

Computational Methods for Natural Neighbour Interpolation in Two and Three Dimensions

M. Sambridge

Research School of Earth Sciences and Centre for Information Science Research, Australian National University, Canberra, ACT 0200, Australia

J. Braun, H. McQueen

Research School of Earth Sciences, Australian National University, Canberra, ACT 0200, Australia

1. Introduction

To interpolate a scalar field specified at a series of ‘reference points’ with irregular distribution in a space of arbitrary dimension is a difficult problem encountered in many areas of fundamental and applied sciences. For example, physical measurements are often made at irregularly spaced locations and need to be interpolated onto a rectangular mesh for processing and analysis. An efficient interpolation method is also necessary to parameterize models which include physical processes at many different scales.

We describe an interpolation method which makes use of some powerful algorithms from the field of computational geometry. Our implementation of the method is efficient and may therefore be used to interpolate large datasets in 2-D or 3-D and to parameterize complex models. We discuss applications of this method to interpolation and to estimation of spatial derivatives for the numerical solution of partial differential equations (PDEs). Our objective is to draw the attention of a wider community to the use of natural neighbours in a range of problems and their current application in geophysics.

2. Voronoi Cells and Delaunay Tessellations

Over the past decade fundamental geometric constructs known as *Voronoi diagrams* and *Delaunay tessellations* have received a large amount of attention (see Okabe, Boots and Sugihara [12] for a review). In 2-D the Voronoi diagram for an irregular set of nodes divides the plane into a set of regions, one for each node, such that all points in a particular region are closer to its node than to any other node. Figure 1a shows the Voronoi diagram for a set of 16 nodes. Each region, or cell, consists of the part of the plane nearest to that node. The cells are unique, space filling and can be defined similarly in any number of dimensions. The Voronoi diagram is regarded as ‘one of the most fundamental and important geometrical constructs determined by an irregular set of points’ (Avis and Bhattacharya [2]).

The other closely related geometric construct which is of interest here is the Delaunay tessellation (or triangulation in 2-D), which was introduced by Voronoi [16] and extended by Delaunay [6]. The Delaunay triangulation of the sixteen nodes in Figure 1a is shown in Figure 1b. The Delaunay triangles are formed by simply connect-

ing the nodes whose Voronoi cells have common boundaries. Again these concepts generalize easily to higher dimensions, e.g. in 3-D we have Delaunay tetrahedra.

Voronoi cells and Delaunay triangles are said to be ‘dual’ to one another, and once one is known the other is defined. Delaunay triangles are of interest because of their useful properties. In a sense they provide the ‘best looking’ set of triangles. By this we mean that they are the set of least ‘long and thin’ triangles that can be generated among the many triangulations that are possible with irregularly distributed points. This is often referred to as the *maximum-minimum angle property* and can be used as the basis of a method for their calculation (Fortune [8]).

Another useful property is that the size of the Delaunay triangles is strongly determined by the density of the nodal distribution. This property is derived directly from the Voronoi cells, whose areas can be used as an inverse measure of the nodal density. Therefore the size of Delaunay triangles will vary enormously when the nodal distribution is highly irregular. This property can be exploited in the development of a Lagrangian finite element method for solving PDEs on an evolving mesh (see section 8).

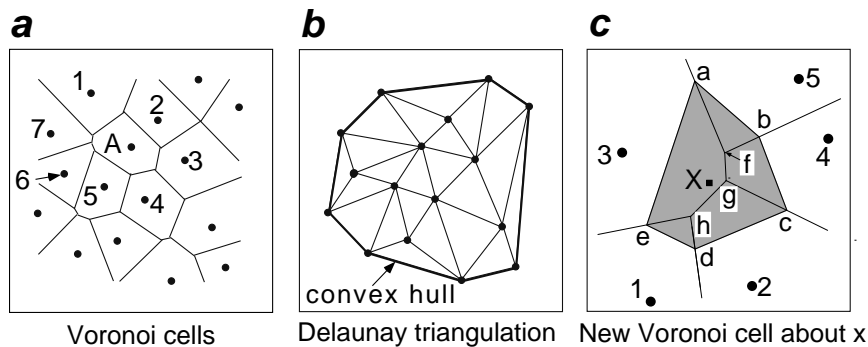


FIGURE 1. a) A set of randomly distributed nodes and their Voronoi cells in a plane. The natural neighbours of A are numbered 1 to 7. b) The Delaunay triangulation is obtained by connecting all pairs of natural neighbours together. c) A test point x and its five natural neighbours. The shaded area is the Voronoi cell about x .

3. Methods for Calculating Delaunay Triangulations

During the past decade a number of algorithms have been devised to calculate Delaunay tessellations in 2-D and 3-D. Early 2-D algorithms, suitable for computer programming are by Lawson [11] and Green and Sibson [9]. A comprehensive list can be found in the review articles by Aurenhammer [1], Fortune [8], and Okabe, Boots and Sugihara [12]. There are several types of algorithm each of which may be suitable for different applications. The ‘edge flipping’ algorithm (see Fortune [8]) starts with an existing non-Delaunay triangulation and uses the maximum-minimum angle property to iteratively update all connections between nodes until they are in a Delaunay configuration. This method can be very useful when an existing Delaunay triangulation in 2-D is known and one wishes to re-calculate the Delaunay connections

after perturbing the position of all of the nodes.

Other types of algorithm include ‘incremental’ methods which add nodes sequentially and update the Delaunay triangulation after each addition. Often methods are developed specifically for the 2-D case and will not work, or quickly become inefficient, in higher dimensions. In 2-D, the method generally accepted as the most efficient is the ‘sweep-line’ algorithm of Fortune [7]. In three and higher dimensions, we found the *qhull* algorithm of Barber and Huhdanpaa [3] to be the most efficient.

4. Natural Neighbours

The idea of a set of ‘nearby surrounding nodes’ is generalized by the definition of *natural neighbour* nodes. The natural neighbours of any node are those in the neighbouring Voronoi cells, or equivalently, those to which the node is connected by the sides of Delaunay triangles. For example in Figure 1a, node A has natural neighbours numbered 1 to 7. (For a thorough discussion of natural neighbours see Watson [17, 18]). Although natural neighbours usually refer to the nodes, one can equally well define a set of natural neighbours to any (x, y) point in the plane, as that set of nodes which would be connected to the point if it were added to the Delaunay triangulation. Again these definitions generalize in a straightforward way to higher dimensions. The importance of natural neighbours is that they represent a set of ‘closest surrounding nodes’ whose number and position is well defined and varies according to the local nodal distribution.

One can think of the natural neighbours about any point as a unique set of nodes that define the ‘neighbourhood’ of the point in the plane. If the distance between nodes is relatively large in some parts, or the distribution is highly anisotropic, then the set of natural neighbours will reflect these features, but nevertheless still represent the best set of nearby surrounding nodes. They are therefore ideal candidates for the basis of a local interpolation scheme, i.e. we have,

$$f(x, y) = \sum_{i=1}^n \phi_i(x, y) f_i \quad (4.1)$$

where, $f(x, y)$ is the interpolated function value, $f_i (i = 1, \dots, n)$ are the data values at the n natural neighbour nodes to the point (x, y) , and the $\phi_i (i = 1, \dots, n)$ are weights associated with each node. The way in which the weights, ϕ_i , are determined will control the smoothness properties of the interpolation. However, since the summation in (4.1) is only over the natural neighbour nodes then regardless of how the ϕ_i are determined, the interpolation is guaranteed to be local. Furthermore the size and shape of the region that can influence any point will adapt naturally to the local variation in node density.

5. Natural Neighbour Interpolation

In natural neighbour interpolation the weights, ϕ_i , in eqn.(4.1) are taken as the *natural neighbour co-ordinates* of the point (x, y) in the plane. Natural neighbour co-ordinates were introduced by Sibson [14, 15] and may be defined in any number

of dimensions. They have a straightforward geometric definition which is most easily explained in 2-D. Consider the five nodes in Figure 1c and the sides of the Voronoi cells. If one adds a point X in cell 3, then a new Voronoi cell can be placed around it (the shaded region in Figure 1c). The Voronoi cell about X overlaps all of the original cells of its natural neighbours. The natural neighbour co-ordinate of X with respect to a neighbour is defined as the ratio of the area of their overlapping Voronoi cells to the total area of the Voronoi cell about X . For example, in Figure 1c the natural neighbour co-ordinate of X with respect to node 3, is the ratio of the areas of the polygon $afghe$ to the polygon $abcde$. The five ‘overlapping regions’ in Figure 1c are known as *second-order* Voronoi cells. The definition of a second-order Voronoi cell between nodes i and j is the region within which all points are closest to node i and second-closest to node j . In Figure 1c the natural neighbour co-ordinates of X are the normalized areas of the five second-order Voronoi cells of X and nodes 1 to 5.

Since natural neighbour co-ordinates are always normalized areas (or volumes in 3-D) we immediately have $0 \leq \phi_i(x, y) \leq 1$ and $\sum_i \phi_i = 1$. Clearly if X were placed on top of any node, say node 3, then the boundaries of the (shaded) Voronoi cell would become identical to that of the original Voronoi cell and so there would be total overlap with cell 3 and zero overlap with all other nodes. From this it directly follows that,

$$\phi_i(\mathbf{x}_j) = \delta_{ij} \quad (5.1)$$

where the vector \mathbf{x}_j is the position of node j . Eqn. (5.1) also holds in 3-D, and means that the interpolated value $f(\mathbf{x})$ at any point, \mathbf{x} , has the useful property of being exactly equal to the original function value at the nodes, i.e. we have,

$$f(\mathbf{x}_i) = f_i. \quad (5.2)$$

The second important property of natural neighbour interpolation is that it is a local procedure. One can think of the natural neighbour co-ordinate, $\phi_i(\mathbf{x})$ as a function describing how node i influences the region around it. It turns out that $\phi_i(\mathbf{x})$ is only non-zero in the union of the n_t circles (or spheres) that pass through the vertices of the n_t Delaunay triangles (or tetrahedra) about node i . In 2-D $n_t = n$, where n is the number of natural neighbours of node i .

The third important property of natural neighbour co-ordinates is that they are continuously differentiable at all points except at their defining node, i (Sibson [14]). This means that the interpolated function, $f(\mathbf{x})$ given by (4.1) is also continuously differentiable everywhere except at the nodes. In our experience the lack of differentiability at the nodes is rarely a problem in practice, and natural neighbour interpolation does not have the limitation, seen in linear interpolation, of discontinuous derivatives across cell boundaries.

In summary natural neighbour interpolation results in an interpolant which fits the original data at the nodes exactly, is local, and guarantees continuity in first derivatives everywhere except at the nodes. Furthermore the size and shape of the local region varies according to the local nodal density distribution, by virtue of

its definition using natural neighbours. These properties hold in any number of dimensions.

6. Natural Neighbour Co-ordinates and their Derivatives in 3-D

In any number of dimensions natural neighbour co-ordinates are defined as the ratio of the volume of a second-order Voronoi cell to the volume of the first-order Voronoi cell about a point \mathbf{x}_p . Since the sum of the second-order cells is equal to the first-order cell, then to calculate Natural Neighbour co-ordinates we need only determine the volume of each second-order Voronoi cell. Watson [18] introduced a method to do this for the 2-D case. Here we describe a new method that can be generalized to any number of dimensions.

The new approach is based on the fact that all second-order Voronoi cells are convex polyhedra (or convex polytopes in higher dimensions). This means that all points inside a second-order Voronoi cell (in dimension n) satisfy a system of linear inequality constraints,

$$\mathbf{Ax} \leq \mathbf{b} \quad (6.1)$$

where the matrix A has $M + 1$ rows and N columns, and $M + 1$ is the number of boundaries (sides) of the cell. By definition the first bounding plane is the perpendicular bisector of \mathbf{x}_p and one of its neighbours, which we shall call \mathbf{x}_0 , and the M remaining are the bisectors of \mathbf{x}_0 and the nodes which are neighbours of both \mathbf{x}_p and \mathbf{x}_0 . In 2-D these facts can easily be verified from Figure 1c, and they readily generalize to any number of dimensions.

A recursive formula for calculating the volume of any convex polyhedron in any number of dimensions was given by Lasserre [10],

$$V(N, A, \mathbf{b}) = \frac{1}{N} \sum_{i=0}^M \frac{b_i}{|a_{it}|} V'_{it}(N - 1, \tilde{A}_{i,t}, \tilde{\mathbf{b}}_t) \quad (6.2)$$

where $\tilde{A}_{i,t}$ is the reduced matrix obtained from A by eliminating the t -th variable using the i -th equation, $\tilde{\mathbf{b}}_t$ is the corresponding reduced vector and a_{it} is the t -th element of A . The value of t is usually chosen so that $|a_{it}|$ is a maximum. The recursive evaluation of (6.2) is best viewed as a tree structure, with V at the root and first $M + 1$ terms as branches attached to V . In the second level of the recursion each reduced matrix and vector correspond to a system of inequalities with one less constraint and one less variable. Therefore each branch is attached to M other branches, and so on for each level. After $N - 1$ recursions we reach the leaf nodes, and for each the system of inequalities is reduced to a set of 1-D constraints. In each case the length of the region satisfying all constraints becomes the final contribution to recursive formula in (6.2).

For any given point \mathbf{x}_p the matrix A and vector \mathbf{b} can be easily determined for each second-order Voronoi cell, and the recursive algorithm leads to an efficient evaluation of their volumes. The accuracy of the second-order Voronoi volumes can be independently verified by comparing their sum to the volume of the first-order cell. We have

done this for a test set of 220 nodes in 3-D and in each case the only discrepancies were of the order of machine epsilon reflecting floating point roundoff error.

A differentiation of V in (6.2) leads, after some algebra, to a new recursive expression for each derivative, whose computation takes about the same time as the volume itself. The situation is simplified by the fact that only the first row of A and \mathbf{b} depend upon \mathbf{x}_p . However, this property is lost for the $i = 0$ term in (6.2). It turns out that for these, and all lower branches of the recursive tree, all elements of the reduced matrix \tilde{A}_{it} and vector $\tilde{\mathbf{b}}_t$ become functions of \mathbf{x}_p . With the aid of some extensive algebra it is possible to keep track of the dependent terms and calculate all derivatives. The validity of the resulting derivative formulae have been verified by comparison to finite difference estimates for a set of irregularly distributed nodes in up to four dimensions.

A more detailed description of this new method to calculate natural neighbour coordinates and their derivatives can be obtained from Braun and Sambridge [5].

7. An Example of Natural Neighbour Interpolation

We have used natural neighbour interpolation to interpolate a large dataset of 688,475 elevation measurements distributed along east-west trending flight lines. The average spacing between data points along flight lines is two orders of magnitude smaller than the flight line spacing. The dataset was interpolated on a 1152×900 rectangular mesh using the recursive formula (6.2). The computation performed on a Sun Sparc 10/51 workstation took 4611 CPU seconds (1 hr 17 min) to complete. It included reading the ASCII dataset from disk, constructing the Delaunay tessellation and, for each point of the rectangular mesh, finding in which Delaunay triangle it is located and computing its natural neighbour co-ordinates. All computations were done in core which required 99Mb of memory.

The result of the interpolation shown in Figure 2 is a smooth topographic surface that passes exactly through all topographic measurements. Notice how the interpolation method is not affected by the strong directional asymmetry in the original dataset.

8. Application to Finite Element Methods

Another area where natural-neighbour interpolation has applications is in the numerical solution of partial differential equations (PDEs). Most numerical methods approximate the solution of PDEs at a series of nodes, within a 2-D or 3-D domain. Two important features are the choice of the spatial distribution of nodes, and the choice of interpolation method used to represent the variation of the solution and its spatial derivatives between the nodes. Delaunay triangulations and natural neighbour interpolation provide an excellent basis for these, and offer several advantages over other methods. For example, in a Lagrangian formulation the mesh ‘follows’, or evolves with, the solution and the distribution of nodes can quickly become highly irregular. In this case, Braun and Sambridge [4] have shown that Delaunay triangles can be used to update, at each time step, the connections between the nodes of the numerical mesh. They give an example of this in crustal deformation which uses

linear interpolation in Delaunay triangles.

By replacing linear interpolation with natural neighbour interpolation it is possible to combine the evolving mesh with a smoother and more flexible interpolant applicable to a wider class of problems. Braun and Sambridge [5] have shown that the natural neighbour interpolant leads to a powerful new class of weighted residual methods for solving PDEs, which they call the *Natural Element Method*. To demonstrate the method they applied the approach to a problem involving the coupling between elasto-plastic deformation and flow in a viscous fluid. However, because of its generality it seems likely that it will find applications in many other problems.

9. Discussion

We have briefly described an efficient method to compute natural neighbour co-ordinates from an irregular distribution of points in 2-D and 3-D. Natural neighbour co-ordinates may be used as a method for obtaining a smooth interpolant over an irregular distribution of points. Despite its apparent complexity, natural neighbour interpolation is practical for problems involving large datasets in 2-D and 3-D. Sambridge et al [13] provides details of algorithms for the 2-D case and discuss applications of natural neighbours to problems in geophysics. The recursive formula which we propose (in section 6) to compute the natural neighbour co-ordinates in 3-D is, in fact, valid in any number of dimensions.

The useful properties of Delaunay triangles and Voronoi cells, and consequently of natural neighbour interpolation and the natural element method, are derived from the fundamental geometric concepts upon which they are based. Although our discussion has been restricted to problems of interpolation and the numerical solution of PDEs, it seems likely that this theory could be used in other computational techniques and perhaps to derive new ones, especially in cases where it is advantageous to use a representation of a scalar field in terms of irregularly distributed points.

References

- [1] Aurenhammer, F., Voronoi Diagrams - A Survey of a Fundamental Geometric Data Structure, *ACM Computing Surveys*, **23**, no. 3, 1991, 345–405.
- [2] Avis and Bhattacharya, Algorithms for computing d -dimensional diagrams and their duals, *Adv. Comput. Res.*, **1**, 1983, 159–180.
- [3] Barber, B., and Huhdanpaa, H., *qhull* (computer program), available via anonymous ftp from geom.umn.edu.
- [4] Braun, J. and Sambridge, M., Dynamical Lagrangian Remeshing (DLR): A new algorithm for solving large strain deformation problems and its application to fault-propagation folding, *Earth Planet. Sci. Lett.*, **124**, 1994, 211–220.
- [5] Braun, J. and Sambridge, M., A numerical method for solving partial differential equations on highly irregular evolving grids, *Nature*, **376**, 1995, 655–660.
- [6] Delaunay, B. N., Sur la sphere vide, *Bull. Acad. Science USSR VII: Class. Sci. Math.*, 793–800, 1934.
- [7] Fortune, S., A sweepline algorithm for Voronoi diagrams, *Algorithmica* **2**, 1987, 153–174.
- [8] Fortune, S., Voronoi diagrams and Delaunay triangulations, in *Computing in Euclidean Geometry*, eds. Du, D. Z. and Hwang, F., World Scientific, 1992.
- [9] Green, P. J. and Sibson R., Computing Dirichlet tessellations in the plane, *Comput. J.*, **21**, 1978, 168–173.

- [10] Lasserre, J. B., An Analytical Expression and an Algorithm for the Volume of a Convex Polyhedron in R^n , *J. of Opt. Theory and Appl.*, **39**, 1983, 363–377.
- [11] Lawson, C. L., Software for C^1 surface interpolation, in *Mathematical Software III*, ed. Rice, J., Academic Press, New York, 1977.
- [12] Okabe, A., Boots, B. and Sugihara, K., 1992. *Spatial Tessellations Concepts and Applications of Voronoi Diagrams*, John Wiley & Sons, Chichester, England.
- [13] Sambridge, M., Braun, J. and McQueen H., Geophysical parametrization and interpolation of irregular data using natural neighbours, *Geophys. J. Int.*, **122**, 1995, 837–857.
- [14] Sibson, R., A vector identity for the Dirichlet tessellation, *Math. Proc. Camb. Phil. Soc.*, **87**, 1980, 151–155.
- [15] Sibson, R., A Brief Description of Natural Neighbour Interpolation, in *Interpreting Multivariate Data*, pp. 21–36, ed. Barnett V., Wiley, Chichester, 1981.
- [16] Voronoi, M. G., Nouvelles applications des parametres continus a la theorie des formes quadratiques, *J. reine Angew. Math.*, **134**, 1908, 198–287.
- [17] Watson, D. F., Natural Neighbour Sorting, *Australian Computer Journal*, **17**, 1985, 189–193.
- [18] Watson, D. F., *Contouring: A Guide to the Analysis and Display of Spatial Data*, Pergamon, Oxford, 1992.

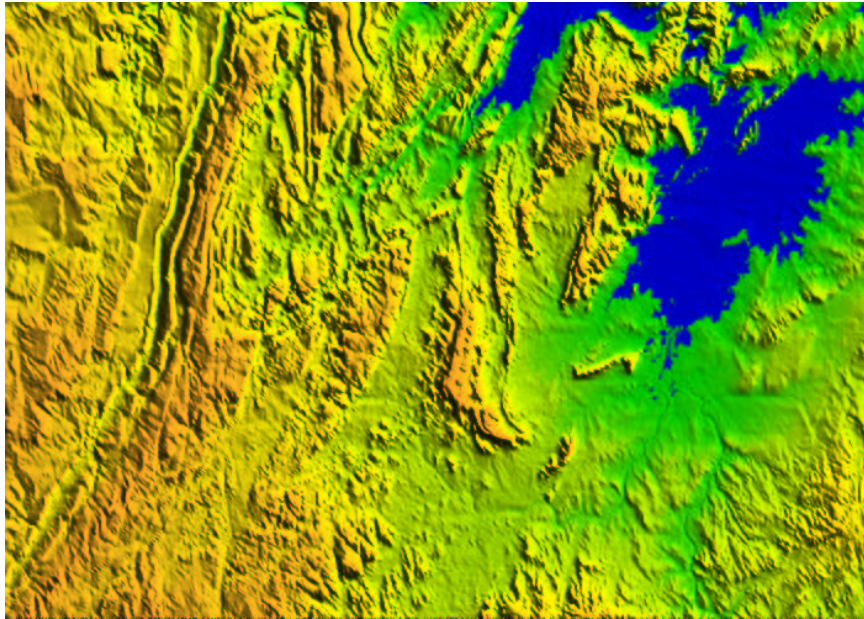


FIGURE 2. Shaded image of a digital terrain model (DTM) obtained by interpolating 688,475 irregularly spaced elevation measurements on a 1152×900 rectangular mesh using the natural neighbour interpolation method (data courtesy of Australian Geological Survey Organisation). The illumination is from the NE.