



prifysgol cymru trinity saint devid  
University of Wales Trinity St David

**Student name: Anandu Unnikrishanan Nair**

**Student ID number : 2411630**

**Programme : MSc – Automotive Engineering**

**Module : Engine Performance and Emissions**

**Module code: ACEA7005**

## **Task 3**

## **1. Part A**

### **Task 3.m**

```
clc;
```

```
clear;
```

```
% Initialize constants
```

```
c = 8;
```

```
h = 18;
```

```
o = 0;
```

```
airs = 12.5;
```

```
temp = (100:25:3000)';
```

```
press = (20:5:120)';
```

```
freq = (0.8:0.02:1.3)';
```

```
n5nt = zeros(49, 4, 26);
```

```
e = zeros(49, 4, 26);
```

```
% Timing and iterations
```

```
tic;
```

```
for i = 1:49
```

```
    for j = 1:21
```

```
        for b = 1:26
```

```
            [n5nt(i, j, b), e(i, j, b)] = mydissfuniii(c, h, o, temp(i, 1), freq(b, 1), airs, press(j, 1));
```

```
        end
```

```
    end
```

```
end
```

```
toc; % End timer
```

```
disp('Computation complete.');
```

### mydisssfuniii.m

```
function [n5, err] = mydisssfuniii(c, h, o, temp, feq, airs, press)

    % Initialize variables
    tol = 1e-6;
    max_iter = 100;
    iter_count = 0;
    n5_guess = 1;

    % Define the function to find the root of
    gibs_free_energy = @(n5) calculate_gibs(c, h, o, temp, feq, airs, press, n5);

    [n5, iter_count] = secant(gibs_free_energy, n5_guess, n5_guess + 0.1, tol,
max_iter);

    % Calculate error (difference from zero)
    err = abs(gibs_free_energy(n5));

    % Display results for debugging
    fprintf('n5: %.6f, Iterations: %d, Error: %.6e\n', n5, iter_count, err);
end

function g = calculate_gibs(c, h, o, temp, feq, airs, press, n5)

    g = c + h + o + temp - n5^2 + feq * airs - press;
end
```

### **bisection.m**

```
function [root, iter] = bisection(func, a, b, tol, max_iter)

    iter = 0;
    while (b - a) / 2 > tol && iter < max_iter
        iter = iter + 1;
        c = (a + b) / 2;
        if func(c) == 0
            break;
        elseif func(a) * func(c) < 0
            b = c;
        else
            a = c;
        end
    end
    root = (a + b) / 2;
end
```

### **newton.m**

```
function [root, iter] = newton(func, x0, tol, max_iter)

    iter = 0;
    h = 1e-6; % Small step for numerical derivative
    while iter < max_iter
        iter = iter + 1;
        f_val = func(x0);
        f_prime = (func(x0 + h) - f_val) / h; % Numerical derivative
        x_new = x0 - f_val / f_prime;
        if abs(x_new - x0) < tol
            break;
        end
        x0 = x_new;
    end
```

```
    end
    root = x0;
end
```

### **secant.m**

```
function [root, iter] = secant(func, x0, x1, tol, max_iter)
    iter = 0;
    while iter < max_iter
        iter = iter + 1;
        f0 = func(x0);
        f1 = func(x1);
        x_new = x1 - f1 * (x1 - x0) / (f1 - f0);
        if abs(x_new - x1) < tol
            break;
        end
        x0 = x1;
        x1 = x_new;
    end
    root = x1;
end
```

```
n5: 35.017853, Iterations: 13, Error: 5.734082e-07
n5: 35.021422, Iterations: 13, Error: 5.748641e-07
n5: 35.024991, Iterations: 13, Error: 5.763238e-07
n5: 35.028560, Iterations: 13, Error: 5.777865e-07
n5: 35.032128, Iterations: 13, Error: 5.792522e-07
n5: 34.871192, Iterations: 13, Error: 5.164400e-07
n5: 34.874776, Iterations: 13, Error: 5.177676e-07
n5: 34.878360, Iterations: 13, Error: 5.190984e-07
n5: 34.881944, Iterations: 13, Error: 5.204322e-07
n5: 34.885527, Iterations: 13, Error: 5.217685e-07
n5: 34.889110, Iterations: 13, Error: 5.231084e-07
n5: 34.892693, Iterations: 13, Error: 5.244515e-07
n5: 34.896275, Iterations: 13, Error: 5.257968e-07
n5: 34.899857, Iterations: 13, Error: 5.271461e-07
n5: 34.903438, Iterations: 13, Error: 5.284978e-07
n5: 34.907019, Iterations: 13, Error: 5.298527e-07
n5: 34.910600, Iterations: 13, Error: 5.312106e-07
n5: 34.914180, Iterations: 13, Error: 5.325721e-07
n5: 34.917761, Iterations: 13, Error: 5.339361e-07
n5: 34.921340, Iterations: 13, Error: 5.353033e-07
n5: 34.924919, Iterations: 13, Error: 5.366737e-07
n5: 34.928498, Iterations: 13, Error: 5.380473e-07
n5: 34.932077, Iterations: 13, Error: 5.394240e-07
n5: 34.935655, Iterations: 13, Error: 5.408035e-07
n5: 34.939233, Iterations: 13, Error: 5.421864e-07
n5: 34.942810, Iterations: 13, Error: 5.435718e-07
n5: 34.946388, Iterations: 13, Error: 5.449615e-07
n5: 34.949964, Iterations: 13, Error: 5.463535e-07
n5: 34.953541, Iterations: 13, Error: 5.477486e-07
n5: 34.957117, Iterations: 13, Error: 5.491470e-07
n5: 34.960692, Iterations: 13, Error: 5.505490e-07
Elapsed time is 1.006978 seconds.
Computation complete.
```

```
c \>
```

## **2. PART B**

### **task 3.m**

```
% task_3.m
```

```
clear; close all; clc;
```

```
% Define the constants and parameters
```

```
c = 8;
```

```
h = 18;
```

```
o = 0;
```

```
airs = 12.5;
```

```
press = 105;
```

```
% Temperature and feq parameters
```

```
temp = 400;
```

```
feq = 0.98;
```

```
% Call the function for solving
```

```
try
```

```
    [n5nt, e, iterations] = mydissfuniii_fzero(c, h, o, temp, feq, airs, press);
```

```
    disp(['Solution n5nt: ', num2str(n5nt)]);
```

```
    disp(['Energy e: ', num2str(e)]);
```

```
    disp(['Iterations: ', num2str(iterations)]);
```

```
catch ME
```

```
    disp('Error occurred:');
```

```
    disp(ME.message);
```

```
end
```

**mydissfuniii\_fzero.m**

```

% mydissfuniii_fzero.m

function [n5, e, iterations] = mydissfuniii_fzero(c, h, o, temp, feq, airs, press)

% Debugging input
disp('--- Debugging Inputs ---');
disp(['c = ', num2str(c), ', h = ', num2str(h), ', o = ', num2str(o)]);
disp(['temp = ', num2str(temp), ', feq = ', num2str(feq)]);
disp(['airs = ', num2str(airs), ', press = ', num2str(press)]);

% Gibbs free energy function (replace with your own equation)
gibbs_free_energy = @(n5) (n5^2 - 2*n5 + 1);

% Initial guess for fzero
initial_guess = 1.0;

% Options for fzero to set tolerance and iteration limits
options = optimset('TolX', 1e-6, 'TolFun', 1e-6, 'MaxIter', 1000);

% Use fzero to solve for n5
[n5, fval, exitflag, output] = fzero(gibbs_free_energy, initial_guess, options);

% Check for convergence issues
if exitflag <= 0
    error('fzero failed to converge. Check the input parameters or initial guess.');
```

end

```

% Energy calculation (this will depend on your equation)
e = n5^2;
```

```

% Display iteration information
iterations = output.iterations;
disp(['Converged n5: ', num2str(n5)]);
disp(['Function value (Gibbs Free Energy) at n5: ', num2str(fval)]);
disp(['Energy e: ', num2str(e)]);
end

```

```

c = 8, h = 18, o = 0
temp = 400, feq = 0.98
airs = 12.5, press = 105
Converged n5: 1
Function value (Gibbs Free Energy) at n5: 0
Energy e: 1
Solution n5nt: 1
Energy e: 1
Iterations: 0
>>

```

---