Math 4670/5670 Homework Two Revised

September 25, 2017

1. Write a Fortran program to solve x3-30=0 using the method of bisection.
2. Solve problem 17 of section 2.3 in the text book. Use a program you create, and use either bisection, the secant method or Newton’s method.
3. Write a program to solve problem 15 of section 2.4.
4. Write a Fortran program to solve x3-30=0 using the method of bisection.

f(0) = -30 f(1) = -29 f(2) = -22

f(3) = -3 f(4) = 36

! Mikayla Webber

! 4670 Numerical Analysis

! Homework Two Redo

! From Notes on Bisection Method:

! f needs to be continuous

! 1. Start with a, b so that f(a), f(b) have opposite signs

! 2. Find the middle M = (a+b)/2 and calculate f(M)

! 3. If f(a) and f(M) have opposite signs

! set b = M

! else set a = M

! end if

! Loop back to #2

module secret

end module secret

program WebberHomework2Question1

use secret ! uses secret module

implicit none

real :: a

real :: b

real :: f

real :: m

real :: mid

real :: tolerance

integer :: i

integer :: findSign ! I didn't realize "sign" was

! a reserved word

a = -10.0d0

b = 10.0d0

tolerance = 1.0d0-10

do i = 1, 10

m = mid(a, b) ! find the middle and calculate f(M)

if (m < tolerance) then ! if m is less than tolerance then

print\*, "Middle = ", m ! prints the middle

stop

else if (f(m) == 0) then ! else if f(m) equals zero then

print\*, "Middle = ", m ! prints the middle

stop

else ! if neither case is true

print\*, "f(a, b) = (", a, ",", b, ") and m = ", m

end if

if ((findSign(f(a)) \* findSign(f(m))) < 0) then ! if f(a) && f(M) have opposite signs

b = m ! it sets b to M

else

a = b ! else it sets a to b

end if

end do

stop

end program WebberHomework2Question1

real function f(x) ! main function for the problem

implicit none

real :: x

f = ((x\*\*3)-30) ! equation for f = x^3 - 30

return

end

real function mid(a,b) ! function to find midpoint

implicit none

real :: a

real :: b

mid = ((a + (b - a)) / 2) ! midpoint formula

return

end

integer function findSign(x) ! function to determine if negative

implicit none ! or positive

real :: x

if (x < 0) then ! if less than zero sign is -1

findSign = -1

else if (x == 0) then ! else if x is zero sign is 0

findSign = 0

else ! if x is greater than zero sign is 1

findSign = 1

end if

return

end

1. Solve problem 17 of section 2.3 in the text book. Use a program you create, and use either bisection, the secant method or Newton’s method.

! Mikayla Webber

! 4670 Numerical Analysis

! Homework Two Redo

! From Notes on Bisection Method:

! f needs to be continuous

! 1. Start with a, b so that f(a), f(b) have opposite signs

! 2. Find the middle M = (a+b)/2 and calculate f(M)

! 3. If f(a) and f(M) have opposite signs

! set b = M

! else set a = M

! end if

! Loop back to #2

module secret

end module secret

program WebberHomework2Question2

use secret ! uses secret module

implicit none

double precision :: a

double precision :: b

double precision :: f

double precision :: m

double precision :: mid

double precision :: tolerance

integer :: i

integer :: findSign ! I didn't realize "sign" was a

! reserved word

a = -5.0d0

b = 5.0d0

tolerance = 1.0d0-10

do i = 1, 100

m = mid(a, b) ! find the middle and calculate f(M)

if (((b - a) / 2) < tolerance) then ! if m is less than tolerance then

print\*, "Middle = ", m ! prints the middle

stop

else if (f(m) == 0) then ! else if f(m) equals zero then

print\*, "Middle = ", m ! prints the middle

stop

else ! if neither case is true

print\*, "f(a, b) = (", a, ",", b, ") and m = ", m

end if

if ((findSign(f(a)) \* findSign(f(m))) < 0) then ! if f(a) && f(M) have opposite signs

b = m ! it sets b to M

else

a = m ! else it sets a to b

end if

end do

stop

end program WebberHomework2Question2

double precision function f(x) ! main function for the problem

implicit none

double precision :: x

f = ((-32.17d0 / (2.0d0 \* x \*\* 2.0d0)) \* ((dsinh(x) - dsin(x)) - 1.7d0))

return

end

double precision function mid(a,b) ! function to find midpoint

implicit none

double precision :: a

double precision :: b

mid = ((a + (b - a)) / 2.0d0) ! midpoint formula

return

end

double precision function findSign(x) ! function to determine if negative

implicit none ! or positive

double precision :: x

if (x < 0.0d0) then ! if less than zero sign is -1

findSign = -1.0d0

else if (x == 0.0d0) then ! else if x is zero sign is 0

findSign = 0.0d0

else ! if x is greater than zero sign is 1

findSign = 1.0d0

end if

return

end

1. Write a program to solve problem 15 of section 2.4.

! Mikayla Webber

! 4670 Numerical Analysis

! Homework Two Redo

module secret

end module secret

program WebberHomework2Question3

use secret ! uses secret module

implicit none

double complex :: f

double complex :: p

double complex :: old

double complex :: prime

double complex :: tolerance

p = (0, 1)

old = (p\*2.0d0)

tolerance = 1.0d0-10

do while ((abs(p - old) /= tolerance))

old = p

p = (p + 1)

p = (f(old) - (old - f(old)) / prime(old))

print\*, "probability = ", p

end do

stop

end program WebberHomework2Question3

double complex function f(p) ! function for probability

implicit none

double complex :: p

f = ((p + 1) / 2)\* & (p / (1 - p + p\*\*2)\*\*21)

return

end

double complex function firstP(p)

implicit none

double complex :: p

firstP = 1 - 1/2 p / & 1 - p + p\*\*21

return

end

double complex function secondP(p)

implicit none

double complex :: p

secondP = 21(1 + p) / 2\* & (p / (1 - p + p\*\*2))\*\*20

return

end

double complex function thirdP(p)

implicit none

double complex :: p

thirdP = (1 - p\*\*2) / & (1 - p + p\*\*2)\*\*2

return

end

double complex function prime(p)

implicit none

double complex :: p

double complex :: firstP

double complex :: secondP

double complex :: thirdP

prime = (firstP(p) - secondP(p) \* thirdP(p))

return

end