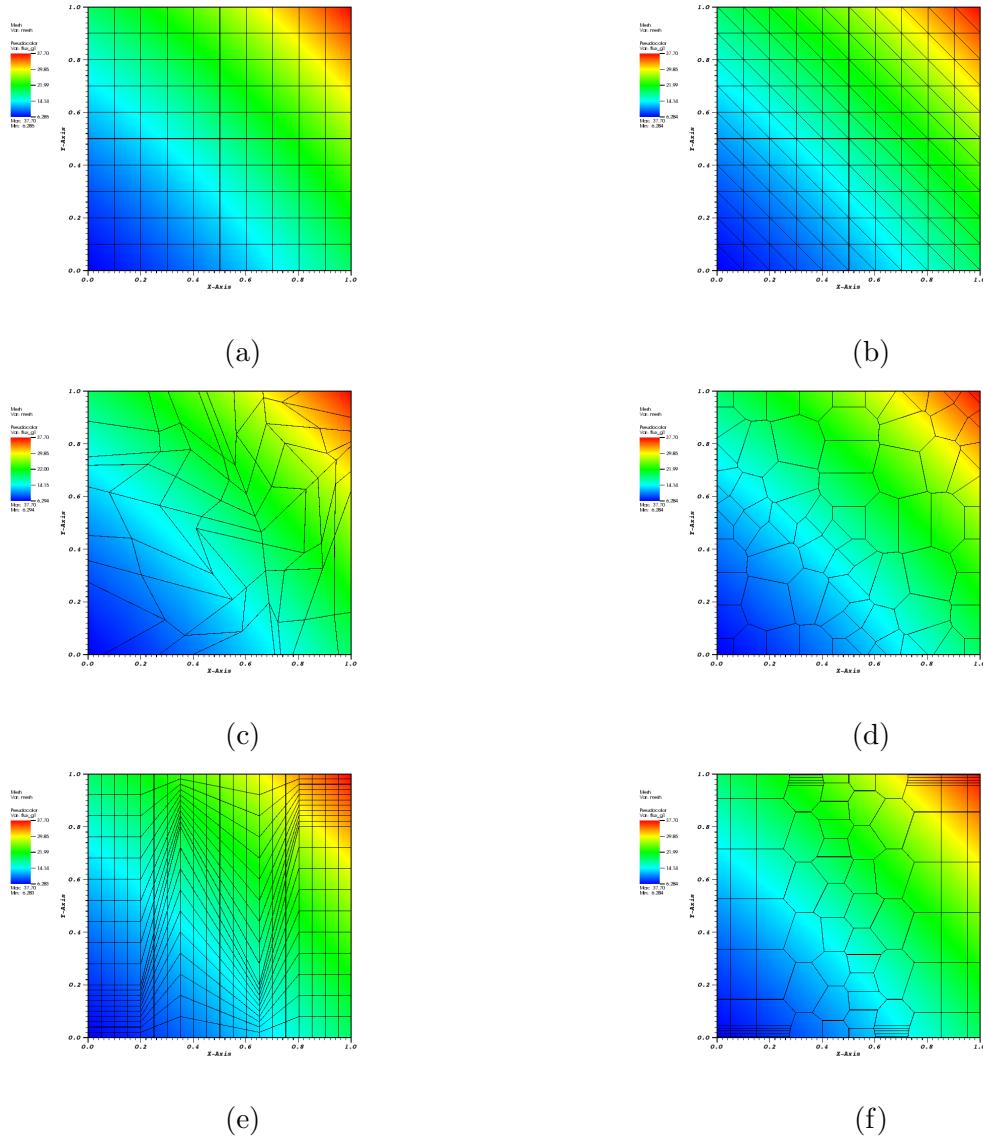


Figure 2.19: Plots of the exactly-quadratic solution with the quadratic serendipity maximum entropy basis functions on (a) cartesian mesh, (b) ordered-triangular mesh, (c) quadrilateral shestakov mesh, (d) sinusoidal polygonal mesh, (e) quadrilateral z-mesh, and (f) polygonal z-mesh.

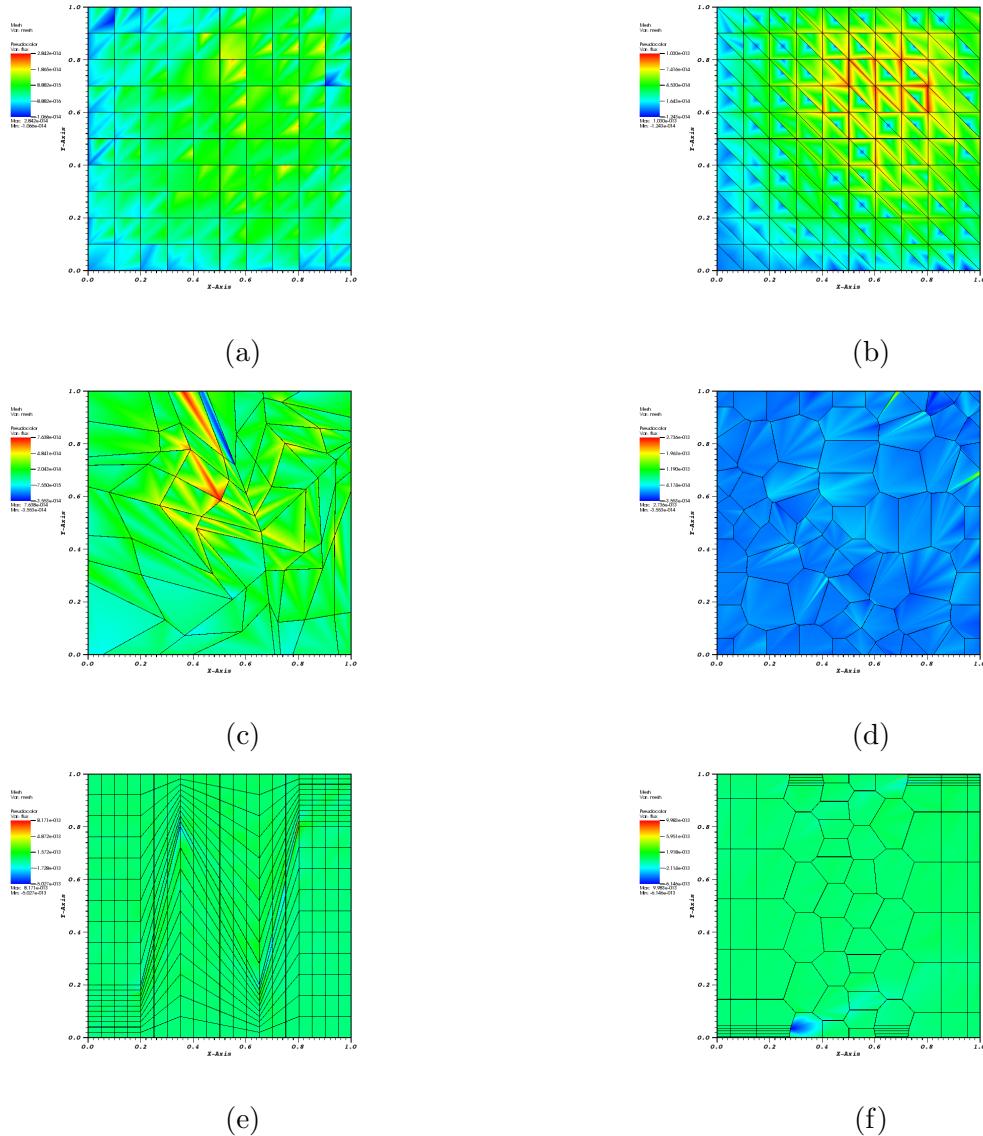


Figure 2.20: Plots of the error of the exactly-quadratic solution with the quadratic serendipity Wachspress basis functions on (a) cartesian mesh, (b) ordered-triangular mesh, (c) quadrilateral shestakov mesh, (d) sinusoidal polygonal mesh, (e) quadrilateral z-mesh, and (f) polygonal z-mesh.

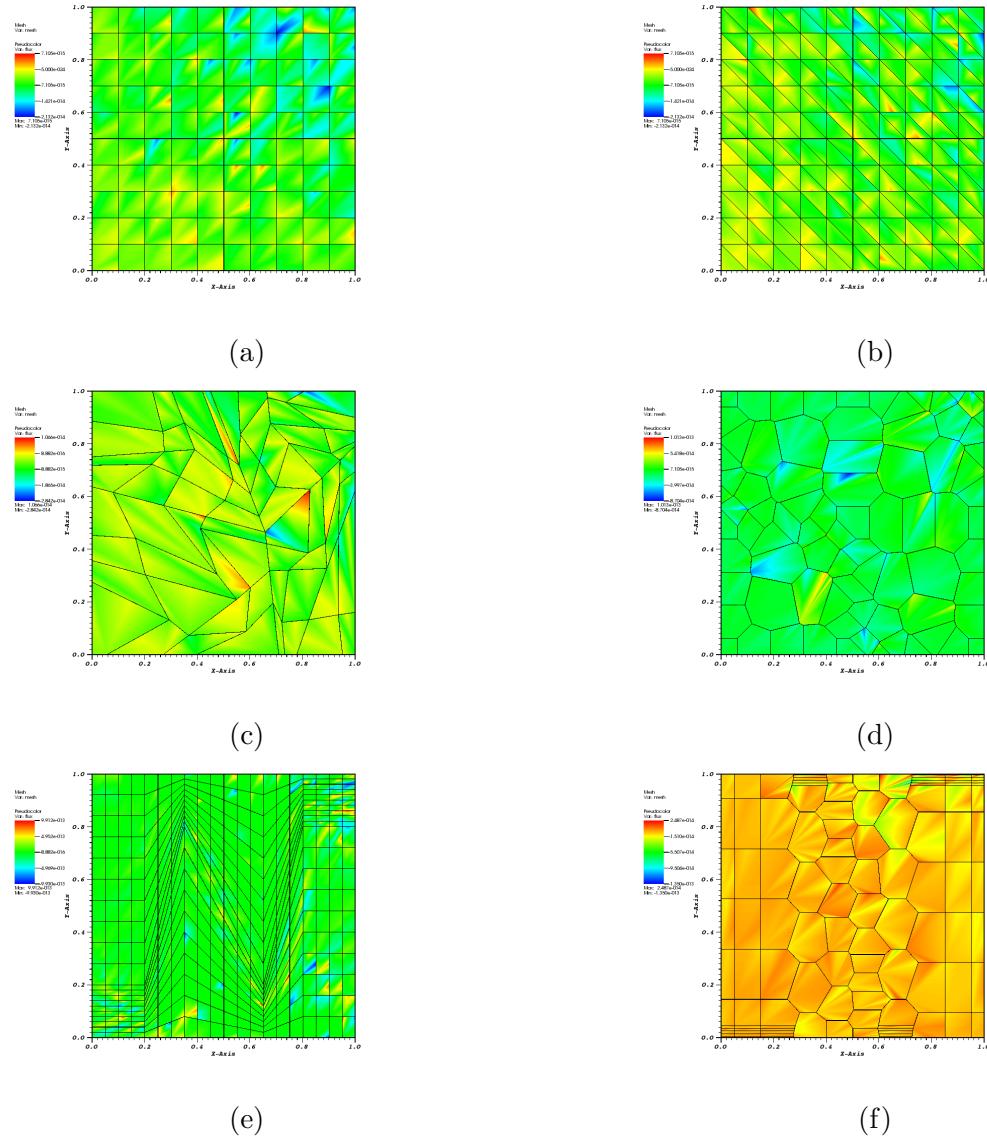


Figure 2.21: Plots of the error of the exactly-quadratic solution with the quadratic serendipity PWL basis functions on (a) cartesian mesh, (b) ordered-triangular mesh, (c) quadrilateral shestakov mesh, (d) sinusoidal polygonal mesh, (e) quadrilateral z-mesh, and (f) polygonal z-mesh.

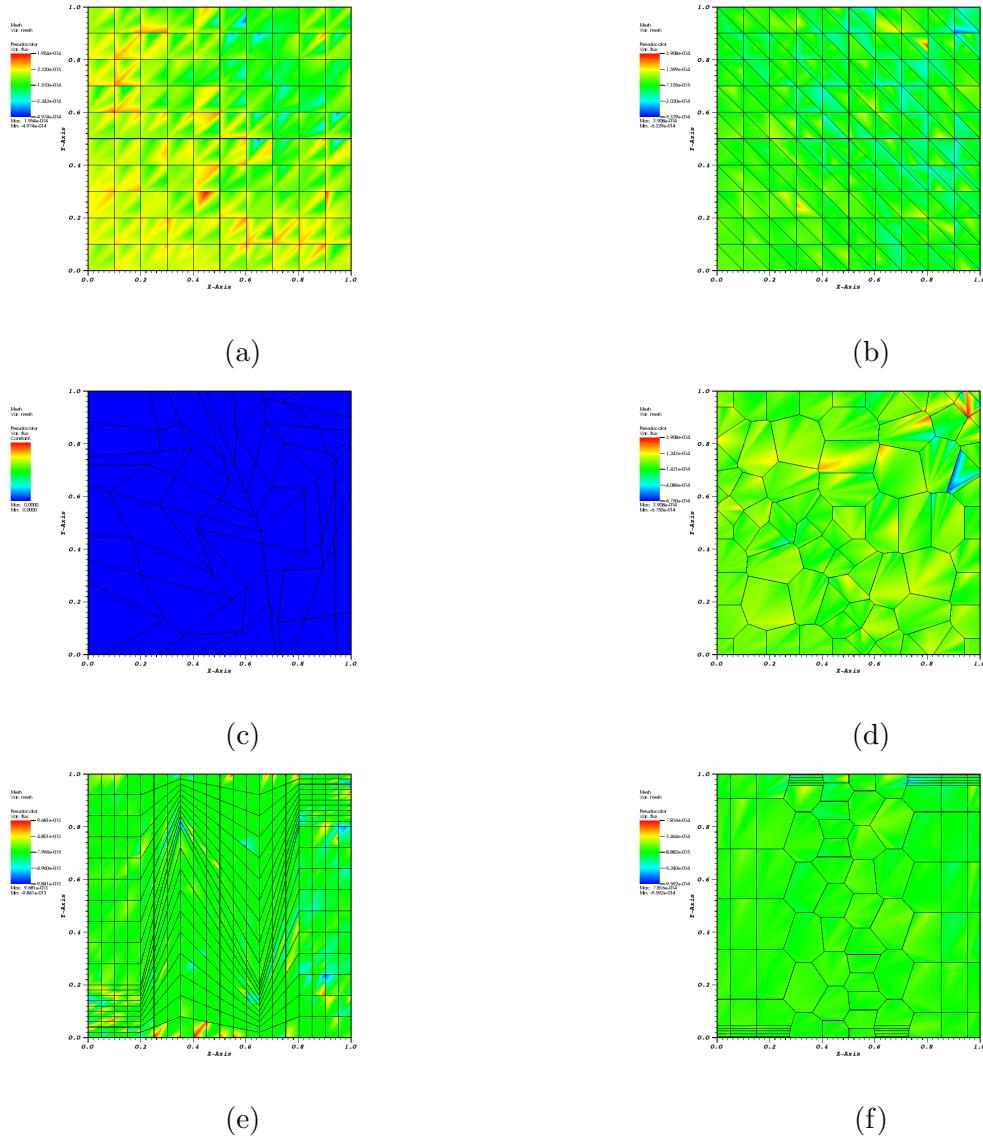


Figure 2.22: Plots of the error of the exactly-quadratic solution with the quadratic serendipity mean value basis functions on (a) cartesian mesh, (b) ordered-triangular mesh, (c) quadrilateral shestakov mesh, (d) sinusoidal polygonal mesh, (e) quadrilateral z-mesh, and (f) polygonal z-mesh.

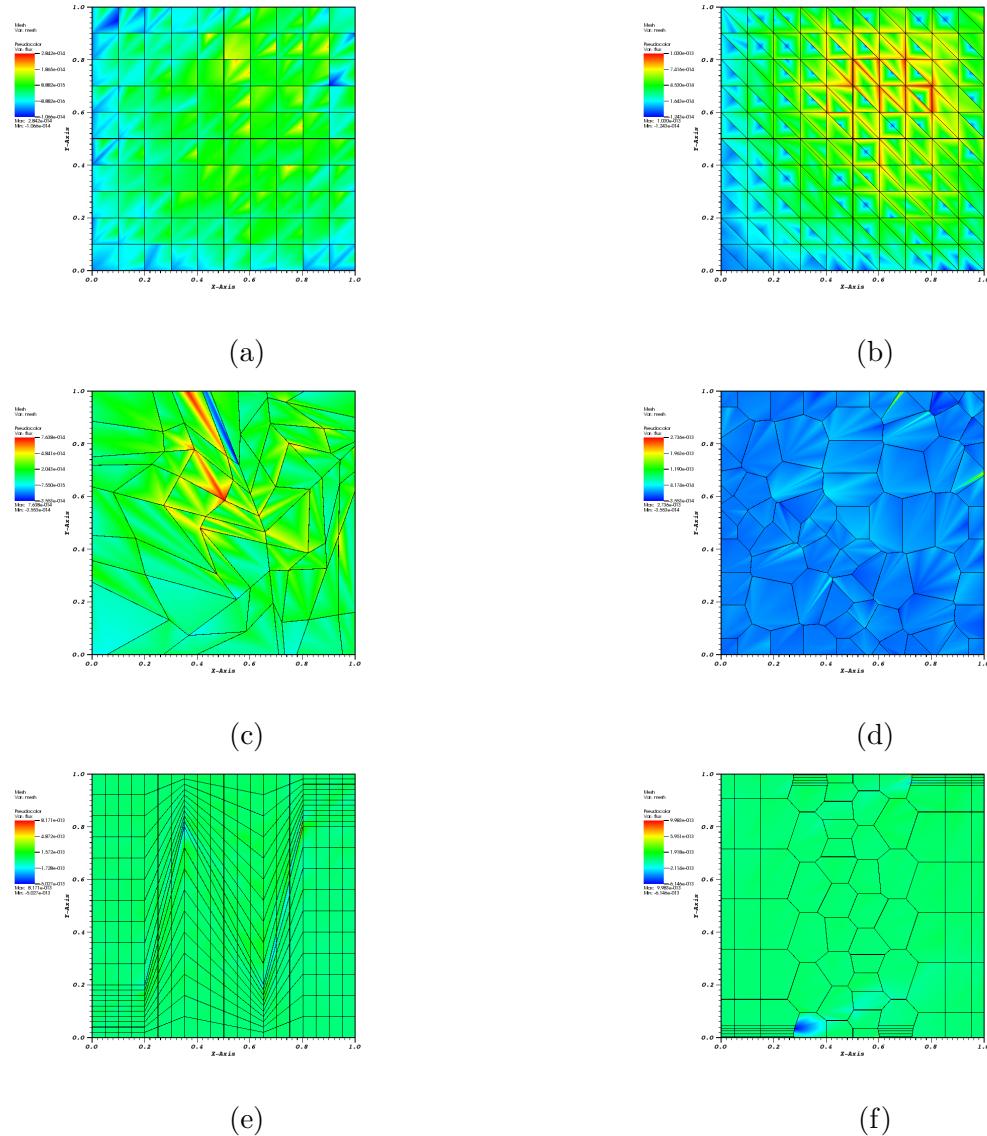


Figure 2.23: Plots of the error of the exactly-quadratic solution with the quadratic serendipity maximum entropy basis functions on (a) cartesian mesh, (b) ordered-triangular mesh, (c) quadrilateral shestakov mesh, (d) sinusoidal polygonal mesh, (e) quadrilateral z-mesh, and (f) polygonal z-mesh.

### 2.5.3 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 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  $C^\infty$  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,

$$\begin{aligned}\Psi^s(x, y) &= \sin(\nu \frac{\pi x}{L_x}) \sin(\nu \frac{\pi y}{L_y}), \\ \Phi^s(x, y) &= 2\pi \sin(\nu \frac{\pi x}{L_x}) \sin(\nu \frac{\pi y}{L_y}),\end{aligned}\tag{2.35}$$

where  $\nu$  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,

$$\begin{aligned}\Psi^g(x, y) &= C_M x(L_x - x)y(L_y - y) \exp\left(-\frac{(x - x_0)^2 + (y - y_0)^2}{\gamma}\right), \\ \Phi^g(x, y) &= 2\pi C_M x(L_x - x)y(L_y - y) \exp\left(-\frac{(x - x_0)^2 + (y - y_0)^2}{\gamma}\right),\end{aligned}\tag{2.36}$$

where the constants in the equations are:

$$C_M = \frac{100}{L_x^2 L_y^2} \quad \gamma = \frac{L_x L_y}{100}.\tag{2.37}$$

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,  $\nu$ , 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  $-2\pi$  and  $2\pi$ , respectively, and the gaussian solution will have a global maximum of  $\frac{225}{32}\pi \approx 22.1$ .

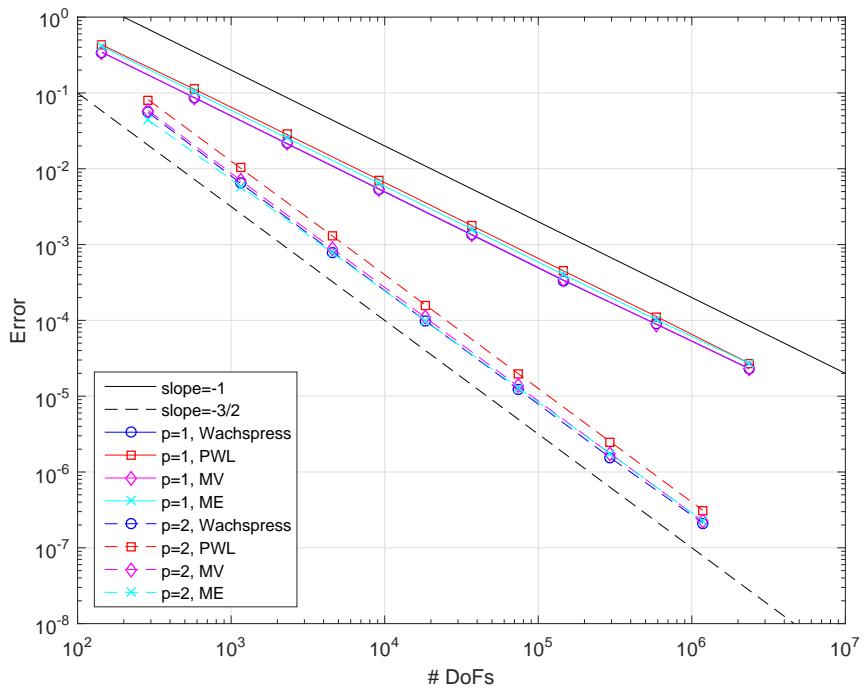


Figure 2.24: Convergence rates for the sinusoid MMS problem on a Cartesian mesh.

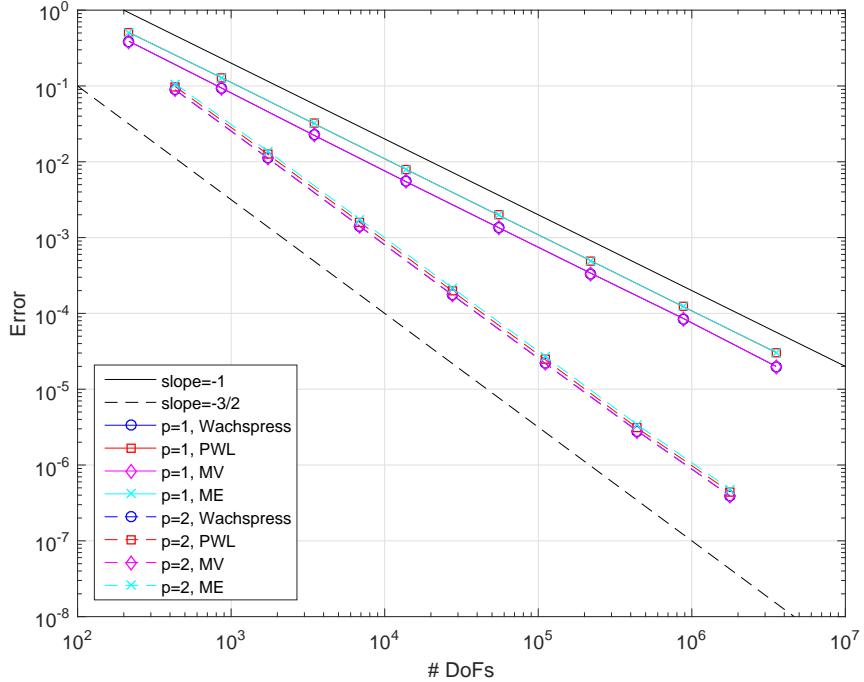


Figure 2.25: Convergence rates for the sinusoid MMS problem on an ordered triangular mesh.

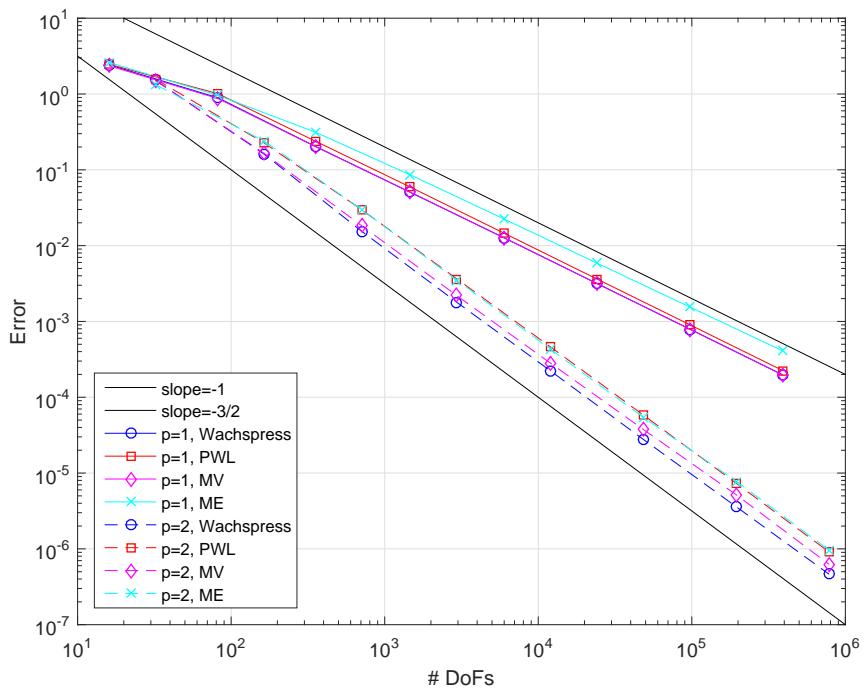


Figure 2.26: Convergence rates for the sinusoid MMS problem on a regular polygonal mesh.

#### 2.5.4 Convergence Rate Analysis in a Purely-Absorbing Medium

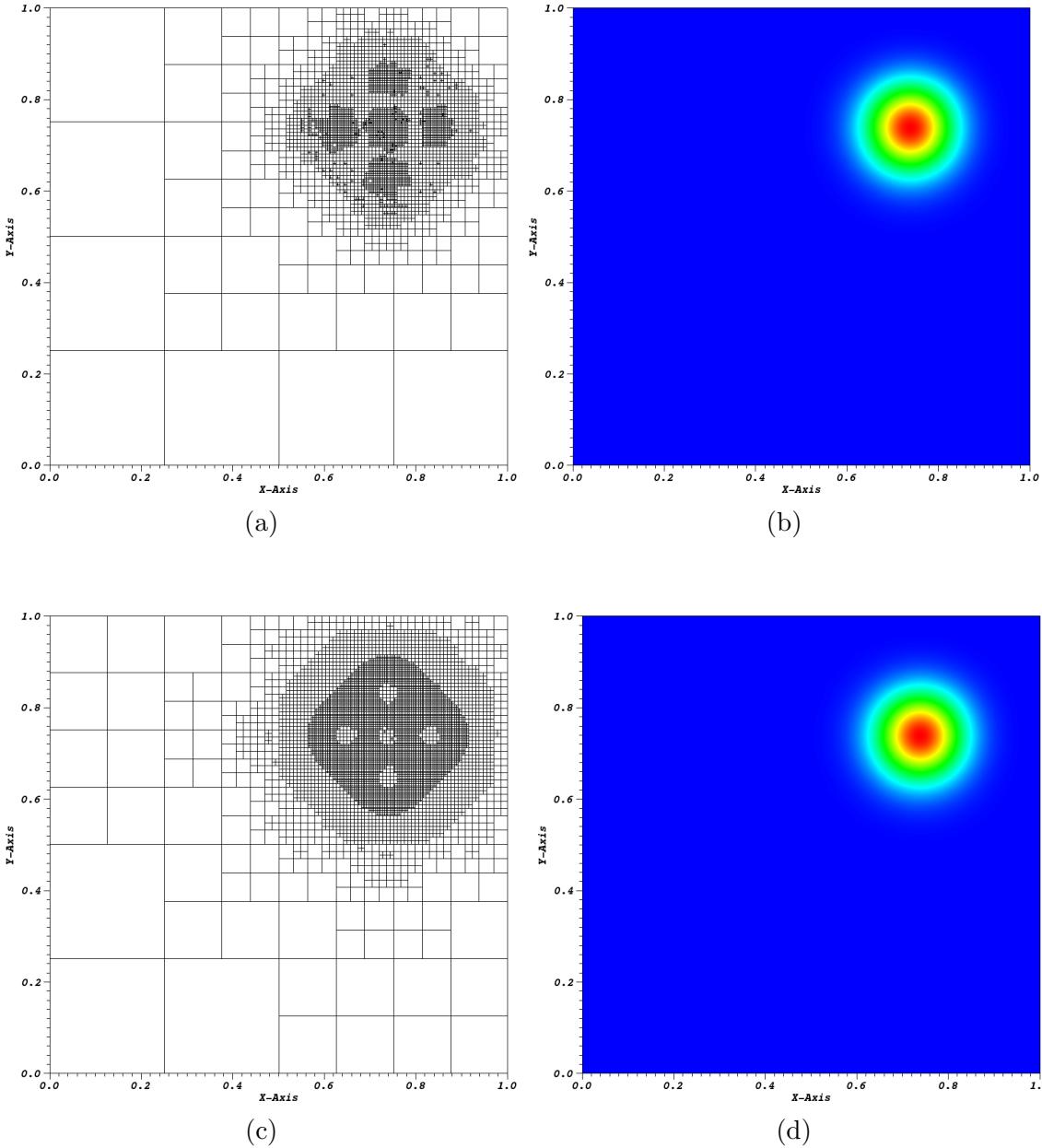


Figure 2.27: AMR meshes and solutions for the gaussian MMS problem using the maximum entropy coordinates: (top) linear basis functions at cycle 15 and (bottom) quadratic serendipity basis functions at cycle 08.

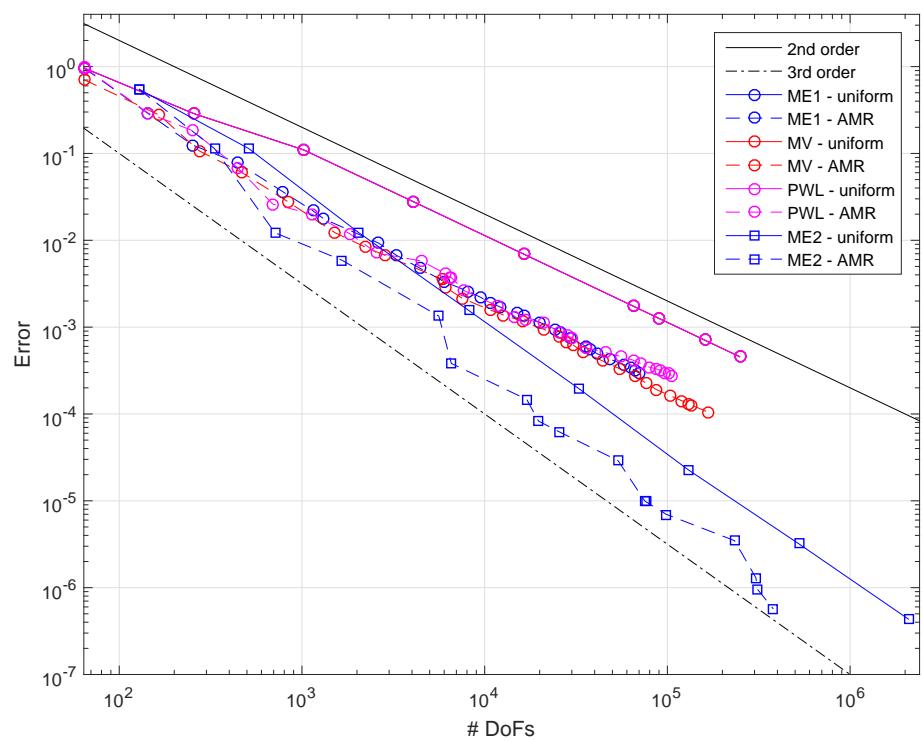


Figure 2.28:  $L_2$  error norm for the gaussian MMS problem using the various 2D polygonal basis functions

### 2.5.5 Searchlight Problem

The next example models a beam or searchlight. Similar problems were investigated in Dedner and Vollm  ller [19] and Wang and Ragusa [20]. 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 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 2.1 and 2.2 as well as

## 2.6 Conclusions

In this chapter, we have presented

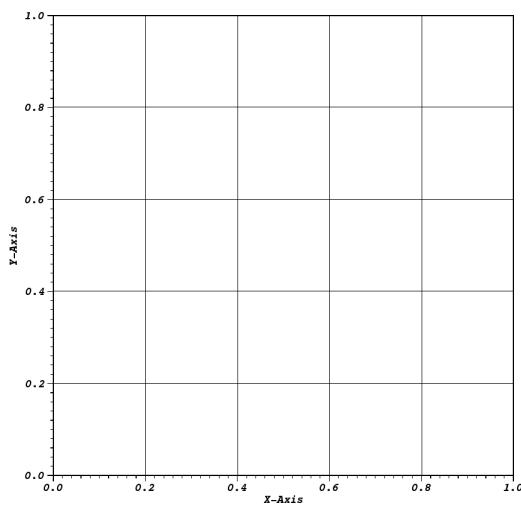


Figure 2.29: Initial mesh configuration for the searchlight problem before any refinement cycles.

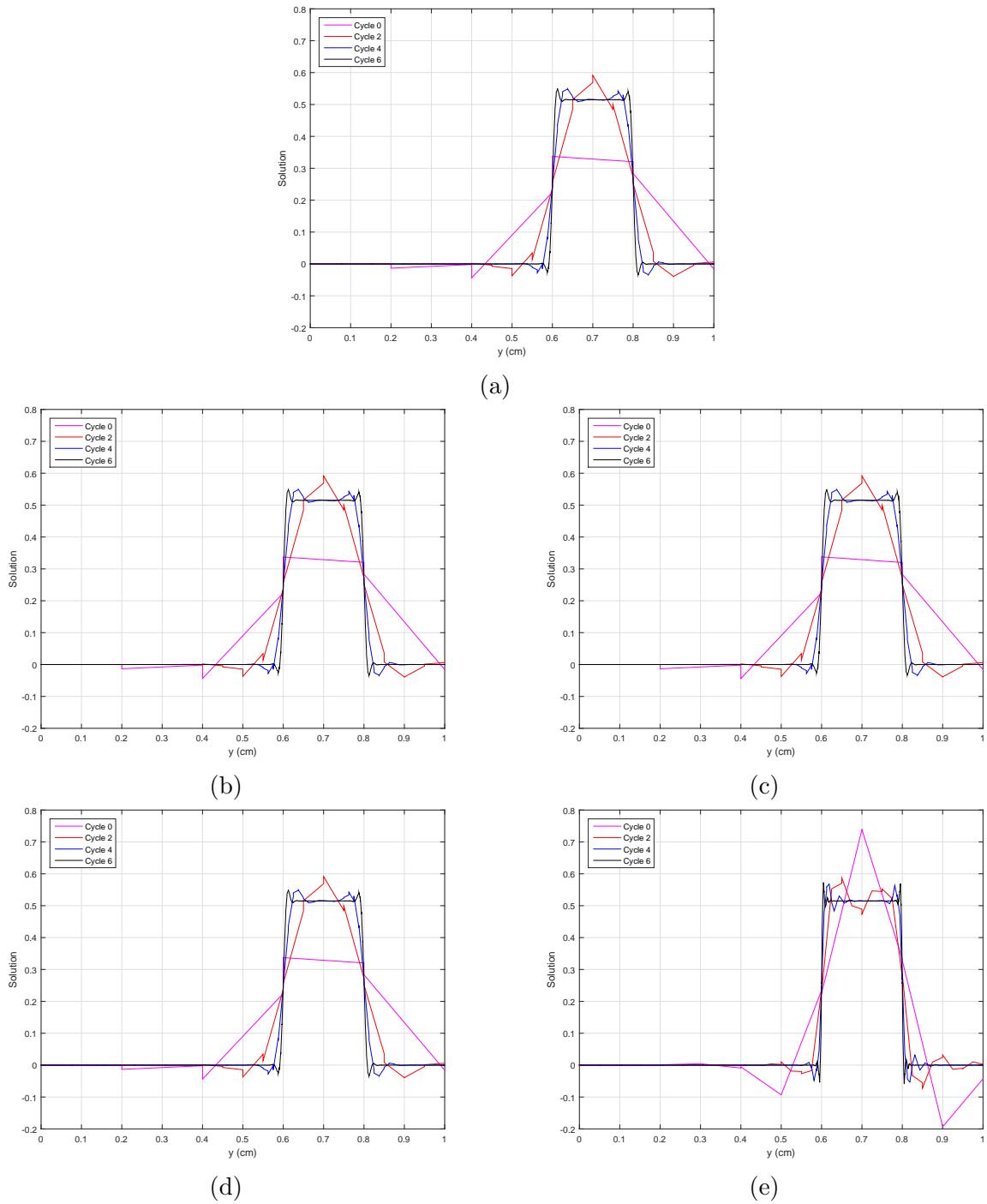


Figure 2.30: Exiting angular flux on the right boundary with uniform refinement using the (a) Wachspress basis functions, (b) PWL basis functions, (c) mean value basis functions, (d) linear maximum entropy coordinates and (e) quadratic serendipity maximum entropy coordinates.

## REFERENCES

- [1] STAR-CCM+, “<http://www.cd-adapco.com>,” (2015).
- [2] A. ERN and J.-L. GUERMOND, *Theory and practice of finite elements*, vol. 159, Springer Science & Business Media (2013).
- [3] J. R. SHEWCHUK, “Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator,” in “Applied computational geometry towards geometric engineering,” Springer, pp. 203–222 (1996).
- [4] J. R. SHEWCHUK, “Delaunay refinement algorithms for triangular mesh generation,” *Computational geometry*, **22**, 1, 21–74 (2002).
- [5] H. SI, “TetGen, a Delaunay-based quality tetrahedral mesh generator,” *ACM Transactions on Mathematical Software (TOMS)*, **41**, 2, 11 (2015).
- [6] C. GEUZAIN and J.-F. REMACLE, “Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities,” *International Journal for Numerical Methods in Engineering*, **79**, 11, 1309–1331 (2009).
- [7] M. S. FLOATER, “Mean value coordinates,” *Computer aided geometric design*, **20**, 1, 19–27 (2003).
- [8] K. HORMANN and M. S. FLOATER, “Mean value coordinates for arbitrary planar polygons,” *ACM Transactions on Graphics (TOG)*, **25**, 4, 1424–1441 (2006).
- [9] N. SUKUMAR, “Construction of polygonal interpolants: a maximum entropy approach,” *International Journal for Numerical Methods in Engineering*, **61**, 12, 2159–2181 (2004).

- [10] 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).
- [11] K. HORMANN and N. SUKUMAR, “Maximum entropy coordinates for arbitrary polytopes,” in “Computer Graphics Forum,” Wiley Online Library (2008), vol. 27, pp. 1513–1520.
- [12] S. KULLBACK and R. A. LEIBLER, “On information and sufficiency,” *The annals of mathematical statistics*, pp. 79–86 (1951).
- [13] E. T. JAYNES, “Information theory and statistical mechanics,” in “Statistical Physics,” (1963), vol. 3, pp. 181–218.
- [14] A. ROSOLEN, D. MILLÁN, and M. ARROYO, “Second-order convex maximum entropy approximants with applications to high-order PDE,” *International Journal for Numerical Methods in Engineering*, **94**, 2, 150–182 (2013).
- [15] 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).
- [16] 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).
- [17] A. SHESTAKOV, D. KERSHAW, and G. ZIMMERMAN, “Test problems in radiative transfer calculations,” *Nuclear science and engineering*, **105**, 1, 88–104 (1990).

- [18] D. S. KERSHAW, “Differencing of the diffusion equation in Lagrangian hydrodynamic codes,” *Journal of Computational Physics*, **39**, 2, 375–395 (1981).
- [19] 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).
- [20] 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).