INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2001; **50**:1–27

Natural neighbour Galerkin methods

N. Sukumar*[†,‡], B. Moran[§], A. Yu Semenov[¶] and V. V. Belikov[||]

Department of Civil Engineering, 2145 Sheridan Road, Northwestern University, Evanston, IL 60208-3109, U.S.A.

SUMMARY

Natural neighbour co-ordinates (Sibson co-ordinates) is a well-known interpolation scheme for multivariate data fitting and smoothing. The numerical implementation of natural neighbour co-ordinates in a Galerkin method is known as the natural element method (NEM). In the natural element method, natural neighbour co-ordinates are used to construct the trial and test functions. Recent studies on NEM have shown that natural neighbour co-ordinates, which are based on the Voronoi tessellation of a set of nodes, are an appealing choice to construct meshless interpolants for the solution of partial differential equations. In Belikov *et al.* (*Computational Mathematics and Mathematical Physics* 1997; **37**(1):9–15), a new interpolation scheme (non-Sibsonian interpolation) based on natural neighbours was proposed. In the present paper, the non-Sibsonian interpolation scheme is reviewed and its performance in a Galerkin method for the solution of elliptic partial differential equations that arise in linear elasticity is studied. A methodology to couple finite elements to NEM is also described. Two significant advantages of the non-Sibson interpolant over the Sibson interpolant are revealed and numerically verified: the computational efficiency of the non-Sibson algorithm in 2-dimensions, which is expected to carry over to 3-dimensions, and the ability to exactly impose essential boundary conditions on the boundaries of convex and non-convex domains. Copyright © 2001 John Wiley & Sons, Ltd.

KEY WORDS: natural neighbour co-ordinates, non-Sibsonian interpolation, natural element method, meshless Galerkin methods, essential boundary conditions

1. INTRODUCTION

The natural element method (NEM) [1] is a Galerkin method for the solution of partial differential equations. In NEM, the test and trial functions are constructed using natural neighbour (Sibson) co-ordinates [2]. Natural neighbour co-ordinates are based on well-known geometric concepts such as the Voronoi diagram and the Delaunay tessellation. The Voronoi diagram and its dual Delaunay

[*] Correspondence to: N. Sukumar, Department of Civil Engineering, Northwestern University, 2145 Sheridan Road, Evanston, IL 60208-3109, U.S.A.

[†] E-mail: n-sukumar@northwestern.edu

[‡] Post-Doctoral Research Fellow, Theoretical and Applied Mechanics

[§] Associate Professor of Civil Engineering

[¶] General Physics Institute, Russian Academy of Sciences, Moscow

[||] Computing Centre, Russian Academy of Sciences, Moscow

Contract/grant sponsor: Natural Science Foundation; contract/grant number: CMS-9732319

Received 23 November 1999
Revised 6 March 2000

tessellation are the most fundamental and useful constructs to define an irregular set of nodes. Recent studies using the natural element method have demonstrated its promise for the solution of partial differential equations that arise in two-dimensional elastostatics [3–5] and elastodynamics [6]. In a recent study [7], a new interpolant (non-Sibsonian interpolant) based on natural neighbours was proposed. In this paper, we present the new implementation of NEM using the non-Sibsonian interpolant.

In NEM [4] as well as other meshless methods [8], the test and trial functions in the Galerkin implementation are constructed on the basis of a set of scattered nodes in $\mathbb{R}^d$. In these methods, the numerical integration of the weak form is carried out using a background cell/element structure. In several meshless methods, moving least squares approximants [9] are used to construct the trial and test spaces. The properties of interpolation of nodal data, ease of imposing essential boundary conditions, and neighbour relationships that are based on the local distribution and density of nodes at a given point are some of the most important advantages of natural neighbour interpolants over moving least squares approximants.

In Belikov *et al.* [7], the non-uniqueness of interpolation schemes based on natural neighbours was shown, and the non-Sibsonian interpolant was proposed. The local harmonic property and construction of higher-order interpolation schemes using the non-Sibsonian interpolant were introduced in Belikov and Semenov [10]. As opposed to the Sibson interpolant which is based on the area (volume) of overlap of first-order Voronoi polygons (polyhedra) in $\mathbb{R}^2$ ($\mathbb{R}^3$), the non-Sibsonian interpolant requires computation of Lebesgue measures of order $d-1$ in $\mathbb{R}^d$. Since most of the properties are common to both schemes, the better computational performance and ease of implementation of the non-Sibsonian interpolation method over the Sibson interpolant renders the non-Sibsonian interpolant an interesting and computationally attractive choice for the numerical solution of partial differential equations.

The outline of this papers is as follows. In the following section, the notion of natural neighbours is described and the Sibson and non-Sibsonian interpolation methods are introduced. The computational algorithm used in the two methods is also presented. In Section 3, issues pertaining to the imposition of essential boundary conditions using the two interpolants are discussed, and a methodology to couple the natural element method to finite elements is described in Section 5. In Section 6, the governing equations of elastostatics together with the Galerkin formulation for NEM are described. Numerical experiment results in two-dimensional elasticity are presented in Section 7. We close in Section 8 with some concluding remarks on the performance and accuracy of the natural element method.

2. NATURAL NEIGHBOURS

The notions of natural neighbours and natural neighbour interpolation were introduced by Sibson [2] as a means for data fitting and smoothing. The Voronoi diagram and its dual Delaunay triangulation, which are used in the construction of the natural neighbour interpolant, are useful geometric constructs that define an irregular set of points (nodes). For simplicity of exposition, we consider two-dimensional Euclidean space $\mathbb{R}^2$; the theory, however, is applicable in a general d-dimensional framework. Given a distribution of points (nodes) in the plane, the Delaunay simplices partition the convex hull Ω of the points into regions Ω_i such that $\Omega = \bigcup_{i=1}^{t} \Omega_i$. In Delaunay interpolation (constant strain finite elements), a linear interpolant is constructed over each Delaunay triangle. The Delaunay triangulation of a set of nodes is non-unique, and hence the interpolation is sensitive

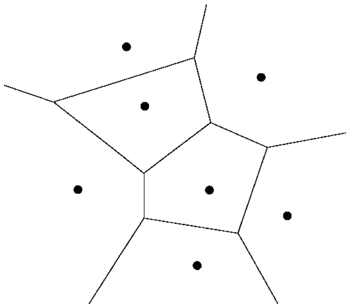

Figure 1. Voronoi diagram $V(N)$ for a set $\mathbf{N}$ of seven nodes.

to geometric perturbations of the position of the nodes. As opposed to the Delaunay triangulation, its dual the Voronoi diagram is unique. We now formally introduce the Voronoi diagram and thereafter the definition of natural neighbours.

Consider a set of distinct nodes $\mathbf{N}=\{n_1,n_2,\dots,n_M\}$ in $\mathbb{R}^2$. The Voronoi diagram (or 1st-order Voronoi diagram) of the set $\mathbf{N}$ is a subdivision of the plane into regions T_I (closed and convex, or unbounded), where each region T_I is associated with a node n_I, such that any point in T_I is closer to n_I (nearest neighbour) than to any other node $n_J \in N$ $(J \neq I)$—T_I is the locus of points closer to n_I than to any other node. The regions T_I are the Voronoi cells of n_I. In mathematical terms, the Voronoi polygon T_I is defined as [11]

$$T_I=\{\mathbf{x}\in\mathbb{R}^2:\ d(\mathbf{x},\mathbf{x}_I)<d(\mathbf{x},\mathbf{x}_J)\ \forall J\neq I\} \tag{1}$$

where $d(\mathbf{x}_I,\mathbf{x}_J)$, the Euclidean metric, is the distance between $\mathbf{x}_I$ and $\mathbf{x}_J$. The Voronoi diagram for a set $\mathbf{N}$ consisting of seven nodes is shown in Figure 1. The concept of nearest neighbours and neighbouring nodes is embedded in the first-order Voronoi diagram. By a similar extension, one can construct higher order (k-order, $k>1$) Voronoi diagrams in the plane. Of particular interest in the present context is the case $k=2$, which is the second-order Voronoi diagram. The second-order Voronoi diagram of the set of nodes $\mathbf{N}$ is a subdivision of the plane into cells T_{IJ}, where each region T_{IJ} is associated with a nodal-neighbour-pair (n_I,n_J) (k-tuple for the k-order Voronoi diagram), such that T_{IJ} is the locus of all points that have n_I as the nearest neighbour, and n_J as the second nearest neighbour. It is emphasized that the cell T_{IJ} is non-empty if and only if n_I and n_J are neighbours. The second-order Voronoi cell T_{IJ} $(I\neq J)$ is defined as [2].

$$T_{IJ}=\{\mathbf{x}\in\mathbb{R}^2:\ d(\mathbf{x},\mathbf{x}_I)<d(\mathbf{x},\mathbf{x}_J)<d(\mathbf{x},\mathbf{x}_K)\ \forall K\neq I,J\} \tag{2}$$

In order to quantify the neighbour relation for any point $\mathbf{x}$ introduced into the tessellation, Sibson [2] used the concept of second-order Voronoi cells, and thereby introduced natural neighbours and natural neighbour co-ordinates. The notion of neighbouring nodes is broadened and generalized by the definition of natural neighbours. In Figure 2(a), a point $\mathbf{x}$ is introduced into the Voronoi diagram of the set $\mathbf{N}$ shown in Figure 1. If $\mathbf{x}$ is considered as a node along with the set of nodes $\mathbf{N}$, then the natural neighbours of $\mathbf{x}$ are those nodes which form an edge of a triangle with $\mathbf{x}$ in the new tesselation (triangulation). A straight-forward means to arrive at the same end is to use the *empty circumcircle criterion* [12]—if $\mathrm{DT}(n_J,n_K,n_L)$ is any Delaunay triangle of the nodal set $\mathbf{N}$, then the circumcircle of DT contains no other nodes of $\mathbf{N}$. By doing so, we arrive at the result

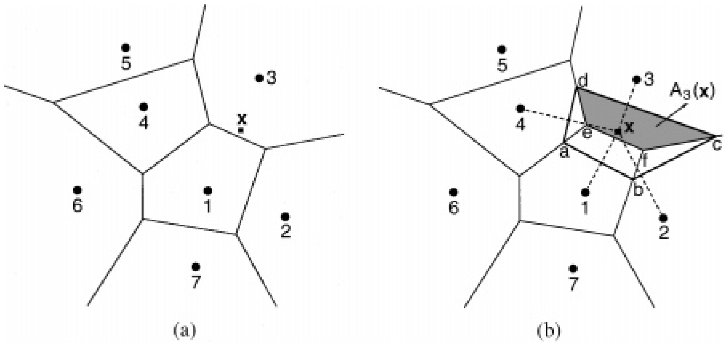

Figure 2. Construction of natural neighbours: (a) Original Voronoi diagram and **x**; and (b) 1st-order and 2nd-order Voronoi cells about **x**.

that if **x** lies within the circumcircle of triangle DT(n_I, n_J, n_K), then n_I, n_J, and n_K are natural neighbours of **x**. In Figure 2(b), the perpendicular bisectors from point **x** to its natural neighbours are constructed and the Voronoi cell $T_{\mathbf{x}}$ (closed polygon *abcd*) is obtained. It is observed that **x** has four ($n = 4$) natural neighbours, namely nodes 1, 2, 3, and 4.

2.1. Sibson interpolation

Natural neighbour co-ordinates are used as the interpolating functions in natural neighbour (Sibson) interpolation. We refer to Figure 2 in order to define the natural neighbour co-ordinates for a point **x** in the plane. Let $\kappa(\mathbf{x})$ be a Lebesgue measure (length, area, or volume in 1D, 2D, or 3D, respectively) of $T_{\mathbf{x}}$, and $\kappa_I(\mathbf{x})$ ($I = 1$–4) be that of $T_{\mathbf{x}I}$. In two-dimensions, the measures are areas, and hence we denote $A(\mathbf{x}) \equiv \kappa(\mathbf{x})$ and $A_I(\mathbf{x}) \equiv \kappa_I(\mathbf{x})$. The natural neighbour co-ordinates of **x** with respect to a natural neighbour I is defined as the ratio of the area of overlap of the Voronoi cells T_I and $T_{\mathbf{x}}$ to the total area of the Voronoi cell of **x**:

$$\phi_I(\mathbf{x}) = \frac{A_I(\mathbf{x})}{A(\mathbf{x})} \tag{3}$$

where I ranges from 1 to n, and $A(\mathbf{x}) = \sum_{J=1}^{n} A_J(\mathbf{x})$. The four regions shown in Figure 2(b) are the second-order cells, whereas their union (closed polygon *abcd*) is a first-order Voronoi cell. Referring to Figure 2, the shape function $\phi_3(\mathbf{x})$ is given by

$$\phi_3(\mathbf{x}) = \frac{A_3(\mathbf{x})}{A(\mathbf{x})} \tag{4}$$

If the point **x** coincides with a node ($\mathbf{x} = \mathbf{x}_I$), $\phi_I(\mathbf{x}) = 1$, and all other shape functions are zero. The properties of positivity, interpolation, and partition of unity directly follow:

$$0 \leqslant \phi_I(\mathbf{x}) \leqslant 1, \quad \phi_I(\mathbf{x}_J) = \delta_{IJ}, \quad \sum_{I=1}^{n} \phi_I(\mathbf{x}) = 1 \text{ in } \Omega \tag{5}$$

Natural neighbour shape functions also satisfy the local co-ordinate property [2], namely

$$\mathbf{x} = \sum_{I=1}^{n} \phi_I(\mathbf{x}) \mathbf{x}_I \tag{6}$$

which, in conjunction with Equation (5) imply that the natural neighbour interpolant spans the space of linear polynomials (linear completeness).

Natural neighbour interpolation has primarily been used in the area of data interpolation and modelling of geophysical phenomena [13–15]. The support of the shape function $\phi_I(\mathbf{x})$ is the intersection of the convex hull with the union of all Delaunay circumcircles that pass through node I [16]. Natural neighbour shape functions are C^∞ everywhere, except at the nodes where they are C^0 [2, 16]. Farin [16] proposed a C^1 natural neighbour interpolant based on Bernstein–Bézier simplices, and Sukumar and Moran [5] developed a computational methodology for its application to fourth-order elliptic PDEs. In one-dimension, natural neighbour interpolation is identical to linear finite elements [4]; in the particular case of three natural neighbours, n-n interpolation is precisely barycentric co-ordinates; and for four natural neighbours at the vertices of a rectangle, bilinear interpolation is realized [16]. A detailed description and discussion of the above properties of natural neighbour interpolants can be found in Sukumar *et al.* [4].

2.2. *Non-Sibsonian interpolation*

We give the following rigorous definition of the non-Sibsonian interpolant [7]. Let $\mathbf{N} = \{n_1, n_2, \dots, n_M\}$ be a set of distinct nodes in $\mathbb{R}^d$. We denote the Voronoi cell of node $\mathbf{x}_I$ by T_I: $T_I = \{\mathbf{x} \in \mathbb{R}^d\colon\ d(\mathbf{x}, \mathbf{x}_I) < d(\mathbf{x}, \mathbf{x}_J),\ J \neq I\}$. In addition, define $t_{IJ} = \{\mathbf{x} \in \bar{T}_I \cap \bar{T}_J,\ J \neq I\}$, where $d(\cdot\,,\cdot)$ is the Euclidean metric, $\bar{T}_I = T_I \cup \partial T_I$ is the closure of set T_I, and t_{IJ} may be an empty set. If $d(\mathbf{x}_I, \mathbf{x}_J) \neq 0$, then

$$\sum_{\substack{J \\ J \neq I}} \frac{|t_{IJ}|\mathbf{x}_J}{d(\mathbf{x}_I, \mathbf{x}_J)} = \mathbf{x}_I \left[\sum_{\substack{J \\ J \neq I}} \frac{|t_{IJ}|}{d(\mathbf{x}_I, \mathbf{x}_J)} \right] \tag{7}$$

where $|\cdot|$ denotes the Lebesgue measure in $\mathbb{R}^{d-1}$. In terms of the notation used above, the non-Sibsonian shape function $\phi_I(\mathbf{x})$ is defined as

$$\phi_I(\mathbf{x}) = \frac{\dfrac{|t_{\mathbf{x}I}|}{d(\mathbf{x}, \mathbf{x}_I)}}{\sum_{J=1}^{n} \dfrac{|t_{\mathbf{x}J}|}{d(\mathbf{x}, \mathbf{x}_J)}} \tag{8}$$

Consider the Voronoi diagram and a point $\mathbf{x}$ in the plane as shown in Figure 2(b). The point $\mathbf{x}$ has four natural neighbours, and in Figure 3 the Voronoi cell of the point $\mathbf{x}$ and its neighbours are illustrated. The distance $s_I(\mathbf{x})$ is the Lebesgue measure (length in $\mathbb{R}^2$) of the Voronoi edge associated with node I, and $h_I(\mathbf{x})$ is the perpendicular distance between the Voronoi edge of node I to the point $\mathbf{x}$. The non-Sibsonian (nS) shape function $\phi_I(\mathbf{x})$ is defined by [7]

$$\phi_I(\mathbf{x}) = \frac{\alpha_I(\mathbf{x})}{\sum_{J=1}^{n} \alpha_J(\mathbf{x})}, \quad \alpha_J(\mathbf{x}) = \frac{s_J(\mathbf{x})}{h_J(\mathbf{x})} \tag{9}$$

It is noted that in $\mathbb{R}^2$, the computational complexity of nS shape function depends only on the ratio of a Lebesgue measure of $\mathbb{R}$ divided by a linear dimension. In a general d-dimensional setting, the dependence is on the ratio of a Lebesgue measure of $\mathbb{R}^{d-1}$ divided by a linear dimension. An immediate comparison with the natural neighbour shape function reveals that the computational effort in $\mathbb{R}^d$ for the natural neighbour interpolant is co-dimensional (d-dimensional volumes), whereas for the non-Sibsonian interpolant it is one order less (($n-1$)-dimensional volumes).

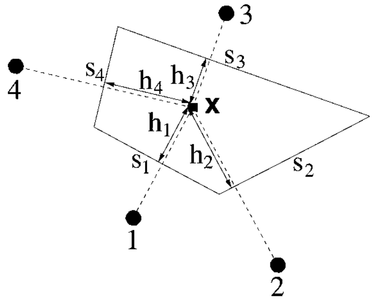

Figure 3. Non-Sibsonian interpolation.

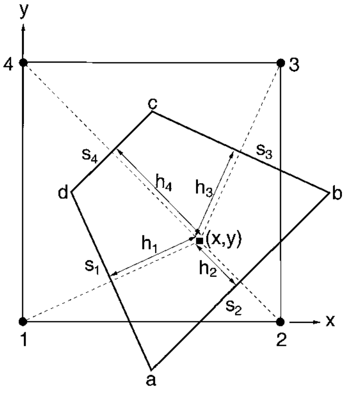

Figure 4. Bilinear interpolation on a regular grid ($n = 4$).

Consider an interpolation scheme for a vector-valued function $\mathbf{u}(\mathbf{x}) : \Omega \rightarrow \mathbb{R}^2$, in the form:

$$\mathbf{u}^h(\mathbf{x}) = \sum_{I=1}^{n} \phi_I(\mathbf{x})\mathbf{u}_I \tag{10}$$

where $\mathbf{u}_I$ $(I = 1, 2, \ldots, n)$ are the vectors of nodal displacements at the n natural neighbours, and $\phi_I(\mathbf{x})$ are the Sibson (see Equation (3)) or the non-Sibsonian shape functions defined in Equation (9). In the natural element method, the trial and test functions are constructed using the approximation indicated in Equation (10).

2.3. *Properties*

The non-Sibsonian interpolant is based on the notion of natural neighbours, and hence most of the properties of the natural neighbour interpolant are also shared by the non-Sibsonian interpolant. Properties such as partition of unity, positivity, interpolation, and support and regularity of the shape functions are common to both interpolants. In addition, due to the above properties, both interpolants ensure the boundedness of the norm of the interpolated result: $\|u\| \leqslant \max u_I$. Interpolants that are linear combinations of the Sibson and non-Sibsonian interpolants are also valid natural neighbour-based interpolants that can be used for the solution of PDEs. We address the linear completeness of the non-Sibsonian interpolant and discuss univariate, bivariate, and higher-order interpolation within the context of non-Sibsonian interpolation.

2.3.1. Linear completeness. The non-Sibsonian interpolant has linear precision, i.e. it spans the space of linear polynomials in $\mathbb{R}^d$ [7]. In the finite element literature, the ability of the interpolant to reproduce constant and linear displacement fields is known as linear completeness—a necessary condition for convergence for second-order elliptic PDEs such as Poisson's equation and elastostatics. The proof that the non-Sibsonian interpolation has linear completeness follows:

Proof. Consider a linear displacement field in the form:

$$\mathbf{u}(\mathbf{x}) = \boldsymbol{\alpha} + \boldsymbol{\beta}^{\mathrm{T}}\mathbf{x} \tag{11}$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are constant vectors. The exact nodal displacements are given by

$$\mathbf{u}_I = \boldsymbol{\alpha} + \boldsymbol{\beta}^{\mathrm{T}} \mathbf{x}_I \tag{12}$$

where I is the index for any particular node. Consider the NEM trial function, namely

$$\mathbf{u}^h(\mathbf{x}) = \sum_{I=1}^{n} \phi_I(\mathbf{x}) \mathbf{u}_I \tag{13}$$

where $\mathbf{u}_I$ is the vector of nodal displacements for node I. On using Equation (12) in the above equation, we obtain

$$\mathbf{u}^h(\mathbf{x}) = \boldsymbol{\alpha} \sum_{I=1}^{n} \phi_I(\mathbf{x}) + \boldsymbol{\beta}^{\mathrm{T}} \sum_{I=1}^{n} \phi_I(\mathbf{x}) \mathbf{x}_I \tag{14}$$

By adding and subtracting $\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}$ to the above equation and noting that $\sum_I \phi_I(\mathbf{x}) = 1$, we have

$$\mathbf{u}^h(\mathbf{x}) = \boldsymbol{\alpha} + \boldsymbol{\beta}^{\mathrm{T}} \mathbf{x} + \boldsymbol{\beta}^{\mathrm{T}} \sum_{I=1}^{n} \phi_I(\mathbf{x})(\mathbf{x}_I - \mathbf{x}) \tag{15}$$

and hence to complete the proof it suffices to show that the following equality holds:

$$\sum_{I=1}^{n} \phi_I(\mathbf{x})(\mathbf{x}_I - \mathbf{x}) = \mathbf{0} \tag{16}$$

We first show that the above equality holds in the planar (two-dimensional) case. To this end, transforming from (x, y) in $\mathbb{R}^2$ to $z = x + \mathrm{i}y$ in the complex plane $\mathbb{C}$, the above criterion can be re-written as

$$\sum_{I=1}^{n} \phi_I(z)(z_I - z) = 0 \tag{17}$$

which on substituting the explicit form of the non-Sibsonian shape function from Equation (9) becomes

$$\frac{\sum_{I=1}^{n} (z_I - z) \dfrac{s_I(z)}{h_I(z)}}{\sum_{J=1}^{n} \dfrac{s_J(z)}{h_J(z)}} = 0 \tag{18}$$

To prove the above, we note that the sides of the Dirichlet cell that contain (x, y) are the complex numbers $\tilde{z}_I$. Since the polygon has a closed contour, the following identity holds:

$$\sum_{I=1}^{n} \tilde{z}_I = 0 \tag{19}$$

Introducing the trigonometric form $\tilde{z}_I = s_I(z)\exp(\mathrm{i}\varphi_I(z))$, we rearrange the above equation as follows:

$$\sum_{I=1}^{n} \tilde{z}_I = \sum_{I=1}^{n} s_I(z)\exp(\mathrm{i}\varphi_I(z)) = \mathrm{i}\sum_{I=1}^{n} s_I(z)\exp(\mathrm{i}\varphi_I(z) - \mathrm{i}\pi/2) = 0 \tag{20}$$

and since $|z_I - z| = 2h_I(z)$, we can write

$$\mathrm{i}\sum_{I=1}^{n} [h_I(z)\exp(\mathrm{i}\varphi_I(z) - \mathrm{i}\pi/2)]\frac{s_I(z)}{h_I(z)} = \frac{\mathrm{i}}{2}\sum_{I=1}^{n} (z_I - z)\frac{s_I(z)}{h_I(z)} = 0 \tag{21}$$

which proves Equation (18) and hence

$$\mathbf{u}^h(\mathbf{x}) = \boldsymbol{\alpha} + \boldsymbol{\beta}^{\mathrm{T}}\mathbf{x} = \mathbf{u}(\mathbf{x}) \tag{22}$$

which completes the proof for the planar case. □

Now, we prove the same for the case of arbitrary $\mathbb{R}^d$. By analogy we consider a linear displacement field $\mathbf{u}(\mathbf{x})$ in $\mathbb{R}^d$ and show that Equation (16) holds for it identically. Following the 2-dimensional case, the criterion to be proved for the d-dimensional case has the following vector form:

$$\frac{\sum_{I=1}^{n} (\mathbf{x}_I - \mathbf{x})\dfrac{s_I(\mathbf{x})}{h_I(\mathbf{x})}}{\sum_{J=1}^{n} \dfrac{s_J(\mathbf{x})}{h_J(\mathbf{x})}} = \mathbf{0} \tag{23}$$

To prove the above, consider the closed surface S of the Voronoi polyhedron that encloses $\mathbf{x} \in \mathbb{R}^d$ and has the volume V. Now, consider the vector identity (Green's theorem):

$$\int_V \nabla f \,\mathrm{d}V = \oint_S f \,\mathrm{d}\mathbf{S} \tag{24}$$

On substituting $f = 1$ in the above equation, we obtain

$$\int_V \nabla f \,\mathrm{d}V = \mathbf{0} = \oint_S \mathrm{d}\mathbf{S} = \sum_{I=1}^{n} \frac{(\mathbf{x}_I - \mathbf{x})s_I(\mathbf{x})}{|\mathbf{x}_I - \mathbf{x}|} \tag{25}$$

and since $|\mathbf{x}_I - \mathbf{x}| = 2h_I(\mathbf{x})$, we have

$$\sum_{I=1}^{n} \frac{(\mathbf{x}_I - \mathbf{x})s_I(\mathbf{x})}{2h_I(\mathbf{x})} = \mathbf{0} \tag{26}$$

which proves Equation (23) and hence

$$\mathbf{u}^h(\mathbf{x}) = \boldsymbol{\alpha} + \boldsymbol{\beta}^{\mathrm{T}}\mathbf{x} = \mathbf{u}(\mathbf{x}) \tag{27}$$

which completes the proof for the general d-dimensional case. □

2.3.2. Univariate interpolation. In 1-dimension, the non-Sibsonian shape function given in Equation (8) is undefined since the Lebesgue measure of a point is zero. However, by considering univariate interpolation as a limiting case of bivariate interpolation, it is readily seen that linear interpolation is realized in 1-dimension (see Section 3.1 too). Hence the equivalence with one-dimensional linear finite elements is obtained, as was the case for the Sibson interpolant [4].

2.3.3. Bivariate interpolation. Case I: $n=3$ *or* $n=4$. If a point $\mathbf{x}$ has three neighbours, then by a unique argument or by considering the linear reproducing conditions given in Section 2.3.1, barycentric co-ordinates are realized by nS interpolation. The proof is similar to that outlined in Sukumar *et al.* [4] for the Sibson interpolant. For the case of four natural neighbours ($n=4$) at the vertices of a rectangle, bilinear interpolation on the rectangle is obtained. The proof follows:

Proof. Consider a point $\mathbf{x}$ with four natural neighbours located at the vertices of a unit square: $(x_1, y_1)=(0,0)$, $(x_2, y_2)=(1,0)$, $(x_3, y_3)=(1,1)$, and $(x_4, y_4)=(0,1)$ (Figure 4). The first-order (dark line) Voronoi cell of point $\mathbf{x}$ is shown in Figure 4. By definition of the non-Sibsonian shape functions, we can write

$$\phi_I(\mathbf{x})=\frac{\dfrac{s_I(\mathbf{x})}{h_I(\mathbf{x})}}{\sum_{J=1}^{4}\dfrac{s_J(\mathbf{x})}{h_J(\mathbf{x})}} \quad (I=1\text{–}4) \tag{28}$$

where $s_I(\mathbf{x})$ and $h_I(\mathbf{x})$ are indicated in Figure 4.

By recalling the definition of 1st-order Voronoi cells, it is clearly seen that vertex a is the centre of the circle that circumscribes the triangle $\triangle_{12\mathbf{x}}$, and proceeding likewise, b is the circumcentre of triangle $\triangle_{23\mathbf{x}}$, c that of $\triangle_{34\mathbf{x}}$, and d that of $\triangle_{41\mathbf{x}}$. These co-ordinates are computed to be

$$a_1=\frac{1}{2}, \qquad a_2=\frac{-x+x^2+y^2}{2y} \tag{29a}$$

$$b_1=\frac{1+y-x^2-y^2}{2(1-x)}, \quad b_2=\frac{1}{2} \tag{29b}$$

$$c_1=\frac{1}{2}, \qquad c_2=\frac{1+x-x^2-y^2}{2(1-y)} \tag{29c}$$

$$d_1=\frac{-y+x^2+y^2}{2x}, \qquad d_2=\frac{1}{2} \tag{29d}$$

We note that $s_1(\mathbf{x})=|\vec{ab}|$, $s_2(\mathbf{x})=|\vec{bc}|$, $s_3(\mathbf{x})=|\vec{cd}|$, $s_4(\mathbf{x})=|\vec{da}|$, and $2h_I=d(\mathbf{x},\mathbf{x}_I)$. By carrying out simple algebraic calculations, we obtain

$$\frac{s_1(\mathbf{x})}{h_1(\mathbf{x})}=\frac{x+y-x^2-y^2}{xy} \tag{30a}$$

$$\frac{s_2(\mathbf{x})}{h_2(\mathbf{x})}=\frac{x+y-x^2-y^2}{y(1-x)} \tag{30b}$$

$$\frac{s_3(\mathbf{x})}{h_3(\mathbf{x})}=\frac{x+y-x^2-y^2}{(1-x)(1-y)} \tag{30c}$$

$$\frac{s_4(\mathbf{x})}{h_4(\mathbf{x})}=\frac{x+y-x^2-y^2}{x(1-y)} \tag{30d}$$

and hence the non-Sibsonian shape functions are

$$\phi_1(\mathbf{x}) = (1 - x)(1 - y) \tag{31a}$$

$$\phi_2(\mathbf{x}) = x(1 - y) \tag{31b}$$

$$\phi_3(\mathbf{x}) = xy \tag{31c}$$

$$\phi_4(\mathbf{x}) = y(1 - x) \tag{31d}$$

which are bilinear finite element shape functions. The above derivation is easily generalized to the rectangle (linear transformation of a square), and hence bilinear interpolation on the rectangle is realized by the non-Sibsonian interpolant. □

Case II: *Arbitrary n.* We consider the computation of non-Sibsonian shape functions for the case in which the point $\mathbf{x} = (x, y)$ has an arbitrary number of natural neighbours. Let $\mathbf{x}_m = (x_m, y_m)$ and $\mathbf{x}_n = (x_n, y_n)$ be two adjacent natural neighbours for the point $\mathbf{x}$, where $n = m + 1$ or $n = m - 1$ (Figure 5). We assume clockwise orientation to be the positive sense. In Figure 5(a), line AB is perpendicular to $\mathbf{x}_m - \mathbf{x}$ and line CB is perpendicular to $\mathbf{x}_n - \mathbf{x}$. Any point along the line AB has the form:

$$\overrightarrow{AB} : \tfrac{1}{2}(\mathbf{x}_m + \mathbf{x}) + t_m(\mathbf{x}_m - \mathbf{x})_\perp \tag{32}$$

where t_m is a scalar parameter. In addition,

$$\mathbf{x}_m + \mathbf{x} = (x_m + x, y_m + y) \tag{33a}$$

$$(\mathbf{x}_m - \mathbf{x})_\perp = (x_m - x, y_m - y)_\perp = (y_m - y, -x_m + x) \tag{33b}$$

where $\mathbf{q}_\perp = \mathbf{q} \wedge \mathbf{e}_3$. Now, any point along CB has the form:

$$\overrightarrow{BC} : \tfrac{1}{2}(\mathbf{x}_n + \mathbf{x}) - t_n(\mathbf{x}_n - \mathbf{x})_\perp \tag{34}$$

where t_n is a scalar parameter. In addition,

$$\mathbf{x}_n + \mathbf{x} = (x_n + x, y_n + y) \tag{35a}$$

$$(\mathbf{x}_n - \mathbf{x})_\perp = (x_n - x, y_n - y)_\perp = (y_n - y, -x_n + x) \tag{35b}$$

Now, the condition for line intersection at point B can be stated as

$$\tfrac{1}{2}(\mathbf{x}_m + \mathbf{x}) + t_m(\mathbf{x}_m - \mathbf{x})_\perp = \tfrac{1}{2}(\mathbf{x}_n + \mathbf{x}) - t_n(\mathbf{x}_n - \mathbf{x})_\perp \tag{36}$$

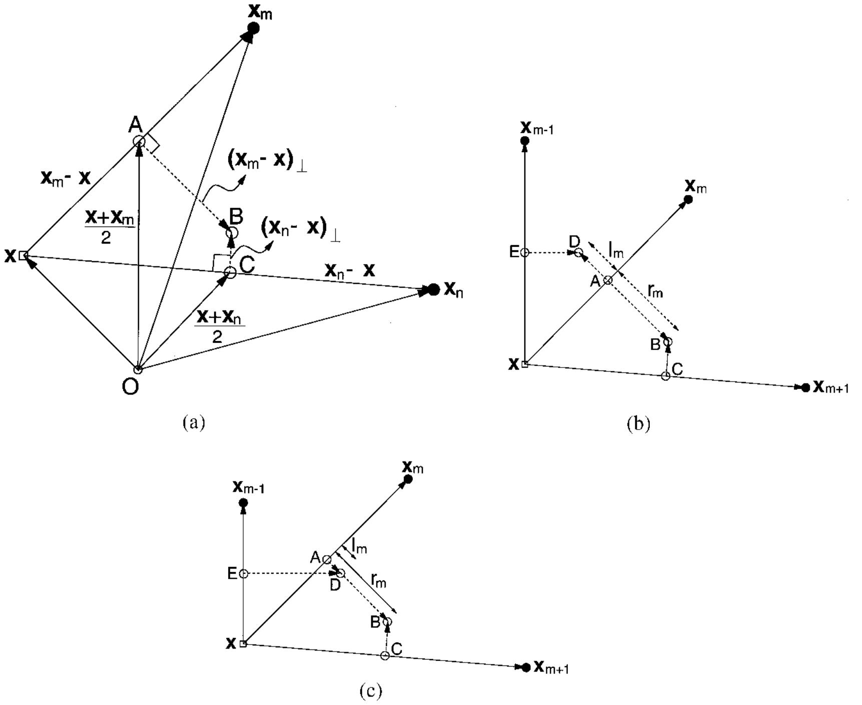

Figure 5. Computation of non-Sibsonian interpolant in $\mathbb{R}^2$: (a) Geometric construction; (b) $r_m > 0$ and $l_m < 0$; and (c) $r_m > 0$ and $l_m > 0$.

and hence we obtain the following system of equations for t_m and t_n:

$$-(y_m - y)t_m - (y_n - y)t_n = \frac{x_m - x_n}{2} \tag{37a}$$

$$(x_m - x)t_m + (x_n - x)t_n = \frac{y_m - y_n}{2} \tag{37b}$$

Solving the above system, we obtain

$$t_m = \frac{1}{2} \frac{(x_m - x_n)(x_n - x) + (y_m - y_n)(y_n - y)}{(x_m - x)(y_n - y) - (x_n - x)(y_m - y)} \tag{38a}$$

$$t_n = -\frac{1}{2} \frac{(x_m - x_n)(x_m - x) + (y_m - y_n)(y_m - y)}{(x_m - x)(y_n - y) - (x_n - x)(y_m - y)} \tag{38b}$$

Now, we note that

$$\frac{|\overrightarrow{AB}|}{\frac{1}{2}|\mathbf{x}_m - \mathbf{x}|} = 2|t_m| \tag{39}$$

which can be readily inferred from Figure 5(a). By taking into account $n = m + 1$ and $n = m - 1$ (see the cases shown in Figures 5(b) and 5(c)), we can compute s_m/h_m as

$$s_m/h_m = |r_m - l_m| \tag{40a}$$

where

$$r_m = \frac{(x_m - x_{m+1})(x_{m+1} - x) + (y_m - y_{m+1})(y_{m+1} - y)}{(x_m - x)(y_{m+1} - y) - (x_{m+1} - x)(y_m - y)} \tag{40b}$$

$$l_m = \frac{(x_m - x_{m-1})(x_{m-1} - x) + (y_m - y_{m-1})(y_{m-1} - y)}{(x_m - x)(y_{m-1} - y) - (x_{m-1} - x)(y_m - y)} \tag{40c}$$

On using the above relations for all the natural neighbours, the non-Sibsonian shape functions are evaluated using Equation (9).

2.3.4. Higher-order interpolation. Belikov and Semenov [17] proposed a compact procedure for the generation of higher-order interpolation using non-Sibsonian interpolants. For kth-order interpolation at the point $\mathbf{x}$, it is sufficient that $u(\mathbf{x})$ satisfies the equation $\triangle^k u = 0$, where $\triangle$ is the Laplacian operator. Then, for the neighbours of $\mathbf{x}$, a sequence of values $w_1, w_2, \ldots, w_{m-1}$ are computed where the w_i satisfy the relations $\triangle u = w_1$, $\triangle w_1 = w_2$, $\triangle w_2 = w_3, \ldots, \triangle w_{k-1} = w_k = 0$. The computation of $u(\mathbf{x})$ is performed by going through the above sequence of equations in reverse order.

Let us first consider an interpretation of non-Sibsonian interpolation. Using Equation (10), we can write the non-Sibsonian trial function $u^h(\mathbf{x})$ for a scalar-valued function $u(\mathbf{x})$ in the form:

$$\sum_{I=1}^{n} \frac{u_I - u^h(\mathbf{x})}{2h_I(\mathbf{x})} s_I(\mathbf{x}) = 0 \tag{41}$$

Thus, the continuous analogue of the above discrete form is

$$\oint_S \frac{\partial u}{\partial n}\, \mathrm{d}S = \int_V \triangle u \, \mathrm{d}V = 0 \tag{42}$$

or

$$\triangle u = 0, \quad \triangle = \frac{\partial^2}{\partial x_1^2} + \cdots + \frac{\partial^2}{\partial x_d^2} \tag{43}$$

where Gauss's (divergence) theorem has been used to convert the surface integral into a volume integral. Thus, Equation (10) provides an approximate local discrete solution of the harmonic equation in $\mathbb{R}^d$. We can refer to $\phi_I(\mathbf{x})$ as "harmonic" co-ordinates [17].

Now, for the specific case of $k = 2$ (second-order interpolation) we obtain two equations: $\triangle u = w$, $\triangle w = 0$, and hence the final formulas for the calculation of the second-order interpolant have the following form:

$$w_I = \frac{1}{V_I} \sum_{J=1}^{p_I} \frac{u_J - u_I}{h_J^I} s_J^I \tag{44a}$$

$$w^h(\mathbf{x}) = \sum_{I=1}^{n} \phi_I(\mathbf{x}) w_I \tag{44b}$$

$$u^h(\mathbf{x}) = \sum_{I=1}^{n} \phi_I(\mathbf{x}) u_I - \frac{w^h(\mathbf{x}) V(\mathbf{x})}{\sum_{J=1}^{n} \dfrac{s_J(\mathbf{x})}{h_J(\mathbf{x})}} \tag{44c}$$

Here, n is the number of natural neighbours for point $\mathbf{x}$; p_I is the number of natural neighbours for point $\mathbf{x}_I$, and V_I is the area (volume) of the corresponding Dirichlet cell. In determining the natural neighbours for point $\mathbf{x}_I$, one considers the nodal set $\mathbf{N} - n_I$ (see Section 2). Therefore, the natural neighbours p_I for point $\mathbf{x}_I$ are easily seen to be those nodes which are connected to node $\mathbf{x}_I$ in the Delaunay triangulation of the set $\mathbf{N}$. First the parameters w_I are evaluated using Equation (44a), and then $w^h(\mathbf{x})$ is obtained from Equation (44b) which is used in Equation (44c) to compute $u^h(\mathbf{x})$ at $\mathbf{x}$.

For the derivation of Equation (44), the following equalities in three different forms are used:

$$\text{Continuous form:} \quad \triangle u = w \tag{45a}$$

$$\text{Integral form:} \quad \oint_S \frac{\partial u}{\partial n}\, \mathrm{d}S = \int_V w \, \mathrm{d}V \tag{45b}$$

$$\text{Discrete form:} \quad \sum_{J=1}^{p} \frac{u_J - u_K}{h_J^K} s_J^K = w_K V_K \tag{45c}$$

where p is the number of natural neighbour for the point $\mathbf{x}_K$. Comparing the above to Equation (44), we note that if $\mathbf{x}_K = \mathbf{x}$, then $p = n$, $V_K = V(\mathbf{x})$, and $s_J^K = s_J(\mathbf{x})$ and $h_J^K = h_J(\mathbf{x})$; if $\mathbf{x}_K = \mathbf{x}_I$, then $p = p_I$.

We now proceed to re-cast the second-order non-Sibsonian interpolant in the standard trial function form:

$$u^h(\mathbf{x}) = \sum_{I=1}^{m} \psi_I(\mathbf{x}) u_I \tag{46}$$

where $\psi_I(\mathbf{x})$ is the second-order non-Sibsonian shape function for node I at $\mathbf{x}$ and $\mathbf{u}_I$ $(I = 1, 2, \ldots, m)$ are the vectors of nodal displacements. Using Equation (44), we can immediately write:

$$u^h(\mathbf{x}) = \sum_{I=1}^{n} \phi_I(\mathbf{x})[u_I - \beta(\mathbf{x}) w_I], \quad \beta(\mathbf{x}) = \frac{V(\mathbf{x})}{\sum_{J=1}^{n} \dfrac{s_J(\mathbf{x})}{h_J(\mathbf{x})}} \tag{47}$$

Let $\mathbf{N}_\mathbf{x} = \{n_{\mathbf{x}1}, n_{\mathbf{x}2}, \ldots, n_{\mathbf{x}n}\}$ be a set of n natural neighbours for the point $\mathbf{x}$ and $\mathbf{N}_I = \{n_{I1}, n_{I2}, \ldots, n_{Ip_I}\}$ be the p_I natural neighbours for the point $\mathbf{x}_I$ $(n_{\mathbf{x}I} \notin \mathbf{N}_I)$. The set $\tilde{\mathbf{N}}$ of all nodes in the approximation is: $\tilde{\mathbf{N}} = \mathbf{N}_\mathbf{x} \cup \bigcup_{I=1}^{n} \mathbf{N}_I$. The cardinality of $\tilde{\mathbf{N}}$ is m. The expression for w_I is given in Equation (44a). We note that for any given node in $\mathbf{N}_\mathbf{x}$, there are two distinct sources of contribution from the second term of the above equation. By some simple algebraic manipulations, we arrive at the following:

$$\psi_I(\mathbf{x}) = \phi_I(\mathbf{x})\left[1+\beta(\mathbf{x})\frac{1}{V_I}\sum_{J=1}^{p_I}\frac{s_J^I}{h_J^I} - \beta(\mathbf{x})\sum_{J=1}^{p_I}\frac{1}{V_J}\frac{s_I^J}{h_I^J}\right], \quad n_I \in \mathbf{N}_\mathbf{x} \tag{48a}$$

$$\psi_I(\mathbf{x}) = -\beta(\mathbf{x})\sum_{\substack{J=1\\ n_I\in \mathrm{N}_J}}^{n}\frac{\phi_J(\mathbf{x})}{V_J}\frac{s_I^J}{h_I^J}, \quad n_I \notin \mathbf{N}_\mathbf{x} \tag{48b}$$

and since $s_I^J = s_J^I$ and $h_I^J = h_J^I$, we obtain the result

$$u^h(\mathbf{x}) = \sum_{I=1}^{m}\psi_I(\mathbf{x})u_I \tag{49a}$$

where

$$\psi_I(\mathbf{x}) = \phi_I(\mathbf{x})\left[1+\beta(\mathbf{x})\sum_{J=1}^{p_I}\frac{s_J^I}{h_J^I}\left(\frac{1}{V_I}-\frac{1}{V_J}\right)\right], \quad n_I \in \mathbf{N}_\mathbf{x} \tag{49b}$$

$$\psi_I(\mathbf{x}) = -\beta(\mathbf{x})\sum_{\substack{J=1\\ n_I\in \mathrm{N}_J}}^{n}\frac{\phi_J(\mathbf{x})}{V_J}\frac{s_I^J}{h_I^J}, \quad n_I \notin \mathbf{N}_\mathbf{x} \tag{49c}$$

The above higher-order scheme is simple and computationally attractive in numerical calculations [17, 18].

2.4. *Computational algorithm*

In Sukumar *et al.* [4], Watson's algorithm [19] was used for the computation of the Sibson interpolant. A drawback of Watson's algorithm is that it fails for points that lie on the edge of a Delaunay triangle. In order to overcome this shortcoming, we adopted the Bowyer–Watson algorithm [20, 21] as the basis for the evaluation of the Sibson and non-Sibsonian interpolants.

Natural neighbour (Sibson) interpolation is based on the area (volume) of intersection of polygons (polyhedrons) in $\mathbb{R}^2$ ($\mathbb{R}^3$). The polyhedron intersection problem is a non-trivial task in computational geometry, and hence easy-to-implement algorithms that are computationally attractive for Sibson computations are still unavailable. Owens [22] proposed a three-dimensional algorithm for natural neighbour interpolation and Braun and Sambridge [1] used Lasserre's algorithm [23] to compute the natural neighbour shape function in their PDE application. See Aftosmis [24] for a description of the many issues involved in the polyhedron intersection problem. Since the non-Sibsonian interpolant relies on the evaluation of a Lebesgue measure of one dimension less than the Sibson interpolant, the implementation of the non-Sibsonian interpolant in, both, 2-dimensions and 3-dimensions is viable. The essential ingredients of the computational algorithm for non-Sibsonian interpolation are presented in Table I. Standard template library (STL) containers in C++ are used in the implementation. The STL containers `map` and `multi-map` are used, where `map` is a container that maps an integer key to another integer or floating-point number, and `multi-map` is a map from an integer key to many integers or floating-point numbers. These containers provide fast sorting and searching on keys, which eases the implementation and leads to better computational efficiency.

Table I. Pseudo-code for computation of non-Sibsonian interpolant.

```
• Compute natural neighbour set $\mathbf{N}$ and set $\mathbf{T}$ of deleted simplices for point $\mathbf{x}$
    — Find simplex $t$ containing $\mathbf{x}$ and set $\mathbf{T} \leftarrow t$
    — Test all neighbouring simplices $t_i$ of $t$. If $\|\mathbf{x}-\mathbf{v}_i\|^2 < R_i^2$, then update $\mathbf{T} \leftarrow \{\mathbf{T}, t_i\}$
    — Set $\mathbf{N} = \{n_I\colon n_I \in t_i,\ t_i \in \mathbf{T}\}$

• Create boundary facet set $\mathbf{F}$ ($\mathbf{F} \leftarrow \{\}$)
    — Let $f_j$ be a facet of $t_i \in \mathbf{T}$ and $\hat{t}_j$ be its neighbouring simplex. For each $t_i \in \mathbf{T}$, if ($\hat{t}_j \notin \mathbf{T}$ or $\hat{t}_j = 0$) update $\mathbf{F} \leftarrow \{\mathbf{F}, f_j\}$

• Consider new triangles (tetrahedrons) $s_j \in \mathbf{S}$ formed by a facet $f_j \in \mathbf{F}$ and the point $\mathbf{x}$. Each $s_j$ contains the circumcentre $\mathbf{c}_{ji}$ and its derivatives $\mathbf{c}_{ji,k}$. Create a multi-map between $n_I \in \mathbf{N} \rightarrow s_j$ for all $s_j \in \mathbf{S}$

• Computations for each $n_I \in \mathbf{N}$ ($A = 0$; $A_k = 0$)
    — Using the multi-map, the set $\mathbf{V}_I$ of Voronoi vertices $v_i$ for the node $n_I$ is obtained
    — If $\mathbf{V}_I$. size() $\neq 2$, re-order the vertices in counter-clockwise orientation; in $\mathbb{R}^3$, the re-ordering of the vertices is carried out on the plane which contains all the vertices.
    — Compute $s_I$ [length (area) in $\mathbb{R}^2$ ($\mathbb{R}^3$)] and its derivatives $s_{I,k}$
    — Compute $h_I = d(\mathbf{x}, \mathbf{x}_I)$ and $h_{I,k}$
    — Evaluate $\alpha_I = s_I/h_I$ and its derivatives $\alpha_{I,k}$
    — Update $A \leftarrow A + \alpha_I$, $A_k \leftarrow A_k + \alpha_{I,k}$

• Shape function $\phi_I(\mathbf{x}) = \alpha_I/\Sigma_J\, \alpha_J$ and its derivatives $\phi_{I,k}(\mathbf{x})$ are evaluated for all $n_I \in \mathbf{N}$
```

In Figure 6(a), a regular 5×5 grid is shown and in Figures 6(b) and 6(c), the Sibson and non-Sibsonian shape function associated with node A are illustrated.

3. IMPOSITION OF ESSENTIAL BOUNDARY CONDITIONS

The Sibson interpolant is precisely linear on the boundary of convex domains; however, this property of linear behaviour is not exactly satisfied for the Sibson interpolant if the boundary is part of a non-convex domain [4]. In practice, with sufficient refinement along such a boundary, almost linear behaviour is obtained. In most meshless methods that are based on moving least-squares approximants, the approximations do not interpolate nor yield precisely linear behaviour on the essential boundary which makes imposition and satisfaction of essential conditions non-trivial in these classes of meshless methods. In the following sub-sections, we discuss the behaviour of the non-Sibsonian interpolant on the boundary of convex and non-convex domains. It is shown that the displacements are precisely linear on the essential boundary in both cases, and hence essential boundary conditions in NEM using the non-Sibsonian shape functions can be imposed exactly as in finite elements.

3.1. *Convex domain*

The discrete model consists of a set of nodes N that describes a convex domain Ω, with $\partial\Omega$ represented by the boundary of the convex hull CH(N). On the boundary of the convex hull, the

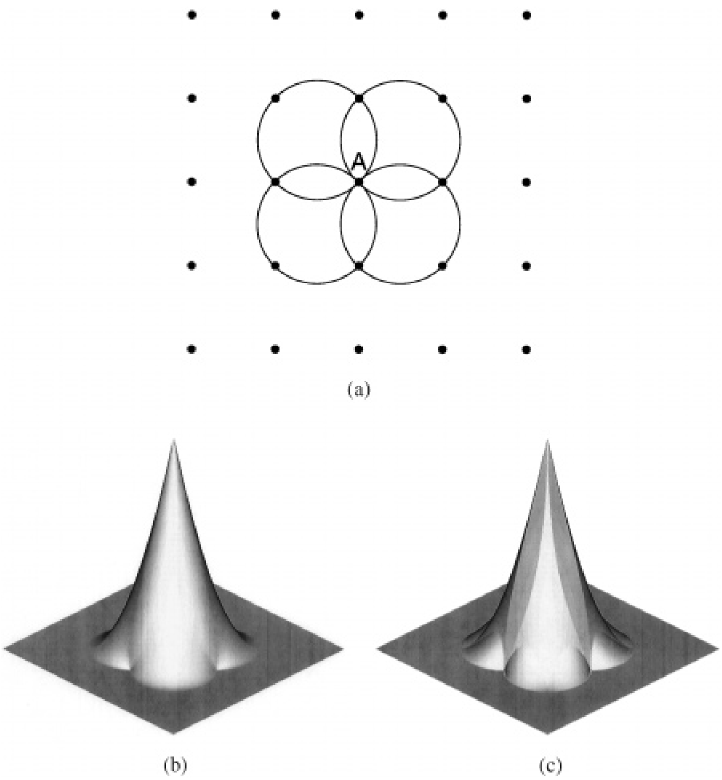

Figure 6. Sibson and non-Sibsonian shape functions: (a) Nodal grid; (b) Sibson shape function $\phi_A^S(\mathbf{x})$; and (c) non-Sibson shape function $\phi_A^{\mathrm{nS}}(\mathbf{x})$.

trial functions $\mathbf{u}^h(\mathbf{x})$ are strictly linear between two nodes that belong to an edge of a Delaunay triangle. The proof follows:

Proof. Consider a typical Delaunay triangle which has one edge (two nodes) along the boundary of the convex hull, and the trial functions $\mathbf{u}^h(\xi)$ are to be evaluated at a point ξ along the edge 1–2 (Figure 7). For simplicity and for ease of illustration, we assume that ξ has only three natural neighbours, namely nodes 1, 2, and 3. We use a local co-ordinate system ξ along the edge 1–2 such that $\xi=0$ at node 1 and $\xi=1$ at node 2. The first-order Voronoi cell for ξ is shown in Figure 7. By definition, the non-Sibsonian shape functions can be written as

$$\phi_I(\xi)=\frac{\dfrac{s_I(\xi)}{h_I(\xi)}}{\displaystyle\sum_{J=1}^{3}\frac{s_J(\xi)}{h_J(\xi)}}, \quad (I=1-3) \tag{50}$$

Since the length of the Voronoi edge associated with nodes 1 and 2 on the boundary of the convex hull is unbounded, we can express the Lebesgue measures $s_1(\xi)$, $s_2(\xi)$, and $s_3(\xi)$

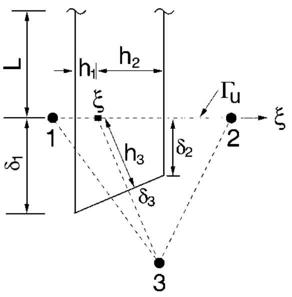

Figure 7. Linear behaviour of $\mathbf{u}^h(\xi)$ along the boundary of a convex domain.

as

$$s_1(\xi) = \lim_{L\to\infty} L + \delta_1(\xi), \quad s_2(\xi) = \lim_{L\to\infty} L + \delta_2(\xi), \quad s_3(\xi) = \delta_3(\xi) \tag{51}$$

where $\delta_1(\xi)$, $\delta_2(\xi)$, and $\delta_3(\xi)$ are finite. In addition, we also have $h_1(\xi) = \xi/2$, $h_2(\xi) = (1-\xi)/2$, and $h_3(\xi)$ is finite. On using Equation (50), we can write

$$\phi_1(\xi) = \lim_{L\to\infty} \frac{(L+\delta_1(\xi))(1-\xi)h_3(\xi)}{(L+\delta_1(\xi))(1-\xi)h_3(\xi) + (L+\delta_2(\xi))\xi h_3(\xi) + \delta_3(\xi)(1-\xi)\xi} \tag{52a}$$

$$\phi_2(\xi) = \lim_{L\to\infty} \frac{(L+\delta_2(\xi))\xi h_3(\xi)}{(L+\delta_1(\xi))(1-\xi)h_3(\xi) + (L+\delta_2(\xi))\xi h_3(\xi) + \delta_3(\xi)(1-\xi)\xi} \tag{52b}$$

$$\phi_3(\xi) = \lim_{L\to\infty} \frac{\delta_3(\xi)(1-\xi)\xi}{(L+\delta_1(\xi))(1-\xi)h_3(\xi) + (L+\delta_2(\xi))\xi h_3(\xi) + \delta_3(\xi)(1-\xi)\xi} \tag{52c}$$

Taking the limit as $L \to \infty$ in the above equations, we obtain

$$\phi_1(\xi) = 1-\xi, \quad \phi_2(\xi) = \xi, \quad \phi_3(\xi) = 0 \tag{53}$$

and hence along the edge 1–2, the shape function contributions from only nodes 1 and 2 are non-zero. The above result is in general true, even if more than three natural neighbours are considered. This is so, since the Lebesgue measure associated with all interior nodes is finite—similar to $\delta_3(\xi)$ in Figure 7. Using the above equation, the trial functions at the point ξ can be written as

$$\mathbf{u}^h(\xi) = (1-\xi)\mathbf{u}_1 + \xi\mathbf{u}_2 \tag{54}$$

which are linear functions, and hence the proof. □

3.2. Non-convex domain

Consider a set of nodes $\mathbf{N}$ that describes a non-convex domain $\Omega \subset \mathbb{R}^2$. Consider a Γ_1 ($\Gamma_1 \subset \Gamma = \partial\Omega$) which renders the domain to be non-convex. For purpose of illustration, we choose a non-convex domain bounded by two concentric circles. The discrete model (one-quarter) along with the Voronoi diagram are shown in Figure 8. It is evident that the Voronoi cells for the nodes along Γ_i

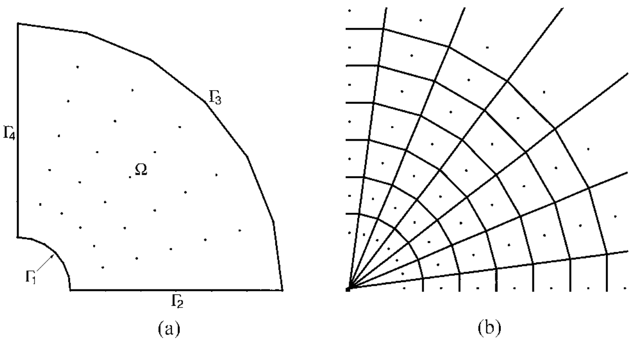

Figure 8. Linear behaviour of $\mathbf{u}^h(\mathbf{x})$ along the boundary of a non-convex domain: (a) Nodal discretization; and (b) Voronoi diagram.

$(i=2\text{–}4)$ are unbounded, whereas the Voronoi cells for the nodes along Γ_1 are bounded and therefore have finite areas (Figure 8(b)). The Sibson interpolant is strictly linear along $\tilde{\Gamma}=\Gamma_2\cup\Gamma_3\cup\Gamma_4$; however, the interpolant is not linear between adjacent nodes on the boundary Γ_1 since for $\mathbf{x}\in\Gamma_1$, there exists non-zero shape function contributions at $\mathbf{x}$ from some interior nodes [4].

Let the essential boundary $\Gamma_u=\Gamma_1$. In Figure 9, we show four contiguous nodes (nodes 1, 2, a, and b) along Γ_u. Let ξ be a local co-ordinate system along the edge 1–2 such that $\xi=0$ at node 1 and $\xi=1$ at node 2. We consider a point $\xi\in\Gamma_u$ which is located along the Delaunay edge 1–2. If a point $\mathbf{x}$ lies inside the circumcircle of a Delaunay triangle, then the Delaunay triangle is known as the circum-triangle of point $\mathbf{x}$. By the Delaunay empty circumcircle criterion, all circum-triangles of point ξ must have nodes 1 and 2 as its vertices. This immediately precludes the nodes along Γ_u that are shown by open circles (Figure 9) to be natural neighbours of ξ. We assume that ξ has four natural neighbours, namely nodes numbered 1–4 and shown by the dark circles in Figure 9. The Voronoi cell for point ξ and the length measures $s_I(\mathbf{x})$ and $h_I(\mathbf{x})$ that appear in the definition of the non-Sibsonian interpolant are shown in Figure 9. We note the Lebesgue measure $s_I(\xi)$ associated with nodes 1 and 2 are unbounded, and those for the interior nodes 3 and 4 are finite. It is apparent that the computation of the shape functions is now similar to the convex case, and by following similar arguments (see the proof in Section 3.1), we again arrive at the conclusion that the NEM trial functions using non-Sibsonian shape functions are precisely linear along the edge 1–2, i.e.

$$\mathbf{u}^h(\xi)=(1-\xi)\mathbf{u}_1+\xi\mathbf{u}_2 \tag{55}$$

A consequence of the above discussion is that by choosing non-Sibson shape function as trial and test functions in the natural element method, essential boundary conditions can be directly imposed on the nodes, as in the finite element method—this is due to the interpolating property of the shape functions and the linearity of the approximation along the essential boundary of the domain. It is to be noted that the above inference is rigorously true for convex as well as non-convex domains. The natural element method with non-Sibsonian shape functions is the only meshless Galerkin method that we are aware of that will exactly satisfy (linear) essential boundary conditions.

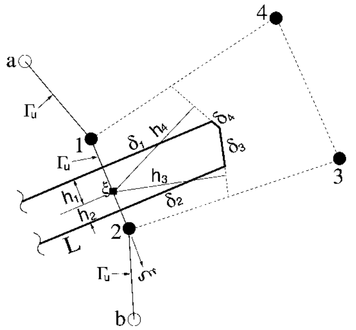

Figure 9. Non-Sibsonian interpolation along the boundary of a non-convex domain.

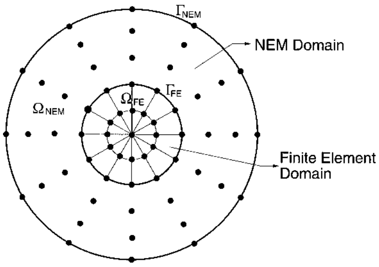

Figure 10. Finite element and natural element method coupling.

4. FE AND NEM COUPLING

The coupling between linear finite elements and the natural element method using the non-Sibsonian interpolant, is straight-forward. In Figure 10, a domain $\Omega = \Omega_{\mathrm{FE}} \cup \Omega_{\mathrm{NEM}}$ is illustrated, where Ω_{FE} is the finite element domain and Ω_{NEM} is the NEM domain. Since the non-Sibsonian interpolant is precisely linear on the boundary of convex as well as non-convex domains, a seamless procedure emerges to implement the coupling. By adopting ‘regional’ interpolation, i.e. finite element interpolation carried out in Ω_{FE} and NEM interpolation in Ω_{NEM}, the trial function in the domain Ω is given by

$$\mathbf{u}^h(\mathbf{x}) = \begin{cases} \sum_I \phi_I^{\mathrm{FE}}(\mathbf{x})\mathbf{u}_I & \text{if } \mathbf{x} \in \Omega_{\mathrm{FE}} \\ \sum_I \phi_I^{\mathrm{NEM}}(\mathbf{x})\mathbf{u}_I & \text{if } \mathbf{x} \in \Omega_{\mathrm{NEM}} \end{cases} \tag{56}$$

which completes the coupling. As opposed to coupling finite elements to other meshless methods (see Belytschko *et al.* [25] for instance), in the present implementation, the coupling is natural with no need for a blending domain nor modified shape functions.

5. GOVERNING EQUATIONS

In this section, we present the governing equations of linear elastostatics, together with the weak form and the discrete system for the natural element method.

5.1. Strong form

Consider a body which is described by an open bounded domain $\Omega \subset \mathbb{R}^2$, with boundary Γ. The boundary Γ is composed of the sets Γ_u and Γ_t, such that $\Gamma = \Gamma_u \cup \Gamma_t$ and $\Gamma_u \cap \Gamma_t = \emptyset$. The field equations of elastostatics are

$$\nabla \cdot \boldsymbol{\sigma} + \mathbf{b} = 0 \quad \text{in } \Omega \tag{57a}$$

$$\boldsymbol{\sigma} = \mathbf{C} : (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^*) \tag{57b}$$

$$\boldsymbol{\varepsilon} = \boldsymbol{\nabla}_{\mathrm{s}}\mathbf{u} \tag{57c}$$

where $\boldsymbol{\nabla}_{\mathrm{s}}$ is the symmetric gradient operator, $\mathbf{b}$ is the body force vector per unit volume, $\boldsymbol{\varepsilon}$ is the small strain tensor, $\boldsymbol{\varepsilon}^*$ is an imposed eigenstrain tensor, and $\mathbf{C}$ is the material moduli tensor for a homogeneous isotropic material. The eigenstrain tensor is included in the formulation to permit the treatment of a transformation strain problem that appears in Section 6.2.

Displacement (essential) boundary conditions are imposed on Γ_u and traction (natural) boundary condition on Γ_t. The essential and natural boundary conditions are:

$$\mathbf{u} = \bar{\mathbf{u}} \quad \text{on } \Gamma_u \tag{58a}$$

$$\boldsymbol{\sigma} \cdot \mathbf{n} = \bar{\mathbf{t}} \quad \text{on } \Gamma_t \tag{58b}$$

where $\mathbf{n}$ is the unit outward normal to Ω, and $\bar{\mathbf{u}}$ and $\bar{\mathbf{t}}$ are prescribed displacements and tractions, respectively.

5.2. *Weak form*

Let $\mathbf{u}$ be the displacement solution for the stated elastostatic boundary value problem, with $\boldsymbol{\sigma}(\mathbf{u})$ the corresponding Cauchy stress tensor. Let $\mathbf{u} \in \mathbf{V}$ be the displacement trial solution, and $\mathbf{v} \in \mathbf{V}_0$ be any set of kinematically admissible test functions (virtual displacements). The space $\mathbf{V} = \mathbb{H}^1(\Omega)$ is the Sobolev space of functions with square-integrable first derivatives in Ω, and $\mathbf{V}_0 = \mathbb{H}^1_0(\Omega)$ is the Sobolev space of functions with square-integrable first derivatives in Ω and vanishing values on the essential boundary Γ_u. The weak form of the governing equation and associated boundary conditions can be written as

$$\begin{gathered}\text{Find } \mathbf{u} \in \mathbf{V} \text{ such that} \\ \int_{\Omega} \boldsymbol{\sigma}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{v})\, \mathrm{d}\Omega = \int_{\Omega} \mathbf{b} \cdot \mathbf{v}\, \mathrm{d}\Omega + \int_{\Gamma_t} \bar{\mathbf{t}} \cdot \mathbf{v}\, \mathrm{d}\Gamma \quad \forall \mathbf{v} \in \mathbf{V}_0\end{gathered} \tag{59}$$

5.3. *Discrete system*

Consider the Bubnov–Galerkin implementation for NEM in two-dimensional linear elasticity. In NEM, finite-dimensional subspaces $\mathbf{V}^h \subset \mathbf{V}$ and $\mathbf{V}^h_0 \subset \mathbf{V}_0$ are used as the approximating trial and test spaces. The weak form for the discrete problem can be stated as:

$$\begin{gathered}\text{Find } \mathbf{u}^h \in \mathbf{V}^h \subset \mathbf{V} \text{ such that} \\ \int_{\Omega^h} \boldsymbol{\sigma}(\mathbf{u}^h) : \boldsymbol{\varepsilon}(\mathbf{v}^h) = \int_{\Omega^h} \mathbf{b} \cdot \mathbf{v}\, \mathrm{d}\Omega + \int_{\Gamma^h_t} \bar{\mathbf{t}} \cdot \mathbf{v}\, \mathrm{d}\Gamma \quad \forall \mathbf{v}^h \in \mathbf{V}^h_0 \subset \mathbf{V}_0\end{gathered} \tag{60}$$

In a Bubnov–Galerkin procedure, the trial functions $\mathbf{u}^h$ as well as the test functions $\mathbf{v}^h$ are represented as linear combinations of the same shape functions. The trial and test functions are

$$\mathbf{u}^h(\mathbf{x}) = \sum_I \phi_I(\mathbf{x})\mathbf{u}_I, \quad \mathbf{v}^h(\mathbf{x}) = \sum_I \phi_I(\mathbf{x})\mathbf{v}_I \tag{61}$$

where $\phi_I(\mathbf{x})$ are the NEM shape functions.

On substituting the trial and test functions from Equation (61) in Equation (60), and using the arbitrariness of nodal variations, the following discrete system of linear equations is obtained:

$$\mathbf{K}\mathbf{d} = \mathbf{f} \tag{62}$$

where

$$\mathbf{K}_{IJ} = \int_{\Omega^h} \mathbf{B}_I^{\mathrm{T}} \mathbf{C} \mathbf{B}_J \, \mathrm{d}\Omega \tag{63a}$$

$$\mathbf{f}_I = \int_{\Gamma_t^h} \phi_I \, \bar{\mathbf{t}} \, \mathrm{d}\Gamma + \int_{\Omega^h} \phi_I \mathbf{b} \, \mathrm{d}\Omega + \int_{\Omega^h} \mathbf{B}_I^{\mathrm{T}} \mathbf{C} \boldsymbol{\varepsilon}^* \, \mathrm{d}\Omega \tag{63b}$$

where $\mathbf{C}$ is the constitutive matrix for an isotropic linear elastic material, and $\mathbf{B}_I$ is the matrix of shape function derivatives which is given by

$$\mathbf{B}_I = \begin{bmatrix} \phi_{I,x} & 0 \\ 0 & \phi_{I,y} \\ \phi_{I,y} & \phi_{I,x} \end{bmatrix} \tag{64}$$

6. NUMERICAL RESULTS

The application of the natural element method to an inclusion problem in small displacement two-dimensional elastostatics, in the absence of body forces, is presented. First, we conduct an eigenanalysis to study the properties of the Sibson and non-Sibsonian interpolating spaces. Numerical integration is carried out using symmetric quadrature rules for a triangle. In this paper, three point quadrature rule is used in the numerical integration of the weak form.

6.1. Eigenanalysis

In order to study the properties of the approximation spaces of NEM, we consider the linear independence of the shape functions [26]. To this end, we consider the following discrete eigenvalue problem:

$$\mathbf{M}^h \mathbf{d}^h = \lambda^h \mathbf{d}^h \tag{65}$$

where $\mathbf{d}^h$ and λ^h are the eigenvectors and eigenvalues of $\mathbf{M}^h$, and $\mathbf{M}^h$ is the mass matrix which is given by

$$\mathbf{M}_{IJ}^h = \int_{\Omega^h} \phi_I \phi_J \, \mathrm{d}\Omega \tag{66}$$

The condition number $\kappa(\mathbf{M}^h)$ of the matrix $\mathbf{M}^h$ is defined as the ratio of the maximum eigenvalue $\lambda_{\max}$ to that of the minimum eigenvalue $\lambda_{\min}$. The condition number is used as a measure of the linear independence of the shape functions, with $\kappa = 1$ indicating optimality. The EISPACK [27] eigensolver package is used to solve the eigenvalue problem. In Table II, the condition number

Table II. Linear independence of NEM shape functions.

		κ^h		κ^h_{D}
Grids	Nodes	Sibson	Non-Sibson	FEM
Uniform	25	3.83	3.57	7.00
	289	3.83	3.57	7.00
	4225	3.83	3.57	7.00
Non-uniform	70	35.65	40.80	43.17
	278	2146.44	1887.29	3422.31

is computed for three different nodal discretizations. The condition number is computed for a regular nodal grid (Figure 11(a)) and uniform refinements of the same grid; in addition, results are also computed for a grid with random location of nodes (Figure 11(b)), and for the irregular focused grid shown in Figure 11(c). The condition numbers obtained for NEM using the Sibson and non-Sibsonian interpolants are of the same order. The NEM results for both uniform and non-uniform nodal discretizations are better than those obtained using constant strain finite elements which indicates that the matrix system is well-conditioned and the NEM approximation spaces are linearly independent.

6.2. *Infinite plate with an inclusion*

We consider the problem of an inclusion (α-phase) with a constant eigenstrain $\boldsymbol{\varepsilon}^*_\alpha$ in an infinite matrix (β-phase). In Figure 12, a graphical representation of the problem is illustrated. The exact displacement vector solution in polar co-ordinates is given by Mura [28]

$$u_r(r) = \begin{cases} C_1 r & r \leqslant R \\ C_1 \frac{R^2}{r} & r \geqslant R \end{cases} \tag{67a}$$

$$u_\theta = 0 \tag{67b}$$

where

$$C_1 = \frac{(\mu^\alpha + \lambda^\alpha)\bar{\varepsilon}^*_\alpha}{\mu^\alpha + \lambda^\alpha + \mu^\beta} \tag{68}$$

In the above equation, μ and λ are the Lamé constants of the respective phases, and the eigenstrain $\bar{\varepsilon}^*_\alpha$ is a constant dilatational strain. The material properties used in the numerical computation are [29]: $\lambda^\alpha = 497.16$, $\mu^\alpha = 390.63$ in the α-phase, and the constants in the β-phase are $\lambda^\beta = 656.79$, $\mu^\beta = 338.35$. These correspond to $E^\alpha = 1000$, $\nu^\alpha = 0.28$, $E^\beta = 900$, and $\nu^\beta = 0.33$. A constant dilatational eigenstrain $\bar{\varepsilon}^*_\alpha = 0.01$ is assumed in the analysis, and the associated eigenstrain tensor is $\boldsymbol{\varepsilon}^*_\alpha = \bar{\varepsilon}^*_\alpha(\mathbf{e}_1\mathbf{e}_1 + \mathbf{e}_2\mathbf{e}_2)$.

The numerical model (quarter symmetry) is shown in Figure 13, where the nodal discretization consists of 647 nodes, with 114 nodes in the inclusion, 520 nodes in the matrix, and 13 nodes along the interface $r = R$. The outer radius $R_0 = 200$ is sufficiently large in comparison to the radius of the inclusion $R = 5$, so as to adequately represent the infinite matrix. Essential boundary

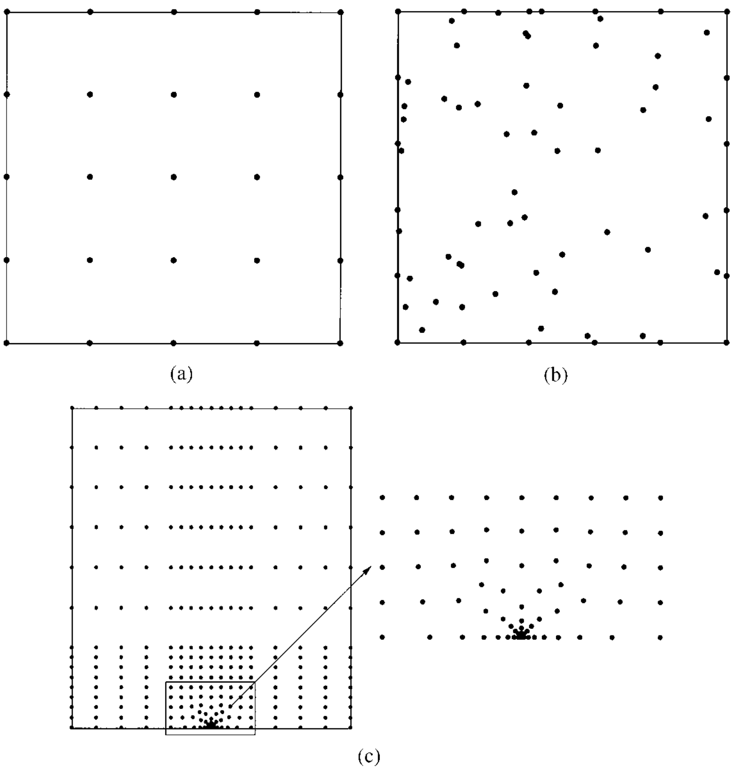

Figure 11. Nodal grids for eigenanalysis: (a) uniform grid (25 nodes); (b) random set (70 nodes); and (c) irregular focused grid (278 nodes).

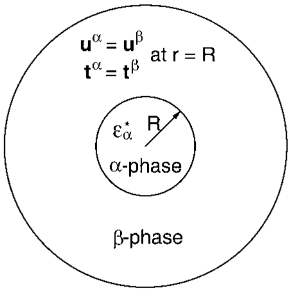
$\mathbf{u}^{\alpha} = \mathbf{u}^{\beta}$
$\mathbf{t}^{\alpha} = \mathbf{t}^{\beta}$
at r = R
ε^{*}_{α}
R
α-phase
β-phase

Figure 12. Inclusion embedded in an infinite matrix.

conditions are imposed along the lines of symmetry, and the outer radius $R_0 = 200$ is traction free. Plane strain conditions are assumed in the numerical computations. The numerical computations are carried out using the Sibson interpolant, non-Sibsonian interpolant, and the FE–NEM coupling procedure outlined in Section 4. The NEM solution recovered the cylindrical symmetry in the

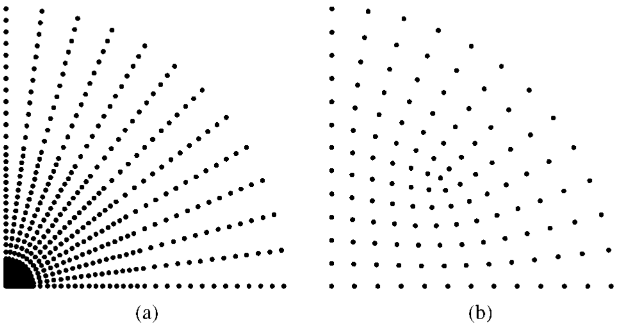

Figure 13. Nodal discretization for inclusion in an infinite matrix problem: (a) quarter model; (b) inclusion (α phase).

solution, and hence results are presented as a function of only the radial distance. The numerical results using the Sibson interpolant, non-Sibsonian interpolant, and the FE–NEM coupling technique compare favourably to each other. In Figure 14, a comparison of the numerical and exact solution is presented. The results shown in Figure 14 are computed along a radial line ($r=0$ to $r=25$) at $\theta=30°$. Along the radial line, 30 equi-spaced output points are chosen within the inclusion, and 30 equi-spaced points in the matrix. Excellent agreement between the NEM and the analytical solution is observed. The strains as well as the stresses are in good agreement with the exact solution. The oscillations in the radial and hoop strains are negligible; they are, however, a bit more pronounced in the stress solutions.

In order to show the linear behaviour of the non-Sibsonian interpolant along the boundary Γ_α, we consider two nodes A and B along Γ_α (Figure 15(a)). In Figure 15(b), the non-Sibsonian shape functions ϕ_A and ϕ_B are plotted along the element edge, with ξ a local co-ordinate along A–B. The numerical results are computed along the line $\tilde{\mathbf{x}}_A$ to $\mathbf{x}_B$, where $\tilde{\mathbf{x}}_A=(x_A+\varepsilon, y_A)$ with $\varepsilon=10^{-12}$ being a small tolerance. The tolerance is required in the numerical computations since for a point $\mathbf{x}\in\Gamma_\alpha$, the length of the Voronoi edges associated with nodes A and B is unbounded. It is seen from Figure 15 that a linear displacement approximation along A–B is realized in the numerical computations, which supports the theoretical proof presented in Section 3.2. This shows that essential boundary conditions in NEM using the non-Sibsonian interpolant can be prescribed exactly as in finite elements. In addition, these numerical results validate the FE–NEM coupling technique that is described in Section 4.

7. CONCLUSIONS

In this paper, we introduced the use of natural neighbour-based interpolants (Sibson and non-Sibsonian) for the solution of elliptic partial differential equations. The numerical implementation of these interpolants in a Galerkin method is known as the natural element method (NEM). The notion of natural neighbours relies on the Voronoi diagram which is unique for a given set of scattered nodes in $\mathbb{R}^d$. The local density and spatial location of nodes is taken into account in the construction of the natural neighbour-based interpolants. Some of the most important properties

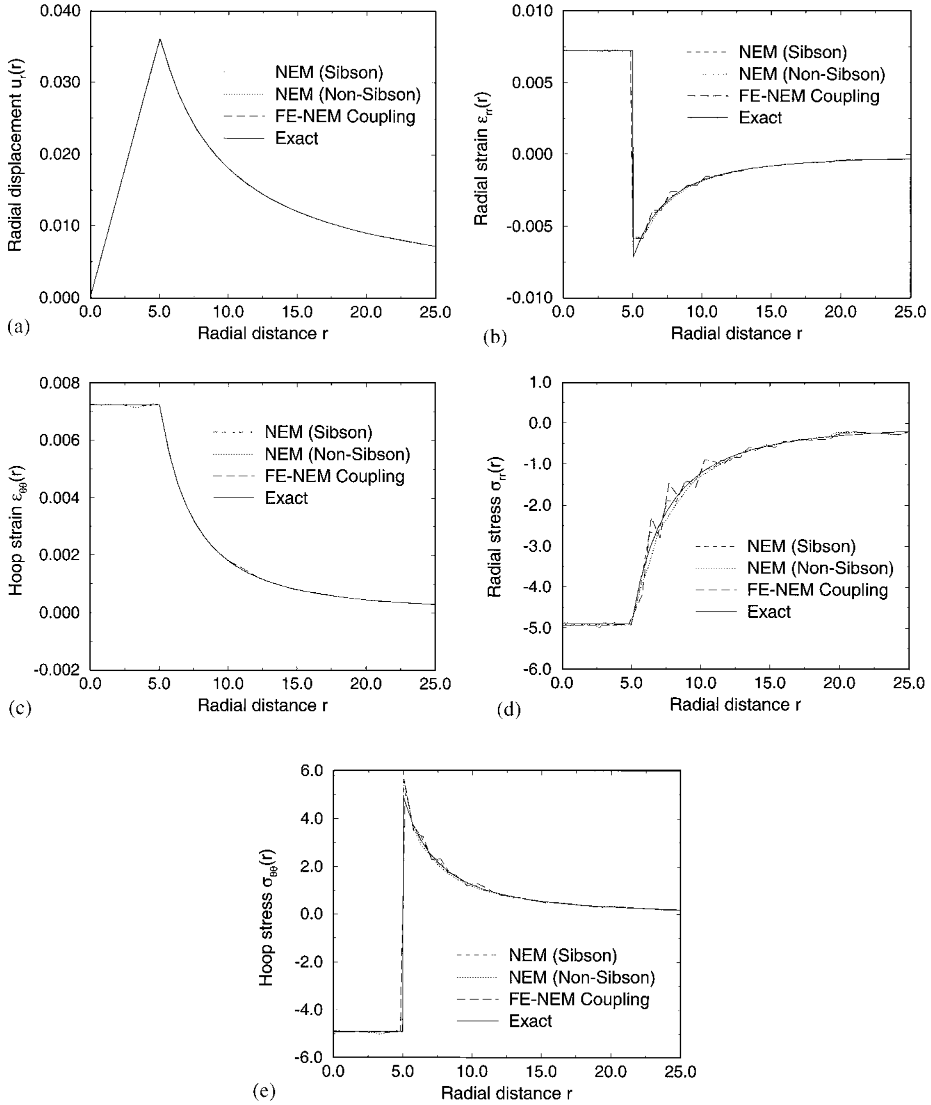

Figure 14. Comparison of NEM and the exact solution for an inclusion with a dilatational eigenstrain in an infinite matrix: (a) radial displacement $u_r(r)$; (b) radial strain $\varepsilon_{rr}(r)$; (c) hoop strain $\varepsilon_{\theta\theta}(r)$; (d) radial stress $\sigma_{rr}(r)$; and (e) hoop stress $\sigma_{\theta\theta}(r)$.

of the non-Sibsonian interpolant were reviewed and new results on the imposition of essential boundary conditions were presented.

An eigenanalysis of the Sibson and non-Sibsonian interpolants was carried out, and the meshless interpolating spaces were found to be uniformly linearly independent. The computational efficiency of the non-Sibsonian interpolants over the Sibson interpolant was demonstrated and a versatile algorithm for 2- and 3-dimensional non-Sibsonian computations was presented. The NEM results

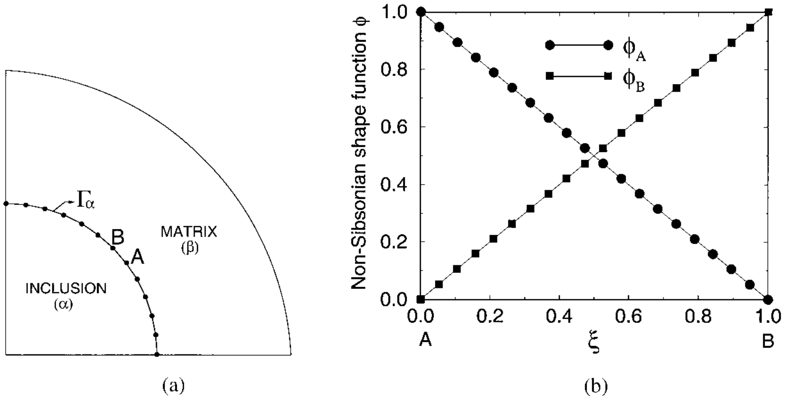

Figure 15. Linear approximation along a non-convex boundary: (a) quarter model; and (b) non-Sibsonian shape functions along Γ_α.

for the problem of an inclusion with a constant eigenstrain, embedded in an infinite matrix, were found to be in good agreement with the exact solution. Essential boundary conditions in NEM using the non-Sibsonian interpolants can be imposed as in finite elements, for both, convex, and non-convex domains. A rigorous proof to this fact was presented, which was numerically verified. A simple technique to couple finite elements to the natural element method was also described. The robust and appealing properties of natural neighbour-based interpolants, together with the seamless means to couple NEM to FEM and the ease of modelling material interfaces opens up many exciting areas of mechanics research that can be explored by this new technique.

ACKNOWLEDGEMENTS

The authors are grateful for the research support of the National Science Foundation through contract CMS-9732319.

REFERENCES

1. Braun J, Sambridge M. A numerical method for solving partial differential equations on highly irregular evolving grids. *Nature* 1995; **376**:655–660.
2. Sibson R. A vector identity for the Dirichlet tesselation. *Mathematical Proceedings of the Cambridge Philosophical Society* 1980; **87**:151–155.
3. Sukumar N. The natural element method in solid mechanics. *Ph.D. Thesis*, Theoretical and Applied Mechanics, Northwestern University: Evanston, IL, USA, 1998.
4. Sukumar N, Moran B, Belytschko T. The natural element method in solid mechanics. *International Journal for Numerical Methods in Engineering* 1998; **43**(5):839–887.
5. Sukumar N, Moran B. C^1 natural neighbour interpolant for partial differential equations. *Numerical Methods for Partial Differential Equations* 1999; **15**(4):417–447.
6. Bueche D, Sukumar N, Moran B. Dispersive properties of the natural element method. *Computational Mechanics* 2000; **25**(2/3):207–219.
7. Belikov VV, Ivanov VD, Kontorovich VK, Korytnik SA, Semenov AY. The non-Sibsonian interpolation: a new method of interpolation of the values of a function on an arbitrary set of points. *Computational Mathematics and Mathematical Physics* 1997; **37**(1):9–15.
8. Belytschko T, Krongauz Y, Organ D, Fleming M, Krysl P. Meshless methods: an overview and recent developments. *Computer Methods in Applied Mechanics and Engineering* 1996; **139**:3–47.

9. Lancaster P, Salkauskas K. *Curve and Surface Fitting*. Academic Press: London, 1986.
10. Belikov VV, Semenov AY. Non-Sibsonian interpolation on arbitrary system of points in euclidean space and adaptive generating isolines algorithm. In *Numerical Grid Generation in Computational Field Simulations*, Cross M, Soni BK, Thompson JF, Hauser J, Eiseman PR (eds). University of Greenwich: London, UK 1998; 277–286.
11. Green PJ, Sibson RR. Computing Dirichlet tessellations in the plane. *The Computer Journal* 1978; **21**:168–173.
12. Lawson CL. Software for C^1 surface interpolation. In *Mathematical Software III*, Rice JR (ed.), vol. 3. Academic Press: New York, NY, 1977.
13. Watson DF. *Contouring*: *A Guide to the Analysis and Display of Spatial Data*. Pergamon Press: Oxford, 1992.
14. Jones NL, Owens SJ, Perry EC. Plume characterization with natural neighbour interpolation. *Proceedings GEOENVIRONMENT 2000*, *Geotechnical Engineering and Environmental Engineering Divisions/ASCE*. New York, NY 1995; 331–345.
15. Braun J, Sambridge M, McQueen H. Geophysical parameterization and interpolation of irregular data using natural neighbours. *Geophysical Journal International* 1995; **122**:837–857.
16. Farin G. Surfaces over Dirichlet tessellations. *Computer Aided Geometric Design* 1990; **7**(1–4):281–292.
17. Belikov VV, Semenov AY. New non-Sibsonian interpolation on arbitrary system of points in Euclidean space. In *15th IMACS World Congress*, *Numerical Mathematics*, vol. 2, Wissen Tech. Verlag: Berlin 1997; 237–242.
18. Belikov VV, Semenov AY. Non-Sibsonian interpolation on arbitrary system of points in Euclidean space and adaptive isolines generation. *Applied Numerical Mathematics* 2000; **32**(4):371–387.
19. Watson DF. *nngridr*: *An implementation of natural neighbour interpolation*. David Watson, 1994.
20. Bowyer A. Computing Dirichlet tessellations. *Computer Journal* 1981; **24**:162–166.
21. Watson DF. Computing the n-Dimensional Delaunay tessellation with application to Voronoi polytopes. *The Computer Journal* 1981; **24**(2):167–172.
22. Owens SJ. An implementation of natural neighbour interpolation in three dimensions. *Master's Thesis*, Brigham Young University, 1992.
23. Lasserre JB. An analytical expression and an algorithm for the volume of a convex polyhedron in R^n. *Journal of Optimization Theory and Applications* 1983; **39**(3):363–377.
24. Aftosmis AF. Solution adaptive cartesian grid methods for aerodynamic flows with complex geometries. Lecture Notes for 28th Computational Fluid Dynamics Lecture Series. von Karman Institute for Fluid Dynamics: Rhode-Saint-Genèse, Belgium, 1997.
25. Belytschko T, Organ D, Krongauz Y. A coupled finite element-element-free Galerkin method. *Computational Mechanics* 1995; **17**:186–195.
26. Strang G, Fix G. *An Analysis of the Finite Element Method*. Prentice-Hall: Englewood Cliffs, NJ, 1973.
27. Smith BT, Boyle JM, Garbow BS, Ikebe Y, Klema VC, Moler CB. *Matrix Eigensystem Routines—EISPACK Guide*. Springer: New York, 1974.
28. Mura T. *Micromechanics of Defects in Solids*. Martinus Nijhoff: The Hague, Netherlands, 1987.
29. Cordes LW, Moran B. Treatment of material discontinuity in the element-free Galerkin method. *Computer Methods in Applied Mechanics and Engineering* 1996; **139**:75–89.