

Санкт-Петербургский политехнический университет Петра Великого

Институт прикладной математики и механики

Кафедра «Прикладная математика»

**Отчет по дисциплине «Вычислительные комплексы» по
лабораторной работе №3
«Построение разбиения области
между замкнутыми кривыми»**

Выполнил студент группы 3630102/60201

Чепулис М.А.

Преподаватель: Баженов А.Н.

Санкт-Петербург

2019

Оглавление

Постановка задачи	3
Теория	3
Радиус кривизны	3
Построение разбиения сетки	3
Реализация.....	3
Результаты	4
Обсуждение	6
Литература	7
Приложение.....	8
Код программы на Matlab	8

Постановка задачи

Считать данные из g-файла

Вычислить магнитную ось

Построить кривую сепаратрисы, отметить на графике магнитную ось

Вычислить радиусы кривизны в каждой точке сепаратрисы

Применить к полученным результатам медианно сглаживание

Построить разбиение области сепаратрисы на сетку (паутинку)

Теория

Сепаратрисы - граница плазмы, последняя замкнутая поверхность магнитного потока

Магнитная ось – экстремум магнитного потока

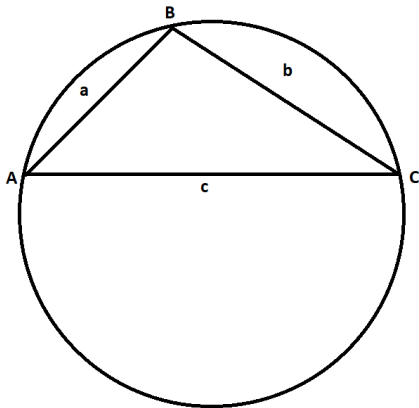
Радиус кривизны

По теореме синусов: $\frac{|a|}{\sin(C)} = 2 * R$

Из скалярного произведения вычислим косинус угла: $\cos(C) = \frac{(b,c)}{|b||c|}$

Тогда $\sin(C) = \sqrt{1 - \cos(C)^2}$

Следовательно $R = \frac{|a|}{2 * \sin(C)}$



Построение разбиения сетки

- Делим область на 2 сектора (по экватору)
- В каждом секторе находим точку с максимальным радиусом кривизны
- Найдены 4 особые точки (2 на экваторе, 2 в точках с максимальным радиусом кривизны)
- Внутри каждого сектора проводим ещё N дополнительных отрезков. Итого получено 4N отрезка
- Находим у каждого из отрезков середину и соединяем их между собой
- В итоге получили разбиение пространства (сетку)

Реализация

Все задания были выполнены на языке программирования Matlab в среде разработки MATLAB R2017b [\[1\]](#)

Данные из фала считаны функцией «gfile_extractor_1t»

[2]

Радиус кривизны вычислялся по 3 точкам (как радиус окружности, описанной вокруг треугольника)

$R(i)$ вычисляется по трём точкам: $A = p(i-1)$, $B = p(i)$, $C = p(i+1)$, где p – точки сепаратрисы

Для крайних точек сепаратрисы учитывается её замкнутость

Результаты

Рассматривается набор данных: 34363

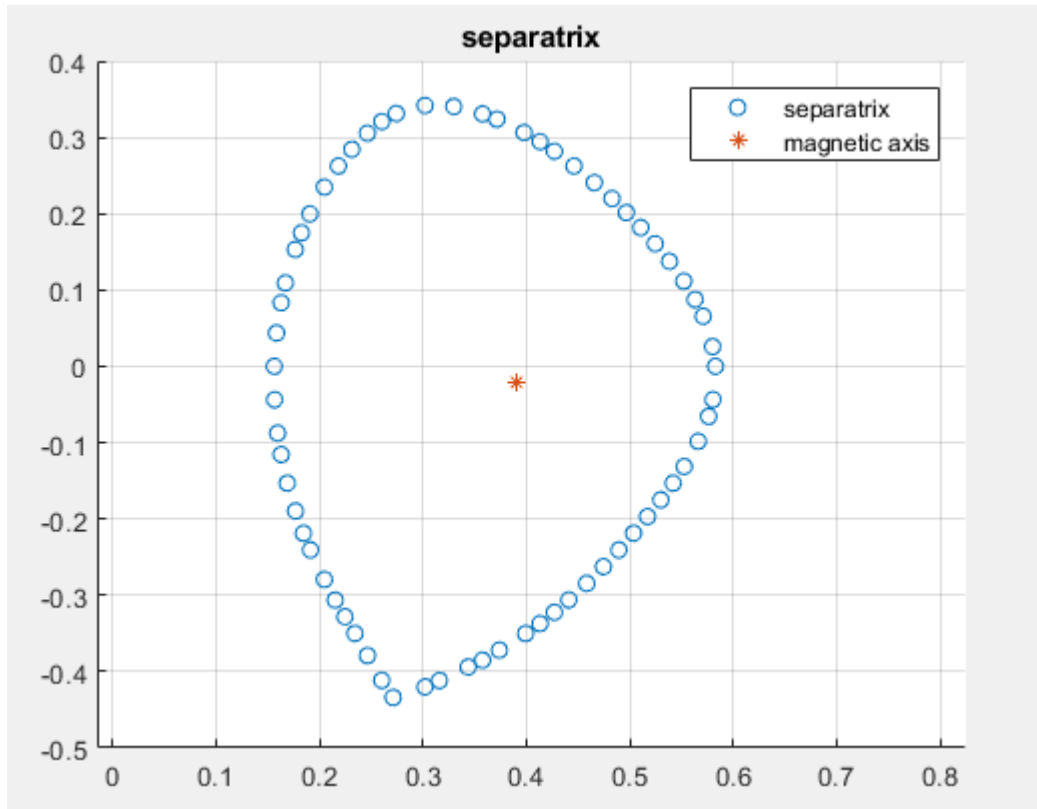


Рисунок 1 график сепаратрисы

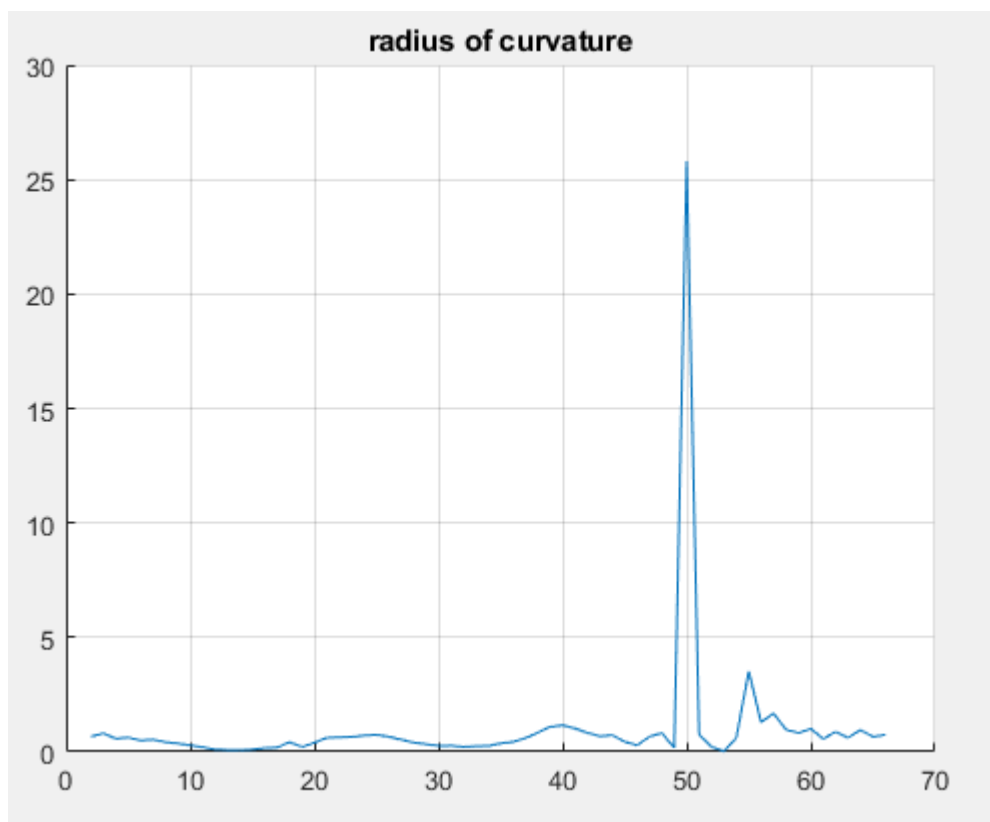


Рисунок 2 радиус кривизны

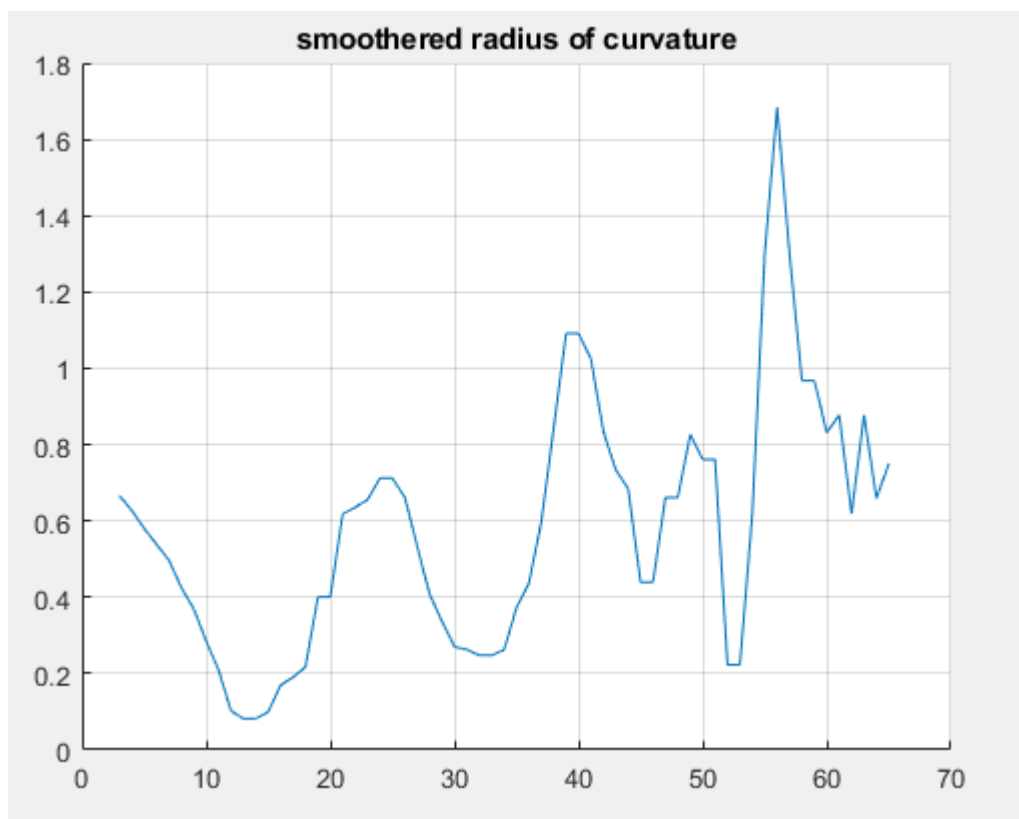


Рисунок 3 радиус кривизны после медианного сглаживания

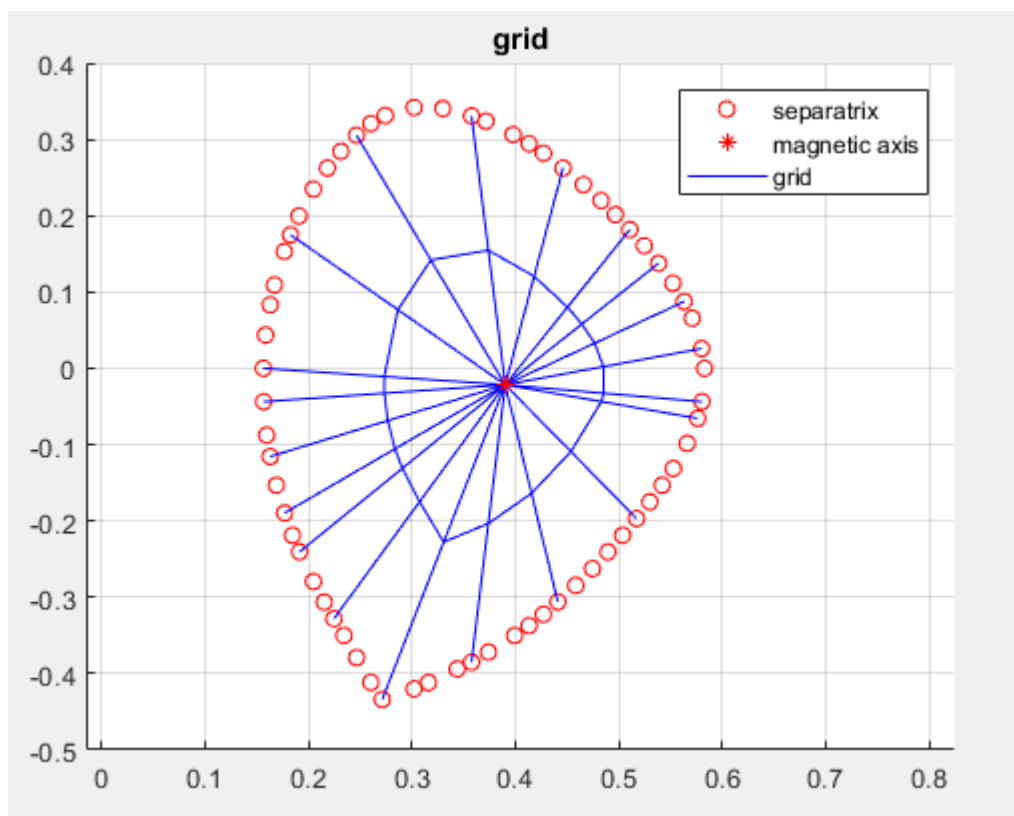


Рисунок 4 разбиение сетки для сепаратрисы

Обсуждение

После медианного сглаживания «пики» и «впадины» радиуса кривизны стали более отчетливые

Было получено разбиение пространства (сетка)

Литература

- [1] Документация по Матлаб [Электронный ресурс]
Режим доступа: <https://www.mathworks.com/help/> (дата обращения октябрь 2019)
- [2] Код функции g_file_extractor_1t [электронный ресурс, облачное хранилище]
Режим доступа: <https://cloud.mail.ru/public/5o3T/4G4dD71hL> (дата обращения сентябрь 2019)
- [3] Пособие к Лабораторным работам [электронный ресурс, облачное хранилище]
Режим доступа: <https://cloud.mail.ru/public/4ra6/5wwqBzMBC/LabPractics.pdf> (дата обращения октябрь 2019)

Приложение

Код программы на Matlab

Lab_3_separatrix

```
%%
clear all
%%
[flux,RBDY,ZBDY,NBDY,R,Z,time,rdim,zdim] = gfile_extractor_1t(34363, 000162, 65);
%%
[arr, ind_arr] = min(flux);
[flux_min, min_j] = min (arr);
min_i = ind_arr(min_j);

figure()
grid on
hold on
plot([RBDY, RBDY(1)], [ZBDY, ZBDY(1)]);
magnet_axis = [R(min_j), Z(min_i)];
plot(magnet_axis(1), magnet_axis(2), "o");
title("separatrix")
legend("separatrix", "magnetic axis")

%%
%вычисляем отдельно для первой и последней точки, учитывая замкнутость кривой
a = [RBDY(NBDY), ZBDY(NBDY)];
b = [RBDY(1), ZBDY(1)];
c = [RBDY(2), ZBDY(2)];
r = get_curv_radius(a, b, c);
for i=2:NBDY - 1
    a = [RBDY(i-1), ZBDY(i-1)];
    b = [RBDY(i), ZBDY(i)];
    c = [RBDY(i + 1), ZBDY(i + 1)];
    tmp_r = get_curv_radius(a, b, c);
    r = [r, tmp_r];
end

a = [RBDY(NBDY-1), ZBDY(NBDY-1)];
b = [RBDY(NBDY), ZBDY(NBDY)];
c = [RBDY(1), ZBDY(1)];
tmp_r = get_curv_radius(a, b, c);
r = [r, tmp_r];
figure()
grid on
hold on
plot(r);
title("radius of curvature")

%%

r_smoothed = medfilt1(r);
figure()
grid on
hold on
plot(r_smoothed);
title("smoothered radius of curvature")
%%
% sector = 1:NBDY;
% sectors = {sector};
% N = 2;
% while length(sectors) < 4 * N
%     result = {};
%     for i = 1:length(sectors)
%         tmp_sector = sectors{i};
```



```

%         if( length(tmp_sector) <= 4)
%             result = {result{:} tmp_sector};
%         else
%
%             [s1, s2, s3, s4] = split_sector(tmp_sector, r);
%             result = {result{:} s1 s2 s3 s4};
%         end
%     end
%     sectors = result
% end

N = 5;
init_sector = 1:NBDY;
[s1, s2, s3, s4] = split_sector_easy(init_sector, r_smoothed);
sectors = {s1, s2, s3, s4};
result = {};
for i = 1:length(sectors)
    tmp_sector = sectors{i};
    tmp = split_linspace(tmp_sector, N);
    result = {result{:} tmp{:}};
end
sectors = result;

segments = [];
middle = magnet_axis;
R_segments = [];
Z_segments = [];

% src_segments = []
% for i = 1:length(sectors)
%     tmp_sector = sectors{i};
%     cur_ind = tmp_sector(1);
%     tmp_segment = [R_segments, mid_point(1)]
%                 [Z_segments, mid_point(2)]
%
%     cur_point = [RBDY(cur_ind), ZBDY(cur_ind)];
%     plot([cur_point(1), middle(1)], [cur_point(2), middle(2)], 'b')
%
% end
for i = 1:length(sectors)
    tmp_sector = sectors{i};
    cur_ind = tmp_sector(1);
    cur_point = [RBDY(cur_ind), ZBDY(cur_ind)];
    mid_point = get_center(cur_point, middle);
    R_segments = [R_segments, mid_point(1)];
    Z_segments = [Z_segments, mid_point(2)];
end

figure()
grid on
hold on
plot([RBDY, RBDY(1)], [ZBDY, ZBDY(1)], "r");
plot(magnet_axis(1), magnet_axis(2), "or");
title("grid")

for i = 1:length(sectors)
    tmp_sector = sectors{i};
    cur_ind = tmp_sector(1);
    cur_point = [RBDY(cur_ind), ZBDY(cur_ind)];
    plot([cur_point(1), middle(1)], [cur_point(2), middle(2)], 'b')
end

```

```

for i = 1:length(R_segments)-1
    plot([R_segments(i), R_segments(i + 1)], [Z_segments(i), Z_segments(i + 1)], 'b')
end
plot([R_segments(length(R_segments)), R_segments(1)], [Z_segments(length(R_segments)), Z_segments(1)], 'b')
plot(magnet_axis(1), magnet_axis(2), "or");
legend("separatrix", "magnetic axis", "grid")

%%

```

Get_curv_radius

```

function r = get_curv_radius(A, B, C)
    a = B - A;
    b = C - B;
    c = C - A;
    cos = sum(c.*b) / (norm(c)*norm(b));
    sin = sqrt(1 - cos^2);
    r = norm(a) / (2 * sin);
end

```

split_sector

```

function [s1, s2, s3, s4] = split_sector(sector, r)
    separator_ind = fix(length(sector) / 2); % делим нацело на 2 части

    first_sector = sector(1:separator_ind);
    second_sector = sector(separator_ind + 1:length(sector));
    [min_val , first_min_ind] = min(r(first_sector));
    [max_val , first_max_ind] = max(r(first_sector));

    [min_val , second_min_ind] = min(r(second_sector));
    [max_val , second_max_ind] = max(r(second_sector));
    second_min_ind = second_min_ind + length(first_sector); % т.к. индекс получаем
относительно массива r(second_sector) (36 элементов), а хотим иметь абсолютный
    second_max_ind = second_max_ind + length(first_sector);

    sep_1 = min(first_min_ind, first_max_ind);
    sep_2 = max(first_min_ind, first_max_ind);
    sep_3 = min(second_min_ind, second_max_ind);
    sep_4 = max(second_min_ind, second_max_ind);
    if(sep_4 == length(sector))
        s1 = sector(1 : sep_1);
    else
        s1 = sector([sep_4 + 1: length(sector), 1 : sep_1]);
    end
    s2 = sector(sep_1 + 1 : sep_2);
    s3 = sector(sep_2 + 1 : sep_3);
    s4 = sector(sep_3 + 1: sep_4);
end

```

split_sector_easy

```

function [s1, s2, s3, s4] = split_sector_easy(sector, r)
    separator_ind = fix(length(sector) / 2); % делим нацело на 2 части

    first_sector = sector(1:separator_ind);
    second_sector = sector(separator_ind + 1:length(sector));
    [max_val , first_max_ind] = max(r(first_sector));
    [max_val , second_max_ind] = max(r(second_sector));

```

```

second_max_ind = second_max_ind + length(first_sector); % т.к. индекс получаем
относительно массива r(second_sector) (36 элементов), а хотим иметь абсолютный

sep_1 = first_max_ind;
sep_2 = length(first_sector);
sep_3 = second_max_ind;
sep_4 = length(sector);

s1 = sector(1 : sep_1);
s2 = sector(sep_1 + 1 : sep_2);
s3 = sector(sep_2 + 1 : sep_3);
s4 = sector(sep_3 + 1 : sep_4);
end

```

split_linspace

```

function [result] = split_linspace(sector, N)
    result = {};
    cur_step = ceil(length(sector) / N);
    cur_left = 1;
    cur_right = cur_step;
    reduce = ceil(length(sector));
    for i = 1:N
        if(reduce == 0)
            return
        end
        tmp = sector(cur_left:cur_right);
        result = {result{:} tmp};
        reduce = reduce - length(tmp);
        cur_left = cur_right + 1;
        cur_step = ceil(reduce / (N - i));
        cur_right = cur_right + cur_step;
    end
end

```

get_center

```

function [result] = split_linspace(A, B)
    result = A + (B - A) / 2;
end

```