

01-1program. for. txt

```

C *****
C 第1回ゼミ
C   三角化法による連立方程式の解法
C   三角化法(LU分解法・非対称行列)による逆行列の計算
C
C 作成者 伊賀 久晃      作成日 2004. 5. 2
C *****
C
      dimension a(10, 10), s(10, 10), ia(10, 10), b(10, 1), y(10, 1), x(10, 1)
      dimension l(10, 10), u(10, 10), il(10, 10), iu(10, 10), d(10, 10)

      real a, l, u, s, il, iu, ia, b, y, x, d

      integer i, j, n

      open(1, file='input01.txt', status='old')
      open(2, file='input02.txt', status='old')
C
      read(1, '(2x, i8)') n

      write(*, *) 'A行列'

      do 1 i=1, n
        read(1, '(10f8.4)') (a(i, j), j=1, n)
        read(2, '(f8.4)') b(i, 1)
        write(*, '(10f8.4)') (a(i, j), j=1, n)
1 continue

      call crout(l, u, a, n)
      call seki(s, l, u, n)
      call invl(il, l, n)
      call invu(iu, u, n)

      write(*, *) ' (問1)  A行列のLU分解後のL行列 '
      do 2 i=1, n
        write(*, '(10f8.4)') (l(i, j), j=1, n)
2 continue

      write(*, *) ' A行列のLU分解後のU行列 '
      do 3 i=1, n
        write(*, '(10f8.4)') (u(i, j), j=1, n)
3 continue

      write(*, *) ' L行列とU行列の積S行列 (検証) '
      do 4 i=1, n
        write(*, '(10f8.4)') (s(i, j), j=1, n)
4 continue

      write(*, *) ' (問2)  L行列の逆行列IL行列 '
      do 5 i=1, n
        write(*, '(10f8.4)') (il(i, j), j=1, n)
5 continue

      call seki(s, l, il, n)

      write(*, *) ' L行列とL行列の逆行列IL行列の積S行列 (検証) '
      do 6 i=1, n
        write(*, '(10f8.4)') (s(i, j), j=1, n)

```

```

6 continue

  write(*,*) 'U行列の逆行列IU行列'
  do 7 i=1, n
    write(*, ' (10f8. 4)') (iu(i, j), j=1, n)
7 continue

  call seki(s, u, iu, n)

  write(*,*) 'U行列とU行列の逆行列IU行列の積S行列 (検証)'
  do 8 i=1, n
    write(*, ' (10f8. 4)') (s(i, j), j=1, n)
8 continue

  call seki(ia, iu, il, n)

  write(*,*) '(問3) A行列の逆行列IA行列をIU行列とIL行列より算出'
  do 9 i=1, n
    write(*, ' (10f8. 4)') (ia(i, j), j=1, n)
9 continue

  call seki(y, il, b, n)

  write(*,*) '(問4) IL行列とBベクトルよりYベクトルを算出'
  do 900 i=1, n
    write(*, ' (10f8. 4)') y(i, 1)
900 continue

  call seki(x, iu, y, n)

  write(*,*) 'IU行列とYベクトルよりXベクトルを算出'
  do 910 i=1, n
    write(*, ' (10f8. 4)') x(i, 1)
910 continue

  call inverse(n, a, d)

  write(*,*) '(問5) クラウト法による逆行列を求める
&サブルーチンを用いた結果'
  do 920 i=1, n
    write(*, ' (10f8. 4)') (d(i, j), j=1, n)
920 continue

  call seki(s, ia, b, n)

  write(*,*) '(問6) BベクトルにIA行列を乗じてXベクトルを算出'
  do 930 i=1, n
    write(*, ' (10f8. 4)') s(i, 1)
930 continue
  stop
  end
c
c-----
c  subroutine crout(l, u, a, n)
c  クラウト法を用いLU分解するサブルーチン
  dimension a(10, 10), l(10, 10), u(10, 10), d(10, 10)
  real a, l, u, d
  integer i, j, k, n

```

```

do 10 i=1, n
  do 10 j=1, n
    d(i, j)=a(i, j)
    if(i. le. j) then
      do 11 k=1, i-1
        d(i, j)=d(i, j)-d(i, k)*d(k, j)
11      continue
        if(i. lt. j) d(i, j)=d(i, j)/d(i, i)
      else
        do 12 k=1, j-1
          d(i, j)=d(i, j)-d(i, k)*d(k, j)
12        continue
        end if
      10 continue
c   L行列とU行列に要素を分ける
  do 13 i=1, n
    do 13 j=1, n
      if(i. ge. j) then
        l(i, j)=d(i, j)
        if(i. eq. j) then
          u(i, j)=1. 00
        else
          u(i, j)=0. 00
        end if
      else
        u(i, j)=d(i, j)
        l(i, j)=0. 00
      end if
    13 continue

    return
  end

```

```

c
c -----
c   subroutine seki (s, l, u, n)
c  行列の乗算を行うサブルーチン
  dimension s(10, 10), l(10, 10), u(10, 10)
  real      s, l, u
  integer  i, j, k, n

  do 20 i=1, n
    do 20 j=1, n
      s(i, j)=0. 0
      do 20 k=1, n
        s(i, j)=s(i, j)+l(i, k)*u(k, j)
20    continue

    return
  end

```

```

c
c -----
c   subroutine invl (il, l, n)
c  下三角の逆行列を求めるサブルーチン
  dimension il(10, 10), l(10, 10)
  real      il, l
  integer  i, j, k, n

```

```

do 30 i=1, n
  do 30 j=1, n
    if(i. eq. j) il(i, j)=1. 0/l(i, j)
    if(i. lt. j) il(i, j)=0. 0
    if(i. gt. j) then
      il(i, j)=0. 0
      do 31 k=1, i-1
        il(i, j)=il(i, j)+l(i, k)*il(k, j)
31      continue
        il(i, j)=-il(i, j)/l(i, i)
      end if
30 continue

  return
end

c
c-----
subroutine invu(iu, u, n)
c 上三角の逆行列を求めるサブルーチン
dimension iu(10, 10), u(10, 10)
real iu, u
integer i, j, k, n

do 40 i=1, n
  do 40 j=1, n
    if(i. eq. j) iu(i, j)=1. 0
    if(i. gt. j) iu(i, j)=0. 0
    if(i. lt. j) then
      iu(i, j)=0. 0
      do 41 k=i, j-1
        iu(i, j)=iu(i, j)-iu(i, k)*u(k, j)
41      continue
      end if
40 continue

  return
end

c
c-----
subroutine inverse(n, a, d)
c クラウト法による逆行列を求めるサブルーチン
c a: 入力行列
c d: 行列aの逆行列
dimension a(10, 10), b(10), c(10), d(10, 10)

do 154 i=1, n
  do 151 j=1, n
    b(j)=a(j, i)
151 a(j, i)=0. 0

  a(i, i)=1. 0

  do 152 j=1, n
152 c(j)=a(i, j)/b(i)

  do 153 j=1, n
    do 153 k=1, n
153 a(j, k)=a(j, k)-c(k)*b(j)

```

01-1program. for. txt

```
do 154 j=1, n
154 a(i, j)=c(j)
do 156 i=1, n
do 156 j=1, n
156 d(i, j)=a(i, j)
return
end
```

input01.txt

4  
1.0000 1.0000 0.0000 3.0000  
2.0000 1.0000 -1.0000 1.0000  
3.0000 -1.0000 -1.0000 2.0000  
-1.0000 2.0000 3.0000 -1.0000

input02.txt

4.0000  
1.0000  
-3.0000  
4.0000

## 01-2program. for. txt

```

C *****
C 第1回ゼミ
C   三角化法による連立方程式の解法
C   三角化法(コレスキー分解法・対称行列)による逆行列の計算
C
C 作成者 東海林 勉      作成日 2004. 5. 10
C *****
C
C   real a(10, 10), b(10), t(10, 10), s(10, 10),
C   &    a1(10, 10), b1(10), aa(10, 10), tt(10, 10),
C   &    ss(10, 10), y(10), x(10), d(10, 10)
C
C   open(5, file='input2. dat', status='old')
C   read(5, *) n
C
C**** 三角行列要素ゼロ設定*****
C
C   do 120 i=1, n
C   do 120 j=1, n
C       t(i, j)=0.0
C       ss(i, j)=0.0
C   120 continue
C
C**** データの読み込み*****
C
C   write(6, *) 'A 行列'
C   do 1 i=1, n
C   read(5, *) (a(i, j), j=1, n)
C   write(6, '(10f8.4)') (a(i, j), j=1, n)
C   1 continue
C
C   write(6, *) 'B ベクトル'
C   read(5, *) (b(i), i=1, n)
C   do 2 i=1, n
C   write(6, '(10f8.4)') b(i)
C   2 continue
C
C**** (1) 下三角行列Λの計算*****
C   t: 下三角行列Λ
C
C   do 3 j=1, n-1
C   t(j, j)=a(j, j)
C   t(j, j)=sqrt(t(j, j))
C   al=1.0/t(j, j)
C   do 3 k=n, j+1, -1
C   t(k, j)=al*a(k, j)
C   t1=t(k, j)
C   do 3 i=k, n
C   a(i, k)=a(i, k)-t(i, j)*t1
C   3 continue
C   t(n, n)=a(n, n)
C   t(n, n)=sqrt(t(n, n))
C
C   write(6, *) '(1) 下三角行列 Λ'
C   do 4 i=1, n
C   write(6, '(10f8.4)') (t(i, j), j=1, n)
C   4 continue
C

```



```

c***** 上三角行列  $\Lambda T$  の計算 ( $\Lambda$  の転置) *****
c   s: 上三角行列  $\Lambda T$ 
c
      do 10 i=1, n
      do 10 j=1, n
        s(i, j)=t(j, i)
10    continue
      write(6, *) ' 上三角行列  $\Lambda T$ '
      do 15 i=1, n
        write(6, '(10f8.4)') (s(i, j), j=1, n)
15    continue
c
c***** (2) チェック { $\Lambda * \Lambda T = A$ } *****
c
      do 20 i=1, n
      do 20 j=1, n
        a2=0.0
        do 21 k=1, n
          a2=t(i, k)*s(k, j)+a2
21    continue
      a1(i, j)=a2
20    continue
c
      write(6, *) '(2) チェック A行列'
      do 25 i=1, n
        write(6, '(10f8.4)') (a1(i, j), j=1, n)
25    continue
c
c***** (3) 上三角逆行列計算  $\Lambda T^{-1}$  ***
c   ss: 上三角逆行列  $\Lambda T^{-1}$ 
c
      ss(1, 1)=1.0/s(1, 1)
      do 30 j=2, n
        ss(j, j)=1.0/s(j, j)
        m=j-1
        do 30 k=1, m
          ss1=0.0
          do 31 i=k, m
            ss1=ss1-ss(k, i)*s(i, j)
31    continue
          ss(k, j)=ss1*ss(j, j)
30    continue
c
      write(6, *) '(3) 上三角逆行列  $\Lambda T^{-1}$ '
      do 35 i=1, n
        write(6, '(10f8.4)') (ss(i, j), j=1, n)
35    continue
c
c***** 下三角逆行列計算  $\Lambda^{-1}$  *****
c   tt: 下三角逆行列  $\Lambda^{-1}$ 
c
      do 40 i=1, n
      do 40 j=1, n
        tt(i, j)=ss(j, i)
40    continue
c
      write(6, *) ' 下三角逆行列  $\Lambda^{-1}$ '
      do 45 i=1, n

```

```

                                01-2program.for.txt
        write(6,' (10f8.4)') (tt(i,j),j=1,n)
45 continue
C
C***** (4) Aの逆行列の計算 *****
C   aa:Aの逆行列 A-1
C
        do 60 i=1,n
            do 60 j=1,n
                aa1=0.0
                do 61 k=1,n
                    aa1=ss(i,k)*tt(k,j)+aa1
61          continue
                aa(i,j)=aa1
60          continue
C
        write(6,*) '(4) A逆行列 A(-1)'
        do 65 i=1,n
            write(6,' (10f8.4)') (aa(i,j),j=1,n)
65          continue
C
C***** (5) ベクトルYの計算 *****
C
C
        do 80 i=1,n
            y1=0.0
            do 81 k=1,i
                y1=tt(i,k)*b(k)+y1
81          continue
            y(i)=y1
80          continue
C
        write(6,*) '(5) ベクトルY'
        do 85 i=1,n
            write(6,' (10f8.4)') y(i)
85          continue
C
C***** ベクトルXの値 *****
C
        do 90 i=1,n
            x1=0.0
            do 91 k=i,n
                x1=ss(i,k)*y(k)+x1
91          continue
            x(i)=x1
90          continue
C
        write(6,*) ' ベクトルX'
        do 95 i=1,n
            write(6,' (10f8.4)') x(i)
95          continue
C
C***** (6) チェック AX=B *****
C
        do 100 i=1,n
            b2=0.0
            do 101 k=1,n
                b2=a1(i,k)*x(k)+b2
101         continue

```

```

        b1(i)=b2
100 continue
C
        write(6,*) '(6) チェック B'
        do 105 i=1, n
            write(6, '(10f8.4)') b1(i)
105 continue
C
C***** (7) 一般的な方法で求めた結果 *****
C
        call inverse(n, a, d)

        write(6,*) '(7) 一般的な方法で求めた結果'
        do 110 i=1, n
            write(6, '(10f8.4)') (d(i, j), j=1, n)
110 continue
C
C***** (8) ベクトルXを $X=A^{-1}B$ より算出*****
C
        do 130 i=1, n
            do 130 j=1, n
                s(i, j)=0.0
                do 130 k=1, n
                    s(i, j)=s(i, j)+d(i, k)*b(k)
130 continue

        write(6,*) '(8) ベクトルX'
        do 140 i=1, n
            write(6, '(10f8.4)') s(i, 1)
140 continue

        stop
        end

C-----
C      subroutine inverse(n, a, d)
C      一般的な逆行列を求めるサブルーチン
C      a: 入力行列
C      d: 行列aの逆行列
        dimension a(10, 10), b(10), c(10), d(10, 10)

        do 154 i=1, n
            do 151 j=1, n
                b(j)=a(j, i)
151      a(j, i)=0.0

            a(i, i)=1.0

            do 152 j=1, n
152      c(j)=a(i, j)/b(i)

            do 153 j=1, n
                do 153 k=1, n
153      a(j, k)=a(j, k)-c(k)*b(j)

            do 154 j=1, n
154      a(i, j)=c(j)

```

01-2program. for. txt

```
do 156 i=1, n
  do 156 j=1, n
156 d(i, j)=a(i, j)
  return
end
```

input2.txt

```
5
9 -6 6 0 0
-6 5 0 1 0
6 0 29 4 -9
0 1 4 5 -2
0 0 -9 -2 26
7
5
4
6
8
```

## 01-3program. for. txt

```

C *****
C 第1回ゼミ
C   三角化法による連立方程式の解法
C   三角化法(変形コレスキー分解)による逆行列の計算
C
C 作成者 東海林 勉      作成日 2004. 5. 17
C *****
C
C   real a(10, 10), b(10, 10), t(10, 10), tt, d(10, 10), dl(10, 10),
&   u(10, 10), r(10, 10), dd(10, 10), s(10, 10),
&   rr(10, 10), dy(10, 10), dinv(10, 10), y(10, 10),
&   uinv(10, 10), x(10, 10), bchk(10, 10)
C
C   open(1, file='input3. dat', status='old')
C   read(1, *) n
C
C**** 三角行列要素ゼロ設定*****
C
C   do 120 i=1, n
C   do 120 j=1, n
C       t(i, j)=0. 0
C   120 continue
C
C**** データの読み込み*****
C
C   write(6, *) 'A 行列'
C   do 1 i=1, n
C   read(1, *) (a(i, j), j=1, n)
C   write(6, '(10f8. 4)') (a(i, j), j=1, n)
C   1 continue
C
C   write(6, *) 'B ベクトル'
C   do 2 i=1, n
C   read(1, *) b(i, 1)
C   write(6, '(10f8. 4)') b(i, 1)
C   2 continue
C
C**** A=LDU *****
C
C   do 10 i=1, n
C   do 10 j=1, n
C   if(i. ge. j) then
C   if(i. eq. j) then
C       if(i. eq. 1. and. j. eq. 1) then
C           t(i, j)=a(i, j)
C       else
C           tt1=0. 0
C           do 11 k=1, i-1
C           tt=t(i, k)**2. 0*t(k, k)
C           tt1=tt+tt1
C   11 continue
C           t(i, j)=a(i, j)-tt1
C       end if
C   else
C       if(j. eq. 1) then
C           t(i, j)=a(i, j)/t(1, 1)
C       else
C           tt1=0. 0

```

```

        do 12 k=1, i-1
            tt=t(i, k)*t(j, k)*t(k, k)
            tt1=tt+tt1
12      continue
        t(i, j)=(a(i, j)-tt1)/t(j, j)
        end if
    end if
end if
10 continue
C
C**** L・D・U の計算 *****
C
    do 50 i=1, n
        do 50 j=1, n
            if(i. eq. j) then
                d(i, j)=t(i, j)
            else
                dl(i, i)=1.0
                dl(i, j)=t(i, j)
            end if
50      continue
C
    call tench(n, dl, u)
C
    write(6, *) '▽(1)-----'
    write(6, *) 'L'
    do 60 i=1, n
        write(6, '(10f8. 4)') (dl(i, j), j=1, n)
60      continue
C
    write(6, *) 'D'
    do 55 i=1, n
        write(6, '(10f8. 4)') (d(i, j), j=1, n)
55      continue
C
    write(6, *) 'U'
    do 110 i=1, n
        write(6, '(10f8. 4)') (u(i, j), j=1, n)
110     continue
C
C**** D^(1/2) *****
C dd:D^(1/2)
    write(6, *) 'D^1/2'
    do 111 i=1, n
        do 111 j=1, n
            dd(i, j)=sqrt(d(i, j))
111     continue
C
C **** Λ=LD^(1/2) *****
    call matrix(n, dl, dd, r)
    write(6, *) '▽(2)-----'
    write(6, *) 'Λ=LD^(1/2)'
    do 100 i=1, n
        write(6, '(10f8. 4)') (r(i, j), j=1, n)
100     continue
C
C **** ΛT=D^(1/2)L^T *****
    call matrix(n, dd, u, rr)

```

```

                                01-3program. for. txt
write(6,*) '  $\Lambda^T = D^{(1/2)} L^T$ '
do 122 i=1, n
write(6, '(10f8.4)') (rr(i, j), j=1, n)
122 continue

c**** L-1 *****
call inverse(n, dl, s)
c**** Y' *****
call matrix(n, s, b, dy)
write(6,*) '  $\nabla(3)$ -----',
write(6,*) ' Y'
do 135 i=1, n
write(6, '(10f8.4)') (dy(i, j), j=1, n)
135 continue
c**** D-1 *****
call inverse(n, d, dinv)
c**** Y *****
call matrix(n, dinv, dy, y)
write(6,*) ' Y'
do 137 i=1, n
write(6, '(10f8.4)') y(i, 1)
137 continue
c**** U-1 *****
call tench(n, s, uinv)
c**** X *****
call matrix(n, uinv, y, x)
write(6,*) ' X'
do 139 i=1, n
write(6, '(10f8.4)') x(i, 1)
139 continue
c
c**** チェック AX=B *****
call matrix(n, a, x, bchk)
write(6,*) '  $\nabla(4)$ -----',
write(6,*) ' B'
do 140 i=1, n
write(6, '(10f8.4)') bchk(i, 1)
140 continue

stop
end

c*****
c      転 置
c*****
subroutine tench(n, bm, am)
real am(10, 10), bm(10, 10)
do 70 i=1, n
do 70 j=1, n
am(i, j)=bm(j, i)
70 continue
return
end

c
c*****
c      行列の乗算
c*****
subroutine matrix(n, aa, bb, ab)
real aa(10, 10), bb(10, 10), ab(10, 10)

```



01-3program. for. txt

```
do 15 i=1, n
do 15 j=1, n
sum=0. 0
do 16 k=1, n
sum=aa(i, k)*bb(k, j)+sum
16 continue
ab(i, j)=sum
15 continue
return
end
c
C*****
c 逆行列の算定
C*****
subroutine inverse(n, cc, s)
real cc(10, 10), s(10, 10)
do 150 i=1, n
do 150 j=1, n
if(i. eq. j) then
s(i, j)=1. 0/cc(i, j)
else
sum1=0. 0
do 151 k=j, i-1
sum1=sum1+cc(i, k)*s(k, j)
151 continue
s(i, j)=-(sum1/cc(i, i))
end if
150 continue
return
end
c
```

input3.txt

```
5
9 -6 6 0 0
-6 5 0 1 0
6 0 29 4 -9
0 1 4 5 -2
0 0 -9 -2 26
7
5
4
6
8
```

02-1program. for. txt

```
C *****
C 第2回ゼミ
C 非線形方程式の数値解析法
C 1階ニュートン法
C
C 作成者 東海林 勉      作成日 2004. 5. 17
C *****
C 一次微係数利用
C
C f(x)=x**3+3*x**2-5=0
C
C      real x, f(10), df(10), dx(10)
C      integer n
C      eps=0.0001
C      x=1.0
C      n=10
C      write(6, '( '      x      dx      df(x)      f(x)'' )')
C
C      do k=1, n
C          f(k)=(x**3)+(3*x**2)-5
C          df(k)=3*x**2+6*x
C          dx(k)=-f(k)/df(k)
C          write(6, '(4f9.5)') x, dx(k), df(k), f(k)
C          if(abs(f(k)).lt.eps) go to 111
C          x=x+dx(k)
C      end do
C
C 111 write(6,*) k, '回'
C      stop
C      end
```

02-2program. for. txt

```

C *****
C 第2回ゼミ
C 非線形方程式の数値解析法
C ベイリー法
C
C 作成者 東海林 勉      作成日 2004. 5. 17
C *****
C 二次微係数利用
C
C f(x)=x**3+3*x**2-5=0
C
C      real x1, f(10), df(10), ddf(10), dff(10)
C      real aa, bb
C      integer n
C      eps=0.0001
C      x1=1.0
C      n=10
C      write(6, '( ' x      dx      df      ddf(x)  f(x)'' )')
C
C      do k=1, n
C          f(k)=(x1**3.0)+(3.0*x1**2.0)-5.0
C          df(k)=3.0*x1**2.0+6.0*x1
C          ddf(k)=6.0*x1+6.0
C          aa=(df(k)**2-2*ddf(k)*f(k))**0.5
C          bb=1/ddf(k)
C          dff(k)=(-df(k)+aa)*bb
C          write(6, '(6f9.5)') x1, dff(k), df(k), ddf(k), f(k)
C          if(abs(f(k)).le.eps) go to 111
C          x1=x1+dff(k)
C      end do
C
C 111 write(6, *) k, '回'
C      stop
C      end

```

02-3program. for. txt

```

C *****
C 第2回ゼミ
C 非線形方程式の数値解析法
C ニュートン法による連立方程式の解法(一般的逆行列法)
C
C 作成者 東海林 勉          作成日 2004. 5. 17
C *****
C 一次微係数利用
C
C F=x**2+y**2-4
C G=(x**2)/9+y**2-1
C -----
C      real a(10, 10), dGx, dGy, ainv(10, 10), ddx(10, 10),
C      &      b(10, 10)
C
C      x=1. 0
C      y=1. 0
C      n=2. 0
C      eps=0. 00001
C      write(6, *) ' x          y          dx          dy          F          G'
C
C      do 200 k=1, 20
C      F=x**2+y**2-4
C      G=(x**2)/9+y**2-1
C      dGx=(2*x)/9
C      dGy=2*y
C      dFx=2*x
C      dFy=2*y
C      a(1, 1)=dFx
C      a(1, 2)=dFy
C      a(2, 1)=dGx
C      a(2, 2)=dGy
C      b(1, 1)=-F
C      b(2, 1)=-G
C
C      call inverse(n, a, ainv)
C      call matrix(n, ainv, b, ddx)
C      4 continue
C      dx=ddx(1, 1)
C      dy=ddx(2, 1)
C      write(6, '(10f8. 5)') x, y, dx, dy, F, G
C      if(abs(F). le. eps. and. abs(G). le. eps) go to 111
C      x=x+dx
C      y=y+dy
C      200 continue
C      111 write(6, *) k, '回'
C      stop
C      end
C *****
C          逆行列の算定
C *****
C      subroutine inverse(n, a1, d)
C      a: 入力行列
C      d: 行列aの逆行列
C      dimension a1(10, 10), b(10), c(10), d(10, 10)
C      do 154 i=1, n
C      do 151 j=1, n
C      b(j)=a1(j, i)

```

```

c
151 a1(j, i)=0.0
    a1(i, i)=1.0
    do 152 j=1, n
152 c(j)=a1(i, j)/b(i)
    do 153 j=1, n
    do 153 k=1, n
153 a1(j, k)=a1(j, k)-c(k)*b(j)
    do 154 j=1, n
154 a1(i, j)=c(j)
c
    do 156 i=1, n
    do 156 j=1, n
156 d(i, j)=a1(i, j)
    return
    end
C*****
c          行列の乗算
C*****
    subroutine matrix(n, aa, bb, ab)
    real aa(10, 10), bb(10, 10), ab(10, 10)
c
    do 15 i=1, n
    do 15 j=1, n
    sum=0.0
    do 16 k=1, n
        sum=aa(i, k)*bb(k, j)+sum
16 continue
        ab(i, j)=sum
15 continue
    return
    end

```

02-4program. for. txt

```

C *****
C 第2回ゼミ
C 非線形方程式の数値解析法
C ニュートン法による連立方程式の解法(コレスキー分解法)
C
C 作成者 東海林 勉          作成日 2004. 5. 29
C *****
C 一次微係数利用
C
C F=16*x**2+5*x*y+50*y**2-74
C G=4*x**2+25*x*y**2+10*y-41
C-----
C      real a(10, 10), dGx, dGy, ainv(10, 10), ddx(10, 10),
      &      b(10, 10), u(10, 10), l(10, 10), iu(10, 10), il(10, 10)
C
C      write(6, *) ' <Newton法>'
      x=1. 0
      y=1. 0
      n=2. 0
      eps=0. 0001
C
C      do 100 i=1, 2
C
C      write(6, ' (a5, f5. 2, a5, f5. 2)') ' x=', x, '      y=', y
      write(6, *) ' x          y          dx          dy          F          G'
C
C      do 200 k=1, 20
      F=16. *x**2. +5. *x*y+50. *y**2. -74.
      G=4. *x**2. +25. *x*y**2. +10. *y-41.
      dFx=32. *x+5. *y
      dFy=5. *x+100. *y
      dGx=8. *x+25. *y**2.
      dGy=50. *x*y+10.
      a(1, 1)=dFx
      a(1, 2)=dFy
      a(2, 1)=dGx
      a(2, 2)=dGy
      b(1, 1)=-F
      b(2, 1)=-G
C
C      call crout(l, u, a, n)
      call invu(iu, u, n)
      call invl(il, l, n)
      call matrix(n, iu, il, ainv)
      call matrix(n, ainv, b, ddx)
      dx=ddx(1, 1)
      dy=ddx(2, 1)
      write(6, ' (10f8. 5)') x, y, dx, dy, F, G
      if(abs(F). le. eps. and. abs(G). le. eps) go to 111
      x=x+dx
      y=y+dy
200  continue
111  write(6, *) k, '回'
      write(6, *)
      x=2. 0
      y=1. 0
100  continue

```

```

stop
end
c
c-----
subroutine crout(l, u, a, n)
c クラウト法を用いLU分解するサブルーチン
dimension a(10, 10), l(10, 10), u(10, 10), d(10, 10)
real a, l, u, d
integer i, j, k, n

do 10 i=1, n
  do 10 j=1, n
    d(i, j)=a(i, j)
    if(i. le. j) then
      do 11 k=1, i-1
        d(i, j)=d(i, j)-d(i, k)*d(k, j)
11      continue
        if(i. lt. j) d(i, j)=d(i, j)/d(i, i)
      else
        do 12 k=1, j-1
          d(i, j)=d(i, j)-d(i, k)*d(k, j)
12      continue
        end if
      10 continue
c L行列とU行列に要素を分ける
do 13 i=1, n
  do 13 j=1, n
    if(i. ge. j) then
      l(i, j)=d(i, j)
      if(i. eq. j) then
        u(i, j)=1. 00
      else
        u(i, j)=0. 00
      end if
    else
      u(i, j)=d(i, j)
      l(i, j)=0. 00
    end if
  13 continue

  return
end
c
c-----
subroutine invl(il, l, n)
c 下三角の逆行列を求めるサブルーチン
dimension il(10, 10), l(10, 10)
real il, l
integer i, j, k, n

do 30 i=1, n
  do 30 j=1, n
    if(i. eq. j) il(i, j)=1. 0/l(i, j)
    if(i. lt. j) il(i, j)=0. 0
    if(i. gt. j) then
      il(i, j)=0. 0
      do 31 k=1, i-1
        il(i, j)=il(i, j)+l(i, k)*il(k, j)

```



```

31     continue
      il(i, j)=-il(i, j)/l(i, i)
    end if
30 continue

      return
    end

c
c-----
c      subroutine invu(iu, u, n)
c      上三角の逆行列を求めるサブルーチン
c      dimension iu(10, 10), u(10, 10)
c      real      iu, u
c      integer  i, j, k, n

      do 40 i=1, n
        do 40 j=1, n
          if(i. eq. j) iu(i, j)=1. 0
          if(i. gt. j) iu(i, j)=0. 0
          if(i. lt. j) then
            iu(i, j)=0. 0
            do 41 k=i, j-1
              iu(i, j)=iu(i, j)-iu(i, k)*u(k, j)
41          continue
            end if
40 continue

      return
    end

c
c-----
c      subroutine matrix(n, aa, bb, ab)
c      行列の乗算を行うサブルーチン
c      real aa(10, 10), bb(10, 10), ab(10, 10)
c
      do 15 i=1, n
        do 15 j=1, n
          sum=0. 0
          do 16 k=1, n
            sum=aa(i, k)*bb(k, j)+sum
16        continue
          ab(i, j)=sum
15        continue
      return
    end

```

02-5program. for. txt

```

C *****
C 第2回ゼミ
C 非線形方程式の数値解析法
C ベイリー法による連立方程式の解法
C
C 作成者 東海林 勉      作成日 2004. 5. 29
C *****
C 二次微係数利用
C
C F=16*x**2+5*x*y+50*y**2-74
C G=4*x**2+25*x*y**2+10*y-41
C-----
C      real dGx, dGy, b(10, 10), a2(10, 10),
C      & a2inv(10, 10), ddx2(10, 10), l(10, 10), u(10, 10),
C      & il(10, 10), iu(10, 10)
C
C      write(6, *) ' <Bailey法>'
C
C      x=1. 0
C      y=1. 0
C      n=2. 0
C      eps=0. 0001
C
C      do 100 i=1, 2
C        write(6, '(a5, f5. 2, a5, f5. 2)') ' x=', x, ' y=', y
C        write(6, *) ' x          y          dx          dy          F          G'
C
C      do 200 k=1, 20
C        F=16. *x**2. +5. *x*y+50. *y**2. -74.
C        G=4. *x**2. +25. *x*y**2. +10. *y-41.
C        dFx=32. *x+5. *y
C        dFy=5. *x+100. *y
C        dGx=8. *x+25. *y**2.
C        dGy=50. *x*y+10.
C        ddFx=32.
C        ddFy=100.
C        ddFxy=5.
C        ddGx=8.
C        ddGy=50. *x
C        ddGxy=50. *y
C        b(1, 1)=-F
C        b(2, 1)=-G
C
C        a2(1, 1)=dFx-F*ddFx/(2. *dFx)-F*ddFxy/(2. *dFy)
C        a2(1, 2)=dFy-F*ddFy/(2. *dFy)-F*ddFxy/(2. *dFx)
C        a2(2, 1)=dGx-G*ddGx/(2. *dGx)-G*ddGxy/(2. *dGy)
C        a2(2, 2)=dGy-G*ddGy/(2. *dGy)-G*ddGxy/(2. *dGx)
C
C      call crout(l, u, a2, n)
C      call invu(iu, u, n)
C      call invl(il, l, n)
C      call matrix(n, iu, il, a2inv)
C      call matrix(n, a2inv, b, ddx2)
C      dx=ddx2(1, 1)
C      dy=ddx2(2, 1)
C      write(6, '(10f8. 5)') x, y, dx, dy, F, G
C      if(abs(F). le. eps. and. abs(G). le. eps) go to 121
C      x=x+dx

```

```

        y=y+dy
200  continue
121  write(6,*) , k, '回'
        write(6,*) , k, '回'
        x=2.0
        y=1.0
100  continue

        stop
        end

c
c-----
c      subroutine crout(l, u, a, n)
c      クラウト法を用いLU分解するサブルーチン
c      dimension a(10, 10), l(10, 10), u(10, 10), d(10, 10)
c      real    a, l, u, d
c      integer i, j, k, n

c      do 10 i=1, n
c        do 10 j=1, n
c          d(i, j)=a(i, j)
c          if(i. le. j) then
c            do 11 k=1, i-1
c              d(i, j)=d(i, j)-d(i, k)*d(k, j)
11          continue
c          if(i. lt. j) d(i, j)=d(i, j)/d(i, i)
c        else
c          do 12 k=1, j-1
c            d(i, j)=d(i, j)-d(i, k)*d(k, j)
12          continue
c        end if
c      10 continue
c      L行列とU行列に要素を分ける
c      do 13 i=1, n
c        do 13 j=1, n
c          if(i. ge. j) then
c            l(i, j)=d(i, j)
c            if(i. eq. j) then
c              u(i, j)=1.00
c            else
c              u(i, j)=0.00
c            end if
c          else
c            u(i, j)=d(i, j)
c            l(i, j)=0.00
c          end if
c        13 continue

c      return
c      end

c
c-----
c      subroutine invl(il, l, n)
c      下三角の逆行列を求めるサブルーチン
c      dimension il(10, 10), l(10, 10)
c      real    il, l
c      integer i, j, k, n

```

```

do 30 i=1, n
  do 30 j=1, n
    if(i. eq. j) il(i, j)=1. 0/l(i, j)
    if(i. lt. j) il(i, j)=0. 0
    if(i. gt. j) then
      il(i, j)=0. 0
      do 31 k=1, i-1
        il(i, j)=il(i, j)+l(i, k)*il(k, j)
31      continue
        il(i, j)=-il(i, j)/l(i, i)
      end if
30 continue

  return
end

c
c-----
c      subroutine invu(iu, u, n)
c      上三角の逆行列を求めるサブルーチン
c      dimension iu(10, 10), u(10, 10)
c      real iu, u
c      integer i, j, k, n

c      do 40 i=1, n
c      do 40 j=1, n
c      if(i. eq. j) iu(i, j)=1. 0
c      if(i. gt. j) iu(i, j)=0. 0
c      if(i. lt. j) then
c      iu(i, j)=0. 0
c      do 41 k=i, j-1
c      iu(i, j)=iu(i, j)-iu(i, k)*u(k, j)
41      continue
c      end if
40 continue

c      return
c      end

c
c-----
c      subroutine matrix(n, aa, bb, ab)
c      行列の乗算を行うサブルーチン
c      real aa(10, 10), bb(10, 10), ab(10, 10)
c
c      do 15 i=1, n
c      do 15 j=1, n
c      sum=0. 0
c      do 16 k=1, n
c      sum=aa(i, k)*bb(k, j)+sum
16      continue
c      ab(i, j)=sum
15      continue
c      return
c      end

```

02-6program. for. txt

```

c *****
c 第2回ゼミ
c 非線形方程式の数値解析法
c ニュートン法とベイリー法を併用した連立方程式の解法
c
c 作成者 宮武 真由子      作成日 2004. 5. 29
c *****
c 一次微係数利用
c 二次微係数利用
c
c F=16*x**2+5*x*y+50*y**2-74
c G=4*x**2+25*x*y**2+10*y-41
c -----
      dimension a(10, 10), b(10, 10), ainv(10, 10), ddx(10, 10)
      dimension l(10, 10), il(10, 10), u(10, 10), iu(10, 10)
      real F, G, DFX, DFY, DGX, DGY, DDFX, DDFY, DDGX, DDGY, DDFXY, DDGXY, x, y, dx, dy
      integer n

      x=1. 0
      y=1. 0
      n=2. 0

      do 999 i=1, 2
      open(2, file='out. dat', status='unknown')
      write (2, *) ' K X          Y          DX          DY          F          G'
      do K=1, 100

c ----- NEWTON METHOD -----
      F=16. 0*x**2. 0+5. 0*x*y+50. 0*y**2. 0-74. 0
      G=4. 0*x**2. 0+25. 0*x*y**2. 0+10. 0*y-41. 0
      DFX=32. 0*x+5. 0*y
      DFY=5. 0*x+100. 0*y
      DGX=8. 0*x+25. 0*y**2. 0
      DGY=50. 0*x*y+10. 0
      DDFX=32. 0
      DDFY=100. 0
      DDGX=8. 0
      DDGY=50. 0*x
      DDFXY=5. 0
      DDGXY=50. 0*y

      a(1, 1)=DFX
      a(1, 2)=DFY
      a(2, 1)=DGX
      a(2, 2)=DGY
      b(1, 1)=-F
      b(2, 1)=-G

      call crout(l, u, a, n)
      call invu(iu, u, n)
      call invl(il, l, n)
      call matrix(n, iu, il, ainv)
      call matrix(n, ainv, b, ddx)

      dx=ddx(1, 1)
      dy=ddx(2, 1)

c ----- BAILEY METHOD -----

```

02-6program. for. txt

```

a(1, 1)=DFX-1/2. 0*DDFX*dx-1/2. 0*DDFX*dy
a(1, 2)=DFY-1/2. 0*DDFY*dy-1/2. 0*DDFY*dx
a(2, 1)=DGX-1/2. 0*DDGX*dx-1/2. 0*DDGX*dy
a(2, 2)=DGY-1/2. 0*DDGY*dy-1/2. 0*DDGY*dx
b(1, 1)=-F
b(2, 1)=-G

```

```

call crout(l, u, a, n)
call invu(iu, u, n)
call invl(il, l, n)
call matrix(n, iu, il, ainv)
call matrix(n, ainv, b, ddx)

```

```

dx=ddx(1, 1)
dy=ddx(2, 1)

```

```

c ----- OUTPUT -----
  write (2, '(7f10. 5)') x, y, DX, DY, F, G
c
  IF (ABS(F). LE. 1E-4. and. ABS(G). LE. 1E-4) GO TO 100

  x=x+dx
  y=y+dy

  end do
100 continue
c
  x=2. 0
  y=1. 0
999 continue

  stop
  end

```

```

c -----
subroutine crout(l, u, a, n)
dimension a(10, 10), l(10, 10), u(10, 10), d(10, 10)
real a, l, u, d
integer i, j, k, n

do 10 i=1, n
  do 10 j=1, n
    d(i, j)=a(i, j)
    if(i. le. j) then
      do 11 k=1, i-1
        d(i, j)=d(i, j)-d(i, k)*d(k, j)
11      continue
      if(i. lt. j) d(i, j)=d(i, j)/d(i, i)
    else
      do 12 k=1, j-1
        d(i, j)=d(i, j)-d(i, k)*d(k, j)
12      continue
    end if
10 continue

do 13 i=1, n
  do 13 j=1, n
    if(i. ge. j) then

```

```

        l(i, j)=d(i, j)
        if(i. eq. j) then
            u(i, j)=1. 00
        else
            u(i, j)=0. 00
        end if
    else
        u(i, j)=d(i, j)
        l(i, j)=0. 00
    end if
13 continue

    return
end

```

---

```

c
subroutine invl(il, l, n)
dimension il(10, 10), l(10, 10)
real    il, l
integer i, j, k, n

do 30 i=1, n
do 30 j=1, n
    if(i. eq. j) il(i, j)=1. 0/l(i, j)
    if(i. lt. j) il(i, j)=0. 0
    if(i. gt. j) then
        il(i, j)=0. 0
        do 31 k=1, i-1
            il(i, j)=il(i, j)+l(i, k)*il(k, j)
31        continue
        il(i, j)=-il(i, j)/l(i, i)
    end if
30 continue

    return
end

```

---

```

c
subroutine invu(iu, u, n)
dimension iu(10, 10), u(10, 10)
real    iu, u
integer i, j, k, n

do 40 i=1, n
do 40 j=1, n
    if(i. eq. j) iu(i, j)=1. 0
    if(i. gt. j) iu(i, j)=0. 0
    if(i. lt. j) then
        iu(i, j)=0. 0
        do 41 k=i, j-1
            iu(i, j)=iu(i, j)-iu(i, k)*u(k, j)
41        continue
    end if
40 continue

    return
end

```

```
c-----  
      subroutine matrix(n, aa, bb, ab)  
      real aa(10, 10), bb(10, 10), ab(10, 10)  
      do i=1, n  
        do j=1, n  
          sum=0.0  
          do k=1, n  
            sum=aa(i, k)*bb(k, j)+sum  
          end do  
          ab(i, j)=sum  
        end do  
      end do  
      return  
      end
```



02-7program.for.txt

```

C *****
C 第2回ゼミ
C 非線形方程式の数値解析法
C 不等流計算 1階ニュートン法
C
C 作成者 東海林 勉          作成日 2004. 5. 29
C *****
C 一次微係数利用
  real l1
  open(6, file='問題4(1).out', status='unknown')
  write(6, *) '問題4 解答 1次微係数'
C ②地点
  h1=4.5
  h2=4.5
  aI1=2000.0
  l1=1200.0
  b1=300.0
  b2=280.0
  write(6, *) '▽②地点'
  call ftoryu (h1, h2, aI1, l1, B1, B2)
C ③地点
  h1=h2
  aI1=1500.0
  l1=950.0
  b1=280.0
  b2=250.0
  write(6, *) '▽③地点'
  call ftoryu (h1, h2, aI1, l1, B1, B2)
C ④地点
  h1=h2
  aI1=700.0
  l1=1350.0
  b1=250.0
  b2=220.0
  write(6, *) '▽④地点'
  call ftoryu (h1, h2, aI1, l1, B1, B2)
stop
end
C
C*****
  subroutine ftoryu (h1, h2, aI1, l1, B1, B2)
  real l1
  a=1.0
  q=300.0
  an=0.025
  g=9.8
  eps=0.00001
  AA=(A*Q**2)/(2*g*B2**2)
  BB=(an**2*Q**2*l1)/(2*B2**2)
  write(6, *) '      h2          f(h2)      f'(h2)      Δh2 '
  F1=h1-l1/aI1+(a*Q**2.0)/(2.0*g*B1**2*h1**2.0)
  & +(an**2.0*Q**2.0*l1)/(2.0*B1**2.0*h1**(10.0/3.0))
  do 1 i=1, 30
    F2=h2+(AA/(h2**2.0))-(BB/(h2**(10.0/3.0)))
    dF2=1.0-2.0*AA/(h2**3.0)+(10.0*BB)/(3.0*h2**(13.0/3.0))
    ff=F2-F1
    dh2=-(ff/dF2)
    write(6, '(7f10.5)') h2, ff, dF2, dh2

```

```
02-7program.for.txt
  if (ABS(ff).le.eps ) go to 100
  h2=h2+dh2
1 continue
100 return
end
```

## 02-8program.for.txt

```

c *****
c 第2回ゼミ
c 非線形方程式の数値解析法
c 不等流計算 2階ニュートン法
c
c 作成者 東海林 勉          作成日 2004. 5. 29
c *****
c 二次微係数利用
  real l1
  open(6, file='問題4(2).out', status='unknown')
  write(6,*) '問題4 解答 <2次微係数>'
c ②地点
  h1=4.5
  h2=4.5
  aI1=2000.0
  l1=1200.0
  b1=300.0
  b2=280.0
  write(6,*) '▽②地点'
  call ftoryu 2dimensions (h1, h2, aI1, l1, B1, B2)
c ③地点
  h1=h2
  aI1=1500.0
  l1=950.0
  b1=280.0
  b2=250.0
  write(6,*) '▽③地点'
  call ftoryu 2dimensions (h1, h2, aI1, l1, B1, B2)
c ④地点
  h1=h2
  aI1=700.0
  l1=1350.0
  b1=250.0
  b2=220.0
  write(6,*) '▽④地点'
  call ftoryu 2dimensions (h1, h2, aI1, l1, B1, B2)
  stop
  end
c
c*****
  subroutine ftoryu 2dimensions (h1, h2, aI1, l1, B1, B2)
  real l1
  a=1.0
  q=300.0
  an=0.025
  g=9.8
  eps=0.0001
  AA=(A*Q**2.0)/(2.0*g*B2**2.0)
  BB=(an**2.0*Q**2.0*l1)/(2.0*B2**2.0)
  F1=h1-l1/aI1+(a*Q**2.0)/(2.0*g*B1**2.0*h1**2.0)
  & +(an**2.0*Q**2.0*l1)/(2.0*B1**2.0*h1**2.0*(10.0/3.0))
  write(6,*) ' h1      h2      f      df      ddf      dh'
  do 1 i=1,5
    F2=h2+(AA/(h2**2.0))-(BB/(h2**2.0*(10.0/3.0)))
    dF2=1.0-2.0*AA/(h2**3.0)+(10.0*BB)/(3.0*h2**2.0*(13.0/3.0))
    ddF2=6.0*AA/(h2**4.0)-130.0*BB/(9.0*h2**3.0*(16.0/3.0))
    F2=F2-F1
    write(6, ' (6f8.4)') h1, h2, F2, dF2, ddF2, dh2

```

```

                                02-8program. for. txt
if (ABS(F2). le. eps ) go to 100
dh2=(-dF2+(dF2**2. 0-2. 0*F2*ddF2)**0. 5)/ddF2
dh3=(-dF2-(dF2**2. 0-2. 0*F2*ddF2)**0. 5)/ddF2
h3=h2+dh2
h4=h2+dh3
F3=h3+(AA/(h3**2. 0))-(BB/(h3**(10. 0/3. 0)))
F4=h4+(AA/(h4**2. 0))-(BB/(h4**(10. 0/3. 0)))
if(F3. lt. F4) then
h2=h3
else
h2=h4
end if
1 continue
100 return
end

```

06program.for.txt

```

C *****
C 第6回ゼミ
C 一価線形・非線形貯留関数法の最適化
C 一般化貯留関数モデルパラメータ同定
C (1価の非線形貯留方程式のパラメータ最適化)
C
C 作成者 東海林 勉          作成日 2004. 8. 6
C *****
C
C k11 の最適化
C 一階ニュートン(newton)法
C ヤコビアン(jacobian)行列 (一次微係数)
C*****
C
C 有効雨量と直接流出高の入力
C storage-discharge relationship
C s = k11*q**p1
C optimization of parameters k11
C p1 = 0.6
C -----
C character suikei*80, kasenmei*80, chiten*80, flood*80
C
C dimension er (200), uu (200), eu (200), pas (200)
C
C common/st1/wk, wp1, rain
C common/st2/h
C common/date/area, nq, nr, xnq
C common/discha/r (200), qo (200), qc (200)
C common/keisu1/a1, f, g
C
C open(1, file='input.dat', status='unknown')
C open(6, file='1価非線形1次微係数による最適化.out', status='unknown')
C read(1, '(a80)') suikei          !水系名
C read(1, '(a80)') kasenmei       !河川名
C read(1, '(a80)') chiten         !地点名
C read(1, '(a80)') flood          !洪水年月日
C read(1, '(f8.4)') area          !流域面積
C read(1, '(f8.4)') rave          !平均有効雨量強度
C read(1, '(2i8)') nr, nq        !データ数(有効雨量・直接流出高)
C read(1, '(8f10.6)') (r(i), i=1, nr) !有効雨量
C read(1, '(8f10.6)') (qo(i), i=1, nq) !直接流出高
C
C data nl, kount, eps/6, 20, 0.001/ !分割時間, 繰返計算回数, 収束条件
C data wk10, wp10/22.7602, 0.6/    !パラメータK1初期値
C--OUTPUT 1
C write(6, *) '1価の非線形貯留関数法 (一階ニュートンによる最適化) '
C write(6, '( )')
C write(6, '(洪水:', a80)') flood
C write(6, '(水系名:', a80)') suikei
C write(6, '(河川名:', a80)') kasenmei
C write(6, '(地点名:', a80)') chiten
C write(6, '( )')
C write(6, '(データ数=', i3, ' 時間間隔=1/', i2)') nr, nl
C write(6, '( )')
C write(6, 210)
C 210 format(4x, '流域面積', 3x, '平均有効雨量強度', 4x, 'k1初期値')
C write(6, '(f10.1, 2f16.4)') area, rave, wk10
C write(6, '( )')

```

```

write(6, 209)
209 format(3x, '回数', 6x, 'k', 11x, 'Δk', 6x, '目的関数', 6x, 'RMSE', 7x,
& 'KAI2')
c-----
      xnl=n1
      h=1.0/xnl
      wkz=1.0
      wp1=1.0/wp10
      xnq=nq
      do 10 j=1, kount
      wk=wk10*wkz
      u=0.0
      x=0.0
      eut=0.0
      uut=0.0
      sum=0.0
      skai2=0.0
      do 15 ll=1, nq
      rain=0.0
      if(ll.le.nr) rain=r(ll)
      qq=qo(ll)
      qs=0.0
      if(qq.gt.0.0) qs=sqrt(qq)
      do 20 k=1, n1
c** solution of sensitivity equation
c
      call nolin1(x, u, 1)
c
c** solution of system equation
c
      call nolin1(x, u, 2)
c
      if(x.le.0.0) x=0.0

20 continue
      pas(ll)=u*x**(wp1-1.)*wp1      !変数変換
      pas(ll)=wk10*pas(ll)         !規準化感度係数
      qcc=x**wp1
      qc(ll)=qcc
      err=qq-qcc
      er(ll)=err
      eu(ll)=er(ll)*pas(ll)
      uu(ll)=pas(ll)**2
      eut=eut+eu(ll)
      uut=uut+uu(ll)
      sum=sum+er(ll)**2
      if(qq.eq.0.0) go to 15
      erw=er(ll)/qs
      skai2=skai2+erw**2
15 continue
      xj=sum/xnq                    !目的関数
      sum=sqrt(sum/xnq)              !RMSE
      skai2=skai2/xnq                !KAI2
      delwk=eut/uut
      ck=delwk/wkz
      write(6, '(i5, 5f12.4)') j, wk, wk10*delwk, xj, sum, skai2
      if(abs(ck).lt.eps) go to 111
      wkz=wkz+delwk

```

10 continue

```

c-----
111 rtot=0.0
    qtot=0.0
    qctot=0.0
    do 1000 i=1,nr
        rtot=rtot+r(i)
1000 continue
    do 1001 i=1,nq
c** total observed runoff depth
        qtot=qtot+qo(i)
c** total computed runoff depth
        qctot=qctot+qc(i)
1001 continue

        call accuracy(qtot, qctot, qmaxo, qmaxc, qmax, qqmax, hiryu, peakg,
            & qsotai, qsotai2, qsotai3, qsotai4, qsotai5)

c--OUTPUT 2
    write(6, '()')
    write(6, 211)
211 format(2x, '収束回数', 2x, 'k1最適値', 5x, '目的関数', 4x, 'RMSE',
    & 8x, 'KAI2', 8x, 'Jpe', 9x, 'Jre')
        write(6, '(i7, 6f12.4)') j, wk, xj, sum, skai2, peakg, qsotai
        write(6, '()')
        write(6, 214)
214 format(5x, 'Ew', 9x, 'E', 9x, 'Ev', 8x, 'Ep')
        write(6, '(4f10.4)') qsotai2, qsotai3, qsotai4, qsotai5
        write(6, '()')
        write(6, 212)
212 format(4x, '比流量', 3x, '総雨量', 4x, '観測総流出', 3x, '計算総流出', 3x,
    & '観測ピーク値', 3x, '計算ピーク値')
        write(6, '(f10.4, 2f9.2, 3f14.2)') hiryu, rtot, qtot, qctot, qmax, qqmax
        write(6, '()')
        write(6, 213)
213 format(x, 'No', 4x, '有効雨量', 3x, '直接流出高', 4x, '計算流出高')
        do i=1, nq
            write(6, '(i3, 4f12.4)') i, r(i), qo(i), qc(i)
        end do
    stop
end

c-----
subroutine nolin1(x, u, ijk)
c** ijk = 1; solve sensitivity equation
c** ijk = 2; solve system equation
    common/st1/wk, wp1, rain
    common/st2/h
    common/keisu1/a1, f, g

c
    y1=x
c    if(y1.gt.0.0) go to 12
c    if(y1.lt.0.0) y1=0.0
c    go to 14
    a1=y1**(wp1-1.0)
    c=y1**wp1
    a=-wp1*a1/wk
    b=(c-rain)/(wk**2)

```

```

d=(wp1-1.0)*c/wk+rain/wk
c** elements of phi
f=exp(a*h)
c** elements of gamma
g=h
if(a.ne.0.0) g=(exp(a*h)-1.0)/a
if(ijk.eq.2) go to 104
c** solve sensitivity equation
u1=f*u+g*b
u=u1
return
c** solve system equation
104 y=f*x+g*d
x=y
return
end

c-----
subroutine accuracy(qtot, qctot, qmaxo, qmaxc, qmax, qqmax, hiryu, peakg,
& qsotai, qsotai2, qsotai3, qsotai4, qsotai5)

common/discha/r (200), qo (200), qc (200)
common/date/area, nq, nr, xnq

qsotai=0.0
qsotai2=0.0
qsotai3=0.0
qmaxo=-999.99
qmaxc=-999.99
do 502 i=1, nq
if(qo(i).gt.qmaxo) qmaxo=qo(i) !観測ピーク
if(qc(i).gt.qmaxc) qmaxc=qc(i) !計算ピーク
502 continue
qmax=qmaxo*area/3.6
qqmax=qmaxc*area/3.6
iko=0.0
do 501 i=1, nq
if(qo(i).lt.0.04) go to 500
iko=iko+1
qsotai=qsotai+abs(qo(i)-qc(i))/qo(i)
qsotai2=qsotai2+((qo(i)-qc(i))/qo(i))**2
500 continue
qsotai3=qsotai3+((qo(i)-qc(i))/qmaxo)**2
501 continue
xiko=iko
hiryu=qmax/area !比流量
peakg=abs(qmaxo-qmaxc)/qmaxo !JPE
qsotai=qsotai/xiko !JRE
qsotai2=qsotai2/xiko !Ew
qsotai3=qsotai3/xnq !E
qsotai4=(qtot-qctot)/qtot !Ev
qsotai5=(qmaxo-qmaxc)/qmaxo !Ep
return
end

```



湧別川  
湧別川  
丸瀬布

平成13年9月11日

802.0  
1. 6993  
94 94  
1. 357979 1. 953697 1. 979319 5. 188506 4. 810578 4. 086749 3. 971449  
2. 882503  
3. 260431 3. 696010 3. 318081 3. 446193 3. 779282 3. 113103 2. 197108  
2. 562225  
3. 087481 4. 413433 4. 195644 4. 093155 0. 871157 0. 704612 0. 813506  
1. 383602  
1. 556552 1. 921669 0. 871157 1. 729502 1. 249085 0. 640556 0. 743045  
0. 416362  
0. 230600 0. 704612 0. 416362 0. 762262 0. 992862 0. 608528 0. 743045  
0. 871157  
0. 435578 0. 640556 0. 454795 0. 416362 0. 102489 0. 000000 0. 454795  
0. 000000  
0. 128111 0. 000000 0. 064056 0. 044839 0. 000000 0. 000000 0. 000000  
0. 000000  
0. 000000 0. 000000 0. 000000 0. 000000 0. 000000 0. 000000 0. 000000  
0. 000000  
0. 000000 0. 000000 0. 000000 0. 000000 0. 000000 0. 000000 0. 000000  
0. 000000  
0. 000000 0. 000000 0. 000000 0. 000000 0. 000000 0. 000000 0. 000000  
0. 000000  
0. 000000 0. 000000 0. 000000 0. 000000 0. 000000  
0. 003411 0. 016374 0. 023523 0. 129497 0. 230115 0. 385174 0. 590781  
0. 829241  
0. 960343 1. 071099 1. 141725 1. 191243 1. 265854 1. 401141 1. 552027  
1. 738390  
1. 857268 1. 920897 2. 025711 2. 154036 2. 350736 2. 589758 2. 642277  
2. 745701  
2. 658581 2. 572659 2. 476555 2. 315424 2. 180500 2. 080521 1. 941551  
1. 886504  
1. 792466 1. 671622 1. 610329 1. 549916 1. 472195 1. 404925 1. 330138  
1. 291139  
1. 269595 1. 231183 1. 209932 1. 155533 1. 077776 1. 049374 0. 974257  
0. 985421  
0. 957754 0. 922666 0. 858049 0. 883587 0. 769014 0. 729123 0. 717712  
0. 657983  
0. 613118 0. 628866 0. 577852 0. 586670 0. 595585 0. 551600 0. 560320  
0. 542833  
0. 518979 0. 508154 0. 459365 0. 448736 0. 419614 0. 378855 0. 368495  
0. 376090  
0. 336065 0. 302652 0. 321444 0. 299643 0. 266719 0. 296658 0. 246708  
0. 236739  
0. 237942 0. 194971 0. 169075 0. 170033 0. 144431 0. 134804 0. 109691  
0. 065021  
0. 080586 0. 056255 0. 037168 0. 018300 0. 004383

input.txt

07-1program.for.txt

```

C *****
C 第7回ゼミ
C 二価の線形・非線形貯留関数法の最適化
C 一般化貯留関数モデルパラメータ同定
C (2価の非線形貯留方程式のパラメータ最適化)
C
C 作成者 星 清          作成日 2004.11.8
C *****
C
C friction factor fc の1次元探索
C 一階ニュートン(newton)法
C ヤコビアン(jacobian)行列 (一次微係数)
C*****
C
C 有効雨量と直接流出高の入力
C k11=2.8235*fc*area**0.24
C optimize the friction factor fc
C
C k12=0.2835*k11**2*rave**(-0.2648)
C
C storage-discharge relationship
C s = k11*q**p1 + k12*d(q**p2)/dt
C optimization of parameters k11 and k12
C p1 = 0.6 and p2 = 0.4648
C
C -----
C character suikei*128, kasenmei*128, chiten*128, flood*128
C
C dimension pas(200,9), dpa(9)
C dimension x(9), u(18), z(9), co(9)
C
C common/st1/wk1, wk2, wp1, wp2, rain
C common/st2/n, m, m1, m2, h, h2, h3, h4
C common/st3/nq, nr, xnq, area
C common/st4/alph3, alph4, alph5
C common/discha/r(200), qo(200), qc(200), q00(200)
C
C open(1, file='input.dat', status='unknown')
C open(6, file='2価非線形1次微係数によるfc最適化.dat', status='
& unknown')
C
C fc=friction factor          fc初期値
C** kount=maximum number of iterations      繰返回数
C** eps=convergence limit          収束条件
C data fc, wp1, zp2 /1.5, 0.6, 0.4648/
C data n, m, nr, nq, nl, kount, eps/2, 1, 94, 94, 6, 20, 0.001/
C
C read(1,3) suikei            !水系名
C read(1,3) kasenmei         !河川名
C read(1,3) chiten           !地点名
C read(1,3) flood            !洪水年月日
3 format(a128)
C read(1,'(f8.4)') area       !流域面積
C read(1,2) (r(i), i=1, nr)   !有効雨量
C read(1,2) (qo(i), i=1, nq) !直接流出高
2 format(8f10.0)
C
C rave = 平均雨量強度
C nzero=0

```

## 07-1program. for. txt

```

rtot=0.0
do 199 i=1, nr
  if(r(i).eq.0.0) go to 199
  nzero=nzero+1
  rtot=rtot+r(i)
199 continue
  xnz=nzero
  rave=rtot/xnz

c
c  initial values of parameters
c
  alph1=2.8235*area**0.24
  alph4=alph1
  alph2=0.2835*rave**(-0.2648)
  alph3=alph1**2*alph2

c
c  normalized parameters
  yk1=fc
  co(1)=fc

c---OUTPUT 1
  write(6,*)' 2価の非線形貯留関数法 (一階ニュートン法による (fc) 最適化
&)'
  write(6,' ( )')
  write(6,' ( '洪水:',a20)') flood
  write(6,' ( '水系名:',a10)') suikei
  write(6,' ( '河川名:',a10)') kasenmei
  write(6,' ( '地点名:',a20)') chiten
  write(6,' ( )')
  write(6,' ( 'データ数=',i3,' '時間間隔=1/',i2)') nq, n1
  write(6,' ( )')
  write(6,210)
210 format(x,' 流域面積',3x,' 雨量強度',2x,' fc初期値')
  write(6,' (f10.1,3f10.4)') area, rave, yk1
  write(6,' ( )')
  write(6,777)
777 format(4x,'回数',4x,' fc',7x,' MSE',7x,' RMSE',6x,' KAI2')

c-----
  xnl=n1
  h=1./xnl
  h2=h**2
  h3=h2*h
  h4=h3*h
  m1=m+1
  m2=2*m
  xnq=nq
  wp2=1./zp2

c
c**  start optimization (newton method)
c
  fac=0.0
  do 999 kll=1, kount

c
c  updated parameters
  wk1=alph1*yk1
  zk2=alph3*yk1**2
  wk2=1./zk2
  alph5=2.*alph3*yk1

```

## 07-1program. for. txt

```

c
c   initial values
do 800 i=1, n
800  x(i)=0.0
c   initial values of sensitivity coefficients
do 802 i=1, m2
802  u(i)=0.0
c
c   skai2=0.0
sum=0.0
su2=0.0
seu=0.0
c
c   do 30 ll=1, nq
rain=0.0
if(ll. le. nr) rain=r(ll)
qq=qo(ll)
qs=0.0
if(qq. gt. 0.0) qs=sqrt(qq)
do 40 k=1, n1
c**  solution of sensitivity equation
c
c   call nolinfc(x, u, 1)
c
c**  solution of system equation
c
c   call nolinfc(x, u, 2)
c
c   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.)
c**  computed direct runoff depth
qc(ll)=qcc
err=qq-qcc
if(qq. eq. 0.0) go to 49
erw=err/qs
skai2=skai2+erw**2
49  continue
sum=sum+err**2
c**  sensitivity coefficients
do 42 i=1, m
42  pas(ll, i)=cc*u(i)
c**  normalized sensitivity coefficients
do 43 i=1, m
43  pas(ll, i)=co(i)*pas(ll, i)
su2=su2+pas(ll, 1)**2
c
c**  error between observed and computed discharges
pas(ll, m1)=err
seu=seu+pas(ll, m1)*pas(ll, 1)
30  continue
c
c   xj=sum/xnq
skai2=skai2/xnq
sum=sqrt(sum/xnq)
!目的関数
!KAI2
!RMSE

```

## 07-1program.for.txt

```

c** solve the correction terms of parameters
dpa(1)=seu/su2
c
z(1)=yk1
c
write(6,779) kkl1,yk1,sum**2,sum,skai2
779 format(1h,i5,4f10.4)
do 875 i=1,m
col = abs(dpa(i)/z(i))
if(col.gt.eps) go to 877
875 continue
go to 879
877 fac = 0.5*(1.+fac)
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
yk1=z(1)
c
999 continue
c-----
879 continue
rtot=0.0
do 1003 i=1,nr
rtot=rtot+r(i)
1003 continue
qtot=0.0
qctot=0.0
do 1002 i=1,nq
c** total observed runoff depth
qtot=qtot+qo(i) !総実測流出高
c** total computed runoff depth
qctot=qctot+qc(i) !総計算流出高
1002 continue
c
call accuracy(qtot,qctot,qmaxo,qmaxc,qmax,qqmax,hiryu,peakg,
& qsotai,qsotai2,qsotai3,qsotai4,qsotai5) !精度評価指標
c
c--OUTPUT 2
write(6,'()')
write(6,211)
211 format(3x,'収束回数',3x,'k1',9x,'k2',10x,'MSE',9x,'RMSE',
& 8x,'KAI2',8x,'Jpe',9x,'Jre')
write(6,'(i7,7f12.4)') kkl1,wk1,zk2,xj,sum,skai2,peakg,qsotai
write(6,'()')
write(6,214)
214 format(5x,'Ew',9x,'E',9x,'Ev',8x,'Ep')
write(6,'(4f10.4)') qsotai2,qsotai3,qsotai4,qsotai5
write(6,'()')
write(6,212)
212 format(4x,'比流量',3x,'総雨量',4x,'観測総流出',3x,'計算総流出',3x,
& '観測ピーク値',3x,'計算ピーク値')
write(6,'(f10.4,2f9.2,3f14.2)') hiryu,rtot,qtot,qctot,qmax,qqmax
write(6,'()')
write(6,204)

```

## 07-1program. for. txt

```

204 format(6x, 'No', 10x, '有効雨量', 8x, '実測流出高', 6x, '計算流出高')
do 205 i=1, nq
write(6, 207) i, r(i), qo(i), qc(i)
205 continue
207 format(i8, f16. 4, 5f16. 4)
stop
end

c
c-----
c      subroutine nolinf(x, u, ijk)
c**    s = k11*q**p1 + k12*d(q**p2)/dt
c**    x(k+1)= phi*x(k) + gamma*b(k)
c**    ijk = 1; solve sensitivity equation
c**    ijk = 2; solve system equation
c
dimension x(1), u(1), y(9), uu(18), b(9)
common/st1/wk1, wk2, wp1, wp2, rain
common/st2/n, m, m1, m2, h, h2, h3, h4
common/st4/alph3, alph4, alph5

c
con=wk1*wk2*wp1*wp2
a=0. 0
c=0. 0
d=0. 0
e=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
14 a1=-con*(wp1*wp2-1.)*a*y2-wk2*wp2*c
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
c**  solve sensitivity equation
b(1)=-wk2*wp1*wp2*d*y2
b(2)=wk2**2*(wk1*wp1*wp2*d*y2 + e - rain)
b3=alph4*b(1)+alph5*b(2)

c
uu(1)= f1*u(1) + f2*u(2) + g2*b3
uu(2)= f3*u(1) + f4*u(2) + g4*b3
do 20 i=1, m2
20 u(i)=uu(i)

```

```

return
104 continue
c
c** solve system equation
b1=con*(wp1*wp2-1.)*d*y2+wk2*(wp2-1.)*e+wk2*rain
c
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 accuracy(qtot, qctot, qmaxo, qmaxc, qmax, qqmax, hiryu, peakg,
& qsotai, qsotai2, qsotai3, qsotai4, qsotai5)

common/st3/nq, nr, xnq, area
common/discha/r(200), qo(200), qc(200), q00(200)

qsotai=0.0
qsotai2=0.0
qsotai3=0.0
qmaxo=-999.99
qmaxc=-999.99
do 502 i=1, nq
if(qo(i).gt.qmaxo) qmaxo=qo(i)      !観測ピーク
if(qc(i).gt.qmaxc) qmaxc=qc(i)    !計算ピーク
502 continue
qmax=qmaxo*area/3.6
qqmax=qmaxc*area/3.6
iko=0.0
do 501 i=1, nq
if(qo(i).lt.0.04) go to 500
iko=iko+1
qsotai=qsotai+abs(qo(i)-qc(i))/qo(i)
qsotai2=qsotai2+((qo(i)-qc(i))/qo(i))**2
500 continue
qsotai3=qsotai3+((qo(i)-qc(i))/qmaxo)**2
501 continue
xiko=iko
hiryu=qmax/area      !比流量
peakg=abs(qmaxo-qmaxc)/qmaxo      !JPE
qsotai=qsotai/xiko      !JRE
qsotai2=qsotai2/xiko      !Ew
qsotai3=qsotai3/xnq      !E
qsotai4=(qtot-qctot)/qtot      !Ev
qsotai5=(qmaxo-qmaxc)/qmaxo      !Ep
return
end

```



湧別川  
湧別川  
丸瀬布

平成13年9月11日

802.0  
 1. 357979 1. 953697 1. 979319 5. 188506 4. 810578 4. 086749 3. 971449  
 2. 882503  
 3. 260431 3. 696010 3. 318081 3. 446193 3. 779282 3. 113103 2. 197108  
 2. 562225  
 3. 087481 4. 413433 4. 195644 4. 093155 0. 871157 0. 704612 0. 813506  
 1. 383602  
 1. 556552 1. 921669 0. 871157 1. 729502 1. 249085 0. 640556 0. 743045  
 0. 416362  
 0. 230600 0. 704612 0. 416362 0. 762262 0. 992862 0. 608528 0. 743045  
 0. 871157  
 0. 435578 0. 640556 0. 454795 0. 416362 0. 102489 0. 000000 0. 454795  
 0. 000000  
 0. 128111 0. 000000 0. 064056 0. 044839 0. 000000 0. 000000 0. 000000  
 0. 000000  
 0. 000000 0. 000000 0. 000000 0. 000000 0. 000000 0. 000000 0. 000000  
 0. 000000  
 0. 000000 0. 000000 0. 000000 0. 000000 0. 000000 0. 000000 0. 000000  
 0. 000000  
 0. 000000 0. 000000 0. 000000 0. 000000 0. 000000 0. 000000 0. 000000  
 0. 000000  
 0. 000000 0. 000000 0. 000000 0. 000000 0. 000000 0. 000000 0. 000000  
 0. 000000  
 0. 000000 0. 000000 0. 000000 0. 000000 0. 000000  
 0. 003411 0. 016374 0. 023523 0. 129497 0. 230115 0. 385174 0. 590781  
 0. 829241  
 0. 960343 1. 071099 1. 141725 1. 191243 1. 265854 1. 401141 1. 552027  
 1. 738390  
 1. 857268 1. 920897 2. 025711 2. 154036 2. 350736 2. 589758 2. 642277  
 2. 745701  
 2. 658581 2. 572659 2. 476555 2. 315424 2. 180500 2. 080521 1. 941551  
 1. 886504  
 1. 792466 1. 671622 1. 610329 1. 549916 1. 472195 1. 404925 1. 330138  
 1. 291139  
 1. 269595 1. 231183 1. 209932 1. 155533 1. 077776 1. 049374 0. 974257  
 0. 985421  
 0. 957754 0. 922666 0. 858049 0. 883587 0. 769014 0. 729123 0. 717712  
 0. 657983  
 0. 613118 0. 628866 0. 577852 0. 586670 0. 595585 0. 551600 0. 560320  
 0. 542833  
 0. 518979 0. 508154 0. 459365 0. 448736 0. 419614 0. 378855 0. 368495  
 0. 376090  
 0. 336065 0. 302652 0. 321444 0. 299643 0. 266719 0. 296658 0. 246708  
 0. 236739  
 0. 237942 0. 194971 0. 169075 0. 170033 0. 144431 0. 134804 0. 109691  
 0. 065021  
 0. 080586 0. 056255 0. 037168 0. 018300 0. 004383

07-2program. for. txt

```

C *****
C 第7回ゼミ
C   二価の線形・非線形貯留関数法の最適化
C   一般化貯留関数モデルパラメータ同定
C   (2価の非線形貯留方程式のパラメータ最適化)
C
C 作成者 星 清          作成日 2004. 11. 8
C *****
C
C   friction factor fc の1次元探索
C   ダビドン(davidon)法
C   ヤコビアン(jacobian)行列 (一次微係数)
C   ヘイシアン(hessian)行列 (二次微係数)
C*****
C
C   有効雨量と直接流出高の入力
C   k11=2. 8235*fc*area**0. 24
C   optimize the friction factor fc
C
C   k12=0. 2835*k11**2*rave**(-0. 2648)
C
C   storage-discharge relationship
C   s = k11*q**p1 + k12*d(q**p2)/dt
C   optimization of parameters k11 and k12
C   p1 = 0. 6 and p2 = 0. 4648
C
C -----
C   character suikei*128, kasenmei*128, chiten*128, flood*128
C
C   dimension djo(5), ho(5, 5), ddi(5), djo2(5, 5), dpa(5)
C   common/st1/r(100), qo(200), qc(200), er(200)
C   common/st2/n, m, m1, m2, nl, h, h2, h3, h4, area
C   common/st3/nr, nq, xnq, wk1, wk2, wp1, wp2, zk2, zp2, rain
C   common/st4/alph3, alph4, alph5
C   common/st5/xj, skai2, co(5), z(5)
C
C   open(1, file=' input. dat', status=' unknown')
C   open(6, file=' 2価非線形ダビドン法によるfc最適化. dat', status='
&   unknown')
C
C   fc=friction factor          fc初期値
C** kount=maximum number of iterations      繰返回数
C** eps=convergence limit          収束条件
C   data fc, wp1, zp2/1. 5, 0. 6, 0. 4648/
C   data n, m, nr, nq, nl, kount, eps/2, 1, 94, 94, 6, 20, 0. 001/
C
C   read(1, 3) suikei          !水系名
C   read(1, 3) kasenmei       !河川名
C   read(1, 3) chiten         !地点名
C   read(1, 3) flood          !洪水年月日
C 3 format(a128)
C   read(1, '(f8. 4)') area    !流域面積
C   read(1, 2) (r(i), i=1, nr) !有効雨量
C   read(1, 2) (qo(i), i=1, nq) !直接流出高
C 2 format(8f10. 0)
C
C   rave = 平均雨量強度
C   nzero=0
C   rtot=0. 0

```

## 07-2program. for. txt

```

do 199 i=1, nr
  if(r(i).eq.0.0) go to 199
  nzero=nzero+1
  rtot=rtot+r(i)
199 continue
  xnz=nzero
  rave=rtot/xnz
c
c   initial values of parameters
c
  alph1=2.8235*area**0.24
  alph4=alph1
  alph2=0.2835*rave**(-0.2648)
  alph3=alph1**2*alph2
c
c   normalized parameters
  yk1=fc
  co(1)=fc
c
c---OUTPUT 1
  write(6,*)'2価の非線形貯留関数法 (ダビドン法による(fc)最適化) '
  write(6,'()')
  write(6,'(','洪水:',a20)') flood
  write(6,'(','水系名:',a10)') suikei
  write(6,'(','河川名:',a10)') kasenmei
  write(6,'(','地点名:',a20)') chiten
  write(6,'()')
  write(6,'(','データ数=',i3,' 時間間隔=1/',i2)') nq, n1
  write(6,'()')
  write(6,210)
210 format(x,'流域面積',3x,'雨量強度',2x,'fc初期値')
  write(6,'(f10.1,3f10.4)') area, rave, yk1
  write(6,'()')
  write(6,777)
777 format(4x,'回数',4x,'fc',7x,'MSE',7x,'RMSE',6x,'KAI2')
c-----
c
  xnl=n1
  h=1./xnl
  h2=h**2
  h3=h2*h
  h4=h3*h
  m1=m+1
  m2=2*m
  xnq=nq
  wp2=1./zp2
c
c**  start optimization (davidon method)
c
  fac=0.0
  do 999 kll=1, kount
c
c   updated parameters
  wk1=alph1*yk1
  zk2=alph3*yk1**2
  wk2=1./zk2
  alph5=2.*alph3*yk1
c

```

```

                                07-2program. for. txt
call davidonfc(sum, djo, djo2, ho)
c
c sum: objective function
c djo: jacobian matrix of objective function
c djo2: hessian matrix of objective function
c ho: inverse mmatrix of djo2
c
z(1)=yk1
c** solve the correction terms of parameters
do 932 i=1, m
ddi(i)=0.0
do 934 j=1, m
934 ddi(i)=ddi(i)+ho(i, j)*djo(j)
dpa(i)=-ddi(i)
932 continue
c
write(6, 779) kkl1, yk1, xj, sum, skai2
779 format(1h, i5, 4f10.4)
do 875 i=1, m
col = abs(dpa(i)/z(i))
if(col. gt. eps) go to 877
875 continue
go to 879
877 fac = 0.5*(1. +fac)
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
yk1=z(1)
c
999 continue
879 continue
c
rtot=0.0
do 1003 i=1, nr
rtot=rtot+r(i)
1003 continue
c
qtot=0.0
qctot=0.0
do 1002 i=1, nq
c** total observed runoff depth
qtot=qtot+qo(i) !総実測流出高
c** total computed runoff depth
qctot=qctot+qc(i) !総計算流出高
1002 continue
c
call accuracy(qtot, qctot, qmaxo, qmaxc, qmax, qqmax, hiryu, peakg,
& qsotai, qsotai2, qsotai3, qsotai4, qsotai5) !精度評価指標
c
c--OUTPUT 2
write(6, '()')
write(6, 211)
211 format(3x, '収束回数', 3x, 'k1', 9x, 'k2', 10x, 'MSE', 9x, 'RMSE',
& 8x, 'KAI2', 8x, 'Jpe', 9x, 'Jre')

```

```

                                07-2program.for.txt
write(6, '(i7, 7f12. 4)') kkl1, wk1, zk2, xj, sum, skai2, peakg, qsotai
write(6, '()')
write(6, 214)
214 format(5x, 'Ew', 9x, 'E', 9x, 'Ev', 8x, 'Ep')
write(6, '(4f10. 4)') qsotai2, qsotai3, qsotai4, qsotai5
write(6, '()')
write(6, 212)
212 format(4x, '比流量', 3x, '総雨量', 4x, '観測総流出', 3x, '計算総流出', 3x,
&'観測ピーク値', 3x, '計算ピーク値')
write(6, '(f10. 4, 2f9. 2, 3f14. 2)') hiryu, rtot, qtot, qctot, qmax, qqmax
write(6, '()')
write(6, 204)
204 format(6x, 'No', 10x, '有効雨量', 8x, '実測流出高', 6x, '計算流出高')
do 205 i=1, nq
write(6, 207) i, r(i), qo(i), qc(i)
205 continue
207 format(i8, f16. 4, 5f16. 4)
stop
end

c
c-----
c      subroutine davidonfc(sum, djp, dj2, pin)
c      first and second derivatives of objective function in
c      terms of parameters
c      compute the hessian matrix of objective function (dj2)
c      and its inverse matrix (pin)
c      dimension djp(5), x(2), u(10), pax(5), uq2(4, 4), ho(4, 4)
c      dimension u1(4)
c      dimension dj2(5, 5), pp(5, 5), pin(5, 5)
c      common/st1/r(100), qo(200), qc(200), er(200)
c      common/st2/n, m, m1, m2, n1, h, h2, h3, h4, area
c      common/st3/nr, nq, xnq, wk1, wk2, wp1, wp2, zk2, zp2, rain
c      common/st4/alph3, alph4, alph5
c      common/st5/xj, skai2, co(5), z(5)

c
c      sum=0. 0
c      skai2=0. 0

c
c      do 900 i=1, m
c      djp(i)=0. 0
c      do 902 j=i, m
c      ho(i, j)=0. 0
902 continue
900 continue
c      initial values of states
c      do 800 i=1, n
800 x(i)=0. 0
c      initial values of sensitivity coefficients
c      do 802 i=1, m2
802 u(i)=0. 0
c      do 804 i=1, m2
804 u1(i)=0. 0
c      do 804 i=1, m2
804 continue

c
c      do 30 ll=1, nq
c      rain=0. 0
c      if(ll. le. nr) rain=r(ll)
c      qq=qo(ll)

```

```

qs=0.0
if(qq.gt.0.0) qs=sqrt(qq)
do 40 k=1,nl
c** solutions of sensitivity equation
c** (first and second derivatives)
c
call nolinf(x,u,u1,1)
c
c** solution of system equation
c
call nolinf(x,u,u1,2)
c
if(x(1).le.0.0) x(1)=0.0
40 continue
qcc=0.0
cc=0.0
cc1=0.0
if(x(1).le.0.0) go to 41
qcc=x(1)**wp2
cc=wp2*x(1)**(wp2-1.)
cc1=wp2*(wp2-1.)*x(1)**(wp2-2.)
41 continue
c** computed direct runoff depth
qc(11)=qcc
err=qq-qcc
er(11)=err
if(qq.eq.0.0) go to 49
erw=err/qs
skai2=skai2+erw**2
49 continue
sum=sum+err**2
c first sensitivity coefficients
do 910 i=1,m
910 pax(i)=cc*u(i)
c second sensitivity coefficients
do 911 i=1,m
do 911 j=i,m
uq2(i,j)=cc*u1(i)+cc1*u(i)*u(j)
911 continue
c hessian matrix
do 912 i=1,m
do 912 j=i,m
ho(i,j)=ho(i,j)+err*uq2(i,j)-pax(i)*pax(j)
912 continue
c jacobian matrix
do 920 i=1,m
920 djp(i)=djp(i)+err*pax(i)
30 continue
c
c objective function
c
xj=sum/xnq !目的関数
skai2=skai2/xnq !KAI2
sum=sqrt(sum/xnq) !RMSE
c
c jacobian matrix of objective function
c in terms of standardized parameters
do 930 i=1,m

```

```

930 djp(i)=-2.*co(i)*djp(i)/xnq
c hessian matrix of objective function
do 932 i=1,m
do 932 j=i,m
djp2(i,j)=-2.*co(i)*co(j)*ho(i,j)/xnq
932 continue
do 935 i=1,m
do 935 j=1,m
935 pp(i,j)=djp2(i,j)
c
c compute the inverse(pp) of hessian matrix(pp)
c
do 937 i=1,m
do 937 j=1,m
937 pin(i,j)=1./pp(i,j)
c
return
end
c
c-----
c subroutine nolinf(x,u,u1,ijk)
c
c** ijk = 1; solve sensitivity equations in terms of
c first derivatives(jacobian matrix) and
c second derivatives(hessian matrix)
c
c** ijk = 2; solve system equation
c
c** s = k1*q**p1 + k2*d(q**p2)/dt
c** x(k+1)= phi*x(k) + gamma*b(k)
c
dimension x(1),u(1),uu(9),y(9),b(9)
dimension u1(4),uu1(4),ua(4)
dimension ccc(2,2),dz(2,2),dw(2,2),db(2,2)
dimension ak(4),bk(4)
common/st2/n,m,m1,m2,nl,h,h2,h3,h4,area
common/st3/nr,nq,xnq,wk1,wk2,wp1,wp2,zk2,zp2,rain
common/st4/alph3,alph4,alph5
c
con=wk1*wk2*wp1*wp2
a=0.0
c=0.0
d=0.0
e=0.0
az=0.0
ez=0.0
c
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
az=y1**(wp1*wp2-3.)
ez=y1**(wp2-2.)

```

```

14 a1=-con*(wp1*wp2-1.)*a*y2-wk2*wp2*c
    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
c**  solve sensitivity equation(first derivatives)
c
    b(1)=-wk2*wp1*wp2*d*y2
    b(2)=wk2**2*(wk1*wp1*wp2*d*y2 + e - rain)
c
    b3=alph4*b(1)+alph5*b(2)
c
    uu(1)= f1*u(1) + f2*u(2) + g2*b3
    uu(2)= f3*u(1) + f4*u(2) + g4*b3
    do 20 i=1, m2
20  u(i)=uu(i)
c
c**  solve hessian equation(second derivatives)
    c2=wp1*wp2
c
c    ccc(i, j)=da1/dx (first derivatives in terms of x)
c
    ccc(1, 1)=-con*(c2-1.)*(c2-2.)*az*y2
&    -wk2*wp2*(wp2-1.)*ez
    ccc(1, 2)=-con*(c2-1.)*a
    ccc(2, 1)=ccc(1, 2)
    ccc(2, 2)=0.0
c
c    dz(i, j)=da1/dk (first derivatives in terms of k)
c
    dz(1, 1)=-wk2*c2*(c2-1.)*a*y2
    dz(1, 2)=wk2*con*(c2-1.)*a*y2+wk2**2*wp2*c
    dz(2, 1)=-wk2*c2*d
    dz(2, 2)=wk2*con*d
c
c    db(i, j)=db1/dx (=transpose of dz(i, j))
c
    do 306 i=1, m2
    do 306 j=1, m2
306  db(j, i)=dz(i, j)
c
c    dw(i, j)=db/dk (first derivatives in terms of k)
c
    dw(1, 1)=0.0
    dw(1, 2)=wk2**2*c2*d*y2
    dw(2, 1)=dw(1, 2)
    dw(2, 2)=2.*wk2**3*(-wk1*c2*d*y2-e+rain)
c

```



```

c*** elements of forcing function in the
c second sensitivity equation
c
c ua(1)=alph4
c ua(2)=alph5
c
c*** (dda1/ddfc)=(da1/dx)*u+(da1/dk)*ua
c
c do 202 i=1, m2
c s=0.0
c do 195 k=1, m2
195 s=s+ccc(i, k)*u(k)+dz(i, k)*ua(k)
c ak(i)=s
202 continue
c
c*** (ddb1/ddfc)=(db1/dx)*u+(db1/dk)*ua
c
c do 204 i=1, m2
c s=0.0
c do 206 k=1, m2
206 s=s+db(i, k)*u(k)+dw(i, k)*ua(k)
c bk(i)=s
204 continue
c
c*** hhz=ut*ak(i)+uat*bk(i)
c
c s=0.0
c do 209 k=1, m2
209 s=s+u(k)*ak(k)+ua(k)*bk(k)
c hhz=s+2.*alph3*b(2)
c
c uu1(i)= w(i);hessian matrix in terms of x1 and x2
c
c uu1(1)=f1*u1(1)+f2*u1(2)+g2*hhz
c uu1(2)=f3*u1(1)+f4*u1(2)+g4*hhz
c do 224 i=1, m2
224 u1(i)=uu1(i)
c
c return
c
c 104 continue
c** solve system equation
c b1=con*(wp1*wp2-1.)*d*y2+wk2*(wp2-1.)*e+wk2*rain
c
c y(1) = f1*x(1)+f2*x(2)+g2*b1
c y(2) = f3*x(1)+f4*x(2)+g4*b1
c do 102 i=1, n
102 x(i)=y(i)
c return
c end
c
c-----
c subroutine accuracy(qtot, qctot, qmaxo, qmaxc, qmax, qqmax, hiryu, peakg,
c & qsotai, qsotai2, qsotai3, qsotai4, qsotai5)
c
c common/st1/r(100), qo(200), qc(200), er(200)
c common/st2/n, m, m1, m2, n1, h, h2, h3, h4, area
c common/st3/nr, nq, xnq, wk1, wk2, wp1, wp2, zk2, zp2, rain

```

## 07-2program. for. txt

```

qsotai=0.0
qsotai2=0.0
qsotai3=0.0
qmaxo=-999.99
qmaxc=-999.99
do 502 i=1,nq
  if(qo(i).gt.qmaxo) qmaxo=qo(i)      !観測ピーク
  if(qc(i).gt.qmaxc) qmaxc=qc(i)    !計算ピーク
502 continue
  qmax=qmaxo*area/3.6
  qqmax=qmaxc*area/3.6
  iko=0.0
  do 501 i=1,nq
    if(qo(i).lt.0.04) go to 500
    iko=iko+1
    qsotai=qsotai+abs(qo(i)-qc(i))/qo(i)
    qsotai2=qsotai2+((qo(i)-qc(i))/qo(i))**2
500 continue
    qsotai3=qsotai3+((qo(i)-qc(i))/qmaxo)**2
501 continue
    xiko=iko
    hiryu=qmax/area                !比流量
    peakg=abs(qmaxo-qmaxc)/qmaxo  !JPE
    qsotai=qsotai/xiko            !JRE
    qsotai2=qsotai2/xiko          !Ew
    qsotai3=qsotai3/xnq          !E
    qsotai4=(qtot-qctot)/qtot    !Ev
    qsotai5=(qmaxo-qmaxc)/qmaxo  !Ep
  return
end

```

湧別川  
湧別川  
丸瀬布

平成13年9月11日

802.0  
 1. 357979 1. 953697 1. 979319 5. 188506 4. 810578 4. 086749 3. 971449  
 2. 882503  
 3. 260431 3. 696010 3. 318081 3. 446193 3. 779282 3. 113103 2. 197108  
 2. 562225  
 3. 087481 4. 413433 4. 195644 4. 093155 0. 871157 0. 704612 0. 813506  
 1. 383602  
 1. 556552 1. 921669 0. 871157 1. 729502 1. 249085 0. 640556 0. 743045  
 0. 416362  
 0. 230600 0. 704612 0. 416362 0. 762262 0. 992862 0. 608528 0. 743045  
 0. 871157  
 0. 435578 0. 640556 0. 454795 0. 416362 0. 102489 0. 000000 0. 454795  
 0. 000000  
 0. 128111 0. 000000 0. 064056 0. 044839 0. 000000 0. 000000 0. 000000  
 0. 000000  
 0. 000000 0. 000000 0. 000000 0. 000000 0. 000000 0. 000000 0. 000000  
 0. 000000  
 0. 000000 0. 000000 0. 000000 0. 000000 0. 000000 0. 000000 0. 000000  
 0. 000000  
 0. 000000 0. 000000 0. 000000 0. 000000 0. 000000 0. 000000 0. 000000  
 0. 000000  
 0. 000000 0. 000000 0. 000000 0. 000000 0. 000000 0. 000000 0. 000000  
 0. 000000  
 0. 000000 0. 000000 0. 000000 0. 000000 0. 000000  
 0. 003411 0. 016374 0. 023523 0. 129497 0. 230115 0. 385174 0. 590781  
 0. 829241  
 0. 960343 1. 071099 1. 141725 1. 191243 1. 265854 1. 401141 1. 552027  
 1. 738390  
 1. 857268 1. 920897 2. 025711 2. 154036 2. 350736 2. 589758 2. 642277  
 2. 745701  
 2. 658581 2. 572659 2. 476555 2. 315424 2. 180500 2. 080521 1. 941551  
 1. 886504  
 1. 792466 1. 671622 1. 610329 1. 549916 1. 472195 1. 404925 1. 330138  
 1. 291139  
 1. 269595 1. 231183 1. 209932 1. 155533 1. 077776 1. 049374 0. 974257  
 0. 985421  
 0. 957754 0. 922666 0. 858049 0. 883587 0. 769014 0. 729123 0. 717712  
 0. 657983  
 0. 613118 0. 628866 0. 577852 0. 586670 0. 595585 0. 551600 0. 560320  
 0. 542833  
 0. 518979 0. 508154 0. 459365 0. 448736 0. 419614 0. 378855 0. 368495  
 0. 376090  
 0. 336065 0. 302652 0. 321444 0. 299643 0. 266719 0. 296658 0. 246708  
 0. 236739  
 0. 237942 0. 194971 0. 169075 0. 170033 0. 144431 0. 134804 0. 109691  
 0. 065021  
 0. 080586 0. 056255 0. 037168 0. 018300 0. 004383

09program.for.txt

```

C *****
C 第9回ゼミ
C 損失項を含む貯留関数法(1段タンク型貯留関数モデル)
C
C 作成者 星 清、東海林 勉          作成日 2005.3.7
C *****
C
C 損失項を含む貯留関数法のパラメータ同定
C*** モデル定数 c11, c12及び c13の同時最適化 ***
C
C 一階ニュートン法 (newton method)
C ヤコビアン(jacobian)行列 (一次微係数使用)
C*****
C
C storage-discharge relationship
C  $s = k11*q^{p1} + k12*d(q^{p2})/dt$ 
C  $ds/dt = r - q - b + q0$ 
C  $q0 = qb * \exp(-ram*t)$  !基底流出量
C qb = 初期流出高
C ram = ハイドログラフ低減部の減衰係数
C b = k13*q (b = 損失高)
C k11 = c11*A**(0.24)
C k12 = c12*k11**2*(rave)**(-0.2648)
C c13 = 1 + k13
C p1 = 0.6 and p2 = 0.4648
C
C r= 観測雨量
C q= 観測流量
C
C*****
C optimization of parameters c11, c12 and c13
C*****
C -----
C character kouzui*128, suikei*128, kasen*128
C & , chiten*128
C
C dimension q(200), pas(200,9), dpa(9), co(9)
C dimension x(9), u(18), z(9)
C
C common/st1/r(200), q0(200), qc(200), er(200), q00(200)
C common/st2/n, m, m1, m2, n1, h, h2, h3, h4, area
C common/st3/nr, nq, xnq, wk1, wk2, wp1, wp2, zk2, zp2, wc3, rain, q0in
C
C open(1, file='input.dat', status='unknown')
C open(5, file='損失項を含む貯留関数法.out', status='unknown')
C
C p1=0.6 p2=0.4648
C data wp1, zp2/0.6, 0.4648/
CC
C write(5,*)'損失項を含む貯留関数法(1段タンク型貯留関数モデル)'
C write(5,*)'一階ニュートン法(一次微係数使用)'
C write(5,*)'モデル定数 c11, c12及び c13の同時最適化'
C write(5,'()')
CC ncase:計算洪水数
C read(1,*) ncase
C write(5,'('計算洪水数:',i3)') ncase
C write(5,'()')
C

```

```

do 9999 kk=1, ncase
c
  read(1, 3) suihei !水系名
  read(1, 3) kasen !河川名
  read(1, 3) chiten !観測所名
  read(1, 3) kouzui !洪水名
c
cc  area : 流域面積
c    ipas : 計算の実行判別 (1:計算する 0:計算しない)
c    n: 微分方程式の階数 (n=2) m: モデル定数の個数 (m=3) nl: 分割数 (nl=10)
c    kount: 最大繰り返し回数 (20) p: 収束条件 (p=0.001)
  read(1, '(f8.0, i5)') area, ipas
  read(1, 4) n, m, nl, kount, p
c    ramda : ハイドログラフ低減部の減衰係数 (ram)
  read(1, '(f8.0)') ram
c    初期値 : c11, c12, c13
  read(1, 5) cc1, cc2, cc3
cc   nr: 雨量データ数 nq: 流量データ数
  read(1, '(2i5)') nr, nq
cc   実測雨量データ入力
  do i=1, nq
    r(i)=0.0
  end do
  read(1, 2) (r(i), i=1, nr)
cc   実測流量データ入力 (m3/s)
  read(1, 2) (q(i), i=1, nq)
c
c    流出高 (mm/h) に変換
  do 1001 i=1, nq
1001 q0(i)=3.6*q(i)/area
c
  if(ipas.eq.0) go to 9999
c
c    rave = 平均雨量強度
  nzero=0
  rtot=0.0
  do 199 i=1, nr
    if(r(i).eq.0.0) go to 199
    nzero=nzero+1
    rtot=rtot+r(i)
199 continue
  xnz=nzero
  rave=rtot/xnz
cc   基準化パラメータ
  yc1=1.0
  yc2=1.0
  yc3=1.0
c
  co(1)=cc1
  co(2)=cc2
  co(3)=cc3
c
  1 format(4i8, 4f8.0)
  2 format(10f8.0)
  3 format(a128)
  4 format(4i5, 2f8.0)
  5 format(3f8.0)
c

```

```

                                09program.for.txt
write(5,('水系:',a80)) suihei      !水系
write(5,('河川名:',a80)) kasen     !河川名
write(5,('観測地点:',a80)) chiten  !観測地点
write(5,('洪水年月日:',a80)) kouzui !洪水年月日
write(5,(' '))
write(5,('流量データ数=' i3,' ', 計算時間間隔=1/' ', i2')) nq, n1
write(5,(' '))
write(5,('λ (ramda)=' , f8.4')) ram
write(5,(' '))
write(5,210)
210 format('流域面積', '平均雨量強度', 'c11初期値', 'c12初期値',
& 'c13初期値')
write(5,('2f10.2,3f13.3')) area, rave, cc1, cc2, cc3
c
xnl=n1
h=1./xnl
h2=h**2
h3=h2*h
h4=h3*h
m1=m+1
m2=2*m
xnq=nq
q01=q0(1)      !初期流量
do 20 i=1, nq
q02=q0(1)*exp(-ram*i)
q00(i)=(q01+q02)*0.5 !基底流出量
q01=q02
20 continue
c
write(5,(' '))
write(5,(' '))
write(5,*)' No c11 c12 c13 rmse kai2'
c
最適化開始 (一階ニュートン法)
c
fac=0.0
wp2=1./zp2
do 999 kkll=1, kount
c
モデル定数の更新
wk1=cc1*yc1*area**0.24
zk2=cc2*yc2*wk1**2*rave**(-0.2648)
wk2=1./zk2
wc3=cc3*yc3
c
状態変量初期値
do 800 i=1, n
800 x(i)=0.0
x(1)=q0(1)**zp2
c
感度係数初期値
do 802 i=1, m2
802 u(i)=0.0
sum=0.0
skai2=0.0
c
do 30 ll=1, nq
rain=0.0
if(ll.le.nr) rain=r(ll)

```

```

q0in=q00(11)
qq=q0(11)
qs=0.0
if(qq.gt.0.0) qs=sqrt(qq)
do 40 k=1,n1
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.)
c 計算流出高
qc(11)=qcc
c 誤差項
err=qq-qcc
er(11)=err
if(qq.eq.0.0) go to 49
erw=err/qs
skai2=skai2+erw**2
49 continue
sum=sum+err**2
c** sensitivity coefficients
c
pas(11,1)=cc*u(1)*area**0.24 !c11に関する感度係数
pas(11,2)=cc*u(2)*wk1**2*rave**(-0.2648) !c12に関する感度係数
pas(11,3)=cc*u(3) !c13に関する感度係数
c 規準化感度係数の計算
do 43 i=1,m
43 pas(11,i)=co(i)*pas(11,i)
c
c** error between observed and computed discharges
pas(11,m1)=err
30 continue
c
skai2=skai2/xnq !kai2
sum=sqrt(sum/xnq) !rmse
c
write(5,779) kk11,cc1*yc1,cc2*yc2,cc3*yc3,sum,skai2
779 format(1h ,i5,5f8.3)
c
z(1)=yc1
z(2)=yc2
z(3)=yc3
c** component regression method
c** to solve the correction terms of parameters
call momreg(9,m1,200,nq,pas,dpa)
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)
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
        yc1=z(1)
        yc2=z(2)
        yc3=z(3)
999  continue
879  continue
c
      qtot=0. 0
      qctot=0. 0
      do 1002 i=1, nq
c** total observed runoff depth
        qtot=qtot+q0(i) !総実測流出高
c** total computed runoff depth
        qctot=qctot+qc(i) !総計算流出高
      1002 continue
c
      call accuracy(qtot, qctot, qmaxo, qmaxc, qmax, qqmax, hiryu, peakg,
        & qsotai, qsotai2, qsotai3, qsotai4, qsotai5) !精度評価指標
c
c---OUTPUT 2
      write(5, '()')
      write(5, 211)
211  format(3x, '収束回数', 4x, 'c11', 9x, 'c12', 9x, 'c13',
        &8x, 'rmse', 8x, 'kai2', 9x, 'Jpe', 9x, 'Jre')
        write(5, '(i7, 8f12. 3)') kkl1, cc1*yc1, cc2*yc2, cc3*yc3,
        &sum, skai2, peakg, qsotai
        write(5, '()')
        write(5, 214)
214  format(5x, 'Ew', 9x, 'E', 9x, 'Ev', 8x, 'Ep')
        write(5, '(4f10. 3)') qsotai2, qsotai3, qsotai4, qsotai5
        write(5, '()')
        write(5, 212)
212  format(4x, '比流量', 3x, '総雨量', 4x, '観測総流出', 3x, '計算総流出', 3x,
        &'観測ピーク値', 3x, '計算ピーク値')
        write(5, '(3f10. 3, 4f15. 3)') hiryu, rtot, qtot, qctot, qmax, qqmax
        write(5, '()')
        write(5, '()')
        write(5, 204)
c
204  format('          NO', 6x, '実測雨量', 6x, '実測流出高', 6x, '計算流出高'
        & , 6x, '基底流出高', 8x, '損失高')
c
      do i=1, nq
        write(5, 207) i, r(i), q0(i), qc(i), q00(i), qc(i)*(cc3*yc3-1. 0)
      end do
207  format(i8, f12. 2, 5f16. 4)
c
9999 continue
c
      stop
      end
c
c-----
      subroutine accuracy(qtot, qctot, qmaxo, qmaxc, qmax, qqmax, hiryu, peakg,

```



```

                                09program. for. txt
& qsotai, qsotai2, qsotai3, qsotai4, qsotai5)
c
  common/st1/r (200), q0 (200), qc (200), er (200), q00 (200)
  common/st2/n, m, m1, m2, n1, h, h2, h3, h4, area
  common/st3/nr, nq, xnq, wk1, wk2, wp1, wp2, zk2, zp2, wc3, rain, q0in
  qsotai=0. 0
  qsotai2=0. 0
  qsotai3=0. 0
  qmaxo=-999. 99
  qmaxc=-999. 99
  do 502 i=1, nq
    if(q0(i). gt. qmaxo) qmaxo=q0(i)      !観測ピーク
    if(qc(i). gt. qmaxc) qmaxc=qc(i)     !計算ピーク
502 continue
    qmax=qmaxo*area/3. 6
    qqmax=qmaxc*area/3. 6
    iko=0. 0
    do 501 i=1, nq
      if(q0(i). lt. 0. 04) go to 500
      iko=iko+1
      qsotai=qsotai+abs(q0(i)-qc(i))/q0(i)
      qsotai2=qsotai2+((q0(i)-qc(i))/q0(i))**2
500 continue
      qsotai3=qsotai3+((q0(i)-qc(i))/qmaxo)**2
501 continue
      xiko=iko
      hiryu=qmax/area                    !比流量
      peakg=abs(qmaxo-qmaxc)/qmaxo      !Jpe (ピーク相対誤差)
      qsotai=qsotai/xiko                 !Jre (ハイドログラフ相対誤差)
      qsotai2=qsotai2/xiko               !Ew
      qsotai3=qsotai3/xnq                !E
      qsotai4=(qtot-qctot)/qtot          !Ev
      qsotai5=(qmaxo-qmaxc)/qmaxo       !Ep
    return
  end

c
c-----
c      subroutine gesto(x, u, ijk)
cc
c      損失項を含む貯留関数法(1段タンク型貯留関数モデル)
c      のパラメータ同定
cc
c**      s = k1*q**p1 + k2*d(q**p2)/dt
c**      x(k+1)= phi*x(k) + gamma*b(k)
cc*****
c      一階ニュートン法(newton method)
c
c      ヤコビアン(jacobian)行列 (一次微係数)
cc*****
c**      ijk = 1; solve sensitivty equation
c**      ijk = 2; solve differential equation
      dimension x(1), u(1), y(9), uu(18), b(9)
      common/st2/n, m, m1, m2, n1, h, h2, h3, h4, area
      common/st3/nr, nq, xnq, wk1, wk2, wp1, wp2, zk2, zp2, wc3, rain, q0in
c
      con=wk1*wk2*wp1*wp2
      a=0. 0
      c=0. 0

```

```

d=0.0
e=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
14 a1=-con*(wp1*wp2-1.)*a*y2-wk2*wp2*c*wc3
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*wc3 - 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*wc3*(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

```

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
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

```

	1	2	3	10	20	0.0010				
湧別川										
湧別川										
丸瀬布										
平成13年9月11日	802.0	1								
	0.014									
	10.000	0.150		1.500						
	144	144								
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2.12	0.00	0.00	0.10	0.34	0.00	0.65	4.07	3.45	4.05	
5.18	3.05	3.09	8.10	7.51	6.38	6.20	4.50	5.09	5.77	
1.36	5.38	5.90	4.86	3.43	4.00	4.82	6.89	6.55	6.39	
1.16	1.10	1.27	2.16	2.43	3.00	1.36	2.70	1.95	1.00	
0.68	0.65	0.36	1.10	0.65	1.19	1.55	0.95	1.16	1.36	
0.10	1.00	0.71	0.65	0.16	0.00	0.71	0.00	0.20	0.00	
0.00	0.07	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
9.09	9.40	9.40	9.40	9.09	9.09	9.09	9.09	9.09	9.09	
12.43	9.09	9.09	8.78	8.78	9.09	9.09	9.40	9.72	10.69	
275.79	16.30	18.86	43.45	66.84	102.36	149.14	203.24	233.43	259.08	
554.89	287.79	305.39	336.51	371.10	413.59	441.05	456.20	480.53	510.10	
473.50	609.12	621.79	645.81	627.38	609.21	588.78	553.86	524.78	503.48	
333.57	462.21	442.24	416.29	403.62	391.13	374.80	360.79	345.10	337.39	
251.65	325.99	322.23	311.09	294.74	289.39	273.63	277.10	271.91	265.07	
202.94	258.31	233.77	225.86	224.29	211.96	202.94	207.43	197.04	199.98	
162.12	194.12	197.04	194.12	189.78	188.35	178.45	177.06	171.55	163.45	
	164.79	156.84	150.38	155.54	151.66	145.30	152.95	142.80	141.55	

input.txt

142.80									
134.20	129.41	130.60	125.87	124.70	120.08	111.11	115.55	111.11	
107.83									
104.61	102.48	102.48	101.43	98.30	97.27	95.22	93.19	90.20	
89.21									
88.23	85.31	82.44	80.56	80.56	78.69	79.62	78.69	75.94	
73.51									
71.46	71.46	70.58	68.84	67.98	67.98	66.27	65.42	61.27	
59.64									
59.64	58.84	58.84	57.25						

10program. for. txt

```

C *****
C 第10回ゼミ
C   フィルター分離法を活用した2段タンク型貯留関数モデル
C   地下水流出成分を含む貯留関数法
C   (2段タンク型非線形貯留関数モデル) のパラメータ同定
C
C 作成者 星 清、東海林 勉          作成日 2005. 5. 16
C *****
C
C***   フィルター分離法による地下水流出成分分離          ***
C***   表面・中間流出成分(非線形)に関して定数最適化          ***
C***   地下水流出成分(線形)は最適化しない          ***
C***   モデル定数  c11, c12及びc13の同時最適化          ***
C
C   一階ニュートン法 (newton method)
C   ヤコビアン(jacobian)行列 (一次微係数使用)
C*****
C***   surface-subsurface runoff component
C       s1 = k11*q1**p1 + k12*d(q1**p2)/dt
C       ds1/dt = r-q1-b
C       q1 = surface-subsurface runoff component
C       b = k13*q1 (b = 地下水成分への浸透供給量)
C       k11 = c11*A**(0. 24)
C       k12 = c12*k11**2*(rave)**(-0. 2648)
C       c13 = 1 + k13
C       p1 = 0. 6 and p2 = 0. 4648
C
C***   groundwater flow component
C       s2 = k21*q2 + k22*d(q2)/dt
C       ds2/dt = b - q2
C       q2 = groundwater flow component
C       k21 = c1*k22
C       k22 = k13/c0
C       c0 = (delta/tc)**2
C       c1 = delta**2/tc
C
C***   q = q1 + q2 (total runoff)
C
C       delta( $\delta$ ) = 減衰係数 (非振動解条件)
C        $\delta = 2.1$  を使用
C       tc = 地下水成分分離時定数
C       tc : ハイドログラフ低減部の減衰係数 ( $\lambda$  : ramda) の逆数
C
C***   入力条件   ; r= 観測雨量   ; q= 観測流量
C
C*****
C   optimization of parameters c11, c12 and c13
C*****
C
C   character suikei*128, kasen*128,
C   & chiten*128, kouzui*128
C
C   dimension q(200), q1(200), q2(200), qr(200, 20)
C   dimension pas(200, 9), dpa(9), co(9)
C   dimension x(9), u(18), z(9)
C
C   common/st1/r(200), q0(200), qc(200), er(200), qc1(200), qc2(200)
C   common/st2/n, m, m1, m2, n1, h, h2, h3, h4, area

```

```

                                10program. for. txt
common/st3/nr, nq, xnq, wk1, wk2, wp1, wp2, zk2, zp2, wc3, rain
common/st4/wk21, wk22, c00, c01
common/st5/sum, skai2
C
open(1, file=' input. dat', status=' unknown')
open(6, file=' 2段タンク (分離非線形). out', status=' unknown')
C
wp1=0. 6
zp2=0. 4648
delta=2. 1
C
write(6, *) ' 地下水流出成分を含む貯留関数法'
write(6, *) ' (2段タンク型非線形貯留関数モデル)'
write(6, *) ' フィルター分離法による地下水流出成分分離'
write(6, *) ' 一階ニュートン法 (一次微係数使用)'
write(6, *) ' モデル定数 c11, c12及びc13の同時最適化'
write(6, ' ()')
cc  ncase: 計算洪水数
read(1, *) ncase
write(6, ' ( ' 計算洪水数 : ' ', i3)') ncase
write(6, ' ()')
C
read(1, 3) suikei !水系名
read(1, 3) kasen !河川名
read(1, 3) chiten !観測所名
read(1, 3) kouzui !洪水名
C
cc  area : 流域面積
C  ipas : 計算の実行判別 (1:計算する 0:計算しない)
C  n: 微分方程式の階数 (n=2) m: モデル定数の個数 (m=3) nl: 分割数 (nl=10)
C  kount: 最大繰り返し回数 (20) p: 収束条件 (p=0. 001)
read(1, ' (f8. 0, i5)') area, ipas
read(1, 4) n, m, nl, kount, p
C  tc : 流出成分分離時定数 (ハイドログラフ低減部の減衰係数 (ram) の逆数)
read(1, ' (f8. 0)') tc
C  初期値 : c11, c12, c13
read(1, 5) cc1, cc2, cc3
cc  nr: 雨量データ数 nq: 流量データ数
read(1, ' (2i5)') nr, nq
cc  実測雨量データ入力
do i=1, nq
r(i)=0. 0
end do
read(1, 2) (r(i), i=1, nr)
cc  実測流量データ入力 (m3/s)
read(1, 2) (q(i), i=1, nq)
C
C  流出高 (mm/h) に変換
do 1001 i=1, nq
1001 q0(i)=3. 6*q(i)/area !実測流出高
C
nzero=0
rtot=0. 0
do 199 i=1, nr
if(r(i). eq. 0. 0) go to 199
nzero=nzero+1 !雨量の個数 (0以外)
rtot=rtot+r(i) !総雨量
199 continue

```



10program. for. txt

```
xnz=nzero
rave=rtot/xnz
cc 基準化パラメータ      !平均雨量強度
    yc1=1.0
    yc2=1.0
    yc3=1.0
c
    co(1)=cc1
    co(2)=cc2
    co(3)=cc3
c
    1 format(4i8,4f8.0)
    2 format(10f8.0)
    3 format(a128)
    4 format(4i5,2f8.0)
    5 format(3f8.0)
c
c---OUTPUT 1
    write(6,('水系:',a80)) suihei      !水系
    write(6,('河川名:',a80)) kasen     !河川名
    write(6,('観測地点:',a80)) chiten  !観測地点
    write(6,('洪水年月日:',a80)) kouzui !洪水年月日
    write(6,())
    write(6,('流量データ数 =',i3,' 計算時間間隔 =1/',i2)) nq,nl
    write(6,())
    write(6,('分離時定数(Tc) =',f8.2)) tc
    write(6,())
    write(6,('p1 =',f8.4,' p2 =',f8.4)) wp1,zp2
    write(6,())
    write(6,210)
210 format('流域面積',f10.2,' 平均雨量強度',f13.3,' c11初期値',f10.2,' c12初期値',f10.2,
& ' c13初期値',f10.2)
    write(6,(2f10.2,3f13.3)) area,rave,cc1,cc2,cc3
    write(6,())
    write(6,*) ' No   c11   c12   c13   rmse   kai2'
c
c 全流出高時系列の離散化データ作成
do 11 i=1,nq
do 21 j=1,nl
    if(i.eq.1) then
        qr(i,j)=q0(i)
    else
        qr(i,j)=(q0(i-1)+q0(i))/2.
    end if
21 continue
11 continue
c
    xnl=nl
    h=1./xnl
    h2=h**2
    h3=h2*h
    h4=h3*h
    m1=m+1
    m2=2*m
c
c--- フィルター分離法による地下水流出成分の分離 ---
    c00=(delta/tc)**2
    c01=delta**2/tc
```

```

c   地下水流出成分の係数
aa1=-c00
aa2=-c01
aa3=aa1+aa2**2
aa4=aa1+aa3

c
c   地下水流出成分の計算
c** 2段目タンクのΦの値
ff1 = 1. +0.5*aa1*h2+aa1*aa2*h3/6. +aa1*aa3*h4/24.
ff2 = h*(1. + 0.5*aa2*h+aa3*h2/6. +aa2*aa4*h3/24.)
ff3 = aa1*ff2
ff4 = 1. +aa2*h+0.5*aa3*h2+aa2*aa4*h3/6. +
& (aa1*aa3+aa2**2*aa4)*h4/24.
c** 2段目タンクのγの値
gg2 = h2*(0.5+aa2*h/6. +aa3*h2/24.)
gg4=ff2

c
y1=0.0
y2=0.0
nnn=0
x1=c00*q0(1)
do 10 i=1, nq
do 20 j=1, nl
yy1=ff1*y1+ff2*y2+gg2*x1
yy2=ff3*y1+ff4*y2+gg4*x1
y1=yy1
y2=yy2
if(y1. le. 0.0) y1=0.0
x1=c00*qr(i, j)
20 continue

c
c   表面・中間流出成分と地下水流出成分の分離
if(q0(i). ge. y1) then
q2(i)=y1          !q2: 地下水流出成分 (q2<q0→q2=y1)
q1(i)=q0(i)-q2(i) !q1: 表面・中間流出成分
nnn=nnn+1        !分離時刻
else
q1(i)=0.0
q2(i)=q0(i)      !q2: 地下水流出成分 (q2>q0→q2=q0)
end if
10 continue

c
wp2=1./zp2
fac=0.0
sum=0.0
skai2=0.0

c
c---- 1段目タンクにおける表面・中間流出成分の最適化 ----
c   最適化データ数(nq = nnn)
nq0=nq
nq=nnn

c
c   最適化開始(一階ニュートン法)
c
do 999 kkl=1, kount

c
c** モデル定数の更新
c   表面・中間流出成分(1段目タンク)の貯留係数

```

10program. for. txt

```
wk1=cc1*yc1*area**0.24
zk2=cc2*yc2**wk1**2*rave**(-0.2648)
wk2=1./zk2
wc3=cc3*yc3
C
c** 流量と感度係数の初期値設定
do 800 i=1, n
800 x(i)=0.0
x(1)=q1(1)**zp2
C
do 803 i=1, 2*m
803 u(i)=0.0
C
do 30 ll=1, nq
rain=0.0
if(ll.le.nr) rain=r(ll)
qq=q1(ll)
qs=0.0
if(qq.gt.0.0) qs=sqrt(qq)
C
do 40 k=1, nl
c** solution of sensitivity equation !感度係数計算
call onetank(x, u, 1)
c** solution of differential equation !流出高計算
call onetank(x, u, 2)
if(x(1).le.0.0) x(1)=0.0
40 continue
C
c** 表面・中間流出成分
qc1(ll)=0.0
if(x(1).gt.0.0) qc1(ll)=x(1)**wp2 !計算直接流出高
cc=0.0
if(x(1).gt.0.0) cc=wp2*x(1)**(wp2-1.)
c** 誤差項
err=q1(ll)-qc1(ll)
er(ll)=err
if(qq.eq.0.0) go to 49
erw=err/qs
skai2=skai2+erw**2
49 continue
sum=sum+err**2
C
c** sensitivity coefficients of surface-subsurface runoff depth
in terms of c11, c12 and c13
C
pas(ll, 1)=(cc*u(1))*area**0.24 !c11に関する感度係数
pas(ll, 2)=(cc*u(2))*wk1**2*rave**(-0.2648) !c12に関する感度係数
pas(ll, 3)=cc*u(3) !c13に関する感度係数
C
c** sensitivity coefficients of normalized parameters
規準化感度係数の計算
do 43 i=1, m
43 pas(ll, i)=co(i)*pas(ll, i)
C
c** error between observed and computed discharges
pas(ll, m1)=err
30 continue
C
```

10program. for. txt

```

xng=nq
skai2=skai2/xng          !kai2
sum=sqrt(sum/xng)       !rmse
C
write(6,779) kkll, cc1*yc1, cc2*yc2, cc3*yc3, sum, skai2
779 format(1h , i5, 5f8. 3)
C
z(1)=yc1
z(2)=yc2
z(3)=yc3
c** component regression method
c** to solve the correction terms of parameters (dpa)
call momreg(9, m1, 200, nq, pas, dpa)
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)
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
yc1=z(1)
yc2=z(2)
yc3=z(3)
999 continue
879 continue
C
C 表面・中間流出成分と地下水流出成分の再現計算(データ数 = nq)
C
nq=nq0
xng=nq
C
C 地下水流出成分(2段目タンク)の貯留係数
zk22=(wc3-1. 0)/c00
wk22=1. 0/zk22
wk21=c01*zk22
C
C 流量と感度係数の初期値設定
do 85 i=1, 2*n
85 x(i)=0. 0
x(1)=q0(1)**zp2
do 87 i=1, 2*m2
87 u(i)=0. 0
C
c*** 1段目タンクと2段目タンクの流出高を同時計算
do 130 ll=1, nq
rain=0. 0
if(ll. le. nr) rain=r(ll)
do 140 k=1, n1
c** solution of differential equation
call tanksto2(x, u, 2)
if(x(1). le. 0. 0) x(1)=0. 0
if(x(3). le. 0. 0) x(3)=0. 0

```

```

140  continue
c
c  表面・中間流出成分
    qc1(11)=0.0
    if(x(1).gt.0.0) qc1(11)=x(1)**wp2
c  地下水流出成分
    qc2(11)=x(3)
c  全流出高(計算流出高)
    qcc=qc1(11)+qc2(11)
    qc(11)=qcc
130  continue
c
    qtot=0.0
    qctot=0.0
    do 1002 i=1,nq
c** total observed runoff depth
    qtot=qtot+q0(i)                !総実測流出高
c** total computed runoff depth
    qctot=qctot+qc(i)             !総計算流出高
1002 continue
c
    call accuracy(qtot, qctot, qmaxo, qmaxc, qmax, qqmax, hiryu, peakg,
    & qsotai, qsotai2, qsotai3, qsotai4, qsotai5) !精度評価指標
c
c---OUTPUT 2
    write(6, '()')
    write(6, 211)
211  format(3x, '収束回数', 4x, 'c11', 9x, 'c12', 9x, 'c13',
    & 8x, 'rmse', 8x, 'kai2', 9x, 'Jpe', 9x, 'Jre')
    write(6, '(i7, 8f12.3)') kkl1, cc1*yc1, cc2*yc2, cc3*yc3,
    & sum, skai2, peakg, qsotai
    write(6, '()')
    write(6, 214)
214  format(5x, 'Ew', 9x, 'E', 9x, 'Ev', 8x, 'Ep')
    write(6, '(4f10.3)') qsotai2, qsotai3, qsotai4, qsotai5
    write(6, '()')
    write(6, 212)
212  format(4x, '比流量', 3x, '総雨量', 4x, '観測総流出', 3x, '計算総流出', 3x,
    & '観測ピーク値', 3x, '計算ピーク値')
    write(6, '(3f10.3, f12.3, 2f16.3, i14)') hiryu, rtot, qtot, qctot, qmax,
    & qqmax
    write(6, '()')
    write(6, '()')
    write(6, 204)
204  format(7x, 'NO', 6x, '実測雨量', 7x, '実測流出高', 5x, '地下水流出高',
    & 4x, '表面中間流出高', 3x, '計算表面中間流出', 3x, '浸透供給高', 3x,
    & '計算地下水', 2x, '計算全流出高<表中+地下>')
c
    do i=1, nq
    write(6, 207) i, r(i), q0(i), q2(i), q1(i), qc1(i), qc1(i)*(wc3-1.0),
    & qc2(i), qc(i)
    end do
207  format(i8, f12.2, 4f16.4, f19.4, 3f16.4)
c
    stop
    end
c
c*****

```

```

                                10program. for. txt
subroutine onetank(x, u, ijk)
C-----
C   地下水流出成分を含む貯留関数法
C   1段目タンクにおける非線形貯留関数モデルのパラメータ同定
C
C   一階ニュートン法 (newton method)
C   ヤコビアン(jacobian)行列 (一次微係数使用)
C-----
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(3)
common/st2/n, m, m1, m2, n1, h, h2, h3, h4, area
common/st3/nr, nq, xnq, wk1, wk2, wp1, wp2, zk2, zp2, wc3, rain
C
con=wk1*wk2*wp1*wp2
a=0.0
c=0.0
d=0.0
e=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
C   1段目タンクの係数
14  a1=-con*(wp1*wp2-1.)*a*y2-wk2*wp2*c*wc3
    a2=-con*d
    a3=a1+a2**2
    a4=a1+a3
C**  elements of phi matrix (transition matrix)
C   1段目タンクのΦの値
    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
C   1段目タンクのγの値
    g2 = h2*(0.5+a2*h/6.+a3*h2/24.)
    g4=f2
C
    if(ijk.eq.2) go to 104
C
C**  solve sensitivity equation
C**  1段目タンクにおける感度係数の計算
C   感度方程式の強制項計算
    b(1)=-wk2*wp1*wp2*d*y2
    b(2)=wk2**2*(wk1*wp1*wp2*d*y2 + e*wc3 - rain)
    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

```

```

10program.for.txt
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
c
104 continue
c** solve system equation
c** 1段目タンクの状態変量 (y1, y2) の計算
b1=con*(wp1*wp2-1.)*d*y2+wk2*wc3*(wp2-1.)*e+wk2*rain
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 tanksto2(x, u, ijk)
c
C*****
c 地下水流出成分を含む貯留関数法
c (2段タンク型非線形貯留関数モデル)
c 表面・中間流出成分と地下水流出成分の同時計算
C*****
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/st2/n, m, m1, m2, n1, h, h2, h3, h4, area
common/st3/nr, nq, xnq, wk1, wk2, wp1, wp2, zk2, zp2, wc3, rain
common/st4/wk21, wk22, c00, c01
c
con=wk1*wk2*wp1*wp2
wk13=wc3-1.0
a=0.0
c=0.0
d=0.0
e=0.0
y1=x(1)
y2=x(2)
y3=x(3)
y4=x(4)
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
c 1段目タンクの係数
14 a1=-con*(wp1*wp2-1.)*a*y2-wk2*wp2*c*wc3
a2=-con*d
a3=a1+a2**2
a4=a1+a3
c 2段目タンクの係数
a5=-c00/wk13
a6=-c01

```

10program. for. txt

```

a7=a5+a6**2
a8=a5+a7
c** elements of phi matrix (transition matrix)
c 1段目タンクの中 の値
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
c 1段目タンク のγ の値
g2 = h2*(0. 5+a2*h/6. +a3*h2/24.)
g4=f2
c 2段目タンクの中 の値
f5 = 1. +0. 5*a5*h2+a5*a6*h3/6. +a5*a7*h4/24.
f6 = h*(1. + 0. 5*a6*h+a7*h2/6. +a6*a8*h3/24.)
f7 = a5*f6
f8 = 1. +a6*h+0. 5*a7*h2+a6*a8*h3/6. +
& (a5*a7+a6**2*a8)*h4/24.
c 2段目タンク 目 のγ の値
g6 = h2*(0. 5+a6*h/6. +a7*h2/24.)
g8=f6

c
if(ijk. eq. 2) go to 104

c
c** solve sensitivity equation
c** 1段目タンクにおける感度係数の計算
c 感度方程式の強制項計算
b(1)=-wk2*wp1*wp2*d*y2
b(2)=wk2**2*(wk1*wp1*wp2*d*y2 + e*wc3 - rain)
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)
c** 2段目タンクにおける感度係数の計算
c 感度方程式の強制項計算
m3=m2+m
b(4)=c00*wp2*c*u(1)
b(5)=c00*wp2*c*u(2)
b(6)=c00*wp2*c*u(3)+(c00/wk13**2)*y3
do 17 i=1, m
17 uu(i+m2)= f5*u(i+m2) + f6*u(i+m3) + g6*b(i+m)
do 19 i=m1, m2
19 uu(i+m2) = f7*u(i+m) + f8*u(i+m2) + g8*b(i)
do 20 i=1, 2*m2
20 u(i)=uu(i)
return

c
104 continue
c** solve system equation
c** 1段目タンクの状態変量 (y1, y2) の計算
b1=con*(wp1*wp2-1.)*d*y2+wk2*wc3*(wp2-1.)*e+wk2*rain
y(1) = f1*x(1)+f2*x(2)+g2*b1
y(2) = f3*x(1)+f4*x(2)+g4*b1
c** 2段目タンクの状態変量 (y3, y4) の計算
b2=c00*e
y(3) = f5*x(3)+f6*x(4)+g6*b2

```



```

                                10program. for. txt
y(4) = f7*x(3)+f8*x(4)+g8*b2
do 102 i=1, 2*n
102 x(i)=y(i)
return
end

c
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

```

10program. for. txt

```

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
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))
a1=1./u(j, j)
do 5 k=n, j+1, -1
u(k, j)=a1*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
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
c
c*****
subroutine accuracy(qtot, qctot, qmaxo, qmaxc, qmax, qqmax, hiryu, peakg,
& qsotai, qsotai2, qsotai3, qsotai4, qsotai5)
c
common/st1/r(200), q0(200), qc(200), er(200), qc1(200), qc2(200)
common/st2/n, m, m1, m2, n1, h, h2, h3, h4, area

```

10program. for. txt

common/st3/nr, nq, xnq, wk1, wk2, wp1, wp2, zk2, zp2, wc3, rain  
 common/st5/sum, skai2

c

```

qsotai=0.0
qsotai2=0.0
qsotai3=0.0
sum=0.0
skai2=0.0
qmaxo=-999.99
qmaxc=-999.99
do 502 i=1, nq
  if(q0(i).gt.qmaxo) qmaxo=q0(i)      !観測ピーク
  if(qc(i).gt.qmaxc) qmaxc=qc(i)    !計算ピーク
502 continue
  qmax=qmaxo*area/3.6
  qqmax=qmaxc*area/3.6
  iko=0.0
  do 501 i=1, nq
    err=q0(i)-qc(i)                  !誤差項
    if(q0(i).lt.0.04) go to 500
    iko=iko+1
    qsotai=qsotai+abs(err)/q0(i)
    qsotai2=qsotai2+(err/q0(i))**2
500 continue
    sum=sum+err**2
    skai2=skai2+(err**2)/q0(i)
    qsotai3=qsotai3+(err/qmaxo)**2
501 continue
    xiko=iko
    hiryu=qmax/area                  !比流量
    peakg=abs(qmaxo-qmaxc)/qmaxo    !Jpe (ピーク相対誤差)
    qsotai=qsotai/xiko              !Jre (ハイドログラフ相対誤差)
    qsotai2=qsotai2/xiko            !Ew
    sum=sqrt(sum/xnq)               !rmse
    skai2=skai2/xnq                 !kai2
    qsotai3=qsotai3/xnq            !E
    qsotai4=(qtot-qctot)/qtot      !Ev
    qsotai5=(qmaxo-qmaxc)/qmaxo    !Ep
  return
end

```

c

1  
湧別川  
湧別川  
丸瀬布  
平成13年9月11日  
802.0 1  
2 3 10 20 0.0010  
75.8  
10.000 0.150 1.500  
144 144  
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00  
0.00  
0.00 0.00 0.10 0.34 0.00 0.65 4.07 3.45 4.05  
2.12  
3.05 3.09 8.10 7.51 6.38 6.20 4.50 5.09 5.77  
5.18  
5.38 5.90 4.86 3.43 4.00 4.82 6.89 6.55 6.39  
1.36  
1.10 1.27 2.16 2.43 3.00 1.36 2.70 1.95 1.00  
1.16  
0.65 0.36 1.10 0.65 1.19 1.55 0.95 1.16 1.36  
0.68  
1.00 0.71 0.65 0.16 0.00 0.71 0.00 0.20 0.00  
0.10  
0.07 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00  
0.00  
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00  
0.00  
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00  
0.00  
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00  
0.00  
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00  
0.00  
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00  
0.00  
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00  
0.00  
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00  
9.09  
9.40 9.40 9.40 9.09 9.09 9.09 9.09 9.09 9.09  
9.09  
9.09 9.09 8.78 8.78 9.09 9.09 9.40 9.72 10.69  
12.43  
16.30 18.86 43.45 66.84 102.36 149.14 203.24 233.43 259.08  
275.79  
287.79 305.39 336.51 371.10 413.59 441.05 456.20 480.53 510.10  
554.89  
609.12 621.79 645.81 627.38 609.21 588.78 553.86 524.78 503.48  
473.50  
462.21 442.24 416.29 403.62 391.13 374.80 360.79 345.10 337.39  
333.57  
325.99 322.23 311.09 294.74 289.39 273.63 277.10 271.91 265.07  
251.65  
258.31 233.77 225.86 224.29 211.96 202.94 207.43 197.04 199.98  
202.94  
194.12 197.04 194.12 189.78 188.35 178.45 177.06 171.55 163.45  
162.12  
164.79 156.84 150.38 155.54 151.66 145.30 152.95 142.80 141.55

input.txt

142.80									
134.20	129.41	130.60	125.87	124.70	120.08	111.11	115.55	111.11	
107.83									
104.61	102.48	102.48	101.43	98.30	97.27	95.22	93.19	90.20	
89.21									
88.23	85.31	82.44	80.56	80.56	78.69	79.62	78.69	75.94	
73.51									
71.46	71.46	70.58	68.84	67.98	67.98	66.27	65.42	61.27	
59.64									
59.64	58.84	58.84	57.25						

11-1program. for. txt

```

C *****
C 第11回ゼミ
C 全流出量の感度係数を用いた
C 2段タンク型貯留関数モデルの最適化
C 地下水流出成分を含む貯留関数法
C (2段タンク型非線形貯留関数モデル) のパラメータ同定
C
C 作成者 星 清、東海林 勉          作成日 2005. 4. 1
C *****
C
C*** 全流出量の感度係数を用いる          ***
C*** [表面・中間流出成分(非線形) + 地下水流出成分(線形)]          ***
C*** に関して定数最適化                  ***
C*** モデル定数 c11, c12及びc13の同時最適化          ***
C
C 一階ニュートン法 (newton method)
C ヤコビアン(jacobian)行列 (一次微係数使用)
C*****
C*** surface-subsurface runoff component          ***
C s1 = k11*q1**p1 + k12*d(q1**p2)/dt
C ds1/dt = r-q1-b
C q1 = surface-subsurface runoff component
C b = k13*q1 (b = 地下水成分への浸透供給量)
C k11 = c11*A**(0. 24)
C k12 = c12*k11**2*(rave)**(-0. 2648)
C c13 = 1 + k13
C p1 = 0. 6 and p2 = 0. 4648
C
C*** groundwater flow component          ***
C s2 = k21*q2 + k22*d(q2)/dt
C ds2/dt = b - q2
C q2 = groundwater flow component
C k21 = c1*k22
C k22 = k13/c0
C c0 = (delta/tc)**2
C c1 = delta**2/tc
C
C*** q = q1 + q2 (total runoff)
C
C delta( $\delta$ ) = 減衰係数 (非振動解条件)
C  $\delta = 2. 1$  を使用
C tc = 地下水成分分離時定数
C tc : ハイドログラフ低減部の減衰係数 ( $\lambda$  : ramda) の逆数
C
C*** 入力条件 ; r= 観測雨量 ; q= 観測流量
C
C*****
C optimization of parameters c11, c12 and c13
C*****
C -----
C character kouzui*128, suiwei*128, kasen*128
C & , chiten*128
C
C dimension q(200), pas(200, 9), dpa(9), co(9)
C dimension x(9), u(18), z(9)
C dimension qc1(200), qc2(200)
C
C common/st1/r(200), q0(200), qc(200), er(200)

```

```

                                11-1program. for. txt
common/st2/n, m, m1, m2, n1, h, h2, h3, h4, area
common/st3/nr, nq, xnq, wk1, wk2, wp1, wp2, zk2, zp2, wc3, rain
common/st4/wk21, wk22, c00, c01
C
C
open(1, file=' input. dat', status=' unknown')
open(5, file=' 2dan-1new. out', status=' unknown')
C
C
p1=0.6  p2=0.4648 delta=2.1
data wp1, zp2, delta/0.6, 0.4648, 2.1/
CC
write(5, *) ' 地下水流成分を含む貯留関数法'
write(5, *) ' 全流出量の感度係数を用いた'
write(5, *) ' 2段タンク型非線形貯留関数モデルの最適化'
write(5, *) ' 一階ニュートン法(一次微係数使用)'
write(5, *) ' モデル定数 c11, c12及びc13の同時最適化'
write(5, '()')
CC
ncase: 計算洪水数
read(1, *) ncase
write(5, (' 計算洪水数 : ', i3)) ncase
write(5, '()')
C
do 9999 kk=1, ncase
C
read(1, 3) suikei !水系名
read(1, 3) kasen !河川名
read(1, 3) chiten !観測所名
read(1, 3) kouzui !洪水名
C
CC
area : 流域面積
C
ipas : 計算の実行判別(1:計算する 0:計算しない)
C
n: 微分方程式の階数(n=2) m: モデル定数の個数(m=3) n1: 分割数(n1=10)
C
kount: 最大繰り返し回数(20) p: 収束条件(p=0.001)
read(1, ('f8.0, i5')) area, ipas
read(1, 4) n, m, n1, kount, p
C
tc : 流出成分分離時定数 (ハイドログラフ低減部の減衰係数(ram)の逆数)
read(1, ('f8.0')) tc
C
初期値 : c11, c12, c13
read(1, 5) cc1, cc2, cc3
CC
nr: 雨量データ数 nq: 流量データ数
read(1, ('2i5')) nr, nq
CC
実測雨量データ入力
do i=1, nq
r(i)=0.0
end do
read(1, 2) (r(i), i=1, nr)
CC
実測流量データ入力(m3/s)
read(1, 2) (q(i), i=1, nq)
C
C
流出高 (mm/h) に変換
do 1001 i=1, nq
1001 q0(i)=3.6*q(i)/area
C
if(ipas. eq. 0) go to 9999
C
C
rave = 平均雨量強度
nzero=0
rtot=0.0

```

11-1program. for. txt

```

do 199 i=1, nr
  if(r(i).eq.0.0) go to 199
  nzero=nzero+1
  rtot=rtot+r(i)
199 continue
  xnz=nzero
  rave=rtot/xnz
cc   基準化パラメータ
  yc1=1.0
  yc2=1.0
  yc3=1.0
c
  co(1)=cc1
  co(2)=cc2
  co(3)=cc3
c
  1 format(4i8, 4f8.0)
  2 format(10f8.0)
  3 format(a128)
  4 format(4i5, 2f8.0)
  5 format(3f8.0)
c
  write(5, ('水系:', a80)) suihei      !水系
  write(5, ('河川名:', a80)) kasen     !河川名
  write(5, ('観測地点:', a80)) chiten  !観測地点
  write(5, ('洪水年月日:', a80)) kouzui !洪水年月日
  write(5, (''))
  write(5, ('流量データ数=' i3, ' 計算時間間隔=1/' i2, ')) nq, n1
  write(5, (''))
  write(5, ('分離時定数(Tc) =' f8.2, ')) tc
  write(5, (''))
  write(5, 210)
210 format('流域面積', ' 平均雨量強度', ' c11初期値', ' c12初期値',
& ' c13初期値')
  write(5, ('(2f10.2, 3f13.3)')) area, rave, cc1, cc2, cc3
c
  xnl=n1
  h=1./xnl
  h2=h**2
  h3=h2*h
  h4=h3*h
  m1=m+1
  m2=2*m
  xnq=nq
c
c   地下水流出成分の定数決定
c   c0, c1の計算
c
  c00=(delta/tc)**2
  c01=delta**2/tc
c
  write(5, (''))
  write(5, (''))
  write(5, '*') No   c11   c12   c13   rmse   kai2'
c
c   最適化開始 (一階ニュートン法)
c
  fac=0.0

```



```

wp2=1./zp2
do 999 kkll=1, kount
c
c***      モデル定数の更新
c 表面・中間流出成分(1段目タンク)の貯留係数
      wk1=cc1*yc1*area**0.24
      zk2=cc2*yc2*wk1**2*rave**(-0.2648)
      wk2=1./zk2
      wc3=cc3*yc3
c
c 地下水流出成分(2段目タンク)の貯留係数
      zk22=(wc3-1.0)/c00
      wk22=1.0/zk22
      wk21=c01*zk22
c 流量と感度係数の初期値設定
do 800 i=1, 2*n
800  x(i)=0.0
      x(1)=q0(1)**zp2
do 802 i=1, 2*m2
802  u(i)=0.0
c
      sum=0.0
      skai2=0.0
c
do 30 ll=1, nq
      rain=0.0
      if(ll.le.nr) rain=r(ll)
      qq=q0(ll)
      qs=0.0
      if(qq.gt.0.0) qs=sqrt(qq)
do 40 k=1, n1
c**  solution of sensitivity equation
      call tanksto2(x, u, 1)
c**  solution of differential equation
      call tanksto2(x, u, 2)
      if(x(1).le.0.0) x(1)=0.0
      if(x(3).le.0.0) x(3)=0.0
40  continue
c
c 表面・中間流出成分
      qc1(ll)=0.0
      if(x(1).gt.0.0) qc1(ll)=x(1)**wp2
c 地下水流出成分
      qc2(ll)=x(3)
c 全流出高(計算流出高)
      qcc=qc1(ll)+qc2(ll)
      qc(ll)=qcc
      cc=0.0
      if(x(1).gt.0.0) cc=wp2*x(1)**(wp2-1.)
c 誤差項
      err=qq-qcc
      er(ll)=err
      if(qq.eq.0.0) go to 49
      erw=err/qs
      skai2=skai2+erw**2
49  continue
      sum=sum+err**2
c

```

```

                                11-1program.for.txt
c** sensitivity coefficients of total runoff depth
c   in terms of c11,c12 and c13
c
      pas(11,1)=(cc*u(1)+u(7))*area**0.24           !c11に関する感度係数
      pas(11,2)=(cc*u(2)+u(8))*wk1**2*rave**(-0.2648) !c12に関する感度係数
      pas(11,3)=cc*u(3)+u(9)                       !c13に関する感度係数
c
c   sensitivity coefficients of normalized parameters
c   規準化感度係数の計算
      do 43 i=1,m
43   pas(11,i)=co(i)*pas(11,i)
c
c** error between observed and computed discharges
      pas(11,m1)=err
30   continue
c
      skai2=skai2/xnq                               !kai2
      sum=sqrt(sum/xnq)                             !rmse
c
      write(5,779) kk11,cc1*yc1,cc2*yc2,cc3*yc3,sum,skai2
779 format(1h ,i5,5f8.3)
c
      z(1)=yc1
      z(2)=yc2
      z(3)=yc3
c** component regression method
c** to solve the correction terms of parameters(dpa)
      call momreg(9,m1,200,nq,pas,dpa)
      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)
      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
      yc1=z(1)
      yc2=z(2)
      yc3=z(3)
999   continue
879   continue
c
      qtot=0.0
      qctot=0.0
      do 1002 i=1,nq
c** total observed runoff depth
      qtot=qtot+q0(i)                               !総実測流出高
c** total computed runoff depth
      qctot=qctot+qc(i)                             !総計算流出高
1002 continue
c
      call accuracy(qtot,qctot,qmaxo,qmaxc,qmax,qqmax,hiryu,peakg,
& qstotai,qstotai2,qstotai3,qstotai4,qstotai5) !精度評価指標

```

## 11-1program. for. txt

```

c
c---OUTPUT 2
  write(5, ' ( ) ')
  write(5, 211)
211 format(3x, '収束回数', 4x, 'c11', 9x, 'c12', 9x, 'c13',
&8x, 'rmse', 8x, 'kai2', 9x, 'Jpe', 9x, 'Jre')
  write(5, '(i7, 8f12. 3)') kkl1, cc1*yc1, cc2*yc2, cc3*yc3,
&sum, skai2, peakg, qsotai
  write(5, ' ( ) ')
  write(5, 214)
214 format(5x, 'Ew', 9x, 'E', 9x, 'Ev', 8x, 'Ep')
  write(5, '(4f10. 3)') qsotai2, qsotai3, qsotai4, qsotai5
  write(5, ' ( ) ')
  write(5, 212)
212 format(4x, '比流量', 3x, '総雨量', 4x, '観測総流出', 3x, '計算総流出', 3x,
&'観測ピーク値', 3x, '計算ピーク値')
  write(5, '(3f10. 3, f12. 3, 2f16. 3)') hiryu, rtot, qtot, qctot, qmax, qqmax
  write(5, ' ( ) ')
  write(5, ' ( ) ')
  write(5, 204)
204 format(7x, 'N0', 6x, '実測雨量', 7x, '実測流出高',
& 3x, '計算表面中間流出', 3x, '浸透供給高', 3x,
&'計算地下水', 2x, '計算全流出高<表中+地下>')
c
  do i=1, nq
  write(5, 207) i, r(i), q0(i), qc1(i), qc1(i)*(wc3-1. 0),
& qc2(i), qc(i)
  end do
207 format(i8, f12. 2, 4f16. 4, f19. 4, 3f16. 4)
c
9999 continue
c
  stop
  end
c
c-----
c
  subroutine accuracy(qtot, qctot, qmaxo, qmaxc, qmax, qqmax, hiryu, peakg,
& qsotai, qsotai2, qsotai3, qsotai4, qsotai5)
c
  common/st1/r(200), q0(200), qc(200), er(200)
  common/st2/n, m, m1, m2, n1, h, h2, h3, h4, area
  common/st3/nr, nq, xnq, wk1, wk2, wp1, wp2, zk2, zp2, wc3, rain
  qsotai=0. 0
  qsotai2=0. 0
  qsotai3=0. 0
  qmaxo=-999. 99
  qmaxc=-999. 99
  do 502 i=1, nq
  if(q0(i). gt. qmaxo) qmaxo=q0(i)      !観測ピーク
  if(qc(i). gt. qmaxc) qmaxc=qc(i)    !計算ピーク
502 continue
  qmax=qmaxo*area/3. 6
  qqmax=qmaxc*area/3. 6
  iko=0. 0
  do 501 i=1, nq
  if(q0(i). lt. 0. 04) go to 500
  iko=iko+1
  qsotai=qsotai+abs(q0(i)-qc(i))/q0(i)

```

```

                                11-1program. for. txt
qsotai2=qsotai2+((q0(i)-qc(i))/q0(i))**2
500 continue
qsotai3=qsotai3+((q0(i)-qc(i))/qmaxo)**2
501 continue
xiko=iko
hiryu=qmax/area                                !比流量
peakg=abs(qmaxo-qmaxc)/qmaxo                    !Jpe (ピーク相対誤差)
qsotai=qsotai/xiko                              !Jre (ハイドログラフ相対誤差)
qsotai2=qsotai2/xiko                            !Ew
qsotai3=qsotai3/xnq                             !E
qsotai4=(qtot-qctot)/qtot                       !Ev
qsotai5=(qmaxo-qmaxc)/qmaxo                     !Ep
return
end

c
c-----
c      subroutine tanksto2(x, u, ijk)
c
c*****
c      地下水流出成分を含む貯留関数法
c      (2段タンク型非線形貯留関数モデル) のパラメータ同定
c
c      一階ニュートン法 (newton method)
c      ヤコビアン(jacobian)行列 (一次微係数使用)
c*****
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/st2/n, m, m1, m2, n1, h, h2, h3, h4, area
common/st3/nr, nq, xnq, wk1, wk2, wp1, wp2, zk2, zp2, wc3, rain
common/st4/wk21, wk22, c00, c01

c
con=wk1*wk2*wp1*wp2
wk13=wc3-1.0
a=0.0
c=0.0
d=0.0
e=0.0
y1=x(1)
y2=x(2)
y3=x(3)
y4=x(4)
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
c      1段目タンクの係数
14 a1=-con*(wp1*wp2-1.)*a*y2-wk2*wp2*c*wc3
a2=-con*d
a3=a1+a2**2
a4=a1+a3
c      2段目タンクの係数
a5=-c00/wk13

```

11-1program. for. txt

```

a6=-c01
a7=a5+a6**2
a8=a5+a7
c** elements of phi matrix (transition matrix)
c 1段目タンクの中の値
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
c 1段目タンクのγの値
g2 = h2*(0. 5+a2*h/6. +a3*h2/24.)
g4=f2
c 2段目タンクの中の値
f5 = 1. +0. 5*a5*h2+a5*a6*h3/6. +a5*a7*h4/24.
f6 = h*(1. + 0. 5*a6*h+a7*h2/6. +a6*a8*h3/24.)
f7 = a5*f6
f8 = 1. +a6*h+0. 5*a7*h2+a6*a8*h3/6. +
& (a5*a7+a6**2*a8)*h4/24.
c 2段タンク目のγの値
g6 = h2*(0. 5+a6*h/6. +a7*h2/24.)
g8=f6
c
c if(ijk. eq. 2) go to 104
c
c** solve sensitivity equation
c** 1段目タンクにおける感度係数の計算
c 感度方程式の強制項計算
b(1)=-wk2*wp1*wp2*d*y2
b(2)=wk2**2*(wk1*wp1*wp2*d*y2 + e*wc3 - rain)
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)
c** 2段目タンクにおける感度係数の計算
c 感度方程式の強制項計算
m3=m2+m
b(4)=c00*wp2*c*u(1)
b(5)=c00*wp2*c*u(2)
b(6)=c00*wp2*c*u(3)+(c00/wk13**2)*y3
do 17 i=1, m
17 uu(i+m2)= f5*u(i+m2) + f6*u(i+m3) + g6*b(i+m)
do 19 i=m1, m2
19 uu(i+m2) = f7*u(i+m) + f8*u(i+m2) + g8*b(i)
do 20 i=1, 2*m2
20 u(i)=uu(i)
return
c
104 continue
c** solve system equation
c** 1段目タンクの状態変量 (y1, y2) の計算
b1=con*(wp1*wp2-1.)*d*y2+wk2*wc3*(wp2-1.)*e+wk2*rain
y(1) = f1*x(1)+f2*x(2)+g2*b1
y(2) = f3*x(1)+f4*x(2)+g4*b1
c** 2段目タンクの状態変量 (y3, y4) の計算
b2=c00*e

```

11-1program. for. txt

```

y(3) = f5*x(3)+f6*x(4)+g6*b2
y(4) = f7*x(3)+f8*x(4)+g8*b2
do 102 i=1, 2*n
102 x(i)=y(i)
return
end

c
c-----
c      subroutine momreg(n1, n, m1, md, x, dpa)
c**    component regression method
c**    compute the correction terms of parameters (dpa)
c      implicit double precision (a-h, o-z)
      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-----
c      subroutine lowtri(n1, n, p, binv)
c**    lower triangular cholesky factorization
c**    p = u*b

```

## 11-1program. for. txt

```

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)
c      dimension p(n1, n1), binv(n1, n1)
c      dimension u(9, 9), b(9, 9)
c
c      do 5 j=1, n-1
c      u(j, j)=abs(p(j, j))
c      u(j, j)=sqrt(u(j, j))
c      a1=1./u(j, j)
c      do 5 k=n, j+1, -1
c      u(k, j)=a1*p(k, j)
c      be=u(k, j)
c      do 5 i=k, n
5      p(i, k)=p(i, k)-u(i, j)*be
c      u(n, n)=abs(p(n, n))
c      u(n, n)=sqrt(u(n, n))
c
c**  b = transpose of u
c      do 40 i=1, n
c      do 40 j=i, n
40      b(i, j)=u(j, i)
c
c**  compute inverse of b(i, j)
c      binv(1, 1)=1./b(1, 1)
c      do 50 j=2, n
c      binv(j, j)=1./b(j, j)
c      jm1=j-1
c      do 50 k=1, jm1
c      sum=0.0
c      do 52 i=k, jm1
52      sum = sum - binv(k, i)*b(i, j)
50      binv(k, j)=sum*binv(j, j)
c      return
c      end
c
c-----
c      subroutine sqcov(n1, n, m1, md, x, cov)
c**  compute covariance matrix
c      implicit double precision (a-h, o-z)
c      dimension x(m1, n1), cov(n1, n1)
c      do 10 i=1, n
c      do 10 j=1, i
c      s=0.
c      do 12 k=1, md
12      s=s+x(k, i)*x(k, j)
10      cov(i, j)=s
c      return
c      end
c
c

```

```

1
湧別川
湧別川
丸瀬布
平成13年9月11日
802.0      1
  2     3    10   20  0.0010
  75.8
  9.000    0.150  1.500
 144   144
  0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
0.00
  0.00    0.00    0.10    0.34    0.00    0.65    4.07    3.45    4.05
2.12
  3.05    3.09    8.10    7.51    6.38    6.20    4.50    5.09    5.77
5.18
  5.38    5.90    4.86    3.43    4.00    4.82    6.89    6.55    6.39
1.36
  1.10    1.27    2.16    2.43    3.00    1.36    2.70    1.95    1.00
1.16
  0.65    0.36    1.10    0.65    1.19    1.55    0.95    1.16    1.36
0.68
  1.00    0.71    0.65    0.16    0.00    0.71    0.00    0.20    0.00
0.10
  0.07    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
0.00
  0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
0.00
  0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
0.00
  0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
0.00
  0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
0.00
  0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
0.00
  0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
0.00
  0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
0.00
  0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
0.00
  0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
0.00
  0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
0.00
  0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
0.00
  9.40    9.40    9.40    9.09    9.09    9.09    9.09    9.09    9.09
9.09
  9.09    9.09    8.78    8.78    9.09    9.09    9.40    9.72    10.69
12.43
 16.30   18.86   43.45   66.84  102.36  149.14  203.24  233.43  259.08
275.79
 287.79  305.39  336.51  371.10  413.59  441.05  456.20  480.53  510.10
554.89
 609.12  621.79  645.81  627.38  609.21  588.78  553.86  524.78  503.48
473.50
 462.21  442.24  416.29  403.62  391.13  374.80  360.79  345.10  337.39
333.57
 325.99  322.23  311.09  294.74  289.39  273.63  277.10  271.91  265.07
251.65
 258.31  233.77  225.86  224.29  211.96  202.94  207.43  197.04  199.98
202.94
 194.12  197.04  194.12  189.78  188.35  178.45  177.06  171.55  163.45
162.12
 164.79  156.84  150.38  155.54  151.66  145.30  152.95  142.80  141.55

```



input.txt

142.80									
134.20	129.41	130.60	125.87	124.70	120.08	111.11	115.55	111.11	
107.83									
104.61	102.48	102.48	101.43	98.30	97.27	95.22	93.19	90.20	
89.21									
88.23	85.31	82.44	80.56	80.56	78.69	79.62	78.69	75.94	
73.51									
71.46	71.46	70.58	68.84	67.98	67.98	66.27	65.42	61.27	
59.64									
59.64	58.84	58.84	57.25						

11-2program. for. txt

```

C *****
C 第11回ゼミ
C 全流出量の感度係数を用いた
C 2段タンク型貯留関数モデルの最適化
C 地下水流出成分を含む貯留関数法
C (2段タンク型非線形貯留関数モデル) のパラメータ同定
C
C 作成者 星 清、宮武 真由子      作成日 2005. 6. 13
C *****
C
C*** 全流出量の感度係数を用いる          ***
C*** [表面・中間流出成分(非線形) + 地下水流出成分(線形)]          ***
C*** に関して定数最適化                  ***
C*** モデル定数 c11, c12及びc13の同時最適化          ***
C
C   ダビドン(davidon)法
C   ヤコビアン(jacobian)行列 (一次微係数)
C   ヘシアン(hessian)行列 (二次微係数)
C*****
C*** surface-subsurface runoff component
C   s1 = k11*q1**p1 + k12*d(q1**p2)/dt
C   ds1/dt = r-q1-b
C   q1 = surface-subsurface runoff component
C   b = k13*q1 (b = 地下水成分への浸透供給量)
C   k11 = c11*A**(0. 24)
C   k12 = c12*k11**2*(rave)**(-0. 2648)
C   c13 = 1 + k13
C   p1 = 0. 6 and p2 = 0. 4648
C
C*** groundwater flow component
C   s2 = k21*q2 + k22*d(q2)/dt
C   ds2/dt = b - q2
C   q2 = groundwater flow component
C   k21 = c1*k22
C   k22 = k13/c0
C   c0 = (delta/tc)**2
C   c1 = delta**2/tc
C
C*** q = q1 + q2 (total runoff)
C   delta( $\delta$ ) = 減衰係数 (非振動解条件)
C    $\delta = 2. 1$  を使用
C   tc = 地下水成分分離時定数
C   tc : ハイドログラフ低減部の減衰係数( $\lambda$  : ramda)の逆数
C
C*** 入力条件 ; r= 観測雨量 ; q= 観測流量
C
C*****
C   optimization of parameters c11, c12 and c13
C
C   character kouzui*128, suikei*128, kasen*128
C   & , chiten*128
C
C   dimension q(200)
C   dimension z(9)
C   dimension ddi(3), ho(3, 3), djo(3), djo2(3, 3), dpa(3)
C
C   common/st1/r(200), q0(200), qc(200), er(200), qc1(200), qc2(200)
C   common/st2/n, m, m1, m2, nl, h, h2, h3, h4, area

```

```

                                11-2program. for. txt
common/st3/nr, nq, xnq, wk1, wk2, wp1, wp2, zk2, zp2, wc3, rain
common/st4/wk21, wk22, c00, c01
common/st5/sum, skai2, co(3), beta(3)
C
open(1, file=' input. dat', status=' unknown')
open(6, file=' 2dan-davi. out', status=' unknown')
C
C
p1=0.6  p2=0.4648  delta=2.1
wp1=0.6
zp2=0.4648
delta=2.1
CC
write(6,*)' 地下水流出成分を含む貯留関数法'
write(6,*)' 全流出量の感度係数を用いた'
write(6,*)' 2段タンク型非線形貯留関数モデルの最適化'
write(6,*)' ダビドン法 (一次・二次感度係数使用)'
write(6,*)' モデル定数 c11, c12及びc13の同時最適化'
write(6,'()')
CC
ncase:計算洪水数
read(1,*) ncase
write(6,'('計算洪水数:',i3)') ncase
write(6,'()')
C
do 9999 kk=1, ncase
C
read(1,3) suikei !水系名
read(1,3) kasen !河川名
read(1,3) chiten !観測所名
read(1,3) kouzui !洪水名
C
CC
area : 流域面積
C
ipas : 計算の実行判別(1:計算する 0:計算しない)
C
n:微分方程式の階数(n=2) m:モデル定数の個数(m=3) nl:分割数(nl=10)
C
kount:最大繰り返し回数(20) p:収束条件(p=0.001)
read(1,'(f8.0,i5)') area, ipas
read(1,4) n, m, nl, kount, p
C
tc : 流出成分分離時定数 (ハイドログラフ低減部の減衰係数(ram)の逆数)
read(1,'(f8.0)') tc
C
初期値 : c11, c12, c13
read(1,5) cc1, cc2, cc3
CC
nr:雨量データ数 nq:流量データ数
read(1,'(2i5)') nr, nq
CC
実測雨量データ入力
do i=1, nq
r(i)=0.0
end do
read(1,2) (r(i), i=1, nr)
CC
実測流量データ入力(m3/s)
read(1,2) (q(i), i=1, nq)
C
C
流出高 (mm/h) に変換
do 1001 i=1, nq
1001 q0(i)=3.6*q(i)/area
C
if(ipas.eq.0) go to 9999
C
C
rave = 平均雨量強度
nzero=0

```

```

rtot=0.0
do 199 i=1, nr
  if(r(i).eq.0.0) go to 199
  nzero=nzero+1
  rtot=rtot+r(i)
199 continue
  xnz=nzero
  rave=rtot/xnz
cc  基準化パラメータ
  yc1=1.0
  yc2=1.0
  yc3=1.0
c
  co(1)=cc1
  co(2)=cc2
  co(3)=cc3
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, ('水系:', a80)) suikei      !水系
  write(6, ('河川名:', a80)) kasen    !河川名
  write(6, ('観測地点:', a80)) chiten !観測地点
  write(6, ('洪水年月日:', a80)) kouzui !洪水年月日
  write(6, ('))
  write(6, ('流量データ数 =', i3, '  計算時間間隔 =1/', i2, ')) nq, n1
  write(6, ('))
  write(6, ('分離時定数(Tc) =', f8.2, ')) tc
  write(6, ('))
  write(6, 210)
210 format('流域面積', '  平均雨量強度', '  c11初期値', '  c12初期値',
& '  c13初期値')
  write(6, ('2f10.2, 3f13.3')) area, rave, cc1, cc2, cc3
c
  xnl=n1
  h=1./xnl
  h2=h**2
  h3=h2*h
  h4=h3*h
  m1=m+1
  m2=2*m
  xnq=nq
c
c  地下水流出成分の定数決定
c  c0, c1の計算
c
  c00=(delta/tc)**2
  c01=delta**2/tc
c
  write(6, ('))
  write(6, ('))
  write(6, *) '  No    c11    c12    c13    rmse    kai2'
c
  fac=0.0
  wp2=1./zp2

```

```

beta(1)=area**0.24
beta(3)=1.0
C
C*** 最適化開始 (davidon法) ***
C
do 999 kll=1, kount
C
C*** モデル定数の更新
C 表面・中間流出成分(1段目タンク)の貯留係数
wk1=cc1*yc1*area**0.24
zk2=cc2*yc2*wk1**2*rave**(-0.2648)
beta(2)=wk1**2*rave**(-0.2648)
wk2=1./zk2
wc3=cc3*yc3
C
C 地下水流出成分(2段目タンク)の貯留係数
zk22=(wc3-1.0)/c00
wk22=1.0/zk22
wk21=c01*zk22
C
call davidon6(djo, djo2, ho)
C
C djo: jacobian matrix of objective function
C djo2: hessian matrix of objective function
C ho: inverse mmatrix of djo2
C
write(6, 779) kll, cc1*yc1, cc2*yc2, cc3*yc3, sum, skai2
779 format(1h , i5, 5f8.3)
C
z(1)=yc1
z(2)=yc2
z(3)=yc3
C
C** solve the correction terms of parameters
do 932 i=1, m
ddi(i)=0.0
do 934 j=1, m
934 ddi(i)=ddi(i)+ho(i, j)*djo(j)
dpa(i)=-ddi(i)
932 continue
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)
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
yc1=z(1)
yc2=z(2)
yc3=z(3)
999 continue

```

```

879  continue
c
      qtot=0.0
      qctot=0.0
      do 1002 i=1,nq
c** total observed runoff depth
      qtot=qtot+q0(i)          !総実測流出高
c** total computed runoff depth
      qctot=qctot+qc(i)       !総計算流出高
1002 continue
c
      call accuracy(qtot, qctot, qmaxo, qmaxc, qmax, qqmax, hiryu, peakg,
      & qsotai, qsotai2, qsotai3, qsotai4, qsotai5) !精度評価指標
c
c---OUTPUT 2
      write(6, '()')
      write(6, 211)
211 format(3x, '収束回数', 4x, 'c11', 9x, 'c12', 9x, 'c13',
      & 8x, 'rmse', 8x, 'kai2', 9x, 'jpe', 9x, 'jre')
      write(6, '(i7, 8f12.3)') kkl1, cc1*yc1, cc2*yc2, cc3*yc3,
      & sum, skai2, peakg, qsotai
      write(6, '()')
      write(6, 214)
214 format(5x, 'Ew', 9x, 'E', 9x, 'Ev', 8x, 'Ep')
      write(6, '(4f10.3)') qsotai2, qsotai3, qsotai4, qsotai5
      write(6, '()')
      write(6, 212)
212 format(4x, '比流量', 3x, '総雨量', 4x, '観測総流出', 3x, '計算総流出', 3x,
      & '観測ピーク値', 3x, '計算ピーク値')
      write(6, '(3f10.3, f12.3, 2f16.3, i14)') hiryu, rtot, qtot, qctot, qmax,
      & qqmax
      write(6, '()')
      write(6, '()')
      write(6, 204)
204 format(7x, 'NO', 6x, '実測雨量', 7x, '実測流出高',
      & 3x, '計算表面中間流出', 3x, '浸透供給高', 3x,
      & '計算地下水', 2x, '計算全流出高<表中+地下>')
c
      do i=1,nq
      write(6, 207) i, r(i), q0(i), qc1(i), qc1(i)*(wc3-1.0),
      & qc2(i), qc(i)
      end do
207 format(i8, f12.2, 4f16.4, f19.4, 3f16.4)
c
9999 continue
c
      stop
      end
c
c-----
c      subroutine davidon6(djp, dj2, pin)
c*****
c      地下水流出成分を含む貯留関数法
c      (2段タンク型非線形貯留関数モデル)のパラメータ同定
c****
c      全流出量の感度係数を用いた
c      2段タンク型非線形貯留関数モデルの最適化
c      (表面・中間流出成分+地下水流出成分)

```

11-2program. for. txt

```

CCCC
C   ダビドン法 (davidon method)
C   ヤコビアン (jacobian) 行列 (一次微係数使用)
C   ヘシアン (hessian) 行列 (二次微係数)
C*****
C   first and second derivatives of objective function in
C   terms of parameters
C   compute the hessian matrix of objective function (dj2)
C   and its inverse matrix (pin)
C
C   dimension x(4), u(12)
C   dimension pax(3), uq2(3,3), ho(3,3), djp(3), dj2(3,3), pp(3,3)
C   dimension u1(3,3), u2(3,3), u3(3,3), u4(3,3), pin(3,3)
C
C   common/st1/r(200), q0(200), qc(200), er(200), qc1(200), qc2(200)
C   common/st2/n, m, m1, m2, n1, h, h2, h3, h4, area
C   common/st3/nr, nq, xnq, wk1, wk2, wp1, wp2, zk2, zp2, wc3, rain
C   common/st4/wk21, wk22, c00, c01
C   common/st5/sum, skai2, co(3), beta(3)
C
C   do 900 i=1, m
C       djp(i)=0.0
C   do 902 j=i, m
C       ho(i, j)=0.0
902 continue
900 continue
C
C   流量と感度係数の初期値設定
C   do 800 i=1, 2*n
800   x(i)=0.0
C       x(1)=q0(1)**zp2
C   do 802 i=1, 2*m2
802   u(i)=0.0
C   二次感度係数の初期値設定
C   do 804 i=1, m
C   do 804 j=i, m
C       u1(i, j)=0.0
C       u2(i, j)=0.0
C       u3(i, j)=0.0
C       u4(i, j)=0.0
804 continue
C
C   sum=0.0
C   skai2=0.0
C
C   do 30 ll=1, nq
C   rain=0.0
C   if(ll. le. nr) rain=r(ll)
C   qq=q0(ll)
C   qs=0.0
C   if(qq. gt. 0.0) qs=sqrt(qq)
C   do 40 k=1, n1
C**  solution of sensitivity equation
C
C   call tanksto3(x, u, u1, u2, u3, u4, 1)
C
C**  solution of differential equation
C

```

```

                                11-2program. for. txt
call tanksto3(x, u, u1, u2, u3, u4, 2)
c
  if(x(1). le. 0. 0) x(1)=0. 0
  if(x(3). le. 0. 0) x(3)=0. 0
40 continue
c
c   表面・中間流出成分
qc1(11)=0. 0
  if(x(1). gt. 0. 0) qc1(11)=x(1)**wp2
c   地下水流出成分
qc2(11)=0. 0
  if(x(3). gt. 0. 0) qc2(11)=x(3)
c   全流出高(計算流出高)
qcc=qc1(11)+qc2(11)
qc(11)=qcc
cc=0. 0
dd1=0. 0
  if(x(1). le. 0. 0) go to 41
  cc=wp2*x(1)**(wp2-1.)
  dd1=wp2*(wp2-1.)*x(1)**(wp2-2.)
41 continue
c   誤差項
err=qq-qcc
er(11)=err
  if(qq. eq. 0. 0) go to 49
  erw=err/qs
  skai2=skai2+erw**2
49 continue
  sum=sum+err**2
c
c   一次微係数の算定
c   first sensitivity coefficients in terms of k11,k12 and c13
do 910 i=1, m
910 pax(i)=cc*u(i)+u(i+m2)
c   first sensitivity coefficients in terms of c11,c12 and c13
do 908 i=1, m
908 pax(i)=beta(i)*pax(i)
c
c   二次微係数の算定
c   second sensitivity coefficients in terms of k11,k12 and c13
do 911 i=1, m
do 911 j=i, m
  uq2(i, j)=cc*u1(i, j)+dd1*u(i)*u(j)+u3(i, j)
911 continue
c   second sensitivity coefficients in terms of c11,c12 and c13
do 915 i=1, m
do 915 j=i, m
  uq2(i, j)=beta(i)*beta(j)*uq2(i, j)
915 continue
c   hessian matrix [G]
do 912 i=1, m
do 912 j=i, m
  ho(i, j)=ho(i, j)+err*uq2(i, j)-pax(i)*pax(j)
912 continue
c   jacobian vector [g]
do 920 i=1, m
920 djp(i)=djp(i)+err*pax(i)
30 continue

```



11-2program. for. txt

```

C
C*** objective function ***
C
      skai2=skai2/xnq                !kai2
      sum=sqrt(sum/xnq)              !rmse
C
C      jacobian matrix of objective function
C      in terms of standardized parameters
C      do 930 i=1, m
930  djp(i)=-2. *co(i)*djp(i)/xnq
C      hessian matrix of objective function
C      do 932 i=1, m
C      do 932 j=i, m
      dj2(i, j)=-2. *co(i)*co(j)*ho(i, j)/xnq
932  continue
C      do 934 i=1, m
C      do 934 j=i, m
      dj2(j, i)=dj2(i, j)
934  continue
C      do 935 i=1, m
C      do 935 j=1, m
935  pp(i, j)=dj2(i, j)
C
C      compute the inverse(pin) of hessian matrix(pp)
C
C      call inverse(m, pp, pin)
C
C      return
C      end
C
C-----
C      subroutine tanksto3(x, u, u1, u2, u3, u4, ijk)
C*****
C      地下水流出成分を含む貯留関数法
C      (2段タンク型非線形貯留関数モデル) のパラメータ同定
C***
C*** 全流出量の感度係数を用いた
C*** 2段タンク型非線形貯留関数モデルの最適化
C*** (表面・中間流出成分+地下水流出成分)
C***
C*** ダビドン法 (davidon method)
C*** ヤコビアン(jacobian)行列 (一次微係数使用)
C*** ヘシアン(hessian)行列 (二次微係数)
C*****
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(4), u(12), y(4), uu(12), b(6)
      dimension u1(3, 3), uu1(3, 3), u2(3, 3), uu2(3, 3)
      dimension u3(3, 3), uu3(3, 3), u4(3, 3), uu4(3, 3)
      dimension ccc(2, 2), dz(2, 3), dw(3, 3), ua(3, 2), uat(2, 3)
      dimension hhz(3, 3), ak(3, 3), bk(3, 3), db(3, 2)
      dimension hhz2(3, 3), ck(3, 3), ckt(3, 3), dw2(3, 3)
      common/st2/n, m, m1, m2, n1, h, h2, h3, h4, area
      common/st3/nr, nq, xnq, wk1, wk2, wp1, wp2, zk2, zp2, wc3, rain
      common/st4/wk21, wk22, c00, c01
C

```

```

con=wk1*wk2*wp1*wp2
wk13=wc3-1.0
a=0.0
c=0.0
d=0.0
e=0.0
f=0.0
g=0.0
y1=x(1)
y2=x(2)
y3=x(3)
y4=x(4)
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=y1**(wp1*wp2-3.)
   g=y1**(wp2-2.)
c   1段目タンクの係数
14 a1=-con*(wp1*wp2-1.)*a*y2-wk2*wp2*c*wc3
   a2=-con*d
   a3=a1+a2**2
   a4=a1+a3
c   2段目タンクの係数
   a5=-c00/wk13
   a6=-c01
   a7=a5+a6**2
   a8=a5+a7
c** elements of phi matrix (transition matrix)
c   1段目タンクのφの値
   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
c   1段目タンクのγの値
   g2 = h2*(0.5+a2*h/6.+a3*h2/24.)
   g4=f2
c   2段目タンクのφの値
   f5 = 1. +0.5*a5*h2+a5*a6*h3/6.+a5*a7*h4/24.
   f6 = h*(1. + 0.5*a6*h+a7*h2/6.+a6*a8*h3/24.)
   f7 = a5*f6
   f8 = 1. +a6*h+0.5*a7*h2+a6*a8*h3/6.+
& (a5*a7+a6**2*a8)*h4/24.
c   2段タンク目のγの値
   g6 = h2*(0.5+a6*h/6.+a7*h2/24.)
   g8=f6
c
c   if(ijk.eq.2) go to 104
c
c** solve sensitivity equation (first derivatives)
c** 1段目タンクにおける一次感度係数の計算
   b(1)=-wk2*wp1*wp2*d*y2
   b(2)=wk2**2*(wk1*wp1*wp2*d*y2 + e*wc3 - rain)

```

11-2program. for. txt

```

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)
c** 2段目タンクにおける一次感度係数の計算
m3=m2+m
b(4)=c00*wp2*c*u(1)
b(5)=c00*wp2*c*u(2)
b(6)=c00*wp2*c*u(3)+(c00/wk13**2)*y3
do 17 i=1, m
17 uu(i+m2)= f5*u(i+m2) + f6*u(i+m3) + g6*b(i+m)
do 19 i=m1, m2
19 uu(i+m2)= f7*u(i+m) + f8*u(i+m2) + g8*b(i)
c
c** solve hessian equation(second derivatives)
c 1段目タンクにおける二次感度係数の計算
c
c2=wp1*wp2
c
c ccc(i, j)=da1/dx (first derivatives in terms of x)
c
ccc(1, 1)=-con*(c2-1.)*(c2-2.)*f*y2
& -wk2*wp2*wc3*(wp2-1.)*g
ccc(1, 2)=-con*(c2-1.)*a
ccc(2, 1)=ccc(1, 2)
ccc(2, 2)=0. 0
c
c dz(i, j)=da1/dk (first derivatives in terms of k)
c
dz(1, 1)=-wk2*c2*(c2-1.)*a*y2
dz(1, 2)=wk2*con*(c2-1.)*a*y2+wk2**2*wp2*wc3*c
dz(1, 3)=-wk2*wp2*c
dz(2, 1)=-wk2*c2*d
dz(2, 2)=wk2*con*d
dz(2, 3)=0. 0
c
c db(i, j)=db1/dx (=transpose of dz(i, j))
c
do 306 i=1, 2
do 306 j=1, m
306 db(j, i)=dz(i, j)
c
c dw(i, j)=db/dk (first derivatives in terms of k)
c
dw(1, 1)=0. 0
dw(1, 2)=wk2**2*c2*d*y2
dw(1, 3)=0. 0
dw(2, 1)=dw(1, 2)
dw(2, 2)=2.*wk2**3*(-wk1*c2*d*y2- e*wc3+ rain)
dw(2, 3)=wk2**2*e
dw(3, 1)=dw(1, 3)
dw(3, 2)=dw(2, 3)
dw(3, 3)=0. 0
c
c*** elements of forcing function in the
c second sensitivity equation
ij=0

```

## 11-2program. for. txt

```

do 191 j=1, 2
do 191 i=1, m
  ij=i j+1
191 ua(i, j)=u(i j)
c transpose of ua
do 193 i=1, m
do 193 j=1, 2
193 uat(j, i)=ua(i, j)
c
c*** (dda1/ddk)=(da1/dx)*uat+(da1/dk)
c
do 202 i=1, 2
do 202 j=1, m
  s=0. 0
do 195 k=1, 2
195 s=s+ccc(i, k)*uat(k, j)
  ak(i, j)=dz(i, j)+s
202 continue
c
c*** (ddb1/ddk)=(db1/dx)*uat+(db1/dk)
c
do 204 i=1, m
do 204 j=1, m
  s=0. 0
do 206 k=1, 2
206 s=s+db(i, k)*uat(k, j)
  bk(i, j)=dw(i, j)+s
204 continue
c
c*** hhz(i, j)=ua*ak(i, j)+bk(i, j)
c
do 207 i=1, m
do 207 j=1, m
  s=0. 0
do 209 k=1, 2
209 s=s+ua(i, k)*ak(k, j)
  hhz(i, j)=bk(i, j)+s
207 continue
c
c** 1段目タンクの二次感度係数
c uu1(i, j)= w11(i, j);hessian matrix in terms of x1
do 220 i=1, m
do 220 j=i, m
220 uu1(i, j)=f1*u1(i, j)+f2*u2(i, j)+g2*hhz(i, j)
c
c uu2(i, j)= w12(i, j);hessian matrix in terms of x2
do 222 i=1, m
do 222 j=i, m
222 uu2(i, j)=f3*u1(i, j)+f4*u2(i, j)+g4*hhz(i, j)
c
c
c 2段目タンクにおける二次感度係数の計算
c dz2(i, j)=da/dk (first derivatives in terms of k)
ccc dz2(1, 3)=c00/wk13**2
c
c dw2(i, j)=db/dk (first derivatives in terms of k)
c
cc=wp2*c

```

```

dd1=wp2*(wp2-1.)*g
c
do 231 i=1, m
do 231 j=i, m
dw2(i, j) = c00*(dd1*u(i)*u(j)+cc*u1(i, j))
dw2(j, i)=dw2(i, j)
231 continue
dw2(3, 3)=dw2(3, 3)-2.*c00*y3/(wk13**3)
c
c*** elements of forcing function in the
c second sensitivity equation
do 192 j=1, m
do 192 i=1, m
ck(i, j)=0.0
192 continue
c
c*** ua2*ak(i, j)
do 194 i=1, m
ck(i, m) = (c00/wk13**2)*u(i+m2)
194 continue
c transpose of ck (db/dx*uat)
do 198 i=1, m
do 198 j=1, m
198 ckt(j, i)=ck(i, j)
cc
c*** hhz2(i, j)=ck(i, j)+ckt(i, j)+dw2(i, j)
c
do 208 i=1, m
do 208 j=i, m
208 hhz2(i, j)=ck(i, j)+ckt(i, j)+dw2(i, j)
c
c** 2段目タンクの二次感度係数
c uu3(i, j)= w21(i, j);hessian matrix in terms of x3
do 226 i=1, m
do 226 j=i, m
226 uu3(i, j)=f5*u3(i, j)+f6*u4(i, j)+g6*hhz2(i, j)
c uu4(i, j)= w22(i, j);hessian matrix in terms of x4
do 228 i=1, m
do 228 j=i, m
228 uu4(i, j)=f7*u3(i, j)+f8*u4(i, j)+g8*hhz2(i, j)
c
c 感度係数の更新
do 20 i=1, 2*m2
20 u(i)=uu(i)
do 224 i=1, m
do 224 j=i, m
u1(i, j)=uu1(i, j)
u2(i, j)=uu2(i, j)
u3(i, j)=uu3(i, j)
224 u4(i, j)=uu4(i, j)
c
return
c
104 continue
c
c** solve system equation
c** 1段目タンクの状態変量(y1, y2)の計算
b1=con*(wp1*wp2-1.)*d*y2+wk2*wc3*(wp2-1.)*e+wk2*rain

```

```

                                11-2program. for. txt
y(1) = f1*x(1)+f2*x(2)+g2*b1
y(2) = f3*x(1)+f4*x(2)+g4*b1
c** 2  段目タンクの状態変量 (y3, y4) の計算
b2=c00*e
y(3) = f5*x(3)+f6*x(4)+g6*b2
y(4) = f7*x(3)+f8*x(4)+g8*b2
do 102 i=1, 2*n
102  x(i)=y(i)
    return
    end

c
c-----
    subroutine inverse(n, a, d)
c
c  compute the inverse of definite matrix
c  逆行列の算定
c  a=input matrix
c  d=inverse of matrix a
c
    dimension a(3, 3), b(9), c(9), d(3, 3)
    do 154 i=1, n
    do 151 j=1, n
    b(j)=a(j, i)
151  a(j, i)=0. 0
    a(i, i)=1. 0
    do 152 j=1, n
152  c(j)=a(i, j)/b(i)
    do 153 j=1, n
    do 153 k=1, n
153  a(j, k)=a(j, k)-c(k)*b(j)
    do 154 j=1, n
154  a(i, j)=c(j)
    do 156 i=1, n
    do 156 j=1, n
156  d(i, j)=a(i, j)
    return
    end

c
c-----
    subroutine accuracy(qtot, qctot, qmaxo, qmaxc, qmax, qqmax, hiryu, peakg,
& qsotai, qsotai2, qsotai3, qsotai4, qsotai5)
c
    common/st1/r(200), q0(200), qc(200), er(200), qc1(200), qc2(200)
    common/st2/n, m, m1, m2, n1, h, h2, h3, h4, area
    common/st3/nr, nq, xnq, wk1, wk2, wp1, wp2, zk2, zp2, wc3, rain
    common/st5/sum, skai2, co(3), beta(3)
c
    qsotai=0. 0
    qsotai2=0. 0
    qsotai3=0. 0
    sum=0. 0
    skai2=0. 0
    qmaxo=-999. 99
    qmaxc=-999. 99
    do 502 i=1, nq
    if(q0(i). gt. qmaxo) qmaxo=q0(i)      !観測ピーク
    if(qc(i). gt. qmaxc) qmaxc=qc(i)    !計算ピーク
502  continue

```

11-2program. for. txt

```

qmax=qmaxo*area/3.6
qqmax=qmaxc*area/3.6
iko=0.0
do 501 i=1,nq
err=q0(i)-qc(i)                !誤差項
if(q0(i).lt.0.04) go to 500
iko=iko+1
qsotai=qsotai+abs(err)/q0(i)
qsotai2=qsotai2+(err/q0(i))**2
500 continue
sum=sum+err**2
skai2=skai2+(err**2)/q0(i)
qsotai3=qsotai3+(err/qmaxo)**2
501 continue
xiko=iko
hiryu=qmax/area                !比流量
peakg=abs(qmaxo-qmaxc)/qmaxo   !Jpe (ピーク相対誤差)
qsotai=qsotai/xiko             !Jre (ハイドログラフ相対誤差)
qsotai2=qsotai2/xiko           !Ew
sum=sqrt(sum/xnq)              !rmse
skai2=skai2/xnq                !kai2
qsotai3=qsotai3/xnq           !E
qsotai4=(qtot-qctot)/qtot     !Ev
qsotai5=(qmaxo-qmaxc)/qmaxo   !Ep
return
end

```





input.txt

142.80									
134.20	129.41	130.60	125.87	124.70	120.08	111.11	115.55	111.11	
107.83									
104.61	102.48	102.48	101.43	98.30	97.27	95.22	93.19	90.20	
89.21									
88.23	85.31	82.44	80.56	80.56	78.69	79.62	78.69	75.94	
73.51									
71.46	71.46	70.58	68.84	67.98	67.98	66.27	65.42	61.27	
59.64									
59.64	58.84	58.84	57.25						

13-1program. for. txt

```

c *****
c 第13回ゼミ
c 河道におけるKinematic wave法と貯留関数法の関係
c 無次元Kinematic waveモデルによる解
c (一山と二山三角形流入ハイドロによる応答)
c
c 作成者 宮武 真由子      作成日 2004. 12. 27
c *****
c
c      dimension q0(500), qs0(100), qs1(100), qs(500), ss(500)
c      integer t, h, j, iq
c      real m, alph, tr, ta, ta1, ta2, ta3, te, dt, ti, dx, sp, qi1, qi2
c      real ome, er, f, ff, fff, c, d, q1, q2, f1, f2, as
c
c      open(2, file=' kinema. out', status=' unknown')
c
c ----- input data -----
c      m=0.6
c      alph=1.0      ! 無次元の場合  $\alpha=1$ 
c      dt=0.05      ! サンプル間隔
c      ti=int(1.0/dt)
c      dx=0.02      ! 河道長 $l=1$ を50分割
c      sp=int(1.0/dx)
c
c 一山か二山を選択
c      write(*,*) 'input the number of peaks inflow data. (1or2)'
c      read(*,*) iq
c      if (iq.eq.2) go to 300
c
c --- input q (1peak) ---
c      ta=6.0
c      tr=12.0      ! ta/tr=0.50の場合
c      te=14.0
c      q0(1)=0.0
c
c      do t=1, te*ti
c          if((t.gt.0.0).and.(t.le.ta*ti)) then
c              q0(t)=2.0*t*dt/ta
c          else if((t.gt.ta*ti).and.(t.le.tr*ti)) then
c              q0(t)=-2.0*(t*dt-tr)/(tr-ta)
c          else if((t.gt.tr*ti).and.(t.le.te*ti)) then
c              q0(t)=0.0
c          end if
c      end do
c      go to 400
c
c --- input q (2peaks) ---
c 300 ta1=1.5
c      ta2=3.0
c      ta3=7.5
c      tr=12.0
c      te=14.0
c      qi1=1.1429    ! 第1ピーク値
c      qi2=2.2857    ! 第2ピーク値
c
c      q0(1)=0.0
c      do t=1, te*ti
c          if((t.gt.0.0).and.(t.le.ta1*ti)) then

```

```

                                13-1program. for. txt
    q0(t)=qi1*t*dt/ta1
    else if((t. gt. ta1*ti). and. (t. le. ta2*ti)) then
    q0(t)=-qi1*(t*dt-ta2)/(ta2-ta1)
    else if((t. gt. ta2*ti). and. (t. le. ta3*ti)) then
    q0(t)=qi2*(t*dt-ta2)/(ta3-ta2)
    else if((t. gt. ta3*ti). and. (t. le. tr*ti)) then
    q0(t)=-qi2*(t*dt-tr)/(tr-ta3)
    else if((t. gt. tr*ti). and. (t. le. te*ti)) then
    q0(t)=0.0
    end if
end do

c ----- calculation of q -----
400 do j=1, sp
    qs0(j)=0.0      ! 初期条件
end do

do i=1, te*ti-1
    do k=1, sp
        qs1(k)=0.0
    end do
    qs0(1)=q0(i); qs1(1)=q0(i+1)    ! 上流端流入流量の更新

do j=1, sp-1
    ome=dt/dx*qs1(j)+alph*qs0(j+1)**m
    er=0.01*ome
c  --- initial condition qs1(j+1) ---
    if ((qs0(j+1). eq. 0.0). and. (qs1(j). eq. 0.0)) then
        qs1(j+1)=0.0
        go to 200
    endif
    c= ((qs0(j+1)+qs1(j))/2.0)**(m-1.0)
    qs1(j+1)=(dt/dx*qs1(j)+alph*m*qs0(j+1)*c)
+      / (dt/dx+alph*m*c)

do h=1, 100
c  write(2,*) i, j, h
c  -----
    f=dt/dx*qs1(j+1)+alph*qs1(j+1)**m
    ff=dt/dx+alph*m*qs1(j+1)**(m-1.0)
    fff=alph*m*(m-1.0)*qs1(j+1)**(m-2.0)
    d=abs((ff/fff)**2.0-2.0*(f-ome)/fff)**0.5
    q1=qs1(j+1)-ff/fff+d
    q2=qs1(j+1)-ff/fff-d
    if (q1. lt. 0.00001) then
        qs1(j+1)=0.0
        go to 100
    else if (q2. lt. 0.00001) then
        qs1(j+1)=0.0
        go to 100
    endif
    f1=dt/dx*q1+alph*q1**m
    f2=dt/dx*q2+alph*q2**m
    if (abs(f1-ome). le. abs(f2-ome)) then
        qs1(j+1)=q1
        if (abs(f1-ome). le. er) go to 100
    else
        qs1(j+1)=q2

```

```

                                13-1program. for. txt
        if (abs(f2-ome).le.er) go to 100
    end if
    continue
end do

100    continue
end do

200    continue
    as=0.0
    do k=1, sp          ! 流量の受け渡し計算
        qs0(k)=qs1(k)
        as=as+qs1(k)**m
    end do
    qs(i+1)=qs1(sp)
    ss(i+1)=dx*(as-(qs1(1)+qs1(sp))/2.0)
end do

c ----- output data -----
    write (2,*) ' m          alph'
    write (2, '(2f11.6)') m, alph
    write (2,*) ' n          q0          qs          ss'
do i=1, te*ti
    write (2, '(i7, 3f12.7)') i, q0(i), qs(i), ss(i)
end do

stop
end

```

13-2program. for. txt

```

c *****
c 第13回ゼミ
c 河道におけるKinematic wave法と貯留関数法の関係
c 無次元貯留関数法による解
c (一山と二山三角形流入ハイドロによる応答)
c
c 作成者 宮武 真由子      作成日 2004. 12. 27
c *****
c
    dimension q0(500), qs(500), ss(500)
    real k1, k2, p1, p2
    real x1, x2, y1, y2, d
    real m, tr, ta, ta1, ta2, ta3, te, dt, ti, qi1, qi2
    real a1, a2, a3, a4, wa, wb, wc, wd, rb, rd, h, h2, h3, h4
    integer t, hh, i, iq

    open(2, file='nls. out', status='unknown')

c ----- input data -----
    m=0. 6
    dt=0. 05      ! サンプ リング 間隔
    ti=int(1. 0/dt)
    tk=0. 01      ! 計算時間間隔
    tt=int(dt/tk)
    k1=0. 9145
    k2=0. 5635
    p1=0. 7453
    p2=0. 2200

    h=tk; h2=h**2. 0; h3=h2*h; h4=h3*h

c 一山か二山を選択
    write(*,*) 'input the number of peaks inflow data. (1or2)'
    read(*,*) iq
    if (iq. eq. 2) go to 300

c --- input q (1peak) ---
    ta=6. 0
    tr=12. 0      ! ta/tr=0. 50の場合
    te=14. 0
    q0(1)=0. 0

    do t=1, te*ti
        if((t. gt. 0. 0). and. (t. le. ta*ti)) then
            q0(t)=2. 0*t*dt/ta
        else if((t. gt. ta*ti). and. (t. le. tr*ti)) then
            q0(t)=-2. 0*(t*dt-tr)/(tr-ta)
        else if((t. gt. tr*ti). and. (t. le. te*ti)) then
            q0(t)=0. 0
        end if
    end do
    go to 400

c --- input q (2peaks) ---
300 ta1=1. 5
    ta2=3. 0
    ta3=7. 5
    tr=12. 0

```

13-2program. for. txt

```

te=14.0
qi1=1.1429      ! 第1ピーク値
qi2=2.2857      ! 第2ピーク値

q0(1)=0.0
do t=1, te*ti
  if((t. gt. 0.0). and. (t. le. ta1*ti)) then
    q0(t)=qi1*t*dt/ta1
  else if((t. gt. ta1*ti). and. (t. le. ta2*ti)) then
    q0(t)=-qi1*(t*dt-ta2)/(ta2-ta1)
  else if((t. gt. ta2*ti). and. (t. le. ta3*ti)) then
    q0(t)=qi2*(t*dt-ta2)/(ta3-ta2)
  else if((t. gt. ta3*ti). and. (t. le. tr*ti)) then
    q0(t)=-qi2*(t*dt-tr)/(tr-ta3)
  else if((t. gt. tr*ti). and. (t. le. te*ti)) then
    q0(t)=0.0
  end if
end do

```

c ----- calculation of q, s -----  
c ----- second order nonlinear storage function method -----

```

400 do i=1, te*ti
  do hh=1, tt
    if (y1. eq. 0.0) then
      a1=0.0
      go to 100
    endif
    a1=-k1/k2*p1/p2*(p1/p2-1.0)*y1**(p1/p2-2.0)*y2
    +
    -y1**(1.0/p2-1.0)/(k2*p2)
100  a2=-k1/k2*p1/p2*y1**(p1/p2-1.0)
    a3=a1+a2**2.0
    a4=a1+a3
c----- elements of phi matrix
    wa=1.0+0.5*a1*h2+a1*a2*h3/6.0+a1*a3*h4/24.0
    wb=h*(1.0+0.5*a2*h+a3*h2/6.0+a2*a4*h3/24.0)
    wc=a1*wb
    wd=1.0+a2*h+0.5*a3*h2+a2*a4*h3/6.0+(a1*a3+a2**2.0*a4)*h4/24.0
c----- elements of gamma matrix
    rb=h2*(0.5+a2*h/6.0+a3*h2/24.0)
    rd=wb
c
    +
    d=k1/k2*p1/p2*(p1/p2-1.0)*y2*y1**(p1/p2-1.0)
    + (1.0/p2-1.0)*y1**(1.0/p2)/k2+q0(i)/k2
    x1=wa*y1+wb*y2+rb*d
    x2=wc*y1+wd*y2+rd*d
    y1=x1
    y2=x2
    if(y1. lt. 0.0) y1=0.0
  end do

  qs(i)=x1**(1.0/p2)
  ss(i)=k1*x1**(p1/p2)+k2*x2
end do

```

13-2program. for. txt

```
c ----- output data -----  
  write (2,*)' k1          k2          p1          p2          m'  
  write (2,'(5f11. 6)') k1,k2,p1,p2,m  
  write (2,*)' n          q0          qs          ss'  
do i=1,te*ti  
  write (2,'(i7,4f12.7)') i,q0(i),qs(i),ss(i)  
end do  
  
stop  
end
```

14-1programl. f. txt

```
C *****
C 第14回ゼミ
C 複合流域における貯留関数法の最適化
C 有効雨量を用いた一般化貯留関数法
C
C 作成者 秋田 大輔          作成日 2005. 6. 27
C *****
C
C 2価の非線形貯留方程式のパラメータ最適化
C 河道追跡における一般化貯留関数モデルパラメータ同定
C 支川に分割流域と河道を含む計算を考慮
C
C 上流端河道流入量がない場合の計算例
C
Ccccc  有効雨量と直接流出高の入力  ccccc
C
C***  河道追跡における貯留関数法は無次元で計算 ***c
C.....
C 一階ニュートン(newton)法
C friction factor fc の1次元探索
C ヤコビアン(jacobian)行列 (一次微係数)
C*****
C storage-discharge relationship
C  $s = k_{11} * q^{p1} + k_{12} * d(q^{p2}) / dt$ 
C optimization of parameters k11 and k12
C  $p1 = 0.6$  and  $p2 = 0.4648$ 
C
C  $k_{11} = 2.8235 * fc * area^{0.24}$ 
C  $k_{12} = 0.2835 * k_{11}^{*2} * rave^{*-0.2648}$ 
C*****
C optimize the friction factor fc
C*****
C
C=====
C character suikei*128, kasen*128, chiten*128, kouzui*128,
  & gp*128, chitenr*128, chiten1*128, chitenl*128
C
C dimension q(200)
C dimension x(9), u(18), er(200)
C dimension rave(50), uut(18), ut(200, 20), qc(200), uua(200), qqt(200),
  & qinlt(200, 20)
C
C common/st1/wk1, wk2, wp1, wp2, rain, q0in, fc, zfc
C common/st2/n, m, m1, m2, h, h2, h3, h4
C ++++++
C 未知パラメータの個数 m=1
C ++++++
C common/date/iy, im, id, ih, nq, nr, nban, area(50)
C common/discha/r0(50, 200), qr0(200), qt(200, 20)
C common/st3/nl, kount, p, ncal, nind, ndat
C パラメータの初期値設定
C common/st4/cp1, cp2
C common/tit/title
C common/st5/jdg(90), chitenl(10), cleg(40), alph(40), cmm(40)
C common/st6/qinl(200), chitenr(20), chiten1
C
C common/alpha/alp(3)
C
```



```

                                14-1program1. f. txt
dimension dlqt (5, 200, 20), dlut (5, 200, 20), dlqbt (5), r (50, 200), q0 (200)
c
open(1, file=' input3. dat', status=' unknown')
open(5, file=' 河道追跡fc. out', status=' unknown')
c
write(5, *) ' 分割流域流出・河道追跡を含む複合流域解析'
write(5, *) ' 有効雨量と直接流出高を用いた一般化貯留関数法の最適化'
write(5, *) ' friction factor fc の1次元探索'
write(5, *) ' 河道追跡における貯留関数法は無次元で計算'
write(5, *) ' 一階ニュートン法(一次微係数使用)'
write(5, ' ()')
c
c==== READ =====
      call input
c==== 流量・雨量の設定 =====
c
      areatot=0.0
      do j=1, nban
      areatot=areatot+area(j)
      end do
c
      do i=1, nq
      q0(i)=qr0(i)*areatot/3.6
      q(i)=q0(i)
      end do
c
      do i=1, nban+1
      do j=1, nr
      r(i, j)=r0(i, j)
      end do
      end do
c
c==== 分割流域における平均雨量強度の算出 =====
c
      do j=1, nban
      rtot=0.0
      nrt=0
      do i=1, nr
      if(r(j, i).gt.0.0) then
      rtot=rtot+r(j, i)
      nrt=nrt+1
      end if
      end do
      rave(j)=rtot/float(nrt)
      end do
c
c 基準点での平均流域雨量強度
c trave = 基準点平均雨量強度
c nzero=0
c rtot=0.0
c do 199 i=1, nr
c if(r(nban+1, i).eq.0.0) go to 199
c nzero=nzero+1
c rtot=rtot+r(nban+1, i)
199 continue
c xnz=nzero
c trave=rtot/xnz
c

```

## 14-1programl. f. txt

```

write(5, '()')
write(5, 210)
210 format(' 流域面積', ' 平均有効雨量強度', ' fc初期値')
write(5, ' (2f11. 2, f13. 3)') areatot, trave, fc
c
c==== パラメータ設定 =====
c
zfc=1. 0
c
gp=' N0      fc      rmse      kai2'
write(5, ' (/a80)') gp
c
dt=3600. 0      !サンプリング時間は1時間
xnl=n1
xnq=nq
fac=0. 0
c
c 上流端河道流入量を分割数(n1)ごとに設定
c
      do i=1, nq
      do k=1, n1
      if(i. eq. 1) then
qinlt(i, k)=qinl(i)
      else
qinlt(i, k)=qinl(i-1)+float(k)*(qinl(i)-qinl(i-1))/xnl
      end if
      end do
      end do
c
c==== 最適化開始 (一階ニュートン法) =====
c
do 999 kkl=1, kount
c
ibn=0
icn=0
      do ll=1, nq
      do i=1, n1
      if(nind. eq. 0) then
qt(ll, i)=0. 0
      else
qt(ll, i)=qinlt(ll, i)
      end if
      ut(ll, i)=0. 0
      end do
      qqt(ll)=qt(ll, n1)
      end do
      qbt=0. 0
      iscl=0
      jscl=0
c
do 900 nnl=1, ndat
c
jdg=1 ; 河道追跡計算、jdg=0 ; 分割流域流出計算
jjdg=jjdg(nnl)
kscl=jjdg/10
c 支川合流判断
if(jscl. eq. 0. and. kscl. eq. 0) go to 906
if(kscl. ne. 0) go to 907
c 本川へ合流、保存量(上流端流量)全てを合計して下流へ

```

```

do kkl=1, jscl
do ll=1, nq
do i=1, nl
qt(ll, i)=qt(ll, i)+dlqt(kkl, ll, i)
ut(ll, i)=ut(ll, i)+dlut(kkl, ll, i)
end do
qqt(ll)=qqt(ll)+dlqt(kkl, ll, nl)
end do
qbt=qbt+dlqbt(kkl)
end do
jscl=0
iscl=0
c ** 支川合流点で下流に河道も流域もない場合
if(jjdg. eq. 2) go to 905
go to 906
907 if(kscl-iscl) 902, 906, 904
c 支川合流終了、同じ支川に分割流域・河道が結合、
c 支川から新たな支川発生
902 if(jscl. le. 1) stop
c 合流支川部に保存量（上流端流量）を加える
do ll=1, nq
do i=1, nl
qt(ll, i)=qt(ll, i)+dlqt(jscl, ll, i)
ut(ll, i)=ut(ll, i)+dlut(jscl, ll, i)
end do
qqt(ll)=qqt(ll)+dlqt(jscl, ll, nl)
end do
qbt=qbt+dlqbt(jscl)
jscl=jscl-1
iscl=kscl
go to 906
904 jscl=jscl+1
c 新たな支川が発生、ランク（jscl）を加え、
c 支川番号（iscl）を保存する
c 今まで累加した量を保存し、流出量等をクリアする
iscl=kscl
do ll=1, nq
do i=1, nl
dlqt(jscl, ll, i)=qt(ll, i)
qt(ll, i)=0.
dlut(jscl, ll, i)=ut(ll, i)
ut(ll, i)=0.
end do
dlqt(jscl, ll, nl)=qqt(ll)
qqt(ll)=0.
end do
dlqbt(jscl)=qbt
qbt=0.
906 jjdg=jjdg-kscl*10
c 分割流域か河道かの判断
c
if(jjdg. eq. 1) go to 901
c
c==== 分割流域における流出計算 =====
c
c ibn: 河道番号
c
c ibn=ibn+1

```

```

c      分割流域での流出計算時間間隔
      h=1./xn1
      h2=h*h
      h3=h2*h
      h4=h3*h
c 貯留関数モデル定数 k1, k2, p1, p2の設定
      wk1=2.8235*fc*zfc*area(ibn)**0.24
      zk2=0.2835*wk1**2*rave(ibn)**(-0.2648)
      wk2=1./zk2
      wp1=cp1
      zp2=cp2
      wp2=1./zp2

c
      alp(1)=2.8235*area(ibn)**0.24
      alp(2)=0.2835*rave(ibn)**(-0.2648)
      alp(3)=alp(1)**2*alp(2)

c
c 流量と感度係数の初期値設定
      do i=1, n
      x(i)=0.0
      end do
      x(1)=0.0
      do i=1, 2
      u(i)=0.0
      end do

c
      do ll=1, nq
      rain=0.0
      if(ll.le.nr) rain=r(ibn, ll)
      do 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
      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.)

c
c      流域流出量の累計 (次の河道への流入量)
c      流域流出量の感度係数ベクトルの累計 (次の河道へ伝達)
      qt(ll, k)=qt(ll, k)+qcc*area(ibn)/3.6
      ut(ll, k)=ut(ll, k)+area(ibn)/3.6*cc*u(1)
      end do
      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.)
      qc(ll)=qcc
      qqt(ll)=qt(ll, nl)
      qbt=qbt
      uua(ll)=cc*u(1)*fc

c
c      qt: 分割流域からの流出量 (次の河道流入量) (累計) (計算時間間隔毎)
c      qqt: 分割流域からの流出量 (次の河道流入量) (累計)
c      (1時間毎) チェック用
c      ut: 分割流域の感度係数 (累計) (計算時間間隔毎)

```

```

c      qbt:流域毎の初期流量の合計
c      uua:感度係数ベクトル (1時間毎) チェック用
c
c      end do
c
c      go to 905
901 continue
c
c==== 河道追跡計算 (無次元領域) =====
c
c      icn=icn+1
c
c      ta, trの抽出 .....
c      call tatrst( nq, qqt, ts, ta, tr )
c      ta/tr=0.5と仮定しても実用上問題なし
c      ta = tr*0.5
c
c      貯留関数モデル定数k3, k4, p3, p4を計算....
c      amm=cmm(icn)
c
c      call kpxcal( amm, ta, tr, wk1, wk02, wp1, wp02 )
c
c      totq = 0.0
c      ncalgt = 0
c      do i = nint(ts), nint(ts+tr)
c          totq = totq+qqt(i)
c          ncalgt = ncalgt+1
c      end do
c
c      上流端平均流入量 (qm)
c      qm = totq/float(ncalgt)
c
c      時間規準化 (t* = ttt)
c      ttt = alph(icn)*cleg(icn)*qm**(cmm(icn)-1.0)
c
c      wk3=wk1
c      zk4=wk02
c      wp3=wp1
c      zp4=wp02
c      河道追跡における無次元計算時間間隔 (h)
c      h=dt/ttt/xn1
c      h2=h*h
c      h3=h2*h
c      h4=h3*h
c
c      wk1=wk3
c      zk2=zk4
c      wk2=1./zk2
c      wp1=wp3
c      zp2=zp4
c      wp2=1./zp2
c      河道流量と感度係数の初期値設定
c      do i=1, n
c          x(i)=0.0
c      end do
c      x(1)=(qt(1,1)/qm)**zp2
c

```

```

u(1)=0.0
u(2)=0.0
c
do 31 ll=1, nq
do 41 k=1, nl
rain=qt(ll, k)/qm
c 無次元量へ変換
c 強制項 (=上流端流入量に関する感度係数)
uut(1)=ut(ll, k)/qm
c** solution of sensitivity equation
call gesto2(x, u, uut, 1)
c** solution of differential equation
call gesto2(x, u, uut, 2)
if(x(1).le.0.0) x(1)=0.0
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.)
c 実領域での流出量と感度係数へ変換
qt(ll, k)=qcc*qm
ut(ll, k)=cc*u(1)*qm
41 continue
c
qqt(ll)=qt(ll, nl)
31 continue
c
905 continue
c
900 continue
c
c===== 目的関数値と相対誤差計算 =====
c
sum=0.0
sotai=0.0
sigev=0.0
sigv=0.0
c
do 32 ll=1, nq
qq=q0(ll)
qs=0.0
if(qq.gt.0.0) qs=sqrt(qq)
c 誤差項
err=qq-qt(ll, nl)
er(ll)=err
if(qq.eq.0.0) go to 49
erw=(err/qs)*(3.6/areatot)**0.5
skai2=skai2+erw**2
sotai=sotai+abs(err)/qq
49 continue
sum=sum+(err*3.6/areatot)**2
c
c 1時間毎の感度係数ベクトル抽出
sigev=sigev+er(ll)*ut(ll, nl)*fc
sigv=sigv+(ut(ll, nl)*fc)**2
32 continue
c
sum=sqrt(sum/xnq) !rmse
skai2=skai2/xnq !kai2

```

## 14-1programl. f. txt

```

C
write(5, '(i2, 3f9. 4)') kkl1, fc*zfc, sum, skai2
C
dfc=sigev/sigv
C
col = abs(dfc/zfc)
if(col.gt.p) go to 877
go to 879
877 fac = 0.5*(1.+fac)
dfc=fac*dfc
xnew=zfc+dfc
if(xnew.gt.0.0) zfc=zfc+dfc
999 continue
879 continue
C
C==== WRITE =====
C
rtot=0.0
qtot=0.0
qctot=0.0
qsotai=0.0
qsotai2=0.0
qsotai3=0.0
qmax=-999.99
qmaxc=-999.99
do 502 i=1, nq
if(q0(i).gt.qmax) qmax=q0(i) !観測ピーク
if(qt(i, n1).gt.qmaxc) qmaxc=qt(i, n1) !計算ピーク
502 continue
iko=0.0
do 503 i=1, nq
rtot=rtot+r(nban+1, i)
qtot=qtot+q0(i)
qctot=qctot+qt(i, n1)
if(q0(i).lt.0.04*areatot/3.6) go to 500
iko=iko+1
qsotai=qsotai+abs(q0(i)-qt(i, n1))/q0(i)
qsotai2=qsotai2+((q0(i)-qt(i, n1))/q0(i))**2
500 continue
qsotai3=qsotai3+((q0(i)-qt(i, n1))/qmax)**2
503 continue
xiko=iko
qtot0=3.6*qtot/areatot !観測流出高
qctot0=3.6*qctot/areatot !計算流出高
hiryu=qmax/areatot !比流量
peakg=abs(qmax-qmaxc)/qmax !Jpe (ピーク相対誤差)
qsotai=qsotai/xiko !Jre (ハイドログラフ相対誤差)
qsotai2=qsotai2/xiko !Ew
qsotai3=qsotai3/xnq !E
qsotai4=(qtot-qctot)/qtot !Ev
qsotai5=(qmax-qmaxc)/qmax !Ep
C
C--OUTPUT 2
write(5, '()')
write(5, 211)
211 format(2x, '収束回数', 5x, 'fc', 8x, 'rmse'
& , 8x, 'kai2', 9x, 'Jpe', 9x, 'Jre')
write(5, '(i7, 8f12. 3)') kkl1, fc*zfc, sum, skai2, peakg, qsotai

```

## 14-1programl. f. txt

```

write(5, '()')
write(5, 214)
214 format(7x, 'Ew', 9x, 'E', 8x, 'Ev', 8x, 'Ep')
write(5, '(4f10.3)') qsotai2, qsotai3, qsotai4, qsotai5
write(5, '()')
write(5, 212)
212 format(4x, '比流量', 4x, '総雨量', 2x, '観測総流出', 3x, '計算総流出', 3x,
& '観測ピーク値', 3x, '計算ピーク値')
write(5, '(3f10.3, 4f15.3)') hiryu, rtot, qtot0, qctot0, qmax, qmaxc
write(5, '()')
C
write(5, 204)
204 format('      NO', 6x, '流域平均有効雨量', 4x, '直接流出高'
& ', 4x, '計算直接流出高')
do i=1, nq
write(5, 207) i, r(nban+1, i), qr0(i), qt(i, nl)*3.6/areatot
end do
207 format(i8, 3f16.4)
C
stop
end
C
C=====
subroutine input
C=====
C ***** 流域流出解析と河道追跡のためのデータ入力
C 分割流域の有効雨量を入力
C 検証地点での直接流出高入力
C
character suikei*128, kasen*128, chiten*128, kouzui*128,
& gp*128, chitenr*128, chiten1*128, chitenl*128
C
common/st1/wk1, wk2, wp1, wp2, rain, q0in, fc, zfc
common/st2/n, m, m1, m2, h, h2, h3, h4
common/date/iy, im, id, ih, nq, nr, nban, area(50)
common/discha/r0(50, 200), qr0(200), qt(200, 20)
common/st3/nl, kount, p, ncal, nind, ndat
common/st4/cp1, cp2
common/tit/title
common/st5/jdg(90), chitenl(10), cleg(40), alph(40), cmm(40)
common/st6/qinl(200), chitenr(20), chiten1
C
データの読み込み
C
n: 微分方程式階数 (n=2)   m: モデル定数の数 (m=1)
C
nl: 1時間の計算分割数
C
kount: 最大繰り返し計算回数   p: 収束条件
C
nban: 流域数   ncal: 河道数   nc2: 下流に流域・河道がない合流点の数
C
nind=0: 上流端流入量なし nind=1: 上流端流入量有り
C
jdg: 流域、河道の計算判別、計算順序---0: 流域 1: 河道
C
cleg: 河道長   alph:  $\alpha$    cmm: m   area: 分割流域面積
C
r: 流域雨量   q0: 基準点流量   qinl: 上流端河道流入量
C
流出モデル定数の初期値 (cp1=0.6, cp2=0.4648)
cp1=0.6
cp2=0.4648
C
read(1, 3) suikei      !水系名
read(1, 3) kasen      !河川名
read(1, 3) chiten     !観測所名

```



```

                                14-1programl. f. txt
    read(1,3) kouzui             !洪水年月日
    3 format(a128)
    4 format(4i5, 2f8. 0)
    5 format(3f8. 0)
    read(1,4) n, m, nl, kount, p
    read(1, '(f8. 0)') fc        !fc初期値入力
    read(1, '(2i5)') nr, nq      !雨量個数・流出高個数
    read(1, '(4i5)') iy, im, id, ih !洪水年月日
    read(1, '(4i5)') nban, ncal, nind, nc2
    ndat=nban+ncal+nc2
    read(1, '(20i5)') (jdg(i), i=1, ndat)
c 河道定数入力
    if(ncal.ge.1) then
    do j=1, ncal
    read(1, '(a40)') chitenl(j)
    read(1, '(i5, 3f8. 0)') no, cleg(j), alph(j), cmm(j)
    end do
    end if
c 分割流域の面積と流域平均雨量
    do j=1, nban
    read(1, '(a40)') chitenr(j)
    read(1, '(i5, f8. 0)') no, area(j)
    read(1, '(10f8. 0)') (r0(j, i), i=1, nr) !分割流域の有効雨量
    end do
    read(1, '(a40)') gp
    read(1, *) (r0(nban+1, i), i=1, nr)
    read(1, '(a40)') chiten1
c 実測流量
    read(1, *) (qr0(i), i=1, nq)      !検証地点での直接流出高入力
c
c 入力データ出力
c
    write(5, ('水系:', a80)) suikei      !水系
    write(5, ('河川名:', a80)) kasen     !河川名
    write(5, ('観測地点:', a80)) chiten  !観測地点
    write(5, ('洪水年月日:', a80)) kouzui !洪水年月日
    write(5, ('洪水開始年月日時:', 4i5)) iy, im, id, ih
    write(5, (''))
    write(5, ('流量データ数=', i3, ' 計算時間間隔= 1/', i2)) nq, nl
    write(5, (''))
    write(5, '(4i5)') nban, ncal, nind, nc2
    write(5, (''))
    write(5, '(20i5)') (jdg(i), i=1, ndat)
c
    return
    end
c
c=====
    subroutine gesto(x, u, ijk)
c=====
c ***** 分割流域における流出計算と感度係数算定
c
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)
    common/st1/wk1, wk2, wp1, wp2, rain, q0in, fc, zfc

```

```

common/st2/n, m, m1, m2, h, h2, h3, h4
c
common/alpha/alp (3)
c
wp3=wp1*wp2-1. 0
con=wk1*wk2*wp1*wp2
a=0. 0
c=0. 0
d=0. 0
e=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
14 a1=-con*(wp1*wp2-1. )*a*y2-wk2*wp2*c
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
c
if (ijk. eq. 2) go to 104
c
c** solve sensitivity equation
c 感度方程式の強制項計算
b=alp (1) * (-wk2* (wp1*wp2) *y1**wp3*y2)
& +2. 0*alp (3) *fc*zfc
& * (wk2**2* (wk1* (wp1*wp2) *y1**wp3*y2+y1**wp2-rain))
uu (1) = f1*u (1) + f2*u (2) + g2*b
uu (2) = f3*u (1) + f4*u (2) + g4*b
c
u (1)=uu (1)
u (2)=uu (2)
c
return
104 continue
c** solve system equation
b1=con*(wp1*wp2-1. )*d*y2+wk2*(wp2-1. )*e+wk2*rain
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 gesto2(x, u, uut, ijk)
C=====
C ***** 河道追跡流量と感度係数の算定
C 数値解法は流域流出計算と同じ
C
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), uut(1)
common/st1/wk1, wk2, wp1, wp2, rain, q0in, fc, zfc
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
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
14 a1=-con*(wp1*wp2-1.)*a*y2-wk2*wp2*c
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
C
if(ijk.eq.2) go to 104
C
C** solve sensitivity equation
C 強制項 (=上流端流入量に関する感度係数)
b=wk2*uut(1)
uu(1) = f1*u(1) + f2*u(2) + g2*b
uu(2) = f3*u(1) + f4*u(2) + g4*b
C
u(1)=uu(1)
u(2)=uu(2)
C
return
104 continue
C** solve system equation
b1=con*(wp1*wp2-1.)*d*y2+wk2*(wp2-1.)*e+wk2*rain
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  tatrst( ncalx, qq, ts, ta, tr )
c=====
c ***** 洪水継続時間とピーク流量生起時刻の算定
c
      dimension  qq(1), qtot(0:720), xdt(720)
c
      qtot(0) = 0.0
      qmax    = -9999.
      do i = 1, ncalx
          xdt(i) = float(i)
          qtot(i) = qtot(i-1)+qq(i)
          if( qq(i).ge.qmax ) then
              qmax = qq(i)
              iqmx = i
          end if
      end do
c
      q25x = qmax*0.25
      iq25 = 0
      do i = 1, iqmx
          if( qq(i).gt.q25x ) then
              iq25 = i-1
              go to 100
          end if
      end do
100  if( iq25.eq.0 ) iq25 = iqmx
      jq25 = 0
      do i = ncalx, iqmx, -1
          if( qq(i).gt.q25x ) then
              jq25 = i+1
              go to 200
          end if
      end do
200  if( jq25.ge.ncalx ) jq25 = iqmx
c
      rmin = 999.
      its = 0
      do i = 2, iq25-1
          call sokanl( xdt, qtot, 1, i, r1 )
          call sokanl( xdt, qtot, i, iq25, r2 )
          r = (abs(1.-r1)+abs(1.-r2))*0.5
          if( r.lt.rmin ) then
              rmin = r
              its = i
          end if
      end do
      if( its.eq.0 ) its = 1
c
      rmin = 999.
      ite = 0
      do i = jq25+1, ncalx-1
          if( i.eq.140 ) then

```

```

end if
  call sokanl( xdt, qtot, i, ncalx, r1 )
  call sokanl( xdt, qtot, jq25, i, r2 )
  r = (abs(1.-r1)+abs(1.-r2))*0.5
  if( r. lt. rmin ) then
    rmin = r
    ite = i
  end if
end do
if( ite. eq. 0 ) ite = ncalx
c
ts = 1.0
itsa = iqmx-its
if( itsa. gt. 0 ) then
  a = (qmax-qq(its))/float(itsa)
  b = qq(its)-a*float(its)
  if( nint(a*1000.). eq. 0 ) then
    ts = 1.
  else
    ts = -b/a
    if( ts. le. 0. ) ts = 1.
  end if
  ta = float(iqmx)-ts
else
  ta = 1.0
end if
itsa = iqmx-ite
if( itsa. lt. 0 ) then
  a = (qmax-qq(ite))/float(itsa)
  b = qq(ite)-a*float(ite)
  if( nint(a*1000.). eq. 0 ) then
    te = ncalx
  else
    te = -b/a
  end if
  tr = te-ts
  if( tr. ge. ncalx ) tr = ncalx
else
  tr = ncalx
end if
c
return
end
c
=====
subroutine sokanl( x, y, n1, n2, r )
c=====
dimension  x(1), y(1), xl(720), yl(720)
c
m = 0
if( n1. lt. n2 ) then
  n3 = 1
else
  n3 = -1
end if
do i = n1, n2, n3
  if(x(i). ge. 0.0001 .and. y(i). ge. 0.0001) then
    m = m+1

```

```

        xl(m) = x(i)
        yl(m) = y(i)
    end if
end do
if( m. le. 1 ) then
    r = 0.0
else
c
    sumxy = 0.0
    sumx  = 0.0
    sumy  = 0.0
    sumx2 = 0.0
    sumy2 = 0.0
    do i = 1, m
        sumxy = sumxy+xl(i)*yl(i)
        sumx  = sumx +xl(i)
        sumy  = sumy +yl(i)
        sumx2 = sumx2+xl(i)*xl(i)
        sumy2 = sumy2+yl(i)*yl(i)
    end do
c
    zm  =float(m)
    buns = sumxy-sumx*sumy/zm
    bunbo = (sumx2-sumx*sumx/zm)*(sumy2-sumy*sumy/zm)
    if( bunbo. gt. 0.0 ) then
        r  = buns/bunbo
    else
        r  = 1.0
    end if
end if
c
return
end
c
c=====
c      subroutine  kpxcal( amm, ta, tr, ak1, ak2, ap1, ap2 )
c=====
c ***** 河道追跡貯留関数モデル定数算定の近似式
c
dimension  a(0:3), b(0:4, 4), c(0:4, 4), d(0:4, 4), e(0:4, 4)
data b /   0.8545,  -4.8430,  33.8238,  -72.2531,  49.7646
1         ,   0.2404,  -1.1777,   8.0276,  -16.2876,   9.8026
2         ,   0.5844,   0.9107,  -32.9092,  96.5526,  -80.2872
3         ,  -0.3999,  10.4083,  -47.3310,  71.7788,  -31.9668 /
data c /   0.5229,  22.7545, -154.6917,  327.8772, -224.9687
1         ,   1.0848,  23.9708, -126.7464,  230.5159, -124.7640
2         ,  -0.9989, -18.7643,  216.3786, -546.4481,  422.4272
3         ,   1.3325, -34.6802,  151.6250, -223.8849,  95.8506 /
data d /  -0.8426, -39.5757,  249.2689, -512.7692,  345.5463
1         ,   0.1929, -19.1466,  90.2282, -157.3522,  79.5772
2         ,   2.7263,  53.1813, -427.1458,  976.2463, -713.3760
3         ,  -0.7202,  24.5035, -98.6764,  138.5036, -52.1388 /
data e /   0.2763,  22.4122, -130.0060,  259.4679, -171.6907
1         ,   0.0000,   0.0000,   0.0000,   0.0000,   0.0000
2         ,  -1.1199, -37.4282,  249.4726, -535.5305,  376.2721
3         ,   0.0000,   0.0000,   0.0000,   0.0000,   0.0000 /
c
tar = ta/tr

```

## 14-1programl. f. txt

```

c
call a03cal( a(0), b(0,1), c(0,1), d(0,1), e(0,1), tar )
ak1 = a(0)+a(1)*amm+a(2)*amm**2. +a(3)*amm**3.
c
call a03cal( a(0), b(0,2), c(0,2), d(0,2), e(0,2), tar )
ak2 = a(0)*exp(a(1)*amm+a(2)*amm**2.)
c
call a03cal( a(0), b(0,3), c(0,3), d(0,3), e(0,3), tar )
ap1 = a(0)+a(1)*amm+a(2)*amm**2. +a(3)*amm**3.
c
call a03cal( a(0), b(0,4), c(0,4), d(0,4), e(0,4), tar )
ap2 = a(0)+a(1)*amm+a(2)*amm**2.
c
return
end
c
=====
c
subroutine a03cal( a, b, c, d, e, tar )
c
=====
c
implicit double precision (a-h, o-z)
dimension a(0:3), b(0:4), c(0:4), d(0:4), e(0:4)
c
a(0) = b(0)+b(1)*tar+b(2)*tar**2. +b(3)*tar**3. +b(4)*tar**4.
a(1) = c(0)+c(1)*tar+c(2)*tar**2. +c(3)*tar**3. +c(4)*tar**4.
a(2) = d(0)+d(1)*tar+d(2)*tar**2. +d(3)*tar**3. +d(4)*tar**4.
a(3) = e(0)+e(1)*tar+e(2)*tar**2. +e(3)*tar**3. +e(4)*tar**4.
c
return
end

```

湧別川  
湧別川  
丸瀬布

2001年9月12日洪水  
2 1 10 50 0.0010  
3.000  
93 93

2001 9 10 20  
6 2 0 0  
0 0 1 0 10 11 10 0

河道A

1 20500.0 1.3834 0.6765

河道B

2 7300.0 1.5532 0.6642

A R E A 1

1 130.17

1.6399 2.2037 2.2805 5.1248 4.7661 4.5611 4.5611 2.5624 3.5617  
3.8436  
3.4849 3.5617 3.8436 3.2030 1.9218 2.5624 2.9211 4.4073 4.1255  
3.8436  
0.6406 0.6406 0.6406 1.2812 1.2812 1.9218 0.6406 1.9218 0.9993  
0.6406  
0.6406 0.3587 0.0000 0.6406 0.3587 0.9225 0.9993 0.3587 0.6406  
0.6406  
0.3587 0.6406 0.2819 0.3587 0.0000 0.0000 0.4400 0.0000 0.0000  
0.0000  
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000  
0.0000  
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000  
0.0000  
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000  
0.0000  
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000  
0.0000  
0.0000 0.0000 0.0000

A R E A 2

2 143.80

1.3709 2.4727 2.0115 5.1248 5.0351 4.0230 4.0230 2.5624 3.2927  
3.8436  
3.7539 3.2927 3.8436 3.2030 1.9218 2.5624 2.6521 4.9454 4.3945  
3.8436  
0.6406 0.6406 0.6406 1.2812 1.2812 1.9218 0.6406 1.9218 0.7303  
0.6406  
0.6406 0.0897 0.0000 0.6406 0.0897 1.1915 0.7303 0.0897 0.6406  
0.6406  
0.0897 0.6406 0.5509 0.0897 0.0000 0.0000 0.8600 0.0000 0.0000  
0.0000  
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000  
0.0000  
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000  
0.0000  
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000  
0.0000  
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000  
0.0000  
0.0000 0.0000 0.0000

A R E A 3



## input3. txt

3	82.94								
1.4285	1.6848	2.2485	5.3170	4.6508	4.5995	4.6508	2.8379	3.5297	
3.5810									
2.9788	3.7667	3.6514	3.1325	2.2229	2.5624	3.4272	3.8884	3.8884	
4.2664									
0.9097	0.8328	0.7175	1.5054	1.5823	1.9218	0.9097	1.6848	1.5182	
0.6406									
0.8648	0.6406	0.2691	0.8328	0.6406	0.9545	1.4285	0.8776	0.8648	
0.9097									
0.7943	0.6406	0.2691	0.6406	0.2242	0.0000	0.4200	0.0000	0.0700	
0.0000									
0.3000	0.0500	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
0.0000									
0.0000	0.0000	0.3000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
0.0000									
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
0.0000									
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
0.0000									
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
0.0000									
A R E A 4									
4	280.31								
1.3773	1.8129	1.6976	5.1248	5.0287	3.3952	3.4016	3.2094	2.9788	
3.8436									
3.4208	3.3055	3.8436	3.2030	2.2485	2.5624	2.9852	4.5995	4.3817	
4.1703									
0.9673	0.6406	0.9609	1.2812	1.6079	1.9218	0.9673	1.5951	1.3837	
0.6406									
0.6406	0.4228	0.3267	0.6406	0.4228	1.5054	0.7431	0.7431	0.6406	
0.9673									
0.4228	0.6406	0.5445	0.4228	0.0000	0.0000	0.8500	0.0000	0.5000	
0.0000									
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
0.0000									
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
0.0000									
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
0.0000									
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
0.0000									
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
0.0000									
A R E A 5									
5	44.18								
0.8584	1.3260	1.7040	5.5092	4.9967	3.5425	3.7475	3.3760	2.9852	
3.3760									
2.7802	3.5874	3.4656	3.1133	2.6072	2.5624	3.6258	4.0614	4.0614	
4.8237									
1.2812	1.0186	0.8968	1.7040	1.9666	1.9218	1.2812	1.3260	1.8770	
0.6406									
1.0634	0.6406	0.6406	1.0186	0.6406	1.4990	1.4478	1.2364	1.0634	
1.2812									
0.9801	0.6406	0.6406	0.6406	0.4228	0.0000	1.0000	0.0000	0.3400	
0.0000									
0.6000	0.0700	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
0.0000									
0.0000	0.0000	0.6000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
0.0000									
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
0.0000									

input3. txt

0. 0000  
 0. 0000 0. 0000 0. 0000 0. 0000 0. 0000 0. 0000 0. 0000 0. 0000 0. 0000  
 0. 0000  
 0. 0000 0. 0000 0. 0000  
 A R E A 6  
 6 120. 60  
 1. 1531 1. 7873 2. 1780 5. 2593 4. 1127 5. 1120 4. 2344 2. 6969 3. 4592  
 3. 2030  
 2. 8186 3. 5938 3. 7091 2. 6969 2. 5624 2. 5624 3. 5874 3. 8436 3. 8436  
 4. 1127  
 1. 0250 0. 7751 0. 8904 1. 6656 1. 9218 1. 9218 1. 0250 1. 7873 1. 4157  
 0. 6406  
 1. 0250 0. 6406 0. 3844 0. 7751 0. 6406 1. 0250 1. 4157 0. 7751 1. 0250  
 1. 0250  
 0. 5189 0. 6406 0. 3844 0. 6406 0. 3844 0. 0000 0. 6000 0. 0000 0. 0000  
 0. 0000  
 0. 2100 0. 3900 0. 0000 0. 0000 0. 0000 0. 0000 0. 0000 0. 0000 0. 0000  
 0. 0000  
 0. 0000 0. 0000 0. 2100 0. 0000 0. 0000 0. 0000 0. 0000 0. 0000 0. 0000  
 0. 0000  
 0. 0000 0. 0000 0. 0000 0. 0000 0. 0000 0. 0000 0. 0000 0. 0000 0. 0000  
 0. 0000  
 0. 0000 0. 0000 0. 0000 0. 0000 0. 0000 0. 0000 0. 0000 0. 0000 0. 0000  
 0. 0000  
 0. 0000 0. 0000 0. 0000  
 丸瀬布地点流域平均有効雨量  
 1. 361771 1. 950706 1. 978062 5. 186077 4. 808627 4. 087803 3. 974663  
 2. 882071  
 3. 259262 3. 694348 3. 319379 3. 451363 3. 782676 3. 114672 2. 201214  
 2. 562400  
 3. 086621 4. 413493 4. 192839 4. 095963 0. 875699 0. 701524 0. 812183  
 1. 385475  
 1. 560614 1. 921800 0. 875699 1. 730052 1. 250055 0. 640600 0. 744875  
 0. 419946  
 0. 235099 0. 701524 0. 419946 1. 224938 0. 993238 0. 609455 0. 744875  
 0. 875699  
 0. 436246 0. 640600 0. 455753 0. 419946 0. 104275 0. 000000 0. 711448  
 0. 000000  
 0. 200726 0. 000000 0. 095656 0. 067673 0. 000000 0. 000000 0. 000000  
 0. 000000  
 0. 000000 0. 000000 0. 000000 0. 000000 0. 000000 0. 000000 0. 095656  
 0. 000000  
 0. 000000 0. 000000 0. 000000 0. 000000 0. 000000 0. 000000 0. 000000  
 0. 000000  
 0. 000000 0. 000000 0. 000000 0. 000000 0. 000000 0. 000000 0. 000000  
 0. 000000  
 0. 000000 0. 000000 0. 000000 0. 000000 0. 000000 0. 000000 0. 000000  
 0. 000000  
 0. 000000 0. 000000 0. 000000 0. 000000 0. 000000  
 丸瀬布地点直接流出高  
 0. 003411 0. 016374 0. 023523 0. 129497 0. 230115 0. 385174 0. 590781  
 0. 829241  
 0. 960343 1. 071099 1. 141725 1. 191243 1. 265854 1. 401141 1. 552027  
 1. 738390  
 1. 857268 1. 920897 2. 025711 2. 154036 2. 350736 2. 589758 2. 642277  
 2. 745701  
 2. 658581 2. 572659 2. 476555 2. 315424 2. 180500 2. 080521 1. 941551  
 1. 886504

input3.txt

1. 792466	1. 671622	1. 610329	1. 549916	1. 472195	1. 404925	1. 330138
1. 291139						
1. 269595	1. 231183	1. 209932	1. 155533	1. 077776	1. 049374	0. 974257
0. 985421						
0. 957754	0. 922666	0. 858049	0. 883587	0. 769014	0. 729123	0. 717712
0. 657983						
0. 613118	0. 628866	0. 577852	0. 586670	0. 595585	0. 551600	0. 560320
0. 542833						
0. 518979	0. 508154	0. 459365	0. 448736	0. 419614	0. 378855	0. 368495
0. 376090						
0. 336065	0. 302652	0. 321444	0. 299643	0. 266719	0. 296658	0. 246708
0. 236739						
0. 237942	0. 194971	0. 169075	0. 170033	0. 144431	0. 134804	0. 109691
0. 065021						
0. 080586	0. 056255	0. 037168	0. 018300	0. 004383		

14-2programl. for. txt

```

C *****
C 第14回ゼミ
C 複合流域における貯留関数法の最適化
C 損失項を含む貯留関数法
C
C 作成者 星 清、松木賢治          作成日 2005. 7. 4
C *****
C
C 分割流域流出計算手法：損失項を含む貯留関数法
C (1段タンク型非線形貯留関数モデル) のパラメータ同定
C 支川に分割流域と河道追跡を含む計算を考慮
C 河道追跡は無次元領域にて計算した後、解を実領域に変換
C
C 上流端条件を考慮した河道追跡が可能
C (subroutine inputのnindで制御)
C
C***   モデル定数 c11, c12及び c13の同時最適化   ***
C
C   一階ニュートン法 (newton method)
C   ヤコビアン (jacobian) 行列 (一次微係数使用)
C*****
C
C   storage-discharge relationship
C    $s = k11*q^{p1} + k12*d(q^{p2})/dt$ 
C    $ds/dt = r - q - b + q0$ 
C    $q0(t) = qb * \exp(-ram*t)$           !基底流出量
C   qb = 初期流出高
C   ram = ハイドログラフ低減部の減衰係数
C   b = k13*q (b = 損失高)
C   k11 = c11*A**(0.24)
C   k12 = c12*k11**2*(rave)**(-0.2648)
C   c13 = 1 + k13
C   p1 = 0.6 and p2 = 0.4648
C
C   r= 観測雨量
C   q= 観測流量
C*****
C   optimization of parameters c11, c12 and c13
C*****
C
C   -----
C   character gp*128, chitenr*128, chiten1*128, chitenl*128
C
C   dimension q(200), pas(200,9), dpa(9), x(9), u(18)
C   dimension er(200), z(9), uut(18), ut(18,200,20)
C   dimension co(9), rave(50), qb(50), qb0(50)
C   dimension qc(200), uua(3,200), qqt(200), qinlt(200,20)
C   dimension dlqt(5,200,20), dlut(5,18,200,20), dlqbt(5)
C
C   common/st1/wk1, wk2, wp1, wp2, wc3, rain, q0in
C   common/st2/n, m, m1, m2, h, h2, h3, h4
C   common/date/iy, im, id, ih, nq, nr, nban
C   common/discha/r(50,200), q0(200), qt(200,20), q00(50,200), area(50)
C   common/st3/nl, kount, p, ncal, nind, ndat
C   common/st4/cc1, cc2, cc3, cp1, cp2, areatot, trave, ram, areaj
C   common/st5/jdg(90), chitenl(10), cleg(40), alph(40), cmm(40)
C   common/st6/qinl(200), chitenr(20), chiten1
C

```

14-2program1. for. txt

```

C
open(1, file=' input. dat', status=' unknown')
open(5, file=' 河道追跡. out', status=' unknown')
C
C
write(5, *) ' 分割流域流出・河道追跡を含む複合流域解析'
write(5, *) ' 損失項を含む貯留関数法'
write(5, *) ' 1段タンク型非線形貯留関数モデルの最適化'
write(5, *) ' 一階ニュートン法(一次微係数使用)'
write(5, *) ' モデル定数 c11, c12及びc13の同時最適化'
write(5, ' ()')
C
C
データ入力
C
*****
call input
C
*****
分割流域別初期流量及び基底流出成分の計算
areatot=0.0
do j=1, nban
areatot=areatot+area(j)
end do
areatot=areatot+areaj !上流端流域面積を追加
do j=1, nban
qb0(j)=q0(1)*area(j)/areatot
qb(j)=3.6*qb0(j)/area(j)
C
地下水流出成分の低減係数(ram : 入力データ)
q01=qb(j)
do i=1, nq
q02=qb(j)*exp(-ram*i)
q00(j, i)=(q01+q02)*0.5 !基底流出量

q01=q02
end do
end do
C
C
基準点流量を流出高変換(mm/h)
do 1001 i=1, nq
1001 q(i)=3.6*q0(i)/areatot
C
分割流域における平均雨量強度の計算
do j=1, nban
rtot=0.0
nrt=0
do i=1, nr
if(r(j, i).gt.0.0) then
rtot=rtot+r(j, i)
nrt=nrt+1
end if
end do
rave(j)=rtot/float(nrt)
end do
C
C
基準点での平均流域雨量強度
trave = 基準点平均雨量強度
nzero=0
rtot=0.0
do 199 i=1, nr
if(r(nban+1, i).eq.0.0) go to 199

```

## 14-2program1. for. txt

```

nzero=nzero+1
rtot=rtot+r (nban+1, i)
199 continue
xnz=nzero
trave=rtot/xnz
C
write(5, ' ( ) ')
write(5, 210)
210 format(' 流域面積', ' 平均雨量強度', ' c11初期値', ' c12初期値',
& ' c13初期値')
write(5, ' (2f10. 2, 3f13. 3)') areatot, trave, cc1, cc2, cc3
C
C 規準化パラメータの設定
yc1=1. 0
yc2=1. 0
yc3=1. 0
C モデル定数の初期値
co(1)=cc1
co(2)=cc2
co(3)=cc3
C
C
write(5, ' ( ) ')
write(5, ' ( ) ')
gp=' No c11 c12 c13' //
& ' rmse kai2'
write(5, ' (a80)') gp
C
dt=3600. 0
xnl=n1
m1=m+1
m2=2*m
xnq=nq
fac=0. 0
C
C 上流端河道流入量を分割数(n1)ごとに設定
C
do i=1, nq
do k=1, n1
if(i. eq. 1) then
qinlt(i, k)=qinl(i)
else
qinlt(i, k)=qinl(i-1)+float(k)*(qinl(i)-qinl(i-1))/xnl
end if
end do
end do
C
C 最適化開始 (一階ニュートン法)
C
do 999 kkll=1, kount
C
ibn=0
icn=0
do ll=1, nq
do i=1, n1
if(nind. eq. 0) then
qt(ll, i)=0. 0
else

```

```

qt(11, i)=qinlt(11, i)
end if
ut(1, 11, i)=0.0
ut(2, 11, i)=0.0
ut(3, 11, i)=0.0
end do
c
qqt(11)=qt(11, n1)
c
end do
qbt=0.0
  iscl=0
  jscl=0
c
do 900 n1=1, ndat
c
jdg=1 ; 河道追跡計算、jdg=0 ; 分割流域流出計算
jjdg=jjdg(n1)
kscl=jjdg/10
c
  支川合流判断
  if(jscl. eq. 0. and. kscl. eq. 0) go to 906
  if(kscl. ne. 0) go to 907
c
  本川へ合流、保存量（上流端流量）全てを合計して下流へ
  do kkl=1, jscl
    do 11=1, nq
      do i=1, n1
        qt(11, i)=qt(11, i)+dlqt(kkl, 11, i)
        do j=1, 3
          ut(j, 11, i)=ut(j, 11, i)+dlut(kkl, j, 11, i)
        end do
      end do
      qqt(11)=qqt(11)+dlqt(kkl, 11, n1)
    end do
    qbt=qbt+dlqbt(kkl)
  end do
  jscl=0
  iscl=0
c ** 支川合流点で下流に河道も流域もない場合
  if(jjdg. eq. 2) go to 905
  go to 906
907 if(kscl-iscl) 902, 906, 904
c
  支川合流終了、同じ支川に分割流域・河道が結合、
c
  支川から新たな支川発生
902 if(jscl. le. 1) stop
c
  合流支川部に保存量（上流端流量）を加える
  do 11=1, nq
    do i=1, n1
      qt(11, i)=qt(11, i)+dlqt(jscl, 11, i)
      do j=1, 3
        ut(j, 11, i)=ut(j, 11, i)+dlut(jscl, j, 11, i)
      end do
    end do
    qqt(11)=qqt(11)+dlqt(jscl, 11, n1)
  end do
  qbt=qbt+dlqbt(jscl)
  jscl=jscl-1
  iscl=kscl
  go to 906
904 jscl=jscl+1

```

```

                                14-2program1. for. txt
c      新たな支川が発生、ランク (jscl) を加え、
c      支川番号 (iscl) を保存する
c      今まで累加した量を保存し、流出量等をクリアする
iscl=kscl
do ll=1, nq
  do i=1, nl
    dlqt(jscl, ll, i)=qt(ll, i)
    qt(ll, i)=0.
    do j=1, 3
      dlut(jscl, j, ll, i)=ut(j, ll, i)
      ut(j, ll, i)=0.
    end do
  end do
  dlqt(jscl, ll, nl)=qqt(ll)
  qqt(ll)=0.
end do
dlqbt(jscl)=qbt
qbt=0.
906 jjdg=jjdg-kscl*10
c      分割流域か河道かの判断
c
      if(jjdg. eq. 1) go to 901
c
      分割流域における流出計算.....
c      ibn: 河道番号
c
      ibn=ibn+1
c      分割流域での流出計算時間間隔
      h=1. /xn1
      h2=h*h
      h3=h2*h
      h4=h3*h
c***      モデル定数の更新
c
      表面・中間流出成分の定数
c      貯留関数モデル定数 k11, k12, p1, p2の設定
      wk1=cc1*yc1*area(ibn)**0.24
      zk2=cc2*yc2*wk1**2*rave(ibn)**(-0.2648)
      wk2=1. /zk2
      wp1=cp1
      zp2=cp2
      wp2=1. /zp2
      wc3=cc3*yc3
c
      流量と感度係数の初期値設定
c      do 800 i=1, n
800 x(i)=0.0
      x(1)=qb(ibn)**zp2
      do 802 i=1, m2
802 u(i)=0.0
c
      do 30 ll=1, nq
      rain=0.0
      if(ll. le. nr) rain=r(ibn, ll)
      q0in=q00(ibn, ll)
      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
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.)

C
C      流域流出量の累計（次の河道への流入量）
C      流域流出量の感度係数ベクトルの累計（次の河道へ伝達）
C
qt(11,k)=qt(11,k)+qcc*area(ibn)/3.6
ut(1,11,k)=ut(1,11,k)+area(ibn)/3.6*cc*u(1)*area(ibn)**0.24
ut(2,11,k)=ut(2,11,k)+area(ibn)/3.6*cc*u(2)*wk1**2
& *rave(ibn)**(-0.2648)
ut(3,11,k)=ut(3,11,k)+area(ibn)/3.6*cc*u(3)
40 continue

C
qc(11)=qcc
qqt(11)=qt(11,n1)
qbt=qbt+qb0(ibn)
C      表面・中間流出成分の感度係数
uua(1,11)=cc*u(1)*area(ibn)**0.24
uua(2,11)=cc*u(2)*wk1**2*rave(ibn)**(-0.2648)
uua(3,11)=cc*u(3)

C
C      qt:分割流域からの流出量（次の河道流入量）（累計）（計算時間間隔毎）
C      qqt:分割流域からの流出量（次の河道流入量）（累計）
C      （1時間毎）チェック用
C      ut:分割流域の感度係数（累計）（計算時間間隔毎）
C      qbt:流域毎の初期流量の合計
C      uua:感度係数ベクトル（1時間毎）チェック用
C
30 continue

C
go to 905
901 continue

C
C      河道追跡計算（無次元領域）.....
icn=icn+1

C
C
C      ta, trの抽出.....
call tatrst( nq, qqt, ts, ta, tr )
C      ta/tr=0.5と仮定しても実用上問題なし
ta = tr*0.5

C
C      河道追跡用貯留関数モデル定数 k3, k4, p3, p4を計算....
amm=cmm(icn)
C      無次元領域での貯留関数モデル定数 k3, k4, p3, p4
C
call kpxcal( amm, ta, tr, wk1, wk02, wp1, wp02 )

C
totq = 0.0
ncalgt = 0
do i = nint(ts), nint(ts+tr)
totq = totq+qqt(i)
ncalgt = ncalgt+1

```

## 14-2program1. for. txt

```

end do
C
C      上流端平均流入量 (qm)
qm = totq/float(ncalgt)
C
C      時間規準化 (t* = ttt)
ttt = alph(icn)*cleg(icn)*qm**(cmm(icn)-1.0)
C
C      河道追跡用貯留関数モデル定数の保存
wk3=wk1
zk4=wk02
wp3=wp1
zp4=wp02
C      河道追跡における無次元計算時間間隔 (h)
h=dt/ttt/xn1
h2=h*h
h3=h2*h
h4=h3*h
C
C      河道追跡計算のための貯留関数モデル定数の変数変換
wk1=wk3
zk2=zk4
wk2=1./zk2
wp1=wp3
zp2=zp4
wp2=1./zp2
C      河道流量と感度係数の初期値設定
do 804 i=1, n
804 x(i)=0.0
x(1)=(qt(1,1)/qm)**zp2
do 806 i=1, m2
806 u(i)=0.0
C
do 31 ll=1, nq
do 41 k=1, nl
rain=qt(ll,k)/qm
C      無次元量へ変換
C      強制項 (=上流端流入量に関する感度係数)
uut(1)=ut(1,ll,k)/qm
uut(2)=ut(2,ll,k)/qm
uut(3)=ut(3,ll,k)/qm
C** solution of sensitivity equation
call gesto2(x,u,uut,1)
C** solution of differential equation
call gesto2(x,u,uut,2)
if(x(1).le.0.0) x(1)=0.0
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.)
C      実領域での流出量と感度係数へ変換
qt(ll,k)=qcc*qm
ut(1,ll,k)=cc*u(1)*qm
ut(2,ll,k)=cc*u(2)*qm
ut(3,ll,k)=cc*u(3)*qm
41 continue
C
qqt(ll)=qt(ll,nl)

```

```

31 continue
C
905 continue
C
900 continue
C
C 目的関数(sum):カイ2乗(skai2):相対誤差(sotai)
C sum=0.0
C skai2=0.0
C sotai=0.0
C
C do 32 ll=1, nq
C qq=q0(ll)
C qs=0.0
C if(qq.gt.0.0) qs=sqrt(qq)
C 誤差項
C err=qq-qt(ll, nl)
C er(ll)=err
C if(qq.eq.0.0) go to 49
C erw=(err/qs)*(3.6/areatot)**0.5
C skai2=skai2+erw**2
C sotai=sotai+abs(err)/qq
49 continue
C sum=sum+(err*3.6/areatot)**2
C
C 1時間毎の感度係数ベクトル抽出
C** sensitivity coefficients
C do 42 i=1, m
42 pas(ll, i)=ut(i, ll, nl)
C 感度係数の規準化
C do 43 i=1, m
43 pas(ll, i)=co(i)*pas(ll, i)
C
C** error between observed and computed discharges
C pas(ll, m1)=err
32 continue
C
C sum=sqrt(sum/xnq) !rmse
C skai2=skai2/xnq !kai2
C
C write(5, '(i5, 3f8.3, 3f10.3)') kkl1, cc1*yc1, cc2*yc2, wc3,
C & sum, skai2
C
C z(1)=yc1
C z(2)=yc2
C z(3)=yc3
C
C ニュートン法によるモデル定数補正值計算
C (成分回帰分析法の利用)
C** component regression method
C** to solve the correction terms of parameters
C call momreg(9, m1, 200, nq, pas, dpa)
C do 875 i=1, m
C col = abs(dpa(i)/z(i))
C if(col.gt.p) go to 877
875 continue
C go to 879

```

```

877 fac = 0.5*(1.+fac)
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
cc
if(z(3)*cc3.lt.1.0) z(3)=1.0/cc3
cc
yc1=z(1)
yc2=z(2)
yc3=z(3)
999 continue
879 continue
c
c
rtot=0.0
qtot=0.0
qctot=0.0
qsotai=0.0
qsotai2=0.0
qsotai3=0.0
qmax=-999.99
qmaxc=-999.99
do 502 i=1,nq
if(q0(i).gt.qmax) qmax=q0(i) !観測ピーク
if(qt(i,nl).gt.qmaxc) qmaxc=qt(i,nl) !計算ピーク
502 continue
iko=0.0
do 503 i=1,nq
rtot=rtot+r(nban+1,i)
qtot=qtot+q0(i)
qctot=qctot+qt(i,nl)
if(q0(i).lt.0.04*areatot/3.6) go to 500
iko=iko+1
qsotai=qsotai+abs(q0(i)-qt(i,nl))/q0(i)
qsotai2=qsotai2+((q0(i)-qt(i,nl))/q0(i))**2
500 continue
qsotai3=qsotai3+((q0(i)-qt(i,nl))/qmax)**2
503 continue
xiko=iko
qtot0=3.6*qtot/areatot !観測流出高
qctot0=3.6*qctot/areatot !計算流出高
hiryu=qmax/areatot !比流量
peakg=abs(qmax-qmaxc)/qmax !Jpe (ピーク相対誤差)
qsotai=qsotai/xiko !Jre (ハイドログラフ相対誤差)
qsotai2=qsotai2/xiko !Ew
qsotai3=qsotai3/xnq !E
qsotai4=(qtot-qctot)/qtot !Ev
qsotai5=(qmax-qmaxc)/qmax !Ep
c
c--OUTPUT 2
write(5,'()')
write(5,211)
211 format(2x,'収束回数',5x,'c11',9x,'c12',9x,'c13',
&8x,'rmse',8x,'kai2',9x,'Jpe',9x,'Jre')

```

```

14-2program1. for. txt
write(5, '(i7, 8f12. 3)') kkl1, cc1*yc1, cc2*yc2, wc3,
&sum, skai2, peakg, qsotai
write(5, '()')
write(5, 214)
214 format(7x, 'Ew', 9x, 'E', 8x, 'Ev', 8x, 'Ep')
write(5, '(4f10. 3)') qsotai2, qsotai3, qsotai4, qsotai5
write(5, '()')
write(5, 212)
212 format(4x, '比流量', 4x, '総雨量', 2x, '観測総流出', 3x, '計算総流出', 3x,
&'観測ピーク値', 3x, '計算ピーク値')
write(5, '(3f10. 3, 4f15. 3)') hiryu, rtot, qtot0, qctot0, qmax, qmaxc
write(5, '()')
C
write(5, 204)
204 format('      NO', 6x, '実測雨量', 6x, '実測流出高', 6x, '計算流出高')
do i=1, nq
write(5, 207) i, r(nban+1, i), q(i), qt(i, nl)*3. 6/areatot
end do
207 format(i8, f12. 2, 2f16. 4)
C
stop
end
C
-----
C
***** 流域流出解析と河道追跡のためのデータ入力
C
subroutine input
C
-----
character suikei*128, kasen*128, chiten*128, kouzui*128,
& gp*128, chitenr*128, chiten1*128, chitenl*128
C
common/st1/wk1, wk2, wp1, wp2, wc3, rain, q0in
common/st2/n, m, m1, m2, h, h2, h3, h4
common/date/iy, im, id, ih, nq, nr, nban
common/discha/r(50, 200), q0(200), qt(200, 20), q00(50, 200), area(50)
common/st3/nl, kount, p, ncal, nind, ndat
common/st4/cc1, cc2, cc3, cp1, cp2, areatot, trave, ram, areaj
common/st5/jdg(90), chitenl(10), cleg(40), alph(40), cmm(40)
common/st6/qinl(200), chitenr(20), chiten1
C
データの読み込み
C
n: 微分方程式階数 (n=2)   m: モデル定数の数 (m=3)
nl: 1時間の計算分割数 (nl=10)
kount: 最大繰り返し計算回数 (kount=20)   p: 収束条件 (p=0. 001)
nban: 流域数   ncal: 河道数   nc2: 下流に流域・河道がない合流点の数
nind=0: 上流端流入量なし nind=1: 上流端流入量有り
jdg: 流域、河道の計算判別、計算順序---0: 流域   1: 河道
cleg: 河道長   alph:  $\alpha$    cmm: m   area: 流域面積
r: 流域雨量   q0: 基準点流量   qinl: 上流端河道流入量
C
p1=0. 6   p2=0. 4648   ! 流出モデル定数
data cp1, cp2/0. 6, 0. 4648/
C
read(1, 3) suikei      !水系名
read(1, 3) kasen      !河川名
read(1, 3) chiten     !観測所名
read(1, 3) kouzui     !洪水年月日

```

## 14-2program1. for. txt

```

3 format(a128)
4 format(4i5, f8. 0)
5 format(3f8. 0)
  read(1, 4) n, m, nl, kount, p
c   ram = ハイドログラフ低減部の減衰係数
  read(1, '(f8. 0)') ram
c   初期値 : c11, c12, c13
  read(1, 5) cc1, cc2, cc3
c   nr: 雨量データ数  nq: 流量データ数
  read(1, '(2i5)') nr, nq
  read(1, '(4i5)') iy, im, id, ih
  read(1, '(4i5)') nban, ncal, nind, nc2
  ndat=nban+ncal+nc2
  read(1, '(20i5)') (jdg(i), i=1, ndat)
c   河道定数入力
  do j=1, ncal
  read(1, '(a40)') chitenl(j)
  read(1, '(i5, 3f8. 0)') no, cleg(j), alph(j), cmm(j)
  end do
c   nind=1; 上流端流入量がある場合の入力
  if(nind. ge. 1) then
  read(1, '(a40)') gp
  read(1, '(f8. 0)') areaj          !上流端流域面積を追加
  read(1, '(10f8. 0)') (qinl(i), i=1, nq)
  write(5, '()')
  write(5, *)' nind = 1 : 上流端流入量境界条件'
  write(5, '(a40)') gp
  end if
c   分割流域の面積と流域平均雨量
  do j=1, nban
  read(1, '(a40)') chitenr(j)
  read(1, '(i5, f8. 0)') no, area(j)
  read(1, '(10f8. 0)') (r(j, i) , i=1, nr)
  end do
c
c   基準点における流域平均雨量入力
c
c   read(1, '(a40)') gp
  read(1, '(10f8. 0)') (r(nban+1, i) , i=1, nr)
c
c   基準点実測流量の入力(m3/sec)
  read(1, '(a40)') chitenl
  read(1, '(10f8. 0)') (q0(i), i=1, nq)
c
c   入力データ出力
c
  write(5, '( '水系: ', a80)') suihei          !水系
  write(5, '( '河川名: ', a80)') kasen        !河川名
  write(5, '( '観測地点: ', a80)') chiten     !観測地点
  write(5, '( '洪水年月日: ', a80)') kouzui   !洪水年月日
  write(5, '( '洪水開始年月日時: ', 4i5)') iy, im, id, ih
  write(5, '()')
  write(5, '( '流量データ数=' i3, ' ' 計算時間間隔= 1/' , i2)') nq, nl
  write(5, '()')
  write(5, '( 'λ (ramda)= ', f8. 4)') ram
  write(5, '()')
  write(5, '(4i5)') nban, ncal, nind, nc2
  write(5, '()')

```

```

                                14-2programl. for. txt
write(5, '(20i5)') (jdg(i), i=1, ndat)
C
  return
  end
C
C-----
C      *****  分割流域における流出計算と感度係数算定  *****
C
  subroutine gesto(x, u, ijk)
C*****
C
  C      storage-discharge relationship
  C      s = k11*q**p1 + k12*d(q**p2)/dt
  C      ds/dt = r-q-b+q0
  C      q0(t) =qb*exp(-ram*t)      !基底流出量
  C      qb = 初期流出高
  C      ram = ハイドログラフ低減部の減衰係数
  C      b = k13*q (b = 損失高)
  C      k11 = c11*A**(0.24)
  C      k12 = c12*k11**2*(rave)**(-0.2648)
  C      c13 = 1 + k13
  C      p1 = 0.6 and p2 = 0.4648
C
  C      r= 観測雨量
  C      q= 観測流量
C
C**   ijk = 1; solve sensitivity equation
C**   ijk = 2; solve differential equation
C*****
  dimension x(1), u(1), y(9), uu(18), b(9)
  common/st1/wk1, wk2, wp1, wp2, wc3, 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
  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
14 a1=-con*(wp1*wp2-1.)*a*y2-wk2*wp2*c*wc3
  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

```

```

14-2programl. for. txt
g2 = h2*(0.5+a2*h/6.+a3*h2/24.)
g4=f2
if(ijk.eq.2) go to 104
c** solve sensitivity equation
c 感度方程式の強制項計算
b(1)=-wk2*wp1*wp2*d*y2
b(2)=wk2**2*(wk1*wp1*wp2*d*y2 + e*wc3 - 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*wc3*(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-----
c ***** 河道追跡流量と感度係数の算定 *****
c 数値解法は流域流出計算と同じ
c
subroutine gesto2(x, u, uut, 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), uut(1)
common/st1/wk1, wk2, wp1, wp2, wc3, 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
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
14 a1=-con*(wp1*wp2-1.)*a*y2-wk2*wp2*c
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.

```



```

                                14-2program1. for. txt
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
c 強制項 (=上流端流入量に関する感度係数)
b(1)=wk2*uut(1)
b(2)=wk2*uut(2)
b(3)=wk2*uut(3)
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*(wp2-1.)*e+wk2*rain
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-----
c ***** 洪水継続時間とピーク流量生起時刻の算定
c
c subroutine tatrst( ncalx, qq, ts, ta, tr )
c dimension qq(1), qtot(0:720), xdt(720)
c
c qtot(0) = 0.0
c qmax = -9999.
c do i = 1, ncalx
c xdt(i) = float(i)
c qtot(i) = qtot(i-1)+qq(i)
c if( qq(i). ge. qmax ) then
c qmax = qq(i)
c iqmx = i
c end if
c end do
c
c q25x = qmax*0.25
c iq25 = 0
c do i = 1, iqmx
c if( qq(i). gt. q25x ) then
c iq25 = i-1
c go to 100
c end if
c end do
100 if( iq25. eq. 0 ) iq25 = iqmx
c jq25 = 0
c do i = ncalx, iqmx, -1

```

```

14-2program1. for. txt
      if( qq(i).gt.q25x ) then
          jq25 = i+1
          go to 200
      end if
    end do
200   if( jq25. ge. ncalx )  jq25 = iqmx
c
    rmin = 999.
    its = 0
    do i = 2, iq25-1
        call sokanl( xdt, qtot, 1, i, r1 )
        call sokanl( xdt, qtot, i, jq25, r2 )
        r = (abs(1.-r1)+abs(1.-r2))*0.5
        if( r. lt. rmin ) then
            rmin = r
            its = i
        end if
    end do
    if( its. eq. 0 )  its = 1
c
    rmin = 999.
    ite = 0
    do i = jq25+1, ncalx-1
        if( i. eq. 140 ) then
            icv = 0
        end if
        call sokanl( xdt, qtot, i, ncalx, r1 )
        call sokanl( xdt, qtot, jq25, i, r2 )
        r = (abs(1.-r1)+abs(1.-r2))*0.5
        if( r. lt. rmin ) then
            rmin = r
            ite = i
        end if
    end do
    if( ite. eq. 0 )  ite = ncalx
c
    ts = 1.0
    itsa = iqmx-its
    if( itsa. gt. 0 ) then
        a = (qmax-qq(its))/float(itsa)
        b = qq(its)-a*float(its)
        if( nint(a*1000.). eq. 0 ) then
            ts = 1.
        else
            ts = -b/a
            if( ts. le. 0. )  ts = 1.
        end if
        ta = float(iqmx)-ts
    else
        ta = 1.0
    end if
    itsa = iqmx-ite
    if( itsa. lt. 0 ) then
        a = (qmax-qq(ite))/float(itsa)
        b = qq(ite)-a*float(ite)
        if( nint(a*1000.). eq. 0 ) then
            te = ncalx
        else

```

```

        te = -b/a
    end if
    tr = te-ts
    if( tr.ge.ncalx ) tr = ncalx
else
    tr = ncalx
end if
c
return
end
c
c-----
subroutine sokanl(x, y, n1, n2, r)
dimension x(1), y(1), xl(720), yl(720)
c
m = 0
if( n1.lt.n2 ) then
    n3 = 1
else
    n3 = -1
end if
do i = n1, n2, n3
    if(x(i).ge.0.0001 .and. y(i).ge.0.0001) then
        m = m+1
        xl(m) = x(i)
        yl(m) = y(i)
    end if
end do
if( m.le.1 ) then
    a = 1.0
    b = 0.0
    r = 0.0
else
c
    sumxy = 0.0
    sumx = 0.0
    sumy = 0.0
    sumx2 = 0.0
    sumy2 = 0.0
    do i = 1, m
        sumxy = sumxy+xl(i)*yl(i)
        sumx = sumx +xl(i)
        sumy = sumy +yl(i)
        sumx2 = sumx2+xl(i)*xl(i)
        sumy2 = sumy2+yl(i)*yl(i)
    end do
c
    zm =float(m)
    buns1 = sumxy-sumx*sumy/zm
    bunbo = (sumx2-sumx*sumx/zm)*(sumy2-sumy*sumy/zm)
    if( bunbo.gt.0.0 ) then
        r = buns1/bunbo
    else
        r = 1.0
    end if
end if
c
return

```

```

end
c
c-----
c      ***** 河道追跡貯留関数モデル定数算定の近似式
c
c      subroutine kpxcal( amm, ta, tr, ak1, ak2, ap1, ap2 )
c
c      dimension  a(0:3), b(0:4, 4), c(0:4, 4), d(0:4, 4), e(0:4, 4)
c      data b /    0. 8545,  -4. 8430,  33. 8238,  -72. 2531,  49. 7646
c      1          ,    0. 2404,  -1. 1777,   8. 0276,  -16. 2876,   9. 8026
c      2          ,    0. 5844,   0. 9107,  -32. 9092,   96. 5526,  -80. 2872
c      3          ,   -0. 3999,  10. 4083,  -47. 3310,   71. 7788,  -31. 9668 /
c      data c /    0. 5229,  22. 7545, -154. 6917,  327. 8772, -224. 9687
c      1          ,    1. 0848,  23. 9708, -126. 7464,  230. 5159, -124. 7640
c      2          ,   -0. 9989, -18. 7643,  216. 3786, -546. 4481,  422. 4272
c      3          ,    1. 3325, -34. 6802,  151. 6250, -223. 8849,   95. 8506 /
c      data d /   -0. 8426, -39. 5757,  249. 2689, -512. 7692,  345. 5463
c      1          ,    0. 1929, -19. 1466,   90. 2282, -157. 3522,   79. 5772
c      2          ,    2. 7263,  53. 1813, -427. 1458,  976. 2463, -713. 3760
c      3          ,   -0. 7202,  24. 5035, -98. 6764,  138. 5036,  -52. 1388 /
c      data e /    0. 2763,  22. 4122, -130. 0060,  259. 4679, -171. 6907
c      1          ,    0. 0000,   0. 0000,   0. 0000,   0. 0000,   0. 0000
c      2          ,   -1. 1199, -37. 4282,  249. 4726, -535. 5305,  376. 2721
c      3          ,    0. 0000,   0. 0000,   0. 0000,   0. 0000,   0. 0000 /
c
c      tar = ta/tr
c
c      call a03cal( a(0), b(0, 1), c(0, 1), d(0, 1), e(0, 1), tar )
c      ak1 = a(0)+a(1)*amm+a(2)*amm**2. +a(3)*amm**3.
c
c      call a03cal( a(0), b(0, 2), c(0, 2), d(0, 2), e(0, 2), tar )
c      ak2 = a(0)*exp(a(1)*amm+a(2)*amm**2. )
c
c      call a03cal( a(0), b(0, 3), c(0, 3), d(0, 3), e(0, 3), tar )
c      ap1 = a(0)+a(1)*amm+a(2)*amm**2. +a(3)*amm**3.
c
c      call a03cal( a(0), b(0, 4), c(0, 4), d(0, 4), e(0, 4), tar )
c      ap2 = a(0)+a(1)*amm+a(2)*amm**2.
c
c      return
c      end
c
c-----
c      subroutine a03cal( a, b, c, d, e, tar )
c      implicit double precision (a-h, o-z)
c      dimension  a(0:3), b(0:4), c(0:4), d(0:4), e(0:4)
c
c      a(0) = b(0)+b(1)*tar+b(2)*tar**2. +b(3)*tar**3. +b(4)*tar**4.
c      a(1) = c(0)+c(1)*tar+c(2)*tar**2. +c(3)*tar**3. +c(4)*tar**4.
c      a(2) = d(0)+d(1)*tar+d(2)*tar**2. +d(3)*tar**3. +d(4)*tar**4.
c      a(3) = e(0)+e(1)*tar+e(2)*tar**2. +e(3)*tar**3. +e(4)*tar**4.
c
c      return
c      end
c
c-----
c      ***** ニュートン法による最適化手法
c      ***** 成分回帰分析法によるモデル定数補正值算定

```

## 14-2program1. for. txt

```

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)
  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)
  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))
a1=1./u(j, j)
do 5 k=n, j+1, -1
u(k, j)=a1*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
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

```



## input\_1.txt

0.00									
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00									
A R E A 3									
3	82.94								
0.00	0.00	0.00	0.00	0.30	0.18	0.00	1.00	4.05	3.97
3.67									
2.23	2.63	3.51	8.30	7.26	7.18	7.26	4.43	5.51	
5.59									
4.65	5.88	5.70	4.89	3.47	4.00	5.35	6.07	6.07	
6.66									
1.42	1.30	1.12	2.35	2.47	3.00	1.42	2.63	2.37	
1.00									
1.35	1.00	0.42	1.30	1.00	1.49	2.23	1.37	1.35	
1.42									
1.24	1.00	0.42	1.00	0.35	0.00	0.42	0.00	0.07	
0.00									
0.30	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
0.00									
0.00	0.00	0.30	0.00	0.00	0.00	0.00	0.00	0.00	
0.00									
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
0.00									
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
0.00									
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
0.00									
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
0.00									
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
0.00									
A R E A 4									
4	280.31								
0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.66	4.00	2.64
3.32									
2.15	2.83	2.65	8.00	7.85	5.30	5.31	5.01	4.65	
6.00									
5.34	5.16	6.00	5.00	3.51	4.00	4.66	7.18	6.84	
6.51									
1.51	1.00	1.50	2.00	2.51	3.00	1.51	2.49	2.16	
1.00									
1.00	0.66	0.51	1.00	0.66	2.35	1.16	1.16	1.00	
1.51									
0.66	1.00	0.85	0.66	0.00	0.00	0.85	0.00	0.50	
0.00									
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
0.00									
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
0.00									
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
0.00									
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
0.00									
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
0.00									
A R E A 5									
5	44.18								
0.00	0.00	0.00	0.00	0.60	0.47	0.00	1.00	4.07	2.26
1.66									
1.34	2.07	2.66	8.60	7.80	5.53	5.85	5.27	4.66	





	input_1.txt								
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
丸瀬布地点実績流量	9.09	9.09	9.09	8.78	8.78	9.09	9.09	9.40	9.72
10.69	12.43	16.30	18.86	43.45	66.84	102.36	149.14	203.24	233.43
259.08	275.79	287.79	305.39	336.51	371.10	413.59	441.05	456.20	480.53
510.10	554.89	609.12	621.79	645.81	627.38	609.21	588.78	553.86	524.78
503.48	473.50	462.21	442.24	416.29	403.62	391.13	374.80	360.79	345.10
337.39	333.57	325.99	322.23	311.09	294.74	289.39	273.63	277.10	271.91
265.07	251.65	258.31	233.77	225.86	224.29	211.96	202.94	207.43	197.04
199.98	202.94	194.12	197.04	194.12	189.78	188.35	178.45	177.06	171.55
163.45	162.12	164.79	156.84	150.38	155.54	151.66	145.30	152.95	142.80
141.55	142.80	134.20	129.41	130.60	125.87	124.70	120.08	111.11	115.55
111.11	107.83	104.61	102.48	102.48	101.43	98.30	97.27	95.22	93.19
90.20									

湧別川  
湧別川  
開盛

2001年9月12日洪水  
2 3 10 50 0.001

0.014  
10.000 0.150 1.500  
110 110  
2001 9 10 10  
8 5 1 0  
1 0 0 1 0 1 0 10 11 10 1 0 0

河道C  
3 10400 1.6246 0.7030

河道D  
4 10000 1.7880 0.7457

河道E  
5 2700 2.5811 0.7057

河道F  
6 18200 0.6202 0.8894

河道G  
7 3800 1.5965 0.8186

丸瀬布地点実績流量

802.00  
9.09 9.09 9.09 8.78 8.78 9.09 9.09 9.40 9.72  
10.69  
12.43 16.30 18.86 43.45 66.84 102.36 149.14 203.24 233.43  
259.08  
275.79 287.79 305.39 336.51 371.10 413.59 441.05 456.20 480.53  
510.10  
554.89 609.12 621.79 645.81 627.38 609.21 588.78 553.86 524.78  
503.48  
473.50 462.21 442.24 416.29 403.62 391.13 374.80 360.79 345.10  
337.39  
333.57 325.99 322.23 311.09 294.74 289.39 273.63 277.10 271.91  
265.07  
251.65 258.31 233.77 225.86 224.29 211.96 202.94 207.43 197.04  
199.98  
202.94 194.12 197.04 194.12 189.78 188.35 178.45 177.06 171.55  
163.45  
162.12 164.79 156.84 150.38 155.54 151.66 145.30 152.95 142.80  
141.55  
142.80 134.20 129.41 130.60 125.87 124.70 120.08 111.11 115.55  
111.11  
107.83 104.61 102.48 102.48 101.43 98.30 97.27 95.22 93.19  
90.20

A R E A 7

7 71.50  
0.01 0.00 0.00 0.00 1.98 0.00 1.00 4.99 2.02  
1.99  
1.01 2.99 2.99 7.99 5.02 8.97 5.00 4.00 4.99  
4.01  
4.00 5.01 6.00 3.02 4.99 4.01 5.99 6.01 5.99  
6.02  
2.01 1.02 1.99 3.00 4.00 3.00 2.00 3.00 2.00  
1.00  
2.00 1.01 1.01 1.00 1.01 2.00 2.02 1.02 2.01  
1.99  
0.02 1.01 1.01 1.01 1.00 0.01 1.00 0.01 0.00



input\_2.txt

0.00									
AREA 1 0									
10	10.44								
1.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	4.00	4.00
1.00	2.00	2.00	2.00	6.00	7.00	5.00	4.00	4.00	4.00
5.00	4.00	6.00	6.00	5.00	4.00	5.00	5.00	7.00	5.00
8.00	3.00	3.00	1.00	3.00	4.00	3.00	2.00	3.00	2.00
1.00	2.00	2.00	2.00	1.00	3.00	2.00	4.00	3.00	3.00
1.00	2.00	2.00	2.00	2.00	1.00	1.00	1.00	1.00	0.00
1.00	1.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	1.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
AREA 1 1									
11	169.59								
0.00	0.00	0.00	0.00	0.00	0.17	0.00	1.83	6.50	1.83
0.17	2.00	0.34	1.17	3.01	7.17	2.34	3.17	4.34	3.17
6.00	5.83	6.66	6.83	6.66	4.83	4.83	5.00	7.83	8.66
8.66	2.00	1.83	2.00	2.00	3.00	3.83	4.50	2.83	3.83
2.66	1.83	1.83	1.00	1.00	1.00	2.17	1.83	2.83	1.83
1.17	1.00	1.00	1.00	1.00	0.83	0.00	1.00	0.83	0.17
0.00	0.00	0.00	0.83	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
AREA 1 2									
12	110.81								
0.53	0.00	0.00	0.00	0.00	0.00	0.00	1.47	5.41	3.06
0.53	2.00	1.06	1.53	4.12	7.00	3.59	3.53	4.00	3.53
5.47	4.94	6.47	6.47	5.94	4.47	5.00	5.00	7.47	6.88
8.47	2.53	2.53	1.47	2.53	3.53	3.47	3.41	3.00	2.94



input\_2. txt

0.00									
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00									
開盛地点	流域平均雨量(7-14流域)								
0.3276	0.0000	0.0000	0.0374	0.4525	0.0000	1.3735	5.3124	2.6193	
0.8535									
1.7730	1.4426	1.8535	5.1416	6.7287	4.6075	3.9287	4.1812	3.8535	
5.2558									
4.8189	6.1120	6.3361	5.3664	4.5631	4.7011	5.2270	7.1465	6.8648	
7.9210									
2.3276	2.0647	1.6350	2.5546	3.5172	3.3735	3.1228	2.8907	2.8563	
1.7470									
1.9281	1.7011	1.3276	1.0374	1.6525	2.0719	2.6193	2.5101	2.2557	
1.2989									
1.1740	1.3276	1.3276	1.3276	0.9281	0.3276	1.0000	0.7011	0.0719	
0.3276									
0.3650	0.1896	0.7011	0.0000	0.0000	0.0000	0.3276	0.0000	0.3276	
0.0000									
0.0000	0.0000	0.0374	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
0.0000									
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
0.0000									
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
0.0000									
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
0.0000									
開盛地点	実績流量								
48.20	48.20	45.63	45.63	45.63	45.63	45.63	45.63	50.84	
56.33									
62.10	66.77	72.60	78.68	87.59	108.30	118.71	137.68	174.25	
219.24									
291.50	341.30	377.81	419.76	471.49	514.24	571.34	640.27	708.46	
765.50									
824.74	917.74	971.56	1004.58	958.17	958.17	952.57	919.35	886.71	
881.33									
875.96	823.21	797.44	777.13	742.20	722.61	688.95	684.20	665.40	
660.73									
656.09	637.67	619.52	615.02	597.20	579.63	562.33	549.53	532.68	
507.91									
495.75	475.80	464.03	448.56	437.13	418.41	407.38	396.49	392.90	
382.92									
378.60	370.02	365.76	359.43	353.15	344.87	338.72	334.65	330.61	
326.59									
322.59	316.64	312.71	304.92	302.99	299.14	295.32	291.52	287.74	
285.87									
282.13	275.54	268.43	254.49	247.65	234.27	227.72	221.26	214.89	
208.62									
196.35	190.35	184.45	172.93	172.93	167.30	161.77	156.34	150.99	
145.74									

14-3program. for. txt

```

C *****
C 第14回ゼミ
C 複合流域における貯留関数法の最適化
C 地下水流出を含む貯留関数法
C
C 作成者 星 清、松木賢治      作成日 2005. 7. 19
C *****
C
C 分割流域流出計算手法：地下水流出成分を含む貯留関数法
C (2段タンク型非線形貯留関数モデル) のパラメータ同定
C 支川に分割流域と河道追跡を含む計算を考慮
C 河道追跡は無次元領域にて計算した後、解を実領域に変換
C
C 上流端条件を考慮した河道追跡が可能
C (subroutine inputのnindで制御)
C
C*** モデル定数 c11, c12及びc13の同時最適化 ***
C
C 一階ニュートン法 (newton method)
C ヤコビアン (jacobian) 行列 (一次微係数使用)
C*****
C
C*** surface-subsurface runoff component
C  $s1 = k11*q1**p1 + k12*d(q1**p2)/dt$ 
C  $ds1/dt = r - q1 - b$ 
C  $q1 = \text{surface-subsurface runoff component}$ 
C  $b = k13*q1$  (b = 地下水成分への浸透供給量)
C  $k11 = c11*A**(0.24)$ 
C  $k12 = c12*k11**2*(rave)**(-0.2648)$ 
C  $c13 = 1 + k13$ 
C  $p1 = 0.6$  and  $p2 = 0.4648$ 
C
C*** groundwater flow component
C  $s2 = k21*q2 + k22*d(q2)/dt$ 
C  $ds2/dt = b - q2$ 
C  $q2 = \text{groundwater flow component}$ 
C  $k21 = c1*k22$ 
C  $k22 = k13/c0$ 
C  $c0 = (\text{delta}/tc)**2$ 
C  $c1 = \text{delta**2}/tc$ 
C
C***  $q = q1 + q2$  (total runoff)
C
C 地下水成分分離に必要な定数
C
C delta : 減衰係数 ( $\delta > 2$ )
C  $\text{delta}(\delta) = \text{減衰係数}$  (非振動解条件)
C  $\delta = 2.1$  を使用
C tc : 地下水流出成分の分離時定数
C tc : ハイドログラフ低減部の減衰係数 ( $\lambda : \text{ramda}$ ) の逆数
C
C*** 入力条件 ; r= 観測雨量 ; q= 観測流量
C
C*****
C optimization of parameters c11, c12 and c13
C*****
C

```

---

character gp\*128, chitenr\*128, chiten1\*128, chitenl\*128





```

end do
C
C 基準点での平均流域雨量強度
C trave = 基準点平均雨量強度
nzero=0
rtot=0.0
do 199 i=1, nr
if(r(nban+1, i).eq.0.0) go to 199
nzero=nzero+1
rtot=rtot+r(nban+1, i)
199 continue
xnz=nzero
trave=rtot/xnz
C
write(5, ' ( ) ')
write(5, 210)
210 format(' 流域面積', ' 平均雨量強度', ' c11初期値', ' c12初期値',
& ' c13初期値')
write(5, ' (2f10. 2, 3f13. 3)') areatot, trave, cc1, cc2, cc3
C
C
C 規準化パラメータの設定
yc1=1.0
yc2=1.0
yc3=1.0
cc
モデル定数の初期値
co(1)=cc1
co(2)=cc2
co(3)=cc3
C
cc 地下水流出成分（2段目タンク）の定数設定
C c0, c1の計算
c00=(delta/tc)**2
c01=delta**2/tc
C
write(5, ' ( ) ')
write(5, ' ( ) ')
gp=' No c11 c12 c13' //
& ' rmse kai2'
write(5, ' (a80)') gp
C
dt=3600.0
xn1=n1
m1=m+1
m2=2*m
xnq=nq
fac=0.0
C
C 上流端河道流入量を分割数(n1)ごとに設定
C
do i=1, nq
do k=1, n1
if(i.eq.1) then
qinlt(i, k)=qinl(i)
else
qinlt(i, k)=qinl(i-1)+float(k)*(qinl(i)-qinl(i-1))/xn1
end if
end do

```

```

end do
c
c      最適化開始 (一階ニュートン法)
c
do 999 kkl=1, kount
c
  ibn=0
  icn=0
  do ll=1, nq
  do i=1, nl
  if(nind.eq.0) then
  qt(ll, i)=0.0
  else
  qt(ll, i)=qinlt(ll, i)
  end if
  ut(1, ll, i)=0.0
  ut(2, ll, i)=0.0
  ut(3, ll, i)=0.0
  end do

c
  qqt(ll)=qt(ll, nl)
c
  end do
  qbt=0.0
  iscl=0
  jscl=0
c
do 900 nnl=1, ndat
c
  jdj=1 ; 河道追跡計算、jdg=0 ; 分割流域流出計算
  jjdg=jdj(nnl)
  kscl=jjdg/10
c
  支川合流判断
  if(jscl.eq.0.and.kscl.eq.0) go to 906
  if(kscl.ne.0) go to 907
c
  本川へ合流、保存量 (上流端流量) 全てを合計して下流へ
  do kkl=1, jscl
  do ll=1, nq
  do i=1, nl
  qt(ll, i)=qt(ll, i)+dlqt(kkl, ll, i)
  do j=1, 3
  ut(j, ll, i)=ut(j, ll, i)+dlut(kkl, j, ll, i)
  end do
  end do
  qqt(ll)=qqt(ll)+dlqt(kkl, ll, nl)
  end do
  qbt=qbt+dlqbt(kkl)
  end do
  jscl=0
  iscl=0
c
  ** 支川合流点で下流に河道も流域もない場合
  if(jjdg.eq.2) go to 905
  go to 906
907 if(kscl-iscl) 902, 906, 904
c
  支川合流終了、同じ支川に分割流域・河道が結合、
c
  支川から新たな支川発生
902 if(jscl.le.1) stop
c
  合流支川部に保存量 (上流端流量) を加える
  do ll=1, nq

```

14-3program. for. txt

```

do i=1, n1
  qt(11, i)=qt(11, i)+dlqt(jscl, 11, i)
  do j=1, 3
    ut(j, 11, i)=ut(j, 11, i)+dlut(jscl, j, 11, i)
  end do
end do
qqt(11)=qqt(11)+dlqqt(jscl, 11, n1)
end do
qbt=qbt+dlqbt(jscl)
jscl=jscl-1
isc1=kscl
go to 906
904 jscl=jscl+1
c      新たな支川が発生、ランク (jscl) を加え、
c      支川番号 (isc1) を保存する
c      今まで累加した量を保存し、流出量等をクリアする
isc1=kscl
do 11=1, nq
  do i=1, n1
    dlqt(jscl, 11, i)=qt(11, i)
    qt(11, i)=0.
    do j=1, 3
      dlut(jscl, j, 11, i)=ut(j, 11, i)
      ut(j, 11, i)=0.
    end do
  end do
  dlqt(jscl, 11, n1)=qqt(11)
  qqt(11)=0.
end do
dlqbt(jscl)=qbt
qbt=0.
906 jjdg=jjdg-kscl*10
c      分割流域か河道かの判断
c
c      if(jjdg.eq.1) go to 901
c
c      分割流域における流出計算.....
c      ibn:河道番号
c
c      ibn=ibn+1
c      分割流域での流出計算時間間隔
h=1./xnl
h2=h*h
h3=h2*h
h4=h3*h
c
c***   モデル定数の更新
c
c      表面・中間流出成分(1段目タンク)の定数
c      貯留関数モデル定数 k11, k12, p1, p2の設定
wk1=cc1*yc1*area(ibn)**0.24
zk2=cc2*yc2*wk1**2*rave(ibn)**(-0.2648)
wk2=1./zk2
wp1=cp1
zp2=cp2
wp2=1./zp2
wc3=cc3*yc3
c

```

```

c      地下水流出成分(2段目タンク)の貯留係数
zk22=(wc3-1.0)/c00
wk22=1.0/zk22
wk21=c01*zk22

c
c      流量と感度係数の初期値設定
do 800 i=1, 2*n
800  x(i)=0.0
      x(1)=qb(ibn)**zp2
      do 802 i=1, 2*m2
802  u(i)=0.0

c
      do 30 ll=1, nq
      rain=0.0
      if(ll.le.nr) rain=r(ibn, ll)
      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
      if(x(3).le.0.0) x(3)=0.0

c
c      全流出高(計算流出高)
qcc=x(1)**wp2+x(3)
cc=0.0
if(x(1).gt.0.0) cc=wp2*x(1)**(wp2-1.)

c
c      流域流出量の累計(次の河道への流入量)
c      流域流出量の感度係数ベクトルの累計(次の河道へ伝達)
c
      qt(ll, k)=qt(ll, k)+qcc*area(ibn)/3.6
      ut(1, ll, k)=ut(1, ll, k)+area(ibn)/3.6*(cc*u(1)+u(7))
      & *area(ibn)**0.24
      ut(2, ll, k)=ut(2, ll, k)+area(ibn)/3.6*(cc*u(2)+u(8))*wk1**2
      & *rave(ibn)**(-0.2648)
      ut(3, ll, k)=ut(3, ll, k)+area(ibn)/3.6*(cc*u(3)+u(9))
40  continue

c
      qc(ll)=qcc
      qqt(ll)=qt(ll, nl)
      qbt=qbt+qb0(ibn)
c      表面・中間流出成分と地下水流出成分の感度係数を合計
      uua(1, ll)=(cc*u(1)+u(7))*area(ibn)**0.24
      uua(2, ll)=(cc*u(2)+u(8))*wk1**2*rave(ibn)**(-0.2648)
      uua(3, ll)=cc*u(3)+u(9)

c
c      qt: 分割流域からの流出量(次の河道流入量) (累計) (計算時間間隔毎)
c      qqt: 分割流域からの流出量(次の河道流入量) (累計)
c           (1時間毎) チェック用
c      ut: 分割流域の感度係数(累計) (計算時間間隔毎)
c      qbt: 流域毎の初期流量の合計
c      uua: 感度係数ベクトル(1時間毎) チェック用
c
30  continue

c
      go to 905
901  continue

```

14-3program. for. txt

```

C
C 河道追跡計算（無次元領域）.....
icn=icn+1
C
C
C ta, trの抽出.....
call tatrst( nq, qqt, ts, ta, tr )
C ta/tr=0.5と仮定しても実用上問題なし
  ta = tr*0.5
C
C 河道追跡用貯留関数モデル定数 k3, k4, p3, p4を計算....
amm=cmm(icn)
C 無次元領域での貯留関数モデル定数 k3, k4, p3, p4
call kpxcal( amm, ta, tr, wk1, wk02, wp1, wp02 )
C
  totq = 0.0
  ncalgt = 0
  do i = nint(ts), nint(ts+tr)
    totq = totq+qqt(i)
    ncalgt = ncalgt+1
  end do
C
  上流端平均流入量(qm)
  qm = totq/float(ncalgt)
C
  時間規準化(t* = ttt)
  ttt = alph(icn)*cleg(icn)*qm**(cmm(icn)-1.0)
C
C 河道追跡用貯留関数モデル定数の保存
wk3=wk1
zk4=wk02
wp3=wp1
zp4=wp02
C 河道追跡における無次元計算時間間隔(h)
h=dt/ttt/xnl
h2=h*h
h3=h2*h
h4=h3*h
C
C 河道追跡計算のための貯留関数モデル定数の変数変換
wk1=wk3
zk2=zk4
wk2=1./zk2
wp1=wp3
zp2=zp4
wp2=1./zp2
C 河道流量と感度係数の初期値設定
do 804 i=1, n
804 x(i)=0.0
  x(1)=(qt(1,1)/qm)**zp2
do 806 i=1, m2
806 u(i)=0.0
C
  do 31 ll=1, nq
  do 41 k=1, nl
  rain=qt(ll, k)/qm
C 無次元量へ変換
C 強制項（=上流端流入量に関する感度係数）

```

## 14-3program. for. txt

```

      uut(1)=ut(1, ll, k)/qm
      uut(2)=ut(2, ll, k)/qm
      uut(3)=ut(3, ll, k)/qm
c**  solution of sensitivity equation
      call gesto2(x, u, uut, 1)
c**  solution of differential equation
      call gesto2(x, u, uut, 2)
      if(x(1).le.0.0) x(1)=0.0
      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.)
c      実領域での流出量と感度係数へ変換
      qt(ll, k)=qcc*qm
      ut(1, ll, k)=cc*u(1)*qm
      ut(2, ll, k)=cc*u(2)*qm
      ut(3, ll, k)=cc*u(3)*qm
41 continue
c
      qqt(ll)=qt(ll, nl)
31 continue
c
905 continue
c
900 continue
c
      目的関数(sum):カイ2乗(skai2):相対誤差(sotai)
      sum=0.0
      skai2=0.0
      sotai=0.0
c
      do 32 ll=1, nq
      qq=q0(ll)
      qs=0.0
      if(qq.gt.0.0) qs=sqrt(qq)
c      誤差項
      err=qq-qt(ll, nl)
      er(ll)=err
      if(qq.eq.0.0) go to 49
      erw=(err/qs)*(3.6/areatot)**0.5
      skai2=skai2+erw**2
      sotai=sotai+abs(err)/qq
49 continue
      sum=sum+(err*3.6/areatot)**2
c
      1時間毎の感度係数ベクトル抽出
c**  sensitivity coefficients
      do 42 i=1, m
42 pas(ll, i)=ut(i, ll, nl)
c      感度係数の規準化
      do 43 i=1, m
43 pas(ll, i)=co(i)*pas(ll, i)
c
c**  error between observed and computed discharges
      pas(ll, m1)=err
32 continue
c
      sum=sqrt(sum/xnq)      !rmse

```

```

                                14-3program. for. txt
skai2=skai2/xnq                !kai2
C
  write(5, '(i5, 3f8. 3, 2f10. 3)') kkll, cc1*yc1, cc2*yc2, wc3,
& sum, skai2
C
  z(1)=yc1
  z(2)=yc2
  z(3)=yc3
C
  ニュートン法によるモデル定数補正值計算
  (成分回帰分析法の利用)
C** component regression method
C** to solve the correction terms of parameters
  call momreg(9, m1, 200, nq, pas, dpa)
  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)
    do 878 i=1, m
      dpa(i)=fac*dpa(i)
878 continue
    do 991 i=1, m
      xnew=z(i)+dpa(i)
      if(xnew. ge. 0. 001) z(i)=z(i)+dpa(i)
991 continue
CC
  if(z(3)*cc3. lt. 1. 0) z(3)=1. 0/cc3
CC
  yc1=z(1)
  yc2=z(2)
  yc3=z(3)
999 continue
879 continue
C
C
  rtot=0. 0
  qtot=0. 0
  qctot=0. 0
  qsotai=0. 0
  qsotai2=0. 0
  qsotai3=0. 0
  qmax=-999. 99
  qmaxc=-999. 99
  do 502 i=1, nq
    if(q0(i). gt. qmax) qmax=q0(i)      !観測ピーク
    if(qt(i, nl). gt. qmaxc) qmaxc=qt(i, nl) !計算ピーク
502 continue
    iko=0. 0
    do 503 i=1, nq
      rtot=rtot+r(nban+1, i)
      qtot=qtot+q0(i)
      qctot=qctot+qt(i, nl)
      if(q0(i). lt. 0. 04*areatot/3. 6) go to 500
      iko=iko+1
      qsotai=qsotai+abs(q0(i)-qt(i, nl))/q0(i)
      qsotai2=qsotai2+((q0(i)-qt(i, nl))/q0(i))**2

```



## 14-3program. for. txt

```

500 continue
   qsotai3=qsotai3+((q0(i)-qt(i,nl))/qmax)**2
503 continue
   xiko=iko
   qtot0=3.6*qtot/areatot      !観測流出高
   qctot0=3.6*qctot/areatot    !計算流出高
   hiryu=qmax/areatot         !比流量
   peakg=abs(qmax-qmaxc)/qmax  !Jpe (ピーク相対誤差)
   qsotai=qsotai/xiko         !Jre (ハイドログラフ相対誤差)
   qsotai2=qsotai2/xiko       !Ew
   qsotai3=qsotai3/xnq        !E
   qsotai4=(qtot-qctot)/qtot  !Ev
   qsotai5=(qmax-qmaxc)/qmax  !Ep
c
c---OUTPUT 2
   write(5, ' ( ) ')
   write(5, 211)
211 format(2x, '収束回数', 5x, 'c11', 9x, 'c12', 9x, 'c13',
&8x, 'rmse', 8x, 'kai2', 9x, 'Jpe', 9x, 'Jre')
   write(5, ' (i7, 8f12. 3) ') kkl1, cc1*yc1, cc2*yc2, wc3,
&sum, skai2, peakg, qsotai
   write(5, ' ( ) ')
   write(5, 214)
214 format(7x, 'Ew', 9x, 'E', 8x, 'Ev', 8x, 'Ep')
   write(5, ' (4f10. 3) ') qsotai2, qsotai3, qsotai4, qsotai5
   write(5, ' ( ) ')
   write(5, 212)
212 format(4x, '比流量', 4x, '総雨量', 2x, '観測総流出', 3x, '計算総流出', 3x,
&'観測ピーク値', 3x, '計算ピーク値')
   write(5, ' (3f10. 3, 4f15. 3) ') hiryu, rtot, qtot0, qctot0, qmax, qmaxc
   write(5, ' ( ) ')
c
   write(5, 204)
204 format('          NO', 6x, '実測雨量', 6x, '実測流出高', 6x, '計算流出高')
   do i=1, nq
   write(5, 207) i, r(nban+1, i), q(i), qt(i,nl)*3.6/areatot
   end do
207 format(i8, f12. 2, 2f16. 4)
c
   stop
   end
c
c-----
c ***** 流域流出解析と河道追跡のためのデータ入力
c
c   subroutine input
c
c   character suikei*128, kasen*128, chiten*128, kouzui*128,
& gp*128, chitenr*128, chiten1*128, chitenl*128
c
c   common/st1/wk1, wk2, wp1, wp2, wc3, wk21, wk22, rain
c   common/st2/n, m, m1, m2, h, h2, h3, h4
c   common/date/iy, im, id, ih, nq, nr, nban
c   common/discha/r (50, 200), q0(200), qt(200, 20), area(50)
c   common/st3/nl, kount, p, ncal, nind, ndat
c   common/st4/cc1, cc2, cc3, cp1, cp2, delta, tc, areatot, trave, areaj
c   common/st5/jdg(90), chitenl(10), cleg(40), alph(40), cmm(40)
c   common/st6/qinl(200), chitenr(20), chiten1

```

## 14-3program. for. txt

```

c
c   データの読み込み
c
c   n: 微分方程式階数 (n=2)   m: モデル定数の数 (m=3)
c   nl: 1時間の計算分割数 (nl=10)
c   kount: 最大繰り返し計算回数 (kount=20)   p: 収束条件 (p=0.001)
c   nban: 流域数   ncal: 河道数   nc2: 下流に流域・河道がない合流点の数
c   nind=0: 上流端流入量なし nind=1: 上流端流入量有り
c   jdг: 流域、河道の計算判別、計算順序—0: 流域   1: 河道
c   cleg: 河道長   alph:  $\alpha$    cmm: m   area: 流域面積
c   r: 流域雨量   q0: 基準点流量   qinl: 上流端河道流入量
c
c   p1=0.6   p2=0.4648   delta=2.1   ! 流出モデル定数
c   data cp1, cp2, delta/0.6, 0.4648, 2.1/
c
c   read(1,3) suikei   !水系名
c   read(1,3) kasen   !河川名
c   read(1,3) chiten  !観測所名
c   read(1,3) kouzui  !洪水年月日
3  format(a128)
4  format(4i5, 2f8.0)
5  format(3f8.0)
c   read(1,4) n, m, nl, kount, p
c   tc : 流出成分分離時定数
c   tc =ハイドログラフ低減部の減衰係数(ramda)の逆数
c   read(1, '(f8.0)') tc
c   初期値 : c11, c12, c13
c   read(1,5) cc1, cc2, cc3
cc  nr: 雨量データ数   nq: 流量データ数
c   read(1, '(2i5)') nr, nq
c   read(1, '(4i5)') iy, im, id, ih
c   read(1, '(4i5)') nban, ncal, nind, nc2
c   ndat=nban+ncal+nc2
c   read(1, '(20i5)') (jdg(i), i=1, ndat)
c   河道定数入力
c   if(ncal.ge.1) then
c   do j=1, ncal
c   read(1, '(a40)') chitenl(j)
c   read(1, '(i5, 3f8.0)') no, cleg(j), alph(j), cmm(j)
c   end do
c   end if
c   nind=1: 上流端流入量がある場合の入力
c   if(nind.ge.1) then
c   read(1, '(a40)') gp
c   read(1, '(f8.0)') areaj   !上流端流域面積を追加
c   read(1, '(10f8.0)') (qinl(i), i=1, nq)
c   write(5, '()')
c   write(5, *) 'nind = 1 : 上流端流入量境界条件'
c   write(5, '(a40)') gp
c   end if
c   分割流域の面積と流域平均雨量の入力
c   do j=1, nban
c   read(1, '(a40)') chitenr(j)
c   read(1, '(i5, f8.0)') no, area(j)
c   read(1, '(10f8.0)') (r(j, i), i=1, nr)
c   end do
c
c   基準点における流域平均雨量入力

```

## 14-3program. for. txt

```

C
  read(1, '(a40)') gp
  read(1, '(10f8.0)') (r(nban+1, i), i=1, nr)
C
  基準点実測流量の入力(m3/sec)
  read(1, '(a40)') chiten1
  read(1, '(10f8.0)') (q0(i), i=1, nq)
C
  入力データ出力
C
  write(5, ('水系:', a80)) suihei      !水系
  write(5, ('河川名:', a80)) kasen    !河川名
  write(5, ('観測地点:', a80)) chiten !観測地点
  write(5, ('洪水年月日:', a80)) kouzui !洪水年月日
  write(5, ('洪水開始年月日時:', 4i5)) iy, im, id, ih
  write(5, (''))
  write(5, ('流量データ数=', i3, ' 計算時間間隔= 1/', i2)) nq, n1
  write(5, (''))
  write(5, ('分離時定数(Tc) =', f8.2)) tc
  write(5, (''))
  write(5, (4i5)) nban, ncal, nind, nc2
  write(5, (''))
  write(5, (20i5)) (jdg(i), i=1, ndat)
C
  return
  end
C
-----
C
C**  *****  分割流域における流出計算と感度係数算定  *****
C  表面・中間流出成分と地下水流出成分を同時に計算
C
  subroutine gesto(x, u, ijk)
C
C**  ***  surface-subsurface runoff component
  s1 = k11*q1**p1 + k12*d(q1**p2)/dt
  ds1/dt = r-q1-b
  q1 = surface-subsurface runoff component
  b = k13*q1 (b = 地下水成分への浸透供給量)
  k11 = c11*A**(0.24)
  k12 = c12*k11**2*(rave)**(-0.2648)
  c13 = 1 + k13
  p1 = 0.6 and p2 = 0.4648
C
C***  groundwater flow component
  s2 = k21*q2 + k22*d(q2)/dt
  ds2/dt = b - q2
  q2 = groundwater flow component
  k21 = c1*k22
  k22 = k13/c0
  c0 = (delta/tc)**2
  c1 = delta**2/tc
C
C***  q = q1 + q2 (total runoff)
C
  delta( $\delta$ ) = 減衰係数 (非振動解条件)
   $\delta = 2.1$  を使用
  tc = 地下水成分分離時定数
  tc : ハイドログラフ低減部の減衰係数 ( $\lambda$  : ramda) の逆数

```

14-3program. for. txt

```

C
C***  入力条件  ; r= 観測雨量  ; q= 観測流量
C
C**   x(k+1)= phi*x(k) + gamma*b(k)
C**   ijk = 1; solve sensitivity equation
C**   ijk = 2; solve differential equation
C
      dimension x(1), u(1), y(9), uu(18), b(9)
      common/st1/wk1, wk2, wp1, wp2, wc3, wk21, wk22, rain
      common/st2/n, m, m1, m2, h, h2, h3, h4
      common/st7/c00, c01
C
      con=wk1*wk2*wp1*wp2
      wk13=wc3-1.0
      a=0.0
      c=0.0
      d=0.0
      e=0.0
C      1 段目タンクの状態変量
      y1=x(1)
      y2=x(2)
C      2 段目タンクの状態変量
      y3=x(3)
      y4=x(4)
      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
C      1 段目タンクの係数
14  a1=-con*(wp1*wp2-1.)*a*y2-wk2*wp2*c*wc3
      a2=-con*d
      a3=a1+a2**2
      a4=a1+a3
C      2 段目タンクの係数
      a5=-c00/wk13
      a6=-c01
      a7=a5+a6**2
      a8=a5+a7
C**   elements of phi matrix (transition matrix)
C**   1 段タンク目のΦの値
      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
C**   1 段タンク目のγの値
      g2 = h2*(0.5+a2*h/6. +a3*h2/24.)
      g4=f2
C**   elements of phi matrix (transition matrix)
C**   2 段タンク目のΦの値
      f5 = 1. +0.5*a5*h2+a5*a6*h3/6. +a5*a7*h4/24.
      f6 = h*(1. + 0.5*a6*h+a7*h2/6. +a6*a8*h3/24.)
      f7 = a5*f6
      f8 = 1. +a6*h+0.5*a7*h2+a6*a8*h3/6. +

```

```

                                14-3program. for. txt
& (a5*a7+a6**2*a8)*h4/24.
c** elements of gamma matrix
c** 2段タンク目の $\gamma$ の値
g6 = h2*(0.5+a6*h/6. +a7*h2/24.)
g8=f6
c
  if(ijk. eq. 2) go to 104
c
c** solve sensitivity equation
c** 1段目タンクにおける感度係数の計算
c 感度方程式の強制項計算
  b(1)=-wk2*wp1*wp2*d*y2
  b(2)=wk2**2*(wk1*wp1*wp2*d*y2 + e*wc3 - rain)
  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)
c** 2段目タンクにおける感度係数の計算
c 感度方程式の強制項計算
  m3=m2+m
  b(4)=c00*wp2*c*u(1)
  b(5)=c00*wp2*c*u(2)
  b(6)=c00*wp2*c*u(3)+(c00/wk13**2)*y3
  do 17 i=1, m
17 uu(i+m2)= f5*u(i+m2) + f6*u(i+m3) + g6*b(i+m)
  do 19 i=m1, m2
19 uu(i+m2) = f7*u(i+m) + f8*u(i+m2) + g8*b(i)
  do 20 i=1, 2*m2
20 u(i)=uu(i)
c
  return
104 continue
c** solve system equation
c
c** 1段目タンクの状態変量(y1, y2)の計算
b1=con*(wp1*wp2-1.)*d*y2+wk2*wc3*(wp2-1.)*e+wk2*rain
y(1) = f1*x(1)+f2*x(2)+g2*b1
y(2) = f3*x(1)+f4*x(2)+g4*b1
c
c** 2段目タンクの状態変量(y3, y4)の計算
b2=c00*e
y(3) = f5*x(3)+f6*x(4)+g6*b2
y(4) = f7*x(3)+f8*x(4)+g8*b2
do 102 i=1, 2*n
102 x(i)=y(i)
  return
  end
c
c-----
cc ***** 河道追跡流量と感度係数の算定 *****
c 数値解法は流域流出計算と同じ
c
  subroutine gesto2(x, u, uut, ijk)
c
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

```

14-3program.for.txt

```

c**   ijk = 2: solve differential equation
c
      dimension x(1), u(1), y(9), uu(18), b(9), uut(1)
      common/st1/wk1, wk2, wp1, wp2, wc3, wk21, wk22, rain
      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
      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
14   a1=-con*(wp1*wp2-1.)*a*y2-wk2*wp2*c
      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
c       強制項 (=上流端流入量に関する感度係数)
      b(1)=wk2*uut(1)
      b(2)=wk2*uut(2)
      b(3)=wk2*uut(3)
      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*(wp2-1.)*e+wk2*rain
      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-----
c       ***** 洪水継続時間とピーク流量生起時刻の算定

```

## 14-3program. for. txt

```

c
subroutine tatrst( ncalx, qq, ts, ta, tr )
dimension qq(1), qtot(0:720), xdt(720)
c
qtot(0) = 0.0
qmax = -9999.
do i = 1, ncalx
    xdt(i) = float(i)
    qtot(i) = qtot(i-1)+qq(i)
    if( qq(i).ge.qmax ) then
        qmax = qq(i)
        iqmx = i
    end if
end do
c
q25x = qmax*0.25
iq25 = 0
do i = 1, iqmx
    if( qq(i).gt.q25x ) then
        iq25 = i-1
        go to 100
    end if
end do
100 if( iq25.eq.0 ) iq25 = iqmx
jq25 = 0
do i = ncalx, iqmx, -1
    if( qq(i).gt.q25x ) then
        jq25 = i+1
        go to 200
    end if
end do
200 if( jq25.ge.ncalx ) jq25 = iqmx
c
rmin = 999.
its = 0
do i = 2, iq25-1
    call sokanl( xdt, qtot, 1, i, r1 )
    call sokanl( xdt, qtot, i, iq25, r2 )
    r = (abs(1.-r1)+abs(1.-r2))*0.5
    if( r.lt.rmin ) then
        rmin = r
        its = i
    end if
end do
if( its.eq.0 ) its = 1
c
rmin = 999.
ite = 0
do i = jq25+1, ncalx-1
    if( i.eq.140 ) then
        icv = 0
    end if
    call sokanl( xdt, qtot, i, ncalx, r1 )
    call sokanl( xdt, qtot, jq25, i, r2 )
    r = (abs(1.-r1)+abs(1.-r2))*0.5
    if( r.lt.rmin ) then
        rmin = r
        ite = i
    end if
end do

```

```

        end if
    end do
    if( ite. eq. 0 )  ite = ncalx
c
    ts  = 1.0
    itsa = iqmx-its
    if( itsa. gt. 0 )  then
        a = (qmax-qq(its))/float(itsa)
        b = qq(its)-a*float(its)
        if( nint(a*1000.). eq. 0 )  then
            ts = 1.
        else
            ts = -b/a
            if( ts. le. 0. )  ts = 1.
        end if
        ta = float(iqmx)-ts
    else
        ta = 1.0
    end if
    itsa = iqmx-ite
    if( itsa. lt. 0 )  then
        a = (qmax-qq(ite))/float(itsa)
        b = qq(ite)-a*float(ite)
        if( nint(a*1000.). eq. 0 )  then
            te = ncalx
        else
            te = -b/a
        end if
        tr = te-ts
        if( tr. ge. ncalx )  tr = ncalx
    else
        tr = ncalx
    end if
c
    return
end
c
-----
c
subroutine sokanl (x, y, n1, n2, r)
dimension  x(1), y(1), xl(720), yl(720)
c
m = 0
if( n1. lt. n2 )  then
    n3 = 1
else
    n3 = -1
end if
do i = n1, n2, n3
    if(x(i). ge. 0.0001 . and. y(i). ge. 0.0001) then
        m = m+1
        xl(m) = x(i)
        yl(m) = y(i)
    end if
end do
if( m. le. 1 )  then
    a = 1.0
    b = 0.0
    r = 0.0

```



```

else
c
    sumxy = 0.0
    sumx  = 0.0
    sumy  = 0.0
    sumx2 = 0.0
    sumy2 = 0.0
    do i = 1, m
        sumxy = sumxy+xl(i)*yl(i)
        sumx  = sumx +xl(i)
        sumy  = sumy +yl(i)
        sumx2 = sumx2+xl(i)*xl(i)
        sumy2 = sumy2+yl(i)*yl(i)
    end do
c
    zm  =float(m)
    buns = sumxy-sumx*sumy/zm
    bunbo = (sumx2-sumx*sumx/zm)*(sumy2-sumy*sumy/zm)
    if( bunbo. gt. 0.0 ) then
        r  = buns/bunbo
    else
        r  = 1.0
    end if
end if
c
return
end
c
c
c-----
c
c      ***** 河道追跡貯留関数モデル定数算定の近似式
c
c      subroutine kpxcal( amm, ta, tr, ak1, ak2, ap1, ap2 )
c
c      dimension  a(0:3), b(0:4, 4), c(0:4, 4), d(0:4, 4), e(0:4, 4)
c      data b /    0.8545,  -4.8430,  33.8238,  -72.2531,  49.7646
c      1      ,    0.2404,  -1.1777,   8.0276,  -16.2876,   9.8026
c      2      ,    0.5844,   0.9107,  -32.9092,  96.5526,  -80.2872
c      3      ,   -0.3999,  10.4083,  -47.3310,  71.7788,  -31.9668 /
c      data c /    0.5229,  22.7545, -154.6917,  327.8772, -224.9687
c      1      ,    1.0848,  23.9708, -126.7464,  230.5159, -124.7640
c      2      ,   -0.9989, -18.7643,  216.3786, -546.4481,  422.4272
c      3      ,    1.3325, -34.6802,  151.6250, -223.8849,  95.8506 /
c      data d /   -0.8426, -39.5757,  249.2689, -512.7692,  345.5463
c      1      ,    0.1929, -19.1466,  90.2282, -157.3522,  79.5772
c      2      ,    2.7263,  53.1813, -427.1458,  976.2463, -713.3760
c      3      ,   -0.7202,  24.5035, -98.6764,  138.5036, -52.1388 /
c      data e /    0.2763,  22.4122, -130.0060,  259.4679, -171.6907
c      1      ,    0.0000,   0.0000,   0.0000,   0.0000,   0.0000
c      2      ,   -1.1199, -37.4282,  249.4726, -535.5305,  376.2721
c      3      ,    0.0000,   0.0000,   0.0000,   0.0000,   0.0000 /
c
c      tar = ta/tr
c
c      call a03cal( a(0), b(0, 1), c(0, 1), d(0, 1), e(0, 1), tar )
c      ak1 = a(0)+a(1)*amm+a(2)*amm**2. +a(3)*amm**3.
c
c      call a03cal( a(0), b(0, 2), c(0, 2), d(0, 2), e(0, 2), tar )
c      ak2 = a(0)*exp(a(1)*amm+a(2)*amm**2.)

```

14-3program. for. txt

```

c
call a03cal( a(0), b(0,3), c(0,3), d(0,3), e(0,3), tar )
ap1 = a(0)+a(1)*amm+a(2)*amm**2.+a(3)*amm**3.
c
call a03cal( a(0), b(0,4), c(0,4), d(0,4), e(0,4), tar )
ap2 = a(0)+a(1)*amm+a(2)*amm**2.
c
return
end
c
-----
c
subroutine a03cal( a,b,c,d,e,tar )
c
implicit double precision (a-h,o-z)
dimension a(0:3),b(0:4),c(0:4),d(0:4),e(0:4)
c
a(0) = b(0)+b(1)*tar+b(2)*tar**2.+b(3)*tar**3.+b(4)*tar**4.
a(1) = c(0)+c(1)*tar+c(2)*tar**2.+c(3)*tar**3.+c(4)*tar**4.
a(2) = d(0)+d(1)*tar+d(2)*tar**2.+d(3)*tar**3.+d(4)*tar**4.
a(3) = e(0)+e(1)*tar+e(2)*tar**2.+e(3)*tar**3.+e(4)*tar**4.
c
return
end
c
-----
c
***** ニュートン法による最適化手法
c
***** 成分回帰分析法によるモデル定数補正值算定
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)
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)
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))
a1=1./u(j, j)
do 5 k=n, j+1, -1
u(k, j)=a1*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
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
```

湧別川  
湧別川  
丸瀬布

2001年9月12日洪水

2 3 10 50 0.001

75.8

10.000 0.150 1.500

110 110

2001 9 10 10

6 2 0 0

0 0 1 0 10 11 10 0

河道A

1 20500 1.3834 0.6765

河道B

2 7300 1.5532 0.6642

A R E A 1

1 130.17

0.00 0.00 0.00 0.00 0.00 0.00 0.56 4.00 4.56

5.44

2.56 3.44 3.56 8.00 7.44 7.12 7.12 4.00 5.56

6.00

5.44 5.56 6.00 5.00 3.00 4.00 4.56 6.88 6.44

6.00

1.00 1.00 1.00 2.00 2.00 3.00 1.00 3.00 1.56

1.00

1.00 0.56 0.00 1.00 0.56 1.44 1.56 0.56 1.00

1.00

0.56 1.00 0.44 0.56 0.00 0.00 0.44 0.00 0.00

0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00

A R E A 2

2 143.80

0.00 0.00 0.00 0.00 0.00 0.00 0.14 4.00 4.14

5.86

2.14 3.86 3.14 8.00 7.86 6.28 6.28 4.00 5.14

6.00

5.86 5.14 6.00 5.00 3.00 4.00 4.14 7.72 6.86

6.00

1.00 1.00 1.00 2.00 2.00 3.00 1.00 3.00 1.14

1.00

1.00 0.14 0.00 1.00 0.14 1.86 1.14 0.14 1.00

1.00

0.14 1.00 0.86 0.14 0.00 0.00 0.86 0.00 0.00

0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00





	input2_1.txt								
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
丸瀬布地点実績流量	9.09	9.09	9.09	8.78	8.78	9.09	9.09	9.40	9.72
10.69	12.43	16.30	18.86	43.45	66.84	102.36	149.14	203.24	233.43
259.08	275.79	287.79	305.39	336.51	371.10	413.59	441.05	456.20	480.53
510.10	554.89	609.12	621.79	645.81	627.38	609.21	588.78	553.86	524.78
503.48	473.50	462.21	442.24	416.29	403.62	391.13	374.80	360.79	345.10
337.39	333.57	325.99	322.23	311.09	294.74	289.39	273.63	277.10	271.91
265.07	251.65	258.31	233.77	225.86	224.29	211.96	202.94	207.43	197.04
199.98	202.94	194.12	197.04	194.12	189.78	188.35	178.45	177.06	171.55
163.45	162.12	164.79	156.84	150.38	155.54	151.66	145.30	152.95	142.80
141.55	142.80	134.20	129.41	130.60	125.87	124.70	120.08	111.11	115.55
111.11	107.83	104.61	102.48	102.48	101.43	98.30	97.27	95.22	93.19
90.20									



湧別川  
湧別川  
開盛

2001年9月12日洪水  
2 3 10 50 0.001

75.80  
10.000 0.150 1.500  
110 110  
2001 9 10 10  
8 5 1 0  
1 0 0 1 0 1 0 10 11 10 1 0 0

河道C  
3 10400 1.6246 0.7030

河道D  
4 10000 1.7880 0.7457

河道E  
5 2700 2.5811 0.7057

河道F  
6 18200 0.6202 0.8894

河道G  
7 3800 1.5965 0.8186

丸瀬布地点実績流量

802.00  
9.09 9.09 9.09 8.78 8.78 9.09 9.09 9.40 9.72  
10.69  
12.43 16.30 18.86 43.45 66.84 102.36 149.14 203.24 233.43  
259.08  
275.79 287.79 305.39 336.51 371.10 413.59 441.05 456.20 480.53  
510.10  
554.89 609.12 621.79 645.81 627.38 609.21 588.78 553.86 524.78  
503.48  
473.50 462.21 442.24 416.29 403.62 391.13 374.80 360.79 345.10  
337.39  
333.57 325.99 322.23 311.09 294.74 289.39 273.63 277.10 271.91  
265.07  
251.65 258.31 233.77 225.86 224.29 211.96 202.94 207.43 197.04  
199.98  
202.94 194.12 197.04 194.12 189.78 188.35 178.45 177.06 171.55  
163.45  
162.12 164.79 156.84 150.38 155.54 151.66 145.30 152.95 142.80  
141.55  
142.80 134.20 129.41 130.60 125.87 124.70 120.08 111.11 115.55  
111.11  
107.83 104.61 102.48 102.48 101.43 98.30 97.27 95.22 93.19  
90.20

A R E A 7

7 71.50  
0.01 0.00 0.00 0.00 1.98 0.00 1.00 4.99 2.02  
1.99  
1.01 2.99 2.99 7.99 5.02 8.97 5.00 4.00 4.99  
4.01  
4.00 5.01 6.00 3.02 4.99 4.01 5.99 6.01 5.99  
6.02  
2.01 1.02 1.99 3.00 4.00 3.00 2.00 3.00 2.00  
1.00  
2.00 1.01 1.01 1.00 1.01 2.00 2.02 1.02 2.01  
1.99  
0.02 1.01 1.01 1.01 1.00 0.01 1.00 0.01 0.00



input2\_2.txt

0.00									
AREA 1 0									
	10	10.44							
1.00	1.00	0.00	0.00	0.00	0.00	0.00	1.00	4.00	4.00
1.00	2.00	2.00	2.00	6.00	7.00	5.00	4.00	4.00	4.00
5.00	4.00	6.00	6.00	5.00	4.00	5.00	5.00	7.00	5.00
8.00	3.00	3.00	1.00	3.00	4.00	3.00	2.00	3.00	2.00
1.00	2.00	2.00	2.00	1.00	3.00	2.00	4.00	3.00	3.00
1.00	2.00	2.00	2.00	2.00	1.00	1.00	1.00	1.00	0.00
1.00	1.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	1.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
AREA 1 1									
	11	169.59							
0.17	0.00	0.00	0.00	0.00	0.17	0.00	1.83	6.50	1.83
0.17	2.00	0.34	1.17	3.01	7.17	2.34	3.17	4.34	3.17
6.00	5.83	6.66	6.83	6.66	4.83	4.83	5.00	7.83	8.66
8.66	2.00	1.83	2.00	2.00	3.00	3.83	4.50	2.83	3.83
2.66	1.83	1.83	1.00	1.00	1.00	2.17	1.83	2.83	1.83
1.17	1.00	1.00	1.00	1.00	0.83	0.00	1.00	0.83	0.17
0.00	0.00	0.00	0.83	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
AREA 1 2									
	12	110.81							
0.53	0.53	0.00	0.00	0.00	0.00	0.00	1.47	5.41	3.06
0.53	2.00	1.06	1.53	4.12	7.00	3.59	3.53	4.00	3.53
5.47	4.94	6.47	6.47	5.94	4.47	5.00	5.00	7.47	6.88
8.47	2.53	2.53	1.47	2.53	3.53	3.47	3.41	3.00	2.94



input2\_2.txt

0.00									
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00									
開盛地点	流域平均雨量(7-14流域)								
0.3276	0.0000	0.0000	0.0374	0.4525	0.0000	1.3735	5.3124	2.6193	
0.8535									
1.7730	1.4426	1.8535	5.1416	6.7287	4.6075	3.9287	4.1812	3.8535	
5.2558									
4.8189	6.1120	6.3361	5.3664	4.5631	4.7011	5.2270	7.1465	6.8648	
7.9210									
2.3276	2.0647	1.6350	2.5546	3.5172	3.3735	3.1228	2.8907	2.8563	
1.7470									
1.9281	1.7011	1.3276	1.0374	1.6525	2.0719	2.6193	2.5101	2.2557	
1.2989									
1.1740	1.3276	1.3276	1.3276	0.9281	0.3276	1.0000	0.7011	0.0719	
0.3276									
0.3650	0.1896	0.7011	0.0000	0.0000	0.0000	0.3276	0.0000	0.3276	
0.0000									
0.0000	0.0000	0.0374	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
0.0000									
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
0.0000									
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
0.0000									
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
0.0000									
開盛地点	実績流量								
48.20	48.20	45.63	45.63	45.63	45.63	45.63	45.63	50.84	
56.33									
62.10	66.77	72.60	78.68	87.59	108.30	118.71	137.68	174.25	
219.24									
291.50	341.30	377.81	419.76	471.49	514.24	571.34	640.27	708.46	
765.50									
824.74	917.74	971.56	1004.58	958.17	958.17	952.57	919.35	886.71	
881.33									
875.96	823.21	797.44	777.13	742.20	722.61	688.95	684.20	665.40	
660.73									
656.09	637.67	619.52	615.02	597.20	579.63	562.33	549.53	532.68	
507.91									
495.75	475.80	464.03	448.56	437.13	418.41	407.38	396.49	392.90	
382.92									
378.60	370.02	365.76	359.43	353.15	344.87	338.72	334.65	330.61	
326.59									
322.59	316.64	312.71	304.92	302.99	299.14	295.32	291.52	287.74	
285.87									
282.13	275.54	268.43	254.49	247.65	234.27	227.72	221.26	214.89	
208.62									
196.35	190.35	184.45	172.93	172.93	167.30	161.77	156.34	150.99	
145.74									

17program.for.txt

```

C *****
C 第17回ゼミ
C 単一流域におけるカルマン・フィルタ理論を用いた実時間流出予測
C 有効雨量を用いた一般化貯留関数法+カルマンフィルタ
C
C 作成者 片山 直樹      作成日 2005.12.5
C *****
C
C     iflg=0 実測雨量を用いる
C     iflg=1 過去3時間移動平均雨量を用いる
C
C     s = k1*q**p1 + k2*d(q**p2)/dt
C     ds/dt = fr - q
C     k1 = 2.8235 * fc * A**0.24
C     k2 = 0.2835 * k1**2 * rr**(-0.2648)   !rr:平均有効雨量強度
C     p1 = 0.6 (固定)
C     p2 = 0.4648 (固定)
C     状態変量
C     x1=q**p2
C     x2=d(q**p2)/dt
C     f  :流出率
C     fc :流域平均粗度
C     r  :予測雨量
C
C 平均有効雨量強度を逐次計算する
C 平均雨量(現時間までの総雨量/ゼロでない雨の時間数) = 平均雨量強度
C -----
C     character suikei*80, kasen*80, chiten*80, title*80, rname(2)*22
C
C     dimension x(8), x_prn(8)
C     dimension p1(5,5), p2(5,5), q(5,5), u(5,5), u2(5,5)
C     dimension xex(5), p1ex(5,5), p2ex(5,5)
C     dimension p33(5,5), p3ex(5,5)
C     dimension qq(200), reav(200), dis(24,200)
C     dimension r(200), qq0(200), hrea(200)
C     dimension xpr(24), xpr_d(24,200), sig(24,200), sigq(24,200)
C     dimension ar(6), br(6)
C
C     common /st1/ wk1, wk2, zk2, wf, wfc, wp1, wp2, zp2, rain, n, area
C     common /st2/ h, h2, h3, h4
C     common /st3/ alph1, alph2, alph4, akei, bkei
C     common /hqcw/ ax(5), bx(5), hx(5), qx(5)
C
C     open(5, file='data.dat', status='unknown')
C     open(6, file='kalckH.out', status='unknown')
C     open(50, file='kalckQ.out', status='unknown')
C
C     n=2          ! 更新する状態変量x1, x2
C     n1=2        ! f, fcに関する配列
C     n2=1        ! rに関する配列
C     iyoso=6     ! 予測時間幅
C     wp1=0.6     ! モデル定数p1
C     zp2=0.4648 ! モデル定数p2
C     wp2=1.0/zp2
C
C     n1=10       ! 計算時間間隔 (1時間の分割数)
C     xn1=n1
C     h=1.0/xn1

```

17program. for. txt

```

h2=h**2
h3=h2*h
h4=h3*h
C
rtot=0.0
nrah=0
C
C*****
C データ読み込み開始
C   水系名
C     read(5, '(a80)') suihei
C
C   河川名
C     read(5, '(a80)') kasen
C
C   地点名
C     read(5, '(a80)') chiten
C
C   洪水名
C     read(5, '(a80)') title
C
C   流域面積
C     read(5, '(5f8.0)') area
C
C   f, fc
C     read(5, '(3f8.0)') pa_f, pa_fc
C
C   開始年、月、日、時
C     read(5, '(4i5)') iy, im, id, ih
C
C   降雨、水位データ数
C     read(5, '(2i5)') nr, nq
C
C   雨量
C     read(5, '(10f8.0)') (r(i), i=1, nr)
C
C   水位
C     read(5, '(10f8.0)') (hrea(i), i=1, nq)
C     read(5, '(i5)') nhq           ! H-Q式数
C     do j = 1, nhq
C       read(5, '(3f8.0)') ax(j), bx(j), hx(j) ! a, b, 適用下限水位
C       qx(j)=ax(j)*(hx(j)+bx(j))**2
C     end do
C     do j = nhq, 5
C       ax(j)=ax(nhq)
C       bx(j)=bx(nhq)
C       hx(j)=hx(nhq)
C       qx(j)=qx(nhq)
C     end do
C   流量算定
C     call hqcurv( 1, 1, nq, hrea, qq0 )
C
C... <実測雨量:0 or 過去3時間移動平均雨量:1>予測フラグ....
C   rname(0)='=実測雨量'
C   rname(1)='=過去3時間移動平均雨量'
C   read(5, '(i5)') iflg
C
C   補正定数の設定

```

```

c   実績雨量の場合
do i=1,6
  ar(i)=1.0
  br(i)=1.0
end do
c   過去3時間移動平均雨量の場合
if(iflg.eq.1) then
  data ar/1.021, 1.252, 1.394, 1.0, 1.0, 1.0/
  data br/0.851, 0.686, 0.585, 1.0, 1.0, 1.0/
  do i=1,nr
    reav(i)=(r(i)+r(i-1)+r(i-2))/3
    if(i.eq.2) reav(i)=(r(i)+r(i-1))/2
    if(i.eq.1) reav(i)=r(i)
  end do
end if

c
c   データ読み込み終了
C*****
  wf=pa_f
  wfc=pa_fc
C*****
c   ヘッダ部分出力開始
write(6,*)'一般化貯留関数法+カルマンフィルターによる洪水予測計算'
write(6,('地点名 : ',a80))chiten
write(6,('開始時間 : ',i4,'年',i2,'月',
+       i2,'日',i2,'時'))iy,im,id,ih
write(6,('流域面積 : ',f8.2))area
write(6,('分割数(nl) : ',i2))nl
write(6,('予測雨量(iflg) : ',i2,a22))iflg,rname(iflg)
write(6,806)
806 format(/' <---貯留指数--->',
+       '<---パラメータ初期値--->')
write(6,('6x, ',p1',',6x, ',p2',',7x, ',f',
+       ',6x, ',fc'))
write(6,('2f8.4,2f8.3'))wp1,zp2,wf,wfc
write(6,807)
807 format(/19x,' <----- 予測値 ----->',
+       '<----- 標準偏差 ----->',
+       '<---モデル定数--->')
write(6,808)
808 format(' NO ',x,'雨量 実測値',3x,
&         '1時間後',1x,'2時間後',1x,'3時間後',1x,2x,
&         '1時間後',2x,'2時間後',2x,'3時間後',2x,
&         'f',fc')
write(50,*)'一般化貯留関数法+カルマンフィルターによる洪水予測計算'
write(50,*)'計算流量出力'
write(50,('地点名 : ',a80))chiten
write(50,887)
887 format(/19x,' <----- 予測値 ----->',
+       '<----- 標準偏差 ----->')
write(50,888)
888 format(' NO ',2x,'実測流量',,4x,
&         '1時間後',3x,'2時間後',3x,'3時間後',4x,
&         '1時間後',3x,'2時間後',3x,'3時間後')
c   ヘッダ部分出力終了
C*****
c
do i=1,nq

```



17program. for. txt

```
    qq1=qq0(i)
    if(qq1. le. 1. 0) qq1=1. 0
    qq(i)=3. 6*qq1/area ! qq(i) : 観測流出高
end do
write(50, ' (' ' 1' ', f10. 3, 20x)' )qq(1)*area/3. 6 !流量の初期値印刷
C
C 初期値の設定
    call input(x, p1, p2, p33, q, u, u2, qq(1))
C
C*****
C 計算開始
C*****
    do 3000 ll=2, nq+1 ! llは現在時刻のループ
C
    do i = 1, n+n1
        x_prn(i) = x(i)
    end do
C
C 平均雨量強度を逐次計算する
C 平均雨量を平均雨量強度として与える
C 現時間までの総雨量/ゼロでない雨の時間数
    if(ll. le. nr) rtot = rtot+r(ll-1)
    if(r(ll-1) .gt. 0. 0) nrah=nrah+1
    nrahy=0
    rtoty=0. 0
C
C 状態変量X1(x1, x2)の保存
    do i=1, n
        xex(i)=x(i)
    end do
C
C X1に関する(2*2)の分散・共分散の保存
    do i=1, n
        do j=1, n
            p1ex(i, j)=p1(i, j)
        end do
    end do
C
C X2に関する(2*4)の分散・共分散の保存
    do i=1, n
        do j=1, n1
            p2ex(i, j)=p2(i, j)
        end do
    end do
C
C X3に関する(2*1)の分散・共分散の保存
    do i=1, n
        do j=1, n2
            p3ex(i, j)=p33(i, j)
        end do
    end do
C
C*****
C 予測計算開始
C*****
C
    do lt=1, iyoso
        rain=0. 0
```

```

17program. for. txt
if(iflg. eq. 0 .and. ll+lt. le. nr) rain=r(ll+lt-1) !実測雨量
if(iflg. eq. 1 .and. ll+lt. le. nr) rain=reav(ll-1) !3時間移動平均雨
量
c
c
平均雨量強度を予測雨量を踏まえて逐次更新する
if( rain .gt. 0.0 ) then
  nrahy=nrahy+1
  rtoty=rtoty+rain
  rah = x(3) * (rtot+rtoty) / (nrah+nrahy)
end if
if( rah. lt. 1.0 ) rah = 1.0 !平均雨量強度re≤1のときre=1にする
c
wk1 = 2.8235 * x(4) * area**0.24
zk2 = 0.2835* wk1**2 * rah**(-0.2648)
wk2 = 1.0 / zk2
c
c
予測雨量の分散 (過去3時間移動平均雨量選択時)
if(iflg. eq. 1) then
  rain0 = rain
  if( rain .lt. 0.1 ) rain0=0.1
  rain0 = ar(ll)*rain0**br(ll)
  u2(1,1) = akei**2.0 * ll * rain0**(2.0*bkei)
end if
c
c
状態変量、誤差分散・共分散の伝達
do k=1, n1
  call no2kal(x, p1, p2, p33, u, u2)
  if(x(1). le. 0.0) then
    x(1)=qq(1)**zp2
    x(2)=0.0
  end if
end do
c
c
乗算的ノイズ
xnoise=p1(1,1)+(x(1)*alph1)**2
c
c
予測流量算出
xpr(ll)=area*x(1)**wp2/3.6
xpr_d(ll, ll)=xpr(ll) ! 予測流量
c
c
予測誤差分散算出
call hqcurv( 2, 1, 1, hpr, xpr(ll) )
dis(ll, ll)=hpr
call hqcheck(hpr, no_hq)
b11 = 1.0 / (ax(no_hq))**0.5
varq=(wp2*(x(1)**(wp2-1.0)))**2.0
& * (area/3.6)**2.0*xnoise !流量について
varh = (b11**2.0)/4.0*xpr(ll)**(-1.0)*varq !水位について
c
c
標準偏差算出
sigq(ll, ll) = 1.645*varq**0.5 !流量90.0%信頼区間
sig(ll, ll) = 1.645*varh**0.5 !水位90.0%信頼区間
c
sig(ll, ll) = varh**0.5 !68.3%信頼区間 (標準偏差)
c
end do
c*****
c 予測計算終了

```

17program.for.txt

```

C*****
C
C*****
C      計算結果出力
      write(6,809) ll-1, r(ll-1), hrea(ll-1),
      &          (dis(i,ll), i=1,3),
      &          (sig(i,ll), i=1,3),
      &          (x_prn(i), i=n+1, n+n1)
809  format(i3, f6. 2, f8. 2, 2x, 3f8. 2, x, 3f9. 4, 2x, 3f8. 4)
      write(50,899) ll, qq(ll)*area/3.6, (xpr_d(i,ll), i=1,3),
      &          (sigq(i,ll), i=1,3)
899  format(i3, f10. 3, x, 3f10. 3, x, 3f10. 3)
C*****
C
C      状態変量X1(x1, x2)を元に戻す
      do i=1, n
        x(i)=xex(i)
      end do

C
C      X1に関する(2*2)の分散・共分散を元に戻す
      do i=1, n
        do j=1, n
          p1(i, j)=p1ex(i, j)
        end do
      end do

C
C      X2に関する(2*2)の分散・共分散を元に戻す
      do i=1, n
        do j=1, n1
          p2(i, j)=p2ex(i, j)
        end do
      end do

C
C      X3に関する(2*1)の分散・共分散を元に戻す
      do i=1, n
        do j=1, n2
          p33(i, j)=p3ex(i, j)
        end do
      end do

C
      rain=0.0
      if(ll+1.le.nr) rain=r(ll)

C
C      状態変量、誤差分散・共分散の伝達
      u2(1,1) = 0.0
      do k=1, n1
        call no2kal(x, p1, p2, p33, u, u2)
        if(x(1).lt.0.0) then
          x(1)=qq(1)**zp2
          x(2)=0.0
        end if
      end do

C
C      P1に乗算的ノイズを設定
      do i=1, n
        q(i, i)=(x(i)*alph1)**2 ! 観測誤差の分散 R (ck*α4)^2
        p1(i, i)=p1(i, i)+q(i, i)
      end do

```

17program. for. txt

```

C
    z=qq(11) ! 現在時刻の観測流出高
C
C
    カルマンフィルタによる状態変量、誤差分散・共分散の更新
    call no2upd(x, p1, p2, u, z)
C
3000 continue
C*****
C    計算終了
C*****
C
    stop
    end
C
C
    subroutine no2kal (x, p1, p2, p33, u, u2)
C-----
C
    カルマンフィルタによる伝達方程式
    s = k1*q**p1 + k2*d(q**p2)/dt
C
    extrapolation equation of kalman filter
    linearization of nonlinear vector equation
    x(k+1) = phi*x(k) + gamma*d(k)
C-----
C
    dimension x(1), y(5), f(15), g(4), z(10)
    dimension p1(5,5), p2(5,5), p3(5,5), p4(5,5), p5(5,5)
    dimension phi1(5,5), phi2(5,5), phi3(5,5)
    dimension u(5,5), u2(5,5)
    dimension p33(5,5)
C
    common /st1/ wk1, wk2, zk2, wf, wfc, wp1, wp2, zp2, rain, n, area
    common /st2/ h, h2, h3, h4
C
    n1=2      ! f, fcに関する配列
    n2=1      ! rに関する配列
C
C. . 方程式の拡張部分を配列に割り当てる
    z(1) = x(3)      ! 状態変量 f
    z(2) = x(4)      ! 状態変量 fc
    z(3) = rain      ! 状態変量 r
C
    cc1=0.0
    cc2=0.0
    cc3=0.0
C
    y1=x(1)
    y2=x(2)
C
C. . 計算上の変数への割り当て
    if(y1.gt.0.0) then
        cc1=y1**(wp1*wp2-1.0) ! x1^(p1/p2-1)
        cc2=y1**(wp1*wp2-2.0) ! x1^(p1/p2-2)
        cc3=y1**(wp2-1.0)    ! x1^(1/p2-1)
    end if
    e1 = wk1*wk2*wp1*wp2 ! (k1/k2)*(p1/p2)
    e2 = wp1*wp2        ! p1/p2
C
C. . a21~a25の計算

```

```

                                17program.for.txt
a21 = -e1*(e2-1.0)*cc2*y2 - wk2*wp2*cc3
a22 = -e1*cc1
a23 = z(3)*wk2
cccc a24 = (-wk2*e2*cc1*y2) * 2.8235*area**0.24 !2005.11.02
a24 = 2.8235*area**0.24 * (wk2*e2*cc1*y2
+      + 2.0*y1**wp2*wk2/wk1 - 2.0*z(1)*z(3)*wk2/wk1)
a25 = z(1)*wk2
c. . b2の計算
cccc b2 = e1*e2*cc1*y2 + wk2*y1**wp2*(wp2-1.0) - z(1)*z(3)*wk2
b2 = e1*(e2-2)*cc1*y2 + wk2*y1**wp2*(wp2-3.0) + z(1)*z(3)*wk2
c
c. . 計算上の変数への割り当て
a9 = a21 + a22**2.0
a10 = 2.0*a21*a22 + a22**3.0
bt1 = h/2.0 + a22*h2/6.0 + a9*h3/24.0
bt2 = 1.0 + a22*h/2.0 + a9*h2/6.0 + a10*h3/24.0
c
c. . Φ、Γの要素の計算
f(1) = 1.0 + a21*h*bt1 ! φ11
f(2) = h*bt2 ! φ12
f(3) = a21*f(2) ! φ21
f(4) = 1.0 + a22*h + a9*h2/2.0 + a10*h3/6.0
+      + (a21*a9+a22*a10)*h4/24.0 ! φ22
c
g(1) = h*bt1 ! γ12
g(2) = f(2) ! γ22
c
f(5) = a23*g(1) ! φ13
f(6) = a24*g(1) ! φ14
f(7) = a23*f(2) ! φ23
f(8) = a24*f(2) ! φ24
c
f(9) = a25*g(1) ! φ15
f(10) = a25*g(2) ! φ25
c
c Φ1、Φ2、Φ3の配列への割り当て ( i:行、j:列 )
c ..... phi1(i,j) and phi2(i,j) and phi3(i,j) matrices .....
c k=0
do i=1, n
do j=1, n
k=k+1
phi1(i,j)=f(k) ! Φ1... (2*2) 行列
end do
end do
c
k=4
do i=1, n
do j=1, n1
k=k+1
phi2(i,j)=f(k) ! Φ2... (2*2) 行列
end do
end do
c
k=8
do i=1, n
do j=1, n2
k=k+1
phi3(i,j)=f(k) ! Φ3... (2*1) 行列

```

```

    end do
  end do
C
C 状態変量の伝達方程式（状態変量X1, X2, X3の更新）
C ..... extrapolation of state variables .....
  do i=1, n
    y(i)=0.0
    do j=1, n
      y(i)=y(i)+phi1(i, j)*x(j)      !  $\phi$ 1の項
    end do
    do j=1, n1
      y(i)=y(i)+phi2(i, j)*z(j)      !  $\phi$ 2の項
    end do
    do j=1, n2
      y(i)=y(i)+phi3(i, j)*z(3)      !  $\phi$ 3の項
    end do
    y(i)=y(i)+g(i)*b2                !  $\gamma$ の項
  end do
C
  do i=1, n
    x(i)=y(i)                        ! 状態変量の更新
  end do
C
C 誤差分散・共分散行列の伝達方程式
C ..... extrapolation equation .....
C
C p1(i, j) : extrapolation of covariance matrix
C
C  $\Phi_1 P_1 \Phi_1^T(p_3)$  の計算
  call mul31(2, 2, 2, 2, phi1, p1, phi1, p3)
C
C  $\Phi_1 P_2 \Phi_2^T(p_4)$  の計算
  call mul31(2, 2, 2, 2, phi1, p2, phi2, p4)
C
C  $\Phi_1 P_1 \Phi_1^T(p_3) + \Phi_1 P_2 \Phi_2^T(p_4)$ 
  do i=1, n
    do j=i, n
      p3(i, j)=p3(i, j)+p4(i, j)
    end do
  end do
C
C  $\Phi_2 P_2 \Phi_2^T(p_4)$  を加える
  do i=1, n
    do j=i, n
      p3(i, j)=p3(i, j)+p4(j, i)
    end do
  end do
C
C  $\Phi_1 P_3 \Phi_3^T(p_4)$  の計算
  call mul31(2, 2, 2, 1, phi1, p33, phi3, p4)
C
C  $\Phi_1 P_3 \Phi_3^T(p_4)$  を加える
  do i=1, n
    do j=i, n
      p3(i, j)=p3(i, j)+p4(i, j)
    end do
  end do

```

17program. for. txt

```

C
C   Φ3P3TΦ1T(p4T) を加える
      do i=1, n
        do j=i, n
          p3(i, j)=p3(i, j)+p4(j, i)
        end do
      end do

C
C   Φ2U Φ2T(p4) の計算
      call mul31(2, 2, 2, 2, phi2, u, phi2, p4)

C
C   Φ3U2Φ3T(p5) の計算
      call mul31(2, 2, 1, 1, phi3, u2, phi3, p5)

C
C   P1(更新値) (2*2) 行列の計算
      do i=1, n
        do j=i, n
          p1(i, j)=p3(i, j)+p4(i, j)+p5(i, j)
          if(i. ne. j) p1(j, i)=p1(i, j)
        end do
      end do

C
C   p2(i, j) : extrapolation of covariance matrix
C
C   Φ1P2 の計算 (2*2) 行列
      call mul21(2, 3, 2, phi1, p2, p3)

C
C   Φ2U の計算 (2*2) 行列
      call mul21(2, 3, 3, phi2, u, p4)

C
      do i=1, n
        do j=1, n1
          p2(i, j)=p3(i, j)+p4(i, j)          ! P2(更新値) (2*2) 行列の計算
        end do
      end do

C
C   p3(i, j) : extrapolation of covariance matrix
C
C   Φ1P3 の計算 (2*1) 行列
      call mul21(2, 1, 2, phi1, p33, p3)

C
C   Φ3U2 の計算 (2*1) 行列
      call mul21(2, 1, 1, phi3, u2, p4)

C
      do i=1, n
        do j=1, n2
          p33(i, j)=p3(i, j)+p4(i, j)        ! P3(更新値) (2*1) 行列の計算
        end do
      end do

C
      return
      end

C
C
      subroutine no2upd(x, p1, p2, u, z)


---


C
C       カルマンフィルターによる更新方程式
C       update equation of kalman filter

```

```

C -----
C dimension x(1), p1(5,5), p2(5,5), u(5,5)
C dimension xm1(5), xm2(5), xkal(10)
C dimension p3(5,5), zm1(5,5), zm2(5)
C
C common /st1/ wk1, wk2, zk2, wf, wfc, wp1, wp2, zp2, rain, n, area
C common /st3/ alph1, alph2, alph4, akei, bkei
C
C n1=2      !パラメータに関する配列(f, fc)
C
C err=z-x(1)**wp2
C
C ck=x(1)**wp2
C
C 観測誤差分散 (H-Q変換の誤差を考慮しない)
C  rq=(ck*alph4)**2
C
C  hh1 = wp2 * x(1)**(wp2-1.0)      ! h1の計算
C  s = p1(1,1) * hh1**2 + rq ! HPHI+R (カルマンゲインの右辺第2項)
C
C カルマンゲインK1, K2の計算 (状態変量x1, x2に対するもの)
C  do i=1, n
C    xm1(i) = p1(i,1) * hh1
C    xkal(i) = xm1(i) / s
C  end do
C
C カルマンゲインK3~K4の計算 (f, fcに対するもの)
C  do i=1, n1
C    xm2(i) = p2(1,i) * hh1
C    xkal(i+n) = xm2(i) / s
C  end do
C
C 状態変量x1, x2, f, fcの更新
C  do i=1, n+n1
C    x(i) = x(i) + xkal(i) * err
C  end do
C
C マトリックスM1の計算
C  zm1(1,1) = 1.0 - hh1 * xkal(1)
C  zm1(1,2) = 0.0
C  zm1(2,1) = -hh1 * xkal(2)
C  zm1(2,2) = 1.0
C
C マトリックスM2の計算
C  zm2(1) = -hh1 * xkal(3)
C  zm2(2) = -hh1 * xkal(4)
C
C X1に関する(2*2)の誤差分散・共分散P1の更新
C  call mul21(2,2,2, zm1, p1, p3)
C  do i=1, n
C    do j=i, n
C      p1(i,j) = p3(i,j)
C      if( i. ne. j ) p1(j,i) = p1(i,j)
C    end do
C  end do
C
C パラメータの誤差分散・共分散Uの更新
C  do i=1, n1

```



17program. for. txt

```
do j=i, n1
  u(i, j) = zm2(i) * p2(1, j) + u(i, j)
  if( i. ne. j ) u(j, i) = u(i, j)
end do
end do
C
C X2に関する(2*2)の誤差分散・共分散P2の更新
call mul21(2, 2, 2, zm1, p2, p3)
do i=1, n
  do j=1, n1
    p2(i, j) = p3(i, j)
  end do
end do
C
return
end
C
C
C subroutine mul31(np, n2, n3, n4, a, b, c, d)
C -----
C multiplication of matrices a, b, and ct
C -----
C dimension a(5, 5), b(5, 5), c(5, 5), d(5, 5)
C
C do 10 i=1, np
C do 10 j=1, n2
C d(i, j)=0.0
C do 12 k=1, n3
C do 12 l=1, n4
12 d(i, j)=d(i, j)+a(i, k)*b(k, l)*c(j, l)
10 continue
return
end
C
C
C subroutine mul21(np, n2, n3, a, b, c)
C -----
C multiplication of matrices a and b
C -----
C dimension a(5, 5), b(5, 5), c(5, 5)
C do 10 i=1, np
C do 10 j=1, n2
C c(i, j)=0.0
C do 12 k=1, n3
12 c(i, j)=c(i, j)+a(i, k)*b(k, j)
10 continue
return
end
C
C
C subroutine input(x, p1, p2, p33, q, u, u2, qinit)
C -----
C input data for implementing kalman filter
C 初期値の設定
C -----
C dimension x(1), p1(5, 5), p2(5, 5)
C dimension p33(5, 5)
C dimension q(5, 5), u(5, 5)
```

## 17program. for. txt

```

dimension u2(5,5)
c
common /st1/ wk1, wk2, zk2, wf, wfc, wp1, wp2, zp2, rain, n, area
common /st2/ h, h2, h3, h4
common /st3/ alph1, alph2, alph4, akei, bkei
common /st4/ u_wf, u_wfc
c
alph1 = 0.10      ! システム誤差としてx1, x2に乘じる定数
alph2 = 0.10      ! 初期値の誤差分散に乘ずる定数
alph4 = 0.10      ! 観測誤差としてckに乘じる定数
c
  予測雨量の誤差分散定数設定
akei = 1.135      ! 北海道全域
bkei = 0.88       ! 北海道全域
c
n1=2   ! f, fcに関する配列
n2=1   ! rに関する配列
c
do i=1, n
  x(i)=0.0
end do
c
x(1)=qinit**zp2  ! 状態変量x1の初期値
x(2)=0.0         ! 状態変量x2の初期値
x(3)=wf          ! 状態変量x3の初期値
x(4)=wfc        ! 状態変量x4の初期値
c
do i=1, n
  do j=1, n
    p1(i, j) = 0.0  ! X2に関する(2*2)の分散・共分散
  end do
end do
p1(1, 1) = (alph2*x(1))**2.0 ! X1(x1)に関する(2*2)の分散・共分散
p1(2, 2) = (alph2*x(1))**2.0 ! X1(x2)          "
c
do i=1, n
  do j=1, n1
    p2(i, j) = 0.0  ! X2に関する(2*2)の分散・共分散
  end do
end do
c
do i=1, n
  do j=1, n2
    p33(i, j) = 0.0 ! X3に関する(2*1)の分散・共分散
  end do
end do
c
do i=1, n
  do j=1, n
    q(i, j) = 0.0  ! システム誤差 分散・共分散
  end do
end do
c
do i=1, n1
  do j=1, n1
    u(i, j) = 0.0  ! U1のパラメータ誤差 分散・共分散
  end do
end do

```

17program. for. txt

```
u(1, 1)=(wf *0. 1)**2
u(2, 2)=(wfc*0. 1)**2
c
do i=1, n2
  do j=1, n2
    u2(i, j) = 0. 0    ! U2のパラメータ誤差 分散・共分散
  end do
end do
c
return
end
c
c
subroutine hqcurv(itype, ist, ied, h, q)
-----
c
c      水位流量曲線
c      itype=1:水位 --> 流量
c      itype=2:流量 --> 水位
c
-----
dimension h(1), q(1)
common/hqcv/ ax(5), bx(5), hx(5), qx(5)
go to (1, 2), itype
c
1 do 10 j=ist, ied
  if(h(j). ge. 0. 0) then
    k=1
    if(h(j). ge. hx(2)) k=2
    if(h(j). ge. hx(3)) k=3
    if(h(j). ge. hx(4)) k=4
    if(h(j). ge. hx(5)) k=5
    q(j)=ax(k)*(h(j)+bx(k))**2. 0
  else
    q(j)=-999
  endif
10 continue
return
c
2 do 20 j=ist, ied
  if(q(j). ge. 0. 0) then
    k=1
    if(q(j). ge. qx(2)) k=2
    if(q(j). ge. qx(3)) k=3
    if(q(j). ge. qx(4)) k=4
    if(q(j). ge. qx(5)) k=5
    h(j)=(q(j)/ax(k))**0. 5-bx(k)
  else
    h(j)=-999
  endif
20 continue
return
end
c
c
subroutine hqcheck(hdata, no_hq)
-----
c
c      水位流量曲線ナンバーをチェック
c
-----
common/hqcv/ ax(5), bx(5), hx(5), qx(5)
```

17program. for. txt

c

```
no_hq=1
if(hdata.ge.hx(2)) no_hq=2
if(hdata.ge.hx(3)) no_hq=3
if(hdata.ge.hx(4)) no_hq=4
if(hdata.ge.hx(5)) no_hq=5
```

c

```
return
end
```



18program.for.txt

```

C *****
C 第18回ゼミ
C 複合流域におけるカルマン・フィルタ理論を用いた実時間流出予測
C 有効雨量を用いた一般化貯留関数法+カルマンフィルタ
C
C 作成者 貞本 均          作成日 2006.4.27
C *****
C
C *   流域・河道ネットワーク識別データによる自動マトリックス作成
C *   水位～流量曲線は複数設定で、交点を求めて使う
C *   予測雨量は流域毎の値とする
C *   河道流入量 I を前時刻のxであらわす
C *   k2をfcの関数とした場合
C -----
C       main program ( kalmnfc )
C       s = k1*q**p1+k2*d(q**p2)/dt
C       ds/dt = f*r-q
C       k1=2.8235*fc*a^0.24
C       k2=0.2835*k1^2*reave^-0.2648
C *****
C *   n; number of state variables          *
C *   n1; number of parameters (runoff model) *
C *   n2; number of predicted rainfall      *
C *   nl; delt t control  delt t =1./nl    *
C *   iyoso; lead time of forecast         *
C *   状態量の選択                          *
C *   kase=1: x                             *
C *   kase=2: x, c                          *
C *   kase=3: x, r                          *
C *   kase=4: x, c, r                       *
C *   更新する状態量の選択                  *
C *   jconl=0: xのみ更新                    *
C *   jconl=1: xとモデル定数更新           *
C *****
C
C       character*80 title
C 共通common文を別ファイルから自動的に取り込む
C       include 'yosokufcj.cmn'
C
C       character snam(350)*8
C       dimension x(253), re(50)
C       dimension xex(253), ppex(253,253), mdays(12)
C
C       open(10, file='inputfc.dat', status='old')
C       open(6, file='kalmanfck2-j.out', status='unknown')
C
C       data mdays /31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31/
C
C ::::::::::::::::::::::::::::::::::::::::::::::::::::
C
C       mdays(2)=28
C       read(10, '(a80)') title
C       write(0,*) title
C
C       call input(snam)
C
C       wp1=0.6
C       zp2=0.4648

```

```

wp2=1. 0/zip2
xnl=float(nl)
hx=dt/3600. 0/xnl
h2=hx**2.
h3=h2*hx
h4=h3*hx

c
do 1200 ikse=1, nkse
ac1=ck1
ac2=ck2
kase=jkase(ikse)
write(0, *) ' case=', kase
n=(nryu+nkad)*2
n1=0
n2=0
nc=n
go to ( 1, 2, 3, 4 ), kase
4 n2=nryu      ! x+c+r
2 n1=2        ! x+c
go to 1
3 n2=nryu      ! x+r
1 continue    ! x
nx=n+n1+n2
go to ( 7, 8, 7, 8 ), kase
8 if(jconl. eq. 1) then ! モデル定数更新
nc=n+n1
end if
write(77, 2124) title
ll=1
write(77, 7831) kase, ll, ac1, ac2
2124 format(a80/1x, ' case time          f          fc' )
7 continue
i1=n+1
i2=i1+n1
write(21, 2122) title, kase, jconl, nx
write(22, 2123) title
2122 format(/13x, a80/16x, ' case=', i2, 2x, ' jconl=', i2, 4x, ' nx=', i3)
2123 format(a80)
c* ゼロクリアー
do 7010 i=1, 253 ! (流域数+河道数)*2=<50
do 7010 j=1, 253
fmx(i, j)=0. 0
amx(i, j)=0. 0
ax(i, j)=0. 0
pp(i, j)=0. 0
f(i, j)=0. 0
g(i, j)=0. 0
7010 continue

c
iy=iy0
im=im0
id=id0
ih=ih0
ih=ih-1
if(mod(iy, 4). eq. 0) mdays(2)=29
iy=iy-1900

c
c ##### main program start #####

```

## 18program. for. txt

```

C
C* 初期値誤差の設定
  call shoki(qq(1), x)          ! qqはmm/hr
  zbkal(1)=hh0(1)
  zakal(1)=hh0(1)

C
C  ===== forecasting =====
C  do 3000 ll=1, nq
C  write(0, *) 'll=', ll

C
C    if(ll. eq. 1) go to 7839 !初期値の状態から予測を始める

C
C    do 212 i=1, nryu
212  re(i)=r(i, ll)
      z=qq(ll)          ! mm/h hq変換された流出高を観測値とする

C
C    ..... xk+1(-) <== xk(+) .....
C    lt=0
C    do 200 k=1, nl
C    ..... extrapolation equation of kalman filtering .....
C* 状態量と誤差の推移計算 実測降雨誤差無し
      if(n2. gt. 0) then
        do 201 i=i2, nx
          pp(i, i)=0. 0
201    continue
        end if

C
C    -----
C    call no2kal(ll, lt, k, re, x)

C
C    call wrcal(ll, k, x)
200  continue

C
C    ..... update equation of kalman filtering .....
C* 観測値によるフィルタリング
C* system error add
      do 205 i=1, n
        q(i, i)=(x(i)*alph2)**2.
        pp(i, i)=pp(i, i)+q(i, i)
205  continue

C
C    -----
C    call no2upd(ll, z, x)

C
C* c1, c2, c3 の更新値 print
      if(kase. eq. 1 .or. kase. eq. 3) go to 7839
      if(jconl. ge. 1) then
        write(77, 7831) kase, ll, ac1, ac2
      end if
7831 format(2i5, 3f10. 5)
7839 continue

C
C    ..... store statistics on t=k .....
C    do 210 i=1, nc
210  xex(i)=x(i)
      do 215 i=1, nx
        do 215 j=1, nx
215  ppex(i, j)=pp(i, j)
          xc1=ac1
          xc2=ac2

C

```



```

c ..... iyoso-hour ahead prediction .....
c
  do 250 lt=1, iyoso
c 予測雨量は流域毎の値とする
  do 213 j=1, nryu
    re(j)=rp(j, ll, lt)
    if(n2.gt.0) then
      i=i2+j-1
c* 予測雨量がゼロのとき0.1mm相当の誤差分散を与えておく。
      pp(i, i)=ap*ap*float(lt)*0.1**(2.*bp)
      if(re(j).gt.0.1) pp(i, i)=ap*ap*float(lt)*re(j)**(2.*bp)
    end if
  213 continue
c
  do 260 k=1, nl
c -----
  call no2kal(ll, lt, k, re, x)
c -----
  260 continue
c * system error add
  do 270 i=1, n
    q(i, i)=(x(i)*alph2)**2.
    pp(i, i)=pp(i, i)+q(i, i)
  270 continue
c
c* x(1)から水位hへの変換及び誤差伝播 *** 状態量から水位の算定***
c -----
  call xtoh(y, py, x)
c -----
  ypr(lt)=y
  yva(lt)=py
  250 continue
c
  3001 continue
  ih=ih+1
  if(ih.gt.24) then
    ih=1
    id=id+1
    if(id.gt.mdays(im)) then
      id=1
      im=im+1
      if(im.gt.12) then
        im=1
        iy=iy+1
      end if
    end if
  end if
  kkm(ll)=im
  kd(ll)=id
  kh(ll)=ih
c
  do 310 i=1, nc
  310 x(i)=xex(i)
  do 315 i=1, nx
  do 315 j=1, nx
  315 pp(i, j)=ppex(i, j)
  go to ( 25, 26, 25, 26 ) kase
  26 continue

```

```

        if(jconl.eq.1) then
          ac1=xc1
          ac2=xc2
        end if
25 continue
c * 予測結果を格納する
    call store(11)
3000 continue
c * 予測結果をファイルに出力する
    call fileo
c * 全地点のアウトプット*
    nkp=(nall-1)/14+1
    do 3435 k=1,nkp
      kp1=(k-1)*14+1
      kp2=k*14
      if(kp2.gt.nall) kp2=nall
      write(6,3456) (snam(kp),kp=kp1,kp2)
      do 3434 i=1,nq
        if(k.ne.nkp) write(6,3457) i,nl,(wr(kp,i,nl),kp=kp1,kp2)
        if(k.eq.nkp) write(6,3458) i,nl,(wr(kp,i,nl),kp=kp1,kp2)
        *
        , r(nryu+1,i),wr(nall+1,i,nl)
3434 continue
3435 continue
3456 format(2x,'11',2x,'k',14a8)
3457 format(i4,i3,14f8.2)
3458 format(i4,i3,16f8.2)
c
1200 continue
    stop
    end
c
c*****
    subroutine input(snam)
    include 'yosokufcj.cmn'
    character snam(350)*8
    dimension a1(350),a2(350),a3(350),a4(350)
c ***** 流域流出解析と河道追跡のためのデータ入力
c nl: 1時間の計算分割数
c dt: データ入力時間間隔なども含む計算の基本時間間隔(標準は3600sec)
c xrave:>0 のときrave=xrave とする(全流域同一値が使われる)
c hiq: 河道上流端平均流入量の比流量(0.5m3/s/km2を標準値とする)
c ck1,ck2: モデル定数
c nr: 雨量入力時間数 nq: 計算時間数(=水位流量入力時間数)
c nko : 計算地点数
c nryu: 流域数(雨量入力流域) nqin: 上流端流入量入力地点数
c nkad: 追跡計算地点数(河道および直下が河道でない合流点の合計数)
c nall: 全地点数 r: 流域毎時雨量 qd: 上流端毎時流入量
c a1(i): area: 流域面積(km2) --> 上流端流入の流域面積も入力すること
c a2(i): cleg: 河道長(m) a3(i): alph:  $\alpha$  a4(i): cmm: m
c hlm: 実測水位がhlm以上の期間についてRMSEを計算する
    read(10,'(i5,5x,3f10.0)') nl,dt,xrave,hiq
    read(10,'(8f10.0)') ck1,ck2 ! f0,fc0
    read(10,'(4i5)') iy0,im0,id0,ih0
    read(10,'(2i5,f10.0)') nr,nq,hlm
    read(10,'(4i5)') nko
    if(nl .le. 0) nl=10
    if(dt .lt. 1.) dt=3600. ! dt=3600sec が基本
    if(hiq .le. 0.) hiq=0.5

```

## 18program. for. txt

```

nrp=nq
if(nr.le.0.or.nq.le.0) stop "nr=0 or nq=0"
if(nko.le.0) stop "nko=0"
nryu=0
nqin=0
nkad=0
ngor=0
nminus=0
sama=0.0      ! 流域のみの合計面積 (km2)
suma=0.0      ! 上流端流入の面積も含む全面積 (km2)
do 10 i=1, nko
read(10, '(4x, 5i4, 10x, 4f10.0)') mmod(i), mwr(i)
*      , los(i), lwr(i, 1), lwr(i, 2)
*      , a1(i), a2(i), a3(i), a4(i)

nm1=mmod(i)-mmod(i)/10*10
if(nm1.eq.1) go to 11 !<--流域
if(nm1.eq.2) go to 12 !<--上流端
if(nm1.eq.3) go to 13 !<--河道追跡
if(nm1.eq.4) go to 14 !<--合流点
stop
11 nryu=nryu+1
area(nryu)=a1(i)
sama=sama+a1(i)
go to 10
12 nqin=nqin+1
arez(nqin)=a1(i) ! 上流端流入量についても面積を入力する
suma=suma+a1(i)
go to 10
13 nkad=nkad+1
cleg(nkad)=a2(i)      ! L
alph(nkad)=a3(i)     ! α
cmm(nkad)=a4(i)      ! m
if(los(i).ge.2) go to 10
c* 流域qと河道inが同じ場所を探す
if(lwr(i, 1).eq.mwr(i)) then
nminus=nminus+1
end if
go to 10
14 continue
ngor=ngor+1
10 continue
suma=suma+sama
nall=nryu*2+nkad*2+ngor+nqin-nminus
write(66, 6613) nall, nryu, nkad, ngor, nqin, nminus
6613 format(' nall--nminus', 6i3)
c
if(nryu.le.0) stop '流域数>0 にすること'
c
c* 全地点の名前readする (右詰めで入力した方がきれいに出力される)
read(10, '(8(2x, a8))') (snam(i), i=1, nall)
c
c* 雨量データread
if(nryu.le.0) go to 7101
do 29 i=1, nryu+1
do 29 j=1, nq
r(i, j)=0.0
29 continue

```

```

    read(10,*)
    do 20 j=1,nr
    read(10,'(50f10.0)') (r(i,j),i=1,nryu)
20 continue
c* readした雨量データをwrに格納
    nry=0
    do 21 nks=1,nko
    nm1=mmod(nks)-mmod(nks)/10*10
    if(nm1.ne.1) go to 21
    nry=nry+1
    iw=mwr(nks)
    do 22 j=1,nr
    do 22 k=1,nl
    wr(iw,j,k)=r(nry,j)
22 continue
21 continue
c* 流域平均雨量計算
    do 723 j=1,nr
    r(nryu+1,j)=0.0
723 continue
    do 724 j=1,nr
    do 724 i=1,nryu
    r(nryu+1,j)=r(nryu+1,j)+r(i,j)*area(i)/sama
724 continue
    do 725 i=1,nryu
    rave(i)=0.0
    ir=0
    do 726 j=1,nr
    if(r(i,j).gt.0.) then
    rave(i)=rave(i)+r(i,j)
    ir=ir+1
    end if
726 continue
    rave(i)=rave(i)/float(ir)      ! 流域ごとの平均雨量
    if(xrave.gt.0.) rave(i)=xrave ! 予測の時は通常xraveを与えておく
725 continue
7101 continue
c* 上流端流入量データread&格納
    if(nqin.le.0) go to 7102
    read(10,*)
    do 30 j=1,nq
    read(10,'(10f10.0)') (qd(i,j),i=1,nqin)
30 continue
    nzy=0
    do 31 nks=1,nko
    nm1=mmod(nks)-mmod(nks)/10*10
    if(nm1.ne.2) go to 31
    nzy=nzy+1
    iw=mwr(nks)
    do 32 j=1,nq
    wr(iw,j,nl)=qd(nzy,j)
    if(j.eq.1) qqq=qd(nzy,j)
    if(j.gt.1) qqq=qd(nzy,j-1)
    dqz=(qd(nzy,j)-qqq)/float(nl)
    do 166 k=1,nl
    wr(iw,j,k)=qqq+dqz*float(k)
166 continue
32 continue

```

```

31 continue
7102 continue
c 各計算地点の上流全面積を求める areak(nko)
  asum=0.0
  iryu=0
  jqin=0
  icn=0
  ist=1
  do 950 nks=1, nko
    nm10=mmod(nks)/10+1      ! 本川・支川の区別
    nm1=mmod(nks)-mmod(nks)/10*10
    if(nm10.eq.ist) go to 800
    if(nm10.lt.ist) go to 801
    dare(ist)=asum
    asum=0.0
    ist=ist+1
    go to 800
801 continue
    asum=asum+dare(ist-1)
    ist=ist-1
800 continue
    go to ( 821, 822, 823, 824 ) , nm1
821 continue
    iryu=iryu+1
    asum=asum+area(iryu)
    areak(nks)=area(iryu) ! 流域は常に最上流端である
cx  areak(nks)=asum
    go to 950
822 continue
    jqin=jqin+1
    asum=asum+arez(jqin)
    areak(nks)=arez(jqin) ! 上流端なのだからその上流はない
cx  areak(nks)=asum
    go to 950
823 continue
    icn=icn+1
824 continue
    areak(nks)=asum
    if(nm1.eq.4) go to 950

c
c 無次元領域で計算したk3, k4を使って実領域のk3, k4に換算する
  amm=cmm(icn)
  call kpxcal( amm, wk01, wk02, wp01, wp02 )
  qm=hiq*3.6 ! mm/hr --->q* ハットの こと

c
c k3, k4をm3, hr系に換算する
  an=areak(nks) ! A
  e1=alph(icn)*cleg(icn) ! αL(河道長の単位は[km]では無く[m])
  e2=cmm(icn) ! m
  e3=an/3.6
  e4=1000.*an
  e5=e3**e2*e1*qm**(e2-wp01)/e4
  e6=e3**(2.*e2-1.)/e4*e1*e1/3600.*qm**(2.*e2-wp02-1.)
  vk3(icn)=wk01*e5 ! Kh3
  zk4=wk02*e6 ! Kh4
  vk4(icn)=1./zk4
  vp3(icn)=wp01 ! p3
  zp4(icn)=wp02 ! p4

```

950 continue

```

C*****
cs jysr=1 予測雨量を(現時刻を含む)前3時間平均雨量とし、自動計算する。
cs jysr=0 予測雨量をreadする(1流域分だけ入力し、全流域同一値とする)
cs jysr=2 予測雨量をreadする(流域ごとに入力する)
cs jysr<0 完全予測雨量とする(予測雨量の入力なし)
c* alph1: model parameter error
c* alph2: system noise
c* alph3: 観測誤差vk (観測誤差分散 $rq=(ck*alph3)**2$ )
c* beta: initial value error
c* ap,bp: predicted rainfall error  $ap**2*1*rp**2bp$ 
C*****
C
  read(10, 104) jcon1, jysr, iyoso,
  & nkse, (jkase(j), j=1, nkse)
  if(iyoso. le. 0) iyoso=6
  read(10, 101) alph1, alph2, alph3, beta, ap, bp
  if(alph1. gt. 900.) read(10, 101) (alpc(i), i=1, 3) ! c11, c12, c13の標準
偏差を与えるとき
  write(21, 401) alph1, alph2, alph3, beta, ap, bp, xrave, hiq
  write(21, 402) ck1, ck2, (alpc(i), i=1, 3)
100 format(16i5)
101 format(8f10. 0)
102 format(a80)
103 format(16f5. 0)
104 format(2i2, 6x, 15i5)
108 format(10f8. 0)
401 format(18x, 'alph1      alph2      alph3      beta'
* 8x, 'ap', 8x, 'bp', 5x, 'xrave', 7x, 'hiq', /13x, 8f10. 3)
402 format(21x, 'ff', 8x, 'fc', 6x, 'alc1', 6x, 'alc2'
*      , 6x, 'alc3' /13x, 5f10. 3)
C
C+++++++ h-q 式とその数nhc read+++++++
c* h0:H-Q式の基準標高  $q=a0(h+a1)^2$  の h が、h0からの高さを
c* 使うようになっているときに入力する。
  read(10, 105) nhc, h0
105 format(i5, 8f10. 0)
  write(21, 405) nhc, h0
405 format(13x, 'nhc=', i3, 10x, 'h0=', f9. 3, 1x,
*      'q=a(h+b)^2', 10x, 'h=b0+b1*sqrt(q)' /
*      13x, 9x, 'a', 9x, 'b', 8x, 'b0', 8x, 'b1')
  do 160 i=1, nhc
  read(10, 101) ab0(i), ab1(i) !  $q=ab0(i)*(h-h0+ab1(i))^2$ 
  bb0(i)=-(ab1(i)-h0)
  bb1(i)=1./sqrt(ab0(i))
  write(21, 403) i, ab0(i), ab1(i), bb0(i), bb1(i)
403 format(10x, i2, 1x, 4f10. 4)
160 continue
  if(nhc. le. 4) then
  do 161 i=nhc+1, 5
161 write(21, 403) i
  end if
cccccccccc hq 式の交点を hc, qc とする ccccccccccccccccccc
  if(nhc. le. 1) go to 176
  do 163 i=1, nhc-1
  hc(i)= bb0(i)+bb1(i)*(bb0(i+1)-bb0(i))/(bb1(i)-bb1(i+1))
163 continue
  do 175 i=1, nhc-1

```

```

                                18program. for. txt
175 qc(i)=ab0(i)*(hc(i)+ab1(i)-h0)**2.    ! m3/s
176 continue
c
c * 実測水位をreadして流量変換する
  read(10,107) ij
107 format(i2)
  if(ij. eq. 0) read(10,101) (hh0(i), i=1, nq)
  if(ij. eq. 1) read(10,108) (hh0(i), i=1, nq)
  do 168 i=1, nq
  jj=1
  if(nhc. le. 1) go to 223
  do 222 k=1, nhc-1
  jj=k
  if(hh0(i). lt. hc(k)) go to 223
222 continue
  jj=nhc
223 continue
  qz(i)=((hh0(i)-bb0(jj))/bb1(jj))**2.
  if(qz(i). lt. 0.) then
6771  write(6,6771) i, jj, hh0(i), bb0(jj), bb1(jj), qz(i)
  format(2i4, 4f10. 3)
  stop "hq変換後流量が負になる"
  end if
  if(i. eq. 1) qqq=qz(i)
  if(i. gt. 1) qqq=qz(i-1)
  dqz=(qz(i)-qqq)/float(nl)
  do 167 k=1, nl
  wr(nall+1, i, k)=qqq+dqz*float(k)
167 continue
  write(66,6611) i, wr(nall+1, i, nl), qz(i)
6611 format(i5, 2f10. 2)
168 continue
  do 190 i=1, nq
  qq(i)=qz(i)*3. 6/suma    ! mm/h
190 continue
c
c * 予測雨量
  if(jysr. eq. 1) then ! 流域ごとに流域平均雨量の前3時間平均雨量を予測雨量とする
  do 164 i=1, nryu
  do 164 j=1, nrp
  if(j. eq. 1) rp(i, j, 1)=r(i, j)
  if(j. eq. 2) rp(i, j, 1)=(r(i, j-1)+r(i, j))/2.
  if(j. ge. 3) rp(i, j, 1)=(r(i, j-2)+r(i, j-1)+r(i, j))/3.
  do 164 k=2, iyoso
  rp(i, j, k)=rp(i, j, 1)
164 continue
  elseif(jysr. eq. 0) then ! 予測雨量を1ケース入力して全流域同一値にする
  read(10,*)
  do 180 j=1, nrp
  read(10, '(24f8. 0)') (rp(1, j, k), k=1, iyoso)
180 continue
  do 181 i=2, nryu
  do 181 j=1, nrp
  do 181 k=1, iyoso
  rp(i, j, k)=rp(1, j, k)
181 continue
  elseif(jysr. eq. 2) then ! 予測雨量を入力値にする -->流域の数だけ入力

```

する

```

do 182 i=1, nryu
  read(10, *)
  do 182 j=1, nrp
    read(10, '(24f8.0)') (rp(i, j, k), k=1, iyoso)
182 continue
    elseif(jysr.lt.0) then ! 完全予測雨量にする
      do 183 i=1, nryu
        do 183 j=1, nrp
          do 183 k=j+1, j+iyoso
            rp(i, j, k-j)=r(i, k)
183 continue
          end if
        end if
      end if
    end if
  end if
end if

c
c * 予測上流端流入量
  if(nqin.gt.0) then ! 予測上流端流入量を与える(readする)
    do 189 j=1, nqin
      read(10, *)
      do 189 i=1, nq
        read(10, '(24f10.0)') (qdp(i, j, k), k=1, iyoso)
189 continue
      end if
    end if
  return
end

c
c*****
  subroutine xtoh(y, py, x)
c *****
c * x1=>h and cal. error of predicted water level *
c * y:流出高の期待値 *
c * py:流出高の誤差分散 *
c *****
c*
  include 'yosokufcj.cmn'
  dimension x(253), h(253, 253), aa(253, 253), bb(253, 253)
  dimension ppy(253, 253), kkb(5)

c
  do 110 i=1, nx
    do 110 j=1, nx
110 h(i, j)=0.0
c H行列の成分を指定
  y=0.0
  ist0=1
  iryu=nryu
  icn=nkad
  kry=0
  kka=0
  kkc=0
  do 301 i=1, 5
301 kkb(i)=0
  iat=ist0
  do 302 nks=nko, 1, -1
  call eqsys(nks, nm1, ist0, iat, iryu, icn, kry,
  * kka, kkb, kkc, kkd, ihh, p2h, ah, xh, x)
  if(kkd.ge.1.or.nm1.eq.4) go to 302
  if(nm1.eq.2) then
    iw=mwr(nks)
    p2h=1.0

```



```

                                18program. for. txt
    ah=areak(nks)
    xh=zr(iw)*3.6/ah                ! mm/hr
    go to 304
end if
h(1, ihh)=xh**(1./p2h-1.)/p2h*ah/suma
304 continue
y=y+xh**(1./p2h)*ah/suma          ! mm/hr 予測流出高
if(mmod(nks).eq.3) go to 333
302 continue
333 continue
c
    call matmul(1, nx, nx, h, pp, aa)
    call matt(1, nx, h, bb)
    call matmul(1, nx, 1, aa, bb, ppy)
    py=ppy(1, 1)                   ! 予測流出高の誤差分散(mm/hr)^2
c *
    return
end
c
c *-----
    subroutine store(ll)
    include 'yosokufcj. cmn'
    do 100 i=1, iyoso
    zqm=suma /3.6*ypr(i)            ! m3/s
    call qtoh(jj, zqm, zh)
    qf(ll, i)=zqm
    hf(ll, i)=zh
    sdh=0.0
    if(yva(i).lt.0.0) go to 110
    sdq=(suma/3.6)*sqrt(yva(i))    ! 予測流量の標準偏差
    sdh=bb1(jj)/2.*suma/3.6*sqrt(yva(i)/zqm) ! 予測水位の標準偏差
110 continue
cx    hl(ll, i)=hf(ll, i)-sdh      !標準偏差
cx    hu(ll, i)=hf(ll, i)+sdh
    ql(ll, i)=qf(ll, i)-1.645*sdq  !流量の90%信頼区間
    qu(ll, i)=qf(ll, i)+1.645*sdq
    hl(ll, i)=hf(ll, i)-1.645*sdh  !水位の90%信頼区間
    hu(ll, i)=hf(ll, i)+1.645*sdh
100 continue
    return
end
c
c ///////////////////////////////////////////////////////////////////
    subroutine fileo
    include 'yosokufcj. cmn'
c
    np=nq
c
    call gosa
    write(21, 661) (rmse(j), j=1, 6)
    write(21, 662) (ajre(j), j=1, 3)
    write(21, 663) (ajpe(j), j=1, 3)
661 format(49x, 'rmse', 6f10.5)
662 format(49x, 'jre', 3f10.5)
663 format(49x, 'jpe', 3f10.5)
c
    write(21, *) '水位予測結果'
    write(21, 625)

```

```

                                18program. for. txt
write(22,*) '流量予測結果'
write(22,625)
625 format(56x,'predict',25x,'σ',28x,'m-σ',26x,'m+σ'
* /4x,' m d h ',6x,'rain',4x,'observ  before  after',
* 4(5x,' 1hr',7x,' 2hr',7x,' 3hr',2x))
do 100 i=1,np
write(21,634) i,kkm(i),kd(i),kh(i),r(nryu+1,i),hh0(i),zbkal(i)
* ,zakal(i),(hf(i,j),j=1,3),(hu(i,j)-hf(i,j),j=1,3)
* ,(hl(i,j),j=1,3),(hu(i,j),j=1,3)
write(22,634) i,kkm(i),kd(i),kh(i),r(nryu+1,i),qz(i)
* ,qbkal(i)*(suma/3.6),qakal(i)*(suma/3.6),(qf(i,j),j=1,3)
* ,(qu(i,j)-qf(i,j),j=1,3),(ql(i,j),j=1,3),(qu(i,j),j=1,3)
100 continue
634 format(4i3,1x,f10.2,3f10.3,4(3f10.3))
return
end

c
c#####
subroutine no2kal(ll,lt,k,re,x)
-----
c
c      nonlinear storage function model
c      s = k1*q**p + k2*dq/dt
c      extrapolation equation of kalman filter
c      linearization of nonlinear vector equation
c      x(k+1) = phi*x(k) + gamma*d(k)
-----
c
include 'yosokufcj.cmn'
dimension x(253),y(253),re(50)
dimension dm(253,253),bb(253,253),cc(253,253)
c*
do 130 i=1,nx
do 130 j=1,nx
dm(i,j)=0.0
ax(i,j)=0.0
amx(i,j)=0.0
130 fmx(i,j)=0.0
c
iryu=0
icn=0
nzy=0
do 950 nks=1,nko
nm1=mmod(nks)-mmod(nks)/10*10
go to ( 221,222,223,950 ),nm1
221 continue
c* 流域流出計算
iryu=iryu+1
rain=re(iryu)
call ryuik(iryu,rain,x)
go to 950
222 continue
c* 上流端流入計算
nzy=nzy+1
iw=mwr(nks)
zr(iw)=wr(iw,ll,k)
if(lt.gt.0) then
if(lt.eq.1) qqq=qd(nzy,ll)
if(lt.gt.1) qqq=qdp(ll,nzy,lt-1)
dqz=(qdp(ll,nzy,lt)-qqq)/float(nl)

```

```

                                18program. for. txt
      zr(iw)=qqq+dqz*float(k)
    end if
    go to 950
c* 河道追跡&合流計算
  223 continue
      icn=icn+1
      call chase(nks, icn, x)
  950 continue
c
c* d2 を求める
      call matmul(nx, nx, 1, amx, ax, cc)
      call matsub(nx, 1, fmx, cc, dm)
c
c* 線形方程式の差分化
c*  $dx/dt=ax+d \Rightarrow x(k+1)=f(k)x(k)+g(k)d(k)$ 
      call risan
c
c      ..... calculation of state variables .....
      do 140 i=1, n
        y(i)=0.0
        do 145 j=1, nx
  145 y(i)=y(i)+f(i, j)*ax(j, 1)
          do 150 j=1, nx
  150 y(i)=y(i)+g(i, j)*dm(j, 1)
        140 continue
      c
        do 160 i=1, n
          x(i)=y(i)
          if(mod(i, 2).eq.1 .and. x(i).lt.0.0) x(i)=0.0
  160 continue
      c
      ..... extrapolation equation .....
      c      pp(i, j) : extrapolation of covariance matrix
c* 誤差共分散行列の伝達  $p(k+1)=\Phi(k)p(k)\Phi^t(k)$ 
      call matmul(nx, nx, nx, f, pp, bb)
      call matt(nx, nx, f, cc)
      call matmul(nx, nx, nx, bb, cc, pp)
c
      return
      end
c
c=====
      subroutine ryuik(iryu, rain, x)
      include 'yosokufcj. cmn'
      dimension x(253)
c
      k1=(iryu-1)*2+1
      k2=iryu*2
      wk1=2.8235*ac2*area(iryu)**0.24
      zk2=0.2835*wk1*wk1*rave(iryu)**(-0.2648)
      wk2=1./zk2
c      ***** 分割流域における流出計算
      c1=0.0
      c2=0.0
      c3=0.0
      c4=0.0
      y1=x(k1)
      y2=x(k2)

```

```

if(y1.le.0.0) go to 100
c1=y1**(wp1*wp2-1.0)
c2=y1**(wp1*wp2-2.0)
c3=y1**(wp2-1.0)
c4=y1**wp2
100 continue
e1 = wk1*wk2*wp1*wp2
e2 = wp1*wp2
c* dx/dt=a(x)*x+d(x)
c* f(x)=>fmx, a(x)=>amx, x=>ax
amx(k1,k2)=1.0
amx(k2,k1) = -e1*(e2-1.0)*c2*y2-wk2*wp2*c3
amx(k2,k2) = -e1*c1
fmx(k1,1)=y2
fmx(k2,1)=-e1*c1*y2-wk2*c4+ac1*rain*wk2
ax(k1,1)=y1
ax(k2,1)=y2
go to ( 1, 2, 3, 4 ), kase
4 continue
amx(k2,i2+iryu-1)=ac1*wk2
ax(i2+iryu-1,1)=rain
x(i2+iryu-1)=rain
go to 2
3 continue
amx(k2,i1+iryu-1)=ac1*wk2
ax(i1+iryu-1,1)=rain
x(i1+iryu-1)=rain
go to 1
2 continue
amx(k2,i1 )=rain*wk2 ! r/k2=df2/df
cx *** df2/dfc=df2/dk1*dk1/dfc+df2/dk2*dk2/dk1*dk1/dfc
amx(k2,i1+1)=(e1*c1*y2+2.*wk2*c4-2.*ac1*rain*wk2)/ac2
ax(i1 ,1)=ac1
ax(i1+1,1)=ac2
x(i1 )=ac1
x(i1+1)=ac2
1 continue
return
end

c
c =====
c subroutine chase(nks, icn, x)
c include 'yosokufcj.cmn'
c dimension x(253), kkb(5)

c
c ** 河道流入量iの設定 **
c 上流端流入は現ステップt=kの値を入力する
c
c
c an=areak(nks)
c l1=nryu*2+icn*2-1
c l2=l1+1
c wk3=vk3(icn)
c wp3=vp3(icn)
c wk4=vk4(icn)
c wp4=1./zp4(icn)

c
c ist0=mmod(nks)/10+1
c iryu=0

```

## 18program. for. txt

```

do i=nks-1, 1, -1
  nn1=mmod(i)-mmod(i)/10*10
  if(nn1.eq.1) then
    iryu=iryu+1
  end if
end do
rain=0.0
icm=icn-1
kry=0
kka=0
kkc=0
do 309 i=1, 5
309 kkb(i)=0
  iat=ist0
  do 304 i=nks-1, 1, -1
    call eqsys(i, nn1, ist0, iat, iryu, icm, kry,
*          kka, kkb, kkc, kkd, ihh, p2h, ah, xh, x)
    if(kkd.eq.2) go to 304
    if(kkd.eq.1 .or. nn1.eq.4) go to 304
    if(nn1.eq.2) then
      iw=mwr(i)
      p2h=1.0
      ah=areak(i)
      xh=zr(iw)*3.6/ah          ! mm/hr
      go to 303
    end if
    amx(l2, ihh)=wk4/p2h*xh**(1./p2h-1.)*ah/an
303 continue
    rain=rain+xh**(1./p2h)*ah/an          ! t-1
304 continue
305 continue
c
  con=wk3*wk4*wp3*wp4
  a=0.0
  c=0.0
  d=0.0
  e=0.0
  y1=x(l1)          ! t-1の値
  y2=x(l2)
  if(y1.gt.0.0) go to 12
  if(y1.lt.0.0) y1=0.0
  go to 14
12 a=y1**(wp3*wp4-2.)
  c=y1**(wp4-1.)
  d=y1**(wp3*wp4-1.)
  e=y1**wp4
14 continue
  a1=-con*(wp3*wp4-1.)*a*y2-wk4*wp4*c
  a2=-con*d
  amx(l1, l2)=1.0
  amx(l2, l1)=a1
  amx(l2, l2)=a2
  fmx(l1, 1)=y2
  fmx(l2, 1)=-con*d*y2-wk4*e+rain*wk4  ! rainは t-1 の値
  ax(l1, 1)=y1          ! xt-1
  ax(l2, 1)=y2
  return
end

```

## 18program. for. txt

```

c
c =====
c ***** 河道追跡貯留関数モデル定数算定の近似式
c
c   subroutine kpxcal( amm, ak1, ak2, ap1, ap2 )
c
c   dimension  a(0:3), b(0:4, 4), c(0:4, 4), d(0:4, 4), e(0:4, 4)
c   data b /    0.8545, -4.8430, 33.8238, -72.2531, 49.7646
c   1          ,    0.2404, -1.1777,  8.0276, -16.2876,  9.8026
c   2          ,    0.5844,  0.9107, -32.9092, 96.5526, -80.2872
c   3          ,   -0.3999, 10.4083, -47.3310, 71.7788, -31.9668 /
c   data c /    0.5229, 22.7545, -154.6917, 327.8772, -224.9687
c   1          ,    1.0848, 23.9708, -126.7464, 230.5159, -124.7640
c   2          ,   -0.9989, -18.7643, 216.3786, -546.4481, 422.4272
c   3          ,    1.3325, -34.6802, 151.6250, -223.8849, 95.8506 /
c   data d /   -0.8426, -39.5757, 249.2689, -512.7692, 345.5463
c   1          ,    0.1929, -19.1466, 90.2282, -157.3522, 79.5772
c   2          ,    2.7263, 53.1813, -427.1458, 976.2463, -713.3760
c   3          ,   -0.7202, 24.5035, -98.6764, 138.5036, -52.1388 /
c   data e /    0.2763, 22.4122, -130.0060, 259.4679, -171.6907
c   1          ,    0.0000,  0.0000,  0.0000,  0.0000,  0.0000
c   2          ,   -1.1199, -37.4282, 249.4726, -535.5305, 376.2721
c   3          ,    0.0000,  0.0000,  0.0000,  0.0000,  0.0000 /
c
c   tar=0.5
c   call a03cal( a(0), b(0, 1), c(0, 1), d(0, 1), e(0, 1), tar )
c   ak1 = a(0)+a(1)*amm+a(2)*amm**2.+a(3)*amm**3.
c
c   call a03cal( a(0), b(0, 2), c(0, 2), d(0, 2), e(0, 2), tar )
c   ak2 = a(0)*exp(a(1)*amm+a(2)*amm**2.)
c
c   call a03cal( a(0), b(0, 3), c(0, 3), d(0, 3), e(0, 3), tar )
c   ap1 = a(0)+a(1)*amm+a(2)*amm**2.+a(3)*amm**3.
c
c   call a03cal( a(0), b(0, 4), c(0, 4), d(0, 4), e(0, 4), tar )
c   ap2 = a(0)+a(1)*amm+a(2)*amm**2.
c
c   return
c   end
c
c =====
c   subroutine a03cal( a, b, c, d, e, tar )
c   dimension  a(0:3), b(0:4), c(0:4), d(0:4), e(0:4)
c
c   a(0) = b(0)+b(1)*tar+b(2)*tar**2.+b(3)*tar**3.+b(4)*tar**4.
c   a(1) = c(0)+c(1)*tar+c(2)*tar**2.+c(3)*tar**3.+c(4)*tar**4.
c   a(2) = d(0)+d(1)*tar+d(2)*tar**2.+d(3)*tar**3.+d(4)*tar**4.
c   a(3) = e(0)+e(1)*tar+e(2)*tar**2.+e(3)*tar**3.+e(4)*tar**4.
c
c   return
c   end
c
c =====
c*   subroutine risan
c*   線形方程式の差分化
c*   include 'yosokufcj. cmn'
c* dx/dt=ax+b => x(k+1)=f(k)x(k)+g(k)b(k)
c   dimension um(253, 253), at(253, 253)

```

```

                                18program.for.txt
dimension aw(253,253),bw(253,253),cw(253,253)
c
  t=hx
c* テイラー展開の項数計算
  atmax=0.
  do 100 i=1,nx
  do 100 j=1,nx
    at(i,j)=amx(i,j)*t
    ata=abs(at(i,j))
    if(ata.gt.atmax) atmax=ata
  100 continue
c*
c* matrix要素の大きさにかかわらず項数一定
  rm=4.
  km=4
c
c* 単位行列
  call mati(nx,um)
c*  $x(k+1)=f(k)*x(k)+g(k)*b(k)$   $f(k), g(k)$  の計算
c
  if(km.eq.1) then
    call matad(nx,nx,um,at,f)
    call matc(nx,nx,um,t,g)
  else
    ao=1./rm
    call matc(nx,nx,at,ao,aw)
    call matad(nx,nx,aw,um,bw)
    if(km.eq.2) go to 130
    do 140 j=km-1,2,-1
      bo=1./float(j)
      call matc(nx,nx,at,bo,aw)
      call matmul(nx,nx,nx,aw,bw,cw)
      call matad(nx,nx,cw,um,bw)
140    continue
130    call matmul(nx,nx,nx,at,bw,cw)
      call matad(nx,nx,cw,um,f)
      call matc(nx,nx,bw,t,g)
    endif
  return
end
c
=====
subroutine no2upd(ll,z,x)
c
-----
c
  update equation of kalman filter
c
=====
c
include 'yosokufcj.cmn'
dimension x(253),h(253,253)
dimension w1(253,253),w2(253,253),w3(253,253)
dimension gain(253,253),ht(253,253)
dimension kkb(5)
c
c 更新前流出高
  zq=wr(nall,ll,nl) ! m3/s
  qbkal(ll)=zq*3.6/suma ! mm/h
  sqq=sqrt(zq)
c 計算流量を水位変換するためのjjを計算する。
  jj=1
  if(nhc.le.1) go to 9

```

```

do 8 i=1, nhc-1
  jj=i
  if(zq. lt. qc(i)) go to 9
8 continue
  jj=nhc
9 continue
c
  zcal=bb0(jj)+bb1(jj)*sqg ! 水位 elm
  zbkal(ll)=zcal ! elm
  err=z-qbkal(ll) ! mm/h
c
c gain : kalman gain
c* 観測方程式の線形化
  do 110 i=1, nx
  do 110 j=1, nx
110 h(i, j)=0.0
c H行列の成分を指定
  ist0=1
  iryu=nryu
  icn=nkad
  kry=0
  kka=0
  kkc=0
  do 309 i=1, 5
309 kkb(i)=0
  iat=ist0
  do 304 nks=nko, 1, -1
  call eqsys(nks, nm1, ist0, iat, iryu, icn, kry,
  * kka, kkb, kkc, kkd, ihh, p2h, ah, xh, x)
  if(kkd. ge. 1 .or. nm1. eq. 2 .or. nm1. eq. 4) go to 304
  h(1, ihh)=xh**(1./p2h-1.)/p2h*ah/suma
  if(mmod(nks). eq. 3) go to 305
304 continue
305 continue
c* カルマン・ゲイン行列の計算
  call matmul(1, nx, nx, h, pp, w1)
  call matt(1, nx, h, ht)
  call matmul(1, nx, 1, w1, ht, w3)
c* rq: 観測誤差
  ck=qbkal(ll)
  rq=ck*alph3*ck*alph3
c
  hphtr=w3(1, 1)+rq
  w2(1, 1)=1./hphtr
c
  call matmul(nx, nx, 1, pp, ht, w1)
  call matmul(nx, 1, 1, w1, w2, gain)
c
c ..... update state variables .....
c* 状態量の更新 x(+)
  do 120 i=1, nc
  x(i)=x(i)+gain(i, 1)*err
  if(i. gt. n) go to 120
  if(mod(i, 2). eq. 1 .and. x(i). lt. 0.) x(i)=0.0
120 continue
c 更新後流出高
  qakal(ll)=0.0
  ist0=1

```



```

    iryu=nryu
    icn=nkad
    kry=0
    kka=0
    kkc=0
    do 306 i=1, 5
306 kkb(i)=0
    iat=ist0
    do 307 nks=nko, 1, -1
    call eqsys(nks, nm1, ist0, iat, iryu, icn, kry,
*          kka, kkb, kkc, kkd, ihh, p2h, ah, xh, x)
    if(kkd. ge. 1 .or. nm1. eq. 4) go to 307
    if(nm1. eq. 2) then
        iw=mwr(nks)
        p2h=1. 0
        ah=areak(nks)
        xh=wr(iw, ll, nl)*3. 6/ah          ! mm/hr
    end if
    qakal(ll)=qakal(ll)+xh**(1./p2h)*ah/suma  ! mm/hr
    if(mmod(nks). eq. 3) go to 308
307 continue
308 continue
    zq=qakal(ll)*suma /3. 6          ! m3/s
    sqq=sqrt(zq)
c   計算流量を水位変換するためのjjを計算する。
    jj=1
    if(nhc. le. 1) go to 6
    do 7 i=1, nhc-1
        jj=i
        if(zq. lt. qc(i)) go to 6
    7 continue
    jj=nhc
    6 continue

c
    zcal=bb0(jj)+bb1(jj)*sqq
    zakal(ll)=zcal

c
c   ..... update parameters ..... * 誤差分散行列の更新
c* (i-kh)*p
    call mati(nx, w1)
    call matmul(nx, 1, nx, gain, h, w2)
    call matsub(nx, nx, w1, w2, w3)
    call matmul(nx, nx, nx, w3, pp, w2)
c* p(+)
    do 111 i=1, nx
        do 111 j=1, nx
    111 pp(i, j)=w2(i, j)

c
c* モデル定数のオンライン更新(jconl=1)
    go to ( 21, 23, 21, 23 ), kase
    23 continue
    go to ( 21, 22 ) jconl+1
    22 continue
    ac1=x(i1)
    ac2=x(i1+1)

c
    21 continue
    return

```

```

end
C
C =====
C      subroutine shoki(qinit,x)                ! qinit=qq(1)
C      -----
C      initial values for implementing kalman filter
C      -----
C      include 'yosokufcj.cmn'
C      dimension x(253), ca(3)
C *****
C *          初期値誤差設定                      *
C *****
C      ..... initial values of state variables .....
C      x(1) : initial discharge (input)
C          ca(1)=ac1
C          ca(2)=ac2
C *
C      do 300 i=1, n      ! =(nryu+nkad)*2
C      x(i)=0.0
C      iii=(i-nryu*2-1)/2+1
C      gp2=zp2
C      if(i.gt.nryu*2) gp2=zp4(iii)
C      if(mod(i,2).eq.1) x(i)=qinit**gp2      ! mm/hr 全地点同じ値で問題なし
C
C      300 continue
C          call wrcal(1, n1, x)
C *
C      do 400 i=1, nx
C      do 400 j=1, nx
C      pp(i, j) = 0.0
C      if(j.eq.i) then
C          if(mod(j,2).eq.1) pp(i, j) = (x(j)*beta)**2.
C          if(mod(j,2).eq.0) pp(i, j) = pp(i-1, j-1)
C      end if
C      400 continue
C      if(n1.gt.0) then
C          k=0
C          do i=i1, i1+n1-1
C          k=k+1
C          pp(i, i)=(alph1*ca(k))**2.
C          if(alph1.gt.900.) pp(i, i)=alpc(k)**2.
C          end do
C      end if
C *      q(i, j) : system error covariance
C      do 420 i=1, n
C      do 420 j=1, n
C      q(i, j) = 0.0
C      420 continue
C      return
C      end
C
C =====
C      subroutine mati(n, a)
C * 単位行列の作成
C      dimension a(253, 253)
C      do 100 i=1, n
C      do 100 j=1, n
C      100 a(i, j)=0.

```

```

c
do 110 i=1, n
110 a(i, i)=1.0
return
end

c
c =====
subroutine matt(m, n, a, b)
c* 転値行列の作成
dimension a(253, 253), b(253, 253)
do 10 i=1, m
do 10 j=1, n
b(j, i)=a(i, j)
10 continue
return
end

c
c =====
subroutine matad(m, n, a, b, c)
c* 行列の和
dimension a(253, 253), b(253, 253), c(253, 253)
do 100 i=1, m
do 100 j=1, n
c(i, j)=a(i, j)+b(i, j)
100 continue
return
end

c
c =====
subroutine matc(m, n, a, c, b)
c* 行列の定数倍
dimension a(253, 253), b(253, 253)
do 100 i=1, m
do 100 j=1, n
100 b(i, j)=a(i, j)*c
return
end

c
c =====
subroutine matmul(l, m, n, a, b, c)
c* 行列の積
dimension a(253, 253), b(253, 253), c(253, 253)
do 10 i=1, l
do 20 j=1, n
c(i, j)=0.
do 30 k=1, m
c(i, j)=c(i, j)+a(i, k)*b(k, j)
30 continue
20 continue
10 continue
return
end

c
c =====
subroutine matsub(m, n, a, b, c)
c* 行列の差
dimension a(253, 253), b(253, 253), c(253, 253)
do 100 i=1, m

```

```

do 100 j=1, n
c(i, j)=a(i, j)-b(i, j)
100 continue
return
end

c
c =====
subroutine gosa
include 'yosokufcj. cmn'
dimension sa(300, 6), u(3), k(3), nrmse(3)
ko=1
uo=0. 0
do i=1, 3
k(i)=1
u(i)=0. 0
nrmse(i)=0
end do
do i=1, 300
do j=1, 6
sa(i, j)=0. 0
end do
end do

c
do 10 i=1, nq
if(uo. lt. hh0(i)) then
uo=hh0(i)
ko=i
endif
do 10 j=1, 3
if(u(j). lt. hf(i, j)) then
u(j)=hf(i, j)
k(j)=i+j
endif
10 continue
do 11 j=1, 3
do 11 i=j+1, nq
sa(i, j)=hh0(i)-hf(i-j, j)
if(hh0(i). ge. hlm) then ! hh0>=hlm のみでrmseを算定する
sa(i, j+3)=hh0(i)-hf(i-j, j)
nrmse(j)=nrmse(j)+1
endif
11 continue

c
do 22 j=1, 3
s1=0. 0
s2=0. 0
s3=0. 0
do 20 i=j+1, nq
s1=s1+sa(i, j)**2.
s2=s2+abs(sa(i, j)/hh0(i))
s3=s3+sa(i, j+3)**2.
20 continue
rmse(j)=sqrt(s1/float(nq-j))
rmse(j+3)=sqrt(s3/float(nrmse(j)))
ajre(j)=s2/float(nq-j)
ajpe(j)=(u(j)-uo)/uo
jtpe(j)=k(j)-ko
22 continue

```

```

return
end
c
c =====
subroutine qtoh (jj, zq, zh)
include 'yosokufcj. cmn'
jj=1
if(nhc. le. 1) go to 263
do 262 i=1, nhc-1
jj=i
if(zq. lt. qc(i)) go to 263
262 continue
jj=nhc
263 continue
zh=bb0(jj)+bb1(jj)*sqrt(zq)
return
end
c
c =====
c * 各地点の合流定式化と予測地点の観測方程式定式化
subroutine eqsys (nks, nm1, ist0, ist, iryu, icn, kry,
* kka, kkb, kkc, kkd, ihh, p2h, ah, xh, x)
include 'yosokufcj. cmn'
dimension x(253), kkb(5)
c
kkd=0
nm10=mmod(nks)/10+1
nm1=mmod(nks)-mmod(nks)/10*10
if(nm10. eq. ist) go to 200
if(nm10. lt. ist) go to 201
cx 次の次数の支川に移る。
kkb(ist)=kka
ist=ist+1
go to 200
cx 前の次数の支川に戻る。
201 continue
if(nm10. lt. ist0) go to 202
kkb(ist)=kka
ist=ist-1
kka=kkb(ist)
go to 200
202 continue
kkd=2
go to 300
200 continue
go to ( 221, 222, 223, 300 ) , nm1
221 continue ! 流域
kry=kry+1
if(kka. gt. 0) go to 298
kryu=iryu-kry+1
p2h=zp2
ah=area(kryu)
ihh=2*kryu-1
xh=x(ihh)
go to 300
222 continue ! 上流端流入量
if(kka. gt. 0) go to 298
cx iw=mwr(nks)

```

## 18program. for. txt

```

cx  p2h=1.0
cx  ah=areak(nks)
cx  xh=zr(iw)*3.6/ah          ! mm/hr
    go to 300
223 continue                ! 河道追跡&合流計算
    kkc=kkc+1
    if(kka.gt.0) go to 298
cx  if(kka.gt.0) go to 300
    kkad=icn-kkc+1
    kka=kka+1
    p2h=zp4(kkad)
    ah=areak(nks)
    ihh=2*(nryu+kkad)-1
    xh=x(ihh)
    go to 300
298 continue
    kkd=1
300 continue
    return
    end

c
c =====
c  subroutine wrcal(ll,k,x)
c  include 'yosokufcj.cmn'
c  dimension x(253)
c *** 全地点の流出量を計算し、wrに記憶する ***
    iryu=0
    icn=0
    do 940 nks=1,nko
    na1=mmod(nks)-mmod(nks)/10*10
    go to (221,940,223,223),na1
221 continue                ! 流域
    iw=mwr(nks)+1
    iryu=iryu+1
    k1=2*iryu-1
    wr(iw,ll,k)=x(k1)**wp2*area(iryu)/3.6
    go to 940
223 continue                ! 直下流が河道でない合流点または河
道
    iw=mwr(nks)
    s=0.0
    do 941 i=1,los(nks)
    jw=lwr(nks,i)
    s=s+wr(jw,ll,k)
941 continue
    wr(iw,ll,k)=s
    if(na1.eq.4) go to 940
    iw=mwr(nks)+1
    icn=icn+1
    k1=2*nryu+2*icn-1
    wr(iw,ll,k)=x(k1)**(1./zp4(icn))*areak(nks)/3.6
940 continue
    return
    end

```

H. 13.	9. 10	湧別川	丸瀬布	inputfc.txt		H. 12HQ				
10		3600	3.0	FC	複合流域	<--nl, DT, xrave, hiq				
	0. 6406	2. 5410		0. 5		<--F0 (CK1), Fc0 (CK2)				
2001	09	10	10			<--iy, im, id, ih				
111	111					<--nr, nq				
12						<--nko				
1	1	1	0	0	0	1-Ryu	130. 17	0.	0.	0.
2	1	3	0	0	0	2-Ryu	143. 80	0.	0.	0.
3	3	5	2	2	4	A-Kado	0. 0	20500.	1. 3834	0. 6765
4	1	7	0	0	0	3-Ryu	82. 94	0.	0.	0.
5	4	9	2	6	8	1-Gou	0. 0	0.	0.	0.
6	11	10	0	0	0	4-Ryu	280. 31	0.	0.	0.
7	13	11	1	11	0	B-Kado	0. 0	7300.	1. 5532	0. 6642
8	11	13	0	0	0	5-Ryu	44. 18	0.	0.	0.
9	14	15	2	12	14	2-Gou	0. 0	0.	0.	0.
10	4	16	2	9	15	3-Gou	0. 0	0.	0.	0.
11	1	17	0	0	0	6-Ryu	120. 60	0.	0.	0.
12	4	19	2	16	18	4-Gou	0. 0	0.	0. 0	0. 0
	1-R	1-Q	2-R	2-Q	A-IN	A-OT	3-R			
3-Q										
	1-Gou	4-R	B-IN	B-OT	5-R	5-Q	2-Gou			
3-Gou										
	6-R	6-Q	丸瀬布							
	1-リュウイキ	2-リュウイキ	3-リュウイキ	4-リュウイキ	5-リュウイキ	6-リュウイキ				
	0. 00	0. 00	0. 00	0. 00	0. 00	0. 00				
	0. 00	0. 00	0. 00	0. 00	0. 00	0. 00				
	0. 00	0. 00	0. 00	0. 00	0. 00	0. 00				
	0. 00	0. 00	0. 30	0. 00	0. 60	0. 21				
	0. 00	0. 00	0. 18	0. 50	0. 47	0. 79				
	0. 00	0. 00	0. 00	0. 00	0. 00	0. 00				
	0. 56	0. 14	1. 00	0. 66	1. 00	1. 00				
	4. 00	4. 00	4. 05	4. 00	4. 07	4. 39				
	4. 56	4. 14	3. 97	2. 64	2. 26	3. 40				
	5. 44	5. 86	3. 67	3. 32	1. 66	3. 19				
	2. 56	2. 14	2. 23	2. 15	1. 34	1. 80				
	3. 44	3. 86	2. 63	2. 83	2. 07	2. 79				
	3. 56	3. 14	3. 51	2. 65	2. 66	3. 40				
	8. 00	8. 00	8. 30	8. 00	8. 60	8. 21				
	7. 44	7. 86	7. 26	7. 85	7. 80	6. 42				
	7. 12	6. 28	7. 18	5. 30	5. 53	7. 98				
	7. 12	6. 28	7. 26	5. 31	5. 85	6. 61				
	4. 00	4. 00	4. 43	5. 01	5. 27	4. 21				
	5. 56	5. 14	5. 51	4. 65	4. 66	5. 40				
	6. 00	6. 00	5. 59	6. 00	5. 27	5. 00				
	5. 44	5. 86	4. 65	5. 34	4. 34	4. 40				
	5. 56	5. 14	5. 88	5. 16	5. 60	5. 61				
	6. 00	6. 00	5. 70	6. 00	5. 41	5. 79				
	5. 00	5. 00	4. 89	5. 00	4. 86	4. 21				
	3. 00	3. 00	3. 47	3. 51	4. 07	4. 00				
	4. 00	4. 00	4. 00	4. 00	4. 00	4. 00				
	4. 56	4. 14	5. 35	4. 66	5. 66	5. 60				
	6. 88	7. 72	6. 07	7. 18	6. 34	6. 00				
	6. 44	6. 86	6. 07	6. 84	6. 34	6. 00				
	6. 00	6. 00	6. 66	6. 51	7. 53	6. 42				
	1. 00	1. 00	1. 42	1. 51	2. 00	1. 60				
	1. 00	1. 00	1. 30	1. 00	1. 59	1. 21				







YosokuFcJ. cmn. txt

```

c** yosoku1. cmn **
common /var1/ n, n1, n2, nx, nc, i1, i2, i3
common /var2/ kase, jconl, nkse, jysr, iyoso, jkase (4)
common /st1/ wp1, wp2, zp2, ac1, ac2, ac3
common /st2/ hx, h2, h3, h4
common /st3/ alph1, alph2, alph3, beta, beta2, alpc (3), ap, bp
common /st5/ ab0 (8), ab1 (8), bb0 (8), bb1 (8)
common /st6/ hc (8), qc (8), nhc, h0
common /st7/ pp (253, 253), q (253, 253), ypr (24), yva (24)
common /st4/ fmx (253, 253), amx (253, 253), ax (253, 253)
common /st8/ f (253, 253), g (253, 253)
common /st9/ hlm, rmse (6), ajre (3), ajpe (3), jtpe (3) ! 06/01/21
common /outp/ hh0 (300), hf (300, 24), hu (300, 24), hl (300, 24)
*
common /outpq/ qf (300, 24), qu (300, 24), ql (300, 24), qbkal (300)
*
common /time/ kkm (300), kd (300), kh (300)
common /date/ iy0, im0, id0, ih0, nr, nq, nl
common /data2/ suma, area (50), arez (50), areak (350), dare (5)
common /discha/ r (51, 300), qz (300), wr (350, 300, 20), rave (50), qb
common /mod1/ nko, nryu, nqin, nkad, ngor, nall
common /mod2/ mmod (160), mwr (160), los (160), lwr (160, 2)
common /st10/ ck1, ck2, cc3, ramd, xrave, hiq, dt
common /st11/ cleg (50), alph (50), cmm (50)
common /st13/ vk3 (50), vk4 (50), vp3 (50), zp4 (50)
common /st14/ qd (10, 300), qdp (300, 10, 24), zr (350), rp (50, 300, 24)

```