


Caracas, 14 de junio de 2.006

## ACTA

Los abajo firmantes, miembros del jurado por el Consejo de Escuela de Ingeniería Mecánica, para evaluar el Trabajo Especial de Grado presentado por los bachilleres: **COLINA ERNESTO** y **FONTEN WILLIAM**

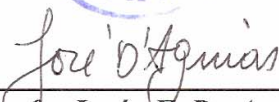
**“MODELIZACION DE FLUJO BIDIMENSIONAL A SUPERFICIE LIBRE UTILIZANDO EL METODO DE BEAM AND WARMING MODIFICADO”**

Consideran que el mismo cumple con los requisitos exigidos por el plan de estudio conducente al Título de Ingeniero Mecánico.

  
\_\_\_\_\_  
**Prof. Josué Leon**  
Jurado



  
\_\_\_\_\_  
**Prof. Rodolfo Berrios**  
Jurado

  
\_\_\_\_\_  
**Prof. José F. De Aguiar**  
Tutor

“Hacia el 2008: Conmemoración del Cincuentenario de la Autonomía Universitaria”

TRABAJO ESPECIAL DE GRADO

**MODELIZACIÓN DE FLUJO BIDIMENSIONAL A SUPERFICIE LIBRE  
UTILIZANDO EL MÉTODO DE BEAM AND WARMING MODIFICADO**

Presentado por los Bachilleres:  
Colina H. Ernesto A.  
Fontén G. William M.  
ante la Universidad Central de Venezuela  
para optar al Título de Ingeniero Mecánico

Caracas, 2006

TRABAJO ESPECIAL DE GRADO

**MODELIZACIÓN DE FLUJO BIDIMENSIONAL A SUPERFICIE LIBRE  
UTILIZANDO EL MÉTODO DE BEAM AND WARMING MODIFICADO**

Tutor Académico: Prof. José F. De Aguiar

Presentado por los Bachilleres:  
Colina H. Ernesto A.  
Fontén G. William M.  
ante la Universidad Central de Venezuela  
para optar al Título de Ingeniero Mecánico

Caracas, 2006

A:

Maria E., Gómez A.

Blanca C., Gómez A.

Hilda M., Gómez A.

Anilda J., Hernández

Ernesto Colina P.

Marnie J., Colina H.

Tania R., De Sousa N.

## **AGRADECIMIENTOS**

Nos gustaría expresar nuestro reconocimiento a todos los que colaboraron, de una manera u otra, en la realización de este trabajo. Particularmente, a nuestro tutor y director del proyecto, el profesor José F. De Aguiar, por habernos guiado meticulosamente en la realización del mismo.

Valoramos también agradecer a la Sra. Marnie Colina, Sra. Elvia Gómez, Sra. Blanca Gómez, Sra. Hilda Gómez y la Srta. Tania De Sousa por brindarnos apoyo económico y suministro de material para la construcción de este trabajo. Al Sr. Simón Martínez por su asesoramiento en como debe realizarse estas obras.

Finalmente, le enviamos nuestro más caluroso agradecimiento a nuestros padres por su siempre apoyo incondicional en la ejecución de los proyectos que nos ayudará a alcanzar el crecimiento profesional, lo mismo que a los miembros de mi familia y amigos.

**Colina H., Ernesto A. y Fonten G., William M.**

**MODELIZACIÓN DE FLUJO BIDIMENSIONAL A SUPERFICIE LIBRE  
UTILIZANDO EL MÉTODO DE BEAM AND WARMING MODIFICADO**

**Tutor Académico: Prof. De Aguiar D., José F. Tesis. Caracas. U.C.V. Facultad de Ingeniería. Escuela de Ingeniería Mecánica. 2006. 122 pág.**

Palabras Clave: Saint Venant, Método Beam Warming, Jacobiano Roe, Batimetría.

En el presente trabajo se modelaron las ecuaciones de Saint Venant, utilizando una modificación del método de Beam and Warming para que sea conservativo. La modificación del método estuvo basada en el Jacobiano aproximativo de Roe.

Se realizó un análisis exhaustivo de todo lo concerniente al modelo hidrodinámico, los detalles relacionados al esquema de Beam and Warming, la modificación propuesta, las condiciones fronteras y el criterio de estabilidad.

Resultados de pruebas bidimensionales se compararon con las soluciones analíticas y experimentales; y se obtuvieron también por otros métodos numéricos para comprobar la mejor aproximación. Se demostró que el método de Beam and Warming modificado obtuvo resultados bastantes aproximados a los analíticos.

La capacidad de aplicación del modelo en el campo industrial fue puesta en evidencia a través de ejemplos reales de simulación de los flujos combinados con fronteras irregulares y una batimetría compleja.

## ÍNDICE

DEDICATORIA.....	III
AGRADECIMIENTOS.....	IV
RESUMEN.....	V
ÍNDICE.....	VI
LISTA DE TABLAS.....	VIII
LISTA DE FIGURAS.....	IX
INTRODUCCIÓN.....	1
CAPÍTULO I: REVISIÓN DE LITERATURA.....	4
CAPÍTULO II: ECUACIONES DE SAINT-VENANT.....	5
2.1 Introducción.....	5
2.2 Hipótesis de las ecuaciones de Saint-Venant.....	5
2.3 Las ecuaciones gobernantes.....	6
2.4 Las ecuaciones de Saint-Venant en forma matricial.....	7
CAPÍTULO III: MÉTODO NUMÉRICO.....	9
3.1 Introducción.....	9
3.2 Esquema de Beam and Warming.....	9
3.3 Formulación general.....	10
3.4 Esquema Factorial.....	14
3.5 Descomposición de flujo implícito factorial.....	15
3.6 Diferenciación Espacial.....	16
3.7 Condiciones iniciales.....	22
3.8 Implantación de las condiciones fronteras.....	23
3.8.1 Para una frontera cerrada.....	23
3.8.2 Para una frontera abierta.....	23
3.8.2.1 Flujo Supercrítico.....	24
3.8.2.2 Flujo Subcrítico.....	24
3.9 Criterio de estabilidad.....	24

3.10 Modificación Sugerida.....	25
CAPÍTULO IV: RESULTADOS Y DISCUSIÓN.....	32
4.1 Introducción.....	32
4.2 Ensayos Bidimensionales.....	32
4.2.1 Canal con expansión brusca.....	32
4.2.2 Depósito Circular.....	39
4.2.3 Vorticidad.....	43
4.2.4 Contracción de canal abierto.....	46
4.2.5 Caso real Lago de Maracaibo.....	48
CONCLUSIÓN.....	52
RECOMENDACIONES.....	54
REFERENCIAS BIBLIOGRÁFICAS.....	55
APÉNDICE A: DESCRIPCIÓN DEL PROGRAMA.....	57
A.1 Descripción de las subrutinas.....	57
A.2 Descripción de archivos de datos.....	59
APÉNDICE B: LISTADO DEL PROGRAMA.....	65
APÉNDICE C: MUESTRA DE ENTRADA Y SALIDA DEL PROGRAMA.....	103
C.1 Muestra de entrada.....	105
C.2 Muestra de salida.....	106



## **LISTA DE TABLAS**

Tabla 3.1: Indicativo de diferentes formulaciones del Esquema de Beam and Warming.....	10
---	----

## LISTA DE FIGURAS

Figura 3.1: Malla de cálculo del esquema	
Beam and Warming.....	11
Figura 3.2: Malla del cálculo del esquema Beam and Warming	
Modificado.....	30
Figura 4.1: Dominio del cálculo (canal con expansión brusca).....	33
Figura 4.2: Descripción de las variables.....	33
Figura 4.3: Experiencias de los diferentes investigadores.....	35
Figura 4.4: Vectores velocidades para ( $t = 500 \text{ s}$ , $\Delta x = 100$ , $\Delta y = 100$ , $Q = 1 \text{ m}^3/\text{min.}$ , y $\Delta t = 10 \text{ s}$ ).....	36
Figura 4.5: Perfil tridimensional de la expansión brusca	
para ( $Z = 3 \text{ m}$ , $D = 2 \text{ m}$ ).....	37
Figura 4.6: Perfil tridimensional del fondo (expansión brusca).....	37
Figura 4.7: Curva $U^*$ en función de $Y^*$ para ( $(X - X_r) / W_1 = - 0.6$ (sección 1)).....	38
Figura 4.8: Curva $U^*$ en función de $Y^*$ para ( $(X - X_r) / W_1 = 1.4$ (sección 3)).....	38
Figura 4.9: Curva $U^*$ en función de $Y^*$ para ( $(X - X_r) / W_1 = - 2.6$ (sección 2)).....	39
Figura 4.10: Dominio del cálculo (depósito circular).....	40
Figura 4.11: Vectores velocidades para ( $t = 150 \text{ s}$ , $\Delta x = 15$ , $\Delta y = 15$ , $Q = 500 \text{ m}^3/\text{min.}$ y $\Delta t = 3 \text{ s}$ ).....	41
Figura 4.12: Perfil tridimensional del depósito circular	
para ( $Z = 15 \text{ m}$ , $D = 1 \text{ m}$ ).....	42
Figura 4.13: Perfil tridimensional del fondo (depósito circular).....	42
Figura 4.14: Vectores velocidades para ( $t = 300 \text{ s}$ , $\Delta x = 10$ , $\Delta y = 10$ , $Q = 15000 \text{ m}^3/\text{min.}$ y $\Delta t = 6 \text{ s}$ ).....	44
Figura 4.15: Perfil tridimensional de vorticidad.....	45

Figura 4.16: Perfil tridimensional del fondo para ( $Z = 3$ m, $D = 3$ m).....	45
Figura 4.17: Vectores velocidades para ( $t = 250$ s, $\Delta x = 100$ , $\Delta y = 100$ , $Q = 7500$ m <sup>3</sup> /min. y $\Delta t = 5$ s).....	46
Figura 4.18: Perfil tridimensional de contracción de canal abierto.....	47
Figura 4.19: Perfil tridimensional del fondo para ( $Z = 3$ m, $D = 2$ m).....	47
Figura 4.20: Vectores velocidades para ( $t = 10000$ s, $\Delta x = 1$ Km., $\Delta y = 1$ Km., $Q = 6000$ m <sup>3</sup> /min. y $\Delta t = 200$ s).....	50
Figura 4.21: Perfil tridimensional del Lago de Maracaibo.....	51
Figura 4.22: Perfil tridimensional del fondo para ( $Z = 30$ m, $D = 5-30$ m).....	51
Figura A.1: Definiciones límites y tipos de celdas.....	63
Figura A.2: Análisis de resultado tipo límite.....	64
Figura C.1: Vectores velocidades para ( $t = 750$ s, $\Delta x = 100$ m, $\Delta y = 100$ m, $Q = 7500$ m <sup>3</sup> /min. y $\Delta t = 15$ s).....	103
Figura C.2: Perfil tridimensional (data muestra).....	104
Figura C.3: Perfil tridimensional del fondo para ( $Z = 15$ m, $D = 5$ m).....	104

## INTRODUCCIÓN

Los problemas que tratan de flujo a superficie libre han sido estudiados durante mucho tiempo. Los sistemas de ecuaciones diferenciales parciales que rigen estos fenómenos no pueden ser resueltos analíticamente a excepción de ciertos casos simplificados. Es para esto que las soluciones numéricas fueron el centro de interés de estudio. Varios modelos de flujo a superficie libre han sido desarrollados en búsquedas anteriores (Abbott 1979, Cunge y Al 1980, Abbott y Basco 1989).

Las ecuaciones son hiperbólicas es por eso que pensamos en métodos de las características. Pero es muy difícil de implementar en dos dimensiones. Hasta en una dimensión, si un choque se forma.

Los métodos de diferencias finitas han sido desarrollados anteriormente. Por ejemplo, el método de Preissmann, que con frecuencia tiene una utilización errónea en las regiones subcríticas y supercríticas. Ésta no posee la propiedad del método de las derivadas espaciales.

Ionesco y Abbott (1967) desarrollaron un método adecuado para las derivadas espaciales a través de una función de control.

Fenema y Chaudhry (1987), fueron los primeros que utilizaron el esquema de Beam and Warming para problemas de flujo a superficie libre. También, varias aportaciones al método anterior han sido realizadas por Tchamen y Kahawita (1992). El esquema de Beam and Warming es completamente no conservativo porque varios términos son evaluados de un modo no conservativo. Esto trae como consecuencia un error significativo en la solución.

Para las ecuaciones de Euler en el gas dinámico, varios métodos de diferencias finitas han sido desarrollados y ellos reemplazan las propiedades conservativas. Estos métodos han sido resueltos esencialmente con el esquema de Riemann.

Roe (1981) propone una aproximación a los resultados de Riemann. El construye un aproximado para las ecuaciones de Euler, que satisfacen ciertas propiedades relacionadas a la conservación y a la consistencia de las ecuaciones gobernantes.

Glaister (1988) utiliza la técnica de Roe (1981), en las ecuaciones de flujo a superficie libre, a través de una aproximación de Riemann. Ésta es idéntica a la hecha por Roe (1981).

El presente trabajo propone, en su primera parte, una modificación del método de Beam and Warming para que este sea conservativo a través del concepto del jacobiano aproximativo, y como segunda parte, la corrida de ejemplos bidimensionales teóricos y reales. La superioridad del modelo propuesto es demostrada en los capítulos siguientes.

El primer capítulo presenta una revisión no exhaustiva de los estudios que se remiten a este proyecto, de la resolución de las ecuaciones de Saint-Venant. El segundo capítulo presenta las ecuaciones de Saint-Venant, así como una discusión breve de sus variables.

El tercer capítulo presenta la aplicación del esquema de Beam and Warming con la modificación propuesta y su algoritmo de resolución.

El cuarto capítulo contiene una serie de pruebas bidimensionales para la validación del modelo hidrodinámico desarrollado en el capítulo tres.

## CAPÍTULO I

### REVISIÓN DE LITERATURA

En este capítulo, presentamos una revisión no exhaustiva de la literatura que concierne a los flujos a superficie libre.

Las ecuaciones diferenciales parciales unidimensionales de Saint-Venant son conocidas desde hace mucho tiempo, Saint-Venant (1871). Hansen (1962) fue probablemente el primero que presenta la formulación de profundidad media en dos dimensiones. Más tarde, varios otros investigadores como Leendertse (1967), Küipers y Vreugdenhil (1973), Lai (1986), García y Kahawita (1983) aplicaron los conceptos de modelización bidimensional a un gran número de situaciones de flujos a superficie libre.

Los primeros que desarrollan un método adecuado para las derivadas espaciales a través de una función de control fueron Abbott y Ionesco (1967).

Beam and Warming (1976) desarrollaron un método que fue utilizado en dinámica de los fluidos. El primero que hay que utilizar de Beam and Warming en las ecuaciones de flujo a superficie libre es Fennema (1985). Trabajos posteriores sobre este método han sido hechos por Tchamen y Kahawita (1992).

Roe (1981) propone una aproximación a la solución de Riemann y él construye allí un jacobiano aproximativo para las ecuaciones de Euler, que responde a ciertas propiedades relacionadas a la conservación y la consistencia de las ecuaciones gobernantes. Glaister (1988), fue el primero que emplea la técnica de Roe para resolver las ecuaciones de propagación de las ondas en el agua.

## CAPÍTULO II

### ECUACIONES DE SAINT-VENANT

#### **2.1 Introducción.**

En el cálculo del campo de flujo en los ríos, los estuarios, etc., utilizamos las ecuaciones hidrodinámicas de **Saint-Venant**. Estas ecuaciones expresan los principios de conservación de la masa y de la cantidad de movimiento. Ellas son un caso particular de las ecuaciones tridimensionales de Navier-Stokes.

En el caso de los flujos a superficie libre, estas últimas son integradas sobre la profundidad, con la hipótesis de una distribución de presión sobre la vertical hidrostática lo que implica que las aceleraciones verticales son débiles. El resultado de esta integración es la ecuación célebre de **Saint-Venant** para flujos a superficies libres. Las ecuaciones de **Saint-Venant** son presentadas aquí bajo la forma conservativa. Ellas son de tipo hiperbólico-parabólico y están basadas en las hipótesis siguientes:

#### **2.2 Hipótesis de las ecuaciones de Saint-Venant.**

- El agua es incompresible.
- La pendiente media de la cama del canal es débil.
- Los efectos de resistencia al flujo pueden ser descritos con la ayuda de leyes empíricas utilizadas en los flujos permanentes.

En las páginas siguientes, presentaremos las ecuaciones de Saint-Venant seguidas de una discusión breve de sus variables. Las ecuaciones serán presentadas más tarde bajo la forma matricial.



### 2.3 Ecuaciones Gobernantes.

**Ecuación de Continuidad:**

$$\frac{\partial h}{\partial t} + \frac{\partial uh}{\partial x} + \frac{\partial vh}{\partial y} = 0 \quad (2.1)$$

**Ecuación de cantidad de movimiento en la dirección 'X':**

$$\frac{\partial uh}{\partial t} + \frac{\partial \left( u^2 h + \frac{gh^2}{2} \right)}{\partial x} + \frac{\partial uvh}{\partial y} = gh (S_{ox} - S_{fx}) \quad (2.2)$$

**Ecuación de cantidad de movimiento en la dirección 'Y':**

$$\frac{\partial vh}{\partial t} + \frac{\partial \left( v^2 h + \frac{gh^2}{2} \right)}{\partial y} + \frac{\partial uvh}{\partial x} = gh (S_{oy} - S_{fy}) \quad (2.3)$$

Donde:

"X" y "Y" = coordenadas horizontales.

t = tiempo.

u = u (x, y, t) = velocidad en la dirección 'X'.

v = v (x, y, t) = velocidad en la dirección 'Y'.

h = h (x, y, t) = profundidad instantánea.

g = aceleración gravitacional

Zf (x, y, t) = cota del fondo medida a partir del datum.

$S_{ox}$  y  $S_{oy}$  = Pendientes naturales del canal según la dirección "X" y "Y" respectivamente.

$$S_{ox} = \frac{-\partial z_f}{\partial x} \quad y \quad S_{oy} = \frac{-\partial z_f}{\partial y} \quad (2.4)$$

$S_{fx}$  y  $S_{fy}$  = pendientes de energía debidas al corte sobre el fondo del canal según la dirección "X" y "Y" respectivamente.

Muy frecuentemente, la aproximación siguiente, utilizando la fórmula de Manning es hecha de esta forma:

$$S_{fx} = \frac{n^2 u (u^2 + v^2)^{1/2}}{h^{4/3}} \quad y \quad S_{fy} = \frac{n^2 v (u^2 + v^2)^{1/2}}{h^{4/3}} \quad (2.5)$$

#### 2.4 Las ecuaciones de Saint-Venant en forma matricial.

Para simplificar la representación del esquema numérico de Beam and Warming modificado, las ecuaciones son escritas aquí bajo la forma matricial.

$$U_x + E_x + F_y + S = 0 \quad (2.6)$$

Donde :

$$U = \begin{bmatrix} h \\ uh \\ vh \end{bmatrix} \quad E = \begin{bmatrix} uh \\ u^2 + \frac{1}{2}gh^2 \\ uvh \end{bmatrix} \quad F = \begin{bmatrix} vh \\ uvh \\ v^2 + \frac{1}{2}gh^2 \end{bmatrix}$$
$$S = \begin{bmatrix} 0 \\ gh(S_{ox} - S_{fx}) \\ gh(S_{oy} - S_{fy}) \end{bmatrix} \quad (2.7)$$

Donde "E" y "F" son llamados vectores de flujo, "U" es la matriz de las variables y "S" es el término fuente,  $uh$  y  $vh$  son pues flujos unitarios según  $x$  y  $y$  respectivamente.

## CAPÍTULO III

### MÉTODO NUMÉRICO

#### **3.1 Introducción.**

Las ecuaciones que rigen los fenómenos de flujos a superficie libre han sido presentadas en las ecuaciones (2.1), (2.2) y (2.3). Estas ecuaciones pueden ser resueltas utilizando el método de las características, diferencias finitas, elementos finitos, volúmenes finitos y los métodos espectrales.

El método de las diferencias finitas fue uno de los métodos más a menudo utilizado. Esto es debido a la complejidad de los otros métodos en cuanto a su aplicación en dos dimensiones.

En este capítulo, todos los detalles del esquema de Beam and Warming serán presentados, así como las condiciones fronteras y el criterio de estabilidad.

#### **3.2 Esquema de Beam and Warming.**

Este método ha sido desarrollado para resolver las ecuaciones de Navier-Stokes en los flujos compresibles. El esquema puede ser empleado en la simulación de los problemas de flujo a superficie libre.

La formulación general es presentada, seguida del algoritmo de resolución. Las técnicas del método o alternación en las direcciones de cálculo han sido incorporadas al esquema para estudiar los casos de flujo subcrítico y supercrítico simultáneamente. Del mismo modo, una discusión sobre condiciones fronteras será abordada en el capítulo presente.

### 3.3 Formulación General.

El sistema de ecuaciones (2.1), (2.2) y (2.3) puede ser resuelto de modo general utilizando la aproximación siguiente:

$$\frac{\partial U}{\partial t} \cong \frac{U^{n+1} - U^n}{\Delta t} = \left(\frac{\Delta U}{\Delta t}\right)^{n+1} = \frac{\theta_1}{1+\theta_2} \left(\frac{\partial U}{\partial t}\right)^{n+1} + \frac{1-\theta_1}{1+\theta_2} \left(\frac{\partial U}{\partial t}\right)^n + \frac{\theta_2}{1+\theta_2} \left(\frac{\partial U}{\partial t}\right)^{n-1} \quad (3.1)$$

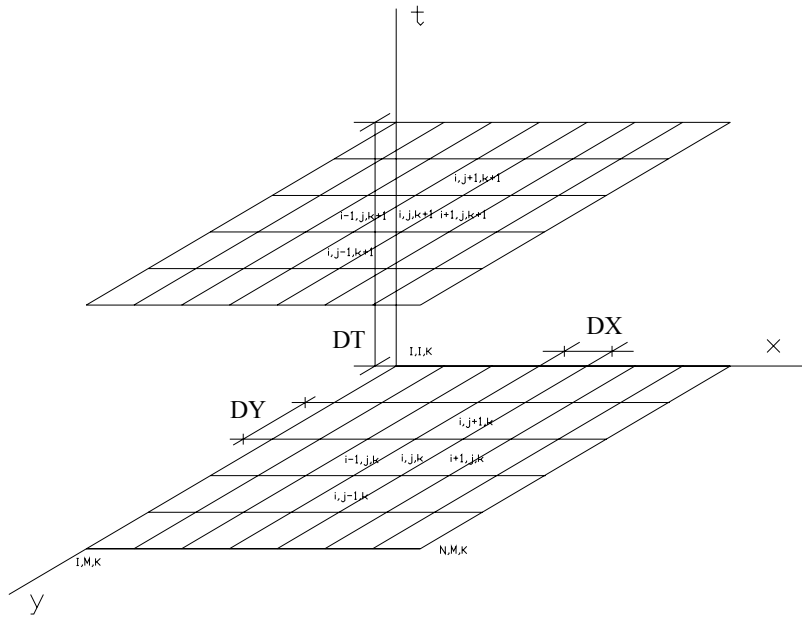
Donde:

$$\Delta U^{n+1} = U^{n+1} - U^n \quad (3.2)$$

La ecuación (3.1) relaciona diferentes formulaciones, en función de la elección de los parámetros  $\theta_1$  y  $\theta_2$ . Por ejemplo, **Richtmyer and Morton (1967)** y **Toune (1990)**, utilizaron los valores que son presentados en la tabla (3.1).

Tabla (3.1) Indicativo de las diferentes formulaciones del esquema Beam and Warming.

$\theta_1$	$\theta_2$	Método	Error de Truncamiento
0	0	Euler Explícito	$O(\Delta t)^2$
0	-1/2	Leapfrog Explícito	$O(\Delta t)^3$
1	0	Euler Implícito	$O(\Delta t)^2$
1/2	0	Trapezoidal Implícito	$O(\Delta t)^3$
1	1/2	3-Point Backward Implicit (3 punto Hacia atrás Implícito)	$O(\Delta t)^3$



**Figura (3.1) Malla de cálculo del esquema Beam and Warming.**

Reemplazando  $\left(\frac{\partial U}{\partial t}\right)$  en la ecuación original en términos de vectores de flujo, variables dependientes y de los términos fuentes, obtenemos:

$$U^{k+1} = U^k - \Delta t \left[ \frac{\theta_1}{1+\theta_2} \left( \frac{\partial E}{\partial X} + \frac{\partial F}{\partial Y} + S \right)^{k+1} + \frac{1-\theta_1}{1+\theta_2} \left( \frac{\partial E}{\partial X} + \frac{\partial F}{\partial Y} + S \right)^k \right] + \Delta t \frac{\theta_2}{1+\theta_2} \left( \frac{\partial U}{\partial t} \right)^{k-1} \quad (3.3)$$

La no linealidad de los términos hace que se encuentren ciertas dificultades con la evolución del nivel del tiempo, es decir, la incrementación de  $\Delta t$ . Estos vectores de flujo pueden ser lineales utilizando la expansión de Taylor para cada nivel del tiempo. Por ejemplo, la expansión de la matriz "E" queda de la siguiente manera:

$$E^{k+1} = E^k + \Delta t \frac{\partial E^k}{\partial t} + \frac{(\Delta t)^2}{2} \frac{\partial^2 E^k}{\partial t^2} + \dots \quad (3.4)$$

Utilizando la regla de derivación en cadena obtenemos:

$$\frac{\partial E^k}{\partial t} = \frac{\partial E^k}{\partial U} \frac{\partial U^k}{\partial t} \quad (3.5)$$

Simplificando, vemos que  $\frac{\partial E^k}{\partial U}$  es igual al Jacobiano "A". De este resultado, podemos escribir que:

$$\frac{\partial E^k}{\partial t} = A^k \frac{\partial U^k}{\partial E} \quad (3.6)$$

Si se toma los dos primeros términos de la expansión de Taylor, que se sustituye en la ecuación (3.3) y escribimos  $\frac{\partial U^k}{\partial E}$ , están en forma de diferencia:

$$E^{k+1} = E^k + A^k (U^{k+1} - U^k) \quad (3.7)$$

Procedemos del mismo modo para las expresiones siguientes:

$$F^{k+1} = F^k + B^k (U^{k+1} - U^k) \quad (3.8)$$

$$S^{k+1} = S^k + Q^k (U^{k+1} - U^k) \quad (3.9)$$

Donde:

"Q" y "B" representan el Jacobiano de los términos fuentes "S" y "F" respectivamente.

Luego, sustituimos (3.7), (3.8) y (3.9) en (3.3), combinamos los términos del mismo nivel del tiempo y aislamos los términos  $U^{k+1}$ .

Obtenemos entonces:

$$\begin{aligned}
 U^{k+1} = & U^k - \Delta t \left[ \frac{\theta_1}{1-\theta_2} \left( \frac{\partial}{\partial x} A^k U^{k+1} + \frac{\partial}{\partial y} B^k U^{k+1} + Q^k U^{k+1} \right) \right. \\
 & - \frac{\theta_1}{1+\theta_2} \left( \frac{\partial}{\partial x} A^k U^k + \frac{\partial}{\partial y} B^k U^k + Q^k U^k \right) + \frac{1}{1+\theta_2} \left( \frac{\partial}{\partial x} A^k U^k + \frac{\partial}{\partial y} B^k U^k + Q^k U^k \right) \\
 & \left. + \frac{1}{1+\theta_2} \left( \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} + S \right)^k + \Delta t \frac{\theta_2}{1+\theta_2} \left( \frac{\partial U}{\partial t} \right)^{k+1} \right] \quad (3.10)
 \end{aligned}$$

Reagrupamos los coeficientes en función de sus posiciones con relación al tiempo, y obtenemos:

$$\begin{aligned}
 \left[ I + \Delta t \frac{\theta_1}{1+\theta_2} \left( \frac{\partial}{\partial x} A^k + \frac{\partial}{\partial y} B^k + Q^k \right) \right] U^{k+1} = & \left[ I + \Delta t \frac{\theta_1}{1+\theta_2} \left( \frac{\partial}{\partial x} A^k + \frac{\partial}{\partial y} B^k + Q^k \right) \right] U^k \\
 - \Delta t \frac{1}{1+\theta_2} \left( \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} + S \right)^k + \Delta t \frac{\theta_2}{1+\theta_2} \left( \frac{\partial U}{\partial t} \right)^{k+1} & \quad (3.11)
 \end{aligned}$$



Donde:

"I" es la matriz identidad.

Si se utiliza la diferencia siguiente  $\Delta_t U^{k+1} = U^{k+1} - U^k$  y se reemplaza en (3.11), obtenemos:

$$\left[ I + \Delta t \frac{\theta_1}{1 + \theta_2} \left( \frac{\partial}{\partial x} A^k + \frac{\partial}{\partial y} B^k + Q^k \right) \right] \Delta_t U^{k+1} = -\Delta t \frac{1}{1 + \theta_2} \left( \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} + S \right)^k + \frac{\theta_2}{1 + \theta_2} \Delta t U^k \quad (3.12)$$

### 3.4 Esquemas Factoriales.

Uno de los problemas que se presenta tratando de resolver la ecuación (3.12), es la generación de las gruesas matrices, lo que es muy difícil a nivel de cálculos. Sin embargo, podemos utilizar una herramienta como la factorización de la ecuación con el fin de precaverse de este problema. En ese caso, una matriz con una pequeña longitud de banda puede ser obtenida reduciendo el problema bidimensional a dos problemas unidimensionales. Sean las ecuaciones siguientes:

$$\left[ I + \Delta t \frac{\theta_1}{1 + \theta_2} \frac{\partial}{\partial x} A^k \right] \left[ I + \Delta t \frac{\theta_1}{1 + \theta_2} \left( \frac{\partial}{\partial x} B + Q \right)^k \right] \Delta_t U^{k+1} = -\Delta t \frac{1}{1 + \theta_2} \left( \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} + S \right)^k + \frac{\theta_2}{1 + \theta_2} \Delta t U^k \quad (3.13)$$

Las ecuaciones en forma factorial se obtienen a través de una serie de pasos, los cuales son directamente dependientes. Estos pasos son:

$$\left[ I + \Delta t \frac{\theta_1}{1 + \theta_2} \frac{\partial}{\partial x} A^k \right] \Delta_t \hat{U} = -\Delta t \frac{1}{1 + \theta_2} \left( \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} + S \right)^k + \frac{\theta_2}{1 + \theta_2} \Delta t U^k$$

$$\left[ I + \Delta t \frac{\theta_1}{1 + \theta_2} \left( \frac{\partial}{\partial x} B + Q \right)^k \right] \Delta_t U^{k+1} = \Delta_t \hat{U}$$

$$U^{k+1} = U^k + \Delta_t U^{k+1} \tag{3.14}$$

Donde:

$\Delta_t \hat{U}$ , es un valor intermedio de la solución según el eje "X".

### 3.5 Descomposición de flujo implícito factorial.

El procedimiento de factorización, descrito anteriormente no tolera la presencia de flujos subcríticos y supercríticos simultáneamente, lo que se ve un inconveniente.

La descomposición del flujo puede ser incorporada al algoritmo precedente, donde las submatrices, son utilizadas para operar correctamente el conjunto de variables con una diferencia espacial propicia.

$$\left[ I + \Delta t \frac{\theta_1}{1 + \theta_2} \frac{\partial}{\partial x} (A^+ + A^-)^k \right] \Delta_t \hat{U} = -\Delta t \frac{1}{1 + \theta_2} \left[ \frac{\partial E^+}{\partial x} + \frac{\partial E^-}{\partial x} + \frac{\partial F^+}{\partial y} + \frac{\partial F^-}{\partial y} + S \right]^k + \frac{\theta_2}{1 + \theta_2} \Delta t U^k \quad (3.15)$$

$$\left[ I + \Delta_t \frac{\theta_1}{1 + \theta_2} \left( \frac{\partial}{\partial y} B^+ + \frac{\partial}{\partial y} B^- + Q \right)^k \right] \Delta_t U^{k+1} = \Delta_t \hat{U} \quad (3.16)$$

$$U^{k+1} = U^k + \Delta_t U^{k+1} \quad (3.17)$$

### 3.6 Diferenciación Espacial.

La eventualidad de tener un límite del número de puntos en el modelo utilizado para la aproximación de las derivadas proviene del ancho de la banda del sistema de ecuaciones que hay que resolver. Además, si las aproximaciones hechas para modelar las derivadas espaciales no son compactas, la resolución de tal sistema se hace más difícil.

Las aproximaciones utilizadas en nuestro caso son:

$$\frac{\partial U^+_{i,j}}{\partial x} = \frac{1}{\Delta x} \nabla_x U_{i,j} \quad (3.18)$$

$$\frac{\partial U^-_{i,j}}{\partial x} = \frac{1}{\Delta x} \Delta_x U_{i,j} \quad (3.19)$$

$$\Delta_x U_{i,j} = U_{i+1,j} - U_{i,j} \quad (3.20)$$

$$\nabla_x U_{i,j} = U_{i,j} - U_{i-1,j} \quad (3.21)$$

Para las derivadas de segundo orden:

$$\frac{\partial^2 U_{i,j}}{\partial x^2} = \frac{1}{\Delta x^2} \delta x^2 U_{i,j} + O(\Delta x^2) \quad (3.22)$$

Donde:

$$\delta x^2 U_{i,j} = U_{i+1,j} - 2U_{i,j} + U_{i-1,j} \quad (3.23)$$

Para los términos:

$$\frac{\partial \left[ (A^n_{i,j})^+ \Delta t U_{i,j}^n \right]}{\partial x} = \frac{(A^n_{i,j})^+ \Delta t U^n_{i,j} - (A^n_{i-1,j})^+ \Delta t U^n_{i-1,j}}{\Delta x} \quad (3.24)$$

$$\frac{\partial \left[ (A^n_{i,j})^- \Delta t U_{i,j}^n \right]}{\partial x} = \frac{(A^n_{i+1,j})^- \Delta t U^n_{i+1,j} - (A^n_i)^- \Delta t U^n_i}{\Delta x} \quad (3.25)$$

$$\left[ \frac{\partial F^n_{i,j}}{\partial x} \right]^+ = \frac{(A^n_{i,j})^+ \nabla U^n_{i,j}}{\Delta x} = \frac{(A^n_{i,j})^+ (U^n_{i,j} - U^n_{i-1,j})}{\Delta x} \quad (3.26)$$

$$\left[ \frac{\partial F^n_{i,j}}{\partial x} \right]^- = \frac{(A^n_{i,j})^- \Delta x U^n_{i,j}}{\Delta x} = \frac{(A^n_{i,j})^- (U^n_{i+1,j} - U^n_{i,j})}{\Delta x} \quad (3.27)$$

Después de la aplicación de las aproximaciones hechas anteriormente en las ecuaciones (3.15), (3.16) y (3.17), obtenemos:

$$\begin{aligned} & \left[ I + \Delta t \frac{\theta_1}{1+\theta_2} (\nabla_x A^+_{i,j} + \nabla_x A^-_{i,j})^k \right] \Delta t \hat{U} = \\ & - \frac{\Delta t}{\Delta x} \frac{1}{1+\theta_2} (A^+_{i,j} \nabla_x U_{i,j} + A^-_{i,j} \Delta_x U_{i,j})^k \\ & - \frac{\Delta t}{\Delta y} \frac{1}{1+\theta_2} (B^+_{i,j} \nabla_y U_{i,j} + B^-_{i,j} \Delta_y U_{i,j})^k \\ & - \Delta t \frac{1}{1+\theta_2} S^k_{i,j} + \frac{\theta_2}{1+\theta_2} \Delta t U^k_{i,j} \end{aligned} \quad (3.28)$$

$$\left[ I + \Delta t \frac{\theta_1}{1+\theta_2} \left( \frac{1}{\Delta y} (\nabla_y B^+_{i,j} + \nabla_y B^-_{i,j}) + Q_{i,j} \right)^k \right] \Delta t U^{k+1} = \Delta t \hat{U} \quad (3.29)$$

$$U^{k+1} = U^k + \Delta t U^{k-1} \quad (3.30)$$

Donde A y B son las matrices Jacobianas utilizadas en el esquema de Beam and Warming. Estas son presentadas más abajo:

$$A = \frac{\partial F}{\partial U} = \begin{bmatrix} 0 & 1 & 0 \\ -u^2 + c^2 & 2u & 0 \\ -uv & v & u \end{bmatrix} \quad (3.31)$$

Donde:

$$\lambda_1 = u, \quad \lambda_2 = u + c, \quad \lambda_3 = u - c$$

$$A^* = \begin{bmatrix} \lambda_2^*(c-u) + \lambda_3^*(u+c) & \frac{\lambda_2 - \lambda_3}{2c} & 0 \\ \frac{(\lambda_2^* - \lambda_3^*)(c^2 - u^2)}{2c} & \lambda_2^*(u+c) - \lambda_3^*(u-c) & 0 \\ \frac{v(\lambda_2^*(c-u) + \lambda_3^*(u+c))}{2c} - \lambda_1^*v & \frac{v(\lambda_2^* - \lambda_3^*)}{2c} & \lambda_1^* \end{bmatrix} \quad (3.32)$$

$$B = \frac{\partial G}{\partial U} = \begin{bmatrix} 0 & 0 & 1 \\ -uv & v & u \\ -v^2 + c^2 & 0 & 2v \end{bmatrix} \quad (3.33)$$

Donde:

$$\lambda_1 = v, \quad \lambda_2 = v + c, \quad \lambda_3 = u - c$$

$$B^* = \begin{bmatrix} \frac{\lambda_2^*(c-v) + \lambda_3^*(v+c)}{2c} & 0 & \frac{\lambda_2^* - \lambda_3^*}{2c} \\ u \left[ \frac{\lambda_2^*(c-v) + \lambda_3^*(v+c)}{2c} \right] - \lambda_1^* u & \lambda_1^* & \frac{u(\lambda_2^* - \lambda_3^*)}{2c} \\ \frac{(\lambda_3^* - \lambda_2^*)(v^2 - c^2)}{2c} & 0 & \frac{[\lambda_2^*(v+c) - \lambda_3^*(v-c)]}{2c} \end{bmatrix} \quad (3.34)$$

Donde:

$$\lambda_1^+ = \frac{\lambda_1 + |\lambda_1|}{2} \quad \lambda_1^- = \frac{\lambda_1 - |\lambda_1|}{2} \quad * = -, + \quad (3.35)$$

Para resolver el sistema de ecuaciones (3.28), (3.29) y (3.30), sugerimos la construcción de un sistema tridiagonal por bloques, de la manera siguiente:

$$\begin{bmatrix} Bl_1 & Cl_1 & 0 \\ Al_i & Bl_i & Cl_i \\ 0 & Al_i & Bl_i & Cl_i \\ & \ddots & \ddots & \ddots \\ & & Al_{n-1} & Bl_{n-1} & Cl_{n-1} \\ & & & Al_n & Bl_n \end{bmatrix} \begin{bmatrix} \Delta_i \hat{U}_1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \Delta_i \hat{U}_n \end{bmatrix} = \begin{bmatrix} RHS_1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ RHS_n \end{bmatrix} \quad (3.36)$$

Donde:

Al, Bl, Cl son las matrices de 3x3 y  $\Delta_t U$  son las variables de flujo ( $\Delta h$ ,  $\Delta uh$ ,  $\Delta vh$ ). Después de haber obtenido la solución en la dirección 'X', el paso siguiente consiste en resolver, de una manera similar, un sistema tridiagonal por bloques en la dirección 'Y'.

Una vez obtenida la solución en la dirección 'Y', rescribimos las variables de flujo de la manera siguiente:

$$\begin{aligned}
 h_{i,j}^{k+1} &= h_{i,j}^k + \Delta_t h_{i,j}^{k+1} \\
 u_{i,j}^{k+1} &= \frac{(uh)_{i,j}^k + \Delta_t (uh)_{i,j}^{k+1}}{h_{i,j}^{k+1}} \\
 v_{i,j}^{k+1} &= \frac{(vh)_{i,j}^k + \Delta_t (vh)_{i,j}^{k+1}}{h_{i,j}^{k+1}}
 \end{aligned}
 \tag{3.37}$$

Los coeficientes del sistema tridiagonal de una malla de "n" puntos en la dirección "X", son:

$$BI_i = Z\theta_1 [A_{i,j}^+ - A_{i,j}^-] + I \quad i = 2 \dots n - 1 \tag{3.38}$$

$$AI_i = -Z\theta_1 (A_{i,j-1})^+ \quad i = 2 \dots n \tag{3.39}$$

$$CI_i = Z\theta_1 (A_{i+1,j})^- \quad i = 1 \dots n - 1 \tag{3.40}$$



$$RHS_i = -Z \left[ A_{i,j}^n \nabla_x U_{i,j}^n + A_{i,j}^{n-} \Delta x U_{i,j}^n \right] - Z \Delta x \left( S_{i,j}^n - \theta_1 \Delta x \Delta t U_{i,j}^{n-1} \right)$$

$$i = 2 \dots n - 1 \quad (3.41)$$

Y según la dirección "Y":

$$BI_j = \left[ I + Z \theta_1 (B_{i,j}^+ - B_{i,j}^-) + \Delta y Z Q_{i,j} \right] \quad j = 2 \dots m - 1 \quad (3.42)$$

$$AI_j = \theta_1 Z (-B_{i,j-1}^+) \quad j = 2 \dots m \quad (3.43)$$

$$CI_j = \theta_1 Z (B_{i,j+1}) \quad j = 1 \dots m - 1 \quad (3.44)$$

$$RHS = \Delta_t \hat{U}_{i,j} \quad j = 1 \dots m - 1 \quad (3.45)$$

$$Z = \frac{\Delta t}{\Delta x} \left( \frac{1}{1 + \theta_2} \right) \quad (3.46)$$

Donde:

$Q_{i,j}$  es el Jacobiano de S.

### 3.7 Condiciones Iniciales.

Las condiciones a  $t = 0$  de todas las variables (u, v, h) deben ser especificadas en todos los puntos del dominio:

$$\begin{aligned}
u(x, y, 0) &= u_0 \\
v(x, y, 0) &= v_0 \\
h(x, y, 0) &= h_0
\end{aligned}
\tag{3.47}$$

Estos valores iniciales son suficientes para que la solución converja, gracias al carácter hiperbólico del sistema de ecuaciones.

En ciertos problemas,  $u_0 = 0$  y  $v_0 = 0$  pueden ser especificadas al mismo tiempo que  $h_0$  tiende hacia cero '0' con arreglo al problema que hay que estudiar.

### **3.8 Implantación de las condiciones fronteras.**

Tres condiciones son necesarias para cada punto de la frontera que puede estar abierta o cerrada.

#### **3.8.1 Para una frontera cerrada:**

No hay ninguna velocidad normal para la pared, y al mismo tiempo,  $\Delta t (u_h)$  o  $\Delta t (v_h)$  pueden ser algún arreglo en función de su orientación. También, se utilizará el costado positivo de "A" o "B" frente al eje "X" o "Y" positivo, y el costado negativo de "A" o "B" frente al eje negativo de "X" o "Y".

#### **3.8.2 Para una frontera abierta:**

Si se puede conocer por anticipado el comportamiento del flujo, sugerimos:

### 3.8.2.1 Flujo Supercrítico:

Aguas arriba, la velocidad longitudinal  $u$ , la velocidad transversal  $v$ , y la profundidad  $h$  deben ser especificadas y no varían durante las iteraciones. Aguas abajo, todas las variables,  $u$ ,  $v$  y  $h$ , son extrapoladas a partir de los puntos interiores del dominio.

### 3.8.2.2 Flujo Subcrítico:

Aguas arriba, la velocidad longitudinal  $u$  y la velocidad transversal  $v$  son especificadas. La profundidad  $h$  es extrapolada a partir de los puntos interiores del dominio. Aguas abajo, la profundidad  $h$  es especificada y las velocidades  $u$  y  $v$  son extrapoladas a partir de los puntos interiores del dominio.

## 3.9 Criterio de Estabilidad.

Como otro esquema numérico, el esquema de **Beam and Warming** debe satisfacer las condiciones de estabilidad de Courant-Friedrichs-Lewy (C.F.L). En este estudio, dos métodos son utilizados para escoger el paso de tiempo ( $\Delta t$ ), para que el esquema sea estable. Un método depende de un ( $\Delta t$ ) global y el otro de un ( $\Delta t$ ) local.

El ( $\Delta t$ ) global se calcula sobre el dominio de modo siguiente:

$$\Delta t = \text{MIN}(\Delta t_c)_{i,j} \quad (3.48)$$

Donde:

$$\Delta t_c = \left[ \frac{|u| + c}{\Delta x} + \frac{|v| + c}{\Delta y} \right]^{-1} \quad (3.49)$$

El  $(\Delta t)$  local se calcula en cada punto del dominio de modo siguiente:

$$\Delta t_{i,j} = \text{MIN} \left[ (\Delta t_{cx}, \Delta t_{cy}) \right]_{i,j} \quad (3.50)$$

$$\Delta t_{cx} = \left[ \frac{|u|}{\Delta x} + \frac{c}{\Delta x} \right]^{-1}$$

$$\Delta t_{cy} = \left[ \frac{|v|}{\Delta y} + \frac{c}{\Delta y} \right]^{-1}$$

Donde  $c$  es la celeridad  $\sqrt{gh}$

### 3.10 Modificación Sugerida para las Ecuaciones.

En las ecuaciones de Euler utilizadas en la dinámica de los gases, el vector de flujo tiene una propiedad particular, dada por la expresión siguiente:  $Ex = (AU)_x$  y  $Fy = (BU)_x = AU_x$ . Esta identidad existe porque las ecuaciones son unas funciones homogéneas de primer orden, y porque la forma del esquema numérico de estas ecuaciones son conservativas cuando los vectores se descomponen. Esta propiedad no es válida para las ecuaciones de flujo a superficie libre (Ecuaciones de Saint-Venant), las cuales son unas funciones homogéneas de segundo orden (sin

término fuente). Por esta razón la forma conservativa de "E" y "F" no es conocida, y ella ha sido evaluada como:  $Ex = AUx$  y  $Fy = BUy$ . Resulta de eso que el lado derecho de la ecuación (3.15) es no-conservativa.

$$\frac{\partial E}{\partial x} = A \frac{\partial U}{\partial x} = A^+ \frac{\partial U}{\partial x} + A^- \frac{\partial U}{\partial x} \quad (3.51)$$

De este modo, la evaluación no-conservativa de la derivada espacial de **E** y **F**, en una formulación conservativa, tiene como consecuencia que el esquema se ve significativamente afectado. Esto introduce errores en la solución.

La modificación sugerida se basa en el concepto del Jacobiano aproximativo que ha sido propuesto por Roe (1981). Según la aproximación de Roe, el Jacobiano aproximativo se construye para cada punto adyacente de modo siguiente:

$$\frac{\partial E}{\partial x} = \hat{A} \frac{\partial U}{\partial x} = \hat{A}_{i-1/2}^+ \frac{\partial U}{\partial x} + \hat{A}_{i+1/2}^- \frac{\partial U}{\partial x} \quad (3.52)$$

Ahora el problema se reduce a definir el Jacobiano aproximativo de "**A**" y "**B**". Roe (1981), deriva las velocidades y profundidades a  $i + \frac{1}{2}$  y  $i - \frac{1}{2}$  así como sigue:

$$\begin{aligned}
u_{i-1/2} &= \frac{u_i \sqrt{h_i} + u_{i-1} \sqrt{h_{i-1}}}{\sqrt{h_i} + \sqrt{h_{i-1}}} \\
u_{i+1/2} &= \frac{u_{i+1} \sqrt{h_{i+1}} + u_i \sqrt{h_i}}{\sqrt{h_{i+1}} + \sqrt{h_i}} \\
v_{i-1/2} &= \frac{v_i \sqrt{h_i} + v_{i-1} \sqrt{h_{i-1}}}{\sqrt{h_i} + \sqrt{h_{i-1}}} \\
v_{i+1/2} &= \frac{v_{i+1} \sqrt{h_{i+1}} + v_i \sqrt{h_i}}{\sqrt{h_{i+1}} + \sqrt{h_i}}
\end{aligned} \tag{3.53}$$

$$h_{i\pm 1/2} = \sqrt{h_i h_{i\pm 1}}$$

$$c_{i\pm 1/2} = \sqrt{\frac{g(h_i + h_{i\pm 1})}{2}}$$

Las matrices Jacobianas "A" y "B", son evaluadas en el estado medio de Roe. Por ejemplo, la matriz "A" se hace:

$$\hat{A}^* = \begin{bmatrix} \lambda_2^*(\hat{c} - \hat{u}) + \lambda_3^*(\hat{u} + \hat{c}) & \frac{\lambda_2 - \lambda_3}{2\hat{c}} & 0 \\ \frac{(\lambda_2^* - \lambda_3^*)(\hat{c}^2 - \hat{u}^2)}{2\hat{c}} & \lambda_2^*(\hat{u} + \hat{c}) - \lambda_3^*(\hat{u} - \hat{c}) & 0 \\ \frac{\hat{v}(\lambda_2^*(\hat{c} - \hat{u}) + \lambda_3^*(\hat{u} + \hat{c}))}{2\hat{c}} - \lambda_1^* \hat{v} & \frac{\hat{v}(\lambda_2^* - \lambda_3^*)}{2\hat{c}} & \lambda_1^* \end{bmatrix} \tag{3.54}$$

Donde:

$$\lambda_1 = \hat{u} \quad \lambda_2 = \hat{u} + \hat{c} \quad \lambda_3 = \hat{u} - \hat{c}$$

$$\lambda_1^+ = \frac{\lambda_1 + |\lambda_1|}{2} \quad \lambda_1^- = \frac{\lambda_1 - |\lambda_1|}{2} \quad (3.55)$$

El símbolo (^) sobre las variables indica que éstas son evaluadas en el estado medio de Roe (Roe Average). Para la matriz  $A^-$ , sus coeficientes son evaluados a  $\mathbf{i} + \frac{1}{2}$  y para la matriz  $A^+$  sus coeficientes son evaluados a  $\mathbf{i} - \frac{1}{2}$  según las ecuaciones (3.51).

Procedemos del mismo modo para las matrices "B".

El término no homogéneo  $S' = \begin{bmatrix} 0 \\ gh(S_{ox} - S_{fx}) \\ gh(S_{oy} - S_{fy}) \end{bmatrix}$  es tratado de modo siguiente:

Primero, expresamos 'S' como  $fx + fy$ , donde:

$$fx = \begin{bmatrix} 0 \\ gh(S_{ox} - S_{fx}) \\ 0 \end{bmatrix} \quad fy = \begin{bmatrix} 0 \\ 0 \\ gh(S_{oy} - S_{fy}) \end{bmatrix} \quad (3.56)$$

En segundo lugar, tratamos el término  $fx$ , para el cual Glaister (1989) demuestra, que puede ser aproximado de la siguiente manera:

fx como:

$$fx = -\frac{1}{2\Delta x} \sum_{i=1}^3 \lambda_i \beta_i e_i \quad (3.57)$$

en el cual:

$$\beta = \pm \frac{gh(\Delta Z - 2\Delta x S_{fx})}{2c(u \pm c)} \quad \beta_1 = 0 \quad (3.58)$$

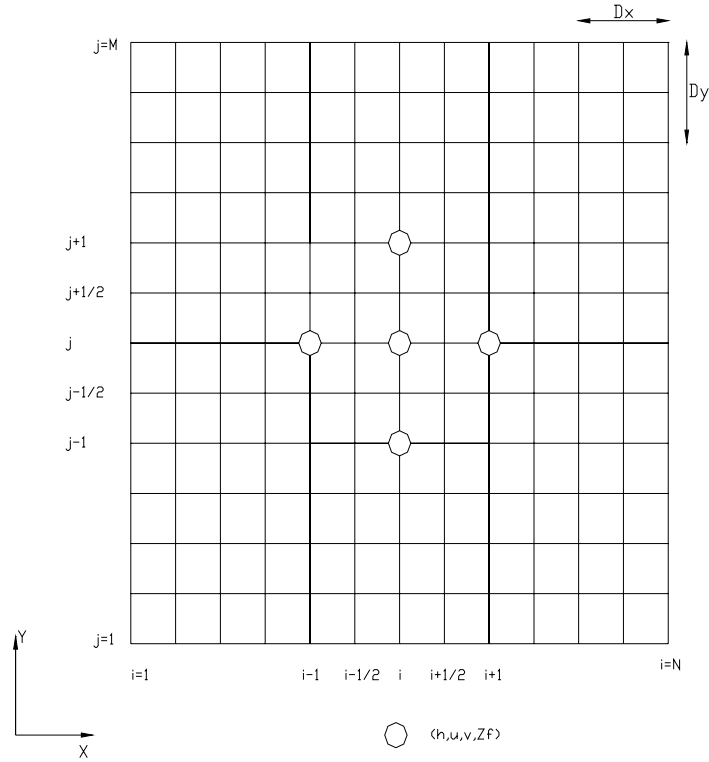
$$\begin{aligned} \lambda_1 &= u & e_1 &= (0,0,1)^t \\ \lambda_2 &= u + c & e_2 &= (1, u + c, v)^t \\ \lambda_3 &= u - c & e_3 &= (1, u - c, v)^t \end{aligned} \quad (3.59)$$

$$\Delta Z = Z_{i+1} - Z_{i-1} \quad (3.60)$$

Donde  $\lambda$  y  $e$  son respectivamente los valores propios y vectores propios, y  $Z$  es la elevación del fondo.

Procedemos de manera similar para el término "Fy".





**Figura (3.2) Malla del cálculo del esquema Beam and Warming modificado.**

Las ecuaciones serán evaluadas como sigue:

$$\begin{aligned}
 & \left[ I + \Delta_t \frac{\theta_1}{1 + \theta_2} \frac{\partial}{\partial X} (A_{i-1/2,j}^+ + A_{i+1/2,j}^-)^k \right] \Delta_t \hat{U} = \\
 & - \frac{\Delta t}{\Delta x} \frac{1}{1 + \theta_2} (A_{i-1/2,j}^+ \nabla_x U_{i,j} + A_{i+1/2,j}^- \Delta_x U_{i,j})^k \\
 & - \frac{\Delta t}{\Delta x} \frac{1}{1 + \theta_2} (B_{i,j-1/2}^+ \nabla_y U_{i,j} + B_{i,j+1/2}^- \Delta_y U_{i,j})^k \\
 & - \Delta t \frac{1}{1 + \theta_2} S_{i,j}^k + \frac{\theta_2}{1 + \theta_2} \Delta_t U_{i,j}^k
 \end{aligned} \tag{3.61}$$

$$\left[ I + \Delta_t \frac{\theta_1}{1 + \theta_2} \left( \nabla_y B_{i,j-1/2}^+ + \Delta_y B_{i,j+1/2}^- + Q_{i,j} \right)^k \right] \Delta_t U^{k+1} = \Delta_t \hat{U} \quad (3.62)$$

$$U^{k+1} = U^k + \Delta_t U^{k+1} \quad (3.63)$$

Serán resueltas según la ecuación (3.36).

## CAPÍTULO IV

### RESULTADOS Y DISCUSIÓN

#### **4.1 Introducción.**

Con el fin de evaluar el comportamiento general del modelo matemático basado en el esquema de Beam and Warming modificado, varias pruebas bidimensionales han sido efectuadas. Para hacerlas, existen soluciones analíticas que serán empleadas con el fin de verificar las características fundamentales del esquema numérico. Un ejemplo de estas soluciones analíticas es el criterio de estabilidad.

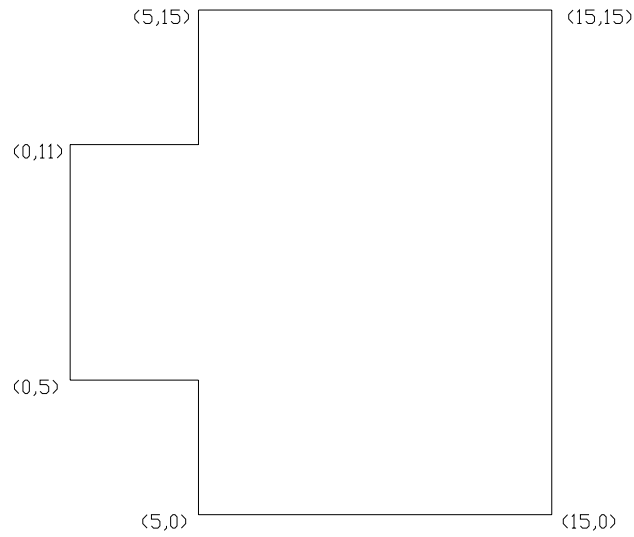
Problemas complejos en dos dimensiones serán presentados con el fin de mostrar el funcionamiento del modelo y el comportamiento de éste frente a la variación de diferentes parámetros presentes en la solución.

Finalmente, un problema general en dos dimensiones con batimetría variable y fronteras irregulares, confirma la adaptabilidad del modelo matemático para aplicaciones industriales.

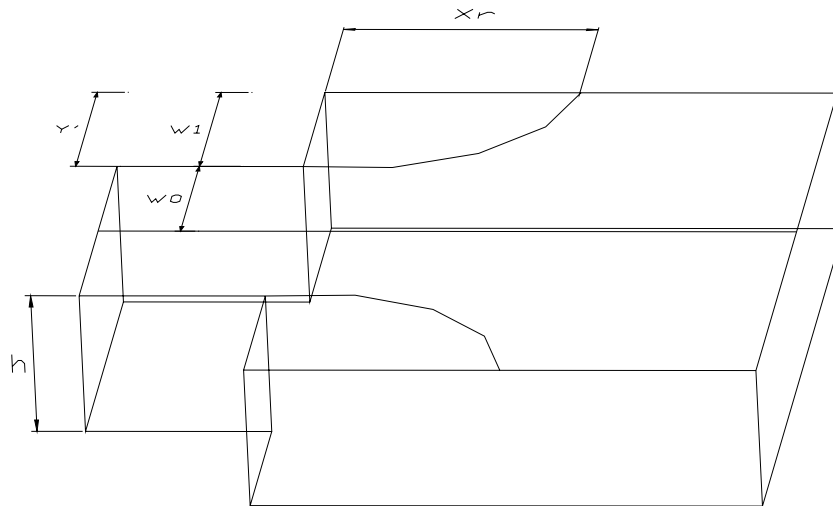
#### **4.2 Ensayos Bidimensionales.**

##### **4.2.1 Canal con expansión brusca.**

Una de las pruebas bidimensionales para evaluar nuestro modelo, es el estudio del caso de un canal rectangular en el cual existe una expansión brusca, tal como está ilustrado en la figura (4.1). En ese caso, analizamos la longitud de vorticidad.



**Figura (4.1) Dominio del cálculo.**



**Figura (4.2) Descripción de las variables.**

Para esto empleamos, para fines de comparación las experiencias realizadas por Moss y Al (1977) que efectuaron una serie de medidas en un canal rectangular con expansión brusca. Sus resultados son presentados bajo forma adimensional en la

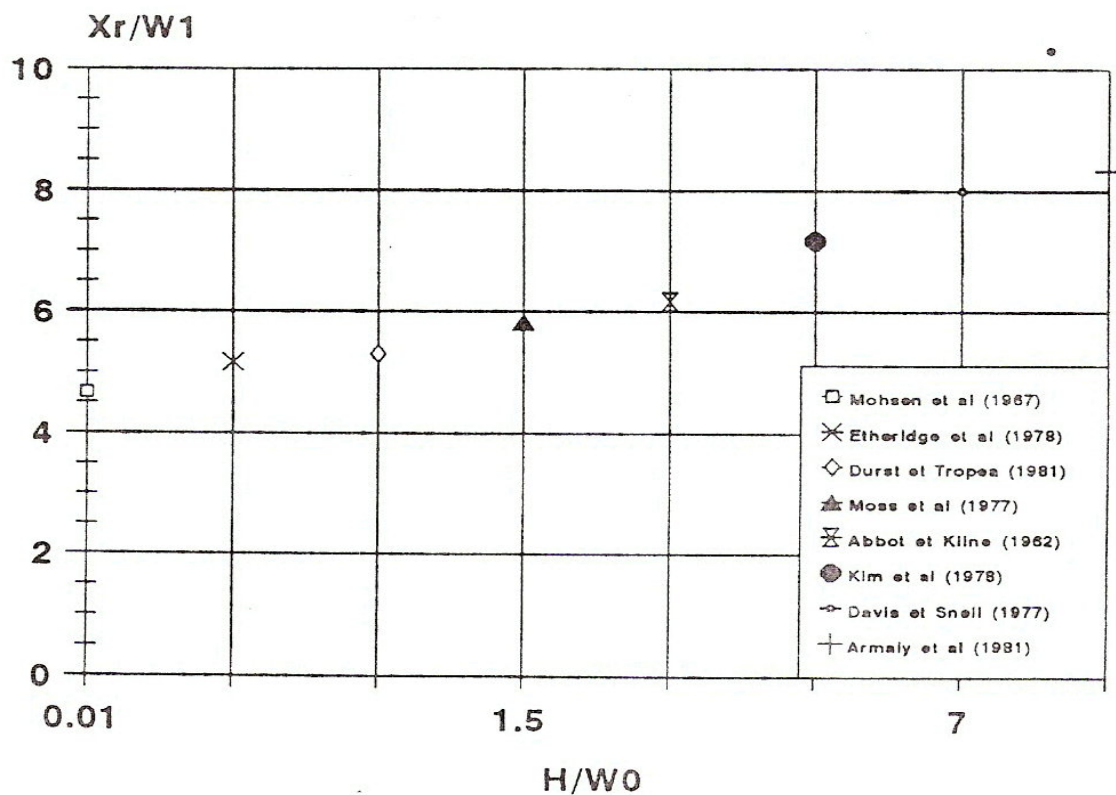
figura (4.3). En estos resultados intervienen la longitud de la vorticidad, medida como lo indica la figura (4.2), y las relaciones siguientes:  $\frac{X_r}{W_1}$  versus  $\frac{h}{W_0}$ . Las velocidades serán medidas en tres secciones transversales a lo largo del eje longitudinal y localizado con la relación  $X_r$ , de manera siguiente:

Sección 1:  $(X - X_r) / W_1 = - 2.6$

Sección 2:  $(X - X_r) / W_1 = - 0.6$

Sección 3:  $(X - X_r) / W_1 = 1.4$

La figura (4.3) reagrupa los resultados experimentales para un valor de  $X_r$  con arreglo a las diferentes alturas, tales como los presentados por Chapman y Kuo (1985).



### Figura (4.3) Experiencias de los diferentes investigadores.

Ellos establecieron la relación  $\frac{u}{u_0}$  versus  $\frac{y}{W_1}$  según el gráfico (4.2) donde:

$$U^* = \frac{U}{U_0} \quad y \quad Y^* = \frac{Y}{W_1} \quad (4.1)$$

$u_0$  es la velocidad de entrada al canal y "Y" es la distancia medida a partir de la pared, tal como está señalada en la figura (4.2).

El dominio del cálculo es mostrado en la figura (4.1). Donde las velocidades (u, v) aplican. La profundidad  $h = 1$  m y el flujo  $Q = 0.6$  m<sup>3</sup>/s. El coeficiente de Manning "n" es igual a 0.015 y el coeficiente de forma  $1 + \frac{W_1}{W_0}$  es igual a 1.45.

Los valores de  $\frac{X_r}{W_1}$  obtenidos son 4.8, y los valores experimentales se sitúan entre 4.5 y 5, figura (4.3). En la figura (4.7) (sección 1) podemos observar que, en la región situada cerca del punto de vorticidad, los resultados obtenidos son iguales a la solución experimental. Aguas arriba y aguas abajo del punto de vorticidad (sección 2 y sección 3), obtenemos velocidades inferiores a las obtenidas experimentalmente.

**Profundidad**

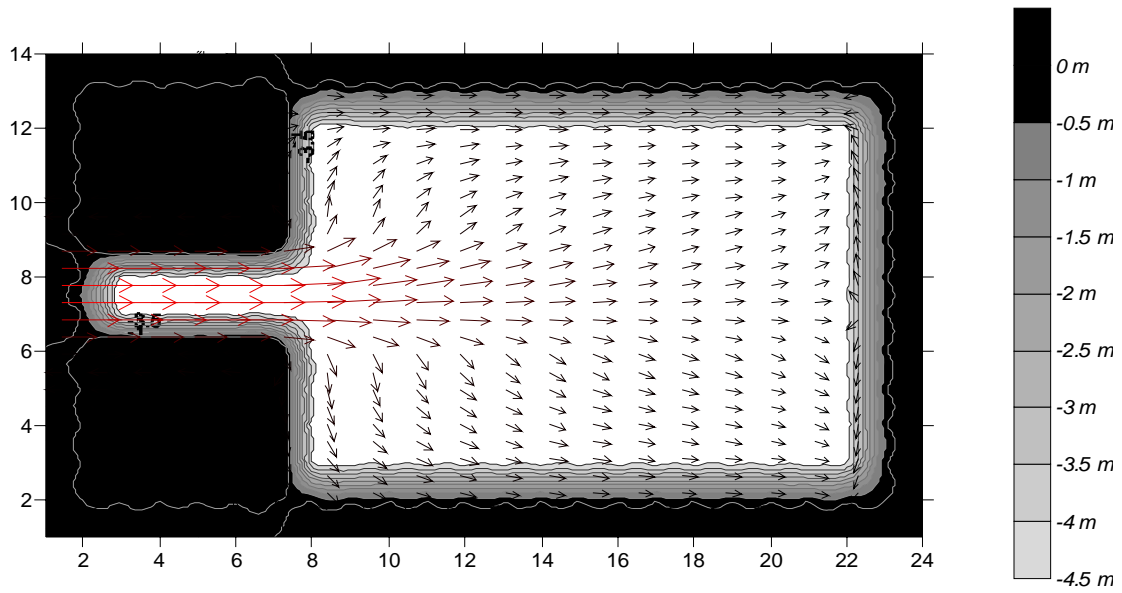


Figura (4.4) Vectores velocidades para ( $t = 500 \text{ s}$ ,  $\Delta x = 100$ ,  $\Delta y = 100$ ,  $Q = 1 \text{ m}^3 / \text{s}$  y  $\Delta t = 10 \text{ s}$ ).

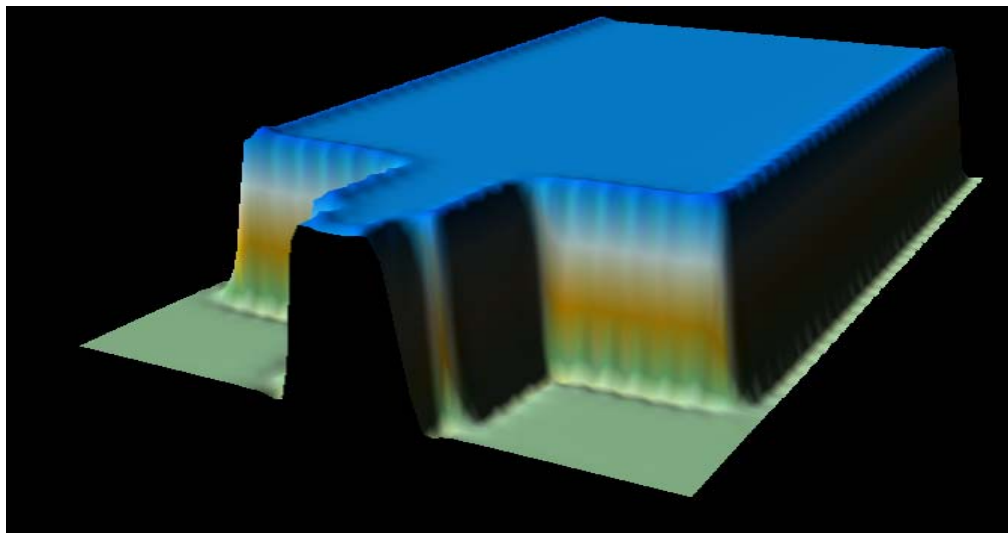


Figura (4.5) Perfil tridimensional de la expansión brusca para ( $Z = 3 \text{ m}$ ,  $D = 2 \text{ m}$ ).

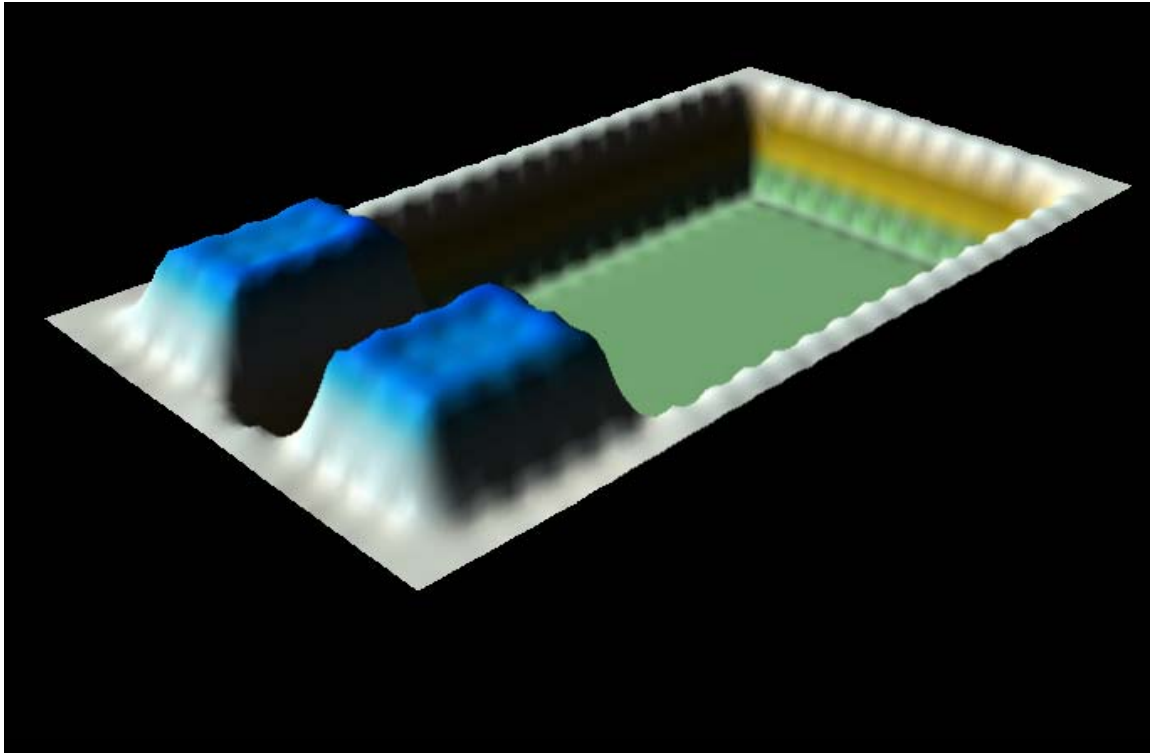
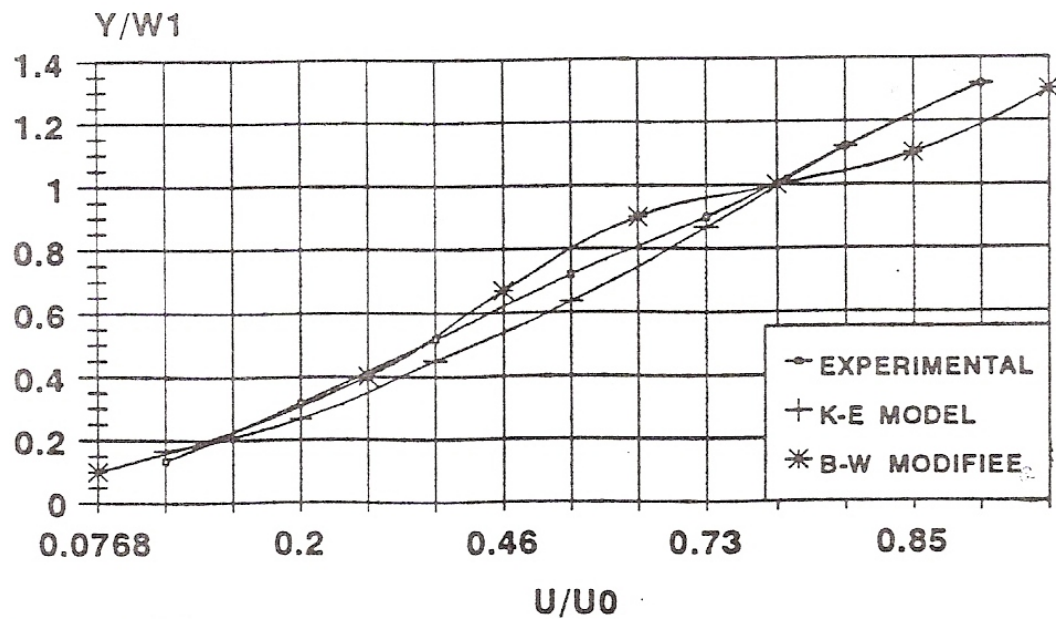
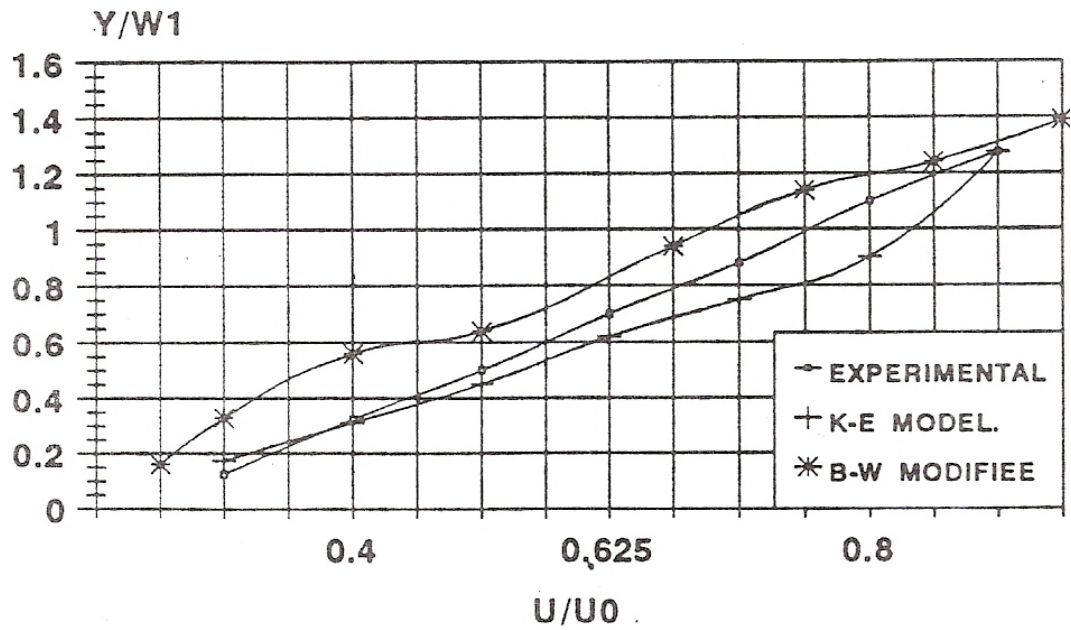


Figura (4.6) Perfil tridimensional del fondo (expansión brusca).

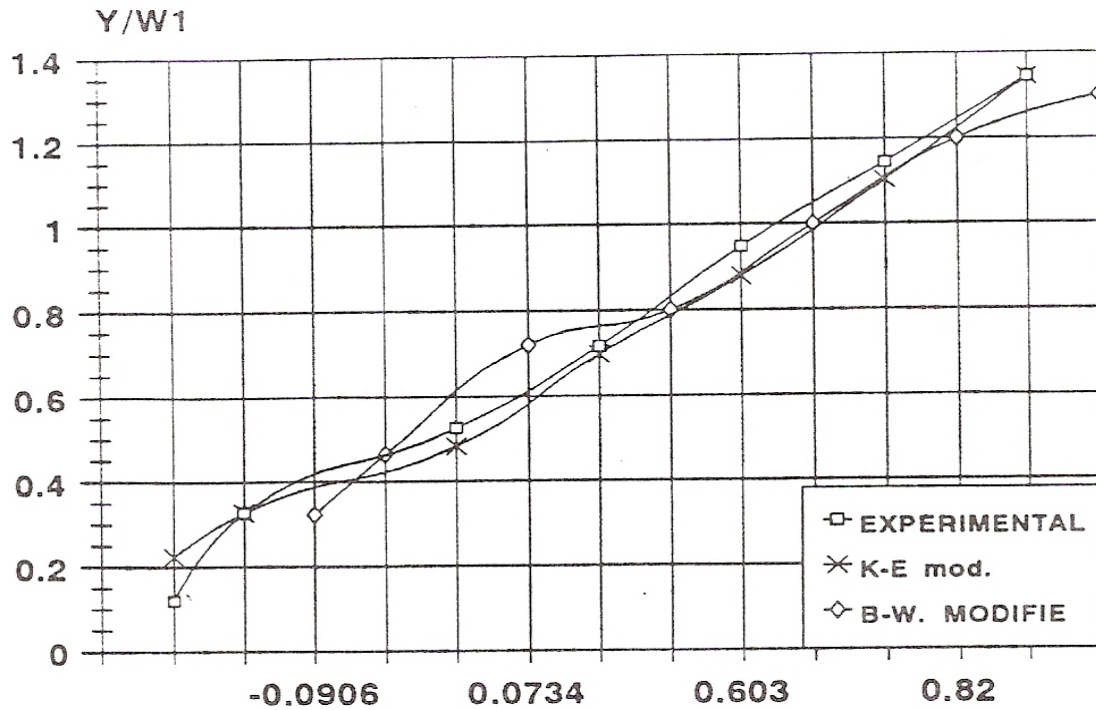




**Figura (4.7) Curva  $U^*$  en función de  $Y^*$  para  $(X - X_r) / W_1 = - 0.6$   
(sección 1)**



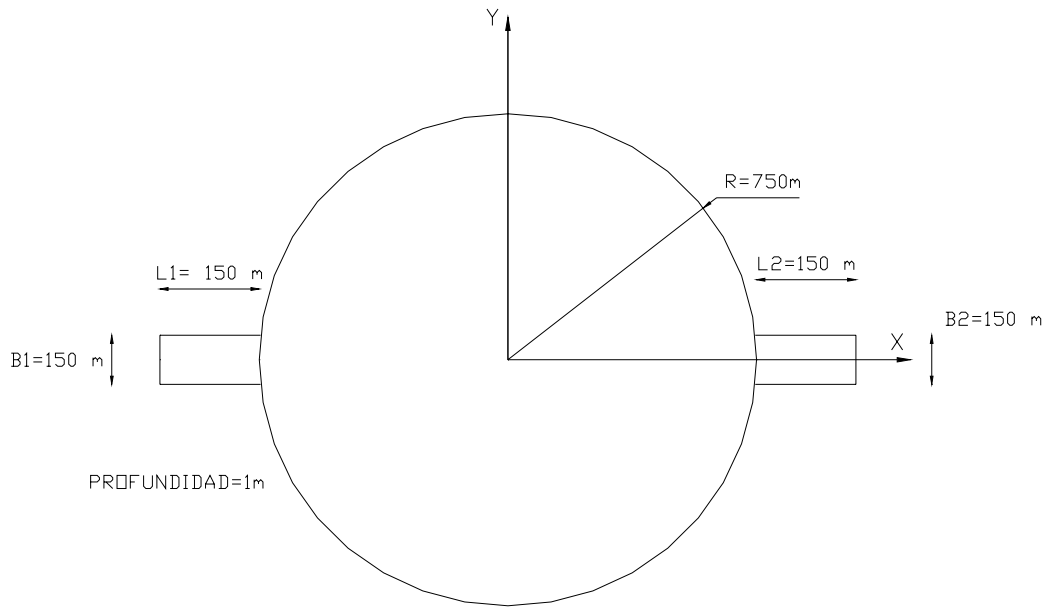
**Figura (4.8) Curva  $U^*$  en función de  $Y^*$  para  $(X - X_r) / W_1 = 1.4$   
(sección 3)**



**Figura (4.9) Curva  $U^*$  en función de  $Y^*$  para  $(X - X_r) / W_1 = - 2.6$   
(sección 2)**

#### 4.2.2 Depósito Circular.

Una de las pruebas de validación utilizada, es la de un depósito circular para el cual se empleó una geometría en una escala ampliada (1000:1) a la utilizada por Denis (1974), tal como está ilustrado en la figura (4.9). En este depósito, la entrada y la salida del flujo, son opuestas diametralmente. Para el modelo de Denis, la longitud de la entrada y de la salida es igual a 0.156 m y la longitud de cada canal es de 0.3 m. El radio del depósito es de 0.75 m y la velocidad de entrada es de 0.1 m/s.



**Figura (4.10) Dominio del cálculo (depósito circular).**

La malla utilizada en ese caso es de  $120 \times 100$  en el depósito circular,  $\Delta t = 3\text{ s}$  y el tiempo de simulación es de  $150\text{ s}$ . La figura (4.10) muestra que el flujo es caracterizado por dos zonas de recirculación establecidas en cada costado del eje del "X".

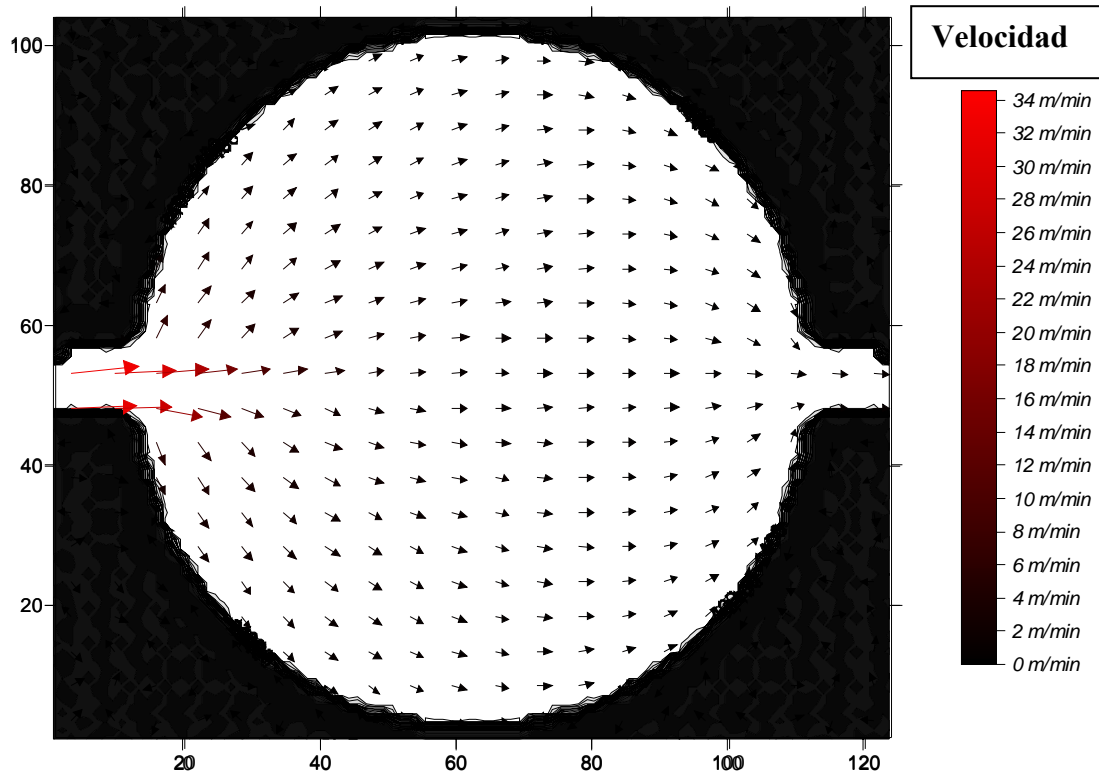
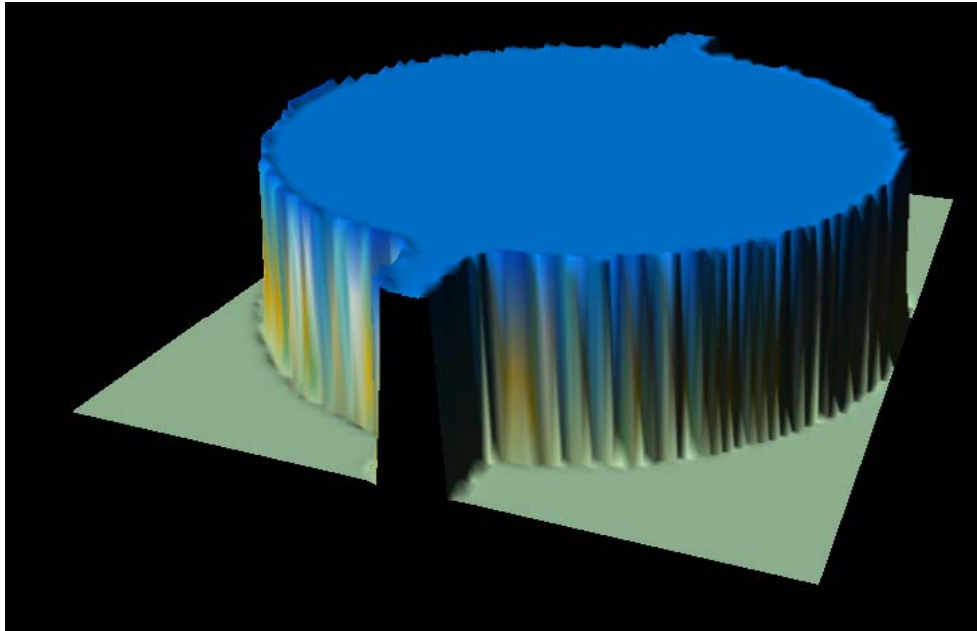
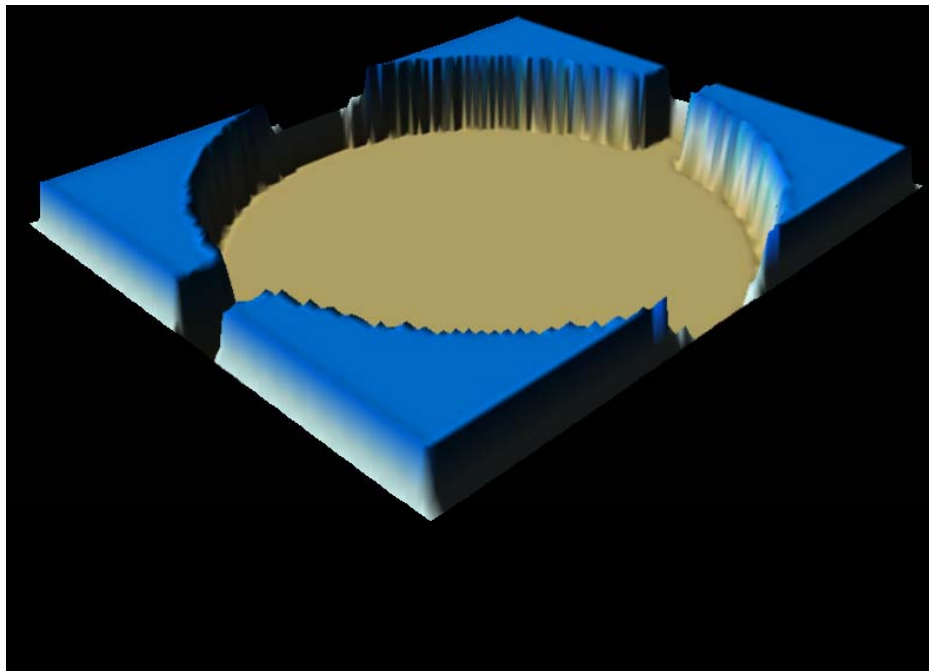


Figura (4.11) Vectores velocidades para ( $t = 150 s$ ,  $\Delta x = 15$ ,  $\Delta y = 15$ ,  $Q = 500 \text{ m}^3 / \text{min}$ . y  $\Delta t = 3 s$ ).



**Figura (4.12) Perfil tridimensional del depósito circular para ( $Z = 15$  m,  $D = 1$  m).**



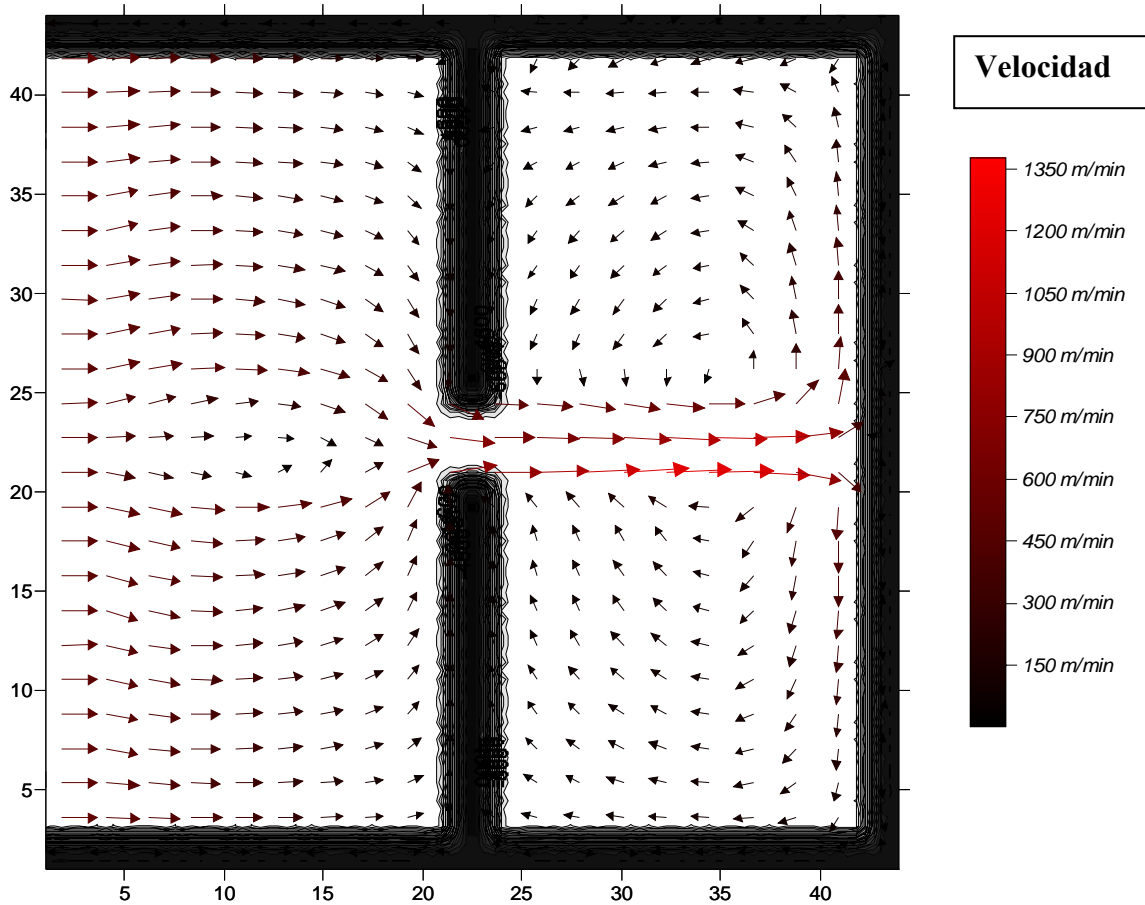
**Figura (4.13) Perfil tridimensional del fondo (depósito circular).**

Si se compara con la prueba de Denis, los torbellinos ocupan una posición similar, pero la zona de recirculación encontrada es un poco más débil en nuestro caso.

#### **4.2.3 Vorticidad.**

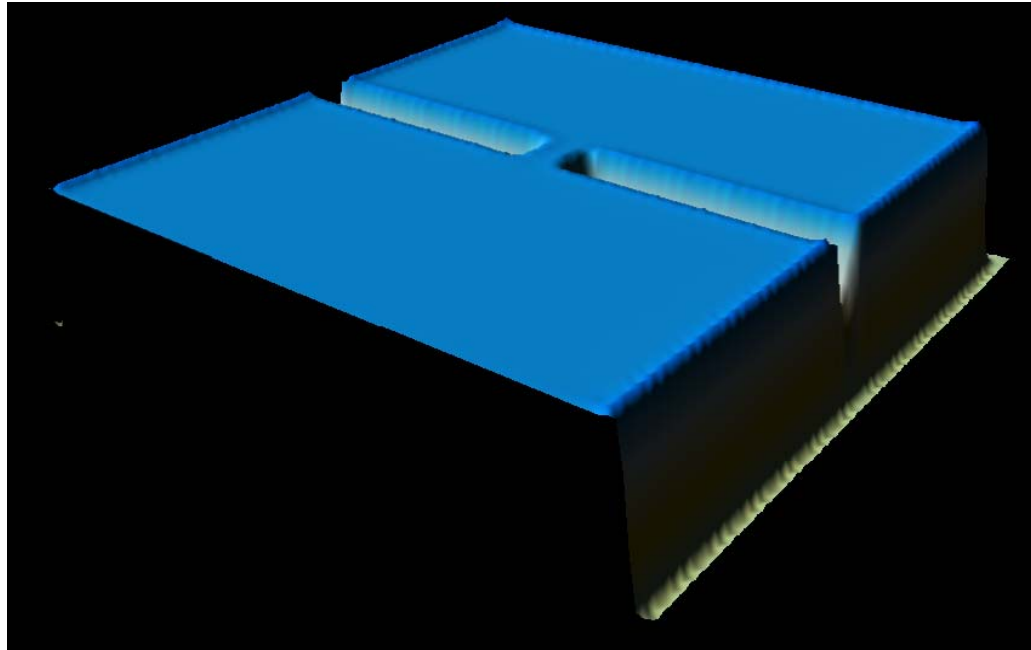
La prueba para la validación de este caso consiste en un depósito rectangular en el que existe una pared con una abertura que opera como obstáculo del flujo, el fluido entra a la región de estudio con una dirección y sale en otras dos perpendicular a la inicial. El  $(\Delta x)$  y  $(\Delta y)$  utilizado en este caso es igual a 10 m, para un mallado de 40x40 puntos que abarca toda la región. La profundidad se mantiene constante y es de 3 m en todo el dominio.

El tiempo de cada paso de iteraciones  $(\Delta t)$  fue de 6 s dando un tiempo total de evaluación de 300 s. El caudal utilizado es de  $0,1 \frac{m^3}{s}$  y la altura de la superficie libre del agua es de 3 m con respecto a un datum fijado. A continuación se presentan los gráficos obtenidos al desarrollar el ejemplo.

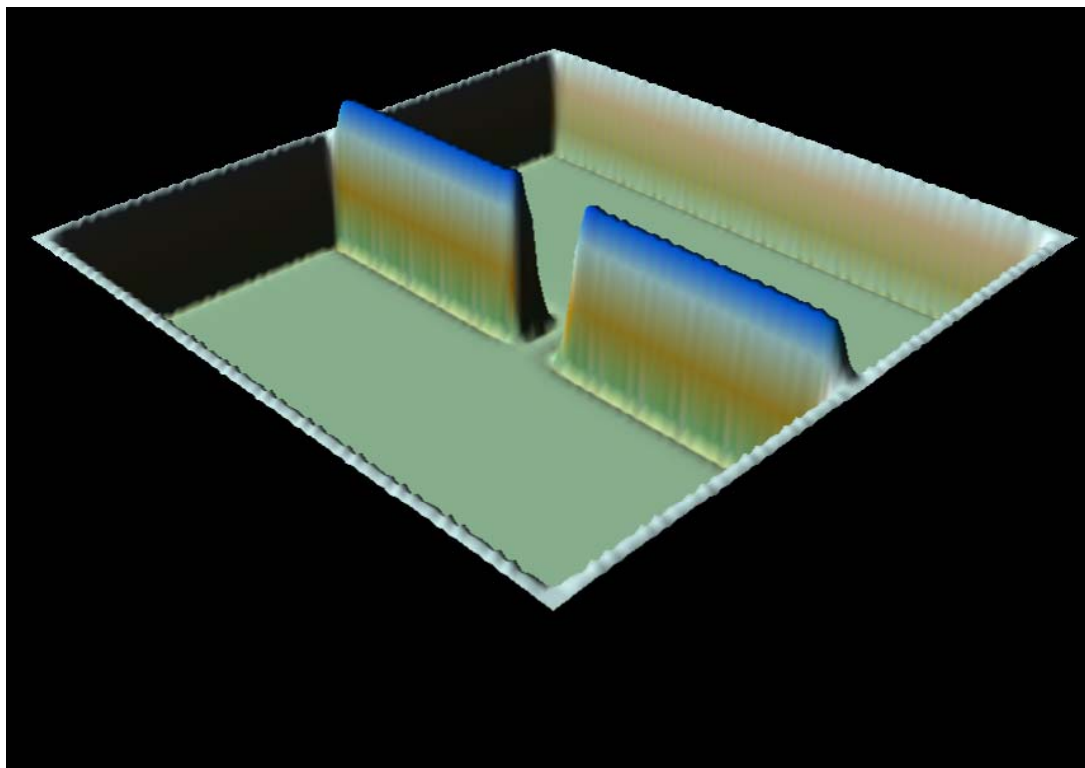


**Figura (4.14) Vectores velocidades para ( $t = 300 \text{ s}$ ,  $\Delta x = 10$ ,  $\Delta y = 10$ ,  $Q = 0.1 \text{ m}^3$  /min. y  $\Delta t = 6 \text{ s}$ ).**

En este punto de la evaluación del modelo, es posible apreciar la generación de vórtices simétricos respecto al eje x, en el campo de velocidades del fluido justo después del choque contra la pared frontera del ejemplo en estudio



**Figura (4.15) Perfil tridimensional de vorticidad.**

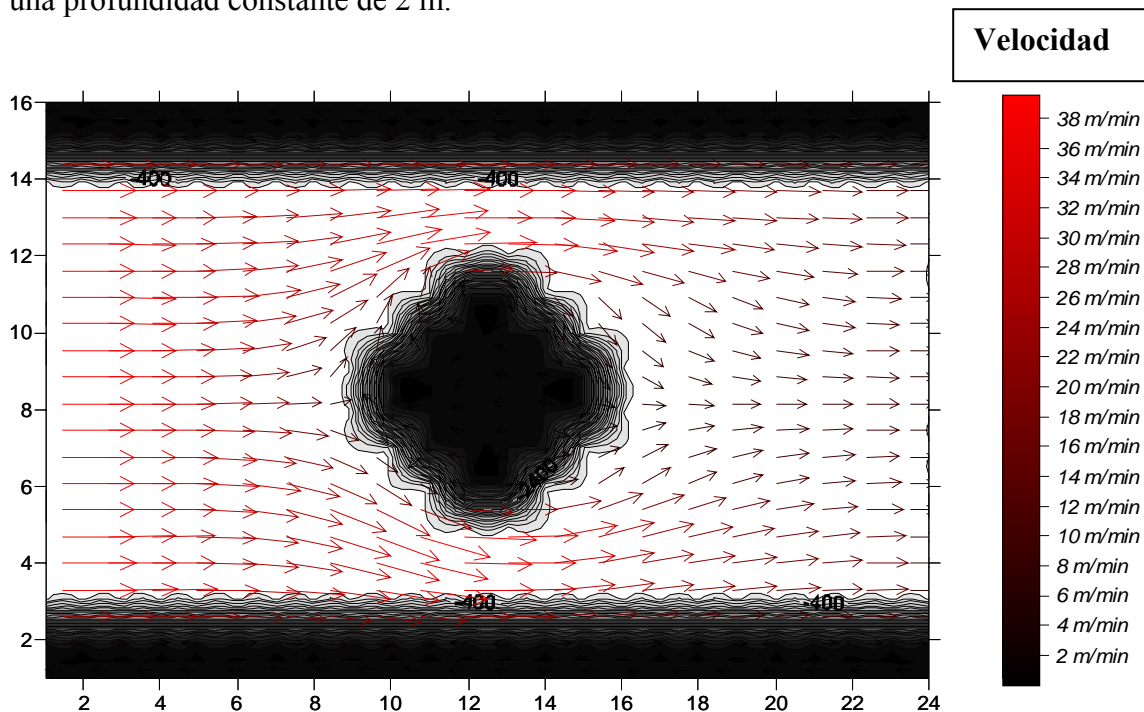


**Figura (4.16) Perfil tridimensional del fondo para ( $Z = 3$  m,  $D = 3$  m).**



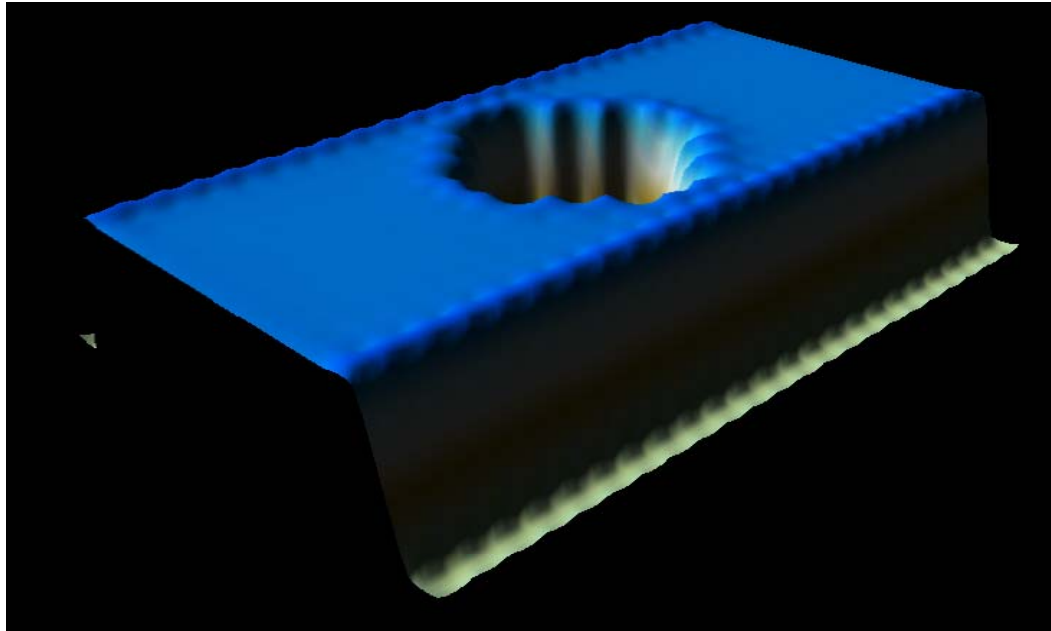
#### 4.2.4 Contracción de canal abierto.

El ejemplo utilizado consta de un canal con la entrada y salida del fluido en la misma dirección y sentido, con un obstáculo (isla) en el trayecto que ocasiona variaciones en el patrón de movimiento del agua. Se usó un mallado en la región de estudio de 20x12 puntos con un espaciado entre cada uno de 100 m ( $\Delta x$ ,  $\Delta y$ ) y posee una profundidad constante de 2 m.

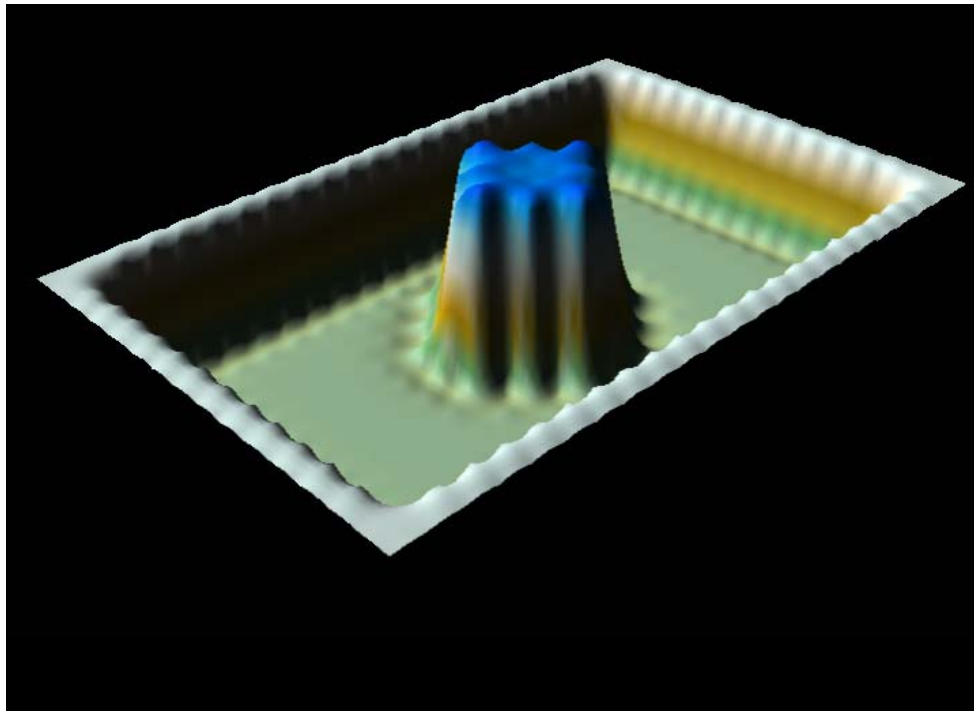


**Figura (4.17) Vectores velocidades para ( $t = 250$  s,  $\Delta x = 100$ ,  $\Delta y = 100$ ,  $Q = 7500$   $m^3/min.$  y  $\Delta t = 5$  s).**

La evaluación de este ejemplo se realizó con un paso de tiempo de 5 s, dando así un tiempo total de estudio de 250 s, para un caudal inicial de  $7500 m^3/min.$  Al existir algún elemento que impida el desarrollo regular del fluido se genera un comportamiento particular en el campo de velocidades que debido a la geometría es simétrico respecto al eje x.



**Figura (4.18) Perfil tridimensional de contracción de canal abierto.**



**Figura (4.19) Perfil tridimensional del fondo para ( $Z = 3$  m,  $D = 2$  m).**

#### 4.2.5 Caso real Lago de Maracaibo.

Hasta el presente, efectuamos simulaciones con modelos relativamente simples. Un caso industrial de aplicación del modelo de Beam and Warming modificado es el caso del Lago de Maracaibo.

En ese caso, haremos frente a variaciones importantes de la forma y de la batimetría. Este ejemplo se representa con un modelo hidrodinámico tridimensional. El lago pertenece al sistema de Maracaibo, el cual está constituido también por: El Estrecho de Maracaibo, Bahía el Tablazo y el Golfo de Venezuela. Pero para razones de este trabajo se tomará solo en cuenta el lago por tener la mayor afluencia de flujo.

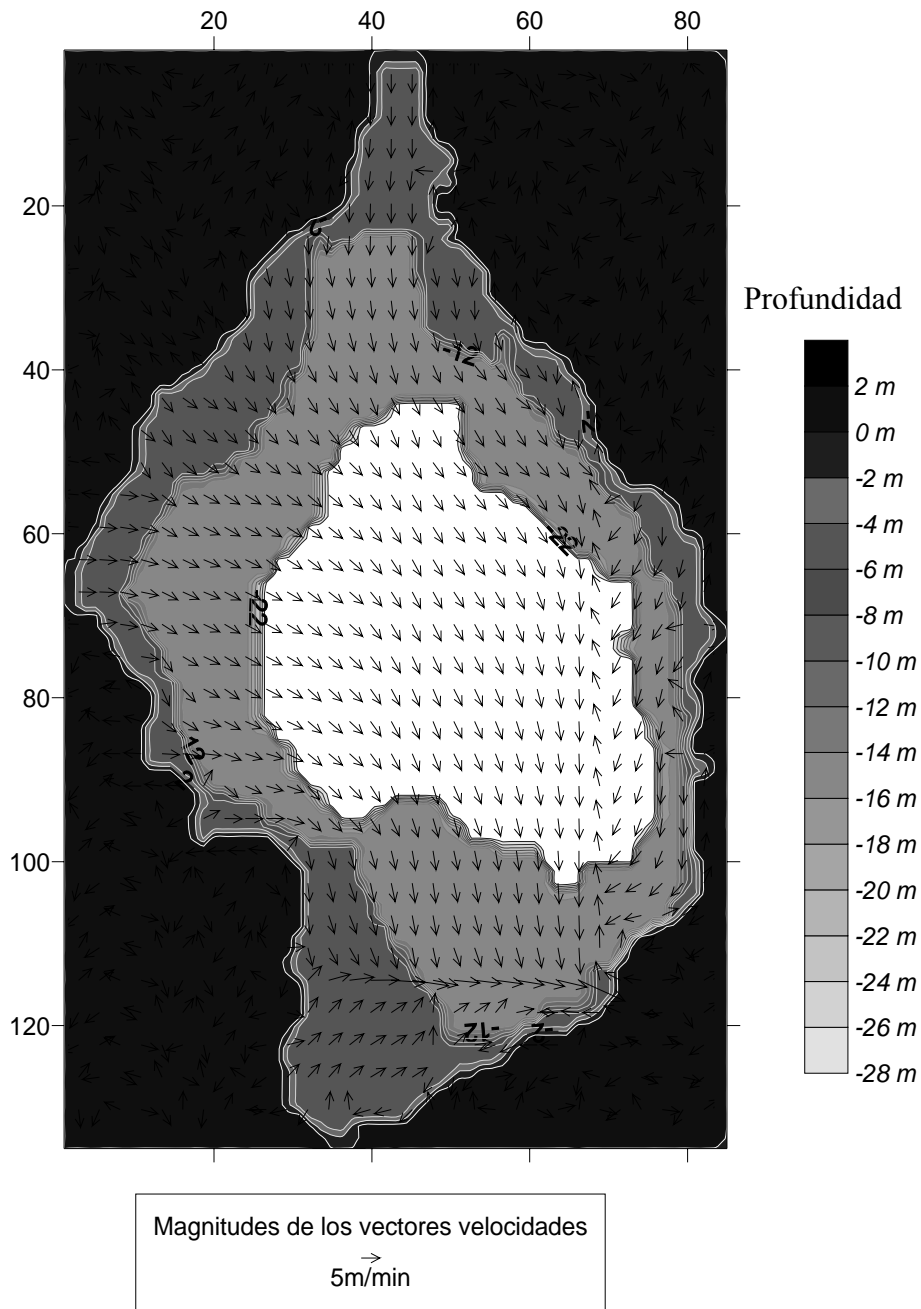
Las dimensiones horizontales del Lago de Maracaibo son de sur a norte 132 kilómetros y de este a oeste 82 kilómetros. La profundidad máxima es un poco más de 30 m.

La malla utilizada es lo suficientemente grande ( $\Delta x = 1 \text{ Km.}$  y  $\Delta y = 1 \text{ Km.}$ ) para abarcar en su totalidad la longitud del lago. La escala de la malla vertical tiene un espaciado de 5 m, y se utilizó para representar la elevación de la superficie del agua. El flujo de la marea de entrada al Lago de Maracaibo es de unos  $6000 \text{ m}^3 / \text{min.}$

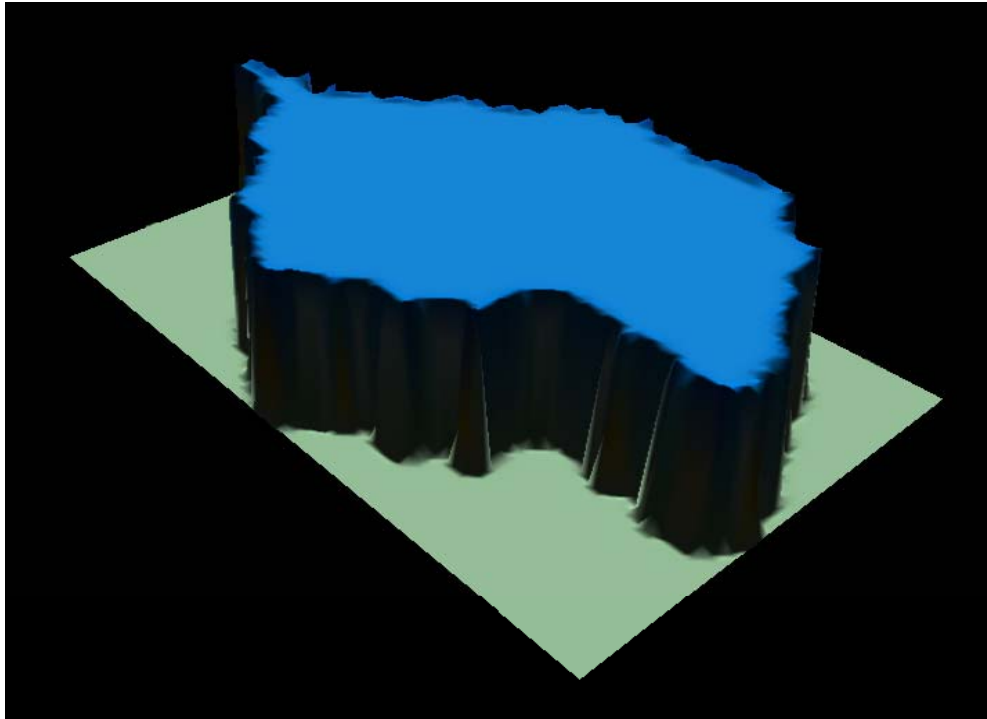
El lago se encuentra cerrado en casi toda su extensión, a excepción de algunas zonas en donde se encuentra presente las desembocaduras de algunos ríos y el canal de entrada. Este último se encuentra en la zona El Tablazo, y es por donde entra la marea. En los lugares en donde el lago se encuentra cerrado, existen oscilaciones al nivel de la superficie. El aumento ó la disminución del nivel son debido a la amortiguación ocasionada por el rozamiento sobre el fondo. Este fenómeno es

acompañado de una inversión del sentido de las velocidades de la zona hacia la salida.

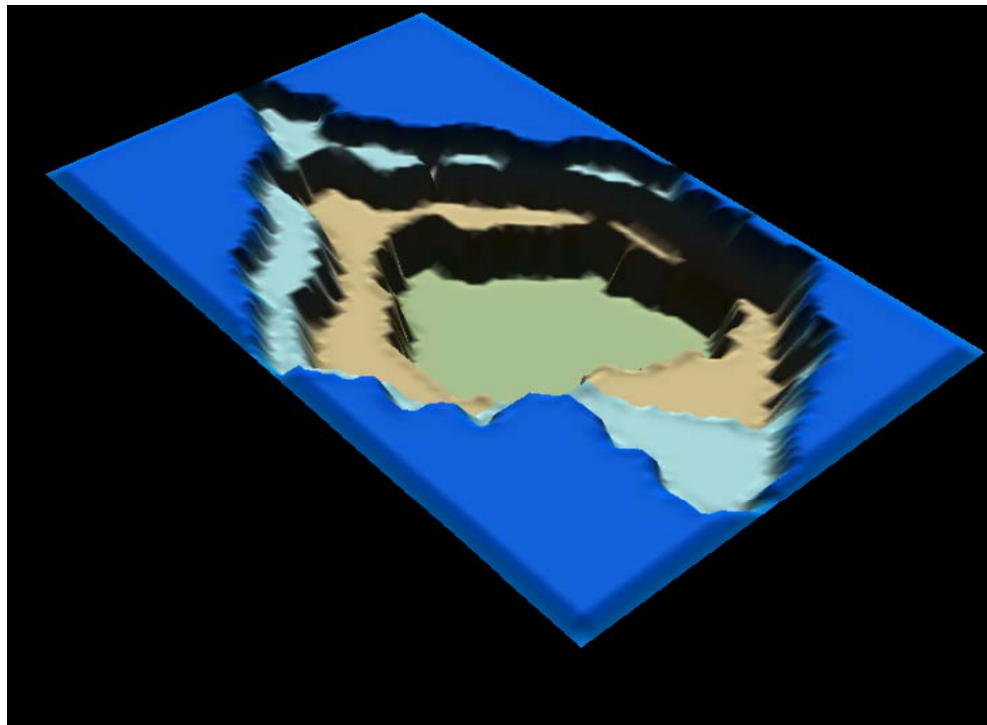
La figura (4.20) muestra la formación gradual de una zona de recirculación y puede ser debido a la batimetría entre la entrada del lago y el lago mismo. Para el análisis de este fenómeno pasajero hay que utilizar  $\Delta t$  más pequeños para respetar las condiciones reales del problema, debidas a la variación del flujo a la entrada.



**Figura (4.20) Vectores velocidades para ( $t = 10000$  s,  $\Delta x = 1$  Km.,  $\Delta y = 1$  Km.,  $Q = 6000$  m<sup>3</sup> /min. y  $\Delta t = 200$  s).**



**Figura (4.21) Perfil tridimensional del Lago de Maracaibo.**



**Figura (4.22) Perfil tridimensional del fondo para ( $Z = 30$  m,  $D = 5-30$  m)**

## **CONCLUSIÓN**

En esta memoria, el método numérico de Beam and Warming ha sido presentado con modificación en la evaluación de sus matrices jacobianas utilizando la técnica del jacobiano aproximativo, propuesto por Roe.

Así como fue señalado en los capítulos anteriores, el modelo numérico ha sido validado con una serie de pruebas, y al mismo tiempo, ha sido comparado con soluciones analíticas y experimentales que se citan en la literatura. También, la capacidad del modelo ha sido puesta a prueba para la aplicación industrial de los flujos confinados por fronteras irregulares y batimetría compleja, obteniendo un comportamiento acertado.

Después de la simulación realizada sobre el método numérico de Beam and Warming, podemos concluir que en los problemas de nivel de agua y su velocidad, los resultados obtenidos son semejantes a la solución analítica.

En los problemas donde se presentaba un cambio brusco de gradiente, el margen de error es prácticamente eliminado si se lo compara con la solución analítica. Además, el aumento del tiempo necesario para el cálculo en la versión modificada, es un poco menor al de la versión original; y está alrededor del 10 %. Esto nos permite afirmar que el esquema de Beam and Warming modificado es más preciso que el esquema de Beam and Warming original y no representa ningún tipo de variación significativa al tiempo de cálculo dispuesto por el método sin modificación.

Varias simulaciones han sido realizadas utilizando valores diferentes del número de Courant. Éste varía entre 0,1 y 0,9; observamos que el número de Courant no afecta el resultado de la solución. La solución converge, en poco tiempo, para altos valores del número de Courant. Sin embargo, varios investigadores sugieren un intervalo de 0,75 a 0,85.



## **RECOMENDACIONES**

Las recomendaciones siguientes son sugeridas a los futuros investigadores en este dominio:

Es necesario desarrollar un método para una malla no cartesiana, lo que podría simplificar el trabajo, cuando el problema que hay que estudiar tiene una geometría compleja. Por ejemplo, fronteras irregulares con curvaturas dentro del dominio de cálculo.

Ampliar el campo de problemas a modelar mediante la continua calibración del esquema de Beam and Warming, según se requiera, para lograr la predicción de campos de velocidades y alturas, ocasionados por variaciones abruptas en las condiciones iniciales del problema.

Realizar ajustes a la plataforma del programa para lograr simulaciones de modelos con batimetrías mucho más complejas. Esto permitiría la modelación de innumerables ejemplos, sin que exista ninguna restricción para el desarrollo de los mismos.

## REFERENCIAS BIBLIOGRÁFICAS

- **ABDELFETTAH, F.**, 1994, *Évaluation d'une méthode d'Euler-Lagrange pour la modélisation des écoulements à surface libre et de l'équation de transport*, Mémoire de maîtrise ès sciences appliquées. École Polytechnique de Montréal., Canada.
- **BEAM, R.M. & WARMING, R.F.**, 1976, *An implicit finite-difference algorithm for hyperbolic systems in conservation-law form*, Comput. Phys., pp. 87-110.
- **BEAM, R.M. & WARMING, R.F.**, 1978, *Implicit factored schemes for conservation laws*, Symposium on computational Fluid Dynamics, NY. SIAM-AMS. Vol. 11, pp 85-129.
- **BISSON, J.**, 1994, *Modélisation d'écoulement à surface libre en deux dimensions utilisant un schéma implicite*, Ecole Polytechnique de Montréal.
- **BOLINAGA, J.J.**, 1992, *Mecánica Elemental de los Fluidos*, Universidad Católica Andrés Bello. Fundación Polar, Caracas.
- **CHAUDHRY, M.H.**, 1993, *Open Channel Flows*, Prentice Hall, Englewood Cliffs, New Jersey.
- **DE AGUIAR, J.F.**, 1995, *Modélisation des Écoulements Bidimensionnels Avec Transports des Polluants. Applications dans la Qualité de l'eau*, Mémoire présenté en vue de l'obtention du diplôme de maîtrise ès sciences appliquées. Université de Montréal, Département de Génie Civil, École Polytechnique.
- **FENNEMA, R.J. & CHAUDHRY, M.H.**, 1987, *Simulation of one-dimensional Dam-Break Flows*, pp. 41-51.
- **FENNEMA, R.J. & CHAUDHRY, M.H.**, 1990, *Implicit methods for two-dimensional unsteady free-surface flows*, ASCE, pp. 1013-1034.

- **FENNEMA, R.J.**, 1985, *Numerical Solution of two Dimensional Transient Free-Surface Flows*, A dissertation Presented to Washington State University, in partial fulfilment of the requirement for the degree of Doctor of Philosophy.
- **GARCIA, F.R. & KAHAWITA, R.**, 1986, *Numerical solution of the Saint-Venant equations with the McCormack finite difference Scheme*, pp. 259-274.
- **HANSEN, I. & MILER, J. & FINDIKAKIS, A.**, 1998, *Three Dimensional Hydrodynamic Modelling of Maracaibo System, Venezuela*, [http://www.iarg.org/e-library/beijing\\_proceedings](http://www.iarg.org/e-library/beijing_proceedings). Recuperado el 22 de junio de 2006.
- **MOLINARO, P. & NATALE, L.**, 1994, *Modelling of Flood Propagation over inially dries Areas*, American Society of Civil Engineers.
- **POTTER, M. & WIGGERT D.C.**, 2002, *Mecánica de Fluidos*, Prentice Hall Hispanoamericana S.A., 3ª Edición, México DF.
- **ROE, P.L.**, 1981, *Approximate Riemann Solvers, Parameter Vectors and Difference Schemes*, pp. 357-372.
- **ROE, P.L.**, 1986, *Characteristic-Based Scheme for the Euler Equations*, pp. 337-365.
- **ROCA, V.**, 1998, *Introducción a la Mecánica de Fluidos*, Editorial Limusa, México DF.
- **ZHANG, H.**, 1984, *Mathematical Modeling of Two-Dimensional Hydraulic Problems Using Boundary-Fitted Coordinates*. Mémoire présenté en vue de l'obtention du grade de maître ès sciences appliquées. Université de Montréal, Département de Génie Civil, École Polytechnique.

## **APÉNDICE A: DESCRIPCIÓN DEL PROGRAMA**

### **A.1 Descripción de las subrutinas:**

El programa desarrollado en esta investigación ha sido escrito en el lenguaje de programación estándar FORTRAN. A continuación se describe brevemente el funcionamiento del programa principal y las subrutinas principales usadas en el modelo computacional.

#### **1) Programa MFESM:**

Llama las subrutinas de entrada principales y calcula las constantes iniciales. Controla los procedimientos de iteración y llama las subrutinas computacionales, SOLVU, SOLVV y SOLVD

#### **2) Subrutina SOLVU:**

Soluciona las velocidades en la dirección x en el tiempo  $n + 1$  para un barrido, solo formando la matriz apropiada y luego invirtiéndola. Incluye todos los límites requeridos.

#### **3) Subrutina SOLVV:**

Soluciona las velocidades en la dirección y en el tiempo  $n + 1$  para un barrido, solo formando la matriz apropiada y luego invirtiéndola. Incluye todos los límites requeridos

4) Subrutina SOLVD:

Soluciona las profundidades y elevaciones de superficie del agua en el tiempo  $n + 1$ . Los límites están incorporados en estas subrutinas.

5) Subrutinas CHBO1, CHBO2 y SPEC:

Resuelve los elementos de la matriz en posiciones de frontera según el tipo de la condición contorno existente, específico o derivada.

6) Subrutinas BNDEXV, BNDEXD y CALFLO:

Soluciona los valores de frontera externos necesarios para la velocidad y profundidades. Calcula la velocidad en posiciones de entrada y salida del flujo y profundidad.

7) Subrutinas UPSTR y CENTR:

Resuelve los valores intermedios según el esquema que debe ser incorporado, aguas arriba o diferenciación central.

8) Subrutina TRIDAG:

Invierte y soluciona matrices tridiagonales que consisten en los vectores A, B, C y R. Devuelve la solución en R.

9) Subrutinas WRITEM y WRITER:

Imprime el resultado deseado en notación de una matriz rectangular.

10) Subrutina UPDATE:

Actualiza las variables para el dominio temporal para el cálculo inestable. Toma valores en el tiempo  $n + 1$  y los convierte a valores en el tiempo  $n$ .

11) Subrutina READIN:

Lee en todos los datos iniciales e imprime de los datos. Realiza los cálculos iniciales requeridos para inicializar las matrices.

12) Subrutinas CALC1 y CALC2:

Calcula todas las constantes requeridas para la solución de las variables, para simulaciones estables o inestables.

A.2 Descripción del archivo de Datos:

Esta sección describe detalladamente los datos que son requeridos a fin de dirigir el programa. Cada archivo es descrito individualmente.

a) Parámetros de tiempo y escritura:

READ (7, \*) DELT, NTIME, NWRITE

Donde:

DELT: paso de tiempo.

NTIME: número de iteraciones de tiempo.

NWRITE: intervalo de escritura de salida.

b) Constantes Seleccionadas

```
READ (7, *) GAM, G, CHEZY, SLIP
```

Donde:

GAM: Coeficiente de difusión.

G: Aceleración de la gravedad.

CHEZY: Coeficiente de Chezy.

SLIP: Condición de slip.

c) Parámetros de iteración

```
READ (7, *) NOIT1, NOIT2
```

Donde:

NOIT1: número de iteraciones estables.

NOIT2: número de iteraciones por cada paso de tiempo.

d) Tamaño de las celdas.

```
READ (7, *) IMAX, JMAX
```

Donde:

IMAX: número de elementos en la dirección x

JMAX: número de elementos en la dirección y

e) Dimensiones de las celdas:

```
READ (7, *) (DX (I), I = 3, IMAXP2)
```

```
READ (7, *) (DY (J), J = 3, JMAXP2)
```

Donde:

DX (I) = dimensión de cada celda en dirección x.

DY (J) = dimensión de cada celda en dirección y.

f) Cell types:

```
READ (7, *) (ITYP (I, J), I = 3, IMAXP2)
```

Donde:

ITYP: definición del tipo de celda.

g) Número de entradas y salidas:

```
READ (7, *) NBOUN
```

Donde:

NBOUN: numero de celdas de frontera.



h) Información de frontera:

```
READ (7, *) I, J, IVEL (I, J), IBV (I, J), IBD (I, J), Q (I, J)
```

Donde:

I, J: coordenadas de ubicación.

IVEL: dirección del flujo.

IBV: velocidad de contorno.

IBD: profundidades de bordes.

Q: flujo específico de entrada.

i) Campo de profundidades:

```
READ (7, *) (D (I, J), I = 3, IMAXP2)
```

Donde:

D = profundidad inicial del agua.

j) Elevación de la superficie:

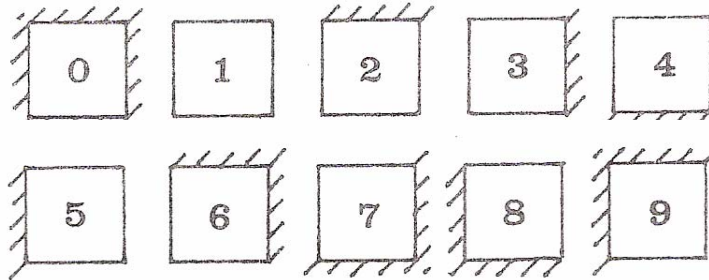
```
READ (7, *) ELEVS
```

Donde:

ELEVS = elevación inicial de la superficie de agua.

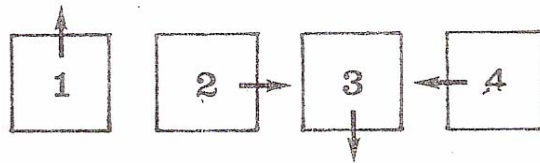
ITYP (I, J)

Tipo de Celda



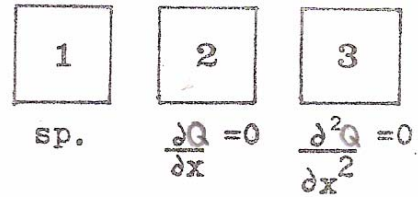
IVEL (I, J)

Tipo de Límite



IBV (I, J)

Definición de la velocidad límite



IBD (I, J)

Definición de la profundidad límite

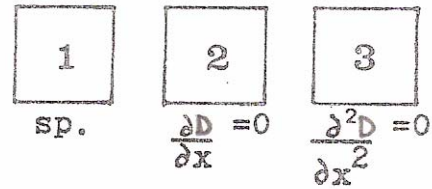


Figura (A.1) Definiciones límites y tipos de celdas.

	inlet		
outlet	Q = sp. D = sp.	Q = sp. $\partial D/\partial x = 0 \checkmark$	Q = sp. $\partial^2 D/\partial x^2 = 0$
Q = sp. D = sp.	* stable	* stable	unstable
Q = sp. $\partial D/\partial x = 0$	* stable	unstable	unstable
$\partial Q/\partial x = 0$ D = sp.	* stable	* stable $\checkmark$ * accurate	* stable
$\partial Q/\partial x = 0$ $\partial D/\partial x = 0$	* stable	unstable	unstable
$\partial^2 Q/\partial x^2 = 0$ $\partial D/\partial x = 0$	unstable	unstable	unstable
$\partial Q/\partial x = 0$ $\partial^2 D/\partial x^2 = 0$	* stable	* stable	unstable

Figura (A.2) Análisis de resultado tipo límite.

**APENDICE B: LISTADO DEL PROGRAMA**

```

C =====
C =
C =                      MFESM
C =
C =          MODELO DE FLUJO DE EMBALSE SEMI-IMPLÍCITO
C =
C =          - DOS DIMENSIONES
C =          - PROMEDIO VERTICAL
C =          - INTEGRADO LATERALMENTE
C =          - MALLA NO-UNIFORME
C =          - BARRIDOS DE VELOCIDADES
C =          - ESQUEMA PARA PROFUNDIDADES
C =          - CÁLCULOS DE SUPERFICIE LIBRE
C =          - OPCIONES DE ITERACIÓN
C =
C =====

```

```

      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 VAR (138,138)
      INTEGER*4 AORC
      INTEGER*4
I,J,K, JJ, ITYP, IVEL, IBV, IBD, IMAX, JMAX, IMAXP1, JMAXP1
      e          IMAXP2, JMAXP2, IMAXP3, JMAXP3, IMAXP4, JMAXP4
      COMMON/CELL/DX (138), CDX (138), DY (138), CDY (138)
      COMMON/BOUND/ITYP (138,138), IVEL (138,138), IBV
(138,138)
      e          , IBD(138,138)
      COMMON/VELX/U(138,138), UPP(138,138)
      COMMON/VELY/V(138,138), VPP(138,138)
      COMMON/DEP/D(138,138), DP(138,138), DPP(138,138)
      COMMON/TDAG/A(138), B(138), C(138), R(138)
      COMMON/FLOW/Q(138,138)
      COMMON/CONS1/DELT, G, CHEZY, GAM, NOIT1, NOIT2, NTIME
      e          , NWRITE
      COMMON/CONS2/DELT1, DELT2, GC2, SLIP
      COMMON/CONU1/GDELX(138), GDT2DX(138), GDTDXP(138)
      e          , GDTDXM(138)
      COMMON/CONU2/GMDXPU(138), GMDXMU(138), GMDYMU(138)

```

```

e          ,GMDYPU(138)
COMMON/CONV1/GDELY(138),GDT2DY(138),GDTDYP(138)
e          ,GDTDYM(138)
COMMON/CONV2/GMDYPV(138),GMDYMV(138),GMDXPV(138)
e          ,GMDXMV(138)
COMMON/VARZ1/ZDIMJ(138,138),ZDIJ1(138,138),ZDIPJ(138
,138)
e          ,ZRHS(138,138)
COMMON/VARZ2/ZDIJM(138,138),ZDIJ2(138,138),ZDIJP(138
,138)
COMMON/SIZE0/IMAX,JMAX
COMMON/SIZE1/IMAXP1,JMAXP1
COMMON/SIZE2/IMAXP2,JMAXP2
COMMON/SIZE3/IMAXP3,JMAXP3
COMMON/SIZE4/IMAXP4,JMAXP4
COMMON/ELEV/Z(138,138),ZPP(138,138)

OPEN(7,FILE='data contraccion.txt',STATUS='OLD')
OPEN(8,FILE='salida.txt',STATUS='OLD')
OPEN(9,FILE='VELOCIDADx.DAT',STATUS='OLD')
OPEN(10,FILE='PROFUNDIDAD.DAT',STATUS='OLD')
OPEN(11,FILE='fondo.txt',STATUS='OLD')
OPEN(12,FILE='VELOCIDADy.Dat',STATUS='OLD')
WRITE(6,999)
999  FORMAT('1')

C  =====
C  =
C  =          LEA Y ESCRIBA DATOS          =
C  =
C  =====
      CALL READIN
      WRITE(6,6001)
      WRITE(8,6001)
6001  FORMAT('/', 'CONDICIONES INICIALES:', '/')
      CALL WRITER(U,V,Z)

C
C      INICIALICE CONSTANTES INDEPENDIENTES DEL TIEMPO
C
      CALL CALC1
      CALL CALC2

```

```

C      =====
C      =
C      =      ITERACIONES INICIALES (CÁLCULOS ESTABLES)      =
C      =
C      =====

      IF(NOIT1.EQ.0) GO TO 1002
      WRITE(6,6004)
6004  FORMAT('VALORES DE ITERACIONES PARA CONDICIONES
ESTABLES=',//)
      DO 100 ITER1=1,NOIT1
      CALL SOLVU
      CALL SOLVV
100   WRITE(6,6005) UPP(2,4),vpp(2,4)

6005  FORMAT(2F10.2)

      DO I1=1,IMAXP4
      DO J1=1,JMAXP4
      WRITE(9,*)I1,J1,UPP(I1,J1)
      WRITE(12,*)I1,J1,VPP(I1,J1)
      ENDDO
      ENDDO

      CALL SOLVD

      WRITE(6,6002) NOIT1
      WRITE(8,6002) NOIT1
6002  FORMAT(//,'LUEGO DE',I4,' ITERACIONES INICIALES, SE
OBTIENE:
e      ',//)
      CALL WRITER(UPP,VPP,ZPP)

      CALL UPDATE
1002  CONTINUE

```

```

C      =====
C      =
C      =          ITERACIONES DEL TIEMPO (CÁLCULOS          =
C      =          INESTABLES)                               =
C      =
C      =====
NWRIT=1
DO 900 IT=1,NTIME

C      ITERE DE SER DESEADO

DO 901 ITER2=1,NOIT2
CALL SOLVU
901  CALL SOLVV
CALL SOLVD

C      ESCRIBA RESULTADOS CUANDO SEA DESEADO

IF(NWRIT.NE.NWRITE) GO TO 1000
TIME=DELT*IT
WRITE(6,6000) IT,DELT,TIME
WRITE(8,6000) IT,DELT,TIME
6000  FORMAT(/,' PASO DE TIEMPO
NUMERO=',I3,T100,'DELT=',F10.2
e      ,//,' TIEMPO TOTAL=',F10.2,/)
CALL WRITER(UPP,VPP,ZPP)
NWRIT=0
1000  NWRIT=NWRIT+1

C      VARIABLES DE INCREMENTO PARA EL DOMINIO TEMPORAL

CALL UPDATE
900  CONTINUE

C      =====
C      =
C      =          LAZO DE TIEMPO FINAL Y PROGRAMA PRINCIPAL  =
C      =
C      =====
rewind(7)

```

```

rewind(8)
rewind(9)
REWIND(10)
REWIND(11)
REWIND(12)
Close(8)
close(7)
close(9)
CLOSE(10)
CLOSE(11)
CLOSE(12)
STOP
END

```

```

SUBROUTINE SOLVU

```

```

C =====
C =
C = SOLUCIONES PARA LAS VELOCIDADES "X" A =
C = TRAVÉS DE 1 PASO DE TIEMPO COMPLETO =
C =
C =====
IMPLICIT REAL*8 (A-H,O-Z)
REAL*8 VAR(138,138)
INTEGER*4 AORC
INTEGER*4
I,J,K, JJ, ITYP, IVEL, IBV, IBD, IMAX, JMAX, IMAXP1, JMAXP1
e IMAXP2, JMAXP2, IMAXP3, JMAXP3, IMAXP4, JMAXP4
COMMON/CELL/DX(138), CDX(138), DY(138), CDY(138)

COMMON/BOUND/ITYP(138,138), IVEL(138,138), IBV(138,138)
e , IBD(138,138)
COMMON/VELX/U(138,138), UPP(138,138)
COMMON/VELY/V(138,138), VPP(138,138)
COMMON/DEP/D(138,138), DP(138,138), DPP(138,138)
COMMON/TDAG/A(138), B(138), C(138), R(138)
COMMON/FLOW/Q(138,138)
COMMON/CONS1/DELT, G, CHEZY, GAM, NOIT1, NOIT2, NTIME
e , NWRITE
COMMON/CONS2/DELT1, DELT2, GC2, SLIP
COMMON/CONU1/GDELX(138), GDT2DX(138), GDTDXP(138)
e , GDTDXM(138)
COMMON/CONU2/GMDXPU(138), GMDXMU(138), GMDYMU(138)

```



```

e          ,GMDYPU(138)
COMMON/CONV1/GDELY(138),GDT2DY(138),GDTDYP(138)
e          ,GDTDYM(138)
COMMON/CONV2/GMDYPV(138),GMDYMV(138),GMDXPV(138)
e          ,GMDXMV(138)
COMMON/VARZ1/ZDIMJ(138,138),ZDIJ1(138,138),ZDIPJ(138
,138)
e          ,ZRHS(138,138)
COMMON/VARZ2/ZDIJM(138,138),ZDIJ2(138,138),ZDIJP(138
,138)
COMMON/SIZE0/IMAX,JMAX
COMMON/SIZE1/IMAXP1,JMAXP1
COMMON/SIZE2/IMAXP2,JMAXP2
COMMON/SIZE3/IMAXP3,JMAXP3
COMMON/SIZE4/IMAXP4,JMAXP4
COMMON/ELEV/Z(138,138),ZPP(138,138)

```

```

C =====
C =
C = COMIENCE EL BARRIDO EN LA DIRECCIÓN "X" =
C =
C =====

```

```

DO 100 J=3,JMAXP2
JP=J+1
JM=J-1
DYJ=DY(J)
DO 101 I=2,IMAXP2

```

```

C VARIABLES PARA VELOCIDADES DE RESBALÓN

```

```

CDXI=CDX(I)
IP=I+1
IM=I-1
UIJM=UPP(I,JM)
UIJ=UPP(I,J)
UIJP=UPP(I,JP)
UIMJ=UPP(IM,J)
UIPJ=UPP(IP,J)
VIJ=VPP(I,J)
VIPJ=VPP(IP,J)
VIJM=VPP(I,JM)

```

```

VIPJM=VPP(IP,JM)

C      CALCULE TODAS LAS VELOCIDADES DE RESBALÓN

      ITYPP=ITYP(I,J)+1

      GO TO (1000,1001,1002,1003,1004,1099,1003,1003,1004
e        ,1002),ITYPP

C      ITYP=0

1000 IF(IVEL(IP,J).EQ.4) GO TO 1010
      CALL SPEC(I,0)
      GO TO 2099
1010 IF(IBV(IP,J).EQ.1) GO TO 1011
      VIJ=SLIP*VIPJ
      VIJM=SLIP*VIPJM
      IF(IVEL(IP,JP).NE.4) UIJP=SLIP*UIJ
      IF(IVEL(IP,JM).NE.4) UIJM=SLIP*UIJ
      GO TO 1099
1011 CALL CALFLO(Q(IP,J),IVEL(IP,J),VEL,IP,J)
      CALL SPEC(I,VEL)
      GO TO 2099

C      ITYP=1

1001 ITYPP=ITYP(IP,J)
      GO TO (1099,1020,1099,1021,1099,1020,1021,1099,
e        1099),ITYPP
1020 UIJP=SLIP*UIJ
      VIPJ=SLIP*VIJ
      GO TO 1099
1021 UIJM=SLIP*UIJ
      VIPJM=SLIP*VIJM
      GO TO 1099

C      ITYP=2,9

1002 UIJP=SLIP*UIJ
      IF(IVEL(I,J).EQ.1.AND.IVEL(I,JP).EQ.1) GO TO 1099
      VIJ=SLIP*VIPJ
      GO TO 1099

```

```

C      ITYP=3,6,7

1003 IF(IVEL(I,J).EQ.2) GO TO 1050
      CALL SPEC(I,0)
      GO TO 2099
1050 IF(IBV(I,J).EQ.1) GO TO 1051
      VIPJ=SLIP*VIJ
      VIPJM=SLIP*VIJM
      IF(IVEL(I,JP).NE.2) UIJP=SLIP*UIJ
      IF(IVEL(I,JM).NE.2) UIJM=SLIP*UIJ
      GO TO 1099
1051 CALL CALFLO(Q(I,J),IVEL(I,J),VEL,I,J)
      CALL SPEC(I,VEL)
      GO TO 2099

C      ITYP=4,8

1004 UIJM=SLIP*UIJ
      IF(IVEL(I,J).EQ.3.AND.IVEL(I,JP).EQ.3) GO TO 1099
      VIJM=SLIP*VIPJM

C
1099 CONTINUE

C      VARIABLES PARA PROFUNDIDADES

      DIJ=DPP(I,J)
      DIMJ=DPP(IM,J)
      DIPJ=DPP(IP,J)
      DIJM=DPP(I,JM)
      DIJP=DPP(I,JP)
      DIPJP=DPP(IP,JP)
      DIPJM=DPP(IP,JM)
      DIP2J=DPP(I+2,J)

C      CALCULAR AGUAS ARRIBA Y TÉRMINOS DE DIFERENCIACIÓN
CENTRAL

      CALL UPSTR(UIMJ+UIJ,UNL,UPL)
      CALL UPSTR(UIJ+UIPJ,UNR,UPR)
      CALL CENTR(DIPJ,DIP2J,DIPJU)
      CALL CENTR(DIJ,DIPJ,DIJU)
      CALL CENTR(DIMJ,DIJ,DIMJU)
      CALL CENTR(DIPJ,DIPJP,DIPJV)

```

```

CALL  CENTR(DIPJM,DIPJ,DIPJMV)
CALL  CENTR(DIJ,DIJP,DIJV)
CALL  CENTR(DIJM,DIJ,DIJMV)
CALL  CENTR(VIJM,VIPJM,VIJMU)
CALL  CENTR(VIJ,VIPJ,VIJU)

C    =====
C    =
C    =  CALCULE COEFICIENTES PARA INTRODUCCIÓN DE LA  =
C    =  MATRIZ                                          =
C    =
C    =
C    =====

A(I)=-GDTDXM(I)*DIMJU-UPL*UIMJ/CDXI-GMDXMU(I)
B(I)=DELT1+(GDTDXM(I)+GDTDXP(I))*DIJU+(UPR-UNL)*UIJ
e
/CDXI+(GMDXMU(I)+GMDXPU(I))+GC2/DIJU*(abs(UIJ*UIJ)
e      +(abs(VIJU+VIJMU)/2.0)**2)**(1/2)
C(I)=-GDTDXP(I)*DIPJU+UNR*UIPJ/CDXI-GMDXPU(I)
R(I)=UIJM*(-VIJMU/2.0/DYJ-GMDYMU(J))
e      +UIJ*((VIJU-VIJMU)/2.0/DYJ+(GMDYMU(J)
e      +GMDYPU(J)))+UIJP*(VIJU/2.0/DYJ-GMDYPU(J))
e      -GDT2DX(I)/DYJ*(VPP(IP,JM)*DIPJMV-VPP(IP,J)
e      *DIPJV+VPP(I,J)*DIJV-VPP(I,JM)*DIJMV)
e      -GDELX(I)*(Z(IP,J)-Z(I,J))+U(I,J)*DELT1

C      LÍMITES PUESTOS EN A,B,C

IF(IVEL(IP,J).NE.4) GO TO 2000
CALL CHBO1(IBV(IP,J),-1,I,DIPJ/DIJ)
GO TO 2099
2000 IF(IVEL(I,J).NE.2) GO TO 2099
CALL CHBO1(IBV(I,J),1,I,DIJ/DIPJ)

2099 CONTINUE
101  CONTINUE

```

```

C =====
C =
C =  MATRIZ SOLUCIÓN PARA CONSEGUIR VELOCIDADES "U" =
C =
C =====
CALL TRIDAG(2,IMAXP2)
DO 102 I=2,IMAXP2
102 UPP(I,J)=R(I)
100 CONTINUE

C      CALCULE TODAS LAS VELOCIDADES LÍMITES

      DO 300 I=3,IMAXP2
      DO 300 J=3,JMAXP2
      IV=IVEL(I,J)+1
      GO TO (300,300,5001,300,5002),IV
5001 CALL BNDEXV(IBV(I,J),UPP(I+1,J),UPP(I,J),UPP(I-1,J)
e      ,DPP(I,J)/DPP(I+1,J))
      GO TO 300
5002 CALL BNDEXV(IBV(I,J),UPP(I-2,J),UPP(I-1,J),UPP(I,J)
e      ,DPP(I,J)/DPP(I-1,J))
300 CONTINUE

      RETURN
      END

SUBROUTINE SOLVV
C =====
C =
C =      SOLUCIÓN PARA VELOCIDADES "Y" A =
C =      TRAVÉS DE 1 PASO DE TIEMPO COMPLETO =
C =
C =
C =====
IMPLICIT REAL*8 (A-H,O-Z)
REAL*8 VAR(138,138)
INTEGER*4 AORC
INTEGER*4
I,J,K,JK,ITYP,IVEL,IBV,IBD,IMAX,JMAX,IMAXP1,JMAXP1
e      IMAXP2,JMAXP2,IMAXP3,JMAXP3,IMAXP4,JMAXP4
COMMON/CELL/DX(138),CDX(138),DY(138),CDY(138)

```

```

COMMON/BOUND/ITYP(138,138),IVEL(138,138),IBV(138,138)
  e      ,IBD(138,138)
COMMON/VELX/U(138,138),UPP(138,138)
COMMON/VELY/V(138,138),VPP(138,138)
COMMON/DEP/D(138,138),DP(138,138),DPP(138,138)
COMMON/TDAG/A(138),B(138),C(138),R(138)
COMMON/FLOW/Q(138,138)
COMMON/CONS1/DELT,G,CHEZY,GAM,NOIT1,NOIT2,NTIME
                                     e      ,NWRITE
COMMON/CONS2/DELT1,DELT2,GC2,SLIP
COMMON/CONU1/GDELX(138),GDT2DX(138),GDTDXP(138)
  e      ,GDTDXM(138)
COMMON/CONU2/GMDXPU(138),GMDXMU(138),GMDYMU(138)
  e      ,GMDYPU(138)
COMMON/CONV1/GDELY(138),GDT2DY(138),GDTDYP(138)
  e      ,GDTDYM(138)
COMMON/CONV2/GMDYPV(138),GMDYMV(138),GMDXPV(138)
  e      ,GMDXMV(138)
COMMON/VARZ1/ZDIMJ(138,138),ZDIJ1(138,138),ZDIPJ(138
,138)
  e      ,ZRHS(138,138)
COMMON/VARZ2/ZDIJM(138,138),ZDIJ2(138,138),ZDIJP(138
,138)
COMMON/SIZE0/IMAX,JMAX
COMMON/SIZE1/IMAXP1,JMAXP1
COMMON/SIZE2/IMAXP2,JMAXP2
COMMON/SIZE3/IMAXP3,JMAXP3
COMMON/SIZE4/IMAXP4,JMAXP4
COMMON/ELEV/Z(138,138),ZPP(138,138)

C  =====
C  =
C  =      COMIENCA BARRIDO EN DIRECCIÓN "Y"      =
C  =
C  =====
DO 100 I=3,IMAXP2
IP=I+1
IM=I-1
DXI=DX(I)
DO 101 J=2,JMAXP2

```

C            VARIABLES PARA VELOCIDADES DE RESBALÓN

```
CDYJ=CDY(J)
JP=J+1
JM=J-1
VIJM=VPP(I,JM)
VIJ=VPP(I,J)
VIJP=VPP(I,JP)
VIMJ=VPP(IM,J)
VIPJ=VPP(IP,J)
UIJ=UPP(I,J)
UIJP=UPP(I,JP)
UIMJ=UPP(IM,J)
UIMJP=UPP(IM,JP)
```

C            CALCULE TODAS LAS VELOCIDADES DE RESBALÓN

```
ITYPP=ITYP(I,J)+1
GO TO (1000,1001,1002,1003,1099,1004,1002,1003,1004
e      ,1002),ITYPP
```

C            ITYP=0

```
1000 IF(IVEL(I,JP).EQ.3) GO TO 1010
      CALL SPEC(J,0)
      GO TO 2099
1010 IF(IBV(I,JP).EQ.1) GO TO 1011
      UIJ=SLIP*UIJP
      UIMJ=SLIP*UIMJP
      IF(IVEL(IP,JP).NE.3) VIPJ=SLIP*VIJ
      IF(IVEL(IM,JP).NE.3) VIMJ=SLIP*VIJ
      GO TO 1099
1011 CALL CALFLO(Q(I,JP),IVEL(I,JP),VEL,I,JP)
      CALL SPEC(J,VEL)
      GO TO 2099
```

C            ITYP=1

```
1001 ITYPP=ITYP(I,JP)
      GO TO (1099,1099,1020,1099,1021,1020,1099,1099
e      ,1021),ITYPP
1020 UIJP=SLIP*UIJ
      VIPJ=SLIP*VIJ
```

```

        GO TO 1099
1021 UIMJP=SLIP*UIMJ
      VIMJ=SLIP*VIJ
        GO TO 1099

C      ITYP=2,6,9

1002 IF(IVEL(I,J).EQ.1) GO TO 1050
      CALL SPEC(J,0)
        GO TO 2099
1050 IF(IBV(I,J).EQ.1) GO TO 1051
      UIMJP=SLIP*UIMJ
      UIJP=SLIP*UIJ
      IF(IVEL(IM,J).NE.1) VIMJ=SLIP*VIJ
      IF(IVEL(IP,J).NE.1) VIPJ=SLIP*VIJ
        GO TO 1099
1051 CALL CALFLO(Q(I,J),IVEL(I,J),VEL,I,J)
      CALL SPEC(J,VEL)
        GO TO 2099

C      ITYP=3,7

1003 VIPJ=SLIP*VIJ
      IF(IVEL(I,J).EQ.2.AND.IVEL(I,JP).EQ.2) GO TO 1099
      UIJ=SLIP*UIJP
        GO TO 1099

C      ITYP=5,8

1004 VIMJ=SLIP*VIJ
      IF(IVEL(I,J).EQ.4.AND.IVEL(I,JP).EQ.4) GO TO 1099
      UIMJ=SLIP*UIMJP

1099 CONTINUE

C      VARIABLES PARA PROFUNDIDADES

      DIJ=DPP(I,J)
      DIMJ=DPP(IM,J)
      DIPJ=DPP(IP,J)
      DIJM=DPP(I,JM)
      DIJP=DPP(I,JP)
      DIMJP=DPP(IM,JP)

```



```

DIPJP=DPP(IP,JP)
DIJP2=DPP(I,J+2)

C      CALCULAR AGUAS ARRIBAS Y TÉRMINOS DE DIFERENCIACIÓN
CENTRAL

CALL UPSTR(VIJM+VIJ,VNB,VPB)
CALL UPSTR(VIJ+VIJP,VNT,VPT)
CALL CENTR(DIJP,DIJP2,DIJPV)
CALL CENTR(DIJ,DIJP,DIJV)
CALL CENTR(DIJM,DIJ,DIJMV)
CALL CENTR(DIJP,DIPJP,DIJPU)
CALL CENTR(DIMJP,DIJP,DIMJPU)
CALL CENTR(DIJ,DIPJ,DIJU)
CALL CENTR(DIMJ,DIJ,DIMJU)
CALL CENTR(UIMJ,UIMJP,UIMJV)
CALL CENTR(UIJ,UIJP,UIJV)

C      =====
C      =
C      =      CALCULE COEFICIENTES PARA INTRODUCCIÓN DE LA      =
C      =      MATRIZ                                          =
C      =
C      =====

A(J)=-GDTDYM(J)*DIJMV-VPB*VIJM/CDYJ-GMDYMV(J)
B(J)=DELTA1+(GDTDYM(J)+GDTDYP(J))*DIJV+(VPT-VNB)*VIJ
e      /CDYJ+(GMDYMV(J)+GMDYPV(J))+GC2/DIJV*(VIJ*VIJ
e      +(UIMJV+UIJV)/2.0)**2)**(1/2)
C(J)=-GDTDYP(J)*DIJPV+VNT*VIJP/CDYJ-GMDYPV(J)

C

R(J) =VIMJ*(-UIMJV/2.0/DXI-GMDXMV(I))
e      +VIJ*((UIJV-UIMJV)/2/DXI+(GMDXMV(I)+
e      GMDXPV(I))+VIPJ*(UIJV/2/DXI-GMDXPV(I))
e      -GDT2DY(J)/DXI*(UPP(IM,JP)*DIMJPU-UPP(I,JP)
e      *DIJPU+UPP(I,J)*DIJU-UPP(IM,J)*DIMJU)
e      -GDELY(J)*(Z(I,JP)-Z(I,J))+V(I,J)*DELTA1

C      LÍMITES PUESTOS EN A,B,C

IF (IVEL(I,JP).NE.3) GO TO 2000
CALL CHBO1(IBV(I,JP),-1,J,DIJP/DIJ)
GO TO 2099

```

```

2000 IF(IVEL(I,J).NE.1) GO TO 2099
      CALL CHBO1 (IBV(I,J),1,J,DIJ/DIJP)

2099 CONTINUE
101  CONTINUE

C      =====
C      =
C      =  MATRIZ SOLUCIÓN PARA CONSEGUIR VELOCIDADES "V" =
C      =
C      =====
      CALL TRIDAG (2,JMAXP2)
      DO 102 J=2,JMAXP2
102  VPP(I,J)=R(J)
100  CONTINUE

C          CALCULE TODAS LAS VELOCIDADES LÍMITES

      DO 300 J=3,JMAXP2
      DO 300 I=3,IMAXP2
      IV=IVEL(I,J)+1
      GO TO (300,5001,300,5002,300),IV
5001 CALL BNDEXV(IBV(I,J),VPP(I,J+1),VPP(I,J),VPP(I,J-1)
      e          ,DPP(I,J)/DPP(I,J+1))
      GO TO 300
5002 CALL BNDEXV(IBV(I,J),VPP(I,J-2),VPP(I,J-1),VPP(I,J)
      e          ,DPP(I,J)/DPP(I,J-1))
300  CONTINUE

      RETURN
      END

      SUBROUTINE SOLVD
C      =====
C      =
C      =  SOLUCIONES PARA LAS PROFUNDIDADES USANDO 2 =
C      =  PASOS DE TIEMPO =
C      =
C      =====
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 VAR(138,138)

```

```

      INTEGER*4 AORC

      INTEGER*4
      I , J , K , JJ , ITYP , IVEL , IBV , IBD , IMAX , JMAX , IMAXP1 , JMAXP1
      e  IMAXP2 , JMAXP2 , IMAXP3 , JMAXP3 , IMAXP4 , JMAXP4
      COMMON/CELL/DX(138) , CDX(138) , DY(138) , CDY(138)

COMMON/BOUND/ITYP(138,138) , IVEL(138,138) , IBV(138,138)
      e      , IBD(138,138)
      COMMON/VELX/U(138,138) , UPP(138,138)
COMMON/VELY/V(138,138) , VPP(138,138)
COMMON/DEP/D(138,138) , DP(138,138) , DPP(138,138)
COMMON/TDAG/A(138) , B(138) , C(138) , R(138)
COMMON/FLOW/Q(138,138)
COMMON/CONS1/DELT , G , CHEZY , GAM , NOIT1 , NOIT2 , NTIME
      e      , NWRITE
COMMON/CONS2/DELT1 , DELT2 , GC2 , SLIP
COMMON/CONU1/GDELX(138) , GDT2DX(138) , GDTDXP(138)
      e      , GDTDXM(138)
COMMON/CONU2/GMDXPU(138) , GMDXMU(138) , GMDYMU(138)
      e      , GMDYPU(138)
COMMON/CONV1/GDELY(138) , GDT2DY(138) , GDTDYP(138)
      e      , GDTDYM(138)
COMMON/CONV2/GMDYPV(138) , GMDYMV(138) , GMDXPV(138)
      e      , GMDXMV(138)
COMMON/VARZ1/ZDIMJ(138,138) , ZDIJ1(138,138) , ZDIPJ(138
,138)
      e      , ZRHS(138,138)
COMMON/VARZ2/ZDIJM(138,138) , ZDIJ2(138,138) , ZDIJP(138
,138)
COMMON/SIZE0/IMAX , JMAX
COMMON/SIZE1/IMAXP1 , JMAXP1
COMMON/SIZE2/IMAXP2 , JMAXP2
COMMON/SIZE3/IMAXP3 , JMAXP3
COMMON/SIZE4/IMAXP4 , JMAXP4
COMMON/ELEV/Z(138,138) , ZPP(138,138)

C  =====
C  =
C  =      COMIENGE EL PRIMER BARRIDO (DIRECCIÓN - X)      =
C  =
C  =====

```

```

DO I1=1,IMAXP4
DO J1=1,JMAXP4
WRITE(10,*)I1,J1,ZPP(I1,J1)
ENDDO
ENDDO

```

```

DO 100 J=3,JMAXP2
DYR=DY(J)

```

```

JP=J+1
JM=J-1

```

```

DO 101 I=3,IMAXP2
DXR=DX(I)

```

```

IM=I-1
IP=I+1

```

C                   NO CALCULE SI EL LÍMITE ESTÁ FUERA DEL EMBALSE

```

IF(ITYP(I,J).NE.0) GO TO 100
CALL SPEC(I,-5000)
GO TO 101

```

1000 CONTINUE

C                   CALCULAR AGUAS ARRIBA LOS PARÁMETROS

```

CALL UPSTR(UPP(IM,J),UNL,UPL)
CALL UPSTR(UPP(I,J),UNR,UPR)
CALL UPSTR(VPP(I,JM),VNB,VPB)
CALL UPSTR(VPP(I,J),VNT,VPT)

```

C                   CALCULE COEFICIENTES PARA AMBOS BARRIDOS

```

ZDIMJ(I,J)=-UPP(IM,J)*UPL/DXR
ZDIJ1(I,J)=(UPP(I,J)*UPR-UPP(IM,J)*UNL)/DXR
ZDIPJ(I,J)=UPP(I,J)*UNR/DXR
ZDIJM(I,J)=-VPP(I,JM)*VPB/DYR
ZDIJ2(I,J)=(VPP(I,J)*VPT-VPP(I,JM)*VNB)/DYR
ZDIJP(I,J)=VPP(I,J)*VNT/DYR

```

```

C =====
C =
C = CALCULE A,B,C,R PARA INTRODUCCIÓN DE LA MATRIZ =
C =
C =====

```

```

A(I)=ZDIMJ(I,J)
B(I)=DELT2+ZDIJ1(I,J)
C(I)=ZDIPJ(I,J)
R(I)=DELT2*D(I,J)-ZDIJM(I,J)*DPP(I,JM)
e   -ZDIJ2(I,J)*DPP(I,J)-ZDIJP(I,J)*DPP(I,JP)

```

```

C           LÍMITES PUESTOS EN A,B,C

```

```

IV=IVEL(I,J)+1
GO TO (101,101,1001,101,1002),IV
1001 CALL CHBO2(IBD(I,J),1,I,D(IP,J))

```

```

GO TO 101
1002 CALL CHBO2(IBD(I,J),-1,I,D(IM,J))
101 CONTINUE

```

```

C           MATRIZ SOLUCIÓN PARA CONSEGUIR PROFUNDIDADES
INTERMEDIAS

```

```

CALL TRIDAG(3,IMAXP2)
DO 102 I=3,IMAXP2
102 DP(I,J)=R(I)
100 CONTINUE

```

```

C           CALCULE PROFUNDIDADES LÍMITES (EN ENTRADAS Y
SALIDAS)

```

```

DO 300 I=3,IMAXP2
DO 300 J=3,JMAXP2
IV=IVEL(I,J)+1
GO TO (300,300,301,300,302),IV
301 CALL BNDEXD(IBD(I,J),D(I+1,J),DP(I+1,J),DP(I,J)
e           ,DP(I-1,J))
GO TO 300
302 CALL BNDEXD(IBD(I,J),D(I-1,J),DP(I-1,J),DP(I,J)

```

```

          e      ,DP(I+1,J))
300  CONTINUE

C  =====
C  =
C  =   COMIENZE EL SEGUNDO BARRIDO (DIRECCIÓN - Y)   =
C  =
C  =====

C  DO 200 I=3,IMAXP2
C  DO 201 J=3,JMAXP2

C          NO CALCULE SI EL LÍMITE ESTÁ FUERA DEL
EMBALSE

          IF(ITYP(I,J).NE.0) GO TO 2000
          CALL SPEC(J,-5000.0)
          GO TO 201
2000 CONTINUE

C  =====
C  =
C  =   CALCULE A,B,C,R PARA INTRODUCCIÓN DE LA MATRIZ   =
C  =
C  =====
A(J)=ZDIJM(I,J)
B(J)=DELT2+ZDIJ2(I,J)
C(J)=ZDIJP(I,J)
R(J)=DELT2*DP(I,J)-ZDIMJ(I,J)*DP(I-1,J)
e      -ZDIJ1(I,J)*DP(I,J)-ZDIPJ(I,J)*DP(I+1,J)

C          LÍMITES PUESTOS EN A,B,C

          IV=IVEL(I,J)+1
          GO TO (201,2001,201,2002,201),IV
2001 CALL CHBO2(IBD(I,J),1,J,D(I,J+1))
          GO TO 201
2002 CALL CHBO2(IBD(I,J),-1,J,D(I,J-1))
201  CONTINUE

C          MATRIZ SOLUCIÓN PARA OBTENER PROFUNDIDADES FINALES

```

```

        CALL TRIDAG(3,JMAXP2)
        DO 202 J=3,JMAXP2
202    DPP(I,J)=R(J)
200    CONTINUE

C          CALCULE PROFUNDIDADES LÍMITES

        DO 400 I=3,IMAXP2
        DO 400 J=3,JMAXP2
        IV=IVEL(I,J)+1
        GO TO (400,401,402,403,404),IV
401    CALL BNDEXD(IBD(I,J),D(I,J+1),DPP(I,J+1),DPP(I,J)
        e          ,DPP(I,J-1))
        CALL BNDEXD(IBD(I,J),D(I,J+2),DPP(I,J+2),DPP(I,J+1)
        e          ,DPP(I,J))
        GO TO 400
402    CALL BNDEXD(IBD(I,J),D(I+1,J),DPP(I+1,J),DPP(I,J)
        e          ,DPP(I-1,J))
        CALL BNDEXD(IBD(I,J),D(I+2,J),DPP(I+2,J),DPP(I+1,J)
        e          ,DPP(I,J))
        GO TO 400
403    CALL BNDEXD(IBD(I,J),D(I,J-1),DPP(I,J-1),DPP(I,J)
        e          ,DPP(I,J+1))
        CALL BNDEXD(IBD(I,J),D(I,J-2),DPP(I,J-2),DPP(I,J-1)
        e          ,DPP(I,J))
        GO TO 400
404    CALL BNDEXD(IBD(I,J),D(I-1,J),DPP(I-1,J),DPP(I,J)
        e          ,DPP(I+1,J))
        CALL BNDEXD(IBD(I,J),D(I-2,J),DPP(I-2,J),DPP(I-1,J)
        e          ,DPP(I,J))
400    CONTINUE

C          CALCULE ELEVACIONES Z

        DO 500 I=1,IMAXP4
        DO 500 J=1,JMAXP4
500    ZPP(I,J)=DPP(I,J)+Z(I,J)-D(I,J)

        RETURN
        END

```

```

SUBROUTINE CHBO1( IBOUN, AORC, IJ, DEP)
C =====
C =
C =      CALCULE A, B, C PARA LÍMITES DE VELOCIDAD      =
C =
C =====
IMPLICIT REAL*8 (A-H, O-Z)
REAL*8 VAR(138, 138)
INTEGER*4 AORC
INTEGER*4
I, J, K, JJ, ITYP, IVEL, IBV, IBD, IMAX, JMAX, IMAXP1, JMAXP1
e      IMAXP2, JMAXP2, IMAXP3, JMAXP3, IMAXP4, JMAXP4
COMMON/CELL/DX(138), CDX(138), DY(138), CDY(138)

COMMON/BOUND/ITYP(138, 138), IVEL(138, 138), IBV(138, 138)
e      , IBD(138, 138)
COMMON/VELX/U(138, 138), UPP(138, 138)
COMMON/VELY/V(138, 138), VPP(138, 138)
COMMON/DEP/D(138, 138), DP(138, 138), DPP(138, 138)
COMMON/TDAG/A(138), B(138), C(138), R(138)
COMMON/FLOW/Q(138, 138)
COMMON/CONS1/DELT, G, CHEZY, GAM, NOIT1, NOIT2, NTIME
e      , NWRITE
COMMON/CONS2/DELT1, DELT2, GC2, SLIP
COMMON/CONU1/GDELX(138), GDT2DX(138), GDTDXP(138)
e      , GDTDXM(138)
COMMON/CONU2/GMDXPU(138), GMDXMU(138), GMDYMU(138)
e      , GMDYPU(138)
COMMON/CONV1/GDELY(138), GDT2DY(138), GDTDYP(138)
e      , GDTDYM(138)
COMMON/CONV2/GMDYPV(138), GMDYMV(138), GMDXPV(138)
e      , GMDXMV(138)
COMMON/VARZ1/ZDIMJ(138, 138), ZDIJ1(138, 138), ZDIPJ(138
, 138)
e      , ZRHS(138, 138)
COMMON/VARZ2/ZDIJM(138, 138), ZDIJ2(138, 138), ZDIJP(138
, 138)
COMMON/SIZE0/IMAX, JMAX
COMMON/SIZE1/IMAXP1, JMAXP1
COMMON/SIZE2/IMAXP2, JMAXP2
COMMON/SIZE3/IMAXP3, JMAXP3
COMMON/SIZE4/IMAXP4, JMAXP4
COMMON/ELEV/Z(138, 138), ZPP(138, 138)

```



```

        IF(AORC.EQ.-1) GO TO 1000
        IF(IBOUN.EQ.2) GO TO 1001
        B(IJ)=B(IJ)+2*C(IJ)
        A(IJ)=A(IJ)-C(IJ)
        GO TO 1002
1001 B(IJ)=B(IJ)+C(IJ)*DEP
1002 C(IJ)=0
        RETURN
1000 IF(IBOUN.EQ.2) GO TO 1004
        B(IJ)=B(IJ)+2*A(IJ)
        C(IJ)=C(IJ)-A(IJ)
        GO TO 1005
1004 B(IJ)=B(IJ)+A(IJ)*DEP
1005 A(IJ)=0
        RETURN

```

```

        ENTRY CHBO2(IBOUN,AORC,IJ,DEP)

```

```

C      =====
C      =
C      =      CALCULE A,B,C,R PARA LÍMITES DE PROFUNDIDAD =
C      =
C      =====
        IF(AORC.EQ.-1) GO TO 3000
        GO TO (3001,3002,3003)IBOUN
3001 R(IJ)=R(IJ)-C(IJ)*DEP
        GO TO 3004
3002 B(IJ)=B(IJ)+C(IJ)
        GO TO 3004
3003 B(IJ)=B(IJ)+2*C(IJ)
        A(IJ)=A(IJ)-C(IJ)
3004 C(IJ)=0
        RETURN

3000 GO TO (4001,4002,4003)IBOUN
4001 R(IJ)=R(IJ)-A(IJ)*DEP
        GO TO 4004
4002 B(IJ)=B(IJ)+A(IJ)
        GO TO 4004
4003 B(IJ)=B(IJ)+2*A(IJ)
        C(IJ)=C(IJ)-A(IJ)
4004 A(IJ)=0
        RETURN

```

```

      ENTRY SPEC(I,RHS)
C =====
C =
C = SOLUCIONES A,B,C,R PARA CASOS ESPECIFICADOS =
C =
C =====
      A(I)=0
      B(I)=1
      C(I)=0
      R(I)=RHS

      RETURN

      ENTRY BNDEXV( IBOUN,T1,T2,T3,DEPT)
C =====
C =
C = SOLUCIONES PARA VELOCIDADES LÍMITES EXTERNAS =
C =
C =====
      GO TO (5000,5001,5002),IBOUN
5000 T1=T2
      RETURN
5001 T1=T2*DEPT
      RETURN
5002 T1=2*T2-T3
      RETURN

      ENTRY BNDEXD( IBOUN,T1,T2,T3,T4)
C =====
C =
C = SOLUCIONES PARA PROFUNDIDADES LÍMITES EXTERNAS =
C =
C =====
      GO TO (2000,2001,2002)IBOUN
2000 T2=T1
      RETURN
2001 T2=T3
      RETURN
2002 T2=2*T3-T4
      RETURN

```

```

        ENTRY UPSTR(U1,VN,VP)
C      =====
C      =
C      =      SOLUCIONES PARA TÉRMINOS DE DIFERENCIACIÓN      =
C      =      AGUAS ARRIBA                                     =
C      =
C      =====
        VN=0
        VP=0
        IF (U1) 6000,6001,6002
6000  VN=1
        RETURN
6002  VP=1
6001  RETURN

        ENTRY CENTR(DL,DR,DM)
C      =====
C      =
C      =      SOLUCIONES PARA TÉRMINOS DE DIFERENCIACIÓN      =
C      =      CENTRAL                                           =
C      =
C      =====
        DM=(DL+DR)/2
        RETURN

        ENTRY TRIDAG(NL,NR)
C      =====
C      =
C      =      SOLVES A TRIDIAGONAL MATRIX                      =
C      =      RETURNS SOLUTION IN 'R'                          =
C      =
C      =====

        NM=NR-1

        DO 100 I3=NL,NM

        C(I3)=C(I3)/B(I3)
        A(I3+1)=A(I3+1)/B(I3+1)
100   R(I3)=R(I3)/B(I3)
        R(NR)=R(NR)/B(NR)
        NP=NL+1
        DO 101 I2=NP,NR

```

```

B(I2)=1-A(I2)*C(I2-1)
R(I2)=(R(I2)-A(I2)*R(I2-1))/B(I2)
IF(I2.NE.NR) C(I2)=C(I2)/B(I2)
101 CONTINUE

```

```

DO 102 II4=NL,NM
I4=NM-II4+NL
R(I4)=R(I4)-C(I4)*R(I4+1)
102 CONTINUE
RETURN
END

```

```

SUBROUTINE CALFLO(FLOW,IV,VEL,I,J)
C =====
C =
C = CALCULE VELOCIDAD LÍMITE BASADA EN FLUJO =
C =
C =====
IMPLICIT REAL*8 (A-H,O-Z)
REAL*8 VAR(138,138)
INTEGER*4 AORC
INTEGER*4
I,J,K,JJ,ITYP,IVEL,IBV,IBD,IMAX,JMAX,IMAXP1,JMAXP1
e IMAXP2,JMAXP2,IMAXP3,JMAXP3,IMAXP4,JMAXP4
COMMON/CELL/DX(138),CDX(138),DY(138),CDY(138)

COMMON/BOUND/ITYP(138,138),IVEL(138,138),IBV(138,138)
e ,IBD(138,138)
COMMON/VELX/U(138,138),UPP(138,138)
COMMON/VELY/V(138,138),VPP(138,138)
COMMON/DEP/D(138,138),DP(138,138),DPP(138,138)
COMMON/TDAG/A(138),B(138),C(138),R(138)
COMMON/FLOW/Q(138,138)
COMMON/CONS1/DELT,G,CHEZY,GAM,NOIT1,NOIT2,NTIME
e ,NWRITE
COMMON/CONS2/DELT1,DELT2,GC2,SLIP
COMMON/CONU1/GDELX(138),GDT2DX(138),GDTDXP(138)
e ,GDTDXM(138)
COMMON/CONU2/GMDXPU(138),GMDXMU(138),GMDYMU(138)
e ,GMDYPU(138)
COMMON/CONV1/GDELY(138),GDT2DY(138),GDTDYP(138)
e ,GDTDYM(138)

```

```

COMMON/CONV2/GMDYPV(138),GMDYMV(138),GMDXPV(138)
e          ,GMDXMV(138)
COMMON/VARZ1/ZDIMJ(138,138),ZDIJ1(138,138),ZDIPJ(138
,138)
e          ,ZRHS(138,138)
COMMON/VARZ2/ZDIJM(138,138),ZDIJ2(138,138),ZDIJP(138
,138)
COMMON/SIZE0/IMAX,JMAX
COMMON/SIZE1/IMAXP1,JMAXP1
COMMON/SIZE2/IMAXP2,JMAXP2
COMMON/SIZE3/IMAXP3,JMAXP3
COMMON/SIZE4/IMAXP4,JMAXP4
COMMON/ELEV/Z(138,138),ZPP(138,138)

IVE=IV+1
GO TO (1000,1001,1002,1003,1004)IVE
1001 VEL=FLOW/DX(I)/(D(I,J)+D(I,J+1))*2
RETURN
1002 VEL=FLOW/DY(J)/(D(I,J)+D(I+1,J))*2
RETURN
1003 VEL=FLOW/DX(I)/(D(I,J)+D(I,J-1))*2
RETURN
1004 VEL=FLOW/DY(J)/(D(I,J)+D(I-1,J))*2
1000 RETURN
END

SUBROUTINE WRITEM(VAR,IL,IB,JL,JB)
C =====
C =
C = ESCRIBE UNA SOLA MATRIX =
C =
C =====

REAL*8 VAR(138,138)
DO 100 JJ=JL,JB
J=JB-JJ+JL
WRITE(6,888)
WRITE(8,888)
888 FORMAT(' ')
WRITE(8,7000) (VAR(I,J),I=IL,IB)
100 WRITE(6,7000) (VAR(I,J),I=IL,IB)
7000 format(13f6.2)

```

```

RETURN
END
SUBROUTINE WRITER(U,V,Z)
C =====
C =
C =     ESCRIBE RESULTADOS CON TÍTULOS     =
C =
C =====
IMPLICIT REAL*8 (A-H,O-Z)
REAL*8 VAR(138,138)
INTEGER*4 AORC
INTEGER*4
I,J,K, JJ, ITYP, IVEL, IBV, IBD, IMAX, JMAX, IMAXP1, JMAXP1
e          IMAXP2, JMAXP2, IMAXP3, JMAXP3, IMAXP4, JMAXP4
COMMON/CELL/DX(138), CDX(138), DY(138), CDY(138)

COMMON/BOUND/ITYP(138,138), IVEL(138,138), IBV(138,138)
e          , IBD(138,138)
COMMON/DEP/D(138,138), DP(138,138), DPP(138,138)
COMMON/TDAG/A(138), B(138), C(138), R(138)
COMMON/FLOW/Q(138,138)
COMMON/CONS1/DELT,G,CHEZY,GAM,NOIT1,NOIT2,NTIME
e          ,NWRITE
COMMON/CONS2/DELT1,DELT2,GC2,SLIP
COMMON/CONU1/GDELX(138),GDT2DX(138),GDTDXP(138)
e          ,GDTDXM(138)
COMMON/CONU2/GMDXPU(138),GMDXMU(138),GMDYMU(138)
e          ,GMDYPU(138)
COMMON/CONV1/GDELY(138),GDT2DY(138),GDTDYP(138)
e          ,GDTDYM(138)
COMMON/CONV2/GMDYPV(138),GMDYMV(138),GMDXPV(138)
e          ,GMDXMV(138)
COMMON/VARZ1/ZDIMJ(138,138),ZDIJ1(138,138),ZDIPJ(138
,138)
e          ,ZRHS(138,138)
COMMON/VARZ2/ZDIJM(138,138),ZDIJ2(138,138),ZDIJP(138
,138)
COMMON/SIZE0/IMAX,JMAX
COMMON/SIZE1/IMAXP1,JMAXP1
COMMON/SIZE2/IMAXP2,JMAXP2
COMMON/SIZE3/IMAXP3,JMAXP3
COMMON/SIZE4/IMAXP4,JMAXP4

```

```

        WRITE(6,6000)
        WRITE(8,*)( '      ' )
        WRITE(8,6000)
6000 FORMAT(' VELOCIDADES EN X',/)
        CALL WRITEM(U,2,IMAXP2,3,JMAXP2)
        WRITE(6,6001)
        WRITE(8,6001)
6001 FORMAT(//,'VELOCIDAES EN Y',/)
        CALL WRITEM(V,3,IMAXP2,2,JMAXP2)
        WRITE(6,6002)
        WRITE(8,6002)
6002 FORMAT(//,'ELEVACIONES Z',/)
        CALL WRITEM(Z,2,IMAXP3,2,JMAXP3)
        WRITE(6,999)
        WRITE(8,999)
999  FORMAT('1')
        RETURN
        END

```

SUBROUTINE UPDATE

```

C  =====
C  =
C  = VARIABLES DE ACTUALIZACIONES DURANTE EL TIEMPO =
C  =
C  =====
        IMPLICIT REAL*8 (A-H,O-Z)
        real*8 var(138,138)
        INTEGER*4 AORC
        INTEGER*4
I,J,K, JJ, ITYP, IVEL, IBV, IBD, IMAX, JMAX, IMAXP1, JMAXP1
        e          IMAXP2, JMAXP2, IMAXP3, JMAXP3, IMAXP4, JMAXP4
        COMMON/CELL/DX(138),CDX(138),DY(138),CDY(138)

COMMON/BOUND/ITYP(138,138),IVEL(138,138),IBV(138,138)
        e          ,IBD(138,138)
        COMMON/VELX/U(138,138),UPP(138,138)
        COMMON/VELY/V(138,138),VPP(138,138)
        COMMON/DEP/D(138,138),DP(138,138),DPP(138,138)
        COMMON/TDAG/A(138),B(138),C(138),R(138)
        COMMON/FLOW/Q(138,138)
        COMMON/CONS1/DELT,G,CHEZY,GAM,NOIT1,NOIT2,NTIME
        e          ,NWRITE
        COMMON/CONS2/DELT1,DELT2,GC2,SLIP

```

```

COMMON/CONU1/GDELX(138),GDT2DX(138),GDTDXP(138)
e          ,GDTDXM(138)
COMMON/CONU2/GMDXPU(138),GMDXMU(138),GMDYMU(138)
e          ,GMDYPU(138)
COMMON/CONV1/GDELY(138),GDT2DY(138),GDTDYP(138)
e          ,GDTDYM(138)
COMMON/CONV2/GMDYPV(138),GMDYMV(138),GMDXPV(138)
e          ,GMDXMV(138)
COMMON/VARZ1/ZDIMJ(138,138),ZDIJ1(138,138),ZDIPJ(138
,138)
e          ,ZRHS(138,138)
COMMON/VARZ2/ZDIJM(138,138),ZDIJ2(138,138),ZDIJP(138
,138)
COMMON/SIZE0/IMAX,JMAX
COMMON/SIZE1/IMAXP1,JMAXP1
COMMON/SIZE2/IMAXP2,JMAXP2
COMMON/SIZE3/IMAXP3,JMAXP3
COMMON/SIZE4/IMAXP4,JMAXP4
COMMON/ELEV/Z(138,138),ZPP(138,138)

DO 100 I=1,IMAXP3
DO 100 J=2,JMAXP3
100 U(I,J)=UPP(I,J)
DO 101 I=2,IMAXP3
DO 101 J=1,JMAXP3
101 V(I,J)=VPP(I,J)
DO 102 I=1,IMAXP4
DO 102 J=1,JMAXP4
D(I,J)=DPP(I,J)
102 Z(I,J)=ZPP(I,J)

RETURN
END

SUBROUTINE READIN
C =====
C =
C = LEE DATOS E INICIALIZA MATRICES =
C =
C =====
IMPLICIT REAL*8 (A-H,O-Z)
REAL*8 VAR(138,138)

```



```

      INTEGER*4
      I , J , K , JJ , ITYP , IVEL , IBV , IBD , IMAX , JMAX , IMAXP1 , JMAXP1
      e
      IMAXP2 , JMAXP2 , IMAXP3 , JMAXP3 , IMAXP4 , JMAXP4 , AORC
      COMMON/CELL/DX(138) , CDX(138) , DY(138) , CDY(138)

      COMMON/BOUND/ITYP(138,138) , IVEL(138,138) , IBV(138,138)
      e
      , IBD(138,138)
      COMMON/VELX/U(138,138) , UPP(138,138)
      COMMON/VELY/V(138,138) , VPP(138,138)
      COMMON/DEP/D(138,138) , DP(138,138) , DPP(138,138) , F(138
      ,138)
      COMMON/TDAG/A(138) , B(138) , C(138) , R(138)
      COMMON/FLOW/Q(138,138)
      COMMON/CONS1/DELT , G , CHEZY , GAM , NOIT1 , NOIT2 , NTIME
      e
      , NWRITE
      COMMON/CONS2/DELT1 , DELT2 , GC2 , SLIP
      COMMON/CONU1/GDELX(138) , GDT2DX(138) , GDTDXP(138)
      e
      , GDTDXM(138)
      COMMON/CONU2/GMDXPU(138) , GMDXMU(138) , GMDYMU(138)
      e
      , GMDYPU(138)
      COMMON/CONV1/GDELY(138) , GDT2DY(138) , GDTDYP(138)
      e
      , GDTDYM(138)
      COMMON/CONV2/GMDYPV(138) , GMDYMV(138) , GMDXPV(138)
      e
      , GMDXMV(138)
      COMMON/VARZ1/ZDIMJ(138,138) , ZDIJ1(138,138) , ZDIPJ(138
      ,138)
      e
      , ZRHS(138,138)
      COMMON/VARZ2/ZDIJM(138,138) , ZDIJ2(138,138) , ZDIJP(138
      ,138)
      COMMON/SIZE0/IMAX , JMAX
      COMMON/SIZE1/IMAXP1 , JMAXP1
      COMMON/SIZE2/IMAXP2 , JMAXP2
      COMMON/SIZE3/IMAXP3 , JMAXP3
      COMMON/SIZE4/IMAXP4 , JMAXP4
      COMMON/ELEV/Z(138,138) , ZPP(138,138)

```

```

C          LEA TIEMPO Y CONSTANTES

```

```

      READ(7,* ) DELT , NTIME , NWRITE
      WRITE(8,* ) ( ' ' )
      WRITE(6,6000) DELT , NTIME , NWRITE
      WRITE(8,6000) DELT , NTIME , NWRITE

```

```

6000 FORMAT(' DELT=',F10.5,/, ' N° DE ITERACIONES DE
TIEMPO='
e      ,I4,/, ' INTERVALO DE ESCRITURA=',I10,/)
READ(7,*) GAM,G,CHEZY,SLIP

WRITE(6,6001) GAM,G,CHEZY,SLIP
WRITE(8,6001) GAM,G,CHEZY,SLIP
6001 FORMAT(' GAMMA=',F10.1,/, ' G=',F10.1,/, '
CHEZY=',F10.1,/,
e      ' SLIP=',F10.1,/)
READ(7,*) NOIT1,NOIT2
WRITE(6,6002) NOIT1,NOIT2
WRITE(8,6002) NOIT1,NOIT2
6002 FORMAT(' N° DE ITERACIONES INICIALES=',I3,/,
e      ' N° DE ITERACIONES POR PASO DE
TIEMPO=',I3,/)
READ(7,*) IMAX,JMAX
WRITE(6,6003) IMAX,JMAX
WRITE(8,6003) IMAX,JMAX
6003 FORMAT(' IMAX=',I3,/, ' JMAX=',I3,/)

IMAXP1=IMAX+1
IMAXP2=IMAX+2
IMAXP3=IMAX+3
IMAXP4=IMAX+4
JMAXP1=JMAX+1
JMAXP2=JMAX+2
JMAXP3=JMAX+3
JMAXP4=JMAX+4

```

C            LEA DIMENSIONES DE CELDA

```

READ(7,*) (DX(I),I=3,IMAXP2)
READ(7,*) (DY(J),J=3,JMAXP2)
WRITE(6,6012)
WRITE(6,6004) (DX(I),I=3,IMAXP2)
WRITE(6,6012)
WRITE(6,6004) (DY(J),J=3,JMAXP2)
WRITE(8,6012)
WRITE(8,6004) (DX(I),I=3,IMAXP2)
WRITE(8,6012)

```

```

        WRITE(8,6004) (DY(J),J=3,JMAXP2)
6012 FORMAT(/,' DIMENSIONES DE LAS CELDAS= ',/)
6004 FORMAT(2X,10F6.0,/)

```

```

        DO 100 I=1,2
        DX(I)=DX(3)
100   DX(I+IMAXP2)=DX(IMAXP2)

```

```

        DO 101 J=1,2
        DY(J)=DY(3)
101   DY(J+JMAXP2)=DY(JMAXP2)

```

```

        DO 102 I=1,IMAXP3
102   CDX(I)=(DX(I)+DX(I+1))/2
        DO 103 J=1,JMAXP3
103   CDY(J)=(DY(J)+DY(J+1))/2

```

C            LEE TIPOS DE CELDA E INFORMACIÓN LÍMITE

```

        DO 105 I=1,IMAXP4
        DO 105 J=1,JMAXP4
105   IVEL(I,J)=0

```

```

        DO 104 JJ=3,JMAXP2
        J=JMAXP2-JJ+3
104   READ(7,*) (ITYP(I,J),I=3,IMAXP2)

```

```

        DO 106 I=1,IMAXP4
        DO 106 J=1,2
        ITYP(I,J)=0
        JJ=JMAXP2+J
106   ITYP(I,JJ)=0
        DO 107 J=3,JMAXP2
        DO 107 I=1,2
        ITYP(I,J)=0
        II=IMAXP2+I
107   ITYP(II,J)=0
        WRITE(6,6007)
        WRITE(8,6007)
        DO 108 JJ=3,JMAXP2
        J=JMAXP2-JJ+3

```

```

        WRITE(6,6006) (ITYP(I,J),I=3,IMAXP2)
108  WRITE(8,6006) (ITYP(I,J),I=3,IMAXP2)
        READ(7,*) NBOUN
        WRITE(6,6005)
        WRITE(8,6005)
        DO 120 KK=1,NBOUN
        READ(7,*) I,J,IVEL(I,J),IBV(I,J),IBD(I,J),Q(I,J)
        WRITE(6,*) I,J,IVEL(I,J),IBV(I,J),IBD(I,J),Q(I,J)
120  WRITE(8,*) I,J,IVEL(I,J),IBV(I,J),IBD(I,J),Q(I,J)

6005 FORMAT(//,' INFORMACION DE CONTORNO DE LA CELDA',/)
6006 FORMAT(40I5)
6007 FORMAT(//,' TIPO DE CELDA (ITYP(I,J))',/)

C      =====
C      =
C      =
C      =          LEA TODAS LAS PROFUNDIDADES          =
C      =
C      =
C      =====

        WRITE(6,6010)
        WRITE(8,6010)
        DO 111 JJ=3,JMAXP2
        WRITE(6,888)
        WRITE(8,888)
888  FORMAT('  ')
        J=JMAXP2-JJ+3
        READ(7,*) (D(I,J),I=3,IMAXP2)
        WRITE(8,6011) (D(I,J),I=3,IMAXP2)
111  WRITE(6,6011) (D(I,J),I=3,IMAXP2)
6010 FORMAT(//,' PROFUNDIDADES DE ENTRADA',/)
6011 FORMAT(10F5.1)

        DO I12=1,IMAXP4
        DO J12=1,JMAXP4
        F(I12,J12)=-1*D(I12,J12)
        WRITE(11,*)I12,J12,F(I12,J12)
        ENDDO
        ENDDO

```

```

DO 112 I=1,IMAXP4
DO 112 J=1,2
JJ=J+JMAXP2
D(I,J)=-1
112 D(I,JJ)=-1

DO 113 J=3,JMAXP2
DO 113 I=1,2
II=I+IMAXP2
D(I,J)=-1
113 D(II,J)=-1

WRITE(6,556)
WRITE(8,556)
556 FORMAT(//,' MATRIZ DE IVEL')
DO 668 J=3,JMAXP2
WRITE(6,886)
WRITE(8,886)
886 FORMAT(' ',/)
WRITE(8,6019) (IVEL(I,J),I=3,IMAXP2)
668 WRITE(6,6019) (IVEL(I,J),I=3,IMAXP2)
6019 FORMAT(10I5)

DO 114 I=3,IMAXP2
DO 114 J=3,JMAXP2
IVP=IVEL(I,J)+1
GO TO (114,1001,1002,1003,1004),IVP
1001 D(I,J+1)=D(I,J)
D(I,J+2)=D(I,J)
GO TO 114
1002 D(I+1,J)=D(I,J)
D(I+2,J)=D(I,J)
GO TO 114
1003 D(I,J-1)=D(I,J)
D(I,J-2)=D(I,J)
GO TO 114
1004 D(I-1,J)=D(I,J)
D(I-2,J)=D(I,J)

114 CONTINUE
READ(7,*)ELEVS
WRITE(6,6088)ELEVS
WRITE(8,6088)ELEVS

```

```

6088 FORMAT(//, ' ELEVACION DE SUPERFICIE LIBRE DE
      AGUA= ', F6.2)

      DO 110 I=1, IMAXP4
      DO 110 J=1, JMAXP4
110   Z(I, J)=ELEVS

      WRITE(6, 555)
      WRITE(8, 555)
555   FORMAT(/, 'ARREGLO TOTAL DE PROFUNDIDADES (D)', /)
      DO 666 J=1, JMAXP4
      WRITE(6, 881)
      WRITE(8, 881)
881   FORMAT('  ')
      WRITE(8, 6013) (D(I, J), I=1, IMAXP4)
666   WRITE(6, 6013) (D(I, J), I=1, IMAXP4)
6013  FORMAT(14F5.1, /)

C      CALCULA PROFUNDIDADES Y VELOCIDADES ESPECIFICADAS

      DO 115 I=1, IMAXP4
      DO 115 J=1, JMAXP4
      IF(D(I, J).GE.0) GO TO 119
      D(I, J)=-5000.0
      Z(I, J)=-5000.0

119   DP(I, J)=D(I, J)
      DPP(I, J)=D(I, J)
115   ZPP(I, J)=Z(I, J)

      WRITE(6, 505)
      WRITE(8, 505)
505   FORMAT(/, 'ARREGLO TOTAL DE PROFUNDIDADES (Z)', /)
      DO 9999 J=1, JMAXP4
      WRITE(6, 9998) (Z(I, J), I=1, IMAXP4)
9999  WRITE(8, 9998) (Z(I, J), I=1, IMAXP4)
9998  FORMAT(14f5.0)
      DO 116 I=1, IMAXP4
      DO 116 J=1, JMAXP4
      U(I, J)=0
116   V(I, J)=0
      DO 117 I=3, IMAXP2

```

```

DO 117 J=3,JMAXP2
IVP=IVEL(I,J)+1
GO TO (117,2001,2002,2003,2004),IVP
2001 V(I,J)=Q(I,J)/DX(I)/D(I,J)
V(I,J+1)=V(I,J)
GO TO 117
2002 U(I,J)=Q(I,J)/DY(J)/D(I,J)
U(I+1,J)=U(I,J)
GO TO 117
2003 V(I,J-1)=(Q(I,J)/DX(I))/D(I,J)
V(I,J-2)=V(I,J-1)
GO TO 117
2004 U(I-1,J)=(Q(I,J)/DY(J))/D(I,J)
U(I-2,J)=U(I-1,J)

117 CONTINUE
DO 118 I=1,IMAXP4
DO 118 J=1,JMAXP4
UPP(I,J)=U(I,J)
118 VPP(I,J)=V(I,J)

WRITE(6,700)
WRITE(8,700)
700 FORMAT(/,'MATRIZ INICIAL DE U',/)
DO 600 J=1,JMAXP4
WRITE(6,800)
WRITE(8,800)
800 FORMAT(' ')
WRITE(8,6099) (U(I,J),I=1,IMAXP4)
600 WRITE(6,6099) (U(I,J),I=1,IMAXP4)
6099 FORMAT(15F4.0)

RETURN
END

```

```

SUBROUTINE CALC1
C =====
C =
C = CALCULA TIEMPOS CONSTANTES INDEPENDIENTES =
C =
C =====

```

```

      IMPLICIT REAL*8 (A-H,O-Z)
      real*8 var(138,138)
      INTEGER*4
I ,J,K, JJ ,ITYP ,IVEL ,IBV ,IBD ,IMAX ,JMAX ,IMAXP1 ,JMAXP1
      e
IMAXP2 ,JMAXP2 ,IMAXP3 ,JMAXP3 ,IMAXP4 ,JMAXP4 , AORC
      COMMON/CELL/DX(138) ,CDX(138) ,DY(138) ,CDY(138)

COMMON/BOUND/ITYP(138,138) ,IVEL(138,138) ,IBV(138,138)
      e      ,IBD(138,138)
COMMON/VELX/U(138,138) ,UPP(138,138)
COMMON/VELY/V(138,138) ,VPP(138,138)
COMMON/DEP/D(138,138) ,DP(138,138) ,DPP(138,138)
COMMON/TDAG/A(138) ,B(138) ,C(138) ,R(138)
COMMON/FLOW/Q(138,138)
COMMON/CONS1/DELT,G,CHEZY ,GAM,NOIT1 ,NOIT2 ,NTIME
      e      ,NWRITE
COMMON/CONS2/DELT1 ,DELT2 ,GC2 ,SLIP
COMMON/CONU1/GDELX(138) ,GDT2DX(138) ,GDTDXP(138)
      e      ,GDTDXM(138)
COMMON/CONU2/GMDXPU(138) ,GMDXMU(138) ,GMDYMU(138)
      e      ,GMDYPU(138)
COMMON/CONV1/GDELY(138) ,GDT2DY(138) ,GDTDYP(138)
      e      ,GDTDYM(138)
COMMON/CONV2/GMDYPV(138) ,GMDYMV(138) ,GMDXPV(138)
      e      ,GMDXMV(138)
COMMON/VARZ1/ZDIMJ(138,138) ,ZDIJ1(138,138) ,ZDIPJ(138
,138)
      e      ,ZRHS(138,138)
COMMON/VARZ2/ZDIJM(138,138) ,ZDIJ2(138,138) ,ZDIJP(138
,138)
COMMON/SIZE0/IMAX ,JMAX
COMMON/SIZE1/IMAXP1 ,JMAXP1
COMMON/SIZE2/IMAXP2 ,JMAXP2
COMMON/SIZE3/IMAXP3 ,JMAXP3
COMMON/SIZE4/IMAXP4 ,JMAXP4
COMMON/ELEV/Z(138,138) ,ZPP(138,138)
WRITE(8,*>(' '))

GC2=(G/CHEZY)/CHEZY
DO 100 I=2 ,IMAXP2
GDELX(I)=G/CDX(I)
GMDXPU(I)=GAM/CDX(I)/DX(I+1)

```



```

100  GMDXMU(I)=GAM/CDX(I)/DX(I)

      DO 101 J=3,JMAXP2
      GMDYPU(J)=GAM/DY(J)/CDY(J)
101  GMDYMU(J)=GAM/DY(J)/CDY(J-1)

      DO 102 J=2,JMAXP2
      GDELY(J)=G/CDY(J)
      GMDYPV(J)=GAM/CDY(J)/DY(J+1)
102  GMDYMV(J)=GAM/CDY(J)/DY(J)

      DO 103 I=3,IMAXP2
      GMDXPV(I)=(GAM/DX(I))/CDX(I)
103  GMDXMV(I)=(GAM/DX(I))/CDX(I-1)

      RETURN

      ENTRY CALC2
C      =====
C      =
C      =      CALCULA TIEMPOS CONSTANTES DEPENDIENTES      =
C      =
C      =====
      DELT1=1/DELT
      DELT2=2/DELT

      DO 200 I=2, IMAXP2
      GDT2DX (I) =GDELX (I)*DELT
      GDTDXP (I) =GDT2DX (I)/DX (I+1)
      GDTDXM (I) =GDT2DX (I)/DX (I)
200  WRITE (6,*) GDT2DX (I), GDTDXP (I), GDTDXM (I)

      DO 201 J=2, JMAXP2
      GDT2DY (J) =GDELY (J)*DELT
      GDTDYP (J) =GDT2DY (J)/DY (J+1)
201  GDTDYM (J) =GDT2DY (J)/DY (J)

      RETURN
      END

```

## APÉNDICE C: MUESTRA DE ENTRADA Y SALIDA DEL PROGRAMA

Este apéndice lista la entrada y salida del ejemplo muestra (ver puntos C.1 y C.2). La entrada es listada exactamente como aparece para la simulación. La salida representa los resultados actuales obtenidos a partir de los datos de entrada.

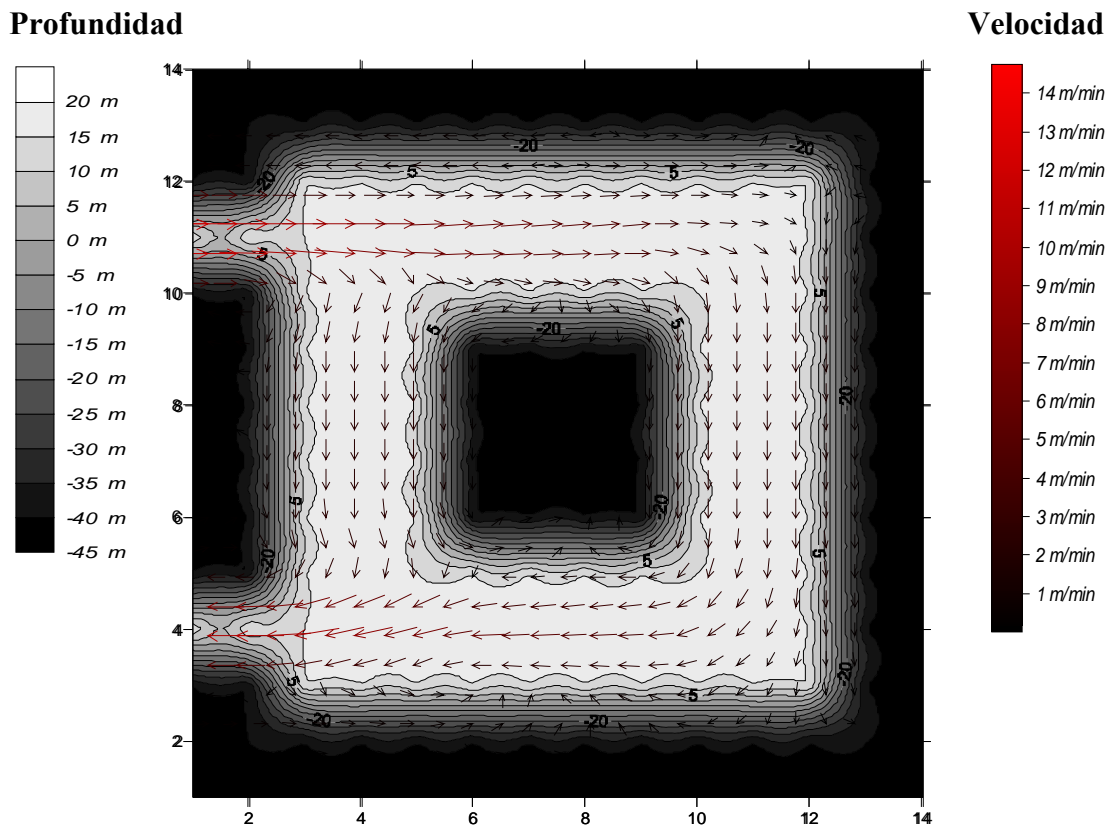
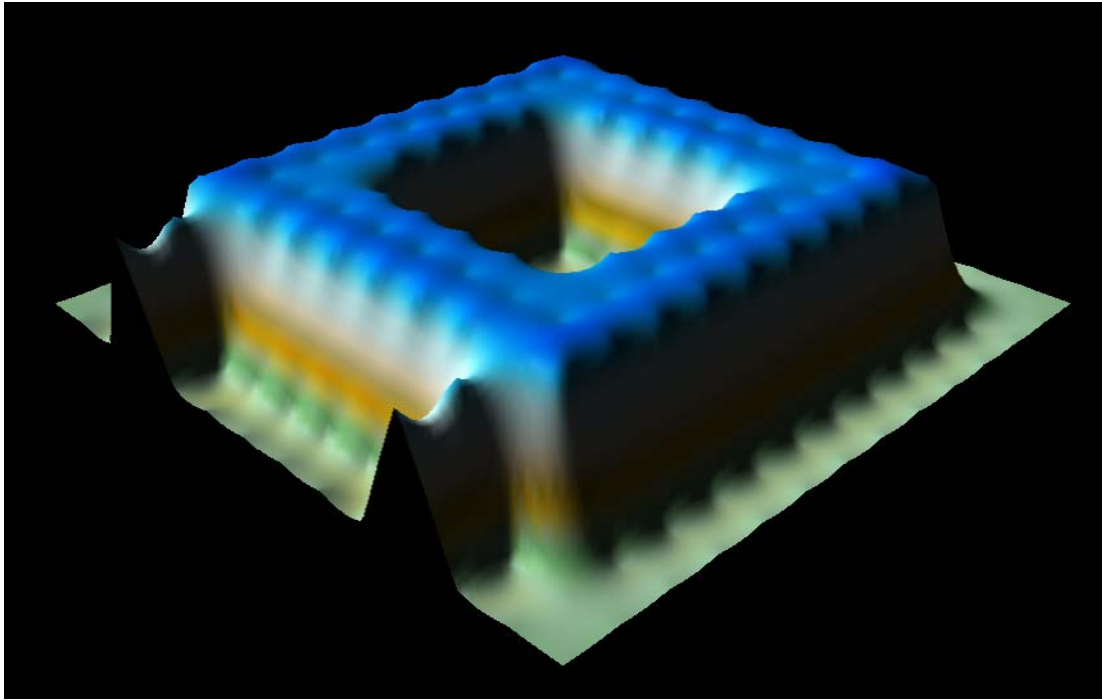
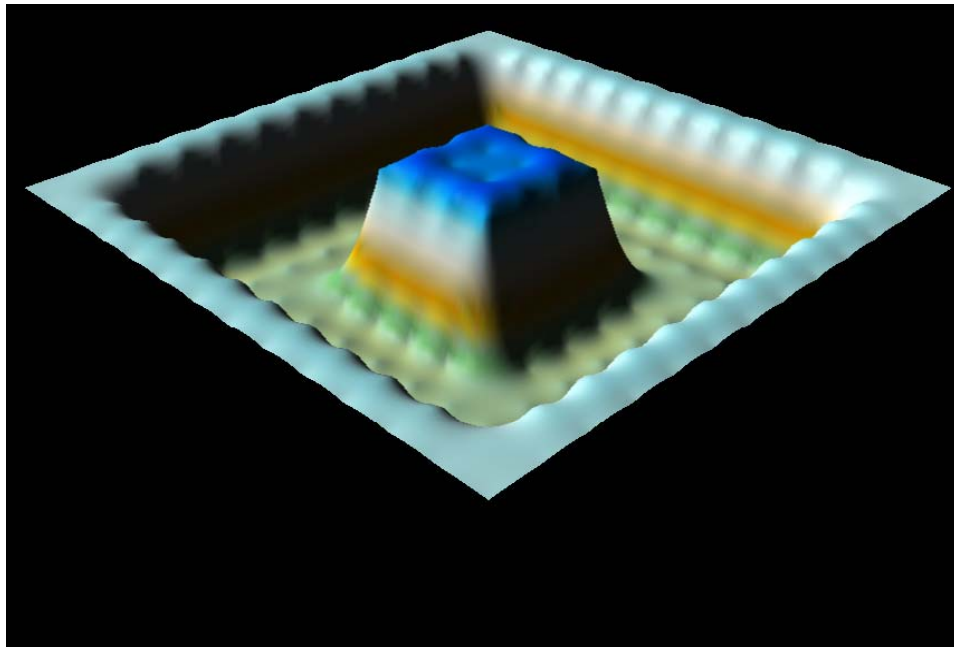


Figura (C.1) Vectores velocidades para ( $t = 750$  s,  $\Delta x = 100$  m,  $\Delta y = 100$  m,  $Q = 7500$  m<sup>3</sup> /min. y  $\Delta t = 15$  s).



**Figura (C.2) Perfil tridimensional (data muestra).**



**Figura (C.3) Perfil tridimensional del fondo para ( $Z = 15$  m,  $D = 5$  m)**

C.1 Muestra de entrada:

15.0 50 50  
0.0 35304.0 4200.0 -1.0

20 1  
10 10

100. 100. 100. 100. 100. 100. 100. 100. 100. 100.  
100. 100. 100. 100. 100. 100. 100. 100. 100. 100.

9 2 2 2 2 2 2 2 2 6  
5 1 1 1 1 1 1 1 1 3  
5 1 1 4 4 4 4 1 1 3  
5 1 3 0 0 0 0 5 1 3  
5 1 3 0 0 0 0 5 1 3  
5 1 3 0 0 0 0 5 1 3  
5 1 3 0 0 0 0 5 1 3  
5 1 1 2 2 2 2 1 1 3  
5 1 1 1 1 1 1 1 1 3  
8 4 4 4 4 4 4 4 4 7

2

3 11 4 1 2 7500.0  
3 4 4 2 1 0.0

5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0  
5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0  
5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0  
5.0 5.0 5.0 -1.0 -1.0 -1.0 -1.0 5.0 5.0 5.0  
5.0 5.0 5.0 -1.0 -1.0 -1.0 -1.0 5.0 5.0 5.0  
5.0 5.0 5.0 -1.0 -1.0 -1.0 -1.0 5.0 5.0 5.0  
5.0 5.0 5.0 -1.0 -1.0 -1.0 -1.0 5.0 5.0 5.0  
5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0  
5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0  
5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0

15

C.2 Muestra de salida:

DELT= 15.00000  
N° DE ITERACIONES DE TIEMPO= 50  
INTERVALO DE ESCRITURA= 50

GAMMA= 0.0  
G= 35304.0  
CHEZY= 4200.0  
SLIP= -1.0

N° DE ITERACIONES INICIALES= 20  
N° DE ITERACIONES POR PASO DE TIEMPO= 1

IMAX= 10  
JMAX= 10

DIMENSIONES DE LAS CELDAS=

100. 100. 100. 100. 100. 100. 100. 100. 100.  
100.

DIMENSIONES DE LAS CELDAS=

100. 100. 100. 100. 100. 100. 100. 100. 100.  
100.

TIPO DE CELDA (ITYP (I,J))

9	2	2	2	2	2	2	2	2	6
5	1	1	1	1	1	1	1	1	3
5	1	1	4	4	4	4	1	1	3
5	1	3	0	0	0	0	5	1	3
5	1	3	0	0	0	0	5	1	3
5	1	3	0	0	0	0	5	1	3
5	1	3	0	0	0	0	5	1	3
5	1	1	2	2	2	2	1	1	3
5	1	1	1	1	1	1	1	1	3

8 4 4 4 4 4 4 4 4 7  
 INFORMACION DE CONTORNO DE LA CELDA

3 11 4 1 2  
 7500.000000000000

3 4 4 2 1  
 0.0000000000000000E+000

PROFUNDIDADES DE ENTRADA

5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0
5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0
5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0
5.0	5.0	5.0	-1.0	-1.0	-1.0	-1.0	5.0	5.0	5.0
5.0	5.0	5.0	-1.0	-1.0	-1.0	-1.0	5.0	5.0	5.0
5.0	5.0	5.0	-1.0	-1.0	-1.0	-1.0	5.0	5.0	5.0
5.0	5.0	5.0	-1.0	-1.0	-1.0	-1.0	5.0	5.0	5.0
5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0
5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0
5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0

MATRIZ DE IVEL

0	0	0	0	0	0	0	0	0	0
4	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	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	0	0	0
0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

ELEVACION DE SUPERFICIE LIBRE DE AGUA= 15.00

ARREGLO TOTAL DE PROFUNDIDADES (D)

```

-1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -
1.0
-1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -
1.0
-1.0 -1.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 -1.0 -
1.0
5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 -1.0 -
1.0
-1.0 -1.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 -1.0 -
1.0
-1.0 -1.0 5.0 5.0 5.0 -1.0 -1.0 -1.0 -1.0 5.0 5.0 5.0 -1.0 -
1.0
-1.0 -1.0 5.0 5.0 5.0 -1.0 -1.0 -1.0 -1.0 5.0 5.0 5.0 -1.0 -
1.0

-1.0 -1.0 5.0 5.0 5.0 -1.0 -1.0 -1.0 -1.0 5.0 5.0 5.0 -1.0 -
1.0
-1.0 -1.0 5.0 5.0 5.0 -1.0 -1.0 -1.0 -1.0 5.0 5.0 5.0 -1.0 -
1.0
-1.0 -1.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 -1.0 -
1.0
5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 -1.0 -
1.0
-1.0 -1.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 -1.0 -
1.0
-1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -
1.0
-1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -1.0 -
1.0

```

ARREGLO TOTAL DE PROFUNDIDADES (Z)

```

*****
**
*****
**
***** 15. 15. 15. 15. 15. 15. 15. 15. 15.
15.*****
 15. 15. 15. 15. 15. 15. 15. 15. 15. 15.
15.*****
***** 15. 15. 15. 15. 15. 15. 15. 15.
15.*****
***** 15. 15. 15.***** 15. 15.
15.*****
***** 15. 15. 15.***** 15. 15.
15.*****

```

```

***** 15. 15. 15.***** 15. 15.
15.*****
***** 15. 15. 15.***** 15. 15.
15.*****
***** 15. 15. 15. 15. 15. 15. 15. 15.
15.*****
15. 15. 15. 15. 15. 15. 15. 15. 15. 15.
15.*****
***** 15. 15. 15. 15. 15. 15. 15. 15.
15.*****
*****
**
*****
**

```

MATRIZ INICIAL DE U

```

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.
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.
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.
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.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
15. 15. 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. 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.

```

CONDICIONES INICIALES:

VELOCIDADES EN X

```

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
15.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

```



VELOCIDADES EN Y

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

ELEVACIONES Z

```
*****
****
***** 15.00 15.00 15.00 15.00 15.00 15.00 15.00 15.00 15.00 15.00
15.00*****
 15.00 15.00 15.00 15.00 15.00 15.00 15.00 15.00 15.00 15.00
15.00*****
***** 15.00 15.00 15.00 15.00 15.00 15.00 15.00 15.00 15.00 15.00
15.00*****
***** 15.00 15.00 15.00***** 15.00 15.00
15.00*****
***** 15.00 15.00 15.00***** 15.00 15.00
15.00*****
***** 15.00 15.00 15.00***** 15.00 15.00
15.00*****
***** 15.00 15.00 15.00 15.00 15.00 15.00 15.00 15.00 15.00 15.00
15.00*****
 15.00 15.00 15.00 15.00 15.00 15.00 15.00 15.00 15.00 15.00
15.00*****
***** 15.00 15.00 15.00 15.00 15.00 15.00 15.00 15.00 15.00 15.00
15.00*****
*****
****
1
```

LUEGO DE 20 ITERACIONES INICIALES, SE OBTIENE:

VELOCIDADES EN X

```
***** -0.31 -0.61 -0.91 -0.45 0.01 0.45 0.89 0.59 0.30*****
```

```

15.00 13.11 11.24 9.39 8.33 7.29 6.26 5.24 3.49 1.75*****
***** -0.32 -0.64 -0.95 -0.48 -0.02 0.44 0.89 0.59 0.29*****
***** 0.00 0.00***** 0.00 0.00*****
***** 0.00 0.00***** 0.00 0.00*****
***** 0.00 0.00***** 0.00 0.00*****
***** 0.00 0.00***** 0.00 0.00*****
***** 0.27 0.55 0.83 0.40 -0.03 -0.45 -0.86 -0.57 -0.29*****
-11.95-10.34 -8.77 -7.24 -6.46 -5.69 -4.94 -4.21 -2.80 -1.40*****
***** 0.27 0.55 0.84 0.41 -0.02 -0.43 -0.84 -0.56 -0.28*****

```

VELOCIDADES EN Y

```

*****
-0.27 -0.26 -0.26 0.51 0.50 0.49 0.48 -0.26 -0.26 -0.26
-2.12 -2.09 -2.07 -0.51 -0.50 -0.50 -0.49 -1.98 -1.96 -1.96
-2.41 -2.37 -2.34***** -2.24 -2.22 -2.22
-2.37 -2.33 -2.31***** -2.20 -2.18 -2.18
-2.33 -2.30 -2.27***** -2.16 -2.15 -2.14
-2.30 -2.26 -2.23***** -2.12 -2.11 -2.10
-2.26 -2.23 -2.20***** -2.09 -2.07 -2.07
-1.95 -1.92 -1.88 -0.41 -0.40 -0.39 -0.38 -1.77 -1.75 -1.75
-0.31 -0.31 -0.32 0.40 0.39 0.39 0.38 -0.32 -0.32 -0.32
*****

```

ELEVACIONES Z

```

*****
****
***** 15.00 15.00 15.00 15.00 15.00 15.00 15.00 15.00 15.00
15.00*****
15.00 15.00 15.00 15.00 15.00 15.00 15.00 15.00 15.00
15.00*****
***** 15.00 15.00 15.00 15.00 15.00 15.00 15.00 15.00 15.00
15.00*****
***** 15.00 15.00 15.00 0.00 0.00 0.00 0.00 15.00 15.00
15.00*****
***** 15.00 15.00 15.00 0.00 0.00 0.00 0.00 15.00 15.00
15.00*****
***** 15.00 15.00 15.00 0.00 0.00 0.00 0.00 15.00 15.00
15.00*****

```

```

***** 15.00 15.00 15.00 15.00 15.00 15.00 15.00 15.00 15.00
15.00*****
  15.00 15.00 15.00 15.00 15.00 15.00 15.00 15.00 15.00
15.00*****
***** 15.00 15.00 15.00 15.00 15.00 15.00 15.00 15.00 15.00
15.00*****
*****
****
1

```

```

PASO DE TIEMPO NUMERO= 50
DELT=      15.00

```

```

TIEMPO TOTAL=      750.00

```

VELOCIDADES EN X

```

***** -0.41 -0.78 -1.13 -0.64 -0.15  0.35  0.84  0.55  0.28*****
  15.00 13.33 11.60  9.80  8.79  7.77  6.73  5.68  3.81  1.91*****
***** -0.46 -0.88 -1.26 -0.76 -0.25  0.27  0.80  0.52  0.26*****
***** -0.01 -0.01***** 0.00  0.00*****
***** -0.01 -0.01***** 0.00  0.00*****
***** -0.01 -0.01***** 0.00  0.00*****
***** -0.01 -0.01***** 0.00  0.00*****
*****  0.21  0.45  0.73  0.23 -0.25 -0.71 -1.15 -0.77 -0.39*****
-14.39-12.33-10.38 -8.54 -7.56 -6.62 -5.71 -4.84 -3.21 -1.60*****
*****  0.24  0.52  0.82  0.32 -0.17 -0.64 -1.09 -0.73 -0.37*****

```

VELOCIDAES EN Y

```

*****
-0.41 -0.36 -0.34  0.50  0.50  0.50  0.51 -0.28 -0.27 -0.27
-2.07 -2.09 -2.13 -0.51 -0.52 -0.53 -0.54 -2.15 -2.16 -2.17
-2.52 -2.50 -2.51***** -2.41 -2.41 -2.42

```

```

-2.52 -2.50 -2.49***** -2.41 -2.40 -2.41
-2.52 -2.49 -2.48***** -2.40 -2.40 -2.40
-2.53 -2.49 -2.46***** -2.40 -2.39 -2.39
-2.53 -2.49 -2.44***** -2.39 -2.38 -2.38
-2.31 -2.24 -2.16 -0.49 -0.47 -0.45 -0.43 -2.01 -1.99 -1.98
-0.25 -0.29 -0.31 0.49 0.48 0.46 0.45 -0.37 -0.37 -0.37
*****

```

ELEVACIONES Z

```

*****
****

***** 15.00 15.00 15.00 15.00 15.00 15.00 15.00 15.00 15.00
15.00*****

 15.00 15.00 15.00 15.00 15.00 15.00 15.00 15.00 15.00
15.00*****

***** 15.00 15.00 15.00 15.00 15.00 15.00 15.00 15.00 15.00
15.00*****

***** 15.00 15.00 15.00 0.00 0.00 0.00 0.00 15.00 15.00
15.00*****

***** 15.00 15.00 15.00 0.00 0.00 0.00 0.00 15.00 15.00
15.00*****

***** 15.00 15.00 15.00 0.00 0.00 0.00 0.00 15.00 15.00
15.00*****

***** 15.00 15.00 15.00 0.00 0.00 0.00 0.00 15.00 15.00
15.00*****

***** 15.00 15.00 15.00 15.00 15.00 15.00 15.00 15.00 15.00
15.00*****

***** 15.00 15.00 15.00 15.00 15.00 15.00 15.00 15.00 15.00
15.00*****

*****
****
1

```