```
function [c_l,c_dw] = DiamondAirfoil(M, alpha, epsilon1, epsilon2)
%Uses shock-expansion theory to solve for the sectional lift and wave-drag
 coefficients for a diamond-wedge airfoil
%   Breaks wing into 4 plates, and 4 zones
%   Zone 1 after first shock on top
%   Zone 2 after expansion fan on top
%   Zone 3 after first shock on bottom
%   Zone 4 after expansion fan on bottom

% M_inf
%           /M1 |/    M2    /
%         /   _ -^--__       /
%        (<-----------> (
%         \  M3|\   M4    \
%           \                     \

%figure out geometry first
%   Makes 2 right triangles with equal hight,
%   Scales trangles to have combines length of 1
%   Scaled Hypoanouse makes used length
H1 = (1/sind(epsilon1));
H2 = (1/sind(epsilon2));
a1 = (1/tand(epsilon1));
a2 = (1/tand(epsilon2));
R = (1/(a1+a2));

length1 = R*H1;
length2 = R*H2;


%Zone 1 top front plate
Theta1 = (alpha-epsilon1);
if (Theta1 == 0)
beta1 = asind(1/M);
else
beta1 = ObliqueShockBeta(M,abs(Theta1),1.4,'Weak');
end
Mn0 = M*sind(beta1);
P1 = 1+((7/6)*((Mn0^2)-1));
Mn1 = sqrt(  ( (1+((0.2)*(Mn0^2)) ) /  (  (1.4*(Mn0^2))  -  0.2   )  )   );
M1 = Mn1/sind(beta1-abs(Theta1));

%Zone 3 bottom front plate

Theta3 = (alpha+epsilon1);
if (Theta3 == 0)
beta3 = asind(1/M);
else
beta3 = ObliqueShockBeta(M,abs(Theta3),1.4,'Weak');
end
Mn0 = M*sind(beta3);
P3 = 1+((7/6)*((Mn0^2)-1));
```

```
Mn3 = sqrt(  ( (1+((0.2)*(Mn0^2)) ) /  (  (1.4*(Mn0^2))  -  0.2   )  )   );
M3 = Mn3/sind(beta3-abs(Theta3));

%Zone 2 top expanision fan

Theta2 = (epsilon1+epsilon2);
nu2 = sqrt(2.4/0.4)*atand(sqrt((0.4/2.4)*((M1^2)-1)))-atand(sqrt(((M1^2)-1)));
[M2,~,~] = flowprandtlmeyer(1.4,(Theta2+nu2),'nu');
P2 = P1*(  (1+(0.2*(M1^2)))  /  (1+(0.2*(M2^2)))  )^3.5;


%Zone 4 bottom expanision fan

Theta4 = (epsilon1+epsilon2);
nu4 = sqrt(2.4/0.4)*atand(sqrt((0.4/2.4)*((M3^2)-1)))-atand(sqrt(((M3^2)-1)));
[M4,~,~] = flowprandtlmeyer(1.4,(Theta4+nu4),'nu');
P4 = P3*(  (1+(0.2*(M3^2)))  /  (1+(0.2*(M4^2)))  )^3.5;

%Apply pressures to find lift and drag

F1 = length1*P1;
F3 = length1*P3;
F2 = length2*P2;
F4 = length2*P4;

L = (cosd(epsilon1)*(F3-F1))+(cosd(epsilon2)*(F4-F2));
D = (sind(epsilon1)*(F3+F1))-(sind(epsilon2)*(F4+F2));

c_l = L/(0.7*(M^2));
c_dw = D/(0.7*(M^2));

end
```

```
Not enough input arguments.

Error in DiamondAirfoil (line 20)
H1 = (1/sind(epsilon1));
```

Published with MATLAB® R2021b