



Nonlinear graded Delaunay meshes



David A. Field^{*,1}, María-Cecilia Rivara

Department of Computer Science, University of Chile, Santiago, Chile

ARTICLE INFO

Article history:

Received 31 July 2013

Received in revised form

21 May 2014

Accepted 9 June 2014

Available online 19 July 2014

Keywords:

Nonlinear mesh generation

Two dimensional

Triangular elements

ABSTRACT

Starting with a boundary triangulation an algorithm automatically generates graded densities of triangular elements inside arbitrary polygonal domains. Interior nodes generated inside acute triangles receive spacing values from Bezier based nonlinear spacing functions defined on interior edges and a nonlinear interpolant defined over triangles. Nodal spacing values, adaptive breakpoints and a single G -value control densities of nodes and offer a range of graded meshes. Predetermined nodes with spacing values and internal boundaries provide additional initial control of nodal densities.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

From its inception in continuum and structural mechanics [1,2, early history] the Finite Element Method has extended its applications to fluid dynamics, metal forming, CAD-CAM integration, geology, manufacturing and medicine as representative examples, see [3] for a comprehensive list of research in 40 countries. New finite element methods can incorporate material discontinuities [4,5], be mesh free [6, background mesh voxels], have moving meshes [7,8] and other mesh options. These references demonstrate that triangulations have been integral to many finite element applications and methods.

From the earliest days of numerical FEM calculations based on triangulations of Courant, Argyris and Turner [9–11] have played an important role. Motivated by improving numerical calculations in daily engineering design and analyses of automotive parts, our paper deals with triangulations used in applying FEM to continuum and structural mechanics. Each vehicle has thousands of metal parts and whenever even small design changes occur many interdependent parts must undergo validations which often require finite element analyses. We present a Delaunay triangulation algorithm for nonlinear graded meshes and create typical meshes for improving three automotive applications chosen for their frequent daily use, design engineering cycles and for adaptive analyses in commercial finite element software.

Starting with a Delaunay boundary triangulation (the Delaunay characteristic is not required but helpful) or a point location in an established Delaunay triangulation, a single global parameter determines the amount of nonlinear grading by applying triangular

interpolants and Bezier curves which then assign each new node a distance that limits encroachment from subsequent nodal insertions. This algorithm which builds upon the structure in Frey's selective refinement [12] differs from prominent methods such as the advancing front algorithm of Suhara and Fukusa [13] and longest edge bisections of Rivara [14,15] which were not designed for controlling variable mesh gradations. Improvements for gradation in advancing front methods, by solving a differential equation [16] or partitioning the meshing domain [17] for example, do not have a control of gradation embedded in the node generating algorithm. Algorithms specifically designed for controlling mesh gradation, by specifying local metrics that may overlap [18] for example, do not have global simplicity and ease of implementation as the algorithm presented in Section 3 of this paper.

Section 2 of this paper briefly reviews the original use of selective refinement [12], a popular term now used to denote the improvement of existing meshes. This section also identifies its straightforward structure as a basis for further development. Section 3 presents a novel nonlinear Delaunay incremental insertion algorithm based on Computer aided Design concepts independent of those used to construct finite element geometries. This section also introduces nonlinear functions defined on edges of triangles and uses a transfinite interpolant [19,20] on triangles, and ends with a summary of the algorithm. Section 4 compares the selective refinement to nonlinear Delaunay incremental insertion on automotive applications. Section 5 provides a summary.

2. Selective refinement

Frey's selective refinement algorithm [12] includes methods for locating nodes on the boundary of meshing domains. Preliminary

^{*} Corresponding author.

E-mail address: dfield08@gmail.com (D.A. Field).

¹ Supported by the Fulbright Foundations of Chile and the United States.

to generating meshes, each boundary node receives a spacing value equal to dividing the minimum distance to its two adjacent nodes by the square root of two. Meshing then begins with a Delaunay boundary triangulation.

Selective refinement continues with the following steps. Insert nodes only inside triangles that contain their circumcenters and on the line segment connecting the incenter and the circumcenter of a triangle. The location on the segment depends on the normalized shape ratio (NSR) ρ , $\rho = 2r/R$, and a linear interpolation giving weight $(1-\rho)$ to the incenter and ρ to the circumcenter. The new node must also satisfy two other criteria. Its distance to the vertices of the triangle cannot be less than the spacing values at the vertices and its insertion into the mesh cannot generate a vertex with degree less than 4 or greater than 8. Linear interpolation of the spacing values at the vertices of the triangle containing the new node assigns its spacing value. Laplacian smoothing constitutes the final step.

Authors subsequent to Frey use the term selective refinement. Their methods often require a stopping criterion or a proof that their algorithms end, see [21–23] for examples. Though not mentioned in [12] Frey's selective refinement intrinsically stops because each node in the mesh lies at the center of a circular disk with radius equal to the spacing value at the node and no new node can be inserted inside this disk. Since the smallest disk cannot have a radius less than the least spacing value at a boundary node, the minimum number of disks having this smallest radius that can cover the entire mesh provides an upper bound for the number of nodes that can be inserted.

The radii of these disks perform another crucial role. According to [12], since the spacing value at a node defined as the distance to its nearest node produces a mesh with “larger spacings than desired” especially “near the boundary nodes”, division by the square root of 2 becomes necessary to “produce triangles which are as well shaped as possible.” Fig. 1 justifies these unsupported assertions. In Fig. 1 the vertices **a** and **b** on the boundary each have minimum distances of one unit from their nearest boundary node. The gray area in the top portion of Fig. 1 indicates the area within the meshing domain where new nodes cannot be inserted due to defining spacing values at boundary nodes **a** and **b** equal to the minimum distances to their neighbors. The smaller gray area at the bottom portion of Fig. 1 indicates the commensurate area using spacing values which divide these minimum distances at **a** and **b** by the square root of two. Consequently, this division allows insertions into the area enclosed by the dots. A node inserted at **e**, the apex of this area, produces an equilateral triangle and a node at **r**, the bottom, produces a 45° right triangle.

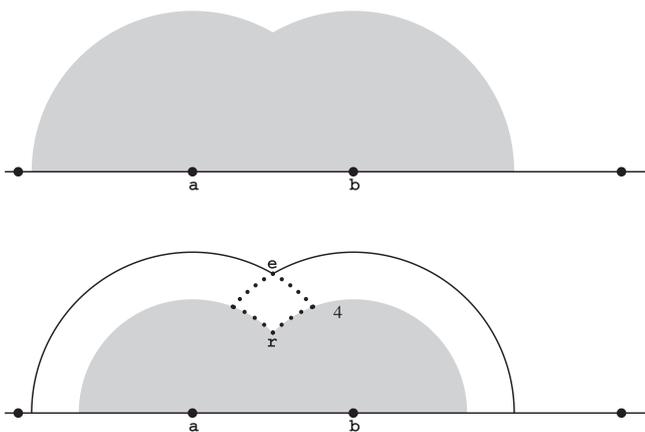


Fig. 1. The top gray area contains the forbidden area of node insertion when using minimum spacing values at boundary nodes **a** and **b**. Minimum spacing values divided by $\sqrt{2}$ generate a smaller gray area enabling insertions **e** and **r** that create an equilateral triangle and a 45° right triangle in the area bounded by sample nodes.

In summary, this linear selective refinement method relies on five core ideas that enable nonlinear gradation of meshes: (1) start with a boundary triangulation, (2) insert new nodes in acute triangles, (3) locate a node inside a triangle based on its shape, (4) constrain the insertion of a node in a triangle by the spacing values assigned to its vertices, and (5) smooth the generated mesh.

3. Nonlinear interpolation and spacing

The algorithm in the previous section restricts spacing values at nodes not to exceed the maximum and minimum spacing values on the boundary. We will add a nonlinear dimension to (3) in this structure that automatically allows greater spacing values among nodes and use the Laplace Delaunay smoothing [24] to maintain Delaunay triangulations. Furthermore a single parameter will control nonlinearity.

The following two subsections use barycentric coordinates in standard linear interpolation and its extension to nonlinear transfinite interpolation over a triangle. The third subsection defines nonlinear spacing functions required for nonlinear transfinite interpolation on the edges of triangles. One parameter will dictate nodal densities implemented by the nonlinear interpolant. The fourth subsection summarizes the new nonlinear algorithm.

3.1. Linear interpolation

For a point **p** in the triangle τ with vertices **v**₁, **v**₂ and **v**₃ the standard linear finite element uses the ratios

$$b_i = \frac{\Delta(\mathbf{v}_j \mathbf{p} \mathbf{v}_k)}{\Delta(\mathbf{v}_1 \mathbf{v}_2 \mathbf{v}_3)}, i \neq j, i \neq k, j \neq k, 1 \leq i \leq 3, 1 \leq j \leq 3 \text{ and } 1 \leq k \leq 3, \tag{3.1}$$

where $\Delta(\mathbf{abc})$ denotes the area of a triangle with vertices **a**, **b** and **c** to define the barycentric coordinates (b_1, b_2, b_3) of **p** in τ . For a triangle with spacing values d_1, d_2 and d_3 at the vertices **v**₁, **v**₂ and **v**₃ calculate the linear spacing value $L(\mathbf{p})$ at an arbitrary point **p** in τ by

$$L(\mathbf{p}) = b_1 d_1 + b_2 d_2 + b_3 d_3. \tag{3.2}$$

3.2. Nonlinear interpolation over triangles

Many nonlinear interpolation schemes over triangles have been constructed to interpolate values at the vertices of triangles, see [25–27] for established references that introduce interpolants on triangles. We have chosen Nielson's side-vertex method [20] incorporating the finite elements attributed to Marshall [19]. The side vertex method has the advantage of interpolating arbitrary nonlinear functions defined on edges without imposing derivative conditions. It also conveniently relies on barycentric coordinates described in the previous subsection.

Using a standard triangle U with vertices $(0, 0)$, $(1, 0)$ and $(0, 1)$, the side-vertex transfinite interpolation method reproduces functions defined on the edges of U with the following equation:

$$F(x, y) = (1-x)F\left(0, \frac{y}{1-x}\right) + (1-y)F\left(\frac{x}{1-y}, 0\right) + (x+y)F\left(\frac{x}{x+y}, \frac{y}{x+y}\right) - xF(1, 0) - yF(0, 1) - (1-x-y)F(0, 0), \tag{3.3}$$

where $y/(1-x) = 0$ at $(1, 0)$, $x/(1-y) = 0$ at $(0, 1)$ and $x/(x+y) = y/(x+y) = 0$ at $(0, 0)$. On the right-hand side of Eq. (3.3) the values of F on the boundary of U are located at the intersections of the cevians from the vertices through (x, y) inside the triangle.

On an arbitrary triangle τ with vertices **v**₁, **v**₂ and **v**₃ let $g_{ij}(v_{ij}(t))$ denote a nonlinear spacing function defined on the edge

$\mathbf{v}_i \mathbf{v}_j$ where $v_{ij}(t) = (1-t)\mathbf{v}_i + t\mathbf{v}_j$, $0 \leq t \leq 1$. The functions defined on the edges of the triangle τ that share a common vertex must have the same spacing value at the shared vertex. Note that the function $g_{ij}(v_{ij}(t))$ is defined on the edge opposite to the vertex \mathbf{v}_k where $i \neq k \neq j$.

Let \mathbf{v}_1 correspond with $(1, 0)$, \mathbf{v}_2 with $(0, 1)$, and \mathbf{v}_3 with $(0, 0)$ in Eq. (3.3). Using the barycentric coordinates $\mathbf{p} = (b_1, b_2, b_3)$ defined in Eq. (3.1), let $b_1 = x$ and $b_2 = y$ in Eq. (3.3). Associate the function g_{32} on edge $\mathbf{v}_3 \mathbf{v}_2$, where $b_1 = 0$, with $F(0, y/(1-x))$ and let $t = b_2/(1-b_1)$. Associate the function g_{31} on edge $\mathbf{v}_3 \mathbf{v}_1$, where $b_2 = 0$, with $F(x/(1-y), 0)$ and let $t = b_1/(1-b_2)$. Associate the function g_{21} on edge $\mathbf{v}_2 \mathbf{v}_1$, where $b_3 = 0$, with $F(x/(x+y), y/(x+y))$ and let $t = b_1/(b_1 + b_2)$. With these correspondences Eq. (3.3) becomes

$$F(\mathbf{p}) = (1-b_1)g_{32}\left(v_{32}\left(\frac{b_2}{1-b_1}\right)\right) + (1-b_2)g_{31}\left(v_{31}\left(\frac{b_1}{1-b_2}\right)\right) + (b_1+b_2)g_{21}\left(v_{21}\left(\frac{b_1}{b_1+b_2}\right)\right) - b_1g_{31}(\mathbf{v}_1) - b_2g_{32}(\mathbf{v}_2) - (1-b_1-b_2)g_{31}(\mathbf{v}_3). \tag{3.4}$$

The side-vertex method has an intrinsic drawback common to many interpolation schemes. Whereas linear interpolation over a triangle attains its extremal values at the vertices, the side-vertex method can attain its extrema inside the triangle when the extrema on the edges do not occur at the vertices. For example, the early stages of inserting nodes into a boundary triangulation can create nearly isosceles triangles with large base angles at vertices on the boundary. Such triangles can be chosen for inserting new nodes before these triangles have been destroyed by previous insertions.

Fig. 2 illustrates an isosceles triangle (not to scale), where $\mathbf{v}_1 = (0, -1)$, $\mathbf{v}_2 = (0, 1)$ and $\mathbf{v}_3 = (\tan(\theta), 0)$. The triangle has base angles θ , $\pi/4 < \theta < \pi/2$ to guarantee that it contains its circumcenter and be eligible for insertion of a new node \mathbf{p} . Let the spacing values equal s at the vertices \mathbf{v}_1 and \mathbf{v}_2 on the boundary and u at \mathbf{v}_3 where $u > s$. In Fig. 2 the cevians from vertices \mathbf{v}_i , $i = 1, 2, 3$, through \mathbf{p} intersect the opposite edges at \mathbf{s}_i . Let the spacing values at $\mathbf{s}_1, \mathbf{s}_2$ and \mathbf{s}_3 equal z, z and s , respectively. To evaluate $F(\mathbf{p})$ in Eq. (3.4) the symmetry of the isosceles triangle and the constant spacing value s on the boundary produce

$$F(\mathbf{p}) = z - b_3(u - z). \tag{3.5}$$

When $u > z > s$, even though as the base angle θ approaches $\pi/2$ the normalized shape ratio approaches zero so that the location of \mathbf{p} approaches the incenter and b_3 approaches zero since $\tan(\theta/2)/\tan(\theta)$ approaches zero, appropriate values of θ , u , z , and s can easily imply $F(\mathbf{p}) < s$. Due to this inequality, our implementation of the side-vertex method imposes a lower bound on $F(\mathbf{p})$ in (3.5) to prevent \mathbf{p} from receiving a spacing value lower than the smallest spacing value at the vertices of the triangle containing \mathbf{p} . This lower bound, greater than or equal to the minimum spacing value on the boundary triangulation, adds the important benefit that nonlinear selective refinement must terminate due to avoiding random loci of progressively smaller triangles. To exceed this lower bound at a particular location either a priori or a posteriori to initial mesh generation, insert an interior node with a desired spacing value at a desired location. Section 4

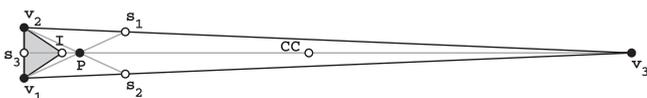


Fig. 2. I, CC and P denote the incenter, the circumcenter and the insertion point, respectively, in the special case of evaluating Eq. (3.4) with edge $\mathbf{v}_1 \mathbf{v}_2$ on the boundary and spacing value at \mathbf{v}_3 greater than at \mathbf{v}_1 and \mathbf{v}_2 .

contains an important application of mesh generation using an a posteriori insertion.

3.3. Nonlinear edge functions

To produce nonlinear spacing values, the transfinite interpolant in Section 3.2 requires nonlinear interpolations of the spacing values at the endpoints of each edge in a triangle. Based upon many experiments, we define edge functions that receive input from the spacing values at the endpoints, from the length of the edge and from a parameter that controls the density of nodes in the mesh. We represent these functions with Bezier curves.

Denote the endpoints of an edge by \mathbf{v}_1 and \mathbf{v}_2 and let their respective spacing values equal d_1 and d_2 . Without loss in generality suppose $d_1 \leq d_2$ and the edge has length d_{len} . Unless the boundary of the triangulation has a predetermined nonlinear spacing function, if \mathbf{v}_1 and \mathbf{v}_2 are adjacent vertices on the boundary then define a linear spacing function between \mathbf{v}_1 and \mathbf{v}_2 . An interior edge $\mathbf{v}_1 \mathbf{v}_2$ has linear spacing only when $d_{len} \leq d_1 + d_2$. Otherwise split the edge at the midpoint or adaptively at

$$\mathbf{m} = \frac{d_1}{d_1 + d_2} \mathbf{v}_1 + \frac{d_2}{d_1 + d_2} \mathbf{v}_2. \tag{3.6}$$

At the point \mathbf{m} define a spacing value d_m by

$$d_m = d_2 + G\sqrt{d_1 d_{len}}, \quad 0 < G, \tag{3.7}$$

where G regulates the spacing values assigned to inserted nodes. d_m depends on d_{len} because the longer the edge the greater the space between the nodes at the ends of the edge. d_m should be higher than d_2 and should also depend on d_1 . A sufficiently high G -value will prevent the insertion of any nodes. $G=1$ can be such a value.

On each normalized segment of the edge split by \mathbf{m} , define a cubic Bezier function:

$$B_3(t) = \mathbf{b}_0(1-t)^3 + \mathbf{b}_1(1-t)^2t + \mathbf{b}_2(1-t)t^2 + \mathbf{b}_3t^3, \quad 0 \leq t \leq 1. \tag{3.8}$$

Define functions $B^{[1]}(t)$ and $B^{[2]}(t)$ having control points $\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3$, and $\mathbf{b}_3, \mathbf{b}_4, \mathbf{b}_5, \mathbf{b}_6$, respectively, as follows:

$$\mathbf{b}_0 = \begin{pmatrix} 0 \\ d_1 \end{pmatrix}, \quad \mathbf{b}_1 = \begin{pmatrix} w_1 \\ d_1 \end{pmatrix}, \quad \mathbf{b}_2 = \begin{pmatrix} w_1 \\ d_m \end{pmatrix}, \quad \mathbf{b}_3 = \begin{pmatrix} 1 \\ d_m \end{pmatrix} \tag{3.9}$$

and

$$\mathbf{b}_3 = \begin{pmatrix} 0 \\ d_m \end{pmatrix}, \quad \mathbf{b}_4 = \begin{pmatrix} w_2 \\ d_m \end{pmatrix}, \quad \mathbf{b}_5 = \begin{pmatrix} w_2 \\ d_2 \end{pmatrix}, \quad \mathbf{b}_6 = \begin{pmatrix} 1 \\ d_2 \end{pmatrix}. \tag{3.10}$$

Even though each function has repeated values in their coordinates, d_1, w_1 and d_m in $B^{[1]}(t)$ and d_m, w_2 and d_2 in $B^{[2]}(t)$, the two segmented Bezier Curve maintains 2nd-degree continuity at the join \mathbf{m} . Locating these control points close to \mathbf{m} enhances the ability of G in Eq. (3.7) to regulate the number of inserted nodes.

The evaluation of the Bezier curve defined on an edge $\mathbf{v}_i \mathbf{v}_j$ begins by parameterizing the edge starting from the vertex with

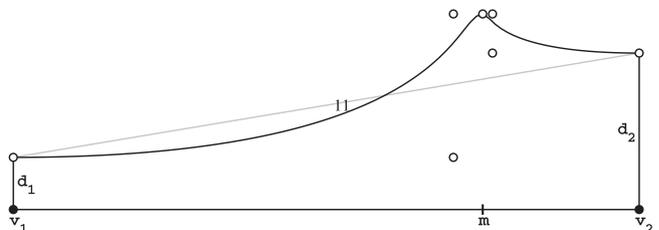


Fig. 3. The circles denote Bezier points used to define the edge function on edge $\mathbf{v}_1 \mathbf{v}_2$ with an adaptive breakpoint \mathbf{m} and unequal spacing values at its endpoints and $G = 1/8$.

the smallest spacing value. Fig. 3 shows an edge function with an adaptive \mathbf{m} when $w_1 = 15/16$, $w_2 = 1/16$ and $G = 1/8$. Note how much of the edge function lies below the gray line that indicates the spacing values of a linear edge function. The flatness at the ends of the Bezier curve contributes to the smooth transition of spacing values calculated along edges sharing a common node.

To evaluate $B^{(1)}(t)$ and $B^{(2)}(t)$ at \mathbf{x} on an edge $\mathbf{v}_1\mathbf{v}_2$ of length d_{len} , let $\|\mathbf{v}_1 - \mathbf{x}\| = d_x$ and $t_x = d_x/d_{len}$. Letting $t = t_x/t_m$, where t_m equals $1/2$ when \mathbf{m} lies fixed at the midpoint of edge $\mathbf{v}_1\mathbf{v}_2$ and equals $d_2/(d_1 + d_2)$ when adaptively locating \mathbf{m} on edge $\mathbf{v}_1\mathbf{v}_2$, evaluate $B^{(1)}(t)$ if $0 \leq t_x \leq t_m$. Otherwise, let $t = (t_x - t_m)/(1 - t_m)$ and evaluate $B^{(2)}(t)$.

Remark. Different nonlinear triangular interpolants and edge functions such as higher order finite elements can be applied to generate nonlinear graded meshes.

3.4. Algorithm for nonlinear Delaunay graded meshes

- Initializations
 - Create a Delaunay boundary triangulation.
 - Create a list of acute triangles.
 - Assign each boundary node a spacing value equal to dividing its minimum distance to all other boundary nodes by $\sqrt{2}$. (The spacing value at a node defines the radius of a disk centered at the node within which no other node may be inserted.)
 - Choose a G -value, $\frac{1}{8} \leq G \leq \frac{1}{32}$.
- Incremental Delaunay insertions (Repeat until the list of acute triangles is empty.)
 - Choose the first triangle in the list of acute triangles.
 - Using the normalized shape ratio ρ , $\rho = 2r/R$, and linear interpolation giving weight $(1 - \rho)$ to the incenter and ρ to the circumcenter define a new node in the acute triangle.
 - Test whether the node lies inside any of the discs centered at the vertices of the triangle.
 - If the new node lies in any of the discs eliminate the triangle from the list and choose the next acute triangle.
 - Otherwise, evaluate Eqs. (3.6)–(3.10) to determine the nonlinear spacing interpolation function along each edge of the acute triangle.
 - Assign the new node its spacing value by evaluating Eqs. (3.3) and (3.4) without exceeding the lowest spacing value at the vertices.

Insert the new node into the triangulation and reinitialize the list of acute triangles.

- Perform Laplace Delaunay Smoothing.

Remark. This algorithm produces nonlinear graded Delaunay meshes implemented as an *iterative* method. A slight simplification in the algorithm can produce a *stationary* method by using the triangles in the boundary triangulation to assign a spacing value to every new node. In this case the boundary triangle containing a new node in a current acute triangle determines the spacing value of the new node. This dependency on the boundary triangulation has the advantage of not requiring a lower bound on the nonlinear interpolation in Eqs. (3.4) and (3.5). However, the dependency echoes initial boundary triangulations and produces patchy triangulations that necessitate using the iterative algorithm.

4. Nonlinear Delaunay graded meshes

This section begins by discussing the iterative implementation of the algorithm in Section 3.4 along with numerical characterizations of graded meshes. The section then illustrates nonlinear Delaunay graded meshes with examples that improve FEM analyses for continuum and structural mechanics in the automotive industry.

4.1. Implementing and quantifying nonlinear Delaunay graded meshes

The iterative algorithm in Section 3.4 depends on searches for acute triangles in current triangulations. Depending on data structures, the searches can be very efficient. Choosing the first available versus the worst shaped acute triangle does make a difference. The latter only needs to update an ordered current list of worst acute triangles and calculate a “worst” criterion for each new acute triangle produced when inserting a new mesh into the current mesh. However, the worst triangle approach tends to generate noticeably inferior meshes. The first available strategy tends to spread the insertions of new nodes from places where better triangles already exist. We chose the first available triangle option.

Meshes with higher G -values produce less dense meshes with steeper gradations. Meshes with lower G -values become increasingly uniform. Meshes with fixed midpoints produce predominantly uniform interior meshes. To obtain comparable numbers of

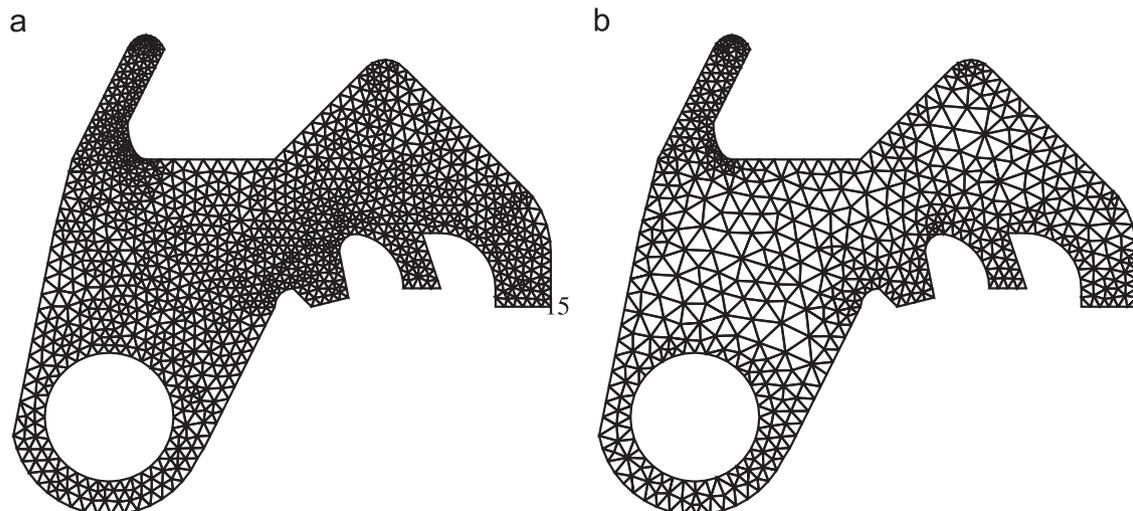


Fig. 4. Meshes of a standard automotive part generated by (a) linear selective refinement and (b) nonlinear selective refinement using $G = 1/16$ and adaptive midpoints have 1056 and 447 internal nodes.

elements using fixed and adaptive midpoints, meshes using fixed midpoints require lower G -values.

The *spacing* at a node (different from its spacing value) measures the minimum length of all edges connected to the node. Since the *minimum spacing*, the shortest edge in a mesh, can differ substantially from the *maximum spacing*, the largest spacing in a mesh, define the *global spacing factor* as the maximum spacing divided by the minimum spacing. Large global spacing factors imply steeper gradations from high densities to low densities of elements. For a mesh with a uniform distribution of boundary nodes a global spacing factor up to 10 typically indicates that the mesh should have a good gradation of elements. Global spacing factors less than 5 indicate smaller variations in mesh density and more uniformly shaped elements. The global spacing factor can be used to choose appropriate G -values in Eq. (3.7) by iterating G until achieving a predetermined global spacing factor.

The *boundary spacing factor* limits the calculation of the global spacing factor to boundary nodes. For meshes with boundary spacing factors larger than 5, global spacing factor larger than 10 can also indicate well graded meshes. When a global spacing factor becomes very large, dividing the maximum spacing by the largest boundary spacing yields a quotient comparable to global spacing factors for meshes generated from uniform boundary spacings. Call this quotient the *adjusted global spacing factor*. Adjusted global spacing factors less than 10 tend to indicate good mesh gradations.

In Section 3.3 splitting edges of a triangle either in two equal lengths or adaptively as defined in Eq. (3.6) provide distinct *fixed* and *adaptive* types of nonlinear Delaunay refinements. These splittings led to investigating four versions of nonlinear Delaunay refinements, a stationary and an iterative method each with fixed midpoints or adaptive breakpoints. In each version investigations included appraising good G -values for d_m in Eq. (3.7) and for the control points in Eqs. (3.9) and (3.10). We recommend an iterative method, adaptive breakpoints, G -values between $1/8$ and $1/32$ in Eq. (3.7), and control points $w_1 = 15/16$ and $w_2 = 1/16$.

4.2. Examples and comparisons

In the automotive industry parts often change daily due to new designs and engineering analyses. Since a part change frequently engenders further analyses and changes to interdependent parts and since vehicles have thousands of parts, any speedup through fewer elements and graded meshes mirroring adaptivity in each of the myriad analyses accumulates to significant cost savings. Initial meshes with good quality boundary elements and fewer interior elements are advantageous in this endeavor. The first two examples compare nonlinear Delaunay grading with the linear selective refinement algorithm [12] used for initial meshes of automotive parts.

Fig. 4 presents two initial meshes of a typical flat thick automotive metal part. The mesh on the left created with linear selective refinement has 1056 internal nodes. The mesh on the right created with nonlinear Delaunay using a G -value of $1/16$ has 447 internal nodes for a nodal reduction of 57.67%. In the left mesh the elements have an average NSR-value of 0.95 and the lowest NSR-value of an element equals 0.59. By comparison the average NSR-value in the right mesh equals 0.94 and the lowest NSR-value of an element equals 0.62.

Analyses of sheet metal forming in the automotive industry provided the motivation for linear selective refinement, see [12, Figs. 8–11] which departs from the linearity of the basic algorithm presented in Section 2 to produce quadratic variation in mesh density in limited situations. Other authors use changes in local metrics to produce nonlinear local variation in mesh density, see Borouchaki et al. [18] for example. The algorithm in Section 3 for generating nonlinear graded meshes provides a simpler alternative

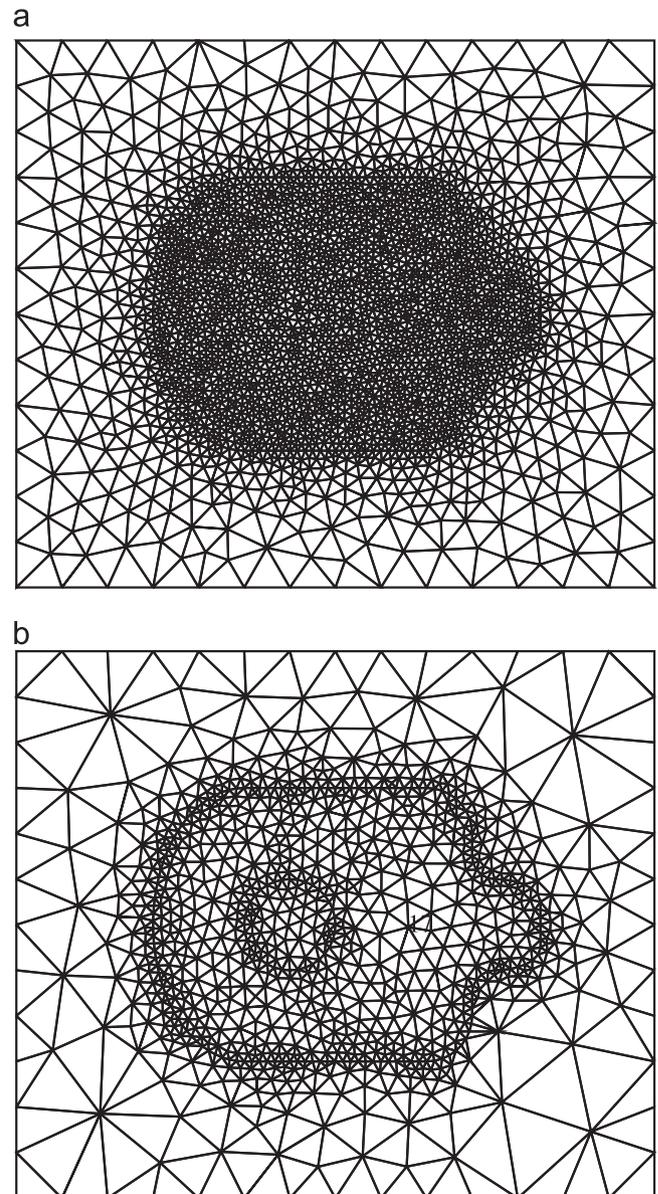


Fig. 5. Mesh (a), generated by linear selective refinement, and mesh (b), generated by nonlinear selective refinement, illustrate initial meshes for analyzing stamping sheet metal blanks.

for complex geometries. The following example examines a stamped sheet metal template used for testing bending radii, meshes and FEM software. In this example inner boundaries determining the location of bends demonstrate the easy control of local mesh densities. Compute the spacings for internal boundary nodes as ordinary boundary nodes, that is, use the minimum distance to their neighbors in the boundary triangulation to compute their spacing values.

As the mesh in Fig. 5(a) shows, the linear selective refinement in [12] produces a very dense interior mesh that does not distinguish where bending occurs. In addition to the 239 inner and outer boundary nodes, this mesh contains 2460 internal nodes. The complexity of bending in this normative template overwhelmed the limited application of quadratic variation in [12].

Fig. 5(b) displays the result of a nonlinear graded mesh generated using $G=1/16$. The mesh not only identifies the delineation of severe bending radii at internal boundaries but it also delivers a substantial similarity to an adaptively refined mesh. Whereas a comparable *proprietary* adaptive mesh produced by

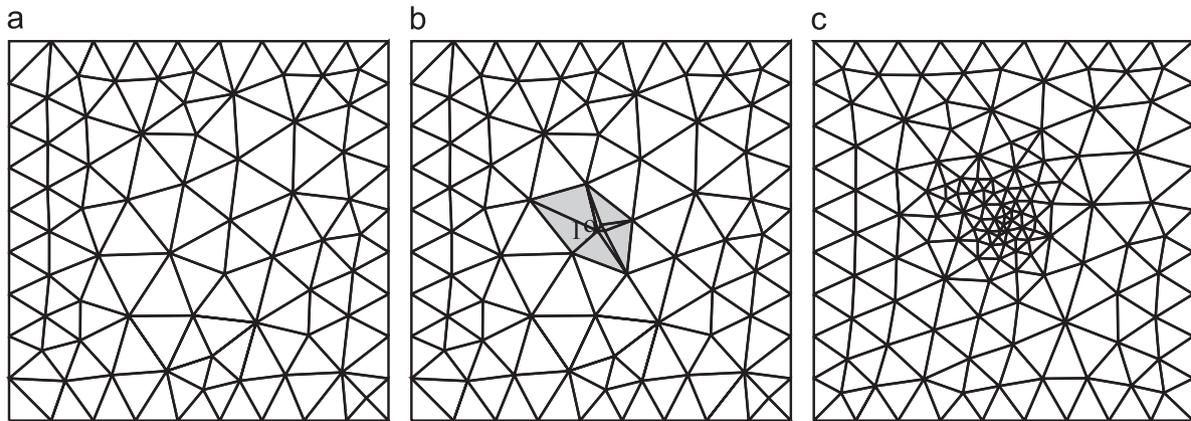


Fig. 6. The three stages for incorporating a spot weld into an existing Delaunay triangulation.

FEM software contained many noncompatible elements, Fig. 5 (b) contains only Delaunay triangles. 723 internal nodes have been inserted in this nonlinear graded mesh for a 70.6% reduction in the number of internal nodes. The elements have an average NSR-value of 0.93 with one outlier element having the lowest NSR-value of 0.39, see [28, Fig. 2a] to visually track along a level curve all triangles having this acceptable NSR-value. This mesh has a global spacing factor of 16.65 and an adjusted global spacing factor of 2.46. Although the graded meshes in previous figures were generated with $G = 1/16$, generating meshes with $G = 1/8$ had desirable quality as well.

A particularly troublesome ad hoc task for engineers occurs when analyses suggest the addition of spot welding to enhance bonding of sheet metal with adhesives. Engineers routinely increase mesh densities manually at locations of the spot welds and occasionally return to the original mesh to move the locations of the spot welds. Since the use of Delaunay meshes has become commonplace, nonlinear Delaunay grading easily resolves this tedious and very time consuming chore.

The meshes displayed in Fig. 6 show three phases in this resolution. Fig. 6(a) shows a Laplace Delaunay smoothed mesh into which a welding spot will be introduced. The light gray area in Fig. 6(b) denotes the polygon surrounding the welding spot; the dark triangle could be replaced by a different Delaunay triangulated shape. If a node in the original triangulation lies too close to the weld, a distance less than the diameter of the weld, then it should be removed. Its removal poses no difficulty because a Delaunay retriangulation of its insertion polygon, the vertices connected to the node, preserves the Delaunay triangulation. Such a node must be removed before inserting the vertices denoting the spot weld.

Fig. 6(c) displays the results of the third phase. In this phase since the mesh in Fig. 6(a) has been Laplace Delaunay smoothed, recalculate the spacing values at each vertex in Fig. 6(b). Next, replace the spacing values at the vertices of the polygon surrounding the spot weld to the maximum spacing value at the vertices defining the spot weld. Last, continue creating a nonlinear Delaunay graded mesh.

5. Conclusions

An algorithm has produced nonlinear graded Delaunay meshes using a novel combination of a nonlinear triangular interpolant and Bezier curves. Meshes with steep gradients in nodal density have been automatically created with one parameter controlling nonlinear distributions of densities from locations where geometry and physics demand fine densities to areas needing fewer nodes.

A nonlinear interpolant controls the gradation of triangles through Bezier edge functions dependent on edge lengths, spacing values at the endpoints and a fixed G -parameter. Breakpoints at the center of edges give relatively uniform interior meshes versus adaptive breakpoints which yield greater global mesh gradations. Adaptive breakpoints and G -values between $1/8$ and $1/32$ have been recommended for well graded meshes. Initializing a mesh with artificial interior boundaries and singleton vertices with preassigned spacing values establishes local mesh densities. The method and framework of the algorithm accommodates different nonlinear triangular interpolants and nonlinear edge functions.

New nodes cannot be inserted inside any circular disc defined by a radius equal to the spacing value of its nodal center. The minimum number of discs whose radii equal the least spacing value of a boundary node that can cover the meshing domain provides an upper bound on the number of nodes that nonlinear selective refinement can insert. Therefore the algorithm must terminate.

Acknowledgment

The first author gratefully acknowledges discussions on selective refinement with William H. Frey, the author of [12].

References

- [1] R.W. Clough, Original formulation of the finite element method, *Finite Elem. Anal. Des.* 7 (1990) 89–101.
- [2] K.K. Gupta, J.L. Meek, A brief history of the beginning of the finite element method, *Int. J. Numer. Methods Eng.* 39 (1996) 3761–3774.
- [3] (<http://www.robertschneiders.de/meshgeneration//research.html>).
- [4] N. Moës, J. Dolbow, T. Belytschko, A finite element method for crack growth without remeshing, *Int. J. Numer. Methods Eng.* 46 (1999) 131–150.
- [5] Y. Cai, X. Zhuang, H. Zhu, A generalized and efficient method for finite cover generation in the numerical manifold method, *Int. J. Comput. Methods* 10 (2013) 1350028, <http://dx.doi.org/10.1142/S021987621350028X> (19 pp).
- [6] T. Rabczuk, T. Belytschko, Adaptivity for structured meshfree particle methods in 2D and 3D, *Int. J. Numer. Methods Eng.* 63 (2005) 1559–1582.
- [7] E. Onate, S.R. Idelsohn, F. Del Pin, R. Aubry, The particle finite element method. An overview, *Int. J. Comput. Methods* 20 (2004) 267–308.
- [8] I. Malcev, O. Ghattas, Dynamic-mesh finite element method for Lagrangian computational fluid dynamics, *Finite Elem. Anal. Des.* 38 (2002) 965–982.
- [9] R. Courant, Variational methods for the solution of problems of equilibrium and vibrations, *Bull. Am. Math. Soc.* 49 (1943) 1–23.
- [10] J.H. Argyris, Energy theorems and structural analysis, Part I, *Aircraft Eng.* 26 (1954) 383.
- [11] M.J. Turner, R.W. Clough, H.C. Martin, L.T. Topp, Stiffness and deflection analysis of complex structures, *J. Aeronaut. Sci.* 25 (1956) 805–823.
- [12] W.H. Frey, Selective refinement: a new strategy for automatic node placement in graded triangular meshes, *Int. J. Numer. Methods Eng.* 24 (1987) 2183–2200.
- [13] J. Suhara, J. Fukuda, Automatic mesh generation for finite element analysis, in: J.T. Oden, R.W. Clough, Y. Yamamoto (Eds.), *Advances in Computational*

- Methods in Structural Mechanics and Design, University of Alabama Press, Tuscaloosa, AL, 1972, pp. 607–624.
- [14] M.C. Rivara, Mesh refinement processes based on the generalized bisection of simplices, *SIAM J. Numer. Anal.* 21 (1984) 604–613.
- [15] M.C. Rivara, New longest-edge algorithms for the refinement and/or improvement of unstructured triangulations, *Int. J. Numer. Methods Eng.* 40 (1997) 3313–3324.
- [16] S. Pirzadeh, Structured background grids for generation of unstructured grids by advancing-front method, *AIAA J.* 31 (1993) 257–265.
- [17] J.C. Cavendish, Automatic triangulation of arbitrary planar domains for the finite element method, *Int. J. Numer. Methods Eng.* 8 (1974) 679–696.
- [18] H. Borouchaki, F. Hecht, P.J. Frey, Mesh gradation control, *Int. J. Numer. Methods Eng.* 43 (1998) 1143–1165.
- [19] J.A. Marshall, Some applications of blending function techniques to finite element methods (Ph.D. thesis), University of Dundee, 1975.
- [20] G.M. Nielson, The side-vertex method for interpolation in triangles, *J. Approx. Theory* 25 (1979) 318–336.
- [21] J.A. Ruppert, Delaunay refinement algorithm for quality 2-dimensional mesh generation, *J. Algorithms* 18 (1995) 548–585.
- [22] L.P. Chew, Guaranteed-quality mesh generation for curved surfaces, in: Proceedings of the Ninth Annual Symposium on Computational Geometry, San Diego, CA, 1993, pp. 274–280.
- [23] L.P. Chew, Guaranteed-quality Delaunay meshing in 3D, in: Proceedings of the 13th Annual Symposium on Computational Geometry, Nice, France, 1997, pp. 391–393.
- [24] D.A. Field, Laplacian smoothing and Delaunay triangulations, *Commun. Appl. Numer. Methods* 4 (1988) 709–712.
- [25] R.E. Barnhill, G.B. Birkoff, W.J. Gordon, Smooth interpolation in triangles, *J. Approx. Theory* 8 (1973) 114–128.
- [26] R.E. Barnhill, G. Farin, C^1 quintic interpolation over triangles: two explicit representations, *Int. J. Numer. Methods Eng.* 17 (1981) 1763–1778.
- [27] M.J.D. Powell, M.A. Sabin, Piecewise quadratic approximations over triangles, *ACM Trans. Math. Softw.* 3 (1997) 316–325.
- [28] D.A. Field, Qualitative measures for initial meshes, *Int. J. Numer. Methods Eng.* 47 (2000) 887–906.