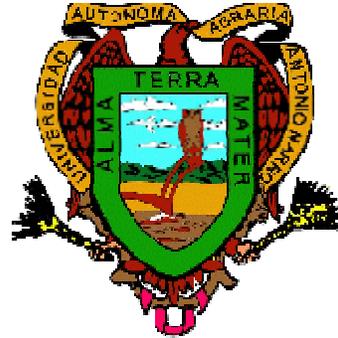


Universidad Autónoma Agraria “Antonio Narro”

División de ingeniería



Alternativas de programación con enfoque en la hidráulica.

Trabajo de Tesis

Presentada como requisito parcial para obtener el título de:
Ingeniero Agrónomo en Irrigación

Por:

Juan Fernando Pérez Calvo

Buenavista, Saltillo, Coahuila, México.

Marzo 2009

**Universidad Autónoma Agraria
"Antonio Narro"**

A sus hermanos y personas con me apoyaron en cualquier buen bienestar

División de ingeniería
Departamento de Riego y Drenaje

Alternativas de programación con enfoque en la hidráulica

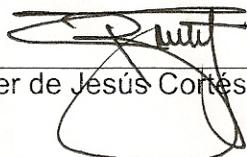
Trabajo de TESIS

Presentado por:
Juan Fernando Pérez Calvo

Que somete a consideración del Honorable jurado examinador como requisito parcial
para obtener el título de:
Ingeniero Agrónomo en Irrigación

Aprobada

El presidente del jurado



Dr. Javier de Jesús Cortés Bracho

Vocal



M.C. Sergio Z. Garza Vara

Vocal



Universidad Autónoma Agraria
"ANTONIO NARRO" Dr. Daniel Gómez García

El coordinador de la División de Ingeniería



Dr. Raúl Rodríguez Gutiérrez

Buenavista, Saltillo, Coahuila. Marzo de 2009

**Universidad Autónoma Agraria
“Antonio Narro”**

División de ingeniería
Departamento de Riego y Drenaje

Alternativas de programación con enfoque en la hidráulica

Trabajo de TESIS

Presentado por:
Juan Fernando Pérez Calvo

Que somete a consideración del Honorable jurado examinador como requisito parcial
para obtener el título de:
Ingeniero Agrónomo en Irrigación

Aprobada
El presidente del jurado

Dr. Javier de Jesús Cortés Bracho

Vocal

Vocal

M.C. Sergio Z. Garza Vara

Dr. Daniel Gómez García

El coordinador de la División de Ingeniería

Dr. Raúl Rodríguez Gutiérrez

Buenavista, Saltillo, Coahuila. Marzo de 2009

Agradecimiento

A mis hermanos y personas que me ayudaron en tener un buen bienestar.

Dedicatoria

A mi novia Erika Carina Ruiz Cruz y mi familia, a toda.

Índice de contenido

Agradecimiento	i
Dedicatoria	ii
Índice de contenido	iii
Resumen	v
I.- Introducción	1
1.1.- Objetivos	3
II.- Revisión de literatura	4
III.- Materiales y métodos	11
3.1.- Salidas múltiples laterales	13
3.1.1.- Descripción	13
3.1.2.- Objetivo del programa	13
3.1.3.- Diagrama de flujo	14
3.1.4.- Ejemplos	15
3.1.5.- Códigos	18
3.1.5.1.- VISUAL BASIC 6.0	18
3.1.5.2.- MATLAB 6.5.....	27
3.2.- Flujo uniforme en estado permanente para canales rectangulares, trapeciales y triangulares; Simétricos y asimétricos.	30
3.2.1.- Descripción	30
3.2.2.- Objetivo del programa	30
3.2.3.- Diagrama de flujo	31
3.2.4.- Ejemplos	32
3.2.5.- Códigos	38
3.2.5.1.- VISUAL BASIC 6.0.....	38
3.2.5.2.- MATLAB 6.5.....	47
3.3.- Flujo gradualmente variado	50
3.3.1.- Descripción	50
3.3.2.- Objetivo del programa	50
3.3.3.- Diagrama de flujo	51

3.3.4.- Ejemplos	52
3.3.5.- Códigos	54
3.3.5.1- VISUAL BASIC 6.0.....	54
3.3.5.2.- MATLAB 6.5.....	64
IV.- Resultados y observaciones	66
V.- Conclusiones	69
VI.- Bibliografías	70
VII.- Apéndices	72
Apéndice I: Método de aproximaciones sucesivas	72
Apéndice II: Método del trapecio	75
Apéndice III: Salidas múltiples laterales	77
Apéndice IV: Flujo uniforme permanente	80
Apéndice V: Flujo gradualmente variado	82
Apéndice VI: Resultados: Salidas múltiples laterales	86
Apéndice VII: Concepto de estructuras de control	88
Apéndice VIII: Estructuras de control en MATLAB 6.5 Y VISUAL BASIC 6.0	91
Apéndice IX: Descripción de símbolos utilizados en los diagramas de flujo	92
Apéndice X: Variables utilizadas en el diagrama de flujo salidas múltiples laterales	93
Apéndice XI: Variables utilizadas en el diagrama de flujo flujo uniforme permanente	94
Apéndice XII: Variables utilizadas en el diagrama de flujo flujo gradualmente variado	95
Apéndice XIII: Descripción de la variables utilizadas en los códigos ..	96

Resumen

La ingeniería hidráulica requiere la utilización de métodos numéricos para la solución de los problemas reales que se presentan en su entorno físico. Por esto se vio la necesidad de crear los programas para su rápida ejecución, considerando como lenguajes: MATLAB 6.5 y VISUAL BASIC 6.0

Ambos lenguajes de programación (MATLAB 6.5 Y VISUAL BASIC 6.0), y los lenguajes existentes son, indiscutiblemente, indispensables para la programación en cualquier enfoque a que se les asigne.

En este trabajo, los temas de la hidráulica a programar son: Salidas múltiples laterales con gasto en tránsito sobrante igual a cero, Cálculo del tirante normal de flujo en canales simples rectangulares, triangulares, trapezoidales; simétricos y asimétricos, y Cálculo de la Longitud del Flujo gradualmente variado.

En fin, podemos deducir que el lenguaje de mayor facilidad para manejo numérico, que es la base fundamental para estos programas codificados, es en MATLAB 6.5; en VISUAL BASIC 6.0 se torna un poco tedioso, por no decir laborioso; en el sentido de codificación en los controles. Igual que con otros lenguajes.

MATLAB por ser una herramienta de alto nivel, puede requerir hasta un orden de magnitud menos de esfuerzo que con lenguajes de programación convencionales, como Fortran, Pascal, C/C++, Java o VISUAL BASIC.

MATLAB es un programa de cálculo técnico y científico. Sin embargo, en lo personal, también es un lenguaje de programación para iniciar en el entorno de la programación; tanto básica hasta avanzada, por su característica de simplicidad o legibilidad.

Introducción

Los avances en la tecnología siempre van parejos con progresos en los lenguajes de programación y con nuevas ayudas para simplificar el uso del programador, con lo cual un número mayor de usuarios se beneficia de él. Pero la necesidad de hacer programas para resolver problemas específicos quizás nunca desaparecerá.

La ingeniería hidráulica requiere de muchos métodos numéricos para la solución de los problemas físicos que se presentan en su entorno real, por ello se vio la necesidad de crear los programas o “software” para la rápida ejecución de estos, originados a partir de estudios reales, considerando como lenguajes de programación: MATLAB 6.5 (Laboratorio de Matrices) y VISUAL BASIC 6.0 (Código de instrucciones simbólicas de propósito general para principiantes).

El término programación se utiliza para designar a la elaboración de códigos; el desarrollo de sistemas complejos se denomina ingeniería de software. Ha adquirido gran importancia por su gran versatilidad, por su manejo práctico a la solución de problemas mediante una serie de información ordenada y sistematizada. (<http://www.lenguajes-de-programacion.com/programacion.shtml>).

Este trabajo expone códigos de programas con enfoque en la hidráulica con dos lenguajes de programación: MATLAB 6.5 y VISUAL BASIC 6.0.

Este proyecto también enfoca una breve discusión sobre las ventajas e inconvenientes de los lenguajes de programación (MATLAB 6.5 Y VISUAL BASIC 6.0) teniendo como factor importante las características de aprendizaje.

Algunos de los lenguajes de programación, (<http://lenguajes-de-programacion.com/herramientas-de-programacion.shtml>)

Programas que permiten crear rutinas, programas y utilitarios:

Basic

Pascal

Programación para sistemas:

C

C++

Programación orientada hacia sistemas de gestión empresarial como nóminas y contabilidad:

Cobol 6

Programación específica para cálculos matemáticos y/o numéricos:

Fortran 90

Matlab

Programación para ambientes gráficos:

Visual Basic 6.0

Delphi 8

Visual C.

Programación para la creación de páginas WEB para Internet:

Html

Java 1.0

1.1.- Objetivos.

Los objetivos que se consideran en este trabajo son:

Exponer tres tipos de problemas de la hidráulica, para su solución, con dos lenguajes de programación: MATLAB 6.5 y VISUAL BASIC 6.0.

Identificar un lenguaje de programación adecuado a un modelo o problema específico.

Predisponer la versatilidad de un lenguaje de programación de acuerdo a las características de manejo y aprendizaje.

REVISIÓN DE LITERATURA

En matemáticas, ciencias de la computación, y disciplinas relacionadas, un algoritmo (del latín, *dixit algorithmus* y éste a su vez del matemático persa al-Jwarizmi) es una lista bien definida, ordenada y finita de operaciones que permite hallar la solución a un problema. Dado un estado inicial y una entrada, a través de pasos sucesivos y bien definidos se llega a un estado final, obteniendo una solución. Los algoritmos son objeto de estudio de la algoritmia, (<http://es.wikipedia.org/wiki/algoritmo>).

Un programa es un conjunto de operaciones especificadas en un determinado lenguaje de programación y para un computador concreto, susceptible de ser ejecutado (o compilado o interpretado), (<http://es.wikipedia.org/wiki/programacion>).

En informática, la programación es un proceso por el cual se escribe (en un lenguaje de programación), se prueba, se depura y se mantiene el código fuente de un programa informático. Dentro de la informática, los programas son los elementos que forman el software, que es el conjunto de las instrucciones que ejecuta el hardware de una computadora para realizar una tarea determinada. Por lo tanto, la programación es una de las principales áreas dentro de la informática, (<http://es.wikipedia.org/wiki/programacion>).

Un lenguaje de programación es un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Es utilizado para controlar el comportamiento físico y lógico de una máquina, (http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n).

Un diagrama de flujo es la representación gráfica de un proceso. Un diagrama de flujo tiene como objetivo facilitarnos la comprensión de un algoritmo o simplificar el análisis de un proceso. El diagrama de flujo consta de símbolos como cuadros, rombos, óvalos, etc. que representan a cada uno de los pasos a seguir durante un

proceso. Estos símbolos están conectados por flechas de un solo sentido y nos indican la secuencia en que se van desarrollando las distintas tareas, (<http://elticus.com/?contenido=19>).

La computadora es una herramienta que no es capaz de resolver problemas por si misma, sino que es necesario introducir en su memoria una serie de instrucciones que le ordenen paso a paso que ejecute las operaciones necesarias, y resuelva los problemas de interés. La computadora es la culminación de dispositivos de cálculo como el ábaco, regla de cálculo, tablas, nomogramas, calculadora de escritorio, (Luthe *et. al*, 1985; Citado por Hernández, 1987).

El lenguaje de programación más apropiado para usarse en un programa en particular incluye equilibrar las características funcionales del lenguaje con aspectos como costo, control y complejidad. Un factor importante que debe considerarse al seleccionar cualquier lenguaje de programación es la cantidad de control directo que se necesita para operar el hardware, (<http://www.monografias.com/trabajos26/lenguajes-programacion/lenguajes-programacion.shtml>).

MATLAB es uno de los lenguajes de programación más utilizados en el ámbito de la investigación debido a su gran capacidad para el procesamiento de cálculos matemáticos. Además de que cuenta con cajas de herramientas (ToolBoxes) que contienen controles que facilitan aún más la programación de aplicaciones específicas en diferentes áreas del conocimiento.

VISUAL BASIC es un lenguaje visual que desciende del lenguaje de programación BASIC (Beginners All-purpose Symbolic Instruction Code; código de instrucciones simbólicas de propósito general para principiantes). Creado con la intención de simplificar la programación utilizando un ambiente de desarrollo completamente gráfico que facilitara la creación de interfaces gráficas y en cierta

medida también la programación misma. Su primera versión fue presentada en 1991, (http://es.wikipedia.org/wiki/Visual_Basic).

MATLAB se usa en una gran variedad de áreas de aplicación, como en el proceso de señales y comunicación, proceso de imágenes, el diseño y análisis de sistemas de control, la biología, la geología, las finanzas, la economía, la instrumentación, etcétera. La arquitectura abierta de MATLAB facilita su uso y el de sus productos acompañantes para explorar datos y crear herramientas personalizadas que permitan obtener con rapidez información y confieran ventajas competitivas.

VISUAL BASIC posee varias bibliotecas para manejo de bases de datos, pudiendo conectar con cualquier base de datos a través de ODBC (Informix, DBase, Acces, Mysql, SQL Server, etcétera) a través de ADO. Es utilizado principalmente para aplicaciones de gestión de empresas, debido a la rapidez con la que puede hacerse un programa que utilice una base de datos sencilla, además de la abundancia de programadores en este lenguaje, (http://es.wikipedia.org/wiki/Visual_Basic).

El elemento (variable) básico en MATLAB es la matriz compleja de doble precisión, estructura que abarca realmente todo tipo de datos, desde escalares tales como números reales o complejos, hasta vectores o matrices de tamaños arbitrarios, que no requieren ser dimensionados, manteniendo valores de 8 bytes por cada elemento que integran a cada elemento de ésta, (<http://www.di.uniovi.es/~dani/asignaturas/transparencias-leccion20.pdf>).

VISUAL BASIC maneja como datos esenciales en su lenguaje los siguientes tipos de variables con dimensiones: Double, Single, Long, Integer, Boolean, Byte, Currency, Date, Objet, String, Variant. Conteniendo 8 bytes, 4 bytes, 4 bytes, 2 bytes, 2 bytes, 1 bytes, 8 bytes, 8 bytes, 4 bytes, 10 bytes, 16 y 22 bytes, respectivamente. Con la declaración exacta de estas en cada proceso de un

programa o procedimiento, nos hace eficiente el rendimiento de cálculo por el motivo de no ocupar demasiada memoria RAM en su ejecución.

MATLAB fue escrito originalmente en FORTRAN, actualmente está escrito en lenguaje C. MATLAB es un lenguaje de programación amigable con características más avanzadas y mucho más fáciles de usar que los lenguajes de programación como BASIC, PASCAL Ó C, (<http://www.monografias.com/trabajos5/matlab/matlab.shtml>).

El lenguaje BASIC fue en parte basado en FORTRAN II y en parte en Algol 60, con adiciones para hacerlo apropiado para tiempo compartido y aritmética de matrices, BASIC fue implementado por primera vez en la mainframe GE-265³, que soportaba múltiples terminales. Contrario a la creencia popular, era un lenguaje compilado al momento de su introducción.

MATLAB distingue entre mayúsculas y minúsculas (no son iguales). Y no necesita declaración ninguna de las variables a emplear en un código de cualquier magnitud que tome ésta. En este sentido, los nombres de función se introducirán en minúsculas.

VISUAL BASIC 6.0 no distingue entre mayúsculas y minúsculas. Para cálculos enteros, reales o decimales, no necesita declaración de cada elemento o variable. Caso contrario para manejo de base de datos, vectores o arreglos.

MATLAB es el nombre abreviado de “MATrix LABoratory”. MATLAB es un programa para realizar cálculos numéricos con vectores y matrices. Como caso particular puede también trabajar con números escalares, tanto reales como complejos. Una de las capacidades más atractivas es la de realizar una amplia variedad de gráficos en dos y tres dimensiones. MATLAB tiene también un lenguaje de programación propio,

(http://www.ucm.es/info/Geofis/practicas_ICNYP/modulos_1_7_2005-06_grupo_D.pdf).

VISUAL BASIC es un lenguaje de fácil aprendizaje pensando para programadores principiantes como expertos, guiado por eventos, y centrado en un motor de formularios que facilita el rápido desarrollo de aplicaciones gráficas, (<http://www.monografias.com/trabajos62/visual-basic-prototipo/visual-basic-prototipo.shtml>).

MATLAB trabaja bajo los siguientes archivos o extensiones para la ejecución de un programa. El archivo “.m” (scripts files), en este se ponen secuencialmente comandos de MATLAB que se ejecutan en ese orden al introducir el nombre del fichero que hemos creado. El archivo “.fig”, junto con el archivo “.m”, permite programación basado en interfaz gráfica. El archivo “.p”, son archivos pre compilados a partir de un archivo “.m”

VISUAL BASIC 6.0 trabaja con los siguientes archivos o extensiones para la ejecución de un programa (lo elemental). El archivo “.frm”, contiene el código y el aspecto de un formulario del proyecto. El archivo “.vbp” contiene información de todos los archivos del proyecto.

MATLAB es la abreviatura de MATrix LABoratory (laboratorio de matrices). Se trata de un software matemático muy versátil que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M). Está disponible para las plataformas Unix, Windows y Apple Mac OS X, (<http://www.taringa.net/posts/downloads/1410159/Matlab,-en-100mb.html>).

Microsoft VISUAL BASIC x.0 para versiones desde la 1.0 hasta 6.0, (con las diferencias entre las versiones desde la 1.0 (MS-DOS/Windows 3.1) hasta la 3.0 (16 bits, Windows 3.1) y las de la 4.0 (16/32 bits, Windows 3.1/95/NT) hasta la 6.0 (32 bits, Windows 9x/Me/NT/2000/XP/VISTA, (http://es.wikipedia.org/wiki/Visual_Basic)).

Si se dispone de una computadora y debe resolverse un gran número de problemas de estimación del tirante normal, entonces la mejor aproximación puede ser un procedimiento numérico de tanteos, (French, 1988).

En general, el cálculo más difícil y tedioso del flujo normal ocurre cuando Q, S, T, n son conocidos y y_n debe ser estimado. En Tal caso, no es posible una solución explícita de la ecuación siguiente y el problema debe ser solucionado por tanteos, diagramas de curvas o métodos numéricos. (French, 1988).

$$Q = f(T, n, S, y_n)$$

donde :

Q = gasto en el canal

T = factor de forma del canal

y_n = tirnte normal del flujo

S = pendiente longitudinal del cnal

n = rugosidad del canal

$$Q = \frac{\phi}{n} x A x R^{2/3} x \sqrt{S}$$

$\phi = 1, \text{ sistema metrico}$

Incluso con la llegada de “grandes” computadoras digitales de alta velocidad (*mainframe*), minicomputadoras y calculadoras de mano, se ha prescindido en gran medida de la necesidad de soluciones gráficas de la ecuación de flujo gradualmente variado, todavía existen situaciones en que las soluciones gráficas pueden ser efectivas. Por ejemplo, en canales prismáticos los métodos de paso directo en “y” y de integración directa predicen la distancia longitudinal para un tirante especificado. En muchos casos sería más conveniente que el tirante se pudiese estimar para una distancia longitudinal especificada. (French, 1988).

Para cuantificar los tirantes del perfil del flujo gradualmente variado, tomando en cuenta la ubicación de cada tirante a lo largo del flujo, es necesario integrar la ecuación diferencial dinámica del flujo gradualmente variado. Esta ecuación

diferencial, en la actualidad, es imposible integrarse con una integral directa, tal como se muestra la siguiente expresión. (Chow, 1985, Citado por Hernández, 1987).

donde :

$Y =$ Funciona integrar

$S_o =$ Pendiente del canal

$S_f =$ Pendiente de la línea de energía total

$$Y = \int \left(\frac{S_o - S_f}{1 + \alpha x \frac{d}{dy} \left(\frac{V^2}{2g} \right)} \right) dx \quad \alpha = \text{Coeficiente de Coriolis}$$

$\frac{d}{dy} \left(\frac{V^2}{2g} \right) =$ Derivada de la carga por velocidad, respecto al tirante
 $dx =$ Diferencial de integración

(Braud y Soom, 198; Citado por Salvador, 1997). En conductos a presión con salidas múltiples laterales, indican que la presión a lo largo de la línea lateral cambia debido a las pérdidas de carga por fricción y a la elevación por lo cual también la descarga varía de un emisor a otro. Esto ocasiona una aplicación no uniforme del agua.

La distribución de la presión a lo largo de una línea de riego por goteo, distribuidora o lateral, es controlada por el descenso de energía provocada por la fricción del agua así como por la ganancia o pérdida de energía ocasionada por la pendiente según sea descendente o ascendente. La variabilidad de la descarga de los emisores a lo largo de la línea lateral, es función de la longitud total, la presión a la entrada, el espaciamiento entre emisores, su descarga nominal y del gasto total de entrada a la línea, (Wu y Gitlin, 1974; Citado por Salvador, 1997).

MATERIALES Y MÉTODOS

1.- Lugar y fecha de establecimiento:

El presente proyecto se elaboró en el Departamento de Riego y Drenaje, de la Universidad Autónoma Agraria "Antonio Narro". En Buenavista, Saltillo, Coahuila.

2.- Materiales y equipos requeridos:

Los materiales o equipos que se manejaron para el proyecto, son una computadora con las siguientes características o equivalente.

Plataforma: Windows.

Sistema operativo: Microsoft Windows XP Profesional Versión 2002.

Equipo: Intel(R), Celeron(R), C.P.U. 2.2 Ghz, 480 MB de RAM.

Capacidad: 41.2 GB en C.

3.- Consideraciones estadísticas:

El software a utilizar es: MATLAB 6.5 y VISUAL BASIC 6.0. Temas a considerar en este proyecto.

El estudio de este trabajo trata sobre una alternativa de programación sobre los siguientes temas de la hidráulica general:

- a. Salidas múltiples laterales con gasto en tránsito sobrante igual a cero.
- b. Cálculo del tirante normal en flujo uniforme en estado permanente en canales simples triangulares, trapeciales y rectangulares; simétricos y asimétricos.
- c. Cálculo de la longitud del flujo gradualmente variado.

Son tres temas a codificar y los aspectos hidráulicos a obtener son:

a.- Salidas múltiples laterales con gasto en tránsito sobrante igual a cero.

Gasto en los tramos de la lateral.

Pérdida de carga en los tramos de la lateral.

Gasto por orificio de salida o descarga de la lateral.

Altura piezométrica o presión sobre las salidas o descarga de los orificios.

b.- Flujo uniforme en estado permanente en canales simples triangulares, trapeciales y rectangulares; simétricos y asimétricos.

Tirante normal.

Área hidráulica de la sección transversal del canal.

Perímetro hidráulico de la sección transversal del canal.

Número de Froude.

Tipo de flujo que circula por el canal (crítico, subcrítico y supercrítico).

c.- Cálculo de la longitud del flujo gradualmente variado.

Incremento de los tirantes a lo largo del canal.

Área hidráulica de la sección transversal a lo largo del flujo del canal.

Perímetro hidráulico de la sección transversal a lo largo del flujo del canal.

Radio hidráulico de la sección transversal a lo largo del flujo del canal.

Valores de la función flujo gradualmente variado, a integrar numéricamente.

Áreas parciales de la función flujo gradualmente variado.

Longitud del flujo gradualmente variado.

Se tomaron estos apartados por que se encuentran dentro del área de estudio y para tener un panorama o retroalimentación más general de esta disciplina. Se dieron énfasis a estos métodos por razones de complejidad (métodos numéricos), simplicidad (en lo que respecta a códigos en los lenguajes).

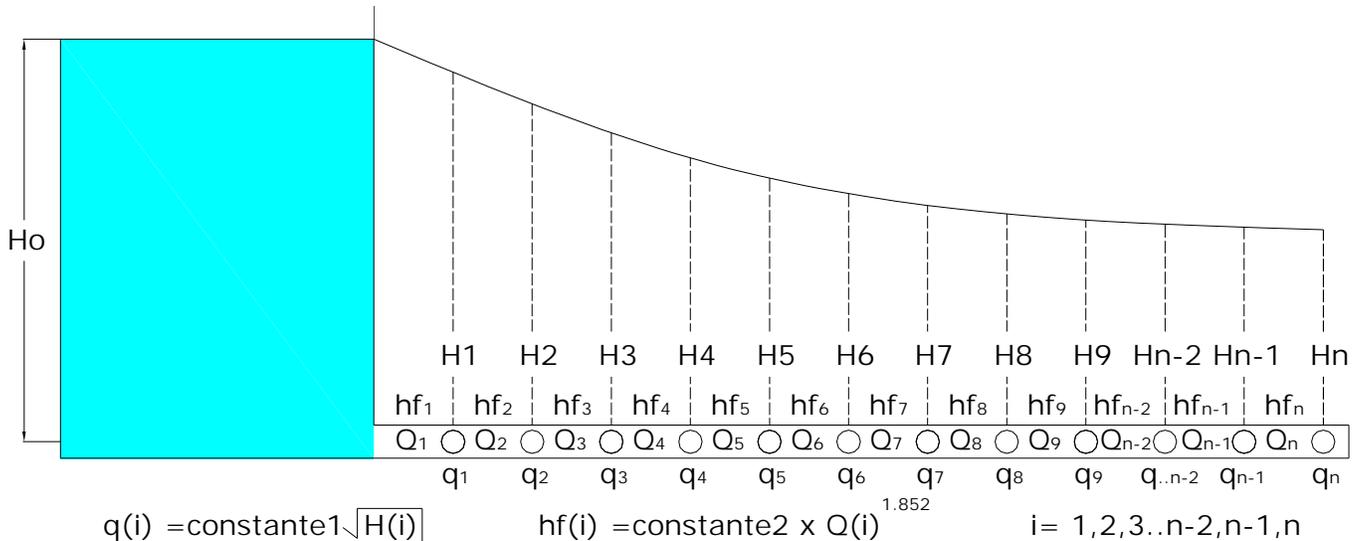
3.1.- SALIDAS MÚLTIPLES LATERALES.

3.1.1.- DESCRIPCIÓN.

Para calcular la fricción en forma manual, comúnmente se calcula primero la fricción como si el caudal total de entrada pasara a través de toda la longitud de la manguera. Para convertir esta fricción así calculada en la fricción que ocurriría en una manguera "lateral", en la cual toda el agua se descarga a través de salidas de emisores se usa el factor "F" para varias salidas. (<http://www.itrc.org/riego/capitulo5.pdf>). ([Ver Tabla No. 2, apéndice III](#)).

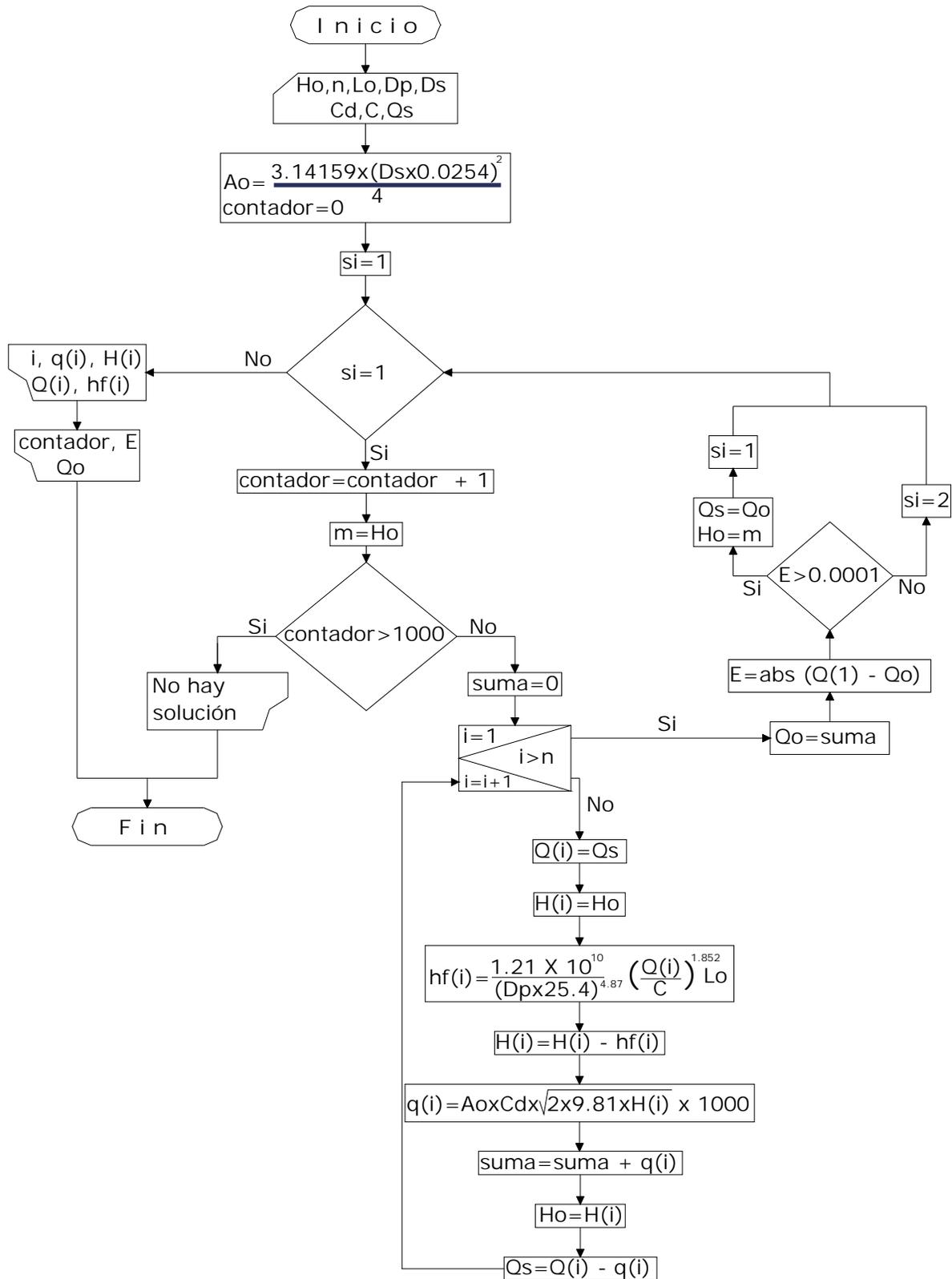
3.1.2.- OBJETIVO DEL PROGRAMA.

Calcular el gasto que alimenta a la tubería principal, el gasto en cada orificio, sus respectivas cargas de operación por cada orificio y la pérdida por fricción en cada tramo o segmento de la tubería principal conociendo la presión de entrada en la tubería principal, que se presente en cualquier tubería lateral o principal. Se utilizó el método de aproximaciones sucesivas. Ver fórmulas en el siguiente diagrama de flujo.



$Q_1 = q_1 + Q_2$	$H_0 = hf_1 + H_1$	$H_1 = H_0 - hf_1$	$Q_2 = Q_1 - q_1$	
$Q_2 = q_2 + Q_3$	$H_1 = hf_2 + H_2$	$H_2 = H_1 - hf_2$	$Q_3 = Q_2 - q_2$	<i>Solución del problema</i>
$Q_{n-1} = q_{n-1} + Q_n$	$H_2 = hf_3 + H_3$	$H_3 = H_2 - hf_3$	$Q_n = Q_{n-1} - q_{n-1}$	
$Q_n = q_n$	$H_3 = hf_4 + H_4$	$H_{n-1} = H_{n-2} - hf_{n-1}$	$Q_n = q_n$	
	$H_{n-1} = hf_n + H_n$	$H_n = H_{n-1} - hf_n$		$\sum_{i=1}^n q(i) = Q_1 = Q_n$

3.1.3.- DIAGRAMA DE FLUJO.



3.1.4.- EJEMPLOS

Ejemplo 1

Se tiene una tubería lateral de longitud igual a 50 metros, el espacio entre las salidas o los orificios es de 10 metros (L_o), la tubería principal tiene un diámetro nominal de 6 pulgadas (D_p), el diámetro de salida de los orificios es de 2 pulgadas (D_s), el material de la tubería es de PVC $C=150$, la presión con la que se opera el sistema es de 1 mca* (0.1Kg/cm^2) (H_o), el coeficiente de descarga en las salidas es de 0.7 (C_d), el número de salidas en la línea lateral es de 5 (n). Calcular los gastos en cada salida del sistema (litros/segundo).

Datos.

$$L = 50 \text{ m}$$

$$L_o = 10 \text{ m}$$

$$D_p = 6''$$

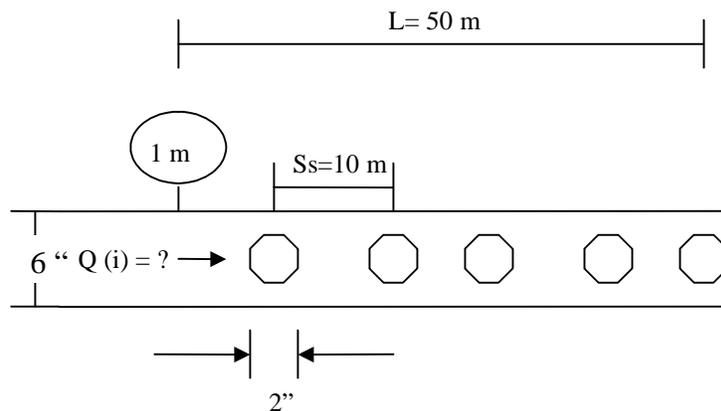
$$H_o = 1 \text{ mca}^*$$

$$C = 150 \text{ adimensional}$$

$$D_s = 2''$$

$$C_d = 0.7 \text{ adimensional}$$

$$n = 5$$



Incógnitas.

$$Q(i) = ?$$

$$q(i) = ?$$

$$H(i) = ?$$

$$h_f(i) = ?$$

Incógnita que se propone.

$$Q(1) = Q_s$$

*mca = metros de columna de agua.

[Ver resultados en el Apéndice VI, Tabla No. 1](#)

Ejemplo 2

La tubería principal de una fuente de alimentación tiene una longitud igual a 15 metros, el espacio entre las salidas o los orificios es cada 1 metro (L_o), la tubería principal tiene un diámetro nominal de 4 pulgadas (D_p), el diámetro de salida de los orificios es de 1 pulgada (D_s), el material de la tubería es de PVC C=150, la presión con la que se opera el sistema es de 1 mca* (0.1 Kg/cm^2) (H_o), el coeficiente de descarga en los emisores es de 0.7 (C_d), el número de salidas en la línea lateral es de 15 (n). Calcular los gastos en cada salida del sistema (litros/segundo), si la pendiente longitudinal es literalmente cero.

Datos.

$$L = 15 \text{ m}$$

$$L_o = 1 \text{ m}$$

$$D_p = 4''$$

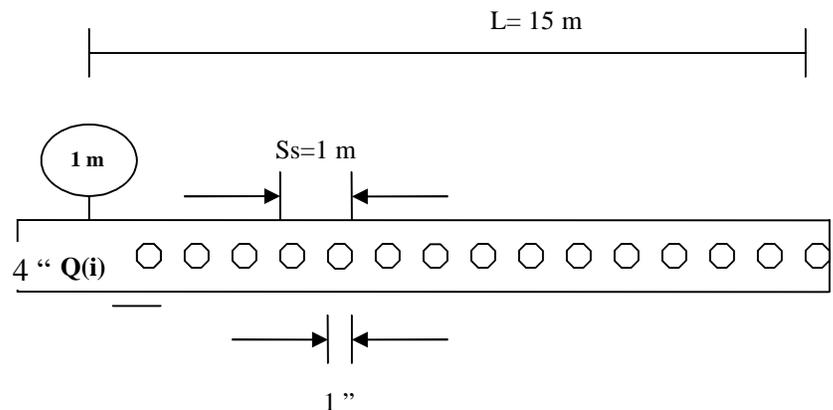
$$H_o = 1 \text{ mca}^*$$

$$C = 150 \text{ adimensional}$$

$$D_s = 1''$$

$$C_d = 0.7 \text{ adimensional}$$

$$n = 15$$



Incógnitas

$$Q(i) = ?$$

$$q(i) = ?$$

$$H(i) = ?$$

$$h_f(i) = ?$$

Incógnita que se propone.

$$Q(1) = Q_s$$

*mca = metros de columna de agua.

[Ver resultados en el Apéndice VI, Tabla No. 2](#)

Ejemplo 3

La tubería principal de una fuente de alimentación tiene una longitud igual a 150 metros, el espacio entre las salidas o los orificios es cada 50 metros (L_o), la tubería principal tiene un diámetro nominal de 2 pulgadas (D_p), el diámetro de salida de los orificios es de $\frac{1}{2}$ pulgada (D_s), el material de la tubería es de acero galvanizado $C=135$, la presión con la que se opera el sistema es de 10 mca* (1 Kg/cm^2) (H_o), el coeficiente de descarga en los emisores es de 0.7 (C_d), el número de salidas en la línea lateral es de 3 (n). Calcular los gastos en cada salida del sistema (litros/segundo), si la pendiente longitudinal es literalmente cero.

Datos.

$$L = 150 \text{ m}$$

$$L_o = 50 \text{ m}$$

$$D_p = 2''$$

$$H_o = 10 \text{ mca}^*$$

$$C = 135 \text{ adimensional}$$

$$D_s = 0.5''$$

$$C_d = 0.7 \text{ adimensional}$$

$$n = 3$$

Incógnitas.

$$Q(i) = ?$$

$$q(i) = ?$$

$$H(i) = ?$$

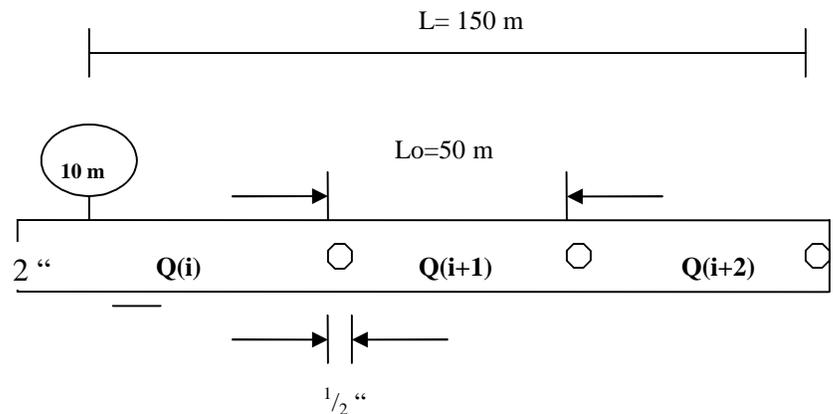
$$h_f(i) = ?$$

Incógnita que se propone.

$$Q(1) = Q_s$$

*mca = metros de columna de agua.

[Ver resultados en el Apéndice VI, Tabla No. 3](#)



3.1.5.- CÓDIGOS.

3.1.5.1.- VISUAL BASIC 6.0.

‘ Declaración de tipo de variables.

Dim i As Integer, fercho As Integer

Dim n As Integer

Dim msg1 As String

‘ Procedimiento para hacer el cálculo y entrada de las variables.

Private Sub Command1_Click()

ReDim H(n) As Double, hf(n) As Double, Q(n) As Double, P(n) As Double

‘ Declaración de variables de entrada en sus respectivos textos.

Lo = Val(Text1.Text) ‘ Entra la separación entre cada salidas múltiples.

Dp = Val(Text2.Text) ‘ Entra el diámetro que contiene a las salidas múltiples.

Ho = Val(Text3.Text) ‘ Entra la carga de presión disponible.

C = Val(Text4.Text) ‘ Entra el coeficiente de Hazen-Williams.

Ds = Val(Text5.Text) ‘ Entra el diámetro de los orificios de las salidas múltiples.

Cd = Val(Text6.Text) ‘ Entra el coeficiente de descarga del orificio.

Qs = Val(Text7.Text) ‘ Entra el gasto supuesto o inicial para iniciar el cálculo.

‘ Declaración de constantes.

Ao = 3.141592654 * ((Ds * 0.0254) ^ 2) / 4

contador = 0 ‘ Contador de cálculos durante la ejecución del programa.

si = 1 ‘ Procedimiento para iniciar el primer cálculo.

Do While si = 1 ‘ Inicia el ciclo para la ejecución del programa.

‘ mientras la condición sea verdadera (1=1)

contador = contador + 1 ‘ Contador de cálculos del programa.

If contador >= 1000 Then ‘ Procedimiento para abortar el programa.

MsgBox "No hay solución", vbCritical + vbOKOnly, "Mensaje": Exit Sub

End If

suma = 0 ‘ Para acumular el gasto total de los n orificios.

m = Ho

‘ Procedimiento que determina el número de salidas a calcular.

```

For i = 1 To n
    Q(i) = Abs(Qs)      ' Valor del gasto inicial, y hasta llegar a qn salidas.
    H(i) = Abs(Ho)     ' Valor de la carga hidráulica inicial sobre los orificios.
    ' Valor de la pérdida de carga por cada tramo de tubería.
    hf(i) = (((12100000000# * Lo) / ((Dp * 25.4) ^ 4.87)) * (Q(i) ^ 1.852)) / (C ^ 1.852)
    H(i) = Abs(H(i) - hf(i)) ' Valor de la carga hidráulica siguiente H(i+1).
    ' Calcula el gasto de salida en el orificio, (litros/segundo).
    P(i) = Abs((Ao * Cd * ((2 * 9.81 * H(i)) ^ 0.5)) * 1000)
    suma = Abs(suma + P(i)) ' Acumula los gastos de los orificios.
    Ho = H(i) ' Valor de la carga hidráulica siguiente H(i+1), para empezar
              ' nuevamente el cálculo.
    Qs = Abs(Q(i) - P(i)) ' Valor del gasto siguiente Q(i+1).
Next i

Qo = suma ' Suma de los gastos de los orificios, lo nombramos gasto obtenido.
E = Abs(Q(1) - Qo) ' Cálculo de el error absoluto.
' Condición que hace que el ciclo while continúe o aborte, y continúa ala
' siguiente sentencia del while, que es imprimir y finalizar la ejecución del
' programa.
If E > 0.0001 Then ' Si el error obtenido es mayor a 0.0001, inicia otro cálculo.
    Qs = Qo ' Nuevo gasto de inicio.
    Ho = m ' Esto es para que la carga se mantenga constante.
Else ' Termina el ciclo e imprime los datos calculados.
    si = 2
End If
' Segunda sentencia del while.
Loop
' Procedimiento para imprimir en un msflexgrid los valores calculados.
For i = 1 To n
    ' Imprime el número de salidas laterales.
    MSFlexGrid1.Row = i
    MSFlexGrid1.Col = 0

```

```
MSFlexGrid1.Text = Format$(Format$(i), "###" + "")
```

‘ Imprime el gasto de salida en los orificios.

```
MSFlexGrid1.Row = i
```

```
MSFlexGrid1.Col = 1
```

```
MSFlexGrid1.Text = Format$(Format$(P(i)), "##0.#####0" + "")
```

‘ Imprime el valor de la carga hidráulica sobre los orificios.

```
MSFlexGrid1.Row = i
```

```
MSFlexGrid1.Col = 2
```

```
MSFlexGrid1.Text = Format$(Format$(H(i)), "##0.#####0" + "")
```

‘ Imprime el gasto que fluye en cada tramo de la tubería.

```
MSFlexGrid1.Row = i
```

```
MSFlexGrid1.Col = 3
```

```
MSFlexGrid1.Text = Format$(Format$(Q(i)), "##0.#####0" + "")
```

‘ Imprime la pérdida de carga en cada tramo de la tubería.

```
MSFlexGrid1.Row = i
```

```
MSFlexGrid1.Col = 4
```

```
MSFlexGrid1.Text = Format$(Format$(hf(i)), "##0.#####0" + "")
```

Next i

```
Text9.Text = Format$(Qo, "##.#####")
```

```
Text10.Text = Format$(E, "0.#####")
```

```
Text11.Text = Format$(contador, "###")
```

End Sub

‘ Procedimiento para limpiar textos.

Private Sub Command2_Click()

```
Text1.Text = ""
```

```
Text2.Text = ""
```

```
Text3.Text = ""
```

```
Text4.Text = ""
```

```
Text5.Text = ""
```

```
Text6.Text = ""
```

```
Text7.Text = ""
Text8.Text = ""
Text9.Text = ""
Text10.Text = ""
Text11.Text = ""
    For i = 1 To n
MsflexGrid1.Row = i
MsflexGrid1.Col = 0
MsflexGrid1.Text = ""
MsflexGrid1.Row = i
MsflexGrid1.Col = 1
MsflexGrid1.Text = ""

MsflexGrid1.Row = i
MsflexGrid1.Col = 2
MsflexGrid1.Text = ""

MsflexGrid1.Row = i
MsflexGrid1.Col = 3
MsflexGrid1.Text = ""

MsflexGrid1.Row = i
MsflexGrid1.Col = 4
MsflexGrid1.Text = ""
Next i
End Sub
```

‘ Procedimiento para abortar o cerrar la aplicación del programa.

```
Private Sub Command3_Click()
End
End Sub
```

‘ Procedimiento para avanzar con la tecla ENTER.

```
Private Sub Text1_KeyPress(KeyAscii As Integer)
```

```
If KeyAscii = 13 Then
```

```
Text2.SetFocus
```

```
End If
```

```
End Sub
```

```
Private Sub Text2_KeyPress(KeyAscii As Integer)
```

```
If KeyAscii = 13 Then
```

```
Text3.SetFocus
```

```
End If
```

```
End Sub
```

```
Private Sub Text3_KeyPress(KeyAscii As Integer)
```

```
If KeyAscii = 13 Then
```

```
Text4.SetFocus
```

```
End If
```

```
End Sub
```

```
Private Sub Text4_KeyPress(KeyAscii As Integer)
```

```
If KeyAscii = 13 Then
```

```
Text5.SetFocus
```

```
End If
```

```
End Sub
```

```
Private Sub Text5_KeyPress(KeyAscii As Integer)
```

```
If KeyAscii = 13 Then
```

```
Text6.SetFocus
```

```
End If
```

```
End Sub
```

```
Private Sub Text6_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
Text7.SetFocus
End If
End Sub
```

```
Private Sub Text7_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
Text8.SetFocus
End If
End Sub
```

```
Private Sub Text8_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
n = Val(Text8.Text)
Text8.Text = Fix(n)
    If n < 1 Then ' Mensaje de caja de texto
        MsgBox "El número debe ser mayor de CERO", vbCritical + vbOKOnly, "Mensaje": Exit Sub
    End If
Grid1.SetFocus
Grid1.Rows = n + 1
Grid1.Col = 1
Grid1.Col = 2
Grid1.Col = 3
Grid1.Col = 4
Command1.SetFocus
End If
End Sub
```

' Procedimiento para activar el cursor en la primer caja de texto.

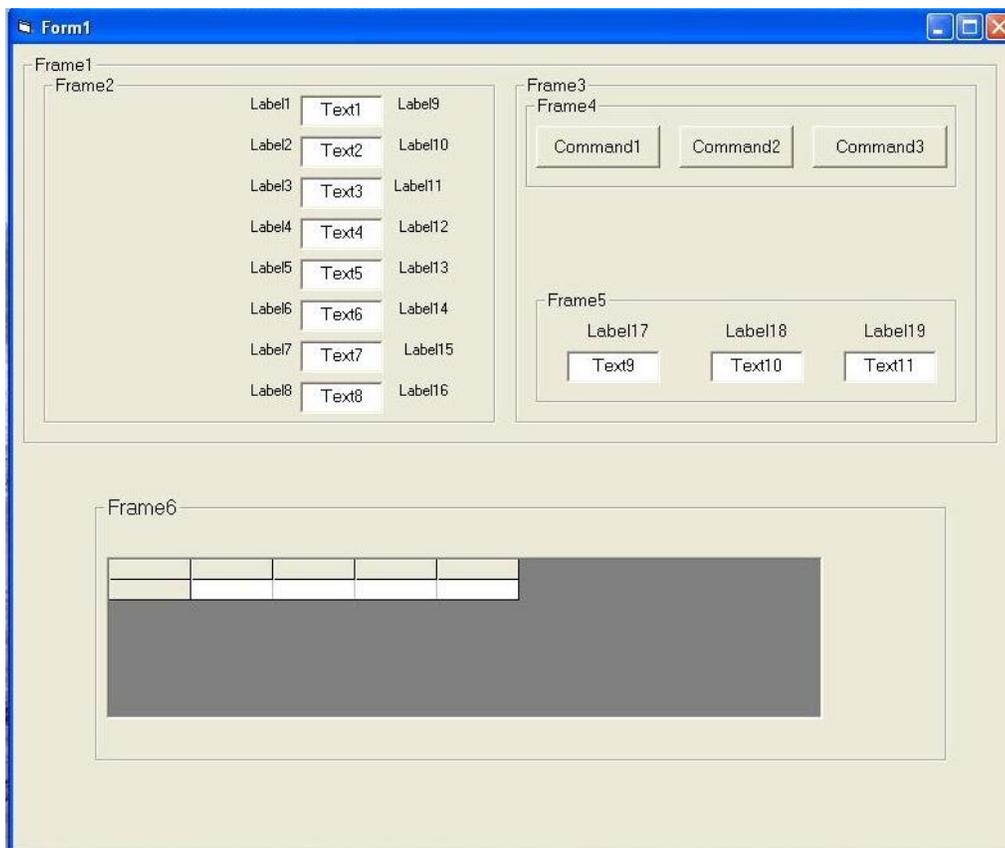
```
Private Sub Form_Activate()
Text1.SetFocus
End Sub
```

' Procedimiento para cargar los letreros de referencia en el MsflexGrid1.

```
Private Sub Form_Load()  
MsflexGrid1.ColWidth(0) = 900  
MsflexGrid1.ColWidth(1) = 2065  
MsflexGrid1.ColWidth(2) = 2005  
MsflexGrid1.ColWidth(3) = 2000  
MsflexGrid1.ColWidth(4) = 2650  
  
MsflexGrid1.TextMatrix(0, 0) = "Indices"  
MsflexGrid1.TextMatrix(0, 1) = "Gasto por orificio (lps)"  
MsflexGrid1.TextMatrix(0, 2) = "Carga en los orificios (m)"  
MsflexGrid1.TextMatrix(0, 3) = "Gasto/Tramo (l/s)"  
MsflexGrid1.TextMatrix(0, 4) = "Pérdida de carga/Tramo (m)"  
End Sub
```

Notas importantes sobre la forma del programa.

- Los **Frame(1,2,3,4,5,6)**, en este programa únicamente se utiliza para darle un toque estético.
- Las etiquetas o **Label(1,2,3,4,5,6,7,8,9, ... ,19)**, en este programa solo se utiliza para colocar letreros y reconocer a la variable de entrada.
- Los **Text(1,2,3,4,5,6,7,8)**, son las cajas para la entrada de los datos, es necesario que vaya en ese orden para que no se dificulte al copiar el código del programa.
- Los **Command(1,2,3)**, ejecutan al programa.
- En los **Text(9,10,11)**, se imprimen los resultados; Gasto de entrada, Error, número de iteraciones, respectivamente
- Colocar un **MsFlexGrid1**, solo hay que colocar en la tabla de propiedad **cols 5** (número de columnas).



3.1.5.2.- MATLAB 6.5.

```
clc          % Limpia la ventana de inicio de MATLAB.
clear all   % Limpia la memoria de MATLAB.
disp(' ')   % Comando para generar textos de impresión.
disp(' ')   % Comando para generar textos de impresión.
% Comando para generar textos de impresión.
disp(' Programa elaborado por: JUAN FERNANDO PÉREZ CALVO ')
disp(' ')   % Comando para generar textos de impresión.
% Comando para generar textos de impresión.
disp(' Cálculo de gastos y carga hidráulica en los orificios, cálculo de ')
disp(' gastos en cada tramo con sus respectivas pérdidas de carga. ')
disp(' ')
% Comando para entrada de datos, con sus respectivas especificaciones
% dentro del mismo comando y sus respectivas unidades.
Lo=input('Espacio entre orificios [metros]= ');
Dp=input('Diámetro de la principal [pulgadas]= ');
Ho=input('Presión de entrada [metros]= ');
C=input('Coef. de Hazen-William C [adimensional]= ');
Ds=input('Diámetro del orificio ID [pulgadas]= ');
Cd=input('Coef. de descarga en los emisores Cd [adimensional]= ');
Qs=input('Cuánto supones el gasto [litros/segundo]= ');
n=input('Número de salidas (S.M.L.) [adimensional]= ');
disp(' ')   % Comando para generar textos de impresión.
% Declaración de constantes.
Ao = 3.141592654 * ((Ds * 0.0254) ^ 2) / 4;
contador=0; % Contador de cálculos durante la ejecución del programa.
si=1;      % Procedimiento para iniciar el primer cálculo.
while si == 1 % Inicia el ciclo para la ejecución del programa
% mientras la condición sea verdadera (1=1).
    contador=contador + 1; % Contador de cálculos del programa.
```

```

suma=0;
m=Ho; % Asignación de la carga inicial a la variable m.
% Procedimiento que determina el número de salidas a calcular.
for i = 1 : 1 : n
    Q(i)=Qs; % Valor del gasto inicial, y hasta llegar a qn salidas.
    H(i)=Ho; % Valor de la carga hidráulica inicial sobre los orificios.
% Valor de la pérdida de carga por cada tramo de tubería.
    hf(i) = (((12100000000 * Lo) / ((Dp * 25.4) ^ 4.87)) * (Q(i) ^ 1.852)) / (C ^ 1.852);
    H(i)=H(i) - hf(i); % Valor de la carga hidráulica siguiente H(i+1).
% Calcula el gasto en el orificio, (litros/segundo).
    P(i) = abs((Ao * Cd * ((2 * 9.81 * H(i)) ^ 0.5)) * 1000);
    suma=suma + P(i); % Acumula los gastos de los orificios.
    Ho=H(i); % Valor de la carga hidráulica siguiente H(i+1), para empezar
            % nuevamente el cálculo.
    Qs=Q(i) - P(i); % Valor del gasto siguiente Q(i+1).
end % Finaliza el estatuto for.
Qo=suma; % La suma de los gastos de los orificios, lo nombramos gasto obtenido.
E=abs(Q(1) - Qo); % Cálculo de el error absoluto.
% Condición que hace que el ciclo while continúe o aborte, y continúa a la
% siguiente sentencia del while, que es imprimir y finalizar la ejecución del
% programa.
if E >= 0.0001 % Si el error obtenido es mayor a 0.0001, inicia otro cálculo.
    Qs=Qo; % Nuevo gasto de inicio.
    Ho=m; % Esto es para que la carga se mantenga constante.
else % Termina el ciclo e imprime los datos calculados.
    si=2;
end % Fin de la instrucción if.
end % Fin de la instrucción while.
disp(' ') % Comando para generar textos de impresión.
disp(' ') % Comando para generar textos de impresión.

```

```

% Procedimiento para imprimir los valores calculados.
% i = 1,2,3... n.
% Imprime el gasto de salida en los orificios.
% Imprime el valor de la carga hidráulica sobre los orificios.
% Imprime el gasto en transición en cada tramo de la tubería.
% Imprime la pérdida de carga en cada tramo de la tubería.
fprintf('=====\n')
for i = 1 : 1 : n
fprintf('Solución      q(%1.0f)=%6.5f      H(%1.0f)=%6.5f      Q(%1.0f)=%6.5f
hf(%1.0f)=%6.5f      \n',i,P(i),i,H(i),i,Q(i),i,hf(i))
end
fprintf('=====\n')
% Este procedimiento imprime el gasto entrante al sistema (Q1), el margen de error
% calculado, el total de cálculos realizados.
fprintf('=====\n')
fprintf('Solución Q(1)= %6.5f  Error= %6.5f  Iteraciones= %6.5f\n',suma,E,contador)
fprintf('=====\n')
disp(' ')

```

3.2.- FLUJO UNIFORME EN ESTADO PERMANENTE PARA CANALES, TRIANGULARES, TRAPECIALES Y RECTANGULARES; SIMÉTRICOS O ASIMÉTRICOS.

3.2.1.- DESCRIPCIÓN.

El flujo uniforme, es aquel que tomando como criterio el espacio, las características hidráulicas no cambian entre dos secciones separadas a una distancia, es decir: $(dv/dx) = 0$ ([figura 1, Apéndice IV](#)). (<http://www.fing.uach.mx>).

El flujo permanente es aquel en el que tomando como criterio el tiempo, las características hidráulicas permanecen constantes, es decir: $(dv/dt) = 0$ ([figura 2, Apéndice IV](#)), (<http://www.fing.uach.mx>).

3.2.2.- OBJETIVO DEL PROGRAMA.

Calcular el tirante normal, área hidráulica, perímetro hidráulico, radio hidráulico, número de Froude, tipo de flujo. Utilizaremos el método numérico de iteraciones sucesivas.

$$Q = \frac{Am \times Rh^{2/3} \times \sqrt{So}}{n} \quad \dots \text{Ecuación de Manning}$$

$$Rh = \frac{Am}{Pm} \quad \text{Entonces...}$$

$$\frac{Q \times n}{\sqrt{So}} = Am \times \frac{Am^{2/3}}{Pm^{2/3}}$$

$$\frac{Q \times n}{\sqrt{So}} = \frac{Am^{5/3}}{Pm^{2/3}}$$

$$Pm^{2/3} \times \frac{Q \times n}{\sqrt{So}} = Am^{5/3}$$

Nota: Consultar las literales que integran la ecuación de Manning en la pagina 94.

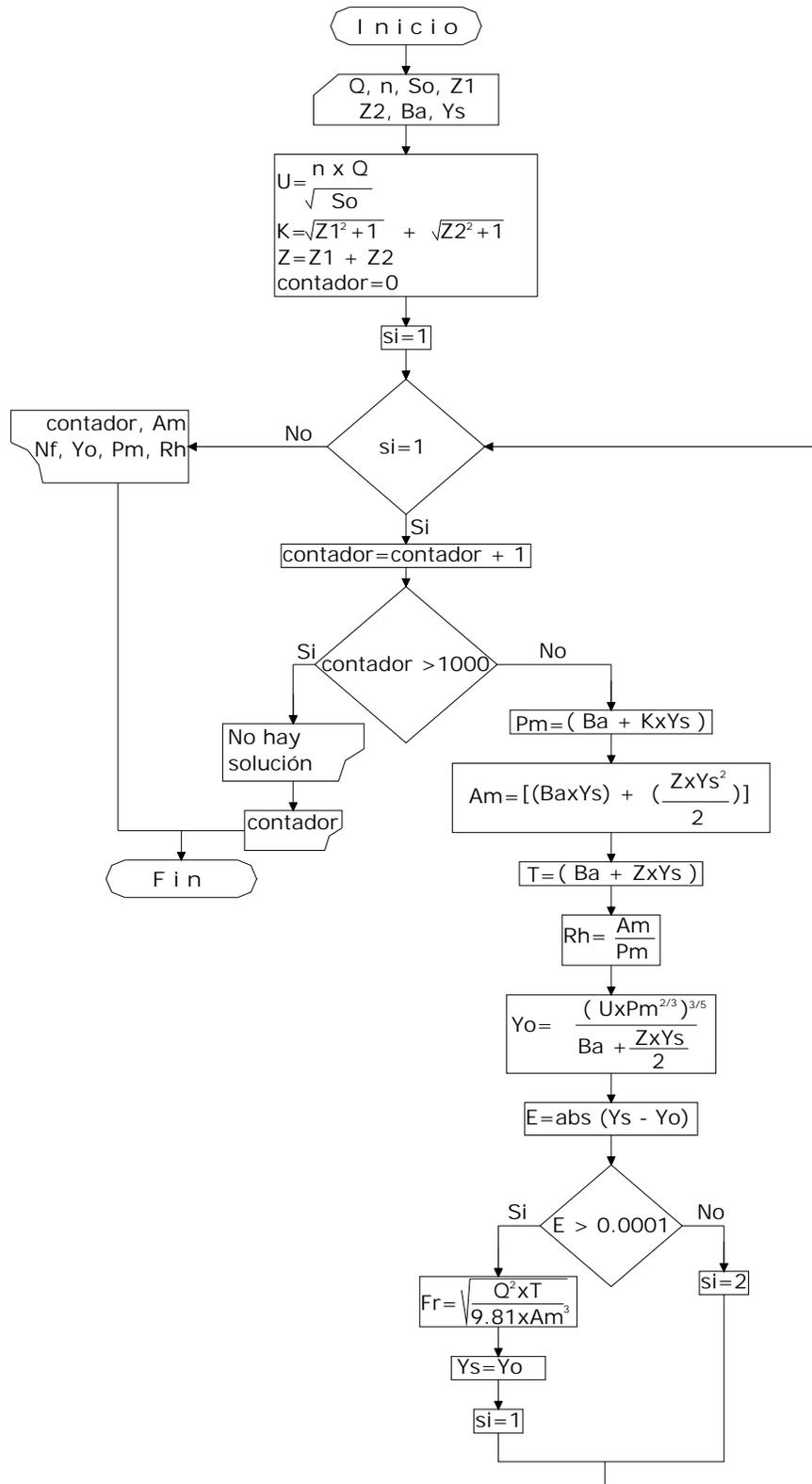
El criterio de solución del problema consiste en que el miembro derecho y miembro izquierdo de la ecuación

$$Pm^{2/3} \times \frac{Q \times n}{\sqrt{So}} = Am^{5/3}$$

tienden hacer iguales. En este caso se propuso un margen de error menor de 0.0001, es decir;

$$Absoluto(Ys - Yo) \leq 0.0001$$

3.2.3.- DIAGRAMA DE FLUJO.



3.2.4.- EJEMPLOS.

Ejemplo 1

Calcular el tirante normal de un canal rectangular por el cual fluye un gasto de $4.5 \text{ m}^3/\text{s}$ de agua (Q), con una pendiente de 0.0049 m/m (S_o), un ancho de plantilla de 1.335 metros (B_a). El coeficiente de rugosidad es de 0.012 (n). De acuerdo al número de Froude, determine si el flujo es crítico, subcrítico o supercrítico. Calcular el tirante normal (Y_n), Área hidráulica, Perímetro hidráulico. (Ver figura).

Datos:

$$Q = 4.5 \text{ m}^3/\text{s}$$

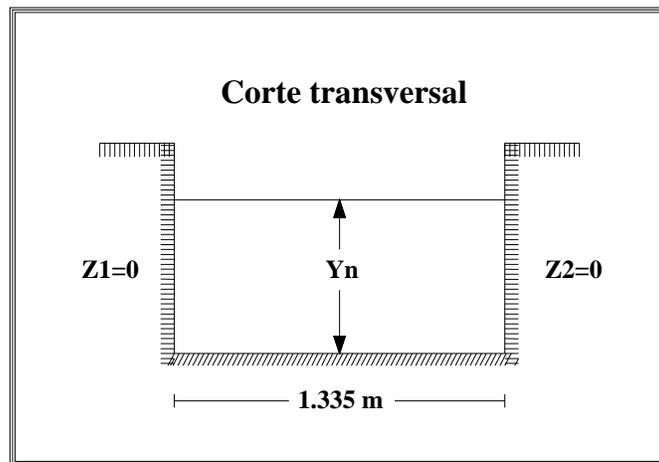
$$n = 0.012$$

$$S_o = 0.0049$$

$$Z_1 = 0$$

$$Z_2 = 0$$

$$B_a = 1.335 \text{ m}$$



Incógnitas.

$$Y_n = ?$$

$$Fr = ?$$

Incógnita que se propone:

$$Y_s$$

Solución aproximada

Tirante normal	Área hidráulica	Perímetro hidráulico	Radio hidráulico	No. de Froude Fr	Tipo de flujo
(metros)	(metros ²)	(metros)	(metros)		
1.0502	1.4021	3.4355	0.4081	0.9999	Crítico

Se hicieron 7 iteraciones sucesivas.

Ejemplo 2

Por un canal trapecial (simétrico) para un gasto de $11 \text{ m}^3/\text{s}$ de agua (Q), considerando una pendiente de 0.019 m/m (S_o) y además se estima que la rugosidad es de 0.025 (n), con un ancho de plantilla de 4.5 metros (B_a), los taludes del canal son de $3:1$ (Z_1, Z_2). Determine el tirante normal (Y_n) y diga si el flujo es crítico, subcrítico o supercrítico. (Ver figura).

Datos:

$$Q = 11 \text{ m}^3/\text{s}$$

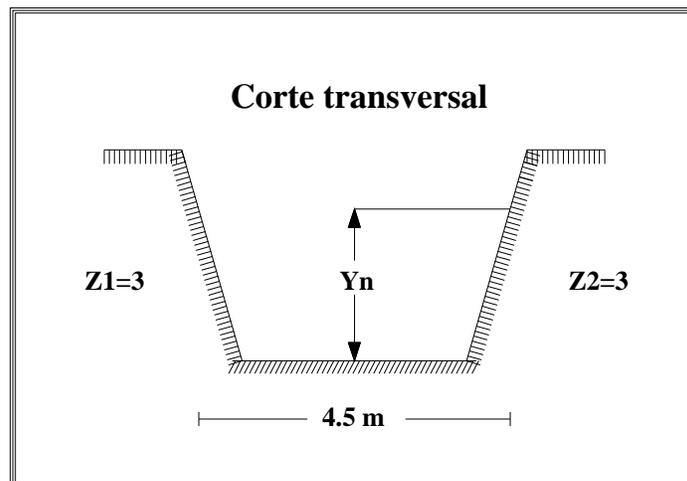
$$n = 0.025$$

$$S_o = 0.019$$

$$Z_1 = 3:1$$

$$Z_2 = 3:1$$

$$B_a = 4.5 \text{ m}$$



Incógnitas.

$$Y_n = ?$$

$$Fr = ?$$

Incógnita que se propone.

$$Y_s$$

Solución aproximada

Tirante normal	Área hidráulica	Perímetro hidráulico	Radio hidráulico	No. de Froude Fr	Tipo de flujo
(metros)	(metros ²)	(metros)	(metros)		
0.5634	3.4881	8.0637	0.4326	1.5134	Supercrítico

Se hicieron 6 iteraciones sucesivas.

Ejemplo 3

Por un canal triangular (simétrico) circula agua a razón de $0.08 \text{ m}^3/\text{s}$ (Q), trazado con una pendiente de 0.0016 m/m (S_o), con una rugosidad de 0.017 (n) y taludes de $1.5:1$ (Z_1, Z_2). Clasificar el tipo de flujo de acuerdo al número de Froude (crítico, supercrítico o subcrítico) y calcular el tirante normal (Y_n). (Ver figura).

Datos:

$$Q = 0.08 \text{ m}^3/\text{s}$$

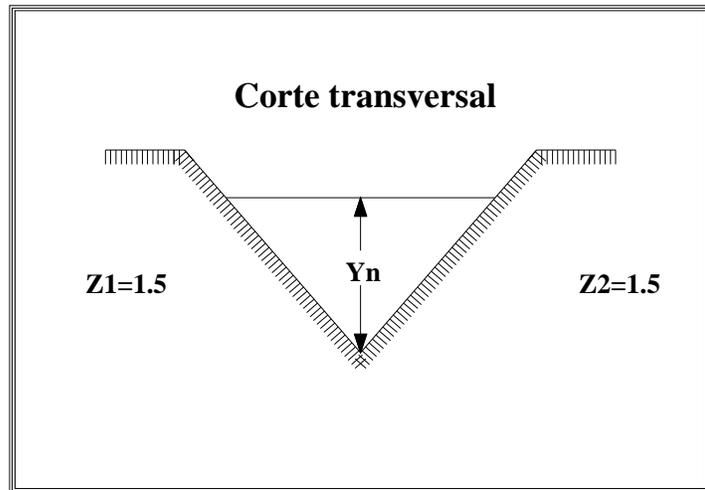
$$n = 0.017$$

$$S_o = 0.0016$$

$$Z_1 = 1.5$$

$$Z_2 = 1.5$$

$$B_a = 0 \text{ m}$$



Incógnitas.

$$Y_n = ?$$

$$Fr = ?$$

Incógnita que se propone.

Y_s

Solución aproximada

Tirante normal	Área hidráulica	Perímetro hidráulico	Radio hidráulico	No. de Froude Fr	Tipo de flujo
(metros)	(metros ²)	(metros)	(metros)		
0.30094	0.1359	1.0851	0.1252	0.4847	Subcrítico

Se hicieron 23 iteraciones sucesivas.

Ejemplo 4

Un canal trapecial (asimétrico) transporta un gasto de $4 \text{ m}^3/\text{s}$ de agua (Q). La pendiente es de 0.0038 m/m (S_o) y una rugosidad de 0.013 (n), los taludes son de $1:1$, $0:90$ (Z_1 , Z_2), respectivamente, la plantilla del canal es de 0.939 metros (B_a). Determine con el número de Froude si el flujo es crítico, subcrítico o supercrítico. Calcule el tirante normal (Y_n), (Ver figura).

Datos:

$$Q = 4 \text{ m}^3/\text{s}$$

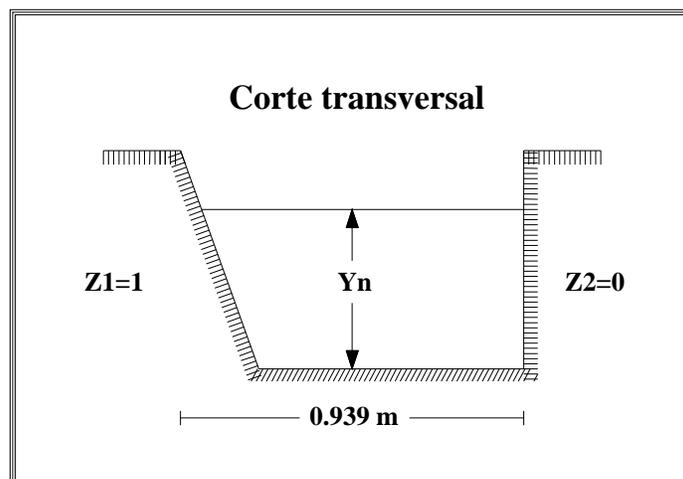
$$n = 0.013$$

$$S_o = 0.0038$$

$$Z_1 = 1:1$$

$$Z_2 = 0$$

$$B_a = 0.939 \text{ m}$$



Incógnitas.

$$Y_n = ?$$

$$Fr = ?$$

Incógnita que se propone.

$$Y_s$$

Solución aproximada

Tirante normal	Área hidráulica	Perímetro hidráulico	Radio hidráulico	No. de Froude Fr	Tipo de flujo
(metros)	(metros ²)	(metros)	(metros)		
1.01705	1.4722	3.3944	0.4337	0.9999	Crítico

Se hicieron 4 iteraciones sucesivas.

Ejemplo 5

Un canal triangular (asimétrico) conduce un gasto de $0.04 \text{ m}^3/\text{s}$ de agua (Q) con una pendiente de 0.00026 m/m (S_o) y rugosidad de 0.02 (n), los taludes son de $1.54:1$ y $2:1$ (Z_1, Z_2), respectivamente, se desea calcular el tirante normal (Y_n), así como el tipo de flujo que circula (subcrítico, crítico o supercrítico), (Ver figura).

Datos.

$$Q = 0.04 \text{ m}^3/\text{s}$$

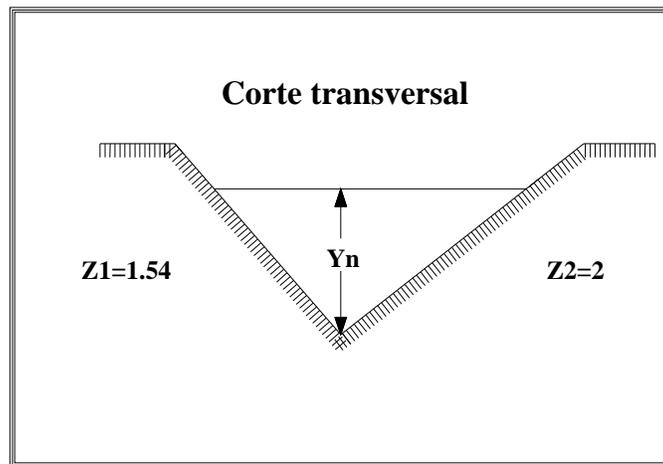
$$n = 0.02$$

$$S_o = 0.00026$$

$$Z_1 = 1.54:1$$

$$Z_2 = 2:1$$

$$B_a = 0 \text{ m}$$



Incógnitas.

$$Y_n = ?$$

$$Fr = ?$$

Incógnita que se propone.

Y_s

Solución aproximada

Tirante normal	Área hidráulica	Perímetro hidráulico	Radio hidráulico	No. de Froude Fr	Tipo de flujo
(metros)	(metros ²)	(metros)	(metros)		
0.32234	0.1839	1.3127	0.1401	0.173	Subcrítico

Se hicieron 23 iteraciones sucesivas.

Ejemplo 6

Un canal triangular (asimétrico) conduce un gasto de $0.06 \text{ m}^3/\text{seg}$ de agua (Q) con una pendiente de 0.024 m/m (S_o) y rugosidad de 0.015 (n), los taludes son $0:90$ y $1.5:1$ (Z_1, Z_2), respectivamente, se desea calcular el tirante normal (Y_n), así como el tipo de flujo que circula (subcrítico, crítico o supercrítico), (Ver figura).

Datos.

$$Q = 0.06 \text{ m}^3/\text{seg}$$

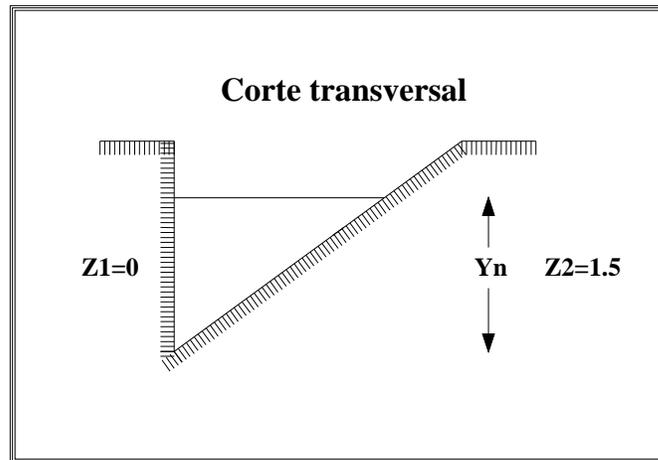
$$n = 0.015$$

$$S_o = 0.024$$

$$Z_1 = 0$$

$$Z_2 = 1.5:1$$

$$B_a = 0 \text{ m}$$



Incógnitas.

$$Y_n = ?$$

$$Fr = ?$$

Incógnita que se propone.

Y_s

Solución aproximada

Tirante normal	Área hidráulica	Perímetro hidráulico	Radio hidráulico	No. de Froude Fr	Tipo de flujo
(metros)	(metros ²)	(metros)	(metros)		
0.22466	0.0379	0.6297	0.0601	1.5098	Supercrítico

Se hicieron 23 iteraciones sucesivas.

3.2.5.- CÓDIGOS.

3.2.5.1.- VISUAL BASIC 6.0.

‘ Declaración de variables.

Dim NT As Integer, contador As Integer, i As Integer

Dim msg1 As String

‘ Lleva a cabo la entrada de los datos y ejecuta la aplicación.

Private Sub Command1_Click()

Q = Val(Text1.Text) ‘ Entrada del gasto para conducción en el canal.

n = Val(Text2.Text) ‘ Entra el coeficiente de rugosidad del canal.

So = Val(Text3.Text) ‘ Entra la pendiente del canal.

Z1 = Val(Text4.Text) ‘ Entra el talud izquierdo del canal.

Z2 = Val(Text5.Text) ‘ Entra el talud derecho del canal.

Ba = Val(Text6.Text) ‘ Entra la base del canal.

Ys = Val(Text7.Text) ‘ Entrada del tirante supuesto para iniciar el cálculo.

‘ Declaración de constantes.

$U = (Q * C) / (So ^ 0.5)$

$K = ((Z1 * Z1 + 1) ^ 0.5) + ((Z2 * Z2 + 1) ^ 0.5)$

$Z = Z1 + Z2$

contador = 0 ‘ Procedimiento para acumular una sumatoria.

si = 1 ‘ Procedimiento para empezar el bucle while .

Do While si = 1 ‘ Inicia el cálculo hasta que la condición sea verdadera.

contador = contador + 1 ‘ Procedimiento para acumular una sumatoria.

If contador > 1000 Then ‘ Procedimiento para terminar la ejecución

MsgBox "No hay solución por este método" ‘ del programa.

Exit Sub

End If

$i = \text{contador}$ ' Procedimiento para iniciar el MSFlexgrid1.
 $P_m = (B_a + (K * Y_s))$ ' Calcula el perímetro mojado.
 $A_m = ((B_a * Y_s) + (Z * Y_s * Y_s/2))$ ' Calcula el área hidráulica.
 $R_h = A_m / P_m$ ' Calcula el radio hidráulico.
 $T = B_a + Z * Y_s$ ' Calcula el ancho superficial.
 $Y_o = ((U * (P_m ^ (2 / 3))) ^ (3 / 5)) / (B_a + (Z * Y_s) / 2)$ ' Tirante obtenido.
 $E = \text{Abs}(Y_s - Y_o)$ ' Error absoluto de los tirantes obtenidos y supuestos.

' Procedimiento para imprimir el número de Iteraciones, Área hidráulica,
 ' Perímetro mojado, Radio hidráulico, Tirante supuesto y Tirante obtenido.

```

MSFlexGrid1.Row = i
MSFlexGrid1.Col = 0
MSFlexGrid1.Text = Format(contador, "##")

MSFlexGrid1.Row = i
MSFlexGrid1.Col = 1
MSFlexGrid1.Text = Format(Am, "##0.0000")

MSFlexGrid1.Row = i
MSFlexGrid1.Col = 2
MSFlexGrid1.Text = Format(Pm, "##0.0000")

MSFlexGrid1.Row = i
MSFlexGrid1.Col = 3
MSFlexGrid1.Text = Format(Rh, "##0.0000")

MSFlexGrid1.Row = i
MSFlexGrid1.Col = 4
MSFlexGrid1.Text = Format(Ys, "##0.0000")

MSFlexGrid1.Row = i
  
```

```

MSFlexGrid1.Col = 5
MSFlexGrid1.Text = Format(Yo, "##0.000000")
    If E >= 0.0001 Then ' Procedimiento que controla el ciclo, al comparar
                        ' el error absoluto con el valor propuesto.
        Ys = Yo        ' Procedimiento para asignar el nuevo valor del tirante.
' Calcula el número de Froude.
    Fr = ((Q * Q * T) / (9.81 * Am * Am * Am)) ^ 0.5
        Else
si = 2                ' Procedimiento para terminar el ciclo, siempre y cuando
                    ' el error sea menor que el valor propuesto (0.0001).

    End If
' Procedimiento para imprimir en varias cajas de texto los valores después de
' haberse cumplido la condición propuesta, con el estatuto if.
    Text8.Text = Format(Yo, "##0.#####000")
    Text9.Text = Format(E, "0.#####")
    Text10.Text = Format(contador)
    Text11.Text = Format(Ys, "##0.#####")
    Text12.Text = Format(Fr, "0.#####")
Loop                ' Fin del ciclo while.
Grid1.Rows = contador + 2 ' Determina el número de filas para un MSFlexGrid1.
NT = contador      ' Asignación de filas para el MSFlexGrid1, después de varias
                    ' iteraciones realizadas por el ciclo while.
If Fr < 0.95 Then  ' Evalúa el número de Froude calculado.
    Hola = "Subcrítico"
    Text13.Text = Hola        ' Imprime el tipo de flujo en el texto.
End If

If Fr > 1.05 Then  ' Evalúa el número de Froude calculado.
    Hola = "Supercrítico"
    Text13.Text = Hola        ' Imprime el tipo de flujo en el texto.
End If

```

```

    If Fr > 0.95 And Fr < 1.05 Then ' Evalúa el número de Froude calculado.
        Hola = "Crítico"
        Text13.Text = Hola ' Imprime el tipo de flujo en el texto.
    End If
End Sub ' Fin del command1.

```

' Procedimiento para limpiar todos los textos (resultados), que se imprimen.

```
Private Sub Command2_Click()
```

' Limpia todas las cajas de texto.

```
Text1.Text = ""
```

```
Text2.Text = ""
```

```
Text3.Text = ""
```

```
Text4.Text = ""
```

```
Text5.Text = ""
```

```
Text6.Text = ""
```

```
Text7.Text = ""
```

```
Text8.Text = ""
```

```
Text9.Text = ""
```

```
Text10.Text = ""
```

```
Text11.Text = ""
```

```
Text12.Text = ""
```

```
Text13.Text = ""
```

' Limpia los datos del MSFlexGrid1 (filas hasta *n*, y columnas 1,2,3,4,5).

```
For i = 1 To NT
```

```
MSFlexGrid1.Row = i
```

```
MSFlexGrid1.Col = 0
```

```
MSFlexGrid1.Text = ""
```

```
MSFlexGrid1.Row = i
```

```
MSFlexGrid1.Col = 1
MSFlexGrid1.Text = ""
```

```
MSFlexGrid1.Row = i
MSFlexGrid1.Col = 2
MSFlexGrid1.Text = ""
MSFlexGrid1.Row = i
MSFlexGrid1.Col = 3
MSFlexGrid1.Text = ""
```

```
MSFlexGrid1.Row = i
MSFlexGrid1.Col = 4
MSFlexGrid1.Text = ""
```

```
MSFlexGrid1.Row = i
MSFlexGrid1.Col = 5
MSFlexGrid1.Text = ""
```

```
Next i           ' Fin del ciclo for.
End Sub         ' Finaliza el command2.
```

```
Private Sub Command3_Click() ' Cierra la aplicación que estamos usando.
End
End Sub                ' Finaliza el command3.
```

```
Private Sub Form_Activate() ' Activa la primer caja de texto.
Text1.SetFocus
End Sub
```

' Procedimiento para brincar de una caja de texto a otra con la tecla ENTER.

```
Private Sub Text1_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
Text2.SetFocus
```

```
End If  
End Sub
```

```
Private Sub Text2_KeyPress(KeyAscii As Integer)  
If KeyAscii = 13 Then  
Text3.SetFocus  
End If  
End Sub
```

```
Private Sub Text3_KeyPress(KeyAscii As Integer)  
If KeyAscii = 13 Then  
Text4.SetFocus  
End If  
End Sub
```

```
Private Sub Text4_KeyPress(KeyAscii As Integer)  
If KeyAscii = 13 Then  
Text5.SetFocus  
End If  
End Sub
```

```
Private Sub Text5_KeyPress(KeyAscii As Integer)  
If KeyAscii = 13 Then  
Text6.SetFocus  
End If  
End Sub
```

```
Private Sub Text6_KeyPress(KeyAscii As Integer)  
If KeyAscii = 13 Then  
Text7.SetFocus  
End If  
End Sub
```

```

Private Sub Text7_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
Command1.SetFocus
End If
End Sub
' Permite cargar algunas propiedades para el MSFlexGrid1 (en particular
' los encabezados; fila 0 y columna 0,1,2,3,4,5)
Private Sub Form_Load()
Text7.Enabled = False
Grid1.ColWidth(0) = 1000
Grid1.ColWidth(1) = 800
Grid1.ColWidth(2) = 1000
Grid1.ColWidth(3) = 800
Grid1.ColWidth(4) = 1500
Grid1.ColWidth(5) = 1500
Grid1.TextMatrix(0, 0) = "Iteraciones"
Grid1.TextMatrix(0, 1) = "Áreas"
Grid1.TextMatrix(0, 2) = "Perímetros"
Grid1.TextMatrix(0, 3) = "Radios"
Grid1.TextMatrix(0, 4) = "Tirantes supuestos"
Grid1.TextMatrix(0, 5) = "Tirantes obtenidos"
End Sub

```

Notas importantes sobre la forma del programa.

- Los **Frame(1,2,3,4,5,6,7,8)**, en este programa únicamente se utiliza para darle un toque estético.
- Las **Label(1,2,3, ... 20)** se utilizó en este programa para darle nombre a la variable de entrada.
- Los **Text(1,2,3,4,5,6,7)** en este programa se utiliza para entrada de las variables a emplear en el cálculo, se recomienda que vaya en el orden en que se muestra para no alterar el código al copiarlo.
- Los **Text(8,9,10,11,12,13)** se emplea en este programa para impresión de algunos resultados.
- El **MSFlexGrid1** para imprimir resultados de los cálculos y en la ventana de propiedades colocar el valor de 6 en *Cols* y 100 en *Rows*.

The screenshot shows a Windows form titled "Form1" with a blue title bar. The form contains several nested frames and controls:

- Frame1** (outermost frame) contains:
 - Frame2** (top-left area) containing:
 - Frame3** (left side) with labels Label1-Label6 and text boxes Text1-Text3.
 - Labels Label7-Label9 and text boxes Text4-Text6.
 - Labels Label13 and Text7.
 - Frame4** (right side) containing three command buttons: Command1, Command2, and Command3.
- Frame5** (middle area) containing labels Label15-Label18 and text boxes Text8-Text11.
- Frame6** (bottom-left area) containing:
 - Frame7** (top-left of Frame6) containing a small table with 2 columns and 2 rows.
 - A large grey rectangular area below Frame7.
- Frame8** (bottom-right area) containing labels Label19 and Label20, and text boxes Text12 and Text13.

Vista de diseño del formulario, y con esto solo falta copiar el código del programa y pegarlo en la ventana de codificación, para poder ejecutarlo.

Form1

Entrada de datos:

Gasto del canal = m3/seg Talud 1= Adimensional

Coef. de rugosidad = Adimensional Talud 2= Adimensional

Pendiente del canal = Adimensional Ancho del canal = metros

Tirante supuesto = metros

Proceso de datos:

Calcular

Limpiar

Salir

Salida de datos:

Tirante normal Error Iteraciones Tirante

Salida de datos:

Iteraciones	Áreas	Perímetros	Radios	Tirante supuesto	Tirante obtenido

Número de Froude

Tipo de flujo

3.2.5.2.- MATLAB 6.5.

```
clc          % Limpia la ventana de inicio de MATLAB.
clear all   % Limpia la memoria de MATLAB.
disp(' ')   % Comando para generar textos de impresión.
disp(' ')   % Comando para generar textos de impresión.
% Comando para generar textos de impresión, mi presentación.
disp(' Programa elaborado por: JUAN FERNANDO PÉREZ CALVO ')
disp(' ')
disp(' Cálculo del tirante normal para flujo uniforme en estado permanente ')
disp(' para canales Triangulares, Trapeciales, Rectangulares; ')
disp(' Simétricos o Asimétricos ')
disp(' ')
disp(' ')
% Comando para entrada de datos, con sus respectivas especificaciones
% dentro del mismo comando y sus respectivas unidades.
Q = input('Gasto del canal [mc/seg] Q= ');
n = input('Coef. de rugosidad [adimensional] n= ');
So = input('Pendiente del canal [m/m] So= ');
Z1 = input('Talud izqdo. del canal [adimensional] Z1= ');
Z2 = input('Talud der. del canal [adimensional] Z2= ');
Ba = input('Base de la plantilla [metros] Ba= ');
Ys = input('Tirante que se propone [metros] Ys= ');
disp(' ') % Comando para generar textos de impresión.
% Declaración de constantes.
U = (Q * n) / (So ^ 0.5);
K = ((Z1 * Z1 + 1) ^ 0.5) + ((Z2 * Z2 + 1) ^ 0.5);
Z = Z1 + Z2;

contador = 0; % Contador de cálculos durante la ejecución del programa.
si = 1; % Procedimiento para iniciar el primer cálculo.
```

```

while si == 1          % Inicia el ciclo para la ejecución del programa.
contador = contador + 1;

% Procedimiento para terminar la ejecución del programa.
if contador > 1000
disp( 'No hay solución por este método') % Comando para generar textos de
                                         % impresión.

end

i = contador;
Pm = (Ba + (K * Ys));          % Calcula el perímetro mojado.
Am = ((Ba * Ys) + (Z * Ys * Ys / 2)); % Calcula el área hidráulica.
Rh = (Am / Pm);              % Calcula el radio hidráulico.
T = Ba + Z * Ys;            % Calcula el ancho superficial.
Yo = ((U * (Pm ^ (2 / 3))) ^ (3 / 5)) / (Ba + (Z * Ys) / 2); % Tirante obtenido.
E = abs(Ys - Yo);          % Error absoluto de los tirantes obtenidos y propuestos.

% Procedimiento para imprimir el tirante propuesto, Área hidráulica,
% Perímetro mojado, Radio hidráulico, y Tirante obtenido.
fprintf('Tirante propuesto.(%2.0f)=%6.4f  Área propuesta.(%2.0f)=%6.4f  Perímetro
propuesto.(%2.0f)=%6.4f          Radio propuesto.(%2.0f)=%6.4f          Tirante
obtenido.(%2.0f)=%6.6f\n',i,Ys,i,Am,i,Pm,i,Rh,i,Yo)
if E >= 0.0001    % Procedimiento que controla el ciclo, al comparar
                 % el error absoluto con el valor propuesto.
Ys = Yo;        % Procedimiento para asignar el nuevo valor del tirante.
Fr = ((Q * Q * T) / (9.81 * Am * Am * Am)) ^ 0.5; % Calcula el número de
                                                % Froude.

else
si = 2;        % Procedimiento para terminar el ciclo, siempre y cuando
              % el error sea menor que el valor propuesto (0.0001).

end

end

```

```

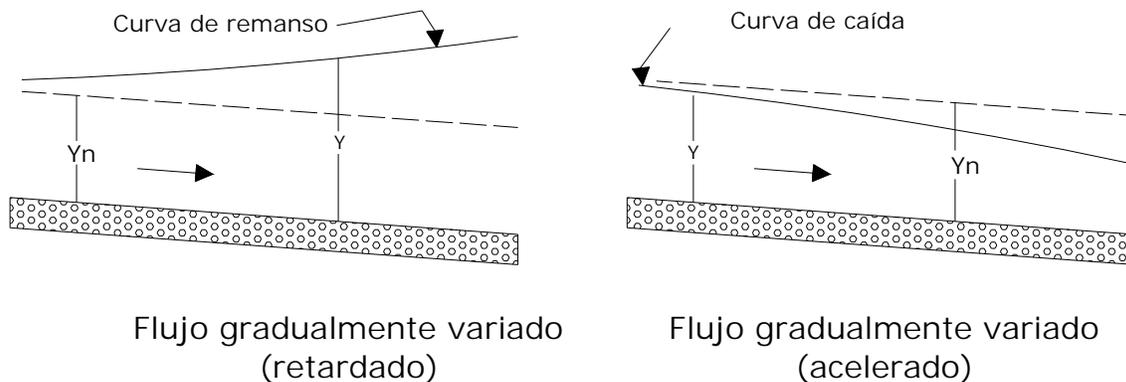
if Fr <= 0.95                % Evalúa el número de Froude calculado.
    Hola='El flujo es subcrítico'; % Imprime el tipo de flujo en el texto.
    elseif Fr >= 1.05        % Evalúa el número de Froude calculado.
        Hola='El flujo es supercrítico'; % Imprime el tipo de flujo en el texto.
    elseif Fr > 0.95 & Nf <1.05 % Evalúa el número de Froude calculado.
        Hola='El flujo es crítico'; % Imprime el tipo de flujo en el texto.
end                            % Finaliza el estatuto if.
disp(' ') % Comando para generar textos de impresión.
% Procedimiento para imprimir el Tirante obtenido, Área hidráulica,
% Perímetro mojado, Radio hidráulico, y total de cálculos realizados.
fprintf('=====\n')
fprintf('Tirante real=%6.6f    Área real=%6.4f    Perímetro real=%6.4f    Radio
real=%6.4f    Se hicieron %3.0f cálculos\n',Yo,Am,Pm,Rh,i)
fprintf('=====\n')
fprintf('El número de Froude es=%6.6f\n',Fr)
fprintf('=====\n')
disp(                                Hola) % Llamando la rutina Hola.
fprintf('=====\n')

```

3.3 FLUJO GRADUALMENTE VARIADO (FGV).

3.3.1.- DESCRIPCIÓN.

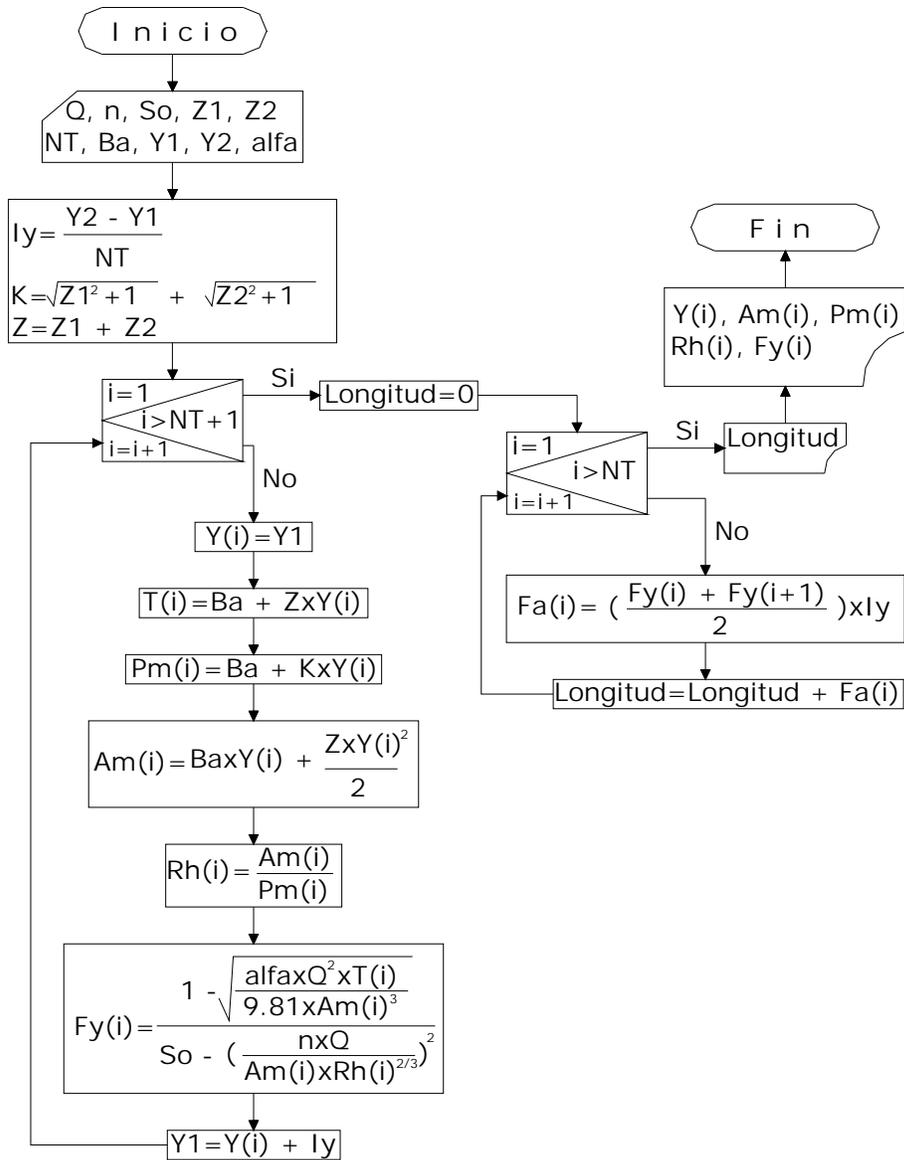
El flujo variado puede ser clasificado como rápidamente variado o gradualmente variado. En el primer caso (rápidamente variado) la profundidad de flujo cambia abruptamente en una distancia comparativamente corta, por ejemplo en un resalto hidráulico. En el otro caso (gradualmente variado), se requieren distancias mayores para que alcancen a desarrollarse los perfiles de flujo gradualmente variado. En un canal con flujo permanente uniforme pueden existir causas que retardan o aceleran la corriente de forma que pasa a condiciones variadas que se manifiestan por un aumento o disminución de la profundidad del flujo, respectivamente.



3.3.2.- OBJETIVO DEL PROGRAMA.

Calcular la longitud del flujo gradualmente variado en canales uniformes (Simétricos o Asimétricos), rectangulares, trapeciales, triangulares.

3.3.3.- DIAGRAMA DE FLUJO.



3.3.4.- EJEMPLOS.

Ejemplo 1

Un canal trapecial simétrico de mampostería, con ancho de plantilla de 6.1 metros (B_a), tiene un coeficiente de rugosidad de 0.025 (n), con taludes 2:1 (Z_1, Z_2) la pendiente con la que está trazado el canal es de 1 metro de desnivel por 1000 metro de longitud (S_o), el gasto o caudal que transporta es de $28 \text{ m}^3/\text{seg}$ (Q). Los tirantes alternos son de 1.14 metros aguas abajo (Y_1) y 1.71 metros aguas arriba (Y_2). Si este canal termina en una caída libre, determine la longitud del flujo gradualmente variado acelerado.

Datos.

$$Q = 28 \text{ m}^3/\text{s}$$

$$S_o = 1 \text{ m}/1000 \text{ m} = 0.001$$

$$B_a = 6.1 \text{ m}$$

$$Z_1 = 2$$

$$Z_2 = 2$$

$$\text{Coriolis} = 1$$

$$Y_1 = 1.14 \text{ m}$$

$$Y_2 = 1.71 \text{ m}$$

$$n = 0.025$$

Dato propuesto

Número de trapecios = A mayor segmentos, mayor aproximación.

Solución.

Para ser más precisos se propuso 2,000 tirantes. Dando una longitud total de **-242.518 m aproximadamente.**

Ejemplo 2

Después de pasar por el aliviadero de una presa, un gasto de $243 \text{ m}^3/\text{s}$ (Q) pasan a través de un cuenco de hormigón rugosidad de 0.013 (n) con pendiente casi nula (S_o). La velocidad del agua medida en la base del aliviadero es de 12.6 metros/segundo (V_1) y la anchura del cuenco o canal es de 54 metros (B_a). Estas condiciones producirán un flujo gradualmente variado retardado. Los tirantes (Y_1, Y_2) calculados fueron de 0.357 metros y 0.405 metros respectivamente. A fin de que el flujo gradualmente variado retardado esté dentro del canal. Con qué longitud deberá construirse el cuenco.

Datos

$$Q = 243 \text{ m}^3/\text{s}$$

$$S_o \approx 0.0$$

$$B_a = 54 \text{ m}$$

$$Z_1 = 0$$

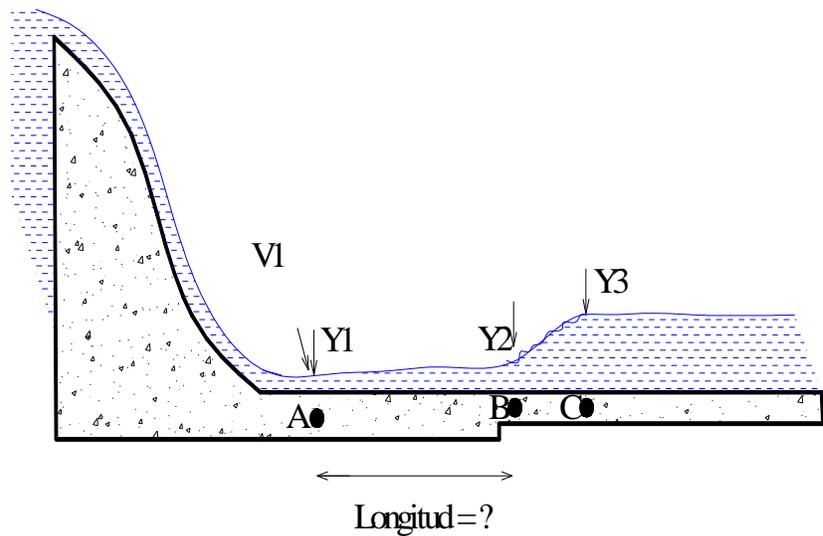
$$Z_2 = 0$$

$$\text{Coriolis} = 1$$

$$Y_1 = 0.357 \text{ m}$$

$$Y_2 = 0.405 \text{ m}$$

$$n = 0.013$$



Dato propuesto

Número de trapecios = A mayor segmentos, mayor aproximación.

Solución.

Para ser más precisos **se propuso 100 tirantes**. Dando como longitud total de **20 metros aproximadamente**, es decir del **punto A** trazar el canal rectangular con las mismas características a **20 m** hasta el **punto B** para que el flujo gradualmente variado retardado se forme dentro del canal.

3.3.5.- CÓDIGOS.

3.3.5.1.- VISUAL BASIC 6.0.

‘ Declaración de variables.

Dim NT As Integer

‘ Entrada de las variables e inicia la aplicación del programa.

Private Sub Command1_Click()

Dim i As Integer

Dim Longitud, Q As Variant

ReDim Y(NT + 1) As Double, T(NT + 1) As Double, Pm(NT + 1) As Double, A(NT + 1) As Double, Rh(NT + 1) As Double

ReDim Num(NT + 1) As Double, Den(NT + 1) As Double, Fy(NT + 1) As Double

ReDim Fa(NT) As Double

Q = Val(Text1.Text) ‘ Entrada del Gasto que circula en el canal.

So = Val(Text2.Text) ‘ Entrada de la pendiente del canal.

Ba = Val(Text3.Text) ‘ Entrada de la base o plantilla del canal.

Z1 = Val(Text4.Text) ‘ Entrada del talud izquierdo del canal.

Z2 = Val(Text5.Text) ‘ Entrada del talud derecho del canal.

alfa = Val(Text6.Text) ‘ Entrada del coeficiente de Coriolis.

Y1 = Val(Text7.Text) ‘ Entrada del tirante aguas abajo del canal.

Y2 = Val(Text8.Text) ‘ Entrada del tirante aguas arriba del canal.

n = Val(Text9.Text) ‘ Entrada de la rugosidad del canal
(coeficiente de Manning).

‘ Declaración de constantes.

ly = (Y2 - Y1) / NT

K = ((Z1 ^ 2) + 1) ^ 0.5 + ((Z2 ^ 2) + 1) ^ 0.5

Z = Z1 + Z2

‘ Inicia la asignación de datos para el cálculo de las variables.

For i = 1 To NT + 1

Y(i) = Y1 ‘ Inicia el cálculo del Tirante.

T(i) = Ba + (Z * Y(i)) ‘ Calcula el ancho superficial del canal.

$Pm(i) = Ba + (K * Y(i))$ ' Calcula el perímetro mojado del canal.

$Am(i) = (Ba * Y(i)) + (Z * (Y(i) ^ 2) / 2)$ ' Calcula el área hidráulica.

$Rh(i) = Am(i) / Pm(i)$ ' Calcula el radio hidráulico.

$Num(i) = 1 - ((alfa * (Q^2) * T(i)) / (9.81 * (A(i)^3)))$ ' Parte de la función.

$Den(i) = So - (((n * Q) / Am(i))^2 * (1 / (Rh(i)^{4/3})))$ ' Parte de la función.

$Fy(i) = (Num(i) / Den(i))$ ' Valores de la función a integrar.

$Y1 = Y(i) + ly$ ' Nuevo tirante calculado.

Next i ' Finaliza el primer cálculo de asignación de tirantes con incrementos.

Longitud = 0

For i = 1 To NT ' Instrucción que calcula el área bajo la curva.

$Fa(i) = (Fy(i) + Fy(i + 1)) * ly / 2$ ' Fórmula para calcular las áreas.

Longitud = Longitud + Fa(i) ' Acumula la sumatoria de las áreas.

' Calcula el área o longitud con respecto al tirante antecesor y sucesor.

MSFlexGrid1.Row = i + 1

MSFlexGrid1.Col = 6

MSFlexGrid1.Text = Format\$(Format\$(Fa(i)), "####0.0000" + "")

' Calcula el área o longitud total acumulados de la función propuesta.

MSFlexGrid1.Row = i + 1

MSFlexGrid1.Col = 7

MSFlexGrid1.Text = Format\$(Format\$(Longitud), "####0.0000" + "")

' Longitud del flujo gradualmente variado.

Text11.Text = Format\$((Longitud), "####0.#####" + "")

Next i

' Imprime valores de: número de tirantes, tirantes con incrementos,

' áreas, perímetro hidráulico y radio hidráulico respecto a los tirantes, y

' valor de la función a integrar.

For i = 1 To NT + 1

MSFlexGrid1.Row = i

MSFlexGrid1.Col = 0

MSFlexGrid1.Text = Format\$(Format\$(i), "####" + "")

MSFlexGrid1.Row = i

```

MSFlexGrid1.Col = 1
MSFlexGrid1.Text = Format$(Format$(Y(i)), "#####.0000" + "")
MSFlexGrid1.Row = i
MSFlexGrid1.Col = 2
MSFlexGrid1.Text = Format$(Format$(Am(i)), "#####.0000" + "")
MSFlexGrid1.Row = i
MSFlexGrid1.Col = 3
MSFlexGrid1.Text = Format$(Format$(Pm(i)), "#####.0000" + "")
MSFlexGrid1.Row = i
MSFlexGrid1.Col = 4
MSFlexGrid1.Text = Format$(Format$(Rh(i)), "###0.0000" + "")
MSFlexGrid1.Row = i
MSFlexGrid1.Col = 5
MSFlexGrid1.Text = Format$(Format$(Fy(i)), "#####0.0000" + "")

```

Next i

End Sub ' Finaliza el comando de cálculo.

' Comando para limpiar las cajas de textos después de un cálculo.

Private Sub Command2_Click()

For i = 1 To NT + 1

```

MsFlexGrid1.Row = i
MsFlexGrid1.Col = 0
MsFlexGrid1.Text = ""

```

```

MsFlexGrid1.Row = i
MsFlexGrid1.Col = 1
MsFlexGrid1.Text = ""
MsFlexGrid1.Row = i
MsFlexGrid1.Col = 2
MsFlexGrid1.Text = ""

```

MsFlexGrid1.Row = i
MsFlexGrid1.Col = 3
MsFlexGrid1.Text = ""

MsFlexGrid1.Row = i
MsFlexGrid1.Col = 4
MsFlexGrid1.Text = ""

MsFlexGrid1.Row = i
MsFlexGrid1.Col = 5
MsFlexGrid1.Text = ""

MsFlexGrid1.Row = i
MsFlexGrid1.Col = 6
MsFlexGrid1.Text = ""
MsFlexGrid1.Row = i
MsFlexGrid1.Col = 7
MsFlexGrid1.Text = ""

Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
Text5.Text = ""
Text6.Text = ""
Text7.Text = ""
Text8.Text = ""
Text9.Text = ""
Text10.Text = ""
Text11.Text = ""

```

    Next i          ' Finaliza el estatuto for.
End Sub          ' Fin del command2 "Limpiar".
' Finaliza o cierra la aplicación del programa.
Private Sub Command3_Click()
End
End Sub
' Procedimiento para cargar el MsFlexGrid1 que estamos usando.
Private Sub Form_Load()
MsFlexGrid1.ColWidth(0) = 900          ' Definimos el ancho de la columna 0.
MsFlexGrid1.ColWidth(1) = 1720       ' Definimos el ancho de la columna 1.
MsFlexGrid1.ColWidth(2) = 1300       ' Definimos el ancho de la columna 2.
MsFlexGrid1.ColWidth(3) = 1350       ' Definimos el ancho de la columna 3.
MsFlexGrid1.ColWidth(4) = 1350       ' Definimos el ancho de la columna 4.
MsFlexGrid1.ColWidth(5) = 1600       ' Definimos el ancho de la columna 5.
MsFlexGrid1.ColWidth(6) = 1720       ' Definimos el ancho de la columna 6.
MsFlexGrid1.ColWidth(7) = 1710       ' Definimos el ancho de la columna 7.
' Cargamos en la fila cero y en cada respectiva columna los letreros que se
' mencionan.
MsFlexGrid1.TextMatrix(0, 0) = "Iteraciones"
MsFlexGrid1.TextMatrix(0, 1) = "Tirantes calculados"
MsFlexGrid1.TextMatrix(0, 2) = "Valor del área"
MsFlexGrid1.TextMatrix(0, 3) = "Perímetro mojado"
MsFlexGrid1.TextMatrix(0, 4) = "Radio hidráulico"
MsFlexGrid1.TextMatrix(0, 5) = "Valor de la función"
MsFlexGrid1.TextMatrix(0, 6) = "Áreas de la Función"
MsFlexGrid1.TextMatrix(0, 7) = "Longitud acumulada"
End Sub
' Procedimiento para brincar de una caja de texto a otra con la tecla ENTER.
Private Sub Text1_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then

```

```
Text2.SetFocus
```

```
End If
```

```
End Sub
```

```
Private Sub Text2_KeyPress(KeyAscii As Integer)
```

```
If KeyAscii = 13 Then
```

```
Text3.SetFocus
```

```
End If
```

```
End Sub
```

```
Private Sub Text3_KeyPress(KeyAscii As Integer)
```

```
If KeyAscii = 13 Then
```

```
Text4.SetFocus
```

```
End If
```

```
End Sub
```

```
Private Sub Text4_KeyPress(KeyAscii As Integer)
```

```
If KeyAscii = 13 Then
```

```
Text5.SetFocus
```

```
End If
```

```
End Sub
```

```
Private Sub Text5_KeyPress(KeyAscii As Integer)
```

```
If KeyAscii = 13 Then
```

```
Text6.SetFocus
```

```
End If
```

```
End Sub
```

```
Private Sub Text6_KeyPress(KeyAscii As Integer)
```

```
If KeyAscii = 13 Then
```

```
Text7.SetFocus
```

```
End If
End Sub
```

```
Private Sub Text7_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
Text8.SetFocus
End If
End Sub
```

```
Private Sub Text8_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
Text9.SetFocus
End If
End Sub
```

```
Private Sub Text9_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
Text10.SetFocus
End If
End Sub
```

```
Private Sub Text10_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
NT = Val(Text10.Text) ' Entrada del número de trapecios para calcular la longitud del
                    ' flujo gradualmente variado.
MSFlexGrid1.SetFocus
MSFlexGrid1.Rows = NT + 2
MSFlexGrid1.Col = 1
MSFlexGrid1.Col = 2
MSFlexGrid1.Col = 3
MSFlexGrid1.Col = 4
```

```
MSFlexGrid1.Col = 5  
MSFlexGrid1.Col = 6  
MSFlexGrid1.Col = 7  
Command1.SetFocus  
End If  
End Sub
```

Notas importantes sobre la forma del programa.

- Los **Frame(1,2,3,4,5)**, no son necesarios ya que este solo sirve para darle un toque estético al programa.
- Las etiquetas o **Label(1,2,3,4,5,6,7,8,9,10,11)**, solo se emplea para darle nombre o para reconocer a la variable de entrada.
- Los **Text(1,2,3,4,5,6,7,8,9,10)**, es necesario que vaya en ese orden, es decir el **Text10** tiene que ir al final de todos ellos.
- Los **Command(1,2,3)**, colocar el letrero de Calcular, Limpiar y Salir o Fin, respectivamente.
- En el **MSFlexGrid1**, irse a la propiedad y colocar 8 en *cols*.
- En el **Text11** únicamente se imprimirá de forma total el valor de la longitud del flujo gradualmente variado.

The screenshot shows a Windows application window titled "Form1". The form contains several nested frames and controls:

- Frame1** (outermost container):
 - Frame2** (top section):
 - Frame3** (top row of controls):
 - Label1, Text1, Label4
 - Label7, Text4, Label11
 - Label15, Text7, Label18
 - Label2, Text2, Label5
 - Label8, Text5, Label12
 - Label16, Text8, Label19
 - Label3, Text3, Label6
 - Label9, Text6, Label13
 - Label17, Text9, Label20
- Label10, Text10, Label14 (centered below Frame3)
- Frame4** (bottom row of controls):
 - Command1
 - Command2
 - Command3

- Frame5** (bottom left section):
- Frame6** (MSFlexGrid): A grid with 8 columns and 2 rows. The top row is white, and the bottom row is shaded gray.
- Frame7** (bottom right section):
- Label21, Text11, Label22

Vista de diseño del formulario, y con esto solo falta copiar el código del programa y pegarlo en la ventana de codificación, para poder ejecutarlo.

The screenshot shows a Windows application window titled "Form1". The form is organized into three main sections:

- Entrada de datos:** A group box containing input fields for:
 - Gasto del canal = [] m3/seg
 - Talud 1 = [] Adimensional
 - Tirante 1 = [] metros
 - Pendiente del canal = [] Adimensional
 - Talud 2 = [] Adimensional
 - Tirante 2 = [] metros
 - Ancho del canal = [] metros
 - Coef. de Coriolis = [] Adimensional
 - Coef. de rugosidad = [] Adimensional
 - No. de trapecios = [] Adimensional
- Proceso de datos:** A group box containing three buttons: "Calcular", "Limpiar", and "Salir".
- Salida de datos:** A group box containing:
 - A table with 6 columns: Iteraciones, Tirantes calculados, Valor del área, Perímetro mojado, Radio hidráulico, and Valor de la función. The first row is highlighted in green.
 - A scrollable area below the table.
 - A field for "Longitud = [] metros".

3.3.5.2.- MATLAB 6.5.

```
clc          % Limpia la ventana de inicio de MATLAB.
clear all   % Limpia la memoria de MATLAB.
            % Comando para generar textos de impresión.

disp(' ')
disp(' ')
disp(' Programa elaborado por: JUAN FERNANDO PÉREZ CALVO ')
disp(' ')
disp('          Cálculo de la longitud del flujo gradualmente variado ')
disp('          para canales triangulares, trapeciales, rectangulares. ')
disp(' ')
% Comando para entrada de datos, con sus respectivas especificaciones
% dentro del mismo comando y sus respectivas unidades.
Q = input('Dame el gasto [mc/seg] Q= ');
So = input('Dame el valor de la pendiente [m/m]So= ');
Ba = input('Dame el ancho de plantilla [metros] Ba= ');
Z1 = input('Dame el talud izq. [metros] Z1= ');
Z2 = input('Dame el talud der. [metros] Z2= ');
alfa = input('Dame el Coef. de Coriolis [adimensional] alfa= ');
Y1 = input('Valor del tirante inicial [metros] Y(1)= ');
Y2 = input('Valor del tirante final [metros] Y(n)= ');
n = input('Rugosidad del canal [adimensional] n= ');
NT = input('Número de Trapecios [adimensional] NT= ');
disp(' ') % Comando para generar textos de impresión.

% Declaración de constantes.
Z = Z1 + Z2; % Suma de taludes.
K = ((Z1 ^ 2) + 1) ^ 0.5 + ((Z2 ^ 2) + 1) ^ 0.5; % Calcula un dato para llamarlo
                                                % posteriormente.
ly = (Y2 - Y1) / NT; % Cálculo o valor para incrementar los tirantes.
for i = 1 : NT + 1 % Inicia la asignación de tirantes con incremento.
```

```

Y(i) = Y1; % Inicia el cálculo del Tirante.
T(i) = Ba + (Z * Y(i)); % Calcula el ancho superficial del canal.
Pm(i) = Ba + (K * Y(i)); % Calcula el perímetro hidráulico del canal.
Am(i) = (Ba * Y(i)) + (Z * (Y(i) ^ 2) / 2); % Calcula el área hidráulica.
Rh(i) = Am(i) / Pm(i); % Calcula el radio hidráulico.
Num(i) = 1 - ((alfa * (Q ^ 2) * T(i)) / (9.81 * (Am(i) ^ 3))); % Parte de la función.
Den(i) = So - (((n * Q) / Am(i)) ^ 2) * (1 / (Rh(i) ^ (4 / 3))); % Parte de la función.
Fy(i) = (Num(i) / Den(i)); % Valores de la función FGV a integrar.
Y1 = Y(i) + ly; % Nuevo tirante calculado.
end % Finaliza el estatuto for.
disp(' ') % Comando para generar textos de impresión.

% Procedimiento papara calcular el área bajo la curva de la función a
% integrar, con el ciclo for.
Longitud= 0;
for i= 1 : NT
    Fa(i) = (Fy(i) + Fy(i + 1)) * ly / 2; % Fórmula del trapecio.
    Longitud = Longitud + Fa(i); % Acumula la sumatoria de las áreas
                                % de la función a integrar, es decir, longitud del
                                % flujo gradualmente variado.
end % Fin del ciclo for.

% Imprime valores de: Tirantes con incrementos, áreas, perímetro mojado y radio
% hidráulico respecto a los tirantes, y valor de la función a integrar.
for i = 1 : NT + 1
    fprintf('Tirante (%2.0f)=%6.4f Área (%2.0f)=%6.4f Perímetro (%2.0f)=%6.4f
Radio (%2.0f)=%6.4f Función (%2.0f)=%6.4f\n',i,Y(i),i,Am(i),i,Pm(i),i,Rh(i),i,Fy(i))
end

% Imprime el cálculo de la longitud del flujo gradualmente variado.
fprintf(' =====\n')
fprintf(' La longitud del flujo gradualmente variado es L= %6.5f\n', Longitud)
fprintf(' =====\n')

```

4.- Resultados y observaciones.

En cada tema de estudio, al fin de ellas, se exponen los códigos de los diagramas de flujo que rigen a cada programa, describiendo detalladamente cada proceso del código.

Cuando afrontamos un proyecto de software, sea del tipo que sea, la primera consideración que se hace es la elección de herramienta de trabajo. En este caso un lenguaje de programación. MATLAB 6.5 O VISUAL BASIC 6.0:

Criterios principales:

- Conocer el lenguaje de programación.
- El tiempo total de desarrollo del proyecto.
- La calidad del resultado final.
- Relación esfuerzo-resultado

Ambos lenguajes de programación están ordenados o regidos por el mismo diagrama de flujo o falso algoritmo, la única discrepancia es, que en las estructuras de control se agregan u omiten algunos términos para ambos lenguajes ([Ver apéndice VIII](#)).

Dentro de los programas con los que se cuenta actualmente para trabajar en matemáticas aplicadas e ingeniería, está MATLAB (MATrix LABoratory), que ha venido incrementando su popularidad en los últimos años, debido a su gran versatilidad, pues cuenta con una gran cantidad de funciones que pueden ser usadas por los trabajadores de diversas áreas, debido a las funciones incorporadas y a su facilidad en el trabajo con vectores y matrices, puede requerir mucho menos tiempo de programación que los demás lenguajes.

Las variables en MATLAB no necesitan ningún tipo de declaración y pueden almacenar sucesivamente distintos tipos de datos: enteros, reales, escalares, matriciales, caracteres, etc. se crean simplemente, asignándoles un valor. En

VISUAL BASIC 6.0, es necesario la declaración y dimensionamiento de todas las variables que se involucran en el código.

Los códigos elaborados en este proyecto, ambos lenguajes de programación son eficaces (nos proporcionan los valores que ordenamos en el código), pero es más eficiente en MATLAB 6.5, siempre y cuando éste lo manejemos en variables vectoriales.

A pesar que en VISUAL BASIC 6.0, en estos programas elaborados, maneja controles insertados, estos no retardan la ejecución o el tiempo de aplicación del programa.

El lenguaje VISUAL BASIC 6.0 está orientado a la creación de objetos, y no a numéricos, por ende, es buena herramienta para el área de estudio.

No se manejó controles insertables en MATLAB 6.5, debido a la complejidad de este programa.

MATLAB 6.5, por ser código en “escalera” un error en tiempo de ejecución nos hace correr por completo nuevamente el programa.

En MATLAB 6.5, se tiene la ventaja de programar en formato “escalera” (.m) y formato “de controles insertables” (.fig); mediante el comando GUIDE.

En este apartado solo expresamos las características más convenientes respecto a nuestro tema de estudio, cabe aclarar que ambas tienen peculiaridades aún más sofisticadas. Base de datos, librerías, operaciones con ciertos hardware, interacción con otros software, etc., para el caso de VISUAL BASIC 6.0. Procesamiento de señales, simulación de circuitos, interacción con otros software, etc., para MATLAB 6.5.

VISUAL BASIC 6.0	MATLAB 6.5
VENTAJAS	VENTAJAS
Es un lenguaje simple y fácil de aprender.	Es un lenguaje simple y fácil de aprender.
Buena precisión	Alta precisión
Su mayor facilidad radica en el dibujado de formularios, mediante el arrastre de controles.	También radica en el dibujado de formularios, mediante el comando GUIde, para las versiones de 6.5 en adelante. (No esta muy completo, aún)
Es un lenguaje RAD (Desarrollo Rápido de Aplicaciones), centrado en conseguir en el menor tiempo posible los resultados deseados.	Se basa en el nuevo JIT-Accelerator, que opera sobre numerosos tipos de datos y de vectores, así como bloques de programación, para acelerar el rendimiento de las funciones.
Carece de funciones matemáticas.	Amplio soporte matemático.
La sintaxis está cercana al lenguaje humano.	La sintaxis de programación está cercana al lenguaje humano.
Magnífica ayuda (MSND library)	Magnífica ayuda
Tiene una ligera implementación de POO	No tiene una ligera implementación de POO
Gran parte del trabajo en el diseño de formularios está realizado, gracias a la gran gama de controles incorporados junto al lenguaje que ahorran costes de tiempo de desarrollo.	
Permite crear controles personalizados fácilmente del mismo modo que el diseño de formularios.	
Por defecto permite la programación sin declaración de variables, en algunos casos.	Por defecto permite la programación sin declaración de variables.
DESVENTAJAS	DESVENTAJAS
Sólo genera ejecutables para Windows.	Distribución de ejecutables (no lo integra la versión 6.5, hacia atrás).
NO permite características de programación avanzada.	Problemas eventuales de velocidad
La forma de programación que plantea VISUAL BASIC ha ocasionado que muchos programadores de Visual Basic practiquen malas costumbres.	La forma de programación que plantea MATLAB ha ocasionado que muchos programadores de MATLAB practiquen malas costumbres.

5.- Conclusiones.

Respecto a programación, con enfoque en la hidráulica, es más fácil programar en MATLAB 6.5, debido a que, en este lenguaje, no se consideró necesariamente controles insertables para su codificación (MsFlexGrid1, etc., en el caso de VISUAL BASIC 6.0). También considero la facilidad en MATLAB 6.5 por su legibilidad en la sintaxis, que en VISUAL BASIC 6.0 son imprescindibles.

A pesar que MATLAB es un lenguaje de alto nivel, lo considero como principio a MATLAB 6.5, por su manejo tan práctico, además de hacernos más eficientes, y por lo que nos da como consecuencia familiarizarnos de manera rápida al entorno de la programación, debido a la simplicidad del lenguaje (sintaxis).

Dejo a VISUAL BASIC 6.0 en segundo rubro, debido a que su aprendizaje implica más manejo de conceptos (manejo de objetos, definición de propiedades), declaraciones de variables, etcétera.

5.- Bibliografías.

French, H, R. 1988. Hidráulica de canales abiertos. Primera edición. Editorial McGraw-Hill. Mexico. 703 p.p.

Garza, V. S. Z. 2009. APUNTES DE HIDRAULICA DE CANALES. UAAAN. Depto. de Riego y Drenaje. Saltillo, Coahuila. México.

Garza, V. S. Z. 2009. APUNTES DE HIDRAULICA II. UAAAN. Depto. de Riego y Drenaje. Saltillo, Coahuila. México.

Giles, V. R. 1969. MECANICA de los FLUIDOS e HIDRAULICA. Segunda edición. Editorial McGraw-Hill. México. 267 p.p.

Hernández, M. F. 1987. Programación y métodos numéricos utilizados en la solución de problemas de canales con flujo uniforme y gradualmente variado en estado permanente. Tesis de Licenciatura. UAAAN. Depto. de Riego y Drenaje. Saltillo, Coahuila. México.

<http://elticus.com/?contenido=19>

<http://es.wikipedia.org/wiki/algoritmo>

http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n

<http://es.wikipedia.org/wiki/programacion>

http://es.wikipedia.org/wiki/Visual_Basic

<http://lenguajes-de-programacion.com/herramientas-de-programacion.shtml>

<http://lenguajes-de-programacion.com/programacion.shtml>

<http://www.di.uniovi.es/~dani/asignaturas/transparencias-leccion20.pdf>

<http://www.fing.uach.mx/licenciatura/civil/MANUALES%20DE%20PRACTICAS/MANUAL%20DE%20PRACTICAS%20DE%20HIDRAULICA%20II.pdf>

<http://www.itrc.org/riego/capitulo5.pdf>

<http://www.monografias.com/trabajos5/matlab/matlab.shtml>

<http://www.monografias.com/trabajos13/tumatlab/tumatlab.shtml?monosearch>

<http://www.monografias.com/trabajos26/lenguajes-programacion/lenguajes-programacion.shtml>

<http://www.monografias.com/trabajos62/visual-basic-prototipo/visual-basic-prototipo.shtml>

<http://www.taringa.net/posts/downloads/1410159/Matlab,-en-100mb.html>

http://www.ucm.es/info/Geofis/practicas_ICNYP/modulos_1_7_2005-06_grupo_D.pdf

Salvador, I. R. 1997. Análisis hidráulico de un lateral de riego por goteo. Tesis de maestría. UAAAN. Depto. de Riego y Drenaje. Saltillo, Coahuila. México.

6.- Apéndices.

Apéndice I

Método de aproximaciones sucesivas

El método de las aproximaciones sucesivas es uno de los procedimientos más importantes y más sencillos de codificar. Considérese una función cualquiera " $f(x)$ " (función continua que se desea determinar sus raíces reales) de manera que cuando " x " toma el valor de " x_n " la función es igual a cero $f(x_n) = 0$.

$$f(x) = 0 \dots\dots\dots \text{Ecuación 1}$$

Si sumamos " x " en ambos lados de la *Ecuación 1* se tiene:

$$f(x) + x = x \dots\dots\dots \text{Ecuación 2}$$

Se sustituye $f(x) + x$ por la ecuación equivalente:

$$g(x) = x \dots\dots\dots \text{Ecuación 3}$$

Intercambiando de miembros de ambas ecuaciones:

$$x = g(x) \dots\dots\dots \text{Ecuación 4}$$

Como primer intento de aproximación de una raíz real de la función continua se supone un valor de $x = x_1$, se sustituye en *Ecuación 4*, y se tiene x_2 , es decir:

$$x_2 = g(x_1)$$

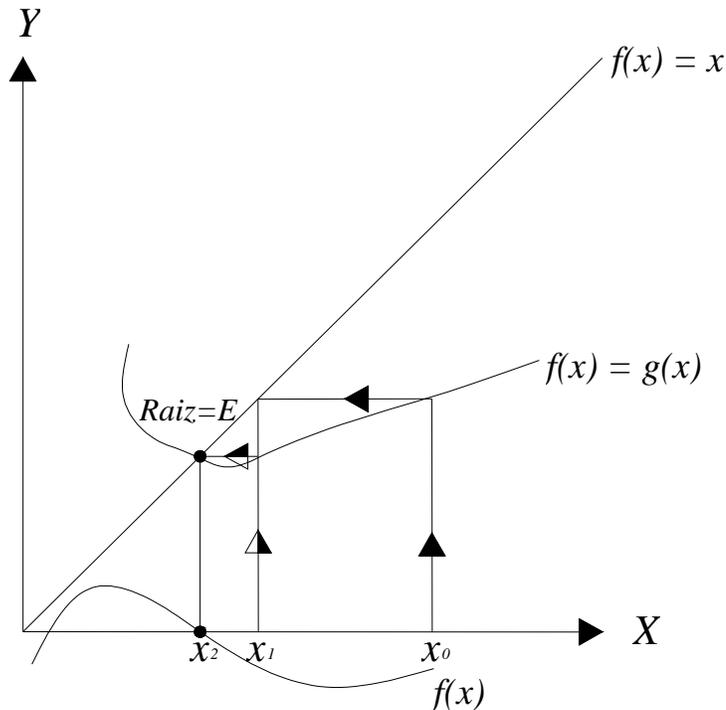
Si el valor de x_2 no es aproximadamente igual a x_1 , entonces el nuevo valor que se sustituye en la *Ecuación 4* es x_2 y así sucesivamente. Este proceso se puede sintetizar de la siguiente forma:

$$x_n = g(x_{n-1}) \quad n = 1, 2, \dots \dots\dots \text{Ecuación 5}$$

Si esta secuencia es convergente es decir, tiende hacia un límite, la solución E es:

$$E = \lim_{n \rightarrow \infty} x_n$$

Explicación geométrica mediante el gráfico.



Condición finalizada.

Cuando el valor absoluto del cociente entre la diferencia de dos términos consecutivos de la sucesión y uno de los términos, sea menor que cierta cantidad E .

$$\left| \frac{x_{n+1} - x_n}{x_{n+1}} \right| \leq E$$

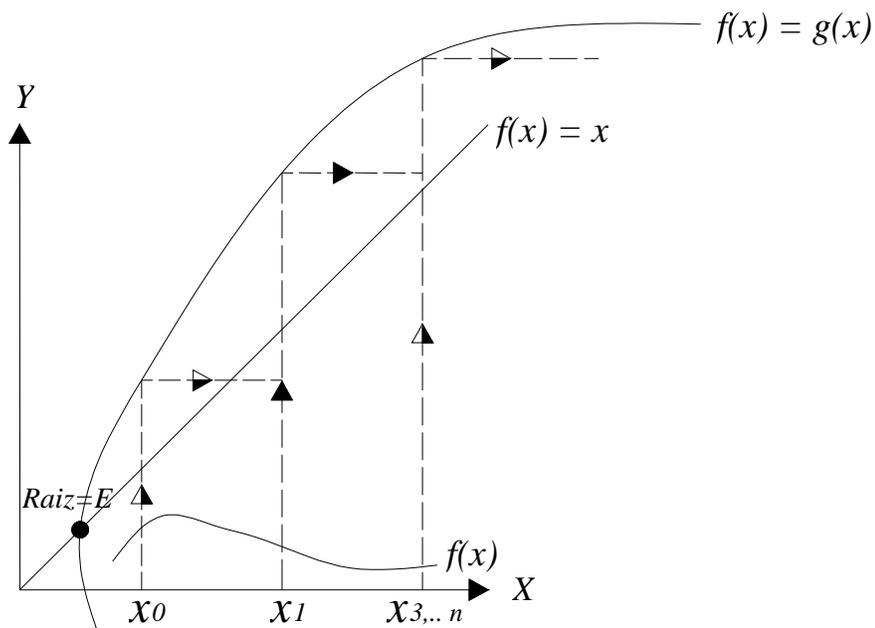
Criterio de convergencia

No todas las funciones se pueden resolver por este método, solamente si el valor absoluto de la derivada de la función $g(x)$ en la vecindad de la raíz E es menor o igual que la unidad (la pendiente de la recta bisectriz del primer cuadrante es uno).

$$|g'(x)| < 1 \quad \text{El método } \mathbf{Converge} \text{ linealmente.}$$

$|g'(x)| \geq 1$ El método **Diverge**

En la figura, podemos ver como es imposible encontrar la solución marcada por un punto negro en la intersección entre la curva y la recta bisectriz del primer cuadrante, ya que la sucesión x_i diverge.



Apéndice II

Método del trapecio

Los procedimientos de integración aproximada se emplean cuando la integración ordinaria sea muy complicada, cuando una integral indefinida no se pueda expresar mediante funciones elementales, o bien, cuando el integrando $f(x)$ venga definido por una tabla de valores.

Deducción del método del trapecio

Sea el área limitada por la curva $y = f(x)$, el eje X y las ordenadas en los extremos $x = a$ y $x = b$. Dividamos dicha área en n franjas verticales de anchura $h = (b - a) / n$ y consideremos la franja i limitada por el arco $P_{i-1}P_i$ de $y = f(x)$. Un valor aproximado del área de esta franja es

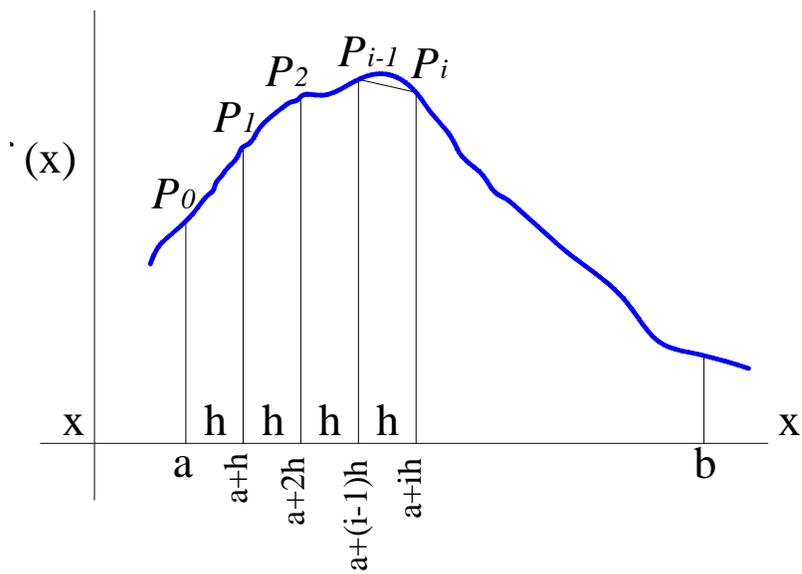
$$\frac{1}{2}h\{ f[a + (i-1)h] + f(a + ih) \}$$

que es el área del trapecio que resulta al sustituir el arco $P_{i-1}P_i$ por el segmento rectilíneo $P_{i-1}P_i$. Al efectuar esta sustitución en todas las franjas se obtiene:

$$\int_a^b f(x)dx \approx \frac{h}{2}\{ f(a) + f(a+h) \} + \frac{h}{2}\{ f(a+h) + f(a+2h) \} + \dots + \frac{h}{2}\{ f[a + (n-1)h] + f(b) \}$$

O sea

$$\int_a^b f(x) dx = \frac{h}{2} \{ f(a) + 2[f(a+h) + f(a+2h) + \dots + f(a+[n-1]h)] + f(b) \}$$



Apéndice III

Salidas múltiples laterales

El tema en mangueras laterales ha sido discutido con profundidad en numerosas publicaciones desde los primeros años de la década de 1970. Existen muchas fórmulas, programas de computadora, tablas y ayudas gráficas disponibles para solucionar la hidráulica de las manguera laterales. Los autores recomiendan el uso de pequeños programas de computadora.

Las diferentes alternativas de solución de este tipo de problema difieren aproximadamente hasta un 10 %. Esto significa que no existe ningún método para la solución de los problemas típicos de la hidráulica de las laterales.

Aunque la fricción real depende también de unos cuantos factores, estos factores incluyen la variación de la temperatura del agua a lo largo de la tubería, el tamaño y la forma de las barbillas de emisores o de las protuberancias de las mangueras y la variación exacta en el espesor de pared de la manguera.

Uno de los procedimientos más utilizados es utilizando la ecuación de Hazen-Williams, en la cual se presenta a continuación. Esta *Ecuación 1* es para una tubería sin salidas, en la cual solo hay un diámetro y en ella todo el flujo que entra o fluye a lo largo de toda la longitud,

$$h_f = \frac{1.21 \times 10^{10}}{D^{4.87}} * \left(\frac{Q}{C} \right)^{1.852} * L$$

Ecuación 1

Donde:

D = Diámetro interno del tubo, mm.

Q = Gasto a través del tubo, lps.

L = Longitud total del tubo o la manguera, m.

C = Coeficiente de "rugosidad" de Hazen-Williams (Ver la siguiente Tabla No. 1)

Tabla No. 1. Coeficiente de rugosidad del de Hazen-Williams (C)

Material	Valores "C" de Hazen-Williams
Concreto	130
Asbesto cemento	140
Aluminio con acoplamiento cada 30'	120
Hierro nuevo	130
Hierro viejo (15 años)	100
PVC	
Diámetro interior(ID) >127 mm (5 ")	150
38.1 mm < ID <= 127 mm	146
ID < 38.1 mm1	143
Manguera de goteo (polietileno) con emisores. Valor promedio equivalente para toda la longitud de la manguera.	
ID = 12.7 mm	132
ID = 17.8 mm	139
ID = 22.9 mm	145
ID = 27.9 mm	148

Para el caso de una tubería de un solo diámetro (o manguera), con varias salidas uniformemente espaciadas a través de las cuales se descarga toda el agua, la fricción total puede estimarse como,

$$hf \text{ de una lateral} = hf * F$$

Donde:

hf = La pérdida total calculada, m. Ecuación 1

F = Factor para salida múltiple, (ver tabla No. 2)

hf de una lateral = Es la pérdida de carga total de la longitud del sistema, m.

Tabla No. 2 Factor de salida múltiple "F"

Número de salidas	F
1	1
6	0.44
10	0.40
20	0.38
40	0.37
100	0.36
Mayores de 100	0.351

Apéndice IV

Flujo uniforme - permanente

El flujo uniforme y permanente comprende dos condiciones de flujo. El flujo permanente, como se define para flujo en tuberías, se refiere a la condición según la cual las características del flujo en un punto no varían con el tiempo ($\partial V / \partial t = 0$, $\partial y / \partial t = 0$, etc.). El flujo uniforme se refiere a la condición según la cual la profundidad, pendiente, velocidad y sección recta permanecen constantes en una longitud dada del canal ($\partial y / \partial L = 0$, $\partial V / \partial L = 0$).

El flujo uniforme-permanente se observa solo en los canales prismáticos (simétricos o asimétricos) con pendientes muy pequeñas y muy largas y lejos de sus extremos, donde el tirante hidráulico y la velocidad media del canal, y otras características hidráulicas, permanecen constantes en toda la sección a lo largo del canal.

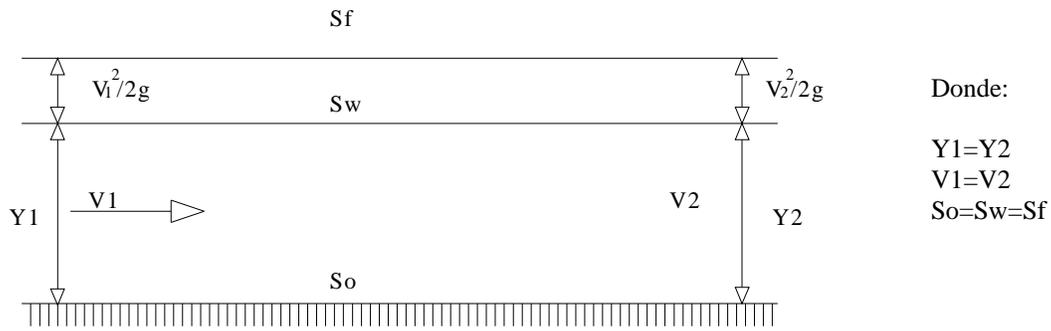


Figura 1. Perfil longitudinal de un canal, mostrando flujo uniforme.

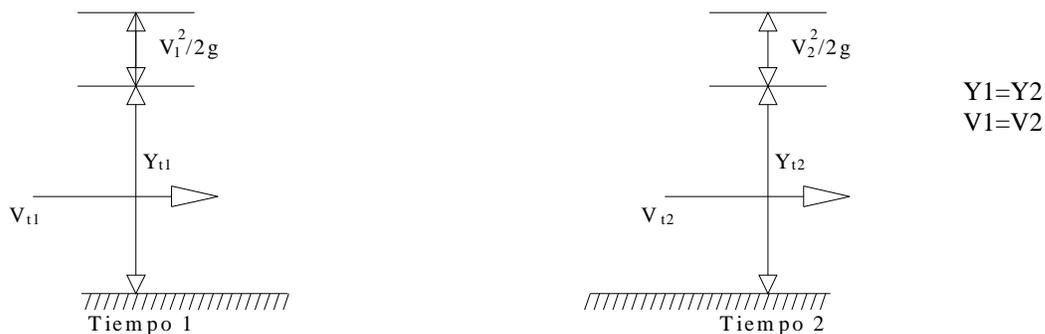
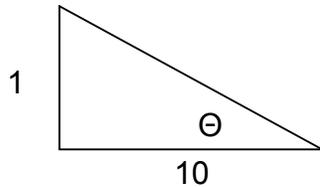


Figura 2. Perfil longitudinal de un canal, mostrando flujo permanente.

Resumiendo, las condiciones del flujo uniforme son.

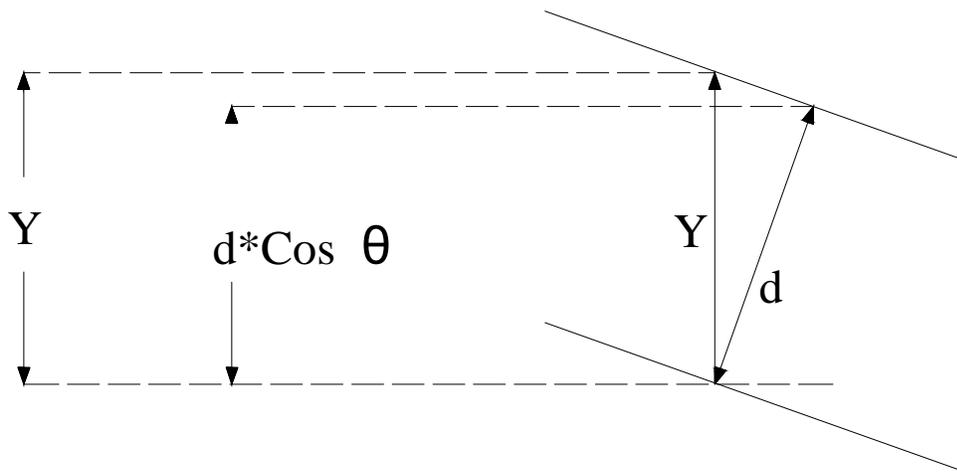
- 1.- Flujo permanente.
- 2.- Pendiente del canal no muy grande. (1:10).



$$\theta = \text{ArcTan} (1/10) = \text{ArcTan} 0.1 = 5.715$$

$$\text{Cos } \theta = \text{Cos } 5.71^\circ = 0.995 \approx 1.0$$

a).- El tirante hidráulico es prácticamente el mismo si se mide en forma vertical o si se mide normal al piso del canal.



b).- $\text{Cos } \theta \approx 1$

- 3.- El canal es prismático. La sección transversal es constante.
- 4.- El coeficiente de rugosidad es constante a lo largo del canal.

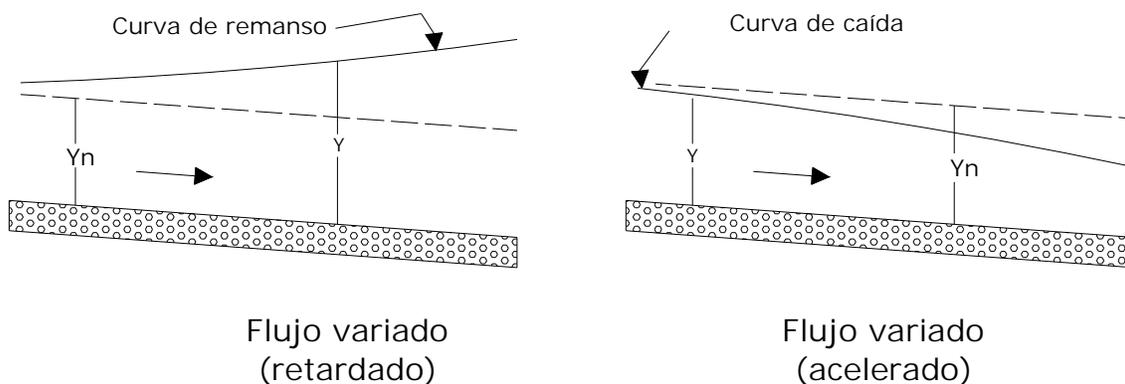
Apéndice V

Flujo gradualmente variado

El flujo variado puede ser clasificado como rápidamente variado o gradualmente variado. En el flujo rápidamente variado (FRV) la profundidad de flujo cambia abruptamente en una distancia comparativamente corta, por ejemplo en un resalto hidráulico. En el flujo gradualmente variado (FGV), se requieren distancias mayores para que alcancen a desarrollarse los perfiles de flujo gradualmente variado. En un canal con flujo permanente uniforme pueden existir causas que retardan o aceleran la corriente de forma que pasa a condiciones variadas que se manifiestan por un aumento o disminución de la profundidad del flujo, respectivamente.

Flujo variado retardado. Se presenta cuando la velocidad del flujo disminuye, y por ende aumenta la profundidad, en el sentido de la corriente. Algunas causas que retardan el flujo son: disminución brusca de la pendiente del canal; interposición de obstáculos en el lecho del canal como vertederos, presas, compuertas de control. Para condiciones iniciales de flujo uniforme lento, se tendrá flujo gradualmente variado; para flujo uniforme rápido se presentará un resalto hidráulico al pasar a condiciones de remanso.

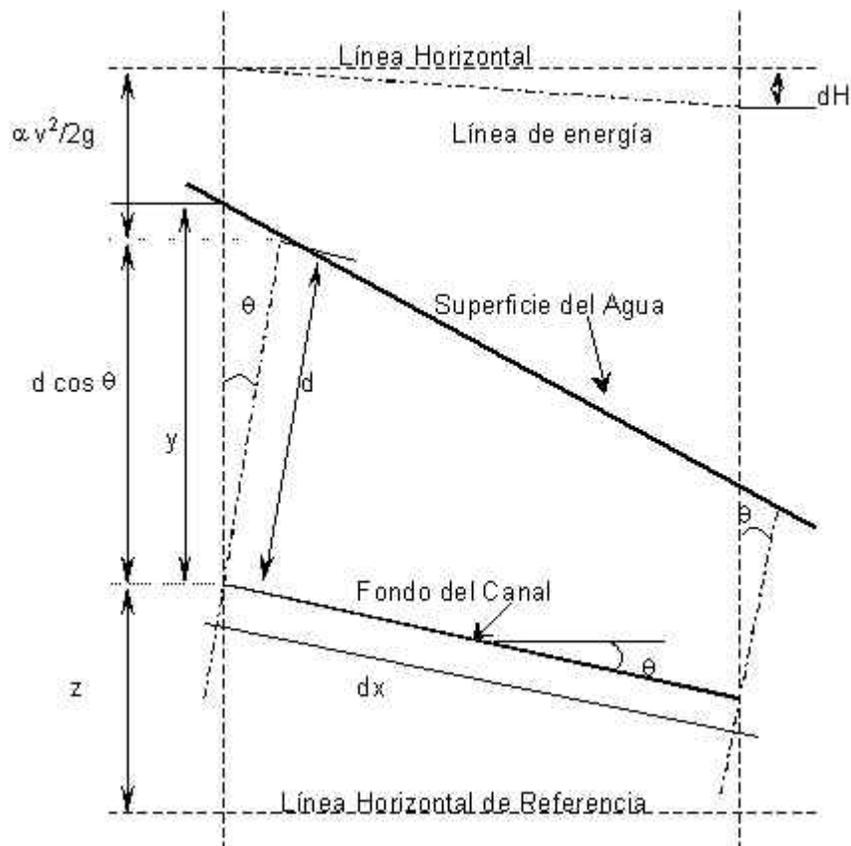
Flujo variado acelerado. Se presenta cuando la velocidad del flujo aumenta, y por ende la profundidad disminuye, en sentido de la corriente; ocurre cuando la pendiente del canal aumenta bruscamente o cuando existe una caída vertical.



El flujo gradualmente variado, es un flujo permanente cuya profundidad varía de manera gradual a lo largo del canal. Se tendrán en cuenta las siguientes hipótesis:

1. La pérdida de altura en una sección es igual que la de un flujo uniforme con las mismas características de velocidad y radio hidráulico.
2. La pendiente del canal es pequeña ($<10\%$). Esto quiere decir que la profundidad del flujo puede medirse verticalmente o perpendicularmente al fondo del canal y no se requiere hacer corrección por presión ni por arrastre del aire.
3. El canal es prismático.
4. Los coeficientes de distribución de la velocidad y el de rugosidad son constantes en el tramo considerado.

Ecuación dinámica del flujo gradualmente variado



En cualquier sección transversal la energía total H está dada por la expresión:

$$H = \alpha \frac{v^2}{2g} + d \cos \theta + Z \quad \dots \text{Ecuación 1}$$

Donde "H", "Z", "d" y "θ" son según se muestran en la figura, "α" es el coeficiente de energía y "v" es la velocidad media del flujo a través de la sección.

Se asume que "θ" y "α" son constantes en el tramo del canal.

Tomando el piso del canal como el eje "x" y derivando la ecuación (1) con respecto a x se obtiene,

$$\frac{dH}{dx} = \alpha \frac{d}{dx} \left(\frac{v^2}{2g} \right) + \cos \theta \frac{dd}{dx} + \frac{dZ}{dx} \quad \dots \text{Ecuación 2}$$

Si S_f es la pendiente de la línea de energía $S_f = -\frac{dH}{dx}$, S_o la pendiente del piso del canal $S_o = -\frac{dZ}{dx}$. *sustituyendo en la ecuación 2*

$$-S_f + S_o = \alpha \frac{d}{dx} \left(\frac{v^2}{2g} \right) + \cos \theta \frac{dd}{dx} \quad \dots \text{multiplicando por } \frac{dx}{dd}$$

$$(-S_f + S_o) \frac{dx}{dd} = \left(\alpha \frac{d}{dd} \left(\frac{v^2}{2g} \right) + \cos \theta \right)$$

ordenando y resolviendo para $\frac{dx}{dd}$ se tiene:

$$\frac{dx}{dd} = \frac{\cos \theta + \alpha \frac{d}{dx} \left(\frac{v^2}{2g} \right)}{S_o - S_f}$$

$\cos \theta = 1$ Para $\theta < 10^\circ$ flujo uniforme

$$\frac{dx}{dd} = \frac{1 + \alpha \frac{d}{dx} \left(\frac{v^2}{2g} \right)}{S_o - S_f} \quad \dots\dots \text{Ecuación 3}$$

como el flujo es uniforme $S_o = S_f$ solo si Δx es pequeño

$$Q = \frac{Am}{n} Rh^{2/3} S_o^{1/2} \quad \dots\dots \text{sustituyendo } (S_o \text{ por } S_f) \text{ y despejando } S_f \text{ y reemplazando en la ecuación 3, se tiene :}$$

$$\frac{dx}{dd} = \frac{1 + \alpha \frac{d}{dx} \left(\frac{v^2}{2g} \right)}{S_o - \left(\frac{nQ}{Am} \right)^2 \frac{1}{Rh^{4/3}}}$$

$$\alpha \frac{d}{dx} \left(\frac{v^2}{2g} \right) = - \sqrt{\frac{\alpha Q^2 T}{g A^3}} \quad \text{entonces :}$$

$$\frac{dx}{dd} = \frac{1 - \sqrt{\frac{\alpha Q^2 T}{g A^3}}}{S_o - \left(\frac{nQ}{Am} \right)^2 \frac{1}{Rh^{4/3}}}$$

$$x = \int_{d^2}^{d^1} \frac{1 - \sqrt{\frac{\alpha Q^2 T}{g A^3}}}{S_o - \left(\frac{nQ}{Am} \right)^2 \frac{1}{Rh^{4/3}}} dd \quad \text{sustituyendo } d \text{ por } Y(\text{tirante})$$

$$x = \int_{Y_1}^{Y_2} \frac{1 - \sqrt{\frac{\alpha Q^2 T}{g A^3}}}{S_o - \left(\frac{nQ}{Am} \right)^2 \frac{1}{Rh^{4/3}}} dY \quad \text{Ecuación dinamica del flujo gradualmente variado}$$

Apéndice VI

Resultados: Salidas múltiples laterales

Tabla 1 Valores obtenidos en el programa del ejemplo 1

Salidas	Gasto en los orificios	Carga hidráulica en los orificios	Gasto en cada tramo de la tubería	Pérdida de carga por tramo en la tubería
	litros/segundo	mca*	litros/segundo	metro
1	5.8834	0.8764	27.6595	0.1235
2	5.6108	0.7971	21.7761	0.0793
3	5.4476	0.7514	16.1653	0.0456
4	5.3697	0.7300	10.7176	0.0213
5	5.3480	0.7241	5.3479	0.0058

El gasto en la entrada del sistema es de $Q = 27.6595$ l/s

Tabla 2 Valores obtenidos en el programa del ejemplo 2

Salidas	Gasto en los orificios	Carga hidráulica en los orificios	Gasto en cada tramo de la tubería	Pérdida de carga por tramo en la tubería
	litros/segundo	ma*	litros/segundo	metro
1	1.5295	0.9477	20.7422	0.057807
2	1.4925	0.9024	19.2127	0.050373
3	1.4598	0.8634	17.7202	0.043530
4	1.4314	0.8301	16.2603	0.037246
5	1.4070	0.8021	14.8288	0.031494
6	1.3864	0.7787	13.4217	0.026249
7	1.3694	0.7597	12.0353	0.021495
8	1.3556	0.7444	10.6659	0.017216
9	1.3447	0.7326	9.3103	0.013403
10	1.3366	0.7237	7.9655	0.010051
11	1.3307	0.7174	6.6289	0.007158
12	1.3268	0.7132	5.2981	0.004729
13	1.3246	0.7108	3.9712	0.002774
14	1.3235	0.7096	2.6466	0.001308
15	1.3232	0.7093	1.3231	0.000362

El gasto en la entrada del sistema es de $Q = 20.7423$ l/s

Tabla 3 Valores obtenidos en el programa del ejemplo 3

Salidas	Gasto en los orificios	Carga hidráulica en los orificios	Gasto en cada tramo de la tubería	Pérdida de carga por tramo en la tubería
	litros/segundo	mca*	litros/segundo	metro
1	1.06865	7.4026	3.0077	2.5973
2	0.98198	6.2506	1.9391	1.1520
3	0.95720	5.9390	0.9571	0.3115

El gasto en la entrada del sistema es de $Q = 3.00785$ l/s

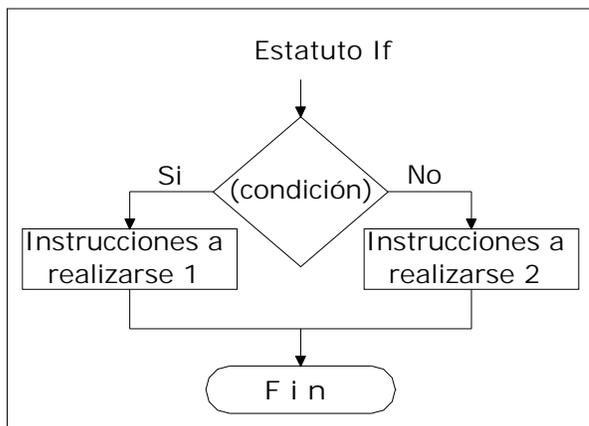
*mca = metros de columna de agua.

Apéndice VII

Concepto de estructuras de control

1.- Estructura o estatuto if.

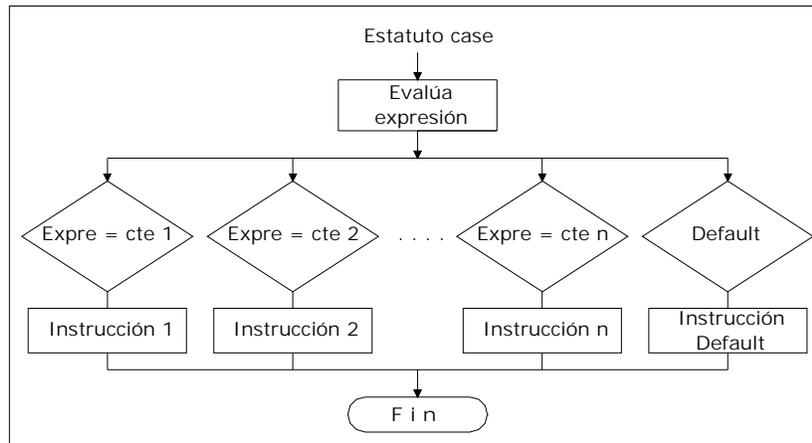
El estatuto if o estatuto si es una estructura selectiva que sirve para seguir una o varias líneas de ejecución cuando se cumple una determinada condición, sea ésta, verdadera o falsa.



2.- Estructura o estatuto case.

El estatuto case o estatuto acaso, se usa en decisiones múltiples, que prueba si una expresión coincide con uno de un número de valores constantes enteros y traslada el control adecuadamente.

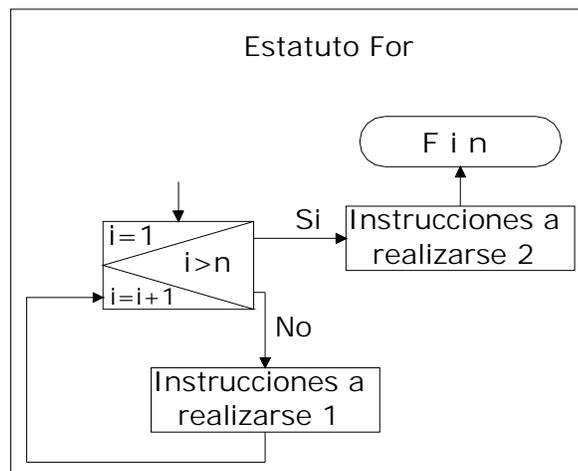
Cada case se etiqueta con uno o mas valores constantes enteros o expresiones constantes enteras. Si un case coincide con el valor de la expresión, la ejecución comienza en éste. Todas las expresiones case deben ser diferentes. El etiquetado como default se ejecuta si ninguno de los otros se satisface (es optativo).



3.- Estructura o estatuto for.

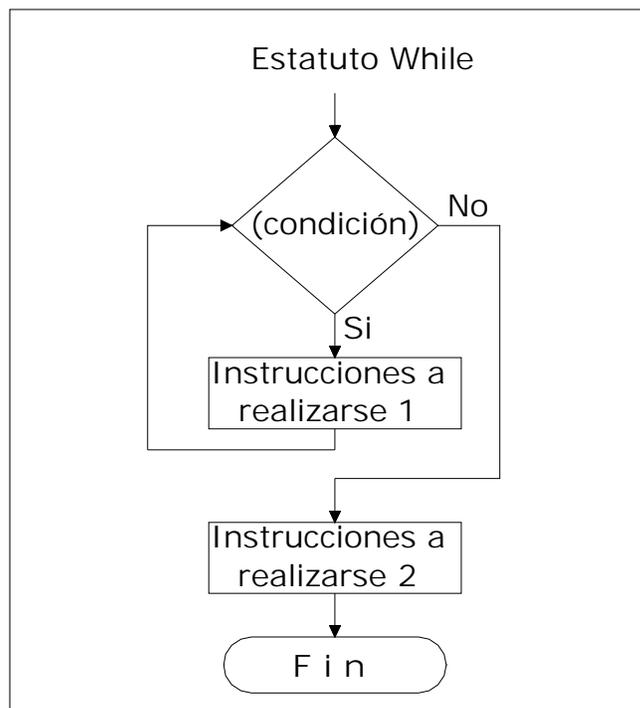
El bucle for o bucle para es una estructura de control en la que se puede indicar el número máximo de iteraciones.

Su uso principal se orienta a los vectores, pudiendo modificar, agregar, eliminar o consultar datos que se encuentren según el índice. Por esto último, una condición mínima del vector es que debe ser ordenado, por que si se intenta leer un dato inexistente, esto genera un error de programación.



4.- Estructura o estatuto while.

El Bucle while o bucle mientras es una estructura de la mayoría de los lenguajes de programación estructurados cuyo propósito es repetir un bloque de código mientras una condición se mantenga verdadera.



Apéndice VIII

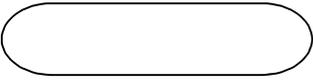
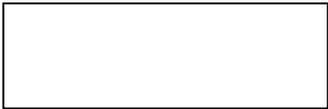
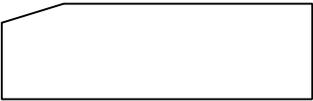
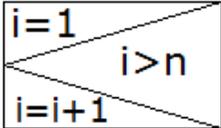
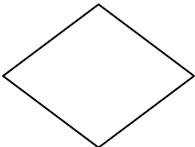
Estructuras de control

MATLAB	VISUAL BASIC
<code>if</code> (condición) Sentencias verdaderas <code>else</code> Sentencias falsas <code>end</code>	<code>If</code> (condición) <code>Then</code> Sentencias verdaderas <code>Else</code> Sentencias falsas <code>End If</code>
<code>switch</code> (variable_de_entrada) <code>case</code> 1 <code>case</code> 2 . <code>case</code> n <code>otherwise</code> otras instrucciones <code>end</code>	<code>Select Case</code> (Variable_de_entrada) <code>Case</code> 1 <code>Case</code> 2 . <code>Case</code> n <code>Case else</code> Otras instrucciones <code>End Select</code>
<code>for</code> i=Val_inicial : Incrto* : Val_final Sentencias a ejecutar <code>end</code>	<code>For</code> i=Val_inicial <code>To</code> Val_final <code>Step</code> Incrto*. Sentencia a ejecutar <code>Next</code> i
<code>while</code> (condición) Sentencias a ejecutar <code>End</code>	<code>Do While</code> (condición) Sentencias a ejecutar <code>Loop</code>

*Incrto=Incremento

Apéndice IX

Descripción de símbolos utilizados en los diagramas de flujo

Símbolo	Función o descripción
	Figura de inicio o final de un programa.
	Figura de dirección de flujo.
	Figura para proceso de datos.
	Figura de captura de datos.
	Figura de cálculo secuencial.
	Figura de decisión de control.
	Figura de impresión.

Apéndice X

Variables utilizadas en el diagrama de flujo salidas múltiples laterales.

H_o = carga de entrada al sistema.

n = número de salidas laterales en una tubería.

L_o = espaciamiento entre salidas.

D_p = diámetro de la tubería que contiene las salidas (principal).

D_s = diámetro de descarga de las salidas.

C_d = coeficiente de descarga en las salida.

C = coeficiente de rugosidad del material.

contador= cuenta de los cálculos hecho por el programa.

Q_s = gasto que se asume a la entrada al sistema.

A_o = área del orificio de la salida o descarga.

$h_f(i)$ = pérdida de energía en cada tramo o sección, $i = 1, 2, 3, \dots, n$

$Q(i)$ = gastos en cada sección o tramo de la tubería, $i = 1, 2, 3, \dots, n$

$H(i)$ = carga sobre la salida de los orificios.

$q(i)$ = gasto en la salida de los orificios.

Q_o = gasto acumulado de los gasto en la salida de los orificios.

E = error de cálculo.

suma = sumatoria de gastos de las salidas de los orificios para asignárselos a Q_o .

m = valor de H_o .

si = inicia el primer cálculo del programa.

Apéndice XI

Variables utilizadas en el diagrama de flujo flujo uniforme permanente.

Q = gasto de abastecimiento en el canal.

n = coeficiente de rugosidad del canal.

So = pendiente longitudinal del canal.

Z1 = pendiente del talud izquierdo.

Z2 = pendiente del talud derecho.

Ba = ancho de plantilla del canal.

Ys = tirante normal propuesto.

U = constante funcional.

K = constante funcional.

Z = suma de taludes.

Pm = perímetro hidráulico de la sección transversal del flujo del canal.

Am = área hidráulica de la sección transversal del flujo del canal.

T = ancho superficial del espejo del agua.

Rh = radio hidráulico de la sección transversal del flujo del canal.

Fr = número de Froude.

Yo = tirante normal obtenido en el cálculo.

E = error en términos absolutos.

contador = iteraciones realizadas en el cálculo.

si = inicia el primer cálculo del programa.

Apéndice XII

Variables utilizadas en los diagramas de flujo flujo gradualmente variado

Y1 = tirante inicial.

Y2 = tirante final.

Q = gasto de abastecimiento en el canal.

So = pendiente longitudinal del canal.

Ba = ancho de plantilla del canal.

Z1 = talud izquierdo del canal.

Z2 = talud derecho del canal.

alfa = coeficiente de Coriolis.

n = coeficiente de rugosidad del canal.

NT = número total de trapecios.

Z = asignación de ambos taludes.

K = constante de funcionalidad.

ly = incremento para cada tirante.

Y(i) = tirantes, para $i = 1, 2, 3, \dots, n+1$.

T(i) = ancho superficial del espejo del agua, para $i = 1, 2, 3, \dots, n+1$.

Pm(i) = perímetro hidráulico de la sección transversal del flujo del canal, para
 $i = 1, 2, 3, \dots, n+1$.

Am(i) = área hidráulica de la sección transversal del flujo del canal, para
 $i = 1, 2, 3, \dots, n+1$.

Rh(i) = radio hidráulico de la sección transversal del flujo del canal, para
 $i = 1, 2, 3, \dots, n+1$.

Fy(i) = Ecuación dinámica del flujo gradualmente variado, función a integrar, para
 $i = 1, 2, 3, \dots, n+1$.

Fa(i) = Integración de la ecuación del flujo gradualmente variado por el método del trapecio, para obtener el área, $i = 1, 2, 3, \dots, n$.

Longitud = área acumulada o longitud del flujo gradualmente variado.

Apéndice XIII

Descripción de las variables utilizadas en los códigos

Salidas múltiples laterales

Descripción	Unidad
Lo = espacio entre las salidas.	metros
Dp = diámetro de la tubería regante.	pulgadas
Ho = presión a la entrada de la tubería con salidas múltiples.	metros
C = coeficiente de rugosidad.	adimensional
Ds = diámetro de las salidas.	pulgadas
Cd = coeficiente de descarga en las salidas.	adimensional
Qs = gasto propuesto en el tramo 1.	litros/segundo
n = número de salidas de descarga.	adimensional
contador = cuenta los cálculos que se ejecutan.	-
suma = suma todos los gastos de las salidas.	litros/segundo
Q(i) = gasto en el tramo, i=1, 2, 3, ... n	litros/segundo
H(i) = carga en el orificio, i=1, 2, 3, ... n	metros
hf(i) = pérdida de carga en el tramo i=1, 2, 3, ... n	metros
Ao = área del orificio	metros ²
q(i) = gasto en la salida del orificio, i=1, 2, ... n	litros/segundo
E = error	litros/segundo

Flujo uniforme permanente

Descripción	Unidad
Q = gasto de abastecimiento del canal.	litros/segundo
n = coeficiente de rugosidad del canal.	adimensional
So = pendiente longitudinal del canal	metros/metros
Z1 = talud izquierdo del canal.	adimensional
Z2 = talud derecho del canal.	adimensional

Ba = base o ancho de plantilla del canal.	metros
Ys = tirante propuesto en el canal.	metros
contador = identifica el número de cálculos.	-
U = constante de función.	-
K = constante de función.	-
Z = suma de los dos taludes del canal.	adimensional
Pm = perímetro hidráulico transversal del canal.	metros
Am = área hidráulica transversal del canal.	metros ²
Rh = radio hidráulico transversal del canal.	metros
Yo = tirante obtenido del canal.	metros
E = error.	metros
Fr = número de Froude.	adimensional

Flujo gradualmente variado

Descripción	Unidad
Y(i) = tirante del flujo a lo largo del canal, i=1, 2, ... , n+1	metros
T(i) = ancho del espejo a lo largo del canal, i= 1, 2, ..., n+1	metros
Pm(i) = perímetro hidráulico a lo largo del canal, i= 1, 2, ... n+1	metros
Am(i) = área hidráulica a lo largo del canal, i=1, 2, ... n+1	metros ²
Rh(i) = radio hidráulico a lo largo del canal, i= 1, 2, ... n+1	metros
Fy(i) = Ecuación flujo gradualmente variado, i= 1, 2, ... n+1	metros
Fa(i) = áreas parciales para encontrar la longitud del FGV.	metros ²
Longitud = longitud del flujo gradualmente variado.	metros