Homework Assignment Number Twelve Solutions

Problem 1.

Consider the system of non-linear ordinary differential equation:

$$\frac{dy_1}{dx} = \frac{k_{11}y_1}{k_{12}y_2} + b_1 x$$
$$\frac{dy_2}{dx} = \sqrt{k_{21}y_1 + k_{22}y_2 + b_2 x}$$

with the initial conditions

$$y_1(x=0) = y_{1o} = 1.0$$
$$y_2(x=0) = y_{2o} = 0.1$$

where

$$k = \begin{bmatrix} -0.1 & 0.1 \\ 0.03 & 0.02 \end{bmatrix} \text{ and } b = \begin{bmatrix} -0.01 \\ 0.0 \end{bmatrix}$$

(a) Determine the behavior of $y_1(x)$ and $y_2(x)$ from $0 \le x \le 20$.
(b) Determine the values of $y_1(x)$ and $y_2(x)$ at $x = 10$.

Solution:

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

```
function dydx = funkeval(x,y);
k = [-0.1, 0.1; 0.03, 0.02];
b = [-0.01; 0.0];
dydx(1) = k(1,1)*y(1) / (k(1,2)*y(2)) + b(1)*x;
dydx(2) = sqrt (k(2,1)*y(1) + k(2,2)*y(2) + b(2)*x);
```

We choose to solve from $x_0 = 0$ to $x_f = 20$. We first use n = 100 steps. At the command prompt, type:

```
>>  [x,y]=rk4n(100,0,20,[1.0,0.1]);
```

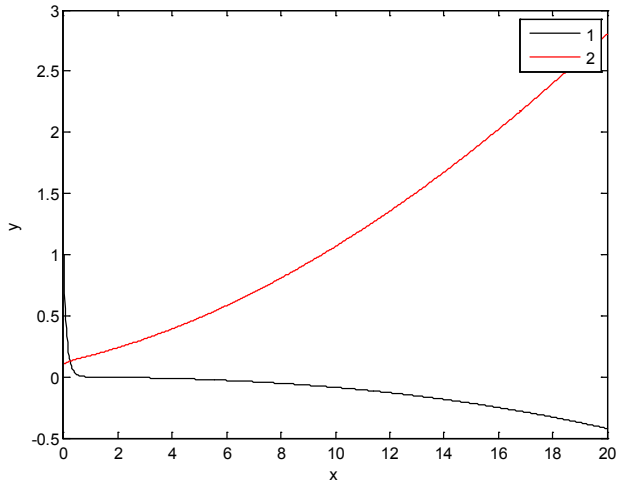
However this gives us some warnings.

```
Warning: Imaginary parts of complex X and/or Y arguments ignored
> In rk4n at 85
Warning: Imaginary parts of complex X and/or Y arguments ignored
> In rk4n at 85
```

So we choose to solve more intervals, n = 1000 steps. At the command prompt, type:

```
>>  [x,y]=rk4n(1000,0,20,[1.0,0.1]);
```

This time we have no warning messages. The plot looks like



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

(b) Determine the values of $y_1(x)$ and $y_2(x)$ at $x = 10$.

There are different ways to do this. We could remove the semicolon and look through the tabular data of x and y. Alternatively, we could simply stop the calculation at 10. If we want the exact answer that appears in the plot, then we need to use the same interval size. The interval size in the plot is $\Delta x = (x_f - x_0)/n = (20-0)/1000 = 0.02$. So, if we want the simulation to end at 10 with the same interval size we need $n = (x_f - x_0)/\Delta x = (10-0)/0.02 = 500$ intervals.

```
>>  [x,y]=rk4n(500,0,10,[1.0,0.1]);
```

If there are 500 intervals, there are 501 points. The last point corresponds to x = 10. We can confirm this as follows.

```
>> x(501)
ans =     10
```

The values of y at x = 10 are therefore

```
>> y(501,:)
ans =   -0.0862    1.0637
```

Problem 2. (12 points)
We are testing a polymer membrane designed to catalytically filter microbes. The concentration of microbe A, $C_{A,0}$, on one side of the membrane, located at $z = 0$, is $C_{A,0}$. The concentration of microbe A, $C_{A,f}$, on the other side of the membrane, located at $z = L$, is $C_{A,f}$. Inside the polymer membrane, a chemical agent kills the microbe. The following equation describes the profile of the microbe concentration within the membrane,

$$0 = D\frac{d^2C_A}{dz^2} - k\sqrt{C_A}$$

Answer the following questions.

(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) For a membrane of thickness, $L = 5$ cm, and the following numerical values, $D = 1.0 \cdot 10^{-6} \frac{cm^2}{s}$, $k = 2.8 \cdot 10^{-9} \frac{\left(\frac{mol}{\ell}\right)^{1/2}}{s}$, $C_{A,0} = 1.0 \cdot 10^{-2} \frac{mol}{\ell}$ and $C_{A,f} = 2.2 \cdot 10^{-4} \frac{mol}{\ell}$, find the concentration gradient of the microbe at $z = 0$.
(e) Sketch the concentration profile.
(f) Verify that your discretization resolution was sufficient.

Solution

(a) Is this ODE problem linear or nonlinear?

The problem is nonlinear due to the presence of the square root of the concentration.

(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.

$$y_1 = C_A \qquad\qquad y_2 = \frac{dC_A}{dz}$$

Step 2. Write ODEs for the new variables.

$$\frac{dy_1}{dz} = y_2 \qquad \frac{dy_2}{dz} = \frac{k}{D}\sqrt{y_1} \quad \left(\text{from } 0 = D\frac{dy_2}{dz} - k\sqrt{y_1}\right)$$

Step 3. Write initial conditions for the new variables.

$$y_1(z=0) = C_{A,0} \qquad y_2(z=0) = \left.\frac{dC_A}{dz}\right|_0$$

(d) For a membrane of thickness, L = 5 cm, and the following numerical values, $D = 1.0 \cdot 10^{-6} \frac{cm^2}{s}$, $k = 2.8 \cdot 10^{-9} \frac{\left(\frac{mol}{\ell}\right)^{1/2}}{s}$, $C_{A,0} = 1.0 \cdot 10^{-2} \frac{mol}{\ell}$ and $C_{A,f} = 2.2 \cdot 10^{-4} \frac{mol}{\ell}$, find the gradient of the concentration at z = 0, $\left.\frac{dC_A}{dz}\right|_0$.

(e) Sketch the concentration profile.

(f) Verify that your discretization resolution was sufficient.

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);
D = 1.0e-6; % cm^2/s
k = 2.8e-09; % (mol/l)^(1/2)/s
dydx(1) = y(2);
dydx(2) = k/D*sqrt(y(1));
```

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

```
function f = funkeval(x)
xo = 0;
yo_1 = 1.0e-2;
yo_2 = x;
xf = 5;
yf = 2.2e-4;
n = 1000;
[x,y]=rk4n(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 average slope as my initial guess.

```
>> average_slope = (2.2e-4 - 1.0e-2)/5
average_slope =   -0.0020

>> [x0,err] = nrnd1(average_slope)
```

This command generated the following output:

```
icount = 1 xold = -1.956000e-03 f = 2.961144e-03 df = 5.393071e+00 xnew = -2.505065e-03  err =
1.000000e+02
icount = 2 xold = -2.505065e-03 f = -2.274794e-05 df = 5.496912e+00 xnew = -2.500926e-03  err =
1.654712e-03
icount = 3 xold = -2.500926e-03 f = -4.690321e-09 df = 5.495251e+00 xnew = -2.500925e-03  err =
3.412827e-07

x0 =  -0.002500925398199
err =     3.412827042024479e-07
```

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

$$\left.\frac{dC_A}{dz}\right|_0 = -0.0025 \frac{\text{mol}}{\ell \cdot \text{cm}}$$

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

```
>> [x,y]=rk4n(1000,0,5,[1.0e-2,-0.002500925398199]);
>> y_n1000  = y(1001,1)
y_n1000  =  2.199999997600626e-04
```

When we use 1000 intervals in the Runge Kutta code, we obtain a concentration at the far end of the membrane as

$$C_A(z=L) = 2.2 \cdot 10^{-4} \frac{mol}{\ell}$$

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

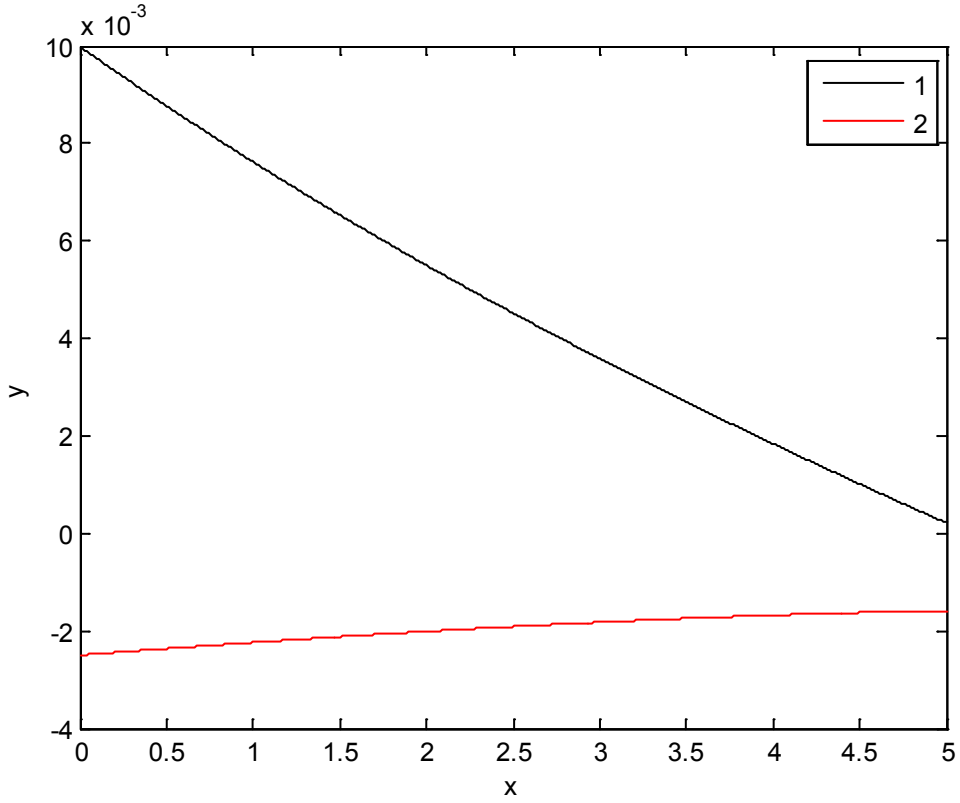
```
>> [x,y]=rk4n(10000,0,5,[1.0e-2,-0.002500925398199]);
>> y_n10000 = y(10001,1)
y_n10000 =  2.199999997597912e-04
```

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

So the concentration on the far side of the membrane is

$$C_A(z=L) = 2.2 \cdot 10^{-4} \frac{mol}{\ell}$$

A plot of the profile is shown below. The black line marked "1" is the concentration of A. The red line marked "2" is the concentration gradient.


x 10^-3
10
8
6
4
2
0
-2
-4
y
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
x
1
2