10/4/2022

Homework 3

BME 7410

Submitted by,

Rana Mozumder

DEPARTMENT OF BME, VANDERBILT UNIVERSITY

5.3 Saturation of oxygen in water:

@ For bisection method, if the initial guesses are at a, b, to reach the minimum tolerance level, Tol(x), we need need minimum of n iterations which can be given by,

$$\frac{|b-a|}{2^n} \leq Tol(x)$$

$$\Rightarrow \frac{|b-a|}{Tol(x)} \leq 2^n$$

$$\Rightarrow \log \frac{|b-a|}{Tol(x)} \leq n \log 2$$

$$\Rightarrow n \geq \frac{\log \frac{|b-a|}{Tol(x)}}{\log 2}$$

Here, for $a = 0°c$, $b = 35°c$ and $Tol(x) = 0.05°c$,

$$n \geq \frac{\log \frac{|35-0|}{0.05}}{\log 2}$$

$$\Rightarrow n \geq 9.45$$

So, we could need at least 10 iterations to reach the tolerance level of 0.05°c.

(b) Here the function would be,

$$f(T_a) = -139.34411 + \frac{1.575701 \times 10^5}{T_a} - \frac{6.642308 \times 10^7}{T_a^2}$$

$$+ \frac{1.243800 \times 10^{10}}{T_a^3} - \frac{8.621949 \times 10^{11}}{T_a^4} - \ln(sow)$$

Now, we solve it for sow = 8, 10, 14 ;

initial guesses, [a b] = [0 35]

and Tol (Ta) = Tol (f (Ta)) = 0.05

The corresponding MatLab scripts and results are shown below –

Matlab Scripts:

```
function f = Sat_T(x)

%Variables
%x=temperature in Kelvin
%Sow=Saturation of oxygen in water

%converting the temperature from degree celcius to Kelvin
x=x+273.15;

%determining higher orders of temperature
x2 = x^2;
x3 = x^3;
x4 = x^4;

Sow = 8;%need to change the value to 10 and 14 for other two cases

%function
f = -139.34411 + 1.575701*1e5/x -6.642308*1e7/x2 + 1.243800*1e10/x3 ...
    -8.621949*1e11/x4 -log(Sow);


function bisectionmethod(func,ab, tolx, tolfx)
% Bisection algorithm used to solve a nonlinear equation in x

% Input variables
% func : non-linear function
% ab   : bracketing interval [a, b]
% tolx : tolerance for error in estimating root
% tolfx: tolerance for error in function value at solution

% Other variables
maxloops = 50; % maximum number of iterations allowed

% Root-containing interval [a b]
a = ab(1);
b = ab(2);
fa = feval(func, a); % feval evaluates func at point a
fb = feval(func, b); % feval evaluates func at point b
% Min number of iterations required to meet criterion of Tolx
minloops = ceil(log(abs(b-a)/tolx)/log(2));
         % ceil rounds towards +Inf

fprintf('Min iterations for reaching convergence = %2d \n',minloops)
fprintf(' i     x      f(x)  \n'); % \n is carriage return

% Iterative solution scheme
for i = 1:maxloops
    x = (a+b)/2; % mid-point of interval
    fx = feval(func,x);
    fprintf('%3d %5.4f %5.4f \n',i,x,fx);
    if (i >= minloops && abs(fx) < tolfx)
        break % Jump out of the for loop
    end
```

```
    if (fx*fa < 0)  % [a x] contains root
        fb = fx; b = x;
    else            % [x b] contains root
        fa = fx; a = x;
    end
end
```

Results:

For Sow=8;

>> bisectionmethod('Sat_T', [0 35], 0.05, 0.05)

Min iterations for reaching convergence = 10

i	x	f(x)
1	17.5000	0.1787
2	26.2500	0.0096
3	30.6250	-0.0675
4	28.4375	-0.0295
5	27.3438	-0.0101
6	26.7969	-0.0003
7	26.5234	0.0046
8	26.6602	0.0022
9	26.7285	0.0009
10	26.7627	0.0003

For Sow=10;

>> bisectionmethod('Sat_T', [0 35], 0.05, 0.05)

Min iterations for reaching convergence = 10

i	x	f(x)
1	17.5000	-0.0445
2	8.7500	0.1510
3	13.1250	0.0495
4	15.3125	0.0016
5	16.4062	-0.0216
6	15.8594	-0.0101
7	15.5859	-0.0042
8	15.4492	-0.0013
9	15.3809	0.0002
10	15.4150	-0.0006

For Sow=14;

>> bisectionmethod('Sat_T', [0 35], 0.05, 0.05)

Min iterations for reaching convergence = 10

i	x	f(x)
1	17.5000	-0.3809
2	8.7500	-0.1855
3	4.3750	-0.0756
4	2.1875	-0.0173
5	1.0938	0.0127
6	1.6406	-0.0024
7	1.3672	0.0052
8	1.5039	0.0014
9	1.5723	-0.0005
10	1.5381	0.0005

As predicted in ⓐ, each of the three trials took 10 iterations to converge within the tolerance limit.

From the equation, if the temperature raises, the saturation falls. That's proved from the results as well.

for $SOW = 14$ mg/L, $T_a = 1.5381°C$

for $SOW = 10$ mg/L, $T_a = 15.4150°C$

for $SOW = 8$ mg/L, $T_a = 26.7627°C$

So, the reverse order is very clear.

5.6 Osteoporosis in Chinese women:

The relation between speed of sound (sos) and age is given by.

$$SOS = 3383 + 39.9Y - 0.78Y^2 + 0.0039Y^3$$

where, SOS = speed of sound

Y = age of women

Now, for applying Newton's method, we need to build an error function and minimize it to get closer to the root.

$$f(x) = 3383 + 39.9x - 0.78x^2 + 0.0039x^3 - SOS$$

here, x = age of women.

$$\therefore f'(x) = 39.9 - 0.78 \times 2x + 0.0039 \times 3x^2$$

We will solve ~~if~~ it for SOS = 3850 ms^{-1} and with initial guess $x = 45$ yrs. The tolerances are taken to be 0.1.

The corresponding Matlab scripts and results are shown below.

Matlab Scripts:

```
function [fx, fxderiv] = SOS_Age(x)
%variables
%sos=speed of sound
%x=age of women

%we are looking for the age of a patient whose sos=3850 m/s
sos=3850;

%the function to minimize
fx = 3383 + 39.9*x -0.78*x^2 +0.0039*x^3-sos;

%and its derivative
fxderiv = 39.9 -0.78*2*x + 0.0039*3*x^2;


function newtonsmethod(func, x0, tolx, tolfx)
% Newton's method used to solve nonlinear equation in x

% Input variables
% func : nonlinear function
% x0   : initial guess value
% tolx : tolerance for error in estimating root
% tolfx: tolerance for error in function value at solution

% Other variables
maxloops = 20;

[fx, fxderiv] = feval(func,x0);
fprintf(' i   x(i)    f(x(i))    f''(x(i)) \n');
% Iterative solution scheme
for i = 1:maxloops
    x1 = x0 - fx/fxderiv;
    [fx, fxderiv] = feval(func,x1);
    fprintf('%2d  %5.4f  %7.6f  %7.6f \n',i,x1,fx,fxderiv);
    if (abs(x1 - x0) <=tolx && abs(fx) < tolfx)
        break % Jump out of the for loop
    end
    x0 = x1;
end
>> newtonsmethod('SOS_Age', 45, 0.1, 0.1)
```

i	x(i)	f(x(i))	f'(x(i))
1	60.7983	-47.892450	-11.697083
2	56.7039	-1.418701	-10.938706
3	56.5742	-0.001969	-10.908274
4	56.5741	-0.000000	-10.908231

So, the woman's most likely age would be 56.5741 yrs.