

Chapter IV

Numerical Techniques

4.1 Forward and Inverse Fourier Transforms in Two Dimensions

We consider magnetohydrostatics in the case where α is a constant in the force-free magnetic field modeling. The techniques which we used to solve this problem involve two-dimensional Fourier transforms (Bracewell, 1986), so we use the fast Fourier transform as a numerical technique. The Fourier transform (FT) is a very useful and simple method for solving linear equations.

A two-dimensional function $f(x, y)$ has a two-dimensional transform $F(u, v)$, and the relations between the two are

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-i2\pi(ux+vy)} dx dy, \quad (4.1)$$

which is called the (forward) Fourier transform and

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) e^{i2\pi(ux+vy)} du dv, \quad (4.2)$$

the inverse Fourier transform.

These equations explain the analysis of the two-dimensional function $f(x, y)$ in the form of waves $e^{i2\pi(ux+vy)}$. In our thesis work, we use both forward and inverse Fourier transforms and use functions $f(x, y)$ and $F(u, v)$ as $B(x, y)$ and $\hat{B}(u, v)$, which represent the magnetic field in x - y space and u - v

space, respectively. First, we use the forward Fourier transform to transform the boundary condition $B_1(x, y, 0)$ to $\hat{B}_1(u, v, 0)$. Then we solve the force-free field equation by our generalized equations, as described in Chapter II. Then we will have the solution in three dimensions, and finally we use the inverse Fourier transforms to convert \hat{B} back to B in rectangular spatial coordinates. For the numerical technique we approximated these integrals discretely and used the fast Fourier transform method, which is a standard, efficient method for this problem.

4.2 Discrete Fourier Transform in Two Dimensions

In the previous section we showed the Fourier integral, but in our work the input data (from a magnetogram) are represented by a finite set of discrete values. Thus eq. (4.1) can be approximated by a discrete Fourier transform of a finite number of data points $N_1 \times N_2$ (Press et al, 1992):

$$F(u_{n_1}, v_{n_2}) \approx \sum_{k=0}^{N_2-1} \sum_{j=0}^{N_1-1} f(x_j, y_k) e^{-2\pi i(u_{n_1} x_j + v_{n_2} y_k)} \Delta_1 \Delta_2, \quad (4.3)$$

and similarly, the inverse Fourier transform becomes

$$f(x_j, y_k) \approx \frac{1}{N_1 N_2} \sum_{n_2=0}^{N_2-1} \sum_{n_1=0}^{N_1-1} F(u_{n_1}, v_{n_2}) e^{2\pi i(u_{n_1} x_j + v_{n_2} y_k)} \Delta_1 \Delta_2, \quad (4.4)$$

where

$$x_j \equiv j \Delta_1, \quad j = 0, 1, 2, \dots, N_1 - 1$$

$$y_k \equiv k \Delta_2, \quad k = 0, 1, 2, \dots, N_2 - 1$$

$$u_{n_1} \equiv \frac{n_1}{N_1 \Delta_1}, \quad n_1 = 0, \dots, N_1 - 1$$

$$v_{n_2} \equiv \frac{n_2}{N_2 \Delta_2}, \quad n_2 = 0, \dots, N_2 - 1$$

and

$$F(u_{n_1}, v_{n_2}) \approx F_{n_1, n_2} \Delta_1 \Delta_2,$$

so we can rewrite eq. (4.3) as

$$F_{n_1, n_2} = \sum_{k=0}^{N_2-1} \sum_{j=0}^{N_1-1} f(x_j, y_k) e^{-2\pi i(n_1 j/N_1 + n_2 k/N_2)} \quad (4.5)$$

and eq. (4.4) as.

$$f(x_j, y_k) \approx \frac{1}{N_1 N_2} \sum_{n_2=0}^{N_2-1} \sum_{n_1=0}^{N_1-1} F(u_{n_1}, v_{n_2}) e^{2\pi i(n_1 j/N_1 + n_2 k/N_2)} \quad (4.6)$$

For our work we use $\Delta_1 = 1/\cos\varphi$ and $\Delta_2 = 1$, where φ is the angle between the line of sight and the vertical direction, so Δ_2 depends on the line of sight angle. The input data have 1 pixel per arc second, so we are using length units of 1" or about 3288 km at the solar surface. We assume that F and f are periodic with period $N_1\Delta_1$, $N_2\Delta_2$, $1/\Delta_1$, and $1/\Delta_2$ in x , y , u and v , respectively. Thus a problem which occurs when we use the discrete FT is called aliasing, which means that the periodicity can affect our results (a numerical artifact). We resolved this problem by expanding the boundaries to a width of twice the width of the region of interest, and setting the data outside the region of interest to zero to essentially eliminate the effect of the periodicity assumption in our program.

4.3 Magnetic Field Line Visualization

When we obtained the magnetic field in three dimensions from the Fourier transform, next we created a magnetic field line which is defined as a curve which is tangent at any point to the direction of the field. There is no limit to the number of lines which may be drawn in a given flow field, since they are lines and have no thickness. The differential equation determining these lines is

$$d\mathbf{l} \times \mathbf{B} = 0, \quad (4.7)$$

where $d\mathbf{l}$ is a small displacement vector along the magnetic field with components $(dx, dy$ and $dz)$ and \mathbf{B} is the magnetic field vector. Then

$$(dx \hat{i} + dy \hat{j} + dz \hat{k}) \times (B_x \hat{i} + B_y \hat{j} + B_z \hat{k}) = 0, \quad (4.8)$$

$$\begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ dx & dy & dz \\ B_x & B_y & B_z \end{vmatrix} = 0$$

$$(B_x dy - B_y dz)\hat{i} + (B_x dz - B_z dx)\hat{j} + (B_y dx - B_z dy)\hat{k} = 0.$$

The solution of this equation is

$$\frac{dx}{B_x} = \frac{dy}{B_y} = \frac{dz}{B_z}, \quad (4.9)$$

where $B_x, B_y,$ and B_z are the x, y and z components of the magnetic field.

For our thesis, we set

$$\frac{ds}{|\mathbf{B}|} = \frac{dx}{B_x} = \frac{dy}{B_y} = \frac{dz}{B_z}, \quad (4.10)$$

where $ds = \sqrt{dx^2 + dy^2 + dz^2}$ and $|\mathbf{B}| = \sqrt{B_x^2 + B_y^2 + B_z^2}$. This equation will give us the magnetic field line, starting from each initial point on the solar surface (photosphere). For this work, we choose $ds = 0.01$ arc second as the step size near the spot and use 0.1 arc second as the step far from the boundary.

4.4 Selection of Initial Points of Visualized Magnetic Field Lines

The initial points of visualized magnetic field lines are chosen so that the density of the points is proportional to the magnitude of the magnetic field. To accomplish this, we have developed a novel technique. We start with an initial distribution of n points in a plane, and move the points in order to minimize the functional

$$V = \sum_{i=1}^n (\Phi_i - \Phi_0)^2, \quad (4.11)$$

where Φ_i is the estimated magnitude of the flux of B_y through a triangle including point i and its two nearest neighbors, and Φ_0 is the total flux divided by n . The goal is to achieve a spacing of points such that each triangle contains a flux close to Φ_0 .

4.5 Interpolation and Image Drawing Method

4.5.1 Linear interpolation

Linear interpolation is the simplest way to find an unknown value between two known data points in one dimension (Figure 4.1).

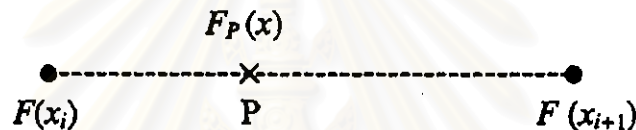


Figure 4.1: Linear interpolation

We have data at two points, $F(x_i)$ and $F(x_{i+1})$, and we can estimate F at the point P by using the following formula:

$$\begin{aligned} F_P(x) &= \frac{x_{i+1} - x}{x_{i+1} - x_i} F(x_i) + \frac{x - x_i}{x_{i+1} - x_i} F(x_{i+1}) \\ &= (1 - f_x) F(x_i) + f_x F(x_{i+1}) \end{aligned}$$

where f_x is the fractional distance of P from x_i to x_{i+1} ,

$$f_x = \frac{x - x_i}{x_{i+1} - x_i},$$

and f_x should be between 0 and 1 for interpolation and < 0 or > 1 for extrapolation.

4.5.2 Bilinear interpolation

For our work, we use bilinear interpolation when we rotate the magnetic field on the boundary to the center of disk, and also to find the magnetic field at the points we want. Bilinear interpolation is shown in Figure 4.2.

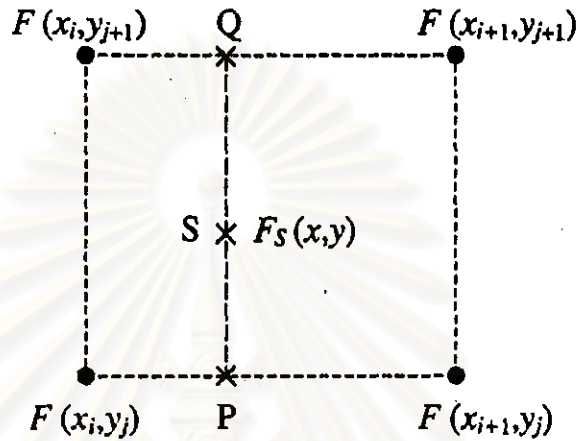


Figure 4.2: Bilinear interpolation.

The figure shows the known data at four points, $F(x_i, y_j)$, $F(x_{i+1}, y_j)$, $F(x_i, y_{j+1})$, and $F(x_{i+1}, y_{j+1})$. The unknown function value at point S can be estimated by projection of the points to points P and Q and using linear interpolation to find F_P and F_Q .

At point P:

$$\begin{aligned} F_P(x, y_j) &= \frac{x_{i+1} - x}{x_{i+1} - x_i} F(x_i, y_j) + \frac{x - x_i}{x_{i+1} - x_i} F(x_{i+1}, y_j) \\ &= (1 - f_x) F(x_i, y_j) + f_x F(x_{i+1}, y_j). \end{aligned}$$

At point Q:

$$\begin{aligned} F_Q(x, y_{j+1}) &= \frac{x_{i+1} - x}{x_{i+1} - x_i} F(x_i, y_{j+1}) + \frac{x - x_i}{x_{i+1} - x_i} F(x_{i+1}, y_{j+1}) \\ &= (1 - f_x) F(x_i, y_{j+1}) + f_x F(x_{i+1}, y_{j+1}). \end{aligned}$$

At point S:

$$\begin{aligned} F_S(x, y) &= \frac{y_{j+1} - y}{y_{j+1} - y_j} F_P(x, y_j) + \frac{y - y_j}{y_{j+1} - y_j} F_Q(x, y_{j+1}) \\ &= (1 - f_y) F_P(x, y_j) + f_y F_Q(x, y_{j+1}), \end{aligned}$$

where

$$f_y = \frac{y - y_j}{y_{j+1} - y_j}.$$

Thus the function at point S is given by

$$\begin{aligned} F_s(x, y) &= (1 - f_y)(1 - f_x)F(x_i, y_j) \\ &\quad + (1 - f_y)f_x F(x_{i+1}, y_j) \\ &\quad + f_y(1 - f_x)F(x_i, y_{j+1}) \\ &\quad + f_y f_x F(x_{i+1}, y_{j+1}). \end{aligned}$$

4.5.3 Linear interpolation in three dimensions

Linear interpolation in 3D was used while we were creating the magnetic field line in 3D to find the direction of the vector field which helps us to visualize the magnetic loops from our simulations. Linear interpolation in 3D is shown in Figure 4.3.

To estimate the unknown function value at point S is analogous to finding the unknown function value in linear and bilinear interpolation:

At point A:

$$\begin{aligned} F_A(x, y_j, z_k) &= \frac{x_{i+1} - x}{x_{i+1} - x_i} F(x_i, y_j, z_k) + \frac{x - x_i}{x_{i+1} - x_i} F(x_{i+1}, y_j, z_k) \\ &= (1 - f_x) F(x_i, y_j, z_k) + f_x F(x_{i+1}, y_j, z_k). \end{aligned}$$

At point B:

$$\begin{aligned} F_B(x, y_j, z_{k+1}) &= \frac{x_{i+1} - x}{x_{i+1} - x_i} F(x_i, y_j, z_{k+1}) + \frac{x - x_i}{x_{i+1} - x_i} F(x_{i+1}, y_j, z_{k+1}) \\ &= (1 - f_x) F(x_i, y_j, z_{k+1}) + f_x F(x_{i+1}, y_j, z_{k+1}). \end{aligned}$$

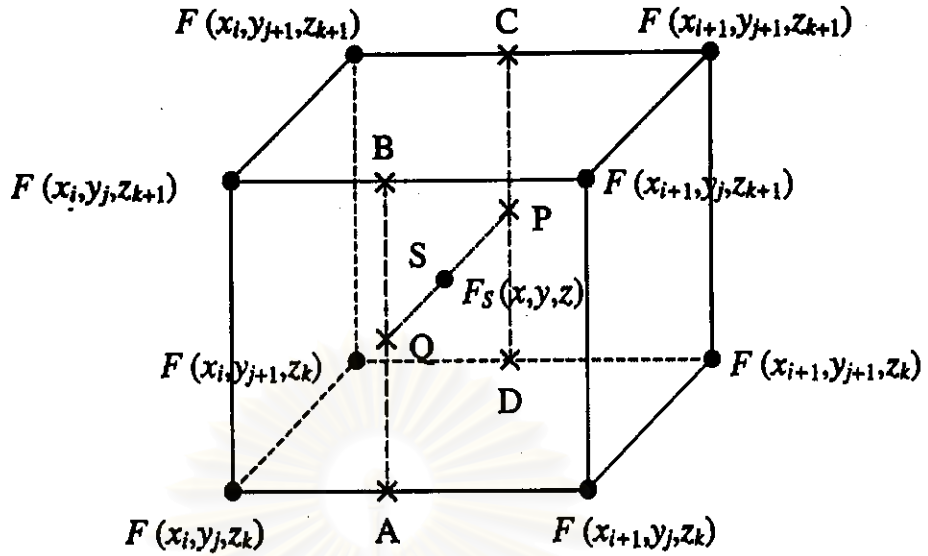


Figure 4.3: Linear interpolation in three dimensions.

At point Q:

$$\begin{aligned}
 F_Q(x, y_j, z) &= \frac{z_{k+1} - z}{z_{k+1} - z_k} F_A(x, y_j, z_k) + \frac{z - z_k}{z_{k+1} - z_k} F_B(x, y_j, z_{k+1}) \\
 &= (1 - f_x) F_A(x, y_j, z_k) + f_x F_B(x, y_j, z_{k+1}) \\
 &= (1 - f_x)(1 - f_x) F(x_i, y_j, z_k) \\
 &\quad + f_x(1 - f_x) F(x_{i+1}, y_j, z_k) \\
 &\quad + (1 - f_x) f_x F(x_i, y_j, z_{k+1}) \\
 &\quad + f_x f_x F(x_{i+1}, y_j, z_{k+1}),
 \end{aligned}$$

where

$$f_x = \frac{z - z_k}{z_{k+1} - z_k}.$$

For points D, C and P, we calculate in the same way as at points A, B and Q, respectively, so we get:

At point D:

$$F_D(x, y_{j+1}, z_k) = (1 - f_x) F(x_j, y_{j+1}, z_k) + f_x F(x_{i+1}, y_{j+1}, z_k).$$

At point C:

$$F_C(x, y_{j+1}, z_{k+1}) = (1 - f_x)F(x_i, y_{j+1}, z_{k+1}) + f_x F(x_{i+1}, y_{j+1}, z_{k+1}).$$

At point P:

$$\begin{aligned} F_P(x, y_{j+1}, z) &= (1 - f_x)F_D(x, y_{j+1}, z_k) + f_x F_C(x, y_{j+1}, z_{k+1}) \\ &= (1 - f_x)(1 - f_x)F(x_i, y_{j+1}, z_k) \\ &\quad + f_x(1 - f_x)F(x_{i+1}, y_{j+1}, z_k) \\ &\quad + (1 - f_x)f_x F(x_i, y_{j+1}, z_{k+1}) \\ &\quad + f_x f_x F(x_{i+1}, y_{j+1}, z_{k+1}). \end{aligned}$$

Thus the function value at point S is

$$\begin{aligned} F_S(x, y, z) &= \frac{y_{j+1} - y}{y_{j+1} - y_j} F_Q(x, y_j, z) + \frac{y - y_j}{y_{j+1} - y_j} F_P(x, y_{j+1}, z) \\ &= (1 - f_y)F_Q(x, y_j, z) + f_y F_P(x, y_{j+1}, z) \\ &= (1 - f_x)(1 - f_y)(1 - f_x)F(x_i, y_j, z_k) \\ &\quad + f_x(1 - f_y)(1 - f_x)F(x_{i+1}, y_j, z_k) \\ &\quad + (1 - f_x)(1 - f_y)f_x F(x_i, y_j, z_{k+1}) \\ &\quad + f_x(1 - f_y)f_x F(x_{i+1}, y_j, z_{k+1}) \\ &\quad + (1 - f_x)f_y(1 - f_x)F(x_i, y_{j+1}, z_k) \\ &\quad + f_x f_y(1 - f_x)F(x_{i+1}, y_{j+1}, z_k) \\ &\quad + (1 - f_x)f_y f_x F(x_i, y_{j+1}, z_{k+1}) \\ &\quad + f_x f_y f_x F(x_{i+1}, y_{j+1}, z_{k+1}). \end{aligned}$$

4.5.4 Image drawing method

To input magnetogram data, we had to figure out the format of FITS image files, and how to read the data into our program. In the program before

data are processed by the forward Fourier transform routine, we rotated the data counter-clockwise by $\beta = 26^\circ$ so that the region of interest was in the equatorial plane of the solar disk to make it easier for us to calculate the magnetic field in three dimensions. We selected only a small area, including the active region, for our simulation. In this case, for the original magnetogram image, we let the y_1 -axis be along the vertical, the x_1 -axis in the horizontal direction, and the z_1 -axis in the direction toward the observer, as shown in Figure 4.4; we call this coordinate system 1. (Note that magnetogram data are always taken with the solar North pole oriented along \hat{y}_1 , in projection.)

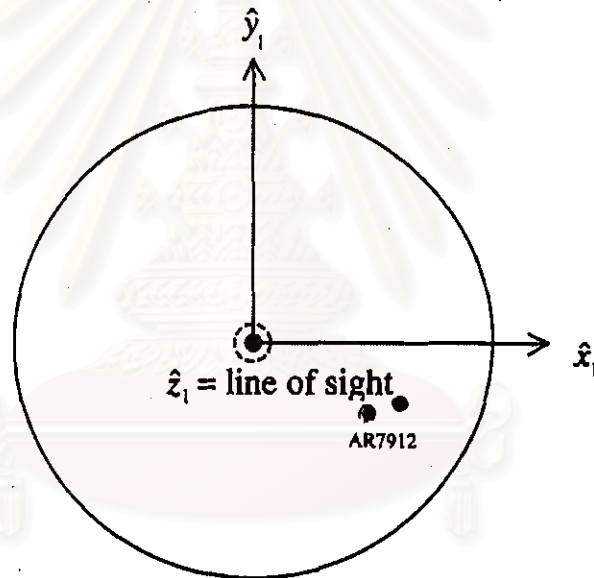


Figure 4.4: Coordinate system 1.

Therefore, the equations for the first rotation are

$$\left. \begin{aligned} x_2 &= x_1 \cos \beta - y_1 \sin \beta, \\ y_2 &= y_1 \cos \beta + x_1 \sin \beta, \\ z_2 &= z_1, \end{aligned} \right\} \quad (4.12)$$

and now, in terms of (x_2, y_2, z_2) , coordinate system 2, AR7912 lies near $y_2 = 0$ and $x_2 > 0$. At this point we transform to coordinate system 3, where x_3 and y_3 are individually, linearly rescaled so that $x_3 = y_3 = 0$ at the lower left corner of the simulation region, \vec{x}_3 and \vec{y}_3 are axes projected onto the solar surface, and z_3 is perpendicular to the solar surface. After this, we perform other numerical processes: FT, solution of eqs. (2.41) - (2.43), inverse FT's, minimization, and field line tracing. When we have coordinates (x_3, y_3, z_3) for the field lines, we transform back to coordinates as in Figure 4.4 (coordinate system 4) with x_4 and y_4 in the plane of the solar disk by using these equations:

$$\left. \begin{aligned} x_4 &= R \sin \psi_0 + x_3 \cos \psi_0 + z_3 \sin \psi_0, \\ y_4 &= y_3, \\ z_4 &= R \cos \psi_0 - x_3 \sin \psi_0 + z_3 \cos \psi_0. \end{aligned} \right\} \quad (4.13)$$

where ψ_0 is the angle from the disk center to the active region in equatorial plane (about 30°), and R is the solar radius (about 697,000 km).

Next, we reversed the first rotation in order return AR7912 back to the same position as in the initial input data by the inverse of eq. (4.12):

$$\left. \begin{aligned} x_5 &= x_4 \cos \beta + y_4 \sin \beta, \\ y_5 &= y_4 \cos \beta - x_4 \sin \beta, \\ z_5 &= z_4. \end{aligned} \right\} \quad (4.14)$$

The angle between the pole of the Sun and the y -axis is called the angle of inclination, γ (about 6°). This can also be thought of as the Earth's latitude with respect to the solar equator. We therefore transformed into coordinate system 6 with y_6 along the solar North pole (axis of rotation) using

$$\left. \begin{aligned} x_6 &= x_5, \\ y_6 &= y_5 \cos \gamma + z_5 \sin \gamma, \\ z_6 &= -y_5 \sin \gamma + z_5 \cos \gamma. \end{aligned} \right\} \quad (4.15)$$

Then we transformed the data to the date and time of interest by taking into account the angular velocity of the Sun, ω ; or more precisely, the angular velocity at the latitude of AR7912, which could be found from the magnetogram pictures shown in section 3.2. We measured the change in position of the sunspots between the times of the pictures. In this work we chose the magnetic data on 15th and 18th October, 1995 to calculate the angular velocity, and found that the synodic angular velocity of the Sun at this latitude was about 1.39×10^{-4} degrees per second. Then we got the change in the azimuthal angle by multiplying the angular velocity and time difference as

$$\Omega = \omega \Delta t, \quad (4.16)$$

where Δt is the time difference between the starting date and the date of interest.

The rotation equations are

$$\left. \begin{aligned} x_7 &= x_6 \cos \Omega + z_6 \sin \Omega, \\ y_7 &= y_6, \\ z_7 &= -x_6 \sin \Omega + z_6 \cos \Omega. \end{aligned} \right\} \quad (4.17)$$

Finally, we transformed the North pole to the same position as in images taken from Earth by

$$\left. \begin{aligned} x_8 &= x_7, \\ y_8 &= y_7 \cos \gamma - z_7 \sin \gamma, \\ z_8 &= y_7 \sin \gamma + z_7 \cos \gamma. \end{aligned} \right\} \quad (4.18)$$

These were all the steps to trace the simulation images; after that we used these data to plot with a commercial plotting program such as gnuplot.

4.6 Summary of Simulation Procedure

For our simulations, we can separate our program into steps as shown in Figure 4.5. First, to read source data into our program, the program will read

the data at the region we are interested in into a variable $N_1 \times N_2$ array where $N_1 = 2^n$ and $N_2 = 2^m$ for integral n and m . We had the program automatically read in data from only the region of interest for October 15th, 18th, 20th and 21st, 1995. For other days we will input it by ourselves. Then the complete analysis procedure is as described in earlier subsections. A flow chart is shown in Figure 4.5.

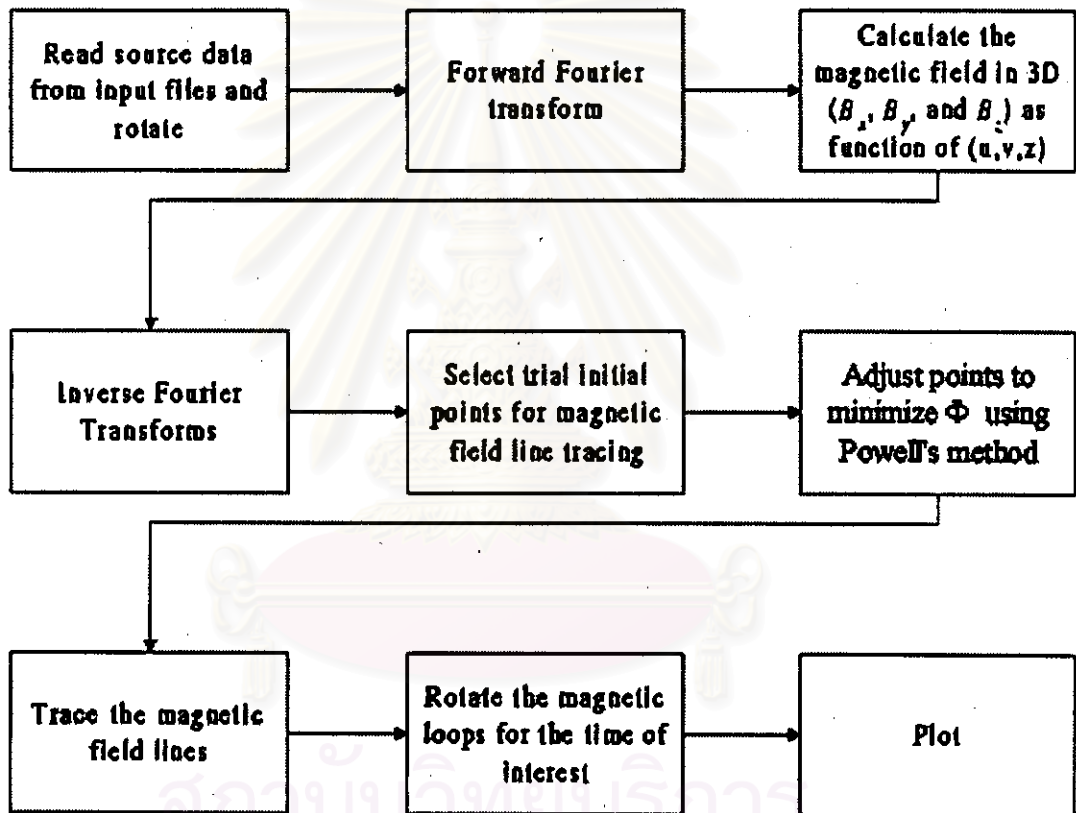


Figure 4.5: Overall flow chart of the simulation procedure.