Heliyon 5 (2019) e02319

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.heliyon.com

New approaches for Delaunay triangulation and optimisation

Logah Perumal [*]

Faculty of Engineering and Technology, Multimedia University, Jalan Ayer Keroh Lama, Bukit Beruang, 75450, Melaka, Malaysia

ARTICLE INFO

Keywords:
Mechanical engineering
Finite element methods
Mathematical modeling
Geometry
Computing methodology
Computer simulation
Generalised equation
Mapping
Sample points
Delaunay triangulation
Mesh optimisation
Aspect ratio
Element skewness

ABSTRACT

New techniques are presented for Delaunay triangular mesh generation and element optimisation. Sample points for triangulation are generated through mapping (a new approach). These sample points are later triangulated by the conventional Delaunay method. Resulting triangular elements are optimised by addition, removal and relocation of mapped sample points (element nodes). The proposed techniques (generation of sample points through mapping for Delaunay triangulation and mesh optimisation) are demonstrated by using Mathematica software. Simulation results show that the proposed techniques are able to form meshes that consist of triangular elements with aspect ratio of less than 2 and minimum skewness of more than 45°.

1. Introduction

Mesh creation is one of the crucial steps in Finite Element Method (FEM). There are various methods to create a triangular mesh, but Delaunay is the most commonly used. Delaunay triangulation generates triangular meshes without any overlap or void by connecting sample points which are scattered within the problem domain. Connection of the sample points is accomplished by first forming circles in which each circle contains 3 sample points on its circumference and does not contain any other point within the circle. These 3 sample points are then connected to form a triangular element. Delaunay triangulation tends to maximise the smallest angle which is formed within an element through flipping technique [1].

Thus, sample points need to be generated within the problem domain first in order to initiate the triangulation (mesh creation). Some of the techniques to generate the sample points are described in the section 1.1. These techniques generate initial sample points which require further optimisation in order to obtain required distribution (uniform or adaptive), depending on the density/size functions. The optimisation can be either carried out before or after triangulation. These two types of optimisation techniques are described in section 1.2.

In this work, sample points are generated for a problem domain through mapping (a new approach) and the mapped sample points are later triangulated via conventional Delaunay method. The mapping is accomplished by using generalised equation which was proposed for numerical integration within finite element method (FEM) and extended finite element method (XFEM) in earlier works [2, 3, 4]. Triangular elements that are formed are later optimised by using a new optimisation framework. The proposed techniques (generation of sample points through mapping, triangulation and mesh optimisation) are demonstrated by using Mathematica software. Modelling and execution of mapping of sample points in Mathematica is simpler and faster, since the mapping involves fully numerical algorithms (symbolic manipulation has been eliminated entirely) [4]. The mapped sample points are later triangulated by using programming function "*DelaunayTriangulation*". Optimisation of the triangular elements is done by addition, removal and relocation of sample points. Corresponding Mathematica functions to carry out the three tasks above are "*Append*", "*Delete*" and "*ReplacePart*", respectively.

The new approach to generate sample points in the problem domain is presented in section 2 while the framework for the element optimisation is presented in section 3. Some examples are provided in section 4. The paper is finally concluded in section 5, with some remarks on the advantages and disadvantages of the proposed technique.

[*] Corresponding author.
E-mail address: logah.perumal@mmu.edu.my.

https://doi.org/10.1016/j.heliyon.2019.e02319
Received 26 June 2019; Received in revised form 19 July 2019; Accepted 13 August 2019

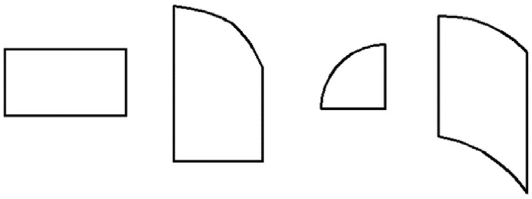

Fig. 1. Examples of decomposed domains according to Fubini's theorem. From left: a domain with 4 constant lines, a domain with 3 constant lines and 1 curve, a domain with 2 constant lines and 1 curve, and a domain with 2 parallel constant lines and 2 curves.

1.1. Techniques to generate sample points

There are various methods that can be used to generate sample points within a problem domain for the purpose of triangular mesh generation. Some of the techniques are:

a. Poisson disk sampling [5]
b. Spectra based sampling [6]
c. Point insertion method [7]
d. Bubble-packing method [8, 9]
e. NURBS parametric equation [10]
f. Initialisation tree [11]
g. Point cloud [12]
h. Artificial spot pattern [13]
i. Atomic packing arrangements [14]
j. Quadtree-like background mesh [15, 16]

Brief descriptions about the listed techniques are provided below.

1.1.1. Poisson disk sampling

Poisson disk sampling technique can be implemented in various forms or styles, which are given different names such as dart throwing, relaxation (relaxation dart throwing and Lloyd's relaxation), tiling (Penrose tiling, Shade's Poisson disk tiles, tiled blue noise samples, recursive Wang tiles and edge-based, template and corner-based Poisson disk tiles) and importance sampling (hierarchical importance sampling) [5].

Dart throwing works by generating sample points within the problem domain randomly. Upon substantial amount of time, the method would create a dense distribution within the problem domain. The distribution is maintained uniform by deleting sample points which do not meet the minimum separation condition. The minimum separation condition (set by the user) is simply the minimum distance between the already existing sample point and the new sample point. The process will stop when there is no empty space for the sample points to occupy while meeting the minimum separation condition.

Relaxation schemes (Lloyd's) is often used to optimise the sample points which are generated from the Poisson disk sampling methods.

Tiling is a method which generates sample points faster compared to dart throwing (dart throwing generates one point at a time). This is made possible by the use of a set of tiles with several preoccupied sample points. These preoccupied sample points within a set of tiles (for example 8 tiles) is located such a way that they satisfy the minimum separation condition (these preoccupied sample points can be generated by using the dart throwing method). The problem domain is then fully covered with duplicates of this set of tiles.

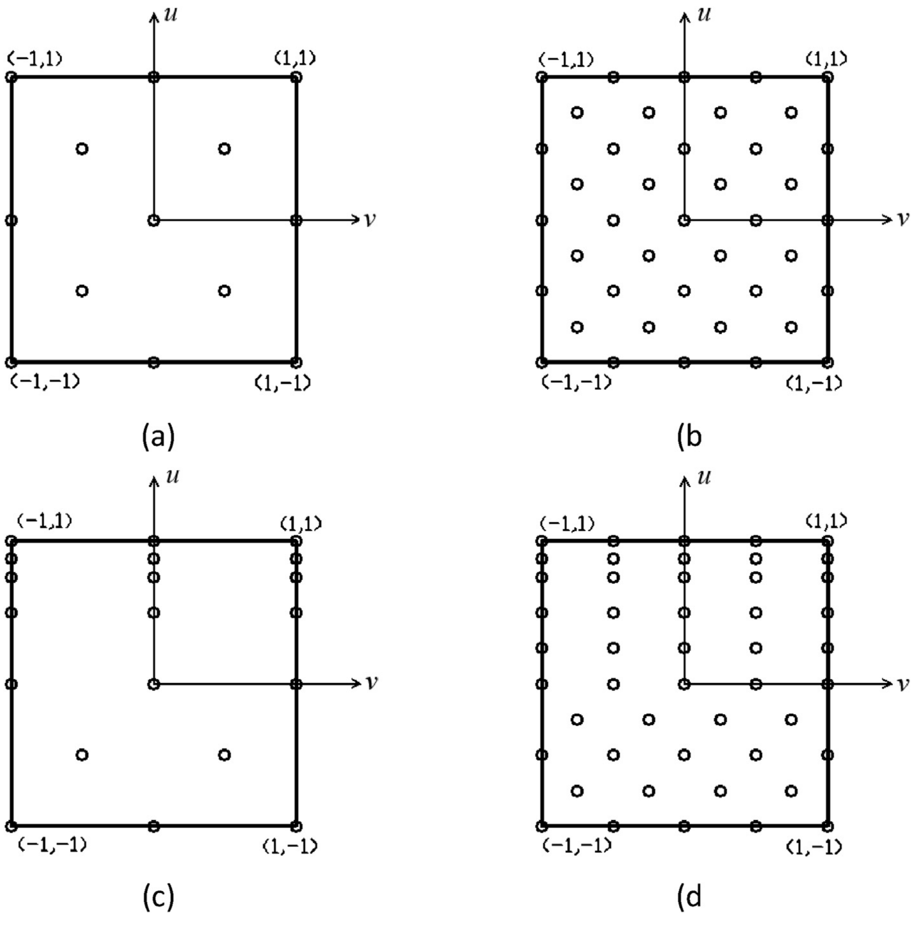

Fig. 2. Reference square domains with sample points (a) SP13 (b) SP41 (c) ASP20 (d) ASP48.

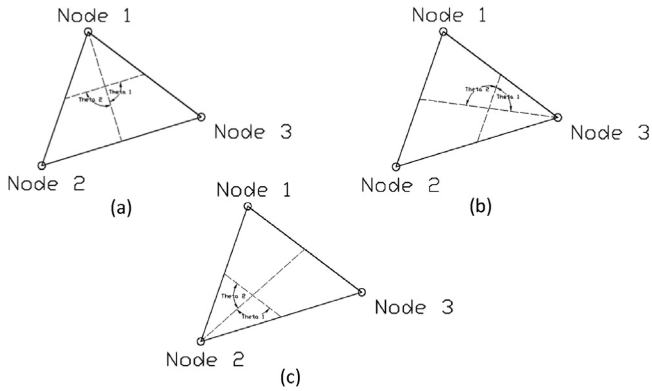

Fig. 3. Three combinations of intersections within a triangular element (a) skewness corresponding to node 1 (b) skewness corresponding to node 2 (c) skewness corresponding to node 3.

Importance sampling is a form of adaptive sampling. The method works together with density/importance functions in order to generate sample points which adapt to either to the geometry of the domain or to the change of field variable within the domain. Density/importance functions are used to locate regions which require higher number of elements or denser meshes [17]. In some cases, the adaptive function is also known as sizing function [7, 18].

1.1.2. *Spectra based sampling*

Spectra based sampling generates sample points that match a particular Fourier spectrum function [6]. This technique enables generation of sample points with different noise patterns, as required by the user (not limited to blue noise samples).

1.1.3. *Point insertion method*

In this method, sample points are inserted into the problem domain continuously as meshes are created. The boundary of the problem domain is discretised first to generate boundary sample points and then interior sample points are inserted (one at a time) such a way that they satisfy the Delaunay criterion as well as the density/sizing functions, if present. This technique is also known as advancing front approach. Other techniques have been proposed for the point insertion. Such approaches are addition of points at the centre of existing triangles, at the centre of element circumcircle, along a Voronoi segment and along edges [7].

1.1.4. *Bubble-packing method*

In this method, the sample points are in the form of bubble-like spheres/circles which are randomly generated within the problem domain. The triangulation is formed by connecting the centres of three adjacent bubble-like spheres/circles through Delaunay method (the centre of the bubbles act as sample points). The initial bubbles are placed by subdivision of various dimensional geometries. For example, curves are subdivided by using a binary tree, a surface by a quadtree and a volume by an octree [8]. In another work, the initial bubbles are placed randomly within the problem domain by following a relationship which is in terms of the area of the surface and desired triangle area or edge length [9]. However, gaps and overlap of bubbles might occur during the initial bubble placement.

The initial bubbles are then made uniform and optimised by restricting distance between two adjacent bubbles. This is done by introducing an inter-bubble force between the two adjacent bubbles. When two bubbles are very close to each other, a repulsive force is generated. On the other hand, attractive force is produced when two bubbles are far apart from each other. Force balance is reached when two adjacent bubbles reached stable distance. This approach is similar to van der Waals force [8]. Stable configuration (force balance is achieved) for the bubbles is obtained through dynamic simulation.

1.1.5. *NURBS parametric equation*

Non-Uniform Rational B-Splines (NURBS) is a modelling technique which can represent arbitrary curve (or surface in 3D) by using parametric equation which is in terms of parameter u (NURBS curves are defined within the range of u_{min} and u_{max}), control points, weights and B-splines basis functions. The control points and weighs can be altered to match the curve or surface being modelled.

In [10], the authors developed a method to form triangulations for 3D surfaces by first parametrising the NURBS curves in physical domain. NURBS hull is then formed through connection of control points (in rows and columns) in the actual physical domain. These control points are later mapped to a parameter domain. These control points in the parameter domain act as initial sample points for triangulation. Distribution of these initial sample points is made uniform or non-uniform (in case of adaptive meshing) through physical simulation such as molecular dynamics method. Once the distribution of the sample points has been optimised in the parameter domain, these sample points are then re-mapped to the physical domain. Finally triangulation takes place based on Delaunay method.

In [19], the authors formed guide lines instead of mapping the control points directly onto the parameter domain.

In another similar work [20], the authors first mapped the NURBS boundary surfaces (boundary curves of trimmed surfaces) from the physical surface onto the parameter domain and later generated the initial sample points directly onto the parameter domain. The initial uniform sample points are created based on distance function between the sample points. Some of the sample points lie outside the problem domain, which are later eliminated through Monte Carlo simulation.

1.1.6. *Initialisation tree*

This method (proposed in [11]) works by first forming a mesh consisting of several quadrangles within the problem domain (background mesh). The quadrangles are formed such a way that the density of the quadrangles is according to the geometry segments. The resulting quadrangles are called initialisation tree [11]. Initial sample points are generated by inserting points into these quadrangles individually, either by randomly or structured.

In case of random point distribution (within a quadrangle), the sample points are generated by using a pseudo-random coordinate pair within the quadrangle. On the other hand, the structured distribution is generated by inserting one or more points into the quadrangle according to specific patterns. Four patterns are described, which are insertion of: one point at the centre of the quadrangle, four points which are equally spaced to form a square, five points with combination of the first and the second patterns (four points forming a square and a point at the centre), and nine points which are equally spaced to form a 3 by 3 square.

The mesh of quadrangles is later removed, leaving behind the initial sample points which are scattered within the problem domain. This initial distribution of sample points is optimised by using physics based simulation such as particle dynamics.

1.1.7. *Point cloud*

Point cloud is a collection of sample points in space (normally in 3D). It is usually generated by using 3D scanners. Point clouds are used in generating triangular elements (embedded in 3D) in image modelling. The point cloud can be created through the technique known as Structure from Motion or computed for the pixels of the region of interest [12]. Some of the software packages which enable generation of point cloud (in other application areas) are Autocad [21], ArcGIS [22], Civil 3D [23] and so on.

1.1.8. *Artificial spot pattern*

This technique uses grey-scale (bitmap) image to form initial sample

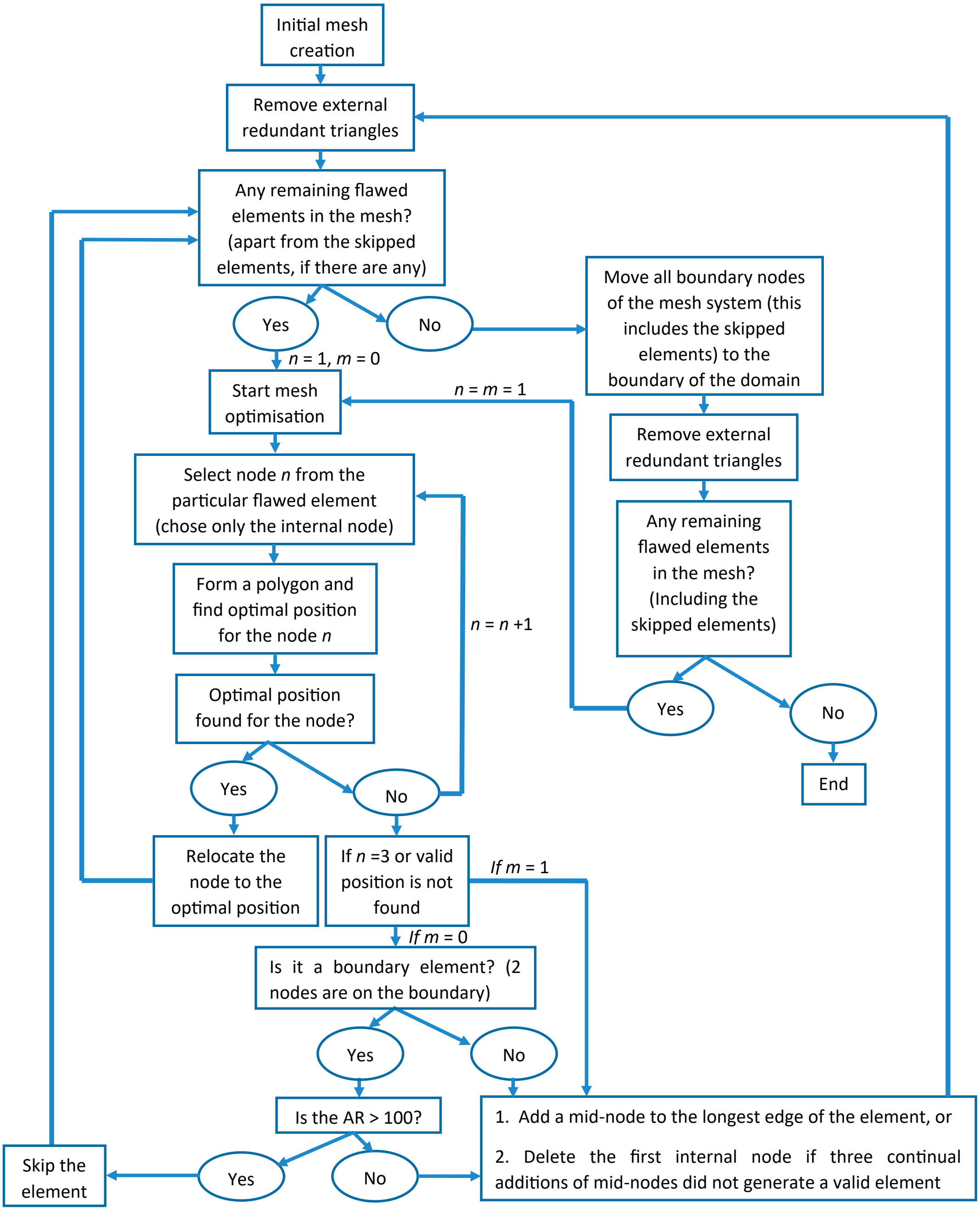
Initial mesh creation
Remove external redundant triangles
Any remaining flawed elements in the mesh? (apart from the skipped elements, if there are any)
Yes
No
n = 1, m = 0
Move all boundary nodes of the mesh system (this includes the skipped elements) to the boundary of the domain
Start mesh optimisation
n = m = 1
Remove external redundant triangles
Select node n from the particular flawed element (chose only the internal node)
Any remaining flawed elements in the mesh? (Including the skipped elements)
Form a polygon and find optimal position for the node n
n = n +1
Optimal position found for the node?
Yes
No
Yes
No
End
Relocate the node to the optimal position
If n =3 or valid position is not found
If m = 1
If m = 0
Is it a boundary element? (2 nodes are on the boundary)
Yes
No
Is the AR > 100?
1. Add a mid-node to the longest edge of the element, or
2. Delete the first internal node if three continual additions of mid-nodes did not generate a valid element
Skip the element
Yes
No

Fig. 4. Framework for mesh optimisation.

points. A grey-scale image, which actually is the problem domain, consists of different colour contours (such as black, dark grey, light grey, and white) which form different coloured regions within the domain. Initial sample points (known as artificial spots) are placed on the interface between (or boundary of) different coloured regions, including the boundary of the problem domain.

These initial sample points are then fed into the Gray–Scott model to generate more sample points in the interior of the domain (interior of the different coloured regions). Gray–Scott model generates more sample points by using finite difference method [13].

1.1.9. Atomic packing arrangements

The author in [14] created the initial sample points based on atomic packing arrangements in a crystal lattice. Body-Centered Cubic (BCC) and cubic lattice arrangements are mimicked for the positioning of the initial sample points (also known as seed points). Some randomness in the distribution of the sample points is introduced through implementation of randomness parameter *f*. Mimicry of Face-Centered Cubic (FCC) and Hexagonal Close-Packed (HCP) arrangements were found to produce dense sample points which were not supported by the software utilised in their work.

1.1.10. Quadtree-like background mesh

This technique is an adaptation of strategies used for meshing and detection of discontinuity in extended finite element method (XFEM) [2]. Utilisation of this technique to generate initial sample points can be seen in [15].

The initial sample points are either randomly generated or generated

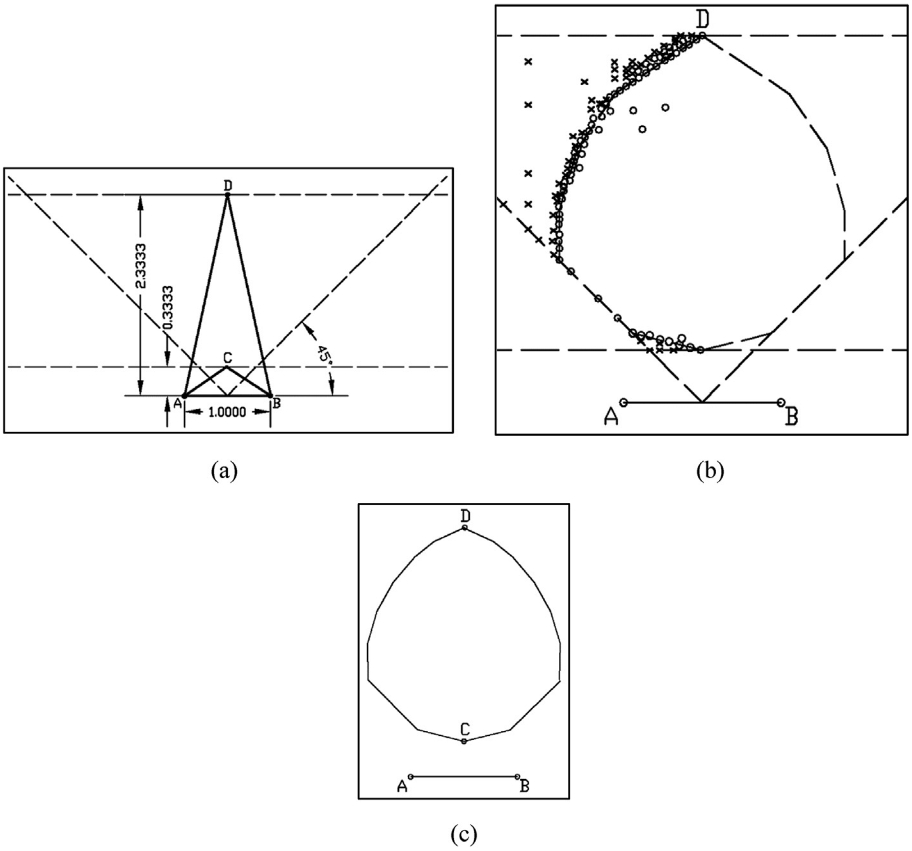

Fig. 5. Determination of valid region. (a) defining initial boundaries based on isosceles triangles. (b) checking validity of sample points within the region bounded by the initial boundaries (c) finalised valid region.

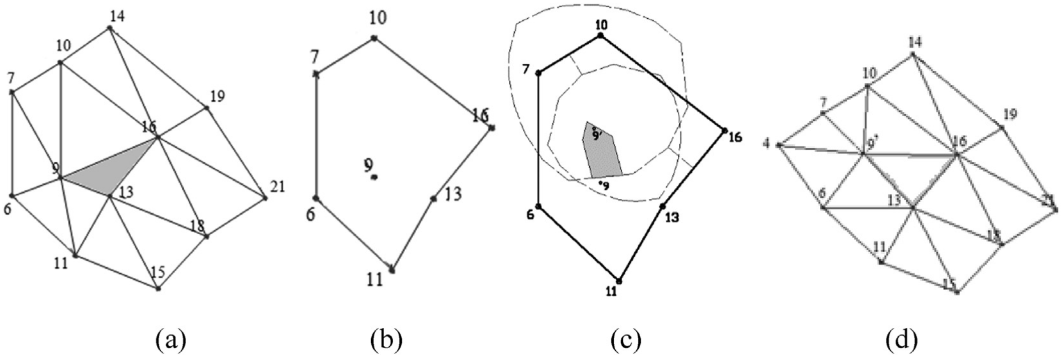

Fig. 6. Optimisation of a flawed element through node relocation. (a) a mesh consisting of a flawed element 9-13-16. (b) formation of a polygonal structure based on the selected node. (c) formation of final valid area (greyed) based on the intersections of individual valid regions. (d) mesh after the optimisation.

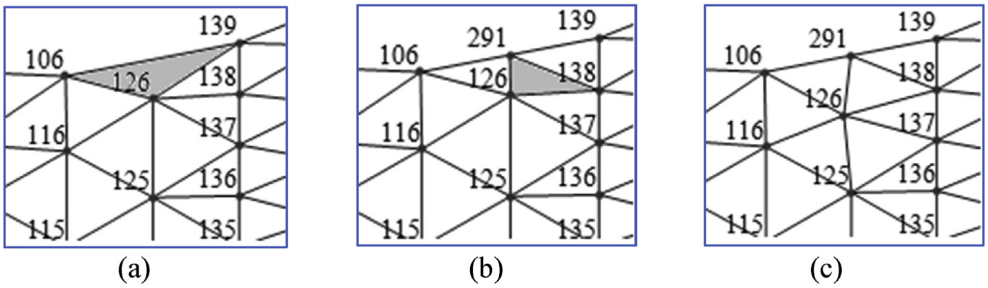

Fig. 7. Optimisation of a flawed element through node insertion and relocation. (a) a mesh consisting of a flawed element 106-126-139. (b) insertion of a mid-node 291 along the longest edge of the element. (c) relocation of node 126.

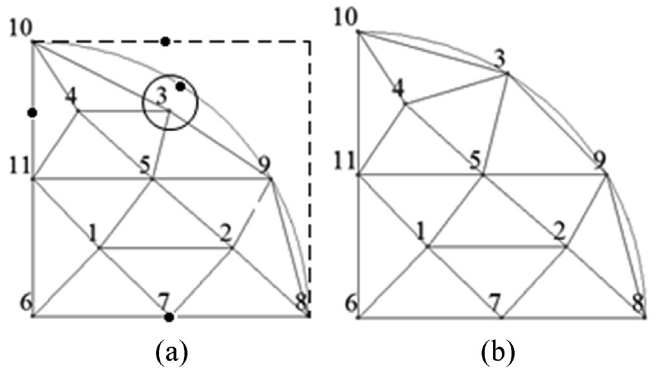

Fig. 8. Displacement of boundary node of a mesh system to the domain boundary (a) inaccurate positioning of node 3 (b) optimised mesh.

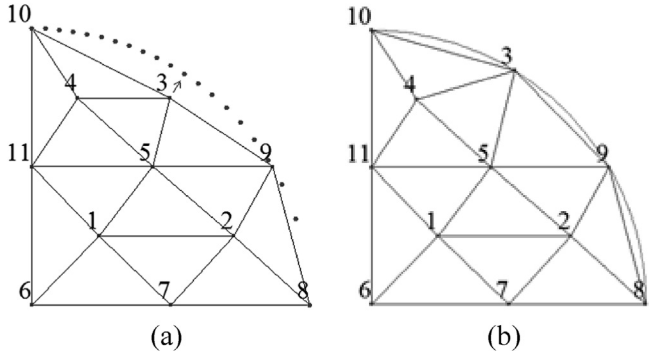

Fig. 9. Displacement of boundary node based on the second approach. (a) inaccurate positioning of node 3 (b) optimised mesh.

according to the density function (adaptive meshing). Those sample points which are located outside the problem domain are later detected (based on the sign of level set function) and moved randomly to the interior of problem domain for the former method, while the latter generates the sample points within the problem domain directly, due to the density functions (density functions are described for the regions within the problem domain).

Sample points for the boundary of the problem domain are obtained with the help of multi-resolution based nested hierarchical background mesh consisting of grid blocks. This mesh is similar to adaptive quadtree mesh and another similar application is seen in [11] (known as initialisation tree, as described in section 1.1.6). Boundary of the problem domain cuts and passes through some of the grid blocks. These specific grid blocks are known as cut cells. One sample point is inserted at the centre of each cut cell. These sample points may not be located directly onto the boundary curve. Therefore, these sample points inside the cut cells are later moved onto the boundary of the problem domain through mapping. Quadtree-like background mesh is also utilised by [16] to generate sample points. The author used intersecting points of the boundary of the problem domain with the cut cells as sample points. In alternate case, corners of a block can also be taken as sample points.

1.2. *Optimisation*

There are two methods to generate optimised mesh. One is by optimising the initial sample points which are generated based on the methods described in section 1.1 and the other is to optimise the finite elements which are formed as the result of meshing. These two optimisation methods are briefly reviewed below.

The sample points which are generated from the methods listed above are later optimised (improved uniformity of the initial sample point distribution) based on sampling criterion. Some of the techniques used to optimise the distribution are the centroidal Voronoi tessellations [24], kernel-based sampling [25, 26], sequential sampling strategy [27], particle based approach (particle kernel) [15], smoothed particle hydrodynamics [15], molecular dynamics, particle dynamics, Monte Carlo, other physics-based simulations [11] and extrinsic methods [28]. In [29], the authors used truss equilibrium model system to optimise distribution of the sample points. Another interesting optimisation of the sample points

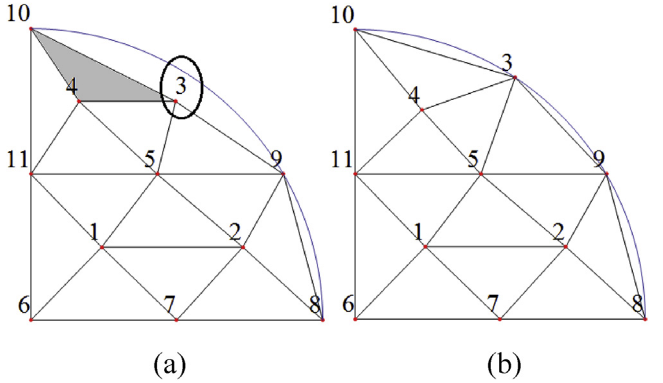

Fig. 11. Triangulation by using SP13. (a) initial triangulation. (b) optimised mesh.

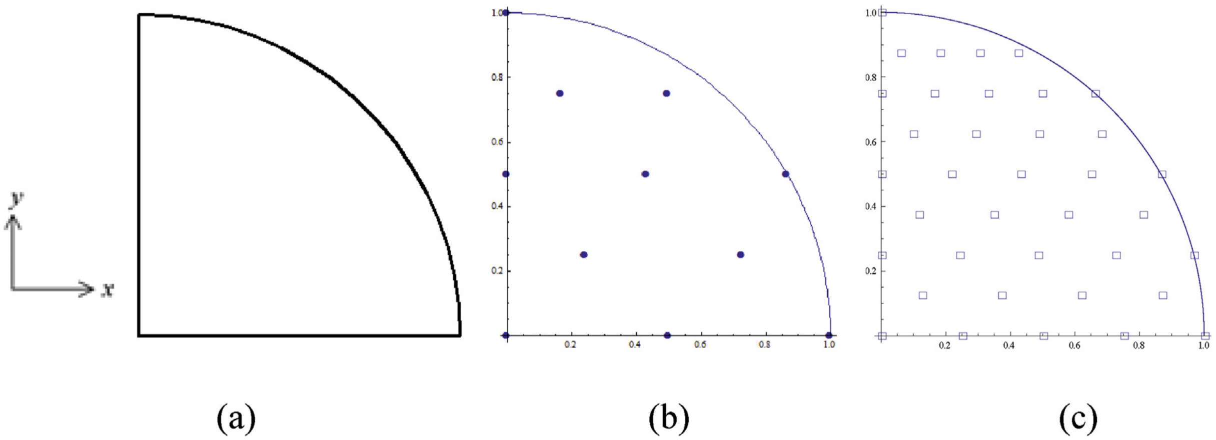

Fig. 10. Sample point generation for quadrant of a circle. (a) problem domain. (b) Initial distribution of sample points by using SP13. (c) Initial distribution of sample points by using SP41.

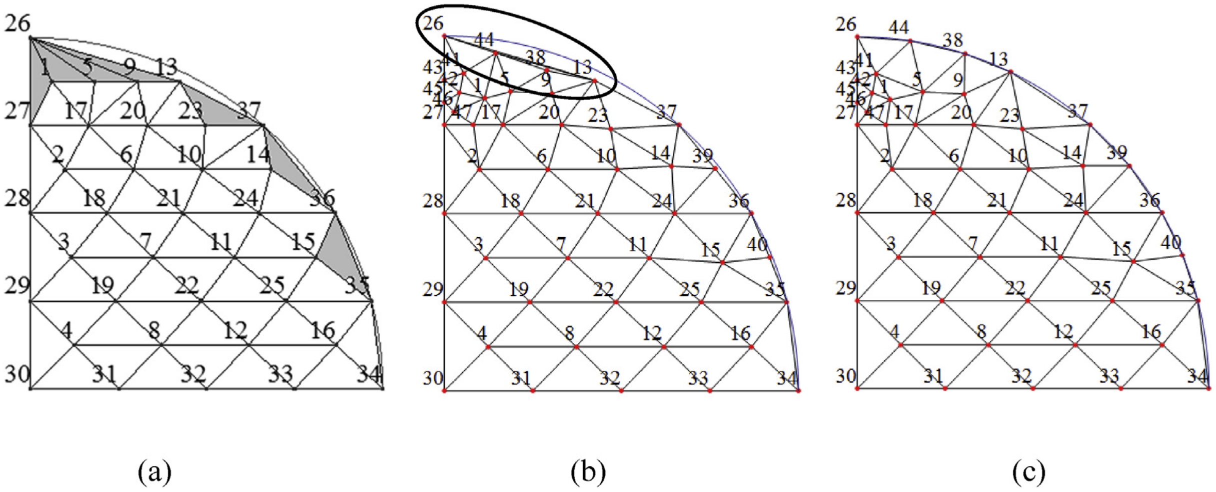

Fig. 12. Triangulation by using SP41. (a) initial triangulation. (b) redundant triangles inside the mesh with AR>100. (c) optimised mesh.

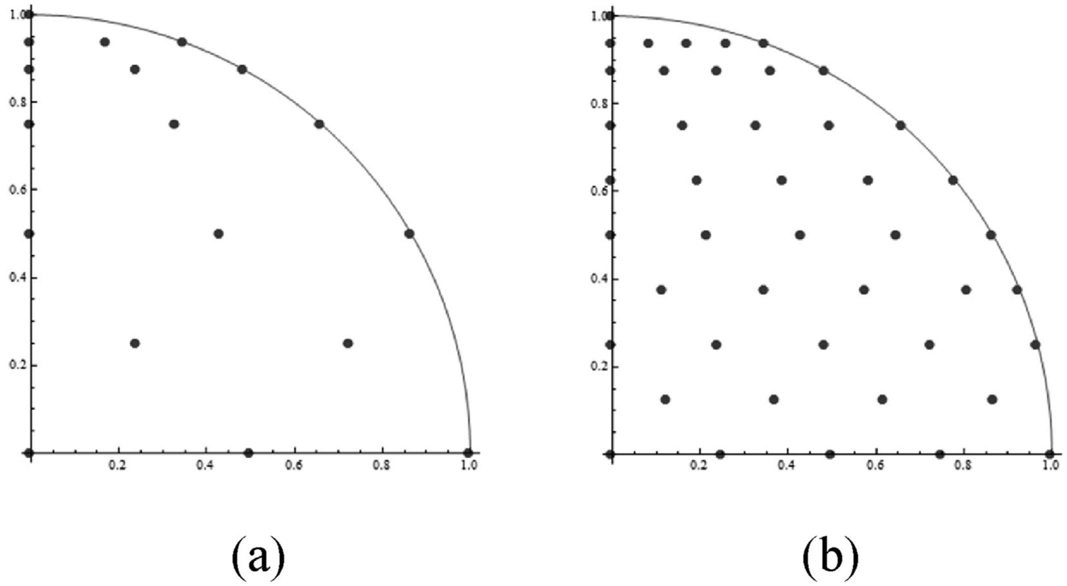

Fig. 13. Adaptive sample point generation for quadrant of a circle. (a) Initial distribution of sample points by using ASP20. (b) Initial distribution of sample points by using ASP48.

can be seen in [30], in which the authors developed an Adaptive Weighted Locally Optimal Projection (AWLOP) operator to distribute the sample points based on the human visual perception.

Mesh optimisation is categorised into three [31], which are adaptivity, smoothing and swapping.

Adaptivity involves mesh refinement at certain regions within the problem domain that require higher precision. The mesh refinement is usually accomplished by replacing coarse mesh with finer mesh through nested grid formation (similar to quadtree mesh) at selected regions, while the rest of domain consist of coarse mesh. The coarse and fine regions are connected by using transition elements. This technique is usually implemented for quadrilateral and hexahedral meshes. Another approach for adaptive mesh optimisation without nested grid formation is presented in [32]. The authors implemented several operations such as edge collapse and face split for the mesh optimisation near transition regions. This approach is similar to swapping.

Smoothing involves relocation of sample points/nodes to improve the mesh quality, without changing the connectivity. This technique can be further categorised to three, which are Laplacian, optimisation-based and physics-based approaches [33]. Laplacian approach improves a mesh by relocating the nodes by arithmetic mean of connected nodes, which was later improved by introducing quality metric and conditional updating [33]. Optimisation approach uses certain algorithms to meet the pre-defined standards that are related to mesh quality. Thus, smoothing technique can be employed to achieve different optimisation goals, depending on the parameter selected as the evaluation standard [31]. For example, the authors in [34] selected triangle average quality as the evaluation standard to achieve optimal mesh, with the help of Particle Swarm Optimisation (PSO) technique. Other examples of evaluation standards are minimum or maximum angle, aspect ratio and distortion metrics. The last approach, that is physics-based approach assumes the mesh as a deformable media and forces are applied to achieve the optimal element shape [33].

Swapping improves mesh quality by changing shape of the elements, through operations such as edge and face swapping. This technique changes the mesh connectivity. Other optimisation works are available in

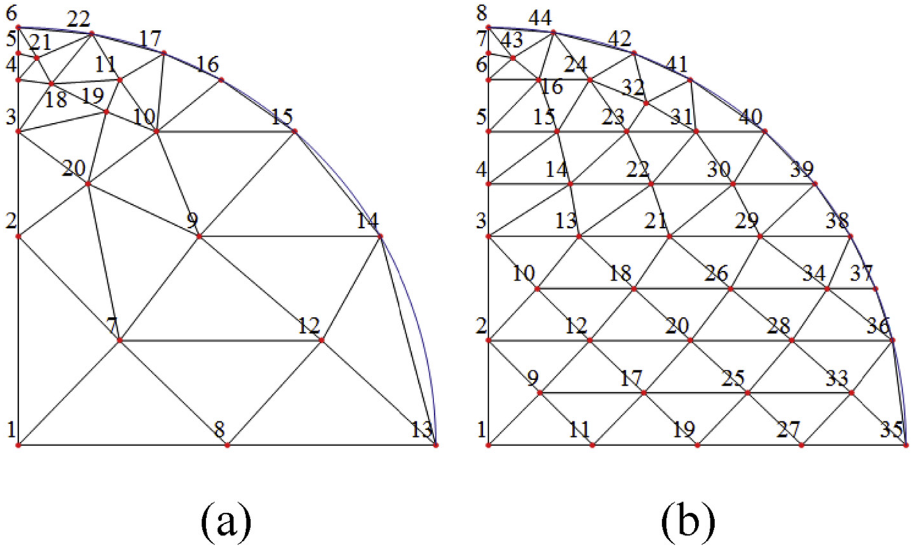

Fig. 14. Adaptive mesh generation for quadrant of a circle. (a) Adaptive mesh by using ASP20. (b) Adaptive mesh by using ASP48.

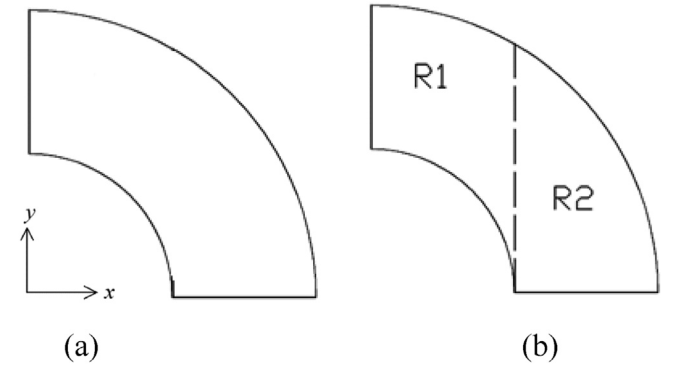

Fig. 15. Decomposition of a problem domain according to Fubini's theorem. (a) domain before decomposition (b) domain after decomposition into 2 subdomains R1 and R2.

the literature. For example in [35], the authors eliminated obtuse triangles by addition and removal of nodes, followed by optimisation-based mesh smoothing while optimisation of a mesh based on aspect ratio is seen in [36]. Optimisation of elements in ANSYS is achieved by inserting or removing nodes and swapping edge/face [37].

The technique presented in this work can be categorised as optimisation-based mesh improvement via point insertion, deletion and relocation. Evaluation standards chosen for the mesh optimisation are the aspect ratio (should be less than 2) and skewness (minimum skewness should be more than 45°). Point insertion and deletion changes the mesh connectivity (similar to swapping), while relocation does not give any impact to the connectivity (similar to smoothing) most of the time. Adaptivity is attained by using different densities of sample points in the reference domains (described in section 2) to form fine and coarse meshes.

2. Design

Sample points for a problem domain can be generated through a mapping technique. The mapping can be accomplished by using generalised equation which was introduced in previous works [2, 3, 4]. The generalised equation for 2 dimensional mapping is recalled here:

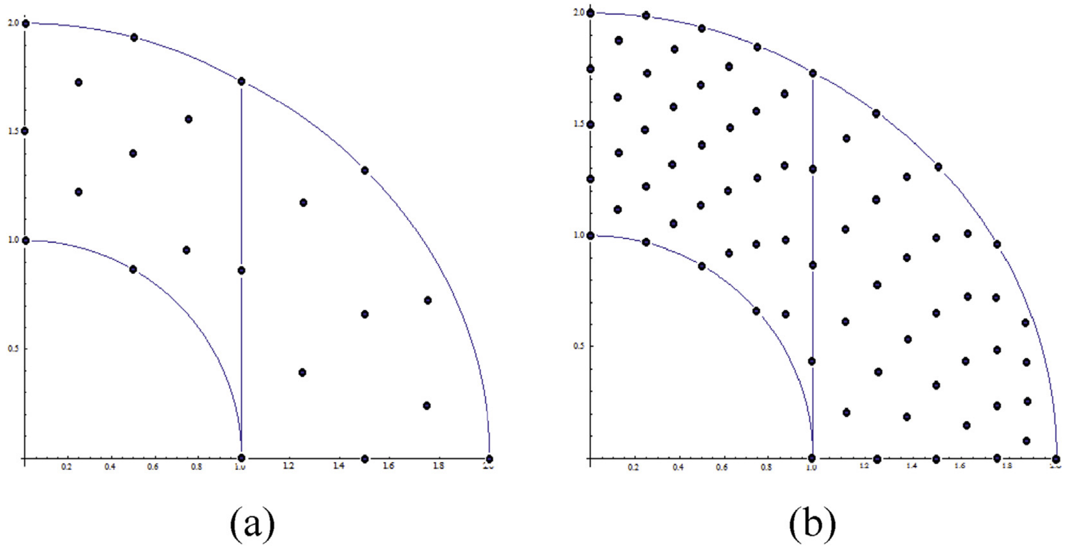

Fig. 16. Generation of sample points. (a) distribution of sample points for SP13 (b) distribution of sample points for SP41.

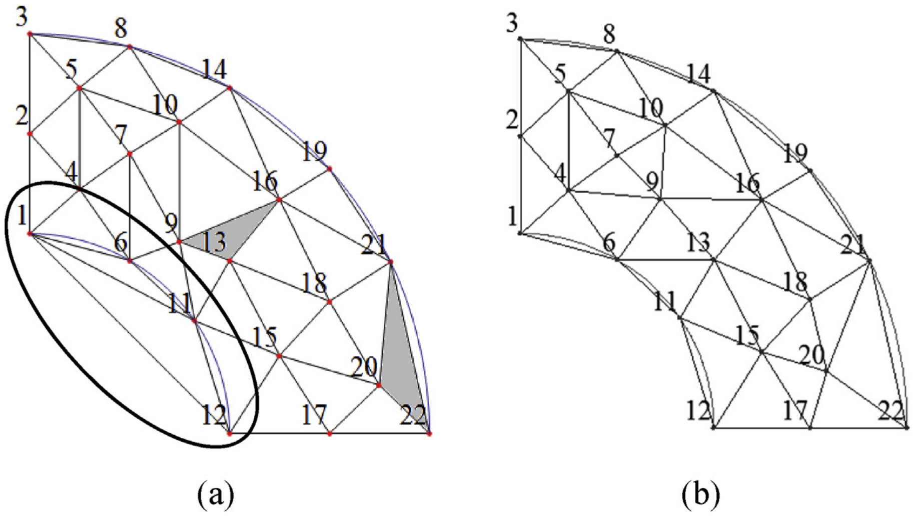

Fig. 17. Coarse mesh optimisation for quadrant of hollow cylinder. (a) initial triangulation for the coarse mesh associated with SP13 (b) final optimised mesh.

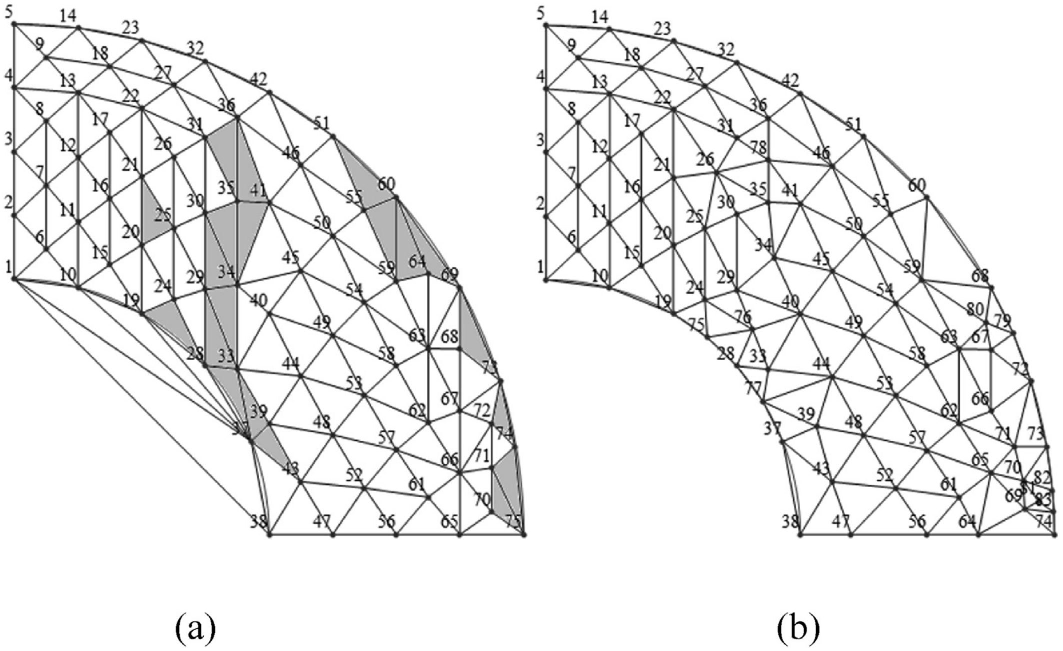

Fig. 18. Fine mesh optimisation for quadrant of hollow cylinder. (a) initial triangulation for the fine mesh associated with SP41 (b) final optimised mesh.

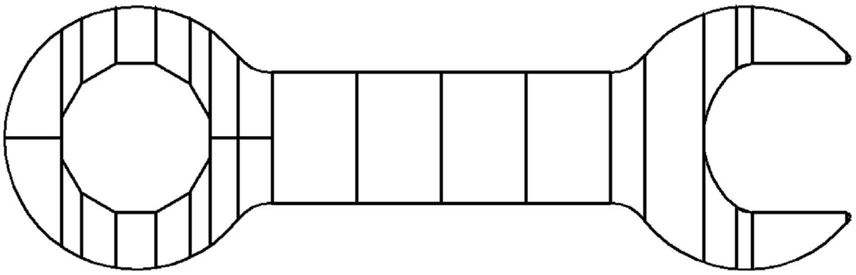

Fig. 19. Decomposition of a wrench domain according to Fubini's theorem.

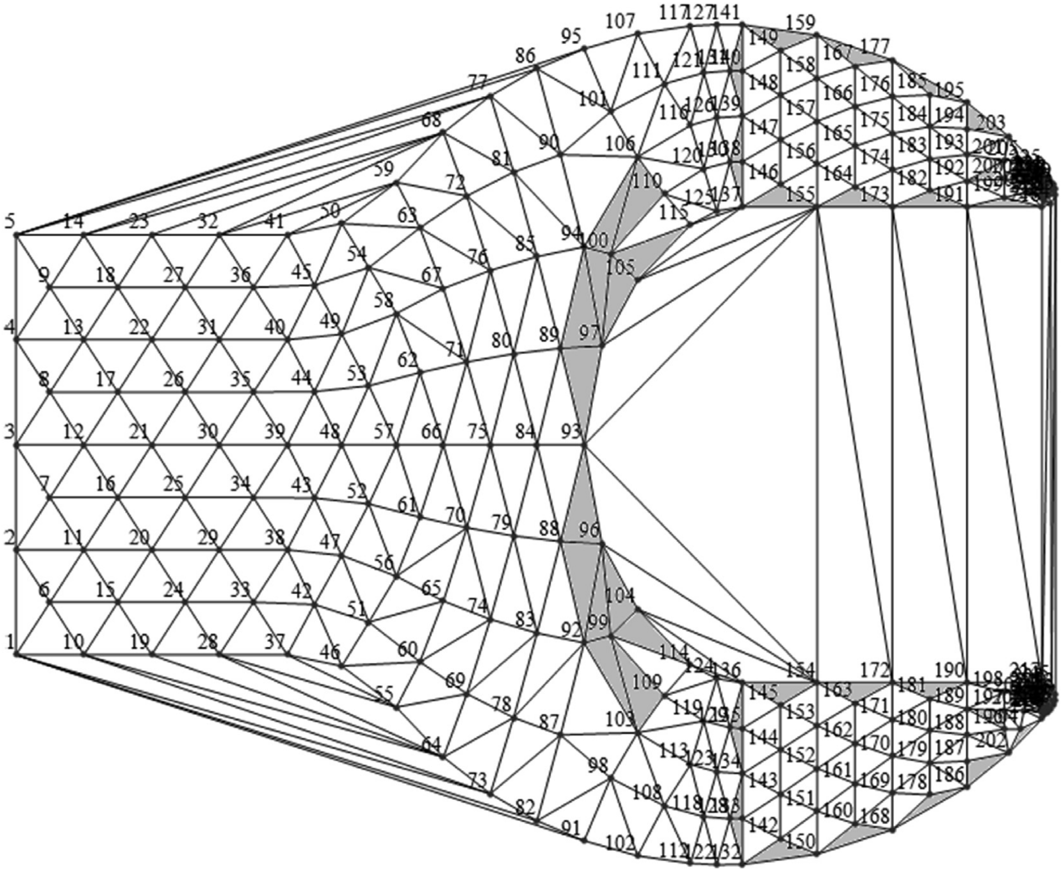

Fig. 20. Initial mesh for the head.

$$\left.\begin{array}{c} I_1 = \displaystyle\int_a^b \int_{r(x)}^{s(x)} f(x,y)\,dydx \\ or \\ I_2 = \displaystyle\int_a^b \int_{r(y)}^{s(y)} f(x,y)\,dxdy \end{array}\right\} = \int_L^U \int_L^U f\left(m_x u + c_x, m_y v + c_y\right) m_x m_y dvdu$$

$$\begin{aligned} &for\, I_1: \\ &m_x = \frac{a-b}{L-U};\quad m_y = \frac{r(m_x u + c_x) - s(m_x u + c_x)}{L-U}; \\ &c_x = \frac{(b\times L)-(a\times U)}{L-U};\quad c_y = \frac{(s(m_x u + c_x)\times L) - (r(m_x u + c_x)\times U)}{L-U} \\ &for\, I_2: \\ &m_x = \frac{r\left(m_y v + c_y\right) - s\left(m_y v + c_y\right)}{L-U};\quad m_y = \frac{a-b}{L-U}; \\ &c_x = \frac{\left(s\left(m_y v + c_y\right)\times L\right) - \left(r\left(m_y v + c_y\right)\times U\right)}{L-U};\quad c_y = \frac{(b\times L)-(a\times U)}{L-U} \end{aligned} \tag{1}$$

The left hand side of the Eq. (1) is Fubini's representation of a problem domain that is bounded by constant lines (vertical or horizontal lines) a and b, and inclined lines or curves described by polynomial functions r and s. The right hand side of the equation represents a reference geometry bounded by constant lines U and L. Mapping of reference geometry to arbitrary domain in physical system can then be achieved by using numerical transformations given by I_1 or I_2. The functions $f(x, y)$ and $f(u, v)$ represent coordinates of entities in physical problem domain and reference domain (sample points), respectively.

In order to perform the mapping, a problem domain needs to be bounded by a, b, r and s, according to the Fubini's theorem. For that reason, the domain needs to be decomposed into several regions which are enclosed by either one of the following:

a. 4 constant lines.
b. 3 constant lines and 1 inclined line or curve.
c. 2 constant lines and 1 inclined line or curve.
d. 2 parallel constant lines and 2 inclined lines or curves.

Examples of domains enclosed by the 4 options given above are shown in Fig. 1.

The next concept is to develop reference domains (bounded by U and L) with sample points (with fixed coordinates) that can be mapped to the physical problem domain by using Eq. (1). Four types of reference domains are presented here, which are bounded by $U = 1$ and $L = -1$ (the domain is a square with coordinate range of -1 to 1). Each of the reference domain contains different number and distribution of reference sample points, as shown in Fig. 2. SP13 and SP41 represent reference domains with 13 and 41 sample points respectively, which are distributed uniformly. The sample points are distributed uniformly within the reference domain, to enable uniform mesh generation. ASP20 and ASP48 represent reference domains with 20 and 48 sample points respectively, which are distributed nonuniformly. ASP20 and ASP48 consist of a greater number of sample points on the upper half of the reference domains compared to their lower half. Nonuniform distribution of sample points enables adaptive meshing, due to the different density of sample points within the domain (region with higher number of sample points leads to finer/denser mesh).

Once the problem domain is decomposed and the sample points are mapped to the problem domain, triangulation can be initiated by employing the conventional Delaunay method, by using the function "*DelaunayTriangulation*". The next section describes the optimisation technique for the elements.

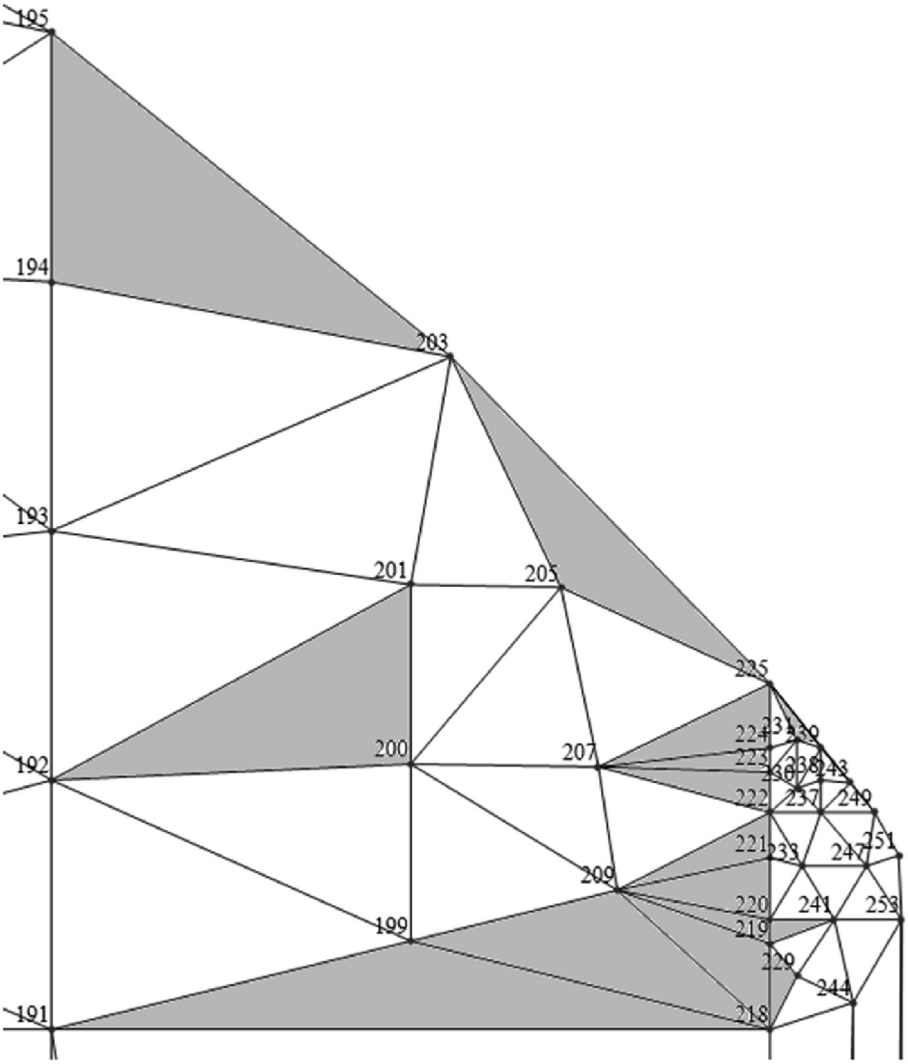

Fig. 21. Initial mesh for the tip (close up view).

3. Methodology

Details of the techniques used for mesh optimisation are explained in this section. Mesh quality is defined first and a framework for optimisation is introduced. Subsequent subsections explain the methodologies used to optimise the elements in a mesh.

3.1. *Mesh quality and a framework for optimisation*

Quality of a triangular element can be defined based on two parameters that are the aspect ratio and skewness. These two parameters are selected among other parameters to determine the validity of an element, since they play important role in providing reliable results. A valid triangular element holds aspect ratio of less than 2 and minimum skewness of more than 45°. Aspect ratio (AR) is calculated based on the formulas below:

$$AR = \frac{abc}{8(s-a)(s-b)(s-c)}, \tag{2}$$

$$s = \frac{1}{2}(a+b+c) \tag{3}$$

a, b, and c represent length of each side of a triangular element. The skewness is determined by calculating the lowest angle formed by intersection of two lines within the triangle. The first line is drawn such a way that it connects one of the triangular nodes to the midpoint of opposite side. The second line connects midpoints of the other two sides. Since there are 3 nodes for a triangular element, therefore total of three combinations of such intersection can be obtained. These combinations are shown in Fig. 3. Skewness for a particular triangular element is represented by the smallest angle which is formed by the intersections in the three configurations shown in Fig. 3.

Optimisation of the triangular elements is done by addition, removal and relocation of sample points (or nodes) until all the elements meet the desired quality (aspect ratio of less than 2 and minimum skewness of more than 45°). Corresponding Mathematica functions to carry out the three tasks above are “*Append*”, “*Delete*” and “*ReplacePart*”, respectively. Framework for optimisation is shown in Fig. 4.

3.2. *Initial mesh creation*

Initial mesh is created through conventional Delaunay method, by using the Mathematica function “*DelaunayTriangulation*”. This function generates triangular meshes according to the distribution of sample points in the domain. There is possibility for redundant triangles to be formed outside the intended problem domain. These redundant triangles can be removed based on two approaches, which are either by using integration limit functions a, b r and s in Eq. (1) or by using Mathematica

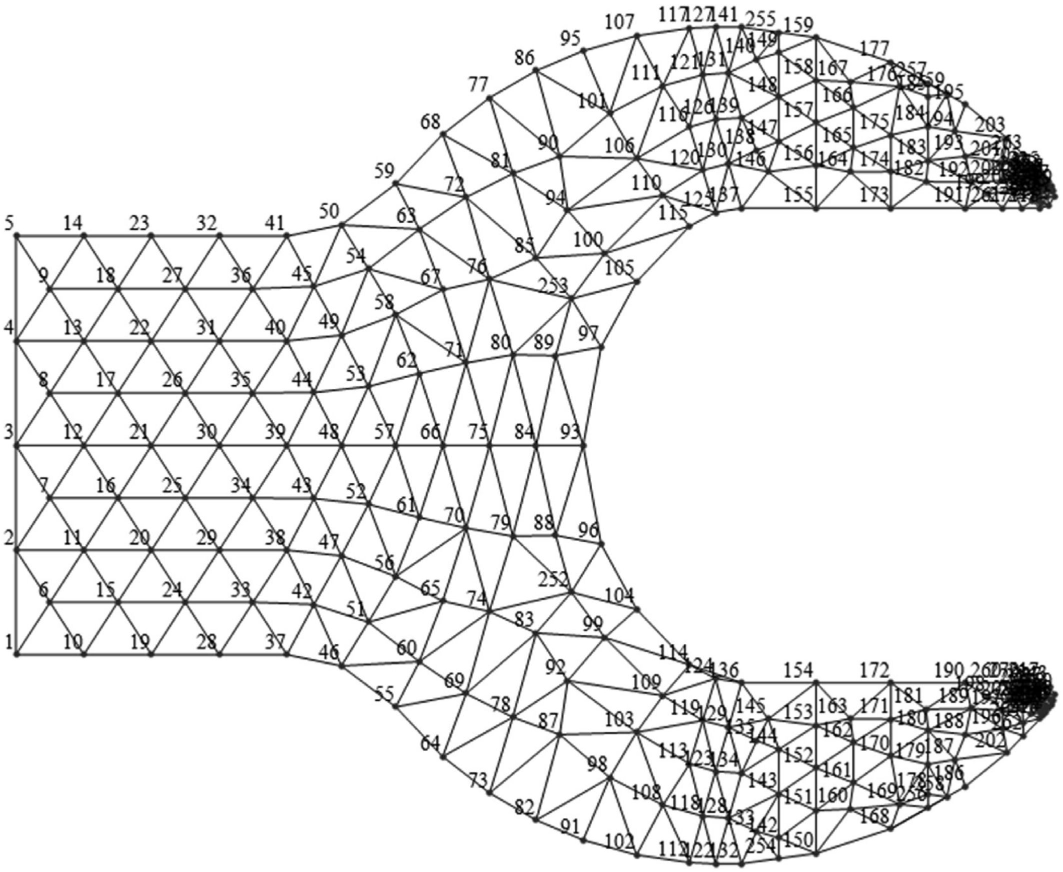

Fig. 22. Optimised mesh for the head.

functions "*RegionMember*" and "*Polygon*".

The first approach requires parametric equations of the boundaries. The outer redundant triangles can be detected (and later removed) by using the following criteria:

$$For\ I_1:\quad a \le x_c \le b\ and\ r(x_c) \le y_c \le s(x_c)$$

$$For\ I_2:\quad a \le y_c \le b\ and\ r(y_c) \le x_c \le s(y_c)$$

x_c and y_c are coordinates of centre of a triangular element. If coordinate of the centre of a triangular element does not fall within the given range, then it is identified as a redundant triangle.

The second approach is based on the boundary discretisation. This approach can be used when equations of the boundaries (such as free formed curves) are not available. Free formed curves can be approximated by using Bezier curves by altering position of the control points until desired accuracy is obtained (Mathematica function for Bezier curve is "*BezierCurve*"). Coordinates of the control points are later used to compute sample points on the curves (boundary discretisation). The problem domain is then classified as a polygon by assigning the coordinates of boundary sample points to the Mathematica function "*Polygon*". External redundant triangles are detected when centre of the triangular elements are not situated within the polygonalised problem domain (corresponding Mathematica function is "*RegionMember*").

3.3. Valid region and node relocation

Next, the Mathematica algorithm identifies all the flawed elements in the mesh, according to the aspect ratio and skewness. If a flawed element is located at the interior of the problem domain, then the element can be adjusted to meet the required quality by moving one of the three nodes of the triangular element. Therefore, a new approach is introduced here to identify the ideal position to relocate one of the nodes of a flawed element such a way that it would satisfy the required criteria.

This is done by defining a valid region for the node relocation. An edge of a triangular element (edge AB) with length of 1 unit is taken as reference to define the valid region as shown in Fig. 5(a). Two isosceles triangles (labelled as ABC and ABD) are formed, with minimum skewness of 46° for triangle ABC and 45° for triangle ABD. Corresponding aspect ratios are 1.70 and 1.75, respectively. Any lower positioning of point C (vertically) or any upper positioning of point D (vertically) will cause the minimum skewness to be lower than 45° for the triangle ABC or ABD, respectively. Two constant horizontal lines are drawn (that pass-through points C and D) as upper and lower boundaries for the reference valid region. Afterward, two inclined lines that make an angle of 45° to the horizontal line are drawn, to indicate the boundary for minimum skewness of 45°. These boundaries are shown by dashed lines in Fig. 5(a).

Sample points are randomly placed inside the region bounded by the four linear lines and later individually connected to points A and B to form triangles. Minimum skewness of these triangles is checked and the corresponding sample points are marked as a cross or circle, to indicate suitability of the particular locations of the sample points as ideal relocation for the flawed element's third node. Fig. 5(b) shows suitability of some of sample points within region bounded by the four linear lines. Circle indicates suitable position for relocation, while cross indicates invalid position. The final reference valid region as shown in Fig. 5(c) is obtained by connecting the circles in Fig. 5(b) by using linear lines and merge with the initial boundaries in Fig. 5(a). The reference valid region shown in Fig. 5(c) can be implemented for any arbitrary edges through scaling, rotation and displacement of this reference region.

Therefore, in order to determine the ideal position for a node in physical system, a polygonal structure is first formed based on the triangular elements that share the particular node. The polygonal

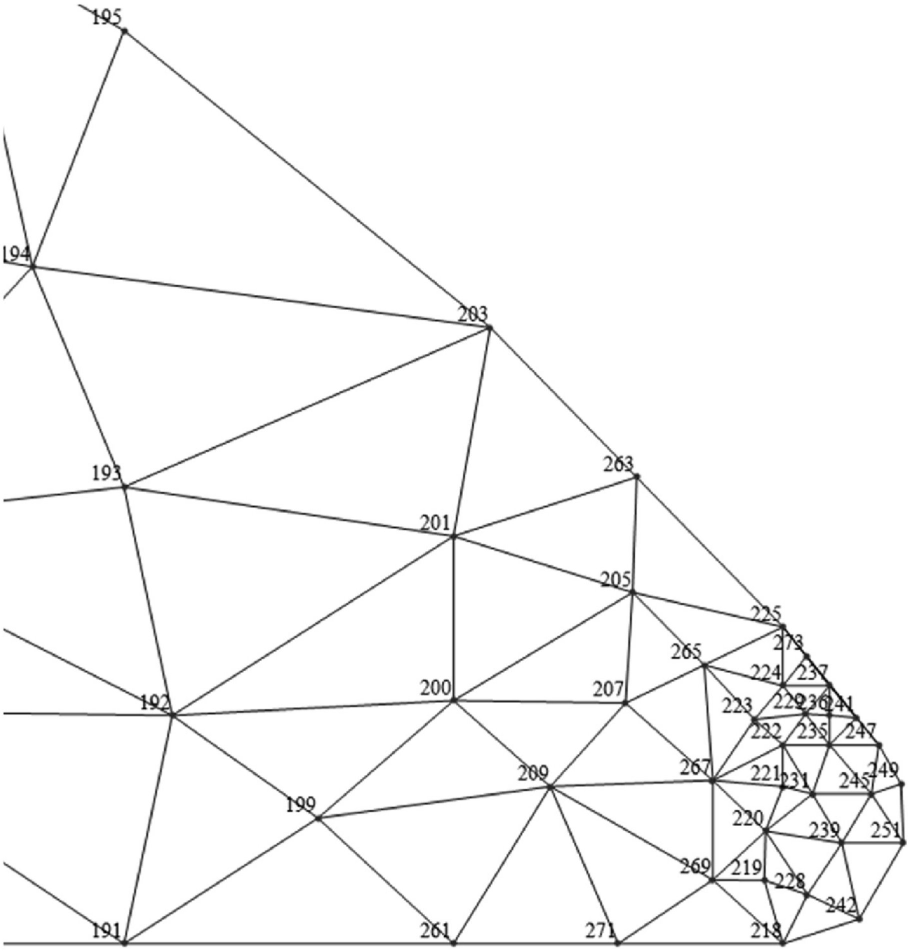

Fig. 23. Optimised mesh for the tip (close up view).

structure is formed by connecting edges (that are not connected to the particular node) of these triangular elements. The ideal position for the node will be anywhere within the area of valid region that is common for all the edges of the polygonal structure. The common area within the valid regions of all the edges of the polygonal structure is determined by generating random sample points within the valid region corresponding to each of the polygonal edges. Sample points which are common or present in all the valid regions corresponding to all the polygonal edges would be the ideal positions for the node. An example is shown in Fig. 6. Flawed element (AR = 1.73 and minimum skewness = 41.73°) is the triangular element with nodes labelled as 9-13-16 (node sequence is according to local node numbering) in physical system as shown in Fig. 6(a). The algorithm selects the first internal local node of the element for evaluation, by default. The algorithm moves on to the next internal local node of the element for evaluation, if optimal position is not found for the former node. Boundary nodes are fixed and cannot be relocated. In this example, the first local internal node 9 is selected for relocation, and therefore a polygonal structure is formed based on the node 9 as shown in Fig. 6(b). Valid region for each edge of the polygonal domain is generated by scaling, rotation and displacement of the reference valid region (Fig. 5(c)). Example of valid regions for edges 7–10 and 13–16 are shown in Fig. 6(c). Similar regions are generated for the remaining edges of the polygon and common area is determined, as shown by the grey area in Fig. 6(c). Relocation of node 9 within the final valid region (greyed area) is shown in Fig. 6(c) (labelled as 9′). It can be seen that node 9′ makes new connection with node 4, thus changes the connectivity. Optimised mesh is shown in Fig. 6(d).

3.4. *Node insertion, relocation and deletion*

However, not all the elements can be optimised merely by relocating one of the element nodes based on this approach. This is due to surrounding elements that share the same node would be affected by the relocation of the node. When optimisation is not possible by node relocation, an addition of a sample point/node can be done to the mesh. This insertion can be done to the middle of the longest edge of the flawed element. An example is shown in Fig. 7.

The greyed flawed element 106-126-139 in Fig. 7(a) is a boundary element with aspect ratio of 3.24 (less than 100). Optimal position for the internal node 126 is not found within the polygonal structure formed by nodes: 106-139-138-137-125-116-106. Therefore a mid-node (node 291) is added to the longest edge of the flawed element, which is edge 106–139 as shown in Fig. 7(b). Inclusion of the mid-node causes a new flawed element 126-138-291 to be formed (greyed in Figure (b)). The first internal local node for the new flawed element is 126, followed by node 138. For this case, optimal position for the internal node 126 is available within the polygonal structure formed by the nodes: 291-138-137-125-116-106-291. Node 126 is then relocated to form optimised elements as shown in Fig. 7(c). A particular internal node can be deleted, if three continual additions of mid-nodes did not yield a valid element.

3.5. *Relocation of boundary nodes*

If the flawed element is a boundary element with aspect ratio of greater than 100, then the particular element is initially skipped since addition of mid-nodes can lead to unresolved loop/iteration. Once the

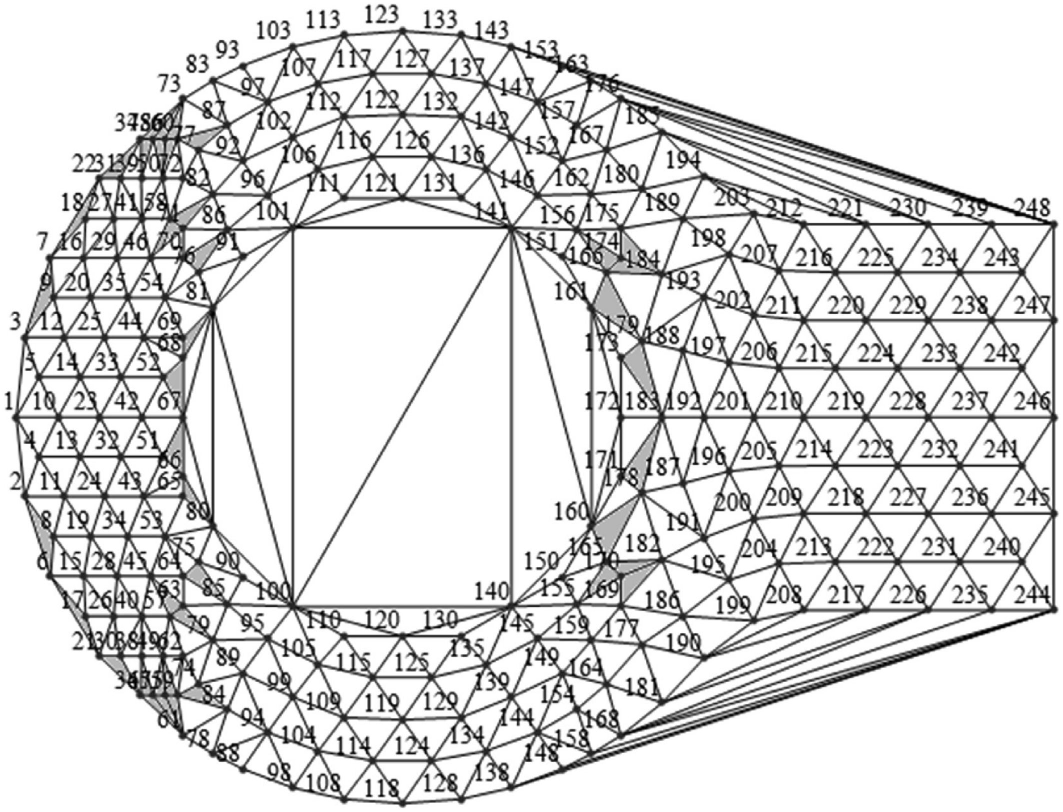

Fig. 24. Initial mesh for the socket.

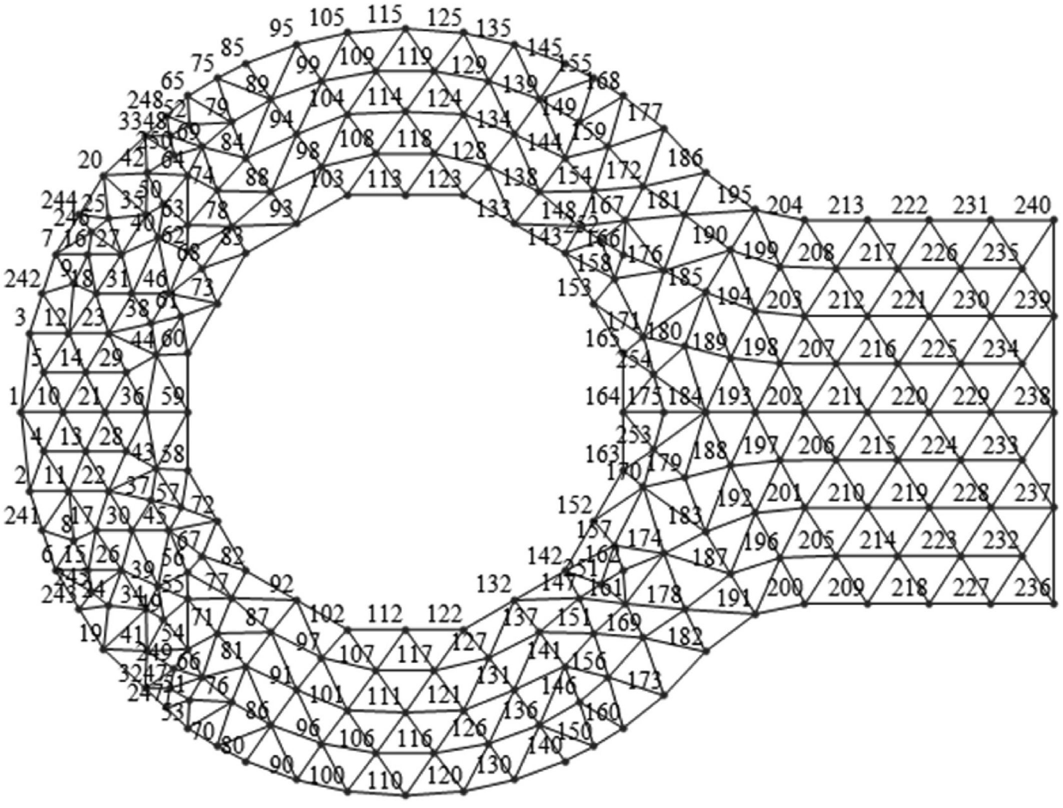

Fig. 25. Optimised mesh for the socket.

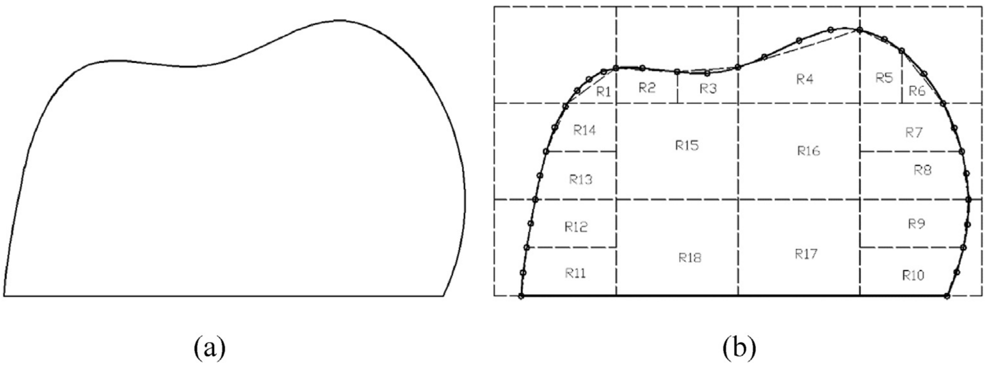

Fig. 26. Preparation of a domain for mesh generation (a) domain with free formed curves and a linear boundary. (b) partitioning of the domain.

rest of the elements have been optimised, the skipped elements are revisited. These flawed boundary elements are dealt with by moving the element nodes to the domain boundary. Boundary nodes of a mesh are identified by first finding element edges that are not shared by any other element (boundary edges). Nodes of these edges are then classified as boundary nodes of the mesh system. There are two approaches to move the boundary nodes of a mesh to the domain boundary.

The first approach is to find a point/location on the boundary of the problem domain which yields minimum distance between a particular boundary node of a mesh system to one of the integration limit functions *a*, *b r* and *s* in Eq. (1). The boundary node of the mesh system can then be moved to the nearest point on the boundary. An example is shown in Fig. 8.

Node 3 of the mesh in Fig. 8(a) is inaccurately positioned and leads to poor boundary representation. This error is eliminated by moving the node 3 to the boundary, as shown in Fig. 8(b). Correct location of node 3 on the domain boundary is obtained by calculating minimum distance from the original node 3 (node 3 in Fig. 8(a)) to the lines/curves enclosing the domain, that is the integration limits of Eq. (1): *a*, *b*, *r* and *s*. Total of four minimum distances and corresponding points on the lines/curves enclosing the domain would be calculated (due to the four integration limits). These four points are shown in Fig. 8(a) by using dark filled circles, for enclosure based on I_1 in Eq. (1). Minimum of these four distances and the corresponding point would be the correct location for node 3 on the boundary of the domain.

The other approach to move the boundary nodes of the mesh to the domain boundary is by using minimum distance between the boundary nodes of a mesh system to the discretised boundary sample points which are obtained through Bezier curves (function "*BezierCurve*" as described earlier). This approach can be used when equations of the boundaries (such as free formed curves) are not available. A particular boundary node of a mesh system is then moved to the location of a boundary sample point which is nearest to it. An example is shown in Fig. 9.

Fig. 9 shows similar problem domain as the previous example, which is quadrant of a circle. Assuming that the equation of the arc is unknown, the arc is discretised by using Bezier curve and the boundary sample points are shown in Fig. 9(a). The boundary node 3 is moved to the domain boundary (by replacing the respective boundary sample point) based on the minimum distance obtained. Fig. 9(b) shows the final optimised mesh.

Once the boundary nodes are moved to the domain boundary, the algorithm checks for any remaining flawed elements in the mesh. This step is crucial in order to make minor adjustments if flaws occur due to the displacement of nodes to the domain boundary. The optimisation is terminated once there is no more flawed element in the mesh system.

4. Results and discussion

Examples of mesh generation by using the proposed techniques are presented here. Total of four examples are shown, involving geometries with linear lines and curves. The optimisation is carried out based on the framework in Fig. 4.

4.1. *Example 1: meshing of quadrant of a circle*

The problem domain is bounded by two constant lines and one curve, as shown in Fig. 10(a). The Fubini's criterion is satisfied and domain decomposition is not required. The sample points are generated by letting $U = 1$, $L = -1$, $a = 0$, $b = 1$, $r(y) = 0$ and $s(y) = \sqrt{1-y^2}$ into Eq. (1). Distribution of sample points by using SP13 and SP41 are shown in Fig. 10 (b) and (c), respectively. It can be seen that the sample points are uniformly distributed within the domain. Resultant triangulations are shown in Figs. 11 and 12, respectively. Flaws within the mesh are either circled or highlighted by grey colour.

Initial mesh using SP13 is shown in Fig. 11(a). Redundant triangles are not formed. The mesh consists of a flawed element 10-4-3 and node 3 is found to be not on the problem boundary. Flawed element 10-4-3 is optimised by relocating internal node 4. No further flawed element is detected in the mesh. Node 3 is later moved to the domain boundary based on the first approach described in previous section. Final optimised mesh is shown in Fig. 11(b).

As for fine mesh using SP41, initial triangulation produces highly skewed elements near the top and flawed elements at the curved boundary, as shown in Fig. 12(a). Triangular elements with aspect ratio of greater than 100 are skipped and the rest of the elements are optimised as shown in Fig. 12(b). The skipped elements (circled in Fig. 12(b)) are later revisited and dealt with by moving the boundary nodes of the mesh (nodes 13, 38 and 44) onto the domain boundary. It can be also seen that movement of nodes 39 and 40 onto the domain boundary improves accuracy of the geometry representation. These two nodes are mid-nodes which were added to the mesh during optimisation.

Meshes in Fig. 12 show more elements are required at the top in order to capture the narrowing shape of the domain. Therefore, ASP20 and ASP48 can be used to adapt to the change. These two reference domains consist of sample points which are distributed non-uniformly. More sample points are located on the upper half of the domain compared to the lower half. Fig. 13 shows adaptive sample point generation for the quadrant of a circle by using ASP20 and ASP48. Resultant optimised meshes are shown in Fig. 14.

4.2. *Example 2: meshing of quadrant of hollow cylinder*

Quadrant of a hollow cylinder as shown in Fig. 15(a) is taken as the problem domain to be meshed. The domain is enclosed by two constant lines and 2 curves, which does not fulfil the requirements of Fubini's theorem as mentioned in previous section 2. Therefore, the domain is decomposed into two regions, named R1 and R2 as shown in Fig. 15(b).

Sample points for the problem domain are generated by mapping reference sample points to R1 and R2, independently. Resultant

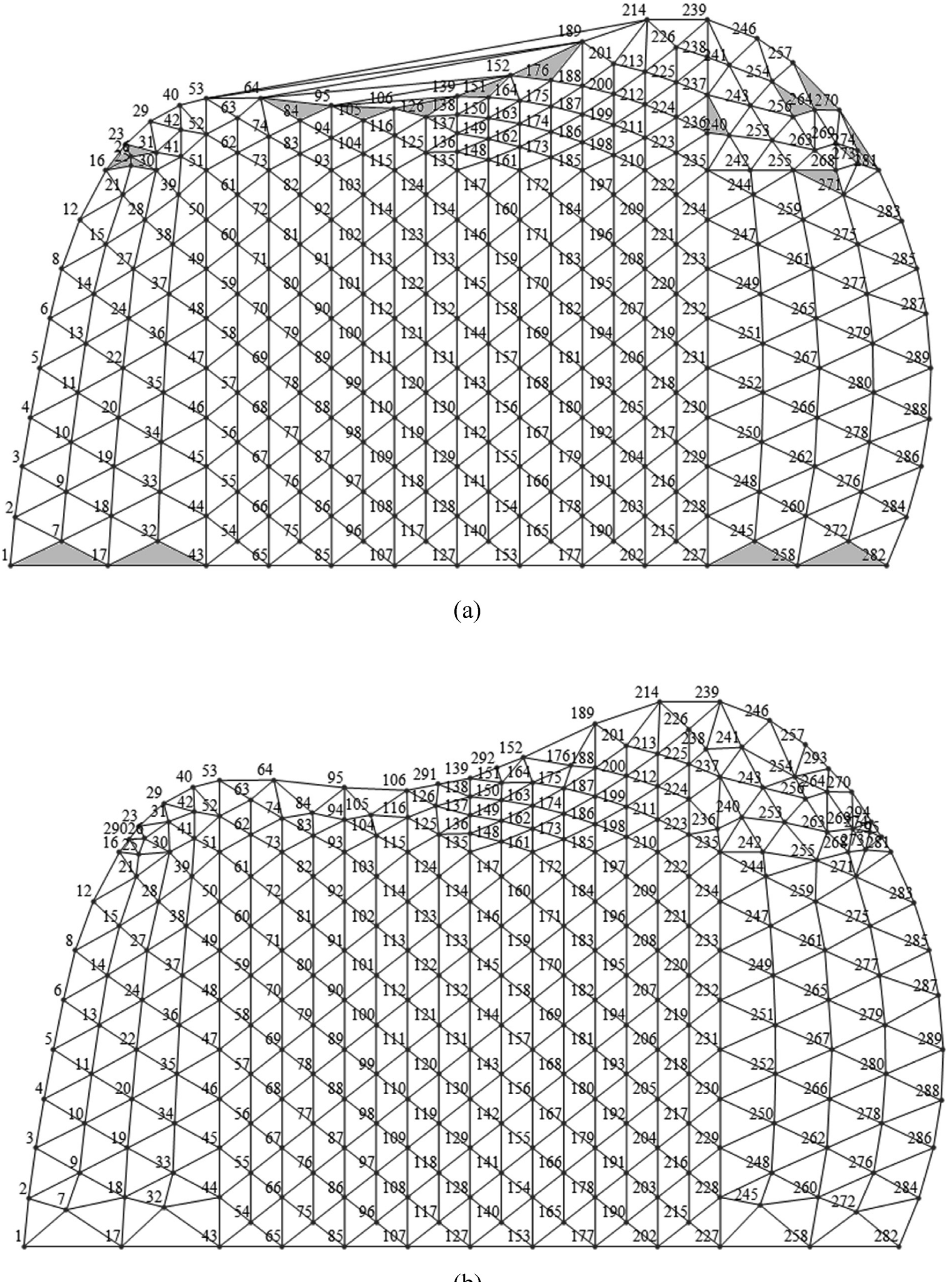

Fig. 27. Meshing of free formed curves (a) initial mesh. (b) optimised mesh.

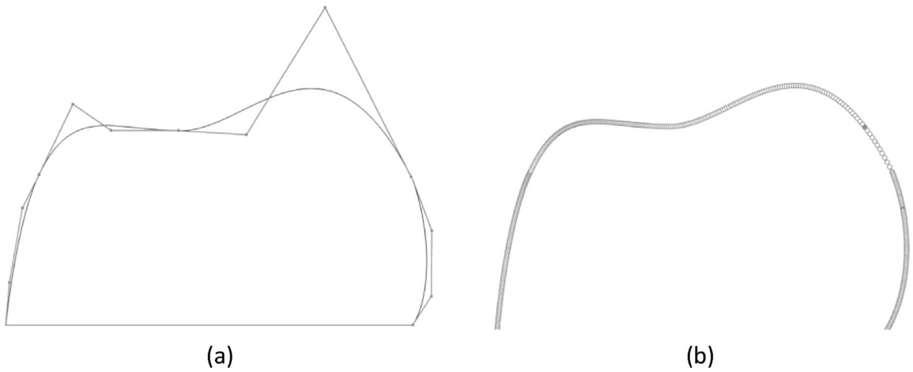

Fig. 28. Parameterisation of free formed curves (a) approximation of the curves by using Bezier curves. (b) resultant discretisation based on Bezier curves.

distribution of the sample points for the entire domain by using SP13 and SP41 (separately) are as shown in Fig. 16(a) and (b), respectively. Figs. 17(a) and 18(a) show the triangulation of the mapped sample points and elements of poor quality are greyed. Some redundant triangles which are generated during the initial mesh creation are deleted during optimisation. The coarse mesh is optimised by relocating nodes 9 and 20. Optimised mesh is shown in Fig. 17(b). The optimised fine mesh as shown in Fig. 18(b) is obtained through relocation of 10 nodes and addition of 9 nodes. The mesh is further improved by moving the boundary nodes of the mesh system to the domain boundary.

4.3. *Example 3: meshing of a wrench*

Problem domain is first decomposed into several regions according to the Fubini's theorem as shown in Fig. 19. These regions are later mapped with sample points via SP13 and SP41. Both SP13 and SP41 are used together in order to obtain uniform distribution of the sample points. The tip of the head needs to be triangulated by using small elements with higher density in order to accurately represent the geometry accurately. Initial mesh with flawed elements (greyed) for the head is shown in Fig. 20. Close up view of the tip is shown in Fig. 21. Corresponding optimised meshes are shown in Figs. 22 and 23, respectively. Similarly, initial and optimised meshes for the socket are shown in Figs. 24 and 25, respectively.

4.4. *Example 4: meshing of a free formed curve*

In this example, a domain with free formed curves and linear side is used to demonstrate application of the proposed techniques. The domain is as shown in Fig. 26(a). Parametric equations for the curves are not available by default. Quadrangles are formed at the background to decompose the problem domain according to the Fubini's requirements, as shown in Fig. 26(b). This technique is similar to initialisation tree and quadtree-like background mesh (techniques described in sections 1.1.6 and 1.1.10). There are two ways to generate sample points for the boundaries. One is by assuming the curves as linear lines and later the sample points are either moved or added to the boundaries, or parametric equations for the curve can be obtained by using Bezier curves and data fitting. The parametric equations can then be used in Eq. (1) to place the sample points onto the curve boundaries. The first method is presented here, while the second method is briefly described at the end of this section. Initial mesh generation is shown in Fig. 27(a) and corresponding optimised mesh is shown in Fig. 27(b).

Parameterisation of the free formed curves can be done by using Bezier curves and data fitting. Free formed curves can be approximated by using Bezier curves by altering position of the control points until desired accuracy is obtained, as shown in Fig. 28(a) (Mathematica function for Bezier curve is "*BezierCurve*"). Total of 4 Bezier curves (each with four control points) are used to represent the curves. Coordinates of the control points are later used to compute sample points on the curves (discretisation). Resultant discretisation (sample points on the curves) is shown in Fig. 28(b). Coordinates of these boundary sample points can then be parameterised by using nonlinear data fitting function in Mathematica, called "*NonlinearModelFit*".

5. Conclusions

New techniques for generation of sample points and optimisation of resultant meshes have been successfully tested and validated in this work. The techniques presented in this work have advantages and disadvantages as compared to other existing techniques. Advantages of the techniques proposed in this work are:

a. The elements in the mesh is guaranteed to be within the specified range (aspect ratio of less than 2 and minimum skewness of more than 45°), since optimisation is carried out onto the triangular elements directly and not onto the distribution of the sample points.
b. Sample points can be located directly onto the domain boundaries, with the help of surface parameterisation.
c. Modelling and execution of Eq. (1) in Mathematica is simpler and faster, since the mapping involves fully numerical algorithms (symbolic manipulation has been eliminated entirely) [4].
d. Complex density functions are not required, since adaptive mesh generation can be formed through background octree mesh or by altering distribution of sample points in the reference domain (such as ASP20 and ASP48).
e. Complex optimisation algorithms such as molecular dynamics, particle dynamics, Monte Carlo and other physics based optimisations are avoided.

Disadvantages of the techniques are:

a. Iterative optimisation.
b. Requires partitioning of the problem domain based on Fubini's theorem.

Further work can be done based on the following:

a. Distribution and number of sample points in the reference domains can be altered to obtain different results.
b. Orientation of ASP20 and ASP48 can be changed through rotations (multiples of 90°), in order to obtain adaptive mesh at different regions within a domain.

Declarations

Author contribution statement

Logah Perumal: Conceived and designed the experiments; Performed the experiments; Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data; Wrote the paper.

Funding statement

This work was supported by the Research Management Centre (RMC) of Multimedia University, Malaysia (Grant nos. MMUI/130070 and MMUI/160047), which enabled purchase of required software and equipment for this work.

Competing interest statement

The authors declare no conflict of interest.

Additional information

No additional information is available for this paper.

References

[1] M.D. Berg, O. Cheong, M.V. Kreveld, M. Overmars, Computational Geometry: Algorithms and Applications, Springer, Berlin, 2008.

[2] L. Perumal, C. Tso, L.T. Leng, Analysis of thin plates with holes by using exact geometrical representation within XFEM, J. Adv. Res. 7 (3) (2016) 445–452.

[3] L. Perumal, Integration techniques for two dimensional domains, Int. J. Renew. Energy Technol. 03 (07) (2014) 487–494.

[4] L. Perumal, T.T. Mon, Generalized equations for numerical integration over two dimensional domains using quadrature rules, Integ. Math. Theor. Appl. 03 (04) (2012) 333–346.

[5] A. Lagae, P. Dutré, A comparison of methods for generating Poisson disk distributions, Comput. Graph. Forum 27 (1) (2008) 114–129.

[6] Y. Zhou, H. Huang, L. Wei, R. Wang, Point sampling with general noise spectrum, ACM Trans. Graph. 31 (4) (2012) 1–11.

[7] S.J. Owen, A survey of unstructured mesh generation technology, in: Proceedings 7th International Meshing Roundtable, 1998, October, pp. 239–267.

[8] S. Kenji, C.G. David, Bubble mesh: automated triangular meshing of non-manifold geometry by sphere packing, ACM Symp. Solid Model. Appl. (1995) 409–419.

[9] Q.S. Wang, J. Ye, H. Wu, B.Q. Gao, P. Shepherd, A triangular grid generation and optimization framework for the design of free-form grid shells, Comput. Aided Des. 113 (2019) 96–113.

[10] A. Zheleznyakova, Molecular dynamics-based triangulation algorithm of free-form parametric surfaces for computer-aided engineering, Comput. Phys. Commun. 190 (2015) 1–14.

[11] P.W. Fackler, Physics-Based Point Placement by Particle Dynamics Simulation (Master thesis), University of Tennessee, 2013.

[12] J. Peyrot, F. Payan, M. Antonini, From stereoscopic images to semi-regular meshes, Signal Process. Image Commun. 40 (2016) 97–110.

[13] H. Notsu, D. Ueyama, M. Yamaguchi, A self-organized mesh generator using pattern formation in a reaction–diffusion system, Appl. Math. Lett. 26 (2) (2013) 201–206.

[14] C.J. Murphy, Nonlinear Finite Element Modeling of Cellular Materials under Dynamic Loading and Comparison to Experiments (Open Access Master's Theses. Paper 806), University of Rhode Island, 2016, https://digitalcommons.uri.edu/theses/806.

[15] L. Fu, L. Han, X.Y. Hu, N.A. Adams, An isotropic unstructured mesh generation method based on a fluid relaxation analogy, Comput. Methods Appl. Mech. Eng. 350 (2019) 396–431.

[16] L. Chen, H. Wei, M. Wen, An interface-fitted mesh generator and virtual element methods for elliptic interface problems, J. Comput. Phys. 334 (2017) 327–348.

[17] C. Donohue, V. Ostromoukhov, Fast generation of importance-sampled point sets with associated Delaunay triangulation, in: GraphiCon'2007, 2007. Russia, Moscow, June 23-27, 2007.

[18] J. Chen, Z. Liu, Y. Zheng, P. Zheng, J. Zheng, Z. Xiao, C. Yu, Automatic sizing functions for 3D unstructured mesh generation, Proc. Eng. 203 (2017) 245–257.

[19] B. Gao, T. Li, T. Ma, J. Ye, J. Becque, I. Hajirasouliha, A practical grid generation procedure for the design of free-form structures, Comput. Struct. 196 (2018) 292–310.

[20] Q. Wang, B. Gao, T. Li, H. Wu, J. Kan, B. Hu, A triangular mesh generator over free-form surfaces for architectural design, Autom. ConStruct. 93 (2018) 280–292.

[21] About Working with Point Clouds, 2019. Retrieved from, https://knowledge.autodesk.com/support/autocad/getting-started/caas/CloudHelp/cloudhelp/2018/ENU/AutoCAD-Core/files/GUID-C0C610D0-9784-4E87-A857-F17F1F7FEEBE-htm.html.

[22] ArcGIS Pro, 2019. Retrieved from, https://pro.arcgis.com/en/pro-app/.

[23] Civil 3D, Civil Engineering Software, 2019. Retrieved from, https://www.autodesk.com/products/civil-3d/overview?_ga=2.149668598.1321374229.1560405582-1439067448.1556005056.

[24] Q. Du, M.D. Gunzburger, L. Ju, Constrained centroidal Voronoi tessellations for surfaces, SIAM J. Sci. Comput. 24 (5) (2003) 1488–1506.

[25] Z. Zhong, J. Hua, Kernel-based adaptive sampling for image reconstruction and meshing, Comput. Aided Geomet. Des. 43 (2016) 68–81.

[26] R. Fattal, Blue-noise point sampling using kernel density model, in: ACM SIGGRAPH 2011 Papers on - SIGGRAPH 11, 2011.

[27] P. Jiang, Y. Zhang, Q. Zhou, L. Shu, A sequential sampling strategy for Kriging metamodel based on Delaunay triangulation and TOPSIS, in: 17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 2016.

[28] W. Gong, Y. Liu, K. Tang, T. Wu, Approximate Delaunay mesh reconstruction and quality estimation from point samples, J. Comput. Appl. Math. 274 (2015) 23–34.

[29] P. Persson, G. Strang, A simple mesh generator in MATLAB, SIAM Rev. 46 (2) (2004) 329–345.

[30] L. Xing, X. Zhang, C.C. Wang, K. Hui, Highly parallel algorithms for visual-perception-guided surface remeshing, IEEE Comp. Graph. Appl. 34 (1) (2014) 52–64.

[31] J. Park, S.M. Shontz, An alternating mesh quality metric scheme for efficient mesh quality improvement, Proc. Comp. Sci. 4 (2011) 292–301.

[32] S. Chalasani, D. Thompson, B. Soni, Topological adaptivity for mesh quality improvement, Num. Grid Gener. Comput. Field Simul. (2002) 107–116.

[33] R. Durand, B. Pantoja-Rosero, V. Oliveira, A general mesh smoothing method for finite elements, Finite Elem. Anal. Des. 158 (2019) 17–30.

[34] W. Wang, L. Duan, Y. Bai, H. Wang, H. Shao, S. Zhong, A triangle mesh standardization method based on particle Swarm optimization, PLoS One 11 (8) (2016).

[35] D. Khan, D. Yan, Y. Wang, K. Hu, J. Ye, X. Zhang, High-quality 2D mesh generation without obtuse and small angles, Comput. Math. Appl. 75 (2) (2018) 582–595.

[36] X. Huang, D. Xu, Aspect-ratio based triangular mesh smoothing, in: ACM SIGGRAPH 2017 Posters on - SIGGRAPH 17, 2017.

[37] A.C. Scott, R.T. Joseph, L.S. Matthew, An approach to combined laplacian and optimization-based smoothing for triangular, quadrilateral, and quad-dominant meshes, in: 7th International Meshing Roundtable, 1998, pp. 479–494.