Strategies for Generating Well Centered Tetrahedral Meshes on Industrial Geometries

Sean Walton, Oubay Hassan, and Kenneth Morgan

Abstract This paper outlines some recent developments in the process of generating well centered tetrahedral meshes. A well centered tetrahedron contains its circumcentre, which is a basic property required for a valid co-volume discretisation. Although most work in this area has focussed on improving meshes generated using classical techniques, in this paper we consider modification of the generation procedure itself. A simple lattice point insertion technique is introduced and the potential of the technique for generating well centered meshes is demonstrated. This is accomplished by comparing, for some complex geometries, the meshes generated with the meshes created by a standard Delaunay mesh refinement technique. Despite the simplicity of the lattice point insertion method, the comparison is found to be favourable and the method is shown to produce good well centered elements in the vicinity of the geometry.

1 Introduction

Co-volume algorithms, such as the marker and cell (MAC) algorithm for the solution of the Navier Stokes equations [1] or the Yee scheme for the solution of Maxwell's equations [2], exhibit a high degree of computational efficiency in terms of their low operation count and their low storage requirements. These properties, together with the simplicity of the algorithm, have made the Yee scheme a favoured computational solution technique for industrial electromagnetic simulations. However, these algorithms are highly sensitive to the quality of the mesh employed. Specifically, they require well centered meshes, i.e. meshes in which each element contains its circumcentre.

The primary goal of this work is to develop a technique which automatically generates a well centered unstructured mesh around an arbitrary geometry in three dimensions. It has been shown that it is possible for uniform meshes, and with enough optimisation, for non-uniform meshes, to obtain well centered meshes for

S. Walton • O. Hassan (✉) • K. Morgan
College of Engineering, Swansea University, Swansea SA2 8PP, Wales, UK
e-mail: s.p.walton@swansea.ac.uk

S. Perotto, L. Formaggia (eds.), *New Challenges in Grid Generation and Adaptivity for Scientific Computing*, SEMA SIMAI Springer Series 5,
DOI 10.1007/978-3-319-06053-8_8

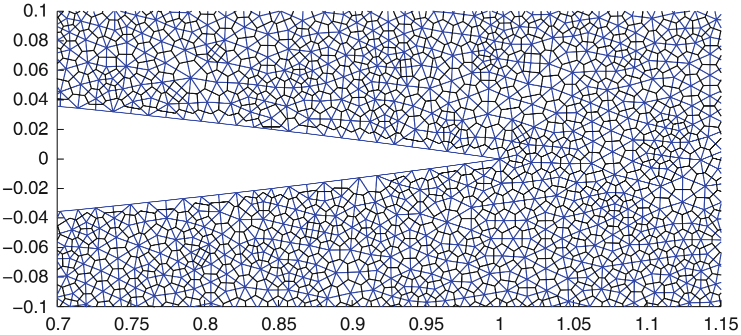

Fig. 1 A detail of an example well centered mesh in two dimensions

general two dimensional geometries [3–5]. An example of such a mesh is shown in Fig. 1. This mesh was generated using a Delaunay refinement technique and was optimised using a modified cuckoo search [4–6]. Our work is currently focussed on attempting to obtain the equivalent level of mesh quality in three dimensions.

This paper details some recent developments aimed at addressing the problem of generating well centered meshes for industrial geometries. At present, the problem remains unsolved, but the results presented here demonstrate the recent progress that has been achieved. The paper is structured in the following way. The mesh quality required for successful implementation of co-volume algorithms is discussed. Volume mesh generation that is accomplished by extending the stitching methods that was successfully implemented in two dimensions [3] is described. Near field meshes generated by the Delaunay insertion of ideal lattice is highlighted. Previously developed mesh optimisation techniques are then briefly outlined and applied to the meshes initially generated using the lattice insertion technique. The quality of the meshes generated using the proposed technique for three representative geometries, consisting of an ONERA M6 wing, a full B60 geometry and a full F16 geometry, is illustrated. The paper concludes with a discussion and suggestions for further work.

1.1 Co-volume Algorithms and Their Mesh Requirements

In its most basic form, a classical co-volume method is staggered in time and is implemented on a pair of orthogonal staggered Cartesian meshes in space. The mutually orthogonal pair of meshes are termed the primal and the dual mesh. The resulting discretisation is second order accurate, in both space and time, on uniform meshes and has the additional advantages of preserving the energy, and maintaining the amplitude, of plane waves. This enables the approximation of electromagnetic fields near sharp edges, vertices and wire structures, without a need for excessive local mesh refinement [7].

The success of a co-volume solver is, however, particularly sensitive to the quality of the meshes employed [4]. On general meshes, second order accuracy of a co-volume scheme can only be ensured if [8]:

- The primal and dual meshes are mutually orthogonal, i.e. the edges of the primal mesh pass orthogonally through the faces of the dual mesh.
- A dual mesh node must lie inside its corresponding primal mesh element, otherwise integrals over primal mesh elements will be approximated in terms of solution values located outside the elements.
- The nodes of the dual mesh lie at the centroid of corresponding primal mesh and the dual mesh edges pass through the centroids of the corresponding primal mesh faces.

The most obvious choice, for meshes with these properties, is to employ a primal Delaunay mesh and its Voronoi dual [8]. This is advantageous because the first condition above is then met automatically. The second and third conditions are, however, not met for a general set of nodes. When all the conditions are met, the set of nodes is said to form a centroidal Voronoi tesselation (CVT). The aim of the current work is to develop new mesh generation and optimisation techniques, which will make the use of co-volume techniques feasible for complex three dimensional geometries.

In isolation, a tetrahedral element whose faces are equilateral triangles has a Voronoi node which sits at its centroid. Note, however, that such tetrahedra do not fill space. To date, only one type of tetrahedron has been found which fills space and satisfies all of the above conditions [9–11]. Each face of this tetrahedron is an isosceles triangle, with one side of length 1 and the other two, shorter, sides of length $\sqrt{3}/2$. The tiling of space which is created by these tetrahedra is illustrated in Fig. 2, where each internal node is connected to 14 neighbouring nodes in a regular lattice. Clearly, a tiling of this form will be unable to mesh the volume surrounding a general surface. In addition, in the analysis of practical problems, it is normally desirable that the local mesh size matches a user-defined target mesh spacing at each point in space. It has proved possible to obtain well centered volume meshes, for regions lying inside simple shapes, by using optimisation techniques [4, 12, 13]. However, it is yet to be shown that this can also be achieved for general geometries with specific mesh spacing functions.

Since the use of the Delaunay Voronoi dual ensures mutual orthogonality, our current work is focused on producing well centered meshes. Much of the current literature, in this field, employs traditional meshing techniques and attempts to improve the mesh through the use of a variety of mesh cosmetics procedures. Cosmetic procedures are when either the positions, connectivity of the mesh nodes are adjusted after the initial generation, in an attempt to improve quality. Although it is not technically the same as the problem of interest here, the problem of calculating a CVT of a set of nodes is particularly relevant. CVT techniques aim to minimise an energy function [14], which is related to the deviation of the primal Delaunay nodes from the mass centroid of the Voronoi cell in which they lie. Such methods, which are iterative by nature, require that the CVT energy is calculated several times and

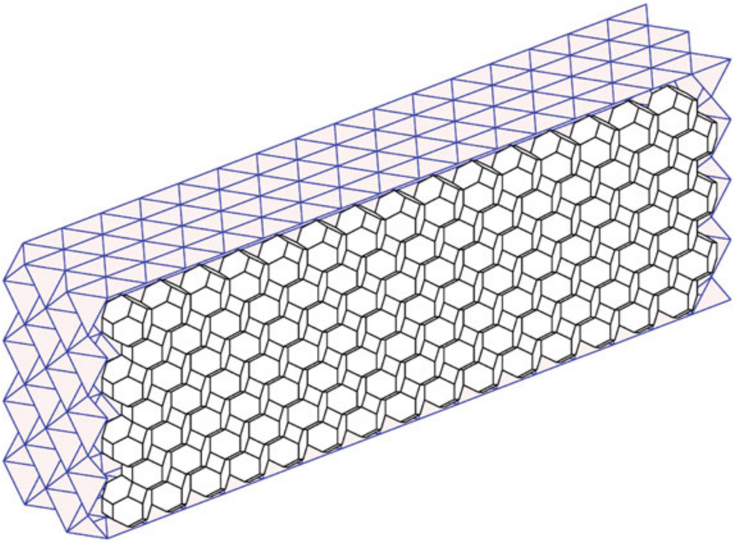

Fig. 2 Detail of a mesh of ideal tetrahedral elements, showing the surface Delaunay faces and the internal Voronoi cells

this can be an expensive task, when it is applied to large meshes of practical interest. It is apparent that it is possible to find a CVT for a range of examples, but even state of the art techniques, using quasi-Newton based optimisation methods, require that the CVT energy be calculated hundreds of times for simple geometries [15].

During our own efforts at directly optimising meshes for this type of application we introduced the concept of reduced order mesh optimisation [4, 5]. By applying reduced order modelling techniques we were able to represent full meshes, with a large number of nodes, with much fewer degrees of freedom. The reduced number of degrees of freedom allowed the application of gradient free optimisation techniques. To date this approach has only been explored once so more work in this area is needed, to ascertain its strengths and weaknesses. However, our work still showed that a large number of iterations is required to eliminate all non well centered elements [4, 5]. For this reason, it is important to develop mesh generation techniques that aim, at the outset, to create a high quality mesh. If this can be achieved, it will, hopefully, ensure a reduction in the time spent on cosmetics.

2 Volume Mesh Generation Techniques

Two dimensional surface meshing is now well established. For example, using advancing front techniques, close to ideal surface meshes can be obtained for a wide range of geometries, provided, of course, that the geometry definitions are water-tight. Therefore, it will be assumed that high quality surface meshes can be obtained for the geometries of interest. However, it might be necessary, at some later stage, to consider introducing additional specific surface mesh generation techniques, if we find that existing surface meshes prevent us from obtaining the volume meshes of the desired quality.

The challenge of generating well centered tetrahedral meshes for complex geometries has previously been approached using a stitching method [3]. This method starts by employing an advancing front technique [16] to generate a local high quality triangulation close to the boundaries. This triangulation is then stitched to an ideal mesh which covers the remainder of the computational domain. The advancing front technique places nodes at ideal locations and generates the elements in layers, while advancing away from the boundary surfaces. Although this approach is effective and relatively simple in two dimensions, its extension to three dimensions is clearly dependent on the ability to efficiently create high quality triangulation in the vicinity of the geometry.

Here we consider an alternative mesh generation approach that combine the classical Delaunay refinement method with an ideal mesh lattice for the creation of high quality near field mesh.

2.1 Delaunay Refinement

The Delaunay refinement method [17] requires a surface mesh defining the boundary and a user-defined mesh spacing function. Initially, the surface mesh nodes are inserted into a convex hull which contains the entire computational domain. Once this is completed, the boundary triangulation of the surface is recovered, resulting in a Delaunay triangulation of these nodes. New nodes are then inserted sequentially at the centroids of existing elements that do not comply with the required mesh spacing, with a new Delaunay mesh of the nodes being produced at each stage. Since we are looping over the elements in the mesh, we immediately know the element into which the node should be inserted. Furthermore, since the insertion of new nodes takes place inside existing elements, there is no need to check for self intersection of the primal meshes faces and edges. This results in a very fast algorithm. This process of centroid insertion continues until the node density is within a user defined-factor of the local value specified by the mesh spacing function [18].

2.2 Lattice Point Insertion

To ensure the creation of well centered meshes, we start by considering the ideal mesh shown in Fig. 2. Here, each node in this lattice is connected to 14 neighbouring nodes, as shown in Fig. 3. Each tetrahedron is well centered when these nodes are triangulated in a Delaunay fashion. The method that is proposed for generating

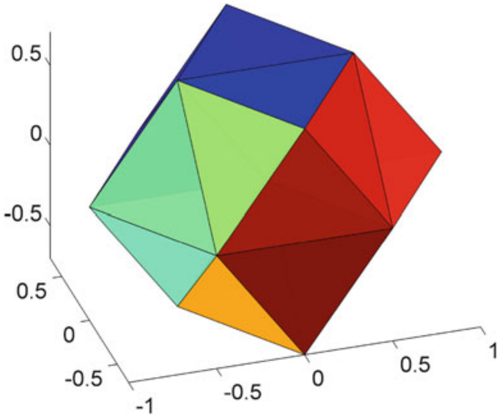

Fig. 3 A single point from the lattice of Fig. 2, with its 14 neighbours lying either on the sphere of radius $\sqrt{3}/2$ or on the sphere of radius $1/\sqrt{2}$

volume meshes is then to use these 14 nodes as a template for locating nodes in space. This approach, which requires a surface mesh and a defined spacing function, may then be simply described by the following steps:

1. For each node on the surface:
   (a) Calculate the value δ of mesh spacing function at the node.
   (b) Generate the co-ordinates of the 14 lattice nodes described above, such that the long edges of the isosceles triangular faces are of length δ.
   (c) Check each of the 14 nodes in turn; store the nodes lying inside the computational domain in a list, $\mathbf{x}_{temp}$.
2. When $\mathbf{x}_{temp}$ is generated, loop over each node currently in the volume mesh; delete any point in $\mathbf{x}_{temp}$ that lies within 0.5δ of an existing point.
3. Now loop over the remaining nodes in $\mathbf{x}_{temp}$ and, for each point:
   (a) Calculate the spacing function, δ^i, at the point $\mathbf{x}^i_{temp}$.
   (b) Perform a search for nodes either already in the mesh or in the list which lie within a radius δ^i of $\mathbf{x}^i_{temp}$.
   (c) Calculate the spacing, δ^j, at each of the nodes found by this search.
   (d) If any point lies within $0.5max(\delta^i, \delta^j)$ of $\mathbf{x}^i_{temp}$ then delete $\mathbf{x}^i_{temp}$.
4. Insert the remaining nodes in $\mathbf{x}_{temp}$ into the mesh in a Delaunay fashion. During this insertion, a check is made to ensure that the minimum distance between a new point and the cavity into which it is inserted is greater than the local spacing. If this is not the case, the point is rejected. This step is the main difference between the lattice technique considered here and others [19].

This process is then repeated for the desired number of layers using the last set of nodes $\mathbf{x}_{temp}$ as the starting surface or until all the nodes are rejected in the final step. Connecting the nodes in a Delaunay fashion rather than specifying a connectivity is justified since it is known that, if a well centered triangulation of a set of nodes exists it is unique and it is the Delaunay triangulation [20].

At first glance, it may appear that this algorithm will be expensive, due to the nearest neighbour searching. However, many data structures have been developed for this operation and, in this implementation, a KD-tree library is employed. This enables searches for neighbours within a given range to be performed in $O(\log N)$ operations, where N denotes the total number of nodes [21]. Other costs associated with point insertion and background space calculation can be reduced by employing careful data structure and storage. For instance when building the mesh we store the elements connected to each node, this allows a good first guess for which element the new lattice nodes originating from that node lie inside.

2.3 Examples

It is recognised that meshing techniques can be highly sensitive to the nature of the geometry under consideration. Several three dimensional examples, are considered to illustrate the potential of the proposed methodology. The examples that have been selected are of particular interest to the aerospace community. Meshes are generated using both lattice insertion and the original Delaunay refinement. No mesh cosmetic steps are performed on the meshes presented in this section and the target mesh spacing function is the same for both generation techniques. Since the extension of the stitching method requires the generation of near field elements that exhibit the required qualities for co-volume methods, at this stage, only elements that are attached to boundary faces and boundary nodes are considered in the analysis. In the following, the term bad element is meant to refer to an element whose circumcentre lies outside the element.

The first example is the problem of generating a volume mesh, for the ONERA M6 wing. The surface mesh, shown in Fig. 4 consists of 16,178 triangular elements and 8,091 nodes. Table 1 details the mesh statistics for the two meshes generated for

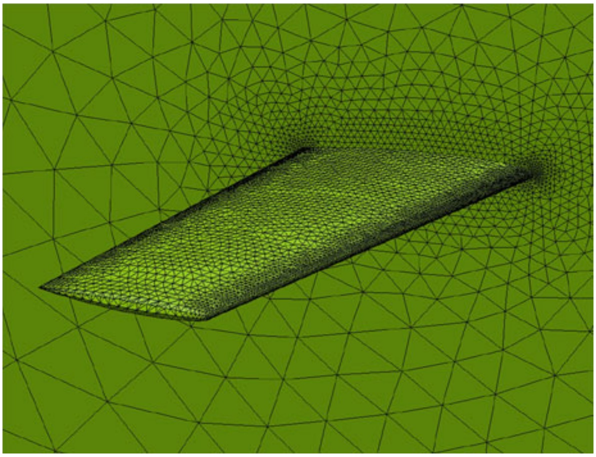

Fig. 4 ONERA M6 wing: detail of the surface mesh

Table 1 Comparison of mesh quality measures for the ONERA M6 wing

	Delaunay refinement	Lattice point insertion
No. nodes	93,066	79,591
No. elements	572,711	485,324
No. bad elements touching a surface	24,637	15,157
No. (%) surface elements connected to a bad element	2,309 (14 %)	318 (2 %)

this geometry. It is clear that a considerable reduction in the number of bad elements has been achieved. The percentage of surface elements which are connected to bad elements is very low, at 2 % when using lattice point insertion and at 14 % when using Delaunay refinement. This is due to the fact that the proposed technique was able to better represent the spacing function. Figure 5a shows a view of the volume mesh generated using the Delaunay refinement technique. The elements shown in black are the bad elements. There also appears to be a high concentration of bad elements at the leading edge of the wing, where the curvature is the highest. Figure 5b shows a view, at the same location, of the volume mesh generated using the lattice insertion technique. The bad elements are again shaded black, while a number of elements on the surface of the wing are well structured and are almost ideal. This mesh for the ONERA M6 geometry appears to be more structured and the effect of the lattice insertion is quite clear.

The second example is that of a B60 full aircraft configuration. the surface mesh consists of 144,240 triangular elements and 72,122 nodes. A view of the surface mesh is shown in Fig. 6. The corresponding mesh statistics for the B60 geometry are presented in Table 2. For this configuration, significantly fewer nodes are inserted using the lattice insertion method. As with the previous ONERA M6 wing example, the total percentage of bad elements is similar for both generation techniques, but the percentage of bad elements connected to the surface is much less when lattice insertion is employed. Figure 7a, b show similar views of the meshes created using Delaunay refinement and lattice point insertion respectively. As reflected by the statistics, in Fig. 7b it is apparent that there are areas of the mesh, attached to the boundaries, which almost match the ideal mesh. This can be explained by the fact that, the Delaunay insertion of the lattice nodes results in the ideal that satisfy the mesh quality requirements. There is the obvious question of how the constant orientation meets with the surfaces. It may be beneficial to adjust the orientation of the inserted lattice nodes at the surface. Alternatively, it may be possible to move the nodes on the surface, so as to produce a better match with the orientation of the lattice. These are possible modifications that we are currently considering.

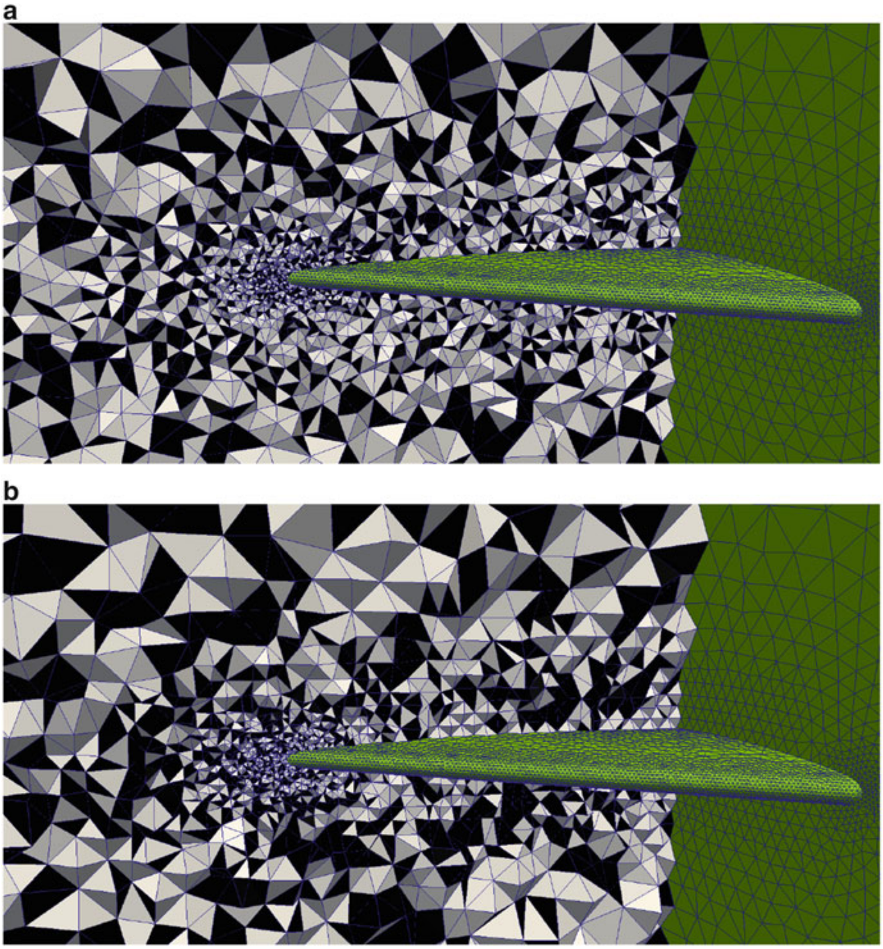

Fig. 5 ONERA M6 wing: detail of the volume mesh generated using (**a**) Delaunay refinement; (**b**) lattice point insertion

The final example is a complete generic F16 configuration that provides a geometry that is challenging to mesh. The surface mesh, shown in Fig. 8 consists of 291,378 triangular element and 145,699 nodes.The mesh statistics that are obtained for this example are given in Table 3. Despite the increased geometric complexity, the conclusions reached are similar to those for the other two examples. The percentage of bad elements in the complete mesh is about the same for both techniques, but the percentage of surface triangles which connect to bad elements is

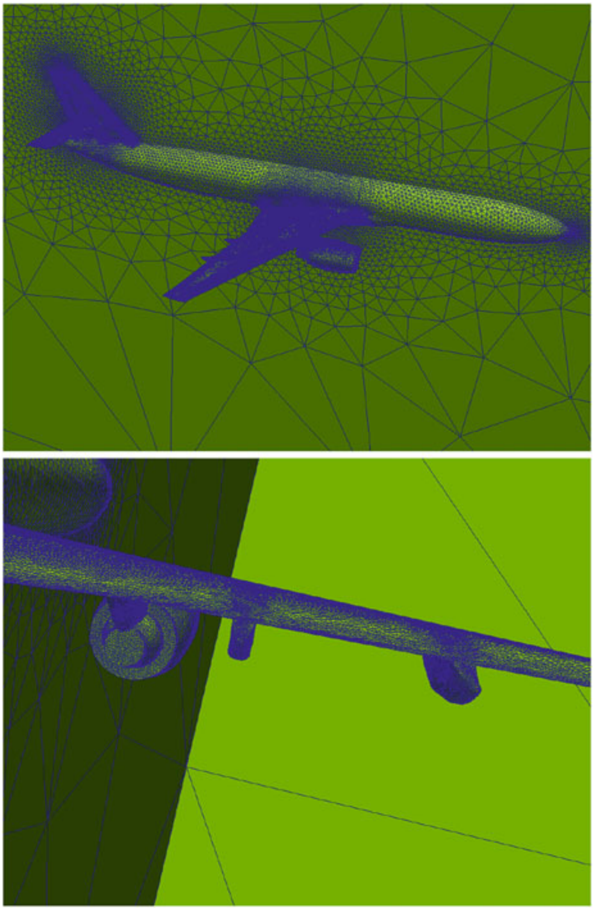

Fig. 6 B60: details of the surface mesh, indicating the complicated nature of the geometry on the underside of the wing

Table 2 Comparison of mesh quality measures for the B60

	Delaunay refinement	Lattice point insertion
No. nodes	1,358,561	1,221,477
No. elements	8,511,731	7,665,517
No. bad elements touching a surface	204,293	142,817
No. (%) surface elements connected to a bad element	18,417 (13 %)	4,356 (3 %)

significantly less when using lattice point insertion. Figure 9a, b show similar views of the meshes created by Delaunay refinement and lattice insertion respectively. In these figures, the surface elements are not plotted. It is apparent that there are areas of the mesh generated using lattice insertion which closely resemble the structured

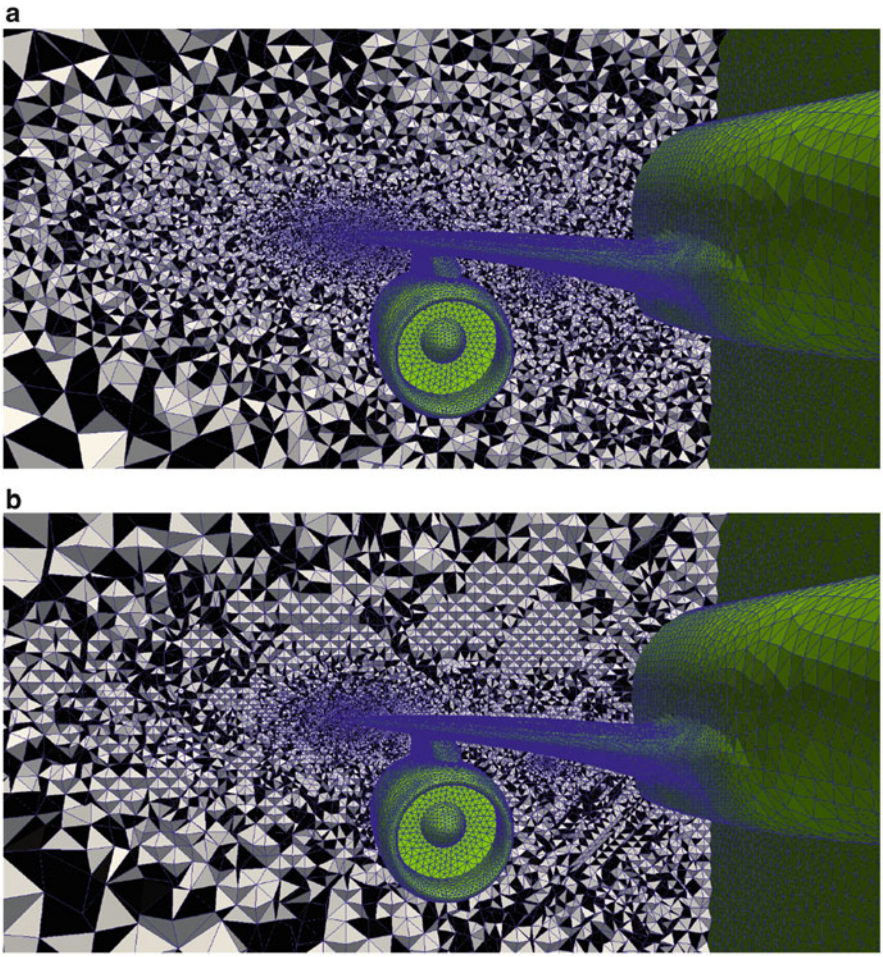

Fig. 7 B60: detail of the volume mesh generated using (**a**) Delaunay refinement; (**b**) lattice point insertion

ideal mesh. This effect is even clearer when comparing Fig. 10a, b, which show wider views of the meshes generated using Delaunay refinement and lattice insertion respectively. Bad elements are not coloured in these figures. Pairs of elements in Fig. 10b which lie on the same circumsphere may be potential candidates for merging [7] before the mesh is employed with a co-volume scheme.

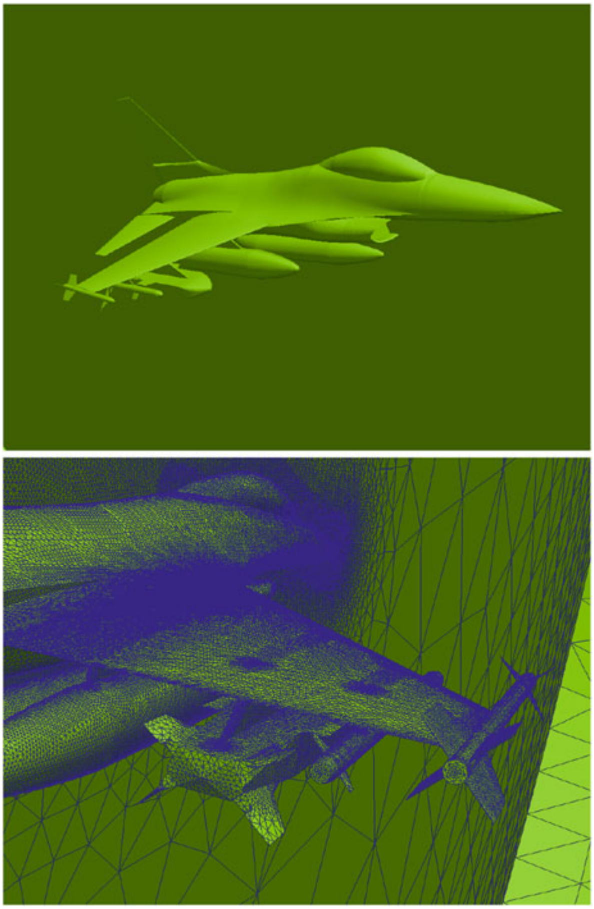

Fig. 8 F16 configuration, details of the surface and the surface mesh, indicating the challenges posed by the presence of the attached stores

Table 3 Comparison of mesh quality measures for the generic F16

	Delaunay refinement	Lattice point insertion
No. nodes	2,308,738	1,848,771
No. elements	14,389,566	11,367,474
No. bad elements touching a surface	424,796	309,181
No. (%) surface elements connected to a bad element	42,292 (14 %)	8,150 (3 %)

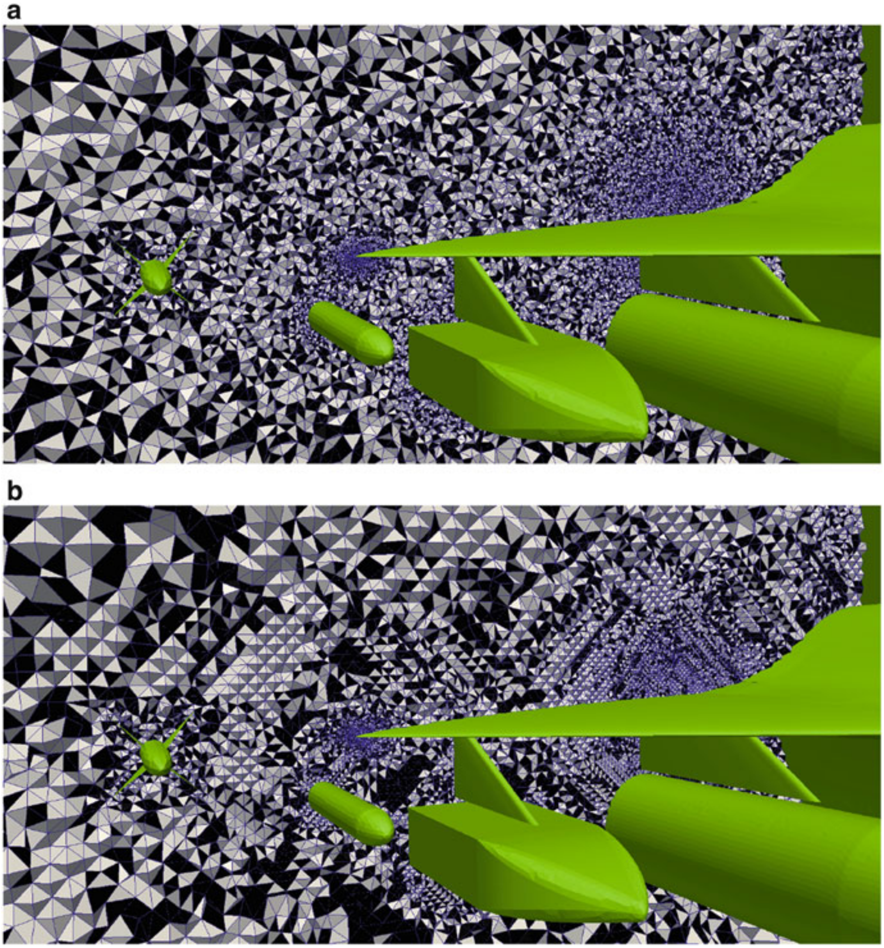

Fig. 9 Generic F16: detail of the volume mesh generated using (**a**) Delaunay refinement; (**b**) lattice point insertion

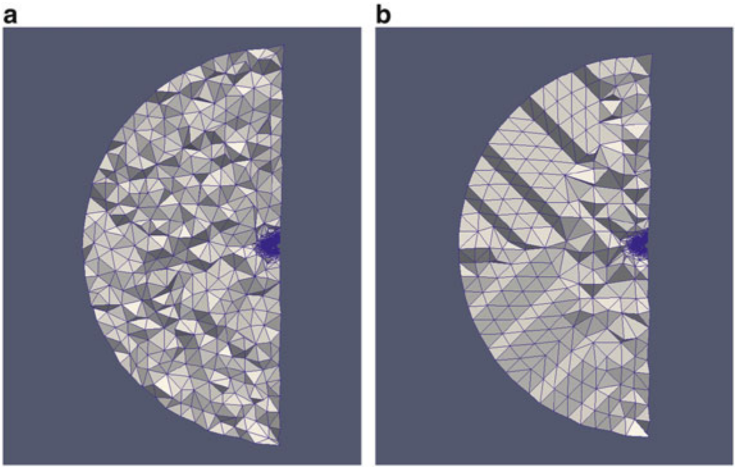

Fig. 10 Generic F16: wider view of the volume mesh generated using (**a**) Delaunay refinement; (**b**) lattice point insertion

3 Mesh Optimisation

Mesh optimisation techniques attempt to minimise an objective function, which has been constructed to represent any mesh quality measure of interest. Classical gradient based optimisation techniques prove ineffective at finding global optima, particularly when considering non-smooth objective functions [22], which are typical of mesh quality measures [23]. This motivates the use of gradient free techniques, which employ large populations of potential solutions and which move around the search space evaluating the objective function. This means any optimum that is found is likely to be global and, since no objective function gradient information is required, the smoothness of the objective function has no effect on the process. Examples illustrating the application of gradient free techniques to mesh optimisation can be found in the literature [4, 23, 24]. Here, we consider the application of the gradient free optimisation technique, modified cuckoo search, to the meshes that have been generated using the lattice point insertion technique.

3.1 Modified Cuckoo Search (MCS)

MCS [6] is a metaheuristic search algorithm that is inspired by the reproduction strategy of cuckoos. Cuckoo eggs have evolved to mimic the eggs of local birds by the process of laying eggs in the nests of host birds. These birds will discover eggs that do not resemble their own and destroy them. Thus, only cuckoo eggs which resemble the host bird eggs will pass genetic information to the next generation of cuckoos. MCS uses this simple mechanism as a basis for an optimisation algorithm. The algorithm turns out to be very effective, for high-dimensional objective functions. It has also been shown on benchmark functions that MCS has a high convergence rate to the global minimum.

3.2 Local Coordinate Optimisation

As noted above, the number of nodes in industrial meshes makes it impossible to apply a global gradient optimisation technique without some modification. Instead, a local approach is adopted, in which the position of each node of the Delaunay primal mesh is optimised in turn. Since MCS has a very high convergence rate, this is computationally feasible [4]. The objective function

$$F(k) = \sum_{i=1}^{E} W_i \frac{\|\mathbf{C}_i - \mathbf{V}_i\|}{\delta_i} \tag{1}$$

is minimised at each node, where k is the node index, E is the number of elements which include node k, the index i nodes to the elements including node k, $\mathbf{C}_i$ is the position vector of the centroid of element i, $\mathbf{V}_i$ is the position vector of the Voronoi vertex of i, and δ_i is the mean edge length of element i. The weight W_i is equal to zero if V_i is inside element i and equal to one otherwise. For each node, MCS is applied for five generations and the position of the node is moved towards the optimally located position [4]. After the volume mesh is generated, this procedure is applied recursively until no improvement in global mesh quality is obtained.

3.3 Local Weight Optimisation

Using this technique, it is not always possible to eliminate all bad elements from a mesh. A second approach is adopted, in which the requirement that a dual edge be a bisector of the corresponding Delaunay edge is relaxed, but the corresponding Voronoi vertex is moved in such a way to retain orthogonality [7]. To illustrate this process consider the two dimensional element shown in Fig. 11. If circles of equal radius are positioned such that their centres lie on the nodes of a Delaunay element, then the corresponding Voronoi vertex is located at the point of intersection of the common chords of these circles, as shown in Fig. 11a. By reducing the radius of the circle at B, the Voronoi vertex is pulled inside the element as shown in Fig. 11b. Changing this radius will also affect the position of Voronoi vertices in surrounding elements such that orthogonality is obtained. By allocating a weight to each Delaunay node, which corresponds to the change in the radius of the circle

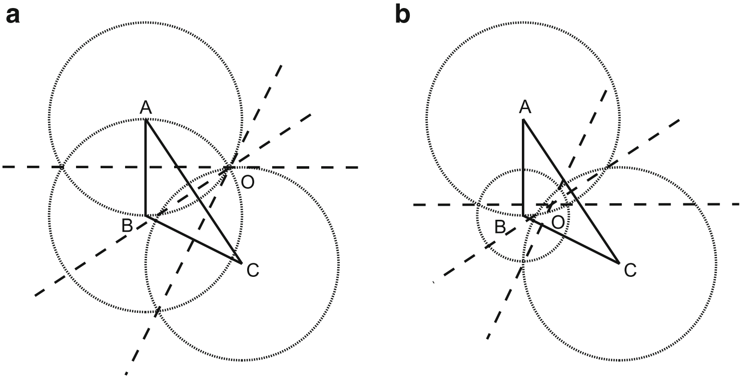

Fig. 11 (**a**) A bad element ABC and the corresponding Voronoi vertex O; (**b**) moving the Voronoi vertex O to repair the bad element ABC

Table 4 Quality measures for the ONERA M6 wing mesh generated using lattice point insertion and modified cuckoo search optimisation

No. nodes	78,999
No. elements	485,254
No. of bad elements touching a surface	4,154
No. (%) surface elements connected to a bad element	99 (0.6 %)

(or the sphere in three dimensions), the coordinates of each Voronoi vertex can be readily obtained [4].

After this coordinate optimisation procedure is completed, MCS is used to optimise the weight of each node, in turn, using the objective function of Eq. (1). This procedure is repeated until no further improvement is obtained.

3.4 Examples

The optimisation procedures described above are applied to the meshes generated previously using the lattice point insertion technique. The results are presented in the same fashion as previously. All optimisation processes were run until no further improvement was produced. To give an idea of the runtime requirements for this approach, the M6 example was completed in around half an hour, while the other examples were left to run overnight for around 8–12 h.

The quality measures for the ONERA M6 volume mesh, generated and optimised as described above, are given in Table 4. A view of a cut through the mesh is shown in Fig. 12. Bad elements are coloured black in this figure. The optimisation process reduces the total percentage of faces which are connected to a bad element has reduced from 2 to 0.6 %.

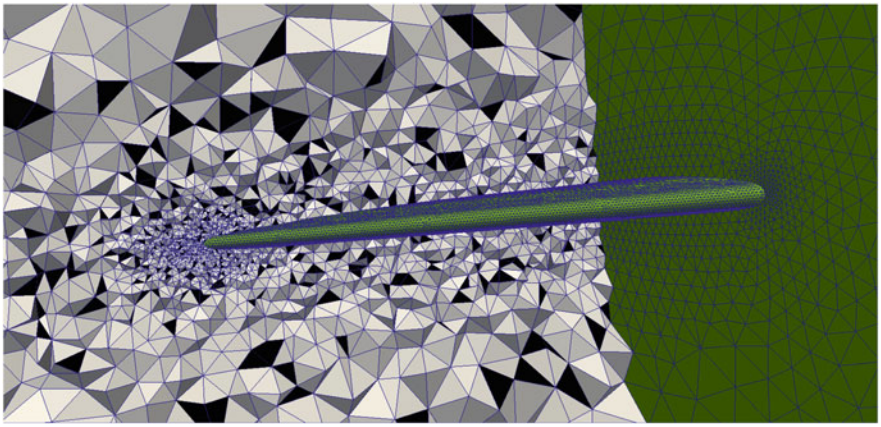

Fig. 12 ONERA M6 wing: detail of the volume mesh generated using lattice point insertion and modified cuckoo search optimisation

Table 5 Quality measures for the B60 mesh generated using lattice insertion and modified cuckoo search optimisation

No. nodes	1,120,620
No. elements	7,086,250
No. bad elements touching a surface	47,305
No. (%) surface elements connected to a bad element	1,550 (1 %)

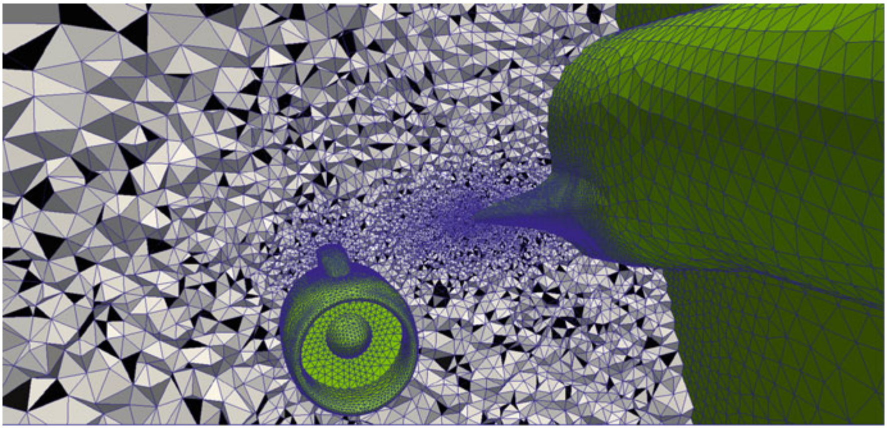

Fig. 13 B60: detail of the volume mesh generated using lattice point insertion and modified cuckoo search optimisation

The quality measures for the mesh produced for the B60, after optimisation, are presented in Table 5. A view of a cut through this mesh is shown in Fig. 13 and the bad elements are again coloured black. The percentage of surface elements connected to bad elements has also dropped from 3 to 1 %.

Table 6 Quality measures for the generic F16 mesh generated using lattice insertion and modified cuckoo search optimisation

No. nodes	1,657,926
No. elements	10,280,144
No. bad elements touching a surface	106,631
No. (%) surface elements connected to a bad element	3,432 (2 %)

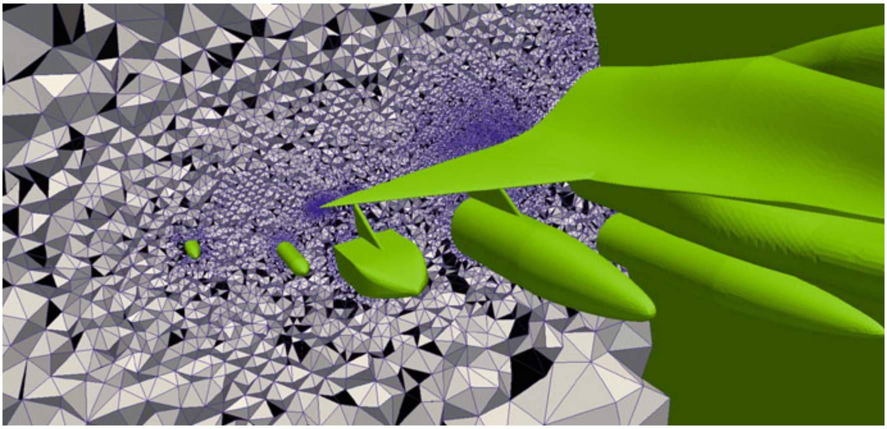

Fig. 14 Generic F16: detail of the volume mesh generated using lattice point insertion and modified cuckoo search optimisation

The results for the mesh created for the generic F16 show the same trend as the other examples. The mesh quality measures are given in Table 6 and a detail of a cut through the mesh is shown in Fig. 14. In this figure, black elements indicate bad elements. The optimisation process reduces the percentage of faces which are connected to bad elements from 3 to 2 %.

4 Conclusions

In this paper, we have detailed our current line of investigation into the problem of generating well centred tetrahedral meshes for general geometries. The lattice insertion method appears to be a promising approach, which offers certain advantages over competing methods to produce the quality required for a co-volume schemes. The steps involved in the algorithm are based on a few simple rules, which gives confidence that the method will be robust when it is applied to problems involving complex geometries. This claim is supported by the quality of the initial results obtained for the examples presented here. Future investigations will concentrate on the possibility of controlling the orientation of the lattice, of merging elements as part of the mesh generation process and of preparing the surface mesh by moving nodes into positions better suited for building the lattice. We have also shown that, by additionally applying previously developed optimisation techniques, we can significantly improve the quality of these meshes. The generation of this high quality near field meshes will then enable the extension of the stitching methods into three dimensions.

Acknowledgements Sean Walton acknowledges the financial support for this work provided by the UK Engineering and Physical Sciences Research Council (EPSRC) under Research Grant EP/K000705.

References

1. Harlow, F.H., Welch, J.E.: Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. Phys. Fluids **8**, 2182–2189 (1965)
2. Yee, K.: Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media. IEEE Trans. Antennas Propag. **14**, 302–307 (1966)
3. Sazonov, I., Wang, D., Hassan, O., Morgan, K., Weatherill, N.: A stitching method for the generation of unstructured meshes for use with co-volume solution techniques. Comput. Methods Appl. Mech. Eng. **195**, 1826–1845 (2006)
4. Walton, S., Hassan, O., Morgan, K.: Reduced order mesh optimisation using proper orthogonal decomposition and a modified cuckoo search. Int. J. Numer. Methods Eng. **93**, 527–550 (2013)
5. Walton, S.: Gradient free optimisation in selected engineering applications. PhD Thesis, Swansea University (2013)
6. Walton, S., Hassan, O., Morgan, K.: Modified cuckoo search: A new gradient free optimisation algorithm. Chaos Solitons Fractals **44**, 710–718 (2011)
7. Xie, Z.Q., Hassan, O., Morgan, K.: Tailoring unstructured meshes for use with a 3D time domain co-volume algorithm for computational electromagnetics. Int. J. Numer. Methods Eng. **87**, 48–65 (2011)
8. Sazonov, I., Hassan, O., Morgan, K., Weatherill, N.P.: Smooth Delaunay–Voronoi dual meshes for co-volume integration schemes. In: Pebay, P.P. (ed.) Proceedings of the 15th International Meshing Roundtable, pp. 529–541. Springer, Berlin (2006)
9. Naylor, D.J.: Filling space with tetrahedra. Int. J. Numer. Methods Eng. **44**, 1383–1395 (1999)
10. Eppstein, D., Sullivan, J.M., Üngör, A.: Tiling space and slabs with acute tetrahedra. Comput. Geom. Theory Appl. **27**, 237–255 (2004)
11. Sazonov, I., Hassan, O., Morgan, K., Weatherill, N.P.: Generating the Voronoï–Delaunay dual diagram for co-volume integration schemes. In: Gold, C.M. (ed.) Proceedings of the 4th International Symposium on Voronoï Diagrams in Science and Engineering, ISPRS, IEEE CPS, pp. 199–204 (2007)
12. van der Zee, E., Hirani, A., Guoy, D.: Triangulation of simple 3D Shapes with well-centered tetrahedra. In: Garimella, R. (ed.) Proceedings of the 17th International Meshing Roundtable. Springer, Berlin (2008)
13. van der Zee, E., Hirani, A., Guoy, D., Ramos, E.: Well-centered triangulation. SIAM J. Sci. Comput. **31**, 4497–4523 (2010)
14. Yan, D.-M., Wang, W., Lévy, B., Liu, Y.: Efficient computation of clipped Voronoi diagram for mesh generation. Comput. Aid. Des. **45**, 843–852 (2013)
15. Liu, Y., Wang, W., Lévy, B., Sun, F., Yan, D.-M., Lu, L., Yang, C.: On centroidal Voronoi tessellation – Energy smoothness and fast computation. ACM Trans. Graph. **28**, 4 (2009)
16. Peraire, J., Vahdati, M., Morgan, K., Zienkiewicz, O.C.: Adaptive remeshing for compressible flow computations. J. Comput. Phys. **72**, 449–466 (1987)

17. Weatherill, N.P., Hassan, O.: Efficient three-dimensional Delaunay triangulation with automatic point creation and imposed boundary constraints. Int. J. Numer. Methods Eng. **37**, 2005–2040 (1994)
18. Morgan, K., Peraire, J., Peiro, J.: Unstructured grid methods for compressible flows. Report 787: Special Course on Unstructured Grid Methods for Advection Dominated Flows, AGARD, Paris, pp. 5.1–5.39 (1992)
19. Radovitzky, R., Ortiz, M.: Tetrahedral mesh generation based on node insertion in crystal lattice arrangements and advancing-front-Delaunay triangulation. Comput. Methods Appl. Mech. Eng. **187**, 543–569 (2000)
20. Rajan, V.T.: Optimality of the Delaunay triangulation in Rd. Discrete Comput. Geom. **12**, 189–202 (1994)
21. Kennel, M.: KDTREE 2: Fortran 95 and C++ software to efficiently search for near neighbors in a multi-dimensional Euclidean space. ArXiv Physics e-prints arXiv:physics/0408067v2 (2004)
22. Praveen, C., Duvigneau, R.: Low cost PSO using metamodels and inexact pre-evaluation: application to aerodynamic shape design. Comput. Methods Appl. Mech. Eng. **198**, 1087–1096 (2009)
23. Park, J., Shontz, S.M.: Two derivative-free optimization algorithms for mesh quality improvement. Proc. Comput. Sci. **1**, 387–396 (2010)
24. Yilmaz, A., Kuzuoglu, M.: A particle swarm optimization approach for hexahedral mesh smoothing. Int. J. Numer. Methods Fluids **60**, 55–78 (2009)