

UWTSD

Module

Task 3

ACHA7002

2504057

12/06/2025

## Contents

|       |                                                                  |   |
|-------|------------------------------------------------------------------|---|
| 1     | Task 3a .....                                                    | 1 |
| 1.1   | Table of own functions results compared with given function..... | 1 |
| 2     | Task 3b .....                                                    | 1 |
| 2.1   | Table of research of MATLAB/Online functions .....               | 1 |
| 2.1.1 | Additional researched functions findings .....                   | 2 |
| 2.2   | Table of tried MATLAB/Online functions results .....             | 2 |
| 3     | Referencing.....                                                 | 2 |
| 4     | Appendices.....                                                  | 3 |
| 4.1   | Own code and functions task 3a .....                             | 3 |
| 4.2   | Own code task 3b .....                                           | 6 |

## 1 Task 3a

### 1.1 Table of own functions results compared with given function

| FUNCTION      | TIME(s)    | ITERATIONS | RESULT DIFF       | LINES OF CODE* |
|---------------|------------|------------|-------------------|----------------|
| mydisssfuniii | 1.406483** | Nan        | 0.039619651349001 | 223            |
| bisect3       | 11.170291  | 19/value   | 0.039620208740234 | 170            |
|               |            |            |                   |                |

\*Lines of code is th total lines summing up main script and functions.

\*\* Time for “mydisssfuniii” is for the base code provide, putting more steps increases the time.

## 2 Task 3b

### 2.1 Table of research of MATLAB/Online functions

| Function | Ease of use                                                                  | Efficiency                                     | Accurancy                                                        | Additional information                                                                                  |
|----------|------------------------------------------------------------------------------|------------------------------------------------|------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------|
| fsolve   | <b>Medium</b> – Requires writing residual equations but general and flexible | <b>Good</b> – Efficient even with many species | <b>Very High</b> – Robust for equilibrium + atom balance systems | Ideal for combustion products, multi-species equilibrium, or thermodynamic models with several unknowns |
| fzero    | <b>Very High</b> – Simple to use, minimal setup required                     | <b>Excellent</b> – Very fast for 1-D problems  | <b>High</b> – Precise for scalar equations                       | Best for single equilibrium reaction or simple degree-of-dissociation calculations                      |

|               |                                                                                               |                                                                    |                                                                            |                                                                                              |
|---------------|-----------------------------------------------------------------------------------------------|--------------------------------------------------------------------|----------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
| mydisssfuniii | <b>Low</b> – Hard to generalize, requires manual coding of balances and equilibrium relations | <b>Moderate</b> – Can be fast in simple systems, but scales poorly | <b>Uncertain</b> – Accuracy depends entirely on correct manual formulation | Suitable for demonstrating reaction mechanisms, learning how equilibrium equations are built |
|---------------|-----------------------------------------------------------------------------------------------|--------------------------------------------------------------------|----------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|

### 2.1.1 Additional researched functions findings

## 2.2 Table of tried MATLAB/Online functions results

| FUNCTION      | TIME(s)    | ITERATIONS | RESULT DIFF       | LINES OF CODE* |
|---------------|------------|------------|-------------------|----------------|
| mydisssfuniii | 1.406483** | Nan        | 0.039619651349001 | 223            |
| fzero         | 20.488634  | Nan        | 0.039619651347068 | 126            |
| fsolve        | 26.136042  | Nan        | 0.039619651347068 | 126            |

\*Lines of code is the total lines summing up main script and functions.

\*\* Time for “mydisssfuniii” is for the base code provide, putting more steps increases the time.

## 3 Referencing

[1] Matlab. “fzero, Root of nonlinear function.” Matlab. Accessed: Dec. 3, 2025. [Online]. Available: <https://uk.mathworks.com/help/releases/R2025b/matlab/ref/fzero.html>

[2] Matlab. “fsolve, Solve system of nonlinear equations.” Matlab. Accessed: Dec. 3, 2025. [Online]. Available: <https://uk.mathworks.com/help/optim/ug/fsolve.html>

## 4 Appendices

### 4.1 Own code and functions task 3a

```

clc
clear
temp=(100:25:3000)';
press=(20:5:120)';
feq=(0.8:0.02:1.3)';
n5nt=zeros(49,4,26);
params.carbon = 8; % Storage the constant to include in the function
params.hydrogen = 18; % Storage the constant to include in the function
params.oxygen = 0; % Storage the constant to include in the function
params.air = 12.5; % Storage the constant to include in the function
%TEMP CO2      H2O          CO
params.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
3100     -395.298      -71.298       -375.824
3200     -395.122      -65.43      -383.792
3300     -394.932      -59.558       -391.73
3400     -394.726      -53.681       -399.638
3500     -394.506      -47.801       -407.517
3600     -394.271      -41.916       -415.366
3700     -394.022      -36.027       -423.188
3800     -393.756      -30.133       -430.981
3900     -393.477      -24.236       -438.746
4000     -393.183      -18.334       -446.485
4100     -392.874      -12.427       -454.196

```

|      |          |        |            |
|------|----------|--------|------------|
| 4200 | -392.55  | -6.516 | -461.881   |
| 4300 | -392.21  | -0.6   | -469.539   |
| 4400 | -391.857 | 5.32   | -477.171   |
| 4500 | -391.488 | 11.245 | -484.778   |
| 4600 | -391.105 | 17.175 | -492.359   |
| 4700 | -390.706 | 23.111 | -499.915   |
| 4800 | -390.292 | 29.052 | -507.446   |
| 4900 | -389.862 | 34.998 | -514.951   |
| 5000 | -389.419 | 40.949 | -522.433   |
| 5100 | -388.959 | 46.906 | -529.89    |
| 5200 | -388.486 | 52.869 | -537.323   |
| 5300 | -387.996 | 58.838 | -544.732   |
| 5400 | -387.493 | 64.811 | -552.118   |
| 5500 | -386.974 | 70.791 | -559.48    |
| 5600 | -386.439 | 76.777 | -566.818   |
| 5700 | -385.89  | 82.769 | -574.133   |
| 5800 | -385.324 | 88.767 | -581.424   |
| 5900 | -384.745 | 94.77  | -588.694   |
| 6000 | -384.148 | 100.78 | -595.939]; |

```

for i=1:49
    for j=1:21
        for b=1:26
            params.T = temp(i); % Storage the constant to include in the function
            params.P = press(j); % Storage the constant to include in the function
            params.feq = feq(b); % Storage the constant to include in the function

            f = @(n5nt) mydissfuniii_own(n5nt,params); % Use 'params' to include
all the constant need to Gibbs energy operation
            % In that way we can develop our myfissfuniii into one variable
            % that our bisect function can manage.

            n5nt(i,j,b) = bisect3(f, 0, 0.2, 50, 1e-6);

        end
    end
end

```

```

-----

function error = mydissfuniii_own(n5nt, params)
    % unpack parameters
    carbon      = params.carbon;
    hydrogen    = params.hydrogen;
    oxygen      = params.oxygen;
    temperature = params.T;
    fuelequiv   = params.feq;
    molar_air_stoich = params.airs;
    pressure    = params.P;

```

```

gibbseng = params.gibbseng;

```

```

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

%Equilibrium constants
kco2=exp((- (gco2-gco))*10^3/(8314.46*temperature));
kh2o=exp((- gh2o)*10^3/(8314.46*temperature));

%Goodger method for calculation of dissociation

a=sqrt(n5nt*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= n5nt - (n5./ntsum);

-----

% a) bisection function in MATLAB
function root = bisect3(f,a,b,imax,tol)
% Input:
% f --> function to be evaluated
% a,b --> interval limits
% imax --> maximum number of iterations
% tol --> convergence criterion
% Output:
% root --> x position of the root
% initialise parameters
x1=a;
f1=feval(f,x1);
x3=b;
f3=feval(f,x3);
i=1;
% CHECK: verify that there is indeed a root in the interval [a b].
if f1*f3>0
    fprintf('No root in the specified interval')
end

% Begin iteration loop
while abs(x3 - x1) > tol && i< imax % Limit of interaction

    x2 = (x3+x1)/2;
    f2 = feval(f, x2);
    root = x2;
    % Find which half interval contains root
    if f2 == 0
        root = x2;

    elseif f1 * f2 < 0
        x3 = x2;
        f3 = f2;
    else

```

```

        x1 = x2;
        f1 = f2;
    end
    %fprintf('Root guess at x = %10.6f after %d iterations and an error of %d.\n',
    root, i, abs(x3 - x1)); % Convergence table

    i = i + 1;

end
fprintf('Root found at x = %10.6f after %d iterations and an error of %d.\n',
root, i, abs(x3 - x1)) %final guess

```

## 4.2 Own code task 3b

----- With fzero -----

```

clc
clear
temp=(100:25:3000)';
press=(20:5:120)';
freq=(0.8:0.02:1.3)';
n5nt=zeros(49,4,26);
params.carbon = 8; % Storage the constant to include in the fuction
params.hydrogen = 18; % Storage the constant to include in the fuction
params.oxygen = 0; % Storage the constant to include in the fuction
params.air = 12.5; % Storage the constant to include in the fuction
%TEMP CO2      H2O      CO
params.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   |
| 3100 | -395.298 | -71.298 | -375.824   |
| 3200 | -395.122 | -65.43  | -383.792   |
| 3300 | -394.932 | -59.558 | -391.73    |
| 3400 | -394.726 | -53.681 | -399.638   |
| 3500 | -394.506 | -47.801 | -407.517   |
| 3600 | -394.271 | -41.916 | -415.366   |
| 3700 | -394.022 | -36.027 | -423.188   |
| 3800 | -393.756 | -30.133 | -430.981   |
| 3900 | -393.477 | -24.236 | -438.746   |
| 4000 | -393.183 | -18.334 | -446.485   |
| 4100 | -392.874 | -12.427 | -454.196   |
| 4200 | -392.55  | -6.516  | -461.881   |
| 4300 | -392.21  | -0.6    | -469.539   |
| 4400 | -391.857 | 5.32    | -477.171   |
| 4500 | -391.488 | 11.245  | -484.778   |
| 4600 | -391.105 | 17.175  | -492.359   |
| 4700 | -390.706 | 23.111  | -499.915   |
| 4800 | -390.292 | 29.052  | -507.446   |
| 4900 | -389.862 | 34.998  | -514.951   |
| 5000 | -389.419 | 40.949  | -522.433   |
| 5100 | -388.959 | 46.906  | -529.89    |
| 5200 | -388.486 | 52.869  | -537.323   |
| 5300 | -387.996 | 58.838  | -544.732   |
| 5400 | -387.493 | 64.811  | -552.118   |
| 5500 | -386.974 | 70.791  | -559.48    |
| 5600 | -386.439 | 76.777  | -566.818   |
| 5700 | -385.89  | 82.769  | -574.133   |
| 5800 | -385.324 | 88.767  | -581.424   |
| 5900 | -384.745 | 94.77   | -588.694   |
| 6000 | -384.148 | 100.78  | -595.939]; |

```

for i=1:49
    for j=1:21
        for b=1:26
            params.T = temp(i); % Storage the constant to include in the fuction
            params.P = press(j); % Storage the constant to include in the fuction
            params.feq = feq(b); % Storage the constant to include in the fuction

            f = @(n5nt) mydissfuniii_ow(n5nt,params); % Use 'params' to include
all the constant need to Gibbs energy operation
            % In that way we can develop our myfissfuniii into one variable.

            n5nt(i,j,b) = fzero(f,[0,0.2]); % Solving with fzero function

        end
    end
end

----- With fsolve -----

clc
clear
temp=(100:25:3000)';

```

```

press=(20:5:120)';
freq=(0.8:0.02:1.3)';
n5nt=zeros(49,4,26);
params.carbon = 8; % Storage the constant to include in the fuction
params.hydrogen = 18; % Storage the constant to include in the fuction
params.oxygen = 0; % Storage the constant to include in the fuction
params.air = 12.5; % Storage the constant to include in the fuction
%TEMP CO2      H2O      CO
params.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
3100      -395.298      -71.298      -375.824
3200      -395.122      -65.43      -383.792
3300      -394.932      -59.558      -391.73
3400      -394.726      -53.681      -399.638
3500      -394.506      -47.801      -407.517
3600      -394.271      -41.916      -415.366
3700      -394.022      -36.027      -423.188
3800      -393.756      -30.133      -430.981
3900      -393.477      -24.236      -438.746
4000      -393.183      -18.334      -446.485
4100      -392.874      -12.427      -454.196
4200      -392.55      -6.516      -461.881
4300      -392.21      -0.6      -469.539
4400      -391.857      5.32      -477.171
4500      -391.488      11.245      -484.778
4600      -391.105      17.175      -492.359
4700      -390.706      23.111      -499.915
4800      -390.292      29.052      -507.446
4900      -389.862      34.998      -514.951

```

|      |          |        |            |
|------|----------|--------|------------|
| 5000 | -389.419 | 40.949 | -522.433   |
| 5100 | -388.959 | 46.906 | -529.89    |
| 5200 | -388.486 | 52.869 | -537.323   |
| 5300 | -387.996 | 58.838 | -544.732   |
| 5400 | -387.493 | 64.811 | -552.118   |
| 5500 | -386.974 | 70.791 | -559.48    |
| 5600 | -386.439 | 76.777 | -566.818   |
| 5700 | -385.89  | 82.769 | -574.133   |
| 5800 | -385.324 | 88.767 | -581.424   |
| 5900 | -384.745 | 94.77  | -588.694   |
| 6000 | -384.148 | 100.78 | -595.939]; |

```
for i=1:49
    for j=1:21
        for b=1:26
            params.T = temp(i); % Storage the constant to include in the fuction
            params.P = press(j); % Storage the constant to include in the fuction
            params.feq = feq(b); % Storage the constant to include in the fuction

            f = @(n5nt) mydissfuniii_own(n5nt,params); % Use 'params' to include
all the constant need to Gibbs energy operation
            % In that way we can develop our mydissfuniii into one variable.

            n5nt(i,j,b) = fsolve(f,0.2); % Solving with fsolve function
        end
    end
end
```