Numerical Methods I

Roots of Equations: Bracketing Methods

The Bisection Method

```
program main
    implicit none

    real, external :: getRootWithBisection, getRootWithFalsePosition,
getRootWithFalsePositionModified

    real :: xLow, xHigh
    real, parameter :: tolerance = 1.0e-5
    real :: rootB, rootFP, rootFPM

    xLow = 2.0
    xHigh = 0.5
    rootB = getRootWithBisection(xLow, xHigh, tolerance)

    xLow = 2.0
    xHigh = 0.5
    rootFP = getRootWithFalsePosition(xLow, xHigh, tolerance)

    xLow = 2.0
    xHigh = 0.5
    rootFPM = getRootWithFalsePositionModified(xLow, xHigh, tolerance)

    write(*,20) "Root with Bisection = ", rootB
    write(*,20) "Root with False Position = ", rootFP
    write(*,20) "Root with Modified False Position = ", rootFPM
    20 format(a36, f9.4)
end program main
```

```
real function getRootWithBisection(xLow, xHigh, tolerance)
    implicit none

    real, external :: getFunction
    logical, external :: haveOppositeSigns

    real, intent(inout) :: xLow, xHigh
    real, intent(in) :: tolerance

    real :: error
    integer :: iteration

    real :: fLow, fHigh
    real :: xMid, fMid

    write(*,*)
    write(*,*) "Bisection Method"
    write(*,*)

    if(xLow > xHigh) then
        write(*,*) "The lower bound is greater than the upper bound.  Swapping..."
        write(*,*)
        call swapNumbers(xLow, xHigh)
    end if

    fLow = getFunction(xLow)
    fHigh = getFunction(xHigh)
    if(fLow == 0) then
        write(*,*) "The initial guess for xLow is the exact root."
        write(*,*)
        getRootWithBisection = xLow
        return
    end if
    if(fHigh == 0) then
        write(*,*) "The initial guess for xHigh is the exact root."
        write(*,*)
        getRootWithBisection = xHigh
        return
    end if

    if(haveOppositeSigns(fLow, fHigh) .eqv. .false.) then
        stop "Bisection Method: Bounds don't bracket the root."
    end if

    error = abs((xHigh - xLow) / 2)
    iteration = 0

    xMid = (xLow + xHigh) / 2
    fMid = getFunction(xMid)

    write(*,10) " | ", "No.", " | ", "xLow", " | ", "xHigh", " | ", "fLow", " | ",
"xMid", " | ", "fMid", " | ", "Error", " | "
    do while(error > tolerance)
        iteration = iteration + 1
        xMid = (xLow + xHigh) / 2
        fMid = getFunction(xMid)

        write(*,20) " | ", iteration, " | ", xLow, " | ", xHigh, " | ", fLow, " | ",
xMid, " | ", fMid, " | ", error, " | "

        if(haveOppositeSigns(fLow, fMid)) then
            xHigh = xMid
            fHigh = fMid
        else if(haveOppositeSigns(fHigh, fMid)) then
            xLow = xMid
            fLow = fMid
        else
            write(*,*) "Exact Root Found"
            write(*,*)
            exit
        end if
        error = (xHigh - xLow) / 2
    end do

    write(*,*)
    write(*,*) "Error (Bisection) = ", error
    write(*,*)
    10 format(a3, a4, a3, a7, a3, a7, a3, a7, a3, a7, a3, a7, a3, a7, a3)
    20 format(a3, i4, a3, f7.2, a3, f7.2, a3, f7.2, a3, f7.2, a3, f7.2, a3, f7.2, a3)
    getRootWithBisection = xMid
end function getRootWithBisection
```

The False Position Method

```fortran
real function getRootWithFalsePosition(xLow, xHigh, tolerance)
    implicit none

    real, external :: getFunction
    logical, external :: haveOppositeSigns

    real, intent(inout) :: xLow, xHigh
    real, intent(in) :: tolerance

    real :: fLow, fHigh
    real :: xIntercept, fIntercept
    real :: xInterceptPrevious = 0.0

    real :: error

    integer :: iteration = 0

    write(*,*)
    write(*,*) "False Position Method"
    write(*,*)

    if(xLow > xHigh) then
        write(*,*) "The lower bound is greater than the upper bound.  Swapping..."
        write(*,*)
        call swapNumbers(xLow, xHigh)
    end if

    fLow = getFunction(xLow)
    fHigh = getFunction(xHigh)

    if(fLow == 0) then
        write(*,*) "The initial guess for xLow is the exact root."
        write(*,*)
        getRootWithFalsePosition = xLow
        return
    end if
    if(fHigh == 0) then
        write(*,*) "The initial guess for xHigh is the exact root."
        write(*,*)
        getRootWithFalsePosition = xHigh
        return
    end if

    if(haveOppositeSigns(fLow, fHigh) .eqv. .false.) then
        stop "Error (False Position): Initial guesses don't bracket the root."
    end if

    write(*,10) "|", "No.", "|", "xLow", "|", "xHigh", "|", "xIntercept", "|"
    do while((error > tolerance) .or. (iteration <= 2))
        iteration = iteration + 1
        xIntercept = xHigh - (fHigh * ((xHigh - xLow) / (fHigh - fLow)))
        fIntercept = getFunction(xIntercept)

        write(*,20) "|", iteration, "|", xLow, "|", xHigh, "|", xIntercept, "|"

        if(haveOppositeSigns(fLow, fIntercept)) then
            xHigh = xIntercept
            fHigh = fIntercept
        else if(haveOppositeSigns(fHigh, fIntercept)) then
            xLow = xIntercept
            fLow = fIntercept
        else
            write(*,*) "Exact Root Found"
            write(*,*)
            exit
        end if

        error = abs(xIntercept - xInterceptPrevious)
        xInterceptPrevious = xIntercept
    end do

    write(*,*)
    write(*,*) "Error (False Position) = ", error
    write(*,*)
    10 format(a3, a10, a3, a10, a3, a10, a3, a10, a3)
    20 format(a3, i10, a3, f10.4, a3, f10.4, a3, f10.4, a3)

    getRootWithFalsePosition = xIntercept
end function getRootWithFalsePosition
```

The Modified False Position Method

```
real function getRootWithFalsePositionModified(xLow, xHigh, tolerance)
    implicit none

    real, external :: getFunction
    logical, external :: haveOppositeSigns

    real, intent(inout) :: xLow, xHigh
    real, intent(in) :: tolerance

    real :: fLow, fHigh
    real :: xIntercept, fIntercept
    real :: xInterceptPrevious = 0

    real :: error

    integer :: iteration = 0

    integer :: xLowMoves = 0, xHighMoves = 0

    write(*,*)
    write(*,*) "Modified False Position Method"
    write(*,*)

    if(xLow > xHigh) then
        write(*,*) "The lower bound is greater than the upper bound.  Swapping..."
        write(*,*)
        call swapNumbers(xLow, xHigh)
    end if

    fLow = getFunction(xLow)
    fHigh = getFunction(xHigh)

    if(fLow == 0) then
        write(*,*) "The initial guess for xLow is the exact root."
        write(*,*)
        getRootWithFalsePositionModified = xLow
        return
    end if
    if(fHigh == 0) then
        write(*,*) "The initial guess for xHigh is the exact root."
        write(*,*)
        getRootWithFalsePositionModified = xHigh
        return
    end if

    if(haveOppositeSigns(fLow, fHigh) .eqv. .false.) then
        stop "Error (Modified False Position): Initial guesses don't bracket the
root."
    end if

    write(*,10) "|", "No.", "|", "xLow", "|", "xHigh", "|", "xIntercept", "|"
    do while((error > tolerance) .or. (iteration <= 2))
        iteration = iteration + 1

        xIntercept = xHigh - (fHigh * ((xHigh - xLow) / (fHigh - fLow)))
        fIntercept = getFunction(xIntercept)

        write(*,20) "|", iteration, "|", xLow, "|", xHigh, "|", xIntercept, "|"

        if(haveOppositeSigns(fLow, fIntercept)) then
            xHigh = xIntercept
            fHigh = fIntercept
            xHighMoves = xHighMoves + 1
            xLowMoves = 0
        else if(haveOppositeSigns(fHigh, fIntercept)) then
            xLow = xIntercept
            fLow = fIntercept
            xLowMoves = xLowMoves + 1
            xHighMoves = 0
        else
            write(*,*) "Exact Root Found"
            write(*,*)
            exit
        end if

        error = abs(xIntercept - xInterceptPrevious)
        xInterceptPrevious = xIntercept

        if(xLowMoves == 2) then
            write(*,*) "The upper bound has not moved in the last two iterations.
Modifying fHigh..."
            write(*,*)
            fHigh = fHigh / 2
            xLowMoves = 0
        end if
        if(xHighMoves == 2) then
            write(*,*) "The lower bound has not moved in the last two iterations.
Modifying fLow..."
            write(*,*)
            fLow = fLow / 2
            xHighMoves = 0
        end if
    end do

    write(*,*)
    write(*,*) "Error (Modified False Position) = ", error
    write(*,*)
    10 format(a3, a10, a3, a10, a3, a10, a3, a10, a3)
    20 format(a3, i10, a3, f10.4, a3, f10.4, a3, f10.4, a3)

    getRootWithFalsePositionModified = xIntercept
end function getRootWithFalsePositionModified
```

Common Procedures

```fortran
real function getFunction(x)
    implicit none

    real, intent(in) :: x

    getFunction = (x**10) - 1
!    getFunction = (3 * x) + sin(x) - exp(x)
end function getFunction



logical function haveOppositeSigns(number1, number2)
    implicit none

    real, intent(in) :: number1, number2

    haveOppositeSigns = .false.

    if((number1 * number2) < 0) then
        haveOppositeSigns = .true.
    end if
end function haveOppositeSigns


subroutine swapNumbers(number1, number2)
    implicit none

    real, intent(inout) :: number1, number2

    real :: swapper

    swapper = number1
    number1 = number2
    number2 = swapper
end subroutine swapNumbers
```

Output

```
Bisection Method

The lower bound is greater than the upper bound.  Swapping...

|  No. |     xLow |    xHigh |     fLow |     xMid |     fMid |    Error |
|    1 |     0.50 |     2.00 |    -1.00 |     1.25 |     8.31 |     0.75 |
|    2 |     0.50 |     1.25 |    -1.00 |     0.88 |    -0.74 |     0.38 |
|    3 |     0.88 |     1.25 |    -0.74 |     1.06 |     0.83 |     0.19 |
|    4 |     0.88 |     1.06 |    -0.74 |     0.97 |    -0.27 |     0.09 |
|    5 |     0.97 |     1.06 |    -0.27 |     1.02 |     0.17 |     0.05 |
|    6 |     0.97 |     1.02 |    -0.27 |     0.99 |    -0.08 |     0.02 |
|    7 |     0.99 |     1.02 |    -0.08 |     1.00 |     0.04 |     0.01 |
|    8 |     0.99 |     1.00 |    -0.08 |     1.00 |    -0.02 |     0.01 |
|    9 |     1.00 |     1.00 |    -0.02 |     1.00 |     0.01 |     0.00 |
|   10 |     1.00 |     1.00 |    -0.02 |     1.00 |    -0.00 |     0.00 |
|   11 |     1.00 |     1.00 |    -0.00 |     1.00 |     0.00 |     0.00 |
|   12 |     1.00 |     1.00 |    -0.00 |     1.00 |    -0.00 |     0.00 |
|   13 |     1.00 |     1.00 |    -0.00 |     1.00 |     0.00 |     0.00 |
|   14 |     1.00 |     1.00 |    -0.00 |     1.00 |    -0.00 |     0.00 |
|   15 |     1.00 |     1.00 |    -0.00 |     1.00 |     0.00 |     0.00 |
|   16 |     1.00 |     1.00 |    -0.00 |     1.00 |    -0.00 |     0.00 |
|   17 |     1.00 |     1.00 |    -0.00 |     1.00 |     0.00 |     0.00 |

Error (Bisection) =    5.72204590E-06
```

```
 False Position Method

 The lower bound is greater than the upper bound.  Swapping...

  |       No.  |      xLow  |     xHigh  |xIntercept  |
  |         1  |    0.5000  |    2.0000  |    0.5015  |
  |         2  |    0.5015  |    2.0000  |    0.5029  |
  |         3  |    0.5029  |    2.0000  |    0.5044  |
  |         4  |    0.5044  |    2.0000  |    0.5058  |
  |         5  |    0.5058  |    2.0000  |    0.5073  |
  |         6  |    0.5073  |    2.0000  |    0.5088  |
  |         7  |    0.5088  |    2.0000  |    0.5102  |
  |         8  |    0.5102  |    2.0000  |    0.5117  |
  |         9  |    0.5117  |    2.0000  |    0.5131  |
  |        10  |    0.5131  |    2.0000  |    0.5146  |
  |        11  |    0.5146  |    2.0000  |    0.5160  |
  |        12  |    0.5160  |    2.0000  |    0.5175  |
  |        13  |    0.5175  |    2.0000  |    0.5189  |
  |        14  |    0.5189  |    2.0000  |    0.5204  |
  |        15  |    0.5204  |    2.0000  |    0.5218  |
  |        16  |    0.5218  |    2.0000  |    0.5232  |
  |        17  |    0.5232  |    2.0000  |    0.5247  |
  |        18  |    0.5247  |    2.0000  |    0.5261  |
  |        19  |    0.5261  |    2.0000  |    0.5276  |
  |        20  |    0.5276  |    2.0000  |    0.5290  |
  |        21  |    0.5290  |    2.0000  |    0.5304  |
  |        22  |    0.5304  |    2.0000  |    0.5319  |
  |        23  |    0.5319  |    2.0000  |    0.5333  |
  |        24  |    0.5333  |    2.0000  |    0.5347  |
  |        25  |    0.5347  |    2.0000  |    0.5361  |
  |        26  |    0.5361  |    2.0000  |    0.5376  |
  |        27  |    0.5376  |    2.0000  |    0.5390  |
  |        28  |    0.5390  |    2.0000  |    0.5404  |
  |        29  |    0.5404  |    2.0000  |    0.5418  |
  |        30  |    0.5418  |    2.0000  |    0.5433  |

-~-
-~-
-~-

  |       845  |    0.9989  |    2.0000  |    0.9989  |
  |       846  |    0.9989  |    2.0000  |    0.9989  |
  |       847  |    0.9989  |    2.0000  |    0.9989  |
  |       848  |    0.9989  |    2.0000  |    0.9989  |
  |       849  |    0.9989  |    2.0000  |    0.9989  |
  |       850  |    0.9989  |    2.0000  |    0.9989  |
  |       851  |    0.9989  |    2.0000  |    0.9990  |
  |       852  |    0.9990  |    2.0000  |    0.9990  |
  |       853  |    0.9990  |    2.0000  |    0.9990  |
  |       854  |    0.9990  |    2.0000  |    0.9990  |

 Error (False Position) =    9.89437103E-06
```

```
 Modified False Position Method

 The lower bound is greater than the upper bound.  Swapping...

  |       No.  |      xLow  |     xHigh  |xIntercept  |
  |         1  |    0.5000  |    2.0000  |    0.5015  |
  |         2  |    0.5015  |    2.0000  |    0.5029  |
 The upper bound has not moved in the last two iterations.  Modifying fHigh...

  |         3  |    0.5029  |    2.0000  |    0.5058  |
  |         4  |    0.5058  |    2.0000  |    0.5088  |
 The upper bound has not moved in the last two iterations.  Modifying fHigh...

  |         5  |    0.5088  |    2.0000  |    0.5146  |
  |         6  |    0.5146  |    2.0000  |    0.5203  |
 The upper bound has not moved in the last two iterations.  Modifying fHigh...

  |         7  |    0.5203  |    2.0000  |    0.5318  |
  |         8  |    0.5318  |    2.0000  |    0.5432  |
 The upper bound has not moved in the last two iterations.  Modifying fHigh...

  |         9  |    0.5432  |    2.0000  |    0.5656  |
  |        10  |    0.5656  |    2.0000  |    0.5876  |
 The upper bound has not moved in the last two iterations.  Modifying fHigh...

  |        11  |    0.5876  |    2.0000  |    0.6302  |
  |        12  |    0.6302  |    2.0000  |    0.6714  |
 The upper bound has not moved in the last two iterations.  Modifying fHigh...

  |        13  |    0.6714  |    2.0000  |    0.7482  |
  |        14  |    0.7482  |    2.0000  |    0.8181  |
 The upper bound has not moved in the last two iterations.  Modifying fHigh...

  |        15  |    0.8181  |    2.0000  |    0.9336  |
  |        16  |    0.9336  |    2.0000  |    0.9960  |
 The upper bound has not moved in the last two iterations.  Modifying fHigh...

  |        17  |    0.9960  |    2.0000  |    1.0057  |
  |        18  |    0.9960  |    1.0057  |    0.9999  |
  |        19  |    0.9999  |    1.0057  |    1.0000  |
 The upper bound has not moved in the last two iterations.  Modifying fHigh...

  |        20  |    1.0000  |    1.0057  |    1.0000  |

 Error (Modified False Position) =    5.12599945E-06



             Root with Bisection =    1.0000
         Root with False Position =    0.9990
Root with Modified False Position =    1.0000
```