



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior de Linares

Trabajo Fin de Grado

**CREACIÓN DE APLICACIÓN
GRÁFICA CON MATLAB PARA EL
ANÁLISIS DE UNA TURBINA
PELTON.**

Alumno: Sergio Sánchez Castro

Tutor: Prof. D. Mario Miró Barnés
Prof. D^a. María Rocío Bolaños Jiménez

Depto.: Ingeniería Mecánica y Minera

Julio, 2014

Índice de contenido

1. RESUMEN	8
2. INTRODUCCIÓN	9
2.1 Tipos de centrales	9
2.2 Componentes de la turbina Pelton	12
2.2.1 Rodete	12
2.2.2 Inyector	13
2.2.2.1 Aguja de regulación	14
2.2.2.2 Deflector	15
2.3 Banco de ensayo	15
3. OBJETIVOS Y METODOLOGÍA	18
4. RESULTADOS Y DISCUSIÓN	19
4.1 Curvas de funcionamiento	19
4.1.1 Teoría general	19
4.1.2 Curvas dimensionales	23
4.1.3 Análisis dimensional. Curvas adimensionales	27
4.1.4 Semejanza	30
4.1.5 Velocidad específica y diámetro específico	32
4.4 INTERFAZ GRÁFICA	33
4.5 ESTUDIO DEL INYECTOR	54
4.5.1 Descripción del inyector	54
4.5.2 Caracterización de la aguja del inyector	58
4.5.3 Simulación numérica con ANSYS FLUENT	66
5. PLANOS	89
6. ANEXOS	91
6.1 Código de la interfaz gráfica	91

6.1.1 Código de la ventana de Inicio.	91
6.1.2 Código de la ventana del Menú Principal.....	92
6.1.3 Código de la ventana del botón de Ensayo.	97
6.1.4 Código de la ventana Gráficas.	98
6.1.5 Código de la ventana Coeficiente de desagüe.	105
6.1.6 Código de la ventana Semejanza.	108
6.1.7 Código de la ventana Inyector.	112
7. FUTURAS LÍNEAS DE TRABAJO	114
7.1 Compilación de la interfaz gráfica para poder ejecutarla en ordenadores sin MATLAB.....	114
7.2 Vídeo tutorial	114
7.3 Mejorar la velocidad de ejecución del programa.....	115
8. REFERENCIAS	115

Índice de figuras y planos

Figura 2.1 Esquema de una central de agua fluente.....	9
Figura 2.2 Esquema de una central de agua embalsada.....	10
Figura 2.3 Esquema de una central de bombeo.....	10
Figura 2.4 Ejemplos de rodete de turbina pelton.....	13
Figura 2.5 Configuración rodete-inyector verticales.....	14
Figura 2.6 Configuración rodete-inyector horizontales.....	14
Figura 2.7 En la figura la aguja que utiliza nuestra turbina de ensayos.....	14
Figura 2.8 Disposición más usual de los componentes del inyector.....	15
Figura 2.9 Banco de ensayos	15

Figura 2.10 Cuerpo, manómetro y mando regulador del inyector de la turbina	17
Figura 2.11 Dinamómetro y vista de su montaje en el freno de prony	17
Figura 2.12 Freno de prony	17
Figura 4.1 Esquema de velocidades de entrada y salida del rodete	22
Figura 4.2 Representación de los vectores de velocidad del rodete	22
Figura 4.3 Esquema de las velocidades de salida	22
Figura 4.4 Par motor real y teórico	24
Figura 4.5 Potencia útil teórica y real	25
Figura 4.6 Potencia hidráulica real y teórica	26
Figura 4.7 Representación de los rendimientos	27
Figura 4.8 Tablas de datos para introducir los valores obtenidos del ensayo	33
Figura 4.9 Vista aumentada de las tablas de datos	34
Figura 4.10 Marcado el archivo sobre el que tenemos que hacer doble click	35
Figura 4.11 Interfaz errónea debida a no ejecutarse el programa correctamente	35
Figura 4.12 Ventana el editor de Matlab	36
Figura 4.13 Primera interfaz del programa	36
Figura 4.14 La opción por defecto es “No”	37
Figura 4.15 Menú principal de la interfaz gráfica	38
Figura 4.16 Ventana de la interfaz gráfica “Vídeos del inyector”	39
Figura 4.17 Captura de pantalla del vídeo de la vista explosionada	40
Figura 4.18 Vídeo del montaje	40
Figura 4.19 Captura de pantalla del vídeo de llenado del inyector	41
Figura 4.20 Captura de pantalla de la ventana gráfica	42

Figura 4.21 Captura de pantalla de la ventana “Gráficas” al iniciarla	43
Figura 4.22 Botones de la barra de herramientas	43
Figura 4.23 Vista aumentada de uno de los espacios de trabajo de la ventana “Gráficas”	44
Figura 4.24 Vista aumentada del menú desplegable de la posición del inyector	44
Figura 4.25 Vista aumentada del menú desplegable de la pestaña gráficas.....	45
Figura 4.26 Vista aumentada del menú desplegable caudales	45
Figura 4.27 Vista aumentada de los botones representar y limpiar ejes	45
Figura 4.28 Figura que muestra los caudales para distintas posiciones del inyector...	46
Figura 4.29 Figura que muestra el potencias de la ventana “Gráficas”.....	47
Figura 4.30 Figura en la que se representa el par para todos los caudales y cada una de las posiciones del inyector	47
Figura 4.31En la figura se representa la potencia útil para todos los caudales y cada una de las posiciones del inyector	48
Figura 4.32 En la figura se representa la potencia hidráulica para todos los caudales y cada una de las posiciones del inyector.....	49
Figura 4.33 En la figura se representa el rendimiento para todos los caudales y cada una de las posiciones del inyector	50
Figura 4.34 Ventana Coeficiente de desagüe al ser iniciada.....	51
Figura 4.35 Coeficiente de desagüe y recta de ajuste ya representados.....	51
Figura 4.36 Ventana Semejanza al ser iniciada.....	52
Figura 4.37 Ventana Semejanza mostrando un ejemplo de su funcionamiento.....	53
Figura 4.38 Representación 3D del inyector con todos sus componentes	54
Figura 4.39 Cuerpo principal del inyector	54

Figura 4.40 Vista interior del cuerpo principal	55
Figura 4.41 Vista superior de la tobera de salida.....	55
Figura 4.42 Vista inferior o interna de la tobera de salida.....	55
Figura 4.43 Vista en alzado de la aguja	56
Figura 4.44 Vista detallada de la argolla excéntrica de la aguja	57
Figura 4.45 Vista en alzado de la aguja y la base de la misma	57
Figura 4.46 Vista en detalle del cabezal de la aguja	58
Figura 4.47 Vista de la rosca de la base de la aguja	58
Figura 4.48 Vista seccionada de las piezas aguja y tobera en su posición cerrada	59
Figura 4.49 Vista real del conjunto en la turbina	59
Figura 4.50 Vista seccionada de las piezas aguja y tobera	59
Figura 4.51 Vista real del conjunto en la turbina	60
Figura 4.52 Vista de la turbina antes de incluir la escala graduada en el mando regulador	61
Figura 4.53 Vista de las 10 marcas realizadas en la parte posterior del cuerpo del inyector	62
Figura 4.54 En la derecha vista sin las escala graduada	62
Figura 4.55 Vista detallada de la pegatina del mando del inyector	62
Figura 4.56 Vista del detalle de las marcas para facilitar la correcta posición del mando con respecto al cero	63
Figura 4.57 En la figura podemos ver la posición “1/2” del mando	63
Figura 4.58 Esquema de los diámetros: exterior (tobera) e interior (aguja)	64
Figura 4.59 Representación esquematizada de la punta de la aguja	65
Figura 4.60 Vista en corte de la zona por la que fluye el agua dentro del inyector	67

Figura 4.61 Representación 3D de la zona fluida	67
Figura 4.62 Vista de la malla tetraédrica	68
Figura 4.63 En azul tenemos la entrada y en rojo la salida	68
Figura 4.64 Panel general del programa ANSYS FLUENT de nuestra simulación ...	69
Figura 4.65 Panel de configuración del modelo de turbulencia	69
Figura 4.66 Los materiales de trabajo obviamente son el agua líquida	70
Figura 4.67 Selección de órdenes para los métodos de resolución	71
Figura 4.68 Residuos	71
Figura 4.69 Monitor de presión	71
Figura 4.70 Plano de corte para las vistas de los gradientes de velocidad y presión ...	72
Figura 4.71 En la figura se aprecia como el chorro no es horizontal	73
Figura 4.72 Vista de las trayectorias de las partículas fluidas	73
Figura 4.73 Vista de los vectores velocidad en la zona del cabezal de la aguja	74
Figura 4.74 Representación aumentada de los vectores velocidad en las zonas de turbulencia	74
Figura 4.75 Aquí vemos la recirculación provocada por la argolla de la aguja	75
Figura 4.76 Contornos de presión	75
Figura 4.77 Selección del modelo de turbulencia	76
Figura 4.78 Residuos	76
Figura 4.79 Este monitor sí que es bueno ya que toma un valor constante	77
Figura 4.80 Representación de los contornos de velocidad con las líneas que muestran el ángulo de desviación del chorro	78
Figura 4.81 Arriba: Vista global de los vectores de velocidad con las líneas que muestran el ángulo de desviación del chorro	78

Figura 4.82 Representación de la malla estructurada	80
Figura 4.83 Residuos	81
Figura 4.84 Monitor de presión	81
Figura 4.85 Gradientes de velocidad con líneas auxiliares que indican el ángulo del chorro	82
Figura 4.86 Gradientes de presión	83
Figura 4.87 En la figura se muestran rodeados por un círculo rojo los elementos mejorados en el inyector propuesto	84
Figura 4.88 Representación de la zona fluida del inyector propuesto	85
Figura 4.89 Malla estructurada del inyector propuesto	85
Figura 4.90 Residuos	86
Figura 4.91 Tras cambiar los órdenes de cálculo del solver llegamos a unos valores estables del monitor de presión	86
Figura 4.92 Gradientes de velocidad	87
Planos del inyector actual.....	91
Planos del inyector propuesto.....	100

1. RESUMEN

El departamento de Ingeniería Mecánica y Minera, en concreto, el de Mecánica de Fluidos de la Escuela Politécnica Superior de Linares cuenta con un banco de ensayo de una turbina Pelton.

Este banco de ensayos no dispone de ningún tipo de sensores electrónicos ni programa informático para monitorizar el ensayo. Además, la escueta información escrita que adjunta el banco no describe las curvas de trabajo de la máquina en ningún régimen de giro, siendo éstas necesarias para el correcto aprendizaje de los alumnos y la correcta descripción del modo de trabajo de la turbina.

Otros bancos de trabajo como el de la bomba centrífuga computerizada que hay en el mismo laboratorio, sí que disponen al menos de un software que recoge los datos que se obtienen en tiempo real durante del ensayo. El banco de la turbina Pelton no dispone de ningún sensor digital y todas las medidas se han de realizar usando instrumentos de medición mecánicos anexos al banco. Por esta razón sería ideal el desarrollo de una interfaz gráfica que trabajara con los datos obtenidos durante los ensayos y nos devuelva las curvas de funcionamiento de la turbina para distintos regímenes de giro.

Esta interfaz gráfica podría mejorar la docencia, ya que los alumnos pueden trabajar con los datos obtenidos en la realización de este Trabajo de Fin de Grado o podrán tomar datos propios y trabajar igualmente con ellos de forma intuitiva y sencilla.

Por otro lado, además de la realización de la aplicación gráfica, se ha llevado a cabo un estudio del inyector de la turbina. Para ello, se tomaron todas las medidas de las piezas que lo componen y así poder llevar a cabo una simulación numérica computacional. Este estudio de componentes a su vez, permitió analizar el sistema de posicionamiento de la aguja del inyector caracterizándolo, para así poder posicionar la aguja con total precisión mejorando la calidad de ensayos posteriores.

Por todo lo dicho anteriormente, junto a la motivación de demostrar los conocimientos adquiridos en los estudios de Grado en Ingeniería Industrial Especialidad Mecánica y el poder aportar un valor añadido a un banco de ensayos existente que facilite tanto la docencia a profesores como el aprendizaje a alumnos, me ha llevado a tomar la decisión de realizar este Trabajo de Fin de Grado.

2. INTRODUCCIÓN

2.1 Tipos de centrales

Como ya sabemos las centrales hidráulicas tienen un papel importante en la generación de energía eléctrica hoy día. Estas centrales cuentan con turbinas hidráulicas que aprovechan la energía de un fluido que pasa a través de ellas para producir un movimiento de rotación que, transferido mediante un eje, mueve directamente una máquina o bien un generador que transforma la energía mecánica en eléctrica. Dentro de estas centrales existen muchos tipos así como diversas formas para clasificarlas.

Para su clasificación podemos atender al tipo de agua que utilizan:

-Agua fluente: En este tipo de centrales (figura 2.1) parte del caudal del río se deriva a la central de forma que el impacto sobre el cauce natural del río es menor que en las centrales de agua embalsada. Podríamos emplear estas centrales en zonas con grandes desniveles (alta montaña) ya que así aprovecharíamos el salto de agua natural. En España no son habituales ya que necesitan un caudal constante todo el año.



Figura 2.1. Esquema de una central de agua fluente. Fuente: <http://tecnologia-cardones.wikispaces.com/4>

-Agua embalsada: En este otro tipo de central (figura 2.2) se construye un embalse artificial. Parte del agua acumulada irá destinada a alimentar las turbinas y si el caudal del río que lo alimenta es suficiente el agua será evacuada por el aliviadero de la presa. Este

tipo de presas se emplean cuando no disponemos de un gran salto natural, así suplimos la falta de presión con un mayor caudal gracias al embalse.

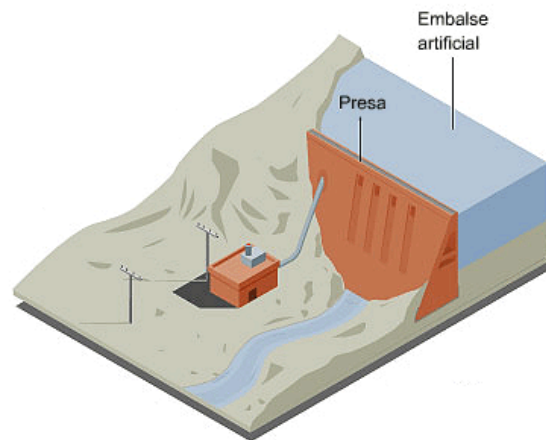


Figura 2.2. Esquema de una central de agua embalsada. Fuente : <http://www.vam.org.es/blog/xpost.php?id=minihidraulica>.

Dentro de las centrales de agua embalsada:

-Centrales de regulación de caudal: Como su propio nombre indica estas centrales cuentan con un mecanismo de regulación del caudal que se variará en función de la época del año en la que nos encontremos.

-Centrales de bombeo: Una central hidroeléctrica de bombeo (figura 2.3) es un tipo especial de central hidroeléctrica que tiene dos embalses. El agua contenida en el embalse situado en el nivel más bajo (embalse inferior), es bombeada durante las horas de menor demanda eléctrica al depósito situado en la cota más alta (embalse superior), con el fin de turbinarla, posteriormente, para generar electricidad en las horas de mayor consumo eléctrico.

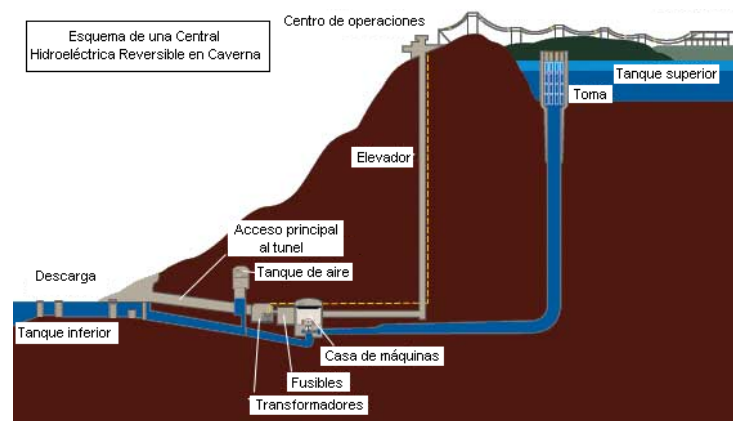


Figura 2.3. Esquema de una central de bombeo. Fuente:
http://es.wikipedia.org/wiki/Central_hidroel%C3%A9ctrica_reversible

Otra clasificación sería atendiendo a la altura del salto:

- Centrales de alta presión: Saltos mayores a 200 metros. Usual mente con caudales que rondan los $20 \text{ m}^3/\text{s}$.
- Centrales de media presión: Comprenden saltos de entre 200 a 20 metros. Sus caudales más usuales son de hasta $200 \text{ m}^3/\text{s}$.
- Centrales de baja presión: Saltos menores de 20 metros. Con caudales por turbina de unos $300 \text{ m}^3/\text{s}$.

Finalmente otra clasificación ampliamente usada es en función del tipo de turbina empleada en la central hidráulica:

- Turbinas Francis: Son turbinas de reacción. Se adaptan bien a una gran variedad de alturas de saltos ($H=30\sim 550$ metros) y de caudales, razón por la cual es la más instalada. A medida que se evoluciona a menores alturas y mayores caudales, el rodete modifica su geometría desde flujo casi radial con sección de entrada muy pequeña hasta flujo casi acial con sección de entrada muy grande. Sus velocidades específicas son medias.
- Turbinas Kaplan: En estas el flujo a su paso por el rodete es totalmente axial. En éstas los álabes del rodete cambian automáticamente de posición buscando siempre que el agua entre tangente a los mismos sea cual fuere la carga demandada a la central. Estas turbinas se adaptan bien a grandes caudales y pequeños saltos ($H=4\sim 90$ metros).
- Turbinas de bulbo: Estas se adaptan muy bien a pequeños saltos y caudales muy grandes. Se emplean por ejemplo en las estaciones maremotrices. A diferencia de las Kaplan éstas no cuentan con álabes móviles y normalmente el alternador está situado justo en la entrada del agua de manera que todo el conjunto es mucho más compacto que los montajes de las turbinas Kaplan.
- Turbinas Pelton: Son las únicas turbinas de acción. Se adaptan bien a pequeños caudales y grandes alturas ($H=100\sim 1800$ metros). Así pues, la potencia en la misma se consigue más por la altura que por el caudal. Su velocidad específica es baja.

2.2 Componentes de la turbina Pelton

De las que se acaban de describir, la turbina Pelton es la que nos interesa ya que es con la que cuenta nuestro banco de ensayos. En concreto la turbina Pelton es una turbina de acción, esto quiere decir, que en ella el fluido de trabajo no sufre un cambio de presión importante en su paso a través de rodete. Toda la energía potencial que posee el fluido se convierte en energía cinética gracias al inyector que expulsa el agua a presión ambiente. Tras ser expulsada el agua impacta con el rodete Pelton para hacerlo girar y así mover el generador o cualquier elemento que nos interese.

Las turbinas Pelton se adaptan bien a pequeños caudales y grandes saltos (entre 100 y 1800 metros). Así pues, la potencia se consigue más por altura que por el caudal. La velocidad específica de las turbinas Pelton es generalmente baja.

Ahora pasaremos a describir las distintas partes de la turbina. La turbina Pelton se compone de:

- Rodete

- Inyector

Dentro del inyector tenemos:

- Aguja de regulación

- Deflector

2.2.1 Rodete

Consta de una rueda con cucharas alrededor (figura 2.4), a las que podemos llamar también álabes, sobre las que actúa el chorro del inyector. El tamaño y número de cucharas dependen de las características de la instalación y/o de la velocidad específica. Cuanto menor sea el caudal y mayor la altura del salto, menor será el diámetro del chorro. Así, una velocidad específica pequeña demandará un solo inyector con diámetro de chorro pequeño y en consecuencia muchas cucharas pequeñas. En cambio, una velocidad específica mayor, requerirá cucharas más grandes y en menor número.

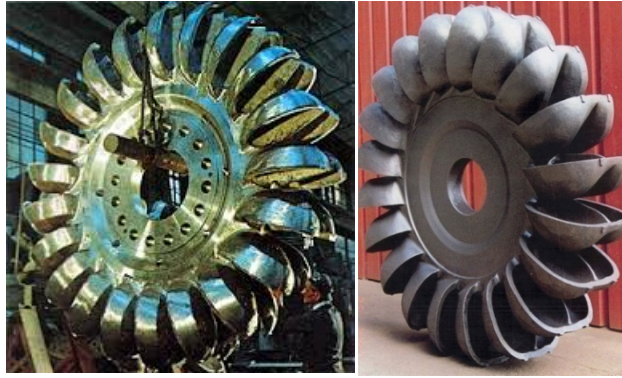


Figura 2.4. Ejemplos de rodete pelton. Fuente:
<http://faeitch2011.wordpress.com/unidad-2-turbinas-hidroelectricas/2-turbina-pelton/>

Cada vez que una cuchara entra en el campo de acción del chorro sufrirá un rechazo, de no ser porque ésta tiene una mella aproximadamente un 10% mayor que el diámetro del chorro. De manera que éste no actúa hasta tenerla enfrente.

Las cucharas tienen forma de copa elíptica dividida por una cresta afilada en dos partes simétricas. La cresta divide al chorro en dos mitades. Al dividirse el chorro en dos mitades aparece una componente axial de la fuerza que ejerce el agua sobre el rodete pero se compensa por dividirse el chorro en dos y las componentes son iguales y de sentido contrario.

2.2.2 Inyector

El inyector es una tobera diseñada para reducir hasta los valores deseados el caudal, y con ello la pérdida de carga en la conducción de acceso a la turbina. La energía disponible, prácticamente en forma de presión a la entrada de la tobera, se transforma a lo largo de ésta en velocidad, y además en su totalidad como corresponde a una turbina de acción.

Existen múltiples configuraciones de disposición del inyector, figura 2.5 y figura 2.6, algunas más usuales son:

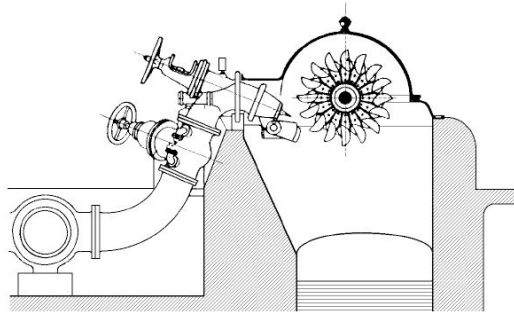


Figura 2.5. Configuración rodete-inyector verticales.

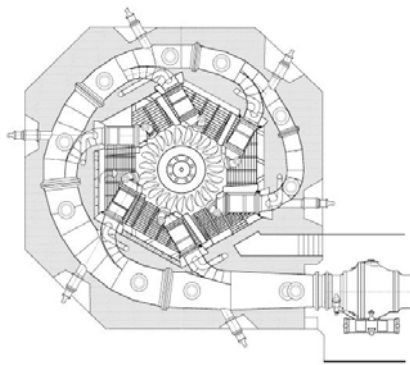


Figura 2.6. Configuración rodete inyector horizontales.

Si con un inyector, actuando adecuadamente sobre la geometría del rodete, podemos conseguir velocidades específicas de entre 10 y 30, con dos, cuatro, y hasta con 6 inyectores, podemos ir sucesivamente aumentando el campo de aplicación de la turbina Pelton. Se están utilizando cada vez más los inyectores de entrada recta.

2.2.2.1 Aguja de regulación

El inyector lleva en su interior una aguja de regulación (figura 2.7), que se desplaza entre dos posiciones límite de caudales nulo y máximo. Mandada por un servomotor (o cualquier otro medio mecánico), esta aguja ocupa en cada momento la posición correspondiente a la potencia exigida a la turbina. El actuador que mueve la aguja puede ser exterior o interior.



Figura 2.7. En la figura la aguja que utiliza la turbina de nuestro banco de ensayos, casi idéntica a las empleadas en las instalaciones industriales.

2.2.2.2 Deflector

Cuando disminuye la carga, hay que actuar sobre el caudal más rápidamente de lo que interesa a efectos de golpe de ariete. Para tal fin, cada inyector, figura 2.8, lleva incorporado al mecanismo de regulación un deflector, que intercepta inmediatamente el chorro, parcial o totalmente, mientras la aguja más lentamente ocupa la posición correspondiente a la nueva carga.

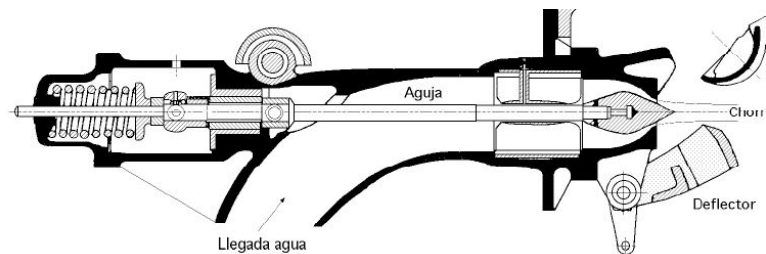


Figura 2.8. Disposición más usual de los componentes del inyector. Fuente: Mecánica de fluidos incompresibles, Agüera, José.

2.3 Banco de ensayo

El área de Mecánica de Fluidos cuenta con un banco de ensayos de una turbina Pelton se muestra en la figura 2.9. Este banco se utiliza en una de las asignaturas del grado de ingeniería industrial mecánica para realizar prácticas.



Figura 2.9. Banco de ensayos. Fuente: Guión de prácticas.

Los elementos que se enumeran en la figura son:

1. Turbina Pelton.
2. Dinamómetro de fricción (freno de Prony): consiste en una banda de fricción que frena una polea acoplada al eje de la turbina. La fuerza de la banda se regula mediante un bastidor y se mide mediante los dinamómetros que hay en los extremos. Mediante este sistema se puede calcular por tanto la potencia que se obtiene en el eje de la turbina.
3. Varilla dinamómetro.
4. Manómetro.
5. Tacómetro.
6. Entrada de agua al inyector.
7. Accionamiento del tapón para medir.
8. Regulador de apertura del inyector.
9. Banco de ensayo con bomba centrífuga incorporada, que servirá para simular el salto sobre el que se colocaría la turbina Pelton.
10. Lector para nivel de agua.

A diferencia de otros bancos éste no cuenta con ningún tipo de sensor electrónico que guarde variables durante el ensayo.

Proceso de toma de medidas:

El ensayo a realizar consistirá en los siguientes pasos:

1. Abrir completamente la válvula del banco de ensayos.
2. Fijar una posición del inyector.
3. Anotar el caudal que indica el rotámetro.
4. Anotar la presión que indica el manómetro situado en el inyector (bar) lo podemos ver en la figura 2.10. Obtenemos la altura neta transformando la presión.



Figura 2.10. Cuerpo, manómetro y mando regulador del inyector de la turbina.

5. Colocar el dinamómetro de fricción en su posición más alta (la polea no gira) podemos verlo en la figura 2.11. Anotar las fuerzas que mide en los extremos (F_1 y F_2 en N).



Figura 2.11. Dinamómetro y vista de su montaje en el freno de prony.

6. Repetir el paso anterior para 10 posiciones del dinamómetro. Para ello, ir soltándolo hasta llegar hasta que la fuerza sea nula. En cada posición hay que medir la velocidad de giro del rotor con el tacómetro (Ω en r.p.m.).

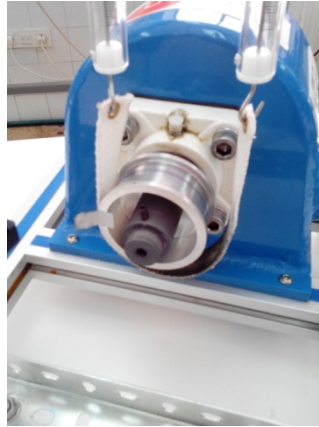


Figura 2.12. Freno de prony.

7. Repetir para otros dos caudales, es decir, es decir, para otras dos posiciones de la válvula del banco (3 en total).
8. Repetir para otras dos posiciones del inyector (3 en total).

Nota: el diámetro del rodete (D) es 0.9 m y el de la polea donde se acopla el dinamómetro es 0.05 m.

3. OBJETIVOS Y METODOLOGÍA

Los objetivos y metodología seguidos en el presente trabajo han sido:

- Toma exhaustiva de medidas en el banco de ensayos para la correcta caracterización de la turbina. Las medidas abarcan todo el rango de trabajo de la turbina Pelton.
- Análisis y procesado de los datos anteriormente obtenidos para seguidamente realizar un estudio teórico (dimensional) para el procesado y representación de las distintas variables. Tras esto, se lleva a cabo un estudio teórico (adimensional) para igualmente procesar y representar las distintas variables adimensionales.
- Se aplica la teoría de semejanza, para, usando los resultados obtenidos podamos obtener los parámetros de funcionamiento óptimos de otra turbina semejante. Además se procesa y representa la velocidad de embalamiento adimensional así como con el coeficiente de desagüe.
- Finalmente se procede a la realización de una interfaz gráfica (programada en MATLAB) que aúne todos los apartados anteriores.

Otro objetivo no menos importante es que la interfaz gráfica tenga las siguientes características, a saber:

1. Sencilla de utilizar.
2. El usuario ha de poder utilizar los datos obtenidos en los ensayos realizados con el banco de trabajo.
3. Ha de tener un desarrollo secuencial de forma que el usuario se sienta guiado por la interfaz mientras la utiliza.

4. RESULTADOS Y DISCUSIÓN

4.1 Curvas de funcionamiento

4.1.1 Teoría general

Obtención del triángulo de velocidad teórica del agua a la entrada del inyector:

$$\frac{c_1^2}{2} + \frac{P_1}{\rho} + gH_1 = \frac{c_2^2}{2} + \frac{P_2}{\rho} + gH_2$$

(Ec. 4.1.)

Siendo:

c_1 : La velocidad de la partícula fluida en el punto 1.

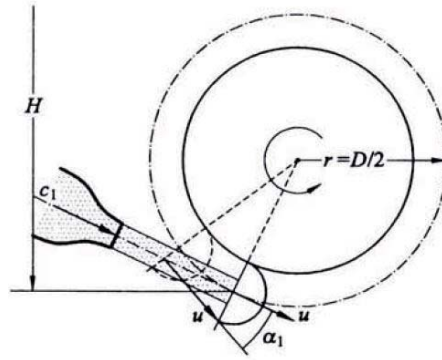
P_1 : La presión de la partícula en el punto 1.

H_1 : La altura del punto 1.

g : El valor de la gravedad terrestre.

ρ : La densidad del fluido de trabajo (en nuestro caso agua).

El subíndice 2 denota la salida del inyector.



En el punto 1 (agua contenida en la presa) la velocidad se desprecia, como la presión es la atmosférica al igual que a la salida del inyector las presiones también se anularán. En el punto 2 (salida del inyector) la velocidad es nuestra incógnita y la altura desaparece ya que es cero por estar en el cero de nuestro sistema de referencia. Obtenemos así:

$$c_1 = \sqrt{2gH}$$

(Ec. 4.2.)

Siendo:

c_1 : La velocidad de la partícula fluida en el punto 1.

H : La altura del punto 1.

g : El valor de la gravedad terrestre.

Debido a las distintas pérdidas en los distintos elementos de la instalación (tubería forzada, inyector y rodete) la velocidad se ve disminuida un valor C_v que es el coeficiente de desagüe.

$$c_1 = C_v \sqrt{2gH}$$

(Ec. 4.3.)

Siendo:

C_v : Coeficiente de desagüe que aminorar la velocidad teórica de salida un factor determinado en función del diseño y rendimientos de los distintos componentes de la turbina. Para nuestro caso el valor mayor obtenido ha sido de 0.68, es decir, el factor

disminuye en un 32% el valor teórico de salida indicándonos así que nuestra turbina no cuenta con un diseño excesivamente bueno.

Para el caso ideal será igual a uno pero comúnmente ronda 0.97-0.99 en las turbinas usadas en la industria. Más adelante veremos que en nuestro caso es mucho más elevado y estudiaremos por qué empleando simulaciones numéricas.

Continuando con las velocidades, para el rodete, tendremos:

$$u = \frac{\pi D n}{60}$$

(Ec. 4.4.)

Con n en rpm.

Siendo:

u: La velocidad tangencial del rodete.

D: El diámetro del rodete.

n: El número de revoluciones.

Como la distancia del eje del chorro al eje del rodete ($r = D/2$) es prácticamente la misma a la entrada y a la salida de la cuchara, se tiene:

$$u_1 = u_2 = u$$

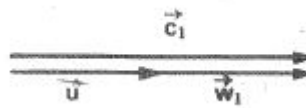
(Ec. 4.5.)

Siendo:

u: La velocidad tangencial del rodete. El subíndice 1 indica la velocidad a la entrada y el subíndice 2 la velocidad a la salida.

Como ya sabemos la turbina Pelton se compone básicamente de un inyector y del rodete. Del inyector sale el agua que impactará contra el rodete y de este obtendremos el par que necesitamos.

Los vectores de velocidad del agua que sale del inyector son:



$$w_1 = c_1 - u$$

(Ec. 4.6.)

Siendo:

c_1 : La velocidad máxima teórica de salida del fluido.

u : La velocidad del rodete.

w_1 : La velocidad relativa entre ambos.

Obtención del triángulo de velocidades de salida.

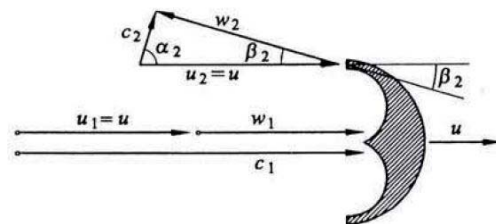


Figura 4.1. Esquema de velocidades de entrada y salida del rodete. Fuente: Mecánica de fluidos incompresibles, Agüera, José.

Para el caso de tener un rodete ideal el cual tiene un ángulo $\beta_2=0^\circ$ los vectores de velocidad de salida son:

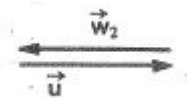


Figura 4.2. Representación de los vectores de velocidad del rodete y velocidad relativa.

Para este caso el agua que impacta en el rodete retorna en la misma dirección que el agua que procede del inyector, pero obviamente, esto no puede darse en la realidad ya que el flujo a la salida de una cuchara podría chocar con el revés de la siguiente. Teóricamente el rendimiento sería máximo y la componente axial de la fuerza que ejerce el agua en el eje del rodete nula.

Por ello el triángulo de velocidad de salida real es:

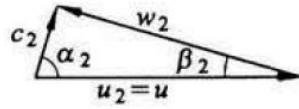


Figura 4.3. Esquema de las velocidades de salida. Fuente: Mecánica de fluidos incompresibles, Agüera, José.

Teóricamente $w_2 = w_1$ pero debido a que existen pérdidas por fricción en las cucharas, tenemos:

$$w_2(\text{real}) = k_w w_1 \quad (k_w < 1)$$

(Ec. 4.7.)

Velocidad absoluta de salida:

La velocidad absoluta de salida w_2 , tangente a la cuchara, y la tangencial u , dan como resultante la velocidad absoluta de salida c_2 que se pierde. Tomando el valor teórico $w_2 = w_1$ se obtiene para el segundo término de la ecuación de Euler:

$$c_2 \cos \alpha_2 = u (1 + \cos \beta_2) - c_1 \cos \beta_2$$

(Ec. 4.8.)

Para una turbina concreta, la velocidad c_2 y en consecuencia el ángulo α_2 sólo varía si modificamos la velocidad de giro, pues β_2 es fijo y c_1 varía con la carga. Si la velocidad, tangencial u es constante, como corresponde a una turbina instalada, el triángulo de salida no varía con la carga (suponiendo c_1 constante). Para más información consultar referencia [3].

4.1.2 Curvas dimensionales

4.1.2.1 Par

El par motor interior podemos obtenerlo puesto que:

$$P_i = \omega M_i = \eta_h \gamma Q H = \eta_h \rho Q c_1^2 / 2$$

(Ec. 4.9.)

Y $u = \omega D/2$, pudiendo obtener:

$$M_i = \eta_h \frac{D \rho Q c_1^2}{4 u}$$

(Ec. 4.10.)

Para obtener el par interior teórico:

$$\eta_h = 2 (1 + \cos \beta_2) \frac{u}{c_1} \left(1 - \frac{u}{c_1}\right)$$

(Ec. 4.11.)

$$M_i = 2 (1 + \cos \beta_2) \frac{u}{c_1} \left(1 - \frac{u}{c_1}\right) \frac{D \rho Q c_1^2}{4 u}$$

(Ec. 4.12.)

Para un caudal Q y una altura H concretos, resulta ser una función lineal de u/c_1 . De manera que el par motor máximo se tiene cuando $u/c_1=0$ (rodete frenado).

Par motor exterior. Este es medido en el banco de ensayos y es también una función lineal hasta $u/c_1 \approx 0.5$. Para valores mayores disminuye cada vez más pronunciadamente, anulándose así para $u/c_1 \approx 0.8$ variando un poco según la carga, como podemos ver en la figura 4.4. Esto se debe a que con valores de u/c_1 mayores que los valores de diseño el chorro no completa su acción sobre las cucharas, y además, el revés de las mismas choca con el agua y la dispersa dentro de la envolvente, aparte de que las pérdidas mecánicas aumentan con la velocidad de giro del rodete.

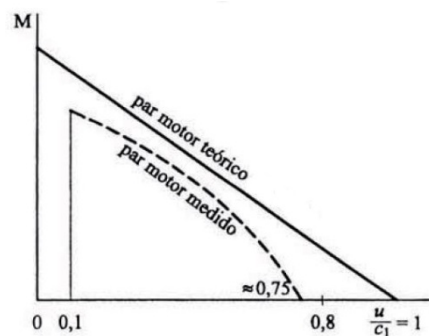


Figura 4.4. Par motor real y teórico frente a u/c_1 . Fuente: Mecánica de fluidos incompresibles, Agüera, José.

De cara la práctica para obtener el par necesitamos la velocidad angular del rodete y la fuerza que medimos en el banco de ensayo:

$$C = \frac{|F_1 - F_2|0.05}{2}$$

(Ec. 4.13.)

Siendo F_1 y F_2 son las fuerzas medidas en el banco de ensayo. El valor 0.05 se corresponde con el diámetro de la rueda que roza en el freno de Prony. Consultar referencia número [3] para obtener mayor información.

4.1.2.2 Potencia efectiva o potencia al freno

La potencia efectiva, potencia al freno y/o potencia exterior en el eje, es igual a la interior, menos la perdida causada por los rozamientos externos, o mecánicos. Esta sólo puede medirse en el banco de ensayos.

Como tenemos el par así como las revoluciones podemos obtenerla:

$$P_u = C \omega$$

(Ec. 4.14.)

Siendo:

C: El par obtenido del ensayo.

ω : Las revoluciones en rad/s.

Como obtenemos revoluciones por minuto habremos de utilizar la siguiente fórmula:

$$P_u = \frac{2 \pi}{60} \omega C$$

(Ec 4.15.)

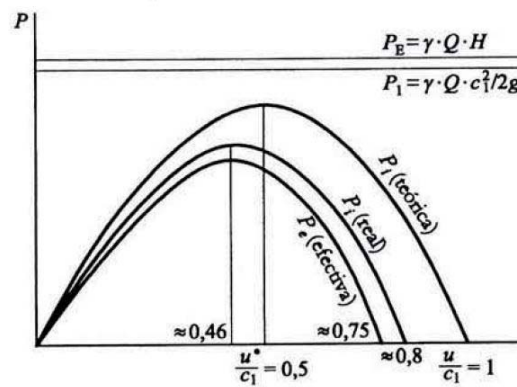


Figura 4.5. Potencia útil teórica y real frente a u/c_1 . Fuente: Mecánica de fluidos incompresibles, Agüera, José.

Como puede comprobarse en la figura 4.5 y como veremos al analizar los resultados obtenidos del ensayo con la aplicación gráfica la potencia útil tiene un máximo para $u/c_1=0.5$

4.1.2.3 Potencia hidráulica

La potencia hidráulica no depende de u/c_1 (teóricamente) de forma que al representarlo en un diagrama Potencia hidráulica-rpm su representación será una línea recta paralela al eje de abscisas. Como en la turbina Pelton no hay pérdidas volumétricas, la potencia recibida por el rodete será:

$$P_H = \rho g Q H$$

(Ec. 4.16.)

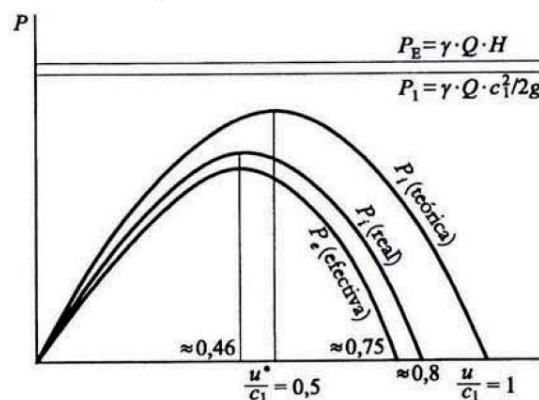


Figura 4.6. Potencia hidráulica real y teórica frente u/c_1 . Fuente: Mecánica de fluidos incompresibles, Agüera, José.

En la figura 4.6 podemos ver cómo la potencia hidráulica no depende de u/c_1 o del número de revoluciones.

4.1.2.4 Rendimiento global

El rendimiento global en las turbinas Pelton reales ronda 0.85-0.91 (Datos obtenidos de la referencia [3]). En nuestra turbina no alcanzamos ni tan siquiera el 0.3 en la mayoría de los casos. Esto se debe a numerosas pérdidas, principalmente en el inyector, que veremos más adelante.

$$\eta_h = \frac{P_u}{P_H} = \frac{C \omega}{\rho g Q H}$$

(Ec. 4.17.)

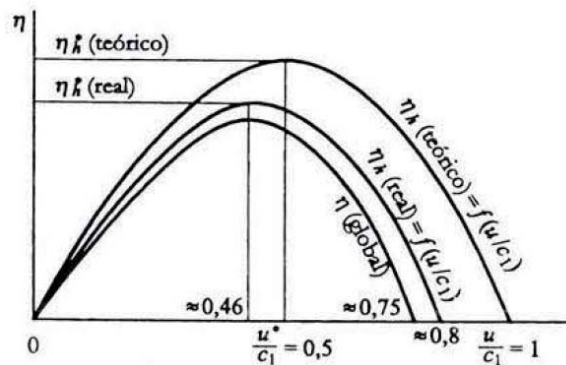


Figura 4.7. Representación de los rendimientos en función de u/c_1 .

4.1.3 Análisis dimensional. Curvas adimensionales

El análisis dimensional enseña un método algebraico sencillo para reducir el número de variables a un conjunto menor de variables adimensionales. Con esto se explica que este método haya llegado a ser un instrumento matemático de trabajo para los experimentadores.

Dado un conjunto de variables, que intervienen en un fenómeno, se denomina conjunto completo de productos adimensionales a un conjunto formado por productos adimensionales de dichas variables, tal que cada producto independiente de los demás, y cualquier otro producto adimensional de las variables especificadas se pueda obtener

multiplicando todos los productos adimensionales, que forman el conjunto, previamente elevados a potencias determinadas.

Teorema de Vaschy-Buckingham:

$$Q = F(\rho, \mu, gH_n, D, \Omega, l_1, l_2, \dots, l_n)$$

$$C = F(\rho, \mu, gH_n, D, \Omega, l_1, l_2, \dots, l_n)$$

$$W_u = F(\rho, \mu, gH_n, D, \Omega, l_1, l_2, \dots, l_n)$$

$$W_h = F(\rho, \mu, gH_n, D, \Omega, l_1, l_2, \dots, l_n)$$

(Ec 4.18.)

Tomamos ρ como unidad de masa, gH_n como unidad de tiempo y D como unidad de longitud.

$$M \quad L \quad T$$

$$\begin{array}{c} \rho \\ gH_n \\ D \end{array} \begin{vmatrix} 1 & -3 & 0 \\ 0 & 2 & -2 \\ 0 & 1 & 0 \end{vmatrix} = 2 \neq 0$$

(Ec 4.19.)

Luego las tres magnitudes son independientes.

$$\Pi_Q = F(\Pi_\mu, \Pi_\Omega, \Pi_{l_1}, \Pi_{l_2}, \dots, \Pi_{l_n})$$

(Ec 4.20.)

$$\Pi_Q \equiv \frac{Q}{\rho^a \cdot (gH_n)^b \cdot D^c} \Rightarrow \left[\frac{L^3}{T} \right] = \left[\frac{M}{L^3} \right]^a \cdot \left[\frac{L^2}{T^2} \right]^b \cdot [L]^c \Rightarrow \Pi_Q \equiv \frac{Q}{D^2 \cdot \sqrt{gH_n}}$$

(Ec 4.21.)

$$\Pi_C \equiv \frac{C}{\rho^a \cdot (gH_n)^b \cdot D^c} \Rightarrow \left[\frac{ML^2}{T^2} \right] = \left[\frac{M}{L^3} \right]^a \cdot \left[\frac{L^2}{T^2} \right]^b \cdot [L]^c \Rightarrow \Pi_C \equiv \frac{C}{\rho \cdot gH_n \cdot D^3}$$

(Ec 4.22.)

$$\Pi_{\mu} \equiv \frac{\mu}{\rho^a \cdot (gH_n)^b \cdot D^c} \Rightarrow \left[\frac{M}{LT} \right] = \left[\frac{M}{L^3} \right]^a \cdot \left[\frac{L^2}{T^2} \right]^b \cdot [L]^c \Rightarrow \Pi_{\mu} \equiv \frac{\mu}{\rho \cdot D \cdot \sqrt{gH_n}}$$

(Ec 4.23.)

La expresión obtenida para Π_Q es correcta, pero es más representativo definir:

$$\Pi_{\Omega}^* = \frac{\Pi_Q}{2\sqrt{2}}$$

(Ec 4.24.)

De forma que:

$$\Pi_{\Omega}^* = \frac{\Omega D}{2\sqrt{2gH_n}}$$

(Ec 4.25.)

Pudiendo ésta ser transformada en:

$$\Pi_{\Omega}^* = \frac{u}{c_{ideal}}$$

(Ec 4.26.)

En el caso de las longitudes que definen la máquina:

$$\Pi_{l1} = \frac{l_1}{D}, \Pi_{l2} = \frac{l_2}{D}, \dots, \Pi_{ln} = \frac{l_n}{D}$$

(Ec 4.27.)

Para todo el ensayo se utiliza la misma máquina de manera que los grupos Π_i permanecerán constantes y no tendrán transcendencia siempre que exista semejanza geométrica.

$$\Pi_C \equiv \frac{C}{\rho^a \cdot (gH_n)^b \cdot D^c} \Rightarrow \left[\frac{ML^2}{T^2} \right] = \left[\frac{M}{L^3} \right]^a \cdot \left[\frac{L^2}{T^2} \right]^b \cdot [L]^c \Rightarrow \Pi_C \equiv \frac{C}{\rho \cdot gH_n \cdot D^3}$$

(Ec 4.28.)

$$\Pi_{W_h} \equiv \frac{W_h}{\rho^a \cdot (gH_n)^b \cdot D^c} \Rightarrow \left[\frac{ML^2}{T^3} \right] = \left[\frac{M}{L^3} \right]^a \cdot \left[\frac{L^2}{T^2} \right]^b \cdot [L]^c \Rightarrow \Pi_{W_h} \equiv \frac{W_h}{\rho \cdot (gH_n)^{3/2} \cdot D^2}$$

(Ec 4.29)

$$\Pi_{W_u} \equiv \frac{W_u}{\rho^a \cdot (gH_n)^b \cdot D^c} \Rightarrow \left[\frac{ML^2}{T^3} \right] = \left[\frac{M}{L^3} \right]^a \cdot \left[\frac{L^2}{T^2} \right]^b \cdot [L]^c \Rightarrow \Pi_{W_u} \equiv \frac{W_u}{\rho \cdot (gH_n)^{3/2} \cdot D^2}$$

(Ec 4.30.)

$$\eta_h = \frac{W_u}{W_h}$$

(Ec 4.31.)

Como podremos ver más adelante al representar las distintas curvas comprobaremos que existe semejanza física para los distintos caudales ensayados. Además como en todo el ensayo es la misma máquina, sabemos que existe semejanza geométrica.

En la referencia [1] encontraremos una demostración más exhaustiva de la demostración de los grupos Pi.

4.1.4 Semejanza

Cuando se ha de recurrir a experimentación porque la teoría no pueda resolver y/o predecir los resultados, se harán ensayos con modelos, a escala reducida generalmente. En nuestro caso podríamos suponer que la turbina del banco de ensayos es el modelo de una turbina mayor que deseamos construir.

En los ensayos de máquinas hidráulicas la fuerza preponderante es la viscosidad. Por tanto, según lo estudiado el modelo y el prototipo, además de ser geoméricamente semejantes, deberían ensayarse a igual número de Reynolds para conservar la semejanza dinámica. En la práctica esto resulta imposible. En los ensayos de turbinas hidráulicas se tropieza con la dificultad de ensayar la turbina modelo bajo el salto requerido por la

igualdad de número de Reynolds en el modelo y el prototipo. De ahí que según la práctica universal: en los ensayos de máquinas hidráulicas se hace la hipótesis de que la semejanza geométrica implica la semejanza mecánica.

Esto equivale a suponer que la viscosidad no entra en juego y por tanto que los rendimientos del modelo y del prototipo son iguales. Aunque en la realidad no sucede así, ha conducido a excelentes resultados, excepto a lo que respecta a predicción de rendimientos. Parte de la información expuesta se ha obtenido de la referencia [3].

Como aclaración para que exista la semejanza geométrica debe darse las siguientes condiciones:

$$\frac{l_p}{l_m} = \lambda \quad \frac{l_p^2}{l_m^2} = \lambda^2 \quad \frac{l_p^3}{l_m^3} = \lambda^3$$

(Ec 4.32.)

Lo que implica que tanto las longitudes, superficies y volúmenes del modelo han de guardar con el prototipo una relación fija (factor de escala).

En resumen las leyes de semejanza sirven para:

1. Predecir el comportamiento de una máquina de distinto tamaño pero geoméricamente semejante, a otra cuyo comportamiento se conoce, trabajando en las mismas condiciones.
2. Predecir el comportamiento de una misma máquina cuando varía alguna de sus características.

Las leyes de semejanza en turbinas hidráulicas son:

Primera ley: Los número de revoluciones son directamente proporcionales a la raíz cuadrada de las alturas netas.

$$\frac{n_p}{n_m} = \frac{\sqrt{H_p}}{\sqrt{H_m}}$$

(Ec 4.33.)

Segunda ley: Los caudales son directamente proporcionales a la raíz cuadrada de las alturas netas.

$$\frac{Q_p}{Q_m} = \frac{\sqrt{H_p}}{\sqrt{H_m}}$$

(Ec 4.34.)

Tercera ley: Las potencias útiles son directamente proporcionales a las alturas netas elevadas a 3/2.

$$\frac{P_p}{P_m} = \left(\frac{H_p}{H_m}\right)^{3/2}$$

(Ec 4.35.)

Cuarta ley: Los números de revoluciones son inversamente proporcionales a los diámetros.

$$\frac{n_p}{n_m} = \frac{D_p}{D_m}$$

(Ec 4.36.)

Quinta ley: Los caudales son directamente proporcionales a los cuadrados de los diámetros.

$$\frac{Q_p}{Q_m} = \left(\frac{D_p}{D_m}\right)^2$$

(Ec 4.37.)

Sexta ley: Las potencias útiles son directamente proporcionales a los diámetros al cuadrado.

$$\frac{P_p}{P_m} = \left(\frac{D_p}{D_m}\right)^2$$

(Ec 4.38.)

En todas las ecuaciones anteriores se ha seguido la siguiente notación:

Para denotar la presión (P), caudal (Q) o diámetro (D) del prototipo se ha añadido un subíndice “p”. Para referirnos al modelo se ha empleado un subíndice “m”. Para mayor información sobre las ecuaciones y su demostración consultar las referencias [1] y [3].

4.1.5 Velocidad específica y diámetro específico

Si operamos con las seis leyes antes vistas y eliminamos los diámetros obtendremos la velocidad específica. Si por el contrario eliminamos las revoluciones tendremos el diámetro específico, así llegamos a los dos siguientes resultados:

$$n_s = n \frac{\sqrt{P_u}}{(H)^{5/4}}$$

(Ec 4.39.)

$$D_s = D \frac{(H)^{3/4}}{\sqrt{P_u}}$$

(Ec 4.40.)

Siendo:

n_s : La velocidad específica de la turbina.

n : El número de revoluciones.

P_u : La potencia útil.

H : La altura disponible.

D_s : Diámetro específico.

D : Diámetro del rodete.

4.4 INTERFAZ GRÁFICA

En este apartado se expondrá pormenorizadamente las distintas interfaces que componen el programa explicándolas una a una.

Pero primero, y siguiendo el orden lógico de la interfaz, vamos a explicar cómo el alumno debe rellenar las tablas con los datos obtenidos en el ensayo.

Dentro de la carpeta de la interfaz gráfica encontramos un archivo llamado “DATOS_Alumno.xls”. Éste es una hoja de Excel como se muestra a continuación:

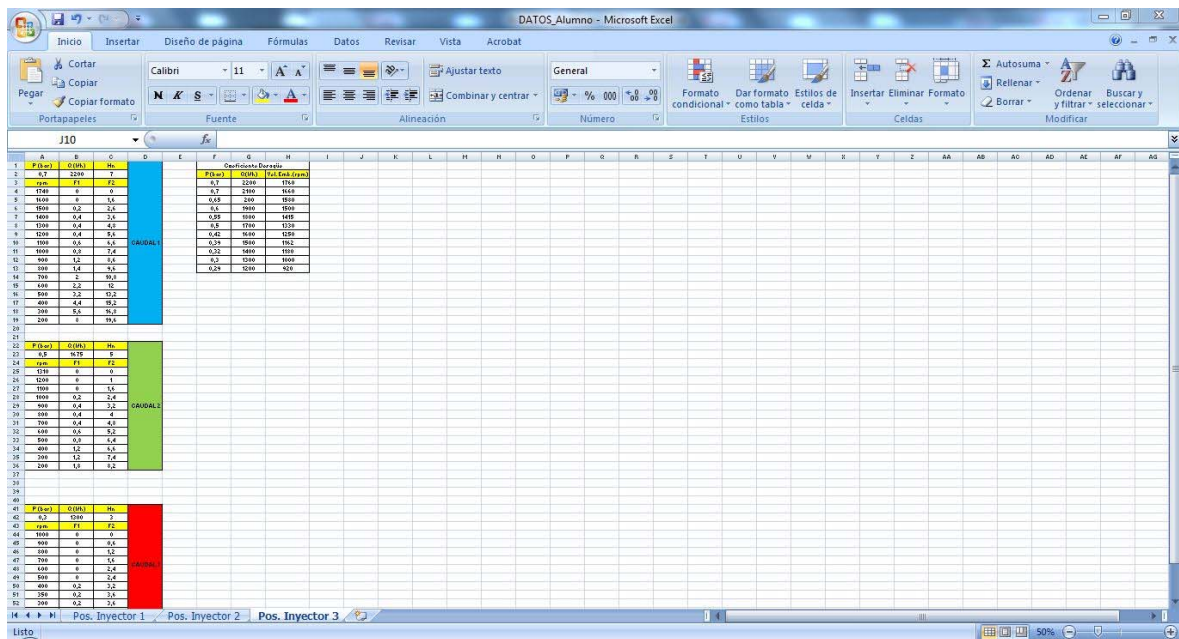


Figura 4.8. Tablas de datos para introducir los valores obtenidos del ensayo.

Vista general. Nótese los distintos colores para los distintos caudales (figura 4.8). En la interfaz gráfica se asocia el caudal 1 con el color azul, caudal 2 con el color verde y el caudal 3 con el color rojo, para que así el alumno relacione más fácilmente los distintos caudales una vez represente todas las gráficas.

P (bar)	Q (l/h)	Hn	CAUDAL 1	Coeficiente Desagüe		
0,7	2200	7		P(bar)	Q(l/h)	Vel. Emb.(rpm)
rpm	F1	F2		0,7	2200	1760
1740	0	0		0,7	2100	1660
1600	0	1,6		0,65	200	1580
1500	0,2	2,6		0,6	1900	1500
1400	0,4	3,6		0,55	1800	1415
1300	0,4	4,8		0,5	1700	1330
1200	0,4	5,6		0,42	1600	1250
1100	0,6	6,6		0,39	1500	1162
1000	0,8	7,4		0,32	1400	1180
900	1,2	8,6		0,3	1300	1000
800	1,4	9,6		0,29	1200	920
700	2	10,8				
600	2,2	12				
500	3,2	13,2				
400	4,4	15,2				
300	5,6	16,8				
200	8	19,6				

Figura 4.9. Vista aumentada de las tablas de datos.

En esta vista más detalla, figura 4.9, podemos ver los datos a rellenar por el alumno. Esta hoja ya rellena corresponde a los datos que tomé en el laboratorio para la ejecución de mi trabajo de fin de grado y que también se adjunta con el programa. De cara a entregar el programa al alumno estas tablas (DATOS_Alumno.xlsx) estarán vacías. Pero

para la correcta ejecución de la interfaz gráfica, para que el tribunal pueda evaluar mi trabajo así como para yo poder trabajar con la interfaz, he incluido en las tablas del alumno los datos que obtuve yo en el laboratorio.

Para cada posición del inyector deberemos anotar: La presión medida en bares que nos facilita el manómetro situado a la entrada del inyector (al rellenar la presión la altura aparecerá automáticamente), el caudal que podemos medir gracias al rotámetro de pie, las revoluciones medidas con el tacómetro y sus correspondientes valores de fuerzas que obtenemos de los dinamómetros.

La tabla de la derecha se ofrece para el cálculo del coeficiente de desagüe a mano por parte de los alumnos.

Ahora se explicará como arrancar la GUI.

La primera opción es dirigirnos a la carpeta donde se encuentra el programa y hacer doble click directamente sobre el archivo “inicio.fig”:

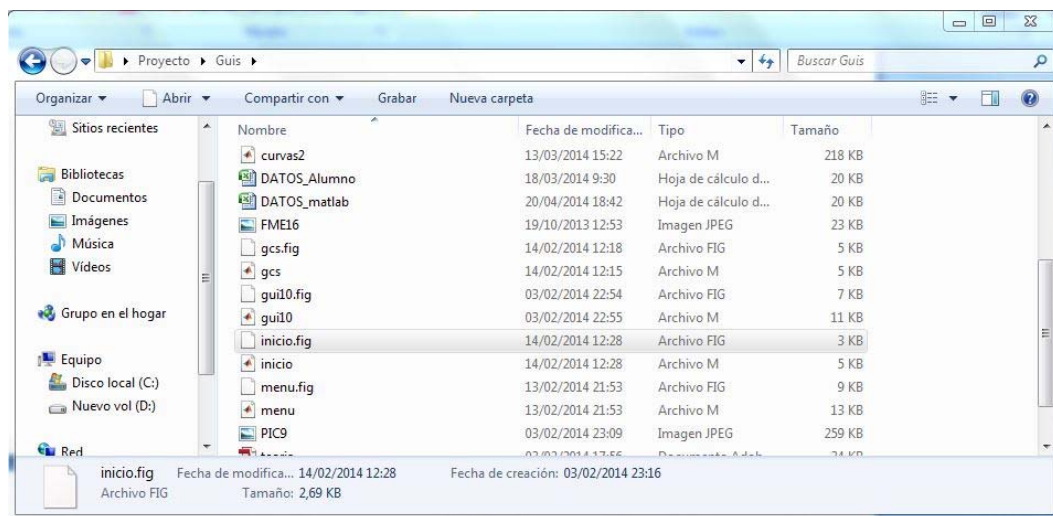


Figura 4.10. Aparece marcado el archivo sobre el que tenemos que hacer doble click.

Mi interfaz esta realizada con la versión 2013 de Matlab, en algunas ocasiones al ejecutar directamente desde el archivo de extensión “.fig” con una versión anterior, puede dar problemas como este:

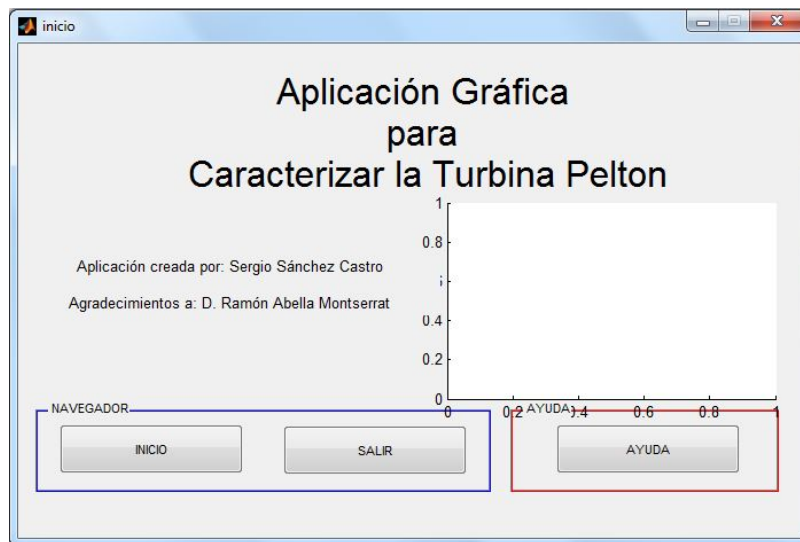


Figura 4.11. Interfaz errónea debida a no ejecutarse el programa correctamente.

Como podemos ver (figura 4.11) aparecen unos ejes que no deberían aparecer en lugar de la imagen de fondo que tendría que representarse.

Para solventarlo tan sólo debemos ejecutar el archivo “inicio.m” abriéndose la siguiente ventana:

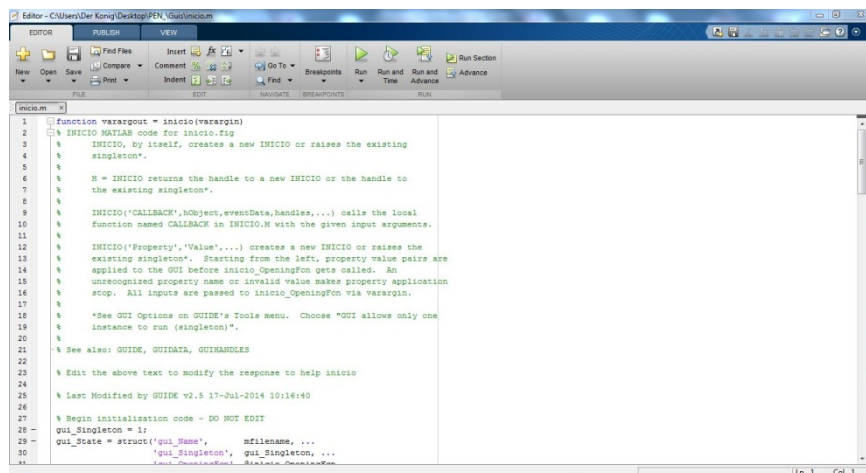


Figura 4.12. Ventana del editor de Matlab.

Para arrancar debemos hacer click en “play” el triángulo verde superior, como cuando ejecutamos un archivo común desde el editor de Matlab. Figura 4.12.

Una vez arrancado correctamente esta es la ventana que se abre cuando abrimos el archivo inicio.fig o, también si ejecutamos el archivo inicio.m y hacemos click and play.



Figura 4.13. Primera interfaz del programa.

Como vemos tiene un diseño sencillo. Se compone del título, autor del TFG, agradecimientos a D. Ramón Abella Montserrat que me facilitó cierta información acerca de una turbina Pelton real (hablaremos de ello más adelante) y de tres botones.

Los tres botones (Inicio, salir y ayuda) nos ayudarán a navegar por la interfaz gráfica. El botón ayuda se repetirá en cada una de las interfaces para la comodidad del usuario.

El botón de inicio dará paso a la siguiente ventana de la interfaz. En ella podremos seleccionar las distintas posibilidades de la misma, acceder al contenido facilitado por Ramón Abella, visionar unos vídeos del inyector, etc...

El botón de salida, como su nombre nos indica, nos permite salir de la aplicación. También es posible abandonarla en cualquier momento haciendo uso de la equis en la esquina superior derecha (como en cualquier otra aplicación).

Si hacemos click sobre el botón de salida aparecerá un cuadro de diálogo que nos pregunta si realmente deseamos salir (figura 4.14).

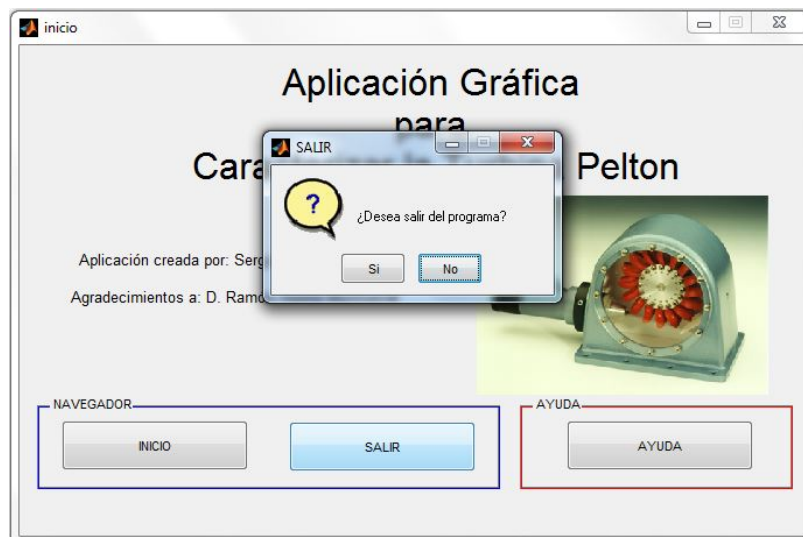


Figura 4.14. La opción por defecto señalada es “No”.

Finalmente el botón de ayuda, incluye la ayuda del programa para guiar al usuario ante las dudas que pudieran aparecer al hacer uso de la aplicación.

Cuando clicamos el botón de “INICIO” aparece una nueva ventana (figura 4.15), que ofrece todas las posibilidades de la aplicación.

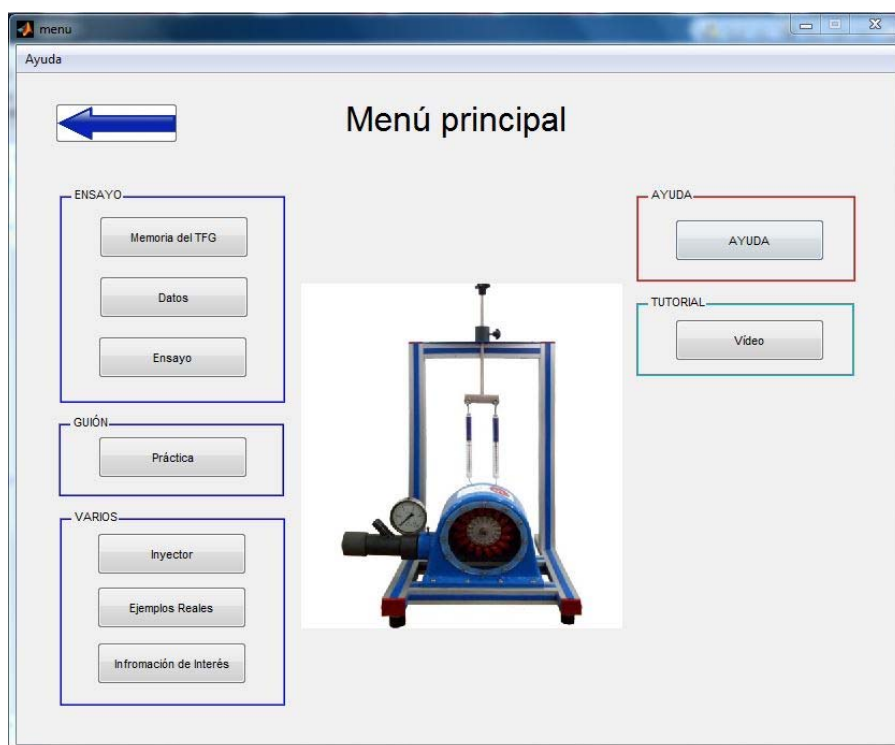


Figura 4.15. Menú principal de la interfaz gráfica.

Menú principal

Como en algunas ventanas no hay espacio para incluir el botón de “AYUDA” se ha incluido también en la barra de tareas que hay en la parte superior de todas las ventanas de la interfaz salvo en la de inicio.

Comenzando por la esquina superior izquierda, destaca una flecha azul con sentido a la izquierda. Esta flecha nos ayuda a navegar hacia atrás por las distintas ventanas. No cambiará de posición apareciendo siempre en la esquina superior izquierda (ya que estamos acostumbrados a encontrarla en otras aplicaciones). Si dejamos el cursor del ratón sobre la flecha aparecerá una ventana con la información del botón: “Atrás”. De esta forma aunque siendo intuitivo se clarifica por completo el uso de este botón.

Si seguimos hacia abajo nos encontramos con un panel de botones titulado “ENSAYO”. Aquí aparecen los botones relacionados con la memoria del TFG así como los relacionados con la toma de datos de la práctica “Datos” y el botón que nos llevará a unas nuevas ventanas para realizar la caracterización de la turbina (“Ensayo”).

Debajo de este panel, sigue uno titulado “Guión” que incluye el guión de la práctica. Tan sólo tenemos que pulsar el botón para que se abra el documento pdf usado para la realización de la práctica.

Más abajo aparece el panel “Varios”. Este incluye los botones titulados: Inyector, Ejemplos reales e Información de interés. Veamos la función de cada uno:

Inyector: Cuando pulsemos este botón se abrirá una nueva ventana:



Figura 4.16. Ventana de la interfaz “Vídeos del inyector”.

Esta ventana (figura 4.16) incluye los botones de “Ayuda” y “Atrás” anteriormente explicados. Su título nos indica “Videos del inyector” y aparecen tres botones diferentes que al pulsarlos abrirán diferentes vídeos del inyector.

El primero (“Vista explosionada”) es un vídeo realizado con un software de diseño paramétrico en 3D. En el se muestra el conjunto del inyector desmontado en vista explosionada y posteriormente se acoplan todos sus elementos.

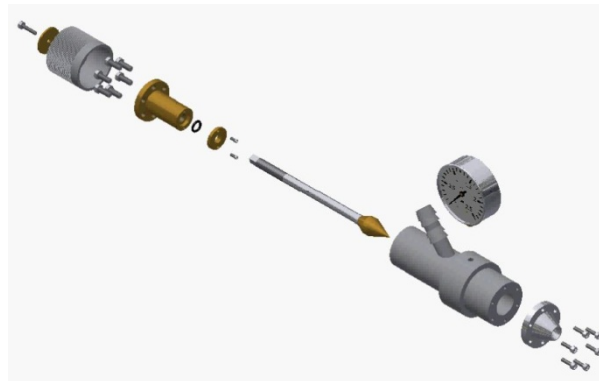


Figura 4.17. Captura de pantalla del vídeo de la vista explosionada.

En el segundo (“Montaje generador de electricidad”) aparece un vídeo en el que encontramos la turbina Pelton con un montaje que permite la producción de energía eléctrica. Acoplado a la turbina se muestran una bombilla, un motor y un multímetro. Puede verse cómo la turbina está generando realmente suficiente energía eléctrica para alimentar los tres elementos. Este montaje se realizó con motivo de unas jornadas de visita a la universidad.



Figura 4.18. Vídeo del montaje. En la lectura del multímetro leemos 0.3 voltios.

En el tercer vídeo, figura 4.19, (“Simulación de llenado del inyector”) aparece una simulación transitoria cualitativa de cómo se produce el llenado del inyector. Se pueden ver las dos fases involucradas (agua en color negro y aire en color blanco). Destaco que pese al corto espacio de tiempo simulado (no suficiente para el completo llenado del inyector) que es de 2 décimas de segundo, la simulación abarcó cinco días completos de cálculo (puede verse la fecha en el vídeo) dejando el ordenador calculando 24 horas seguidas durante los cinco días.

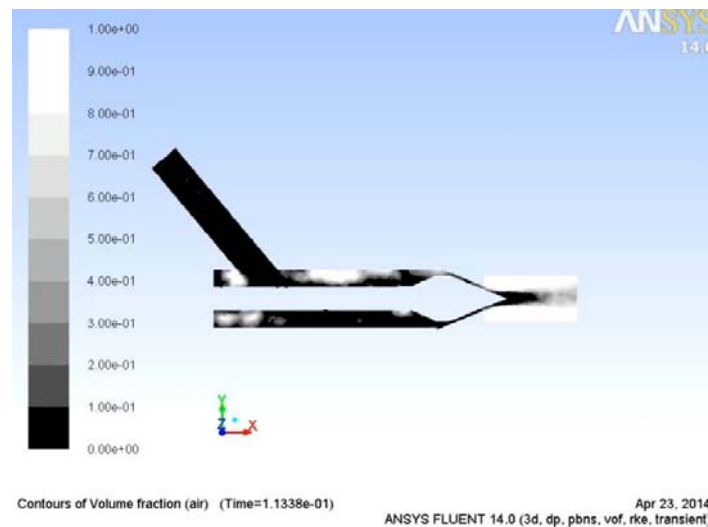


Figura 4.19. Captura de pantalla del vídeo de llenado del inyector.

Las zonas blancas dentro del inyector son burbujas de aire que aparecen en el periodo transitorio de llenado del mismo.

Debemos aclarar que a no ser que dispongamos de un ordenador muy potente, la reproducción de los vídeos, no se realizará correctamente. Esto se debe a que Matlab “lee” uno a uno los fotogramas de cada vídeo y los representa. Esto consume muchos recursos y memoria. Por ello para una reproducción más fluida recomiendo entrar dentro de la carpeta del programa (que contiene los vídeos) y reproducirlos desde ahí.

Retornando a la ventana del “Menú principal” tenemos el botón titulado “Ejemplos reales”. Aquí se muestran imágenes de distintas válvulas y componentes de turbinas Pelton de instalaciones llevadas a cabo en distintos lugares de la tierra, quedando así bien claro, el potencial de este tipo de turbina y su amplio rango de utilización.

Más abajo nos topamos con “Información de interés”. Aquí se adjunta la información que me facilitó D. Ramón Abella Montserrat. Cabe destacar que es Ingeniero Industrial, colaborador adscrito al departamento de ingeniería energética y mecánica de fluidos en la ETS de ingenieros de Sevilla y cuenta con una dilatada experiencia profesional en una afamada compañía de energía eléctrica. La información que me facilitó se corresponde con un estudio de mantenimiento modificativo de la central hidroeléctrica de Pampaneira (Granada).

En la parte derecha de la pantalla veremos el botón de ayuda (no hay nada nuevo que aclarar) y un botón “Tutorial”. No se ha llevado a cabo un vídeo explicativo del funcionamiento de la turbina, por ello se incluye como futura línea de trabajo ya que podría resultar interesante aclarando este vídeo dudas que tenga el alumno antes de entrar al laboratorio. Por ello se deja habilitado el espacio para este vídeo explicativo.

Volvamos al botón de “ENSAYO”. Este es el más importante del programa. Nos dará paso a una ventana (figura 4.20) en la que podremos seleccionar una de estas tres posibilidades:

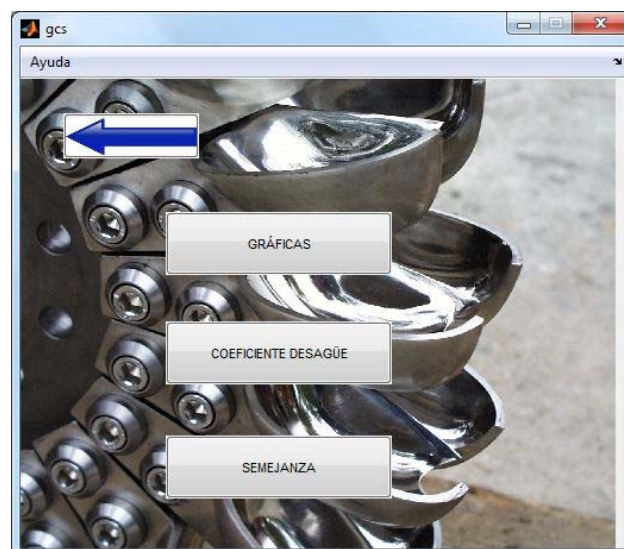


Figura 4.20. Captura de pantalla de la ventana “Gráficas, coeficiente de desagüe y semejanza”.

Gráficas, Coeficiente de desagüe y semejanza.

Analicemos cada una de ellas.

Dentro del botón “GRÁFICAS” aparecerá esta ventana:

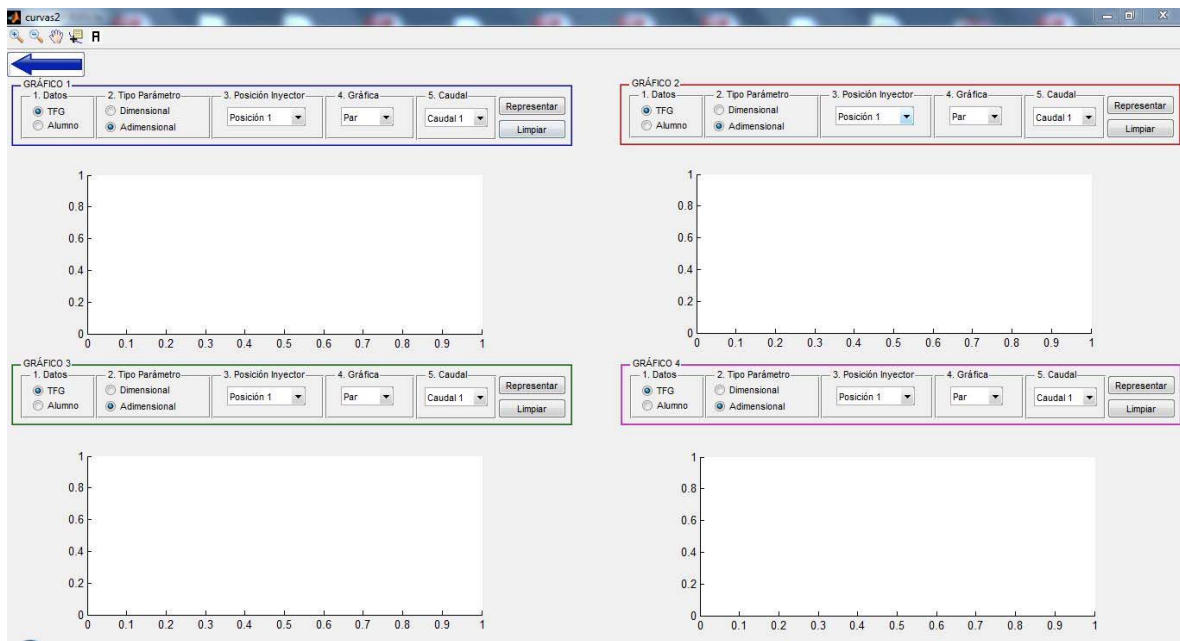


Figura 4.21. Captura de pantalla de la ventana “Gráficas” al iniciarla.

A primera vista vemos que contiene una barra de tareas superior (figura 4.21). En ella se incluyen herramientas para hacer zoom sobre las gráficas que dibujemos así como herramientas para poder mover los distintos gráficos y poder visualizar los valores de los distintos puntos. El icono que contiene una A mayúscula es el botón de ayuda. Veámoslo en mayor detalle (figura 4.22).



Figura 4.22. Botones de la barra de herramientas.

De izquierda a derecha: Zoom acercar, Zoom alejar, Mover gráfica, Cursor para visualizar valores concretos y botón de ayuda.

En la parte inferior vemos cuatro ejes de coordenadas distintos. Cada uno va titulado como “Gráfico X” siendo la X el número de gráfico. Los 4 gráficos realizan las mismas funciones y se componen de los mismos elementos con la misma disposición. Sólo cambia el color de los marcos para mejorar la visualización. Como todos son iguales estudiaremos más de cerca uno de ellos.

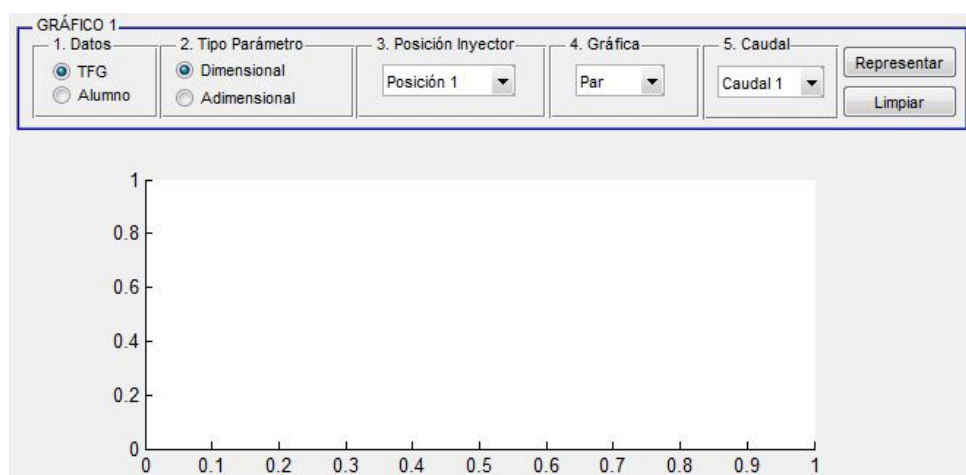


Figura 4.23. Vista aumentada de uno de los espacios de trabajo de la ventana “Gráficas”.

Vemos (figura 4.23) cómo existe una numeración dentro del recuadro de “GRÁFICO 1”. Esta es orientativa para que el usuario se sienta guiado y no se pierda con la multitud de parámetros que podemos seleccionar.

En el recuadro “1.Datos” podemos seleccionar dos opciones distintas excluyentes: TFG o Alumno. Si seleccionamos TFG todos los cálculos se realizarán con los datos que obtuve al realizar mis ensayos. En cambio, si seleccionamos Alumno, los cálculos se realizarán con los datos que el alumno haya tomado en el laboratorio.

En el siguiente recuadro “2.Tipo de parámetro” Tenemos las opciones: Dimensional y Adimensional. Estas nos permiten seleccionar el tipo de parámetros a trabajar como ya vimos en los apartados 4.1.2 Curvas dimensionales y 4.1.3 Curvas adimensionales.

El recuadro “3.Posición del inyector” (figura 4.24) Permite elegir la posición del inyector que queremos graficar. Esto se corresponde con las distintas pestañas del documento Excel a rellenar por el alumno. Cada pestaña es una posición del inyector.

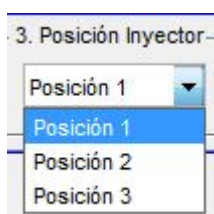


Figura 4.24. Vista aumentada del menú desplegable de la posición del inyector.

El recuadro “4.Gráfica” (figura 4.25) Nos permite seleccionar la variable dimensional o adimensional: Par, potencia útil, potencia hidráulica o rendimiento.

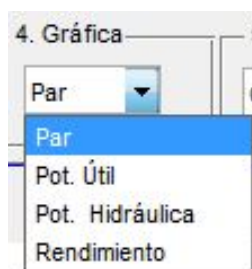


Figura 4.25. Vista aumentada del menú desplegable de la pestaña gráficas.

Finalmente el recuadro “5.Caudal” (figura 4.26) permitirá que elijamos uno o todos los caudales ensayados para una misma posición del inyector.

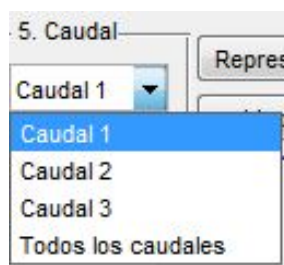


Figura 4.26. Vista aumentada del menú desplegable caudales.

Una vez seleccionados las distintas variables tan sólo tendremos que pulsar “Representar” y aparecerá la gráfica o gráficas que hayamos elegido (figura 4.27).

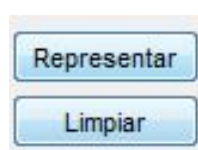


Figura 4.27. Vista aumentada de los botones representar y limpiar ejes.

Cuando queramos eliminar lo representado sólo tendremos que hacer click dos veces sobre limpiar y los ejes de coordenadas quedarán limpios para poder representar de nuevo.

Ahora que ya sabemos cómo funciona esta ventana explicaremos el por qué se ha estructurado en cuatro “Gráficos” distintos. Las variables a representar son cuatro, a saber: Par, potencia útil, potencia hidráulica o rendimiento. Con cuatro ejes de coordenadas distintos el alumno puede representar las cuatro variables a la vez identificándolas de un

solo golpe de vista. Además puede comparar sus datos con los que he obtenido yo. Otra ventaja es que podemos representar la misma variable pero para distintas posiciones del inyector y ver como varía. O podemos representar la misma variable o varias en el mismo o en varios ejes a la vez. Como vemos las posibilidades son muy amplias y en función de las necesidades el alumno podrá “jugar” mucho con la aplicación.

Algunos ejemplos de las posibilidades de esta ventana:

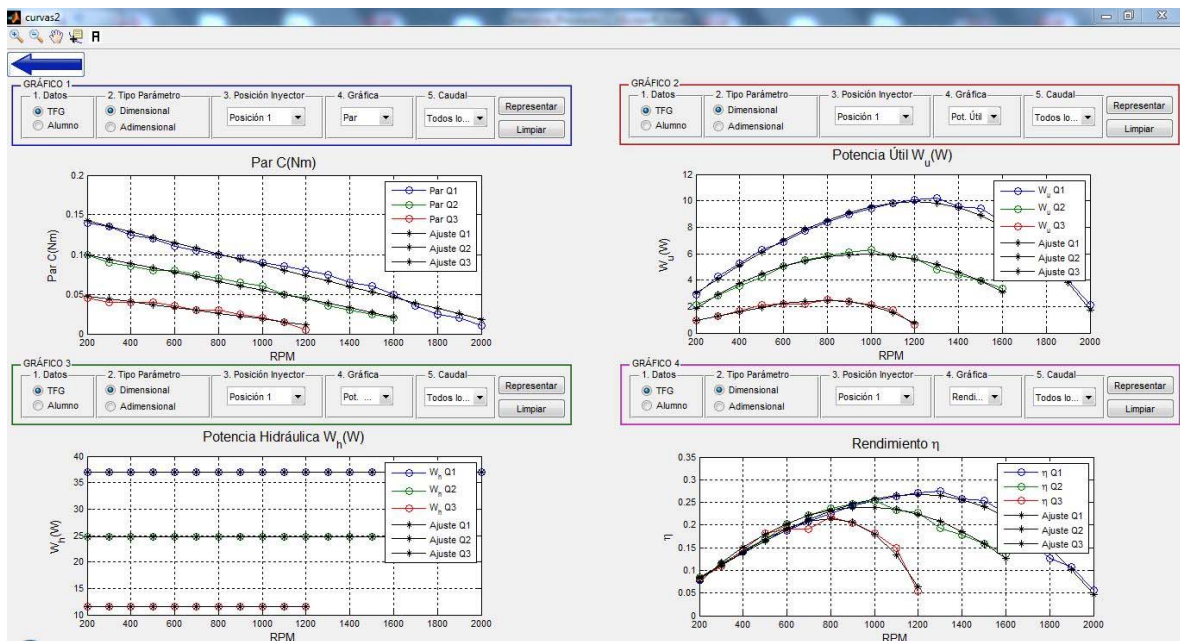


Figura 4.28. Figura que muestra representados todos los caudales de las distintas gráficas para la misma posición del inyector

Aquí (figura 4.28) se representan en cada uno de los ejes coordenados: Par en gráfico 1, Potencia útil en gráfico 2, Potencia hidráulica en gráfico 3 y Rendimientos en gráfico 4.

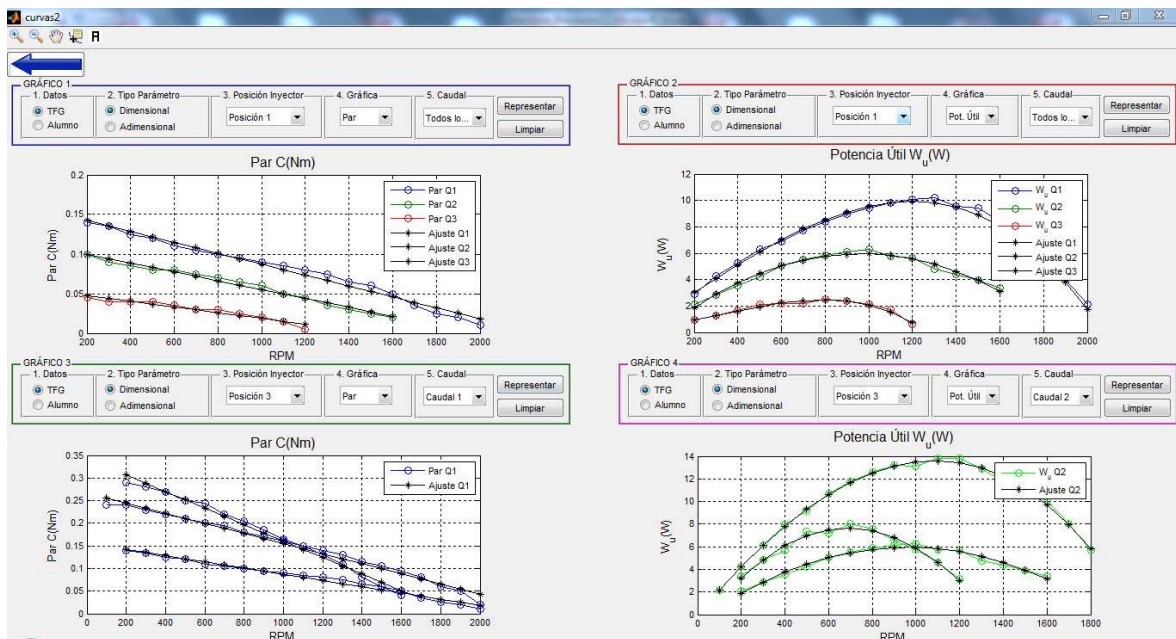


Figura 4.29. Figura que muestra el potencial de la ventana “Gráficas”.

En esta otra figura (figura 4.29) se muestran en los gráficos superiores para una misma posición del inyector par y potencia útil. En los gráficos de la parte inferior tenemos el mismo caudal pero se ha ido variando las posiciones del inyector (por ello las curvas comparten el mismo color).

Ahora analicemos con más detenimiento los resultados que obtenemos en esta ventana de la interfaz gráfica.

Para el par obtenemos:

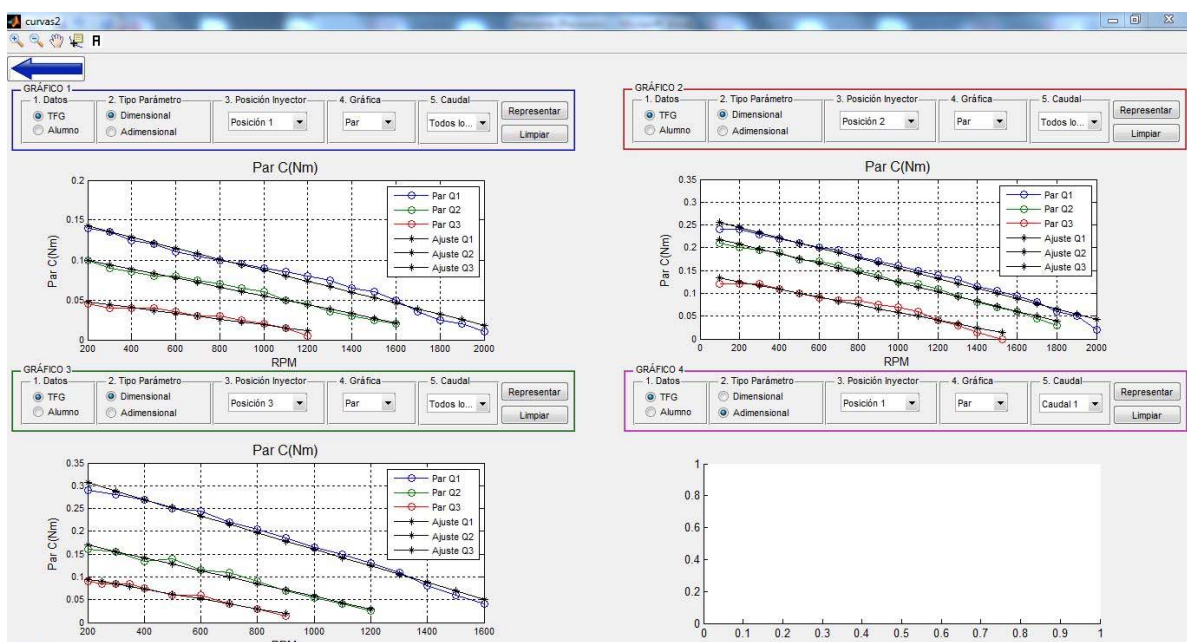


Figura 4.30. En la figura se representa el par para todos los caudales y cada una de las posiciones del inyector.

Como vimos en teoría el Par frente a Rpm devuelve una curva lineal. En general cuanto mayor son las revoluciones menor es el par obtenido para todos los casos. Esto es así ya que estamos ejerciendo menor fuerza en el eje de la turbina con el freno de Prony.

Para el primer ensayo (gráfico 1, figura 4.30) la presión a la entrada del inyector era muy alta. Esto nos indica que la aguja del inyector se encontraba en una posición cercana al cierre y por ello tenemos poco caudal, obteniendo así valores menores para el par. Si lo comparamos con el gráfico 3 para el que la posición de la aguja era más abierta (con un caudal mayor) se comprueba como los valores de par son mayores llegando a duplicarlos. Como vemos para todas las curvas es común que para un menor caudal menor es el par obtenido.

En cuanto a la potencia útil:

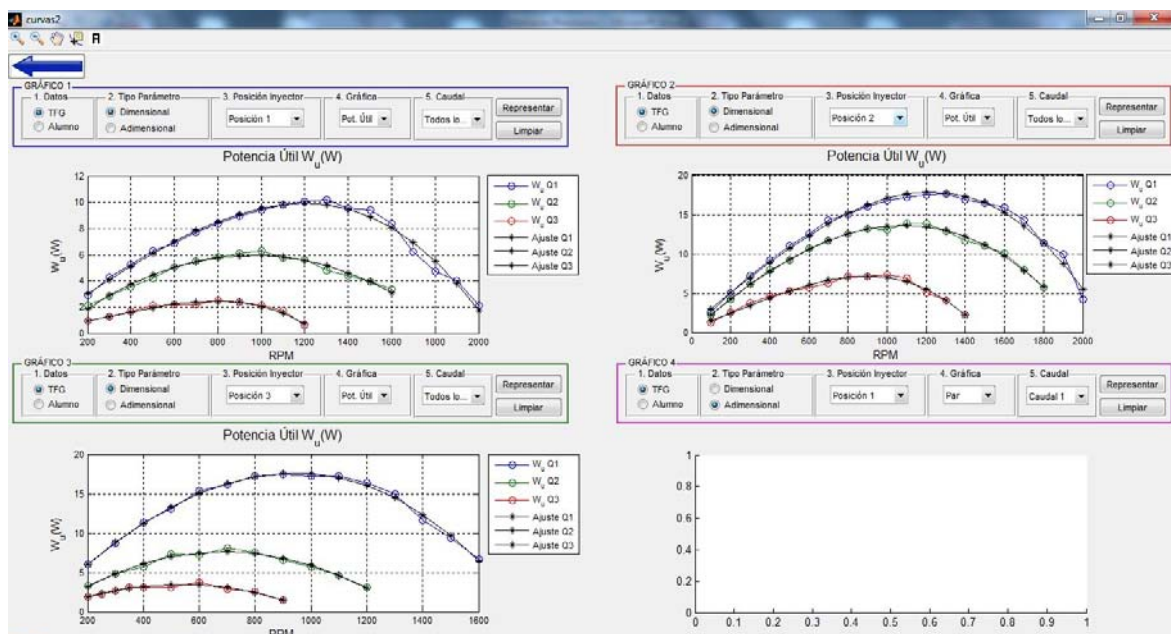


Figura 4.31. En la figura se representa la potencia útil para todos los caudales y cada una de las posiciones del inyector.

En el apartado de curvas dimensionales se demuestra cómo la potencia obtenida es un paraboloide. Puede apreciarse cómo la caída de potencia a un menor número de revoluciones es más acusada. Esto se debe a las pérdidas mecánicas de la turbina. También puede apreciarse la caída a altas revoluciones ya que las pérdidas mecánicas aumentan con

la velocidad de giro de la turbina pero se ve minorado debido a la mayor energía que el agua aporta al sistema.

Al igual que antes donde menos potencia obtenemos es en el gráfico 1 (figura 4.31) ya que es el que menor caudal tiene. Por el contrario la potencia en el gráfico 3 es cercana a los 20 vatios con un caudal mayor. Puede observarse para los caudales 1 obtenemos mayores resultados que para los caudales 3. Esto es debido al mayor caudal (mayor cantidad de movimiento) durante el ensayo.

Potencia hidráulica:

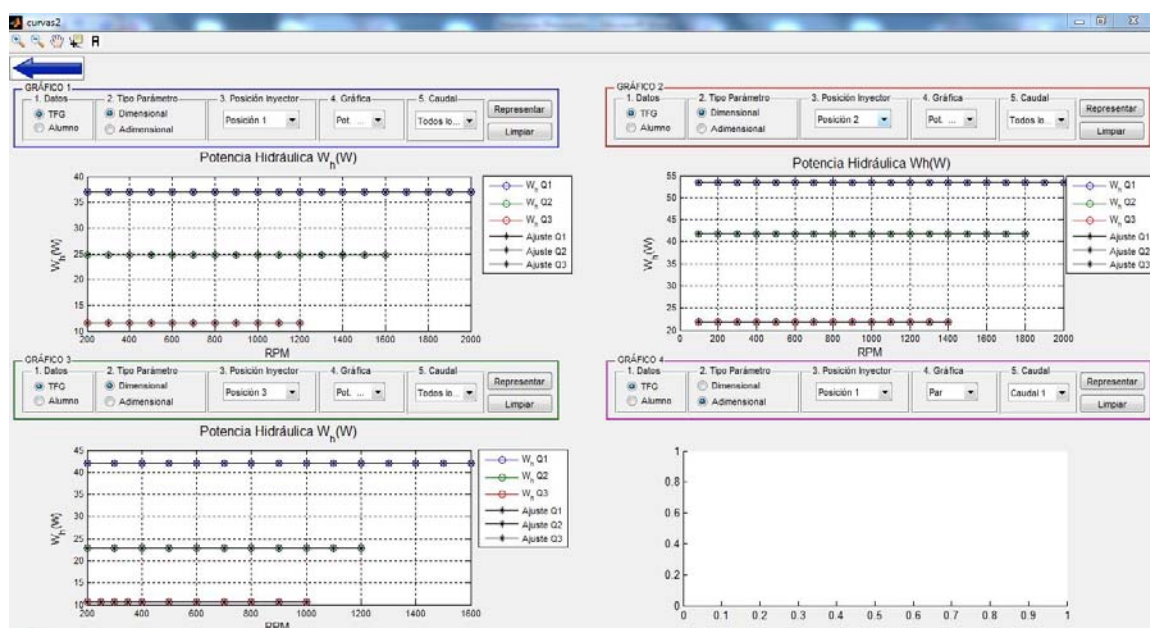


Figura 4.32. En la figura se representa la potencia hidráulica para todos los caudales y cada una de las posiciones del inyector.

Debido a que la cota de agua en teoría no varía (suponemos un embalse siempre lleno) y que al realizar el ensayo la altura que nos proporciona la bomba no varía la potencia que podremos obtener el fluido será constante para cualquier revolución en la que nos encontremos. Como vemos (figura 4.32) a mayor caudal mayor es la potencia que podemos obtener dentro de una misma posición del inyector.

Finalmente para los rendimientos tenemos:

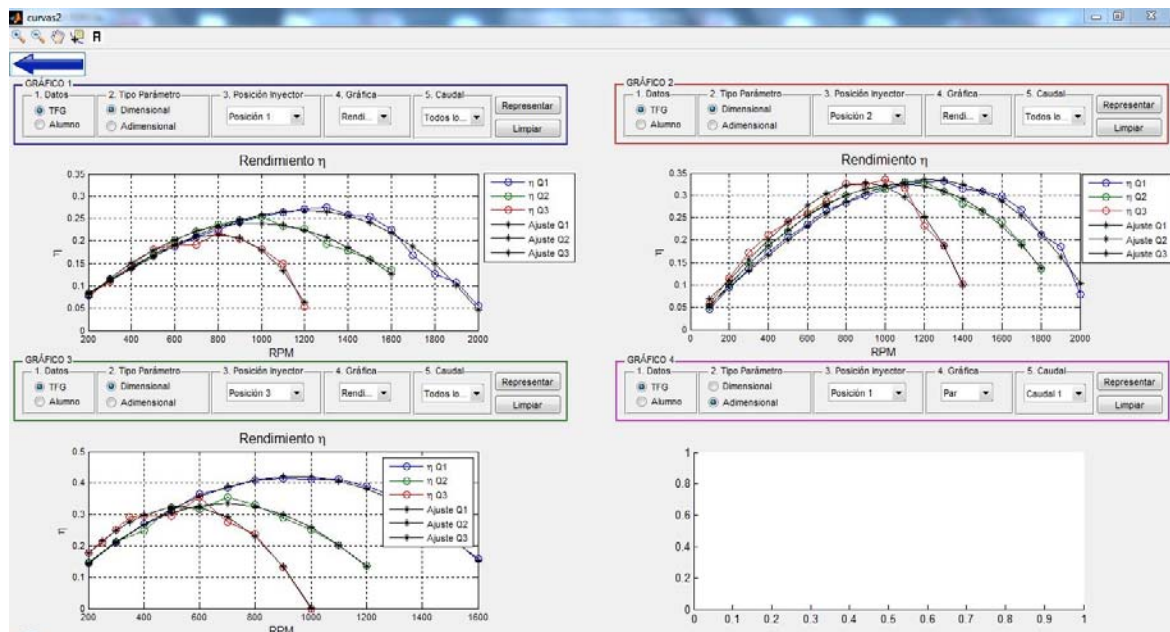


Figura 4.33. En la figura se representa el rendimiento para todos los caudales y cada una de las posiciones del inyector.

Tiene sentido que el par máximo se sitúe en el rango medio de revoluciones (figura 4.33) ya que es en este donde los valores de par y revoluciones son “mayores” con respecto a los de los extremos donde: O tenemos un valor de rpm muy alto y un par muy bajo o tenemos un par alto pero valores bajos de rpm.

En las turbinas Pelton reales usadas en la industria los rendimientos comúnmente alcanzados rondan el 0.85-0.91. Nosotros para nuestra turbina en el mejor de los casos llegamos al 0.41. Esto me llevó a observar con más detenimiento los distintos elementos que componen la turbina para hallar cual de ellos es el causante de unos rendimientos tan bajos. Obviando los componentes tales como juntas, rodamientos y otros elementos normalizados me centré en el rodete y el inyector.

El rodete pese a ser de construcción en plástico tiene un buen diseño si lo comparamos con rodets reales y con las directrices de diseño que podremos consultar en la bibliografía. Me sorprendió especialmente que las cucharas pese a ser de un tamaño tan reducido cuentan con una correcta cresta para dividir el chorro en dos. Algunos diseños análogos a nuestra turbina obvian esta cresta o es muy pequeña para así ahorrar costes y ser las cucharas de construcción más sencilla.

Por el contrario el inyector con un pequeño análisis visual podemos ver diversas razones por las que sería el causante de los pobres rendimientos obtenidos. El análisis del inyector se lleva a cabo en el apartado “4.5 Estudio del inyector”.

Nótese como gracias a contar con diversos gráficos dentro de una misma ventana hemos podido comparar al unísono los distintos gráficos para las distintas posiciones del inyector.

Si volvemos a atrás veremos la ventana que nos permite elegir entre: gráficos, coeficiente de desagüe y semejanza. Ahora analizaremos la ventana de coeficiente de desagüe.

Al pulsar el botón correspondiente aparece esta ventana:

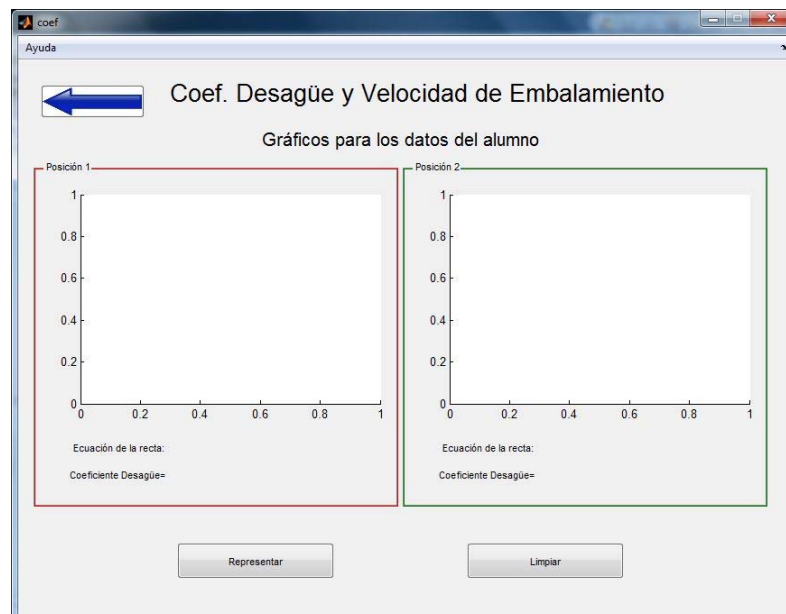


Figura 4.34. Ventana “Coeficiente de desagüe” al ser iniciada.

En ella (figura 4.34) aparecen los botones que ya estamos acostumbrados a ver: Botón de atrás, representar y limpiar gráficos. Además la ventana especifica que las gráficas que vamos a obtener sólo son de los datos que el alumno introduce.

Una vez representados tendremos (para este ejemplo con los datos que tomé en el laboratorio, figura 4.35):

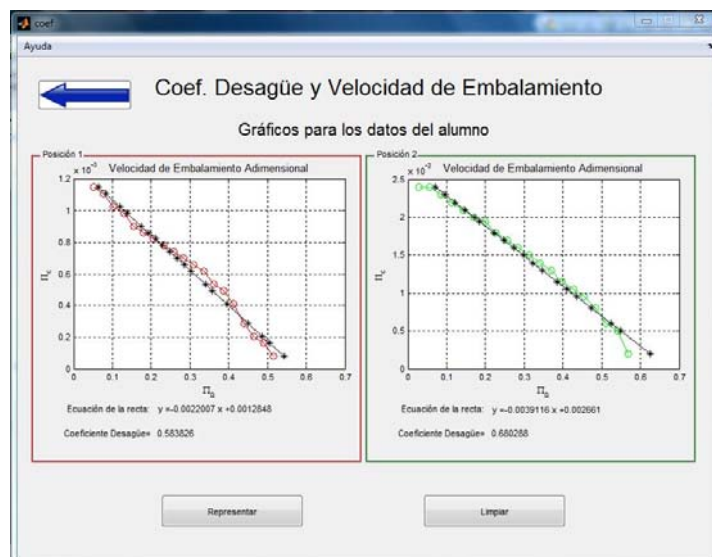


Figura 4.35. Coeficiente de desagüe y recta de ajuste ya representados.

En los ejes es representado el adimensional de par frente al adimensional de las revoluciones. Vemos cómo las dos rectas se adecúan bastante bien al ajuste. Estas gráficas se corresponden con la posición 1 y 2 del inyector como puede verse en las esquinas superiores izquierdas de cada gráfico. Además del coeficiente de desagüe que como vemos es mayor para la posición 2 con un mayor caudal (esto nos indica que la turbina funciona mejor para unos caudales altos que para caudales bajos) también se calcula la ecuación de la recta del ajuste de los datos obtenidos. El coeficiente de desagüe se corresponde con el corte de la recta de ajuste con el eje de abscisas.

Finalmente tenemos la ventana de “Semejanza” (figura 4.36).

Figura 4.36. Ventana “Semejanza” al ser iniciada.

Esta ventana se divide en dos mitades: La superior para realizar el cálculo por semejanza de turbinas con igual diámetro (a la ensayada) pero con alturas disponibles distintas. Y la inferior que nos permite obtener el cálculo de las distintas variables por semejanza cuando tenemos un mismo salto para dos turbinas con diámetros de rodete distintos.

En cualquiera de los dos casos al calcular el programa devuelve los valores máximos teóricos de revoluciones, caudales y potencias.

Aquí vemos un ejemplo:

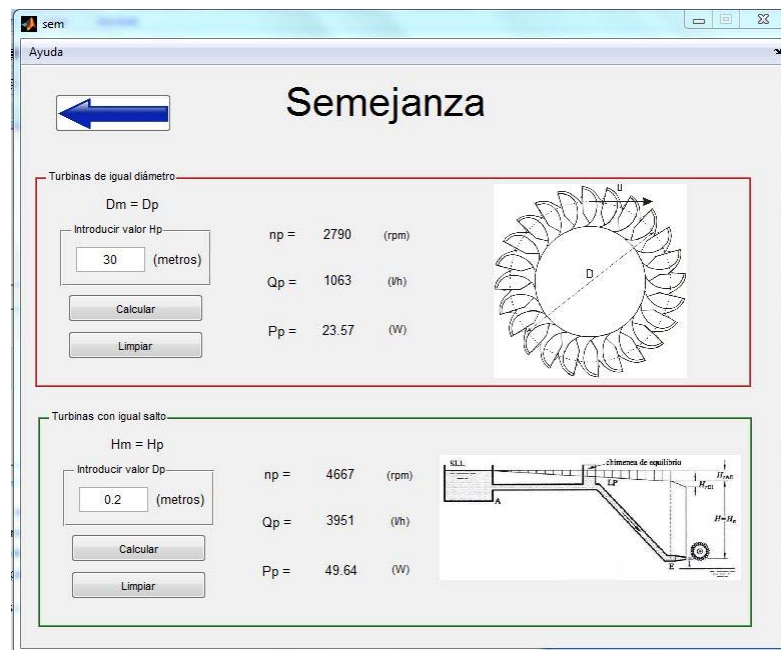


Figura 4.37. Ventana de “Semejanza” mostrando un ejemplo de su funcionamiento.

Para una altura de 30 metros (figura 4.37) obtendríamos en la turbina prototipo un valor de revoluciones de 2790, un caudal de 1063 l/h y una potencia de 23.57 W.

Para el caso de una turbina prototipo de diámetro de rodete 0.2m (la nuestra tiene un diámetro de 0.09 m) tendríamos: 4667 rpm, un caudal de 3951 l/h y 49.64 W.

Como vemos estos valores guardan relación y sentido con los obtenidos en el banco de ensayos. Bien es cierto que si introducimos valores muy grandes o muy pequeños en relación a las alturas y al diámetro real de la turbina obtendremos unos resultados inverosímiles. Por ejemplo si introducimos el valor de 1 m para el rodete de la turbina

prototipo resultará un valor de 23000 rpm con una potencia de 1241 W, claramente esto no es posible. Para una turbina de 1 metro de diámetro se pueden obtener potencias mucho más altas. Debemos estos resultados, no a un fallo del programa o mal aplicación de las leyes de semejanza, sino a la suposición que hicimos al deducirlas. Supusimos semejanza geométrica y mecánica. Esto es así para prototipos dentro de un rango de escala no muy lejano al modelo pero cuando nos alejamos y el factor de escala es mayor, los efectos viscosos así como las pérdidas mecánicas si juegan un papel importante, distorsionando los valores que obtendríamos realmente.

4.5 ESTUDIO DEL INYECTOR

4.5.1 Descripción del inyector

Uno de los principales elementos de la turbina Pelton es el inyector (véase el apartado 2.2 Componentes de la turbina Pelton). Juega un papel muy importante en el rendimiento final de la turbina, y el obtener con los datos tomados en el laboratorio, unos rendimientos tan bajos me llevó a pensar que la razón de estos podría estar en un diseño poco optimizado del inyector de nuestra turbina de ensayos.



Figura 4.38. Representación 3D del inyector con todos sus componentes.

En la figura 4.38 podemos ver la representación tridimensional del conjunto del inyector con todas sus piezas y el manómetro.

Para mayor detalle se incluyen fotografías de todos los componentes en el estado en el que estaban conforme se fueron desmontando.

Cuerpo:



Figura 4.39. Cuerpo principal del inyector.

Este es el cuerpo del inyector (figura 4.39). Aparecen también en la figura la arandela y el tornillo de sujeción del mando que nos permite variar la posición de la aguja. Exteriormente se encuentra en buen estado. Cabe destacar el gran ángulo que forma la entrada de la toma de agua con la horizontal (50°). Este gran ángulo como se verá más adelante provoca circulaciones del agua dentro del inyector y en consecuencia en lugar de tener un flujo mas laminar tendremos un flujo muy turbulento.

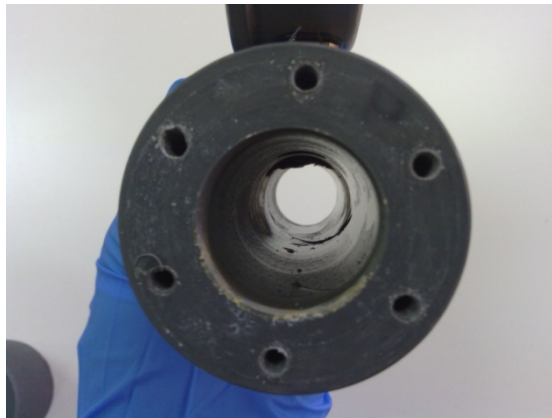


Figura 4.40. Vista interior del cuerpo principal.

En la parte interior si nos fijamos veremos unas marcas (figura 4.40). Estas marcas parecen desgarros en el plástico probablemente producidas en la fabricación de la pieza. Por desgracia este tipo de irregularidades provocan recirculaciones del fluido de trabajo que hacen que la eficiencia del inyector se vea disminuida.

Tobera de salida:



Figura 4.41. Vista superior de la tobera de salida.

Esta pieza (figura 4.41) es de vital importancia ya que es la última parte que el agua toca antes de abandonar el inyector. El diámetro interior es el máximo posible para la salida del agua. Como vemos tras sucesivos ensayos, cal y otras impurezas, se van sedimentando a la salida. Estas impurezas aumentan la rugosidad de las paredes e incrementan las pérdidas en el inyector ya que las impurezas aumentan el rozamiento del agua con el metal.



Figura 4.42. Vista inferior o interna de la tobera.

Esta es la parte posterior de la misma pieza (figura 4.42). Es la parte que el agua “ve” al ser expulsada. Apreciamos que la superficie no está perfectamente acabada y de nuevo vemos muchas impurezas adheridas en las paredes. Esta es otra razón por la que tenemos unos rendimientos bajos.

Aguja:



Figura 4.43. Vista en alzado de la aguja.

En esta figura (figura 4.43) tenemos una vista global de la aguja del inyector. Aparecen también la junta tórica y la tapa del mecanismo de movimiento de la aguja. Se aprecian las formas poco hidrodinámicas que tiene la cabeza de la aguja. Todas las formas con aristas vivas provocan pérdidas y circulaciones del fluido ayudando así a empobrecer el rendimiento.



Figura 4.44. Vista detallada de la argolla excéntrica de la aguja.

En la base de la cabeza de la aguja (figura 4.44) existe una especie de argolla que rodea al vástago. Esto es especialmente dañino de cara al flujo que ha de circular alrededor de la aguja como se verá perfectamente en la simulación numérica. Pero además está mal ejecutada y puede apreciarse la gran excentricidad que tiene la argolla. Aunque no se puede apreciar en las fotos el conjunto cabeza-vástago también tiene una pequeña excentricidad que provoca que la zona de descarga de agua no sea un toroide perfecto obteniendo de esa forma un chorro irregular.



Figura 4.45. Vista en alzado de la aguja y la base de la misma.

En esta figura (figura 4.45) vemos la aguja montada en el soporte que permite el desplazamiento de la misma.

El resto de elementos (tornillería) no influyen en los rendimientos del inyector. En el caso de la junta tórica si estuviese en mal estado podría haber fugas de agua que contribuirían a un mal funcionamiento pero en este caso la junta está en perfectas condiciones.

4.5.2 Caracterización de la aguja del inyector

Hasta ahora la selección de la posición de la aguja del inyector se hacía por tanteo. No se disponía de ninguna forma de seleccionar una posición concreta para llevar a cabo los ensayos. Esto presentaba inconvenientes como el no saber si las distintas posiciones de trabajo que habíamos tomado estaban muy cerca o lejos entre sí, o por ejemplo, en el caso de movimiento accidental de la aguja a mitad del ensayo provocaba tener que repetir todas las medidas desde el principio. Para que esto no ocurra más he llevado a cabo la caracterización de la aguja del inyector.

Comencemos definiendo los elementos de los que disponemos:



Figura 4.46. Vista en detalle del cabezal de la aguja.

La aguja cuenta con aristas vivas (figura 4.46) así que para obtener la función del área de paso de agua no tendremos que hacer ninguna simplificación.

Otra característica importante es el tipo de rosca de la aguja que es la que nos marca el avance de la misma al girar el mando regulador.



Figura 4.47. Vista de la rosca de la base de la aguja.

La rosca es de diámetro 10 y paso 1.5 mm (figura 4.47). Esto nos servirá para conocer la posición de la aguja en cada momento.

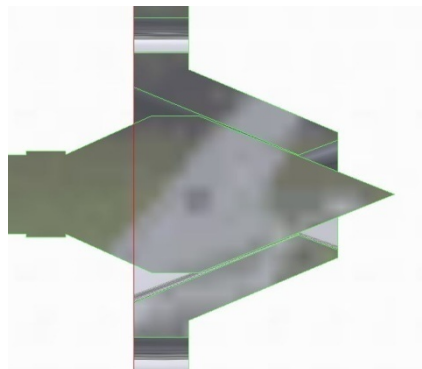


Figura 4.48. Vista seccionada de las piezas aguja y tobera en su posición cerrada.

En la figura 4.48 aparecen la tobera de salida de agua y el cabezal de la aguja. Vista en corte de la posición más adelantada de la aguja (cerrada completamente).



Figura 4.49. Vista real del conjunto en la turbina. Posición más adelantada de la aguja.

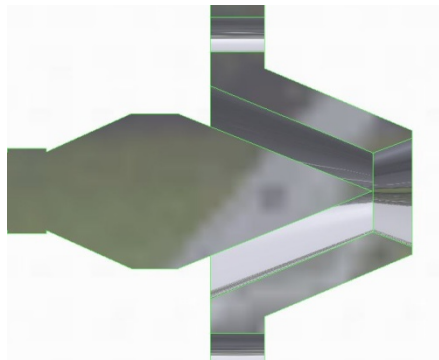


Figura 4.50. Vista seccionada de las piezas aguja y tobera en su posición más atrasada.

Posición más atrasada (figura 4.50). No tiene sentido una posición más atrás de la aguja ya que la sección de paso no será mayor al diámetro interior de la tobera.

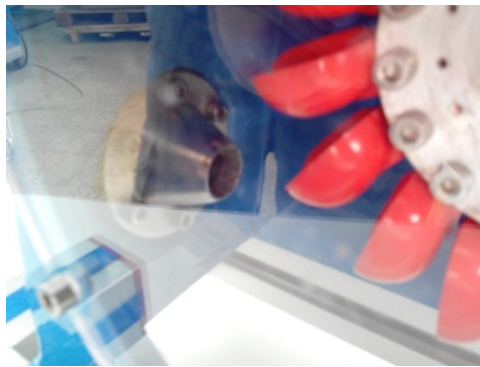


Figura 4.51. Vista real del conjunto en la turbina. Esta es la posición más atrás de la aguja.

Primero obtengamos el recorrido de la aguja en el eje horizontal. Para ello necesitamos el paso de la rosca que sabemos es de 1.5. Esto nos indica que por cada vuelta completa habremos adelantado o atrasado la aguja 1.5mm. Para distintos ángulos obtenemos el avance:

$$1 \text{ revolución} \rightarrow 360^\circ = 2\pi \text{ radianes} \rightarrow x = 1.5 \text{ mm}$$

$$1/2 \text{ revolución} \rightarrow 180^\circ = \pi \text{ radianes} \rightarrow x = 0.75 \text{ mm}$$

$$1/4 \text{ revolución} \rightarrow 90^\circ = \frac{\pi}{2} \text{ radianes} \rightarrow x = 0.375 \text{ mm}$$

$$1/8 \text{ revolución} \rightarrow 45^\circ = \frac{\pi}{4} \text{ radianes} \rightarrow x = 0.1875 \text{ mm}$$

(Ec. 4.41.)

Siendo x el desplazamiento longitudinal de la aguja cuando accionamos el mando regulador.

Ya sabemos el avance de la aguja pero ¿Cómo saber en qué posición se encuentra en cada momento?



Figura 4.52. Vista de la turbina antes de incluir la escala graduada en el mando regulador.

Antes no era posible conocer con exactitud la posición de la aguja ya que de fábrica el inyector no cuenta con ningún tipo de escala en el mando regulador de la aguja (figura 4.52).

Este problema se resolvió haciendo unas marcas en el cuerpo del inyector que nos indican el número de vueltas completas que hemos dado y se hizo una pegatina para la parte posterior del mando regulador para contar las fracciones de vuelta.

Las marcas se realizaron con la ayuda de un torno.



Figura 4.53. Vista de las 10 marcas realizadas en la parte posterior del cuerpo del inyector.

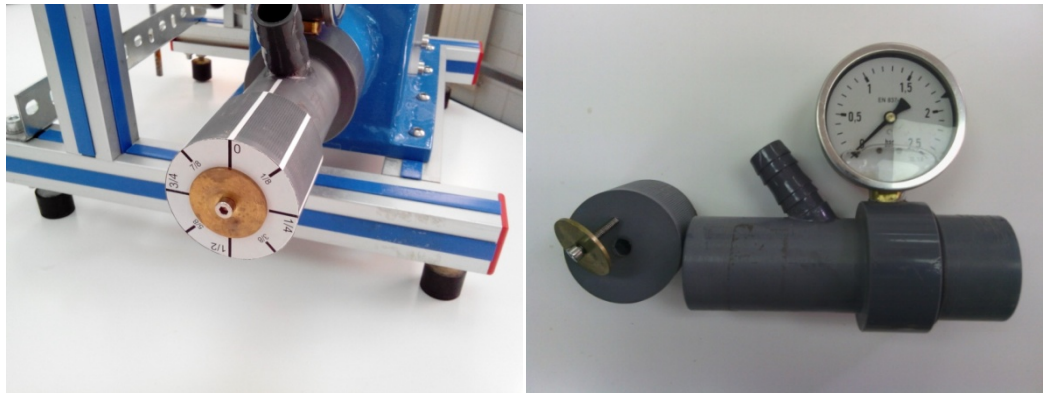


Figura 4.54. En la derecha vista sin la escala graduada. En la izquierda vista actual del inyector.

En la figura 4.54 en la que se muestra la pegatina y las marcas blancas para la correcta alineación del mando con el cuerpo (fotografía izquierda). En la parte derecha vemos como era el mando regulador de la aguja antes de la realización de la pegatina.

Como ejemplo podemos ver en las siguientes figuras distintas posiciones del inyector.



Figura 4.55. Vista detallada de la pegatina del mando del inyector.

Con el mando cerrado por completo con área de paso de agua nula figura 4.55. apréciase la alineación de las marcas del mando con la marca del inyector.

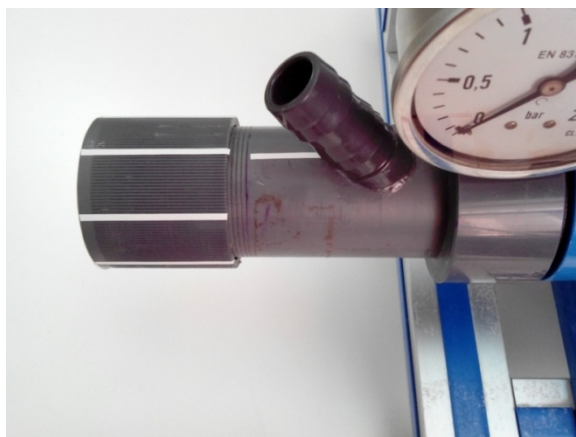


Figura 4.56. Vista del detalle de las marcas para facilitar la correcta posición del mando con respecto al cero.

En esta otra figura (figura 4.56) contamos con 4 vueltas completas del mando y si nos fijamos en la parte posterior la pegatina nos indicara la fracción de vuelta que tenemos.

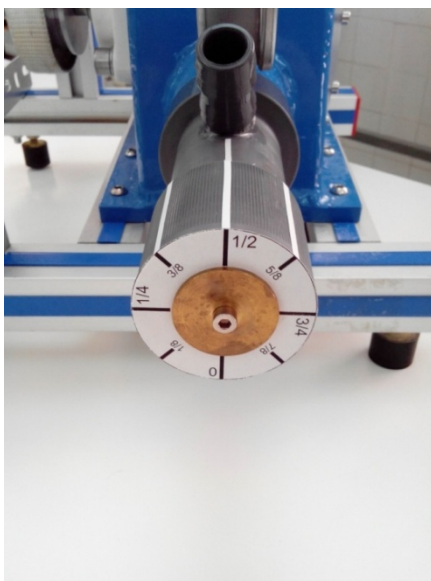


Figura 4.57. En la figura podemos ver la posición "1/2" del mando.

Como vemos (figura 4.57) tenemos $\frac{1}{2}$ vuelta. Por lo que la posición de la aguja con respecto al cero que sería la posición más adelantada (cerrada) sería de 4 vueltas + $\frac{1}{2}$ vuelta = $4 \cdot 1.5 + 0.75 = 6.75$ mm.

Con esta solución podremos seleccionar perfectamente las posiciones que queremos ensayar del inyector y además no importa si la aguja es desplazada por accidente porque podremos retornar a la posición que habíamos elegido.

Como ya conocemos la posición exacta de la aguja podemos obtener el valor de la sección de paso de agua deduciendo la función que la controla.

La fórmula del área de paso del agua que la aguja permite será función de su posición.

$$A = \frac{\pi}{4} (D_e^2 - D_i^2)$$

(Ec. 4.42.)

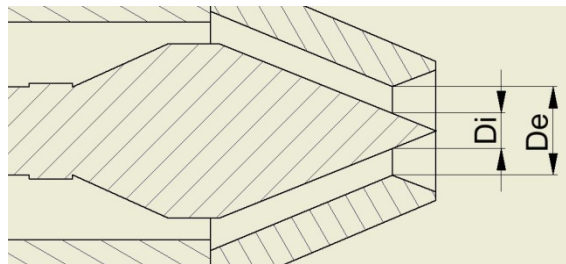


Figura 4.58. Esquema de los diámetros: exterior (tobera) e interior (aguja).

Siendo:

A = área de paso (toroide que queda de restar el círculo de “ D_e ” menos “ D_i ” de la aguja)

D_e^2 = cuadrado del diámetro exterior

D_i^2 = cuadrado del diámetro interior

Conocemos:

$$D_e = 10.2 \text{ mm}$$

(Ec. 4.43.)

El ángulo que forma el cono de la aguja con la horizontal:

$$\gamma = 21.8^\circ$$

(Ec. 4.44.)

Y sabemos que el diámetro interior es función de la posición de la aguja que como hemos visto antes podemos calcular:

$$D_i(x)$$

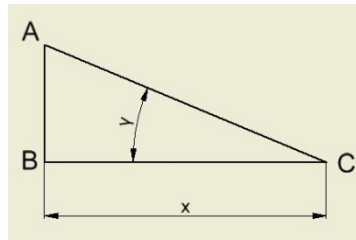


Figura 4.59. Representación esquematizada de la punta de la aguja.

En la figura 4.59 se muestra la representación teórica de la mitad de la cabeza de la aguja. El segmento AB representa el radio de la sección a calcular de la aguja.

Aplicando trigonometría tenemos:

$$\left. \begin{array}{l} AB = AC \sin \gamma \\ BC = AC \cos \gamma \end{array} \right\} \rightarrow AC = \frac{BC}{\cos \gamma}$$

$$AB = BC \tan \gamma$$

(Ec. 4.45.)

$AB = x \tan \gamma \rightarrow$ Es el radio, queremos obtener el diámetro interior D_i

$$D_i = 2 x \tan \gamma \quad \text{Para } x=[0, 0.1275] \text{ (m)}$$

(Ec. 4.46.)

De manera que el movimiento total de la aguja son 12.75 mm , si dividimos entra el paso de rosca de 1.5 mm obtenemos 8.5 vueltas completas posibles.

La fórmula del área queda como:

$$A = \frac{\pi}{4} [0.0102^2 - (2x \tan 28.1)^2]$$

(Ec. 4.47.)

Siendo:

x: la posición calculada con la escala del mando del inyector (se debe introducir en metros).

Como ejemplo, apliquemos la ecuación a una de las simulaciones llevadas a cabo.

Para la simulación teníamos cinco vueltas del mando regulador de la aguja:

$$5 \text{ vueltas} \approx x = 7.44 \text{ mm}$$

Haciendo uso de la ecuación 4.47 tenemos:

$$A = \frac{\pi}{4} [0.0102^2 - (2 * 0.00744 * \tan 28.1)^2] = 3.2133 \times 10^{-3} \text{ m}^2$$

(Ec 4.48)

Para la simulación teníamos un caudal de 2200 l/h, y como $Q \text{ (m}^3/\text{s)} = V \text{ (m/s)} * A \text{ (m}^2\text{)}$:

$$\text{Velocidad de salida} = \frac{2200 \frac{\text{l}}{\text{h}} * \frac{1 \text{ h}}{3600 \text{ s}} * \frac{10^{-3} \text{ m}^3}{1 \text{ l}}}{3.2133 \times 10^{-3} \text{ m}^2} = 19.01 \frac{\text{m}}{\text{s}}$$

(Ec 4.49)

Como veremos más adelante en el apartado: 4.5.3 Simulación numérica con ANSYS FLUENT, este valor obtenido de 19.01 m/s se asemeja a los valores de velocidad pico obtenidos en las distintas simulaciones, valiéndonos así para confirmar la validez de las distintas simulaciones realizadas.

4.5.3 Simulación numérica con ANSYS FLUENT

Por todos es bien sabido el potencial que nos brindan los softwares de simulación numérica. Para mis simulaciones he elegido Ansys Fluent por dos razones: La primera por ser un reconocido programa en el mundo industrial de probada eficacia y porque en una de las asignaturas de la carrera realizamos varias prácticas con él.

La principal razón de que llevara a cabo la simulación fue el hecho de que la turbina cuenta con bajísimo rendimiento comparado con las turbinas Pelton reales. Por otra parte la simple observación de los componentes del inyector nos lleva a pensar que no es un diseño muy optimizado. Al desmontarlo pude ver algunas de las razones que me han llevado a estudiarlo como son: Las impurezas debidas al agua, daños posiblemente de fabricación en el cuerpo del inyector, un gran ángulo de entrada del agua en el inyector, mal ajuste de la tobera con el cuerpo del inyector creando así oquedades que empeoran el flujo del agua y una mala longitud de la pieza que le da movilidad a la aguja. A través de la simulación numérica validaremos todas estas hipótesis.

Para realizar las distintas simulaciones llevadas a cabo primero debemos obtener la zona de fluido. No es válido introducir las distintas piezas del inyector directamente en el

programa de mallado (las que podemos ver en los planos) sino que habremos de reproducir la zona hueca por la que circula el agua a su paso por el inyector para mallarla (con un programa de mallado) y posteriormente usando el programa de simulación obtendremos unos resultados.

En la figura podemos ver el corte longitudinal del inyector con todas sus piezas. Para poder llevar a cabo la simulación debemos reproducir en 3D la zona marcada en gris (figura 4.60) que es la zona por donde circula el agua, nuestra región de estudio.

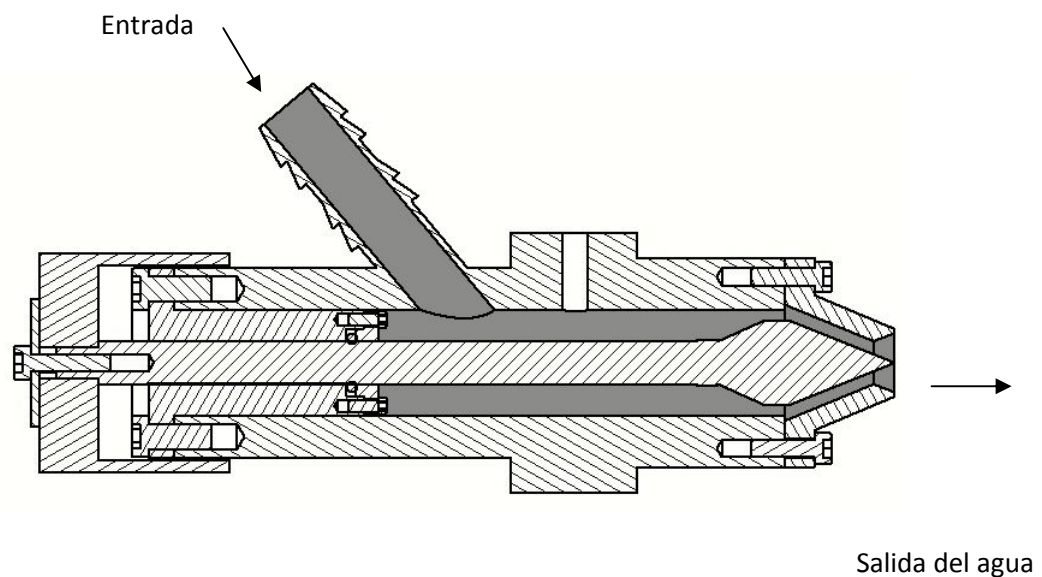


Figura 4.60. Vista en corte de la zona por la que fluye el agua dentro del inyector.

Aquí podemos ver como quedaría la zona de estudio antes de mallarla (figura 4.61).

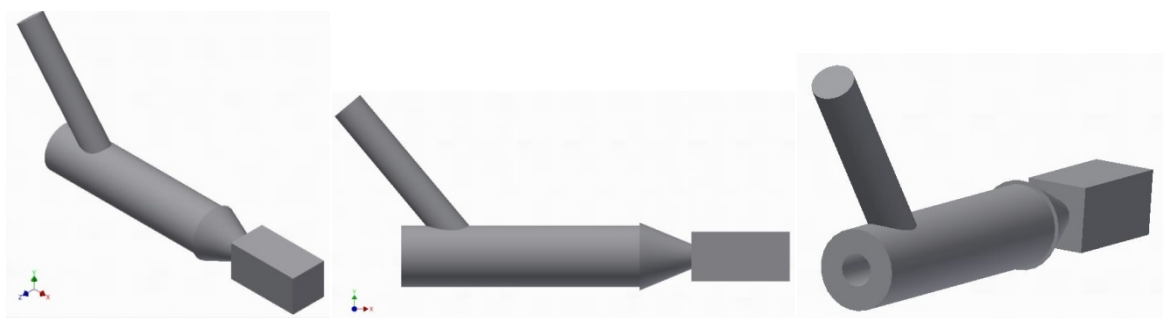


Figura 4.61. Representación 3D de la zona fluida.

En la figura 4.61 podemos ver cómo están reproducidos todos los detalles de la zona por la que fluye el agua dentro del inyector. El paralelepípedo que se observa hace las veces de tobera de salida. En esa zona el agua sale a presión ambiente y tiene una longitud suficiente para no interferir en los resultados del programa.

Para todas las simulaciones esta zona que representa por la que fluye el agua será la misma. Anteriormente hemos indicado que antes de realizar la simulación debíamos mallar la zona de estudio. Se han llevado a cabo varias simulaciones con distintos tipos de malla que se irán especificando a lo largo del documento.

Las primeras simulaciones llevadas a cabo fueron dos simulaciones con mallas tetraédricas y con la misma configuración del solver (solución).

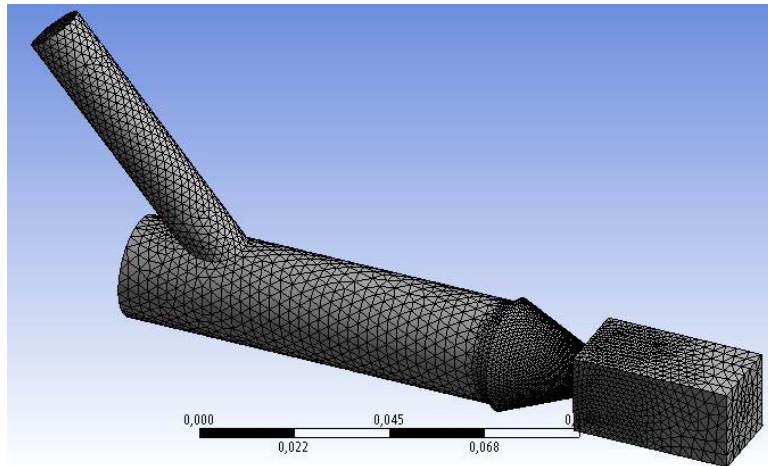


Figura 4.62. Vista de la malla tetraédrica.

Aquí (figura 4.62) podemos ver la malla ya elaborada. Los distintos tipos de tamaño de elemento se deben al refinado que se realizó en las zonas con mayores gradientes de velocidad.

Comencemos por la simulación k- ϵ .

Primero cargamos la malla en el programa Fluent.

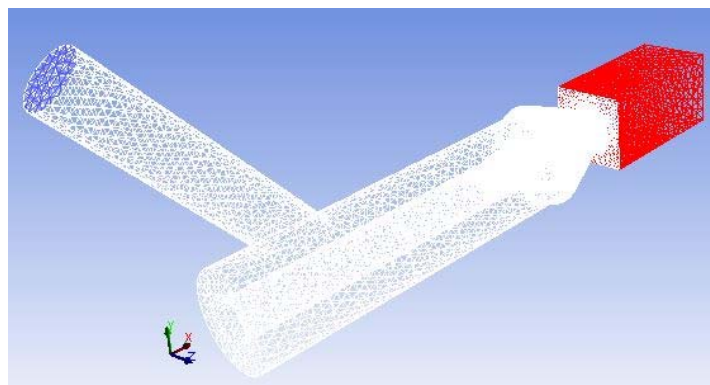


Figura 4.63. En azul tenemos la entrada y en rojo la salida.

Se comprueba que la malla se cargó correctamente y que tiene una calidad correcta. Para mayor información sobre la calidad del mallado ver la referencia [11].

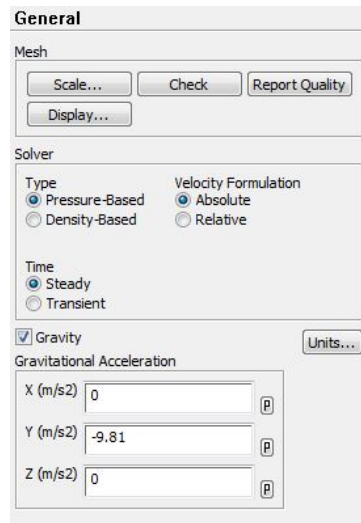


Figura 4.64. Panel general del programa ANSYS FLUENT de nuestra simulación.

En la figura 4.64 se muestra la configuración del solver. Se elige pressure-based porque no habrá grandes cambios en la densidad del fluido de estudio y steady porque la simulación es estacionaria. También incluimos el valor de la gravedad.

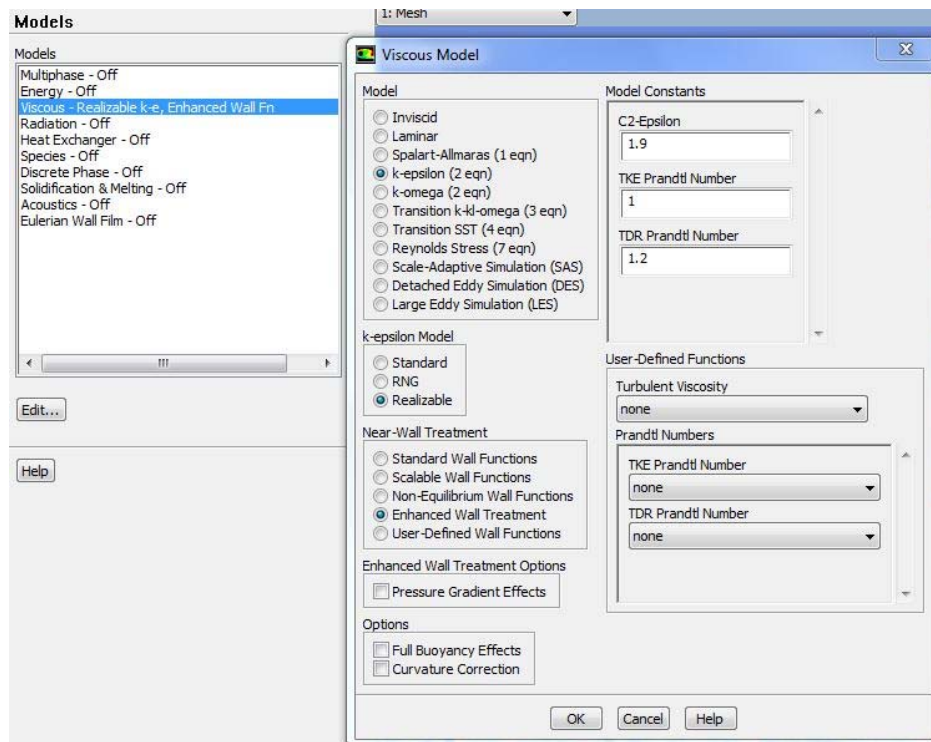


Figura 4.65. Panel de configuración del modelo de turbulencia.

En cuanto a los modelos, todos están desactivados (figura 4.65), salvo el modelo de turbulencia que para este caso es el k-epsilon Realizable para un cálculo fiable pero que no requiera muchos recursos del ordenador de trabajo (más recursos mayor tiempo de cálculo). Además como nos interesa que ocurre a lo largo de las paredes de la aguja y tobera (zona de estudio importante) se ha seleccionado Enhanced wall treatment para un mejor estudio.

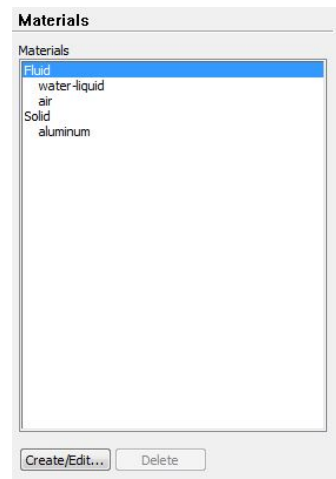


Figura 4.66. Los materiales de trabajo obviamente son el agua líquida.

En cuanto a las condiciones de contorno:

-Entrada: Se seleccionó velocity inlet, es decir, que nosotros indicamos la velocidad de entrada del agua. El vector de velocidad de entrada del agua es perpendicular a la superficie con una magnitud de 11m/s y un diámetro hidráulico de 0.0135

-Salida: Para ésta seleccionamos pressure outlet (salida a presión). Como estamos a presión ambiente a la salida la presión a introducir es cero ya que introducimos la manométrica. También introducimos su diámetro hidráulico.

Para los métodos resolutivos (figura 4.67) elegí primeros órdenes en las primeras iteraciones para luego pasar a órdenes mayores y así tener una simulación más sólida.

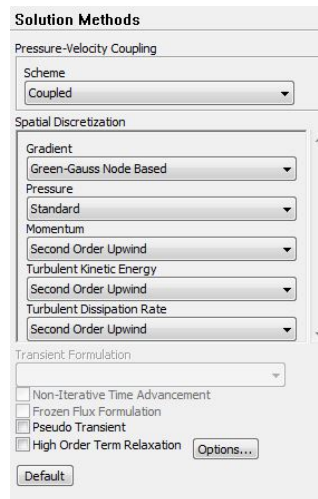


Figura 4.67. Selección de órdenes para los métodos de resolución.

Una vez se ha llevado a cabo el cálculo debemos comprobar los residuos, monitores y los flujos de entrada y salida. Esto se hace para saber si la simulación ha ido correctamente o no, veamos nuestro caso.

Residuos:

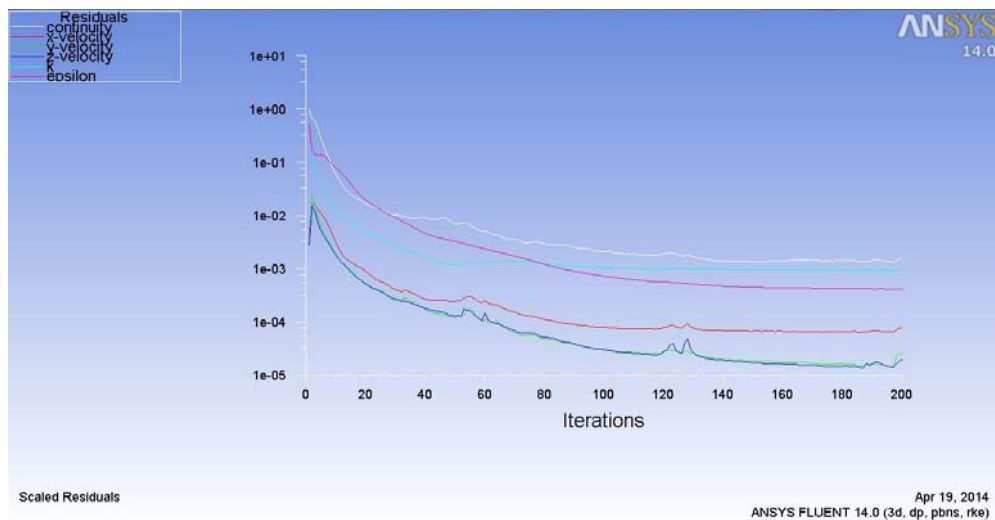


Figura 4.68. Residuos.

En principio cuanto más bajo sea el orden de los residuos (figura 4.68) mejor, en nuestro caso la continuidad no llega a orden 3. Esto podría indicar que la simulación no ha ido muy bien. Para mayor información acerca de los residuos, flujos y monitores consultar las referencias [10] y [11].

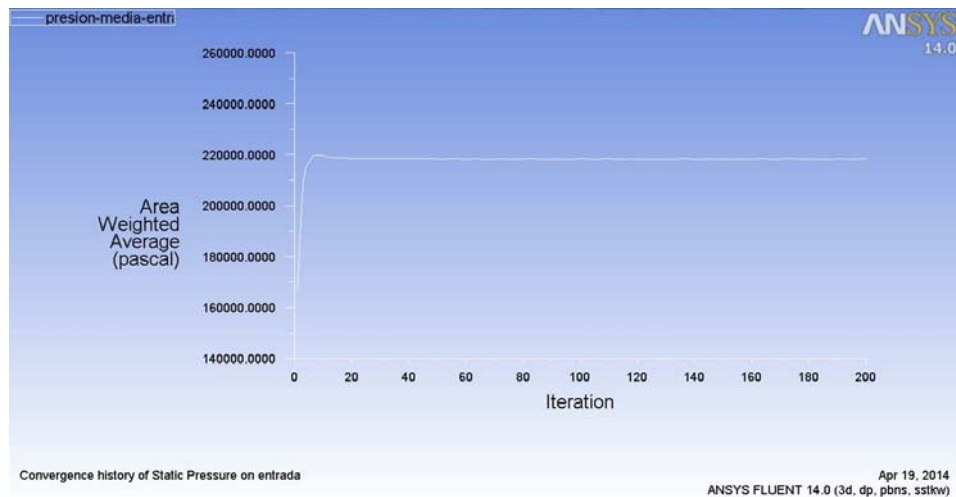


Figura 4.69. Monitor de presión.

Este es el monitor de presión media (figura 4.69). Como vemos tiene un valor constante al llegar a un cierto número de iteraciones, esto es un buen resultado.

Por último veamos los flujos de entrada y salida.

Mass Flow Rate	(kg/s)
entrada	0.6111
salida	-0.61107029
Net	2.9709668e-05

La diferencia de flujo de entrada y salida es de orden - 5 de manera que es un buen indicativo de que la simulación se ha llevado a cabo correctamente.

Pese a obtener un orden alto en los residuos damos por válida la simulación. Ahora veremos una vista que corta a la zona de estudio por la mitad longitudinalmente como se muestra en la figura 4.70.

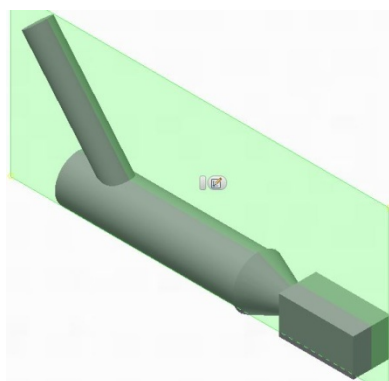


Figura 4.70. Plano de corte para las vistas de los gradientes de velocidad y presión.

Velocidades:

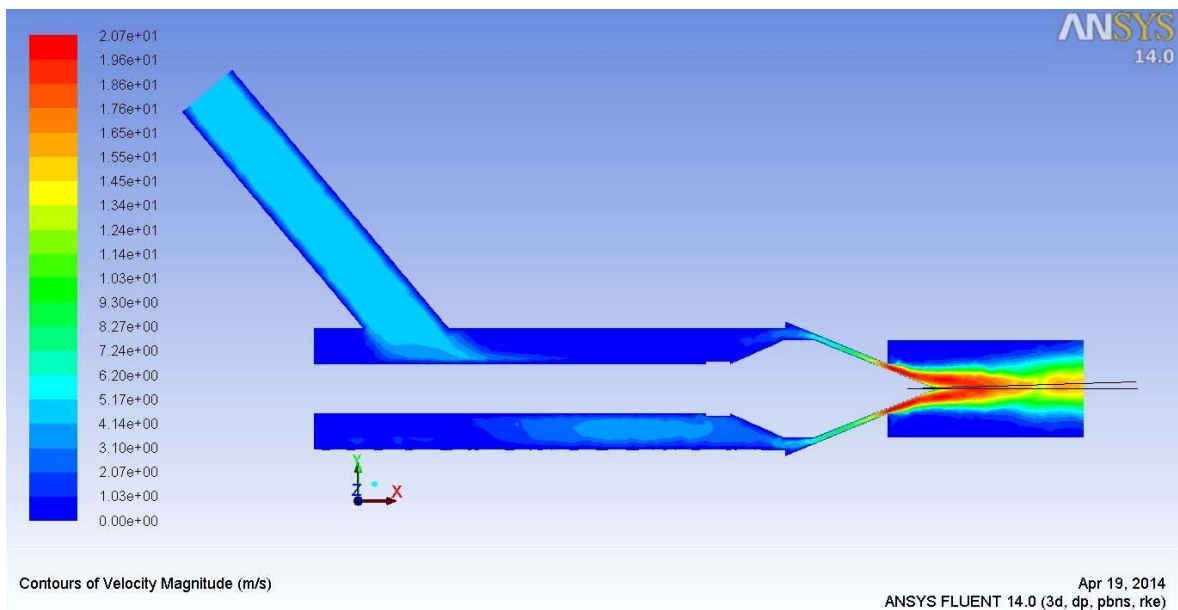


Figura 4.71. En la figura se aprecia como el chorro no es horizontal sino que tiende a subir con un pequeño ángulo.

En la figura 4.71 vemos los distintos contornos de velocidades. Vemos como el agua entra con una velocidad uniforme por la entrada. El agua tiende a seguir su camino hacia abajo una vez llega al cuerpo del inyector y vemos una zona de mayor cantidad de movimiento en la parte media inferior para luego pasar a la estrecha zona de limitada por la aguja y la tobera donde aumenta la velocidad hasta alcanzar su valor máximo de salida. Las zonas más oscuras tienen menor velocidad y esto no es bueno. Tenemos una pérdida de velocidad importante en casi todo el inyector y esto no es interesante ya que estamos ralentizando el agua para luego aumentar su velocidad otra vez. Esto implica que no toda el agua que entra sale uniformemente y eso se traduce en pérdidas por recirculación. Además el gran ángulo de entrada del agua con respecto a la horizontal hace que el agua tienda a fluir más por la parte inferior de la aguja que por la parte superior llegando así a que obtengamos un chorro que tiende a desplazarse hacia arriba provocando que el agua no impacte en las cucharas correctamente (en la imagen se ha dibujado como líneas de referencia la horizontal y la que marca la dirección real del chorro de agua al salir).

Si analizamos más detenidamente los vectores de velocidad podemos verlo mejor.

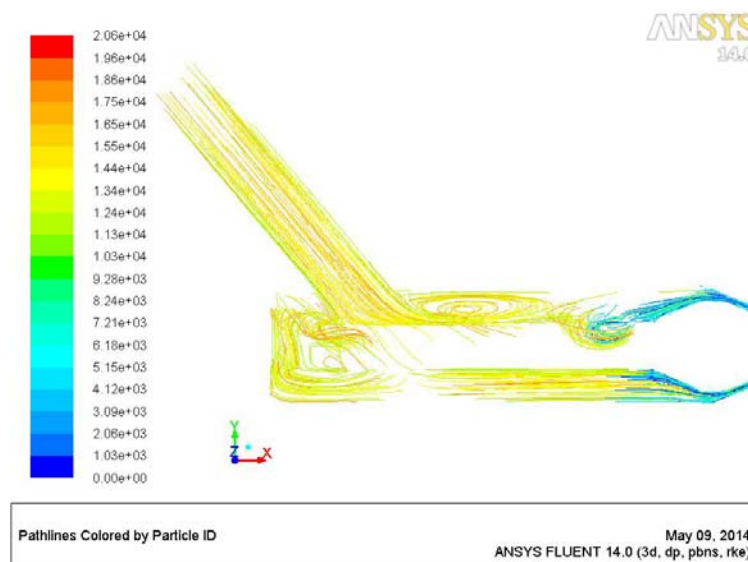


Figura 4.72. Vista de las trayectorias de las partículas fluidas.

Podemos apreciar (figura 4.72) muy bien las circulaciones de las partículas. Las zonas más problemáticas que cuentan con mayor turbulencia son la superior y la parte posterior en la que queda una zona vacía debido a que la pieza que da movilidad a la aguja (en los planos es la pieza “Base Aguja”) no tiene una mayor longitud para disminuir esta zona inservible.

Haciendo zoom en las partes anteriormente comentadas de la aguja (argolla y aristas vivas) y el mal acople entre el cuerpo del inyector y la tobera vemos lo siguiente.

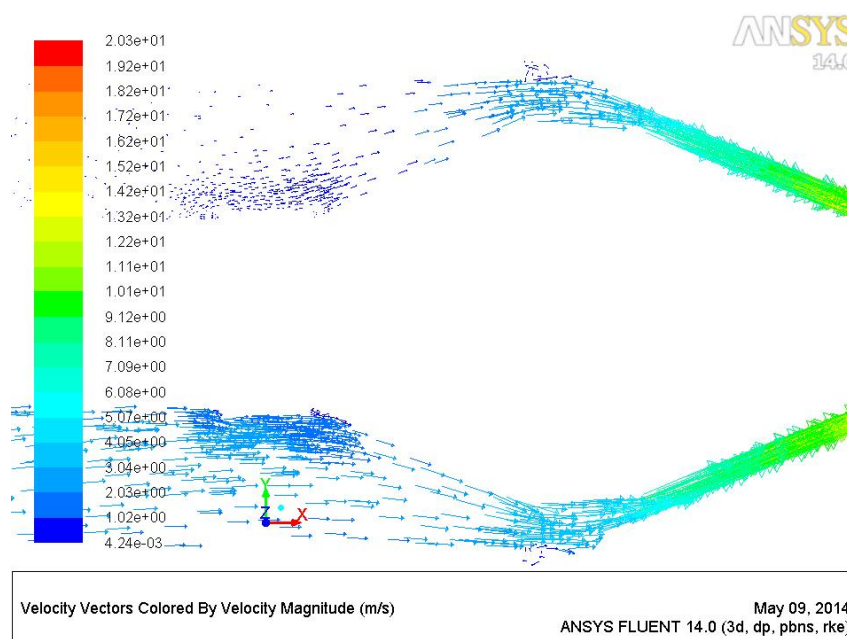


Figura 4.73. Vista de los vectores velocidad en la zona del cabezal de la aguja.

Se aprecia cómo los vectores tienen un cambio brusco de dirección y de magnitud en las zonas de la argolla y ajuste tobera-cuerpo debido a las aristas vivas (figura 4.73).

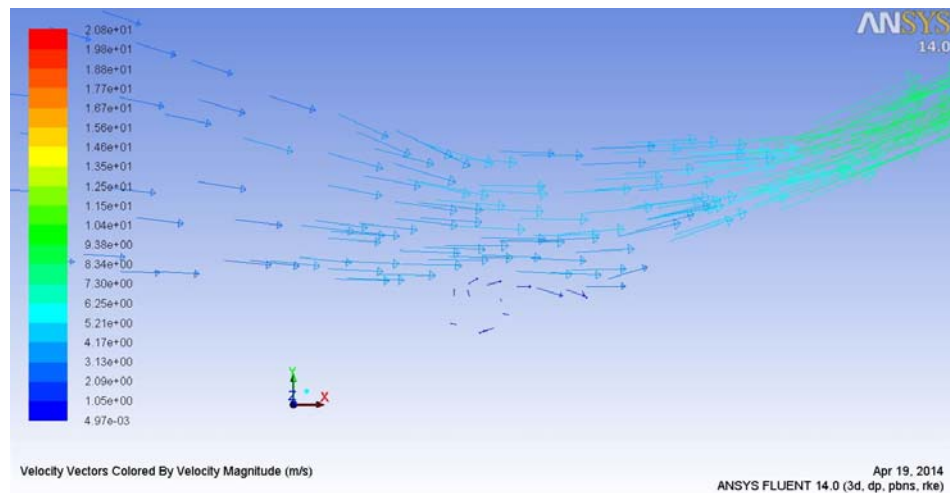


Figura 4.74. Representación aumentada de los vectores de velocidad en las zonas de turbulencia debidas al mal acople tobera-inyector.

En esta otra figura 4.74 se aprecia claramente cómo cambia de dirección los vectores velocidad debido al mal acabado del inyector (zona del ajuste tobera-cuerpo). El que haya vectores que tienen un sentido contrario al natural que sigue el agua nos indica que existe una zona de recirculación (turbulenta).

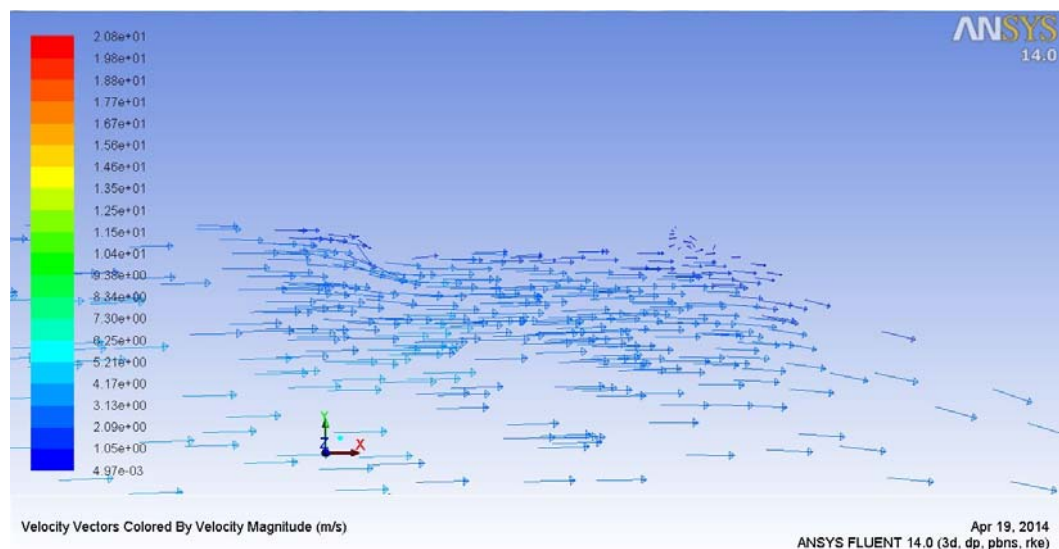


Figura 4.75. Aquí vemos la recirculación provocada por la argolla de la aguja.

Presión:

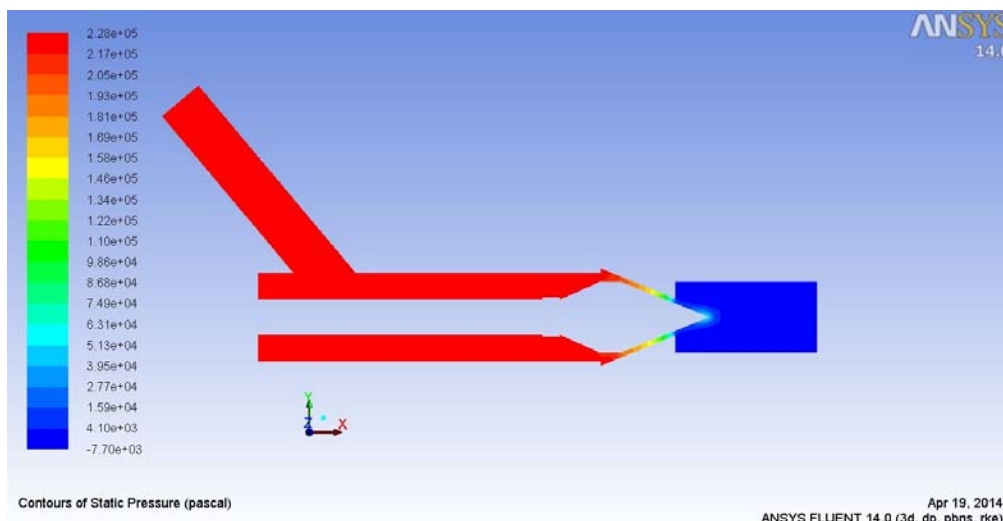


Figura 4.76. Contornos de presión,

Esta figura 4.76 nos ayuda a corroborar la validez de la simulación ya que por la teoría podíamos imaginar que la presión lejos de la salida sería la de la instalación (que nos proporciona la altura del salto de agua del que disponemos) y al acercarnos a la salida tendríamos la presión ambiente.

Ahora veremos la simulación con k-omega.

Con respecto a la configuración de los parámetros del programa es la misma salvo por el modelo de turbulencia que para este caso es el k-omega. Tanto la zona de fluido como la malla también es la misma.

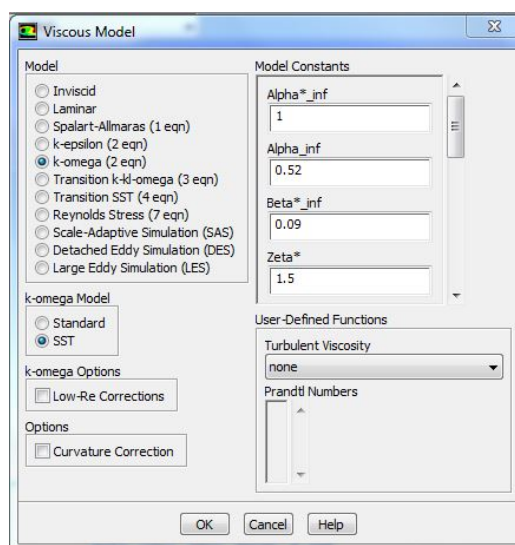


Figura 4.77. Selección el modelo de turbulencia.

Se eligió este modelo (figura 4.77) para poder comparar los resultados con el k-epsilon. Al seleccionar el modelo SST el parámetro y^+ (tamaño de celda cercana a la pared) ha de ser menor que 2, en nuestra malla tan sólo la parte refinada cumple con esa especificación. Se intentó refinar toda la malla pero entonces el cálculo se hacía imposible debido a la gran cantidad de celdas a resolver (mi ordenador es poco potente de cara al cálculo).

Pasemos a comprobar la calidad de la simulación:

Residuos:

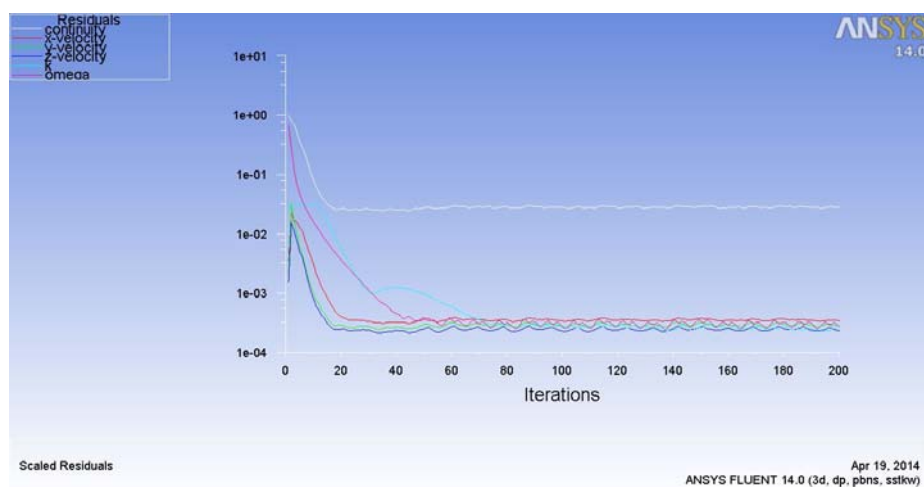


Figura 4.78. Residuos.

De nuevo la ecuación de continuidad da problemas y no converge. No consigue llegar ni a orden menor que 2 (figura 4.78).

Monitor:

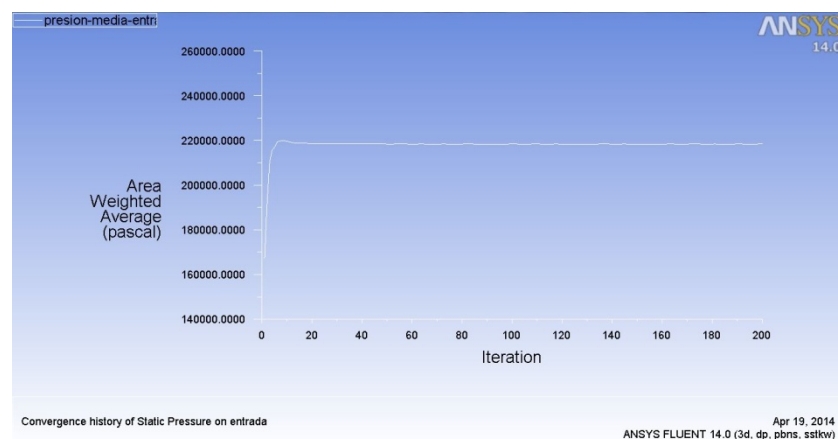


Figura 4.79. Este monitor sí que es bueno ya que toma un valor constante.

Flujos:

Mass Flow Rate	(kg/s)
entrada	0.6111
salida	-0.61107029
Net	2.9709668e-05

De nuevo el flujo es de orden muy pequeño.

Aunque los residuos no son excesivamente buenos damos por buena la simulación.

Velocidad:

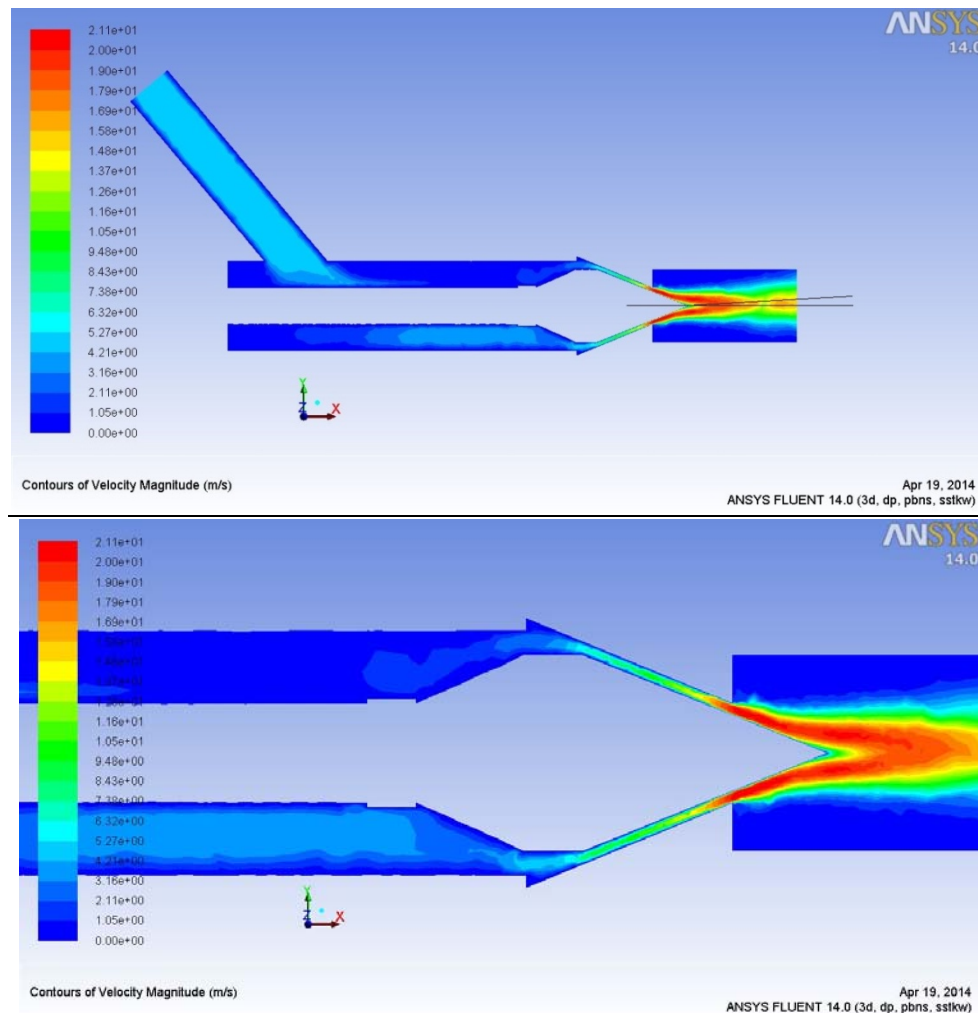


Figura 4.80. Representación de los contornos de velocidad con las líneas que muestran el ángulo de desviación del chorro.

En cuanto a los contornos de velocidad (figura 4.80) obtenemos unos muy parecidos a los anteriores tanto en la forma de los contornos como las distintas magnitudes de velocidad. En esta simulación también se aprecia como el chorro de agua sale hacia arriba y no horizontal.

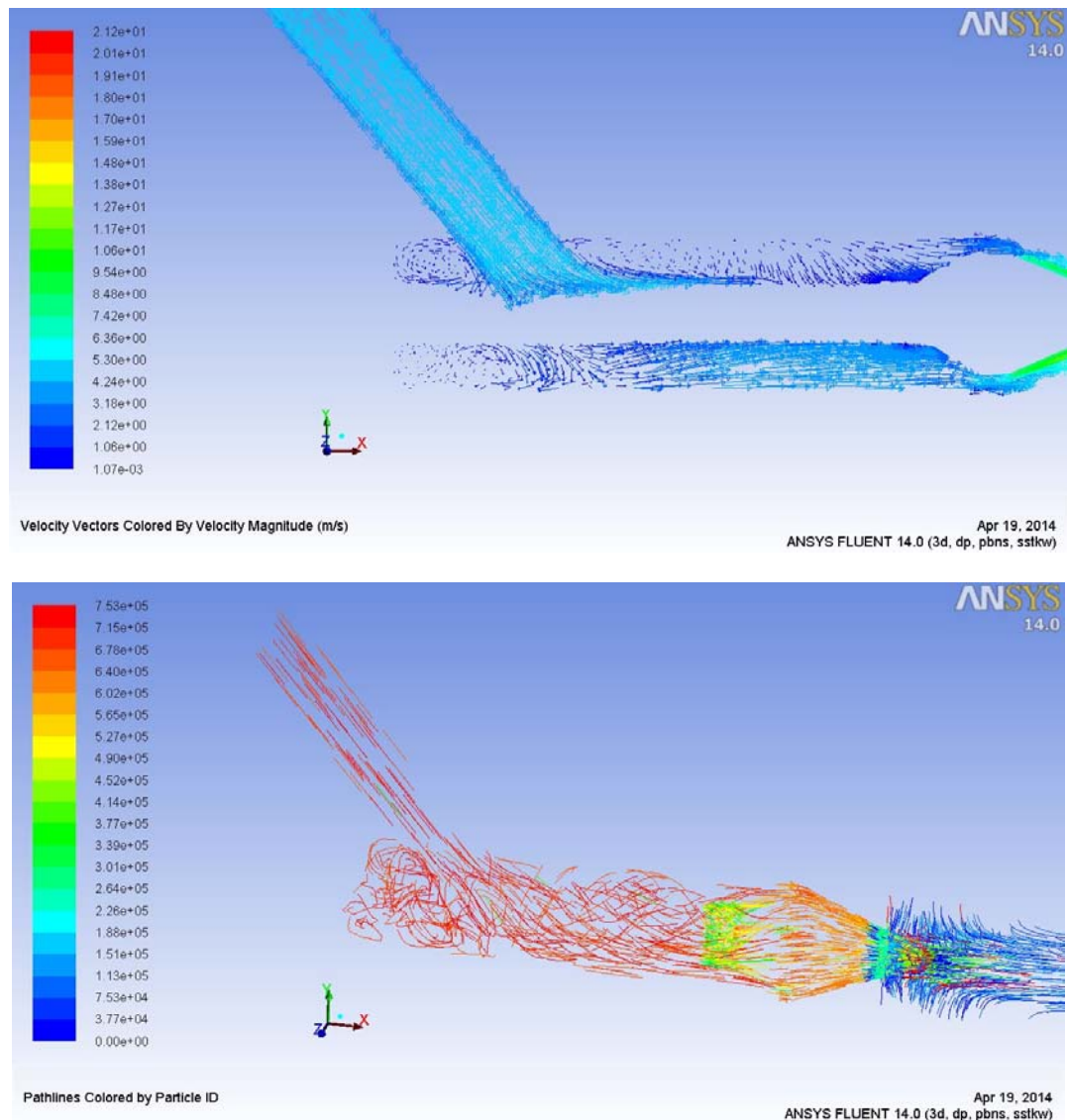


Figura 4.81. Arriba: Vista global de los vectores de velocidad. Abajo: vista de las trayectorias turbulentas de las partículas fluidas.

En la figura 4.81 apreciamos las recirculaciones de agua tan importantes que hay dentro del inyector. Todas ellas contribuyen a un bajo rendimiento.

Ahora pasaremos a exponer las simulaciones llevadas a cabo con una malla distinta. En estas otras se ha utilizado una malla de cubos (quad) figura 4.82.

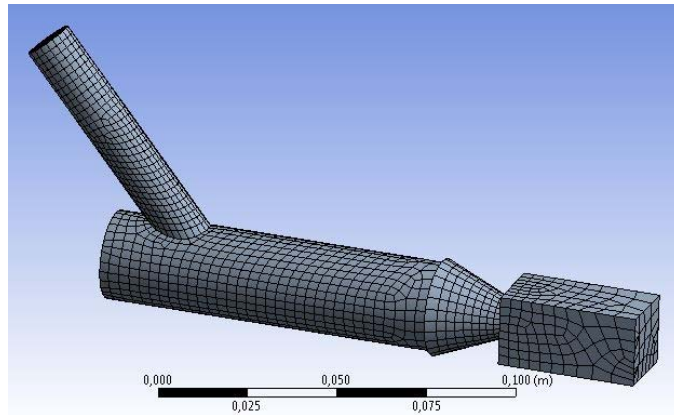


Figura 4.82. Representación de la malla estructurada.

Como vemos con este tipo de elementos obtenemos una malla estructurada. Su uso es muy recomendable cuando se han de realizar simulaciones de fluidos (las mallas de tetraedros son más indicadas para las simulaciones de elementos finitos). Algunas de las ventajas de las mallas estructuradas son: el menor número de elementos, mejor alineación de la malla con el sentido del flujo de fluido, mejora la convergencia de los residuos y para mi caso las simulaciones con malla estructurada fueron del orden de varios minutos más rápidas que las no estructuradas anteriores.

Con la misma configuración anterior que para el modelo k-epsilon se llevo a cabo la simulación con la malla estructurada, obteniendo:

Residuos:

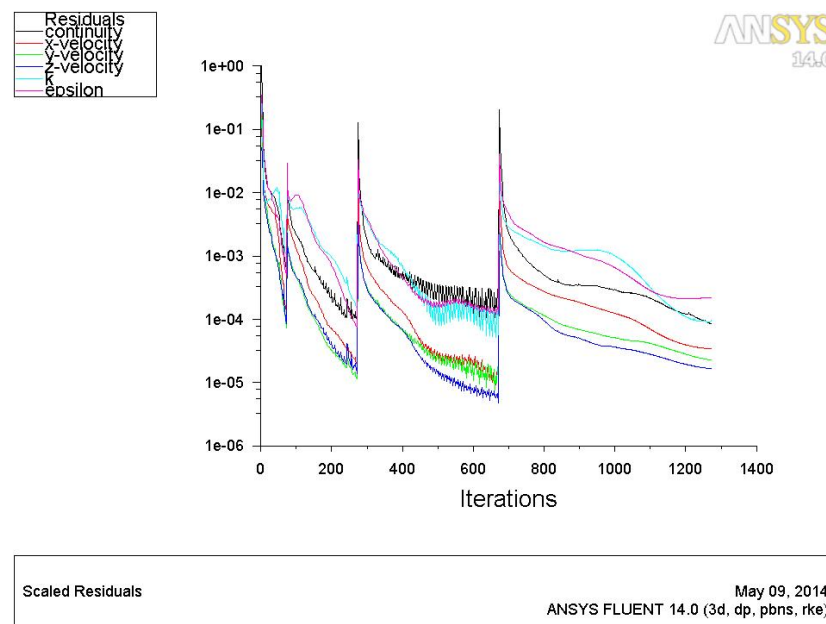


Figura 4.83. Residuos.

Los residuos (figura 4.83) convergen mejor que en las simulaciones anteriores. Aparecen esos grandes saltos ya que los que se muestran son los residuos de diversos cálculos. No son excelentes pero si son válidos ya que todos son menores del orden 3 y algunos sobrepasan el orden 5.

Monitor:

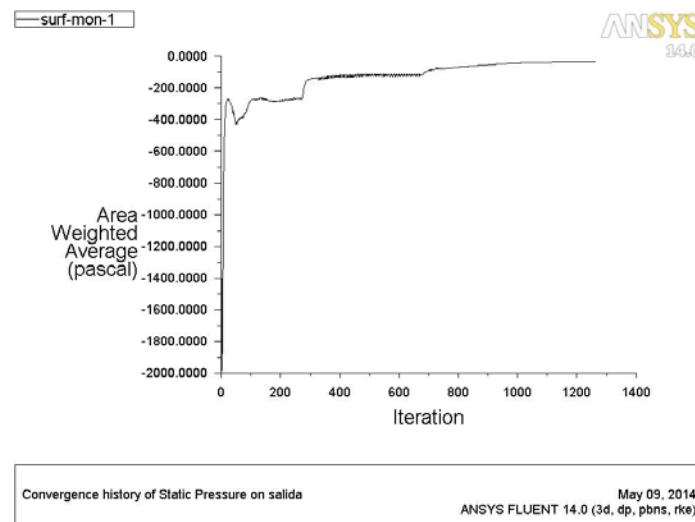


Figura 4.84. Monitor de presión.

Al igual que en los residuos el monitor (figura 4.84) aparece irregular por los distintos cálculos llevados a cabo. De cualquier forma vemos como en el último cálculo el monitor de presión llega a un valor constante.

Flujos:

"Flux Report"

Mass Flow Rate	(kg/s)
----------------	--------

entrada	0.6111
salida	-0.61110239

Net	-2.3892367e-06
-----	----------------

El flujo es muy bueno pues es de un orden menor que en las anteriores simulaciones.

Por todo lo anterior damos por buena la simulación. Veamos los contornos de velocidad y presión:

Velocidad:

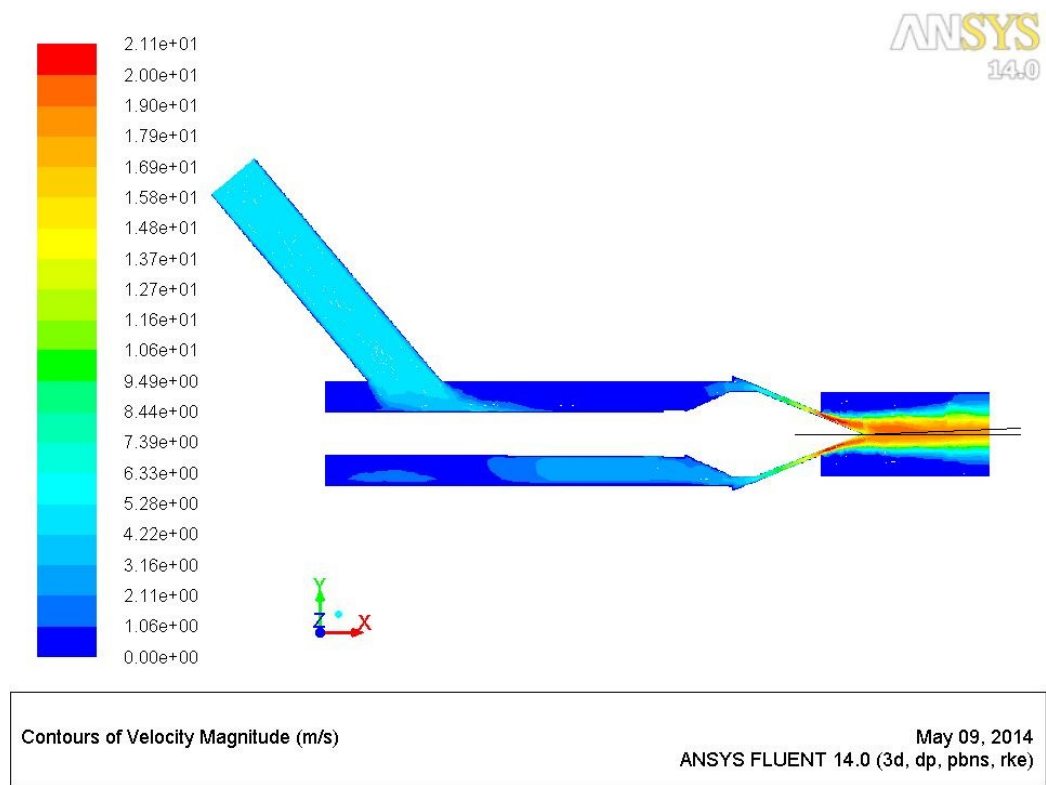


Figura 4.85. Gradientes de velocidad con líneas auxiliares que indican el ángulo del chorro.

Los contornos de velocidad (figura 4.85) son muy similares a los anteriores. Bien es cierto que disminuyen las zonas de menor velocidad en el cuerpo y obtenemos un chorro en la salida con contornos más sólidos que con las mallas tetraédricas. En cuanto a los valores de la velocidad son muy parecidos a los obtenidos anteriormente. De nuevo apreciamos cómo el chorro no sale horizontal.

Presión:

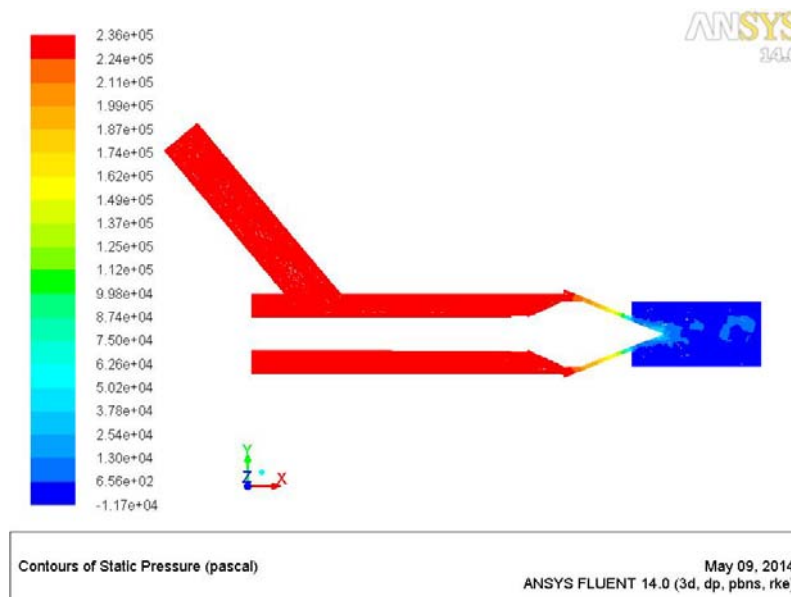


Figura 4.86. Gradientes de presión.

Al igual que la velocidad se parecen mucho a los obtenidos con la malla anterior. Los valores del gradiente toman también los mismos valores (figura 4.86).

En resumen podemos destacar que para este caso el parámetro que ha marcado la diferencia no ha sido el modelo de turbulencia elegido sino la malla. Una buena malla es muy importante ya que aparte de permitir obtener unos resultados válidos agiliza el cálculo (en mi caso en las simulaciones más largas me ahorró entorno a 10 minutos).

De cara a los resultados de las simulaciones podemos darlos por buenos aunque no podamos comprobarlos experimentalmente (no disponemos de equipo para comprobar las velocidades o presiones). Todos los gradientes, tanto los de velocidades como los de presiones, se ajustan con lo que a priori la teoría nos indica.

Ya que hemos dado por buenas las distintas simulaciones podemos afirmar que una parte muy importante por la que la turbina Pelton de que disponemos tenga unos resultados tan pobres se debe al inyector. Tanto las impurezas debidas al uso como las distintas circulaciones que se generan debido a un no muy acertado diseño de la entrada de agua, de unos acabados superficiales no muy buenos y el poco ajuste entre piezas contribuyen al bajo rendimiento.

Ya que hemos identificado los problemas del inyector yo propongo una variante, que manteniendo el estilo primitivo, mejora ciertos aspectos. Aquí se incluyen algunas

figuras descriptivas del mismo pero en el apartado de planos se adjuntan todas las piezas en detalle.

Los problemas del actual inyector son:

- Gran ángulo de entrada del agua.

- Mal ajuste cuerpo-tobera de salida.

- Aristas vivas en el cuerpo del inyector y aguja.

- Argolla en el vástago de la aguja.

- Incorrecta longitud de la pieza “Base aguja” quedando esta muy corta y permitiendo circulaciones innecesarias de fluido.

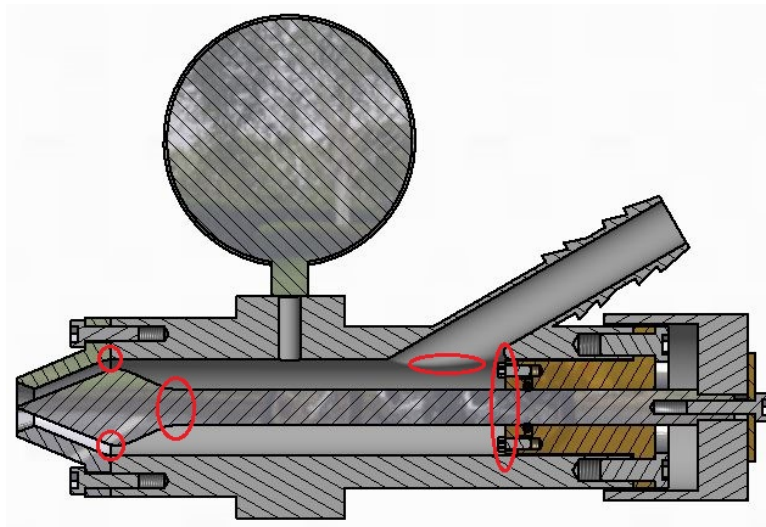


Figura 4.87. En la figura se muestran rodeados por un cirulo rojo los elementos mejorados en el inyector propuesto.

Como solución podemos ver en la figura 4.87 las medidas que se han adoptado:

- El ángulo de entrada del agua ha sido reducido en 20°.

- La zona triangular que aparecía en la unión cuerpo-tobera se ha eliminado.

- Se han redondeado las aristas de la aguja con radios de acuerdo así como en la entrada del agua.

- Se ha eliminado la argolla del vástago de la aguja.

-Se ha variado la longitud de la pieza “Base Aguja”.

Para comprobar si estos cambios darían efecto en el caso de fabricar un nuevo inyector se ha realizado una simulación variando la zona fluida con todos los cambios introducidos en el inyector.

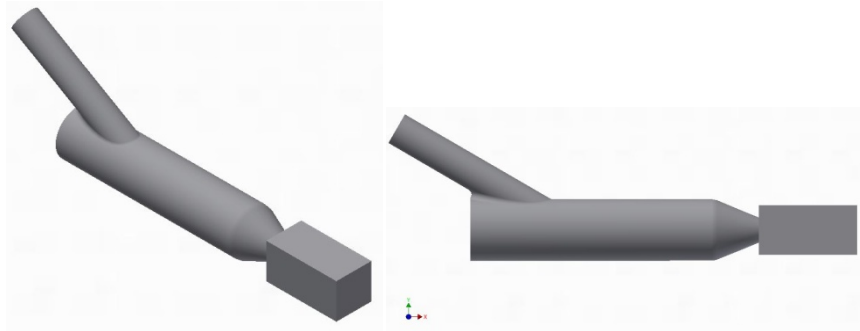


Figura 4.88. Representación de la zona fluida del inyector propuesto.

Podemos apreciar en la figura 4.88 la mayor suavidad de todas las partes de la zona fluida, en especial, la delantera.

Para la simulación se realizó una malla estructurada tipo quad (figura 4.89). Se refinó en las partes en las que se podría dar una mayor vorticidad del agua.

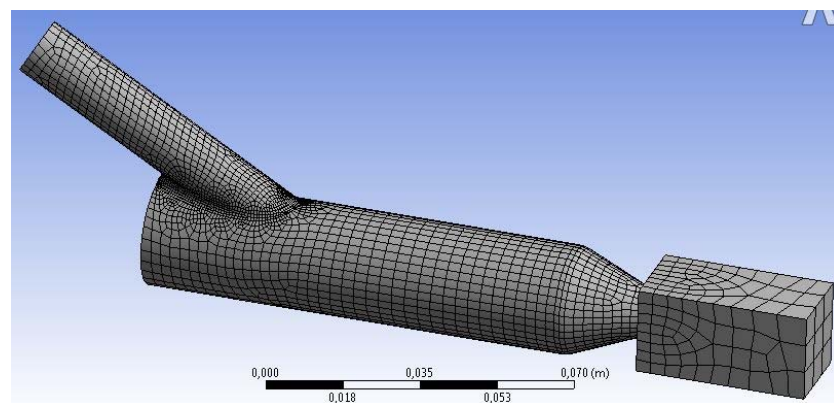


Figura 4.89. Malla estructurada del inyector propuesto.

Todos los demás parámetros de la simulación son iguales a la simulación k-epsilon anteriormente llevada a cabo.

Residuos:

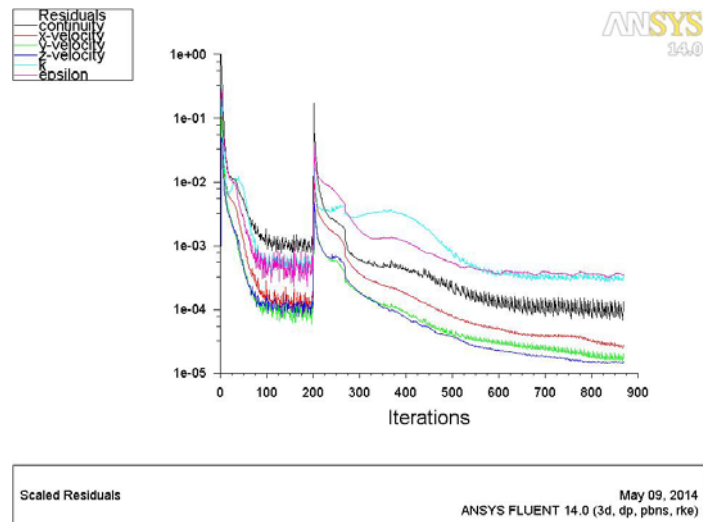


Figura 4.90. Residuos.

Los residuos, figura 4.90, no son excelentes pero son suficientemente buenos, todos ellos sobrepasan el orden 3.

Monitor:

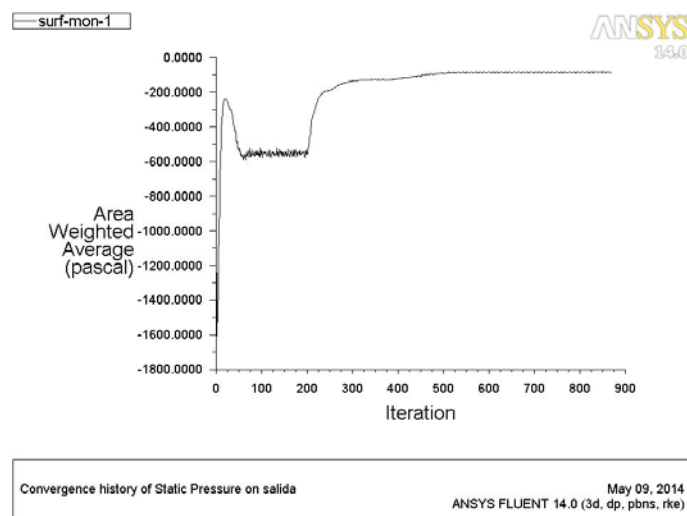


Figura 4.91. Tras cambiar los órdenes de cálculo del solver llegamos a unos valores estables del monitor de presión.

Flujos:

"Flux Report"

Mass Flow Rate (kg/s)

entrada	0.6111
salida	-0.61110652

Net	-6.517611e-06

El flujo tiene de nuevo orden menos 6 como en la última simulación realizada con malla quad.

Por todo esto la simulación es válida.

Velocidades:

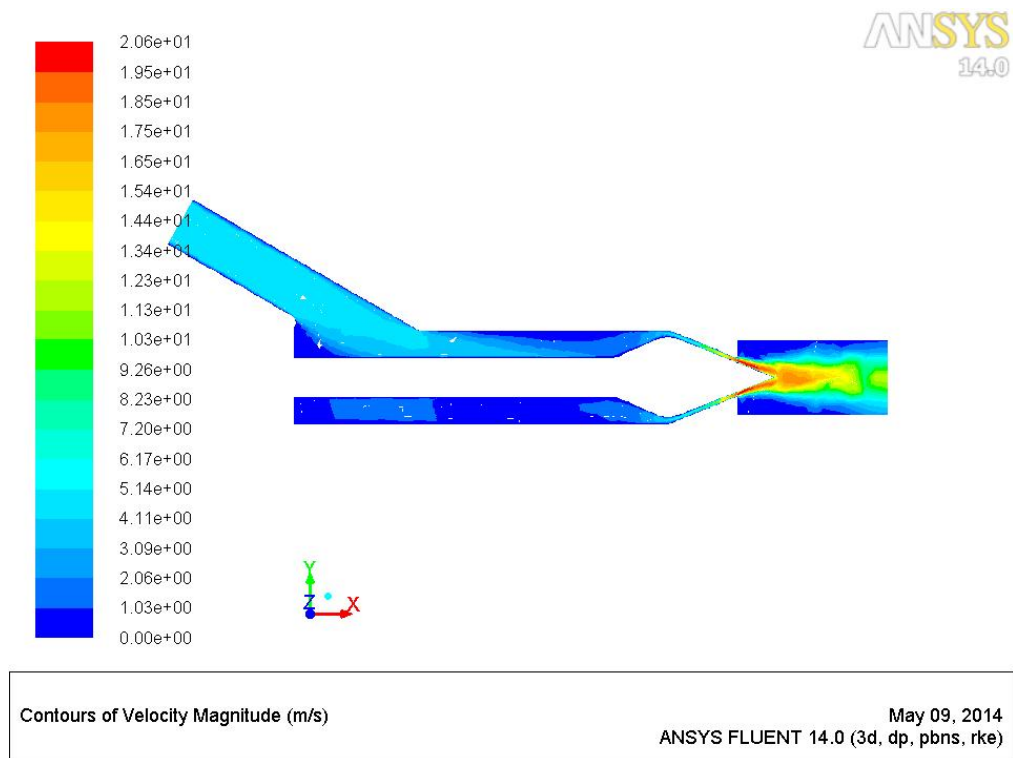


Figura 4.92. Gradientes de velocidad.

Vemos en la figura 4.92 como la zona de menor cantidad de movimiento es menor ahora (zonas de azul más oscuro dentro del inyector), aunque debido a que la entrada de agua no es en la misma dirección que el eje de la aguja siempre habrá circulación. Podemos ver que ahora el chorro de agua es casi paralelo al eje horizontal de salida. Es más promediado al paralelepípedo de salida. Aunque las velocidades no aumentan (no es

posible aunque tuviéramos un inyector ideal) la mejora de la cantidad de movimiento global del inyector ayudaría a obtener un mayor rendimiento.

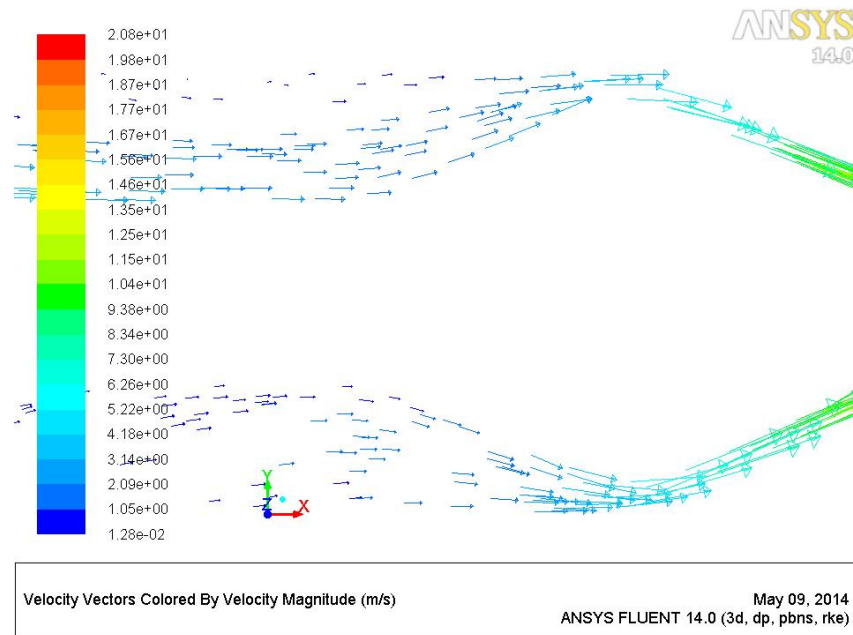


Figura 4.93. Vista aumentada del cabezal de la aguja. Están representados los vectores de velocidad.

Podemos ver (figura 4.93) como ahora los vectores cambian de dirección más suavemente y que en ningún caso toman sentido contrario al de salida del agua. La zona representada es la parte posterior del cabezal de la aguja.

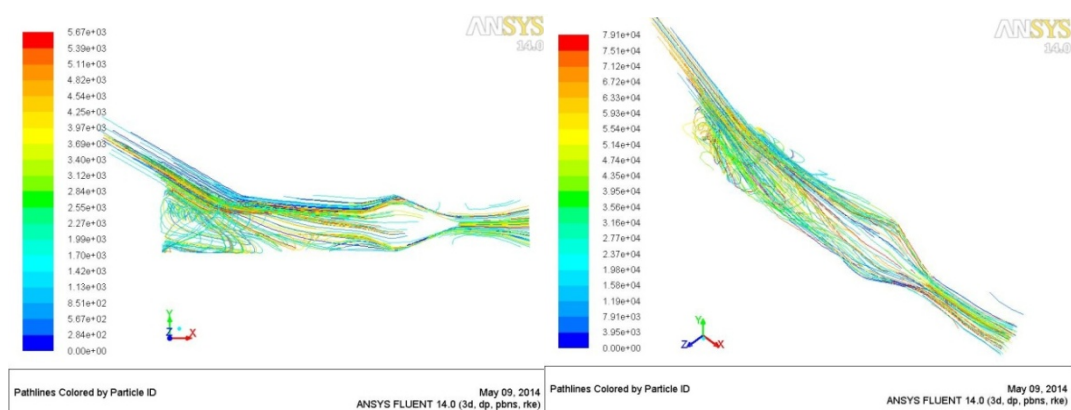


Figura 4.94. Vista de las trayectorias de las partículas fluidas. Nótese que no son tan turbulentas como para el inyector actual.

Se aprecia (figura 4.94) como las líneas de corriente toman trayectorias más fluidas que para el inyector actual. La única zona de gran vorticidad es la parte posterior, pero a menos que hagamos un diseño completamente nuevo del inyector, no podrá desaparecer.

Presión:

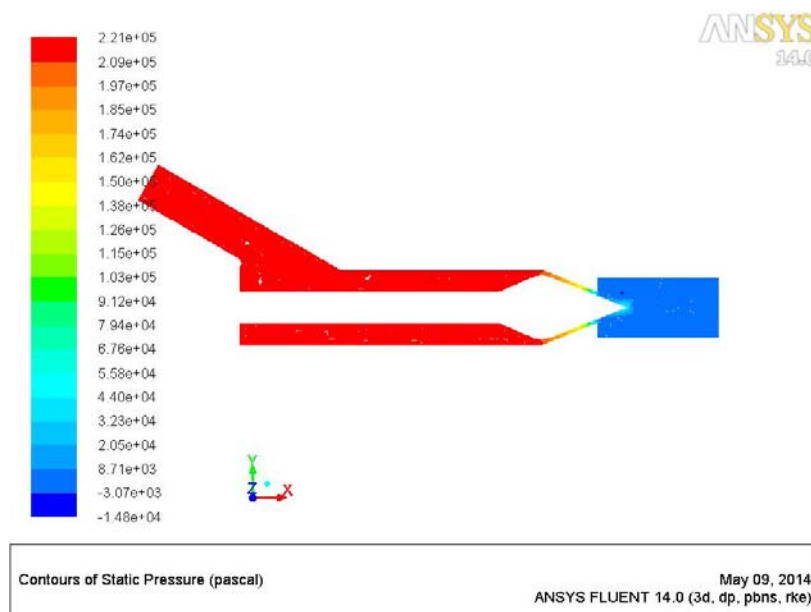


Figura 4.95. Gradientes de presiones.

De nuevo (figura 4.95) obtenemos unos contornos y valores parecidos a los resultados anteriores. Cosa que era de esperar.

En resumen, podemos concluir que el inyector propuesto mejora ciertos aspectos del actual inyector, y que por consiguiente, mejoraría los rendimientos obtenidos con la turbina del banco de ensayos.

5. PLANOS

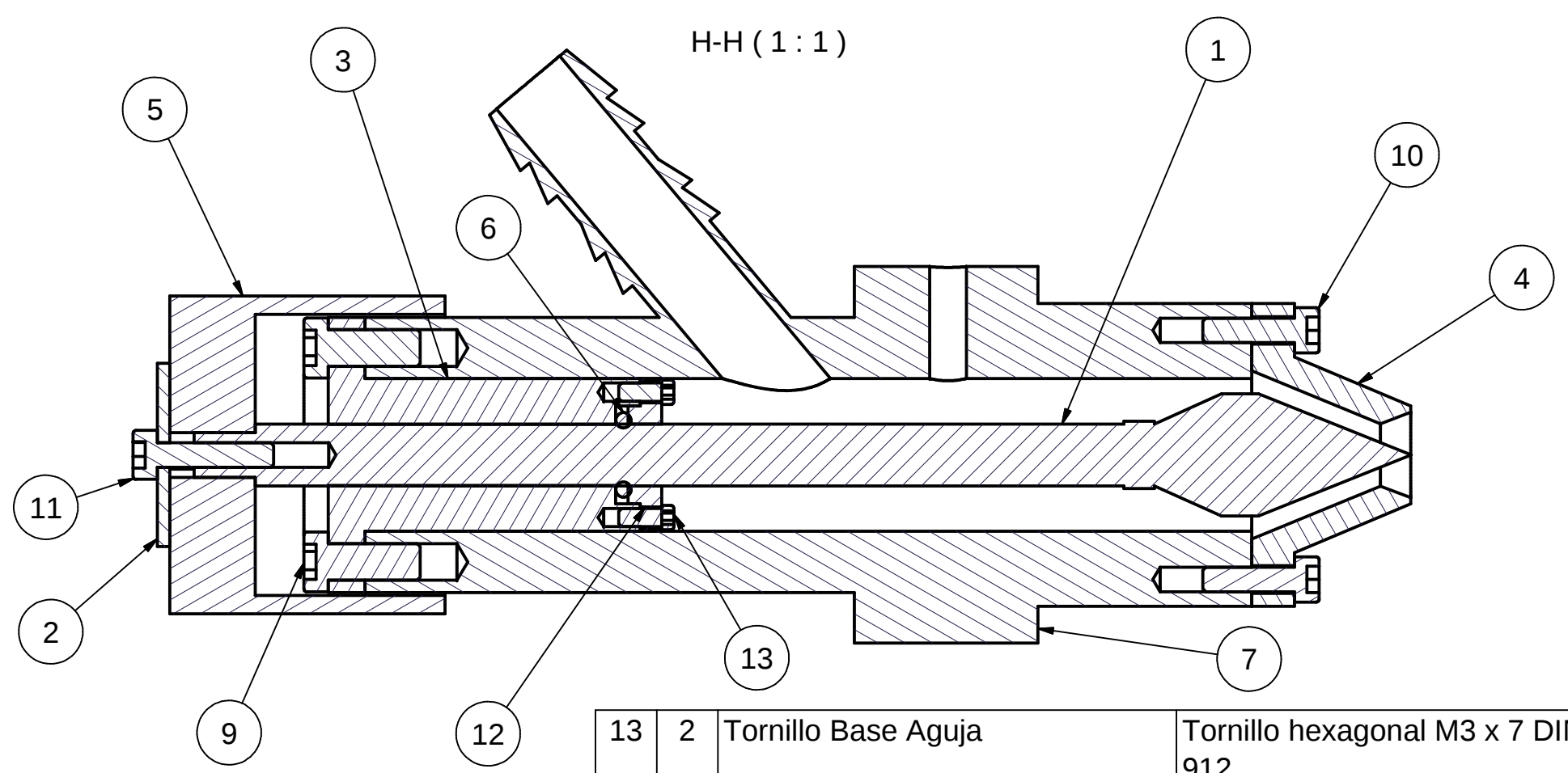
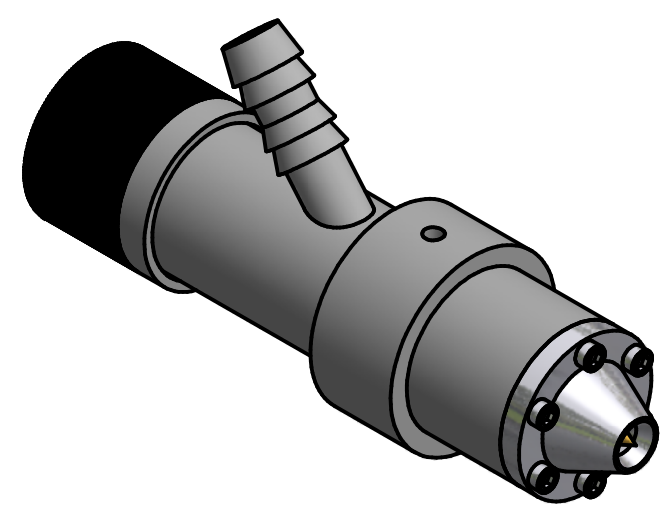
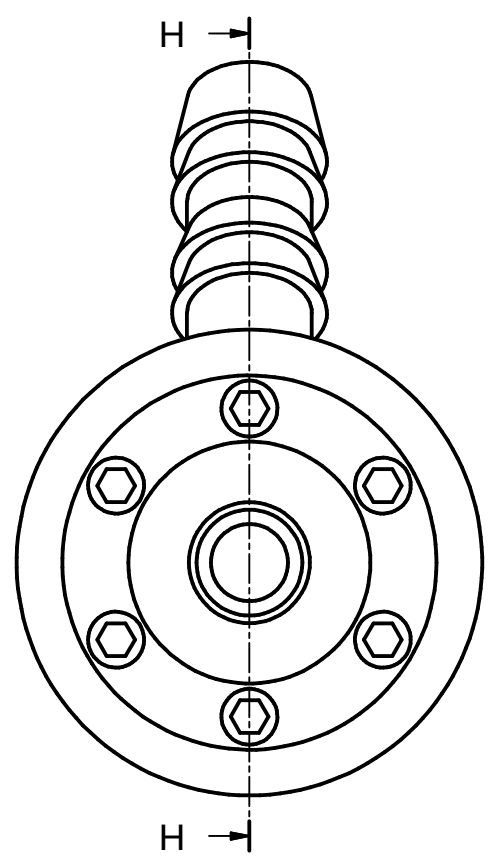
Se adjuntan los planos del inyector actual y del inyector propuesto.

Los primeros me han servido para llevar a cabo las deducciones de la posición de la aguja y el área de paso de agua así como para efectuar las simulaciones numéricas que nos han ayudado a descubrir el por qué de los bajos rendimientos de la turbina Pelton del banco de ensayos.

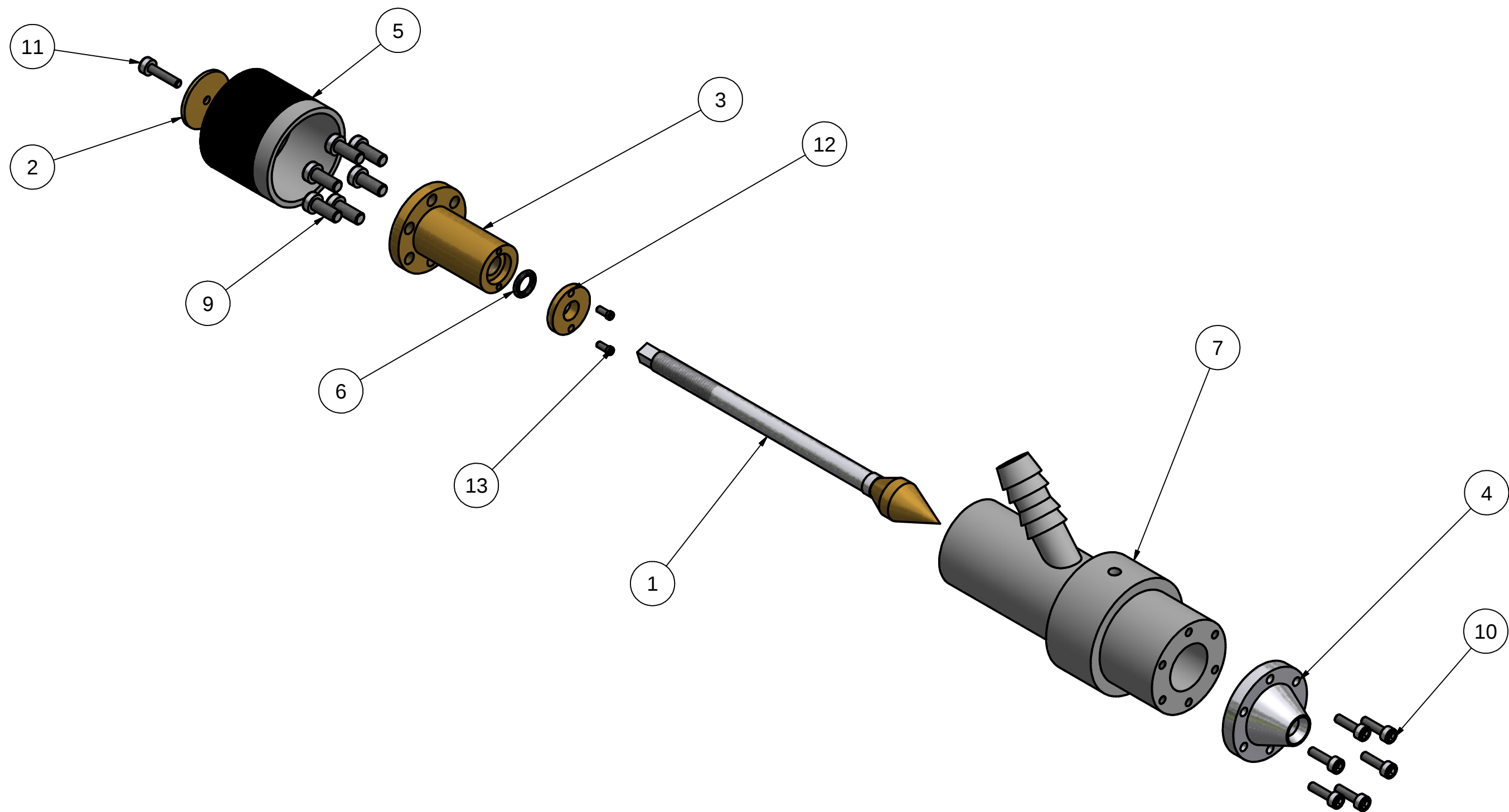
Los planos del inyector propuesto se adjuntan también para que si en un futuro, se quisiera mejorar la máquina o aumentar las capacidades docentes de la turbina, se pudiera construir otro inyector mejorado con el que comparar los resultados del actual inyector.

PRODUCIDO POR UN PRODUCTO EDUCATIVO DE AUTODESK

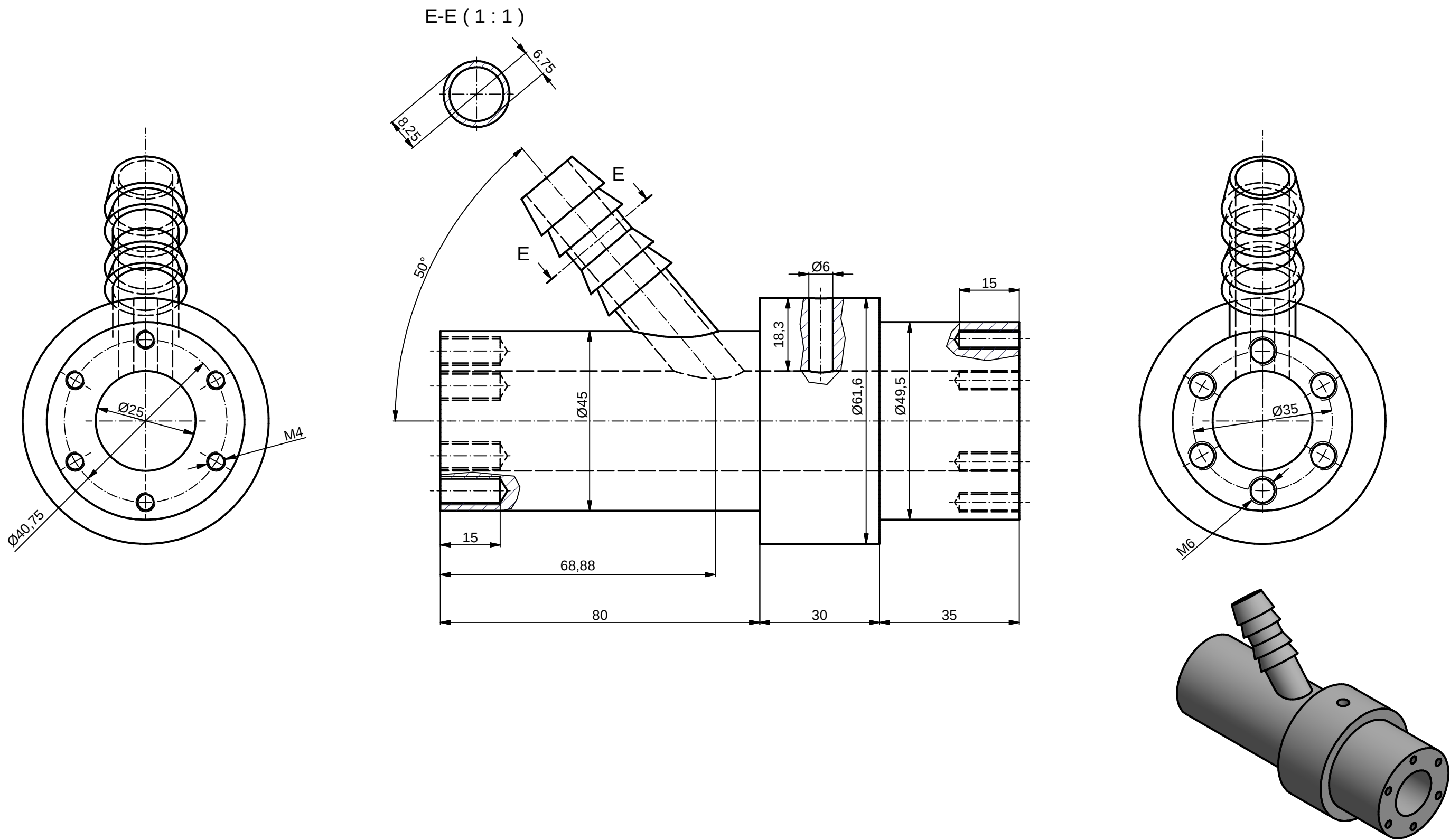
PRODUCIDO POR UN PRODUCTO EDUCATIVO DE AUTODESK



13	2	Tornillo Base Aguja	Tornillo hexagonal M3 x 7 DIN 912
12	1	Tapa	
11	1	Tornillo Aguja	Tornillo hexagonal M4 x 19 DIN 912
10	6	Tornillos Delanteros	Tornillo hexagonal M4 x 15 DIN 912
9	6	Tornillos Traseros	Tornillo hexagonal M6 x 15 DIN 912
7	1	Cuerpo	
6	1	Junta Torica	OR 9 x 2.5 DIN ISO 3601
5	1	Mando	
4	1	Boquilla	
3	1	Base Aguja	
2	1	Arandela Sujeccion	
1	1	Aguja	
Mar	Ctd	Designación	Descripción
		FECHA	NOMBRE
DIBUJADO			 ESCUELA POLITÉCNICA SUPERIOR LINARES UNIVERSIDAD DE JAÉN
COMPROBADO			
C. CALIDAD			
ESCALA 1:1		DESIGNACIÓN	
		Creación de aplicación gráfica con matlab	
		Lista de piezas del inyector	
		Nº DE PLANO	
		1/18	
		SUSTITUYE A	
		SUSTITUIDO POR	



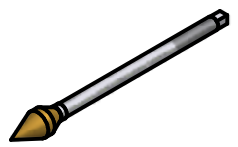
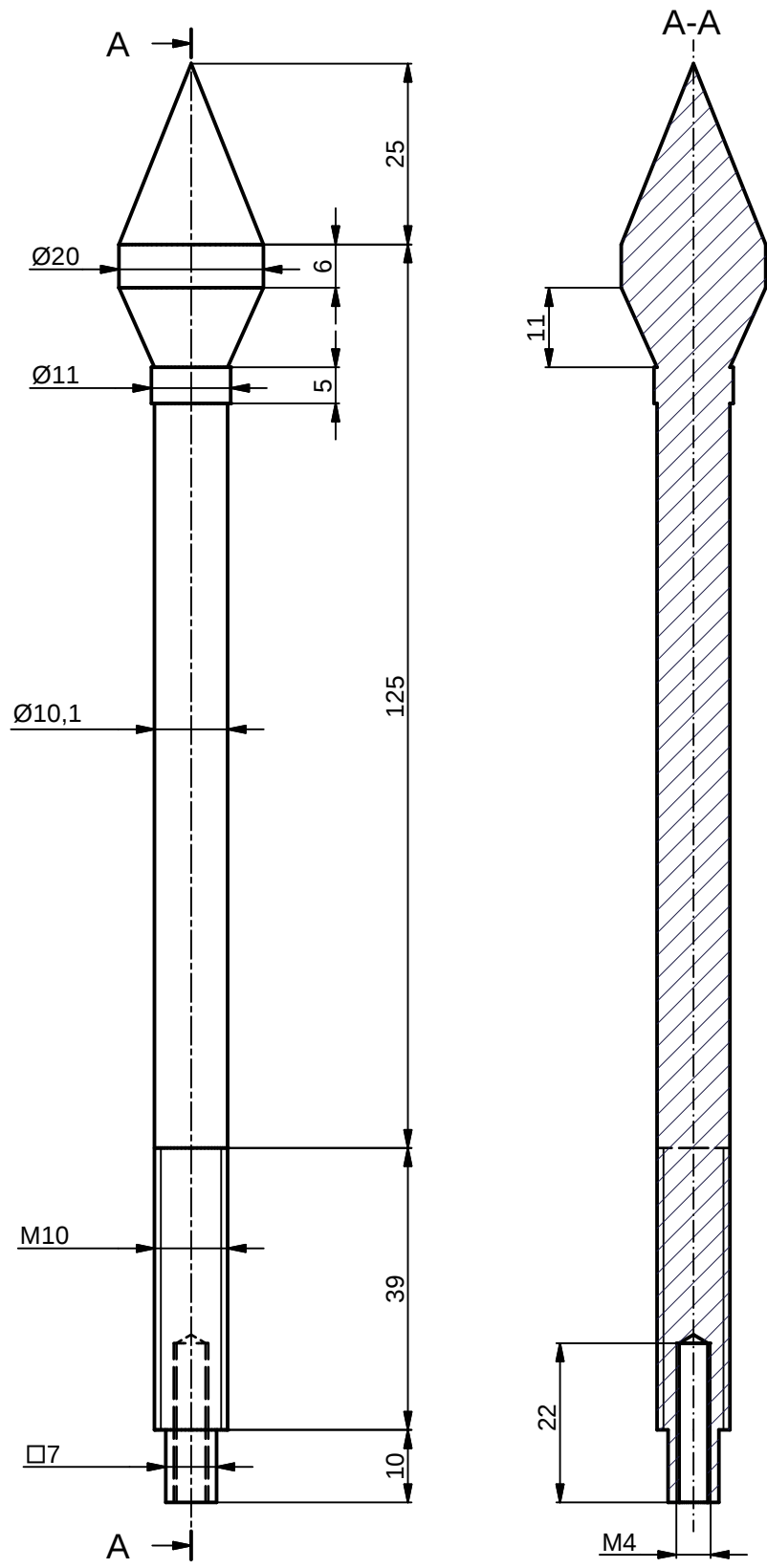
	FECHA	NOMBRE	FIRMAS	 ESCUELA POLITÉCNICA SUPERIOR LINARES UNIVERSIDAD DE JAÉN
DIBUJADO				
COMPROBADO				
C. CALIDAD				
ESCALA 1:2	DESIGNACIÓN			Nº DE PLANO
	Creación de aplicación gráfica con matlab			2/18
	Vista explosionada			SUSTITUYE A
				SUSTITUIDO POR



	FECHA	NOMBRE	FIRMAS	 ESCUELA POLITÉCNICA SUPERIOR LINARES UNIVERSIDAD DE JAÉN
DIBUJADO				
COMPROBADO				
C. CALIDAD				
ESCALA 1:1	DESIGNACIÓN Creación de aplicación gráfica con matlab Cuerpo			Nº DE PLANO 3/18
				SUSTITUYE A
				SUSTITUIDO POR

PRODUCIDO POR UN PRODUCTO EDUCATIVO DE AUTODESK

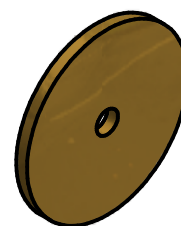
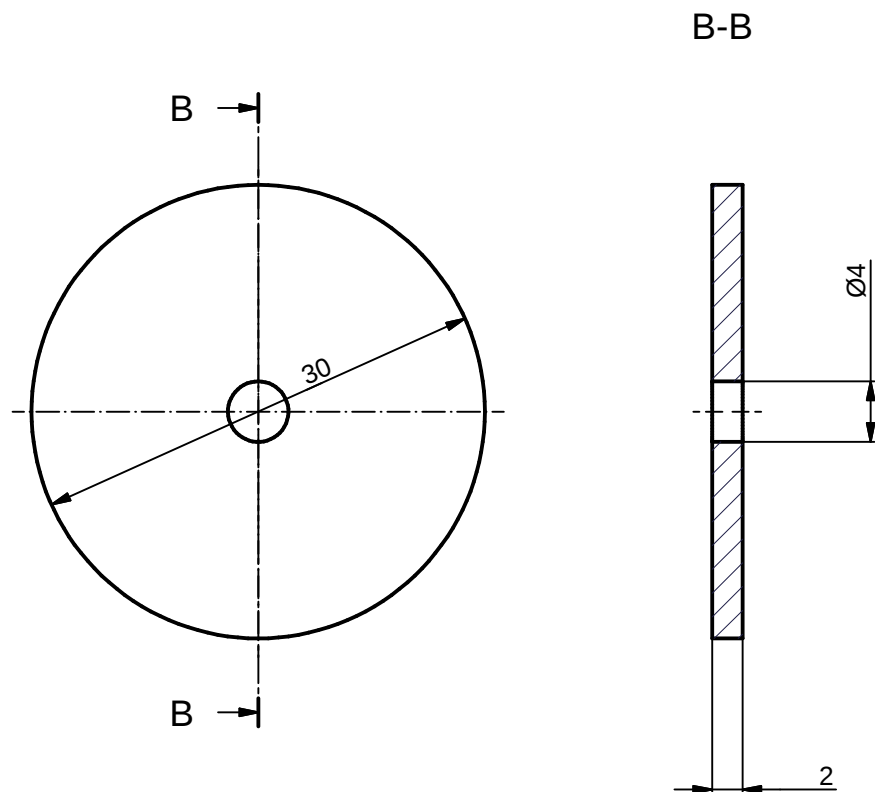
PRODUCIDO POR UN PRODUCTO EDUCATIVO DE AUTODESK



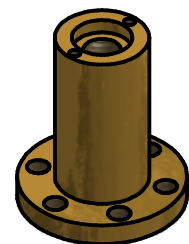
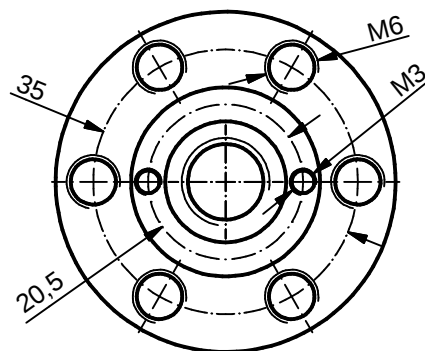
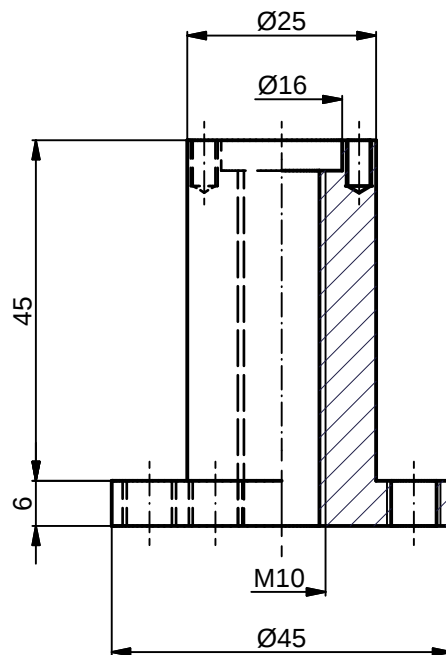
	FECHA	NOMBRE	FIRMAS	 ESCUELA POLITÉCNICA SUPERIOR LINARES UNIVERSIDAD DE JAÉN
DIBUJADO				
COMPROBADO				
C. CALIDAD				
ESCALA 1:1	DESIGNACIÓN			Nº DE PLANO
	Creación de aplicación gráfica con matlab			4/18
	Aguja			SUSTITUYE A
				SUSTITUIDO POR

PRODUCIDO POR UN PRODUCTO EDUCATIVO DE AUTODESK

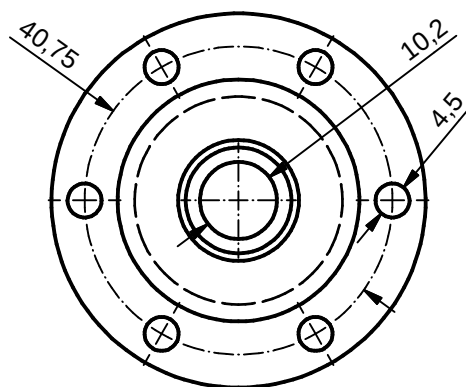
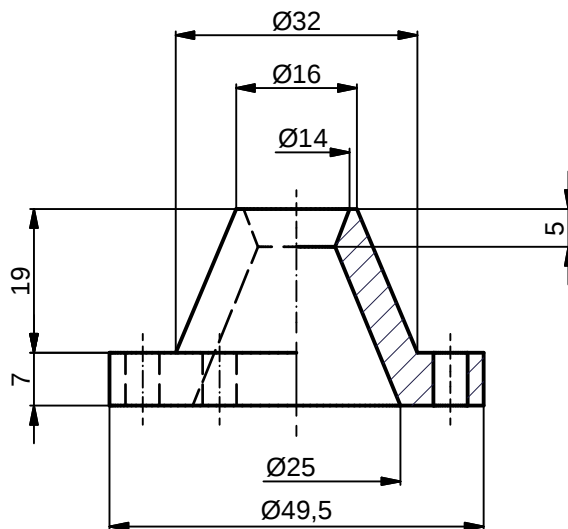
PRODUCIDO POR UN PRODUCTO EDUCATIVO DE AUTODESK



	FECHA	NOMBRE	FIRMAS	 ESCUELA POLITÉCNICA SUPERIOR LINARES UNIVERSIDAD DE JAÉN
DIBUJADO				
COMPROBADO				
C. CALIDAD				
ESCALA 2:1	DESIGNACIÓN Creación de aplicación gráfica con matlab Arandela sujección			Nº DE PLANO 5/18
				SUSTITUYE A
				SUSTITUIDO POR

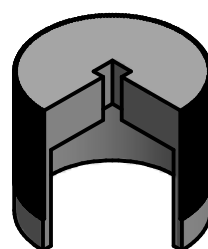
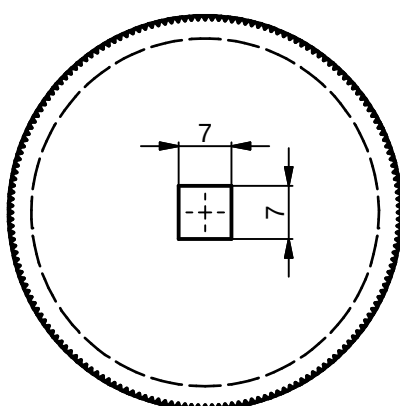
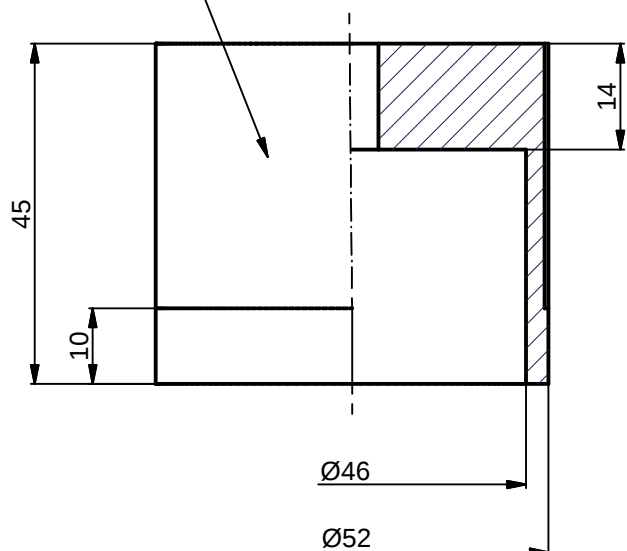


	FECHA	NOMBRE	FIRMAS	 ESCUELA POLITÉCNICA SUPERIOR LINARES UNIVERSIDAD DE JAÉN	
DIBUJADO					
COMPROBADO					
C. CALIDAD					
ESCALA 1:1	DESIGNACIÓN Creación de aplicación gráfica con matlab Base aguja				Nº DE PLANO 6/18
					SUSTITUYE A
					SUSTITUIDO POR

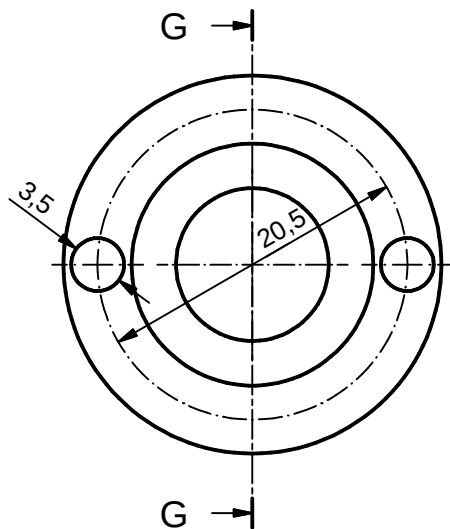


	FECHA	NOMBRE	FIRMAS	 ESCUELA POLITÉCNICA SUPERIOR LINARES UNIVERSIDAD DE JAÉN	
DIBUJADO					
COMPROBADO					
C. CALIDAD					
ESCALA 1:1	DESIGNACIÓN Creación de aplicación gráfica con matlab Tobera				Nº DE PLANO 7/18
					SUSTITUYE A
					SUSTITUIDO POR

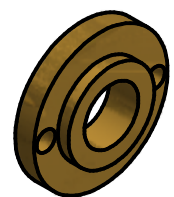
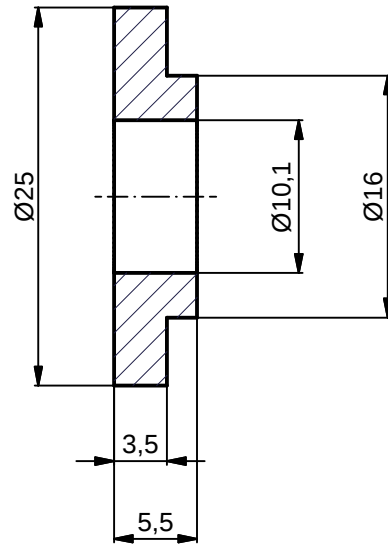
MOLETEADO



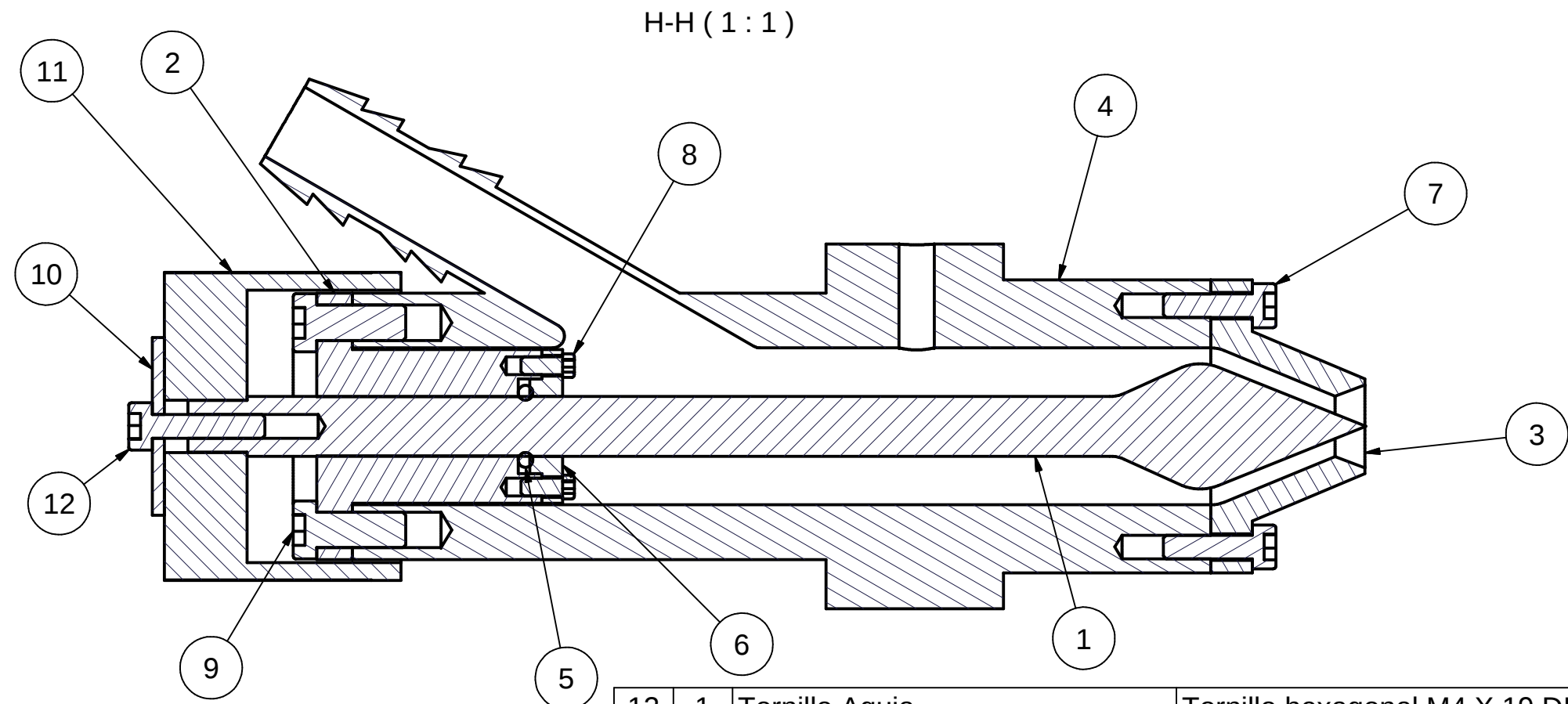
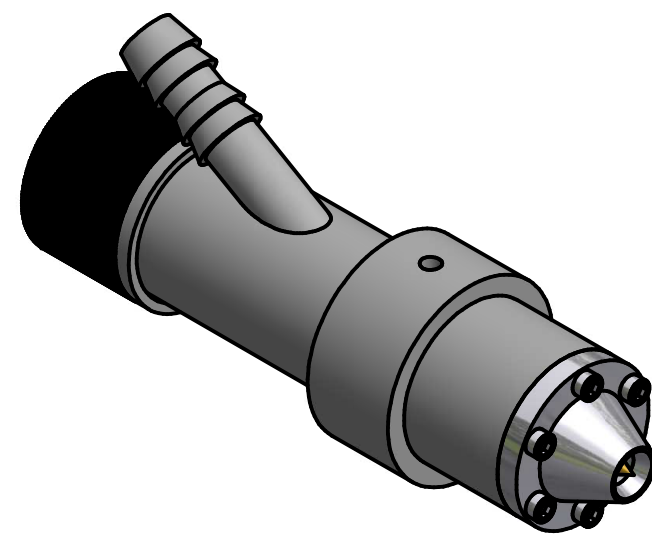
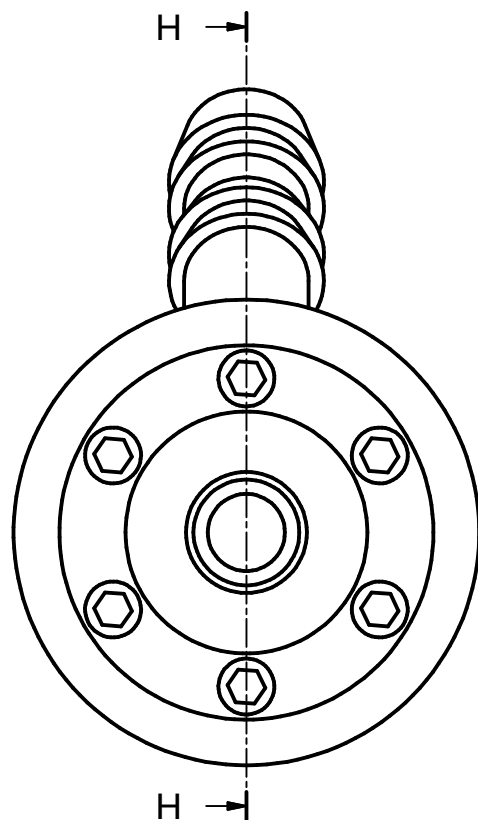
	FECHA	NOMBRE	FIRMAS	 ESCUELA POLITÉCNICA SUPERIOR LINARES UNIVERSIDAD DE JAÉN	
DIBUJADO					
COMPROBADO					
C. CALIDAD					
ESCALA 1:1	DESIGNACIÓN Creación de aplicación gráfica con matlab Mando regulador				Nº DE PLANO 8/18
					SUSTITUYE A
					SUSTITUIDO POR



G-G (2 : 1)

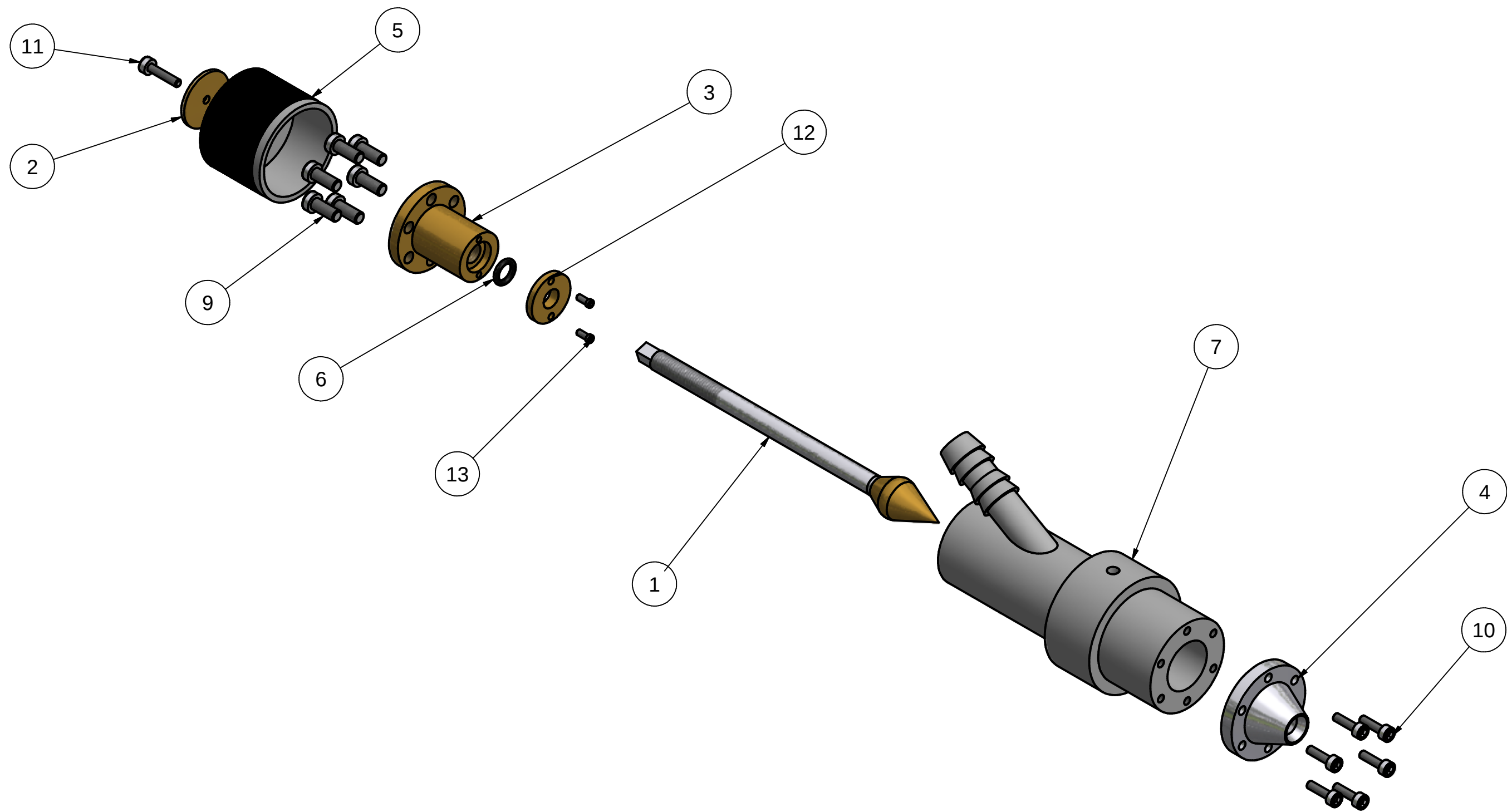


	FECHA	NOMBRE	FIRMAS	 ESCUELA POLITÉCNICA SUPERIOR LINARES UNIVERSIDAD DE JAÉN	
DIBUJADO					
COMPROBADO					
C. CALIDAD					
ESCALA 2:1	DESIGNACIÓN Creación de aplicación gráfica con matlab Tapa				Nº DE PLANO 9/18
					SUSTITUYE A
					SUSTITUIDO POR

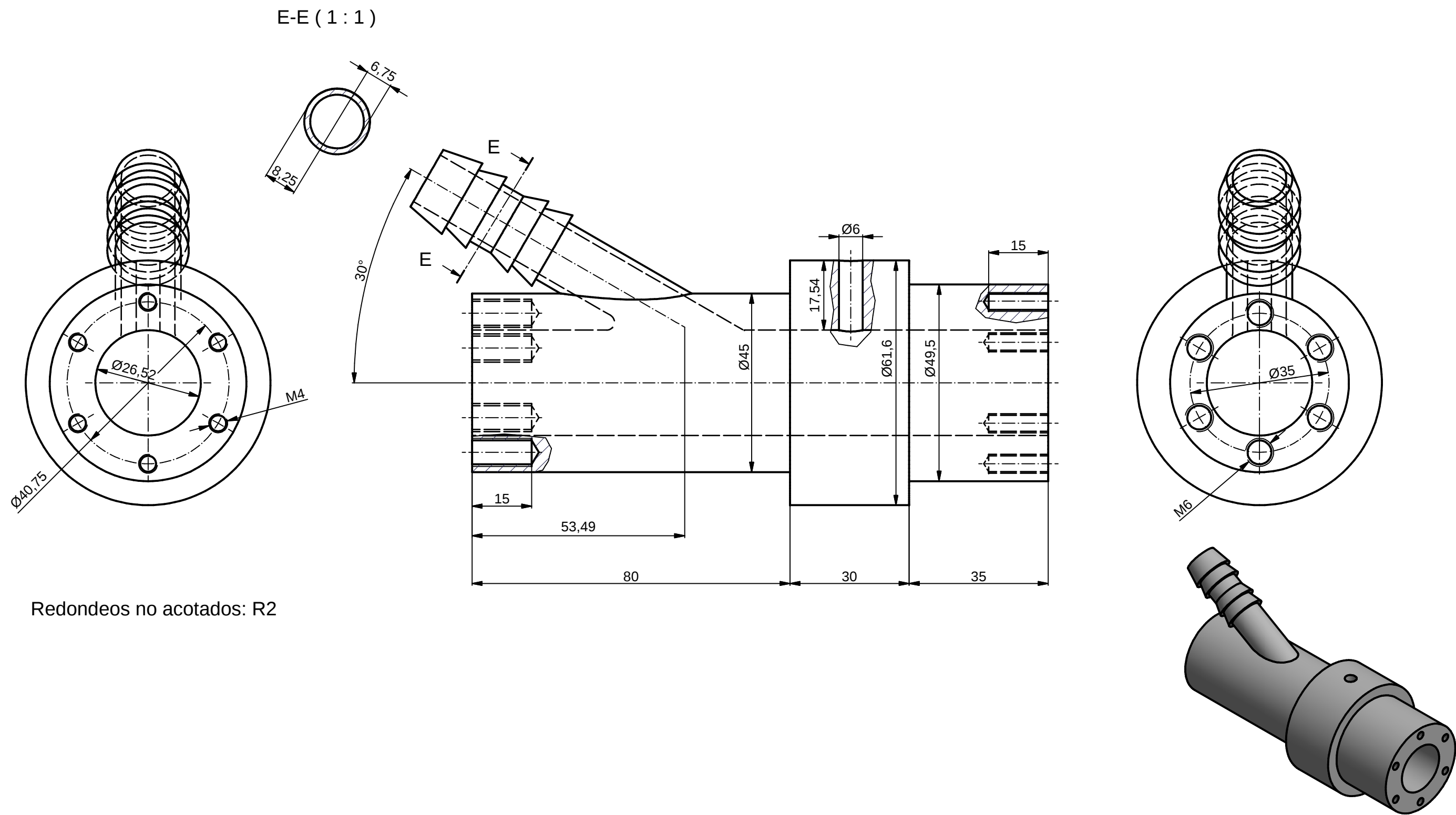


12	1	Tornillo Aguja	Tornillo hexagonal M4 X 19 DIN 912
11	1	Mando	
10	1	Arandela Sujeccion	
9	6	Tornillos Traseros	Tornillo hexagonal M6 x 15 DIN 912
8	2	Tornillos Base Aguja	Tornillo hexagonal M3 x 7 DIN 912
7	6	Tornillos Delanteros	Tornillo hexagonal M4 x 15 DIN 912
6	1	Tapa	
5	1	Junta torica	OR 9 x 2.5 DIN ISO 3601
4	1	Cuerpo	
3	1	Boquilla	
2	1	Base Aguja	
1	1	Aguja	

Mar	Ctd	Designación			Descripción	
		FECHA	NOMBRE	FIRMAS	 ESCUOLA POLITÉCNICA SUPERIOR LINARES UNIVERSIDAD DE JAÉN	
DIBUJADO						
COMPROBADO						
C. CALIDAD						
ESCALA 1:1		DESIGNACIÓN Creación de aplicación gráfica con matlab Lista inyector propuesto			Nº DE PLANO 10/18	
					SUSTITUYE A	
					SUSTITUIDO POR	



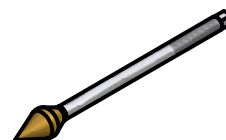
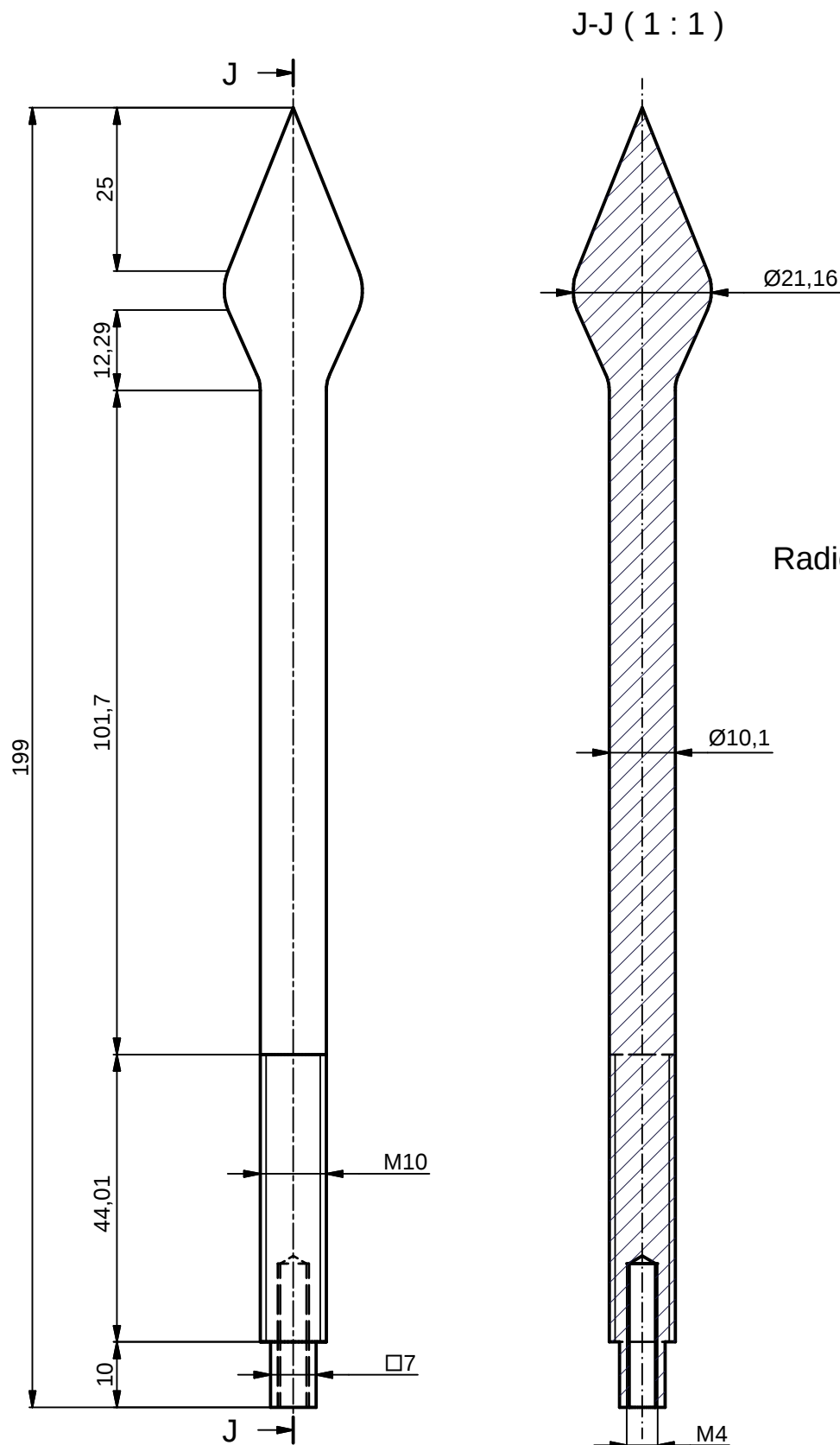
	FECHA	NOMBRE	FIRMAS	 ESCUELA POLITÉCNICA SUPERIOR LINARES UNIVERSIDAD DE JAÉN	
DIBUJADO					
COMPROBADO					
C. CALIDAD					
ESCALA 1:2	DESIGNACIÓN			Nº DE PLANO	
	Creación de aplicación gráfica con matlab			11/18	
	Vista explosionada (propuesto)			SUSTITUYE A	
				SUSTITUIDO POR	



	FECHA	NOMBRE	FIRMAS	 ESCUELA POLITÉCNICA SUPERIOR LINARES UNIVERSIDAD DE JAÉN
DIBUJADO				
COMPROBADO				
C. CALIDAD				
ESCALA 1:1	DESIGNACIÓN			Nº DE PLANO
	Creación de aplicación gráfica con matlab			12/18
	Cuerpo (propuesto)			SUSTITUYE A
				SUSTITUIDO POR

PRODUCIDO POR UN PRODUCTO EDUCATIVO DE AUTODESK

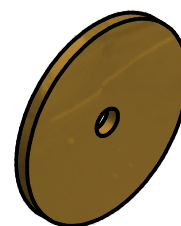
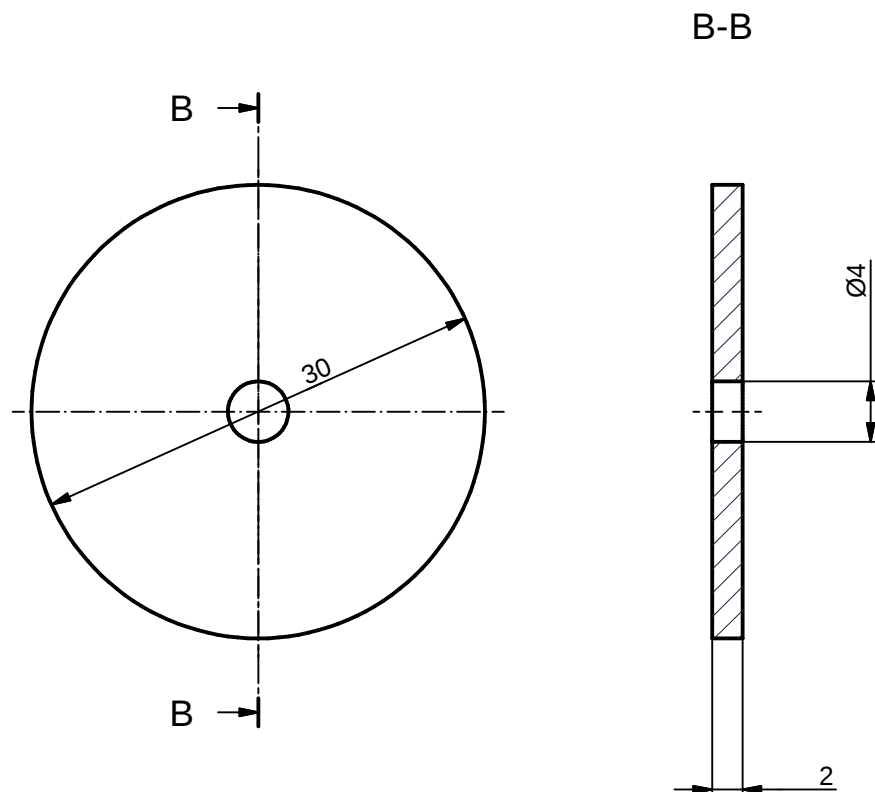
PRODUCIDO POR UN PRODUCTO EDUCATIVO DE AUTODESK



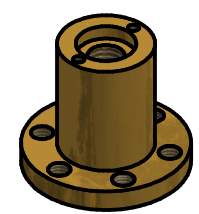
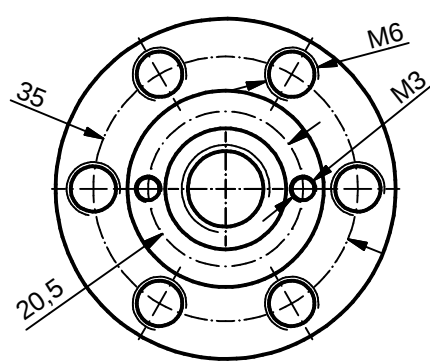
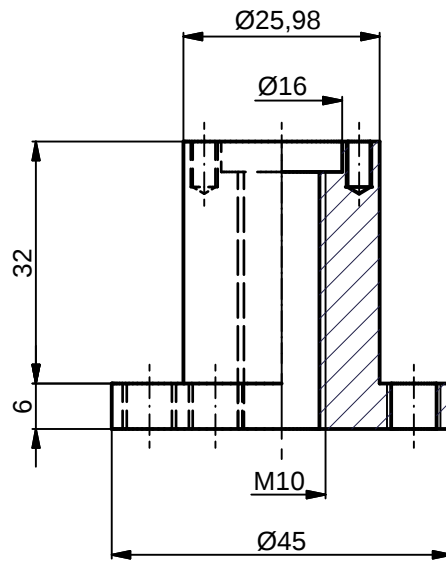
	FECHA	NOMBRE	FIRMAS	 ESCUELA POLITÉCNICA SUPERIOR LINARES UNIVERSIDAD DE JAÉN	
DIBUJADO					
COMPROBADO					
C. CALIDAD					
ESCALA 1:1	DESIGNACIÓN Creación de aplicación gráfica con matlab Aguja (propuesto)				Nº DE PLANO 13/18
					SUSTITUYE A
					SUSTITUIDO POR

PRODUCIDO POR UN PRODUCTO EDUCATIVO DE AUTODESK

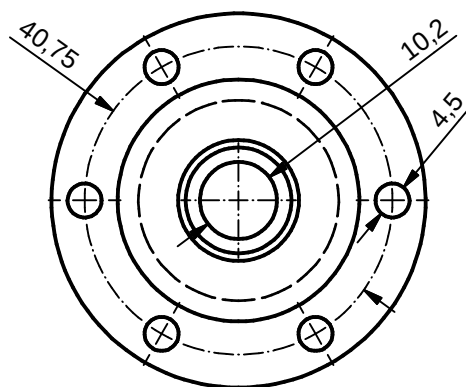
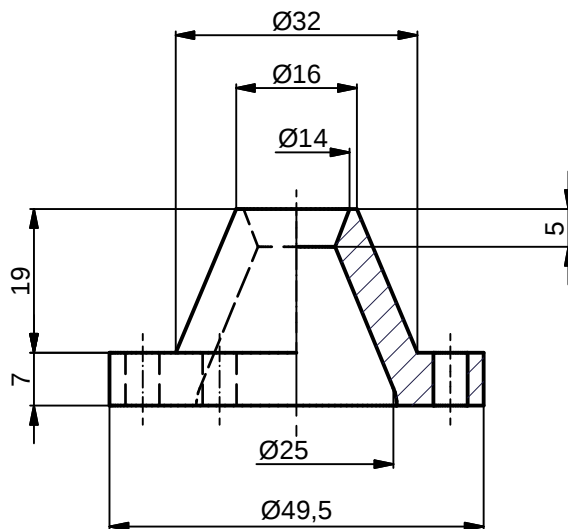
PRODUCIDO POR UN PRODUCTO EDUCATIVO DE AUTODESK



	FECHA	NOMBRE	FIRMAS	 ESCUELA POLITÉCNICA SUPERIOR LINARES UNIVERSIDAD DE JAÉN
DIBUJADO				
COMPROBADO				
C. CALIDAD				
ESCALA 2:1	DESIGNACIÓN Creación de aplicación gráfica con matlab Arandela sujección (propuesto)			Nº DE PLANO 14/18
				SUSTITUYE A
				SUSTITUIDO POR

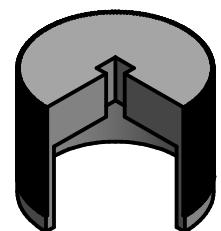
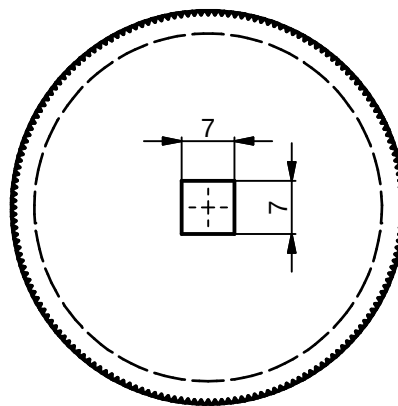
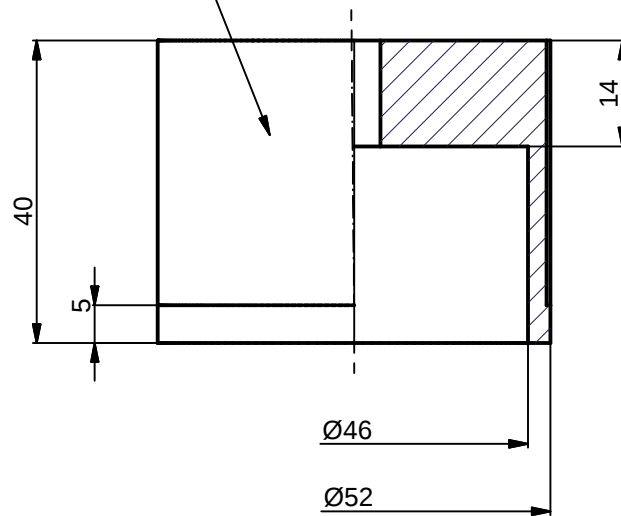


	FECHA	NOMBRE	FIRMAS	 ESCUELA POLITÉCNICA SUPERIOR LINARES UNIVERSIDAD DE JAÉN
DIBUJADO				
COMPROBADO				
C. CALIDAD				
ESCALA 1:1	DESIGNACIÓN Creación de aplicación gráfica con matlab Base aguja (propuesto)			Nº DE PLANO 15/18
				SUSTITUYE A
				SUSTITUIDO POR

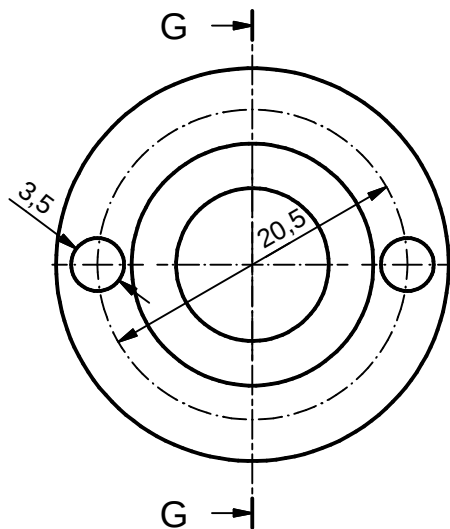


	FECHA	NOMBRE	FIRMAS	 ESCUELA POLITÉCNICA SUPERIOR LINARES UNIVERSIDAD DE JAÉN	
DIBUJADO					
COMPROBADO					
C. CALIDAD					
ESCALA 1:1	DESIGNACIÓN Creación de aplicación gráfica con matlab Tobera (propuesto)				Nº DE PLANO 16/18
					SUSTITUYE A
					SUSTITUIDO POR

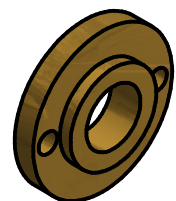
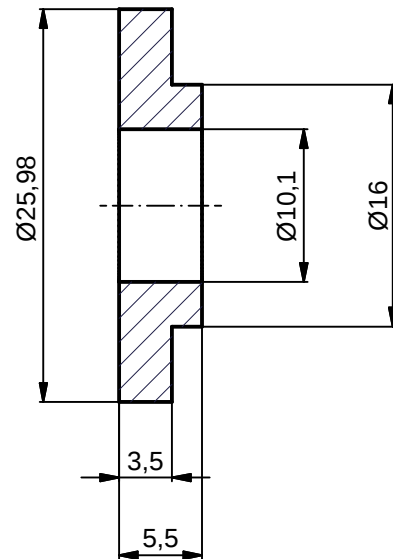
MOLETEADO



	FECHA	NOMBRE	FIRMAS	 ESCUELA POLITÉCNICA SUPERIOR LINARES UNIVERSIDAD DE JAÉN	
DIBUJADO					
COMPROBADO					
C. CALIDAD					
ESCALA 1:1	DESIGNACIÓN Creación de aplicación gráfica con matlab Mando regulador (propuesto)				Nº DE PLANO 17/18
					SUSTITUYE A
					SUSTITUIDO POR



G-G (2 : 1)



	FECHA	NOMBRE	FIRMAS	 ESCUELA POLITÉCNICA SUPERIOR LINARES UNIVERSIDAD DE JAÉN	
DIBUJADO					
COMPROBADO					
C. CALIDAD					
ESCALA 2:1	DESIGNACIÓN Creación de aplicación gráfica con matlab Tapa (propuesta)				Nº DE PLANO 18/18
					SUSTITUYE A
					SUSTITUIDO POR

6. ANEXOS

6.1 Código de la interfaz gráfica

Aquí expongo el distinto código de la interfaz gráfica ordenado en función de las distintas ventanas de las que se compone.

6.1.1 Código de la ventana de Inicio.

```
% --- Executes just before inicio is made visible.
function inicio_OpeningFcn(hObject, eventdata, handles, varargin)

%Fondo1
axes(handles.axes4)
background = imread('1800.jpg');
axis off;
imshow(background);

% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin   command line arguments to inicio (see VARARGIN)

% Choose default command line output for inicio
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes inicio wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = inicio_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in AYUDA.
function AYUDA_Callback(hObject, eventdata, handles)
% hObject    handle to AYUDA (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
open('ayuda.pdf');

% --- Executes on button press in inicio.
function inicio_Callback(hObject, eventdata, handles)
% hObject    handle to inicio (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```

% handles    structure with handles and user data (see GUIDATA)
close all;
run('menu'); %abre menu

% --- Executes on button press in salir.
function salir_Callback(hObject, eventdata, handles)
% hObject    handle to salir (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
opc=questdlg('¿Desea salir del programa?', 'SALIR', 'Si', 'No', 'No');
if strcmp(opc, 'No')
return;
end
clear,clc,close all

% --- Executes during object creation, after setting all properties.
function fondo0_CreateFcn(hObject, eventdata, handles)
% hObject    handle to fondo0 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: place code in OpeningFcn to populate fondo0

% --- Executes during object creation, after setting all properties.
function axes1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to axes1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: place code in OpeningFcn to populate axes1

% --- Executes during object creation, after setting all properties.
function figure1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to figure1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

```

6.1.2 Código de la ventana del Menú Principal.

```

% --- Executes just before menu is made visible.
function menu_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin    command line arguments to menu (see VARARGIN)

% Choose default command line output for menu
handles.output = hObject;
% Update handles structure
guidata(hObject, handles);

%Fondo
axes(handles.axes2)
background = imread('FME16.jpg');

```

```

axis off;
imshow(background);

%Coloca una imagen en boton atrás
[a,map]=imread('1.jpg');
[r,c,d]=size(a);
x=ceil(r/38);
y=ceil(c/110);
g=a(1:x:end,1:y:end,:);
g(g==255)=5.5*255;
set(handles.atras,'CData',g);

% UIWAIT makes menu wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = menu_OutputFcn(hObject, eventdata, handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject     handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function edit1_Callback(hObject, eventdata, handles)
% hObject     handle to edit1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%         str2double(get(hObject,'String')) returns contents of edit1 as a
double

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject     handle to edit1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton9.
function pushbutton9_Callback(hObject, eventdata, handles)
% hObject     handle to pushbutton9 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% --- Executes on button press in practica.
function practica_Callback(hObject, eventdata, handles)
% hObject     handle to practica (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

```

```

open('Pelton.pdf');

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in historia.
function historia_Callback(hObject, eventdata, handles)
% hObject    handle to historia (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close all;
run('inyector');

% --- Executes on button press in ejemplosreales.
function ejemplosreales_Callback(hObject, eventdata, handles)
% hObject    handle to ejemplosreales (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
open('Ejemplos_reales.pdf');

```

```

% --- Executes on button press in informacion.
function informacion_Callback(hObject, eventdata, handles)
% hObject      handle to informacion (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
open('Ramon_Abella.pdf');

% --- Executes on button press in pushbutton19.
function pushbutton19_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton19 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
open('ayuda.pdf');

% --- Executes on button press in video.
function video_Callback(hObject, eventdata, handles)
% hObject      handle to video (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% --- Executes on button press in graficos.
function graficos_Callback(hObject, eventdata, handles)
% hObject      handle to graficos (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
close all;
run('gcs'); % abre curvas1

% --- Executes on button press in datos.
function datos_Callback(hObject, eventdata, handles)
% hObject      handle to datos (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
open('DATOS_Alumno.xlsx');

% --- Executes on button press in teoria.
function teoria_Callback(hObject, eventdata, handles)
% hObject      handle to teoria (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
open('teoria.pdf'); % abre teoria

% --- Executes during object creation, after setting all properties.
function axes1_CreateFcn(hObject, eventdata, handles)
% hObject      handle to axes1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: place code in OpeningFcn to populate axes1

% -----
function Untitled_1_Callback(hObject, eventdata, handles)
% hObject      handle to Untitled_1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
open('ayuda.pdf');

% --- Executes during object creation, after setting all properties.
function fondo1_CreateFcn(hObject, eventdata, handles)

```

```

% hObject    handle to fondo1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: place code in OpeningFcn to populate fondo1

% -----
function uipanel10_ButtonDownFcn(hObject, eventdata, handles)
% hObject    handle to uipanel10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in atras.
function atras_Callback(hObject, eventdata, handles)
% hObject    handle to atras (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close all;
run('inicio'); % abre inicio

% --- Executes during object creation, after setting all properties.
function uipanel10_CreateFcn(hObject, eventdata, handles)
% hObject    handle to uipanel10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% --- Executes during object creation, after setting all properties.
function fondo10_CreateFcn(hObject, eventdata, handles)
% hObject    handle to fondo10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: place code in OpeningFcn to populate fondo10

% --- Executes during object creation, after setting all properties.
function fondo_menu_CreateFcn(hObject, eventdata, handles)
% hObject    handle to fondo_menu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: place code in OpeningFcn to populate fondo_menu

% --- Executes during object creation, after setting all properties.
function figure1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to figure1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% --- Executes on button press in pushbutton23.
function pushbutton23_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton23 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```
% --- Executes during object creation, after setting all properties.
function axes2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to axes2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
%Carga imagen de fondo

% Hint: place code in OpeningFcn to populate axes2

% --- Executes during object creation, after setting all properties.
function pushbutton19_CreateFcn(hObject, eventdata, handles)
% hObject    handle to pushbutton19 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
```

6.1.3 Código de la ventana del botón de Ensayo.

```
% --- Executes just before gcs is made visible.
function gcs_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to gcs (see VARARGIN)

% Choose default command line output for gcs
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

axes(handles.axes1)
background = imread('PIC9.jpg');
axis off;
imshow(background);

%Coloca una imagen en boton atrás
[a,map]=imread('1.jpg');
[r,c,d]=size(a);
x=ceil(r/38);
y=ceil(c/110);
g=a(1:x:end,1:y:end,:);
g(g==255)=5.5*255;
set(handles.atras, 'CData',g);

% UIWAIT makes gcs wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = gcs_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close all;
run('curvas2');

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in atras.
function atras_Callback(hObject, eventdata, handles)
% hObject    handle to atras (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close all;
run('menu');

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close all;
run('coef');

% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close all;
run('sem');

% -----
function Untitled_1_Callback(hObject, eventdata, handles)
% hObject    handle to Untitled_1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
open('ayuda.pdf');

```

6.1.4 Código de la ventana Gráficas.

```

% --- Executes just before curvas2 is made visible.
function curvas2_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to curvas2 (see VARARGIN)

% Choose default command line output for curvas2
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

%Coloca una imagen en boton atrás
[a,map]=imread('1.jpg');
[r,c,d]=size(a);
x=ceil(r/28);
y=ceil(c/109);
g=a(1:x:end,1:y:end,:);
g(g==255)=5.5*255;
set(handles.atras,'CData',g);

warning ('off');

% UIWAIT makes curvas2 wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = curvas2_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in atras.
function atras_Callback(hObject, eventdata, handles)
% hObject    handle to atras (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
close all;
run('gcs');

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% --- Executes on button press in bor2.
function bor2_Callback(hObject, eventdata, handles)
% hObject    handle to bor2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% --- Executes on button press in plot2.

```

```

function plot2_Callback(hObject, eventdata, handles)
% hObject      handle to plot2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

%% TFG Dimensional Pos1 Par
if (get(handles.tfg2,'Value')==1 && get(handles.dim2,'Value')==1 &&
get(handles.pos2,'Value')==1 && get(handles.graf2,'Value')==1 &&
get(handles.caudal2,'Value')==4);%Todas las graficas Par
    axes(handles.axes2);%Limpia ejes

    filename = 'DATOS_matlab.xlsx';
    sheet = 1;
    xlRange = 'A5:C23';

    q1 = xlsread(filename, sheet, xlRange);

    filename = 'DATOS_matlab.xlsx';
    sheet = 1;
    xlRange = 'A31:C45';

    q2 = xlsread(filename, sheet, xlRange);

    filename = 'DATOS_matlab.xlsx';
    sheet = 1;
    xlRange = 'A53:C63';

    q3 = xlsread(filename, sheet, xlRange);

    fdifq1=(q1(:,3)-q1(:,2))*(0.05/2);
    fdifq2=(q2(:,3)-q2(:,2))*(0.05/2);
    fdifq3=(q3(:,3)-q3(:,2))*(0.05/2);

    pq1=polyfit(q1(:,1),fdifq1,1);
    pq2=polyfit(q2(:,1),fdifq2,1);
    pq3=polyfit(q3(:,1),fdifq3,1);

    axes(handles.axes2),plot(q1(:,1),fdifq1,'-o',q2(:,1),fdifq2,'-
o',q3(:,1),fdifq3,'-o',q1(:,1),polyval(pq1,q1(:,1)),'-
k*',q2(:,1),polyval(pq2,q2(:,1)),'-k*',q3(:,1),polyval(pq3,q3(:,1)),'-
k*'),grid on,set(gca, 'FontSize',
8),xlabel('RPM','FontSize',10),ylabel('Par
C(Nm)','FontSize',10),title('Par C(Nm)','FontSize',12),legend('Par
Q1','Par Q2','Par Q3','Ajuste Q1','Ajuste Q2','Ajuste Q3');,hold on;

elseif (get(handles.tfg2,'Value')==1 && get(handles.dim2,'Value')==1 &&
get(handles.pos2,'Value')==1 && get(handles.graf2,'Value')==1 &&
get(handles.caudal2,'Value')==1);%Par Q1
    axes(handles.axes2);%Limpia ejes

    filename = 'DATOS_matlab.xlsx';
    sheet = 1;
    xlRange = 'A5:C23';

```

```

q1 = xlsread(filename, sheet, xlRange);

fdifq1=(q1(:,3)-q1(:,2))*(0.05/2);

pq1=polyfit(q1(:,1),fdifq1,1);

axes(handles.axes2),plot(q1(:,1),fdifq1,'-
o',q1(:,1),polyval(pq1,q1(:,1)),'-k*'),grid on,set(gca, 'FontSize',
8),xlabel('RPM','FontSize',10),ylabel('Par
C(Nm)','FontSize',10),title('Par C(Nm)','FontSize',12),legend('Par
Q1','Ajuste Q1');,hold on;

%% TFG Dimensional Pos1 Wu
if (get(handles.tfg2,'Value')==1 && get(handles.dim2,'Value')==1 &&
get(handles.pos2,'Value')==1 && get(handles.graf2,'Value')==2 &&
get(handles.caudal2,'Value')==4);%Todas las graficas Wu
    axes(handles.axes2);%Limpia ejes

filename = 'DATOS_matlab.xlsx';
sheet = 1;
xlRange = 'A5:C23';

q1 = xlsread(filename, sheet, xlRange);

filename = 'DATOS_matlab.xlsx';
sheet = 1;
xlRange = 'A31:C45';

q2 = xlsread(filename, sheet, xlRange);

filename = 'DATOS_matlab.xlsx';
sheet = 1;
xlRange = 'A53:C63';

q3 = xlsread(filename, sheet, xlRange);

fdifq1=(q1(:,3)-q1(:,2))*(0.05/2);
fdifq2=(q2(:,3)-q2(:,2))*(0.05/2);
fdifq3=(q3(:,3)-q3(:,2))*(0.05/2);

wuq1=fdifq1.*q1(:,1)*2*pi/60;
wuq2=fdifq2.*q2(:,1)*2*pi/60;
wuq3=fdifq3.*q3(:,1)*2*pi/60;

wq1=polyfit(q1(:,1),wuq1,3);
wq2=polyfit(q2(:,1),wuq2,3);
wq3=polyfit(q3(:,1),wuq3,3);

axes(handles.axes2),plot(q1(:,1),wuq1,'-o',q2(:,1),wuq2,'-
o',q3(:,1),wuq3,'-o',q1(:,1),polyval(wq1,q1(:,1)),'-
k*',q2(:,1),polyval(wq2,q2(:,1)),'-k*',q3(:,1),polyval(wq3,q3(:,1)),'-
k*'),grid on,set(gca, 'FontSize',
8),xlabel('RPM','FontSize',10),ylabel('W_u(W)','FontSize',10),title('Pote
ncia Útil W_u(W)','FontSize',12),legend('W_u Q1','W_u Q2','W_u
Q3','Ajuste Q1','Ajuste Q2','Ajuste Q3');,hold on;

```



```

fdifq1=(q1(:,3)-q1(:,2))*0.05/2;

wuq1=fdifq1.*q1(:,1)*2*pi/60;

whq1=(9.81*1000*qh1(:,2)*10^-3/3600*qh1(:,3))*v1;% por un vector de unos
en columna para que sea matriz

nq1=wuq1/whq1*v1;% por un vector de unos en columna para que sea matriz

np1=polyfit(q1(:,1),nq1,3);

axes(handles.axes2),plot(q1(:,1),nq1,'-o',q1(:,1),polyval(np1,q1(:,1)),'-
k*'),grid on,set(gca,'FontSize',
8),xlabel('RPM','FontSize',10),ylabel('\eta','FontSize',10),title('Rendim
iento \eta','FontSize',12),legend('\eta Q1','Ajuste Q1');,hold on;

```

6.1.5 Código de la ventana Coeficiente de desagüe.

```

% --- Executes just before coef is made visible.
function coef_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to coef (see VARARGIN)

% Choose default command line output for coef
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

%Coloca una imagen en boton atrás
[a,map]=imread('1.jpg');
[r,c,d]=size(a);
x=ceil(r/38);
y=ceil(c/110);
g=a(1:x:end,1:y:end,:);
g(g==255)=5.5*255;
set(handles.atras,'CData',g);

% UIWAIT makes coef wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = coef_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.

```

```

function pushbutton1_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

%%%%%%%%%%%%%% EJE 1
% Pi C

filename = 'DATOS_Alumno.xlsx';
sheet = 1;
xlRange = 'A2:C2';

qh1 = xlsread(filename, sheet, xlRange);

filename = 'DATOS_Alumno.xlsx';
sheet = 1;
xlRange = 'A5:C23';

q1 = xlsread(filename, sheet, xlRange);

pifdifq1=((q1(:,3)-q1(:,2))*0.05/2)/(1000*9.81*qh1(:,3)*0.09^3);

% Pi n

pinq1=((q1(:,1)*2*pi/60)*0.09)/(2*(2*9.81*qh1(:,3))^(1/2));

pinpq1=polyfit(pinq1,pifdifq1,1);

d1pinpq1=polyfit(pifdifq1,pinq1,1);

str1=num2str(pinpq1(1,1),5);
%str2=num2str(p1,5);
str2=num2str(pinpq1(1,2),5);
t1='y = ';
t2=str1;
t3=' x + ';
t4=str2;
%ecuacion recta
res=strcat(t1,t2,t3,t4);
set(handles.text7,'String',res);

%coef desague

cof=pinpq1(1,2)*-1/pinpq1(1,1);
set(handles.text10,'String',cof);

%%%%%%%%%%%%%% EJE 2
% Pi C

filename = 'DATOS_matlab.xlsx';
sheet = 2;
xlRange = 'A2:C2';

qh12 = xlsread(filename, sheet, xlRange);

filename = 'DATOS_matlab.xlsx';
sheet = 2;
xlRange = 'A5:C24';

```

```

q12 = xlsread(filename, sheet, xlRange);

pifdifq12=((q12(:,3)-q12(:,2))*0.05/2)/(1000*9.81*qh12(:,3)*0.09^3);

% Pi n

pinq12=((q12(:,1)*2*pi/60)*0.09)/(2*(2*9.81*qh12(:,3))^(1/2));

pinpq12=polyfit(pinq12,pifdifq12,1);

d2pinpq12=polyfit(pifdifq12,pinq12,1);

str12=num2str(pinpq12(1,1),5);
str22=num2str(pinpq12(1,2),5);
t12='y = ';
t22=str12;
t32=' x + ';
t42=str22;
%ecuacion recta
res2=strcat(t12,t22,t32,t42);
set(handles.text13,'String',res2);

%coef desague

cof2=pinpq12(1,2)*-1/pinpq12(1,1);
set(handles.text14,'String',cof2);

axes(handles.axes1),plot(pinq1,pifdifq1,'-
or',polyval(d1pinpq1,pifdifq1),pifdifq1,'-k*'),grid on,set(gca,
'FontSize',
8),xlabel('\Pi_\Omega','FontSize',8),ylabel('\Pi_c','FontSize',8),title('
Velocidad de Embalamiento Adimensional','FontSize',10);hold on;
axes(handles.axes6),plot(pinq12,pifdifq12,'-
og',polyval(d2pinpq12,pifdifq12),pifdifq12,'-k*'),grid on,set(gca,
'FontSize',
8),xlabel('\Pi_\Omega','FontSize',8),ylabel('\Pi_c','FontSize',8),title('
Velocidad de Embalamiento Adimensional','FontSize',10);hold on;

% --- Executes on button press in bor.
function bor_Callback(hObject, eventdata, handles)
% hObject      handle to bor (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
axes(handles.axes1);
cla

axes(handles.axes6);
cla

ini=char(' ');
set(handles.text7,'String',ini);
ini=char(' ');
set(handles.text10,'String',ini);

ini=char(' ');
set(handles.text13,'String',ini);
ini=char(' ');
set(handles.text14,'String',ini);

```

```

%Limpia

% --- Executes during object creation, after setting all properties.
function text7_CreateFcn(hObject, eventdata, handles)
% hObject    handle to text7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% --- Executes on button press in atras.
function atras_Callback(hObject, eventdata, handles)
% hObject    handle to atras (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close all;
run('gcs');

% -----
function Untitled_1_Callback(hObject, eventdata, handles)
% hObject    handle to Untitled_1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
open('ayuda.pdf');

```

6.1.6 Código de la ventana Semejanza.

```

% --- Executes just before sem is made visible.
function sem_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin    command line arguments to sem (see VARARGIN)

% Choose default command line output for sem
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

axes(handles.axes1)
background1 = imread('img_rodete.jpg');
axis off;
imshow(background1);

axes(handles.axes2)
background2 = imread('img_salto.jpg');
axis off;
imshow(background2);
%Coloca una imagen en boton atrás
[a,map]=imread('1.jpg');
[r,c,d]=size(a);
x=ceil(r/38);
y=ceil(c/110);
g=a(1:x:end,1:y:end,:);
g(g==255)=5.5*255;

```

```

set(handles.atras, 'CData',g);

% UIWAIT makes sem wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = sem_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in atras.
function atras_Callback(hObject, eventdata, handles)
% hObject handle to atras (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
close all;
run('gcs');

function edit1_Callback(hObject, eventdata, handles)
% hObject handle to edit1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
% str2double(get(hObject,'String')) returns contents of edit1 as a
double
clc
hp1=str2double(get(handles.edit1,'String'));

if isnan(hp1)
errorlg('El valor debe ser numérico','ERROR')
set(handles.edit1,'String',0);
hp1=0;
end

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

```

```

rpm1=2100;
dm1=0.09;
dp1=0.09;

filename = 'DATOS_Alumno.xlsx';
sheet = 1;
xlRange = 'C2';

hm = xlsread(filename, sheet, xlRange);

filename = 'DATOS_Alumno.xlsx';
sheet = 1;
xlRange = 'B2';

q1 = xlsread(filename, sheet, xlRange);

hp1=str2double(get(handles.edit1,'String'));
np1=(rpm1*((hp1)^(1/2))/(hm^(1/2)));
np11=num2str(np1,4);
set(handles.text6,'String',np11);

hp1=str2double(get(handles.edit1,'String'));
qp1=(q1*((hp1)^(1/2))/(hm^(1/2)));
qp11=num2str(qp1,4);
set(handles.text7,'String',qp11);

fdifq=(3.2)*(0.05/2);
wuq1=fdifq*1200*2*pi/60;

hp1=str2double(get(handles.edit1,'String'));
wup1=(wuq1*((hp1)^(3/2))/(hm^(3/2)));
wup11=num2str(wup1,4);
set(handles.text8,'String',wup11);

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
ini=char(' ');
set(handles.text6,'String',ini);
ini=char(' ');
set(handles.text7,'String',ini);
ini=char(' ');
set(handles.text8,'String',ini);

% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
ini=char(' ');
set(handles.text17,'String',ini);
ini=char(' ');
set(handles.text18,'String',ini);
ini=char(' ');
set(handles.text19,'String',ini);

% --- Executes on button press in pushbutton5.

```

```

function pushbutton5_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton5 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
rpm1=2100;
dm1=0.09;

filename = 'DATOS_Alumno.xlsx';
sheet = 1;
xlRange = 'C2';

hm = xlsread(filename, sheet, xlRange);
filename = 'DATOS_Alumno.xlsx';
sheet = 1;
xlRange = 'B2';

q1 = xlsread(filename, sheet, xlRange);
dp=str2double(get(handles.edit2,'String'));
np2=(rpm1*(dp))/(dm1);
np22=num2str(np2,4);
set(handles.text17,'String',np22);

dp=str2double(get(handles.edit2,'String'));
qp2=(q1*((dp)^2)/dm1^2);
qp22=num2str(qp2,4);
set(handles.text18,'String',qp22);

fdif2=(3.2)*(0.05/2);
wuq2=fdif2*1200*2*pi/60;

dp=str2double(get(handles.edit2,'String'));
wup_1=(wuq2*((dp)^2)/dm1^2);
wup22=num2str(wup_1,4);
set(handles.text19,'String',wup22);

function edit2_Callback(hObject, eventdata, handles)
% hObject      handle to edit2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of edit2 as a
double

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% -----

```

```
function Untitled_1_Callback(hObject, eventdata, handles)
% hObject    handle to Untitled_1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
open('ayuda.pdf');
```

6.1.7 Código de la ventana Inyector.

```
% --- Executes just before inyector is made visible.
function inyector_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to inyector (see VARARGIN)

% Choose default command line output for inyector
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

%Coloca una imagen en boton atrás
[a,map]=imread('1.jpg');
[r,c,d]=size(a);
x=ceil(r/38);
y=ceil(c/110);
g=a(1:x:end,1:y:end,:);
g(g==255)=5.5*255;
set(handles.atras,'CData',g);

%Fondo1
axes(handles.axes1)
background = imread('IMG_1.jpg');
axis off;
imshow(background);

% UIWAIT makes inyector wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = inyector_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```

% leer imagenes del video
vid = VideoReader('explosion.avi');
sz = [vid.Height vid.Width];
mov = read(vid, [1 vid.NumberOfFrames]);

% preparar GUI
p = get(0, 'DefaultFigurePosition');
hFig = figure('Menubar','none', 'Resize','off', ...
    'Position',[p(1:2) sz(2) sz(1)]);

% reproducir video
movv = struct('cdata',squeeze(num2cell(mov,[1 2 3])), 'colormap',[]);
movie(hFig, movv, 999, vid.FrameRate);

% --- Executes on button press in atras.
function atras_Callback(hObject, eventdata, handles)
% hObject      handle to atras (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
close all;
run('menu');

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton3 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% leer imagenes del video
vid = VideoReader('montaje.mp4');
sz = [vid.Height vid.Width];
mov = read(vid, [1 vid.NumberOfFrames]);

% preparar GUI
p = get(0, 'DefaultFigurePosition');
hFig = figure('Menubar','none', 'Resize','off', ...
    'Position',[p(1:2) sz(2) sz(1)]);

% reproducir video
movv = struct('cdata',squeeze(num2cell(mov,[1 2 3])), 'colormap',[]);
movie(hFig, movv, 9999, vid.FrameRate);

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton4 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% leer imagenes del video
vid2 = VideoReader('Inyector.mp4');
sz2 = [vid2.Height vid2.Width];
mov2 = read(vid2, [1 vid2.NumberOfFrames]);
% preparar GUI
p2 = get(0, 'DefaultFigurePosition');
hFig2 = figure('Menubar','none', 'Resize','off', 'Position',[p(1:2) sz(2)
sz(1)]);

% reproducir video

```

```

movv2 = struct('cdata',squeeze(num2cell(mov,[1 2 3])), 'colormap',[]);
movie2(hFig2, movv2, 999, vid.FrameRate);

% -----
function ayuda_Callback(hObject, eventdata, handles)
% hObject    handle to ayuda (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
open('ayuda.pdf');

% --- Executes during object creation, after setting all properties.
function axes1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to axes1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

```

7. FUTURAS LÍNEAS DE TRABAJO

7.1 Compilación de la interfaz gráfica para poder ejecutarla en ordenadores sin MATLAB

Existe la posibilidad de compilar la interfaz gráfica generada en MATLAB creándose así un archivo ejecutable (.exe) que se puede instalar en cualquier ordenador para así no tener que haber instalado previamente MATLAB para poder ejecutarla. Para poder llevar a cabo esta tarea es necesario disponer de dos ordenadores de arquitecturas x32 y x64 bits. Son necesarios los dos porque se necesitan para generar dos ejecutables distintos y así conseguir que la interfaz gráfica compilada funcione en cualquier ordenador dependiendo de su arquitectura. Yo sólo dispongo de arquitectura x64 bits de forma que dejo esta posibilidad como futura línea de trabajo.

7.2 Vídeo tutorial

Para la realización de la práctica los alumnos cuentan con un guión de prácticas en el que se detalla todo lo necesario para llevar a cabo la práctica a realizar con el banco de ensayos. De cualquier forma, el incluir un vídeo en el que se describiera la información del guión de prácticas sobre el propio banco de ensayos poniéndolo en marcha, etcétera... Podría clarificar las posibles dudas del alumno al llegar al laboratorio agilizando así la toma de medidas.

Incluyo varios vídeos pero ninguno es un tutorial. Uno de los vídeos es una simulación numérica transitoria descriptiva de cómo se llena el inyector de agua, otro muestra el montaje de la turbina Pelton como una fuente de producción de energía eléctrica (en el vídeo se ve como alimenta un motor eléctrico, una bombilla y aparece también un multímetro realizando una medida) y un vídeo del explosionado de los distintos elementos del inyector (realizado con un programa de diseño paramétrico en 3D).

7.3 Mejorar la velocidad de ejecución del programa

Una mejora en la velocidad de ejecución del programa podría ser posible. Aunque en general la interfaz es bastante rápida, en la parte del ensayo debido a la gran cantidad de parámetros y gráficas a representar, puede tardar unos segundos en calcularlo todo. Si en lugar de leer los datos del ensayo cada vez que calculamos guardamos la información en variables internas de MATLAB la velocidad de cálculo aumentaría. Pero esto supondría una reprogramación del código de la interfaz.

8. REFERENCIAS

- [1] Mataix, Claudio. Mecánica de fluidos y maquinas hidráulicas. Madrid: Ediciones del castillo S.A, 1986
- [2] W.D, James, R.F, Donald. Dinámica de los fluidos. Méjico: Trillas, 1975
- [3] Agüera, Jose. Mecánica de fluidos incompresibles y turbo máquinas hidráulicas. Madrid: Ciencia 3
- [4] González Marcos. Máquinas de fluidos. País Vasco: OCW, 2009
- [5] Polo, Manuel. Turbomáquinas hidráulicas.
- [6] Crespo, Antonio. Mecánica de fluidos. Madrid: Top Printer Plus, 2006
- [7] D, Anderson. Computational fluid dynamics. Maryland: McGraw Hill, 1995
- [8] Ansys Inc. Ansys workbench user's guide. Canonsburg: SAS IP,2009

- [9] Ansys Inc. Ansys fluent user's guide. Canonsburg: SAS IP,2011
- [10] Ansys Inc. Ansys fluent theory guide. Canonsburg: SAS IP,2011
- [11] Ansys Inc. Ansys fluent meshing user's guide. Canonsburg: SAS IP,2010
- [12] Ansys Inc. Lecture 2 Introduction to the CFD methodology. Canonsburg: SAS IP,2012
- [13] Ansys Inc. Lecture 3 Boundary conditions. Canonsburg: SAS IP,2012
- [14] Ansys Inc. Lecture 4 Post processing. Canonsburg: SAS IP,2012
- [15] Ansys Inc. Lecture 5 Solver settings. Canonsburg: SAS IP,2012
- [16] Ansys Inc. Lecture 7 Turbulence. Canonsburg: SAS IP,2012
- [17] Ansys Inc. Lecture 9 Non-conformal interfaces and moving zones. Canonsburg: SAS IP,2012
- [18] Ansys Inc. Lecture 11 Transient flow modelling. Canonsburg: SAS IP,2012
- [19] Norma UNE 1039:1994. Madrid: AENOR, 1994
- [20] Norma UNE 1032:1982. Madrid: AENOR, 1982
- [21] Vera, Coello et Al. Ingeniería fluidomecánica. Madrid: Paraninfo, 2012
- [22] <http://www.mathworks.es/>