Fortran 90 source code

```
      PROGRAM RAERD_MAIN
      IMPLICIT REAL(I-N)
      INTEGER MX1,MX2
      PARAMETER (MX1=1000,MX2=500)
      INTEGER K,I,J,IQ,N1,N2,DT,T0,T(MX1)
      INTEGER SA_SUM1,SA_SUM2,SA_SUM3,OC_SUM,SA_EPS,QB_out

      COMMON /FRC/Q_in(MX1),V(MX1),Z(MX1),Q_out(MX1)
      COMMON /RC/Q_d(MX1)
      COMMON /CC/QQ(MX2),VV(MX2),ZZQ(MX2),ZZV(MX2)

      OPEN(1,FILE='INPUT1.TXT')
      OPEN(2,FILE='INPUT2.TXT')
      OPEN(3,FILE='OUTPUT.TXT')

      READ(1,*)DT,IQ,N1,N2

READ(1,*)SA_EPS,LAMD_CGM,LAMD_MIU,LAMD_MAX,LAMD_MIN,Qm_MAX,Qm_MI
N,LAMD_CGM1,LAMD_MAX1,LAMD_MIN1
      READ(1,*)Z_fpwl,Z_check
      READ(1,*)(Q_d(I),I=1,IQ)
      READ(2,*)(QQ(I),I=1,N1),(ZZQ(I),I=1,N1),&
               (VV(I),I=1,N2),(ZZV(I),I=1,N2)
       OC_SUM=0
       SA_SUM3=0
10     SA_SUM1=0
       LAMD1=LAMD_CGM1*NF(IDUM)+LAMD_MIU
       IF (LAMD1.LE.LAMD_MAX1.AND.LAMD1.GE.LAMD_MIN1) THEN
      Zzero=LAMD1*Z_fpwl
      WRITE(*,*)Zzero
      SA_SUM3=SA_SUM3+1
      CALL IP(ZZQ,QQ,Zzero,QB_out,N1)
100   SA_SUM2=0
      LAMD=LAMD_CGM*NF(IDUM)+LAMD_MIU
      IF (LAMD.LE.LAMD_MAX.AND.LAMD.GE.LAMD_MIN) THEN
          SA_SUM1=SA_SUM1+1
          Q_out(1)=QB_out*LAMD
200   Qm=Qm_MIN+NF(IDUM)*(Qm_MAX-Qm_MIN)
          IF(Qm.LE.Qm_MAX.AND.Qm.GE.Qm_MIN) then
                SA_SUM2=SA_SUM2+1
                do I=1,IQ
                    Q_in(I)=Q_d(I)*Qm/Qd
```

```
                    end do
                        CALL FR(IQ,DT,N1,N2,Z_MAX,Zzero)
                       IF(Z_MAX.GE.MAX) THEN
                        MAX=Z_MAX
                        ENDIF
                        IF (Z_MAX.GE.Z_check) THEN
                            OC_SUM=OC_SUM+1
                         eND IF

                       IF (SA_SUM2<SA_EPS) THEN

                          GOTO 200
                       ELSE
                         IF (SA_SUM1<SA_EPS) THEN
                             GOTO 100
                              ELSE
                         IF (SA_SUM3<SA_EPS) THEN
                             GOTO 10
                          else
                            goto 1000
                          end if
                       end if
                  goto 100
                END IF
             ELSE
               GOTO 200
             END IF
           ELSE
               GOTO 100
       END IF
     ELSE
               GOTO 10
       END IF


1000    RR_FC=OC_SUM*1.0/SA_EPS**3


!        **********************************************************************
!        ----------          Computed Result          ----------

         WRITE(3,*)"
         WRITE(3,*)'                                                  Comprehensive risk rate '
         WRITE(3,2000)
```

```
      WRITE(3,*)'     Total number of samples：                                    ',SA_EPS**3
      WRITE(3,*)
      WRITE(3,*)'      Times of the water level exceeds the dam crest elevation：    ',OC_SUM
      WRITE(3,*)
      WRITE(3,*)'     Maximum water level                                          ',MAX
      WRITE(3,*)
      WRITE(3,2200)RR_FC*100
2000  FORMAT(66('-'))
2200  FORMAT('                    Flood risk：                              ',F8.2,'%')
      WRITE(3,2000)
      END
!     ----------------------------------------------------------------
```

```
!
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++
!                                  N(0,1) Distributed Random Number Subprogram
!
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++
      function NF(idum)
      implicit real(i-n)
      integer idum
      real ran2,nf,v1,v2,fac

100   v1=ran2(idum)
      v2=ran2(idum)
      if (v1.gt.1.0) goto 100
      if (v1.eq.0.0) goto 100
      if (v2.gt.1.0) goto 100
      if (v2.eq.0.0) goto 100
      NF=sqrt(-2*log(v1))* cos(2*3.1415926*v2)
      end
!     ----------------------------------------------------------------
```

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+
!                          N(0,1) Distributed Random Number Subprogram
!
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+
FUNCTION RAN2(idum)
INTEGER idum,IM1,IM2,IMM1,IA1,IA2,IQ1,IQ2,&
            IR1,IR2,NTAB,NDIV
REAL RAN2,AM,EPS,RNMX
PARAMETER (IM1=2147483563,IM2=2147483399,AM=1./IM1,&
                  IMM1=IM1-1,IA1=40014,IA2=40692,IQ1=53668,&
                  IQ2=52774,IR1=12211,IR2=3791,NTAB=32,&
                  NDIV=1+IMM1/NTAB,EPS=1.2e-7,RNMX=1.-EPS)
INTEGER idum2,j,k,iv(NTAB),iy
SAVE iv,iy,idum2
DATA idum2/123456789/, iv/NTAB*0/, iy/0/
if (idum<=0) then
   idum=max(-idum,1)
   idum2=idum
   do j=NTAB+8,1,-1
      k=idum/IQ1
      idum=IA1*(idum-k*IQ1)-k*IR1
      if (idum<0) idum=idum+IM1
      if (j<=NTAB) iv(j)=idum
   end do
   iy=iv(1)
endif
k=idum/IQ1
idum=IA1*(idum-k*IQ1)-k*IR1
if (idum<0) idum=idum+IM1
k=idum2/IQ2
idum2=IA2*(idum2-k*IQ2)-k*IR2
if (idum2<0) idum2=idum2+IM2
j=1+iy/NDIV
iy=iv(j)-idum2
iv(j)=idum
if(iy<1) iy=iy+IMM1
RAN2=min(AM*iy,RNMX)
END

!        ----------------------------------------------------------------
```

```
!
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                        !                  Linear interpolation Subprogram
   ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
        subroutine IP(xx,yy,x,y,n)
        implicit real(i-n)
        integer k,n
        real xx(n),yy(n)

        if (x.ge.xx(n).or.x.le.xx(1)) then
            if (x.ge.xx(n)) then
                y=yy(n)
            else
                y=yy(1)
            end if
        end if
        do k=1,n-1
           if (x.ge.xx(k).and.x.le.xx(k+1)) then
               if(xx(k).eq.xx(k+1))then
                  y=(yy(k)+yy(k+1))*0.5
               else
                   l1=(x-xx(k+1))/(xx(k)-xx(k+1))
                   l2=(x-xx(k))/(xx(k+1)-xx(k))
                   y=yy(k)*l1+yy(k+1)*l2
               end if
           end if
        end do
        end
!       ----------------------------------------------------------------




!
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
!
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
!                              Reservoir flood routing Subprogram
!       ----------------------------------------------------------------
        SUBROUTINE FR(IQ,DT,N1,N2,Z_MAX,Zzero)
        IMPLICIT REAL(I-N)
        PARAMETER(MX1=1000,MX2=500)                !,MX3=5
        INTEGER K,I,J,IQ,N1,N2,DT,T0,T(MX1)
```

```
      COMMON /FRC/Q_in(MX1),Q_out(MX1),V(MX1),Z(MX1)
      COMMON /CC/QQ(MX2),VV(MX2),ZZQ(MX2),ZZV(MX2)


      K=1
      Z(1)=Zzero
      CALL IP(ZZV,VV,Z(1),V(1),N2)


500   Q1=Q_in(K)
      Q2=Q_in(K+1)
      V1=V(K)
      IN=1
      D1=Q_out(K)
      D2=D1
1000  V2=V1+DT*3600*(0.5*(Q1+Q2)-0.5*(D1+D2))
      CALL IP(VV,ZZV,V2,Z_CURVE,N2)
      CALL IP(ZZQ,QQ,Z_CURVE,D_CURVE,N1)
      IF (ABS(D2-D_CURVE).LE.1.0E-2) THEN
          Q_out(K+1)=D2
      ELSE
          D2=0.5*(D2+D_CURVE)
          IN=IN+1
          IF (IN.GT.2000) THEN
              DATAQ=ABS(D2-D_CURVE)
              IF(DATAQ.GT.1.0) WRITE(*,*)' Iterative calculation does not meet the
convergence accuracy requirements, please check！ △q= ', DATAQ
              GOTO 1100
          END IF
          GOTO 1000
      END IF
1100  V(K+1)=V2
      Z(K+1)=Z_CURVE

      K=K+1
      IF (K.LT.IQ) GOTO 500

      CALL ORDER(IQ,Z_MAX)
      END
!     ---------------------------------------------------------------
```

```
 !
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+
!                                   Z_MAX
!
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+
        subroutine ORDER(IQ,Z_MAX)
        implicit real(I-N)
        parameter(MX1=1000)

        integer IQ,I,J
        real Z_MAX
        common /FRC/Q_in(MX1),Q_out(MX1),V(MX1),Z(MX1)

        Z_MAX=Z(1)
        do I=2,IQ
            if (Z(I)>Z_MAX) then
                Z_MAX=Z(I)
            end if
        end do
        end
!       ----------------------------------------------------------------------
```