INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2013; **96**:247–268
Published online 3 September 2013 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/nme.4567

On the interpolation of normal vectors for triangle meshes

P.-A. Ubach[1,2,*,†], C. Estruch[1,2] and J. Garcia-Espinosa[1,2]

[1] *Centre Internacional de Mètodes Numèrics en Enginyeria (CIMNE) http://www.cimne.upc.edu*
[2] *Universitat Politècnica de Catalunya (UPC), C. Esteve Terrades 5, Parc Mediterrani de la Tecnologia, 08860 Castelldefels, Spain*

ABSTRACT

This paper analyzes the problem arising from the need to assign information about the normal vectors to the surface at the nodes of a mesh of triangles. Meshes of triangles do not have normals uniquely defined at the nodes. A widely used technique to compute the normal direction at any given node is to compute the weighted average of the normals of each surrounding triangle.

The present study proposes new weighting factors to compute the normal directions at the nodes of the mesh of triangles of a general surface. Previous weights found in the literature used the geometric dimensions of the triangles themselves to design the weighting factors. The new factors are proposed using the triangles' circumscribed circles dimensions. The new weights provide superior results than the ones obtained by previous best practices for a wide range of surfaces.

An advanced framework based on the approachability of smooth surfaces by quadrics is presented and used. This framework helps to understand the improved performance of the presented factors with respect to other factors found in the literature. A comprehensive numerical comparison analysis is performed, and the most precise of all factors is clearly identified. Copyright © 2013 John Wiley & Sons, Ltd.

Received 8 October 2011; Revised 3 August 2013; Accepted 6 August 2013

KEY WORDS: averaging normal vectors; normal interpolation; approximate normal to a surface; triangle meshes; weighting factors; geometric modeling

1. INTRODUCTION

Sometimes, when analyzing surfaces with the help of triangle meshes, it is necessary to have information about the direction of the normal direction to the surface at the nodes. A triangle-based linear discretization (with C^0 continuity) does not provide direct and unambiguous information about the surface normal at the nodes of the mesh. This circumstance also extends to the edges of the triangles. In general, each node is connected to more than one triangle. Each of those triangles does have a normal vector uniquely defined. Therefore, there is a need to infer the normal vector at the node from the many normal vectors of the triangles sharing that node (Figure 1). This is a problem that appears in many fields of computational mechanics:

- CFD analysis requires information about the normal vectors to the surfaces surrounding the fluid. This is required, for example, to determine the force applied by the fluid pressure on the surface [1].

*Correspondence to: P.-A. Ubach, CIMNE, Edificio C3, Parc Mediterrani de la Tecnologia, UPC C./ Esteve Terrades 5, 08860 Castelldefels, Barcelona, Spain.

†E-mail: ubach@cimne.upc.edu

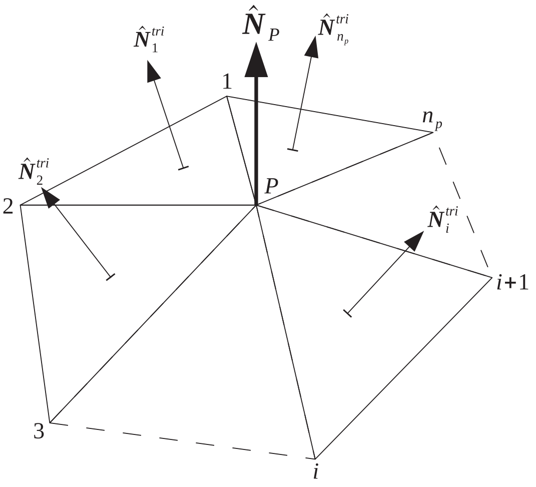

$\hat{\boldsymbol{N}}_P$
$\hat{\boldsymbol{N}}_1^{tri}$
$\hat{\boldsymbol{N}}_{n_p}^{tri}$
$\hat{\boldsymbol{N}}_2^{tri}$
$\hat{\boldsymbol{N}}_i^{tri}$
1
2
3
i
$i+1$
n_p
P

Figure 1. Schematic representation of a point P belonging to a surface discretized by a set of n_P triangles. Each triangle has a unique unit normal vector defined to it $\left(\hat{N}_i^{tri}\right)$, and the goal is to find the best interpolation of those vectors to obtain the normal at point P: $\hat{N}_P$.

- Shell analysis requires computing the curvature of a surface. Some advanced formulations need deriving the normals at the nodes in order to compute the surface's curvature [2, 3].
- Fluid-structure interaction analysis requires transferring information between the fluid and the structure. In [4], Tezduyar *et al.* propose using the normals at the nodes of a surface mesh to project the scalar pressure field from the fluid to the structure. In later publications [5], they move away from the problem of computing the normals at the nodes by using the normals of the elements at the integration points.

Because errors in the primitive function involve larger errors in its derivative, it is of the utmost importance to minimize the error in the computation of the normals at the nodes in the mesh of triangles if we want to use this information in the estimation of curvatures. This is a topic that has raised the interest of many authors in the past, specially in the domain of computer-aided geometrical design. Some authors study this as an interpolation problem: Meek and Walton [6] make an asymptotic study on the error of various structured and unstructured interpolation and fitting approaches, and OuYang and Feng [7] also compare weighting and fitting methods by performing an average of several numerical computations. Excellent reviews of all these methods can be found in [8, 9].[‡] Some remarkable works in this field are the developments by Max [10], Meyer *et al.* [11], and more recently Langer *et al.* [12].

1.1. Problem statement

In this paper, the authors use the interpolation (weighting) approach, using the topological information provided by the mesh. Thus, given a set of triangles forming a discretization of a surface S, let

$$\left\{\hat{\boldsymbol{N}}_1^{tri}, \ldots, \hat{\boldsymbol{N}}_{n_p}^{tri}\right\} \in \mathbb{R}^3 \tag{1}$$

[‡] For an excellent review of interpolation and fitting methods aimed at deriving curvatures from triangular meshes, the reader can refer to [13]

DOI: 10.1002/nme

be the set of unit normal vectors of the triangles around a node $P \in S$. The goal is to find the weighting factors:

$$w^i \in \mathbb{R}, i = 1 \div n_p \mid \hat{N}_P = \frac{\sum_{i=1}^{n_p} w^i \hat{N}_i^{tri}}{\| \sum_{i=1}^{n_p} w^i \hat{N}_i^{tri} \|} + \vec{\mathcal{O}} \tag{2}$$

where $\hat{N}_P$ is the actual unit normal vector of S at P and $\vec{\mathcal{O}}$ represents the error made, so that $\vec{\mathcal{O}}$ is minimized (see Figure 1 for a graphical representation).

1.2. Analysis of the approach

The main concern when deriving the weights w^i is to find a good mapping function between the unit normal of each triangle $\left(\hat{N}_i^{tri}\right)$ and its local representation of the surface S at the point of study P. Previous works on this field have mostly taken one of two different approaches:

- either the derivation of the weights takes only into account the discrete information of the triangles in the mesh and neglects the higher order nature of the surface being approximated,
- or the higher order nature of the surface is accounted for, but then the need to fit high order functions forces to drop the idea of using a formula similar to Equation (2).

Next, we will describe a theoretic framework that tries to bridge both approaches.

Let us consider the surface S and its local Taylor expansion series at point P. Then, by truncating this series at the second order terms, we will obtain a quadric. So now we have two different approximations to S at P, namely

- the set of triangles with a vertex at P
- and the quadric.

If we now make the assumption that the quadric interpolates all the vertices of the set of triangles, the error we are making is of the order of h^2. Being h a measure of the size of the triangles. This allows us to reinterpret the relationship between $\hat{N}_i^{tri}$ and the surface S. $\hat{N}_i^{tri}$ can be thought of as the normal direction of a plane section of the quadric.

We will still make one more assumption. This is that the conic resulting from the intersection with the quadric is a closed curve. This may seem like a very strong assumption. But in fact, it is no more limiting than the restriction that represents the triangles themselves. What this assumption implies is that the section of the quadric is bounded; just like the triangles are.

In no way are we implying in the current argument that this framework will produce approximations with an error bounded by $\mathcal{O} \propto h^2$.[§] What this analysis provides us is a framework to enrich the information provided by the set of $\hat{N}_i^{tri}$ and their corresponding triangles.

1.3. Outline

In Section 2, we present a review of different weighting factors previously used in the literature, and we also propose new factors taking advantage of the framework presented earlier. In Section 3, we introduce the methodology followed for studying the numerical performance of each of the factors presented. Section 4 gives a thorough overview of the results obtained. The different factors are compared against each other in a wide range of cases. Emphasis is made in presenting the results graphically, although a table with numerical values is also presented at the very end. Additionally, in Section 5, the reader can see an explicit example of the improvements and accuracy provided by

[§] This can indeed happen if we do fit a quadric and obtain the information of the normal at P directly from the quadric [6, 9].

the most precise of the proposed weights. Finally, in Section 6, the main conclusions are summarized, and the authors also highlight the importance of the results presented in the fields of numerical methods and computational geometry.

2. PREVIOUS AND NEW WEIGHTING FACTORS PROPOSED

Taking into account the framework presented in Section 1.2 and assuming that plane sections of the surface approached by a quadric produce closed conics,[¶] then we can infer that each triangle is providing information about an ellipse. However, there is too little information in the triangle to determine a circumscribing ellipse. We can make an approximation to the ellipse using the circumscribed circle instead.

We will use this result to propose different weighting factors based on the magnitudes of the circumscribed circle.

Note that this approach has similarities to the one proposed by Max in [10]. In his work, Max makes the assumption that the surface S can be approached locally at P by a sphere. Therefore, he derives the weights that will yield the exact normal at P if the surface S was a sphere. We could arrive to the result of using the circumscribed circles by making sections of the sphere. However, we esteem that the framework presented is more complete, as the different sources of error are decomposed for further analysis.

Because real meshes used in finite element problems represent surfaces with very different shapes, our aim is to improve the result obtained by Max in [10] for a random surface. Clearly, nothing will beat Max's formula to recover normals in spheres. But can we better approximate the curvature of a broader range of surfaces? A new collection of weighting factors is presented to evaluate comparatively their exactitude and versatility in front of others presented previously in the literature.

From this point, we make an abuse of notation. We will refer to the definition of each set of weighting factors using a descriptive subindex, and we will omit the superscript:

$$w^{i}_{description} \equiv w_{description} \tag{3}$$

2.1. Determination of weights to approximate the normal to a circle

Let us first show the solution for the problem reduced to two dimensions. For the 2D case, Max [10] and Linhard *et al.* [2] show the way that leads to the exact solution (Figure 2 and Equations (4) to (7)).

$$\hat{N}_1 = \left(\cos\left(\tfrac{1}{2}(\tfrac{\pi}{2} + \theta_1)\right), \sin\left(\tfrac{1}{2}(\tfrac{\pi}{2} + \theta_1)\right)\right) \tag{4}$$

$$l_{C_1} = \left|2R \cdot \cos\left(\tfrac{1}{2}(\tfrac{\pi}{2} + \theta_1)\right)\right| \tag{5}$$

$$\hat{N}_2 = \left(\cos\left(\tfrac{1}{2}(\tfrac{\pi}{2} + \theta_2)\right), \sin\left(\tfrac{1}{2}(\tfrac{\pi}{2} + \theta_2)\right)\right) \tag{6}$$

$$l_{C_2} = \left|2R \cdot \cos\left(\tfrac{1}{2}(\tfrac{\pi}{2} + \theta_2)\right)\right| \tag{7}$$

l_{C_1} and l_{C_2} are the respective lengths of segments C_1 and C_2 in Figure 2. While $\hat{N}_1$ and $\hat{N}_2$ are their respective unit normal vectors pointing outward. The angles θ_1 and θ_2 mark the position of

[¶] There are three classes of quadrics that cannot produce closed conic sections: hyperbolic paraboloid, hyperbolic cylinder, and parabolic cylinder.

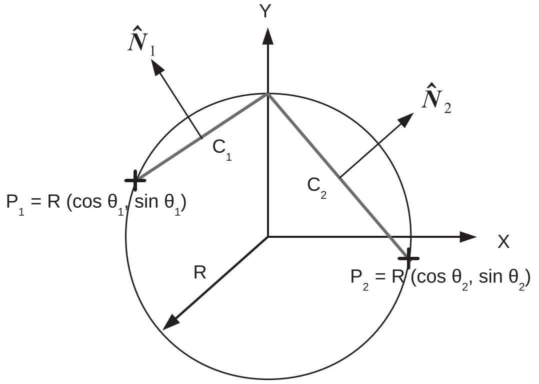

Y
$\hat{N}_1$
$\hat{N}_2$
C_1
C_2
$P_1 = R (\cos \theta_1, \sin \theta_1)$
X
R
$P_2 = R (\cos \theta_2, \sin \theta_2)$

Figure 2. Schematic representation of the problem reduced to two dimensions. A curve is approximated/represented by a circle with its center at the origin. And two chords (representing segments that discretize and approximate an arch of the circle) meeting at the intersection of the circle with the vertical axis are depicted. Similarly, the normal vectors to each of the segments are shown.

points P_1 and P_2 on the circle and are measured counterclockwise with origin on the X axis. Operating, we can verify that the sum $\hat{N}_1/l_{C_1} + \hat{N}_2/l_{C_2}$ yields a vector with only vertical component. So, in this 2D example, the weighting factors would be defined as follows:

$$w^i = \frac{1}{l_{C_i}} \underset{not}{\equiv} w^i_{1/l_C} \tag{8}$$

2.2. *Inverse of the circumscribed circle's diameter and internal angle (new)*

We start with this construction for didactic purposes. In order to keep the development simple, let us consider that the quadric that approximates the surface S is a sphere of radius R centered at the origin (this will force the sections to yield circles). The node under study will be the north pole, that is, the intersection of the positive Z axis with the sphere's surface. Firstly, we must note that—unlike the 2D case—the number of triangles meeting at the north pole in which we can discretize the sphere is not determined. Let us consider then the most simple case in which the north pole is surrounded by only three triangles (Figure 3).

In order to select a magnitude of the circumscribed circle taking advantage of the 2D case result, we may proceed as follows:

(1) Let us consider the three unit normal vectors to each of the triangles surrounding the north pole as free vectors.
(2) Once in the free vectors framework, we can consider these vectors as being also normal to a geometric entity different from the triangle. As explained in the framework presented in Section 1.2, we will choose a magnitude of the circumscribed circle to the triangle. In Figure 3, this circle is represented in blue color for the triangle defined by points A, B, and C.
(3) Let us consider now the diameter of the circumscribed circle with an end in A (north pole). Because A is a vertex of the triangle, it is also a point on the circle. In Figure 3, this diameter is represented in dark green color.
(4) This diameter defines at its opposite end another point on the sphere. Let us call this new point A'. A and A' define a great circle of the sphere of radius R. Using this construction, we can partially recover the layout of the 2D problem. In this way, the great circle becomes the circle in 2D (Figure 2), and the diameter $\overline{AA'}$ becomes the segment associated to the unit normal vector. Thus, the normal vector to the triangle defined by A, B, and C can be seen as lying within the same plane that contains the great circle with center at the origin that connects A and A' and perpendicular to $\overline{AA'}$. In Figure 3, the great circle connecting A and A' is represented in light green color.
(5) Applying the result obtained in the 2D case, we already know that if we divide the unit normal vectors by the length of the diameters of the corresponding circumscribed circles, we obtain a

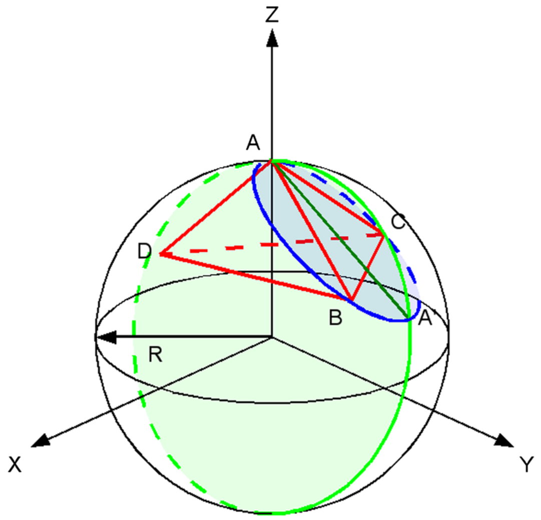


Figure 3. Schematic representation of the 3D construction presented in Section 2.2. A sphere with its center at the origin and three triangles (red) sharing a node located at the intersection of the sphere's surface and the Z axis named north pole (point A) are depicted. Similarly, the circumscribed circle (blue) to the triangle formed by points A, B, and C, as well as its diameter (dark green) through point A are also shown. Finally, also the great circle (light green) that intersects A and A' is shown.

set of normal vectors all having in common the magnitude of their horizontal component (i.e., their projection on the XY plane). This magnitude is $1/2R$.

(6) The only step missing now would be to do a second ponderation that minimizes the sum of all the horizontal projections of the vectors. We propose to use as second complementary factor: the triangle's internal angle at the vertex occupied by the node considered. In fact, other authors have previously used this weighting factor in an isolated manner [14][15].

Equation (9) shows the formula of the weighting factor to apply to the unit normal vector to a triangle $\widehat{ABC}$ considering the inverse of the diameter of the circumscribed circle to the triangle and the internal angle of the triangle in $\hat{A}$.

$$w_{\alpha/\varnothing_\circ} = \frac{\sin\alpha}{\| \overrightarrow{BC} \|} \cdot \alpha \tag{9}$$

where α is calculated as follows:

$$\alpha = \arccos\left(\frac{\overrightarrow{AB} \cdot \overrightarrow{AC}}{\| \overrightarrow{AB} \| \cdot \| \overrightarrow{AC} \|} \right) \tag{10}$$

and $\sin\alpha$ can be computed as

$$\sin\alpha = \frac{\| \overrightarrow{AB} \times \overrightarrow{AC} \|}{\| \overrightarrow{AB} \| \cdot \| \overrightarrow{AC} \|} \tag{11}$$

2.3. *Inverse of the circumscribed circle's area (new)*

The weights constructed following the process described in Section 2.2 rely heavily on the assumption that the surface is sufficiently approximated by a sphere. That is because one of the assumptions is that all the great circles have the same radius R. In the more general case, this hypothesis will not hold.

Therefore, let us propose the use of the area of the circumscribed circle as the magnitude to use. Assuming that smaller circles will provide information more local to P, we will use the inverse of the area as the weighting factor. Equation (12) shows the formula of the weighting factor for

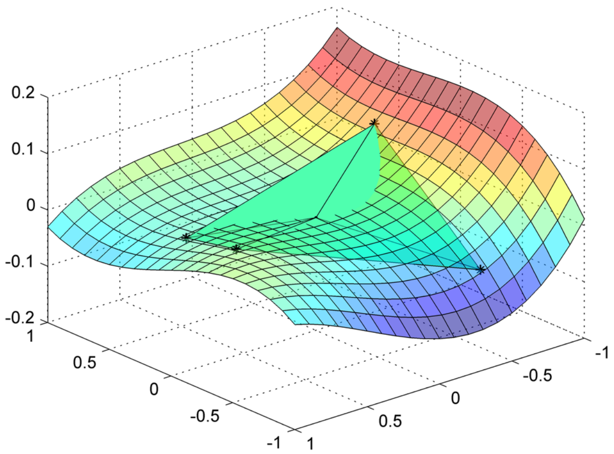


Figure 4. Random cubic surface and four-triangle mesh.

the calculation of the vector normal to a triangle $\widehat{ABC}$ considering the inverse of the area of the circumscribed circle to the triangle. As in Equation (9), we have removed as well any constant coefficients.

$$w_{1/A_{\circ}} = \left(\frac{\sin\alpha}{\| \overrightarrow{BC} \|} \right)^2 \tag{12}$$

2.4. Exact solution for a spherical surface

Max [10] deduces which are the weights that return the exact normals at the nodes of a sphere's triangular mesh. In addition, these weights are extremely simple to compute as shown in Equation (13). This weight is the sine of the internal angle of the $\widehat{ABC}$ triangle in $\hat{A}$ divided by the lengths of the adjacent edges $\overline{AB}$ and $\overline{AC}$. Moreover, in [10], there is a study that shows the properties when approximating the vertex normals by using these weighting factors for an arbitrary set of surfaces.

$$w_{\sin(\alpha)/(b\cdot c)} = \frac{\sin\alpha}{\| \overrightarrow{AB} \| \cdot \| \overrightarrow{AC} \|} \tag{13}$$

2.5. Inverse of the triangle's area

Out of simplicity, different authors [2, 3] have proposed using the inverse of the area of the triangle as weight. Implicitly, the argument is that the smaller the triangle, the more representative its normal becomes with respect to the normal at the node under study. Therefore, the weight applied when computing the average should be larger. However, this argument does not care about the nature of the surface being approximated. Equation (14) shows the formula of the weighting factor for the calculation of the vector normal to a triangle $\widehat{ABC}$ considering the inverse of the triangle's area.

$$w_{1/A_{\triangle}} = \frac{1}{\| \overrightarrow{AB} \times \overrightarrow{AC} \|} \tag{14}$$

2.6. Other weights tested

In order to better understand the weight proposed in Section 2.2, we have decomposed it into its two main components (see Equations (15) and (16)) and introduced them into the numerical analysis presented on Section 3. After drawing preliminary conclusions, we have also proposed a new

DOI: 10.1002/nme

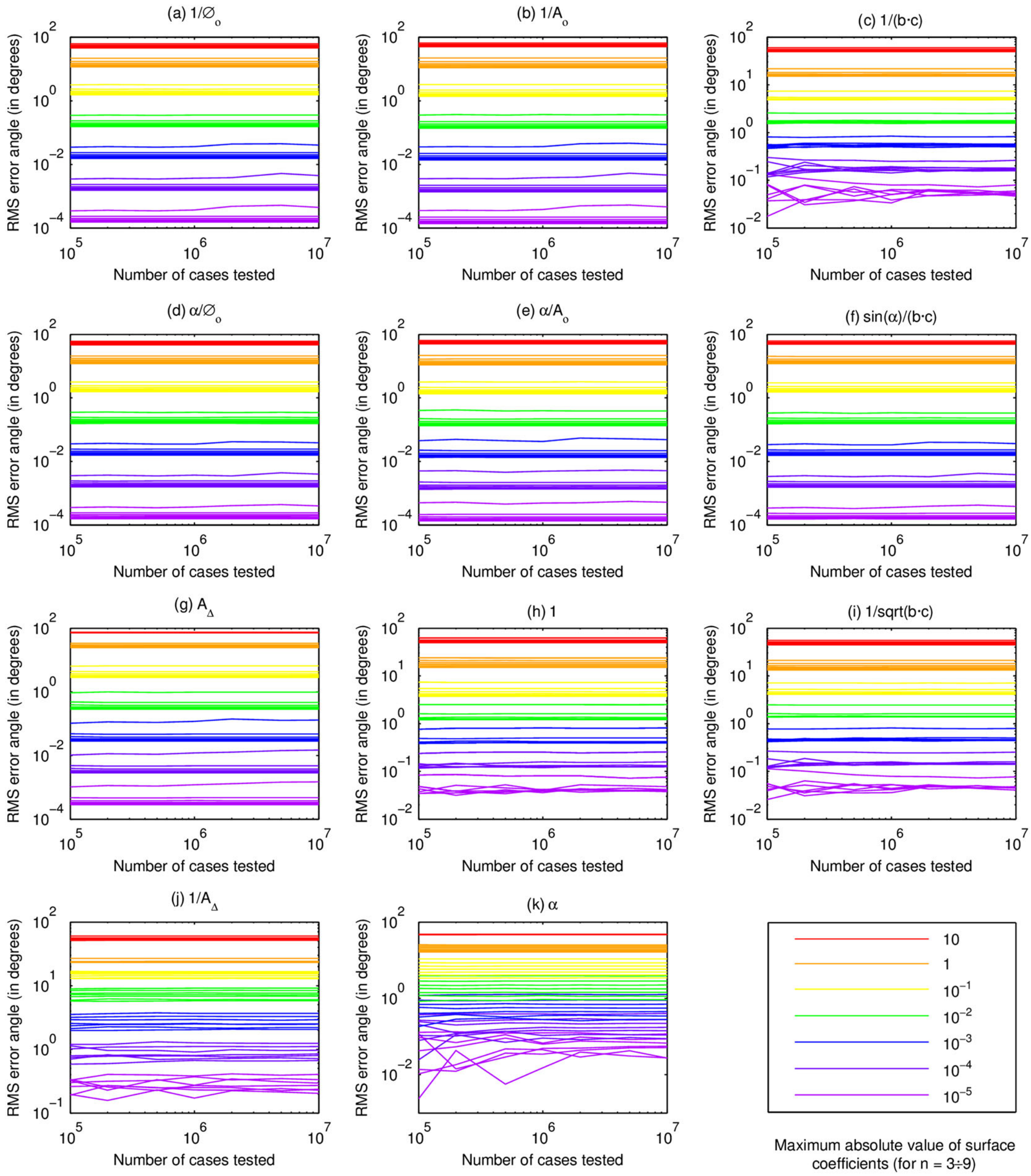


Figure 5. Convergence plots depending on the number of cases considered (n_S) for each one of the 11 weighting factors. (a) $1/\varnothing_\circ$, (b) $1/A_\circ$, (c) $1/(b \cdot c)$, (d) $\alpha/\varnothing_\circ$, (e) $\alpha/A_\circ$, (f) $\sin(\alpha)/(b \cdot c)$, (g) A_Δ, (h) 1, (i) $1/\sqrt{(b-c)}$, (j) $1/A_\Delta$, and (k) α. RMS, root-mean-square.

weighting factor expressed by Equation (17).

$$w_{1/\varnothing_\circ} = \frac{\sin\alpha}{\|\overrightarrow{BC}\|} \tag{15}$$

$$w_\alpha = \alpha \tag{16}$$

$$w_{\alpha/A_\circ} = \left(\frac{\sin\alpha}{\|\overrightarrow{BC}\|}\right)^2 \cdot \alpha \tag{17}$$

Additional weights included in the analysis are the area of the triangle (Equation (18)) that was studied by Meek and Walton [6], equal weights (Equation (19)) suggested by Gouraud [16], and finally two other weights (Equations (20) and (21)) as tested by Max [10].

$$w_{A_\Delta} = \|\overrightarrow{AB} \times \overrightarrow{AC}\| \tag{18}$$

$$w_1 = 1 \tag{19}$$

$$w_{1/(b\cdot c)} = \frac{1}{\|\overrightarrow{AB}\| \cdot \|\overrightarrow{AC}\|} \tag{20}$$

$$w_{1/\sqrt{(b\cdot c)}} = \frac{1}{\sqrt{\|\overrightarrow{AB}\| \cdot \|\overrightarrow{AC}\|}} \tag{21}$$

3. NUMERICAL ANALYSIS

With the goal of comparing and verifying the goodness and accuracy of the previous weighting factors, a numerical analysis has been performed with all of them. The starting point of this analysis is the methodology used by Max in [10], conveniently extended. We have chosen this methodology because it allows us to reproduce asymptotic behavior as studied by Meek and Walton [6] but at the

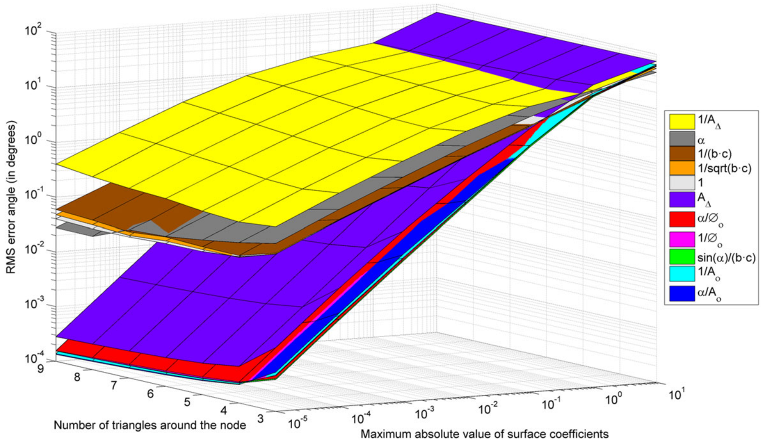


Figure 6. Errors made in the approximation of the vertex normals for the 11 weighting factors depending on the k_{max} and n_p. RMS, root-mean-square.

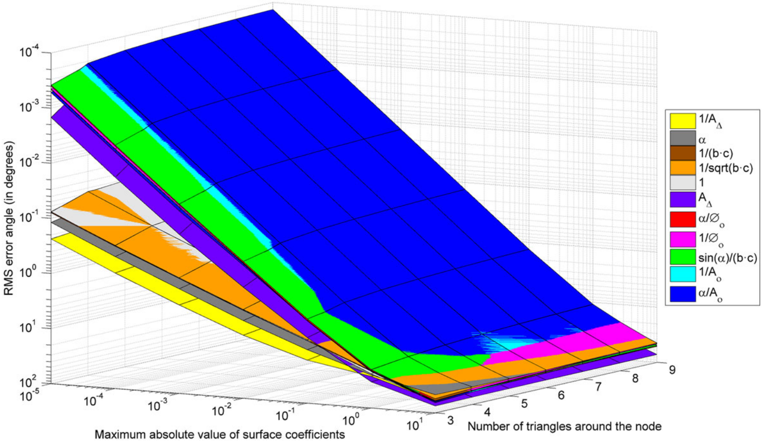


Figure 7. Errors made in the approximation of the vertex normals for the 11 weighting factors depending on the k_{max} and n_p. Vertical axis reversed. RMS, root-mean-square.

DOI: 10.1002/nme

same time retains the capability of finding differences between weights with equivalent asymptotic properties without loss of generality.

First of all, cubic surfaces of the form

$$\phi(x,y) = Ax^2 + Bxy + Cy^2 + Dx^3 + Ex^2y + Fxy^2 + Gy^3 \tag{22}$$

have been randomly generated. By construction, the origin of coordinates belongs to the surface, and the normal at this point coincides with the Z axis, which will be taken as a reference. Because one of the goals of the study is to evaluate the influence of the surface's curvature on the precision of the calculation of the vertex normal, the coefficients of this surface will be taken as follows:

$$\{A, B, C, D, E, F, G\} \in U(-k_{max}, k_{max}) \tag{23}$$

where U refers to the uniform distribution of probability in the indicated interval. The value of k_{max} will be set according to the following set of values:

$$k_{max} = \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10\} \tag{24}$$

Secondly, when the surface has been obtained, a mesh of n_p triangles is built. This mesh is obtained after determining the number of surrounding points ($n_p = 3 \div 9$) to the origin of coordinates

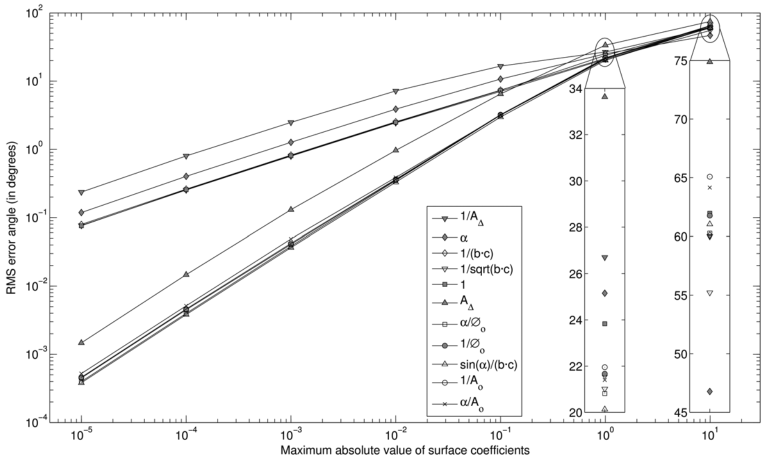


Figure 8. Errors made in the approximation of the vertex normals for the 11 weighting factors depending on k_{max}, for $n_p = 3$. RMS, root-mean-square.

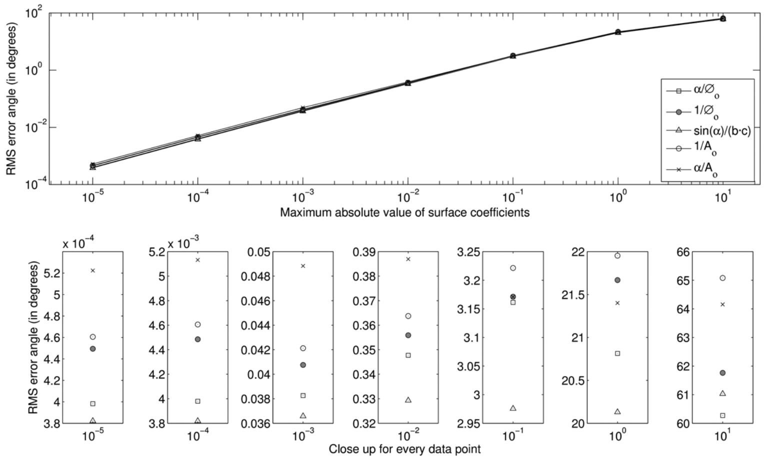


Figure 9. Errors made in the approximation of the vertex normals for the best 5 weighting factors depending on k_{max}, for $n_p = 3$. RMS, root-mean-square.

DOI: 10.1002/nme

and placed on the surface $\phi(x, y)$. In this way, several kinds of meshes are taken into account. With this purpose, n_p vertices are generated in polar coordinates (r, θ, z) surrounding the origin. The values of r and θ are randomly and uniformly assigned in the intervals $(0, 1]$ and $[0, 2\pi]$, respectively. The values of θ are sorted in increasing order. The angles α_i at the internal vertex for each of the n_p triangles can be obtained as the difference between the angle θ_{i+1} and the previous one θ_i, rejecting the cases where the result is greater than π (otherwise the angle would not belong to a triangle). Translating the vertices to Cartesian coordinates $(r, \theta) \rightarrow (x, y)$, these are then located on the surface according to $z = \phi(x, y)$.

With the aim of carrying out a statistical research, a big enough number of n_s random surfaces must be generated. Because we need to determine the number of surfaces n_s that is statistically representative for the study, the following sequence of values is set:

$$n_s = \{10^5, 2 \cdot 10^5, 5 \cdot 10^5, 10^6, 2 \cdot 10^6, 5 \cdot 10^6, 10^7\} \tag{25}$$

A statistical convergence analysis using this sequence is performed to determine how many surfaces are required to be taken into account in the study (Figure 5). As an example, a random surface

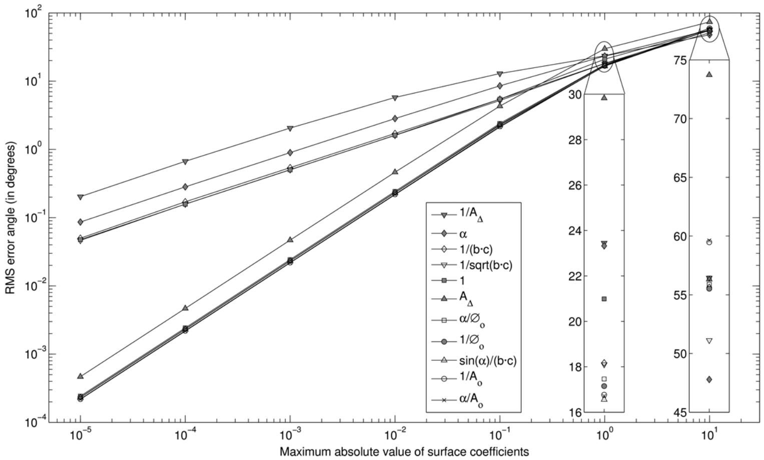


Figure 10. Errors made in the approximation of the vertex normals for the 11 weighting factors depending on k_{max}, for $n_p = 4$. RMS, root-mean-square.

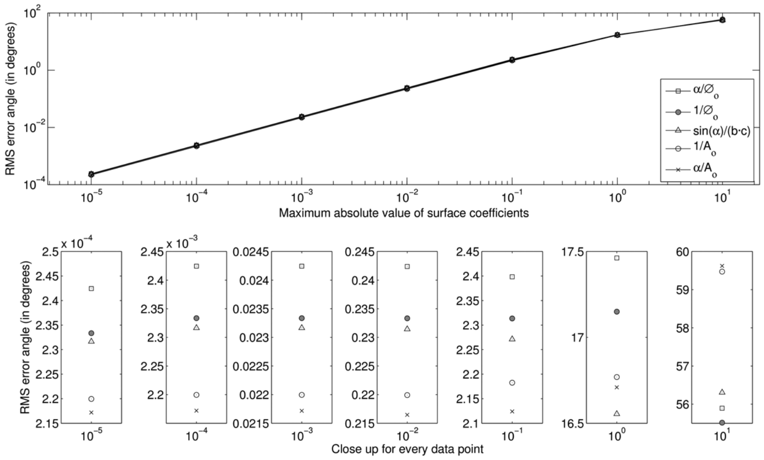


Figure 11. Errors made in the approximation of the vertex normals for the best 5 weighting factors depending on k_{max}, for $n_p = 4$. RMS, root-mean-square.

DOI: 10.1002/nme

is shown in Figure 4 along with a randomly generated triangular mesh (in this case, four triangles are drawn).

Once the surface and its associated triangular mesh are obtained, the approximated normal vector in the node is calculated as the sum of the unit normals of each triangle multiplied by the weighting factor chosen in each case, as it is shown in Equation (26).

$$N_j = \sum_{i=1}^{n_p} w^i \hat{N}_i^{tri} \tag{26}$$

where w^i is the weighting factor chosen and $\hat{N}_i^{tri}$ is the unit normal of the i-th triangle. The approximated unit normal must be calculated a posteriori, if needed.

After the generation of the n_s surfaces and the calculation of the corresponding approximated vertex normals, the angular deviations with respect to the exact normal are computed, and the

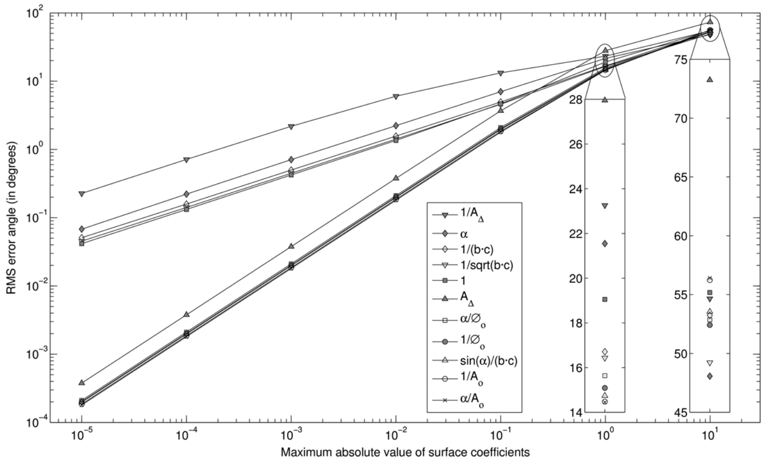


Figure 12. Errors made in the approximation of the vertex normals for the 11 weighting factors depending on k_{max}, for $n_p = 5$. RMS, root-mean-square.

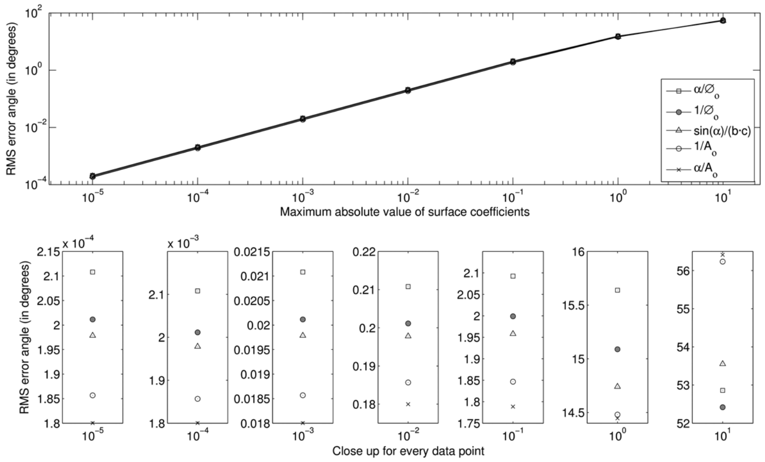


Figure 13. Errors made in the approximation of the vertex normals for the best 5 weighting factors depending on k_{max}, for $n_p = 5$. RMS, root-mean-square.

DOI: 10.1002/nme

root-mean-square error (RMSE) angle in degrees is estimated as shown in Equation (27) for each weighting factor.

$$RMSE = \sqrt{\sum_{j=1}^{n_s} \frac{d_j^2}{n_s}} \tag{27}$$

where d_j are the deviations—in degrees—of the normal relative to each one of the n_s surfaces, as it is shown in Equation (28):

$$d_j = \arccos(\hat{\boldsymbol{N}}_{\boldsymbol{z}} \cdot \hat{\boldsymbol{N}}_{\boldsymbol{j}}) \cdot \frac{180}{\pi} \tag{28}$$

The exact normal to the surface is parallel to the Z axis, $\hat{\boldsymbol{N}}_{\boldsymbol{z}} = (0, 0, 1)$, and $\hat{\boldsymbol{N}}_{\boldsymbol{j}}$ is the approximated normal vector calculated in each case by using the corresponding weighting factor and normalized properly.

As seen earlier, three variables have been considered in the calculations:

- k_{max}: maximum absolute value of surface coefficients

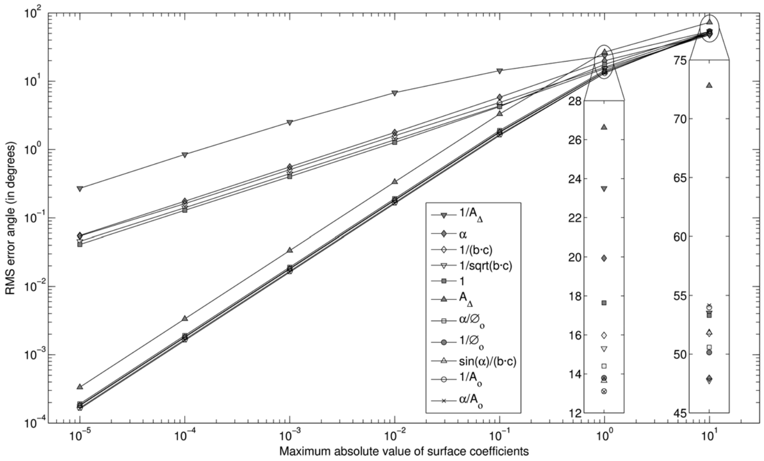


Figure 14. Errors made in the approximation of the vertex normals for the 11 weighting factors depending on k_{max}, for $n_p = 6$. RMS, root-mean-square.

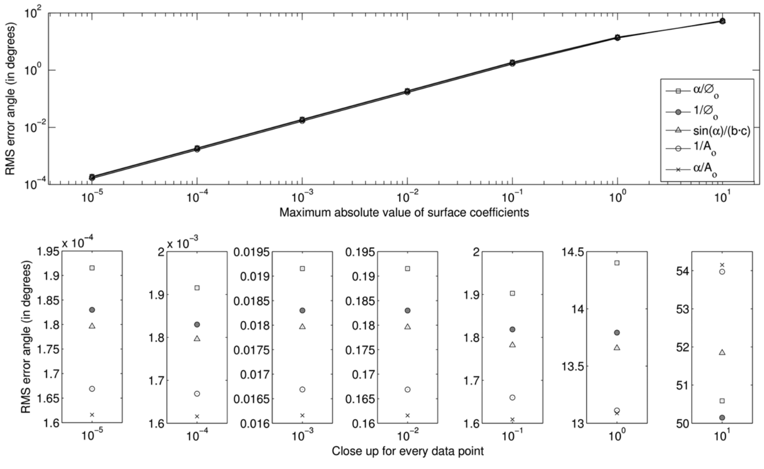


Figure 15. Errors made in the approximation of the vertex normals for the best 5 weighting factors depending on k_{max}, for $n_p = 6$. RMS, root-mean-square.

DOI: 10.1002/nme

- n_p: number of triangles around the node whose normal vector is estimated
- n_s: number of surfaces generated

Considering that each variable can take seven different values, all their possible combinations have been calculated (7^3), obtaining in each case the angular error for the 11 weighting factors.

The first aspect that needs to be determined is which is the minimum value of n_s required to provide statistically significant results with a small interval of confidence. With this objective, a set of convergence plots has been carried out (Figure 5), showing the error made in the approximation of the normal for

- each weighting factor w,
- each value of k_{max}, and
- for every value of n_p.

As it is shown in the legend, the different values of k_{max}—which modulate the degree of curvature of the surfaces—are separated in color strips. Each of these strips is composed by a set of curves, representing each one a value of n_p. The variable n_s is represented in the horizontal axis. The results obtained show a group of six factors (Figure 5(a), (b), and (d)–(g)) with very little variation

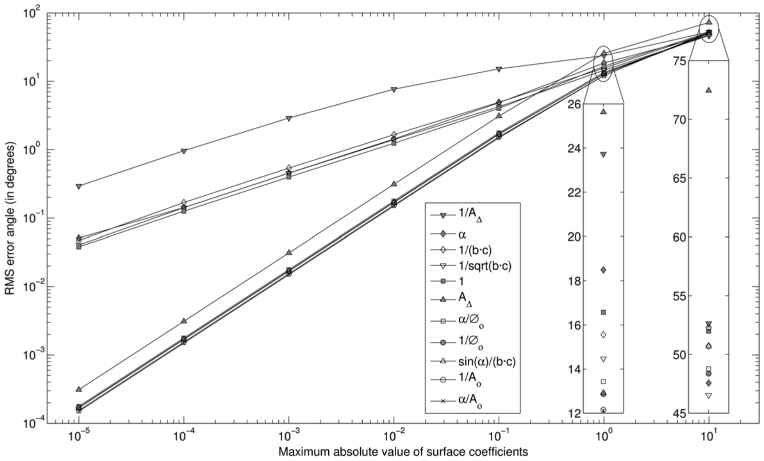


Figure 16. Errors made in the approximation of the vertex normals for the 11 weighting factors depending on k_{max}, for $n_p = 7$. RMS, root-mean-square.

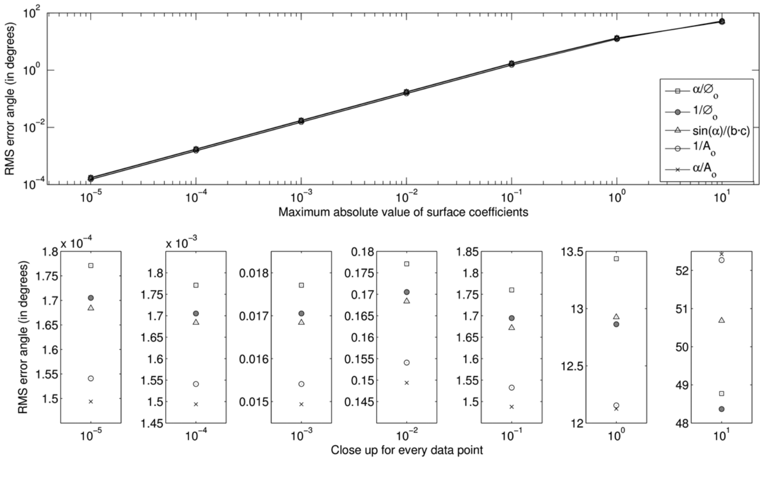


Figure 17. Errors made in the approximation of the vertex normals for the best 5 weighting factors depending on k_{max}, for $n_p = 7$. RMS, root-mean-square.

and bounded errors, except for the cases where the number of triangles around the node is equal to 3, which seem to not present good convergence. This fact offers a first conclusion: The confidence interval of the results corresponding to meshes with $n_p = 3$ needs to be greater than that for the rest of the values of n_p. On the other side, the other five factors (Figure 5(c) and (h)–(k)) show bigger errors and a worse convergence to reach a stable value, with high oscillations for a lower number of cases. Nevertheless, if the results obtained with the generation of 10 million are taken into account, this second group of factors seems to converge. For this reason, from now on, the rest of the study will be performed considering always the results obtained with 10 million surfaces generated.

4. RESULTS

In this section, the authors will represent the results obtained in the study using $n_s = 10^7$ surfaces for each of the cases considered and varying the values of k_{max} and n_p.

The domain of study can be visualized in Figure 6 ($n_p = 3 \div 9$; $k_{max} = \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$) and the RMSE obtained for each of the weighting factors in this domain. The factors have been sorted depending on the errors obtained for most of the domain of study, so at the top of the legend, there is indicated the worst ranked factor, and at the bottom, the best one. In order to

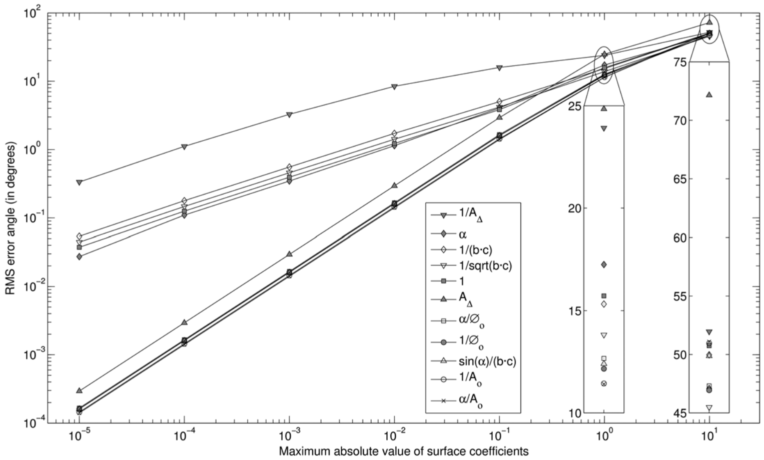


Figure 18. Errors made in the approximation of the vertex normals for the 11 weighting factors depending on k_{max}, for $n_p = 8$. RMS, root-mean-square.

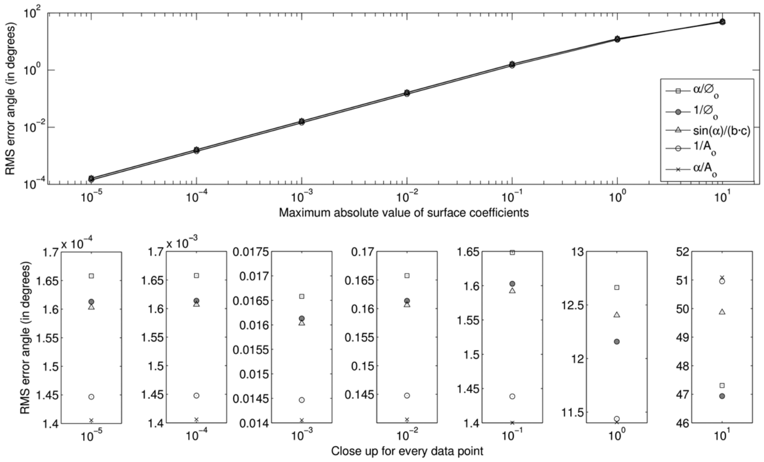


Figure 19. Errors made in the approximation of the vertex normals for the best 5 weighting factors depending on k_{max}, for $n_p = 8$. RMS, root-mean-square.

DOI: 10.1002/nme

show a complementary view of the domain of study and the results, the same plot is represented in Figure 7 but with the vertical axis reversed.

As commented in Section 3, there are two main groups of weighting factors with similar behavior. It also can be observed that for the situation where k_{max}—which controls the surface's curvature—takes values around 1 and 10, the errors made by all the factors have similar orders of magnitude. Nevertheless, it should be considered that in these cases, the error is huge (several tens of degrees).

Likewise, it can be seen that the cases with three triangles around the node ($n_p = 3$) generate bigger errors and even alter the order of the weighting factors. This result deserves some comment. The two factors that deliver the best results in the case where $n_p = 3$ are the two factors with the assumption that the surface S is approximated by a sphere (Equations (9) and (13)). And for values of $n_p > 3$, their precision drops. This can be explained because in general, a sphere will not approximate well a surface represented by more than three triangles. It is worth mentioning that a regular mesh of triangles features six triangles around each node.

Looking at Figure 7, the authors observe clearly that the most precise weighting factor is the one given by the combination of the triangle's interior angle and the inverse of the circumscribed circumference's area related to this triangle (Equation (17)).

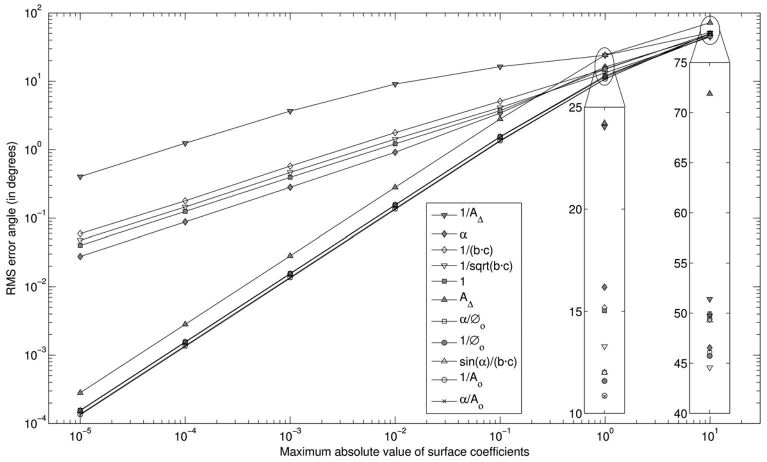


Figure 20. Errors made in the approximation of the vertex normals for the 11 weighting factors depending on k_{max}, for $n_p = 9$. RMS, root-mean-square.

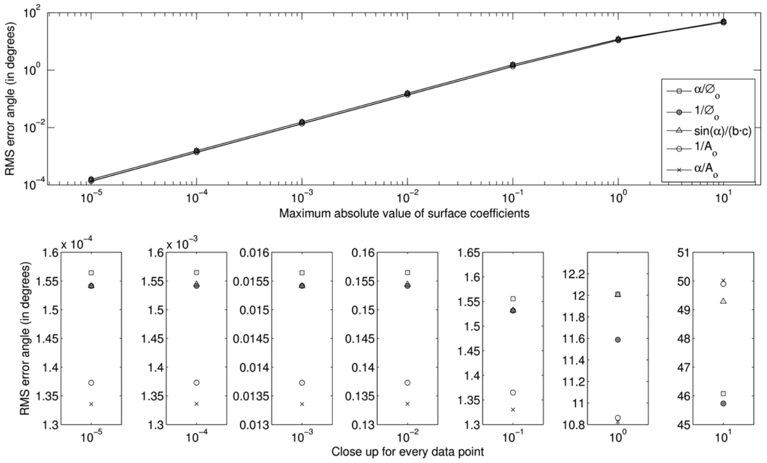


Figure 21. Errors made in the approximation of the vertex normals for the best 5 weighting factors depending on k_{max}, for $n_p = 9$. RMS, root-mean-square.

DOI: 10.1002/nme

Moreover, it must be emphasized that contrary to our intuition, the weighting factor that uses the inverse of the triangle's area is the worst of all of those considered, as shown by the results.

For more details, in Figures 8 to 22, several cuts of the surfaces of Figure 6 are shown, according to the number of triangles. In this way, each figure represents the RMSE depending on the k_{max} for every value of n_p. To achieve better clarity, detailed figures with the five best weighting factors are also included. See also in Figure 22 a cut of Figure 6, showing the results obtained for the surfaces with a value of $k_{max} = 10^{-1}$. This figure shows that for all the factors except for two,—inverse of the triangle's area and inverse of the product of the adjacent edges (this one diverging slightly)—the RMSE when computing the vertex normals is reduced as the number of triangles in the mesh increases. Seeing Figure 22, it is easy to understand the combination of the factor in Equation (16) with the factor in Equation (12). Because the weight that uses the internal angle as a factor is the one that improves the most as the number of triangles increases. This justifies the directional value of the information provided by the angle. In Figure 23 the five best weighting factors are shown, with their corresponding detailed plots.

A noteworthy result is that all four new weights proposed in this study (see Equations (9), (12), (15), and (17)) are among the five factors showing a better behavior (see, e.g., the excellent correlation displayed in Figures 9, 11, 13, 15, 17, 19, and 21). This result seems to validate the three principal hypotheses set forth in the framework presented in Section 1.2:

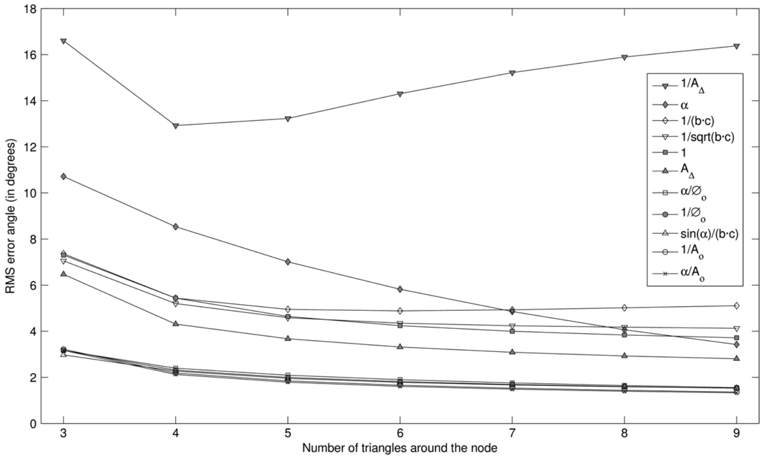


Figure 22. Errors made in the approximation of the vertex normals for the 11 weighting factors depending on n_p, for $k_{max} = 10^{-1}$. RMS, root-mean-square.

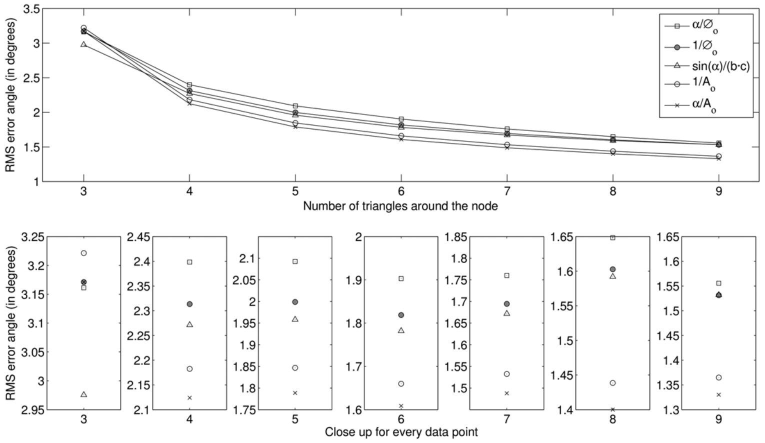


Figure 23. Errors made in the approximation of the vertex normals for the best 5 weighting factors depending on n_p, for $k_{max} = 10^{-1}$.

DOI: 10.1002/nme

Table I. Errors made in the approximation of the vertex normals for the 11 weighting factors f depending on k_{max} and n_p, for $n_s = 10^7$.

w	n_p							k_{max}
	3	4	5	6	7	8	9	
$1/A_\triangle$	$2.362\cdot 10^{-1}$	$2.031\cdot 10^{-1}$	$2.269\cdot 10^{-1}$	$2.702\cdot 10^{-1}$	$2.935\cdot 10^{-1}$	$3.346\cdot 10^{-1}$	$4.031\cdot 10^{-1}$	10^{-5}
α	$1.182\cdot 10^{-1}$	$8.620\cdot 10^{-2}$	$6.784\cdot 10^{-2}$	$5.528\cdot 10^{-2}$	$5.118\cdot 10^{-2}$	$2.735\cdot 10^{-2}$	$2.765\cdot 10^{-2}$	
$1/(b\cdot c)$	$7.918\cdot 10^{-2}$	$4.998\cdot 10^{-2}$	$5.094\cdot 10^{-2}$	$5.431\cdot 10^{-2}$	$4.700\cdot 10^{-2}$	$5.392\cdot 10^{-2}$	$5.944\cdot 10^{-2}$	
$1/\sqrt{(b\cdot c)}$	$7.652\cdot 10^{-2}$	$4.632\cdot 10^{-2}$	$4.537\cdot 10^{-2}$	$4.563\cdot 10^{-2}$	$4.050\cdot 10^{-2}$	$4.442\cdot 10^{-2}$	$4.773\cdot 10^{-2}$	
1	$7.573\cdot 10^{-2}$	$4.761\cdot 10^{-2}$	$4.166\cdot 10^{-2}$	$4.103\cdot 10^{-2}$	$3.793\cdot 10^{-2}$	$3.744\cdot 10^{-2}$	$4.000\cdot 10^{-2}$	
$A_\triangle$	$1.473\cdot 10^{-3}$	$4.679\cdot 10^{-4}$	$3.772\cdot 10^{-4}$	$3.353\cdot 10^{-4}$	$3.108\cdot 10^{-4}$	$2.942\cdot 10^{-4}$	$2.817\cdot 10^{-4}$	
$\alpha/\varnothing_\circ$	$3.982\cdot 10^{-4}$	$2.424\cdot 10^{-4}$	$2.108\cdot 10^{-4}$	$1.915\cdot 10^{-4}$	$1.771\cdot 10^{-4}$	$1.658\cdot 10^{-4}$	$1.564\cdot 10^{-4}$	
$1/\varnothing_\circ$	$4.494\cdot 10^{-4}$	$2.334\cdot 10^{-4}$	$2.012\cdot 10^{-4}$	$1.830\cdot 10^{-4}$	$1.705\cdot 10^{-4}$	$1.613\cdot 10^{-4}$	$1.541\cdot 10^{-4}$	
$\sin(\alpha)/(b\cdot c)$	$3.821\cdot 10^{-4}$	$2.316\cdot 10^{-4}$	$1.978\cdot 10^{-4}$	$1.796\cdot 10^{-4}$	$1.684\cdot 10^{-4}$	$1.603\cdot 10^{-4}$	$1.541\cdot 10^{-4}$	
$1/A_\circ$	$4.605\cdot 10^{-4}$	$2.199\cdot 10^{-4}$	$1.857\cdot 10^{-4}$	$1.669\cdot 10^{-4}$	$1.541\cdot 10^{-4}$	$1.447\cdot 10^{-4}$	$1.373\cdot 10^{-4}$	
$\alpha/A_\circ$	$5.224\cdot 10^{-4}$	$2.172\cdot 10^{-4}$	$1.800\cdot 10^{-4}$	$1.616\cdot 10^{-4}$	$1.494\cdot 10^{-4}$	$1.406\cdot 10^{-4}$	$1.336\cdot 10^{-4}$	
$1/A_\triangle$	$7.998\cdot 10^{-1}$	$6.688\cdot 10^{-1}$	$7.132\cdot 10^{-1}$	$8.472\cdot 10^{-1}$	$9.641\cdot 10^{-1}$	1.110	1.242	10^{-4}
α	$4.016\cdot 10^{-1}$	$2.822\cdot 10^{-1}$	$2.215\cdot 10^{-1}$	$1.748\cdot 10^{-1}$	$1.418\cdot 10^{-1}$	$1.096\cdot 10^{-1}$	$8.799\cdot 10^{-2}$	
$1/(b\cdot c)$	$2.610\cdot 10^{-1}$	$1.709\cdot 10^{-1}$	$1.581\cdot 10^{-1}$	$1.612\cdot 10^{-1}$	$1.682\cdot 10^{-1}$	$1.782\cdot 10^{-1}$	$1.787\cdot 10^{-1}$	
$1/\sqrt{(b\cdot c)}$	$2.528\cdot 10^{-1}$	$1.577\cdot 10^{-1}$	$1.409\cdot 10^{-1}$	$1.404\cdot 10^{-1}$	$1.419\cdot 10^{-1}$	$1.467\cdot 10^{-1}$	$1.455\cdot 10^{-1}$	
1	$2.565\cdot 10^{-1}$	$1.574\cdot 10^{-1}$	$1.320\cdot 10^{-1}$	$1.288\cdot 10^{-1}$	$1.257\cdot 10^{-1}$	$1.254\cdot 10^{-1}$	$1.252\cdot 10^{-1}$	
$A_\triangle$	$1.467\cdot 10^{-2}$	$4.679\cdot 10^{-3}$	$3.772\cdot 10^{-3}$	$3.353\cdot 10^{-3}$	$3.108\cdot 10^{-3}$	$2.939\cdot 10^{-3}$	$2.818\cdot 10^{-3}$	
$\alpha/\varnothing_\circ$	$3.980\cdot 10^{-3}$	$2.424\cdot 10^{-3}$	$2.108\cdot 10^{-3}$	$1.915\cdot 10^{-3}$	$1.771\cdot 10^{-3}$	$1.658\cdot 10^{-3}$	$1.565\cdot 10^{-3}$	
$1/\varnothing_\circ$	$4.485\cdot 10^{-3}$	$2.334\cdot 10^{-3}$	$2.012\cdot 10^{-3}$	$1.830\cdot 10^{-3}$	$1.705\cdot 10^{-3}$	$1.614\cdot 10^{-3}$	$1.542\cdot 10^{-3}$	
$\sin(\alpha)/(b\cdot c)$	$3.819\cdot 10^{-3}$	$2.316\cdot 10^{-3}$	$1.978\cdot 10^{-3}$	$1.796\cdot 10^{-3}$	$1.684\cdot 10^{-3}$	$1.607\cdot 10^{-3}$	$1.544\cdot 10^{-3}$	
$1/A_\circ$	$4.605\cdot 10^{-3}$	$2.200\cdot 10^{-3}$	$1.857\cdot 10^{-3}$	$1.669\cdot 10^{-3}$	$1.541\cdot 10^{-3}$	$1.448\cdot 10^{-3}$	$1.373\cdot 10^{-3}$	
$\alpha/A_\circ$	$5.132\cdot 10^{-3}$	$2.172\cdot 10^{-3}$	$1.800\cdot 10^{-3}$	$1.616\cdot 10^{-3}$	$1.494\cdot 10^{-3}$	$1.406\cdot 10^{-3}$	$1.336\cdot 10^{-3}$	
$1/A_\triangle$	2.481	2.055	2.183	2.516	2.904	3.275	3.665	10^{-3}
α	1.266	$8.960\cdot 10^{-1}$	$7.082\cdot 10^{-1}$	$5.567\cdot 10^{-1}$	$4.534\cdot 10^{-1}$	$3.455\cdot 10^{-1}$	$2.822\cdot 10^{-1}$	
$1/(b\cdot c)$	$8.224\cdot 10^{-1}$	$5.391\cdot 10^{-1}$	$4.991\cdot 10^{-1}$	$5.074\cdot 10^{-1}$	$5.368\cdot 10^{-1}$	$5.544\cdot 10^{-1}$	$5.753\cdot 10^{-1}$	
$1/\sqrt{(b\cdot c)}$	$7.939\cdot 10^{-1}$	$5.037\cdot 10^{-1}$	$4.476\cdot 10^{-1}$	$4.413\cdot 10^{-1}$	$4.527\cdot 10^{-1}$	$4.564\cdot 10^{-1}$	$4.648\cdot 10^{-1}$	
1	$8.081\cdot 10^{-1}$	$5.013\cdot 10^{-1}$	$4.219\cdot 10^{-1}$	$4.008\cdot 10^{-1}$	$3.964\cdot 10^{-1}$	$3.935\cdot 10^{-1}$	$3.945\cdot 10^{-1}$	
$A_\triangle$	$1.304\cdot 10^{-1}$	$4.678\cdot 10^{-2}$	$3.772\cdot 10^{-2}$	$3.353\cdot 10^{-2}$	$3.108\cdot 10^{-2}$	$2.942\cdot 10^{-2}$	$2.817\cdot 10^{-2}$	
$\alpha/\varnothing_\circ$	$3.826\cdot 10^{-2}$	$2.424\cdot 10^{-2}$	$2.108\cdot 10^{-2}$	$1.915\cdot 10^{-2}$	$1.771\cdot 10^{-2}$	$1.658\cdot 10^{-2}$	$1.564\cdot 10^{-2}$	
$1/\varnothing_\circ$	$4.076\cdot 10^{-2}$	$2.334\cdot 10^{-2}$	$2.012\cdot 10^{-2}$	$1.830\cdot 10^{-2}$	$1.705\cdot 10^{-2}$	$1.613\cdot 10^{-2}$	$1.541\cdot 10^{-2}$	
$\sin(\alpha)/(b\cdot c)$	$3.658\cdot 10^{-2}$	$2.316\cdot 10^{-2}$	$1.978\cdot 10^{-2}$	$1.796\cdot 10^{-2}$	$1.684\cdot 10^{-2}$	$1.603\cdot 10^{-2}$	$1.541\cdot 10^{-2}$	
$1/A_\circ$	$4.212\cdot 10^{-2}$	$2.199\cdot 10^{-2}$	$1.857\cdot 10^{-2}$	$1.669\cdot 10^{-2}$	$1.541\cdot 10^{-2}$	$1.447\cdot 10^{-2}$	$1.373\cdot 10^{-2}$	
$\alpha/A_\circ$	$4.883\cdot 10^{-2}$	$2.172\cdot 10^{-2}$	$1.800\cdot 10^{-2}$	$1.616\cdot 10^{-2}$	$1.494\cdot 10^{-2}$	$1.406\cdot 10^{-2}$	$1.336\cdot 10^{-2}$	
$1/A_\triangle$	7.173	5.766	6.022	6.797	7.654	8.420	9.113	10^{-2}
α	3.880	2.826	2.226	1.778	1.431	1.139	$9.155\cdot 10^{-1}$	
$1/(b\cdot c)$	2.534	1.709	1.568	1.590	1.653	1.723	1.772	
$1/\sqrt{(b\cdot c)}$	2.443	1.603	1.415	1.385	1.397	1.415	1.428	
1	2.493	1.610	1.352	1.268	1.238	1.222	1.215	
$A_\triangle$	$9.650\cdot 10^{-1}$	$4.631\cdot 10^{-1}$	$3.766\cdot 10^{-1}$	$3.352\cdot 10^{-1}$	$3.107\cdot 10^{-1}$	$2.939\cdot 10^{-1}$	$2.818\cdot 10^{-1}$	
$\alpha/\varnothing_\circ$	$3.477\cdot 10^{-1}$	$2.424\cdot 10^{-1}$	$2.108\cdot 10^{-1}$	$1.915\cdot 10^{-1}$	$1.771\cdot 10^{-1}$	$1.658\cdot 10^{-1}$	$1.565\cdot 10^{-1}$	
$1/\varnothing_\circ$	$3.559\cdot 10^{-1}$	$2.333\cdot 10^{-1}$	$2.011\cdot 10^{-1}$	$1.830\cdot 10^{-1}$	$1.705\cdot 10^{-1}$	$1.614\cdot 10^{-1}$	$1.541\cdot 10^{-1}$	
$\sin(\alpha)/(b\cdot c)$	$3.293\cdot 10^{-1}$	$2.314\cdot 10^{-1}$	$1.978\cdot 10^{-1}$	$1.796\cdot 10^{-1}$	$1.684\cdot 10^{-1}$	$1.606\cdot 10^{-1}$	$1.544\cdot 10^{-1}$	
$1/A_\circ$	$3.637\cdot 10^{-1}$	$2.199\cdot 10^{-1}$	$1.857\cdot 10^{-1}$	$1.669\cdot 10^{-1}$	$1.541\cdot 10^{-1}$	$1.448\cdot 10^{-1}$	$1.373\cdot 10^{-1}$	
$\alpha/A_\circ$	$3.870\cdot 10^{-1}$	$2.165\cdot 10^{-1}$	$1.800\cdot 10^{-1}$	$1.616\cdot 10^{-1}$	$1.494\cdot 10^{-1}$	$1.406\cdot 10^{-1}$	$1.336\cdot 10^{-1}$	
$1/A_\triangle$	16.61	12.92	13.23	14.30	15.22	15.90	16.38	$\mathbf{10^{-1}}$
α	10.72	8.536	7.011	5.824	4.858	4.064	3.427	
$1/(b\cdot c)$	7.362	5.439	4.949	4.884	4.931	5.018	5.109	
$1/\sqrt{(b\cdot c)}$	7.055	5.203	4.586	4.352	4.240	4.176	4.131	
1	7.305	5.439	4.643	4.242	4.000	3.837	3.719	
$A_\triangle$	6.472	4.309	3.674	3.317	3.089	2.930	2.808	
$\alpha/\varnothing_\circ$	3.161	2.398	2.093	1.903	1.760	1.648	1.556	
$1/\varnothing_\circ$	3.171	2.314	1.999	1.819	1.695	1.603	1.531	
$\sin(\alpha)/(b\cdot c)$	2.975	2.271	1.958	1.782	1.672	1.592	1.532	
$1/A_\circ$	3.221	2.183	1.847	1.660	1.533	1.439	1.365	
$\alpha/A_\circ$	3.171	2.124	1.789	1.609	1.488	1.400	1.330	
$1/A_\triangle$	26.71	23.45	23.25	23.50	23.73	23.91	24.03	**1**
α	25.15	23.32	21.55	19.93	18.49	17.25	16.19	
$1/(b\cdot c)$	21.60	18.17	16.71	15.98	15.56	15.32	15.17	
$1/\sqrt{(b\cdot c)}$	21.01	18.10	16.43	15.31	14.47	13.82	13.29	
1	23.83	20.99	19.05	17.64	16.57	15.72	15.03	
$A_\triangle$	33.63	29.82	27.96	26.62	25.63	24.84	24.22	
$\alpha/\varnothing_\circ$	20.81	17.46	15.64	14.40	13.44	12.66	12.01	
$1/\varnothing_\circ$	21.67	17.15	15.09	13.79	12.86	12.16	11.59	
$\sin(\alpha)/(b\cdot c)$	20.13	16.55	14.74	13.66	12.93	12.40	12.00	
$1/A_\circ$	21.95	16.77	14.48	13.11	12.15	11.44	10.86	
$\alpha/A_\circ$	21.40	16.71	14.45	13.09	12.12	11.41	10.83	
$1/A_\triangle$	60.00	56.44	54.66	53.51	52.64	51.96	51.41	**10**
α	46.78	47.79	48.07	47.94	47.56	47.08	46.55	
$1/(b\cdot c)$	60.11	55.64	53.24	51.76	50.72	49.97	49.41	
$1/\sqrt{(b\cdot c)}$	55.22	51.12	49.23	47.75	46.53	45.48	44.58	
1	61.96	56.42	55.18	53.29	51.98	50.75	49.72	
$A_\triangle$	74.88	73.71	73.24	72.80	72.45	72.14	71.89	
$\alpha/\varnothing_\circ$	60.27	55.89	52.86	50.59	48.77	47.31	46.08	
$1/\varnothing_\circ$	61.76	55.51	52.42	50.15	48.37	46.94	45.73	
$\sin(\alpha)/(b\cdot c)$	61.03	56.31	53.55	51.85	50.68	49.87	49.29	
$1/A_\circ$	65.08	59.47	56.23	53.97	52.27	50.95	49.90	
$\alpha/A_\circ$	64.16	59.62	56.41	54.15	52.43	51.09	50.02	

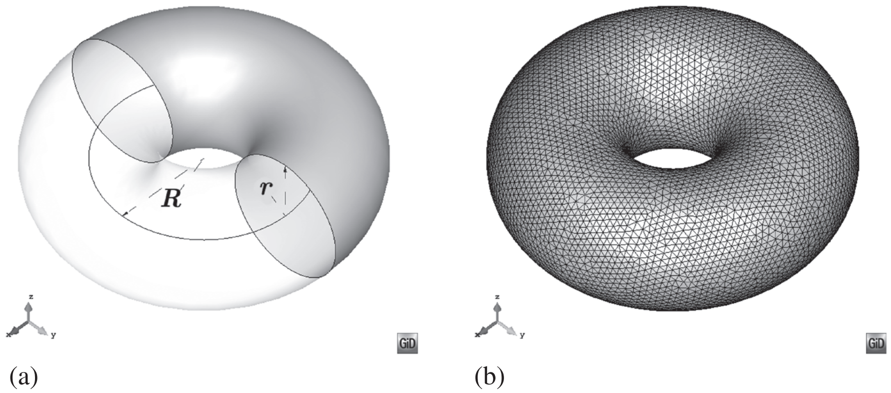


Figure 24. Torus used for the practical example. The normals are computed at every node of the mesh. (a) Geometric definition and (b) view of the mesh generated. The mesh has 7083 nodes and 14,166 elements.

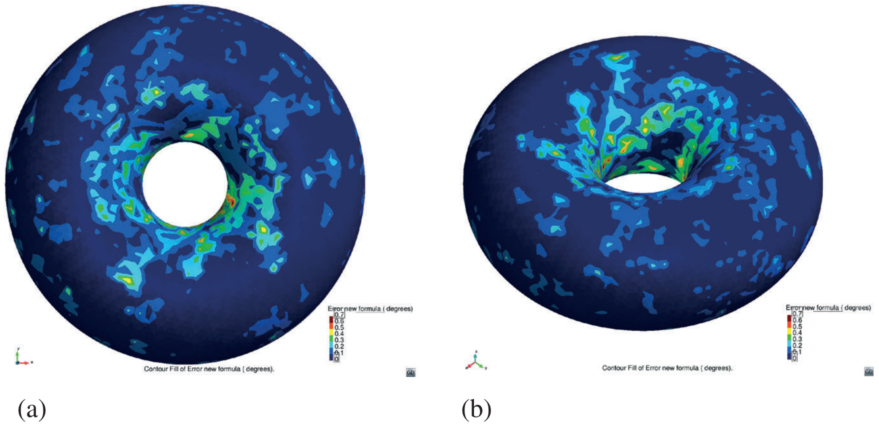

Error new formula (degrees)
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Contour Fill of Error new formula (degrees).
GiD
(a)
(b)

Figure 25. Errors made in the approximation of the vertex normals for the best weighting factor ($w_{\alpha/A_\circ}$) on a torus geometry. (a) Top view of the torus and (b) isometric view of the torus.

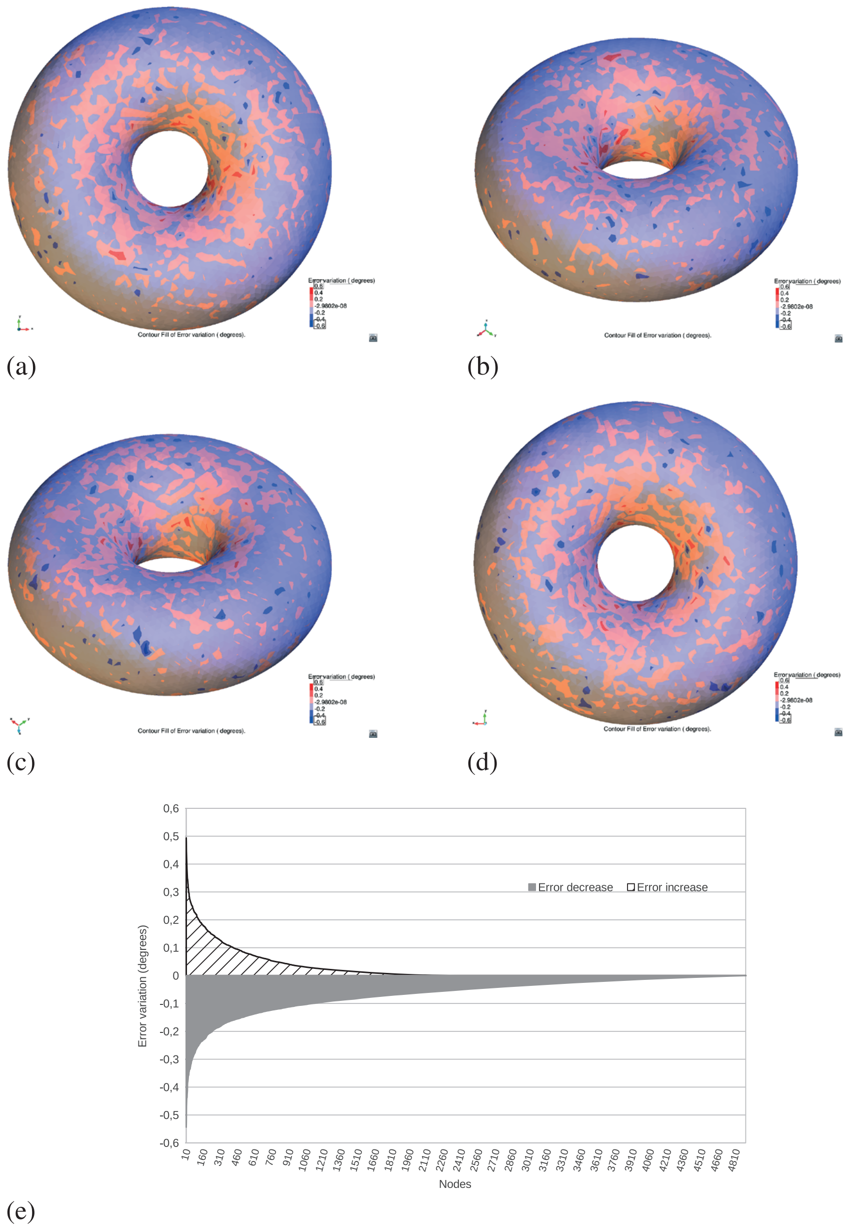
- Smooth surfaces can be approximated locally by quadrics.
- The normal direction to the element is not representative of the element's geometry but instead of the geometry resulting from the intersection of the element's plane and the local surface.
- The plane section of a quadric that approximates locally the discretized surface is, in general, a conic, but more precisely, it will almost always be a closed conic.

That is why, we propose using the geometric properties of circles when working with a mesh of triangles.

All these plots support the result obtained by Meek and Walton in [6]. Remains an open-question finding a weight-averaging method with non-uniform data which can provide normal estimates with better than first order accuracy.

With the purpose of determining quantitatively the quality of the different weighting factors, the authors present in Table I the errors made in each case for all the values of k_{max} and n_p. The factors are sorted placing the worst ranked ones at the top and the best ones at the bottom. The analysis of all these data shows the improvement of the present results with respect to the ones obtained in [10]. Comparing the best weighting factor (Equation (17)) with the one proposed by Max (exact solution for the sphere, Equation (13)), it can be observed that the error in the estimation of the vertex normals has been reduced by around 10%.

Finally, considering the unexpected result already commented on page 263 in relation to the area of the triangle and its inverse, the authors decided to try other combinations of weighting factors to check whether better results were obtained. Thus, some vertex normals have been approximated by



(a) (b)

(c) (d)

(e)

Figure 26. Different representations of the error variation obtained using the two different weights. Positive values represent an *increase* of the error when using $w_{\alpha/A_\circ}$ with respect to $w_{\sin(\alpha)/(b\cdot c)}$, while negative values represent a *decrease* of the error. (a) Top view of the torus, (b) isometric view of the torus, (c) isometric view of the torus (inverse angle), and (d) bottom view of the torus, and (e) histogram of the error variation for all the nodes in the mesh. The values have been reordered.

using weights such as the circumscribed circumference's area, its diameter, the inverse of the triangle's interior angle, and other combinations of several factors. In all cases, the results were worse than the ones already presented in this article.

5. PRACTICAL EXAMPLE

We want to introduce here a representative example of the advantage provided by the best factor with respect to the previous best factor found in the literature. A torus has been selected as the geometry of choice. The torus' geometry presents all kinds of Gaussian curvatures, which is ideal for demonstrative purposes. We have defined the torus dimensions using the golden ratio ($R/r = (1+\sqrt{5})/2$) (Figure 24). Then, we have created an irregular mesh over the torus surface in order to have random triangles for every vertex over the otherwise regular surface of the torus (Figure 24).

The normals are computed using two different weights. On one hand, we use $w_{\sin(\alpha)/(b\cdot c)}$ in Equation (13), considered to be the best existing weight in the literature. On the other hand, we use the best performing weight we have found in our study: $w_{\alpha/A_\circ}$ in Equation (17). For every node in the mesh, the error incurred by each formula is computed with respect to the actual normal vector to the torus' surface at that point. Then, we measure whether $w_{\alpha/A_\circ}$ *increases* or *decreases* the error computed when using $w_{\sin(\alpha)/(b\cdot c)}$.

Figure 25 provides a graphical representation of the error obtained when using the factor $w_{\alpha/A_\circ}$ for interpolating the normals of the torus.

Figure 26 clearly shows that there are many more cases (more than double) where the $w_{\alpha/A_\circ}$ factor reduces the error for the normal interpolation with respect to the $w_{\sin(\alpha)/(b\cdot c)}$ factor. Figure 26(a) to (d) represents the result over the torus geometry, while Figure 26(e) represents the same result as a distribution function.

6. CONCLUSIONS

A detailed analysis of the problem arising from the approximation of the normal vectors at the nodes of a triangle mesh has been presented. At the same time, a theoretical framework for the study of these approximations based on the assumption that smooth surfaces can be in turn approximated by quadrics has been proposed. The results obtained by the authors are not only of interest for the computational mechanics community but also for the computer graphics and computer-aided design communities.

A number of justified alternative weighting factors—for the approximation of normal vectors at the nodes of a triangle mesh—have been compared. From this comparison, it has been possible to propose a new formula combining the properties of different factors in order to obtain a new weighting factor for the approximation of the normal vectors at the nodes of a triangle mesh. This is the factor that produces the most precise overall results. This new weighting factor referred to in the paper as $w_{\alpha/A_\circ}$ yields more precise results than the results obtained with factors used previously in the literature.

In order to approximate the normal vectors at the nodes of a triangle mesh using a weighted average rule (see Equation (2)), the weight consisting in the interior angle of the triangle at the node considered divided by the area of the circumscribed circle to the triangle ($w_{\alpha/A_\circ}$) is recommended. The mathematical expression of this weighting factor as a function of the coordinates of the nodes of the $\widehat{ABC}$ triangle is shown in Equation (17).

Users who might be concerned by the use of trigonometric and root functions, and their computational efficiency, may consider the use of another weight presented in this paper, referred to as $w_{1/A_\circ}$, and defined in Equation (12) or the formula proposed by Max [10] (referred to as $w_{\sin(\alpha)/(b\cdot c)}$ in Equation (13)), depending on the form in which the normal vectors to each triangle are provided.

The present results can be applied to a large number of problems in computer modeling where the precise characterization of surfaces is important.

ACKNOWLEDGEMENTS

This work relates to Department of the Navy Grant issued by Office of Naval Research Global. The United States Government has a royalty-free license throughout the world in all copyrightable material contained herein; N62909-10-1-7053

REFERENCES

1. Casadei F, Potapov S. Permanent fluid-structure interaction with non-conforming interfaces in fast transient dynamics. *Computer Methods in Applied Mechanics and Engineering* 2004; **193**:4157–4194.

DOI: 10.1002/nme

2. Linhard J, Wüchner R, Bletzinger KU. Upgrading membranes to shells—the CEG rotation free shell element and its application in structural analysis. *Finite Elements in Analysis and Design* 2007; **44**(1-2):63 –74. (Available from: http://www.sciencedirect.com/science/article/pii/S0168874X07001254).
3. Ubach PA, Oñate E. New rotation-free finite element shell triangle accurately using geometrical data. *Computer Methods in Applied Mechanics and Engineering* 2009; **199**:383–391.
4. Tezduyar T, Sathe S, Pausewang J, Schwaab M, Christopher J, Crabtree J. Interface projection techniques for fluid–structure interaction modeling with moving-mesh methods. *Computational Mechanics* 2008; **43**:39–49.
5. Takizawa K, Wright S, Christopher J, Tezduyar TE. Multiscale sequentially-coupled FSI computation in parachute modeling. In *International Conference on Textile Composites and Inflatable Structures*, Oñate E, Kröplin B, Bletzinger KU (eds). CIMNE: Barcelona, Spain, 2011.
6. Meek D, Walton D. On surface normal and Gaussian curvature approximations given data sampled from a smooth surface. *Computer Aided Geometric Design* 2000; **17**:521–543.
7. OuYang D, Feng HY. On the normal vector estimation for point cloud data from smooth surfaces. *Computer-Aided Design* 2005; **37**:1071–1079.
8. Jiao X, Zha H. Consistent computation of first- and second-order differential quantities for surface meshes. In *SPM '08: Proceedings of the 2008 ACM symposium on solid and physical modeling*, Spencer SN (ed.). ACM: Suny, Stony Brook, NY, USA, 2008; 159–170.
9. Petitjean S. A survey of methods for recovering quadrics in triangle meshes. *ACM Computing Surveys* 2002; **34**(2):211–262.
10. Max N. Weights for computing vertex normals from facet normals. *Journal of graphics, GPU and game tools* 1999; **4**(2):1–6.
11. Meyer M, Desbrun M, Schröder P, Barr AH. *Discrete Differential-Geometry Operators for Triangulated 2-Manifolds*. Mathematics and Visualization, Springer-Verlag, 2003.
12. Langer T, Belyaev A, Seidel HP. Exact and interpolatory quadratures for curvature tensor estimation. *Computer Aided Geometric Design* 2007; **24**:443–463.
13. Gatzke TD, Grimm CM. Estimating curvature on triangular meshes. *International Journal of Shape Modeling* 2006; **12**(1):1–28.
14. Shirman LA, Séquin CH. Local surface interpolation with Bézier patches. *Computer Aided Geometric Design* 1987; **4**:279–295.
15. Thürmer G, Wüthrich C. Computing vertex normals from polygonal facets. *Journal of graphics, gpu and game tools* 1998; **3**(1):43–46.
16. Gouraud H. Continuous shading of curved surfaces. *IEEE Transactions on Computers* 1971; **C-20**:623–629.

DOI: 10.1002/nme