Chapter 3

Natural Neighbor Interpolation

In this chapter, a detailed description of the construction, properties, and numerical computations of natural neighbor 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. As a preliminary introduction and a means to set the stage for what follows, we first present a brief description of Voronoi diagrams and Delaunay triangles in the context of natural neighbor interpolation.

3.1 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 work, 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 neighbor) 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 (Green and Sibson, 1978)

$$T_I = \{\mathbf{x} \in \mathbb{R}^2 \,:\, d(\mathbf{x}, \mathbf{x}_I) < d(\mathbf{x}, \mathbf{x}_J) \;\forall\; J \neq I\}, \tag{3.1}$$

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 3.1a and 3.1b, respectively. In Fig. 3.1a 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 $\mathbb{R}^d$) of the line joining nodes n_I and n_J $(J \neq I)$. Consequently, for all nodes n_I that are inside the convex hull[1], the Voronoi cells are closed and convex, while the cells associated with nodes on the boundary of the convex hull are unbounded (see Fig. 3.1b). The nearest neighbor 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 (1986) and Okabe, Boots, and Sugihara (1992), while Aurenhammer (1996) presents a comprehensive review of Voronoi polygons.

The Delaunay triangulation, which is the straight-line dual of the Voronoi di-

[1]The convex hull $\mathrm{CH}(N)$ of the set of nodes N is the smallest convex set containing N.

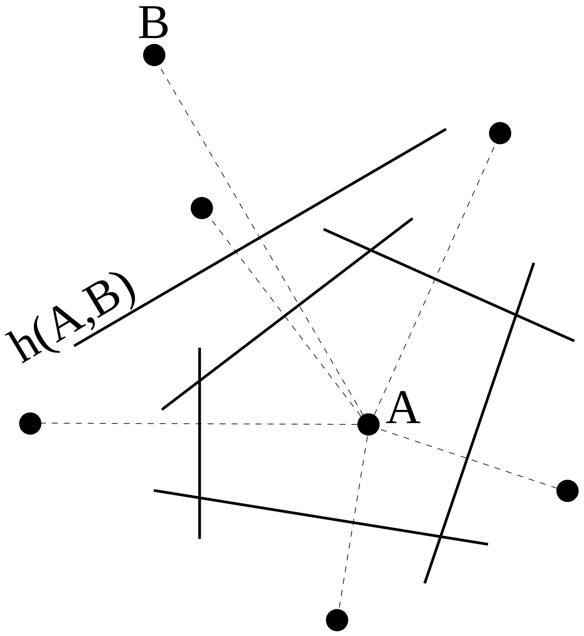
agram, is constructed by connecting the nodes whose Voronoi cells have common boundaries—see Fig. 3.1c. The duality between the two implies that there is a Delaunay edge between two nodes in the plane if and only if their Voronoi cells share a common edge. Among all triangles, the Delaunay triangles maximize the minimum angle (Lawson, 1977). Another important property of Delaunay triangles is the *empty circumcircle criterion* (Lawson, 1977)—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 neighbor interpolation, these circles are known as natural neighbor circumcircles (Watson, 1992). The center of the natural neighbor 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 Fig. 3.1d, the natural neighbor 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 $\mathbb{R}^{d+1}$ (Brown, 1979), which is also the basis for computing the Voronoi diagram in $\mathbb{R}^d$. The importance of the Delaunay triangulation in the context of the solution of PDEs stems from its use in mesh generation for the finite element method. Incremental insertion (Lawson, 1977) and point insertion (Bowyer, 1981; Watson, 1981) are among the early algorithms developed to compute Delaunay triangulations. Today, some of the prominent Delaunay triangulation algorithms are those based on incremental insertion (Lawson, 1977), divide-and-conquer (Lee and Schachter, 1980), and plane sweep (Fortune, 1987). The optimum time-complexity of

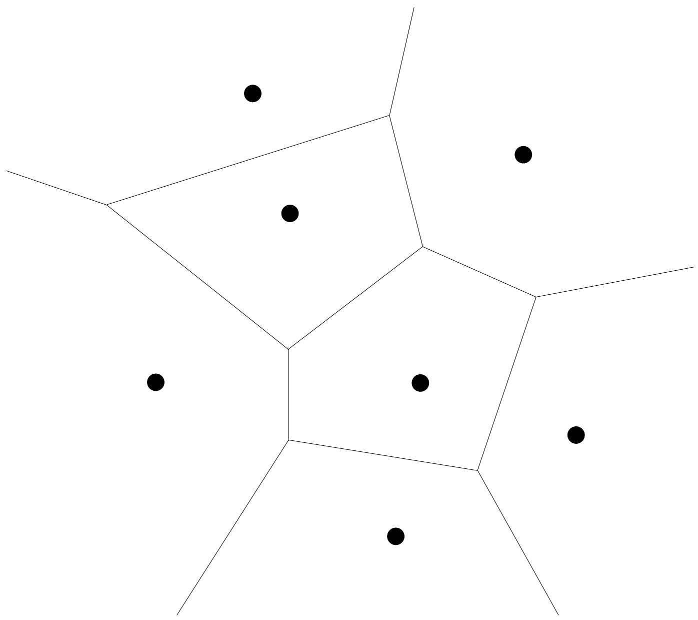
Delaunay triangulation algorithms is $\mathcal{O}(n \log n)$. Shewchuk (1996c) has carried out a comparison of the computational costs and speeds of the above three algorithms. In this work, we use the package *Triangle* (Shewchuk, 1996b); the nodal discretization and Delaunay triangles are displayed using the package *Show Me* (Shewchuk, 1996a). In three-dimensions and higher, the *qhull* package (Barber and Huhdanpaa, 1994; Barber, Dobkin, and Huhdanpaa, 1996) which is based on the quickhull algorithm is considered to be the most versatile. A discussion on randomized algorithms to compute Voronoi diagrams and Delaunayq tessellations can be found in Mulmuley (1994). For details on Delaunay triangulation and mesh generation, see Fortune (1995) and the review article by Bern and Eppstein (1995).

3.2 Construction

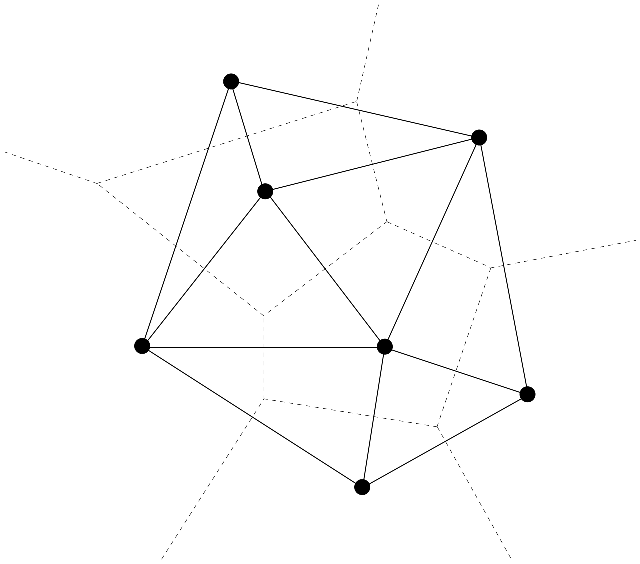
Natural neighbor coordinates were introduced by Sibson (1980) as a means for data interpolation and smoothing. The concept of nearest neighbors and neighboring nodes is embedded in the first-order Voronoi diagram (Section 3.1). By a similar extension, one can construct higher order (k-order, $k > 1$) Voronoi diagrams in the plane. Of particular interest in the context of natural neighbor coordinates 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-neighbor-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 neighbor, and n_J as the second nearest neighbor. It is emphasized that the cell T_{IJ} is non-empty if and only if n_I and n_J are neighbors. The second-order Voronoi cell T_{IJ} $(I \neq J)$ is



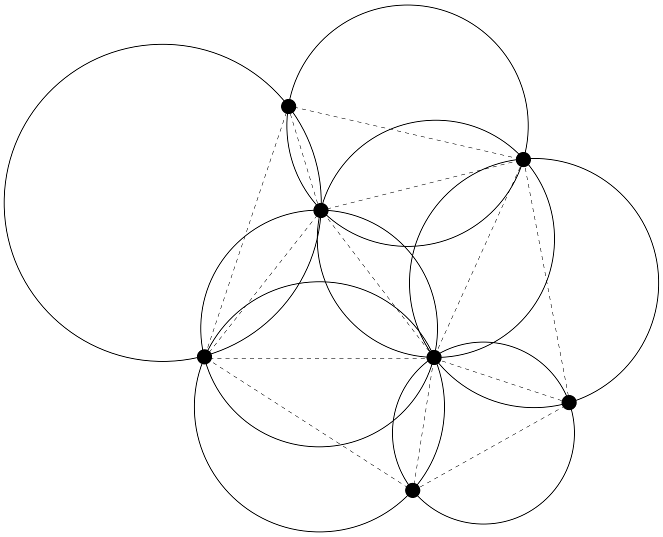
(a)



(b)



(c)



(d)

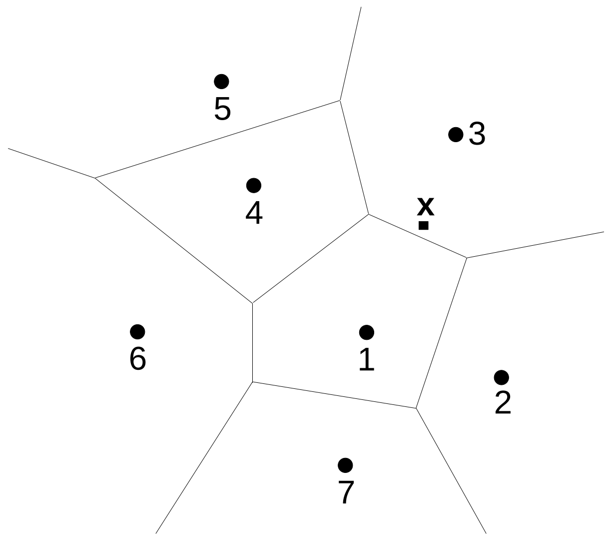
Figure 3.1: Geometric structures for a set N of seven nodes. (a) Voronoi cell for node A, (b) Voronoi diagram $V(N)$, (c) Delaunay triangulation DT(N), and (d) Natural neighbor circumcircles.

defined as (Sibson, 1980)

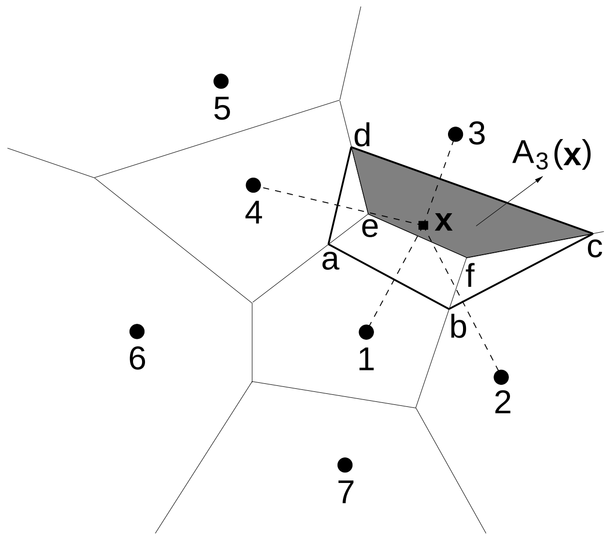
$$T_{IJ} = \{\mathbf{x} \in \mathbb{R}^2 \,:\, d(\mathbf{x}, \mathbf{x}_I) < d(\mathbf{x}, \mathbf{x}_J) < d(\mathbf{x}, \mathbf{x}_K) \;\forall\; K \neq I,\, J\}. \tag{3.2}$$

In order to quantify the neighbor relation for any point $\mathbf{x}$ introduced into the tessellation, Sibson (1980) used the concept of second-order Voronoi cells, and thereby introduced natural neighbors and natural neighbor coordinates. The notion of neighboring nodes is broadened and generalized to yield a new measure of "neighborliness" by the definition of natural neighbors. In Fig. 3.2a, a point $\mathbf{x}$ is introduced into the Voronoi diagram of the set N discussed in Section 3.1. If $\mathbf{x}$ is tessellated along with the set of nodes N, then the natural neighbors of $\mathbf{x}$ are those nodes which form an edge of a triangle with $\mathbf{x}$ in the new triangulation. A straight-forward means to arrive at the same end is to use the empty circumcircle criterion (see Section 3.1). 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 neighbors. In Fig. 3.2b, the perpendicular bisectors from point $\mathbf{x}$ to its natural neighbors 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 neighbors, namely nodes 1, 2, 3, and 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 neighbor coordinates of $\mathbf{x}$ with respect to a natural neighbor I is defined as the ratio of the area of overlap of the Voronoi cells T_I and $T_{\mathbf{x}}$ to the total area of the



(a)



(b)

Figure 3.2: Construction of natural neighbor coordinates. (a) Original Voronoi diagram and $\mathbf{x}$, and (b) $1st$-order and $2nd$-order Voronoi cells about $\mathbf{x}$.

Voronoi cell of $\mathbf{x}$:

$$\phi_I(\mathbf{x}) = \frac{A_I(\mathbf{x})}{A(\mathbf{x})}, \tag{3.3}$$

where I ranges from 1 to n, and $A(\mathbf{x}) = \sum_{J=1}^{n} A_J(\mathbf{x})$. The four regions shown in Fig. 3.2b are the second-order cells, while their union (closed polygon $abcd$) is a first-order Voronoi cell. Referring to Fig. 3.2, the shape function $\phi_3(\mathbf{x})$ is given by

$$\phi_3(\mathbf{x}) = \frac{A_3(\mathbf{x})}{A(\mathbf{x})}. \tag{3.4}$$

The derivatives of the n-n coordinates are obtained by differentiating Eq. (3.3):

$$\phi_{I,j}(\mathbf{x}) = \frac{A_{I,j}(\mathbf{x}) - \phi_I(\mathbf{x})A_{,j}(\mathbf{x})}{A(\mathbf{x})} \quad (j = 1,\, 2), \tag{3.5}$$

where Eq. (3.3) has been used to arrive at the above expression.

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

$$\mathbf{u}^h(\mathbf{x}) = \sum_{I=1}^{n} \phi_I(\mathbf{x})\mathbf{u}_I, \tag{3.6}$$

where $\mathbf{u}_I$ $(I = 1, 2, \ldots, n)$ are the vectors of nodal displacements at the n natural neighbors, and $\phi_I(\mathbf{x})$ are the shape functions associated with each node. In the context of natural neighbor interpolation, the shape functions $\phi_I(\mathbf{x})$ are taken as the n-n coordinates of the point $\mathbf{x}$ in the plane. It is noteworthy to point out here that since the shape functions have compact support (Section 3.3.4), Eq. (3.6) 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 neighbor shape function" or simply "shape function" are used synonymously for natural neighbor coordinates.

3.3 Properties

In this section, the properties of natural neighbor 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 (1990b).

3.3.1 Interpolation

By definition of the shape function given in Eq. (3.3), the following property is self-evident:

$$0 \leq \phi_I(\mathbf{x}) \leq 1. \tag{3.7}$$

Now, referring to Fig. 3.2b, we note that if $\mathbf{x}$ were to coincide with any node, say node 3 for instance, then it is readily seen that $\phi_3(\mathbf{x}) = 1$ and $\phi_I(\mathbf{x}) = 0$, $I \neq 3$. Therefore, the NEM and FE shape functions share the following property:

$$\phi_I(\mathbf{x}_J) = \delta_{IJ} \tag{3.8}$$

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 $\mathbf{u}^h(\mathbf{x}_I) \neq \mathbf{u}_I$, i.e., the nodal parameters $\mathbf{u}_I$ are not the nodal displacements.

3.3.2 Partition of Unity

By construction (see Eq. (3.3)), we have the following relation:

$$\sum_{I=1}^{n} \phi_I(\mathbf{x}) = 1 \quad \text{in } \Omega, \tag{3.9}$$

where n is the number of natural neighbors for the point $\mathbf{x}$. Hence, by virtue of Eq. (3.7) and the above property, we note that the shape functions form a partition of unity (Melenk and Babuška, 1996). The implication is that this interpolant can exactly reproduce constant functions; in addition, the approximation space can also be enriched by additional functions. In Section 4.2.1, this is discussed at greater length in the context of enriching the NEM interpolant for crack problems.

3.3.3 Linear Completeness

For a second-order PDE such as elastostatics, linear consistency or completeness (Hughes, 1987) is the ability of the interpolant to exactly reproduce constant and linear displacement fields. Sibson (1980) has shown (see also Piper (1993)) that the natural neighbor shape functions satisfy the local coordinate property, namely

$$\mathbf{x} = \sum_{I=1}^{n} \phi_I(\mathbf{x}) \mathbf{x}_I, \tag{3.10}$$

which indicates that the shape functions can exactly reproduce the geometrical coordinates. We now show that Eqs. (3.9) and (3.10) together imply that linear completeness is satisfied.

Proof. Consider a linear displacement field in the form:

$$\mathbf{u}(\mathbf{x}) = \boldsymbol{\alpha} + \boldsymbol{\beta}^T \mathbf{x}, \tag{3.11}$$

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

$$\mathbf{u}_I = \boldsymbol{\alpha} + \boldsymbol{\beta}^T \mathbf{x}_I, \tag{3.12}$$

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

$$\mathbf{u}^h(\mathbf{x}) = \sum_{I=1}^{n} \phi_I(\mathbf{x}) \mathbf{u}_I, \tag{3.13}$$

where $\mathbf{u}_I$ is the vector of nodal displacements for node I. On using Eq. (3.12) in the above equation, we obtain

$$\mathbf{u}^h(\mathbf{x}) = \boldsymbol{\alpha} \sum_{I=1}^{n} \phi_I(\mathbf{x}) + \boldsymbol{\beta}^T \sum_{I=1}^{n} \phi_I(\mathbf{x}) \mathbf{x}_I, \tag{3.14}$$

which on using Eqs. (3.9) and (3.10) reduces to

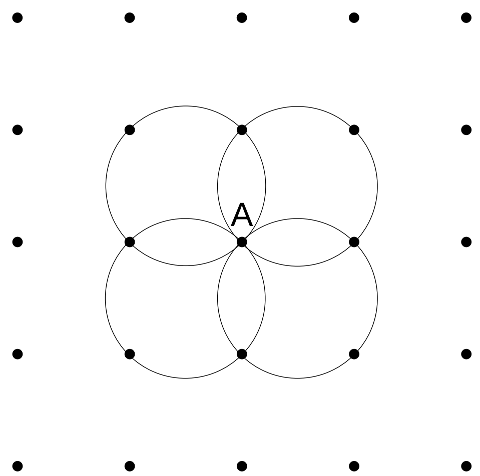
$$\mathbf{u}^h(\mathbf{x}) = \boldsymbol{\alpha} + \boldsymbol{\beta}^T \mathbf{x} = \mathbf{u}(\mathbf{x}), \tag{3.15}$$

which completes the proof. □

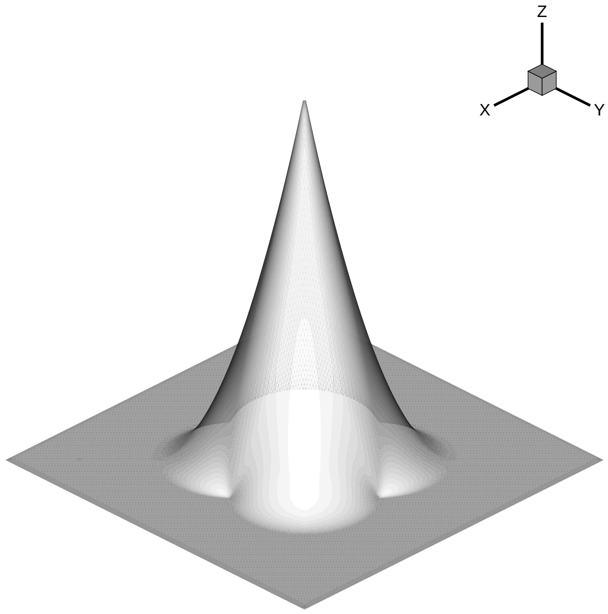
3.3.4 Supports and Natural Neighbors

Consider node $I \in N$, where N is the set consisting of n natural neighbors for a point $\mathbf{x} \in \Omega \subset \mathbb{R}^2$. 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 (Farin, 1990b). In Fig. 3.3a, a unit square is discretized by 25 (5×5) equi-spaced nodes. The support for node A is illustrated in Fig. 3.3b—node A is located at the center where $\phi_A(\mathbf{x}_A)$ takes on the value of unity. The support is clearly seen to be 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 (Watson, 1992).

By comparing the supports of the interpolating functions in NEM to those in some of the widely-used surface approximation schemes, a lot is revealed about the nature of NEM interpolation and its inherent merits. Consider Shepard's interpolant (Shepard, 1968) or moving least squares approximations (Lancaster and Salkauskas, 1981), 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 larger weight at $\mathbf{x}$ than those that are at farther distances.



(a)



(b)

Figure 3.3: Support for NEM shape function. (a) Nodal grid, and (b) Shape function $\phi_A(\mathbf{x})$ for node A.

Natural neighbor 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 (Ahuja, 1982; Gold, 1989). Ahuja (1982) delineates the merits of using Voronoi neighborhoods in the analysis of dot patterns in pattern recognition, while Gold (1989) discusses this property in the context of surface interpolation. An instructive illustration that illuminates the above concept is shown in Fig. 3.4. The number of natural neighbors n is a function of position $\mathbf{x}$ as well as the nodal density. In d-dimensions, the number of natural neighbors is at least $d+1$, with an attainable lower bound (Sibson, 1981). In Fig. 3.5, 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 neighbors. 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 Fig. 3.5. In Fig. 3.6a, an irregular arrangement of nodes is shown, and the associated variation in n is presented as a contour plot in Fig. 3.6b. In this case, one can see the dependence on nodal arrangement and nodal density.

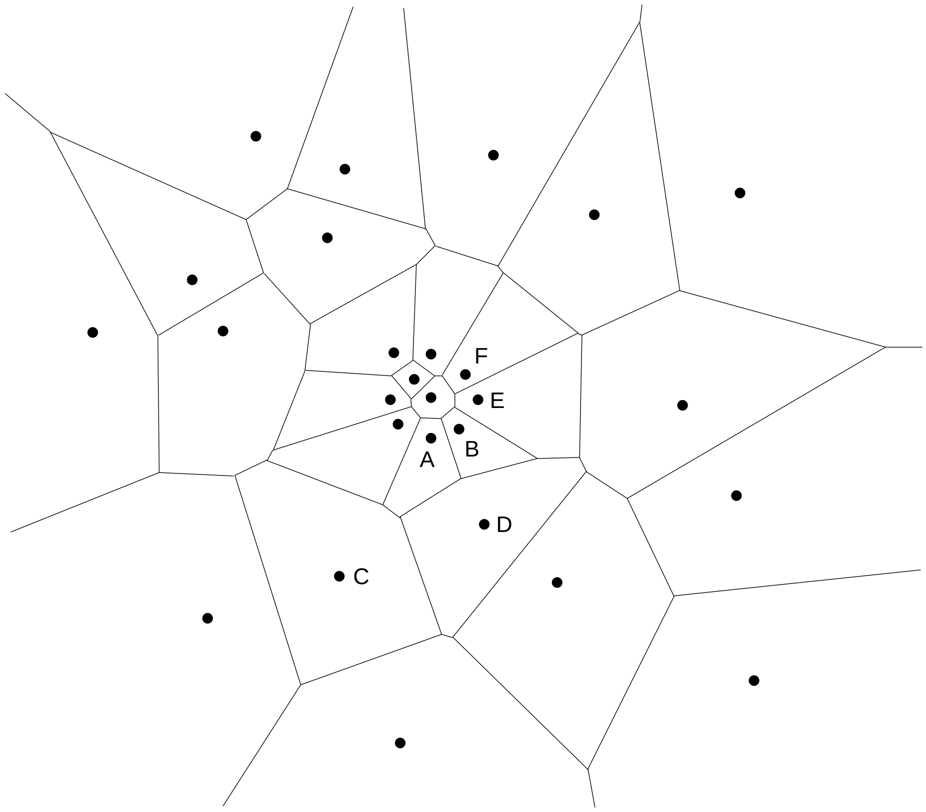


Figure 3.4: Voronoi neighbors. B, C, and D are neighbors of A, but E and F are not, even though $d(A,E) < d(A,D) < d(A,C)$ and $d(A,F) < d(A,D) < d(A,C)$.

3.3.5 Smoothness

The smoothness (or regularity) of the NEM shape functions is discussed. Natural neighbor shape functions are C^∞ everywhere, except at the nodes where they are C^0 (Sibson, 1980; Farin, 1990b; Brown, 1994). Referring to Fig. 3.2, 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

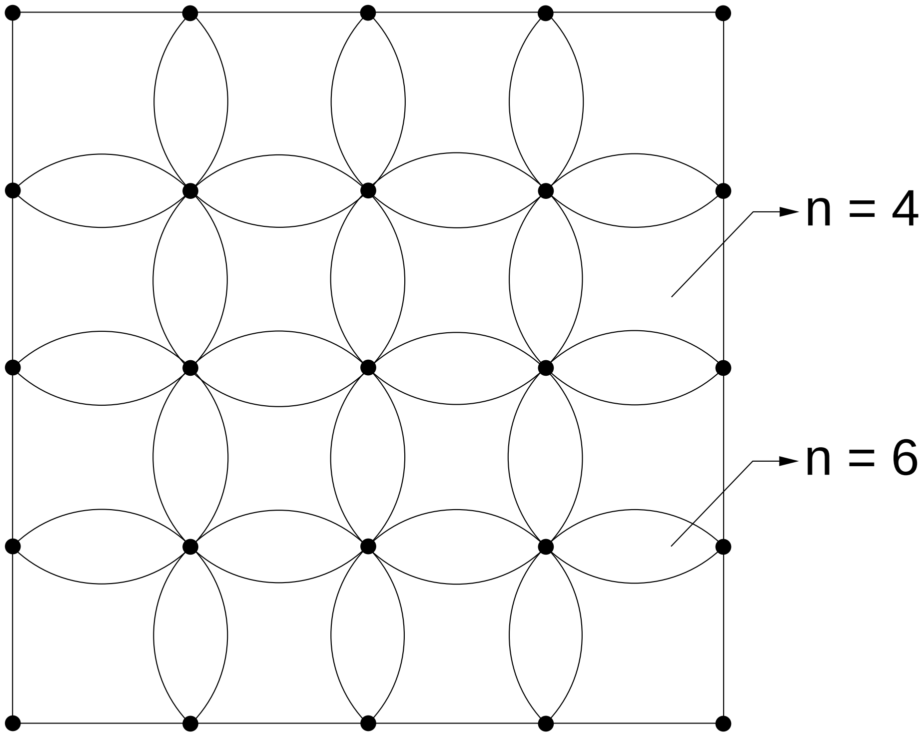
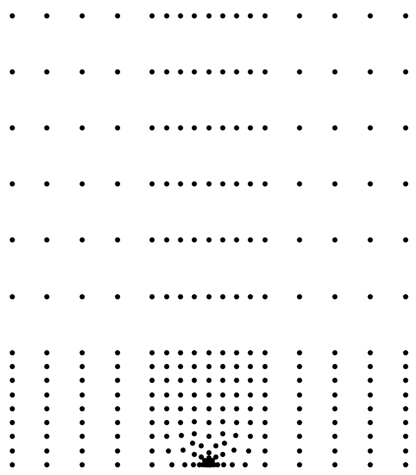
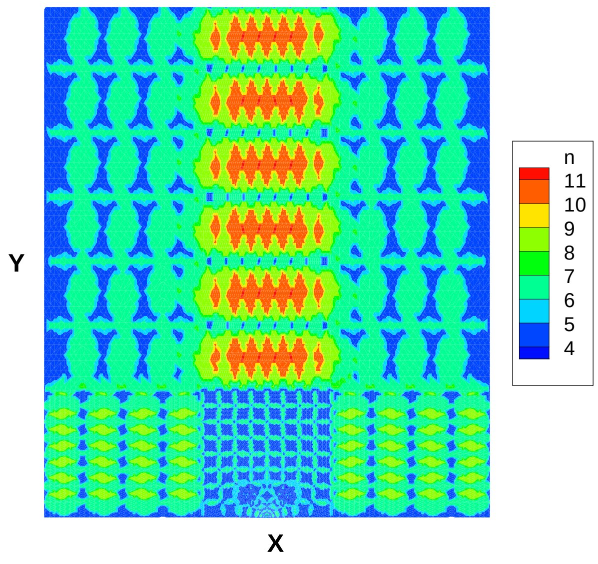


Figure 3.5: Variation of natural neighbors for a regular grid.

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 3.4). Consider a point $\mathbf{x}$ ($\mathbf{x} \neq \mathbf{x}_J$) which lies on a natural neighbor circumcircle of node I such that $\mathbf{x} \in \partial\Omega_{sI}$. Then, only one circum-triangle contributes to the area calculations in Eq. (3.43). 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 (1990b) showed that $\phi_I(\mathbf{x})$ is only C^0 at the nodes. This fact is also seen when one considers the support



(a)



(b)

Figure 3.6: Variation of natural neighbors for an irregular grid. (a) Nodal grid, and (b) Contour plot of natural neighbors.

of the shape functions (Fig. 3.3). 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 $\left(C^1(\Omega)\right)$ forms of natural neighbor interpolation have been proposed by many authors (Sibson, 1981; Farin, 1990b; Traversoni, 1993). Sibson (1981) used a weighted least squares fit to modify the original n-n interpolation scheme (Sibson, 1980) to obtain a $C^1(\Omega)$ interpolant everywhere. In one-dimension, it reduces to a Hermite cubic polynomial. Farin (1990b) proposed a $C^1(\Omega)$ natural neighbor interpolant by embedding Sibson's natural neighbor coordinates in the surface representation of a Bernstein-Bézier cubic simplex; in addition, Farin (1990b) recognized Sibson's interpolant to be a building block for higher order surface schemes, and used Hardy's multiquadric (Hardy, 1971) interpolation scheme as a particular example for such an application. Traversoni and Palacios (1991) reformulated Sibson's interpolant in terms of Bernstein polynomials, and using the notion of covering spheres (Traversoni and Palacios, 1991), 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 (see Chapter 5).

3.3.6 Interpolation in One-Dimension

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

Proof. Consider a 1D bar of length L which is discretized by M unequally spaced nodes (Fig. 3.7a). It is evident that the Voronoi vertices lie at the mid-point between any two adjacent nodes. In Fig. 3.7a, 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 neighbors, while the points on the boundary have only one natural neighbor. 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 coordinate $\xi = (x - x_I)/(x_{I+1} - x_I)$, where $\xi \in [0, 1]$, and a local node numbering system: $n_I \to 1$ and $n_{I+1} \to 2$ (Fig. 3.7b). Consider a point $\xi \in \Omega_I$. The second-order Voronoi cells about ξ are shown in Fig. 3.7b. Using Eq. (3.3), the shape functions can be written as

$$\phi_I(\xi) = \frac{L_{\xi I}}{L_{\xi 1} + L_{\xi 2}} \quad (I = 1,\, 2), \tag{3.16}$$

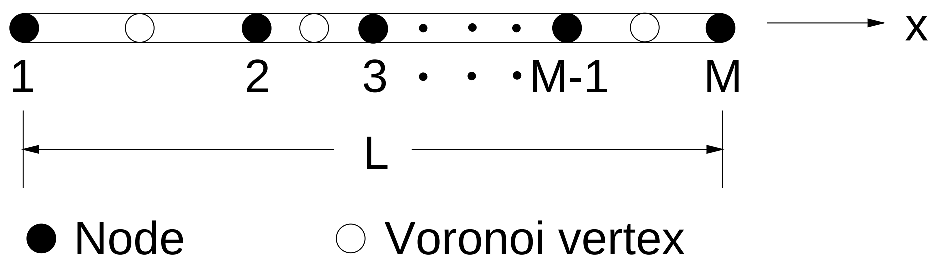
where $L_{\xi 1} = (1 - \xi)/2$ and $L_{\xi 2} = \xi/2$. On using these in the above equation, we obtain the result:

$$\phi_1(\xi) = 1 - \xi, \quad \phi_2(\xi) = \xi, \tag{3.17}$$

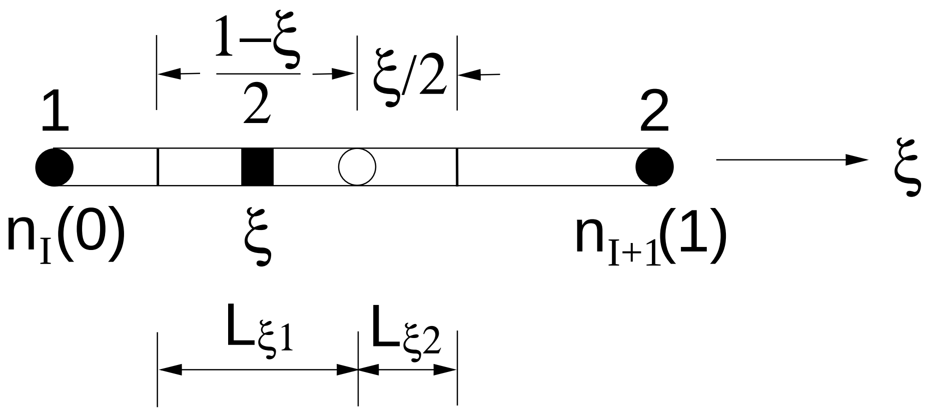
which are precisely 1D linear finite element shape functions. □

3.3.7 Interpolation in Two-Dimensions

The equivalence of NEM shape functions to barycentric coordinates 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 (Farin, 1990b).



(a)



(b)

Figure 3.7: NEM shape functions in one-dimension. (a) Physical space, and (b) Reference space.

Three Natural Neighbors If a point $\mathbf{x}$ has three natural neighbors ($n = 3$), then the NEM shape functions are precisely the barycentric coordinates, 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 (Farin, 1990b). Here we use the linear reproducing conditions to prove the correspondence. Let the natural neighbors of point $\mathbf{x} = (x, y)$ be nodes 1, 2, and 3, with coordinates (x_I, y_I), $I = 1$–3 (Fig. 3.8). Using Eqs. (3.9) and (3.10), the following conditions are met by the NEM shape functions:

$$\sum_{I=1}^{3} \phi_I(\mathbf{x}) = 1, \tag{3.18}$$

$$\sum_{I=1}^{3} \phi_I(\mathbf{x}) x_I = x, \tag{3.19}$$

$$\sum_{I=1}^{3} \phi_I(\mathbf{x}) y_I = y, \tag{3.20}$$

which in matrix form can be written as

$$\begin{bmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{bmatrix} \begin{Bmatrix} \phi_1(\mathbf{x}) \\ \phi_2(\mathbf{x}) \\ \phi_3(\mathbf{x}) \end{Bmatrix} = \begin{Bmatrix} 1 \\ x \\ y \end{Bmatrix}. \tag{3.21}$$

The solution of the above system of linear equations is:

$$\phi_1(\mathbf{x}) = \frac{D_1(\mathbf{x})}{D(\mathbf{x})}, \quad \phi_2(\mathbf{x}) = \frac{D_2(\mathbf{x})}{D(\mathbf{x})}, \quad \phi_3(\mathbf{x}) = \frac{D_3(\mathbf{x})}{D(\mathbf{x})}, \tag{3.22}$$

where

$$D(\mathbf{x}) = \begin{vmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{vmatrix} = \begin{vmatrix} 1 & 0 & 0 \\ x_1 & x_2 - x_1 & x_3 - x_1 \\ y_1 & y_2 - y_1 & y_3 - y_1 \end{vmatrix} = 2A(\mathbf{x}), \tag{3.23}$$

and $D_1(\mathbf{x}) = 2A_1(\mathbf{x})$, $D_2(\mathbf{x}) = 2A_2(\mathbf{x})$, and $D_3(\mathbf{x}) = 2A_3(\mathbf{x})$ (Fig. 3.8). In Eq. (3.23), $A(\mathbf{x})$ is the area of $\triangle_{123}$. Hence we can write the shape functions as

$$\phi_1(\mathbf{x}) = \frac{A_1(\mathbf{x})}{A(\mathbf{x})}, \quad \phi_2(\mathbf{x}) = \frac{A_2(\mathbf{x})}{A(\mathbf{x})}, \quad \phi_3(\mathbf{x}) = \frac{A_3(\mathbf{x})}{A(\mathbf{x})}, \tag{3.24}$$

which are precisely the barycentric coordinates for the point $\mathbf{x}$. $\square$

Four Natural Neighbors (Regular Grid) For a regular rectangular nodal grid, if a point $\mathbf{x}$ has four natural neighbors ($n = 4$), then bilinear interpolation on the rectangle is obtained. A geometric proof is provided by Farin (1990b). Here we use the definition of natural neighbor 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 neighbors that are located at the vertices of a quadrilateral.

Proof. Consider a point $\mathbf{x}$ with four natural neighbors 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 $(x_4, y_4) = (0,1)$ (Fig. 3.9). The first-order (dark line) and second-order Voronoi cells about $\mathbf{x}$ are shown in Fig. 3.9. By definition of the NEM shape functions, we can write

$$\phi_I(\mathbf{x}) = \frac{A_I(\mathbf{x})}{A(\mathbf{x})} \quad (I = 1\text{–}4), \tag{3.25}$$

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 center of the unit square with coordinates $(1/2, 1/2)$.

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 center of the circle that circumscribes the triangle $\triangle_{12\mathbf{x}}$, and proceeding likewise, b is the circumcenter of triangle $\triangle_{23\mathbf{x}}$, c that of $\triangle_{34\mathbf{x}}$, and d that of $\triangle_{41\mathbf{x}}$—see Section 3.4 also. Using Eq. (3.37), these coordinates are computed to be

$$a_1 = \frac{1}{2}, \qquad a_2 = \frac{-x + x^2 + y^2}{2y}, \tag{3.26a}$$

$$b_1 = \frac{1 + y - x^2 - y^2}{2(1-x)}, \qquad b_2 = \frac{1}{2}, \tag{3.26b}$$

$$c_1 = \frac{1}{2}, \qquad c_2 = \frac{1 + x - x^2 - y^2}{2(1-y)}, \tag{3.26c}$$

$$d_1 = \frac{-y + x^2 + y^2}{2x}, \qquad d_2 = \frac{1}{2}. \tag{3.26d}$$

Using the formula for the area of a triangle given in Eq. (3.35), we obtain

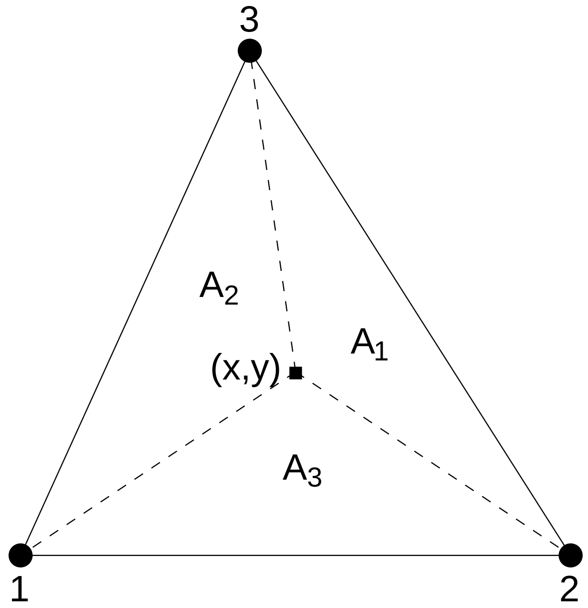


Figure 3.8: Barycentric coordinates ($n = 3$).

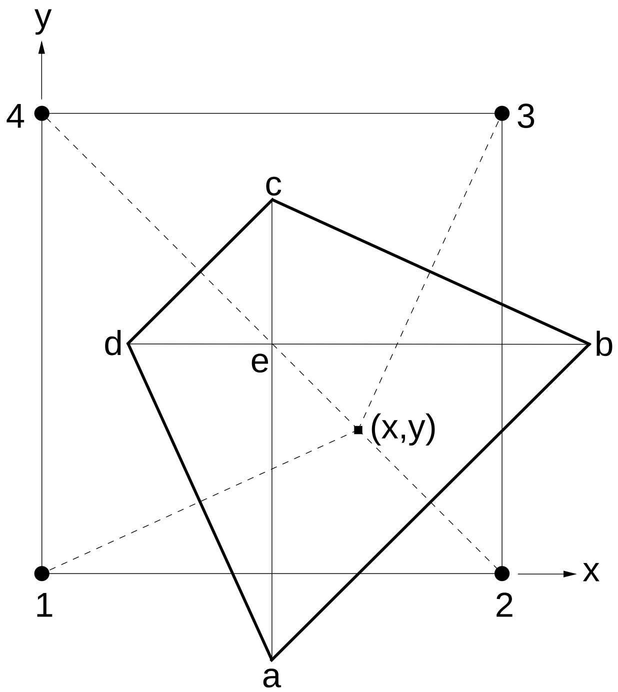


Figure 3.9: Bilinear interpolation on a regular grid ($n = 4$).

$$A_1(\mathbf{x}) = \frac{(x^2 + y^2 - x - y)^2}{8xy}, \tag{3.27a}$$

$$A_2(\mathbf{x}) = \frac{(x^2 + y^2 - x - y)^2}{8y(1-x)}, \tag{3.27b}$$

$$A_3(\mathbf{x}) = \frac{(x^2 + y^2 - x - y)^2}{8(1-x)(1-y)}, \tag{3.27c}$$

$$A_4(\mathbf{x}) = \frac{(x^2 + y^2 - x - y)^2}{8x(1-y)}. \tag{3.27d}$$

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

$$A(\mathbf{x}) = \frac{(x^2 + y^2 - x - y)^2}{8xy(1-x)(1-y)}, \tag{3.28}$$

and hence on using Eqs. (3.27) and (3.28) in Eq. (3.25), the NEM shape functions can be written as

$$\phi_1(\mathbf{x}) = (1-x)(1-y), \tag{3.29a}$$

$$\phi_2(\mathbf{x}) = x(1-y), \tag{3.29b}$$

$$\phi_3(\mathbf{x}) = xy, \tag{3.29c}$$

$$\phi_4(\mathbf{x}) = y(1-x), \tag{3.29d}$$

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. $\square$

3.3.8 Interpolation in Multi-Dimensions

On the basis of earlier discussions and results, it is observed that natural neighbor interpolation in higher-dimensions is a particular generalization of univariate interpolation in one-dimension (Farin, 1990b). 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 1D Lagrange interpolation rules. From a mathematical and geometric viewpoint, the natural-neighbor approach is appealing in more ways than one: a sound mathematical- and geometrical-basis for construction, neighbor-relationships that reflect the spatial adjacency between nodes, and a computationally feasible and attractive choice. In the remainder of this thesis, we show by examples, how these properties of n-n interpolation are also of merit as a paradigm for the numerical solution of PDEs.

3.3.9 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 a convex domain Ω, 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)[2].

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

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 (Fig. 3.10a). For simplicity and for ease of illustration, we assume that ξ has only three natural neighbors, namely nodes 1, 2, and 3. We use a local coordinate 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 in Fig. 3.10b. By definition, the shape functions can be written as

$$\phi_I(\xi) = \frac{A_I(\xi)}{A(\xi)} \quad (I = 1,\ 2,\ 3), \tag{3.30}$$

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

$$A_1(\xi) = \lim_{L\to\infty} L\xi_1 + \delta_1, \quad A_2(\xi) = \lim_{L\to\infty} L\xi_2 + \delta_2, \quad A_3(\xi) = \delta_3, \tag{3.31}$$

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 Fig. 3.10b. On using Eq. (3.30), we can write

$$\phi_1(\xi) = \lim_{L\to\infty} \frac{L(1-\xi)+2\delta_1}{L+2\delta_1+2\delta_2+2\delta_3}, \tag{3.32a}$$

$$\phi_2(\xi) = \lim_{L\to\infty} \frac{L\xi+2\delta_2}{L+2\delta_1+2\delta_2+2\delta_3}, \tag{3.32b}$$

$$\phi_3(\xi) = \lim_{L\to\infty} \frac{2\delta_3}{L+2\delta_1+2\delta_2}. \tag{3.32c}$$

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

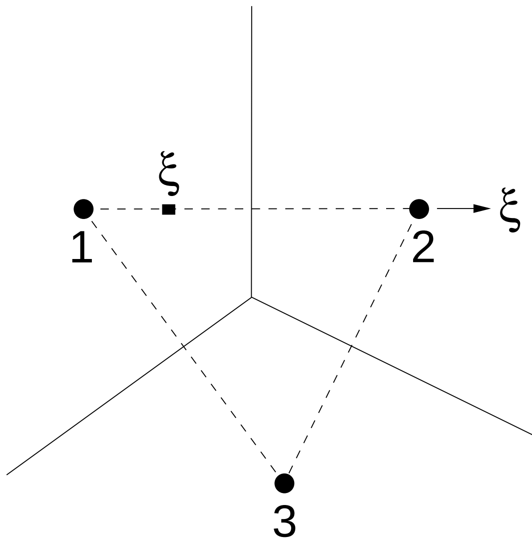
$$\phi_1(\xi) = 1-\xi, \quad \phi_2(\xi) = \xi, \quad \phi_3(\xi) = 0, \tag{3.33}$$

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 neighbors 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 Fig. 3.10b. Using the above equation, the trial functions at the point ξ can be written as

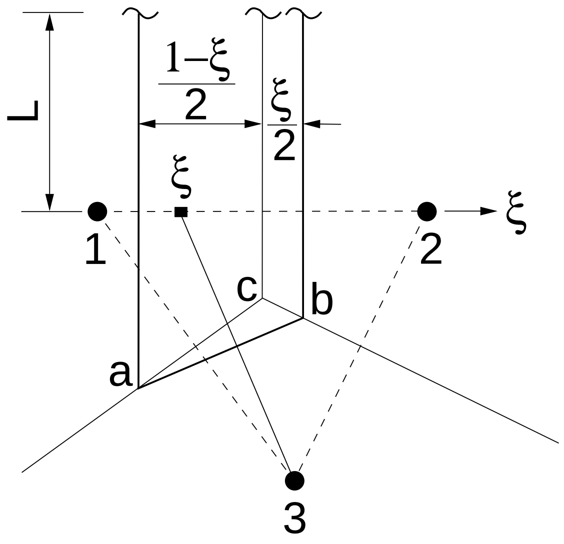
$$\mathbf{u}^h(\xi) = (1-\xi)\mathbf{u}_1 + \xi\mathbf{u}_2 \tag{3.34}$$

which are linear functions, and hence the proof. $\square$

Non-Convex Domains Consider a set of nodes N that describes a non-convex domain Ω. Consider a Γ_1 ($\Gamma_1 \subset \Gamma = \partial\Omega$) which renders the domain to be non-



(a)

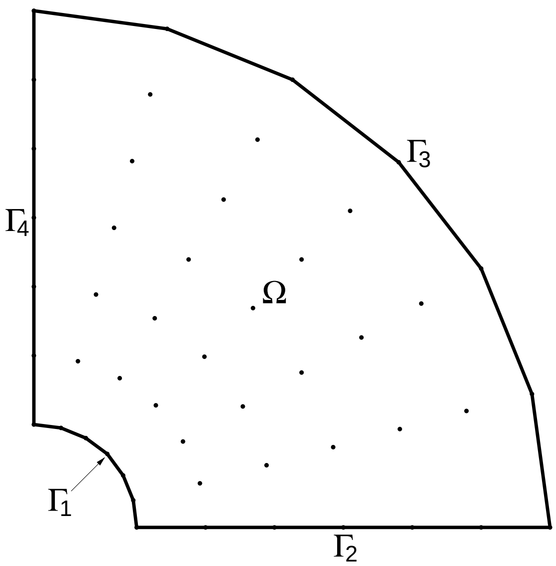


(b)

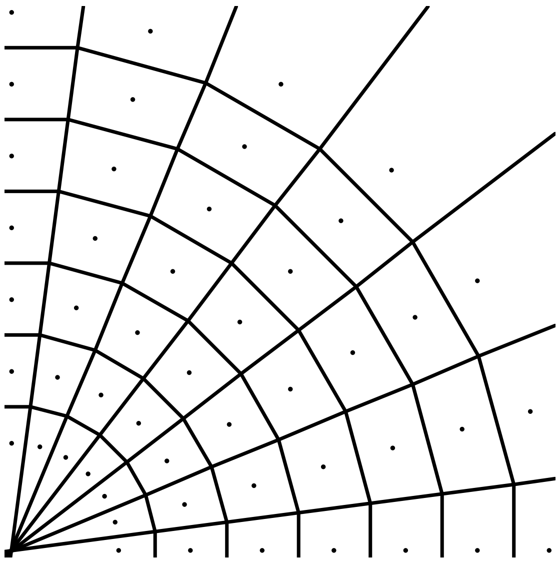
Figure 3.10: Linear behavior of $\mathbf{u}^h(\xi)$ along the boundary of a convex domain. (a) Original Voronoi cells ($n = 3$) and ξ, and (b) $1st$-order and $2nd$-order Voronoi cells about ξ.

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 Fig. 3.11. 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 (Fig. 3.11b). The proof outlined for the case of convex domains holds for $\tilde{\Gamma} = \Gamma_2 \cup \Gamma_3 \cup \Gamma_4$ 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 Fig. 3.11 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% 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 approximation to 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 need to be carried out on a per problem basis to determine the appropriate nodal discretization of Γ_1 which is



(a)



(b)

Figure 3.11: Linear behavior of $\mathbf{u}^h(\mathbf{x})$ along the boundary of a non-convex domain. (a) Nodal discretization, and (b) Voronoi diagram.

required to yield accurate results.

3.4 Numerical Computational Procedure for NEM Shape Functions

In this work, Watson's algorithm (Watson, 1994) is used to compute the natural neighbor shape functions. The algorithmic details and computational procedure is described in Watson (1994) as well as Braun, Sambridge, and McQueen (1995). First, we provide well-known geometric formulas for the area, circumcenter, and circumradius of a triangle, and then go on to present the computational implementation of the shape function algorithm.

3.4.1 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

$$A = \frac{(a_1 - c_1)(b_2 - c_2) - (b_1 - c_1)(a_2 - c_2)}{2}. \tag{3.35}$$

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

$$\begin{aligned} A_{,i}(\mathbf{x}) = {} & \frac{\big(a_1(\mathbf{x}) - c_1\big) b_{2,i}(\mathbf{x}) + \big(b_2(\mathbf{x}) - c_2\big) a_{1,i}(\mathbf{x})}{2} \\ & - \frac{\big(b_1(\mathbf{x}) - c_1\big) a_{2,i}(\mathbf{x}) + \big(a_2(\mathbf{x}) - c_2\big) b_{1,i}(\mathbf{x})}{2}, \end{aligned} \tag{3.36}$$

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

3.4.2 Circumcenter 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 $\mathbf{c} = (c_1, c_2)$ (Fig. 3.12). The circumcircle of $\triangle_{ABC}$ is the circle that passes through the three vertices of the triangle. Its center is the circumcenter $v(v_1, v_2)$ and radius R is the circumradius. The coordinates of the circumcenter are the perpendicular bisectors of the edges of the triangle. The coordinates of the circumcenter are found by either: (a) considering the equation of a plane through three points in an augmented space $\mathbb{R}^3$, 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 coordinates of the circumcenter. By considering the edges AC and BC in Fig. 3.12, 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 circumcenter of $\triangle_{ABC}$:

$$v_1 = \frac{(a_1^2 - c_1^2 + a_2^2 - c_2^2)(b_2 - c_2) - (b_1^2 - c_1^2 + b_2^2 - c_2^2)(a_2 - c_2)}{D}, \tag{3.37a}$$

$$v_2 = \frac{(b_1^2 - c_1^2 + b_2^2 - c_2^2)(a_1 - c_1) - (a_1^2 - c_1^2 + a_2^2 - c_2^2)(b_1 - c_1)}{D}, \tag{3.37b}$$

where D which is four times the area of $\triangle_{ABC}$ $\big($refer to Eq. (3.35)$\big)$, is given by

$$D = 2\big[(a_1 - c_1)(b_2 - c_2) - (b_1 - c_1)(a_2 - c_2)\big]. \tag{3.37c}$$

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 differences in the numerical implementation. For instance, $a_1^2 - c_1^2$ in Eq. (3.37a) 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 coordinates a_1, a_2, b_1, and b_2 are independent of x_1 and x_2. The derivatives of the coordinates of the circumcenter can then be written in the following simplified form:

$$v_{1,1}(\mathbf{x}) = \frac{\big(x_1 - v_1(\mathbf{x})\big) D_{,1}(\mathbf{x})}{D(\mathbf{x})}, \tag{3.38a}$$

$$v_{1,2}(\mathbf{x}) = \frac{\big(\alpha + x_2 D_{,1}(\mathbf{x}) - v_1(\mathbf{x}) D_{,2}(\mathbf{x})\big)}{D(\mathbf{x})}, \tag{3.38b}$$

$$v_{2,1}(\mathbf{x}) = \frac{\big(-\alpha + x_1 D_{,2}(\mathbf{x}) - v_2(\mathbf{x}) D_{,1}(\mathbf{x})\big)}{D(\mathbf{x})}, \tag{3.38c}$$

$$v_{2,2}(\mathbf{x}) = \frac{\big(x_2 - v_2(\mathbf{x})\big) D_{,2}(\mathbf{x})}{D(\mathbf{x})}, \tag{3.38d}$$

where $v_1(\mathbf{x})$, $v_2(\mathbf{x})$, and $D(\mathbf{x})$ are given in Eq. (3.37) and

$$\alpha = (b_1 + a_1)(b_1 - a_1) + (b_2 + a_2)(b_2 - a_2), \tag{3.39a}$$

$$D_{,1}(\mathbf{x}) = 2(a_2 - b_2), \tag{3.39b}$$

and

$$D_{,2}(\mathbf{x}) = 2(b_1 - a_1). \tag{3.39c}$$

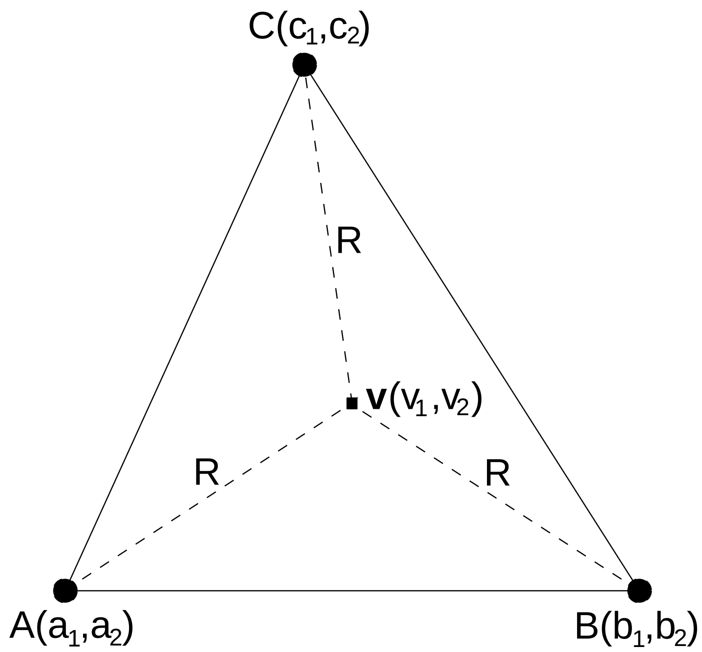


Figure 3.12: Circumcenter and circumradius of a triangle.

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

$$R^2(\mathbf{x}) = \big(a_1 - v_1(\mathbf{x})\big)^2 + \big(a_2 - v_2(\mathbf{x})\big)^2. \tag{3.40}$$

3.4.3 Shape Function Computations

The algorithm proposed by Watson (Watson, 1994) is used to compute the shape functions. In order to present the algorithm and its numerical implementation, we consider Fig. 3.2b, where the construction of natural neighbor shape functions is illustrated. In Fig. 3.13, the original nodal set, the Voronoi diagram (dashed lines), and the natural neighbor circumcircles are shown. The point $\mathbf{x}$ is introduced into the set, and the natural neighbors of $\mathbf{x}$ as well as the values of the natural neighbor shape functions and their derivatives at $\mathbf{x}$ are to be computed.

The computational methodology for the shape function evaluation is based on the fact that the second-order Voronoi cells about $\mathbf{x}$ (Fig. 3.2b) 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 circumcenter and square of the circumradius of the natural neighbor (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, circumcenter, and circumradius of a triangle are presented in Sections 3.4.1 and 3.4.2. In the assembly-step, prior to the shape function computation, the number of natural neighbors n and their global nodal numbers along with the number of triangles t_n associated with the natural neighbors as well as their global triangle numbers, are computed. A simple means to determine if a node is a natural neighbor of a point $\mathbf{x}$ is to use the empty circumcircle criterion indicated in Section 3.1—if the square of the Euclidean distance from $\mathbf{x}$ to the center of a natural neighbor 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 neighbors of the point $\mathbf{x}$:

$$\|\mathbf{v}-\mathbf{x}\|^2 < R^2 , \tag{3.41}$$

where $\mathbf{v} = (v_1, v_2)$ is the center of the natural neighbor circumcircle and R, which is given in Eq. (3.40), is its radius. Braun and Sambridge (1995) used Lawson's algorithm (Lawson, 1977) as part of the neighbor-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 Eq. (3.41) needs to be checked for all the Delaunay triangles. In this work, since relatively few nodes ($M < 3000$) 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 circumcenter $\mathbf{v} = (v_1, v_2)$ that is selected on the basis of Eq. (3.41), 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 circumcenter of the new triangles so constructed and their derivatives in the following compact form:

$$\mathbf{c}_i(\mathbf{x}) = \Theta(\mathbf{a}_j, \mathbf{a}_k, \mathbf{x}), \tag{3.42a}$$

$$\mathbf{c}_{i,m}(\mathbf{x}) = \Theta_{,m}(\mathbf{a}_j, \mathbf{a}_k, \mathbf{x}), \tag{3.42b}$$

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 circumcenter of a triangle and its derivatives which are given in Eqs. (3.37) and (3.38), respectively. To elaborate, $\mathbf{c}_1$ refers to the circumcenter 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 circumcenter of the triangle with vertices $(1, 2, \mathbf{x})$.

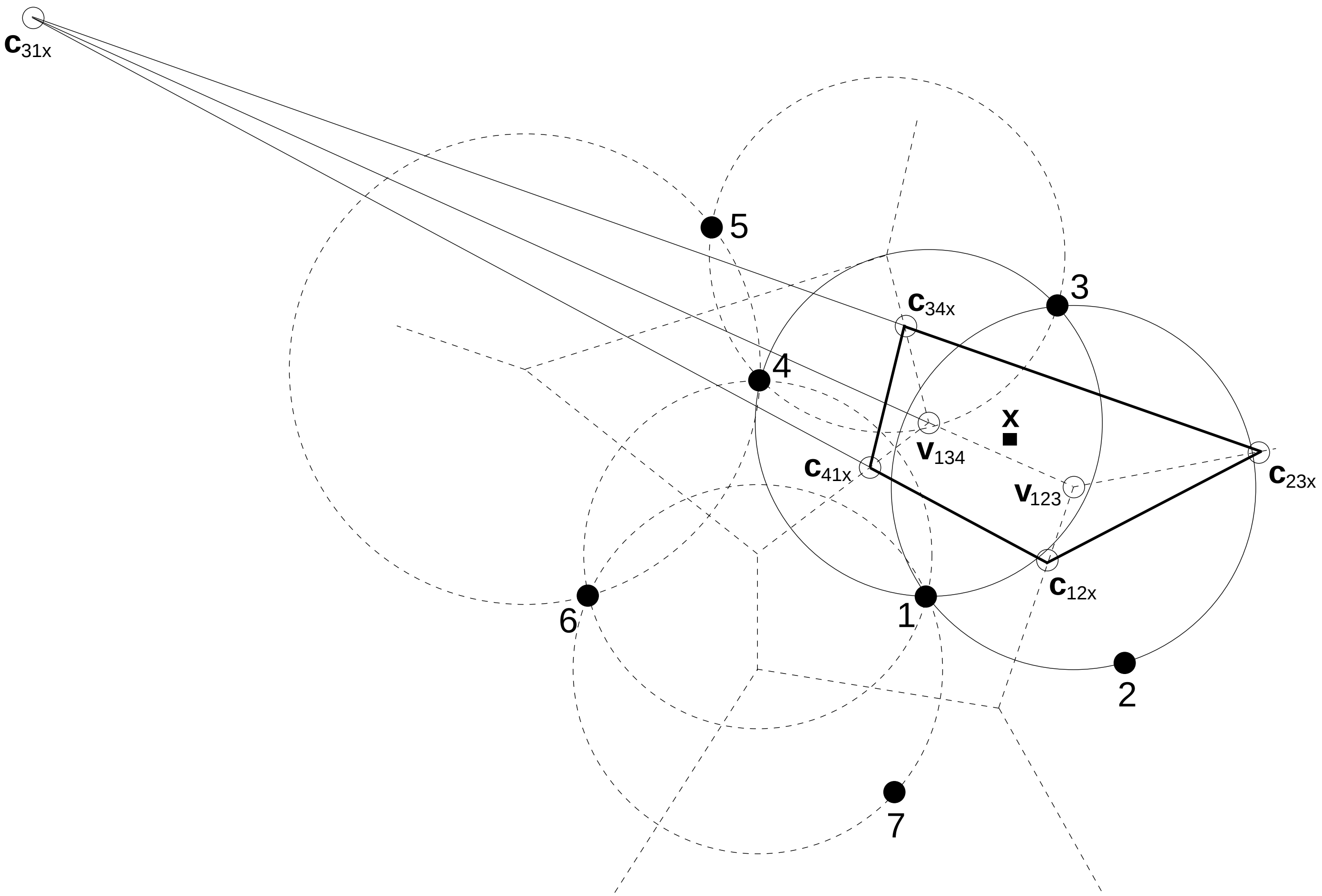


Figure 3.13: Computation of natural neighbor shape functions.

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

$$\alpha_{it}(\mathbf{x}) = \Psi\big(\mathbf{c}_j(\mathbf{x}), \mathbf{c}_k(\mathbf{x}), \mathbf{v}\big), \tag{3.43a}$$

$$\alpha_{it,m}(\mathbf{x}) = \Psi_{,m}\big(\mathbf{c}_j(\mathbf{x}), \mathbf{c}_k(\mathbf{x}), \mathbf{v}\big), \tag{3.43b}$$

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 Eqs. (3.35) and (3.36), 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:

$$\beta_I(\mathbf{x}) \leftarrow \beta_I(\mathbf{x}) + \alpha_{it}(\mathbf{x}), \tag{3.44a}$$

$$\beta_{I,m}(\mathbf{x}) \leftarrow \beta_{I,m}(\mathbf{x}) + \alpha_{it,m}(\mathbf{x}), \tag{3.44b}$$

where $\beta_I(\mathbf{x})$ and $\beta_{I,m}(\mathbf{x})$ are set to zero for all natural neighbors 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

$$A_I(\mathbf{x}) = \beta_I(\mathbf{x}), \tag{3.45a}$$

$$A_{I,m}(\mathbf{x}) = \beta_{I,m}(\mathbf{x}), \tag{3.45b}$$

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

$$A(\mathbf{x}) = \sum_{I=1}^{n} A_I(\mathbf{x}), \tag{3.46a}$$

$$A_{,m}(\mathbf{x}) = \sum_{I=1}^{n} A_{I,m}(\mathbf{x}). \tag{3.46b}$$

We reproduce Eqs. (3.3) and (3.5) below for the natural neighbor shape functions and their derivatives:

$$\phi_I(\mathbf{x}) = \frac{A_I(\mathbf{x})}{A(\mathbf{x})}, \tag{3.47a}$$

$$\phi_{I,m}(\mathbf{x}) = \frac{A_{I,m}(\mathbf{x}) - \phi_I(\mathbf{x})A_{,m}(\mathbf{x})}{A(\mathbf{x})}. \tag{3.47b}$$

Substituting for $A_I(\mathbf{x})$, $A_{I,m}(\mathbf{x})$, $A(\mathbf{x})$, and $A_{,m}(\mathbf{x})$ from Eqs. (3.45a), (3.45b), (3.46a), and (3.46b), 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 neighbors. A pseudo-code for the shape function computations is presented in Table 3.1.

We use Fig. 3.13 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 circumcenter of a triangle also serve as a label for the point. It is seen in Fig. 3.13 that the point $\mathbf{x}$ lies within only two circles (dark lines), namely $\mathbf{v}_{123}$ and $\mathbf{v}_{134}$ which

1. Determine natural neighbors I ($I = 1 \to n$) and Delaunay circum-triangles t ($t = 1 \to t_n$) for $\mathbf{x}$—see Eq. (3.41);
2. Initialize $\phi_I = d\phi_{I1} = d\phi_{I2} = 0 \; \forall \; I$; $A = dA_1 = dA_2 = 0$;
3. for $t = 1 \to t_n$ {
   - set $\mathbf{v}$ = circumcenter of $\triangle_t$—see Eq. (3.37);
   - for $i = 1 \to 3$ {
     - set j and k using cyclic order;
     - compute $\mathbf{c}_i$ and $\mathbf{c}_{i,m}$ based on Eq. (3.42) and store;

     }
   - for $i = 1 \to 3$ {
     - set g = global node number of local node i;
     - set I = position of g in the natural neighbor nodal array ($1 \leq I \leq n$);
     - compute α_{it} and $\alpha_{it,m}$ based on Eq. (3.43);
     - 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 $I = 1 \to n$ {
   - set $\phi_I \leftarrow \dfrac{\phi_I}{A}$, $d\phi_{I1} \leftarrow \dfrac{d\phi_{I1} - \phi_I dA_1}{A}$, $d\phi_{I2} \leftarrow \dfrac{d\phi_{I2} - \phi_I dA_2}{A}$;

   }

Table 3.1: Pseudo-code for natural neighbor shape function computations at $\mathbf{x}$.

are the position vectors for the the circumcenter (open circle) of the natural neighbor circumcircles. Hence the natural neighbors 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 circumcenters (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} $(i = 1–3)$ in Eq. (3.43). 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 $\to$ global nodal associations, we obtain the following results:

$$A_1(\mathbf{x}) = \Psi(\mathbf{c}_{31\mathbf{x}}, \mathbf{c}_{12\mathbf{x}}, \mathbf{v}_{123}) + \Psi(\mathbf{c}_{41\mathbf{x}}, \mathbf{c}_{31\mathbf{x}}, \mathbf{v}_{134}) \equiv \text{Area}\,(\Box_{12\mathbf{x}-123-134-41\mathbf{x}}), \tag{3.48a}$$

$$A_2(\mathbf{x}) = \Psi(\mathbf{c}_{12\mathbf{x}}, \mathbf{c}_{23\mathbf{x}}, \mathbf{v}_{123}) \equiv \text{Area}\,(\triangle_{12\mathbf{x}-23\mathbf{x}-123}), \tag{3.48b}$$

$$A_3(\mathbf{x}) = \Psi(\mathbf{c}_{23\mathbf{x}}, \mathbf{c}_{31\mathbf{x}}, \mathbf{v}_{123}) + \Psi(\mathbf{c}_{13\mathbf{x}}, \mathbf{c}_{34\mathbf{x}}, \mathbf{v}_{134}) \equiv \text{Area}\,(\Box_{134-123-23\mathbf{x}-34\mathbf{x}}), \tag{3.48c}$$

$$A_4(\mathbf{x}) = \Psi(\mathbf{c}_{34\mathbf{x}}, \mathbf{c}_{41\mathbf{x}}, \mathbf{v}_{134}) \equiv \text{Area}\,(\triangle_{34\mathbf{x}-41\mathbf{x}-134}), \tag{3.48d}$$

$$A(\mathbf{x}) = \sum_{I=1}^{4} A_I(\mathbf{x}), \tag{3.48e}$$

$$\phi_I(\mathbf{x}) = \frac{A_I(\mathbf{x})}{A(\mathbf{x})} \quad (I = 1,\ 2,\ 3,\ 4). \tag{3.48f}$$

It is immediately seen that the areas computed for each node in Fig. 3.13 are precisely those seen in Figures 3.2b and 3.13. 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 (Watson, 1994) 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 circumcenter ($R(\mathbf{x}) = \pm\infty$). 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 straightforward approach is to compute the area of the polygons ($2nd$-order Voronoi cells) directly. Since the vertices of the polygon are the circumcenter of triangles formed from the natural neighbor nodal set and the point $\mathbf{x}$, one can use the algorithm mentioned in O'Rourke (1994) 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 (Lasserre, 1983) for volume of convex polytopes in R^d is based on solving a constrained (bounded hyperplanes) linear programming problem to compute the volume of the polytopes. Braun and Sambridge (1995) used this algorithm to compute the natural neighbor shape function in their PDE application. This approach is robust, and is applicable at any point in the domain for shape function computations.

3.4.4 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, in addition to 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 $\mathbf{K}$. All other steps are common to both, and hence we focus out 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 3.1, the optimum time-complexity of triangulation algorithms in 2D 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 neighbors for a sampling point $\mathbf{x}$. 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 (Lawson, 1977), which is a $\mathcal{O}(n)$ algorithm, is a suitable choice.

3. The numerical computations of the NEM shape functions is outlined in Section 3.4. 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 $\mathbf{K}$ 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 $\mathbf{K}$. In NEM, since the number of neighbors for a point $\mathbf{x}$ is fairly small ($n_{max} = 11$ for the irregular grid in Fig. 3.6), the nodal-based assembly is not overly time-intensive.

On the basis of the numerical results that are presented in Section 4.4, 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.

3.5 NEM, FEM, and Meshless Methods

We compare and contrast the natural element method to the finite element method and meshless methods. The local interpolant in FEM is constructed on the basis of an element structure, and in most meshless methods the local approximation is based on weight functions of compact support. The NEM interpolant uses a normalized Lebesgue measure that takes into account the distribution of nodes and their relation to each other in space—it has a very distinct meshless character for one can associate an anisotropic weight function of compact support with its construction. Interpolation to nodal data is realized in FEM and NEM which facilitates ease of imposing essential boundary conditions; in Galerkin methods that use meshless approximations, typically Lagrange multipliers, modified variational principles, or coupling to finite elements is adopted to enforce essential boundary conditions. In one-dimension, the natural element method is identical to linear finite elements, but no such precise association with finite elements exists for other meshless methods. All these Galerkin methods possess the property of linear completeness which is a sufficient condition for convergence in the application to second-order elliptic PDEs. The distinction lies only in the manner in which the linear completeness conditions are enforced or obtained. In meshless methods, the approximation spaces beget the continuity of the weight function for smooth problems and hence higher-order continuous (C^k, $k > 0$) trial functions are readily constructed; traditional FEM are C^0, and NEM is smooth everywhere except at the nodes where it is C^0. The local approximation spaces in finite elements are uniformly independent (Strang and Fix, 1973). On the other hand, irregular nodal arrangements in meshless approximations such as EFG do result in marked deterioration in the properties of the approximation space. This is so since isotropic weight functions are noticeably biased towards regions of higher nodal density, at the expense of equi-directionality which is contended to be an essential ingredient for good approximability. The density and spatial location of nodes are taken into account in the construction of the natural neighbor interpolant. This allows for a natural means to construct stable approximation spaces that are based on irregular nodal arrangements. In this work, 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 Delaunay triangles for the convergence of the method (Babuška and Aziz, 1976), no such constraints are imposed on the shape, size, and angles of the Delaunay triangles in NEM. Hence we see that NEM does bear similarities to both

FEM and meshless methods. If FEM and meshless methods such as EFG are deemed to lie at the opposite ends of a spectrum, one would be inclined and justified to place NEM in the middle.