

Finite element algorithm with adaptive quadtree-octree mesh refinement

G. P. Nikishkov*

(Received 18 October 2004; revised 24 January 2005)

Abstract

Certain difficulties with the use of quadrilateral or hexahedral finite elements are related to mesh refinement and to element compatibility and quality after refinement. In this paper, special refinement elements are presented that make possible connecting two special elements to one edge of an 8 node quadrilateral element (2D). The main idea in refinement elements is to place some midside nodes outside the element area and to modify element shape functions in order to maintain continuity at a refinement edge. Special refinement elements allow to adaptively refine mesh in such a way that it fits the quadtree data structure. In the three-dimensional case, hexahedral elements with 20 nodes are employed. Placement of some nodes outside the element volume allows us to create a compatible octree refinement scheme.

*University of Aizu, Aizu–Wakamatsu, JAPAN. <mailto:niki@u-aizu.ac.jp>

See <http://anziamj.austms.org.au/V46/CTAC2004/Niki> for this article, © Austral. Mathematical Soc. 2005. Published 8 March 2005, amended March 18, 2005. ISSN 1446-8735

Contents

1	Introduction	C16
2	Special quadratic quadrilaterals	C17
3	Finite element quadtree mesh refinement	C20
3.1	Finite element solution	C20
3.2	Mesh Refinement	C21
3.3	Results	C24
4	Octree mesh refinement in 3D	C24
5	Conclusion	C27
	References	C28

1 Introduction

Quadrilateral and hexahedral finite elements are widely used for problem solutions. However, it is difficult to perform mesh refinement preserving compatibility and element quality after refinement [1].

Methods of quadrilateral mesh refinement in the case of connecting geometrically incompatible elements are presented in [3, 4]. Forting and Tanguy [3] consider connection of two Lagrangian 9-node 2D elements to one edge of a bigger element of the same type. The displacement compatibility is enforced by introduction of additional constraints in the global finite element equation system. This provides convergence of the finite element solution but the finite element mesh remains geometrically incompatible at a refinement edge. Kagan et al. [4] focus on the so-called p -refinement of the mesh, during which additional nodes are added to the element without

changing its shape. The proposed technique to enforce compatibility is a coordinate transformation introduced into the refined element.

Our approach to mesh refinement is to replace one 2D quadratic quadrilateral element by four elements of the same type. In order to perform 1:2 mesh refinement while maintaining element quality and solution compatibility, we use special refinement elements, with some nodes outside the element area. An analogous approach in the three-dimensional case allows us to create special elements, four of which can be connected to one face of an ordinary element.

The following algorithm is used here for adaptive mesh refinement with control of local errors. Starting with a coarse mesh, element refinement iterations are carried out. At each iteration, the finite element equation system is solved that provides field nodal values for a current mesh configuration. Local element errors are estimated and compared to the specified error tolerance. Elements with excessive errors are divided into four special elements (2D) or eight special elements (3D). The mesh refinement iteration terminates when no element splits occur during an iteration.

2 Special quadratic quadrilaterals

The isoparametric quadratic quadrilateral element shown Figure 1,a has eight nodes, which are used both for element shape definition and for field interpolation [2, e.g.]. Any function specified at element nodes is interpolated inside the element with the use of shape functions N_i , which are defined in local coordinates ξ, η ($-1 \leq \xi, \eta \leq 1$):

$$u(\xi, \eta) = \sum N_i(\xi, \eta)u_i;$$

$$N_i = \frac{1}{2}(1 - \xi^2)(1 + \eta\eta_i), \quad i = 2, 6;$$

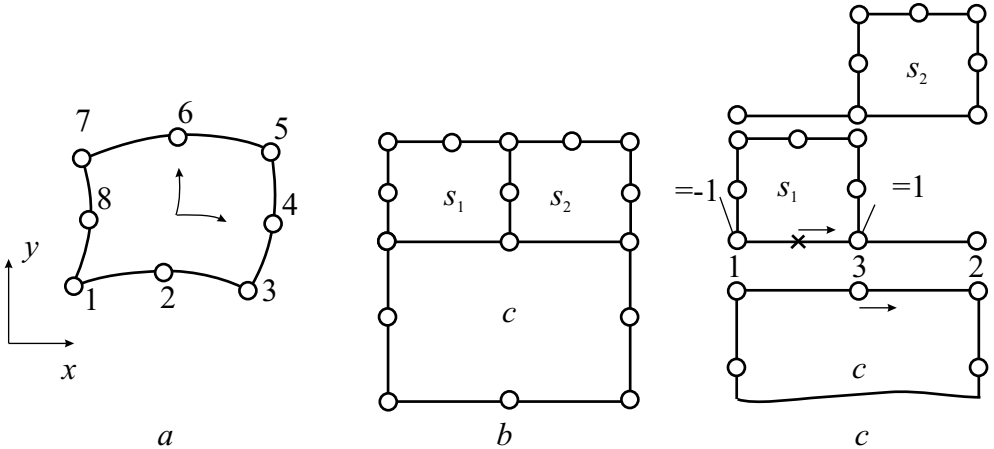


FIGURE 1: a) Quadratic isoparametric quadrilateral element; b) Mesh refinement 1:2; c) The special elements s_1 and s_2 have been vertically shifted in order to illustrate the node positions of each element.

$$\begin{aligned}
 N_i &= \frac{1}{2}(1 + \xi\xi_i)(1 - \eta^2), \quad i = 4, 8; \\
 N_i &= \frac{1}{4}(1 + \xi\xi_i)(1 + \eta\eta_i) - \frac{1}{2}(N_{i-1} + N_{i+1}), \quad i = 1, 3, 5, 7. \quad (1)
 \end{aligned}$$

Here u is a function value at some point inside the element; u_i are function values at element nodes; ξ_i and η_i are values of the local coordinates ξ and η at the node i . In the formula for N_1 we interpret N_0 as N_8 .

With the use of ordinary quadrilateral elements it is difficult to refine the mesh. Mesh refinement with quadrilaterals can lead to severe element distortions since, for compatibility reasons, the edges of adjacent elements must coincide exactly.

We propose here special quadratic quadrilateral elements, which allow to connect two elements to one edge of the bigger element. Special refinement elements are characterized by placement of some midside nodes outside the element area and by modification of the shape functions. An idea of special

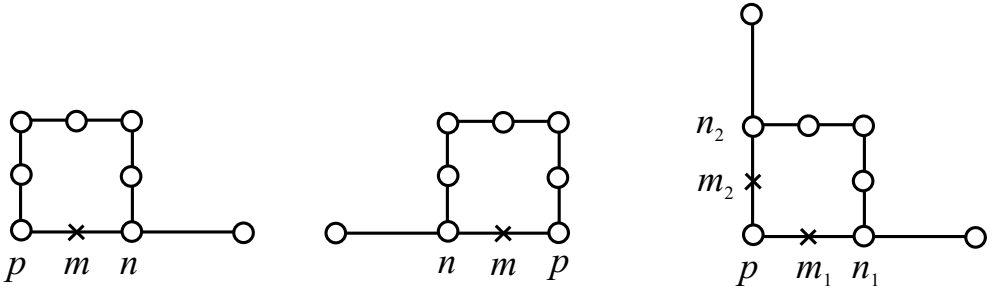


FIGURE 2: Typical configurations of special refinement elements.

elements for mesh refinement is illustrated in Figure 1,b-c. Mesh refinement 1:2 is performed with two special elements s_1 and s_2 , which have nodes located outside of the element area.

Modification of the shape functions in the special element is done in the following way. In order to have field continuity along the element edge 1-3-2 the function value at point $\xi = 0$ in the special element s_1 should be equal to the function value at point $\rho = -0.5$ in the conventional neighboring element c : $u(\xi = 0) = u(\rho = -0.5)$. Then the compatibility equation is

$$u(\xi = 0) = u(\rho = -0.5) = \frac{3}{8}u_1 + \frac{3}{4}u_3 - \frac{1}{8}u_2. \quad (2)$$

The modified shape functions are combinations of the ordinary shape functions:

$$N'_1 = N_1 + \frac{3}{8}N_2, \quad N'_2 = -\frac{1}{8}N_2, \quad N'_3 = N_3 + \frac{3}{4}N_2. \quad (3)$$

Figure 2 shows typical configurations of special refinement elements. If a special element contains two outside nodes, as shown in Figure 2 on the right, then modification (3) is applied twice using nodes m_1 and m_2 .

3 Finite element quadtree mesh refinement

3.1 Finite element solution

I demonstrate the finite element algorithm with quadtree mesh refinement on a problem of surface approximation with minimization of a global error and with limitation of a local error. Assume that an approximated surface is defined at each point. A simple example of such definition is a height function defined on a two-dimensional domain. Another possibility is definition of a surface by a point cloud. In the latter case assume that the surface height is obtained at each point by interpolation of values using neighboring points.

An element approximation error E_e for a surface segment covered by the element e is

$$E_e = \int_{A_e} (f - u)^2 dA, \quad (4)$$

where f is the exact surface height, u is the height approximation for the e th element, and A_e is the element area. The element error serves as a natural local error indicator for quadtree mesh refinement.

Determine values of the height u for the current element mesh by minimization of a global approximation error, which is the sum of element approximation errors

$$E = \int_A (f - u)^2 dA = \sum_e E_e = \sum_e \int_{A_e} (f - N_i u_i)^2 dA. \quad (5)$$

Minimization of the global error functional leads to a global finite element equation system:

$$\mathbf{K}\mathbf{U} = \mathbf{R}, \quad k_{ij}^e = \int_{A_e} N_i N_j dA, \quad r_i^e = \int_{A_e} N_i f dA, \quad (6)$$

where \mathbf{K} is a global "stiffness" matrix; \mathbf{U} is a vector of nodal height values and \mathbf{R} is a right-hand side vector. The global matrix \mathbf{K} and the right-hand

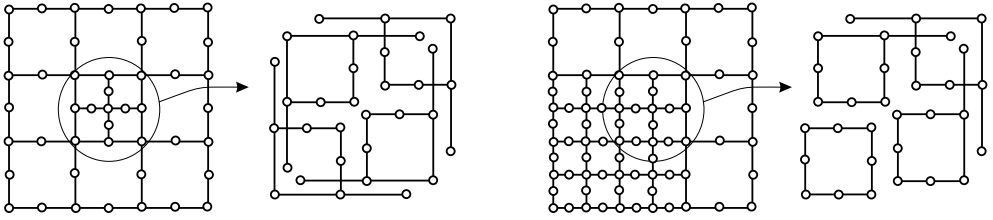


FIGURE 3: Four special refinement elements replace one quadrilateral element.

side vector \mathbf{R} are respectively composed of element “stiffness” matrices \mathbf{k}_e and element vectors \mathbf{r}_e using standard finite element assembly procedures.

For the solution of the finite element equation system (6) it is reasonable to use iterative methods such as preconditioned conjugate gradient method since good initial approximation for the solution vector \mathbf{U} can be provided in the beginning and after each refinement step.

3.2 Mesh Refinement

Surface approximation starts with some minimal mesh consisting of ordinary quadrilaterals. When the approximation domain is a square then it is possible to use just one quadratic quadrilateral element as a starting mesh. According to the adaptive refinement procedure, carry out mesh subdivision for elements where an error indicator of the type (4) is greater than the specified error tolerance.

A quadrilateral element to be subdivided is replaced by four quadrilateral elements. See Figure 3 for examples of the element subdivision for the case of regular surrounding elements and for the case of a partially refined mesh.

Mesh data after any number of refinements (one element is replaced by four elements) can be conveniently kept in a quadtree data structure. Every

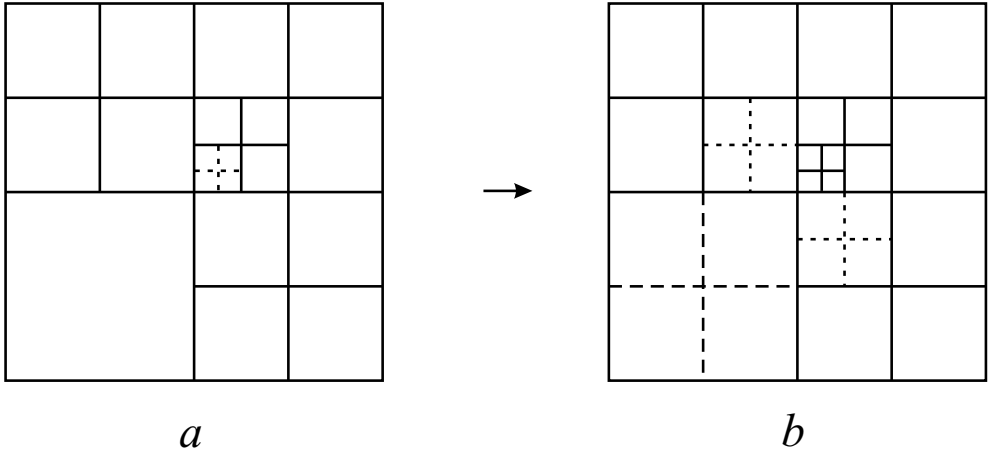
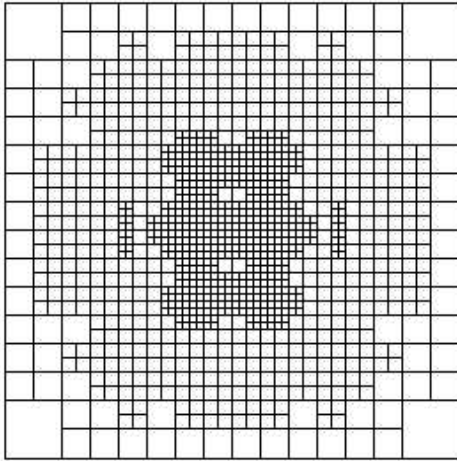


FIGURE 4: Split of one element into four (a). Quadtree balancing: split of neighbors, then split of the next neighbor (b). Initial element split and element splits for quadtree balancing are shown by dashed lines.

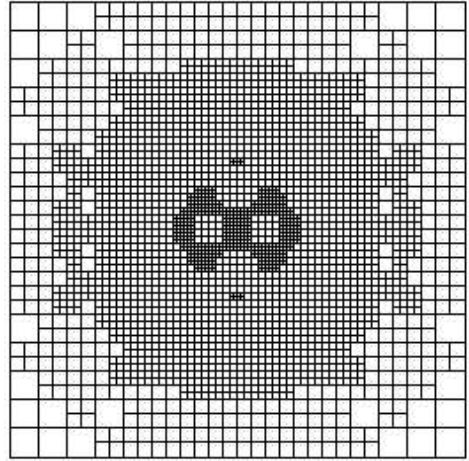
node in the quadtree topologically corresponds to a square. Internal nodes have exactly four children. Leafs (nodes without children) represent actual elements, which are ordinary elements or special refinement elements with outside nodes.

After splitting elements with excessive errors, the refined mesh may have such configuration in which it is impossible to provide continuity across some element boundaries. Figure 4,a presents an example of a quadtree with discontinuity element splits.

A quadtree is said to be unbalanced if there exists a square that is adjacent to more than two squares at one of its edges [5]. In a balanced quadtree, adjacent quadtree blocks differ by at most one level, that is, not more than two elements can share one edge of the neighboring element. Balance a quadtree by additional subdivisions of some elements as shown in Figure 4,b. This example demonstrates that adaptive refinement may necessitate secondary subdivision of adjacent elements. For any balanced quadtree, ensure continu-



Error = 0.0005, 6 iterations



Error = 0.0001, 7 iterations

FIGURE 5: Quadtrees of quadratic quadrilaterals characterized by different height approximation errors.

ity using special refinement elements with outside nodes and modified shape functions.

During quadtree balancing, a given leaf should be split if its level is higher than any level of its neighboring leaves by 2. Newly created leaves should be put into list of candidates for splitting since they may necessitate secondary subdivisions as shown in Figure 4,b.

The process of surface modeling by a mesh of quadratic quadrilateral elements begins with some initial mesh. During mesh refinement step, first, the finite element equation system (6) is assembled and solved. The height values are obtained for all mesh nodes. Then for each element estimate the local element error E_e (4) and compare to the specified error tolerance. Splitting of elements with excessive errors is performed. Balance the element quadtree at the end of each refinement iteration. Terminate mesh refinement when no splits occur during an iteration.

3.3 Results

The proposed finite element algorithm with quadtree adaptive refinement is demonstrated on the approximation of a surface defined analytically on a square domain:

$$f = 0.5e^{-0.16(x^2+y^2)} \sin(2x) \cos(2y),$$

$$-5 \leq x \leq 5, \quad -5 \leq y \leq 5.$$

See that the height range for the above function is $[-0.5, 0.5]$ and the size in the height direction is 1.

The following error measure is used for mesh refinement,

$$\bar{E}_e = \sqrt{\frac{1}{A_e} \int_{A_e} (f - u)^2 dA},$$

where f is the exact surface height; u is the height approximation and A_e is the element area. This error measure \bar{E}_e differs from the error (4): E_e is divided by the element area and the square root is taken. The error indicator \bar{E}_e is measured in length units and is regarded as some averaged absolute error in height over an element.

Figures 5 and 6 show the results of surface approximation by quadratic quadrilateral elements. Figure 5 shows element quadtrees for two error tolerance values: 0.0005 (6 iterations) and 0.0001 (7 iterations). Figure 6 presents visualization of the approximated surface (error tolerance 0.0005) with the use of Java 3D.

4 Octree mesh refinement in 3D

An octree subdivision of volume is a three-dimensional generalization of the two-dimensional quadtree refinement procedure. We assume that an initial

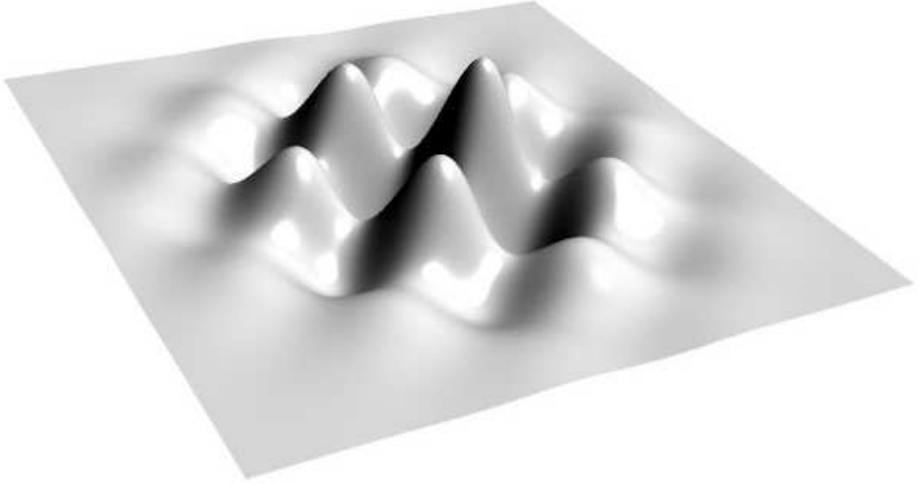


FIGURE 6: Visualization of a surface approximated by quadratic quadrilaterals.

finite element mesh consists of 20-node hexahedral elements. Figure 7,a shows the connection of four special refinement elements to one face of an ordinary element. Some nodes of the special element are moved outside its volume thus providing continuity across a refinement face.

A face of the special element for 1:4 mesh refinement and a face of the ordinary element are depicted in Figure 7,b. In the special element, its volume is connected to the face area 1-3-(5)-7. The local coordinate area $-1 \leq \xi, \eta \leq 1$ is also related to the face area 1-3-(5)-7. As in the 2D case, it is possible to write down a compatibility equations for nodes (2), (4), (5), (6) and (8). The following modification of the shape functions allows one to create the special element for 1:4 mesh refinement:

$$N'_1 = N_1 + \frac{3}{8}(N_2 + N_8) - \frac{3}{16}(N_4 + N_6) - \frac{1}{4}N_5;$$

$$N'_2 = -\frac{1}{8}N_2 - \frac{3}{16}(N_4 + N_6) - \frac{1}{4}N_5;$$

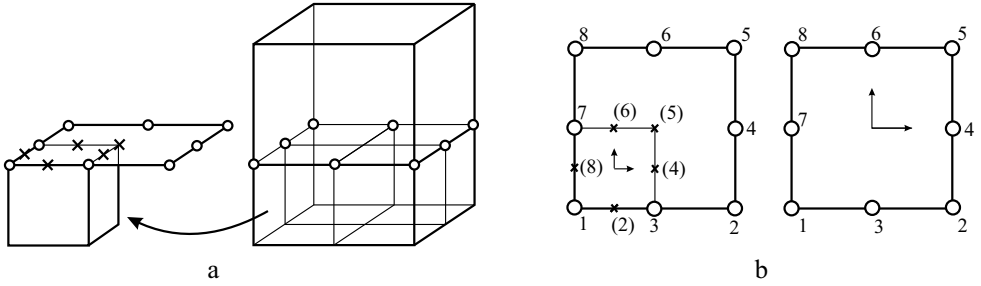


FIGURE 7: a) Connection of four special elements to one face of an ordinary element; b) Node numbering on faces of a special element and an ordinary element.

$$\begin{aligned}
 N'_3 &= N_3 + \frac{3}{4}(N_2 + N_4) + \frac{1}{2}N_5 + \frac{3}{8}N_6; \\
 N'_4 &= \frac{3}{8}N_4 + \frac{1}{2}N_5 + \frac{1}{4}N_6; \\
 N'_5 &= -\frac{1}{4}N_5 - \frac{3}{16}(N_4 + N_6); \\
 N'_6 &= \frac{3}{8}N_6 + \frac{1}{2}N_5 + \frac{1}{4}N_4; \\
 N'_7 &= N_7 + \frac{3}{4}(N_6 + N_8) + \frac{1}{2}N_5 + \frac{3}{8}N_4; \\
 N'_8 &= -\frac{1}{8}N_8 - \frac{3}{16}(N_4 + N_6) - \frac{1}{4}N_5.
 \end{aligned} \tag{7}$$

During adaptive mesh refinement with the octree subdivision, special elements with one, two or three enlarged faces appear in the mesh. Perform modification of the shape functions of the type (7) for each such enlarged face.

5 Conclusion

We have discussed a finite element procedure with quadtree (2D) and octree (3D) adaptive mesh refinement. In the two-dimensional case, special refinement elements based on quadratic quadrilateral elements are developed. It is possible to connect two special elements to one edge of an ordinary quadrilateral quadratic element. The special refinement element is characterized by placement of one or two midside nodes outside the element area and by modification of element shape functions. This helps maintain continuity across all element edges.

Utilise a local error indicator for the adaptive mesh subdivision. Subdivide elements with excessive local errors into four elements each. After such subdivisions the element mesh is topologically equivalent to a quadtree. The quadtree data structure stores element data and enables navigation through the mesh.

Testing of the proposed finite element quadtree refinement algorithm is done on a surface height approximation problem. Several refinement iterations are enough for surface approximation with high precision.

Three-dimensional adaptive mesh refinement is based on the octree data structure. An initial mesh is composed of 20-node hexahedral elements. Four special refinement elements are connected to one face of the ordinary element with preserving continuity of sought functions and geometry.

Special refinement elements for the quadtree and octree mesh subdivision are easy to implement. The presented finite element algorithm can be used for solution of various problems.

References

- [1] P. J. Frey and P.-L. George, *Mesh Generation. Application to Finite Elements*. Hermes, 2000. C16
- [2] K.-J. Bathe, *Finite Element Procedures*, Prentice-Hall, 1996. C17
- [3] M. Fortin and P. Tanguy, A non-standard mesh refinement procedure through node labelling. *Int. J. Numer. Meth. Eng.*, 20:1361–1365, 1984. C16
- [4] P. Kagan, A. Fischer and P. Z. Bar-Yosef, Mechanically based design: adaptive refinement for B-spline finite element. *Proc. Int. Conf on Shape Modeling and Applications, Genova, Italy, May 7–11, 2001*, 345–352, 2001. C16
- [5] M. de Berg, M. van Kreveld, M. Overmars and O. Schwarzkopf, *Computational Geometry. Algorithms and Applications*. Springer, 2000. C22