Apéndice A

Algoritmo de Rosenbrock para variables no restringidas

A.1 Introducción

El inglés Howard Harry Rosenbrock en la década de los sesenta creó un método de optimización para encontrar el máximo o mínimo de funciones no lineales con variables restringidas y no restringidas. El método se basa en la técnica de búsqueda directa, en otras palabras, esta técnica no requiere derivadas de la función objetivo. Una de las aplicaciones en Ingeniería Civil correspondiente al área de hidrología, es por ejemplo: El cálculo de parámetros estadísticos de las distribuciones de probabilidad de una muestra de gastos máximos históricos, otra aplicación es el análisis del hidrograma unitario.

A.2 Procedimiento del algoritmo

El procedimiento supone una función unimodal y dependiente de múltiples variables no restringidas como:

$$F(x_1, x_2, x_3, \dots x_n)$$

Sea un conjunto de direcciones $E_n(e_1, e_2, e_3, \dots e_n)$ de vectores unitarios linealmente independientes y mutuamente ortogonales entre sí, paralelos a los ejes x_n, es decir que $v_i^T \cdot v_j = 0$ para $i \neq j$. Además de considerar un conjunto de longitudes de paso $P_n(\delta_1, \delta_2, \delta_3, \dots \delta_n)$. Los cambios sucesivos en los ejes x_n están dados por $\xi_n = e_n \cdot \delta_n$

Se supone un vector inicial de variables independientes x_n, un conjunto de direcciones E_n y un conjunto de longitudes de paso P_n. Después de cada evaluación en la función objetivo F, el nuevo valor $F_1 = x + \xi$ se compara con el mejor valor anterior F_0 obtenido hasta el momento. Si se desea encontrar un mínimo entonces $F_1(x + \xi) \leq F_0$, de modo que la solución se mueve hacia adelante $(x + \xi)$, es decir, el "punto actual" x_i se sustituye por

$(x+\xi)$ y el valor de F_0 se sustituye por $F_0(x+\xi)$. Si por otra parte $F_1(x+\xi) \geq F_0$, el actual punto x_i y el valor de F_0 se mantienen con sus valores originales.

Después de cada evaluación de F_i, la correspondiente longitud de paso δ_i se modifica según los siguientes criterios: Si la evaluación fue "exitosa", es decir, si $F_1(x+\xi) \leq F_0$ cuando se busca un mínimo, el valor de δ_i se multiplica por un factor de expansión $\alpha \geq 1$. Si la evaluación no tuvo éxito, o sea $F_1(x+\xi) \geq F_0$, δ_i se multiplica por un factor de contracción $(-\beta)$, para $0 < \beta \leq 1$. Rosenbrock (1960) dedujo experimentalmente que para dar a un buen rendimiento a un problema con dificultad, los factores estándar son $\alpha = 3.0$ y $\beta = -0.5$.

El efecto de este procedimiento es que si un cambio paralelo a la dirección e_i fue un éxito, entonces la próxima vez que se haga un paso en esta dirección será en el mismo sentido y tres veces más $(3.0 * \delta)$. Si el paso no tuvo éxito, entonces, cuando llegue el momento de utilizar e_i de nuevo, el paso se hará en el sentido contrario y sólo tendrá la mitad de la longitud $(0.5 * \delta)$.

Una secuencia de pruebas sin éxito con e_i, hace que δ_i disminuya de manera constante en magnitud y oscilar en el signo. Una secuencia de pruebas exitosas hace que δ_i crezca rápidamente, manteniendo el mismo signo. Si δ_i es en un principio extremadamente grande o demasiado pequeño, rápidamente se ajustará a la magnitud requerida.

Este procedimiento continúa hasta que una longitud de paso δ_i se ha realizado con éxito en la dirección de e_i, seguido de un fracaso, para cada evento i. Cuando esto ha sucedido, el procedimiento se interrumpe y los ejes $e_i = e_i^1$ se sustituyen por un nuevo conjunto de direcciones e_i^2 de vectores unitarios ortogonales obtenidos mediante el proceso de ortogonalización de ***Gram-Schmidt***.

Sea λ_i la suma de todas las longitudes de paso δ_i desde la última rotación de ejes. Entonces los e_i^2 se calculan de la siguiente manera:

$$\begin{array}{lccccc} A_1 = & \lambda_1 e_1^1 + & \lambda_2 e_2^1 + & \lambda_3 e_3^1 + & \cdots + & \lambda_n e_n^1 \\ A_2 = & & \lambda_2 e_2^1 + & \lambda_3 e_3^1 + & \cdots + & \lambda_n e_n^1 \\ \vdots & \vdots & & \vdots & \vdots & \vdots \\ A_n = & & & & & \lambda_n e_n^1 \end{array} \tag{A.1}$$

donde

A_1 vector que une al punto inicial y final obtenido con el conjunto de vectores unitarios $e_1^2, e_2^2, e_3^2, \ldots, e_n^2$

A_2 suma de todos los avances hechos en todas las direcciones excepto la primera

El nuevo conjunto de vectores unitarios ortogonales $e_1^1, e_2^1, e_3^1, \ldots, e_n^1$, se obtienen rotando los ejes coordenados de la siguiente manera:

$$B_1 = A_1 \tag{A.2}$$

$$e_1^2 = \frac{B_1}{\|B_1\|} \tag{A.3}$$

$$B_2 = A_2 - (A_2 \cdot e_1^2) \cdot e_1^2 \quad (A.4)$$

$$e_2^2 = \frac{B_2}{\|B_2\|} \quad (A.5)$$

$$\vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots$$

$$B_n = A_n - \sum_{i=1}^{n} (A_n \cdot e_i^2) \cdot e_i^2 \quad (A.6)$$

$$e_n^2 = \frac{B_n}{\|B_n\|} \quad (A.7)$$

A cada rotación de ejes se le llama ***etapa***. El proceso termina cuando algún criterio de convergencia se satisface.

A.3 Código del Algoritmo

Proponer:

Un punto inicial $X(x_1, x_2, \dots, x_n)$
Una tolerancia o criterio de convergencia $\varepsilon > 0$;
Un factor de expansión α, en el intervalo, $\alpha > 1$
Un factor de contracción β, en el intervalo, $0 \leq \beta \leq 1$
Las direcciones iniciales (vectores unitarios) $e_1, e_2, \dots\dots, e_n$
Las longitudes de paso iniciales $\delta_1, \delta_2, \dots\dots, \delta_n > 0$

```
!==========================================================================================
!===       OPTIMIZACION DE UNA FUNCION NO LINEAL MEDIANTE EL METODO DE ROSENBROCK      ====
!===                                                                                   ====
!=== Algoritmo de Rosenbrock para minimizar una función no lineal, unimodal y          ====
¡=== dependiente de múltiples variables no restringidas. FORTRAN 2003                  ====
¡===                                        Por Luis Eusebio Ramírez Salazar 2010  ====
!==========================================================================================

PROGRAM Optimizacion_fx
   IMPLICIT NONE

   INTEGER, PARAMETER        :: dim = 5
   REAL,DIMENSION(1:dim)     :: xi
   INTEGER                   :: in_reo,maxrot_ejes
   REAL                      :: tol,delta,falpha,fbeta
   OPEN(UNIT=100,FILE='Resultados.dat',STATUS='UNKNOWN',ACTION='WRITE')

   tol         = 1.0E-10
   delta       = 0.1
   falpha      = 3.0
   fbeta       = 0.5
   in_reo      = 3
   maxrot_ejes = 20

   xi(1) = 5.0
   xi(2) = -1.0
   xi(3) = 11.0
   xi(4) = -3.0
   xi(5) = 8.0

   CALL Optimizar_fx (dim,xi,tol,delta,falpha,fbeta,in_reo,maxrot_ejes)
```

```
CONTAINS

!**************************************************************************************
!*** Subrutina Optimizar_fx ()                                                      ***
!*** Esta subrutina calcula los valores óptimos xi de la función no lineal,          ***
¡*** utilizando el Algoritmo de Rosenbrock para variables no restringidas.          ***
!**************************************************************************************

SUBROUTINE Optimizar_fx (n,x,tole,delt,alpha,beta,in,max_rotejes)
   IMPLICIT NONE

   INTEGER, INTENT(IN)  :: n
   REAL, DIMENSION(1:n) :: x,deltta,cond,d
   REAL, DIMENSION(n,n) :: e,a
   INTEGER              :: i,j,k,l,m,p,r,h,ip,ix,jx,in,jn,neval_Fx,nrot_ejes,max_rotejes
   REAL                 :: tole,delt,alpha,beta,F0,F1,b,bety

   !Primera evaluación de la Función Objetivo, utilizando los valores propuestos xi.

   CALL Fx(n,x,F0)
   neval_Fx = 1
   nrot_ejes = 1

   !Se proponen valores de longitud de paso delta(i) y vectores unitarios e(i,j).

   DO i=1,n
      deltta(i) = delt
   END DO
   DO i=1,n
      DO j=1,n
         e(i,j) = 0.0
         IF (i==j) e(i,j) = 1.0
      END DO
   END DO

   !Impresión de las condiciones iniciales propuestas por el usuario.

   WRITE(100,05) ('=',i=1,62),'OPTIMIZACION DE UNA FUNCION NO LINEAL "METODO DE
                 ROSENBROCK"',('=',i=1,62)
   WRITE(100,08)
   WRITE(100,10) alpha,beta,tole
   WRITE(100,15)
   DO i=1,n
      WRITE(100,20) i,deltta(i)
   END DO
   WRITE(100,25)
   DO i=1,n
      WRITE(100,30) i,(e(i,j),j=1,n)
   END DO
   WRITE(100,35)
   DO i=1,n
      WRITE(100,40) i,x(i)
   END DO
   WRITE(100,45) F0
   WRITE(100,48) ('*',i=1,9),('*',i=1,9)

   !Comienza la optimización mediante el Algoritmo de Rosenbrock para variables
   no restringidas.

   DO WHILE (F0 >= tole)
      DO i=1,n
         cond(i) = 2.0
         d(i)    = 0.0
      END DO
```

```
99 DO i=1,n
      DO j=1,n
         x(j) = x(j) + deltta(i) * e(i,j)
      END DO
      CALL Fx(n,x,F1)
      neval_Fx = neval_Fx + 1
      IF (F1 <= F0) THEN
         d(i) = d(i) + deltta(i)
         deltta(i) = alpha * deltta(i)
         F0 = F1
         IF (cond(i) > 1.5) cond(i) = 1.0
      ELSE
         DO j=1,n
            x(j) = x(j) - deltta(i) * e(i,j)
         END DO
         deltta(i) = -beta * deltta(i)
         IF (cond(i) < 1.5) cond(i) = 0.0
      END IF
   END DO
DO jx=1,n
   IF(cond(jx) > 0.5) GO TO 99
END DO

!Rotación de ejes mediante el proceso de ortogonalización de Gram-Schmidt.

DO k=1,n
   DO l=1,n
      a(k,l) = 0.0
   END DO
END DO
DO h=1,n
   DO j=1,n
      DO l=h,n
         a(h,j) = a(h,j) + d(l) * e(l,j)
      END DO
   END DO
END DO
b = 0.0
DO j=1,n
   b = b + a(1,j)**2
END DO
b = SQRT(b)
DO j=1,n
   e(1,j) = a(1,j)/b
END DO
DO jn=1,in
   DO p=2,n
      ip = p-1
      DO m=p,n
         bety = 0.0
         DO k=1,n
            bety = bety - a(m,k) * e(ip,k)
         END DO
         DO j=1,n
            a(m,j) = a(m,j) + bety * e(ip,j)
         END DO
      END DO
      bety = 0.0
      DO k=1,n
         bety = bety + a(p,k)**2
      END DO
      bety = SQRT(bety)
      DO k=1,n
         e(p,k) = a(p,k)/bety
```

```
            END DO
         END DO
         IF (jn == in) EXIT
         DO r=2,n
            DO j=1,n
               a(r,j) = e(r,j)
            END DO
         END DO
      END DO

      !Impresión de resultados parciales.

      WRITE(100,50) nrot_ejes
      WRITE(100,55) neval_Fx
      WRITE(100,60) F0
      WRITE(100,65)
      DO ix=1,n
         WRITE(100,70) ix,x(ix)
      END DO

      nrot_ejes = nrot_ejes + 1
      IF (nrot_ejes >= max_rotejes) EXIT

   END DO

   !Impresión de resultados finales.

   WRITE(100,75) ('-',i=1,62),('-',i=1,62)
   WRITE(100,80) nrot_ejes
   WRITE(100,85) neval_Fx
   WRITE(100,90) F0
   WRITE(100,95)
   DO ix=1,n
      WRITE(100,97) ix,x(ix)
   END DO

   !Formatos para impresiones de resultados

   05 FORMAT(/,2X,62A1,/,3X,A60,/,2X,62A1,//,2X,'PARAMETROS INICIALES')
   08 FORMAT(/,2X,'f(x)=5*(X(1)-1)**2+4*(X(2)-2)**2+3*(X(3)-3)**2+2*(X(4)-4)**2+
            (X(5)-5)**2')
   10 FORMAT(/,2X,'Alpha =',F5.2,4X,'Beta =',F5.2,//,2X,'Tolerancia al error =',1PE8.1)
   15 FORMAT(/,2X,'Valores iniciales de longitud de paso :')
   20 FORMAT(/,2X,'delta',I2,X,'=',F4.1)
   25 FORMAT(/,2X,'Vectores unitarios iniciales ei:')
   30 FORMAT(/,2X,'e',I2,X,'= (',5F4.1,X,')')
   35 FORMAT(/,2X,'Valores iniciales de las variables independientes xi:')
   40 FORMAT(/,2X,'x',I2,X,'=',F10.3)
   45 FORMAT(/,2X,'Valor inicial de la función objetivo F(x) =',E16.6)
   48 FORMAT(//,2X,9A1,3X,'COMIENZA LA OPTIMIZACION DE LA FUNCION',3X,9A1)
   50 FORMAT(//,2X,'Etapa numero',I3)
   55 FORMAT(/,5X,'Numero de veces que se evaluó la función objetivo :',I4)
   60 FORMAT(/,5X,'Valor parcial de la función objetivo =',E16.6)
   65 FORMAT(/,5X,'Valores óptimos de las variables independientes ;')
   70 FORMAT(/,5X,'x',I2,X,'=',E16.6)
   75 FORMAT(//,2X,62A1,/,25X,'RESULTADOS FINALES',/,2X,62A1)
   80 FORMAT(/,2X,'Numero total de etapas :',I3)
   85 FORMAT(/,5X,'Numero de veces que se evaluó la función objetivo :',I4)
   90 FORMAT(/,5X,'Valor optimo de la función objetivo F(x) =',F16.6)
   95 FORMAT(/,5X,'Valores óptimos de las variables independientes ;')
   97 FORMAT(/,5X,'x',I2,X,'=',F16.5)

END SUBROUTINE Optimizar_fx
```

```
   !*************************************************************************************
   !*** Subrutina Fx()                                                                ***
   !*** Esta subrutina calcula el valor parcial de la Función Objetivo Fx             ***
   !*************************************************************************************

   SUBROUTINE Fx(n,x,F)
      IMPLICIT NONE

      INTEGER, INTENT(IN)     ::n
      REAL, DIMENSION(1:n)    ::x
      REAL                    ::F

      !Ingresar la función para minimizarla con el Algoritmo de Rosenbrock
      !Si el usuario desea maximizar la función, únicamente cambiar los signos

      F = (5*(X(1)-1)**2)+(4*(X(2)-2)**2)+(3*(X(3)-3)**2)+(2*(X(4)-4)**2)+((X(5)-5)**2)

   END SUBROUTINE Fx

END PROGRAM Optimizacion_fx
```

Referencias

A.1. Bazaraa, M.S., Shetty, C.M. ***"Nonlinear programming: Theory and algorithms"***. Wiley-Interscience, New Jersey, USA, 2006.

A.2. Beveridge, Gordon, S.G. ***"Optimization: Theory and practice"***. McGraw-Hill, New York, USA, 1970.

A.3. Bultheel, A. ***"Remark on Algorithm 450"***. Communication of the ACM. Vol. 17, No 8 (August 1974), 470.

A.4. Chapman, S.J. ***"FORTRAN 90/95 for Scientist and Engineers"***. McGraw-Hill Higher Education, Boston, USA, 2004.

A.5. Davies, M.A ***"Remark on Algorithm 450"***. ACM Transactions on Mathematical Software. Vol. 2, No 3 (September 1976), 300-301.

A.6. Himmelblau, D.M. ***"Applied nonlinear programming"***. McGraw-Hill, New York, USA, 1972.

A.7. Kuester, J.L., Mize.J.H. ***"Optimization techniques with FORTRAN"***. McGraw-Hill, New York, USA, 1973.

A.8. Machura, M., Mulawa, A. ***"Rosenbrock function minimization"***. Communication of the ACM. Vol. 16, No 8 (August 1973), 482-483.

A.9. Martinez, B.J., Requena.R.I., Marín, R.N ***"Programación estructurada con FORTRAN 90/95"***. Universidad de Granada, España, 2006.

A.10. Mordecai, A. ***"Nonlinear programming: Analysis and methods"***. Prentice-Hall, New Jersey, 1976.

A.11. Rojas, V.F.J. ***"Aplicación del método de Rosenbrock en la optimización de parámetros hidrológicos"***. Tesis de Licenciatura Facultad de Ingeniería, UNAM, 1978.

A.12. Rosenbrock, H.H. ***"An automatic method for finding the greatest or least value of a function"***. The Computer Journal. Vol. 3, No 3 (March 1960), 175-184.

A.13. Rosenbrock, H.H., Storey, C. ***"Computational techniques for chemical engineers"***. Pergamon, Oxford, UK, 1966.

A.14. Schwefel, H.P. ***"Evolution and optimum seeking"***. John-Wiley, New York, USA, 1994.

Apéndice B

Ajuste de una serie de gastos máximos anuales por medio de la función de distribución de probabilidad "Gumbel Dos Poblaciones"

```
!==========================================================================================
!===             AJUSTE DE UNA SERIE DE GASTOS MEDIOS MAXIMOS ANUALES MEDIANTE         ===
!===        LA FUNCION DE DISTRIBUCION DE PROBABILIDAD "GUMBEL DOS POBLACIONES" (G2P)   ===
!===                                                                                    ===
!=== Este programa calcula la magnitud de un cierto Gasto Máximo Anual asociado a un    ===
!=== periodo de retorno Tr, por medio del ajuste de una función de probabilidad G2P ó   ===
!=== Gumbel Mixta. La estimación de parámetros estadísticos de dicha función se         ===
!=== realiza con el algoritmo de Rosenbrock para minimizar una función no lineal,       ===
!=== unimodal y dependiente de múltiples variables no restringidas. Para estimar la     ===
!=== extrapolación de eventos de diseño para diferentes periodos de retorno se emplea   ===
!=== el Método Numérico de Newton-Raphson.                                              ===
!===                                                                                    ===
!=== Lenguaje de programación utilizado: FORTRAN 2003.                                  ===
!=== Elaborado por: Luis Eusebio Ramírez Salazar 2010 (Tesis de Licenciatura UNAM).     ===
!==========================================================================================


PROGRAM Eventos_Qd
   IMPLICIT NONE

   INTEGER, PARAMETER                :: dimqff = 150, dimqc = 2
   INTEGER, PARAMETER                :: dimp   = 5
   INTEGER, PARAMETER                :: dalf   = 11
   INTEGER, PARAMETER                :: dTr    = 12
   REAL, DIMENSION(1:dimqff)         :: Qi,Qr,P,Tre,Z,Qc,fQc
   REAL, DIMENSION(1:dimp)           :: pi,x
   REAL, DIMENSION(1:dalf)           :: fa
   REAL, DIMENSION(1:dTr)            :: Tr,QTr,fQTr
   REAL, DIMENSION(dimqff,dimqc)     :: QA
   INTEGER                           :: i,j,k,dimqf,nqc,nre,nmre,A1,A2,Salida
   REAL                              :: tol,delta,falpha,fbeta,fQ,Media1,Media2,DesEst1,&
                                        DesEst2,sumq,EEA
   CHARACTER(LEN=30)                 :: Estación
   OPEN(UNIT=100,FILE='Registros.dat',STATUS='UNKNOWN',ACTION='READ')
```

```
!Condiciones iniciales para optimizar la Función G2P con el Método de Rosenbrock.

tol    = 1.0E-10
delta  = 0.1
fa     = (/2.0,2.1,2.2,2.3,2.4,2.5,2.6,2.7,2.8,2.9,3.0/) ! falpha >= 1
fbeta  = 0.5                                             ! 0 < fbeta <= 1
nre    = 3

!Ingreso de información general.
!------------------------------

WRITE(*,01) ("=",i=1,87),("=",i=1,87)
WRITE(*,02)
WRITE(*,03)
WRITE(*,04)
WRITE(*,05)
READ(*,*)   Estación
WRITE(*,06)
READ(*,'(I4)') A1
WRITE(*,07)
READ(*,'(I4)') A2
WRITE(*,08)
READ(*,*)   dimqf
WRITE(*,09)
READ(*,*)   nqc
WRITE(*,10)
READ(*,*)   nmre
CALL SYSTEM('CLS')
WRITE(*,11)
PAUSE
CALL SYSTEM('CLS')

!Lectura del registro de Gastos Máximos de la estación hidrométrica requerida.
!------------------------------------------------------------------------------

DO j=1,dimqc
   DO i=1,dimqf
      READ(100,*) QA(i,j)
   END DO
END DO
CLOSE(UNIT=100)

DO k=1,dimqf
   Qi(k)=QA(k,1)
END DO

!IMPRESION DE RESULTADOS FINALES.
!--------------------------------

WRITE(*,12) ("=",i=1,87),("=",i=1,87)
WRITE(*,13)
WRITE(*,14)
WRITE(*,15)
WRITE(*,16) Estación
WRITE(*,17) A1
WRITE(*,18) A2
WRITE(*,19) dimqf
WRITE(*,20) nqc
```

```
!Impresión de parámetros estadísticos iniciales de la Función G2P.
!-----------------------------------------------------------------

WRITE(*,21) ("-",k=1,87),("-",k=1,87)

CALL Estadistica (dimp,dimqf,pi,Qi,nqc,Media1,Media2,DesEst1,DesEst2)

WRITE(*,22)
WRITE(*,'(/,5X,"Media1  =",F10.3)') Media1
WRITE(*,'(/,5X,"DesEst1 =",F10.3)') DesEst1
WRITE(*,'(/,5X,"Media2  =",F10.3)') Media2
WRITE(*,'(/,5X,"DesEst2 =",F10.3)') DesEst2
WRITE(*,23)
DO i=1,dimp
   WRITE(*,'(/,5X,"p",I1,1X,"=",F10.3)') i,pi(i)
END DO
CALL Funcion_fQ (dimp,dimqf,pi,Qi,fQ)
WRITE(*,24) fQ                          !Evaluación inicial de la Función Objetivo f(Q).

DO i=1, dimp
   x(i)=pi(i)                           ! Parámetros estadísticos iniciales.
END DO
DO j=1,dimqf
   P(j)=REAL(j)/(dimqf+1)               !Probabilidad P(Qr).
END DO
DO k=1,dimqf
   Tre(k)=1/(1-P(k))                    !Periodo de retorno Tr.
END DO
DO i=1,dimqf
   Z(i)=-LOG(LOG(Tre(i)/(Tre(i)-1)))   !Abscisa Z para bosquejar la grafica Z vs (Qr,Qc)
END DO

!Impresión de parámetros estadísticos óptimos de la Función G2P.
!---------------------------------------------------------------

WRITE(*,25) ("-",k=1,87),("-",k=1,87)
CALL MinEEA (dimp,dimqf,dalf,pi,Qi,fa,P,tol,delta,falpha,fbeta,nre,nmre)
WRITE(*,26) falpha
WRITE(*,27) fbeta
DO k=1, dimp
pi(k)= x(k)
END DO
CALL Optimizar_fQ (dimp,dimqf,pi,Qi,tol,delta,falpha,fbeta,nre,nmre)
CALL Funcion_fQ (dimp,dimqf,pi,Qi,fQ)

WRITE(*,28)
DO i=1,dimp
   WRITE(*,29) i,pi(i)
END DO
WRITE(*,30) fQ     !Evaluación final de la Función Objetivo f(Q).

!Ajuste a la serie Qi con la distribución G2P e impresión de tabla de cálculo final.
!-----------------------------------------------------------------------------------

WRITE(*,31) ("-",j=1,87),("-",j=1,87)

CALL Ajuste_G2P (dimp,dimqf,pi,Qi,P,Qc,fQc)

WRITE(*,32)
WRITE(*,33) ("=",i=1,4),("=",i=1,9),("=",i=1,3),("=",i=1,9),("=",i=1,5),&
            ("=",i=1,8),("=",i=1,6),("=",i=1,10),("=",i=1,5)
DO k=1,dimqf
   WRITE(*,34) INT(QA(k,2)),QA(k,1),K,Qr(k),P(k),Tre(K),Z(K),Qc(k),fQc(k)
END DO
WRITE(*,35) ("-",k=1,87)
```

```
sumq=0.0
DO i=1,dimqf
   sumq=sumq+(Qr(i)-Qc(i))**2
END DO
EEA=SQRT(sumq/(dimqf-dimp))
WRITE(*,36) EEA

!Extrapolación probabilística de los eventos de diseño Qc, para distintos periodos de retorno Tr.
!-----------------------------------------------------------------------------------

CALL Extr_GastDis (dimp,dimqf,dTr,pi,Qc,Tr,QTr,fQTr)

WRITE(*,37)
WRITE(*,38)
WRITE(*,39) ("=",i=1,8),("=",i=1,10),("=",i=1,5)
DO j=1,dTr
   WRITE(*,40) INT(Tr(j)),QTr(j),fQTr(j)
END DO
WRITE(*,41) ("*",i=1,10),("*",i=1,10)
WRITE(*,*)  'Presiona 1+Enter para salir del programa QG2P'
READ(*,*)   Salida

!Formatos de impresión.
!----------------------

01 FORMAT(/,87A1,/,12X,'AJUSTE DE UNA SERIE DE GASTOS MEDIOS MAXIMOS ANUALES MEDIANTE'&
           ,/,7X,'LA FUNCION DE DISTRIBUCION DE PROBABILIDAD "GUMBEL DOS POBLACIONES"
           (G2P)',/,87A1)
02 FORMAT(/,'Lenguaje de Programación: FORTRAN 2003',/&
           ,/,'Ajuste de Distribución de Probabilidad: Gumbel Dos Poblaciones (Gonzales
           1970)')
03 FORMAT(/,'f(Q) = (exp(-exp(-((Q-b1)/a1))))*(p+((1-p)*(exp(-exp(-((Q-b2)/a2))))))')
04 FORMAT(/,'** INGRESE LA INFORMACION REQUERIDA')
05 FORMAT(/,'Nombre de la Estación Hidrométrica: (30 caracteres)')
06 FORMAT(/,'Año de registro inicial:')
07 FORMAT(/,'Año de registro final:')
08 FORMAT(/,'Tamaño del registro de Gastos Máximos Anuales')
09 FORMAT(/,'Número de Gastos Máximos Ciclónicos')
10 FORMAT(/,'Número máximo de rotación de ejes (nmre = 20)')
11 FORMAT(/,2X,'*** PRECAUCION !!!',/&
           ,/,2X,'Para que se pueda ejecutar este programa es necesario generar un
           archivo que tenga'&
           ,/,2X,'por nombre "Registros" con extensión .dat, pegarlo en la misma carpeta
           en donde'&
           ,/,2X,'se localice este archivo ejecutable .exe. El archivo generado
           Registros.dat, for-'&
           ,/,2X,'zosamente tendrá que contener la siguiente información y orden que se
           indica;',/&
           ,/,2X,'Escribir o pegar en el archivo .dat, los Gastos Máximo Anuales
           (originales, sin'&
           ,/,2X,'ordenar) en forma de columna. Dejar un espacio al termino de dicha
           columna y pos-'&
           ,/,2X,'teriormente pegar las fechas (anos) de registro: véase el siguiente
           ejemplo',/&
           ,/,2X,'Q1',/,2X,'Q2',/,2X,'Qn',/,2X,'espacio',/,2X,'1945',/,2X,'1946',/,2X,'2
           010',/&
           ,/,2X,'Es aconsejable generar primero el archivo .dat, para evitar errores
           fatales.'&
           ,/,2X,'Si ya genero el archivo .dat presione ENTER para continuar.',/)
 12 FORMAT(/,2X,87A1,/,13X,'AJUSTE DE UNA SERIE DE GASTOS MEDIOS MAXIMOS ANUALES
           MEDIANTE'&
           ,/,8X,'LA FUNCION DE DISTRIBUCION DE PROBABILIDAD "GUMBEL DOS
           POBLACIONES"(G2P)',/,2X,87A1)
13 FORMAT(/,5X,'Este programa calcula la magnitud de un cierto Gasto Máximo Anual
```

```
            asociado a un'&
            ,/,5X,'periodo de retorno Tr, por medio del ajuste de una función de
            probabilidad G2P'&
            ,/,5X,'o Gumbel Mixta. La estimación de parámetros estadísticos de dicha
            función se '&
            ,/,5X,'realiza con el algoritmo de Rosenbrock para minimizar una función no
            lineal,'&
            ,/,5X,'unimodal y dependiente de múltiples variables no restringidas. Para
            estimar'&
            ,/,5X,'los gastos máximos anuales o eventos de diseño y las extrapolaciones
            para dife-'&
            ,/,5X,'rentes periodos de retorno se emplea el Método Numérico de Newton
            Raphson.')
14 FORMAT(/,5X,'Lenguaje de Programación: FORTRAN 2003',/&
            ,/,5X,'Ajuste de Distribución de Probabilidad: Gumbel Dos Poblaciones
            (Gonzales 1970)')
15 FORMAT(/,5X,'f(Q) = (exp(-exp(-((Q-p2)/p1))))*(p5+((1-p5)*(exp(-exp(-((Q-p4)&
            /p3))))))')
16 FORMAT(/,5X,'Nombre de la Estación Hidrometrica:',1X,A30)
17 FORMAT(/,5X,'Año de registro inicial:',1X,I4)
18 FORMAT(/,5X,'Año de registro final:',1X,I4)
19 FORMAT(/,5X,'Tamano del registro de Gastos Máximo Anuales; dimqf =',1X,I3)
20 FORMAT(/,5X,'Numero de Gastos Máximo Ciclónicos; nqc =',1X,I2)
21 FORMAT(/,2X,87A1,/,17X,"ESTIMACION DE PARAMETROS ESTADISTICOS INICIALES &
            (X MOMENTOS)",/,2X,87A1)
22 FORMAT(/,5X,"Estadisticos muestrales;")
23 FORMAT(/,5X,"Parametros estadísticos de la función Gumbel dos Poblaciones;")
24 FORMAT(/,5X,"Valor inicial de la función objetivo f(Q) =",F10.6)
25 FORMAT(/,2X,87A1,/,11X,"OPTIMIZACION DE DE PARAMETROS ESTADISTICOS &
            (ALGORITMO DE ROSENBROCK)",/,2X,87A1)
26 FORMAT(/,5X,'Factor de escala de aumento en los incrementos;   a =',1X,F3.1)
27 FORMAT(/,5X,'Factor de escala de reducción en los incrementos; b =',1X,F3.1)
28 FORMAT(/,5X,"Parametros estadísticos óptimos de la función Gumbel dos Poblaciones;")
29 FORMAT(/,5X,"p",I1,1X,"=",F10.3)
30 FORMAT(/,5X,"Valor optimo de la función objetivo f(Q) =",F10.6)
31 FORMAT(/,2X,87A1,/,15X,"CALCULO DE GASTOS MAXIMOS DE DISENO Qc(m3/s) &
            (Newton Raphson)",/,2X,87A1)
32 FORMAT(/,T5,"Ano",T12,1X,"Q(m3/s)",T24,1X,"k",T30,1X,"Qr(m3/s)",T42,"P(Qr)",&
            T50,"Tr(anos)",T61,2X,"Z",T70,1X,"Qc(m3/s)",T83,"F(Qc)")
33 FORMAT(T5,4A1,T12,9A1,T24,3A1,T30,9A1,T42,5A1,T50,8A1,T61,6A1,T70,10A1,T83,5A1,/)
34 FORMAT(T5,I4,T12,F9.2,T24,I3,T30,F9.2,T42,F5.3,T50,F7.3,T61,F6.3,T70,F10.3,T83,F5.3)
35 FORMAT(2X,87A1)
36 FORMAT(/,4X,"Error Estándar de Ajuste   EEA =",F10.3,1X,"m3/s")
37 FORMAT(/,4X,"Extrapolacion probabilística de los eventos de diseño Qc")
38 FORMAT(/,T5,"Tr(años)",T16,1X,"Qc(m3/s)",T29,"F(Qc)")
39 FORMAT(T5,8A1,T16,10A1,T29,5A1,/)
40 FORMAT(T5,1X,I5,T16,F10.3,T29,F5.3)
41 FORMAT(//,2X,10A1,5X,"Programa elaborado por RAMIREZ SALAZAR &
            LUIS EUSEBIO 2010",5X,10A1,/)


CONTAINS

!****************************************************************************************
!*** Subrutina Estadística ()                                                         ***
!*** Esta subrutina calcula el análisis estadístico; Estadísticos muestrales de       ***
¡*** Gastos Q, y Estimación de los parámetros estadísticos por momentos de la         ***
¡*** Función de Distribución G2P.                                                     ***
!****************************************************************************************

SUBROUTINE Estadistica (dp,dqf,p,Q,nqci,Media_1,Media_2,DesEst_1,DesEst_2)
   IMPLICIT NONE

   INTEGER, INTENT(IN)     :: dp,dqf,nqci
   REAL, DIMENSION(1:dqf)  :: Q
```

```
   REAL, DIMENSION(1:dp)   :: p
   INTEGER                 :: i,j,k,r
   REAL                    :: temp,sumx1,sumx2,sums1,sums2,Media_1,Media_2,&
                              DesEst_1,DesEst_2

   DO i=1,dqf-1
      DO j=i+1,dqf
         IF (Q(i)>Q(j)) THEN
            temp=Q(i)
            Q(i)=Q(j)
            Q(j)=temp
         END IF
      END DO
   END DO

   sumx1=0.0
   DO i=1,dqf-nqci
      sumx1=sumx1+Q(i)
   END DO
   Media_1=sumx1/(dqf-nqci)

   sumx2=0.0
   DO j=dqf-nqci+1,dqf
      sumx2=sumx2+Q(j)
   END DO
   Media_2=sumx2/nqci

   sums1=0.0
   DO k=1,dqf-nqci
      sums1=sums1+(Q(k)-Media_1)**2
   END DO
   DesEst_1=SQRT(sums1/((dqf-nqci)-1))

   sums2=0.0
   DO r=dqf-nqci+1,dqf
      sums2=sums2+(Q(r)-Media_2)**2
   END DO
   DesEst_2=SQRT(sums2/(nqci-1))

   p(1) = 0.779697*DesEst_1              !Alfa1
   p(2) = Media_1-(0.450053*DesEst_1)    !Beta1
   p(3) = 0.779697*DesEst_2              !Alfa2
   p(4) = Media_2-(0.450053*DesEst_2)    !Beta2
   p(5) = (REAL(dqf)-nqci)/dqf           !P

END SUBROUTINE Estadistica

!*****************************************************************************************
!*** Subrutina Funcion_fQ ()                                                          ***
!*** Esta subrutina calcula el valor parcial de la Función Objetivo f(Q)              ***
!*****************************************************************************************

SUBROUTINE Funcion_fQ (dp,dqf,p,Q,FO)
   IMPLICIT NONE

   INTEGER, INTENT(IN)     :: dp,dqf
   REAL, DIMENSION(1:dp)   :: p
   REAL, DIMENSION(1:dqf)  :: Q
   INTEGER                 :: i
   REAL                    :: a1,a2,E,FO

   E = 0.0
   DO i=1,dqf
      a1 = EXP(-EXP(-((Q(i)-p(2))/p(1))))
```

```
      a2 = EXP(-EXP(-((Q(i)-p(4))/p(3))))
      E  = E+(((REAL(i)/(dqf+1))-(a1*(p(5)+((1-p(5))*a2))))**2)*1.0
   END DO
   FO = E

END SUBROUTINE Funcion_fQ

!*************************************************************************************
!*** Subrutina Optimizar_fQ ()                                                    ***
!*** Esta subrutina calcula los parámetros estadísticos óptimos p(i)              ***
!*** utilizando el algoritmo de Rosenbrock para variables no restringidas.        ***
!*************************************************************************************

SUBROUTINE Optimizar_fQ (dp,dqf,p,Q,tole,delt,alpha,beta,nr,max_rotejes)
   IMPLICIT NONE

   INTEGER, INTENT(IN)     :: dp,dqf
   REAL, DIMENSION(1:dp)   :: p,deltta,cond,d
   REAL, DIMENSION(1:dqf)  :: Q
   REAL, DIMENSION(dp,dp)  :: e,a
   INTEGER                 :: i,j,k,m,r,s,t,ni,nr,nrot_ejes,max_rotejes
   REAL                    :: tole,delt,alpha,beta,F0,F1,b,bb

   !Primera evaluación de la Función Objetivo, utilizando los valores propuestos xi.

   CALL Funcion_fQ (dp,dqf,p,Q,F0)

   nrot_ejes = 1

   !Se proponen valores de longitud de paso delta(i) y vectores unitarios e(i,j).

   DO i=1,dp
      deltta(i) = delt
   END DO
   DO j=1,dp
      DO k=1,dp
         e(j,k) = 0.0
         IF (j==k) e(j,k) = 1.0
      END DO
   END DO

   !Comienza la optimización mediante el método de Rosenbrock para variables no
   !restringidas.

   DO WHILE (F0 >= tole)
      DO i=1,dp
         cond(i) = 2.0
         d(i)    = 0.0
      END DO
      90 DO j=1,dp
            DO k=1,dp
               p(k) = p(k) + deltta(j) * e(j,k)
            END DO

            CALL Funcion_fQ (dp,dqf,p,Q,F1)

            IF (F1 <= F0) THEN
               d(j) = d(j) + deltta(j)
               deltta(j) = alpha * deltta(j)
               F0 = F1
               IF (cond(j) > 1.5) cond(j) = 1.0
            ELSE
               DO m=1,dp
                  p(m) = p(m) - deltta(j) * e(j,m)
               END DO
```

```
               deltta(j) = -beta * deltta(j)
               IF (cond(j) < 1.5) cond(j) = 0.0
            END IF
         END DO
      DO r=1,dp
         IF(cond(r) > 0.5) GO TO 90
      END DO

      !Rotación de ejes mediante el proceso de ortogonalización de Gram-Schmidt.

      DO i=1,dp
         DO j=1,dp
            a(i,j) = 0.0
         END DO
      END DO
      DO k=1,dp
         DO m=1,dp
            DO r=k,dp
               a(k,m) = a(k,m) + d(r) * e(r,m)
            END DO
         END DO
      END DO
      b = 0.0
      DO s=1,dp
         b = b + a(1,s)**2
      END DO
      b = SQRT(b)
      DO t=1,dp
         e(1,t) = a(1,t)/b
      END DO
      DO ni=1,nr
         DO i=2,dp
            j = i-1
            DO k=i,dp
               bb = 0.0
               DO m=1,dp
                  bb = bb - a(k,m) * e(j,m)
               END DO
               DO r=1,dp
                  a(k,r) = a(k,r) + bb * e(j,r)
               END DO
            END DO
            bb = 0.0
            DO s=1,dp
               bb = bb + a(i,s)**2
            END DO
            bb = SQRT(bb)
            IF (bb == 0.0) EXIT
            DO t=1,dp
               e(i,t) = a(i,t)/bb
            END DO
         END DO
         IF (ni == nr) EXIT
         DO i=2,dp
            DO j=1,dp
               a(i,j) = e(i,j)
            END DO
         END DO
      END DO
      nrot_ejes = nrot_ejes + 1
      IF (nrot_ejes >= max_rotejes) EXIT
   END DO

END SUBROUTINE Optimizar_fQ
```

```
!***************************************************************************************
!*** Subrutina Ajuste_G2P ()                                                        ***
!*** Esta subrutina calcula los Gastos Máximos de Diseño Qc(m3/s) para              ***
!*** diferentes periodos de retorno Tr(anos), utilizando el método de Newton-Raphson. ***
!***************************************************************************************

SUBROUTINE Ajuste_G2P (dp,dqf,p,Q,PQ,Qcal,fQcal)
   IMPLICIT NONE

   INTEGER, INTENT(IN)     :: dp,dqf
   REAL, DIMENSION(1:dp)   :: p
   REAL, DIMENSION(1:dqf)  :: Q,PQ,Qcal,fQcal
   INTEGER                 :: i
   REAL                    :: tol,Gast,Prob

   Tol = 1.0E-07

   DO i=1,dqf
      Gast = Q(i)
      Prob = PQ(i)
      DO WHILE (ABS(fQ_NR(dp,p,Gast,Prob)) >= tol)
         Gast = Gast - fQ_NR(dp,p,Gast,Prob)/dfQ(dp,p,Gast)
      END DO
      Qcal(i)  = Gast
      fQcal(i) = fQ_NR(dp,p,Gast,Prob)
   END DO

END SUBROUTINE Ajuste_G2P

!**************************************************************************
!** Función fQ_NR()                                                      **
!** "Esta función calcula el valor parcial de la función objetivo f(Qc)   **
!**************************************************************************

REAL FUNCTION fQ_NR (dp,p,Gast,Prob)
   IMPLICIT NONE

   REAL, INTENT(IN):: Gast,Prob
   INTEGER         :: dp
   REAL            :: a,b,p(dp)

   a = EXP(-EXP(-((Gast-p(2))/p(1))))
   b = EXP(-EXP(-((Gast-p(4))/p(3))))

   fQ_NR = (a*(p(5)+((1-p(5))*b))) - Prob

END FUNCTION fQ_NR

!**************************************************************************
!** Función dfQ()                                                        **
!** "Esta función calcula el valor la derivada de la función f(Qc)"       **
!**************************************************************************

REAL FUNCTION dfQ (dp,p,Gast)
   IMPLICIT NONE

   REAL, INTENT(IN):: Gast
   INTEGER         :: dp
   REAL            :: a,b,c,d,p(dp)

   a = EXP(-EXP(-((Gast-p(2))/p(1))))
   b = EXP(-EXP(-((Gast-p(4))/p(3))))
   c = EXP(-((Gast-p(2))/p(1)))
   d = EXP(-((Gast-p(4))/p(3)))
```

```
   dfQ = a*(((p(5)/p(1))*c)+(((1-p(5))/(p(1)*p(3)))*b*((p(3)*c)+(p(1)*d))))

END FUNCTION dfQ

!****************************************************************************************
!*** Subrutina MinEEA ()                                                             ***
!*** Esta subrutina estima el Mínimo Error Estándar de Ajuste                        ***
!*** MIM(EEA) y posteriormente                                                       ***
!*** calcula los factores óptimos a y b para el Algoritmo de Rosenbrock.             ***
!****************************************************************************************

SUBROUTINE MinEEA (dp,dqf,dal,p,Q,fal,PQ,tole,delt,alfa,beta,nr,max_rotejes)
   IMPLICIT NONE

   INTEGER, INTENT(IN)     :: dp,dqf,dal
   REAL, DIMENSION(1:dp)   :: p,x
   REAL, DIMENSION(1:dqf)  :: Q,PQ,Qcal,fQcal
   REAL, DIMENSION(1:dal)  :: fal,EE
   INTEGER                 :: i,j,k,m,r,nr,max_rotejes
   REAL                    :: tole,delt,alfa,beta,alpha,sq,Minimo

   sq=0.0
   DO i=1,dp
      x(i)=p(i)
   END DO

   DO j=1,dal
      DO k=1,dp
         p(k)=x(k)
      END DO
      alpha=fal(j)
      CALL Optimizar_fQ (dp,dqf,p,Q,tole,delt,alpha,beta,nr,max_rotejes)
      CALL Ajuste_G2P (dp,dqf,p,Q,PQ,Qcal,fQcal)
      DO m=1,dqf
         sq=sq+(Q(m)-Qcal(m))**2
      END DO
      EE(j)=SQRT(sq/(dqf-dp))
      sq=0.0
   END DO
   Minimo = MIN(EE(1),EE(2),EE(3),EE(4),EE(5),EE(6),EE(7),EE(8),EE(9),EE(10),EE(11))
   DO r=1,dal
      IF(Minimo == EE(r)) EXIT
   END DO
   alfa = fal(r)

END SUBROUTINE MinEEA


!*****************************************************************************
!** Subrutina Extr_GastDis ()                                               **
!** Esta subrutina extrapola los eventos de diseño para diferentes Periodos  **
!** de Retorno TR = 2,5,10,20,50,100,200,500,1000,2000,5000,10000 (anos)    **
!*****************************************************************************

SUBROUTINE Extr_GastDis (dp,dqf,dT,p,Qcal,T,QT,fQT)
   IMPLICIT NONE

   INTEGER, INTENT(IN)     :: dp,dqf,dT
   REAL, DIMENSION(1:dp)   :: p
   REAL, DIMENSION(1:dT)   :: T,QT,fQT,PQ
   REAL, DIMENSION(1:dqf)  :: Qcal
   INTEGER                 :: i,j,k
   REAL                    :: tol,Gast,Prob
```

```
      tol  = 1.0E-07
      IF (MOD(dqf,2)==0) THEN
         k=dqf/2      !Tr *= 2 años
      ELSE
         k=(dqf+1)/2  !Tr  = 2 años
      END IF
      Gast = Qcal(k)

      T = (/2,5,10,20,50,100,200,500,1000,2000,5000,10000/)

      DO i=1,dT
         PQ(i)=1-(1/REAL(T(i)))
      END DO

      DO j=1,dT
         Prob = PQ(j)
         DO WHILE (ABS(fQ_NR(dp,p,Gast,Prob)) >= tol)
            Gast = Gast - fQ_NR(dp,p,Gast,Prob)/dfQ(dp,p,Gast)
         END DO
         QT(j)  = Gast
         fQT(j) = fQ_NR(dp,p,Gast,Prob)
      END DO

   END SUBROUTINE Extr_GastDis

END PROGRAM Eventos_Qd
```

Referencias

B.1. Campos A. D.F., ***"Estimación de los parámetros óptimos de la distribución Gumbel mixta por medio del algoritmo de Rosenbrock"***. Ingeniería Hidráulica en México. Vol. IV No. 1, enero-abril, 1989. pp 9-18.

B.2. Chapman, S.J. ***"FORTRAN 90/95 for Scientist and Engineers"***. McGraw-Hill Higher Education, Boston, USA, 2004.

B.3. Davies, M.A ***"Remark on Algorithm 450"***. ACM Transactions on Mathematical Software. Vol. 2, No 3, september 1976, pp 300-301.

B.4. Kuester, J.L., Mize.J.H. ***"Optimization techniques with FORTRAN"***. McGraw-Hill, New York, USA, 1973.

B.5. Martinez, B.J., Requena.R.I., Marín, R.N ***"Programación estructurada con FORTRAN 90/95"***. Universidad de Granada, España, 2006.

B.6. V. Balderrama. ***"Métodos Numéricos"***. Trillas, México, 1990.

Apéndice C

Componentes de un vaso de almacenamiento de una presa

C.1. Introducción

Una cuenca es una zona de la superficie terrestre tal que, si fuera impermeable, todas las gotas de lluvia que caen sobre ella tienden a ser drenadas hacia un punto de salida, una cuenca se delimita por un parteaguas. Un embalse es una ampliación del valle por donde escurre una corriente, susceptible de cerrarse por medio de una presa con el fin de almacenar sus aguas. Una presa es una barrera que se construye al paso de una corriente de agua para almacenarla.

En este apartado se hace referencia a los conceptos hidrológicos fundamentales para el diseño de vasos, que son de gran importancia en hidrología, ya que constituyen las bases para el dimensionamiento de las presas y obras de aprovechamiento y protección contra inundaciones.

C.2. Características de los almacenamientos

Los vasos de almacenamiento de una presa sirven para regular los escurrimientos de un río, es decir, almacenan el volumen de agua en exceso que se acumula de la temporada de lluvia para utilizarlo en la época de estiaje.

En la Fig. C.1, se presenta, en forma esquemática, el hidrograma anual de escurrimiento en un río y una demanda considerada constante durante todo el año, que es mayor que la aportación del río en los meses de julio a noviembre, por lo que sería necesario, almacenar el volumen sobrante para satisfacer la demanda en los meses en que los escurrimientos del río no alcanzan, éste volumen se necesitaría guardar en un vaso de almacenamiento.

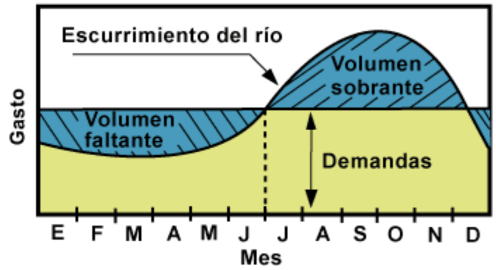

Figura C.1. Hidrograma anual de escurrimiento.

Los vasos de almacenamiento pueden tener los siguientes propósitos: irrigación, generación de energía eléctrica, control de avenidas, abastecimiento de agua potable, navegación, acuacultura, recreación, retención de sedimentos.

Los principales componentes, capacidades y niveles de un vaso de almacenamiento se muestran en la Fig. C.2.

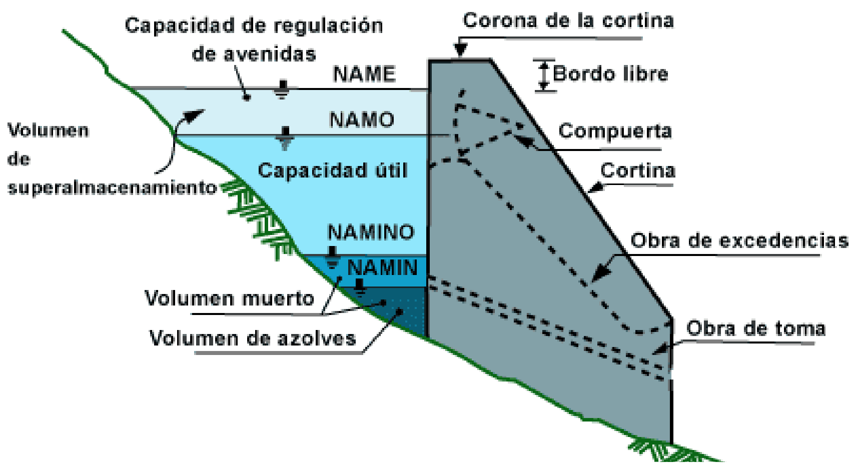

Figura C.2. Principales componentes de un vaso de almacenamiento.

El **NAMINO** (nivel de aguas mínimas de operación) es el nivel más bajo con el que puede operar la presa. En el caso de presas para generación de energía eléctrica, el NAMINO se fija de acuerdo a la carga mínima requerida por las turbinas.

El volumen muerto es el que queda bajo del NAMINO; es un volumen del que no se puede disponer. El volumen de azolves es el que queda abajo del nivel de la toma y se reserva para recibir el acarreo de sólidos por el río durante la vida útil de la presa. Los sedimentos en una presa se reparten a lo largo del embalse, depositándose los más gruesos al principio del mismo y los más finos cerca de la cortina.

El **NAMO** (nivel de aguas máximas ordinarias o de operación) es el nivel máximo del agua en que se debe operar la presa para satisfacer las demandas.

El vertedor u obra de excedencias es la estructura que sirve para desalojar los volúmenes excedentes de agua que pueden poner el peligro la seguridad de la obra. Si el vertedor no es controlado por compuertas, el NAMO coincide con su cresta o punto más alto. En el caso de que la descarga por el vertedor esté controlada (generalmente por compuertas), el

NAMO puede estar por arriba de la cresta e incluso cambiar a lo largo del año. De tal forma, que en el estiaje es posible fijar un NAMO mayor que en época de avenidas, ya que la probabilidad de que se presente una avenida de consideración es menor en el estiaje. En la operación de una presa los niveles del agua fluctúan entre el NAMINO y el NAMO. El volumen que se almacena entre el NAMINO y el NAMO se llama volumen o capacidad útil y es con el que se satisfacen las demandas de agua.

El **NAME** (nivel de aguas máximas extraordinarias) es el nivel más alto que debe alcanzar el agua en el vaso bajo cualquier condición. El volumen que queda entre este nivel y el NAMO, llamado superalmacenamiento, se emplea para controlar las avenidas que se presentan cuando el nivel en el vaso está cercano al NAMO. El bordo libre es el espacio entre el NAME y la máxima elevación de la cortina (corona) y está destinado a contener el oleaje y la marea producidos por el viento, así como a compensar las reducciones en la altura de la cortina provocadas por sus asentamientos (Aparicio, M.F.J., 2005).

Obra de toma. Estructura que permite la extracción de agua en forma controlada del embalse para demandar algún fin deseado.
Obra de excedencias. Estructura que permite que los excedentes de agua pasen de nuevo a la corriente, sin peligro para la presa.
Obra de desvío. Son obras de carácter temporal, que tienen por objeto controlar adecuadamente la corriente durante la construcción de la presa.

C.3. Clasificación de las presas

La clasificación de las presas es de acuerdo a sus materiales de construcción y a su concepción estructural, a continuación se presenta un cuadro sinóptico con la clasificación de las mismas. En la Fig. C.3 se esquematiza los elementos de un aprovechamiento hidráulico superficial (Vega et al., 1992).

Presas
- **Materiales sueltos (tierra y roca)**
  - Tierra
    - Relleno hidráulico
    - Sección homogénea compactada
  - Materiales graduados
  - Enrocamiento
    - Núcleo de tierra (impermeable)
    - Pantalla de concreto
- **Materiales cementados (concreto y mampostería)**
  - Gravedad
    - Masiva
    - Aligerada
  - Contrafuertes
    - Machones
    - Losas planas
    - Arcos o bóvedas múltiples
  - Arco y bóveda

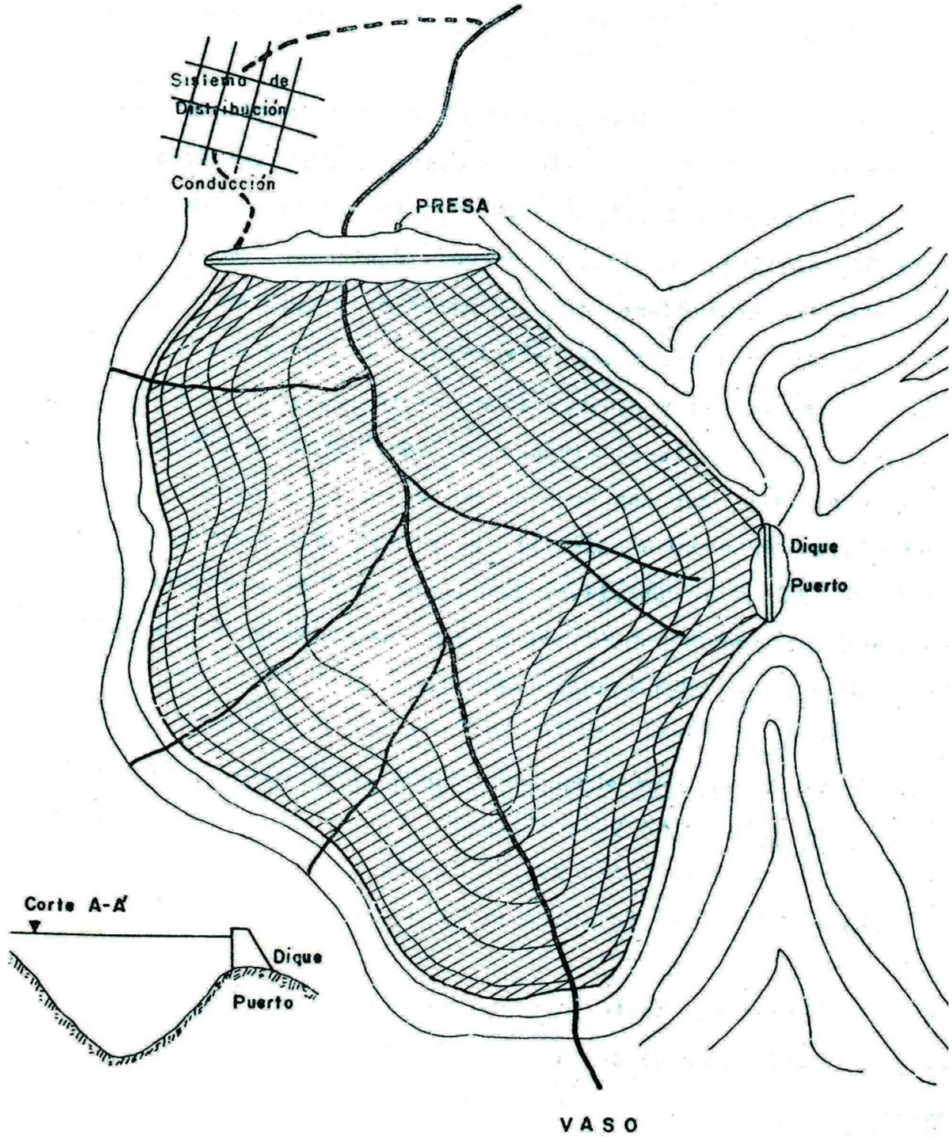

Figura C.3. Elementos de un aprovechamiento hidráulico superficial.

Referencias

C.1. Aparicio, M.F.J. ***"Fundamentos de Hidrología de superficie"***. Limusa, México, 2005.

C.2. Vega, R.O., Arreguín, C.F.I. ***"Presas de almacenamiento y derivación"***. División de Estudios de Posgrado, Facultad de Ingeniería, UNAM, México, 1992.