



UCA

Universidad
de Cádiz

Escuela Superior
de Ingeniería

TRABAJO DE FIN DE GRADO

GRADO EN INGENIERÍA AEROESPACIAL

**PREDISEÑO TERMODINÁMICO DE
COMPRESORES AXIALES BAJO LA
HIPÓTESIS DE TRIDIMENSIONALIDAD**

AUTOR: ALEJANDRO SÁNCHEZ TORRES

Cádiz, julio 2023



UCA

Universidad
de Cádiz

Escuela Superior
de Ingeniería

TRABAJO DE FIN DE GRADO

GRADO EN INGENIERÍA AEROESPACIAL

**PREDISEÑO TERMODINÁMICO DE
COMPRESORES AXIALES BAJO LA
HIPÓTESIS DE TRIDIMENSIONALIDAD**

DIRECTOR: JOSÉ ANTONIO VÉLEZ GODIÑO

AUTOR: ALEJANDRO SÁNCHEZ TORRES

Cádiz, julio 2023

Índice general

Índice de figuras	5
Índice de tablas	8
I Introducción	9
0.1. Contexto	10
0.2. Objetivos	10
0.3. Alcance	10
II Estado de la tecnología	12
1. Máquinas Térmicas	13
2. Transformaciones energéticas en turbomáquinas	18
2.1. Ecuación de Euler	18
2.2. Ecuación de la energía	20
3. Análisis de escalonamiento radial y axial	21
3.1. Escalonamiento elemental de compresor centrífugo	21
3.2. Escalonamiento elemental de compresor axial	22
4. Compresores axiales	25
4.1. Hipótesis de bidimensionalidad	25
4.2. Aspectos geométricos	26
4.3. Fuerzas en una cascada	27
4.4. Pérdidas de una cascada	29
4.5. Transformaciones energéticas en compresores axiales	30
4.6. Triángulo de velocidades y parámetros adimensionales	30

4.6.1. Triángulos de velocidades	30
4.6.2. Parámetros adimensionales	31
4.6.3. Optimización geométrica en un compresor axial	32
4.7. Correlación de Lieblein para cascadas de compresor	32
4.8. Pérdidas anulares y secundarias	33
4.9. Regla de Carter	34
5. Leyes torsionales	35
5.0.1. Ley del torbellino libre ($n = -1$)	36
5.0.2. Ley exponencial ($n = 0$)	36
5.0.3. Ley de grado de reacción constante ($n = 1$)	37
III Diseño termodinámico	38
6. Consideraciones e hipótesis	39
6.1. Hipótesis	39
6.2. Especificaciones de diseño	41
7. Diseño bidimensional	42
7.1. Número de escalonamientos	42
7.2. Triángulo de velocidades	42
7.3. Pérdidas energéticas	44
7.3.1. Pérdidas en los perfiles	44
7.3.2. Pérdidas anulares	44
7.3.3. Pérdidas secundarias	45
7.3.4. Pérdidas totales	45
7.4. Trabajo, rendimiento y relación de compresión del escalonamiento	45
7.5. Parámetros termodinámicos y geométricos	46
7.6. Geometría de los álabes	47
7.7. Resolución del resto de escalonamientos	48
8. Diseño tridimensional	51
8.1. Ley de torbellino libre	51
8.2. Ley exponencial	53
8.3. Ley de grado de reacción constante	53

IV	Software utilizado	55
9.	Matlab y App Designer	56
9.1.	Programación del caso bidimensional	56
9.2.	Programación del caso tridimensional	57
9.3.	Uso de <i>App Designer</i>	57
V	Ejemplo de uso	58
10.	Diseño bidimensional	59
10.1.	Especificaciones y cálculo	59
10.2.	Triángulo de velocidades	60
10.3.	Geometría de las cascadas	61
10.4.	Pérdidas energéticas	63
10.5.	Parámetros termodinámicos	64
10.5.1.	Presiones	64
10.5.2.	Temperaturas	65
10.5.3.	Densidades	66
10.5.4.	Prestaciones	66
11.	Diseño tridimensional	68
11.1.	Ley de torbellino libre	69
11.1.1.	Triángulos de velocidades	69
11.1.2.	Pérdidas, trabajo y rendimiento total a total	71
11.1.3.	Variación de ángulos	71
11.1.4.	Variación de coeficientes adimensionales	72
11.2.	Ley de grado de reacción constante	73
11.2.1.	Triángulos de velocidades	73
11.2.2.	Pérdidas, trabajo y rendimiento total a total	74
11.2.3.	Variación de ángulos	75
11.2.4.	Variación de coeficientes adimensionales	76
11.2.5.	Dificultades con la ley de velocidad axial	76
11.3.	Ley Exponencial	78
11.3.1.	Triángulos de velocidades	78
11.3.2.	Pérdidas, trabajo y rendimiento total a total	79

11.3.3. Variación de ángulos	80
11.3.4. Variación de coeficientes adimensionales	81
VI Conclusiones	82
VII Código utilizado	85
11.4. Funciones	86
11.4.1. f_caso_2D	86
11.4.2. f_triangulo_de_velocidades	91
11.4.3. f_geometria_entrada	92
11.4.4. f_perdidas	93
11.4.5. f_trabajo	93
11.4.6. f_torbellino_libre	94
11.4.7. f_exponencial	95
11.4.8. f_reaccion_constante	97
11.5. App Designer	99
Bibliografía	156

Índice de figuras

1.	Imagen de un motor aeronáutico	10
1.1.	Representación de una máquina de vapor. [1]	14
1.2.	Imagen de una turbina de vapor. [2]	14
1.3.	Ciclo seguido por un motor Stirling [3]	14
1.4.	Imagen de un motor aeronáutico [4]	15
1.5.	Motor Wankel. [5]	15
1.6.	Compresor rotativo.[6]	16
1.7.	Esquema de compresor centrífugo. [6]	16
1.8.	Esquema de compresor axial. [6]	16
1.9.	Compresor de parsons. [7]	17
3.1.	Elementos de un compresor centrífugo [8]	21
3.2.	Diagrama entalpía-entropía de un escalonamiento axial [9]	22
3.3.	Diagrama entalpía-entropía para varios escalonamientos axiales [10]	23
4.1.	Compresor de radio medio constante [9]	25
4.2.	Ejemplos de la familia NACA-65[11]	26
4.3.	Geometría de la cascada[9]	27
4.4.	Ángulos de la cascada[9]	27
4.5.	Fuerzas aerodinámicas[9]	28
4.6.	Línea de compresión en una cascada[9]	29
4.7.	velocidades en un escalonamiento de compresor axial[9]	31
4.8.	Imagen de las capas límite mencionadas [9]	33
4.9.	Representación gráfica de la Correlación de Lieblein. [9]	33
6.1.	Variación del calor específico del aire a presión constante con la temperatura	40
6.2.	Distribución del perfil	41

6.3. Imagen del perfil	41
7.1. Diagrama de flujo para la determinación de los ángulos de flujo	43
7.2. Triángulo de velocidades de un compresor axial. [12]	44
7.3. Diagrama de flujo para el cálculo de la desviación en el rotor	48
7.4. Diagrama de flujo del proceso	50
10.1. Pantalla para introducir las especificaciones	60
10.2. Imagen con los resultados referentes al triángulo de velocidades	61
10.3. Geometría de la cascada	62
10.4. Valores numéricos para cada cascada	62
10.5. Representación de cómo se vería el perfil del compresor	63
10.6. Valores referentes a las pérdidas	64
10.7. Variación de la presión y presión total por escalonamiento	65
10.8. Variación de la temperatura y temperatura total por escalonamiento	65
10.9. Variación de la densidad y calor específico por escalonamiento	66
10.10 Variación de la relación de compresión por escalonamiento y valor del trabajo específico	67
11.1. Mensaje de advertencia	68
11.2. Pantalla para elegir la ley torsional	69
11.3. Triángulos de velocidades en el primer escalonamiento	70
11.4. Triángulos del último escalonamiento	70
11.5. Tabla con los valores de dicha pestaña	71
11.6. Variación de los ángulos	72
11.7. Variación de los coeficientes adimensionales con el radio	72
11.8. Triángulos de velocidades en el primer escalonamiento	73
11.9. Triángulos de velocidades en el último escalonamiento	74
11.10 Tabla con los valores de dicha pestaña	74
11.11 Variación de los ángulos	75
11.12 Variación de los coeficientes adimensionales con el radio	76
11.13 Valor del coeficiente de flujo para estos cambios	77
11.14 Mensaje de error	77
11.15 Nuevo valor del coeficiente de flujo	78
11.16 Triángulos de velocidades en el primer escalonamiento	79
11.17 Triángulos de velocidades en el último escalonamiento	79

11.18 Tabla con los valores de dicha pestaña	80
11.19 Variación de los ángulos	80
11.20 Variación de los coeficientes adimensionales con el radio	81

Índice de tablas

1.1. Tabla resumen de las máquinas térmicas anteriores.	16
6.1. Valores de las constantes para el aire. [13]	39
11.1. Valores del rendimiento total a total del primer escalonamiento para cada caso de diseño. . .	83

Parte I

Introducción

0.1. Contexto

Un compresor es una turbomáquina que permite comprimir un gas. En función del tipo de fluido o la geometría que tengan, pueden ser de varios tipos, como se explicará más adelante. Este proyecto se centra en el diseño de compresores axiales, que son muy utilizados en aplicaciones industriales, como en motores aeronáuticos, en la producción de energía en centrales eléctricas, o bien en procesos que requieran refrigeración y acondicionamiento de gases. El diseño de estos compresores axiales no es una tarea sencilla, y es bastante común realizar labores de prediseño mediante el uso de programas informáticos. El uso de estos programas informáticos es importante para optimizar el rendimiento de estas turbomáquinas, y en general, facilitar el análisis del resto de parámetros que involucran al prediseño de estos compresores. Este tipo de prediseños permiten abordar posteriormente el diseño completo del compresor con una mayor claridad, por lo que su utilidad es considerable.

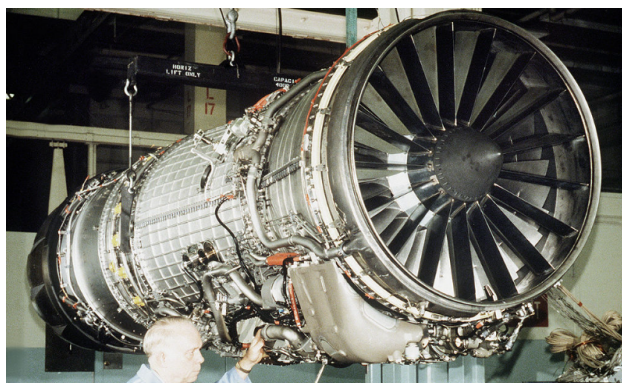


Figura 1: Imagen de un motor aeronáutico

Es importante, por tanto, diferenciar que un prediseño no implica la caracterización completa del compresor (diseño completo), sino que tiene como objetivo la determinación de las principales características del mismo (como por ejemplo, el rendimiento total de la máquina), que serán las que condicionen tanto sus prestaciones como su coste. De esta manera, un fabricante puede estimar ofertas invirtiendo pocos recursos, para más adelante estudiar un diseño completo.

0.2. Objetivos

El objetivo de este proyecto es el desarrollo de una metodología para el prediseño termodinámico de un compresor axial, utilizando tanto hipótesis bidimensionales como hipótesis tridimensionales. A partir de una serie de especificaciones que podrán ser definidas por un usuario, se calcularán los diversos parámetros termodinámicos y geométricos que conforman dicho compresor axial.

Asimismo, se buscará realizar una interfaz gráfica para mostrar dichos resultados. Esta herramienta será lo más intuitiva posible, de forma que pueda ser empleada por usuarios con conocimientos limitados sobre este tipo de turbomáquinas y sus características completas, tratando de facilitar la comparación entre distintos casos de estudio.

0.3. Alcance

La determinación de los resultados se llevará a cabo mediante el uso de un programa de cálculo utilizando el software *Matlab*, y concretamente, su módulo *App Designer*, que servirá para realizar la interfaz gráfica en cuestión. La interfaz tendrá implementadas una serie de funciones que serán las encargadas de realizar los

cálculos. En esta interfaz gráfica, una vez ejecutada, serán introducidas las especificaciones mencionadas en el apartado anterior, que darán como resultado la determinación del prediseño termodinámico y geométrico del compresor axial adecuado, utilizando dos modelos distintos: uno considerando la hipótesis bidimensional, y otro tomando la hipótesis tridimensional. Se establecerá adecuadamente la diferenciación o similitud entre ellas cuando proceda en función de las características del compresor axial especificado.

Asimismo, se excluye en este proyecto la simulación mediante Mecánica de Fluidos Computacional (CFD). Tampoco se va a realizar la caracterización geométrica detallada de dicho compresor, ya que esto implicaría el uso de Software de diseño, el cual queda fuera del alcance de este proyecto.

Parte II

Estado de la tecnología

Capítulo 1

Máquinas Térmicas

Una máquina de fluido es aquella máquina que tiene como objetivo intercambiar energía con un fluido. En función del tipo de fluido que las atraviesa, estas máquinas se pueden clasificar en hidráulicas y térmicas:

- Máquinas Hidráulicas: el fluido que las atraviesa es incompresible, es decir, líquidos como el agua, aunque también podrían trabajar con gases en condiciones de incompresibilidad. Un ejemplo de máquina hidráulica puede ser una bomba de agua.
- Máquinas Térmicas: en ellas, el fluido utilizado es compresible, lo que normalmente implica el uso de un gas.

Una diferencia fundamental entre estos dos tipos de máquinas, es que mientras que ambas utilizan la energía mecánica del fluido en movimiento (la suma de las energías cinética y potencial), las máquinas térmicas, además, utilizan la energía térmica del fluido.

A continuación se desarrollará una clasificación de las máquinas térmicas, con el objetivo de entender bien su funcionamiento y sus variantes, hasta llegar al compresor axial, que será la máquina empleada en este trabajo. Dentro de las máquinas térmicas existen dos clasificaciones fundamentales:

- En función del flujo utilizado, se pueden clasificar las máquinas térmicas en volumétricas o en turbomáquinas:
 - Volumétricas: son aquellas máquinas térmicas en las cuales el fluido es discreto, es decir, no existe un flujo continuo.
 - Turbomáquinas: en las cuales el flujo sí es continuo.
- En función del intercambio de energía producido, existen las máquinas térmicas motoras y las máquinas térmicas generadoras:
 - Motoras: en ellas, la energía del fluido disminuye, la cual es aprovechada por la máquina.
 - Generadoras: aquellas que aportan energía al fluido.

A continuación se mostrarán y desarrollarán ejemplos de cada tipo.

Dentro de las máquinas motoras, existen dos clasificaciones extra que se pueden emplear, en función de si la combustión de los gases empleados para el intercambio energético se producen en el interior o en el exterior de la máquina. La máquina de vapor, invento desencadenante de la primera revolución industrial, es una máquina térmica motora de combustión externa volumétrica.

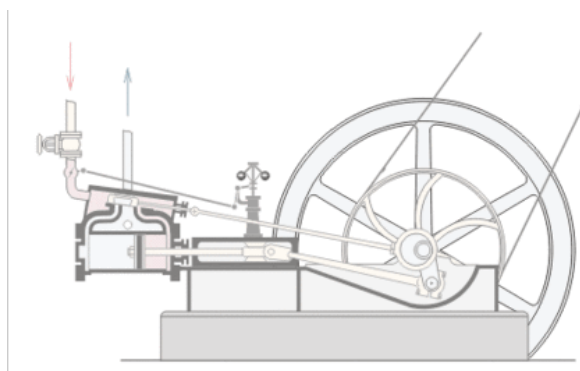


Figura 1.1: Representación de una máquina de vapor. [1]

La turbina de vapor, un desarrollo posterior y que hoy en día es muy utilizada para producir electricidad en centrales nucleares, es del mismo estilo, pero utilizando un flujo continuo, por lo que se habla de turbomáquina en este caso.



Figura 1.2: Imagen de una turbina de vapor. [2]

Existen otras máquinas similares a las dos mencionadas anteriormente, pero que utilizan gases no condensables. Una de ellas es el motor Stirling, que se puede ver en la siguiente imagen y que es de tipo volumétrico.

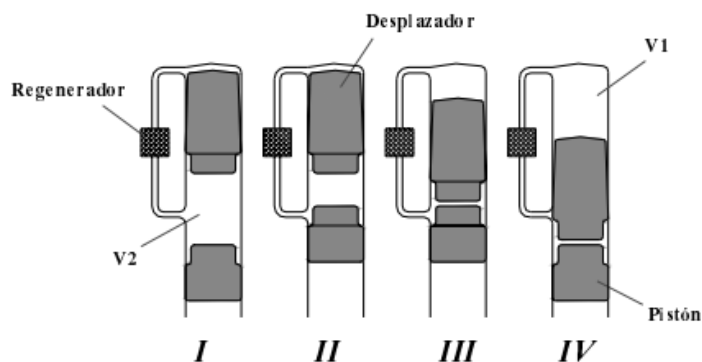


Figura 1.3: Ciclo seguido por un motor Stirling [3]

La otra es la turbina de gas, muy empleada en motores aeronáuticos, que como es lógico, se trata de una

turbomáquina y que puede ser radial o axial en función del sentido del flujo.

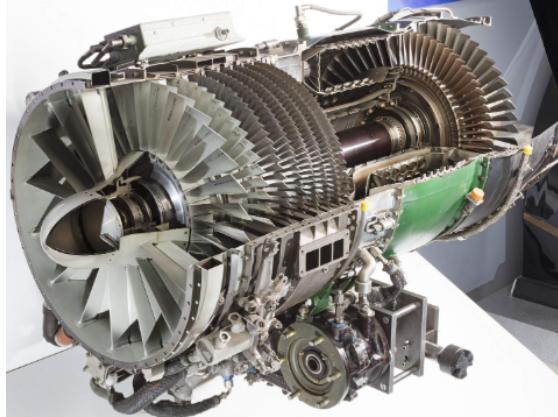


Figura 1.4: Imagen de un motor aeronáutico [4]

En la anterior imagen se observa la turbina axial en la parte posterior del motor. Por otra parte, dentro de las máquinas motoras de combustión de combustión interna, se puede destacar el motor wankel, que es de tipo volumétrico.

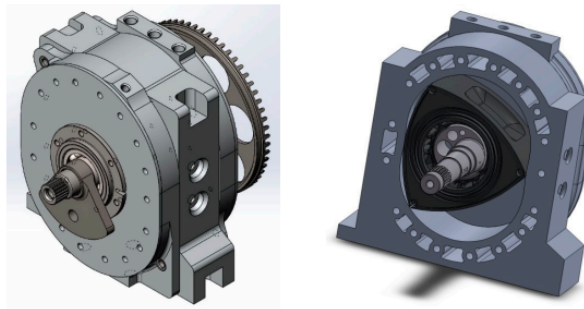


Figura 1.5: Motor Wankel. [5]

Una vez vistos ejemplos de máquinas térmicas motoras, se van a mostrar ejemplos de máquinas térmicas generadoras. El ejemplo más claro de este tipo de turbomáquinas son los compresores, que se pueden dividir en varios tipos.

Por ejemplo, dentro de los compresores volumétricos, se pueden destacar los compresores reciprocantes y los compresores rotativos.

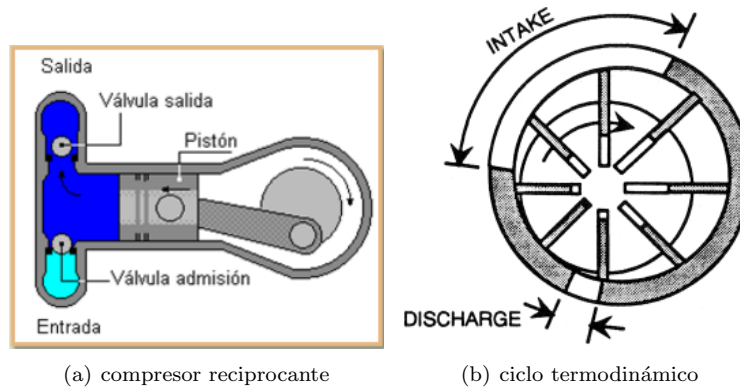


Figura 1.6: Compresor rotativo.[6]

Por otro lado, se tienen los compresores que trabajan con flujo continuo y dentro de ellos, se pueden dividir en centrífugos o axiales, en función del sentido del flujo.

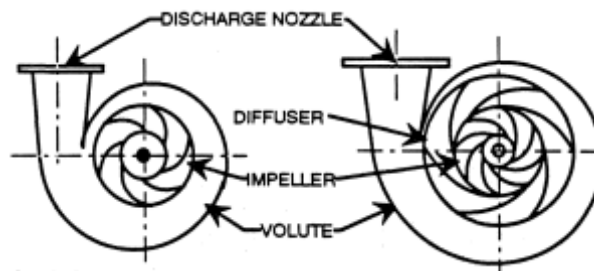


Figura 1.7: Esquema de compresor centrífugo. [6]

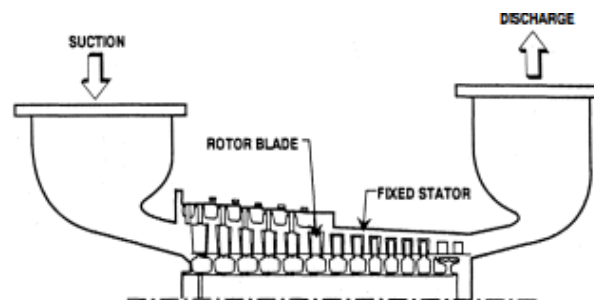


Figura 1.8: Esquema de compresor axial. [6]

Por tanto, el compresor axial de gas, que será la turbomáquina objeto de estudio en este proyecto, queda definida como una máquina térmica generadora, con un flujo de gas no condensable y continuo.

	MOTORAS	GENERADORAS
VOLUMÉTRICAS	Motores Wankel y Stirling	Compresores recíprocos y rotativos
TURBOMÁQUINAS	Turbinas (gas o vapor)	Compresores axiales y centrífugos

Tabla 1.1: Tabla resumen de las máquinas térmicas anteriores.

El concepto de funcionamiento del compresor axial es conocido desde mediados del siglo XIX, cuando Tournaire lo presentó en la Academia Francesa de las Ciencias. [14]

En cuanto al compresor axial, el ingeniero británico Charles Parsons creó un prototipo que contaba con 81 escalonamientos y un rendimiento del 70%. El funcionamiento de esta máquina era el de una turbina axial trabajando a la inversa, y se dejó de fabricar en 1908 por problemas aerodinámicos en los perfiles. Este fue el principal problema del compresor axial durante la primera mitad del siglo XX, y se trataban fundamentalmente de problemas de desprendimiento de la capa límite. No es hasta la segunda mitad del siglo XX, cuando se producen notables mejoras en el diseño de estas turbomáquinas. [9]

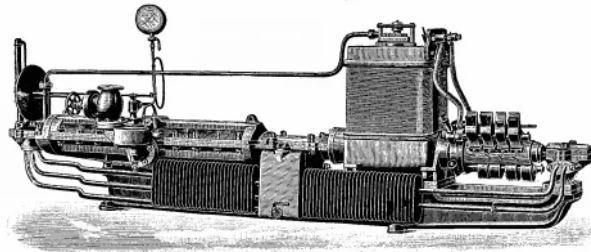


Figura 1.9: Compresor de parsons. [7]

La primera guerra mundial y la necesidad de las grandes potencias por desarrollar la aviación, supusieron un elemento clave en el desarrollo de la aeronáutica y la mecánica de fluidos (teoría del perfil aerodinámico), produciéndose avances muy importantes en estos campos que serían claves para la mejora de los compresores en el futuro. Más adelante, en vísperas de la segunda guerra mundial, se produce otro impulso tecnológico muy importante, y los compresores axiales entran en escena convirtiéndose en elementos fundamentales de los motores aeronáuticos, ya que su uso pasa a ser fundamental para comprimir el aire y aumentar la eficiencia de la combustión. Durante estos años, la eficiencia de los compresores se aumentó reduciendo el espacio entre álabes y el aumento del espesor de los mismos. El año 1958 se puede considerar el inicio del compresor axial en la normalidad de la industria, ocurriendo este año la desclasificación de los informes NACA, en los cuales están fuertemente basados todos los diseños de compresores actuales. [14]

En la actualidad, el compresor axial está muy asentado en la industria aeroespacial por los motivos explicados anteriormente. Entre las posibles mejoras que se buscan de cara al futuro, se encuentran:

- Incremento en la relación de presiones.
- Mejora en los sistemas que eviten sobretensiones. [15]
- Ampliar el rango operacional.
- Reducir fugas de flujo en la punta del compresor.
- Soluciones numéricas más exactas para las ecuaciones de Navier-Stokes.
- Desarrollo de perfiles supersónicos para mayor relación de presiones.
- Enfriamiento del compresor mediante la inyección de agua por etapas. [11]

Capítulo 2

Transformaciones energéticas en turbomáquinas

Una vez que se ha hecho la clasificación de las distintas máquinas térmicas, es el momento de comprender los fundamentos físicos más básicos de las turbomáquinas. Como se ha explicado anteriormente, las turbomáquinas son aquellas máquinas térmicas con un flujo continuo. Normalmente, se diferencian dos elementos distintos en una turbomáquina; el rotor y el estator:

- Rotor: en él se produce transformación energética y trabajo.
- Estator: también se produce transformación de energía, pero no trabajo.

2.1. Ecuación de Euler

La ecuación de Euler es el principio físico fundamental en el que se basan las turbomáquinas, y permite evaluar el trabajo específico intercambiado.

Se tiene un fluido que atraviesa un tubo de corriente T, que gira con velocidad angular Ω alrededor de un eje Z. Utilizando un sistema de coordenadas cilíndricas, se tiene la velocidad tangencial U, la velocidad radial R, y la velocidad axial A. Si se define un sistema de referencia solidario al giro, se tiene la siguiente ecuación:

$$\vec{C} = \vec{W} + \vec{U} \quad (2.1)$$

donde W es la velocidad relativa.

A continuación se aplica el teorema del momento cinético:

$$\frac{d}{dt}(\vec{r} \times m\vec{c}) = \vec{M} \quad (2.2)$$

que nos indica que el momento resultante en el volumen de control es igual a la derivada respecto al tiempo de la cantidad de movimiento de la masa del interior de dicho volumen.

Aplicando esta ecuación al tubo de corriente definido (T), se tienen dos contribuciones, una generada por el flujo de momento cinético a través de la superficie que delimita el volumen de control, y la variación respecto al tiempo de dicho momento cinético, que será nula por considerar la hipótesis de estado estacionario.

$$\frac{d}{dt}(\vec{r} \times m\vec{c}) = \int_S (\vec{r} \times \vec{c}) \cdot (\rho \cdot \vec{c} \cdot d\vec{s}) + \frac{d}{dt} \int (\vec{r} \times \vec{c}) \cdot (\rho \cdot d\vec{v}) \quad (2.3)$$

Por otro lado, la única componente de la velocidad que genera momento respecto al eje Z es la velocidad tangencial U. Llamando a las superficies de entrada y salida de flujo como S_1 y S_2 respectivamente, la ecuación que se tiene es la siguiente:

$$M_z = \int_{S_1} (\vec{r} \times \vec{c}_u)_1 \cdot (\rho \cdot \vec{c} \cdot d\vec{s})_1 + \int_{S_2} (\vec{r} \times \vec{c}_u)_2 \cdot (\rho \cdot \vec{c} \cdot d\vec{s})_2 \quad (2.4)$$

Tomando la ecuación de forma escalar, y considerando condiciones homogéneas en cada superficie, se tiene:

$$M_z = -c_{u1} \cdot \rho_1 \cdot c_1 \int_{S_1} (r \cdot \cos\gamma \cdot ds)_1 + c_{u2} \cdot \rho_2 \cdot c_2 \int_{S_2} (r \cdot \cos\gamma \cdot ds)_2 \quad (2.5)$$

donde γ es el ángulo formado por las velocidades absolutas con las normales a las superficies de entrada y salida.

Aunque el radio de la sección va a ser variable, es correcto aproximar este valor por el de un *radio medio*, por lo que puede ser extraído de la integral, como r_1 y r_2 respectivamente.

La ecuación de la continuidad de la masa indica que:

$$\dot{m} = \dot{m}_1 = \dot{m}_2 = \rho_1 \cdot c_1 \int_{S_1} (\cos\gamma \cdot ds)_1 = \rho_2 \cdot c_2 \int_{S_2} (\cos\gamma \cdot ds)_2 \quad (2.6)$$

Por lo que se pueden anular estos términos, lo que deja la siguiente ecuación, teniendo en cuenta que el par motor tiene sentido contrario al momento (par) sobre el fluido:

$$M_{mz} = -M_z = \dot{m} \cdot (r_1 \cdot c_{u1} - r_2 \cdot c_{u2}) \quad (2.7)$$

Tomando en cuenta la velocidad angular Ω y sabiendo que $u = \Omega \cdot r$, es posible calcular la potencia desarrollada:

$$\dot{W} = \Omega \cdot M_{mz} = \dot{m} \cdot (u_1 \cdot c_{u1} - u_2 \cdot c_{u2}) \quad (2.8)$$

Y finalmente, tomando valores específicos, se llega a la **Ecuación de Euler para turbomáquinas** [8]:

$$\hat{w} = (u_1 \cdot c_{u1} - u_2 \cdot c_{u2}) \quad (2.9)$$

Es decir, los cambios en la dirección del flujo originan una diferencia de presiones entre las dos caras de un álabe, y esta diferencia de presiones, a su vez, origina una fuerza.

Resulta conveniente cambiar el signo a la ecuación de Euler cuando se trabaja con turbomáquinas generadores (como es el caso del compresor).

$$\hat{w} = (u_2 \cdot c_{u2} - u_1 \cdot c_{u1}) \quad (2.10)$$

2.2. Ecuación de la energía

Partiendo de la ecuación 2.1, se cumple que:

$$w_1^2 = c_1^2 + u_1^2 - 2u_1 \cdot c_{u1} \quad (2.11)$$

$$w_2^2 = c_2^2 + u_2^2 - 2u_2 \cdot c_{u2} \quad (2.12)$$

despejando los valores de c_{u1} y c_{u2} y sustituyendo en la ecuación de Euler 2.9, se tiene:

$$\hat{w} = \left(\frac{c_1^2}{2} - \frac{c_2^2}{2} \right) + \left(\frac{w_2^2}{2} - \frac{w_1^2}{2} + \frac{u_1^2}{2} - \frac{u_2^2}{2} \right) \quad (2.13)$$

De acuerdo con el primer principio de la termodinámica aplicado a sistemas adiabáticos (que no se va a desarrollar aquí), se tiene:

$$\hat{w} = \left(\frac{c_1^2}{2} - \frac{c_2^2}{2} \right) + (h_1 - h_2) = (h_{01} - h_{02}) \quad (2.14)$$

donde h_0 es la entalpía de remanso o estática (para la velocidad absoluta).

Si se igualan las dos ecuaciones anteriores, entonces se obtiene la **Ecuación de la Energía** [8]:

$$h_{01R} + \frac{u_2^2 - u_1^2}{2} = h_{02R} \quad (2.15)$$

donde en esta ocasión h_{01R} es referida respecto a la velocidad relativa.

Es decir, la variación de entalpía se produce debido a la expansión del fluido y las fuerzas centrífugas, cuyo trabajo evitan la conservación de la entalpía total relativa.

Capítulo 3

Análisis de escalonamiento radial y axial

Como se ha expuesto anteriormente, una turbomáquina se compone de partes móviles (rótora) e inmóviles (estática). Al conjunto de un rotor y un estator se le denomina escalonamiento. En este apartado se van a tratar de comprender los escalonamientos, tanto de compresores axiales como de compresores radiales.

3.1. Escalonamiento elemental de compresor centrífugo

Los compresores centrífugos se caracterizan por un flujo circular. Tienen un rango de aplicación más amplio que los compresores axiales y pueden alcanzar mayores relaciones de presiones, pero trabajan con menor gasto másico y el número de escalonamientos suele estar limitado.

Al igual que en los compresores axiales, el escalonamiento de un compresor centrífugo se compone de dos partes, además de la voluta:

- Rodete: pueden ser de tipo abierto o cerrado, y es la parte encargada de convertir la potencia mecánica en velocidad absoluta y presión.
- Difusor: que pueden o no presentar álabes, y que convierte la energía cinética del flujo en presión.
- Voluta: se encarga de guiar el flujo a la salida.

El salto de entalpía en estos compresores se debe a la difusión del flujo relativo y al trabajo realizado por las fuerzas centrífugas.

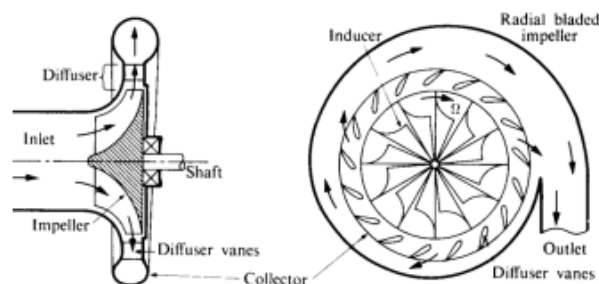


Figura 3.1: Elementos de un compresor centrífugo [8]

3.2. Escalonamiento elemental de compresor axial

El objetivo principal es definir rendimientos y pérdidas, en primer lugar de un compresor elemental, y posteriormente de un conjunto de varios escalonamientos. Es importante notar que se considera el escalonamiento como adiabático. Por otro lado, la principal hipótesis que simplifica las ecuaciones de un compresor axial a diferencia del centrífugo, es que la velocidad axial permanece constante.

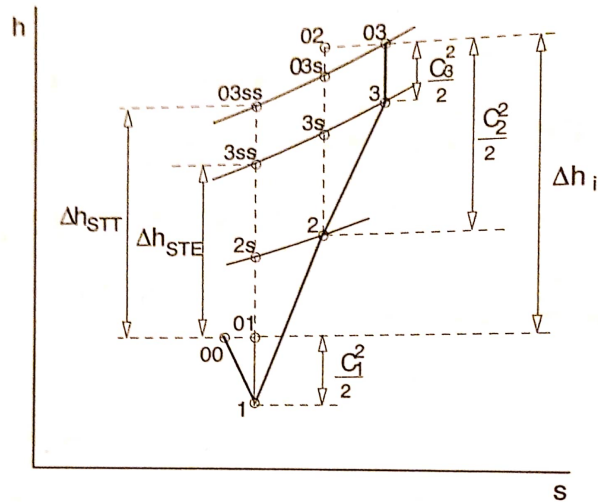


Figura 3.2: Diagrama entalpía-entropía de un escalonamiento axial [9]

En primer lugar, es importante comprender la nomenclatura; el punto 1 es la entrada al compresor (entrada del rotor), el punto 2 la zona intermedia entre rotor y estator, y el punto 3 la salida del estator. El subíndice s indica procesos isentrópicos de referencia.

El trabajo específico se obtiene como:

$$\hat{w} = h_{03} - h_{01} \quad (3.1)$$

Otro parámetro clave es el rendimiento total a total:

$$\eta_{TT} = \frac{h_{3ss} - h_{01}}{h_{03} - h_{01}} \quad (3.2)$$

Las pérdidas que se producen en el escalonamiento son el fruto de una suma de pérdidas por fricción en los perfiles, paredes, pérdidas a la salida, u otras pérdidas secundarias, pérdidas intersticiales, y pérdidas por rozamiento de disco. También existen pérdidas mecánicas o pérdidas por calor. Se pueden calcular de la siguiente manera:

$$Y_{ESC} = Y_R + Y_E = h_{03} - h_{03s} \quad (3.3)$$

Ya se han visto los fundamentos termodinámicos para un escalonamiento, pero como se ha dicho, un compresor axial se va a componer siempre de varios de estos conjuntos, por lo que conviene estudiarlos de manera conjunta.

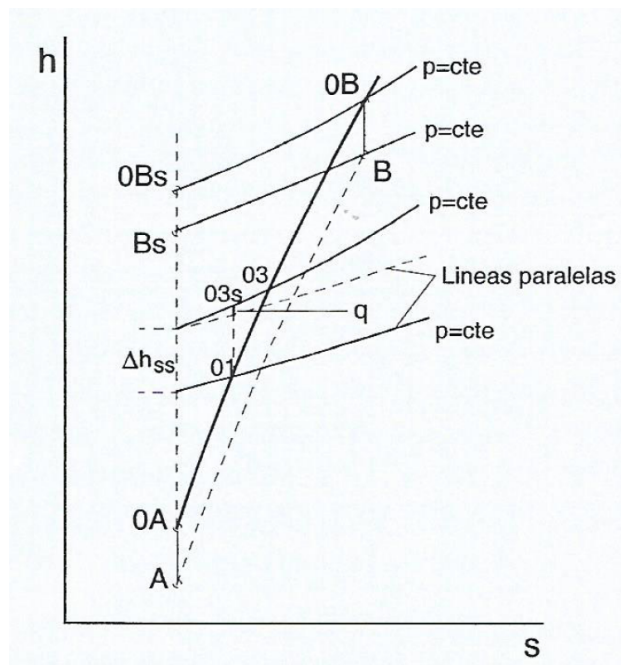


Figura 3.3: Diagrama entalpía-entropía para varios escalonamientos axiales [10]

Lo que ilustra la imagen anterior, es que el salto de entalpía de la máquina es la suma de los saltos de los escalonamientos. Esto se resume en que el trabajo específico queda como:

$$\hat{w} = h_{0B} - h_{0A} = \sum (h_{03} - h_{01}) \quad (3.4)$$

Por otro lado, el rendimiento total a total se expresa como:

$$\eta_{TT} = \frac{h_{0Bs} - h_{0A}}{h_{0B} - h_{0A}} \quad (3.5)$$

Y finalmente las pérdidas:

$$Y_{ESC} = Y_R + Y_E = h_{0B} - h_{0Bs} \quad (3.6)$$

Una vez definidos estos parámetros tanto para escalonamiento individual como para máquina, conviene preguntarse las similitudes y diferencias que pueden observarse.

Es posible demostrar que la relación entre el rendimiento de un escalonamiento y el de la máquina viene dado por la siguiente relación:

$$\eta_i = \eta_i' \cdot \frac{1}{Z} \quad (3.7)$$

donde el valor prima es el rendimiento de un escalonamiento individual, y Z un parámetro denominado *coeficiente de contrarrecuperación*, que es mayor que uno. Esto indica que el rendimiento de escalonamiento individual es mayor que el de la máquina. Este coeficiente de contrarrecuperación depende del rendimiento individual de cada escalonamiento, de la relación de presiones de la máquina, y del número de escalonamientos de la misma. Las tendencias se muestran a continuación:

- Existe proporcionalidad entre el rendimiento de la máquina y el del escalonamiento.
- Z es inversamente proporcional al rendimiento del escalonamiento.
- Z es proporcional al número de escalonamientos.
- Z es proporcional a la relación de presiones de la máquina.

Capítulo 4

Compresores axiales

Una vez que se han estudiado las transformaciones energéticas y un escalonamiento elemental, es el momento de centrar toda la atención en el compresor axial, caracterizándolo completamente.

Como ya se ha explicado anteriormente, un compresor axial se compone de varios escalonamientos formados por un rotor y un estátor. El rotor es la parte encargada de aumentar la energía cinética absoluta mientras reduce la velocidad relativa, para así incrementar la presión, mientras que el estátor convierte la energía cinética en presión.

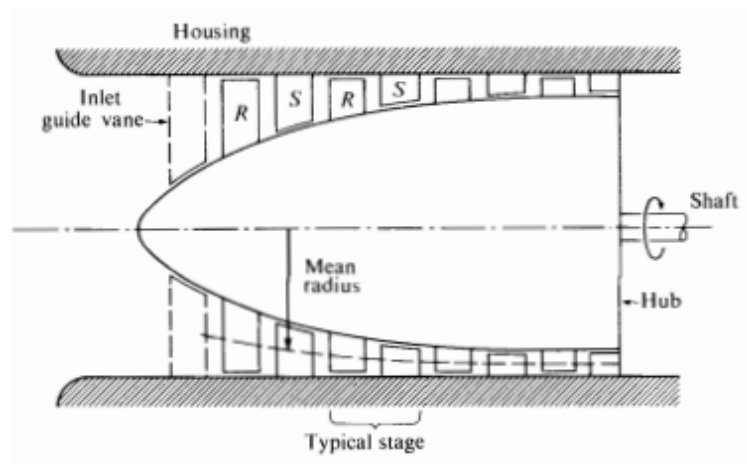


Figura 4.1: Compresor de radio medio constante [9]

Como se puede observar en la figura anterior, la sección media se va estrechando, y además el compresor axial puede contar con Inlet Guide Vanes para facilitar la entrada del flujo. Típicamente se considera la velocidad tangencial como constante durante todo el camino seguido por el fluido.

4.1. Hipótesis de bidimensionalidad

La hipótesis de bidimensionalidad es la asunción de que el flujo no tiene componentes radiales de velocidad. Esta hipótesis simplifica sobremedida el análisis, y se cumple cuando la altura del álabe es mucho menor que el diámetro medio de la corona $\frac{H}{D_m} \ll 1$. Esto permite estudiar el compresor axial mediante la definición de cascada plana de álaves. Además, las características del flujo se toman para el valor del radio medio del álabe, sin importar la altura del mismo. [9]

4.2. Aspectos geométricos

El perfil del álabe queda definido por una serie de parámetros:

- Línea media: es la línea que divide el espesor del perfil de forma igualitaria.
- Cuerda del perfil: línea que une los bordes de ataque y salida.
- Distribución de espesores: relación entre el espesor del perfil y la línea media.

Entre los perfiles más utilizados en compresores axiales, destacan los siguientes:[9]

- NACA-65: Desarrollados por el Comité Asesor Nacional para la Aeronáutica, hoy NASA, en Estados Unidos. La línea media es un arco de circunferencia y su espesor máximo es de un 40 % de la cuerda. Se usan para flujo subsónico.
- C-4 (GB): Basados en arcos de circunferencia, aunque también se han usado arcos de parábola, tienen un espesor máximo del 30 % de la cuerda. Se usan en flujo subsónico.
- DCA: Doble arco de circunferencia, con un espesor del 50 %. Se trata de álabes supercríticos, ya que están adaptados para que la corriente pueda ser supersónica a su paso.
- Supersónicos: Con curvaturas muy pequeñas, del orden del 2 % de la cuerda.
- PVD (Distribución de Velocidad Predefinida): Son un tipo especial de álabe que se fabrican de forma específica tras analizar la distribución de velocidades.

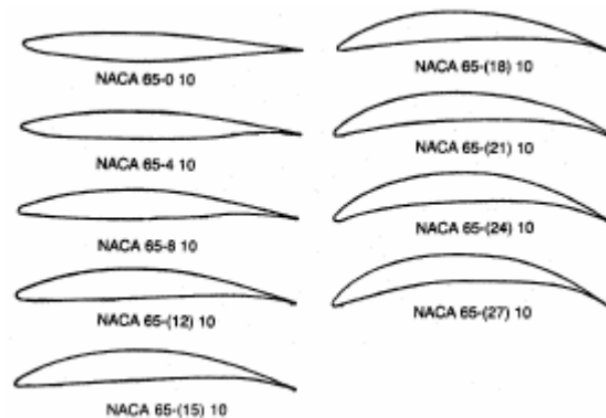


Figura 4.2: Ejemplos de la familia NACA-65[11]

Los NACA y C-4 se recomiendan para Mach a la entrada menores a 0.78, mientras que los DCA son óptimos para flujo transónico.

Una vez definida la geometría del perfil, se definen los parámetros geométricos de una cascada. Esta tiene la siguiente configuración:

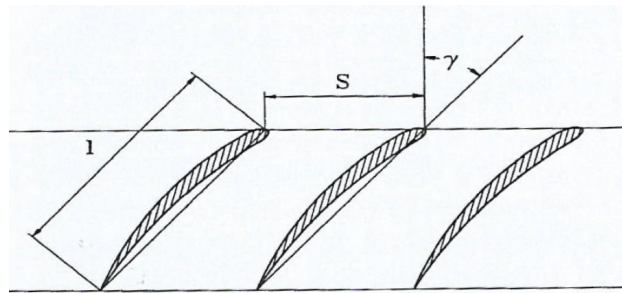


Figura 4.3: Geometría de la cascada[9]

Queda por tanto definida por el paso S (o distancia entre perfiles), y el ángulo de calado γ , que forma la cuerda con la dirección perpendicular al frente de la cascada.

Una cascada también está definida por una serie de ángulos dependientes del flujo.

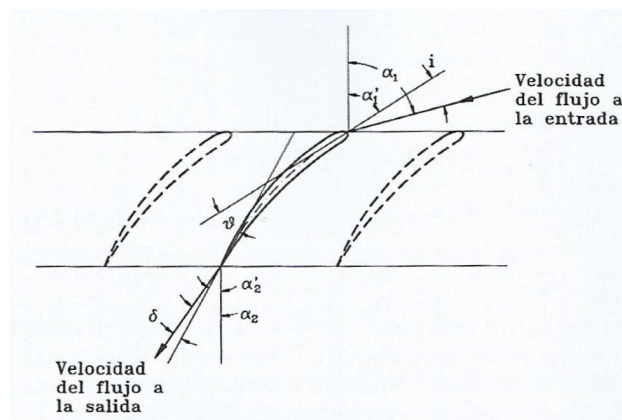


Figura 4.4: Ángulos de la cascada[9]

Los ángulos α_1 y α_2 son los que forman las velocidades de entrada y salida respecto a la perpendicular al frente, mientras que α_1' y α_2' son los que forman la tangente a la línea media de la entrada y salida con la perpendicular. A partir de aquí, se definen los siguientes ángulos:

- Ángulo de incidencia: $i = \alpha_1 - \alpha_1'$
- Ángulo de desviación: $\delta = \alpha_2 - \alpha_2'$
- Ángulo de deflexión: $\epsilon = \alpha_1 - \alpha_2$

Se puede demostrar que la relación existente entre los tres ángulos es:

$$\epsilon = i + \theta - \delta \quad (4.1)$$

donde $\theta = \alpha_1' - \alpha_2'$

4.3. Fuerzas en una cascada

La interacción del flujo con el álabe genera una serie de fuerzas aerodinámicas que son importantes de cuantificar, y eso es lo que se va a realizar en este apartado.

Si se define un volumen de control alrededor del perfil y se aplica conservación de la masa, se tiene:

$$H_1 \cdot s \cdot c_{a1} \cdot \rho_1 = H_2 \cdot s \cdot c_{a2} \cdot \rho_2 \quad (4.2)$$

Típicamente se considera que la velocidad axial es constante, por lo que $H_1 = H_2$.

Si se aplica conservación del momento cinético en las direcciones axial y tangencial:

- Dirección axial:

$$\int_S p dS_{tat} + p_1 s H_1 - p_2 s H_2 + A_H = \dot{m}(c_{a2} - c_{a1}) \quad (4.3)$$

donde A_H es la fuerza ejercida por el perfil sobre el fluido. Simplificando queda que la fuerza por unidad de altura de álabe es:

$$A = (p_2 - p_1) \cdot s \quad (4.4)$$

- Dirección tangencial:

$$\int_S p dS_{tat} + T = \rho \cdot s \cdot c_a (c_{u1} - c_{u2}) \quad (4.5)$$

en este caso T es la fuerza tangencial ejercida por el perfil sobre el fluido por unidad de álabe. Teniendo en cuenta que $c_{u1} = c_a \operatorname{tg} \alpha_1$ y $c_{u2} = c_a \operatorname{tg} \alpha_2$;

$$T = \rho \cdot s \cdot c_a^2 (\operatorname{tg} \alpha_1 - \operatorname{tg} \alpha_2) \quad (4.6)$$

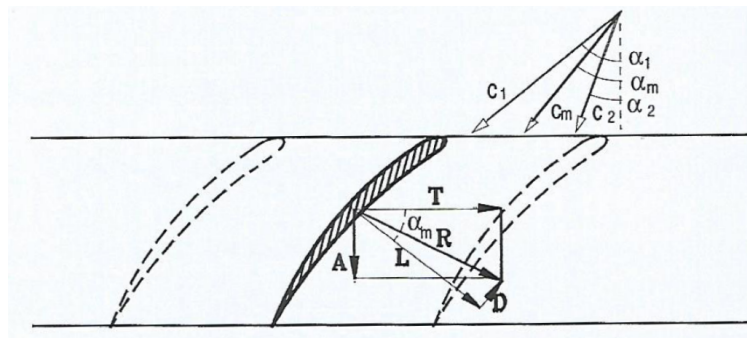


Figura 4.5: Fuerzas aerodinámicas[9]

Como ya se conocen las fuerzas sobre el perfil, es posible darle un enfoque aerodinámico al problema y calcular las fuerzas aerodinámicas. Se define un parámetro denominado *velocidad mediana* c_m .

$$L = \frac{1}{2} C_L \rho l c_m^2 = T \cos \alpha_m + A \sin \alpha_m \quad (4.7)$$

$$D = \frac{1}{2} C_D \rho l c_m^2 = T \sin \alpha_m - A \cos \alpha_m \quad (4.8)$$

Es importante hacer notar que estas ecuaciones, aplicadas en una cascada de rotor, tienen la misma definición, pero habría que utilizar la velocidad relativa en lugar de la absoluta.

4.4. Pérdidas de una cascada

A continuación se van a estudiar las pérdidas que se producen en una cascada y a evaluar el rendimiento de la misma. La evolución seguida es la que se muestra en la imagen posterior.

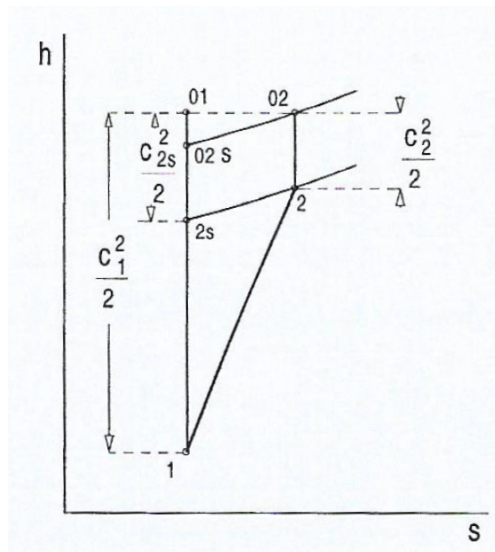


Figura 4.6: Línea de compresión en una cascada[9]

Si se aplica ecuación de la energía a la cascada al proceso real y al proceso isentrópico, se tiene:

$$h_1 + \frac{c_1^2}{2} = h_2 + \frac{c_2^2}{2} \quad (4.9)$$

$$h_1 + \frac{c_1^2}{2} = h_{2s} + \frac{c_{2s}^2}{2} \quad (4.10)$$

respectivamente. Combinando ambas ecuaciones queda:

$$h_2 - h_{2s} = \frac{c_{2s}^2}{2} - \frac{c_2^2}{2} \quad (4.11)$$

Si se hace lo mismo con la ecuación de Bernoulli queda:

$$\frac{p_1}{\rho} + \frac{c_1^2}{2} = \frac{p_2}{\rho} + \frac{c_2^2}{2} + \frac{\Delta p_0}{\rho} \quad (4.12)$$

$$\frac{p_1}{\rho} + \frac{c_1^2}{2} = \frac{p_2}{\rho} + \frac{c_{2s}^2}{2} \quad (4.13)$$

Combinándolas:

$$\frac{\Delta p_0}{\rho} = \frac{c_{2s}^2}{2} - \frac{c_2^2}{2} \quad (4.14)$$

Si ahora se combinan las ecuaciones 4.11 y 4.14, se tiene:

$$h_2 - h_{2s} = \frac{\Delta p_0}{\rho \frac{c_1^2}{2}} \quad (4.15)$$

A partir de aquí, se pueden definir algunos coeficientes. El primero es el coeficiente adimensional de pérdidas:

$$\zeta = \frac{h_2 - h_{2s}}{\frac{c_1^2}{2}} = \frac{\Delta p_0}{\rho \frac{c_1^2}{2}} \quad (4.16)$$

Y otro coeficiente muy utilizado es el coeficiente de aumento de la velocidad.

$$\Phi = \frac{c_{1s}}{c_1} \quad (4.17)$$

4.5. Transformaciones energéticas en compresores axiales

Si se aplica lo visto en el capítulo 1, en el cual se concluía que $\hat{w} = h_{02} - h_{01}$ lo que para un observador fijo implica que en el rotor se cumple que:

$$h_2 - h_1 = \hat{w} + \frac{c_1^2}{2} - \frac{c_2^2}{2} \quad (4.18)$$

para un observador relativo, simplemente se tiene:

$$h_2 - h_1 = \frac{w_1^2}{2} - \frac{w_2^2}{2} \quad (4.19)$$

Es decir, el aumento de presión se debe a la difusión del flujo relativo.

Para el estátor, en cambio, el aumento de presión se debe a la difusión del flujo absoluto, ya que $h_{02} = h_{03}$

$$h_3 - h_2 = \frac{c_2^2}{2} - \frac{c_3^2}{2} \quad (4.20)$$

Finalmente, en el rotor también se cumplen las ecuaciones de Euler y de la energía, 2.9 y 4.11, teniendo en cuenta que el término de las velocidades tangenciales se anulan al ser consideradas constantes. [8]

4.6. Triángulo de velocidades y parámetros adimensionales

4.6.1. Triángulos de velocidades

En un escalonamiento de compresor axial, se tienen las siguientes componentes de velocidades:

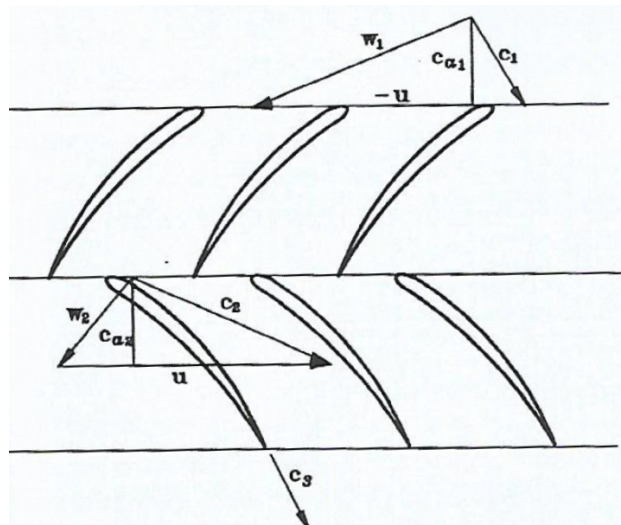


Figura 4.7: velocidades en un escalonamiento de compresor axial[9]

Estos dos triángulos se pueden superponer formando uno único, que es de gran utilidad en la resolución de problemas de compresores axiales.

4.6.2. Parámetros adimensionales

Los parámetros adimensionales que se van a definir a continuación son de uso extendido y útil en compresores axiales. [9]

- Coeficiente de flujo: cociente entre la velocidad axial (es conveniente recordar que se suele suponer como constante a ambos lados del rotor) y la velocidad tangencial.

$$\phi = \frac{C_a}{u} \quad (4.21)$$

- Coeficiente de carga: Trabajo específico del escalonamiento adimensionalizado con el cuadrado de la velocidad periférica.

$$\psi = \frac{w_i}{u^2} \quad (4.22)$$

- Grado de reacción: es un parámetro adimensional que indica la relación entre el salto entálpico en el rotor y el trabajo específico en el escalonamiento.

$$R = \frac{h_2 - h_1}{h_{02} - h_{01}} \quad (4.23)$$

Existen una serie de ecuaciones que relacionan estos parámetros adimensionales con los ángulos de flujo, que se demuestran a continuación, suponiendo la hipótesis de **velocidad axial constante**:

Partiendo de que el grado de reacción se puede expresar como :

$$R = \frac{w_{u1} + w_{u2}}{2 \cdot u} \quad (4.24)$$

o lo que es lo mismo:

$$R = \frac{(c_a \cdot \tan\beta_1) + (c_a \cdot \tan\beta_2)}{2 \cdot u} = \frac{\phi}{2} \cdot (\tan\beta_1 + \tan\beta_2) \quad (4.25)$$

Teniendo en cuenta la definición de coeficiente de flujo.

Mediante modificaciones trigonométricas, estas ecuaciones se pueden expresar también como:

$$R = \frac{(u - c_a \cdot \tan\alpha_1) + (u - c_a \cdot \alpha_2)}{2 \cdot u} = 1 - \frac{\phi}{2} \cdot (\tan\alpha_1 + \tan\alpha_2) \quad (4.26)$$

Y de igual manera:

$$R = \frac{(u - c_a \cdot \tan\alpha_1) + (c_a \cdot \beta_2)}{2 \cdot u} = \frac{1}{2} + \frac{\phi}{2} \cdot (\tan\alpha_1 + \tan\alpha_2) \quad (4.27)$$

4.6.3. Optimización geométrica en un compresor axial

La elección de la geometría del compresor axial vendrá determinada por un serie de factores, como las prestaciones que deba satisfacer, su coste, o el tamaño. Dentro de las prestaciones, se suelen considerar tres: el rendimiento, el trabajo específico, y que el funcionamiento sea estable. A continuación se hace una descripción general de todas ellas. El rendimiento normalmente va a depender de los parámetros adimensionales vistos anteriormente. Típicamente, tanto el grado de reacción como el coeficiente de flujo óptimos son 0.5. La solidez tiene un valor típico entre 1.2 y 1.4, y el coeficiente de carga un valor de en torno a 0.4. Por otro lado es conveniente trabajar con escalonamientos de repetición. [9]

4.7. Correlación de Lieblein para cascadas de compresor

Generalmente, las pérdidas en un compresor axial se producen por el engrosamiento de la capa límite, especialmente en la cara de succión. Este engrosamiento de la capa límite está directamente relacionado con la difusión del flujo. [9]

Se define la relación de difusión como:

$$DR = \frac{\cos\alpha_2}{\cos\alpha_1} \left(1,12 + k(i - i_{ref})^{1,43} + \frac{0,61}{\sigma} \cos^2\alpha_1 (\tan\alpha_1 - \tan\alpha_2) \right) \quad (4.28)$$

Su valor suele estar entre 1.9 y 2.

Lieblein (investigador de la NACA) determinó que se cumple la siguiente relación:

$$\frac{\theta_2}{l} = \frac{0,004}{1 - 1,17 \ln DR} \quad (4.29)$$

donde θ_2 se define como la pérdida de cantidad de movimiento en el borde de salida, expresada como:

$$\theta_2 = \int_{\delta_s}^{\delta_p} \frac{v}{c} \left(1 - \frac{v}{c} \right) dy \quad (4.30)$$

donde v es la velocidad de la estela, c la velocidad central del flujo, y δ_s y δ_p los valores de espesor de las capas límites de succión y presión, respectivamente.



Figura 4.8: Imagen de las capas límite mencionadas [9]

Lieblein también determinó la relación de este coeficiente θ_2 con el coeficiente de pérdidas ζ :

$$\zeta_p = 2 \left(\frac{\theta_2}{l} \sigma \frac{\cos^2 \alpha_1}{\cos^2 \alpha_2} \right) \quad (4.31)$$

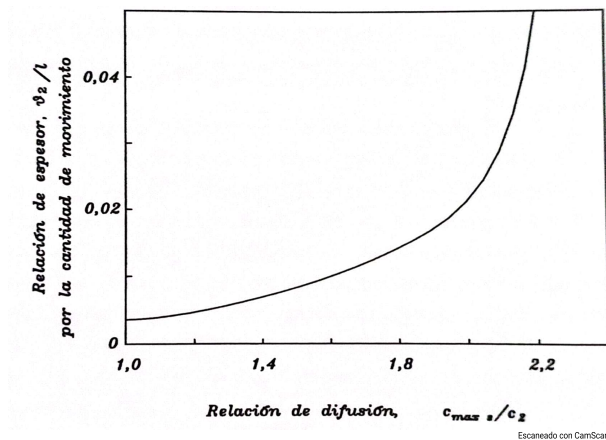


Figura 4.9: Representación gráfica de la Correlación de Lieblein. [9]

En esta correlación, se ve cómo la relación de difusión equivalente no debe superar el valor de 1.95, aproximadamente.

4.8. Pérdidas anulares y secundarias

Además de las pérdidas mencionadas anteriormente, también son contribuidoras importantes las pérdidas anulares y las pérdidas secundarias. Se calculan de la siguiente manera: [9]

$$\zeta_a = 0,02 \frac{S}{H} \sigma \frac{\cos^2 \alpha_1}{\cos^3 \alpha_m} \quad (4.32)$$

$$\zeta_s = 0,018C_L^2\sigma \frac{\cos^2\alpha_1}{\cos^3\alpha_m} \quad (4.33)$$

La suma total de pérdidas serán estas dos más las pérdidas del perfil.

$$\zeta = \zeta_p + \zeta_a + \zeta_s \quad (4.34)$$

4.9. Regla de Carter

La regla de Carter es una ecuación que permite determinar el ángulo de desviación: [8]

$$\delta = m \cdot \theta \left(\frac{1}{\sigma} \right)^n \quad (4.35)$$

donde para compresores, n toma valor 0.5, y $m = 0,23 \cdot \left(2\frac{a}{l}\right)^2 + \frac{\alpha_2}{500}$

Capítulo 5

Leyes torsionales

Todos los apartados anteriores que se han estado explicando, cumplían la hipótesis de flujo bidimensional. Sin embargo, ¿qué ocurre cuando esta hipótesis no se cumple? ($H/D_m > 1/9$) [9]

- las velocidades tangenciales varían con el radio.
- no se pueden despreciar las velocidades radiales.

La principal consecuencia de todo ello es que la deflexión varía con el radio, es decir, se tiene un **álabe torsionado**. En compresores resulta fundamental la torsión de los álabes, debido a la gran facilidad de desprendimiento que tienen. El objetivo será por tanto encontrar la geometría adecuada.

La ecuación de Euler ya no es válida al evaluarla únicamente en el radio medio. En su lugar, se deben utilizar las ecuaciones de Navier-Stokes, de las cuales la ecuación de Euler no es más que una simplificación para el caso 2D. A partir de estas ecuaciones, y en combinación con la ecuación de los gases ideales, se adoptan distintas hipótesis:

- Flujo estacionario, adiabático y no viscoso.
- Flujo axilsimétrico.
- Se desprecia la componente radial del gradiente de velocidad en dirección axial.
- Proceso homoentrópico.

Todo ello lleva a la ecuación de equilibrio radial simplificado, que muestra la relación entre la velocidad axial y la velocidad tangencial, y por ende, la relación entre la velocidad axial y la componente radial ($c_u = F(r)$): [9]

$$\frac{\delta h_0}{\delta r} = \frac{c_u}{r} \frac{\delta(r c_u)}{\delta r} + c_a \frac{\delta c_a}{\delta r} \quad (5.1)$$

La ecuación de equilibrio requiere una condición de contorno, y a eso es a lo que se le denomina ley torsional $c_u = F(r)$, de las cuales, existen varias que se van a describir a continuación.

Las leyes torsionales más comunes mantienen el trabajo específico constante. Como el trabajo específico se calculaba como $\hat{w} = U(c_{u2} - c_{u1})$, entonces la resta de las velocidades tangenciales debe ser inversamente proporcional al radio, lo que deja la siguiente relación genérica para compresores:

$$c_{u2} = a \cdot r^n + \frac{b_2}{r} \quad c_{u1} = a \cdot r^n - \frac{b_1}{r} \quad (5.2)$$

Será cada ley torsional la que determine el valor de dichos coeficientes.

5.0.1. Ley del torbellino libre ($n = -1$)

Esta ley implica un movimiento irrotacional del flujo, lo que implica que:

$$\frac{dc_a}{dr} = 0 \quad \frac{dr \cdot c_u}{dr} = 0 \quad (5.3)$$

La velocidad tangencial en esta ley se calcula como:

$$c_{u1} = \frac{b_1}{r} \quad c_{u2} = \frac{b_2}{r} \quad (5.4)$$

El trabajo específico queda por tanto:

$$\hat{w} = \Omega \cdot r \left(\frac{b_2}{r} - \frac{b_1}{r} \right) = \Omega(b_2 - b_1) \quad (5.5)$$

A partir de aquí también se podrían calcular el grado de reacción o la evolución de los distintos ángulos. Algunos de los inconvenientes que presenta esta ley son:

- Aumento creciente de las deflexiones en el rotor, lo que implica más pérdidas y menos rendimiento
- Fuerte aumento del número de Mach en la entrada del rotor, afectando al rendimiento.
- Fuerte torsión del álabe, lo que implica dificultades constructivas

5.0.2. Ley exponencial ($n = 0$)

En este caso, existe una variación de la componente tangencial menor a la ley anterior. El parámetro b se considera constante.

La velocidad también se halla como:

$$c_{u1} = a - \frac{b}{r} \quad c_{u2} = a + \frac{b}{r} \quad (5.6)$$

Y se puede deducir que la velocidad axial se calcula como:

$$c_{a1} = (cte - 2(a^2 \cdot \ln r + a \cdot \frac{b}{r}))^{\frac{1}{2}} \quad c_{a2} = (cte - 2(a^2 \cdot \ln r - a \cdot \frac{b}{r}))^{\frac{1}{2}} \quad (5.7)$$

El trabajo específico queda:

$$\hat{w} = \Omega \cdot r \left(A + \frac{B}{r} - \left(A - \frac{B}{r} \right) \right) \quad (5.8)$$

sustituyendo:

$$\hat{w} = 2 \cdot b \cdot \Omega \quad (5.9)$$

5.0.3. Ley de grado de reacción constante ($n = 1$)

En esta ley también se mantiene constante el parámetro b .

La velocidad tangencial es:

$$c_{u1} = a - \frac{b}{r} \quad c_{u2} = a + \frac{b}{r} \quad (5.10)$$

Y de igual manera a la ley anterior, la velocidad axial queda:

$$c_{a1} = (cte - 2a(a \cdot r^2 - 2b \cdot \ln r))^{\frac{1}{2}} \quad c_{a2} = (cte - 2a(a \cdot r^2 + 2b \cdot \ln r))^{\frac{1}{2}} \quad (5.11)$$

Manteniendo constante el grado de reacción se tiene:

$$R = 1 - \frac{c_{u1} + c_{u2}}{2U} \quad (5.12)$$

entonces:

$$c_{u1} + c_{u2} = 2U(1 - R) \quad (5.13)$$

Por tanto el trabajo específico queda:

$$\hat{w} = 2 \cdot b \cdot \Omega \quad (5.14)$$

El principal problema de esta ley torsional, es que no cumple la ecuación del equilibrio radial.

Parte III

Diseño termodinámico

Capítulo 6

Consideraciones e hipótesis

A continuación, se va a definir termodinámicamente el problema a tratar, tanto en caso bidimensional como tridimensional.

6.1. Hipótesis

En primer lugar, se han tomado las siguientes hipótesis de partida:

- Se usará aire seco como fluido.
- Se tomarán calores específicos variables, según la ecuación:

$$\frac{c_p}{R} = c_1 + c_2 \cdot T + c_3 \cdot T^2 + c_4 \cdot T^3 + c_5 \cdot T^4 \quad (6.1)$$

donde los coeficientes mostrados varían para cada gas, tomando como referencia el libro de la bibliografía. Estos coeficientes son válidos hasta valores de 1000 K. Se muestran en la siguiente tabla:

Gas	$c_1 (K^{-1})$	$c_2 (K^{-2})$	$c_3 (K^{-3})$	$c_4 (K^{-4})$	$c_5 (K^{-5})$
Aire	3,653	$-1,337 \cdot 10^{-3}$	$3,394 \cdot 10^{-6}$	$-1,913 \cdot 10^{-9}$	$0,2763 \cdot 10^{-12}$

Tabla 6.1: Valores de las constantes para el aire. [13]

Dicha variación de calor específico se puede ver en la figura 6.1.

Para cada escalonamiento, se calculará dicho calor específico a la entrada y se mantendrá constante hasta la salida. Esta aproximación se puede considerar válida ya que los saltos de presión y temperatura de cada escalonamiento no van a ser muy elevados.

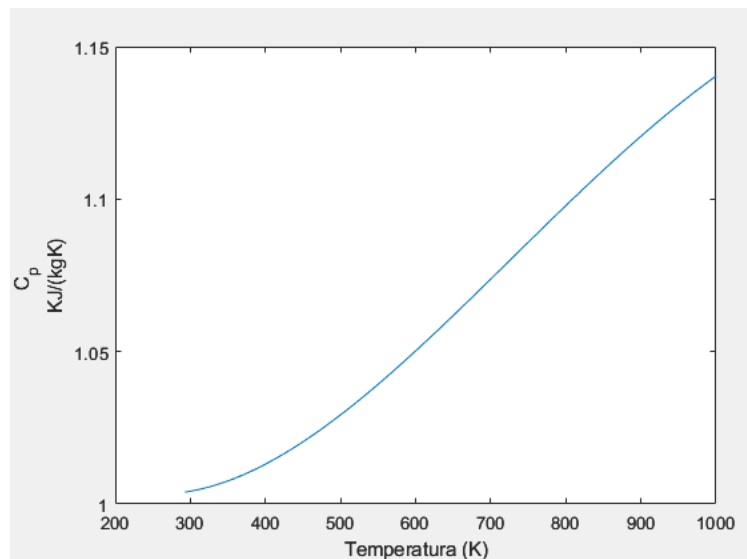


Figura 6.1: Variación del calor específico del aire a presión constante con la temperatura

- Se considera al compresor como un sistema adiabático, es decir, sin intercambio de calor con el exterior.
- Se desprecian los esfuerzos viscosos.
- Los escalonamientos serán tomados como escalonamientos de repetición. Esto implica que la velocidad de entrada y salida del escalonamiento es la misma ($c_1 = c_3$), y lo mismo sucede con sus ángulos ($\alpha_1 = \alpha_3$).
- La velocidad axial se tomará como constante en cada escalonamiento, una hipótesis bastante común en compresores axiales.
- Se considera el radio medio como constante. El prediseño bidimensional tomará como referencia dicho radio medio.
- Se considera que el flujo incide exactamente con la geometría del álabe, es decir, se considera incidencia nula.
- Se toma un valor constante de 0.4 en la relación paso/cuerda.
- El número de Mach relativo a la entrada se inicializará con un valor de 0.85, que se considera el límite para evitar la aparición de flujo supersónico en el interior de la turbomáquina, lo cual supondría pérdida de rendimiento al trabajar a carga parcial. Este valor se reducirá en cada iteración hasta conseguir la compresión deseada.
- El compresor utilizará perfiles NACA 65. El perfil concreto que será utilizado será el NACA 65(10)-10. Su geometría adimensional se muestra en la siguiente imagen:

NACA 65-(10)10 Series		
$x/c \sim \%$	$y/c \sim \%$	$t_b/c \sim \%$
0	0	0
0.5	0.250	1.544
0.75	0.350	1.864
1.25	0.535	2.338
2.5	0.930	3.480
5.0	1.580	4.354
7.5	2.120	5.294
10	2.585	6.080
15	3.365	7.332
20	3.980	8.286
25	4.475	9.006
30	4.860	9.520
35	5.150	9.848
40	5.355	9.992
45	5.475	9.926
50	5.515	9.624
55	5.475	9.060
60	5.355	8.292
65	5.150	7.364
70	4.860	6.312
75	4.475	5.168
80	3.980	3.974
85	3.365	2.770
90	2.585	1.620
95	1.580	0.612
100	0	0

Figura 6.2: Distribución del perfil

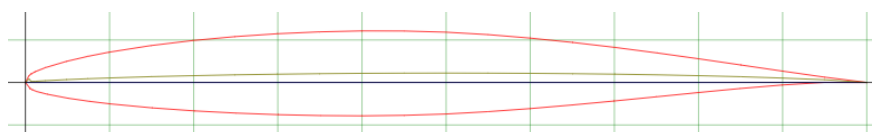


Figura 6.3: Imagen del perfil

6.2. Especificaciones de diseño

Estos serán los parámetros que podrá definir el usuario:

- Gasto másico, \dot{m} , en kg/s.
- Solidez de la cascada, σ , que podrá variar entre 1 y 1.4, con el objetivo de evitar elevadas difusiones.
- Velocidad radial, Ω , en revoluciones por minuto.
- Grado de reacción R y coeficiente de flujo Φ . Ambos parámetros deberían estar cerca de 0.5. El grado de reacción podrá ir desde 0.4 hasta 0.6, y el coeficiente de flujo tendrá el mismo límite inferior y un valor de 0.7 como máximo.
- Condiciones de remanso a la entrada (temperatura y presión), T_{01} y P_{01} , en Kelvin y Kilopascales, respectivamente.
- Relación de difusión equivalente en el rotor, DR_R , que no podrá ser mayor a 1.95, aunque es conveniente que se aproxime lo máximo a este valor.
- Relación de compresión total, rc_{Tot}

Capítulo 7

Diseño bidimensional

Una vez definidas las hipótesis de partida y las especificaciones que serán tomadas, es el momento de comenzar a determinar el prediseño bidimensional del compresor axial.

7.1. Número de escalonamientos

En primer lugar, es necesario conocer el número de escalonamientos que tendrá el compresor. Para ello, se va a tomar como referencia una relación de compresión ideal por escalonamiento de 1.25, que es óptima para compresores. Esto no significa que cada escalonamiento vaya a tener dicha relación de compresión, sino que se toma como un valor objetivo. Se debe cumplir que:

$$r_{C_{total}} = r_{C_{esc}}^n \quad (7.1)$$

siendo n el número de escalonamientos. Despejando y redondeando al alza, se puede obtener el número de escalonamientos que debería poseer el compresor.

7.2. Triángulo de velocidades

Para obtener el triángulo de velocidades se utilizan el grado de reacción, el coeficiente de flujo, y la relación de difusión equivalente. Se utiliza este último parámetro en vez de el coeficiente de carga (típicamente usado) para optimizar la geometría del perfil al máximo, sin entrar en el desprendimiento de la capa límite. Se sabe que la relación de difusión equivalente en el rotor, para una incidencia nula y tomando un valor de incidencia de referencia también como nula, se calcula como:

$$DR_R = \frac{\cos\beta_2}{\cos\beta_1} \cdot \left[1,12 + \frac{0,61}{\sigma} \cdot \cos^2\beta_1 \cdot (\tan\beta_1 - \tan\beta_2) \right] \quad (7.2)$$

Y además, se tiene que, de las expresiones obtenidas en el apartado 4.6.2:

$$R = \frac{\Phi}{2} \cdot (\tan\beta_1 + \tan\beta_2) \quad (7.3)$$

Con este sistema de dos ecuaciones y dos incógnitas se pueden obtener los valores de los ángulos β_1 y β_2 , y además:

$$R = \frac{1}{2} + \frac{\Phi}{2} \cdot (\tan\beta_2 - \tan\alpha_1) \tag{7.4}$$

$$R = 1 - \frac{\Phi}{2} \cdot (\tan\alpha_1 + \tan\alpha_2) \tag{7.5}$$

Con lo que se obtienen el valor de los ángulos α_2 y α_1 .

Finalmente, como se tiene la hipótesis de escalonamiento de repetición, el ángulo α_3 y la velocidad c_3 tienen el mismo valor que α_1 y c_1 , respectivamente.

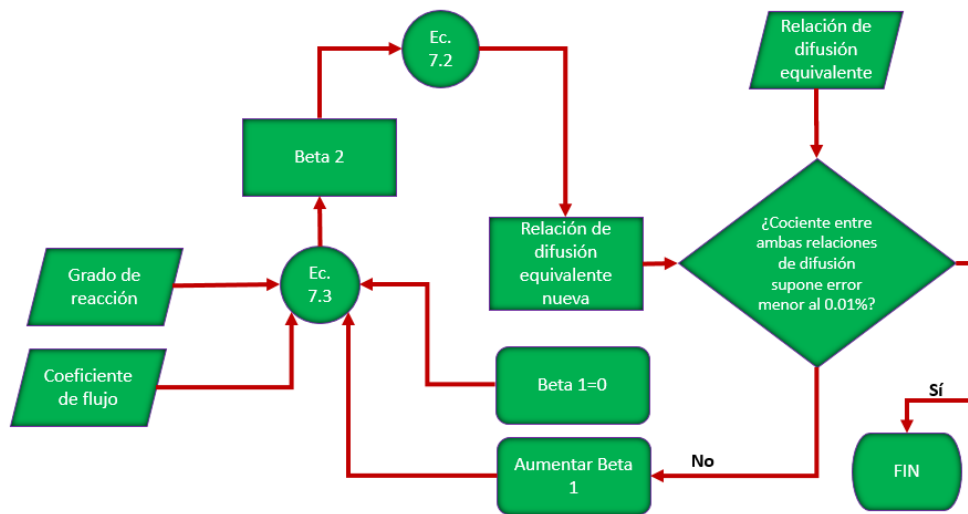


Figura 7.1: Diagrama de flujo para la determinación de los ángulos de flujo

El coeficiente de carga puede calcularse como:

$$\psi = \phi \cdot (\tan\alpha_2 - \tan\alpha_1) \tag{7.6}$$

Con ello se tiene definida la forma del triángulo de velocidades, quedando por determinar su dimensión. Para ello se utiliza el valor del número de mach a la entrada.

$$M = \frac{w_1}{\sqrt{\gamma \cdot R \cdot T_1}} \tag{7.7}$$

Faltaría por determinar la temperatura, la cual puede obtenerse de:

$$T_1 = T_{01} - \frac{c_1^2}{2 \cdot C_p} \tag{7.8}$$

Por trigonometría del triángulo de velocidades se sabe que:

$$c_1 \cdot \cos\alpha_1 = w_1 \cdot \cos\beta_1 = c_a \tag{7.9}$$

Esto provoca la reescritura de la ecuación anterior como:

$$T_1 = \frac{T_{01}}{\left[1 + \frac{M^2 \cdot (\gamma - 1)}{2} \cdot \left(\frac{\cos \beta_1}{\cos \alpha_2}\right)^2\right]} \quad (7.10)$$

Con todos estos cálculos ya sería posible definir los triángulos de velocidades.

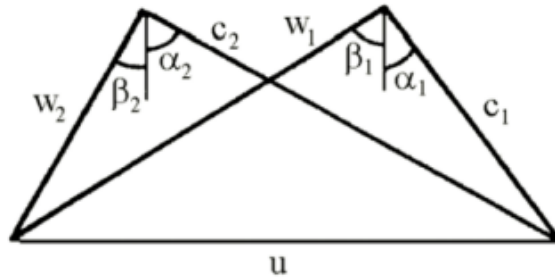


Figura 7.2: Triángulo de velocidades de un compresor axial. [12]

$$w_2 = \frac{c_a}{\cos \beta_2} \quad (7.11)$$

$$c_2 = \frac{c_a}{\cos \alpha_2} \quad (7.12)$$

$$u = w_1 \cdot \sin \beta_1 + w_2 \cdot \sin \beta_2 \quad (7.13)$$

La relación de difusión en el estátor, que también deberá salir menor a 1.95, es por tanto inmediata:

$$DR_E = \frac{\cos \alpha_1}{\cos \alpha_2} \cdot \left[1,12 + \frac{0,61}{\sigma} \cdot \cos^2 \alpha_2 \cdot (\tan \alpha_2 - \tan \alpha_1)\right] \quad (7.14)$$

7.3. Pérdidas energéticas

Una vez definido el triángulo de velocidades, se puede proseguir con el cálculo de las pérdidas energéticas, que como se ha expuesto anteriormente, se dividen en tres tipos.

7.3.1. Pérdidas en los perfiles

Para determinar las pérdidas en los perfiles, se utiliza la ecuación (4.31), ya que queda perfectamente determinada al conocer los ángulos y el valor de la relación de difusión equivalente. Simplemente hay que tener en cuenta si se está en un rotor (ángulos relativos) o en un estátor (ángulos absolutos) a la hora de calcularlas. Esto también aplica al resto de pérdidas que se muestran a continuación.

7.3.2. Pérdidas anulares

Para calcular estas pérdidas, se utiliza la ecuación (4.32). Quedaría por establecer el valor de $\frac{S}{H}$, que en este caso se toma como 0.4 de acuerdo con la literatura. [9].

Por otro lado, el ángulo α_m se calcula con la relación:

$$\tan\alpha_m = \frac{\tan\alpha_1 + \tan\alpha_2}{2} \quad (7.15)$$

7.3.3. Pérdidas secundarias

De nuevo, se sigue la ecuación 4.33 para realizar el cálculo. El coeficiente de sustentación se calcula como:

$$C_L = 2 \cdot \frac{1}{\sigma} (\tan\alpha_1 - \tan\alpha_2) \cos\alpha_m \quad (7.16)$$

7.3.4. Pérdidas totales

Las pérdidas totales se calcularán como:

$$\zeta = \zeta_p + \zeta_a + \zeta_s \quad (7.17)$$

tanto para rotor como para estátor.

Finalmente, se tienen las pérdidas como:

$$Y_{rotor} = \zeta_R \cdot \frac{w_1^2}{2} \quad (7.18)$$

$$Y_{estator} = \zeta_E \cdot \frac{c_2^2}{2} \quad (7.19)$$

Estas pérdidas, por las características elegidas para el compresor, se van a mantener constantes en todos los escalonamientos en este caso bidimensional.

7.4. Trabajo, rendimiento y relación de compresión del escalonamiento

Una vez definidas las pérdidas, se está en disposición de calcular el rendimiento del escalonamiento, para lo cual se necesitará el trabajo específico del mismo. Posteriormente, se podrá hallar también la relación de compresión de esta etapa.

El trabajo específico se puede hallar fácilmente conociendo el triángulo de velocidades:

$$W_u = u \cdot (c_{u2} - c_{u1}) \quad (7.20)$$

Este trabajo específico será constante para todos los escalonamientos, debido al mantenimiento de los triángulos de velocidades.

Con ello ya se podría hallar el rendimiento:

$$\eta_{TT} = 1 - \frac{\zeta_R \cdot \frac{w_1^2}{2} \cdot \frac{T_3}{T_2} + \zeta_E \cdot \frac{c_2^2}{2}}{W_u} \quad (7.21)$$

Finalmente, para la relación de compresión, se sabe que:

$$r_c = \frac{p_{03ss}}{p_{01}} = \left(\frac{T_{03ss}}{T_{01}} \right)^{\frac{\gamma}{\gamma-1}} \quad (7.22)$$

Dicha temperatura T_{03ss} puede obtenerse gracias a:

$$\eta_{TT} = \frac{h_{03ss} - h_{01}}{h_{03} - h_{01}} \quad (7.23)$$

Y se puede demostrar de aquí que, teniendo en cuenta la relación entre entalpía total y temperatura total está marcada por el calor específico, y que $W_u = h_{03} - h_{01}$:

$$\frac{\eta_{TT}}{c_p} = \frac{T_{03ss} - T_{01}}{W_u} \quad (7.24)$$

Y sabiendo que c_p se puede expresar como $\frac{R \cdot \gamma}{\gamma - 1}$

$$T_{03ss} = T_{01} + \frac{\eta_c \cdot W_u \cdot (\gamma - 1)}{R \cdot \gamma} \quad (7.25)$$

Como los valores del calor específico son variables, en este caso sí que se va a tener variación de la relación de compresión por escalonamiento.

7.5. Parámetros termodinámicos y geométricos

Finalmente, es posible determinar la geometría de entrada del compresor, y los parámetros termodinámicos del mismo. En primer lugar, como se tiene la velocidad tangencial y la velocidad radial, es posible calcular el diámetro medio como:

$$D_m = \frac{u}{\pi \cdot n} \quad (7.26)$$

donde n es la velocidad radial expresada en revoluciones por minuto.

Además, con la ecuación de continuidad, se sabe que:

$$\dot{m}_a = A_1 \cdot c_a \cdot \rho_1 = \pi \cdot D_m \cdot H \cdot c_a \cdot \rho_1 \quad (7.27)$$

Por lo que, si se consigue hallar el valor de la densidad en este punto, quedaría determinada la altura del álabe. Para ello, es necesario determinar la presión en dicho punto, lo cual puede realizarse mediante la relación isentrópica:

$$p_1 = p_{01} \cdot \left(\frac{T_1}{T_{01}} \right)^{\frac{\gamma}{\gamma-1}} \quad (7.28)$$

Y finalmente se obtendría la densidad:

$$\rho_1 = \frac{p_1}{R \cdot T_1} \quad (7.29)$$

Y con ello la altura del álabe en el rotor.

Se han calculado los parámetros termodinámicos a la entrada del rotor. Sin embargo, quedan por definir estas mismas variables a la entrada y salida del estátor.

Conociendo el trabajo específico, calculado en el apartado anterior, se sabe que:

$$W_u = h_{02} - h_{01} = c_p \cdot (T_{02} - T_{01}) \quad (7.30)$$

con lo que resulta inmediato el cálculo de la temperatura total a la salida del rotor, y por tanto, a la salida del estátor, ya que la temperatura total en el estátor se conserva.

Mediante la siguiente ecuación:

$$h_{02} = h_2 + \frac{c_2^2}{2} \quad (7.31)$$

Se podría obtener el valor de la temperatura T_2 , y lo mismo se haría para el valor de T_3 con la relación:

$$h_3 - h_2 = \frac{c_2^2}{2} - \frac{c_3^2}{2} \quad (7.32)$$

Para calcular la presión en el punto 2, se hace uso de la relación isentrópica:

$$\frac{P_2}{P_1} = \left(\frac{T_{2s}}{T_1} \right)^{\frac{\gamma}{\gamma-1}} \quad (7.33)$$

Sin embargo, a priori se desconoce el valor de T_{2s} . No obstante, se sabe que las pérdidas en el rotor (que ya son conocidas) se pueden calcular como

$$Perdidas_{rotor} = h_2 - h_{2s} \quad (7.34)$$

Una vez conocida la presión a la salida del rotor, se podría calcular la densidad en dicho punto conforme a la ecuación 7.29, y una vez hallada esta densidad, mediante la ecuación de continuidad se puede hallar la altura del álabe del estátor.

Para terminar de definir el primer escalonamiento y poder seguir con el diseño, es necesario calcular el valor de la presión a la salida del estátor. Para ello, se utiliza de nuevo la relación isentrópica:

$$P_3 = P_{03} \cdot \left(\frac{T_3}{T_{03}} \right) \quad (7.35)$$

7.6. Geometría de los álaves

Ya solo quedaría por determinar la geometría del álabe. Como se ha explicado en apartados anteriores, existe una desviación entre el ángulo de salida del álabe y el ángulo de salida del flujo. Por ello, se propone construir los álaves de forma que ya tengan en cuenta esa desviación, de forma que el flujo no se vea afectado, manteniendo de esta manera los triángulos de velocidades íntegros.

Se puede definir la desviación como:

$$\delta = \beta_2 - \beta_2' \quad (7.36)$$

Se sabe además, por la regla de Carter (4.35), que la desviación en una cascada de compresor puede calcularse como:

$$\delta = 0,23 \cdot \left(2 \frac{a}{l}\right)^2 + \frac{\beta_2}{500} \cdot \theta \left(\frac{1}{\sigma}\right)^{0,5} \quad (7.37)$$

para una cascada de rotor. En caso de un estator, simplemente se cambiaría el ángulo β_2 por α_2 . El valor de $\frac{a}{l}$ se toma como 0.5, al tratarse los perfiles NACA 65 de arcos de circunferencia.

Por otro lado, la curvatura se sabe que es:

$$\theta = \beta_{1'} - \beta_2' \quad (7.38)$$

Si se imponem que la incidencia en el rotor sea nula, entonces los ángulos β_1 y $\beta_{1'}$ coinciden. Por tanto, quedaría un sistema de dos ecuaciones y dos incógnitas, pudiendo obtener de esta manera la desviación en el rotor, y lo mismo se haría con el estator, utilizando los ángulos pertinentes.

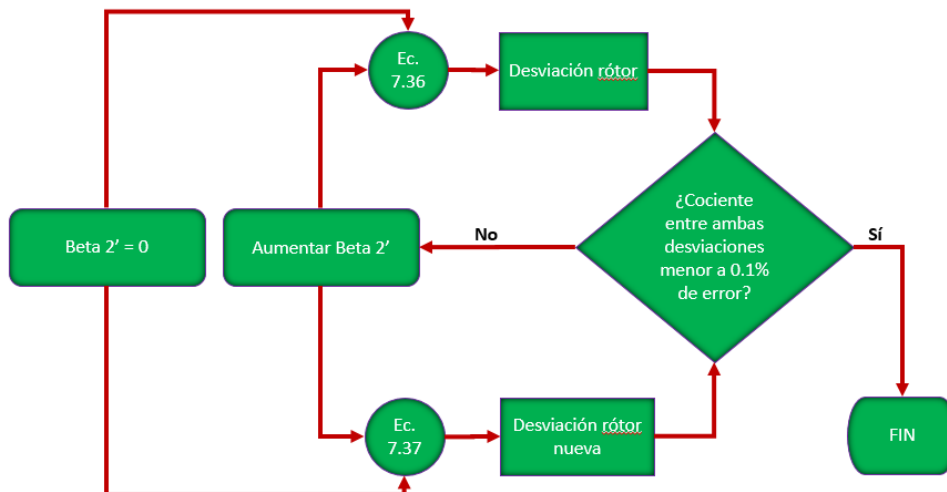


Figura 7.3: Diagrama de flujo para el cálculo de la desviación en el rotor

Como los escalonamientos son de repetición, los valores calculados serían válidos para todas las cascadas del compresor, quedando así determinada la geometría de los álabes.

7.7. Resolución del resto de escalonamientos

Para resolver el resto de escalonamientos, simplemente se toman los valores de las propiedades termodinámicas de salida del estator como las variables de entrada del siguiente rotor, repitiéndose el proceso de cálculo, pero esta vez conociendo los triángulos de velocidades, trabajo específico, pérdidas... Por lo que el cálculo es algo más sencillo.

Como se ha dicho, las variables de salida del estator son las mismas que las variables de entrada del siguiente rotor, es decir:

$$P_{01}(i+1) = P_{03}(i) ; P_1(i+1) = P_3(i) \quad (7.39)$$

$$T_{01}(i+1) = T_{03}(i) ; T_1(i+1) = T_3(i) \quad (7.40)$$

Esto implica también una igualdad entre densidades $\rho_1(i+1) = \rho_3(i)$.

Una vez que se tienen las variables de entrada de este escalonamiento, como los triángulos de velocidades son ya conocidos, se pueden obtener el resto de parámetros, comenzando por la salida del rotor, de la siguiente manera:

$$T_{02} = T_{01} + \frac{W_u}{c_p} ; T_{02} = T_{03} \quad (7.41)$$

$$T_2 = T_1 + \frac{W_u + \frac{c_1^2}{2} - \frac{c_2^2}{2}}{c_p} \quad (7.42)$$

$$P_2 = P_1 \cdot \left(\frac{T_{2s}}{T_1} \right)^{\left(\frac{\gamma}{\gamma-1} \right)} \quad (7.43)$$

$$P_{02} = P_{01} \cdot \left(\frac{T_2}{T_{02}} \right)^{\left(\frac{\gamma}{\gamma-1} \right)} \quad (7.44)$$

Por otra parte, para el cálculo de las variables de salida del estátor, se tiene:

$$T_3 = T_2 + \frac{\frac{c_2^2}{2} - \frac{c_3^2}{2}}{c_p} \quad (7.45)$$

$$P_{03} = P_{01} \cdot r_c \quad (7.46)$$

$$P_3 = P_{03} \cdot \left(\frac{T_3}{T_{03}} \right)^{\left(\frac{\gamma}{\gamma-1} \right)} \quad (7.47)$$

Quedarían por determinar las densidades y los parámetros geométricos como las alturas de los álabes, algo que se puede obtener de la misma manera que para el primer escalonamiento.

Una vez concluido el cálculo, se debe asegurar que la relación de compresión total que se obtiene, es similar a la especificada. En caso contrario, lo que se hará será reducir el valor de las velocidades de entrada hasta que los valores de ambas relaciones de presión sean próximas, como se explicó anteriormente.

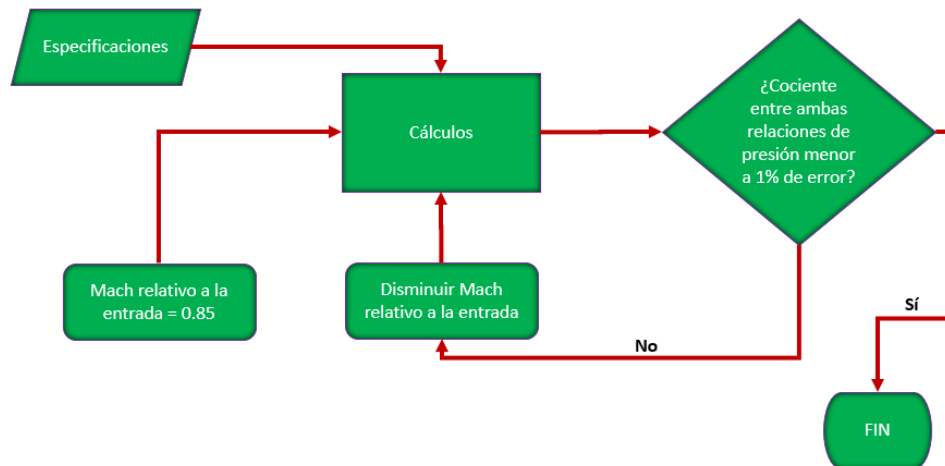


Figura 7.4: Diagrama de flujo del proceso

Para terminar, es interesante calcular el rendimiento total a total del compresor completo, que se puede hallar como:

$$\eta_{TT} = \frac{h_{0Bs} - h_{0A}}{W_{total}} \quad (7.48)$$

donde se recuerda que los puntos A y B son entrada y salida del compresor, respectivamente. El trabajo específico total es sencillo de calcular, basta con multiplicar el trabajo específico por escalonamiento por el número de estos. Faltaría por determinar h_{0Bs} , que se obtendría como:

$$h_{0Bs} = T_{0Bs} \cdot c_p \quad (7.49)$$

Y finalmente, T_{0Bs} se puede hallar como:

$$T_{0Bs} = T_{0A} \cdot \left(\frac{P_{0B}}{P_{0A}} \right)^{\left(\frac{\gamma-1}{\gamma} \right)} \quad (7.50)$$

Con dicha temperatura, ya se tendría el valor de h_{Bs} y con ello, el valor del rendimiento completo del compresor, el cual debe ser menor al de los escalonamientos.

Capítulo 8

Diseño tridimensional

Como se explicó en la parte de Estado de la Tecnología, si la relación entre la altura del álabe y el diámetro medio es demasiado grande $\left(\frac{H}{D_m} < \frac{1}{9}\right)$ [9], la hipótesis de bidimensionalidad deja de ser válida, y el perfil pasa a tener dependencia radial. Por tanto, será necesario el uso de las tres leyes torsionales definidas anteriormente. En este diseño se abordarán todas ellas, haciendo hincapié en la forma de los triángulos de velocidades en función del radio. De esta manera se podrá conocer también el lugar donde se produce el número de Mach máximo, y su valor, pudiendo verificar que este no supera el límite de 0.85. Además, se analizarán las pérdidas producidas para este caso tridimensional.

Para llevar a cabo este apartado, se van a considerar nueve puntos de cálculos:

- A la altura de la raíz del álabe.
- A la altura del radio medio, al igual que en el caso bidimensional.
- A la altura de la cabeza del álabe.
- Los puntos intermedios, de forma que haya equidistancia.

De esta manera se podrá observar la evolución seguida de los distintos parámetros calculados a lo largo del álabe.

En este caso tridimensional, los cálculos que se van a llevar a cabo son los siguientes:

- Triángulos de velocidades en función del radio: Ahora la hipótesis de tomar todo el triángulo de velocidades con su valor en el radio medio no es válida.
- Cálculo de las pérdidas en función del radio: de forma similar a lo que se realiza con los triángulos aplica también a las pérdidas, que se van a ver modificadas en este caso tridimensional.
- Coeficientes adimensionales.
- Determinación de los rendimientos total a total de cada escalonamiento

8.1. Ley de torbellino libre

En esta ley, la velocidad axial y el trabajo específico se mantienen constante, y existe la siguiente relación entre la componente tangencial de las velocidades absolutas y el radio:

$$c_{u1} = \frac{b_1}{r_1} \quad ; \quad c_{u2} = \frac{b_2}{r_2} \quad (8.1)$$

Como se tiene el caso bidimensional resuelto, es posible particularizar esas dos relaciones para el radio medio, y así obtener los valores de las constantes b_1 y b_2 .

Teniendo en cuenta que se conoce la velocidad axial, por conservarse la del problema bidimensional, se tiene entonces que terminar de hallar las relaciones para el triángulo de velocidades. A continuación se muestran todas las expresiones para determinarlas.

$$u(r) = \frac{u_m}{r_m} \cdot r \quad (8.2)$$

$$c_1(r) = \sqrt{c_a^2 + c_{u1}^2} \quad (8.3)$$

$$c_2(r) = \sqrt{c_a^2 + c_{u2}^2} \quad (8.4)$$

$$w_{u1}(r) = u(r) - c_{u1}(r) \quad ; \quad w_{u2}(r) = u(r) - c_{u2}(r) \quad (8.5)$$

$$w_1(r) = \sqrt{c_a^2 + w_{u1}^2} \quad (8.6)$$

Las velocidades $c_2(r)$ y $w_2(r)$ se determinarían de manera análoga. Finalmente, los ángulos:

$$\beta_1(r) = \arctan\left(\frac{w_{u1}(r)}{c_a}\right) \quad ; \quad \beta_2(r) = \arctan\left(\frac{w_{u2}(r)}{c_a}\right) \quad (8.7)$$

$$\alpha_1(r) = \arctan\left(\frac{c_{u1}(r)}{c_a}\right) \quad ; \quad \alpha_2(r) = \arctan\left(\frac{c_{u2}(r)}{c_a}\right) \quad (8.8)$$

También es posible analizar la variación del grado de reacción con el radio.

$$R(r) = \frac{w_{u1}(r) + w_{u2}(r)}{2 \cdot u(r)} \quad (8.9)$$

Una buena práctica cuando se resuelve el caso tridimensional, es calcular cuál es el valor máximo del número de Mach, y comprobar que no existe flujo supersónico. Para ello, se debe calcular la ley de temperaturas en función del radio.

$$T_1(r) = T_{01} - \frac{c_1^2(r)}{2 \cdot C_p} \quad (8.10)$$

Con esto se pueden hallar tanto el Mach en la raíz (asociado con la la velocidad w_1) como el Mach en la cabeza (vinculado con c_1). Estos valores del número de Mach serán los máximos de todo el compresor, lo cual será útil para conocer, en efecto, cuál es el valor máximo del número de Mach en el compresor diseñado, aun sabiendo que en ningún caso su valor excederá la cifra de 0.85.

$$M_c = \frac{c_1(r)}{\sqrt{\gamma \cdot R \cdot T_1(r)}} \quad (8.11)$$

$$M_r = \frac{w_1(r)}{\sqrt{\gamma \cdot R \cdot T_1(r)}} \quad (8.12)$$

Para el cálculo de pérdidas y del rendimiento, se procede igual que para el caso bidimensional, solo que ahora la relación de difusión equivalente depende del radio, y por tanto las pérdidas también lo harán. Sin embargo, es importante hacer notar que las pérdidas anulares y secundarias no dependen del radio.

$$DR_{eR}(r) = \frac{\cos\beta_2(r)}{\cos\beta_1(r)} \left[1,12 + \frac{0,61}{\sigma} \cdot \cos^2\beta_1(r) \cdot (\tan\beta_1(r) - \tan\beta_2(r)) \right] \quad (8.13)$$

$$DR_{eE}(r) = \frac{\cos\alpha_1(r)}{\cos\alpha_2(r)} \left[1,12 + \frac{0,61}{\sigma} \cdot \cos^2\alpha_2(r) \cdot (\tan\alpha_2(r) - \tan\alpha_1(r)) \right] \quad (8.14)$$

Es posible hallar el rendimiento total a total, recordando que el trabajo específico se mantiene constante, como:

$$W_{is}(r) = W_i - Y_e(r) - \frac{T_3(r)}{T_2(r)} Y_r(r) ; \quad \eta_{TT} = \frac{W_{is}}{W_i} \quad (8.15)$$

8.2. Ley exponencial

Esta ley, a diferencia de la anterior, no mantiene la velocidad axial constante, pero sí el trabajo específico. En este caso, las variaciones de velocidad tangencial responden a las ecuaciones:

$$c_{u2} = a + \frac{b}{r} ; \quad c_{u1} = a - \frac{b}{r} \quad (8.16)$$

Es posible demostrar que para un compresor axial, y suponiendo un proceso homentrópico y homentálpico, las velocidades axiales siguen las siguientes igualdades:

$$c_{a1} = (cte_1 - 2(a^2 \cdot \ln r + a \cdot \frac{b}{r}))^{0,5} \quad (8.17)$$

$$c_{a2} = (cte_2 - 2(a^2 \cdot \ln r - a \cdot \frac{b}{r}))^{0,5} \quad (8.18)$$

Donde las constantes pueden hallarse imponiendo una condición de contorno, normalmente en el radio medio.

Una vez halladas las leyes de velocidades axiales, el resto de los cálculos se realizan igual que para la ley torsional anterior, obteniéndose los triángulos de velocidades, grado de reacción, números de Mach, relaciones de difusión en el rotor y en el estátor, pérdidas y rendimiento.

8.3. Ley de grado de reacción constante

Esta ley es del mismo estilo que la exponencial, manteniéndose el trabajo específico constante, pero no la velocidad axial. Las velocidades tangenciales se obtienen de:

$$c_{u2} = a \cdot r + \frac{b}{r} ; \quad c_{u1} = a \cdot r - \frac{b}{r} \quad (8.19)$$

Y las velocidades axiales se obtienen como:

$$c_{a1} = (cte_1 - 2a(a \cdot r^2 - 2b \cdot \ln r))^{0,5} \quad (8.20)$$

$$c_{a2} = (cte_2 - 2a(a \cdot r^2 + 2b \cdot \ln r))^{0,5} \quad (8.21)$$

El resto de cálculos se realizarían de la misma manera que las leyes anteriores.

Parte IV

Software utilizado

Capítulo 9

Matlab y App Designer

Como se definió en un principio, el software a utilizar para programar este diseño termodinámico será *Matlab*, en concreto la versión *Matlab R2022B*. La elección de este software se debe a varios motivos:

- Se trata de un software utilizado en el Grado de Ingeniería Aeroespacial, por lo que resulta familiar para los estudiantes.
- Es un software especializado en trabajar con vectores y matrices, lo cual facilita enormemente la programación y reduce enormemente el tiempo de cálculo.

A continuación, se explicará el proceso seguido, mediante el uso de funciones, que posteriormente serán implementadas en la herramienta informática. Todos los cálculos realizados están directamente relacionados con lo explicado en la parte de Diseño Termodinámico. Para ver en detalle partes del código, se recomienda visitar la última parte del proyecto, *Código Utilizado VII*.

9.1. Programación del caso bidimensional

La función encargada de calcular el caso bidimensional se trata de *f_caso_2D*. Esta función recibe los valores de las especificaciones requeridas y obtiene todos los valores que se requeridos para el cálculo bidimensional.

Es importante añadir que esta función llama a su vez a otras subfunciones. Estas subfunciones son las siguientes:

- *f_triangulo_de_velocidades*. Esta función se encarga de calcular los parámetros relativos a los triángulos de velocidades, que en el caso bidimensional son constantes en cada escalonamiento, por lo cual es útil definir una función para dicho cálculo.
- *f_geometria_entrada*. La geometría de entrada es importante para definir el punto de partida geométrico del compresor, por lo que también se define como una función independiente.
- *f_perdidas*. Otro cálculo que es relativamente sencillo de hallar y que es constante en cada escalonamiento son las pérdidas energéticas.
- *f_trabajo*. Finalmente, existe una función definida para el cálculo del trabajo por escalonamiento.

9.2. Programación del caso tridimensional

El caso tridimensional toma como punto de partida las variables obtenidas en el caso bidimensional, ya que como ya se sabe, la manera de determinar los nuevos parámetros mediante leyes torsionales, dependen de unas constantes que se obtienen utilizando el radio medio como condición de contorno.

Existen tres funciones diferentes, una para cada ley torsional.

- *f_torbellino_libre*. Realiza los cálculos utilizando la ley torsional de torbellino libre.
- *f_exponencial*. Uso de la ley exponencial para determinar el resultado tridimensional.
- *f_reaccion_constante*. De forma análoga, realiza los cálculos utilizando la ley de grado de reacción constante.

Con todo ello se concluiría el uso de scripts para la realización de los cálculos, debiendo implementar estas funciones dentro de *App Designer*

9.3. Uso de *App Designer*

Para llevar a cabo la elaboración de la interfaz gráfica, se utilizará el módulo de matlab llamado *App Designer*, que permite diseñar una interfaz gráfica de usuario de forma sencilla y cómoda.

Dentro del código de esta herramienta, se ejecutará en primer lugar la función *f_caso_2D*, de forma que se obtengan los cálculos bidimensionales. Esto provocará que se muestren una serie de resultados, que serán comentados en la siguiente parte, V Ejemplo de Uso. Posteriormente, para el cálculo tridimensional, se podrá elegir entre cualquiera de las tres leyes torsionales, ejecutándose el código de las mismas y mostrando los resultados tridimensionales.

Entre las posibilidades que ofrece este módulo, se describen las más importantes para este proyecto:

- *Edit Field (Numeric)*: se trata de un campo numérico, que puede o bien recibir valores o devolverlos.
- *Label*: No es más que un manera de colocar texto en la interfaz.
- *Button*: Botón, permite realizar una operación, que puede serle ordenada mediante un *Callback*.
- *Axes*: Ejes. Permite mostrar gráficas con ejes definidos.
- *Table*: Tabla: Otorga la posibilidad de realizar tablas.

Todas las gráficas y tablas que se muestran en la app informática están calculadas dentro del propio código de la misma.

Como se ha comentado anteriormente, la asignación a los botones (concretamente, el botón que calcula el caso bidimensional, y los tres botones para calcular las leyes torsionales) se lleva a cabo mediante *Callbacks*. En estos *Callbacks* se coloca el código a ejecutarse cuando se pulse el botón.

Parte V

Ejemplo de uso

Capítulo 10

Diseño bidimensional

Para ilustrar el funcionamiento de la herramienta informática desarrollada, se va a llevar a cabo un ejemplo de cálculo.

10.1. Especificaciones y cálculo

Este caso de ejemplo tendrá las siguientes especificaciones:

- Gasto másico, \dot{m} , de 20 kg/s.
- Solidez de la cascada, σ de 1.
- Velocidad radial Ω de 11500 revoluciones por minuto.
- Grado de reacción R y coeficiente de flujo ϕ de 0.55 y 0.7 respectivamente.
- Temperatura total a la entrada de 293 Kelvin, y presión total a la entrada de 101.3 Kilopascales, es decir, temperatura ambiente y presión atmosférica.
- Relación de difusión equivalente en el rotor, DR_{eqR} , de 1.75.
- Relación de compresión total de 6.

Al abrir la aplicación informática, aparecerá la siguiente interfaz de usuario, en la cual se podrán introducir dichas especificaciones:

Especificaciones	Resultados 2D	Ley torsional	Resultados 3D
DISEÑO BIDIMENSIONAL			
Temp de remanso a la entrada (K)	<input type="text" value="293"/>		
Presión de remanso a la entrada (kPa)	<input type="text" value="101.5"/>		
Solidez de la cascada	<input type="text" value="1"/>	(Valor entre 1 y 1.4)	
Revoluciones por minuto	<input type="text" value="1.15e+04"/>		
Grado de reacción	<input type="text" value="0.55"/>	(Valor entre 0.4 y 0.6)	
Coefficiente de flujo	<input type="text" value="0.7"/>	(Valor entre 0.4 y 0.7)	
Gasto másico (kg/s)	<input type="text" value="20"/>		
Relación de difusión en el rotor	<input type="text" value="1.75"/>	(No puede ser mayor a 1.95)	
Relación de presiones total	<input type="text" value="6"/>		
<input type="button" value="CALCULAR"/>			

Figura 10.1: Pantalla para introducir las especificaciones

Estos campos numéricos estarán acotados donde corresponda, de forma que si se introduce un valor no apto, la herramienta mostrará un mensaje de error. Una vez definidos los parámetros, se puede pulsar el botón "calcular", que efectuará todos los cálculos bidimensionales.

10.2. Triángulo de velocidades

La herramienta cuenta con varias pestañas en su parte superior, que pueden ir siendo seleccionadas por el usuario. Es interesante por ejemplo comprobar los valores gráficos y numéricos del triángulo de velocidades (velocidades y ángulos), mostrados en la pestaña correspondiente.

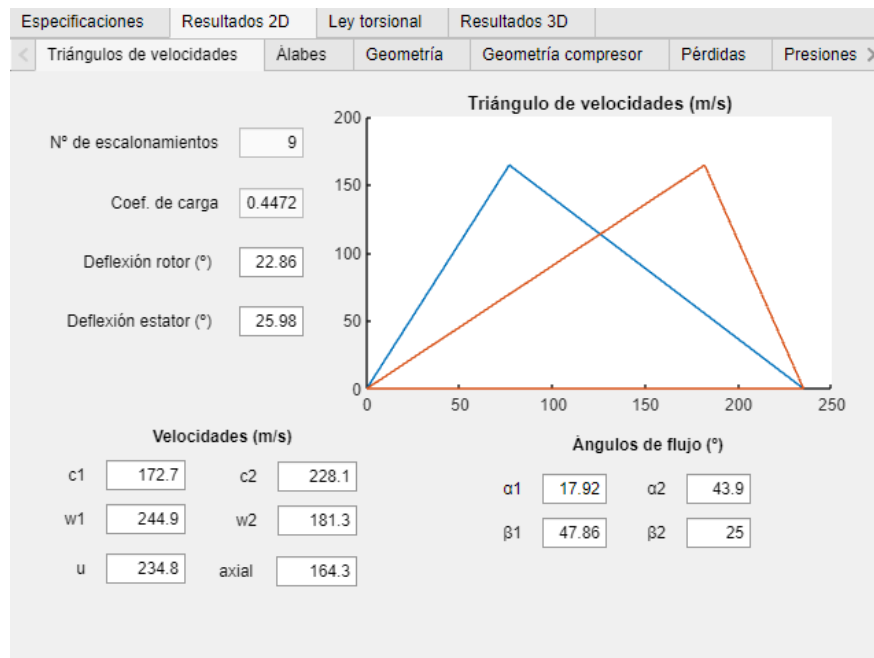


Figura 10.2: Imagen con los resultados referentes al triángulo de velocidades

Además, se muestran otros resultados interesantes, como el número de escalonamientos que tiene el compresor, y el coeficiente de carga obtenido. También se muestran las deflexiones en rotor y estator, de forma que no superan valores críticos, habitualmente deben ser menores a 40 grados. Además, el valor de coeficiente de carga se asemeja al valor óptimo de 0.45. El triángulo no es simétrico, algo coherente con el grado de reacción impuesto.

10.3. Geometría de las cascadas

La siguiente pestaña muestra una gráfica adimensionalizada de cómo se vería una cascada, además de los ángulos de flujo y de la geometría. Por otro lado, también se muestran los valores numéricos de los ángulos geométricos, la desviación, curvatura y el ángulo de calado.

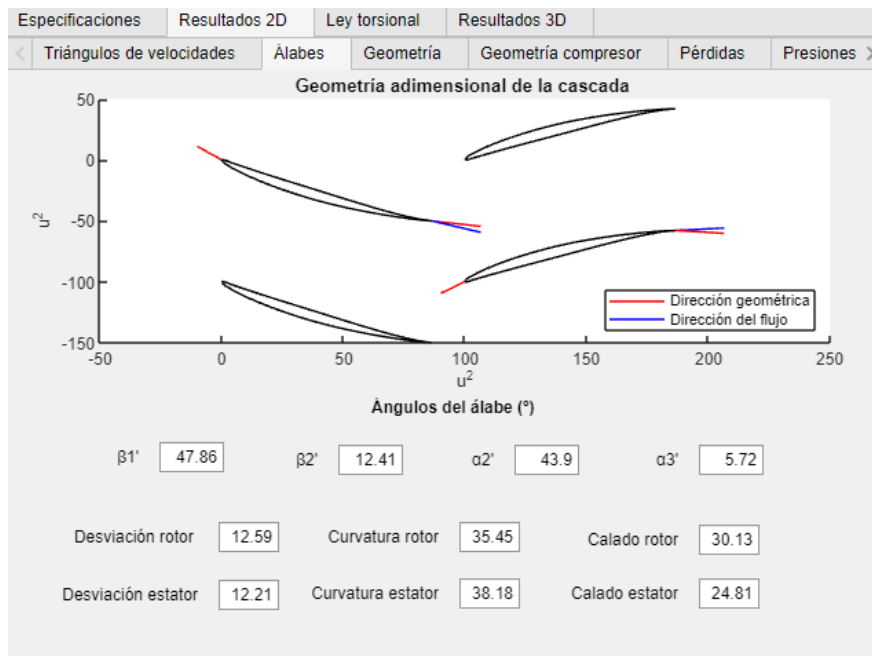


Figura 10.3: Geometría de la cascada

Como la incidencia se ha tomado como nula, la dirección del flujo y la dirección tangente a la entrada del rotor y estator son coincidentes. No ocurre lo mismo a la salida, donde el flujo se representa con color azul, y en rojo, la dirección de la salida geométrica.

Además de esto, en la siguiente pestaña se muestran los valores del paso, altura del álabe, cuerda, y el número de álaves, todo ello para cada uno de los escalonamientos, y tanto del rotor como del estator.

ESCALONAMIENTOS					
Escalonamientos	Altura del álabe (cm)	Paso (cm)	Cuerda (cm)	Nº de Álaves	
1.0000	9.3746	3.7116	3.7498	33.0000	▲
2.0000	7.7759	3.1406	3.1103	39.0000	
3.0000	6.5458	2.6060	2.6183	47.0000	
4.0000	5.5798	2.2270	2.2319	55.0000	
5.0000	4.8078	1.9138	1.9231	64.0000	
6.0000	4.1816	1.6779	1.6726	73.0000	▼

ESCALONAMIENTOS					
Escalonamientos	Altura del álabe (cm)	Paso (cm)	Cuerda (cm)	Nº de Álaves	
1.0000	8.4430	3.4023	3.3772	36.0000	▲
2.0000	7.0615	2.8485	2.8246	43.0000	
3.0000	5.9864	2.4017	2.3946	51.0000	
4.0000	5.1339	2.0414	2.0536	60.0000	
5.0000	4.4469	1.7751	1.7788	69.0000	
6.0000	3.8854	1.5504	1.5542	79.0000	▼

Figura 10.4: Valores numéricos para cada cascada

Como observación, se puede comprobar que la altura de cada rotor es mayor que la del estator correspondiente,

y que la altura global del compresor va disminuyendo conforme se avanza en el eje axial. Dicha altura puede verse en la pestaña siguiente, 'Geometría compresor', así como la altura inferior del mismo.

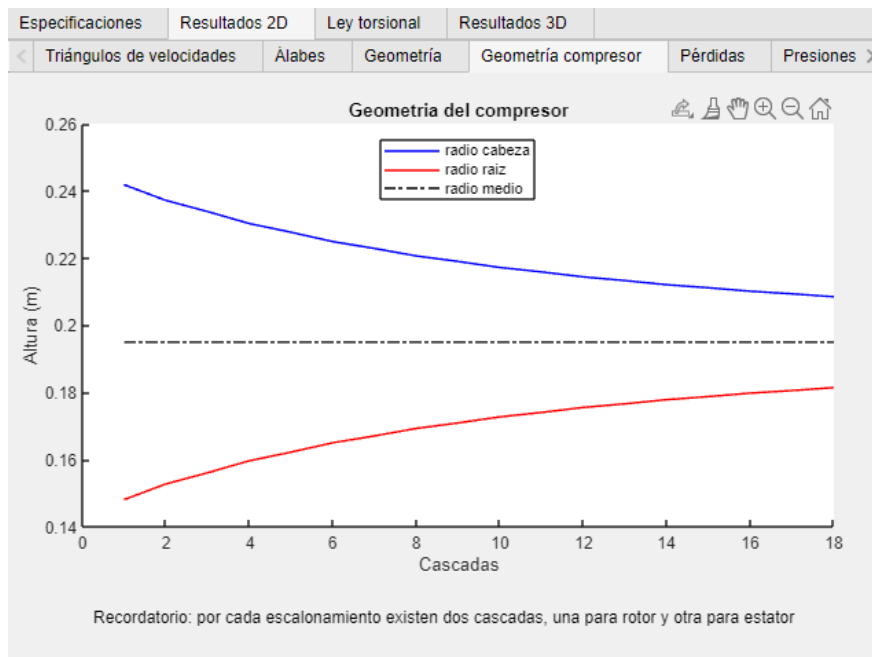


Figura 10.5: Representación de cómo se vería el perfil del compresor

10.4. Pérdidas energéticas

Las pérdidas energéticas también son mostradas en formato tabla, para el rotor y el estator, y en función del escalonamiento. Estas pérdidas se dividen en los tres tipos comentados anteriormente (perfil, anulares y secundarias), y el valor total de estas pérdidas en julios por kilogramo.

Especificaciones		Resultados 2D		Ley torsional		Resultados 3D	
< Triángulos de velocidades		Álabes		Geometría		Geometría compresor	
						Pérdidas	
						Presiones >	
ROTOR							
Escalonamientos	Coef. Perfil	Coef. Anulares	Coef. Secundarias	Pérdidas (J/kg)			
1	0.0140	0.0074	0.0168	1.1471e+03	▲		
2	0.0140	0.0074	0.0168	1.1471e+03			
3	0.0140	0.0074	0.0168	1.1471e+03			
4	0.0140	0.0074	0.0168	1.1471e+03			
5	0.0140	0.0074	0.0168	1.1471e+03			
6	0.0140	0.0074	0.0168	1.1471e+03	▼		
ESTATOR							
Escalonamientos	Coef. Perfil	Coef. Anulares	Coef. Secundarias	Pérdidas (J/kg)			
4	0.0139	0.0070	0.0181	1.0136e+03	▲		
5	0.0139	0.0070	0.0181	1.0136e+03			
6	0.0139	0.0070	0.0181	1.0136e+03			
7	0.0139	0.0070	0.0181	1.0136e+03			
8	0.0139	0.0070	0.0181	1.0136e+03			
9	0.0139	0.0070	0.0181	1.0136e+03	▼		
Relación de difusión estator		1.746		(< 1.95)			

Figura 10.6: Valores referentes a las pérdidas

Dado que el grado de reacción es ligeramente mayor que 0.5, esto implica que las pérdidas no se reparten con igualdad entre el rotor y el estator, siendo los coeficientes en el estator algo mayores.

También se muestra el resultado numérico de la relación de difusión en el estator, en la cual se debe comprobar que el valor no excede de 1.95. En este caso, por la misma razón que se ha explicado antes, el valor de esta relación de difusión equivalente en el estator es ligeramente distinta a la del rotor, que no se muestra aquí al ser un valor especificado y conocido por el usuario, y que tampoco debía ser mayor a 1.95.

10.5. Parámetros termodinámicos

Las siguientes tres ventanas muestran los parámetros termodinámicos, también para cada escalonamiento. Estos parámetros incluyen la temperatura, la temperatura total, la presión, la presión total, la densidad, y el calor específico, cada uno de ellos evaluados en los tres puntos de la cascada (entrada del rotor, salida del rotor, y salida del estator).

10.5.1. Presiones

La primera pestaña muestra resultados gráficos de la presión y la presión total en cada uno de los tres puntos de la cascada, y por escalonamiento.

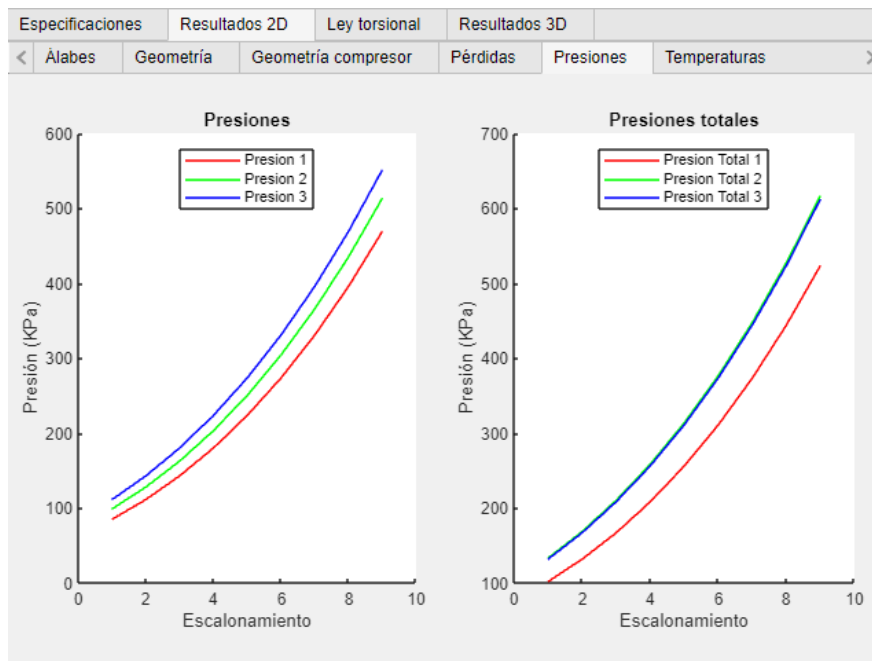


Figura 10.7: Variación de la presión y presión total por escalonamiento

Lógicamente, en un compresor ambas variables deben aumentar con cada escalonamiento. El valor de la presión total de salida es de unos 617 KPa, con lo cual se comprueba que la relación de compresión especificada (6), se cumple.

10.5.2. Temperaturas

Una pestaña idéntica pero para las temperaturas y temperaturas totales.

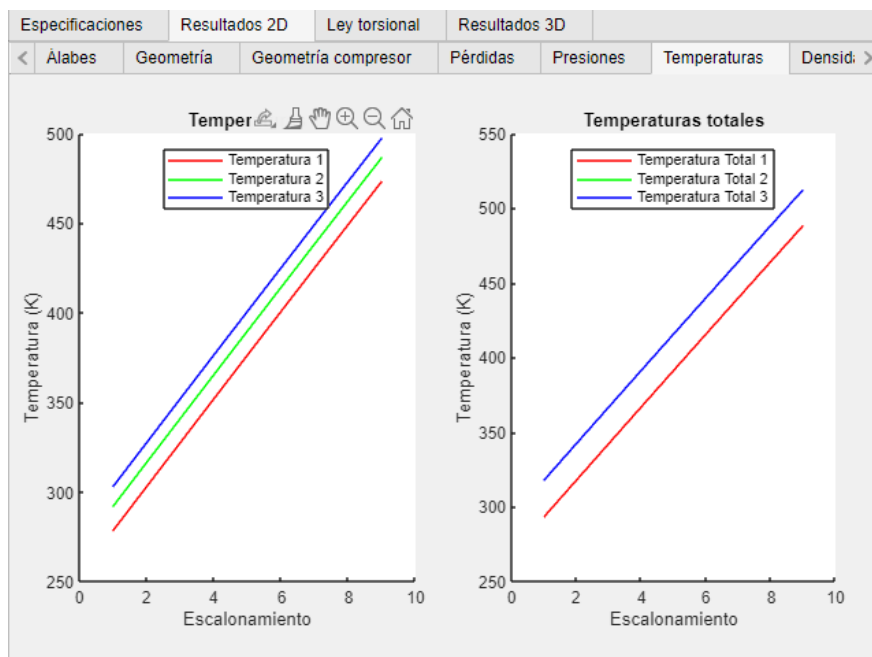


Figura 10.8: Variación de la temperatura y temperatura total por escalonamiento

De nuevo, su valor debe aumentar conforme se avanza en el compresor. Es importante advertir cómo los valores de T_{02} y T_{03} son coincidentes.

10.5.3. Densidades

Finalmente, se muestran los resultados para la densidad, que en realidad es una variable dependiente de la presión y la temperatura, y como ambas suben, a la densidad le sucederá lo mismo. También se muestran los resultados de la variación del calor específico por escalonamiento.

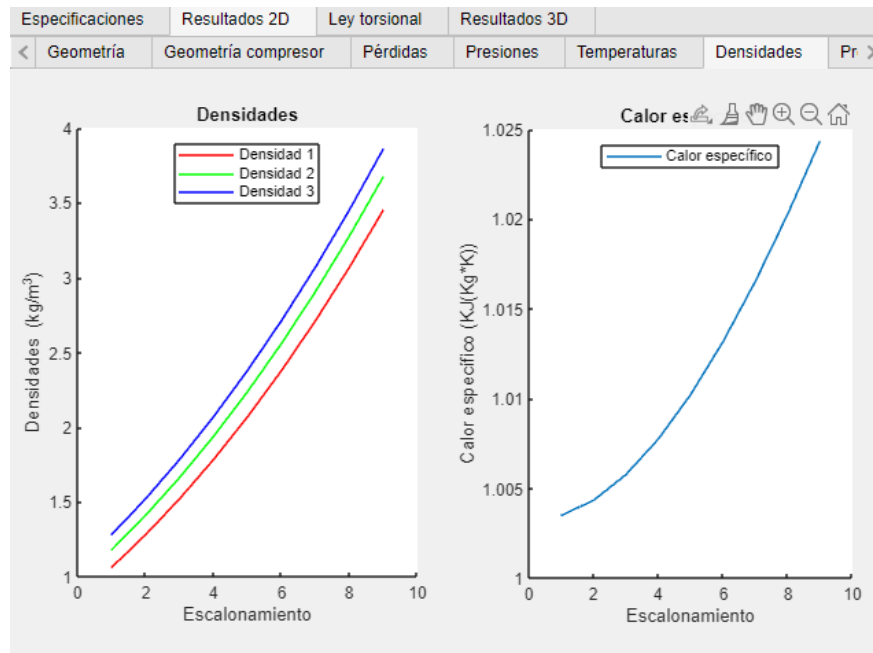


Figura 10.9: Variación de la densidad y calor específico por escalonamiento

10.5.4. Prestaciones

Finalmente, se muestra el valor del trabajo específico por escalonamiento, la relación de compresión de cada escalonamiento y el rendimiento por escalonamiento y total del compresor.

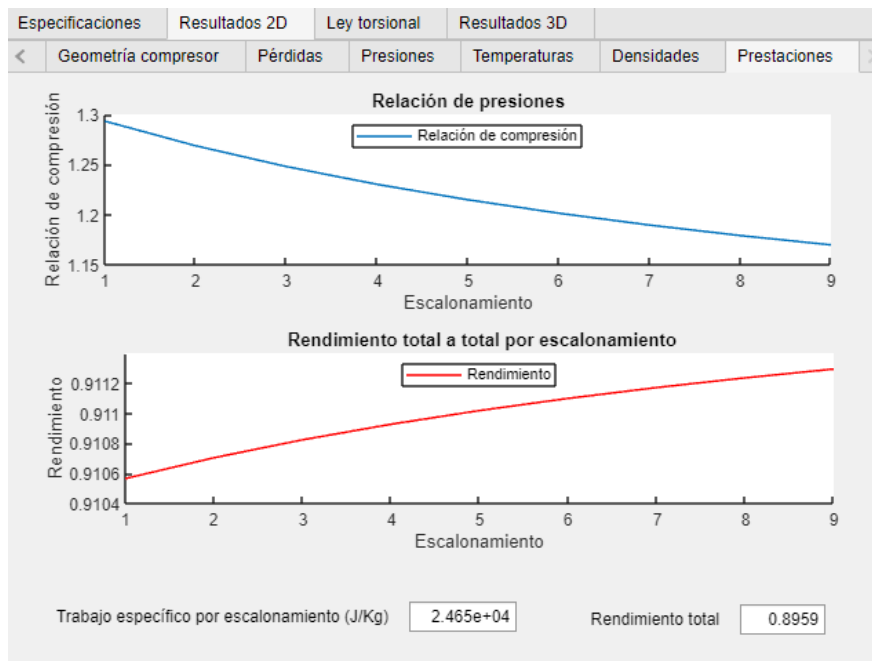


Figura 10.10: Variación de la relación de compresión por escalonamiento y valor del trabajo específico

Se puede observar cómo dicha relación de compresión va disminuyendo a medida que se avanza. Además, la relación de compresión de cada escalonamiento se asemeja bastante al valor límite definido originalmente, de 1.25.

Por otro lado el rendimiento va aumentando con cada escalonamiento, y además, se comprueba cómo el rendimiento del compresor completo es menor a los rendimientos de cualquiera de los escalonamientos, algo que concuerda con la teoría.

Capítulo 11

Diseño tridimensional

La herramienta mostrará la siguiente advertencia cuando la hipótesis bidimensional no sea válida:

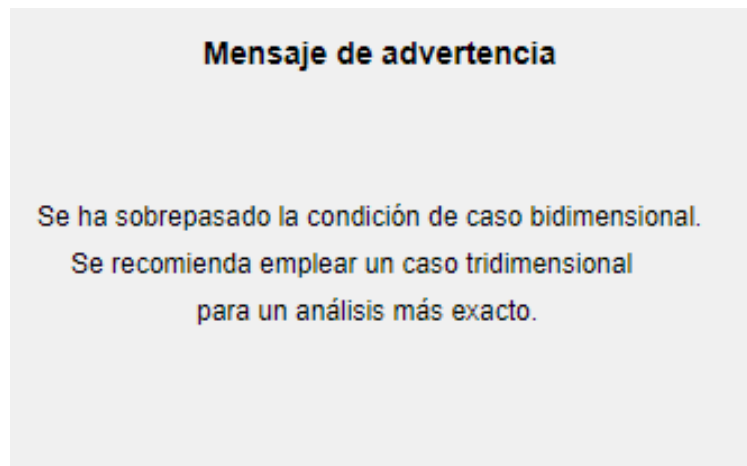


Figura 11.1: Mensaje de advertencia

Una vez comprobado el caso bidimensional, se está en disposición de realizar un análisis tridimensional. Para ello, se debe elegir la ley torsional que se va a emplear en la pestaña correspondiente.

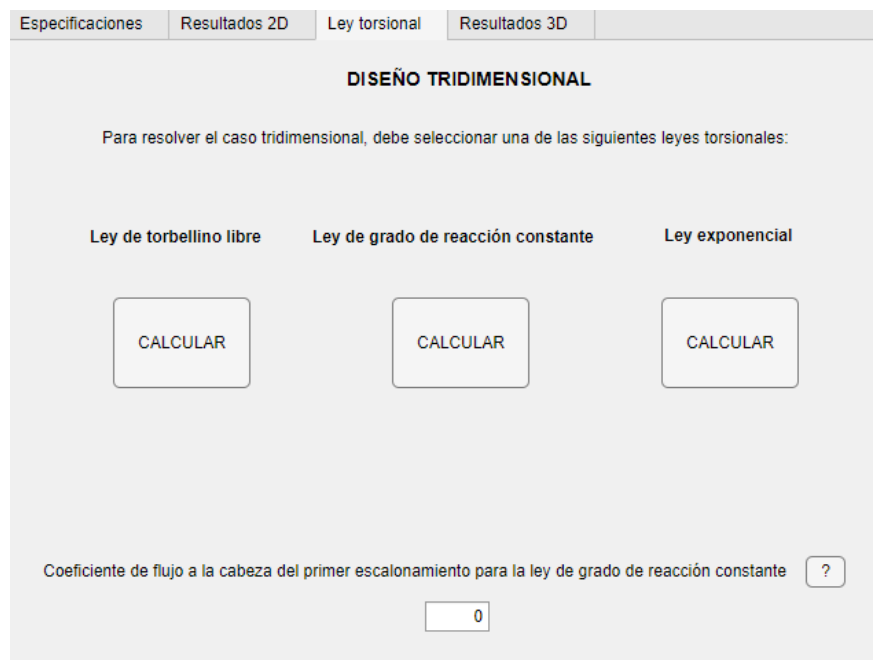


Figura 11.2: Pantalla para elegir la ley torsional

11.1. Ley de torbellino libre

11.1.1. Triángulos de velocidades

Si se elige la ley de torbellino libre, en la primera pestaña de resultados, se verán los triángulos de velocidades, tanto en la raíz, como en el radio medio, como en la cabeza del álabe, para el primer escalonamiento. En este caso se puede ver cómo la velocidad axial es constante para esta ley, así como un ángulo negativo de entrada del flujo. También se muestra el Mach máximo, que se recuerda, no debe ser superior a 0.85, aunque en esta herramienta eso no va a ser posible ya que el Mach máximo ya está impuesto.

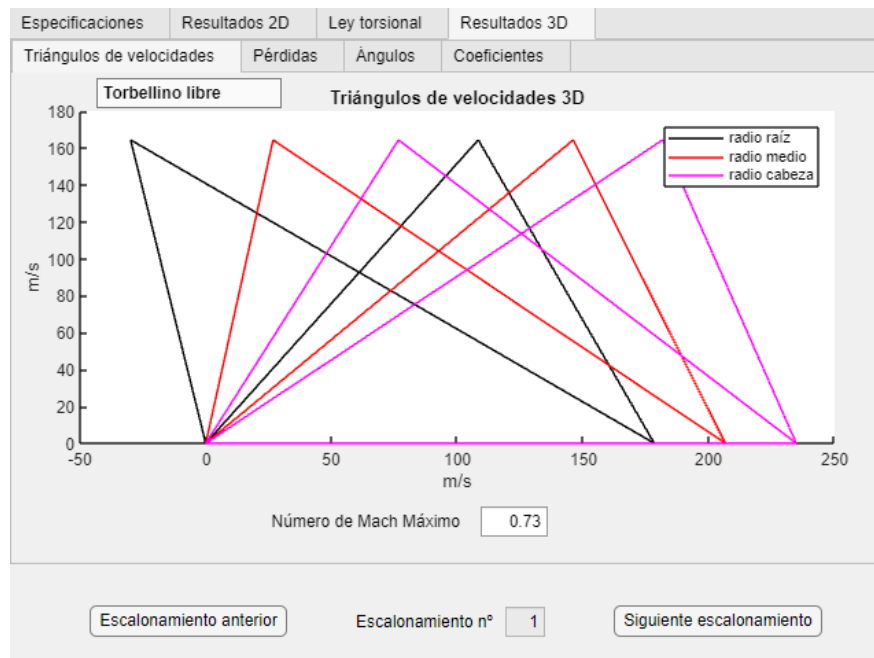


Figura 11.3: Triángulos de velocidades en el primer escalonamiento

En el caso de que se quiera comprobar el escalonamiento siguiente al actual, se debe pulsar el botón correspondiente, y viceversa con el anterior. Se puede ver cómo a medida que se avanza, las variaciones de ángulos van siendo cada vez menos agresivas.

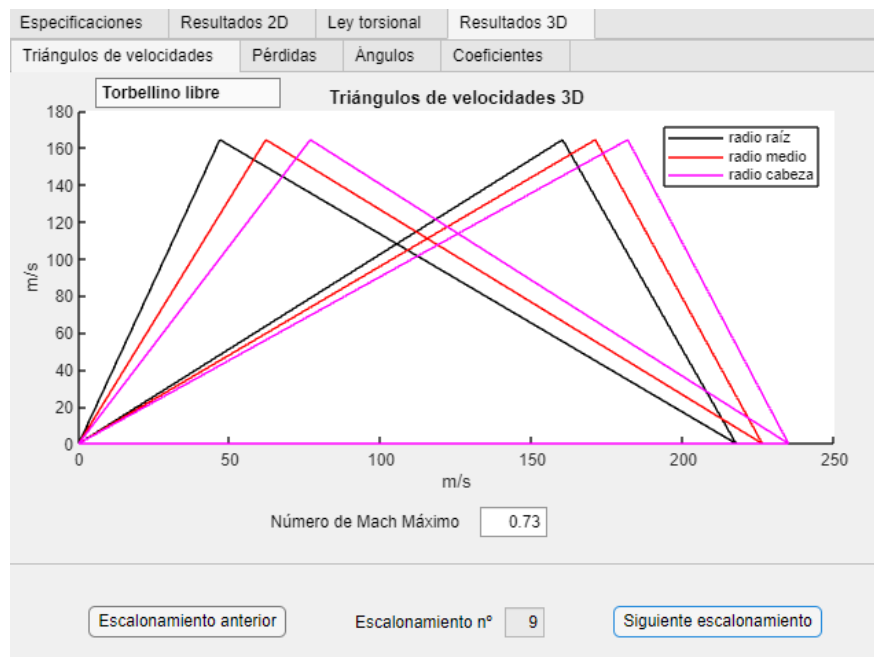


Figura 11.4: Triángulos del último escalonamiento

11.1.2. Pérdidas, trabajo y rendimiento total a total

En este apartado se muestran los valores para las pérdidas (tanto en rotor como estator), el trabajo específico, y el rendimiento total. Estos resultados se muestran para el radio medio, el radio a la raíz, y el radio a la cabeza, para el primer escalonamiento. Se pueden calcular sucesivos escalonamientos de la misma manera que para los triángulos, pulsando en los botones inferiores. Asimismo, se muestran los valores del trabajo específico y el rendimiento por escalonamiento.

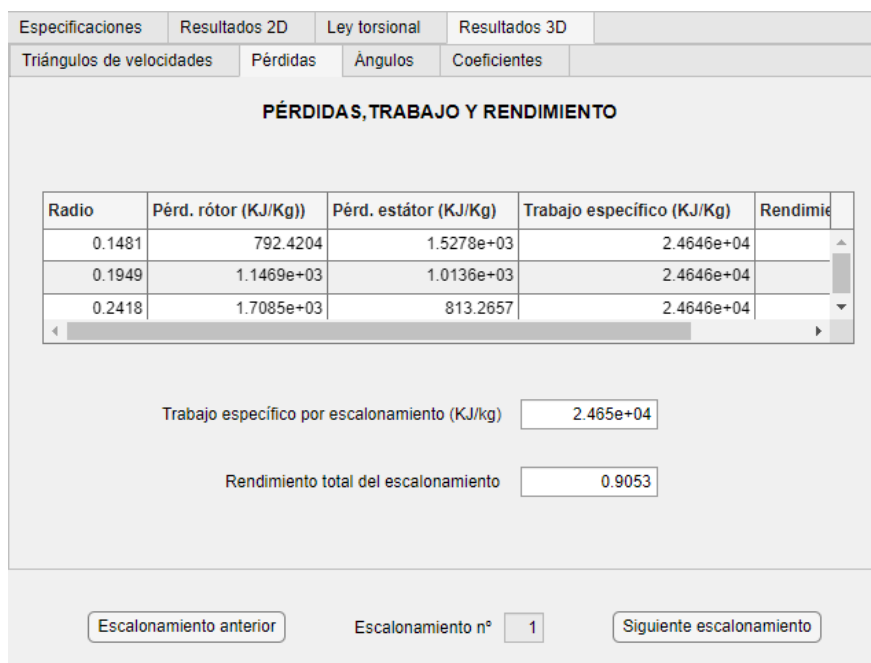


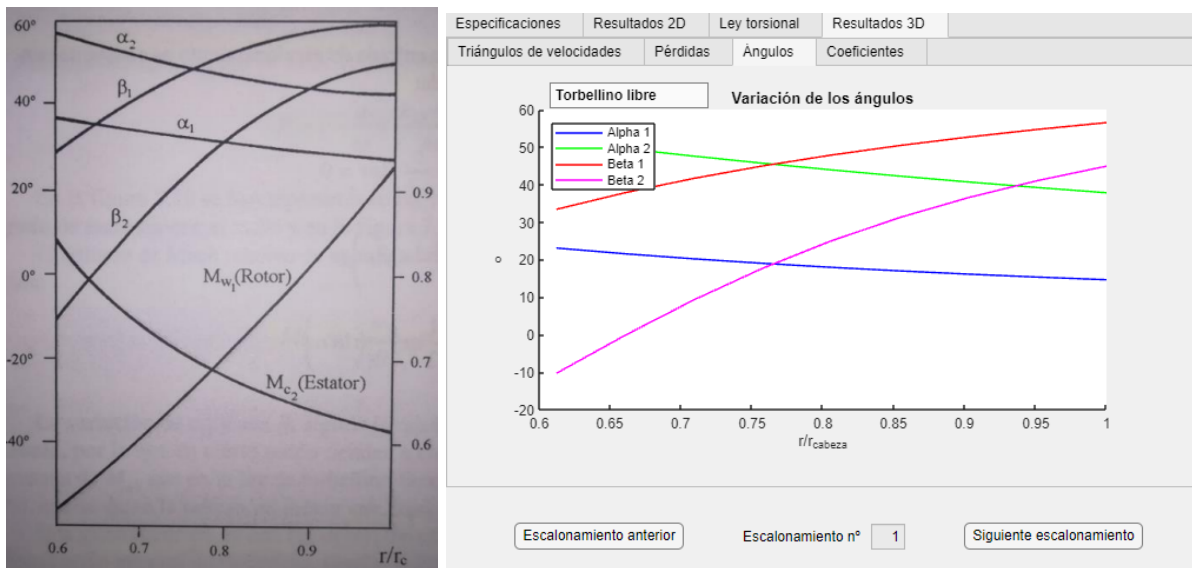
Figura 11.5: Tabla con los valores de dicha pestaña

Por ejemplo, se puede comprobar cómo, efectivamente, el trabajo específico se mantiene constante con el radio (algo que ocurre en las tres leyes torsionales). Por otro lado, los valores de las pérdidas en el rotor van aumentando conforme se avanza en el radio, mientras que para el estator ocurre el proceso contrario. En cuanto al rendimiento total a total, en este caso se tiene el mayor valor en el punto medio.

Además, en el radio medio, el valor total de las pérdidas es el mismo que en el caso bidimensional, algo que tiene sentido.

11.1.3. Variación de ángulos

Un resultado muy interesante es ver cómo varían los ángulos con el radio, por cada escalonamiento, y compararlo con la teoría de leyes torsionales.



(a) Variación teórica

(b) Resultados para el ejemplo de uso

Figura 11.6: Variación de los ángulos

Así, se puede ver cómo los ángulos relativos siguen una tendencia positiva, y con unos valores cerca del radio de cabeza parecidos a los teóricos. Lo mismo sucede con los ángulos absolutos, que siguen por el contrario una tendencia negativa con el radio.

11.1.4. Variación de coeficientes adimensionales

Un análisis similar puede realizarse con los coeficientes adimensionales.

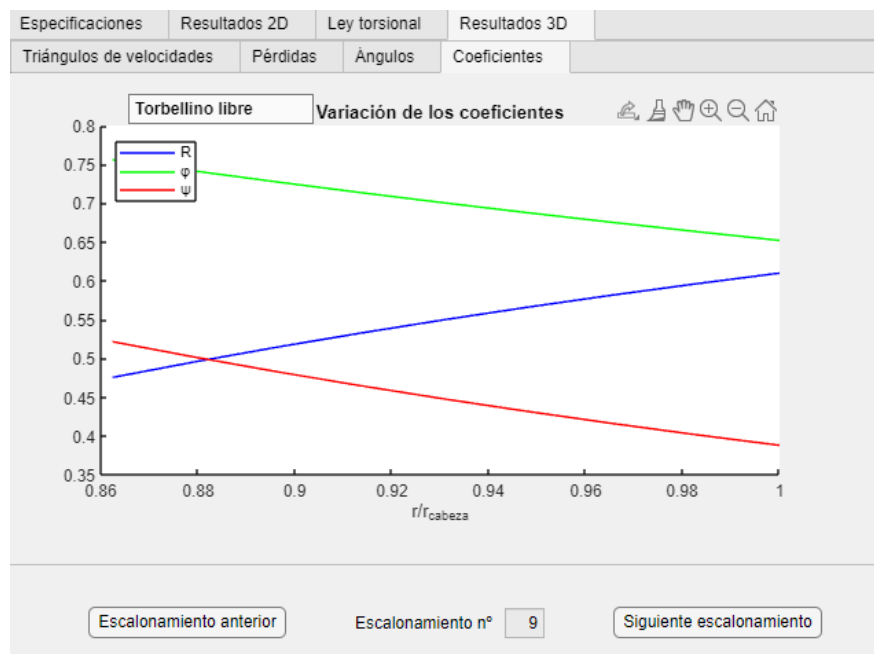


Figura 11.7: Variación de los coeficientes adimensionales con el radio

Se puede ver cómo el grado de reacción aumenta, mientras que los coeficientes de flujo y carga disminuyen. Además, los valores en el radio medio coinciden, lógicamente, con el caso bidimensional.

11.2. Ley de grado de reacción constante

A continuación, se muestran los resultados para la ley de grado de reacción constante. Los resultados son de la misma naturaleza que la ley anterior, por lo que la explicación será de mayor brevedad.

11.2.1. Triángulos de velocidades

Así pues, se muestran los resultados para los triángulos de velocidades, en el primer y último escalonamiento.

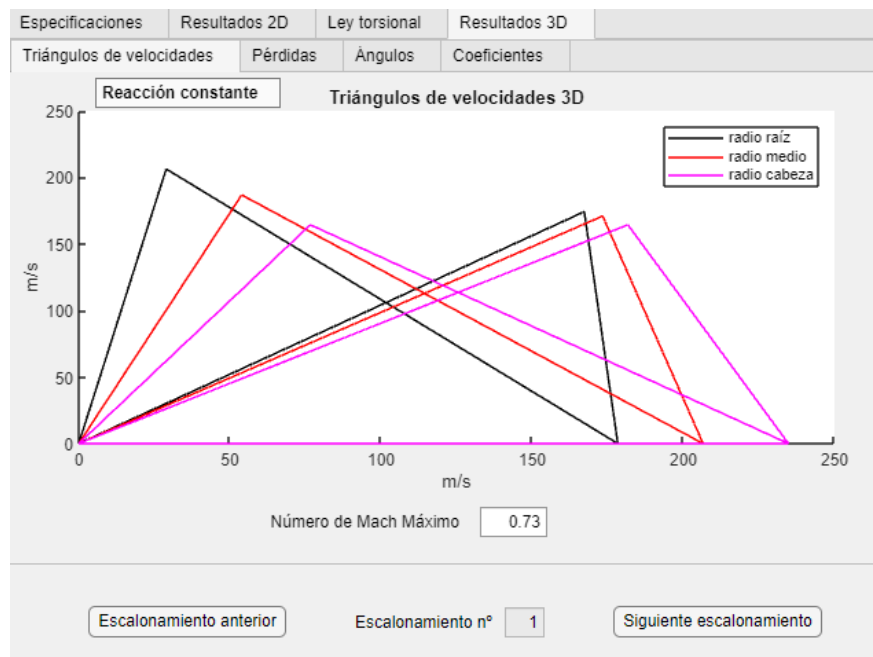


Figura 11.8: Triángulos de velocidades en el primer escalonamiento

Como es lógico, la velocidad axial no se mantiene constante.

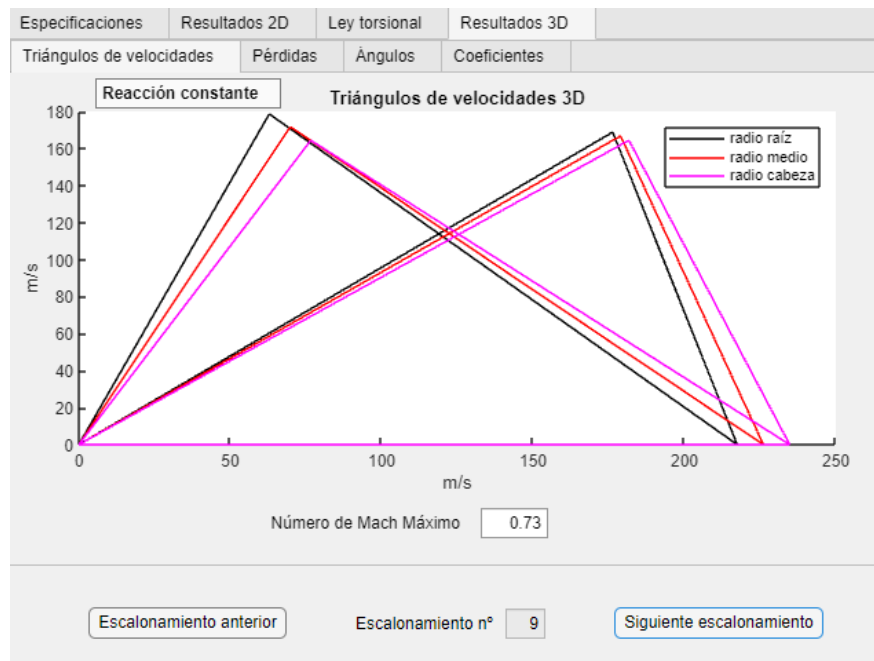


Figura 11.9: Triángulos de velocidades en el último escalonamiento

Y en este escalonamiento final, la variación del triángulo de velocidades con el radio es cada vez menor.

11.2.2. Pérdidas, trabajo y rendimiento total a total

De nuevo, se muestra la tabla correspondiente para los valores de pérdidas, trabajo y rendimiento.

PÉRDIDAS, TRABAJO Y RENDIMIENTO				
Radio	Pérd. rotor (KJ/Kg)	Pérd. estátor (KJ/Kg)	Trabajo específico (KJ/Kg)	Rendimiento
0.1481	1.2018e+03	1.2707e+03	2.4646e+04	
0.1949	1.1474e+03	1.0138e+03	2.4646e+04	
0.2418	1.1728e+03	1.1999e+03	2.4646e+04	

Trabajo específico por escalonamiento (KJ/kg)

Rendimiento total del escalonamiento

Figura 11.10: Tabla con los valores de dicha pestaña

11.2.3. Variación de ángulos

A continuación se muestran las variaciones de los ángulos con el radio.

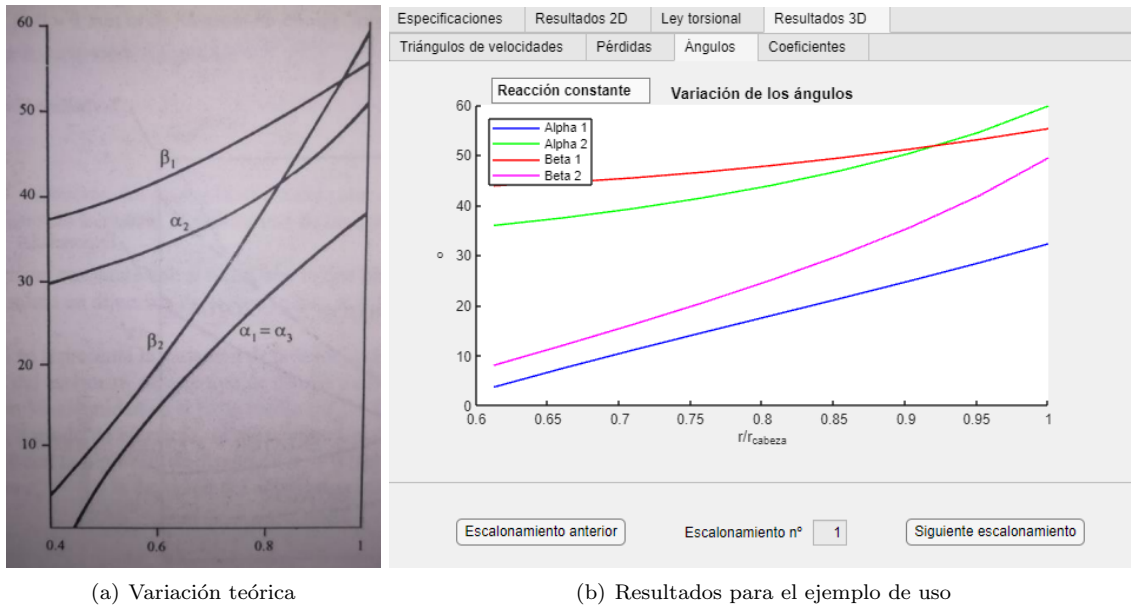


Figura 11.11: Variación de los ángulos

Se puede ver cómo este ley torsional es menos agresiva que la de torbellino libre, manteniendo unas variaciones angulares muy suaves con el radio.

11.2.4. Variación de coeficientes adimensionales

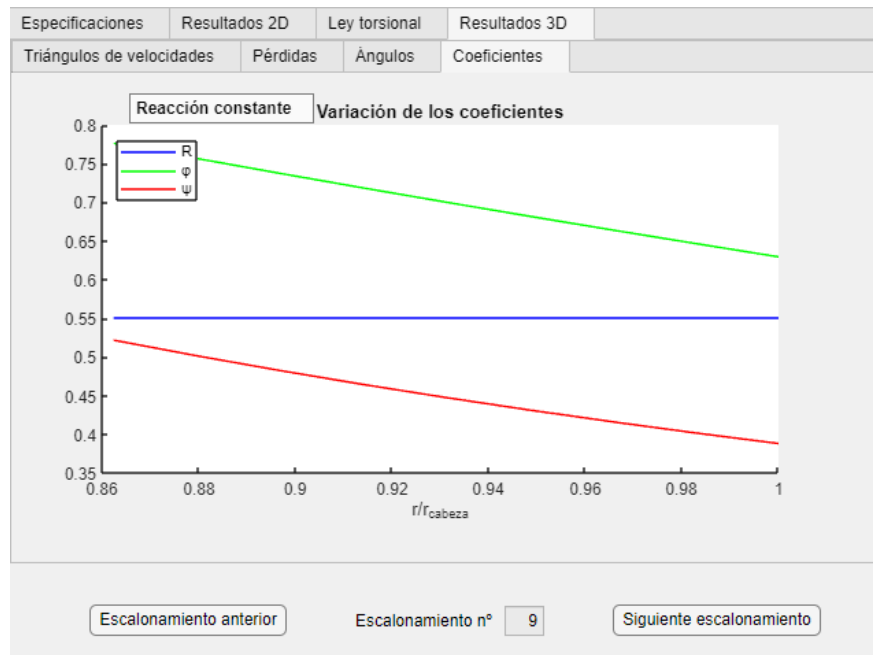


Figura 11.12: Variación de los coeficientes adimensionales con el radio

Como su nombre indica, esta ley mantiene de forma aproximada el grado de reacción como constante con el radio. En el hipotético caso de que la velocidad axial fuese constante, el grado de reacción también lo sería, y se incumpliría la ecuación de equilibrio radial.

11.2.5. Dificultades con la ley de velocidad axial

En las leyes en las que la velocidad axial no es constante, la determinación de la misma depende de la posición del radio, y de las constantes a y b , como ya se explicó en la ecuación 8.16. Esto puede provocar que, en ciertas condiciones, la velocidad axial no pueda ser calculada, ya que aparecen valores complejos, debido a la presencia de valores negativos dentro de la raíz cuadrada. Para evitar esto, la herramienta dispone de un método de prevención, en el cual se calculan las revoluciones por minuto necesarias para que no se produzca este problema, imponiendo además que el coeficiente de flujo mínimo en la cabeza del álabe del primer escalonamiento (que es el punto más desfavorable) sea mayor que 0.35.

Por ejemplo, si a las especificaciones del ejemplo de uso, se cambiase el coeficiente de flujo por un valor de 0.5, y la relación de difusión equivalente en el rotor por un valor de 1.9, ocurriría lo siguiente:

Ley de torbellino libre	Ley de grado de reacción constante	Ley exponencial
<input type="button" value="CALCULAR"/>	<input type="button" value="CALCULAR"/>	<input type="button" value="CALCULAR"/>
Coeficiente de flujo a la cabeza del primer escalonamiento para la ley de grado de reacción constante <input type="button" value="?"/>		
<input type="text" value="0.2231"/>		

Figura 11.13: Valor del coeficiente de flujo para estos cambios

Como se ve, el coeficiente de flujo es menor que 0.35, por lo que el siguiente mensaje de error es lanzado en otra ventana:

MENSAJE DE ERROR

Cuando el coeficiente de flujo en la cabeza del primer escalonamiento para la ley de $R=cte$ es menor que 0.35, el programa deja de calcular, ya que es un valor demasiado pequeño de velocidad. En su lugar, se propone reducir las revoluciones por minuto al valor que aparece en pantalla, y volver a calcular

rev/min

Figura 11.14: Mensaje de error

En el cual, se sugiere la nueva velocidad radial.

Si se aplica el cambio propuesto y de vuelve a calcular la ley de grado de reacción constante, se obtiene:

Especificaciones	Resultados 2D	Ley torsional	Resultados 3D
------------------	---------------	---------------	---------------

DISEÑO TRIDIMENSIONAL

Para resolver el caso tridimensional, debe seleccionar una de las siguientes leyes torsionales:

Ley de torbellino libre	Ley de grado de reacción constante	Ley exponencial
<input type="button" value="CALCULAR"/>	<input type="button" value="CALCULAR"/>	<input type="button" value="CALCULAR"/>

Coeficiente de flujo a la cabeza del primer escalonamiento para la ley de grado de reacción constante

Figura 11.15: Nuevo valor del coeficiente de flujo

Como la ley de grado de reacción constante es la más desfavorable de todas en este aspecto, cualquier caso de estudio que sea válido para esta ley, lo será para las demás.

11.3. Ley Exponencial

Finalmente, quedaría por determinar de manera análoga la ley exponencial.

11.3.1. Triángulos de velocidades

De nuevo, se muestra cómo varían los triángulos de velocidad con el radio.

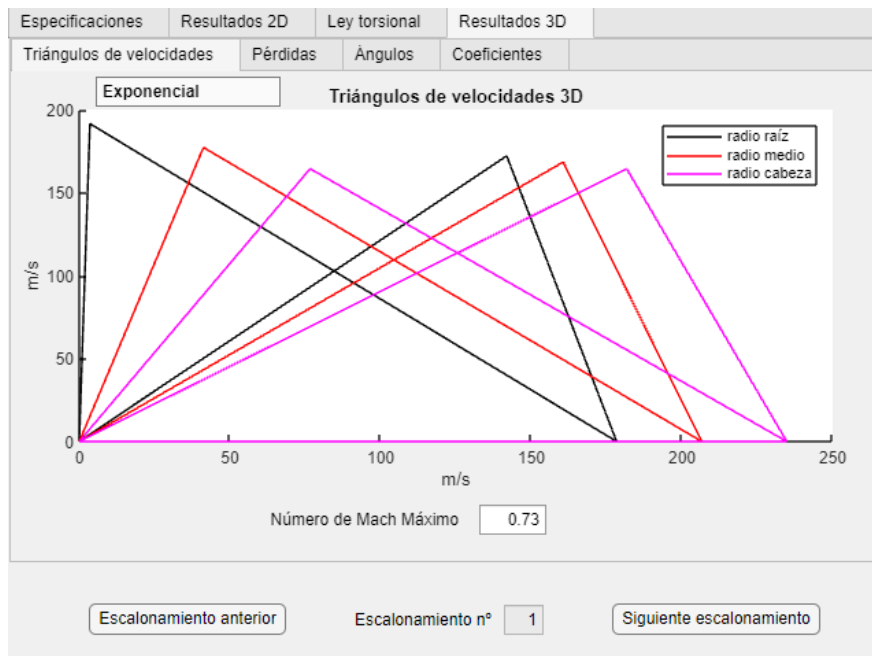


Figura 11.16: Triángulos de velocidades en el primer escalonamiento

Esta ley también es más suave que la ley de torbellino libre, pero más agresiva que la ley de grado de reacción constante.

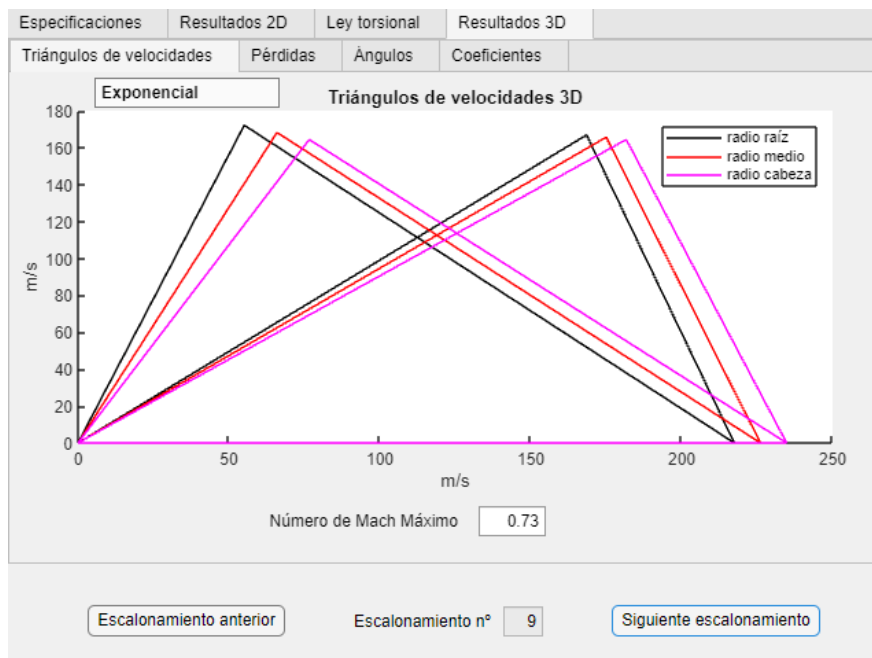


Figura 11.17: Triángulos de velocidades en el último escalonamiento

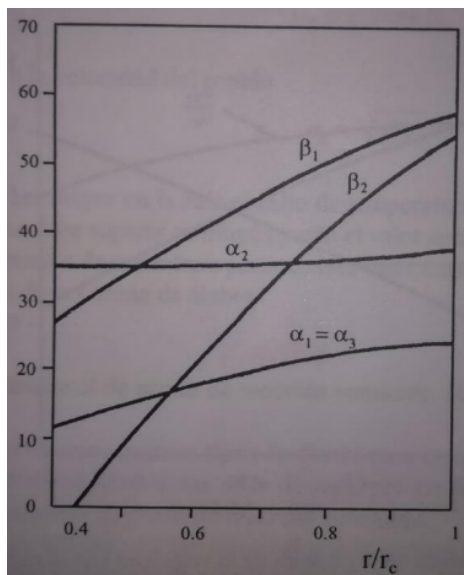
11.3.2. Pérdidas, trabajo y rendimiento total a total

Los resultados para las pérdidas, rendimiento y trabajo se muestran para la ley exponencial.

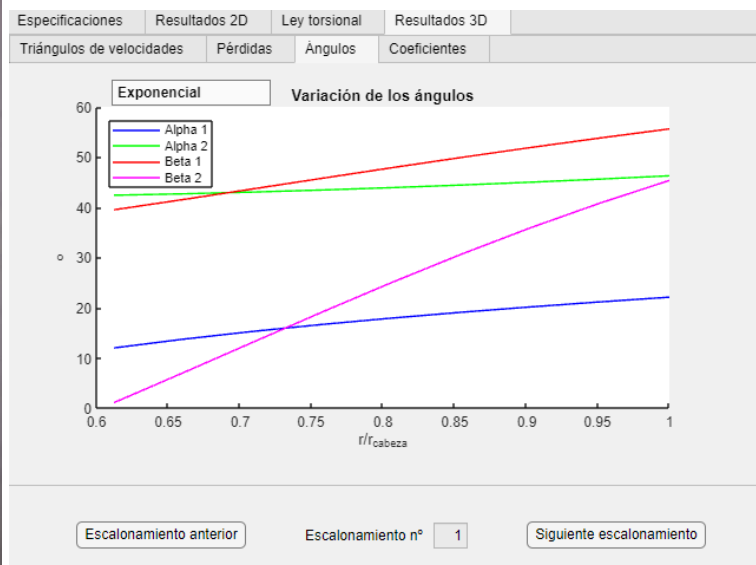


Figura 11.18: Tabla con los valores de dicha pestaña

11.3.3. Variación de ángulos



(a) Variación teórica



(b) Resultados para el ejemplo de uso

Figura 11.19: Variación de los ángulos

En esta ley, el grado de reacción también aumenta con el radio, mientras que coeficientes de flujo y carga disminuyen.

11.3.4. Variación de coeficientes adimensionales

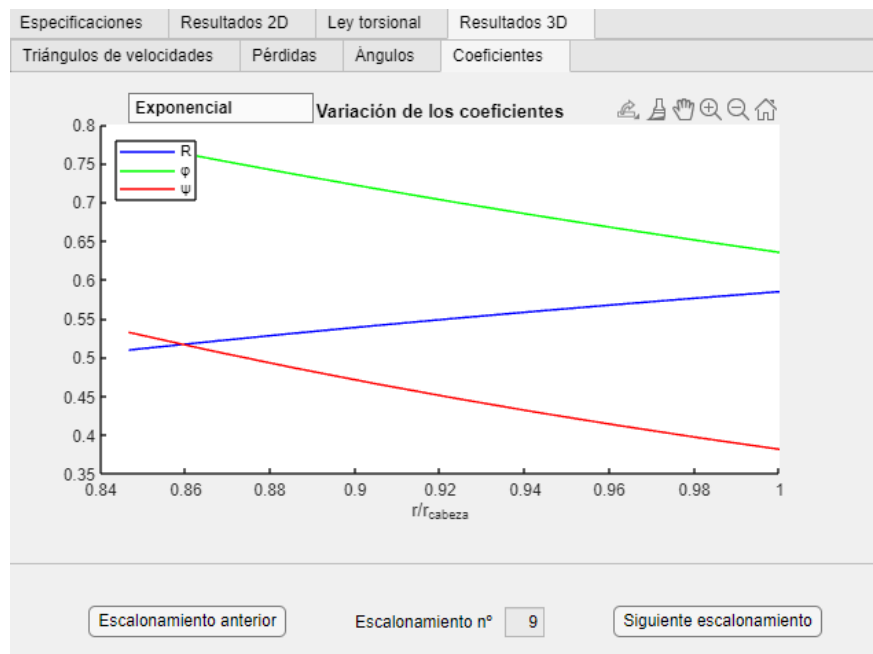


Figura 11.20: Variación de los coeficientes adimensionales con el radio

La variación de los coeficientes adimensionales es similar a la ley de grado de reacción constante, pero con variación de dicho grado de reacción.

Parte VI

Conclusiones

El objetivo primordial de este proyecto es de carácter didáctico, es decir, poder proporcionar a un usuario (normalmente un estudiante), la capacidad de aplicar de una forma visual y sencilla los conocimientos de las asignaturas *Sistemas de Propulsión Aeroespacial* y *Motores de Aeronaves*. La interfaz gráfica creada muestra una gran cantidad de resultados, tanto gráficos así como numéricos, que permiten al usuario afianzar los conocimientos sobre compresores axiales. El proyecto toma como base los fundamentos de las máquinas térmicas, aplicadas a compresores axiales, para modelar el diseño termodinámico, analizando detalladamente los procesos físicos que se llevan a cabo, y la influencia de los mismos en los resultados.

El caso bidimensional es el más completo, y supone una determinación termodinámica y geométrica completa, con unos resultados que encajan con los fundamentos teóricos expuestos en el proyecto. El caso tridimensional es algo más difuso, ya que no está completo, pero permite la visualización de los aspectos más importantes, como son los triángulos de velocidades o las prestaciones del mismo. Los resultados muestran una gran coherencia para cada ley torsional, asemejándose a la teoría.

En cuanto al software utilizado, se puede considerar una muy buena elección, tanto a nivel de cálculo, ya que como se ha explicado, *Matlab*, es ideal para el uso de vectores, matrices, y reducción de tiempos de cálculo, como a nivel de interfaz gráfica, que si bien no es muy sofisticada, permite una visualización correcta. Por ejemplo, comparado con lo que se realiza en la práctica de *Sistemas de Propulsión Aeroespacial*, donde se utiliza el programa *Engineering Equation Solver*, el cálculo es más rápido, no hay problemas de convergencia de ecuaciones como puede ocurrir de vez en cuando en el mencionado programa, y además, la interfaz gráfica supone una ventaja considerable, ya que permite mostrar tablas y gráficas, mientras que en EES han de ser creadas para su visualización. Entre algunas desventajas con respecto a EES, se puede comentar que dicho programa no requiere de iteraciones, algo que en este proyecto se ha utilizado con cierta frecuencia.

La principal finalidad del programa es el cálculo termodinámico. Para ello se utilizan tanto la hipótesis bidimensional como leyes torsionales para modelar un diseño tridimensional, cuando la hipótesis 2D no es apta. Es por tanto, fundamental, comprobar que ambos modelos muestran cierta similitud de resultados, y esto puede ser fácilmente comprobado observando las semejanzas en parámetros como rendimiento, trabajo específico por escalonamiento, y pérdidas. Si bien es cierto que a nivel local (con respecto a los álabes) las diferencias son notables en función del radio, a nivel máquina, no suponen una gran diferencia numérica.

COMPARATIVAS DE RENDIMIENTOS TOTAL A TOTAL	2D	Torbellino Libre	R. constante	Exponencial
Rendimiento total a total (primer escalonamiento)	0.9106	0.9053	0.9052	0.9082
Rendimiento total a total (último escalonamiento)	0.9113	0.9117	0.9121	0.9119

Tabla 11.1: Valores del rendimiento total a total del primer escalonamiento para cada caso de diseño.

En la tabla anterior se puede comprobar, en efecto, como todos los rendimientos totales a totales calculados (para el primer y último escalonamiento) para este ejemplo son bastante similares, con errores inferiores al 1 %.

Esta herramienta es solo una aproximación sencilla que puede ser mejorada, por ejemplo, con lo siguientes desarrollos futuros:

- Utilización de otros perfiles: En esta herramienta, se ha utilizado un perfil NACA 65, que resultan sencillos ya que se tratan de perfiles cuya línea es un arco de circunferencia, por lo que resultan sencillos de programar. Entre otros perfiles a poder utilizar, se destacan los C-4, los DCA, o los PVD, todos ellos nombrados en el apartado *Aspectos geométricos 4.2*.
- En este proyecto se ha utilizado como hipótesis radio medio constante, pero se podrían emplear otros modelos distintos, como radio medio variable, o radio externo constante.
- Diferentes fluidos de trabajo: En este proyecto se ha elegido el aire como fluido de trabajo por ser de largo el más utilizado en aplicaciones que conlleven el uso de compresores axiales. Sin embargo, podrían utilizarse otros fluidos, condiciones atmosféricas de otros cuerpos celestes, etc.
- Distinta correlación para el cálculo de pérdidas: Como se ha visto a lo largo del proyecto, la correlación

de Lieblein ha sido la empleada para el cálculo de las pérdidas energéticas, pero podrían utilizar otras, y observar los cambios encontrados.

- Mejora de la herramienta, por ejemplo, de forma que los resultados fuesen variando a medida que se modificasen, en una pantalla estática, las especificaciones, sin tener que llevar a cabo de nuevo todo el cálculo.
- Incorporación de otras leyes torsionales: En el presente trabajo, tres leyes torsionales han sido programadas (torbellino libre, grado de reacción constante, y exponencial). Estas leyes son las más comunes y proporcionan resultados diferentes que permiten ver las diferentes naturalezas de cada una de ellas. Sin embargo, existen otras leyes torsionales que pueden ser empleadas, como por ejemplo [9]:
 - Ley con gasto másico específico constante.
 - Ley con ángulo de salida del rotor constante.
- Integración del álabe: Para el diseño tridimensional, se lleva a cabo una discretización del álabe, en el caso de este proyecto, de 9 puntos (10 tramos). Sin embargo, lo verdaderamente exacto sería calcular la integral, y con ello quedaría determinado todo el dominio del álabe.

En general, se ha cumplido el objetivo de prediseño, que como se explicó en la parte introductoria de este proyecto, supone la determinación de parámetros genéricos del compresor, para conocer de manera aproximada sus prestaciones y estimar la viabilidad de un diseño completo.

Parte VII

Código utilizado

11.4. Funciones

11.4.1. f_caso_2D

```

1  %% FUNCION PARA EL CALCULO BIDIMENSIONAL
2
3  function [alpha_1,alpha_2,alpha_2_prima,alpha_3,...
4          alpha_3_prima,beta_1,beta_1_prima,beta_2,beta_2_prima,...
5          c_1,c_2,c_3,c_a,c_p,c_u1,c_u2,coef_R,...
6          curvatura_estator,curvatura_rotor,D_m,...
7          desviacion_estator,desviacion_rotor,DF_eq_E,eta_c,eta_TT,...
8          eta_politropico,gamma,H_estator,H_rotor,Lieblein_e,Lieblein_r,...
9          M_w_1,n_esc,P_01,P_02,P_03,P_1,P_2,P_3,perdidas_estator,...
10         perdidas_rotor,psi,r_c,r_c_estator,r_c_rotor,r_r_estator,r_r_rotor,...
11         rc_total_obtenida,rho_1,rho_2,rho_3,s_estator,s_rotor, ...
12         T_01,T_02,T_03,T_03ss,T_1,T_2,T_2s,T_3,T_total_final,u,w_1,w_2,...
13         W_u,zeta_a_e,zeta_a_r,zeta_e,zeta_p_e,zeta_p_r,zeta_r,...
14         zeta_s_e,zeta_s_r]...
15 = f_caso_2D(T_01_primer,P_01_primer,sigma,n,m_dot,DF_eq_R,R,phi,rc_total)
16
17 % Coeficientes para el calor especifico
18
19 coef_alpha=3.653;
20 coef_beta=-1.337e-3;
21 coef_gamma=3.294e-6;
22 coef_delta=-1.913e-9;
23 coef_epsilon=0.2763e-12;
24
25 rc_ideal = 1.25; % Relacion de compresion objetivo o ideal
26
27 %% ESPECIFICACIONES
28
29 coef_R=287; % dependientes del fluido. En este caso aire 287
30 gamma_primer=1.4; % se supone este valor inicial de gamma 1.4
31
32 M_w_1 = 0.85; % se comienza a iterar con este valor, que disminuir .
33
34 % calculo del numero de escalonamientos
35
36 n_esc_aprox = log(rc_total) / log(rc_ideal);
37 n_esc = ceil(n_esc_aprox); % redondea al entero superior
38
39 % DEFINICION DE VECTORES
40
41 P_01=zeros(1,n_esc);
42 P_1=zeros(1,n_esc);
43 P_02=zeros(1,n_esc);
44 P_2=zeros(1,n_esc);
45 P_03=zeros(1,n_esc);
46 P_3=zeros(1,n_esc);
47 T_01=zeros(1,n_esc);
48 T_1=zeros(1,n_esc);
49 T_02=zeros(1,n_esc);
50 T_2=zeros(1,n_esc);
51 T_03=zeros(1,n_esc);

```



```

52 T_3=zeros(1,n_esc);
53 T_2s=zeros(1,n_esc);
54
55 rho_1=zeros(1,n_esc);
56 rho_2=zeros(1,n_esc);
57 rho_3=zeros(1,n_esc);
58 H_rotor=zeros(1,n_esc);
59 H_estator=zeros(1,n_esc);
60 s_rotor=zeros(1,n_esc);
61 s_estator=zeros(1,n_esc);
62 eta_TT=zeros(1,n_esc);
63
64 r_c=zeros(1,n_esc);
65 T_03ss=zeros(1,n_esc);
66 c_p=zeros(1,n_esc);
67
68 r_r_rotor=zeros(1,n_esc);
69 r_c_rotor=zeros(1,n_esc);
70 r_r_estator=zeros(1,n_esc);
71 r_c_estator=zeros(1,n_esc);
72
73 rc_total_obtenida=rc_ideal^n_esc; % Para comenzar la iteracion
74
75 while rc_total_obtenida / rc_total > abs(1.01)
76
77 %% Resolucion del triangulo de velocidades, que sera valido para todos los
78 %% escalonamientos
79
80 [alpha_1,alpha_2,beta_1,beta_2,c_a,u,c_1,c_2,w_1,w_2,psi,T_1_primer]= ...
81 f_triangulo_de_velocidades ...
82 (R,coef_R,gamma_primer,phi,DF_eq_R,sigma,M_w_1,T_01_primer);
83
84 c_3 = c_1;
85
86 DF_eq_E=cosd(alpha_1)/cosd(alpha_2)* ...
87 (1.12 + 0.61/sigma * cosd(alpha_2)^2 * (tand(alpha_2) - tand(alpha_1)));
88
89 % Si no se cumplen las condiciones, el c digo se detendr
90 if DF_eq_E > 1.95 || DF_eq_R > 1.95 || sigma<1 || ...
91     sigma>1.4 || R<0.4 || R>0.6 || phi<0.4 || phi>0.7
92
93     error='Ha habido un error'
94     return
95 end
96
97 % Calculo del calor especifico en el punto 1 del primer escalonamiento
98
99 c_p_primer=coef_R/1000*(coef_alpha+coef_beta*T_1_primer+coef_gamma* ...
100 T_1_primer^2+coef_delta*T_1_primer^3+coef_epsilon*T_1_primer^4);
101 gamma_primer=c_p_primer/(c_p_primer-coef_R/1000);
102
103 % Llamada a la funcion de calculo de geometria de entrada
104
105 [D_m,P_1_primer,rho_1_primer,H_rotor_primer] = f_geometria_entrada ...
106 (u,n,gamma_primer,coef_R,T_1_primer,T_01_primer,P_01_primer,m_dot,c_a);

```

```

107
108 % Llamada a la funcion de calculo de perdidas
109
110 [beta_m,alpha_m,Lieblein_r,Lieblein_e,zeta_p_r,zeta_p_e,zeta_a_r, ...
111 zeta_a_e,zeta_s_r,zeta_s_e,zeta_e,zeta_r,C_l_r,C_l_e,perdidas_rotor, ...
112 perdidas_estator] = ...
113 f_perdidas(sigma,beta_1,beta_2,alpha_1,alpha_2,DF_eq_R,DF_eq_E,w_1,c_2);
114
115 % Llamada a la funcion para calculo del trabajo y rendimiento
116
117 [W_u,eta_c,c_u2,c_u1] = ...
118 f_trabajo(c_2,c_1,w_1,alpha_1,alpha_2,u,zeta_r,zeta_e);
119
120 for i=1:n_esc
121     if i==1 % Primer escalonamiento
122
123         % Calculo de las variables restantes
124
125         T_01(i)=T_01_primer;
126         P_01(i)=P_01_primer;
127         gamma(i)=gamma_primer;
128         c_p(i)=c_p_primer;
129
130         T_02_primer = T_01_primer + W_u / (1000 * c_p(i));
131         T_03_primer = T_02_primer;
132         T_2_primer = (W_u + c_1^2/2 - c_2^2/2)/(c_p(i)*1000) + T_1_primer;
133         T_3_primer = (c_2^2/2 - c_3^2/2) / (c_p(i) * 1000) + T_2_primer;
134         eta_TT_primer = 1 - (perdidas_rotor*T_3_primer/T_2_primer ...
135         + perdidas_estator) / W_u;
136         T_03ss(i) = T_01(i) + ...
137         (eta_TT_primer * W_u * (gamma(i) - 1))/(coef_R * gamma(i));
138         r_c(i) = (T_03ss(i)/T_01(i))^(gamma(i) / (gamma(i)-1));
139         P_03_primer = P_01_primer* r_c(i);
140         T_2s_primer = T_2_primer - perdidas_rotor / (c_p(i) * 1000) ;
141         P_2_primer = P_1_primer*(T_2s_primer/T_1_primer)^ ...
142         (gamma(i)/(gamma(i)-1));
143         P_02_primer = ...
144         P_2_primer / (T_2_primer/T_02_primer)^(gamma(i)/(gamma(i)-1));
145         P_3_primer = ...
146         P_03_primer*(T_3_primer/T_03_primer)^(gamma(i)/(gamma(i)-1));
147         rho_2_primer = P_2_primer / (coef_R / 1000 * T_2_primer);
148         rho_3_primer = P_3_primer / (coef_R / 1000 * T_3_primer);
149         s_rotor_primer = H_rotor_primer * 0.4;
150         H_estator_primer = m_dot / (pi * c_a * D_m * rho_2_primer);
151         s_estator_primer = H_estator_primer * 0.4;
152
153         % Definicion de los vectores
154
155         P_01(i)=P_01_primer;
156         P_1(i)=P_1_primer;
157         T_01(i)=T_01_primer;
158         T_1(i)=T_1_primer;
159         P_02(i)=P_02_primer;
160         P_2(i)=P_2_primer;
161         T_02(i)=T_02_primer;

```

```

162     T_2(i)=T_2_primer;
163     T_2s(i)=T_2s_primer;
164     T_3(i)=T_3_primer;
165     eta_TT(i)=eta_TT_primer;
166     P_03(i)=P_03_primer;
167     P_3(i)=P_3_primer;
168     T_03(i)=T_03_primer;
169     T_3(i)=T_3_primer;
170     rho_1(i)=rho_1_primer;
171     rho_2(i)=rho_2_primer;
172     rho_3(i)=rho_3_primer;
173     H_rotor(i)=H_rotor_primer;
174     H_estator(i)=H_estator_primer;
175     s_rotor(i)=s_rotor_primer;
176     s_estator(i)=s_estator_primer;
177
178     % Condiciones de contorno para el siguiente escalonamiento
179
180     P_01(i+1)=P_03(i);
181     T_01(i+1)=T_03(i);
182     P_1(i+1)=P_3(i);
183     T_1(i+1)=T_3(i);
184
185     else % Resto de escalonamientos
186
187     c_p(i)=coef_R/1000*(coef_alpha+coef_beta*T_1(i)+ ...
188     coef_gamma*T_1(i)^2+coef_delta*T_1(i)^3+coef_epsilon*T_1(i)^4);
189     gamma(i)=c_p(i)/(c_p(i)-coef_R/1000);
190
191     T_02(i) = T_01(i) + W_u / (1000 * c_p(i));
192     T_03(i) = T_02(i); %
193     T_2(i) = (W_u + c_1^2/2 - c_2^2/2)/(c_p(i)*1000) + T_1(i);
194     T_3(i) = (c_2^2/2 - c_3^2/2) / (c_p(i) * 1000) + T_2(i);
195     eta_TT(i) = 1 - (perdidas_rotor*T_3(i)/T_2(i) + ...
196     perdidas_estator) / W_u;
197     T_03ss(i) = ...
198     T_01(i) + (eta_TT(i) * W_u * (gamma(i) - 1))/(coef_R * gamma(i));
199     r_c(i) = (T_03ss(i)/T_01(i))^(gamma(i) / (gamma(i)-1));
200     P_03(i) = P_01(i) * r_c(i);
201     T_2s(i) = T_2(i) - perdidas_rotor / (c_p(i) * 1000);
202     P_2(i) = P_1(i)*(T_2s(i)/T_1(i))^(gamma(i)/(gamma(i)-1));
203     P_02(i) = P_2(i) / (T_2(i)/T_02(i))^(gamma(i)/(gamma(i)-1));
204     P_3(i) = P_03(i)*(T_3(i)/T_03(i))^(gamma(i)/(gamma(i)-1));
205
206     % calculo de densidades, alturas y pasos
207
208     rho_1(i)=P_1(i)/(coef_R/1000 * T_1(i));
209     rho_2(i)=P_2(i)/(coef_R/1000 * T_2(i));
210     rho_3(i)=P_3(i)/(coef_R/1000 * T_3(i));
211
212     H_rotor(i) = m_dot / (pi * c_a * D_m * rho_1(i));
213     H_estator(i) = m_dot / (pi * c_a * D_m * rho_2(i));
214
215     s_rotor(i) = 0.4 * H_rotor(i);
216     s_estator(i) = 0.4 * H_estator(i);

```

```

217
218     % se imponen las condiciones de contorno para resolver los
219     % siguientes escalonamientos
220
221     if i<n_esc
222
223         P_01(i+1)=P_03(i);
224         T_01(i+1)=T_03(i);
225         P_1(i+1)=P_3(i);
226         T_1(i+1)=T_3(i);
227
228     end
229
230 end
231
232 end
233
234 % Calculo de relacion de compresion total
235
236 rc_total_obtenida=1;
237 j=1;
238 while j<n_esc+1
239
240     rc_total_obtenida=r_c(j)*rc_total_obtenida
241     j=j+1;
242
243 end
244
245 % Reduccion del Mach en cada iteracion
246
247 M_w_1 = M_w_1 - 0.0025;
248
249 end
250
251 % Radios de raiz y cabeza
252
253 for k=1:n_esc
254     r_r_rotor(k) = D_m/2 - H_rotor(k)/2;
255     r_r_estator(k) = D_m/2 - H_estator(k)/2;
256     r_c_rotor(k) = D_m/2 + H_rotor(k)/2;
257     r_c_estator(k) = D_m/2 + H_estator(k)/2;
258 end
259
260 %% Geometria de los alabes
261
262 % ROTOR
263
264 beta_1_prima=beta_1; % Se considera nula la incidencia
265 beta_2_prima=0;
266 desviacion_rotor=0;
267 desviacion_rotor_it=1;
268 while desviacion_rotor / desviacion_rotor_it < abs(0.999)
269
270 desviacion_rotor_it = beta_2-beta_2_prima;
271 desviacion_rotor = ...

```

```

272 (0.23 + beta_2/200) * (beta_1_prima-beta_2_prima) * (1/sigma)^0.5;
273
274 beta_2_prima = beta_2_prima + 0.01;
275
276 end
277
278 curvatura_rotor = beta_1_prima - beta_2_prima
279
280 % ESTATOR
281
282 alpha_3=alpha_1;
283
284 alpha_2_prima=alpha_2; % Se considera nula la incidencia
285 alpha_3_prima=0;
286 desviacion_estator=0;
287 desviacion_estator_it=1;
288 while desviacion_estator / desviacion_estator_it < abs(0.999)
289
290 desviacion_estator_it = alpha_3-alpha_3_prima;
291 desviacion_estator = ...
292 (0.23 + alpha_3/200) * (alpha_2_prima-alpha_3_prima) * (1/sigma)^0.5;
293
294 alpha_3_prima = alpha_3_prima + 0.01;
295
296 end
297
298 curvatura_estator = alpha_2_prima - alpha_3_prima
299
300 % comprobacion proceso politropico (no usado)
301
302 %eta_politropico = 0.898;
303 eta_politropico=0.875;
304 T_total_final=T_01(1) * ...
305 rc_total_obtenida^((gamma(n_esc)-1)/(gamma(n_esc)*eta_politropico));

```

11.4.2. f_triangulo_de_velocidades

```

1 %% FUNCION PARA CALCULO DE TRIANGULOS DE VELOCIDADES
2
3 function ...
4 [alpha_1,alpha_2,beta_1,beta_2,c_a,u,c_1,c_2,w_1,w_2,psi,T_1_primer]= ...
5 f_triangulo_de_velocidades ...
6 (R,coef_R,gamma_primer,phi,DF_eq_R,sigma,M_w_1,T_01_primer)
7
8 %% Angulos geometricos
9
10 a=0;
11 beta_1_it = 0;
12
13 while a<1
14
15 beta_2_it = atand(R*2/phi - tand(beta_1_it));
16
17 DF_eq_it = cosd(beta_2_it)/cosd(beta_1_it)*(1.12 + 0.61/sigma * ...

```

```

18 cosd(beta_1_it)^2 * (tand(beta_1_it) - tand(beta_2_it));
19
20 if DF_eq_it / DF_eq_R > abs(0.9999)
21     a=1;
22     DF_eq_R = DF_eq_it;
23     beta_1 = beta_1_it;
24     beta_2 = beta_2_it;
25     break
26 else
27     beta_1_it = beta_1_it + 0.001;
28 end
29 end
30
31 alpha_1 = atand(tand(beta_2) - 2/phi * (R - 1/2));
32 alpha_2 = atand(2/phi * (1 - R) - tand(alpha_1));
33
34 T_1_primer = T_01_primer / (1 + M_w_1^2 * ...
35 (gamma_primer-1)/2 *(cosd(beta_1)/cosd(alpha_1))^2);
36
37 %%% Velocidades
38
39
40 w_1 = M_w_1 * (sqrt(gamma_primer * coef_R * T_1_primer));
41
42
43 psi = phi * (tand(alpha_2) - tand(alpha_1));
44
45 c_a=w_1*cosd(beta_1);
46
47 w_2 = c_a / cosd(beta_2);
48 c_1 = c_a / cosd(alpha_1);
49 c_2 = c_a / cosd(alpha_2);
50 u = w_2 * sind(beta_2) + c_2 * sind(alpha_2);

```

11.4.3. f_geometria_entrada

```

1 %%% FUNCION PARA EL CALCULO DE LA GEOMETRIA DE ENTRADA
2
3 function[D_m,P_1_primer,rho_1_primer,H_rotor_primer] = ...
4 f_geometria_entrada ...
5 (u,n,gamma_primer,coef_R,T_1_primer,T_01_primer,P_01_primer,m_dot,c_a)
6
7 % Esta funcion calcula la geometria de entrada del primer escalonamiento
8
9 D_m = u * 60/ (pi * n);
10
11 P_1_primer = ...
12 P_01_primer*(T_1_primer/T_01_primer)^(gamma_primer/(gamma_primer-1));
13 rho_1_primer = P_1_primer / (coef_R/1000 * T_1_primer) ;
14
15 H_rotor_primer = m_dot / (pi * D_m * c_a * rho_1_primer);

```

11.4.4. f_perdidas

```

1 %% FUNCION PARA EL CALCULO DE PERDIDAS
2
3 function[beta_m,alpha_m,Lieblein_r,Lieblein_e,zeta_p_r,zeta_p_e, ...
4     zeta_a_r,zeta_a_e,zeta_s_r,zeta_s_e,zeta_e,zeta_r,C_l_r,C_l_e, ...
5     perdidas_rotor,perdidas_estator] = f_perdidas(sigma,beta_1,beta_2, ...
6     alpha_1,alpha_2,DF_eq_R,DF_eq_E,w_1,c_2)
7
8 %esta funcion calcula las perdidas, que se mantendran en todos los
9 %escalonamientos
10
11 beta_m = atand((tand(beta_1) + tand(beta_2))/2);
12 alpha_m= atand((tand(alpha_1) + tand(alpha_2))/2);
13
14 Lieblein_r = 0.004 / (1 - 1.17 * log(DF_eq_R));
15 Lieblein_e = 0.004 / (1 - 1.17 * log(DF_eq_E));
16
17 zeta_p_r = ...
18 2 * (Lieblein_r * sigma * ((cosd(beta_1))^2 / (cosd(beta_2))^3));
19 zeta_p_e = ...
20 2 * (Lieblein_e * sigma * ((cosd(alpha_2))^2 / (cosd(alpha_1))^3));
21
22 zeta_a_r = ...
23 0.02 * 0.4 * sigma * (cosd(beta_1))^2 / (cosd(beta_m))^3;
24 % se ha elegido la relacion paso/altura del alabe como 0.4
25 zeta_a_e = 0.02 * 0.4 * sigma * (cosd(alpha_2))^2 / (cosd(alpha_m))^3;
26
27 C_l_r = 2 * 1/sigma * (tand(beta_1) - tand(beta_2) ) * cosd(beta_m);
28 C_l_e = 2 * 1/sigma * (tand(alpha_2) - tand(alpha_1) ) * cosd(alpha_m);
29
30 zeta_s_r = 0.018 * C_l_r^2 * sigma * (cosd(beta_1))^2 / (cosd(beta_m))^3;
31 zeta_s_e = 0.018 * C_l_e^2 * sigma * (cosd(alpha_2))^2 / (cosd(alpha_m))^3;
32
33 zeta_r = zeta_p_r + zeta_a_r + zeta_s_r;
34 zeta_e = zeta_p_e + zeta_a_e + zeta_s_e;
35
36 perdidas_rotor = zeta_r * w_1^2 / 2;
37 perdidas_estator = zeta_e * c_2^2 / 2;

```

11.4.5. f_trabajo

```

1 %% FUNCION PARA EL CALCULO DEL TRABAJO
2
3 function[W_u,eta_c,c_u2,c_u1] = ...
4 f_trabajo(c_2,c_1,w_1,alpha_1,alpha_2,u,zeta_r,zeta_e)
5
6 %%%% TRABAJO
7
8 c_u2 = c_2 * sind(alpha_2);
9 c_u1 = c_1 * sind(alpha_1);
10
11 W_u = u * (c_u2 - c_u1);
12

```

```
13 eta_c = 1 - (zeta_r * w_1^2 / 2 + zeta_e * c_2^2 / 2) / W_u;
```

11.4.6. f_torbellino_libre

```
1 % LEY DE TORBELLINO LIBRE
2
3 function[b_1,b_2,c_a_3D,W_u_3D,u_3D,c_u1_3D,c_u2_3D, c_1_3D,c_2_3D, ...
4         w_u1_3D,w_u2_3D,beta_1_3D,beta_2_3D, alpha_1_3D,alpha_2_3D,R_3D, ...
5         T_1_3D,M_c_3D,M_r_3D,DF_eq_R_3D,DF_eq_E_3D, ...
6         zeta_p_R_3D,zeta_p_E_3D, perdidas_rotor_3D,perdidas_estator_3D, ...
7         perdidas_3D,phi_3D,psi_3D,W_u_3D_is,eta_TT_3D,w_1_3D,w_2_3D] ...
8 = f_torbellino_libre(c_u1,c_u2,D_m,c_a,W_u,n_esc,u,r_r_rotor,r_c_rotor, ...
9 T_01,c_p,gamma,coef_R,sigma,zeta_s_r,zeta_a_r,zeta_s_e,zeta_a_e,puntos)
10
11 % Calculos de las constantes
12
13 b_1 = c_u1 * (D_m/2);
14 b_2 = c_u2 * (D_m/2);
15
16 % Esta ley conserva velocidad axial y trabajo especifico con el radio
17
18 c_a_3D = c_a;
19 W_u_3D = W_u;
20
21 for x=1:n_esc
22
23     radio=r_r_rotor(x);
24
25     for y=1:puntos
26         u_3D(y,x)=u/(D_m/2)*radio;
27         c_u1_3D(y,x) = b_1/radio;
28         c_u2_3D(y,x) = b_2/radio;
29         c_1_3D(y,x) = sqrt(c_a_3D^2 + c_u1_3D(y,x)^2);
30         c_2_3D(y,x) = sqrt(c_a_3D^2 + c_u2_3D(y,x)^2);
31         w_u1_3D(y,x) = u_3D(y,x) - c_u1_3D(y,x);
32         w_u2_3D(y,x) = u_3D(y,x) - c_u2_3D(y,x);
33         w_1_3D(y,x) = sqrt(c_a_3D^2 + w_u1_3D(y,x)^2);
34         w_2_3D(y,x) = sqrt(c_a_3D^2 + w_u2_3D(y,x)^2);
35         beta_1_3D(y,x) = atand(w_u1_3D(y,x)/c_a_3D);
36         beta_2_3D(y,x) = atand(w_u2_3D(y,x)/c_a_3D);
37         alpha_1_3D(y,x) = atand(c_u1_3D(y,x)/c_a_3D);
38         alpha_2_3D(y,x) = atand(c_u2_3D(y,x)/c_a_3D);
39         R_3D(y,x) = (w_u1_3D(y,x) + w_u2_3D(y,x))/(2*u_3D(y,x));
40         phi_3D(y,x) = c_a./u_3D(y,x);
41         psi_3D(y,x)= (c_u2_3D(y,x)-c_u1_3D(y,x))./u_3D(y,x);
42
43         % Calculo del n mero de Mach maximo
44
45         T_1_3D(y,x) = T_01(x) - c_1_3D(y,x)^2/(2000*c_p(x));
46         M_c_3D(y,x) = c_1_3D(y,x)/sqrt(gamma(x)*coef_R*T_1_3D(y,x));
47         M_r_3D(y,x) = w_1_3D(y,x)/sqrt(gamma(x)*coef_R*T_1_3D(y,x));
48
49         % Calculo de la relaciones de difusion
50
```



```

51     DF_eq_R_3D(y,x) = ...
52     cosd(beta_2_3D(y,x))/cosd(beta_1_3D(y,x))* ...
53     (1.12 + 0.61/sigma * (cosd(beta_1_3D(y,x)))^2 * ...
54     (tand(beta_1_3D(y,x))-tand(beta_2_3D(y,x)))));
55     DF_eq_E_3D(y,x) = ...
56     cosd(alpha_1_3D(y,x))/cosd(alpha_2_3D(y,x))* ...
57     (1.12 + 0.61/sigma * (cosd(alpha_2_3D(y,x)))^2 * ...
58     (tand(alpha_2_3D(y,x))-tand(alpha_1_3D(y,x)))));
59
60     % Calculo de las perdidas
61
62     zeta_p_R_3D(y,x) = 2*(0.004/(1-1.17*log(DF_eq_R_3D(y,x))))*...
63     sigma*(cosd(beta_1_3D(y,x)))^2/(cosd(beta_2_3D(y,x)))^3;
64     zeta_p_E_3D(y,x) = 2*(0.004/(1-1.17*log(DF_eq_E_3D(y,x))))*...
65     sigma*(cosd(alpha_2_3D(y,x)))^2/(cosd(alpha_1_3D(y,x)))^3;
66
67     zeta_R_3D(y,x)=(zeta_p_R_3D(y,x) + zeta_s_r + zeta_a_r);
68     zeta_E_3D(y,x)=(zeta_p_E_3D(y,x) + zeta_s_e + zeta_a_e);
69
70     perdidas_rotor_3D(y,x) = zeta_R_3D(y,x)*w_1_3D(y,x)^2/2;
71     perdidas_estator_3D(y,x) = zeta_E_3D(y,x)*c_2_3D(y,x)^2/2;
72
73     perdidas_3D(y,x) = ...
74     perdidas_rotor_3D(y,x) + perdidas_estator_3D(y,x);
75
76     W_u_3D_is(y,x) = W_u_3D - perdidas_3D(y,x);
77     eta_TT_3D(y,x) = W_u_3D_is(y,x)/W_u_3D;
78
79     radio = radio + (r_c_rotor(x)-r_r_rotor(x))/(puntos-1);
80
81     end
82 end

```

11.4.7. f_exponencial

```

1  % LEY EXPONENCIAL
2
3  function[a,b,c_a1_3D,c_a2_3D,W_u_3D,u_3D,c_u1_3D,c_u2_3D, c_1_3D,c_2_3D,...
4     w_u1_3D,w_u2_3D,beta_1_3D,beta_2_3D, alpha_1_3D,alpha_2_3D,R_3D, ...
5     T_1_3D,M_c_3D,M_r_3D,DF_eq_R_3D,DF_eq_E_3D, ...
6     zeta_p_R_3D,zeta_p_E_3D, perdidas_rotor_3D,perdidas_estator_3D, ...
7     perdidas_3D,phi_3D,psi_3D,W_u_3D_is,eta_TT_3D,w_1_3D,w_2_3D] ...
8  = f_exponencial(c_u1,c_u2,D_m,c_a,W_u,n_esc,u,r_r_rotor,r_c_rotor, ...
9  T_01,c_p,gamma,coef_R,sigma,zeta_s_r,zeta_a_r,zeta_s_e,zeta_a_e,puntos)
10
11  W_u_3D = W_u;
12
13  b=1;
14  b_it=0;
15  a=0;
16  while b_it/b < abs(0.999)
17     b = D_m/2 * (c_u2-a);
18     b_it = D_m/2 * (a-c_u1);
19     a=a+0.01;

```

```

20 end
21
22 cte_1 = c_a^2 + 2*(a^2*log(D_m/2) + a*b/(D_m/2));
23 cte_2 = c_a^2 + 2*(a^2*log(D_m/2) - a*b/(D_m/2));
24
25 for x=1:n_esc
26     radio=r_r_rotor(x);
27
28     for y=1:puntos
29
30         c_a1_3D(y,x) = sqrt(cte_1 - 2*(a^2*log(radio) + a*b/(radio)));
31         c_a2_3D(y,x) = sqrt(cte_2 - 2*(a^2*log(radio) - a*b/(radio)));
32         u_3D(y,x)=u/(D_m/2)*radio;
33         c_u1_3D(y,x) = a - b/radio;
34         c_u2_3D(y,x) = a + b/radio;
35         c_1_3D(y,x) = sqrt(c_a1_3D(y,x)^2 + c_u1_3D(y,x).^2);
36         c_2_3D(y,x) = sqrt(c_a2_3D(y,x)^2 + c_u2_3D(y,x).^2);
37         w_u1_3D(y,x) = u_3D(y,x) - c_u1_3D(y,x);
38         w_u2_3D(y,x) = u_3D(y,x) - c_u2_3D(y,x);
39         w_1_3D(y,x) = sqrt(c_a1_3D(y,x)^2 + w_u1_3D(y,x)^2);
40         w_2_3D(y,x) = sqrt(c_a2_3D(y,x)^2 + w_u2_3D(y,x)^2);
41         beta_1_3D(y,x) = atand(w_u1_3D(y,x)/c_a1_3D(y,x));
42         beta_2_3D(y,x) = atand(w_u2_3D(y,x)/c_a2_3D(y,x));
43         alpha_1_3D(y,x) = atand(c_u1_3D(y,x)/c_a1_3D(y,x));
44         alpha_2_3D(y,x) = atand(c_u2_3D(y,x)/c_a2_3D(y,x));
45         R_3D(y,x) = (w_u1_3D(y,x) + w_u2_3D(y,x))/(2*u_3D(y,x));
46         phi_3D(y,x) = c_a1_3D(y,x)/u_3D(y,x);
47         psi_3D(y,x)= (c_u2_3D(y,x)-c_u1_3D(y,x))/u_3D(y,x);
48
49         % Calculo del n mero de Mach maximo
50
51         T_1_3D(y,x) = T_01(x) - c_1_3D(y,x)^2/(2000*c_p(x));
52         M_c_3D(y,x) = c_1_3D(y,x)/sqrt(gamma(x)*coef_R*T_1_3D(y,x));
53         M_r_3D(y,x) = w_1_3D(y,x)/sqrt(gamma(x)*coef_R*T_1_3D(y,x));
54
55         % Calculo de la relaciones de difusion
56
57         DF_eq_R_3D(y,x) = ...
58         cosd(beta_2_3D(y,x))/cosd(beta_1_3D(y,x))* ...
59         (1.12 + 0.61/sigma * (cosd(beta_1_3D(y,x)))^2 * ...
60         (tand(beta_1_3D(y,x))-tand(beta_2_3D(y,x))));
61         DF_eq_E_3D(y,x) = ...
62         cosd(alpha_1_3D(y,x))/cosd(alpha_2_3D(y,x))* ...
63         (1.12 + 0.61/sigma * (cosd(alpha_2_3D(y,x)))^2 * ...
64         (tand(alpha_2_3D(y,x))-tand(alpha_1_3D(y,x))));
65
66         % Calculo de las perdidas
67
68         zeta_p_R_3D(y,x) = 2*(0.004/(1-1.17*log(DF_eq_R_3D(y,x))))*...
69         sigma*(cosd(beta_1_3D(y,x)))^2/(cosd(beta_2_3D(y,x)))^3;
70         zeta_p_E_3D(y,x) = 2*(0.004/(1-1.17*log(DF_eq_E_3D(y,x))))*...
71         sigma*(cosd(alpha_2_3D(y,x)))^2/(cosd(alpha_1_3D(y,x)))^3;
72
73         zeta_R_3D(y,x)=(zeta_p_R_3D(y,x) + zeta_s_r + zeta_a_r);

```

```

75     zeta_E_3D(y,x)=(zeta_p_E_3D(y,x) + zeta_s_e + zeta_a_e);
76
77     perdidas_rotor_3D(y,x) = zeta_R_3D(y,x)*w_1_3D(y,x)^2/2;
78     perdidas_estator_3D(y,x) = zeta_E_3D(y,x)*c_2_3D(y,x)^2/2;
79
80     perdidas_3D(y,x) = ...
81     perdidas_rotor_3D(y,x) + perdidas_estator_3D(y,x);
82
83     W_u_3D_is(y,x) = W_u_3D - perdidas_3D(y,x);
84     eta_TT_3D(y,x) = W_u_3D_is(y,x)/W_u_3D;
85
86     radio = radio + (r_c_rotor(x)-r_r_rotor(x))/(puntos-1);
87
88     end
89 end

```

11.4.8. f_reaccion_constante

```

1  % LEY DE GRADO DE REACCION CONSTANTE
2
3  function[a,b,c_a1_3D,c_a2_3D,W_u_3D,u_3D,c_u1_3D,c_u2_3D, c_1_3D,c_2_3D,...
4     w_u1_3D,w_u2_3D,beta_1_3D,beta_2_3D, alpha_1_3D,alpha_2_3D,R_3D, ...
5     T_1_3D,M_c_3D,M_r_3D,DF_eq_R_3D,DF_eq_E_3D, ...
6     zeta_p_R_3D,zeta_p_E_3D, perdidas_rotor_3D,perdidas_estator_3D, ...
7     perdidas_3D,phi_3D,psi_3D,W_u_3D_is,eta_TT_3D,w_1_3D,w_2_3D] ...
8  =f_reaccion_constante(c_u1,c_u2,D_m,c_a,W_u,n_esc,u,r_r_rotor,r_c_rotor,...
9  T_01,c_p,gamma,coef_R,sigma,zeta_s_r,zeta_a_r,zeta_s_e,zeta_a_e,puntos)
10
11  W_u_3D = W_u;
12
13  b=1;
14  b_it=0;
15  a=0;
16  while b_it/b < abs(0.999)
17     b = D_m/2 * (c_u2-a*D_m/2);
18     b_it = D_m/2 * (a*D_m/2-c_u1);
19     a=a+0.01;
20  end
21
22  cte_1 = c_a^2 + 2*a*(a*(D_m/2)^2 - 2*b*log(D_m/2))
23  cte_2 = c_a^2 + 2*a*(a*(D_m/2)^2 + 2*b*log(D_m/2))
24
25  for x=1:n_esc
26
27     radio=r_r_rotor(x);
28
29     for y=1:puntos
30
31         c_a1_3D(y,x) = sqrt(cte_1 - 2*a*(a*radio^2 - 2*b*log(radio)));
32         c_a2_3D(y,x) = sqrt(cte_2 - 2*a*(a*radio^2 + 2*b*log(radio)));
33         u_3D(y,x)=u/(D_m/2)*radio;
34         c_u1_3D(y,x) = a*radio - b/radio;
35         c_u2_3D(y,x) = a*radio + b/radio;
36         c_1_3D(y,x) = sqrt(c_a1_3D(y,x)^2 + c_u1_3D(y,x).^2);

```

```

37 c_2_3D(y,x) = sqrt(c_a2_3D(y,x)^2 + c_u2_3D(y,x).^2);
38 w_u1_3D(y,x) = u_3D(y,x) - c_u1_3D(y,x);
39 w_u2_3D(y,x) = u_3D(y,x) - c_u2_3D(y,x);
40 w_1_3D(y,x) = sqrt(c_a1_3D(y,x)^2 + w_u1_3D(y,x)^2);
41 w_2_3D(y,x) = sqrt(c_a2_3D(y,x)^2 + w_u2_3D(y,x)^2);
42 beta_1_3D(y,x) = atand(w_u1_3D(y,x)/c_a1_3D(y,x));
43 beta_2_3D(y,x) = atand(w_u2_3D(y,x)/c_a2_3D(y,x));
44 alpha_1_3D(y,x) = atand(c_u1_3D(y,x)/c_a1_3D(y,x));
45 alpha_2_3D(y,x) = atand(c_u2_3D(y,x)/c_a2_3D(y,x));
46 R_3D(y,x) = (w_u1_3D(y,x) + w_u2_3D(y,x))/(2*u_3D(y,x));
47 phi_3D(y,x) = c_a1_3D(y,x)/u_3D(y,x);
48 psi_3D(y,x) = (c_u2_3D(y,x) - c_u1_3D(y,x))/u_3D(y,x);
49
50 % Calculo del n mero de Mach maximo
51
52 T_1_3D(y,x) = T_01(x) - c_1_3D(y,x)^2/(2000*c_p(x));
53 M_c_3D(y,x) = c_1_3D(y,x)/sqrt(gamma(x)*coef_R*T_1_3D(y,x));
54 M_r_3D(y,x) = w_1_3D(y,x)/sqrt(gamma(x)*coef_R*T_1_3D(y,x));
55
56 % Calculo de la relaciones de difusion
57
58 DF_eq_R_3D(y,x) = ...
59 cosd(beta_2_3D(y,x))/cosd(beta_1_3D(y,x))* ...
60 (1.12 + 0.61/sigma * (cosd(beta_1_3D(y,x)))^2 * ...
61 (tand(beta_1_3D(y,x)) - tand(beta_2_3D(y,x))));
62 DF_eq_E_3D(y,x) = ...
63 cosd(alpha_1_3D(y,x))/cosd(alpha_2_3D(y,x))* ...
64 (1.12 + 0.61/sigma * (cosd(alpha_2_3D(y,x)))^2 * ...
65 (tand(alpha_2_3D(y,x)) - tand(alpha_1_3D(y,x))));
66
67 % Calculo de las perdidas
68
69 zeta_p_R_3D(y,x) = 2*(0.004/(1-1.17*log(DF_eq_R_3D(y,x))))*...
70 sigma*(cosd(beta_1_3D(y,x)))^2/(cosd(beta_2_3D(y,x)))^3;
71 zeta_p_E_3D(y,x) = 2*(0.004/(1-1.17*log(DF_eq_E_3D(y,x))))*...
72 sigma*(cosd(alpha_2_3D(y,x)))^2/(cosd(alpha_1_3D(y,x)))^3;
73
74 zeta_R_3D(y,x) = (zeta_p_R_3D(y,x) + zeta_s_r + zeta_a_r);
75 zeta_E_3D(y,x) = (zeta_p_E_3D(y,x) + zeta_s_e + zeta_a_e);
76
77 perdidas_rotor_3D(y,x) = zeta_R_3D(y,x)*w_1_3D(y,x)^2/2;
78 perdidas_estator_3D(y,x) = zeta_E_3D(y,x)*c_2_3D(y,x)^2/2;
79
80 perdidas_3D(y,x) = ...
81 perdidas_rotor_3D(y,x) + perdidas_estator_3D(y,x);
82
83 W_u_3D_is(y,x) = W_u_3D - perdidas_3D(y,x);
84 eta_TT_3D(y,x) = W_u_3D_is(y,x)/W_u_3D;
85
86 radio = radio + (r_c_rotor(x) - r_r_rotor(x))/(puntos-1);
87
88 end
89 end

```

11.5. App Designer

```

1 classdef compresor_axial_exported < matlab.apps.AppBase
2
3     % Properties that correspond to app components
4     properties (Access = public)
5         UIFigure                matlab.ui.Figure
6         n_puntos                matlab.ui.control.NumericEditField
7         NmerodepuntosEditFieldLabel  matlab.ui.control.Label
8         ley_actual              matlab.ui.control.NumericEditField
9         ley_actualEditFieldLabel  matlab.ui.control.Label
10        TabGroup                matlab.ui.container.TabGroup
11        EspecificacionesTab     matlab.ui.container.Tab
12        PresinderemansoalaentradaKPaEditFieldLabel  matlab.ui.control.Label
13        rc_total                 matlab.ui.control.NumericEditField
14        RelacindecompresintotalEditFieldLabel  matlab.ui.control.Label
15        DF_eq_R                  matlab.ui.control.NumericEditField
16        RelacindedifusinenelrotorEditFieldLabel  matlab.ui.control.Label
17        m_dot                    matlab.ui.control.NumericEditField
18        GastomsicokgsEditFieldLabel  matlab.ui.control.Label
19        phi                      matlab.ui.control.NumericEditField
20        CoeficientedeflujoEditFieldLabel  matlab.ui.control.Label
21        R                        matlab.ui.control.NumericEditField
22        GradodereaccinEditFieldLabel  matlab.ui.control.Label
23        n                        matlab.ui.control.NumericEditField
24        RevolucionesporminutoEditFieldLabel  matlab.ui.control.Label
25        sigma                    matlab.ui.control.NumericEditField
26        SolidezdelacascadaEditFieldLabel  matlab.ui.control.Label
27        T_01_primer              matlab.ui.control.NumericEditField
28        TempderemansoalaentradaKEditFieldLabel  matlab.ui.control.Label
29        CASOBIDIMENSIONALLabel  matlab.ui.control.Label
30        CALCULARButton          matlab.ui.control.Button
31        Nopuedesermayora195Label  matlab.ui.control.Label
32        Valorentre04y07Label    matlab.ui.control.Label
33        Valorentre04y06Label    matlab.ui.control.Label
34        Valorentre1y14Label     matlab.ui.control.Label
35        P_01_primer              matlab.ui.control.NumericEditField
36        Resultados2DTab        matlab.ui.container.Tab
37        TabGroup2                matlab.ui.container.TabGroup
38        TriangulosdevelocidadesTab  matlab.ui.container.Tab
39        vaxialEditField          matlab.ui.control.NumericEditField
40        vaxialEditFieldLabel    matlab.ui.control.Label
41        deflexion_estator        matlab.ui.control.NumericEditField
42        DeflexinestatorLabel    matlab.ui.control.Label
43        deflexion_rotor         matlab.ui.control.NumericEditField
44        DeflexinrotorLabel      matlab.ui.control.Label
45        uEditField              matlab.ui.control.NumericEditField
46        uEditFieldLabel         matlab.ui.control.Label
47        w2EditField             matlab.ui.control.NumericEditField
48        w2EditFieldLabel       matlab.ui.control.Label
49        c2EditField             matlab.ui.control.NumericEditField
50        c2EditFieldLabel       matlab.ui.control.Label
51        w1EditField             matlab.ui.control.NumericEditField
52        w1EditFieldLabel       matlab.ui.control.Label
53        c1EditField             matlab.ui.control.NumericEditField

```

54	c1EditFieldLabel	matlab.ui.control.Label
55	beta2	matlab.ui.control.NumericEditField
56	EditField_4Label	matlab.ui.control.Label
57	beta1	matlab.ui.control.NumericEditField
58	EditField_3Label	matlab.ui.control.Label
59	alpha2	matlab.ui.control.NumericEditField
60	EditField_2Label	matlab.ui.control.Label
61	alpha1	matlab.ui.control.NumericEditField
62	EditFieldLabel	matlab.ui.control.Label
63	CoefdecargaEditField	matlab.ui.control.NumericEditField
64	CoefdecargaEditFieldLabel	matlab.ui.control.Label
65	NdeescalonamientosEditField	matlab.ui.control.NumericEditField
66	NdeescalonamientosEditFieldLabel	matlab.ui.control.Label
67	VelocidadesmsLabel	matlab.ui.control.Label
68	ngulosdeflujoLabel	matlab.ui.control.Label
69	UIAxes	matlab.ui.control.UIAxes
70	labesTab	matlab.ui.container.Tab
71	CaladoestatorEditField	matlab.ui.control.NumericEditField
72	CaladoestatorEditFieldLabel	matlab.ui.control.Label
73	CaladorotorEditField	matlab.ui.control.NumericEditField
74	CaladorotorEditFieldLabel	matlab.ui.control.Label
75	CurvaturaestatorEditField	matlab.ui.control.NumericEditField
76	CurvaturaestatorEditFieldLabel	matlab.ui.control.Label
77	CurvaturarotorEditField	matlab.ui.control.NumericEditField
78	CurvaturarotorEditFieldLabel	matlab.ui.control.Label
79	DesviacinestatorEditField	matlab.ui.control.NumericEditField
80	DesviacinestatorEditFieldLabel	matlab.ui.control.Label
81	DesviacinrotorEditField	matlab.ui.control.NumericEditField
82	DesviacinrotorEditFieldLabel	matlab.ui.control.Label
83	EditField_4Label_2	matlab.ui.control.Label
84	alpha_3_p	matlab.ui.control.NumericEditField
85	alpha_2_p	matlab.ui.control.NumericEditField
86	EditField_2Label_2	matlab.ui.control.Label
87	EditField_3Label_2	matlab.ui.control.Label
88	beta_2_p	matlab.ui.control.NumericEditField
89	beta_1_p	matlab.ui.control.NumericEditField
90	Label_2	matlab.ui.control.Label
91	ngulosdellabeLabel	matlab.ui.control.Label
92	UIAxes2	matlab.ui.control.UIAxes
93	GeometraTab	matlab.ui.container.Tab
94	ESTATORLabel	matlab.ui.control.Label
95	geometria_est	matlab.ui.control.Table
96	ROTORLabel	matlab.ui.control.Label
97	geometria_rot	matlab.ui.control.Table
98	GeometracompresorTab	matlab.ui.container.Tab
99	Label_7	matlab.ui.control.Label
100	UIAxes7	matlab.ui.control.UIAxes
101	PrdidasTab	matlab.ui.container.Tab
102	RelacindedifusinestatorEditField	matlab.ui.control.NumericEditField
103	RelacindedifusinestatorEditFieldLabel	matlab.ui.control.Label
104	Tampocodebesermayora195Label	matlab.ui.control.Label
105	perdidas_est	matlab.ui.control.Table
106	ESTATORLabel_2	matlab.ui.control.Label
107	ROTORLabel_2	matlab.ui.control.Label
108	perdidas_rot	matlab.ui.control.Table

109	PresionesTab	matlab.ui.container.Tab
110	UIAxes3_2	matlab.ui.control.UIAxes
111	UIAxes3	matlab.ui.control.UIAxes
112	TemperaturasTab	matlab.ui.container.Tab
113	UIAxes3_4	matlab.ui.control.UIAxes
114	UIAxes3_3	matlab.ui.control.UIAxes
115	DensidadesTab	matlab.ui.container.Tab
116	UIAxes3_6	matlab.ui.control.UIAxes
117	UIAxes3_5	matlab.ui.control.UIAxes
118	PrestacionesTab	matlab.ui.container.Tab
119	RendimientototalEditField	matlab.ui.control.NumericEditField
120	RendimientototalEditFieldLabel	matlab.ui.control.Label
121	TrabajoespecficoporescalonamientoKJKgEditField ...	
122	matlab.ui.control.NumericEditField	
123	TrabajoespecficoporescalonamientoKJKgKLabel	matlab.ui.control.Label
124	UIAxes8_2	matlab.ui.control.UIAxes
125	UIAxes8	matlab.ui.control.UIAxes
126	LeytorsionalTab	matlab.ui.container.Tab
127	Button	matlab.ui.control.Button
128	coeficientedeflujoalacabezadelprimerescalonamientoEditField ...	
129	matlab.ui.control.NumericEditField	
130	coefflujocrticoEditFieldLabel	matlab.ui.control.Label
131	LeydegradodereaccinconstanteLabel	matlab.ui.control.Label
132	LeyexponencialLabel	matlab.ui.control.Label
133	LeydetorbellinolibreLabel	matlab.ui.control.Label
134	exponencial	matlab.ui.control.Button
135	reaccion	matlab.ui.control.Button
136	torbellino	matlab.ui.control.Button
137	Label_3	matlab.ui.control.Label
138	CASOTRIDIMENSIONALLabel	matlab.ui.control.Label
139	Resultados3DTab	matlab.ui.container.Tab
140	SiguienteescalonamientoButton	matlab.ui.control.Button
141	escalonamientos	matlab.ui.control.NumericEditField
142	EscalonamientoonLabel_2	matlab.ui.control.Label
143	EscalonamientoanteriorButton	matlab.ui.control.Button
144	TabGroup3	matlab.ui.container.TabGroup
145	TringulosdevelocidadesTab_2	matlab.ui.container.Tab
146	Ley	matlab.ui.control.EditField
147	Mach_Maximo	matlab.ui.control.NumericEditField
148	NmerodeMachMximoEditFieldLabel	matlab.ui.control.Label
149	triangulos_3D	matlab.ui.control.UIAxes
150	Prdidastab_2	matlab.ui.container.Tab
151	RendimientototaldelescalonamientoEditField ...	
152	matlab.ui.control.NumericEditField	
153	RendimientototaldelescalonamientoEditFieldLabel ...	
154	matlab.ui.control.Label	
155	TrabajoespecficoporescalonamientoKJkgEditField ...	
156	matlab.ui.control.NumericEditField	
157	TrabajoespecficoporescalonamientoKJkgEditFieldLabel ...	
158	matlab.ui.control.Label	
159	PRDIDASTRABAJOYRENDIMIENTOLabel	matlab.ui.control.Label
160	perdidastab_3D	matlab.ui.control.Table
161	ngulosTab	matlab.ui.container.Tab
162	Ley_2	matlab.ui.control.EditField
163	UIAxes6	matlab.ui.control.UIAxes


```

164     CoeficientesTab                matlab.ui.container.Tab
165     Ley_3                          matlab.ui.control.EditField
166     UIAxes6_2                      matlab.ui.control.UIAxes
167 end
168
169 % Callbacks that handle component events
170 methods (Access = private)
171
172     % Button pushed function: CALCULARButton
173     function CALCULARButtonPushed(app, event)
174         [alpha_1,alpha_2,alpha_2_prima,alpha_3, ...
175         alpha_3_prima,beta_1,beta_1_prima,beta_2,beta_2_prima, ...
176         c_1,c_2,c_3,c_a,c_p,c_u1,c_u2,coef_R, ...
177         curvatura_estator,curvatura_rotor,D_m, ...
178         desviacion_estator,desviacion_rotor,DF_eq_E,eta_c,eta_TT,...
179         eta_politropico,...
180         gamma,H_estator,H_rotor,Lieblein_e,Lieblein_r,M_w_1,n_esc, ...
181         P_01,P_02,P_03,P_1,P_2,P_3,perdidas_estator,perdidas_rotor, ...
182         psi,r_c,r_c_estator,r_c_rotor,r_r_estator,r_r_rotor, ...
183         rc_total_obtenida,rho_1,rho_2,rho_3,s_estator,s_rotor, suma_perdidas,...
184         T_01,T_02,T_03,T_03ss,T_1,T_2,T_2s,T_3,T_total_final,u,w_1,w_2, ...
185         W_u,zeta_a_e,zeta_a_r,zeta_e,zeta_p_e,zeta_p_r,zeta_r, ...
186         zeta_s_e,zeta_s_r] ...
187 = f_caso_2D(app.T_01_primer.Value,app.P_01_primer.Value,...
188 app.sigma.Value,app.n.Value,app.m_dot.Value,app.DF_eq_R.Value,...
189 app.R.Value,app.phi.Value,app.rc_total.Value);
190
191     deflexion_rot=beta_1-beta_2;
192     deflexion_est=alpha_2-alpha_3;
193
194     %app.Mach_Maximo.Value = max(M_w_1);
195
196     app.NdeescalonamientosEditField.Value = n_esc;
197     app.CoeffdecargaEditField.Value = psi;
198     app.c1EditField.Value = c_1;
199     app.c2EditField.Value = c_2;
200     app.w1EditField.Value = w_1;
201     app.w2EditField.Value = w_2;
202     app.uEditField.Value = u;
203     %app.c3EditField.Value = c_3;
204     app.alpha1.Value = alpha_1;
205     app.alpha2.Value = alpha_2;
206     app.beta1.Value = beta_1;
207     app.beta2.Value = beta_2;
208     %app.alpha3.Value = alpha_3;
209     app.deflexion_rotor.Value=deflexion_rot;
210     app.deflexion_estator.Value = deflexion_est;
211     app.vaxialEditField.Value = c_a;
212
213
214     %%% GEOMETRIA TRIANGULOS
215
216     A=[0,0];
217     B=[w_2*sind(beta_2),w_2*cosd(beta_2)];
218     C=[u,0];

```



```

219 D=[0,0];
220 E=[w_1*sind(beta_1),w_1*cosd(beta_1)];
221 F=[u,0];
222 %figure(1)
223 %figure(gcf);
224 x=[A(1) B(1) C(1) A(1)];
225 y=[A(2) B(2) C(2) A(2)];
226 z=[D(1) E(1) F(1) D(1)];
227 t=[D(2) E(2) F(2) D(2)];
228
229 plot(app.UIAxes,x,y,z,t);
230
231 %% GEOMETRIA CASCADA
232
233 %cuerda del primer escalonamiento
234 cuerda_rotor=s_rotor(1)*app.sigma.Value;
235 cuerda_estator=s_estator(1)*app.sigma.Value;
236 calado_rotor=0.5*(beta_1_prima+beta_2_prima);
237 calado_estator=0.5*(alpha_3_prima+alpha_2_prima);
238
239 app.beta_1_p.Value=beta_1_prima;
240 app.beta_2_p.Value=beta_2_prima;
241 app.alpha_2_p.Value=alpha_2_prima;
242 app.alpha_3_p.Value=alpha_3_prima;
243 app.DesviacinrotorEditField.Value=desviacion_rotor;
244 app.DesviacinestatorEditField.Value=desviacion_estator;
245 app.CurvaturarotorEditField.Value=curvatura_rotor;
246 app.CurvaturaestatorEditField.Value=curvatura_estator;
247 app.CaladorotorEditField.Value=calado_rotor;
248 app.CaladoestatorEditField.Value=calado_estator;
249
250 % rotor de arriba
251
252 x_intrados_r1=[0,0.5,0.75,1.25,2.5,5,7.5,10,15,20,25,30,...
253 35,40,45,50,55,60,65,70,75,80,85,90,95,100];
254 x_extrados_r1=x_intrados_r1;
255 y_r1=-[0,0.25,0.35,0.535,0.93,1.58,2.12,2.585,3.365,3.98,...
256 4.475,4.86,5.15,5.355,5.475,5.515,5.475,5.355,5.150,...
257 4.860,4.475,3.980,3.365,2.585,1.580,0];
258 espesor_r1=[0,1.544,1.864,2.338,3.480,4.354,5.294,6.080,...
259 7.332,8.286,9.006,9.52,9.848,9.992,9.926,9.624,9.06,...
260 8.292,7.364,6.312,5.168,3.974,2.77,1.62,0.612,0];
261 y_intrados_r1=y_r1+espesor_r1/2;
262 y_extrados_r1=y_r1-espesor_r1/2;
263 x_intrados_r1_prima=x_intrados_r1*cosd(calado_rotor)+...
264 y_r1*sind(calado_rotor);
265 x_extrados_r1_prima=x_intrados_r1_prima;
266 y_intrados_r1_prima=y_intrados_r1*cosd(calado_rotor)-...
267 x_intrados_r1*sind(calado_rotor);
268 y_extrados_r1_prima=y_extrados_r1*cosd(calado_rotor)-...
269 x_extrados_r1*sind(calado_rotor);
270
271 % rotor de abajo
272
273 x_intrados_r2_prima=x_intrados_r1_prima;

```

```

274     x_extrados_r2_prima=x_intrados_r1_prima;
275     y_intrados_r2_prima=y_intrados_r1_prima-100*app.sigma.Value;
276     y_extrados_r2_prima=y_extrados_r1_prima-100*app.sigma.Value;
277
278
279     % estator de arriba
280
281     x_intrados_e1_prima=x_intrados_r1_prima+100;
282     x_extrados_e1_prima=x_intrados_e1_prima;
283     y_e1=-y_r1;
284     y_intrados_e1=y_e1+espesor_r1/2;
285     y_extrados_e1=y_e1-espesor_r1/2;
286     y_intrados_e1_prima=y_intrados_e1*cosd(calado_estator)+...
287         x_intrados_r1*sind(calado_estator);
288     y_extrados_e1_prima=y_extrados_e1*cosd(calado_estator)+...
289         x_extrados_r1*sind(calado_estator);
290
291
292     % estator de abajo
293
294     x_intrados_e2_prima=x_intrados_e1_prima;
295     x_extrados_e2_prima=x_intrados_e2_prima;
296     y_intrados_e2_prima=y_intrados_e1_prima-100*app.sigma.Value;
297     y_extrados_e2_prima=y_extrados_e1_prima-100*app.sigma.Value;
298
299     % l neas rotor
300
301
302     x_int_rotor=[0,-10];
303     y_int_rotor=[0,10*tand(beta_1_prima)];
304     x_ext_rotor_geom=[x_intrados_r1_prima(end),...
305         x_intrados_r1_prima(end)+20];
306     y_ext_rotor_geom=[y_intrados_r1_prima(end),...
307         y_intrados_r1_prima(end)-20*tand(beta_2_prima)];
308     x_ext_rotor_fluj=[x_intrados_r1_prima(end),...
309         x_intrados_r1_prima(end)+20];
310     y_ext_rotor_fluj=[y_intrados_r1_prima(end),...
311     y_intrados_r1_prima(end)-20*tand(beta_2_prima+desviacion_rotor)];
312
313     % l neas estator
314
315     x_int_estator=[x_intrados_e2_prima(1),...
316         -10+x_intrados_e2_prima(1)];
317     y_int_estator=[y_intrados_e2_prima(1),...
318         y_intrados_e2_prima(1)-10*tand(alpha_2_prima)];
319     x_ext_estator_geom=[x_intrados_e2_prima(end),...
320         x_intrados_e2_prima(end)+20];
321     y_ext_estator_geom=[y_intrados_e2_prima(end),...
322         y_intrados_e2_prima(end)-20*tand(-alpha_3_prima)];
323     x_ext_estator_fluj=[x_intrados_e2_prima(end),...
324         x_intrados_e2_prima(end)+20];
325     y_ext_estator_fluj=[y_intrados_e2_prima(end),...
326         y_intrados_e2_prima(end)-20*tand...
327         (-alpha_3_prima+desviacion_estator)];
328

```

```

329
330     set(app.UIAxes2, 'DataAspectRatio', [1 2 1])
331     plot(app.UIAxes2, x_intrados_r1_prima, y_intrados_r1_prima, ...
332           "black", x_extrados_r1_prima, y_extrados_r1_prima, "black", ...
333           x_intrados_r2_prima, y_intrados_r2_prima, "black", ...
334           x_extrados_r2_prima, y_extrados_r2_prima, "black", ...
335           x_intrados_e1_prima, y_intrados_e1_prima, "black", ...
336           x_extrados_e1_prima, y_extrados_e1_prima, "black", ...
337           x_intrados_e2_prima, y_intrados_e2_prima, "black", ...
338           x_extrados_e2_prima, y_extrados_e2_prima, "black", ...
339           x_int_rotor, y_int_rotor, "red", x_ext_rotor_geom, ...
340           y_ext_rotor_geom, "red", ...
341           x_ext_rotor_fluj, y_ext_rotor_fluj, "blue", ...
342           x_int_estator, y_int_estator, "red", ...
343           x_ext_estator_geom, y_ext_estator_geom, "blue", ...
344           x_ext_estator_fluj, y_ext_estator_fluj, "red");
345     legend(app.UIAxes2, {'', '', '', '', '', '', '', '', ''}, ...
346           'Dirección geométrica', 'Dirección del flujo'}, ...
347           'Location', 'southeast');
348
349     %%% PRESIONES
350
351     a=P_1;
352     b=P_2;
353     c=P_3;
354     x=1:n_esc;
355     plot(app.UIAxes3_2, x, a, 'red', x, b, 'green', x, c, 'blue')
356     legend(app.UIAxes3_2, {'Presión 1', 'Presión 2', 'Presión 3'}, ...
357           'Location', 'north');
358
359
360
361     a=P_01;
362     b=P_02;
363     c=P_03;
364     x=1:n_esc;
365     plot(app.UIAxes3, x, a, 'red', x, b, 'green', x, c, 'blue')
366     legend(app.UIAxes3, {'Presión Total 1', 'Presión Total 2', ...
367           'Presión Total 3'}, 'Location', 'north');
368
369     %%% TEMPERATURAS
370
371     a=T_1;
372     b=T_2;
373     c=T_3;
374     x=1:n_esc;
375     plot(app.UIAxes3_3, x, a, 'red', x, b, 'green', x, c, 'blue')
376     legend(app.UIAxes3_3, {'Temperatura 1', 'Temperatura 2', ...
377           'Temperatura 3'}, 'Location', 'north');
378
379
380
381     a=T_01;
382     b=T_02;
383     c=T_03;

```

```

384     x=1:n_esc;
385     plot(app.UIAxes3_4,x,a,'red',x,b,'green',x,c,'blue')
386         legend(app.UIAxes3_4,{'Temperatura Total 1',...
387             'Temperatura Total 2','Temperatura Total 3'},...
388             'Location','north');
389
390
391         %%% DENSIDADES
392
393     a=rho_1;
394     b=rho_2;
395     c=rho_3;
396     x=1:n_esc;
397     plot(app.UIAxes3_5,x,a,'red',x,b,'green',x,c,'blue')
398         legend(app.UIAxes3_5,{'Densidad 1','Densidad 2',...
399             'Densidad 3'},'Location','north');
400
401
402     a=c_p;
403     x=1:n_esc;
404     plot(app.UIAxes3_6,x,a)
405         legend(app.UIAxes3_6,{'Calor especifico'},'Location','north');
406
407         %%% PRESTACIONES
408
409     a=r_c;
410     x=1:n_esc;
411     plot(app.UIAxes8,x,a)
412     legend(app.UIAxes8,{'Relacion de compresion'},'Location','north');
413
414     a=eta_TT;
415     x=1:n_esc;
416     plot(app.UIAxes8_2,x,a,'red')
417         legend(app.UIAxes8_2,{'Rendimiento'},'Location','north');
418
419     app.TrabajoespecificoporescalonamientoKJKgEditField.Value=W_u;
420
421
422     suma_perdidas_total=0;
423     for i=1:n_esc
424
425         suma_perdidas_total=suma_perdidas(i)+suma_perdidas_total;
426
427     end
428
429     T_obs=T_01(1)*(P_03(n_esc)/P_01(1))^(gamma((n_esc-1)/2)-1)/gamma((n_esc-
430     h_obs=T_obs*c_p((n_esc-1)/2);
431
432
433     eta_total=(h_obs-c_p(1)*T_01(1))/(W_u*n_esc/1000);
434
435
436
437     app.RendimientototalEditField.Value=eta_total;
438

```

```
439
440     %%% GEOMETRIA
441
442     cuerda_rotor=s_rotor/app.sigma.Value;
443     cuerda_estator=s_estator/app.sigma.Value;
444
445     % rotor
446
447
448
449     for k=1:(n_esc)
450         tabla_geometria_rotor(k,1)=k;
451         tabla_geometria_rotor(k,2)=(H_rotor(k)*100);
452         n_alabes_rotor(k)=round(2*pi*(D_m/2)/s_rotor(k));
453         s_rotor_real(k)=pi*D_m/(n_alabes_rotor(k));
454         tabla_geometria_rotor(k,3)=(s_rotor_real(k)*100);
455         tabla_geometria_rotor(k,4)=(cuerda_rotor(k)*100);
456         tabla_geometria_rotor(k,5)=n_alabes_rotor(k);
457     end
458
459
460
461
462
463     app.geometria_rot.Data = (tabla_geometria_rotor);
464
465
466     % estator
467
468     for l=1:n_esc
469         tabla_geometria_estator(l,1)=1;
470         tabla_geometria_estator(l,2)=H_estator(l)*100;
471         n_alabes_estator(l)=round(2*pi*(D_m/2)/s_estator(l));
472         s_estator_real(l)=pi*D_m/(n_alabes_estator(l));
473         tabla_geometria_estator(l,3)=s_estator_real(l)*100;
474         tabla_geometria_estator(l,4)=cuerda_estator(l)*100;
475         tabla_geometria_estator(l,5)=n_alabes_estator(l);
476     end
477
478     app.geometria_est.Data = tabla_geometria_estator;
479
480
481
482     %%% PERDIDAS
483
484     % rotor
485
486     for i=1:n_esc
487         tabla_perdidas_rotor(i,1)=i;
488         tabla_perdidas_rotor(i,2)=zeta_p_r;
489         tabla_perdidas_rotor(i,3)=zeta_a_r;
490         tabla_perdidas_rotor(i,4)=zeta_s_r;
491         tabla_perdidas_rotor(i,5)=perdidas_rotor;
492     end
493
```

```

494     app.perdidas_rot.Data = tabla_perdidas_rotor;
495
496     % estator
497
498     for j=1:n_esc
499         tabla_perdidas_estator(j,1)=j;
500         tabla_perdidas_estator(j,2)=zeta_p_e;
501         tabla_perdidas_estator(j,3)=zeta_a_e;
502         tabla_perdidas_estator(j,4)=zeta_s_e;
503         tabla_perdidas_estator(j,5)=perdidas_estator;
504     end
505
506     app.perdidas_est.Data = tabla_perdidas_estator;
507
508     app.RelacindedifusinestatorEditField.Value=DF_eq_E;
509
510
511     %%% ALTURAS DE ALABES
512
513     y_superior=zeros(1,2*n_esc);
514     y_inferior=zeros(1,2*n_esc);
515     y_radio_medio=zeros(1,2*n_esc);
516
517     cascadas=1:n_esc*2;
518     for i=1:n_esc
519         y_superior(2*i-1)=D_m/2+H_rotor(i)/2;
520         y_superior(2*i)=D_m/2+H_estator(i)/2;
521         y_inferior(2*i-1)=D_m/2-H_rotor(i)/2;
522         y_inferior(2*i)=D_m/2-H_estator(i)/2;
523         y_radio_medio(2*i-1)=D_m/2;
524         y_radio_medio(2*i)=D_m/2;
525     end
526
527
528
529
530
531     plot(app.UIAxes7,cascadas,y_superior,"blue",cascadas,...
532          y_inferior,"red",cascadas,y_radio_medio,'k-.');
533     legend(app.UIAxes7,{'radio cabeza','radio raiz',...
534          'radio medio'},'Location','north');
535
536
537     %%% ADVERTENCIA
538
539     if H_rotor(1)/D_m > 1/9
540
541         advertencia;
542
543     end
544
545
546
547
548

```

```

549
550
551     end
552
553     % Button pushed function: torbellino
554     function torbellinoButtonPushed(app, event)
555         [alpha_1,alpha_2,alpha_2_prima,alpha_3, ...
556         alpha_3_prima, beta_1,beta_1_prima,beta_2,beta_2_prima, ...
557         c_1,c_2,c_3,c_a,c_p,c_u1,c_u2,coef_R, ...
558         curvatura_estator,curvatura_rotor,D_m, ...
559         desviacion_estator,desviacion_rotor,DF_eq_E,eta_c,eta_TT,...
560         eta_politropico,...
561         gamma,H_estator,H_rotor,Lieblein_e,Lieblein_r,M_w_1,n_esc, ...
562         P_01,P_02,P_03,P_1,P_2,P_3,perdidas_estator,perdidas_rotor, ...
563         psi,r_c,r_c_estator,r_c_rotor,r_r_estator,r_r_rotor, ...
564         rc_total_obtenida,rho_1,rho_2,rho_3,s_estator,s_rotor,...
565         suma_perdidas,...
566         T_01,T_02,T_03,T_03ss,T_1,T_2,T_2s,T_3,T_total_final,u,w_1,w_2, ...
567         W_u,zeta_a_e,zeta_a_r,zeta_e,zeta_p_e,zeta_p_r,zeta_r, ...
568         zeta_s_e,zeta_s_r] ...
569         = f_caso_2D(app.T_01_primer.Value,app.P_01_primer.Value,...
570         app.sigma.Value,app.n.Value,app.m_dot.Value,app.DF_eq_R.Value,...
571         app.R.Value,app.phi.Value,app.rc_total.Value);
572
573         [b_1,b_2,c_a_3D,W_u_3D,u_3D,c_u1_3D,c_u2_3D, c_1_3D,c_2_3D, ...
574         w_u1_3D,w_u2_3D,beta_1_3D,beta_2_3D, alpha_1_3D,alpha_2_3D,R_3D, ...
575         T_1_3D,M_c_3D,M_r_3D,DF_eq_R_3D,DF_eq_E_3D, ...
576         zeta_p_R_3D,zeta_p_E_3D, perdidas_rotor_3D,perdidas_estator_3D, ...
577         perdidas_3D,phi_3D,psi_3D,W_u_3D_is,eta_TT_3D,w_1_3D,w_2_3D] ...
578         = f_torbellino_libre(c_u1,c_u2,D_m,c_a,W_u,n_esc,u,r_r_rotor,r_c_rotor, ...
579         T_01,c_p,gamma,coef_R,app.sigma.Value,zeta_s_r,zeta_a_r,zeta_s_e,...
580         zeta_a_e,app.n_puntos.Value)
581
582     %%% TRIANGULOS 3D
583
584     % posicion inicial
585
586     A_1=[0,0];
587     A_2=[w_2_3D(1,1)*sind(beta_2_3D(1,1)),...
588         w_2_3D(1,1)*cosd(beta_2_3D(1,1))];
589     A_3=[u_3D(1,1),0];
590     A_4=[0,0];
591     A_5=[w_1_3D(1,1)*sind(beta_1_3D(1,1)),...
592         w_1_3D(1,1)*cosd(beta_1_3D(1,1))];
593     A_6=[u_3D(1,1),0];
594
595     x_raiz=[A_1(1) A_2(1) A_3(1) A_4(1)];
596     y_raiz=[A_1(2) A_2(2) A_3(2) A_4(2)];
597     z_raiz=[A_5(1) A_6(1) A_4(1)];
598     t_raiz=[A_5(2) A_6(2) A_4(2)];
599
600     % posicion media
601 %
602     C_1=[0,0];
603     C_2=[w_2_3D(3,1)*sind(beta_2_3D(3,1)),...

```

```

604         w_2_3D(3,1)*cosd(beta_2_3D(3,1))];
605     C_3=[u_3D(3,1),0];
606     C_4=[0,0];
607     C_5=[w_1_3D(3,1)*sind(beta_1_3D(3,1)),...
608         w_1_3D(3,1)*cosd(beta_1_3D(3,1))];
609     C_6=[u_3D(3,1),0];
610
611     x_medio=[C_1(1) C_2(1) C_3(1) C_4(1)];
612     y_medio=[C_1(2) C_2(2) C_3(2) C_4(2)];
613     z_medio=[C_5(1) C_6(1) C_4(1)];
614     t_medio=[C_5(2) C_6(2) C_4(2)];
615
616     % posicion final
617 %
618     E_1=[0,0];
619     E_2=[w_2_3D(5,1)*sind(beta_2_3D(5,1)),...
620         w_2_3D(5,1)*cosd(beta_2_3D(5,1))];
621     E_3=[u_3D(5,1),0];
622     E_4=[0,0];
623     E_5=[w_1_3D(5,1)*sind(beta_1_3D(5,1)),...
624         w_1_3D(5,1)*cosd(beta_1_3D(5,1))];
625     E_6=[u_3D(5,1),0];
626
627     x_cabeza=[E_1(1) E_2(1) E_3(1) E_4(1)];
628     y_cabeza=[E_1(2) E_2(2) E_3(2) E_4(2)];
629     z_cabeza=[E_5(1) E_6(1) E_4(1)];
630     t_cabeza=[E_5(2) E_6(2) E_4(2)];
631
632
633     plot(app.triangulos_3D,x_raiz,y_raiz,...
634         "black",z_raiz,t_raiz,"black", ...
635         x_medio,y_medio,"red",z_medio,t_medio,"red", ...
636         x_cabeza,y_cabeza,"magenta",z_cabeza,t_cabeza,"magenta");
637     legend(app.triangulos_3D,{'','radio ra z',...
638         ','radio medio','','radio cabeza'},'Location','northeast');
639
640     app.escalonamientos.Value=1;
641     app.Mach_Maximo.Value = max(M_w_1);
642     app.Ley.Value='Torbellino libre';
643     app.Ley_2.Value='Torbellino libre';
644     app.Ley_3.Value='Torbellino libre';
645     app.ley_actual.Value = 1;
646
647
648
649
650     %% ANGULOS
651
652     x=r_r_rotor(1)/r_c_rotor(1):(1-r_r_rotor(1)/r_c_rotor(1))/...
653         (app.n_puntos.Value-1):1;
654     alpha_1_g=alpha_1_3D(:,1);
655     alpha_2_g=alpha_2_3D(:,1);
656     beta_1_g=beta_1_3D(:,1);
657     beta_2_g=beta_2_3D(:,1);
658     plot(app.UIAxes6,x,alpha_1_g,"blue",x,alpha_2_g,"green",...

```



```

659         x,beta_1_g,"red",x,beta_2_g,"magenta")
660     legend(app.UIAxes6,{'alpha 1','alpha 2','Beta 1','Beta 2'},...
661         'Location','northwest');
662
663
664     %%% COEFICIENTES
665
666     x=r_r_rotor(1)/r_c_rotor(1):(1-r_r_rotor(1)/r_c_rotor(1))/...
667         (app.n_puntos.Value-1):1;
668     R_3D_g=R_3D(:,1);
669     phi_3D_g=phi_3D(:,1);
670     psi_3D_g=psi_3D(:,1);
671     plot(app.UIAxes6_2,x,R_3D_g,"blue",x,phi_3D_g,"green",...
672         x,psi_3D_g,"red")
673     legend(app.UIAxes6_2,{'R',' ',' '},'Location','northwest');
674
675     %%% PERDIDAS
676
677     tabla_perdidas_3D(1,1)=D_m/2-H_rotor(1)/2;
678     tabla_perdidas_3D(2,1)=D_m/2;
679     tabla_perdidas_3D(3,1)=D_m/2+H_rotor(1)/2;
680     tabla_perdidas_3D(1,2)=perdidas_rotor_3D(1,1);
681     tabla_perdidas_3D(2,2)=...
682         perdidas_rotor_3D((app.n_puntos.Value+1)/2,1);
683     tabla_perdidas_3D(3,2)=perdidas_rotor_3D(app.n_puntos.Value,1);
684     tabla_perdidas_3D(1,3)=perdidas_estator_3D(1,1);
685     tabla_perdidas_3D(2,3)=...
686         perdidas_estator_3D((app.n_puntos.Value+1)/2,1);
687     tabla_perdidas_3D(3,3)=...
688         perdidas_estator_3D(app.n_puntos.Value,1);
689     tabla_perdidas_3D(1,4)=W_u_3D;
690     tabla_perdidas_3D(2,4)=W_u_3D;
691     tabla_perdidas_3D(3,4)=W_u_3D;
692     tabla_perdidas_3D(1,5)=eta_TT_3D(1,1);
693     tabla_perdidas_3D(2,5)=eta_TT_3D((app.n_puntos.Value+1)/2,1);
694     tabla_perdidas_3D(3,5)=eta_TT_3D(app.n_puntos.Value,1);
695
696     app.perdidas_3D.Data = tabla_perdidas_3D;
697     app.TrabajoespecificoporescalonamientoKJkgEditField.Value =...
698         W_u_3D;
699     app.RendimientototaldelescalonamientoEditField.Value=...
700         (tabla_perdidas_3D(1,5)+tabla_perdidas_3D(2,5)+...
701         tabla_perdidas_3D(3,5))/3;
702
703
704
705     end
706
707     % Button pushed function: reaccion
708     function reaccionButtonPushed(app, event)
709         [alpha_1,alpha_2,alpha_2_prima,alpha_3, ...
710         alpha_3_prima, beta_1,beta_1_prima,beta_2,beta_2_prima, ...
711         c_1,c_2,c_3,c_a,c_p,c_u1,c_u2,coef_R, ...
712         curvatura_estator,curvatura_rotor,D_m, ...
713         desviacion_estator,desviacion_rotor,DF_eq_E,...

```

```

714     eta_c, eta_TT, eta_politropico, ...
715     gamma, H_estator, H_rotor, Lieblein_e, Lieblein_r, M_w_1, n_esc, ...
716     P_01, P_02, P_03, P_1, P_2, P_3, perdidas_estator, perdidas_rotor, ...
717     psi, r_c, r_c_estator, r_c_rotor, r_r_estator, r_r_rotor, ...
718 rc_total_obtenida, rho_1, rho_2, rho_3, s_estator, s_rotor, suma_perdidas, ...
719     T_01, T_02, T_03, T_03ss, T_1, T_2, T_2s, T_3, T_total_final, u, w_1, w_2, ...
720     W_u, zeta_a_e, zeta_a_r, zeta_e, zeta_p_e, zeta_p_r, zeta_r, ...
721     zeta_s_e, zeta_s_r] ...
722 = f_caso_2D(app.T_01_primer.Value, app.P_01_primer.Value, ...
723 app.sigma.Value, app.n.Value, app.m_dot.Value, app.DF_eq_R.Value, ...
724 app.R.Value, app.phi.Value, app.rc_total.Value);
725
726     [a, b, c_a1_3D, c_a2_3D, W_u_3D, u_3D, ...
727     c_u1_3D, c_u2_3D, c_1_3D, c_2_3D, ...
728     w_u1_3D, w_u2_3D, beta_1_3D, beta_2_3D, alpha_1_3D, alpha_2_3D, R_3D, ...
729     T_1_3D, M_c_3D, M_r_3D, DF_eq_R_3D, DF_eq_E_3D, ...
730     zeta_p_R_3D, zeta_p_E_3D, perdidas_rotor_3D, perdidas_estator_3D, ...
731     perdidas_3D, phi_3D, psi_3D, W_u_3D_is, eta_TT_3D, w_1_3D, w_2_3D] ...
732 = f_reaccion_constante(c_u1, c_u2, D_m, c_a, ...
733 W_u, n_esc, u, r_r_rotor, r_c_rotor, ...
734 T_01, c_p, gamma, coef_R, app.sigma.Value, zeta_s_r, zeta_a_r, ...
735 zeta_s_e, zeta_a_e, app.n_puntos.Value);
736
737 app.coeficientedeflujoalacabezadelprimerescalonamientoEditField.Value=...
738 phi_3D(app.n_puntos.Value, 1);
739
740     if phi_3D(app.n_puntos.Value, 1) < 0.35;
741
742
743     mensajeerror(app.T_01_primer.Value, app.P_01_primer.Value, ...
744 app.sigma.Value, app.n.Value, app.m_dot.Value, app.DF_eq_R.Value, ...
745 app.R.Value, app.phi.Value, app.rc_total.Value, app.n_puntos.Value);
746
747     else
748
749
750
751
752
753
754
755     %%% TRIANGULOS 3D
756
757     % posicion inicial
758
759     A_1=[0, 0];
760     A_2=[w_2_3D(1, 1)*sind(beta_2_3D(1, 1)), w_2_3D(1, 1)*...
761         cosd(beta_2_3D(1, 1))];
762     A_3=[u_3D(1, 1), 0];
763     A_4=[0, 0];
764     A_5=[w_1_3D(1, 1)*sind(beta_1_3D(1, 1)), w_1_3D(1, 1)*...
765         cosd(beta_1_3D(1, 1))];
766     A_6=[u_3D(1, 1), 0];
767
768     x_raiz=[A_1(1) A_2(1) A_3(1) A_4(1)];

```

```

769 y_raiz=[A_1(2) A_2(2) A_3(2) A_1(2)];
770 z_raiz=[A_4(1) A_5(1) A_6(1) A_4(1)];
771 t_raiz=[A_4(2) A_5(2) A_6(2) A_4(2)];
772
773 % posicion media
774 %
775 C_1=[0,0];
776 C_2=[w_2_3D(3,1)*sind(beta_2_3D(3,1)),w_2_3D(3,1)*...
777 cosd(beta_2_3D(3,1))];
778 C_3=[u_3D(3,1),0];
779 C_4=[0,0];
780 C_5=[w_1_3D(3,1)*sind(beta_1_3D(3,1)),w_1_3D(3,1)*...
781 cosd(beta_1_3D(3,1))];
782 C_6=[u_3D(3,1),0];
783
784 x_medio=[C_1(1) C_2(1) C_3(1) C_1(1)];
785 y_medio=[C_1(2) C_2(2) C_3(2) C_1(2)];
786 z_medio=[C_4(1) C_5(1) C_6(1) C_4(1)];
787 t_medio=[C_4(2) C_5(2) C_6(2) C_4(2)];
788
789 % posicion final
790 %
791 E_1=[0,0];
792 E_2=[w_2_3D(5,1)*sind(beta_2_3D(5,1)),w_2_3D(5,1)*...
793 cosd(beta_2_3D(5,1))];
794 E_3=[u_3D(5,1),0];
795 E_4=[0,0];
796 E_5=[w_1_3D(5,1)*sind(beta_1_3D(5,1)),w_1_3D(5,1)*...
797 cosd(beta_1_3D(5,1))];
798 E_6=[u_3D(5,1),0];
799
800 x_cabeza=[E_1(1) E_2(1) E_3(1) E_1(1)];
801 y_cabeza=[E_1(2) E_2(2) E_3(2) E_1(2)];
802 z_cabeza=[E_4(1) E_5(1) E_6(1) E_4(1)];
803 t_cabeza=[E_4(2) E_5(2) E_6(2) E_4(2)];
804
805 plot(app.triangulos_3D,x_raiz,y_raiz,"black",...
806 z_raiz,t_raiz,"black", ...
807 x_medio,y_medio,"red",z_medio,t_medio,"red", ...
808 x_cabeza,y_cabeza,"magenta",z_cabeza,t_cabeza,"magenta");
809 legend(app.triangulos_3D,{'','radio ra z','',...
810 'radio medio','','radio cabeza'},'Location','northeast');
811
812
813 app.escalonamientos.Value=1;
814 app.Mach_Maximo.Value = max(M_w_1);
815 app.Ley.Value='Reacci n constante';
816 app.Ley_2.Value='Reacci n constante';
817 app.Ley_3.Value='Reacci n constante';
818 app.ley_actual.Value = 2;
819
820
821
822 %%% ANGULOS
823

```

```

824     x=r_r_rotor(1)/r_c_rotor(1):(1-r_r_rotor(1)/...
825         r_c_rotor(1))/(app.n_puntos.Value-1):1;
826     alpha_1_g=alpha_1_3D(:,1);
827     alpha_2_g=alpha_2_3D(:,1);
828     beta_1_g=beta_1_3D(:,1);
829     beta_2_g=beta_2_3D(:,1);
830     plot(app.UIAxes6,x,alpha_1_g,"blue",x,alpha_2_g,...
831         "green",x,beta_1_g,"red",x,beta_2_g,"magenta")
832     legend(app.UIAxes6,['alpha 1','alpha 2','Beta 1','Beta 2'],...
833         'Location','northwest');
834
835
836     %%% COEFICIENTES
837
838     x=r_r_rotor(1)/r_c_rotor(1):(1-r_r_rotor(1)/...
839         r_c_rotor(1))/(app.n_puntos.Value-1):1;
840     R_3D_g=R_3D(:,1);
841     phi_3D_g=phi_3D(:,1);
842     psi_3D_g=psi_3D(:,1);
843     plot(app.UIAxes6_2,x,R_3D_g,"blue",x,phi_3D_g,"green",x,psi_3D_g,"red")
844         legend(app.UIAxes6_2,['R',' ',' '],'Location','northwest');
845
846     %%% PERDIDAS
847
848     tabla_perdidas_3D(1,1)=D_m/2-H_rotor(1)/2;
849     tabla_perdidas_3D(2,1)=D_m/2;
850     tabla_perdidas_3D(3,1)=D_m/2+H_rotor(1)/2;
851     tabla_perdidas_3D(1,2)=perdidas_rotor_3D(1,1);
852     tabla_perdidas_3D(2,2)=perdidas_rotor_3D((app.n_puntos.Value+1)/2,1);
853     tabla_perdidas_3D(3,2)=perdidas_rotor_3D(app.n_puntos.Value,1);
854     tabla_perdidas_3D(1,3)=perdidas_estator_3D(1,1);
855     tabla_perdidas_3D(2,3)=perdidas_estator_3D((app.n_puntos.Value+1)/2,1);
856     tabla_perdidas_3D(3,3)=perdidas_estator_3D(app.n_puntos.Value,1);
857     tabla_perdidas_3D(1,4)=W_u_3D;
858     tabla_perdidas_3D(2,4)=W_u_3D;
859     tabla_perdidas_3D(3,4)=W_u_3D;
860     tabla_perdidas_3D(1,5)=eta_TT_3D(1,1);
861     tabla_perdidas_3D(2,5)=eta_TT_3D((app.n_puntos.Value+1)/2,1);
862     tabla_perdidas_3D(3,5)=eta_TT_3D(app.n_puntos.Value,1);
863
864     app.perdidas_3D.Data = tabla_perdidas_3D;
865
866     app.TrabajoespecificoporescalonamientoKJkgEditField.Value = W_u_3D;
867     app.RendimientototaldelescalonamientoEditField.Value...
868     =(tabla_perdidas_3D(1,5)+tabla_perdidas_3D(2,5)...
869     +tabla_perdidas_3D(3,5))/3;
870
871     end
872
873
874
875
876
877
878     end

```

```

879
880      % Button pushed function: exponencial
881      function exponencialButtonPushed(app, event)
882          [alpha_1,alpha_2,alpha_2_prima,alpha_3, ...
883          alpha_3_prima, beta_1,beta_1_prima,beta_2,beta_2_prima, ...
884          c_1,c_2,c_3,c_a,c_p,c_u1,c_u2,coef_R, ...
885          curvatura_estator,curvatura_rotor,D_m, ...
886          desviacion_estator,desviacion_rotor,DF_eq_E,...
887          eta_c,eta_TT,eta_politropico,...
888          gamma,H_estator,H_rotor,Lieblein_e,Lieblein_r,M_w_1,n_esc, ...
889          P_01,P_02,P_03,P_1,P_2,P_3,perdidas_estator,perdidas_rotor, ...
890          psi,r_c,r_c_estator,r_c_rotor,r_r_estator,r_r_rotor, ...
891          rc_total_obtenida,rho_1,rho_2,rho_3,s_estator,s_rotor, suma_perdidas,...
892          T_01,T_02,T_03,T_03ss,T_1,T_2,T_2s,T_3,T_total_final,u,w_1,w_2, ...
893          W_u,zeta_a_e,zeta_a_r,zeta_e,zeta_p_e,zeta_p_r,zeta_r, ...
894          zeta_s_e,zeta_s_r] ...
895          = f_caso_2D(app.T_01_primer.Value,app.P_01_primer.Value,...
896          app.sigma.Value,app.n.Value,app.m_dot.Value,app.DF_eq_R.Value,...
897          app.R.Value,app.phi.Value,app.rc_total.Value);
898
899          [a,b,c_a1_3D,c_a2_3D,W_u_3D,u_3D,...
900          c_u1_3D,c_u2_3D, c_1_3D,c_2_3D, ...
901          w_u1_3D,w_u2_3D,beta_1_3D,beta_2_3D, alpha_1_3D,alpha_2_3D,R_3D, ...
902          T_1_3D,M_c_3D,M_r_3D,DF_eq_R_3D,DF_eq_E_3D, ...
903          zeta_p_R_3D,zeta_p_E_3D, perdidas_rotor_3D,perdidas_estator_3D, ...
904          perdidas_3D,phi_3D,psi_3D,W_u_3D_is,eta_TT_3D,w_1_3D,w_2_3D] ...
905          = f_exponencial(c_u1,c_u2,D_m,c_a,W_u,n_esc,u,r_r_rotor,r_c_rotor, ...
906          T_01,c_p,gamma,coef_R,app.sigma.Value,zeta_s_r,...
907          zeta_a_r,zeta_s_e,zeta_a_e,app.n_puntos.Value)
908
909      %%% TRIANGULOS 3D
910
911      % posicion inicial
912
913      A_1=[0,0];
914      A_2=[w_2_3D(1,1)*sind(beta_2_3D(1,1)),...
915          w_2_3D(1,1)*cosd(beta_2_3D(1,1))];
916      A_3=[u_3D(1,1),0];
917      A_4=[0,0];
918      A_5=[w_1_3D(1,1)*sind(beta_1_3D(1,1)),...
919          w_1_3D(1,1)*cosd(beta_1_3D(1,1))];
920      A_6=[u_3D(1,1),0];
921
922      x_raiz=[A_1(1) A_2(1) A_3(1) A_4(1)];
923      y_raiz=[A_1(2) A_2(2) A_3(2) A_4(2)];
924      z_raiz=[A_5(1) A_6(1) A_4(1)];
925      t_raiz=[A_5(2) A_6(2) A_4(2)];
926
927      % posicion media
928      %
929      C_1=[0,0];
930      C_2=[w_2_3D(3,1)*sind(beta_2_3D(3,1)),...
931          w_2_3D(3,1)*cosd(beta_2_3D(3,1))];
932      C_3=[u_3D(3,1),0];
933      C_4=[0,0];

```

```

934 C_5=[w_1_3D(3,1)*sind(beta_1_3D(3,1)),...
935       w_1_3D(3,1)*cosd(beta_1_3D(3,1))];
936 C_6=[u_3D(3,1),0];
937
938 x_medio=[C_1(1) C_2(1) C_3(1) C_1(1)];
939 y_medio=[C_1(2) C_2(2) C_3(2) C_1(2)];
940 z_medio=[C_4(1) C_5(1) C_6(1) C_4(1)];
941 t_medio=[C_4(2) C_5(2) C_6(2) C_4(2)];
942
943 % posicion final
944 %
945 E_1=[0,0];
946 E_2=[w_2_3D(5,1)*sind(beta_2_3D(5,1)),...
947       w_2_3D(5,1)*cosd(beta_2_3D(5,1))];
948 E_3=[u_3D(5,1),0];
949 E_4=[0,0];
950 E_5=[w_1_3D(5,1)*sind(beta_1_3D(5,1)),...
951       w_1_3D(5,1)*cosd(beta_1_3D(5,1))];
952 E_6=[u_3D(5,1),0];
953
954 x_cabeza=[E_1(1) E_2(1) E_3(1) E_1(1)];
955 y_cabeza=[E_1(2) E_2(2) E_3(2) E_1(2)];
956 z_cabeza=[E_4(1) E_5(1) E_6(1) E_4(1)];
957 t_cabeza=[E_4(2) E_5(2) E_6(2) E_4(2)];
958
959 plot(app.triangulos_3D,x_raiz,y_raiz,...
960      "black",z_raiz,t_raiz,"black", ...
961      x_medio,y_medio,"red",z_medio,t_medio,"red", ...
962      x_cabeza,y_cabeza,"magenta",z_cabeza,t_cabeza,"magenta");
963 legend(app.triangulos_3D,{'','radio ra z',...
964      ','radio medio','','radio cabeza'],'Location','northeast');
965
966 app.escalonamientos.Value=1;
967 app.Mach_Maximo.Value = max(M_w_1);
968 app.Ley.Value='Exponencial';
969 app.Ley_2.Value='Exponencial';
970 app.Ley_3.Value='Exponencial';
971 app.ley_actual.Value = 3;
972
973
974
975 %% ANGULOS
976
977 x=r_r_rotor(1)/r_c_rotor(1):(1-r_r_rotor(1)/...
978     r_c_rotor(1))/(app.n_puntos.Value-1):1;
979 alpha_1_g=alpha_1_3D(:,1);
980 alpha_2_g=alpha_2_3D(:,1);
981 beta_1_g=beta_1_3D(:,1);
982 beta_2_g=beta_2_3D(:,1);
983 plot(app.UIAxes6,x,alpha_1_g,"blue",...
984      x,alpha_2_g,"green",x,beta_1_g,"red",x,beta_2_g,"magenta")
985 legend(app.UIAxes6,{'alpha 1','alpha 2',...
986      'Beta 1','Beta 2'],'Location','northwest');
987
988

```

```

989      %%% COEFICIENTES
990
991      x=r_r_rotor(1)/r_c_rotor(1):...
992          (1-r_r_rotor(1)/r_c_rotor(1))/(app.n_puntos.Value-1):1;
993      R_3D_g=R_3D(:,1);
994      phi_3D_g=phi_3D(:,1);
995      psi_3D_g=psi_3D(:,1);
996      plot(app.UIAxes6_2,x,R_3D_g,"blue",x,phi_3D_g,"green",x,psi_3D_g,"red")
997          legend(app.UIAxes6_2,{'R','phi','psi'},'Location','northwest');
998
999      %%% PERDIDAS
1000
1001      tabla_perdidas_3D(1,1)=D_m/2-H_rotor(1)/2;
1002      tabla_perdidas_3D(2,1)=D_m/2;
1003      tabla_perdidas_3D(3,1)=D_m/2+H_rotor(1)/2;
1004      tabla_perdidas_3D(1,2)=perdidas_rotor_3D(1,1);
1005      tabla_perdidas_3D(2,2)=perdidas_rotor_3D((app.n_puntos.Value+1)/2,1);
1006      tabla_perdidas_3D(3,2)=perdidas_rotor_3D(app.n_puntos.Value,1);
1007      tabla_perdidas_3D(1,3)=perdidas_estator_3D(1,1);
1008      tabla_perdidas_3D(2,3)=perdidas_estator_3D((app.n_puntos.Value+1)/2,1);
1009      tabla_perdidas_3D(3,3)=perdidas_estator_3D(app.n_puntos.Value,1);
1010      tabla_perdidas_3D(1,4)=W_u_3D;
1011      tabla_perdidas_3D(2,4)=W_u_3D;
1012      tabla_perdidas_3D(3,4)=W_u_3D;
1013      tabla_perdidas_3D(1,5)=eta_TT_3D(1,1);
1014      tabla_perdidas_3D(2,5)=eta_TT_3D((app.n_puntos.Value+1)/2,1);
1015      tabla_perdidas_3D(3,5)=eta_TT_3D(app.n_puntos.Value,1);
1016
1017      app.perdidas_3D.Data = tabla_perdidas_3D;
1018
1019      app.TrabajoespecificoporescalonamientoKJkgEditField.Value = W_u_3D;
1020      app.RendimientototaldelescalonamientoEditField.Value=...
1021      (tabla_perdidas_3D(1,5)+tabla_perdidas_3D(2,5)+tabla_perdidas_3D(3,5))/3;
1022
1023      end
1024
1025      % Button pushed function: SiguienteescalonamientoButton
1026      function SiguienteescalonamientoButtonPushed(app, event)
1027  if app.ley_actual.Value == 1
1028
1029
1030
1031      if app.escalonamientos.Value >= app.NdeescalonamientosEditField.Value
1032
1033
1034
1035          app.escalonamientos.Value = app.escalonamientos.Value;
1036      else
1037
1038          app.escalonamientos.Value = app.escalonamientos.Value + 1;
1039
1040          [alpha_1,alpha_2,alpha_2_prima,alpha_3, ...
1041          alpha_3_prima, beta_1,beta_1_prima,beta_2,beta_2_prima, ...
1042          c_1,c_2,c_3,c_a,c_p,c_u1,c_u2,coef_R, ...
1043          curvatura_estator,curvatura_rotor,D_m, ...

```

```

1044 desviacion_estator,desviacion_rotor,...
1045 DF_eq_E,eta_c,eta_TT,eta_politropico,...
1046 gamma,H_estator,H_rotor,Lieblein_e,Lieblein_r,M_w_1,n_esc,...
1047 P_01,P_02,P_03,P_1,P_2,P_3,perdidas_estator,perdidas_rotor,...
1048 psi,r_c,r_c_estator,r_c_rotor,r_r_estator,r_r_rotor,...
1049 rc_total_obtenida,rho_1,rho_2,rho_3,s_estator,s_rotor,suma_perdidas,...
1050 T_01,T_02,T_03,T_03ss,T_1,T_2,T_2s,T_3,T_total_final,u,w_1,w_2,...
1051 W_u,zeta_a_e,zeta_a_r,zeta_e,zeta_p_e,zeta_p_r,zeta_r,...
1052 zeta_s_e,zeta_s_r] ...
1053 = f_caso_2D(app.T_01_primer.Value,app.P_01_primer.Value,...
1054 app.sigma.Value,app.n.Value,app.m_dot.Value,app.DF_eq_R.Value,...
1055 app.R.Value,app.phi.Value,app.rc_total.Value);
1056
1057 [b_1,b_2,c_a_3D,W_u_3D,u_3D,c_u1_3D,c_u2_3D,c_1_3D,c_2_3D,...
1058 w_u1_3D,w_u2_3D,beta_1_3D,beta_2_3D,alpha_1_3D,alpha_2_3D,R_3D,...
1059 T_1_3D,M_c_3D,M_r_3D,DF_eq_R_3D,DF_eq_E_3D,...
1060 zeta_p_R_3D,zeta_p_E_3D,perdidas_rotor_3D,perdidas_estator_3D,...
1061 perdidas_3D,phi_3D,psi_3D,W_u_3D_is,eta_TT_3D,w_1_3D,w_2_3D] ...
1062 = f_torbellino_libre(c_u1,c_u2,D_m,c_a,W_u,n_esc,u,r_r_rotor,r_c_rotor,...
1063 T_01,c_p,gamma,coef_R,app.sigma.Value,zeta_s_r,...
1064 zeta_a_r,zeta_s_e,zeta_a_e,app.n_puntos.Value);
1065
1066
1067 %%% TRIANGULOS 3D
1068
1069 % posicion inicial
1070
1071 A_1=[0,0];
1072 A_2=[w_2_3D(1,app.escalonamientos.Value)*...
1073     sind(beta_2_3D(1,app.escalonamientos.Value)),...
1074     w_2_3D(1,app.escalonamientos.Value)*...
1075     cosd(beta_2_3D(1,app.escalonamientos.Value))];
1076 A_3=[u_3D(1,app.escalonamientos.Value),0];
1077 A_4=[0,0];
1078 A_5=[w_1_3D(1,app.escalonamientos.Value)*...
1079     sind(beta_1_3D(1,app.escalonamientos.Value)),...
1080     w_1_3D(1,app.escalonamientos.Value)*...
1081     cosd(beta_1_3D(1,app.escalonamientos.Value))];
1082 A_6=[u_3D(1,app.escalonamientos.Value),0];
1083
1084 x_raiz=[A_1(1) A_2(1) A_3(1) A_4(1)];
1085 y_raiz=[A_1(2) A_2(2) A_3(2) A_4(2)];
1086 z_raiz=[A_5(1) A_6(1) A_4(1)];
1087 t_raiz=[A_5(2) A_6(2) A_4(2)];
1088 %
1089 %
1090 % posicion media
1091 %
1092 C_1=[0,0];
1093 C_2=[w_2_3D(3,app.escalonamientos.Value)*sind...
1094     (beta_2_3D(3,app.escalonamientos.Value)),...
1095     w_2_3D(3,app.escalonamientos.Value)*...
1096     cosd(beta_2_3D(3,app.escalonamientos.Value))];
1097 C_3=[u_3D(3,app.escalonamientos.Value),0];
1098 C_4=[0,0];

```



```

1099 C_5=[w_1_3D(3,app.escalonamientos.Value)*...
1100     sind(beta_1_3D(3,app.escalonamientos.Value)),...
1101     w_1_3D(3,app.escalonamientos.Value)*...
1102     cosd(beta_1_3D(3,app.escalonamientos.Value))];
1103 C_6=[u_3D(3,app.escalonamientos.Value),0];
1104
1105 x_medio=[C_1(1) C_2(1) C_3(1) C_1(1)];
1106 y_medio=[C_1(2) C_2(2) C_3(2) C_1(2)];
1107 z_medio=[C_4(1) C_5(1) C_6(1) C_4(1)];
1108 t_medio=[C_4(2) C_5(2) C_6(2) C_4(2)];
1109 %
1110 %
1111 %     % posicion final
1112 %
1113 E_1=[0,0];
1114 E_2=[w_2_3D(5,app.escalonamientos.Value)*...
1115     sind(beta_2_3D(5,app.escalonamientos.Value)),w_2_3D(5,...
1116     app.escalonamientos.Value)*cosd(beta_2_3D(5,...
1117     app.escalonamientos.Value))];
1118 E_3=[u_3D(5,app.escalonamientos.Value),0];
1119 E_4=[0,0];
1120 E_5=[w_1_3D(5,app.escalonamientos.Value)*sind(beta_1_3D(5,...
1121     app.escalonamientos.Value)),w_1_3D(5,...
1122     app.escalonamientos.Value)*cosd(beta_1_3D(5,...
1123     app.escalonamientos.Value))];
1124 E_6=[u_3D(5,app.escalonamientos.Value),0];
1125
1126 x_cabeza=[E_1(1) E_2(1) E_3(1) E_1(1)];
1127 y_cabeza=[E_1(2) E_2(2) E_3(2) E_1(2)];
1128 z_cabeza=[E_4(1) E_5(1) E_6(1) E_4(1)];
1129 t_cabeza=[E_4(2) E_5(2) E_6(2) E_4(2)];
1130
1131 plot(app.triangulos_3D,x_raiz,...
1132     y_raiz,"black",z_raiz,t_raiz,"black", ...
1133     x_medio,y_medio,"red",z_medio,t_medio,"red", ...
1134     x_cabeza,y_cabeza,"magenta",z_cabeza,t_cabeza,"magenta");
1135 legend(app.triangulos_3D',{'','radio ra z','',...
1136     'radio medio','','radio cabeza'],'Location','northeast');
1137
1138
1139
1140 %%% ANGULOS
1141
1142 x=r_r_rotor(app.escalonamientos.Value)/r_c_rotor...
1143     (app.escalonamientos.Value):(1-r_r_rotor...
1144     (app.escalonamientos.Value)/r_c_rotor...
1145     (app.escalonamientos.Value))/(app.n_puntos.Value-1):1;
1146 alpha_1_g=alpha_1_3D(:,app.escalonamientos.Value);
1147 alpha_2_g=alpha_2_3D(:,app.escalonamientos.Value);
1148 beta_1_g=beta_1_3D(:,app.escalonamientos.Value);
1149 beta_2_g=beta_2_3D(:,app.escalonamientos.Value);
1150 plot(app.UIAxes6,x,alpha_1_g,"blue",x,alpha_2_g,...
1151     "green",x,beta_1_g,"red",x,beta_2_g,"magenta")
1152 legend(app.UIAxes6,{'alpha 1','alpha 2','Beta 1',...
1153     'Beta 2'],'Location','northwest');

```

```

1154
1155
1156
1157     %%% COEFICIENTES
1158
1159         x=r_r_rotor(app.escalonamientos.Value)/...
1160             r_c_rotor(app.escalonamientos.Value):...
1161             (1-r_r_rotor(app.escalonamientos.Value)/...
1162 r_c_rotor(app.escalonamientos.Value))/(app.n_puntos.Value-1):1;
1163         R_3D_g=R_3D(:,app.escalonamientos.Value);
1164         phi_3D_g=phi_3D(:,app.escalonamientos.Value);
1165         psi_3D_g=psi_3D(:,app.escalonamientos.Value);
1166 plot(app.UIAxes6_2,x,R_3D_g,"blue",x,phi_3D_g,"green",x,psi_3D_g,"red")
1167         legend(app.UIAxes6_2,{'R',' ',' '},'Location','northwest');
1168
1169
1170     %%% PERDIDAS
1171
1172         tabla_perdidas_3D(1,1)=D_m/2-H_rotor(app.escalonamientos.Value)/2;
1173         tabla_perdidas_3D(2,1)=D_m/2;
1174         tabla_perdidas_3D(3,1)=D_m/2+H_rotor(app.escalonamientos.Value)/2;
1175         tabla_perdidas_3D(1,2)=perdidas_rotor_3D(1,app.escalonamientos.Value);
1176         tabla_perdidas_3D(2,2)=perdidas_rotor_3D...
1177             ((app.n_puntos.Value+1)/2,app.escalonamientos.Value);
1178         tabla_perdidas_3D(3,2)=perdidas_rotor_3D...
1179             (app.n_puntos.Value,app.escalonamientos.Value);
1180         tabla_perdidas_3D(1,3)=perdidas_estator_3D...
1181             (1,app.escalonamientos.Value);
1182         tabla_perdidas_3D(2,3)=perdidas_estator_3D...
1183             ((app.n_puntos.Value+1)/2,app.escalonamientos.Value);
1184         tabla_perdidas_3D(3,3)=perdidas_estator_3D...
1185             (app.n_puntos.Value,app.escalonamientos.Value);
1186         tabla_perdidas_3D(1,4)=W_u_3D;
1187         tabla_perdidas_3D(2,4)=W_u_3D;
1188         tabla_perdidas_3D(3,4)=W_u_3D;
1189         tabla_perdidas_3D(1,5)=eta_TT_3D(1,app.escalonamientos.Value);
1190         tabla_perdidas_3D(2,5)=eta_TT_3D...
1191             ((app.n_puntos.Value+1)/2,app.escalonamientos.Value);
1192         tabla_perdidas_3D(3,5)=eta_TT_30...
1193             (app.n_puntos.Value,app.escalonamientos.Value);
1194
1195         app.perdidas_3D.Data = tabla_perdidas_3D;
1196
1197         app.RendimientototaldelescalonamientoEditField.Value=...
1198         (tabla_perdidas_3D(1,5)+tabla_perdidas_3D(2,5)+tabla_perdidas_3D(3,5))/3;
1199
1200
1201
1202
1203         end
1204
1205
1206 elseif app.ley_actual.Value == 2
1207
1208         if app.escalonamientos.Value >= app.NdeescalonamientosEditField.Value

```

```

1209
1210
1211
1212         app.escalonamientos.Value = app.escalonamientos.Value;
1213     else
1214
1215         app.escalonamientos.Value = app.escalonamientos.Value + 1;
1216
1217         [alpha_1,alpha_2,alpha_2_prima,alpha_3, ...
1218     alpha_3_prima, beta_1,beta_1_prima,beta_2,beta_2_prima, ...
1219     c_1,c_2,c_3,c_a,c_p,c_u1,c_u2,coef_R, ...
1220     curvatura_estator,curvatura_rotor,D_m, ...
1221     desviacion_estator,desviacion_rotor,...
1222     DF_eq_E,eta_c,eta_TT,eta_politropico,...
1223     gamma,H_estator,H_rotor,Lieblein_e,Lieblein_r,M_w_1,n_esc, ...
1224     P_01,P_02,P_03,P_1,P_2,P_3,perdidas_estator,perdidas_rotor, ...
1225     psi,r_c,r_c_estator,r_c_rotor,r_r_estator,r_r_rotor, ...
1226     rc_total_obtenida,rho_1,rho_2,rho_3,s_estator,s_rotor, suma_perdidas,...
1227     T_01,T_02,T_03,T_03ss,T_1,T_2,T_2s,T_3,T_total_final,u,w_1,w_2, ...
1228     W_u,zeta_a_e,zeta_a_r,zeta_e,zeta_p_e,zeta_p_r,zeta_r, ...
1229     zeta_s_e,zeta_s_r] ...
1230     = f_caso_2D(app.T_01_primer.Value,app.P_01_primer.Value,...
1231     app.sigma.Value,app.n.Value,app.m_dot.Value,app.DF_eq_R.Value,...
1232     app.R.Value,app.phi.Value,app.rc_total.Value);
1233
1234     [a,b,c_a1_3D,c_a2_3D,W_u_3D,u_3D,c_u1_3D,c_u2_3D, c_1_3D,c_2_3D, ...
1235     w_u1_3D,w_u2_3D,beta_1_3D,beta_2_3D, alpha_1_3D,alpha_2_3D,R_3D, ...
1236     T_1_3D,M_c_3D,M_r_3D,DF_eq_R_3D,DF_eq_E_3D, ...
1237     zeta_p_R_3D,zeta_p_E_3D, perdidas_rotor_3D,perdidas_estator_3D, ...
1238     perdidas_3D,phi_3D,psi_3D,W_u_3D_is,eta_TT_3D,w_1_3D,w_2_3D] ...
1239     = f_reaccion_constante(c_u1,c_u2,D_m,c_a,W_u,...
1240     n_esc,u,r_r_rotor,r_c_rotor, ...
1241     T_01,c_p,gamma,coef_R,app.sigma.Value,...
1242     zeta_s_r,zeta_a_r,zeta_s_e,zeta_a_e,app.n_puntos.Value)
1243
1244
1245     %% TRIANGULOS 3D
1246
1247     % posicion inicial
1248
1249     A_1=[0,0];
1250     A_2=[w_2_3D(1,app.escalonamientos.Value)*...
1251         sind(beta_2_3D(1,app.escalonamientos.Value)),...
1252         w_2_3D(1,app.escalonamientos.Value)*...
1253         cosd(beta_2_3D(1,app.escalonamientos.Value))];
1254     A_3=[u_3D(1,app.escalonamientos.Value),0];
1255     A_4=[0,0];
1256     A_5=[w_1_3D(1,app.escalonamientos.Value)*...
1257         sind(beta_1_3D(1,app.escalonamientos.Value)),...
1258         w_1_3D(1,app.escalonamientos.Value)*...
1259         cosd(beta_1_3D(1,app.escalonamientos.Value))];
1260     A_6=[u_3D(1,app.escalonamientos.Value),0];
1261
1262     x_raiz=[A_1(1) A_2(1) A_3(1) A_4(1)];
1263     y_raiz=[A_1(2) A_2(2) A_3(2) A_4(2)];

```

```

1264 z_raiz=[A_4(1) A_5(1) A_6(1) A_4(1)];
1265 t_raiz=[A_4(2) A_5(2) A_6(2) A_4(2)];
1266 %
1267 %
1268 %      % posicion media
1269 %
1270 C_1=[0,0];
1271 C_2=[w_2_3D(3,app.escalonamientos.Value)*sind(beta_2_3D(3,...
1272     app.escalonamientos.Value)),w_2_3D(3,...
1273     app.escalonamientos.Value)*cosd(beta_2_3D(3,...
1274     app.escalonamientos.Value))];
1275 C_3=[u_3D(3,app.escalonamientos.Value),0];
1276 C_4=[0,0];
1277 C_5=[w_1_3D(3,app.escalonamientos.Value)*sind(beta_1_3D(3,...
1278     app.escalonamientos.Value)),w_1_3D(3,...
1279     app.escalonamientos.Value)*cosd(beta_1_3D(3,...
1280     app.escalonamientos.Value))];
1281 C_6=[u_3D(3,app.escalonamientos.Value),0];
1282
1283 x_medio=[C_1(1) C_2(1) C_3(1) C_1(1)];
1284 y_medio=[C_1(2) C_2(2) C_3(2) C_1(2)];
1285 z_medio=[C_4(1) C_5(1) C_6(1) C_4(1)];
1286 t_medio=[C_4(2) C_5(2) C_6(2) C_4(2)];
1287 %
1288 %
1289 %      % posicion final
1290 %
1291 E_1=[0,0];
1292 E_2=[w_2_3D(5,app.escalonamientos.Value)*sind(beta_2_3D(5,...
1293     app.escalonamientos.Value)),w_2_3D(5,...
1294     app.escalonamientos.Value)*cosd(beta_2_3D(5,...
1295     app.escalonamientos.Value))];
1296 E_3=[u_3D(5,app.escalonamientos.Value),0];
1297 E_4=[0,0];
1298 E_5=[w_1_3D(5,app.escalonamientos.Value)*sind(beta_1_3D(5,...
1299     app.escalonamientos.Value)),w_1_3D(5,...
1300     app.escalonamientos.Value)*cosd(beta_1_3D(5,...
1301     app.escalonamientos.Value))];
1302 E_6=[u_3D(5,app.escalonamientos.Value),0];
1303
1304 x_cabeza=[E_1(1) E_2(1) E_3(1) E_1(1)];
1305 y_cabeza=[E_1(2) E_2(2) E_3(2) E_1(2)];
1306 z_cabeza=[E_4(1) E_5(1) E_6(1) E_4(1)];
1307 t_cabeza=[E_4(2) E_5(2) E_6(2) E_4(2)];
1308
1309 plot(app.triangulos_3D,x_raiz,y_raiz,"black",z_raiz,t_raiz,"black", ...
1310     x_medio,y_medio,"red",z_medio,t_medio,"red", ...
1311     x_cabeza,y_cabeza,"magenta",z_cabeza,t_cabeza,"magenta");
1312
1313 %
1314 %      % posicion final
1315 %
1316 E_1=[0,0];
1317 E_2=[w_2_3D(5,app.escalonamientos.Value)*...
1318     sind(beta_2_3D(5,app.escalonamientos.Value)),...

```

```

1319         w_2_3D(5,app.escalonamientos.Value)*...
1320         cosd(beta_2_3D(5,app.escalonamientos.Value))];
1321     E_3=[u_3D(5,app.escalonamientos.Value),0];
1322     E_4=[0,0];
1323     E_5=[w_1_3D(5,app.escalonamientos.Value)*...
1324         sind(beta_1_3D(5,app.escalonamientos.Value)),...
1325         w_1_3D(5,app.escalonamientos.Value)*...
1326         cosd(beta_1_3D(5,app.escalonamientos.Value))];
1327     E_6=[u_3D(5,app.escalonamientos.Value),0];
1328
1329     x_cabeza=[E_1(1) E_2(1) E_3(1) E_1(1)];
1330     y_cabeza=[E_1(2) E_2(2) E_3(2) E_1(2)];
1331     z_cabeza=[E_4(1) E_5(1) E_6(1) E_4(1)];
1332     t_cabeza=[E_4(2) E_5(2) E_6(2) E_4(2)];
1333
1334     plot(app.triangulos_3D,x_raiz,y_raiz,"black",z_raiz,t_raiz,"black", ...
1335         x_medio,y_medio,"red",z_medio,t_medio,"red", ...
1336         x_cabeza,y_cabeza,"magenta",z_cabeza,t_cabeza,"magenta");
1337     legend(app.triangulos_3D,{'','radio ra z','',...
1338         'radio medio','','radio cabeza'},'Location','northeast');
1339
1340
1341     %%% ANGULOS
1342
1343     x=r_r_rotor(app.escalonamientos.Value)/...
1344         r_c_rotor(app.escalonamientos.Value):(1-r_r_rotor...
1345         (app.escalonamientos.Value)/r_c_rotor...
1346         (app.escalonamientos.Value))/(app.n_puntos.Value-1):1;
1347     alpha_1_g=alpha_1_3D(:,app.escalonamientos.Value);
1348     alpha_2_g=alpha_2_3D(:,app.escalonamientos.Value);
1349     beta_1_g=beta_1_3D(:,app.escalonamientos.Value);
1350     beta_2_g=beta_2_3D(:,app.escalonamientos.Value);
1351     plot(app.UIAxes6,x,alpha_1_g,"blue",x,alpha_2_g,...
1352         "green",x,beta_1_g,"red",x,beta_2_g,"magenta")
1353     legend(app.UIAxes6,{'alpha 1','alpha 2','Beta 1',...
1354         'Beta 2'},'Location','northwest');
1355
1356
1357
1358     %%% COEFICIENTES
1359
1360     x=r_r_rotor(app.escalonamientos.Value)/r_c_rotor...
1361         (app.escalonamientos.Value):(1-r_r_rotor...
1362         (app.escalonamientos.Value)/r_c_rotor...
1363         (app.escalonamientos.Value))/(app.n_puntos.Value-1):1;
1364     R_3D_g=R_3D(:,app.escalonamientos.Value);
1365     phi_3D_g=phi_3D(:,app.escalonamientos.Value);
1366     psi_3D_g=psi_3D(:,app.escalonamientos.Value);
1367     plot(app.UIAxes6_2,x,R_3D_g,"blue",x,phi_3D_g,...
1368         "green",x,psi_3D_g,"red")
1369     legend(app.UIAxes6_2,{'R',' ',' '},'Location','northwest');
1370
1371
1372     %%% PERDIDAS
1373

```

```

1374     tabla_perdidas_3D(1,1)=D_m/2-H_rotor(app.escalonamientos.Value)/2;
1375         tabla_perdidas_3D(2,1)=D_m/2;
1376     tabla_perdidas_3D(3,1)=D_m/2+H_rotor(app.escalonamientos.Value)/2;
1377     tabla_perdidas_3D(1,2)=perdidas_rotor_3D(1,app.escalonamientos.Value);
1378         tabla_perdidas_3D(2,2)=...
1379             perdidas_rotor_3D...
1380             ((app.n_puntos.Value+1)/2,app.escalonamientos.Value);
1381     tabla_perdidas_3D(3,2)=perdidas_rotor_3D...
1382         (app.n_puntos.Value,app.escalonamientos.Value);
1383     tabla_perdidas_3D(1,3)=perdidas_estator_3D...
1384         (1,app.escalonamientos.Value);
1385     tabla_perdidas_3D(2,3)=perdidas_estator_3D...
1386         ((app.n_puntos.Value+1)/2,app.escalonamientos.Value);
1387     tabla_perdidas_3D(3,3)=perdidas_estator_3D...
1388         (app.n_puntos.Value,app.escalonamientos.Value);
1389     tabla_perdidas_3D(1,4)=W_u_3D;
1390     tabla_perdidas_3D(2,4)=W_u_3D;
1391     tabla_perdidas_3D(3,4)=W_u_3D;
1392     tabla_perdidas_3D(1,5)=eta_TT_3D(1,app.escalonamientos.Value);
1393     tabla_perdidas_3D(2,5)=...
1394     eta_TT_3D((app.n_puntos.Value+1)/2,app.escalonamientos.Value);
1395     tabla_perdidas_3D(3,5)=...
1396     eta_TT_3D(app.n_puntos.Value,app.escalonamientos.Value);
1397
1398     app.perdidas_3D.Data = tabla_perdidas_3D;
1399     app.RendimientototaldelescalonamientoEditField.Value=...
1400     (tabla_perdidas_3D(1,5)+tabla_perdidas_3D(2,5)+tabla_perdidas_3D(3,5))/3;
1401
1402
1403     end
1404
1405 else
1406
1407     if app.escalonamientos.Value >= app.NdeescalonamientosEditField.Value
1408
1409
1410
1411         app.escalonamientos.Value = app.escalonamientos.Value;
1412     else
1413
1414         app.escalonamientos.Value = app.escalonamientos.Value + 1;
1415
1416         [alpha_1,alpha_2,alpha_2_prima,alpha_3, ...
1417         alpha_3_prima, beta_1,beta_1_prima,beta_2,beta_2_prima, ...
1418         c_1,c_2,c_3,c_a,c_p,c_u1,c_u2,coef_R, ...
1419         curvatura_estator,curvatura_rotor,D_m, ...
1420         desviacion_estator,desviacion_rotor,DF_eq_E,...
1421         eta_c,eta_TT,eta_politropico,...
1422         gamma,H_estator,H_rotor,Lieblein_e,Lieblein_r,M_w_1,n_esc, ...
1423         P_01,P_02,P_03,P_1,P_2,P_3,perdidas_estator,perdidas_rotor, ...
1424         psi,r_c,r_c_estator,r_c_rotor,r_r_estator,r_r_rotor, ...
1425         rc_total_obtenida,rho_1,rho_2,rho_3,s_estator,s_rotor,suma_perdidas,...
1426         T_01,T_02,T_03,T_03ss,T_1,T_2,T_2s,T_3,T_total_final,u,w_1,w_2, ...
1427         W_u,zeta_a_e,zeta_a_r,zeta_e,zeta_p_e,zeta_p_r,zeta_r, ...
1428         zeta_s_e,zeta_s_r] ...

```

```

1429 = f_caso_2D(app.T_01_primer.Value,app.P_01_primer.Value,...
1430 app.sigma.Value,app.n.Value,app.m_dot.Value,app.DF_eq_R.Value,...
1431 app.R.Value,app.phi.Value,app.rc_total.Value);
1432
1433 [a,b,c_a1_3D,c_a2_3D,W_u_3D,u_3D,c_u1_3D,c_u2_3D,c_1_3D,c_2_3D,...
1434 w_u1_3D,w_u2_3D,beta_1_3D,beta_2_3D,alpha_1_3D,alpha_2_3D,R_3D,...
1435 T_1_3D,M_c_3D,M_r_3D,DF_eq_R_3D,DF_eq_E_3D,...
1436 zeta_p_R_3D,zeta_p_E_3D,perdidas_rotor_3D,perdidas_estator_3D,...
1437 perdidas_3D,phi_3D,psi_3D,W_u_3D_is,eta_TT_3D,w_1_3D,w_2_3D] ...
1438 = f_exponencial(c_u1,c_u2,D_m,c_a,W_u,n_esc,u,r_r_rotor,r_c_rotor,...
1439 T_01,c_p,gamma,coef_R,app.sigma.Value,zeta_s_r,...
1440 zeta_a_r,zeta_s_e,zeta_a_e,app.n_puntos.Value)
1441
1442
1443 %% TRIANGULOS 3D
1444
1445 % posicion inicial
1446
1447 A_1=[0,0];
1448 A_2=[w_2_3D(1,app.escalonamientos.Value)*...
1449     sind(beta_2_3D(1,app.escalonamientos.Value)),...
1450     w_2_3D(1,app.escalonamientos.Value)*cosd...
1451     (beta_2_3D(1,app.escalonamientos.Value))];
1452 A_3=[u_3D(1,app.escalonamientos.Value),0];
1453 A_4=[0,0];
1454 A_5=[w_1_3D(1,app.escalonamientos.Value)*sind...
1455     (beta_1_3D(1,app.escalonamientos.Value)),...
1456     w_1_3D(1,app.escalonamientos.Value)*cosd...
1457     (beta_1_3D(1,app.escalonamientos.Value))];
1458 A_6=[u_3D(1,app.escalonamientos.Value),0];
1459
1460 x_raiz=[A_1(1) A_2(1) A_3(1) A_4(1)];
1461 y_raiz=[A_1(2) A_2(2) A_3(2) A_4(2)];
1462 z_raiz=[A_5(1) A_6(1) A_4(1)];
1463 t_raiz=[A_5(2) A_6(2) A_4(2)];
1464 %
1465 %
1466 % % posicion media
1467 %
1468 C_1=[0,0];
1469 C_2=[w_2_3D(3,app.escalonamientos.Value)*sind(beta_2_3D(3,...
1470     app.escalonamientos.Value)),w_2_3D(3,...
1471     app.escalonamientos.Value)*cosd(beta_2_3D(3,...
1472     app.escalonamientos.Value))];
1473 C_3=[u_3D(3,app.escalonamientos.Value),0];
1474 C_4=[0,0];
1475 C_5=[w_1_3D(3,app.escalonamientos.Value)*sind(beta_1_3D(3,...
1476     app.escalonamientos.Value)),w_1_3D(3,...
1477     app.escalonamientos.Value)*cosd(beta_1_3D(3,...
1478     app.escalonamientos.Value))];
1479 C_6=[u_3D(3,app.escalonamientos.Value),0];
1480
1481 x_medio=[C_1(1) C_2(1) C_3(1) C_4(1)];
1482 y_medio=[C_1(2) C_2(2) C_3(2) C_4(2)];
1483 z_medio=[C_5(1) C_6(1) C_4(1)];

```



```

1484         t_medio=[C_4(2) C_5(2) C_6(2) C_4(2)];
1485 %
1486 %
1487 %         % posicion final
1488 %
1489         E_1=[0,0];
1490         E_2=[w_2_3D(5,app.escalonamientos.Value)*...
1491             sind(beta_2_3D(5,app.escalonamientos.Value)),...
1492             w_2_3D(5,app.escalonamientos.Value)*...
1493             cosd(beta_2_3D(5,app.escalonamientos.Value))];
1494         E_3=[u_3D(5,app.escalonamientos.Value),0];
1495         E_4=[0,0];
1496         E_5=[w_1_3D(5,app.escalonamientos.Value)*sind(beta_1_3D(5,...
1497             app.escalonamientos.Value)),w_1_3D(5,...
1498             app.escalonamientos.Value)*cosd(beta_1_3D(5,...
1499             app.escalonamientos.Value))];
1500         E_6=[u_3D(5,app.escalonamientos.Value),0];
1501
1502         x_cabeza=[E_1(1) E_2(1) E_3(1) E_4(1)];
1503         y_cabeza=[E_1(2) E_2(2) E_3(2) E_4(2)];
1504         z_cabeza=[E_5(1) E_6(1) E_4(1)];
1505         t_cabeza=[E_5(2) E_6(2) E_4(2)];
1506
1507         plot(app.triangulos_3D,x_raiz,y_raiz,"black",...
1508             z_raiz,t_raiz,"black", ...
1509             x_medio,y_medio,"red",z_medio,t_medio,"red", ...
1510             x_cabeza,y_cabeza,"magenta",z_cabeza,t_cabeza,"magenta");
1511         legend(app.triangulos_3D,{'','radio ra z',...
1512             '', 'radio medio', '', 'radio cabeza'}, 'Location','northeast');
1513
1514
1515         %%% ANGULOS
1516
1517         x=r_r_rotor(app.escalonamientos.Value)/r_c_rotor...
1518             (app.escalonamientos.Value):(1-r_r_rotor(1)/...
1519             r_c_rotor(1))/(app.n_puntos.Value-1):1;
1520         alpha_1_g=alpha_1_3D(:,app.escalonamientos.Value);
1521         alpha_2_g=alpha_2_3D(:,app.escalonamientos.Value);
1522         beta_1_g=beta_1_3D(:,app.escalonamientos.Value);
1523         beta_2_g=beta_2_3D(:,app.escalonamientos.Value);
1524         plot(app.UIAxes6,x,alpha_1_g,"blue",x,alpha_2_g,...
1525             "green",x,beta_1_g,"red",x,beta_2_g,"magenta")
1526         legend(app.UIAxes6,{'alpha 1','alpha 2',...
1527             'Beta 1','Beta 2'}, 'Location','northwest');
1528
1529
1530
1531         %%% COEFICIENTES
1532
1533         x=r_r_rotor(app.escalonamientos.Value)/r_c_rotor...
1534             (app.escalonamientos.Value):(1-r_r_rotor...
1535             (app.escalonamientos.Value)/r_c_rotor...
1536             (app.escalonamientos.Value))/(app.n_puntos.Value-1):1;
1537         R_3D_g=R_3D(:,app.escalonamientos.Value);
1538         phi_3D_g=phi_3D(:,app.escalonamientos.Value);

```



```

1539     psi_3D_g=psi_3D(:, app.escalonamientos.Value);
1540     plot(app.UIAxes6_2,x,R_3D_g,"blue",x,...
1541         phi_3D_g,"green",x,psi_3D_g,"red")
1542         legend(app.UIAxes6_2,{'R',' ',' '},'Location','northwest');
1543
1544
1545     %%% PERDIDAS
1546
1547     tabla_perdidas_3D(1,1)=D_m/2-H_rotor(app.escalonamientos.Value)/2;
1548     tabla_perdidas_3D(2,1)=D_m/2;
1549     tabla_perdidas_3D(3,1)=D_m/2+H_rotor(app.escalonamientos.Value)/2;
1550     tabla_perdidas_3D(1,2)=perdidas_rotor_3D(1,app.escalonamientos.Value);
1551     tabla_perdidas_3D(2,2)=...
1552     perdidas_rotor_3D((app.n_puntos.Value+1)/2,app.escalonamientos.Value);
1553     tabla_perdidas_3D(3,2)=...
1554     perdidas_rotor_3D(app.n_puntos.Value,app.escalonamientos.Value);
1555     tabla_perdidas_3D(1,3)=...
1556     perdidas_estator_3D(1,app.escalonamientos.Value);
1557     tabla_perdidas_3D(2,3)=...
1558     perdidas_estator_3D...
1559     ((app.n_puntos.Value+1)/2,app.escalonamientos.Value);
1560     tabla_perdidas_3D(3,3)=...
1561     perdidas_estator_3D(app.n_puntos.Value,app.escalonamientos.Value);
1562     tabla_perdidas_3D(1,4)=W_u_3D;
1563     tabla_perdidas_3D(2,4)=W_u_3D;
1564     tabla_perdidas_3D(3,4)=W_u_3D;
1565     tabla_perdidas_3D(1,5)=eta_TT_3D(1,app.escalonamientos.Value);
1566     tabla_perdidas_3D(2,5)=...
1567     eta_TT_3D((app.n_puntos.Value+1)/2,app.escalonamientos.Value);
1568     tabla_perdidas_3D(3,5)=...
1569     eta_TT_3D(app.n_puntos.Value,app.escalonamientos.Value);
1570
1571     app.perdidas_3D.Data = tabla_perdidas_3D;
1572     app.RendimientototaldelescalonamientoEditField.Value=...
1573     (tabla_perdidas_3D(1,5)+tabla_perdidas_3D(2,5)+tabla_perdidas_3D(3,5))/3;
1574
1575
1576     end
1577
1578
1579 end
1580
1581     end
1582
1583     % Button pushed function: EscalonamientoanteriorButton
1584     function EscalonamientoanteriorButtonPushed2(app, event)
1585         if app.ley_actual.Value == 1
1586
1587
1588             if app.escalonamientos.Value <= 1
1589
1590
1591
1592                 app.escalonamientos.Value = app.escalonamientos.Value;
1593             else

```

```

1594
1595         app.escalonamientos.Value = app.escalonamientos.Value - 1;
1596
1597         [alpha_1,alpha_2,alpha_2_prima,alpha_3, ...
1598 alpha_3_prima, beta_1,beta_1_prima,beta_2,beta_2_prima, ...
1599 c_1,c_2,c_3,c_a,c_p,c_u1,c_u2,coef_R, ...
1600 curvatura_estator,curvatura_rotor,D_m, ...
1601 desviacion_estator,desviacion_rotor,...
1602 DF_eq_E,eta_c,eta_TT,eta_politropico,...
1603 gamma,H_estator,H_rotor,Lieblein_e,Lieblein_r,M_w_1,n_esc, ...
1604 P_01,P_02,P_03,P_1,P_2,P_3,perdidas_estator,perdidas_rotor, ...
1605 psi,r_c,r_c_estator,r_c_rotor,r_r_estator,r_r_rotor, ...
1606 rc_total_obtenida,rho_1,rho_2,rho_3,s_estator,s_rotor, suma_perdidas,...
1607 T_01,T_02,T_03,T_03ss,T_1,T_2,T_2s,T_3,T_total_final,u,w_1,w_2, ...
1608 W_u,zeta_a_e,zeta_a_r,zeta_e,zeta_p_e,zeta_p_r,zeta_r, ...
1609 zeta_s_e,zeta_s_r] ...
1610 = f_caso_2D(app.T_01_primer.Value,app.P_01_primer.Value,...
1611 app.sigma.Value,app.n.Value,app.m_dot.Value,app.DF_eq_R.Value,...
1612 app.R.Value,app.phi.Value,app.rc_total.Value);
1613
1614 [b_1,b_2,c_a_3D,W_u_3D,u_3D,c_u1_3D,c_u2_3D, c_1_3D,c_2_3D, ...
1615 w_u1_3D,w_u2_3D,beta_1_3D,beta_2_3D, alpha_1_3D,alpha_2_3D,R_3D, ...
1616 T_1_3D,M_c_3D,M_r_3D,DF_eq_R_3D,DF_eq_E_3D, ...
1617 zeta_p_R_3D,zeta_p_E_3D, perdidas_rotor_3D,perdidas_estator_3D, ...
1618 perdidas_3D,phi_3D,psi_3D,W_u_3D_is,eta_TT_3D,w_1_3D,w_2_3D] ...
1619 = f_torbellino_libre(c_u1,c_u2,D_m,c_a,W_u,n_esc,u,r_r_rotor,r_c_rotor, ...
1620 T_01,c_p,gamma,coef_R,app.sigma.Value,zeta_s_r,...
1621 zeta_a_r,zeta_s_e,zeta_a_e,app.n_puntos.Value);
1622
1623
1624     %%% TRIANGULOS 3D
1625
1626     % posicion inicial
1627
1628     A_1=[0,0];
1629     A_2=[w_2_3D(1,app.escalonamientos.Value)*sind...
1630         (beta_2_3D(1,app.escalonamientos.Value)),...
1631         w_2_3D(1,app.escalonamientos.Value)*cosd...
1632         (beta_2_3D(1,app.escalonamientos.Value))];
1633     A_3=[u_3D(1,app.escalonamientos.Value),0];
1634     A_4=[0,0];
1635     A_5=[w_1_3D(1,app.escalonamientos.Value)*sind(beta_1_3D...
1636         (1,app.escalonamientos.Value)),...
1637         w_1_3D(1,app.escalonamientos.Value)*...
1638         cosd(beta_1_3D(1,app.escalonamientos.Value))];
1639     A_6=[u_3D(1,app.escalonamientos.Value),0];
1640
1641     x_raiz=[A_1(1) A_2(1) A_3(1) A_4(1)];
1642     y_raiz=[A_1(2) A_2(2) A_3(2) A_4(2)];
1643     z_raiz=[A_5(1) A_6(1) A_4(1)];
1644     t_raiz=[A_5(2) A_6(2) A_4(2)];
1645 %
1646 %
1647 % posicion media
1648 %

```

```

1649 C_1=[0,0];
1650 C_2=[w_2_3D(3,app.escalonamientos.Value)*sind(beta_2_3D...
1651 (3,app.escalonamientos.Value)),...
1652 w_2_3D(3,app.escalonamientos.Value)*...
1653 cosd(beta_2_3D(3,app.escalonamientos.Value))];
1654 C_3=[u_3D(3,app.escalonamientos.Value),0];
1655 C_4=[0,0];
1656 C_5=[w_1_3D(3,app.escalonamientos.Value)*sind(beta_1_3D...
1657 (3,app.escalonamientos.Value)),w_1_3D(3,...
1658 app.escalonamientos.Value)*cosd(beta_1_3D...
1659 (3,app.escalonamientos.Value))];
1660 C_6=[u_3D(3,app.escalonamientos.Value),0];
1661
1662 x_medio=[C_1(1) C_2(1) C_3(1) C_4(1)];
1663 y_medio=[C_1(2) C_2(2) C_3(2) C_4(2)];
1664 z_medio=[C_5(1) C_6(1) C_4(1)];
1665 t_medio=[C_5(2) C_6(2) C_4(2)];
1666 %
1667 %
1668 % % posicion final
1669 %
1670 E_1=[0,0];
1671 E_2=[w_2_3D(5,app.escalonamientos.Value)*sind(beta_2_3D...
1672 (5,app.escalonamientos.Value)),w_2_3D(5,...
1673 app.escalonamientos.Value)*cosd(beta_2_3D(5,...
1674 app.escalonamientos.Value))];
1675 E_3=[u_3D(5,app.escalonamientos.Value),0];
1676 E_4=[0,0];
1677 E_5=[w_1_3D(5,app.escalonamientos.Value)*sind(beta_1_3D...
1678 (5,app.escalonamientos.Value)),w_1_3D(5,...
1679 app.escalonamientos.Value)*cosd(beta_1_3D...
1680 (5,app.escalonamientos.Value))];
1681 E_6=[u_3D(5,app.escalonamientos.Value),0];
1682
1683 x_cabeza=[E_1(1) E_2(1) E_3(1) E_4(1)];
1684 y_cabeza=[E_1(2) E_2(2) E_3(2) E_4(2)];
1685 z_cabeza=[E_5(1) E_6(1) E_4(1)];
1686 t_cabeza=[E_5(2) E_6(2) E_4(2)];
1687
1688 plot(app.triangulos_3D,x_raiz,y_raiz,"black",z_raiz,t_raiz,"black", ...
1689 x_medio,y_medio,"red",z_medio,t_medio,"red", ...
1690 x_cabeza,y_cabeza,"magenta",z_cabeza,t_cabeza,"magenta");
1691 legend(app.triangulos_3D,{'','radio ra z',...
1692 ','radio medio','','radio cabeza'],'Location','northeast');
1693
1694
1695 %%% ANGULOS
1696
1697 x=r_r_rotor(app.escalonamientos.Value)/...
1698 r_c_rotor(app.escalonamientos.Value):...
1699 (1-r_r_rotor(app.escalonamientos.Value)/...
1700 r_c_rotor(app.escalonamientos.Value))/...
1701 (app.n_puntos.Value-1):1;
1702 alpha_1_g=alpha_1_3D(:,app.escalonamientos.Value);
1703 alpha_2_g=alpha_2_3D(:,app.escalonamientos.Value);

```

```

1704     beta_1_g=beta_1_3D(:,app.escalonamientos.Value);
1705     beta_2_g=beta_2_3D(:,app.escalonamientos.Value);
1706     plot(app.UIAxes6,x,alpha_1_g,"blue",x,alpha_2_g,...
1707          "green",x,beta_1_g,"red",x,beta_2_g,"magenta")
1708     legend(app.UIAxes6,['alpha 1','alpha 2',...
1709          'Beta 1','Beta 2'],'Location','northwest');
1710
1711
1712
1713     %%% COEFICIENTES
1714
1715     x=r_r_rotor(app.escalonamientos.Value)/...
1716         r_c_rotor(app.escalonamientos.Value):...
1717         (1-r_r_rotor(app.escalonamientos.Value)/...
1718         r_c_rotor(app.escalonamientos.Value))/(app.n_puntos.Value-1):1;
1719     R_3D_g=R_3D(:,app.escalonamientos.Value);
1720     phi_3D_g=phi_3D(:,app.escalonamientos.Value);
1721     psi_3D_g=psi_3D(:,app.escalonamientos.Value);
1722     plot(app.UIAxes6_2,x,R_3D_g,"blue",x,phi_3D_g,"green",x,psi_3D_g,"red")
1723         legend(app.UIAxes6_2,['R',' ',' '],'Location','northwest');
1724
1725
1726     %%% PERDIDAS
1727
1728     tabla_perdidas_3D(1,1)=D_m/2-H_rotor(app.escalonamientos.Value)/2;
1729     tabla_perdidas_3D(2,1)=D_m/2;
1730     tabla_perdidas_3D(3,1)=D_m/2+H_rotor(app.escalonamientos.Value)/2;
1731     tabla_perdidas_3D(1,2)=perdidas_rotor_3D(1,app.escalonamientos.Value);
1732     tabla_perdidas_3D(2,2)=...
1733     perdidas_rotor_3D((app.n_puntos.Value+1)/2,app.escalonamientos.Value);
1734     tabla_perdidas_3D(3,2)=...
1735     perdidas_rotor_3D(app.n_puntos.Value,app.escalonamientos.Value);
1736     tabla_perdidas_3D(1,3)=perdidas_estator_3D(1,app.escalonamientos.Value);
1737     tabla_perdidas_3D(2,3)=...
1738     perdidas_estator_3D((app.n_puntos.Value+1)/2,app.escalonamientos.Value);
1739     tabla_perdidas_3D(3,3)=...
1740     perdidas_estator_3D(app.n_puntos.Value,app.escalonamientos.Value);
1741     tabla_perdidas_3D(1,4)=W_u_3D;
1742     tabla_perdidas_3D(2,4)=W_u_3D;
1743     tabla_perdidas_3D(3,4)=W_u_3D;
1744     tabla_perdidas_3D(1,5)=eta_TT_3D(1,app.escalonamientos.Value);
1745     tabla_perdidas_3D(2,5)=...
1746     eta_TT_3D((app.n_puntos.Value+1)/2,app.escalonamientos.Value);
1747     tabla_perdidas_3D(3,5)=...
1748     eta_TT_3D(app.n_puntos.Value,app.escalonamientos.Value);
1749
1750     app.perdidas_3D.Data = tabla_perdidas_3D;
1751     app.RendimientototaldelescalonamientoEditField.Value=...
1752     (tabla_perdidas_3D(1,5)+tabla_perdidas_3D(2,5)+tabla_perdidas_3D(3,5))/3;
1753
1754
1755     end
1756
1757
1758 elseif app.ley_actual.Value == 2

```

```

1759
1760     if  app.escalonamientos.Value <= 1
1761
1762
1763
1764         app.escalonamientos.Value = app.escalonamientos.Value;
1765     else
1766
1767         app.escalonamientos.Value = app.escalonamientos.Value - 1;
1768
1769         [alpha_1,alpha_2,alpha_2_prima,alpha_3, ...
1770 alpha_3_prima, beta_1,beta_1_prima,beta_2,beta_2_prima, ...
1771 c_1,c_2,c_3,c_a,c_p,c_u1,c_u2,coef_R, ...
1772 curvatura_estator,curvatura_rotor,D_m, ...
1773 desviacion_estator,desviacion_rotor,...
1774 DF_eq_E,eta_c,eta_TT,eta_politropico,...
1775 gamma,H_estator,H_rotor,Lieblein_e,Lieblein_r,M_w_1,n_esc, ...
1776 P_01,P_02,P_03,P_1,P_2,P_3,perdidas_estator,perdidas_rotor, ...
1777 psi,r_c,r_c_estator,r_c_rotor,r_r_estator,r_r_rotor, ...
1778 rc_total_obtenida,rho_1,rho_2,rho_3,s_estator,s_rotor, suma_perdidas,...
1779 T_01,T_02,T_03,T_03ss,T_1,T_2,T_2s,T_3,T_total_final,u,w_1,w_2, ...
1780 W_u,zeta_a_e,zeta_a_r,zeta_e,zeta_p_e,zeta_p_r,zeta_r, ...
1781 zeta_s_e,zeta_s_r] ...
1782 = f_caso_2D(app.T_01_primer.Value,app.P_01_primer.Value,...
1783 app.sigma.Value,app.n.Value,app.m_dot.Value,app.DF_eq_R.Value,...
1784 app.R.Value,app.phi.Value,app.rc_total.Value);
1785
1786 [a,b,c_a1_3D,c_a2_3D,W_u_3D,u_3D,c_u1_3D,c_u2_3D, c_1_3D,c_2_3D, ...
1787 w_u1_3D,w_u2_3D,beta_1_3D,beta_2_3D, alpha_1_3D,alpha_2_3D,R_3D, ...
1788 T_1_3D,M_c_3D,M_r_3D,DF_eq_R_3D,DF_eq_E_3D, ...
1789 zeta_p_R_3D,zeta_p_E_3D, perdidas_rotor_3D,perdidas_estator_3D, ...
1790 perdidas_3D,phi_3D,psi_3D,W_u_3D_is,eta_TT_3D,w_1_3D,w_2_3D] ...
1791 = f_reaccion_constante(c_u1,c_u2,D_m,...
1792 c_a,W_u,n_esc,u,r_r_rotor,r_c_rotor, ...
1793 T_01,c_p,gamma,coef_R,app.sigma.Value,...
1794 zeta_s_r,zeta_a_r,zeta_s_e,zeta_a_e,app.n.puntos.Value)
1795
1796
1797     %% TRIANGULOS 3D
1798
1799     % posicion inicial
1800
1801     A_1=[0,0];
1802     A_2=[w_2_3D(1,app.escalonamientos.Value)*sind(beta_2_3D...
1803         (1,app.escalonamientos.Value)),...
1804         w_2_3D(1,app.escalonamientos.Value)*...
1805         cosd(beta_2_3D(1,app.escalonamientos.Value))];
1806     A_3=[u_3D(1,app.escalonamientos.Value),0];
1807     A_4=[0,0];
1808     A_5=[w_1_3D(1,app.escalonamientos.Value)*sind...
1809         (beta_1_3D(1,app.escalonamientos.Value)),...
1810         w_1_3D(1,app.escalonamientos.Value)*cosd...
1811         (beta_1_3D(1,app.escalonamientos.Value))];
1812     A_6=[u_3D(1,app.escalonamientos.Value),0];
1813

```

```

1814 x_raiz=[A_1(1) A_2(1) A_3(1) A_1(1)];
1815 y_raiz=[A_1(2) A_2(2) A_3(2) A_1(2)];
1816 z_raiz=[A_4(1) A_5(1) A_6(1) A_4(1)];
1817 t_raiz=[A_4(2) A_5(2) A_6(2) A_4(2)];
1818 %
1819 %
1820 %      % posicion media
1821 %
1822 C_1=[0,0];
1823 C_2=[w_2_3D(3,app.escalonamientos.Value)*sind...
1824      (beta_2_3D(3,app.escalonamientos.Value)),...
1825      w_2_3D(3,app.escalonamientos.Value)*cosd(beta_2_3D...
1826      (3,app.escalonamientos.Value))];
1827 C_3=[u_3D(3,app.escalonamientos.Value),0];
1828 C_4=[0,0];
1829 C_5=[w_1_3D(3,app.escalonamientos.Value)*sind(beta_1_3D...
1830      (3,app.escalonamientos.Value)),...
1831      w_1_3D(3,app.escalonamientos.Value)*...
1832      cosd(beta_1_3D(3,app.escalonamientos.Value))];
1833 C_6=[u_3D(3,app.escalonamientos.Value),0];
1834
1835 x_medio=[C_1(1) C_2(1) C_3(1) C_1(1)];
1836 y_medio=[C_1(2) C_2(2) C_3(2) C_1(2)];
1837 z_medio=[C_4(1) C_5(1) C_6(1) C_4(1)];
1838 t_medio=[C_4(2) C_5(2) C_6(2) C_4(2)];
1839 %
1840 %
1841 %      % posicion final
1842 %
1843 E_1=[0,0];
1844 E_2=[w_2_3D(5,app.escalonamientos.Value)*sind...
1845      (beta_2_3D(5,app.escalonamientos.Value)),...
1846      w_2_3D(5,app.escalonamientos.Value)*cosd...
1847      (beta_2_3D(5,app.escalonamientos.Value))];
1848 E_3=[u_3D(5,app.escalonamientos.Value),0];
1849 E_4=[0,0];
1850 E_5=[w_1_3D(5,app.escalonamientos.Value)*sind(beta_1_3D...
1851      (5,app.escalonamientos.Value)),...
1852      w_1_3D(5,app.escalonamientos.Value)*...
1853      cosd(beta_1_3D(5,app.escalonamientos.Value))];
1854 E_6=[u_3D(5,app.escalonamientos.Value),0];
1855
1856 x_cabeza=[E_1(1) E_2(1) E_3(1) E_1(1)];
1857 y_cabeza=[E_1(2) E_2(2) E_3(2) E_1(2)];
1858 z_cabeza=[E_4(1) E_5(1) E_6(1) E_4(1)];
1859 t_cabeza=[E_4(2) E_5(2) E_6(2) E_4(2)];
1860
1861 plot(app.triangulos_3D,x_raiz,y_raiz,"black",z_raiz,t_raiz,"black", ...
1862      x_medio,y_medio,"red",z_medio,t_medio,"red", ...
1863      x_cabeza,y_cabeza,"magenta",z_cabeza,t_cabeza,"magenta");
1864 legend(app.triangulos_3D,{'','radio ra z','',...
1865      'radio medio','','radio cabeza'],'Location','northeast');
1866
1867
1868      %% ANGULOS

```

```

1869
1870         x=r_r_rotor(app.escalonamientos.Value)/...
1871             r_c_rotor(app.escalonamientos.Value):...
1872             (1-r_r_rotor(app.escalonamientos.Value)/r_c_rotor...
1873             (app.escalonamientos.Value))/(app.n_puntos.Value-1):1;
1874     alpha_1_g=alpha_1_3D(:,app.escalonamientos.Value);
1875     alpha_2_g=alpha_2_3D(:,app.escalonamientos.Value);
1876     beta_1_g=beta_1_3D(:,app.escalonamientos.Value);
1877     beta_2_g=beta_2_3D(:,app.escalonamientos.Value);
1878     plot(app.UIAxes6,x,alpha_1_g,"blue",x,alpha_2_g,...
1879          "green",x,beta_1_g,"red",x,beta_2_g,"magenta")
1880     legend(app.UIAxes6,['alpha 1','alpha 2',['Beta 1...' ...
1881          ''],'Beta 2'],'Location','northwest');
1882
1883
1884
1885     %%% COEFICIENTES
1886
1887         x=r_r_rotor(app.escalonamientos.Value)...
1888             /r_c_rotor(app.escalonamientos.Value):...
1889             (1-r_r_rotor(app.escalonamientos.Value)/...
1890     r_c_rotor(app.escalonamientos.Value))/(app.n_puntos.Value-1):1;
1891     R_3D_g=R_3D(:,app.escalonamientos.Value);
1892     phi_3D_g=phi_3D(:,app.escalonamientos.Value);
1893     psi_3D_g=psi_3D(:,app.escalonamientos.Value);
1894     plot(app.UIAxes6_2,x,R_3D_g,"blue",x,phi_3D_g,"green",x,psi_3D_g,"red")
1895         legend(app.UIAxes6_2,['R',' ',' '],'Location','northwest');
1896
1897
1898     %%% PERDIDAS
1899
1900     tabla_perdidas_3D(1,1)=D_m/2-H_rotor(app.escalonamientos.Value)/2;
1901     tabla_perdidas_3D(2,1)=D_m/2;
1902     tabla_perdidas_3D(3,1)=D_m/2+H_rotor(app.escalonamientos.Value)/2;
1903     tabla_perdidas_3D(1,2)=perdidas_rotor_3D(1,app.escalonamientos.Value);
1904     tabla_perdidas_3D(2,2)=...
1905     perdidas_rotor_3D((app.n_puntos.Value+1)/2,app.escalonamientos.Value);
1906     tabla_perdidas_3D(3,2)=...
1907     perdidas_rotor_3D(app.n_puntos.Value,app.escalonamientos.Value);
1908     tabla_perdidas_3D(1,3)=...
1909     perdidas_estator_3D(1,app.escalonamientos.Value);
1910     tabla_perdidas_3D(2,3)=...
1911     perdidas_estator_3D((app.n_puntos.Value+1)/2,app.escalonamientos.Value);
1912     tabla_perdidas_3D(3,3)=...
1913     perdidas_estator_3D(app.n_puntos.Value,app.escalonamientos.Value);
1914     tabla_perdidas_3D(1,4)=W_u_3D;
1915     tabla_perdidas_3D(2,4)=W_u_3D;
1916     tabla_perdidas_3D(3,4)=W_u_3D;
1917     tabla_perdidas_3D(1,5)=eta_TT_3D(1,app.escalonamientos.Value);
1918     tabla_perdidas_3D(2,5)=...
1919     eta_TT_3D((app.n_puntos.Value+1)/2,app.escalonamientos.Value);
1920     tabla_perdidas_3D(3,5)=...
1921     eta_TT_3D(app.n_puntos.Value,app.escalonamientos.Value);
1922
1923     app.perdidas_3D.Data = tabla_perdidas_3D;

```



```

1924         app.RendimientototaldelescalonamientoEditField.Value=...
1925     (tabla_perdidas_3D(1,5)+tabla_perdidas_3D(2,5)+tabla_perdidas_3D(3,5))/3;
1926
1927
1928 end
1929
1930 else
1931
1932     if app.escalonamientos.Value <= 1
1933
1934
1935
1936         app.escalonamientos.Value = app.escalonamientos.Value;
1937     else
1938
1939         app.escalonamientos.Value = app.escalonamientos.Value - 1;
1940
1941         [alpha_1,alpha_2,alpha_2_prima,alpha_3, ...
1942 alpha_3_prima, beta_1,beta_1_prima,beta_2,beta_2_prima, ...
1943 c_1,c_2,c_3,c_a,c_p,c_u1,c_u2,coef_R, ...
1944 curvatura_estator,curvatura_rotor,D_m, ...
1945 desviacion_estator,desviacion_rotor,...
1946 DF_eq_E,eta_c,eta_TT,eta_politropico,...
1947 gamma,H_estator,H_rotor,Lieblein_e,Lieblein_r,M_w_1,n_esc, ...
1948 P_01,P_02,P_03,P_1,P_2,P_3,perdidas_estator,perdidas_rotor, ...
1949 psi,r_c,r_c_estator,r_c_rotor,r_r_estator,r_r_rotor, ...
1950 rc_total_obtenida,rho_1,rho_2,rho_3,...
1951 s_estator,s_rotor, suma_perdidas,...
1952 T_01,T_02,T_03,T_03ss,T_1,T_2,T_2s,T_3,T_total_final,u,w_1,w_2, ...
1953 W_u,zeta_a_e,zeta_a_r,zeta_e,zeta_p_e,zeta_p_r,zeta_r, ...
1954 zeta_s_e,zeta_s_r] ...
1955     = f_caso_2D(app.T_01_primer.Value,app.P_01_primer.Value,...
1956 app.sigma.Value,app.n.Value,app.m_dot.Value,app.DF_eq_R.Value,...
1957 app.R.Value,app.phi.Value,app.rc_total.Value);
1958
1959 [a,b,c_a1_3D,c_a2_3D,W_u_3D,u_3D,c_u1_3D,c_u2_3D, c_1_3D,c_2_3D, ...
1960 w_u1_3D,w_u2_3D,beta_1_3D,beta_2_3D, alpha_1_3D,alpha_2_3D,R_3D, ...
1961 T_1_3D,M_c_3D,M_r_3D,DF_eq_R_3D,DF_eq_E_3D, ...
1962 zeta_p_R_3D,zeta_p_E_3D, perdidas_rotor_3D,perdidas_estator_3D, ...
1963 perdidas_3D,phi_3D,psi_3D,W_u_3D_is,eta_TT_3D,w_1_3D,w_2_3D] ...
1964 = f_exponencial(c_u1,c_u2,D_m,c_a,W_u,n_esc,u,r_r_rotor,r_c_rotor, ...
1965 T_01,c_p,gamma,coef_R,app.sigma.Value,zeta_s_r,...
1966 zeta_a_r,zeta_s_e,zeta_a_e,app.n_puntos.Value)
1967
1968
1969     %% TRIANGULOS 3D
1970
1971     % posicion inicial
1972
1973     A_1=[0,0];
1974     A_2=[w_2_3D(1,app.escalonamientos.Value)*sind(beta_2_3D...
1975         (1,app.escalonamientos.Value)),w_2_3D(1,...
1976         app.escalonamientos.Value)*cosd(beta_2_3D(1,...
1977         app.escalonamientos.Value))];
1978     A_3=[u_3D(1,app.escalonamientos.Value),0];

```



```

1979 A_4=[0,0];
1980 A_5=[w_1_3D(1,app.escalonamientos.Value)*sind...
1981     (beta_1_3D(1,app.escalonamientos.Value)),...
1982     w_1_3D(1,app.escalonamientos.Value)*cosd...
1983     (beta_1_3D(1,app.escalonamientos.Value))];
1984 A_6=[u_3D(1,app.escalonamientos.Value),0];
1985
1986 x_raiz=[A_1(1) A_2(1) A_3(1) A_1(1)];
1987 y_raiz=[A_1(2) A_2(2) A_3(2) A_1(2)];
1988 z_raiz=[A_4(1) A_5(1) A_6(1) A_4(1)];
1989 t_raiz=[A_4(2) A_5(2) A_6(2) A_4(2)];
1990 %
1991 %
1992 %     % posicion media
1993 %
1994 C_1=[0,0];
1995 C_2=[w_2_3D(3,app.escalonamientos.Value)*sind(beta_2_3D(3 ...
1996     ,app.escalonamientos.Value)),w_2_3D(3,...
1997     app.escalonamientos.Value)*cosd(beta_2_3D...
1998     (3,app.escalonamientos.Value))];
1999 C_3=[u_3D(3,app.escalonamientos.Value),0];
2000 C_4=[0,0];
2001 C_5=[w_1_3D(3,app.escalonamientos.Value)*sind(beta_1_3D...
2002     (3,app.escalonamientos.Value)),w_1_3D(3,...
2003     app.escalonamientos.Value)*cosd(beta_1_3D(3,...
2004     app.escalonamientos.Value))];
2005 C_6=[u_3D(3,app.escalonamientos.Value),0];
2006
2007 x_medio=[C_1(1) C_2(1) C_3(1) C_1(1)];
2008 y_medio=[C_1(2) C_2(2) C_3(2) C_1(2)];
2009 z_medio=[C_4(1) C_5(1) C_6(1) C_4(1)];
2010 t_medio=[C_4(2) C_5(2) C_6(2) C_4(2)];
2011 %
2012 %
2013 %     % posicion final
2014 %
2015 E_1=[0,0];
2016 E_2=[w_2_3D(5,app.escalonamientos.Value)*sind...
2017     (beta_2_3D(5,app.escalonamientos.Value)),...
2018     w_2_3D(5,app.escalonamientos.Value)*...
2019     cosd(beta_2_3D(5,app.escalonamientos.Value))];
2020 E_3=[u_3D(5,app.escalonamientos.Value),0];
2021 E_4=[0,0];
2022 E_5=[w_1_3D(5,app.escalonamientos.Value)*sind(beta_1_3D...
2023     (5,app.escalonamientos.Value)),...
2024     w_1_3D(5,app.escalonamientos.Value)*...
2025     cosd(beta_1_3D(5,app.escalonamientos.Value))];
2026 E_6=[u_3D(5,app.escalonamientos.Value),0];
2027
2028 x_cabeza=[E_1(1) E_2(1) E_3(1) E_1(1)];
2029 y_cabeza=[E_1(2) E_2(2) E_3(2) E_1(2)];
2030 z_cabeza=[E_4(1) E_5(1) E_6(1) E_4(1)];
2031 t_cabeza=[E_4(2) E_5(2) E_6(2) E_4(2)];
2032
2033 plot(app.triangulos_3D,x_raiz,y_raiz,"black",...

```

```

2034         z_raiz,t_raiz,"black", ...
2035         x_medio,y_medio,"red",z_medio,t_medio,"red", ...
2036         x_cabeza,y_cabeza,"magenta",z_cabeza,t_cabeza,"magenta");
2037     legend(app.triangulos_3D,{'','radio ra z','',...
2038         'radio medio','','radio cabeza'},'Location','northeast');
2039
2040
2041     %%% ANGULOS
2042
2043     x=r_r_rotor(app.escalonamientos.Value)/...
2044         r_c_rotor(app.escalonamientos.Value):(1-r_r_rotor...
2045         (app.escalonamientos.Value)/r_c_rotor...
2046         (app.escalonamientos.Value))/(app.n_puntos.Value-1):1;
2047     alpha_1_g=alpha_1_3D(:,app.escalonamientos.Value);
2048     alpha_2_g=alpha_2_3D(:,app.escalonamientos.Value);
2049     beta_1_g=beta_1_3D(:,app.escalonamientos.Value);
2050     beta_2_g=beta_2_3D(:,app.escalonamientos.Value);
2051     plot(app.UIAxes6,x,alpha_1_g,"blue",x,alpha_2_g,...
2052         "green",x,beta_1_g,"red",x,beta_2_g,"magenta")
2053     legend(app.UIAxes6,{'alpha 1','alpha 2',...
2054         'Beta 1','Beta 2'},'Location','northwest');
2055
2056
2057
2058     %%% COEFICIENTES
2059
2060     x=r_r_rotor(app.escalonamientos.Value)/r_c_rotor...
2061         (app.escalonamientos.Value):(1-r_r_rotor...
2062         (app.escalonamientos.Value)/r_c_rotor...
2063         (app.escalonamientos.Value))/(app.n_puntos.Value-1):1;
2064     R_3D_g=R_3D(:,app.escalonamientos.Value);
2065     phi_3D_g=phi_3D(:,app.escalonamientos.Value);
2066     psi_3D_g=psi_3D(:,app.escalonamientos.Value);
2067     plot(app.UIAxes6_2,x,R_3D_g,"blue",x,phi_3D_g,"green",...
2068         x,psi_3D_g,"red")
2069     legend ...
2070         (app.UIAxes6_2,{'R',' ',' '},'Location','northwest');
2071
2072
2073     %%% PERDIDAS
2074
2075     tabla_perdidas_3D...
2076         (1,1)=D_m/2-H_rotor(app.escalonamientos.Value)/2;
2077     tabla_perdidas_3D(2,1)=D_m/2;
2078     tabla_perdidas_3D...
2079         (3,1)=D_m/2+H_rotor(app.escalonamientos.Value)/2;
2080     tabla_perdidas_3D...
2081         (1,2)=perdidas_rotor_3D(1,app.escalonamientos.Value);
2082     tabla_perdidas_3D...
2083         (2,2)=perdidas_rotor_3D...
2084         ((app.n_puntos.Value+1)/2,app.escalonamientos.Value);
2085     tabla_perdidas_3D(3,2)=perdidas_rotor_3D...
2086         (app.n_puntos.Value,app.escalonamientos.Value);
2087     tabla_perdidas_3D(1,3)=perdidas_estator_3D...
2088         (1,app.escalonamientos.Value);

```

```

2089         tabla_perdidas_3D(2,3)=perdidas_estator_3D...
2090             ((app.n_puntos.Value+1)/2,app.escalonamientos.Value);
2091         tabla_perdidas_3D(3,3)=perdidas_estator_3D...
2092             (app.n_puntos.Value,app.escalonamientos.Value);
2093         tabla_perdidas_3D(1,4)=W_u_3D;
2094         tabla_perdidas_3D(2,4)=W_u_3D;
2095         tabla_perdidas_3D(3,4)=W_u_3D;
2096         tabla_perdidas_3D(1,5)=eta_TT_3D(1,app.escalonamientos.Value);
2097         tabla_perdidas_3D(2,5)=eta_TT_3D...
2098             ((app.n_puntos.Value+1)/2,app.escalonamientos.Value);
2099         tabla_perdidas_3D(3,5)=eta_TT_3D...
2100             (app.n_puntos.Value,app.escalonamientos.Value);
2101
2102         app.perdidas_3D.Data = tabla_perdidas_3D;
2103         app.RendimientototaldelescalonamientoEditField.Value=...
2104         (tabla_perdidas_3D(1,5)+tabla_perdidas_3D(2,5)+tabla_perdidas_3D(3,5))/3;
2105
2106
2107         end
2108
2109
2110     end
2111
2112
2113     end
2114
2115     % Button pushed function: Button
2116     function ButtonPushed(app, event)
2117         informacion;
2118     end
2119 end
2120
2121 % Component initialization
2122 methods (Access = private)
2123
2124     % Create UIFigure and components
2125     function createComponents(app)
2126
2127         % Create UIFigure and hide until all components are created
2128         app.UIFigure = uifigure('Visible', 'off');
2129         app.UIFigure.Position = [100 100 640 480];
2130         app.UIFigure.Name = 'MATLAB App';
2131
2132         % Create TabGroup
2133         app.TabGroup = uitabgroup(app.UIFigure);
2134         app.TabGroup.Position = [1 1 640 480];
2135
2136         % Create EspecificacionesTab
2137         app.EspecificacionesTab = uitab(app.TabGroup);
2138         app.EspecificacionesTab.Title = 'Especificaciones';
2139
2140         % Create P_01_primer
2141         app.P_01_primer = uieditfield(app.EspecificacionesTab, 'numeric');
2142         app.P_01_primer.Position = [270 340 100 22];
2143         app.P_01_primer.Value = 101.5;

```

```

2144
2145     % Create Valorentrey14Label
2146     app.Valorentrey14Label = uilabel(app.EspecificacionesTab);
2147     app.Valorentrey14Label.Position = [386 307 110 22];
2148     app.Valorentrey14Label.Text = '(Valor entre 1 y 1.4)';
2149
2150     % Create Valorentre04y06Label
2151     app.Valorentre04y06Label = uilabel(app.EspecificacionesTab);
2152     app.Valorentre04y06Label.Position = [386 237 120 22];
2153     app.Valorentre04y06Label.Text = '(Valor entre 0.4 y 0.6)';
2154
2155     % Create Valorentre04y07Label
2156     app.Valorentre04y07Label = uilabel(app.EspecificacionesTab);
2157     app.Valorentre04y07Label.Position = [386 203 120 22];
2158     app.Valorentre04y07Label.Text = '(Valor entre 0.4 y 0.7)';
2159
2160     % Create Nopuedesermayora195Label
2161     app.Nopuedesermayora195Label = uilabel(app.EspecificacionesTab);
2162     app.Nopuedesermayora195Label.Position = [386 138 158 22];
2163     app.Nopuedesermayora195Label.Text = '(No puede ser mayor a 1.95)';
2164
2165     % Create CALCULARButton
2166     app.CALCULARButton = uibutton(app.EspecificacionesTab, 'push');
2167     app.CALCULARButton.ButtonPushedFcn =...
2168         createCallbackFcn(app, @CALCULARButtonPushed, true);
2169     app.CALCULARButton.Position = [250 9 135 60];
2170     app.CALCULARButton.Text = 'CALCULAR';
2171
2172     % Create CASOBIDIMENSIONALLabel
2173     app.CASOBIDIMENSIONALLabel = uilabel(app.EspecificacionesTab);
2174     app.CASOBIDIMENSIONALLabel.FontSize = 14;
2175     app.CASOBIDIMENSIONALLabel.FontWeight = 'bold';
2176     app.CASOBIDIMENSIONALLabel.Position = [237 417 161 22];
2177     app.CASOBIDIMENSIONALLabel.Text = 'CASO BIDIMENSIONAL';
2178
2179     % Create TempderemansoalaentradaKEditFieldLabel
2180     app.TempderemansoalaentradaKEditFieldLabel =...
2181         uilabel(app.EspecificacionesTab);
2182     app.TempderemansoalaentradaKEditFieldLabel.HorizontalAlignment = 'right';
2183     app.TempderemansoalaentradaKEditFieldLabel.Position = [68 374 187 22];
2184     app.TempderemansoalaentradaKEditFieldLabel.Text =...
2185         'Temp de remanso a la entrada (K)';
2186
2187     % Create T_01_primer
2188     app.T_01_primer = uieditfield(app.EspecificacionesTab, 'numeric');
2189     app.T_01_primer.Position = [270 374 100 22];
2190     app.T_01_primer.Value = 293;
2191
2192     % Create SolidezdelacascadaEditFieldLabel
2193     app.SolidezdelacascadaEditFieldLabel = uilabel(app.EspecificacionesTab);
2194     app.SolidezdelacascadaEditFieldLabel.HorizontalAlignment = 'right';
2195     app.SolidezdelacascadaEditFieldLabel.Position = [133 306 122 22];
2196     app.SolidezdelacascadaEditFieldLabel.Text = 'Solidez de la cascada';
2197
2198     % Create sigma

```

```

2199     app.sigma = uieditfield(app.EspecificacionesTab, 'numeric');
2200     app.sigma.Limits = [1 1.4];
2201     app.sigma.Position = [270 306 100 22];
2202     app.sigma.Value = 1;
2203
2204     % Create RevolucionesporminutoEditFieldLabel
2205     app.RevolucionesporminutoEditFieldLabel =...
2206         uilabel(app.EspecificacionesTab);
2207     app.RevolucionesporminutoEditFieldLabel.HorizontalAlignment = 'right';
2208     app.RevolucionesporminutoEditFieldLabel.Position = [118 272 137 22];
2209     app.RevolucionesporminutoEditFieldLabel.Text = 'Revoluciones por minuto';
2210
2211     % Create n
2212     app.n = uieditfield(app.EspecificacionesTab, 'numeric');
2213     app.n.Position = [270 272 100 22];
2214     app.n.Value = 11500;
2215
2216     % Create GradodereaccinEditFieldLabel
2217     app.GradodereaccinEditFieldLabel = uilabel(app.EspecificacionesTab);
2218     app.GradodereaccinEditFieldLabel.HorizontalAlignment = 'right';
2219     app.GradodereaccinEditFieldLabel.Position = [151 237 104 22];
2220     app.GradodereaccinEditFieldLabel.Text = 'Grado de reacci n';
2221
2222     % Create R
2223     app.R = uieditfield(app.EspecificacionesTab, 'numeric');
2224     app.R.Limits = [0.4 0.6];
2225     app.R.Position = [270 238 100 22];
2226     app.R.Value = 0.55;
2227
2228     % Create CoeficientedeflujoEditFieldLabel
2229     app.CoficientedeflujoEditFieldLabel = uilabel(app.EspecificacionesTab);
2230     app.CoficientedeflujoEditFieldLabel.HorizontalAlignment = 'right';
2231     app.CoficientedeflujoEditFieldLabel.Position = [148 203 107 22];
2232     app.CoficientedeflujoEditFieldLabel.Text = 'Coeficiente de flujo';
2233
2234     % Create phi
2235     app.phi = uieditfield(app.EspecificacionesTab, 'numeric');
2236     app.phi.Limits = [0.4 0.7];
2237     app.phi.Position = [270 204 100 22];
2238     app.phi.Value = 0.7;
2239
2240     % Create GastomsicokgsEditFieldLabel
2241     app.GastomsicokgsEditFieldLabel = uilabel(app.EspecificacionesTab);
2242     app.GastomsicokgsEditFieldLabel.HorizontalAlignment = 'right';
2243     app.GastomsicokgsEditFieldLabel.Position = [143 170 112 22];
2244     app.GastomsicokgsEditFieldLabel.Text = 'Gasto m sico (kg/s)';
2245
2246     % Create m_dot
2247     app.m_dot = uieditfield(app.EspecificacionesTab, 'numeric');
2248     app.m_dot.Position = [270 170 100 22];
2249     app.m_dot.Value = 20;
2250
2251     % Create RelacindedifusinanelrotorEditFieldLabel
2252     app.RelacindedifusinanelrotorEditFieldLabel =...
2253         uilabel(app.EspecificacionesTab);

```

```

2254 app.RelacinedifusinenelrotorEditFieldLabel.HorizontalAlignment =...
2255     'right';
2256 app.RelacinedifusinenelrotorEditFieldLabel.Position = [85 136 170 22];
2257     app.RelacinedifusinenelrotorEditFieldLabel.Text =...
2258     'Relaci n de difusi n en el rotor';
2259
2260     % Create DF_eq_R
2261     app.DF_eq_R = uieditfield(app.EspecificacionesTab, 'numeric');
2262     app.DF_eq_R.Limits = [-Inf 1.95];
2263     app.DF_eq_R.Position = [270 136 100 22];
2264     app.DF_eq_R.Value = 1.75;
2265
2266     % Create RelacindecompresintotalEditFieldLabel
2267     app.RelacindecompresintotalEditFieldLabel =...
2268     uilabel(app.EspecificacionesTab);
2269 app.RelacindecompresintotalEditFieldLabel.HorizontalAlignment = 'right';
2270 app.RelacindecompresintotalEditFieldLabel.Position = [95 103 160 22];
2271 app.RelacindecompresintotalEditFieldLabel.Text =...
2272     'Relaci n de compresi n total';
2273
2274     % Create rc_total
2275     app.rc_total = uieditfield(app.EspecificacionesTab, 'numeric');
2276     app.rc_total.Position = [270 103 100 22];
2277     app.rc_total.Value = 6;
2278
2279     % Create PresinderemansoalaentradaKPaEditFieldLabel
2280     app.PresinderemansoalaentradaKPaEditFieldLabel = ...
2281     uilabel(app.EspecificacionesTab);
2282 app.PresinderemansoalaentradaKPaEditFieldLabel.HorizontalAlignment =...
2283     'right';
2284     app.PresinderemansoalaentradaKPaEditFieldLabel.Position =...
2285     [41 340 214 22];
2286     app.PresinderemansoalaentradaKPaEditFieldLabel.Text = ...
2287     'Presi n de remanso a la entrada (KPa)';
2288
2289     % Create Resultados2DTab
2290     app.Resultados2DTab = uitab(app.TabGroup);
2291     app.Resultados2DTab.Title = 'Resultados 2D';
2292
2293     % Create TabGroup2
2294     app.TabGroup2 = uitabgroup(app.Resultados2DTab);
2295     app.TabGroup2.Position = [1 1 638 455];
2296
2297     % Create Tri ngulosdevelocidadesTab
2298     app.Tri ngulosdevelocidadesTab = uitab(app.TabGroup2);
2299     app.Tri ngulosdevelocidadesTab.Title = 'Tri ngulos de velocidades';
2300
2301     % Create UIAxes
2302     app.UIAxes = uiaxes(app.Tri ngulosdevelocidadesTab);
2303     title(app.UIAxes, 'Tri ngulo de velocidades')
2304     xlabel(app.UIAxes, 'Z')
2305     app.UIAxes.Position = [236 181 374 236];
2306
2307     % Create ngulosdeflujoLabel
2308     app.ngulosdeflujoLabel = uilabel(app.Tri ngulosdevelocidadesTab);

```

```
2309     app.ngulosdeflujoLabel.FontWeight = 'bold';
2310     app.ngulosdeflujoLabel.Position = [406 160 115 22];
2311     app.ngulosdeflujoLabel.Text = ' ngulos de flujo ( )';
2312
2313     % Create VelocidadesmsLabel
2314     app.VelocidadesmsLabel = uilabel(app.TringulosdevelocidadesTab);
2315     app.VelocidadesmsLabel.FontWeight = 'bold';
2316     app.VelocidadesmsLabel.Position = [81 171 106 22];
2317     app.VelocidadesmsLabel.Text = 'Velocidades (m/s)';
2318
2319     % Create NdeescalonamientosEditFieldLabel
2320     app.NdeescalonamientosEditFieldLabel = ...
2321         uilabel(app.TringulosdevelocidadesTab);
2322     app.NdeescalonamientosEditFieldLabel.HorizontalAlignment = 'right';
2323     app.NdeescalonamientosEditFieldLabel.Position = [25 368 128 22];
2324     app.NdeescalonamientosEditFieldLabel.Text = 'N de escalonamientos';
2325
2326     % Create NdeescalonamientosEditField
2327     app.NdeescalonamientosEditField = ...
2328         uieditfield(app.TringulosdevelocidadesTab, 'numeric');
2329     app.NdeescalonamientosEditField.Editable = 'off';
2330     app.NdeescalonamientosEditField.Position = [168 368 47 22];
2331
2332     % Create CoefdecargaEditFieldLabel
2333     app.CofdecargaEditFieldLabel = ...
2334         uilabel(app.TringulosdevelocidadesTab);
2335     app.CofdecargaEditFieldLabel.HorizontalAlignment = 'right';
2336     app.CofdecargaEditFieldLabel.Position = [69 324 84 22];
2337     app.CofdecargaEditFieldLabel.Text = 'Coef. de carga';
2338
2339     % Create CoefdecargaEditField
2340     app.CofdecargaEditField = ...
2341         uieditfield(app.TringulosdevelocidadesTab, 'numeric');
2342     app.CofdecargaEditField.Editable = 'off';
2343     app.CofdecargaEditField.Position = [168 324 47 22];
2344
2345     % Create EditFieldLabel
2346     app.EditFieldLabel = uilabel(app.TringulosdevelocidadesTab);
2347     app.EditFieldLabel.BackgroundColor = [0.9412 0.9412 0.9412];
2348     app.EditFieldLabel.HorizontalAlignment = 'right';
2349     app.EditFieldLabel.Position = [349 116 25 22];
2350     app.EditFieldLabel.Text = ' 1 ';
2351
2352     % Create alpha1
2353     app.alpha1 = uieditfield(app.TringulosdevelocidadesTab, 'numeric');
2354     app.alpha1.Position = [389 116 47 22];
2355
2356     % Create EditField_2Label
2357     app.EditField_2Label = uilabel(app.TringulosdevelocidadesTab);
2358     app.EditField_2Label.HorizontalAlignment = 'right';
2359     app.EditField_2Label.Position = [454 116 25 22];
2360     app.EditField_2Label.Text = ' 2 ';
2361
2362     % Create alpha2
2363     app.alpha2 = uieditfield(app.TringulosdevelocidadesTab, 'numeric');
```



```
2364     app.alpha2.Position = [494 116 47 22];
2365
2366     % Create EditField_3Label
2367     app.EditField_3Label = uilabel(app.TringulosdevelocidadesTab);
2368     app.EditField_3Label.HorizontalAlignment = 'right';
2369     app.EditField_3Label.Position = [349 84 25 22];
2370     app.EditField_3Label.Text = ' 1 ';
2371
2372     % Create beta1
2373     app.beta1 = uieditfield(app.TringulosdevelocidadesTab, 'numeric');
2374     app.beta1.Position = [389 84 47 22];
2375
2376     % Create EditField_4Label
2377     app.EditField_4Label = uilabel(app.TringulosdevelocidadesTab);
2378     app.EditField_4Label.HorizontalAlignment = 'right';
2379     app.EditField_4Label.Position = [454 84 25 22];
2380     app.EditField_4Label.Text = ' 2 ';
2381
2382     % Create beta2
2383     app.beta2 = ...
2384         uieditfield(app.TringulosdevelocidadesTab, 'numeric');
2385     app.beta2.Position = [494 84 47 22];
2386
2387     % Create c1EditFieldLabel
2388     app.c1EditFieldLabel = uilabel(app.TringulosdevelocidadesTab);
2389     app.c1EditFieldLabel.HorizontalAlignment = 'right';
2390     app.c1EditFieldLabel.Position = [31 126 25 22];
2391     app.c1EditFieldLabel.Text = 'c1';
2392
2393     % Create c1EditField
2394     app.c1EditField = ...
2395         uieditfield(app.TringulosdevelocidadesTab, 'numeric');
2396     app.c1EditField.Position = [71 126 58 22];
2397
2398     % Create w1EditFieldLabel
2399     app.w1EditFieldLabel = uilabel(app.TringulosdevelocidadesTab);
2400     app.w1EditFieldLabel.HorizontalAlignment = 'right';
2401     app.w1EditFieldLabel.Position = [31 94 25 22];
2402     app.w1EditFieldLabel.Text = 'w1';
2403
2404     % Create w1EditField
2405     app.w1EditField = ...
2406         uieditfield(app.TringulosdevelocidadesTab, 'numeric');
2407     app.w1EditField.Position = [71 94 58 22];
2408
2409     % Create c2EditFieldLabel
2410     app.c2EditFieldLabel = uilabel(app.TringulosdevelocidadesTab);
2411     app.c2EditFieldLabel.HorizontalAlignment = 'right';
2412     app.c2EditFieldLabel.Position = [156 125 25 22];
2413     app.c2EditFieldLabel.Text = 'c2';
2414
2415     % Create c2EditField
2416     app.c2EditField = ...
2417         uieditfield(app.TringulosdevelocidadesTab, 'numeric');
2418     app.c2EditField.Position = [196 125 58 22];
```



```
2419
2420     % Create w2EditFieldLabel
2421     app.w2EditFieldLabel = uilabel(app.TringulosdevelocidadesTab);
2422     app.w2EditFieldLabel.HorizontalAlignment = 'right';
2423     app.w2EditFieldLabel.Position = [156 93 25 22];
2424     app.w2EditFieldLabel.Text = 'w2';
2425
2426     % Create w2EditField
2427     app.w2EditField = ...
2428         uieditfield(app.TringulosdevelocidadesTab, 'numeric');
2429     app.w2EditField.Position = [196 93 58 22];
2430
2431     % Create uEditFieldLabel
2432     app.uEditFieldLabel = uilabel(app.TringulosdevelocidadesTab);
2433     app.uEditFieldLabel.HorizontalAlignment = 'right';
2434     app.uEditFieldLabel.Position = [31 58 25 22];
2435     app.uEditFieldLabel.Text = 'u';
2436
2437     % Create uEditField
2438     app.uEditField = ...
2439         uieditfield(app.TringulosdevelocidadesTab, 'numeric');
2440     app.uEditField.Position = [71 58 58 22];
2441
2442     % Create DeflexinrotorLabel
2443     app.DeflexinrotorLabel = uilabel(app.TringulosdevelocidadesTab);
2444     app.DeflexinrotorLabel.HorizontalAlignment = 'right';
2445     app.DeflexinrotorLabel.Position = [49 281 99 22];
2446     app.DeflexinrotorLabel.Text = 'Deflexi n rotor ( )';
2447
2448     % Create deflexion_rotor
2449     app.deflexion_rotor = ...
2450         uieditfield(app.TringulosdevelocidadesTab, 'numeric');
2451     app.deflexion_rotor.Position = [168 281 47 22];
2452
2453     % Create DeflexinestatorLabel
2454     app.DeflexinestatorLabel = uilabel(app.TringulosdevelocidadesTab);
2455     app.DeflexinestatorLabel.HorizontalAlignment = 'right';
2456     app.DeflexinestatorLabel.Position = [37 238 111 22];
2457     app.DeflexinestatorLabel.Text = 'Deflexi n estator ( )';
2458
2459     % Create deflexion_estator
2460     app.deflexion_estator = ...
2461         uieditfield(app.TringulosdevelocidadesTab, 'numeric');
2462     app.deflexion_estator.Position = [168 238 47 22];
2463
2464     % Create vaxialEditFieldLabel
2465     app.vaxialEditFieldLabel = uilabel(app.TringulosdevelocidadesTab);
2466     app.vaxialEditFieldLabel.HorizontalAlignment = 'right';
2467     app.vaxialEditFieldLabel.Position = [137 56 41 22];
2468     app.vaxialEditFieldLabel.Text = 'v. axial';
2469
2470     % Create vaxialEditField
2471     app.vaxialEditField = ...
2472         uieditfield(app.TringulosdevelocidadesTab, 'numeric');
2473     app.vaxialEditField.Position = [195 56 59 22];
```

```
2474
2475     % Create labesTab
2476     app.labesTab = uitab(app.TabGroup2);
2477     app.labesTab.Title = ' labes ';
2478
2479     % Create UIAxes2
2480     app.UIAxes2 = uiaxes(app.labesTab);
2481     title(app.UIAxes2, 'geometr a adimensional de la cascada ')
2482     xlabel(app.UIAxes2, 'u^2')
2483     ylabel(app.UIAxes2, 'u^2')
2484     zlabel(app.UIAxes2, 'Z')
2485     app.UIAxes2.Position = [17 203 595 219];
2486
2487     % Create ngulosdellabeLabel
2488     app.ngulosdellabeLabel = uilabel(app.labesTab);
2489     app.ngulosdellabeLabel.FontWeight = 'bold';
2490     app.ngulosdellabeLabel.Position = [266 175 123 22];
2491     app.ngulosdellabeLabel.Text = ' ngulos del labe ( )';
2492
2493     % Create Label_2
2494     app.Label_2 = uilabel(app.labesTab);
2495     app.Label_2.HorizontalAlignment = 'right';
2496     app.Label_2.Position = [72 139 25 22];
2497     app.Label_2.Text = ' 1 ''';
2498
2499     % Create beta_1_p
2500     app.beta_1_p = uieditfield(app.labesTab, 'numeric');
2501     app.beta_1_p.Position = [112 139 47 22];
2502
2503     % Create beta_2_p
2504     app.beta_2_p = uieditfield(app.labesTab, 'numeric');
2505     app.beta_2_p.Position = [242 137 47 22];
2506
2507     % Create EditField_3Label_2
2508     app.EditField_3Label_2 = uilabel(app.labesTab);
2509     app.EditField_3Label_2.HorizontalAlignment = 'right';
2510     app.EditField_3Label_2.Position = [202 137 25 22];
2511     app.EditField_3Label_2.Text = ' 2 ''';
2512
2513     % Create EditField_2Label_2
2514     app.EditField_2Label_2 = uilabel(app.labesTab);
2515     app.EditField_2Label_2.HorizontalAlignment = 'right';
2516     app.EditField_2Label_2.Position = [330 137 25 22];
2517     app.EditField_2Label_2.Text = ' 2 ''';
2518
2519     % Create alpha_2_p
2520     app.alpha_2_p = uieditfield(app.labesTab, 'numeric');
2521     app.alpha_2_p.Position = [370 137 47 22];
2522
2523     % Create alpha_3_p
2524     app.alpha_3_p = uieditfield(app.labesTab, 'numeric');
2525     app.alpha_3_p.Position = [504 137 47 22];
2526
2527     % Create EditField_4Label_2
2528     app.EditField_4Label_2 = uilabel(app.labesTab);
```

```
2529     app.EditField_4Label_2.HorizontalAlignment = 'right';
2530     app.EditField_4Label_2.Position = [464 137 25 22];
2531     app.EditField_4Label_2.Text = ' 3  ';
2532
2533     % Create DesviacinrotorEditFieldLabel
2534     app.DesviacinrotorEditFieldLabel = uilabel(app.labesTab);
2535     app.DesviacinrotorEditFieldLabel.HorizontalAlignment = 'right';
2536     app.DesviacinrotorEditFieldLabel.Position = [45 81 95 22];
2537     app.DesviacinrotorEditFieldLabel.Text = 'Desviaci n rotor ';
2538
2539     % Create DesviacinrotorEditField
2540     app.DesviacinrotorEditField = uieditfield(app.labesTab, 'numeric');
2541     app.DesviacinrotorEditField.Position = [155 81 44 22];
2542
2543     % Create DesviacinestatorEditFieldLabel
2544     app.DesviacinestatorEditFieldLabel = uilabel(app.labesTab);
2545     app.DesviacinestatorEditFieldLabel.HorizontalAlignment = 'right';
2546     app.DesviacinestatorEditFieldLabel.Position = [36 39 104 22];
2547     app.DesviacinestatorEditFieldLabel.Text = 'Desviaci n estator';
2548
2549     % Create DesviacinestatorEditField
2550     app.DesviacinestatorEditField = uieditfield(app.labesTab, 'numeric');
2551     app.DesviacinestatorEditField.Position = [155 39 44 22];
2552
2553     % Create CurvaturarotorEditFieldLabel
2554     app.CurvaturarotorEditFieldLabel = uilabel(app.labesTab);
2555     app.CurvaturarotorEditFieldLabel.HorizontalAlignment = 'right';
2556     app.CurvaturarotorEditFieldLabel.Position = [229 81 86 22];
2557     app.CurvaturarotorEditFieldLabel.Text = 'Curvatura rotor';
2558
2559     % Create CurvaturarotorEditField
2560     app.CurvaturarotorEditField = uieditfield(app.labesTab, 'numeric');
2561     app.CurvaturarotorEditField.Position = [330 81 44 22];
2562
2563     % Create CurvaturaestatorEditFieldLabel
2564     app.CurvaturaestatorEditFieldLabel = uilabel(app.labesTab);
2565     app.CurvaturaestatorEditFieldLabel.HorizontalAlignment = 'right';
2566     app.CurvaturaestatorEditFieldLabel.Position = [217 40 98 22];
2567     app.CurvaturaestatorEditFieldLabel.Text = 'Curvatura estator';
2568
2569     % Create CurvaturaestatorEditField
2570     app.CurvaturaestatorEditField = uieditfield(app.labesTab, 'numeric');
2571     app.CurvaturaestatorEditField.Position = [330 40 44 22];
2572
2573     % Create CaladorotorEditFieldLabel
2574     app.CaladorotorEditFieldLabel = uilabel(app.labesTab);
2575     app.CaladorotorEditFieldLabel.HorizontalAlignment = 'right';
2576     app.CaladorotorEditFieldLabel.Position = [418 79 71 22];
2577     app.CaladorotorEditFieldLabel.Text = 'Calado rotor';
2578
2579     % Create CaladorotorEditField
2580     app.CaladorotorEditField = uieditfield(app.labesTab, 'numeric');
2581     app.CaladorotorEditField.Position = [504 79 44 22];
2582
2583     % Create CaladoestatorEditFieldLabel
```

```

2584     app.CaladoestatorEditFieldLabel = uilabel(app.labesTab);
2585     app.CaladoestatorEditFieldLabel.HorizontalAlignment = 'right';
2586     app.CaladoestatorEditFieldLabel.Position = [406 40 83 22];
2587     app.CaladoestatorEditFieldLabel.Text = 'Calado estator';
2588
2589     % Create CaladoestatorEditField
2590     app.CaladoestatorEditField = uieditfield(app.labesTab, 'numeric');
2591     app.CaladoestatorEditField.Position = [504 40 44 22];
2592
2593     % Create GeometraTab
2594     app.GeometraTab = uitab(app.TabGroup2);
2595     app.GeometraTab.Title = 'Geometr a';
2596
2597     % Create geometria_rot
2598     app.geometria_rot = uitable(app.GeometraTab);
2599     app.geometria_rot.ColumnName = {'Escalonamientos'; ...
2600 'Altura del labe (cm)'; 'Paso (cm)'; 'Cuerda (cm)'; 'N de labes '};
2601     app.geometria_rot.RowName = {};
2602     app.geometria_rot.Position = [25 242 590 162];
2603
2604     % Create ROTORLabel
2605     app.ROTORLabel = uilabel(app.GeometraTab);
2606     app.ROTORLabel.FontWeight = 'bold';
2607     app.ROTORLabel.Position = [290 405 48 22];
2608     app.ROTORLabel.Text = 'ROTOR';
2609
2610     % Create geometria_est
2611     app.geometria_est = uitable(app.GeometraTab);
2612     app.geometria_est.ColumnName = {'Escalonamientos'; ...
2613 'Altura del labe (cm)'; 'Paso (cm)'; 'Cuerda (cm)'; 'N de labes '};
2614     app.geometria_est.RowName = {};
2615     app.geometria_est.Position = [25 24 590 162];
2616
2617     % Create ESTATORLabel
2618     app.ESTATORLabel = uilabel(app.GeometraTab);
2619     app.ESTATORLabel.FontWeight = 'bold';
2620     app.ESTATORLabel.Position = [284 196 60 22];
2621     app.ESTATORLabel.Text = 'ESTATOR';
2622
2623     % Create GeometracompresorTab
2624     app.GeometracompresorTab = uitab(app.TabGroup2);
2625     app.GeometracompresorTab.Title = 'Geometr a compresor';
2626
2627     % Create UIAxes7
2628     app.UIAxes7 = uiaxes(app.GeometracompresorTab);
2629     title(app.UIAxes7, 'Geometria del compresor')
2630     xlabel(app.UIAxes7, 'Cascadas')
2631     ylabel(app.UIAxes7, 'Altura (m)')
2632     zlabel(app.UIAxes7, 'Z')
2633     app.UIAxes7.Position = [9 66 600 347];
2634
2635     % Create Label_7
2636     app.Label_7 = uilabel(app.GeometracompresorTab);
2637     app.Label_7.Position = [63 24 525 22];
2638     app.Label_7.Text = ...

```

```

2639         ['Recordatorio: por cada escalonamiento ...' ...
2640         'existen dos cascadas, una para rotor y otra para estator'];
2641
2642         % Create PrdidasTab
2643         app.PrdidasTab = uitab(app.TabGroup2);
2644         app.PrdidasTab.Title = 'P rdidas';
2645
2646         % Create perdidas_rot
2647         app.perdidas_rot = uitable(app.PrdidasTab);
2648         app.perdidas_rot.ColumnName = {'Escalonamientos'; ...
2649 'coef. Perfil '; 'coef. Anulares'; 'coef. Secundarias'; 'P rdidas (J/kg)'};
2650         app.perdidas_rot.RowName = {};
2651         app.perdidas_rot.Position = [24 242 590 162];
2652
2653         % Create ROTORLabel_2
2654         app.ROTORLabel_2 = uilabel(app.PrdidasTab);
2655         app.ROTORLabel_2.FontWeight = 'bold';
2656         app.ROTORLabel_2.Position = [290 405 48 22];
2657         app.ROTORLabel_2.Text = 'ROTOR';
2658
2659         % Create ESTATORLabel_2
2660         app.ESTATORLabel_2 = uilabel(app.PrdidasTab);
2661         app.ESTATORLabel_2.FontWeight = 'bold';
2662         app.ESTATORLabel_2.Position = [284 217 60 22];
2663         app.ESTATORLabel_2.Text = 'ESTATOR';
2664
2665         % Create perdidas_est
2666         app.perdidas_est = uitable(app.PrdidasTab);
2667         app.perdidas_est.ColumnName = {'Escalonamientos'; ...
2668 'coef. Perfil'; 'coef. Anulares'; 'coef. Secundarias'; 'P rdidas (J/kg)'};
2669         app.perdidas_est.RowName = {};
2670         app.perdidas_est.Position = [25 41 590 176];
2671
2672         % Create Tampocodebesermayora195Label
2673         app.Tampocodebesermayora195Label = uilabel(app.PrdidasTab);
2674         app.Tampocodebesermayora195Label.Position = [343 8 185 22];
2675         app.Tampocodebesermayora195Label.Text = ...
2676         '(Tampoco debe ser mayor a 1.95)';
2677
2678         % Create RelacindedifusinestatorEditFieldLabel
2679         app.RelacindedifusinestatorEditFieldLabel = ...
2680         uilabel(app.PrdidasTab);
2681         app.RelacindedifusinestatorEditFieldLabel.HorizontalAlignment = 'right';
2682         app.RelacindedifusinestatorEditFieldLabel.Position = ...
2683         [58 8 153 22];
2684         app.RelacindedifusinestatorEditFieldLabel.Text = ...
2685         'Relaci n de difusi n estator';
2686
2687         % Create RelacindedifusinestatorEditField
2688         app.RelacindedifusinestatorEditField = ...
2689         uieditfield(app.PrdidasTab, 'numeric');
2690         app.RelacindedifusinestatorEditField.Editable = 'off';
2691         app.RelacindedifusinestatorEditField.Position = [226 8 100 22];
2692         % Create PresionesTab
2693         app.PresionesTab = uitab(app.TabGroup2);

```

```

2694     app.PresionesTab.Title = 'Presiones';
2695
2696     % Create UIAxes3
2697     app.UIAxes3 = uiaxes(app.PresionesTab);
2698     title(app.UIAxes3, 'Presiones totales')
2699     xlabel(app.UIAxes3, 'Escalonamiento')
2700     ylabel(app.UIAxes3, 'Presi n (KPa)')
2701     zlabel(app.UIAxes3, 'Z')
2702     app.UIAxes3.Position = [326 26 298 381];
2703
2704     % Create UIAxes3_2
2705     app.UIAxes3_2 = uiaxes(app.PresionesTab);
2706     title(app.UIAxes3_2, 'Presiones')
2707     xlabel(app.UIAxes3_2, 'Escalonamiento')
2708     ylabel(app.UIAxes3_2, 'Presi n (KPa)')
2709     zlabel(app.UIAxes3_2, 'Z')
2710     app.UIAxes3_2.Position = [9 26 296 381];
2711
2712     % Create TemperaturasTab
2713     app.TemperaturasTab = uitab(app.TabGroup2);
2714     app.TemperaturasTab.Title = 'Temperaturas';
2715
2716     % Create UIAxes3_3
2717     app.UIAxes3_3 = uiaxes(app.TemperaturasTab);
2718     title(app.UIAxes3_3, 'Temperaturas')
2719     xlabel(app.UIAxes3_3, 'Escalonamiento')
2720     ylabel(app.UIAxes3_3, 'Temperatura (K)')
2721     zlabel(app.UIAxes3_3, 'Z')
2722     app.UIAxes3_3.Position = [9 26 296 381];
2723
2724     % Create UIAxes3_4
2725     app.UIAxes3_4 = uiaxes(app.TemperaturasTab);
2726     title(app.UIAxes3_4, 'Temperaturas totales')
2727     xlabel(app.UIAxes3_4, 'Escalonamiento')
2728     ylabel(app.UIAxes3_4, 'Temperatura (K)')
2729     zlabel(app.UIAxes3_4, 'Z')
2730     app.UIAxes3_4.Position = [325 26 287 381];
2731
2732     % Create DensidadesTab
2733     app.DensidadesTab = uitab(app.TabGroup2);
2734     app.DensidadesTab.Title = 'Densidades';
2735
2736     % Create UIAxes3_5
2737     app.UIAxes3_5 = uiaxes(app.DensidadesTab);
2738     title(app.UIAxes3_5, 'Densidades')
2739     xlabel(app.UIAxes3_5, 'Escalonamiento')
2740     ylabel(app.UIAxes3_5, 'Densidades (kg/m^3)')
2741     zlabel(app.UIAxes3_5, 'Z')
2742     app.UIAxes3_5.Position = [9 26 296 381];
2743
2744     % Create UIAxes3_6
2745     app.UIAxes3_6 = uiaxes(app.DensidadesTab);
2746     title(app.UIAxes3_6, 'Calor espec fico')
2747     xlabel(app.UIAxes3_6, 'Escalonamiento')
2748     ylabel(app.UIAxes3_6, 'Calor espec fico (KJ(Kg*K))')

```

```

2749     xlabel(app.UIAxes3_6, 'Z')
2750     app.UIAxes3_6.Position = [326 25 296 381];
2751
2752     % Create PrestacionesTab
2753     app.PrestacionesTab = uitab(app.TabGroup2);
2754     app.PrestacionesTab.Title = 'Prestaciones';
2755
2756     % Create UIAxes8
2757     app.UIAxes8 = uiaxes(app.PrestacionesTab);
2758     title(app.UIAxes8, 'Relaci n de compresi n')
2759     xlabel(app.UIAxes8, 'Escalonamiento')
2760     ylabel(app.UIAxes8, 'Relaci n de compresi n')
2761     xlabel(app.UIAxes8, 'Z')
2762     app.UIAxes8.Position = [26 257 578 163];
2763
2764     % Create UIAxes8_2
2765     app.UIAxes8_2 = uiaxes(app.PrestacionesTab);
2766     title(app.UIAxes8_2, ' Rendimiento por escalonamiento')
2767     xlabel(app.UIAxes8_2, 'Escalonamiento')
2768     ylabel(app.UIAxes8_2, 'Rendimiento')
2769     xlabel(app.UIAxes8_2, 'Z')
2770     app.UIAxes8_2.Position = [28 84 578 163];
2771
2772     % Create TrabajoespecificoporescalonamientoKJKgKLabel
2773     app.TrabajoespecificoporescalonamientoKJKgKLabel = ...
2774         uilabel(app.PrestacionesTab);
2775     app.TrabajoespecificoporescalonamientoKJKgKLabel.HorizontalAlignment = ...
2776         'right';
2777     app.TrabajoespecificoporescalonamientoKJKgKLabel.Position = ...
2778         [22 26 256 22];
2779     app.TrabajoespecificoporescalonamientoKJKgKLabel.Text = ...
2780         'Trabajo espec fico por escalonamiento (KJ/Kg)';
2781
2782     % Create TrabajoespecificoporescalonamientoKJKgEditField
2783     app.TrabajoespecificoporescalonamientoKJKgEditField = ...
2784         uieditfield(app.PrestacionesTab, 'numeric');
2785     app.TrabajoespecificoporescalonamientoKJKgEditField.Position = ...
2786         [293 26 78 22];
2787
2788     % Create RendimientototalEditFieldLabel
2789     app.RendimientototalEditFieldLabel = uilabel(app.PrestacionesTab);
2790     app.RendimientototalEditFieldLabel.HorizontalAlignment = 'right';
2791     app.RendimientototalEditFieldLabel.Position = [420 24 98 22];
2792     app.RendimientototalEditFieldLabel.Text = 'Rendimiento total';
2793
2794     % Create RendimientototalEditField
2795     app.RendimientototalEditField = ...
2796         uieditfield(app.PrestacionesTab, 'numeric');
2797     app.RendimientototalEditField.Position = [533 24 62 22];
2798
2799     % Create LeytorsionalTab
2800     app.LeytorsionalTab = uitab(app.TabGroup);
2801     app.LeytorsionalTab.Title = 'Ley torsional';
2802
2803     % Create CASOTRIDIMENSIONALLabel

```



```
2804     app.CASOTRIDIMENSIONALLabel = uilabel(app.LeytorsionalTab);
2805     app.CASOTRIDIMENSIONALLabel.FontSize = 14;
2806     app.CASOTRIDIMENSIONALLabel.FontWeight = 'bold';
2807     app.CASOTRIDIMENSIONALLabel.Position = [248 417 170 22];
2808     app.CASOTRIDIMENSIONALLabel.Text = 'CASO TRIDIMENSIONAL';
2809
2810     % Create Label_3
2811     app.Label_3 = uilabel(app.LeytorsionalTab);
2812     app.Label_3.Position = [70 374 518 22];
2813     app.Label_3.Text = ['Para resolver el caso tridimensional, ...' ...
2814     'debe seleccionar una de las siguientes leyes torsionales: '];
2815
2816     % Create torbellino
2817     app.torbellino = uibutton(app.LeytorsionalTab, 'push');
2818     app.torbellino.ButtonPushedFcn = ...
2819         createCallbackFcn(app, @torbellinoButtonPushed, true);
2820     app.torbellino.Position = [78 203 100 66];
2821     app.torbellino.Text = 'CALCULAR';
2822
2823     % Create reaccion
2824     app.reaccion = uibutton(app.LeytorsionalTab, 'push');
2825     app.reaccion.ButtonPushedFcn = ...
2826         createCallbackFcn(app, @reaccionButtonPushed, true);
2827     app.reaccion.Position = [281 203 100 66];
2828     app.reaccion.Text = 'CALCULAR';
2829
2830     % Create exponencial
2831     app.exponencial = uibutton(app.LeytorsionalTab, 'push');
2832     app.exponencial.ButtonPushedFcn = ...
2833         createCallbackFcn(app, @exponencialButtonPushed, true);
2834     app.exponencial.Position = [477 203 100 66];
2835     app.exponencial.Text = 'CALCULAR';
2836
2837     % Create LeydetorbellinolibreLabel
2838     app.LeydetorbellinolibreLabel = uilabel(app.LeytorsionalTab);
2839     app.LeydetorbellinolibreLabel.FontWeight = 'bold';
2840     app.LeydetorbellinolibreLabel.Position = [61 302 130 22];
2841     app.LeydetorbellinolibreLabel.Text = 'Ley de torbellino libre';
2842
2843     % Create LeyexponencialLabel
2844     app.LeyexponencialLabel = uilabel(app.LeytorsionalTab);
2845     app.LeyexponencialLabel.FontWeight = 'bold';
2846     app.LeyexponencialLabel.Position = [479 303 98 22];
2847     app.LeyexponencialLabel.Text = 'Ley exponencial';
2848
2849     % Create LeydegradodereaccinconstanteLabel
2850     app.LeydegradodereaccinconstanteLabel = uilabel(app.LeytorsionalTab);
2851     app.LeydegradodereaccinconstanteLabel.FontWeight = 'bold';
2852     app.LeydegradodereaccinconstanteLabel.Position = [223 302 211 22];
2853     app.LeydegradodereaccinconstanteLabel.Text = ...
2854         'Ley de grado de reacci n constante';
2855
2856     % Create coefflujocrticoEditFieldLabel
2857     app.coefflujocrticoEditFieldLabel = uilabel(app.LeytorsionalTab);
2858     app.coefflujocrticoEditFieldLabel.HorizontalAlignment = 'right';
```



```

2859     app.coeflujocrticoEditFieldLabel.Position = [14 59 554 22];
2860     app.coeflujocrticoEditFieldLabel.Text = ...
2861         ['coeficiente de flujo a la cabeza del primer...' ...
2862         ' escalonamiento para la ley de grado de reacci n constante'];
2863
2864     % Create coeficientedeflujoalacabezadelprimerescalonamientoEditField
2865     app.coeficientedeflujoalacabezadelprimerescalonamientoEditField =...
2866         uieditfield(app.LeytorsionalTab, 'numeric');
2867     app.coeficientedeflujoalacabezadelprimerescalonamientoEditField.Position...
2868         = [305 26 47 22];
2869
2870     % Create Button
2871     app.Button = uibutton(app.LeytorsionalTab, 'push');
2872     app.Button.ButtonPushedFcn = createCallbackFcn(app,...
2873         @ButtonPushed, true);
2874     app.Button.Position = [582 58 29 23];
2875     app.Button.Text = '?';
2876
2877     % Create Resultados3DTab
2878     app.Resultados3DTab = uitab(app.TabGroup);
2879     app.Resultados3DTab.Title = 'Resultados 3D';
2880
2881     % Create TabGroup3
2882     app.TabGroup3 = uitabgroup(app.Resultados3DTab);
2883     app.TabGroup3.Position = [2 74 636 382];
2884
2885     % Create Tri ngulosdevelocidadesTab_2
2886     app.Tri ngulosdevelocidadesTab_2 = uitab(app.TabGroup3);
2887     app.Tri ngulosdevelocidadesTab_2.Title = ...
2888         'Tri ngulos de velocidades';
2889
2890     % Create triangulos_3D
2891     app.triangulos_3D = uiaxes(app.Tri ngulosdevelocidadesTab_2);
2892     title(app.triangulos_3D, 'Tri ngulos de velocidades 3D')
2893     xlabel(app.triangulos_3D, 'm/s')
2894     ylabel(app.triangulos_3D, 'm/s')
2895     zlabel(app.triangulos_3D, 'Z')
2896     app.triangulos_3D.Position = [8 53 603 296];
2897
2898     % Create NmerodeMachMximoEditFieldLabel
2899     app.NmerodeMachMximoEditFieldLabel =...
2900         uilabel(app.Tri ngulosdevelocidadesTab_2);
2901     app.NmerodeMachMximoEditFieldLabel.HorizontalAlignment = 'right';
2902     app.NmerodeMachMximoEditFieldLabel.Position = [186 20 142 22];
2903     app.NmerodeMachMximoEditFieldLabel.Text = 'N mero de Mach M ximo';
2904
2905     % Create Mach_Maximo
2906     app.Mach_Maximo = ...
2907         uieditfield(app.Tri ngulosdevelocidadesTab_2, 'numeric');
2908     app.Mach_Maximo.Position = [343 20 49 22];
2909
2910     % Create Ley
2911     app.Ley = uieditfield(app.Tri ngulosdevelocidadesTab_2, 'text');
2912     app.Ley.Editable = 'off';
2913     app.Ley.FontWeight = 'bold';

```

```

2914     app.Ley.Position = [63 332 135 22];
2915
2916     % Create PrdidasTab_2
2917     app.PrdidasTab_2 = uitab(app.TabGroup3);
2918     app.PrdidasTab_2.Title = 'P rddidas';
2919
2920     % Create perdidas_3D
2921     app.perdidas_3D = uitable(app.PrdidasTab_2);
2922     app.perdidas_3D.ColumnName = {'Radio'; ...
2923     'P rd. r tor (KJ/Kg)'; 'P rd. est tor (KJ/Kg)';...
2924     'Trabajo espec fico (KJ/Kg)'; 'Rendimiento total a total'};
2925     app.perdidas_3D.RowName = {};
2926     app.perdidas_3D.Position = [25 164 591 111];
2927
2928     % Create PRDIDASTRABAJORENDIMIENTOLabel
2929     app.PRIDASTRABAJORENDIMIENTOLabel = uilabel(app.PrdidasTab_2);
2930     app.PRIDASTRABAJORENDIMIENTOLabel.FontSize = 14;
2931     app.PRIDASTRABAJORENDIMIENTOLabel.FontWeight = 'bold';
2932     app.PRIDASTRABAJORENDIMIENTOLabel.Position = [185 322 266 22];
2933     app.PRIDASTRABAJORENDIMIENTOLabel.Text = ...
2934     'P RDIDAS ,TRABAJO Y RENDIMIENTO';
2935
2936     % Create TrabajoespecificoporescalonamientoKJkgEditFieldLabel
2937     app.TrabajoespecificoporescalonamientoKJkgEditFieldLabel = ...
2938     uilabel(app.PrdidasTab_2);
2939     app.TrabajoespecificoporescalonamientoKJkgEditFieldLabel.HorizontalAlignment...
2940     = 'right';
2941     app.TrabajoespecificoporescalonamientoKJkgEditFieldLabel.Position =...
2942     [105 102 254 22];
2943     app.TrabajoespecificoporescalonamientoKJkgEditFieldLabel.Text =...
2944     'Trabajo espec fico por escalonamiento (KJ/kg)';
2945
2946     % Create TrabajoespecificoporescalonamientoKJkgEditField
2947     app.TrabajoespecificoporescalonamientoKJkgEditField =...
2948     uieditfield(app.PrdidasTab_2, 'numeric');
2949     app.TrabajoespecificoporescalonamientoKJkgEditField.Position =...
2950     [373 102 100 22];
2951
2952     % Create RendimientototaldelescalonamientoEditFieldLabel
2953     app.RendimientototaldelescalonamientoEditFieldLabel =...
2954     uilabel(app.PrdidasTab_2);
2955     app.RendimientototaldelescalonamientoEditFieldLabel.HorizontalAlignment=...
2956     'right';
2957     app.RendimientototaldelescalonamientoEditFieldLabel.Position =...
2958     [152 53 206 22];
2959     app.RendimientototaldelescalonamientoEditFieldLabel.Text =...
2960     'Rendimiento total del escalonamiento';
2961
2962     % Create RendimientototaldelescalonamientoEditField
2963     app.RendimientototaldelescalonamientoEditField =...
2964     uieditfield(app.PrdidasTab_2, 'numeric');
2965     app.RendimientototaldelescalonamientoEditField.Position =...
2966     [373 53 100 22];
2967
2968     % Create ngulosTab

```

```

2969     app.ngulosTab = uitab(app.TabGroup3);
2970     app.ngulosTab.Title = ' ngulos ';
2971
2972     % Create UIAxes6
2973     app.UIAxes6 = uiaxes(app.ngulosTab);
2974     title(app.UIAxes6, 'Variaci n de los ngulos ');
2975     xlabel(app.UIAxes6, 'r/r_{cabeza}')
2976     ylabel(app.UIAxes6, ' ')
2977     zlabel(app.UIAxes6, 'Z')
2978     app.UIAxes6.Position = [39 29 534 311];
2979
2980     % Create Ley_2
2981     app.Ley_2 = uieditfield(app.ngulosTab, 'text');
2982     app.Ley_2.Editable = 'off';
2983     app.Ley_2.FontWeight = 'bold';
2984     app.Ley_2.Position = [88 322 135 22];
2985
2986     % Create CoeficientesTab
2987     app.CoefficientesTab = uitab(app.TabGroup3);
2988     app.CoefficientesTab.Title = 'Coeficientes';
2989
2990     % Create UIAxes6_2
2991     app.UIAxes6_2 = uiaxes(app.CoefficientesTab);
2992     title(app.UIAxes6_2, 'Variaci n de los coeficientes')
2993     xlabel(app.UIAxes6_2, 'r/r_{cabeza}')
2994     zlabel(app.UIAxes6_2, 'Z')
2995     app.UIAxes6_2.Position = [39 29 534 310];
2996
2997     % Create Ley_3
2998     app.Ley_3 = uieditfield(app.CoefficientesTab, 'text');
2999     app.Ley_3.Editable = 'off';
3000     app.Ley_3.FontWeight = 'bold';
3001     app.Ley_3.Position = [87 321 135 22];
3002
3003     % Create EscalonamientoanteriorButton
3004     app.EscalonamientoanteriorButton = uibutton...
3005         (app.Resultados3DTab, 'push');
3006     app.EscalonamientoanteriorButton.ButtonPushedFcn =...
3007     createCallbackFcn(app, @EscalonamientoanteriorButtonPushed2, true);
3008     app.EscalonamientoanteriorButton.Position = [60 21 144 23];
3009     app.EscalonamientoanteriorButton.Text = 'Escalonamiento anterior';
3010
3011     % Create EscalonamientonLabel_2
3012     app.EscalonamientonLabel_2 = uilabel(app.Resultados3DTab);
3013     app.EscalonamientonLabel_2.Position = [254 21 105 22];
3014     app.EscalonamientonLabel_2.Text = 'Escalonamiento n ';
3015
3016     % Create escalonamientos
3017     app.escalonamientos = uieditfield(app.Resultados3DTab, 'numeric');
3018     app.escalonamientos.Editable = 'off';
3019     app.escalonamientos.BackgroundColor = [0.902 0.902 0.902];
3020     app.escalonamientos.Position = [363 21 29 22];
3021
3022     % Create SiguienteescalonamientoButton
3023     app.SiguienteescalonamientoButton =...

```

```

3024         uibutton(app.Resultados3DTab, 'push');
3025     app.SiguienteescalonamientoButton.ButtonPushedFcn = ...
3026         createCallbackFcn(app,...
3027             @SiguienteescalonamientoButtonPushed, true);
3028     app.SiguienteescalonamientoButton.Position = [442 21 152 23];
3029     app.SiguienteescalonamientoButton.Text = ...
3030         'Siguiente escalonamiento';
3031
3032     % Create ley_actualEditFieldLabel
3033     app.ley_actualEditFieldLabel = uilabel(app.UIFigure);
3034     app.ley_actualEditFieldLabel.HorizontalAlignment = 'right';
3035     app.ley_actualEditFieldLabel.Position = [1 -54 59 22];
3036     app.ley_actualEditFieldLabel.Text = 'ley_actual';
3037
3038     % Create ley_actual
3039     app.ley_actual = uieditfield(app.UIFigure, 'numeric');
3040     app.ley_actual.Position = [75 -54 36 22];
3041
3042     % Create NmerodepuntosEditFieldLabel
3043     app.NmerodepuntosEditFieldLabel = uilabel(app.UIFigure);
3044     app.NmerodepuntosEditFieldLabel.HorizontalAlignment = 'right';
3045     app.NmerodepuntosEditFieldLabel.Position = [465 -54 107 22];
3046     app.NmerodepuntosEditFieldLabel.Text = 'N mero de puntos ';
3047
3048     % Create n_puntos
3049     app.n_puntos = uieditfield(app.UIFigure, 'numeric');
3050     app.n_puntos.Limits = [3 100];
3051     app.n_puntos.Position = [597 -54 43 22];
3052     app.n_puntos.Value = 9;
3053
3054     % Show the figure after all components are created
3055     app.UIFigure.Visible = 'on';
3056     end
3057 end
3058
3059 % App creation and deletion
3060 methods (Access = public)
3061
3062     % Construct app
3063     function app = compresor_axial_exported
3064
3065         % Create UIFigure and components
3066         createComponents(app)
3067
3068         % Register the app with App Designer
3069         registerApp(app, app.UIFigure)
3070
3071         if nargin == 0
3072             clear app
3073         end
3074     end
3075
3076     % Code that executes before app deletion
3077     function delete(app)
3078

```

```
3079         % Delete UIFigure when app is deleted
3080         delete(app.UIFigure)
3081     end
3082 end
3083 end
```

Bibliografía

- [1] https://es.wikipedia.org/wiki/M%C3%A1quina_de_vapor.
- [2] <https://www.ge.com/gas-power/products/steam-turbines>.
- [3] Ismael Callejón Agramunt Jesús Andrés Álvarez Flórez. *Máquinas Térmicas Motoras*. Politext.
- [4] <https://webpages.charlotte.edu/gkumar2/HW4.html>.
- [5] A. Zelentsov V. Minin L. Finkelberg, A. Kostuchenkov. Improvement of combustion process of spark-ignited aviation wankel engine.
- [6] D.H. Robison and P. J. Beaty. Compressor types, classifications, and applications.
- [7] <https://www.turbomachinerymag.com/view/parsons-designs-first-axial-flow-compressor-2>.
- [8] José Antonio Vélez Godiño. Apuntes de la asignatura *Sistemas de Propulsión Aeroespacial*.
- [9] Muñoz Torralbo & Valdés del Fresno & Muñoz Domínguez. *Turbomáquinas térmicas: fundamentos del diseño termodinámico*. Universidad Politécnica de Madrid, 2001.
- [10] P. Hill C. Peterson. *Mechanics and thermodynamics of propulsion*. Pearson.
- [11] M. P. Boyce. *Gas Turbine Engineering Handbook*. Gulf Professional Publishing, 2012.
- [12] Tomás Sánchez Lencero. *Turbomáquinas térmicas*. Síntesis.
- [13] K. Wark. *Thermodynamics*. McGraw-Hill.
- [14] Royce N. Brown. *Compressors*. 1997.
- [15] E. Benini. State of art of transonic axial compressors.