Sule Lamido University Journal of Science & Technology Vol. 4 No. 1&2 [July, 2022], pp. 97-105
https://doi.org/10.56471/slujst.v4i.274

Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.com.ng

Examining Inverse Distance Weighting Interpolation Method in Contouring Algorithm

[1]Abdulrazaq Abdulrahim, [2]Mohammed Yahaya Tanko, [3]Muhammad Aminu Umar, [4]Sheidu Salami Tenuche and [5]Aliyu Kufena Muhammad
Department of Computer Science, Ahmadu Bello University, Zaria. Nigeria.
([1]aabdulrahim, [2]tymohammed, [3]maumar, [4]sstenuche)@abu.edu.ng, ([1]rsk7000, [5]kufenah)@gmail.com

Abstract
Representation of results/data graphically depicts a better understanding of the behavior of the results/data. Contour plotting is an easy way of representing results/data. Contouring algorithms use linear interpolation in determining the point of the intersection between contour lines and grid segments when drawing contour lines. Using linear interpolation is not very precise and results in discontinuities at end points. This paper presents and examines the contouring algorithm that uses inverse distance weighting interpolation in determining the point of the intersection between contour lines and grid segments of randomly generated data. Comparison made between the maps produced by these algorithms that use linear, cubic and inverse distance weighting interpolation, showed that the map produced by inverse distance weighting interpolation is wrong because different contour lines cross each other, the map produced by cubic interpolation depicts less information because some contour lines are missing when compared with the map produced by linear interpolation.

Keywords: *graphic representation, contour, contour algorithm, linear interpolation, cubic interpolation*

1. Introduction

To understand the numerical model behavior of any kind of results/data, it is essential to represent such results/data in a graphical format. The examination of a vast amount of results/data is replaced by a simple visual depiction in this graphical format. This graphical representation of results/data is the best way to show a large amount of data to people, which could put them to sleep or/and convey little or no information, when read in plain text. A step toward to such representation of results/data is the use of Contour plotting (Singh & Saini, 2011). A contour plot is the most practical and effective approach to express three-dimensional data graphically on a two-dimensional surface such as a map or a computer screen; it typically shows a bunch of lines, called contours or contour lines, with different values; which are often wavy or forming concentric, irregular closed loops or other patterns.

Contour lines generally do not meet or intersect each other, but connects points of equal quantity (e.g., elevation, pressure, temperature etc.) of the contour area. It is a function of two variables; a contour line of a function g(a,b) with value v, is a collection of all points (a,b) where g(a,b)=v (Aramini, 2012). You've probably used a contour plot if you've ever gone hiking and brought a topographical map with you to show you the lay of the land. Many disciplines primarily use contour plot to analyze/visualize an actual or theoretical surface of application and to see the pattern of distribution or find anomalies in the parameters of the discipline that uses it. The most common applications are displaying data of topological features of an area on a map (Paul, 1987). The depiction of selected features of the earth's surface in graphical form is done on topographic maps. The depiction of the elevation and the shape of the terrain is the unique feature of topographic maps. Such maps represent the physical properties of the terrain in an accessible and legible format, as determined by exact engineering surveys and measurements. The contour plot shown in Figure 2 represents the terrain (hill) in Figure 1.

Sule Lamido University Journal of Science & Technology Vol. 4 No. 1&2 [July, 2022], pp. 97-105
https://doi.org/10.56471/slujst.v4i.274
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.com.ng

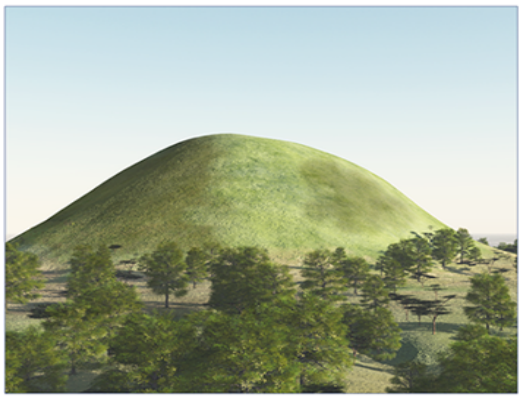

Figure 1. Hill (Source: http://www.bushwalking101.org/interpreting-map-features/)

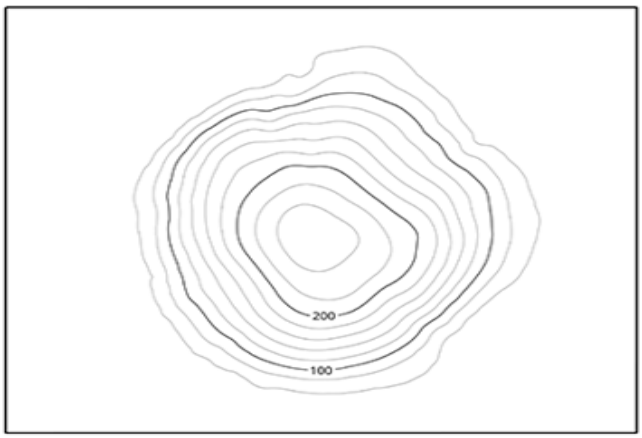

Figure 2. Graphical representation of the Hill (Source: http://www.bushwalking101.org/interpreting-map-features/)

Computer algorithms for contouring spatial data differ in subtle ways that affects the appearance of maps which they produce. Most operate by estimating the values of the surface at a regular grid work of points across the map area. Contour lines are drawn through this grid work by mathematically interpolating the grid edges (segments) with the contour lines (Davis, 1987), in order to determine the point of the intersection between them. Almost all contour drawing algorithms (Davis, n.d.; Meek & Beer, 1976; Singh & Saini, 2011; Snyder, 1978; Yeo, 1984) use linear interpolation to determine the points of the intersection between each contour lines and segments where the contour lines must cross or intersect. In these contour plotting algorithms, linear interpolation is used mainly because of its simple nature to use and to implement and for fast contour plotting. It also produces smooth contour surfaces as long as the line segments joining the points of intersection are short.

Despite the simple, fast, easy to use and to implement nature of the linear interpolation in finding the points of the intersection between segments and contour lines in contour plotting algorithms, and the generation of smooth contour surfaces, it is not very precise and results in discontinuities at end points (Paul, 1999). The contour surfaces generated by these algorithms deviate from the true location (i.e.; the real intersection point between a contour line and a segment) which may be quite too far from the actual intersection point predicted by linear interpolation (Aramini, 2012; Singh & Saini, 2011). So, there is a need for a better interpolation method in determining the points of the intersection between contour lines and segments in order of generated a better and more accurate contour plot.

SLUJST
Sule Lamido University Journal of Science & Technology Vol. 4 No. 1&2 [July, 2022], pp. 97-105
https://doi.org/10.56471/slujst.v4i.274
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.com.ng

The contributions of this paper are to:

- use inverse distance weighting (IDW) interpolation method in determining the points of intersections between contour lines and segments in the contour plotting algorithm
- plot a contour map of a randomly generated numbers with the proposed algorithm and
- compare the map generated by this interpolation method with the maps produced by linear interpolation and cubic interpolation (Abdulrazaq *et al.,* 2021).

2. Related Works

Much information are needed by the contouring algorithms, all they required are the contour values and estimated function values at discrete nodal points. Many researchers have worked on different interpolation schemes in the contour plotting algorithm. Meek *et al* (1976) developed a scheme using linear interpolation which is based on any isoparametric element surface in 2D domain. A linear interpolation-based algorithm for determining the point of the intersection between a contour line and grid edges was developed by (Snyder, 1978). Aramini (2012) developed an algorithm that uses reverse linear interpolation to determine the point of the intersection between a contour line and grid edges then used the algorithm to plot contour map of orbital $7a_1$ of BH_3CO. Yeo (1984) developed an algorithm which is beneficial for a variety of purpose 2D mesh where a linear interpolation is used over triangular elements, which are refined in the domain.

Wu et al., (2020) uses linear interpolation to improve second order machine-learning (ML) based visible-light-positioning (VLP) system when reducing training samples in ML algorithms. Ekoule *et al* (1991) developed a broad triangulation process that solves the issues of traditional triangulation algorithms while simultaneously plotting contours linearly across an element. A cubic interpolation contouring algorithm was proposed by (Goldani-Moghaddam & Enright, 2008) to solve partial differential equation (PDE) via visualization. This algorithm locates the points where contour curves and triangle sides intersect, then creates smooth contour curves to connect these spots. Singh & Saini (2011), developed a very fast algorithm for plotting contour on 2D and 3D domains. This algorithm used linear interpolation to interpolate the three-node triangular and four-node tetrahedral element over the function.

Xiao-Ping (2016) proposed a spline interpolation function to improve the comprehensive regionalized method of linear interpolation that are used in creating gridded digital elevation models (DEMs) from contour lines for geomorphic information. To predict the accurate wear of rough surface contact (spur gears), a numerical model that uses piecewise linear interpolation was proposed by (Wang et al., 2021). To determine the appropriate interpolation method that can be used in carrying out the accurate surface analysis of chromium (Cr) and cadmium (Cd) which have hostile influences on human health. Ouabo et al., (2020) assess the performance of Ordinary kriging and Inverse Distance Weighted (IDW) Interpolation methods. IDW interpolation outperformed Ordinary kriging with regards representation of the concentration of Cd and Cr on the maps of the study area. To study the accurate water pollutants distribution in Wadi El Bey river Watershed, Tunisia. GIS and Inverse Distance Weighted Interpolation (IDW) method were used by (Khouni et al., 2021) in predicting the variation of the pollutants all over the study area.

Sule Lamido University Journal of Science & Technology Vol. 4 No. 1&2 [July, 2022], pp. 97-105
https://doi.org/10.56471/slujst.v4i.274
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.com.ng

3. Contouring Algorithm

The contouring algorithm is shown Algorithm 1

Algorithm 1 Contouring Algorithm

Input: Values at discrete grid points.
Output: Contour map

1: Determine the minimum and maximum values of the grid points
2: Computes a number of contour values by arithmetic method between extreme points using equation 2
3: **for** each contour value
4: Walking through the grid of points looking for any segment that the contour value lays between
5: **if** contour value lays between the segment
6: Determined the point of the intersection between the contour and the segment by any interpolation method
7: **end if**
8: Locate a neighboring segment having a similar property
9: **if** found
10: Repeat Step 6
11: Draw a straight-line segment to connect the previous and current intersection points
12: **end if**
13: **end for**
14: **Return** contour map

3.1 Arithmetic method

This method is used to evaluate contour values that are inside grid nodes range of values. The range of values is obtained by determining the minimum and maximum values of all the grid nodes values. Three parameters are required in generating contour values using this method, they are:

i. The starting (minimum) value
ii. a constant increment between levels (delta) and
iii. the number of contour values to be generated

Equation 1 is used in determining the constant increment between levels:

$$delta = \frac{zmax + zmin}{ncv} \qquad (1)$$

Where $zmax$ and $zmin$ respectively are the maximum and minimum of all values at the grid points and *ncv* is the number of contour values to be evaluated. Contour values would then be: *zmin*, *zmin* + *delta*, *zmin* +2* *delta*,…. *zmin* +(*ncv* -1)* *delta*,

In general, this can be represented in equation 2.

$$cv[i] = zmin + i * delta \qquad (2)$$

Where *cv[i]* is the *ith* contour value

3.2 Interpolation

Interpolation is any procedure for fitting a function to a set of points/values in such a manner that the function intercepts each of the points/values. Unknown points/values of different observation (chemical concentrations, noise levels, rainfall, elevation etc.) can be predicted by interpolation. There are several types of interpolation methods:

SLUJST
Sule Lamido University Journal of Science & Technology Vol. 4 No. 1&2 [July, 2022], pp. 97-105
https://doi.org/10.56471/slujst.v4i.274
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.com.ng

3.2.1 Linear Interpolation

This is the simplest way for obtaining values between data points; it only requires the two data points at the segment's endpoints. The aim is to determine the point of the intersection between contour line and grid segment. Equation 3 shows the formula for determining the point of intersection using linear interpolation:

$$l(x) = z1 + w\delta x \quad (3)$$

Where *l(x)* is the linear interpolant, δx is the scale of the grid segment to a specific unit; given by equation 4.

$$\delta x = \frac{z2 - z1}{unit} \quad (4)$$

*w [0, 1]*is the local variable, given by equation 5:

$$w = \frac{cval - z1}{z2 - z1} \quad (5)$$

z1 and *z2* are the two adjacent values of a segment and cval is the contour value. But a significant issue with linear interpolation is that, it results in discontinuities at each point.

3.2.2 Inverse Distance Weighting

This is an interpolation method that assumes that grid points have a local influence on a contour value, which diminishes with distance. This local influence gives grater weights to the grid points closer to the contour value, while grid points further away from the contour value are given lower weights. The weights of the grid points are used to identify where the contour lines and grid points intersect. Equation 6 shows the formula for utilizing inverse distance weighting interpolation to find the point of intersection between contour lines and grid points.:

$$idw(x) = zi + ziWeight * \delta x \quad (6)$$

idw(x) is the inverse distance weight interpolant, δx is the scale of the grid segment to a specific unit; given by equation 4 and $ziWeight$ is the weight contributed by *zi* as shown in equation 7.

$$ziWeight = \frac{wi}{TWeight} \quad (7)$$

$TWeight$ is the total weight depicted by equation 8.

$$TWeight = \sum_{i=1}^{2} ziWeight \quad (8)$$

wi is the weight contributed by *zi* (the two adjacent values of a segment) to the contour value as given in equation 9. These weights are measured based on the distances of *z1* and *z2* from the contour values and must sum up to 1 as shown in equation 10, to avoid overestimation or underestimation of the intersection point.

$$wi = \frac{1}{di} \quad (9)$$

$i \in 1,2$

$$1 = w1 + w2 \quad (10)$$

4. Experimental Evaluation

In this section, the implementation and evaluation of inverse distance weighting interpolation method in contouring algorithm against linear and cubic interpolation was done.

Sule Lamido University Journal of Science & Technology Vol. 4 No. 1&2 [July, 2022], pp. 97-105
https://doi.org/10.56471/slujst.v4i.274
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.com.ng

4.1 Setup

The implementation of these contouring algorithms is based on the modification of (Rand, 1997); which uses java applet to plot contours based on a given matrix of floating-point values. This implementation was done on a Windows 10 operating system, using NetBeans IDE, which includes the version 2 of JDK. Data used for this experiment are randomly generated.

4.2 Results and Discussion

Figures 3-5 show the maps that are respectively generated by linear, cubic and inverse distance weighting interpolation methods in determining the intersection point between contour lines and grid segments that are randomly generated (Abdulrazaq *et al.,* 2021). Results show that the map generated by the algorithm that uses linear interpolation is smoother and contains more information (contour lines) than the map generated by the algorithm that uses cubic and inverse distance weighting interpolation. This is because, some contour lines on the map produced by the cubic interpolation are missing when compared with the map produced by linear interpolation. The map produced by inverse distance weighting interpolation is not correct because contour lines produced by this method are crossing each other. Which implies that a point can two values (i.e., values of the contour lines that cross each other), which is not possible.

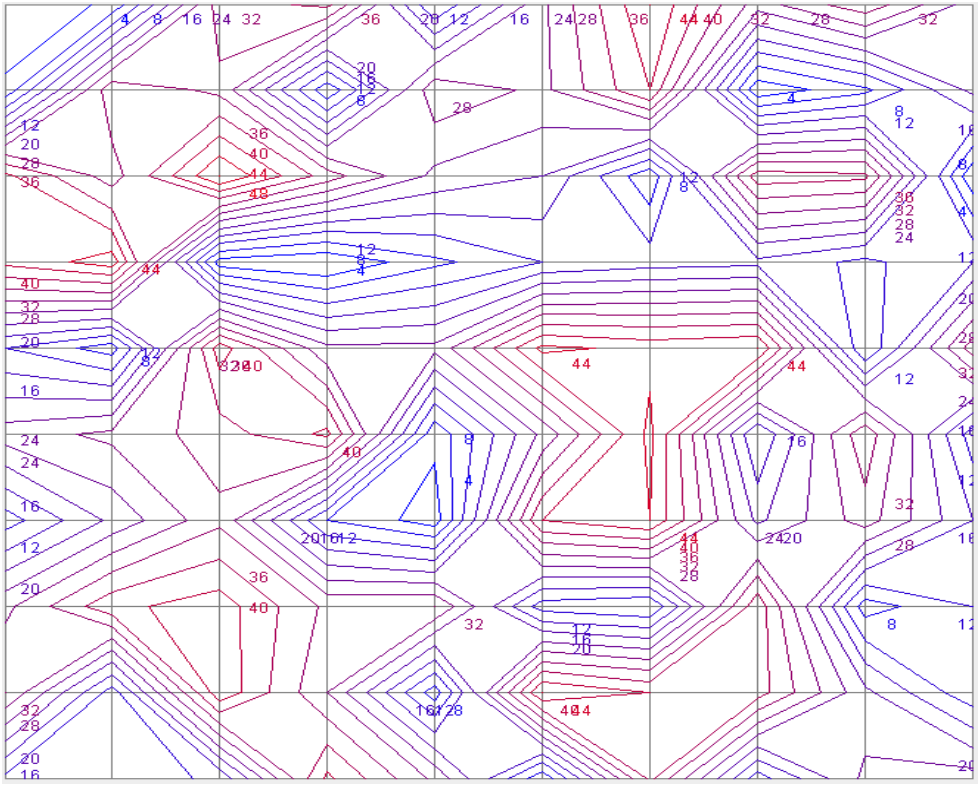

Figure 3. Contouring Map that Uses Linear Interpolation

Sule Lamido University Journal of Science & Technology Vol. 4 No. 1&2 [July, 2022], pp. 97-105
https://doi.org/10.56471/slujst.v4i.274

Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.com.ng

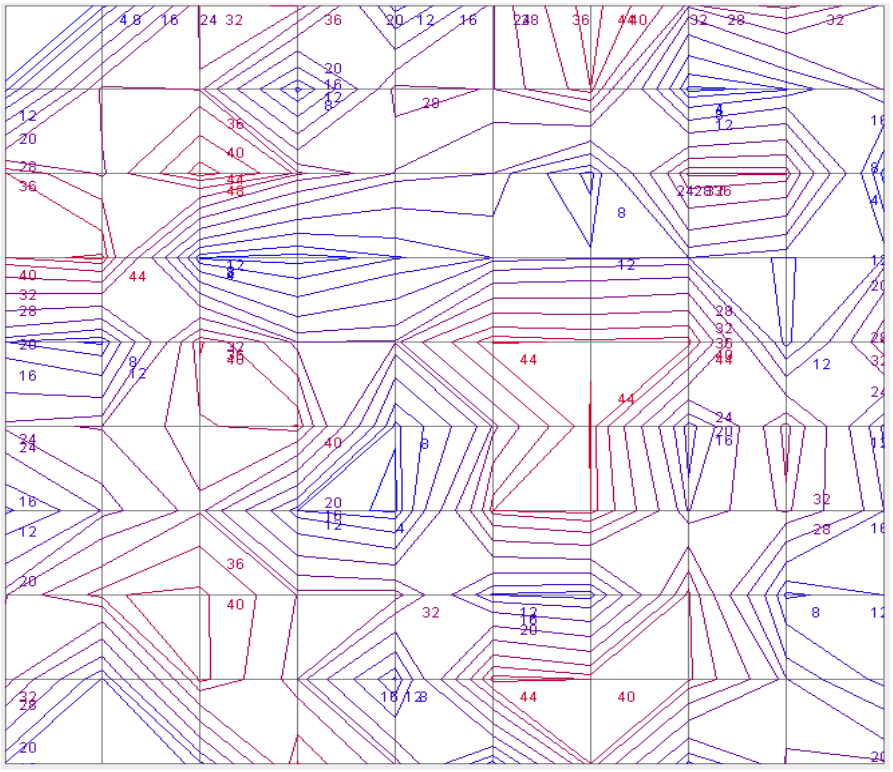

Figure 4. Contouring Map that Uses Cubic Interpolation

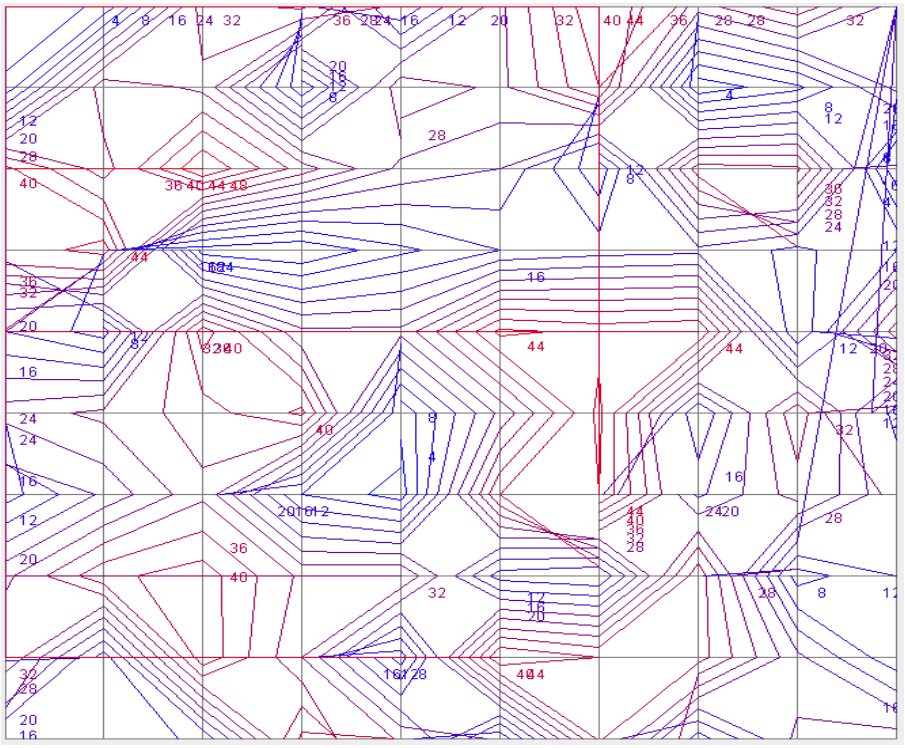

Figure 5. Contouring Map that Uses Inverse Distance Weighting Interpolation

Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.com.ng

5. CONCLUSION

To comprehend the numerical model behavior of any form of result/data, a graphical representation is required. Contour plotting of data is one step toward this type of representation. Algorithms that are used in drawing contour map use linear interpolation in determining the intersection point between contour lines and grid segments. But the use of linear interpolation in determining intersection point is not very precise and results in discontinuities at end points. This paper studied the application of inverse distance weighting interpolation in determining the intersection point between contour lines and grid segments. An application using java programming language was developed to implement the application of linear, cubic and inverse distance weighting interpolations in determining the intersection point between the contour lines and grid segments in the contouring plotting algorithm. Each of these interpolation methods was used to represent randomly generated values. Comparison was made between the maps produced by these algorithms, and it was observed that the map produced by inverse weighting interpolation is wrong because different contour lines are crossing each other and the map produced by cubic interpolation contains less information (i.e., a lesser number of contour lines). This shows that the map produced by linear interpolation shows more information and contour lines.

References

Abdulrazaq Abdulrahim, Mohammed Yahaya Tanko, Muhammad Aminu Umar, Sheidu Salami Tenuche & Aliyu Muhammad Kufena (2021). Investigating Cubic Interpolation in Contouring Algorithm., the 2021 International Conference on Computing & Advances in Information Technology, Zaria, Nigeria., 15–17 November, pp 181–184, 2021.

Aramini, M. (2012). *IMPLEMENTATION OF AN IMPROVED CONTOUR PLOTTING ALGORITHM*. https://www.semanticscholar.org/paper/IMPLEMENTATION-OF-AN-IMPROVED-CONTOUR-PLOTTING-Aramini/cb58ca5d6abe9e50b346f6b9c31dd88d875da126

Davis, J. C. (n.d.). *CONTOURING ALGORITHMS*. 8.

Davis, J. C. (1987). Contour Mapping and Surface Ii. *Science*, *237*(4815), 669–672.

Ekoule, A. B., Peyrin, F. C., & Odet, C. L. (1991). A Triangulation Algorithm from Arbitrary Shaped Multiple Planar Contours. *ACM Transactions on Graphics*, *10*(2), 182–199. https://doi.org/10.1145/108360.108363

Goldani-Moghaddam, H., & Enright, W. H. (2008). Efficient Contouring on Unstructured Meshes for Partial Differential Equations. *ACM Transactions on Mathematical Software*, *34*(4), 19:1-19:25. https://doi.org/10.1145/1377596.1377599

Interpreting map features – Bushwalking 101. (n.d.). Retrieved May 30, 2022, from http://www.bushwalking101.org/interpreting-map-features/

Khouni, I., Louhichi, G., & Ghrabi, A. (2021). Use of GIS based Inverse Distance Weighted interpolation to assess surface water quality: Case of Wadi El Bey, Tunisia. *Environmental Technology & Innovation*, *24*, 101892. https://doi.org/10.1016/j.eti.2021.101892

Meek, J. L., & Beer, G. (1976). Contour Plotting of Data Using Isoparametric Element Representation. *International Journal for Numerical Methods in Engineering*, *10*(4), 954–957. https://doi.org/10.1002/nme.1620100421

Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.com.ng

Ouabo, R. E., Sangodoyin, A. Y., & Ogundiran, M. B. (2020). Assessment of Ordinary Kriging and Inverse Distance Weighting Methods for Modeling Chromium and Cadmium Soil Pollution in E-Waste Sites in Douala, Cameroon. *Journal of Health and Pollution*, *10*(26), 200605. https://doi.org/10.5696/2156-9614-10.26.200605

Paul, B. (1987). *CONREC - A Contouring Subroutine*. http://paulbourke.net/papers/conrec/

Paul, B. (1999). *Interpolation methods*. http://paulbourke.net/miscellaneous/interpolation/

Rand, D. (1997). *Contour Plotting In Java*. http://preserve.mactech.com/articles/mactech/Vol.13/13.09/ContourPlottinginJava/index.html

Singh, C., & Saini, J. S. (2011). A SIMPLE AND FAST CONTOUR PLOTTING ALGORITHM FOR LINEAR 2D AND 3D ELEMENTS. *International Journal of Engineering Science and Technology*, *3*(4), 7.

Snyder, W. V. (1978). Algorithm 531: Contour Plotting [J6]. *ACM Transactions on Mathematical Software*, *4*(3), 290–294. https://doi.org/10.1145/355791.355800

Wang, H., Zhou, C., Wang, H., Hu, B., & Liu, Z. (2021). A Novel Contact Model for Rough Surfaces Using Piecewise Linear Interpolation and Its Application in Gear Wear. *Wear*, *476*, 203685. https://doi.org/10.1016/j.wear.2021.203685

Wu, Y.-C., Hsu, K.-L., Liu, Y., Hong, C.-Y., Chow, C.-W., Yeh, C.-H., Liao, X.-L., Lin, K.-H., & Chen, Y.-Y. (2020). Using Linear Interpolation to Reduce the Training Samples for Regression Based Visible Light Positioning System. *IEEE Photonics Journal*, *12*(2), 1–5. https://doi.org/10.1109/JPHOT.2020.2975213

Xiao-Ping, R., Xue-Tao, Y., Jin, L., Muhammad Aqeel, A., & Xian-Feng, S. (2016). An algorithm for generation of DEMs from contour lines considering geomorphic features. *Earth Sciences Research Journal*, *20*(2), 1–9. https://doi.org/10.15446/esrj.v20n2.55348

Yeo, M. F. (1984). An Interactive Contour Plotting Program. *Engineering Computations*, *1*(3), 273–279. https://doi.org/10.1108/eb023583