```matlab
%--------------------------------------------------------------------
%%%% "BET.m" FUNCTION
%%%% IMPLEMENTED BY SERGIO CUSTODIO - engsergiocustodio@gmail.com
%%%% CONTRIBUTIONS OF PROF. DR. JERSON R. P. VAZ
%--------------------------------------------------------------------

function [POWER,CE,CP,W,AOA,FN,FT,MT]=BET(B,bladeshape,polar,omega,v0,rho,reynolds,ITRMAX,TOL,AOAs);
blade_shape = load(bladeshape);

r = blade_shape(:,1);          %radial position (m)
c = blade_shape(:,2);          %chord (m)
b = blade_shape(:,3);          %twist (degree)
N = length(r);                 %NUMBER OF SECTIONS ALONG THE BLADE

FN=zeros(N,1);
FT=zeros(N,1);
MT=zeros(N,1);
AOA=zeros(N,1);
W=zeros(N,1);
CE=zeros(N,1);
CP=zeros(N,1);
DQ=zeros(N,1);

a0  = 1.0/3;  %INITIAL VALUE FOR THE AXIAL INDUCTION FACTOR
a1  = 0.001;   %INITIAL VALUE FOR THE TANGENTIAL INDUCTION FACTOR
ac  = 0.2;    %APLICATE GLAUERT CORRECTION

for j = N:-1:1
    error = 1;   %INITIAL VALUE FOR THE error
    sigma = c(j)*B/(2*pi*r(j));%solidez
    ITR=1;
    while (error>TOL) && (ITR < ITRMAX)
        phi = atan((1-a0)*v0/((1+a1)*omega*r(j)));
        phid=phi*180/pi;
        alpha = (phid-b(j)); %degree
        W(j) = sqrt(((1-a0)*v0)^2 + ((1+a1)*omega*r(j))^2);

        %APLICATE VITERAN AND CORRIGAN APPROXIMATION(degree)
        if  alpha > AOAs
            [cls,cds,cm] = aerodinamic(polar,AOAs,reynolds);

            muh = (r(end)-r(1))/c(j);
            [cl,cd] = CL_CD_CORRECTION(alpha*pi/180,AOAs*pi/180,cls,cds,muh);
        elseif alpha < -AOAs
            [cls,cds,cm] = aerodinamic(polar,-AOAs,reynolds);
            muh = (r(end)-r(1))/c(j);
            [cl,cd] = CL_CD_CORRECTION(alpha*pi/180,-AOAs*pi/180,cls,cds,muh);

        else
            [cl,cd,cm] = aerodinamic(polar,alpha,reynolds);

        end

        CN = cl*cos(phi)+cd*sin(phi);
        CT = cl*sin(phi)-cd*cos(phi);

        %PRANDTL CORRECTION
        F1 = PRANDTL_FACTOR(B,r(end),r(j),phi);
        F2 = PRANDTL_FACTOR_hub(B,r(1),r(j),phi);
```

```


        F=F1*F2;
        a0 = 1/((4*F*(sin(phi))^2)/(sigma*CN)+1);


        %GLAUERT'S CORRECTION FOR THE AXIAL INDUCTION FACTOR
        if a0 > ac
            K = 4*F*(sin(phi)^2)/(sigma*CN);
            a0 = real(0.5*(2+K*(1-2*ac)-sqrt((K*(1-2*ac)+2)^2+4*(K*(ac^2)-1))));
        end

        a1 = 0.5*(-1.+sqrt(1.+4.*a0*(1.-a0)/(omega*r(j)/v0)^2.));

        phin = atan((1-a0)*v0/((1+a1)*omega*r(j)));
        error=abs((phi-phin)/phin);

        ITR=ITR+1;
    end

    W(j) = sqrt(((1-a0)*v0)^2 + ((1+a1)*omega*r(j))^2);
    AOA(j)=alpha;
    CN = cl*cos(phin)+cd*sin(phin);
    CT = cl*sin(phin)-cd*cos(phin);


    FN(j)=0.5*rho*(W(j)^2)*c(j)*CN;
    FT(j)=0.5*rho*(W(j)^2)*c(j)*CT;
    MT(j)=0.5*rho*(W(j)^2)*c(j)*cm;
    DQ(j)=FT(j)*r(j);

end

THRUST=B*trapz(r,FN);
CE=THRUST/(0.5*rho*(v0^2)*pi*(r(end)^2));
TORQUE=B*trapz(r,DQ);
CQ=TORQUE/(0.5*rho*(v0^2)*pi*(r(end)^3));
POWER=omega*TORQUE;
CP=POWER/(0.5*rho*(v0^3)*pi*(r(end)^2));

%---------------------------------------------------------------------
%%%%%%SUBFUNCTIONS
%---------------------------------------------------------------------
function [CL,CD] = CL_CD_CORRECTION(alf,alfs,CLs,CDs,muh)

if muh <= 50
    CDmax = 1.11 + 0.018*muh;
else
    CDmax = 2.01;
end

Kl = (CLs-CDmax*sin(alfs)*cos(alfs))*sin(alfs)/cos(alfs)^2;
Kd = (CDs-CDmax*sin(alfs)^2)/cos(alfs);

CL = 0.5*CDmax*sin(2*alf) + Kl*cos(alf)^2/sin(alf);
CD = CDmax*sin(alf)^2 + Kd*cos(alf);

%---------------------------------------------------------------------
function [F] = PRANDTL_FACTOR(B,R,Ri,PHI)
if R == Ri
```

```
    f = B*(R-0.99*Ri)/(2.0*Ri*sin(PHI));
    F = 2.0*acos(exp(-f))/pi;
else
    f = B*(R-Ri)/(2.0*Ri*sin(PHI));
    F = 2.0*acos(exp(-f))/pi;
end

%---------------------------------------------------------------------
function [F] = PRANDTL_FACTOR_hub(B,R,Ri,PHI)
if R == Ri
    f = B*(Ri-0.9*R)/(2.0*R*sin(PHI));
    F = 2.0*acos(exp(-f))/pi;
else
    f = B*(Ri-R)/(2.0*R*sin(PHI));
    F = 2.0*acos(exp(-f))/pi;
end

%---------------------------------------------------------------------
function [cl,cd,cm] = aerodinamic(Polar,alpha,reynolds)
fid = fopen(Polar,'r');
formato = str2num(fgetl(fid));
polar=zeros(formato(2),1+3*formato(1));
Re=zeros(1,formato(1));

for i=1:formato(1)
    Re(i) = str2num(fgetl(fid));

    for j=1:formato(2)
        polar2=str2num(fgetl(fid));
        polar(j,1)=polar2(1);
        polar(j,3*i-1)=polar2(2);
        polar(j,3*i)=polar2(3);
        polar(j,3*i+1)=polar2(4);
    end
end
fclose(fid);

for i=1:formato(1)
    Re(2,i) = interp1(polar(:,1),polar(:,i*3-1),alpha,'spline');
    Re(3,i) = interp1(polar(:,1),polar(:,i*3),alpha,'spline');
    Re(4,i) = interp1(polar(:,1),polar(:,i*3+1),alpha,'spline');
end

if size(Re,2)==1
    cl =  Re(2,1);
    cd =  Re(3,1);
    cm =  Re(4,1);
else
    cl = interp1(Re(1,:),Re(2,:),reynolds,'spline');
    cd = interp1(Re(1,:),Re(3,:),reynolds,'spline');
    cm = interp1(Re(1,:),Re(4,:),reynolds,'spline');
end
```