Applied Physics Research; Vol. 5, No. 1; 2013
ISSN 1916-9639 E-ISSN 1916-9647
Published by Canadian Center of Science and Education

Development of Software for Making Contour Plot Using Matlab to Be Used for Teaching Purpose

Lilik Hasanah[1], Mimin Iryanti[1], Nanang Dwi Ardhi[1] & Selly Feranie[1]

[1] Jurusan Pendidikan Fisika, Fakultas Pendidikan Matematika dan Ilmu Pengetahuan Alam, Universitas Pendidikan Indonesia, Indonesia

Correspondence: Lilik Hasanah, Jurusan Pendidikan Fisika, Fakultas Pendidikan Matematika dan Ilmu Pengetahuan Alam, Universitas Pendidikan Indonesia, Jl. Dr. Setiabudhi No 229, Bandung 40135, Indonesia. Tel: 62-22-200-4548. E-mail: lilikhasanah@upi.edu

Received: August 21, 2012 Accepted: September 10, 2012 Online Published: January 9, 2013

doi:10.5539/apr.v5n1p78 URL: http://dx.doi.org/10.5539/apr.v5n1p78

The research is financed by Indonesia university of Education Grant

Abstract

There is several commercial software used to make contour plot, such as Surfer 8. While those softwares provide complete and advanced features, it is not easy to operate and not providing a basic understanding on the internal work of creating contour plot. For teaching purposes, those softwares are not the best choice because of that reasons, other than the high pricing. In this research, we develop a easy to use software to make contour plot, optimized for teaching purposes so that students can see the step by step of creating contour plot using common method. The result is software that can match the Surfer 8 at 78.29% accuracy, yet easy for student to operate.

Keywords: contour plot, surface plot, Matlab, Surfer, data grid, triangulation, convex hull

1. Introduction

For Earth Physics students, Surfer has been used as a standard software package for mapping needs. On latest version of the software, Surfer 8, an extensive list of features are introduced, some of which are intended for advanced users. While it is good to have such complete tools, the software itself is not easy to use and require steep learning curve especially for new students. Students need to know and master some basic to intermediate knowledge of geological mapping. It would be convenience if when the students learn the basic of geological mapping there is a software tool which can assist the student about the steps of making various plots. This software has to be simple enough to use by students while still accommodate the teacher's need to present teaching material easily. Surfer 8 can be used, but in this research we focus on developing low-cost, easy to use and simple software.

We will use Matlab programming environment and language to develop our software. Every steps of creating contour plot, from data preprocessing until creating plot graph, is done using Matlab functions. Surfer 8 will be used only to test the result of our software to determine the level of match between contour plot from our software and from Surfer 8.

Related research whose topic is close to our research is presented by Wang, Liang and Li (2010), who made a software for batch-processing of contour map by using Matlab, which calling external Surfer software to do the data processing and generating contour plot is done by Surfer. This research differs to our research in the way that this research still depend on external Surfer 8 software, while our research use Matlab solely to solve problems in data processing, gridding, and generating contour plot graph.

1.1 Contour Plot

A contour plot is a technique for drawing surface with similar height z, on a 2 dimensional coordinates x and y. For a given z value, lines are drawn from a point in (x, y) coordinates to the next point which has the same z value. The contour plot is one form of alternative to a 3-D surface plotting.

Contour plot as shown in the Figure 1 above, shows an example of surface area that has symmetric contour and has peak in the center of the surface.

The contour plot is consists of:

- Vertical axis: Independent variable 2 (y value)
- Horizontal axis: Independent variable 1 (x value)
- Lines: iso-response values (values with the same z value)

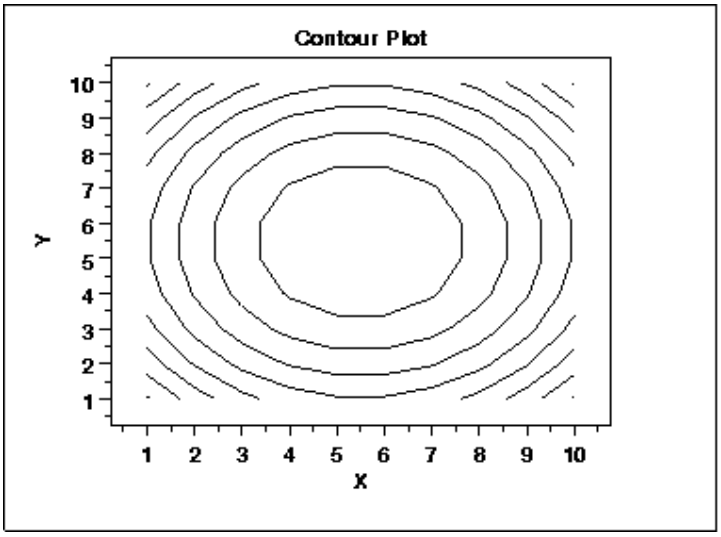

Figure 1. Contour plot

This contour plot shows that the surface is symmetric and peaks in the center.

Each of the independent variables is usually restricted to a regular grid of interest. The actual techniques to determine the correct iso-response values or the z values are usually complex and rely to computer simulation to numerically generate the data. Other variable may be needed to calculate the Z values for drawing the contour lines. Some plotting software requires explicit values for this. Other software has internal function to determine the variable automatically. If the available data do not form a regular grid, typically we need to do an interpolation to form them in a regular grid.

To understand the data from experiments or real survey, first we have to perform data preparation. For one-dimensional data, we need to do a create sequence plot and a histogram plot. For 2-dimensional data, a scatter plot is commonly created first. For 3-dimensional data, 3D surface plot or contour plot should be made, especially for large data sets.

Contour plots is a standard tools provided in most statistical software programs. Several graphics and mathematics program are also including this tool as well. These programs can have varied capabilities for the contour plots they generate. Some of them are just providing a basic contour plot in a rectangular grid while other software can have additional features such as color filled plot or shaded contours. Surfer, a software for plotting contour plot, supports broad contour plot functionalities and used widely. While intermediate to advanced users can use Surfer to draw contour plot to serve their purpose, most Earth Physics students who are still learning the basics of topography mapping and contouring could experience difficulties to interpret various plots produced by Surfer software. A more simple and easy to understand software is needed to teach these students the logic and practical way of making 3D surface and contour plots. In this work, we used Matlab to develop the software as Matlab provides very extensive library we needs for statistical calculations and has outstanding graphical capabilities.

Matlab (Matrix laboratory) is software system package to do numerical computations and graphics easily and interactively. Matlab is specifically designed to perform matrix computations: solving various systems of linear and non linear equations, compute eigenvalues and eigenvectors, matrices factor, and so many others. Additionaly, it has an extensive capabilities of graphical tools, and can easily be extended through programs that written in its own programming language.

1.2 Matlab Contouring Algorithm

As described in (The MathWorks Inc., 2012a), contour plot consists of several levels of contour lines. With input vector v, the elements of v are the contour level values, and the length of v is the number of contour levels that will be plotted by the Matlab function. If v is not supplied, the function algorithm will choose less than 20 contour levels that are divisible by 2 or 5.

The matrix Z and its corresponding X and Y matrices represents each point in Z at each row and column. When these matrices are unspecified, these matrices are inferred. The row and column length can be varies, although they are typically constant, or in other words Z is a regular grid.

Some contouring function in Matlab, such as contourc and contour, pads the height matrix Z with an extra row or column on each edge. It then assigns z-values to the added grid cells that are well below the minimum value of the matrix. The padded values enable those functions to close at the matrix boundary so that they can be filled with color. It also replaces the x, y coordinates containing the low z-values with NaNs to prevent contour lines that pass along matrix edges from being displayed in the final result. This is why contour matrices returned by calling contourf function sometimes gives NaN values. To know whether the dataset that we want to plot the contour map producing NaN values, we can analyze the dataset using Delaunay triangulation to obtain Convex Hull of our data. Data points that lie outside the Convex Hull will produce NaN values. See "Convex Hull" section below.

For a range of [min(Z), max(Z)], assume the current level c is equal to the lowest level of contour to be plotted. The algorithm will checks one edge at a time of every square in the defined grid to check whether c is lies between two z values for the edge points. If it is so, a contour line at c level crosses the edge, and a linear interpolation is then calculated:

$$t=(c-Z_0)/(Z_1-Z_0)$$

where Z_0 is the z value at a edge point, and Z_1 is the z value at the other point.

We continue with indexing a new contour line (i = 1) for level c by interpolating x and y:

$$cx(i) = X_0+t*(X_1-X_0)$$

$$cy(i) = Y_0+t*(Y_1-Y_0)$$

Iterate around the edges of the square; the contour plot ends at the next edge with z values that bracket c. For the next i, we compute t for the edge, and then compute cx(i) and cy(i), as above.

Then mark the square as having been visited and repeat calculating the above variables. If the square is already visited, then the contour line closes at that point. Continue with copying cx, cy, c, and i to the contour line data structure (the matrix returned by contouring functions, described shortly).

Clear all the visited markers, go for the next contour level, and repeat until c reach max (Z).

Extra logic is needed for squares where a contour line plot passes through all four edges (saddle points) to determine which pairs of edges to connect.

1.3 Convex Hull of Finite Dataset

The convex hull of a point set P is the smallest convex set that contains P. If P is finite, the convex hull defines a matrix A and a vector b such that for all x in P, Ax+b <= [0, 1, 2, 3, ...] [National Science and Technology Research Center for Computation and Visualization of Geometric Structures (The Geometry Center, 1993)]. The convex hull of a set of data points is the smallest convex set containing the points.

If the points are all on a line, the convex hull is the line segment joining the outermost two points. In the planar case, the convex hull is a convex polygon unless all points are on the same line. Similarly, in three dimensions the convex hull is in general the minimal convex polyhedron that contains all the points in the set.

When the set X is a nonempty finite subset of the plane (that is, two-dimensional), we may imagine stretching a rubber band so that it surrounds the entire set X and then releasing it, allowing it to contract; when it becomes taut, it encloses the convex hull of X (Knuth, 1992).

In Figure 2, left image is convex set with 2 points (P and Q) inside the convex hull. Any other points that lay outside of the polyhedron are said to be outside of the convex hull.

Convex hull is computed to analyze whether out dataset has points that will produce NaN value when Matlab create contour matrix. Data points which lay outside of the computed convex hull are expected to give NaN

value in contour matrix (The MathWorks Inc., 2012b). These NaN values are not acceptable when we plot them in Matlab, because it will be ignored and will not be shown in the plot, which bring inaccuracies to the contour plot.

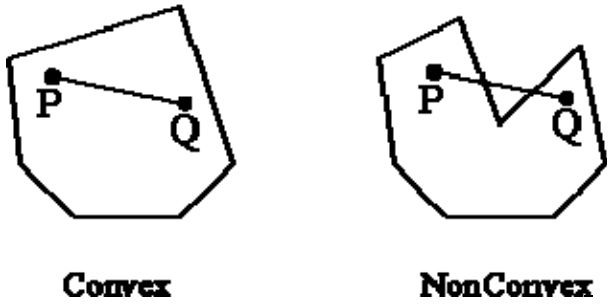

Figure 2. Convex and Non Convex sets

Left image is convex set with 2 points (P and Q) inside the convex hull. Any other points that lay outside of the polyhedron are said to be outside of the convex hull.

To prevent this NaN values in contour matrix, Matlab provides a method to pre-process our datasets before we compute the contour matrix, or data gridding process. See "Data Gridding" section below.

MATLAB provides two ways to compute the convex hull (The MathWorks Inc., 2012a):

- Using the MATLAB functions *convhull* and *convhulln*
- Using the convexHull method provided by the DelaunayTri class

The *convhull* function supports the computation of convex hulls in 2-D and 3-D. The *convhulln* function supports the computation of convex hulls in N-D ($N \geq 2$).

The DelaunayTri class supports 2-D or 3-D computation of the convex hull from the Delaunay triangulation. This computation is not as efficient as the dedicated convhull and convhulln functions. However, if you have a DelaunayTri of a point set and require the convex hull, the convexHull method can compute the convex hull more efficiently from the existing triangulation.

In this paper we will use the DelaunayTri class to compute the convex hull.

1.4 Data Gridding

Gridding is a method of interpolating data from an arbitrary 2D sampling pattern to a uniform grid. Matlab provides a function called *griddata* to do interpolation of scattered data such as our dataset. *Griddata* function fits a surface plot in the form g = f(x, y) to the scattered data in the vectors (x, y, v). The *griddata* function interpolates the surface at the query points specified by (xq, yq) and returns the interpolated values, gq. The surface always passes through the data points defined by x and y.

Griddata function has several interpolation methods that we can use, that are:

1) Linear interpolation (default)
2) Cubic interpolation
3) Natural neighbor interpolation
4) Nearest neighbor interpolation
5) v4 interpolation method

The methods above define the type of surface fitting to the data. The 'cubic' and 'v4' methods can produce smooth surfaces while 'linear' and 'nearest' have discontinuities in the first and zero'th derivatives, respectively. All the methods except 'v4' are based on a Delaunay triangulation of the data.

Occasionally, griddata might return points on or very near the convex hull of the data as NaNs. This is because round-off in the computations sometimes makes it difficult to determine if a point near the boundary is in the convex hull (The MathWorks Inc., 1996).

'v4' is apparently a Greens' function approach as described by Sandwell and David (1987). It uses a full matrix

composed of all of the interpoint distances, so it will extrapolate smoothly without producing NaN value, even if the points are outside the convex hull. This will be slowing the computation for many points, and extremely memory intensive.

The griddata using 'v4' method uses the method proposed by Sandwell and David (1987). The other griddata methods are based on a Delaunay triangulation of the data that uses Qhull (Barber, Dobkin, & Huhdanpaa, 1996) (The Matworks Inc., 1996).

2. Methodology

We develop the software for plotting contour using Matlab. We started by collecting data from field measurement where we have 35 data points collected. These field data consists of location, altitude, and measured gravity data. It then manually processed to calculate several gravity corrections, which used to obtain corrected gravity data. These final data is made available in Microsoft Excel format, consist of three columns, that are latitude and longitude coordinate of each measurement points, and corresponding gravity value. This Excel data will be used as our data input to the software.

Figure 3 describes overall workflow of our research. The software we want to develop is on the left side of the image, and we use Surfer 8 software as a benchmark to analyze the outcome of the software. Data input is applied to both our software and Surfer 8 and then we calculate the error matrix from the output difference of the software and the Surfer 8. Comparison is also done on resulted plots, both surface and contour plot, to show their similarity.

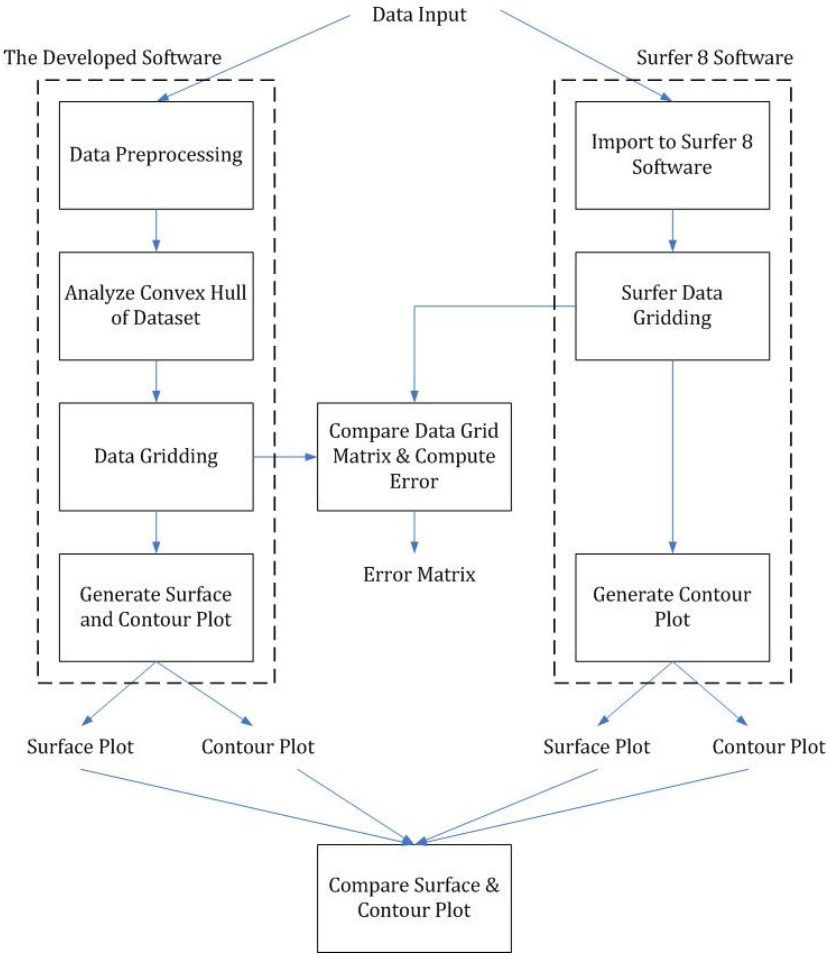

Figure 3. Software development workflow

The software we want to develop is on the left side of the image, and we use Surfer 8 software as a benchmark to analyze the outcome of the software. Data input is applied to both our software and Surfer 8 and then we calculate the error matrix from the output difference of the software and the Surfer 8. Comparison is also done on resulted plots, both surface and contour plot, to show their similarity.

2.1 Data Preprocessing

Even our data input had previously manually processed, it still need to be checked to possible duplication, so we start by data cleaning process. After we get unique dataset, we assign each data column into its corresponding variables. Each data points is unique related to its position in x – y coordinate, so we define a grid mesh in x-y plane to map the position of each data point. The value of data represents the z value in the grid mesh, so each data positioned in 3D space. Data input plot is shown in Figure 4.

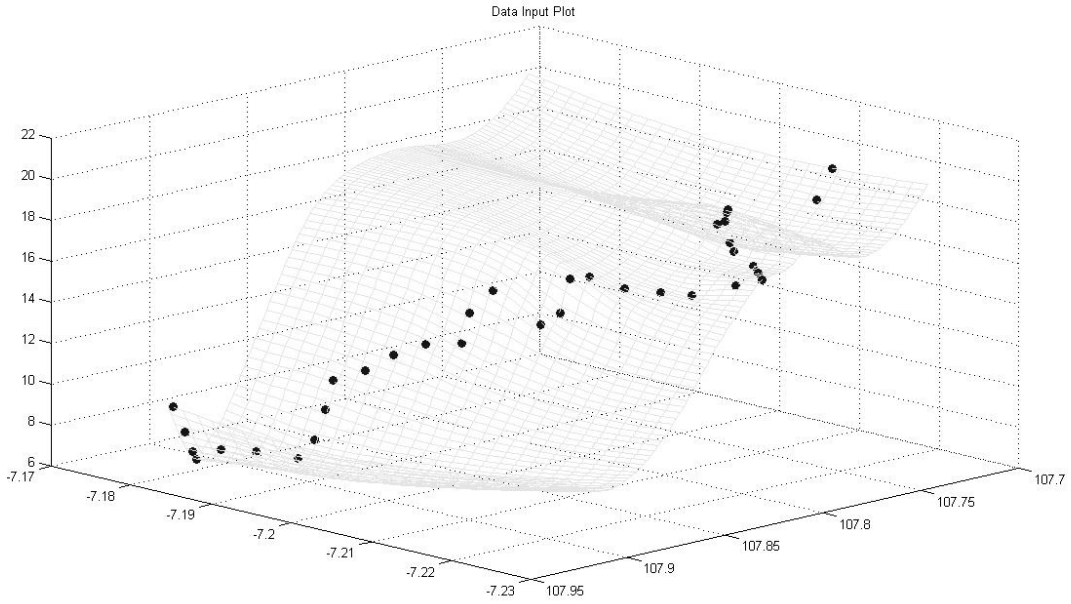

Figure 4. Processed data input points plotted in 3D space, with added surface mesh grid

2.2 Analyze the Convex Hull of Input Dataset

Using Delaunay Triangulation function provided in Matlab, we can analyze the data input set to determine how many points are outside of the convex hull. As stated before, these points will decide which method we should use when performing data gridding process. Data gridding process will fail if the chosen gridding method failed to interpolate to the points outside the convex hull.

In Figure 5, gray-colored box is the convex hull of our data input. Black dots represent data input, and dots inside the box are considered as data points that lay inside convex hull. We can then calculate how many points inside the convex hull, and based on the calculation in our software, from 35 data points, the number is 25 points inside convex hull. Or, in other word, there are 10 data points that lay outside of the convex hull.

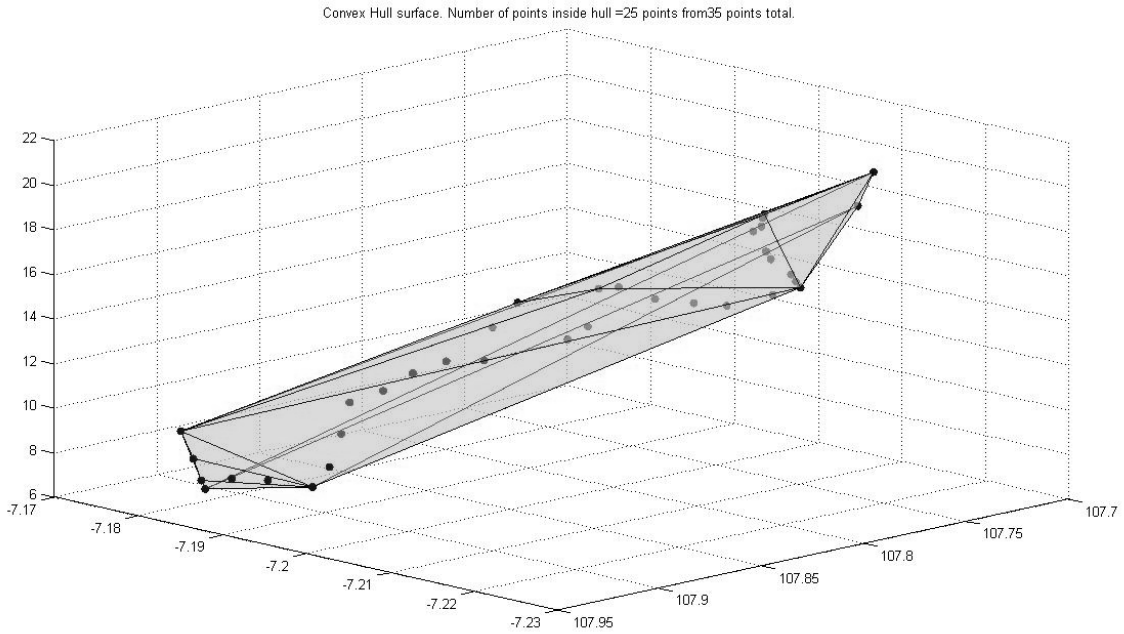

Figure 5. Convex hull of data input. Black dots are data input

2.3 Data Gridding

Data gridding is a method of interpolating data from an arbitrary 2D sampling pattern to a uniform grid. We use Matlab's *griddata* function to perform data gridding to our data input set. Based on our convex hull analysis, there are 10 data points lay outside of the convex hull, so we choose to use 'v4' method inside *griddata* function as way to prevent it to produce undesirable output value (NaN, Not a Number). NaN value will be discarded by Matlab when plotting the surface and contour plot.

v4 data gridding method will perform full matrix calculation for all points in our data input set, so when performing calculation on large datasets, our software require a lot of computing resources and time to finish the calculation. Since we have only 35 data input, the calculation only take several seconds to finish on computer with decent specification.

Our gridding process produces dataset that consist of a {100 x 28} matrix of data or about 2800 data points. V4 method interpolates those numbers of data points from the available 35 points as data input.

3. Result and Discussion

The software is developed to be a Graphical User Interface (GUI) application with easy to use and understand design. Students who will use this software to learn how to plot contour map and internal working of plotting process will find it easy to operate the software. Figure 6 is the sample of software's GUI screen showing the surface and contour plot of given data input.

Comparing the contour plots produced by the software and Surfer 8 as benchmark, we found that both are in good agreement. In Figure 7, left plot (A) is contour plot produced by our software while the plot on the right or (B) produced from Surfer 8 software.

The gravity value scale on both contour plots are almost identical, from around 6 to the highest around 20.5. However the color scale on both plots is cannot reliably used as the basis of deciding similarity between both plots, since Matlab produce the color scale algorithmically while, on the contrary, in Surfer 8 we need to manually set the color scale. Same color could mean different real value on both plots, so in this paper, we use the similarity in iso-lines in the contour plot as guidance to decide.

To see the difference of plots produced by our software with plots produced from Surfer 8 software, we can also compare the surface plots, as shown in the Figure 8.

In Figure 8, we show the difference of surface plot from both softwares. Our software produce surface plots (surface without grid lines) with lower value in the valley at coordinate 107.9 and 107.75, compared to Surfer 8'

surface plot. The same case is also happen to the peak at coordinate 107.8, where our software produce higher peak compared to peak produced by Surfer 8.

As we discussed earlier, our data gridding process produce 2800 data points in a {100 x 28} matrix and these data points are the points plotted in the surface plot in Figure 8 above. Surfer 8 software is also produce 2800 data points, so we can compare both data to calculate how much the difference between both plots is quantitatively.

We define the relative error as the difference between value A and B relative to the reference value B. In this research, we define the relative error to be 10% maximum, so value A is considered to match to the value B if the relative error is lower than 10%. Using this approach and represent A as data from our software and B as data from Surfer 8 calculation, we can calculate relative error of all 2800 data points produced by our software to the one produced by Surfer 8 software. Calculation done in Matlab, and the result is that of 2800 points, 2192 points in our matrix has less than 10% relative error to the Surfer 8 data matrix, or around 78.29%. In other word, both surface plots shown in Figure 8 have similarity at 78.29% confidence level.

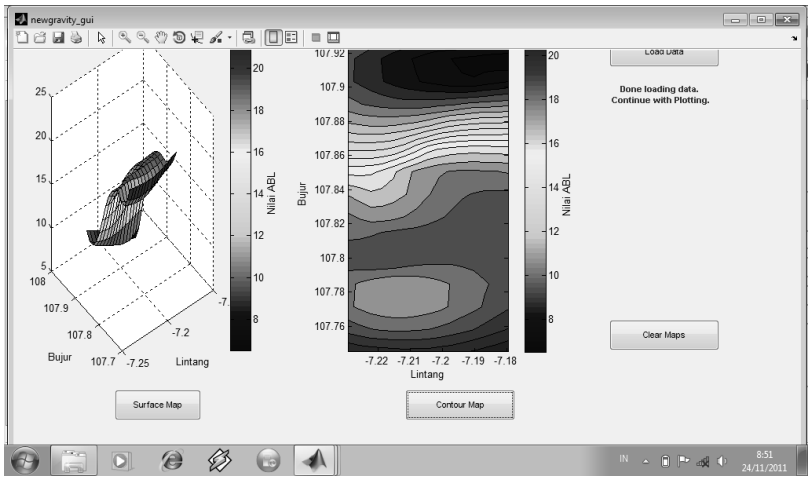

Figure 6. Sample of software's GUI screen

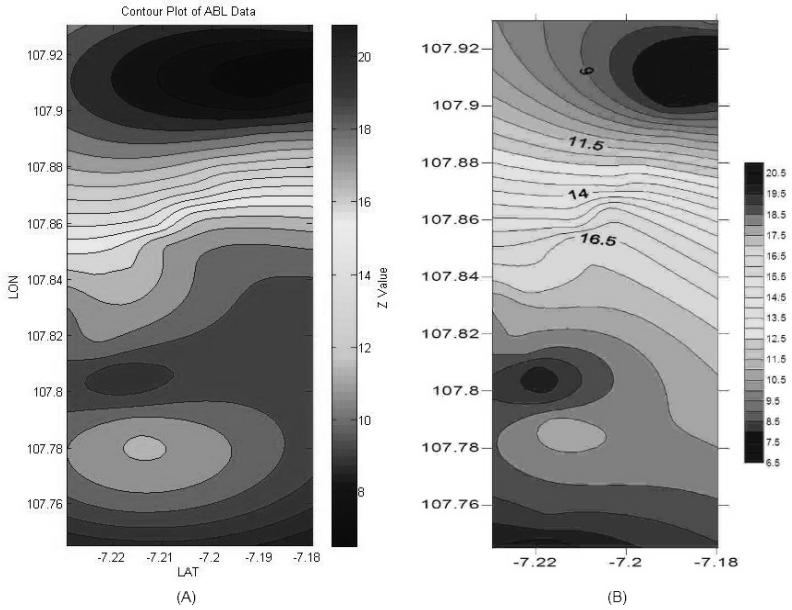

Figure 7. (A) Contour Plot produced by the Software; (B) Contour Plot produced by Surfer 8

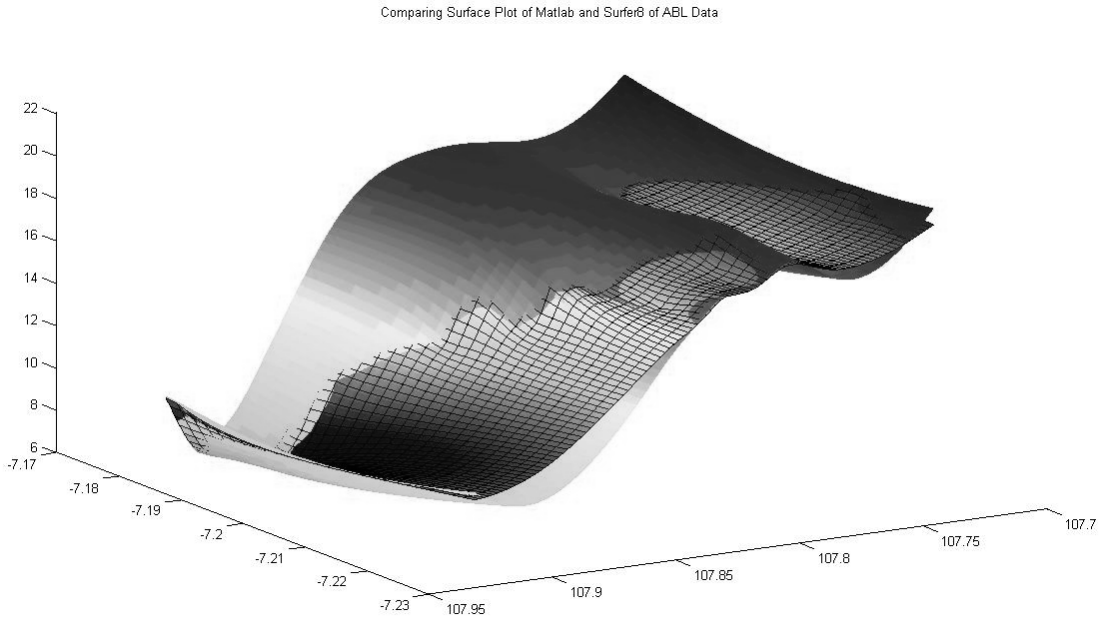

Figure 8. Surface Plot from our software (without grid lines) and from Surfer 8 (with grid lines)

4. Conclusion

In this research we are successfully developed software for plotting contour map from a given data input as an easy to use tools for teaching Earth Physics student. Standard plotting software Surfer 8 is used as a benchmark and our software achieve good result at 78.29% match. Further research can be done to increase the percentage of match, especially to find better data gridding or interpolation method than current 'v4' method used in this research.

References

Barber, C. B., Dobkin, D. P., & Huhdanpaa, H. T. (1996). The Quickhull Algorithm for Convex Hulls. *ACM Transactions on Mathematical Software, 22*(4), 469-483. http://dx.doi.org/10.1145/235815.235821

Knuth, D. E. (1992). *Axioms and hulls, Lecture Notes in Computer Science* (pp. ix+109, 606). Heidelberg: Springer-Verlag. http://dx.doi.org/10.1007/3-540-55611-7

National Science and Technology Research Center for Computation and Visualization of Geometric Structures (The Geometry Center). (1993). University of Minnesota.

Sandwell, D. T. (1987). Biharmonic Spline Interpolation of GEOS-3 and SEASAT Altimeter Data. *Geophysical Research Letters, 14*(2), 139-142. http://dx.doi.org/10.1029/GL014i002p00139

The MathWorks. Inc. (1996). *Matlab Language Reference Manual* (1st ed., pp. 2-342).

The MathWorks. Inc. (2012a). *Matlab R2012a Graphics User Guide* (pp. 5-73).

The MathWorks. Inc. (2012b). *Matlab R2012a Mathematics User Guide* (pp. 6-67).

Wang, P., Liang, S., & Li, W. G. (2010). *Automatic Batch-Processing of Electric Contour Map by Using Matlab Programming with Calling Surfer Software*. Retrieved from http://en.cnki.com.cn/Article_en/CJFDTOTAL-WTHT201001019.htm