Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

Journal of Computational Physics 199 (2004) 589–597

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

Mimetic finite difference methods for diffusion equations on non-orthogonal non-conformal meshes ☆

Konstantin Lipnikov [a,*], Jim Morel [b], Mikhail Shashkov [a]

[a] *Theoretical Division, T-7, Los Alamos National Laboratory, Mail Stop B284, Los Alamos, NM 87545, USA*
[b] *Computer and Computational Sciences Division, CCS-2, Los Alamos National Laboratory, MS D409, Los Alamos, NM 87545, USA*

Received 4 March 2003; received in revised form 18 February 2004; accepted 28 February 2004
Available online 2 April 2004

Abstract

Mimetic discretizations based on the support-operators methodology are derived for non-orthogonal locally refined quadrilateral meshes. The second-order convergence rate on non-smooth meshes is verified with numerical examples.
© 2004 Elsevier Inc. All rights reserved.

PACS: 65N06; 65N22; 80A20

Keywords: Mimetic finite differences; Quadrilateral meshes; Diffusion equation

1. Introduction

The support-operators (SO) method has been used to generate both 2-D and 3-D cell-centered diffusion discretizations on non-orthogonal meshes [6,7]. The SO method can be applied to many forms of the diffusion equation, but it suffices to consider the following first-order form of the diffusion equation:

$$\begin{aligned} \operatorname{div} \mathbf{F} &= Q, \\ \mathbf{F} &= -\mathbf{D}\operatorname{grad}\phi \equiv G\phi, \end{aligned} \tag{1.1}$$

where ϕ denotes a scalar function that we refer to as the intensity, $\mathbf{F}$ denotes a vector function that we refer to as the flux, $\mathbf{D}$ denotes a symmetric positive definite diffusion tensor, G denotes a flux operator, and Q denotes a source function. We use the term "cell centered" to indicate a discretization that expresses the following conservation integral over each cell volume:

[☆] The work was performed at Los Alamos National Laboratory operated by the University of California for the US Department of Energy under contract W-7405-ENG-36.

[*] Corresponding author. Tel.: +1-505-667-1719; fax: +1-505-665-5757.
E-mail addresses: lipnikov@lanl.gov (K. Lipnikov), jim@lanl.gov (J. Morel), shashkov@lanl.gov (M. Shashkov).

0021-9991/$ - see front matter © 2004 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2004.02.016

$$\oint_{A_c} \mathbf{F} \cdot \mathbf{n} \mathrm{d}A = \int_{V_c} Q \mathrm{d}V, \tag{1.2}$$

where A_c is the cell area and V_c is the cell volume. It does not necessarily imply that there are only cell-center intensity unknowns. For instance, there are actually two versions of the SO method: one that has intensity unknowns located only at cell centers, and one that has them located at both cell centers and face centers [5]. These methods are equivalent in the sense that they yield identical cell-center intensity solutions, but the former method has a dense diffusion matrix, while the latter method has a sparse diffusion matrix. The latter method is referred to as the local SO method.

The purpose of this paper is to use the local SO method to generate diffusion discretizations in both *x–y* and *r–z* geometries on non-orthogonal quadrilateral meshes having hanging nodes. Such nodes occur when two cells share a face with one cell. Hanging-node meshes are generally used in adaptive mesh refinement (AMR) algorithms [3,8,9,12]. We do not allow any cell to share a face with more than two cells, or equivalently, there is never more than one hanging node associated with a mesh edge. This is a restriction motivated by considerations of both simplicity and accuracy that is usually seen in quadtree-based adaptive-mesh algorithms [12]. While the vast majority of existing AMR schemes use orthogonal meshes, it is clear that non-orthogonal mesh schemes are more versatile. Non-orthogonal AMR meshes have been used in adaptive arbitrary Lagrangian–Eulerian hydrodynamics algorithms [1]. Our scheme would be suitable for coupled radiation diffusion/hydrodynamics calculations on such meshes.

Our purpose here is not to develop such an AMR diffusion algorithm per se, but rather to develop an SO method suitable for use in such an algorithm. The development of any AMR algorithm requires a posteriori error estimation technique [11]. The development of an SO-compatible posteriori error estimation technique is beyond the scope of this paper. Thus, rather than refine to optimally decrease the error in our convergence studies, we simply start with a multi-level hanging-node mesh and carry out a refinement step by increasing the refinement level in each cell by one. This is later demonstrated in detail.

The treatment of hanging nodes with cell-centered diffusion discretizations is not as straightforward as it might seem, particularly if the diffusion coefficient is discontinuous across an interface. For instance, consider the cells e_i, e_j, and e_k shown in Fig. 1, and assume intensity unknowns, ϕ_i, ϕ_j, and ϕ_k, at each cell center. Even if the mesh is orthogonal and the diffusion coefficient is continuous, one cannot accurately compute the flux across the interface between cells *i* and *j* in terms of a difference between the intensities, ϕ_i and ϕ_j. The intensity, ϕ_k, must be included in the difference expression to obtain a convergent diffusion discretization. If the diffusion coefficient is discontinuous, the situation becomes significantly more complex because the gradient of the intensity is discontinuous as well. One of the first convergent diffusion discretization schemes for orthogonal hanging-node meshes was given by Ewing et al. [3]. They developed a symmetric cell-centered finite-difference scheme in *x–y* geometry for meshes having an even number of

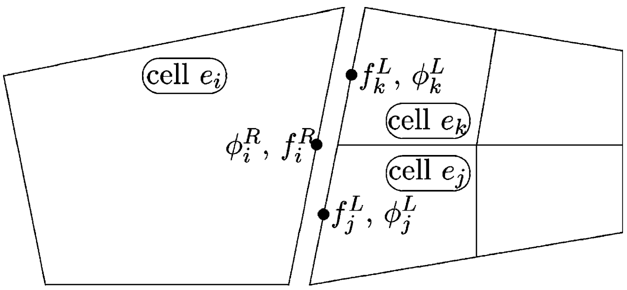

Fig. 1. Inner edge with a hanging node.

hanging nodes per interface. Although they defined their scheme to accommodate discontinuous diffusion coefficients, all of their convergence proofs assumed a continuous diffusion coefficient. Edwards [2] developed a symmetric cell-centered discretization scheme for the diffusion equation associated with the pressure in reservoir simulation. He considered orthogonal hanging-node meshes in *x–y* geometry having an odd number of hanging nodes per interface. To deal with discontinuous diffusion coefficients and the associated discontinuities in the intensity gradient, Edwards initially defined one-sided differences for the interface fluxes on each side of the interface that involved additional interface intensities. Equations for these additional interface intensities were obtained by requiring continuity of the flux across the interface. Because Edwards considered only orthogonal meshes, he was able to locally eliminate these additional interface intensities to obtain interface flux expressions containing only cell-center intensities.

The local SO method is particularly amenable to hanging-node meshes because the discretization process occurs in two steps. The first step is to consider each cell in the mesh as an independent domain and generate an independent discretization for each cell. The second step is to obtain a global discretization by imposing continuity of the intensity and continuity of the normal component of the flux across cell interfaces. Since each cell in a hanging-node mesh is a quadrilateral, the first step in applying the SO method is identical to that for standard quadrilateral meshes. However, imposing continuity of the intensity and continuity of the normal component of the flux is not straightforward on hanging-node meshes because the intensity and flux unknowns do not coincide as they do on standard meshes. Thus, the main task before us is to determine how to impose continuity of the intensity and the normal component of the flux at three-cell hanging-node interfaces. Our approach for imposing continuity at such interfaces is defined by two requirements. The first requirement is to globally satisfy a certain integral identity that is the fundamental basis of the SO method. The second requirement is to admit constant flux and linear intensity solutions.

Our hanging-node SO method can accommodate full symmetric positive definite tensor diffusion coefficients in addition to scalar and diagonal diffusion coefficients. The tensor diffusion capability is demonstrated in Section 3.

The remainder of this paper is organized as follows. In Section 2, we first briefly review the local SO method, and then show how it is generalized for meshes with three-cell hanging-node interfaces. Computational results are given in Section 3, followed by conclusions and recommendations for future work in Section 4.

2. Support-operators method

2.1. Overview of the SO methodology

The basic steps of the SO method are as follows. First, we introduce vector spaces of discrete functions together with inner products. Then, we derive a discrete approximation to the divergence operator which we call the *prime* operator. Finally, the approximation to the flux operator (the *derived* operator) follows naturally from a discrete analog of the Gauss–Green formula (see [5] for more details):

$$\int_{e_i} \mathbf{F}\cdot\mathbf{D}^{-1}(\mathbf{D}\,\mathrm{grad}\,\phi)\,\mathrm{d}V + \int_{e_i} \phi\,\mathrm{div}\,\mathbf{F}\,\mathrm{d}V = \int_{\partial e_i} \phi\mathbf{F}\cdot\mathbf{n}\,\mathrm{d}A. \tag{2.1}$$

We consider a shape regular partition of computational domain into convex quadrilateral cells. Generally speaking, the partition may be non-conforming. In this case we assume additionally that each mesh edge has at most one hanging node.

Each mesh cell e_i is assumed to be homogeneous, but material properties (diffusion tensor) may vary between cells. The intensities (scalars) are defined at both cell centers, ϕ_i^{C}, and the edge centers, ϕ_i^{R}, ϕ_i^{T}, ϕ_i^{L},

ϕ_i^{B}. For instance, ϕ_i^{R} is located at the center of the right edge (see Fig. 1). The vectors are defined in terms of edge-normal components located at the midpoints of cell edges, f_i^{R}, f_i^{T}, f_i^{L}, f_i^{B}. For instance, $f_i^{\mathrm{R}} \approx \mathbf{F} \cdot \mathbf{n}_i^{\mathrm{R}}$.

Let $\vec{f}_i = (f_i^{\mathrm{R}}, f_i^{\mathrm{T}}, f_i^{\mathrm{L}}, f_i^{\mathrm{B}})^{\mathrm{t}}$ and $\vec{\phi}_i = (\phi_i^{\mathrm{C}}, \phi_i^{\mathrm{R}}, \phi_i^{\mathrm{T}}, \phi_i^{\mathrm{L}}, \phi_i^{\mathrm{B}})^{\mathrm{t}}$ be vectors of flux and intensity unknowns, respectively, for mesh cell e_i. The SO method [5] results in the following discretization:

$$\begin{aligned} \mathcal{DIV}_i \vec{f}_i &= Q_i, \\ \vec{f}_i &= \mathcal{G}_i \vec{\phi}_i, \end{aligned} \tag{2.2}$$

where $\mathcal{DIV}_i$ and $\mathcal{G}_i$ denote the prime and derived operators, respectively, and Q_i denotes a value of the source function in the ith cell. Note that the discrete operators are adjoint to each other and satisfy a discrete analog of the Green formula [5]. Moreover, Eqs. (2.2) are exact for linear solutions. The system of cell-based equations is closed by imposing boundary conditions on boundary mesh edges and continuity conditions on interior edges.

2.2. Interface continuity conditions

In this section we describe continuity conditions for primary variables $\mathbf{F}$ and ϕ across cell interfaces. We require that the discrete version of the (2.1) holds over the entire mesh. It is not difficult to see that this requirement will be met if the sum over all cells of discrete Gauss–Green formulas leaves only outer boundary contributions due to cancellation of the interior mesh contributions. For instance, let cell e_i has a common boundary with cells e_j and e_k as shown in Fig. 1. Then, the boundary terms associated with the common interface will cancel if

$$A_i^{\mathrm{R}} f_i^{\mathrm{R}} \phi_i^{\mathrm{R}} = -A_j^{\mathrm{L}} f_j^{\mathrm{L}} \phi_j^{\mathrm{L}} - A_k^{\mathrm{L}} f_k^{\mathrm{L}} \phi_k^{\mathrm{L}}, \tag{2.3}$$

where A_i^{R}, A_j^{L} and A_k^{L} denote length of corresponding edges. The negative sign occurs because the edge normals associated with f_i^{R} and f_j^{L} (resp., f_i^{R} and f_k^{L}) are opposite in direction. To match the number of equations, we have to impose three continuity conditions such that (2.3) is satisfied and the resulting scheme is locally conservative. In the local SO method, the discrete divergence and flux operators are adjoint to each other. Then, condition (2.3) guarantees that the discrete divergence and flux operators over any mesh subset will be adjoint to each other, which in turn implies that matrix S in (3.1) is symmetric and positive definite.

The local conservation property requires that the approximate fluxes satisfy

$$A_i^{\mathrm{R}} f_i^{\mathrm{R}} = -A_j^{\mathrm{L}} f_j^{\mathrm{L}} - A_k^{\mathrm{L}} f_k^{\mathrm{L}}. \tag{2.4}$$

Requiring the scheme to be exact for a uniform flow, i.e., for a constant flux vector, results in the following continuity condition for intensities:

$$A_i^{\mathrm{R}} \phi_i^{\mathrm{R}} = A_j^{\mathrm{L}} \phi_j^{\mathrm{L}} + A_k^{\mathrm{L}} \phi_k^{\mathrm{L}}. \tag{2.5}$$

We may consider either (2.5) or (2.4) as the first interface condition. Let us begin with (2.5). Substituting it into (2.3), we get

$$A_i^{\mathrm{L}} \phi_i^{\mathrm{L}} (f_i^{\mathrm{R}} + f_j^{\mathrm{L}}) + A_k^{\mathrm{L}} \phi_k^{\mathrm{L}} (f_i^{\mathrm{R}} + f_k^{\mathrm{L}}) = 0.$$

It is easy to check that sufficient conditions satisfying both (2.4) and the last equation are

$$f_i^{\mathrm{R}} = -f_j^{\mathrm{L}} = -f_k^{\mathrm{L}}. \tag{2.6}$$

Thus, we end up with a weak interface condition (2.5) for intensities and a strong interface condition (2.6) for fluxes.

Although it may seem appealing to choose (2.4) as the first interface condition, doing so results in the following condition for intensities:

$$\phi_i^{\mathrm{R}} = \phi_j^{\mathrm{L}} = \phi_k^{\mathrm{L}},$$

which is obviously not valid for linear solutions. Thus, we cannot interchange the role of fluxes and intensities in interface conditions, i.e. we cannot impose a weak interface condition (2.4) for fluxes and a strong interface condition for intensities. The result will be a significant lose of accuracy.

In the case of only two neighboring cells with a common edge (e.g., cells e_j and e_k), we need two continuity conditions. Following the above strategy, we get

$$f_j^{\mathrm{T}} = -f_k^{\mathrm{B}} \quad \text{and} \quad \phi_j^{\mathrm{T}} = \phi_k^{\mathrm{B}}. \tag{2.7}$$

The system of equations (2.2) is closed by imposing continuity conditions (2.5)–(2.7) and the boundary conditions. The Dirichlet boundary conditions specify the intensities associated with the boundary edges. The Neumann boundary conditions prescribe values to the normal components of fluxes.

In the case of orthogonal AMR meshes, the system of discrete equations can be reduced to a system for only cell-centered intensities. The orthogonal AMR meshes has been considered by many authors (see e.g. [2,3]). The major advantage of our scheme is ability to treat non-orthogonal AMR meshes and full diffusion tensors.

3. Computational results

In this section we present computational results which demonstrate the accuracy of our method and the efficiency of our solution techniques. We begin by noting that the system (2.2), (2.5)–(2.7) can be easily reduced to a system of linear algebraic equations for interface intensities with a symmetric positive definite matrix:

$$S\vec{\phi} = \vec{b}, \tag{3.1}$$

where $\vec{\phi}$ denotes the global vector of interface intensities and $\vec{b}$ is the source vector. The proof of this result follows the proofs of similar results in the theory of hybrid-mixed finite elements (see, e.g. [4]). In addition to the properties mentioned above, the matrix S is sparse. A few examples of its stencils are shown in Fig. 2.

We note that if the mesh is orthogonal, all the edge-based unknowns can be locally eliminated leaving an algebraic system of type (3.1) for cell-center intensities [2].

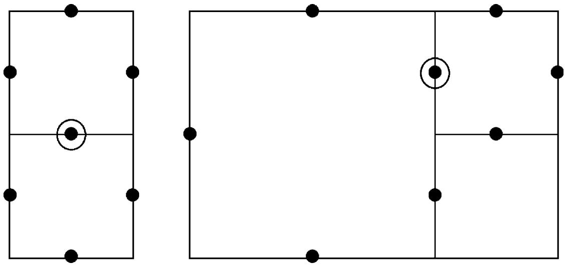

Fig. 2. Stencils of matrix S for edges marked by big circle.

The problem (3.1) can be solved by the preconditioned conjugate gradient method. In the numerical experiments we use the algebraic multigrid preconditioner [10]. It was chosen as an example of a preconditioner applicable to arbitrary matrix stencils.

In *x–y* geometry the SO method is exact on linear solutions for all meshes considered in this section. To the best of our knowledge, there are no other comparable low-order symmetric mixed finite difference schemes on general AMR meshes which are exact on linear solutions.

The first set of calculations addresses the accuracy of the method on highly distorted locally refined meshes in the Cartesian coordinate system. We consider diffusion problem (1.1) on the unit square subject to the Dirichlet boundary conditions. Let

$$\mathbf{D} = \begin{pmatrix} (x+1)^2 + y^2 & -xy \\ -xy & (y+1)^2 \end{pmatrix}$$

and

$$\phi(x,y) = 1 - \tanh\left(\frac{(x-0.5)^2 + (y-0.5)^2}{0.01}\right)$$

be the exact solution. The function has a sharp peak in the middle of the domain and close to zero near the domain boundary. Calculations were performed on a sequence of randomly distorted locally refined and uniformly refined grids. Both sequences of grids are built in two steps. First, we refine grids in logical spaces. After that, we randomly distort positions on mesh points.

Both sequences begin with a 16×16 grid. We use a simple geometric approach to create a few first locally refined meshes. In a logical space, each mesh cell of the initial grid is uniquely identified by two indices i_0 and j_0, $1 \leqslant i_0, j_0 \leqslant 16$. On the first refinement level ($l = 1$), we split mesh cells with indices $4 \leqslant i_0, j_0 \leqslant 13$ into four cells. The new mesh cells are uniquely identified by two indices i_1 and j_1, $1 \leqslant i_1, j_1 \leqslant 20$. On the second refinement level, we split mesh cells with indices $5 \leqslant i_1, j_1 \leqslant 16$ into four cells. On the subsequent levels ($l > 2$) we uniformly refine all mesh cells.

The random grid is generated by moving each mesh point to a random position inside a square centered at the point. The sides of the square are aligned with the coordinate axis and equal 80% of the size of the smallest cell sharing the point. Note that the hanging mesh points are always located in the middle of the associated edges.

The relative L_2 errors ϵ_ϕ and ϵ_f for cell-centered intensities and edge-based fluxes, respectively, are given in Table 1. We observe the second order convergence rate for intensities and the first order convergence rate for fluxes. Even with our naive geometric refinement strategy, we achieve the same accuracy of the solutions but with 4 times less mesh elements. The better performance of our method can be obtained with a refinement strategy based on a posteriori error estimators (see, e.g. [11]) which is beyond the scope of this paper. The locally refined grid at $l = 2$ is shown in Fig. 3.

Table 1 shows also the performance of the *V*-cycle of the algebraic multigrid method. The stopping criterion for the preconditioned conjugate gradient method was the relative decrease in the norm of the residual by factor 10^{-12}. The number of iterations, # it, grows much slower than the number of mesh cells, N. The computational time, denoted by CPU, includes the arithmetical costs for generation of the matrix S, solving the problem (3.1) and recovering the discrete flux and cell-centered intensities. The arithmetical costs per iteration grows linear in N except on coarse grids where cache memory effects play the important role.

The second set of calculations addresses the accuracy of the method on polar meshes in *r–z* coordinate system. We consider a two dimensional problem in the *r–z* coordinate system which is equivalent to the following one dimensional problem:

Table 1
Convergence on a sequence of locally refined and uniform grids

N	ϵ_ϕ	ϵ_f	# it	CPU
Locally refined grids				
256	7.52e−2	8.36e−2	11	0.06 s
556	1.69e−2	3.30e−2	15	0.15 s
988	4.22e−3	1.58e−2	15	0.34 s
3952	1.03e−3	7.40e−3	16	1.51 s
15808	2.61e−4	3.72e−3	17	6.86 s
Uniform grids				
256	7.52e−2	8.36e−2	11	0.06 s
1024	1.63e−2	3.28e−2	12	0.30 s
4096	4.19e−3	1.56e−2	13	1.38 s
16384	1.00e−3	7.33e−3	15	6.35 s
65536	2.50e−4	3.66e−3	17	27.8 s

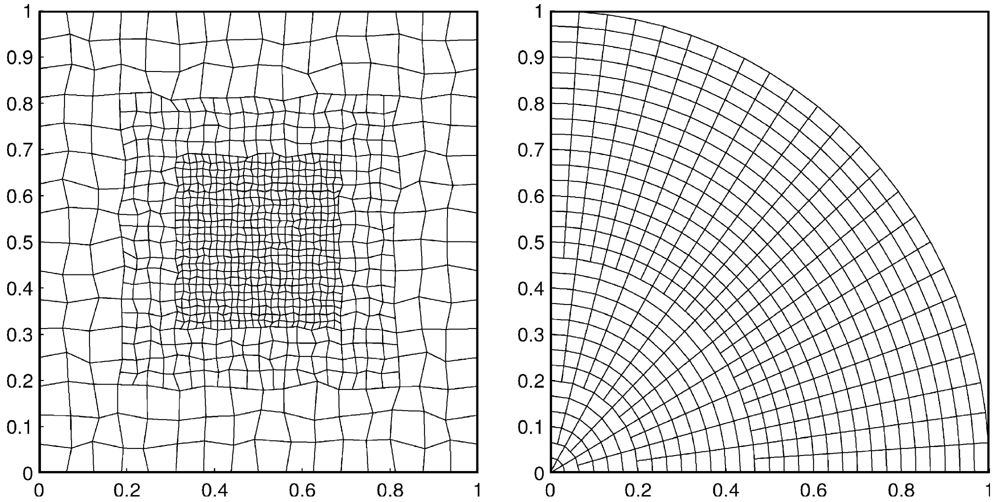

Fig. 3. The AMR grids used in the experiments.

$$-\frac{1}{R^2}\frac{\partial}{\partial R}\left(R^2 D\frac{\partial\phi}{\partial R}\right) = a + bR^2,$$

where R denotes the spherical radius, $R = \sqrt{r^2+z^2}$, and $0 \leqslant R \leqslant 1$. Symmetry preservation on a discrete level is quite important for such problems. Let the diffusion coefficient be scalar and equal to D_1 in the region defined by $0 < R < 0.5$ and equal to D_2 in the rest of the domain. In the r–z coordinate system, we impose the Neumann boundary conditions along $z = 0$ and $r = 0$, and the Dirichlet boundary condition along $R = 1$. This problem admits the analytic solution given in [7].

The computational mesh shown in Fig. 3 is uniform in the r-direction and consists of a few polar blocks with uniform meshes. The size of each of these blocks in the r-direction and the number of angular steps are doubled as we move away from the origin. The discrete solution gives a second order approximation to the exact solution (see, e.g. Fig. 4). However, the detailed analysis shows that the discrete solution exhibits symmetry breaking. In order to study this effect, we plot the trace of the solution at $R = 0.5$ as the function of the polar angle. Note that the amplitude of oscillations is decreased by the grid refinement.

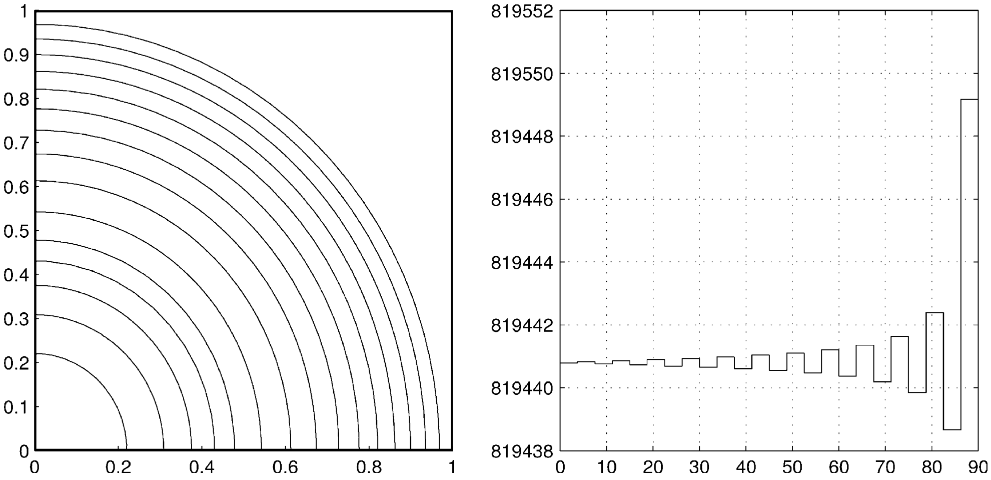

Fig. 4. Solution isolines and solution trace at $R = 0.5$ scaled by 10^7.

4. Conclusions and future work

Our SO method has been shown to be second-order accurate on non-smooth quadrilateral meshes with hanging nodes in both Cartesian and *r–z* geometries. Although the scheme has both cell-center and face-center intensity unknowns, the cell-center unknowns are locally eliminated, resulting in a reduced system consisting only of the face-center unknowns. This reduced system has a symmetric positive-definite coefficient matrix. It has also been shown that the algebraic multigrid method is quite effective for this system with the time-to-solution scaling linearly with the number of unknowns. Thus our SO scheme represents a viable basis for the development of adaptive diffusion solution algorithms for general non-orthogonal quadrilateral meshes.

On *r–z* meshes, our hanging-node scheme does not preserve spherical symmetry. The preservation of such symmetry is a topic for future research. Another topics for future research is the generalization of the hexahedral-mesh SO scheme [6] to hanging-node meshes.

Acknowledgements

The authors thank Dr. Michael Pernice for fruitful discussions of algorithms for generation of AMR meshes.

References

[1] R.W. Anderson, R.B. Pember, N.S. Elliott, An arbitrary Lagrangian–Eulerian method with local structured adaptive mesh refinement for modeling shock hydrodynamics. AIAA paper 2002-0738.

[2] M. Edwards, Elimination of adaptive grid interface errors in the discrete cell centered pressure equation, J. Comp. Phys. 126 (1996) 356–372.

[3] R.E. Ewing, R.D. Lazarov, P.S. Vassilevski, Local refinement techniques for elliptic problems on cell-centered grids I, error analysis, Math. Comp. 56 (1991) 437–461.

[4] F. Brezzi, M. Fortin, Mixed and Hybrid Finite Element Methods, in: Springer Series in Computational Mathematics, volume 15, Springer Verlag, Berlin, 1991.

[5] J. Hyman, J. Morel, M. Shashkov, S. Steinberg, Mimetic finite difference methods for diffusion equations, Comp. Geosciences 6 (3–4) (2002) 333–352.

[6] J. Morel, M. Hall, M. Shashkov, A local support-operators diffusion disretization scheme for hexahedral meshes, J. Comp. Phys. 170 (2001) 338–372.

[7] J. Morel, R. Roberts, M. Shashkov, A local support-operators diffusion disretization scheme for quadrilateral *r–z* meshes, J. Comp. Phys. 144 (1998) 17–51.

[8] R.B. Pember, J.B. Bell, An adaptive Cartesian grid method for unsteady compressible flow in irregular regions, J. Comp. Phys. 120 (1995) 278–304.

[9] J.J. Quirk, A parallel adaptive grid algorithm for computational shock hydrodynamics, Appl. Numer. Math. 20 (1993) 427–453.

[10] K. Stüben, Algebraic multigrid (AMG): experiences and comparisons, Appl. Math. Comput. 13 (1983) 419–452.

[11] R. Verfürth, A posteriori error estimation and adaptive mesh-refinement techniques, J. Comput. Appl. Math. 50 (1994) 67–83.

[12] D.D. Zeeuw, K.G. Powell, An adaptively refined Cartesian mesh solver for the Euler equations, J. Comp. Phys. 104 (1993) 56–68.