**Lab notes**

In this lab, you will be developing 3 VB codes. You are pretty much on your own to build these codes. The codes are:

1) Bisection method (detailed notes and algorithm are given below, please read this before you come to the lab lecture).

2) Direct substitution method (you should have covered this in the class)

3) Newton Raphson method (covered this in the lab). Make sure you use two subprograms—one for the function and one for its derivative when coding this. Here is the template to use

Input xold guess, iter (no of iterations), and tol (tolerance level)

For i = 1 to iter
    Call myfx(x, fx)
    Call mydfx(x, dfx)
    xnew = xold – (fx/dfx)
    err = abs((xnew-xold)/xold)*100
    output xnew, fx, dfx, and error
    If (err < tol) Then
        Exit for
    End if
Next i

**Write your myfx sub here **

**Write your mydfx sub here**

**Root finding using the Bisection Method**

One of the basic numerical approaches to find the root of a nonlinear equation $f(x)$ is the bisection method. The method can be derived from a graphical point of view. Consider a function $f(x)$ shown in Figure 1. By definition, the root of the equation $f(x)$ is nothing but the value of $x$ when the function $f(x)$ becomes zero.

As shown in the figure, to employ bisection method the user should provide two initial guess values $X_L$ and $X_U$ that bound the root of the function. Note, the root need not be at the midpoint, but it has to be in-between these two guess values. The method uses these two guess values to iteratively search for the root in a systematic manner.
Before we learn the search algorithm, we will consider the following question: how can one verify whether a root is present in-between the two arbitrary guess values $X_L$ and $X_U$. Figure 2 shows a case where the user has provided two valid guess values. Under this condition, as shown in the figure, the value of function corresponding to $X_L$, which is $f(X_L)$ will be negative, and the value of function corresponding to $X_U$, which is $f(X_U)$, will be positive, hence the value of the product $f(X_L) \cdot f(X_U)$ will be negative, therefore $f(X_L) \cdot f(X_U) < 0$.

On the other hand, Figure 3 shows two instances where the user had provided invalid guesses for $X_L$ and $X_U$ values, which are not bounding a root. Under this condition, as shown in the figures, the value of the two functions $f(X_L)$ and $f(X_U)$, corresponding to $X_L$ and $X_U$, respectively, will be of the same sign (either negative or positive); therefore, the value of the product $f(X_L) \cdot f(X_U)$ will always be positive, hence $f(X_L) \cdot f(X_U) > 0$.

Therefore, it can be concluded that the equation $f(x) = 0$, where $f(x)$ is a real continuous function, has a root between $X_L$ and $X_U$ if the product $f(X_L) \cdot f(X_U) < 0$. On the other hand, if $f(X_L) \cdot f(X_U) > 0$ then there may not be a root between the two guess values. The validity of above arguments is the central basis of the bisection method. Note, these arguments also assume that the function is monotonic and has only one root between $X_L$ and $X_U$. 

Figure 1  The root should be bound by the two initial guess values
Logic for Checking the Validity of the Initial Guesses for the Bisection Method

The concepts illustrated in Figures 2 and 3 can be summarized in a computer algorithm to first identify whether the guess values are bracketing the roots or not. The algorithm can be written as:
1. Read the values of \( X_L \) and \( X_U \) from the spreadsheet

2. Compute \( f(X_L) \) and \( f(X_U) \) and evaluate the value of their product \( P = f(X_L) \times f(X_U) \)

3. If \( P < 0 \) then it is a valid guess and the root lies between \( X_L \) and \( X_U \), else stop the program and prompt the user “invalid guesses, provide another values.”

**The Details of the Bisection Procedure**

Now that we have verified (using the above algorithm) that a root is present somewhere in the region between the guess points \( X_L \) and \( X_U \), we will first find a mid-point, \( X_M = (X_L + X_U)/2 \). We will use this midpoint to divide the region into two new intervals: 1) \((X_L \text{ and } X_M)\) and 2) \((X_M \text{ and } X_U)\). Now we need to figure out whether the root is between \( X_L \) and \( X_M \) or \( X_M \) and \( X_U \)? We know for sure that one of these intervals is a valid interval that can be used as guess values of the next iteration and the other one should be abandoned. Well, we will once again employ the logic discussed in steps 2 & 3. We will first find the sign of \( f(X_L) \times f(X_M) \) and if it is negative [or if \( f(X_L) \times f(X_U) < 0 \)] then the root should be in between \( X_L \) and \( X_M \), otherwise, it has to be between \( X_M \) and \( X_U \). Therefore, with a single test we can narrow the original range by half. By repeating this process, the width of the interval can be made smaller and smaller and we can eventually find the root. Figure 4 shows conceptual diagram of the progression of this search algorithm. As show in the figure, initially (before iteration-1) we start with \( X_L \) and \( X_U \), which envelopes a wide range. With every iteration, the range will become smaller and smaller and eventually you will land on the exact root. Note, as the solution converge the difference between the values of \( X_L \) and \( X_U \) will be small and also the value of the function \( f(x) \) will be close to zero. You can use one of them as a criterion to check for convergence.

![Figure 4: Progression of bisection algorithm.](image)

**Detailed Computer Algorithm**

The complete computer algorithm for the bisection method can be summarized as follows:
1. Read the values of XL and XU from the spreadsheet
2. Compute \( f(X_L) \) and \( f(X_U) \) and evaluate the product \( P_{test} = f(X_L) \ast f(X_U) \). Note we can use a subprogram to efficiently compute the function values.
3. If \( P_{test} < 0 \) then we have valid guess values and the root lies between the original guess values \( X_L \) and \( X_U \), else stop the program and ask them to provide another guess values.
4. Iterate \( n \) time (where \( n \) is the number of iterations given by the user)
5. Estimate \( X_M = (X_L + X_U)/2 \)
6. Evaluate \( P = f(X_L) \ast f(X_M) \)
   IF \( (P < 0) \), Then (note the root must be between \( X_L \) and \( X_M \))
   Set \( X_U = X_M \)
   ELSE
   Set \( X_L = X_M \)
   END IF
7. IF \( \text{ABS}[f(X_M)] < 0.001 \) (some small value) you have found the root so exit iteration
8. Iterate and go to step 4
9. Output the latest value of \( X_M \) as your best estimate of the root

**PROBLEMS**
1) Find the root of the following non-linear equation and evaluate the value of \( x \) using the Bisection method

\[
\begin{align*}
 f(x) &= -40 + \frac{667}{x}[1 - \exp(-0.15x)] \\
\end{align*}
\]

Perform five iterations by hand and assume initial guess value of for \( X_L = 12 \) and \( X_U = 16 \). Report the updated new values of \( X_L \) and \( X_U \) at the end of each of the iteration.

2) Plot the results of your hand calculations \( X_L \), \( X_U \) and \( X_M \) values for each iteration and develop figures similar to Figures 4 and explain the working of the convergence process for this test problem.

2) Develop VB program for solving the above problem using the Bisection method. Systematically output the \( X_L \) and \( X_U \) values at the end of each of the iteration and verify the values against the hand calculated results.

**Initial guess of 12 and 16**

<table>
<thead>
<tr>
<th>Iter #</th>
<th>( X_L )</th>
<th>( f(X_L) )</th>
<th>( X_U )</th>
<th>( f(X_U) )</th>
<th>( X_M )</th>
<th>( f(X_M) )</th>
<th>( P )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12</td>
<td>-2.0943</td>
<td>16</td>
<td>-2.0943</td>
<td>14</td>
<td>1.808683</td>
<td>11.56738</td>
</tr>
<tr>
<td>2</td>
<td>14</td>
<td>-2.0943</td>
<td>16</td>
<td>-2.0943</td>
<td>15</td>
<td>-0.22009</td>
<td>-0.39806</td>
</tr>
<tr>
<td>3</td>
<td>14.5</td>
<td>-0.22009</td>
<td>15</td>
<td>-0.22009</td>
<td>14.5</td>
<td>0.774025</td>
<td>1.399966</td>
</tr>
<tr>
<td>4</td>
<td>14.75</td>
<td>-0.22009</td>
<td>15</td>
<td>-0.22009</td>
<td>14.875</td>
<td>0.272025</td>
<td>0.210554</td>
</tr>
<tr>
<td>5</td>
<td>14.875</td>
<td>-0.22009</td>
<td>15</td>
<td>-0.22009</td>
<td>14.9375</td>
<td>0.024748</td>
<td>0.006732</td>
</tr>
<tr>
<td>6</td>
<td>14.875</td>
<td>-0.09797</td>
<td>14.9375</td>
<td>-0.09797</td>
<td>14.90625</td>
<td>-0.03669</td>
<td>-0.00091</td>
</tr>
<tr>
<td>7</td>
<td>14.875</td>
<td>-0.03669</td>
<td>14.90625</td>
<td>-0.03669</td>
<td>14.89063</td>
<td>-0.00599</td>
<td>-0.00015</td>
</tr>
<tr>
<td>8</td>
<td>14.875</td>
<td>-0.03669</td>
<td>14.90625</td>
<td>-0.03669</td>
<td>14.89063</td>
<td>-0.00599</td>
<td>-0.00015</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>-----</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>14.875</td>
<td>-0.00599</td>
<td>14.89063</td>
<td>-0.00599</td>
<td>14.88281</td>
<td>0.009375</td>
<td>0.000232</td>
</tr>
<tr>
<td>10</td>
<td>14.88281</td>
<td>-0.00599</td>
<td>14.89063</td>
<td>-0.00599</td>
<td>14.88672</td>
<td>0.001692</td>
<td>1.59E-05</td>
</tr>
<tr>
<td>11</td>
<td>14.88672</td>
<td>-0.00599</td>
<td>14.89063</td>
<td>-0.00599</td>
<td>14.88867</td>
<td>-0.00215</td>
<td>-3.6E-06</td>
</tr>
<tr>
<td>12</td>
<td>14.88672</td>
<td>-0.00215</td>
<td>14.88867</td>
<td>-0.00215</td>
<td>14.8877</td>
<td>-0.00023</td>
<td></td>
</tr>
</tbody>
</table>