

```

clc;
clear;

% Input parameters
c = 8;
h = 18;
o = 0;
airs = 12.5;
temp = (100:25:3000)';
press = (20:5:120)';
feq = (0.8:0.02:1.3)';

% Initialize result storage
n5nt = zeros(49, 4, 26);
iterations = zeros(49, 4, 26);
computationTimes = zeros(49, 4, 26);

% Loop through temperature, pressure, and fuel equivalence ratio
for i = 1:length(temp)
    for j = 1:length(press)
        for b = 1:length(feq)
            [n5nt(i, j, b), iterations(i, j, b), computationTimes(i, j, b)] = ...
                Improved_mydissfuniii(c, h, o, temp(i), feq(b), airs, press(j));
        end
    end
end

% Display average performance metrics
avgIterations = mean(iterations(:));
avgComputationTime = mean(computationTimes(:));

fprintf('Average Iterations: %.2f\n', avgIterations);

Average Iterations: 25.98

fprintf('Average Computation Time: %.4f seconds\n', avgComputationTime);

Average Computation Time: 0.0001 seconds

% Save results to a .mat file for later analysis
save('results.mat', 'n5nt', 'iterations', 'computationTimes');

```

```

function [n5ntvalue, iterCount, computationTime] = Improved_mydissfuniii(carbon,
hydrogen, oxygen, temperature, fuelequiv, molar_air_stoich, pressure)
    % Initialize variables
    tic; % Start time tracking
    iterCount = 0; % Initialize iteration counter

    % Gibbs free energy table (unchanged)
    gibbseng = [0, -393.151, -238.921, -113.805;
        100, -393.683, -236.584, -120.239;
        200, -394.085, -232.766, -128.526;
        298.15, -394.389, -228.582, -137.163;
        300, -394.394, -228.5, -137.328;
        400, -394.675, -223.901, -146.338;
        500, -394.939, -219.051, -155.414;
        600, -395.182, -214.007, -164.486;
        700, -395.398, -208.812, -173.518;
        800, -395.586, -203.496, -182.497;
        900, -395.748, -198.083, -191.416;
        1000, -395.886, -192.59, -200.275;
        1100, -396.001, -187.033, -209.075;
        1200, -396.098, -181.425, -217.819;
        1300, -396.177, -175.774, -226.509;
        1400, -396.24, -170.089, -235.149;
        1500, -396.288, -164.376, -243.74;
        1600, -396.323, -158.639, -252.284;
        1700, -396.344, -152.883, -260.784;
        1800, -396.353, -147.111, -269.242;
        1900, -396.349, -141.325, -277.658;
        2000, -396.333, -135.528, -286.034;
        2100, -396.304, -129.721, -294.372;
        2200, -396.262, -123.905, -302.672;
        2300, -396.209, -118.082, -310.936;
        2400, -396.142, -112.252, -319.165;
        2500, -396.062, -106.416, -327.358;
        2600, -395.969, -100.575, -335.517;
        2700, -395.862, -94.729, -343.643;
        2800, -395.742, -88.878, -351.736;
        2900, -395.609, -83.023, -359.797;
        3000, -395.461, -77.163, -367.826];

    % Gibbs free energy
    gco2 = interp1(gibbseng(:,1), gibbseng(:,2), temperature) * 10^3;
    gh2o = interp1(gibbseng(:,1), gibbseng(:,3), temperature) * 10^3;
    gco = interp1(gibbseng(:,1), gibbseng(:,4), temperature) * 10^3;

    % Equilibrium constants
    kco2 = exp(-(gco2 - gco)) / (8314.46 * temperature));
    kh2o = exp(-(gh2o) / (8314.46 * temperature));

```

```

% Define the function to solve
function error = dissociation_error(n5ntvalue)
    a = sqrt(n5ntvalue * pressure);
    n3 = (carbon * fuelequiv) / (kco2 * a + 1);
    n4 = ((hydrogen / 2) * fuelequiv) / (kh2o * a + 1);
    n1 = carbon * fuelequiv - n3;
    n2 = (hydrogen / 2) * fuelequiv - n4;
    n5 = (molar_air_stoich * 2 + oxygen * fuelequiv - 2 * n1 - n2 - n3) / 2;
    n6 = molar_air_stoich * 3.76;
    ntsum = n1 + n2 + n3 + n4 + n5 + n6;
    error = n5ntvalue - n5 / ntsum;
    iterCount = iterCount + 1; % Count iterations
end

% Use MATLAB's fzero method to solve for n5ntvalue
options = optimset('Display', 'off'); % Suppress output
try
    n5ntvalue = fzero(@dissociation_error, 0.1, options);
catch exception
    warning(['fzero failed to converge: ', exception.message]);
    n5ntvalue = NaN;
end

% Capture computation time
computationTime = toc;
end

```