



Instituto Tecnológico de Tuxtla Gutiérrez

Ingeniería Mecánica

REPORTE DE RESIDENCIA PROFESIONAL

Proyecto:

Desarrollo de un programa en MATLAB para analizar problemas de la asignatura de máquinas de fluidos incompresibles.

Asesor interno:

Ing. Hernán Valencia Sánchez

Revisor 1

M.C. Ignacio Arriola Cárdenas

Revisor 2

Ing. Víctor Manuel Vázquez Ramírez

Presenta:

De Los Santos Lara Pedro Jorge

Índice

Introducción	1
CAPÍTULO I. GENERALIDADES DEL PROYECTO	2
1.1 Justificación	3
1.2 Objetivos	3
1.2.1 Objetivo general	3
1.2.2 Objetivos específicos	3
1.3 Caracterización del área	3
1.3.1 Misión	3
1.3.2 Visión	4
1.3.3 Valores	4
1.4 Problema a resolver	4
1.5 Alcances	4
1.6 Limitaciones	4
CAPÍTULO II. FUNDAMENTO TEÓRICO	6
2. Fundamento teórico	7
2.1 Pérdidas en sistemas de tuberías	7
2.1.1 Pérdidas primarias	7
2.1.2 Pérdidas secundarias	10
2.2 Máquinas hidráulicas	11
2.2.1 Generalidades de las máquinas hidráulicas	11
2.2.2 Bombas centrífugas	18
2.2.3 Turbinas de acción	28
2.2.4 Turbinas de reacción	31
2.3 Desarrollo de interfaces gráficas de usuario en MATLAB	35
2.3.1 Introducción a MATLAB	35
2.3.2 Scripts y funciones	35
2.3.3 Trazo de gráficas bidimensionales	36
2.3.4 Interfaces gráficas utilizando GUIDE	37
2.3.5 Interfaces gráficas utilizando código (Programmatic GUI)	38
CAPÍTULO III. METODOLOGÍA	41

3. Procedimiento y descripción de las actividades realizadas	42
3.1 Consulta de las fuentes de información	42
3.2 Desarrollo de las funciones e interfaces gráficas en MATLAB	42
3.3 Depuración del programa	43
3.4 Elaboración de herramientas adicionales	43
CAPÍTULO IV. RESULTADOS	44
4. Resultados	45
4.1 Interfaz de inicio	45
4.2 Interfaz de selección de datos disponibles	46
4.3 Interfaz de cálculos	47
4.4 Interfaz de resultados	49
4.5 Interfaz de triángulos de velocidades	49
4.6 Interfaz de trazo de gráficas	50
4.7 Interfaz de presión de saturación	51
4.8 Interfaz de coeficiente de pérdidas primarias	52
4.9 Ejemplo del proceso de resolución de problemas	53
Conclusiones y recomendaciones	57
Fuentes de información	58
ANEXOS	59
ANEXO A. Tablas de notación y unidades de medición utilizadas en la GUI	60
ANEXO B. Códigos de las interfaces gráficas	63

Índice de figuras

Figura 2.1 Diagrama de Moody	10
Figura 2.2 Corte meridional de una turbomáquina	13
Figura 2.3 Corte transversal de una turbomáquina	14
Figura 2.4 Triángulos de velocidades a la entrada y salida	16
Figura 2.5 Diversas formas de rodets adaptadas a los requerimientos de Q y H.....	19
Figura 2.6 Sistema difusor de una bomba centrífuga.....	20
Figura 2.7 Esquema de la instalación de una bomba centrífuga.....	22
Figura 2.8 Rodete de una turbina Pelton.....	29
Figura 2.9 Diagrama de velocidades ideales de una turbina Pelton.....	30
Figura 2.10 Rodetes característicos de turbinas de reacción.....	31
Figura 2.11 Distribuidor Fink de una turbina de reacción.....	32
Figura 2.12 Ejemplo de definición de una función.....	36
Figura 2.13 Código de ejemplo del comando plot.....	37
Figura 2.14 Ventana principal del entorno de desarrollo GUIDE.....	37
Figura 2.15 Código de desarrollo de una GUI.....	40
Figura 4.1 Interfaz gráfica de inicio.....	46
Figura 4.2 Mensaje enviado cuando los datos no son suficientes.....	46
Figura 4.3 Interfaz de selección de datos disponibles.....	47
Figura 4.4 Ejemplo de una interfaz de cálculos.....	48
Figura 4.5 Interfaz de resultados.....	48
Figura 4.6 Interfaz gráfica de triángulos de velocidades.....	49
Figura 4.7 Interfaz de trazo de gráficas.....	50
Figura 4.8 Ventana de exportar datos.....	51
Figura 4.9 Interfaz de presión de saturación.....	51
Figura 4.10 Interfaz de cálculo del coeficiente de pérdidas primarias.....	52
Figura 4.11 Datos para calcular coeficiente de pérdidas primarias.....	52
Figura 4.12 Selección de datos disponibles.....	54
Figura 4.13 Inserción de datos de entrada.....	54
Figura 4.14 Datos de salida.....	55
Figura 4.15 Triángulos de velocidades a la entrada y salida.....	55
Figura 4.16 Rango de datos para trazar gráfica.....	56
Figura 4.17 Gráfica de ejemplo (rpm – Potencia de accionamiento).....	56

Índice de tablas

Tabla 2.1 Ecuaciones para calcular el coeficiente de pérdidas primarias.....	9
Tabla 2.2 Coeficientes de pérdida para algunos accesorios comunes.....	11
Tabla 2.3 Componentes gráficos creados con la función uicontrol.....	38
Tabla 2.4 Propiedades de uicontrol.....	39

Introducción

Las herramientas computacionales actuales han pasado a formar parte importante del desarrollo académico y profesional, ampliando las posibilidades que se tienen para afrontar los retos que supone la ingeniería.

Existe una gran cantidad de software cuyas tareas son tan diversas y se ajustan a las necesidades de cada usuario. Pero además se cuenta con lenguajes de programación de alto nivel que permiten elaborar aplicaciones propias, las cuales resultan muy útiles cuando se requiere resolver un problema muy específico o relativamente nuevo, o bien cuando los recursos no sean suficientes para adquirir una licencia comercial.

MATLAB es parte de ese gran número de programas útiles en ingeniería, con la ventaja de que proporciona también un lenguaje de programación nativo, lo cual le ha valido una popularidad importante como software líder en el desarrollo de modelos y prototipos en diversas áreas de la ingeniería.

En el desarrollo de este proyecto se aprovecha la versatilidad de MATLAB, mencionada anteriormente, para crear una aplicación que permita resolver y analizar problemas de máquinas hidráulicas, de manera que facilite a los alumnos de ingeniería mecánica su comprensión y les proporcione una herramienta auxiliar de gran utilidad.

CAPÍTULO I. GENERALIDADES DEL PROYECTO

1.1 Justificación

El uso de un programa de computadora para resolver problemas de cálculo en ingeniería permite al usuario centrar su atención en el análisis de las diversas variables o aspectos que afectan un determinado proceso físico y dejar en un segundo plano las tareas de procesar los datos de forma ordinaria, a la vez que implica el ahorro significativo de tiempo. Además, herramientas como el trazo de gráficas le proporcionan una visión más amplia del comportamiento de los problemas en ingeniería.

1.2 Objetivos

1.2.1 Objetivo general

Proporcionar a los estudiantes de ingeniería mecánica del Instituto Tecnológico de Tuxtla Gutiérrez un apoyo didáctico en el uso de MATLAB como herramienta para el análisis de problemas en el área de máquinas de fluidos incompresibles.

1.2.2 Objetivos específicos

- Desarrollar una aplicación en lenguaje MATLAB que permita resolver problemas de máquinas de fluidos incompresibles.
- Proporcionar herramientas adicionales que permitan analizar los procesos relacionados con las máquinas hidráulicas para facilitar su comprensión

1.3 Caracterización del área

El Instituto Tecnológico de Tuxtla Gutiérrez es un centro de estudios de nivel superior ubicado en la Ciudad de Tuxtla Gutiérrez, Chiapas, México.

Fue fundado el 22 de octubre de 1972, por el entonces Gobernador del Estado, Dr. Manuel Velasco Suárez, inicialmente con el nombre de Instituto Tecnológico Regional de Tuxtla Gutiérrez (ITRTG No. 27), posteriormente se llamaría el Instituto Tecnológico de Tuxtla Gutiérrez (ITTG).

1.3.1 Misión

Formar de manera integral profesionistas de excelencia en el campo de la ciencia y tecnología con actitud emprendedora, respeto al medio ambiente y apego a los valores éticos.

1.3.2 Visión

Ser una institución de excelencia en la educación superior tecnológica del sureste, comprometida con el desarrollo socioeconómico sustentable de la región.

1.3.3 Valores

- El ser humano
- El espíritu de servicio
- El liderazgo
- El trabajo en equipo
- La calidad
- El alto desempeño
- Respeto al medio ambiente

1.4 Problema a resolver

El poco uso de herramientas computacionales para resolver problemas relacionados con las máquinas hidráulicas.

1.5 Alcances

El desarrollo del programa está enfocado en resolver ejercicios de máquinas de fluidos incompresibles, específicamente bombas centrífugas, turbinas de acción y reacción, y además proporcionar herramientas adicionales que faciliten la comprensión de la naturaleza de las turbomáquinas, por ello se incluyen diversas interfaces para graficar datos de interés para el usuario, trazar triángulos de velocidades, calcular coeficientes de pérdidas, entre otras, que permiten resolver los problemas mencionados en un menor tiempo y dando al usuario la posibilidad de centrarse en aspectos de análisis de resultados.

1.6 Limitaciones

MATLAB es muy conocido por su gran capacidad para el cálculo numérico, sin embargo no sucede lo mismo al momento de realizar procesos con cálculo simbólico, aun cuando dispone de un toolbox específico su capacidad de trabajar con expresiones simbólicas es poco comparable con otros software especializados en ese aspecto, tal como Maple.



Por lo tanto es evidente que al momento de proporcionar los datos en las cajas de texto correspondientes de la interfaz de cálculos, estos sean un valor de tipo numérico y no un valor simbólico.

El sistema de unidades utilizado ha sido el Sistema Internacional de Unidades (SI), por lo cual si el usuario necesita realizar cálculos en otro sistema de unidades tendrá que recurrir al conversor de unidades integrado en el programa.

Las prestaciones del programa para resolver los diversos tipos de problemas, como en la mayoría de los programas, están limitadas a la interpretación del usuario acerca de los datos que tiene disponible para un determinado cálculo.

CAPÍTULO II. FUNDAMENTO TEÓRICO

2. Fundamento teórico

2.1 Pérdidas en sistemas de tuberías

Las pérdidas de carga en la tubería son de dos clases: primarias y secundarias.

Las pérdidas de primarias son las pérdidas de superficie en el contacto del fluido con la tubería (capa límite), rozamiento de unas capas de fluido con otras (régimen laminar) o de las partículas de fluido entre sí (régimen turbulento). Tienen lugar en flujo uniforme, por tanto principalmente en los tramos de tubería de sección constante.

Las pérdidas secundarias son las pérdidas de forma, que tienen lugar en las transiciones (estrechamientos o expansiones de la corriente), codos, válvulas, y en toda clase de accesorio de tubería.

Si la conducción es larga (oleoductos, gasoductos, etc.) las pérdidas secundarias tienen poca importancia, pudiendo incluso despreciarse. Si la conducción es corta y complicada, las pérdidas secundarias juegan un papel preponderante.

En el cálculo de las pérdidas de carga en tuberías juegan un papel discriminante dos factores: el que la tubería sea lisa o rugosa y el que el régimen de corriente sea laminar o turbulento.

2.1.1 Pérdidas primarias

A finales del siglo XIX experimentos realizados con tuberías de agua de diámetro constante demostraron que la pérdida de carga era directamente proporcional al cuadrado de la velocidad media en la tubería y a la longitud de la tubería e inversamente proporcional al diámetro de la misma. La fórmula fundamental que expresa lo anterior es la ecuación de Darcy-Weisbach que se muestra enseguida:

$$H_{rp} = f \frac{L}{D} \frac{v^2}{2g} \quad (2.1)$$

Donde

H_{rp} : Pérdida de carga primaria.

f : Coeficiente de pérdida de carga primaria.

L : Longitud de la tubería.

D : Diámetro de la tubería.

v : Velocidad media del fluido

Esta fórmula es de uso universal en el mundo entero en los libros y formularios de hidráulica. A partir de 1940 aproximadamente se ha venido utilizando cada vez más un gráfico llamado diagrama de Moody que actualmente se ha difundido en el mundo entero.

El diagrama de Moody resuelve todos los problemas de pérdidas de carga primarias en tuberías con cualquier diámetro, cualquier material de tubería y cualquier caudal; puede emplearse con tuberías de sección no circular sustituyendo el diámetro D por el radio hidráulico R_h . Se usa para determinar el coeficiente de pérdida de carga primaria, el cual ha de llevarse luego a la ecuación de Darcy-Weisbach.

Cálculo del coeficiente de pérdidas primarias

Todos los casos que pueden presentarse pueden reducirse a la siguiente clasificación en cuatro casos comunes:

1. Régimen laminar

- Con tuberías lisas
- Con tuberías rugosas

2. Régimen turbulento

- Con tuberías lisas
- Con tuberías rugosas

En general el coeficiente de pérdidas primarias depende del número de Reynolds, y de la relación ϵ/D , es decir:

$$f = f\left(\text{Re}, \frac{\epsilon}{D}\right) \quad (2.2)$$

Pero en algunos casos específicos pueden hacerse simplificaciones significativas, a continuación se describen los casos más comunes de calcular el coeficiente de pérdidas primarias.

En la Tabla 2.1 se muestran un conjunto de ecuaciones resumidas para calcular el coeficiente de pérdidas para diversos casos.

Tabla 2.1 Ecuaciones para calcular el coeficiente de pérdidas primarias

TUBERÍAS	RÉGIMEN	AUTOR	ECUACIÓN (NÚMERO DE ECUACIÓN)
Lisas y rugosas	Laminar	Poiseuille	$f = \frac{64}{Re} \quad (2.3)$
Lisas	Turbulento ($Re < 100000$)	Blasius	$f = \frac{0.316}{Re^{1/4}} \quad (2.4)$
Lisas	Turbulento ($Re < 100000$)	Kármán-Prandtl (primera ecuación)	$\frac{1}{\sqrt{f}} = 2 \log(Re \sqrt{f}) - 0.8 \quad (2.5)$
Rugosas	Turbulento (zona de transición)	Colebrook-White	$\frac{1}{\sqrt{f}} = -2 \log \left(\frac{k/D}{3.7} + \frac{2.51}{Re \sqrt{f}} \right) \quad (2.6)$
Rugosas	Turbulento (zona final)	Kármán-Prandtl (segunda ecuación)	$\frac{1}{\sqrt{f}} = 2 \log \left(\frac{D}{2k} \right) + 1.74 \quad (2.7)$

Fuente: Mataix, 1982.

La ecuación de Poiseuille junto con la ecuación de Colebrook-White permite el cálculo del coeficiente f en todos los casos que pueden presentarse en la práctica. Dichas ecuaciones pueden programarse y resolverse mediante un ordenador de manera muy sencilla. Estas mismas ecuaciones han sido representadas gráficamente en el diagrama de Moody para diversos valores de la rugosidad relativa y el número de Reynolds. En la figura 2.1 se muestra un diagrama de Moody típico.

Para calcular las pérdidas primarias utilizando el diagrama de Moody dados los datos de Q , L , D , v y ϵ han de seguirse los pasos que se indican:

1. Seleccionar el valor de la rugosidad absoluta acorde al material
2. Calcular la rugosidad relativa
3. Calcular el número de Reynolds
4. Leer en el diagrama de Moody el valor de f
5. Sustituir los valores en la ecuación de Darcy-Weisbach

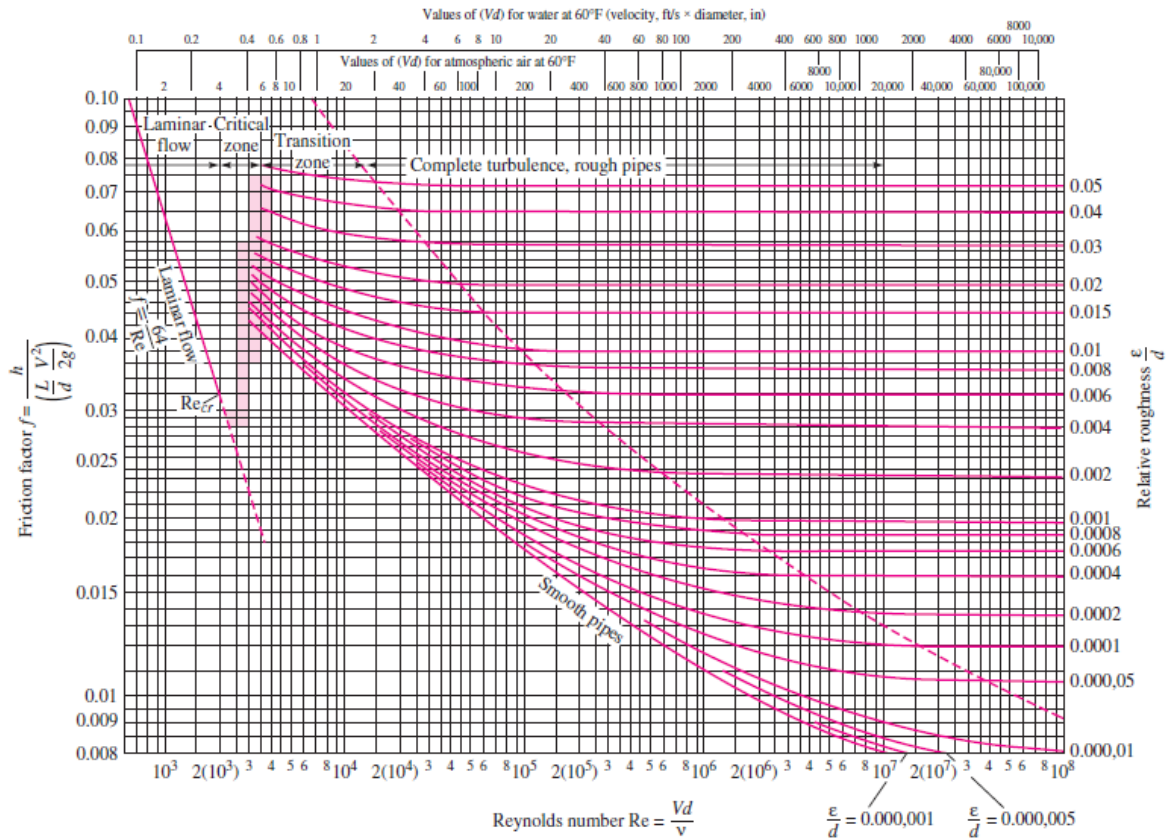


Figura 2.1 Diagrama de Moody

2.1.2 Pérdidas secundarias

Para calcular las pérdidas secundarias podemos hacer uso de la siguiente ecuación, la cual es también muy común en todos los textos de hidráulica:

$$H_{rs} = K \frac{v^2}{2g} \quad (2.8)$$

En donde:

H_{rs} : Pérdida de carga secundaria

K : Coeficiente adimensional de pérdida de carga secundaria

v : Velocidad media en la tubería

El coeficiente K representa la sumatoria de los diversos accesorios que contenga la tubería, es decir:

$$K = \sum_{i=1}^n K_i = K_1 + K_2 + K_3 + \dots + K_n \quad (2.9)$$

Donde cada subíndice representa un accesorio diferente.

Los valores típicos del coeficiente de pérdida de carga K para algunos accesorios comunes pueden observarse en la Tabla 2.2.

Tabla 2.2 Coeficientes de pérdida para algunos accesorios comunes

TIPO DE ACCESORIO	COEFICIENTE DE PÉRDIDA
Codo 45°	0.35
Codo 90°	0.75
Te	1
Retorno en U	1.5
Válvula de compuerta abierta	0.17
Válvula de compuerta semiabierta	4.5
Válvula de globo abierta	6.0
Válvula de globo semiabierta	9.5
Válvula de ángulo abierta	2.0
Válvula de retención de bola	70.0
Válvula de retención de bisagra	2.0

Fuente: Mataix, 1982.

2.2 Máquinas hidráulicas

2.2.1 Generalidades de las máquinas hidráulicas

Las máquinas hidráulicas pertenecen a un grupo muy importante de máquinas que se denominan máquinas de fluido. Las máquinas de fluido son aquellas en las cuales el fluido, o bien proporciona la energía que absorbe la máquina o bien aquellas en que el fluido es el receptor de energía, al que la máquina restituye la energía mecánica absorbida.

La máquina hidráulica es aquella en que el fluido que intercambia su energía no varía sensiblemente su densidad en su paso a través de la máquina, por lo cual en el diseño y estudio de la misma se hace la hipótesis de que $\rho = \text{constante}$.

Clasificación de las máquinas hidráulicas

Para clasificar a las máquinas hidráulicas habrá que poner atención al componente principal de la máquina, es decir aquel en el cual se intercambia la energía mecánica en energía de fluido o viceversa. El componente anterior puede ser el rodete o bien un émbolo.

Basándose en el principio fundamental de funcionamiento, las máquinas hidráulicas se clasifican en turbomáquinas y en máquinas de desplazamiento positivo.

En las máquinas de desplazamiento positivo el órgano intercambiador de energía cede energía al fluido o el fluido a él en forma de energía de presión creada por la variación de volumen. Los cambios en la dirección y valor absoluto de la velocidad del fluido no juegan papel esencial alguno.

En las turbomáquinas, denominadas también máquinas de corriente, los cambios en la dirección y valor absoluto de la velocidad del fluido juegan un papel muy importante.

Tanto las máquinas de desplazamiento positivo como las turbomáquinas se subdividen en motoras y generadoras. Las primeras absorben energía del fluido y restituyen energía mecánica, mientras que las segundas absorben energía mecánica y restituyen energía al fluido.

Ecuación fundamental de las turbomáquinas o ecuación de Euler

La ecuación de Euler es la ecuación fundamental para el estudio de las turbomáquinas. Es la ecuación que expresa la energía intercambiada en el rodete de todas esas máquinas.

Planos de representación de una turbomáquina

Los dos planos de representación de una turbomáquina son el plano o corte el meridional y el plano o corte transversal.

En la figura 2.2 se representa el corte por un plano que contiene al eje de la máquina, que se llama corte meridional, porque en él se representan en su verdadera forma las meridianas de las superficies de revolución de la máquina. En este corte se ven también las aristas de entrada y salida de los álabes, los cuales imparten (bomba) o absorben

(turbina) energía del fluido. Los anchos del rodete a la entrada b_1 y a la salida b_2 se acotan también en este plano de representación.

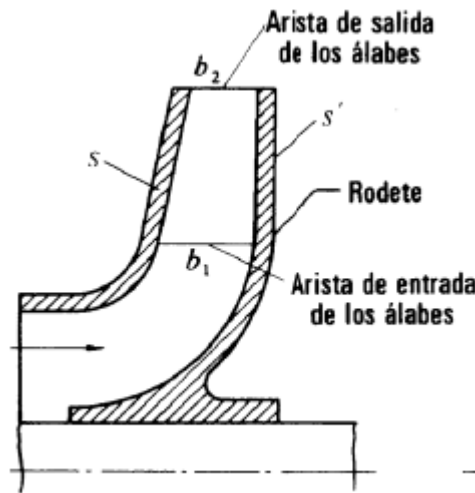


Figura 2.2 Corte meridional

En la figura 2.3 se representa el corte transversal por un plano perpendicular al eje. En el corte transversal de una bomba radial se ve el álabe del rodete en su verdadera forma: el álabe es una superficie cilíndrica con generatrices paralelas al eje de la máquina. Los diámetros de entrada y salida de los álbes, D_1 y D_2 , se acotan e indican en la representación, así como el diámetro del eje, d_e .

Tomando como referencia las figuras 2.2 y 2.3 que representan el rodete de una turbomáquina, específicamente una bomba centrífuga, se hará la deducción de la ecuación de Euler, válida para todas ellas.

Supondremos que la bomba funciona en régimen permanente y que al girar crea una depresión en el rodete penetrando el fluido en el interior de la bomba.

Sea c_1 la velocidad absoluta de una partícula de fluido a la entrada de un álabe. El rodete accionado por el motor de la bomba gira a una velocidad n rpm. En el punto 1 el rodete tiene una velocidad periférica como se indica enseguida:

$$u_1 = \frac{\pi D_1 n}{60} \quad (2.10)$$

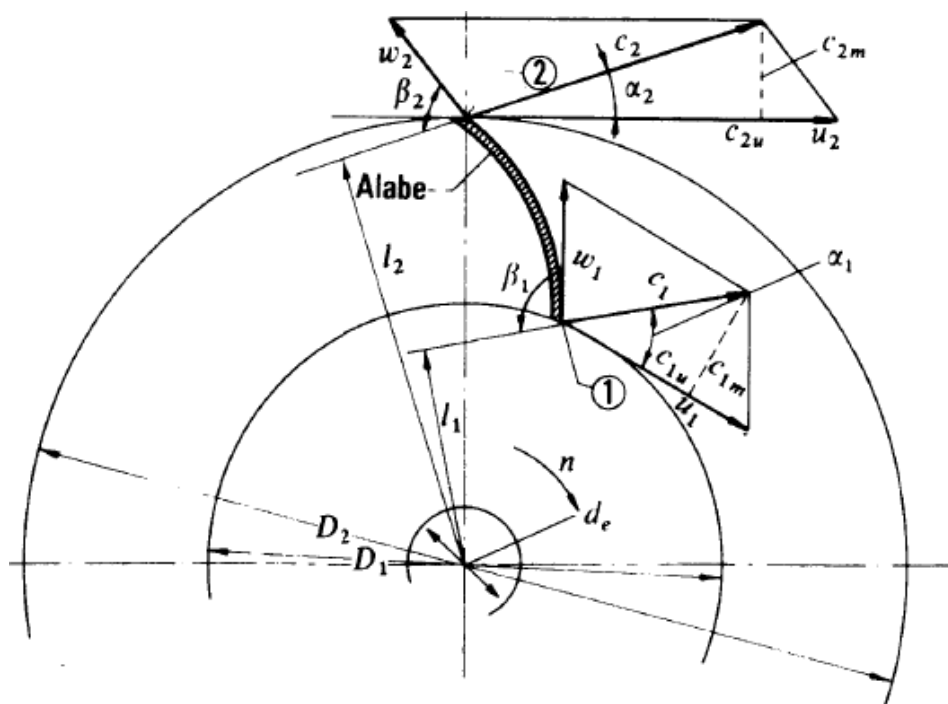


Figura 2.3 Corte transversal de una turbomáquina

Con relación al álabe el fluido se mueve con una velocidad w_1 , llamada velocidad relativa a la entrada. Las tres velocidades c_1 , u_1 y w_1 están relacionadas según la mecánica del movimiento relativo, por la ecuación vectorial:

$$\vec{c}_1 = \vec{u}_1 + \vec{w}_1 \quad (2.11)$$

Suponemos que la tangente del álabe tiene la dirección del vector \vec{w}_1 , con lo que la partícula entra sin choque en el álabe. La partícula guiada por el álabe sale del rodete con una velocidad relativa a la salida \vec{w}_2 , que será tangente álabe en el punto 2. En el punto 2 el álabe tiene una velocidad periférica \vec{u}_2 . La misma composición de velocidades de la ecuación anterior nos proporciona la velocidad absoluta a la salida \vec{c}_2 :

$$\vec{c}_2 = \vec{u}_2 + \vec{w}_2 \quad (2.12)$$

La partícula ha sufrido a su paso por el rodete un cambio de velocidad de \vec{c}_1 a \vec{c}_2 .

Haciendo uso teorema de la cantidad de movimiento se deduce el teorema del momento cinético, el cual aplicado a la línea de corriente al que pertenece la partícula de fluido considerada, será:

$$d\vec{F} = dQ\rho(\vec{c}_2 - \vec{c}_1) \quad (2.13)$$

Tomando momentos de la ecuación anterior respecto al eje de la máquina, se tiene:

$$dM = dQ\rho(l_2c_2 - l_1c_1) \quad (2.14)$$

Siendo,

dM : Momento resultante con relación al eje de la máquina de todas las fuerzas que el rodete ha ejercido sobre las partículas que integran el filamento de corriente considerado para hacerle variar su momento cinético.

dQ : Caudal del filamento

l_2, l_1 : Brazos de momento de los vectores c_2 y c_1 respectivamente.

Supondremos ahora que todas las partículas del fluido entran al rodete a un diámetro D_1 con la misma velocidad c_1 , y salen a un diámetro D_2 con la misma velocidad c_2 . Lo anterior equivale a suponer que todos los filamentos de corriente sufren la misma desviación, lo cual a su vez implica que el número de álabes es infinito para que el rodete guíe al fluido perfectamente. Aplicando esta hipótesis llamada teoría unidimensional o teoría del número infinito de álabes, al hacer la integral de la ecuación 2.14 el paréntesis del segundo miembro será constante, obteniéndose finalmente:

$$M = Q\rho(l_2c_2 - l_1c_1) \quad (2.15)$$

Pero, de la figura 2.3 puede verse con facilidad que:

$$l_1 = r_1 \cos(\alpha_1) \text{ y } l_2 = r_2 \cos(\alpha_2)$$

Por lo cual, sustituyendo en la expresión obtenida anteriormente:

$$M = Q\rho(l_2c_2 \cos \alpha_2 - l_1c_1 \cos \alpha_1) \quad (2.16)$$

Multiplicando el momento por las rpm del rodete obtendremos la potencia que este comunica al fluido, es decir:

$$P_u = M\omega = Q\rho\omega(r_2c_2 \cos \alpha_2 - r_1c_1 \cos \alpha_1) \quad (2.17)$$

Donde $\omega = \frac{2\pi n}{60}$

Si llamamos Y_u a la energía específica intercambiada entre el rodete y el fluido, se tendrá entonces que:

$$P_u = \dot{m}Y_u = Q\rho gH_u \quad (2.18)$$

Igualado la expresión anterior con la ecuación 2.17 se tiene:

$$Y_u = \pm(u_1c_{1u} - u_2c_{2u}) \quad (2.19)$$

$$H_u = \pm \left(\frac{u_1c_{1u} - u_2c_{2u}}{g} \right) \quad (2.20)$$

Las ecuaciones anteriores representan la primera forma de la ecuación de Euler, escrita en términos de energía y en términos de altura. Los signos positivo y negativo se usan dependiendo del tipo de turbomaquinaria, siendo el positivo para máquinas motoras y el negativo para máquinas generadoras.

Triángulos de velocidades

Las ecuaciones vectoriales 2.11 y 2.12 se representan mediante dos triángulos, llamados triángulo de entrada y triángulo de salida, respectivamente.

En la figura 2.4 se muestran ambos triángulos con la notación correspondiente, cuyas descripciones se harán enseguida.

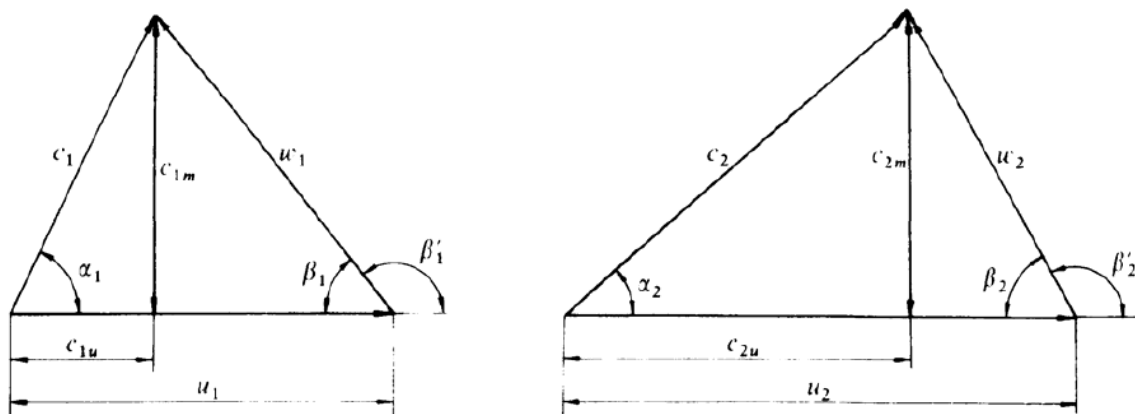


Figura 2.4 Triángulos de velocidades de entrada y salida de una bomba centrífuga

u_1 : Velocidad absoluta del álabe a la entrada o velocidad periférica a la entrada

c_1 : Velocidad absoluta del fluido a la entrada.

w_1 : Velocidad relativa a la entrada (del fluido respecto al álabe)

c_{1m} : Componente meridional de la velocidad absoluta del fluido a la entrada.

c_{1u} : Componente periférica de la velocidad absoluta del fluido a la entrada.

α_1 : Ángulo que forman las dos velocidades c_1 y u_1 .

β_1 : Ángulo que forma w_1 con $-u_1$.

Las descripciones para el triángulo de salida son exactamente las mismas, simplemente cambiando los subíndices correspondientes, y entrada por salida.

Grado de reacción

El grado de reacción de una turbomáquina se refiere al modo como trabaja el rodete. Así, por ejemplo, en una bomba se debe distinguir la altura de presión que da la bomba y la altura de presión que da el rodete de la bomba, H_p . La primera normalmente es mayor que H_p porque la bomba tiene además de un rodete un sistema difusor que transforma la energía dinámica que da el rodete, H_d , en energía de presión, que sumada a la energía presión del rodete constituye la energía de presión que da toda la bomba, análogamente sucede en una turbina. Por tanto:

$$\sigma = \frac{H_p}{H_u} \quad (2.21)$$

Es decir, el cociente de la altura que da (bomba) o absorbe (turbina) el rodete en forma de presión por la altura total que da (bomba) o absorbe (turbina) el rodete (el denominador es la altura de Euler, H_u , en ambos casos), siendo H_u siempre positivo se tienen los siguientes posibles casos del grado de reacción:

Si $H_p < 0$ el grado de reacción es negativo.

Si $H_p = 0$ el grado de reacción es cero.

Si $0 < H_p < H_u$ el grado estará comprendido entre 0 y 1 (más común).

Si $H_p > H_u$ el grado de reacción es mayor a 1.

2.2.2 Bombas centrífugas

Bomba es una máquina que absorbe energía mecánica y restituye al líquido que la atraviesa energía hidráulica.

Elementos constitutivos de una bomba centrífuga

Rodete: gira solidario con el eje de la máquina y consta de un cierto número de álabes que imparten energía al fluido en forma de energía cinética y energía de presión.

Corona directriz: recoge el líquido del rodete y transforma la energía cinética comunicada por el rodete en energía de presión, ya que la sección de paso aumenta en esta corona en la dirección del flujo. Esta corona directriz no existe en todas las bombas; porque encarece su construcción; aunque hace la bomba más eficiente.

Caja espiral: transforma también la energía mecánica en energía de presión, y recoge además con pérdidas mínimas de energía el fluido que sale del rodete, conduciéndolo hasta la tubería de salida o tubería de impulsión.

Tubo difusor troncocónico: realiza una tercera etapa de difusión o sea de transformación de energía dinámica en energía de presión.

El rodete y la clasificación por el número específico de revoluciones

El rodete puede ser de formas muy variadas, e inclusive caprichosas cuando la aplicación particular lo requiere.

Los rodetes se clasifican en cuatro tipos según la forma de sujeción de los álabes, los cuales son:

- a) Rodete cerrado de simple aspiración: las caras anterior y posterior forman una caja: entre ambas caras se fijan los álabes.
- b) Rodete cerrado de doble aspiración.
- c) Rodete semiabierto de simple aspiración: sin la cara anterior, los álabes solo se fijan en la cara posterior.
- d) Rodete abierto de doble aspiración sin cara anterior ni posterior: los álabes se fijan en el núcleo o cubo del rodete.

El rodete de una bomba centrífuga se ha de proyectar de manera que para la Q y H requeridas se obtenga el óptimo rendimiento. En la práctica, los Q y H necesarios varían entre amplios límites, y dentro de ellos puede requerirse cualquier combinación de Q y H con diferentes valores de n , buscando siempre el rendimiento óptimo. Como consecuencia de ello, el rodete de las bombas centrífugas va cambiando insensiblemente de forma para adaptarse a las diferentes condiciones de servicio.

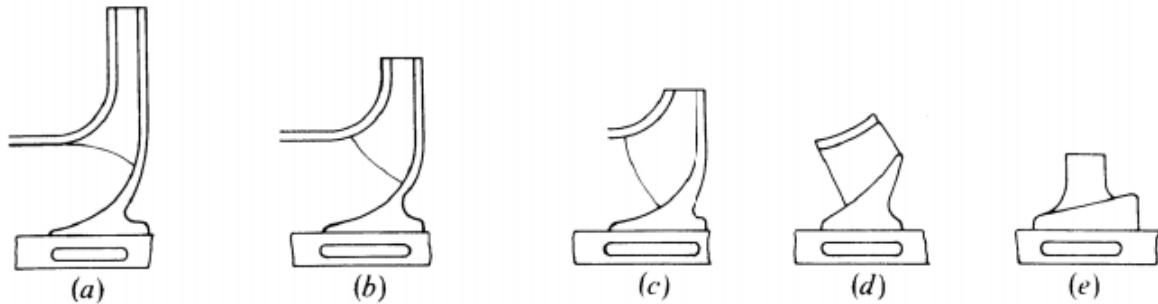


Figura 2.5 Diversas formas de rodetes adaptadas a los requerimientos de Q , H y n .

Así, por ejemplo los rodetes de la figura 2.5 se van poco a poco adaptando a caudales mayores y alturas efectivas más pequeñas. Cada uno de los rodetes de la figura representa a una familia de rodetes geoméricamente semejantes. El tamaño se ajustará a la potencia. Así el tipo e, por ejemplo, podría construirse para 1 kW o para 10 000 kW.

Cada rodete corresponde a un valor de un parámetro de excepcional interés en las turbomáquinas hidráulicas: n_s o número específico de revoluciones. Así, la clasificación más precisa de las bombas centrífugas es una clasificación numérica, asignando a toda la familia de bombas geoméricamente semejantes un número específico de revoluciones, el cual queda definido como se muestra a continuación:

$$n_s = n P^{1/2} H^{-5/4} \quad (2.22)$$

En las bombas centrífugas este número oscila entre 35 y 1800 aproximadamente.

El número específico de revoluciones no es adimensional. Las unidades de n_s que se utilizan en la práctica son muy variadas. En el Sistema Internacional se debería expresar n en rps, P en W y H en m. Sin embargo, en la mayoría de los países regidos por el SI se suelen expresar comúnmente n en rpm, P en CV y H en m.

Si se quiere utilizar el caudal en lugar de potencia para calcular el número específico de revoluciones, entonces recordar que:

$$P = Q\rho gH \text{ (Medida en watts en el SI)}$$

Para convertir la potencia de watts a CV, utilizamos la ecuación que se indica enseguida:

$$P = \frac{Q\rho H}{75} \text{ (Medida en CV)}$$

Sustituyendo el valor de P en la ecuación anterior dada para calcular n_s y suponiendo que el líquido utilizado es agua, podemos simplificar como sigue y obtener una nueva expresión:

$$n_s = nP^{1/2}H^{-5/4} = n\left(\frac{Q\rho H}{75}\right)^{1/2}H^{-5/4}$$

$$n_s = 3.65nQ^{1/2}H^{3/4} \quad (2.23)$$

El sistema difusor

El sistema difusor de una bomba consta de tres elementos:

- Corona directriz
- Caja espiral
- Cono difusor

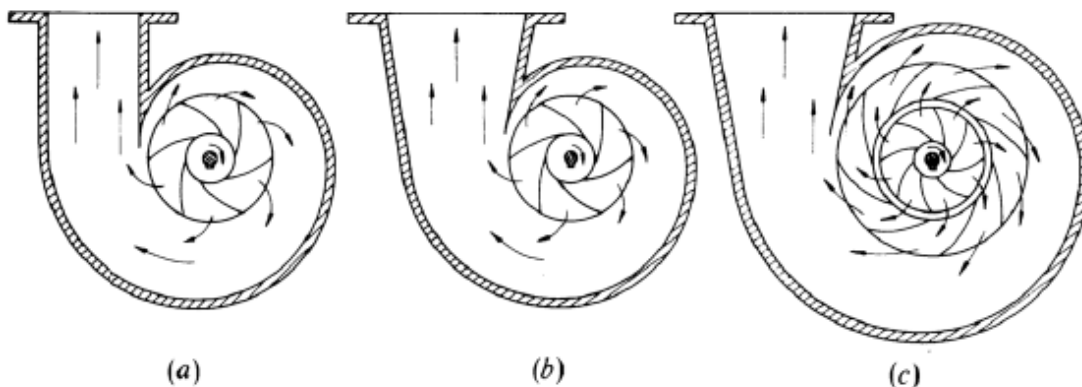


Figura 2.6 Sistema difusor de una bomba que puede ser: a) solo con caja espiral, b) con caja espiral y cono difusor y c) con caja espiral, corona directriz y cono difusor.

El papel de estos tres elementos es el mismo: transformar la energía dinámica que da el rodete en energía de presión con el mínimo posible de pérdidas. El nombre de caja espiral se deriva de una construcción especial de la misma que consiste en una caja formada por dos planos paralelos y cerrada por una superficie cilíndrica cuya directriz es una espiral logarítmica.

Instalación de una bomba centrífuga

La figura 2.7 representa una instalación de bombeo destinada a elevar agua desde un pozo de aspiración hasta un depósito elevado, en esta instalación pueden observarse los siguientes elementos:

Válvula de pie con filtro

Evita la entrada de suciedades al sistema que puedan obstruir el funcionamiento adecuado de la bomba, además que hace posible el cebado de la bomba reteniendo el líquido necesario para tal fin.

Válvulas de compuerta en la aspiración e impulsión.

A veces no se instala la primera; pero de la segunda no se prescinde nunca porque sirve para la regulación del caudal de la bomba.

Válvula de retención en la impulsión

Impide el retroceso del fluido cuando la bomba se para. Es imprescindible si la tubería de impulsión es muy larga o se encuentra a gran presión.

Reductor en la aspiración

Para mejorar la aspiración de la bomba y evitar la cavitación se aumenta a veces el diámetro de la tubería de aspiración. La reducción se hace con un accesorio como el mostrado en la figura anterior para evitar la formación de bolsas de aire en la parte superior.

Para el estudio de la bomba y de la instalación es importante considerar las secciones siguientes que se han indicado en la figura 2.7:

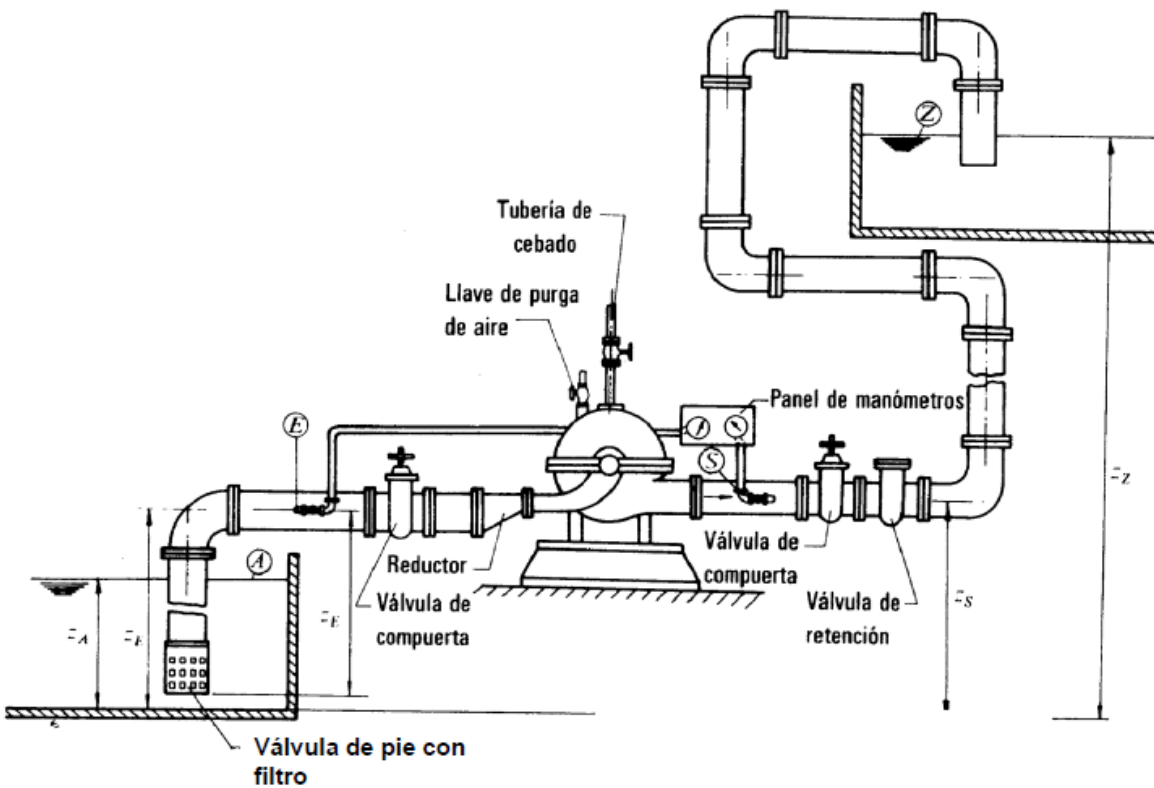


Figura 2.7 Esquema de la instalación de una bomba centrífuga

- Sección A: nivel superior del agua en el pozo de aspiración.
- Sección Z: nivel superior del agua en el depósito de impulsión.
- Sección E: entrada a la bomba.
- Sección S: salida de la bomba.

Altura útil de una bomba

Una instalación consta de una serie de metros de tubería y accesorios; en los tramos rectos hay pérdidas primarias y en los accesorios pérdidas secundarias. El conjunto de las pérdidas anteriores se conocen como pérdidas externas (H_{r-ext}). Además de las pérdidas externas, en el interior de la bomba ocurren pérdidas por superficie y forma, las cuales se agrupan en las pérdidas internas (H_{r-int}).

La altura teórica que da una bomba centrífuga viene dada por la ecuación 2.24, es decir:

$$H_u = \frac{u_2 c_{2u} - u_1 c_{1u}}{g} \quad (2.24)$$

Donde 1 y 2 hacen referencia a la entrada y salida del rodete, respectivamente. H_u es la altura que el rodete imparte al fluido. Si no hubiera pérdidas en el interior de la bomba, H_u sería también el aumento de altura que experimentaría el fluido entre la entrada y salida de la bomba. Sin embargo, en el interior de la bomba (entre las secciones E y S) se producen pérdidas hidráulicas que se mencionaron con anterioridad, por lo cual la altura útil que da la bomba es la altura que imparte el rodete o altura teórica, H_u , menos las pérdidas en el interior de la bomba (H_{r-int}), es decir:

$$H = H_u - H_{r-int} \quad (2.25)$$

Si escribimos la ecuación de Bernoulli entre las secciones E y S, se tiene:

$$\frac{P_E}{\rho g} + z_E + \frac{v_E^2}{2g} + H = \frac{P_S}{\rho g} + z_S + \frac{v_S^2}{2g} \quad (2.26)$$

Por lo cual:

$$H = \frac{P_S - P_E}{\rho g} + z_S - z_E + \frac{v_S^2 - v_E^2}{2g} \quad (2.27)$$

El término $z_S - z_E$ suele ser muy pequeño e incluso cero en bombas de eje vertical.

El término $\frac{v_S^2 - v_E^2}{2g}$ suele ser también muy pequeño o igual a cero: positivo, aunque pequeño si el diámetro de la tubería de aspiración se hace mayor que el de la tubería de impulsión.

Luego, generalmente en muchas de las ocasiones se cumple la ecuación siguiente:

$$H = \frac{P_S - P_E}{\rho g} = M_S + M_E \quad (2.28)$$

Donde M_S es la lectura del manómetro a la salida y M_E la lectura del manómetro a la entrada de la bomba. El signo positivo se pone para designar la suma de las presiones absolutas o bien presiones manométricas considerando que a la entrada suele medirse con un vacuómetro.

Si planteamos la ecuación de Bernoulli entre las secciones A y Z del esquema mostrado en la figura 2.7, se tiene:

$$\frac{P_A}{\rho g} + z_A + \frac{v_A^2}{2g} + H - H_{r-ext} = \frac{P_Z}{\rho g} + z_Z + \frac{v_Z^2}{2g} \quad (2.29)$$

Las pérdidas externas pueden reescribirse como sigue:

$$H_{r-ext} = H_{ra} + H_{ri} + \frac{v_t^2}{2g} \quad (2.30)$$

En donde:

H_{ra} : Pérdidas en la tubería de aspiración

H_{ri} : Pérdidas en la tubería de impulsión

$\frac{v_t^2}{2g}$: Pérdida secundaria en el desagüe en el depósito.

Además, generalmente las áreas del pozo de aspiración y del depósito de impulsión suelen ser lo suficientemente grandes y despreciar los términos $\frac{v_A^2}{2g}$ y $\frac{v_Z^2}{2g}$, por lo cual la expresión para la altura útil sería:

$$H = \frac{P_Z - P_A}{\rho g} + z_Z - z_A + H_{ra} + H_{ri} + \frac{v_t^2}{2g} \quad (2.31)$$

Es muy común que las presiones en el depósito de impulsión y aspiración sean exactamente la atmosférica, lo cual implicaría quitar dichos términos de la ecuación anterior.

Pérdidas, potencias y rendimientos

Las pérdidas internas en la bomba (entre las secciones E y S de la figura 2.7) se pueden clasificar en tres grupos:

- Pérdidas hidráulicas
- Pérdidas volumétricas
- Pérdidas mecánicas

Las pérdidas hidráulicas disminuyen la energía específica útil que la bomba comunica al fluido y consecuentemente la altura útil. Son de dos clases: pérdidas de superficie y pérdidas de forma.

Las pérdidas volumétricas, también llamadas pérdidas intersticiales, son pérdidas de caudal y se dividen en dos clases: pérdidas exteriores q_e y pérdidas interiores q_i . Las pérdidas exteriores constituyen una salpicadura del fluido al exterior, que se escapa por el juego entre la carcasa y el eje de la bomba, que la atraviesa. Las pérdidas interiores son las más importantes y reducen mucho el rendimiento volumétrico de algunas bombas; se producen debido a que a la salida del rodete de una bomba hay más presión que a la entrada, luego parte del fluido en vez de seguir a la caja espiral retrocederá, por el conducto que forma el juego del rodete con la carcasa, a la entrada del rodete para volver a ser impulsado por la bomba.

Las pérdidas mecánicas incluyen las pérdidas por: rozamiento del prensaestopas con el eje de la máquina, rozamiento del eje con los cojinetes, accionamiento de auxiliares.

Potencia de accionamiento

Es la potencia en el eje de la bomba o potencia mecánica que la bomba absorbe, la cual se expresa mediante la siguiente ecuación:

$$P_a = M\omega = \frac{2\pi nM}{60} = 0.1047nM \quad (2.32)$$

La cual es una expresión muy útil en ensayos de bombas realizados en los bancos de prueba, donde n se mide con un cuentarrevoluciones y M con un torsiómetro o midiendo el par de reacción con un motor de accionamiento basculante.

Potencia interna

Es la potencia total transmitida al fluido, o sea la potencia de accionamiento, descontando las pérdidas mecánicas:

$$P_i = P_a - P_m^r \quad (2.33)$$

Es fácil hallar una expresión hidráulica de P_i en función de las pérdidas llamadas internas (hidráulicas y volumétricas).

En efecto el rodete entrega al fluido una energía específica equivalente a una altura $H_u = H + H_{r-int}$ y esta altura se la entrega al caudal bombeado por el rodete, que es $Q + q_e + q_i$, luego:

$$P_i = (Q + q_e + q_i) \rho g (H + H_{r-int}) = (Q + q_e + q_i) \rho g H_u \quad (2.34)$$

Potencia útil

Es la potencia de accionamiento descontando todas las pérdidas de la bomba o equivalentemente la potencia interna descontando todas y sólo las pérdidas internas, luego:

$$P = P_a - P_m^r - P_v^r - P_h^r = P_i - P_v^r - P_h^r \quad (2.35)$$

La potencia útil por otra parte será la invertida en impulsar el caudal útil Q a una altura útil H , entonces:

$$P = Q \rho g H \quad (2.36)$$

Rendimiento hidráulico

Tiene en cuenta todas y sólo las pérdidas de altura total, por lo cual:

$$\eta_h = \frac{H}{H_u} \quad (2.37)$$

Rendimiento volumétrico

Tiene en cuenta las pérdidas volumétricas y su valor es:

$$\eta_v = \frac{Q}{Q + q_e + q_i} \quad (2.38)$$

Rendimiento interno

Tiene en cuenta las pérdidas internas, hidráulicas y volumétricas, y engloba ambos rendimientos, entonces:

$$\eta_i = \frac{P}{P_i} \quad (2.39)$$

O bien

$$\eta_i = \eta_h \eta_v \quad (2.40)$$

Rendimiento mecánico

Está dado por la siguiente ecuación:

$$\eta_m = \frac{P_i}{P_a} \quad (2.41)$$

Rendimiento total

Tiene en cuenta todas las pérdidas en la bomba y su valor es:

$$\eta_{tot} = \frac{P}{P_a} \quad (2.42)$$

También suele expresarse como el producto de los otros rendimientos, es decir:

$$\eta_{tot} = \eta_i \eta_m = \eta_h \eta_v \eta_m \quad (2.43)$$

Altura neta de succión

La altura neta de succión o NPSH (Net Positive Suction Head) es la carga disponible a la entrada de la bomba para evitar la cavitación o evaporación del líquido. La entrada de la bomba, o zona de succión, es la región donde la presión es más baja y donde puede aparecer antes la cavitación. La NPSH se define como:

$$NPSH = \frac{P_E - P_{sv}}{\rho g} + \frac{v_E^2}{2g} \quad (2.44)$$

En donde:

P_E : Presión a la entrada de la bomba

P_{sv} : Presión de saturación del líquido a la temperatura de bombeo

v_E : Velocidad del líquido a la entrada de la bomba

Si la entrada de la bomba está situada a una altura H_s por encima de un depósito cuya superficie libre está a una presión P_A , podemos usar la ecuación de Bernoulli para escribir la NPSH como:

$$NPSH = \frac{P_A - P_{sv}}{\rho g} - H_s - H_{ra} \quad (2.45)$$

En donde H_{ra} representa las pérdidas en la sección de aspiración de la bomba.

2.2.3 Turbinas de acción

Las turbinas de acción tienen la peculiaridad de aprovechar solamente la energía cinética del fluido; no existe gradiente de presión entre la entrada y salida de la máquina.

En la turbina Pelton actual la energía cinética del agua, en forma de chorro libre, se genera en una tobera colocada al final de una tubería de presión. La tobera está provista de una aguja de cierre para regular el gasto, constituyendo el conjunto, el órgano de alimentación y de regulación de la turbina.

El álabe tiene la forma de doble cuchara, con una arista diametral sobre la que incide el agua, produciéndose una desviación simétrica en dirección axial, buscando un equilibrio dinámico de la máquina en esa dirección. Por ser el ataque del agua en sentido tangencial a la rueda también se le denomina turbina tangencial; por tener el fluido un recorrido axial a su paso por el álabe, se clasifica entre las máquinas de tipo axial.

Encuentra justa aplicación la turbina Pelton en aquellos aprovechamientos hidráulicos donde la ponderación de la carga es importante respecto al caudal.

La clasificación más general que puede hacerse de las turbinas Pelton es en tipos de eje horizontal y vertical.

En la disposición de eje horizontal el número de chorros por rueda se reduce generalmente a uno o dos, por resultar complicada la instalación en un plano vertical de las tuberías de alimentación y las agujas de inyección. La rueda queda sin embargo, más accesible para su inspección, lo mismo que los inyectores, con lo que la reparación de averías pequeñas y desgastes por erosión pueden efectuarse sin desmontar la turbina. Encuentra así aplicación, este sistema de montaje, en aquellos casos donde se tienen aguas sucias que producen deterioros o notable acción abrasiva. Con el eje horizontal se

hace también posible instalar turbinas gemelas para un solo generador colocado entre ambas, contrarrestando empujes axiales.

Con la disposición en eje vertical se facilita la colocación del sistema en un plano horizontal, lo que permite aumentar el número de chorros por rueda, se puede así incrementar el caudal y tener mayor potencia por unidad.

El rodete Pelton está constituido por un disco de acero con álabes periféricos en forma de doble cuchara. Estos puede estar fundidos con el disco en una sola pieza o individualmente, sujetándose después al disco mediante bulones.

Triángulos de velocidades de la turbina Pelton

En la figura 2.8 puede verse la forma de las cucharas. Las diferentes dimensiones suelen expresarse tomando como unidad el diámetro del chorro cuando la turbina trabaja a carga nominal, generalmente $\frac{3}{4}$ de la carga máxima.

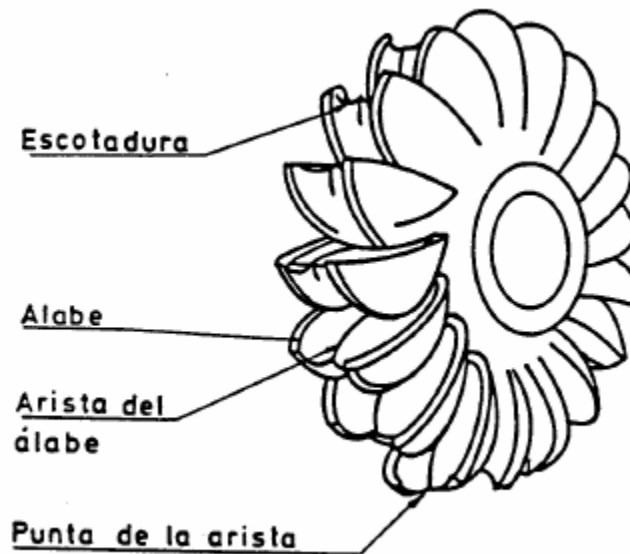


Figura 2.8 Rodete de una turbina Pelton

La trayectoria de una partícula de agua en la cuchara es tangencial, de manera que en las turbinas Pelton se verifica siempre que:

$$u_1 = u_2 = u \quad (2.46)$$

Si no hay rozamiento al ser el flujo en la cuchara de lámina libre idealmente:

$$w_1 = w_2 \quad (2.47)$$

La velocidad real w_2 es algo menor que w_1 .

Si no hay pérdidas en el inyector el chorro sale de este a la atmosfera con una velocidad c_1 que según la ecuación de Torricelli idealmente será:

$$c_1 = \sqrt{2gH} \quad (2.48)$$

Prácticamente, la velocidad real es algo más pequeña y aproximadamente:

$$c_1 = 0.97\sqrt{2gH} \quad (2.49)$$

Idealmente se demuestra que una turbina Pelton alcanza su óptimo rendimiento cuando $u_1 = \frac{1}{2} c_1$. Aunque el rendimiento óptimo suele alcanzarse a una velocidad un poco menor:

$$u_1 = 0.45\sqrt{2gH} \quad (2.50)$$

Idealmente, el ángulo $\alpha_1 = 0^\circ$ y $\beta_1 = 180^\circ$, realmente el ángulo α_1 suele ser algo mayor, aunque siempre muy pequeño.

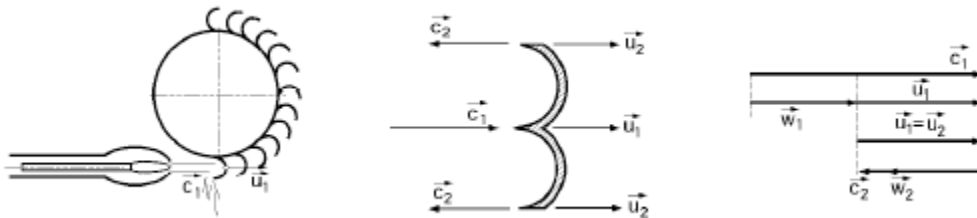


Figura 2.9 Diagrama de velocidades ideales de una turbina Pelton

La turbina Pelton no tiene tubo de aspiración. Como consecuencia no puede aprovecharse la velocidad de salida. Por tanto como la energía cinética a la salida del álabe se pierde es conveniente que sea 0, de esta manera el álabe habrá aprovechado toda la energía, es decir, idealmente $c_2=0$.

El número específico de revoluciones para una turbina Pelton viene dado por la ecuación siguiente:

$$n_s = n P_a^{1/2} H^{-5/4} \quad (2.51)$$

Donde n son las revoluciones por minutos (rpm), P_a la potencia útil dada en unidades de CV y H la altura útil en metros.

2.2.4 Turbinas de reacción

Las turbinas de reacción son de admisión total, es decir, el fluido entra al rodete por toda la periferia. Están constituidas por una tubería forzada, la cámara espiral, el distribuidor, el rodete y el tubo de aspiración o descarga. En este grupo, se encuentran varios tipos de turbinas: las turbinas Francis, que son radiales o mixtas, y las turbinas de Hélice, Kaplan y Bulbo, todas ellas axiales.



Figura 2.10 Rodetes característicos de turbinas de reacción, a la izquierda un rodete Francis, a la derecha un rodete Kaplan.

La tubería forzada conecta el embalse con la cámara espiral. Está diseñada para producir una mínima pérdida de carga, y con chimenea de equilibrio para evitar golpes de ariete, sobre todo para saltos grandes. La tubería es tanto más corta y más gruesa cuanto mayor es la velocidad específica de la turbina, ya que operan a menor altura y mayor caudal.

La cámara espiral tiene como fin igualar las velocidades del fluido en toda la periferia, antes de la entrada del agua al distribuidor.

El distribuidor (Ver figura 2.11) está constituido por una serie de álabes guías orientados, situados en todo el perímetro de entrada de fluido desde la cámara espiral hacia el rodete, encargados de dirigir el fluido hacia el rodete en la dirección más adecuada para aprovechar al máximo su energía. Dado que el ángulo de entrada de la velocidad absoluta al rodete óptimo puede variar con el caudal de agua que atraviesa la turbina, estos álabes poseen un eje que permite orientarlos.

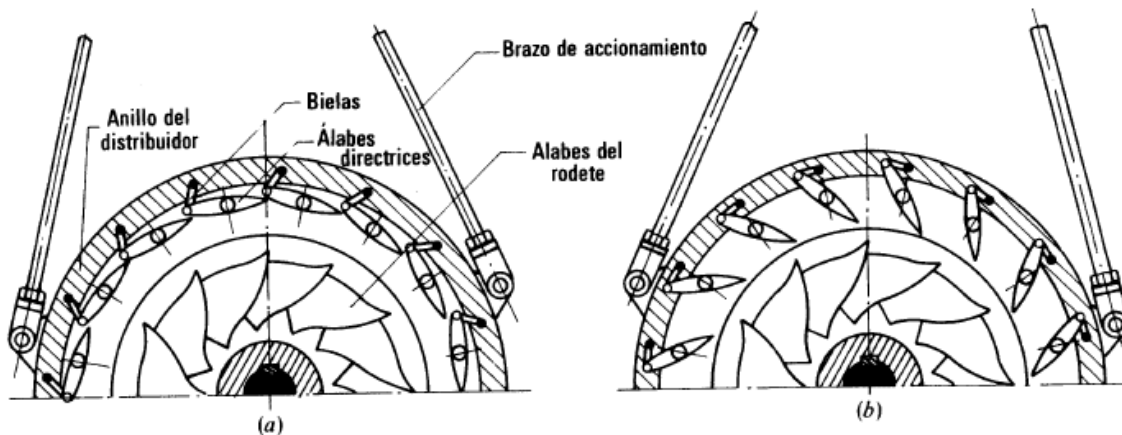


Figura 2.11 Distribuidor Fink de una turbina de reacción a) en posición cerrada, b) en posición abierta.

Las características del rodete de una turbina de reacción dependen del tipo de turbina y, dentro de esta, del valor de la velocidad específica.

El tubo de aspiración o descarga, en turbinas, es interior a la propia turbina, es decir, su salida al canal exterior es lo que señala la sección de salida de la turbina. Su diseño es fundamental, tanto más cuanto menor es la carga del fluido a la entrada de la turbina. De hecho, representa alrededor del 10 % de la altura en turbinas Francis, y hasta el 20-38 % en turbinas Kaplan.

Altura neta

La altura neta es la altura puesta a disposición de la turbina y se calcula tomando la diferencia de energía específica entre las secciones de entrada y salida de la turbina.

La altura neta no es la altura útil aprovechada por la turbina, sino la altura teórica que hubiera aprovechado idealmente (sin pérdidas). Parte de esta altura se disipa, pues, en pérdidas hidráulicas, y el agua intercambia con el rodete una altura menor que la que ha absorbido. Esta última altura que en este caso es la altura hidráulica útil es la altura de Euler. Por tanto:

$$H = H_u + H_{r-int} \quad (2.52)$$

Tanto en las bombas como turbinas, la altura de Euler representa el equivalente en altura de la energía intercambiada entre el fluido y el rodete.

Tomando en cuenta lo anterior y planteando la ecuación de Bernoulli entre las secciones de entrada y salida, podemos escribir la ecuación de la altura útil como sigue:

$$H = \frac{P_E - P_S}{\rho g} + z_E - z_S + \frac{v_E^2 - v_S^2}{2g} \quad (2.53)$$

Para utilizar la ecuación anterior es necesario tener en cuenta los siguientes puntos:

- En toda turbina $P_S / \rho g = 0$ y $z_S = 0$ si se toma como referencia el plano de salida.
- En una turbina Pelton $v_S^2 / \rho g = c^2 / \rho g \simeq 0$
- En toda turbina $v_S^2 / 2g$ es muy pequeña y muchas veces puede despreciarse.
- $P_E / \rho g$ se calcula leyendo convenientemente el manómetro instalado a la entrada de la turbina; $v_E^2 / 2g$ se calcula midiendo el caudal y la sección de entrada.

Si se escribe la ecuación de Bernoulli entre las secciones de aspiración (nivel de aguas arriba) y el nivel de aguas abajo se tiene:

$$\frac{P_A}{\rho g} + z_A + \frac{v_A^2}{2g} - H - H_{r-ext} = \frac{P_Z}{\rho g} + z_Z + \frac{v_Z^2}{2g} \quad (2.54)$$

Donde H_{r-ext} son las pérdidas exteriores a la turbina. Podemos definir la diferencia $z_A - z_Z = H_b$ como altura bruta, además considerando que las presiones en ambos puntos son iguales a la atmosférica y que prácticamente las velocidades son también nulas o despreciables, entonces se tiene:

$$H = H_b - H_{r-ext} \quad (2.55)$$

Siendo:

$$H_{r-ext} = H_{ra} + H_{ri} \quad (2.56)$$

Lo que implica:

$$H = H_b - H_{ra} - H_{ri} \quad (2.57)$$

Pérdidas, potencias y rendimientos

Lo mismo que en las bombas, las pérdidas se clasifican en tres grupos: pérdidas hidráulicas, pérdidas volumétricas y pérdidas mecánicas

La potencia teórica puesta a disposición de la turbina es:

$$P = Q\rho gH \quad (2.58)$$

La potencia útil está dada por:

$$P_a = M\omega = 0.1047nM \quad (2.59)$$

La potencia interna es la potencia suministrada por la turbina descontando la potencia necesaria para vencer los rozamientos mecánicos:

$$P_i = P_a + P_m^r \quad (2.60)$$

Rendimiento hidráulico

$$\eta_h = \frac{H_u}{H} = \frac{u_1 c_{1u} - u_2 c_{2u}}{gH} \quad (2.61)$$

Rendimiento volumétrico

$$\eta_v = \frac{Q - q_e - q_i}{Q} \quad (2.62)$$

Rendimiento interno

$$\eta_i = \frac{P_i}{P} \quad (2.63)$$

Rendimiento mecánico

$$\eta_m = \frac{P_a}{P_i} \quad (2.64)$$

Rendimiento total

$$\eta_{tot} = \frac{P_a}{P} \quad (2.65)$$

$$\eta_{tot} = \eta_i \eta_m = \eta_h \eta_v \eta_m \quad (2.66)$$

2.3 Desarrollo de interfaces gráficas de usuario en MATLAB

2.3.1 Introducción a MATLAB

MATLAB en su contenido básico es un entorno integrado de trabajo que permite el análisis y la computación matemáticos interactivos de forma sencilla, con una gran cantidad de funciones y herramientas adicionales que facilitan la labor del usuario en el desarrollo de modelos y esquemas de simulación. Es además un lenguaje de programación interpretado en el cual pueden implementarse diversos algoritmos, difiriendo con los lenguajes tradicionales debido a la rapidez con la que pueden desarrollarse prototipos muy sofisticados de modelos de ingeniería, aunque generalmente esto represente un sacrificio del rendimiento del código.

2.3.2 Scripts y funciones

En MATLAB se puede crear y guardar código en archivos llamados archivos-m. Un archivo-m es un archivo de texto ASCII similar a los archivos de código fuente de C, Fortran o cualquier otro lenguaje de programación. Se puede crear y editar con el editor/depurador de archivos-m de MATLAB.

Existen dos tipos de archivos-m, llamados scripts y funciones. Un script es simplemente una lista de comandos de MATLAB escritos de manera estructurada que se guardan en un archivo-m para su posterior ejecución desde el área de trabajo. Un ejemplo de script muy sencillo se muestra en la figura 2.13.

Una función es una pieza de código de computación que acepta un argumento de entrada del usuario y produce salida al programa. Las funciones ayudan a programar eficientemente, permitiendo en muchas ocasiones la reutilización de código de uso frecuente.

Las funciones definidas por el usuario se almacenan como archivos-m y MATLAB puede acceder a ellas si están almacenadas en la ruta actual. Cada función creada por el usuario debe comenzar con una línea de definición que contiene:

- La palabra reservada `function`
- Una o más variables que definan la salida de la función
- Un nombre de función, y
- Una o más variables que se usen para el argumento de entrada

Una definición típica de función se muestra enseguida:

```
function s = sumanum(a,b)
s = a+b;
end
```

Figura 2.12 Ejemplo de definición de una función

Una función debe guardarse preferentemente en un archivo del mismo nombre que ella, para facilitar la búsqueda de MATLAB al momento de hacer referencia o llamarle desde otra rutina. Algunas recomendaciones para nombrar funciones son las siguientes:

- El nombre de la función debe comenzar con una letra.
- Puede formarse con letras, números y el guión bajo
- No se pueden usar palabras reservadas
- Permite cualquier longitud, aunque los nombres largos evidentemente no son una buena práctica en programación.

2.3.3 Trazo de gráficas bidimensionales

Para trazar una gráfica en dos dimensiones en MATLAB se pueden utilizar dos procedimientos que generan un resultado muy similar: el trazado utilizando vectores o arreglos unidimensionales o bien el trazado utilizando una expresión simbólica.

Para graficar utilizando vectores se usa el comando `plot`, cuya estructura general es la siguiente: `plot(x, y, 'Propiedad', 'Valor')`.

Donde los primeros argumentos son dos vectores y los siguientes pueden ser una lista de modificadores de propiedades con su valor correspondiente. En la figura 2.13 puede observarse una porción de código ejemplo del comando `plot`.

```
x = 0:0.1:10;  
y = exp(-x/2).*cos(3*x);  
plot(x,y,'color','r','linewidth',2);
```

Figura 2.13 Código ejemplo del comando plot

2.3.4 Interfaces gráficas utilizando GUIDE

MATLAB dispone de un entorno de desarrollo integrado para crear interfaces gráficas de usuario de una manera más sencilla y rápida. En la figura 2.14 se muestra la ventana principal de dicho entorno en la versión R2012b.

Para crear una interfaz gráfica utilizando GUIDE bastará con arrastrar los elementos necesarios al componente principal y ubicarlos en la posición deseada y definir un tamaño a conveniencia mediante las formas habituales de un software común. Enseguida, la parte interesante la supone la porción de código que el programador insertará y que dará una utilidad específica a todos los elementos gráficos, para ello bastará con modificar el código autogenerated por MATLAB en las secciones correspondientes

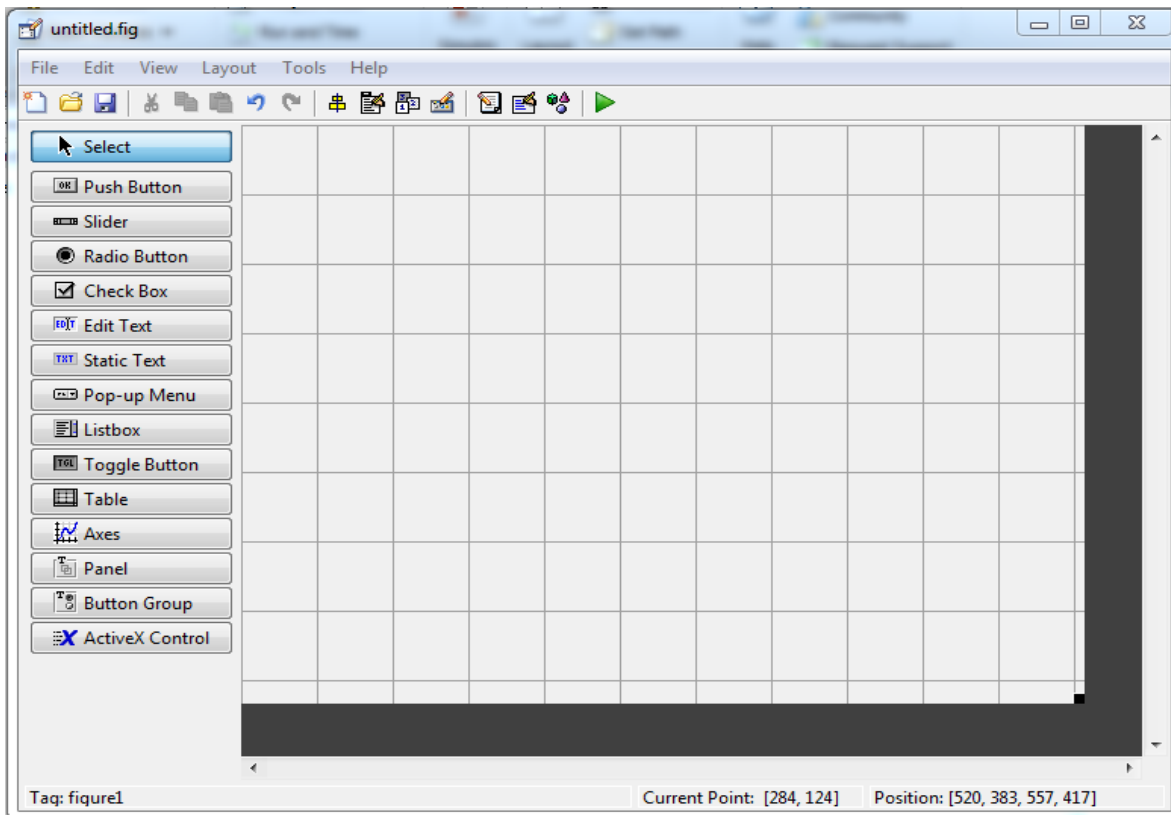


Figura 2.14 Ventana principal del entorno de desarrollo GUIDE

2.3.5 Interfaces gráficas utilizando código (Programmatic GUI)

Para crear una interfaz gráfica de usuario utilizando solamente bloques de código, es necesario hacer uso de los elementos gráficos que proporciona el lenguaje MATLAB. Estos elementos están organizados de manera jerárquica, de tal manera que uno de ellos puede contener a otro.

El objeto gráfico de mayor jerarquía en el desarrollo de interfaces gráficas en MATLAB es el elemento figure, el cual es propiamente una interfaz gráfica en la cual se pueden añadir los demás elementos o controles con los que interactuará el usuario. Para crear un elemento figure basta con escribir dicho comando en un archivo de comandos de MATLAB y modificar sus propiedades de acuerdo a los requerimientos.

Para añadir los controles de usuario (cajas de texto, check box, menú desplegable, etc.) se utiliza la función uicontrol, con la cual se pueden crear todos esos elementos gráficos simplemente pasando como argumentos las propiedades correspondientes para cada caso. En la Tabla 2.3 se muestran los diversos elementos gráficos que pueden crearse con la función uicontrol.

Tabla 2.3 Componentes gráficos creados con la función uicontrol

COMPONENTE	DESCRIPCIÓN
Edit Text	Este componente permite al usuario insertar caracteres en un campo o caja de texto habilitada para edición.
Static Text	Permite mostrar caracteres, pero no editarlos.
Push Button	Es un botón que ejecuta una acción determinada cuando es clicado.
Pop-Up Menu	Menú desplegable en el cual se pueden elegir opciones que conducen a un determinado evento establecido.
Check Box	Permite dos estados, activado o desactivado, que resulta útil en la modificación de propiedades de funciones lógicas.
Radio Button	Agrupado en un panel de botones es útil para permitir la selección de una sola opción.
List Box	Similar a un menú desplegable en funcionamiento, solo que las opciones resultan visibles en todo momento.
Slider	Es una barra deslizable que puede posicionarse en un punto determinado para obtener un valor de un rango preestablecido.
Toogle Button	Mismo funcionamiento que un check box, solo que presenta una apariencia de botón.

Fuente: Marchand, 2003.

Los controles gráficos creados con uicontrol tienen una serie de propiedades que pueden ser modificadas por el usuario que interactúa con ellas, dependiendo del uso para el cual se destine la interfaz gráfica, en la Tabla 2.4 se resumen las propiedades más significativas.

Tabla 2.4 Propiedades de uicontrol

PROPIEDAD	DESCRIPCIÓN
BackgroundColor	Color de fondo del elemento
CData	Matriz insertada por el usuario que modifica el color del elemento gráfico, la cual puede ser una imagen.
Callback	Evento que se produce al interactuar con un elemento, suele ser una función.
FontName	Tipo de fuente utilizada
FontSize	Tamaño de fuente
FontWeight	Estilo de fuente
ForegroundColor	Color de fuente
Position	Posición y tamaño de un elemento en la ventana gráfica o panel al cual pertenece
String	Cadena de caracteres que muestra un elemento asociada a este.
Style	Tipo de control gráfico
Units	Unidades de medida utilizado en el desarrollo de la interfaz gráfica
Tag	Etiqueta que se usa para referenciar los elementos que conforman una estructura de objetos gráficos
Visible	Determina si el elemento es visible al usuario

Fuente: Marchand, 2003.

En la figura 2.15 se muestra el ejemplo de un código para crear una interfaz gráfica utilizando los comandos descritos anteriormente.

```
function guiejemplo
figure('MenuBar','None','NumberTitle','off','Name','Ejemplo',...
      'position',[200 200 180 120],'color',[0.92 0.92 0.94]);

h1 = uicontrol('style','edit','String','',...
              'position',[50 90 80 20]);
h2 = uicontrol('style','text','String','',...
              'position',[50 60 80 20],'backgroundcolor',[0.8 0.8 1]);
uicontrol('style','push','String','Botón',...
          'position',[50 20 80 25],'callback',@mifun);

function mifun(~,~)
    set(h2,'str',get(h1,'str'));
end
end
```

Figura 2.15 Código de desarrollo de una GUI

CAPÍTULO III. METODOLOGÍA

3. Procedimiento y descripción de las actividades realizadas

3.1 Consulta de las fuentes de información

La primera parte del proyecto fue la consulta de las fuentes de información necesarias para sentar las bases sobre las cuales se desarrolló. De manera muy general la búsqueda de información se dividió en dos aspectos: la teoría de las máquinas hidráulicas y el desarrollo de interfaces gráficas de usuario utilizando MATLAB.

Se procedió a clasificar los diversos tipos de máquinas hidráulicas en los grupos que a continuación se enlistan:

- Bombas centrífugas
- Turbinas de reacción
- Turbinas de acción

Con la clasificación anterior, enseguida se extrajeron ejemplos de problemas de diversos libros de mecánica de fluidos y de máquinas hidráulicas, a los cuales se hace referencia en la bibliografía, y se establecieron los grupos de datos de entrada más recurrentes, así como los datos de salida de importancia.

3.2 Desarrollo de las funciones e interfaces gráficas en MATLAB

Inicialmente se trazaron manualmente los prototipos de interfaces gráficas y la distribución de sus diversos elementos para establecer la más adecuada.

Se crearon los íconos que componen el menú de botones asociados a cada interfaz y que tienen como finalidad ofrecer un acceso rápido a las herramientas adicionales que brinda el programa.

Siguiendo las clasificaciones hechas en la primera parte del proyecto, se plantearon y programaron funciones en MATLAB utilizando los datos disponibles como argumentos de entrada. Cada función generalmente se elaboró para resolver problemas que tenían en común la mayoría de sus datos, para con ello minimizar la cantidad de archivos MATLAB que habrían de agruparse en la interfaz final. Para diferenciar cada tipo de problema perteneciente a una misma función se utilizaron números progresivos que indicasen el

subtipo de problema implicado. Evidentemente cada función elaborada dispone de una interfaz gráfica para permitir al usuario una interacción más sencilla.

Además de las interfaces gráficas de cálculo, se crearon funciones adicionales en MATLAB que conforman las diversas herramientas del programa, las cuales son:

- Interfaz de inicio
- Interfaz de selección de datos disponibles
- Interfaz de resultados (datos de salida)
- Interfaz de triángulos de velocidades
- Interfaz de trazo de gráficas
- Interfaz de cálculo de la presión de saturación
- Interfaz de cálculo del coeficiente de pérdidas primarias

3.3 Depuración del programa

La depuración del programa (corrección de errores) se efectuó conforme se elaboraron las funciones correspondientes. Para ello se verificaron los datos de salida proporcionados por el programa con los resultados obtenidos manualmente o bien comparando con los resultados ofrecidos por las diversas fuentes de información consultadas. También se corrigieron durante este proceso las dificultades que implicaba el cambio de trabajar con números escalares en el proceso de cálculo a trabajar con vectores en el proceso del trazado de gráficas.

3.4 Elaboración de herramientas adicionales

En la parte final del proyecto se elaboró un manual de referencia en el cual se documentaron todas las cuestiones referentes al uso del programa, con la descripción de cada una de las interfaces y las consideraciones hechas en los procesos de cálculo que se efectúan. Se añadieron además herramientas adicionales como un conversor de unidades, un archivo con tablas y gráficos relacionados con las máquinas de fluidos incompresibles y una calculadora estándar.

CAPÍTULO IV. RESULTADOS

4. Resultados

La estructuración del programa se basa en las interfaces gráficas que se enlistan a continuación:

- Interfaz de inicio
- Interfaz de selección de datos disponibles
- Interfaz de cálculos
- Interfaz de resultados
- Interfaz de triángulos de velocidades
- Interfaz de trazo de gráficas
- Interfaz de presión de saturación
- Interfaz de coeficiente de pérdidas primarias

Cada una de las anteriores cumple una función específica y tiene una jerarquía de acceso que se describen en los puntos posteriores.

Además de las interfaces anteriores, el programa cuenta con una serie de herramientas adicionales a las cuales se puede acceder a través del menú Herramientas, o bien utilizando la barra de botones ubicada en la parte izquierda de cada una de las interfaces correspondiente.

La documentación correspondiente al uso del programa puede obtenerse accediendo a las secciones de ayuda integradas.

En el Anexo B se resumen la simbología y las unidades de medición utilizadas en los procesos de cálculo ejecutados por el programa.

4.1 Interfaz de inicio

La interfaz de inicio del programa, figura 4.1, es la que se muestra cuando se ejecuta la aplicación y permite seleccionar de un menú desplegable el tipo de máquina hidráulica implicada (bombas centrífugas, turbinas de reacción y turbinas de acción). Además de la posibilidad de acceder a la documentación de ayuda del programa, en la cual se describen todos los componentes, procedimientos y la manera de utilizarlo adecuadamente.



Figura 4.1 Interfaz gráfica de inicio

4.2 Interfaz de selección de datos disponibles

Luego de que se ha seleccionado un tipo de máquina hidráulica, el programa abre una nueva interfaz gráfica como la mostrada en la figura 4.3, en la cual se le indica al programa, mediante un panel que contiene diversas casillas de verificación (check box), los datos disponibles para resolver un determinado problema. Cuando los datos hayan sido marcados en su totalidad se debe proceder a dar clic en el botón que muestra un ícono característico de comprobación, con lo cual el programa ejecuta la orden de verificar si el conjunto de datos seleccionados son suficientes para continuar y abrir una nueva interfaz de cálculo, de lo contrario se mostrará en pantalla un mensaje de advertencia como el que se muestra en la figura 4.2. Si el usuario requiere revisar la notación de forma rápida puede hacerlo desde el botón ubicado en la parte inferior derecha.

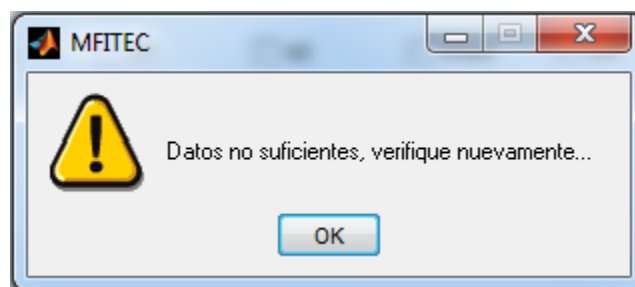


Figura 4.2 Mensaje enviado cuando los datos no son suficientes para continuar con el proceso.

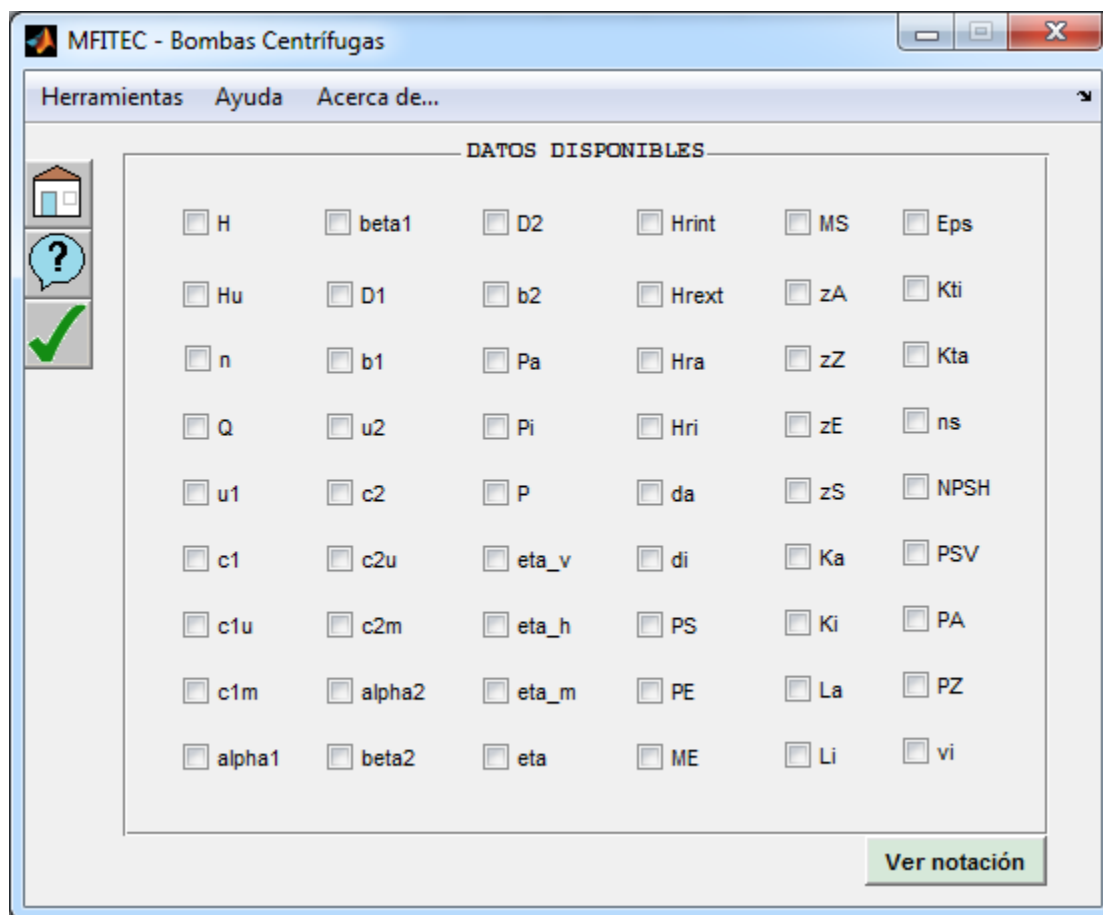


Figura 4.3 Interfaz de selección de datos disponibles

4.3 Interfaz de cálculos

En esta interfaz (Ver figura 4.4) se introducen los datos y se lleva a cabo el proceso de cálculo, y desde donde el usuario puede acceder, a través del menú de botones lateral, a las herramientas de apoyo que le resulten útiles para tal fin. Para introducir los datos han de tomarse en cuenta las unidades de medición para cada magnitud, mismas que están especificadas en la parte derecha de la caja de texto correspondiente a cada una. En las tablas del Anexo B se resumen la notación y unidades utilizadas para las diversas magnitudes.

La interfaz de cálculo integra, como se ha mencionado, un menú de botones y un menú estándar, los cuales contienen las opciones a las cuales el usuario puede acceder para

facilitar los cálculos, tales como tablas, conversor de unidades, calculadora, ecuaciones y otras herramientas.

The screenshot shows the MFITEC-B software window. It has a menu bar with 'Herramientas', 'Ayuda', and 'Acerca de...'. On the left is a vertical toolbar with icons for a home, a triangle with a beta symbol, a graph, a balance scale, a calculator, a plus/minus sign, and a question mark. The main area is titled 'DATOS' and contains several input fields with units: H (m), D1 (m), n (rpm), b1 (m), D2 (m), Q (m³/s), and b2 (m). A 'Calcular' button is located at the bottom right of the input area.

Figura 4.4 Ejemplo de una interfaz de cálculos

The screenshot shows the 'MFITEC - Resultados' window. It displays the title 'DATOS DE SALIDA' in green. Below the title, the following calculated values are listed: beta1 = 46.1893 °, beta2 = 78.5055 °, Pa = 88.29 kW, u1 = 16.9646 m/s, u2 = 18.8496 m/s, c1 = 17.6839 m/s, and c2 = 22.2951 m/s.

Figura 4.5 Interfaz de resultados

4.4 Interfaz de resultados

Esta interfaz, figura 4.5, muestra los datos de salida que se calculan durante el proceso, especificando las unidades en las cuales están dados.

4.5 Interfaz de triángulos de velocidades

En la figura 4.6 se muestra la interfaz gráfica de los triángulos de velocidades de una máquina hidráulica.

La función desarrollada en MATLAB permite trazar los triángulos de velocidades a la entrada y salida, utilizando como argumentos de entrada los valores de la velocidad u del rodete, la velocidad c del fluido y el ángulo α formado entre ambos vectores.

En la interfaz pueden observarse los gráficos de los triángulos a la entrada y salida, y además los valores dados y calculados para las magnitudes características de los triángulos de velocidades.

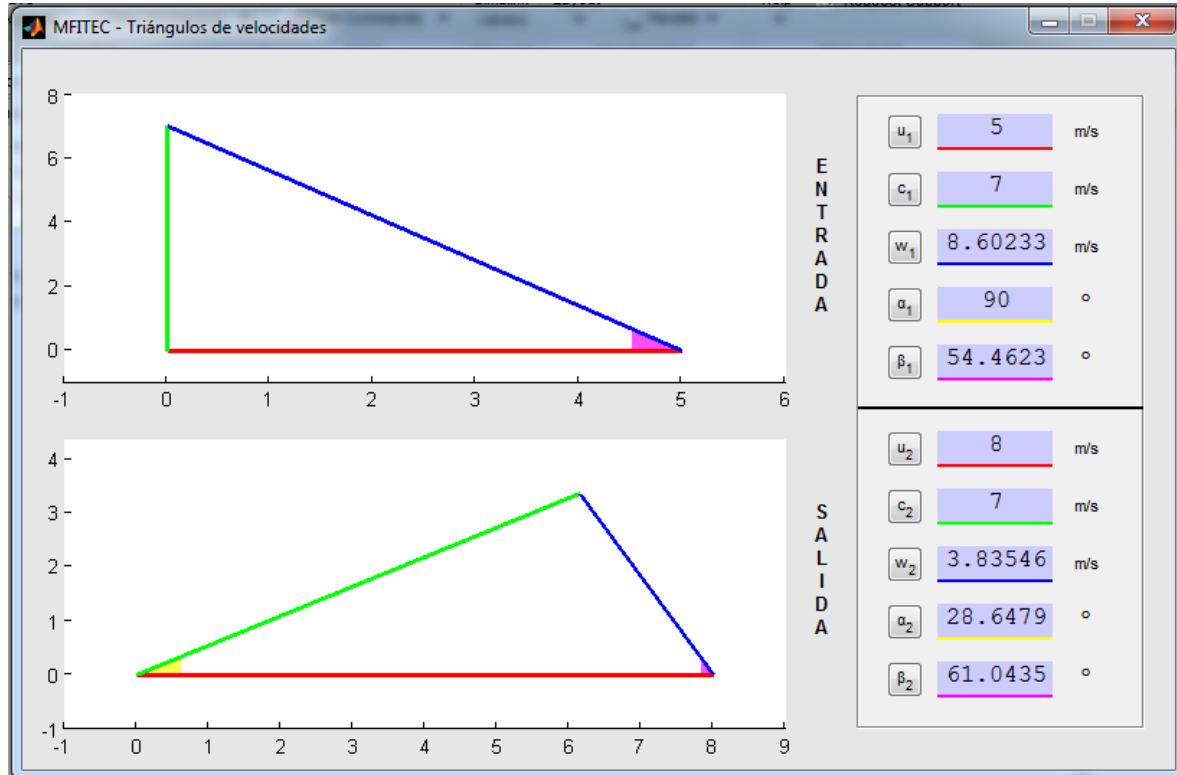


Figura 4.6 Interfaz gráfica de triángulos de velocidades

4.6 Interfaz de trazo de gráficas

La interfaz de trazo gráficas permite graficar en dos dimensiones la variación de un dato de salida respecto a un dato de entrada en un intervalo especificado por el usuario. Incluye la posibilidad de editar la presentación de la gráfica, con modificadores del color, estilo de línea, color de fondo y la activación de la rejilla.

Es posible exportar los datos obtenidos de la gráfica en forma de vectores columnas y guardarlos en un archivo de texto sin formato (*.txt) mediante la interfaz auxiliar que se muestra en la figura 4.8. Así como permite también importar estos datos posteriormente y graficarlos en la interfaz correspondiente.

Además el usuario puede guardar la gráfica como una imagen utilizando la opción correspondiente del menú, el único formato disponible para exportar la gráfica es el PNG (Portable Network Graphics).

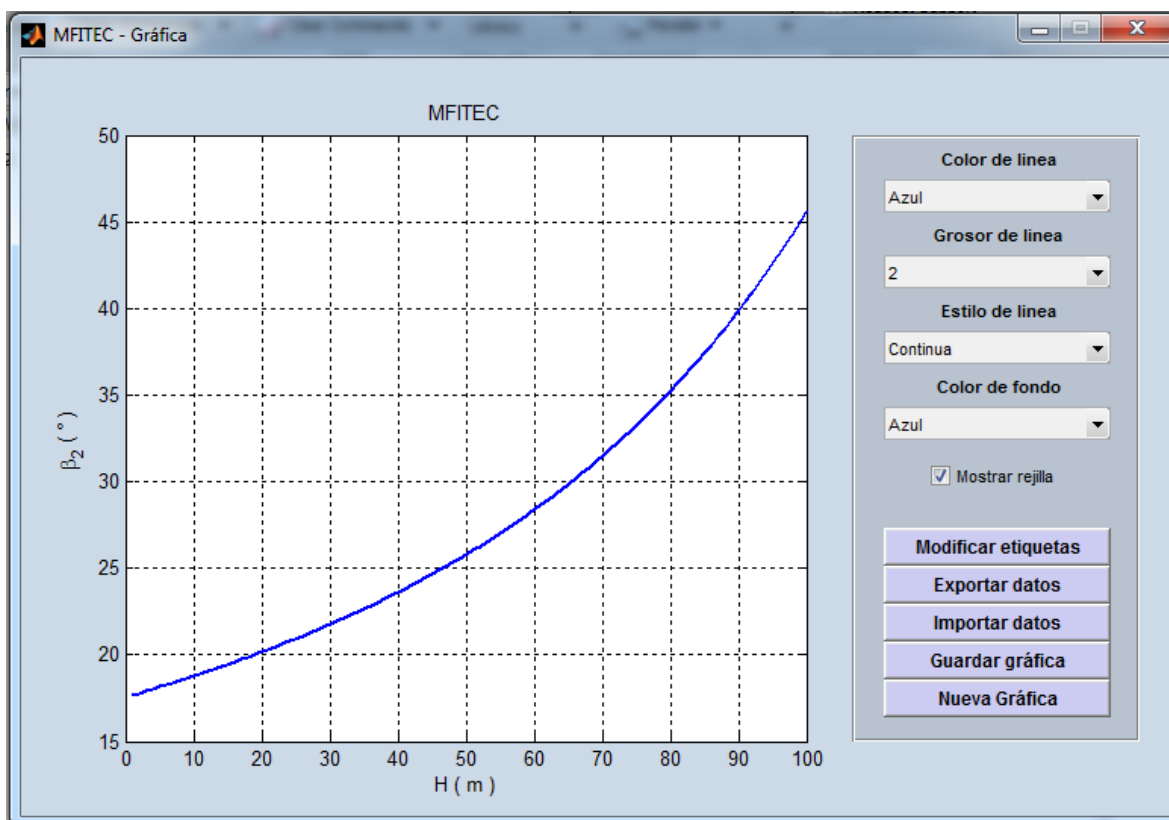
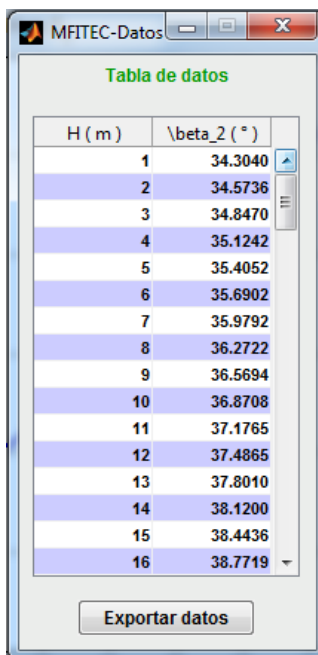


Figura 4.7 Interfaz de trazo de gráficas



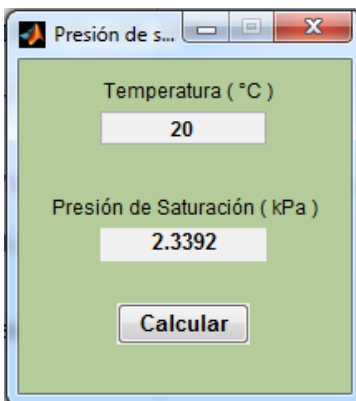
H (m)	\beta_2 (°)
1	34.3040
2	34.5736
3	34.8470
4	35.1242
5	35.4052
6	35.6902
7	35.9792
8	36.2722
9	36.5694
10	36.8708
11	37.1765
12	37.4865
13	37.8010
14	38.1200
15	38.4436
16	38.7719

Exportar datos

Figura 4.8 Ventana de exportar datos.

4.7 Interfaz de presión de saturación

La figura 4.9 muestra la interfaz disponible para calcular la presión de saturación del agua a una determinada temperatura, la temperatura de entrada debe estar dada en °C y la presión de salida estará dada en kPa. El rango abarcado por el programa es de 0.1 a 120°C, lo cual cubre la mayor parte de las posibilidades de cálculo que podrían efectuarse en casos prácticos.



Temperatura (°C)

20

Presión de Saturación (kPa)

2.3392

Calcular

Figura 4.9 Interfaz de presión de saturación.

4.8 Interfaz de coeficiente de pérdidas primarias

Esta interfaz permite calcular el coeficiente de pérdidas primarias utilizando la ecuación de Colebrook-White. La figura 4.10 muestra dicha interfaz.

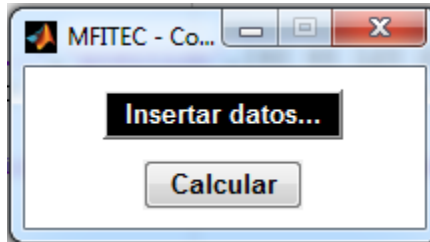


Figura 4.10 Interfaz de cálculo del coeficiente de perdidas primarias.

Para proporcionar los datos de entrada deberá presionarse el botón correspondiente e introducir estos en una nueva ventana gráfica como la de la figura 4.11.

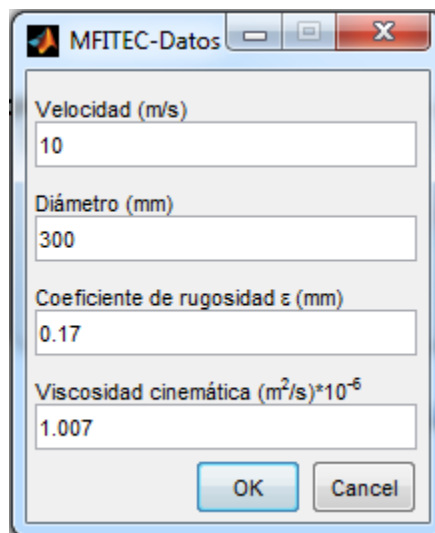


Figura 4.11 Datos para calcular el coeficiente de pérdidas primarias.

4.9 Ejemplo del proceso de resolución de problemas

Ejercicio. Una bomba centrífuga, en la que no se consideran las pérdidas, tiene las siguientes dimensiones:

$$D_1 = 75 \text{ mm}$$

$$D_2 = 300 \text{ mm}$$

$$b_1 = 50 \text{ mm}$$

$$b_2 = 50 \text{ mm}$$

$$\beta_1 = 45^\circ$$

$$\beta_2 = 60^\circ$$

La entrada en los álabes es radial (caso ordinario en las bombas centrífugas). La bomba gira a 500 rpm. El fluido bombeado es agua.

Calcular:

- a) El caudal
- b) La altura que da la bomba
- c) La potencia de accionamiento

Los pasos requeridos para resolver este ejercicio se describen a continuación.

1. Ejecutar la aplicación para acceder a la interfaz de inicio, en esta interfaz seleccionar la opción de bombas centrífugas y pulsar el botón continuar para que el programa le muestre la interfaz de selección de datos disponibles.
2. En la interfaz de datos disponibles seleccionar los datos de entrada (n , D_1 , D_2 , b_1 , b_2 , β_1 , β_2). Una vez se han seleccionado los datos de entrada se debe pulsar el botón de listo para que la aplicación acceda a la interfaz de cálculos (Ver figura 4.12).
3. Proporcionar los datos de entrada en la interfaz de cálculos (Ver figura 4.13) en las unidades de medición que se indican. Pulsar el botón calcular para obtener los resultados o datos de salida calculados por el programa (Ver figura 4.14).
4. Para trazar los triángulos de velocidades debe pulsar el ícono correspondiente y se mostrará la interfaz como en la figura 4.15. Si se requiere graficar un dato de entrada respecto a uno de salida debe pulsar la opción de gráficas y el programa le pedirá un rango para trazarla (Ver figuras 4.16 y 4.17).

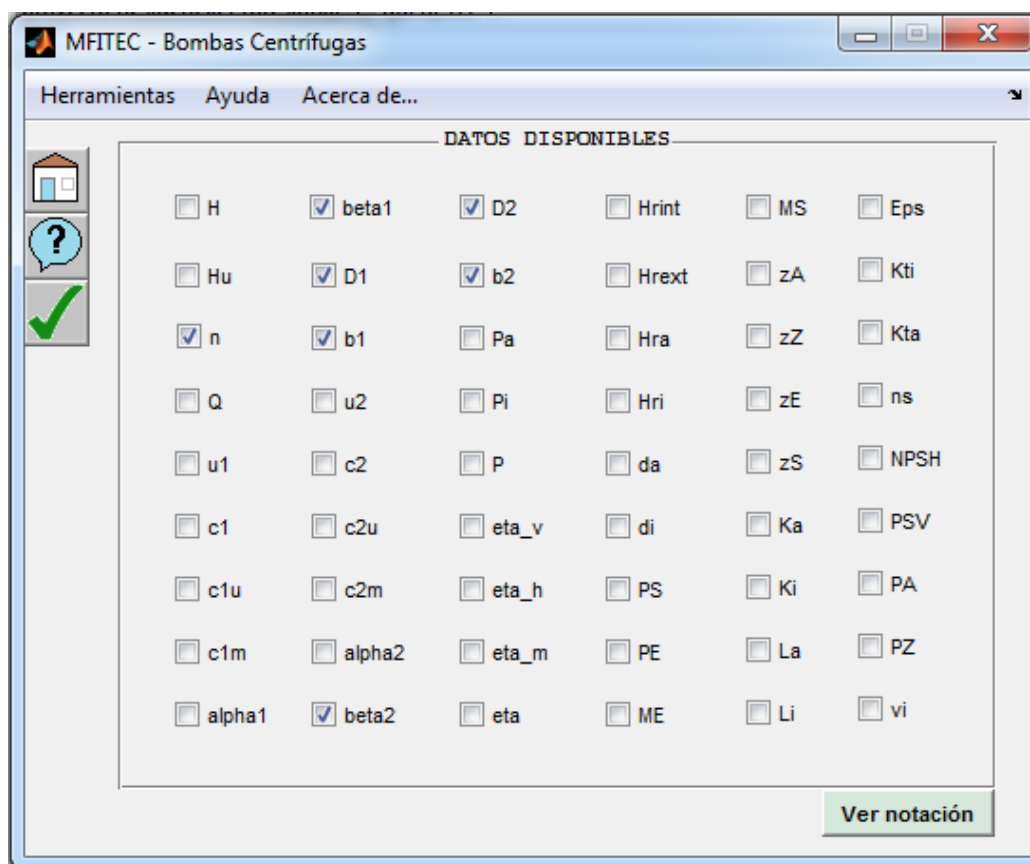


Figura 4.12 Selección de datos disponibles

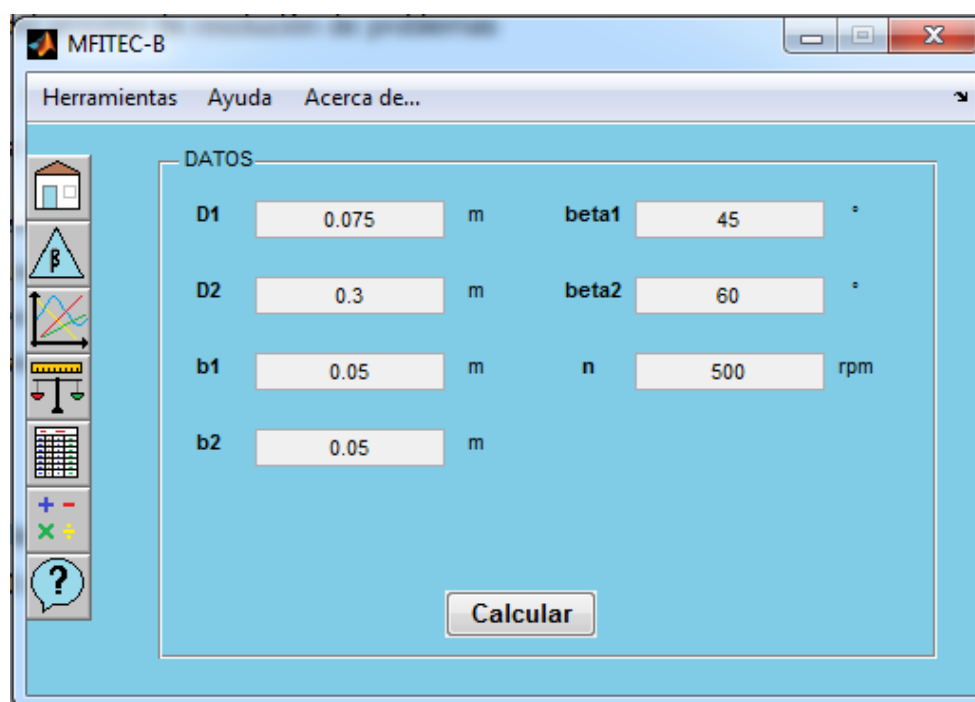


Figura 4.13 Inserción de datos de entrada

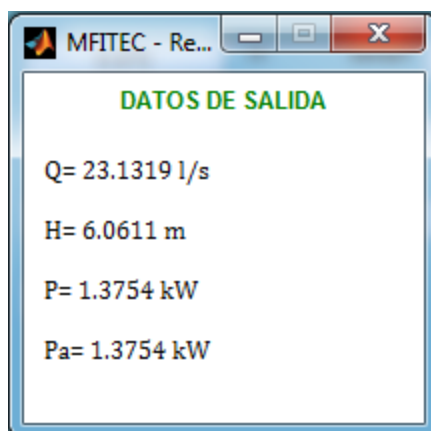


Figura 4.14 Datos de salida

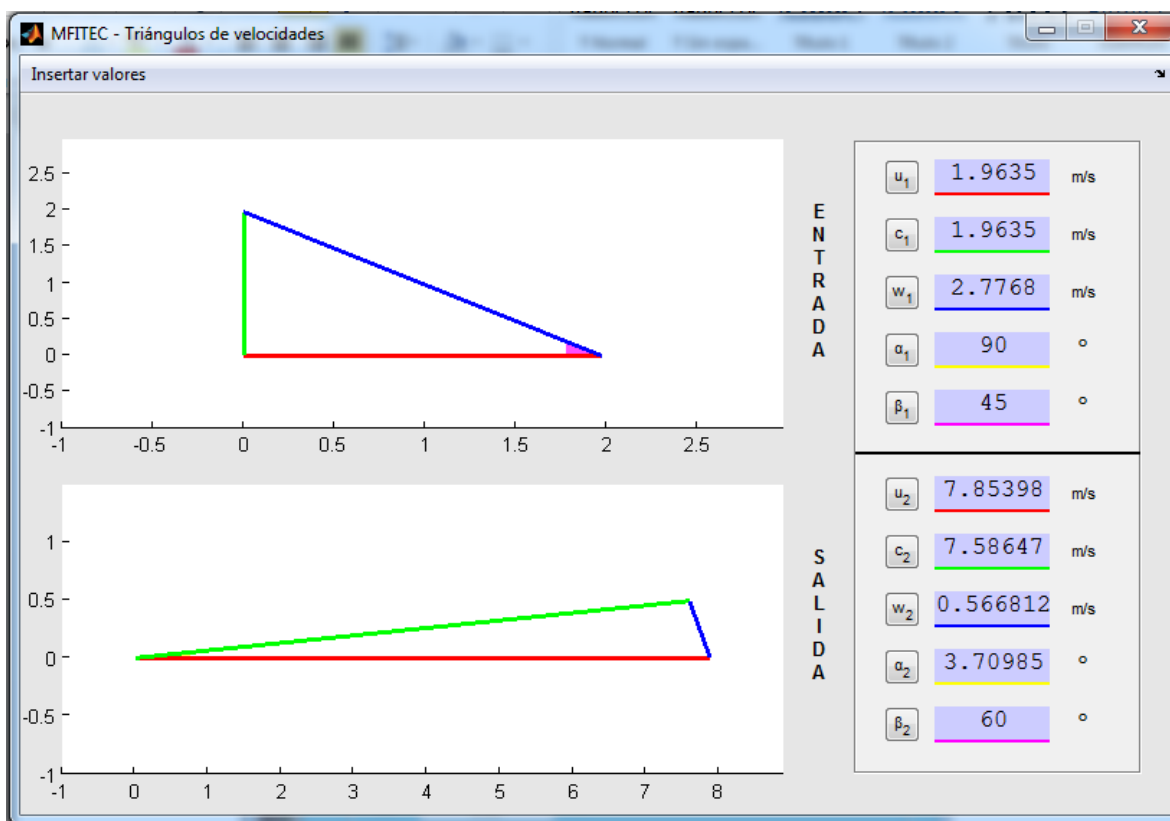


Figura 4.15 Triángulos de velocidades a la entrada y salida.

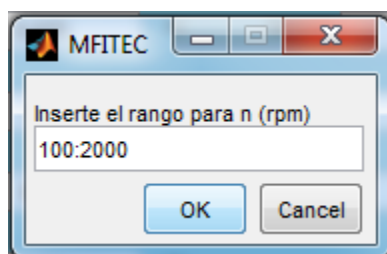


Figura 4.16 Rango de datos para trazar gráfica.

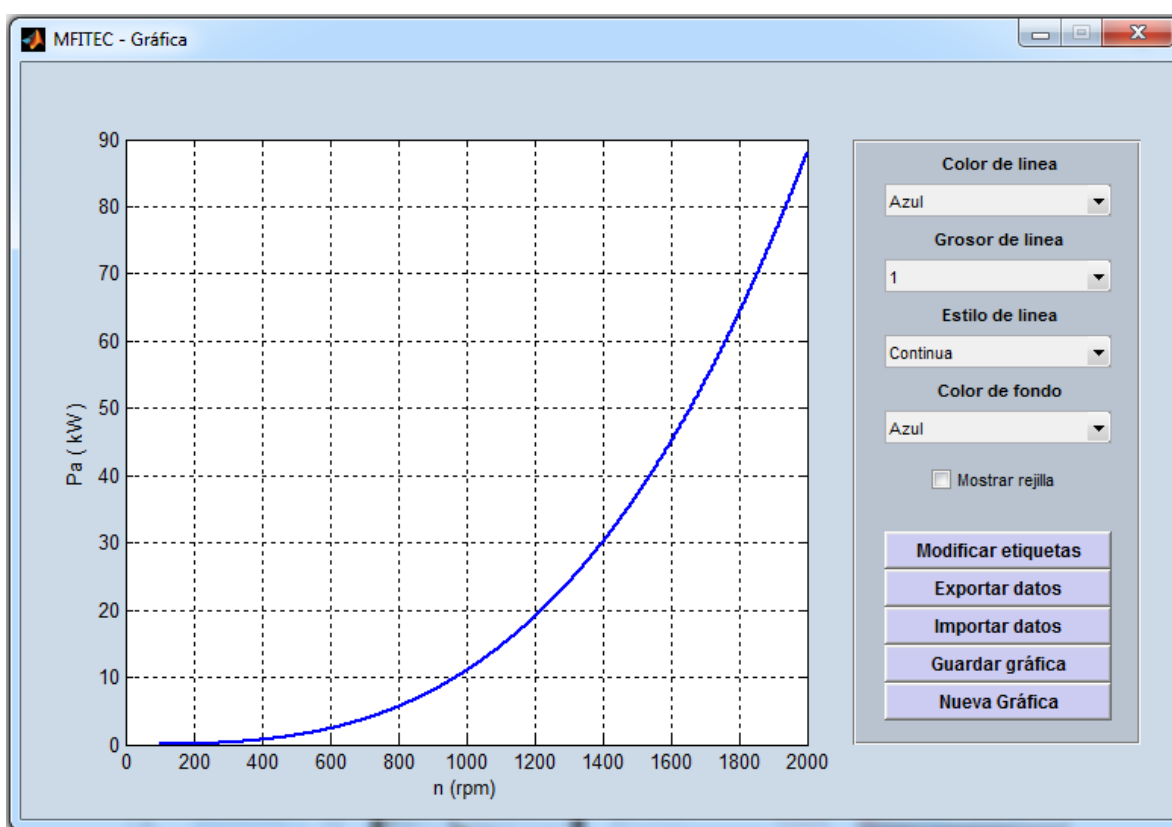


Figura 4.17 Gráfica de ejemplo (rpm – Potencia de accionamiento).

Conclusiones y recomendaciones

El desarrollo del proyecto permitió crear una aplicación capaz de resolver diversos de problemas de máquinas de fluidos incompresibles teniendo como punto de referencia los datos de entrada de los cuales dispone el usuario. Se integraron al programa algunas herramientas que permiten analizar de forma gráfica la importancia de las variables de entrada respecto a un parámetro de salida. Además, se incluyeron aplicaciones adicionales y la documentación correspondiente, tales como una sección de ayuda, tablas y gráficos que posibilitan al usuario interactuar de manera más adecuada con la interfaz brindada por el programa.

La limitación propia de MATLAB para manejar datos simbólicos imposibilitó utilizar un dato de entrada como función de otro, permitiendo solo el uso de datos numéricos, que evidentemente abarca la mayoría de los casos.

Crear un programa utilizando un lenguaje de programación como MATLAB permite crear aplicaciones prácticas en un tiempo reducido, pero implica a la vez dejar en segundo plano aspectos como el rendimiento y la velocidad de ejecución, lo cual pasaría mayor factura si se destinase, por ejemplo, a una aplicación de medición en tiempo real. Por lo cual sería recomendable utilizar un lenguaje de programación más eficaz en esos aspectos, tales como C, C++ o Python, lo cual implicaría desarrollar más herramientas e invertir más tiempo, pero se tendría un programa de mayor calidad. Además, la independencia de ejecución de una aplicación desarrollada en MATLAB tiene sus limitaciones, como el requerimiento de un compilador en tiempo de ejecución sobre el cual se ejecutan las órdenes dadas por el programa.

Fuentes de información

- [1] Hahn, Brian (2007). *Essential MATLAB for engineers and scientists*. 3ra Edición. Butterworth-Heinemann.
- [2] KSB (2010). *Selecting Centrifugal Pumps*. 4ta Edición. KSB Aktiengesellschaft Communications.
- [3] Marchand, Patrick (2003). *Graphics and GUIs with MATLAB*. 3ra Edición. Chapman & Hall/CRC.
- [4] Mataix, Claudio (1982). *Mecánica de fluidos y máquinas hidráulicas*. 2da Edición. Oxford University Press.
- [5] Mott, Robert L. (2006). *Mecánica de fluidos*. 6ta Edición. Prentice Hall.
- [6] Potter, Merle C. y Wiggert, David C. (2007). *Mecánica de fluidos*. 3ra Edición. Internacional Thomson Editores.
- [7] Sotelo, Gilberto (2003). *Hidráulica general, volumen I*. 1ra Edición. Editorial Limusa.
- [8] White, Frank (2008). *Mecánica de fluidos*. 6ta Edición. McGraw Hill.

ANEXOS

ANEXO A. Tablas de notación y unidades de medición utilizadas en la GUI

Para la interfaz de bombas centrífugas

NOTACIÓN	DESCRIPCIÓN	UNIDAD DE MEDICIÓN
Q	Caudal	m ³ /s
H	Altura útil	m
n	Velocidad en rpm	rpm
rho	Densidad del líquido bombeado	kg/m ³
g	Constante de la aceleración de la gravedad	m/s ²
Hu	Altura teórica	m
PA	Presión en el depósito de aspiración	kPa
PZ	Presión en el depósito de impulsión	kPa
PS	Presión a la salida de la bomba	kPa
PE	Presión a la entrada de la bomba	kPa
zA	Cota del depósito de aspiración	m
zZ	Cota del depósito de impulsión	m
zS	Cota de la salida de la bomba	m
zE	Cota de la entrada de la bomba	m
La	Longitud de la tubería de aspiración	m
Li	Longitud de la tubería de impulsión	m
da	Diámetro de la tubería de aspiración	m
di	Diámetro de la tubería de impulsión	m
va	Velocidad en la tubería de aspiración	m/s
vi	Velocidad en la tubería de impulsión	m/s
Hra	Pérdidas en la sección de aspiración	m
Hri	Pérdidas en la sección de impulsión	m
Ka	Coefficiente de pérdidas secundarias en la tubería de aspiración	Adimensional
Ki	Coefficiente de pérdidas secundarias en la tubería de impulsión	Adimensional
Hrext	Pérdidas en el exterior de la bomba	m
Hrint	Pérdidas en el interior de la bomba	m
Pa	Potencia de accionamiento	kW
Pi	Potencia interna	kW
P	Potencia útil	kW
eta	Rendimiento total	Adimensional
eta_h	Rendimiento hidráulico	Adimensional
eta_v	Rendimiento volumétrico	Adimensional
eta_m	Rendimiento mecánico	Adimensional
u	Velocidad de los álabes	m/s
c	Velocidad absoluta del fluido	m/s
cm	Componente meridional de la velocidad del fluido	m/s
cu	Componente periférica de la velocidad del fluido	m/s
w	Velocidad relativa	m/s
alpha	Ángulo formado por c y u	Grados sexagesimales
beta	Ángulo formado w y -u	Grados sexagesimales
NPSH	Altura neta de succión	m

PSV	Presión de saturación del líquido bombeado	kPa
-----	--	-----

Para la interfaz de turbinas de acción

NOTACIÓN	DESCRIPCIÓN	UNIDAD DE MEDICIÓN
Q	Caudal	m ³ /s
n	Velocidad en rpm de la turbina	rpm
H	Altura útil	m
Hu	Altura teórica o de Euler	m
D	Diámetro del rodete	m
d	Diámetro del chorro	m
rho	Densidad del agua	kg/m ³
g	Aceleración de la gravedad	m/s ²
u	Velocidad periférica de la turbina	m/s
c1	Velocidad absoluta del fluido a la entrada	m/s
c2	Velocidad absoluta del fluido a la salida	m/s
w1	Velocidad relativa a la entrada	m/s
w2	Velocidad relativa a la salida	m/s
beta2	Ángulo suplementario de la desviación del chorro	Grados sexagesimales
Ku	Coeficiente de velocidad del rodete	Adimensional
Kc	Coeficiente de velocidad del fluido	Adimensional
Kw	Coeficiente de pérdida por fricción	Adimensional
F	Fuerza transmitida por el fluido al rodete	N
M	Momento transmitido por el fluido al rodete	N*m
Hrint	Pérdidas en la sección comprendida entre la entrada y salida de la turbina	m
Hrext	Pérdidas en las secciones anterior y posterior a la turbina	m
Hb	Altura bruta	m
P	Potencia absorbida, Potencia neta.	kW
Pi	Potencia interna / Potencia impartida del fluido al rodete	kW
Pa	Potencia de accionamiento	kW
eta	Rendimiento total	Adimensional
eta_h	Rendimiento hidráulico	Adimensional
eta_v	Rendimiento volumétrico	Adimensional
eta_m	Rendimiento mecánico	Adimensional

Para la interfaz de turbinas de reacción

NOTACIÓN	DESCRIPCIÓN	UNIDAD DE MEDICIÓN
Q	Caudal	m ³ /s
H	Altura útil	m
Hb	Altura bruta	m
n	Velocidad en rpm	rpm
rho	Densidad del líquido bombeado	kg/m ³
g	Constante de la aceleración de la gravedad	m/s ²
Hu	Altura teórica	m
PA	Presión aguas arriba	kPa
PZ	Presión aguas abajo	kPa
PS	Presión a la salida de la turbina	kPa
PE	Presión a la salida de la turbina	kPa
zA	Cota de aguas arriba	m
zZ	Cota de aguas abajo	m
zS	Cota de la salida de la turbina	m
zE	Cota de la entrada de la turbina	m
fr	Coeficiente de pérdidas primarias	Adimensional
L	Longitud de la tubería de aspiración	m
dt	Diámetro de la tubería de aspiración	m
vE	Velocidad en la entrada de la turbina	m/s
vS	Velocidad en la salida de la turbina	m/s
Hra	Pérdidas en la sección de aspiración	m
Hri	Pérdidas en la sección de impulsión	m
Hrext	Pérdidas en el exterior de la turbina	m
Hrint	Pérdidas en el interior de la turbina	m
Prm	Pérdidas mecánicas	m
Pa	Potencia de útil	kW
Pi	Potencia interna	kW
P	Potencia teórica	kW
eta	Rendimiento total	Adimensional
eta_h	Rendimiento hidráulico	Adimensional
eta_v	Rendimiento volumétrico	Adimensional
eta_m	Rendimiento mecánico	Adimensional
u	Velocidad de los álabes	m/s
c	Velocidad absoluta del fluido	m/s
cm	Componente meridional de la velocidad del fluido	m/s
cu	Componente periférica de la velocidad del fluido	m/s
w	Velocidad relativa	m/s
alpha	Ángulo formado por c y u	Grados sexagesimales
beta	Ángulo formado w y -u	Grados sexagesimales

Nota: Para las velocidades del rodete, 1 indica velocidades a la entrada y 2 velocidades a la salida.

ANEXO B. Códigos de las interfaces gráficas

Interfaz de inicio

```
function Inicio

%% Condiciones de inicio del programa
rho=1000; % Establecer valor de la densidad en 1000 kg/m^3
save('data/mfun/rho.mat','rho');
clearvars('rho');

%% Desarrollo de la interfaz
f = figure('Visible','on','MenuBar','none','resize','off','NumberTitle','off'...
    , 'Name','MFITEC','position',[500,300,300,200],'color',[0.8 0.8 0.9]);

% Icono de la aplicación
warning('off','MATLAB:HandleGraphics:ObsoletedProperty:JavaFrame');
jframe=get(f,'javaframe');
jIcon=javax.swing.ImageIcon('data/imag/triangulo.png');
jframe.setFigureIcon(jIcon);

ax=axes('Parent',f,'units','normalized','position',[0 0 1 1]);
imshow(imread('data/imag/imagen.png'),'parent',ax);

uicontrol('style','text','String','Para continuar seleccione una opción',...
    'position',[30 120 240 20],'fontsize',9,'fontweight','bold',...
    'backgroundcolor',[0.82 0.82 0.82]);

sel=uicontrol('style','popupmenu','String',{'Bombas Centrífugas',...
    'Turbinas de Reacción','Turbinas de Acción'},'position',[70 70 160 30]);

uicontrol('style','pushbutton','String','Continuar >>','position',...
    [100 30 100 30],'backgroundcolor',[0.5 1 0.5],'fontsize',10,...
    'callback',@opcion,'fontweight','bold','tooltipstring',...
    'Continuar a la interfaz de cálculos','FontName','Arial');

%% Botón ayuda
posx=(300/2)-35/2;
posy=200-35;
uicontrol('style','pushbutton','String','', 'position',[posx posy 35 35],...
    'fontsize',10,'fontweight','bold','backgroundcolor',[0.7 0.8 0.8],...
    'cdata',imread('data/imag/ayuda.png'),'tooltipstring','Ayuda',...
    'callback',@goayuda);

path(path,'data/mfun'); % Agregar al path las funciones de "mfun"

%% Funciones
function opcion(~,~)
    n=get(sel,'Value');
    K=struct2array(load('data/mfun/mostrec.mat'));
    switch n
        case 1
            close(gcf);
            run('App_Bombas.m');
            if K==0
                run('data/mfun/recomendaciones/recomendacion(1).m');
            end
        case 2
```

```

        close(gcf);
        run('App_Turbinas_R.m');
    case 3
        close(gcf);
        run('App_Turbinas_A.m');
    end
end

function goayuda(~,~)
    winopen('data/docs/Ayuda.pdf');
end
end

```

Interfaz de cálculos

```

function b_2
%% Descripción General
% REF 1937
% MOD 12/11/13

%% Constantes persistentes
persistent g rho
g=9.81;
rho=struct2array(load('data/mfun/rho.mat'));

%% Características generales

cfondo=[0.5 0.8 0.9];
etsize=8;

%% Interfaz
f = figure('Visible','on','MenuBar','none','resize','off','NumberTitle','off'...
    , 'Name','MFITEC-B','position',[500,300,500,300],'color',cfondo);
centerfig(f);

menu1 = uimenu(f,'Label','Herramientas');
uimenu(menu1,'Label','Conversor de unidades','callback',...
    'winopen('data/mfun/convert.exe')');
uimenu(menu1,'Label','Tablas','callback',...
    'winopen('data/docs/Tablas.pdf')');
uimenu(menu1,'Label','Diagrama de Moody','callback',...
    'winopen('data/docs/Moody.pdf')');
uimenu(menu1,'Label','Presión de saturación del agua (PSV)',...
    'callback','run('data/mfun/presat.m')');
sub11=uimenu(menu1,'Label','Ecuaciones de turbomáquinas');
uimenu(sub11,'Label','Ecuaciones Bombas Centrífugas','callback',...
    'winopen('data/docs/EC_BC.pdf')');
uimenu(sub11,'Label','Ecuaciones Turbinas de reacción','callback',...
    'winopen('data/docs/EC_TR.pdf')');
uimenu(sub11,'Label','Ecuaciones Turbinas de acción','callback',...
    'winopen('data/docs/EC_TA.pdf')');
menu2=uimenu(f,'Label','Ayuda');
uimenu(menu2,'Label','Ayuda en PDF','callback',...
    'winopen('data/docs/Ayuda.pdf')');
uimenu(menu2,'Label','Video Tutorial','callback',...
    'winopen('data/docs/vid01.mp4')');
uimenu(f,'Label','Acerca de...','callback',...
    'run('data/mfun/acerca.m')');

```

```

pp=uipanel(f,'title','DATOS','units','pixels','position',[80 20 400 270],...
'backgroundcolor',cfondo);

eH=uicontrol(pp,'style','text','String','H','position',[10 210 30 20],...
'backgroundcolor',cfondo,'fontsize',etsize,'fontweight','bold');
dH=uicontrol(pp,'style','edit','String','', 'position',[50 210 100 20]);
uH=uicontrol(pp,'style','text','String','m','position',[150 210 30 20],...
'backgroundcolor',cfondo);

en=uicontrol(pp,'style','text','String','n','position',[10 170 30 20],...
'backgroundcolor',cfondo,'fontsize',etsize,'fontweight','bold');
dn=uicontrol(pp,'style','edit','String','', 'position',[50 170 100 20]);
un=uicontrol(pp,'style','text','String','rpm','position',[150 170 30 20],...
'backgroundcolor',cfondo);

eD2=uicontrol(pp,'style','text','String','D2','position',[10 130 30 20],...
'backgroundcolor',cfondo,'fontsize',etsize,'fontweight','bold');
dD2=uicontrol(pp,'style','edit','String','', 'position',[50 130 100 20]);
uD2=uicontrol(pp,'style','text','String','m','position',[150 130 30 20],...
'backgroundcolor',cfondo);

eb2=uicontrol(pp,'style','text','String','b2','position',[10 90 30 20],...
'backgroundcolor',cfondo,'fontsize',etsize,'fontweight','bold');
db2=uicontrol(pp,'style','edit','String','', 'position',[50 90 100 20]);
ub2=uicontrol(pp,'style','text','String','m','position',[150 90 30 20],...
'backgroundcolor',cfondo);

eD1=uicontrol(pp,'style','text','String','D1','position',[210 210 30 20],...
'backgroundcolor',cfondo,'fontsize',etsize,'fontweight','bold');
dD1=uicontrol(pp,'style','edit','String','', 'position',[250 210 100 20]);
uD1=uicontrol(pp,'style','text','String','m','position',[350 210 30 20],...
'backgroundcolor',cfondo);

eb1=uicontrol(pp,'style','text','String','b1','position',[210 170 30 20],...
'backgroundcolor',cfondo,'fontsize',etsize,'fontweight','bold');
db1=uicontrol(pp,'style','edit','String','', 'position',[250 170 100 20]);
ub1=uicontrol(pp,'style','text','String','m','position',[350 170 30 20],...
'backgroundcolor',cfondo);

eQ=uicontrol(pp,'style','text','String','Q','position',[210 130 30 20],...
'backgroundcolor',cfondo,'fontsize',etsize,'fontweight','bold');
dQ=uicontrol(pp,'style','edit','String','', 'position',[250 130 100 20]);
uQ=uicontrol(pp,'style','text','String','m^3/s','position',[350 130 30 20],...
'backgroundcolor',cfondo);

boton=uicontrol(pp,'style','pushbutton','String','Calcular','position',...
[150 20 80 25],'callback',@calcular,'fontsize',10,'fontweight','bold');

%% Menú Botones
posb=250;

uicontrol('style','pushbutton','String','', 'position',[0 posb 35 35],...
'callback',@gohome,'fontsize',10,'fontweight','bold','backgroundcolor',...
[0.7 0.8 0.8],'cdata',imread('data/imag/inicio.png'));

uicontrol('style','pushbutton','String','', 'position',[0 posb-35 35 35],...
'callback',@gotrian,'fontsize',10,'fontweight','bold','backgroundcolor',...
[0.7 0.8 0.8],'cdata',imread('data/imag/triangulo.png'));

uicontrol('style','pushbutton','String','', 'position',[0 posb-2*35 35 35],...
'callback',@gographics,'fontsize',10,'fontweight','bold','backgroundcolor',...

```

```
[0.7 0.8 0.8], 'cdata', imread('data/imag/grafica.png'));

uicontrol('style','pushbutton','String','', 'position',[0 posb-3*35 35 35],...
'callback',@goconv, 'fontsize',10, 'fontweight','bold', 'backgroundcolor',...
[0.7 0.8 0.8], 'cdata', imread('data/imag/unidades.png'));

uicontrol('style','pushbutton','String','', 'position',[0 posb-4*35 35 35],...
'callback',@gotablas, 'fontsize',10, 'fontweight','bold', 'backgroundcolor',...
[0.7 0.8 0.8], 'cdata', imread('data/imag/tabla.png'));

uicontrol('style','pushbutton','String','', 'position',[0 posb-5*35 35 35],...
'callback',@gocalc, 'fontsize',10, 'fontweight','bold', 'backgroundcolor',...
[0.7 0.8 0.8], 'cdata', imread('data/imag/calculadora.png'));

uicontrol('style','pushbutton','String','', 'position',[0 posb-6*35 35 35],...
'callback',@goayuda, 'fontsize',10, 'fontweight','bold', 'backgroundcolor',...
[0.7 0.8 0.8], 'cdata', imread('data/imag/ayuda.png'));

%% Funciones

function calcular(src,event)
    global alpha1 alpha2 c1 c2 u1 u2
    MV=get(findobj('style','edit'),'String');
    if any(strcmp(MV,'')==1)
        warndlg({'Algunos datos no han sido proporcionados',...
        'Por favor verifique...'}, 'DATOS INCOMPLETOS');
    else
        H=str2double(get(dH, 'String'));
        n=str2double(get(dn, 'String'));
        D2=str2double(get(dD2, 'String'));
        b2=str2double(get(db2, 'String'));
        D1=str2double(get(dD1, 'String'));
        b1=str2double(get(db1, 'String'));
        Q=str2double(get(dQ, 'String'));

        alpha1=90*pi/180;
        u1=(pi*D1*n)/60;
        u2=(pi*D2*n)/60;
        Hu=H;
        c2u=(Hu*g)/u2;
        c2m=Q/(pi*b2*D2);
        c1m=Q/(pi*b1*D1);
        c1=c1m;
        c2=sqrt(c2u^2+c2m^2);
        beta1=atan(c1/u1);
        beta2=atan(c2m/(u2-c2u));
        alpha2=atan(c2m/c2u);
        Pa=Q*rho*g*H;

        res1=horzcat('beta1= ', num2str(beta1*180/pi), ' °');
        res2=horzcat('beta2= ', num2str(beta2*180/pi), ' °');
        res3=horzcat('Pa = ', num2str(Pa/10^3), ' kW');
        res4=horzcat('u1 = ', num2str(u1), ' m/s');
        res5=horzcat('u2 = ', num2str(u2), ' m/s');
        res6=horzcat('c1 = ', num2str(c1), ' m/s');
        res7=horzcat('c2 = ', num2str(c2), ' m/s');

        figresultados([res1, '', res2, '', res3, '', res4, '', res5, '', res6, ...
        '', res7],8);
    end
end

function gohome(src,event)
```

```

        close(gcf);
        run('App_Bombas.m');
    end

    function gotrian(src,event)
        global alpha1 alpha2 c1 c2 u1 u2
        triangulos(u1,c1,alpha1,u2,c2,alpha2);
    end

    function gographics(~,~)
        H=insertRango('H (m)', '1:100');
        n=str2double(get(dn, 'String'));
        D2=str2double(get(dD2, 'String'));
        b2=str2double(get(db2, 'String'));
        D1=str2double(get(dD1, 'String'));
        b1=str2double(get(db1, 'String'));
        Q=str2double(get(dQ, 'String'));

        u1=(pi*D1*n)/60;
        u2=(pi*D2*n)/60;
        Hu=H;
        c2u=(Hu*g)/u2;
        c2m=Q/(pi*b2*D2);
        c1m=Q/(pi*b1*D1);
        c1=c1m;
        c2=sqrt(c2u.^2+c2m^2);
        beta1=atan(c1/u1);
        beta2=atan(c2m./(u2-c2u));
        Pa=Q*rho*g*H;
        grafica(H,beta2*180/pi, 'H ( m )', '\beta_2 ( ^\circ )', 'H vs \beta_2');

    end

    function goconv(~,~)
        winopen('data/mfun/convert.exe');
    end

    function gotablas(~,~)
        winopen('data/docs/Tablas.pdf');
    end

    function gocalc(~,~)
        winopen('data/mfun/calc.exe');
    end

    function goayuda(~,~)
        winopen('data/docs/Ayuda.pdf');
    end

end

```

Interfaz de resultados

```

function figresultados(resultados,nl)
    cfondo=[1 1 1];
    f = figure('Visible','on','MenuBar','none','resize','off','NumberTitle',...
        'off','Name','MFITEC - Resultados','position',[100,100,200,75+nl*25],...
        'color',cfondo);
    centerfig(f);

```



```

uicontrol('style','text','String','DATOS DE SALIDA','position',...
[10 75+(nl*25)-25 180 20],'fontweight','bold','fontsize',9,...
'foregroundcolor',[0 0.5 0],'backgroundcolor',cfondo);

uicontrol('style','text','String',resultados,'position',...
[10 10 180 25+nl*25],'fontsize',10,'fontweight','normal',...
'HorizontalAlignment','left','fontname','Cambria',...
'backgroundcolor',cfondo);

end

```

Interfaz de triángulos de velocidades

```

function triangulos(u1,c1,alpha1,u2,c2,alpha2)
% Función para mostrar el diagrama de velocidades de una máquina hidráulica
% Argumentos de entrada: Los que se indican
% Mod 01/10/13
if nargin<1
    valores();
elseif isempty([u1,u2,c1,c2,alpha1,alpha2])
    errordlg('Los datos de entrada no han sido proporcionados','MFITEC');
else

    cfondo=[0.9 0.9 0.9];
    f=figure('NumberTitle','off','Name','MFITEC - Triángulos de velocidades',...
        'MenuBar','None','color',cfondo,'position',[200,200,800,500],...
        'resize','off');
    centerfig(f);

    %% Menú
    uimenu(f,'label','Insertar valores','callback',@valores);

    axS=axes('parent',f,'units','pixels','position',[30 30 500 200]);
    axE=axes('parent',f,'units','pixels','position',[30 270 500 200]);
    pp=uipanel(f,'units','pixels','position',[580 30 200 440]);

    uicontrol('style','text','String',{' ',' ','E','N','T','R','A','D','A'},...
        'position',[540 280 30 180],'backgroundcolor',cfondo,'fontsize',...
        10,'fontweight','bold');
    uicontrol('style','text','String',{' ',' ','S','A','L','I','D','A'},...
        'position',[540 40 30 180],'backgroundcolor',cfondo,'fontsize',...
        10,'fontweight','bold');
    uicontrol(pp,'style','text','text','position',[0 220 198 2],...
        'backgroundcolor','k');

    % Entrada Panel
    uicontrol(pp,'style','pushbutton','String','<html>u<sub>1</sub></html>',...
        'position',[20 400 25 25]);
    dul=uicontrol(pp,'style','text','String',' ','position',[55 400 80 25],...
        'backgroundcolor',[0.8 0.8 1],'fontsize',13,'fontname','courier');
    uicontrol(pp,'style','text','String',' ','position',[55 400 80 2],...
        'backgroundcolor','r');
    uicontrol(pp,'style','text','String','m/s','position',[145 400 25 20]);

    uicontrol(pp,'style','pushbutton','String','<html>c<sub>1</sub></html>',...
        'position',[20 360 25 25]);
    dcl=uicontrol(pp,'style','text','String',' ','position',[55 360 80 25],...
        'backgroundcolor',[0.8 0.8 1],'fontsize',13,'fontname','courier');
    uicontrol(pp,'style','text','String',' ','position',[55 360 80 2],...
        'backgroundcolor','g');
    uicontrol(pp,'style','text','String','m/s','position',[145 360 25 20]);

```

```

uicontrol(pp,'style','pushbutton','String','<html>w<sub>1</sub></html>',...
    'position',[20 320 25 25]);
dw1=uicontrol(pp,'style','text','String','', 'position',[55 320 80 25],...
    'backgroundcolor',[0.8 0.8 1], 'fontsize',13, 'fontname','courier');
uicontrol(pp,'style','text','String','', 'position',[55 320 80 2],...
    'backgroundcolor','b');
uicontrol(pp,'style','text','String','m/s', 'position',[145 320 25 20]);

uicontrol(pp,'style','pushbutton','String','<html>&alpha<sub>1</sub></html>',...
    'position',[20 280 25 25]);
dalpha1=uicontrol(pp,'style','text','String','', 'position',[55 280 80 25],...
    'backgroundcolor',[0.8 0.8 1], 'fontsize',13, 'fontname','courier');
uicontrol(pp,'style','text','String','', 'position',[55 280 80 2],...
    'backgroundcolor','y');
uicontrol(pp,'style','text','String','°', 'position',[145 285 25 20],...
    'fontsize',14);

uicontrol(pp,'style','pushbutton','String','<html>&beta<sub>1</sub></html>',...
    'position',[20 240 25 25]);
dbeta1=uicontrol(pp,'style','text','String','', 'position',[55 240 80 25],...
    'backgroundcolor',[0.8 0.8 1], 'fontsize',13, 'fontname','courier');
uicontrol(pp,'style','text','String','', 'position',[55 240 80 2],...
    'backgroundcolor','m');
uicontrol(pp,'style','text','String','°', 'position',[145 245 25 20],...
    'fontsize',14);

% Salida Panel
uicontrol(pp,'style','pushbutton','String','<html>u<sub>2</sub></html>',...
    'position',[20 180 25 25]);
du2=uicontrol(pp,'style','text','String','', 'position',[55 180 80 25],...
    'backgroundcolor',[0.8 0.8 1], 'fontsize',13, 'fontname','courier');
uicontrol(pp,'style','text','String','', 'position',[55 180 80 2],...
    'backgroundcolor','r');
uicontrol(pp,'style','text','String','m/s', 'position',[145 180 25 20]);

uicontrol(pp,'style','pushbutton','String','<html>c<sub>2</sub></html>',...
    'position',[20 140 25 25]);
dc2=uicontrol(pp,'style','text','String','', 'position',[55 140 80 25],...
    'backgroundcolor',[0.8 0.8 1], 'fontsize',13, 'fontname','courier');
uicontrol(pp,'style','text','String','', 'position',[55 140 80 2],...
    'backgroundcolor','g');
uicontrol(pp,'style','text','String','m/s', 'position',[145 140 25 20]);

uicontrol(pp,'style','pushbutton','String','<html>w<sub>2</sub></html>',...
    'position',[20 100 25 25]);
dw2=uicontrol(pp,'style','text','String','', 'position',[55 100 80 25],...
    'backgroundcolor',[0.8 0.8 1], 'fontsize',13, 'fontname','courier');
uicontrol(pp,'style','text','String','', 'position',[55 100 80 2],...
    'backgroundcolor','b');
uicontrol(pp,'style','text','String','m/s', 'position',[145 100 25 20]);

uicontrol(pp,'style','pushbutton','String','<html>&alpha<sub>2</sub></html>',...
    'position',[20 60 25 25]);
dalpha2=uicontrol(pp,'style','text','String','', 'position',[55 60 80 25],...
    'backgroundcolor',[0.8 0.8 1], 'fontsize',13, 'fontname','courier');
uicontrol(pp,'style','text','String','', 'position',[55 60 80 2],...
    'backgroundcolor','y');
uicontrol(pp,'style','text','String','°', 'position',[145 65 25 20],...
    'fontsize',14);

```

```

uicontrol(pp,'style','pushbutton','String','<html>&beta;<sub>2</sub></html>',...
    'position',[20 20 25 25]);
dbeta2=uicontrol(pp,'style','text','String','', 'position',[55 20 80 25],...
    'backgroundcolor',[0.8 0.8 1], 'fontsize',13, 'fontname','courier');
uicontrol(pp,'style','text','String','', 'position',[55 20 80 2],...
    'backgroundcolor','m');
uicontrol(pp,'style','text','String','', 'position',[145 25 25 20],...
    'fontsize',14);

%% Entrada
ux1=[0 u1];
uy1=[0 0];
cx1=[0 c1*cos(alpha1)];
cy1=[0 c1*sin(alpha1)];
wx1=[0 cx1-ux1]+u1;
wy1=[0 cy1-uy1];
w1=norm([cx1-ux1,cy1-uy1]);
if norm([ux1,uy1])>(c1*cos(alpha1))
    beta1=asin((c1*sin(alpha1))/w1);
else
    beta1=pi-asin(c1*sin(alpha1)/w1);
end
axes(axE);
hold on
h1=plot(ux1,uy1,'r',cx1,cy1,'g',wx1,wy1,'b');
xlim([-1 max([ux1,cx1,wx1])+1]);
ylim([-1 max([uy1,cy1,wy1])+1]);
set(h1,'linewidth',2);
if norm([ux1,uy1])>(c1*cos(alpha1))
    Max1=[0 0.1*c1*cos(alpha1) 0.1*c1*cos(alpha1)];
    May1=[0 0.1*c1*sin(alpha1) 0];
    Mbx1=[u1,u1-0.1*w1*cos(beta1),u1-0.1*w1*cos(beta1)];
    Mby1=[0,0,0.1*w1*sin(beta1)];
    patch(Max1,May1,'y','FaceAlpha',0.7,'EdgeColor','none');
    patch(Mbx1,Mby1,'m','FaceAlpha',0.7,'EdgeColor','none');
else
    Max1=[0 0.1*c1*cos(alpha1) 0.1*c1*cos(alpha1)];
    May1=[0 0.1*c1*sin(alpha1) 0];
    Mbx1=[u1,u1-0.1*u1,u1-0.1*w1*cos(beta1)];
    Mby1=[0,0,0.1*w1*sin(beta1)];
    patch(Max1,May1,'y','FaceAlpha',0.7,'EdgeColor','none');
    patch(Mbx1,Mby1,'m','FaceAlpha',0.7,'EdgeColor','none');
end
%% Salida
ux2=[0 u2];
uy2=[0 0];
cx2=[0 c2*cos(alpha2)];
cy2=[0 c2*sin(alpha2)];
wx2=[0 cx2-ux2]+u2;
wy2=[0 cy2-uy2];
w2=norm([cx2-ux2,cy2-uy2]);
if norm([ux2,uy2])>(c2*cos(alpha2))
    beta2=asin((c2*sin(alpha2))/w2);
else
    beta2=pi-asin(c2*sin(alpha2)/w2);
end
axes(axS);
hold on
h2=plot(ux2,uy2,'r',cx2,cy2,'g',wx2,wy2,'b');
xlim([-1 max([ux2,cx2,wx2])+1]);

```

```

ylim([-1 max([uy2,cy2,wy2])+1]);
set(h2,'linewidth',2);
if norm([ux2,uy2])>(c2*cos(alpha2))
    Max2=[0 0.1*c2*cos(alpha2) 0.1*c2*cos(alpha2)];
    May2=[0 0.1*c2*sin(alpha2) 0];
    Mbx2=[u2,u2-0.1*w2*cos(beta2),u2-0.1*w2*cos(beta2)];
    Mby2=[0,0,0.1*w2*sin(beta2)];
    patch(Max2,May2,'y','FaceAlpha',0.7,'EdgeColor','none');
    patch(Mbx2,Mby2,'m','FaceAlpha',0.7,'EdgeColor','none');
else
    Max2=[0 0.1*c2*cos(alpha2) 0.1*c2*cos(alpha2)];
    May2=[0 0.1*c2*sin(alpha2) 0];
    Mbx2=[u2,u2-0.1*u2,u2-0.1*w2*cos(beta2)];
    Mby2=[0,0,0.1*w2*sin(beta2)];
    patch(Max2,May2,'y','FaceAlpha',0.7,'EdgeColor','none');
    patch(Mbx2,Mby2,'m','FaceAlpha',0.7,'EdgeColor','none');
end

%% Mostrar resultados
set(du1,'String',u1);
set(dc1,'String',c1);
set(dw1,'String',w1);
set(dalpha1,'String',alpha1*180/pi);
set(dbeta1,'String',beta1*180/pi);
set(du2,'String',u2);
set(dc2,'String',c2);
set(dw2,'String',w2);
set(dalpha2,'String',alpha2*180/pi);
set(dbeta2,'String',beta2*180/pi);
end

function valores(~,~)
    conf.Interpreter='tex';
    prompt={'u_1 (m/s)','c_1 (m/s)','\alpha_1 ( ° )','u_2 (m/s)',...
        'c_2 (m/s)','\alpha_2 ( ° )'};
    name='MFITEC-Valores para triángulos de velocidades';
    numlines=1;
    defaultanswer={'10','15','90','12','6','45'};
    val=inputdlg(prompt,name,numlines,defaultanswer,conf);
    u1=str2num(val{1});
    c1=str2num(val{2});
    alpha1=str2num(val{3})*pi/180;
    u2=str2num(val{4});
    c2=str2num(val{5});
    alpha2=str2num(val{6})*pi/180;
    close(gcf);
    triangulos(u1,c1,alpha1,u2,c2,alpha2);
end
end

```

Interfaz de trazo de gráficas

```

function grafica(x,y,xlab,ylab,encab)

if any(isnan(x)) | any(isnan(y))
    errordlg('No se han introducido datos','MFITEC - Error');
else
    cfondo=[0.8 0.85 0.9];
    fg=figure('MenuBar','none','NumberTitle','off','Name','MFITEC - Gráfica',...

```

```

    'position',[100 100 760 500],'color',cfondo,'resize','off');
    centerfig(fg);

    %% Axes y Panel
    ax=axes('parent',fg,'units','pixels','position',[70 50 450 400]);
    pp=uipanel(fg,'units','pixels','position',[550 50 190 400],...
        'backgroundcolor',cfondo*0.9);
    cpanel=get(pp,'backgroundcolor');

    %% Modificadores
    uicontrol(pp,'style','text','String','Color de linea','Position',...
        [20 370 150 20],'backgroundcolor',cpanel,'fontweight','bold');
    scolor=uicontrol(pp,'style','popupmenu','String',{'Azul','Rojo',...
        'Amarillo','Blanco','Negro','Verde','Magenta','Naranja','Rosa','Gris'},...
        'Position',[20 350 150 20],'callback',@selcolor);

    uicontrol(pp,'style','text','String','Grosor de linea','Position',...
        [20 320 150 20],'backgroundcolor',cpanel,'fontweight','bold');
    sgrosor=uicontrol(pp,'style','popupmenu','String',{'1','2','3','4','5',...
        '6','7','8','9','10'},'Position',[20 300 150 20],...
        'callback',@selgrosor);

    uicontrol(pp,'style','text','String','Estilo de linea','Position',...
        [20 270 150 20],'backgroundcolor',cpanel,'fontweight','bold');
    sestilo=uicontrol(pp,'style','popupmenu','String',{'Continua',...
        'Semicontinua','Puntos','Guion Punto'},'Position',[20 250 150 20],...
        'callback',@selestilo);

    uicontrol(pp,'style','text','String','Color de fondo','Position',...
        [20 220 150 20],'backgroundcolor',cpanel,'fontweight','bold');
    scfond=uicontrol(pp,'style','popupmenu','String',{'Azul','Rojo',...
        'Amarillo','Blanco','Negro','Verde','Magenta','Naranja','Rosa','Gris'},...
        'Position',[20 200 150 20],'callback',@selfondo);

    uicontrol(pp,'style','checkbox','str','Mostrar rejilla','position',...
        [50 165 100 20],'backgroundcolor',cpanel,'callback',@rejilla);

    uicontrol(pp,'style','pushbutton','String','Modificar etiquetas',...
        'position',[20 115 150 25],'fontweight','bold','fontsize',9,...
        'callback',@modlab,'backgroundcolor',[0.8 0.8 0.95]);

    uicontrol(pp,'style','pushbutton','String','Exportar datos',...
        'position',[20 90 150 25],'fontweight','bold','fontsize',9,...
        'callback',{@gtabla,xlab,ylab},'backgroundcolor',[0.8 0.8 0.95]);

    uicontrol(pp,'style','pushbutton','String','Importar datos',...
        'position',[20 65 150 25],'fontweight','bold','fontsize',9,...
        'callback',@importar,'backgroundcolor',[0.8 0.8 0.95]);

    uicontrol(pp,'style','pushbutton','String','Guardar gráfica',...
        'position',[20 40 150 25],'fontweight','bold','fontsize',9,...
        'callback',@gimagen,'backgroundcolor',[0.8 0.8 0.95]);

    uicontrol(pp,'style','pushbutton','String','Nueva Gráfica',...
        'position',[20 15 150 25],'fontweight','bold','fontsize',9,...
        'callback',@nueva,'backgroundcolor',[0.8 0.8 0.95]);

    %% Gráfica
    hp=plot(x,y,'linewidth',2);grid;
    xlabel(xlab);

```

```

ylabel(ylab);
title(encab);
datacursormode('on');

end % Fin selector

%% Funciones
function selcolor(src,event)
    A=get(scolor,'value');
    M=[0 0 1;
        1 0 0;
        1 1 0;
        1 1 1;
        0 0 0;
        0 1 0;
        1 0 1;
        0.902 0.373 0;
        0.97 0.75 0.74;
        0.7 0.7 0.7];
    set(hp,'color',M(A,:));
end
function selgrosor(src,event)
    B=get(sgrosor,'value');
    set(hp,'linewidth',B);
end
function selestilo(src,event)
    C=get(sestilo,'value');
    switch C
        case 1
            set(hp,'LineStyle','-');
        case 2
            set(hp,'LineStyle','--');
        case 3
            set(hp,'LineStyle',':');
        case 4
            set(hp,'LineStyle','-.'');
    end
end
function gtabla(src,event,xlab,ylab)
    vx=get(hp,'xdata');
    vy=get(hp,'ydata');
    tablaValores(vx,vy,xlab,ylab);
end
function importar(~,~)
    [filename, pathname] = uigetfile('*.txt',...
        'Seleccione un archivo de datos');
    if isequal(filename,0)
        return;
    else
        archivo=fullfile(pathname, filename);
        datos=importdata(archivo);
        set(hp,'xdata',datos(:,1),'ydata',datos(:,2));
    end
end
function gimagen(src,event)
    [filename, pathname] = uiputfile('*.png','Guardar gráfica como...');
    if isequal(filename,0) || isequal(pathname,0)
        return;
    else

```

```

        faux=figure('visible','off');
        aux=axes('parent',faux);
        copyobj(hp,aux);
        saveas(faux,fullfile(pathname,filename));
        delete(faux);
    end
end

function selfondo(src,event)
    D=get(scfond,'value');
    M=[0 0 1;
        1 0 0;
        1 1 0;
        1 1 1;
        0 0 0;
        0 1 0;
        1 0 1;
        0.902 0.373 0;
        0.97 0.75 0.74;
        0.7 0.7 0.7];
    set(ax,'color',M(D,:));
end

function modlab(~,~)
    conf.Interpreter='tex';
    prompt={'Etiqueta eje horizontal','Etiqueta eje vertical','Titulo'};
    name='MFITEC';
    numlines=1;
    defaultanswer={get(get(ax,'xlabel'),'str'),get(get(ax,'ylabel'),...
        'str'),get(get(ax,'title'),'str')};
    val=inputdlg(prompt,name,numlines,defaultanswer,conf);
    if ~isempty(val)
        set(get(ax,'xlabel'),'str',val{1});
        set(get(ax,'ylabel'),'str',val{2});
        set(get(ax,'title'),'str',val{3});
    else
        return
    end
end

function rejilla(src,~)
    k=get(src,'value');
    if k==1
        grid('on');
    else
        grid('off');
    end
end

function nueva(~,~)
    conf.Interpreter='tex';
    prompt={'Vector de la variable independiente','Función',...
        'Etiqueta eje horizontal','Etiqueta eje vertical'};
    name='MFITEC-Valores para triángulos de velocidades';
    numlines=1;
    defaultanswer={'0:0.01:100','exp(-0.05*x)*sin(x)','x','f(x)'};
    val=inputdlg(prompt,name,numlines,defaultanswer,conf);
    if ~isempty(val)
        x=str2num(val{1});
        y=subs(val{2},'x',x);
        etx=val{3};
        ety=val{4};
        close(gcf);
    end
end

```

```

        grafica(x,y,etx,ety,'');
    else
        warndlg('Operación cancelada','MFITEC');
        return;
    end
end
end % Fin Función

```

Interfaz de presión de saturación

```

function presat
fondoc=randi([5,10],1,3)/10;
f=figure('MenuBar','None','NumberTitle','off','Name','Presión de saturación',...
    'position',[300 200 200 200],'color',fondoc,'resize','off');
centerfig(f);

uicontrol('Style','text','str','Temperatura ( °C )','position',...
    [50 170 105 20],'backgroundcolor',fondoc,'fontsize',9);
dT=uicontrol('style','edit','str','', 'position',[50 150 100 20],'fontsize',9,...
    'fontweight','bold');

uicontrol('Style','text','str','Presión de Saturación( kPa )',...
    'position',[10 100 180 20],'backgroundcolor',fondoc,'fontsize',9);
dPsv=uicontrol('style','text','str','', 'position',[50 80 100 20],...
    'fontsize',10,'fontweight','bold');

uicontrol('style','push','str','Calcular','position',[60 30 80 25],...
    'fontsize',10,'fontweight','bold','callback',@calcular);

helpdlg({'Proporcione la temperatura en el rango [0.01 120]°C',...
    'El resultado está dado en kPa para la Presión de saturación'},...
    'MFITEC');

function calcular(src,event)
    T=str2double(get(dT,'String'));
    M=[0.01 0.6117; % Matriz T (°C) - P (kPa) (Sat)
        5 0.8725;
        10 1.2281;
        15 1.7057;
        20 2.3392;
        25 3.1698;
        30 4.2469;
        35 5.6291;
        40 7.3851;
        45 9.5953;
        50 12.352;
        55 15.763;
        60 19.947;
        65 25.043;
        70 31.202;
        75 38.597;
        80 47.416;
        85 57.868;
        90 70.183;
        95 84.609;
        100 101.42;
        105 120.9;
        110 143.38;
        115 169.18;
        120 198.67];

```



```

        if T>120 || T<0
            set(dPsv,'str','No disponible');
        else
            Psv=interp1(M(:,1),M(:,2),T);
            set(dPsv,'str',num2str(Psv));
        end
    end
end
end

```

Interfaz de coeficiente de pérdidas primarias

```

function coefFriccionPP
% Cálculo del coeficiente de fricción de pérdidas primarias
% en tuberías.
cfondo=[1 1 1];
f=figure('NumberTitle','off','MenuBar','none','Name',...
    'MFITEC - Coeficiente de fricción','position',...
    [300 200 200 80],'color',cfondo,'resize','off');
centerfig(f);

uicontrol('style','pushbutton','String','Insertar datos...','position',...
    [40 45 120 25],'fontsize',10,'fontweight','bold','callback',@datos,...
    'BackgroundColor',[0 0 0],'foregroundcolor','w');

uicontrol('style','pushbutton','String','Calcular','position',...
    [60 10 80 25],'fontsize',10,'fontweight','bold','callback',@calcular);

function calcular(src,event)
    global V D Eps visc
    if any(isnan([Eps,V,D,visc]))
        warndlg('Por favor rellene los campos correspondientes...','...
            'MFITEC Error');
    else
        Re=(V*D)/visc;
        eccooleb=@(F) -1/sqrt(F)-2*log10(((Eps/D)/3.7)+(2.51/(Re*sqrt(F))));
        F=fzero(eccooleb,0.1);
        res1=horzcat('Re= ',num2str(Re));
        res2=horzcat('f= ',num2str(F));
        figresultados({res1,res2},1);
    end
end

function datos(src,event)
    global V D Eps visc
    conf.Interpreter='tex';
    txt={'Velocidad (m/s)','Diámetro (mm)',...
        'Coeficiente de rugosidad \epsilon (mm)',...
        'Viscosidad cinemática (m^2/s)*10^{-6}'};
    nomb='MFITEC-Datos';
    respdef={'10','300','0.17','1.007'};
    resp=inputdlg(txt,nomb,1,respdef,conf);
    V=str2double(resp{1});
    D=str2double(resp{2});
    Eps=str2double(resp{3});
    visc=str2double(resp{4});
end
end

```

Interfaz de selección de datos (Bombas centrífugas)

```
function varargout = App_Bombas(varargin)
% APP_BOMBAS MATLAB code for App_Bombas.fig
%   APP_BOMBAS, by itself, creates a new APP_BOMBAS or raises the existing
%   singleton*.
%
%   H = APP_BOMBAS returns the handle to a new APP_BOMBAS or the handle to
%   the existing singleton*.
%
%   APP_BOMBAS('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in APP_BOMBAS.M with the given input arguments.
%
%   APP_BOMBAS('Property','Value',...) creates a new APP_BOMBAS or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before App_Bombas_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to App_Bombas_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help App_Bombas

% Last Modified by GUIDE v2.5 17-Dec-2013 23:24:31

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @App_Bombas_OpeningFcn, ...
                  'gui_OutputFcn',  @App_Bombas_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% --- Executes just before App_Bombas is made visible.
function App_Bombas_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
movegui('center');
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = App_Bombas_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

% --- Executes on button press in gH.
function gH_Callback(hObject, eventdata, handles)
```

```

% --- Executes on button press in gHu.
function gHu_Callback(hObject, eventdata, handles)

% --- Executes on button press in grho.
function grho_Callback(hObject, eventdata, handles)

% --- Executes on button press in gQ.
function gQ_Callback(hObject, eventdata, handles)

% --- Executes on button press in gul.
function gul_Callback(hObject, eventdata, handles)

% --- Executes on button press in gcl.
function gcl_Callback(hObject, eventdata, handles)

% --- Executes on button press in gclu.
function gclu_Callback(hObject, eventdata, handles)

% --- Executes on button press in gclm.
function gclm_Callback(hObject, eventdata, handles)

% --- Executes on button press in galpha1.
function galpha1_Callback(hObject, eventdata, handles)

% --- Executes on button press in gbetal.
function gbetal_Callback(hObject, eventdata, handles)

% --- Executes on button press in gD1.
function gD1_Callback(hObject, eventdata, handles)

% --- Executes on button press in gb1.
function gb1_Callback(hObject, eventdata, handles)

% --- Executes on button press in gu2.
function gu2_Callback(hObject, eventdata, handles)

% --- Executes on button press in gc2.
function gc2_Callback(hObject, eventdata, handles)

% --- Executes on button press in gc2u.
function gc2u_Callback(hObject, eventdata, handles)

% --- Executes on button press in gc2m.
function gc2m_Callback(hObject, eventdata, handles)

% --- Executes on button press in galpha2.
function galpha2_Callback(hObject, eventdata, handles)

% --- Executes on button press in gbeta2.
function gbeta2_Callback(hObject, eventdata, handles)

% --- Executes on button press in gD2.
function gD2_Callback(hObject, eventdata, handles)

% --- Executes on button press in gb2.
function gb2_Callback(hObject, eventdata, handles)

% --- Executes on button press in gPa.
function gPa_Callback(hObject, eventdata, handles)

% --- Executes on button press in gPi.
function gPi_Callback(hObject, eventdata, handles)

```

```

% --- Executes on button press in gP.
function gP_Callback(hObject, eventdata, handles)

% --- Executes on button press in geta_v.
function geta_v_Callback(hObject, eventdata, handles)

% --- Executes on button press in geta_h.
function geta_h_Callback(hObject, eventdata, handles)

% --- Executes on button press in geta_m.
function geta_m_Callback(hObject, eventdata, handles)

% --- Executes on button press in geta.
function geta_Callback(hObject, eventdata, handles)

% --- Executes on button press in gHrint.
function gHrint_Callback(hObject, eventdata, handles)

% --- Executes on button press in gHrext.
function gHrext_Callback(hObject, eventdata, handles)

% --- Executes on button press in gHra.
function gHra_Callback(hObject, eventdata, handles)

% --- Executes on button press in gHri.
function gHri_Callback(hObject, eventdata, handles)

% --- Executes on button press in gda.
function gda_Callback(hObject, eventdata, handles)

% --- Executes on button press in gdi.
function gdi_Callback(hObject, eventdata, handles)

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% Datos totales
totdatos=get(findobj('style','checkbox'),'value');
ndd=nnz(cell2mat(totdatos));

% Datos necesarios en cada tipo/subtipo
% T1
p1s1=cell2mat(get(findobj('str','H','-or','str','eta_h','-or','str','D2',...
'-or','str','b2','-or','str','beta2','-or','str','n',...
'-or','str','Hrext'),'value'));
p1s2=cell2mat(get(findobj('str','Pa','-or','str','eta','-or','str','D2',...
'-or','str','b2','-or','str','beta2','-or','str','n'),'value'));

% T2
p2s1=cell2mat(get(findobj('str','D1','-or','str','D2','-or','str','b1',...
'-or','str','b2','-or','str','Q','-or','str','H',...
'-or','str','n','-or','str','eta'),'value'));

% T3
p3s1=cell2mat(get(findobj('str','c1','-or','str','c2','-or','str','alpha1',...
'-or','str','alpha2','-or','str','D1','-or','str','D2',...
'-or','str','n'),'value'));

% T4
p4s1=cell2mat(get(findobj('str','di','-or','str','da','-or','str','La',...
'-or','str','Li','-or','str','zZ','-or','str','zA',...
'-or','str','Q','-or','str','eta','-or','str','Eps','-or','str','Ka',...
'-or','str','Ki'),'value'));

```

```

p4s2=cell2mat(get(findobj('str','di','-or','str','da','-or','str','La',...
    '-or','str','Li','-or','str','zZ','-or','str','zA',...
    '-or','str','Q','-or','str','eta','-or','str','Eps'),'value'));
p4s3=cell2mat(get(findobj('str','di','-or','str','da','-or','str','La',...
    '-or','str','Li','-or','str','zZ','-or','str','zA',...
    '-or','str','Q','-or','str','eta_h','-or','str','Eps','-or','str',...
    'Ka','-or','str','Ki','-or','str','eta_v','-or',...
    'str','eta_m'),'value'));

% T5
p5s1=cell2mat(get(findobj('str','D1','-or','str','D2','-or','str','b1',...
    '-or','str','b2','-or','str','betal','-or','str','beta2',...
    '-or','str','n','-or','str','eta'),'value'));
p5s2=cell2mat(get(findobj('str','D1','-or','str','D2','-or','str','b1',...
    '-or','str','b2','-or','str','betal','-or','str','beta2',...
    '-or','str','n','-or','str','ME'),'value'));
p5s3=cell2mat(get(findobj('str','D1','-or','str','D2','-or','str','b1',...
    '-or','str','b2','-or','str','betal','-or','str','beta2',...
    '-or','str','n'),'value'));
p5s4=cell2mat(get(findobj('str','D1','-or','str','D2','-or','str','b1',...
    '-or','str','b2','-or','str','betal','-or','str','beta2',...
    '-or','str','Q','-or','str','eta'),'value'));
p5s5=cell2mat(get(findobj('str','D1','-or','str','D2','-or','str','A1',...
    '-or','str','A2','-or','str','betal','-or','str','beta2',...
    '-or','str','n','-or','str','eta'),'value'));

% T6
p6s1=cell2mat(get(findobj('str','D1','-or','str','D2','-or','str','b1',...
    '-or','str','b2','-or','str','betal','-or','str','beta2',...
    '-or','str','n','-or','str','zS','-or','str','zE','-or','str','zA',...
    '-or','str','da','-or','str','di'),'value'));

% T7
p7s1=cell2mat(get(findobj('str','Q','-or','str','PS','-or','str','PE',...
    '-or','str','zS','-or','str','zE','-or','str','eta'),'value'));
p7s2=cell2mat(get(findobj('str','Q','-or','str','MS','-or','str','ME',...
    '-or','str','zS','-or','str','zE','-or','str','da',...
    '-or','str','di'),'value'));
p7s3=cell2mat(get(findobj('str','Q','-or','str','PS','-or','str','PE',...
    '-or','str','zS','-or','str','zE','-or','str','eta',...
    '-or','str','da','-or','str','di'),'value'));
p7s4=cell2mat(get(findobj('str','Q','-or','str','PS','-or','str','PE',...
    '-or','str','eta'),'value'));
p7s5=cell2mat(get(findobj('str','Q','-or','str','da','-or','str','zA',...
    '-or','str','zE','-or','str','zS','-or','str','Pa',...
    '-or','str','eta'),'value'));

% T8
p8s1=cell2mat(get(findobj('str','zZ','-or','str','zA','-or','str','clm',...
    '-or','str','c2m','-or','str','beta2','-or','str','vi',...
    '-or','str','eta'),'value'));

% T9
p9s1=cell2mat(get(findobj('str','D2','-or','str','n','-or','str','beta2',...
    '-or','str','c2m'),'value'));
p9s2=cell2mat(get(findobj('str','D2','-or','str','n','-or','str','beta2',...
    '-or','str','c2m','-or','str','eta_h'),'value'));
p9s3=cell2mat(get(findobj('str','D2','-or','str','u2','-or','str','beta2',...
    '-or','str','c2m'),'value'));
p9s4=cell2mat(get(findobj('str','D2','-or','str','u2','-or','str','beta2',...
    '-or','str','c2m','-or','str','eta_h'),'value'));

% T10

```

```

p10s1=cell2mat(get(findobj('str','Hra','-or','str','Hri','-or','str','zA',...
    '-or','str','zZ','-or','str','D2','-or','str','b2',...
    '-or','str','beta2','-or','str','n','-or','str','zE','-or','str','zS',...
    '-or','str','da','-or','str','di'),'value'));
p10s2=cell2mat(get(findobj('str','Hra','-or','str','Hri','-or','str','zA',...
    '-or','str','zZ','-or','str','D2','-or','str','b2',...
    '-or','str','beta2','-or','str','n','-or','str','D1'),'value'));

% T11
p11s1=cell2mat(get(findobj('str','zZ','-or','str','zA','-or','str','Q',...
    '-or','str','Hrext','-or','str','eta'),'value'));
p11s2=cell2mat(get(findobj('str','zZ','-or','str','zA','-or','str','Q',...
    '-or','str','Hrext','-or','str','Pa'),'value'));
p11s3=cell2mat(get(findobj('str','zZ','-or','str','zA','-or','str','Q',...
    '-or','str','Hra','-or','str','Hri','-or','str','eta'),'value'));
p11s4=cell2mat(get(findobj('str','zZ','-or','str','zA','-or','str','Q',...
    '-or','str','Hra','-or','str','Hri','-or','str','Pa'),'value'));

% T12
p12s1=cell2mat(get(findobj('str','zZ','-or','str','zA','-or','str','Kta',...
    '-or','str','Kti','-or','str','da','-or','str','di',...
    '-or','str','Q','-or','str','eta'),'value'));
p12s2=cell2mat(get(findobj('str','zZ','-or','str','zA','-or','str','Kta',...
    '-or','str','Kti','-or','str','da','-or','str','di',...
    '-or','str','Q','-or','str','Pa'),'value'));

% T13
p13s1=cell2mat(get(findobj('str','n','-or','str','D1','-or','str','beta1',...
    '-or','str','clm'),'value'));
p13s2=cell2mat(get(findobj('str','ul','-or','str','D1','-or','str','beta1',...
    '-or','str','clm'),'value'));

% T14
p14s1=cell2mat(get(findobj('str','PA','-or','str','PSV','-or','str','Hra',...
    '-or','str','Q','-or','str','H','-or','str','n',...
    '-or','str','NPSH'),'value'));
p14s2=cell2mat(get(findobj('str','PA','-or','str','PSV','-or','str','Hra',...
    '-or','str','zE','-or','str','zA'),'value'));
p14s3=cell2mat(get(findobj('str','PA','-or','str','PSV','-or','str','Hra',...
    '-or','str','Q','-or','str','H','-or','str','n',...
    '-or','str','zA','-or','str','zE'),'value'));
p14s4=cell2mat(get(findobj('str','PA','-or','str','PSV','-or','str','zE',...
    '-or','str','zA','-or','str','Ka','-or','str','La',...
    '-or','str','da','-or','str','Eps','-or','str','Q'),'value'));

% T15
p15s1=cell2mat(get(findobj('str','ns','-or','str','Q','-or','str','H'),'value'));
p15s2=cell2mat(get(findobj('str','ns','-or','str','Q','-or','str','n'),'value'));
p15s3=cell2mat(get(findobj('str','ns','-or','str','H','-or','str','n'),'value'));
p15s4=cell2mat(get(findobj('str','ns','-or','str','P','-or','str','n'),'value'));
p15s5=cell2mat(get(findobj('str','ns','-or','str','P','-or','str','H'),'value'));
p15s6=cell2mat(get(findobj('str','n','-or','str','Q','-or','str','H'),'value'));
p15s7=cell2mat(get(findobj('str','n','-or','str','P','-or','str','H'),'value'));

% Selector -----
if all(p1s1==1) & ndd==nnz(p1s1)
    close(gcf);
    try
        b_1(1);
    catch er
        msgbox(er.message);
    end
end

```

```

elseif all(p1s2==1) & ndd==nnz(p1s2)
    close(gcf);
    b_1(2);
elseif all(p2s1==1) & ndd==nnz(p2s1)
    close(gcf);
    try
        run('b_2.m');
    catch er
        msgbox(er.message);
    end
elseif all(p3s1==1) & ndd==nnz(p3s1)
    close(gcf);
    run('b_3.m');
elseif all(p4s1==1) & ndd==nnz(p4s1)
    close(gcf);
    run('b_4(1).m');
elseif all(p4s2==1) & ndd==nnz(p4s2)
    close(gcf);
    run('b_4(2).m');
elseif all(p4s3==1) & ndd==nnz(p4s3)
    close(gcf);
    run('b_4(3).m');
elseif all(p5s1==1) & ndd==nnz(p5s1)
    close(gcf);
    run('b_5(1).m');
elseif all(p5s2==1) & ndd==nnz(p5s2) % PR
    close(gcf);
    run('b_5(2).m');
elseif all(p5s3==1) & ndd==nnz(p5s3) % PR
    close(gcf);
    run('b_5(3).m');
elseif all(p5s4==1) & ndd==nnz(p5s4)
    close(gcf);
    run('b_5(4).m');
elseif all(p5s5==1) & ndd==nnz(p5s5)
    close(gcf);
    run('b_5(5).m');
elseif all(p6s1==1) & ndd==nnz(p6s1) % PR
    close(gcf);
    run('b_6.m');
elseif all(p7s1==1) & ndd==nnz(p7s1)
    close(gcf);
    run('b_7(1).m');
elseif all(p7s2==1) & ndd==nnz(p7s2)
    close(gcf);
    run('b_7(2).m');
elseif all(p7s3==1) & ndd==nnz(p7s3)
    close(gcf);
    run('b_7(3).m');
elseif all(p7s4==1) & ndd==nnz(p7s4)
    close(gcf);
    run('b_7(4).m');
elseif all(p7s5==1) & ndd==nnz(p7s5)
    close(gcf);
    run('b_7(5).m');
elseif all(p8s1==1) & ndd==nnz(p8s1)
    close(gcf);
    run('b_8.m');
elseif all(p9s1==1) & ndd==nnz(p9s1)
    close(gcf);
    run('b_9(1).m');
elseif all(p9s2==1) & ndd==nnz(p9s2)
    close(gcf);

```

```

        run('b_9(2).m');
elseif all(p9s3==1) & ndd==nnz(p9s3)
    close(gcf);
    run('b_9(3).m');
elseif all(p9s4==1) & ndd==nnz(p9s4)
    close(gcf);
    run('b_9(4).m');
elseif all(p10s1==1) & ndd==nnz(p10s1)
    close(gcf);
    run('b_10(1).m');
elseif all(p10s2==1) & ndd==nnz(p10s2)
    close(gcf);
    run('b_10(2).m');
elseif all(p11s1==1) & ndd==nnz(p11s1)
    close(gcf);
    run('b_11(1).m');
elseif all(p11s2==1) & ndd==nnz(p11s2)
    close(gcf);
    run('b_11(2).m');
elseif all(p11s3==1) & ndd==nnz(p11s3)
    close(gcf);
    run('b_11(3).m');
elseif all(p11s4==1) & ndd==nnz(p11s4)
    close(gcf);
    run('b_11(4).m');
elseif all(p12s1==1) & ndd==nnz(p12s1)
    close(gcf);
    run('b_12(1).m');
elseif all(p12s2==1) & ndd==nnz(p12s2)
    close(gcf);
    run('b_12(2).m');
elseif all(p13s1==1) & ndd==nnz(p13s1)
    close(gcf);
    run('b_13(1).m');
elseif all(p13s2==1) & ndd==nnz(p13s2)
    close(gcf);
    run('b_13(2).m');
elseif all(p14s1==1) & ndd==nnz(p14s1)
    close(gcf);
    run('b_14(1).m');
elseif all(p14s2==1) & ndd==nnz(p14s2)
    close(gcf);
    run('b_14(2).m');
elseif all(p14s3==1) & ndd==nnz(p14s3)
    close(gcf);
    run('b_14(3).m');
elseif all(p14s4==1) & ndd==nnz(p14s4)
    close(gcf);
    run('b_14(4).m');
elseif all(p15s1==1) & ndd==nnz(p15s1)
    close(gcf);
    run('b_15(1).m');
elseif all(p15s2==1) & ndd==nnz(p15s2)
    close(gcf);
    run('b_15(2).m');
elseif all(p15s3==1) & ndd==nnz(p15s3)
    close(gcf);
    run('b_15(3).m');
elseif all(p15s4==1) & ndd==nnz(p15s4)
    close(gcf);
    run('b_15(4).m');
elseif all(p15s5==1) & ndd==nnz(p15s5)
    close(gcf);

```



```

        run('b_15(5).m');
elseif all(p15s6==1) & ndd==nnz(p15s6)
    close(gcf);
    run('b_15(6).m');
elseif all(p15s7==1) & ndd==nnz(p15s7)
    close(gcf);
    run('b_15(7).m');
else
    warndlg('Datos no suficientes, verifique nuevamente...','MFITEC');
end

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
winopen('data/docs/Notacion_b.html');

% --- Executes during object creation, after setting all properties.
function pushbutton2_CreateFcn(hObject, eventdata, handles)

% -----
function Untitled_1_Callback(hObject, eventdata, handles)

% -----
function menu_ayuda_Callback(hObject, eventdata, handles)
winopen('data/docs/Ayuda.pdf');

% -----
function acerca_Callback(hObject, eventdata, handles)
run('acerca.m');

% -----
function menu_convertor_Callback(hObject, eventdata, handles)
winopen('data/mfun/convert.exe');

% -----
function lab_tablas_Callback(hObject, eventdata, handles)
winopen('data/docs/Tablas.pdf');

% -----
function menu_moody_Callback(hObject, eventdata, handles)
winopen('data/docs/moody.pdf');

% --- Executes on button press in gn.
function gn_Callback(hObject, eventdata, handles)

% --- Executes on button press in goinicio.
function goinicio_Callback(hObject, eventdata, handles)
run('Inicio.m');
close('App_Bombas');

% --- Executes on button press in bayuda.
function bayuda_Callback(hObject, eventdata, handles)
winopen('data/docs/Ayuda.pdf');

% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
winopen('data/mfun/convert.exe');

% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)
run('menu_estudio_b.m');

% --- Executes on button press in checkbox35.

```

```

function checkbox35_Callback(hObject, eventdata, handles)

% --- Executes on button press in checkbox36.
function checkbox36_Callback(hObject, eventdata, handles)

% --- Executes on button press in checkbox37.
function checkbox37_Callback(hObject, eventdata, handles)

% --- Executes on button press in gPS.
function gPS_Callback(hObject, eventdata, handles)

% --- Executes on button press in gPE.
function gPE_Callback(hObject, eventdata, handles)

% --- Executes on button press in checkbox40.
function checkbox40_Callback(hObject, eventdata, handles)

% --- Executes on button press in checkbox41.
function checkbox41_Callback(hObject, eventdata, handles)

% --- Executes on button press in checkbox42.
function checkbox42_Callback(hObject, eventdata, handles)

% --- Executes on button press in checkbox43.
function checkbox43_Callback(hObject, eventdata, handles)

% -----
function config_densidad_Callback(hObject, eventdata, handles)
opconf.Interpreter='tex';
nrho=inputdlg({'Inserte un nuevo valor en kg/m^3'}, 'Densidad', 1, {'1000'}, opconf);
rho=str2double(nrho);
save('data/mfun/rho.mat', 'rho');
clearvars('rho');

% --- Executes on button press in gME.
function gME_Callback(hObject, eventdata, handles)

% --- Executes on button press in gMS.
function gMS_Callback(hObject, eventdata, handles)

% --- Executes on button press in gKa.
function gKa_Callback(hObject, eventdata, handles)

% --- Executes on button press in gKi.
function gKi_Callback(hObject, eventdata, handles)

% --- Executes on button press in gEps.
function gEps_Callback(hObject, eventdata, handles)

% --- Executes on button press in gLi.
function gLi_Callback(hObject, eventdata, handles)

% --- Executes on button press in gLa.
function gLa_Callback(hObject, eventdata, handles)

% --- Executes on button press in checkbox51.
function checkbox51_Callback(hObject, eventdata, handles)

% --- Executes on button press in checkbox52.
function checkbox52_Callback(hObject, eventdata, handles)

% --- Executes on button press in checkbox53.
function checkbox53_Callback(hObject, eventdata, handles)

```

```
% --- Executes on button press in checkbox54.  
function checkbox54_Callback(hObject, eventdata, handles)  
  
% --- Executes on button press in checkbox55.  
function checkbox55_Callback(hObject, eventdata, handles)  
  
% --- Executes on button press in checkbox56.  
function checkbox56_Callback(hObject, eventdata, handles)  
  
% --- Executes on button press in checkbox57.  
function checkbox57_Callback(hObject, eventdata, handles)  
  
% --- Executes on button press in checkbox58.  
function checkbox58_Callback(hObject, eventdata, handles)
```