```
%% A flow of program for calculating temperature distribution.
%% A part is omitted, and some subfunctions are needed for working
%% this program. %% Thanks to Mr.Mizuki Kyoda.
%% Matlab@Mathworks
fileID = fopen('radial-profile.txt','r');
C = textscan(fileID,'%f','CommentStyle','frame');
B = reshape(C{1,1},3,[]);
[m,n] = size(B);
B = horzcat(B,[0;0;0]);
p_l = 0.030303030303030; % length(mm)/pixel
R = 3.5; % radius(mm)
width = 252; % image width(pixel)
center = width/2; % center of image
temp_distribution = zeros(width);
Component = 4; % component number
pos_ini = 1;
frame_num = 1;

for pos = 1:n
        if B(2,pos) > B(2,pos+1)
                frame(frame_num,:) = B(3,pos_ini:pos);
                frame_num = frame_num+1;
                pos_ini = pos+1;
        end
end
total_fr = frame_num;

p = B(2,n);
zero = zeros(B(2,1)-1,frame_num-1);
frame2 = vertcat(zero,frame.');
limit = 1000;
random = 0.0100;
for frame_num = 1:total_fr-1
    x = zeros(p,1);
    for i=1:p
        x(i,1) = (i-1) * p_l;
    end
    Data1 = zeros(p,2);
    Data1(:,1) = x(:,1);
    Data1(:,2) = frame2(:,frame_num);
    Data1(1:B(2,1)-1,:) = [];
    count = 0;
    count_end = 10;
    y = Data1(:,2);
    t3 = Data1(:,1);

  for k = 1:limit
    fprintf('Frame%dL EM Iteration %d¥n',frame_num,k)
    lam0 = random*randsample(1:999,Component);
    options = optimset('LargeScale','off','Display','off','TolX',1e-16);
    t2 = Data1(:,1);
    y2 = Data1(:,2);
    plothandle = plotdatapoints(t2,y2);
    F = @(x) fit1(x,Data1,R);
    foutputfcn = @(x,optimvalues,state) fitout(x,optimvalues,state, Data1,plothandle);
    options = optimset(options,'OutputFcn',foutputfcn);
    options = optimset(options,'Algorithm','levenberg-marquardt');
    try
    [lam,resnorm,residual,exitflag,output] = lsqnonlin(F,lam0,[],[],options);

    axis([0 3.59 -0.0005 0.008]);

    A2 = zeros(length(t2),length(lam));
    for j = 1:length(lam)
        A2(:,j) = exp(-t2.*t2/lam(j))-exp(-((R+0)^2)/lam(j));
    end
    cc2 = A2¥y2;
    A3 = zeros(length(t3),1);
    for i = 1:length(lam)
        coefficient = cc2(i)*(exp(-(t3+0).*(t3+0)/lam(i))-exp(-((R+0)^2)/lam(i)));
        A3 = A3+coefficient;
    end
    C = (A3-y).^2;
    RM = mean(C);
    RMS = sqrt(RM);
    catch ME1
        idSegLast = regexp(ME1.identifier, '(_<=:)¥w+$', 'match');
    lam = NaN(Component,1);
```

```
            cc2 = NaN(Component,1);
            end

            B1 = lam>0.15;
            B2 = all(B1);
            B3 = all(cc2);
            if (exitflag >0 && B2>0 && B3>0 && count<count_end)
                count = count+1;
                Multi_result.lam(count,:) = lam;
                Multi_result.cc2(count,:) = cc2;
                Multi_result.RM(count,1) = RM;
                A(:,count) = A3;
            elseif exitflag <= 0
            else
            end
            if count == count_end
                        break
            end
        end

        n = length(y);
        Maximum_log_lilehood = (-n/2)*(log(2*pi)+log(Multi_result.RM)+1);
        AIC = (-2*Maximum_log_lilehood)+(2*(1+2*Component+1));
        for zc = 1:count_end
            if AIC(zc,1) == -Inf
                AIC(zc,1) = 1;
            end
        end
        number = find(AIC == min(AIC), 1);

        result.lam(frame_num,:) = Multi_result.lam(number,:);
        result.cc2(frame_num,:) = Multi_result.cc2(number,:);
        result_A = A(:,number);
        plot(Data1(:,1),result_A,'b',Data1(:,1),Data1(:,2),'r.','LineWidth',2,'MarkerSize',6);
        grid on
        axis([0 3.59 -0.0005 0.008]);
        legend('Approx. curve','Measurement')
        set(gca,'FontSize',20); set(legend,'Box','off'); set(legend,'FontSize',20);
        xlabel('x'); ylabel('ΔA(x)');
        filename = ['Approx./',num2str(Component),'Term num./fitted_curve./',num2str(frame_num),'frame','.bmp'];
        saveas(gcf,filename,'bmp')
        lam = result.lam(frame_num,:);
        cc2 = result.cc2(frame_num,:);
        for i = 1:width
            for j = 1:width
                x = (j-center)/center;
                y = (center-i)/center;
                r = sqrt(x*x+y*y);
                if r<=1;
                    temp_distribution(i,j) = test_func(r,cc2,lam,Component);
                end
            end
        end
        figure('Position', [10 10 400 400])
        contour(temp_distribution)
        grid on
        filename2 = sprintf('%dframe',frame_num);
        Imgfilename = strcat('Approx./',num2str(Component),'Term num./temp_contour./',filename2,'.bmp');
        f = getframe(gcf);
        X = frame2im(f);
        imwrite(X,Imgfilename,'bmp')
        imshow(temp_distribution,'Border','tight')
        a = max(max(temp_distribution));
        b = min(min(temp_distribution));
        map = jet(256); colormap(map)
        cmax = 0.004; % max temp. range
        cmin = 0; % min temp. range
        caxis([cmin cmax]); brighten(0.15); % image brightness
        filename3 = sprintf('%dframe',frame_num);
        Imgfilename    = strcat('Approx./',num2str(Component),'Term num./temp_section./',filename3,'.tif');
        f = getframe(gcf);
        X = frame2im(f);
        imwrite(X,Imgfilename,'tif')
        close all
    end
    save(sprintf('result.mat'))
```