A Study of Cell–Center Finite Volume Methods for Diffusion Equations

Weizhang Huang * and Andrew M. Kappen †

Abstract

We discuss four cell–center finite volume schemes for differencing diffusion equations. Numerical results are presented for comparing the performance of these schemes on a number of meshes with varying degrees of skewness. It is shown that both the new continuous edge flux scheme and the existing vertex flux scheme are very stable and generate second order or nearly second order accurate results on moderately and highly skewed meshes. The other two existing schemes suffer from accuracy and stability problems on these types of meshes.

AMS(MOS) 1991 Subject Classification. 65M99, 35K57
Key Words. finite volume methods, diffusion equation
Abbreviated title. Cell–center Finite Volume Discretizations

1 Introduction

Although the ideas underpinning finite volume (FV) methods have been well understood since their introduction by Varga in 1962 [16], it took the research of Jameson and Caughey [5] and Ni [13] to popularize their application in solving steady state Euler equations. Since that time, FV methods have developed into efficient, robust codes used widely in the aerospace industry for solving fluid dynamics problems. The common method behind all FV formulations is integrating a differential equation over a set of cells partitioning the physical domain Ω. These cells are typically quadrilateral or triangular when $\Omega \subset R^2$ and hexahedral or tetrahedral when $\Omega \subset R^3$. The integration is accomplished by using Gauss's theorem to transform the volume integral over the cell into a surface integral over the cell boundary before discretizing it.

FV methods have become popular for good reasons. They are flexible enough to be applied to complex real world domains, and because these methods work

*Department of Mathematics, the University of Kansas, Lawrence, KS 66045, U.S.A. (*huang@math.ukans.edu*)

†Department of Mathematics, the University of Kansas, Lawrence, KS 66045, U.S.A. (*kappen@math.ukans.edu*).

directly on the physical domain rather than on the computational domain through coordinate transformations, they can easily be used with adaptive mesh strategies. The ability to improve accuracy through adaptive meshes is proving essential to producing accurate, efficient computational fluid dynamics codes [8].

There are three basic classes of FV methods, distinguished by where unknown variables are stored in relation to the cell. These are cell–center, cell–vertex, and cell–edge methods. Cell–center methods are by far the most popular due to their ease of implementation and the close association of the cell residual to the center unknowns [11]. Morton et al. [10, 11, 12] have advocated cell–vertex methods in which the unknowns are held at the mesh nodes. The biggest advantages of their method are due to the small size of the difference stencil involved when a residual equation is defined. They show that this makes cell–vertex schemes less sensitive to mesh distortion than cell–center schemes, but admit that the edge averages used in their cell–center scheme are too simplistic to react to mesh distortions [11]. Cell–edge schemes are relatively uncommon in the literature.

We consider in this paper the application of cell–center methods to the numerical solution of diffusion equations. This issue has been addressed before by several researchers, for examples see [2, 6, 7, 9], and a variety of methods can be found in the literature for differencing diffusion related terms. Most researchers have presented results indicating some degree of success for their application examples. However, we should note that many of these methods have been used only on orthogonal or nearly orthogonal meshes and have never been studied analytically or numerically on meshes with higher degrees of skewness. Furthermore, the differences among some of these methods are very subtle and little work has been done to compare their performance [6, 9]. This lack of comparative research may be the reason that none of these methods have seen clear dominance over the others in application. To add to the confusion, different methods are advocated by recently published books [2, 10] and review [8] for use in differencing diffusion terms.

The objective of this paper is twofold: first a new scheme is derived and then its performance is examined along with three other existing schemes. The new scheme, called the continuous edge flux scheme (CEFS), is derived using an edge flux continuity condition, motivated by the work of Morel et al. [9]. Unlike the method of [9], however, this new scheme does not need to use edge unknowns together with center unknowns. Despite this difference, it will be shown that the CEFS has comparable performance with the scheme presented in [9]. In addition, we show that for cases with continuous materials, the CEFS is closely related to the Taylor series expansion scheme (see below). The three existing schemes selected for investigation are the vertex flux scheme (VFS) presented by Frink [3] and having much in common conceptually with the center–scalar nodal–flux (CN) mimetic finite difference scheme developed by Hyman et al. [1] and Shashkov [14], the center flux scheme (CFS) discussed extensively in [2] (and references therein), and the Taylor series expansion scheme (TSES), a classic scheme investigated in [6] and also discussed in [10]. These difference–based methods are selected because they can easily be implemented on general meshes with different types of cells and can be extended easily to three dimensions. They are also second order approximations on orthogonal meshes and

reduce to five point finite difference schemes for the Laplace operator. In general, difference–based methods are favored by many researchers because they are much simpler than finite element–based methods and more flexible than methods based on coordinate transformations (e.g. see [2, 10]). The comparative study is conducted with a model convection problem on a carefully chosen selection of meshes having varying degrees of skewness.

The remainder of this paper is organized as follows. Section 2 will describe the general formulation of the cell–center FV discretization for diffusion equations. The four schemes are described in detail in section 3. Their features and eigenvalue approximation properties for the Laplace operator on a uniform mesh are discussed in section 4. In section 5, a two dimensional model is given and used for the test example. Numerical results obtained with the four schemes are presented and analyzed for the model problem using four types of meshes. Finally, section 6 contains conclusions and remarks.

2 Cell–Center Finite Volume Method

In this section we briefly describe the cell–center finite volume method applied to two dimensional diffusion equations. To be specific, we consider an equation in the form

$$\begin{aligned} &\frac{\partial u}{\partial t} = \nabla \cdot G + f, \quad (x,y) \in \Omega \\ &G = D\nabla u, \end{aligned} \tag{1}$$

where Ω is a two dimensional simply connected domain, $D = D(x,y,t)$ is the diffusion coefficient matrix (symmetric and positive definite), $f = f(x,y,t)$ is the source term, and G is the diffusive flux. We assume that (1) is supplemented with appropriate initial and boundary conditions and the corresponding initial boundary value problem is well posed.

Let $\vec{r} = [x,y]^T$. Denote by $\vec{r}_X$ the position of a point labeled with X and by u_X the approximation to the unknown function $u(\vec{r},t)$ at point X, i.e. $u_X \approx u(\vec{r}_X,t)$. (Thus u_X is a function of time.) We also use the notation $\vec{r}_{X-Y} = \vec{r}_Y - \vec{r}_X$ for the vector from X to Y. Point labels in relation to cell center C are shown in Figure 1.

The basic idea behind the cell–center finite volume method is to define unknown variables at cell centers and to integrate equation (1) over each cell. Take the cell V (with center C) in Figure 1 as an example. After dividing by the cell area A_C, integration leads to

$$\frac{1}{A_C}\int_V \frac{\partial u}{\partial t} dV = \frac{1}{A_C}\int_V \nabla \cdot G dV + \frac{1}{A_C}\int_V f dV. \tag{2}$$

Approximating the left hand side term and the source term is standard,

$$\frac{1}{A_C}\int_V \frac{\partial u}{\partial t} dV \approx \frac{du_C}{dt} \quad \text{and} \quad \frac{1}{A_C}\int_V f dV \approx f(\vec{r}_C,t). \tag{3}$$

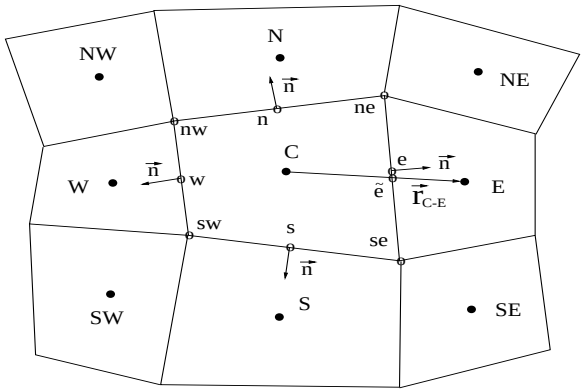

Figure 1: Typical 2D control volume (quadrilateral) and point labels

For the diffusion term, we use Gauss's theorem and approximate the resulting line integral with the midpoint rule. We have

$$\begin{aligned}\frac{1}{A_C}\int_V \nabla \cdot G dV &= \frac{1}{A_C}\oint_{\partial V} G \cdot \vec{n} dl \\ &\approx \frac{1}{A_C}\left(G_e \cdot S\vec{r}_{se-ne} + G_n \cdot S\vec{r}_{ne-nw} + G_w \cdot S\vec{r}_{nw-sw} + G_s \cdot S\vec{r}_{sw-se}\right), \qquad (4)\end{aligned}$$

where S is the skew matrix $S = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$. Substituting (3) and (4) into (2), we obtain

$$\frac{du_C}{dt} = \frac{1}{A_C}\left(G_e \cdot S\vec{r}_{se-ne} + G_n \cdot S\vec{r}_{ne-nw} + G_w \cdot S\vec{r}_{nw-sw} + G_s \cdot S\vec{r}_{sw-se}\right) + f(\vec{r}_C, t). \qquad (5)$$

Thus, the key to differencing diffusion equations is approximating edge center fluxes normal to cell edges.

3 Approximation of Diffusive Flux

Four schemes are described in this section for computing the diffusive flux at edge centers. They are the continuous edge flux scheme (CEFS), the vertex flux scheme (VFS), the center flux scheme (CFS), and the Taylor series expansion scheme (TSES). They are named here according to the manner with which the edge center fluxes are computed. For instance, in the VFS, the flux is first computed at vertices and its value at edge centers is obtained by averaging.

3.1 The Continuous Edge Flux Scheme (CEFS)

The first scheme is a new one derived using the edge flux continuity condition, motivated by the work of Morel et al. [9]. In [9], a scheme is developed for calculating diffusive fluxes based on the continuity of fluxes crossing cell edges. This scheme can be applied to problems involving material discontinuities, because as long as

the discontinuity occurs along cell edges the diffusion coefficient matrix may be calculated unambiguously at the cell center without affecting accuracy. This scheme has been shown to be second order and to work well on a selection of skewed meshes when combined with a fix–up procedure invoked for highly skewed cells. The main disadvantage of the method is that in addition to center variables, it employs two additional sets of unknown variables stored at edges. This increases the complexity and cost of implementing the scheme.

To avoid this disadvantage we propose a derivation called the CEFS scheme which uses the flux continuity condition of [9] in combination with approximation of the values of the unknown function at the vertices. To be specific, let us consider the computation of G_e. We begin with computing ∇u_e^- and ∇u_e^+, the two one–sided gradients to the edge $\vec{r}_{se-ne}$ at e. Approximating directional derivatives with finite differences, we have

$$\left\{ \begin{array}{rcl} \nabla u_e^- \cdot \vec{r}_{se-ne} & \approx & u_{ne} - u_{se} \\ \nabla u_e^- \cdot \vec{r}_{C-e} & \approx & u_e - u_C \end{array} \right. \tag{6}$$

and

$$\left\{ \begin{array}{rcl} \nabla u_e^+ \cdot \vec{r}_{se-ne} & \approx & u_{ne} - u_{se} \\ \nabla u_e^+ \cdot \vec{r}_{E-e} & \approx & u_e - u_E \end{array} \right. \tag{7}$$

The systems (6) and (7) can be solved for ∇u_e^- and ∇u_e^+, respectively, as functions of u_{ne}, u_{se}, u_e and unknown variables u_C and u_E. The formulas are

$$\begin{aligned} \nabla u_e^- &= \frac{1}{a}\left[(u_e - u_C) S\vec{r}_{se-ne} - (u_{ne} - u_{se}) S\vec{r}_{C-e}\right], \\ \nabla u_e^+ &= \frac{1}{b}\left[(u_e - u_E) S\vec{r}_{se-ne} - (u_{ne} - u_{se}) S\vec{r}_{E-e}\right], \end{aligned} \tag{8}$$

where $a = \vec{r}_{C-e} \cdot S\vec{r}_{se-ne}$ and $b = \vec{r}_{E-e} \cdot S\vec{r}_{se-ne}$.

Consider now the flux continuity condition. Denote by G_e^- and G_e^+ the C– and E–side diffusive fluxes at e, respectively. From the definition, these fluxes can be expressed by

$$G_e^- = D_e^- \nabla u_e^-, \qquad G_e^+ = D_e^+ \nabla u_e^+, \tag{9}$$

where D_e^- (or D_e^+) is the limit of the diffusion coefficient taken as the point approaches e from the C (or E) side. The diffusion flux continuity condition at the edge center e is then given by

$$G_e^- \cdot \vec{n} = G_e^+ \cdot \vec{n} \tag{10}$$

or

$$D_e^- \nabla u_e^- \cdot S\vec{r}_{se-ne} = D_e^+ \nabla u_e^+ \cdot S\vec{r}_{se-ne}. \tag{11}$$

Substituting (8) into the above condition, we have

$$\begin{aligned} & b\left[(u_e - u_C)\,(S\vec{r}_{se-ne}) \cdot \left(D_e^- S\vec{r}_{se-ne}\right) - (u_{ne} - u_{se})\,(S\vec{r}_{se-ne}) \cdot \left(D_e^- S\vec{r}_{C-e}\right)\right] = \\ & = a\left[(u_e - u_E)\,(S\vec{r}_{se-ne}) \cdot \left(D_e^+ S\vec{r}_{se-ne}\right) - (u_{ne} - u_{se})\,(S\vec{r}_{se-ne}) \cdot \left(D_e^+ S\vec{r}_{E-e}\right)\right]. \end{aligned} \tag{12}$$

This equation can be solved for u_e as a function of u_{ne}, u_{se}, u_C, and u_E. If values u_{ne} and u_{se} are known (in the sense that they can be expressed in terms of unknown variables), then u_e can be obtained and thus ∇u_e^- and the edge center flux $G_e \cdot S\vec{r}_{se-ne} \equiv G_e^- \cdot S\vec{r}_{se-ne}$ appearing in (5) can be calculated accordingly.

For the computation of u_{se} and u_{ne}, we could use a simple average. Instead, we use the slightly complicated but more accurate formula

$$u_{ne} = u_O + \nabla u_O \cdot \vec{r}_{ne-O}, \tag{13}$$

where O is the center of the cell with vertices C, E, NE, and N, $u_O = (u_C + u_E + u_{NE} + u_N)/4$, and ∇u_O is the computed gradient at O. The computation of ∇u_O is carried out using the common device of writing $(\partial u)/(\partial x) = \mathrm{div}(u, 0)$ and $(\partial u)/(\partial y) = \mathrm{div}(0, u)$, applying Gauss's theorem to appropriate integrals over the cell, and approximating the resulting line integrals with the trapezoidal rule. This gives

$$\begin{aligned} \nabla u_O \;=\; & \frac{1}{A}\left[\frac{1}{2}(u_E + u_{NE})S\vec{r}_{NE-E} + \frac{1}{2}(u_N + u_{NE})S\vec{r}_{N-NE}\right. \\ & \left. + \frac{1}{2}(u_C + u_N)S\vec{r}_{C-N} + \frac{1}{2}(u_E + u_C)S\vec{r}_{E-C}\right], \end{aligned} \tag{14}$$

where A is the cell area. The value of u_{se} can be calculated similarly.

Like many other FV schemes, Neumann boundary conditions can be treated trivially. For Dirichlet boundary conditions, system (6) can, for instance, be used for solving ∇u_e^- when points ne, e, and se are on the boundary and C is inside the domain. The edge flux is then calculated by $G_e = D_e^- \nabla u_e^-$.

As in the method of Morel et al. [9], the use of the flux continuity condition guarantees that a net rate of energy flow across each cell edge is uniquely defined. This is essential to ensure the global energy conservation. Furthermore, if a material discontinuity exits at a cell edge, the flux continuity gives the exact interface condition. In this sense, the condition provides a physical cell interaction. On the other hand, the computation of u_{ne} and u_{se} in the CEFS is certainly not physical and this may introduce additional errors for cases with material discontinuities. To see the significance of this effect, we note that the contributions of u_{ne} and u_{se} to $G_e^- \cdot S\vec{r}_{se-ne}$ in (5) are proportional to $(S\vec{r}_{se-ne}) \cdot (D_e^- S\vec{r}_{C-e})$ and $(S\vec{r}_{se-ne}) \cdot (D_e^+ S\vec{r}_{C-e})$, see (8). If D is a scalar multiple of the identity matrix and both $\vec{r}_{C-e}$ and $\vec{r}_{E-e}$ are perpendicular to $\vec{r}_{se-ne}$, then u_{ne} and u_{se} have no effect on the discretization. From this observation, we can expect that the calculations of u_{ne} and u_{se} will not affect significantly the robustness of the scheme on mildly to moderately skewed meshes.

The CEFS has a nine point stencil for equation (1) when a general quadrilateral mesh is used. On an orthogonal mesh, it reduces to the standard five point finite difference scheme for the Laplace operator. Thus, the CEFS can be regarded as a generalization of the standard five point finite difference scheme on a non–uniform mesh.

3.2 The Vertex Flux Scheme (VFS)

The second scheme to be described is the vertex flux scheme or VFS. It has been used by Frink [3] and recommended by Mavriplis [8]. It is also interesting to note that the VFS has much in common conceptually with the second order mimetic CN finite difference scheme developed by Hyman et al. [1] and Shashkov [14]. The main idea of this method is to first compute the gradient of u at the vertices and then to calculate the diffusive flux at the edge centers by averaging.

Specifically, the edge center flux G_e is computed as follows. First, the gradient of u at vertex ne (and vertex se) is computed. This is accomplished by taking the approximation $\nabla u_{ne} \approx \nabla u_O$ and computing ∇u_O using (14). Then, the edge center flux Ge is computed by averaging,

$$G_e = D_e \frac{1}{2} \left(\nabla u_{ne} + \nabla u_{se} \right), \tag{15}$$

where $D_e = D(\vec{r}_e, t)$.

There exist two ways to treat Dirichlet boundary conditions. The first one is to use formula (14) to compute ∇u_{ne} at the boundary using the cell with vertices N, C, e, and nne. The second one is to use the same boundary treatment employed in the CEFS to directly compute the diffusive flux at the edge center e. Surprisingly, our experience shows that the second approach appears to be inconsistent with the VFS and produces inferior results on non–orthogonal meshes. Thus, our results reflect the use of the first boundary treatment.

On a general quadrilateral mesh, the VFS has a nine point stencil with the calculation for the PDE at C dependent on the unknown variable u at C and its eight neighbors. Interestingly, for the Laplace operator on a rectangular mesh, the VFS reduces to a skewed five point finite difference stencil with dependency on the variables at C and its four corner neighbors SE, NE, NW, and SW. Such a skewed five point finite difference scheme has rarely been used in practice due to its inferior accuracy in comparison with the standard five point finite difference scheme. In fact, Kershaw [7] uses reduction to the standard five point finite difference as a criterion for accepting or dismissing new schemes. Although the VFS fails to meet this criterion, our numerical results show that it is still a highly effective scheme.

3.3 The Center Flux Scheme (CFS)

The third scheme is called the center flux scheme or CFS. It has been studied by several researchers and is discussed extensively by Ferziger and Perić in their book [2] (also see references therein). The idea behind this approach is to compute the flux first at cell centers and interpolate it to edge centers.

The first formula needed is the least squares approximation (interpolation) for u_e based on u_C and u_E. Expanding $u(\vec{r}_C)$ and $u(\vec{r}_E)$ into Taylor series about $\vec{r} = \vec{r}_e$ and minimizing the sum of the coefficients of the first order derivatives in the expansions lead to

$$u_e = \frac{\vec{r}^T_{C-E} \vec{r}_{e-E}}{|\vec{r}_{E-C}|^2} u_C + \frac{\vec{r}^T_{E-C} \vec{r}_{e-C}}{|\vec{r}_{E-C}|^2} u_E. \tag{16}$$

On a uniform rectangular mesh, this approximation is reduced to a simple two point average. To compute edge center flux G_e, ∇u_C and ∇u_E are first computed using a procedure similar to that used for equation (14). The difference is that we use here the midpoint rule to approximate the related line integrals. The resulting formula for ∇u_C is

$$\nabla u_C = \frac{1}{A_C}\left(u_e S\vec{r}_{ne-se} + u_n S\vec{r}_{nw-ne} + u_w S\vec{r}_{sw-nw} + u_s S\vec{r}_{se-sw}\right), \tag{17}$$

where u_e is obtained through (16) and the other edge center values are computed similarly. After ∇u_C and ∇u_E have been obtained, we can compute G_e once again using (16),

$$G_e = D_e\left[\frac{\vec{r}^T_{C-E}\vec{r}_{e-E}}{|\vec{r}_{E-C}|^2}\nabla u_C + \frac{\vec{r}^T_{E-C}\vec{r}_{e-C}}{|\vec{r}_{E-C}|^2}\nabla u_E\right]. \tag{18}$$

Neumann and Dirichlet boundary conditions for this scheme can be treated in the same way as the CEFS.

The stencil for the CFS generally involves thirteen points. The dependency is on the unknowns at C, its eight neighbors, and four far neighbors EE, NN, WW, and SS.

3.4 The Taylor Series Expansion Scheme (TSES)

The last scheme, referred to as the Taylor series expansion scheme or TSES, is the classic approximation and has been studied extensively by Jeng and Chen [6] and discussed in [10].

Unlike the VFS and CFS, the first step in this scheme is to approximate the function values at vertices ne and se by simple averaging,

$$\begin{aligned} u_{ne} &= \frac{1}{4}\left(u_C + u_E + u_{NE} + u_N\right), \\ u_{se} &= \frac{1}{4}\left(u_C + u_E + u_{SE} + u_S\right), \end{aligned} \tag{19}$$

and then to compute the gradient ∇u_e (and therefore G_e) using the Taylor series expansion. The obtained formula is similar to (14) but on a cell with vertices C, se, E, and ne.

Like the CEFS, this scheme also gives a nine point stencil in general and the standard five point finite difference scheme for the Laplacian operator on an orthogonal mesh.

Because of this similarity in stencil, it is interesting to see how the TSES is related to the CEFS. In fact, for the case where the diffusion coefficient D is continuous, the gradient at edge center e obtained with the CEFS is exactly the same as that obtained with the TSES, provided the same values for u_C, u_E, u_{se}, and u_{ne} are used. To show this, we first note that the method used in the CEFS uniquely defines ∇u_e^- and ∇u_e^+ regardless of the continuity of D. However, when D is continuous, we can show that $\nabla u_e^- = \nabla u_e^+ \equiv \nabla u_e$, where ∇u_e is the unique solution of

$$\begin{cases} \nabla u_e \cdot \vec{r}_{se-ne} &\approx u_{ne} - u_{se} \\ \nabla u_e \cdot \vec{r}_{C-E} &\approx u_E - u_C, \end{cases} \tag{20}$$

the gradients generated uniquely by the CEFS. Moreover, it is not difficult to verify that the solution of (20) is the same as the gradient produced by the TSES. Hence, we have proved our assertion.

From this, we can conclude that when D is continuous, the only difference between the CEFS and TSES is the way in which u_{ne} and u_{se} are computed.

4 Application to the Laplacian Operator

Before looking at the full results of our numerical experiments using these four methods, we briefly examine how well each one performs in approximating the eigenvalues of the Laplacian operator on a uniform mesh. For simplicity, we assume that the Laplacian operator is defined on the unit square with homogeneous Dirichlet boundary conditions imposed.

Analytically, the Laplacian operator with these conditions applied has a set of eigenvalues given by $\lambda(k_x, k_y) = (k_x^2 + k_y^2)\pi^2$ for $k_x = 1, 2, 3, \ldots$ and $k_y = 1, 2, 3, \ldots$, thus the smallest eigenvalue is $\lambda = 2\pi^2$. The eigenvalues of the approximation matrix for each method are computed using Matlab, and the errors in the smallest eigenvalue for each case are presented in Table 1. Note that in the current case both the CEFS and the TSES reduce to the standard five point finite difference scheme, so only the CEFS results are presented.

Table 1: Error in calculating the smallest eigenvalue of the Laplacian

$I \times J$	CEFS	CFS	VFS
4×4	0.9940	3.7392	7.8693
8×8	0.2524	0.9941	4.5694
12×12	0.1125	0.4469	3.1720
16×16	0.0634	0.2524	2.4178
24×24	0.0282	0.1125	1.6318

It is easy to see that both the CEFS and CFS have second order convergence rates while the convergence of the VFS is only first order. The CEFS and the TSES have a clear advantage in these results with errors far smaller than the other schemes. The lower order convergence of the VFS may be caused by its skewed nature and/or its inconsistency in stencils around the boundary. In fact, from Figure 2 we can see that the VFS has a skewed five point stencil at the interior points but at the boundary cells its stencils have different dependencies. For instance, along the east boundary, the stencil at C involves C and its four neighbors NW, SW, N, and S.

Figure 2 also shows the matrix patterns for the CEFS and the CFS. The CEFS can easily be seen to be most compact. We should note, however, that both the CEFS and the VFS have nine point stencils on a general mesh, and as will be seen in the next section, a poor approximation to the eigenvalues of the Laplacian operator

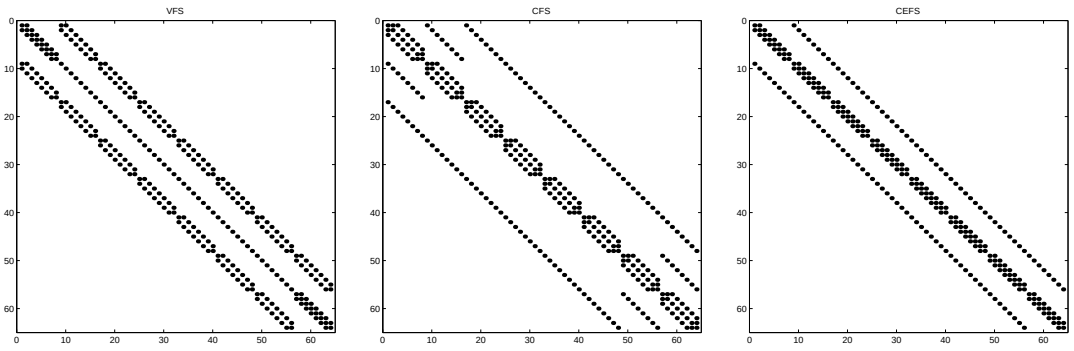

Figure 2: Matrix pattern for VFS, CFS, and CEFS approximations to Laplace's operator on a uniform orthogonal mesh.

on a uniform mesh does not necessarily lead to a poor approximation to the solution of the diffusion equation.

5 Numerical Experiments

We present in this section the numerical results obtained with the four schemes for a model problem. This problem has been used by Shashkov and his coworkers for testing their methods, e.g. see [14]. It is given by

$$\begin{cases} u_t = \nabla \cdot (D \nabla u) + f, & (x,y) \in \Omega \\ u = g, & (x,y) \in \partial\Omega \end{cases} \tag{21}$$

where Ω is a bounded domain to be defined later and the matrix D is a rotation of the diagonal matrix

$$K = \begin{bmatrix} 1 + 2x^2 + y^2 + y^5 & 0 \\ 0 & 1 + x^2 + 2y^2 + x^3 \end{bmatrix} \tag{22}$$

by some angle θ, or

$$D = RKR^T, \qquad R = \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix}. \tag{23}$$

The source function f, the boundary condition g, and the initial conditions are chosen such that the exact solution to the IVBP is $u = e^{-2\pi^2 t} sin(\pi x) sin(\pi y)$. All of the results presented are obtained using $\theta = \pi/4$ in calculating the matrix D.

For all computations, the system of ordinary differential equations resulting from the FV discretization of (21) is integrated in time from $t = 0$ to $t = 0.01$ using the θ–method ($\theta = 0.5$) with a fixed time–step $\Delta t = 0.001$. With this small time step size, the time discretization error in our results can be ignored compared to the error from spatial discretization. The resulting system of algebraic equations from this implicit integration is solved at each time step using the BiCGstab2 [4, 15] iterative method (without special treatments such as scaling or preconditioning) until the mean–square root of the residual is less than 10^{-8}. Four different types of meshes are used. The meshes and numerical results are described in the following sections.

5.1 Uniform Mesh

Initial experiments were run using a uniform mesh for the physical domain $\Omega = [0,1] \times [0,1]$. The number of cell center unknowns in the x and y directions are I and J respectively, and therefore the total number of mesh points is $(I+1)\times(J+1)$.

Our numerical results are summarized in Table 2. In the table, the columns represent errors and convergence ratios for the maximum and L_2 norms as well as the average number of BiCGstab2 iterations required per time–step.

Although all of the methods tested show a clear second order convergence on the uniform mesh, the actual error magnitudes differ greatly. As expected, the CEFS provides the best approximation to the exact solution. In fact, the CEFS errors are nearly one fifth as large as those of the VFS or CFS. Surprisingly, although the VFS is the poorest approximator of eigenvalues (having only first order convergence) for the Laplacian operator on the uniform mesh, it produces second order accurate results which are comparable to those obtained with the CFS. The TSES produces the same results as the CEFS on this mesh, so the TSES results are omitted from the table.

From the table, we can also see that surprisingly that the CFS requires fewer average iterations per time–step to solve the linear system resulting from the implicit integration. The VFS is the second most efficient method, while the CEFS is the most expensive.

Table 2: Results obtained using uniform meshes on $\Omega = [0,1] \times [0,1]$.

		maximum norm		L_2 norm		average iterations
Scheme	$I \times J$	error	ratio	error	ratio	per time-step
VFS	8 × 8	1.49e-02		6.98e-03		2.0
	16 × 16	3.85e-03	3.86	1.83e-03	3.81	3.0
	32 × 32	9.68e-04	3.97	4.69e-04	3.91	5.0
	64 × 64	2.40e-04	4.03	1.17e-04	4.00	11.0
CFS	8 × 8	1.52e-02		7.45e-03		2.0
	16 × 16	3.89e-03	3.90	1.92e-03	3.88	2.9
	32 × 32	9.81e-04	3.96	4.81e-04	3.98	4.1
	64 × 64	2.43e-04	4.04	1.19e-04	4.05	8.6
CEFS	8 × 8	3.36e-03		1.66e-03		2.0
	16 × 16	8.53e-04	3.94	4.09e-04	4.05	3.0
	32 × 32	2.11e-04	4.05	1.01e-04	4.07	7.0
	64 × 64	5.01e-05	4.20	2.35e-05	4.27	15.2

5.2 Sine Domain Mesh

The second type of meshes we use are applied to a sine domain defined as $\Omega = \{(x,y) \mid 0 \le x \le 1, 0 \le y \le Y(x)\}$ where $Y(x) = 1 + (sin(\sigma\pi x))/2$ and σ is a

parameter. This domain has also been used for numerical experiments in [14]. The mesh is defined as

$$x_{ij} = \frac{i}{I}, y_{ij} = \frac{jY(x_{ij})}{J}, \qquad i = 0, 1, \ldots, I, \quad j = 0, 1, \ldots, J. \tag{24}$$

This mesh is mildly skewed when σ is smaller (such as $\sigma = 2$) but can become very skewed for large σ. Two typical meshes for cases $\sigma = 2$ and $\sigma = 6$ are shown in Figure 3.

The results obtained for the case $\sigma = 2$ are listed in Table 3. Since this type of mesh is only mildly skewed, all four schemes show second order convergence. However, unlike on the uniform mesh, the VFS produces the most accurate results for this case.

To see how these schemes perform on more highly skewed meshes, we also ran experiments for the case $\sigma = 6$ where the mesh is non–smooth (for relatively small I and J) and highly skewed. The results are summarized in Table 4. First we note that the VFS still generates the best results when the mesh is refined enough ($I = J \geq 32$) and that the CEFS produces significantly better results than the CFS and the TSES. It is interesting to notice that the VFS loses its stability for coarser versions of this mesh ($I = J$ less than about 24), but all four methods recover second order convergence when the mesh is sufficiently refined.

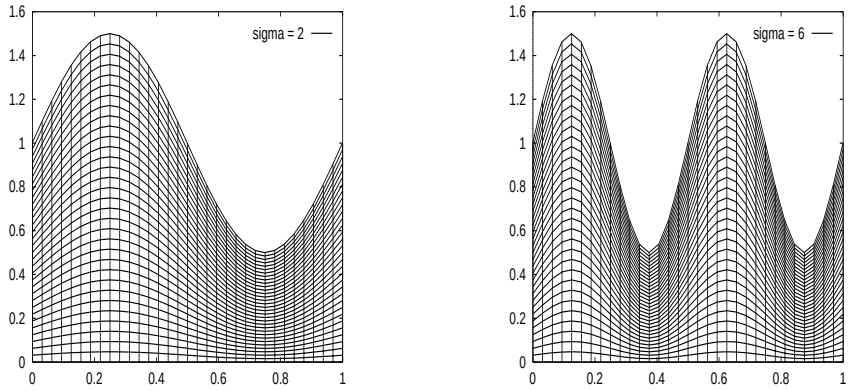

Figure 3: Typical meshes for the sine domain.

5.3 Random Mesh

Next, we present our results for the random mesh experiments. The random mesh over the physical domain $\Omega = [0, 1] \times [0, 1]$ is defined by

$$\begin{aligned} x_{ij} &= \frac{i}{I} + \frac{\sigma}{I}(R_x - 0.5), \\ y_{ij} &= \frac{j}{J} + \frac{\sigma}{J}(R_y - 0.5), \end{aligned}$$

where $\sigma \in [0, 1]$ is a parameter and R_x and R_y are two normalized random variables. When $\sigma = 0$ the mesh is uniform, and it can become highly skewed when σ is close

Table 3: Results obtained using non-orthogonal meshes ($\sigma = 2$) on $\Omega = \{(x, y) \mid 0 \leq x \leq 1, 0 \leq y \leq Y(x)\}$.

Scheme	$I \times J$	maximum norm		L_2 norm		average iterations
		error	ratio	error	ratio	per time-step
VFS	8 × 8	2.99e-02		1.01e-02		3.0
	16 × 16	8.96e-03	3.34	2.51e-03	4.01	8.3
	32 × 32	2.47e-03	3.63	6.19e-04	4.05	19.1
	64 × 64	5.92e-04	4.18	1.50e-04	4.12	43.0
CFS	8 × 8	1.66e-01		4.29e-02		4.0
	16 × 16	4.20e-02	3.95	1.05e-02	4.08	7.6
	32 × 32	9.28e-03	4.52	2.64e-03	3.99	17.2
	64 × 64	2.18e-03	4.25	6.59e-04	4.01	38.4
TSES	8 × 8	1.06e-01		2.90e-02		4.2
	16 × 16	4.18e-02	2.53	1.06e-02	2.73	9.5
	32 × 32	1.26e-02	3.31	3.17e-03	3.35	22.1
	64 × 64	3.40e-03	3.72	8.43e-04	3.76	51.8
CEFS	8 × 8	9.47e-02		2.71e-02		4.1
	16 × 16	2.99e-02	3.17	8.13e-03	3.34	10.0
	32 × 32	7.85e-03	3.81	2.15e-03	3.79	22.8
	64 × 64	2.09e-03	3.76	5.49e-04	3.91	50.1

to one. Figure 4 shows two typical random meshes generated with $\sigma = 0.6$ and $\sigma = 0.8$.

These meshes provide a tough test case. The randomly generated mesh is both skewed and hour–glassed. It is nowhere smooth on its interior and maintains a relatively constant degree of distortion as the number of mesh cells is increased [9]. The random mesh has been used by many researchers for testing their methods and in most cases, σ has been taken to be less than 0.4, e.g. see [1, 9, 14].

Figure 5 summarizes the results obtained with the four schemes on random meshes generated with $\sigma = 0.6$. We note that for a fixed pair of values for I and J, the degree of skewness for the obtained random mesh may vary with different runs. Thus, it is hard to demonstrate the general performance of a method based solely on the results obtained with one run per each pair of values of I and J. We perform 100 runs for each value $I = J$ and the results obtained are then plotted in the figure. From Figure 5, one can see clearly that the CEFS produces the best results for this case and exhibits second order convergence. The VFS generates the next best results, having nearly second order convergence. The CFS comes in third, having almost a first order convergence rate while the TSES is the worst, failing to converge as the number of cells are increased.

Figure 6 shows the results obtained for a much tougher case where the random mesh is generated with $\sigma = 0.8$. (We would like to remind the reader that the mesh

Table 4: Results obtained using non-orthogonal meshes ($\sigma = 6$) on $\Omega = \{(x, y) \mid 0 \leq x \leq 1, 0 \leq y \leq Y(x)\}$.

Scheme	$I \times J$	maximum norm		L_2 norm		average iterations
		error	ratio	error	ratio	per time-step
VFS	8 × 8	8.22e-02		3.03e-02		3.9
	16 × 16	3.92e+12	*	5.82e+11	*	27.7
	32 × 32	3.45e-02	*	8.47e-03	*	53.2
	64 × 64	7.75e-03	4.45	2.51e-03	3.37	130.7
	128 × 128	2.00e-03	3.88	6.60e-04	3.80	318.9
CFS	8 × 8	1.30e+00		3.50e-01		6.6
	16 × 16	2.26e+00	0.58	4.35e-01	0.80	21.1
	32 × 32	1.57e+00	1.44	2.49e-01	1.75	60.6
	64 × 64	9.27e-02	16.90	2.07e-02	12.01	120.0
	128 × 128	2.35e-02	3.94	3.76e-03	5.51	231.7
TSES	8 × 8	4.97e-01		2.02e-01		7.3
	16 × 16	3.26e-01	1.52	1.12e-01	1.80	24.7
	32 × 32	1.36e-01	2.41	4.49e-02	2.50	64.0
	64 × 64	4.67e-02	2.90	1.56e-02	2.87	140.5
	128 × 128	1.35e-02	3.46	4.54e-03	3.44	325.3
CEFS	8 × 8	1.67e-01		6.16e-02		6.5
	16 × 16	1.47e-01	1.14	4.89e-02	1.26	23.5
	32 × 32	7.52e-02	1.96	2.32e-02	2.11	59.5
	64 × 64	2.57e-02	2.92	7.88e-03	2.94	139.9
	128 × 128	7.27e-03	3.54	2.21e-03	3.57	326.9

lines can cross if σ is taken to be greater than or equal to one.) For this case, the convergence of the CEFS is still second order, while the convergence rate for the VFS drops to about 1.5 (compared to the $\sigma = 0.6$ case where the rate is nearly 2). Both the CEFS and the VFS produced stable results for all 100 test runs. The CFS has a nearly first order convergence rate, and the TSES is not convergent.

5.4 Z–Mesh

Finally, we examine the results obtained using a z–mesh as described in [7]. The z–mesh is shown in figure 7, and the numerical results appear in Table 5. On this mesh the VFS and the CEFS both produce excellent second order results. The TSES has only first order convergence, and the CFS is divergent.

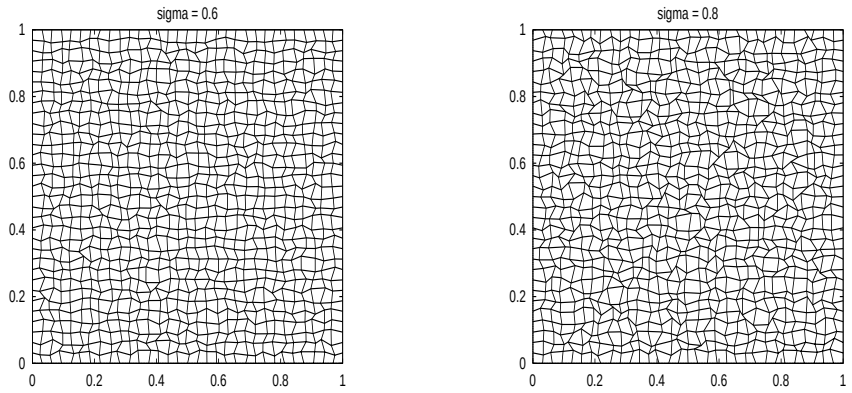

Figure 4: Typical random meshes

6 Conclusions and Remarks

In the previous sections we have described four cell–center finite volume schemes, the continuous edge flux scheme, the vertex flux scheme, the center flux scheme, and the Taylor series expansion scheme for solving diffusion equations. The CEFS is new and the others are existing schemes. Numerical results have been presented to compare the performance of these schemes on a variety of meshes with different degrees of skewness. Four types of meshes have been used, the uniform mesh, the gradually changing mesh (the sine domain mesh), the random mesh, and the z–mesh. It has been shown that both the CEFS and the VFS retain second order accuracy on mildly to highly skewed meshes. As the mesh becomes very highly skewed, the VFS loses some of its convergence rate while the CEFS maintains second order convergence. The CFS and TSES have second order accuracy only on mildly skewed meshes. The CFS loses significant accuracy and stability on moderately and highly skewed meshes, while the TSES fails to converge at all on these meshes.

It should be noted that both the CEFS and the VFS may lose their accuracy and stability on a extremely skewed mesh. For instance, we ran numerical experiments on random meshes with $\sigma = 0.9$ and found that the VFS loses second order accuracy but stays stable while the CEFS retains second order accuracy for most of the meshes but becomes unstable for others. To correct these stability problems, a fix-up procedure like one presented in [9] may be necessary.

Acknowledgment.

This work was supported in part by the NSF under Grant DMS-9626107 and the Kansas Center for Advanced Scientific Computing sponsored by the NSF-EPSCoR/KSTAR program. The computations were done on the machines of the Scientific Computing and Visualization Laboratory (supported by NSF Grant DMS-9628626) at the Department of Mathematics, the University of Kansas.

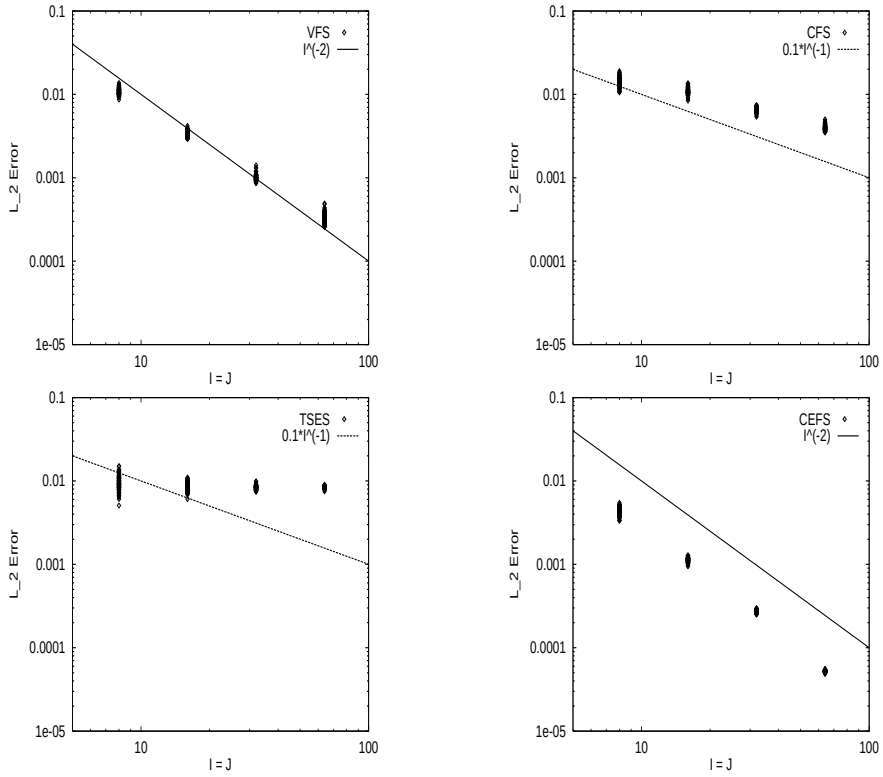

Figure 5: Random Case with $\sigma = 0.6$: L_2 error is plotted against the number of mesh points in one direction.

References

[1] J. E. Castillo, J. M. Hyman, M. J. Shashkov, and S. Steinberg. The sensitivity and accuracy of fourth order finite-difference schemes on nonuniform grids in one dimension. *Comput. Math. Appl.*, 30:41 – 55, 1995.

[2] J. H. Ferziger and M. Perić. *Computational methods for fluid dynamics.* Springer, Berlin, 1996.

[3] N. T. Frink. Recent progress toward a three dimensional unstructured navier-stokes flow solver. Technical Report 94-0061, AIAA Papers, 1994.

[4] M. H. Gutknecht. Variants of bicgstab for matrices with complex spectrum. *SIAM J. Sci. Comput.*, 14:1020 – 1033, 1993.

[5] A. Jameson and D. E. Caughey. A finite volume method for transonic potential flow calculations. In *Proc. AIAA Third Computational Fluid Dynamics Conference, Albuqueque*, pages 35 – 54, 1977.

[6] Y. N. Jeng and J. L. Chen. Geometric conservation law of the finite-volume method for the simpler algorithm and a proposed upwind scheme. *Numer. Heat Transf.*, 22:211 – 234, 1992.

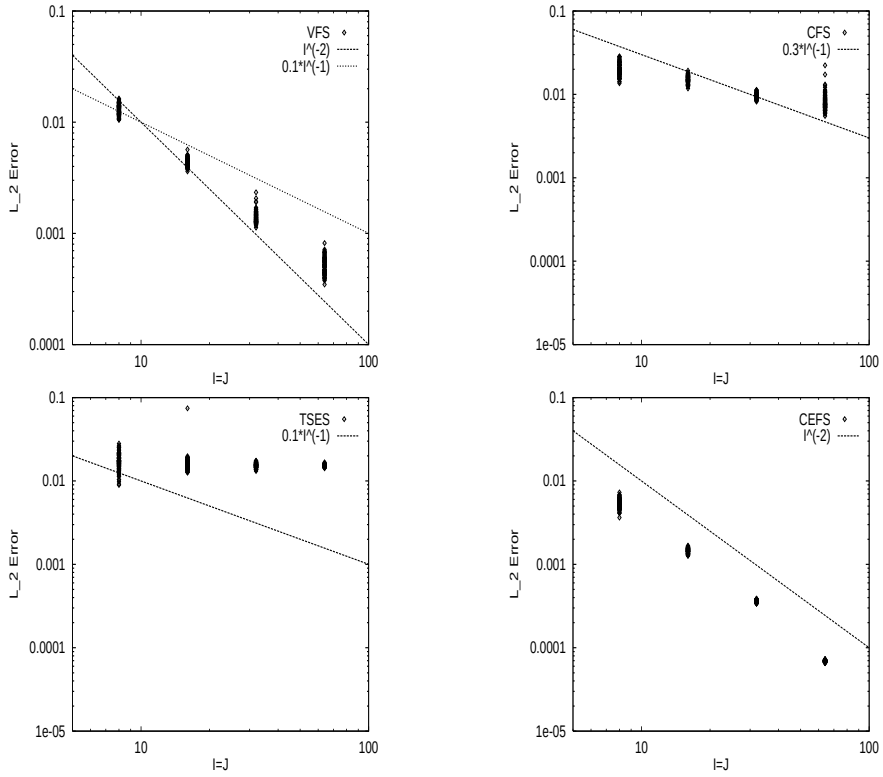

Figure 6: Random Case with $\sigma = 0.8$: L_2 error is plotted against the number of mesh points in one direction.

[7] D. S. Kershaw. Differencing of the diffusion equation in lagrangian hydrodynamic codes. *J. Comput. Phys.*, 76:375 – 395, 1981.

[8] D. J. Mavriplis. Unstructured grid techniques. *Annu. Rev. Fluid. Mech.*, 29:473 – 514, 1997.

[9] J. E. Morel, J. E. Dendy Jr., M. L. Hall, and S. W. White. A cell-centered lagrangian-mesh diffusion differencing scheme. *J. Comput. Phys.*, 103:286 – 299, 1992.

[10] K. W. Morton. *Numerical Solution of Convection-Diffusion Problems.* Chapman & Hall, London, 1996.

[11] K. W. Morton and M. F. Paisley. A finite volume scheme with shock fitting for the steady euler equations. *J. Comput. Phys.*, 80:168 – 203, 1989.

[12] K. W. Morton and E. Süli. Finite volume methods and their analysis. *IMA J. Num. Anal.*, 11:241 – 260, 1991.

[13] Ron-Ho Ni. A multiple-grid scheme for solving the euler equations. *AIAA J.*, 20:1565 – 1571, 1982.

[14] M. Shashkov. *Conservative finite-difference methods on general grids.* CRC Press, Boca Raton, FL., 1996.

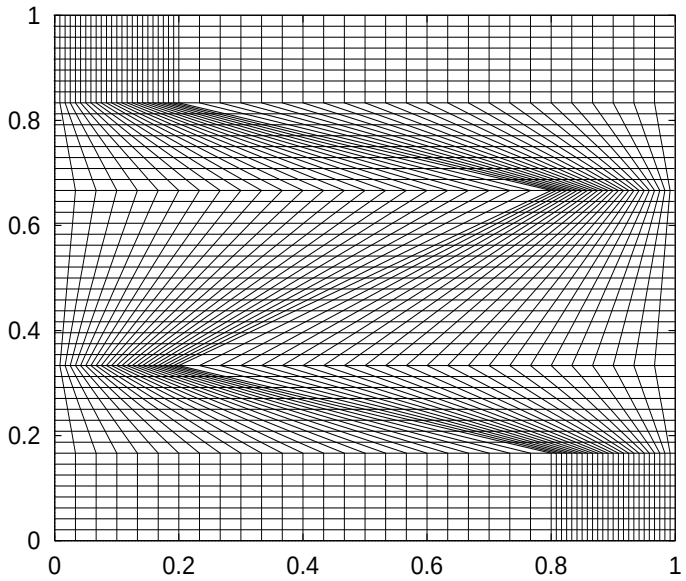

Figure 7: A typical Z-mesh

[15] H. A. van der Vorst. Bi-cgstab: a fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 13:631 – 644, 1992.

[16] R. S. Varga. *Matrix Iterative Anlysis*. Prentice-Hall, Englewood Cliffs, 1962.

Table 5: Results obtained using z–mesh on $\Omega = [0,1] \times [0,1]$.

Scheme	$I \times J$	maximum norm		L_2 norm		average iterations
		error	ratio	error	ratio	per time-step
VFS	12 × 12	4.37e-02		9.40e-03		4.1
	24 × 24	1.19e-02	3.37	2.77e-03	3.39	10.9
	48 × 48	3.59e-03	3.31	7.35e-04	3.77	27.8
	96 × 96	1.01e-03	3.55	1.88e-04	3.91	75.4
CFS	12 × 12	4.20e-01		9.41e-02		3.8
	24 × 24	7.85e-01	*	1.29e-01	*	9.1
	48 × 48	1.16e+05	*	6.33e+03	*	35.4
	96 × 96	1.83e+04	*	8.29e+02	*	124.9
TSES	12 × 12	1.08e-01		4.09e-02		6.2
	24 × 24	8.20e-02	1.32	2.69e-02	1.52	17.4
	48 × 48	5.19e-02	1.58	1.52e-02	1.77	42.8
	96 × 96	2.87e-02	1.81	7.91e-03	1.92	105.8
CEFS	12 × 12	5.32e-02		1.86e-02		5.8
	24 × 24	2.51e-02	2.12	8.53e-03	2.18	14.3
	48 × 48	9.25e-03	2.71	2.93e-03	2.91	37.7
	96 × 96	2.67e-03	3.46	8.26e-04	3.55	92.3