```
%% DBF Foamcutter for Genearl Shapes
% This code is written by Yuting Huang (ythuang96@gmail.com);
% Please report all bug to the author's email address.
% Last updated: 8/26/2018

% This is written for DBF foamcutting, to generate G-code from general shape
% AutoCAD drawings.

%% User Manual
% 1. Export lines and arcs form AutoCAD, save as csv file.
% 2. Copy the csv file to the same folder as this MatLab code.
% 3. Run Code and done!
% Press CRT+C at anytime to terminate code.

%% ------------------------------------------------------------------------
%% ------------------------------------------------------------------------
%% ------------------------------------------------------------------------
%% ------------------------------------------------------------------------
clear all; close all; clc;
tolerance = 0.0002;
accuracy = 2; % length in mm of segments when breaking arc
%% Determine Units
% GUI stuff
UIControl_FontSize_bak = get(0, 'DefaultUIControlFontSize');
set(0, 'DefaultUIControlFontSize', 30);
unit = menu('Is Drawing in millimeters?','Yes','No');
if unit == 1; % if drawing is in mm, continue generation of G-code
    %% Check the current folder for csv files
    D = dir('*.csv');
    if ~length(D) % if no .csv file exsist print error message
        fprintf('I could not find any file with .csv' );
        fprintf(' estension in the current folder.\n' );
        fprintf('Please move the .csv file created by AutoCAD ' );
        fprintf('''eattext'' command into the current ' );
        fprintf('working folder and try again.\n' );
    else
        % Create a menu to select csv files in the current folder
        string = ['file = menu(''I detected ' num2str(length(D)) ...
            ' csv files list below, please select one'',' ];
        for i = 1:length(D); string = [string '''' D(i).name ''',']; end
        string = [string '''None of the above'');'];
        eval(string);
        if file <= length(D);
            filename = D(file).name(1:end-4); clear  string;
            if file <= length(D) && file;
                %%  Inport File
                inport = csvread([filename '.csv'],1,2);
                [m,n] = size(inport);
                % make changes if there are only lines
                if n ==4; inport = [zeros(m,5) , inport];  end
                %% Eliminate 0 length lines
                k = 1;
                for i = 1:size(inport,1);
                    if any(inport(i,6:9) ~= [0 0 0 0]) && ...
            all(inport(i,6:7) == inport(i,8:9));
        m = m-1;
    else temp(k,:) = inport(i,:); k = k+1;
    end
end
inport = temp; clear temp k i D file;
%% Seperate Arc With line
n_arc = 0; n_line = 0;
for i = 1:m;
    if all(inport(i,6:9) == [0 0 0 0]);
        n_arc = n_arc + 1; arc(n_arc,:) = inport(i,1:5);
    else n_line = n_line + 1;
        line(n_line,:) = inport(i,6:9);
    end
end
%% Break Arcs into lines
alllines = line;
for i = 1:n_arc;
    n_segment = ceil(2*pi*arc(i,3)*arc(i,5)/360/accuracy);
    dtheta = arc(i,5)/n_segment;
    arcpoints = zeros(n_segment+1,2);
    for j = 1:n_segment+1;   % break arc into points
        theta = arc(i,4) + (j-1)*dtheta;
        arcpoints(j,:) = arc(i,1:2) +  ...
            arc(i,3).*[cosd(theta),sind(theta)];
    end
    % chage the start and end point so that the arc join
    % the lines
    for j = 1:n_line;
        if abs(arcpoints(1,:) - line(j,1:2)) <= 0.01;
            arcpoints(1,:) = line(j,1:2);
        elseif abs(arcpoints(1,:) - line(j,3:4)) <= 0.01;
            arcpoints(1,:) = line(j,3:4);
        end
        if abs(arcpoints(end,:) - line(j,1:2)) <= 0.01;
            arcpoints(end,:) = line(j,1:2);
        elseif abs(arcpoints(end,:) - line(j,3:4)) <= 0.01;
            arcpoints(end,:) = line(j,3:4);
        end
    end
    % put all lines with arc points together
    alllines = [alllines ; arcpoints(1:end-1,:) ,  ...
        arcpoints(2:end,:)];
end
%% Sort the lines in order
sort(1,:) = alllines(1,:);
alllines(1,:) = [];
for i = 2:size(alllines,1)+1;
    compare = sort(i-1,3:4);
    [n2,~] = size(alllines);
    for j = 1:n2;
        if all(abs(compare - alllines(j,1:2)) <= tolerance);
            sort(i,:) = alllines(j,:);
            alllines(j,:) = []; check = 1;  break;
        elseif all(abs(compare - alllines(j,3:4)) <= tolerance);
            sort(i,1:2) = alllines(j,3:4);
            sort(i,3:4) = alllines(j,1:2);
            alllines(j,:) = []; check = 1;  break;
        end
    end
    if ~check
        % cannot find the another line that connects with
        % the previous
        fprintf('There is an open countour.\n');
        fprintf('This is most likely caused by an ');
        fprintf('extra line underneath a long line.\n');
        fprintf('Please check your drawing.\n');
        return;
    end
    check = 0;
end
sort2 = [sort(:,1:2); sort(end,3:4)];
%% Shift to positive
min_x = min(sort2(:,1)); min_y = min(sort2(:,2));
sort2(:,1) = sort2(:,1) - min_x;
sort2(:,2) = sort2(:,2) - min_y;
max_x = max(sort2(:,1)); max_y = max(sort2(:,2));
%% Plot Curve
figure(1); set(1,'position',[0 0 1920 1080]); hold on;
plotx = sort2(:,1); ploty = sort2(:,2); plot(plotx,ploty );
title('Drawing Unit mm','fontsize',30);
axis equal;
%% Plot number
[n,~] = size(sort2);
j = 1; index = [];
for i = 1:n-1;
    if sort2(i,1) == 0 || sort2(i,1) == max_x  ...
        || sort2(i,2) == 0 || sort2(i,2) == max_y;
        text(plotx(i),ploty(i),sprintf( '%d',j),'fontsize',20);
        j = j+1; index = [index, i];
    end
end
hold off;
%% Determine Start Point
start = index(input('Which point would you like to start?    '));
sort3 = [sort2(start:end-1,:) ; sort2(1:start-1,:);  ...
    sort2(start,:)];
%% Cut Direction Reverse if chosen to
direction = menu(['The Current Cut Direction is shown in' , ...
    'the Figure with Increasing Number,' , ...
    'Reverse Cut Direction?'],'NO','YES']);
if direction == 2; final = rot90(sort3',1);
elseif direction == 1; final = sort3; end
%% Final Plot
clf; hold on;
finalx = final(:,1); finaly = final(:,2);
plot(finalx,finaly);
title('Final shape on Foam Cutter, Drawing Unit mm' , ...
                    'fontsize',30);
                j = 1;
                for i = 1:n
                    if final(i,1) == 0 || final(i,1) == max_x  ...
                        || final(i,2) == 0 || final(i,2) == max_y;
                        text(finalx(i),finaly(i),sprintf( '%d',j),'fontsize',20);
                        j = j+1;
                        if j == 4; break; end
                    end
                end
                axis equal; hold off;
                set(0, 'DefaultUIControlFontSize' , UIControl_FontSize_bak);
                %% Generate G-code
                fidw = fopen([filename  '.txt'],'wt');
                fprintf(fidw,'G21\n'); fprintf(fidw,'M49\n');
                fprintf(fidw,'F80\n'); fprintf(fidw,'S80\n');
                fprintf(fidw,'G1  X % 8.3f  Y % 8.3f  Z % 8.3f  A % 8.3f\n' ...
                    ,0,0,0,0);
                fprintf(fidw,'G1  X % 8.3f  Y % 8.3f  Z % 8.3f  A % 8.3f\n' ,...
                    finalx(1),finaly(1),finalx(1),finaly(1));
                for i = 2:length(finalx)-1
                    fprintf(fidw,'G1  X % 8.3f  Y % 8.3f  Z % 8.3f  A % 8.3f\n' ...
                        ,finalx(i),finaly(i),finalx(i),finaly(i));

                    % Calculate Length
                    length = sqrt((finalx(i)-finalx(i-1))^2 +  ...
                    (finaly(i)-finaly(i-1))^2);
                    if length >= 100;
                        fprintf(fidw,sprintf( 'G4 P%d\n',floor(length/100) ) );
                    end
                    % Add 1 sec pause per 100 mm cut for long cuts
                end
                fprintf(fidw,'G1  X % 8.3f  Y % 8.3f  Z % 8.3f  A % 8.3f\n' ...
                    ,finalx(end),finaly(end),finalx(end),finaly(end));
                fprintf(fidw,'G1  X % 8.3f  Y % 8.3f  Z % 8.3f  A % 8.3f\n' ...
                    ,0,0,0,0);
                fprintf(fidw,'M2');
                fclose(fidw);
                disp(['The G-Code is saved as  ''' filename ...
                    '.txt''  in this folder. ']);
                set(gcf,'PaperUnits','inches','PaperPosition',[0 0 16 9]);
                print(filename,'-dpng','-r240');
            else % if 'none of the above selectee'
                fprintf('Please move your desired file to the current' );
                fprintf(' folder and try again.\n');
            end
        else % if no file selected
            fprintf('Please move your desired file to the current' );
            fprintf(' folder and try again.\n');
        end % end file selection check
    end  % end file exsistence check
else % if drawing not in mm, print error message
    fprintf('Please go back to AutoCAD and use the ''Scale''' );
    fprintf(' command to scale the drawing by 25.4. \n' );
    fprintf('Inches do not provide high enough accuracy\n' );
end % end 'unit' check
```