

САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ им. ПЕТРА ВЕЛИКОГО  
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ И МЕХАНИКИ  
КАФЕДРА ПРИКЛАДНАЯ МАТЕМАТИКА

ЛАБОРАТОРНАЯ РАБОТА №3  
ПОСТРОЕНИЕ РАЗБИЕНИЯ ОБЛАСТИ МЕЖДУ  
ЗАМКНУТЫМИ КРИВЫМИ  
по дисциплине: ВЫЧИСЛИТЕЛЬНЫЕ КОМПЛЕКСЫ

Студент группы 3630102/60201

Митрофанова А.Г.

Преподаватель

Баженов А.Н.

Санкт-Петербург  
2020 год

## Содержание

<b>1</b>	<b>Постановка задачи</b>	<b>3</b>
<b>2</b>	<b>Теория</b>	<b>3</b>
2.1	Вычисление радиуса кривизны . . . . .	3
2.2	Построение разбиения области . . . . .	4
<b>3</b>	<b>Реализация</b>	<b>4</b>
<b>4</b>	<b>Результаты</b>	<b>5</b>
<b>5</b>	<b>Обсуждение</b>	<b>6</b>
<b>6</b>	<b>Литература</b>	<b>7</b>
<b>7</b>	<b>Приложение</b>	<b>7</b>
7.1	main.m . . . . .	7
7.2	find_curv_radius.m . . . . .	8
7.3	find_center.m . . . . .	8
7.4	split_sector_simple.m . . . . .	9
7.5	split_sector_N.m . . . . .	9

## Список иллюстраций

1	Радиус кривизны . . . . .	4
2	График сепаратрисы . . . . .	5
3	График радиуса кривизны . . . . .	5
4	График радиуса кривизны после медианного сглаживания . . . . .	6
5	Разбиение сетки для сепаратрисы . . . . .	6

# 1 Постановка задачи

Необходимо считать данные о сепаратрисе из g-файла. Построить кривую сепаратрисы на графике, вычислить и отметить магнитную ось. Также необходимо вычислить радиус кривизны в каждой точке сепаратрисы и применить к полученным результатам медианное сглаживание. Построить разбиение области сепаратрисы на сетку.

## 2 Теория

На сечении Токамак «Глобус-М» можно увидеть следующие компоненты: стенки вакуумной камеры, сепаратриса (граница плазмы, последняя замкнутая область магнитного потока) и магнитная ось (условный центр тока плазмы). [1]

Данные о геометрическом месте сепаратрисы содержатся в так называемых g-файлах в виде массивов **RBDY**, **ZBDY**. Функция MATLAB

$$[flux, RBDY, ZBDY, NBDY, R, Z, time, rdim, zdim] = \\ = gfile\_extractor\_1t(shot\_number\_test, start\_efit\_time, MagMesh)$$

[2] возвращает

- **RBDY**, **ZBDY** –  $(R, Z)$ -координаты точек сепаратрисы
- **NBDY** – число точек сепаратрисы
- **rdim**, **zdim** – размер сетки по радиусу и вертикали
- **R**, **Z** – координаты точек сетки по радиусу и вертикали
- **flux** – двумерный массив магнитного потока

В окрестности минимума распределения **flux** находится «магнитная ось». Область «магнитной оси» можно считать «центром» плазмы.

### 2.1 Вычисление радиуса кривизны

По теореме синусов

$$\frac{|AB|}{\sin \angle ACB} = 2 \cdot R \quad (1)$$

По скалярному произведению

$$(BC, AC) = |BC| \cdot |AC| \cdot \cos \angle ACB \quad (2)$$

Из (2) можем найти косинус  $\angle ACB$

$$\cos \angle ACB = \frac{(BC, AC)}{|BC| \cdot |AC|} \quad (3)$$

Из (3) найдём синус  $\angle ACB$

$$\sin \angle ACB = \sqrt{1 - \cos^2 \angle ACB} = \sqrt{1 - \left( \frac{(BC, AC)}{|BC| \cdot |AC|} \right)^2} \quad (4)$$

Из (1) и (4) найдем радиус кривизны

$$R = \frac{|AB|}{2 \cdot \sqrt{1 - \left( \frac{(BC, AC)}{|BC| \cdot |AC|} \right)^2}} \quad (5)$$

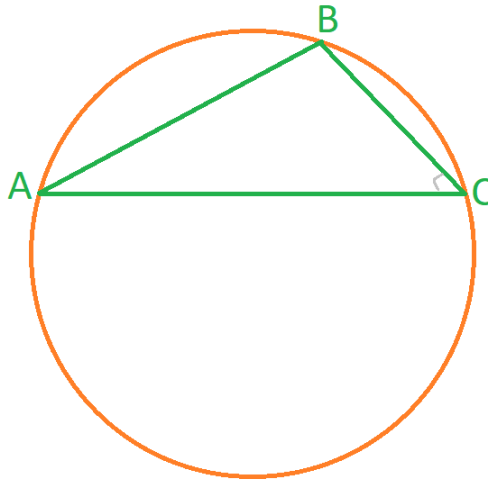


Рис. 1: Радиус кривизны

## 2.2 Построение разбиения области

- Делим область на 2 сектора по экватору
- В каждом секторе выбираем точку с максимальным радиусом кривизны для нахождения опорного направления
- Находим в каждом секторе точку с минимальным радиусом кривизны
- Найдены 4 особые точки. Соединив их с центром получим 4 сектора
- Внутри каждого сектора проводим еще  $N$  дополнительных отрезков. Получено  $4N$  отрезка
- Находим у каждого из отрезков середину и соединяем их между собой

## 3 Реализация

Реализация осуществлена на языке программирования MATLAB в среде разработки MATLAB Online R2019b.

Для считывания информации о сепаратрисе из g-файла была использована функция *gfile\_extractor\_1t* [2]

## 4 Результаты

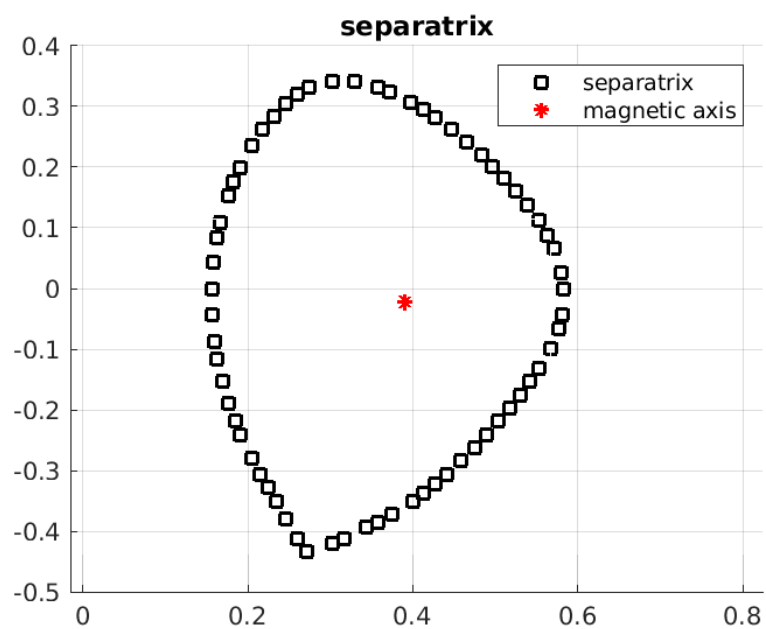


Рис. 2: График сепаратрисы

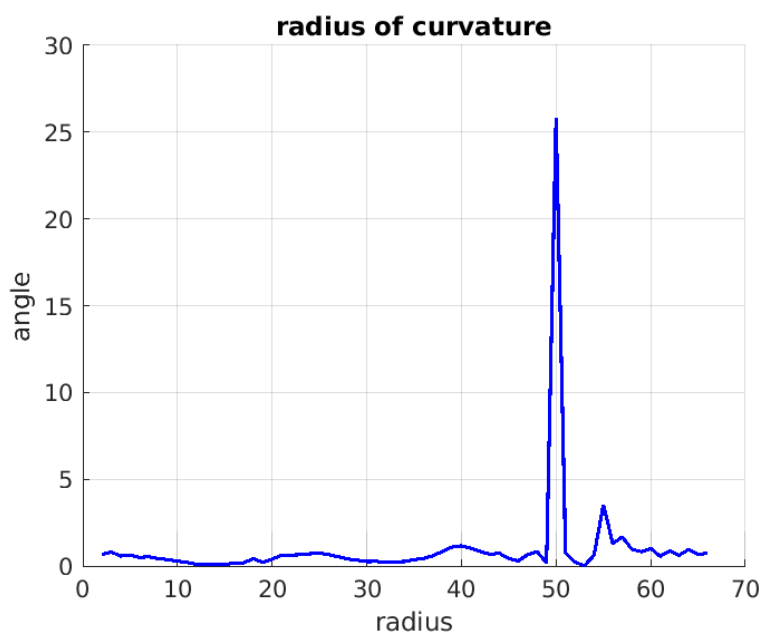


Рис. 3: График радиуса кривизны

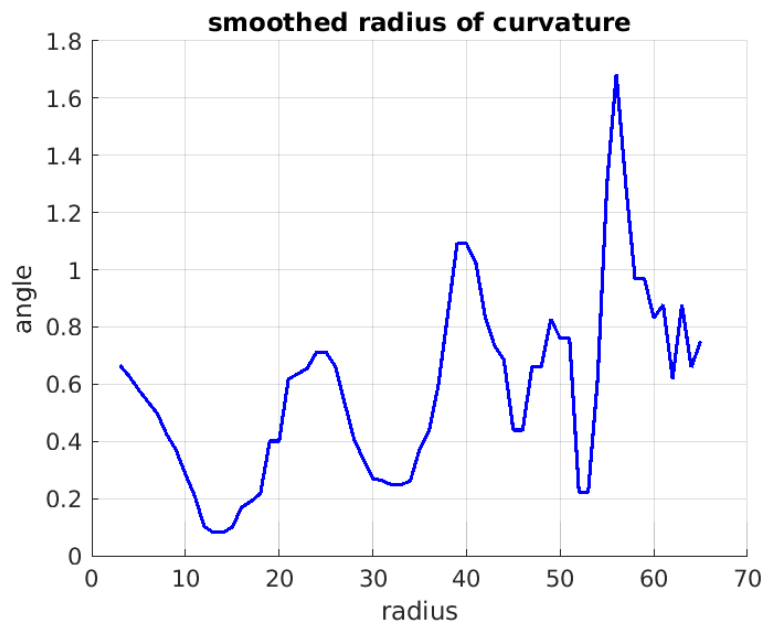


Рис. 4: График радиуса кривизны после медианного сглаживания

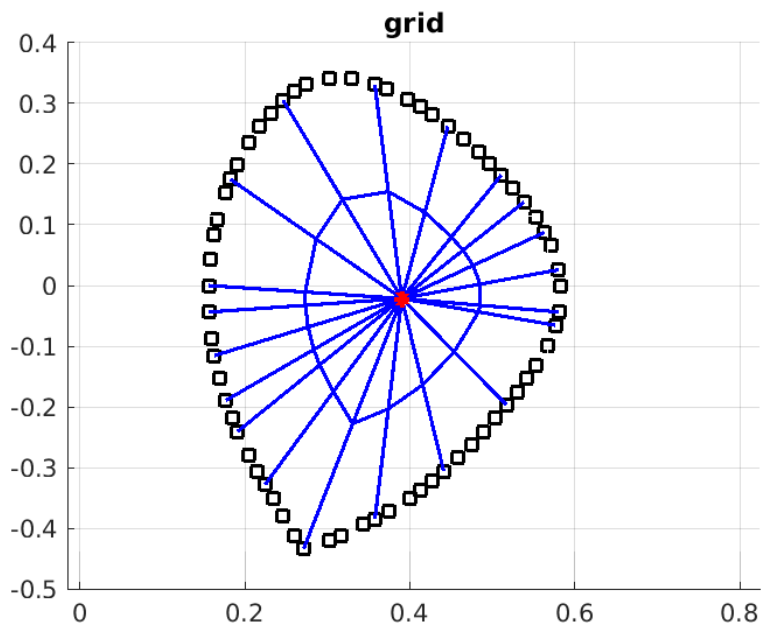


Рис. 5: Разбиение сетки для сепаратрисы

## 5 Обсуждение

После медианного сглаживания на графике радиуса кривизны стали отчётливее видны пики и впадины. Было получено разбиение пространства.

## 6 Литература

### Список литературы

- [1] А.Н. Баженов, Лабораторный практикум. Методический материал. «Вычислительные комплексы» [Электронный ресурс, облачное хранилище]. Режим доступа:  
<https://cloud.mail.ru/public/4ra6/5wwqBzMBC/LabPractics.pdf> (дата обращения: февраль, 2019 г.)
- [2] Код функции, извлекающей информацию о сепаратрисе [Электронный ресурс, облачное хранилище]. Режим доступа:  
<https://cloud.mail.ru/public/5o3T/4G4dD71hL> (дата обращения: февраль, 2019 г.)

## 7 Приложение

### 7.1 main.m

```
1 %%
2 clear all
3 x_lim_left = -0.015;
4 x_lim_right = 0.825; % empirically selected constants for axis scale alignment
5 %% read gfile
6 [flux,RBDry,ZBDry,NBDry,R,Z,time,rDim,zDim]=gfile_extractor_1t(34363,000162,65);
7 %%
8 [arr, ind_arr] = min(flux);
9 [flux_min, min_j] = min(arr);
10 min_i = ind_arr(min_j);
11 %% plot separatrix
12 figure()
13 grid on
14 hold on
15 plot([RBDry, RBDry(1)], [ZBDry, ZBDry(1)], "ks", 'LineWidth', 1.5);
16 magnet_axis = [R(min_j), Z(min_i)];
17 plot(magnet_axis(1), magnet_axis(2), "r*", 'LineWidth', 1.5);
18 xlim([x_lim_left x_lim_right])
19 title("separatrix")
20 legend("separatrix", "magnetic axis")
21 %% calculate radius of curvature separately for the first and last point, taking into
    account the closure of the curve
22 a = [RBDry(NBDry), ZBDry(NBDry)];
23 b = [RBDry(1), ZBDry(1)];
24 c = [RBDry(2), ZBDry(2)];
25 R = find_curv_radius(a, b, c);
26
27 for i = 2 : NBDry - 1
28     a = [RBDry(i - 1), ZBDry(i - 1)];
29     b = [RBDry(i), ZBDry(i)];
30     c = [RBDry(i + 1), ZBDry(i + 1)];
31     tmp = find_curv_radius(a, b, c);
32     R = [R, tmp];
33 end
34
35 a = [RBDry(NBDry - 1), ZBDry(NBDry - 1)];
36 b = [RBDry(NBDry), ZBDry(NBDry)];
37 c = [RBDry(1), ZBDry(1)];
38 tmp = find_curv_radius(a, b, c);
39 R = [R, tmp];
40 %% plot radius of curvature
41 figure()
42 grid on
43 hold on
44 plot(R, "b-", 'LineWidth', 1.5);
45 xlabel('radius');
46 ylabel('angle');
47 title("radius of curvature")
48 %% calculate smoothed radius of curvature
```

```

49 R_sm = medfilt1(R);
50 %% plot radius of curvature
51 figure()
52 grid on
53 hold on
54 plot(R_sm, "b-", 'LineWidth', 1.5);
55 xlabel('radius');
56 ylabel('angle');
57 title("smoothed radius of curvature")
58 %% calculate grid (gossamer)
59 N = 5;
60 init_sector = 1 : NBDY;
61 [s1, s2, s3, s4] = split_sector_simple(init_sector, R_sm);
62 sectors = {s1, s2, s3, s4};
63 result = {};
64 for i = 1 : length(sectors)
65     tmp_sec = sectors{i};
66     tmp = split_sector_N(tmp_sec, N);
67     result = {result{:} tmp{:}};
68 end
69 sectors = result;
70
71 seg = [];
72 middle = magnet_axis;
73 R_seg = [];
74 Z_seg = [];
75
76 for i = 1 : length(sectors)
77     tmp_sec = sectors{i};
78     ind = tmp_sec(1);
79     point = [RBDY(ind), ZBDY(ind)];
80     mid_point = find_center(point, middle);
81     R_seg = [R_seg, mid_point(1)];
82     Z_seg = [Z_seg, mid_point(2)];
83 end
84 %% plot grid
85 figure()
86 grid on
87 hold on
88 plot([RBDY, RBDY(1)], [ZBDY, ZBDY(1)], "ks", 'LineWidth', 1.5);
89 plot(magnet_axis(1), magnet_axis(2), "*r", 'LineWidth', 2);
90 xlim([x_lim_left x_lim_right])
91 title("grid")
92
93
94 for i = 1 : length(sectors)
95     tmp_sector = sectors{i};
96     ind = tmp_sector(1);
97     point = [RBDY(ind), ZBDY(ind)];
98     plot([point(1), middle(1)], [point(2), middle(2)], "b-", 'LineWidth', 1.5);
99 end
100 for i = 1 : length(R_seg) - 1
101     plot([R_seg(i), R_seg(i + 1)], [Z_seg(i), Z_seg(i + 1)], "b-", 'LineWidth', 1.5);
102 end
103 plot([R_seg(length(R_seg)), R_seg(1)], [Z_seg(length(R_seg)), Z_seg(1)], "b-", 'LineWidth', 1.5);
104 plot(magnet_axis(1), magnet_axis(2), "*r", 'LineWidth', 2);

```

## 7.2 find\_curv\_radius.m

```

1 function [R] = find_curv_radius(A, B, C)
2     a = B - A;
3     b = C - B;
4     c = C - A;
5     R = norm(a) / (2 * sqrt(1 - (sum((c) .* (b)) / (norm(c) * norm(b)))^2));
6 end

```

## 7.3 find\_center.m



```

1 function [center] = find_center(inputArg1, inputArg2)
2     center = inputArg1 + (inputArg2 - inputArg1) / 2
3 end

```

## 7.4 split\_sector\_simple.m

```

1 function [s1, s2, s3, s4] = split_sector_simple(sector, R)
2     ind = fix(length(sector) / 2);
3
4     first_sec = sector(1 : ind);
5     second_sec = sector(ind + 1 : length(sector));
6     [~, first_ind] = max(R(first_sec));
7     [~, second_ind] = max(R(second_sec));
8
9     second_ind = second_ind + length(first_sec);
10
11     len_1 = first_ind;
12     len_2 = length(first_sec);
13     len_3 = second_ind;
14     len_4 = length(sector);
15
16     s1 = sector(1 : len_1);
17     s2 = sector(len_1 + 1 : len_2);
18     s3 = sector(len_2 + 1 : len_3);
19     s4 = sector(len_3 + 1 : len_4);
20 end

```

## 7.5 split\_sector\_N.m

```

1 function [result] = split_sector_N(sector, N)
2     result = {};
3     cur_step = ceil(length(sector) / N);
4     cur_left = 1;
5     cur_right = cur_step;
6     reduce = ceil(length(sector));
7     for i = 1 : N
8         if(reduce == 0)
9             return
10        end
11        tmp = sector(cur_left : cur_right);
12        result = {result{:} tmp};
13        reduce = reduce - length(tmp);
14        cur_left = cur_right + 1;
15        cur_step = ceil(reduce / (N - i));
16        cur_right = cur_right + cur_step;
17    end
18 end

```