```
c
c     Shooting Method
      program shoot_1eq
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (N=2)
      double precision Y(N),YOLD(N)
      COMMON h,x1,x2,tol,itmax
      OPEN(11,FILE='shoot_1eq.txt')
c
      h     = 1.d-1
      x1    = 0
      x2    = 1.d0
      tol   = 1.d-12
      epsl  = 1.d-5
      itmax = 10
c
c     provide initial condition for YOLD(N)
      Beta    = 2.d0
      znm1    = 0.1d0
      zn      = znm1+epsl
c
c     Integrate IVP's for the first guess of IC
      YOLD(1) = 1.d0
      YOLD(2) = znm1
      call Adams_Bashford(N,YOLD,Y)
      phinm1  = Y(1)
c
c     Integrate IVP's for the second guess of IC
      YOLD(1) = 1.d0
      YOLD(2) = zn
      call Adams_Bashford(N,YOLD,Y)
      phin    = Y(1)
c
c     Shooting Method Loop
      ishoot = 0
  10  ishoot = ishoot + 1
      if (ishoot .gt. itmax) stop
      write(*,'(5x,a,i3,5x,a,f20.12)')
     !      'iteration =',ishoot,'error =',phin-beta
c
c     Set Termination Criteria for the Shooting Method Loop
      if (dabs(phin-beta) .lt. tol) stop
      znp1 = zn + (beta-phin)*(zn-znm1)/(phin-phinm1)
c
c     Integrate IVP's for the current guess of IC
      YOLD(1) = 1.d0
      YOLD(2) = znp1
      call Adams_Bashford(N,YOLD,Y)
      znm1   = zn
      zn     = znp1
      phinm1 = phin
      phin   = Y(1)
c
      go to 10
1000  write(11,*) 'No Convergence !'
      write(*,*)  'No Convergence !'
c
      stop
      end
c
c     END of main Program
c
c
c     Subroutine Func to Input RHS functions of IVP
      subroutine Func(N,Y,FUN)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      double precision Y(N),FUN(N)
      COMMON h,x1,x2,tol,itmax
c
      do i = 1, N
      FUN(i) = 0.d0
      enddo
```

```
c
      FUN(1) = Y(2)
      FUN(2) = Y(1)
c
      return
      end
c
c     Subroutine Adams_Bashford to integrate IVP's
      subroutine Adams_Bashford(N,YOLD,Y)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      double precision Y(N),YOLD(N),FUNOLD1(N),FUNOLD2(N)
      COMMON h,x1,x2,tol,itmax
c
      istep = 0
      xval  = x1
c
c     Print the Initial solution to output.txt
      write(11,*)
      write(11,'(12x,a,18x,a,17x,a)')  'X','Y(1)','Y(2)'
      write(11,'(3f20.12)') xval,YOLD(1),YOLD(2)
c
c     Evaluate the RHS Function for the Initial Solution
      call Func(N,YOLD,FUNOLD1)
c
c     Start Integration Loop
  10  xval  = xval + h
      istep = istep + 1
c
c     Set Termination Criteria for the Integration Loop
      if((xval-x2) .gt. 1.d-14) return
c
c     For the first time step apply 1st-order explicit Euler
      if (istep .eq. 1) then
        do i = 1, N
          Y(i) = YOLD(i)+h*FUNOLD1(i)
        enddo
      else
c
c     Apply 2nd-order AB Formula for Remaining Time Steps
        do i = 1, N
          Y(i) = YOLD(i)+h/2.d0*(3.d0*FUNOLD1(i)-FUNOLD2(i))
        enddo
      endif
c
c     Print the solution for the current time step to shooting.txt
      write(11,'(3f20.12)') xval,Y(1),Y(2)
c
c     Store Previous Time Values
      do i = 1, N
        YOLD(i)    = Y(i)
        FUNOLD2(i) = FUNOLD1(i)
      enddo
c
c     Evaluate the RHS Function at Current Time
      call Func(N,YOLD,FUNOLD1)
      go to 10
c
      end
```