CH2120

Class 13

# Main Program

**program** mainProgram

**implicit** **none**

**call** mainBisection()

**end** **program** mainProgram

# Bisection Method: Main

**subroutine** mainBisection

**implicit** **none**

**real**, **external** :: bisectionRoot

**real** xLow, xHigh, root

**real** tolerance

**write**(\*,\*)

**write**(\*,\*) "------------------------------------------------"

**write**(\*,\*) "Bisection Method for finding a root of f(x) = 0"

**write**(\*,\*) "------------------------------------------------"

**write**(\*,\*)

**write**(\*,\*) "Enter the lower limit of the initial bracket:"

**read**(\*,\*) xLow

**write**(\*,\*) "Enter the upper limit of the initial bracket:"

**read**(\*,\*) xHigh

**write**(\*,\*) "Enter the tolerance:"

**read**(\*,\*) tolerance

root = bisectionRoot(xLow, xHigh, tolerance)

**write**(\*,\*) "Root = ", root

**end** **subroutine** mainBisection

# bisectionRoot

**real** **function** bisectionRoot(xLowInput, xHighInput, tolerance)

**implicit** **none**

**real**, **external** :: rootFindingFunction

**logical**, **external** :: haveOppositeSigns

**real**, **intent**(in) :: xLowInput, xHighInput

**real**, **intent**(in) :: tolerance

**real** error

**integer** iterations

**real** xLow, xMid, xHigh

**real** fLow, fMid, fHigh

xLow = xLowInput

xHigh = xHighInput

xMid = (xLow + xHigh) / 2

fLow = rootFindingFunction(xLow)

fHigh = rootFindingFunction(xHigh)

**if**(fLow == 0) **then**

bisectionRoot = xLow

**return**

**end** **if**

**if**(fHigh == 0) **then**

bisectionRoot = xHigh

**return**

**end** **if**

**if**(haveOppositeSigns(fLow, fHigh) .**eqv**. .**false**.) **then**

**stop** "Bisection Method Failed: Bounds don't bracket the root."

**end** **if**

error = *abs*(xHigh - xLow) / 2

iterations = 0

**write**(\*,10) " | ", "No.", " | ", "xLow", " | ", "xHigh", " | ", "xMid", " | ", "fLow", " | ", "fMid", " | ", &

"fHigh", " | ", "Error", " | "

10 **format**(a3, a4, a3, a7, a3, a7, a3, a7, a3, a7, a3, a7, a3, a7, a3, a7, a3)

**do** **while**(error > tolerance)

iterations = iterations + 1

xMid = (xLow + xHigh) / 2

fMid = rootFindingFunction(xMid)

error = *abs*(xHigh - xLow) / 2

**write**(\*,20) " | ", iterations, " | ", xLow, " | ", xHigh, " | ", xMid, " | ", fLow, " | ", fMid, " | ", &

fHigh, " | ", error, " | "

20 **format**(a3, i4, a3, f7.2, a3, f7.2, a3, f7.2, a3, f7.2, a3, f7.2, a3, f7.2, a3, f7.4, a3)

**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**

**end** **do**

**write**(\*,\*) "Final error = ", error

**write**(\*,\*) "Iterations = ", iterations

bisectionRoot = xMid

**end** **function** bisectionRoot

# rootFindingFunction

**real** **function** rootFindingFunction(x)

**implicit** **none**

**real**, **intent**(in) :: x

rootFindingFunction = (3 \* x) + *sin*(x) - *exp*(x)

**end** **function** rootFindingFunction

# haveOppositeSigns

**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

# Swap Numbers

*Optional*

**subroutine** swapNumbers(number1, number2)

**implicit** **none**

**real**, **intent**(inout) :: number1, number2

**real** swapper

swapper = number1

number1 = number2

number2 = swapper

**end** **subroutine** swapNumbers

# Output

## Output1

------------------------------------------------

Bisection Method for finding a root of f(x) = 0

------------------------------------------------

Enter the lower limit of the initial bracket:

0

Enter the upper limit of the initial bracket:

1

Enter the tolerance:

1e-6

| No. | xLow | xHigh | xMid | fLow | fMid | fHigh | Error |

| 1 | 0.00 | 1.00 | 0.50 | -1.00 | 0.33 | 1.12 | 0.5000 |

| 2 | 0.00 | 0.50 | 0.25 | -1.00 | -0.29 | 0.33 | 0.2500 |

| 3 | 0.25 | 0.50 | 0.38 | -0.29 | 0.04 | 0.33 | 0.1250 |

| 4 | 0.25 | 0.38 | 0.31 | -0.29 | -0.12 | 0.04 | 0.0625 |

| 5 | 0.31 | 0.38 | 0.34 | -0.12 | -0.04 | 0.04 | 0.0312 |

| 6 | 0.34 | 0.38 | 0.36 | -0.04 | -0.00 | 0.04 | 0.0156 |

| 7 | 0.36 | 0.38 | 0.37 | -0.00 | 0.02 | 0.04 | 0.0078 |

| 8 | 0.36 | 0.37 | 0.36 | -0.00 | 0.01 | 0.02 | 0.0039 |

| 9 | 0.36 | 0.36 | 0.36 | -0.00 | 0.00 | 0.01 | 0.0020 |

| 10 | 0.36 | 0.36 | 0.36 | -0.00 | -0.00 | 0.00 | 0.0010 |

| 11 | 0.36 | 0.36 | 0.36 | -0.00 | 0.00 | 0.00 | 0.0005 |

| 12 | 0.36 | 0.36 | 0.36 | -0.00 | 0.00 | 0.00 | 0.0002 |

| 13 | 0.36 | 0.36 | 0.36 | -0.00 | 0.00 | 0.00 | 0.0001 |

| 14 | 0.36 | 0.36 | 0.36 | -0.00 | -0.00 | 0.00 | 0.0001 |

| 15 | 0.36 | 0.36 | 0.36 | -0.00 | 0.00 | 0.00 | 0.0000 |

| 16 | 0.36 | 0.36 | 0.36 | -0.00 | 0.00 | 0.00 | 0.0000 |

| 17 | 0.36 | 0.36 | 0.36 | -0.00 | -0.00 | 0.00 | 0.0000 |

| 18 | 0.36 | 0.36 | 0.36 | -0.00 | 0.00 | 0.00 | 0.0000 |

| 19 | 0.36 | 0.36 | 0.36 | -0.00 | 0.00 | 0.00 | 0.0000 |

| 20 | 0.36 | 0.36 | 0.36 | -0.00 | -0.00 | 0.00 | 0.0000 |

Final error = 9.53674316E-07

Iterations = 20

Root = 0.360421181

## Output 2

------------------------------------------------

Bisection Method for finding a root of f(x) = 0

------------------------------------------------

Enter the lower limit of the initial bracket:

1

Enter the upper limit of the initial bracket:

2

Enter the tolerance:

1e-6

| No. | xLow | xHigh | xMid | fLow | fMid | fHigh | Error |

| 1 | 1.00 | 2.00 | 1.50 | 1.12 | 1.02 | -0.48 | 0.5000 |

| 2 | 1.50 | 2.00 | 1.75 | 1.02 | 0.48 | -0.48 | 0.2500 |

| 3 | 1.75 | 2.00 | 1.88 | 0.48 | 0.06 | -0.48 | 0.1250 |

| 4 | 1.88 | 2.00 | 1.94 | 0.06 | -0.20 | -0.48 | 0.0625 |

| 5 | 1.88 | 1.94 | 1.91 | 0.06 | -0.06 | -0.20 | 0.0312 |

| 6 | 1.88 | 1.91 | 1.89 | 0.06 | -0.00 | -0.06 | 0.0156 |

| 7 | 1.88 | 1.89 | 1.88 | 0.06 | 0.03 | -0.00 | 0.0078 |

| 8 | 1.88 | 1.89 | 1.89 | 0.03 | 0.01 | -0.00 | 0.0039 |

| 9 | 1.89 | 1.89 | 1.89 | 0.01 | 0.01 | -0.00 | 0.0020 |

| 10 | 1.89 | 1.89 | 1.89 | 0.01 | 0.00 | -0.00 | 0.0010 |

| 11 | 1.89 | 1.89 | 1.89 | 0.00 | -0.00 | -0.00 | 0.0005 |

| 12 | 1.89 | 1.89 | 1.89 | 0.00 | 0.00 | -0.00 | 0.0002 |

| 13 | 1.89 | 1.89 | 1.89 | 0.00 | 0.00 | -0.00 | 0.0001 |

| 14 | 1.89 | 1.89 | 1.89 | 0.00 | -0.00 | -0.00 | 0.0001 |

| 15 | 1.89 | 1.89 | 1.89 | 0.00 | -0.00 | -0.00 | 0.0000 |

| 16 | 1.89 | 1.89 | 1.89 | 0.00 | -0.00 | -0.00 | 0.0000 |

| 17 | 1.89 | 1.89 | 1.89 | 0.00 | 0.00 | -0.00 | 0.0000 |

| 18 | 1.89 | 1.89 | 1.89 | 0.00 | 0.00 | -0.00 | 0.0000 |

| 19 | 1.89 | 1.89 | 1.89 | 0.00 | 0.00 | -0.00 | 0.0000 |

| 20 | 1.89 | 1.89 | 1.89 | 0.00 | 0.00 | -0.00 | 0.0000 |

Final error = 9.53674316E-07

Iterations = 20

Root = 1.89002895

## Output 3

------------------------------------------------

Bisection Method for finding a root of f(x) = 0

------------------------------------------------

Enter the lower limit of the initial bracket:

0

Enter the upper limit of the initial bracket:

2

Enter the tolerance:

1e-6

STOP Bisection Method Failed: Bounds don't bracket the root.