

## ภาคผนวก ก

## โปรแกรม Storage Routing Model

```

c*****
c storage routing model with loss mechanisms
c*****
c input data (mm) and (mm/day) for daily
c rainfall and discharge
c*****
c storage-discharge relationship
c  $s = k11*q^{p1} + k12*d(q^{p2})/dt$ 
c  $ds/dt = r - q - b$ 
c  $k11 = c11*A^{0.24}$ 
c  $k12 = c12*k11^2*(rave)^{-0.2648}$ 
c rave = average rainfall intensity
c  $b = k13*q = (c13-1)*q$ 
c optimization of parameters c11 and c12 and c13
c p1 = 0.6 and p2 = 0.4648
c -----
c character*128,infile*40,gp*128,title1*128,title2*128
& ,title3*128
c
c dimension q(200),pas(200,9),dpa(9),qch(200),rch(200)
c dimension x(9),u(18),er(200),z(9)
c dimension co(9)
c
c common/st1/wk1,wk2,wp1,wp2,wbeta,rain,q0in
c common/st2/n,m,m1,m2,h,h2,h3,h4
c common/date/iy,im,id,ih,nq,nr,area
c common/discha/r(200),q0(200),qc(200),q00(200)
c open(1,file='input-one-tank.txt',status='unknown')

```

```

open(6,file='output-one-tank.txt',status='unknown')
c
data cp1,cp2/0.6,0.4648/
data yk1,yk2,vp1,vp2,vbeta/1.0,1.0,1.0,1.0,1.0/
c
read(1,*) ncase
write(6,'(i5)') ncase
c
kkput=0
do 9999 kk=1,ncase
kkk=kk
read(1,3) title1
read(1,3) title2
read(1,3) title3
read(1,3) title
read(1,'(f8.0,i5)') area,ipas
c area : drainage area (km2)
c ipas : flag for computation
c ipas=1 : execution of computation
c ipas=0 : no computation
read(1,4) n,m,nl,kount,p,rave
c n : order of differential equation (n=2)
c m : number of unknown parameters (m=3)
c nl : number of computational time divisions for a day
c if nl=24, computational time step = 1/24 day (one hour)
c if nl=12, computational time step = 1/12 day (two hours)
c kount : maximum number of iterations (kount=20)
c p : permissible convergence limit (p=0.01 or 0.001)
c rave : average rainfall intensity

```

```

read(1,5) cc1,cc2,c3
c  cc1,cc2,c3 : initial values of c11, c12 and c13
read(1,'(2i8)') nr,nq
c  nr, nq : number of rainfall data and discharge data
do i=1,nq
r(i)=0.0
end do
read(1,2) (r(i) ,i=1,nr)
c  r(i) : daily rainfall series in (mm/day)
read(1,2) (q(i),i=1,nq)
c  q(i) : daily runoff depth series in (mm/day)
do 1001 i=1,nq
1001 q0(i)=q(i)*area/86.4
c  q0(i) : daily discharge data in (m3/sec)
c
if(ipas.eq.0) go to 9999
c
rtot=0.0
nze=0
do 1003 i=1,nr
if(r(i).eq.0.0) go to 1003
nze=nze+1
rtot=rtot+r(i)
1003 continue
xnze=nze
rave=rtot/xnze
c  normalized parameters
yk1=1.0
yk2=1.0

```

```

vp1=1.0
vp2=1.0
vbeta=1.0
c  initial values of parameters
co(1)=cc1
co(2)=cc2
co(3)=c3
c
1 format(4i8,4f8.0)
2 format(10f8.0)
3 format(a128)
4 format(4i5,2f8.0)
5 format(3f8.0)
c
write(6,'(i5)') kk
write(6,'()')
write(6,200) title1
write(6,200) title2
write(6,200) title3
write(6,200) title
write(6,'()')
write(6,('numbur of discharge data=",i10)') nq
write(6,'()')
write(6,210)
210 format(' area',' average rain.',' ini c11',' ini c12',
& ' ini c13 ')
write(6,'(f10.1,f15.2,3f10.3)') area,rave,cc1,cc2,c3
write(6,'()')
200 format(a80)

```

```

c
  xnl=nl
  h=1./xnl
  h2=h**2
  h3=h2*h
  h4=h3*h
  m1=m+1
  m2=2*m
  xnq=nq
c  ram : recession coefficient of groundwater flow component
c  ram = ln(q1/q2)/(t2-t1) in the falling limb of hydrograph
c  Tc = 1/ram (separation time for groundwater flow)
c  which is used in "storage routing model with groundwater
c  flow component"
c  tentative value of ram
c
  ram=0.12
c
  q01=q(1)
  do 20 i=1,nq
    q02=q(1)*exp(-ram*i)
    q00(i)=(q01+q02)*0.5
    q01=q02
  20 continue
c  q00(i) : baseflow component(forcing inputs)
c
c  start optimization
c
  fac=0.0

```

```

do 999 kll=1,kount
wk1=cc1*yk1*area**0.24
zk2=cc2*yk2*wk1**2*rave**(-0.2648)
wk2=1./zk2
wp1=cp1*vp1
zp2=cp2*vp2
wp2=1./zp2
wbeta=c3*vbeta
do 800 i=1,n
800  x(i)=0.0
      x(1)=q(1)**zp2
c    x(1) and x(2) : initial values of state variables
      do 802 i=1,m2
802  u(i)=0.0
c    u(i) : initial values of sensitivity coefficients
      sum=0.0
      sotai=0.0
c
      do 30 ll=1,nq
      rain=0.0
      if(ll.le.nr) rain=r(ll)
      q0in=q00(ll)
      qq=q(ll)
      qs=0.0
      if(qq.gt.0.0) qs=sqrt(qq)
      do 40 k=1,nl
c**  solution of sensitivity equation
      call gesto(x,u,1)
c**  solution of differential equation

```

```

call gesto(x,u,2)
if(x(1).le.0.0) x(1)=0.0
40  continue
qcc=0.0
if(x(1).gt.0.0) qcc=x(1)**wp2
cc=0.0
if(x(1).gt.0.0) cc=wp2*x(1)**(wp2-1.)
cd=0.0
if(x(1).gt.0.0) cd=alog(x(1))
qc(ll)=qcc
c  qc : computed runoff depth series in (mm/day)
err=qq-qcc
er(ll)=err
erw=err/qs
c  erw=err
c  erw=err/qq
sum=sum+err**2
sotai=sotai+abs(err)/qq
c** sensitivity coefficients
c
pas(ll,1)=cc*u(1)*area**0.24
pas(ll,2)=cc*u(2)*wk1**2*rave**(-0.2648)
pas(ll,3)=cc*u(3)
do 43 i=1,m
43  pas(ll,i)=pas(ll,i)*co(i)
do 44 i=1,m
44  pas(ll,i)=pas(ll,i)/qs
c** error between observed and computed discharges
pas(ll,m1)=erw

```

```

30  continue
c
    sum=sqrt(sum/xnq)
    sotai=sotai/xnq
c  sum : root mean squares error
c  sotai : kai squares criterion used in statistical test
c
    z(1)=yk1
    z(2)=yk2
    z(3)=vbeta
c** component regression method
c** to solve the correction terms of parameters
    call momreg(9,m1,200,nq,pas,dpa)
c  dpa : correction terms of parameters c11, c12 and c13
c
    do 875 i=1,m
        col = abs(dpa(i)/z(i))
        if(col.gt.p) go to 877
875  continue
        go to 879
877  fac = 0.5*(1.+fac)
c  fac : relaxation coefficient to prevent divergence
    do 878 i=1,m
        dpa(i)=fac*dpa(i)
878  continue
    do 991 i=1,m
        xnew=z(i)+dpa(i)
        if(xnew.gt.0.0) z(i)=z(i)+dpa(i)
991  continue

```

```

c   new values of normalized parameters
    yk1=z(1)
    yk2=z(2)
    vbeta=z(3)
999 continue
879 continue
c
    do 500 i=1,nq
        q00(i)=q00(i)*area/86.4
500 qc(i)=qc(i)*area/86.4
c   q00(i) : baseflow data in (m3/sec)
c   qc(i) : daily discharge data in (m3/sec)
    qtot=0.0
    qctot=0.0
    q00tot=0.0
    qsotai=0.0
    qmax=-999.99
    qmaxc=-999.99
    do 501 i=1,nq
        qtot=qtot+q0(i)
        qctot=qctot+qc(i)
        q00tot=q00tot+q00(i)
        qsotai=qsotai+abs(q0(i)-qc(i))/q0(i)
        if(q0(i).gt.qmax) qmax=q0(i)
        if(qc(i).gt.qmaxc) qmaxc=qc(i)
501 continue
    qtot0=86.4*qtot/area
    qctot0=86.4*qctot/area
    q00tot0=86.4*q00tot/area

```

```

c
  qmax0=86.4*qmax/area
  qmaxc0=86.4*qmaxc/area
  hiryu=qmax/area
c  hiryu : specific peak discharge(peak flow/area (m3/sec/km2))
c  hiryu should be larger than 0.5 in flood runoff analysis
  peakg = abs(qmax0-qmaxc0)/qmax0
c  Jpe = peakg (relative error of peak values)
c  Jre = qsotai (relative error of hydrographs)
  qsotai=qsotai/float(nq)
c
  write(6,211)
211 format(5x,'c11',7x,'c12',7x,'c13',7x,'Jpe',7x,'Jre',7x,'iterations
  & ')
  write(6,'(5f10.3,i10)') cc1*yk1,cc2*yk2,c3*vbeta,
  & peakg,qsotai,kkll
  write(6,'()')
c
  write(6,212)
212 format('  specific dis.  total rain.  total obser.dis. total',
  & ' com.  dis.obser.peak  com.peak')
  write(6,'(7f15.3)') hiryu,rtot,qtot0,qctot0,qmax,qmaxc
c
  write(6,'()')
c  unit of r(i):mm/day, q0(i),qc(i),q00(i),loss discharge:m3/s
  write(6,204)
204 format('  NO',12x,'rain',8x,'obser. dis.',5x,'com. dis.'
  & ,8x,'baseflow',8x,'loss dis.')
  do i=1,nq

```

```

        write(6,207) i,r(i),q0(i),qc(i),q00(i),q0(i)*(c3*vbeta-1.0)
    end do
207 format(i8,f16.2,5f16.2)
c
9999 continue
c
    stop
    end
c
c
    subroutine gesto(x,u,ijk)
c** s = k1*q**p1 + k2*d(q**p2)/dt
c** x(k+1)= phi*x(k) + gamma*b(k)
c** ijk = 1; solve sensitivity equation
c** ijk = 2; solve differential equation
    dimension x(1),u(1),y(9),uu(18),b(9)
    common/st1/wk1,wk2,wp1,wp2,wbeta,rain,q0in
    common/st2/n,m,m1,m2,h,h2,h3,h4
c
    con=wk1*wk2*wp1*wp2
    a=0.0
    c=0.0
    d=0.0
    e=0.0
    f=0.0
    y1=x(1)
    y2=x(2)
    if(y1.gt.0.0) go to 12
    if(y1.lt.0.0) y1=0.0

```

```

go to 14
12  a=y1**(wp1*wp2-2.)
    c=y1**(wp2-1.)
    d=y1**(wp1*wp2-1.)
    e=y1**wp2
    f=alog(y1)
14  a1=-con*(wp1*wp2-1.)*a*y2-wk2*wp2*c*wbeta
    a2=-con*d
    a3=a1+a2**2
    a4=a1+a3
c** elements of phi matrix (transition matrix)
    f1 = 1. +0.5*a1*h2+a1*a2*h3/6.+a1*a3*h4/24.
    f2 = h*(1. + 0.5*a2*h+a3*h2/6.+a2*a4*h3/24.)
    f3 = a1*f2
    f4 = 1. +a2*h+0.5*a3*h2+a2*a4*h3/6.+
    & (a1*a3+a2**2*a4)*h4/24.
c** elements of gamma matrix
    g2 = h2*(0.5+a2*h/6.+a3*h2/24.)
    g4=f2
    if(ijk.eq.2) go to 104
c** solve sensitivity equation
    b(1)=-wk2*wp1*wp2*d*y2
    b(2)=wk2**2*(wk1*wp1*wp2*d*y2 + e*wbeta - rain-q0in)
    b(3)=-wk2*e
    do 16 i=1,m
16  uu(i)= f1*u(i) + f2*u(i+m) + g2*b(i)
    do 18 i=m1,m2
18  uu(i) = f3*u(i-m) + f4*u(i) + g4*b(i-m)
    do 20 i=1,m2

```

```

20  u(i)=uu(i)
    return
104 continue
c** solve system equation
    b1=con*(wp1*wp2-1.)*d*y2+wk2*wbeta*(wp2-1.)*e+wk2*(rain+q0in)
    y(1) = f1*x(1)+f2*x(2)+g2*b1
    y(2) = f3*x(1)+f4*x(2)+g4*b1
    do 102 i=1,n
102  x(i)=y(i)
    return
    end
c
c
    subroutine momreg(n1,n,m1,md,x,dpa)
c** component regression method
c** compute the correction terms of parameters (dpa)
    dimension x(m1,n1),dpa(1),cov(9,9),g(200,9),y(200,9)
    dimension binv(9,9),coe(9),st(9)
c
    na=n-1
c** compute covariance matrix
    call sqcov(n1,na,m1,md,x,cov)
    do 50 i=1,na
50  st(i)=sqrt(cov(i,i))
    do 52 i=1,na
    s=st(i)
    do 52 j=1,i
    s1=st(j)
    cov(i,j)=cov(i,j)/(s*s1)

```

```

52  cov(j,i)=cov(i,j)
c** factorization of cov(i,j) by lower triangular
c** cholesky method (cov = l * u)
c** l = lower triangular u = upper triangular
c** compute the inverse of u(i,j)
    call lowtri(n1,na,cov,binv)
    do 54 j=1,na
        s=st(j)
        do 54 i=1,md
54  y(i,j)=x(i,j)/s
        do 20 i=1,md
            do 20 j=1,na
                s=0.
                do 22 k=1,j
22  s=s+y(i,k)*binv(k,j)
20  g(i,j)=s
            do 24 i=1,na
                s=0.0
                do 26 j=1,md
26  s=s+g(j,i)*x(j,n)
24  coe(i)=s
            do 30 i=1,na
                s=0.
                do 29 j=i,na
29  s=s+binv(i,j)*coe(j)
30  dpa(i)=s/st(i)
        return
    end
c

```

```

c
    subroutine lowtri(n1,n,p,binv)
c** lower triangular cholesky factorization
c** p = u*b
c** p = symmetric matrix
c** u = lower triangular matrix
c** b = upper triangular matrix (b = ut)
c** binv = inverse matrix of b
c**      = upper triangular matrix
c** compute lower triangular u(i,j)
c
c    implicit double precision (a-h,o-z)
    dimension p(n1,n1),binv(n1,n1)
    dimension u(9,9),b(9,9)
c
    do 5 j=1,n-1
        u(j,j)=abs(p(j,j))
        u(j,j)=sqrt(u(j,j))
        al=1./u(j,j)
        do 5 k=n,j+1,-1
            u(k,j)=al*p(k,j)
            be=u(k,j)
        do 5 i=k,n
5    p(i,k)=p(i,k)-u(i,j)*be
        u(n,n)=abs(p(n,n))
        u(n,n)=sqrt(u(n,n))
c
c** b = transpose of u
    do 40 i=1,n

```

```

do 40 j=i,n
40  b(i,j)=u(j,i)
c
c**  compute inverse of b(i,j)
    binv(1,1)=1./b(1,1)
    do 50 j=2,n
        binv(j,j)=1./b(j,j)
        jm1=j-1
        do 50 k=1,jm1
            sum=0.0
            do 52 i=k,jm1
52  sum = sum - binv(k,i)*b(i,j)
50  binv(k,j)=sum*binv(j,j)
    return
    end
c
c
    subroutine sqcov(n1,n,m1,md,x,cov)
c**  compute covariance matrix
c    implicit double precision (a-h,o-z)
    dimension x(m1,n1),cov(n1,n1)
    do 10 i=1,n
        do 10 j=1,i
            s=0.
            do 12 k=1,md
12  s=s+x(k,i)*x(k,j)
10  cov(i,j)=s
    return
    end

```