# Output Function

Using the vectors of the other functions of the library as input parameters, this function, suitably called by the others, has two fundamental purposes:

1. generate a plot (or more plots) relating to the specific operating curves of propellers, rotors and turbines;
2. generate a text file with all output vectors in column.

## syntax

The inputs of the function are the function identifier as a string and vectors that need to be plotted or transcribed into a text file.

```
[y1,y2] = FunzionidiOutput(txt,v1,v2)
```

Where:

| INPUT | |
|---|---|
| txt | function identifier |
| v1 | x-axis vector |
| v2 | y-axis vector |

| OUTPUT | |
|---|---|
| yi | figure(i) |

## example

A test case for the function is shown below.
It is called the function identifier 'ClCd Xrotor' whose purpose is to output the polar of the airfoil according to the software Xrotor.

```
clc; close all; clear all;
function [y1,y2] = FunzionidiOutput('CdCl_xfoil',v1,v2)
```

An example of a function with an output graph is below: it is underlined that the graph is right as an example but there is no reference to the real polar.

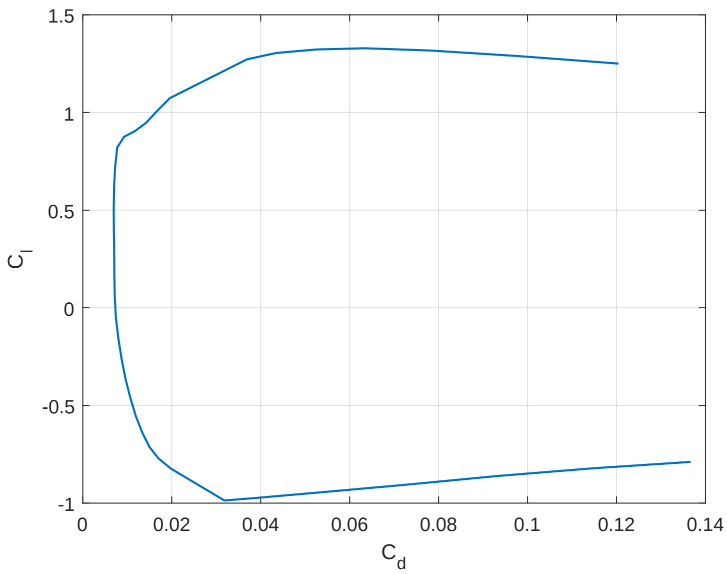


**Figure 1:** Example of output graph

## code listing

```
function [y1,y2,y3] = Outputfunction(txt,v1,v2,v3,v4,v5,v6)

%% Default case
% function y = FunzionidiOutput(input1,input2,axisname)
% [rows,columns]=size(input1);
% for i=1:rows
%     xaxisname=string(axisname(1,i));
%     yaxisname=string(axisname(2,i));
%     y(i)=figure(i);
%     plot(x(i,:),y(i,:),'-k');
%     grid on;
%     xlabel(xaxisname);
%     ylabel(yaxisname);
% end
% end

%% Specifics cases
function_name=txt;
switch function_name
    case 'ClCd_XRotor'
    y1=figure(1)
    plot(v1,v2,'linewidth',1.1);
    grid on;
    xlabel('C_d');
    ylabel('C_l');
    v3==v4==v5==v6==[];
    y2=[];
    y3=[];
    case 'Characteristics_Curve_HO_Windmill'
    y1=figure(1)
    plot(v1,v2,'linewidth',1.1);
    grid on;
    xlabel('\lambda')
    ylabel('C_T')
    y2=figure(2)
    plot(v3,v4,'linewidth',1.1);
    grid on;
    xlabel('\lambda')
    ylabel('C_Q')
    y3=figure(3)
    plot(v5,v6,'linewidth',1.1);
    grid on;
    xlabel('\lambda')
    ylabel('C_P')
    case 'Axial_Descent_Ascent'
    y1=figure(1)
    plot(v1,v2,'linewidth',1.1);
    grid on;
    xlabel('$\widetilde{V}$','Interpreter','latex','FontSize',15);
    ylabel('$\widetilde{w}$','Interpreter','latex','FontSize',15);
    y2=figure(2)
    plot(v3,v4,'linewidth',1.1);
    grid on;
    xlabel('$\widetilde{V}$','Interpreter','latex','FontSize',15);
    ylabel('$\widetilde{P}$','Interpreter','latex','FontSize',15);
    v5==v6==[];
    y3=[];
    case 'RVortexInt'
    y1=figure(1)
    plot(v1,v2);
    grid on;
    axis ([0 1 0 1]);
    text(0.25,1,'Velocity induced by vortex ring:');
    text(0.25,0.90,['fx= ',num2str(fx)]);
    v3==v4==v5==v6==[];
    y2=[];
    y3=[];
    case 'CdCl_xfoil'
    y1=figure(1)
    plot(v1,v2);
```

```
    xlabel('Drag coefficient C_d');
    ylabel('Lift coefficient C_l');
    grid on;
    v3==v4==v5==v6==[];
    y2=[];
    y3=[];
    case 'RotorFF' %subplot
    case 'Opti_prop_P'
    y1 = ['Data_Opti_Prop_P.txt'];
    fid = fopen(y1, 'wt');
    fprintf(fid, '%s\t%s','  efficiency =',eta,' at J =', J);  % header
    fprintf(fid,'\n');
    fprintf(fid, '%s\t%s','  w_conv =',w);  % header
    fprintf(fid,'\n');
    fprintf(fid, '%s\t%s\t%s\t%s\t%s\t%s\n', '  r_adim','  chi','      a(chi)
    ','  a''(chi)','  dCt/dr_adim',' dCp/dr_adim');  % header
    fclose(fid);
    dlmwrite(y1,DATA,'delimiter','\t','precision',['%10.',num2str(6),'f'],'-
    append');
    v3==v4==v5==v6==[];
    y2=[];
    y3=[];
    case 'Opti_prop_T'
    y1 = ['Data_Opti_Prop_T.txt'];
    fid = fopen(y1, 'wt');
    fprintf(fid, '%s\t%s','  efficiency =',eta,' at J =', J);  % header
    fprintf(fid,'\n');
    fprintf(fid, '%s\t%s','  w_conv =',w);  % header
    fprintf(fid,'\n');
    fprintf(fid, '%s\t%s\t%s\t%s\t%s\t%s\n', '  r_adim','  chi','      a(chi)
    ','  a''(chi)','  dCt/dr_adim',' dCp/dr_adim');  % header
    fclose(fid);
    dlmwrite(y1,DATA,'delimiter','\t','precision',['%10.',num2str(6),'f'],'-
    append');
    v3==v4==v5==v6==[];
    y2=[];
    y3=[];
end
end
```