Final Exam
Administered: Monday, December 8, 2014
8:00 AM – 10:00 AM
28 points

Problem 1. (8 points)

Consider the dimerization reaction:

$$2A \rightarrow B$$

The reaction rate is given by

$$rate = C_A^2 k_o e^{-\frac{E_a}{RT}} \qquad \text{[moles/m}^3\text{/sec]}$$

where

concentration of A: C_A [moles/m^3]
prefactor: k_o [m^3/(moles·sec)]
activation energy for reaction: E_a [Joules/mole]
constant: $R = 8.314$ [Joules/mole/K]
temperature: T [K]

Determine the rate constants, k_o and E_a, from experimental data. The reaction is measured at a constant concentration of A, C_A = 2000 mol/m^3, over a range of temperatures. The rate is recorded. The rate as a function of temperature is given in tabular form in the file "xm4p01_f14.txt" on the exam portion of the course website.

Solution:

Convert the data into a linear form necessary for a linear regression.

$$\ln(rate) - 2\ln(C_A) = -\frac{E_a}{RT} + \ln(k_o)$$

This equation is of the form: $y = b_1 x + b_0$ where

$$y = \ln(rate) - 2\ln(C_A),\ b_1 = E_a,\ x = -\frac{1}{RT},\ \text{and}\ b_0 = \ln(k_o).$$

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

I wrote the small script xm4p01_f14.m

```
clear all;

%Temperature    rate of A loss
%K              mole/m^3/s

M = [500    4.019406693
525   8.822457706
550   13.51828924
575   24.46718276
600   36.72515181];

R = 8.314; % J/mol/K
CA = 2000; % mol/m^3
n = max(size(M));
for i = 1:1:n
    x(i) = -1.0/(R*M(i,1));
    y(i) = log(M(i,2)) – 2*log(CA);
end
[b,bsd,MOF] = linreg1(x, y)
Ea = b(2)
ko = exp(b(1))
```

At the command line prompt, I executed the script

```
>> xm4p01_f14
```

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

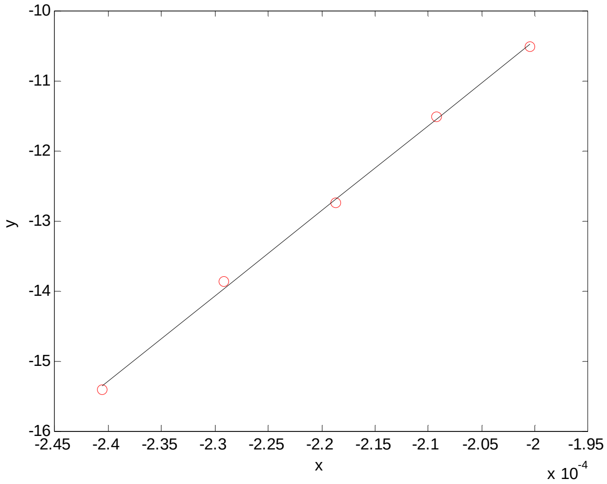
```
>> xm4p01_f14

b =   1.0e+04 *
   -0.0001
    5.4504

bsd =   1.0e+03 *
    0.0005
    2.4340

MOF =    0.9941

Ea =   5.4504e+04
ko =    0.5283
```



The activation energy was 54,500 J/mol.
The rate constant was 0.528 m^3/(mol·sec).

The code also generated a plot.

Problem 2. (6 points)

The gamma function is defined as

$$\Gamma(\alpha) = \int_0^\infty x^{\alpha-1} e^{-x} dx \quad \text{for } \alpha > 0$$

(a) Evaluate the gamma function for $\alpha = 5$ using the intrinsic gamma function. You likely will need to use the `format long` statement in MatLab to get enough digits to display.
(b) How many intervals so you need in the second-order Simpson's method to obtain this result to four significant digits? What did you use as your upper limit of integration?

Solution:

(a) Evaluate the gamma function for $\alpha = 5$ using the intrinsic gamma function.

```
>> format long
>> y = gamma(5)
y =   24
```

(b) How many intervals so you need in the second-order Simpson's method to obtain this result to four significant digits? What did you use as your upper limit of integration?

I used the simpson2.m code and changed the integrand function to

```
function f = funkeval(x)
f = x^4*exp(-x);
```

From the command line prompt.

```
>> integral = simpson2(0,100,10000)
Using the Simpsons Second Order method
to integrate from 0.000000 to 100.000000 with 10000 nintervals,
the integral is 2.400000e+01
integral =  23.999999999999794
```

So using 100 as the upper limit & 10,000 intervals gives about 14 good significant digits.

```
>> integral = simpson2(0,100,1000)
Using the Simpsons Second Order method
to integrate from 0.000000 to 100.000000 with 1000 nintervals,
the integral is 2.400000e+01
integral =  23.999999921216098
```

So using 100 as the upper limit & 1,000 intervals gives about 9 good significant digits.

```
>> integral = simpson2(0,100,100)
Using the Simpsons Second Order method
to integrate from 0.000000 to 100.000000 with 100 nintervals,
the integral is 2.396216e+01
integral =  23.962164020980257
```

So using 100 as the upper limit & 100 intervals gives about 3 good significant digits.

```
>> integral = simpson2(0,100,200)
Using the Simpsons Second Order method
to integrate from 0.000000 to 100.000000 with 200 nintervals,
the integral is 2.399897e+01
integral =  23.998968144328195
```

So using 100 as the upper limit & 200 intervals gives about 4 good significant digits.

Note, using 10 as the upper limit & 10,000 intervals gives only 1 good significant digit, indicating that 10 is not high enough for an accurate evaluation of this integral.

```
>> integral = simpson2(0,10,10000)
Using the Simpsons Second Order method
to integrate from 0.000000 to 10.000000 with 10000 nintervals,
the integral is 2.329794e+01
integral =  23.297935486152902
```

Problem 3. (8 points)

Consider the dimerization reaction:

$$2A \rightarrow B$$

The reaction rate is given by

$$rate = C_A^2 k_o e^{-\frac{E_a}{RT}} \qquad \text{[moles/m}^3\text{/sec]}$$

where

concentration of A: C_A [moles/m^3]
prefactor: k_o [m^3/(moles·sec)]
activation energy for reaction: E_a [Joules/mole]
constant: $R = 8.314$ [Joules/mole/K]
temperature: T [K]

If this reaction were to take place in a continuous stirred tank reactor, the mole balances on A and B would have the form

$$\text{accumulation} = \text{in} - \text{out} + \text{generation}$$

$$V\frac{dC_A}{dt} = C_{A,in}F - C_A F - 2rate \qquad (1)$$

$$V\frac{dC_B}{dt} = C_{B,in}F - C_B F + rate \qquad (2)$$

Equations (1) and (2) represent a set of two non-linear coupled ordinary differential equations.

Assume the following parameter values:

inlet concentration of A: $C_{A,in} = 2{,}000$ [moles/m^3]
inlet concentration of B: $C_{B,in} = 0$ [moles/m^3]
prefactor: $k_o = 0.90$ [m^3/(moles·sec)]
activation energy for reaction: $E_a = 66{,}000$ [Joules/mole]
temperature: $T = 600$ [K]
reactor volume: $V = 1$ [m^3]
flowrate: $F = 0.01$ [m^3/sec]

Also, assume that reactor initially has no A or B in it, so that the initial conditions are given by

initial concentration of A: $C_A(t=0) = 0$ [moles/m^3]
initial concentration of B: $C_B(t=0) = 0$ [moles/m^3]

Numerically solve the ordinary differential equations.
(a) Sketch the plot of the concentrations for A and B.
(b) Report values of the concentrations of A and B at time 200 sec.
(c) Report values of the concentrations of A and B at time 500 sec.
(d) Report values of the concentrations of A and B at time 1000 sec.

Solution:

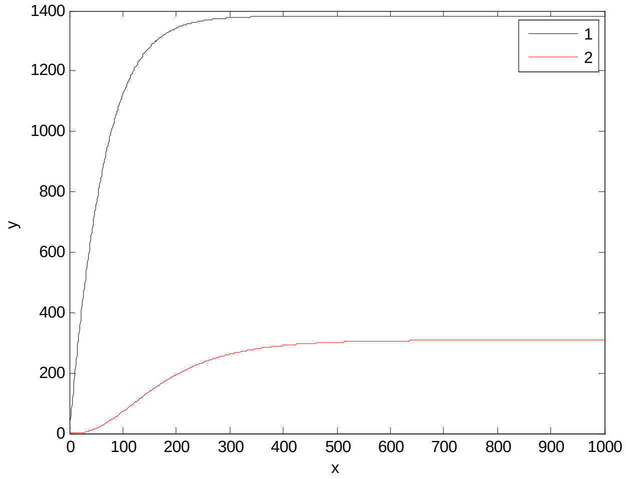
Use the routine rk4n.m. Change the input file to look like:

```
function dydx = funkeval(x,y);
CA = y(1);
CB = y(2);
CAin = 2000.0; % mol/m^3
CBin = 0.0; % mol/m^3
R = 8.314; % J/mol/K
T = 600.0; % K
Ea = 66000.0; % J/mol
ko = 0.90; % m^3/mol/sec
V = 1; % m^3
F = 0.1; % m^3/sec
rate = CA*CA*ko*exp(-Ea/(R*T));
dCAdt = 1/V*(CAin*F – CA*F -2*rate);
dCBdt = 1/V*(CBin*F – CB*F + rate);
dydx(1) = dCAdt;
dydx(2) = dCBdt;
```

We choose to solve from $x_0 = 0$ to $x_f = 1000$. We first use n = 1000 steps. At the command prompt, type:

```
>>  [x,y]=rk4n(1000,0,1000,[0,0]);
```

The code executes without any warning messages. The plot looks like



According to the legend, the black curve is y(1), the concentration of A, and red curve is y(2), the concentration of B.

(b) Report values of the concentrations of A and B at time 200 sec.
(c) Report values of the concentrations of A and B at time 500 sec.
(d) Report values of the concentrations of A and B at time 1000 sec.

Since $\Delta t = {(t_f - t_0)}/{n} = {(1000-0)}/{1000} = 1\,\text{sec}$

Therefore the index in the x vector and y matrix corresponds to time in seconds, where the first point corresponds to the initial condition and the 1001 point corresponds to t = 1000 sec.

So we can answer (b) by accessing the variables x and y as follows:

```
>> x(201)
ans =   200

>> y(201,:)
ans =   1.0e+03 *
    1.3418    0.1938
```

The concentration of A and B at t = 200 s are 1342 and 194 mol/m^3 respectively.

Similarly, we can answer (c) and (d) using an analogous approach.

```
>> x(501)
ans =   500

>> y(501,:)
ans =   1.0e+03 *
    1.3824    0.3021
```

The concentration of A and B at t = 500 s are 1382 and 302 mol/m^3 respectively.

```
>> x(1001)
ans = 1000

>> y(1001,:)
ans =   1.0e+03 *
    1.3825    0.3087
```

The concentration of A and B at t = 1000 s are 1383 and 309 mol/m^3 respectively.

Problem 4. (6 points)

Consider the system exactly as given in Problem 3. Now consider the system at steady state, where the accumulation term on the left hand sides of equation (1) and (2) are set to zero, yielding

$$0 = C_{A,in}F - C_A F - 2rate \tag{3}$$

$$0 = C_{B,in}F - C_B F + rate \tag{4}$$

Equations (3) and (4) represent a set of two non-linear algebraic equations.

Numerically solve the ordinary differential equations.
(a) State a reasonable set of initial guesses for the concentrations of A and B.
(b) Report the converged values of the concentrations of A and B.
(c) Compare these results to the results from problem 3.

Solution:
Use the Newton Raphson with Numerical Derivatives for N equations routine, nrndn.m. Change the input file to look like:

```
function f = funkeval(x)
n = max(size(x));
f = zeros(n,1);
%
CA = x(1);
CB = x(2);
CAin = 2000.0; % mol/m^3
CBin = 0.0; % mol/m^3
R = 8.314; % J/mol/K
T = 600.0; % K
Ea = 66000.0; % J/mol
ko = 0.90; % m^3/mol/sec
V = 1; % m^3
F = 0.01; % m^3/sec
rate = CA*CA*ko*exp(-Ea/(R*T));
dCAdt = 1/V*(CAin*F - CA*F -2*rate);
dCBdt = 1/V*(CBin*F - CB*F + rate);
f(1) = dCAdt;
f(2) = dCBdt;
```

Note that this input file contains the same content as that used in problem 3, but the input and output variables have changed names.

(a) State a reasonable set of initial guesses for the concentrations of A and B.

Reasonable initial guesses can be obtained from part (d) of Problem (3) because the slope is not changing at long times, so that means, the system has also reached steady state.

```
xo = [1383,309]
```

(b) Report the converged values of the concentrations of A and B.

```
>> [x,err,f] = nrndn([1383,309],1.0e-6,1)
iter =    1, err =  3.78e-01 f =  6.19e-03
 iter =    2, err =  2.86e-05 f =  5.42e-07
 iter =    3, err =  1.85e-13 f =  3.14e-15

x =   1.0e+03 *

    1.3825    0.3087


err =   1.8517e-13

f =   3.1402e-15
```

The code converged to six significant digits in three iteration to values of 1383 and 309 moles/m^3 for A and B respectively.

(c) Compare these results to the results from problem 3.

These steady state values agree with long-time values obtained from solving the ODEs. The advantage in solving the ODEs is that, in addition to reaching the steady state values, you also can see the transient behavior as it approaches steady state.