Final Exam Solutions
Administered: Monday, December 12, 2021
10:30 AM – 12:45 PM
38 points

Problem 1. (14 points)

The one-dimensional heat equation can describe heat transfer in a rutile TiO_2 with both heat conduction and radiative heat loss.

$$0 = \frac{d^2T}{dz^2} - \frac{\varepsilon\sigma S}{k_c}\left(T^4 - {T_s}^4\right)$$

A rutile TiO_2 cylindrical rod has diameter 0.05 m and length 0.4 m. One end of the rod is maintained at $T(z = 0) = 500$ K. The other end of the rod is maintained at $T(z = L) = 550$ K. The variables and parameters are defined below.

- T is the temperature, K
- z is the axial position, m
- k_c is the thermal conductivity, $k_c = 9.0 \frac{W}{m \cdot K}$
- d is the diameter of the rod, $d = 0.05\ m$
- l is the length of the rod, $l = 0.4\ m$
- A is the surface area of the rod, $A = \pi d l$
- V is the volume of the rod, $V = \frac{\pi}{4} d^2 l$
- S is the surface area to volume ratio of the rod, $S = {}^{A}/_{V} = \frac{4}{d} = 80.0\ m^{-1}$
- T_{surr} is the surrounding temperature, $T_{surr} = 480\ K$
- σ is the Stefan–Boltzmann constant, $\sigma = 5.670373x10^{-8} \frac{W}{m^2 \cdot K^4}$
- ε is the gray body emissivity $\varepsilon = 0.79$

Answer the following questions and perform the following tasks.

(a) Is this ODE problem linear or nonlinear?
(b) Is this ODE problem an initial value problem or a boundary value problem?
(c) Convert this second order ODE into a system of two first order ODEs.
(d) Find the initial temperature gradient at z = 0. Important hint: a good initial guess for the initial temperature gradient is $-250 \frac{K}{m}$.
(e) Sketch the temperature profile.
(f) Verify that your discretization resolution was sufficient.
(g) What is the temperature in the middle of the rod?

Solution

(a) Is this ODE problem linear or nonlinear?

The problem is nonlinear due to exponent of 4 in the radiation term.

(b) Is this ODE problem an initial value problem or a boundary value problem?

This problem is a boundary value problem because both conditions are not given at the same value of the independent variable, *z*.

(c) Convert this second order ODE into a system of two first order ODEs.

This conversion follows a three step process.

Step 1. Define new variables.

$$y_1 = T \qquad y_2 = \frac{dT}{dz}$$

Step 2. Write ODEs for the new variables.

In this transformation, the first equation is always

$$\frac{dy_1}{dz} = y_2$$

The second equation is substituting the variables in Step 1 into the original ODE.

$$0 = \frac{d^2T}{dz^2} - \frac{\varepsilon\sigma S}{k_c}\left(T^4 - {T_s}^4\right)$$

or

$$\frac{d^2T}{dz^2} = \frac{\varepsilon\sigma S}{k_c}\left(T^4 - {T_s}^4\right)$$

$$\frac{dy_2}{dz} = \frac{\varepsilon\sigma S}{k_c}\left(y_1^4 - {T_s}^4\right)$$

Step 3. Write initial conditions for the new variables.

$$y_1(z=0) = T(z=0) = 500\ K \qquad y_2(z=0) = \left.\frac{dT}{dz}\right|_0 \text{ (not given)}$$

(d) Find the initial temperature gradient at *z* = 0. Important hint: a good initial guess for the initial temperature gradient is $-250\frac{K}{m}$.

(e) Sketch the concentration profile.

This is a boundary value problem. I used as the starting points the two Matlab functions, rk4n.m and nrnd1.m, distributed on the course website in the odesolver_bvp folder.

I modified the input file for rk4n.m, which uses the classical fourth-order Runge-Kutta method to solve a system of n ODEs.

```
function dydx = funkeval(x,y);
sig = 5.670373e-8; % W/m^2/K^4
eps = 0.79;
kc = 9.0; % W/m/K
S = 80.0; % 1/m
Tsurr = 480.0; % K
con = eps*sig*S/kc;
dydx(1) = y(2);
dydx(2) = con*(y(1)^4-Tsurr^4);
```

I also modified the input for nrnd1.m, which uses the Newton Raphson method with numerical derivatives,

```
function f = funkeval(x)
xo = 0.0;
yo_1 = 500.0;
yo_2 = x;
xf = 0.4;
yf = 550.0;
n = 1000;
[x,y]=rk4n_xm4f22(n,xo,xf,[yo_1,yo_2]);
yf_calc = y(n+1,1);
f = yf_calc-yf;
```

At the command line prompt, I needed an initial guess for the initial slope. I used the suggested value of $-250\frac{K}{m}$.

```
>> [x0,err] = nrnd1(-250.0)
```

This command generated the following output:

```
>> [x0,err] = nrnd1(-250)
icount = 1 xold = -2.500000e+02 f = 1.330769e+02 df = 1.186853e+01 xnew = -2.612126e+02  err = 1.000000e+02
icount = 2 xold = -2.612126e+02 f = 1.480814e+01 df = 9.417037e+00 xnew = -2.627851e+02  err = 5.983916e-03
icount = 3 xold = -2.627851e+02 f = 2.252918e-01 df = 9.146423e+00 xnew = -2.628097e+02  err = 9.372441e-05
icount = 4 xold = -2.628097e+02 f = 2.117600e-04 df = 9.142298e+00 xnew = -2.628097e+02  err = 8.813474e-08

x0 = -262.8097
err =   8.8135e-08
```

The code converged because the error was less than the stated tolerance of 10^{-6}. The initial slope is

$$\left.\frac{dT}{dz}\right|_0 = -262.81\frac{K}{m}$$

We can run the Runge Kutta code with this initial condition to generate the result.

```
>> [x,y]=rk4n(1000,0,0.4,[500,-262.8097]);
>> >> y_n1000 = y(1001,1)

y_n1000 = 550.0002
```

When we use 1000 intervals in the Runge Kutta code, we obtain a temperature at the far end of the rod of 550.0 K. That matches our specified boundary condition.

(f) Verify that your discretization resolution was sufficient.

In order to verify that the spatial discretization was sufficiently fine, we also use 10,000 intervals in the Runge Kutta code. For this finer resolution, we obtain a concentration at the far end of the

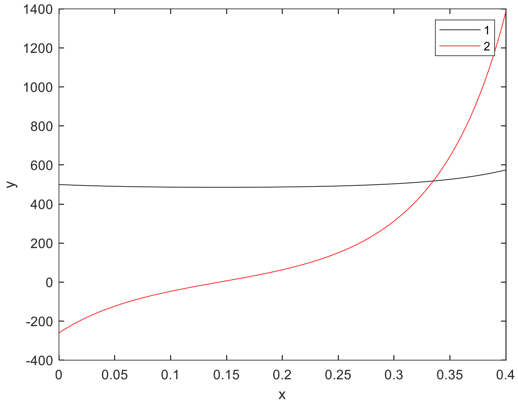
```
>> >> [x,y]=rk4n(10000,0,0.4,[500,-262.8097]);
>> y_n10000 = y(10001,1)

y_n10000 =     550.0002
```

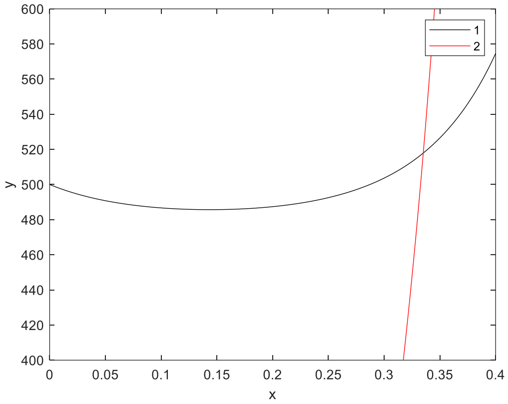
The two results agree, so we had a good discretization resolution.

A plot of the profile is shown below. The black line marked “1” is the temperature. The red line marked “2” is the temperature gradient.

A second plot is also shown which zooms in on the temperature. It is clear that there is sufficient cooling that the temperature actually drops below 500 near the lower temperature reservoir. The command used to zoom in on the plot was

```
>> axis([0 0.4 400 600])
```





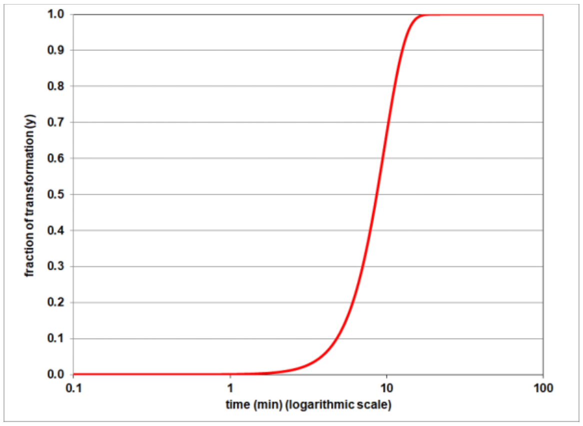
(g) What is the temperature in the middle of the rod?

```
[x,y]=rk4n_xm4f22(n,xo,xf,[yo_1,yo_2]);
zmid = x(501)
zmid = 0.200
ymid = y(501,1)
ymid = 485.9252
```

The temperature in the middle of the rod (z = 0.2 m) is 485.9 K.

Problem 2. (6 points)

“Most solid-state transformations do not occur instantaneously because obstacles impeded the course of the reaction and make it dependent on time.” [Callister, W.D., “Materials Science and Engineering: An Introduction, Wiley & Sons, New York, Fifth Ed., p. 296]. One expression used to describe the fraction of solid that has transformed is the Avrami equation, shown in the equation and Figure below, where y is the fraction of transformation, t is time (min), k is a rate constant (min^{-1}) and n is an exponent.



$$y = 1 - \exp\left(-kt^n\right)$$

This equation can be put into a form in which the unknown parameters, k and n , can be determined with linear regression:

$$\ln\left[\ln\left(\frac{1}{1-y}\right)\right] = \ln(k) + n\ln(t)$$

(a) Using the table of data providing y and t in the file, http://utkstair.org/clausius/docs/mse301/data/xm4p02_f22.txt on the exam portion of the course website, perform a linear regression to determine the mean values of k and n for this data.
(b) Also report the standard deviations of k and n.

Solution:

(s) Using the table of data providing *y* and *t* in the file "xm4p01_f15.txt" on the exam portion of the course website, perform a linear regression to determine the mean values of *k* and *n* for this data.

I used the code linreg1.m for linear regression with one independent variable.

I wrote the small script xm4p02_f20.m

```
clear all;

M = [
0.10  5.43E-07
0.11  5.84E-07
0.12  6.58E-07
...
9.80  6.62E-01
9.90  5.61E-01
10.00 4.86E-01
    ];

n = max(size(M));
for i = 1:1:n
    x(i) = log(M(i,1));
    y(i) = log(log(1.0/(1.0-M(i,2))));
end
[b,bsd,MOF] = linreg1(x, y)
n_mean = b(2)
k_mean = exp(b(1))

n_sd = bsd(2)
k_sd = abs(bsd(1)/b(1))*k_mean
```

At the command line prompt, I executed the script

```
>> xm4p02_f22
```

This generated the following output for the means, standard deviations and Measure of Fit.

```
>> xm4p02_f22

b =
   -7.3117
    3.1955

bsd =
    0.0205
    0.0151

MOF =    0.9960

n_mean =    3.1955
```

```
k_mean =   6.6766e-04

n_sd =    0.0151

k_sd =   1.8761e-06
```

The mean value of *n* is 3.20.
The mean value of *k* is $6.68\text{x}10^{-4}$ min^{-1}.

The code also generated a plot, shown below.

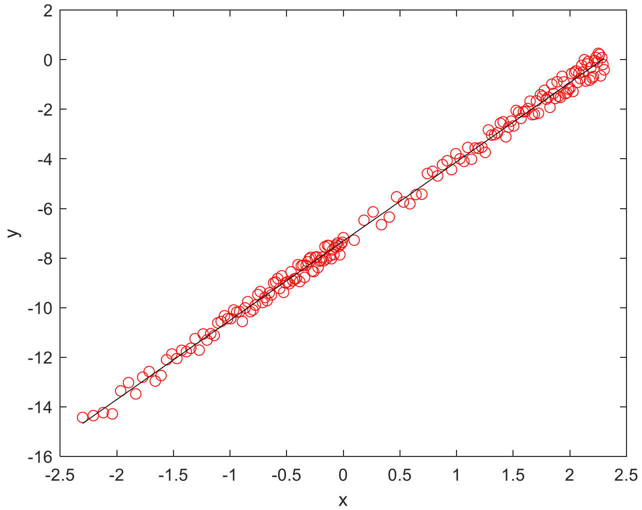
(c) Also report the standard deviations of *k* and *n*.

Since the slope was n, the standard deviation of n is simply the standard deviation of the slope. The standard deviation of *n* is 0.015.

For the standard deviation of *k*, we use the relation for the propagation of error that the relative error of *k* is equal to the relative error of the intercept.

$$\frac{s_k}{\bar{x}_k} = \left|\frac{s_b}{\bar{x}_b}\right| \text{ so } s_k = \left|\frac{s_b}{\bar{x}_b}\right|\bar{x}_k$$

The standard deviation of *k* is $1.88\text{x}10^{-6}$ min^{-1}.



Problem 3. (8 points)

In the data of a spectroscopy experiment two peaks are measured. The first peak corresponds to the abundance of compound A and the second peak corresponds to the abundance of compound B. The signal data is given in the file, http://utkstair.org/clausius/docs/mse301/data/xm4p03_f22.txt. In this data file, the first column is x, the second column contains the values of the peak 1 signal, and the third column contains the values of the peak 2 signal. perform the following tasks.

(a) What is the appropriate numerical method to integrate these signals?
(b) Find the integral of the first peak.
(c) Find the integral of the second peak.
(d) Find the relative abundance of compound A with respect to compound B.

Solution:

(a) What is the appropriate numerical method to integrate these signals?

A method for numerical integration, such as the trapezoidal rule should be used.

(b) Find the integral of the first peak.
(c) Find the integral of the second peak.
(d) Find the relative abundance of compound A with respect to compound B.

I wrote a little script in the file xm4p03_f22.m, which contained the following text

```
clear all;
close all;

datamat = [
3.00  1.4561E-04  3.4030E-05
3.01  1.3889E-04  4.3526E-05
3.02  1.3514E-04  4.9753E-05
...
11.98 1.6152E-06  7.6875E-66
11.99 1.4431E-06  4.5033E-66
12.00 1.5827E-06  3.3075E-66
];

x = datamat(:,1);
peak1 = datamat(:,2);
peak2 = datamat(:,3);
n = length(x);
dx = x(2) - x(1);

% trapezoidal rule for peak one
peak1_first = peak1(1);
peak1_last = peak1(n);
peak1_mid = 0.0;
for i = 2:1:n-1
    peak1_mid = peak1_mid + peak1(i);
end
integral_peak1 = 0.5*dx*(peak1_first + peak1_last + 2.0*peak1_mid)

% trapezoidal rule for peak one
```

```
peak2_first = peak2(1);
peak2_last = peak2(n);
peak2_mid = 0.0;
for i = 2:1:n-1
    peak2_mid = peak2_mid + peak2(i);
end
integral_peak2 = 0.5*dx*(peak2_first + peak2_last + 2.0*peak2_mid)

ratio1to2 = integral_peak1/integral_peak2

% plot optional
figure(1)
plot(x,peak1,'ro');
hold on;
plot(x,peak2,'bo');
xlabel('x');
ylabel('signal');
legend('peak 1','peak 2');
```

At the command line prompt, I typed the command

```
>> xm4p03_f22
```

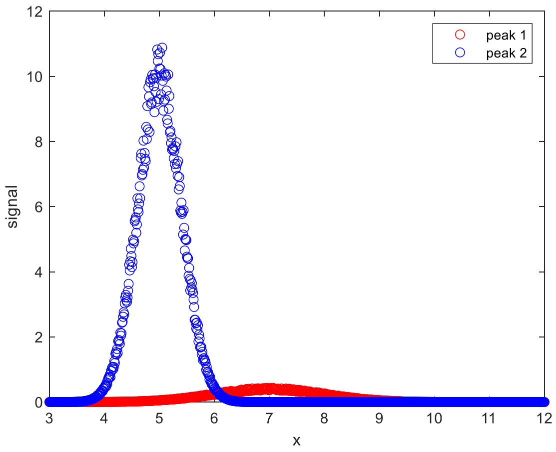
which generated the following output:

```
integral_peak1 =     0.9978
integral_peak2 =     9.9484
ratio1to2 =     0.1003
```

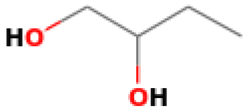
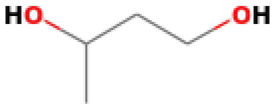
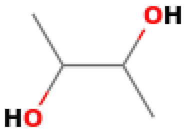
Thus, the integral of Peak 1 is 1.00.
Thus, the integral of Peak 2 is 9.95.
The relative abundance is given by the ratio of the integrals and is thus 0.10.

There is about one tenth as much compound A as compound B.

This script also generated an optional plot of the data.



Problem 4. (10 points)
Consider a mixture of four butanediol ($C_4O_2H_{10}$) isomers shown below.

			
1,2-Butanediol	1,3-Butanediol	1,4-Butanediol	2,3-Butanediol

Equilibrium coefficients, which relate mole fractions of components to one another, are given below:

$$K_{1,3-1,2}^{eq} = x_{1,3}/x_{1,2}$$

$$K_{1,4-1,3}^{eq} = x_{1,4}/x_{1,3}$$

$$K_{2,3-1,4}^{eq} = x_{2,3}/x_{1,4}$$

The sum of the mole fractions is unity.

$$x_{1,2} + x_{1,3} + x_{1,4} + x_{2,3} = 1$$

The equilibrium coefficients are evaluated using the relation

$$K_{1,3-1,2}^{eq} = exp\left(-\frac{\Delta G_{1,3-1,2}}{RT}\right) \text{ where } \Delta G_{1,3-1,2} = 1.2\ kJ/mol$$

$$K_{1,4-1,3}^{eq} = exp\left(-\frac{\Delta G_{1,4-1,3}}{RT}\right) \text{ where } \Delta G_{1,4-1,3} = 0.6\ kJ/mol$$

$$K_{2,3-1,4}^{eq} = exp\left(-\frac{\Delta G_{2,3-1,4}}{RT}\right) \text{ where } \Delta G_{2,3-1,4} = 1.8\ kJ/mol$$

where R is the gas constant, $R = 0.008314 \frac{kJ}{mol}/K$. In this problem we don't know the temperature and we only know one of the mole fractions, $x_{1,2} = 0.4$. The other mole fractions, $x_{1,3}$, $x_{1,4}$ and $x_{2,3}$ are unknown.

(a) Is this system of algebraic equations linear or nonlinear? (2 pts)
(b) Determine the temperature and the composition of the other three isomers. (8 pts) A good initial guess might be

$$\begin{bmatrix} T \\ x_{1,3} \\ x_{1,4} \\ x_{2,3} \end{bmatrix} = \begin{bmatrix} 300 \\ 0.2 \\ 0.2 \\ 0.2 \end{bmatrix}$$

Solution:

(a) Is this system of algebraic equations linear or nonlinear? (2 pts)

This is a set of non-linear algebraic equations, which can be written as follows.

$$x_{1,2}exp\left(-\frac{\Delta G_{1,3-1,2}}{RT}\right)-x_{1,3}=0$$

$$x_{1,3}xp\left(-\frac{\Delta G_{1,4-1,3}}{RT}\right)-x_{1,4}=0$$

$$x_{1,4}exp\left(-\frac{\Delta G_{2,3-1,4}}{RT}\right)-x_{2,3}=0$$

$$x_{1,2}+x_{1,3}+x_{1,4}+x_{2,3}-1=0$$

I will solve this using the Newton Raphson Method with Numerical Approximations to the Derivatives, as implemented in the code `nrndn.m`.

This code requires that I input my system of nonlinear algebraic equations in the function, `funkeval.m`.

```
function f = funkeval(x)
%
%  these two lines force a column vector of length n
%
n = max(size(x));
f = zeros(n,1);
%
%  enter the constants here
%
dG_13_12 = 1.2; % kJ/mol
dG_14_13 = 0.6; % kJ/mol
dG_23_14 = 1.8; % kJ/mol
R = 0.008314; % kJ/mol/K
x12 = 0.4;
%
% identify variables
%
T = x(1);
x13 = x(2);
x14 = x(3);
x23 = x(4);
% compute equilibrium coefficients
Keq_13_12 = exp(-dG_13_12/(R*T));
Keq_14_13 = exp(-dG_14_13/(R*T));
Keq_23_14 = exp(-dG_23_14/(R*T));
% write equations
f(1) = x12*Keq_13_12 - x13;
f(2) = x13*Keq_14_13 - x14;
f(3) = x14*Keq_23_14 - x23;
f(4) = x12 + x13 + x14 + x23 - 1.0;
```

The Newton Raphson method requires an initial guess. I used the suggested initial guess. I want the tolerance to be 1.0^{-6}. I set the print flag to 1. At the command line prompt, I executed the following commands:

```
clear all;
x0 = [300.0 0.2 0.2 0.2];
tol = 1.0e-6;
iprint = 1;
[x,err,f] = nrndn(x0,tol,iprint)
```

This command provided the following output.

```
 iter =    1, err =  2.50e+01 f =  6.05e-02
 iter =    2, err =  1.67e+00 f =  1.24e-03
 iter =    3, err =  1.34e-02 f =  5.02e-06
 iter =    4, err =  8.24e-07 f =  3.19e-10

x =  353.2863    0.2658    0.2167    0.1174

err =   8.2399e-07

f =   3.1906e-10
```

Because the error is less than the specified tolerance, the Newton Raphson method has converged. Therefore the temperature is 353.3 K and the composition of the mixture is given by

$$\underline{x} = \begin{bmatrix} x_{1,2} \\ x_{1,3} \\ x_{1,4} \\ x_{2,3} \end{bmatrix} = \begin{bmatrix} 0.4 \\ 0.266 \\ 0.217 \\ 0.117 \end{bmatrix}$$