



# JURNAL TEKNOLOGI MAKLUMAT DAN SAINS KUANTITATIF

## KANDUNGAN

## Muka Surat

To (Start With) OOP, Or Not OOP: That is Not The Question <i>Syed Ahmad Aljunid</i>	1
Penilaian Tingkah Laku Taklinear Menggunakan Kaedah Empirik <i>Norazan Mohamed Ramli Habsah Midi</i>	19
Towards Developing A Risk Charter for Software Development Projects <i>Noor Habibah Arshad</i>	37
Comparative Performance of Computational Techniques in Retrieving Malay Text <i>Zainab Abu Bakar</i>	51
Algorithm of Magnetic Flux Density on a Plane Generated by a Finite Length Current Source <i>Rashdi Shah Ahmad, Tahir Ahmad, Chew Soon Leong</i>	63
Suatu Kaedah Menganggar Kos Perisian Berasaskan Spesifikasi Formal <i>Abdullah Mohd Zin, Maridah Mohamad Shah, Abd Malik Md Yusof</i>	75
Solving a Constraint Satisfaction Problem by Backtracking Intelligently: A Case Study <i>Muthukkaruppan Annamalai</i>	87
An Empirical Investigation into the Critical Success Factors Used by it Companies of Various Sizes to Adopt Internet Technology <i>Lloyd Tam Yew King</i>	99
Teknologi Maklumat dan Telekerja: Satu Tinjauan Awalan dan Implikasinya di Malaysia <i>Balakrishnan Parasuraman</i>	115

# ALGORITHM OF MAGNETIC FLUX DENSITY ON A PLANE GENERATED BY A FINITE LENGTH CURRENT SOURCE

**\*Rashdi Shah Ahmad, ^Tahir Ahmad dan \*Chew Soon Leong**

\*Jabatan Fizik dan ^Jabatan Matematik

Fakulti Sains

Universiti Teknologi Malaysia

81310 Skudai

Johor Darul Takzim

## ABSTRACT

*The magnetic isofield pattern is very important in the field of magnetoencephalography. The pattern can assist in determining the location of the current sources. In this paper a program written in MATLAB environment, to generate the isofield pattern on a plane above a finite length current source, is described.*

***Keywords: Isofield pattern; Magnetic field; Magnetoencephalography***

## INTRODUCTION

The human brain is the most complex structure and the most important organ. There are at least  $10^{10}$  neurons (cells responsible for transmitting signal) in the outermost layer of the brain called the cerebral cortex. These neurons form a vast signal handling network of about  $10^{14}$  interconnections or synapses. When information is being processed, small currents (ions movement) flow in the system<sup>[1]</sup>. According to the Biot-Savart Law, these small currents will generate magnetic field that can reach the region outside the skull. A noninvasive technique for investigating neuronal activity in the living human brain by measuring this magnetic field is known as magnetoencephalography (MEG).

In the simplest model of the MEG, the neuronal activity center is modelled as a finite segment current source called a current dipole. The current dipole is characterized by six unknown parameters which are: (1) the x-, y- and z- coordinates to specify the location of the dipole (3 parameters); (2)  $\theta$ , the angle between the dipole and the z axis; (3)  $\phi$ , the angle between the projection of the dipole on the xy-plane and the x-axis; and (4) I, the current in the segment. To obtain these parameters, the inverse calculation of the Biot-Savart Law is required. Inverse calculation is different from the forward calculation in the sense that we are trying to find these six parameters from the measurement of the magnetic field instead of calculating the magnetic field at a particular point given these six parameters (forward solution). However, it was shown by Helmholtz that the solution to the inverse problem is not unique; i.e. there are more than one combinations of locations, orientations or current values that will generate the same magnetic field at a particular measuring point.

One way of solving the problem is by a method known as Bayesian (Tarantola, 1982). In brief, the method involves the forward calculation of all possible combination of parameters, and finally decides on the optimum solution based on the least squares method; *a priori* information is very useful to reduce the range of possibilities and thus the computing time.

*A priori* information can be approximately obtained from the magnetic field pattern. In this technique, the magnetic field flux density  $\vec{B}$  is measured simultaneously at many points outside the human head. Then the points having the same value of magnetic flux density are joined by a curve normally called the isofield curves. The pattern of the isofield curves will provide approximate information on the location of the dipole. In this paper, a MATLAB program is written to generate the isofield pattern on a plane situated above a finite current segment. Using this program, a user can key-in the parameters of the current segment and then visualize the isofield generated on the plane. By doing this, the user can learn the relationship between the pattern and the parameters of the dipole.

The magnitude of magnetic flux density,  $|\vec{B}|$  at the measurement point M (see Fig. 1) can be calculated by using the equation below (Sadiku, 1995):

$$|\vec{B}| = \frac{\mu_0 I}{4\pi a} (\cos \theta_1 - \cos \theta_2) \quad (1)$$

where:

$\mu_0$  = the permeability in free space, a constant having the value of  $4\pi \times 10^{-7} \text{ m T A}^{-1}$

$I$  = the magnitude of the current flow in the segment, in Ampere

$a$  = the perpendicular distance from the segment to the measurement point, in meter.

$\theta_1$  = the angle between a line parallel to the segment in the direction of the current flow and the line joining the starting point of the segment A to the measurement point M.

$\theta_2$  = the angle between the line parallel to the segment in the direction of the current flow and the line joining the end point of the segment B to the measurement point M.

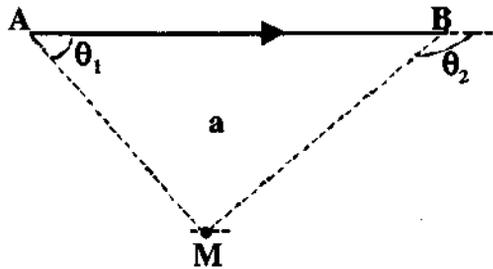


Figure 1: The geometry for magnetic field calculation at point M relative to a finite length current source, AB.

The direction of  $\vec{B}$  is obtained by taking the cross product of the two unit vectors: unit vector of  $\vec{AB}$  and the unit vector along a line joining the current segment to the measurement point. To obtain the direction physically, one only has to hold the segment such that his thumb points in the direction of the conventional current flow (from positive to negative terminal), and the direction of  $\vec{B}$  is given by the curl of the other fingers. In the case as shown in Fig. 1, the direction of the field at M is directed into the page.

However, in order to obtain the magnetic field pattern generated by a finite current source, a multi-point calculation is required. Since the procedure to calculate  $\vec{B}$  for a given current segment is the same for all the points involved, a program written in MATLAB environment is employed to implement the iterative calculation and to plot the isofield curve.

## PROBLEM FORMULATION

The problem we are trying to solve is that: Given a magnitude of current that flows

from point A to point B, we want to calculate the normal component of magnetic flux density  $\vec{B}$  at the measurement point M on the plane. To calculate the field at many points, three variables in equation (1) namely  $a$ ,  $\cos \theta_1$ , and  $\cos \theta_2$ , will be determined by vector analysis. The vectors involved in the calculation are shown in Fig. 2.

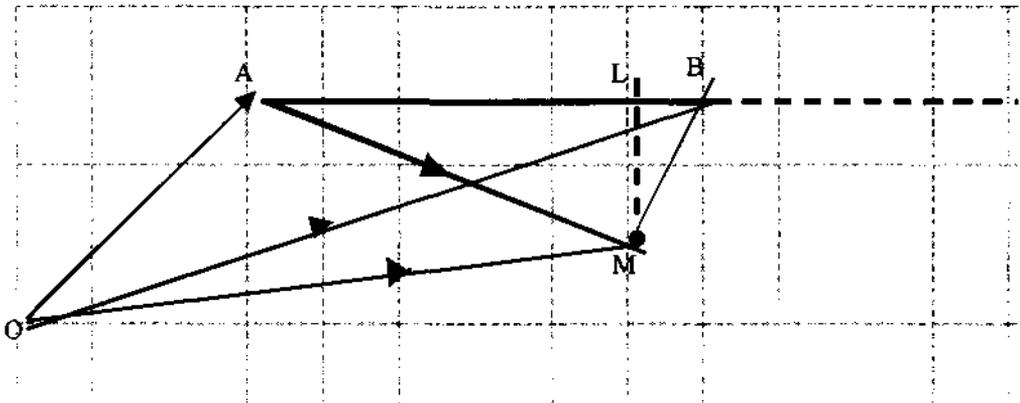


Figure 2: Vectors required in the magnetic field calculation.

The vector in the direction of current flow is given by  $\vec{AB}$ . The vector is calculated using the relation below:

$$\vec{OB} = \vec{OA} + \vec{AB} \quad (2)$$

$$\vec{AB} = \vec{OB} - \vec{OA} \quad (3)$$

A unit vector in the direction of the current flow is given by:

$$\hat{u}_{AB} = \frac{\vec{AB}}{|\vec{AB}|} \quad (4)$$

The vectors  $\vec{AM}$  and  $\vec{BM}$  are determined from:

$$\vec{AM} = \vec{OM} - \vec{OA} \quad (5)$$

$$\vec{BM} = \vec{OM} - \vec{OB} \quad (6)$$

The two corresponding unit vectors are given by:

$$\hat{u}_{AM} = \frac{\vec{AM}}{|\vec{AM}|} \quad (7)$$

$$\hat{u}_{BM} = \frac{\overline{BM}}{|BM|} \quad (8)$$

The cosines of the angles  $\theta_1$  and  $\theta_2$  are then calculated using the equations below:

$$\cos \theta_1 = \hat{u}_{AB} \cdot \hat{u}_{AM} \quad (9)$$

$$\cos \theta_2 = \hat{u}_{AB} \cdot \hat{u}_{BM} \quad (10)$$

The distance between the measurement point to the wire is given by the magnitude of  $\vec{LM}$  which is determined by the procedure below:

$$\vec{LM} = \vec{AM} - \vec{AL} \quad (11)$$

$\vec{AL}$  is the projection of  $\vec{AM}$  in the direction of  $\hat{u}_{AB}$  and it is given by:

$$\vec{AL} = (\overline{AM} \cdot \hat{u}_{AB}) \hat{u}_{AB} \quad (12)$$

$$\vec{LM} = \vec{AM} - (\overline{AM} \cdot \hat{u}_{AB}) \hat{u}_{AB} \quad (13)$$

Since  $\vec{AM}$  and  $\hat{u}_{AB}$  are calculated from  $O\vec{A}$ ,  $O\vec{B}$  and  $O\vec{M}$ , these vectors and the magnitude of the current are the only input required to calculate the field.

If the unit vector of  $\vec{LM}$  is denoted by  $\hat{u}_{LM}$ , the direction of the magnetic field at the measurement point is:

$$\hat{u}_B = \hat{u}_{AB} \times \hat{u}_{LM} \quad (14)$$

Then the magnetic flux density at point M is given by:

$$\vec{B} = \frac{10^{-7} I}{|\vec{LM}|} (\cos \theta_1 - \cos \theta_2) \hat{u}_B \quad (15)$$

To calculate the magnetic field on a plane above a finite length current source, the plane is treated as the xy-plane. The minimum and the maximum values of the two axes are then specified and the plane is subdivided into a two-dimensional array of x and y values. The size of the array will depend on the resolution required, i.e. the distance between two consecutive points along each axis,  $\Delta x$  and  $\Delta y$ . Each point on the array will represent the vector  $O\vec{M}$ . The calculation is then iterated for every point in the array. To visualize the isofield pattern, the contour of the z-component of the magnetic field is plotted. The z-component is chosen because it is the easiest to be measured by most of the magnetic sensors.

With a slight modification to the program, calculation of magnetic field on a plane above an infinite length current source can be implemented. An infinite length current source is just a special case of finite length current source where  $\cos \theta_1$  and  $\cos \theta_2$  can be set to 1 and -1 respectively. The procedure for calculating the values of  $\cos \theta_1$  and  $\cos \theta_2$  can be omitted. The position vectors  $\vec{OA}$  and  $\vec{OB}$  can be any vector that point to two points on the current line.

## ALGORITHM

Composition of equations (1) to (15) is the algorithm for the magnetic flux density on a plane generated by a finite length current source. This algorithm is coded in MATLAB and presented as follow:

```
%File Name :finite length current source
%Author :Dr. Rashdi Shah bin Ahmad
%Written on :04 September,2000
%This program will calculate the magnetic field density on a plane above a finite
%current source
%-----
%VARIABLE LIST:
%
%constant:
    %mu zero :the permeability in free space
%information on current :
    %current : the magnitude of the current
%related to the plane of mapping
    %z : the distance above the origin
    %xMin : the minimum value of the x-axis
    %xMax : the maximum value of the x-axis
    %stepX : the interval between two consecutive points on the x-axis.
    %yMin : the minimum value of the y-axis
    %yMax : the maximum value of the y-axis
    %stepY : the interval between two consecutive points on the y-axis.
%related to point array of the plane
    %row : number of row in the array
    %column : number of column in the array
    % s : the counter and the index for the row of the point array
    % t : the counter and the index for the column of the point array.
%
%vectors
    %vOA : position vector of A
```

```

%vOB : position vector of B
%vOM : position vector of M
%vAB : vector AB
%vAM : vector AM
%vBM : vector BM
%vLM : vector LM
%B : magnetic flux density field (vector quantity)
%magOfAB : magnitude of vAB
%magOfAM : magnitude of vAM
%magOfBM : magnitude of vBM
%magOfLM : magnitude of vLM
%magOfB : magnitude of B
%uAB : unit vector along vAB
%uAM : unit vector along vAM
%uBM : unit vector along vBM
%uLM : unit vector along vLM
%uB : unit vector along the direction of the field
%related to angle
%cos_theta1 : the cosine of the angle as define
%cos_theta2 : the cosine of the angle as define
%related to B
%Bx(s,t) : array to store the x-component of B at coordinate (s,t)
%By(s,t) : array to store the y-component of B at coordinate (s,t)
%Bz(s,t) : array to store the z-component of B at coordinate (s,t)
%-----
%START OF PROGRAM
%-----clear previous variables and the screen-----
clear
clc
%-----
%-----Initialize the start_time with the cpu time -----
%
start_time= cputime
% to be able to calculate the computing time
%-----%
%-----required constant-----
%
mu_zero=4*pi*1e-7
%-----
%-----
%-----INPUT Section-----%

```

```
%magnitude current in Ampere
current=1;
% current segment AB: consist of a wire along the y-axis
%10 cm below the XY plane ;starting at (0.5,0.45,-0.10) and end at (0.5,0.55,-0.1) meter
vOA=[0.5,0.45,-0.1];
vOB=[0.5,0.55,-0.1];
%
%Information of the plane
%height above the origin
z=0.1
%lower and upper boundary of x, in meter
xMin=0;xMax=1;
%lower and upper boundary of y, in meter
yMin=0;yMax=1;
%distance between two consecutive points in x-direction
stepX=0.1;
%distance between two consecutives point in y direction
stepY=0.1;

%-----
%-----CALCULATION-----%
%calculation of unit vector of AB
vAB=vOB-vOA;
magOfAB=sqrt(sum(vAB.*vAB));
% sum(vAB.*vAB) is the command use to do a dot product between two vectors
uAB=vAB/magOfAB;
%-----
%calculation number of the row and column
row=((yMax-yMin)/stepY)+1;
column=((xMax-xMin)/stepX)+1;
%-----
% specifying the value along x-axis
x=xMin:stepX:xMax;
%specifying the value along y-axis
y=yMin:stepY:yMax;
%-----
%generating the mesh: array of xy
[x,y]=meshgrid(x,y);
%
%-----starting the iteration-----
for s = 1:row
    for t = 1:column
```

```
vOM=[x(s,t), y(s,t),z];
% calculation of unit vector along AM
vAM=vOM-vOA;
magOfAM=sqrt(sum(vAM.*vAM));
uAM=vAM/magOfAM;
% calculation of unit vector along BM
vBM=vOM-vOB;
magOfBM=sqrt(sum(vBM.*vBM));
uBM=vBM/magOfBM;
%calculation of theta1 and theta2
cos_theta1=sum(uAM.*uAB);
cos_theta2=sum(uBM.*uAB);
%
%
%determine vLM
%vLM=vAM-vAL; vAL is the projection vAM on uAB
vLM=vAM-sum(vAM.*uAB)*uAB;
%calculate magnitude vLM
magOfLM=sqrt(sum(vLM.*vLM));
%calculating a unit vector along vLM
uLM=vLM/magOfLM;
%determine the direction of the field
uB=cross(uAB,uLM);
%calculate the magnitude of B at point M
magOfB=(mu_zero*current*(cos_theta1-cos_theta2))/(4*pi*magOfLM);
%
%store the value
B=magOfB*uB
Bx(s,t)=B(1,1);
By(s,t)=B(1,2);
Bz(s,t)=B(1,3);

end
end
Bz
%-----determine the computation time-----%
time=cputime-start_time
%-----
%-----plotting the isofield pattern-----
contour(x,y,Bz,20)
%-----END OF PROGRAM-----%
```

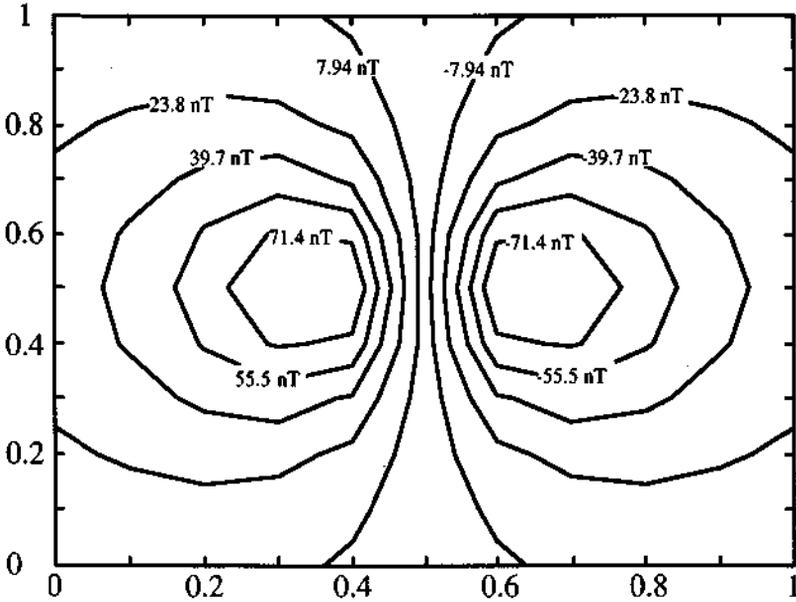


Figure 3: Isofield pattern produced by the program

## IMPLEMENTATION

The isofield pattern on an  $xy$ -plane bounded by  $0 \leq x \leq 1$  (meter) and  $0 \leq y \leq 1$  (meter) due to a 1 cm (0.01 m) current segment with an initial point at (0.5, 0.45, -0.10) and terminal point (0.5, 0.55, -0.1) is shown in Fig. 3. The pattern shown is not exactly similar to the one produced by the program. It has been edited for the purpose of publication. On the screen, each isofield curve is colored based on the value of the  $z$ -component of the magnetic flux density. In this case, the red curve represent the largest positive value of  $\hat{B}_z$ , and the blue curve represent the most negative value of  $\hat{B}_z$ .

To instruct MATLAB to label every isofield curve the following portion of the program should be modified as follows:

```
%-----plotting the isofield pattern-----  
[C,h]=contour(x,y,Bz,20)  
clabel(C,h)  
%-----END OF PROGRAM-----%
```

After running the modified program, MATLAB will produce the same isofield pattern but with the value of  $\hat{B}_z$  scattered all over the pattern. Fortunately, MATLAB allows the user to manually label individual curve by editing the following command:

clabel(C,h) to clabel(C,h,'manual')

With this command, the user can click on any isofield curve in Figure No.1 (the window generated to draw the contour curve) and the  $B_z$  -value of that particular curve is written on it. To stop the labeling process, the user only needs to click outside the plotting region.

To try with different input parameters, the user only needs to change the parameters at the input section of the program. As written, the program can complete the calculation in less than 1 second using a Pentium MMX233 MHz Intel machine with 64 Mbytes RAM. The computing time is basically controlled by the size of the array which is then determined by the step size  $\Delta x$  and  $\Delta y$ . For the step size of 0.1 m, as employed in this program, the isofield curve is not very smooth. Reducing the step size to 0.05 m will produce a smoother curve but with longer computing time of about 4.18 seconds.

## CONCLUSION

A program to calculate the magnetic field flux density on the two-dimensional array of measurement points have been written using MATLAB. Though successful, this program has two shortcomings. Firstly, it is limited to only one current segment whereas in a normal brain, it seems unlikely that only one region of the neurons will be activated at any particular time. Secondly, points on a plane do not represent the actual geometry of a human head. Solving these shortcomings will be our further work.

## REFERENCES

- Hamalainen, M et. al., 1993. Magnetoencephalography - Theory, Instrumentation and Applications to Noninvasive Studies of The Working Human Brain. *Rev. Modern Physics*, Vol. 65, No. 2, 413-497.
- Sadiku, M., 1995. *Element of Electromagnets*. Oxford University Press.
- Tarantola, A. and Valette, B. 1982. Inverse Problems-quest for information. *Journal of Geophysics.*, Vol. 50, 159-170.