# Numerical Methods I

## Roots of Equations: Open Methods

**program** mainOpenMethods

**implicit** **none**

**real**, **external** :: fixedPointIteration

**real**, **external** :: newtonRaphson

**real**, **external** :: secantMethod

**real** :: xGuess

**real** :: xGuess1, xGuess2

**real**, **parameter** :: tolerance = 1.0e-5

**real** :: rootFP, rootNR, rootSM

**write**(\*,\*) "Root-finding (Open Methods)"

xGuess = 0.0

rootFP = fixedPointIteration(xGuess, tolerance)

xGuess = 0.0

rootNR = newtonRaphson(xGuess, tolerance)

xGuess1 = 0.0

xGuess2 = 1.0

rootSM = secantMethod(xGuess1, xGuess2, tolerance)

**write**(\*,10) "Root by fixed-point iteration method = ", rootFP

**write**(\*,10) "Root by Newton Raphson method = ", rootNR

**write**(\*,10) "Root by Secant method = ", rootSM

10 **format**(a39, f9.4)

**end** **program** mainOpenMethods

### The Fixed-Point Iteration Method

**real** **function** fixedPointIteration(xGuess, tolerance)

**implicit** **none**

**real**, **external** :: getFunctionForFixedPoint

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

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

**real** :: x, xPrevious

**real** :: error

**integer** :: iteration = 0

**write**(\*,\*) "The Fixed-Point Iteration Method"

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

**if**(getFunctionForFixedPoint(xGuess) == xGuess) **then**

fixedPointIteration = xGuess

**return**

**end** **if**

x = xGuess

**write**(\*,10) "|", "Iteration", "|", "Root", "|", "error", "|"

**do** **while**((error > tolerance) .**or**. (iteration <= 2))

iteration = iteration + 1

**write**(\*,20) "|", iteration, "|", x, "|", error, "|"

x = getFunctionForFixedPoint(x)

error = *abs*(x - xPrevious)

xPrevious = x

**end** **do**

fixedPointIteration = x

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

10 **format**(a3, a9, a3, a9, a3, a9, a3)

20 **format**(a3, i9, a3, f9.4, a3, f9.4, a3)

**end** **function** fixedPointIteration

### The Newton-Raphson Method

**real** **function** newtonRaphson(xGuess, tolerance)

**implicit** **none**

**real**, **external** :: getFunction, getDerivative

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

**real** :: x, xPrevious

**real** :: error

**integer** :: iteration = 0

**write**(\*,\*) "The Newton-Raphson Method"

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

**if**(getFunction(xGuess) == 0) **then**

newtonRaphson = xGuess

**return**

**end** **if**

x = xGuess

**write**(\*,10) "|", "Iteration", "|", "Root", "|", "Error", "|"

**do** **while**((error > tolerance) .**or**. (iteration <= 2))

iteration = iteration + 1

**if**(getDerivative(x) == 0) **then**

stop "-- Error -- (Newton-Raphson) f'(x) = 0"

**end** **if**

x = x - getFunction(x) / getDerivative(x)

error = *abs*(x - xPrevious)

xPrevious = x

**write**(\*,20) "|", iteration, "|", x, "|", error, "|"

**end** **do**

newtonRaphson = x

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

10 **format**(a3, a10, a3, a7, a3, a7, a3)

20 **format**(a3, i10, a3, f7.2, a3, f7.2, a3)

**end** **function** newtonRaphson

### The Secant Method

**real** **function** secantMethod(xGuessOne, xGuessTwo, tolerance)

**implicit** **none**

**real**, **external** :: getFunction

**real**, **intent**(in) :: xGuessOne, xGuessTwo

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

**real** :: xOne, xTwo, xThree, xThreePrevious

**real** :: fOne, fTwo, fThree

**real** :: slope

**real** :: error

**integer** :: iteration = 0

**write**(\*,\*) "The Secant Method"

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

**if**(getFunction(xGuessOne) == 0) **then**

secantMethod = xGuessOne

**return**

**end** **if**

**if**(getFunction(xGuessTwo) == 0) **then**

secantMethod = xGuessTwo

**return**

**end** **if**

**if**(xOne == xTwo) **then**

**stop** "-- Error -- (Secant Method) The two guesses for x should not be the same."

**end** **if**

xOne = xGuessOne

xTwo = xGuessTwo

fOne = getFunction(xOne)

fTwo = getFunction(xTwo)

**write**(\*,10) "|", "Iteration", "|", "xOne", "|", "xTwo", "|", "Root", "|", "Error", "|"

**do** **while**((error > tolerance) .**or**. (iteration <= 2))

iteration = iteration + 1

slope = (fOne - fTwo) / (xOne - xTwo)

xThree = xOne - ((fOne) / (slope))

! Alternative expressions for xThree

! xThree = xTwo - ((fTwo) / (slope))

! xThree = ((xTwo \* fOne) - (xOne \* fTwo)) / (fOne - fTwo)

fThree = getFunction(xThree)

xOne = xTwo

fOne = fTwo

xTwo = xThree

fTwo = fThree

error = *abs*(xThree - xThreePrevious)

xThreePrevious = xThree

**write**(\*,20) "|", iteration, "|", xOne, "|", xTwo, "|", xThree, "|", error, "|"

**end** **do**

secantMethod = xThree

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

10 **format**(a3, a10, a3, a7, a3, a7, a3, a7, a3, a7, a3)

20 **format**(a3, i10, a3, f7.2, a3, f7.2, a3, f7.2, a3, f7.2, a3)

**end** **function** secantMethod

### Common Procedures

**real** **function** getFunction(x)

**implicit** **none**

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

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

**end** **function** getFunction

**real** **function** getFunctionForFixedPoint(x)

**implicit** **none**

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

getFunctionForFixedPoint = (*exp*(x) - *sin*(x)) / 3

**end** **function** getFunctionForFixedPoint

**real** **function** getDerivative(x)

**implicit** **none**

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

getDerivative = (3) + *cos*(x) - *exp*(x)

**end** **function** getDerivative

### Output

Root-finding (Open Methods)

The Fixed-Point Iteration Method

|Iteration | Root | error |

| 1 | 0.0000 | 0.0000 |

| 2 | 0.3333 | 0.3333 |

| 3 | 0.3561 | 0.0228 |

| 4 | 0.3597 | 0.0036 |

| 5 | 0.3603 | 0.0006 |

| 6 | 0.3604 | 0.0001 |

| 7 | 0.3604 | 0.0000 |

The Newton-Raphson Method

| Iteration | Root | Error |

| 1 | 0.33 | 0.33 |

| 2 | 0.36 | 0.03 |

| 3 | 0.36 | 0.00 |

| 4 | 0.36 | 0.00 |

The Secant Method

| Iteration | xOne | xTwo | Root | Error |

| 1 | 1.00 | 0.47 | 0.47 | 0.47 |

| 2 | 0.47 | 0.31 | 0.31 | 0.16 |

| 3 | 0.31 | 0.36 | 0.36 | 0.06 |

| 4 | 0.36 | 0.36 | 0.36 | 0.00 |

| 5 | 0.36 | 0.36 | 0.36 | 0.00 |

| 6 | 0.36 | 0.36 | 0.36 | 0.00 |

Root by fixed-point iteration method = 0.3604

Root by Newton Raphson method = 0.3604

Root by Secant method = 0.3604