
```

function [To, T, c_star, T_e, V_e, X_throat, epsilon, Cf, X_exit] ...
    = black_magic(gas, P1, phi, hf, type)
% Reference state
Tref = 298;
P_e = 101325; %Pa
P_amb = 101325; %Pa

%Number of species in gas solution
nsp = nSpecies(gas);

%Find relevant species
iC2H4 = speciesIndex(gas, 'C2H4');
iO2 = speciesIndex(gas, 'O2');
iCO2 = speciesIndex(gas, 'CO2');
iH2O = speciesIndex(gas, 'H2O');

% Declare mole ratios
x1 = zeros(nsp,1);
x1(iC2H4) = 1;
x1(iO2) = phi;

% Enthalpy initialization
set(gas, 'T', Tref, 'P', P1, 'X', x1);
mass_frac = massFraction(gas, 'C2H4');
equilibrate(gas, 'HP');

% Enthalpy shift
h1 = enthalpy_mass(gas);
h_correct = h1 + (hf.CH2 - hf.C2H4) * mass_frac;
setState_HP(gas, [h_correct, P1]);
equilibrate(gas, 'HP');

% Prepare for iteration
Ma = 1;
ho1 = enthalpy_mass(gas);
ho2 = 0;
s1 = entropy_mass(gas);
P = P1;

Po = pressure(gas);
To = temperature(gas);
a_o = soundspeed(gas);
rho_o = density(gas);
cp_o = cp_mass(gas);
cv_o = cv_mass(gas);
k_o = cp_o / cv_o;

dT = 1;
dP = 10000;
s2 = 0;

```

```

% Iterate T and P to convergence of enthalpy and entropy
while abs(s2 - s1) / abs(s1) > 0.0001
    %abs(s2 - s1) / abs(s1)
    P = P - dP;
    ho2 = 0;
    T = To;
    while abs(ho2 - ho1) / abs(ho1) > 0.03
        T = T - dT;
        set(gas, 'T', T, 'P', P);
        if strcmp(type, 'dissoc')
            equilibrate(gas, 'HP');
        end
        a_t = soundspeed(gas);
        h2 = enthalpy_mass(gas);
        V2 = Ma * a_t;
        ho2 = h2 + 0.5 * V2^2;
    end
    s2 = entropy_mass(gas);
end
rho_t = density(gas);

% Get c_star
c_star = a_o / k_o * (rho_o / rho_t) * (a_o / a_t);

% Save gas at throat
X_throat = moleFractions(gas);

% Get Temperature Exit
T_e = T;
s3 = 0;
while abs(s3-s2) / abs(s2) > 0.02
    T_e = T_e - dT;
    set(gas, 'T', T_e, 'P', P_e);
    if strcmp(type, 'dissoc')
        equilibrate(gas, 'HP');
    end
    s3 = entropy_mass(gas);
end

h3 = enthalpy_mass(gas);
V_e = sqrt(2 * (ho2 - h3));
rho_e = density(gas);

% Calculate epsilon
% V2 is Vthroat
epsilon = rho_t * V2 / (rho_e * V_e);
Cf = V_e / c_star + (P_e / Po - P_amb / Po) * epsilon;

% Get exit
X_exit = moleFractions(gas);

end

```
