sciendo

Reports on Geodesy and Geoinformatics, 2023, Vol. 116, pp. 9–14

DOI: 10.2478/rgg-2023-0006
Received: 22 May 2023 / Accepted: 25 July 2023
Published online: 25 August 2023

ORIGINAL ARTICLE

Enhancement of inverse-distance-weighting 2D interpolation using accelerated decline

Andrew Carey Ruffhead [1*]

[1]School of Architecture, Computing and Engineering, University of East London, Docklands Campus, University Way, London E16 2RD, United Kingdom.

*ruffhead40b@yahoo.co.uk

Abstract

Two-dimensional interpolation – or surface fitting – is an approximation tool with applications in geodetic datum transformations, terrain modelling and geoid determination. It can also be applied to many other forms of geographic point data, including rainfall, chemical concentrations and noise levels. The problem of fitting of a smooth continuous interpolant to a bivariate function is particularly difficult if the dataset of control points is scattered irregularly. A typical approach is a weighted sum of data values where the sum of the weights is always unity. Weighting by inverse distance to a power is one approach, although a power greater than 1 is needed to ensure smooth results. One advantage over other methods is that data values can be incorporated into the interpolated surface. One disadvantage is the influence of distant points. A simple cut-off limit on distance would affect continuity. This study proposes a transition range of accelerated decline by means of an adjoining polynomial. This preserves smoothness and continuity in the interpolating surface. Case studies indicate accuracy advantages over standard versions of inverse-distance weighting.

Key words: surface-fitting, inverse-distance weighting, 2D interpolation, accelerated decline

1 Introduction

A formula which covers a wide range of interpolation methods can be categorised as "weighted average". The general form is

$$f_P = \frac{\sum w_i f_i}{\sum w_i}. \tag{1}$$

In this formula, P is the point of interest, $\{P_i\}$ is a set of scattered control points and w_i is a weight dependent on P and P_i. The divisor in (1) ensures that the interpolation is exact at every control point. The divisor turns the coefficient of each f_i into a normalised weight.

If the weights are non-negative, as is the case if w_i is derived from the distance between P and P_i, then the interpolated function is constrained by the range of values in $\{f_i\}$. This is a limitation if the user wants some allowance for extrapolation, but at the same time it is a safeguard against instability.

Two-dimensional problems where weighted-average interpolation is used include several from geomatics. Gradka and Kwinta (2018) apply it to terrain modelling, Soycan and Soycan (2003) apply it to geoid determination, Ligas et al. (2022) apply it to quasi-geoid modelling, and Grgić et al. (2016) apply it to geodetic datum transformations. Applications outside geomatics include air temperatures (Musashi et al., 2018), rainfall measurements (Tomczak, 1998), and housing growth (Cho et al., 2005).

2 Inverse distance to a power

Inverse-distance weighting is a scattered-data interpolation algorithm proposed by Shepard (1968). It is very easy to implement. The normalised weights are non-negative quantities whose sum is 1, and this ensures the interpolating function never strays outside the range of the interpolated values being interpolated. The method is considered in all the weighted-average applications quoted above.

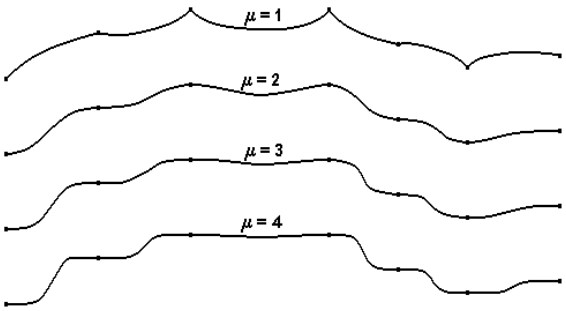

Figure 1. Characteristics of inverse distance interpolation using powers 1, 2, 3 and 4

A continuous and smooth function f is interpolated at scattered control points $\{P_i\}$ by the formula

$$f_P = \frac{\sum f_i/d_i^{\mu}}{\sum 1/d_i^{\mu}}, \tag{2}$$

where d_i denotes the distance from P to P_i. The formula takes the limiting value f_j when P coincides with a control point P_j.

For implementation purposes, there is an alternative form of (2):

$$f_P = \frac{f_j + \sum_{i \neq j} f_i \cdot \left(d_j/d_i\right)^{\mu}}{1 + \sum_{i \neq j} \left(d_j/d_i\right)^{\mu}}, \tag{3}$$

where

$$j = \text{value of } i \text{ that minimises } d_i. \tag{4}$$

The summations in (3) omit the control point nearest to the point of interest. This avoids zero divisors and the processing of large numbers.

The value of μ in formula (2), and hence in (3), must be at least 2 to ensure smooth interpolation. If μ is exactly 2, the control points are said to be weighted by inverse square distances. One argument for having μ greater than 2 is that it limits the influence of distant points.

Figure 1 illustrates the effect of inverse distance interpolation on data which depends on a single variable. None of the data fits is totally satisfactory. Increasing the power μ reduces the dip between the 3rd and 4th points, but accentuates the changes in curvature between other control points. The increased flatness at the control points causes steeper slopes elsewhere, and that steepness is an improbable interpretation of the control data.

In the cases where $\mu > 1$, the main drawback of inverse-distance weighting is that it imposes zero gradients at the control points. When it is used to fit a surface, it has the tendency to generate concentric contours around the control points. This is described by several authors, among them Attaouia et al. (2017) and Musashi et al. (2018), as a bullseye effect. Franke and Nielson (1991) prefer the term "flat spots". As in the one-dimensional case illustrated in Figure 1, the increased flatness at control points that comes from raising μ causes increased steepness elsewhere.

The flat-spots effect can be reduced by splitting the function into a trend model and a signal (analogous to the noise-free version of least-squares collocation; see Ruffhead (1987)). If a suitable trend model is identified, its parameters can be obtained by least-squares optimisation. The residual variable, or signal, is interpolated exactly by inverse distance to a power. The interpolated signal will have zero gradient at the control points, but the overall interpolant at those points will have the same gradient as the trend model. Inverse distance to a power would become a means of interpolating (and thereby eliminating) residuals from the data minus the model.

This researcher believes that extracting a trend model can improve the accuracy of inverse-distance weighting, although he has not seen an explicit statement to that effect in any publication. It was probably regarded as intuitively obvious by Grgić et al. (2016) and Ligas et al. (2022). Both studies used inverse-distance weighting – amongst other methods – to interpolate residuals from a trend model. This paper will test the proposition in its case studies.

A variation of (2) can be applied to control points which have weights based on their perceived reliability:

$$f_P = \frac{\sum w_i f_i/d_i^{\mu}}{\sum w_i/d_i^{\mu}}. \tag{5}$$

Control points with the highest weights will influence the interpolated f within a larger local radius than control points with the lowest weights.

Grgić et al. (2016) apply inverse-distance-to-a-power to interpolate residual transformations. Given that the residual datum shifts are smaller than the original datum shifts, the drawback of zero gradients at control points is less of a problem than in it would be if the method had been applied to the original datum shifts. When inverse square distances were used ($\mu = 2$), the accuracy over Croatia was comparable to that achieved by Kriging and minimum curvature.

In passing, it should be noted that inverse-distance weighting can be modified by introducing a "smoothing factor" into the values of the weights. The control points closest to point P will have the greatest influence on the value of $\bar{f}_P$ but f will not interpolate the control points exactly. One version is given by Tomczak (1998) which claims Keckler (1995) as its source. The latter is a user guide to Surfer Version 6 (Keckler, 1995). The other version is given on page 115 of Surfer (2002). A trial-and-error process for deriving the smoothing factor is suggested in Woodson (2016).

This paper only considers the method of inverse-distance weighting as an exact interpolator. The generic form in Surfer can be used as such provided the smoothing factor to set to zero.

The method of "inverse distance to a power" can be used to generate a rectangular mesh of pseudo-data points from which further interpolation is done by means of bilinear or bicubic functions. This may be unnecessary because of the simplicity of the inverse-distance method itself.

In geospatial science, data is often defined over a region of the Earth represented by an ellipsoid, with coordinates given in latitude and longitude. The method can be applied is one of two ways. A projection could be applied to generate plane coordinates from which distances between points are computed using Pythagoras's theorem. Alternatively, ellipsoidal distances can be computed, either by a rigorous formula for geodesics such as the one in Sodano (1965) or by an approximate formula, such as a Pythagorean estimate using the arc lengths between the latitudes and longitudes of the respective points.

3 Hybrid inverse power function embodying accelerated decline (HIPFEAD)

One of the serious deficiencies of inverse-distance weighting for $\mu > 1$, noted by Franke and Nielson (1991), is "undue influence of points which are far away" especially for $\mu = 2$. Simply removing the influence of points beyond a certain distance would affect continuity and smoothness.

The new method offers a solution to this problem. It involves a new process for calculating weights of function values for distance-related interpolation. It resembles inverse distance to a power. However, it imposes a limit-of-influence, removing the influence of control points beyond a given distance r_{max} from the point of in-

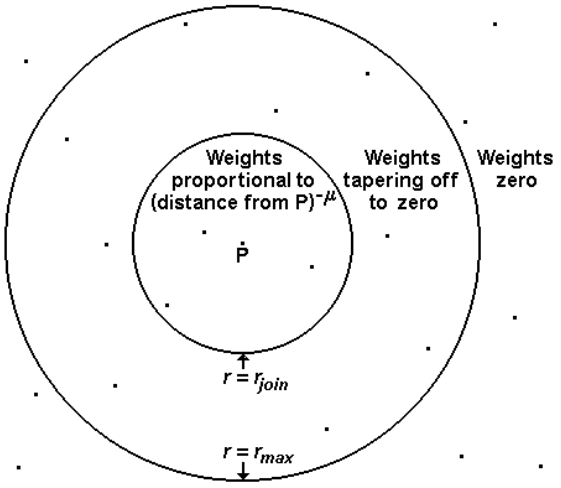

Figure 2. Effect of radial partitioning on the weights defined by the hybrid inverse square function for interpolation at *P*

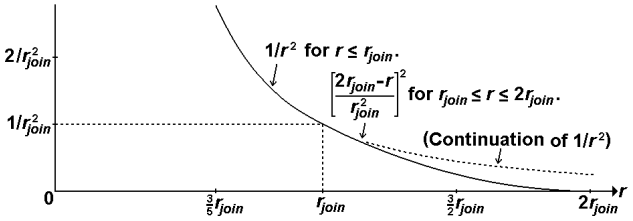

Figure 3. HISFEAD with r_{join} as the defining constant

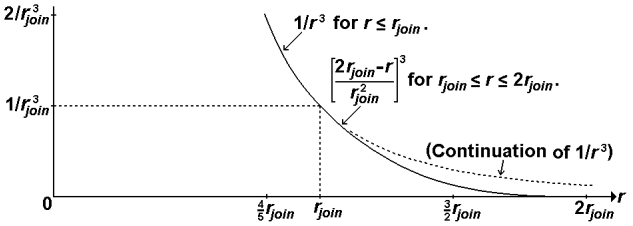

Figure 4. HICFEAD with r_{join} as the defining constant

terest. It does this by a smooth join (at $r = r_{join}$) between inverse distance to a power and a low-degree polynomial function of distance. The latter accelerates the decline of the weighting function, hence its name.

For the weighting of function values at control points on a surface around point *P*, the effect can best be illustrated by Figure 2. The circles defined by $r = r_{join}$ and $r = r_{max}$ can be regarded as "radial partitioning" of the area of interest. The unlabelled dots are illustrative control points.

The simplest form of the polynomial component is $c\,(r_{max} - r)^{\mu}$ where c is a constant. For the weighting function to be continuous and smooth at $r = r_{join}$, the following equations need to be satisfied:

$$1/r_{join}^{\mu} = c\left(r_{max} - r_{join}\right)^{\mu}; \tag{6}$$

$$-\mu/r_{join}^{\mu+1} = -\mu c\left(r_{max} - r_{join}\right)^{\mu-1}. \tag{7}$$

From these equations, it is easily deduced that

$$r_{join} = r_{max} - r_{join}. \tag{8}$$

Substituting into (6),

$$c = 1/r_{join}^{2\mu}. \tag{9}$$

Rearranging (8),

$$r_{max} = 2r_{join}. \tag{10}$$

This ensures a balance between a gradual tapering-off and exclusion of influence from faraway points. A value of r_{max} smaller than $2r_{join}$ would make the transition from $1/r^2$ to zero relatively abrupt. A value of r_{max} larger than $2r_{join}$ would compromise the objective of limiting the number of points used in the interpolant.

The HIPFEAD version of interpolation formula (1) defines the distance-dependent weights as follows:

$$w(r) = \begin{cases} 1/r^{\mu} & \text{if} \quad 0 \le r \le r_{\text{join}}; \\ \left[\left(2r_{join} - r\right)/r_{join}^2\right]^{\mu} & \text{if} \quad r_{join} \le r \le 2r_{join}; \\ 0 & \text{if} \quad r \ge 2r_{join}. \end{cases} \tag{11}$$

The smoothness of the weighting function ensures that HIPFEAD generates a C^1 surface. Figures 3 and 4 illustrate the HIPFEAD weighting function in the cases $\mu = 2$ and $\mu = 3$. These are the subtypes of HIPFEAD considered in this project:

- Hybrid inverse square function embodying accelerated decline (HISFEAD), in which inverse square distance is joined smoothly to a quadratic polynomial. HISFEAD can be considered a modification of weighting by inverse square distance.
- Hybrid inverse cubic function embodying accelerated decline (HICFEAD), in which inverse cubic distance is joined smoothly to a cubic polynomial. HICFEAD can be considered a modification of weighting by inverse cubic distance.

The application of HIPFEAD has a similar problem to inverse distance to a power, namely a near-zero divisor when the point of interest is close to a control point. In (11), $w_i = \infty$ when $d_i = 0$.

The solution is similar to that used in (3) and (4). Define j by equation (4), making it the subscript of the nearest control point. If $d_j \ge r_{join}$ the proximity problem does not arise. If $d_j < r_{join}$, which means the reciprocal of w_j is d_j^{μ}, then formula (1) can be replaced by

$$f_P = [f_j + \sum_{i \neq j} d_j^{\mu} w_i f_i]/[1 + \sum_{i \neq j} d_j^{\mu} w_i]. \tag{12}$$

This avoids the need to compute w_j.

One potential characteristic of HIPFEAD is the interpolant taking a constant value over one or more sub-areas. This will happen if there is an area for which for which only one control point is within distance r_{max}; the interpolant will take the value of f at that control point for the whole of that area. (This is, of course, a characteristic of all sub-areas in the case of nearest-neighbour interpolation.)

A problem arises if a point of interest is more than r_{max} from *every* control point. There are two would-be solutions, but each of them is problematic:

- Setting the interpolated f to zero in sub-areas which are at least r_{max} from all control points; but this introduces discontinuities at the boundaries of those sub-areas.
- Setting the interpolated f to nearest neighbour in sub-areas which are at least r_{max} from all control points; but this introduces discontinuities across lines which are equidistant from two control points.

It is therefore a prerequisite for HIPFEAD that r_{max} is sufficiently large to ensure that *every* possible point of interest is within r_{max} of at least one control point. One way of ensuring this is to impose a rectangular grid over the area of interest, setting a grid interval

of Δl. This ensures that every point in the area is within $\Delta l\sqrt{2}$ of a mesh point. The distance to the nearest control point is computed for each mesh point. The largest of these values is set to d_{min}. Then any value of r_{max} that exceeds $d_{min} + \Delta l\sqrt{2}$ will ensure that every point is within r_{max} of a control point. The quantity $d_{min} + \Delta l\sqrt{2}$ can be considered as a grid-based lower limit on r_{max}.

The upper limit on the choice of r_{max} depends on how far is considered too far to have an influence on the interpolated value of f.

4 Case studies

The case studies are all simulated. They are defined in subsets of the (x, y) plane. Coordinates are in linear units such as (but not necessarily) metres. The function $f(x, y)$ being interpolated can be visualised as a surface, so in terms of graphical representation f can be visualised as a "height".

Data points were generated in the chosen areas using pseudo-random numbers. "Height" values were created by combining smooth continuous functions of different types (exponential, logarithmic, rational, trigonometric, etc) so as to imitate a non-mathematical physical surface. The advantage of so doing is that the accuracy of the interpolants can be measured against the same combinations of functions.

The methods compared were inverse-distance weighting (inverse square and inverse cubic)) and HIPFEAD (HISFEAD and HICFEAD). They were applied initially to the original data and then to the "signal" after the removal of a trend model. In each case, the latter was a bivariate quadratic polynomial of the normalised coordinates that gave the best least-squares fit to the control points.

4.1 Case study 1

The first study considered a square region defined by $0 \le x \le 40000$ and $0 \le y \le 40000$. Pseudo-random numbers were used to generate 1525 data points in the region. 1681 computation points were defined at 1000-unit intersections, so that 160 were boundary points and 1521 were non-boundary points.

The pseudo-physical surface was generated by the following combination of functions:

$$\begin{aligned} f &= 15 + 1.3\sin(x/4000) + 2.3\cos(y/5500) + 261/(x + 123.5) \\ &\quad + 416.9/(40280 - y) + (20000 - x)/(y + 12000) \\ &\quad + 0.9\exp[-\{(x - 21452)^2 + (y - 33461)^2\}/4000000] \\ &\quad - 1.3\exp[-\{(x - 15436)^2 + (y - 22786)^2\}/3000000] \\ &\quad + \exp[-\{1.2\,(x - 37755)^2 + 0.8\,(y - 28044)^2\}/3500000] \\ &\quad - \exp[-\{0.86\,(x - 11458)^2 + 1.14\,(y - 3865)^2\}/5500000]. \end{aligned} \tag{13}$$

An overview of the surface is given in Figure 5. The grid points in the defined region coincide with the decimal points of the "height" values. They are 2500 units apart, which means that the diagram does not capture all the complexities of the surface generated by equation (13).

The trend model obtained from the control data for the purpose of deriving a signal was

$$\begin{aligned} \text{Model} &= 13.48677 - 0.85755U + 0.85864V \\ &\quad + 1.50270U^2 + 0.55667UV + 4.67250V^2. \end{aligned} \tag{14}$$

The normalised coordinates U and V in (14) are defined by

$$U = (x - 20000)/20000 \text{ and } V = (y - 20000)/20000. \tag{15}$$

The results from the four methods are shown in Table 1.

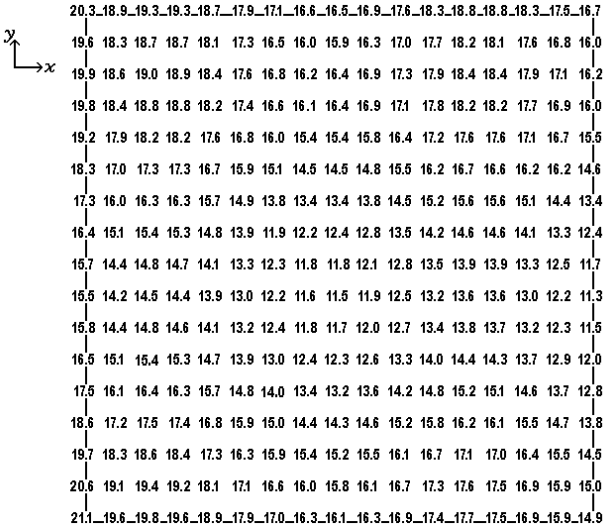

Figure 5. General-pattern surface map for case study 1

4.2 Case study 2

The second study considered a circular region defined by $x^2 + y^2 \le 25000^2$. Pseudo-random numbers were used to generate 1564 data points in the region. 1625 computation points were defined on concentric circles with radii 1000, 2000, …, 25000 units so that the ith circle has $5i$ equally-spaced computation points. That created 125 boundary points on the outermost circle and 1500 non-boundary points on the inner circles.

The pseudo-physical surface was generated by the following combination of functions:

$$\begin{aligned} f &= (250000 - 3x)/(25000 + 0.00004x^2) \\ &\quad + (0.00012y - 2)^2/\sqrt{2 + 0.00004y} \\ &\quad + 3\cos[0.0004\,(x + y)] \\ &\quad + 50000/(750000 + x - y). \end{aligned} \tag{16}$$

An overview of the surface is given in Figure 6. The grid points in the defined region coincide with the decimal points of the "height" values. They are 2500 units apart, which means that the diagram does not capture all the complexities of the surface generated by equation (16).

The trend model obtained from the control data for the purpose of deriving a signal was

$$\begin{aligned} \text{Model} &= 12.23124 - 1.95742U - 10.61535V \\ &\quad - 5.21667U^2 + 1.00573UV + 10.27861V^2. \end{aligned} \tag{17}$$

The normalised coordinates U and V in (17) are defined by

$$U = x/25000 \text{ and } V = y/25000. \tag{18}$$

The results from the four methods are shown in Table 2.

4.3 Case study 3

The third study considered the trapezoidal region illustrated in Figure 7. Pseudo-random numbers were used to generate 1504 data points in the region. 1126 computation points were defined where units-of-1000 gridlines crossed each other (except less than 500 units from the boundary) and where they crossed the boundary. That created 131 boundary points and 995 non-boundary points.

The pseudo-physical surface was generated by the following

Table 1. Quality of fit from the different interpolation methods for Case Study 1

	Inv Square	Inv Cubic	HISFEAD	HICFEAD
RMS for 160 boundary points	1.351 (0.844)	0.906 (0.655)	0.825 (0.674)	0.816 (0.682)
RMS for 1521 non-boundary points	0.512 (0.380)	0.135 (0.124)	0.091 (0.099)	0.086 (0.088)
RMS for all 1681 computation points	0.641 (0.446)	0.308 (0.234)	0.269 (0.228)	0.264 (0.226)

Figures in brackets give the "signal" RMS when the 6-parameter trend model is removed.

Table 2. Quality of fit from the different interpolation methods for Case Study 2

	Inv Square	Inv Cubic	HISFEAD	HICFEAD
RMS for 125 boundary points	3.044 (1.329)	1.238 (0.866)	0.993 (0.855)	0.844 (0.735)
RMS for 1500 non-boundary points	1.447 (1.129)	0.571 (0.537)	0.460 (0.427)	0.399 (0.364)
RMS for all 1625 computation points	1.626 (1.146)	0.647 (0.569)	0.521 (0.471)	0.450 (0.405)

Figures in brackets give the "signal" RMS when the 6-parameter trend model is removed.

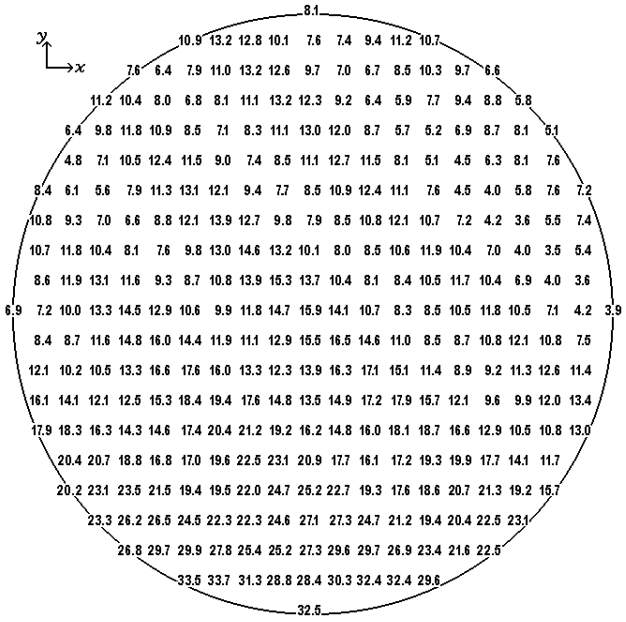

Figure 6. General-pattern surface map for case study 2

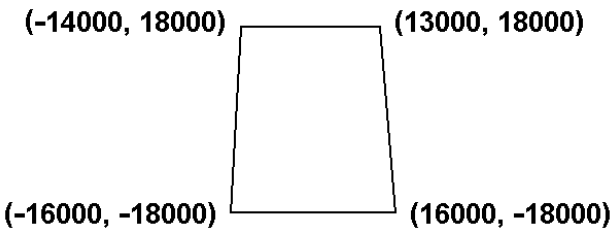

Figure 7. Trapezoidal region for Case Study 3

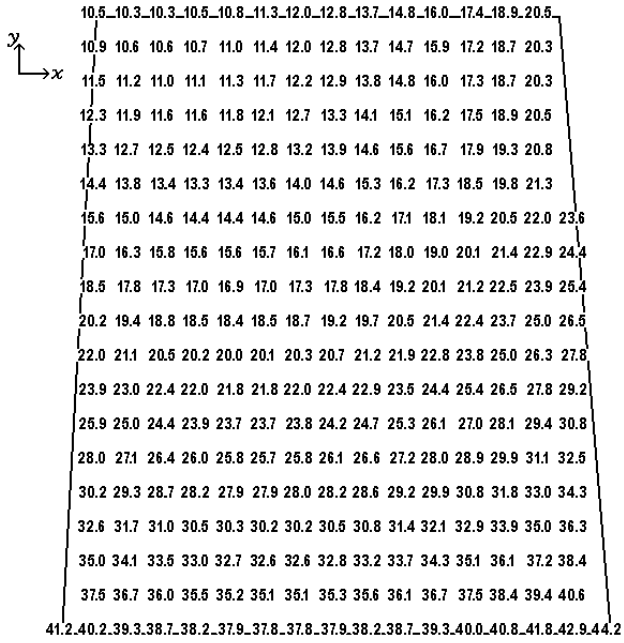

Figure 8. General-pattern surface map for case study 3

combination of functions:

$$\begin{aligned} f = {} & \frac{(x/8000 + 3)^2 + (y/6000 - 5)^2}{\sqrt{4 + x/8000 - y/18000}} \\ & + (y/18000 - 0.2)^2 \ln(14 + x/1600) \\ & + (x/16000 + 0.3)^2 \ln(16 + y/1800) \\ & + \sqrt{7 - x/5333}\exp(y/18000 - 1) \\ & + \sqrt{5 - y/9000}\exp(x/16000 - 1). \end{aligned} \tag{19}$$

An overview of the surface is given in Figure 8. The grid points in the defined region coincide with the decimal points of the "height" values. They are 2000 units apart, which means that the diagram does not capture all the complexities of the surface generated by equation (19).

The trend model obtained from the control data for the purpose of deriving a signal was

$$\begin{aligned} \text{Model} = {} & 19.18808 + 3.92375U - 12.78205V \\ & + 5.29684U^2 + 2.64749UV + 5.97880V^2. \end{aligned} \tag{20}$$

Table 3. Quality of fit from the different interpolation methods for Case Study 3

	Inv Square	Inv Cubic	HISFEAD	HICFEAD
RMS for 131 boundary points	3.491 (0.161)	1.281 (0.085)	0.651 (0.061)	0.579 (0.053)
RMS for 995 non-boundary points	1.176 (0.037)	0.241 (0.011)	0.163 (0.009)	0.172 (0.008)
RMS for all 1126 computation points	1.625 (0.065)	0.492 (0.031)	0.270 (0.022)	0.255 (0.020)

Figures in brackets give the "signal" RMS when the 6-parameter trend model is removed.

The normalised coordinates U and V in (20) are defined by

$$U = x/16000 \text{ and } V = y/18000. \tag{21}$$

The results from the four methods are shown in Table 3.

5 Discussion

In all three case studies the most accurate interpolation method is HICFEAD, followed by HISFEAD, inverse cubic and inverse square. Although this applies to Case Study 3 overall, HISFEAD gives a 5% better fit than HICFEAD at the non-boundary points, but only when there is no trend model.

The case studies confirm the expectation that introducing a trend model and treating the residuals as a signal to be interpolated improves the accuracy of all four methods. The extent depends on the variations in the data and the type of trend model that is chosen. The improvement is least (14% to 30%) in Case Study 1, where the HIPFEAD methods show a slight accuracy reduction at the non-boundary points.

The case studies show a tendency for HICFEAD to give a better fit than HISFEAD. This is part of a general tendency for inverse cubic distance to give a better fit than inverse square distance. Given that the former has a tendency to produce wider flat areas and steeper slopes, it may not always be a better interpolator. The study by Musashi et al. (2018) favoured 2 as the best value of μ, although it should be noted that the interpolation made no use of trend models.

One problem which is common to all the methods based on inverse distance to a power (including HIPFEAD) is the absence of extrapolation outside the bounding polygon of control points. This is the reason for the fit at boundary points being inferior to that at non-boundary points, which are more likely to be in the bounding polygon. This is, in fact, a further reason for using a trend model, since a suitably chosen one will have an element of extrapolation around the bounding polygon.

Inverse-distance weighting and the enhanced versions proposed in this paper are exact interpolants with respect to those data points used as control points. As with other surface-fitting methods, in order to have an independent estimate of accuracy, some data points will need to be set aside as test points or verification points, where the interpolation will produce residuals. In cases where the data is a finite set of actual physically-generated "height" values, interpolation cannot be exact at all of them if there is to be a meaningful accuracy estimate for the region as a whole.

The main conclusion of this study is that a transition range of accelerated decline improves the accuracy of by inverse-distance-weighting 2D interpolation. The diminished influence of distance points is achieved by an adjoining polynomial of the same power as that applied to inverse distance.

As a result, comparisons between inverse-distance weighting and other methods of interpolation need to be reconsidered in the light of the accuracy improvements obtainable using HIPFEAD. To illustrate this point, consider the datum transformation example of Grgić et al. (2016). Inverse square distance weighting was found to give comparable accuracy to Kriging and minimum curvature, and better accuracy than the other methods obtainable from Surfer software. This strongly suggests that HIPFEAD (with $\mu = 2$ or $\mu = 3$) would give superior results to any method in that study.

References

Attaouia, B., Salem, K., Boualem, G., and Bachir, G. (2017). Computation of continuous displacement field from GPS data-comparative study with several interpolation methods. In *Conference Paper (FIG Working Week 2017: Surveying the world of tomorrow-From digitalisation to augmented reality May 29-June 2, Helsinki Finland)*.

Cho, S.-H., English, B. C., and Roberts, R. K. (2005). A spatial analysis of housing growth. *Review of Regional Studies*, 35(3):311–335.

Franke, R. and Nielson, G. M. (1991). Scattered data interpolation and applications: a tutorial and survey. *Geometric Modeling: Methods and Applications*, pages 131–160, doi:10.1007/978-3-642-76404-2_6.

Gradka, R. and Kwinta, A. (2018). A short review of interpolation methods used for terrain modeling. *Geomatics, Landmanagement and Landscape*, 4:29–47, doi:10.15576/GLL/2018.4.29.

Grgić, M., Varga, M., and Bašić, T. (2016). Empirical research of interpolation methods in distortion modeling for the coordinate transformation between local and global geodetic datums. *Journal of surveying engineering*, 142(2):05015004, doi:10.1061/(ASCE)SU.1943-5428.0000154.

Keckler, D. (1995). *The Surfer Manual*. Inc.: Golden, CO, USA.

Ligas, M., Lucki, B., and Banasik, P. (2022). A crossvalidation-based comparison of kriging and IDW in local GNSS/levelling quasigeoid modelling. *Reports on Geodesy and Geoinformatics*, 114(1):1–7, doi:10.2478/rgg-2022-0004.

Musashi, J. P., Pramoedyo, H., and Fitriani, R. (2018). Comparison of inverse distance weighted and natural neighbor interpolation method at air temperature data in Malang region. *CAUCHY: Jurnal Matematika Murni dan Aplikasi*, 5(2):48–54, doi:10.18860/ca.v5i2.4722.

Ruffhead, A. (1987). An introduction to least-squares collocation. *Survey review*, 29(224):85–94, doi:10.1179/sre.1987.29.224.85.

Shepard, D. (1968). A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference, Association for Computing Machinery, New York*, pages 517–524. doi:10.1145/800186.810616.

Sodano, E. M. (1965). General non-iterative solution of the inverse and direct geodetic problems. *Bulletin Géodésique (1946-1975)*, 75:69–89, doi:10.1007/BF02530662.

Soycan, M. and Soycan, A. (2003). Surface modeling for GPS-leveling geoid determination. *Newton's Bulletin*, 1:41–52.

Surfer (2002). *Surfer User's Guide: Contouring and 3D surface mapping for scientists and engineers*. Inc.: Golden, CO, USA.

Tomczak, M. (1998). Spatial interpolation and its uncertainty using automated anisotropic inverse distance weighting (IDW) – Cross-validation/Jackknife approach. *Journal of Geographic Information and Decision Analysis*, 2(2):18–30.

Woodson, J. (2016). *How can I remove the bullseye effect that is created in my Surfer grid?* Inc.: Golden, CO, USA.