INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng. **43**, 839–887 (1998)

THE NATURAL ELEMENT METHOD IN SOLID MECHANICS

N. SUKUMAR[†], B. MORAN[*,‡] AND T. BELYTSCHKO[§]

Department of Civil and Mechanical Engineering, Northwestern University, Evanston, IL 60208, U.S.A.

ABSTRACT

The application of the Natural Element Method (NEM)[1, 2] to boundary value problems in two-dimensional small displacement elastostatics is presented. The discrete model of the domain Ω consists of a set of distinct nodes N, and a polygonal description of the boundary $\partial\Omega$. In the Natural Element Method, the trial and test functions are constructed using natural neighbour interpolants. These interpolants are based on the Voronoi tessellation of the set of nodes N. The interpolants are smooth (C^∞) everywhere, except at the nodes where they are C^0. In one-dimension, NEM is identical to linear finite elements. The NEM interpolant is strictly linear between adjacent nodes on the boundary of the convex hull, which facilitates imposition of essential boundary conditions. A methodology to model material discontinuities and non-convex bodies (cracks) using NEM is also described. A standard displacement-based Galerkin procedure is used to obtain the discrete system of linear equations. Application of NEM to various problems in solid mechanics, which include, the patch test, gradient problems, bimaterial interface, and a static crack problem are presented. Excellent agreement with exact (analytical) solutions is obtained, which exemplifies the accuracy and robustness of NEM and suggests its potential application in the context of other classes of problems—crack growth, plates, and large deformations to name a few. © 1998 John Wiley & Sons, Ltd.

KEY WORDS: natural neighbour interpolation; natural element method; 1st- and 2nd-order Voronoi diagrams; Delaunay triangle; elastostatics

1. INTRODUCTION

The Finite Element Method (FEM) is a well-established numerical method which has been applied to boundary-value problems in different fields of engineering and the applied sciences. In spite of its numerous advantages, there is an on-going thrust in the development and application of new numerical tools which hold promise for certain classes of problems, such as crack growth, plate bending, and modelling of multiscale phenomena. In particular, there has been a tremendous surge in the emergence and development of so-called meshless methods[3–10] for the solution of Partial Differential Equations (PDEs). The mesh-free character of these methods is particularly attractive

* Correspondence to: B. Moran, Department of Civil Engineering, Northwestern University, 2145 Sheridan Road, Evanston, IL60208, U.S.A. E-mail: b-moran@nwu.edu
† Research Assistant, Theoretical and Applied Mechanics
‡ Associate Professor of Civil Engineering
§ Walter P. Murphy Professor of Computational Mechanics

Contract / grant sponsor: Federal Aviation Administration
Contract / grant sponsor: Office of Naval Research

CCC 0029–5981/98/050839–49$17.50 *Received 1 December 1997*

in the area of computational fracture mechanics, such as two-dimensional and three-dimensional crack growth modelling by the Element-Free Galerkin (EFG) method.[11]

In this paper, we assess the potential and capabilities of a recently developed numerical method—coined as Natural Neighbour Finite Elements[1] or Natural Element Method.[2] A detailed description of its implementation for PDEs is given by Braun and Sambridge,[2] who refer to the method as Natural Element Method (NEM), which is the name we choose to use here. The application of NEM to the solution of elliptic boundary value problems in solid mechanics, governed by the equation of equilibrium in elastostatics, is explored. The interpolation scheme used in NEM is known as Natural Neighbour (n–n) interpolation. Natural neighbour interpolation[12, 13] is a multivariate data interpolation scheme,[14] which has primarily been used in data interpolation and modelling of geophysical phenomena. For some of the previous works and typical applications of natural neighbour interpolation, see References 1, 2 and 15–28. Natural neighbour interpolation relies on concepts such as Voronoi diagrams[29] and Delaunay tessellations[30] in computational geometry,[31–33] to construct the interpolant. In spite of its fairly simple and appealing structure, sound theoretical basis for construction, and desirable smoothness properties, n–n interpolation has received little attention in the area of multivariate data interpolation, when compared to other schemes such as Shepard's interpolant,[34] moving least-squares approximants,[35] radial basis functions,[36, 37] or Hardy's multiquadrics.[38] Recent work on n–n interpolation[27] and its application to the modelling of complex fluid-structure interaction phenomena[2] does indicate the merits of the method for the solution of PDEs and suggests that it could be a promising numerical tool in the realm of solid mechanics.

The NEM interpolant is constructed on the basis of the underlying Voronoi tessellation, which is unique for a given set of distinct points (nodes) in the plane. The Delaunay triangles which are the dual of the Voronoi diagram are used in the numerical computation of the NEM interpolant. However, unlike the finite element method where angle restrictions are imposed on the triangles for the convergence of the method,[39] there are no such constraints on the shape, size, and angles of the triangles in NEM. This facilitates random configuration of nodes in space regardless of whether the associated Delaunay triangles are acceptable from a finite element perspective. In the application of NEM to the solution of the equilibrium equation of elastostatics, a displacement-based Galerkin implementation is used. The trial and test functions in the weak (variational) form are constructed using natural neighbour interpolants. The solution of the discrete system of linear equations ($\mathbf{Kd} = \mathbf{f}$) is carried out to obtain the nodal displacement vector $\mathbf{d}$.

The outline of this papers is as follows. In the following section, a brief description of the rudiments of Voronoi diagrams and Delaunay triangles in the context of natural neighbour interpolation is delineated. In Section 3, an extensive discussion of natural neighbour interpolants—its construction, properties, and computational procedure—is presented. The treatment of material discontinuities is addressed in Section 4, while the modelling of non-convex bodies (straight cracks) using NEM is described in Section 5. In Section 6, the governing equations of elastostatics together with the Galerkin formulation for NEM are described. In Section 7, various applications of NEM in two-dimensional solid mechanics, which include problems with homogeneous deformation (patch tests), steep gradients, strain discontinuity (bimaterial interface), as well as those with singularities (crack-tip fields) are presented. The results for the field variables as well as convergence studies are compared to exact (analytical) solutions and to the FEM, wherever appropriate. Finally, in Section 8, some concluding remarks and promising future applications of NEM are mentioned.

2. VORONOI DIAGRAM AND DELAUNAY TESSELLATION

The Voronoi diagram and its dual Delaunay tessellation (covering of a surface with tiles) are one of the most fundamental and useful geometric constructs that define an irregular set of points (nodes). For simplicity, and in keeping with the applications that are pursued in this paper, we consider two-dimensional Euclidean space $\mathbb{R}^2$; the theory, however, is applicable in a general d-dimensional framework. Consider a set of distinct nodes $N = \{n_1, n_2, \ldots, n_M\}$ in $\mathbb{R}^2$. The Voronoi diagram (or 1st-order Voronoi diagram) of the set 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 (also known as Thiessen or Voronoi polygons) of n_I. In mathematical terms, the Voronoi polygon T_I is defined as[40]

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

where $d(\mathbf{x}_I, \mathbf{x}_J)$, the Euclidean metric, is the distance between $\mathbf{x}_I$ and $\mathbf{x}_J$.

The Voronoi cell for node A and the Voronoi diagram for a set N consisting of seven nodes are shown in Figures 1(a) and 1(b), respectively. In Figure 1(a) it is seen that each Voronoi cell T_I is the intersection of finitely many open half-spaces, each being delimited by the perpendicular bisector (hyperplane in $\mathbb{R}^d$). Consequently, for all nodes n_I that are inside the convex hull,¶ the Voronoi cells are closed and convex, while the cells associated with nodes on the boundary of the convex hull are unbounded (see Figure 1(b)). The nearest neighbour problem and many of its variants in computational geometry are prototypical examples that illustrate the use of Voronoi diagrams. However, the implications of Voronoi diagrams are far-reaching, with many applications in the natural sciences, physical sciences, and engineering. A detailed description of the properties and applications of Voronoi diagrams can be found in Boots[41] and Okabe *et al.*,[42] while Aurenhammer[43] presents a comprehensive review on Voronoi polygons.

The Delaunay triangulation, which is the straight-line dual of the Voronoi diagram, is constructed by connecting the nodes whose Voronoi cells have common boundaries—see Figure 1(c). The duality between the two implies that there is a Delaunay edge between two nodes in the plane if and only if their Voronoi cells share a common edge. Among all triangles, the Delaunay triangles maximize the minimum angle.[44] Another important property of Delaunay triangles is the *empty circumcircle criterion*[44]—if $\mathrm{DT}(n_J, n_K, n_L)$ is any Delaunay triangle of the nodal set N, then the circumcircle of DT contains no other nodes of N. In the context of natural neighbour interpolation, these circles are known as natural neighbour circumcircles.[20] The centre of the natural neighbour circumcircle is a vertex of the Voronoi cell. If the nodal set N is such that only three nodes lie on the circumcircle of any Delaunay triangle (non-degenerate case), then precisely three edges meet to form a Voronoi vertex. In Figure 1(d), the natural neighbour circumcircles and the associated Delaunay triangulation are shown.

From an algorithmic viewpoint, since the Voronoi diagram and the Delaunay triangulation share a common bond (duality), the combinatorial structure of either structure is completely determined from its dual. The Voronoi diagram in $\mathbb{R}^d$ is also closely related to the convex hull in $\mathbb{R}^{d+1}$,[45] which is also the basis for computing the Voronoi diagram in $\mathbb{R}^d$. The importance of the

¶The convex hull $\mathrm{CH}(N)$ of the set of nodes N is the smallest convex set containing N

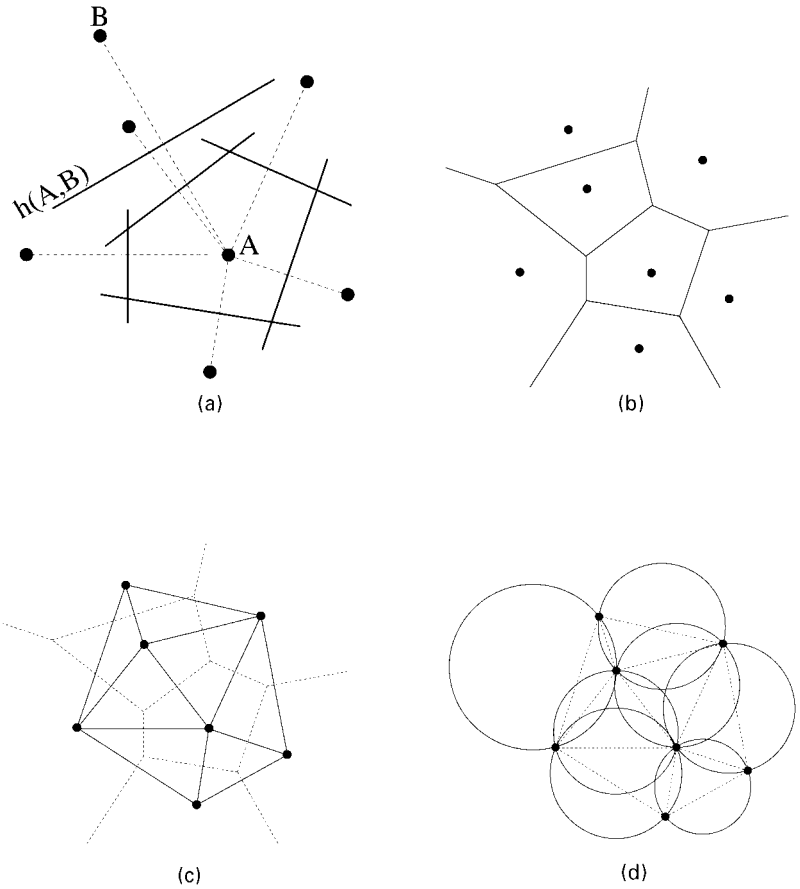


Figure 1. Geometric structures for a set N of seven nodes. (a) Voronoi cell for node A; (b) Voronoi diagram $V(N)$; (c) Delaunay triangulation DT(N); and (d) natural neighbour circumcircles

Delaunay triangulation in the context of the solution of PDEs stems from its use in mesh generation for the finite element method. Incremental insertion[44] and point insertion[46, 47] are among the early algorithms developed to compute Delaunay triangulations. Today, some of the prominent Delaunay triangulation algorithms are those based on incremental insertion,[44] divide-and-conquer,[48] and plane sweep.[49] The optimum time complexity of Delaunay triangulation algorithms is $\mathcal{O}(n \log n)$. Shewchuk[50] has carried out a comparison of the computational costs and speeds of the above three algorithms. In this paper, we use the package *Triangle*;[51] the nodal discretization and Delaunay triangles are displayed using the package *Show Me*.[52] In three-dimensions and higher, the *qhull* package[53, 54] which is based on the quickhull algorithm is considered the most versatile. A discussion on randomized algorithms to compute Voronoi diagrams and Delaunay tessellations can be found in Mulmuley.[55] For details on Delaunay triangulation and mesh generation, see Fortune[56] and the review article by Bern and Eppstein.[57]

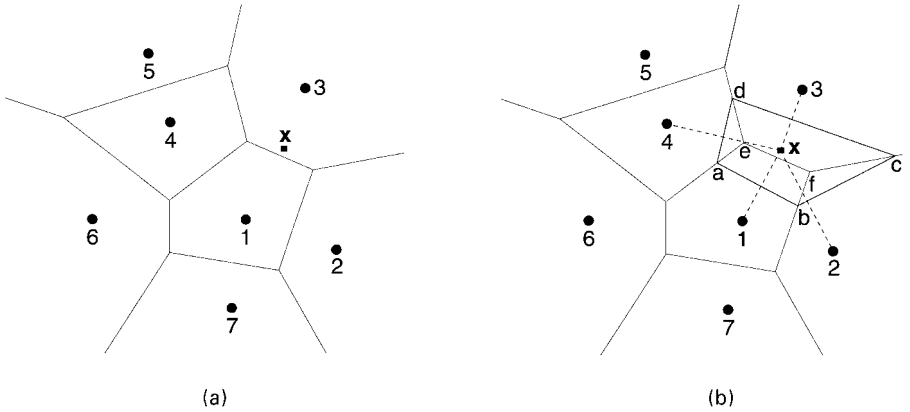


Figure 2. Construction of natural neighbour co-ordinates: (a) original Voronoi diagram and **x**; and (b) 1st-order and 2nd-order Voronoi cells about **x**

3. NATURAL NEIGHBOUR INTERPOLATION

In this section, a detailed description of the construction, properties, and numerical computations of natural neighbour shape functions is presented. Issues pertaining to the imposition of essential boundary conditions and the exact correspondence of NEM with finite elements in one-dimension are also discussed. In the sub-sections that follow, unless stated otherwise, it is implicitly assumed that the distinct set of nodes N represents a convex domain Ω.

3.1. Construction

Natural neighbour co-ordinates were introduced by Sibson[12] as a means for data interpolation and smoothing. The concept of nearest neighbours and neighbouring nodes is embedded in the first-order Voronoi diagram (Section 2). By a similar extension, one can construct higher order (k-order, $k>1$) Voronoi diagrams in the plane. Of particular interest in the context of natural neighbour co-ordinates is the case $k=2$, which is the second-order Voronoi diagram. The second-order Voronoi diagram of the set of nodes 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[12]

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

In order to quantify the neighbour relation for any point **x** introduced into the tessellation, Sibson[12] 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 to yield a new measure of 'neighbourliness' by the definition of natural neighbours. In Figure 2, a point **x** is introduced into the Voronoi diagram of the set N discussed in Section 2. If **x** is tessellated along with the set of nodes N, then the natural neighbours of **x** are those nodes which form an edge of a triangle with **x** in the new triangulation. A straight-forward means to arrive at

the same end is to use the empty circumcircle criterion (see Section 2). By doing so, we arrive at the result that if $\mathbf{x}$ lies within the circumcircle of triangle $\mathrm{DT}(n_I, n_J, n_K)$, then n_I, n_J, and n_K are its natural neighbours. In Figure 2, the perpendicular bisectors from point $\mathbf{x}$ to its natural neighbours are constructed and the Voronoi cell $T_{\mathbf{x}}$ (closed polygon *abcd*) is obtained. It is observed that $\mathbf{x}$ has four ($n=4$) natural neighbours, namely nodes 1–4.

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 $\mathbf{x}$ with respect to a natural neighbour I is defined as the ratio of the area of overlap of their Voronoi cells to the total area of the Voronoi cell of $\mathbf{x}$:

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

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 are the second-order cells, while their union (closed polygon *abcd*) is a first-order Voronoi cell. Referring to Figure 2, the shape function $\phi_1(\mathbf{x})$ is given by

$$\phi_1(\mathbf{x}) = \frac{A_{abfe}}{A_{abcd}} \tag{4}$$

The derivatives of the n–n co-ordinates are obtained by differentiating equation (3):

$$\phi_{I,j}(\mathbf{x}) = \frac{A_{I,j}(\mathbf{x}) - \phi_I(\mathbf{x}) A_{,j}(\mathbf{x})}{A(\mathbf{x})} \quad (j=1,\,2) \tag{5}$$

where equation (3) has been used to arrive at the above expression.

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

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

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 shape functions associated with each node. In the context of natural neighbour interpolation, the shape functions $\phi_I(\mathbf{x})$ are taken as the n–n co-ordinates of the point $\mathbf{x}$ in the plane. It is noteworthy to point out here that since the shape functions have compact support (Section 3.2), equation (6) is a local interpolation scheme. Hereafter, in the context of the Natural Element Method, for the purpose of clarity and in keeping with finite element usage, the expressions 'natural neighbour shape function' or simply 'shape function' are used synonymously for natural neighbour co-ordinates.

3.2. Properties

In this section, the properties of natural neighbour shape functions as well as the NEM interpolant (trial function) are presented. A succinct and elegant discussion on the properties of Sibson's interpolant can be found in Farin.[18]

Interpolation

By definition of the shape function given in equation (3), the following property is self-evident:

$$0 \leqslant \phi_I(\mathbf{x}) \leqslant 1 \tag{7}$$

Now, referring to Figure 2(b), we note that if $\mathbf{x}$ were to coincide with any node, say node 1 for instance, then it is readily seen that $\phi_1(\mathbf{x}) = 1$ and $\phi_I(\mathbf{x}) = 0$, $I \neq 1$. Therefore, the NEM and FE shape functions share the following property:

$$\phi_I(\mathbf{x}_J) = \delta_{IJ} \tag{8}$$

which implies that the NEM interpolant passes through the nodal values. A consequence of this in a Galerkin implementation is that the nodal unknowns $\mathbf{u}_I$ are the nodal displacements, which is in contrast to most meshless approximations, where the nodal parameters $\mathbf{u}_I$ are not the nodal displacements.

Partition of unity

By construction (see equation (3)), we have the following relation:

$$\sum_{I=1}^{n} \phi_I(\mathbf{x}) = 1 \quad \text{in } \Omega \tag{9}$$

where n is the number of natural neighbours, and Ω is the convex hull $\mathrm{CH}(N)$ of the set of nodes shown in Figure 1(c). Hence, by virtue of equation (7) and the above property, we note that the shape functions form a partition of unity.[9] The implication is that the interpolant constructed on the basis of these shape functions can exactly reproduce constant functions. In addition, they can also be used to enrich the interpolant by enlarging the trial function space to include additional terms that have embedded in them functions from the solution space. In Section 5.1, this is discussed at greater length in the context of enriching the NEM interpolant for crack problems.

Linear consistency

For a second-order PDE such as elastostatics, linear consistency or completeness[58] is the ability of the interpolant to exactly reproduce constant and linear displacement fields. Sibson[12] has shown (see also Reference 22) that the natural neighbour shape functions satisfy the local co-ordinate property, namely

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

which indicates that the shape functions can exactly reproduce the geometrical co-ordinates. The local co-ordinate property in conjunction with equation (9) together imply that the linear consistency conditions are satisfied.

Supports and natural neighbours

Consider a set of distinct nodes N, and let us choose a particular one, say node I, where $I \in N$. The support or domain of influence of the shape function $\phi_I(\mathbf{x})$ associated with node I is defined as the closed sub-domain Ω_{sI} such that $\phi_I(\mathbf{x}) > 0$ in Ω_{sI}, $\phi_I(\mathbf{x}) = 0$ on $\partial\Omega_{sI}$, and $\phi_I(\mathbf{x}) = 0$ in $\mathrm{CH}(N) - \Omega_{sI}$. By the circumcircle criterion, it is evident that for $\phi_I(\mathbf{x})$ to have a non-zero contribution at $\mathbf{x}$, the point $\mathbf{x}$ must lie within the circumcircle of a Delaunay triangle that has node I as one of its vertices. It immediately follows from the above argument that the support of the shape function $\phi_I(\mathbf{x})$ is the intersection of the convex hull $\mathrm{CH}(N)$ with the union of all Delaunay circumcircles that pass through node I.[18] In Figure 3(a), a unit square is discretized by 25 (5×5) equi-spaced nodes. The support for node A is illustrated in Figure 3(b)—node A is located at the centre where $\phi_A(\mathbf{x}_A)$ takes on the value of unity. The support is clearly seen to be

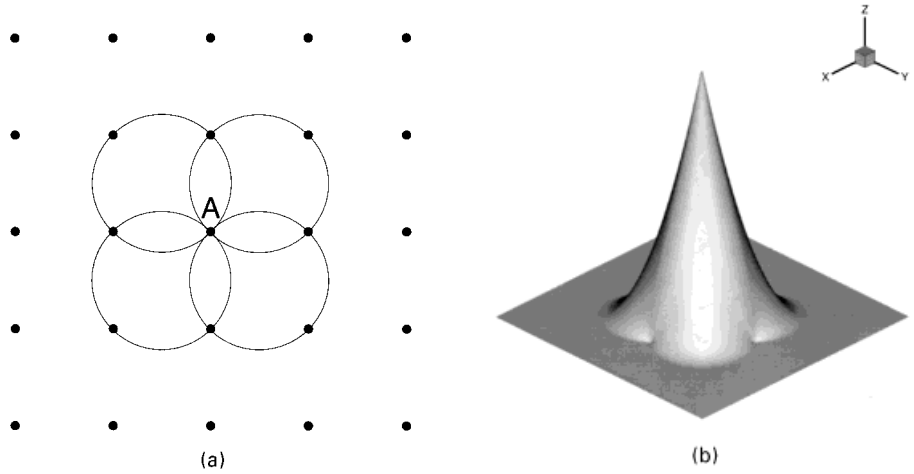


Figure 3. Support for NEM shape function: (a) nodal grid; and (b) shape function $\phi_A(\mathbf{x})$ for node A

the union of Delaunay circumcircles about node A. The surface of the shape function $\phi_A(\mathbf{x})$ is analogous to a taut rubber sheet that is stretched so as to meet the nodal data.[20]

By comparing the supports of the interpolating functions in NEM to those in some of the widely-used surface approximation schemes, much is revealed about the nature of NEM interpolation and its inherent merits. Consider Shepard's interpolant[34] or moving least-squares approximations,[35] which are both based on distance-based weights. The weight function is usually isotropic (circular in 2D and spherical in 3D), non-negative within a circle or ball of some fixed radius, and monotonically decreasing with distance from the point $\mathbf{x}$. The rationale in these approaches is that nodes that are closer to $\mathbf{x}$ are given a higher weight at $\mathbf{x}$ than those that are at farther distances. Natural neighbour interpolation assumes a totally different viewpoint. The weight at a point $\mathbf{x}$ is not dictated by the same length measure in all dimensions, but by the appropriate Lebesgue measure of the space-dimension. This allows for anisotropic supports, where the support size in direction $\mathbf{r}$ is not given by a L_2-metric but is ascertained based on a geometric construct that defines the region-of-interaction between the nodes. In meshless methods, most of which are based on distance-based weights, the handling of irregular arrangement of nodes is non-trivial since contributions at a point $\mathbf{x}$ tend to be disproportionately biased towards areas of higher nodal density. In n–n interpolation, by virtue of construction, the distribution and density of nodes are taken into account in assigning weights to the nodes at a point $\mathbf{x}$. The above geometric relationship between objects (nodes) is referred to as spatial adjacency.[59, 17] Ahuja[59] delineates the merits of using Voronoi neighbourhoods in the analysis of dot patterns in pattern recognition, while Gold[17] discusses this property in the context of surface interpolation. An instructive illustration that illuminates the above concept is shown in Figure 4.

The number of natural neighbours n is a function of position $\mathbf{x}$ as well as the nodal density. In k-dimensions, the number of natural neighbours is at least $k+1$, with an attainable lower bound.[13] In Figure 5, the variation of n within the convex hull of a uniform grid (5×5) is shown. Clearly, for a regular grid, due to degeneracy, any point has at least four (minimum) natural neighbours. Moreover, there are only two possible values for n, namely, $n=4$ or 6, with $n=6$ attained in the lens-shaped regions shown in Figure 5. In Plate 1(a), an irregular arrangement of nodes is shown,

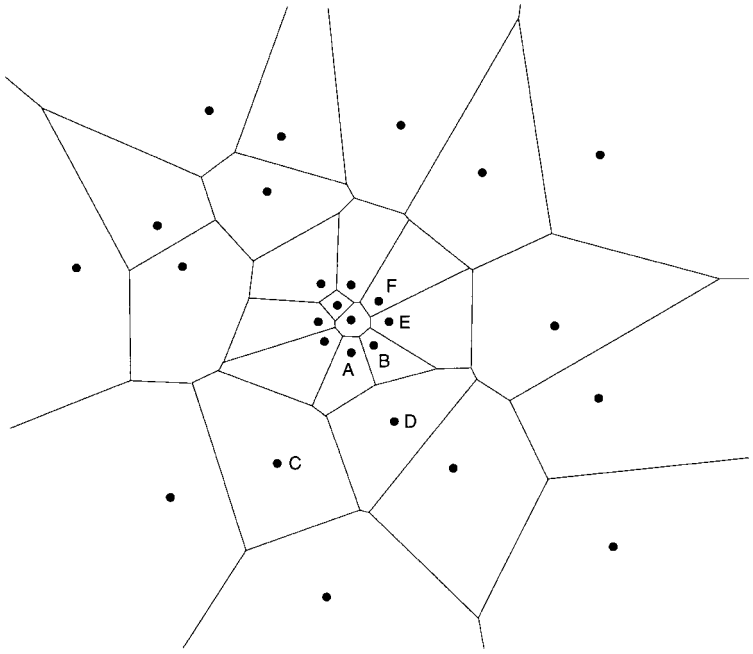


Figure 4. Voronoi neighbours. B, C, and D are neighbours of A, but E and F are not, even though $d(A,E)<d(A,D)<d(A,C)$ and $d(A,F)<d(A,D)<d(A,C)$

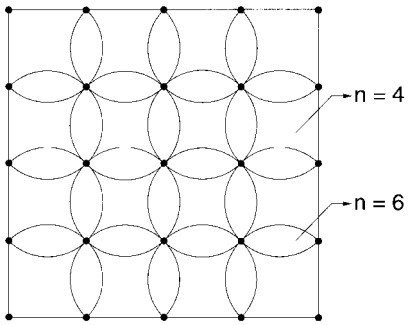


Figure 5. Variation of natural neighbours for a regular grid

and the associated variation in n is presented as a contour plot in Plate 1(b). In this case, one can see the dependence on nodal arrangement and nodal density.

Smoothness

The smoothness (or regularity) of the NEM shape functions is discussed. Natural neighbour shape functions are C^{∞} everywhere, except at the nodes where they are C^0.[12, 18, 24] Referring to Figure 2, and based on earlier observations and inferences, we see that $\phi_I(\mathbf{x})$ is a continuous function of $\mathbf{x}$. The only points of note are the nodes, but since $\phi_I(\mathbf{x}_I)=1$ as $\mathbf{x}\to\mathbf{x}_I$ from any

direction, the continuity of $\phi_I(\mathbf{x})$ is established. The differentiability of the shape function $\phi_I(\mathbf{x})$ at all points, but for the nodes, is also evident. Since $\phi_I(\mathbf{x})$ has compact support and is smoothly varying at all points except when approaching the nodes, its derivatives are C^∞ in $\Omega \backslash \mathbf{x}_I$. This fact is also seen if one considers the computational procedure for the evaluation of NEM shape functions (Section 3.3). Consider a point $\mathbf{x}$ ($\mathbf{x} \neq \mathbf{x}_J$) which lies on a natural neighbour circumcircle of node I such that $\mathbf{x} \in \partial\Omega_{sI}$. Then, only one circum-triangle contributes to the area calculations in equation (38). Since $c_j(\mathbf{x}) = \mathbf{v}$ for $j = 1$–3, the area of the sub-triangles and all its derivatives are identically zero. Hence the shape function $\phi_I(\mathbf{x})$ and all its derivatives (first-order and higher) are identically equal to zero at $\mathbf{x}$, and consequently $\phi_I(\mathbf{x}) \in C^\infty$ in $\Omega \backslash \mathbf{x}_I$. On the basis of the locus of the point $\mathbf{x}$ as it traces out a straight line, Farin[18] showed that $\phi_I(\mathbf{x})$ is only C^0 at the nodes. This fact is also seen when one considers the support of the shape functions (Figure 3). The boundary $\partial\Omega_{sI}$ is C^0 with slope discontinuities at the nodes only, which contributes to the jumps in the first derivatives of $\phi_I(\mathbf{x})$ at the nodal locations.

Smooth ($C^1(\Omega)$) forms of natural neighbour interpolation have been proposed by many authors.[13, 18, 21] Sibson[13] used a weighted least-squares fit to modify the original n–n interpolation scheme[12] to obtain a $C^1(\Omega)$ interpolant everywhere. In one-dimension, it reduces to a Hermite cubic polynomial. Farin[18] recognized Sibson's interpolant to be a building block for higher-order surface schemes, and uses Hardy's multiquadric[38] interpolation scheme as a particular example for such an application. A new $C^1(\Omega)$ scheme based on Bézier simplices is also proposed. Traversoni and Palacios[60] reformulated Sibson's interpolant in terms of Bernstein polynomials, and using the notion of covering spheres,[60] showed how it can be incorporated into spline theory. These modified forms of Sibson's interpolants which possess $C^1(\Omega)$ continuity are suitable candidates for application in Kirchhoff plate bending theory.

Interpolation in one-dimension

In one-dimension, natural neighbour interpolation as described earlier, is identical to linear finite elements. This fact is proven below.

Proof

Consider a 1-D bar of length L which is discretized by M unequally spaced nodes (Figure 6(a)). It is evident that the Voronoi vertices lie at the mid-point between any two adjacent nodes. In Figure 6(a), the dark circles represent the nodes, while the open circles are the Voronoi vertices. A consequence of the above observation is that all points in the open set $(0, L)$ have two natural neighbours, while the points on the boundary have only one natural neighbour. In order to compute the shape functions, we consider the domain (element in FEM) Ω_I between any two adjacent nodes, say n_I and n_{I+1}. Let us use a reference co-ordinate $\xi = (x - x_I)/(x_{I+1} - x_I)$, where $\xi \in [0, 1]$, and a local node numbering system: $n_I \to 1$ and $n_{I+1} \to 2$ (Figure 6(b)). Consider a point $\xi \in \Omega_I$. The second-order Voronoi cells about ξ are shown in Figure 6(b). Using equation (3), the shape functions can be written as

$$\phi_I(\xi) = \frac{L_{\xi I}}{L_{\xi 1} + L_{\xi 2}} \quad (I = 1, 2) \tag{11}$$

where $L_{\xi 1} = (1 - \xi)/2$ and $L_{\xi 2} = \xi/2$. On using these in the above equation, we obtain the result:

$$\phi_1(\xi) = 1 - \xi, \quad \phi_2(\xi) = \xi \tag{12}$$

which are precisely 1-D linear finite element shape functions. □

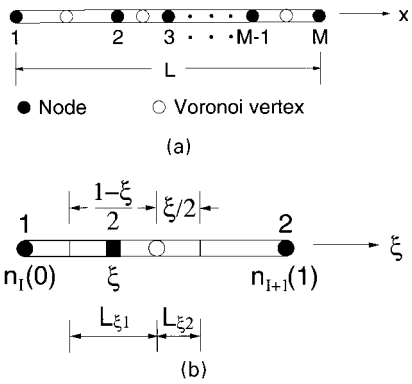


Figure 6. NEM shape functions in one dimension: (a) physical space; and (b) reference space

Interpolation in two-dimensions

The equivalence of NEM shape functions to barycentric co-ordinates and bilinear interpolation, for the special cases of $n=3$ and $n=4$ (regular grid), respectively, are shown. For irregular quadrilaterals and $n>4$, the shape functions are rational quartic functions.[18]

Three natural neighbours

If a point $\mathbf{x}$ has three natural neighbours ($n=3$), then the NEM shape functions are precisely the barycentric co-ordinates, or constant strain triangle finite element shape functions.

Proof

By an argument of uniqueness for barycentric as well as NEM shape functions ($n=3$), this equivalence is immediately seen.[18] Here we use the linear reproducing conditions to prove the correspondence. Let the natural neighbours of point $\mathbf{x}=(x, y)$ be nodes 1, 2, and 3, with co-ordinates (x_I, y_I), $I=1$–3 (Figure 7). Using equations (9) and (10), the following conditions must be met by the NEM shape functions:

$$\sum_{I=1}^{3} \phi_I(\mathbf{x}) = 1 \tag{13}$$

$$\sum_{I=1}^{3} \phi_I(\mathbf{x}) x_I = x \tag{14}$$

$$\sum_{I=1}^{3} \phi_I(\mathbf{x}) y_I = y \tag{15}$$

which in matrix form can be written as

$$\begin{bmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{bmatrix} \begin{Bmatrix} \phi_1(\mathbf{x}) \\ \phi_2(\mathbf{x}) \\ \phi_3(\mathbf{x}) \end{Bmatrix} = \begin{Bmatrix} 1 \\ x \\ y \end{Bmatrix} \tag{16}$$

The solution of the above system of linear equations is

$$\phi_1(\mathbf{x}) = \frac{D_1(\mathbf{x})}{D(\mathbf{x})}, \quad \phi_2(\mathbf{x}) = \frac{D_2(\mathbf{x})}{D(\mathbf{x})}, \quad \phi_3(\mathbf{x}) = \frac{D_3(\mathbf{x})}{D(\mathbf{x})} \tag{17}$$

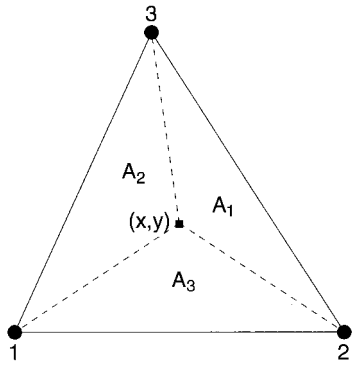


Figure 7. Barycentric co-ordinates ($n=3$)

where

$$D(\mathbf{x})=\begin{vmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{vmatrix}=\begin{vmatrix} 1 & 0 & 0 \\ x_1 & x_2-x_1 & x_3-x_1 \\ y_1 & y_2-y_1 & y_3-y_1 \end{vmatrix}=2A(\mathbf{x}) \tag{18}$$

and $D_1(\mathbf{x})=2A_1(\mathbf{x})$, $D_2(\mathbf{x})=2A_2(\mathbf{x})$, and $D_3(\mathbf{x})=2A_3(\mathbf{x})$ (Figure 7). In equation (18), $A(\mathbf{x})$ is the area of $\triangle_{123}$. Hence we can write the shape functions as

$$\phi_1(\mathbf{x})=\frac{A_1(\mathbf{x})}{A(\mathbf{x})},\quad \phi_2(\mathbf{x})=\frac{A_2(\mathbf{x})}{A(\mathbf{x})},\quad \phi_3(\mathbf{x})=\frac{A_3(\mathbf{x})}{A(\mathbf{x})} \tag{19}$$

which are precisely the barycentric co-ordinates for the point $\mathbf{x}$. □

Four natural neighbours (regular grid)

For a regular rectangular nodal grid, if a point $\mathbf{x}$ has four natural neighbours ($n=4$), then bilinear interpolation on the rectangle is obtained. A geometric proof is provided by Farin.[18] Here we use the definition of natural neighbour shape functions and explicitly carry out the computations to show the equivalence. It is to be noted that the above claim does not hold for the general case of four natural neighbours that are located at the vertices of a quadrilateral.

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 8). The first-order (dark line) and second-order Voronoi cells about $\mathbf{x}$ are shown in Figure 8. By definition of the NEM shape functions, we can write

$$\phi_I(\mathbf{x})=\frac{A_I(\mathbf{x})}{A(\mathbf{x})}\quad (I=1-4) \tag{20}$$

where $A_1(\mathbf{x})$, $A_2(\mathbf{x})$, $A_3(\mathbf{x})$, and $A_4(\mathbf{x})$ are the areas of $\triangle_{eda}$, $\triangle_{eab}$, $\triangle_{ebc}$, and $\triangle_{ecd}$, respectively. In the above equation, $A(\mathbf{x})$ is the area of the first-order Voronoi polygon *abcd* and *e* is the centre of the unit square with co-ordinates $(1/2, 1/2)$.

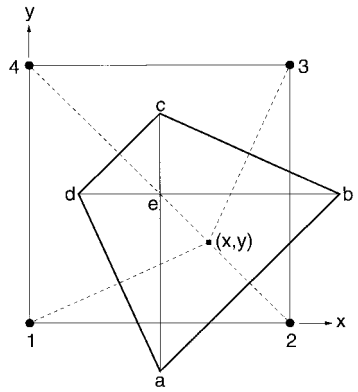


Figure 8. Bilinear interpolation on a regular grid ($n=4$)

We proceed by computing the area of the second-order Voronoi cells. To this end, by recalling the construction of 2nd-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 $\triangle_{41\mathbf{x}}$—see Section 3.3 also. Using equation (32), these co-ordinates are computed to be

$$\begin{aligned} a_1 &= \frac{1}{2}, & a_2 &= \frac{-x+x^2+y^2}{2y} \\ b_1 &= \frac{1+y-x^2-y^2}{2(1-x)}, & b_2 &= \frac{1}{2} \\ c_1 &= \frac{1}{2}, & c_2 &= \frac{1+x-x^2-y^2}{2(1-y)} \\ d_1 &= \frac{-y+x^2+y^2}{2x}, & d_2 &= \frac{1}{2} \end{aligned} \tag{21a}$$

Using the formula for the area of a triangle given in equation (30), we obtain

$$A_1(\mathbf{x}) = \frac{(x^2+y^2-x-y)^2}{8xy} \tag{22a}$$

$$A_2(\mathbf{x}) = \frac{(x^2+y^2-x-y)^2}{8y(1-x)} \tag{22b}$$

$$A_3(\mathbf{x}) = \frac{(x^2+y^2-x-y)^2}{8(1-x)(1-y)} \tag{22c}$$

$$A_4(\mathbf{x}) = \frac{(x^2+y^2-x-y)^2}{8x(1-y)} \tag{22d}$$

Since $A(\mathbf{x}) = \sum_{I=1}^{4} A_I(\mathbf{x})$, we have

$$A(\mathbf{x}) = \frac{(x^2+y^2-x-y)^2}{8xy(1-x)(1-y)} \tag{23}$$

and hence on using equations (22) and (23) in equation (20), the NEM shape functions can be written as

$$\phi_1(\mathbf{x}) = (1-x)(1-y) \tag{24a}$$

$$\phi_2(\mathbf{x}) = x(1-y) \tag{24b}$$

$$\phi_3(\mathbf{x}) = xy \tag{24c}$$

$$\phi_4(\mathbf{x}) = y(1-x) \tag{24d}$$

which are precisely bilinear FE 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 NEM interpolant. □

Interpolation in multi-dimensions

On the basis of earlier discussions and results, it is observed that natural neighbour interpolation in higher dimensions is a generalization of univariate interpolation in one dimension.[18] This appears to be a natural means of extending an interpolation scheme since the interpolant is constructed on the basis of an appropriate Lebesgue measure (volume of polytopes) in the space dimension of interest. This is in stark contrast to finite element interpolation in higher dimensions which is constructed on the basis of the product of 1-D Lagrange interpolation rules. From a mathematical and geometric viewpoint, the natural-neighbour approach is appealing in more ways than one: a sound mathematical- and geometrical-basis for construction, neighbour relationships that reflect the spatial adjacency between nodes, and a computationally feasible and attractive choice. In the remainder of this paper, we show by examples, how these properties of n–n interpolation are also of merit as a paradigm for the numerical solution of PDEs.

Linear precision on the boundary $\partial\Omega$

Issues pertaining to the imposition of essential boundary conditions in NEM for convex and non-convex bodies are addressed here. For convex domains, the discrete representation of $\partial\Omega$ is the boundary of the convex hull CH(N) for a set of distinct nodes N, while for a non-convex domain, the discrete representation of the continuous boundary $\partial\Omega$ is through a Planar Straight Line Graph (PSLG).[‖]

Convex domains

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 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 9(a)). For simplicity and for ease of illustration, we assume that ξ has only three natural neighbours, namely nodes 1–3. 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 1st-order and 2nd-order Voronoi cells about ξ are shown

[‖] A PSLG is a collection of nodes and edges, whose presence is preserved in the partitioning of the domain

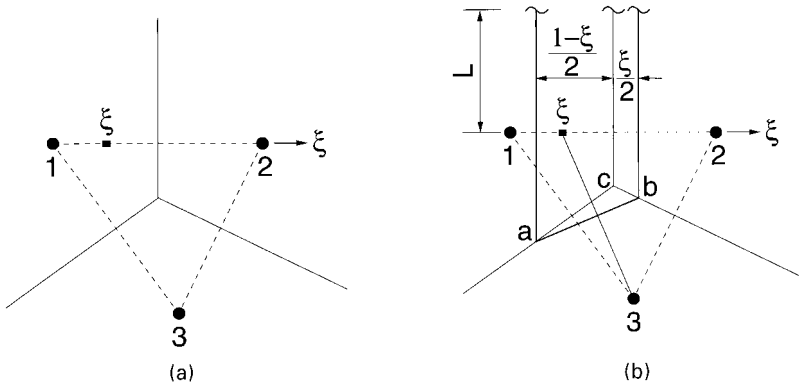


Figure 9. Linear behaviour of $\mathbf{u}^h(\xi)$ along the edge of a boundary Delaunay triangle (convex domain): (a) original Voronoi cells ($n=3$) and ξ; and (b) 1st-order and 2nd-order Voronoi cells about ξ

in Figure 9(b). By definition, the shape functions can be written as

$$\phi_I(\xi) = \frac{A_I(\xi)}{A(\xi)} \quad (I = 1,2,3) \tag{25}$$

where $A(\xi) = \sum_{J=1}^{3} A_J(\xi)$. Since the Voronoi cell for each node on the boundary of the convex hull is unbounded, we can express A_1, A_2, and A_3 as

$$A_1(\xi) = \lim_{L\to\infty} L\xi_1 + \delta_1, \quad A_2(\xi) = \lim_{L\to\infty} L\xi_2 + \delta_2, \quad A_3(\xi) = \delta_3 \tag{26}$$

where $\xi_1 = (1-\xi)/2$, $\xi_2 = \xi/2$ and δ_1, δ_2, and δ_3 are finite. For instance, δ_3 is given by the area of $\triangle_{abc}$ in Figure 9(b). On using equation (25), we can write

$$\phi_1(\xi) = \lim_{L\to\infty} \frac{L(1-\xi) + 2\delta_1}{L + 2\delta_1 + 2\delta_2 + 2\delta_3} \tag{27a}$$

$$\phi_2(\xi) = \lim_{L\to\infty} \frac{L\xi + 2\delta_2}{L + 2\delta_1 + 2\delta_2 + 2\delta_3} \tag{27b}$$

$$\phi_3(\xi) = \lim_{L\to\infty} \frac{2\delta_3}{L + 2\delta_1 + 2\delta_2 + 2\delta_3} \tag{27c}$$

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

$$\phi_1(\xi) = 1 - \xi, \quad \phi_2(\xi) = \xi, \quad \phi_3(\xi) = 0, \tag{28}$$

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 2nd-order Voronoi cell about ξ for all interior nodes are closed polygons and hence the overlapping areas are finite, similar to $A_3(\xi)$ in Figure 9(b). Using the above equation, the trial functions at the point ξ can be written as

$$\mathbf{u}^h(\xi) = (1-\xi)\mathbf{u}_1 + \xi\mathbf{u}_2 \tag{29}$$

which are linear functions, and hence the proof. □

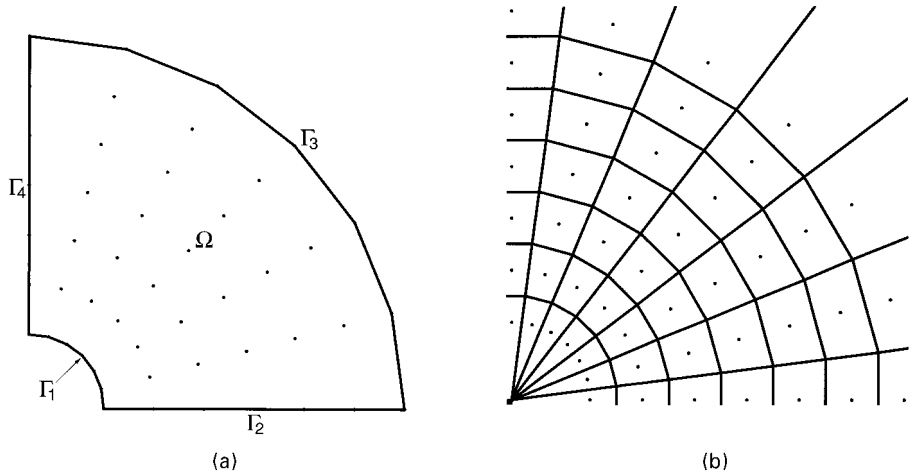


Figure 10. Linear behaviour of $\mathbf{u}^h(\mathbf{x})$ along the boundary of a non-convex domain: (a) nodal discretization, and (b) Voronoi diagram

Non-convex domains

The discrete model consists of a set of nodes N that describes a non-convex domain Ω. Consider a Γ_1 ($\Gamma_1 \subset \partial\Omega$) 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 is shown in Figure 10. It is evident that the Voronoi cells for the nodes along Γ_i ($i = 2-4$) are unbounded, while the Voronoi cells for the nodes along Γ_1 are bounded and therefore have finite areas (Figure 10(b)). The proof outlined for the case of convex domains holds for $\Gamma = \Gamma_2 \cup \Gamma_3 \cup \Gamma_4$ but fails for Γ_1. Therefore, the approximation is not strictly linear between adjacent nodes on the boundary Γ_1. For a point $\mathbf{x}$ located on Γ_1, there exists non-zero contributions from some interior nodes. However, numerical simulations indicate that with adequate discretization of the boundary Γ_1, the shape function contribution of interior nodes are two orders or less in magnitude in comparison to the boundary node values. Consider the nodal grid shown in Figure 10 which consists of 49 nodes with seven nodes along Γ_1. For a point located on Γ_1, the shape function contributions of interior nodes are about 2 per cent of those due to the boundary nodes on Γ_1. Further discretization of the boundary Γ_1 would lead to better accuracy in the imposition of the essential boundary conditions. In general, for non-convex domains, the linearity of the trial function along Γ_1 depends on the nodal discretization as well as the local radius of curvature of Γ_1.

A consequence of the above discussion is that essential boundary conditions in the NEM can be directly imposed on the nodes, in accordance with the finite element method—this is due to the interpolating property of the NEM shape functions and the linearity of the trial functions along the edge of a boundary Delaunay triangle. It is to be noted that the above inference is rigorously true for convex domains; however, for non-convex domains, numerical simulations needs to be carried out on a per problem basis to determine the appropriate nodal discretization of Γ_1 which is required to yield accurate results.

3.3. Numerical computational procedure for NEM shape functions

In this work, Watson's algorithm[23] is used to compute the natural neighbour shape functions. The algorithmic details and computational procedure is described in Watson[23] as well as Braun *et al.*[27] First, we provide well-known geometric formulas for the area, circumcentre, and circumradius of a triangle, and then go on to present the computational implementation of the shape function algorithm.

Area of a triangle

Consider a triangle $t(A,B,C)$ with vertices $A(\mathbf{a})$, $B(\mathbf{b})$, and $C(\mathbf{c})$, where $\mathbf{a}=(a_1,a_2)$, $\mathbf{b}=(b_1,b_2)$, and $\mathbf{c}=(c_1,c_2)$. The signed area of $\triangle_{ABC}$ is given by

$$A=\frac{(a_1-c_1)(b_2-c_2)-(b_1-c_1)(a_2-c_2)}{2} \tag{30}$$

If $\mathbf{a}\equiv\mathbf{a}(\mathbf{x})=(a_1(\mathbf{x}),a_2(\mathbf{x}))$, $\mathbf{b}\equiv\mathbf{b}(\mathbf{x})=(b_1(\mathbf{x}),b_2(\mathbf{x}))$, and c_1 and c_2 are independent of $\mathbf{x}$, then the derivatives of A can be written as

$$\begin{aligned} A_{,i}(\mathbf{x})=&\frac{(a_1(\mathbf{x})-c_1)b_{2,i}(\mathbf{x})+(b_2(\mathbf{x})-c_2)a_{1,i}(\mathbf{x})}{2}\\ &-\frac{(b_1(\mathbf{x})-c_1)a_{2,i}(\mathbf{x})+(a_2(\mathbf{x})-c_2)b_{1,i}(\mathbf{x})}{2} \end{aligned} \tag{31}$$

where $i=1,2$, and a comma denotes differentiation with respect to the appropriate spatial co-ordinate.

Circumcentre and circumradius of a triangle

Consider a triangle $t(A,B,C)$ with vertices $A(\mathbf{a})$, $B(\mathbf{b})$, and $C(\mathbf{c})$, where $\mathbf{a}=(a_1,a_2)$, $\mathbf{b}=(b_1,b_2)$, and $\mathbf{c}=(c_1,c_2)$ (Figure 11). The circumcircle of $\triangle_{ABC}$ is the circle that passes through the three vertices of the triangle. Its centre is the circumcenter $v(v_1,v_2)$ and radius R is the circumradius. The co-ordinates of the circumcentre are the perpendicular bisectors of the edges of the triangle. Its co-ordinates are found by either: (a) considering the equation of a plane though three points, where each point $(x_i, y_i, x_i^2+y_i^2)$ is the map of $\mathbf{x}$ onto $(\mathbf{x},\|\mathbf{x}\|^2)$, or (b) solving for the intersection point of the perpendicular bisector of any two edges of the triangle. We use the latter approach to obtain the co-ordinates of the circumcentre. By considering the edges AC and BC in Figure 11, a system of two linear equations is obtained. On carrying out the solution and after some algebraic manipulations, we obtain the following expressions for the circumcentre of $\triangle_{ABC}$:

$$v_1=\frac{(a_1^2-c_1^2+a_2^2-c_2^2)(b_2-c_2)-(b_1^2-c_1^2+b_2^2-c_2^2)(a_2-c_2)}{D} \tag{32a}$$

$$v_2=\frac{(b_1^2-c_1^2+b_2^2-c_2^2)(a_1-c_1)-(a_1^2-c_1^2+a_2^2-c_2^2)(b_1-c_1)}{D} \tag{32b}$$

where D which is four times the area of $\triangle_{ABC}$ (refer to equation (30)), is given by

$$D=2[(a_1-c_1)(b_2-c_2)-(b_1-c_1)(a_2-c_2)] \tag{32c}$$

From the viewpoint of computational efficiency (reduced multiplication operations), the squared difference of two terms in the above equations is represented as the product of their sums and

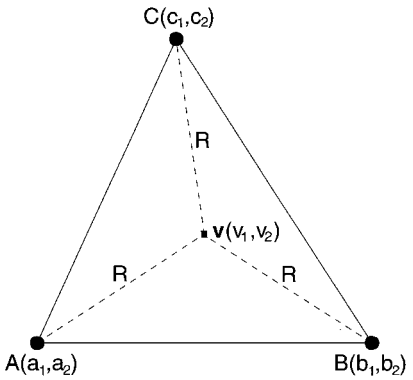


Figure 11. Circumcentre and circumradius of a triangle

differences in the numerical implementation. For instance, $a_1^2 - c_1^2$ in equation (32a) is computed as $(a_1 - c_1)(a_1 + c_1)$, etc.

If the vertex $C(\mathbf{c})$ coincides with the point $\mathbf{x}$, where $\mathbf{x} = (x_1, x_2)$, then $c_1 = x_1$ and $c_2 = x_2$. Assume the co-ordinates a_1, a_2, b_1, and b_2 are independent of x_1 and x_2. The derivatives of the co-ordinates of the circumcentre can then be written in the following simplified form:

$$v_{1,1}(\mathbf{x}) = \frac{(x_1 - v_1(\mathbf{x}))D_{,1}(\mathbf{x})}{D(\mathbf{x})} \tag{33a}$$

$$v_{1,2}(\mathbf{x}) = \frac{(\alpha + x_2 D_{,1}(\mathbf{x}) - v_1(\mathbf{x})D_{,2}(\mathbf{x}))}{D(\mathbf{x})} \tag{33b}$$

$$v_{2,1}(\mathbf{x}) = \frac{(-\alpha + x_1 D_{,2}(\mathbf{x}) - v_2(\mathbf{x})D_{,1}(\mathbf{x}))}{D(\mathbf{x})} \tag{33c}$$

$$v_{2,2}(\mathbf{x}) = \frac{(x_2 - v_2(\mathbf{x}))D_{,2}(\mathbf{x})}{D(\mathbf{x})} \tag{33d}$$

where $v_1(\mathbf{x})$, $v_2(\mathbf{x})$, and $D(\mathbf{x})$ are given in equation (32) and

$$\alpha = (b_1 + a_1)(b_1 - a_1) + (b_2 + a_2)(b_2 - a_2) \tag{34a}$$

$$D_{,1}(\mathbf{x}) = 2(a_2 - b_2) \tag{34b}$$

and

$$D_{,2}(\mathbf{x}) = 2(b_1 - a_1) \tag{34c}$$

By definition, the circumradius is the distance from the circumcentre to a vertex of $\triangle_{ABC}$. Hence the square of the circumradius can be written as

$$R^2(\mathbf{x}) = (a_1 - v_1(\mathbf{x}))^2 + (a_2 - v_2(\mathbf{x}))^2 \tag{35}$$

Shape function computations

The algorithm proposed by Watson[23] is used to compute the shape functions. In order to present the algorithm and its numerical implementation, we consider Figure 2(b), where the construction

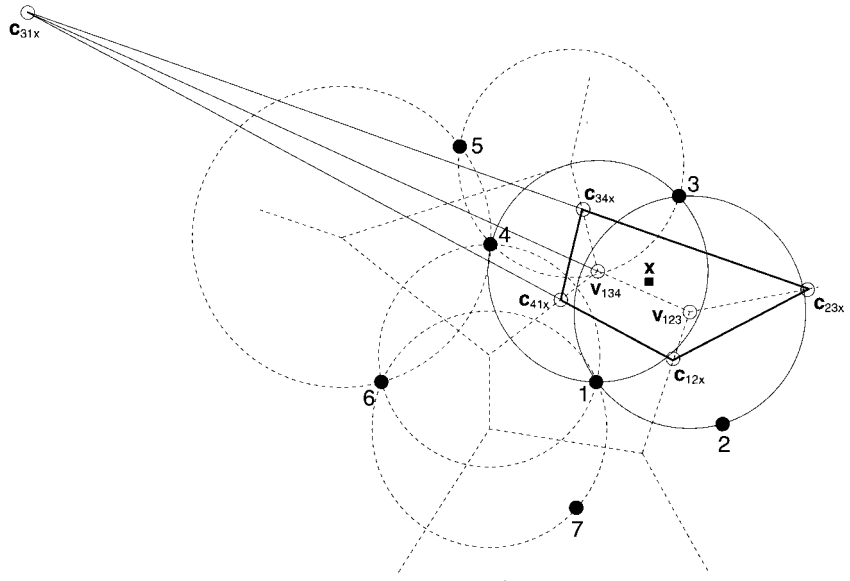


Figure 12. Computation of natural neighbour shape functions

of natural neighbour shape functions is illustrated. In Figure 12, the original nodal set, the Voronoi diagram (dashed lines), and the natural neighbour circumcircles are shown. The point **x** is introduced into the set, and the natural neighbours of **x** as well as the values of the natural neighbour shape functions and their derivatives at **x** are to be computed.

The computational methodology for the shape function evaluation is based on the fact that the second-order Voronoi cells about **x** (Figure 2(b)) can be computed as a summation of signed area of triangles. In the implementation of the NEM, a preprocessing step is the computation of the circumcentre and square of the circumradius of the natural neighbour (Delaunay) circumcircles. It is assumed that the nodal connectivity for each Delaunay triangle is stored in counter-clockwise orientation, and consequently the appropriate sign is automatically chosen when the area of a triangle is computed. The formulas for the area, circumcentre, and circumradius of a triangle are presented in Section 3.3. In the assembly-step, prior to the shape function computation, the number of natural neighbours n and their global nodal numbers along with the number of triangles t_n associated with the natural neighbours as well as their global triangle numbers, are computed. A simple means to determine if a node is a natural neighbour of a point **x** is to use the empty circumcircle criterion indicated in Section 2—if the square of the Euclidean distance from **x** to the centre of a natural neighbour circumcircle (associated with nodes n_I, n_J, and n_K) is less than the square of the radius of the circumcircle, then nodes n_I, n_J, and n_K are natural neighbours of the point **x**:

$$\|\mathbf{v}-\mathbf{x}\|^2 < R^2 \tag{36}$$

where $\mathbf{v}=(v_1, v_2)$ is the centre of the natural neighbour circumcircle and R, which is given in equation (35), is its radius. Braun and Sambridge[2] used Lawson's algorithm[44] as part of the neighbour-search, which is a computationally sound choice if interpolation of a very large nodal

set is considered. This is so, since otherwise the condition given in equation (36) needs to be checked for all the Delaunay triangles. In this work, since relatively few nodes ($M < 2500$) are considered in the nodal discretization of a domain, the above condition is checked for all the Delaunay triangles.

Let the 3-tuple $(\mathbf{p}, \mathbf{q}, \mathbf{r})$ represent a triangle with vertices $\mathbf{p} = (p_1, p_2)$, $\mathbf{q} = (q_1, q_2)$, and $\mathbf{r} = (r_1, r_2)$. Now, for each Delaunay triangle t with circumcentre $\mathbf{v} = (v_1, v_2)$ that is selected on the basis of equation (36), we form a new set of three triangles $\{t_1, t_2, t_3\}$, where each has $\mathbf{x}$ and two of the vertices of the Delaunay triangle t as its vertices. Consider a local numbering system ($i = 1, 2, 3$) for the vertices $\mathbf{a}_i$ of the new triangles t_i, and associate each local node i with its opposite edge (j–k). We can write the circumcentre of the new triangles so constructed and their derivatives in the following compact form:

$$\mathbf{c}_i(\mathbf{x}) = \Theta(\mathbf{a}_j, \mathbf{a}_k, \mathbf{x}) \tag{37a}$$

$$\mathbf{c}_{i,m}(\mathbf{x}) = \Theta_{,m}(\mathbf{a}_j, \mathbf{a}_k, \mathbf{x}) \tag{37b}$$

where $m = 1, 2$ and i, j, and k assume cyclic counter-clockwise permutations of 123. In the above equations, Θ and $\Theta_{,m}$ are functional representations for the circumcentre of a triangle and its derivatives which are given in equations (32) and (33), respectively. To elaborate, $\mathbf{c}_1$ refers to the circumcentre of the triangle with vertices $(2, 3, \mathbf{x})$, $\mathbf{c}_2$ that for the triangle with vertices $(3, 1, \mathbf{x})$, and $\mathbf{c}_3$ is the circumcentre of the triangle with vertices $(1, 2, \mathbf{x})$.

The next step involves the construction of sub-triangles using combinations from the collection $\{\mathbf{c}_1(\mathbf{x}), \mathbf{c}_2(\mathbf{x}), \mathbf{c}_3(\mathbf{x}), \mathbf{v}\}$, such that $\mathbf{v}$ appears in all the sub-triangles. Here too cyclic permutation of 123 is maintained. Clearly, three such sub-triangles can be constructed, namely $(\mathbf{c}_1(\mathbf{x}), \mathbf{c}_2(\mathbf{x}), \mathbf{v})$ [node 3], $(\mathbf{c}_2(\mathbf{x}), \mathbf{c}_3(\mathbf{x}), \mathbf{v})$ [node 1], and $(\mathbf{c}_3(\mathbf{x}), \mathbf{c}_1(\mathbf{x}), \mathbf{v})$ [node 2], where the nodal association is indicated within the square bracket. Now, the area of these sub-triangles and their derivatives can be written as

$$\alpha_{it}(\mathbf{x}) = \Psi(\mathbf{c}_j(\mathbf{x}), \mathbf{c}_k(\mathbf{x}), \mathbf{v}) \tag{38a}$$

$$\alpha_{it,m}(\mathbf{x}) = \Psi_{,m}(\mathbf{c}_j(\mathbf{x}), \mathbf{c}_k(\mathbf{x}), \mathbf{v}) \tag{38b}$$

where $m = 1, 2$ and i, j, and k assume cyclic counter-clockwise permutations of 123. In the above equations, Ψ and $\Psi_{,m}$ are functional representations for the area of a triangle and its derivatives which are given in equations (30) and (31), respectively. By the local $\rightarrow$ global nodal associativity, the area contribution to each global node and its update is made through the following conceptual symbolic assignments:

$$\beta_I(\mathbf{x}) \leftarrow \beta_I(\mathbf{x}) + \alpha_{it}(\mathbf{x}) \tag{39a}$$

$$\beta_{I,m}(\mathbf{x}) \leftarrow \beta_{I,m}(\mathbf{x}) + \alpha_{it,m}(\mathbf{x}) \tag{39b}$$

where $\beta_I(\mathbf{x})$ and $\beta_{I,m}(\mathbf{x})$ are set to zero for all natural neighbours prior to the shape function computations. The above procedure is carried out for all the Delaunay circum-triangles t ($t = 1, 2, \ldots, t_n$). Hence the area of the second-order Voronoi cell and its derivatives are now given by

$$A_I(\mathbf{x}) = \beta_I(\mathbf{x}) \tag{40a}$$

$$A_{I,m}(\mathbf{x}) = \beta_{I,m}(\mathbf{x}) \tag{40b}$$

Table I. Pseudo-code for natural neighbour shape function computations at a point $\mathbf{x}$

1. Determine natural neighbours I $(I=1\rightarrow n)$ and Delaunay circum-triangles t $(t=1\rightarrow t_n)$ for $\mathbf{x}$—see equation (32);
2. Initialize $\phi_I = d\phi_{I1} = d\phi_{I2} = 0\ \forall I$; $A = dA_1 = dA_2 = 0$;
3. for $t=1\rightarrow t_n$ {
   - set $\mathbf{v}$ = circumcentre of $\triangle_t$—see equation (32);
   - for $i=1\rightarrow 3$ {
     - set j and k using cyclic order;
     - compute $\mathbf{c}_i$ and $\mathbf{c}_{i,m}$ based on equation (37) and store;
     }
   - for $i=1\rightarrow 3$ {
     - set g = global node number of local node i;
     - set I = position of g in the natural neighbour nodal array $(1\leqslant I\leqslant n)$;
     - compute α_{it} and $\alpha_{it,m}$ based on equation (38);
     - set $\alpha=\alpha_{it}$, $d\alpha_1=\alpha_{it,1}$, $d\alpha_2=\alpha_{it,2}$;
     - update $\phi_I \leftarrow \phi_I+\alpha$, $d\phi_{I1}\leftarrow d\phi_{I1}+d\alpha_1$, $d\phi_{I2}\leftarrow d\phi_{I2}+d\alpha_2$;
     - update $A\leftarrow A+\alpha$, $dA_1\leftarrow dA_1+d\alpha_1$, $dA_2\leftarrow dA_2+d\alpha_2$;
     }
   }
4. for $I=1\rightarrow n$ {
   - set $\phi_I \leftarrow \dfrac{\phi_I}{A}$, $d\phi_{I1}\leftarrow \dfrac{d\phi_{I1}-\phi_I dA_1}{A}$, $d\phi_{I2}\leftarrow \dfrac{d\phi_{I2}-\phi_I dA_2}{A}$;
   }

for $m=1,2$ and $I=1,2,\ldots,n$. Let

$$A(\mathbf{x}) = \sum_{I=1}^{n} A_I(\mathbf{x}) \tag{41a}$$

$$A_{,m}(\mathbf{x}) = \sum_{I=1}^{n} A_{I,m}(\mathbf{x}) \tag{41b}$$

We reproduce equations (3) and (5) below for the natural neighbour shape functions and their derivatives:

$$\phi_I(\mathbf{x}) = \frac{A_I(\mathbf{x})}{A(\mathbf{x})} \tag{42a}$$

$$\phi_{I,m}(\mathbf{x}) = \frac{A_{I,m}(\mathbf{x}) - \phi_I(\mathbf{x})A_{,m}(\mathbf{x})}{A(\mathbf{x})} \tag{42b}$$

Substituting for $A_I(\mathbf{x})$, $A_{I,m}(\mathbf{x})$, $A(\mathbf{x})$, and $A_{,m}(\mathbf{x})$ from equations (40a), (40b), (41a), and (41b), respectively, in the above equations, the shape function $\phi_I(\mathbf{x})$ and its derivatives $\phi_{I,m}(\mathbf{x})$ are computed for all the natural neighbours. A pseudo-code for the shape function computations is presented in Table I.

We use Figure 12 to illustrate the application of the shape function computations. In order to keep the notations to a minimum, we let the position vector of the circumcentre of a triangle also serve as a label for the point. It is seen in Figure 12 that the point $\mathbf{x}$ lies within only two circles (dark lines), namely $\mathbf{v}_{123}$ and $\mathbf{v}_{134}$ which are the position vectors for the circumcentre (open circle) of the natural neighbour circumcircles. Hence the natural neighbours of $\mathbf{x}$ are nodes 1, 2, 3, and 4, with the associated circumcircles $\mathbf{v}_{123}$ and $\mathbf{v}_{134}$ that correspond to $\triangle_{123}$ and $\triangle_{134}$, respectively. Let

us first consider $\triangle_{123}$ and carry out the steps indicated in the shape function computations. We form new triangles, namely the triplets $(2,3,\mathbf{x})$, $(3,1,\mathbf{x})$ and $(1,2,\mathbf{x})$ with circumcentres (open circles) $\mathbf{c}_{23\mathbf{x}}$, $\mathbf{c}_{31\mathbf{x}}$, and $\mathbf{c}_{12\mathbf{x}}$, respectively. Now, from the set $\{\mathbf{c}_{23\mathbf{x}}, \mathbf{c}_{31\mathbf{x}}, \mathbf{c}_{12\mathbf{x}}, \mathbf{v}_{123}\}$, we can construct three sub-triangles $(\mathbf{c}_{23\mathbf{x}}, \mathbf{c}_{31\mathbf{x}}, \mathbf{v}_{123})$, $(\mathbf{c}_{31\mathbf{x}}, \mathbf{c}_{12\mathbf{x}}, \mathbf{v}_{123})$, and $(\mathbf{c}_{12\mathbf{x}}, \mathbf{c}_{23\mathbf{x}}, \mathbf{v}_{123})$. The area of the above three sub-triangles is indicated by α_{it} $(i = 1\text{–}3)$ in equation (38). By following a similar procedure for $\triangle_{134}$, we obtain the following three sub-triangles: $(\mathbf{c}_{34\mathbf{x}}, \mathbf{c}_{41\mathbf{x}}, \mathbf{v}_{134})$, $(\mathbf{c}_{41\mathbf{x}}, \mathbf{c}_{13\mathbf{x}}, \mathbf{v}_{134})$, and $(\mathbf{c}_{31\mathbf{x}}, \mathbf{c}_{34\mathbf{x}}, \mathbf{v}_{134})$. We note that the triplets $(\mathbf{c}_{41\mathbf{x}}, \mathbf{c}_{31\mathbf{x}}, \mathbf{v}_{134})$ and $(\mathbf{c}_{13\mathbf{x}}, \mathbf{c}_{34\mathbf{x}}, \mathbf{v}_{134})$ are oriented in the counter-clockwise direction, and hence contribute as negative areas. In addition, by noting the local $\rightarrow$ global nodal associations, we obtain the following results:

$$A_1(\mathbf{x}) = \Psi(\mathbf{c}_{31\mathbf{x}}, \mathbf{c}_{12\mathbf{x}}, \mathbf{v}_{123}) + \Psi(\mathbf{c}_{41\mathbf{x}}, \mathbf{c}_{31\mathbf{x}}, \mathbf{v}_{134}) \equiv \text{Area}\,(\square_{12\mathbf{x}-123-134-41\mathbf{x}}) \tag{43a}$$

$$A_2(\mathbf{x}) = \Psi(\mathbf{c}_{12\mathbf{x}}, \mathbf{c}_{23\mathbf{x}}, \mathbf{v}_{123}) \equiv \text{Area}\,(\triangle_{12\mathbf{x}-23\mathbf{x}-123}) \tag{43b}$$

$$A_3(\mathbf{x}) = \Psi(\mathbf{c}_{23\mathbf{x}}, \mathbf{c}_{31\mathbf{x}}, \mathbf{v}_{123}) + \Psi(\mathbf{c}_{13\mathbf{x}}, \mathbf{c}_{34\mathbf{x}}, \mathbf{v}_{134}) \equiv \text{Area}\,(\square_{134-123-23\mathbf{x}-34\mathbf{x}}) \tag{43c}$$

$$A_4(\mathbf{x}) = \Psi(\mathbf{c}_{34\mathbf{x}}, \mathbf{c}_{41\mathbf{x}}, \mathbf{v}_{134}) \equiv \text{Area}\,(\triangle_{34\mathbf{x}-41\mathbf{x}-134}) \tag{43d}$$

$$A(\mathbf{x}) = \sum_{I=1}^{4} A_I(\mathbf{x}) \tag{43e}$$

$$\phi_I(\mathbf{x}) = \frac{A_I(\mathbf{x})}{A(\mathbf{x})} \quad (I = 1, 2, 3, 4) \tag{43f}$$

It is immediately seen that the areas computed for each node in Figure 12 are precisely those seen in Figures 2(b) and 12. As indicated in the above equation, the shape functions for each node are just the areas normalized with respect to the total area.

Watson's algorithm[23] for the computation of NEM shape functions is foolproof for any point that does not lie along an edge of a Delaunay triangle, since the circumcenters of all triangles that need to be constructed are unique. It does, however, fail if a point $\mathbf{x}$ lies along the edge of a Delaunay triangle. This is so, since if the point $\mathbf{x}$ lies along an edge, say A–B of a triangle (A,B,C), then the new triangle $(A,B,\mathbf{x})$ formed in the algorithm has a non-unique circumcentre $(R(\mathbf{x}) = \pm\infty)$. In the numerical integration of the weak form, since all points are interior to the triangle, this situation never arises.

There are alternative means to carry out the area computations. A straight-forward approach is to compute the area of the polygons (2nd-order Voronoi cells) directly. Since the vertices of the polygon are the circumcentre of triangles formed from the natural neighbour nodal set and the point $\mathbf{x}$, one can use the algorithm mentioned in O'Rourke[33] to compute the area of the polygon. An additional step involved in this is that the vertices of the polygon need to be sorted in counter-clockwise orientation. Lasserre's algorithm[61] for volume of convex polytopes in R^d is based on solving a constraint (bounded hyperplanes) linear programming problem to compute the volume of the polytopes. Braun and Sambridge[2] used this algorithm to compute the natural neighbour shape function in their PDE application. This approach is robust, and is applicable at any point in the domain for shape function computations.

Computational costs

The computational costs incurred in the implementation and execution of any numerical method is a critical component in the evaluation of its feasibility, usability, and potential for mid- to large-scale applications. Finite elements, apart from their nice local (polynomial) properties, are a

computationally attractive choice because of the very fast execution times that are attainable. The key differences that demarcate NEM from FEM is in the construction and numerical computation of the shape functions, and in the methodology used to assemble the stiffness matrix **K**. All other steps are common to both, and hence we focus our attention on just the above two differences. In the finite element method, all shape function computations involve basic linear algebraic calculations. In NEM, the key time-consuming steps are:

1. Dirichlet tessellation (Voronoi diagram) of the nodes. As indicated in Section 2, the optimum time-complexity of triangulation algorithms in 2-D is $\mathcal{O}(n \log n)$. The Delaunay triangulation of up to a million nodes can be carried out in less than one minute, and hence in two dimensions, this step is not burdensome.
2. The search for natural neighbours for a sampling point **x**. If the total number of triangles is only in the few thousands, searching through all the triangles, which is a $\mathcal{O}(n^2)$ algorithm, is not prohibitively expensive and is a feasible approach. If the number of triangles is of $\mathcal{O}(10^5)$ or greater, the walking-triangle algorithm due to Lawson[44] which is a $\mathcal{O}(n)$ algorithm, is a suitable choice.
3. The numerical computations of the NEM shape functions is outlined in Section 3.3. All the steps involved in the algorithm are purely algebraic in nature, with no matrix or vector computations.
4. The assembly of the stiffness matrix **K** in NEM is carried out on a nodal basis. In finite elements, an element stiffness matrix structure which is common to all elements is used in the assembly of the global stiffness matrix **K**. In NEM, since the number of neighbours for a point **x** is fairly small ($n_{\max} = 11$ for the irregular grid in Plate 1), the nodal-based assembly is not overly time intensive.

On the basis of the numerical results that are presented in Section 7, the shape function timings on a HP9000/s700 workstation are about a factor of 1–2 slower than that for constant strain finite elements. In two dimensions, since the solution phase dominates for a problem with significant number of degrees of freedom (10^4 or higher), the overall timing comparison of NEM to FEM is even smaller.

4. TREATMENT OF MATERIAL DISCONTINUITIES

The treatment of material discontinuities in a two-dimensional setting using NEM is presented. Some typical examples of material discontinuities are encountered in composite materials, phase transformations, and inclusions in a matrix with different material properties. A nice exposition on the underlying variational formulation and finite element computations for this problem can be found in Mackinnon and Carey.[62]

Consider a two-dimensional body Ω, such that $\Omega = \Omega_1 \cup \Omega_2$, where Ω_1 and Ω_2 have different material properties. The interface between the two regions is Γ (Figure 13). Since kinematically admissible displacement fields must be C^0, it is evident that displacement continuity must be met along the interface: $\mathbf{u}^1 = \mathbf{u}^2$ on Γ. An additional condition arises out of the variational principle—apart from the Euler–Lagrange equations, the natural interface condition is to be satisfied (in a weak sense), namely $\mathbf{t}^1 = \mathbf{t}^2$ along Γ. The above two conditions are automatically satisfied by the finite element interpolant if element boundaries are coincident with the discontinuity interface Γ. This is so, since the displacements are C^0, which allows for displacement gradients to be discontinuous in the normal direction ($\mathbf{n}$) to Γ.

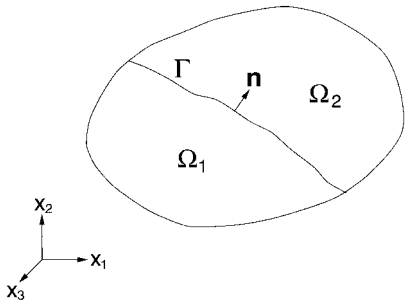


Figure 13. Bimaterial configuration ($\Omega = \Omega_1 \cup \Omega_2$)

The modelling of material discontinuities in the context of NEM follows. If the interface Γ is straight, then the description on convex domains in Section 3.2 holds, and the displacement field **u** along Γ is strictly linear between adjacent nodes. Therefore, for this case, the interface displacement and traction continuity conditions are met. Of course, here, unlike FEM, one must explicitly disallow points in Ω_1 to be influence by nodes in Ω_2 and vice versa. A line-of-sight or visibility criterion,[11] similar to that used by Cordes and Moran[63] is invoked to meet the above requirement. Now, let us consider the case where either Ω_1 or Ω_2 is a non-convex domain. A plate consisting of a quarter-circle is considered. The circular domain Ω_1 is convex, while Ω_2 is non-convex. The circular arc is the material interface Γ that demarcates the two regions. The displacements along Γ are linear when approaching the interface from Ω_1, but are not so when approaching from Ω_2. In Section 3.2, the linearity of the displacement field along Γ for the domain Ω_2 is discussed. As indicated in Section 3.2, with sufficient nodal discretization of the interface Γ, the linearity of the displacement field can be met to within a desired tolerance ε. Since the trial functions are no longer C^0, the numerical implementation assumes the form of a non-conforming method. In Section 7.5, the problem of an inclusion in an infinite matrix is solved using NEM. For the grid shown in Figure 30 with 13 nodes along Γ, the displacement discontinuity is less than 1% across Γ.

5. MODELLING OF NON-CONVEX BODIES

The modelling of non-convex bodies, such as two-dimensional straight or curved cracks using the Natural Element Method is addressed. Cracks are straight or curved lines of discontinuity in a two-dimensional body. From a modelling perspective, the numerical formulation must permit a displacement discontinuity to exist across the crack line. In finite elements, this is facilitated by having coincident nodes with independent degrees of freedom along the line of discontinuity.

Consider a non-convex domain with a straight edge-crack, which is modelled using two distinct crack faces *LC* and *MC* (Figure 14). The finite separation between the crack faces is illustrated for ease of exposition. Akin to the finite element method, one can also use coincident nodes with independent degrees of freedom along the crack line. In Figure 14(a), the nodes and the crack surfaces are shown while in Figure 14(b), the Delaunay triangles and the natural neighbour circumcircles for two triangles are presented. The triangulation of non-convex planar domains can be carried out using either conforming or constrained triangulation.[64, 65] In a conforming triangulation, the empty circumcircle property is preserved, and the triangulation is the strict dual of the Voronoi

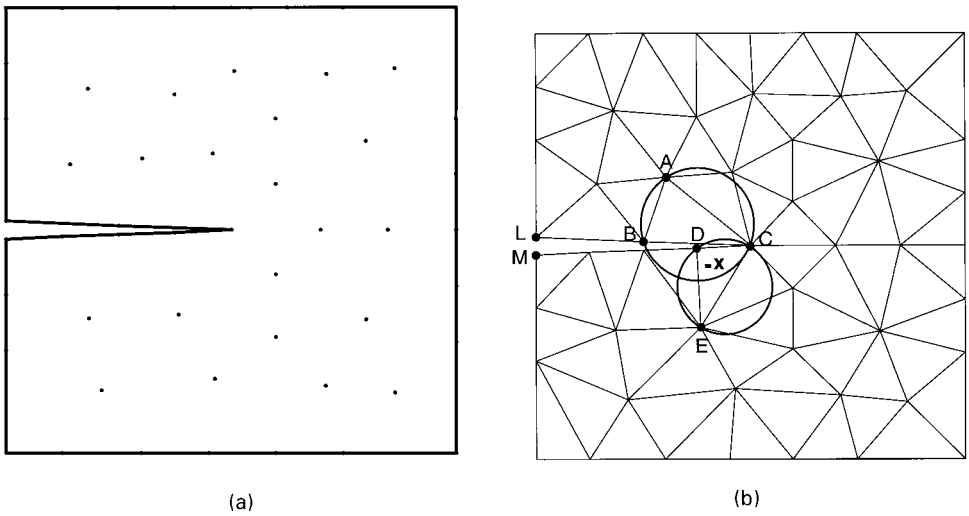


Figure 14. Crack modelling: (a) nodes; and (b) conforming triangulation

diagram, while in a constrained triangulation, the duality property does not hold. In Figure 14(b), the triangulation shown is a conforming one, which ensures the duality with the Voronoi diagram. In order to include the crack surfaces in the numerical modelling, the crack faces LC and MC are treated as boundaries for the upper and lower regions, respectively. In effect, for the domain above the crack line, the Voronoi cells for the nodes along LC are unbounded, while for the domain below the crack line, the Voronoi cells for the nodes along the boundary MC are unbounded. If we construct the circumcircles for $\triangle_{ABC}$ and $\triangle_{CDE}$ as shown in Figrue 14(b), the above description of crack modelling implies that nodes C, D, and E are natural neighbours of point $\mathbf{x}$, while nodes A and B are not. The above approach preserves the continuity of the trial function, and hence does not introduce any non-conformities in its implementation. In the modelling of crack problems where symmetry (two-fold or four-fold) can be invoked such that the domain of interest is convex, no modifications in NEM are required.

5.1. Enriching the interpolant for crack problems

The shape functions $\phi_I(\mathbf{x})$ form a partition of unity. As indicated in Section 3.2, this facilitates the enrichment of the NEM trial function for crack problems, along similar lines as suggested by Melenk and Babuška.[9] In finite elements, the quarter-point element results in $\sqrt{r}$ behaviour in the displacement field along rays emanating from the crack-tip. Since the exact displacement field solution for crack problems is $\mathcal{O}(\sqrt{r})$, this enhancement permits accurate modelling of the radial dependence of the crack-tip field in two dimensions. In the context of NEM, one can extrinsically enhance the trial function by adding terms that have local $\sqrt{r}$ behavior. Let $\Omega_0 \subset \Omega$ be a sub-domain in a region around the crack-tip where the crack-tip singularity dominates. Then the enriched trial function for two dimensional crack problems assumes the form:

$$u_i^h(\mathbf{x}) = \sum_{I=1}^{n} \phi_I(\mathbf{x}) u_{Ii} + \sqrt{r} \sum_{\substack{j=1 \\ x_j \in \Omega_0}}^{m} \phi_J(\mathbf{x}) a_{Ji} \quad (i = 1, 2) \tag{44}$$

where a_{Ji} are additional nodal coefficients assoicated with the m modes in Ω_0. The construction of the discrete system of linear equations using a Galerkin procedure is parallel to that indicated in Section 6. In this paper, the enriched interpolant is not used for the crack problem presented in Section 7.6; the NEM trial function given in equation (6) is used.

6. GOVERNING EQUATIONS AND WEAK FORM

We consider two-dimensional small displacement elastostatics, which is governed by the equation of equilibrium:

$$\boldsymbol{\nabla} \cdot \boldsymbol{\sigma} + \mathbf{b} = 0 \quad \text{in } \Omega \tag{45}$$

where $\Omega \subset \mathbb{R}^2$ is the domain of the body, $\boldsymbol{\nabla}$ is the gradient operator, $\boldsymbol{\sigma}$ is the Cauchy stress for a kinematically admissible displacement field $\mathbf{u}$, and $\mathbf{b}$ is the body force per unit volume. The constitutive relation is given by

$$\boldsymbol{\sigma} = \mathbf{C} : (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^*) \tag{46}$$

where $\boldsymbol{\varepsilon}$ is the small strain tensor, $\boldsymbol{\varepsilon}^*$ is the imposed eigenstrain tensor, and $\mathbf{C}$ is the material moduli tensor. The eigenstrain tensor is included in the formulation to permit the treatment of a transformation strain problem that appears in Section 7.5. The kinematic relation between the small strain tensor and the displacement vector $\mathbf{u}$ is

$$\boldsymbol{\varepsilon} = \boldsymbol{\nabla}_s \mathbf{u} \tag{47}$$

where $\boldsymbol{\nabla}_s$ is the symmetric gradient operator. The essential and natural boundary conditions are

$$\mathbf{u} = \bar{\mathbf{u}} \quad \text{on } \Gamma_u, \quad \mathbf{n} \cdot \boldsymbol{\sigma} = \bar{\mathbf{t}} \quad \text{on } \Gamma_t \quad (\Gamma = \Gamma_u \cup \Gamma_t) \tag{48}$$

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

The weak or variational form (principle of virtual work) associated with equation (45) is

$$\text{Find} \quad \mathbf{u} \in H^1(\Omega) \ni \int_\Omega \nabla_s \delta\mathbf{v} : \boldsymbol{\sigma} \, d\Omega = \int_\Omega \delta\mathbf{v} \cdot \mathbf{b} \, d\Omega + \int_{\Gamma_t} \delta\mathbf{v} \cdot \bar{\mathbf{t}} \, d\Gamma \ \ \forall \delta\mathbf{v} \in H_0^1(\Omega) \tag{49}$$

where $H^1(\Omega)$ is the Sobolev space of functions with square-integrable first derivatives in Ω, and $H_0^1(\Omega)$ is the Sobolev space of functions with square-integrable first derivatives in Ω and vanishing values on the essential boundary Γ_u.

On substituting the trial and test functions in the above equation and using the arbitrariness of nodal variations, the following discrete system of linear equations is obtained:

$$\mathbf{K}\mathbf{d} = \mathbf{f}^{\text{ext}} \tag{50}$$

where

$$\mathbf{K}_{IJ} = \int_\Omega \mathbf{B}_I^{\mathrm{T}} \mathbf{C} \mathbf{B}_J \, d\Omega \tag{51a}$$

$$\mathbf{f}_I^{\text{ext}} = \int_{\Gamma_t} \phi_I \bar{\mathbf{t}} \, d\Gamma + \int_\Omega \phi_I \mathbf{b} \, d\Omega + \int_\Omega \boldsymbol{B}_I^{\mathrm{T}} \mathbf{C} \boldsymbol{\varepsilon}^* \, d\Omega \tag{51b}$$

In the above equations, $\mathbf{B}_I$ is the matrix of shape function derivatives which is given by

$$\mathbf{B}_I = \begin{bmatrix} \phi_{I,x} & 0 \\ 0 & \phi_{I,y} \\ \phi_{I,y} & \phi_{I,x} \end{bmatrix} \tag{52}$$

The constitutive matrix $\mathbf{C}$ for an isotropic linear elastic material is

$$\mathbf{C} = \frac{E}{(1-2\nu)(1+\nu)} \begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & \dfrac{1-2\nu}{2} \end{bmatrix} \quad \text{(plane strain)} \tag{53a}$$

$$\mathbf{C} = \frac{E}{(1-\nu^2)} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \dfrac{1-\nu}{2} \end{bmatrix} \quad \text{(plane stress)} \tag{53b}$$

6.1. NEM implementation

In the context of the Natural Element Method, the stiffness matrix $\mathbf{K}$ and external force vector $\mathbf{f}^{\text{ext}}$ given in equation (51) are assembled on a nodal basis. This choice is necessitated due to the absence of an inherent element structure in NEM akin to the finite element method. In addition, the stiffness matrix in NEM is symmetric and sparse, but not necessarily banded. The Delaunay triangles are used as the background integration cells for the numerical integration computations. An alternative approach proposed by Traversoni[1] involves carrying out the numerical integration over the area of intersection of circumcircles. The idea is appealing since tensor product terms such as $\phi_{I,i}\phi_{J,j}$ are non-zero over the area of intersection of circumcircles about nodes n_I and n_J. In spite of its merits, the mappings and transformations involved appear to be non-trivial, with no cubature scheme currently known for such domains. Hence this approach is not pursued in this paper, and the Delaunay triangles themselves are used in the numerical integration of the weak form.

7. NUMERICAL RESULTS AND DISCUSSIONS

The application of NEM to problems in small displacement two-dimensional elastostatics, in the absence of body forces, is presented. Unless stated otherwise, the material properties chosen in the analyses are: $E = 3 \times 10^7$ psi and $\nu = 0{\cdot}25$. Numerical integration is carried out using symmetric quadrature rules for a triangle.[66, 67] In this paper, three point quadrature rule is used in the numerical integration of the weak form. The error norm computations are carried out using 25 point quadrature rule in each triangle.

7.1. Patch tests

The patch test[68, 69] is the ubiquitous test for the convergence of non-conforming finite element methods. Even though there remains significant disagreement as to whether it is a necessary condition for convergence, its merit and use as a benchmark for the evaluation and validation of

non-conforming elements is unquestionable. For a historical perspective and review of the patch test, see Felippa *et al.*[70] In the context of meshless methods, the relevance of the patch test from the viewpoint of convergence is still an open issue, which needs to be addressed and understood in a better light.

Displacement

In the displacement patch test, a linear displacement field is imposed on the boundary Γ_u ($\Gamma_u = \partial\Omega$) of a domain Ω. If the patch test is passed, the linear displacement field is approximated exactly, and the numerical solution at the interior nodes must be in exact (within machine precision) agreement with the linear solution. This validates, apart from the consistency condition, the accuracy of the numerical integration of the weak form in the Galerkin implementation. In Figure 15, three different nodal discretizations for a unit square are shown. In Figure 15(a), a uniform grid with 16 nodes is considered, and in Figure 15(e), a random set of 70 nodes is chosen. It is noted that for the uniform nodal grid shown in Figure 15(a), the Delaunay triangles are non-unique, even though the Voronoi diagram for the set of nodes is unique. A valid Delaunay triangulation for the regular nodal discretization is shown in Figure 15(b). For the displacement patch test, a linear displacement field $u_i = x_i$ is imposed on the boundary nodes of the unit square.

The NEM displacement patch test results for the relative $L^2(\Omega)$ error norm and relative $H^1(\Omega)$ error norm are presented in Table II. The $L^2(\Omega)$ and $H^1(\Omega)$ error norms are defined as follows:

$$\|\mathbf{u} - \mathbf{u}^h\|_{L^2(\Omega)} = \left(\int_\Omega (u_i - u_i^h)(u_i - u_i^h)\, \mathrm{d}\Omega \right)^{1/2} \tag{54a}$$

$$\|\mathbf{u} - \mathbf{u}^h\|_{H^1(\Omega)} = \left(\int_\Omega [(u_i - u_i^h)(u_i - u_i^h) + (u_{i,j} - u_{i,j}^h)(u_{i,j} - u_{i,j}^h)]\, \mathrm{d}\Omega \right)^{1/2} \tag{54b}$$

where $\mathbf{u}$ and $\mathbf{u}^h$ are the exact and numerical (NEM) displacement solutions, respectively. In Table II, the results are shown for both, 3 point, as well 25 point quadrature within each Delaunay triangle. For the error norm computations, a 25 point quadrature rule is used within each triangle. The error norm results indicate that the displacements and strains in the patch test are accurate to within 10^{-4} and 10^{-3}, respectively. Thus, since machine precision (double precision arithmetic) on a HP9000/s700 workstation is 10^{-16}, the patch test is not met with respect to this measure. Since the NEM interpolant can exactly reproduce a linear displacement field (Section 3.2), the numerical integration of the weak form is the source for the above discrepancy. Inexact numerical integration of the weak form (potential energy functional) is one of the variational crimes.[71] In NEM, two factors lead to the inexact numerical integration, namely, the non-polynomial form of the shape functions and the use of Delaunay triangles for numerical integration which are not coincident with the supports of the shape functions (Section 3.2). As indicated in Section 6.1, the natural domain for the numerical integration is the area of intersection (Ω_c) of natural neighbour circumcircles. There exist cubature rules with polynomial precision for two-dimensional domains such as, square, circle, triangle, and the plane.[72, 73] In addition, adaptive numerical integration procedures for other planar domains that are affine transformation of the above primitives are also available.[74] However, no general polynomial precision scheme for the domain Ω_c is known. It is envisaged that an easy-to-implement and robust numerical cubature scheme over Ω_c that is specifically tailored to the rational quartic form of NEM interpolants could lead to significant improvements in the numerical integration. The above shortcoming of NEM with respect to the patch test is also seen in meshless methods. For instance, in EFG, typical relative $L^2(\Omega)$ error norms in displacements

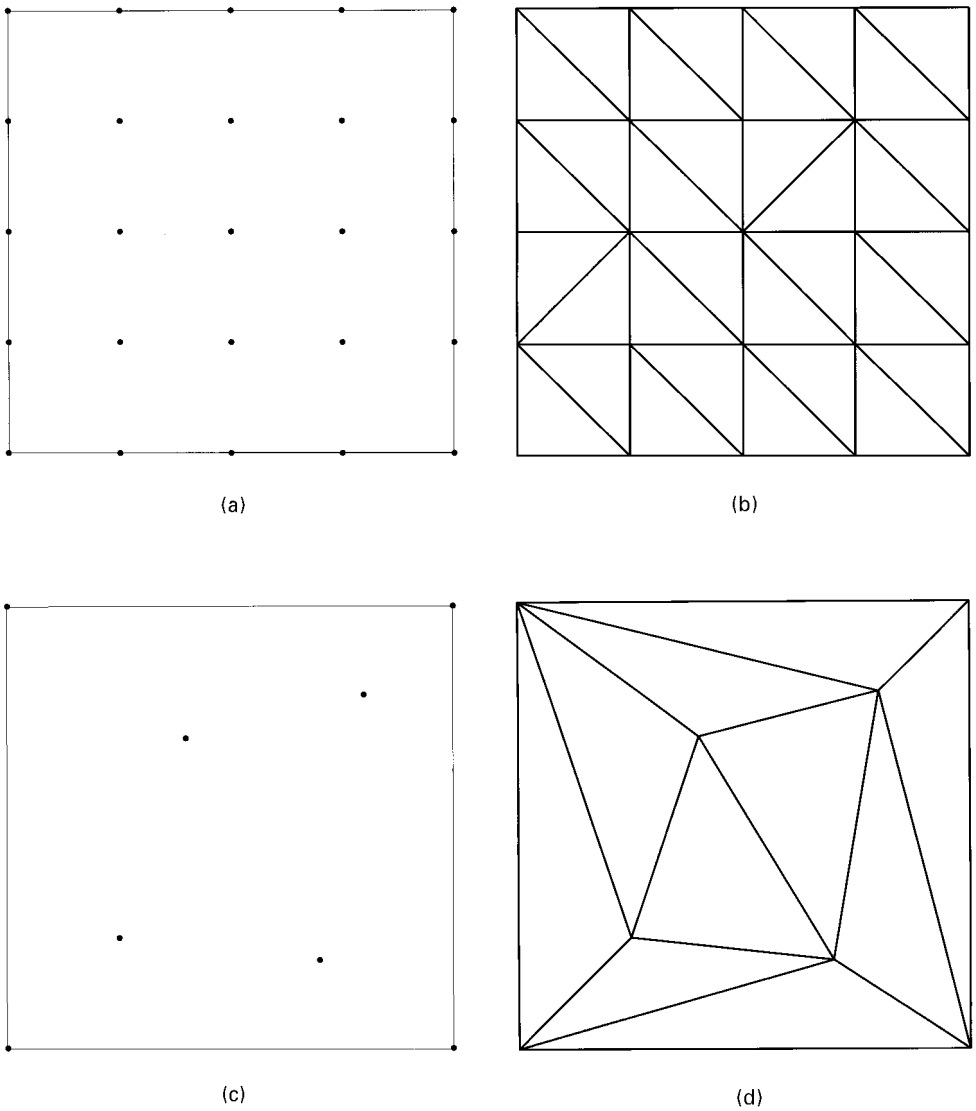


Figure 15. Displacement patch test: (a) and (b) uniform grid (25 nodes); (c) and (d) irregular grid (8 nodes); and (e) and (f) random set (70 nodes)

are obtained to within 10^{-6} accuracy with high Gauss–Legendre product rules over quadrilaterals for numerical integration. This again, is due to the fact that the numerical integration of the weak form is inexact.

Equilibrium

The ability to represent a uniaxial plane stress field is verified by the equilibrium patch test. Consider a unit square plate under a uniaxial stress $\sigma = 1$ psi (plane stress conditions) in the

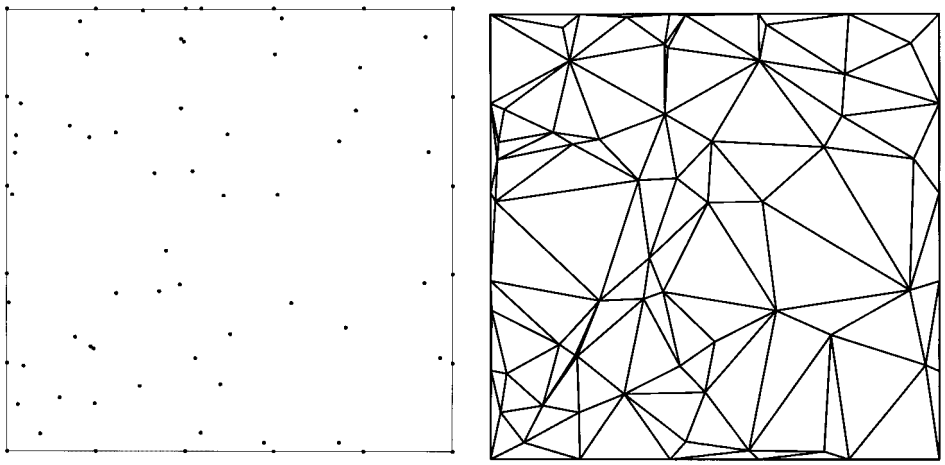


Figure 15. (*Continued*)

Table II. Relative error norms for the displacement patch test

Grids	Quadrature	$\frac{\|\mathbf{u}-\mathbf{u}^h\|_{L^2(\Omega)}}{\|\mathbf{u}\|_{L^2(\Omega)}}$	$\frac{\|\mathbf{u}-\mathbf{u}^h\|_{H^1(\Omega)}}{\|\mathbf{u}\|_{H^1(\Omega)}}$
a	3	$7{\cdot}5\times10^{-4}$	$4{\cdot}0\times10^{-3}$
	25	$8{\cdot}4\times10^{-5}$	$4{\cdot}5\times10^{-4}$
c	3	$9{\cdot}3\times10^{-3}$	$4{\cdot}1\times10^{-2}$
	25	$1{\cdot}1\times10^{-3}$	$4{\cdot}3\times10^{-3}$
e	3	$4{\cdot}4\times10^{-3}$	$9{\cdot}2\times10^{-2}$
	25	$4{\cdot}8\times10^{-4}$	$7{\cdot}9\times10^{-3}$

x_2-direction (Figure 16). The essential boundary conditions are indicated in Figure 16. The exact displacement solution is:

$$u_1 = \frac{v}{E}(1 - x_1) \tag{55a}$$

$$u_2 = \frac{x_2}{E} \tag{55b}$$

The three different nodal grids shown in Figure 15 are considered for the equilibrium patch test, and the relative error norm results are presented in Table III. In Table III, the energy norm $\|\mathbf{u} - \mathbf{u}^h\|_{E(\Omega)}$ is defined as:

$$\|\mathbf{u} - \mathbf{u}^h\|_{E(\Omega)} = \left(\frac{1}{2}\int_\Omega (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^h)^{\mathrm{T}}\mathbf{C}(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^h)\,\mathrm{d}\Omega\right)^{1/2} \tag{56}$$

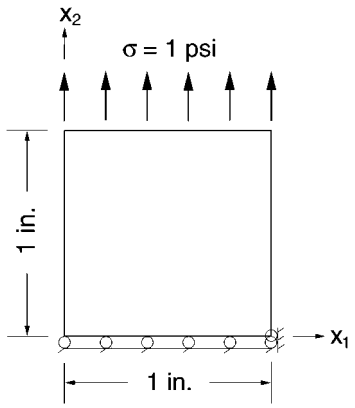


Figure 16. Equilibrium (uniaxial tension) patch test

Table III. Relative error norms for the equilibrium patch test

Grids	Quadrature	$\frac{\|\mathbf{u}-\mathbf{u}^h\|_{L^2(\Omega)}}{\|\mathbf{u}\|_{L^2(\Omega)}}$	$\frac{\|\mathbf{u}-\mathbf{u}^h\|_{E(\Omega)}}{\|\mathbf{u}\|_{E(\Omega)}}$
a	3	$1{\cdot}8\times10^{-3}$	$4{\cdot}5\times10^{-3}$
	25	$1{\cdot}7\times10^{-4}$	$4{\cdot}4\times10^{-4}$
c	3	$8{\cdot}2\times10^{-3}$	$3{\cdot}2\times10^{-2}$
	25	$1{\cdot}4\times10^{-3}$	$4{\cdot}0\times10^{-3}$
e	3	$6{\cdot}1\times10^{-3}$	$9{\cdot}5\times10^{-2}$
	25	$8{\cdot}8\times10^{-4}$	$6{\cdot}9\times10^{-3}$

In Table III, it is seen that the relative error norms in displacement and in energy are 10^{-4} and 10^{-3}, respectively. These are comparable to the patch test results obtained in Section 7.1, and again, the disagreement with the exact solution is due to inexact numerical integration of the weak form.

7.2. Infinite plate with a circular hole

An infinite plate with a traction free circular hole under unidirectional tension along x_1 is considered (Figure 17). The exact solution to this problem is given in Timoshenko and Goodier[75] as well as Szabó and Babuška.[76] The domain *ABCDE* shown in Figure 17 is modelled with the exact tractions imposed along *BC* and *CD*. Due to symmetry, the essential boundary conditions are: $u_2 = 0$ along *AB*, and $u_1 = 0$ along *DE*.

In polar co-ordinates (r, θ), the exact stress distribution for $\sigma_0 = 1$ psi is given by

$$\sigma_{11}(r,\theta) = 1 - \frac{a^2}{r^2}\left(\frac{3}{2}\cos 2\theta + \cos 4\theta\right) + \frac{3}{2}\frac{a^4}{r^4}\cos 4\theta \tag{57a}$$

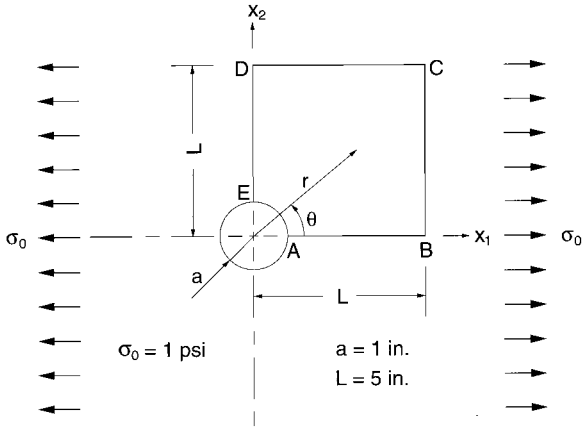


Figure 17. Plate with a circular hole under tension

$$\sigma_{22}(r,\theta) = -\frac{a^2}{r^2}\left(\frac{1}{2}\cos 2\theta - \cos 4\theta)\right) - \frac{3}{2}\frac{a^4}{r^4}\cos 4\theta \tag{57b}$$

$$\sigma_{12}(r,\theta) = -\frac{a^2}{r^2}\left(\frac{1}{2}\sin 2\theta + \sin 4\theta)\right) + \frac{3}{2}\frac{a^4}{r^4}\sin 4\theta \tag{57c}$$

where a is the radius of the circular hole. The displacement components (rigid-body displacement and rotation set to zero) are

$$u_1(r,\theta) = \frac{a}{8\mu}\left[\frac{r}{a}(\kappa+1)\cos\theta + 2\frac{a}{r}((1+\kappa)\cos\theta + \cos 3\theta) - 2\frac{a^3}{r^3}\cos 3\theta\right] \tag{58a}$$

$$u_2(r,\theta) = \frac{a}{8\mu}\left[\frac{r}{a}(\kappa-3)\sin\theta + 2\frac{a}{r}((1-\kappa)\sin\theta + \sin 3\theta) - 2\frac{a^3}{r^3}\sin 3\theta\right] \tag{58b}$$

where μ is the shear modulus and κ (Kolosov constant) is defined as

$$\kappa = \begin{cases} 3-4\nu & \text{(plane strain)} \\ \dfrac{3-\nu}{1+\nu} & \text{(plane stress)} \end{cases} \tag{59}$$

In the numerical computations, $a = 1$ in, $L = 5$ in, and plane strain conditions are assumed. The nodal discretizations used in the computations are shown in Figure 18.

In Figure 20, the rates of convergence (R) in displacement and energy for NEM and constant strain finite elements are presented. The theoretical convergence rates for the displacements and strains using finite elements (non-singular problems) are $R = 2$ and $R = 1$, respectively. It is observed from Figure 20 that the rates of convergence of NEM and FE are about the same, with NEM showing better absolute accuracy in displacements and strains. In Figure 20(a), the stress concentration factor ($\sigma_{11}^{\text{exact}}/\sigma_0 = 3$) at point E is indicated within braces. It is seen that NEM is able to accurately capture the stress concentration at point E. In Figure 19, the numerical and exact normal stress σ_{11} are plotted along the edge ED (see Figure 17). The grid shown in Figure 18(c)

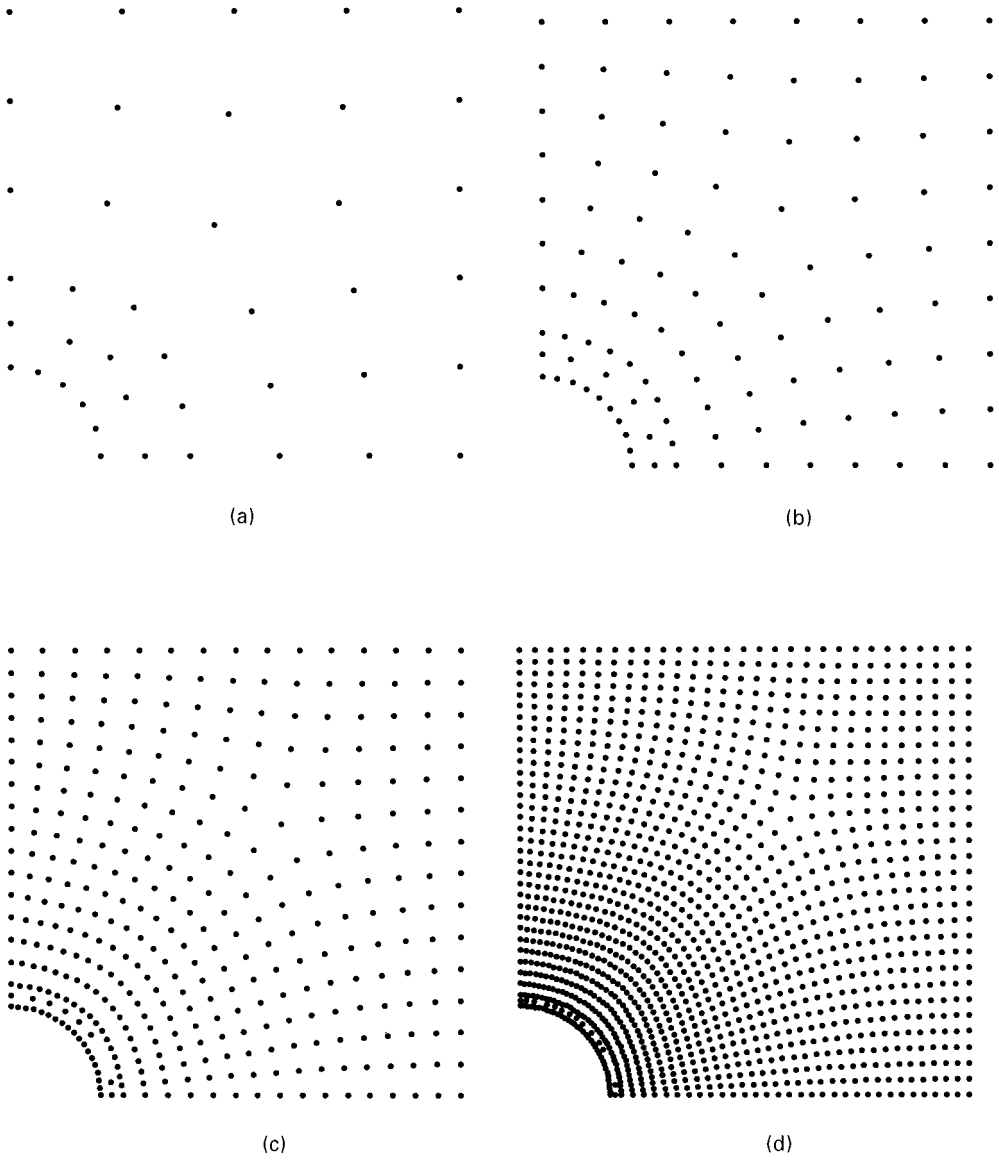


Figure 18. Nodal discretization for the plate with a hole problem: (a) 41 nodes; (b) 108 nodes; (c) 361 nodes; and (d) 1345 nodes

is used, and 240 equi-spaced output points between $r = 1$ in and $r = 5$ in are chosen in the computations. Agreement between the NEM and the exact stress solution is excellent. The displacement along the edge is linear between two adjacent nodes, and hence one observes the jumps in the stress σ_{11} at the nodes.

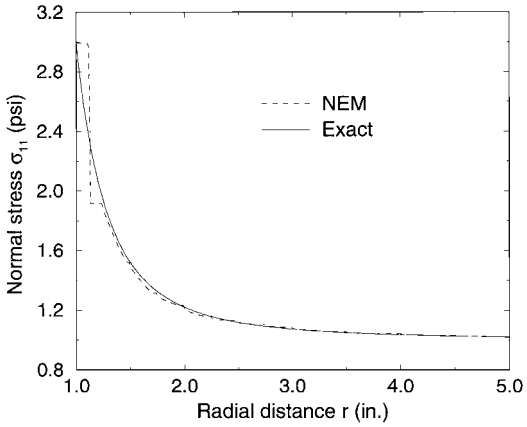


Figure 19. Comparison of σ_{11} for NEM and the exact solution along a radial line ($\theta = 90°$)

7.3. Cantilever beam

In Figure 21, a cantilever beam subjected to a parabolic end load is illustrated. The beam has characteristic length L, height D and unit thickness, and is assumed to be in a state of plane stress. The displacement vector solution is given by[75]

$$u_1 = \frac{-Px_2}{6EI}\left[(6L - 3x_1)x_1 + (2+\nu)x_2^2 - \frac{3D^2}{2}(1+\nu)\right] \tag{60a}$$

$$u_2 = \frac{P}{6EI}[3\nu x_2^2(L - x_1) + (3L - x_1)x_1^2] \tag{60b}$$

while the stresses are

$$\sigma_{11} = \frac{-P(L - x_1)x_2}{I} \tag{61a}$$

$$\sigma_{22} = 0 \tag{61b}$$

$$\sigma_{12} = \frac{P}{2I}\left(\frac{D^2}{4} - x_2^2\right) \tag{61c}$$

where I is the moment of inertia, which for a beam with rectangular cross-section and unit thickness is:

$$I = D^3/12 \tag{61d}$$

In the numerical model, the analytical displacement solution from equation 60 is prescribed on the boundary Γ_u: $x_1 = 0$, $-D/2 \leqslant x_2 \leqslant D/2$ (Figure 21). On the remaining boundaries, exact tractions are specified. The following parameters are used in the numerical computations: $P = -1000$ psi, $D = 1$ in, $L = 4$ in, and plane stress conditions are assumed.

Convergence studies are carried out using four different nodal discretizations, namely 85 nodes, 297 nodes, 1105 nodes, and 1701 nodes. Equal nodal spacing in the x_1- and x_2-direction is used in each of the above grids. In Figure 22, a sample nodal discretization (85 nodes) is shown. The $L^2(\Omega)$ and energy error norms are calculated on the basis of equations (54a) and (56), respectively. In Figure 23, the relative displacement and energy error norms are plotted against the nodal spacing h on a log–log plot. The rate of convergence is indicated by the value of R. The convergence rate for NEM is close to the theoretical rate for finite elements, namely $R = 2$ and $R = 1$ for

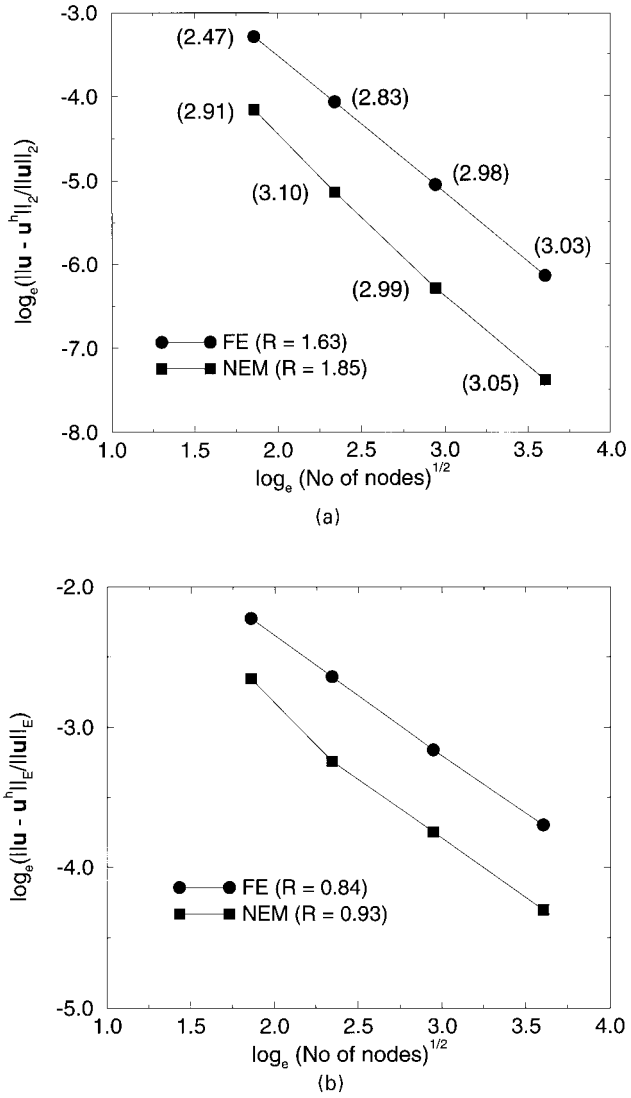


Figure 20. Rate of convergence for the plate with a hole problem: (a) displacement; and (b) energy

displacements and strains, respectively. The absolute accuracy of NEM is observed to be better than that of the finite element solution.

The NEM and analytical stress fields are compared near the mid-line ($x = 2{\cdot}03$ in) of the beam. Since the shape function algorithm fails for points along $x = 2$ in (Delaunay edges), such a choice is necessitated. In Figure 24, the normal and shear stresses at $x = 2{\cdot}03$ in are plotted as a function of y. The nodal grid consisting of 1105 nodes is used, and the stress outputs are carried out at 200 equi-distant points between $y = -0{\cdot}5$ in and $y = 0{\cdot}5$ in. The NEM and the analytical solution are in good agreement. The maximum error in the normal stress σ_{11} is less than 1 per cent, and the shear stress profile also matches the analytical solution well.

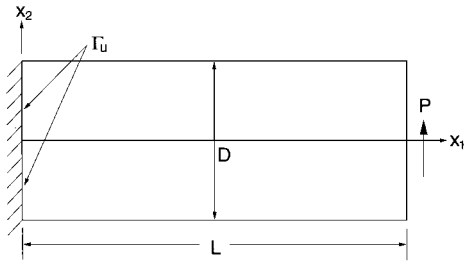


Figure 21. Cantilever beam model

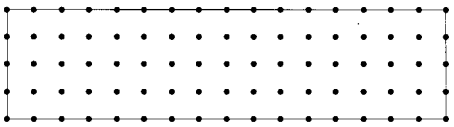


Figure 22. Regular nodal discretization for cantilever beam model (85 nodes)

7.4. Hollow cylinder under internal pressure

Consider a hollow cylinder of internal and external radius of a and b, respectively (Figure 25). A uniform pressure p_i is applied to the inner surface ($r=a$), while the outer surface ($r=b$) is traction free. Due to axi-symmetry, the stresses are independent of θ, and hence only have a radial dependence. The exact solution for the stress components in polar co-ordinates (r,θ) is:[75]

$$\sigma_r(r) = \frac{a^2 p_i}{b^2 - a^2}\left(1 - \frac{b^2}{r^2}\right) \tag{62a}$$

$$\sigma_\theta(r) = \frac{a^2 p_i}{b^2 - a^2}\left(1 + \frac{b^2}{r^2}\right) \tag{62b}$$

$$\sigma_{r\theta} = 0 \tag{62c}$$

Under plane stress conditions, the components of the small strain tensor are

$$\varepsilon_r(r) = \frac{a^2 p_i}{E(b^2 - a^2)}\left\{1 - \nu - \frac{b^2}{r^2}(1+\nu)\right\} \tag{63a}$$

$$\varepsilon_\theta(r) = \frac{a^2 p_i}{E(b^2 - a^2)}\left\{1 - \nu + \frac{b^2}{r^2}(1+\nu)\right\} \tag{63b}$$

$$\varepsilon_{r\theta} = 0 \tag{63c}$$

while the radial and tangential displacements are given by

$$u_r(r) = \frac{a^2 p_i r}{E(b^2 - a^2)}\left\{1 - \nu + \frac{b^2}{r^2}(1+\nu)\right\} \tag{64a}$$

$$u_\theta = 0 \tag{64b}$$

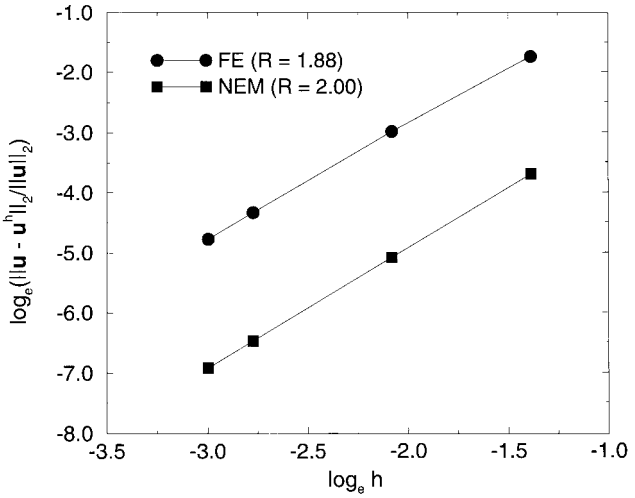


Figure 23. Rate of convergence for the cantilever beam problem: (a) displacement; and (b) energy

In the numerical computations, the following parameters are chosen: $a = 1$ in, $b = 5$ in, $p_i = 3 \times 10^4$ psi, and plane stress conditions are assumed. Due to symmetry, only one-quarter of the specimen is modelled. In the analyses, six different nodal discretizations are considered, namely, 169 nodes, 361 nodes, 625 nodes, 1369 nodes, 1849 nodes, and 2401 nodes. In Figure 26, the nodal grids for 169 and 625 nodes are illustrated. A convergence study is carried out using the above

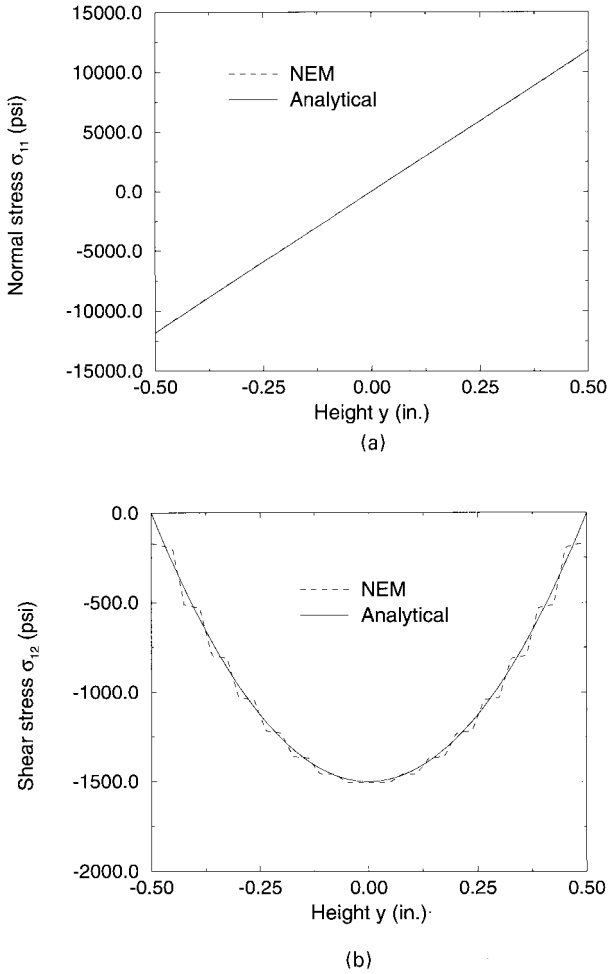


Figure 24. Comparison of normal stress and shear stress for the cantilever beam model: (a) normal stress σ_{11}; and (b) shear stress σ_{12}

six nodal discretizations. The exact $L^2(\Omega)$ and energy norms are given by

$$\|\mathbf{u}\|_{L^2(\Omega)} = \alpha\sqrt{\frac{\pi}{2}}\sqrt{\frac{(1-\nu)^2(b^4-a^4)}{4} + (1-\nu^2)b^2(b^2-a^2) + b^4(1+\nu)^2\log\frac{b}{a}} \tag{65a}$$

$$\|\mathbf{u}\|_{E(\Omega)} = \alpha\sqrt{\frac{\pi E}{4}}\sqrt{(1-\nu)(b^2-a^2) + b^4(1+\nu)\left(\frac{1}{a^2}-\frac{1}{b^2}\right)} \tag{65b}$$

where

$$\alpha = \frac{a^2 p_i}{E(b^2-a^2)} \tag{65c}$$

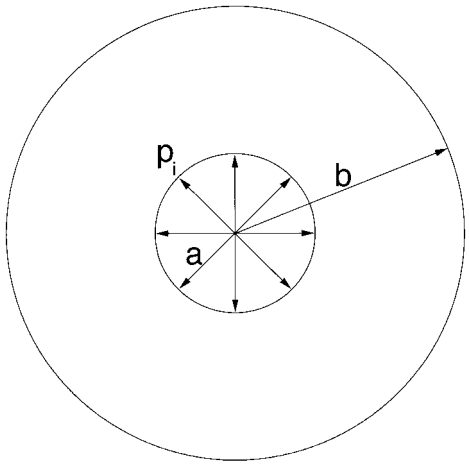


Figure 25. Hollow cylinder under internal pressure p_i

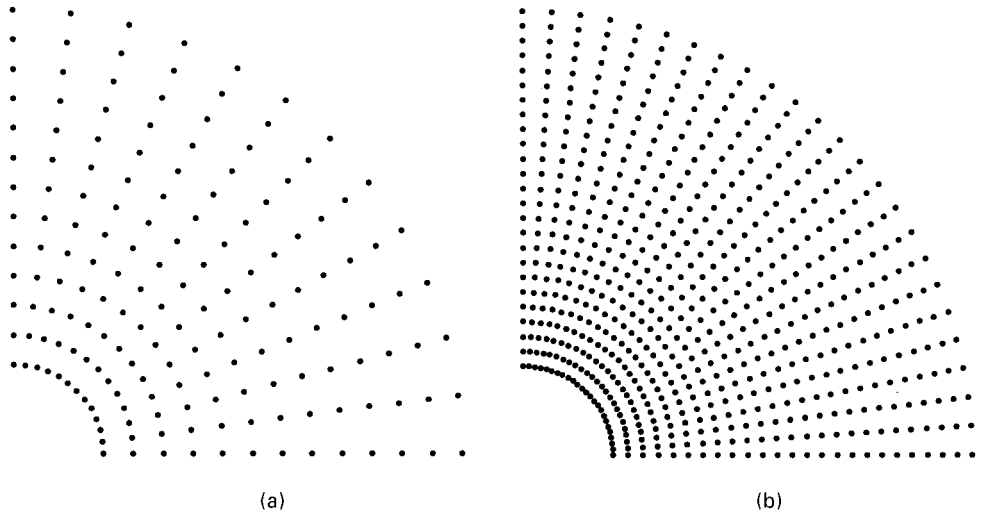


Figure 26. Sample nodal discretizations for the hollow cylinder under internal pressure problem: (a) 169 nodes; and (b) 625 nodes

For the specimen dimensions, internal pressure p_i, and material properties that are considered here, $\|\mathbf{u}\|_{L^2(\Omega)} = 2{\cdot}462 \times 10^{-3}\,\text{in}^2$ and $\|\mathbf{u}\|_{E(\Omega)} = 5{\cdot}605\,\text{psi}^{1/2}\,\text{in}^{1/2}$. In Figure 27, the convergence rate (R) in displacement and energy are presented for NEM and constant strain finite elements. The trends are similar to that observed in the earlier examples, with similar rates for NEM and FE, with NEM showing better absolute accuracy in displacements and strains.

A comparison of the NEM and exact stress fields is carried out using the grid consisting of 1369 nodes. The numerical solution recovered the rotational symmetry of the solution, and hence for the purpose of comparison, we plot the stresses along a radial line ($\theta = 30°$). In Figure 28, the NEM and exact solutions for the radial stress and the hoop stress are presented. In the computations,

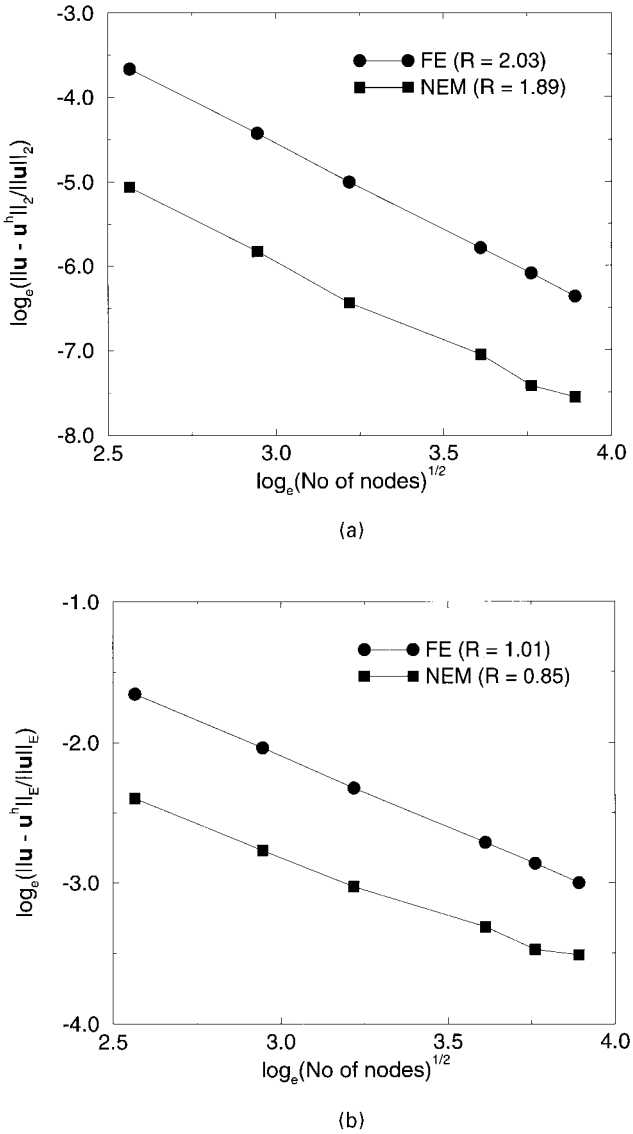


Figure 27. Rate of convergence for the hollow cylinder under internal pressure problem: (a) displacement; and (b) energy

180 equi-distant output points between $r = 1$ in and $r = 5$ in are considered. The NEM and exact stress fields are in good agreement. Since the NEM displacement field is C^0 at the nodes, the oscillatory character of the numerical stresses is present. However, the oscillations are bounded, and they fluctuate about the exact solution, which is analogous to the finite element method, where the stresses are discontinuous across element boundaries.

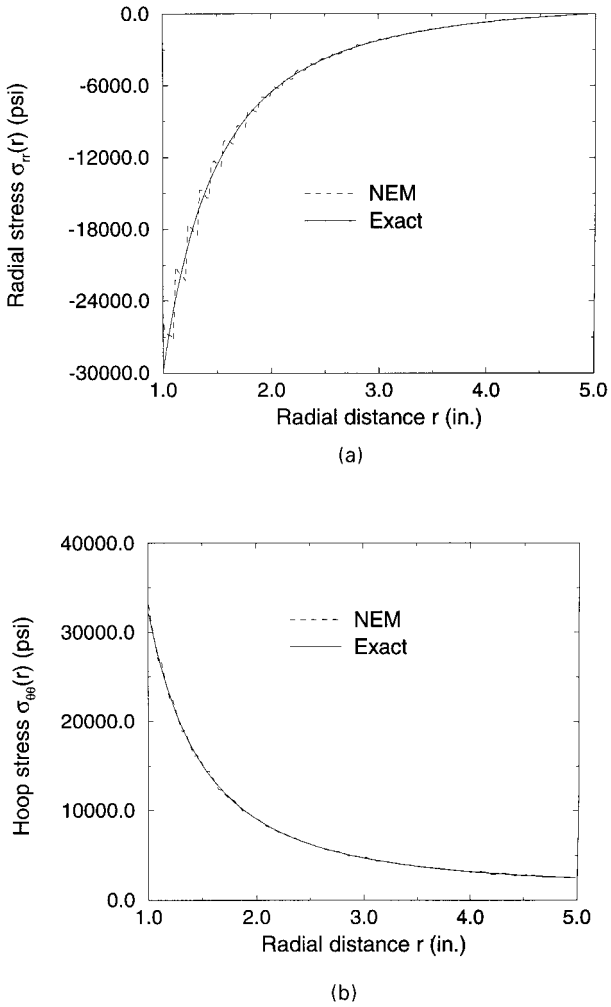


Figure 28. Comparison of the radial and hoop stresses for the hollow cylinder under internal pressure problem: (a) radial stress $\sigma_{rr}(r)$; and (b) hoop stress $\sigma_{\theta\theta}(r)$

7.5. 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 29, a graphical representation of the problem is illustrated. The exact displacement vector solution in polar co-ordinates is given by[77]

$$u_r(r) = \begin{cases} C_1 r & r \leqslant R \\ C_1 \frac{R^2}{r} & r \geqslant R \end{cases} \tag{66a}$$

$$u_\theta = 0 \tag{66b}$$

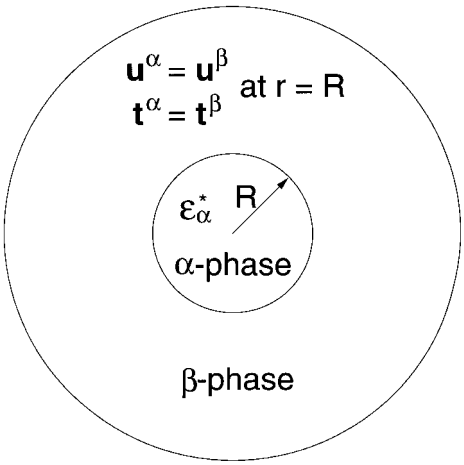


Figure 29. Inclusion embedded in an infinite matrix

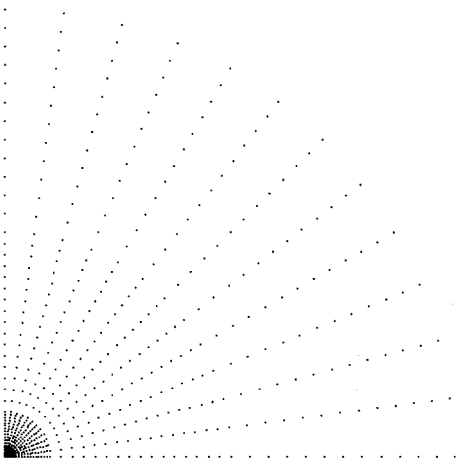


Figure 30. Nodal discretization for inclusion in an infinite matrix problem

where

$$C_1 = \frac{(\mu^\alpha + \lambda^\alpha)\bar{\varepsilon}^*_\alpha}{\mu^\alpha + \lambda^\alpha + \mu^\beta} \tag{67}$$

In the above equation, μ and λ are the Lamé constants of the respective phases, while the eigenstrain $\bar{\varepsilon}^*_\alpha$ is a constant dilatational strain. The material properties used in the numerical computation are:[63] $\lambda^\alpha = 497{\cdot}16$, $\mu^\alpha = 390{\cdot}63$ in the α-phase, while the constants in the β-phase are $\lambda^\beta = 656{\cdot}79$, $\mu^\beta = 338{\cdot}35$. These correspond to $E^\alpha = 1000$, $v^\alpha = 0{\cdot}28$, $E^\beta = 900$, and $v^\beta = 0{\cdot}33$. A constant

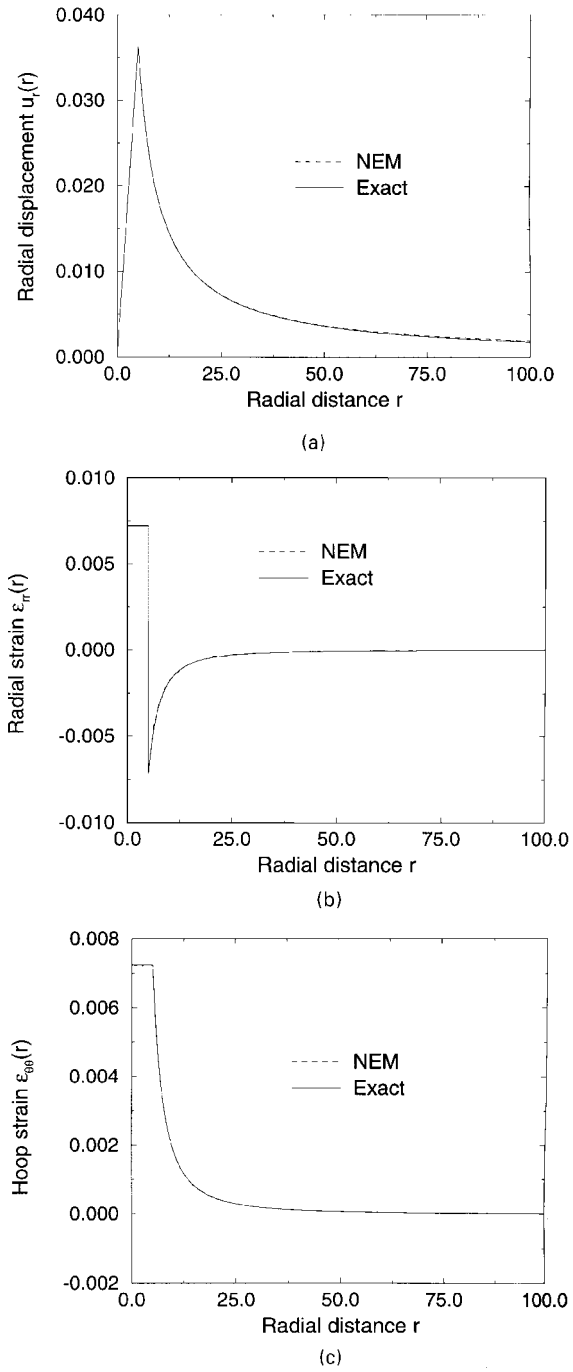
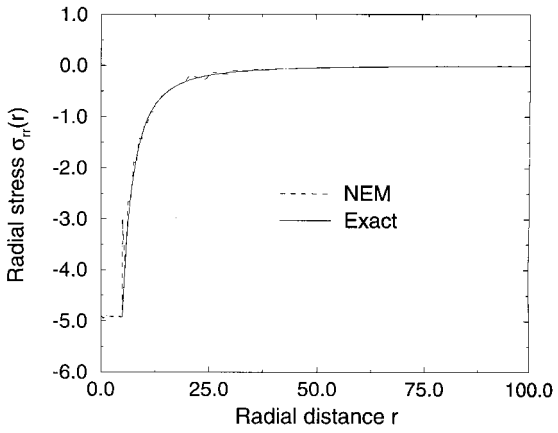
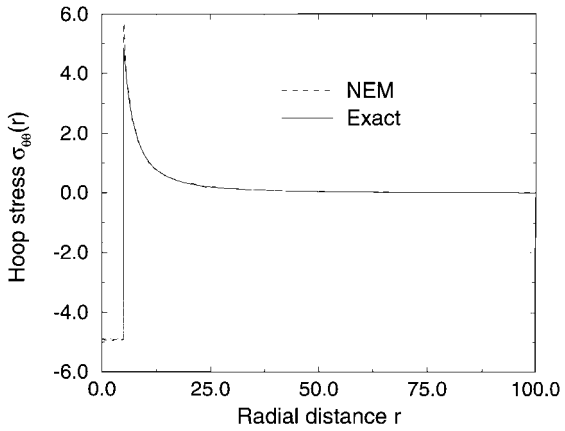


Figure 31. Comparison of NEM and the exact solution for an inclusion with a dilatational eigenstrain in an infinite matrix: (a) radial displacement $u_r(r)$; (b) radial strain $\varepsilon_{rr}(r)$; (c) hoop strain $\varepsilon_{\theta\theta}(r)$; (d) radial stress $\sigma_{rr}(r)$; and (e) hoop stress $\sigma_{\theta\theta}(r)$



(d)



(e)

Figure 31. (*Continued*)

dilatation eigenstrain $\bar{\varepsilon}^*_\alpha = 0{\cdot}01$ 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 30, 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 conditions are imposed along the lines of symmetry, while the outer radius $R_0 = 200$ is traction free. Plane strain conditions are assumed in the numerical computations.

The NEM solution recovered the cylindrical symmetry in the solution, and hence results are presented as a function of only the radial distance. The maximum error in the numerically

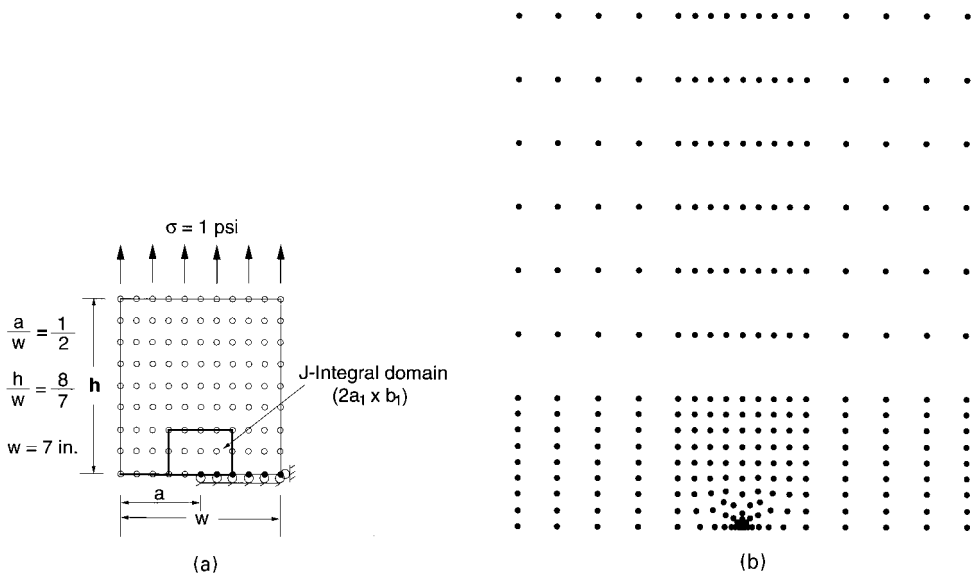


Figure 32. Edge-cracked plate under tension: (a) regular nodal discretization (99 nodes); and (b) refined nodal discretization (278 nodes)

computed tangential displacement u_θ and shear stress $\sigma_{r\theta}$ are $\mathcal{O}(10^{-5})$ and $\mathcal{O}(10^{-4})$, respectively. In Figure 31, a comparison of the NEM and exact solution is presented. The results shown in Figure 31 are computed along a radial line ($r=0$ to $r=100$) at $\theta=30°$. Along the radial line, 30 equi-spaced output points are chosen within the inclusion, and 150 equi-spaced points in the matrix. Excellent agreement between the NEM and the analytical solution is observed. The slight discrepancy in $u_r(r)$ at $r>25$ is due to the finite-dimension of the matrix as opposed to the theoretical solution which is based on an infinite matrix. 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.

7.6. Edge-cracked plate under tension

An edge-cracked plate under pure mode I loading is considered. Due to symmetry, only the upper-half is modelled. In Figure 32(a), the specimen dimensions and loading are indicated; the crack length a is half of the plate width w. Since the domain is convex, no modifications are required in the implementation of NEM. The mode I stress intensity factors (SIFs) are computed using the domain integral[78] form of the contour J-integral.

Two different nodal discretizations are considered: a regular grid consisting of 99 nodes shown in Figure 32(a) and an irregular grid of 278 nodes with focused array of nodes in the vicinity of the crack-tip (Figure 32(b)). In Table IV, the normalized mode I stress intensity factors for both grids are presented. The reference solution for this edge-crack problem is: $K_I^{\text{ref}}/\sigma\sqrt{\pi a}=2{\cdot}8264$.[79] From Table IV it is seen that the SIFs computed using NEM are more accurate than the corresponding values evaluated by constant strain finite elements. Domain independence is also clearly observed for the NEM results. The SIFs obtained using NEM for the refined grid (Figure 32(b)) are within

Table IV. Normalized K_I: Edge-cracked plate under tension

Nodal grid	Domains a_1	Domains b_1	$\frac{K_I}{\sigma\sqrt{\pi a}}$ (NEM)	Error percentage	$\frac{K_I}{\sigma\sqrt{\pi a}}$ (FE)	Error percentage
	2·8	3·0	2·5241	10·7	2·1159	25·1
	2·8	5·0	2·5360	10·3	2·1691	23·3
Regular	3·5	5·0	2·5310	10·5	2·1664	23·3
	3·5	6·0	2·5339	10·4	2·1806	22·8
	3·5	8·0	2·5375	10·2	2·1989	22·2
	1·0	2·0	2·7867	1·4	2·6567	6·0
	1·0	4·0	2·7893	1·3	2·6569	6·0
Refined	1·0	6·0	2·7905	1·3	2·6586	5·9
	1·0	8·0	2·7911	1·2	2·6597	5·9
	3·5	8·0	2·7867	1·4	2·6752	5·3

1·5 per cent of the reference solution results. The domain independence of the SIFs using NEM and the excellent agreement with the reference solution results are encouraging and promising indicators for the application of NEM to SIF evaluation for 2-D crack problems.

8. CONCLUSIONS

In this paper, the application of the Natural Element Method (NEM) to elliptic boundary-value problems in two-dimensional solid mechanics has been studied. In NEM, the interpolants used to construct the trial and test functions are known as natural neighbour (n–n) interpolants. These interpolants are based on the Dirichlet or Voronoi tessellation of a set (N) of distinct nodes in the plane. The Voronoi tessellation is a unique and geometrically fundamental construct that defines a set of nodes. Natural neighbour interpolants have optimum spatial adjacency properties, and are sensitive to the position and the density of nodes. Natural neighbour interpolants are smooth (C^∞) everywhere, except at the nodes where they are C^0. In one-dimension, NEM interpolation is identical to linear finite elements. In NEM, a standard Galerkin procedure is used to obtain the discrete equations. Several problems in solid mechanics have been presented and comparisons made to results obtained using constant strain finite elements as well as to reference solutions to validate the accuracy and convergence of NEM. In the patch tests, issues pertaining to the exact integration of the weak form were raised. Benchmark problems, such as the cantilever beam, plate with a hole, and hollow cylinder under internal pressure, which have gradients in the strain and stress solutions were solved using NEM. The NEM results were in excellent agreement with exact and analytical solutions for these problems. The ability to model material discontinuities was illustrated by applying NEM to the bi-material problem of an inclusion with a constant eigenstrain, embedded in an infinite matrix. The numerical results showed good agreement with the exact solution. The modelling of non-convex bodies, such as crack surfaces, was also described. The benchmark problem of an edge-crack under mode I loading was considered to test the accuracy of NEM in stress intensity factor computations. The stress intensity factors computed using NEM were within 1·5 per cent of the reference solution, and very good domain independence was achieved. The timings involved in evaluating natural neighbour shape functions are only about twice that

of constant strain finite elements, which renders NEM as a very feasible and attractive choice from the computational cost and time viewpoint. The various problems presented in this paper demonstrate the capabilities, versatility, accuracy, and robustness of the Natural Element Method, and provides impetus for its application to other classes of problems such as crack growth, plates, and large deformations in solid as well as fluid mechanics.

ACKNOWLEDGEMENTS

The financial support of the Federal Aviation Administration and the Office of Naval Research is gratefully acknowledged. The first author thanks Dr. David Watson for helpful discussions on natural neighbour interpolation. Helpful comments by Thomas Black during the course of this work are also appreciated.

REFERENCES

1. L. Traversoni, 'Natural neighbor finite elements', *Int. Conf. on Hydraulic Engineering Software*, *Hydrosoft Proc.*, Vol. 2, Computational Mechanics Publications, 1994, pp. 291–297.
2. J. Braun and M. Sambridge, 'A numerical method for solving partial differential equations on highly irregular evolving grids', *Nature*, **376,** 655–660 (1995).
3. J. J. Monaghan, 'An Introduction to SPH', *Comput. Phys. Commun.*, **48,** 89–96 (1988).
4. B. Nayroles, G. Touzot and P. Villon, 'Generalizing the finite element method: diffuse approximation and diffuse elements', *Comput. Mech.*, **10,** 307–318 (1992).
5. T. Belytschko, Y. Y. Lu and L. Gu, 'Element-free Galerkin methods', *Int. J. Numer. Meth. Engng.*, **37,** 229–256 (1994).
6. D. Sulsky, Z. Chen and H. L. Schreyer, 'A particle method for history-dependent materials', *Comput. Meth. Appl. Mech. Engng.*, **118,** 179–186 (1994).
7. W. K. Liu, S. Jun and Y. F. Zhang, 'Reproducing kernel particle methods', *Int. J. Numer. Meth. Engng.*, **20,** 1081–1106 (1995).
8. E. Oñate, S. Idelsohn, O. C. Zienkiewicz, R. L. Taylor and C. Sacco, 'A stabilised finite point method for analysis of fluid mechanics problems', *Comput. Meth. Appl. Mech. Engng.*, **139,** 315–346 (1996).
9. J. M. Melenk and I. Babuška, 'The partition of unity finite element method: Basic theory and applications', *Comput. Meth. Appl. Mech. Engng.*, **139,** 289–314 (1996).
10. C. A. Duarte and J. T. Oden, 'An *H-p* adaptive method using clouds', *Comput. Meth. Appl. Mech. Engng.*, **139,** 237–262 (1996).
11. T. Belytschko, Y. Krongauz, D. Organ, M. Fleming and P. Krysl, 'Meshless methods: an overview and recent developments', *Comput. Meth. Appl. Mech. Engng.*, **139,** 3–47 (1996).
12. R. Sibson, 'A vector identity for the Dirichlet tesselation', *Math. Proc. Cambridge Philos. Soc.*, **87,** 151–155 (1980).
13. R. Sibson, 'A brief description of natural neighbor interpolation', in V. Barnett (ed.), *Interpreting Multivariate Data*, Wiley, Chichester, 1981, pp. 21–36.
14. P. Alfeld, 'Scattered data interpolation in three or more variables', in T. Lyche and L. L. Schumaker (eds.), *Mathematical Methods in Computer Aided Geometric Design*, Academic Press, San Diego, 1989, pp. 1–34.
15. D. F. Watson and G. M. Philip, 'Neighborhood-based interpolation', *Geobyte*, **2**(2), 12–16 (1987).
16. D. F. Watson, 'Natural neighbor sorting on the N-dimensional sphere', *Pattern Recognition*, **21**(1), 63–67 (1988).
17. C. M. Gold, 'Surface interpolation, spatial adjacency and GIS', in J. Raper (ed.), *Three Dimensional Applications in Geographical Information Systems*, Taylor and Francis, London, 1989, pp. 21–35.
18. G. Farin, 'Surfaces over Dirichlet tessellations', *Comput. Aided Geom. Des.*, **7**(1–4), 281–292 (1990).
19. S. J. Owens, 'An implementation of natural neighbor interpolation in three dimensions', *Master's Thesis*, Brigham Young University, 1992.
20. D. F. Watson, *Contouring*: *A Guide to the Analysis and Display of Spatial Data*, Pergamon Press, Oxford, 1992.
21. L. Traversoni, 'An algorithm for natural spline interpolation', *Numer. Algorithms*, **5,** 63–70 (1993).
22. P. Piper, 'Properties of local coordinates based on Dirichlet tessellations', in G. Farin, H. Hagen and H. Noltemeier (eds.), *Geometric Modelling*, Vol. 8, Springer, Wien, 1993, pp. 227–239.
23. D. F. Watson, *nngridr*: *An implementation of natural neighbor interpolation*, David Watson, Claremont, WA, Australia, 1994.
24. J. L. Brown, 'Natural neighbor interpolation on the sphere', in P. J. Laurent, A. L. Méhauté and L. L. Schumaker (eds.), *Wavelets*, *Images and Surface Fitting*, Peters, A.K., Wellesley, MA, 1994, pp. 67–74.

25. N. L. Jones, S. J. Owens and E. C. Perry, 'Plume characterization with natural neighbor interpolation', *Proc. GEOENVIRONMENT 2000*, *Geotechnical Engineering and Environmental Engineering Divisions/ASCE*, New York, NY, 1995, 331–345.
26. J. A. Robinson, 'Image coding with ridge and valley primitives', *IEEE Trans. Commun*., **43**(6), 2095–2102 (1995).
27. J. Braun, M. Sambridge and H. McQueen, 'Geophysical parameterization and interpolation of irregular data using natural neighbors', *Geophys. J. Int*., **122,** 837–857 (1995).
28. C. Perez and L. Traversoni, 'Finite element simulation of shallow waters using natural neighbors techniques', In *Computational Methods in Surface Flows and Transport Problems*: *Internation Conference on Computational Methods in Water Resources*, Vol. 2, 1996, 239–245.
29. G. M. Voronoi, 'Nouvelles applications des paramètres continus à la théorie des formes quadratiques. deuxième Mémoire: Recherches sur les paralléllоèdres primitifs', *J. Reine Angew. Math*., **134,** 198–287 (1908).
30. B. Delaunay, 'Sur la sphère vide. A la memoire de Georges Voronoi', *Izv. Akad. Nauk SSSR*, *Otdelenie Matematicheskih i Estestvennyh Nauk*, **7,** 793–800 (1934).
31. M. I. Shamos, 'Computational Geometry', *Ph.D. Thesis*, Department of Computer Science, Yale University, New Haven, CT, 1978.
32. F. Preparata and M. Shamos, *Computational Geometry*: *An Introduction*, Springer, New York, NY, 1985.
33. J. O'Rourke, *Computational Geometry in C*, Cambridge University Press, Cambridge, 1994.
34. D. Shepard, 'A two-dimensional interpolation function for irregularly spaced points', *ACM National Conf*., 1968, pp. 517–524.
35. P. Lancaster and K. Salkauskas, 'Surfaces generated by moving least squares methods', *Math. Comput*., **37,** 141–158 (1981).
36. S. Rippa, 'Interpolation and smoothing of scattered data by radial basis functions', *Master's Thesis*, Tel Aviv University, 1984.
37. M. J. D. Powell, 'Radial basis functions for multivariable interpolation', *Algorithms for Approximations*, Clarendon Press, Oxford, 1987, pp. 143–168.
38. R. L. Hardy, 'Multiquadric equations of topography and other irregular surfaces', *J. Geophys. Res*., **76,** 1905–1915 (1971).
39. I. Babuška and A. Aziz, 'On the angle condition in the finite element method', *SIAM J. Numer. Anal.*, **13,** 214–227 (1976).
40. P. J. Green and R. R. Sibson, 'Computing Dirichlet tessellations in the plane', *Comput. J*., **21,** 168–173 (1978).
41. B. N. Boots, *Voronoi* (*Thiessen*) *Polygons*, Geo Books, Norwich, U.K., 1986.
42. A. Okabe, B. Boots and K. Sugihara, *Spatial Tessellations*: *Concepts and Applications of Voronoi Diagrams*, Wiley, Chichester, England, 1992.
43. F. Aurenhammer, Voronoi diagrams—a survey of a fundamental geometric data structure', *ACM Trans. Math. Software*, **23,** 469–483 (1996).
44. C. L. Lawson, 'Software for C^1 surface interpolation', in J. R. Rice (ed.), *Mathematical Software III*, Vol. 3, Academic Press, New York, N.Y., 1977.
45. K. Q. Brown, 'Voronoi diagrams from convex hulls', *Inform. Process. Lett*., **9,** 223–228 (1979).
46. A. Bowyer, 'Computing Dirichlet tessellations', *Comput. J*., **24,** 162–166 (1981).
47. D. F. Watson, 'Computing the *n*-dimensional Delaunay tessellation with application to Voronoi polytopes', *Comput. J*., **24**(2), 167–172 (1981).
48. D. T. Lee and B. J. Schachter, 'Two algorithms for constructing a Delaunay triangulation', *Int. J. Comput. Inform. Sci*., **9,** 219–242 (1980).
49. S. Fortune, 'A sweepline algorithm for Voronoi diagrams', *Algorithmica*, **2,** 153–174 (1987).
50. J. R. Shewchuk, 'Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator', *First Workshop on Applied Computational Geometry*, Association for Computing Machinery, New York, May 1996, pp. 124–133.
51. J. R. Shewchuk, *Triangle*: *A Two-Dimensional Quality Mesh Generator and Delaunay Triangulator*, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, 1996b. Available at `http://www.cs.cmu.edu/~quake/triangle.html`.
52. J. R. Shewchuk, *Show Me*: *A Display Program for Meshes and More*, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, 1996a. Available at `http.//www.cs.cmu.edu/~quake/showme.html`.
53. C. B. Barber and H. T. Huhdanpaa, *Qhull* (*Computer Program*), The Geometry Center, University of Minnesota, 1300 South Second Street, Minneapolis, MN 55454, 1994. Available at `http://www.geom.umn.edu/software/qhull/.`
54. C. B. Barber, D. P. Dobkin and H. T. Huhdanpaa, 'The Quickhull algorithm for convex hull', *ACM Trans. Math. Software*, **22,** 469–483 (1996).
55. K. Mulmuley, *Computational Geometry*: *An Introduction Through Randomized Algorithms*, Prentice-Hall, Englewood Cliffs, N.J., 1994.
56. S. Fortune, 'Voronoi diagrams and Delaunay triangulations', in D.-Z. Du and F. K. Hwang (eds.), *Computing in Euclidean Geometry*, *Lecture Notes Series on Computing*, Vol. 1, World Scientific, Singapore, 1995, pp. 193–233.
57. M. Bern and D. Eppstein, 'Mesh generation and optimal triangulation', in D.-Z. Du and F. K. Hwang (eds.), *Computing in Euclidean Geometry*, *Lecture Notes Series on Computing*, Vol. 1, World Scientific, Singapore, 1995, pp. 23–90.

58. T. J. R. Hughes, *The Finite Element Method*, Englewood Cliffs, Prentice-Hall, N.J., 1987.
59. N. Ahuja, 'Dot pattern processing using Voronoi polygons as neighborhoods', *IEEE Trans. Pattern Anal. Mach. Intell.,* **4,** 336–343 (1982).
60. L. Traversoni and O. Palacios, 'A building method for hierarchical covering spheres of a given set of points', in P. J. Laurent, A. L. Méhauté and L. L. Schumaker (eds.), *Curves and Surfaces*, Academic Press, Boston, MA, 1991, pp. 457–460.
61. J. B. Lasserre, 'An analytical expression and an algorithm for the volume of a convex polyhedron in R^n', *J. Optimiz. Theory Appl.*, **39**(3), 363–377 (1983).
62. R. J. Mackinnon and G. F. Carey, 'Treatment of material discontinuities in finite element computations', *Int. J. Numer. Meth. Eng.*, **24,** 393–417 (1987).
63. L. W. Cordes and B. Moran, 'Treatment of material discontinuity in the element-free Galerkin method', *Comput. Meth. Appl. Mech. Engng.*, **139,** 75–89 (1996).
64. S. H. Lo, 'Delaunay triangulation of non-convex planar domains', *Int. J. Numer. Meth. Engng.*, **28**(11), 2695–2707 (1989).
65. C. Borgers, 'Generalized Delaunay triangulations of nonconvex domains', *Comput. Math. Appl.*, **20**(7), 45–49 (1990).
66. J. N. Lyness and D. Jespersen, 'Moderate degree symmetric quadrature rules for the triangle', *J. Inst. Math. Appl.*, **15,** 19–32 (1975).
67. D. A. Dunavant, 'High degree efficient symmetrical Gaussian quadrature rules for the triangle', *Int. J. Numer. Meth. Engng.*, **21,** 1129–1148 (1985).
68. G. R. Bazeley, Y. K. Cheung, B. M. Irons and O. C. Zienkiewicz, 'Triangular elements in plate bending. Conforming and non-conforming solutions', *Proc. 1st Conf. on Matrix Methods in Structural Mechanics*, Wright-Patterson AFB, 1965.
69. B. M. Irons and A. Razzaque, 'Experience with the patch test for convergence of finite elements', in A. K. Aziz (ed.), *The Mathematical Foundations of the Finite Element Method with Applications to Partial Differential Equations*, Academic Press, New York, 1972.
70. C. A. Felippa, B. Haugen and C. Militello, 'From the individual element test to finite element templates: evolution of the patch test', *Int. J. Numer. Meth. Engng.*, **38,** 199–229 (1995).
71. G. Strang and G. Fix, *An Analysis of the Finite Element Method*, Prentice-Hall, Englewood Cliffs, N.J., 1973.
72. A. H. Stroud, *Approximate Calculation of Multiple Integrals*, Prentice-Hall, Englewood Cliffs, N.J., 1971.
73. R. Cools and P. Rabinowitz, 'Monomial cubature rules since 'Stroud': a compilation', *J. Comput. Appl. Math.*, **23,** 1–15 (1997).
74. R. Cools, D. Laurie and L. Pluym, 'Algorithm 764: Cubpack++: A C++ package for automatic two-dimensional cubature', *ACM Trans. Math. Software*, **19,** 309–326 (1997).
75. S. P. Timoshenko and J. N. Goodier, *Theory of Elasticity*, 3rd edn, McGraw Hill, New York, 1970.
76. B. Szabó and I. Babuška, *Finite Element Analysis*, Wiley, New York, 1991.
77. T. Mura, *Micromechanics of Defects in Solids*, Martinus Nijhoff, The Hague, Netherlands, 1987.
78. B. Moran and C. F. Shih, 'Crack tip and associated domain integrals from momentum and energy balance', *Engng. Fracture. Mech.*, **27**(6), 615–641 (1987).
79. H. Tada, P. C. Paris and G. R. Irwin, *The Stress Analysis of Cracks Handbook*, Del Research Corporation, St. Louis, MO, 1977.

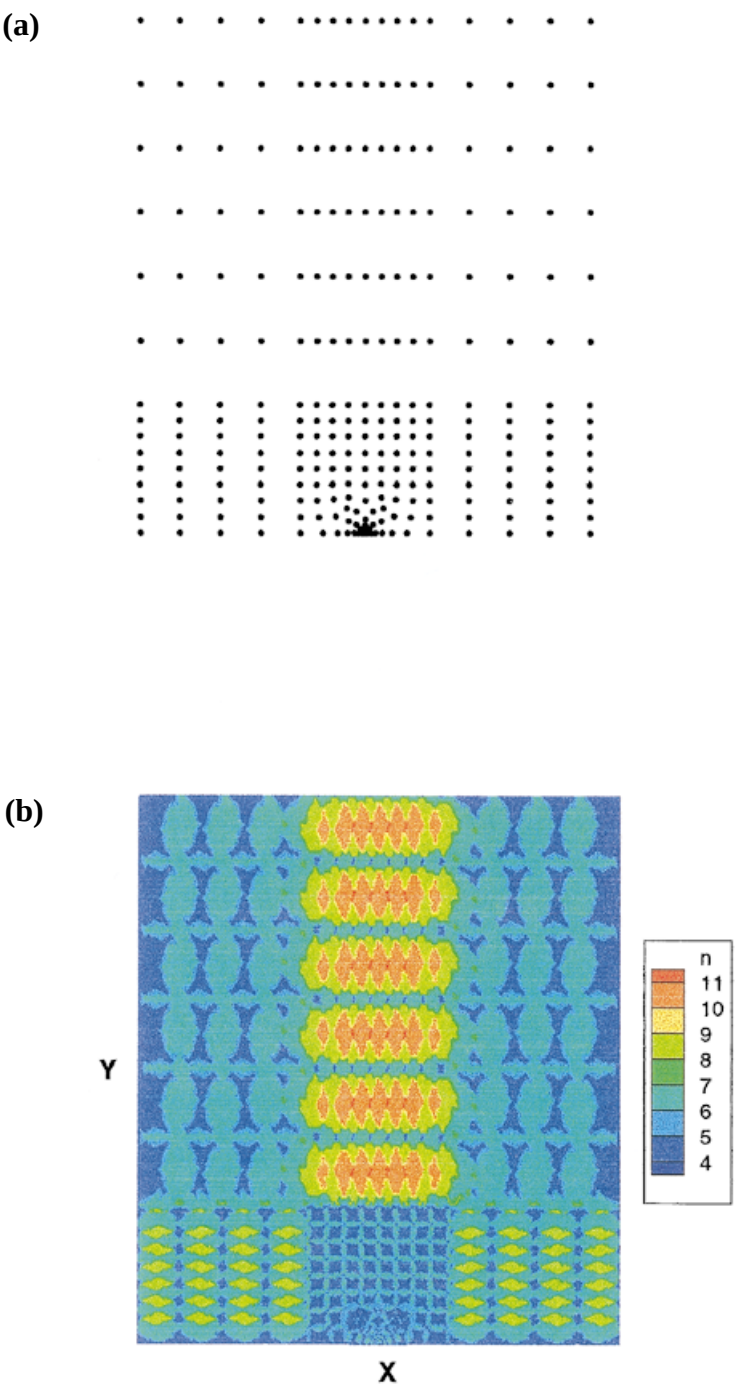


Plate 1. Variation of natural neighbours for an irregular grid: (a) nodal grid;
(b) contour plot of natural neighbours