International Journal of Engineering, Mathematical and Physical Sciences
ISSN: 2517-9934
Vol:5, No:8, 2011

Simulating Gradient Contour and Mesh of a Scalar Field

Usman Ali Khan, Bismah Tariq, Khalida Raza, Saima Malik, Aoun Muhammad

***Abstract*—**This research paper is based upon the simulation of gradient of mathematical functions and scalar fields using MATLAB. Scalar fields, their gradient, contours and mesh/surfaces are simulated using different related MATLAB tools and commands for convenient presentation and understanding. Different mathematical functions and scalar fields are examined here by taking their gradient, visualizing results in 3D with different color shadings and using other necessary relevant commands. In this way the outputs of required functions help us to analyze and understand in a better way as compared to just theoretical study of gradient.

***Keywords*—**MATLAB, Gradient, Contour, Scalar Field, Mesh

I. Introduction

THE motive of this paper is to study about the gradient of scalar fields, its working and functionality and to have a look over its vast applications in different fields of life. For better understanding we took the help of MATLAB and its different commands to simulate gradient of different functions and scalar fields so that we can have a visual study of our topic.A brief introduction of term gradient tells us that it is a Latin word and refers to '*gradual change*' in any quantity. [2]The term gradient (grad) typically refers to the derivative of functions of more than one variable. Gradient of a scalar field is obtained by applying *del* ∇ operator on a scalar field expression, that gives the direction to that field in its increasing dimension, as illustrated by the picture below

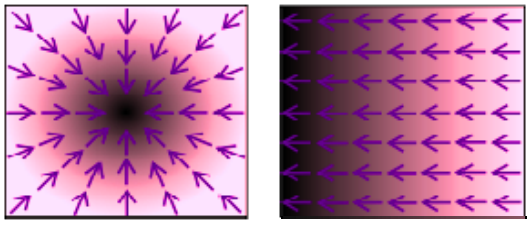

Fig. 1 Direction of gradient of field

In this image, the scalar field is in black and pink, black representing higher values and its gradient is represented by the purple arrows.

Using MATLAB we can draw the mesh or the surfaces of the gradient of the mathematical functions representing different scalar fields. For example the following image is the drawing of the function: $z = x^2 - y^2$. The different tools and commands executed to obtain this surface will be discussed later in the paper.

Mr.Usman Ali Khan is with "University college of Engineering and Technology UCET The Islamia University of Bahawalpur".

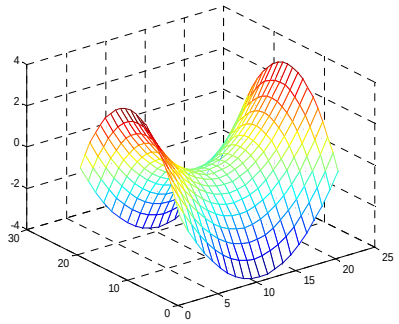

Fig. 2 Direction of gradient of field

II. Gradient

The *gradient* of a scalar field *V* is a vector that represents the magnitude and the direction of the greatest increase rate of *V*. Gradient is a vector (a direction to move) that points in the direction of greatest increase. [3]The gradient of any function or scalar field *F* can be given by following mathematical equation:

$$\text{gradF}(x,y,z) = \nabla F(x,y,z) = (dF/dx, dF/dy, dF/dz)$$

The *gradient* of an R^3 scalar field is an R^3 vector field. In 4-D the *four-gradient* is a vector, formed as a derivative of four scalars. The gradient can also be used to measure how a scalar field changes in other directions, rather than just the direction of greatest change, by taking a dot product. The gradient is zero at a point of maximum location because there is no further direction of increase, rather there is a decrease rate in any direction under consideration. It's just like standing on the top of a mountain and there is a downhill any direction you move. [5]Also the gradient of a vector or a vector field makes no sense and is meaningless because a vector quantity already has a predefined direction and cannot be given a second direction, i.e., a vector cannot have more than one direction.
Above its complexity or convenience issues, gradient is tremendously useful and is applicable in most of the life matters and science.

III. Practical Applications

Gradient can be applied in a large variety of studies such as Engineering field, biomedical field, Astrological fields, Atmospheric studies, etc.

A. *Gradient* is used for the measurement of the steepness of a road and thus is very useful in civil engineering. Let a surface point at sea level (x,y) having a height H(x,y) so the gradient of H, basically a vector defining the direction of steepest slope at that point. And how much the steepness is defined by the magnitude of the gradient of vector H.

B. *Temperature gradient* shows that the particular direction at which the rate of temperature changes rapidly around a specific direction. The temperature gradient is mostly used in the applications of atmospheric science, meteorology and climatology.

C. *Geothermal gradient* is the rate of increasing temperature with respect to increasing depth in the Earth's interior. [6]It is 25–30°C per km of depth in most of the world. This is useful for generating power.

D. In *biomedical and biosensors*, it is used to investigate the protein absorption and cell adhesion. It is also used in fundamental and applied material science studies such as generation of polymers.

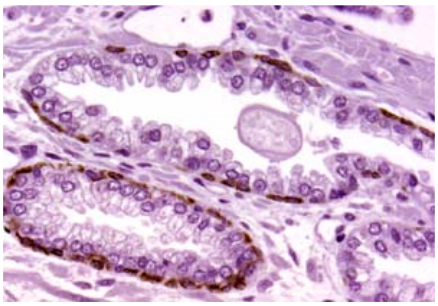

Fig. 3 Protein absorption cell

E. *Gradient* also simplify the development of new analytical techniques, since allowing the testing of many different surfaces under the same environmental conditions, thus minimizing the experimental errors.

F. *Gradien*t also serves as the templates for the demixing of polymer blends

IV. Relevant Commands

The term *MATLAB* stands for ***MAT****rix* ***LAB****oratory*. Therefore while working with MATLAB, both scalars and vectors are considered as in matrix form. Scalar corresponds to a single element, while a vector corresponds to a one dimensional matrix.

The commands that are used in this report, to execute gradient of different scalar fields, are given below along with their functions performed.

A. Gradient();

Its syntax is as [FX,FY] = gradient(F); and it is used to specify the maximum space rate of increase of a scalar field as told by its definition.

B. gradientm();

Its syntax is as follows;
[aspect,slope,gradN,gradE] = gradientm(map,refvec);

This command is used to compute the slope, aspect, and north and east components of the gradient for a regular data grid. The visual percept of the gradient of the function or the field is mentioned by aspect command. If we have the surface map contains elevations in meter, then the results of aspect and slope shows the units of the degrees clockwise from north and up from the horizontal. The component of gradient regarding north and east represents the change in the map variable per meter of distance in the north and east direction[7].

C. meshgrid();

Its syntax is as [X,Y] = meshgrid(x,y); and it is used to change the domain specified by vectors x and y into arrays X and Y, which can be used to evaluate functions of two variables and three-dimensional mesh/surface plots. [7]The rows of the output array X shows the copies of the vector 'columns of the output array Y are copies of the vector 'y'.

Code:

```
x=[2 4 6];
y=[5 6 7];
[X,Y]=meshgrid(x,y)
X =
   2   2   2
   4   4   4
   6   6   6
Y =
   5   5   5
   6   6   6
   7   7   7
```

D. contour();

Its syntax is as contour(Z); and it displays isolines of a matrix. [7]It will give the default number of contour levels depending upon the minimum and maximum values of Z.While the number of contour levels can be set according to the specifications by using contour(Z,n) command, where n is the specified number of contour levels.

Code:

```
v=-5:.5:5;
[x,y]=meshgrid(v);
z=x.*y;
[xx,yy]=gradient(z);
contour(z)
```

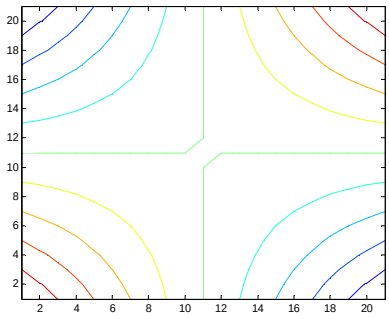

Fig. 4 2D Gradient

E. quiver();

Its syntax is as quiver(x,y,u,v); and it displays velocity vectors as arrows with components (u,v) at the points (x,y).
The command quiver(…,LineSpec) is used to change the colour and style of the velocity vectors. This style and colour is specified by the tool LineSpec.

Code:

```
v=-5:.5:5;
[x,y]=meshgrid(v);
z=x.*y;
[xx,yy]=gradient(z);
contour(z)
hold on
quiver(xx,yy)
hold off
```

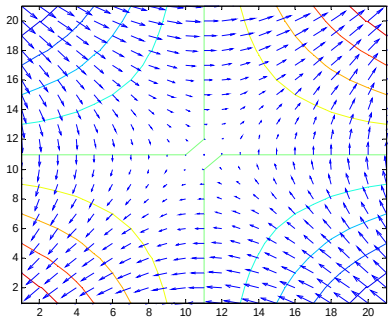

Fig. 5 2D Gradient by quiver command

F. surf();

Its syntax is as: surf(x,y,z); and it is used to view mathematical functions over a rectangular region.surf and surfc create colored surfaces.
The colour of the shaded surface can be changed by using surf(x,y,z,c) command, where c specifying the colour.

Code:

```
v=-5:.5:5;
[x,y]=meshgrid(v);
z=x.*y;
surf(z)
```

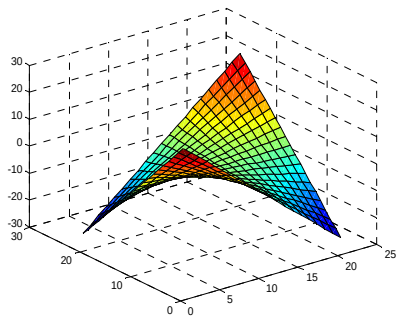

Fig. 6 2D Gradient by surf command

V. Simulating Gradient of a Scalar Function

Some simple codes are discussed to illustrate the working of the commands discussed earlier. Almost similar codes are executed by varying one or more parameters in them, so that the varying outputs could be examined.

A. Gradient of a scalar:

The following code represents the gradient of the equation of a circle or a circular field. This code contains the simple commands to illustrate their working through a simple output shown.

Code:

```
v=-2:.2:2;
[x,y]=meshgrid(v);
z=x.^2 + y.^2;
[px,py]=gradient(z,.5,.5);
contour(v,v,z)
hold on
quiver(v,v,px,py)
xlabel('x'); ylabel('y');
title('Gradient of field z=x^2+y^2')
hold off
```

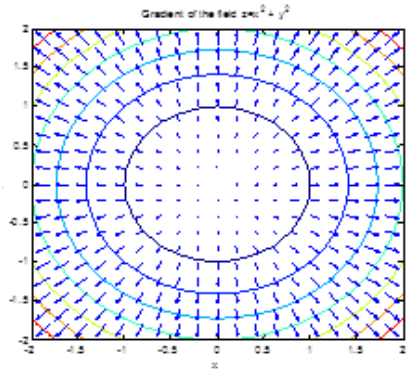

Fig. 7 2D Gradient of Scalar Circular Field at origin

Here the center of the circular field is at the origin O(0,0) but we can also change its center from origin to anywhere we want in the required scenario. In case when it is required to shift the center of this circular field to height h, we made a little variation in the circle equation in the code, as follows:

Code:

```
v=-2:0.2:2;
h=1;
[x,y]=meshgrid(v)
z=(x-h).^2 + (y-h).^2
[px,py]=gradient(z,.5,.5)
contour(v,v,z)
hold on
quiver(v,v,px,py)
xlabel('x');
ylabel('y')
title('Gradient of the field z=(x-h)^2+(y-h)^2')
hold off
```

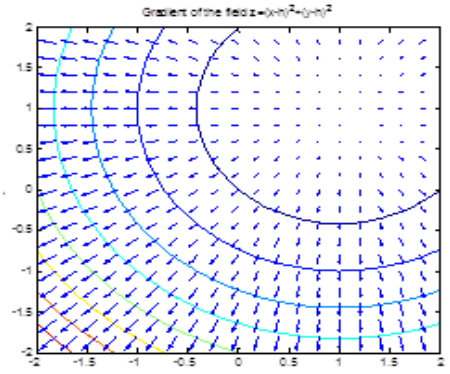

Fig. 8 2D Gradient of Scalar Circular Field at shifted origin

Hence the center is shifted from origin to height h(1,1).Similarly the center can be shifted to any height or depth from the origin by specifying the required height coordinates to the point h(a,b). Where a and b are the real numbers whose values can either be equal or different.

B. 3-D Contours:

This code represents the contours of the given function in 3-dimensional view by making use of contour3() command.

Code:

```
v=-2:0.2:2;
[x,y]=meshgrid(v)
z=x.*exp(-x.^2-y.^2)
[xx,yy]=gradient(z)
contour3(z)
hold on
quiver(xx,yy)
hold off
```

C. LineSpec tool:

In this code the colors and the line style of contours and quivers are changed by using the linespec tool as the following syntax;

contour(…,LineSpec) and quiver(…,LineSpec)

With the help of LineSpec tool, the line style can be changed to dashed line (--), or to the dotted line by using (:), or to the dash dot line style by using (-.).The color of line can also be changed by using the specified color coding words such as r for red, b for blue, m for magenta,

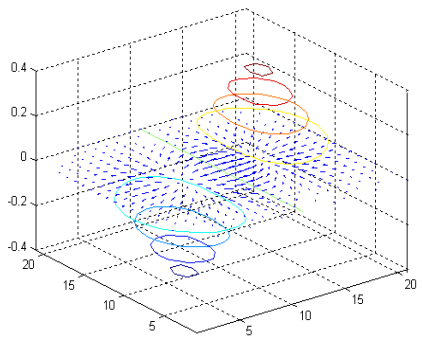

Fig. 9 3D contour

c for cyan color, etc. But the MATLAB has limited colors and line styles.

Code:

```
v=-5:.5:5;
[x,y]=meshgrid(v)
z=x+y
[px,py]=gradient(z,.2,.2);
contour(z,'-.r')
hold on
quiver(px,py,'c')
hold off
```

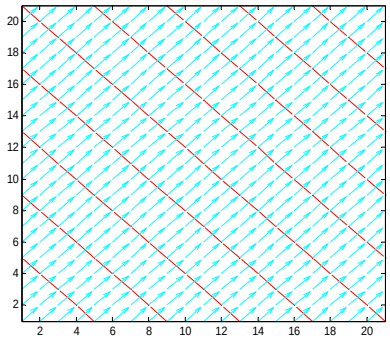

Fig. 10 3D contour vectors with line spacing

In this code a dashed (--) line indicates contour and its colour is set to red using *LineSpec* tool. Similarly with the help of this tool, the cyan color lines indicates the quiver vectors.

D. Mesh and Surface Drawings:

This code tells about how to draw 3D views of mesh and surfaces of the mathematical functions or the fields. This code contains the command of shading(); to assign a colour to the drawn surface or the mesh. Variation could be made by varying shading command to give it different shades.

Code:

```
v=-2:.2:2;
[x,y]=meshgrid(v);
z=x.^2 + y.^2;
[xx,yy]=gradient(z);
mesh(z)
hold on
quiver(xx,yy,'r')
surf(z)
shading interp
hold off
```

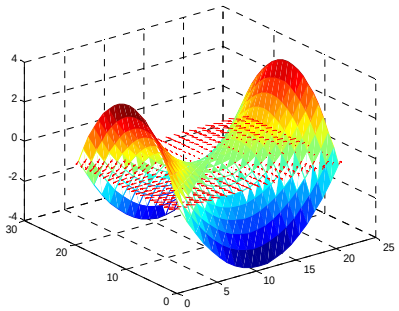

Fig. 11 3D Mesh

To examine the difference between the mesh() and surf()commands, we consider their graphs separately by using the subplot() command in the following code:

Code:

```
v=-2:.2:2;
[x,y]=meshgrid(v);
z=x.^2 - y.^2;
[xx,yy]=gradient(z);
subplot(1,3,1)
mesh(z)
hold on
quiver(xx,yy,'r')
hold off
subplot(1,3,2)
surf(z)
shading interp
hold on
quiver(xx,yy,'m')
hold off
```

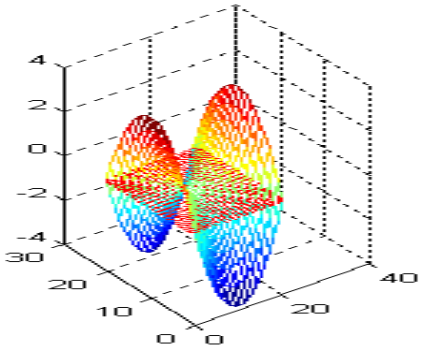

Fig. 12 3D Mesh

This figure represents the *mesh(z)* command, a net like hollow outline of function $z = x^2 - y^2$. While red arrows represent the quiver or the direction of the gradient of *z*.

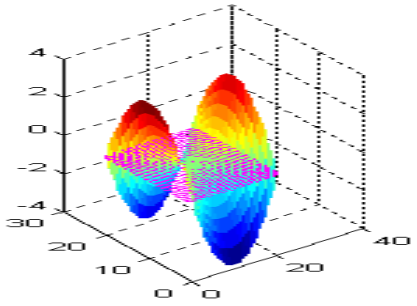

Fig. 13 3D Mesh

This image is the result of the surf(z) command which gives a solid surface of the function z = x2 – y2. And the magenta arrows are the direction of the gradient of z.

E. Simulating Gradient using gradientm() command[7]*:*

In this code we have used the MATLLAB peaks matrix and found out their gradient using gradient() command, so that we can also examine the east and north components of the gradient and also its slop and aspect[7].

Peaks is a function of two variables, obtained by translating and scaling Gaussian distributions, and its syntax is; Z = peaks(n); that returns an n-by-n matrix, and Z is specifying the function using peaks matrix. If the order n of the matrix is not defined, then it will return a 49 x 49 matrix by default.

In this code we have also made use of the worldmap(map,refvec,type) command that defines the geographical limits for the required data grid and that data grid is then displayed using the meshm() command.

Code:

```
datagrid = 500*peaks(100);
gridrv = [ 1000 0 0];
[aspect,slope,gradE] = gradientm(datagrid,gridrv);
figure
worldmap(gradE,gridrv,'none')
meshm(gradE,gridrv,size(datagrid),datagrid)
contourcmap(1,'jet','colorbar','on',...
'ylabelstring','Gradient (East Component)');
view(3)
daspectm meters
camlight
```

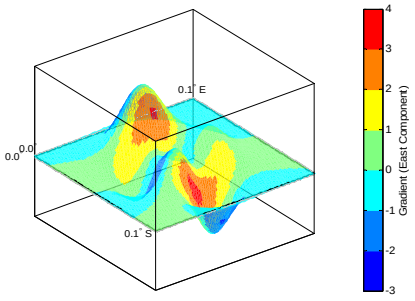

Fig. 14 3D contour with camlight

This image shows east component of gradient.In the next part of the code, the slope of the gradient is displayed.

```
clmo surface
meshm(slope,gridrv,size(datagrid),datagrid)
contourcmap(10,'hot','colorbar','on',...
'ylabelstring','Slope (degrees)')
shading interp
```

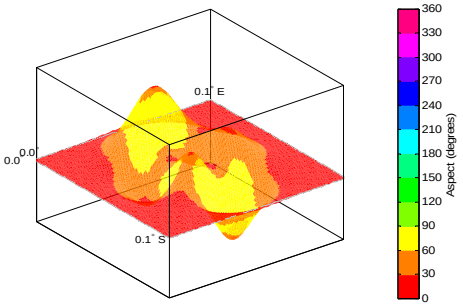

Fig. 15 3D contour with camlight and shading

VI. Conclusion

At the end of this paper, we can say for sure that gradient has a wide range of applications and uses in almost every field of life, listing industrial environments, professional and technical studies and biomedical fields, and it is serving the world of technicalities in a true sense.Moreover with help of MATLAB, the complexities of gradient are made far convenient by using its numerous powerful commands and tools.

References

[1] Matthew N. O. Sadiku, *Elements of Electromagnetics*, 3rd ed. Section 3.5, Gradient of a Scalar. Oxford University Press, 2001, pp. 20–56.

[2] Wilfred Kaplan, *Advanced Calculus*, 4th ed. Section 3.3, The Gradient Field. Addison-Wesley, 1991, pp. 145–165

[3] Schey, H. M. *Div, Grad, Curl, and All That: An Informal Text on Vector Calculus*, 3rd ed. New York: W. W. Norton, 1997,pp 220-260.

[4] N. von Ellenrieder, C. Muravchik, E. Spinelli, A. Nehoraiy, J. Roitmanz, W. Silvaz and S. Kochenz, *Performance of the Electroencephalography Inverse Problem using the Electric Potential Gradient Measurements.* IEEE 2003.E. H. Miller, "A note on reflector arrays (Periodical style—Accepted for publication)," *IEEE Trans. Antennas Propagat.*, to be published.

[5] Morse, P. M. and Feshbach, H. *Methods of Theoretical Physics*, Part I. New York: McGraw-Hill, 1953,ch-6,7

[6] David A. Rothstein, Craig E. Manning. *Geothermal gradients in continental magmatic arcs: Constraints from the eastern Peninsular Ranges batholiths, Baja California, México*, Geological society of America, USA 2003.

[7] *Fundamental of Electromagnetics with MATLAB*, by Karl.E.Lonngren and Sava V.Savov, Randay.J.Jost, 2nd ed. 2007.M. Young, *The Techincal Writers Handbook.* Mill Valley, CA: University Science, 1989.