Visualization of the Plasma Shape in a Force Free Helical Reactor, FFHR

Kunqi Hu¹, Qixiong Wang², Koji Koyamada², Hiroaki Ohtani³, Takuya Goto³, Junichi Miyazawa³

¹Graduate School of Engineering, Kyoto University, Japan
²Academic Center for Computing and Media Studies, Kyoto University, Japan
³National Institute for Fusion Science, Japan
⁴The Graduate University for Advanced Studies, SOKENDAI, Japan

*hu.kunqi.24w@st.kyoto-u.ac.jp

Abstract. Using magnetic field to confine the plasma can realize controlled nuclear fusion. A device called Force Free Helical Reactor (FFHR) using helical coils to generate a magnetic trap is being designed by National Institute for Fusion Science (NIFS) in Japan. Because the FFHR has the plasma with a complicated three-dimensional (3D) structure, 3D design of the in-vessel components is necessary. In the past, the design of the in-vessel components has been conducted based on the discrete two-dimensional poloidal cross-sectional shapes. However, this method is not effective and complete check of the interference is difficult. Since plasma movement follows the magnetic field lines, the plasma shape could be generated from them. Marching cubes method is applied for generating the polygon mesh. A graphic software is used for modifying and rendering.

Keywords: Visualization, Magnetic field lines, Plasma shape, Marching Cubes

1. Introduction

1.1. Magnetic confinement fusion

Nuclear fusion is an ideal means to obtain clean energy through the combination of two or more atomic nuclei. Compared with nuclear fission, the source is available from the sea and the reaction process is much safer. There are various ways to achieve controlled nuclear fusion, among which using magnetic fields is the most possible method to confine the high-density and high-temperature plasma.
The fusion reaction requires an extremely high temperature to break the attraction between the electrons and the nuclei and to overcome the repulsion among the nuclei. By influence of this high-temperature circumstance, the distance between each particle becomes too far to generate new nuclei. The plasma temperature is assumed to be 11keV in the center region and 2keV in the periphery region in the next-generation fusion reactor. Magnetic field can confine the high-temperature plasma without contacting the vessel of the reactor, and maintain the continuous fusion reaction in the reactor.

There are two basic ways to generate twisted magnetic fields confining the plasma at the center of the vessel. One is the combination of two kinds of magnetic fields which are respectively generated from toroidal field coils and poloidal field due to plasma current, such as Tokamak [1,2]. The other way is to directly use helical coils which have a complicated structure, such as Stellarator [3].

This paper focuses on the second type. The device called Force Free Helical Reactor (FFHR) designed by National Institute for Fusion Science (NIFS) has helical coils to generate twisted magnetic field lines [4]. One of the main issues is that the space between the plasma shape and the components along the vessel is limited. The basic method, such as observing the profile of the device and the plasma boundary in two-dimension, lacks an efficient approach to achieve a global view. Accordingly, a three-dimensional (3D) model of the plasma shape should be built to obtain a deeper perception.

1.2. Structure of FFHR

The profiles of this device and the magnetic field lines can be seen in Fig. 1 [5]. In the limited space between the coil and the plasma, blankets modules will be installed in the immediate vicinity of the plasma in order to achieve the most effective utilization of the neutron energy generated by the fusion reaction. The gap between the two blankets is arranged for the magnetic field lines to go through to the divertor for the online removal of impurities from the plasma, while the reactor is still operating. The yellow components are the superconducting coils.

The magnetic field lines profile consists of three parts: Flux surface in an ellipse with pink color, stochastic field lines surrounding the flux surface in dark blue, and the edge surface
layers with divertor legs that extend to the divertor plates [6]. It can be noticed that the space between the inboard blanket and the edge surface layers is extremely narrow. In addition, the paths which extend to the divertor for the impurities also need to be accurately designed. Previously, a shape of the component in the reactor was designed considering a structure of the magnetic field line in two-dimensional cross section at the discrete toroidal angle, that is, the Poincare plots of the magnetic field line as shown in Fig.1. However, the magnetic field line has a 3D structure and the plasma particle has a spread region of the Larmor radius around the magnetic field. Therefore, it is necessary to work three-dimensionally with the plasma for optimization of a 3D shape design of the component in the reactor. The final goal of this research is to construct a 3D model which expresses a plasma existence region by a closed curved surface without any arbitrariness from the magnetic field line data, and to enable us to check interference between the plasma and the 3D design data of the components in the reactor.

In the next section, some related work will be introduced. In section 3, the tracing method of the magnetic field lines and the recorded data format will be provided. Section 4 focuses on the approach for generating a polygon mesh. Section 5 is the visualization result. The evaluation will be discussed in section 6 and the summary in section 7.

2. Related Works

The acquisition of the plasma shape has abundant and thorough research with profiles. Itagaki, Masafumi et al. [7,8] found an effective way to reconstruct the 3D magnetic field profile outside the non-axisymmetric plasma in a reactor and indicated the plasma boundary precisely. However, they did not build a 3D model but visualized the result with profiles of the magnetic field. Peikert, R., & Sadlo, F. [9,10] presented an algorithm for finding vortex rings in vector fields and visualizing them by means of Poincare plots. For now, NIFS also uses Poincare plots to analyze the plasma shape. They have made slices from the magnetic field lines every 0.5 degrees along the toroidal orientation. Since the reactor has repeated structure, the dynamic feature of plasma, also known as the distribution of magnetic field lines on Poincare plots of a certain degree within every 36 degrees is combined into a Poincare plot. Then the plots are put into the profile of the reactor, which is similar to Fig. 1, for the interference check. However, this is still time-consuming and lacks representation of Larmor radius.

Rarely has someone drawn an outline of the plasma shape directly using 3D data. Instead, there is research regarding the use of streamlines to visualize the magnetic field lines in a reactor. Chiariello, Andrea G., et al. [11] provided an effective and fast approach to compute the magnetic field lines. Watanabe, T., et al. [12] used different colors to classify the magnetic field lines of force depending on the connection length. However, the results of those studies were made up of thousands of streamlines which lack recognizability. NIFS developed a virtual-reality (VR) visualization system in which the space relation between plasma and components in the vessel of the Large Helical Device in NIFS could be analyzed by using CAVE system. Results were shown using Virtual LHD software, which could show magnetic field line, particle trajectory, and isosurface of plasma pressure of the device. [13]

Tanaka K. et al. [14] proposed a particle-based rendering method by using 3D points directly as rendering primitives, which can generate quick and high-quality visualization of large-scale particle simulations and easily fuse surrounding objects, underlying the simulated fluid or multiple physical values. In our research, the magnetic field lines are also described as sets.
of points. However we propose using Marching cubes method to describe the feature of those points and rendering with polygon mesh. The reason for using Marching cubes method is described in subsection 3.2.

Marching cubes algorithm is a sequential-traversal method commonly used in medical imaging technology for generating isosurface described by a scalar field. The algorithm was first carried out by Lorensen and Cline in 1987 [15]. Since then, many researchers have focused on extending the basic approach, resolving its ambiguities, and improving its performance. Here, we have combined Marching cubes method with our approach and firstly applied it for obtaining plasma shape.

3. Data Preparation

3.1. Tracing method

The data used for modeling is directly provided by NIFS. In fact, adjustment, observation and data recording of a fusion reactor are highly sophisticated. Since this paper focuses on the visualization of the plasma shape, we will provide an overview of the methods for obtaining data from FFHR [16].

The 3D Variational Moments Equilibrium Code (VMEC) [17,18] is applied to FFHR computing the magnetohydrodynamics (MHD) equilibrium problem. Due to the relatively high speed of the VMEC, the code has become the standard code for calculating 3D plasma equilibria [19]. The code provides the 3D magnetic field distribution inside the vessel. Subsequently, the Magnetic Field Tracing Code (MGTRC) [20] is utilized for tracing the magnetic field lines. Magnetic fields at any point are interpolated using a three-dimensional 4th order spline function. Equations of field lines are integrated with the use of an 8-stage 6th order Runge-Kutta formulation. Each magnetic field line can then be identified following this process [21].

In this calculation of the magnetic field lines, the starting points are set around the X point, at which the poloidal field has zero magnitude, because the field lines that form these starting points move through the separatrix region and/or constitute the divertor legs. Once the point on the field line is out of the range of the vacuum vessel area, the tracing of this magnetic field line will be stopped and the set of points will be recorded. Then, the next field line is calculated from the next starting point. The sequence is executed repeatedly until the number of the magnetic field lines is sufficient enough to represent the plasma shape on the Poincare plots.

3.2. Data format

There are in total 1,598 magnetic field lines stored at this time. Each line was given an index. All of these lines consist of almost $6.9 \times 10^6$ points. A point belonging to a magnetic field line is stored in the following format as Table 1:
Table 1: Data sheet of a point

| Parameter                                      | Value (unit) | Data type |
|------------------------------------------------|--------------|-----------|
| Magnetic field line index                      | $n_m$        | Integer   |
| Toroidal angle index                           | $n_T$        |           |
| Radial coordinate                              | $R$ (m)      |           |
| Toroidal angle                                 | $\phi_T$ (degrees) |       |
| X coordinate                                   | $X$ (m)      |           |
| Y coordinate                                   | $Y$ (m)      |           |
| Z coordinate                                   | $Z$ (m)      |           |
| Magnetic field line connection length          | $L_m$ (m)    | double float |
| Radial component of magnetic field             | $B_r$ (T)    |           |
| Axial component of magnetic field              | $B_A$ (T)    |           |
| Azimuthal component of magnetic field          | $B_t$ (T)    |           |
| Strength of magnetic field                     | $B$ (T)      |           |

From the first line of content is the magnetic field line index that indicates to which magnetic field line this point belongs. The second parameter is the toroidal angle index. When tracing one magnetic field line, except the magnetic field line index, the information of a point will be recorded for each 0.5 degrees. As mentioned above, the NIFS method is checking the Poincare plots. Therefore, the toroidal angle index will be helpful when extracting the points at the same degree but from different magnetic field lines. And the toroidal angle exactly shows the corresponding angle with this index.

The third parameter is the distance between the projection of a point onto the $X$-$Y$ plane and the origin of coordinates. This third parameter is called as a radial distance, and the fifth and sixth parameters are the $X$ and $Y$ coordinates, respectively. With the $Z$ coordinate, the position of a point in 3D space is decided. The eighth parameter indicates the length of a magnetic field line. To compress the data, we selected those magnetic field lines which have length of more than one-half a circle but less than two circles at toroidal orientation from the data. For the reason that shorter lines are supposed to have no relationship with plasma movement, they are usually generated between a coil and another coil at the stochastic region. On the other hand, longer lines cannot provide prominent features since their points are close to each other at the Poincare plots. The last four are three components and the strength of the magnetic field at this point. These three components belong to an independent simple torus coordinate. This information will be used to calculate the Larmor radius at this point. Figure 2 shows some magnetic field lines with different indices. Figure 3 shows all the magnetic field lines after being selected.

Figure 3 explains why it is unsuitable to simply use the streamline visualization method. The critical point is to detect the interference with the components in the reactor. This means the lines located at the outer layer are likely required to be valued. However, as shown in the
figure, different magnetic field lines go across one another due to the twisted magnetic field. It is difficult to recognize the outer lines from those relatively close to the center. This causes a situation in which many more magnetic field lines are rendered than were necessary. The situation not only causes occlusion but also places a burden on the computer. Further, the representing of the Larmor radius will become another troublesome problem. Since the Larmor radius is a radius of the charged particle gyromotion centered at the magnetic field line on the plane whose normal vector is parallel to the magnetic field line, the Larmor radius, that is, the gyromotion, will be represented as a tube or a ring around the field streamline. In this case, the magnetic field line is hidden by the tube, and then the visualization of the field line will definitely cost much more computing resources. That is why we attempt to use the stream surface for representing the plasma shape.

In the next section, we will discuss our approach to building a 3D model of the plasma shape from the magnetic field lines.

Figure 2: Magnetic field lines with indexes of 40, 204, 297, and 365 corresponding to (a), (b), (c), and (d), respectively. The lengths of both (a) and (b) are too short while (c) is too long. (d) is the appropriate length which is selected.
4. Approach

4.1. Volume data

One of the vital points regarding applying Marching cubes is the volume data. Commonly, the volume data to be visualized is scalar volumetric data. It can be defined as a set of 3D points $P_i = (x_i, y_i, z_i)$, each of which has its own scalar value or property [22]. In this paper, $P_i$ is described as the 3D coordinate corresponding to the center of each cube. In this paper, the scalar value is described as the number of the magnetic field lines which go across each cube. In this way, for those adjacent cubes, if there are magnetic field lines that go across them, they will coalesce into a continuous surface at last.

4.2. Cube index

The number of essential cubes has been settled according to the distribution of points from all the magnetic field lines. Unless otherwise specified, all space coordinates appearing in this paper are in meters. Along the $X$ axis, the extrema of the positive and the negative part are approximately 22. The $Y$ axis has the same extrema. In terms of the $Z$ axis, the value shows strictly less than 6. Simultaneously, one has been selected as basic size. The reason is that it will be convenient in detecting the exact cube index. After considering the practical problems encountered during programming, which will be explained in the following part, a so-called blank layer has been built to surround the cubes to which their eight vertices may be assigned. This blank layer will occupy one unit on each side.

Satisfying the conditions mentioned above, we set the smallest specification of cube arrays as $48 \times 48 \times 14$. If the specification is smaller than these values, it will be difficult to understand the characteristics of the magnetic-field line structure. What is more, all the cubes are arranged in the first quadrant. We set three loops for setting the cube index. The bottom loop is counting.
the number of cubes along the $X$ axis, the middle loop, and the top loop are for $Y$ axis, and $Z$ axis, respectively. All the loops start from zero. Here, an equation is used to describe the cube index.

$$
\text{Cube index} = x + yn_{x_{\text{max}}} + zn_{y_{\text{max}}}m_{y_{\text{max}}} + 1
$$

(1)

Where $x, y$ and $z$ are count variables, $n_{x_{\text{max}}}$ and $m_{y_{\text{max}}}$ mean the maximum value of cubes along $X$ and $Y$ axes, respectively. Constant ‘1’ means the first cube which is closest to the origin of coordinates will be marked as ‘1’. Then, the next cube along the $X$ axis is marked as ‘2’. $n_{x_{\text{max}}}$ is the last cube index at $X$ axis marked as ‘48’. Then the second line starts. $y$ becomes ‘1’ and the counting of $X$ axis is reset. The cube index becomes ‘49’. Following this path, all cubes can be recognized and it is convenient to adjust the resolution of the cube arrays by setting a suitable length for each loop. In addition, it is simple to associate with the coordinates of points in space. This is greatly helpful when detecting the number of magnetic field lines going across each cube.

4.3. Detecting method by 3D digital differential analyzer (DDA)

Once we have the cube index, the number of the magnetic field lines which go across each cube can be recorded into a counter corresponding to the cube index. The detecting method to find which cube the line goes across is as follows.

Line segments with various lengths are connected to the magnetic field line. Furthermore, a line segment is decided by two points, the start point and the end point. The next line segment shares the start point with the previous line segment in which this point is the end point. Therefore, a magnetic field line can be separated into two parts. Only the beginning line segment is confirmed with its two points, the start and end points, and for the following line segments, their start points are omitted and their end points are confirmed.

First, for each point, its coordinate will be enlarged $n$ times, where $n$ is integer, according to the set cube arrays. Subsequently, the enlarged coordinate will be rounded to integer so that the equation (1) can also be applied to the calculation of the cube index to which the point should belong. Suppose that $a$ is a coordinate value from Table 1, it should be noticed that $\forall a \in [b, b + 1), b \in N$, and then $a$ will be rounded as $b$. Figure 4 takes a point as an example showing the process. Second, the value of each cube is initialized and assigned with zero. For the first two points, if they are inside the same cube, the value of this cube is just plus one. Otherwise, there will be two cubes whose values will be plus one, respectively. For the following each point, find the cube and check repetition within a cube. Eventually, a key-value pair list can be obtained.

The method we mentioned above is based on the condition that the distribution of the points is suitable compared with the resolution of the cube arrays shown in Fig.5. Generally,
the larger number of cubes the better the result will be. However, we should take the density of the volume data, that is, the number of points (green Xs) per cube, into account when we consider the points by using equation (1). When we suppose large cubes which are expressed by blue cubes in Fig.5, the density of the points is one per cube. While the density is one or zero, when we consider small cubes shown by black cubes, in other words, in the high-resolution case. In this situation, since the resolution of the cube arrays is much higher, the values of some cubes remain zero between two cubes in which the points exist. The zero-density cube seems to be a hole. To prevent this problem, 3D Digital Differential Analyzer (3D DDA) [23] is being applied. 3D DDA has many approaches. The 3D DDA used in this research is that presented by Amanatides and Woo in 1987 [24]. This incremental algorithm changes the float coordinate into integer coordinate and bases on the slope to detect the cubes crossed by the line which improves the computational efficiency. For the same time, this algorithm has no preferred axis. The detected cubes by 3D DDA are shown as yellow colored cubes in Fig.5.

When the 3D DDA returns the detected cubes as integer coordinate, the coordinate then can be directly transmitted into equation (1) and then continue to the second step mentioned above. The value of all cubes between the two cubes in which the green X points exit, as shown in Fig.5, will be plus one.

Figure 5: The magnetic field line (green lines) and the cubes. The magnetic field line is composed of several segments. The line segment is decided by two points (green X). Large cubes and small cubes are expressed by blue and black, respectively. The detected cubes by 3D DDA are colored by yellow.

4.4. The Larmor radius

The Larmor radius would improve the precision when predicting the overlapping of the plasma with the component in the reactor. The charged particle performs gyration motion along the magnetic field line. The radius of the gyration motion depends on the velocity of the particle and the strength of the magnetic field. Suppose a velocity which is perpendicular to the magnetic field as \( v_{\perp} \),

\[
v_{\perp} = v_{th} = \sqrt{\frac{2kT}{m}},
\]

where \( v_{th} \), \( k \), \( T \), and \( m \) represent the thermal velocity, the Boltzmann constant, the temperature and the mass of a charged particle, respectively. As a result, the Larmor radius can be calculated as \( r_L \).
where $q$ and $B$ are the electric charge of the charged particle and the intensity of the magnetic field at that point, respectively.

The next step is to connect two circular profiles. A simple way is to connect the points on the two circles with straight lines as shown in Fig.6. Suppose we have two circles with their normal vectors, $\mathbf{B}_k$ and $\mathbf{B}_{k+1}$, and their centers, $(x_k, y_k, z_k)$ and $(x_{k+1}, y_{k+1}, z_{k+1})$, $k = 1, 2, \cdots$. The number of points on each circle is supposed to be $L$ and the points on one circle are arranged at equal intervals on the circle. The coordinate of the point is $(x_{kl}, y_{kl}, z_{kl})$, $l = 1, 2, 3, \cdots, L$. The center angle of a fan-shape which is formed by the center of the circle and the adjacent points, $(x_{kl}, y_{kl}, z_{kl})$ and $(x_{kl+1}, y_{kl+1}, z_{kl+1})$, is $\alpha = 360^\circ/L$. We next consider how to decide the coordinate of $l = 1$ point on the first $k = 1$ circle as the start point which is on the connected line. We find an intersection point $(x_{11}, y_{11}, z_{11})$ with $x_{11} - x_1 > 0$ of the $k = 1$ circle and the line passing through the center of the $k = 1$ circle with direction vector $\mathbf{B}_1 \times \mathbf{B}_2 = (a, b, c)$, as equation (4) shows.

![Diagram](https://via.placeholder.com/150)

**Figure 6:** Relationship of the two $k$th and $(k+1)$th circles with the normal vectors, $\mathbf{B}_k$ and $\mathbf{B}_{k+1}$, and the lines connecting two points with the coordinates $(x_{kl}, y_{kl}, z_{kl})$ and $(x_{kl+1}, y_{kl+1}, z_{kl+1})$ on the $k$th and $(k+1)$th circles. The angle $\alpha$ is the center angle of a fan-shape which is formed by the center of the circle and the adjacent points, $(x_{kl}, y_{kl}, z_{kl})$ and $(x_{kl+1}, y_{kl+1}, z_{kl+1})$.

\[
\left\{ \begin{array}{l}
(x_{kl} - x_k)^2 + (y_{kl} - y_k)^2 + (z_{kl} - z_k)^2 = r_{Lk}^2 \\
x_{kl} - x_k = \frac{y_{kl} - y_k}{b} = \frac{z_{kl} - z_k}{c}
\end{array} \right.
\]

(4)

where $r_{Lk}$ is the Larmor radius around the magnetic field $\mathbf{B}_k$ at the point $(x_k, y_k, z_k)$, $k = l = 1$ and $x_{kl} - x_k > 0$. After we capture the start point, the remaining points on the $k = 1$ circle can be found by changing $l$ according to the following equation (5),

\[
\left\{ \begin{array}{l}
(x_{kl} - x_k)^2 + (y_{kl} - y_k)^2 + (z_{kl} - z_k)^2 = r_{Lk}^2 \\
(x_{kl}, y_{kl}, z_{kl}) \cdot \mathbf{B}_k = 0 \\
(x_{kl} - x_k) \times a + (y_{kl} - y_k) \times b + (z_{kl} - z_k) \times c = r_{Lk}^2 \times \cos(l-1)\alpha
\end{array} \right.
\]

(5)

where, $k = 1, l = 2, 3, \cdots, L$, and $x_{kl} - x_k > 0$. The points on the next circles with $k > 2$
are also decided by the above method. Only when $k < 3$ do the circles share the same direction vector.

In this way, 3D DDA can also be applied to find the additional cubes. At this stage, we set $L = 8$, that is eight points are arranged on the circle. However, it is possible to generate multilayer with different points to describe the Larmor radius according to the density of the cubes. First of all, in order to verify our algorithm, we ignore the dependencies of the Larmor radius and the plane of the particle gyration motion on the magnetic field $B$ described in equation (3) and Fig.6 at this stage, respectively. The plane on which the particle performs gyration motion has a normal vector parallel to the vector of magnetic field $B$. We let the Larmor radius be constant, which is equal to 0.3, and the plane of the gyration motion be parallel to the poloidal plane in order to reduce the calculation cost.

4.5. Marching cubes method

The main process of this algorithm remains the same. For each cube, there are eight vertices with values. A threshold then is selected for deciding the triangular elements. Here, we want to put emphasis on how the value of each vertex is decided.

In our paper, a value of the vertex of the cube is an average value calculated from the values of eight related cubes. There are twenty-six cubes needed to be captured. A cube has attributes described as ‘cube index’ and ‘value’. Suppose that there is a cube with cube index $n$. This cube with the eight adjacent cubes encompasses it at the same layer, and it can be written by a Matrix C in equation (6) as follows:

$$
\begin{bmatrix}
    n - n_{x_{\text{max}}} - 1 & n - n_{x_{\text{max}}} & n - n_{x_{\text{max}}} + 1 \\
    n - 1 & n & n + 1 \\
    n + n_{x_{\text{max}}} - 1 & n + n_{x_{\text{max}}} & n + n_{x_{\text{max}}} + 1
\end{bmatrix}
$$

(6)

This way of representation shares the same idea of equation (1). And the bottom layer can be obtained by using Matrix C from which are subtracted $n_{x_{\text{max}}} \times n_{y_{\text{max}}}$, while the upper layer can be gained by Matrix C which is added with $n_{x_{\text{max}}} \times n_{y_{\text{max}}}$. However, when a cube belongs to the outermost layer of the cube arrays, this searching method will be incorrect. Because of the continuous sequence we used to mark the cubes, the program is misled into finding some cubes that should not exist at the opposite surface. This is the reason why a blank layer is necessary. After finding the Matrix C, the value of the vertices then can be calculated.

A situation could be imagined in which only a magnetic field line passes through one of the eight cubes. Therefore, the threshold should be smaller than 0.125 and has been decided as 0.1249. This helps prevent some corners of the model from being too sharp.

4.6. Smoothing and reducing the polygon mesh

The triangular elements generated by the Marching cubes algorithm are changed into an object (Obj) file by the program automatically. The Obj file is imported into the Shade 3D, a commercial graphic software for furthering optimization and display [25]. A polygon smoothing function and a polygon reduction function provided by Shade 3D is applied. The polygon smoothing function smooths the irregularities in a polygon mesh by moving the vertices. While the reduction function reduces the number of the faces of the polygon mesh and retains its shape to the
extent possible. The reason for applying these functions will be explained in the next section. Since the threshold is less than vertex value of a cube whose value is equal or greater than one, triangular elements are definitely generated at one cube or some amount of adjacent cubes around this cube. This results in generating a space between the line and the triangular elements. The space can serve as a buffer area when applying the smoothing and reducing function. The modification of the model proves to be possible. It contributes to improving the rendering result.

5. Visualization

The polygon meshes with wireframe and its Phong shading results are shown in the first and second columns in Fig. 7, respectively. The figures in the left side show the original polygon mesh, while the figures in the right side show the modified polygon mesh.

Figure 7: The polygon meshes with wireframe and its Phong shading results. (a) The wireframe of the polygon mesh generated by our program directly. (b) The wireframe of the polygon mesh after utilizing the polygon smoothing and reducing function in Shade 3D. (c) The result under Phong shading for the original polygon mesh. (d) The result under Phong shading for modified by Shade 3D.

The roughness of the original polygon is attributed to two reasons. The first is the distribution of the magnetic field lines, the twisted structure of magnetic field, and the chaotic paths of the magnetic field lines around periphery region [6]. At some places, the lines are close to each other. In contrast, other places are sparse. These sparse places will cause wave-like and fluctuating surface. The second reason is the restriction of the threshold. The scalar value of usual volume data has a smooth transition because they describe a continuous changing, such as the temperature. In this paper, the scalar data expresses the number of magnetic field lines.
For those cubes through which no magnetic field lines go across, the scalar value is set at zero, while scalar values in the other cubes may reach 10 or more. Even though taking the average value, it is much greater than the chosen threshold 0.1249. The insertion point is pushed to a vertex. This causes a surface similar to fish scale pattern under Phong shading.

As shown in Figs. 7 (b) and (d), Shade 3D helps modify the model and seems to perform well. However, the modified model by Shade 3D needs to be proved that the plasma shape is correctly represented. We will introduce our evaluation in the next section. The model with color and simple transparency adjustment is shown in Fig. 8.

![Figure 8: The visualization result with Shade 3D](image)

The inner and outer layers can be recognized from the model, which indicates that Marching cubes algorithm is suitable for the approximate external form of the plasma shape. It can also be observed from the model that some parts have deeper color than the color around it. This may be solved by the selection of the rendering parameter. We are attempting to provide a better visualization result.

We consider that the size of a cube used in the marching cube method may have an impact on the visualization result. Dependence of the cube size will be discussed in our future work.

## 6. Evaluation

As mentioned above, it is necessary to check the accuracy of the modified model. Here, the model of the plasma shape generated by marching cubes is exported from Shade 3D with the Obj file. The coordinate and the sequence of the vertices from each triangular element can be found in this file.

Since direct and efficient comparison between the 3D plasma shape and the original magnetic line data is difficult to perform, we chose returning to the two-dimensional way by cutting the model into slices, then adding original magnetic field line data on the slice with the same
angle to evaluate the performance of our method.

Taking one slice as an example at first, a vertical plane is chosen to cut the plasma shape model at a certain degree for obtaining the exterior and internal contours. The contours consist of intersection points of the triangular elements with the plane. Second, the magnetic field lines are also cut by the vertical plane with the same angle as that of the plane on which the plasma shape model is cut. We draw circles with the Larmor radius around the points of the cut lines on the vertical plane, respectively. All the points with the circles are found as the ground truth in this evaluation. Finally, we plot those two diagrams on one scene. Some representative figures are selected and shown in Fig. 9.

![Figure 9: The comparison of the results at (a) 45, (b) 60 and (c) 315 degrees, respectively. The green points indicate the slice of the model. The blue circles are the points from magnetic field lines with their Larmor radius.](image)

From Fig. 9(a), it is proved that the 3D model on the plane with 45 degrees covers those circles of the original data with no leaking. Most of the figures on the plane with the other degrees show the same result. However, there are a few notable cases. The upper part of the Fig. 9(b) shows a spare area with no circles inside this area. This is because a convex part like a bowl covers up the end of each magnetic field line. These convex parts affect the contour at the next degree because the two slices are very close. As to Fig. 9(c), the contour is non-closed as the circles are sparse there. In addition, a few circles are partly located at the outside of the contour. We believe that the structure of polygons there is quite sensitive to the effect of smoothing and reducing. We will try to improve our program instead of relying on Shade 3D completely.

What is more, the margin between the contour and the circles appears in all cases. This is partly caused by the Marching cubes method itself. In our program, the triangular elements are definitely generated at one or some adjacent cubes around the cube where a line goes across due to the threshold set as we mentioned in Sec.4.6. It is possible to reduce the margin by setting a larger cube array. Nevertheless, the separation of the contour will occur and the calculation time also will increase. Furthermore, a quantitative standard is under discussion.

7. Summary
7.1. Conclusion

In this paper, we study the plasma shape by generating a 3D model through the magnetic field lines. The space between the plasma shape and the inner components of the FFHR is vital for the designing and the managing of the reactor. The widely used analysis methods are mostly based on profiles for inspection. Three-dimensional approaches such as visualizing the magnetic field as streamlines are far from satisfactory due to the interweaving. What is more, it is difficult to find the critical magnetic field lines, of which the plasma shape is composed and lacks the appropriate method representing the Larmor radius that is one of the indispensable methods to check interference. Instead of using streamlines, we concentrated on finding the surface that can describe the plasma shape.

Considering the type of the data, Marching cubes method is applied in this paper for detecting the isosurface. The scalar data is described as the number of the magnetic field lines within a cube. Some searching methods are proposed and combined with this algorithm. Further, Larmor radius can be simply added to the calculation and becomes an integrated part of the 3D model. Shade 3D helps to modify the polygon mesh. To verify the accuracy of the modified model by Shade 3D, we compare the profiles of magnetic field lines in each degree with the slices of our model at the corresponding degrees.

With the approach we proposed, the plasma shape can be outlined directly from the points on magnetic field lines. This will reduce the inspection time of a reactor. The validity is also improved because Larmor radius is available at meantime. However, some visualization elements are expediently added to the model.

7.2. Future work

The first part regards the plasma shape model. Although the roughness of the model has been alleviated through Shade 3D, the interpolation toward each cube remains to be improved. A better mathematical solution is expected to overcome the problem within the stage of programming. At the same time, the margin between the plasma shape and the point group with Larmor radius in the figure should be reduced to as small as possible. In addition, finding a suitable rendering scheme for our model is another issue.

The second part regards the goal of this research. By arranging the plasma shape model into the reactor model, detecting the overlaps automatically is expected. A comparison will be made between this and NIFS’s methods. After taking advice from the domain expert in NIFS, we will decide our next direction.

In terms of visualization, this work can also be applied for different purposes. Except for the engineering analysis, the model can be used as a popularization of science. We would like to use this work for explaining how fusion works safely and has the prospect of solving the energy crisis.

Acknowledgments

This work is supported by the NIFS within the Fusion Engineering Research Project (FERP). This work is also performed with the support and under the auspices of the NIFS Collaboration Research program (NIFS19KNTS057, NIFS18KKGS022 and NIFS17UFFF040), and some calculation is performed on "Plasma Simulator" of NIFS.
References

[1] Pironti, A., & Walker, M. (2005). Fusion, tokamaks, and plasma control: an introduction and tutorial. *IEEE Control Systems Magazine*, 25(5), 30-43.

[2] Ambrosino, G., & Albanese, R. (2005). Magnetic control of plasma current, position, and shape in Tokamaks: a survey or modeling and control approaches. *IEEE Control Systems Magazine*, 25(5), 76-92.

[3] Hartmann, D. A. (2010). Stellarators. *Fusion science and technology*, 57(2T), 46-58.

[4] Sagara, A., Tamura, H., Tanaka, T., Yanagi, N., Miyazawa, J., Goto, T., ... & Takayama, S. (2014). Helical reactor design FFHR-d1 and c1 for steady-state DEMO. *Fusion Engineering and Design*, 89(9-10), 2114-2120.

[5] Goto, T., Miyazawa, J., Tanaka, T., Yanagi, N., & Sagara, A. (2011). A robust design window for the heliotron DEMO reactors. *Plasma and Fusion Research*, 6, 2405083-2405083.

[6] Kobayashi, M., Masuzaki, S., Yamada, I., Narushima, Y., Suzuki, C., Tamura, N., ... & Yoshimura, S. (2013). Control of 3D edge radiation structure with resonant magnetic perturbation fields applied to the stochastic layer and stabilization of radiative divertor plasma in LHD. *Nuclear Fusion*, 53(9), 093032.

[7] Itagaki, M., Maeda, T., Ishimaru, T., Okubo, G., Watanabe, K., Seki, R., & Suzuki, Y. (2011). Three-dimensional Cauchy-condition surface method to identify the shape of the last closed magnetic surface in the Large Helical Device. *Plasma Physics and Controlled Fusion*, 53(10), 105007.

[8] Itagaki, M., Ishimaru, K., Matsumoto, Y., Watanabe, K., Seki, R., & Suzuki, Y. (2013). Improved three-dimensional CCS method analysis for the reconstruction of the peripheral magnetic field structure in a finite beta helical plasma. *Plasma and Fusion Research*, 8, 1402134-1402134.

[9] Peikert, R., & Sadlo, F. (2007, May). Visualization methods for vortex rings and vortex breakdown bubbles. In *Proceedings of the 9th Joint Eurographics/IEEE VGTC conference on Visualization*, Eurographics Association, pp. 211-218

[10] Peikert, R., & Sadlo, F. (2009). Flow topology beyond skeletons: Visualization of features in recirculating flow. In *Topology-Based Methods in Visualization II*. Springer, Berlin. pp. 145-160

[11] Chiariello, A. G., Formisano, A., & Martone, R. (2013). Fast magnetic field computation in fusion technology using GPU technology. *Fusion Engineering and Design*, 88(9-10), 1635-1639.

[12] Watanabe, T., Matsumoto, Y., Hishiki, M., Oikawa, S., Hojo, H., Shoji, M., ... & Mutoh, T. (2006). Magnetic field structure and confinement of energetic particles in the LHD. *Nuclear fusion*, 46(2), 291.

[13] Ohtani, H., Tamura, Y., Kageyama, A., & Ishiguro, S. (2011). Scientific visualization of plasma simulation results and device data in virtual-reality space. *IEEE Transactions on
[14] Tanaka, K., Tanaka, S., Hasegawa, K., Murotani, K., & Koshizuka, S. (2014). translucent visual analysis of large scale 3D point data generated by particle fluid simulation of tsunami water. *Journal of Advanced Simulation in Science and Engineering*, 39(11), 2472-2473.

[15] Newman, T. S., & Yi, H. (2006). A survey of the marching cubes algorithm. *Computers & Graphics*, 30(5), 854-879.

[16] Komori, A., Yamada, H., Imagawa, S., Kaneko, O., Kawahata, K., Mutoh, K., ... & Nagayama, Y. (2010). Goal and achievements of large helical device project. *Fusion Science and Technology*, 58(1), 1-11.

[17] Hirshman, S. P., & Whitson, J. C. (1983). Steepest-descent moment method for three-dimensional magnetohydrodynamic equilibria. *The Physics of Fluids*, 26(12), 3553-3568.

[18] Hirshman, S. P., & Merkel, P. (1986). Three-dimensional free boundary calculations using a spectral Green's function method. *Computer Physics Communications*, 43(1), 143-155.

[19] Terranova, D., Gobbin, M., Boozer, A. H., Hirshman, S. P., Marrelli, L., & Pomphrey, N. (2010). Self-Organized Helical Equilibria in the RFX-Mod Reversed Field Pinch. *Contributions to Plasma Physics*, 50(8), 775-779.

[20] Project webpage. [https://github.com/yasuhiro-suzuki/MGTRC](https://github.com/yasuhiro-suzuki/MGTRC).

[21] Itagaki, M., Maeda, T., Ishimaru, T., Okubo, G., Watanabe, K., Seki, R., & Suzuki, Y. (2011). Three-dimensional Cauchy-condition surface method to identify the shape of the last closed magnetic surface in the Large Helical Device. *Plasma Physics and Controlled Fusion*, 53(10), 105007.

[22] Min Chen, Arie E. Kaufman, & Roni Yagel. (2000). *Volume Graphics*. Springer.

[23] Chang, A. Y. (2001). A survey of geometric data structures for ray tracing. Polytechnic University, Department of Computer and Information Science.

[24] Amanatides, J., & Woo, A. (1987, August). A fast voxel traversal algorithm for ray tracing. In *Eurographics*, Vol. 87, No. 3, pp. 3-10.

[25] Shade 3D, FORUM8 Co. Ltd., [https://shade3d.jp/en/](https://shade3d.jp/en/)