

On a nested refinement of anisotropic tetrahedral grids under Hessian metrics

Shangyou Zhang *

Abstract

Anisotropic grids, having drastically different grid sizes in different directions, are efficient and even necessary in many applications. The grid sizes in certain directions are usually determined locally by a Hessian metric. In this paper, we will show that, given an arbitrary set of mid-edge points, for example, produced by a Hessian metric, a tetrahedral grid can be nestedly refined without any other new points introduced. This paper is to eliminate the skepticism that such a nested refinement must generate pollutions and that the grids might become isotropic locally due to the unwanted auxiliary points. Some algorithms are provided and numerical tests are presented.

Mathematics Subject Classification: 65N50, 65N30.

Keywords: adaptive refinements, tetrahedral finite elements, Hessian metric, nested refinement.

1 Introduction

Adaptive refinements occur in many problems of scientific computation. After an approximate solution is obtained, by some error indicators, the grid used in the computation can be refined, either locally or globally or directionally, so that the discrete solution on the new grid would be more accurate and economical to compute. The traditional grids are isotropic where the grid sizes are uniform in all directions locally. Then some auxiliary points might have to be introduced, not for a better accuracy of the numerical solution, but for the shape regularity of the grid or for conformity of the grid, cf. [7]. For example, in Figure 1, when the edge AC of triangle ABC needs to be refined into 4 sub-edges, due to the shape regularity requirement of isotropic grids, edges AB and BC must be refined once too. In such a case, the sequence of refined grids are not nested and two auxiliary points are introduced.

*Department of Mathematical Sciences, University of Delaware, Newark, DE 19716, USA. szhang@udel.edu

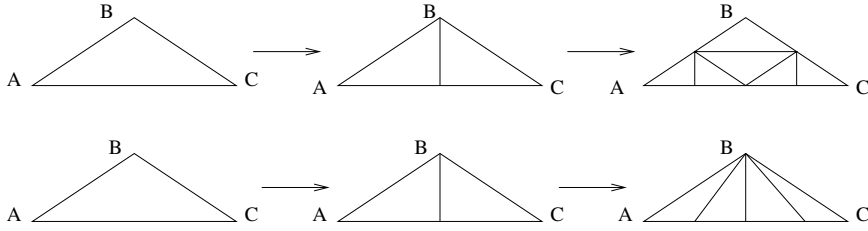





Figure 1: A standard adaptive refinement (top) and an anisotropic refinement (bottom).

Anisotropic grids, having drastically different grid sizes in different directions, are efficient and even necessary in many applications. For example, in the bottom graph of Figure 1, one edge of the triangle is refined into 4, but the other 3 edges are not changed. Depending on the metric, an anisotropic grid might be more isotropic in grid size than an isotropic or a uniform grid. For example, in the finite element method, the error of a computed solution u_h to the smooth solution u is often bounded by the second order Taylor terms of u , $|\xi^T(\nabla^2 u(x))\xi|$. From a computed finite element solution, one can get some kind of approximation to the Hessian matrix $(\nabla^2 u(x))_{2 \times 2}$ or $(\nabla^2 u(x))_{3 \times 3}$. One can refine the previous grid accordingly to reduce the edge length differently at different locations and in different directions, but not remesh. The goal is to make the value of bilinear form $|\xi^T(\nabla^2 u)(x)\xi|$ minimal, where $\xi = y - x$ is the vector between a general point y on the element to the Taylor expansion point x . Here, a long edge (under the Euclid distance) might be “short” under the Hessian metric. For example, we may have a situation where we want the first component of ξ much smaller than the other two components in size. Then we would like to have local tetrahedra of a much smaller size in x direction than that in y and z directions. This would lead to anisotropic grids. Again, an anisotropic or degenerate element under the Euclid metric would be a regular element under the Hessian metric. For more information on the anisotropic grid refinement, we refer to [1, 2, 3, 4, 5, 6, 8] and [9]. When doing adaptive grid refinements, it is common that some edges need to be refined and midpoints on those edges are introduced. In 2D, one can just connect the vertex opposite to the new mid-edge point as  if there is only one new point, or one can connect the two new points in addition as  if two new points are introduced. So the problem is trivial in 2D to get a nestedly sequence of anisotropic grids. In 3D, the situation could be complicated (cf. [10]) as people tend to believe the unwanted refinement would propagate. It is shown in this paper that a nested refinement sequence without any auxiliary points is also possible in 3D, for any given tetrahedral grids. In other words, by a new algorithm, there will be no refinement pollution at all in 3D too.

The rest of the paper is presented as follows. In Section 2, we study all cases of tetrahedral refinement when different numbers of mid-edge points are introduced to one tetrahedron. In Section 3, we define two algorithms and we prove that the nested refinement is guaranteed in both algorithms. But in one of them, there is no auxiliary point added. In Section 4, we show a simple numerical test, where our new algorithms are implemented.

2 Nested refinement with mid-edge points

Before we study the 3D case, let us consider the 2D case again, stated in Section 1. For refining a triangle, we may have three cases, either one or two or three mid-edge points are introduced. The triangle can be considered as a face triangle of a tetrahedron. If one mid-edge point is added to a triangle, then it is refined into two subtriangles by connecting the new point with the opposite vertex, . If three new mid-points are added to the three edges, the triangle is refined into four congruent subtriangles. There is no choice in these cases. However, if two mid-edge points

are added, there is a choice in adding a new line to refine the triangle into three subtriangles, see Figure 2. This is the only decision we have to make in the type of adaptive refinements of tetrahedral grids considered in the paper. Such decisions appear in the following six cases, 2(a), 3(b), 3(c), 4(a), 4(b) and 5. The decisions would be made appropriately so that the new grid is conformal in the sense of the finite element [7, 10]. To be specific, often we do have choices in between the left and the right face refinements of Figure 2, when subdividing one tetrahedron alone. But considering the refinement on the neighboring tetrahedra, without introducing more grid points, many choices will be no longer valid.

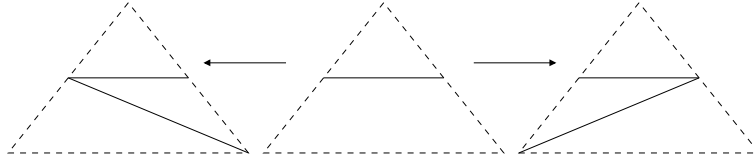


Figure 2: A choice (left or right refinement) in refine a face triangle.

Let us list all cases of refining one tetrahedra after introducing mid-edge points. By the number of new points, we have the following cases, no mid-edge point, 1, 2, 3, 4, 5, or 6 new mid-edge points.

Case 1. One point is added to a tetrahedron.

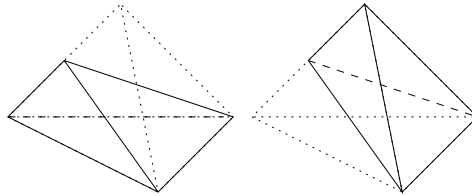


Figure 3: Case 1: One mid-edge point. Refined into two subtetrahedra.

This is the simplest case. There is no choice. Two face triangles are refined while the other two stay. The tetrahedron is refined into two as shown in Figure 3.

Case 2. Two points are added to a tetrahedron. There are two sub-cases.

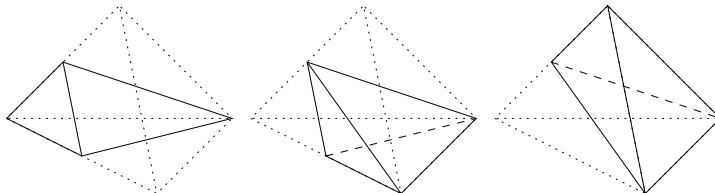


Figure 4: Case 2(a): Two mid-edge points. Refined into three with a choice.

Case 2(a). The two points are on a same triangle. The tetrahedron is refined into three as shown in Figure 4. We note that, there is a choice for linking one of the two new points to a vertex of the original tetrahedron on the same face triangle.

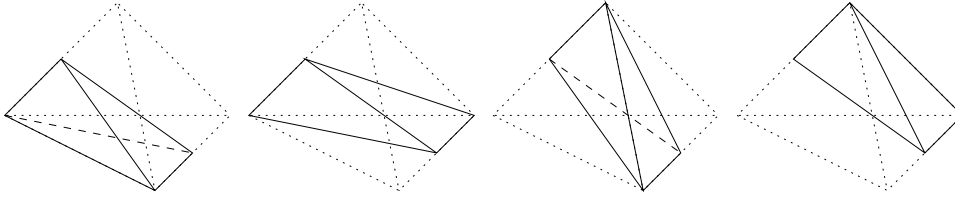


Figure 5: Case 2(b): Two mid-edge points. Refined into four subtetrahedra.

Case 2(b). The two points are on two different triangles. The tetrahedron is refined into four as shown in Figure 5. We note that, there is no choice for the refinement as all four face triangles are each refined into two subtriangles.

Case 3. Three points are added to a tetrahedron. There are three sub-cases.

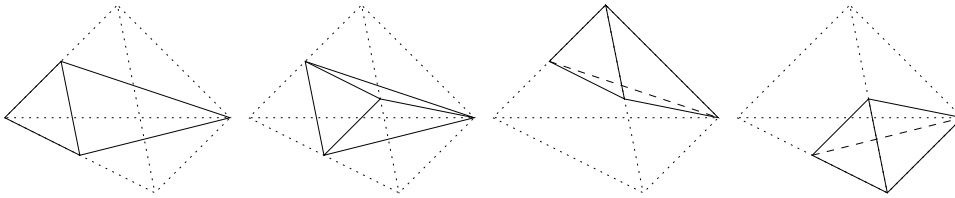


Figure 6: Case 3(a): Three mid-edge points. Refined into four subtetrahedra.

Case 3(a). The three points are on a same triangle. The tetrahedron is refined into four as shown in Figure 6. There is no other choice.

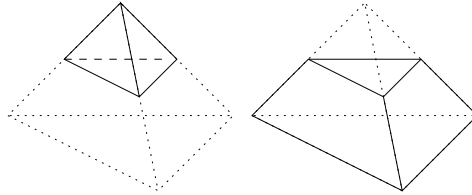


Figure 7: Case 3(b): Three points. The bottom “wedge” is refined in Figures 17–18.

Case 3(b). The three points are around a same vertex. Shown in Figure 7, there is no choice in determine the top subtetrahedron. However, three lines (see Figure 2) will be added to refine the bottom “wedge” so that the wedge would be refined either into three or eight subtetrahedra (to be determined in Figures 17–18).

Case 3(c). The three points are not around a same vertex, neither on a same face triangle. Shown in Figure 8, such a tetrahedron has two face triangles of the type shown in Figure 2.

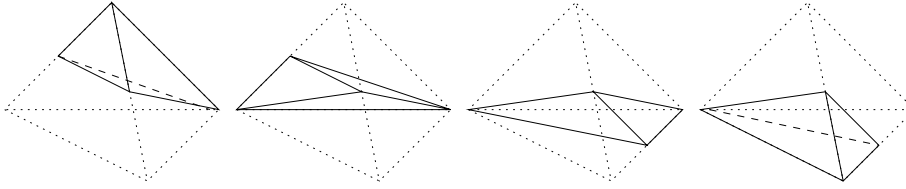


Figure 8: Case 3(c)(1). Three mid-edge points. Refined into four subtetrahedra.

Case 3(c)(1). If we do not introduce any internal edge in the subdivision, there is only one way to do the refinement, as shown in Figure 8. Even in this case, without mirror symmetry, there would be two configurations. Without rotation symmetry, there would be 24 configurations. In this subdivision, however, there is no choice on the two face triangles in subdividing the quadrilateral shown in Figure 2. If two neighboring tetrahedra have this type of subdivision, the interface subtriangles would not match.

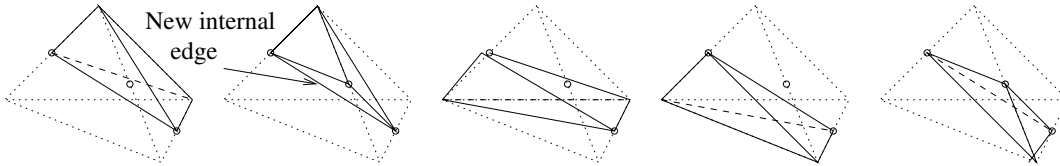




Figure 9: Case 3(c)(2). 3 mid-edge points. Refined into five subtetrahedra.

Case 3(c)(2). Shown in 9. Once we introduce an internal edge, cf. Figure 9, we can cut the sub-quadrilateral (see Figure 2) either way, i.e.  or . So there would be 4 configuration. To illustrate, we only show the two ways for the subdivision of the front triangle in Figure 9 and Figure 10.

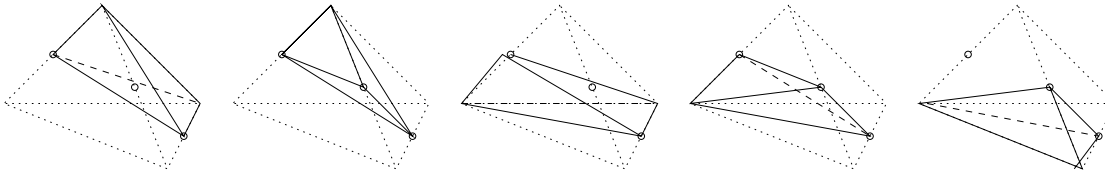


Figure 10: Case 3(c)(3). 3 mid-edge points. Refined into five subtetrahedra.

Case 3(c)(3). Shown in 10. In general, the grid generated by the subdivision of Case 3(c)(1) would be better than those generated by Case 3(c)(2) and Case 3(c)(3). But the latter methods have no restriction on subdividing the face triangles and would not generate any conflict in interface matching.

Case 4. Four points are added to a tetrahedron. There are two sub-cases.

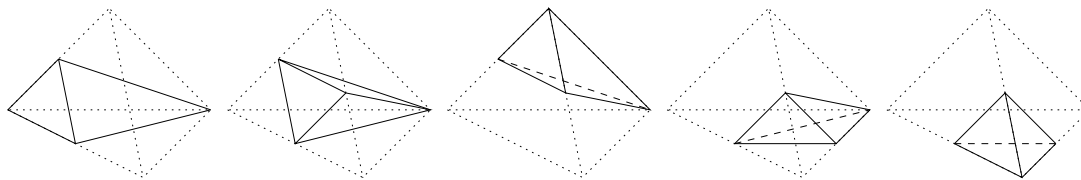


Figure 11: Case 4(a)(1): Four mid-edge points. Refined into five subtetrahedra.

Case 4(a). There must be three points on a same triangle. The fourth point is the midpoint of an edge of two triangles other than the last triangle. These two triangles have the fourth new, mid-edge point, and have the situation depicted in Figure 2. We have two choices for each triangle. There are 4 sub-cases. By mirror symmetry, there are three cases actually, Case 4(a)(1) (Figure 11), Case 4(a)(2) (Figure 12), and Case 4(a)(3) (Figure 13). If we have truly choices among them, we will use Case 4(a)(1) as the subtetrahedra have better shapes since the tetrahedron is refined into 5 in this case, but 6 in the other two cases.

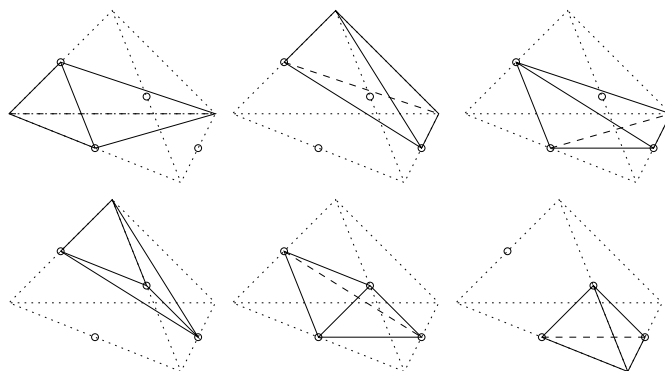


Figure 12: Case 4(a)(2): Four mid-edge points. Refined into six subtetrahedra.

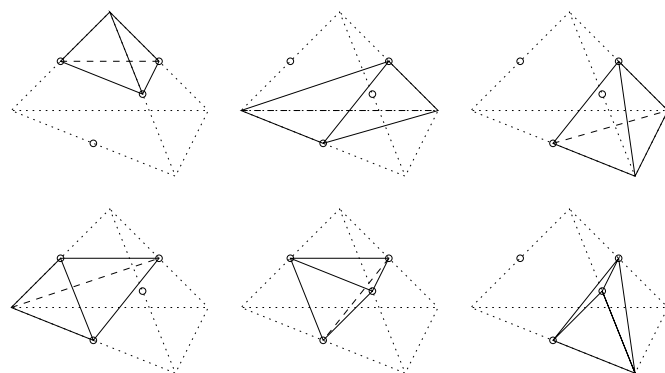


Figure 13: Case 4(a)(3): Four mid-edge points. Refined into six subtetrahedra.

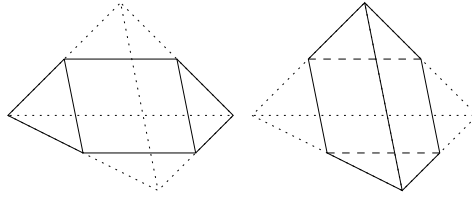


Figure 14: Case 4(b). Four points. The two “wedges” are refined in Figures 17–18.

Case 4(b). Each face triangle has exactly two new points. There are four choices in adding new lines to each of four face triangles so that each face triangle is refined into three, depicted in Figures 2. In other words, we need to subdivide two “wedges” shown in Figure 14. As mentioned above, we will delay the decision on refining the two “wedges” (see Figures 17–18) as the face triangle subdivision must be matched on the two sides by choosing between \triangleleft and \triangleright . So the tetrahedron might be refined into 6, 12, or 18 subtetrahedra, depending on the refinements of the two “wedges”.

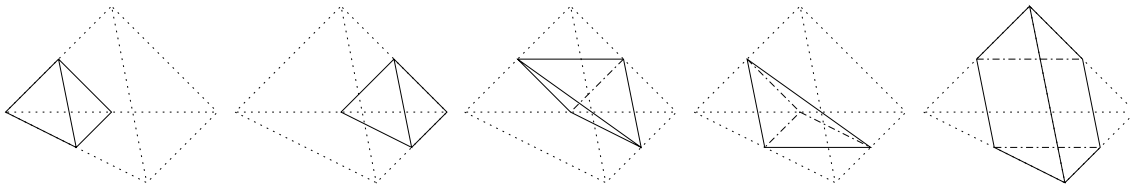


Figure 15: Case 5. Five mid-edge points. There is a choice in determining the third and fourth subtetrahedra. The ‘wedge’ is refined in Figures 17–18.

Case 5. Five points are added to a tetrahedron. There is only one case. Four points would divide the tetrahedron into two parts, as in the Case 4(b) above. There is a new point on one of the two parts. This part is to be refined into 4 subtetrahedra with a choice among the two internal edges, shown in Figure 15. Again the refinement of the other part, a “wedge” is to be refined in Figures 17–18.

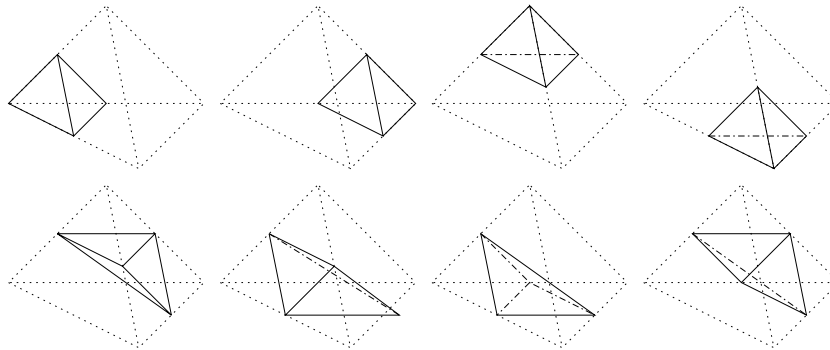


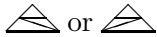
Figure 16: Case 6. Six mid-edge points. Refined into 8 subtetrahedra.

Case 6. Six points are added to a tetrahedron. There is only one case. This is the standard

multigrid refinement, to produce eight subtetrahedra, see Figure 16. There is a new internal edge introduced in the refinement. There are three choices in determining this new internal edge. One may randomly pick one. Or one may use the ordered refinement so that the eight subtetrahedra are similar to the father tetrahedron for the self-reproductive tetrahedra (see [10]); Or one may use the short internal edge among the three under the Hessian metric in use.

The face triangle refinements show in Figure 2 appear in Cases 2(a), 3(b), 4(b), 5, 3(c) and 4(a) above. However, in the last two cases, 3(c) and 4(a), there is no choice as the other alternative in 2 is ruled out by the geometry. For Case 2(a), either diagonal line on the quadrilateral of the face triangle can be used. We then have to study the refinement of “wedges” appearing in Cases 3(b), 4(b) and 5.

3 Refinement of wedges and nested tetrahedral refinements

Given a tetrahedral grid and some mid-edge points, can we find a nested refinement without introducing any new grid points? The answer is yes, and will be shown in a theorem below. Before answering this question, let us consider the subdivision of interface triangles again. From last section, it is clear that a face triangle is uniquely refined except the case of two new mid-edge points, shown in Figure 2, i.e., . Such an interface appears on tetrahedra of Cases 2(a), 3(c), 4(a), 5, 3(b) and 4(b). Among the cases, given, or randomly generated, any one of the two face subdivisions shown in Figure 2, the tetrahedron of Cases 2(a), 3(c), 4(a) and 5 can be refined. Therefore, the refinement question is simply reduced to the subdivision of the “wedges” appearing in Cases 3(b) and 4(b).

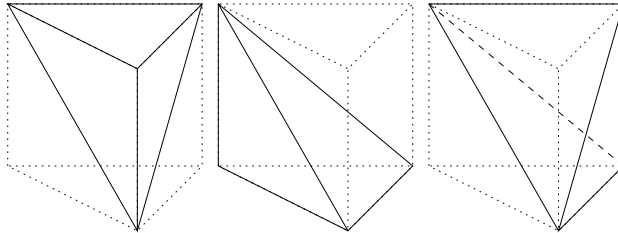


Figure 17: An “wedge” is subdivided into three subtetrahedra.

By a “wedge”, we mean a convex polyhedron show in Figure 17 which consists two opposite triangles connected by three quadrilaterals. For a wedge, to be refined into tetrahedra without adding any new point, we need to add three diagonal lines to the three face quadrilaterals. There are two situations: (A) there are two diagonals linked to one vertex (see Figure 17), and (B) all three diagonals are disjoint (see Figure 18).

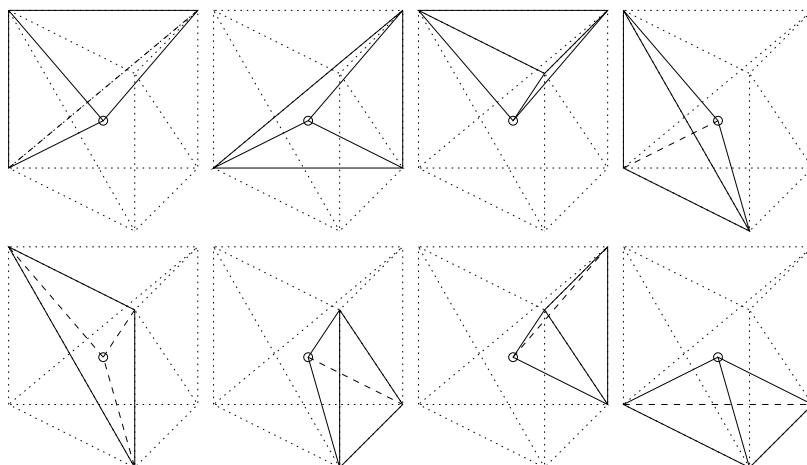


Figure 18: With an internal point, an “wedge” is subdivided into eight subtetrahedra.

For the situation (A), the wedge is refined into three tetrahedra. But for the situation (B), the wedge cannot be subdivided into tetrahedra without using any new point, (see Figures 19 and 18). We then have to introduce an internal point in the latter case (see Figure 18). This is shown in next two lemmas.

Lemma 3.1 *Without adding any new point, the wedge shown in Figure 19 cannot be subdivided into tetrahedra.*

Proof Since we do not add any new point, the bottom triangle ABC in Figure 19 must be a face triangle of a tetrahedron after subdivision. This tetrahedron must be made by the points A, B and C , plus one of points D, E and F . If the tetrahedron is formed by $ABCD$, then a new edge CD is added to the face of the wedge. It would create a new point, the intersection of two diagonals, AF and CD , of a convex, planar quadrilateral $ACFD$. This contradicts to the assumption of no new point. The lemma is proven by symmetry. ■

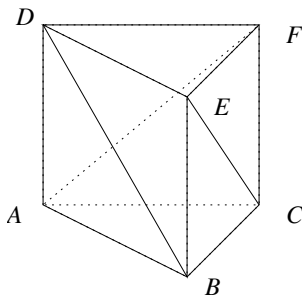


Figure 19: The “wedge” cannot be subdivided into subtetrahedra without a new point.

Lemma 3.2 *By adding one new point anywhere in the interior of wedges defined in Cases 3(b), 4(b) and 5 in Section 2, such a wedge can be subdivided into eight tetrahedra as shown in Figure 18.*

Proof There are two types of wedges in Cases 3(b), 4(b) and 5 in Section 2. The latter two are of the same type. Since the wedges are convex, any internal point can make a non-degenerate tetrahedron with any face triangle, shown in Figure 18. ■

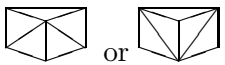
Let us repeat our question at the beginning of the section. Can we avoid the wedges shown in Figure 19 completely? If so, we can refine a tetrahedral grid with any number of mid-edge points added without using any additional grid point. This is shown in the next theorem.

Theorem 3.1 *Given a tetrahedral grid and any number of mid-edge points, there is at least one nested refinement.*

Proof As listed earlier, as long as all tetrahedra of Cases 3(b) and 4(b) can be refined, all tetrahedra would be refined with interface matching. Therefore, we show the theorem by proving the existence of all Cases 3(b) and 4(b) tetrahedra before any refinement of other type of tetrahedra. This is done by the principle of mathematical induction. Let us assume there are totally m Cases 3(b) and 4(b) tetrahedra initially.

- $m = 1$. If it is a Case 3(b) tetrahedron, we randomly select two diagonal edges and select the last face edge of the “wedge” (Figure 7) of the tetrahedron according to the wedge subdivision of Figure 17. In fact, there are 8 configuration of face edges, 6 of Figure 17 and only 2 of Figure 18. If this only tetrahedron is of Case 4(b), we can randomly pick three diagonal edges on the face of the tetrahedron (Figure 14). Next we select the internal edge so that it would lead a type Figure 17 wedge when combined with the two face diagonal edges chosen earlier. Finally, we select the last face diagonal to make the second wedge of type Figure 17.
- $m \leq n - 1$. We assume all m tetrahedra of Cases 3(b) and 4(b) can be refined with faces matched.
- $m = n$. Let us call Ω_1 the union of all m tetrahedra (closed) of Cases 3(b) and 4(b). If the boundary of Ω_1 has 1 (or more) face triangle which has two mid-edge points, i.e., \triangle face triangle. We remove the tetrahedron (T_0) which has this triangle as one face. Then $\Omega_1 \setminus T_0$ consists of $n - 1$ tetrahedra of of Cases 3(b) and 4(b), and it can be refined nested by the induction assumption. The refinement of $\Omega_1 \setminus T_0$ determines the refinement of other three faces of T_0 . We then select a diagonal line between \triangle and \triangle , so that the wedge on the outside of T_0 is a type Figure 17. This can be done as analyzed in the $m = 1$ case.

If Ω_1 has no \triangle boundary triangles, i.e., there is no mid-edge point on the boundary of Ω_1 . This happens only when all outside layer tetrahedra of Ω_1 are of type Case 3(b). In this case, all boundary triangles of Ω_1 are the bottom triangles of Case 3(b) tetrahedra shown in Figure 7. Let us pick up any one boundary vertex of Ω_1 . Let the tetrahedra around the vertex be numbered as T_1, T_2, \dots, T_k , counter-clockwise when viewed from outside Ω_1 . Let $\Omega_2 = \Omega_1 \setminus \cup T_i$. Then Ω_2 consists less than n tetrahedra of Cases 3(b) and 4(b). By the induction assumption, Ω_2 has a nested refinement without any new point. One face triangle of tetrahedron T_1 is determined by the refinement of Ω_2 , either \triangle or \triangle . We pick up the subdivision of the triangle between T_1 and the last tetrahedron T_k so that the wedge of T_1 is

like  (see Figure 7). Then T_1 has a type Figure 17 wedge no matter how the last face (\triangle) quadrilateral is refined. This last face (\triangle) is shared with next tetrahedron T_2 . Therefore, we can repeat the above step, i.e., choose a diagonal on the interface of T_1 and T_2 , so that this diagonal and the diagonal on the interface between T_2 and Ω_2 make the

configuration of Figure 17, i.e., . Repeating k steps, all k tetrahedra, $T_1, T_2, \dots,$ and T_k , are nested refined consistently. ■

Definition 3.1 (Nested refinement with given mid-edge points.) One nested refinement of a tetrahedral grid consists of the following 3 steps.

1. Determine the edges to be refined and introduce a midpoint to each of these edges. The midpoint may be calculated by the Euclidean metric or by a Hessian metric.
2. Refine sequentially all tetrahedra of types of Cases 3(b) and 4(b), where if a face triangle \triangle of a tetrahedron has a choice, we always choose \triangle . Sometimes the third face triangle of a Cases 3(b) tetrahedron, or the fourth face triangle of a Cases 4(b) tetrahedron, may have

choices too, to have sub-wedges of type  or . Two cases may happen.

- (a). All tetrahedra of types of Cases 3(b) and 4(b) are refined where the wedges are refined as in Figure 17.
- (b). After refining some tetrahedra, due to the interface compatibility, we may have no choice for a type Cases 3(b) tetrahedron whose three \triangle faces are subdivided earlier, or for a type Cases 3(b) tetrahedron whose four \triangle faces are subdivided at neighboring tetrahedra. Such a tetrahedron may have a sub-wedge of type Figure 18 which could not be refined without adding any new point other than the mid-edge points given in Step 1. In this, we back up the refinement until the latest choice of \triangle . We change the choice there to \triangle and repeat the refinement. After at most $2^k - 1$ backups, all k tetrahedra of types of Cases 3(b) and 4(b) are nested refined. Here k is the number of 3(b) and 4(b) tetrahedra defined by the mid-edge points in Step 1.

3. Refine all rest tetrahedra.

■

In the step 2(b) of Definition 3.1, in theory, it is very costing to back up $2^k - 1$ times, as each backup would waste in average the refinements of $k/2$ tetrahedra done earlier. Computationally, the backup is very rare. However, the implementation of algorithm and the storage management are complicated due to this recursive backup. To avoid this, we can allow adding one internal point for each tetrahedron of 2(b) in the Definition 3.1. This is lead to a simpler algorithm.

Definition 3.2 (Nested refinement with possible new internal points.) The algorithm is the same as that defined in Definition 3.1 except 2(b), which is replaced by

- 2(b). If all subdivisions of face-triangles of a Case 3(b) or Case 4(b) tetrahedron is determined by earlier subdivision, we refine the tetrahedron so that the sub-wedges are refined either without new point (Figure 17), or with a new internal point (Figure 18).

■

Theorem 3.2 *After one refinement of a conformal finite element grid defined by Definition 3.2, a nested conformal grid is generated.*

Proof It is evident that after one such refinement all interface triangles are shared by the tetrahedra on the two sides. Therefore, the intersection of any two subtetrahedra (closed) would be either empty, or a common vertex, or a common edge, or a common face triangle, of the two tetrahedra.

■

4 Computational test

We test the newly proposed algorithm of Definition 3.1 by refining a cube according to the Hessian metric of the following three functions (with proper constants ϵ and orientations):

$$f_1(x, y, z) = e^{\frac{x}{\epsilon}}, \quad f_2(x, y, z) = e^{\frac{x+y}{\epsilon}}, \quad f_3(x, y, z) = e^{\frac{x+y+z}{\epsilon}}.$$

The metric used in refinement is (with a small constant α)

$$H_i = \alpha \begin{pmatrix} 1 & & \\ & 1 & \\ & & 1 \end{pmatrix} + \nabla^2 f_i, \quad i = 1, 2, 3.$$

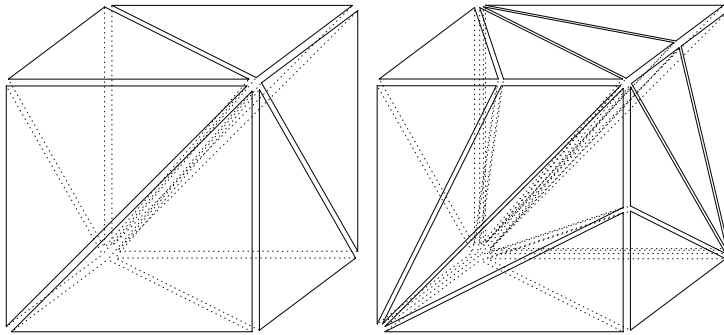


Figure 20: The initial grid and a refinement (high refinement near the upper front corner).

We show some grids or the surface grids. In Figure 20, on the left we have the initial grid. In Figure 20, on the right, we decided to refine the grid at the upper front corner and three points are added on the three edges of the corner. When we continue the refinement in this fashion, we would get very thin, or very long tetrahedra.

With the Hessian metric from $f_1(x, y, z)$, we need to refine grids in one direction, near one face. This leads to grids such as the left one shown in Figure 21, where the top elements form layers of “thin bricks”. For $f_2(x, y, z)$, we need to refine grids in two directions, near one edge. This leads to grids shown as the middle graph in Figure 21, where the elements near the upper right edge form “long sticks”. Finally, for $f_3(x, y, z)$, we need to refine grids in all three directions, but only near one corner. Shown by the right graph in Figure 21, we can see the elements near the upper-right-front corner are much smaller than the elements near the opposite corner.

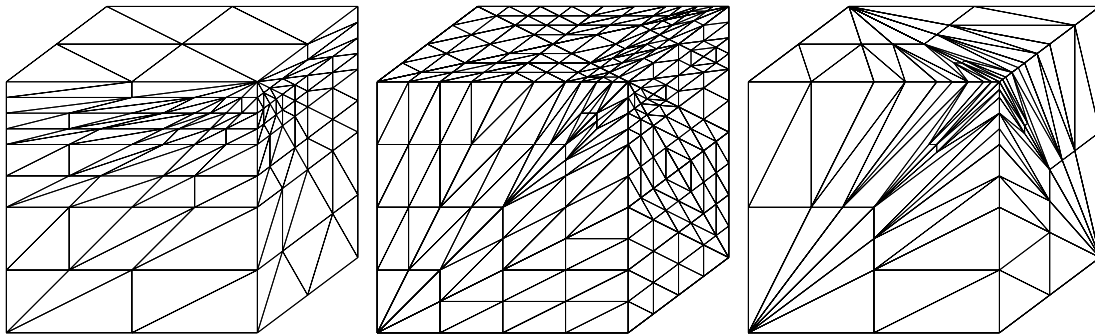


Figure 21: High refinement near a point, in 1, 2 or 3 directions.

References

- [1] T. Apel, S. Nicaise and J. Schoberl, A non-conforming finite element method with anisotropic mesh grading for the Stokes problem in domains with edges, *IMA J. Num. Anal.* 21 (2001) 843–856.
- [2] T. Apel and J. Schoberl, Multigrid methods for anisotropic edge refinement, *SIAM J. Num. Anal.* 40 (2002) 1993–2006.
- [3] T. Apel, S. Grosman, P. K. Jimack and A. Meyer, A new methodology for anisotropic mesh refinement based upon error gradients, *Appl. Num. Math.* 50 (2004) 329–341.
- [4] M. Berzins, A solution-based triangular and tetrahedral mesh quality indicator, *SIAM J. Sci. Compt.* 19 (1998) 2051–2060.
- [5] L. Chen and J. Xu, Optimal Delaunay triangulations. *J. Comput. Math.* 22 (2004) 299–308.
- [6] L. Chen, P. Sun and J. Xu, Optimal anisotropic meshes for minimizing interpolation errors in L^p -norm. *Math. Comp.* (2005), Articles in press, on line.
- [7] P. G. Ciarlet, *The finite element method for elliptic problems*, North-Holland, Amsterdam, 1978.
- [8] W. Huang, Mathematical principles of anisotropic mesh adaptation, *Commun. Comput. Phys.* 1 (2006) 276–310.
- [9] A. Plaza and G. F. Carey, Local refinement of simplicial grids based on the skeleton, *Appl. Num. Math.* 32 (2000) 195–218.
- [10] S. Zhang, Successive subdivisions of tetrahedra and multigrid methods on tetrahedral meshes. *Houston J. of Math.* 21 (1995) 541–556.