

托卡马克装置两种磁场合磁场的磁力线形貌

Log Creative

2020 年 12 月 1 日

1 问题描述

问题 9-5 9-6

研究受控热核反应的托卡马克装置（图1）中，用螺绕环产生的磁场来约束其中的等离子体。设某一托卡马克装置中环管轴线的半径为 $R = 2.0\text{m}$ ，管截面半径为 $r = 1.0\text{m}$ ，环上均匀绕有 $l = 10\text{km}$ 长的水冷铜线。

等离子体除了受到螺绕环电流的磁约束外，还会受到自身的感应电流产生的磁场的约束。试分析这两种磁场的合磁场的磁力线形貌。

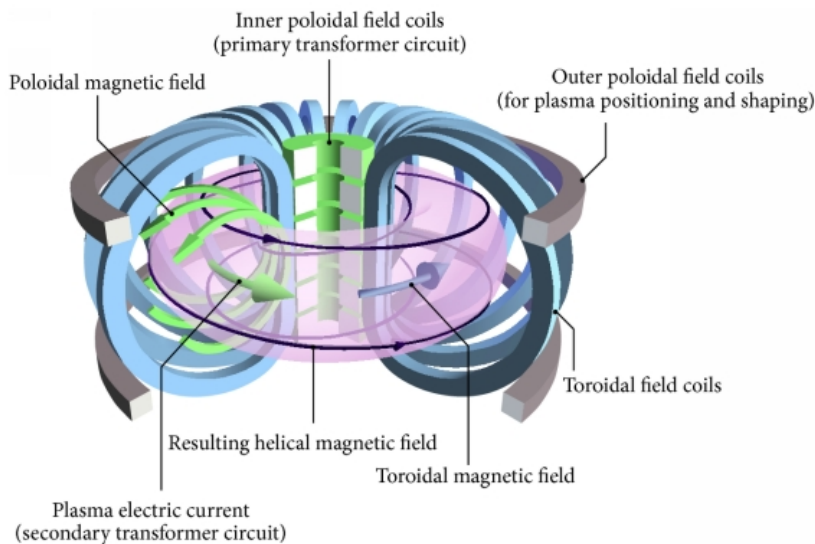


图 1: 托卡马克装置^[1]

2 构建外壳

托卡马克外壳主体是一个环面，我参考了百度百科^[2]上对于环面的数学定义式：

$$\begin{cases} x(u, v) = (R + r \cos v) \cos u \\ y(u, v) = (R + r \cos v) \sin u \\ z(u, v) = r \sin v \end{cases}$$

其中 $u, v \in [0, 2\pi]$ ， R 是管子的中心到画面的中心的距离， r 是圆管的半径。

然后参考了 scipython 上关于环面的示例代码^[3]，采用分段的方式生成环面，如图2所示，代码见附录A。

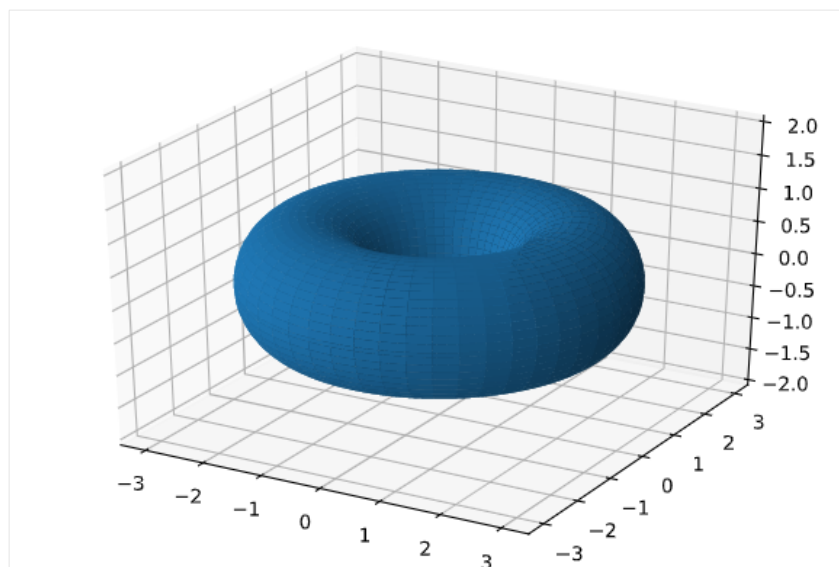


图 2: 环面

3 环向场线圈产生的磁力线

我们近似地将环向场线圈产生的磁力线看作环内均匀的，螺绕环的总匝数为

$$N = \frac{l}{2\pi r} \quad (1)$$

则由安培环路定理可得，

$$B \cdot 2\pi R = \mu_0 NI \quad (2)$$

$$B = \frac{\mu_0 l I}{4\pi^2 R r} = 11.62 \text{T}$$

我们观察最外圈的磁力线，也是一个环形，如图3所示，代码见附录B。

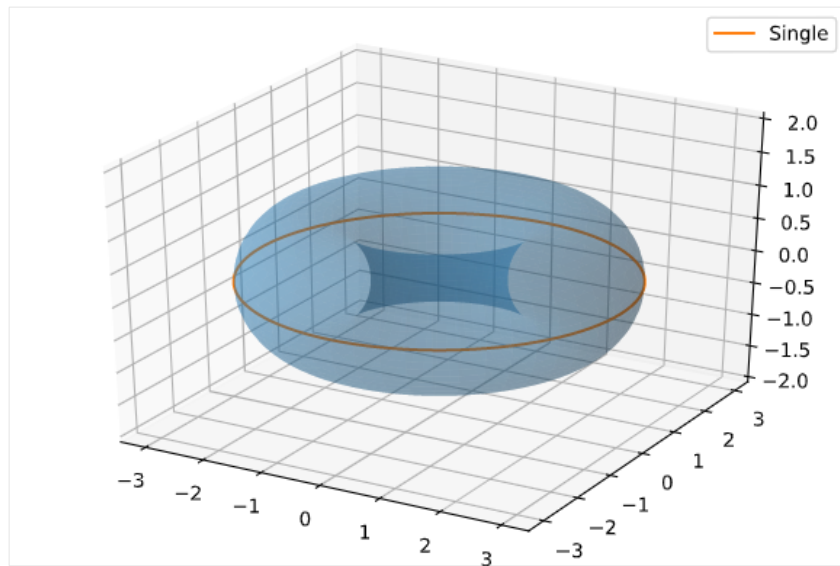


图 3: 环向场线圈产生的磁力线

4 自身的感应电流产生的磁场

如果要将自身的感应电流与其合成，就不再使用上一节的解析画法了。我在此处使用微元法计算每一步的磁力线偏移。

如图4所示，计算由环向场线圈产生的切向磁感应强度的方向向量，首先定义当前位置的半径变量：

$$R_t = \sqrt{x^2 + y^2}$$

然后计算切向方向向量

$$\mathbf{e}_1 = \left(-\frac{y}{R_t}, \frac{x}{R_t}, 0 \right)$$

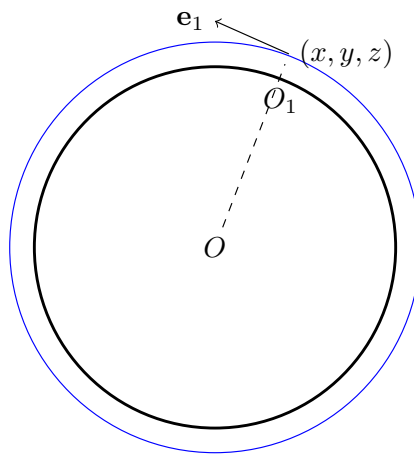


图 4: 切向磁感应强度方向向量 \mathbf{e}_1

然后计算该位置由自身的感应电流产生的小圆切向磁感应强度方向向量。首先计算小圆的中心位置

$$O_1 = \left(R \frac{x}{R_t}, R \frac{y}{R_t}, 0 \right)$$

然后计算小圆半径

$$r' = \left(\frac{R}{R_t} - 1 \right)^2 (x^2 + y^2) + z^2$$

如图5所示，计算小圆切向磁感应强度方向向量

$$\mathbf{e}_2 = \left(-\frac{z}{r'} \frac{x}{R_t}, -\frac{z}{r'} \frac{y}{R_t}, \frac{R - R_t}{r'} \right)$$

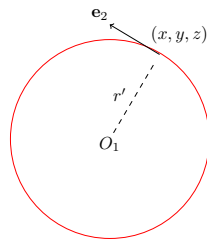


图 5: 小圆切向磁感应强度方向向量 \mathbf{e}_2

则两者合成的效果如图6所示，由于现有知识无法完全量化环流等离子体的磁场，这里按照直线电流产生的磁场与半径成反比来计算：

$$B_2 \propto \frac{1}{r'} \quad (3)$$

大小关系仅为推测，代码见附录C。

会发现磁力线有一定的偏移，这所带来的利处在于可以增加磁感应强度，增加约束；弊处在于会增大等离子体的活动空间，对管壁形成冲击，减少设备的使用寿命，所以必须进行修正。

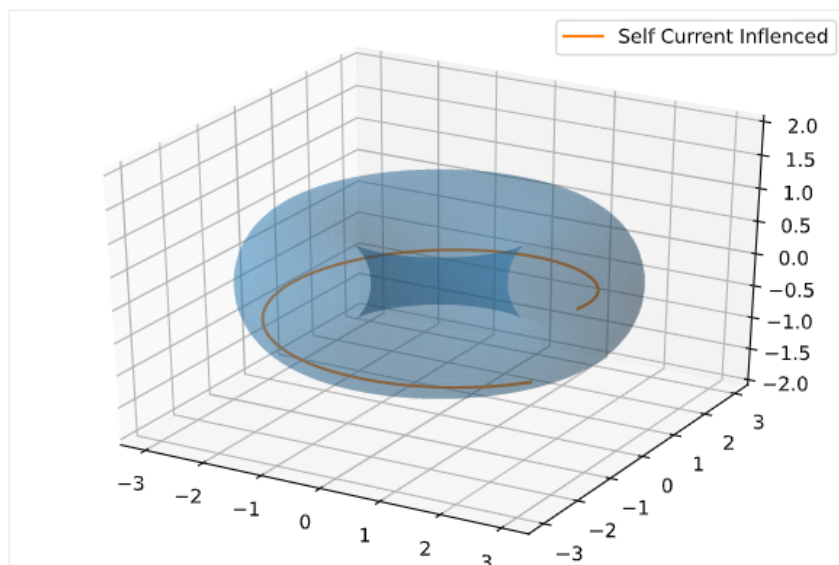


图 6: 合成磁力线

所以还需要极向场线圈进行修正，由于这一部分已经完全超过目前所学的知识^[4]，不再深入探究：

$$\begin{cases} B_r &= -\frac{1}{2\pi r} \frac{\partial \Phi_{\text{core}}^0}{\partial z} = 0 \\ B_z &= \frac{1}{2\pi r} \frac{\partial \Phi_{\text{core}}^0}{\partial r} = -\frac{\mu_0 I_s}{Z_{\text{Fe}}} \end{cases}$$

$$\begin{cases} B_r &= -\frac{1}{2\pi r} \frac{\partial \Phi_{\text{vacuum}}^s}{\partial z} = \mu_0 I_s \frac{z-Z_s}{r} \frac{1}{\sqrt{(r+R_s)^2+(z-Z_s)^2}} \left(-K(k^2) + \frac{r^2+R_s^2+(z-Z_s)^2}{(r-R_s)^2+(z-Z_s)^2} E(k^2) \right) \\ B_z &= \frac{1}{2\pi r} \frac{\partial \Phi_{\text{vacuum}}^s}{\partial r} = \mu_0 I_s \frac{1}{\sqrt{(r+R_s)^2+(z-Z_s)^2}} \left(K(k^2) + \frac{r^2+R_s^2+(z-Z_s)^2}{(r-R_s)^2+(z-Z_s)^2} E(k^2) \right) \end{cases}$$

$$\begin{cases} B_r &= -\frac{1}{2\pi r} \frac{\partial \Phi_{\text{vacuum}}^m}{\partial z} = \mu_0 I_s \frac{z-Z_m}{r} \frac{1}{\sqrt{(r+R_m)^2+(z-Z_m)^2}} \left(-K(k^2) + \frac{r^2+R_m^2+(z-Z_m)^2}{(r-R_m)^2+(z-Z_m)^2} E(k^2) \right) \\ B_z &= \frac{1}{2\pi r} \frac{\partial \Phi_{\text{vacuum}}^m}{\partial r} = \mu_0 I_s \frac{1}{\sqrt{(r+R_m)^2+(z-Z_m)^2}} \left(K(k^2) + \frac{r^2+R_m^2+(z-Z_m)^2}{(r-R_m)^2+(z-Z_m)^2} E(k^2) \right) \end{cases}$$

5 感想

通过这个学期参加 python 物理模型及图像绘制，我对物理问题的解决有了更加直观的认识，培养了我对问题的直觉。感谢老师提供这样的一个机会，写一个非常有意义的大作业，拓宽了我的视野，提高了我的潜在物理水平。

References

- [1] LI S, JIANG H, REN Z, et al. Optimal tracking for a divergent-type parabolic pde system in current profile control[J/OL]. Abstract and Applied Analysis, 2014, 2014: 8. <https://doi.org/10.1155/2014/940965>.
- [2] 百度百科. 环面[EB/OL]. <https://baike.baidu.com/item//2127101?fr=aladdin>.
- [3] SCIPYTHON. A torus[EB/OL]. <https://scipython.com/book/chapter-7-matplotlib/examples/a-torus/>.
- [4] 程际. J-TEXT 托卡马克三维磁场分布研究[D]. 华中科技大学: 华中科技大学, 2014.

A 生成环面

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 from mpl_toolkits.mplot3d import Axes3D
4
5 R = 2.0
6 r = 1.0
7
8 SEG = 100
9
10 u = np.linspace(0, 2. * np.pi, SEG)

```

```

11 v = np.linspace(0, 2. * np.pi, SEG)
12 u, v = np.meshgrid(u, v)
13
14 x = (R + r * np.cos(v)) * np.cos(u)
15 y = (R + r * np.cos(v)) * np.sin(u)
16 z = r * np.sin(v)
17
18 fig1 = plt.figure()
19 ax = fig1.add_axes([0, 0, 1, 1], projection='3d')
20 ax.set_zlim(-2,2)
21 ax.plot_surface(x, y, z, alpha = 1)
22 fig1.show()

```

B 环向场线圈产生的磁力线

```

1 theta = np.linspace(0, 2*np.pi, SEG)
2 x1 = (R + r) * np.cos(theta)
3 y1 = (R + r) * np.sin(theta)
4 z1 = np.linspace(0, 0, SEG)
5
6 fig2 = plt.figure()
7 ax2 = fig2.add_axes([0, 0, 1, 1], projection='3d')
8 ax2.set_zlim(-2,2)
9 ax2.plot_surface(x, y, z, alpha = 0.3)
10 ax2.plot(x1,y1,z1, label='Single')
11 plt.legend(loc = 0, ncol = 2)
12 fig2.show()

```

C 自身的感应电流产生的磁场

```

1 STARTR = R + 0.3 * r
2
3 point = np.array([STARTR, 0, 0])
4 x3 = [point[0]]
5 y3 = [point[1]]
6 z3 = [point[2]]
7
8 DETAIL = 100000
9 STEP = 2 * np.pi * STARTR / DETAIL
10 SUBSTEP = STEP / 30
11 for i in range(0, round(DETAIL*1)):
12     xp = point[0]
13     yp = point[1]
14     zp = point[2]

```

```

15 Rt = (xp * xp + yp * yp) ** 0.5
16 point = [point[0] - yp / Rt * STEP, point[1] + xp / Rt * STEP, point[2]]
17 rr = ((R - (xp*xp + yp*yp)**0.5)**2 + zp*zp) ** 0.5
18 point = [point[0] - (zp / rr * xp / Rt) * SUBSTEP * 1 / rr, point[1] - (zp / rr *
    yp / Rt) * SUBSTEP * 1 / rr, point[2] + (R - Rt) / rr * SUBSTEP * 1 / rr]
19 x3.append(point[0])
20 y3.append(point[1])
21 z3.append(point[2])
22
23 fig3 = plt.figure()
24 ax3 = fig3.add_axes([0, 0, 1, 1], projection='3d')
25 ax3.set_zlim(-2,2)
26 ax3.plot_surface(x, y, z, alpha = 0.3)
27 ax3.plot(x3,y3,z3, label='Self_Current_Influenced')
28 plt.legend(loc = 0, ncol = 2)
29 # ax3.view_init(90, 0)
30 fig3.show()

```