



```

WRITE (UNIT=*, FMT='(A)', ADVANCE='NO') ' Enter target X coordinate, XT (m): '
READ (UNIT=*, FMT=*) XT

WRITE (UNIT=*, FMT='(A)', ADVANCE='NO') ' Enter target Y coordinate, YT (m): '
READ (UNIT=*, FMT=*) YT

WRITE (UNIT=*, FMT='(A)', ADVANCE='NO') ' Enter estimate of launch angle, THETA (deg): '
READ (UNIT=*, FMT=*) THETA_INIT

WRITE (UNIT=*, FMT='(//)')

THETA_INIT = THETA_INIT * PI/180.0D0                                ! convert THETA_INIT to radians

!
! Use Newton's method to solve for angle THETA.
!

I = 1                                                                ! initialize iteration counter
THETA = THETA_INIT                                                  ! initialize THETA to given estimate

DO                                                                    ! start of iteration loop

    THETA_NEXT = THETA - F(THETA,XT,YT,V0) / FP(THETA,XT,YT)        ! find next estimate of THETA (rad)

    WRITE (UNIT=*, FMT='(A,I5,A,F10.5,A)') ' Iteration ', I, ': THETA = ', & ! print results of each iteration
        THETA_NEXT*180.0D0/PI, ' deg'

    IF (ABS(THETA_NEXT-THETA) .LE. TOL) EXIT                          ! if converged, then exit loop

    I = I + 1                                                         ! increment iteration counter

    IF (I .GT. MAX_ITER) THEN                                         ! if max iterations exceeded, then stop
        WRITE (UNIT=*, FMT='(A)') ' Error: maximum iterations exceeded.'
        STOP
    END IF

    THETA = THETA_NEXT                                                ! save THETA for next iteration

END DO                                                                ! end of iteration loop

THETA = THETA_NEXT                                                  ! save final value of THETA (rad)

WRITE (UNIT=*, FMT='(/A,F10.5,A)') ' Converged. THETA = ', & ! print final value of THETA
    THETA*180.0D0/PI, ' deg'

```

```

!
! Check the answer. We check the answer by substituting the value of THETA
! just found into the formula for y(x). We then compare the result with the
! value of y for the target (YT).
!
ERROR = ABS(YT-YX(XT,THETA,V0)) ! error in y (meters)
WRITE (UNIT=*, FMT='(/A,F10.3,A)') ' Error = ', ERROR, ' meters'

STOP

END PROGRAM PROJECTL

!*****
! F
!
! This function computes the value f needed by Newton's method.
!*****

FUNCTION F (THETA, XT, YT, V0) RESULT (Y)

IMPLICIT NONE

DOUBLE PRECISION, INTENT(IN) :: THETA, & ! launch angle (rad)
                                XT,    & ! x coordinate of target (m)
                                YT,    & ! y coordinate of target (m)
                                V0      ! launch speed (m/s)

DOUBLE PRECISION :: Y ! result f (m)

DOUBLE PRECISION, PARAMETER :: G = 9.80 ! acceleration due to gravity (m/s**2)

Y = XT*SIN(2.0D0*THETA) - 2.0D0*YT*COS(THETA)**2 - G*XT**2/V0**2 ! compute f

RETURN

END FUNCTION F

```

```

!*****
! FP
!
! This function computes the value of f' needed by Newton's method.
!*****

```

```

FUNCTION FP (THETA, XT, YT) RESULT (YP)

IMPLICIT NONE

DOUBLE PRECISION, INTENT(IN) :: THETA, &           ! launch angle (rad)
                                XT,      &           ! x coordinate of target (m)
                                YT           ! y coordinate of target (m)
DOUBLE PRECISION :: YP                               ! result f' (m/rad)

YP = 2.0D0*XT*COS(2.0D0*THETA) + 2.0D0*YT*SIN(2.0*THETA)      ! compute f'

RETURN

END FUNCTION FP

```

```

!*****
! YX
!
! This function computes y(x); that is, it returns the value of y for a given value of the coordinate x.
!*****

```

```

FUNCTION YX (X, THETA, V0) RESULT (Y)

IMPLICIT NONE

DOUBLE PRECISION, INTENT(IN) :: X,      &           ! x coordinate (m)
                                THETA, &           ! launch angle (rad)
                                V0       ! launch speed (m/s)
DOUBLE PRECISION :: Y                               ! y coordinate (m)

DOUBLE PRECISION, PARAMETER :: G = 9.80           ! acceleration due to gravity (m/s**2)

Y = (-G/(2.0D0*V0**2*COS(THETA)**2))*X**2 + TAN(THETA)*X      ! compute y coordinate (m)

RETURN

END FUNCTION YX

```