Quadrilateral mesh adaptation by area functional

Sanjay Kumar Khattri*

Department of Mathematics, University of Bergen, Brunsgt 12, 5008 Bergen, Norway

Abstract

We present a new approach for robust mesh adaptation. Optimizing a given quadrilateral mesh to the requirement of the problem is the topic of this paper. This research work deals with generating adaptive quadrilateral meshes using a discrete area functional. Optimization of this functional tries to equidistribute error/gradient/factors that cause error (analytically given or computed on the initial mesh) over each quadrilateral cell. This paper presents various examples of generating adaptive meshes.

Keywords: Quadrilateral mesh; Area functional; Adaptive function, MPFA; Jacobian matrix

1. Introduction

It is widely accepted that quadrilateral meshes in 2D and hexahedral meshes in 3D are better than other meshes in many ways. But unfortunately generating adaptive quality quadrilateral and hexahedral cells has not been much explored.

The accuracy of any numerical method (FD, FV, FE, etc.) depends highly on the quality (smoothness, convexity, orthogonality, linearity, etc.) of the underlying meshes, and the number of nodes and cells in the mesh. It is not always feasible to blindly refine the mesh in the hope of capturing the physics because of the computational resources. It is desired to adapt the grid to the requirement of the underlying problem. We thus generate more grid points and cells in the region of more activity (non-linear changes, high solution gradients, etc.), while maintaining the quality of the underlying mesh. It is always desirable to concentrate the grid points and cells in the region of greatest chemical and physical change instead of the region where the solution has nearly zero gradient. The aim of this paper is to adjust the mesh by getting a feeling of the solution on the initial mesh.

This article is divided as follows: in section 2 a brief introduction to the discrete area functional is presented, in section 3 some numerical examples are presented, and finally section 4 concludes this article.

* Tel.: +47 555 84871; Fax: +47 555 89672; E-mail: sanjay@mi.uib.no

Computational Fluid and Solid Mechanics 2005
K.J. Bathe (Editor)

2. Discrete area functional

The first study of area functionals was done by [1]. Following the author's conclusions they have not been used till now for generating an adaptive mesh. Let a quadrilateral mesh consist of I internal nodes, and let node k be surrounded by four quadrilaterals (the mesh can be unstructured). The authors would like to propose the following form of the area functional for an adaptive grid:

$$F(\mathbf{x},\mathbf{y}) = \sum_{k=1}^{I} [\sum_{i=1}^{4} (J(k_i))^2 \cdot S(k_i)] \tag{1}$$

where $J(k_i)$ is the Jacobian for cell i at node k, and $S(k)$ is the adaptive function, $S(k_i)$ is the value of the adaptive function $S(k)$ at the center of the cell i. $S(k_i)$ must be greater than zero for each cell. $S(k_i)$ is assumed constant for a cell. Optimization of Eq. (1) will equi-distribute the product of the area of each cell and the adaptive function. The cell which has the largest value of $S(k_i)$ will be the smallest. If $S(k_i)$ is the same for each cell then the optimization of Eq. (1) will try to generate cells of equal area.

The Jacobian matrix at a node for a cell is the matrix whose columns are the covariant vectors at that node, and the Jacobian is the determinant of this matrix.

For the node O and the surrounding cells 1, 2, 3 and 4 as shown in Fig. 1 the Jacobian will be expressed as

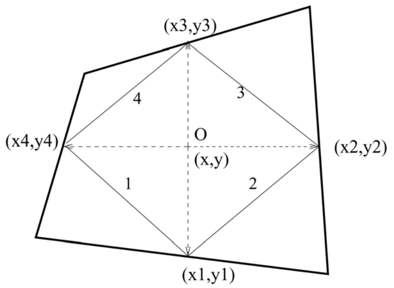

Fig. 1. 2D mesh.

$$J(O1) = \begin{vmatrix} (x4-x) & (x1-x) \\ (y4-y) & (y1-y) \end{vmatrix}, \quad J(O2) = \begin{vmatrix} (x2-x) & (x1-x) \\ (y2-y) & (y1-y) \end{vmatrix}$$

$$J(O3) = \begin{vmatrix} (x2-x) & (x3-x) \\ (y2-y) & (y3-y) \end{vmatrix}, \quad J(O4) = \begin{vmatrix} (x4-x) & (x3-x) \\ (y4-y) & (y3-y) \end{vmatrix}$$

Two important properties of the functional $F(\mathbf{x}, \mathbf{y})$ are:

- The critical point of the functional is a grid for which the product of cell area and the adaptive function is the same for every cell.
- The Hessian is semipositive definite.

Among all the meshes with the same boundary, it is required to determine the one whose inner points minimize the functional (1). The minimization can be performed by iterative algorithms such as Truncated Newton.

A quadrilateral mesh is called convex if each of its cells is convex. A quadrilateral will be convex if the four triangles in it have positive area. Since the Jacobian at a point for a cell is twice the area of the corresponding triangle, the minimization of this functional will make sure that all the triangles of a quadrilateral cell have positive area (given such a minimum exists).

3. Numerical example

Two kinds of numerical example are presented. In the first numerical example, the adaptive function is given analytically, and in the second the adaptive function is derived from the discrete gradient of the solution computed on the initial mesh.

3.1. Example 1

In this example, the initial mesh is a 64×64 cartesian mesh. The adaptive functions are given as

$$S(x,y) \;=\; 1.0 + e^{-\tanh(2000.0(x-0.5)^2 + (2000.0(y-0.5)^2 - 12.50000000)} \tag{2}$$

$$S(x,y) \;=\; 1.0 + e^{\tanh(10.0(x-0.5)^2 + 10.0(y-0.5)^2 - 18.750000000)} \tag{3}$$

$$\begin{aligned} S(x,y) = 1.0 \;+\; & e^{-\tanh(200.0(x-0.25)^2+200.0(y-0.25)^2-12.50000000)} \\ +\; & e^{-\tanh(200.0(x-0.75)^2+200.0(y-0.25)^2-12.50000000)} \\ +\; & e^{-\tanh(200.0(x-0.25)^2+200.0(y-0.75)^2-12.50000000)} \\ +\; & e^{-\tanh(200.0(x-0.75)^2+200.0(y-0.75)^2-12.50000000)} \end{aligned} \tag{4}$$

$$\begin{aligned} S(x,y) = 1.0 \;+\; & e^{-\tanh(200.0(x-0.5)^2+200.0(y-0.5)^2-12.50000000)} \\ +\; & e^{-\tanh(200.0(x-0.25)^2+200.0(y-0.25)^2-12.50000000)} \\ +\; & e^{-\tanh(200.0(x-0.75)^2+200.0(y-0.25)^2-12.50000000)} \\ +\; & e^{-\tanh(200.0(x-0.25)^2+200.0(y-0.75)^2-12.50000000)} \\ +\; & e^{-\tanh(200.0(x-0.75)^2+200.0(y-0.75)^2-12.50000000)} \end{aligned} \tag{5}$$

Figures 2(a), (b), (c) and (d) show the converged adaptive meshes produced by Eqs. (3), (4), (5) and (2) respectively. It is clear from Fig. 2 that all the cells generated by the adaptive functional are convex.

3.2. Example 2

We solve Eq. (6) on a unit square using the multipoint flux approximation (MPFA-O) methods [2]. MPFA are higher-order cell-centered finite volume methods. The exact solution is Eq. (7). We enforce the solution inside the domain by the $f(x,y)$ term and boundary condition:

$$-\nabla \cdot (\nabla u) = f(x,y) \tag{6}$$

$$u(x,y) = e^{-100(x-1/2)^2 - 100(y-1/2)^2} \tag{7}$$

In this example, the adaptive function is computed from the gradient of the discrete solution. The initial mesh is a 31×31 cartesian mesh, as shown in Fig. 2(f). Let the solution be u_h and ∇u_h be its gradient by the MPFA-O discretization.

The adaptive function is defined as follows:

$$S(x,y) = 1.0 + e\left[\frac{(\|u_h\| + \|\nabla u_h\|)}{1.0 + (\|\nabla u_h\|)}\right] \tag{8}$$

Table 1 shows l_2 and l_∞ errors on the initial and adaptive meshes.

4. Conclusion

In this article a new robust (quadrilaterals remain convex) idea for generating adaptive quadrilateral meshes is presented. This idea can be very useful in solving evolutionary problems (parabolic, hyperbolic equations) on adaptive meshes. The author is extending the same idea for a hexahedral mesh in 3D.

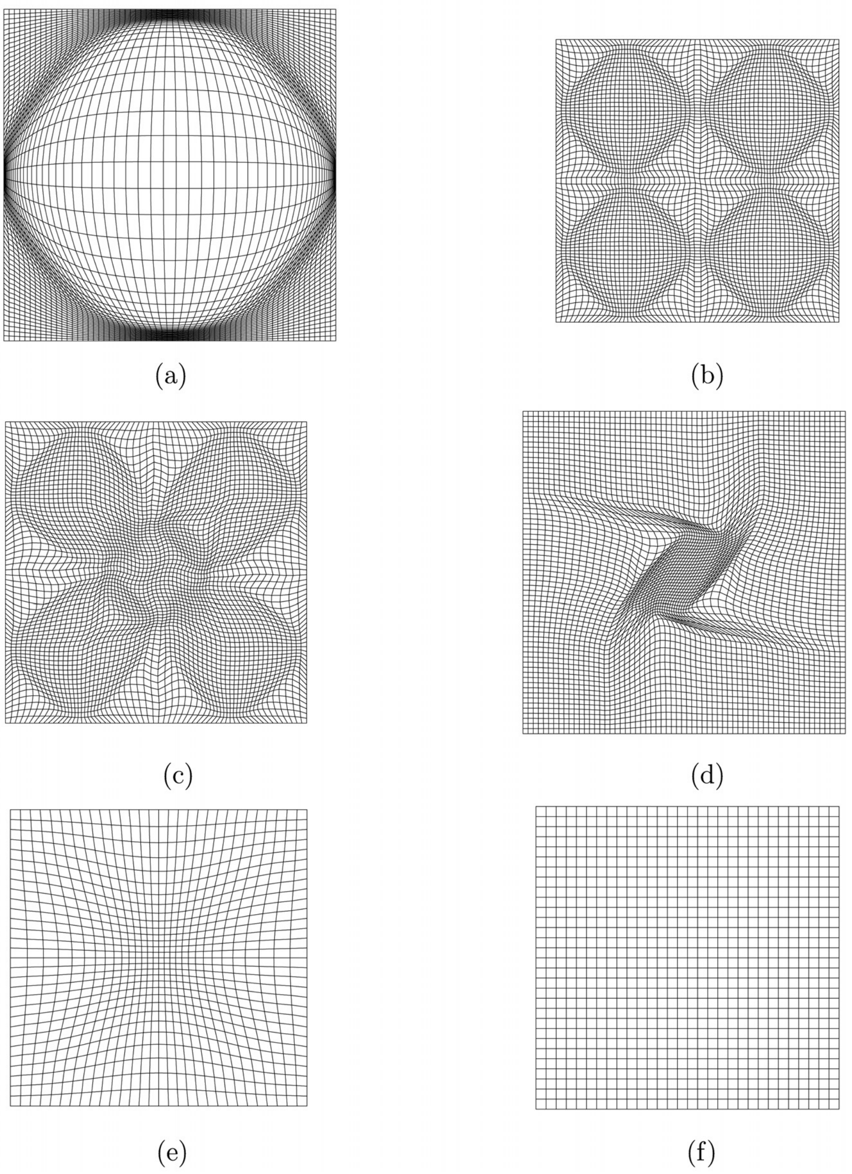

Fig. 2. Meshes a, b, c, d, e were adapted by Eqs (2,3,4,1) respectively. Mesh f is the initial mesh for Eq. (5).

Table 1
Example 2 – error in the L_2 and L_∞ norms

Mesh	$\|u - u_h\|_{l_2}$	$\|u - u_h\|_{l_\infty}$
Initial Fig. 2(f)	0.00259862	0.0256077
Adapted by Eq. (8) Fig. 2(e)	0.000906869	0.00759576

References

[1] Castillo JE. On variational grid generation. Ph.D. thesis, The University of New Mexico, Albuquerque, New Mexico, 1987.

[2] Aavatsmark I. An introduction to multipoint flux approximations for quadrilateral grids. Comput Geosci 2002;6:405–432.