

13  
2ej



# UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERIA

BENEFICIO MAXIMO EN GENERACION  
PARA SISTEMAS HIDROELECTRICOS

## T E S I S

QUE PARA OBTENER EL TITULO DE :  
**INGENIERO EN COMPUTACION**

P R E S E N T A N :

MARIA DOLORES BETANZOS BETANZOS  
OSCAR CAMACHO FLORES  
SANDRA AMARO PEREZ

DIRECTOR DE TESIS :  
DRA. CRISTINA VERDE RODARTE



MEXICO, D. F.

1993.

**TESIS CON  
FALLA DE ORIGEN**



Universidad Nacional  
Autónoma de México



## **UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso**

### **DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

## INDICE

<b>1</b>	<b>INTRODUCCIÓN</b>	<b>1</b>
<b>2</b>	<b>MODELO HIDRÁULICO DEL SISTEMA GRIJALVA</b>	<b>5</b>
2.1	INTRODUCCIÓN.....	5
2.2	PLANTEAMIENTO.....	6
2.3	CONDICIONES DE OPTIMALIDAD.....	13
<b>3</b>	<b>POLÍTICAS DE OPERACIÓN EN LAZO ABIERTO Y LAZO CERRADO</b>	<b>17</b>
3.1	INTRODUCCIÓN.....	17
3.2	SOLUCIÓN A LAS CONDICIONES DE OPTIMALIDAD.....	17
	ALGORITMO.....	18
3.3	POLÍTICA EN LAZO ABIERTO.....	22
3.4	POLÍTICA EN LAZO CERRADO.....	23
3.5	EVALUACIÓN DE LAS POLÍTICAS DE OPERACION.....	25
3.6	COMPARACIÓN ENTRE LAS POLÍTICAS DE OPERACION.....	29
<b>4</b>	<b>PRESENTACIÓN DE RESULTADOS</b>	<b>31</b>
4.1	INTRODUCCIÓN.....	31
4.2	GRAFICACIÓN CON ANIMACION.....	31
4.2.1	PROGRAMA GENERAL.....	33
4.3	SUBROUTINAS UTILIZADAS POR EL PROGRAMA GENERAL.....	36
	INTRODUCCIÓN.....	36
4.3.1	SUBROUTINA QUE INICIALIZA GRÁFICOS.....	36
4.3.2	SUBROUTINA QUE LEE LOS VOLUMENES ALMACENADOS ESTIMADOS EN EL MES SIGUIENTE Y LOS VOLUMENES TURBINADOS ACTUALES.....	37
4.3.3	SUBROUTINA QUE LEE EL AGUA TURBINADA POR LAS PRESAS EN EL MES ACTUAL.....	37
4.3.4	SUBROUTINA QUE DIBUJA UNA PRESA.....	37
4.3.5	SUBROUTINA QUE DIBUJA LA SALIDA DE AGUA DE LAS PRESAS.....	38
4.3.6	SUBROUTINA QUE DIBUJA EL RELLENADO DE AGUA EN LAS PRESAS..	38
4.3.7	SUBROUTINA QUE DIBUJA CUANDO SUBE EL NIVEL DE AGUA EN LAS PRESAS.....	38

4.3.8	SUBROUTINA QUE LEE LOS MESES A PARTIR DEL MES ACTUAL EN EL QUE SE ESTA.....	39
4.3.9	SUBROUTINA QUE ESCRIBE LOS LITROS DIVIDIDOS ENTRE DOS.....	39
4.3.10	SUBROUTINA QUE LEE LOS LITROS DIVIDIDOS ENTRE DOS.....	39
4.3.11	SUBROUTINA QUE LEE Y ESCRIBE LOS BENEFICIOS.....	39
4.3.12	SUBROUTINA QUE LEE Y ESCRIBE LOS BENEFICIOS MEDIOS.....	40
4.3.13	SUBROUTINA QUE GRAFICA LOS BENEFICIOS DESDE EL MES ACTUAL..	40
4.3.14	SUBROUTINA QUE ROTA LOS MESES A PARTIR DEL MES ACTUAL.....	40
4.3.15	SUBROUTINA QUE ESCRIBE LOS MESES A PARTIR DEL MES ACTUAL..	40
4.3.16	SUBROUTINA QUE GRAFICA LOS CODIGOS Y VALORES DE LOS BENEFICIOS.....	41
<b>5 CONCLUSIONES</b>		<b>62</b>
<b>APENDICE A, DESCRIPCION DE LOS PROGRAMAS EJECUTABLES</b>		<b>64</b>
	INTRODUCCION.....	64
A.1	PROGRAMA PRINCIPAL.....	65
A.2	PROGRAMA QUE REALIZA LA OPTIMACION EN TIEMPO REAL.....	73
A.3	PROGRAMA QUE REALIZA LA OPTIMACION CON INFLUJOS ALMACENADOS.....	80
A.4	PROGRAMA QUE RESUELVE LAS CONDICIONES DE OPTIMALIDAD.....	84
A.5	PROGRAMA QUE CALCULA LAMDA, SIGMA Y EL ERROR DE ÉPSILON..	92
A.6	FUNCIÓN QUE CALCULA EL VOLUMEN DE LAS PRESAS.....	95
A.7	FUNCIÓN QUE CALCULA LAS GANANCIAS TOTALES.....	104
A.8	FUNCIÓN QUE SUMA LAS GANANCIAS OBTENIDAS EN DOCE MESES...	107
A.9	PROGRAMA QUE CALCULA LAS GANANCIAS MEDIAS REALES.....	110
A.10	PROGRAMA QUE ROTA MATRICES.....	113
A.11	PROGRAMA DE INICIALIZACIÓN .....	116
<b>APENDICE B, ARCHIVOS EN FORMATO MATLAB</b>		<b>118</b>
<b>APENDICE C, PROGRAMAS DE ASIGNACIÓN</b>		<b>119</b>
<b>APENDICE D, ARCHIVOS EN CÓDIGO ASCII</b>		<b>131</b>
<b>REFERENCIAS Y BIBLIOGRAFÍA</b>		<b>133</b>

---

## 1. INTRODUCCIÓN

---

Desde el punto de vista de aprovechamiento hidroeléctrico el sistema del río Grijalva, formado básicamente por las presas La Angostura, Chiccoasén y Malpaso, es el más importante en el país. Además de esta característica el sistema permite controlar las inundaciones en la planicie gracias a su capacidad de regulación.

Sin embargo, no es suficiente su alta capacidad de generación para obtener un sistema con alto rendimiento; una política de operación adecuada de las turbinas puede aumentar considerablemente los beneficios de generación. Es por esto que en el Instituto de Ingeniería de la UNAM se realizan desde hace varios años ref [1] y [2] estudios tendientes a obtener el mayor rendimiento del sistema.

En particular, el presente trabajo está basado en una investigación anterior y tiene como objetivo el diseñar una política de operación mensual en tiempo real que tome en cuenta información actualizada de los eventos hidrológicos de la cuenca del Grijalva.

Se proponen dos posibles soluciones al problema planteado, una de ellas está basada únicamente en la información estadística del sistema y se denotará política en lazo abierto; la segunda basada en valores reales y en información estadística del sistema, se denota política retroalimentada para un horizonte de optimación en lazo abierto ó política en lazo cerrado. La diferencia básica entre las soluciones propuestas es que la primera realiza el cálculo fuera de línea sin considerar las posibles perturbaciones o variaciones del sistema, y la política en lazo cerrado se basa en el estado actual del sistema.

La fig. 1.1 muestra de manera gráfica el tipo de evolución de los volúmenes mensuales cuando se emplea la política en lazo abierto; esto

es, se obtienen valores de los volúmenes que deben turbinarse y derramarse cada mes del año, ciclicamente calculados con base en un criterio anual de optimación sin considerar los valores presentes de las presas.

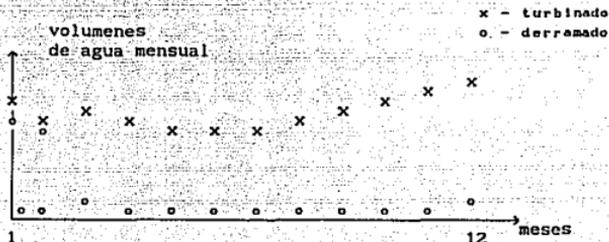


Fig. 1.1

Por el contrario la figura 1.2 muestra el tipo de evolución de los volúmenes mensuales para una política retroalimentada que recibe la información del estado de las presas y calcula con base en ésta, los valores óptimos de volúmenes turbinados y derramados para un horizonte anual.

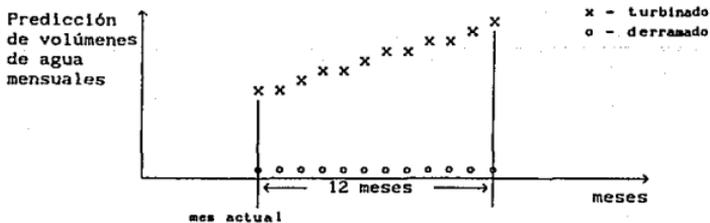


Fig. 1.2

Se hace notar que la política en lazo cerrado, contraria a la de lazo abierto, requiere de la ejecución mensual del programa de optimación. Sin embargo esto no representa ningún problema hoy en día ya que el programa está implantado en una computadora personal. Además la

operación del programa es muy simple y no requiere personal altamente capacitado para su operación. Los resultados se presentan de manera gráfica animada a colores que facilitan su interpretación.

La comparación presentada en este trabajo de ambas políticas con base en los flujos reales durante 30 años, muestra las bondades de la política retroalimentada; se logró un aumento considerable en el beneficio de generación a lo largo de los 30 años de simulación.

La presente tesis se ha organizado de la siguiente forma:

En el capítulo 2 se describe el modelo matemático del sistema Grijalva junto con el beneficio a optimizar, tomado de la ref [1]. Adicionalmente se presentan las condiciones de optimalidad que deben cumplirse para la solución del problema de control óptimo.

Posteriormente, en el capítulo 3 se presenta un mecanismo para encontrar la solución a las condiciones de optimalidad junto con el algoritmo para su implantación. En este mismo capítulo se describen las filosofías de las políticas en lazo abierto y lazo cerrado; tomando como base el algoritmo para la solución a las condiciones de optimalidad. La implantación del algoritmo se realizó en el ambiente del intérprete matemático MATLAB y únicamente la presentación de los resultados se programó en el lenguaje C.

Finalmente se analizan los resultados obtenidos utilizando ambas políticas (lazo abierto y cerrado) y además se comparan con resultados obtenidos mediante una simulación. El papel más importante en la comparación lo juegan el beneficio en generación. Cabe hacer notar que en éste trabajo se tomará el concepto de beneficio y ganancia como sinónimos. Adicionalmente, se comparan las políticas propuestas con la política, obtenida considerando el sistema hidráulicamente desconectado. El resultado de dicha comparación es favorable nuevamente a la política en lazo cerrado.

En el capítulo 4 se describe el programa junto con los procedimientos que realizan la implantación de los resultados amigablemente por medio de gráficas y dibujos animados. Este programa interactúa con MATLAB a través del sistema operativo MS-DOS y está realizado en lenguaje C con el compilador turbo C++ de Borland versión 1.0, además requiere de las siguientes librerías: `stdio.h`, `graphics.h`, `math.h`, `conio.h`, `stdlib.h`, `string.h`.

Las conclusiones del trabajo se presentan en el capítulo 5 haciendo notar el trabajo o estudios futuros que podrían enriquecer la solución propuesta.

En los apéndices se describen brevemente los programas, funciones y archivos utilizados en la realización del sistema, incluyendo diagramas de flujo y listado.

## 2. MODELO HIDRÁULICO DEL SISTEMA GRIJALVA

### 2.1 INTRODUCCIÓN

En este capítulo se describe el modelo hidráulico del sistema Grijalva, formado por las presas La Angostura, Chicoasén y Malpaso, las cuales se encuentran hidráulicamente acopladas en cascada. Es decir, el volumen turbinado y vertido en La Angostura, se considera parte del influjo de la presa Chicoasén; similarmente para Malpaso su influjo depende de el área de captación y los volúmenes derramados por vertedores y turbinas de la presa Chicoasén. La Fig. 2.1 muestra esquemáticamente este acoplamiento, en donde se ha asignado el subíndice  $i$  al elemento  $i$ -ésimo del sistema Grijalva. Así el subíndice 1 corresponde a la primera presa y el subíndice 3, corresponde a la última presa del río, aguas abajo.

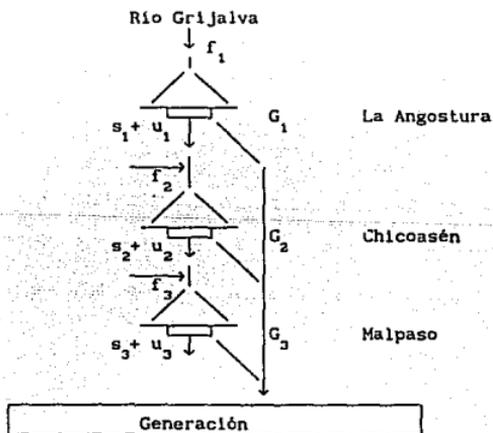


Fig. 2.1 Configuración del sistema

Además en este capítulo se presenta, junto con el criterio de optimización o desempeño deseado, las condiciones de optimalidad que garantizan la solución del problema en términos de control óptimo.

## 2.2 PLANTEAMIENTO

El sistema hidráulico del río Grijalva está formado por 3 presas cuya función es almacenar y permitir regular los volúmenes de agua a lo largo de todo el año con fines de generación de energía eléctrica. Los volúmenes de estas presas satisfacen la ecuación de continuidad discretizada.

Para La Angostura:

$$x_1^{k+1} = x_1^k + f_1^k - u_1^k - s_1^k \quad (2.1)$$

Para Chiccoasén:

$$x_2^{k+1} = x_2^k + f_2^k + u_1^k - u_2^k + s_1^k - s_2^k \quad (2.2)$$

Para Malpaso:

$$x_3^{k+1} = x_3^k + f_3^k + u_2^k - u_3^k + s_2^k - s_3^k \quad (2.3)$$

donde el superíndice k representa el k-ésimo intervalo de tiempo suponiendo que las variables son constantes durante dicho intervalo. Para optimización de las políticas de operación mensual se considera que el intervalo corresponde a 30 días. En las ecuaciones anteriores

$x_1^k$  representa el volumen almacenado durante el periodo k en la presa 1

$u_1^k$  representa el volumen turbinado durante el periodo k en la presa 1

$f_i^k$  representa el volumen de agua introducido en la presa  $i$ , durante el periodo  $k$ , debido a las precipitaciones pluviales captadas en el área.

$s_i^k$  representa el volumen derramado por vertederos durante el periodo  $k$  en la presa  $i$ .

Esto quiere decir que el volumen de la presa  $i$  en el mes  $k+1$  será igual a su volumen en el mes  $k$  más el influjo debido a las precipitaciones pluviales durante el mes  $k$  dentro de su área de captación  $f_i^k$ , más el volumen turbinado y vertido en la presa anterior  $i-1$  durante el mes  $k$ ,  $u_{i-1}^k$  y  $s_{i-1}^k$ , menos el volumen turbinado y vertido en esa presa,  $u_i^k$  y  $s_i^k$ , durante el mes  $k$ .

Debido a condiciones físicas de las presas, tanto el volumen almacenado,  $x_i^k$ , como el volumen turbinado,  $u_i^k$ , están sujetos a restricciones, ya que no es posible almacenar un volumen mayor a la capacidad máxima de cada presa, así como tampoco es posible tener volúmenes negativos. Además el gasto de las turbinas está limitado por un valor máximo que depende de su capacidad para turbinar y por un valor mínimo abajo del cual las turbinas pierden su eficiencia. Por tanto, se considera que los volúmenes están acotados de acuerdo a

$$\underline{x}_i \leq x_i^k < \bar{x}_i \quad (2.4)$$

$$\underline{u}_i \leq u_i^k < \bar{u}_i \quad (2.5)$$

donde  $\bar{x}_i$  es el volumen máximo que puede almacenar la presa  $i$ , y se asigna en función de la capacidad máxima de almacenamiento de cada presa;  $\underline{x}_i$  es el volumen mínimo de la presa  $i$ , asociado a la altura de las tomas de gasto para las turbinas;  $\bar{u}_i$  es el volumen máximo posible que se puede turbinar mensualmente y se define en función del número de turbinas de cada presa, junto con su capacidad máxima;  $\underline{u}_i$  corresponde al volumen mínimo mensual calculado multiplicando

los gastos mínimos permitidos de las turbinas en m/s por 43200.

Como se menciona en la introducción el problema a resolver consiste en determinar la política de operación para generación de energía eléctrica del sistema hidráulico del río Grijalva, de tal manera que se maximice la generación anual, tomando en cuenta que el agua almacenada en la presa al final del año pueda ser usada en años futuros.

Esta tarea no es fácil y existen diferentes alternativas para tratar de resolver este problema. A continuación se adopta la política propuesta por C. Verde [1] en donde se plantea el problema en términos de control óptimo con un criterio que toma en cuenta tanto el volumen almacenado en la presa al final del horizonte, como el costo por generación de electricidad mensual. Siguiendo el criterio reportado en [1] se tiene

$$J_{\text{total}} = E \left\{ h(x^N) + \sum_{k=0}^{N-1} g(x^k, u^k) \right\} \quad (2.6)$$

donde E denota valor esperado. El primer término corresponde al beneficio esperado al final del horizonte y el segundo está asociado al beneficio esperado por generación del sistema. En el criterio (2.6)  $x^k$  y  $u^k$  se definen como

$$(x^k)^T = [x_1^k \quad x_2^k \quad x_3^k]^T \quad (2.7)$$

$$(u^k)^T = [u_1^k \quad u_2^k \quad u_3^k]^T \quad (2.8)$$

Así, el problema consiste en determinar el vector de descarga  $u^k$  para todos los periodos  $k=1, \dots, N$  de tal manera que el criterio (2.6) sea maximizado tomando en cuenta las restricciones físicas de vertederos, capacidad de almacenamiento, características de las turbinas y la ecuación dinámica de continuidad del sistema hidráulico.

Se hace notar que este problema puede o no tener solución, dependiendo de las funciones  $g(x^k, u^k)$  y  $h(x^N)$  seleccionadas y de las restricciones impuestas. Siguiendo la referencia considerada, se tiene la siguiente sumatoria para el horizonte de optimación de un año.

$$g(x^k, u^k) = \sum_{i=1}^3 G_i(u_i^k, \hat{x}_i^k) \quad (2.9)$$

con

$$G_i(u_i^k, \hat{x}_i^k) = c_i^k u_i^k (\alpha_i + \beta_i \hat{x}_i^k + \gamma_i (\hat{x}_i^k)^2) \quad (2.10)$$

donde  $\hat{x}$  representa el valor medio del volumen almacenado en las presas definido como

$$\hat{x}_i^k = \frac{1}{2} (x_i^{k+1} + x_i^k) \quad (2.11)$$

$c^k$  es una constante que pondera el costo de generación en MWH

$\alpha_i$ ,  $\beta_i$  y  $\gamma_i$  son constantes que dependen de la geometría de la presa y fueron tomados de la referencia [1].

Para el término  $h(x^N)$  que pondera el beneficio de dejar agua almacenada al final del año, se consideró la función

$$h(x^{12}) = \sum_{i=1}^3 x_i^{12} \sum_{j=1}^3 [\alpha_j + \beta_j x_j^{12} + \gamma_j (x_j^{12})^2] \quad (2.12)$$

Esto quiere decir que el beneficio de dejar agua almacenada al final del ciclo se pondera más en el primer elemento del río que en el último.

Con objeto de manejar el problema de optimación arriba planteado de manera estructurada, éste se transforma en una forma compacta haciendo uso de vectores y matrices. Como consecuencia el beneficio futuro (2.12)

se reduce a

$$h(x^N) = [x^N]^T [A_N + B_N x^N + C_N y^N] \quad (2.13)$$

en donde se han definido al vector constante  $A_N$  de dimensión 3

$$A_N^T = [\alpha_1 + \alpha_2 + \alpha_3 \quad \alpha_2 + \alpha_3 \quad \alpha_3] \quad (2.14)$$

y las matrices  $B_N$  y  $C_N$  como

$$B_N = \begin{bmatrix} \beta_1 & \frac{1}{2} \beta_2 & \frac{1}{2} \beta_3 \\ \frac{1}{2} \beta_2 & \beta_2 & \frac{1}{2} \beta_2 \\ \frac{1}{2} \beta_3 & \frac{1}{2} \beta_2 & \beta_3 \end{bmatrix} \quad (2.15)$$

$$C_N = \begin{bmatrix} \gamma_1 & \gamma_2 & \gamma_3 \\ 0 & \gamma_2 & \gamma_3 \\ 0 & 0 & \gamma_3 \end{bmatrix} \quad (2.16)$$

Además  $y^N$  es un vector definido como

$$y^k = [x^k \circ x^k] \quad \circ y_1^k = (x_1^k)^2 \quad (2.17)$$

donde  $\circ$  denota la multiplicación de elemento por elemento de dos matrices de las mismas dimensiones.

Igualmente que para el término  $h(x^N)$ , la función  $g(x^k, u^k)$  ec. (2.10) se puede escribir en forma vectorial compacta como

$$g(x^k, u^k) = (u^k)^T C_w^k \left( A_v + \frac{B}{2} [x^{k+1} + x^k] + \frac{C}{2} [(x^{k+1} + x^k) \circ (x^{k+1} + x^k)] \right) \quad (2.18)$$

con ayuda de las siguientes matrices constantes

$$A_v^T = [\alpha_1 \quad \alpha_2 \quad \alpha_3] \quad (2.19)$$

$$B_v^T = [\beta_1 \quad \beta_2 \quad \beta_3] \quad (2.20)$$

$$B = \text{diag} [\beta_1 \quad \beta_2 \quad \beta_3] \quad (2.21)$$

$$C = \text{diag} [\gamma_1 \quad \gamma_2 \quad \gamma_3] \quad (2.22)$$

$$C_w^k = \text{diag} [c^k I_{n,n}] \quad (2.23)$$

Además usando la notación vectorial, las ecuaciones de continuidad (1), (2) y (3) se reducen a

$$x^{k+1} = x^k + f^k + M(u^k + s^k) \quad (2.24)$$

donde

$$(f^k)^T = [f_1^k \quad f_2^k \quad f_3^k]^T \quad (2.25)$$

representa el influjo o perturbación aleatoria del sistema, es decir, es la cantidad de agua captada por las presas debido a precipitaciones pluviales de la cual sólo se conoce su valor medio mensual. Además la matriz cuadrada M, representa el acoplamiento entre las presas, y se define como

$$M = \begin{bmatrix} -1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 1 & -1 \end{bmatrix} \quad (2.26)$$

Igualmente, reescribiendo las restricciones en forma vectorial se tiene

$$\underline{x} \leq x^k < \bar{x} \quad (2.27)$$

$$\underline{u} \leq u^k < \bar{u} \quad (2.28)$$

Finalmente, se considera que el volumen derramado por vertedores también está sujeto a restricciones físicas por lo que se tiene

$$0 < s^k < \bar{s} \quad (2.29)$$

Resumiendo se tiene un criterio a maximizar con restricciones de igualdad, ec. (2.24) y de desigualdad, ecs. (2.27), (2.28) y (2.29). Siguiendo la metodología presentada en [1], el criterio (2.11) se puede reducir a

$$g(x^k, u^k) = (u^k)^T C_w^k [b^k + (d^k \cdot x^k)] + \frac{(d^k \cdot M u^k) + C y^k + C \frac{N}{4} z^k}{2} \\ + (r^k)^T C_w^k C M u^k - (z^k)^T C_w^k C \frac{L}{2} u^k \quad (2.30)$$

en donde  $z$  y  $r$  corresponden a las pseudo-variables y están definidas como

$$z^k = [u^k \cdot u^k] \Rightarrow z_1^k = (u_1^k)^2 \quad \text{para los entradas} \quad (2.31)$$

$$\text{y para los productos cruzados como } r^k = [u^k \cdot x^k] \Rightarrow r_1^k = x_1^k u_1^k \quad (2.32)$$

y los demás vectores se definen como

$$q^k = f^k + M s^k \quad (2.33)$$

$$b^k = A_v + \frac{B}{2} q^k + \frac{C}{4} [q^k \circ q^k] \quad (2.34)$$

$$d^k = B_v + C q^k \quad (2.35)$$

$$\bar{N} = M + 2I \quad (2.36)$$

$$L = M + I \quad (2.37)$$

con I la matriz identidad de dimensión 3X3.

Como consecuencia, el índice de funcionamiento o beneficio global que se desea maximizar se reduce al valor esperado de la suma de las ecuaciones (2.12) y (2.30) obteniendo

$$J_{Total} = E \left\{ [x^{12}]^T [A_N + B_N x^{12} + C_N y^N] + \sum_{k=0}^{11} \left\{ (u^k)^T C_v^k [b^k + d^k \circ x^k] + [d^k \circ M u^k] + C y^k + C \frac{N}{4} z^k \right\} + (r^k)^T C_v^k C M u^k - (z^k)^T C_v^k C \frac{1}{2} u^k \right\} \quad (2.38)$$

sujeto a las restricciones de igualdad, (2.17), (2.26), (2.30) y (2.31) y de desigualdad, (2.27), (2.28) y (2.29).

## 2.3 CONDICIONES DE OPTIMALIDAD

Debido a que el problema de optimación que se desea resolver tiene tanto restricciones de igualdad como de desigualdad, para establecer matemáticamente el planteamiento en el marco de referencia del cálculo de variaciones se debe formular un nuevo criterio o función a optimizar considerando [3]. Por tanto involucrando los operadores de Lagrange y de

Kuhn-Tucker de acuerdo al procedimiento propuesto en [1] se tiene el criterio aumentado

$$J_a = J_{total} + E \left\{ (\mu^{k+1})^T (-y^{k+1} + [x^{k+1} \circ x^{k+1}]) + (\phi^k)^T (-z^k + [u^k \circ u^k]) + (\psi^k)^T (r^k + [u^k \circ x^k]) + (\lambda^{k+1})^T (-x^{k+1} + x^k + q^k + M u^k) + \bar{c}^{k+1} (\bar{x} - x^{k+1}) + \bar{g}^{k+1} (x^{k+1} - \underline{x}) + \bar{g}^k (\bar{u} - u^k) + \bar{g}^k (u^k - \underline{u}) \right\} \quad (2.39)$$

Finalmente después de reordenar términos en el criterio (2.39), y eliminar términos que no afecten la optimación y de realizar manipulaciones algebraicas se obtiene la funcional

$$J_a = E \left\{ (x^{12})^T (B_N x^{12} + [\mu^{12} \circ x^{12}]) + C_N y^N + A_N - \lambda^{12} \right) - (\mu^{12})^T y^N + \sum_{k=1}^{11} \left\{ (u^k)^T C_w^k [b^k + [d^k \circ x^k] + \frac{1}{2} [d^k \circ M u^k]] + C y^k + C \frac{N}{4} z^k + [\phi^k \circ u^k] + [(r^k)^T C_w^k C M - (z^k)^T C_w^k C \frac{1}{2}] u^k - (\mu^k)^T y^k + (x^k)^T [\mu^k \circ x^k] - (\phi^k)^T z^k + (u^k)^T [\psi^k \circ x^k] - (\psi^k)^T r^k - (\lambda^{k+1} - \lambda^k + \xi^{k+1})^T x^k + (\lambda^{k+1})^T (q^k + M u^k) + (\xi^{k+1})^T (q^k + M u^k) + (\sigma^k) u^k + (\xi^{k+1})^T x_o^k + (\sigma^k)^T u_o^k \right\} \quad (2.40)$$

Por lo tanto, se busca la secuencia  $u^k$  que maximice el criterio  $J_a$ . La solución a este problema se puede establecer en términos del cálculo de variaciones y el conjunto de condiciones que deben satisfacer la secuencia  $u^k$  para considerarse candidata a ser solución del problema de optimación se reducen a

$$E \left\{ 2B_N x^N + 2(\mu^N \circ x^N) + C_N [x^N \circ x^N] + A_N - \lambda^N \right\} = 0 \quad (2.41)$$

$$E \left\{ C_N x^N - \mu^N \right\} = 0 \quad (2.42)$$

$$E \left\{ \lambda^{k+1} - \lambda^k + \xi^{k+1} + 2[\mu^k \circ x^k] + \frac{1}{2} C_W^k [(d^k + \psi^k) \circ u^k] \right\} = 0 \quad (2.43)$$

$$E \left\{ -\mu^k + C_W^k C u^k \right\} = 0 \quad (2.44)$$

$$E \left\{ -\phi^k + \frac{1}{2} C_W^k \left( \frac{N}{2} C - L \right) u^k \right\} = 0 \quad (2.45)$$

$$E \left\{ -\psi^k + C_W^k C M u^k \right\} = 0 \quad (2.46)$$

$$E \left\{ C_W^k b^k + M^T (\lambda^{k+1} + \xi^{k+1}) + C_W^k [d \circ x^k] + [\psi^k \circ x^k] + C_W^k C [x^k \circ x^k] + \right. \\ \left. + \sigma^k + \frac{1}{2} (C \frac{N}{2} - L^T) C_W^k [u^k \circ u^k] + M^T C_W^k C [x^k \circ u^k] + \right. \\ \left. + C_W^k [d^k \circ M u^k] + 2C_W^k [\phi^k \circ u^k] \right\} = 0 \quad (2.47)$$

Con la siguiente ecuación de continuidad

$$x_i^{k+1} = x_i^k + f_i^k + u_{i-1}^k - u_i^k + s_{i-1}^k - s_i^k \quad (2.48)$$

Además en caso de sobrepasar los límites de extracción del volumen turbinado, u, y el volumen almacenado en las presas, variable x, se deben de satisfacer las ecuaciones de exclusión. Esto quiere decir que si el volumen en la presa i excede sus límites en el mes k, la variable  $\xi_i^k$  en el mes k es diferente de cero y en caso contrario se

satisface:

$$\xi_1^k(x_1^k - x_{o1}) = 0 \quad (2.49)$$

De la misma forma para la extracción  $u_1^k$ , si ésta sobrepasa sus límites la variable  $\sigma_1^k$  es diferente de cero y en caso contrario se satisface :

$$\sigma_1^k(u_1^k - u_{a1}) = 0 \quad (2.50)$$

En el siguiente capítulo se propone un algoritmo para resolver el conjunto de ecuaciones dinámicas (2.41) a la (2.47) iterativamente junto con la ecuación de continuidad (2.48) y las condiciones de exclusión. Se hace notar que debido a que las ecuaciones (2.41) y (2.42) son condiciones finales, es necesario que el conjunto de ecuaciones se satisfagan en ambos sentidos, esto es, cuando se incrementa o se decrementa el superíndice k.

---

### 3. POLÍTICA DE OPERACIÓN EN LAZO ABIERTO Y LAZO CERRADO

---

#### 3.1 INTRODUCCIÓN

En el presente capítulo se proponen dos mecanismos para implantar la política de operación del sistema Grijalva, basada en las condiciones de optimalidad presentadas en el capítulo 2. En particular las dos estrategias, corresponden a sistemas de control en lazo abierto y lazo cerrado. Además se presenta un algoritmo general para determinar en ambos casos la secuencia  $u^k$  que satisface las condiciones de optimalidad.

Básicamente, la diferencia que existe al aplicar las dos políticas de operación del sistema, consiste en que la política en lazo abierto no toma información del estado de las presas una vez calculada la estrategia  $u^k$  y por el contrario, la política en lazo cerrado retroalimenta la información del estado del sistema calculando la estrategia  $u^k$  mensualmente.

#### 3.2 SOLUCIÓN A LAS CONDICIONES DE OPTIMALIDAD

El conjunto de ecuaciones no lineales de optimalidad, detalladas en el capítulo 2, son dinámicas con condiciones de frontera al final y al inicio de la solución y hasta el momento no se conoce una solución analítica de ellas, por lo que a continuación se presenta un método iterativo para determinar la solución con base en una búsqueda secuencial. El método sintetiza a manera de algoritmo, para facilitar su comprensión.

## ALGORITMO

1.- Calcular las matrices constantes  $A_N$ ,  $B_N$  y  $C_N$  con base en las ecuaciones (2.14), (2.15) y (2.16). También calcular las constantes  $C$ ,  $N$  y  $L$  con base en las ecuaciones (2.22), (2.36) y (2.37) respectivamente.

2.- Proponer la condición final de frontera  $x^{12}$ , es decir, el volumen deseado en las presas después de doce meses.

3.- Resolver la ecuación (2.42) para  $N=12$ , es decir

$$\mu^{12} = C_N x^{12} \quad (3.1)$$

4.- Resolver la ecuación (2.41) para  $N=12$

$$\lambda^{12} = 2B_N x^{12} + 2[\mu^{12} x^{12}] + C_N [x^{12} x^{12}] + A_N \quad (3.2)$$

5.- Proponer valores iniciales de la descarga  $u^k$  durante todo el horizonte de optimación (12 meses). Se hace notar que el algoritmo resultó ser sensible al valor inicial  $u^k$ , por lo que se sugiere probar con diferentes valores sobre todo cuando se está en la etapa de ajuste de parámetros.

6.- Calcular la ecuación de continuidad (2.48), desde  $n=1$  hasta 12. Tomando en cuenta las restricciones físicas de las presas.

7.- Calcular los vectores  $d^k$ ,  $q^k$  y  $h^k$ , desde  $k=1$  hasta 12. Con base en las ecuaciones (2.35), (2.33) y (2.34) respectivamente.

8.- Resolver la ecuación (2.44) para toda  $k$ . Iniciando para  $k=1$  de adelante hacia atrás.

$$\mu^k = C_W^k C u^k \quad (3.3)$$

9.- Resolver la ecuación (2.45), para toda k desde 1 hasta 12.

$$\phi^k = \frac{1}{2} C_W^k \left( \frac{N}{2} C - L \right) u^k \quad (3.4)$$

10.- Resolver la ecuación (2.46), para toda k desde 1 hasta 12:

$$\psi^k = C_W^k C M u^k \quad (3.5)$$

11.- Resolver la ecuación (2.43), recursivamente de atrás hacia adelante inicializando con el valor k=12 calculado en el paso 4

$$\lambda^k = \lambda^{k+1} + \xi^{k+1} + 2[\mu^k \circ x^k] + \frac{1}{2} C_W^k [(d^k + \psi^k) \circ u^k] \quad (3.6)$$

12.- Determinar  $\sigma^k$ , para toda k, de adelante hacia atrás usando la siguiente ecuación.

$$\begin{aligned} \sigma^k = & C_W^k b^k - M^T ( \lambda^k - 2[\mu^k \circ x^k] - \frac{1}{2} C_W^k [(d^k + \psi^k) \circ u^k] ) + \\ & + C_W^k [d \circ x^k] + [\psi^k \circ x^k] + C_W^k C [x^k \circ x^k] + \frac{1}{2} (C \frac{N}{2} - L^T) C_W^k [u^k \circ u^k] + \\ & + M^T C_W^k C [x^k \circ u^k] + C_W^k [d^k \circ M u^k] + 2C_W^k [\phi^k \circ u^k] \end{aligned} \quad (3.7)$$

Esta ecuación se obtiene sustituyendo la suma (  $\lambda^{k+1} + \xi^{k+1}$  ), a partir de la ecuación (2.43) en (2.47). Se hace notar que  $\sigma^k$  es la variable que castiga al sistema cuando los volúmenes de las presas exceden sus límites, por lo tanto se le asigna el valor de cero si el volumen está en el rango permitido de operación.

13. Calcular el error recursivamente, despejando de la ecuación (2.47) el producto  $-M^T \xi^{k+1}$ .

$$\begin{aligned} \text{Error}^k \triangleq -M^T \xi^{k+1} = & C_w^k b^k - M^T (\lambda^{k+1}) + \sigma^k + \\ & + C_w^k [d \cdot x^k] + [\psi^k \cdot x^k] + C_w^k C [x^k \cdot x^k] + \frac{1}{2} (C \frac{\hat{n}^T}{2} - L^T) C_w^k [u^k \cdot u^k] + \\ & + M^T C_w^k C [x^k \cdot u^k] + C_w^k [d^k \cdot M u^k] + 2C_w^k [\phi^k \cdot u^k] \end{aligned} \quad (3.8)$$

14. Calcular los valores de  $\xi^k$  con base en el  $\text{Error}^k$

$$\xi^k = - (M^T) \text{Error}^k \quad (3.9)$$

donde  $\xi^k$  es el vector que castiga al sistema cuando el volumen turbinado excede sus límites. En caso de estar dentro de los límites, la componente  $u_i^k$  se le asigna a la componente  $\xi_i^k$  del vector  $\xi^k$  el valor de cero.

- 15.- Modificar los valores de los volúmenes turbinados ( $u^k$ ) mediante la expresión:

$$u^k = u^k + \text{alfa} (\text{Error}^k) \quad (3.10)$$

donde alfa es un parámetro constante que se ajusta dependiendo de la velocidad de convergencia para la secuencia óptima  $u^k$ .

La comprensión del efecto de la variable alfa en la optimación, se ejemplifica con ayuda de la figura 3.1. En ésta se presenta una funcional dependiente de los volúmenes a turbinar que para el caso tratado corresponde el beneficio. Supóngase que se tiene como resultado en una de las iteraciones el volumen  $u(A)$ , representado en la figura 3.1 como el punto A; para la siguiente iteración se debe modificar dicho volumen en función del valor de alfa y del Error. Así para poder acercarse al valor óptimo  $u^*$ , es obvio que si se selecciona un valor de alfa pequeño, equivale a dar un paso pequeño

y se estará aún lejos del valor óptimo, punto B. Por el contrario si se selecciona un valor de alfa grande, el valor de  $u$  se modificaría demasiado, corriendo el riesgo de pasarse del óptimo  $u^*$  y tener un valor menor de beneficio que en una iteración anterior, punto (C). Por lo que se recomienda en general hacer varias pruebas para determinar una velocidad de convergencia adecuada.

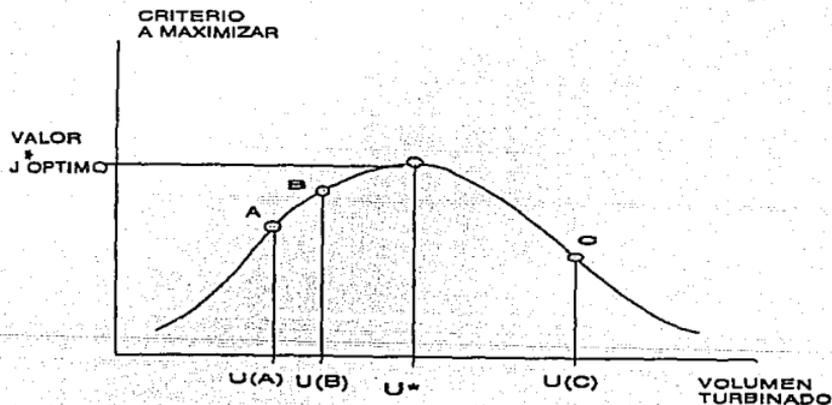


Fig. 3.1 Criterio a optimizar

16.- Calcular los beneficios esperados con la  $u$  obtenida en la iteración. En caso de que el beneficio obtenido sea menor que el de la iteración anterior, se le asigna a  $u$  el valor de la iteración anterior. Por otro lado si los valores de los errores ( $\text{Error}^k$ ) son muy pequeños se considera que se ha encontrado la secuencia  $u^k$  óptima que satisface las condiciones de optimalidad y el algoritmo termina. En caso contrario es necesario reajustar la  $u^k$ , lo cual implica que el algoritmo debe regresar al paso 6. Con objeto de que el algoritmo termine después de un número máximo de iteraciones independientemente del  $\text{Error}^k$ , se le ha dotado de un número máximo de iteraciones seleccionado libremente.

### 3.3 POLÍTICA EN LAZO ABIERTO

La figura 3.2 muestra en forma esquemática la política de un control en lazo abierto en general.



Fig. 3.2 Diagrama de la estrategia de control en lazo abierto

La entrada al controlador es la referencia y determina el valor deseado del sistema, con base en esta referencia el controlador genera señales que afectan al sistema tendiente a lograr el comportamiento requerido de la salida de éste. La ventaja principal de esta estrategia es su costo, ya que no requiere de los sensores en la salida del sistema. Sin embargo es muy sensible a perturbaciones e incertidumbres del sistema.

El diagrama de bloques de la figura 3.3 muestra la estrategia en lazo abierto suponiendo que ésta se implementa en una computadora. Por tanto la tarea es la determinación de los valores de los volúmenes turbinados ( $u$ ) y volúmenes vertidos ( $s$ ) basándose en las condiciones de optimalidad del capítulo 2.

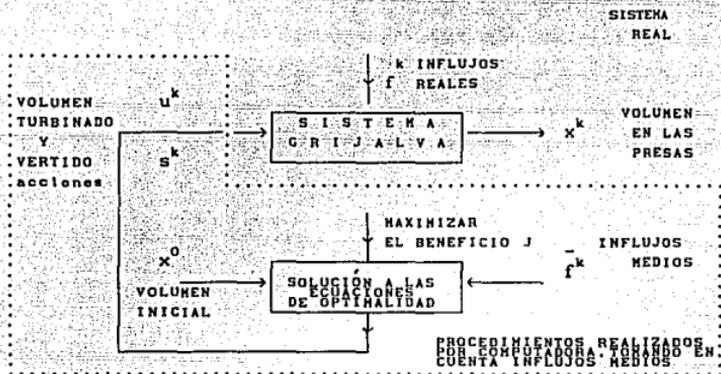


Fig. 3.3 Representación del sistema con la política en lazo abierto

### 3.4 POLÍTICA EN LAZO CERRADO

Como se menciona al inicio, la política en lazo abierto tiene poca robustez, ya que toma valores medios de influjos. Por lo que se supone mejorar el desempeño de la política de operación implantando una ley de control en lazo cerrado, que tome en cuenta toda la información disponible en tiempo real y no únicamente los valores medios de los influjos.

La idea central de la política en lazo cerrado consiste en calcular mensualmente los valores medios  $u^k$  y  $s^k$  obtenidos con base en el volumen real del sistema  $x^0$ . Esto implica el uso de una computadora

operando en tiempo real que mensualmente calcula el volumen turbinado óptimo. Como resultado del problema de optimización, además del volumen a turbinar, la computadora genera predicciones de los volúmenes futuros para el próximo año suponiendo que el influjo real coincide con los valores medios.

Esquemáticamente la figura 3.4 muestra la estrategia propuesta.

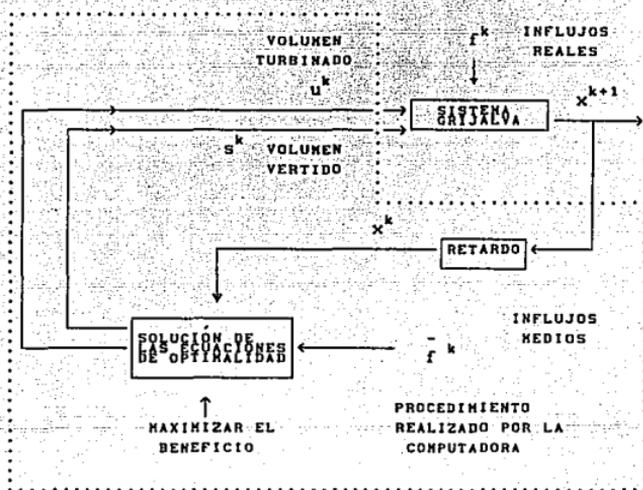


Fig 3.4 Representación del sistema con la política en lazo cerrado

Al igual que en el control en lazo abierto, para el instante  $k$ , se emplean en la solución de las ecuaciones de optimalidad, los valores de influjos medios; sin embargo, en el instante  $k$  los volúmenes de las presas  $x^k$  se suponen conocidos permitiéndole al programa generar los volúmenes turbinados  $u^k$  y los volúmenes derramados por vertederos  $s^k$  con

un reajuste mensual. Este procedimiento se realiza de manera continua mensualmente, aunque únicamente el primer valor de  $u^k$  y  $s^k$  es el que se emplea, ya que para el siguiente mes la nueva información del estado de las presas permite recalcular  $u^k$  y  $s^k$ .

### 3.5 EVALUACIÓN DE LAS POLÍTICAS DE OPERACIÓN

Con objeto de evaluar el beneficio obtenido con las dos políticas propuestas, se simularon, ambos sistemas, tomando los valores de influjos reales registrados desde el año 1959 hasta el año 1988. La tabla de la figura 3.5 reporta la serie de volúmenes turbinados mensuales óptimos  $u^*$  cuando se emplea la política en lazo abierto. Como se mencionó, con ésta estrategia el volumen queda fijo para cada mes independientemente de la cantidad de lluvias del año en cuestión.

	LA ANGOSTURA	CHICDASEN	HALPAGO
ENERO	5.1670e+01	0	1.2000e+03
FEBRERO	2.0997e+03	1.9315e+03	1.0000e+02
MARZO	0	2.2737e-13	2.1480e+03
ABRIL	0	0	1.0000e+02
MAYO	0	1.0000e+02	1.0000e+02
JUNIO	5.1670e+01	0	1.0000e+02
JULIO	2.0480e+03	1.9997e+03	3.7325e+03
AGOSTO	5.1670e+01	5.1670e+01	1.0000e+02
SEPTIEMBRE	0	1.0000e+02	1.0000e+02
OCTUBRE	5.1670e+01	0	1.0000e+02
NOVIEMBRE	3.0326e+03	2.4183e+03	6.4642e+02
DICIEMBRE	5.1670e+01	5.1670e+01	0

Fig. 3.5 Volumen a turbinar medio con la política en lazo abierto

A continuación, en la figura 3.6 se muestran los beneficios obtenidos con una simulación a treinta años utilizando la estrategia en lazo abierto. Para calcular el beneficio medio se empleó la siguiente ecuación

$$J_i = \sum_{j=1-12}^i g^j(x, u_{med}) \quad (3.11)$$

donde  $J$  representa el beneficio real en el sistema para los últimos doce meses,  $x$  son los volúmenes reales y  $u_{med}$  son los valores óptimos medios tomados de la tabla de la figura 3.5. Por otro lado la figura 3.6 muestra los resultados de los beneficios obtenidos por simulación.

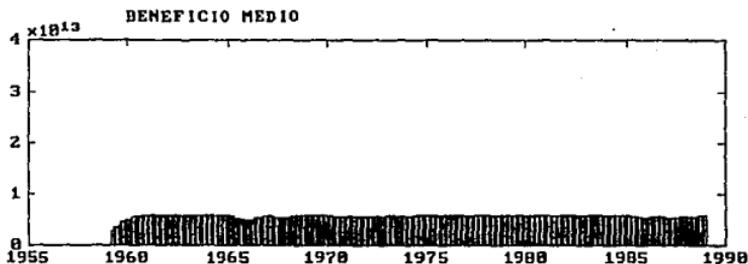


Fig. 3.6 Beneficio real en lazo abierto

Para evaluar la política de operación en lazo cerrado se calcularon dos beneficios, uno tomando en cuenta los valores pasados reales, es decir usando la expresión (3.11) sustituyendo la  $u_{med}$  por la política mensual determinada en lazo abierto, y el segundo considerando el posible beneficio a futuro. Con base en predicciones de los próximos once meses.

El beneficio asociado al futuro, se calcula con base en los volúmenes a turbinar y derramados en las presas en un horizonte a futuro de 12 meses, partiendo de un volumen real almacenado. Por tanto el beneficio se calcula mediante la siguiente expresión

$$J = \sum_{j=1}^{1+12} g^j(x^{\cdot j}, u^{\cdot j}) + h(x^{\cdot 1+12}) \quad (3.12)$$

con  $x^{\cdot}$  y  $u^{\cdot}$  valores óptimos.

Es decir,  $B_{\max}$  se calcula con el beneficio por generación  $g(x,u)$  más el beneficio esperado al final del horizonte  $h(x^N)$ . Como se puede observar, para el cálculo del beneficio por generación se toma en cuenta un volumen real dado inicial  $x(0)$ .

Si las condiciones de predicción del sistema Grijalva se cumplen, y se turbinan el volumen recomendado por la optimización se podría obtener el beneficio a futuro calculado. La figura 3.7 muestra el beneficio a futuro, con base en influjos registrados desde el año 1959 hasta el año 1988.

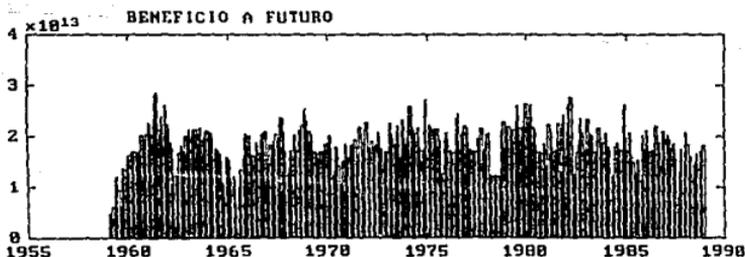


Fig. 3.7 Beneficio a futuro

El Beneficio real a pasado con la política en lazo cerrado, se obtiene al calcular en tiempo real mensualmente los volúmenes  $u^*$  óptimos y evaluar el beneficio adquirido durante los últimos 12 meses de la historia del sistema. Es decir, el beneficio se calcula mediante la siguiente expresión

$$J_1 = \sum_{j=1-12}^1 g^j(x, u^*) \quad (3.13)$$

Ignorándose el costo por dejar agua almacenada. Se hace notar que simulando el sistema de control en lazo cerrado para los treinta años que se tiene registro de las lluvias se puede calcular con 3.13 el beneficio que se habría obtenido si se hubiera aplicado la política propuesta desde el año 1959 hasta 1988. Este beneficio mensual con base en los influjos registrados se presenta en la figura 3.8

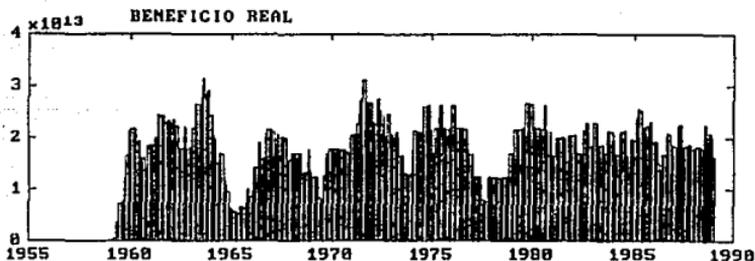


Fig. 3.8 Beneficio real con la política en lazo cerrado propuesto

### 3.6 COMPARACIÓN ENTRE LAS POLÍTICAS DE OPERACIÓN

Con objeto de ampliar la discusión sobre las alternativas para las políticas de operación del sistema, se presenta en la figura 3.9, el beneficio (3.13) calculado cuando se implanta una política de operación en lazo cerrado basada en controles óptimos independientes para cada presa. Esta política se determinó empleando la programación dinámica estocástica y fué tomada de la referencia [2]. Los detalles del método empleado para la determinación de la misma queda fuera del contexto de este trabajo, (ver la referencia [2] para detalles); sin embargo permite argumentar en favor de las políticas en lazo cerrado independientemente del método y criterio propuesto para la solución.

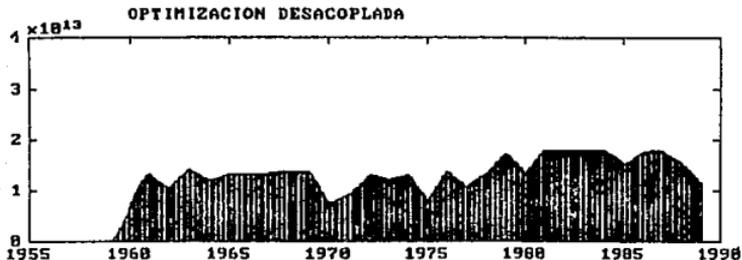


Fig. 3.9 Beneficio real considerando al sistema desacoplado

Analizando primérmente la figura 3.6 donde se representa el beneficio obtenido con la política en lazo abierto durante 30 años, se observa un beneficio prácticamente constante, para todos los años, de  $0.5 \times 10^{13}$ . Es decir es poco sensible a el cambio en las precipitaciones.

Para el caso del beneficio real en lazo cerrado, figura 3.7, se tiene un comportamiento mucho más variable. Como puntos importantes se pueden destacar que se pueden alcanzar beneficios máximos hasta de  $3.2 \times 10^{13}$  y mínimos de  $0.6 \times 10^{13}$ . En este caso, contrario al caso de lazo abierto, las variaciones son muy acentuadas, lo cual implica que la política es sensible a las precipitaciones.

De esto se puede concluir que se puede obtener un mejor beneficio con la política en lazo cerrado que con la de lazo abierto, obviamente la política en lazo cerrado es menos conservadora y por tanto presenta un mayor riesgo de sobrepasar las condiciones físicas de operación en caso de condiciones climatológicas muy adversas.

La comparación del beneficio real obtenida con las dos políticas aquí propuestas y la obtenida con la política asumiendo el sistema hidráulicamente desacoplado, esto es las figuras 3.6, 3.7 y 3.9 muestran que en general la política en lazo abierto es la más conservadora ya que produce un menor beneficio a lo largo de los 30 años simulados. Por el contrario la política en lazo cerrado tomando en cuenta los acoplamientos hidráulicos, genera un mayor beneficio que la política suponiendo desacoplamiento, sin embargo ambas políticas en lazo cerrado permiten generar mayor electricidad lo cual implica que el conocimiento del estado del sistema permite tomar decisiones menos conservadoras.

---

## 4. PRESENTACIÓN DE RESULTADOS

---

### 4.1 INTRODUCCIÓN

Con objeto de facilitar al operador del sistema la lectura de las acciones a ejecutar se dotó al sistema de pantallas gráficas donde se presenta de manera amigable información sobre el estado, política futura y beneficio del sistema. Para ello se utiliza como herramienta de programación el lenguaje Turbo C++ de Borland Versión 1.0. La Fig. 4.1 presenta el gráfico dibujado en la pantalla.

### 4.2 GRAFICACIÓN CON ANIMACIÓN

A continuación se describe de manera general el funcionamiento del programa de graficación, junto con los procedimientos de que consta.

El efecto de animación en la pantalla se logró dejando la parte del dibujo asociada a las presas, vasos, nombres y volúmenes de las presas fija y modificando únicamente el color del ducto que simula el río y la cantidad de agua almacenada.

El dibujo consta de dos recuadros, el primero corresponde a la parte superior de la pantalla y en él se dibuja el diagrama esquemático del sistema Grijalva, junto con sus estados. En el recuadro de la parte inferior se dibuja la gráfica de la evolución del beneficio para los últimos doce meses.

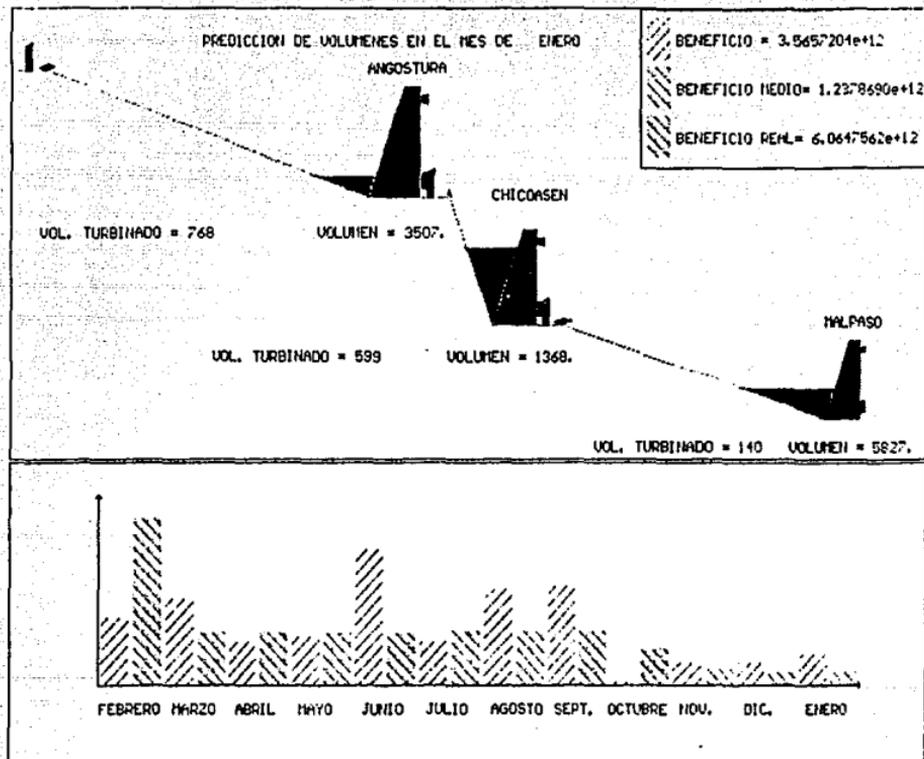


Fig. 4.1 Presentación amigable de resultados

#### 4.2.1 PROGRAMA GENERAL

Para facilitar la ejecución de las tareas, el programa se dividió en módulos específicos.

La Fig. 4.2 muestra el diagrama general del programa de graficación el cual se inicia solicitando las librerías gráficas tipo cadena de caracteres, matemáticas, etc. y declarando todas las variables y subrutinas necesarias para la ejecución del programa. Una vez que se hacen las declaraciones, se procede a pasar al programa principal, que lleva el control, llamando a cada subrutina, cuando éstas sean necesarias.

El programa principal se inicia al declarar los puntos de referencia de cada presa, esto es, el tamaño, altura, ancho y conexión con la siguiente presa.

Después de declarar dichos puntos se utiliza una bandera, asociada a la subrutina INICIALIZA, la cual se activará si los gráficos no fueron inicializados correctamente y el programa termina (esta acción es lógica, ya que es imposible utilizar gráficos si no se cuenta con los elementos necesarios); en caso contrario permanecerá desactivada permitiendo la ejecución del programa.

Posteriormente se leen, de los archivos con extensión .ent, los volúmenes en las presas restringiéndolos a un cierto valor máximo, de tipo gráfico, esto se hace a través de la subrutina LEE\_VOLUMENES. Además se lee el volumen a turbinar, utilizando una transformación de los valores enteros, ya que estos no se pueden graficar, y por lo tanto, hay que convertirlos a su equivalente gráfico o cadenas de caracteres; lo anterior lo hace la subrutina LEE\_AGUA\_TURBINADA.

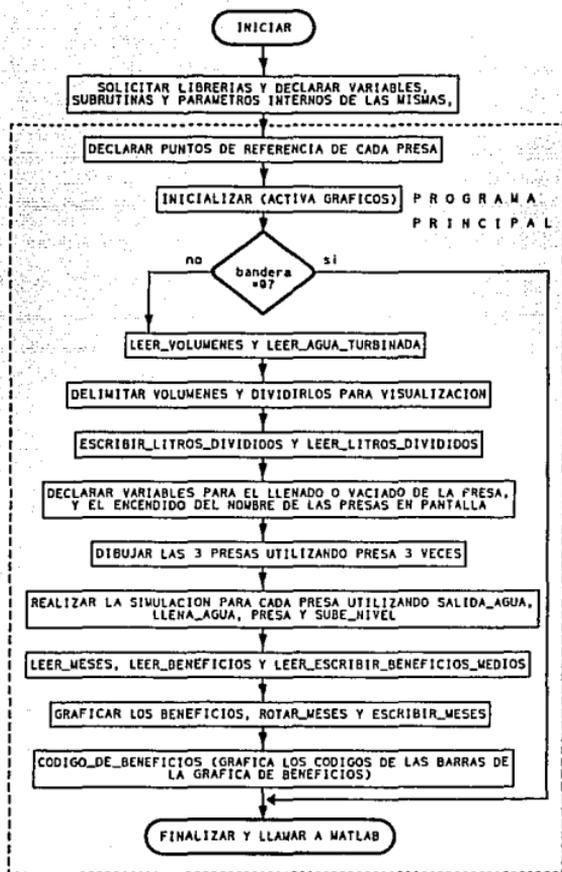


Fig. 4.2 Diagrama general del programa de graficación

El siguiente paso consiste en determinar el nivel del agua en el dibujo asociado al volumen de cada presa, tomando en cuenta los valores máximos permisibles.

La animación del nivel de agua de las presas y de los ríos se realiza dividiendo el dibujo en dos partes a partir de los valores intermedios del volumen de agua en coordenadas del monitor. Esta tarea se ejecuta a través de las rutinas `ESCRIBE_LITROS_DIVIDIDOS` y `LEE_LITROS_DIVIDIDOS`.

Una vez definida la cantidad de agua a dibujar, se declaran las variables que dan color al agua (éste debe ser azul, cuando se quiere llenar la presa o color del fondo del monitor, cuando se indica ausencia de agua) y las que encienden los nombres de cada presa en la pantalla. Una vez hechas estas declaraciones, se dibujan las 3 presas usando la subrutina `PRESA` 3 veces.

Para presentar de manera más real el sistema se vá llenando cada vaso y se vá dibujando el flujo de agua turbinada, ésto se hace combinando las subrutinas `SALIDA_AGUA`, `LLENA_AGUA`, `PRESA` y `SUBE_NIVEL`. Cabe mencionar que este proceso es iterativo y se hace individualmente para cada presa, con la debida variación de sus parámetros. Cuando se debe dibujar una presa casi llena el cálculo del punto intermedio puede quedar fuera del marco de referencia asignado a la presa, es necesario para la animación reajustar los puntos intermedios. Esta tarca es ejecutada por la rutina `LEE_VOLUMENES`.

Para dibujar en pantalla la gráfica de resultados se cargan los 12 meses del año a partir del mes considerado, teniendo así una referencia actual. Esto se hace a través de la subrutina `LEE_MESES`. Otros datos necesarios para dibujar la gráfica son los dos beneficios calculados, el pasado y el real que se está comparando, y que son generados por `MATLAB`. Esto se realiza por medio de las subrutinas `LEE_BENEFICIOS` y `LEE_ESCRIBE_BENEFICIOS_MEDIOS`.

Con los datos anteriores se procede a graficar en forma de barras de diferente textura, los dos beneficios, poniendo un color diferente para cada uno de ellos, por medio de la subrutina `GRAFICA_BENEFICIOS`.

Para terminar, se rota el archivo de los meses y se guarda, a través de las subrutinas `ROTA_MESES` y `ESCRIBE_MESES`. Para indicarle al operador que textura corresponde a cada beneficio se dibuja un recuadro con las texturas asociadas a cada beneficio junto con el valor correspondiente al mes en curso. Esta operación la realiza la subrutina `CODIGO_DE_BENEFICIOS`. Dentro del recuadro se presenta la predicción del beneficio con la política en lazo cerrado.

#### 4.3 SUBROUTINAS UTILIZADAS POR EL PROGRAMA GENERAL

A continuación se presentan las subrutinas utilizadas por el programa amigable de graficación.

#### INTRODUCCIÓN

Debido a que en el lenguaje "C" cuando se está en modo gráfico sólo puede desplegar objetos gráficos o caracteres `ascii` para que aparezcan en el monitor variables del tipo entero, es necesario convertir al entero en su correspondiente cadena de caracteres o "strings". Por tanto se propone guardar las variables del tipo entero en un cierto archivo y leerlas como si estas variables fueran del tipo "string". Así no debe sorprender el ver este tipo de conversión indirecta en distintas partes del programa en C.

##### 4.3.1 SUBROUTINA QUE INICIALIZA GRÁFICOS.

Esta subrutina sirve para inicializar los gráficos dentro del programa,

averiguando el tipo de monitor que se emplea y para enviar un mensaje de error si no se encuentra el archivo de gráficos (.BGI) necesario. Además, abre una ventana de la mitad del monitor, en la parte superior de éste, en donde se presenta el sistema hidráulico, cubriendo el ancho de la pantalla completamente.

#### 4.3.2 SUBROUTINA QUE LEE LOS VOLUMENES ALMACENADOS ESTIMADOS EN EL MES SIGUIENTE Y LOS VOLUMENES TURBINADOS ACTUALES.

Esta subrutina se encarga de abrir un archivo (VOLTURB.ENT), generado desde MATLAB, y el cual tiene los volúmenes predichos para el mes siguiente. En caso de que los pixeles asociados a los volúmenes sean menores que la resolución del monitor se inhibe el dibujo a través de banderas. Además se lee, del mismo archivo, el agua turbinada por las presas en el mes actual, guardando los valores para que puedan ser utilizados posteriormente cuando se dibujen los volúmenes de agua.

#### 4.3.3 SUBROUTINA QUE LEE EL AGUA TURBINADA POR LAS PRESAS EN EL MES ACTUAL.

Esta subrutina guarda en un archivo (TURBINA.ENT) los valores de los volúmenes de agua turbinados por las presas en el mes corriente, los cuales serán leídos como si fueran del tipo cadena de caracteres.

#### 4.3.4 SUBROUTINA QUE DIBUJA UNA PRESA.

Esta subrutina es la encargada de dibujar la cornisa de la presa en la pantalla, junto con el vaso propiamente dicho y el río asociado. Además dibuja los nombres de las presas y los volúmenes turbinados por cada una de ellas. Los comandos para presentar esta información son fijos y

forman parte de la subrutina.

#### 4.3.5 SUBROUTINA QUE DIBUJA LA SALIDA DE AGUA DE LAS PRESAS

Esta subrutina hace la simulación animada del agua que sale por vertederos y que vá directamente a la siguiente presa, se decidió poner animación, para darle mayor realce al programa.

#### 4.3.6 SUBROUTINA QUE DIBUJA EL RELLENADO DE AGUA EN LAS PRESAS.

Esta subrutina hace un dibujo animado tratando de simular la entrada del agua a los vasos, proveniente de la presa anterior y del río.

#### 4.3.7 SUBROUTINA QUE DIBUJA CUANDO SUBE EL NIVEL DE AGUA EN LAS PRESAS.

Esta subrutina dibuja la cantidad de agua de las presas. Una tarea importante de esta subrutina, es la determinación de las coordenadas a partir de las cuales se vá a extender el conjunto de pixeles que dibujan el agua. En particular se utiliza la siguiente fórmula para la determinación del punto de expansión.

$(\text{punto\_final}[0] - \text{altura}/2 - \text{subnivel} / \text{pendiente}) + \text{constantin},$

$(\text{punto\_final}[1] - \text{subnivel}) + 1$

Otra función de esta subrutina es indicar el volumen de agua de cada presa.

#### 4.3.8 SUBROUTINA QUE LEE LOS MESES A PARTIR DEL MES ACTUAL EN EL QUE SE ESTÁ.

Esta subrutina lee los meses del archivo MESES.ENT, caracter por caracter y los va rotando de la misma manera que en el programa principal dentro de MATLAB.

#### 4.3.9 SUBROUTINA QUE ESCRIBE LOS LITROS DIVIDIDOS ENTRE 2.

La simulación animada se hace en dos pasos, para que se vea casi real el movimiento del nivel del agua. Por tanto, es necesario guardar los niveles de agua para poderlos recuperar posteriormente como cadenas. Esta subrutina guarda además los valores de las 3 presas en el archivo LITDIV.ENT. En caso de que el volumen a dibujar exceda los límites, la subrutina restringe los intervalos a rellenar o colorear en las presas.

#### 4.3.10 SUBROUTINA QUE LEE LOS LITROS DIVIDIDOS ENTRE 2.

Esta subrutina lee los litros del archivo LITDIV.ENT, tomándolos como cadenas de caracteres.

#### 4.3.11 SUBROUTINA QUE LEE Y ESCRIBE LOS BENEFICIOS.

Esta subrutina abre el archivo BENEF.ENT, en donde se encuentran los beneficios reales pasados, lee dichos beneficios e incrementa los apuntadores para dejar libre el espacio en donde se almacenará el beneficio real actual, proveniente de la subrutina que lee y escribe los beneficios medios.

#### 4.3.12 SUBROUTINA QUE LEE Y ESCRIBE LOS BENEFICIOS MEDIOS.

Esta subrutina abre el archivo BENMED.ENT, en donde se encuentran los beneficios medios pasados, lee dichos beneficios e incrementa los apuntadores para dejar libre el espacio en donde se almacena el beneficio medio actual. En seguida se abre y se lee el archivo UNBEN.ENT, que contiene los dos beneficios, el real y el medio del mes en cuestión. Estos dos valores se guardan en los espacios correspondientes de BENEF.ENT y BENMED.ENT para futuras corridas. A continuación se abre nuevamente el archivo UNBEN.ENT, para recuperar los valores de los beneficios real y medio, actuales, así como el beneficio real predictorio, con la diferencia de que ahora se leen estos valores como cadenas de characters, para poder imprimir éstos en pantalla. Por último, se escalan los valores pasados de los beneficios, real y medio, para poder dibujarlos en la gráfica.

#### 4.3.13 SUBROUTINA QUE GRAFICA LOS BENEFICIOS DESDE EL MES ACTUAL.

Esta subrutina grafica los beneficios, real y medio mensuales a partir de los últimos 12 meses.

#### 4.3.14 SUBROUTINA QUE ROTA LOS MESES A PARTIR DEL MES ACTUAL.

Esta subrutina se encarga de rotar el renglón de los meses, de la misma manera que el programa principal en MATLAB, manteniendo sincronía con él.

#### 4.3.15 SUBROUTINA QUE ESCRIBE LOS MESES A PARTIR DEL MES ACTUAL.

Esta subrutina abre el archivo MESES.ENT y guarda el vector de los

meses, una vez que fueron rotados.

#### **4.3.16 SUBROUTINA QUE GRAFICA LOS CÓDIGOS Y VALORES DE LOS BENEFICIOS.**

Esta subrutina se encarga de hacer el recuadro, en el que se indican los valores de los beneficios actuales comparados, junto con la predicción del beneficio con la política en lazo cerrado. Además, junto a cada uno de ellos se encuentra un cuadro con un color, éste cuadro, para los 2 primeros casos significa el color que los identifica en la gráfica, para poder diferenciarlos, de ésta manera, podemos identificarlos fácilmente. Al beneficio predictorio se le pone un color blanco, que pudo haber sido cualquier otro, únicamente para diferenciarlo de los otros.

```

/***** PROGRAMA PRINCIPAL *****/
/** ESTE PROGRAMA MANDARA LLAMAR LAS SUBROUTINAS NECESARIAS PARA CREAR */
/***** LOS GRAFICOS DE LAS PRESAS EN CASCADA. *****/

```

```

#include <stdio.h>
#include <graphics.h>
#include <math.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>

#define CLIP_ON 1

int punto_ini_angostu[2], punto_med_angostu[2], punto_fin_angostu[2],
altura_angostu;
int punto_ini_chicoas[2], punto_med_chicoas[2], punto_fin_chicoas[2],
altura_chicoas;
int punto_ini_malpaso[2], punto_med_malpaso[2], punto_fin_malpaso[2],
altura_malpaso;
int veces_angostu, veces_chicoas, veces_malpaso;
float pendiente_angostu, pendiente_chicoas, pendiente_malpaso;
float pend_presa_angostu, pend_presa_chicoas, pend_presa_malpaso;
float aux_punto_med_angostu[2], aux_punto_fin_angostu[2];
float aux_punto_med_chicoas[2], aux_punto_fin_chicoas[2];
float aux_punto_med_malpaso[2], aux_punto_fin_malpaso[2];
float nivel_angostu, nivel_chicoas, nivel_malpaso;
float subnivel_angostu, subnivel_chicoas, subnivel_malpaso;
int punto_aux_angostu[2], veces_aux_angostu;
int punto_aux_chicoas[2], veces_aux_chicoas;
int punto_aux_malpaso[2], veces_aux_malpaso;
float pendiente_aux_angostu;
float pendiente_aux_chicoas;
float pendiente_aux_malpaso;
int color_sal_angostu, color_sal_chicoas, color_sal_malpaso;
int cont_angostu, cont_chicoas, cont_malpaso;
float litros_angostu, litros_chicoas, litros_malpaso;
float lit_max_angostu, lit_max_chicoas, lit_max_malpaso;
char nombre[20], nom_presa_angostu[10], nom_presa_chicoas[10],
nom_presa_malpaso[10];
char turbinin[20];
float turbinin_presa_angostu, turbinin_presa_chicoas,
turbinin_presa_malpaso;
char turbi_presa_angostu[5], turbi_presa_chicoas[5],
turbi_presa_malpaso[5];
char volumen[20];
float vol_angostu[10], vol_chicoas[10], vol_malpaso[10];
char vol_presa_angostu[10][15], vol_presa_chicoas[10][15];
char vol_presa_malpaso[10][15], vol_aux_presa_chicoas[15];
int bandera;
int polig[8];
int turbina[8];
int vertedero[8];
int ola_de_agua[10];

```

```

unsigned int size;
void *ola;
int mueve, contador;
int escurre_de_agua[8];
unsigned int size_escurre;
void *escurre;
int aux, sube_agua;
FILE *nom_arch;
char msg[12];
char mes[12][12];
char auxmes[12];
float tam[12];
float tam_med[12];
char tam_grafica[5];
char tam_med_grafica[5];
char tam_real_grafica[5];
char beneficiin[30], mensajin[60];
float auxilit_angostu, auxilit_chicoas, auxilit_malpaso;
int banda_angostu, banda_chicoas, banda_malpaso;
int constantin;
float escalador;

int INICIALIZA(void);
void LEE_VOLUMENES();
void LEE_AGUA_TURBINADA();
void PRESA(int punto_inicial[2], int punto_medio[2],
           int punto_final[2], int altura, int color_sal,
           char nom_presa[10], char turbi_presa[5]);
void SALIDA_AGUA(int punto_inicial[2], int altura);
void LLENA_AGUA(int punto_medio[2], int altura, float pendiente,
               int veces);
void SUBE_NIVEL(int punto_final[2], int altura, float pendiente,
               float pend_presa, float subnivel, char vol_presa[10|15],
               int cont);

void LEE_MESES();
void ESCRIBE_LITROS_DIVIDIDOS();
void LEE_LITROS_DIVIDIDOS();
void LEE_ESCRIBE_BENEFICIOS();
void LEE_ESCRIBE_BENEFICIOS_MEDIOS();
void GRAFICA_BENEFICIOS();
void ROTA_MESES();
void ESCRIBE_MESES();
void CODIGO_DE_BENEFICIOS();

main()
{
    punto_ini_angostu[0] = 5;
    punto_ini_angostu[1] = 40;
    punto_med_angostu[0] = punto_ini_angostu[0] + 20;
    punto_med_angostu[1] = punto_ini_angostu[1];
    punto_fin_angostu[0] = 280;
    punto_fin_angostu[1] = 120;

```

```

aux_punto_med_angostu[0] = punto_med_angostu[0];
aux_punto_med_angostu[1] = punto_med_angostu[1];
aux_punto_fin_angostu[0] = punto_fin_angostu[0];
aux_punto_fin_angostu[1] = punto_fin_angostu[1];
altura_angostu = 70;
veces_angostu = 39;
pendiente_angostu = (aux_punto_fin_angostu[1] -
aux_punto_med_angostu[1]) /
(aux_punto_fin_angostu[0] - altura_angostu / 2
- aux_punto_med_angostu[0]);
pend_presa_angostu = ((aux_punto_fin_angostu[1] - altura_angostu) -
aux_punto_fin_angostu[1]) /
(((aux_punto_fin_angostu[0] - altura_angostu /
6) - aux_punto_fin_angostu[0] - altura_angostu
/ 2));
punto_ini_chicoas[0] = punto_fin_angostu[0];
punto_ini_chicoas[1] = punto_fin_angostu[1];
punto_med_chicoas[0] = punto_fin_angostu[0] + 20;
punto_med_chicoas[1] = punto_fin_angostu[1];
punto_fin_chicoas[0] = 360;
punto_fin_chicoas[1] = 200;
aux_punto_med_chicoas[0] = punto_med_chicoas[0];
aux_punto_med_chicoas[1] = punto_med_chicoas[1];
aux_punto_fin_chicoas[0] = punto_fin_chicoas[0];
aux_punto_fin_chicoas[1] = punto_fin_chicoas[1];
altura_chicoas = 60;
veces_chicoas = 39;
pendiente_chicoas = (aux_punto_fin_chicoas[1] -
aux_punto_med_chicoas[1]) /
(aux_punto_fin_chicoas[0] - altura_chicoas / 2
- aux_punto_med_chicoas[0]);
pend_presa_chicoas = ((aux_punto_fin_chicoas[1] - altura_chicoas) -
aux_punto_fin_chicoas[1]) /
(((aux_punto_fin_chicoas[0] - altura_chicoas /
6) - aux_punto_fin_chicoas[0] - altura_chicoas
/ 2));
punto_ini_malpaso[0] = punto_fin_chicoas[0];
punto_ini_malpaso[1] = punto_fin_chicoas[1];
punto_med_malpaso[0] = punto_fin_chicoas[0] + 20;
punto_med_malpaso[1] = punto_fin_chicoas[1];
punto_fin_malpaso[0] = 588;
punto_fin_malpaso[1] = 260;
aux_punto_med_malpaso[0] = punto_med_malpaso[0];
aux_punto_med_malpaso[1] = punto_med_malpaso[1];
aux_punto_fin_malpaso[0] = punto_fin_malpaso[0];
aux_punto_fin_malpaso[1] = punto_fin_malpaso[1];
altura_malpaso = 50;
veces_malpaso = 29;
pendiente_malpaso = (aux_punto_fin_malpaso[1] -
aux_punto_med_malpaso[1]) /
(aux_punto_fin_malpaso[0] - altura_malpaso / 2
- aux_punto_med_malpaso[0]);

```

```

pend_presa_chicoas = ((aux_punto_fin_malpaso[1] - altura_malpaso) -
aux_punto_fin_malpaso[1]) /
((aux_punto_fin_malpaso[0] - altura_malpaso /
6) - aux_punto_fin_malpaso[0] - altura_malpaso
/ 2));

```

```

bandera=INICIALIZA();

```

```

if(bandera==0)

```

```

{
    LEE_VOLUMENES();
    LEE_AGUA_TURBINADA();
    lit_max_angostu = 18656.70;
    lit_max_chicoas = 1703.80;
    lit_max_malpaso = 14846.40;
    ESCRIBE_LITROS_DIVIDIDOS();
    LEE_LITROS_DIVIDIDOS();
    color_sal_angostu = 0;
    color_sal_chicoas = 0;
    color_sal_malpaso = 0;
    strcpy(nom_presa_angostu,"ANGOSTURA");
    strcpy(nom_presa_chicoas,"CHICOASEN");
    strcpy(nom_presa_malpaso,"MALPASO");
    PRESA(punto_ini_angostu, punto_med_angostu, punto_fin_angostu,
altura_angostu, color_sal_angostu, nom_presa_angostu,
turbi_presa_angostu);
    PRESA(punto_ini_chicoas, punto_med_chicoas, punto_fin_chicoas,
altura_chicoas, color_sal_chicoas, nom_presa_chicoas,
turbi_presa_chicoas);
    PRESA(punto_ini_malpaso, punto_med_malpaso, punto_fin_malpaso,
altura_malpaso, color_sal_malpaso, nom_presa_malpaso,
turbi_presa_malpaso);
    nivel_angostu = (litros_angostu * 70) / lit_max_angostu;
    subnivel_angostu = nivel_angostu / 2;
    nivel_chicoas = (litros_chicoas * 60) / lit_max_chicoas;
    subnivel_chicoas = nivel_chicoas / 2;
    nivel_malpaso = (litros_malpaso * 50) / lit_max_malpaso;
    subnivel_malpaso = nivel_malpaso/2;
    sube_agua = 0;
    cont_angostu = 0;
    cont_chicoas = 0;
    cont_malpaso = 0;
    for(aux=0;aux<=7;aux++) /* Llena las presas al mismo tiempo */
    {
        sube_agua+=1;
        SALIDA_AGUA(punto_ini_angostu, altura_angostu);
        LLENA_AGUA(punto_med_angostu, altura_angostu,
pendiente_angostu, veces_angostu);
        if(sube_agua==4)
        {
            if(subnivel_angostu>70)
            {
                subnivel_angostu = 69;

```

```

        color_sal_angostu = 1;
        PRESA(punto_ini_angostu, punto_med_angostu,
            punto_fin_angostu, altura_angostu,
            color_sal_angostu, nom_presa_angostu,
            turbi_presa_angostu);
        setfillstyle(SOLID_FILL, BLUE);
    }
    else
    {
        if(subnivel_angostu > (altura_angostu -
            altura_angostu / 6))
        {
            color_sal_angostu = 1;
            PRESA(punto_ini_angostu, punto_med_angostu,
                punto_fin_angostu, altura_angostu,
                color_sal_angostu, nom_presa_angostu,
                turbi_presa_angostu);
            setfillstyle(SOLID_FILL, BLUE);
        }
    }
    strcpy(vol_presa_chicoas[0], vol_aux_presa_chicoas);
    SUBE_NIVEL(punto_fin_angostu, altura_angostu,
        pendiente_angostu, pend_presa_angostu,
        subnivel_angostu, vol_presa_angostu,
        cont_angostu);
    subnivel_angostu += nivel_angostu / 2;
    cont_angostu + = 1;
}
SALIDA_AGUA(punto_fin_angostu, altura_angostu);
LLENA_AGUA(punto_med_chicoas, altura_chicoas,
    pendiente_chicoas, veces_chicoas);
if(sube_agua==4)
{
    if(subnivel_chicoas>60)
    {
        subnivel_chicoas = 59;
        color_sal_chicoas = 1;
        PRESA(punto_ini_chicoas, punto_med_chicoas,
            punto_fin_chicoas, altura_chicoas,
            color_sal_chicoas, nom_presa_chicoas,
            turbi_presa_chicoas);
        setfillstyle(SOLID_FILL, BLUE);
    }
    else
    {
        if(subnivel_chicoas > (altura_chicoas -
            altura_chicoas / 6))
        {
            color_sal_chicoas = 1;
            PRESA(punto_ini_chicoas, punto_med_chicoas,
                punto_fin_chicoas, altura_chicoas,
                color_sal_chicoas, nom_presa_chicoas,

```

```

        turbi_presa_chicoas);
        setfillstyle(SOLID_FILL, BLUE);
    }
}
strcpy(vol_presa_chicoas[0], vol_aux_presa_chicoas);
SUBE_NIVEL(punto_fin_chicoas, altura_chicoas,
pendiente_chicoas, pend_presa_chicoas,
subnivel_chicoas, vol_presa_chicoas,
cont_chicoas);
subnivel_chicoas + = nivel_chicoas / 2;
cont_chicoas+=1;
}
SALIDA_AGUA(punto_fin_chicoas, altura_chicoas);
LLENA_AGUA(punto_med_malpaso, altura_malpaso,
pendiente_malpaso, veces_malpaso);
if(sube_agua==4)
{
    if(subnivel_malpaso>50)
    {
        subnivel_malpaso = 49.5;
        color_sal_malpaso = 1;
        PRESA(punto_ini_malpaso, punto_med_malpaso,
punto_fin_malpaso, altura_malpaso,
color_sal_malpaso, nom_presa_malpaso,
turbi_presa_malpaso);
        setfillstyle(SOLID_FILL, BLUE);
    }
    else
    {
        if(subnivel_malpaso > (altura_malpaso -
altura_malpaso / 6))
        {
            color_sal_malpaso = 1;
            PRESA(punto_ini_malpaso, punto_med_malpaso,
punto_fin_malpaso, altura_malpaso,
color_sal_malpaso, nom_presa_malpaso,
turbi_presa_malpaso);
            setfillstyle(SOLID_FILL, BLUE);
        }
    }
    strcpy(vol_presa_chicoas[0], vol_aux_presa_chicoas);
    SUBE_NIVEL(punto_fin_malpaso, altura_malpaso,
pendiente_malpaso, pend_presa_malpaso,
subnivel_malpaso, vol_presa_malpaso,
cont_malpaso);
    subnivel_malpaso + = nivel_malpaso / 2;
    cont_malpaso + = 1;
    sube_agua = 0;
}
} /*Termina de llenarlas, y ve cual de ellas se sobrepasa */
sube_agua=0;
while(color_sal_angostu!=0)
{

```

```

sube_agua + = 1;
if(color_sal_angostu==1)
{
    punto_aux_chicoas[0] = punto_fin_angostu[0] +
        altura_angostu / 12;
    punto_aux_chicoas[1] = punto_fin_angostu[1] -
        altura_angostu + altura_angostu /
        6;
    pendiente_aux_chicoas = altura_angostu / (20 -
        altura_angostu / 12);
    veces_aux_chicoas = 30;
    LLENA_AGUA(punto_aux_chicoas, altura_angostu,
        pendiente_aux_chicoas, veces_aux_chicoas);
}
if(sube_agua==4)
{
    if(color_sal_angostu==1)
    {
        subnivel_angostu - = nivel_angostu / 2;
        cont_angostu - = 1;
        setfillstyle(SOLID_FILL, BLACK);
        SUBE_NIVEL(punto_fin_angostu, altura_angostu,
            pendiente_angostu, pend_presa_angostu,
            subnivel_angostu, vol_presa_angostu,
            cont_angostu);
        setfillstyle(SOLID_FILL, BLUE);
        subnivel_angostu - = nivel_angostu/2;
        cont_angostu - = 1;
        strcpy(volumen, "VOLUMEN = ");
        strncat(volumen, vol_presa_angostu[cont_angostu+1],
            5);
        setcolor(BLACK);
        settxtstyle(SMALL_FONT, HORIZ_DIR, 4);
        outtextxy(punto_fin_angostu[0] - (altura_angostu),
            punto_fin_angostu[1] + altura_angostu / 4,
            volumen);
        subnivel_angostu = altura_angostu - altura_angostu /
            6;
        setcolor(BROWN);
        constantin=7;
        strcpy(vol_presa_angostu, "15549");
        SUBE_NIVEL(punto_fin_angostu, altura_angostu,
            pendiente_angostu, pend_presa_angostu,
            subnivel_angostu, vol_presa_angostu,
            cont_angostu);
        constantin=5;
        if(subnivel_angostu <= (altura_angostu -
            altura_angostu / 6))
        {
            color_sal_angostu = 0;
            PRESA(punto_ini_angostu, punto_med_angostu,
                punto_fin_angostu, altura_angostu,

```

```

        color_sal_angostu, nom_presa_angostu,
        turbl_presa_angostu);
        setfillstyle(SOLID_FILL, BLUE);
    }
    subnivel_angostu + = nivel_angostu / 2;
    cont_angostu + = 1;
}
sube_agua=0;
}
}
sube_agua=0;
while(color_sal_chicoas!=0)
{
    sube_agua+=1;
    if(color_sal_chicoas==1)
    {
        punto_aux_malpaso[0] = punto_fin_chicoas[0] +
            altura_chicoas / 12;
        punto_aux_malpaso[1] = punto_fin_chicoas[1] -
            altura_chicoas + altura_chicoas /
            6;
        pendiente_aux_malpaso = altura_chicoas / (20 -
            altura_chicoas / 12);
        veces_aux_malpaso = 23;
        LLENA_AGUA(punto_aux_malpaso, altura_chicoas,
            pendiente_aux_malpaso, veces_aux_malpaso);
    }
    if(sube_agua==4)
    {
        if(color_sal_chicoas==1)
        {
            subnivel_chicoas - = nivel_chicoas / 2;
            cont_chicoas - = 1;
            setfillstyle(SOLID_FILL, BLACK);
            SUBE_NIVEL(punto_fin_chicoas, altura_chicoas,
                pendiente_chicoas, pend_presa_chicoas,
                subnivel_chicoas, vol_presa_chicoas,
                cont_chicoas);
            setfillstyle(SOLID_FILL, BLUE);
            subnivel_chicoas - = nivel_chicoas / 2;
            cont_chicoas - = 1;
            strcpy(volumen, "VOLUMEN = ");
            strncat(volumen, vol_presa_chicoas[cont_chicoas+1],
                5);
            setcolor(BLACK);
            setttextstyle(SMALL_FONT, HORIZ_DIR, 4);
            outtextxy(punto_fin_chicoas[0] - (altura_chicoas),
                punto_fin_chicoas[1] + altura_chicoas / 4,
                volumen);
            subnivel_chicoas = altura_chicoas - altura_chicoas /
                6;
            setcolor(BROWN);

```

```

constantin = 7;
strcpy(vol_presa_chicoas, "1419");
SUBE_NIVEL(punto_fin_chicoas, altura_chicoas,
pendiente_chicoas, pend_presa_chicoas,
subnivel_chicoas, vol_presa_chicoas,
cont_chicoas);
constantin=5;
if(subnivel_chicoas <= (altura_chicoas -
altura_chicoas / 6))
{
color_sal_chicoas = 0;
PRESA(punto_ini_chicoas, punto_med_chicoas,
punto_fin_chicoas, altura_chicoas,
color_sal_chicoas, nom_presa_chicoas,
turbi_presa_chicoas);
setfillstyle(SOLID_FILL, BLUE);
}
subnivel_chicoas += nivel_chicoas / 2;
cont_chicoas += 1;
}
sube_agua=0;
}
}
sube_agua=0;
while(color_sal_malpaso!=0)
{
sube_agua += 1;
if(color_sal_malpaso==1)
{
punto_aux_angostu[0] = punto_fin_malpaso[0] +
altura_malpaso / 12;
punto_aux_angostu[1] = punto_fin_malpaso[1] -
altura_malpaso + altura_malpaso /
6;
pendiente_aux_angostu = altura_malpaso / (20 -
altura_malpaso / 12);
veces_aux_angostu = 23;
SALIDA_AGUA(punto_fin_malpaso, altura_malpaso);
LLENA_AGUA(punto_aux_angostu, altura_malpaso,
pendiente_aux_angostu, veces_aux_angostu);
}
if(sube_agua==4)
{
if(color_sal_malpaso==1)
{
subnivel_malpaso -= nivel_malpaso / 2;
cont_malpaso -= 1;
setfillstyle(SOLID_FILL, BLACK);
SUBE_NIVEL(punto_fin_malpaso, altura_malpaso,
pendiente_malpaso, pend_presa_malpaso,
subnivel_malpaso, vol_presa_malpaso,
cont_malpaso);
}
}
}

```

```

setfillstyle(SOLID_FILL,BLUE);
subnivel_malpaso = nivel_malpaso / 2;
cont_malpaso = 1;
strcpy(volumen,"VOLUMEN = ");
strncat(volumen, vol_presa_malpaso[cont_malpaso+1],
5);
setcolor(BLACK);
settextstyle(SMALL_FONT,HORIZ_DIR,4);
outtextxy(punto_fin_malpaso[0] - (altura_malpaso),
punto_fin_malpaso[1] + altura_malpaso / 4,
volumen);
subnivel_malpaso = altura_malpaso - altura_malpaso /
6;
setcolor(BROWN);
constantin = 7;
strcpy(vol_presa_malpaso,"12373");
SUBE_NIVEL(punto_fin_malpaso, altura_malpaso,
pendiente_malpaso, pend_presa_malpaso,
subnivel_malpaso, vol_presa_malpaso,
cont_malpaso);
constantin=5;
if(subnivel_malpaso <= (altura_malpaso -
altura_malpaso / 6))
{
color_sal_malpaso = 0;
PRESA(punto_ini_malpaso, punto_med_malpaso,
punto_fin_malpaso, altura_malpaso,
color_sal_malpaso, nom_presa_malpaso,
turbi_presa_malpaso);
setfillstyle(SOLID_FILL,BLUE);
}
subnivel_malpaso += nivel_malpaso / 2;
cont_malpaso += 1;
}
}
sube_agua = 0;
} /* Terminó de ver cual de las presas se sobrepasa */
LEE_MESES();
setcolor(LIGHTGREEN);
strcpy(mensaje,"PREDICCIÓN DE VOLUMENES EN EL MES DE ");
strcat(mensaje,mes[11]);
outtextxy(130,16,mensaje);
setcolor(BROWN);
LEE_ESCRIBE_BENEFICIOS();
LEE_ESCRIBE_BENEFICIOS_MEDIOS();
GRAFICA_BENEFICIOS();
ROTA_MESES();
ESCRIBE_MESES();
CODIGO_DE_BENEFICIOS();
}
}

```

```

.....
/*          SUBROUTINA QUE INICIALIZA GRAFICOS          */
.....

```

```

int INICIALIZA(void)
{
    int gdriver = DETECT,gmode,errorcode;
    initgraph(&gdriver,&gmode,"");
    errorcode = graphresult();
    if(errorcode != grOk)
    {
        printf("Error grafico:%s", grapherrormsg(errorcode));
        printf("Aprieta cualquier tecla para terminar:");
        getch();
        exit(1);
    }
    setviewport(2,289,637,477,CLIP_ON);
    setcolor(YELLOW);
    rectangle(0,0,635,188);
    setviewport(2,2,637,287,CLIP_ON);
    rectangle(0,0,635,285);
    return 0;
}

```

```

.....
/*          SUBROUTINA QUE LEE LOS VOLUMENES ALMACENADOS ESTIMADOS EN EL MES          */
/*          SIGUIENTE Y LOS VOLUMENES TURBINADOS ACTUALES          */
.....

```

```

void LEE_VOLUMENES()
{
    banda_angostu = 0;
    banda_chicoas = 0;
    banda_malpasso = 0;
    constantin = 5;
    nom_arch = fopen("C:\VOLTURB.ENT", "r");
    fscanf(nom_arch,"%f",&litros_angostu);
    if(litros_angostu < 1680.00)
    {
        constantin = 4;
        auxil1t_angostu = litros_angostu;
        litros_angostu = 1680;
        banda_angostu = 1;
    }
    fscanf(nom_arch,"%f",&litros_chicoas);
    if(litros_chicoas < 730.00)
    {
        constantin = 4;
        auxil1t_chicoas = litros_chicoas;
        litros_chicoas = 730;
        banda_chicoas = 1;
    }
    fscanf(nom_arch,"%f",&litros_malpasso);
    if(litros_malpasso < 1950.00)
    {

```

```

        constantin=4;
        auxiliit_malpasso = litros_malpasso;
        litros_malpasso = 1950;
        banda_malpasso = 1;
    }
    fscanf(nom_arch, "%f", &turbinin_presa_angostu);
    fscanf(nom_arch, "%f", &turbinin_presa_chicoas);
    fscanf(nom_arch, "%f", &turbinin_presa_malpasso);
    fclose(nom_arch);
}

/*****
/*          SUBROUTINA QUE LEE EL AGUA TURBINADA POR LAS PRESAS          */
/*          EN EL MES ACTUAL                                          */
*****/
void LEE_AGUA_TURBINADA()
{
    nom_arch = fopen("C:TURBINA.ENT", "w");
    fprintf(nom_arch, "%f", turbinin_presa_angostu);
    fprintf(nom_arch, "%f", turbinin_presa_chicoas);
    fprintf(nom_arch, "%f", turbinin_presa_malpasso);
    fclose(nom_arch);
    nom_arch = fopen("C:TURBINA.ENT", "r");
    fscanf(nom_arch, "%s", turbi_presa_angostu);
    fscanf(nom_arch, "%s", turbi_presa_chicoas);
    fscanf(nom_arch, "%s", turbi_presa_malpasso);
    fclose(nom_arch);
}

/*****
/*          SUBROUTINA QUE DIBUJA UNA PRESA                          */
*****/
void PRESA(int punto_inicial[2], int punto_medio[2],
           int punto_final[2], int altura, int color_sal,
           char nom_presa[10], char turbi_presa[20])
{
    setcolor(BROWN);
    polig[0] = punto_final[0];
    polig[1] = punto_final[1];
    polig[2] = punto_final[0];
    polig[3] = punto_final[1] - altura;
    polig[4] = punto_final[0] - altura/6;
    polig[5] = punto_final[1] - altura;
    polig[6] = punto_final[0] - altura/2;
    polig[7] = punto_final[1];
    turbina[0] = punto_final[0] + altura/12;
    turbina[1] = punto_final[1] - altura/12;
    turbina[2] = punto_final[0] + altura/12;
    turbina[3] = punto_final[1] - altura/12 - altura/6;
    turbina[4] = punto_final[0];
    turbina[5] = punto_final[1] - altura/12 - altura/6;
    turbina[6] = punto_final[0];
}

```

```

turbina[7] = punto_final[1] - altura/12;
vertedero[0] = punto_final[0] + altura/12;
vertedero[1] = punto_final[1] - altura + altura/6;
vertedero[2] = punto_final[0] + altura/12;
vertedero[3] = punto_final[1] - altura + altura/12;
vertedero[4] = punto_final[0];
vertedero[5] = punto_final[1] - altura + altura/12;
vertedero[6] = punto_final[0];
vertedero[7] = punto_final[1] - altura + altura/6;
setfillstyle(SOLID_FILL,LIGHTGRAY);
fillpoly(4,polig);
setfillstyle(SOLID_FILL,BLUE);
if(color_sal==0)
{
    fillpoly(4,turbina);
    setfillstyle(SOLID_FILL,YELLOW);
}
else
{
    fillpoly(4,turbina);
    setfillstyle(SOLID_FILL,RED);
}
fillpoly(4,vertedero);
line(punto_inicial[0], punto_inicial[1], punto_medio[0],
      punto_medio[1]);
line(punto_medio[0], punto_medio[1], punto_final[0] - altura / 2,
      punto_final[1]);
strcpy(nombre,nom_presa);
strcpy(turbinin,"VOL. TURBINADO = ");
strcpy(turbi_presa, strtok(turbi_presa, "."));
strcat(turbinin, turbi_presa);
setcolor(YELLOW);
settextstyle(SMALL_FONT,HORIZ_DIR,4);
outtextxy(punto_final[0] - (altura * 3.7), punto_final[1] +
          altura / 4, turbinin);
outtextxy(punto_final[0] - (altura/2), punto_medio[1] - 7, nombre);
setcolor(BROWN);
}

```

```

.....
/* SUBROUTINA QUE DIBUJA LA SALIDA DE AGUA DE LAS PRESAS */
.....
void SALIDA_AGUA(int punto_inicial[2], int altura)

```

```

{
    mueve = 0;
    ola_de_agua[0] = punto_inicial[0] + altura/6;
    ola_de_agua[1] = punto_inicial[1];
    ola_de_agua[2] = punto_inicial[0] + altura/6;
    ola_de_agua[3] = punto_inicial[1] - altura/12 - altura/6;
    ola_de_agua[4] = punto_inicial[0] + altura/6 - altura/24;
    ola_de_agua[5] = punto_inicial[1] - altura/12 - altura/6 -
        altura/24;
}

```

```

ola_de_agua[6] = punto_inicial[0] + altura/12;
ola_de_agua[7] = punto_inicial[1] - altura/12 - altura/6;
ola_de_agua[8] = punto_inicial[0] + altura/12;
ola_de_agua[9] = punto_inicial[1];
setcolor(BLUE);
setfillstyle(SOLID_FILL, BLUE);
fillpoly(5, ola_de_agua);
setcolor(BROWN);
size = imagesize(punto_inicial[0] + altura/12, punto_inicial[1] -
    altura/12 - altura/6 - altura/24, punto_inicial[0] +
    altura/6, punto_inicial[1]);
ola = malloc(size);
getimage(punto_inicial[0] + altura/12, punto_inicial[1] - altura/12
    - altura/6 - altura/24, punto_inicial[0] + altura/6,
    punto_inicial[1], ola);
putimage(punto_inicial[0] + altura/12 + mueve, punto_inicial[1] -
    altura/12 - altura/6 - altura/24, ola, XOR_PUT);
for(contador=0; contador<=3; contador++)
{
    delay(20);
    putimage(punto_inicial[0] + altura/12 + mueve, punto_inicial[1]
        - altura/12 - altura/6 - altura/24, ola, XOR_PUT);
    mueve + = altura/12;
}
}

/.....*/
/*          SUBROUTINA QUE DIBUJA EL RELLENADO DE AGUA EN LAS PRESAS          */
/.....*/
void LLENA_AGUA(int punto_medio[2], int altura, float pendiente,
    int veces)
{
    float ordenada;
    mueve = 0;
    ordenada = punto_medio[1] - (pendiente * punto_medio[0]);
    escurre_de_agua[0] = (punto_medio[1] - altura/24 - ordenada) /
        pendiente;
    escurre_de_agua[1] = punto_medio[1] - altura/24;
    escurre_de_agua[2] = punto_medio[0];
    escurre_de_agua[3] = punto_medio[1];
    escurre_de_agua[4] = (punto_medio[1] - altura/24 - ordenada) /
        pendiente + (2 * (escurre_de_agua[2] -
            escurre_de_agua[0]));
    escurre_de_agua[5] = punto_medio[1] - altura/24;
    escurre_de_agua[6] = punto_medio[0];
    escurre_de_agua[7] = punto_medio[1] - altura/12;
    setcolor(BLUE);
    setfillstyle(SOLID_FILL, BLUE);
    fillpoly(4, escurre_de_agua);
    setcolor(BROWN);
    size_escurre = imagesize(escurre_de_agua[0], escurre_de_agua[3],
        escurre_de_agua[4], escurre_de_agua[7]);
}

```

```

escurre = malloc(size_escurre);
getimage(escurre_de_agua[0], escurre_de_agua[7], escurre_de_agua[4],
         escurre_de_agua[3], escurre);
putimage((punto_medio[1] - altura/24 - ordenada + mueve) /
         pendiente, punto_medio[1] - altura/12 + mueve, escurre,
         XOR_PUT);
for(contador=0;contador<=veces;contador++)
{
    delay(15);
    putimage((punto_medio[1] - altura/24 - ordenada + mueve) /
            pendiente, punto_medio[1] - altura/12 + mueve, escurre
            XOR_PUT);
    mueve + = altura/24;
}
}

/...../
/* SUBROUTINA QUE DIBUJA CUANDO SUBE EL NIVEL DE AGUA EN LAS PRESAS */
/...../
void SUBE_NIVEL(int punto_final[2], int altura, float pendiente,
               float pend_presa, float subnivel,
               char vol_presa[10][15], int cont);
{
    line(punto_final[0] - altura / 2 - subnivel / pendiente,
        punto_final[1] - subnivel, punto_final[0] - altura/2 -
        subnivel / pend_presa, punto_final[1] - subnivel);
    floodfill((punto_final[0] - altura/2 - subnivel / pendiente) +
             constantin, (punto_final[1] - subnivel) + 1, BROWN);
    setcolor(BLACK);
    line(punto_final[0] - altura / 2 - subnivel / pendiente + 3,
        punto_final[1] - subnivel, punto_final[0]
        - altura / 2 - subnivel / pend_presa - 2,
        punto_final[1] - subnivel);
    strcpy(volumen,"VOLUMEN = ");
    strncpy(volumen,vol_presa[cont-1],5);
    setcolor(BLACK);
    settextstyle(SMALL_FONT,HORIZ_DIR,4);
    outtextxy(punto_final[0]-(altura),punto_final[1]+altura/4,volumen);
    strcpy(volumen,"VOLUMEN = ");
    strncpy(volumen,vol_presa[cont],5);
    setcolor(YELLOW);
    settextstyle(SMALL_FONT,HORIZ_DIR,4);
    outtextxy(punto_final[0]-(altura),punto_final[1]+altura/4,volumen);
    setcolor(BROWN);
}

/...../
/* SUBROUTINA QUE LEE LOS MESES A PARTIR DEL MES ACTUAL */
/* EN EL QUE SE ESTA */
/...../
void LEE_MESES()
{

```

```

nom_arch = fopen("C:MESES.ENT","r");
for(aux=0;aux<12;aux++)
{
    while(msg[1]!=';')
    {
        fgets(msg,11,nom_arch);
        if(msg[1]!=';')
        {
            strcpy(mes[aux],msg);
        }
    }
    msg[1] = '0';
}
fclose(nom_arch);
}

```

```

/*****
/*      SUBROUTINA QUE ESCRIBE LOS LITROS DIVIDIDOS ENTRE 2
*****/
void ESCRIBE_LITROS_DIVIDIDOS()

```

```

{
    nom_arch = fopen("C:LITDIV.ENT", "w");
    if(banda_angostu != 1)
    {
        vol_angostu[0] = litros_angostu / 2;
    }
    else
    {
        vol_angostu[0] = auxilit_angostu / 2;
    }
    if(banda_chicoas != 1)
    {
        vol_chicoas[0] = litros_chicoas / 2;
    }
    else
    {
        vol_chicoas[0] = auxilit_chicoas / 2;
    }
    if(banda_malpasso != 1)
    {
        vol_malpasso[0] = litros_malpasso / 2;
    }
    else
    {
        vol_malpasso[0] = auxilit_malpasso / 2;
    }
    for(aux=0;aux<2;aux++)
    {
        fprintf(nom_arch,"%f,",vol_angostu[aux]);
        fprintf(nom_arch,"%f,",vol_chicoas[aux]);
        fprintf(nom_arch,"%f,",vol_malpasso[aux]);
        if(aux<1)
        {

```

```

        if(banda_angostu != 1)
        {
            vol_angostu[aux+i]=vol_angostu[aux]+litros_angostu/2;
        }
        else
        {
            vol_angostu[aux+i]=vol_angostu[aux]+auxilit_angostu/2;
        }
        if(banda_chicoas != 1)
        {
            vol_chicoas[aux+i]=vol_chicoas[aux]+litros_chicoas/2;
        }
        else
        {
            vol_chicoas[aux+i]=vol_chicoas[aux]+auxilit_chicoas/2;
        }
        if(banda_angostu != 1)
        {
            vol_malpaso[aux+i]=vol_malpaso[aux]+litros_malpaso/2;
        }
        else
        {
            vol_malpaso[aux+i]=vol_malpaso[aux]+auxilit_malpaso/2;
        }
    }
}
fclose(nom_arch);
}

```

```

/...../
/*          SUBROUTINA QUE LEE LOS LITROS DIVIDIDOS ENTRE 2          */
/...../
void LEE_LITROS_DIVIDIDOS()
{
    nom_arch = fopen("C:LITDIV.ENT", "r");
    for(aux=0;aux<2;aux++)
    {
        fscanf(nom_arch,"%s",vol_presa_angostu[aux]);
        fscanf(nom_arch,"%s",vol_presa_chicoas[aux]);
        fscanf(nom_arch,"%s",vol_presa_malpaso[aux]);
    }
    fclose(nom_arch);
    for(aux=0;aux<2;aux++)
    {
        strcpy(vol_presa_angostu[aux], strtok(vol_presa_angostu[aux],","));
        strcpy(vol_presa_chicoas[aux], strtok(vol_presa_chicoas[aux],","));
        strcpy(vol_presa_malpaso[aux], strtok(vol_presa_malpaso[aux],","));
    }
    strcpy(vol_aux_presa_chicoas,vol_presa_chicoas[0]);
}

```

```

.....
/*          SUBROUTINA QUE LEE Y ESCRIBE LOS BENEFICIOS          */
.....
void LEE_ESCRIBE_BENEFICIOS()
{
    nom_arch = fopen("C:BENEF.ENT", "r");
    for(aux=0;aux<12;aux++)
    {
        fscanf(nom_arch,"%f",&tam[aux]);
    }
    fclose(nom_arch);
    for(aux=1;aux<12;aux++)
    {
        tam[aux-1]=tam[aux];
    }
}

.....
/*          SUBROUTINA QUE LEE Y ESCRIBE LOS BENEFICIOS MEDIOS          */
.....
void LEE_ESCRIBE_BENEFICIOS_MEDIOS()
{
    escalador=0.2e+12;
    nom_arch = fopen("C:BENMED.ENT", "r");
    for(aux=0;aux<12;aux++)
    {
        fscanf(nom_arch,"%f",&tam_med[aux]);
    }
    fclose(nom_arch);
    for(aux=1;aux<12;aux++)
    {
        tam_med[aux-1] = tam_med[aux];
    }
    nom_arch = fopen("C:UNBEN.ENT", "r");
    fscanf(nom_arch,"%f",&tam[11]);
    fscanf(nom_arch,"%f",&tam_med[11]);
    fclose(nom_arch);
    nom_arch = fopen("C:BENEF.ENT", "w");
    for(aux=0;aux<12;aux++)
    {
        fprintf(nom_arch,"%f",tam[aux]);
    }
    fclose(nom_arch);
    nom_arch = fopen("C:BENMED.ENT", "w");
    for(aux=0;aux<12;aux++)
    {
        fprintf(nom_arch,"%f",tam_med[aux]);
    }
    fclose(nom_arch);
    nom_arch = fopen("C:UNBEN.ENT", "r");
    fscanf(nom_arch,"%s",tam_grafica);
    fscanf(nom_arch,"%s",tam_med_grafica);
}

```

```

fscanf(nom_arch,"%s",tam_real_grafica);
fclose(nom_arch);
for(aux=0;aux<12;aux++)
{
    tam[aux] = 139.0 - (tam[aux] / escalador);
    tam_med[aux] = 139.0 - (tam_med[aux] / escalador);
}
}

/*****
/*      SUBROUTINA QUE GRAFICA LOS BENEFICIOS DESDE EL MES ACTUAL      */
*****/
void GRAFICA_BENEFICIOS()
{
    setviewport(2,289,637,477,CLIP_ON);
    setcolor(YELLOW);
    rectangle(0,0,635,188);
    line(60,20,58,22);
    line(58,22,62,22);
    line(62,22,60,20);
    line(60,20,60,140);
    line(588,142,590,140);
    line(590,140,588,138);
    line(588,138,588,142);
    line(60,140,590,140);
    settextstyle(SMALL_FONT,HORIZ_DIR,4);
    for(aux=0;aux<12;aux++)
    {
        outtextxy(55+(aux*44),150,mes[aux]);
    }
    for(aux=0;aux<12;aux++)
    {
        setfillstyle(LTSLASH_FILL,LIGHTMAGENTA);
        bar(62+(aux*44),tam[aux],80+(aux*44),139);
    }
    for(aux=0;aux<12;aux++)
    {
        setfillstyle(LTBKSLASH_FILL,GREEN);
        bar(84+(aux*44),tam_med[aux],102+(aux*44),139);
    }
    setfillstyle(SOLID_FILL,BLUE);
}

/*****
/*      SUBROUTINA QUE ROTA LOS MESES A PARTIR DEL MES ACTUAL      */
*****/
void ROTA_MESES()
{
    strcpy(auxmes,mes[0]);
    for(aux=1;aux<12;aux++)
    {
        strcpy(mes[aux-1],mes[aux]);
    }
}

```

```

strcpy(mes[11],auxmes);
}

.....
/*      SUBROUTINA QUE ESCRIBE LOS MESES A PARTIR DEL MES ACTUAL      */
.....
void ESCRIBE_MESES()
{
    nom_arch = fopen("C:MESES.ENT","w");
    for(aux=0;aux<12;aux++)
    {
        strcpy(msg,mes[aux]);
        strcat(msg," ");
        fprintf(nom_arch,"%12s",msg);
    }
    fclose(nom_arch);
}

.....
/*      SUBROUTINA QUE GRAFICA LOS CODIGOS Y VALORES DE LOS BENEFICIOS      */
.....
void CODIGO_DE_BENEFICIOS()
{
    setviewport(436,3,636,103,CLIP_ON);
    setcolor(YELLOW);
    rectangle(0,0,200,100);
    settextstyle(SMALL_FONT,HORIZ_DIR,4);
    setfillstyle(LTSLASH_FILL,LIGHTMAGENTA);
    bar(5,5,20,31);
    strcpy(beneficln,"BENEFICIO = ");
    strcat(beneficln,tam_grafica);
    outtextxy(25,14,beneficln);
    setfillstyle(LTBKSLASH_FILL,GREEN);
    bar(5,36,20,62);
    strcpy(beneficln,"BENEFICIO MEDIO= ");
    strcat(beneficln,tam_med_grafica);
    outtextxy(25,45,beneficln);
    setfillstyle(LTBKSLASH_FILL,WHITE);
    bar(5,67,20,93);
    strcpy(beneficln,"BENEFICIO REAL= ");
    strcat(beneficln,tam_real_grafica);
    outtextxy(25,76,beneficln);
    getch();
    setfillstyle(SOLID_FILL,BLUE);
}

```

---

## 5. CONCLUSIONES

---

Este trabajo se aboca a resolver el problema de ajustar los volúmenes de agua turbinados mensualmente del sistema de presas del río Grijalva de manera tal que se incremente el beneficio por generación de energía eléctrica anual del sistema hidráulico sin riesgo de desbordar las presas o inundar los valles.

En particular tomando las condiciones de optimalidad reportadas en el Instituto de Ingeniería de la UNAM [1] para sistemas independientes conectados, se implementan dos políticas de operación, una en lazo abierto y otra en lazo cerrado.

La implementación de las estrategias se realizó a través del intérprete MATLAB y fueron concebidas para operar en una computadora personal en forma amigable.

La comparación de ambas políticas por medio de simulación para 30 años de operación muestra como era de esperarse las bondades del sistema retroalimentado o en lazo abierto de generación anual de la energía eléctrica.

Adicionalmente se compararon en este trabajo la política propuesta con la estrategia reportada por Domínguez [2] en donde se considera el sistema Grijalva formado por presas hidráulicamente independientes. Como resultado de la comparación se concluye que la política en lazo cerrado con el sistema acoplado es menos conservadora que la desacoplada, sin embargo es un precio que hay que pagar por el beneficio, mayor riesgo ante precipitaciones no esperadas.

Es importante hacer notar que el costo de la implementación de la política en lazo cerrado se reduce a la instalación de una computadora personal en donde mensualmente el operador deberá introducir el estado, esto es volumen de agua, de las tres presas del sistema y recibe a cambio, por medio de un dibujo mímico en el monitor la cantidad de agua que debe turbinarse durante el mes considerado.

Para trabajos futuros se sugiere modificar el criterio a optimizar con el fin de castigar los derrames de agua por vertedero.

## APÉNDICE A

### DESCRIPCIÓN DE LOS PROGRAMAS EJECUTABLES

#### INTRODUCCIÓN

Este apéndice es una recopilación de los programas que realizan la optimización del sistema Grijalva, escritos para el interprete de MATLAB. Esta compuesto de diagramas de flujo, algoritmos y listados de los programas.

La descripción se inicia con el programa principal llamado `BENMAX.M` y se describen las subrutinas en el orden en que son ejecutados.

## A.1 PROGRAMA PRINCIPAL

### BENMAX.M

BENMAX.M es el programa principal del paquete y llama a todas las funciones y procedimientos involucrados en la solución del problema de optimización. Tiene la opción de trabajar con datos almacenados en el archivo `datos.m` o con los volúmenes reales introducidos por teclado.

En el caso de seleccionar la opción de datos almacenados, los volúmenes de las presas se calculan con base en la información de los escurrimientos y en las extracciones calculadas por el programa.

A continuación se describe brevemente la secuencia del programa con la ayuda de un diagrama de flujo y el listado del programa en el intérprete MATLAB.

- 1.- Especificar los formatos de escritura y limpiar el monitor.
- 2.- Cargar las características propias del sistema que se encuentran en el programa de asignación `matin.m`.
- 3.- Cargar la información sobre las condiciones en las cuales se corre el programa BENMAX. Esta información está dada en forma de banderas por medio de las variables `inicia` y `aparece` que se leen del archivo `datosinicia.mat`.

`inicia` es la bandera que controla si es la primera vez que se corre el programa BENMAX.M. Si `inicia` es igual a cero, es la primera

vez que se corre éste, asignando a la matriz de volumen turbinado resultante de la optimación como volúmenes turbinados medios `umed` que se almacena posteriormente en el archivo `datos1jo.mat`. Por otra parte si `inicia` es igual a uno, se tiene un volumen turbinado medio en `datos1jo.mat` y éste se toma como el valor inicial en la ejecución del algoritmo.

La bandera `aparece` se emplea en `ECSS9.M` y controla la información que se muestra en la pantalla al final de la optimación mensual. Si `aparece` almacena el valor de uno se presentará al final de la optimación de cada mes el volumen a turbinar en el mes actual y en el próximo mes, el volumen almacenado, la predicción del volumen para el próximo mes, el volumen derramado inicial y el beneficio para condición inicial. Además se calcula el volumen estimado en las presas para el siguiente mes y el volumen a turbinar en ese mes, y los cuales serán usados por el programa de gráficos. Por el contrario en tiempo real los resultados se almacenan en archivos que son usados como entradas por el programa de graficación.

- 4.- Elegir opción de operar con volúmenes reales o con influjos almacenados (automáticamente). La bandera `manual` es igual a uno en el caso de influjos almacenados y cero en el caso contrario.
- 5.- Si la opción es tiempo real, la bandera `manual=0`, continuar en caso contrario pasar a el paso 9.
- 6.- Asignar a la variable `aparece` el valor de cero. Lo cual indica que no se imprimen los resultados de la optimación en `ECSS9.M`
- 7.- Si la bandera `inicia` es igual a 0, pasar a 10, en caso contrario detecta que es la primera vez que se corre el programa.
- 8.- Se cargan los datos fijos y datos variables de anteriores optimizaciones encontrados en los archivos de datos `datos1jo.mat` y

datosvar.mat. En datosvar.mat se tienen los datos que cambian continuamente de una corrida a otra y que sirvan como variables iniciales en la siguiente corrida.

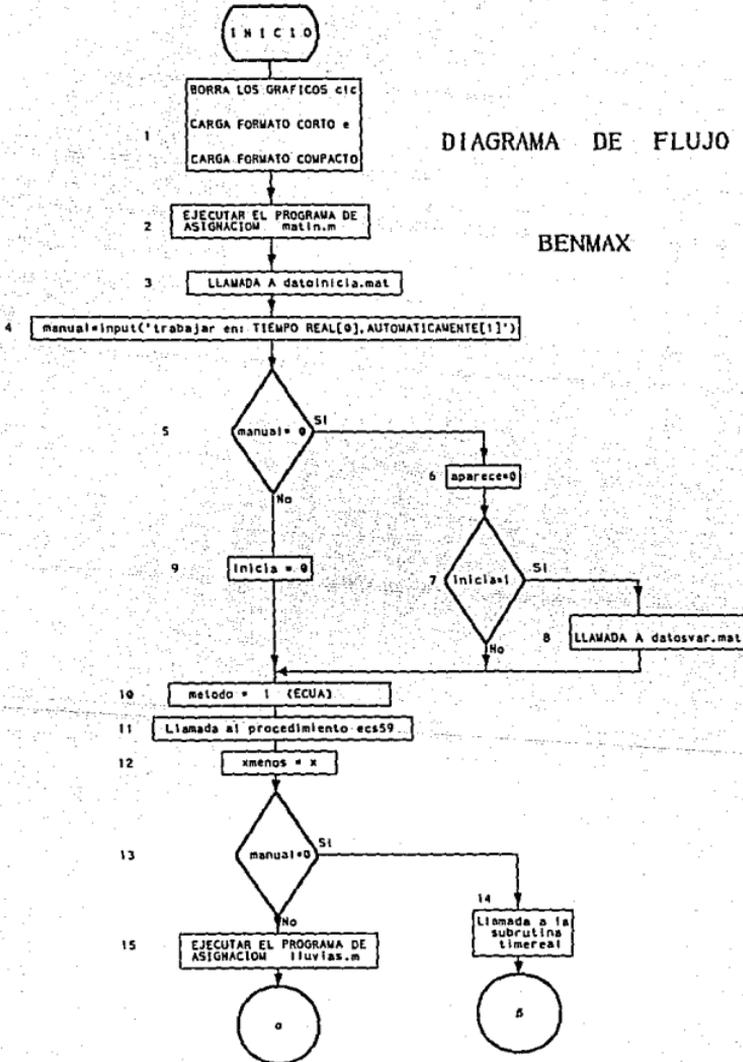
En datofijo.mat contiene la información almacenada originalmente en matin.m y que no se modifica a lo largo del programa. Pasar a el paso 10.

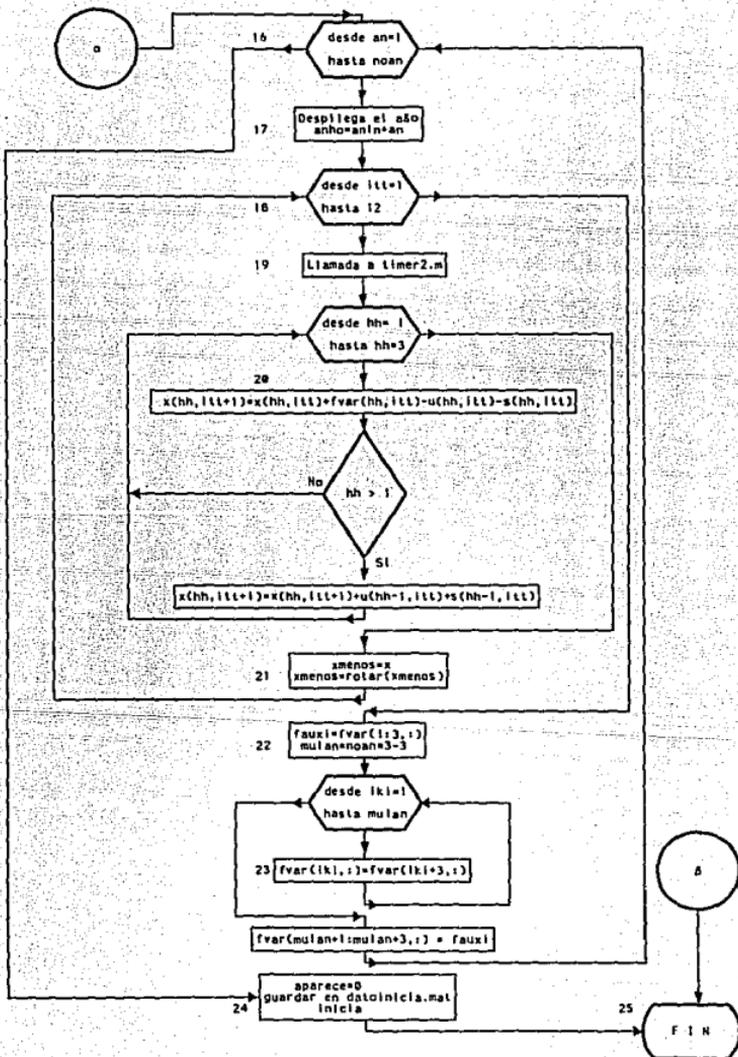
- 9.- Igualar la variable inicia a cero.
- 10.- Indicar que se resolverá la optimización por medio del programa ecua.m. Por medio de la bandera metodo=1.
- 11.- Ejecutar la subrutina exss9, la cual calcula por medio de u la política óptima en lazo cerrado con las presas llenas como condiciones iniciales.
- 12.- Almacenar el volumen calculado para los próximos doce meses, en xmenos, salvando este valor inicial de cada corrida para utilizarlo posteriormente.
- 13.- Si la opción tiempo real esta habilitada, manual=0, continuar en caso contrario ir al paso 15.
- 14.- Ejecutar el procedimiento timereal. En el cual se introducen por teclado el volumen real en las presas, se calcula la solución óptima, y se llama al programa de graficación el cual llama al terminar nuevamente al programa BENMAX.H
- 15.- Cargar el programa de asignación lluvias.m
- 16.- Ejecutar la optimación usando datos almacenados para el número de años especificado en noan; Si noan es mayor que an continuar en caso contrario pasar a 24.

- 17.- Mostrar en pantalla a que año corresponde esta simulación. Que se calcula como la suma del año inicial `anin` más `an`.
- 18.- Ejecutar la optimación en un ciclo de doce meses; si `itt` es igual a doce pasar a 22 ya que se terminó el cálculo para el año `anin`, en caso contrario continuar con el ciclo.
- 19.- Ejecutar el programa `timer2.m`, el cual calcula los valores óptimos. Para el año `anin` y el mes `itt`.
- 20.- Estimar el volumen de cada presa para los próximos doce meses.
- 21.- Almacenar en `xmenos` los volúmenes simulados `x`; esto permite calcular posteriormente las ganancias. Rotar la matriz `xmenos`, para adecuarla al cálculo del mes siguiente. Pasar a 18.
- 22.- Preparar la matriz de influjos para la simulación del siguiente año. Usando la matriz auxiliar `fauxi` y el apuntador `muian` detecta la posición del año que se simula.
- 23.- Rotar las matrices de (3x12) de manera que se almacene en la última posición los influjos del año recién simulado.
- 24.- Cambiar `aparece` con cero, con el fin de apagar la bandera para una corrida posterior. Salvar las banderas de control del sistema, en `datoinicia.m` las variables `inicia` y `aparece`.
- 25.- Fin del algoritmo.

# DIAGRAMA DE FLUJO

BENMAX





```
% NOMBRE DEL ARCHIVO: BENMAX.M
```

```
%
```

```
-----  
% PROGRAMA PRINCIPAL EN EL CUAL SE LLAMAN LAS DIFERENTES OPCIONES  
% PARA OBTENER RESULTADOS, YA SEA TRABAJAR EN TIEMPO REAL, A  
% VARIOS AÑOS AUTOMATICAMENTE, ETC. ENCONTRANDOSE EL BENEFICIO  
% MAXIMO.  
-----
```

```
% P R O G R A M A  
-----
```

```
clc;  
format short e  
format compact  
matin  
load datoInicio  
manual=input('TRABAJAR EN : [0] TIEMPO REAL, [1] AUTOMATICAMENTE : ');  
if manual==0  
aparece=0;  
if inicia==1  
load datosvar  
end  
else  
inicia=0;  
end  
metodo=input(' METODO A UTILIZAR : [0] EPSIGERR, [1] ECUA : ');  
if metodo==0  
alfa=-alfa;  
end  
ecs59  
xmenos=x;  
if manual==0  
timereal;  
else  
lluvias  
for an=1:noan  
disp('AÑO ACTUAL :');  
anho=anin+an;  
disp(anho);  
for itt=1:12  
timer2  
for hh=1:3  
x(hh,itt+1)=x(hh,itt)+fvar(hh,itt)-u(hh,itt)-s(hh,itt);  
if hh > 1  
x(hh,itt+1)=x(hh,itt)+u(hh-1,itt)+s(hh-1,itt);  
end  
end  
xmenos=x;  
xmenos=rotar(xmenos);  
end  
fauxi=fvar(1:3,:);  
mulan=noan*3-3;  
for iki=1:mulan
```

```
fvar(1k1,:)=fvar(1k1+3,:);  
end  
fvar(mulan+1:mulan+3,:)=fauxi;  
end  
aparece=0;  
save datoinicia inicia  
end
```

## A.2 PROGRAMA QUE REALIZA LA OPTIMACIÓN EN TIEMPO REAL

### TIMEREAL.M

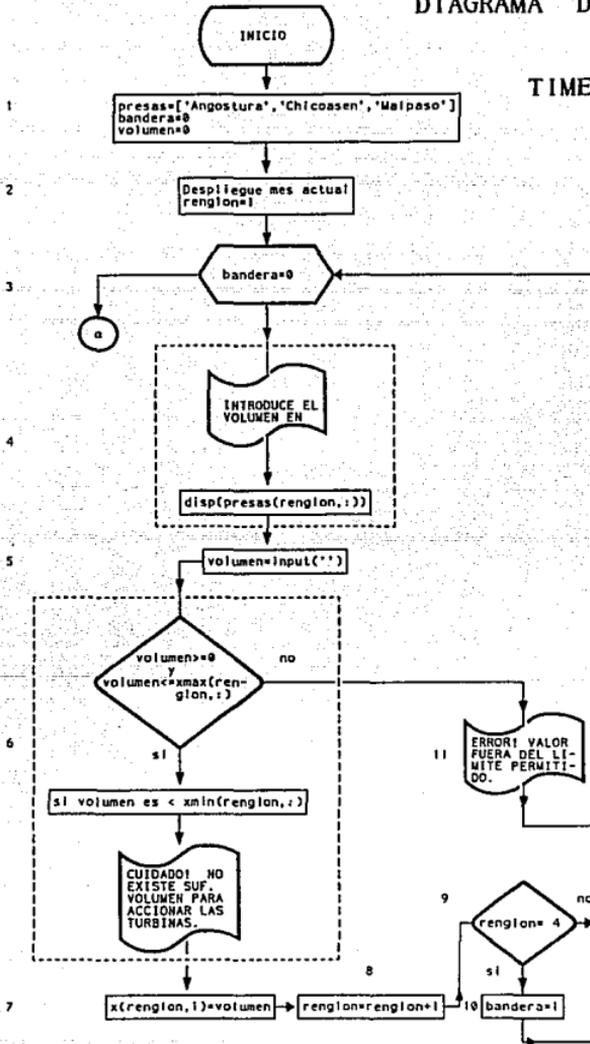
- 1.- Generar el vector que contiene los nombres de las tres presas e inicializar la variable bandera con cero. Además definir la variable volumen con cero.
- 2.- Presentar el mes actual, el cual esta asociado con la primera columna de los datos anuales. Hacer la variable renglon igual a uno, el cual es el contador de las presas en la lectura de volúmenes.
- 3.- Si bandera es igual a cero, continuar, en caso contrario la lectura de datos por teclado se ha terminado y se continúa en el paso 12.
- 4.- Preguntar por el volumen de la presa asociada al índice renglon, presentando el nombre de la presa.
- 5.- Almacenar el volumen proporcionado por teclado en la variable volumen.
- 6.- Detectar si el volumen para la presa renglon es mayor al máximo permitido. En caso afirmativo, pasar al paso 11, en caso contrario continuar presentando un aviso cuando se tiene un volumen menor al requerido para poder turbinar.
- 7.- Asignar en la posición dada por renglon de la primera columna de la matriz x, el valor leído.
- 8.- Incrementar en uno el contador renglon.

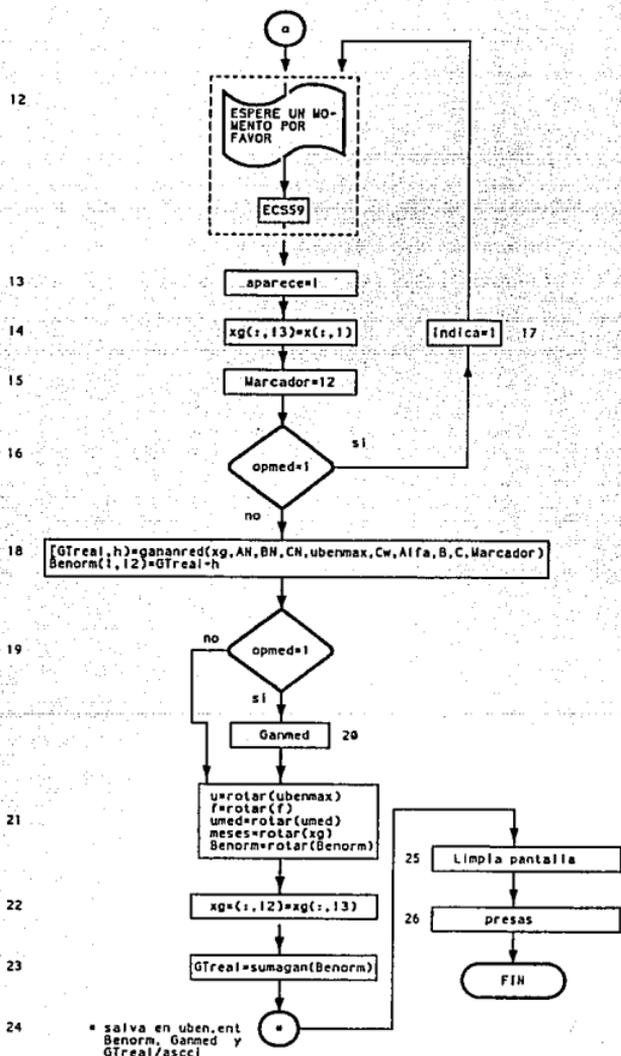
- 9.- Si `renglon` es igual a cuatro, continuar, en caso contrario regresar al paso 3.
- 10.- Asignar a bandera el valor de uno indicando que se ha terminado el proceso de adquisición de datos. Pasar al paso 3.
- 11.- Indicar con un mensaje de error que el volumen introducido está fuera de los límites permitidos por el sistema, regresando al paso 3 en donde se solicita un nuevo dato.
- 12.- Presentar el mensaje "ESPERE UN MOMENTO POR FAVOR" durante el tiempo en que se calcula la optimación a través de la subrutina `EC559.M`.
- 13.- Asignar a la variable `aparece` el valor de uno.
- 14.- Guardar el volumen real actual `x`, en la columna trece de la matriz de los últimos volúmenes reales `xg`; `xg` es una variable auxiliar para estimar los beneficios pasados.
- 15.- Asignar el número de meses considerado en el beneficio, `Marcador=12`.
- 16.- Si `indica`, definida en el archivo `matin.m`, vale cero continuar, en caso contrario ir a 18; esto se realiza para evitar el cálculo del beneficio en el primer mes que se aplica la política.
- 17.- Hacer la variable `indica` igual a uno e ir al paso 12.
- 18.- Calcular la ganancia total real para los últimos doce meses y dejar en `Benora(1,12)` su valor sin considerar el beneficio al final del horizonte, `n`.
- 19.- Preguntar si la variable `opmed` es igual a uno, la cual controla

- el cálculo de la ganancia media, en caso contrario pasar a 21.
- 20.- Calcular el beneficio medio  $G_{amed}$  asociado a la política en lazo abierto.
- 21.- Realizar las rotaciones de:
- El volumen de descarga por turbinas, correspondiente al beneficio máximo,  $u_{benmax}$ .
  - Los influjos mensuales,  $f$ .
  - El volumen medio de descarga por turbinas,  $u_{med}$ .
  - Los meses del año transpuestos, se utiliza transpuesto por la forma en que se almacenaron los meses en  $meses$ .
  - El estado de las presas para los últimos doce meses,  $xg$ .
  - El beneficio real mensual para los últimos doce meses,  $Benorm$ .
- 22.- Recorrer la columna treceava de  $xg$ , a la posición doceava, ya que la función que rota, deja en el doceavo valor, el primer valor de el vector y lo que se desea es el último volumen real en la posición doceava.
- 23.- Llamar a la función  $sumagan$ , la cual suma los últimos doce beneficios mensuales que se encuentran en  $Benorm$ ; El resultado es almacenado en la variable  $G_{real}$  y corresponde al beneficio anual.
- 24.- Salvar la ganancia total anual a futuro  $Benorm$ , la ganancia media  $G_{amed}$  y la ganancia real anual para los últimos doce meses  $G_{real}$  en el archivo llamado  $uben.ent$  en formato  $ascii$  para que pueda ser leído por el programa gráfico.
- 25.- Limpiar el monitor.
- 26.- Llamar al programa  $presas$ , que es la parte que muestra los resultados en forma gráfica.

# DIAGRAMA DE FLUJO

## TIMEREAL





```

% NOMBRE DEL ARCHIVO: TIMEREAL.M
%
% -----
% PROGRAMA LLAMADO POR BENMAX
%
% -----
%                               P R O G R A M A
% -----

presas=['ANGOSTURA';'CHICOASEN';'MALPASO'];
bandera=0;
volumen=0;
disp('MES ACTUAL ');
disp(meses(1,:));
renglon=1;
while (bandera==0)
  disp('INTRODUCE EL VOLUMEN EN ')
  disp(presas(renglon,:));
  volumen=input('');
  if ((volumen >= 0) & (volumen <= xmax(renglon,:)))
    if (volumen < xmin(renglon,:))
      fprintf(' CUIDADO ! NO EXISTE SUFICIENTE VOLUMEN PARA ');
      fprintf('          ACCIONAR LAS TURBINAS          ');
    end
    x(renglon,1)=volumen;
    renglon=renglon+1;
    if (renglon == 4)
      bandera=1;
    end
  else
    fprintf(' ERROR ! VALOR FUERA DEL LIMITE PERMITIDO EN LA PRESA ');
    fprintf('          < VEFIQUE EL VALOR >          ');
  end
end
fprintf(' ESPERE UN MOMENTO POR FAVOR ..... ');
aparece=1;
ecs59;
xg(:,13)=x(:,1);
Marcador=12;
if indica==0
  indica=1;
else
  [GTreal,h] = ganared(xg,AN,BN,CN,ubenmax,Cw,Alfa,B,C,Marcador);
  Benorm(1,12)=GTreal-h;
  if opmed==1
    media;
  end
end
u = rotar(ubenmax);
f = rotar(f);
umed = rotar(umed);
meses = rotar(meses');

```

```
meses=meses';
xg = rotar(xg);
Benorm = rotar(Benorm);
xg(:,12)=xg(:,13);
GTreal=sumagan(Benorm),
fprintf('');
save unben.ent GTreal Ganmed Bemax /ascii
save datosvar f u meses umed xg Benorm Bemed indica x s
aparece=0;
save datoincia inicia
!cls
!presas
!cls
```

**ESTA TESIS NO DEBE  
SALIR DE LA BIBLIOTECA**

### A.3 PROGRAMA QUE REALIZA LA OPTIMACIÓN CON INFLUJOS ALMACENADOS

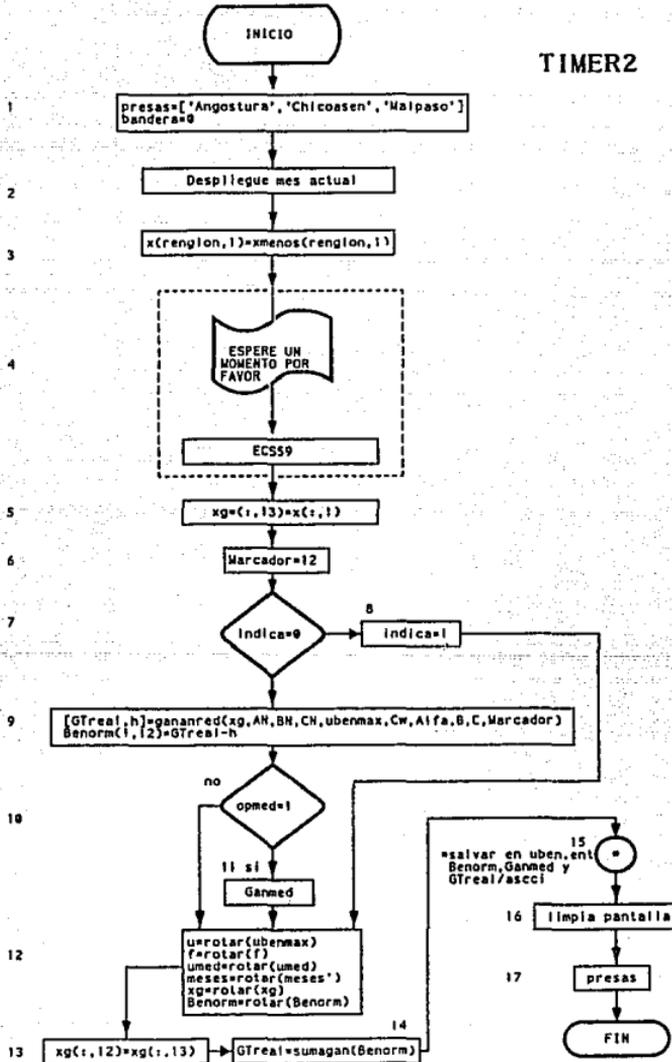
#### TIMER2.M

- 1.- Generar el vector que contiene los nombres de las tres presas e inicializar la variable bandera con cero.
- 2.- Indicar el mes actual, el cual esta asociado con la primera columna de todos los datos anuales.
- 3.- Asignar a la primera columna de la variable  $x$ , el valor de la primera columna de la variable  $x_{\text{menos}}$ . Esto se realiza para iniciar la optimación con un volumen inicial que satisfaga la ecuación de continuidad.
- 4.- Presentar el mensaje " ESPERE UN MOMENTO POR FAVOR " durante el tiempo en que se calcula la optimación a través de la subrutina EC59.
- 5.- Guardar el volumen real actual  $x$  en la columna trece de la matriz de los últimos volúmenes reales  $x_g$ ;  $x_g$  es una variable auxiliar para estimar los beneficios pasados.
- 6.- Asignar el número de meses con el que se calcula el beneficio, esto es Marcados=12.
- 7.- Si indica, definida en el archivo matn.m, vale cero continuar, en caso contrario ir al paso 9; ésto se realiza para evitar el cálculo del beneficio del primer mes que se aplica la política.
- 8.- Hacer la variable indica igual a uno e ir al paso 12.
- 9.- Calcular la ganancia total real para los últimos doce meses y almacenar en Benorm(1,12) su valor sin considerar el beneficio al

- 10.- Preguntar si la variable  $opened$  es igual a uno, el cual controla el cálculo de la ganancia media, en caso contrario pasar a 12.
- 11.- Calcular el beneficio medio  $canmed$  asociado con la política en lazo abierto.
- 12.- Realizar las rotaciones de las siguientes variables.
  - el volumen de descarga por turbinas, correspondiente al beneficio máximo,  $ubenmax$ .
  - los influjos mensuales,  $r$ .
  - el volumen medio de descarga por turbinas,  $umed$ .
  - los meses del año transpuestos, es transpuesto este vector por la forma en que se almacenaron los meses,  $meses$ .
  - el estado de las presas para los últimos doce meses,  $xg$ .
  - el beneficio real mensual para los últimos doce meses,  $Benorm$ .
- 13.- Recorrer la columna treceava de  $xg$  o volumen real a la posición doceava. Esto se requiere porque la función ROTAR, deja la doceava columna en la primera.
- 14.- Llamar la función  $sumagen$ , la cual suma los últimos doce beneficios mensuales almacenados en  $Benorm$  y deja al beneficio anual calculado en la variable  $gtreal$ .
- 15.- Salva la ganancia total anual a futuro,  $Benorm$ , la ganancia media,  $canmed$  y la ganancia real anual para los últimos doce meses,  $gtreal$ , en el archivo llamado  $uben.ent$  usando formato  $ascii$  para que pueda ser leído por el programa de graficación.
- 16.- Limpiar el monitor.
- 17.- Llamar al programa  $presas$  encargado de presentar los resultados en forma gráfica.

# DIAGRAMA DE FLUJO

TIMER2



```

% NOMBRE DEL ARCHIVO: TIMER2.M
%
% -----
% PROGRAMA LLAMDO POR BENMAX
% -----
%
%                               P R O G R A M A
% -----

```

```

presas=['ANGOSTURA';'CHICOASEN';'MALPASO'];
bandera=0;
disp(' MES ACTUAL ');
disp(meses(1,:));
renglon=1;
while (bandera==0)
    x(renglon,1)=xmenos(renglon,1);
    renglon=renglon+1;
    if (renglon == 4)
        bandera=1;
    end
end
fprintf(' ESPERE UN MOMENTO POR FAVOR ..... ');
ecs59;
x(:,13)=x(:,1);
Marcador=12;
if indica==0
    indica=1;
else
    [GTreal,h] = ganared(xg,AN,BN,CN,ubenmax,Cw,Alfa,B,C,Marcador);
    Benorm(1,12)=GTreal-h;
    if opmed==1
        media;
    end
end
u = rotar(ubenmax);
f = rotar(f);
umed = rotar(umed);
meses = rotar(meses');
meses=meses';
xg=rotar(xg);
Benorm = rotar(Benorm);
xg(:,12)=xg(:,13);
GTreal=sumagun(Benorm),
fprintf('');
save unben.ent GTreal Ganmed Bemax /ascii
!cls
!presas
!cls

```

#### A.4 PROGRAMA QUE RESUELVE LAS CONDICIONES DE OPTIMALIDAD

E C S 5 9 . M

- 1.- Inicializar variables auxiliares, con cero,  $B_e$  (contiene los beneficios a futuro),  $B_{e_{max}}$  (contiene el beneficio máximo encontrado),  $p_{control}$  (contiene el número de iteraciones). Asignar a la bandera que controla las iteraciones,  $bandera$ , el valor de uno.
- 2.- Mientras  $bandera$  sea igual a uno, continuar, en caso contrario pasar a el paso 21. Es decir si no se ha llegado al número de iteraciones máximo de la optimización continuar.
- 3.- Incrementar la variable  $p_{control}$ , indicando una iteración más.
- 4.- Ejecutar la función que calcula la ecuación de continuidad con restricciones para los siguientes 12 meses del año,  $vol_{x_{new}}$ , con lo cual se obtienen los volúmenes de las presas durante el próximo año considerando las restricciones de los volúmenes turbinables  $u$ , volúmenes a almacenar  $x$  y los volúmenes vertidos  $v$ .
- 5.- Asignar a la variable  $marcador$  el valor de uno, indicando que se debe calcular la ganancia para el mes actual. Esto se realiza para obtener el beneficio real por mes.
- 6.- Ejecutar la función  $ganamed$ , calculando la ganancia del mes presente.
- 7.- Almacenar la ganancia calculada en el vector  $B_e$ , en la localidad  $p_{control}$ .
- 8.- Si el número de veces que se ajustan las ecuaciones de optimalidad es mayor que cinco, continuar. En caso contrario no se comparan los

beneficios y se debe de ir a 11.

9.- Si el último beneficio calculado es mayor al valor en las iteraciones anteriores, se almacena el beneficio y se continúa. En caso contrario ir al paso 11.

10.- Almacenar el volumen turbinado  $u_{benmax}$ , el volumen de las presas  $x_{benmax}$ , y el volumen que se derrama  $s_{benmax}$ , que generan el beneficio máximo.

11.- Calcular  $\mu$  de acuerdo a la ecuación 2.44, utilizando las variables  $c_w, c_y$  u. Este cálculo es necesario porque se ha cambiado el valor del volumen turbinado  $u$ .

12.- Calcular  $\phi$  de acuerdo a la ecuación 2.45

13.- Calcular  $\psi$  de acuerdo a la ecuación 2.46.

14.- Si el número de iteraciones de la optimación excede a  $p_{max}$ , continuar, en caso contrario ir al paso 16.

15.- Hacer la variable bandera igual a cero. Indicando que se ha llegado al número máximo de iteraciones y se debe regresar a 2.

16.- Llamar a la función  $ecua.m$ , con objeto de satisfacer las ecuaciones de optimación 3.7, 3.8 y 3.9. Encontrando  $\sigma, \xi$  y Error.

17.- Almacenar el volumen turbinado  $u$  en la variable auxiliar  $u_{anterior}$ . Con objeto de poder compararla con el valor de la iteración anterior.

18.- Calcular la nueva aproximación del volumen turbinado  $u$ , usando la expresión 3.10. Esto es la suma del volumen turbinado actual mas la variable  $\sigma$  por el Error calculado en la función  $ecua.m$ .

- 19.- Comparar con la norma euclidiana, la nueva aproximación del volumen turbinado y el volumen turbinado anterior,  $u_{anterior}$ . Si la norma entre la diferencia de volúmenes es menor a 0.001 continuar, en caso contrario ir al paso 2, iniciando otra iteración más y ajustando la nueva  $u$ .
- 20.- Hacer la variable `bandera` igual a cero.
- 21.- Si la variable `aparece` es igual a uno, indicando que se está en la opción de simulación, se presentan en el monitor los resultados y la secuencia continúa. En caso contrario pasar a 23.
- 22.- Almacenar en variables auxiliares los resultados encontrados,  $u = ubenmax(:,1)$  que es el volumen a turbinar en el mes presente,  $x = xbenmax(:,1:2)$  que es el volumen que se tienen en las presas en el mes presente y para el siguiente mes,  $s = sbenmax(:,1)$  que es el volumen que se debe de derramar.
- 23.- Los datos  $volx = xbenmax(:,2)$  y  $volu = ubenmax(:,1)$  se almacenan como caracteres `ascii`. Para leerlo posteriormente en el programa de graficación. Además se almacenan en el archivo `datotem.mat`, el beneficio máximo obtenido `benmax`, así como sus correspondientes valores almacenados en `xbenmax`, `ubenmax` y `sbenmax`.
- 24.- Si la variable `inicia` es cero, indicando que es la primera vez que se ejecuta el programa, es necesario almacenar las variables iniciales en el archivo `datos1jo.mat`. En caso contrario ir a 29.
- 25.- Almacenar el volumen turbinado máximo en `umed`. Igualar a uno las variables `inicia` y `aparece`. Inicializar con ceros las matrices `cont-sig` y `cont_eps`.
- 26.- Si se tiene la opción en tiempo real, `manual=0`, se deben salvar los resultados dejándolos listos para la siguiente corrida. En caso

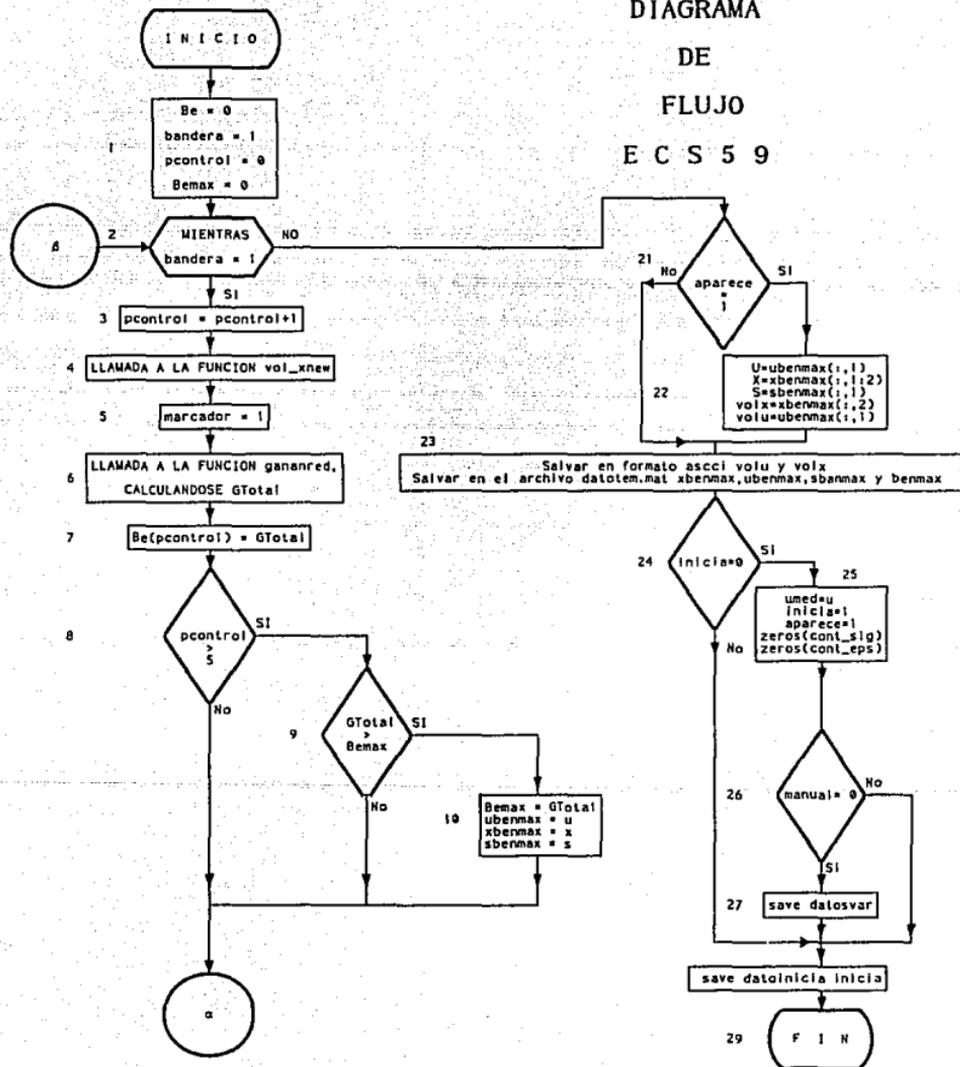
contrario ir a 28.

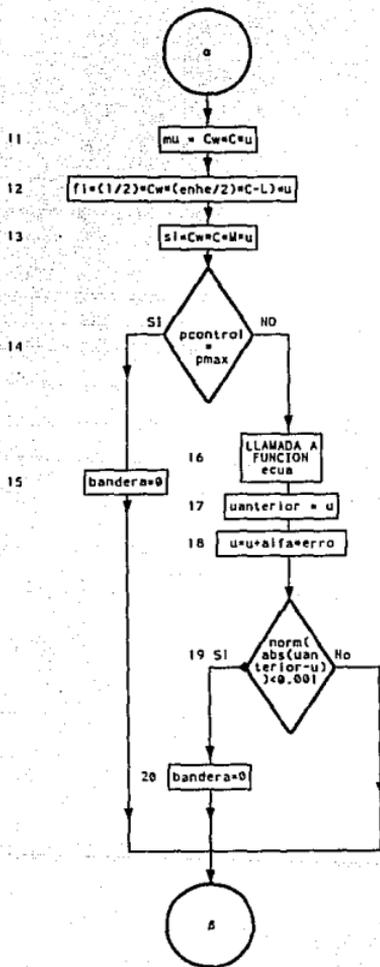
27.- Salvar los datos variables en el archivo `datosvar.mat`

28.- Salvar los datos inicia en el archivo `datosinicia.mat`

29.- Salir de la subrutina y retornar a el programa donde fue llamado.

DIAGRAMA  
DE  
FLUJO  
E C S 5 9





```

% NOMBRE DEL ARCHIVO: ECS59.M
%
%-----
%          PROGRAMA PARA LA SOL. DE LAS COND. 5 Y 9 EN FORMA DE CASCADA
%          PARA EL CALCULO DE U OPTIMA
%-----
%
% SUBROUTINAS UTILIZADAS :
%
%volx_new Funcion que verifica que el agua turbinada este dentro de sus
%          limites, tambien realiza el calculo del agua que sale por
%          vertederos y finalmente calcula el volumen de las presas.
%gananred Funcion que calcula la ganancia que se va obteniendo en los
%          12 meses y para las tres presas en conjunto.
%epsigerr Funcion que calcula los valores de las variables epsilon,
%          sigma y el error generado al utilizar estas variables en las
%          ecuaciones 5 y 9 segun el reporte. Siguiendo un metodo de
%          solucion para estas ecuaciones.
%ecua      Subrutina que calcula los valores de las variables epsilon,
%          sigma y el error generado al utilizar estas variables en las
%          ecuaciones 5 y 9 segun el reporte. Siguiendo un segundo metodo
%          de solucion para estas ecuaciones.
%
%-----
%          P R O G R A M A
%-----

```

```

Be=0;
Bemax=0;
bandera = 1;
pcontrol = 0;
while (bandera==1)
    %Contador de iteraciones
    %While que sirve de control al
    % programa para calcular
    %Control de iteraciones maximo.
    pcontrol = pcontrol + 1;
    [u,s,cont_sig,x,cont_eps,q,b,d] = volx_new(u,umin,umax,s,...
        cont_sig,x,f,xmun,xmax,...
        cont_eps,M,AV,B,C,BV); %Verifica u,
        %calcula s y x Marcador = 1;
    [GTtotal,h] = gananred(x,AN,BN,CN,u,Cw,...
        Alfa,B,C,Marcador); %Funcion
        %que calcula Be(pcontrol) = GTtotal;
    if metodo==0
        if pcontrol>4
            %Para EPSIGERR
            if GTtotal > Bemax
                Bemax=GTtotal;
                ubenmax=u;
                xbenmax=x;
                sbenmax=s;
            end
        end
    else
        if pcontrol>5
            %Para ECUA
            if GTtotal > Bemax
                Bemax=GTtotal;

```

```

        ubenmax=u;
        xbenmax=x;
        sbenmax=s;
    end
end
end
mu=Cw*C*u; % 3x13
fi=(1/2)*Cw*((enhe/2)*C-L)*u; % 3x13
si=Cw*C*M*u; % 3x13
if pcontrol==pmax %Si se llega al numero
                    %de iteraciones
    bandera = 0;
else
    if metodo==1
        ecua;
        uanterior=u;
        u = u + alfa * erro;
    else
        [lamda, sigm, erro, epsilon] = epsigerr(d, x, si, u, M, fi, mu, epsilon, ...
        cont_eps, lamda, Cw, b, C, L, enhe, ...
        sigm, cont_sig);
        uanterior=u;
        u(:,1:12) = u(:,1:12) + alfa * erro;
    end
    if (norm(abs(uanterior-u))<0.001)
        bandera=0;
    end
end
end
end
fprintf('');
if aparece==1
    U=ubenmax(:,1),
    X=xbenmax(:,1:2),
    S=sbenmax(:,1),
    volx=xbenmax(:,2);
    volu=ubenmax(:,1);
    Be=Be;
    Bemax=Bemax,
end
save volturb.ent volx volu /ascii
save datotem xbenmax ubenmax sbenmax Bemax
if (inicia==0)
    umed=ubenmax;
    inicia=1;
    aparece=1;
    cont_sig=zeros(cont_sig);
    cont_eps=zeros(cont_eps);
    if manual==0
        save datosvar f u meses umed xg Benorm Bemad indica x s
    end
    save datoinicia inicia
end
end

```

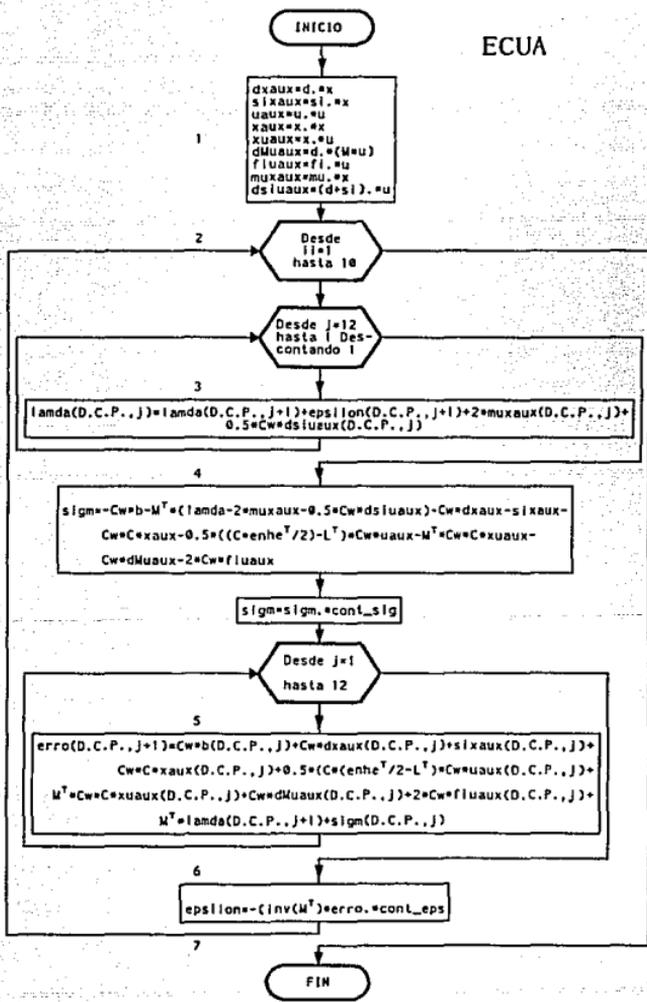
## A.5 PROGRAMA QUE CALCULA LAMDA, SIGMA Y EL ERROR DE ÉPSILON

### E C U A . M

Este programa resuelve iterativamente para los 12 meses del año la ecuación 3.6, de atrás hacia adelante y la ecuación 3.7 y 3.8 de adelante hacia atrás.

- 1.- Asignar las variables auxiliares  $dx_{aux}$ ,  $six_{aux}$ ,  $uaux$ ,  $xaux$ ,  $xuaux$ ,  $dHuaux$ ,  $fiuaux$ ,  $muxaux$  y  $deluaux$  que se van a utilizar en ecuaciones posteriores de acuerdo a la sintaxis de MATLAB.
- 2.- Si el apuntador  $i$  del número de veces que se ajustan las variables  $\lambda$ ,  $\sigma$ ,  $\xi$ , excede el valor de 10 terminar el programa, en caso contrario continuar.
- 3.- Resolver la ecuación en  $\lambda$  (3.6) de atrás hacia adelante, iniciando el cálculo con  $\lambda^{12}$ , por medio del comando for.
- 4.- Resolver la ecuación en  $\sigma$  (3.7) de adelante hacia atrás por medio de un for; en caso de que el volumen almacenado  $x$  no sobrepase los límites se le asigna el valor a la correspondiente  $\sigma$ .
- 5.- Calcular el error en  $\xi$  de las condiciones de optimalidad a partir de la ecuación recursiva (3.8) de adelante hacia atrás.
- 6.- Determinar  $\xi$  a partir de los valores de error (ec. (3.9)); en caso que el volumen turbinado no exceda los límites, se le asigna a  $\xi$  el valor cero.
- 7.- Regresar al paso 2.

ECUA



```

% NOMBRE DEL ARCHIVO: ecua.m
%
% -----
%   Calcula lamda de atras hacia adelante a partir de (5),
%   Calcula sigma a partir de (9) y de (5)
%   Calcula el error M'epsilon de (9)
% -----
%
% -----
%                               P R O G R A M A
% -----
dxaux=d.*x;                               %dxaux=3x13
sixaux=s1.*x;                             %sixaux=3x13
uaux=u.*u;                                %uaux=3x13
xaux=x.*x;                                %xaux=3x13
xuaux=x.*u;                               %xuaux=3x13
dMuaux=d.*(M*u);                         %dMuaux=3x13
fiuaux=f1.*u;                             %fiuaux=3x13
muxaux=mu.*x;                             %muxaux=3x13
dsiuaux=(d+s1).*u;                       %dsiuaux=3x13
muxaux=mu.*x;                             %muxaux=3x13
                                           %Iteración de (9) y (5)
for li=1:10                               %Calcular lamda de N hacia 1
for j=12:-1:1
lamda(:,j)= lamda(:,j+1) + epsilon(:,j+1) + 2 * muxaux(:,j)
           + 0.5 * Cw * dsiuaux(:,j);
end
                                           %Calcular sigm con (lamda+epsilon)
sigm=-Cw*b -M'*(lamda-2*muxaux-0.5*Cw*dsiuaux) -Cw*dxaux -sixaux ..
      -Cw*C*xaux -0.5*((C*enhe'/2)-L')*Cw*uaux -M'*Cw*C*xuaux..
      -Cw*dMuaux -2*Cw*fiuaux;
sigm = sigm.*cont_sig;
                                           %Calcular el error de (9)
for j=1:12
erro(:,j+1)=Cw*b(:,j) + Cw*dxaux(:,j) +sixaux(:,j) +Cw*C*xaux(:,j)..
           +0.5*(C*(enhe'/2)-L')*Cw*uaux(:,j) +M'*Cw*C*xuaux(:,j)..
           +Cw*dMuaux(:,j) +2*Cw*fiuaux(:,j) +M'*lamda(:,j+1) +sigm(:,j);
end
epsilon=-(inv(M')*erro).* cont_eps;
end

```

## A.6 FUNCIÓN QUE CALCULA EL VOLUMEN DE LAS PRESAS.

### V O L X \_ N E W . M

La función `volx_new`, requiere tanto de series de parámetros y variables de entrada como de un conjunto de variables de salida. El conjunto de parámetros de entrada está definido por: `umin`, `umax`, `f`, `xmin`, `xmax`, `H`, `AV`, `B`, `C`, `BV`. Además los valores de las variables `x`, `s`, `u`, `cont_sig`, `cont_eps`, `q`, `b`, `d` se modifican durante la ejecución de la función de tal manera que al final de la función, tanto el volumen almacenado en las presas como el volumen turbinado no violan las restricciones físicas.

- 1.- Verificar si se ha ajustado los doce vectores de volumen por medio del apuntador `i`; En caso afirmativo ir al paso 38; en caso contrario continuar.
- 2.- Verificar si se han ajustado los tres volúmenes para el mes `h`, en caso afirmativo, regresar al paso 2 y en caso contrario continuar.
- 3.- Si el volumen turbinado `u`, es menor que cero continuar, en caso contrario ir al paso 5.
- 4.- Asignar al volumen turbinado el valor mínimo tolerado en la presa y al volumen derramado el valor de cero. Además se enciende la bandera que indica que el volumen turbinado está en el límite `cont_sig=1`.
- 5.- Si el volumen actual de la presa más el volumen proveniente de las precipitaciones es menor que la suma de los volúmenes turbinados más los volúmenes extraídos, continuar, en caso contrario ir a 7.
- 6.- Asignar al volumen turbinado el valor máximo tolerado en la presa y además, asignar a el volumen vertido el volumen máximo turbinado en la presa por dos.

- 7.- Resolver la ecuación de continuidad para la presa h.
- 8.- Si  $h$  es mayor que 1 continúa, en caso contrario, ir al paso 10.
- 9.- Resolver la ecuación de continuidad correspondiente a las presas 2 y 3.
- 10.- Si el volumen de la presa h, sobrepasa el límite máximo, continúa, en caso contrario, ir al paso 17.
- 11.- Encender la bandera `cont_esp`, la cual indica que el volumen se encuentra en un límite, calcular el agua turbinada como la suma del agua que sale por vertederos con la diferencia entre el agua de la presa y el volumen máximo permitido de la misma. El agua que sale por vertederos se hace nula y el volumen de la presa se iguala al volumen máximo permitido.
- 12.- Si el volumen turbinado es mayor o igual al volumen turbinado máximo permitido continuar, en caso contrario, ir al paso 14.
- 13.- Encender la bandera `cont_mig`, la cual indica que el agua turbinada está en un límite. Calcular el volumen que sale por vertederos como la diferencia entre el volumen turbinado y el volumen turbinado máximo permitido. Asignar a el volumen turbinado el volumen turbinado máximo. Ir al paso 2.
- 14.- Si el volumen turbinado es menor que el volumen turbinado mínimo permitido continuar, en caso contrario pasar a 16.
- 15.- Prender la bandera `cont_mig`, la cual indica que el volumen turbinado está en un límite. Ir al paso 2.
- 16.- Apagar la bandera `cont_mig`, la cual indica que el volumen turbinado está en un límite. Ir al paso 2.

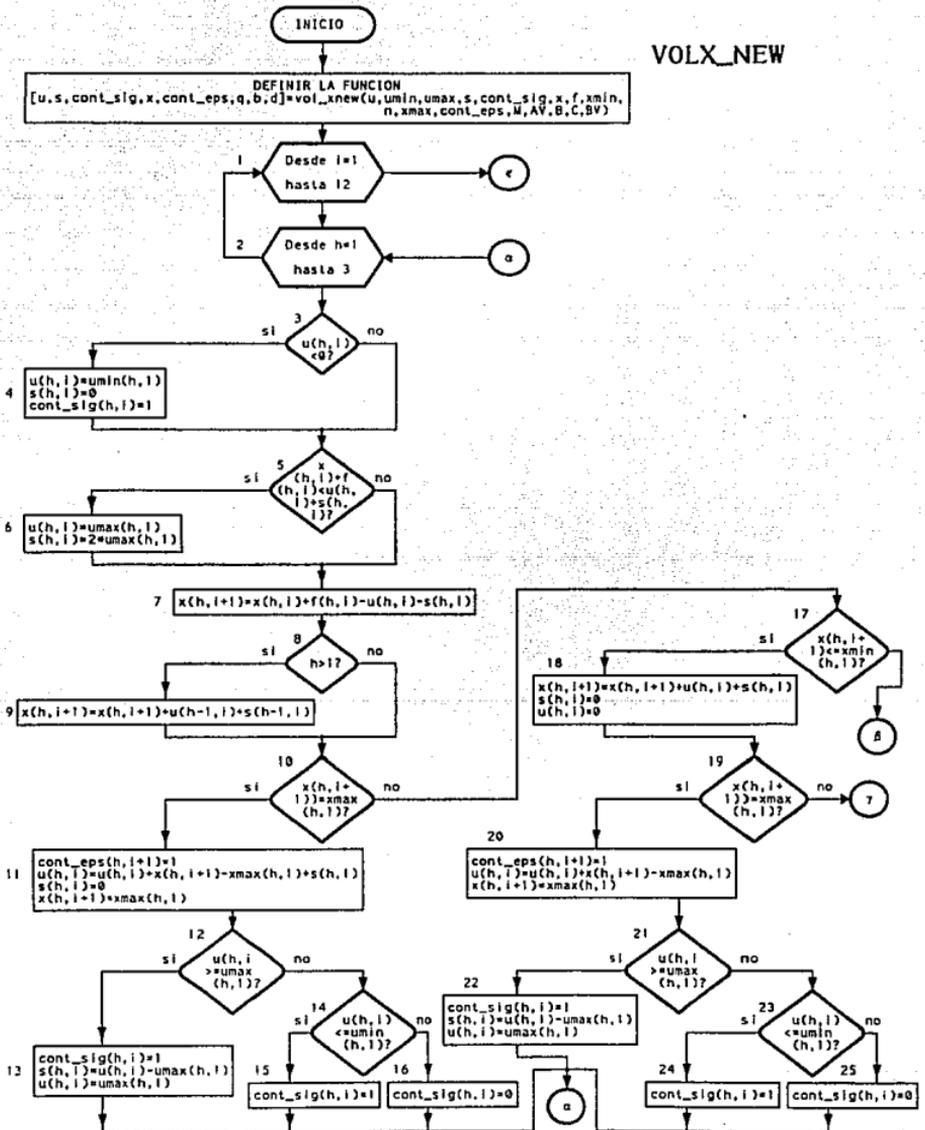
- 17.- Si el volumen de la presa  $h$ , es menor o igual al volumen mínimo permitido continuar, en caso contrario, ir al paso 29.
- 18.- Calcular el volumen de la presa  $h$ , como la suma de el volumen que tiene, mas el volumen turbinado, mas el volumen que derrama por vertederos. Asignar a el volumen turbinado  $u$  y a el volumen derramado por vertederos  $s$ , el valor de cero.
- 19.- Si el volumen de la presa  $h$ , es mayor que el máximo permitido continuar, en caso contrario ir al paso 26.
- 20.- Encender la bandera  $cont\_eps$ , la cual indica que el volumen está en un límite. Calcular el volumen turbinado como la suma del volumen turbinado mas el volumen almacenado en la presa, menos el volumen máximo de la presa. Además el volumen de la presa se hace igual al volumen máximo.
- 21.- Si el volumen turbinado es mayor o igual al volumen turbinado máximo permitido continuar, en caso contrario, ir al paso 23.
- 22.- Encender la bandera  $cont\_sig$ , la cual indica que el volumen turbinado está en un límite. Calcular el volumen derramado por vertederos como la diferencia entre el volumen turbinado y el máximo permitido. Además el volumen turbinado se hace igual al máximo. Ir al paso 2.
- 23.- Si el volumen turbinado es menor que el mínimo permitido continuar, en caso contrario, ir al paso 25.
- 24.- Encender la bandera  $cont\_sig$ , la cual indica que el volumen turbinado está en un límite. Ir al paso 2.
- 25.- Apagar la bandera  $cont\_sig$ , la cual indica que el volumen turbinado está en un límite. Ir al paso 2.

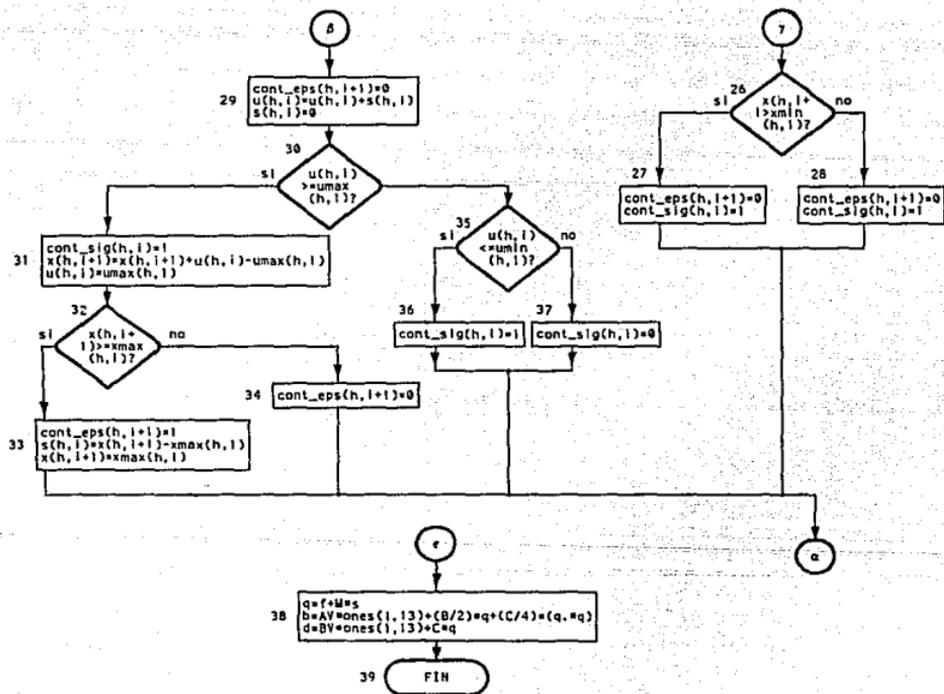
- 26.- Si el volumen de la presa  $h$ , es mayor que el mínimo permitido continuar, en caso contrario, ir al paso 28.
- 27.- Apagar la bandera  $cont\_eps$ , la cual indica que el volumen de la presa está en un límite y prender la bandera  $cont\_sig$ , la cual indica que el volumen turbinado está en un límite. Ir al paso 2.
- 28.- Prender la bandera  $cont\_eps$ , la cual indica que el volumen de la presa está en un límite y prender la bandera  $cont\_sig$ , la cual indica que el volumen turbinado está en un límite. Ir al paso 2.
- 29.- Apagar la bandera  $cont\_eps$ , la cual indica que el volumen de la presa está en un límite, calcular el volumen turbinado como el volumen turbinado más el volumen derramado por vertederos. Asignar al volumen derramado por vertederos el valor de cero.
- 30.- Si el volumen turbinado es mayor o igual al máximo permitido continuar, en caso contrario, ir al paso 34.
- 31.- Prender la bandera  $cont\_sig$ , la cual indica que el volumen turbinado está en un límite. Calcular el volumen de la presa  $h$ , como la suma de el volumen de la presa  $h$  más la diferencia entre el volumen turbinado y el máximo permitido.
- 32.- Si el volumen de la presa es mayor o igual al máximo permitido continuar, en caso contrario, ir al paso 34.
- 33.- Prender la bandera  $cont\_eps$ , la cual indica que el volumen de la presa está en un límite. Calcular el volumen derramado por vertederos como la diferencia entre el volumen de la presa y el máximo permitido. Además el volumen de la presa se hace igual al máximo permitido. Ir al paso 2.
- 34.- Apagar la bandera  $cont\_eps$ , la cual indica que el volumen de la

- presa está en un límite. Ir al paso 2.
- 35.- Si el volumen turbinado es menor o igual al mínimo continuar, en caso contrario, ir al paso 37.
- 36.- Prender la bandera `cont_sig`, la cual indica que el volumen turbinado está en un límite. Ir al paso 2.
- 37.- Apagar la bandera `cont_sig`, la cual indica que el volumen turbinado está en un límite. Ir al paso 2.
- 38.- Calcular las variables `q`, `b` y `d`, debido a que algunas variables internas cambiaron de valor.
- 39.- Retornar al programa donde fué llamado.

# DIAGRAMA DE FLUJO

VOLX\_NEW





```
% NOMBRE DEL ARCHIVO: VOLX_NEW.M
```

```
%
```

```
%
```

```
-----  
% SUBROUTINA QUE CALCULA EL VOLUMEN DE LA PRESA  
-----
```

```
%
```

```
%
```

```
%
```

```
-----  
% P R O G R A M A  
-----
```

```
%
```

```
function [ u , s , cont_sig , x , cont_eps , q , b , d ] =  
    volx_new ( u , umin , umax , s , cont_sig , ...  
    x , f , xmin , xmax , cont_eps , ...  
    M , AV , B , C , BV )
```

```
for i=1:12
```

```
    for h=1:3
```

```
        if u(h,1) < 0
```

```
            u(h,1) = umin(h,1);
```

```
            s(h,1) = 0;
```

```
            cont_sig(h,1) = 1;
```

```
        end
```

```
        if (x(h,1) + f(h,1)) < (u(h,1)+s(h,1))
```

```
            u(h,1) = umax(h,1);
```

```
            s(h,1) = 2*umax(h,1);
```

```
        end
```

```
        x(h,1+1) = x(h,1) + f(h,1) - u(h,1) - s(h,1);
```

```
        if h > 1
```

```
            x(h,1+1) = x(h,1+1) + u(h-1,1) + s(h-1,1);
```

```
        end
```

```
        if x(h,1+1) >= xmax(h,1)
```

```
            cont_eps(h,1+1)=1;
```

```
            u(h,1) = u(h,1) + x(h,1+1) - xmax(h,1) + s(h,1);
```

```
            s(h,1) = 0;
```

```
            x(h,1+1) = xmax(h,1);
```

```
            if u(h,1) >= umax(h,1)
```

```
                cont_sig(h,1)=1;
```

```
                s(h,1) = u(h,1) - umax(h,1);
```

```
                u(h,1) = umax(h,1);
```

```
            else
```

```
                if u(h,1) <= umin(h,1)
```

```
                    cont_sig(h,1) = 1;
```

```
                else
```

```
                    cont_sig(h,1) = 0;
```

```
                end
```

```
            end
```

```
        else
```

```
            if x(h,1+1) <= xmin(h,1)
```

```
                x(h,1+1) = x(h,1+1) + u(h,1) + s(h,1);
```

```
                u(h,1) = 0;
```

```
                s(h,1) = 0;
```

```
                if x(h,1+1) >= xmax(h,1)
```

```
                    cont_eps(h,1+1) = 1;
```

```
                    u(h,1) = u(h,1) + x(h,1+1) - xmax(h,1);
```

```

x(h,1+1) = xmax(h,1);
if u(h,1) >= umax(h,1)
    cont_sig(h,1) = 1;
    s(h,1) = u(h,1) - umax(h,1);
    u(h,1) = umax(h,1);
else
    if u(h,1) <= umin(h,1)
        cont_sig(h,1) = 1;
    else
        cont_sig(h,1) = 0;
    end
end
else
    if x(h,1+1) > xmin(h,1)
        cont_eps(h,1+1) = 0;
        cont_sig(h,1) = 1;
    else
        cont_sig(h,1) = 1;
        cont_eps(h,1+1) = 1;
    end
end
end
else
    cont_eps(h,1+1) = 0;
    u(h,1) = u(h,1) + s(h,1);
    s(h,1) = 0;
    if u(h,1) >= umax(h,1)
        cont_sig(h,1)=1;
        x(h,1+1) = x(h,1+1) + u(h,1) - umax(h,1);
        u(h,1) = umax(h,1);
        if x(h,1+1) >= xmax(h,1)
            cont_eps(h,1+1)=1;
            s(h,1) = x(h,1+1) - xmax(h,1);
            x(h,1+1) = xmax(h,1);
        else
            cont_eps(h,1+1)=0;
        end
    else
        if u(h,1) <= umin(h,1)
            cont_sig(h,1) = 1;
        else
            cont_sig(h,1) = 0;
        end
    end
end
end
end
end
% -----
%      DEFINICION DE VARIABLES AUXILIARES q, b Y d
% -----
q = f + M * s;
b = AV * ones(1,13) + (B/2) * q + (C/4) * (q.*q);
d = BV * ones(1,13) + C * q;

```

## A.7 FUNCIÓN QUE CALCULA LAS GANANCIAS TOTALES

G A N A N R E D . M

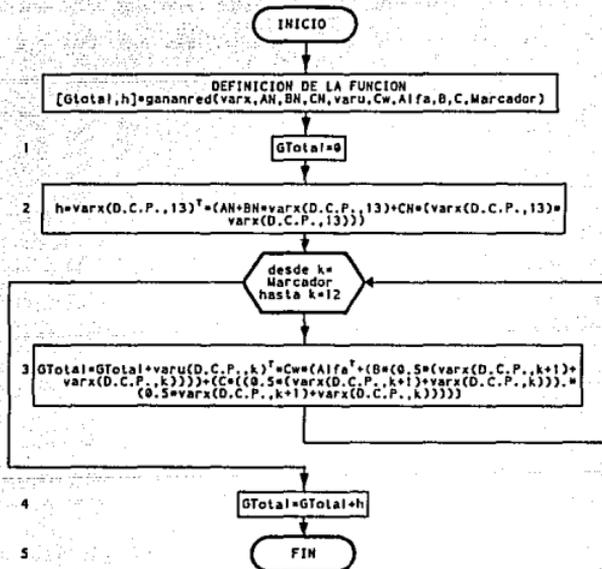
La función `GANANRED`, requiere de las siguientes variables de salida: `Gtotal`, `h`; además de los siguientes parámetros de entrada: `varx`, `AN`, `BN`, `CN`, `varu`, `Cw`, `Aifa`, `B`, `C`, `Mercador`.

- 1.- Limpiar la variable `Gtotal` para que anteriores resultados no afecten el resultado de la función actual.
- 2.- Calcular `h` de acuerdo a la ecuación (2.13).
- 3.- Utilizar una variable `k`, la cual se iguala a otra llamada `Mercador`, ésta controla el número de iteraciones a realizar en el paso 5 y cuenta con dos opciones: 1 ó 12 iteraciones.
- 4.- Calcular `Gtotal` de acuerdo a la ecuación (2.18) para encontrar el beneficio total, éste se calcula una vez para el beneficio real pasado y el beneficio real medio, y se calcula 12 veces para el beneficio a futuro.
- 5.- Sumar `Gtotal` y `h` para obtener el beneficio real total, de acuerdo a la ecuación (2.6).
- 6.- Salir de la función y retornar a el programa donde fué llamada.

# DIAGRAMA DE FLUJO

D.C.P. = DE CADA PRESA

## GANANRED



```
% NOMBRE DEL ARCHIVO: GANANRED.M
```

```
%
```

```
-----  
%           PROGRAMA QUE ENCUENTRA LA GANANCIA OBTENIDA EN LOS 12 MESES PARA  
%           EL SISTEMA DE PRESAS EN CASCADA  
% -----  
% -----  
%
```

```
-----  
%                                           P R O G R A M A  
% -----
```

```
function [GTotal,h] = ganared(varx,AN,BN,CN,varu,Cw,Alfa,B,C,Marcador)
```

```
GTotal=0;
```

```
h=varx(:,13)'*(AN+BN*varx(:,13)+CN*(varx(:,13).*varx(:,13)));
```

```
for k=Marcador:12
```

```
    GTotal=GTotal+varu(:,k)'*Cw*(Alfa'+(B*(0.5*(varx(:,k+1)+.  
        varx(:,k)))))+.  
        (C*((0.5*(varx(:,k+1)+varx(:,k)))*(0.5*(varx(:,k+1)+.  
        varx(:,k))))));
```

```
end
```

```
GTotal=GTotal+h;
```

## A.8 FUNCIÓN QUE SUMA LA GANANCIAS OBTENIDAS EN DOCE MESES

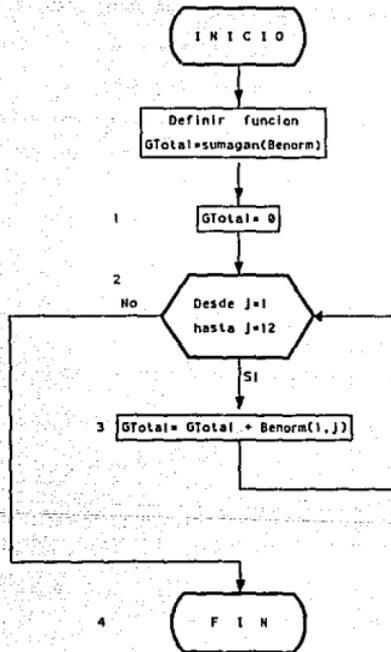
### S U M A G A N . M

La función SUMAGAN calcula el beneficio de los últimos doce meses dejándolo almacenado en `GTotal` y requiere como parámetro de entrada la variable `Benorm`, la cual contiene los beneficios de los últimos doce meses.

- 1.- Inicializar `GTotal` con el valor de cero.
- 2.- Si el apuntador del número de beneficios sumados `j`, es 13 ir al paso 4, en caso contrario continuar.
- 3.- Sumar a la ganancia total anual `GTotal`, el beneficio `Benorm` del `j`-ésimo mes. Incrementar `j` y regresar al paso 2.
- 4.- Salir de la función y retornar a el programa donde fué llamado.

# DIAGRAMA DE FLUJO

## SUMAGAN



```
% NOMBRE DEL ARCHIVO: SUMAGAN.M
```

```
%
```

```
%
```

```
%
```

```
-----  
PROGRAMA QUE ENCUENTRA LA GANANCIA OBTENIDA EN LOS 12 MESES PARA  
EL SISTEMA DE PRESAS EN CASCADA  
-----
```

```
%
```

```
%
```

```
%
```

```
%
```

```
%
```

```
-----  
P R O G R A M A  
-----
```

```
%
```

```
function GTotal = sumagan(Benorm)
```

```
GTotal=0;
```

```
for j=1:12
```

```
    GTotal=GTotal+Benorm(1,j);
```

```
end
```

## A.9 PROGRAMA QUE CALCULA LAS GANANCIAS MEDIAS REALES

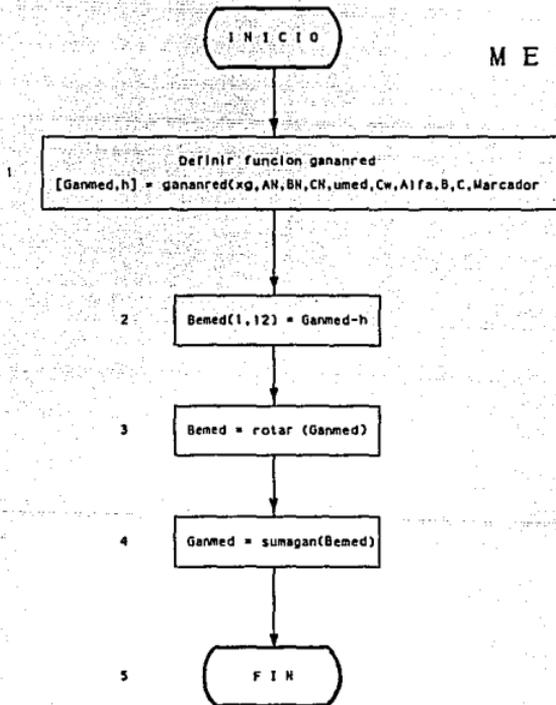
### MEDIA.M

El programa `media.m` calcula la ganancia anual y el beneficio de dejar agua almacenada al final del horizonte cuando se tiene la política en lazo abierto.

- 1.- Ejecutar la función `ganamed`, la cual calcula la ganancia media mensual real `ganmed` y la ganancia al final del horizonte `h`, tomando como parámetros de entrada los volúmenes reales `xg` y los volúmenes turbinados medios `umed`.
- 2.- Calcular la ganancia media anual real `Bemed` restándole a la ganancia total media anual real `ganmed` la contribución del término al final del horizonte `h` y almacenándolo en la columna 12 de `Bemed`, éste último contiene los doce beneficios más recientes.
- 3.- Rotar la matriz de ganancias medias reales `Bemed`, dejando el último valor calculado en la posición once y el más antiguo en la posición doce. Así al ser llamado del programa de nuevo, el valor de la columna doce de `Bemed` se pierde.
- 4.- Calcular la ganancia anual total `Ganmed` en lazo abierto por medio de la función `sumagan.m`.

# DIAGRAMA DE BLOQUES

MEDIA



```

% NOMBRE DEL ARCHIVO: MEDIA.M
%
% -----
% PROGRAMA QUE CALCULA LAS GANANCIAS MEDIAS CON DATOS REALES.
% -----
%
%
% P R O G R A M A
% -----

[Ganmed,h] = ganared(xg,AN,BN,CN,umed,Cw,Alfa,B,C,Marcador);
Bemed(1,12)=Ganmed-h;
Bemed = rotar(Bemed);
Ganmed=sumagan(Bemed),

```

## A.10 PROGRAMA QUE ROTA MATRICES

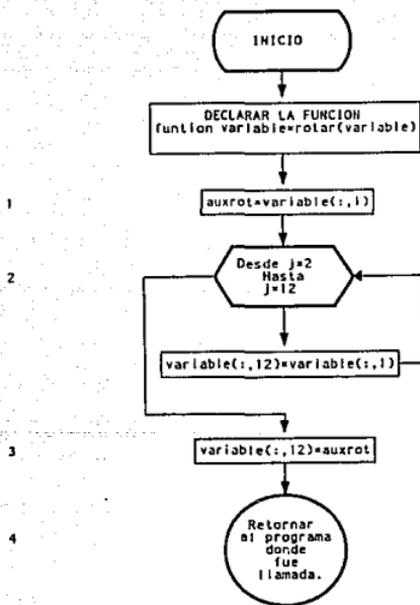
### R O T A R . M

La función `rotar.m` requiere como variable de entrada una matriz de doce columnas de la cual se desean mover las columnas una posición hacia la izquierda. Colocando la primera columna de la matriz en la doceava columna.

- 1.- Asignar a la variable auxiliar `suxrot` el valor de la primera columna de la matriz a rotar.
- 2.- Iterar el apuntador desde  $j=2$  hasta 12. Corriendo al mismo tiempo la columna  $j$  de la matriz a la posición  $j-1$ .
- 3.- Asignar a la columna doceava de la matriz, el valor almacenado inicialmente en `suxrot`.
- 4.- Retornar al programa donde fué llamado.

## DIAGRAMA DE FLUJO

### ROTAR



```
% NOMBRE DEL ARCHIVO: ROTAR.M
```

```
%
```

```
%
```

```
-----  
%          PROGRAMA QUE ENCUENTRA LA GANANCIA OBTENIDA EN LOS 12 MESES PARA  
%          EL SISTEMA DE PRESAS EN CASCADA  
-----
```

```
%
```

```
%
```

```
%
```

```
%
```

```
-----  
%          P R O G R A M A  
-----
```

```
%
```

```
function u = rotar(u)
```

```
auxrot=u(:,1);
```

```
for j=2:12
```

```
    u(:,j-1)=u(:,j);
```

```
end
```

```
u(:,12)=auxrot;
```

## A.11 PROGRAMA DE INICIALIZACIÓN

### INICIO.M

Este programa de inicialización debe de ser ejecutado únicamente la primera vez que se llama a `Benmax.m` y sirve para generar los archivos `datosvar.mat` y `datosinicia.mat` en formatos MATLAB. El programa de optimización requiere de estos archivos para su ejecución. Una vez creados el programa los consulta y los utiliza.

- 1.- Asignar con cero la variable `inicia`.
- 2.- Salvar en el archivo `datosinicia` la variable `inicia`.
- 3.- Ejecutar el programa de asignación `matin.m`, para inicializar las variables y dimensiones de las matrices a utilizar.
- 4.- Salvar en el archivo `datosvar.mat` aquellas variables que cambiarán su valor y que serán variables iniciales en la próxima corrida.

% NOMBRE DEL ARCHIVO: INICIO.M

%

%

-----  
%       PROGRAMA QUE INICIALIZA EL ARCHIVO DATOINICIA Y DATOSVAR, LOS  
%       CUALES CONTIENEN LOS DATOS FIJOS Y VARIABLES EN EL PROGRAMA  
%       -----

%

                                  P R O G R A M A

%

inicia=0;

save datoinicia inicia

matin

save datosvar x s f u meses umed xg Benorm Bemed indica

## APÉNDICE B

### ARCHIVOS EN FORMATO DE MATLAB

El archivo `datosinicia.mat` es generado la primera vez dentro del programa `inicia.m` y posteriormente es actualizado por `Benmax.m` y `Timereal.m` según corresponda. En él se encuentran en formato de MATLAB la bandera de control del sistema `inicia`. `inicia` controla la generación de los datos iniciales la primera vez que se corre el programa `BENMAX.M.` y debe ser igual a cero.

El archivo `datosver.mat` lo genera el programa de inicialización `inicia.m` y guarda las variables y valores requeridos durante la optimación. A continuación se describe el significado de las variables que lo conforman.

`r` = matriz de influjos mensuales

`u` = matriz de volúmenes turbinados

`meses` = matriz de meses del año.

`umed` = matriz de el volumen turbinado medio (este dato es variable porque se rota su valor al actualizar el mes siguiente, pero en valor es fijo).

`xg` = matriz del volumen real (dato que cambia mes con mes al introducir el volumen de la presa mensualmente.

`Benorm` = matriz del beneficio basado en los doce últimos volúmenes reales calculados.

`Bemed` = matriz del beneficio medio o en lazo abierto almacenando los doce valores más recientes.

`indica` = bandera que inhibe el cálculo de la ganancia la primera vez que se corre el programa `indica=0`; si `indica` es igual a uno se tienen corridas anteriores.

`x` = matriz que define al volumen turbinado

`s` = matriz que define el volumen derramado por vertederos

## APÉNDICE C

### PROGRAMAS DE ASIGNACIÓN

Los programas de asignación son programas con extensión `m` que asignan variables dentro del ambiente MATLAB o define dimensiones de matrices.

El programa `matin.m` define los datos y valores de las variables con los que se inicia la optimación y se enlista apartir de la siguiente página.

El programa `lluvias.m` permite asignar los influjos mensuales registrados desde el año de 1959 hasta 1988 y se utiliza en el cálculo de la optimación con la opción de influjos registrados previamente. (Su listado se presenta después de listar `matin.m`)

% NOMBRE DEL ARCHIVO: MATIN.M

%

%

-----  
% DATOS INICIALES PARA EL VOLUMEN DE LAS PRESAS ANGSTURA, CHICOASEN Y  
% MALPASO

% NOTA: CADA RENGLON CORRESPONDE A UNA PRESA EN EL ORDEN ANTES  
% MENCIONADO.

%

%

-----

% DATOS

%

%

format short  
format compact

% INFLUJOS MENSUALES f=3x13

f=[	320	227	205	196	275	820	1172	...
	1433	2244	1917	772	439	439		
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
	598.4	482.4	441.6	418.4	368	260.8	352.8	...
	407.2	500	528	650.4	596.8	596.8		

%

-----  
% VOLUMENES LIMITES DE LAS PRESAS

% MAXIMOS xmax=3x13

xmax={15549.20; 1419.80; 12373.10}\*ones(1,13);

% MINIMOS xmin=3x13

xmin={ 2379.53; 1169.19; 3055.71}\*ones(1,13);

%

-----  
% VOLUMENES TURBINADOS EN LAS PRESAS

% MAXIMOS umax=3x1

umax = {3032.64; 2418.34; 3732.48};

% MINIMOS umin=3x1

umin = {51.67; 100; 100};

%

```

% VOLUMEN ALMACENADO EN LAS PRESAS MENSUALMENTE      x=3x12
x=[4030; 1200; 5100]*ones(1,12);
% -----
% VALORES REALES DE X
xg=zeros(3,13);
% -----
% VALOR INICIAL DEL VOLUMEN TURBINADO MEDIO          u=3x13
umed=zeros(3,13);
% -----
% VALORES INICIALES DE GANANCIAS : MEDIAS Y NORMALES
Benorm=zeros(1,12);
Bemed=zeros(1,12);
% -----
% VALOR INICIAL DEL VOLUMEN TURBINADO                u=3x13
u=[ 51.67   51.67   51.67   51.67  826.67  826.67  826.67...
   930.00  930.00  930.00   51.67   51.67   51.67;
  1200.00 1200.00 1000.00 1000.00 1000.00 1350.00 1350.00...
  1500.00 1500.00 1500.00 1200.00 1200.00 1200.00;
  1200.00 1200.00 1000.00 1000.00 1000.00 1350.00 1350.00...
  1500.00 1500.00 1500.00 1200.00 1200.00 1200.00;
% -----
% AGUA DESECHADA POR VERTEDERO                      s=3x13
s=zeros(3,13);
% -----
% U CORRESPONDIENTE AL BENEFICIO MAXIMO
ubemmax=zeros(3,13);
% -----
% MATRIZ M                                           M=3x3
M=[-1  0  0;
   1 -1  0;
   0  1 -1];

```

```

% -----
%   MATRIZ Identidad           I=3x3
I=eye(3);
% -----
%   MESES DEL PERIODO ANUAL           meses=12x1
meses=[ ENEERO   ';;' FEBRERO   ';;' MARZO     ';;' ABRIL     ';;'
        MAYO    ';;' JUNIO     ';;' JULIO     ';;' AGOSTO    ';;'
        SEPTIEMBRE ';;' OCTUBRE  ';;' NOVIEMBRE ';;' DICIEMBRE '];
% -----
%   DECLARACION DE Alfa, Beta Y Gama           Alfa=Beta=Gama=1x3
Alfa=[ 88.8   490.5   70.4 ];
Beta=[ 0.007   0.0   0.012];
Gama=[-85e-9   0.0   -38e-9];
% -----
%   VECTOR      AN           AN=3x1
AN=[Alfa(1)+Alfa(2)+Alfa(3);
    Alfa(2)+Alfa(3);
    Alfa(3)];
% -----
%   MATRIZ      BN           BN=3x3
BN=[Beta(1)      Beta(2)/2      Beta(3)/2;
    Beta(2)/2    Beta(2)         Beta(2)/2;
    Beta(3)/2    Beta(2)/2      Beta(3) ];
% -----
%   MATRIZ      B           B=3x3
B=[Beta(1)   0.0   0.0 ;
   0.0   Beta(2)   0.0 ;
   0.0   0.0   Beta(3)];
% -----
%   MATRIZ DE COSTO MENSUAL      CN           CN=3x3
CN=[Gama(1) Gama(2) Gama(3);
    0.0   Gama(2) Gama(3);
    0.0   0.0   Gama(3)];

```

```

% -----
% VECTOR AV AV=3x1
AV=Alfa';
% -----
% VECTOR BV TRANSPUESTO BV=3x1
BV=Beta';
% -----
% VECTOR C C=3x3
C=[Gama(1) 0.0 0.0 ;
0.0 Gama(2) 0.0 ;
0.0 0.0 Gama(3)];
% -----
% MATRIZ N=M+2I N=3x3
enhe = M + 2 * I;
% -----
% MATRIZ L = M+I L=3x3
L = M + I;
% -----
% VARIABLE INICIA, ESTA VARIABLE SE OCUPA CUANDO INTRODUCE DATOS EL
% OPERADOR.
aparece=0; %variable inhibidora de impresion de resultados
% en tiempo real
indica=0;
Bemax=0;
% -----
% VARIABLE CW DE COSTOS CW=3x3
Cw= I*2e6;
% -----
% VALOR DE EPSILON INICIALIZADO epsilon=3x13
epsilon=zeros(3,13);

```

```

% -----
% VALOR DE EPSILON AUXILIAR INICIALIZADO          epsilon_nue=3x13
epsilon_nue=zeros(3,13);
% -----
% CONDICION FINAL AL FINAL DEL HORIZONTE          x(:,13)=3x1
% SON LOS VALORES A LOS CUALES SE DESEA LLEGAR AL TERMINO DEL PERIODO
xdes=[15549.20; 1419.80; 12373.10];
% -----
% MATRIZ PARA CONTROLAR LAS EPSILON, INICIALMENTE EN 1;
cont_eps=zeros(3,13);
% -----
% MATRIZ PARA CONTROLAR LAS SIGMAS, INICIALMENTE EN 1;
cont_sig=zeros(3,13);
% -----
% VALOR DE SIGMA INICIALIZADO                      sigma=3x13
sigm=zeros(3,13);
% -----
% INICIALIZANDO LAMDA Y MU AL FINAL DEL HORIZONTE
mu(:,13)=CN*xdes;
lamda(:,13)=2*BN*xdes+2*(mu(:,13). *xdes)+CN*(xdes.*xdes)+AN;
% -----
% INICIALIZANDO ALFA Y EL NUMERO DE ITERACIONES TOTAL
pmax=20;          % No. DE ITERACIONES
alfa=-0.5;       % VALOR DE ALFA
% -----
% INICIALIZANDO EL No. DE AÑOS Y EL AÑO INICIAL
noan=30;         % No. DE AÑOS
anin=1958;      % AÑO INICIAL

```

```
% -----  
%   VARIABLE PARA HACER OPCIONAL EL CALCULO DE LA GANANCIA MEDIA  
opmed=0;           % 0=NO REALIZA EL CALCULO DE LA GANANCIA MEDIA  
                  % 1=REALIZA EL CALCULO DE LA GANANCIA MEDIA  
% -----  
%   VARIABLE INICIALIZADA DE LA GANANCIA MEDIA  
Ganmed=0;         % VALOR INICIAL POR SI NO SE REQUIERE LA OPCION  
% -----
```

% INFLUJOS PARA EL AÑO DE 1959 A 1988

fvar =	315.503	227.471	207.451	173.837	179.786	464.257 ...
	813.336	865.081	2243.111	1083.081	448.191	393.770;
	0.0	0.0	0.0	0.0	0.0	0.0 ...
	0.0	0.0	0.0	0.0	0.0	0.0;
	1001.0	778.0	637.103	702.371	443.911	1389.028 ...
	1045.681	1787.697	1596.825	3664.854	2001.803	962.597;
	277.104	200.435	189.066	151.089	166.792	1145.193 ...
	1713.409	1549.183	1897.410	1809.429	817.498	488.659;
	0.0	0.0	0.0	0.0	0.0	0.0 ...
	0.0	0.0	0.0	0.0	0.0	0.0;
	773.807	516.409	466.502	407.050	406.328	1655.896 ...
	2235.566	2896.560	4357.256	2927.538	2136.589	1163.738;
	346.583	233.523	214.988	194.614	215.508	695.081 ...
	652.896	1065.197	877.742	1948.119	811.156	474.424;
	0.0	0.0	0.0	0.0	0.0	0.0 ...
	0.0	0.0	0.0	0.0	0.0	0.0;
	1087.416	662.580	612.724	452.802	482.922	1115.955 ...
	1541.408	1757.538	1930.496	2553.647	2761.279	871.240;
	304.508	211.249	177.305	160.433	221.233	1347.977 ...
	1488.103	1775.921	2966.093	2335.907	1240.649	573.869;
	0.0	0.0	0.0	0.0	0.0	0.0 ...
	0.0	0.0	0.0	0.0	0.0	0.0;
	803.401	456.221	525.793	609.941	443.146	1267.461 ...
	2114.221	2235.505	5439.029	3624.581	1371.971	750.274;
	387.816	284.954	244.652	212.288	223.902	750.182 ...
	987.491	995.872	1269.345	1248.285	1475.501	498.576;
	0.0	0.0	0.0	0.0	0.0	0.0 ...
	0.0	0.0	0.0	0.0	0.0	0.0;
	678.906	556.391	458.068	313.956	324.439	593.338 ...
	2651.137	2038.187	5149.066	3348.214	1606.493	1044.18;
	340.738	227.181	198.354	203.219	201.382	847.096 ...
	1138.334	1578.805	3272.489	2141.596	777.417	419.364;
	0.0	0.0	0.0	0.0	0.0	0.0 ...
	0.0	0.0	0.0	0.0	0.0	0.0;
	906.768	533.261	433.861	376.207	377.993	1429.805 ...
	3375.749	2031.557	3612.060	3008.599	1012.709	1293.912;
	299.334	208.453	184.953	15;.192	171.806	393.868 ...
	1388.597	1329.945	2431.993	1814.924	877.761	482.909;
	0.0	0.0	0.0	0.0	0.0	0.0 ...
	0.0	0.0	0.0	0.0	0.0	0.0;
	820.628	636.322	491.648	355.079	407.746	1303.843 ...
	1806.699	2534.337	2739.176	5135.918	1851.430	1201.262;

315.787	206.928	173.431	144.338	179.565	764.009	...
1896.390	1170.080	2142.115	1861.120	649.272	474.488;	
0.0	0.0	0.0	0.0	0.0	0.0	...
0.0	0.0	0.0	0.0	0.0	0.0;	
1001.000	778.000	733.000	696.000	711.000	1684.000	...
2644.000	2117.000	3765.000	3932.000	1674.000	998.200;	
304.329	210.884	187.729	149.885	173.835	769.595	...
1106.282	1626.370	1887.906	2608.112	997.323	506.790;	
0.0	0.0	0.0	0.0	0.0	0.0	...
0.0	0.0	0.0	0.0	0.0	0.0;	
899.600	509.000	467.100	392.600	331.300	974.100	...
966.700	1485.000	1970.000	3068.000	1135.000	1007.000;	
361.389	261.659	232.685	243.791	274.292	926.391	...
1691.764	1331.733	2298.372	2491.445	1012.112	488.757;	
0.0	0.0	0.0	0.0	0.0	0.0	...
0.0	0.0	0.0	0.0	0.0	0.0;	
854.800	534.600	415.800	325.500	440.500	1333.000	...
1918.000	1051.000	3304.000	3028.000	1226.000	905.200;	
354.211	246.599	211.371	200.120	169.340	474.051	...
572.263	873.176	965.014	1623.284	613.672	347.926;	
0.0	0.0	0.0	0.0	0.0	0.0	...
0.0	0.0	0.0	0.0	0.0	0.0;	
711.100	419.000	630.200	329.200	429.000	975.700	...
2456.000	5019.000	6205.000	3963.000	2144.000	1228.000;	
260.124	188.265	160.733	138.269	250.226	857.599	...
116.873	577.359	1936.899	1886.543	738.919	444.033;	
0.0	0.0	0.0	0.0	0.0	0.0	...
0.0	0.0	0.0	0.0	0.0	0.0;	
883.300	682.300	497.900	441.400	552.100	995.300	...
2896.000	3673.000	6561.000	3983.000	1869.000	1058.000;	
302.356	199.296	173.988	158.570	220.141	651.453	...
1455.801	3183.813	4208.383	2680.326	1171.989	633.943;	
0.0	0.0	0.0	0.0	0.0	0.0	...
0.0	0.0	0.0	0.0	0.0	0.0;	
764.300	516.500	526.800	451.100	537.600	815.200	...
1224.000	2870.000	3792.000	3450.000	1507.000	885.300;	
422.511	292.955	247.975	193.670	213.503	517.887	...
1837.979	2230.915	2974.214	2386.594	1129.121	598.988;	
0.0	0.0	0.0	0.0	0.0	0.0	...
0.0	0.0	0.0	0.0	0.0	0.0;	
787.700	479.500	401.700	394.300	449.400	1483.000	...
1609.000	1740.000	1614.000	1376.000	841.700	852.700;	
400.792	269.457	243.030	199.569	217.369	412.548	...
651.310	1959.940	2283.426	2409.320	839.079	495.045;	
0.0	0.0	0.0	0.0	0.0	0.0	...

0.0	0.0	0.0	0.0	0.0	0.0;
546.400	429.400	317.700	406.500	408.500	1199.000 ...
1393.000	5119.000	4971.000	4627.000	1918.000	1195.000;
344.709	253.940	213.082	185.307	257.040	866.420 ...
891.851	1128.125	1071.315	862.339	465.796	298.556;
0.0	0.0	0.0	0.0	0.0	0.0 ...
0.0	0.0	0.0	0.0	0.0	0.0;
1052.700	1174.100	1043.300	1018.800	913.200	489.000 ...
1281.500	2015.100	1828.700	1843.000	1984.400	1610.900;
227.617	165.262	146.688	129.601	146.228	607.705 ...
751.310	2438.481	2613.132	3261.019	1114.238	625.163;
0.0	0.0	0.0	0.0	0.0	0.0 ...
0.0	0.0	0.0	0.0	0.0	0.0;
1395.700	1358.400	481.200	582.600	1101.900	1035.100 ...
2022.900	1186.200	1449.900	1553.500	1546.800	1770.300;
433.000	284.000	268.000	211.300	257.000	455.000 ...
730.000	463.000	1412.000	1007.000	338.000	247.000;
0.0	0.0	0.0	0.0	0.0	0.0 ...
0.0	0.0	0.0	0.0	0.0	0.0;
1570.000	895.000	891.200	822.300	558.200	1247.100 ...
1611.100	2186.900	3394.900	2222.900	1962.800	1986.700;
161.000	109.000	90.000	58.000	130.000	319.000 ...
540.000	958.000	2144.000	2081.000	961.000	442.000;
0.0	0.0	0.0	0.0	0.0	0.0 ...
0.0	0.0	0.0	0.0	0.0	0.0;
1578.800	691.000	863.800	625.700	91.400	191.600 ...
712.000	908.000	2739.000	1812.300	1471.700	1232.200;
593.345	516.727	512.742	535.044	626.280	1120.510 ...
1100.290	1398.131	1268.504	783.604	454.289	147.919;
0.0	0.0	0.0	0.0	0.0	0.0 ...
0.0	0.0	0.0	0.0	0.0	0.0;
1092.300	1341.700	932.400	1245.700	1208.100	1553.800 ...
1821.300	1887.900	1819.600	1757.800	1799.800	1693.000;
297.400	160.500	158.700	220.600	298.900	729.800 ...
467.700	1160.200	1596.600	845.700	526.200	366.700;
0.0	0.0	0.0	0.0	0.0	0.0 ...
0.0	0.0	0.0	0.0	0.0	0.0;
1616.100	1378.900	1455.700	1075.400	1272.400	1578.400 ...
1473.200	1433.200	2187.000	1699.700	1588.300	1039.300;
392.400	168.600	200.500	139.100	478.200	722.700 ...
1418.800	1584.300	2346.300	1599.700	549.700	429.600;
0.0	0.0	0.0	0.0	0.0	0.0 ...
0.0	0.0	0.0	0.0	0.0	0.0;
1371.700	1179.000	1115.000	1313.400	1024.700	975.800 ...
919.300	1041.800	2277.500	1681.600	1585.500	1313.400;

432.900	238.200	182.300	209.200	341.000	830.600	...
1227.600	1580.500	3628.500	1689.800	781.500	546.500;	
0.0	0.0	0.0	0.0	0.0	0.0	...
0.0	0.0	0.0	0.0	0.0	0.0;	
1394.700	1270.000	1269.100	1163.500	1202.600	1350.000	...
1234.500	1362.900	2418.300	1777.800	1647.800	1314.700;	
348.200	261.700	219.700	302.800	323.600	571.900	...
781.900	1398.100	2655.300	1848.200	584.300	397.800;	
0.0	0.0	0.0	0.0	0.0	0.0	...
0.0	0.0	0.0	0.0	0.0	0.0;	
1611.000	1530.800	1639.300	1220.700	1341.700	1425.800	...
1504.300	1565.800	1620.100	2049.900	1604.900	1720.200;	
312.600	198.200	262.600	367.700	261.000	1436.200	...
1939.700	2388.100	3197.800	2813.900	909.100	577.200;	
0.0	0.0	0.0	0.0	0.0	0.0	...
0.0	0.0	0.0	0.0	0.0	0.0;	
1742.000	1016.900	1015.000	1176.500	1655.200	1219.200	...
966.700	1095.800	1206.400	1231.600	1015.200	794.400;	
249.700	263.400	229.200	214.900	540.900	1471.300	...
981.600	961.200	2088.900	2345.800	681.200	434.600;	
0.0	0.0	0.0	0.0	0.0	0.0	...
0.0	0.0	0.0	0.0	0.0	0.0;	
741.000	692.400	807.900	739.600	758.300	916.200	...
702.900	703.600	1431.300	1303.700	1113.700	804.800;	
284.700	294.900	292.100	219.800	201.200	713.500	...
1136.400	1329.300	2928.500	1078.700	629.800	407.300;	
0.0	0.0	0.0	0.0	0.0	0.0	...
0.0	0.0	0.0	0.0	0.0	0.0;	
1209.900	1073.600	930.200	828.500	715.000	847.800	...
995.500	1438.100	2033.800	2307.900	1346.400	1027.900;	
258.900	195.700	243.400	207.600	656.100	1511.800	...
1966.300	2559.400	3972.100	1902.100	632.400	476.700;	
0.0	0.0	0.0	0.0	0.0	0.0	...
0.0	0.0	0.0	0.0	0.0	0.0;	
1088.400	934.300	1024.100	988.700	1256.100	1098.400	...
1304.300	1453.200	3005.000	2098.300	1312.200	1373.700;	
315.600	239.200	254.900	196.100	373.500	754.900	...
1137.500	2030.200	2034.700	1499.000	775.100	333.000;	
0.0	0.0	0.0	0.0	0.0	0.0	...
0.0	0.0	0.0	0.0	0.0	0.0;	
1209.200	964.500	1202.600	1275.000	1336.100	1265.700	...
1289.600	1011.700	966.200	951.200	1091.400	1481.100;	
336.600	188.900	168.200	256.000	372.300	876.800	...
1073.600	1154.500	1225.600	792.100	470.100	293.500;	
0.0	0.0	0.0	0.0	0.0	0.0	...

0.0	0.0	0.0	0.0	0.0	0.0;
846.700	902.100	704.900	846.500	864.500	981.400 ...
671.200	604.600	672.400	1330.800	1375.500	962.400];

## APÉNDICE D

### ARCHIVOS EN CÓDIGO ASCII

A continuación se describen brevemente los archivos generados por los programas `timer2.m` o `timereal.m` (según sea el caso) en formato ASCII, para que puedan ser leídos por el programa de graficación escrito en lenguaje C; así mismo, también se describen los archivos generados y utilizados dentro del mismo programa escrito en lenguaje C.

#### `Voiturb.ent`

Este archivo contiene predicciones de los volúmenes de las presas, predichos para el siguiente mes y los volúmenes a turbinar en el mes actual. Es creado desde el programa principal, ya sea por la subrutina `timereal.m` o `timer2.m`, dependiendo de el procedimiento que se esté empleando.

#### `Turbina.ent`

Este archivo es creado desde el programa escrito en lenguaje C, como auxiliar en el procedimiento utilizado para convertir variables enteras. En particular convierte el volumen a turbinar en el mes actual, a su equivalente en ASCII, para que puedan dibujarse en el monitor.

#### `Meses.ent`

Este archivo contiene los meses del año a partir del actual. Es creado desde el programa en C y corre paralelamente al archivo `datosver.mat`, el cual contiene, entre otras cosas, el nombre de los meses utilizados dentro del programa en MATLAB. La razón por la que este archivo no es leído directamente desde el lenguaje de programación C, es que en MATLAB sólo se almacenan enteros, mientras que en C es necesario leer este archivo como un conjunto de cadenas de caracteres.

**Litdiv.ent**

Este archivo contiene los volúmenes de las presas para el mes actual divididos entre dos, esto es debido a que la animación se hace en dos pasos y es necesario guardar los valores de dichos puntos para su posterior utilización. Este archivo es creado desde el programa en C.

**Benef.ent**

Este archivo contiene los beneficios del sistema, para los 12 meses anteriores y es creado desde el programa en C, para la graficación de los beneficios. De éste archivo se irá eliminando el valor del último mes; una vez recorridos los otros 11 valores, se tomará el beneficio del mes actual, del archivo **Unben.ent** para completar los 12 meses del año en curso anterior.

**Benmed.ent**

Este archivo, creado desde el programa en C, contiene los beneficios medios del sistema, si es que se opta el cálculo de los mismos, para los 12 meses anteriores. De éste archivo se irá eliminando el valor del último mes; una vez recorridos los otros 11 valores y se tomará el beneficio del mes actual, del archivo **Unben.ent** para completar los 12 meses del año.

**Unben.ent**

Este archivo contiene los beneficios del sistema con la política en lazo abierto y con la real, para el mes actual. Es creado desde el programa principal en MATLAB, ya sea por la subrutina **tiereal.m** o **tierr2.m**, dependiendo de el procedimiento que se esté empleando.

## REFERENCIAS Y BIBLIOGRAFIA

[ 1 ] DRA. CRISTINA VERDE; POLÍTICAS DE OPERACIÓN PARA CENTRALES HIDROELÉCTRICAS; PROYECTO 9127, NOVIEMBRE 1991; INSTITUTO DE INGENIERÍA, UNAM;

[ 2 ] DR. RAMÓN DOMINGUEZ M; OPERACIÓN INTEGRAL DEL SISTEMA HIDROELÉCTRICO DEL RÍO GRIJALVA; PROYECTO 2307, ABRIL 1993; INSTITUTO DE INGENIERÍA, UNAM;

[ 3 ] DAVID G LUENBERGER; LINEAR AND NOLINEAR PROGRAMMING; EDITORIAL ADDISON WESLEY;

[ 4 ] MANUAL DE REFERENCIA DE LENGUAJE C++; BORLAND VERSION 1.0 EDITORIAL McGRAW HILL;