```
function [eta_LHV, eta_HHV, eta_actual] = lucio(T, P, alpha, lambda)

% Universal gas constant R
R = 8.3144621;

% Molecular masses - kg/kmol
MM.O2 = 32;
MM.N2 = 28.02;
MM.C = 12.01;
MM.H = 1.008;
MM.H2 = 2 * MM.H;
MM.H2O = 18.016;
MM.CO2 = MM.C + MM.O2;
MM.air = 28.97;

Rvar.O2 = R / MM.O2 * 10^3;
Rvar.H2 = R / MM.H2 * 10^3;
Rvar.N2 = R / MM.N2 * 10^3;
Rvar.H2O = R / MM.H2O * 10^3;
Rvar.H2O_liq = 0; %to avoid matlab rounding errors of ln(1)

% Enthalpy (J/kg) and entropy (J/(kg*K)) of formation values
hf.H2O_vap = -241820 / MM.H2O * 1000;
sf.H2O_vap = 188.83 / MM.H2O * 1000;
hf.H2O_liq = -285830 / MM.H2O * 1000;
sf.H2O_liq = 69.92 / MM.H2O * 1000;
hf.O2 = 0;
sf.O2 = 205.04 / MM.O2 * 1000;
hf.N2 = 0;
sf.N2 = 191.61 / MM.N2 * 1000;
hf.H2 = 0;
sf.H2 = 130.68 / MM.H2 * 1000;

% Fuel heating values for H2 (J/kg)
LHV = 120 * 10^6;
HHV = 141.8 * 10^6;

% Calculates the integrals
fun_O2_h = @(T)sp_heats(T,'O2');
fun_O2_s = @(T)sp_heats(T,'O2')./T;
fun_N2_h = @(T)sp_heats(T,'N2');
fun_N2_s = @(T)sp_heats(T,'N2')./T;
fun_H2_h = @(T)sp_heats(T,'H2');
fun_H2_s = @(T)sp_heats(T,'H2')./T;
fun_H2O_vap_h = @(T)sp_heats(T,'H2O_vap');
fun_H2O_vap_s = @(T)sp_heats(T,'H2O_vap')./T;
fun_H2O_liq_h = @(T)sp_heats(T,'H2O_liq');
fun_H2O_liq_s = @(T)sp_heats(T,'H2O_liq')./T;

% Reference conditions
T_standard = 298;
P_standard = 101.325 * 10^3;
```

```
P_sat = exp(-1.2914e8 / T^3 + 8.2048e5 / T^2 - 6522.8 / T + 25.5887);

% Saturation computations
y_max = P_sat / P;
N_a = (0.5 * (lambda - 1) + (0.5 * lambda * 3.76));
y_test = (1 + alpha) / (1 + alpha + N_a);

if y_test > y_max
    beta = (y_max * N_a) / (1 - y_max);
    gamma = 1 + alpha - beta;
else
    beta = 1 + alpha;
    gamma = 0;
end

N_prod.H2O_vap = beta;
N_prod.H2O_liq = gamma;
N_prod.H2O = beta + gamma; % 1 + alpha = beta + gamma
N_prod.O2 = 0.5 * (lambda - 1);
N_prod.N2 = 0.5 * lambda * 3.76;
N_prod.sum = N_prod.H2O_vap + N_prod.O2 + N_prod.N2;

y_prod.H2O_vap = N_prod.H2O_vap ./ N_prod.sum;
y_prod.H2O_liq = N_prod.H2O_liq ./ N_prod.sum;
y_prod.H2O = N_prod.H2O ./ N_prod.sum;
y_prod.N2 = N_prod.N2 ./ N_prod.sum;
y_prod.O2 = N_prod.O2 ./ N_prod.sum;

m_prod.H2O_vap = N_prod.H2O_vap * MM.H2O;
m_prod.H2O_liq = N_prod.H2O_liq * MM.H2O;
m_prod.H2O = N_prod.H2O * MM.H2O;
m_prod.O2 = N_prod.O2 * MM.O2;
m_prod.N2 = N_prod.N2 * MM.N2;
m_prod.sum = m_prod.H2O + m_prod.O2 + m_prod.N2;

mf_prod.H2O_vap = m_prod.H2O_vap ./ m_prod.sum;
mf_prod.H2O_liq = m_prod.H2O_liq ./ m_prod.sum;
mf_prod.H2O = m_prod.H2O ./ m_prod.sum;
mf_prod.N2 = m_prod.N2 ./ m_prod.sum;
mf_prod.O2 = m_prod.O2 ./ m_prod.sum;

N_react.H2 = 1;
N_react.O2 = 0.5 * lambda;
N_react.N2 = 0.5 * lambda * 3.76;
N_react.H2O_vap = alpha;
N_react.sum = N_react.O2 + N_react.N2 + N_react.H2O_vap;

y_react.H2 = N_react.H2 ./ N_react.sum;
y_react.O2 = N_react.O2 ./ N_react.sum;
y_react.N2 = N_react.N2 ./ N_react.sum;
y_react.H2O_vap = N_react.H2O_vap ./ N_react.sum;

m_react.H2 = N_react.H2 * MM.H2;
```

```
m_react.O2 = N_react.O2 * MM.O2;
m_react.N2 = N_react.N2 * MM.N2;
m_react.H2O_vap = N_react.H2O_vap * MM.H2O;
m_react.sum = m_react.H2 + m_react.O2 + m_react.N2 + m_react.H2O_vap;

mf_react.H2 = m_react.H2 ./ m_react.sum;
mf_react.O2 = m_react.O2 ./ m_react.sum;
mf_react.N2 = m_react.N2 ./ m_react.sum;
mf_react.H2O_vap = m_react.H2O_vap ./ m_react.sum;

% Partial pressures using linear mixing rules
P_react.H2 = P;
P_react.O2 = y_react.O2 * P;
P_react.N2 = y_react.N2 * P;
P_react.H2O_vap = y_react.H2O_vap * P;
% Hacky - can't have a -Inf contribution
if (y_react.H2O_vap == 0)
    P_react.H2O_vap = P;
end

P_prod.O2 = y_prod.O2 * P;
P_prod.N2 = y_prod.N2 * P;
P_prod.H2O_vap = y_prod.H2O_vap * P;
P_prod.H2O_liq = P;

% Enthalpy and Gibbs free energy of prod and react - all in J / kg
H.O2 = hf.O2 + integral(fun_O2_h, T_standard, T);
g_react.O2 = H.O2...
    - T * ((sf.O2 + integral(fun_O2_s, T_standard, T))...
    - Rvar.O2 * log(P_react.O2/P_standard));
g_prod.O2 = H.O2...
    - T * ((sf.O2 + integral(fun_O2_s, T_standard, T))...
    - Rvar.O2 * log(P_prod.O2/P_standard));

H.N2 = hf.N2 + integral(fun_N2_h, T_standard, T);
g_react.N2 = H.N2...
    - T * ((sf.N2 + integral(fun_N2_s, T_standard, T))...
    - Rvar.N2 * log(P_react.N2/P_standard));
g_prod.N2 = H.N2...
    - T * ((sf.N2 + integral(fun_N2_s, T_standard, T))...
    - Rvar.N2 * log(P_prod.N2/P_standard));

H.H2 = hf.H2 + integral(fun_H2_h, T_standard, T);
g_react.H2 = H.H2...
    - T * ((sf.H2 + integral(fun_H2_s, T_standard, T))...
    - Rvar.H2 * log(P_react.H2/P_standard));

H.H2O_vap = hf.H2O_vap + integral(fun_H2O_vap_h, T_standard, T);
g_react.H2O_vap = H.H2O_vap...
    - T * ((sf.H2O_vap + integral(fun_H2O_vap_s, T_standard, T))...
    - Rvar.H2O * log(P_react.H2O_vap/P_standard));
g_prod.H2O_vap = H.H2O_vap...
    - T * ((sf.H2O_vap + integral(fun_H2O_vap_s, T_standard, T))...
    - Rvar.H2O * log(P_prod.H2O_vap/P_standard));
```

```
% H.H2O_liq = hf.H2O_liq + integral(fun_H2O_liq_h, T_standard, T);
H.H2O_liq = hf.H2O_liq + 4200*(T-T_standard);
g_prod.H2O_liq = H.H2O_liq...
    - T * ((sf.H2O_liq + 4200*log(T/T_standard))...
    - Rvar.H2O_liq * log(P_prod.H2O_liq/P_standard));


% Reactants & Products:
g.react = mf_react.N2 .* g_react.N2 + mf_react.O2 .* g_react.O2...
    + mf_react.H2 .* g_react.H2 + mf_react.H2O_vap .* g_react.H2O_vap;

g.prod = mf_prod.N2 .* g_prod.N2 +  mf_prod.O2 .* g_prod.O2...
    + mf_prod.H2O_vap .* g_prod.H2O_vap + mf_prod.H2O_liq .* g_prod.H2O_liq;

% Calculate actual enthalpy change for denominator
H_prod = m_prod.N2 * H.N2 + m_prod.O2 * H.O2...
    + m_prod.H2O_vap * H.H2O_vap + m_prod.H2O_liq * H.H2O_liq;
H_prod = H_prod / 1000;
H_react = m_react.N2 * H.N2 + m_react.O2 * H.O2...
    + m_react.H2 * H.H2 + m_react.H2O_vap * H.H2O_vap;
H_react = H_react / 1000;
H_actual = H_react - H_prod;

% Calculate efficiencies
eta_LHV = m_react.sum * (g.react - g.prod) ./ (m_react.H2 * LHV);
eta_HHV = m_react.sum * (g.react - g.prod) ./ (m_react.H2 * HHV);
eta_actual = (m_react.sum / 1000) * (g.react - g.prod) ./ (H_actual);

end
```

Published with MATLAB® R2013a