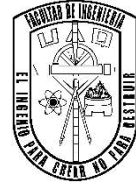


Universidad Autónoma de Querétaro  
Facultad de Ingeniería



OPTIMIZACIÓN DE UN PERFIL AERODINÁMICO DE LA PALA DE UNA  
TURBINA EÓLICA DE BAJA POTENCIA MEDIANTE PSO Y CFD

Opción de titulación  
**Tesis**

Que como parte de los requisitos para obtener el grado de  
MAESTRO EN CIENCIAS (MECATRÓNICA)

**Presenta:**  
Ing. Luis Mauricio García Noverola

**Dirigido por:**  
Dr. Aurelio Domínguez González

La presente obra está bajo la licencia:  
<https://creativecommons.org/licenses/by-nc-nd/4.0/deed.es>



CC BY-NC-ND 4.0 DEED

Atribución-NoComercial-SinDerivadas 4.0 Internacional

### Usted es libre de:

**Compartir** — copiar y redistribuir el material en cualquier medio o formato

La licenciante no puede revocar estas libertades en tanto usted siga los términos de la licencia

### Bajo los siguientes términos:



**Atribución** — Usted debe dar [crédito de manera adecuada](#), brindar un enlace a la licencia, e [indicar si se han realizado cambios](#). Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo de la licenciante.



**NoComercial** — Usted no puede hacer uso del material con [propósitos comerciales](#).



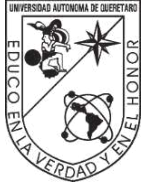
**SinDerivadas** — Si [remezcla, transforma o crea a partir](#) del material, no podrá distribuir el material modificado.

**No hay restricciones adicionales** — No puede aplicar términos legales ni [medidas tecnológicas](#) que restrinjan legalmente a otras a hacer cualquier uso permitido por la licencia.

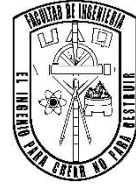
### Avisos:

No tiene que cumplir con la licencia para elementos del material en el dominio público o cuando su uso esté permitido por una [excepción o limitación](#) aplicable.

No se dan garantías. La licencia podría no darle todos los permisos que necesita para el uso que tenga previsto. Por ejemplo, otros derechos como [publicidad, privacidad, o derechos morales](#) pueden limitar la forma en que utilice el material.



**UNIVERSIDAD AUTÓNOMA DE QUERÉTARO**  
**FACULTAD DE INGENIERÍA**  
**CAMPUS SAN JUAN DEL RÍO**



**OPTIMIZACIÓN DE UN PERFIL AERODINÁMICO DE LA PALA DE UNA  
TURBINA EÓLICA DE BAJA POTENCIA MEDIANTE PSO Y CFD**

**TESIS**

Que como parte de los requisitos para obtener el grado de  
**MAESTRO EN CIENCIAS (MECATRÓNICA)**

**Presenta:**

Ing. Luis Mauricio García Noverola

**Dirigido por:**

Dr. Aurelio Domínguez González

**SINODALES**

Dr. Aurelio Domínguez González

Presidente

\_\_\_\_\_  
Firma

Dr. Ángel Pérez Cruz

Secretario

\_\_\_\_\_  
Firma

Dr. Carlos Gustavo Manríquez Padilla

Vocal

\_\_\_\_\_  
Firma

Dr. Isaías Cueva Pérez

Suplente

\_\_\_\_\_  
Firma

Dr. Juan Primo Benítez Rangel

Suplente

\_\_\_\_\_  
Firma

Centro Universitario, Querétaro, Qro.

03 de Junio, 2024

México

## **DEDICATORIA**

A mi familia, por brindarme su apoyo incondicional y cariño en cada etapa de mi vida y, aún a la distancia, su calidez y amor siempre está presente.

A mi querida madre, María Guadalupe de los Ángeles Noverola Muñoz, por su ejemplo y por su incansable amor hacia su familia y sus hijos.

A mi padre, Luis Herminio García Alpuche, por ser mi guía, por su invaluable ejemplo, y por sus consejos cuando más lo necesito.

A mi hermana, Dánae García Noverola, por creer en mí en todo momento y por enviarme mensajes de motivación en los días difíciles.

A mi prometida, mi mejor amiga, mi equipo y mi confidente, Maritza García Cítuc, por estar conmigo en todo momento, por no dejarme caer, por apoyarme incondicionalmente e impulsarme a ser mejor.

Infinitas gracias.

## **AGRADECIMIENTOS**

Al Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONAHCyT) por la beca nacional con número de CVU 1184198.

A la Universidad Autónoma de Querétaro y a la Facultad de Ingeniería por la beca de posgrado con número de expediente 311335.

A mi director de tesis y a mis sinodales, por sus contribuciones para la realización de este trabajo.

A mis profesores, quiénes me brindaron valiosos conocimientos durante mi estancia en la maestría.

A mis compañeros de generación y de cubículo, quiénes hicieron más llevaderas las horas durante el posgrado.

## **INDICE GENERAL**

<b>DEDICATORIA .....</b>	<b>I</b>
<b>AGRADECIMIENTOS .....</b>	<b>II</b>
<b>INDICE GENERAL.....</b>	<b>III</b>
<b>INDICE DE TABLAS .....</b>	<b>VII</b>
<b>INDICE DE FIGURAS .....</b>	<b>VIII</b>
<b>ABREVIATURAS Y SIGLAS.....</b>	<b>XIII</b>
<b>RESUMEN .....</b>	<b>1</b>
<b>ABSTRACT .....</b>	<b>2</b>
<b>I. INTRODUCCIÓN.....</b>	<b>3</b>
1.1 ANTECEDENTES .....	4
1.1.1 Clasificación y características de las turbinas eólicas .....	4
1.1.2 Optimización de turbinas eólicas de eje horizontal.....	8
1.1.2.1 Optimización de la turbina eólica.....	9
1.1.2.2 Optimización de la pala de la turbina.....	9
1.1.2.3 Optimización del perfil aerodinámico .....	13
1.1.3 Optimización de turbinas eólicas de baja potencia.....	14
1.1.4 Optimización de perfiles aerodinámicos para turbinas eólicas .....	18
1.1.4.1 Métodos de parametrización geométrica.....	19
1.1.4.2 Algoritmos Genéticos: Algoritmos Metaheurísticos .....	23
1.1.4.3 Métodos CFD .....	27
1.2 DESCRIPCIÓN DEL PROBLEMA.....	30
1.3 JUSTIFICACIÓN.....	30
1.3.1 Aportes técnicos-científicos .....	30
1.3.2 Impacto social.....	31
1.4 HIPÓTESIS Y OBJETIVOS .....	31
1.4.1 Hipótesis .....	31
1.4.2 Objetivo general .....	31
1.4.3 Objetivos específicos.....	31
1.5 PLANTEAMIENTO GENERAL .....	32
<b>II. FUNDAMENTACIÓN TEÓRICA .....</b>	<b>34</b>

2.1	PERFIL AERODINÁMICO Y CONCEPTOS GENERALES DE AERODINÁMICA .	34
2.1.1	Geometría del perfil aerodinámico.....	34
2.1.2	Comportamiento del perfil aerodinámico.....	35
2.1.3	Gráficas características de un perfil aerodinámico.....	38
2.1.3.1	$CL$ vs. $\alpha$ .....	38
2.1.3.2	$CD$ vs. $CL$ .....	39
2.1.3.3	$CLCD$ vs. $\alpha$ .....	40
2.1.4	Regímenes de flujo de un perfil aerodinámico.....	41
2.1.5	Teoría de placa plana.....	43
2.2	FUNCIÓN DE TRANSFORMACIÓN DE CLASE-FORMA .....	44
2.2.1	Función de clase .....	45
2.2.2	Función de forma.....	46
2.2.3	Representación en componentes de la función de forma .....	48
2.3	OPTIMIZACIÓN Y ALGORITMO PSO.....	50
2.3.1	Optimización .....	50
2.3.2	Algoritmo PSO .....	51
2.4	DINÁMICA DE FLUIDOS COMPUTACIONAL.....	56
2.4.1	Teoría de Volúmenes de Control.....	56
2.4.1.1	Teorema del Transporte de Reynolds.....	57
2.4.2	Leyes de conservación de la mecánica de fluidos .....	60
2.4.2.1	Ley de conservación de la masa .....	60
2.4.2.2	Ley de conservación de la cantidad de movimiento.....	60
2.4.3	Ecuaciones de movimiento de la dinámica de fluidos computacional	61
2.4.4	Ecuaciones de movimiento en ANSYS Fluent.....	62
2.4.4.1	Ecuación de conservación de la masa .....	62
2.4.4.2	Ecuación de conservación de la cantidad de movimiento .....	63
2.4.5	Modelado de turbulencia en ANSYS Fluent.....	64
2.4.5.1	Modelo de turbulencia SST k- $\omega$ .....	65
2.5	SOFTWARES EMPLEADOS EN EL MARCO DE TRABAJO .....	66
2.5.1	MATLAB .....	66

2.5.2	ANSYS Workbench .....	66
2.5.2.1	SpaceClaim.....	68
2.5.2.2	Meshing .....	68
2.5.2.3	Fluent.....	69
<b>III.</b>	<b>METODOLOGÍA.....</b>	<b>70</b>
3.1	PERFIL AERODINÁMICO S809.....	71
3.1.1	Características .....	71
3.1.2	Condiciones de trabajo en régimen de máxima sustentación .....	72
3.2	PROBLEMA DE OPTIMIZACIÓN .....	73
3.2.1	Definición de las variables de diseño con el método CST .....	73
3.2.2	Límites geométricos de las variables de diseño .....	74
3.2.3	Definición del problema de optimización .....	81
3.3	ANÁLISIS CFD DEL PERFIL S809 .....	82
3.3.1	Simulación CFD del perfil S809 .....	82
3.3.1.1	Geometría del modelo computacional.....	83
3.3.1.2	Discretización del modelo computacional .....	86
3.3.1.3	Análisis numérico del modelo computacional .....	90
3.3.2	Automatización del análisis numérico.....	92
3.4	ALGORITMO PSO .....	94
3.4.1	Parámetros de control y población inicial .....	94
3.5	IMPLEMENTACIÓN DEL MARCO DE TRABAJO POA .....	95
3.5.1	Esquema del proyecto en ANSYS Workbench.....	95
3.5.2	Acoplamiento del optimizador-analizador con MATLAB y ANSYS Workbench	97
3.6	VALIDACIÓN DEL PERFIL ÓPTIMO MEDIANTE ANÁLISIS CFD .....	99
<b>IV.</b>	<b>RESULTADOS.....</b>	<b>102</b>
4.1	CURVA DE SUSTENTACIÓN DEL PERFIL S809 CON ANÁLISIS CFD .....	102
4.2	OPTIMIZACIÓN DEL PERFIL AERODINÁMICO.....	103
4.3	VALIDACIÓN DEL PERFIL ÓPTIMO MEDIANTE CURVAS DE SUSTENTACIÓN 109	
4.4	ANÁLISIS DE PARÁMETROS Y PROBLEMAS DE CONVERGENCIA .....	110



4.4.1	Redefinición de parámetros del modelo de turbulencia CFD .....	111
4.4.2	Resolución del diseño del experimento .....	111
4.4.2.1	Curva de sustentación del perfil S809 .....	112
4.4.2.2	Optimización del perfil aerodinámico .....	112
4.4.2.3	Validación del perfil óptimo mediante curvas de sustentación	120
4.4.3	Contornos de velocidad y presión de los perfiles aerodinámicos....	121
<b>V.</b>	<b>CONCLUSIONES Y PROSPECTIVAS .....</b>	<b>127</b>
<b>VI.</b>	<b>ANEXOS .....</b>	<b>129</b>
6.1	ANEXO 1. PSEUDOCÓDIGO DEL ALGORITMO PSO.....	129
6.2	ANEXO 2. COORDENADAS DEL PERFIL AERODINÁMICO S809. ....	130
6.3	ANEXO 3. <i>SCRIPT</i> EN PYTHON PARA LA GEOMETRÍA DEL MODELO. ...	132
6.4	ANEXO 4. <i>SCRIPT</i> EN PYTHON PARA LA DISCRETIZACIÓN DEL MODELO. 136	
6.5	ANEXO 5. <i>JOURNAL FILE</i> PARA EL ANÁLISIS CFD DEL PERFIL S809 EN ANSYS WORKBENCH. ....	141
6.6	ANEXO 6. <i>SCRIPT</i> EN MATLAB PARA EL ANÁLISIS NUMÉRICO AUTOMATIZADO DEL PERFIL S809. ....	143
6.7	ANEXO 7. <i>JOURNAL FILE</i> PARA EL ANÁLISIS CFD DEL PROCESO DE OPTIMIZACIÓN EN ANSYS WORKBENCH. ....	144
6.8	ANEXO 8. <i>SCRIPT</i> EN MATLAB PARA EL ACOPLAMIENTO DEL ANALIZADOR-OPTIMIZADOR. ....	146
<b>VII.</b>	<b>REFERENCIAS .....</b>	<b>151</b>

## INDICE DE TABLAS

Tabla 1. Clasificación de las HASWT (Karthikeyan et al., 2015). .....	15
Tabla 2. Mejoras de rendimiento aerodinámico de HASWT (Karthikeyan et al., 2015). ....	15
Tabla 3. Argumentos de la línea de comando para ejecutar ANSYS Workbench (ANSYS Inc., 2023a). .....	67
Tabla 4. Valores de los coeficientes y parámetros del perfil parametrizado NACA 0012...	75
Tabla 5. Valores de los coeficientes y parámetros del perfil parametrizado S809.....	80
Tabla 6. Límites de las variables de diseño. ....	81
Tabla 7. Características principales de los controles de malla en el modelo.....	89
Tabla 8. Estadísticas de la malla del modelo.....	90
Tabla 9. Parámetros de control para la inicialización del algoritmo PSO.....	94

## INDICE DE FIGURAS

Figura 1. Subsistemas y componentes en una turbina eólica (González et al., 2009).....	5
Figura 2. Área del rotor $A$ y velocidad nominal del viento $u$ (Zezatti & Castro, 2019). .....	6
Figura 3. Estructura y componentes de: a) HAWT y b) VAWT tipo Darrieus (Tawfiq et al., 2019).....	7
Figura 4. Disco actuador de una turbina. $U$ es la velocidad media del aire; los subíndices 1, 2, 3 y 4 indican las posiciones del fluido (Manwell et al., 2010). .....	11
Figura 5. Gráfica de coeficientes de la turbina de Betz (Manwell et al., 2010). .....	12
Figura 6. Forma típica y regiones de la pala de una turbina eólica (Schubel & Crossley, 2012).....	13
Figura 7. Efectos de $\Omega$ sobre un perfil aerodinámico actuando en la región media la pala (Burton et al., 2011).....	14
Figura 8. Distribución de presión sobre un perfil, a) $AoA = 0$ , b) $AoA \neq 0$ . .....	18
Figura 9. Diagrama de flujo del proceso de optimización (Abdelwahed & Abd El-Rahman, 2020).....	22
Figura 10. Clasificación de los métodos de optimización (Ramírez-Ochoa et al., 2022). ...	23
Figura 11. Clasificación de los algoritmos metaheurísticos (Ramírez-Ochoa et al., 2022). ..	24
Figura 12. Diagrama del planteamiento general de la metodología. ....	33
Figura 13. Parámetros geométricos de un perfil aerodinámico (Sadraey, 2013). .....	34
Figura 14. Cambio de flujo de velocidad sobre un perfil. a) $AoA = 0$ , b) $AoA \neq 0$ . .....	35
Figura 15. Movimiento de la fuerza resultante. a) fuerza en el $cp$ , b) adición de fuerzas equivalentes, c) fuerza en el $ac$ (Sadraey, 2013).....	36
Figura 16. Fuerzas de sustentación, arrastre y momento de cabeceo en el $ac$ (Sadraey, 2013).....	36
Figura 17. Variación típica del $CL$ vs. $\alpha$ (Sadraey, 2013).....	38
Figura 18. Variación típica del $CD$ vs. $CL$ (Sadraey, 2013). .....	39
Figura 19. Variación típica del $CD$ vs. $CL$ para un perfil con flujo laminar (Sadraey, 2013). .....	40
Figura 20. Variación típica del $CLCD$ vs. $\alpha$ (Sadraey, 2013).....	41
Figura 21. $CL$ y $CD$ del perfil S809, $Re = 7.5 \times 10^5$ (Manwell et al., 2010).....	42
Figura 22. Pérdida de sustentación del perfil por turbulencia en el ángulo $\alpha_s$ . .....	42

Figura 23. Variaciones del $CL$ , $CD$ y $CLCD$ para la teoría de la placa plana (J. Tangler & Kocurek, 2005). .....	43
Figura 24. Variables de diseño claves de un perfil aerodinámico (Kulfan & Bussoletti, 2006). .....	45
Figura 25. Geometrías de clases fundamentales.....	46
Figura 26. Componentes de la función de forma unitaria con $BPnx$ de grado 3. ....	49
Figura 27. Componentes de la función de forma unitaria con $BPnx$ de grado 5. ....	49
Figura 28. Componentes de la función de forma unitaria con $BPnx$ de grado 7. ....	50
Figura 29. Definición gráfica de la función $f(x)$ (Simon, 2013).....	51
Figura 30. Movimiento de una partícula en PSO (Ramírez-Ochoa et al., 2022).....	53
Figura 31. Diagrama de flujo del algoritmo PSO.....	56
Figura 32. Volumen de control con sus variables de campo de presión y velocidad (Cengel & Cimbala, 2006). .....	57
Figura 33. Geometría sencilla de un sistema de fluido (Cengel & Cimbala, 2006).....	58
Figura 34. VC infinitesimal fijo en coordenadas cartesianas mostrando los flujos máscicos $ment$ y $msal$ en las caras perpendiculares al eje $x$ (White, 2004). .....	61
Figura 35. Diagrama de flujo de la metodología. ....	70
Figura 36. Geometría del perfil aerodinámico S809. ....	71
Figura 37. Comparación $CL$ vs. $\alpha$ de los experimentos OSU, CSU y DUT. ....	73
Figura 38. Comparación del perfil NACA0012 con el perfil parametrizado. ....	75
Figura 39. Comparación del NACA0012 con el perfil parametrizado variando $Au1$ y $Al1$ . .....	76
Figura 40. Comparación del NACA0012 con el perfil parametrizado variando $Au2$ y $Al2$ . .....	76
Figura 41. Comparación del NACA0012 con el perfil parametrizado variando $Au3$ y $Al3$ . .....	77
Figura 42. Comparación del NACA0012 con el perfil parametrizado variando $Au4$ y $Al4$ . .....	77
Figura 43. Comparación del NACA0012 con el perfil parametrizado variando $Au5$ y $Al5$ . .....	78

Figura 44. Comparación del NACA0012 con el perfil parametrizado variando $Au_6$ y $A1_6$ . .....	78
Figura 45. Comparación del NACA0012 con el perfil parametrizado variando $Au_7$ y $A1_7$ . .....	79
Figura 46. Comparación del NACA0012 con el perfil parametrizado variando $Au_8$ y $A1_8$ . .....	79
Figura 47. Comparación del perfil S809 con el perfil parametrizado. ....	80
Figura 48. Esquema del proyecto en ANSYS Workbench.....	83
Figura 49. Geometría del modelo computacional para el análisis (Rodríguez & Celis, 2022). .....	84
Figura 50. Superficie del perfil S809 desde sus coordenadas. ....	85
Figura 51. Modelo 2D y el <i>script editor</i> de <i>SpaceClaim</i> .....	85
Figura 52. Detalle de la superficie del perfil S809 en el modelo. ....	86
Figura 53. Malla estructurada del modelo computacional. ....	86
Figura 54. Detalle de la malla próxima a la superficie del perfil. ....	87
Figura 55. Control <i>rear-length</i> sobre la malla.....	87
Figura 56. Control <i>domain-height</i> cercano al borde de salida.....	88
Figura 57. Elementos de malla en la capa límite de la superficie del perfil.....	88
Figura 58. Control <i>airfoil-bottom</i> en la superficie inferior del perfil. ....	89
Figura 59. Control <i>leading-edge</i> aplicado al borde de ataque.....	89
Figura 60. Configuración del Fluent Launcher. ....	91
Figura 61. Diagrama del análisis CFD automatizado del perfil S809.....	93
Figura 62. Geometrías de las partículas de la población inicial del enjambre. ....	95
Figura 63. Esquema del proyecto de la optimización del perfil. ....	96
Figura 64. Parámetros de entrada y salida del análisis numérico de la optimización. ....	97
Figura 65. Diagrama del marco de trabajo POA para la optimización del perfil. ....	98
Figura 66. Detalle de la superficie del perfil óptimo en el modelo. ....	100
Figura 67. Detalle de la malla próxima a la superficie del perfil óptimo. ....	100
Figura 68. Gráfica de la comparación de la curva de sustentación del perfil S809 con datos experimentales.....	102
Figura 69. Geometrías de las partículas en la iteración $k = 1$ . ....	103

Figura 70. Geometrías de las partículas en la iteración $k = 2$ .	104
Figura 71. Geometrías de las partículas en la iteración $k = 3$ .	104
Figura 72. Geometrías de las partículas en la iteración $k = 4$ .	105
Figura 73. Geometrías de las partículas en la iteración $k = 5$ .	105
Figura 74. Geometrías de las partículas en la iteración $k = 6$ .	106
Figura 75. Geometrías de las partículas en la iteración $k = 7$ .	106
Figura 76. Geometrías de las partículas en la iteración $k = 8$ .	107
Figura 77. Geometrías de las partículas en la iteración $k = 9$ .	107
Figura 78. Geometrías de las partículas en la iteración $k = 10$ .	108
Figura 79. Comparación de la geometría del perfil óptimo con el perfil S809.	108
Figura 80. Gráfica del VFP a lo largo de las iteraciones del algoritmo PSO.	109
Figura 81. Comparación de las curvas de sustentación del perfil óptimo con el perfil S809. .....	110
Figura 82. Gráfica de la comparación de la curva de sustentación del perfil S809 con datos experimentales. ....	112
Figura 83. Geometrías de las partículas de la población inicial del enjambre.	113
Figura 84. Geometrías de las partículas en la iteración $k = 1$ .	113
Figura 85. Geometrías de las partículas en la iteración $k = 2$ .	114
Figura 86. Geometrías de las partículas en la iteración $k = 3$ .	114
Figura 87. Geometrías de las partículas en la iteración $k = 4$ .	115
Figura 88. Geometrías de las partículas en la iteración $k = 5$ .	115
Figura 89. Geometrías de las partículas en la iteración $k = 6$ .	116
Figura 90. Geometrías de las partículas en la iteración $k = 7$ .	116
Figura 91. Geometrías de las partículas en la iteración $k = 8$ .	117
Figura 92. Geometrías de las partículas en la iteración $k = 9$ .	117
Figura 93. Geometrías de las partículas en la iteración $k = 10$ .	118
Figura 94. Comparación de la geometría del perfil óptimo con el perfil S809.	118
Figura 95. Gráfica del VFP a lo largo de las iteraciones del algoritmo PSO.	119
Figura 96. VFP promedio del algoritmo PSO.	120
Figura 97. Comparación de las curvas de sustentación del perfil óptimo con el perfil S809. .....	121

Figura 98. Contornos de velocidad del perfil S809 a un $\alpha = 10^\circ$ .....	122
Figura 99. Contornos de presión del perfil S809 a un $\alpha = 10^\circ$ . ....	122
Figura 100. Contornos de velocidad del perfil óptimo a un $\alpha = 10^\circ$ . ....	123
Figura 101. Contornos de presión del perfil óptimo a un $\alpha = 10^\circ$ . ....	123
Figura 102. Contornos de velocidad del perfil óptimo a un $\alpha = 13^\circ$ . ....	124
Figura 103. Contornos de presión del perfil óptimo a un $\alpha = 13^\circ$ . ....	124
Figura 104. Contornos de velocidad del perfil S809 a un $\alpha = 18^\circ$ .....	125
Figura 105. Contornos de presión del perfil S809 a un $\alpha = 18^\circ$ . ....	125
Figura 106. Contornos de velocidad del perfil óptimo a un $\alpha = 18^\circ$ . ....	125
Figura 107. Contornos de presión del perfil óptimo a un $\alpha = 18^\circ$ . ....	126

## **ABREVIATURAS Y SIGLAS**

AEP:	Anual Energy Production
AMDEE:	Asociación Mexicana de Energía Eólica
AM-PSO:	Adaptative Mutation-Particle Swarm Optimization
AoA:	Angle of Attack
BEM:	Blade Element Momentum
BézierGAN:	Bézier Generative Adversarial Network
BiMADS:	Biobjective Mesh Adaptive Direct Search
BSplineGAN:	BSpline Generative Adversarial Network
CFD:	Computational Fluid Dynamics
CMA-ES:	Covariance Matrix Adaptation-Evolution Strategy
COE:	Cost Of Energy
CST:	Class-Shape Transformation
CSU:	Colorado State University
DE:	Differential Evolution
DES:	Detached Eddy Simulation
DUT:	Delf University of Technology
EOA:	Evolutionary Optimization Algorithms
GA:	Genetic Algorithms
GPU:	Graphics Processing Unit
GUI:	Graphical User Interface
HAWT:	Horizontal Axis Wind Turbines
HASWT:	Horizontal Axis Small Wind Turbines
HPC:	High Performance Computing
IEC:	International Electrotechnical Commission
IPCC:	Intergovernmental Panel on Climate Change
LE:	Leading Edge
LES:	Large Eddy Simulation
MATLAB:	Matrix Laboratory
MOPSO:	Multi-Objective Particle Swarm Optimization
NREL:	National Renewable Energy Laboratory



OF:	Objective Function
OOP:	Object-Oriented Programming
OSU:	Ohio State University
PARSEC:	Parametric Sections
POA:	Parametrización-Optimización-Analizador
PSO:	Particle Swarm Optimization
RANS:	Reynolds Averaged Navier-Stokes
RBF:	Radial-Base Function
RSM:	Response Surface Methodology
SC:	Superficie de Control
SERI:	Solar Energy Research Institute
SST:	Shear-Stress Transport
SVD:	Singular-Value Decomposition
TE:	Trailing Edge
TTR:	Teorema de Transporte de Reynolds
TVR:	Turbulent Viscosity Ratio
VAWT:	Vertical Axis Wind Turbines
VC:	Volumen de Control
WT:	Wind Turbines

## RESUMEN

En este trabajo se desarrolló una metodología para optimizar un perfil aerodinámico mediante un marco de trabajo que incluye un método de parametrización, un algoritmo de optimización y una herramienta de análisis numérico, con la finalidad de maximizar el coeficiente de sustentación del perfil en su régimen de máxima sustentación, con el antecedente de que obtener un perfil óptimo para esta condición de trabajo permitirá diseñar una turbina eólica de eje horizontal de baja potencia óptima con respecto a su potencia de salida; la metodología incluyó la definición de las características y condiciones de trabajo del perfil de referencia S809 mediante datos experimentales, la definición de la función objetivo y las variables de diseño del perfil mediante el método de parametrización, la implementación del análisis numérico del perfil de referencia, el desarrollo del algoritmo de optimización y su acoplamiento con la herramienta de análisis, y la validación del perfil óptimo mediante la comparación de su curva de sustentación con la curva de sustentación del perfil de referencia. Los resultados indicaron una mejora del coeficiente de sustentación del perfil óptimo de 3.736 % respecto al perfil S809 en el ángulo de ataque  $\alpha = 10^\circ$ , mientras que el  $C_{L,max}$  del perfil óptimo mejoró un 9.8 % respecto al  $C_{L,max}$  del perfil S809, el cual se obtuvo a un ángulo de pérdida de  $\alpha = 13^\circ$  con un coeficiente de sustentación máximo de  $C_{L,max} = 1.019$ . Finalmente, se encontró que para mejorar la eficiencia de convergencia del modelo de turbulencia CFD, fue necesario modificar un parámetro para el cálculo de la viscosidad turbulenta.

**Palabras clave:** optimización, perfil aerodinámico, coeficiente de sustentación, algoritmo PSO, análisis CFD.

## ABSTRACT

In this work, a methodology was developed to optimize an airfoil by means of a framework that includes a parameterization method, an optimization algorithm and a numerical analysis tool, with the purpose of maximizing the lift coefficient of the airfoil in its maximum lift regime, with the background that obtaining an optimum airfoil for this working condition will allow the design of a low power horizontal axis wind turbine with optimum power output; the methodology included the definition of the characteristics and working conditions of the S809 reference airfoil by means of experimental data, the definition of the objective function and the airfoil design variables by means of the parameterization method, the implementation of the numerical analysis of the reference airfoil, the development of the optimization algorithm and its coupling with the analysis tool, and the validation of the optimum airfoil by comparing its lift curve with the lift curve of the reference airfoil. The results indicated an improvement of the lift coefficient of the optimum airfoil of 3.736 % with respect to the S809 airfoil at the angle of attack  $\alpha = 10^\circ$ , while the  $C_{L,max}$  of the optimum airfoil improved by 9.8 % with respect to the  $C_{L,max}$  of the S809 airfoil, which was obtained at a stall angle of  $\alpha = 13^\circ$  with a maximum lift coefficient of  $C_{L,max} = 1.019$ . Finally, it was found that to improve the convergence efficiency of the CFD turbulence model, it was necessary to modify a parameter for the calculation of the turbulent viscosity.

**Keywords:** optimization, airfoil, lift coefficient, PSO algorithm, CFD analysis.

## I. INTRODUCCIÓN

En un intento por reducir las emisiones de gases de efecto invernadero y combatir el incremento de la temperatura media anual del planeta, las naciones del mundo se han inclinado por el uso de las energías renovables con el objetivo de disponer de energía confiable, barata y limpia. Una de las fuentes de energía renovable más prometedoras es la energía eólica, cuyo crecimiento ha sido exponencial en las últimas décadas. Para el año 2020, la capacidad de energía eólica instalada a nivel global incrementó más de un 90% con respecto a la instalada en el 2019, siendo la región de Norteamérica la de mayor capacidad nueva instalada. Mientras que para el 2021, Norteamérica se convirtió en la tercera región con mayor capacidad nueva instalada después de las regiones del Pacífico Asiático y Europa (Costoya et al., 2023).

En lo que respecta a México, entre el 2016 y 2018, el desarrollo de la energía eólica creció aceleradamente debido a la certidumbre jurídica y los bajos costos de estas tecnologías. Sin embargo, a partir del 2019 se registró una disminución debido a factores como: 1) la falta de voluntad y política pública a favor de la transición energética, 2) la falta de inversión en la modernización de la red eléctrica, 3) la parálisis administrativa para inversionistas privados por las autoridades del sector, y 4) los constantes y profundos cambios regulatorios en la ley energética. Consecuentemente, para el 2022 la Asociación Mexicana de Energía Eólica (AMDEE) registró una potencia instalada de 7,317 MW provenientes de parques eólicos, mientras que el total de electricidad generada por energía eólica superó los 20,000 GWh equivalente al 6.09% de la generación total de electricidad en México, siendo un total de 70 parques eólicos distribuidos en 15 estados de la república (AMDEE, 2023).

Cabe mencionar que, en el estado de Querétaro, a finales del 2023 entró en operación el parque eólico de San Pedro, Huimilpan, el cual actualmente se encuentra generando 30 MW mediante 15 turbinas eólicas, además de que se considera un crecimiento a futuro de generar hasta 100 MW, según informes del director de la Agencia Estatal de Energía del Estado de Querétaro, Mauricio Reyes Caracheo (Rosas, 2024).

A diferencia de las turbinas eólicas de grandes potencias, las cuales son instaladas en sitios con condiciones óptimas del recurso del viento, las turbinas eólicas de baja potencia son empleadas para producir energía con condiciones limitadas del viento. Debido

a esto, optimizar los parámetros asociados a la geometría de las palas y del perfil aerodinámico de estas es una cuestión importante de diseño, ya que esto permitiría obtener turbinas eólicas con rotores y palas más pequeñas, pero con la capacidad de producir más energía con limitación del recurso eólico (Karthikeyan et al., 2015).

Este trabajo consiste en optimizar el perfil aerodinámico de una turbina eólica de eje horizontal de baja potencia mediante la maximización de su coeficiente de sustentación definido en el régimen de máxima sustentación. La finalidad de este estudio es obtener un perfil aerodinámico óptimo en comparación con el perfil de referencia S809, lo cual permitiría diseñar una turbina eólica de eje horizontal de baja potencia con la capacidad de generar mayor potencia. Se utilizó una metodología que engloba el método de la función de transformación de clase y forma como herramienta de parametrización geométrica, un algoritmo de optimización y un modelo de turbulencia mediante dinámica de fluidos computacional para analizar el comportamiento aerodinámico del perfil.

## **1.1 ANTECEDENTES**

### **1.1.1 Clasificación y características de las turbinas eólicas**

La industria de las energías renovables ha ido en aumento a nivel global desde finales del siglo pasado debido al continuo agotamiento de las fuentes de energía fósiles. Consecuentemente, uno de los segmentos que mayor auge ha tenido en esta industria tanto en investigación como en desarrollo es la energía eólica. Las tecnologías que aprovechan este tipo de energía son las turbinas eólicas (*Wind Turbines*, WT) ya que convierten la energía cinética del viento en energía eléctrica a través de su subsistema de captación y su subsistema mecánico (González et al., 2009), como se observa en la Figura 1.

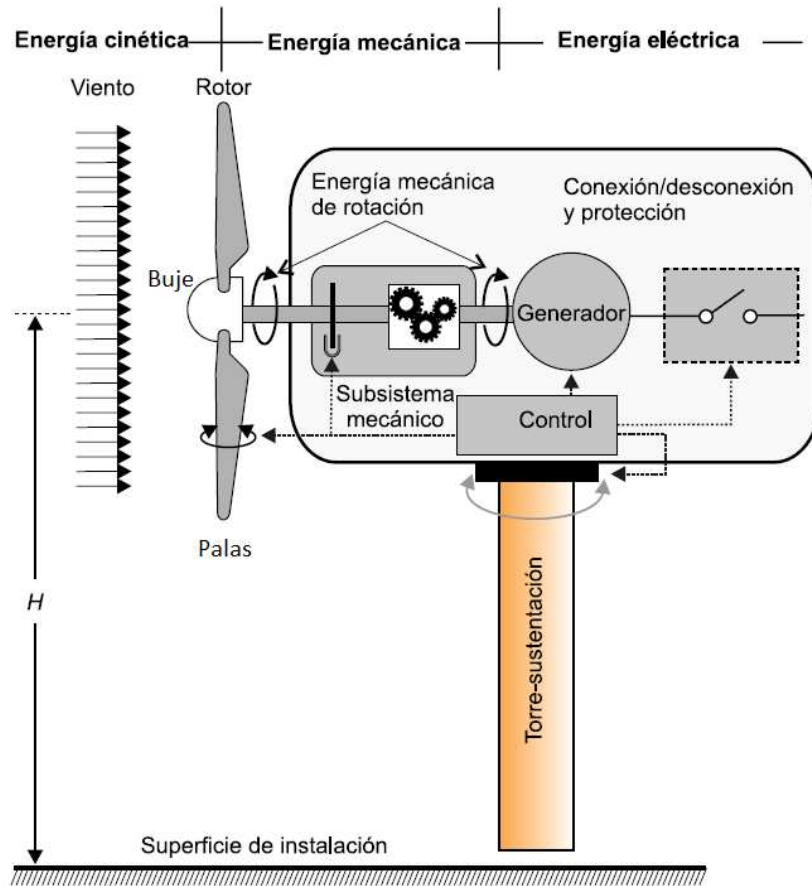


Figura 1. Subsistemas y componentes en una turbina eólica (González et al., 2009).

El subsistema de captación está integrado por el rotor de la turbina que a su vez está compuesto por las palas y el buje. Estos componentes generan el área de barrido  $A$  del rotor, véase la Figura 2, la cual es el área circular formada por el diámetro que generan las palas al entrar en movimiento y es un factor clave para determinar la cantidad de energía que puede capturar la turbina.

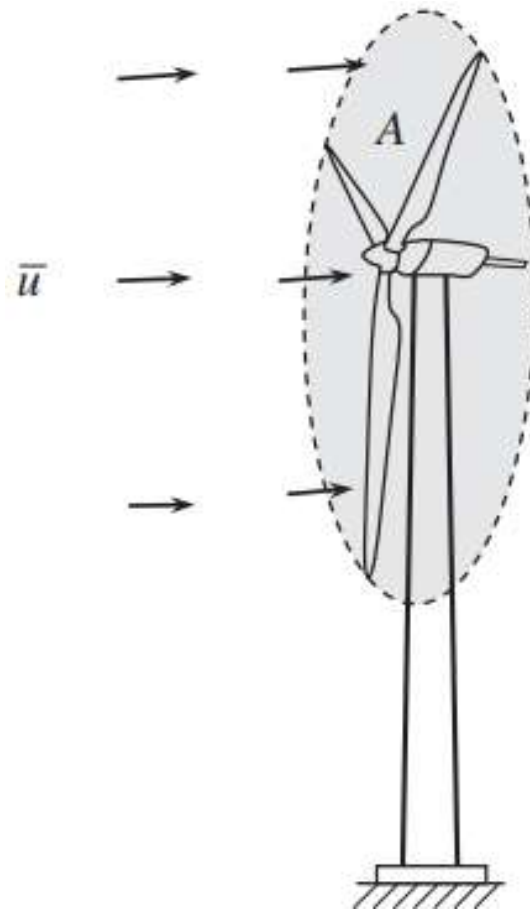


Figura 2. Área del rotor  $A$  y velocidad nominal del viento  $\bar{u}$  (Zezatti & Castro, 2019).

Las dos clasificaciones de turbinas eólicas más comunes a lo largo de la historia son las turbinas eólicas de eje horizontal (*Horizontal Axis Wind Turbines*, HAWT) y las turbinas eólicas de eje vertical (*Vertical Axis Wind Turbines*, VAWT), en la Figura 3 se observan la estructura y componentes generales de cada una. Las HAWT han sido las más utilizadas para la producción de energía a gran escala debido a su gran capacidad de potencia y mayor balance entre beneficios y limitaciones en comparación con las VAWT, de tal manera que estas últimas prácticamente han desaparecido del mercado actual y son empleadas mayormente en aplicaciones residenciales y de baja escala (González et al., 2009).

Actualmente, ambas tecnologías son investigadas con el fin de incrementar el mercado de las energías renovables bajo la necesidad global de disminuir el uso de fuentes de energías fósiles, satisfacer la demanda energética y obtener costos de energía competitivos, por lo que es esencial que diseñadores y fabricantes busquen soluciones

de diseño óptimas para mejorar el rendimiento de las WT. Cabe mencionar que el incremento del mercado eólico es esencial para cumplir con estatutos internacionales en materia de transición energética y desarrollo sustentable, como lo son la Agenda 2030, el Acuerdo de París, y el Panel Intergubernamental sobre el Cambio Climático (*Intergovernmental Panel on Climate Change, IPCC*) (Moallemi et al., 2022).

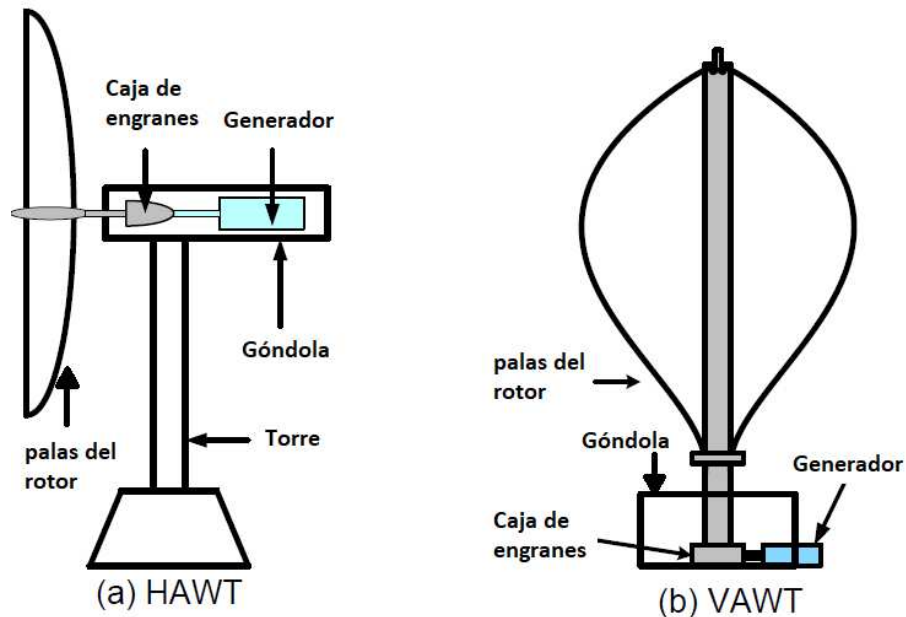


Figura 3. Estructura y componentes de: a) HAWT y b) VAWT tipo Darrieus (Tawfiq et al., 2019).

Sin embargo, el desarrollo e instalación de HAWT de grandes potencias ha disminuido en los últimos años mientras que el mercado se ha direccionado a incrementar la investigación y desarrollo de turbinas eólicas de baja potencia, o también conocidas como pequeñas turbinas eólicas de eje horizontal (*Horizontal Axis Small Wind Turbines, HASWT*), debido a la crisis financiera y a las ventajas que esta tecnología ofrece (Scappatici et al., 2016). Las HASWT se emplean en la generación de energía en hogares, industrias, comercios, granjas y pueblos aislados, así como en sistemas de generación de energía híbridos.

De acuerdo con la Comisión Electrotécnica Internacional (*International Electrotechnical Commission, IEC*), las HASWT se caracterizan por tener un área de barrido del rotor de  $A \leq 200 \text{ m}^2$  y potencias nominales usualmente de  $P \leq 30 \text{ kW}$



(International Electrotechnical Commission, 2006). Es importante mencionar que el mercado de las HASWT es una de las actuales apuestas de la industria de las energías renovables para conseguir redes de distribución de electricidad viables como una alternativa a las redes eléctricas descentralizadas. Por otro lado, a pesar del impulso que ha tenido este mercado, estas tecnologías aún presentan retos con relación a costos, rendimiento técnico, entorno construido, barreras y oportunidades de ubicación (Moreira Chagas et al., 2020).

### 1.1.2 Optimización de turbinas eólicas de eje horizontal

El diseño de turbinas eólicas se considera un problema de optimización ya que se busca maximizar uno o más objetivos considerando un conjunto de restricciones de diseño. El estudio y desarrollo de las turbinas eólicas modernas inició desde la década de 1980, por lo que diversos autores de la comunidad científica han propuesto diferentes objetivos, métodos de solución, restricciones, algoritmos, herramientas y modelos para satisfacer las necesidades de los distintos problemas de optimización de HAWT.

En optimización, la función objetivo (*Objective Function*, OF) es aquella que es evaluada para encontrar la solución óptima del problema y que depende de un conjunto de variables. La solución final deberá cumplir con un conjunto de restricciones de diseño, con la finalidad de obtener el máximo o mínimo valor (solución óptima) de la OF, además, los problemas de optimización comúnmente cuentan con límites de las variables de diseño que deben mantener la solución dentro de un cierto dominio de la función (Hussain et al., 2019a).

En un estudio (Chehouri et al., 2015) se identificó que existen tres objetos de optimización en el diseño de HAWT: 1) la optimización de la turbina eólica, 2) la optimización de la pala de la turbina y 3) la optimización del perfil aerodinámico, cada uno de ellos presenta diferentes OF, variables de diseño y restricciones, los cuales se verán en las siguientes subsecciones.

### 1.1.2.1 Optimización de la turbina eólica

Las funciones objetivo en este tipo de problemas de optimización al principio se enfocaron en la maximización del coeficiente de potencia ( $C_p$ ) de las turbinas, lo cual generó el desarrollo de rotores de grandes potencias, con palas de gran tamaño y de grandes espesores. Debido a que este tipo de optimización se realiza considerando una velocidad de viento nominal específica, posteriormente la tendencia de las OF se centró en la maximización de la producción anual de energía (*Annual Energy Production*, AEP), la cual involucra variables como el periodo de tiempo de uso de la WT (*e.g.* un año) y el espectro del flujo de viento a una velocidad específica. Finalmente, debido a la necesidad de que la energía eólica sea competitiva respecto a costos, la tendencia de las OF evolucionó a la minimización del costo de energía (*Cost Of Energy*, COE), que es la relación entre los costos totales de la turbina eólica y la AEP.

Uno de los primeros enfoques que utilizaron los diseñadores para minimizar el COE fue la restricción del peso de las palas del rotor y el incremento de su longitud, esto provocó que se utilizaran perfiles aerodinámicos con mejor rendimiento, lo cual afecta directamente la estabilidad de la estructura y el costo del rotor. De acuerdo con Chehourri et al. (2015), existen cuatro categorías de OF principales: 1) la minimización del COE, 2) la maximización de la AEP, 3) la minimización de la masa de las palas, y 4) las funciones multiobjetivo, en donde cada una de ellas está sujeta a diferentes restricciones de diseño, ya sean geométricas, aerodinámicas o físicas.

Un ejemplo es el estudio realizado por Verma et al. (2022) en donde se maximizó la AEP y se minimizó la masa de las palas de una turbina del Laboratorio Nacional de Energía Renovable (*National Renewable Energy Laboratory*, NREL) *Phase VI* mediante un algoritmo genético multiobjetivo, de tal manera que mejoraron significativamente la capacidad de producción de energía de la WT aumentando el  $C_p$  un 5.43% y reduciendo el peso de las palas en un 13%, obteniendo un rotor ligeramente menos pesado.

### 1.1.2.2 Optimización de la pala de la turbina

Las palas son uno de los componentes más importantes debido a que influyen directamente en la estabilidad estructural y el desempeño aerodinámico de las WT. Los primeros estudios de optimización de estos componentes se realizaron dejando a un lado la

influencia de la forma del perfil aerodinámico y considerando la geometría entera de la pala, es decir, se empleó una forma fija del perfil y se utilizaron variables de diseño geométricas y físicas del componente, (*e.g.* giro de la pala, fatiga del componente, etc.).

Jureczko et al. (2005) optimizaron la pala de una turbina puramente desde un enfoque estructural, sin embargo, consideraron que estos problemas de optimización deben ser vistos bajo múltiples criterios: 1) minimización de las vibraciones de la pala, 2) maximización de la potencia generada, 3) minimización de los costos de materiales, 4) estabilidad local y global de la estructura de la pala, y 5) requisitos de resistencia de la estructura de la pala, recalando que muchas veces estos criterios son contradictorios debido a que minimizar el peso del componente compromete la estabilidad de la turbina a su vez que favorece la reducción de costos, entre otras cosas.

Por otra parte, mejorar el rendimiento aerodinámico de las palas depende del estudio aeroelástico de las cargas generadas sobre estos componentes, lo cual se refiere a la interacción de la estructura de la WT con el flujo de viento; el modelo matemático más utilizado para el estudio aeroelástico es la teoría del momento del elemento de la pala (*Blade Element Momentum*, BEM). El modelo BEM considera las palas de la turbina como un disco actuador, en donde el flujo es considerado dentro de un volumen de control en 2D en forma de tubo alrededor del disco, por lo que con el desarrollo del modelo se calculan los coeficientes de sustentación y arrastre del perfil aerodinámico a lo largo de la distribución de la pala para obtener el torque y el empuje de la WT (Manwell et al., 2010). En la Figura 4 se observa el modelo del disco actuador de una turbina para un análisis unidimensional.

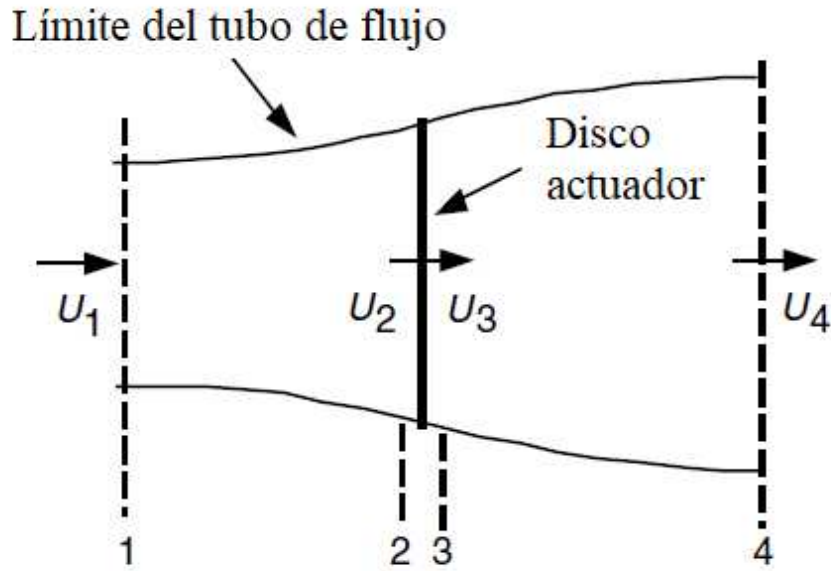


Figura 4. Disco actuador de una turbina.  $U$  es la velocidad media del aire; los subíndices 1, 2, 3 y 4 indican las posiciones del fluido (Manwell et al., 2010).

Del modelo BEM se deriva la teoría del límite de Betz, la cual establece que el coeficiente de potencia máximo ( $C_{P_{max}}$ ) teórico que puede tener el rotor de una turbina es de  $C_{P_{max}} = 16/27$ , siendo esta la máxima cantidad de energía que puede ser extraída del viento, y que además se ve afectada por efectos como la turbulencia del flujo detrás del rotor, pérdidas de punta de la pala y las fuerzas de arrastre de esta (Manwell et al., 2010). En la Figura 5 se observa una gráfica del  $C_P$  y el coeficiente de empuje ( $C_T$ ) de una turbina de Betz ideal en función del factor de inducción axial ( $a$ ).  $U$  y  $U_4$  son las velocidades del viento a la entrada y salida de la turbina, respectivamente.

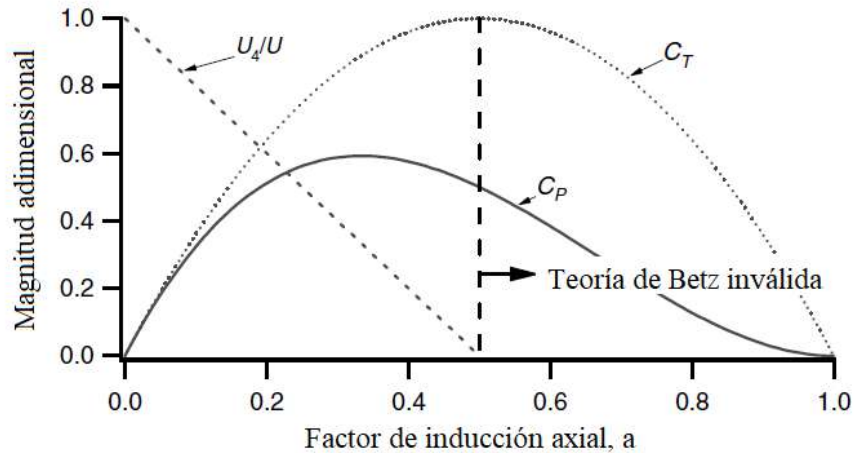


Figura 5. Gráfica de coeficientes de la turbina de Betz (Manwell et al., 2010).

A pesar de que el modelo BEM es simple, preciso, intuitivo y rápido, asume la ausencia de fuerzas radiales en el flujo y la generación de turbulencia debido a que estas se consideran dentro de un flujo en 3D (Lanzafame et al., 2015), por lo que distintos autores han propuesto factores de corrección para estas consideraciones y para el cálculo de los coeficientes de sustentación y arrastre; los estudios realizados con mejoras del modelo BEM han demostrado resultados favorecedores, como los reportados por Hamlaoui et al. (2018) y El Khchine & Sriti (2018), sin embargo, emplear estos factores de corrección normalmente implica incrementar la complejidad del modelo en términos del costo computacional requerido para su solución.

También es importante mencionar que se han desarrollado diferentes códigos o softwares cuyo solucionador se basa en la teoría BEM, tales como AeroDyn (Moriarty & Hansen, 2005), HARP\_Opt (Sale, 2010) y XFOIL (Drela, 1989), por mencionar algunos. Un ejemplo es el estudio de Johnson et al. (2016), donde se utilizó XFOIL para predecir los coeficientes de sustentación y arrastre de la pala de una turbina con características específicas como parte de una metodología extendida del modelo BEM, para comparar el comportamiento del rotor con mediciones experimentales controladas.

De igual manera, Chehouri et al. (2015) menciona que a medida que avanzó la computación, se realizaron los primeros estudios que emplearon la dinámica de fluidos computacional (*Computational Fluid Dynamics*, CFD) para el diseño del rotor de una turbina, los cuales se hicieron a partir de un análisis aeroelástico preliminar con el modelo

BEM, tal que el análisis de cargas del modelo CFD dependía de la eficiencia del modelo preliminar. Este autor menciona también que los métodos de cálculo CFD se dividen en: 1) *Direct Navier-Stokes* (DNS), 2) *Euler simulation*, 3) *Reynolds Averaged Navier-Stokes* (RANS), 4) *Large Eddy Simulation* (LES), y 5) *Detached Eddy Simulation* (DES). Los métodos de CFD serán abordados detalladamente en una sección posterior.

### 1.1.2.3 Optimización del perfil aerodinámico

El desarrollo de perfiles aerodinámicos es un área importante en las disciplinas aeroespacial y de generación de energía, ya que existen distintas aplicaciones en las que estos pueden ser empleados (*e.g.* aviones, turbinas de vapor, turbinas hidroeléctricas). En el caso de las turbinas eólicas, las necesidades de optimización de perfiles están en función de las características deseables, las cuales se dividen en: 1) estructurales y 2) aerodinámicas.

Por la parte estructural, el espesor máximo del perfil ( $T_{max}$ ) y la distribución adecuada de la cuerda del perfil son los parámetros más importantes para la región de raíz (*root*) de la pala ya que estos proveen fuerza y rigidez a esta, además de garantizar compatibilidad constructiva. Mientras que, por la parte aerodinámica, el coeficiente de sustentación  $C_L$ , así como la relación de los coeficientes de sustentación y arrastre  $C_L/C_D$  del perfil son los parámetros más importantes para la región de punta (*tip*) (Chehouri et al., 2015).

En la Figura 6 se muestra la forma típica de la pala de una WT y sus regiones, mientras que en la Figura 7 se muestra un perfil aerodinámico como sección transversal de la región media de la pala de una HAWT y cómo influye la velocidad angular ( $\Omega$ ) del flujo de viento sobre el perfil.

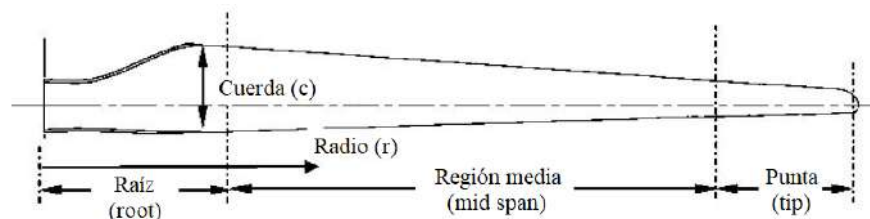


Figura 6. Forma típica y regiones de la pala de una turbina eólica (Schubel & Crossley, 2012).

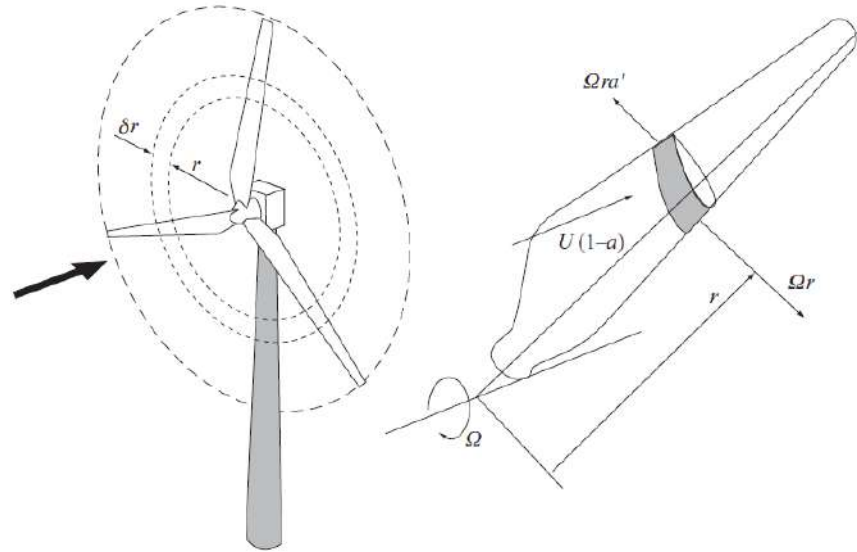


Figura 7. Efectos de  $\Omega$  sobre un perfil aerodinámico actuando en la región media la pala (Burton et al., 2011).

Los estudios consultados de optimización de perfiles aerodinámicos se abordarán con mayor detalle en las secciones siguientes debido a que se hará referencia a trabajos realizados para cada método de optimización.

### 1.1.3 Optimización de turbinas eólicas de baja potencia

A diferencia de las HAWT de gran escala que son diseñadas para sitios con un gran potencial del recurso eólico, las pequeñas turbinas eólicas son adecuadas para sitios con menor recurso del viento, además cuentan con características como un bajo costo, portabilidad, poca generación de ruido y ser libres de mantenimiento en estructura, por lo que esta tecnología es conveniente para zonas urbanas y rurales.

En un estudio (Karthikeyan et al., 2015) se sugiere que las HASWT se clasifican en tres conjuntos según su potencia de salida y el radio del rotor, como se observa en la Tabla 1, además menciona que su  $C_p$  se encuentra comúnmente alrededor de 0.25. Por otra parte, debido al tamaño de las palas, el inicio de funcionamiento de las HASWT es más complicado (principalmente en las micro) ya que esto depende del torque aerodinámico producido por la energía cinética del viento, siendo esta una de las causas que generan la necesidad de optimizarlas para que sean competitivas en el mercado energético, principalmente con respecto a su rendimiento aerodinámico.

Tabla 1. Clasificación de las HASWT (Karthikeyan et al., 2015).

Clasificación	Potencia de salida	Radio del rotor
Micro	$< 1 \text{ kW}$	1.5 m
Gama media ( <i>mid-range</i> )	$< 5 \text{ kW}$	2.5 m
Mini	$\geq 20 \text{ kW}$	5 m

Desde de la década de 1980, el NREL se dedicó a desarrollar familias de perfiles aerodinámicos especialmente para palas de WT, tarea que en los últimos años continuaron instituciones como la *Delf University of Technology* (DUT) (Timmer & Van Rooij, 2003). Entre los perfiles desarrollados se encuentran el BW-3, S822, S823 y el NACA2412, por mencionar algunos. De esto destaca que los perfiles con buen rendimiento aerodinámico en flujos con números de Reynolds ( $Re$ ) de  $Re \leq 5 \times 10^5$  son los más adecuados para HASWT (Karthikeyan et al., 2015).

Suresh & Rajakumar (2019) diseñaron una HASWT de 2 kW con un radio del rotor de 1.8 m.; emplearon el software QBlade para analizar aerodinámicamente una variedad de perfiles de distintas familias, con ello obtuvieron los coeficientes  $C_L$  y las relaciones  $C_L/C_D$  de cada uno, determinando que el perfil SD7080 es el más adecuado de acuerdo con su  $C_L/C_D$  y su comportamiento de pérdida suave para un rango del ángulo de ataque (*Angle of Attack*, AoA) o  $\alpha$  de  $5^\circ \leq \alpha \leq 9^\circ$ . Finalmente obtuvieron el diseño de la turbina y su curva de potencia característica empleando el modelo BEM.

Karthikeyan et al. (2015) mencionan que las investigaciones de optimización de turbinas de baja potencia se dividen en estudios de: 1) palas de la turbina y 2) perfil aerodinámico, en donde cada uno de ellos presenta diferentes parámetros de optimización y resultados. Este autor resume los estudios más significativos como se muestran en la Tabla 2, por lo que se sugiere consultar su trabajo para más detalles. En los párrafos siguientes se mencionan algunas investigaciones.

Tabla 2. Mejoras de rendimiento aerodinámico de HASWT (Karthikeyan et al., 2015).

Elemento	Parámetros	Modificaciones y resultados
Perfil aerodinámico	Borde de salida	Más delgado con superficie curvada: <ul style="list-style-type: none"> <li>• Mejor relación sustentación-arrastre</li> </ul>
		Espalda plana: <ul style="list-style-type: none"> <li>• Mejor resistencia estructural</li> <li>• Menor costo de material</li> </ul>



		<ul style="list-style-type: none"> <li>• Velocidades de arranque menores</li> </ul>
	Línea de peralte o <i>camber line</i>	Perfil delgado con menor longitud de cuerda: <ul style="list-style-type: none"> <li>• Mejor rendimiento</li> <li>• Menor tamaño y peso</li> <li>• Fácil control de ángulo de inclinación</li> </ul>
Pala	Cuerda, distribución de giro y ángulo de inclinación	Mejor comportamiento de arranque a un ángulo de 20°
	Paso doble del perfil	Pala sin torsión conectada por dos perfiles con ángulos de inclinación optimizados

Tang et al. (2015) desarrollaron un modelo matemático para optimizar la cuerda y la distribución de ángulo de giro de una HASWT de 10 kW. Emplearon como OF el  $C_p$  en función del factor de inducción axial y teniendo en cuenta las pérdidas de punta y de centro, así como los efectos de arrastre. Igualmente, emplearon XFOIL para calcular los coeficientes  $C_L$  y  $C_D$  ajustados a parámetros del viento y parámetros geométricos de la turbina, mientras que para la optimización utilizaron una función de minimización integrada en MATLAB.

Shen et al. (2016) optimizaron la pala de una HASWT de 600 W, para ello consideraron como variables de diseño la distribución de la cuerda y ángulo de giro a lo largo de esta, además de la forma del perfil aerodinámico a lo largo de la curvatura de la pala. Consiguieron la geometría de la pala como un número finito de perfiles aerodinámicos apilados, obtenidos mediante curvas de Bézier. Para la parte de la optimización, utilizaron el método de superficie de elevación como modelo matemático para describir el comportamiento de la pala y lo acoplaron a un micro algoritmo genético para maximizar la AEP, su capacidad de arranque inicial y el empuje del rotor. Los resultados mostraron un incremento en los primeros dos objetivos comparados con la optimización de palas de geometrías de dos dimensiones.

De acuerdo con Manwell et al. (2010), el comportamiento de un perfil aerodinámico se clasifica en tres regímenes de flujo: 1) régimen de flujo adjunto (*attached flow*), 2) régimen de mayor sustentación (*high lift/stall development*), y 3) régimen de estancamiento total (*flat plate/fully stalled*). El modelo BEM es útil para predecir el comportamiento aerodinámico de una turbina eólica en condiciones de operación, sin embargo, otros métodos deben ser empleados para predecir el comportamiento de una pala cuando el perfil trabaja en los regímenes de mayor sustentación y de estancamiento total, ya

que en estos empiezan a aparecer los efectos de turbulencia. Para ello, las ecuaciones propuestas por Viterna son útiles ya que modifican el modelo BEM para estimar el comportamiento del perfil para un rango de ángulos de ataque mayor (J. Tangler & Kocurek, 2005).

Consecuentemente, en un estudio (Hassanzadeh et al., 2016) se optimizó la turbina *UAE phase VI* para maximizar la AEP. Se emplearon la distribución de la cuerda y el ángulo de giro a lo largo de la pala como variables de diseño, mientras que para la predicción del comportamiento aerodinámico se utilizaron las ecuaciones de Viterna acopladas a un algoritmo genético. Los resultados de este estudio se compararon con otros que emplearon diferentes modificaciones del modelo BEM, por lo que obtuvieron una ganancia del 8.51% de la AEP y una turbina con mejor rendimiento entre velocidades de arranque y de parada.

Por otra parte, Ram et al. (2018) optimizaron la pala de una HASWT de 20 kW empleando perfiles aerodinámicos adecuados para flujos con  $Re \leq 5 \times 10^5$ , con la finalidad de diseñar una turbina que desprece los efectos de rugosidad en el borde de ataque de la pala y las limitaciones del recurso eólico. Obtuvieron los perfiles para las regiones de raíz, media y punta de la pala, empleando curvas de Bézier y un algoritmo genético, además usaron XFOIL para predecir los valores de  $C_L$  y  $C_D$ , por lo que una vez teniendo la forma de la pala emplearon el código HARP\_Opt para maximizar la AEP en un 2.3%, con una velocidad de arranque de la turbina de 2 m/s .

Finalmente, Muhsen et al. (2020) optimizaron la pala de una HASWT mediante un proceso iterativo y el modelo BEM, para ello seleccionaron los parámetros de diseño iniciales de la turbina (*e.g.* número de palas, radio del rotor, velocidades de operación y propiedades del fluido) y seleccionaron los perfiles S1210 y S1223 para la región de raíz y punta de la pala, respectivamente. Posteriormente utilizaron XFOIL para predecir los coeficientes  $C_L$  y  $C_D$  así como los mejores ángulos  $\alpha$ . El proceso iterativo se realizó en MATLAB, mientras que el cálculo del  $C_p$  de la turbina se hizo en QBlade y se obtuvieron los valores óptimos de la cuerda y  $\alpha$  para cada sección de la pala. Como resultado obtuvieron una turbina de 650 W con un  $C_p = 0.445$  para una velocidad de 5.5 m/s.

#### 1.1.4 Optimización de perfiles aerodinámicos para turbinas eólicas

El diseño y optimización de perfiles aerodinámicos son la base para la obtención de turbinas eólicas competentes ya que este es el elemento principal para mejorar su rendimiento aerodinámico. La diferencia de presión que se forma sobre el perfil aerodinámico debido al flujo de viento es lo que produce las fuerzas necesarias sobre las palas para producir el torque y generar energía eléctrica en la WT, la cual está en función del ángulo de ataque dispuesto en el perfil, siendo el ángulo de ataque el que se forma entre la línea de cuerda del perfil y la dirección del flujo de viento (Sadraey, 2013). En la Figura 8 se observan dos situaciones de distribución de presión sobre un perfil aerodinámico, siendo la superficie superior la de succión y la superficie inferior la de presión.

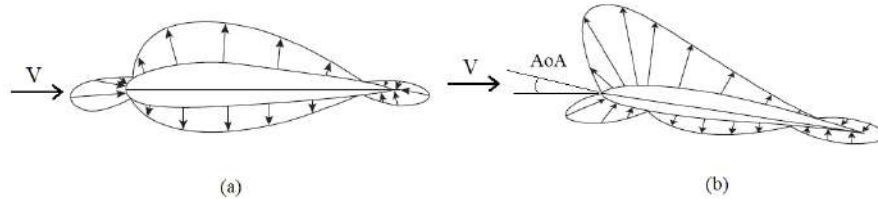


Figura 8. Distribución de presión sobre un perfil, a)  $AoA = 0$ , b)  $AoA \neq 0$ .

En las últimas décadas se han desarrollado perfiles aerodinámicos específicos para WT, entre ellos se encuentran las familias de perfiles desarrolladas por el NREL, la DUT, la serie RISO, y la serie FFA-W. Tangler & Somers (1995) desarrollaron en conjunto con el *Solar Energy Research Institute* (SERI) la familia de perfiles NREL-S aplicados específicamente a HAWT.

El diseño y optimización de perfiles aerodinámicos continúa siendo un área de desarrollo de acuerdo con las necesidades de aplicación, requerimientos de potencia y condiciones de operación. Los dos enfoques básicos del diseño de perfiles son el diseño inverso y los algoritmos de búsqueda directa (Sharma et al., 2020). El primero se basa en la optimización por gradiente y es útil cuando se involucran parámetros aerodinámicos como la presión del fluido, mientras que el segundo se basa en la optimización numérica y se emplea normalmente cuando se involucran parámetros de la forma aerodinámica (*e.g.* la geometría del perfil).

En los primeros años de desarrollo, el diseño inverso se empleó con bastante entusiasmo, sin embargo, los métodos de diseño automatizados que requieren un proceso iterativo son ampliamente utilizados en la actualidad. Sharma et al. (2020) establecen que los métodos de optimización de perfiles aerodinámicos se pueden dividir en cuatro categorías: 1) métodos de parametrización, 2) algoritmos genéticos, 3) métodos CFD, y 4) métodos adjuntos. Los métodos adjuntos se mencionarán de manera breve, mientras que los primeros tres serán abordados detalladamente en secciones posteriores.

Los métodos adjuntos son uno de los más recientes en el área de optimización y se emplean normalmente con herramientas CFD. En cuestión de diseño de perfiles aerodinámicos, los métodos adjuntos están relacionados con la optimización de la topología de la forma y del mallado del modelo geométrico, los cuales son procesos que en métodos iterativos representan un gran costo computacional (Sharma et al., 2020). Estos métodos se consideran como una técnica que identifica los cambios requeridos en los parámetros de entrada dentro del proceso de optimización para obtener los efectos deseados en los parámetros de salida o en la OF, debido a que es una manera efectiva de calcular el gradiente de una función. En otras palabras, estos métodos encuentran la forma óptima (*e.g.* geometría del perfil) bajo condiciones de operación preestablecidas (Kelecyc, 2021).

En un estudio (K. Wang et al., 2019) se optimizó el perfil RAE2822 con el objetivo de minimizar el  $C_D$  bajo condiciones restringidas de sustentación y el área de la geometría. El marco de trabajo combinó la parametrización geométrica mediante curvas *B-splines* y análisis CFD, mientras que el análisis de sensibilidad del mallado se realizó a través del método Galerkin empleado en las ecuaciones de Euler para flujo compresible con la finalidad de obtener los gradientes requeridos de las variables de diseño en el proceso de optimización de acuerdo con la teoría del método adjunto.

#### 1.1.4.1 Métodos de parametrización geométrica

La parametrización geométrica consiste en establecer un conjunto de variables que definen la ecuación de la geometría del perfil; un método de parametrización eficiente se caracteriza por manejar la mayor cantidad del espacio de diseño de la geometría con la mínima cantidad de variables de diseño necesarias. De acuerdo con Masters et al. (2017), los métodos de parametrización se dividen en: 1) constructivos y 2) deformativos. Los

primeros definen la geometría del perfil puramente mediante la magnitud de las variables de diseño, mientras que los segundos toman una geometría existente y la deforman hasta crear el nuevo perfil.

Entre los métodos constructivos más empleados están el método *B-splines*, el método de Transformación de Clase-Forma (*Class-Shape Transformation*, CST), el método SVD (*Singular-Value Decomposition*) y el método de secciones parametrizadas (PARSEC), mientras que en los métodos deformativos se encuentran las funciones Hicks-Henne, el método de deformación por Curvas de Bézier y el método de elementos del dominio de la función de base radial (*Radial-Base Function*, RBF). Cada uno de ellos presenta diferentes variables de diseño para describir la geometría del perfil aerodinámico.

Masters et al. (2017) realizaron un estudio comparativo de los siete métodos mencionados, para ello utilizaron una base de datos de más de 2000 perfiles aerodinámicos y reconstruyeron las geometrías empleando cada uno de los métodos con el objetivo de analizar la convergencia de los coeficientes  $C_D$  y  $C_L$  respecto al incremento de las variables de diseño. De igual manera, en cada reconstrucción de los perfiles obtuvieron la mejor aproximación mediante el método de mínimos cuadrados y calcularon el error de tolerancia geométrica. Los resultados indicaron que el método SVD es el más eficiente respecto al número de variables mínimo para cubrir el espacio de diseño del perfil, seguido del método CST.

El espacio de diseño disponible en métodos de parametrización es un factor importante para entender la dependencia de las variables de diseño al momento de crear o deformar una geometría. Grey & Constantine (2018) estudiaron el espacio de diseño activo de los métodos CST y PARSEC y calcularon los coeficientes  $C_L$  y  $C_D$  mediante CFD. El estudio de parametrización lo realizaron con el perfil de referencia NACA0012 y obtuvieron los nuevos perfiles utilizando 10 y 11 variables de diseño para los métodos CST y PARSEC, respectivamente. Con los resultados obtuvieron una estimación de las fronteras de las variables para cada método y concluyeron que los valores obtenidos de  $C_L$  tienen una influencia unidimensional del espacio de diseño, mientras que los valores de  $C_D$  mantienen una influencia bidimensional.

Anitha et al. (2018) optimizaron el perfil NACA4412 empleando una metodología basada en un método de parametrización, un algoritmo de optimización y un analizador del

rendimiento aerodinámico. Realizaron la optimización del perfil probando la combinación de distintas herramientas dentro de la metodología, las cuales fueron: 1) *B-splines*, algoritmo genético y XFOIL, 2) funciones Hicks-Henne, optimización por enjambre de partículas y XFOIL, y 3) PARSEC, optimización por enjambre de partículas y XFOIL, siendo la OF maximizar la relación  $C_L/C_D$  del perfil. De igual manera realizaron la simulación CFD del perfil optimizado para visualizar la distribución de presión en él.

En un estudio (Hansen, 2018) se optimizaron perfiles aerodinámicos para WT considerando la pérdida de rendimiento debido a la contaminación en el borde de ataque. Se implementó el método CST para parametrizar tres perfiles con espesores de 18%, 21% y 25%; como algoritmo de optimización se empleó el CMA-ES (*Covariance Matrix Adaptation-Evolution Strategy*) en conjunto de una función de penalidad adaptativa para restringir los criterios de diseño relacionados con el  $T_{max}$  y su posición, el espesor del borde de salida, el punto de funcionamiento en parada y la pérdida del  $C_L$ , entre otros. Por otra parte, para el análisis aerodinámico se utilizó una modificación del código XFOIL para predecir mejor el comportamiento en parada y la relación  $C_L/C_D$  del perfil.

En otro estudio, Abdelwahed & Abd El-Rahman (2020) optimizaron el perfil SG6043 para una HASWT teniendo en cuenta cuatro diferentes escenarios, los cuales fueron maximizar la relación  $C_L/C_D$  del perfil, obtener la distribución óptima del  $\alpha$  del perfil en la pala, la distribución de las variables de diseño del perfil respecto a su cuerda y la AEP de la pala con el perfil optimizado. En la Figura 9 se observa el diagrama de flujo de acuerdo con la metodología empleada. Utilizaron el método de curvas de Bézier para la parametrización de la geometría, lo cual generó la población inicial del algoritmo genético para el proceso de optimización, mientras que la evaluación de los coeficientes se realizó mediante el código XFOIL.

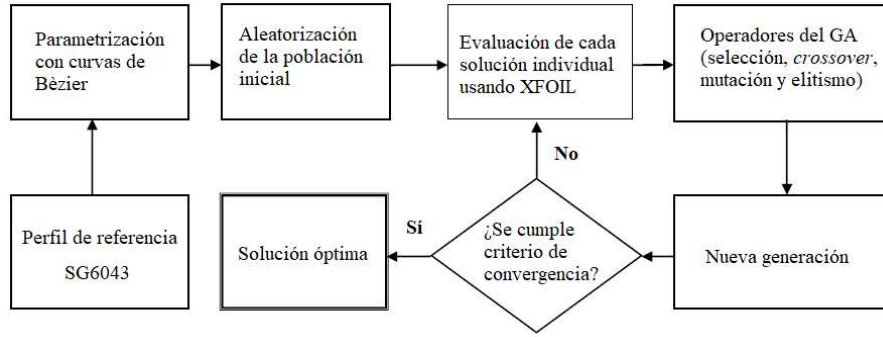


Figura 9. Diagrama de flujo del proceso de optimización (Abdelwahed & Abd El-Rahman, 2020).

En muchas ocasiones obtener el diseño optimizado de un perfil tiende a ser un proceso iterativo demasiado largo, sobre todo cuando el análisis aerodinámico se realiza con métodos CFD. Algunos investigadores han apostado por emplear modelos de aprendizaje profundos para reducir las dimensiones del espacio de diseño con el fin de seleccionar adecuadamente las geometrías efectivas y acelerar el proceso de optimización.

En un estudio (Chen et al., 2020) se emplearon redes neuronales generativas basadas en curvas de Bézier (BézierGAN) para parametrizar perfiles aerodinámicos de gran rendimiento en un tiempo menor y optimizarlos mediante un algoritmo de optimización. Por otra parte, Du et al. (2021) emplearon redes generativas adversariales basadas en el método *B-Splines* (BSplineGAN) y optimizaron perfiles óptimos mediante una combinación de un perceptrón multicapa y redes neuronales recurrentes.

Finalmente, algunos investigadores han sugerido modelos de parametrización alternativos para la reconstrucción de perfiles aerodinámicos los cuales consideran distintas variaciones de los parámetros del perfil (*e.g.* borde de ataque, borde de salida, ubicación del espesor máximo, etc.) dentro del espacio de diseño, tales como los trabajos realizados por Kostas et al. (2020) con su modelo de parametrización basado en un método de análisis isogeométrico del elemento límite, y por Negahban et al. (2023) con su modelo de parametrización de forma libre para la optimización del borde de salida. Por otra parte, Tortora et al. (2023) propusieron un modelo de parametrización para perfiles aerodinámicos basado en el método de curvas *B-Splines*, asegurando la continuidad de las superficies de las geometrías y flexibilidad en el manejo de los parámetros.

#### 1.1.4.2 Algoritmos Genéticos: Algoritmos Metaheurísticos

Como se mencionó anteriormente, los algoritmos de búsqueda directa basados en la optimización numérica se han empleado para optimizar perfiles aerodinámicos, así como también optimizar las palas de WT. De acuerdo con Hussain et al. (2019b), los métodos de optimización se dividen en métodos exactos y heurísticos, como se observa en la Figura 10.

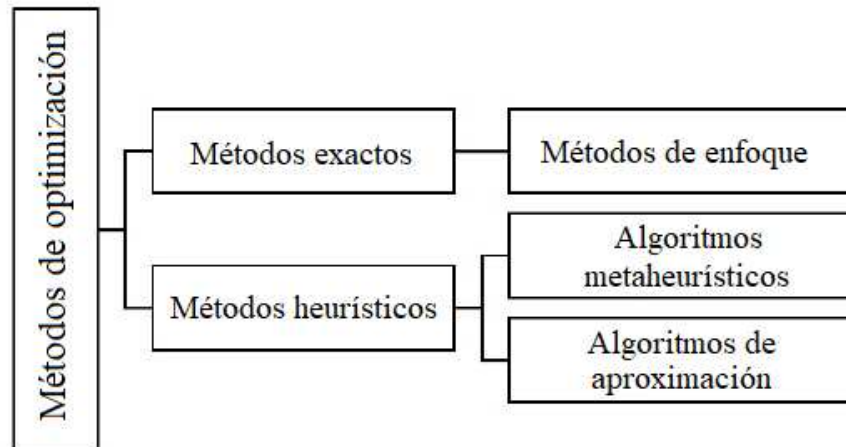


Figura 10. Clasificación de los métodos de optimización (Ramírez-Ochoa et al., 2022).

Los exactos se utilizan para obtener soluciones óptimas pero su costo computacional es muy grande, mientras que los heurísticos entregan soluciones cercanas a la solución óptima, pero con mayor rapidez (Ramírez-Ochoa et al., 2022). De los últimos se derivan los algoritmos metaheurísticos, los cuales están en un nivel más alto que el enfoque heurístico ya que estos algoritmos se caracterizan por ser más simples, precisos y rápidos, además de que su solución se basa en la inteligencia de conjuntos de poblaciones.

Ramírez-Ochoa et al. (2022) también señalan cuatro categorías de algoritmos metaheurísticos, clasificados de acuerdo con sus características y comportamientos: 1) basados en solución única o de población, 2) inspirados o no por la naturaleza, 3) iterativos o codiciosos, y 4) con o sin memoria. En la Figura 11 se observan las categorías y algunos algoritmos clasificados dentro de estas.



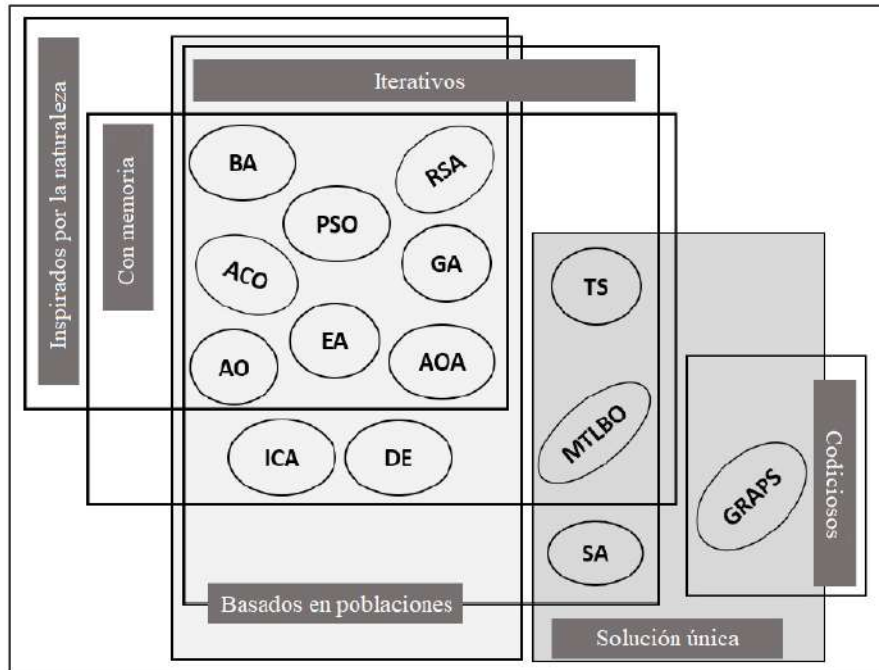


Figura 11. Clasificación de los algoritmos metaheurísticos (Ramírez-Ochoa et al., 2022).

Si bien hay algoritmos que comparten más de una característica, existen los llamados algoritmos de optimización evolutivos (*Evolutionary Optimization Algorithms*, EOA), los cuales se caracterizan principalmente por estar basados en poblaciones y/o ser inspirados por la naturaleza (Simon, 2013). Entre los EOA más conocidos se encuentran los algoritmos genéticos (*Genetic Algorithms*, GA), optimización por enjambre de partículas (*Particle Swarm Optimization*, PSO), y evolución diferencial (*Differential Evolution*, DE) que, al estar basados en poblaciones, proveen un conjunto de soluciones en cada iteración para mejorar la búsqueda local de la solución óptima mediante la información proporcionada por sus vecinos (soluciones individuales). Una de las características principales de los EOA es que la mayoría son inicializados con soluciones aleatorias.

Los GA son algoritmos pioneros en el área de optimización, están inspirados en la naturaleza de acuerdo con la teoría de evolución de Charles Darwin y han sido utilizados en una amplia variedad de problemas. Un ejemplo de GA es el estudio realizado por Perez-Cruz et al. (2012), en donde realizaron la optimización del factor Q de una micro viga en voladizo con pequeños huecos de separación considerando restricciones como el máximo esfuerzo permitido, la frecuencia fundamental y la constante del resorte de la viga.

En otra investigación (Pourrajabian et al., 2021) se optimizó la pala de una HAWT mediante dos GA, el primero utilizando un enfoque binario y el segundo con un enfoque continuo respecto a las variables de diseño. Se utilizaron las distribuciones del ángulo de ataque y la cuerda en la pala de la turbina como variables de diseño, mientras que la OF fue la maximización de potencia de la WT. Los resultados indicaron que el comportamiento del GA continuo superó al binario respecto a precisión y al tiempo computacional requerido, además de señalar que emplear un tamaño de población pequeño con un número de generaciones grande acelera la tasa de convergencia del problema.

En otro estudio (Akram & Kim, 2021b) se empleó un GA junto con el método CST para optimizar el perfil NREL-S821 para flujos subsónicos y el perfil RAE-2822 para flujos transónicos; se minimizó el  $C_D$  de los nuevos perfiles a un  $\alpha$  específico, mientras que para la evaluación de los coeficientes se empleó XFOIL. Los resultados de los perfiles optimizados se compararon a través de datos experimentales y análisis CFD.

En lo que respecta a los algoritmos DE, algunos autores mencionan que estos proveen una mejor convergencia global y robusticidad que los GA, además de requerir un menor número de parámetros. Wang et al. (2022) emplearon un algoritmo DE para extraer los parámetros desconocidos de corriente y voltaje del modelo inexplorado de un sistema fotovoltaico, con el objetivo de estimar su comportamiento. En otro estudio (Muratoglu et al., 2021) se empleó un algoritmo DE para optimizar la pala de una turbina hidro-cinética con control en parada, con lo cual obtuvieron perfiles optimizados respecto a su comportamiento de sustentación y arrastre.

Algunos estudios han realizado comparaciones entre los GA y los algoritmos PSO mediante diversas funciones de prueba, como es el caso de Hassan et al. (2005), quienes indican que el PSO mantiene una efectividad similar a los GA, pero mejora significativamente el costo computacional. Yousaf et al. (2020) implementaron un algoritmo PSO con una integral de error al cuadrado como OF para optimizar el comportamiento de una nano-red eléctrica inteligente de tres fuentes energéticas.

El algoritmo PSO fue desarrollado por James Kennedy y Russel Eberhart en 1995 y ha sido utilizado ampliamente en problemas de optimización, este algoritmo se basa tanto en el comportamiento colectivo como en el individual de cada integrante de la población.

Algunos conceptos básicos que conforman al PSO son la “inercia”, la “influencia de la sociedad”, y “la influencia de los vecinos” (Simon, 2013).

Sin embargo, debido a que cada problema de optimización varía en complejidad de acuerdo con el área de disciplina, diversos autores han optado por estudiar variaciones del algoritmo con la finalidad de mejorar su efectividad para encontrar soluciones globales óptimas a los problemas. Khurana & Massey (2015) desarrollaron un algoritmo AM-PSO (*Adaptative Mutation-Particle Swarm Optimization*) en donde se empleó un operador de mutación con el fin de mejorar la viabilidad computacional del algoritmo y de la solución. Igualmente, aplicaron el AM-PSO en funciones de prueba y en la optimización de un perfil aerodinámico logrando reducir la diversidad de soluciones y redireccionando la solución a la óptima global.

La mayoría de los problemas de optimización reales se consideran como problemas multiobjetivo (Simon, 2013), en donde la solución óptima se consigue mediante el cumplimiento de más de una meta, sin embargo, dependiendo la naturaleza del problema, estas metas pueden entrar en conflicto (*e.g.* minimizar el costo de un puente y maximizar su resistencia). Un ejemplo es el estudio realizado por Kaviani & Nejat (2017) en el cual optimizaron una HAWT WindPact 1.5 MW empleando el método CST y un algoritmo MOPSO (*Multi-Objective Particle Swarm Optimization*). Las funciones objetivo utilizadas fueron la maximización de potencia de la turbina y la minimización del nivel medio de presión sonora de esta, mientras que para su evaluación emplearon el modelo BEM y un modelo aeroacústico, respectivamente.

Asimismo, Zhang et al. (2020) optimizaron el perfil CRA09-A empleado en WT mediante el método CST y un EOA basado en la solución de Pareto. Este estudio consistió en mejorar la capacidad de maniobra y de operación estacionaria de la turbina sin reducir el espesor del perfil aerodinámico; utilizaron un enfoque de restricciones múltiples relacionadas a los coeficientes  $C_L$  y  $C_D$ , así como del espesor máximo del mismo. Por otra parte, las OF fueron maximizar el coeficiente  $C_L$  y la relación  $C_L/C_D$  del perfil optimizado dentro de un rango de números de Match y de  $\alpha$ . Finalmente validaron el comportamiento del nuevo perfil mediante experimentación en un túnel de viento.

Por otra parte, algunas investigaciones han empleado algoritmos híbridos para la optimización de perfiles aerodinámicos, lo cual es la combinación de dos algoritmos. Tal es

el caso de Masoumi & Jalili (2019), quienes optimizaron un perfil aerodinámico empleando un algoritmo compuesto por el PSO y el algoritmo *Big Bang-Big Crunch*. Implementaron el enfoque de diseño inverso utilizando la diferencia de presión entre la geometría base y la deseada; también usaron el enfoque de búsqueda directa mediante tres funciones objetivo: 1) maximizar la relación  $C_L/C_D$ , 2) maximizar el  $C_L$  con un  $C_D$  constante, y 3) minimizar el  $C_D$  con un  $C_L$  constante, mientras que para la evaluación de estas funciones emplearon un modelo de ecuaciones de Euler.

#### 1.1.4.3 Métodos CFD

Como se mencionó antes, los métodos CFD han sido empleados en los últimos años en estudios de optimización de perfiles aerodinámicos y HAWT, principalmente debido a los grandes avances de la computación. Una manera de reducir el costo computacional de las simulaciones numéricas es mediante la inclusión de modelos de ecuaciones de turbulencia, esto se realiza empleando las ecuaciones RANS y LES. Este tipo de ecuaciones mantienen la misma estructura con la única diferencia de que las RANS manejan escalas de tiempo mucho más grandes que las escalas típicas de las LES durante el análisis del flujo del fluido, por lo que las segundas requieren un poco más de recurso computacional (Heinz, 2020).

Las ecuaciones RANS para flujos compresibles e incompresibles se han aplicado usualmente en problemas donde el número de Reynolds es relativamente pequeño (*e.g.* HAWT) y es necesario el análisis de los efectos de viscosidad del fluido en la capa límite (Sharma et al., 2020). Por otra parte, la investigación realizada por Heinz (2020) propone el desarrollo de métodos de turbulencia híbridos entre las ecuaciones RANS-LES para simular mejor el comportamiento de la separación del flujo de fluido en aplicaciones aeroespaciales y en WT, concluyendo que los métodos híbridos muestran un panorama favorecedor en el área de CFD a pesar de que aún se encuentran en desarrollo.

En un estudio (Yang et al., 2014) se empleó CFD con el modelo de turbulencia SST  $k-\omega$  para obtener las cargas aerodinámicas de las palas del rotor MEXICO y los coeficientes  $C_L$  y  $C_D$ , así como para determinar la presión en los perfiles de la pala en  $\alpha$  cercanos a la región raíz. Posteriormente, se analizó el comportamiento del rotor mediante

un modelo BEM mejorado con las correcciones necesarias para incluir las fuerzas tangenciales en este, con la premisa de que el modelo BEM clásico no considera estos efectos en 3D. Los resultados indicaron que los datos extraídos con CFD corresponden al comportamiento real del rotor mediante datos experimentales.

Un estudio similar (Bouhelal et al., 2017) se realizó con el mismo rotor MEXICO para comparar el modelo BEM y el método CFD con el modelo de turbulencia  $k-\omega$  para distintas relaciones de velocidades de punta de la pala de la turbina. Los resultados coincidieron en que las predicciones aerodinámicas obtenidas mediante CFD son más precisas. Asimismo, Noronha & Krishna (2021) emplearon CFD con el modelo de turbulencia  $k-\omega$  para determinar los perfiles aerodinámicos más adecuados para HASWT de entre una base de datos de perfiles disponibles en la literatura. El análisis se realizó con diferentes  $Re$  y  $\alpha$ , mientras que la selección se obtuvo con los valores de  $C_L$  y la relación  $C_L/C_D$ , siendo el perfil SG6043 el más adecuado para bajas velocidades.

Por otra parte, Belamadi et al. (2016) analizaron el comportamiento de un perfil S809 ranurado mediante métodos CFD. El análisis numérico en 2D se realizó con los modelos de turbulencia Standard  $k-\epsilon$ , Spallart-Allmaras,  $k-\omega$  y SST  $k-\omega$ . Los resultados indicaron que el sistema de control de ranurado mejoró el comportamiento del perfil únicamente en el rango de ángulos de ataque de  $10^\circ \leq \alpha \leq 20^\circ$ , mientras que para ángulos pequeños aumentó el arrastre del perfil ranurado comparado con el original.

Asimismo, Morgado et al. (2016) compararon el análisis aerodinámico de los perfiles E387 y S1223 del código XFOIL con el modelo de turbulencia SST  $k-\omega$  y el modelo de transición  $k-kl-\omega$  empleados en CFD. Los análisis se realizaron para un  $Re = 2.0 \times 10^5$  y se compararon con los parámetros obtenidos mediante datos experimentales. Los resultados indicaron que el código XFOIL muestra valores suficientes para el análisis en 2D de la geometría del perfil, sin embargo, recalcaron la importancia de implementar modelos CFD fiables para simulaciones en 2D en estado estacionario, así como en análisis en estado transitorio y con geometrías en 3D.

Los métodos CFD son utilizados comúnmente en optimización de perfiles aerodinámicos como herramienta para analizar los parámetros de presión,  $C_L$  y  $C_D$  dentro de un marco de trabajo Parametrización-Optimización-Analizador (POA). En un estudio (T. Zhang et al., 2016) se optimizó el perfil NACA0012 empleando el método CST, un GA

multi-isla y CFD. Para la simulación numérica se realizó un análisis de la densidad del mallado y se utilizó el modelo de turbulencia Spalart-Allmaras, mientras que el rendimiento del algoritmo se comparó con un algoritmo de programación no lineal cuadrática de Lagrange. Los resultados del perfil óptimo indicaron una mejora de los coeficientes  $C_L$ ,  $C_D$  y su relación, señalando como prospectiva la mejora del rendimiento en parada del perfil.

En otro estudio realizado por Benim et al. (2018) se realizó la optimización en 2D de perfiles para maximizar el empuje y la estabilidad de una HASWT. En el proceso se empleó la parametrización con el método de curvas Hermite para crear la geometría, se empleó el método RSM (*Response Surface Methodology*) para relacionar las variables de diseño ( $Re$  y  $\alpha$ ) con las OF. Por otro lado, el algoritmo BiMADS (*Biobjective Mesh Adaptive Direct Search*) se utilizó para optimizar las variables de diseño, mientras que el modelo SST k- $\omega$  se aplicó para el cálculo de los coeficientes y la evaluación de las OF.

Finalmente, Rodríguez & Celis (2022) emplearon CFD acoplado a un GA para optimizar el perfil NACA4412 en la región media de la pala con la finalidad de maximizar la potencia de una HASWT de 5kW. Utilizaron curvas de Bézier para parametrizar la geometría y el modelo de turbulencia SST k- $\omega$  para el cálculo de los coeficientes del perfil, mientras que el diseño de la turbina se hizo mediante el modelo BEM. Los resultados indicaron un coeficiente de potencia máximo de  $C_p = 0.4658$  y señalaron que los perfiles ubicados en un 25% al 90% a lo largo de la pala son representativos para maximizar la potencia.

Algunos otros estudios que emplean el marco de trabajo POA para la optimización de perfiles aerodinámicos y WT son los realizados por Wu et al. (2019), Kim et al. (2020), Akram & Kim (2021a), Akbari et al. (2022) y Kaviani & Moshfeghi (2023a), en los cuales se emplean diversos métodos de parametrización, algoritmos de optimización y herramientas de análisis de las cargas aerodinámicas.

En este estudio se realizó la optimización del perfil aerodinámico S809 empleando características del flujo de viento adecuadas para HASWT. La metodología desarrollada empleó un marco de trabajo POA, el cual combinó el método de parametrización CST, el algoritmo de optimización PSO, y el modelo de turbulencia SST k- $\omega$  empleado mediante análisis CFD. Se consideró como función objetivo la maximización del  $C_L$  dentro del régimen de máxima sustentación del perfil aerodinámico, mientras que el

acoplamiento de las herramientas de optimización y análisis se hizo con MATLAB y ANSYS Workbench.

## 1.2 DESCRIPCIÓN DEL PROBLEMA

Aún existe la necesidad de desarrollar diseños óptimos de HASWT para superar las desventajas intrínsecas que tienen respecto al recurso eólico disponible, por lo que mejorar su rendimiento aerodinámico es vital para que se consoliden y sean utilizadas ampliamente en el mercado. Adicionalmente, se identificó que los estudios de optimización de perfiles aerodinámicos no han sido lo suficientemente explorados considerando la operación del perfil en su régimen de máxima sustentación. Por otra parte, los métodos CFD se han utilizado para el estudio de las cargas aerodinámicas en perfiles considerando los efectos del flujo turbulento, sin embargo, aún continúan en investigación y desarrollo. Además, los métodos CFD no se han utilizado lo suficiente como herramienta de análisis aerodinámico en estudios que emplean el marco de trabajo Parametrización-Optimización-Analizador.

## 1.3 JUSTIFICACIÓN

### 1.3.1 Aportes técnicos-científicos

Optimizar un perfil aerodinámico mediante la maximización del coeficiente de sustentación en su régimen de máxima sustentación permitirá mejorar el comportamiento general del perfil en todos sus regímenes de trabajo, además de que incrementar el coeficiente de sustentación máximo del perfil permitirá incrementar la potencia pico del rotor durante el proceso de diseño de la turbina eólica. De igual manera, el acoplamiento de las herramientas empleadas en el marco de trabajo POA, permitirá automatizar el proceso de optimización debido a la facilidad del uso de *scripts* en cada proceso interno de ANSYS Workbench, lo cual se ha conseguido como consecuencia de los avances computacionales y de compatibilidad en los últimos años de los softwares empleados.

### 1.3.2 Impacto social

El recurso eólico disponible en regiones o aplicaciones donde las velocidades del viento son bajas normalmente es menor al requerido por las turbinas comerciales, por lo que desarrollar diseños óptimos de perfiles aerodinámicos para ese rango de velocidades permitiría mejorar el rendimiento de HASWT con esas condiciones y obtener más beneficio del uso de estos sistemas. La aportación de este trabajo permite optimizar perfiles aerodinámicos únicamente empleando recurso computacional, lo cual reduce los tiempos y costos del proceso en comparación con experimentación de prototipos en túneles de viento.

## 1.4 HIPÓTESIS Y OBJETIVOS

A continuación, se plantea la hipótesis y se definen el objetivo general y los objetivos específicos, los cuales son la base para el desarrollo de la metodología propuesta.

### 1.4.1 Hipótesis

La optimización del perfil aerodinámico de la pala de una turbina eólica de eje horizontal de baja potencia se obtiene mediante la maximización del coeficiente de sustentación del perfil en su régimen de máxima sustentación.

### 1.4.2 Objetivo general

Optimizar el perfil aerodinámico S809 mediante la maximización de su coeficiente de sustentación correspondiente a su régimen de máxima sustentación. Para la parametrización se empleará el método CST y para la optimización se usará el algoritmo PSO desarrollados en MATLAB, mientras que para el análisis aerodinámico se utilizará un modelo de turbulencia CFD en ANSYS Workbench.

### 1.4.3 Objetivos específicos

- a) Definir las características y condiciones de trabajo del perfil aerodinámico S809 en el régimen de máxima sustentación para obtener la curva



característica de su coeficiente de sustentación mediante bases de datos experimentales.

- b) Definir la función objetivo y las variables de diseño del perfil aerodinámico a través de los coeficientes de la función de forma del método CST.
- c) Implementar el análisis del perfil aerodinámico S809 en ANSYS Workbench con un modelo de turbulencia CFD para validar el método de análisis.
- d) Desarrollar el algoritmo PSO empleando MATLAB para establecer los parámetros de control iniciales del algoritmo y la población inicial del enjambre.
- e) Acoplar el optimizador y el analizador empleando el marco de trabajo POA con MATLAB y ANSYS Workbench para maximizar el coeficiente de sustentación del perfil aerodinámico en el régimen de máxima sustentación.
- f) Comparar el rendimiento del perfil aerodinámico óptimo con el rendimiento del perfil S809 mediante las curvas características de sus coeficientes de sustentación para validar la mejora del diseño del perfil.

## **1.5 PLANTEAMIENTO GENERAL**

En la Figura 12 se observa el diagrama del planteamiento general de este trabajo, cuya finalidad fue servir como guía para cumplir con el objetivo general y los objetivos específicos planteados y así validar la hipótesis descrita.

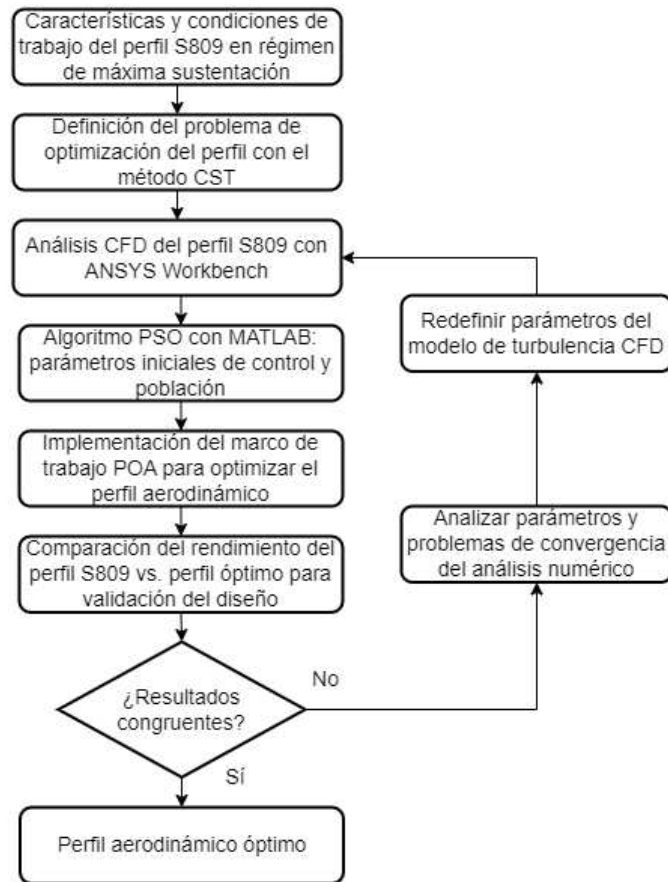


Figura 12. Diagrama del planteamiento general de la metodología.

En el diagrama de la Figura 12 se observa que después de haber analizado mediante CFD el perfil S809, de implementar el marco de trabajo POA para obtener el perfil aerodinámico óptimo y de comparar el rendimiento del perfil S809 con el rendimiento de perfil óptimo mediante las curvas del coeficiente de sustentación, se analizará la congruencia de los resultados para validar la metodología y la mejora del diseño del perfil aerodinámico; en caso de que se muestre incongruencia en los resultados, se optará por analizar los parámetros de la optimización, así como los posibles problemas de convergencia del análisis numérico, lo que llevará a redefinir ciertos parámetros, principalmente aquellos que influyen en el modelo de turbulencia empleado en la simulación CFD.

## II. FUNDAMENTACIÓN TEÓRICA

Las siguientes secciones contienen los fundamentos teóricos de los métodos y herramientas que se emplean en la metodología de este proyecto, por lo que a continuación se muestran los aspectos y ecuaciones necesarias para su comprensión.

### 2.1 PERFIL AERODINÁMICO Y CONCEPTOS GENERALES DE AERODINÁMICA

En esta sección se describe la geometría del perfil aerodinámico y los conceptos generales de aerodinámica que influyen en el comportamiento del perfil.

#### 2.1.1 Geometría del perfil aerodinámico

En la Figura 13 se observan los parámetros geométricos típicos de un perfil aerodinámico.

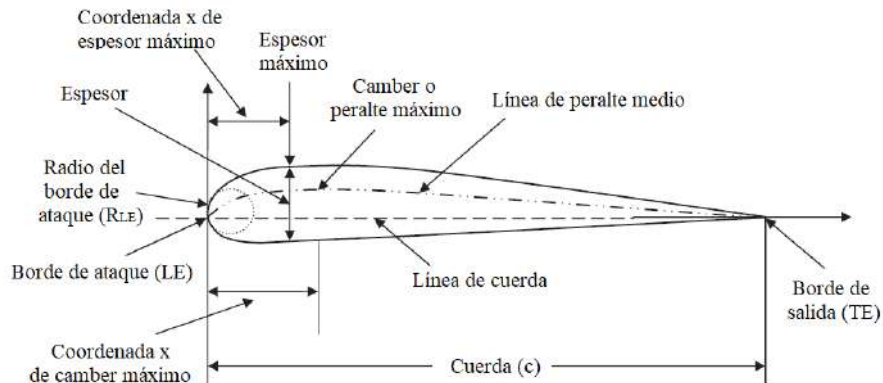


Figura 13. Parámetros geométricos de un perfil aerodinámico (Sadraey, 2013).

Como se muestra en la Figura 13, la cuerda ( $c$ ) es la longitud característica del perfil y es aquella que mide la longitud entre el borde de ataque (*Leading Edge*, LE) y el borde de salida (*Trailing Edge*, TE), siendo estos la entrada y salida del flujo, respectivamente. Por otra parte, el peralte o *camber* es el abultamiento que se genera normalmente en la superficie superior del perfil, tal que, si la línea de *camber* media es una línea recta, se dice que el perfil es simétrico. El radio del borde de ataque ( $R_{LE}$ ) es el radio que se forma entre el inicio de las superficies superior e inferior en la región del borde de

ataque, mientras que el espesor máximo ( $T_{max}$ ) es aquella distancia máxima que se forma de manera perpendicular entre las superficies superior e inferior (Sadraey, 2013).

### 2.1.2 Comportamiento del perfil aerodinámico

Cualquier sección transversal que corta la pala de una WT es llamada perfil aerodinámico. En un perfil con *camber* positivo la presión estática en la superficie superior es menor que la presión del ambiente, mientras que la presión estática en la superficie inferior es mayor a esta. Esta diferencia de presión, la cual incrementa a medida que aumenta el ángulo de ataque del perfil, se genera debido al incremento de la velocidad del viento en la superficie superior respecto a la superficie inferior (Sadraey, 2013). En la Figura 14 se muestra una representación del cambio del flujo de la velocidad en un perfil con diferentes AoA.

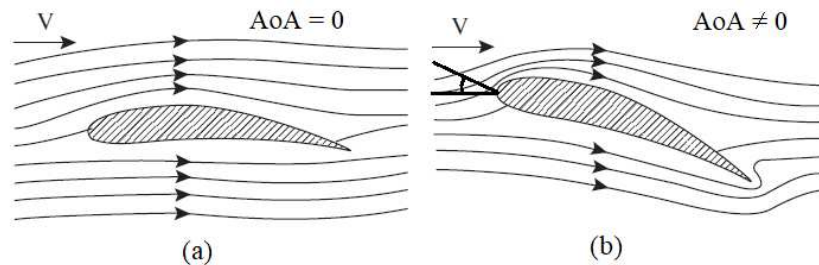


Figura 14. Cambio de flujo de velocidad sobre un perfil. a)  $AoA = 0$ , b)  $AoA \neq 0$ .

La presión total del fluido es la integración de la presión sobre cada punto de la superficie del perfil. La fuerza resultante ( $F$ ) debido al flujo se ubica en el centro de presión ( $cp$ ) geométrico del perfil, y la ubicación del  $cp$  depende de la velocidad de flujo y del  $\alpha$  del perfil, es decir, a velocidades pequeñas el  $cp$  se encuentra cercano al borde de ataque y a velocidades mayores, se mueve hacia el borde de salida. Por otra parte, el centro aerodinámico ( $ac$ ) es un concepto importante para la estabilidad del perfil, ya que si se considera este punto como la ubicación donde se aplica la fuerza resultante, se genera un momento de cabeceo ( $M_o$ ) (*pitching moment*) en la geometría, como se muestra en la Figura 15. Este momento se visualiza como el producido por la fuerza resultante actuando a una distancia particular del borde de ataque. De acuerdo con la teoría subsónica de perfiles

aerodinámicos, el  $ac$  se encuentra a una distancia del 25% de la cuerda del perfil desde su borde de ataque (Sadraey, 2013).

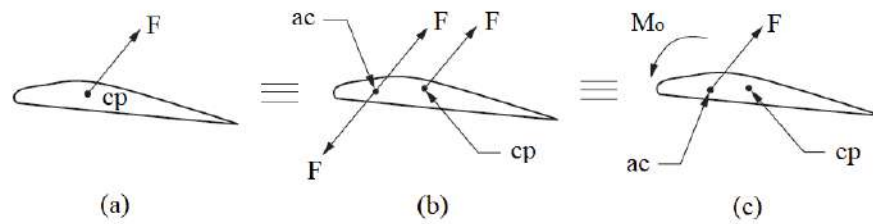


Figura 15. Movimiento de la fuerza resultante. a) fuerza en el  $cp$ , b) adición de fuerzas equivalentes, c) fuerza en el  $ac$  (Sadraey, 2013).

Por lo tanto, la fuerza aerodinámica resultante sobre un perfil se descompone en la fuerza de sustentación ( $L$ ), la fuerza de arrastre ( $D$ ) y en el momento de cabeceo ( $M_o$ ). La fuerza de sustentación es la componente perpendicular a la dirección relativa del viento, mientras que la fuerza de arrastre es la componente paralela a esta, como se observa en la Figura 16.

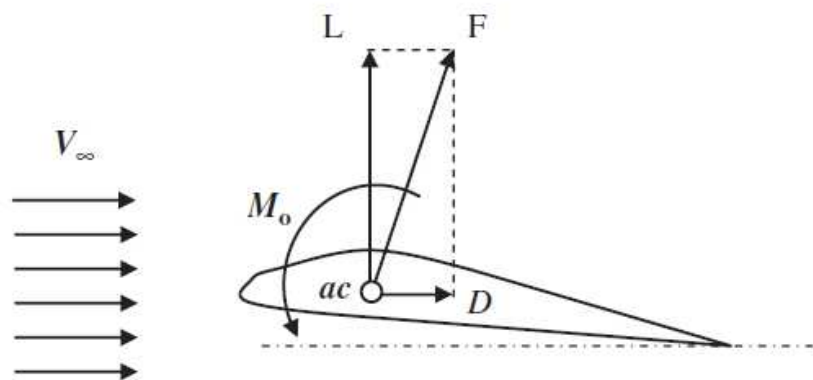


Figura 16. Fuerzas de sustentación, arrastre y momento de cabeceo en el  $ac$  (Sadraey, 2013).

La selección y diseño de perfiles aerodinámicos normalmente involucra el uso de los coeficientes bidimensionales de estas fuerzas y del momento, determinados por el  $\alpha$  del perfil y el número de Reynolds del fluido que actúa sobre él (Sadraey, 2013). En las ecuaciones (1), (2) y (3) se definen los coeficientes de sustentación ( $C_L$ ), arrastre ( $C_D$ ) y de momento de cabeceo ( $C_m$ ), respectivamente.

$$C_L = \frac{L}{\frac{1}{2}\rho V^2(c \cdot 1)} \quad (1)$$

$$C_D = \frac{D}{\frac{1}{2}\rho V^2(c \cdot 1)} \quad (2)$$

$$C_m = \frac{M_o}{\frac{1}{2}\rho V^2 c(c \cdot 1)} \quad (3)$$

Donde:

$\rho$  = densidad del aire

$V$  = velocidad del viento relativo

$c$  = cuerda del perfil

$(c \cdot 1)$  = cuerda del perfil por tramo unitario

Por otra parte, el coeficiente de presión ( $C_p$ ) que actúa sobre el perfil aerodinámico y el número de Reynolds ( $Re$ ) se definen en las ecuaciones (4) y (5), respectivamente.

$$C_p = \frac{p - p_\infty}{\frac{1}{2}\rho V^2} \quad (4)$$

$$Re = \frac{VL_{flow}}{v} = \frac{\rho VL_{flow}}{\mu} \quad (5)$$

Donde:

$p$  = presión sobre la superficie del perfil

$p_\infty$  = presión del fluido a una distancia alejada del perfil

$V$  = velocidad del viento relativo

$L_{flow}$  = longitud característica del perfil en el dominio del fluido

$\rho$  = densidad del fluido

$\mu$  = viscosidad dinámica del fluido

$v = \mu/\rho$  = viscosidad cinemática del fluido

Los perfiles aerodinámicos para HASWT son diseñados para trabajar con  $\alpha$  pequeños, donde la fuerza  $L$  es mayor que la fuerza  $D$ , no obstante, a medida que aumenta el ángulo de ataque del perfil, la sustentación disminuye y el arrastre aumenta. Usualmente el comportamiento adecuado de un perfil aerodinámico se evalúa de acuerdo con el comportamiento de estos coeficientes, siendo los más importantes los siguientes: a)  $C_L$  vs  $\alpha$ , b)  $C_D$  vs  $C_L$ , y 3)  $C_L/C_D$  vs  $\alpha$ .

### 2.1.3 Gráficas características de un perfil aerodinámico

#### 2.1.3.1 $C_L$ vs. $\alpha$

La gráfica típica del comportamiento  $C_L$  vs.  $\alpha$  se observa en la Figura 17.

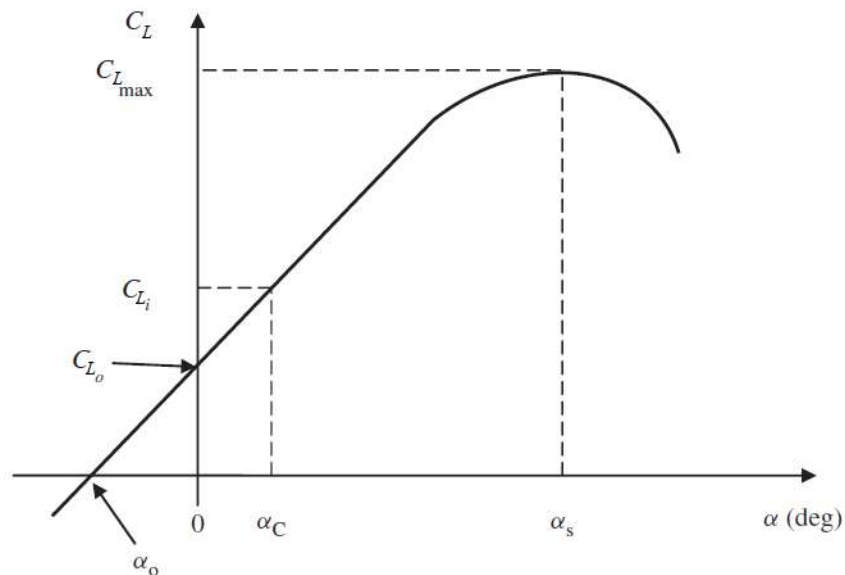


Figura 17. Variación típica del  $C_L$  vs.  $\alpha$  (Sadraey, 2013).

El ángulo de pérdida ( $\alpha_s$ ) es el ángulo en el cual el perfil entra en “pérdida”, esto significa que el  $C_L$  ya no incrementa a pesar de que el ángulo de ataque se hace mayor. El coeficiente de sustentación máximo ( $C_{L,max}$ ) es el valor máximo de sustentación que puede alcanzar el perfil, normalmente sucede en el punto de  $\alpha_s$ . En la práctica, perfiles diseñados con estos parámetros con valores lo más grande posibles son deseables. El ángulo de cero sustentación ( $\alpha_o$ ) es aquel en donde no existe fuerza de sustentación en el perfil. El

coeficiente de sustentación ideal ( $C_{L,i}$ ) es aquel en donde el arrastre del perfil no varía significativamente con variaciones del  $\alpha$ , siendo este usualmente el que corresponde al coeficiente de arrastre mínimo. Al  $C_{L,i}$  también corresponde el ángulo de ataque  $\alpha_c$ , mientras que el coeficiente de sustentación ( $C_{L,o}$ ) es el correspondiente a un ángulo de ataque de cero (Sadraey, 2013).

### 2.1.3.2 $C_D$ vs. $C_L$

La gráfica típica del comportamiento  $C_D$  vs.  $C_L$  para un perfil aerodinámico con *camber* positivo se observa en la Figura 18.

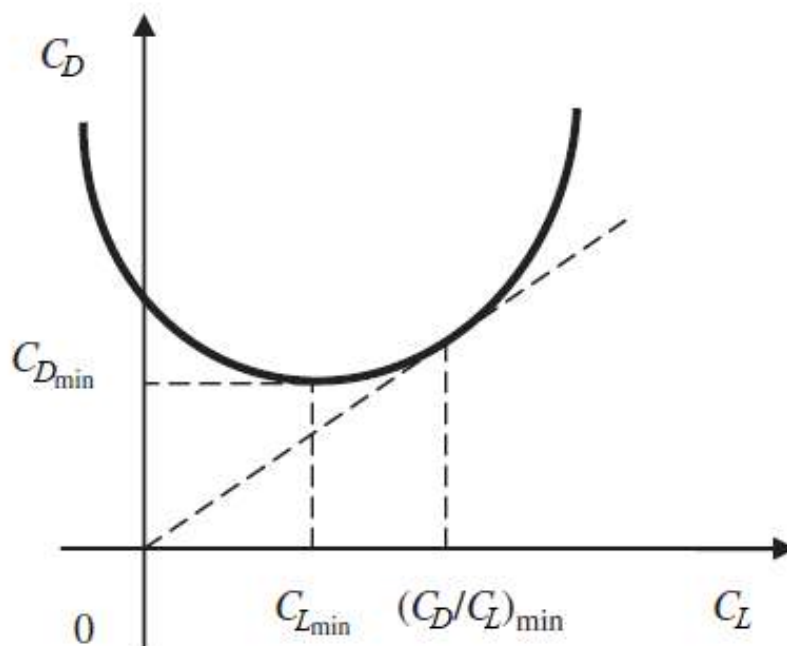


Figura 18. Variación típica del  $C_D$  vs.  $C_L$  (Sadraey, 2013).

El punto mínimo de la gráfica es el coeficiente de arrastre mínimo ( $C_{D,min}$ ) el cual corresponde al coeficiente de sustentación mínimo ( $C_{L,min}$ ). Por otra parte, la línea recta que pasa por el origen y cruza de manera tangente la gráfica denota la pendiente mínima, la cual es de gran importancia ya que corresponde al momento de operación del perfil cuando la relación sustentación-arrastre es máxima, dado que  $C_D/C_{L,min} = C_L/C_{D,max}$ .



La obtención del  $C_L/C_{D_{max}}$  y el  $C_{D,min}$  son características deseables de vital importancia en el diseño de perfiles aerodinámicos, sin embargo, usualmente estas no suceden al mismo tiempo (Sadraey, 2013).

La Figura 19 muestra la variación típica del  $C_D$  vs.  $C_L$  para un perfil sobre un flujo laminar, en donde la principal diferencia es la forma de cubo en la parte mínima de la gráfica. Esto sucede debido a que el  $C_{D,min}$  no varía dentro de un rango de coeficientes de sustentación, lo cual permite mantener el perfil con un arrastre mínimo en un rango de ángulos de ataque. Los puntos en la forma de cubo donde se encuentran el  $C_{L,i}$  y el  $C_{L,d}$  son los coeficientes de sustentación ideal y de diseño, respectivamente. El  $C_{L,d}$  ocurre normalmente cuando la relación de los coeficientes de sustentación y arrastre es máxima (Sadraey, 2013).

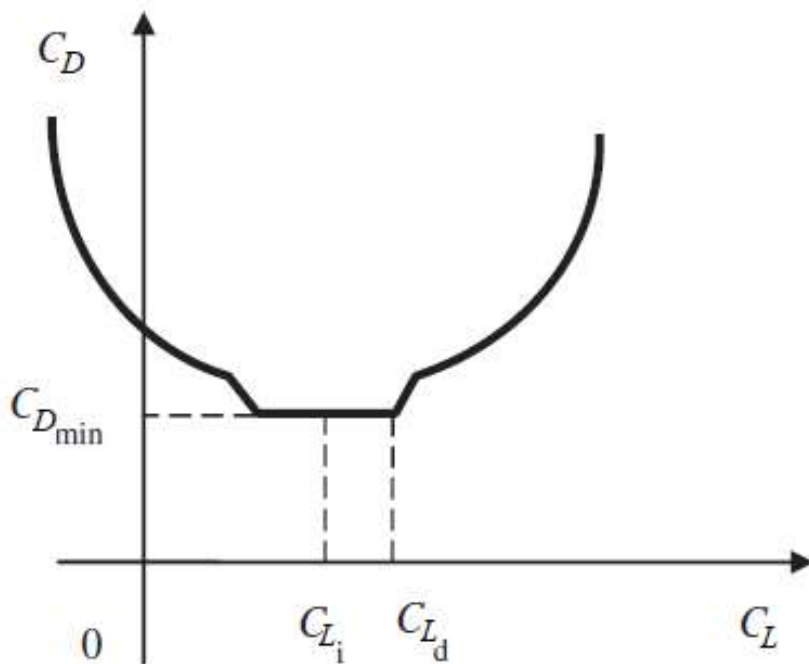


Figura 19. Variación típica del  $C_D$  vs.  $C_L$  para un perfil con flujo laminar (Sadraey, 2013).

### 2.1.3.3 $C_L/C_D$ vs. $\alpha$

La gráfica típica del comportamiento  $C_L/C_D$  vs.  $\alpha$  se observa en la Figura 20.

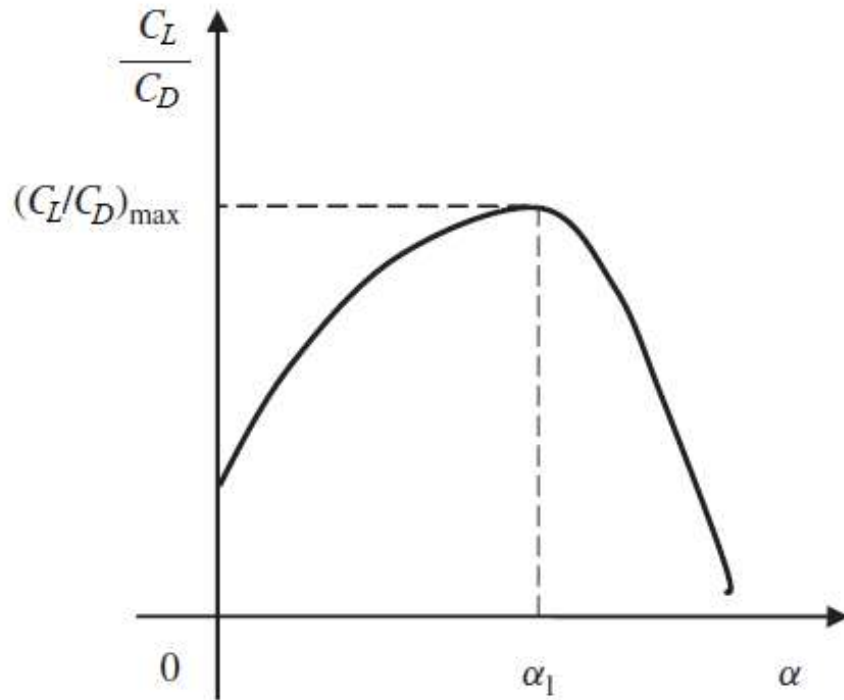


Figura 20. Variación típica del  $C_L/C_D$  vs.  $\alpha$  (Sadraey, 2013).

La característica más importante de la gráfica de la Figura 20 se observa en el punto máximo de esta, el cual corresponde al valor máximo de la relación de los coeficientes de sustentación y arrastre  $\left(\frac{C_L}{C_{D_{\max}}}\right)$  y a su ángulo de ataque ( $\alpha_1$ ) en donde se considera el momento óptimo en el que el perfil aerodinámico puede operar (Sadraey, 2013).

#### 2.1.4 Regímenes de flujo de un perfil aerodinámico

Como se mencionó en los antecedentes, un perfil aerodinámico opera dentro de tres regímenes de flujo: 1) régimen de flujo adjunto, 2) régimen de máxima sustentación y 3) régimen de estancamiento total. En la Figura 21 se observa el comportamiento de los coeficientes del perfil S809 para un  $Re = 7.5 \times 10^5$ , el cual es un perfil comúnmente utilizado en HASWT y se utilizará para describir los regímenes (Manwell et al., 2010).

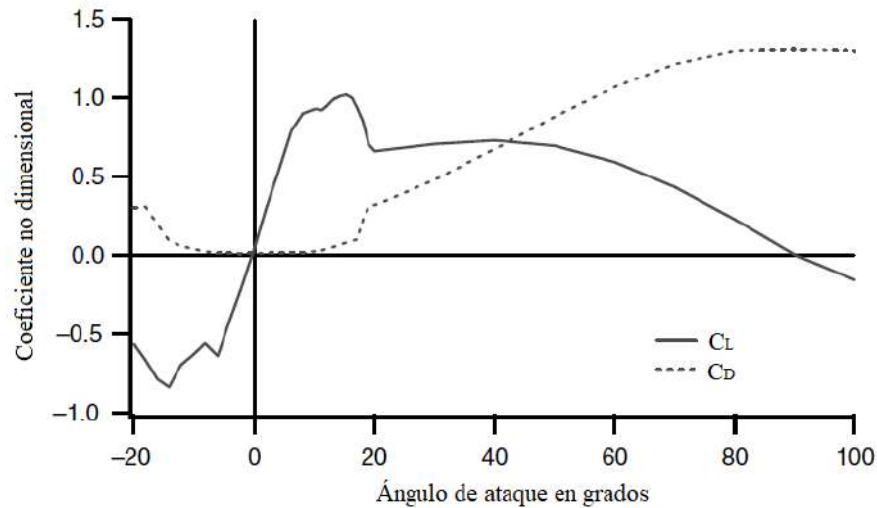


Figura 21.  $C_L$  y  $C_D$  del perfil S809,  $Re = 7.5 \times 10^5$  (Manwell et al., 2010).

El régimen de flujo adjunto es aquel en el que el flujo del fluido está unido a la superficie superior del perfil; se caracteriza porque el  $C_L$  incrementa con el ángulo de ataque y el  $C_D$  es relativamente bajo. Para el perfil S809 este régimen abarca aproximadamente hasta un  $\alpha = 10^\circ$  (Manwell et al., 2010).

El régimen de máxima sustentación es aquel en el que el  $C_L$  obtiene su máximo valor a medida que el perfil entra en pérdida. Cuando el perfil aerodinámico alcanza el ángulo de pérdida  $\alpha_s$  la superficie superior del perfil experimenta una separación de la capa límite del fluido, lo cual produce la formación de estelas (turbulencia), como se muestra en la Figura 22, y por lo tanto la sustentación se reduce y se incrementa el arrastre (Manwell et al., 2010).

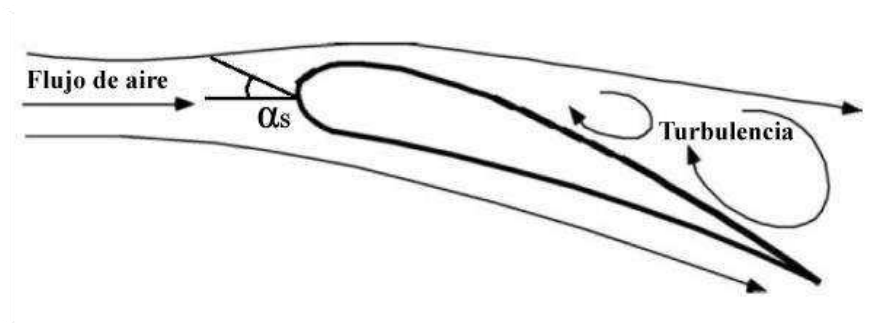


Figura 22. Pérdida de sustentación del perfil por turbulencia en el ángulo  $\alpha_s$ .

Por otra parte, el régimen de estancamiento total se produce inmediatamente después de que el perfil alcanza el ángulo de pérdida hasta llegar aproximadamente a  $90^\circ$ . En este régimen el perfil pierde sustentación e incrementa su arrastre, actuando prácticamente como una placa plana (Manwell et al., 2010).

### 2.1.5 Teoría de placa plana

La relación  $C_L/C_D$  dentro de la teoría de placa plana proporciona una guía útil para el comportamiento de perfiles aerodinámicos y palas de WT que consideran un flujo del fluido totalmente separado de la geometría (J. Tangler & Kocurek, 2005). Las variaciones del  $C_L$ ,  $C_D$  y  $C_L/C_D$  vs  $\alpha$  para la teoría de placa plana se muestran en la Figura 23.

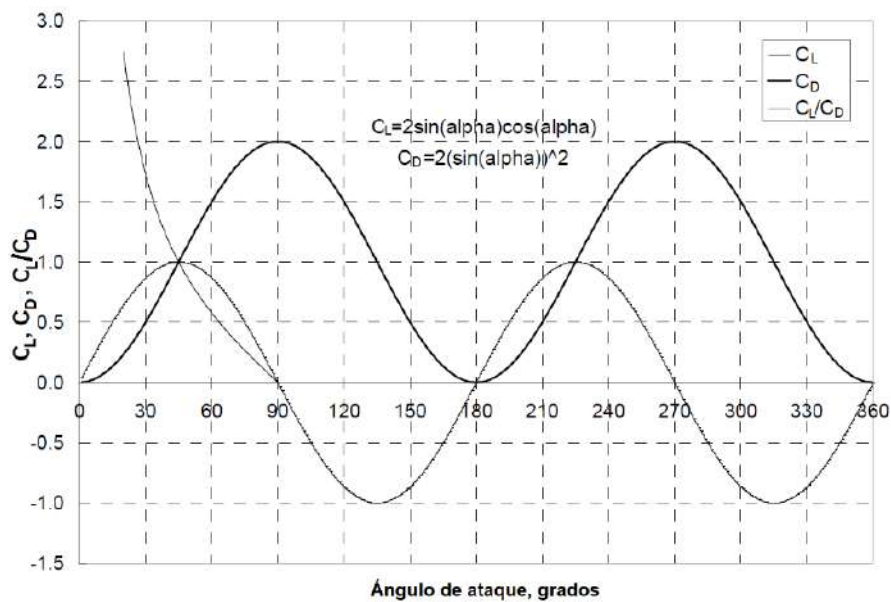


Figura 23. Variaciones del  $C_L$ ,  $C_D$  y  $C_L/C_D$  para la teoría de la placa plana (J. Tangler & Kocurek, 2005).

Para  $\alpha$  de  $20^\circ$ ,  $35^\circ$  y  $45^\circ$  la relación  $C_L/C_D$  tiene un valor ideal de 2.75, 1.73 y 1.0 respectivamente, también se observa que en ángulos de  $90^\circ$ , el  $C_L$  tiene un valor de 0. Esta teoría dicta el comportamiento aproximado de perfiles aerodinámicos en el régimen de estancamiento total en un rango de ángulos de ataque de  $30^\circ \leq \alpha \leq 90^\circ$ . Adicionalmente, las variaciones de los coeficientes en los regímenes de mayor sustentación y de

estancamiento total son altamente dependientes de la geometría de la pala y del perfil aerodinámico (J. Tangler & Kocurek, 2005).

## 2.2 FUNCIÓN DE TRANSFORMACIÓN DE CLASE-FORMA

La parametrización de perfiles aerodinámicos es parte del proceso de diseño y optimización de estos. Como se mencionó en los antecedentes, el método de parametrización mediante la función de Transformación de Clase-Forma, es un método constructivo ampliamente utilizado.

Existen dos tipos de clases en lo que respecta a los componentes de cualquier sistema aerodinámico: 1) las formas de tipo ala-perfil, y 2) las formas de tipo cuerpo de sección transversal. En el caso de los perfiles aerodinámicos para turbinas eólicas, estas se encuentran dentro de la clase 1 al ser geometrías de “nariz redondeada” y extremos de salida en punta que se describen matemáticamente como funciones continuas no analíticas, esto debido a su pendiente infinita en el borde de ataque y sus variaciones de curvatura sobre sus superficies. El fundamento de este método consiste en una función de transformación que simplifica la descripción matemática del perfil en una función analítica con variables de diseño fáciles de controlar, evitando las singularidades características de la sección redondeada de la geometría (Kulfan & Bussolletti, 2006).

Las variables de diseño necesarias que define el método CST para describir un perfil aerodinámico en los ejes de coordenadas X y Z se observan en la Figura 24. El radio del borde de ataque ( $R_{LE}$ ) afecta principalmente la capacidad del ángulo de ataque del perfil, mientras que la superficie inferior (o intradós) de este influye en la fuerza de arrastre, por otra parte, la superficie superior (o extradós) del perfil aerodinámico y el ángulo de cierre del borde de salida ( $\beta$ ) determinan la característica de separación de flujo en el perfil. La cuerda  $c$  es la longitud característica definida por la línea recta que conecta el borde de ataque con el borde de salida. El espesor del borde de salida ( $\Delta Z_{TE}$ ) es aquel que define la separación entre la cuerda y la última coordenada del borde de salida para cada superficie del perfil. El espesor máximo ( $T_{max}$ ) es la distancia máxima entre el extradós y el intradós medido perpendicularmente a la cuerda, mientras que ( $X_{max}$ ) representa la ubicación del espesor máximo sobre el eje X (Kulfan & Bussolletti, 2006).

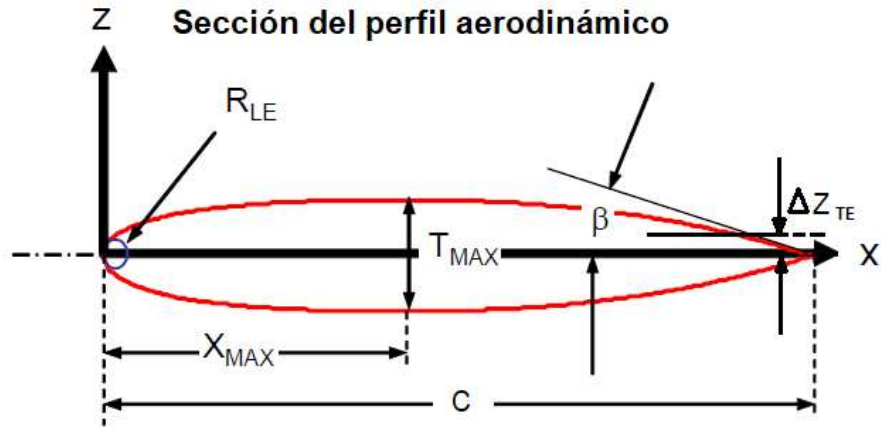


Figura 24. Variables de diseño claves de un perfil aerodinámico (Kulfan & Bussoletti, 2006).

De acuerdo con el método CST, la expresión matemática para definir las coordenadas de un perfil aerodinámico se observa en la ecuación (6). Note que en este análisis el subíndice *LE* significa borde de ataque y el subíndice *TE* significa borde de salida (Kulfan, 2008).

$$\zeta(\psi) = C_{N2}^{N1}(\psi) S(\psi) + \psi \frac{\Delta Z_{TE}}{c} \quad (6)$$

Donde:

$\psi = x/c$ , coordenada en la dirección x

$\zeta = z/c$ , coordenada en la dirección z

$C_{N2}^{N1}(\psi)$  = función de clase

$S(\psi)$  = función de forma

$\frac{\Delta Z_{TE}}{c}$  = término del espesor del borde de salida

### 2.2.1 Función de clase

La definición de la función de clase se observa en la ecuación (7).

$$C(\psi) = \psi^{N1}(1 - \psi)^{N2} \quad (7)$$

Donde:

$N1, N2$  son exponentes que definen la clase de la geometría del perfil aerodinámico.

El primer término  $\psi$  es el encargado de generar la nariz redondeada, mientras que el término  $(1 - \psi)$  es necesario para conservar el borde de salida. Distintas combinaciones de exponentes en la función de clase definen las clases de geometrías aerodinámicas fundamentales. Como se muestra en la Figura 25, las combinaciones a)  $N1 = 0.5$  y  $N2 = 1$  forman un típico perfil NACA con nariz redondeada y un acabado en punta, en b)  $N1 = 0.5$  y  $N2 = 0.5$  forman un perfil elíptico con nariz y acabado redondeados, en c)  $N1 = 1$  y  $N2 = 1$  forman un perfil biconvexo con nariz y acabado en punta, mientras que en d)  $N1 = 0.75$  y  $N2 = 0.75$  generan un cuerpo tipo Sears-Haack. Para perfiles aerodinámicos empleados en WT es común utilizar la clase 1 con los valores de  $N1 = 0.5$  y  $N2 = 1$  (Kulfan, 2008).

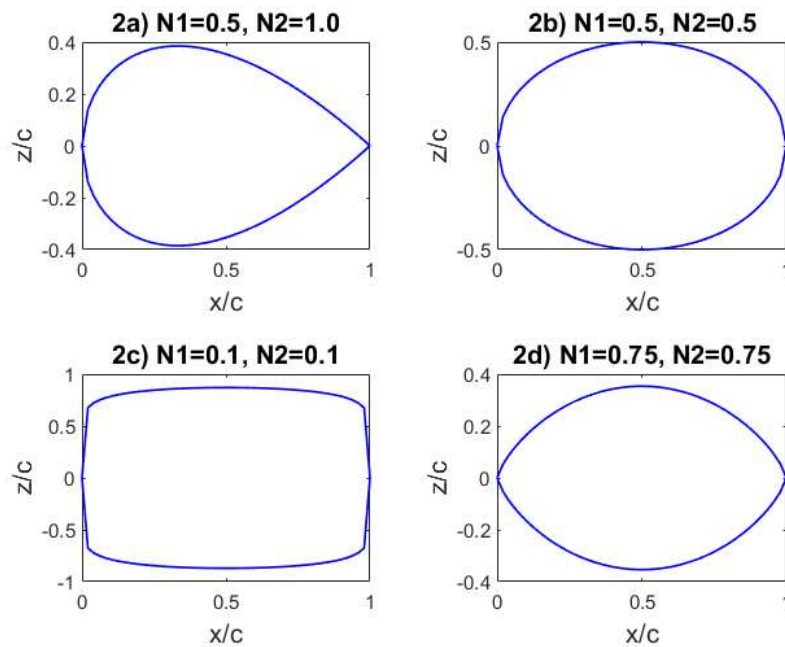


Figura 25. Geometrías de clases fundamentales.

### 2.2.2 Función de forma

La función de forma es aquella que describe la forma única de la geometría entre el borde de ataque y el borde de salida. De la ecuación (6), la función de forma se define en la ecuación (8). Con esta ecuación, es posible obtener la función de forma de un perfil aerodinámico específico conociendo sus coordenadas y su función de clase (Kulfan, 2008).

$$S(\psi) = \frac{\zeta(\psi) - \psi \frac{\Delta Z_{TE}}{c}}{\psi^{N1}(1 - \psi)^{N2}} \quad (8)$$

De igual manera, la función de forma  $S(\psi)$  se representa por conveniencia como una función polinomial como se muestra en la ecuación (9).

$$S(\psi) = \sum_{i=1}^{N+1} [A_i[\psi]^i] \quad (9)$$

Donde:

$A_i$  son los coeficientes de la función polinomial.

Se ha demostrado que las variables  $R_{LE}$ ,  $\beta$  y  $Z_{TE}$  están relacionadas con los coeficientes de frontera  $A_1$  y  $A_{N+1}$  de la función de forma (Kulfan, 2008), como se muestra en las ecuaciones (10) y (11).

$$S(0) = A_1 = \sqrt{2 \frac{R_{LE}}{c}} \quad (10)$$

$$S(1) = A_{N+1} = \tan\beta + \frac{\Delta Z_{TE}}{c} \quad (11)$$

Es importante mencionar que, para asegurar continuidad entre la superficie superior e inferior en la parte del borde de ataque del perfil, los coeficientes de la función de forma de cada superficie en ese límite deben ser iguales (Kulfan, 2008), para ello se igualan los coeficientes en la frontera como se muestra en la ecuación (12), donde los subíndices  $u$  y  $l$  indican la superficie superior (*upper*) e inferior (*lower*), respectivamente.

$$A_{u_{i=1}} = A_{l_{i=1}} \quad (12)$$

Donde:

$A_{u_{i=1}}$  es el primer coeficiente de  $S(\psi)$  para la superficie superior

$A_{l_{i=1}}$  es el primer coeficiente de  $S(\psi)$  para la superficie inferior



### 2.2.3 Representación en componentes de la función de forma

La función de forma unitaria, es decir  $S(\psi) = 1$ , no genera afectaciones a la geometría formada por la función de clase. Sin embargo, variaciones simples de los coeficientes de la función de forma permiten descomponerla para generar cambios en las variables  $R_{LE}$ ,  $\beta$  y  $Z_{TE}$  y tener control sobre estas. El método CST utiliza el polinomio de Bernstein para descomponer la función de forma unitaria (Kulfan, 2008), como se observa en la ecuación (13), siendo  $K_{r,N}$  el coeficiente binomial definido en la ecuación (14). Estas dos ecuaciones permiten variar los valores de los coeficientes polinomiales de la ecuación (9).

$$S_i(x) = BPn(x) = \sum_{r=0}^N [K_{r,N}(1-x)^{N-r}x^r] \quad (13)$$

$$K_{r,n} = \binom{N}{r} = \frac{N!}{r!(N-r)!} \quad (14)$$

Donde:

$BPn(x)$  = polinomio de Bernstein

$N$  = orden del polinomio de Bernstein

$r$  = índice de cero a  $N$

Una propiedad de los polinomios de Bernstein es que la suma de cada polinomio resulta en la unidad, debido a que se trata de polinomios ortogonales, por lo que es conveniente emplearlos para obtener la función de forma unitaria (Kulfan, 2008). Los polinomios de Bernstein de grado 3, 5 y 7, se observan respectivamente en las Figuras 26, 27 y 28, en ellas se aprecian las funciones de forma unitarias (línea roja) generadas por el polinomio de Bernstein y los componentes que forman la función de forma unitaria (líneas de colores), variando el valor de  $r$  desde 0 hasta  $N$ .

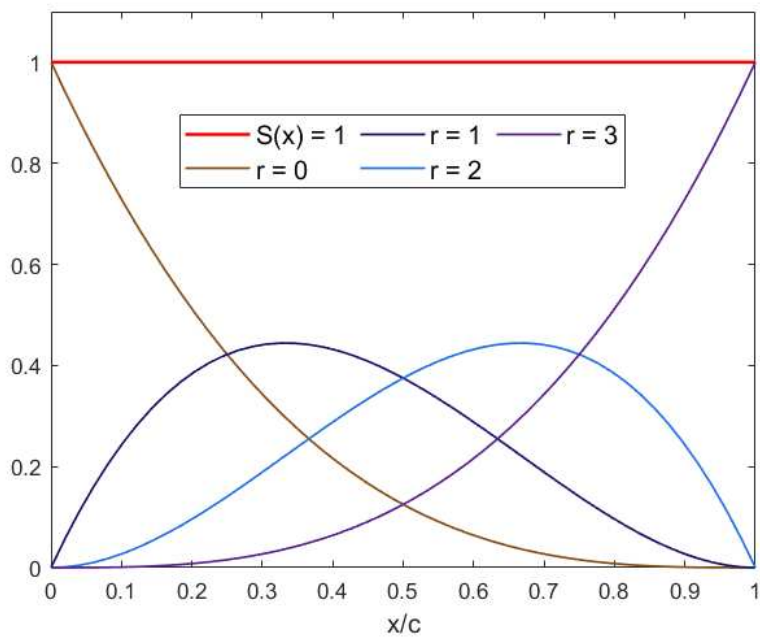


Figura 26. Componentes de la función de forma unitaria con  $BPn(x)$  de grado 3.

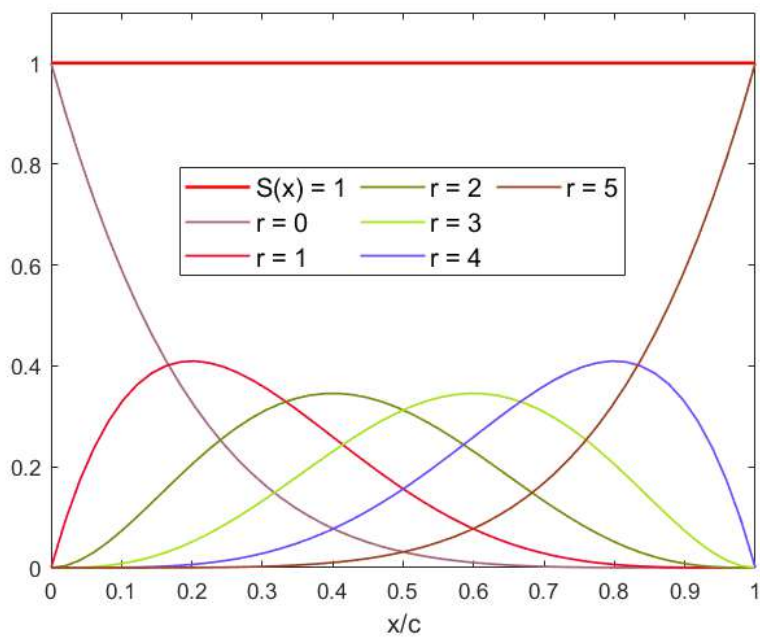


Figura 27. Componentes de la función de forma unitaria con  $BPn(x)$  de grado 5.

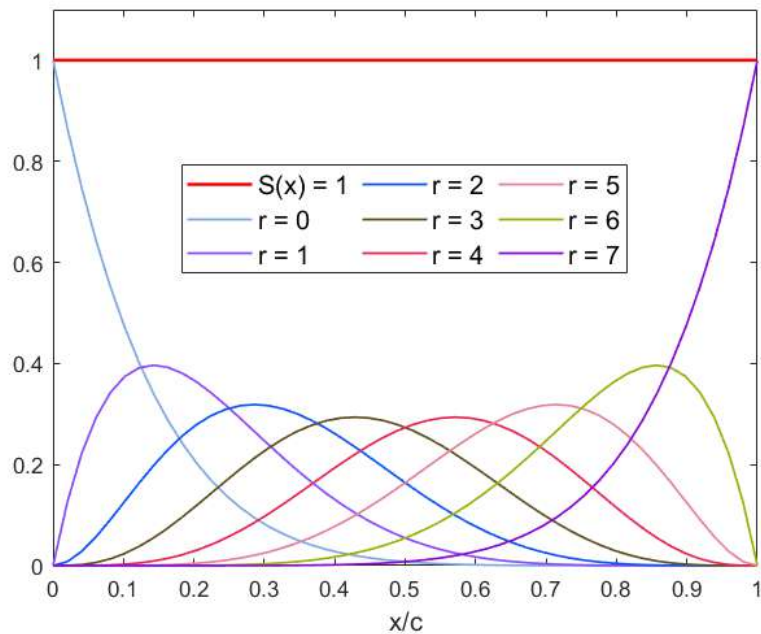


Figura 28. Componentes de la función de forma unitaria con  $BPn(x)$  de grado 7.

Conociendo el desarrollo del método CST, es posible construir la geometría de un perfil aerodinámico empleando una combinación de coeficientes del polinomio de la función de forma tanto para la superficie superior como para la superficie inferior.

## 2.3 OPTIMIZACIÓN Y ALGORITMO PSO

### 2.3.1 Optimización

Un problema de optimización consiste en encontrar el valor máximo o mínimo que satisface una función, la cual se conoce ampliamente como función objetivo. Cuando se trata de un problema de minimización, la OF se conoce como función de costo, por el contrario, se llama función de aptitud cuando se trata de un problema de maximización. La variable independiente, la cual es comúnmente el vector de valores de  $x$  del que depende la función  $f(x)$ , también se le conoce como variable de diseño (Simon, 2013).

Para aclarar los conceptos básicos de optimización, se considera el problema de minimización de la función  $f(x) = x^4 + 5x^3 + 4x^2 - 4x + 1$ , el cual, al ser una ecuación de cuarto grado, tiene tres puntos estacionarios, los cuales se resuelven al derivar  $f(x)$  e igualarla a cero (Simon, 2013). Por lo tanto, si  $f'(x) = 0$ , los valores de  $x$  que resuelven

esta ecuación son  $x_1 = -2.96$ ,  $x_2 = -1.10$  y  $x_3 = 0.31$ . En la Figura 29 se observa la definición gráfica de la función.

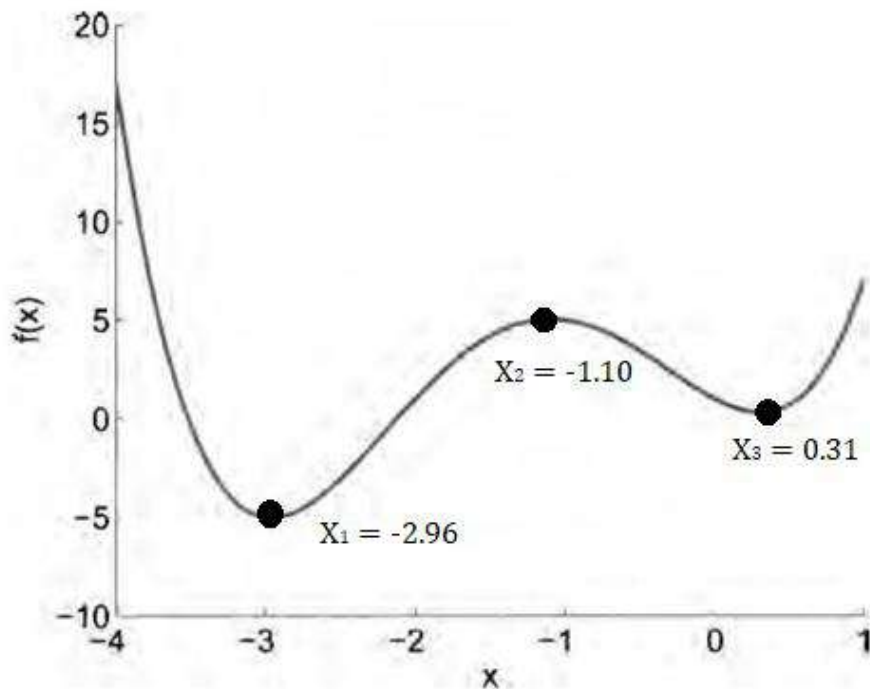


Figura 29. Definición gráfica de la función  $f(x)$  (Simon, 2013).

Los valores de  $x_1$  y  $x_3$  se conocen como mínimos locales mientras que  $x_2$  se conoce como el máximo local, además, el valor de  $x_1$  también se considera como el mínimo global dentro del dominio de  $-4 \leq x \leq 1$  debido a que es el valor mínimo en ese rango.

En la vida real, la forma gráfica de encontrar los valores óptimos de una función no es suficiente para optimizar un problema, para ello los EOA son métodos ampliamente utilizados con una gran capacidad de optimización, como lo es el algoritmo PSO.

### 2.3.2 Algoritmo PSO

El algoritmo PSO creado por Russell Eberhart y James Kennedy en 1995, es una de las variantes de los EOA la cual se inspira en el comportamiento de los enjambres de animales, como las parvadas o los cardúmenes, los cuales exhiben un comportamiento

colectivo inteligente, incluso se considera que el algoritmo PSO imita también el comportamiento colectivo de los humanos (Simon, 2013).

Una de las ventajas significativas del algoritmo PSO es su capacidad de lidiar con problemas complejos y de alta dimensionalidad. Su naturaleza basada en poblaciones y su enfoque en la exploración global permite encontrar soluciones prometedoras incluso en grandes espacios de búsqueda (Simon, 2013). Además, el PSO es relativamente fácil de implementar y de ajustar, lo cual lo hace atractivo para su aplicación en diversos dominios, como la optimización de funciones matemáticas, el diseño de redes neuronales, la planificación de rutas, la optimización de sistemas de energía, la optimización de elementos mecánicos, entre otros (Ramírez-Ochoa et al., 2022).

En PSO, una población de partículas (el enjambre) se mueve a través de un espacio de búsqueda multidimensional (el espacio de diseño) dado que la posición de cada partícula está en función de las variables de diseño, y cada una representa una posible solución al problema. Dichas partículas actualizan su velocidad y posición en cada iteración (periodo) considerando sus posiciones pasadas y presentes, de tal manera que cada partícula es influenciada por todo el enjambre hasta encontrar la mejor posición global, la cual representa la solución óptima de la OF, ya sea encontrando su máximo o su mínimo (Ramírez-Ochoa et al., 2022).

El movimiento de cada partícula de la población está determinado por dos componentes principales: 1) la mejor posición del enjambre conocida en todos los periodos (*PBest*), y 2) la mejor posición en el enjambre encontrada por cualquier partícula en el periodo actual (*GBest*). En la Figura 30 se observa un esquema del movimiento de una partícula en el algoritmo PSO. Cada partícula ajusta su velocidad y posición en función de estas dos influencias, lo que les permite explorar y explotar el espacio de búsqueda de manera eficiente. La búsqueda se realiza mediante iteraciones sucesivas y el algoritmo converge hacia la solución óptima a medida que avanza (Ramírez-Ochoa et al., 2022).

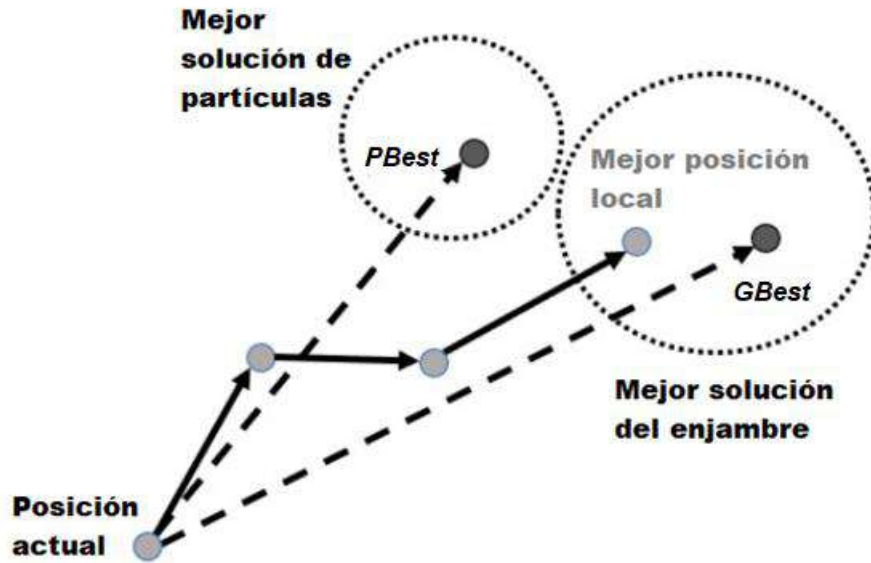


Figura 30. Movimiento de una partícula en PSO (Ramírez-Ochoa et al., 2022).

Con base en la literatura referente al algoritmo PSO, se sabe que uno de sus principales problemas es que su solución tiende a caer en máximos/mínimos locales dentro del espacio de búsqueda debido a una baja diversidad en las partículas de la población, lo cual influye en las características de exploración y explotación del algoritmo, siendo estas la capacidad de buscar nuevas estrategias y usar estrategias existentes en periodos pasados para encontrar la solución, respectivamente (Simon, 2013).

Para combatir esto, Xin et al. (2009) proponen un coeficiente de peso de inercia multietapa y linealmente decreciente que modifica la nueva velocidad de las partículas en cada iteración, con la finalidad de incrementar la diversidad en las partículas de la población y aumentar la exploración en el espacio de diseño del algoritmo. El cálculo del coeficiente de inercia empleado en este estudio se muestra en la ecuación (15).

$$\omega(t) = \omega_{max} - t \left( \frac{\omega_{max} - \omega_{min}}{T} \right) \quad (15)$$

Donde:

$\omega$  = peso de inercia

$\omega_{max}$  = valor máximo del peso de inercia

$\omega_{min}$  = valor mínimo del peso de inercia

$t$  = iteración actual

$T$  = número máximo de iteraciones

De esta manera, se establecen los valores máximos y mínimos del coeficiente  $\omega$ , de acuerdo con el segundo término del lado derecho en la ecuación (15) el valor del peso de inercia disminuye linealmente con el periodo de actualización del enjambre. Cabe mencionar que se han propuesto otras modificaciones del coeficiente de peso de inercia para el algoritmo PSO, para lo que se recomienda estudiar a detalle el trabajo realizado por Fan & Akhter (2021).

El proceso del algoritmo PSO se puede dividir en tres fases principales: 1) inicialización de la población, 2) actualización de la posición del  $PBest$ , y 3) la obtención de la posición del  $GBest$  (Ramírez-Ochoa et al., 2022). La inicialización consiste en inicializar los valores de los parámetros de control, entre los que se encuentran el valor máximo ( $\omega_{max}$ ) y mínimo ( $\omega_{min}$ ) del peso de inercia, los factores de aprendizaje  $c_1$  y  $c_2$ , los factores de aleatoriedad  $e_1$  y  $e_2$ , el número máximo de iteraciones o periodos ( $T$ ), y los vectores iniciales de posición ( $x_i$ ) y velocidad ( $v_i$ ) de las partículas.

En cada iteración, la velocidad y posición de las partículas se actualizan de acuerdo con las ecuaciones (16) y (17) respectivamente, después se evalúa la función objetivo y se obtienen las posiciones  $PBest$  y  $GBest$ . Finalmente, la posición del  $GBest$  óptimo se obtiene realizando este proceso hasta que se cumpla la condición de convergencia, la cual suele ser un número máximo de iteraciones establecido (e.g. mientras  $t \leq T$ , para  $T = 1000$ ), en donde  $t$  es la iteración actual y  $T$  es el número máximo de iteraciones.

$$v_i^{t+1} = \omega \cdot v_i^t + c_1 \cdot e_1(Gbest_i^t - x_i^t) + c_2 \cdot e_2(Pbest_i^t - x_i^t) \quad (16)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (17)$$

Donde:

$t$  = iteración actual

$i$  = número de partícula dentro de la población

$\omega$  = coeficiente del peso de inercia

$c_1$  = factor de aprendizaje social

$c_2$  = factor de aprendizaje cognitivo

$e_1, e_2$  = factores de aleatoriedad entre 0 y 1

$Gbest_i$  = mejor posición del enjambre en el periodo dada por cualquier partícula

$Pbest_i$  = mejor posición de cada partícula del enjambre en todos los periodos

En la Figura 31 se muestra el diagrama de flujo del algoritmo PSO empleado, mientras que en el ANEXO 1 se encuentra el pseudocódigo utilizado para desarrollar el algoritmo.

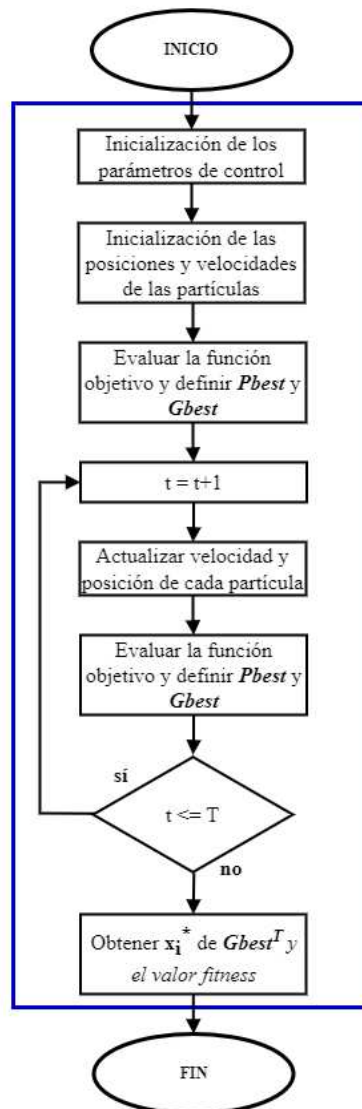




Figura 31. Diagrama de flujo del algoritmo PSO.

En el pseudocódigo, el subíndice  $d$  en el vector de variables independientes  $\mathbf{x}$  indica el número de variables de diseño de las que depende la función objetivo, mientras que el superíndice  $*$  en el vector de variables independientes  $\mathbf{x}$  indica la solución óptima.

## 2.4 DINÁMICA DE FLUIDOS COMPUTACIONAL

En la siguiente sección se muestran los fundamentos de la teoría de volúmenes de control, las leyes de conservación de la mecánica de fluidos y los modelos de turbulencia empleados en el análisis numérico como base para el entendimiento de los métodos CFD.

### 2.4.1 Teoría de Volúmenes de Control

El movimiento de un fluido puede analizarse desde dos enfoques: 1) la descripción lagrangiana o 2) la descripción euleriana. El primero se realiza mediante una descripción detallada del flujo en cada punto  $(x, y, z)$  del campo del fluido, mientras que el segundo se realiza mediante el balance entre el fluido que entra y sale de una región finita del espacio para determinar los efectos netos sobre este (White, 2004). El segundo enfoque es la base de la teoría de volúmenes de control, siendo el volumen de control (VC) la región finita sujeta al análisis.

En la descripción euleriana no hay necesidad de seguir el rastro de posición y velocidad de cada partícula de fluido, en su lugar, se definen variables de campo, las cuales son funciones del espacio y del tiempo dentro del VC, y pueden ser escalares o vectoriales (Cengel & Cimbala, 2006). Por mencionar algunas, en las ecuaciones (18), (19) y (20) se definen los campos de presión, velocidad y aceleración para un flujo tridimensional no estacionario en coordenadas cartesianas, respectivamente, siendo el campo de presión una variable escalar, mientras que las otras dos son variables vectoriales. En la Figura 32 se muestra la representación de un VC con algunas de sus variables de campo.

$$P = P(x, y, z, t) \quad (18)$$

$$\vec{V} = \vec{V}(x, y, z, t) \quad (19)$$

$$\vec{a} = \vec{a}(x, y, z, t) \quad (20)$$

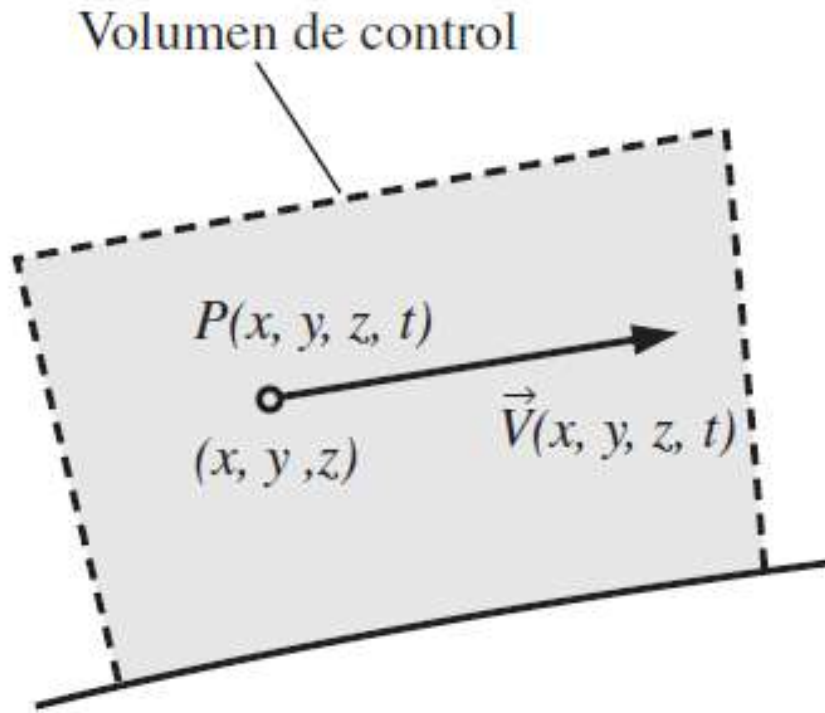


Figura 32. Volumen de control con sus variables de campo de presión y velocidad (Cengel & Cimbala, 2006).

La parte nuclear de la teoría de volúmenes de control es que no importa lo que suceda a las partículas del fluido por separado, sino que se centra la atención en las variables de campo de cualquiera que sea la partícula de fluido en el VC, por lo que todas las variables se definen en cualquier ubicación  $(x, y, z)$  y en cualquier instante  $t$  (Cengel & Cimbala, 2006).

#### 2.4.1.1 Teorema del Transporte de Reynolds

Un VC se considera como un sistema abierto en donde existe flujo de masa a través de sus límites, conocidos como superficies de control (SC), además, un VC también puede moverse y deformarse durante un proceso. La mayoría de los principios de la mecánica de fluidos se refieren a las razones de cambio respecto del tiempo de las propiedades de los sistemas, sin embargo, con frecuencia conviene trabajar con VC; el Teorema de Transporte de Reynolds (TTR) proporciona la relación entre las razones de

cambio respecto del tiempo de las propiedades de un sistema con las razones de cambio de un volumen de control (Cengel & Cimbala, 2006).

Para entender el significado fundamental del TTR, se analiza la configuración geométrica sencilla del sistema mostrado en la Figura 33. En la geometría, la dirección del flujo del fluido está indicada por  $V_1$ , mientras que el VC es el volumen fijo entre las SC (1) y (2), las cuales son normales a la dirección del flujo. En el instante  $t$ , el sistema coincide con el VC (región sombreada gris), sin embargo, en el instante  $t + \Delta t$ , el sistema se mueve en la dirección del flujo, siendo el mismo fluido, pero ocupando la región  $VC - I + II$ , lo cual representa el volumen de control menos el flujo de entrada durante  $\Delta t$  más el flujo de salida durante  $\Delta t$  (Cengel & Cimbala, 2006).

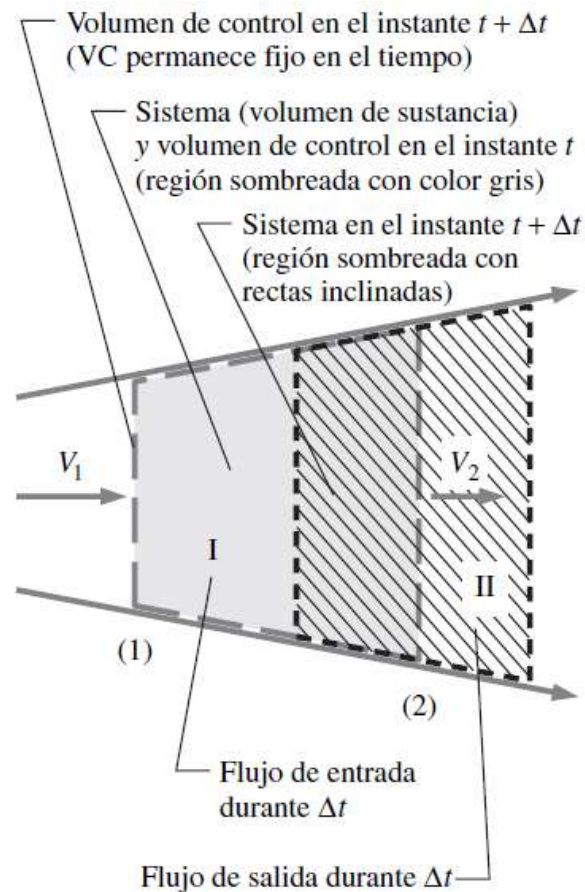


Figura 33. Geometría sencilla de un sistema de fluido (Cengel & Cimbala, 2006).

Ahora, si se considera cualquier propiedad extensiva del fluido (e.g. masa, cantidad de movimiento, energía) como  $B$ , la explicación del movimiento del fluido anterior se define en las ecuaciones (21) y (22).

$$B_{sist} = B_{VC}, \text{ en el instante } t \quad (21)$$

$$B_{sist} = B_{VC} - B_I + B_{II}, \text{ en el instante } t + \Delta t \quad (22)$$

Al restar la ecuación (21) de (22), dividir entre  $\Delta t$ , y emplear la definición de derivada se obtiene la ecuación (23), o bien, la ecuación (24) si se consideran las propiedades intensivas, es decir  $b = B/m$ .

$$\frac{dB_{sist}}{dt} = \frac{dB_{VC}}{dt} - \dot{B}_{ent} + \dot{B}_{sal} \quad (23)$$

$$\frac{dB_{sist}}{dt} = \frac{dB_{VC}}{dt} - b_1 \rho_1 V_1 A_1 + b_2 \rho_2 V_2 A_2 \quad (24)$$

Donde:

$$\frac{dB_{sist}}{dt} = \text{razón de cambio de } B \text{ en el sistema}$$

$$\frac{dB_{VC}}{dt} = \text{razón de cambio de } B \text{ en el VC}$$

$$\dot{B}_{ent} = \text{flujo de } B \text{ a la entrada}$$

$$\dot{B}_{sal} = \text{flujo de } B \text{ a la salida}$$

$$A_1, A_2 = \text{áreas de las secciones transversales en las superficies de control 1 y 2}$$

Para generalizar el proceso, se considera un área diferencial  $dA$  sobre la SC, siendo  $\vec{n}$  su vector normal exterior unitario. Debido a que el producto punto  $\vec{V} \cdot \vec{n}$  es la componente normal de la velocidad, la razón neta del flujo de salida a través de la SC se muestra en la ecuación (25), mientras que la cantidad total de  $B$  dentro del VC se muestra en la ecuación (26). Por lo tanto, el TTR para un volumen de control fijo se observa en la ecuación (27).

$$\dot{B}_{neta} = \dot{B}_{sal} - \dot{B}_{ent} = \int_{SC} \rho b \vec{V} \cdot \vec{n} dA \quad (25)$$

$$B_{VC} = \int_{VC} \rho b dV \quad (26)$$

$$\frac{dB_{sist}}{dt} = \frac{d}{dt} \int_{VC} \rho b dV + \int_{SC} \rho b \vec{V} \cdot \vec{n} dA \quad (27)$$

La ecuación (27) del TTR varía dependiendo si se trata de un VC fijo, móvil o deformable, sin embargo, en este estudio solo es necesario demostrar el TTR para volúmenes de control fijos.

#### 2.4.2 Leyes de conservación de la mecánica de fluidos

Las ecuaciones fundamentales de la mecánica de fluidos son las ecuaciones de conservación de la masa, conservación de la cantidad de movimiento y conservación de la energía, las cuales se analizan desde el enfoque de volúmenes de control y hacen referencia a que cualquier cantidad conservada se debe balancear durante un proceso (Cengel & Cimbala, 2006), sin embargo, debido al alcance de este proyecto y la fenomenología del problema a resolver se estudiarán solamente las primeras dos.

##### 2.4.2.1 Ley de conservación de la masa

Para un VC, el balance de masa se define en forma de razón como se muestra en la ecuación (28). Por otra parte, la ecuación de conservación de la masa también suele llamarse ecuación de continuidad.

$$\dot{m}_{ent} - \dot{m}_{sal} = \frac{dm_{VC}}{dt} \quad (28)$$

Donde:

$\frac{dm_{VC}}{dt}$  = razón de cambio de masa en las fronteras del VC

$\dot{m}_{ent}$  = razón total del flujo de masa que entra al VC

$\dot{m}_{sal}$  = razón total del flujo de masa que sale del VC

##### 2.4.2.2 Ley de conservación de la cantidad de movimiento

El producto de la masa y la velocidad de un sistema se llama cantidad de movimiento, y se define como una masa  $m$  que avanza a una velocidad  $\vec{V}$ . Sabiendo esto, la ecuación de conservación de la cantidad de movimiento se obtiene aplicando la Segunda

Ley de Newton a un fluido a través de un volumen de control, como se observa en la ecuación (29).

$$\vec{F} = m\vec{a} = m \frac{d\vec{V}}{dt} = \frac{d(m\vec{V})}{dt} \quad (29)$$

Donde:

$m$  = masa del sistema o del VC

$\vec{V}$  = velocidad del sistema o del fluido que fluye por el VC

### 2.4.3 Ecuaciones de movimiento de la dinámica de fluidos computacional

Una vez comprendida la teoría de volúmenes de control y todas sus implicaciones, la dinámica de fluidos computacional consiste en aplicar las leyes de conservación en un volumen de control infinitesimal para obtener ecuaciones diferenciales que describen el movimiento del fluido, como se observa en la Figura 34, las cuales son resueltas mediante análisis numérico (White, 2004).

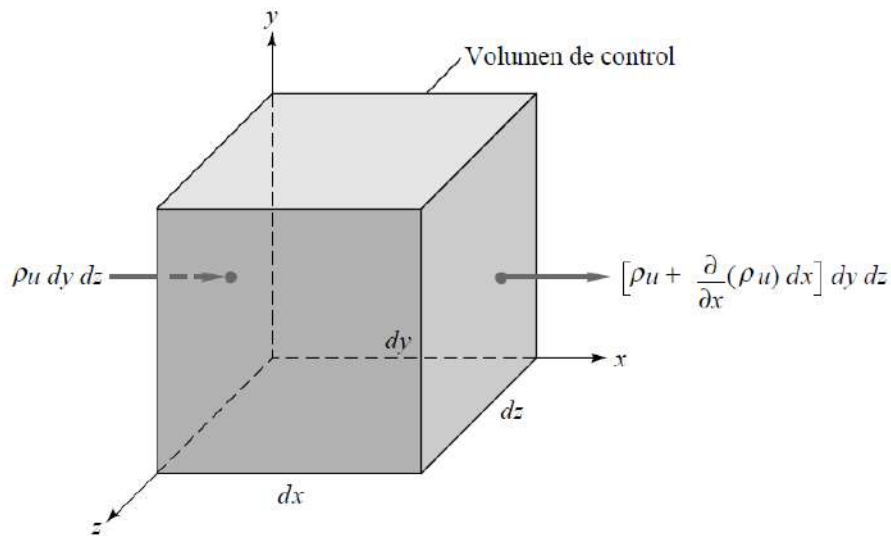


Figura 34. VC infinitesimal fijo en coordenadas cartesianas mostrando los flujos másicos  $\dot{m}_{ent}$  y  $\dot{m}_{sal}$  en las caras perpendiculares al eje x (White, 2004).

En las ecuaciones (30) y (31) se observan la ecuación de continuidad y la ecuación de Navier-Stokes para un flujo laminar estacionario de un fluido viscoso, newtoniano e

incompresible, respectivamente. La primera representa la ecuación de conservación de la masa, mientras que la segunda es una ecuación de transporte que representa el transporte de la cantidad de movimiento lineal en el dominio computacional (Cengel & Cimbala, 2006).

$$\nabla \cdot \vec{V} = 0 \quad (30)$$

$$(\vec{V} \cdot \nabla) \vec{V} = -\frac{1}{\rho} \nabla P + \nu \nabla^2 \vec{V} \quad (31)$$

Donde:

$\vec{V}$  = velocidad del fluido

$\rho$  = densidad del fluido

$\nu = \mu/\rho$  = viscosidad cinética del fluido

P = presión del fluido

#### 2.4.4 Ecuaciones de movimiento en ANSYS Fluent

Para todos los tipos de flujo de fluidos, ANSYS Fluent resuelve las ecuaciones de conservación de la masa (continuidad) y de conservación de la cantidad de movimiento (Navier-Stokes). Por otra parte, cuando se considera un flujo turbulento, adicionalmente se resuelven ecuaciones de transporte definidas en modelos de turbulencia (ANSYS Inc., 2021a).

##### 2.4.4.1 Ecuación de conservación de la masa

En la ecuación (32) se muestra la ecuación de conservación de la masa tanto para flujos compresibles como incompresibles que emplea ANSYS Fluent, mientras que en la ecuación (33) se muestra para geometrías axial-simétricas en 2D (ANSYS Inc., 2021a).

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{V}) = S_m \quad (32)$$

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho V_x) + \frac{\partial}{\partial r}(\rho V_r) + \frac{\rho V_r}{r} = S_m \quad (33)$$

Donde:

$S_m$  = término de masa añadida a la fase continua

$x$  = coordenada axial

$r$  = coordenada radial

$V_x$  = velocidad axial

$V_r$  = velocidad radial

#### 2.4.4.2 Ecuación de conservación de la cantidad de movimiento

En la ecuación (34) se observa la ecuación de conservación de la cantidad de movimiento para un marco de referencia del fluido inerte (ANSYS Inc., 2021a).

$$\frac{\partial}{\partial t}(\rho \vec{V}) + \nabla \cdot (\rho \vec{V} \vec{V}) = -\nabla p + \nabla \cdot (\bar{\tau}) + \rho \vec{g} + \vec{F} \quad (34)$$

Donde:

$p$  = presión estática

$\bar{\tau}$  = tensor de esfuerzos

$\rho \vec{g}$  = fuerza gravitacional sobre el fluido

$\vec{F}$  = fuerzas externas

Adicionalmente, el tensor de esfuerzos se define en la ecuación (35).

$$\bar{\tau} = \mu[(\nabla \vec{V} + \nabla \vec{V}^T) - \frac{2}{3} \nabla \cdot \vec{V} I] \quad (35)$$

Donde:

$\mu$  = viscosidad molecular

$I$  = tensor unitario

$\frac{2}{3} \nabla \cdot \vec{V} I$  = dilatación volumétrica

Por otra parte, para geometrías axial-simétricas en 2D, las ecuaciones de conservación de la cantidad de movimiento axial y radial se muestran en (36) y (37), respectivamente (ANSYS Inc., 2021a).



$$\begin{aligned}
& \frac{\partial}{\partial t}(\rho V_x) + \frac{1}{r} \frac{\partial}{\partial x}(r \rho V_x V_x) + \frac{1}{r} \frac{\partial}{\partial r}(r \rho V_r V_x) \\
& = -\frac{\partial p}{\partial x} + \frac{1}{r} \frac{\partial}{\partial x} \left[ r \mu \left( 2 \frac{\partial V_x}{\partial x} - \frac{2}{3} (\nabla \cdot \vec{V}) \right) \right] + \frac{1}{r} \frac{\partial}{\partial r} \left[ r \mu \left( \frac{\partial V_x}{\partial r} + \frac{\partial V_r}{\partial x} \right) \right] \\
& + F_x
\end{aligned} \quad (36)$$

$$\begin{aligned}
& \frac{\partial}{\partial t}(\rho V_r) + \frac{1}{r} \frac{\partial}{\partial x}(r \rho V_x V_r) + \frac{1}{r} \frac{\partial}{\partial r}(r \rho V_r V_r) \\
& = -\frac{\partial p}{\partial r} + \frac{1}{r} \frac{\partial}{\partial x} \left[ r \mu \left( \frac{\partial V_r}{\partial x} + \frac{\partial V_x}{\partial r} \right) \right] + \frac{1}{r} \frac{\partial}{\partial r} \left[ r \mu \left( 2 \frac{\partial V_r}{\partial r} - \frac{2}{3} (\nabla \cdot \vec{V}) \right) \right] \\
& - 2\mu \frac{V_r}{r^2} + \frac{2}{3} \frac{\mu}{r} (\nabla \cdot \vec{V}) + \rho \frac{V_z^2}{r} + F_r
\end{aligned} \quad (37)$$

Donde:

$$\nabla \cdot \vec{V} = \frac{\partial V_x}{\partial x} + \frac{\partial V_r}{\partial r} + \frac{V_r}{r}$$

$V_z$  = velocidad de remolino

#### 2.4.5 Modelado de turbulencia en ANSYS Fluent

Cuando se habla de modelar los efectos de turbulencia en el flujo de un fluido, las variables de solución en la ecuación de Navier-Stokes (cantidad de movimiento) se descomponen en componentes fluctuantes y promediadas en el tiempo (ANSYS Inc., 2021a). En la ecuación (38) se observan los componentes de velocidad.

$$u_i = \bar{u}_i + u'_i \quad (38)$$

Donde:

$\bar{u}_i$  = componente media de la velocidad

$u'_i$  = componente fluctuante de la velocidad

Al substituir esta expresión en las variables de flujo de la ecuación de Navier-Stokes, se obtienen las ecuaciones de Navier-Stokes promediadas en el tiempo (RANS). En las ecuaciones (39) y (40) se muestran las ecuaciones RANS en forma de tensores cartesianos, en donde los términos adicionales que aparecen representan los efectos de turbulencia, los cuales deben ser modelados.

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i} (\rho u_i) = 0 \quad (39)$$

$$\begin{aligned} \frac{\partial}{\partial t} (\rho u_i) + \frac{\partial}{\partial x_j} (\rho u_i u_j) \\ = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j} \left[ \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \delta_{ij} \frac{\partial u_l}{\partial x_l} \right) \right] + \frac{\partial}{\partial x_j} (-\rho \overline{u'_i u'_j}) \end{aligned} \quad (40)$$

Donde:

$-\rho \overline{u'_i u'_j}$  = esfuerzo de Reynolds

#### 2.4.5.1 Modelo de turbulencia SST k- $\omega$

El modelo de turbulencia k- $\omega$  desarrollado por David Wilcox considera las ecuaciones de transporte de la turbulencia del fluido para flujos con números de Reynolds pequeños, compresibles y con efectos de propagación del flujo cortante. Este modelo presenta dos ecuaciones de transporte, una para la energía cinética de turbulencia ( $k$ ) y la otra para la razón de disipación específica ( $\omega$ ). Adicionalmente, el modelo k- $\omega$  ha sido modificado a través de los años con la finalidad de ser más adecuado para análisis de fluidos específicos (ANSYS Inc., 2021a).

El modelo de turbulencia *Shear-Stress Transport* (SST) k- $\omega$  desarrollado por Menter presenta mejoras de independencia del flujo libre en regiones del fluido cercanas a las fronteras de superficie, y además tiene en cuenta el transporte del esfuerzo cortante de turbulencia en la definición de la viscosidad turbulenta, lo cual lo hace un modelo adecuado para una clase más variada de flujos (*e.g.* ondas de choque transónicas, perfiles aerodinámicos) (ANSYS Inc., 2021a).

En las ecuaciones (41) y (42) se muestran las ecuaciones de transporte del modelo SST k- $\omega$ .

$$\frac{\partial}{\partial t} (\rho k) + \frac{\partial}{\partial x_i} (\rho k u_i) = \frac{\partial}{\partial x_j} \left( \Gamma_k \frac{\partial k}{\partial x_j} \right) + G_k - Y_k + S_k + G_b \quad (41)$$

$$\frac{\partial}{\partial t} (\rho \omega) + \frac{\partial}{\partial x_i} (\rho \omega u_i) = \frac{\partial}{\partial x_j} \left( \Gamma_\omega \frac{\partial \omega}{\partial x_j} \right) + G_\omega - Y_\omega + D_\omega + S_\omega + G_{\omega b} \quad (42)$$

En las ecuaciones anteriores, el término  $G$  representa la generación,  $\Gamma$  representa la difusividad efectiva,  $Y$  representa la disipación,  $D$  es el término de difusión cruzada,  $S$  es

el término fuente definida por el usuario, y  $G_b$  es el término producido por la flotabilidad, donde los subíndices  $k$  y  $\omega$  indican que son para la energía cinética de turbulencia y para la razón de disipación específica, respectivamente. Para conocer el desarrollo del cálculo de los términos mostrados en las ecuaciones (41) y (42) se recomienda revisar a detalle la guía teórica de ANSYS Fluent (ANSYS Inc., 2021a).

## **2.5 SOFTWARES EMPLEADOS EN EL MARCO DE TRABAJO**

En esta sección se presentan las generalidades de los programas MATLAB, ANSYS Workbench, y los módulos de ANSYS empleados.

### **2.5.1 MATLAB**

MATLAB es un sistema de cómputo numérico cuyo nombre deriva de laboratorio de matrices (*Matrix Laboratory*, MATLAB), precisamente porque su lenguaje de programación basado en matrices, el lenguaje M, permite la visualización de datos y la obtención de información a través de estos dentro del ámbito científico y de la ingeniería. Además, puede ejecutar sus análisis en conjuntos de datos de mayor tamaño y expandirse a *clusters* y nubes. El código de MATLAB se puede integrar con otros lenguajes, lo que le permite desplegar algoritmos y aplicaciones en sistemas web, empresariales o de producción (The MathWorks Inc., 2023).

### **2.5.2 ANSYS Workbench**

ANSYS Workbench es un software en el cual se pueden trabajar proyectos de análisis numérico (*e.g.* CFD, FEA) dentro de un mismo espacio de trabajo. El flujo de trabajo de los proyectos se puede visualizar de manera gráfica a través del esquema de proyecto, el cual se genera mediante sistemas para los distintos tipos de análisis. Estos sistemas contienen componentes que representan los pasos secuenciales para realizar el análisis, y también contienen los softwares internos o módulos para completar los procesos del análisis numérico, como la generación del modelo computacional y del mallado del modelo (ANSYS Inc., 2013).

ANSYS Workbench tiene la habilidad de grabar las acciones que el usuario realiza mediante la interfaz gráfica de usuario (*Graphical User Interface*, GUI), lo que se refiere al concepto de *journaling*. Los *journal files* son *scripts* en Python, y pueden ser creados o modificados, lo que se refiere al concepto de *scripting*. Los *journal files* son una herramienta poderosa de ANSYS Workbench ya que permiten realizar análisis de manera automatizada (ANSYS Inc., 2013).

Para grabar un *journal file*, es necesario seguir los siguientes pasos en la GUI del esquema de proyecto: 1) seleccionar *file > Scripting > Record Journal*, 2) especificar nombre y ubicación del archivo y seleccionar *Save*, 3) Usar la GUI para realizar el análisis, y 4) seleccionar *file > Scripting > Stop Recording Journal*. Por otra parte, la creación de *scripts* requiere conocimientos generales de la programación orientada a objetos (*Object-Oriented Programming*, OOP) (ANSYS Inc., 2013).

Una característica importante es que ANSYS Workbench puede ser ejecutado desde la línea de comandos del sistema operativo, lo cual facilita la automatización de *scripts*. Por ejemplo, para ejecutar ANSYS Workbench desde la ubicación de instalación predeterminada de un sistema operativo Windows de 64 bits se emplea la línea de comando **C:\Program Files\ANSYS Inc\V232\Framework\bin\win64\runwb2** (ANSYS Inc., 2023a). En la Tabla 3 se muestran los argumentos que acepta la línea de comando para controlar las operaciones de archivos y la manera en que se ejecuta ANSYS Workbench.

Tabla 3. Argumentos de la línea de comando para ejecutar ANSYS Workbench (ANSYS Inc., 2023a).

Argumento	Operación
<i>-B</i>	Ejecuta Ansys Workbench en modo <i>Batch</i> . Este modo no muestra la GUI y abre una ventana de consola. La funcionalidad de la ventana de consola es la misma que la de la ventana de comandos de Ansys Workbench.
<i>-R WorkbenchScriptFile</i>	Reproduce el <i>script</i> de Ansys Workbench especificado al iniciarse. Si se especifica junto con <i>-B</i> , Ansys Workbench se iniciará en modo <i>Batch</i> , ejecutará el <i>script</i> y se cerrará al finalizar su ejecución.
<i>-I</i>	Ejecuta Ansys Workbench en modo interactivo (modo predeterminado), pero si se especifica junto con <i>-B</i> , se abren tanto la GUI como la ventana de la consola.
<i>-X</i>	Ejecuta Ansys Workbench en modo interactivo y luego sale al finalizar la ejecución del <i>script</i> .

	Normalmente se utiliza junto con <i>-R</i> .
<i>-F WorkbenchProjectFile</i>	Carga el archivo de proyecto de Ansys Workbench especificado en el arranque.
<i>-E command</i>	Ejecuta el comando de <i>scripting</i> de Ansys Workbench especificado al iniciarse. Puede emitir varios comandos, separados por punto y coma (;), o especificar este argumento varias veces y los comandos se ejecutarán en orden.

Finalmente, existen funciones que se pueden ejecutar en los *journal files* de ANSYS Workbench para leer los *scripts* en Python para la generación automatizada de la geometría y el mallado del modelo, para lo cual se recomienda revisar la guía de *scripting* de ANSYS Workbench (ANSYS Inc., 2023a).

#### 2.5.2.1 SpaceClaim

*SpaceClaim* es el software que utiliza ANSYS Workbench en su sistema de creación de la geometría. Este software es bastante robusto para la generación y manipulación rápida de modelos computacionales en 2D y 3D.

La característica de *SpaceClaim* más importante que se utilizará en este trabajo es su función de grabación de *scripts*, la cual se emplea mediante su herramienta *Script Recording* y es bastante útil para la automatización en la generación de geometrías. Los *scripts* de *SpaceClaim* para grabar las operaciones se generan en lenguaje Python y se trabajan mediante el editor de *scripts* integrado. Para más detalles de las capacidades de *scripting* de este software se recomienda consultar la guía de usuario de *SpaceClaim* (ANSYS Inc., 2023b).

#### 2.5.2.2 Meshing

*Meshing* es el software que utiliza ANSYS Workbench en su sistema de creación de la malla para discretizar la geometría del modelo computacional del análisis. Este software proporciona herramientas de mallado robustas y fáciles de usar que simplifica el proceso de generación de mallas, además de que tienen la ventaja de estar altamente automatizadas junto con un grado moderado a alto de control por parte del usuario. De manera similar, *Meshing* tiene la capacidad de grabar las operaciones de mallado mediante *scripts* en Python para automatizar este proceso. Para más información de las capacidades

de *scripting* de este software se recomienda consultar la guía de usuario de *Meshing* (ANSYS Inc., 2021b).

### 2.5.2.3 Fluent

En los primeros años de la década de 1980, ANSYS Fluent se convirtió en el primer software comercial de CFD en emplear una GUI en su flujo de trabajo más que solo una entrada de línea de comandos, lo que lo ha caracterizado como uno de los softwares comerciales para análisis de fluidodinámica más populares en la industria. A lo largo de su historia, el software ha pasado por diferentes puntos críticos en su desarrollo de forma paralela con el progreso y mejora de distintas tecnologías (Petrone, 2022).

En 2014, Fluent introdujo el uso de unidades de procesamiento gráfico (*Graphics Processing Unit*, GPU) de la mano de los procesadores NVIDIA, mientras que hoy en día cuenta con capacidad de procesamiento en paralelo con múltiples GPUs, además, en 2016, gracias a la computación de alto rendimiento (*High-Performance Computing*, HPC) Fluent rompió un récord empleando más de 172 mil núcleos de procesamiento en un estudio de la Universidad de Stuttgart (ANSYS Inc., 2016).

Por otra parte, en 2018 Fluent implementó la capacidad *Mosaic Meshing* la cual permite la transición entre distintos tipos de elementos de malla en geometrías complejas para generar mallas dinámicas, logrando simulaciones notablemente más rápidas y precisas con la necesidad de utilizar menos RAM (ANSYS Inc., n.d.).

### III. METODOLOGÍA

En el siguiente capítulo se describe la metodología que se llevó a cabo para cumplir con los objetivos planteados, para ello se siguieron los pasos del diagrama de flujo de la metodología mostrado en la Figura 35.

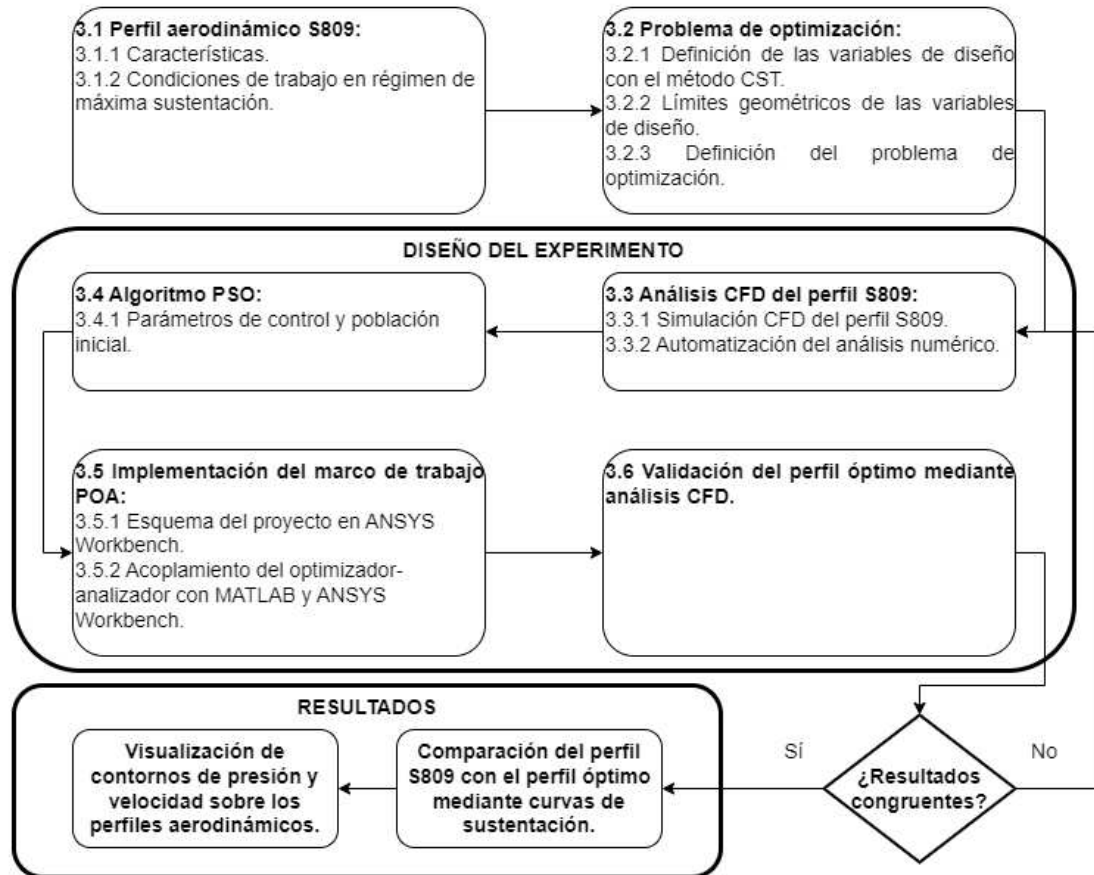


Figura 35. Diagrama de flujo de la metodología.

De acuerdo con el diagrama de la Figura 35, una vez que se definieron las características y condiciones de trabajo del perfil S809 y se definió el problema de optimización, se realizó el diseño del experimento definido por los pasos 3.3, 3.4, 3.5 y 3.6, y cuyo proceso se conformó por tres diferentes tipos de análisis numéricos: 1) análisis CFD del perfil S809 con  $0^\circ \leq \alpha \leq 30^\circ$ , 2) análisis CFD del perfil durante el proceso de optimización, y 3) análisis CFD del perfil óptimo con  $0^\circ \leq \alpha \leq 30^\circ$ . Finalmente, con los resultados obtenidos se evaluó la congruencia de las curvas de sustentación para validar el perfil óptimo y, de ser necesario, se efectuaron los pasos de analizar los parámetros y

problemas de convergencia, y redefinir los parámetros del modelo de turbulencia CFD indicados en el diagrama de la Figura 12 hasta obtener resultados congruentes de las curvas de sustentación, para finalmente obtener los contornos de presión y velocidad sobre los perfiles aerodinámicos para comparar gráficamente su comportamiento.

### 3.1 PERFIL AERODINÁMICO S809

#### 3.1.1 Características

La mayoría de los perfiles aerodinámicos que se empleaban en HAWT fueron desarrollados originalmente para aviones (*e.g.* perfiles NACA), por lo que surgió la necesidad de desarrollar perfiles que se adecúen a las necesidades de las WT. Consecuentemente, el perfil S809 fue diseñado por el NREL y validado en distintos experimentos en túneles de viento, siendo un perfil para flujo laminar de 21% de espesor normalmente empleado en flujos de viento con  $Re \leq 2 \times 10^6$  (Somers, 1997). La geometría normalizada del perfil S809 se observa en la Figura 36, mientras que sus coordenadas se encuentran en el ANEXO 2.

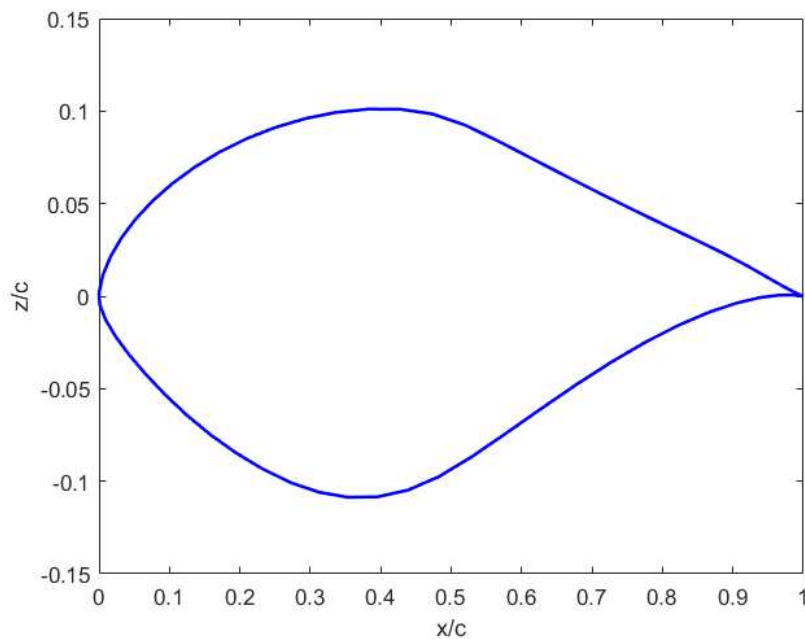


Figura 36. Geometría del perfil aerodinámico S809.



### 3.1.2 Condiciones de trabajo en régimen de máxima sustentación

Los datos de perfiles aerodinámicos obtenidos de experimentos en túneles de viento son importantes ya que este tipo de datos son los que utilizan los diseñadores para desarrollar las palas y rotores de las WT. Es importante que estos correspondan al comportamiento exacto del perfil puesto a prueba, por lo que los datos de túneles de viento deberían representar la misma cuerda, rango de  $Re$ , rango de  $\alpha$ , ubicación de las tomas de presión e imperfecciones de la superficie del perfil. Sin embargo, no es posible coincidir con todos esos criterios en los distintos experimentos de túneles de viento.

Los datos experimentales obtenidos por la *Delft University of Technology* (DUT) se caracterizan porque emplearon un modelo 2D del perfil de 0.6 m de cuerda, se realizaron para un rango de número de Reynolds de  $1.0 \times 10^6 \leq Re \leq 3.0 \times 10^6$  y para ángulos de ataque de  $\alpha \leq 20^\circ$ . Por otra parte, los datos experimentales de la *Ohio State University* (OSU) se caracterizan porque emplearon un modelo 2D del perfil con una cuerda de 0.457 m, se realizaron para  $Re$  de  $7.5 \times 10^5$ ,  $1.0 \times 10^6$  y  $2.0 \times 10^6$ , y para ángulos de ataque de  $\alpha \leq 30^\circ$  (Butterfield et al., 1992).

La *Colorado State University* (CSU) también realizó experimentación en túnel de viento para el perfil S809. La cuerda del modelo 2D del perfil empleado fue de 0.457 m (misma que OSU), mientras que las mediciones se hicieron para  $Re$  de  $3.0 \times 10^5$ ,  $5.0 \times 10^5$  y  $6.5 \times 10^5$  para  $\alpha \leq 90^\circ$ , por lo que estos datos representan el comportamiento del perfil en sus tres regímenes de flujo debido al rango de  $\alpha$  empleado. Los datos experimentales obtenidos por la CSU se consideraron los más adecuados para este estudio debido a que los criterios de medición empleados se asemejan a los criterios empleados en estudios de experimentación del rotor completo de una turbina, por lo que representan el comportamiento exacto de un perfil empleado en la pala de una HAWT (Butterfield et al., 1992).

En la Figura 37 se observa la comparación de  $C_L$  vs.  $\alpha$  de los experimentos mencionados, cada uno con su respectivo números de Reynolds. Estos datos experimentales se obtuvieron de Hand et al. (Hand et al., 2001).

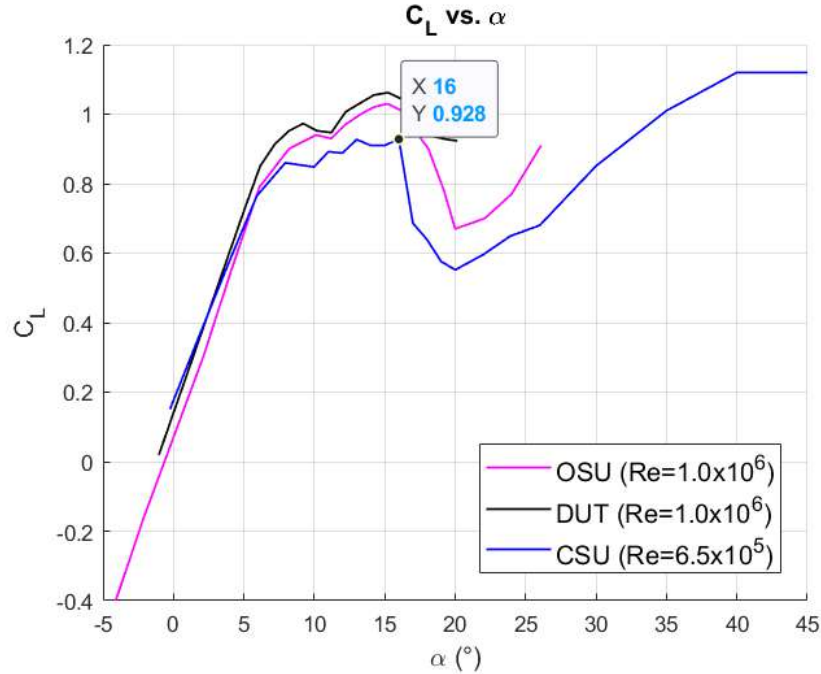


Figura 37. Comparación  $C_L$  vs.  $\alpha$  de los experimentos OSU, CSU y DUT.

De acuerdo con la Figura 37, el experimento de la CSU incluye datos en el rango más amplio del ángulo de ataque, además de corresponder a una velocidad de flujo de viento para HASWT con un  $Re = 6.5 \times 10^5$ , por lo que estos se consideraron como base para validar el análisis numérico realizado con CFD, siendo el coeficiente de sustentación máximo de  $C_{L,max} = 0.928$  a un ángulo de pérdida de  $\alpha_s = 16^\circ$  indicado por el punto señalado sobre la curva color azul.

## 3.2 PROBLEMA DE OPTIMIZACIÓN

### 3.2.1 Definición de las variables de diseño con el método CST

Se utilizó el método CST para parametrizar el perfil aerodinámico S809 y establecer el espacio de búsqueda de las variables de diseño. En la función de clase se emplearon coeficientes con valor de  $N1 = 0.5$  y  $N2 = 1.0$  de acuerdo con la ecuación (7), por lo que se definió como clase fundamental el perfil de nariz redondeada. Por otra parte, la función de forma se utilizó para obtener las variables de diseño mediante los coeficientes del polinomio de Bernstein, de acuerdo con las ecuaciones (13) y (14).

Se empleó un  $BPn(x)$  de orden  $N = 7$  debido a que en el estudio realizado por Kaviani & Moshfegui (2023b) se comprobó que los polinomios de este orden generan un error mínimo cuando se parametriza la geometría con perfiles base de la familia NREL-S, misma del perfil S809, además de que un orden mayor genera “abultamientos” en la zona de succión del borde de salida. De tal manera que al ser el polinomio de Bernstein de orden 7, se consideraron ocho coeficientes por superficie del perfil.

Adicionalmente, como se mostró en la Figura 36, el perfil S809 tiene un espesor del borde de salida de  $\Delta Z_{TE} = 0$  para sus dos superficies, por lo que según las ecuaciones (10) y (11), se sustituyeron los coeficientes de frontera  $A_1$  y  $A_{N+1}$  por los parámetros  $R_{LE}/c$  y  $\beta$  para cada superficie, siendo un total de 16 coeficientes o variables de diseño.

### 3.2.2 Límites geométricos de las variables de diseño

Las variables de diseño presentan límites que indican el rango en el que sus valores fluctúan para corresponder con el espacio de búsqueda del problema de optimización. En el caso de optimización de perfiles aerodinámicos, estos límites dependen de la geometría del perfil.

En el estudio realizado por García-Noverola et al. (2023) se analizó la influencia de los coeficientes de la función de forma del método CST teniendo como referencia el perfil simétrico NACA0012, con la finalidad de visualizar cómo afecta variar los coeficientes en la generación de la geometría del perfil.

Se parametrizó el perfil empleando un  $BPn(x)$  de orden  $N = 7$ , como se muestra en la Figura 38, y se obtuvieron los valores de los parámetros  $R_{LE}/c$  y  $\beta$  correspondientes a los valores de los coeficientes de frontera, los cuales se consideraron iguales para las dos superficies al tratarse de un perfil simétrico, de acuerdo con la ecuación (12). En la Tabla 4 se observa el vector inicial de los coeficientes y los valores obtenidos de los parámetros. El subíndice  $i$  representa el número del coeficiente, siendo  $1 \leq i \leq N + 1$ .

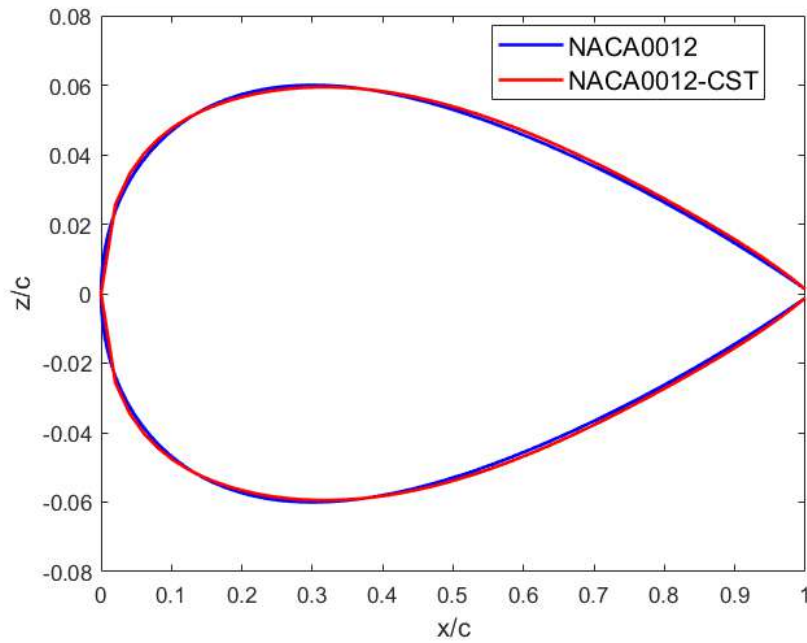


Figura 38. Comparación del perfil NACA0012 con el perfil parametrizado.

Tabla 4. Valores de los coeficientes y parámetros del perfil parametrizado NACA 0012.

Variable/parámetro	Valor
$A_{i=1,\dots,N+1}$	[0.19, 0.14, 0.16, 0.15, 0.15, 0.15, 0.13, 0.17]
$R_{LE}/c$	0.0181
$\beta$	9.5779 °

Posteriormente, se modificaron los valores de los coeficientes en intervalos de +0.05 y -0.05, tal que mientras se modificó el valor de un coeficiente, los demás permanecieron igual al vector original, en las Figuras 39-46 se observa la comparación del perfil NACA0012 con las variaciones de cada uno de los coeficientes. Se concluyó que valores negativos de los coeficientes en la superficie superior generan la misma curva en la superficie superior y viceversa, al ser un perfil simétrico.

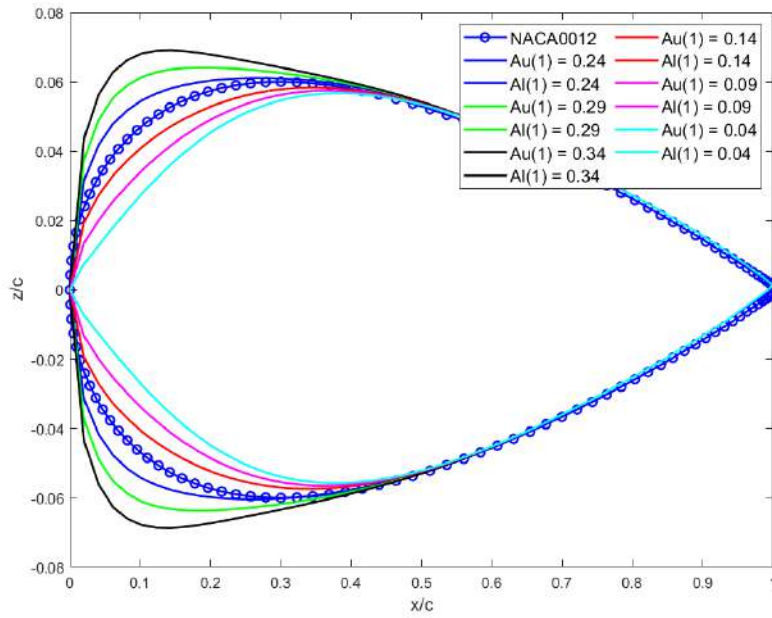


Figura 39. Comparación del NACA0012 con el perfil parametrizado variando  $Au_1$  y  $Al_1$ .

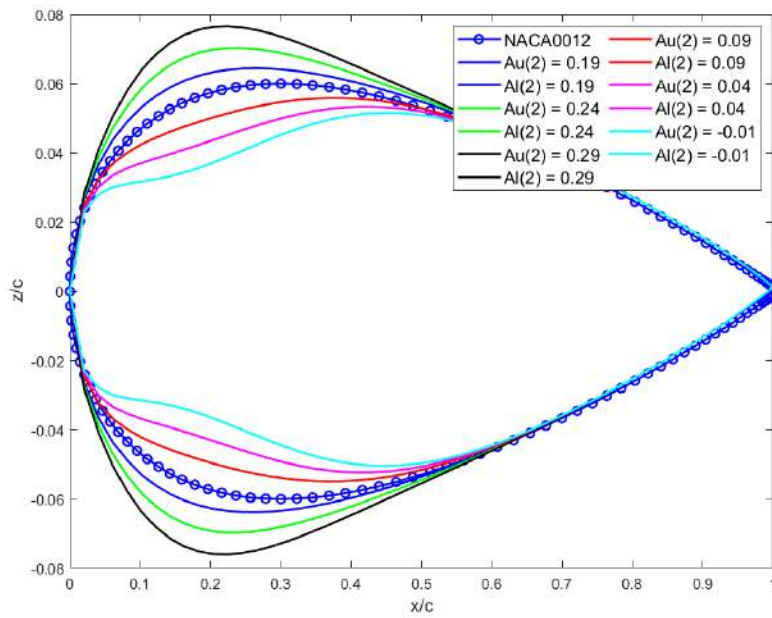


Figura 40. Comparación del NACA0012 con el perfil parametrizado variando  $Au_2$  y  $Al_2$ .

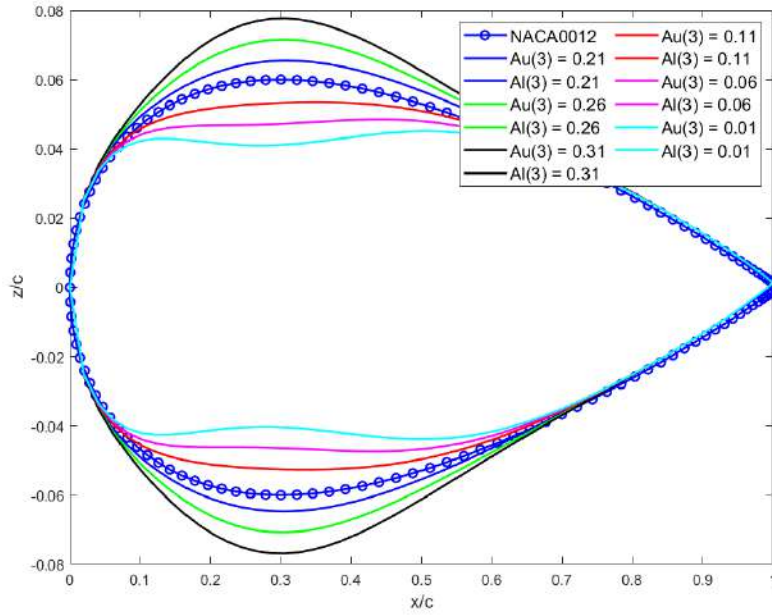


Figura 41. Comparación del NACA0012 con el perfil parametrizado variando  $Au_3$  y  $Al_3$ .

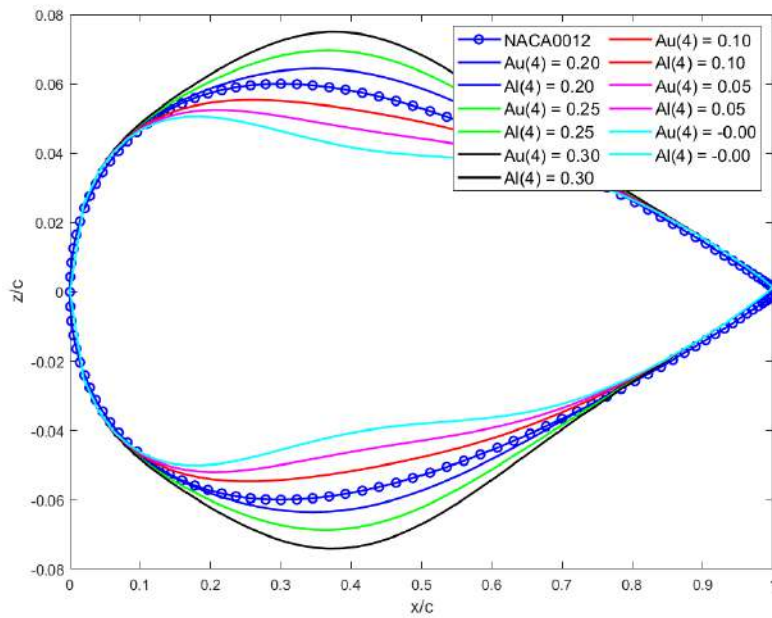


Figura 42. Comparación del NACA0012 con el perfil parametrizado variando  $Au_4$  y  $Al_4$ .

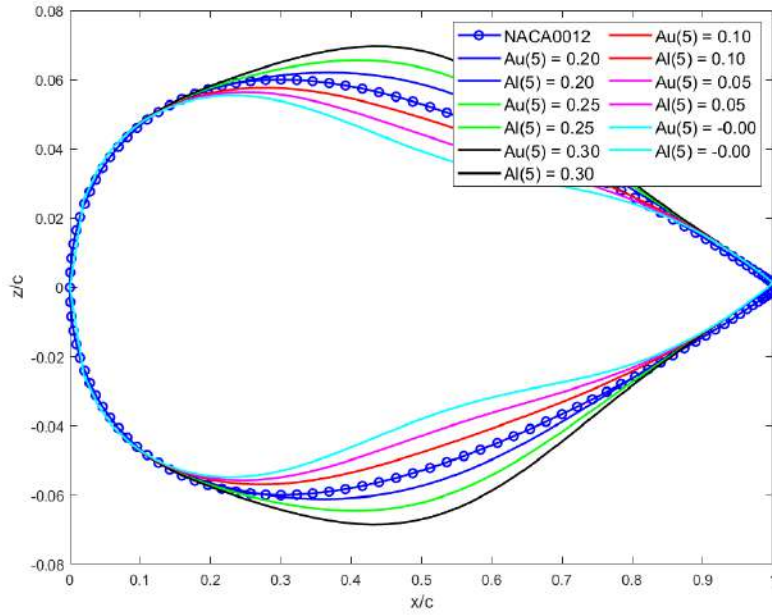


Figura 43. Comparación del NACA0012 con el perfil parametrizado variando  $Au_5$  y  $Al_5$ .

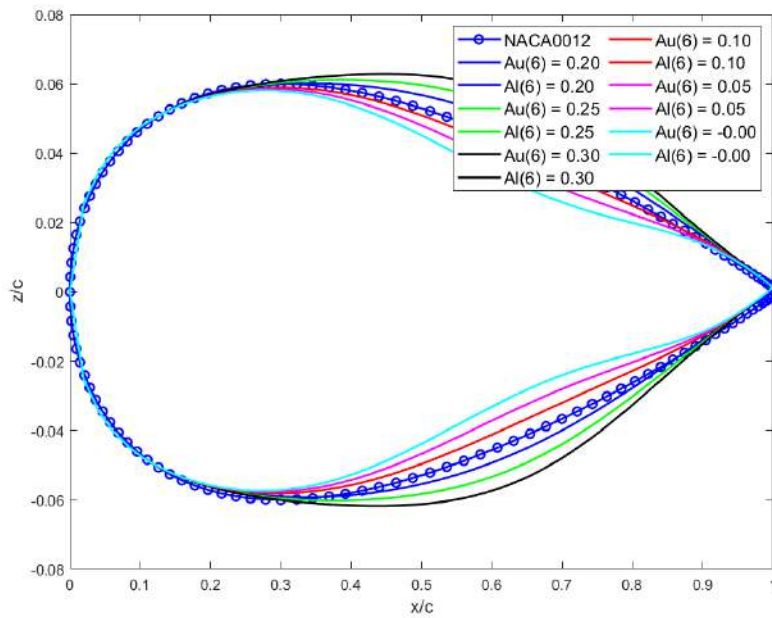


Figura 44. Comparación del NACA0012 con el perfil parametrizado variando  $Au_6$  y  $Al_6$ .

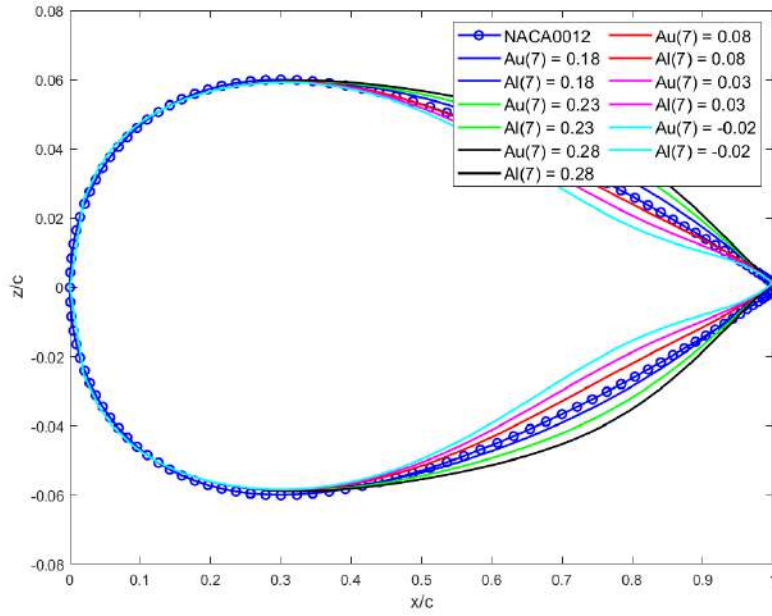


Figura 45. Comparación del NACA0012 con el perfil parametrizado variando  $Au_7$  y  $Al_7$ .

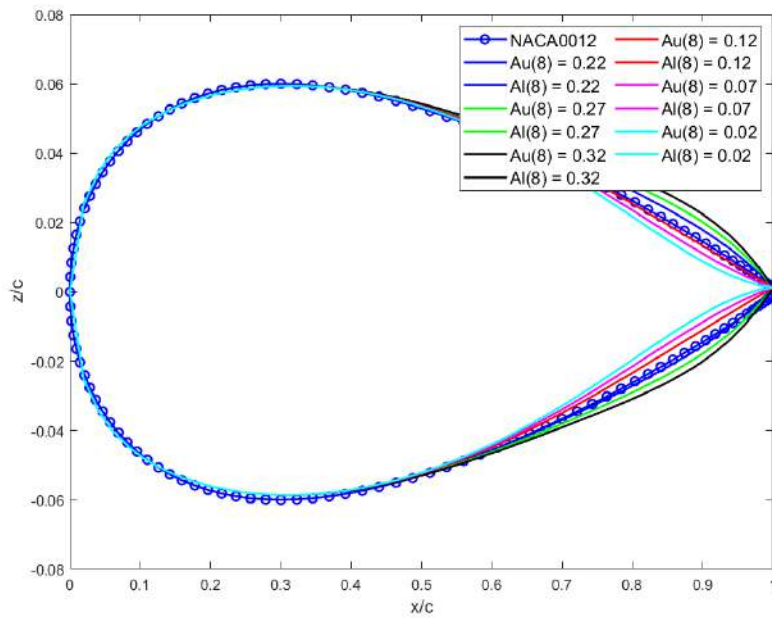


Figura 46. Comparación del NACA0012 con el perfil parametrizado variando  $Au_8$  y  $Al_8$ .



De manera similar, se parametrizó el perfil S809 en cuestión y se obtuvieron los valores de los coeficientes de la función de forma y sus respectivos parámetros  $R_{LE}/c$  y  $\beta$ , con lo que se establecieron los valores iniciales de las 16 variables de diseño del perfil. En la Figura 47 se muestra el perfil parametrizado en comparación con el perfil S809, mientras que en la Tabla 5 se muestran los vectores de los coeficientes y los valores obtenidos de los parámetros para ambas superficies, correspondiendo los subíndices  $u$  y  $l$  a la superficie superior e inferior, respectivamente.

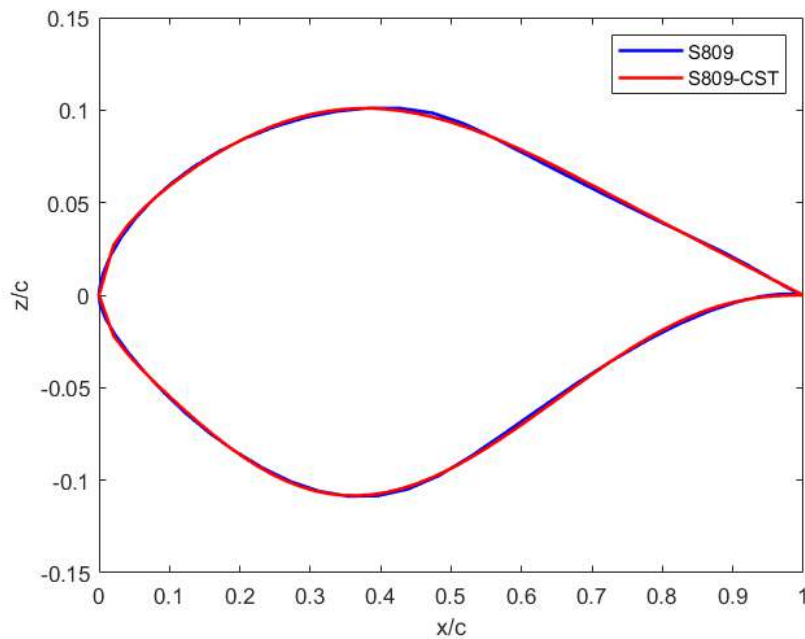


Figura 47. Comparación del perfil S809 con el perfil parametrizado.

Tabla 5. Valores de los coeficientes y parámetros del perfil parametrizado S809.

Variable/parámetro	Valor
$Au_{i=1,\dots,N+1}$	[0.19, 0.2, 0.28, 0.294, 0.288, 0.201, 0.225, 0.19]
$Al_{i=1,\dots,N+1}$	[0.15, 0.19, 0.33, 0.36, 0.235, 0.215, 0.02, $1 \times 10^{-5}$ ]
$Ru_{LE}/c$	0.0181
$Rl_{LE}/c$	0.0112
$\beta_u$	$10.758^\circ$
$\beta_l$	$5.7296 \times 10^{-4}^\circ$

Una vez realizado esto, se decidió utilizar como límites los valores de las variables de diseño del perfil parametrizado S809 con un margen de  $\pm 50\%$  para cada variable. Los límites de las variables de diseño se muestran en la Tabla 6.

Tabla 6. Límites de las variables de diseño.

Variable	Superficie superior ( $u$ )		Superficie inferior ( $l$ )	
	Límite inferior	Límite superior	Límite inferior	Límite superior
$R_{LE}/c$	$R_{LE}/c (1 - 0.5)$	$R_{LE}/c (1 + 0.5)$	$R_{LE}/c (1 - 0.5)$	$R_{LE}/c (1 + 0.5)$
$A_2$	$A_2(1 - 0.5)$	$A_2(1 + 0.5)$	$A_2(1 - 0.5)$	$A_2(1 + 0.5)$
$A_3$	$A_3(1 - 0.5)$	$A_3(1 + 0.5)$	$A_3(1 - 0.5)$	$A_3(1 + 0.5)$
$A_4$	$A_4(1 - 0.5)$	$A_4(1 + 0.5)$	$A_4(1 - 0.5)$	$A_4(1 + 0.5)$
$A_5$	$A_5(1 - 0.5)$	$A_5(1 + 0.5)$	$A_5(1 - 0.5)$	$A_5(1 + 0.5)$
$A_6$	$A_6(1 - 0.5)$	$A_6(1 + 0.5)$	$A_6(1 - 0.5)$	$A_6(1 + 0.5)$
$A_7$	$A_7(1 - 0.5)$	$A_7(1 + 0.5)$	$A_7(1 - 0.5)$	$A_7(1 + 0.5)$
$\beta$	$\beta(1 - 0.5)$	$\beta(1 + 0.5)$	$\beta(1 - 0.5)$	$\beta(1 + 0.5)$

### 3.2.3 Definición del problema de optimización

Se definió la función objetivo  $f(\mathbf{x})$  de acuerdo con la ecuación (1) y los fundamentos del régimen de máxima sustentación del perfil, por lo tanto, la OF se muestra en la ecuación (43).

$$f(\mathbf{x}) = C_{L\alpha_s} \quad (43)$$

Asimismo, el problema de optimización se definió de la siguiente manera como se observa en las ecuaciones (44) a la (61).

Encontrar:

$$\mathbf{x} \left( \frac{Ru_{LE}}{c}, \frac{Rl_{LE}}{c}, Au_2, Au_3, Au_4, Au_5, Au_6, Au_7, Al_2, Al_3, Al_4, Al_5, Al_6, Al_7, \beta_u, \beta_l \right) \quad (44)$$

Para obtener:

$$\max f(x) \quad (45)$$

Sujeto a los límites geométricos:

$$Ru_{LE}/c (1 - 0.5) \leq \frac{Ru_{LE}}{c} \leq Ru_{LE}/c (1 + 0.5) \quad (46)$$

$$Rl_{LE}/c (1 - 0.5) \leq \frac{Rl_{LE}}{c} \leq Rl_{LE}/c (1 + 0.5) \quad (47)$$

$$Au_2(1 - 0.5) \leq Au_2 \leq Au_2(1 + 0.5) \quad (48)$$

$$Al_2(1 - 0.5) \leq Al_2 \leq Al_2(1 + 0.5) \quad (49)$$

$$Au_3(1 - 0.5) \leq Au_3 \leq Au_3(1 + 0.5) \quad (50)$$

$$Al_3(1 - 0.5) \leq Al_3 \leq Al_3(1 + 0.5) \quad (51)$$

$$Au_4(1 - 0.5) \leq Au_4 \leq Au_4(1 + 0.5) \quad (52)$$

$$Al_4(1 - 0.5) \leq Al_4 \leq Al_4(1 + 0.5) \quad (53)$$

$$Au_5(1 - 0.5) \leq Au_5 \leq Au_5(1 + 0.5) \quad (54)$$

$$Al_5(1 - 0.5) \leq Al_5 \leq Al_5(1 + 0.5) \quad (55)$$

$$Au_6(1 - 0.5) \leq Au_6 \leq Au_6(1 + 0.5) \quad (56)$$

$$Al_6(1 - 0.5) \leq Al_6 \leq Al_6(1 + 0.5) \quad (57)$$

$$Au_7(1 - 0.5) \leq Au_7 \leq Au_7(1 + 0.5) \quad (58)$$

$$Al_7(1 - 0.5) \leq Al_7 \leq Al_7(1 + 0.5) \quad (59)$$

$$\beta_u(1 - 0.5) \leq \beta_u \leq \beta_u(1 + 0.5) \quad (60)$$

$$\beta_l(1 - 0.5) \leq \beta_l \leq \beta_l(1 + 0.5) \quad (61)$$

### 3.3 ANÁLISIS CFD DEL PERFIL S809

#### 3.3.1 Simulación CFD del perfil S809

Una vez definidas las condiciones de trabajo relevantes en el régimen de máxima sustentación, se realizó la simulación CFD del comportamiento aerodinámico del perfil S809 para validar el funcionamiento de la herramienta de análisis del marco de trabajo POA propuesto. Se utilizaron los softwares MATLAB y ANSYS 2023R2 versión

estudiante con el fin de realizar el análisis automatizado del perfil S809 y obtener los coeficientes  $C_L$  y  $C_D$  para el rango de ángulos de ataque de  $0^\circ \leq \alpha \leq 30^\circ$ . Es importante mencionar también que todos los análisis numéricos en este estudio se realizaron con una computadora portátil Dell Inspiron G5 15, la cual cuenta con un sistema operativo de 64 bits, un procesador Intel® Core™ i7-10750H de 10<sup>a</sup>. generación con 6 núcleos de hasta 5.0 GHz, 16 GB de memoria RAM y una GPU independiente NVIDIA® GeForce RTX™ 2060 de 6 GB de memoria.

Primero se estableció un esquema del proyecto en ANSYS Workbench, como se observa en la Figura 48, donde se emplearon individualmente los sistemas *Geometry*, *Mesh* y *Fluent* para tener un mejor control en cada uno de los procesos de geometría, mallado y análisis, respectivamente, además, se enlazaron los componentes de cada sistema para realizar el análisis secuencial. Sumado a esto, se establecieron parámetros de entrada (*input parameters*) y de salida (*output parameters*) para controlar las condiciones iniciales del fluido en el componente *setup* del sistema *Fluent*, lo cual será explicado más adelante.

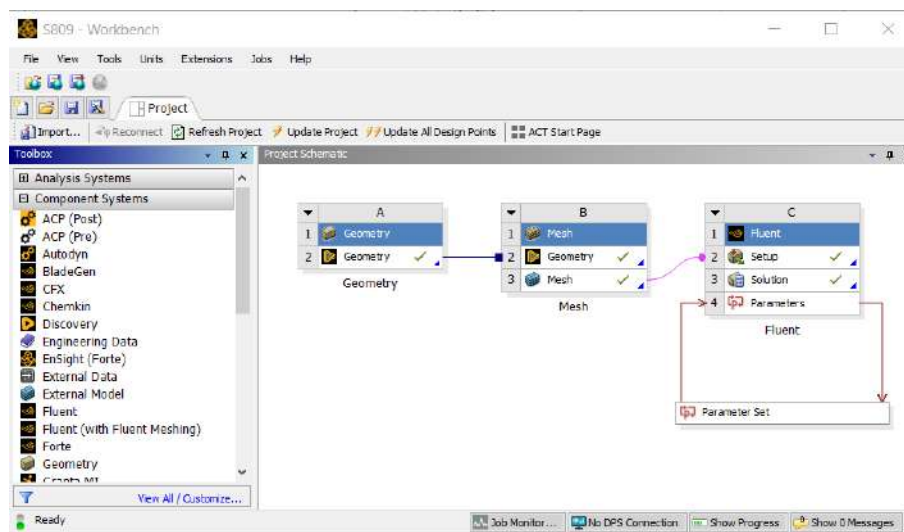


Figura 48. Esquema del proyecto en ANSYS Workbench.

### 3.3.1.1 Geometría del modelo computacional

Para analizar correctamente el comportamiento aerodinámico del flujo de fluido alrededor del perfil, es necesario que el flujo perturbado por la superficie de este se estabilice y recupere sus condiciones iniciales, por eso se recomienda que los límites del dominio del modelo se encuentren lo suficientemente lejos del perfil aerodinámico.

El dominio de tipo C es una geometría utilizada comúnmente para el análisis de perfiles aerodinámicos ya que permite cambios rápidos en los componentes de la velocidad del fluido sin la necesidad de cambiar la malla, lo cual permite hacer análisis a diferentes  $Re$  y  $\alpha$ , como en los estudios realizados por Mauro et al. (2017) y Rodríguez & Celis (2022).

La geometría del modelo computacional empleada se muestra en la Figura 49, donde  $c$  indica la cuerda del perfil, mientras que *Velocity-inlet* son las fronteras de entrada de fluido, *Pressure-outlet* es la frontera de salida de fluido y *Wall* es la frontera de la superficie del perfil. Estas fronteras se especificarán más adelante en la sección de análisis del modelo.

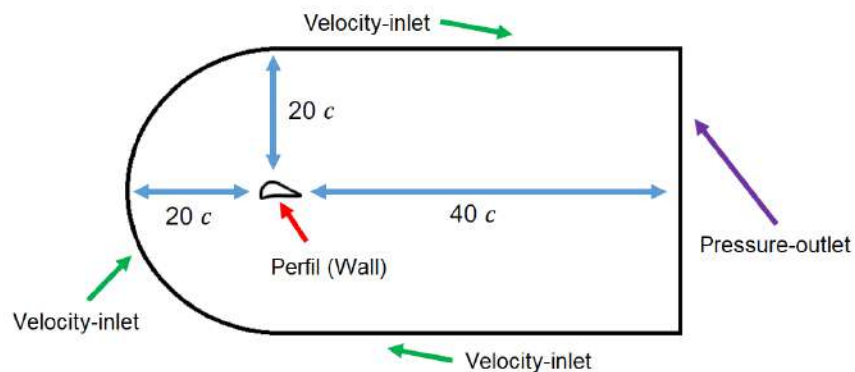


Figura 49. Geometría del modelo computacional para el análisis (Rodríguez & Celis, 2022).

El modelo computacional se generó con el software *SpaceClaim* en el componente *Geometry* dentro del sistema *Geometry*, para ello se empleó un archivo de texto para crear la superficie del perfil S809 con los puntos de sus coordenadas, como se muestra en la Figura 50, considerando que  $c = 1\text{ m}$ . Por otra parte, para crear el modelo computacional completo se utilizó un *script* en lenguaje Python en el cual se escribió el código para importar el archivo de coordenadas y realizar las operaciones que generaron el dominio del fluido alrededor de la superficie del perfil. En la Figura 51 se observa el modelo generado por el *script* en Python, mientras que el código del *script* se encuentra en el ANEXO 3.

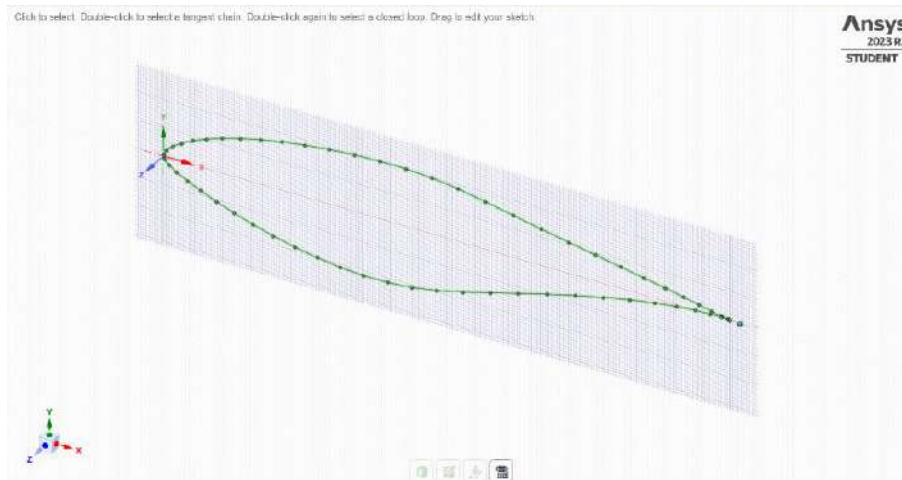


Figura 50. Superficie del perfil S809 desde sus coordenadas.

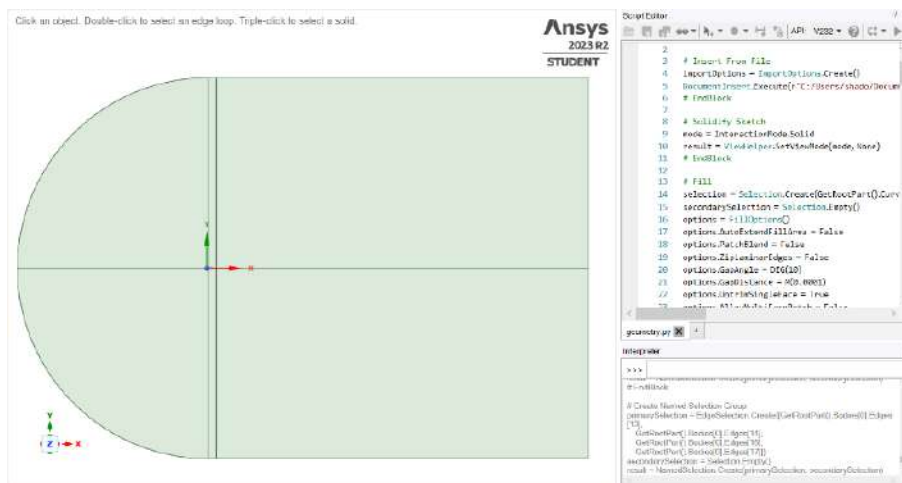


Figura 51. Modelo 2D y el script editor de SpaceClaim.

Las divisiones que se observan en la geometría del modelo se definieron para generar una malla estructurada, lo cual se explicará más adelante. Consecuentemente, en la Figura 52 se muestra un acercamiento de la superficie del perfil en el modelo.

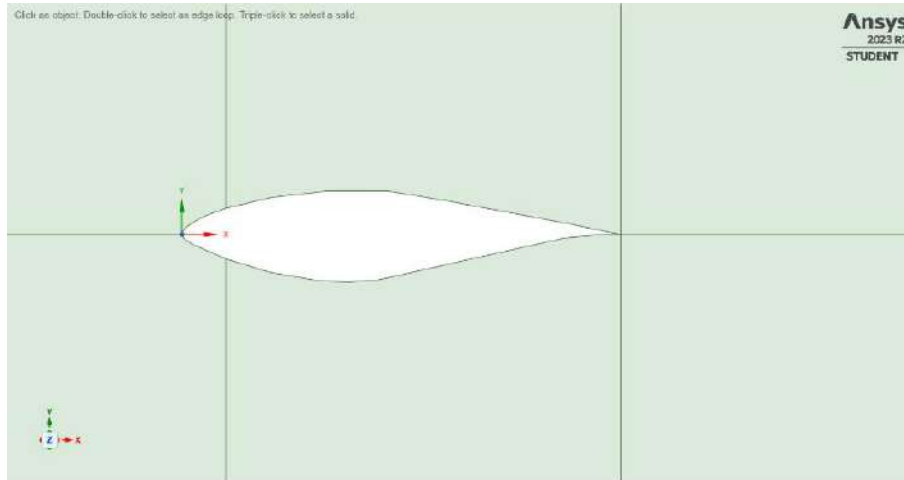


Figura 52. Detalle de la superficie del perfil S809 en el modelo.

### 3.3.1.2 Discretización del modelo computacional

Una vez generada la geometría, la discretización (o mallado) del modelo se realizó con *Meshing* en el componente *Mesh* dentro del sistema *Mesh*. Se generó una malla estructurada aprovechando la geometría del dominio tipo C, con el antecedente de que las mallas estructuradas presentan soluciones de mayor calidad comparadas con las no estructuradas (Rodríguez & Celis, 2022). La malla completa del modelo se muestra en la Figura 53, mientras que en la Figura 54 se observa un acercamiento a la malla próxima a la superficie del perfil aerodinámico.

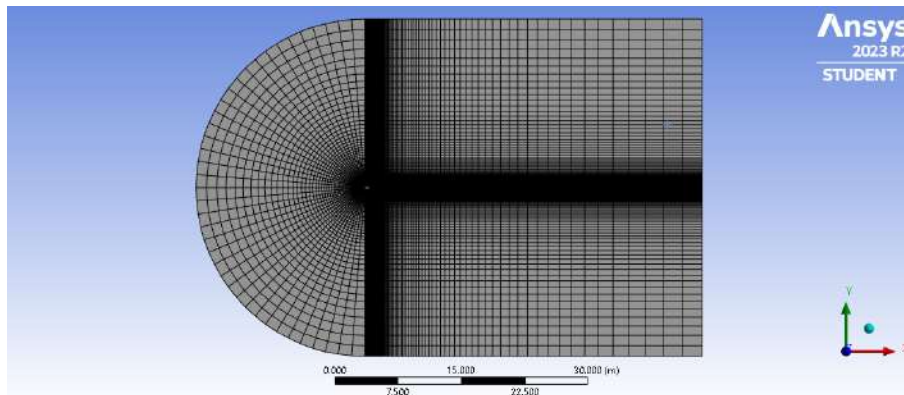


Figura 53. Malla estructurada del modelo computacional.

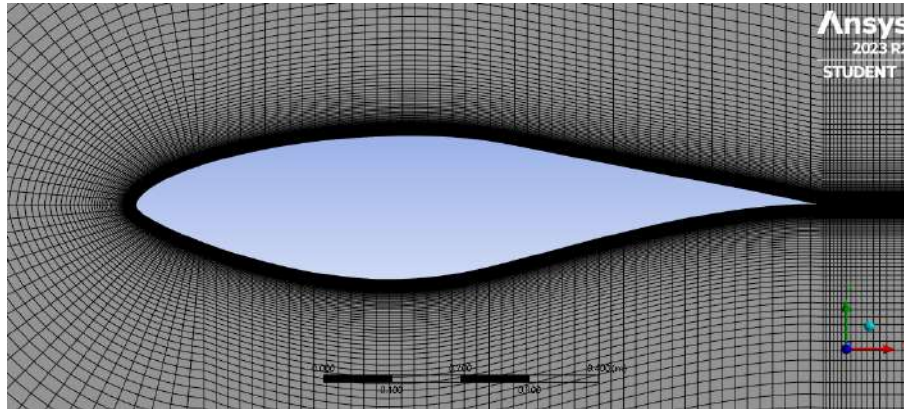


Figura 54. Detalle de la malla próxima a la superficie del perfil.

Cabe mencionar que se aplicaron controles de tipo *sizing* en las divisiones del modelo para tener un mayor control sobre la malla estructurada. El control *rear-length* se utilizó en la zona detrás del borde de salida del perfil, mientras que el control *domain-height* se empleó principalmente para definir la capa límite de la superficie del perfil, lo cual es útil para analizar de manera precisa la transición del fluido laminar a turbulento. En las Figuras 55 y 56 se observan los controles *rear-length* y *domain-height* sobre la malla del modelo, respectivamente, mientras que en la Figura 57 se muestran los elementos de malla en la capa límite del perfil.

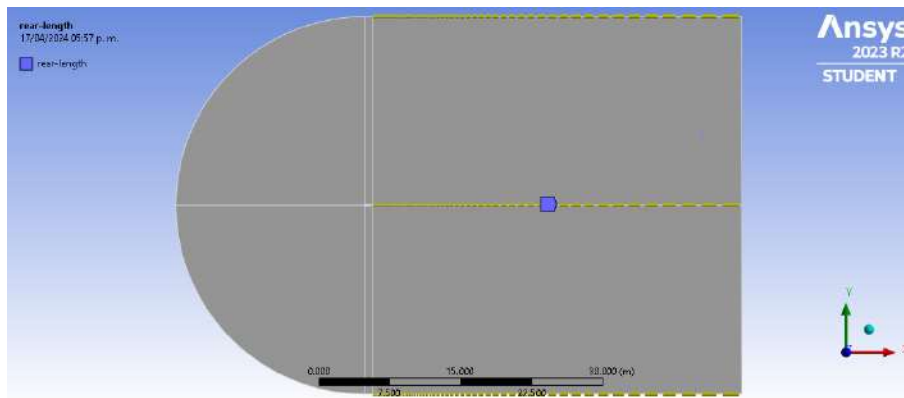


Figura 55. Control *rear-length* sobre la malla.



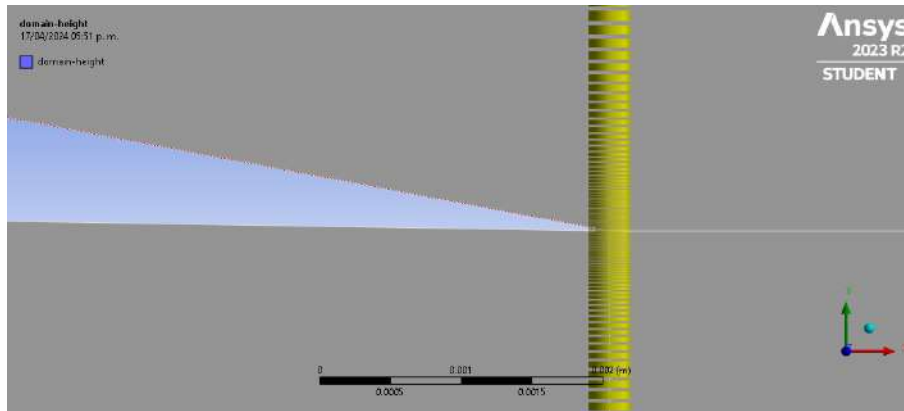


Figura 56. Control *domain-height* cercano al borde de salida.

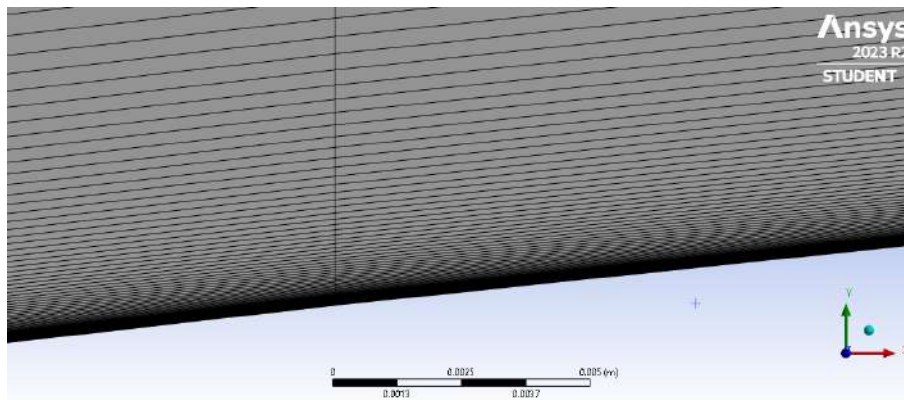


Figura 57. Elementos de malla en la capa límite de la superficie del perfil.

También se aplicaron controles sobre la superficie superior, la superficie inferior, y el borde de ataque del perfil aerodinámico. En la Figura 58 se muestra el control *airfoil-bottom* empleado en un gran porcentaje de la superficie inferior, el cual fue similar al control *airfoil-top* de la superficie superior, mientras que en la Figura 59 se observa el control *leading-edge* aplicado al borde de ataque.

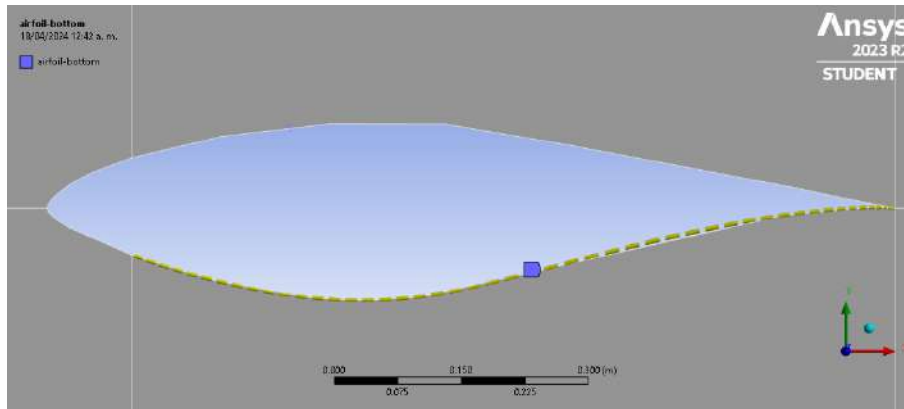


Figura 58. Control *airfoil-bottom* en la superficie inferior del perfil.

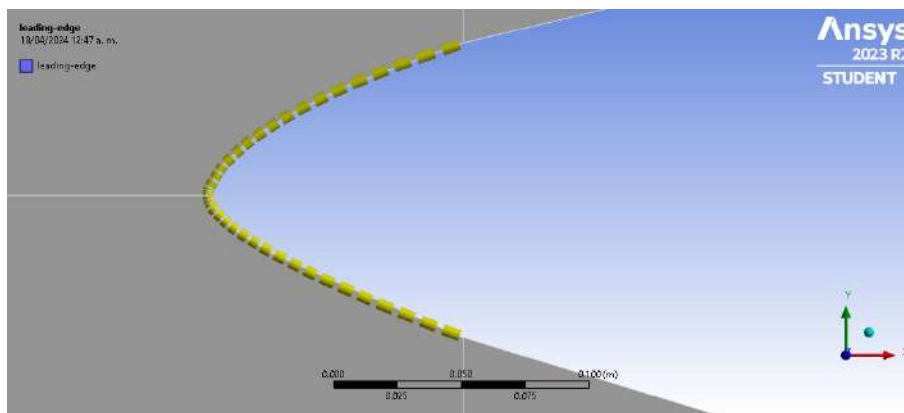


Figura 59. Control *leading-edge* aplicado al borde de ataque.

De igual manera, se aplicó un control de tipo *face meshing* sobre todas las caras de la geometría del modelo consideradas como fluido con la finalidad de que se generen elementos de malla estructurados; en la Tabla 7 se observan las características principales definidas en cada control sobre la malla del modelo.

Tabla 7. Características principales de los controles de malla en el modelo.

Nombre del control	Tipo (cantidad)	Comportamiento	Captura la curvatura	Bias	Factor bias	Método de mapeo
<i>rear-length</i>	No. de divisiones (100)	<i>Hard</i>	No	Sí	500	
<i>domain-height</i>	No. de divisiones (200)	<i>Hard</i>	Sí	Sí	500,000	
<i>airfoil-top</i>	No. de divisiones (50)	<i>Hard</i>	No	No		
<i>airfoil-bottom</i>	No. de divisiones (50)	<i>Hard</i>	No	No		

<i>leading-edge</i>	No. de divisiones (20)	<i>Hard</i>	No	No		
<i>Face meshing</i>						Cuadriláteros

Posteriormente, para el control global de la malla, en las características de tipo *sizing* se emplearon elementos de orden lineal con un tamaño de  $3.6056\text{ m}$  a una tasa de crecimiento de 1.2, mientras que el tamaño mínimo de los elementos en curvatura fue de  $3.6056 \times 10^{-2}\text{ m}$ . Por otro lado, para las características de tipo *inflation* se usó una inflación de transición suave con una tasa de transición de 0.272 y una tasa de crecimiento de 1.2. Finalmente, como consecuencia de todos los controles de malla aplicados se generaron 68,000 elementos tipo *QuadShell4*. Las estadísticas que se obtuvieron una vez creada la malla del modelo se muestran en la Tabla 8.

Tabla 8. Estadísticas de la malla del modelo.

Estadística	Cantidad
Nodos	68,440
Elementos	68,000
Nodos de esquina	68,440
Elementos tipo <i>Shell</i>	68,000
Elementos tipo <i>QuadShell4</i>	68,000

Es importante mencionar que al igual que en la geometría del modelo, en el proceso de discretización también se generó un *script* en Python para automatizar todas las operaciones que se utilizaron en la generación del mallado, el cual servirá más adelante en el proceso de automatización del análisis en ANSYS Workbench. El código del *script* para generar la malla del modelo se muestra en el ANEXO 4.

### 3.3.1.3 Análisis numérico del modelo computacional

Primeramente, se establecieron las características del análisis numérico en el componente *setup* dentro del sistema *Fluent*. Se configuró *Fluent* a través del *Launcher* para realizar el análisis con doble precisión y un procesamiento en paralelo con 4 núcleos,

siendo esto lo que nos permitió al ser una versión de estudiante del software, como se observa en la Figura 60.

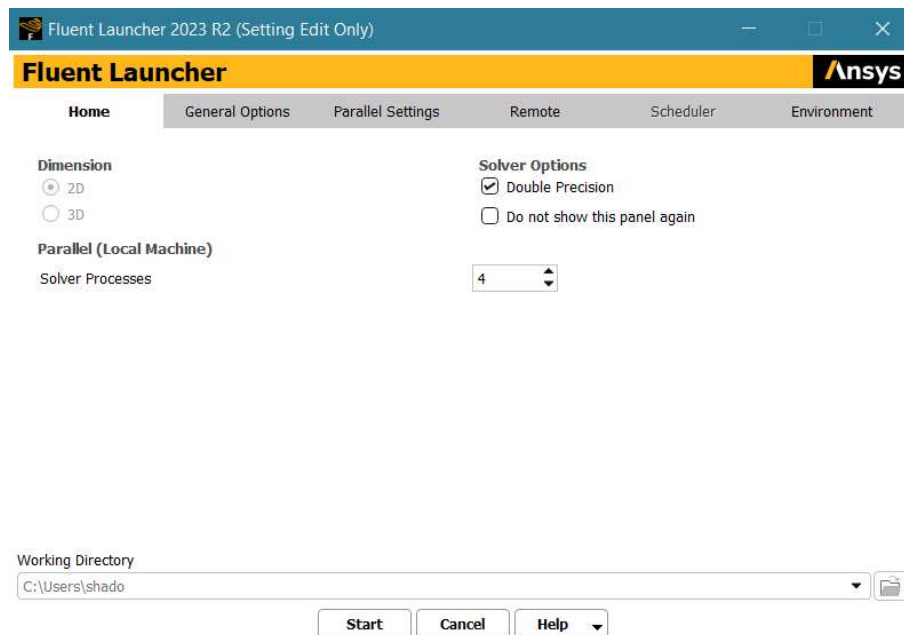


Figura 60. Configuración del Fluent Launcher.

Se consideró utilizar en el análisis las características empleadas en el estudio realizado por Mauro et al. (Mauro et al., 2017), para ello se estableció la configuración del solucionador en estado estable, de tipo *pressure-based* y con una formulación de velocidad absoluta. Como material para el dominio del fluido se empleó aire con una densidad de  $\rho = 1.225 \text{ kg/m}^3$  y una viscosidad dinámica de  $\mu = 1.7894 \times 10^{-5}$ , y de igual manera se definió el modelo de turbulencia de dos ecuaciones SST k- $\omega$ .

Con respecto a las condiciones de frontera, en la frontera *Velocity-inlet* se estableció el método de magnitud y dirección, por lo que de acuerdo con la ecuación (5), a un  $Re = 6.5 \times 10^5$  (mismo que los datos del CSU) le corresponde una velocidad de viento relativo de  $V = 9.49 \text{ m/s}$ . Adicionalmente, los componentes X y Y del flujo de viento se establecieron como *input parameters*, siendo estos los correspondientes al seno y coseno del ángulo de ataque  $\alpha$ . Además, se utilizó una intensidad de turbulencia del 0.1% y una relación de viscosidad turbulenta (*Turbulent Viscosity Ratio*) de  $TVR = 10$ , estos niveles de baja turbulencia corresponden a los producidos normalmente en túneles de viento.

En la frontera de *Pressure-outlet* se estableció el método de dirección de flujo como normal a la frontera con una presión manométrica de  $0 Pa$  y con las mismas especificaciones de turbulencia que en la frontera de entrada. Por otra parte, la frontera *Wall* correspondiente a la superficie del perfil se estableció como estacionaria y con la condición de no deslizamiento. Igualmente se estableció que los cálculos del análisis se inicialicen desde la frontera de entrada *inlet*.

Con respecto a los métodos de solución se estableció para el algoritmo *pressure-velocity* el esquema *coupled*, para la discretización espacial del gradiente se utilizó el esquema *least squares cell-based*, mientras que, para las ecuaciones de presión, momentum y turbulencia se utilizó el esquema *second order-upwind*. El criterio para evaluar la convergencia de la solución se estableció en obtener un valor constante de residuales de  $1 \times 10^{-6}$  para todas las ecuaciones.

Asimismo, se definieron monitores para obtener los valores de  $C_L$  y  $C_D$  en la solución del modelo y se consideraron como *output parameters* para ser almacenados en ANSYS Workbench. Finalmente, se definió la inicialización híbrida de la solución y se estableció el análisis en 1000 iteraciones.

### 3.3.2 Automatización del análisis numérico

Una vez establecidas las condiciones del *setup* del análisis en *Fluent*, se generó un *journal file* para actualizar los *input parameters*, realizar la simulación CFD en *Fluent*, obtener los resultados de los coeficientes de sustentación y arrastre del perfil como *output parameters* dentro del rango de ángulos de ataque de  $0^\circ \leq \alpha \leq 30^\circ$  y ser comparados con los datos experimentales mencionados anteriormente. El código del *journal file* empleado en el análisis se muestra en el ANEXO 5, mientras que el diagrama del proceso de análisis automatizado para obtener los coeficientes del perfil S809 se muestra en la Figura 61.

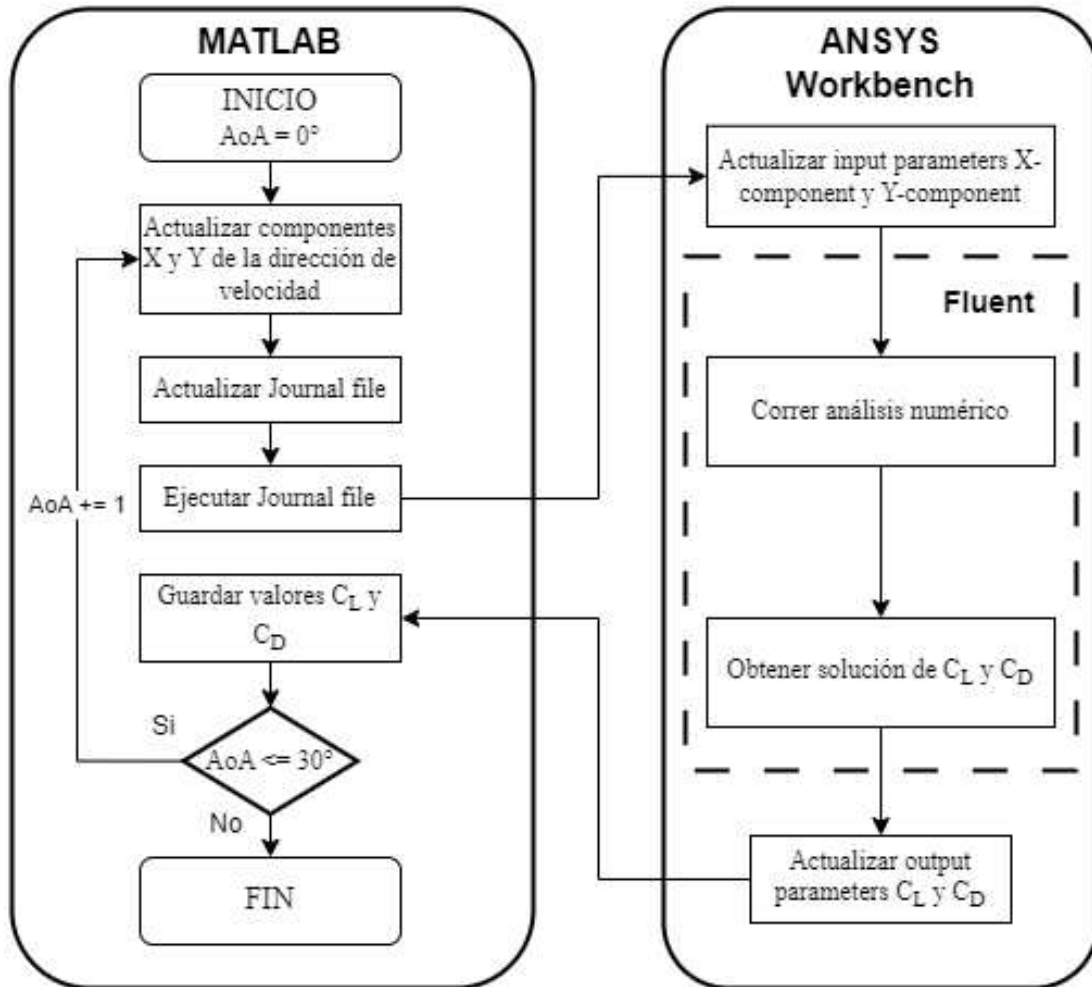


Figura 61. Diagrama del análisis CFD automatizado del perfil S809.

El proceso de automatización se realizó acoplando los softwares MATLAB y ANSYS Workbench mediante un *script* en MATLAB, el cual se utilizó para iterar el valor de  $\alpha$  (*AoA*) en el rango establecido, calcular los componentes X y Y de la velocidad del fluido, actualizar el *journal file*, ejecutar intrínsecamente el esquema de proyecto de ANSYS Workbench de la Figura 48, y realizar el análisis en *Fluent* del modelo generado para obtener los valores de los coeficientes y finalmente ser almacenados en MATLAB. El *script* en MATLAB empleado para la automatización del análisis numérico se observa en el ANEXO 6.

### 3.4 ALGORITMO PSO

#### 3.4.1 Parámetros de control y población inicial

Una vez definido el problema de optimización y realizado el análisis numérico del perfil S809, se desarrolló el algoritmo PSO de acuerdo con los fundamentos descritos. Para ello, se definieron los parámetros de control necesarios para la fase de inicialización, los cuales se observan en la Tabla 9.

Tabla 9. Parámetros de control para la inicialización del algoritmo PSO.

Parámetro (variable en el algoritmo)	Nombre del parámetro	Valor
$d$	Número de variables de diseño	16
$p$	Número de partículas de la población	10
$\omega_{min}$	Factor de peso de inercia mínimo	0.1
$\omega_{max}$	Factor de peso de inercia máximo	0.9
$c_1$	Factor de aprendizaje cognitivo	2
$c_2$	Factor de aprendizaje social	2
$e_1, e_2$	Vectores de factores de aleatoriedad	Vector de números aleatorios entre 0 y 1
$T$	Número máximo de iteraciones	10

Posteriormente, se empleó el pseudocódigo del ANEXO 1 para desarrollar el algoritmo en MATLAB. Se utilizó una función propia del método CST desarrollada en MATLAB, para lo cual se consideró un orden del  $BPn(x)$  de  $N = 7$  y los límites de las variables de diseño definidos en el problema de optimización para generar la población inicial de partículas del enjambre. Las geometrías de las partículas de la población inicial se observan en la Figura 62, en donde  $p_1$  indica la partícula 1,  $p_2$  la partícula 2, y así sucesivamente hasta la partícula 10.

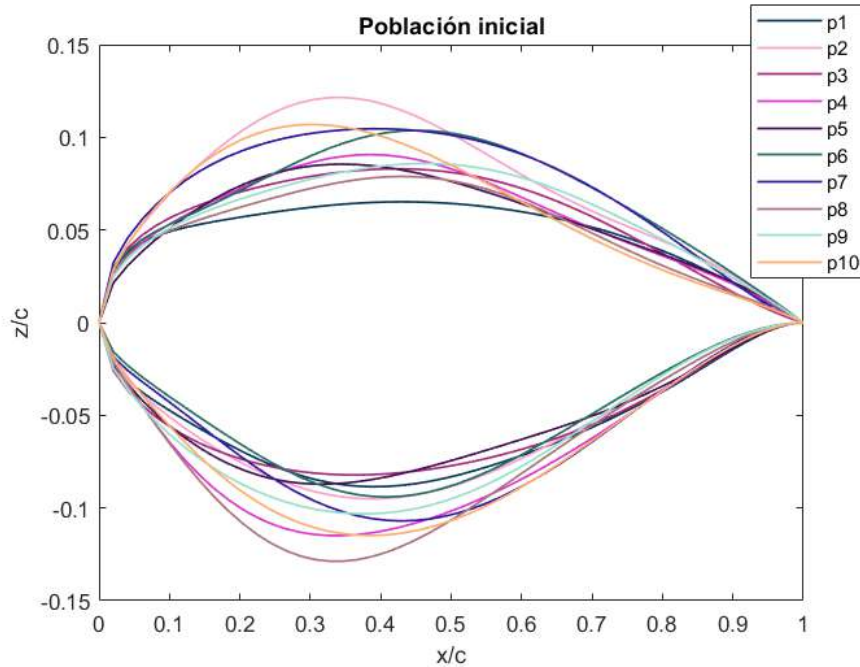


Figura 62. Geometrías de las partículas de la población inicial del enjambre.

Finalmente, se establecieron las variables de diseño de cada partícula como el vector de posición inicial de cada partícula en el enjambre, mientras que el vector de velocidad de cada partícula fue inicializado con valores de cero.

### 3.5 IMPLEMENTACIÓN DEL MARCO DE TRABAJO POA

Para implementar el marco de trabajo POA fue necesario definir el esquema de proyecto en ANSYS Workbench empleado para el analizador. Posteriormente, se implementó la parte de parametrización y optimización a través de un *script* en MATLAB para acoplar el método CST con el algoritmo PSO y el analizador CFD.

#### 3.5.1 Esquema del proyecto en ANSYS Workbench

El esquema del proyecto en ANSYS Workbench utilizado en la optimización del perfil aerodinámico se muestra en la Figura 63, el cual es similar al empleado en el análisis numérico del perfil S809 ya que también empleó los sistemas *Geometry*, *Mesh* y *Fluent* para los procesos de geometría, mallado y análisis, respectivamente. Es importante



mencionar que los *scripts* desarrollados en Python para automatizar la generación de la geometría (ANEXO 3) y la discretización del modelo (ANEXO 4) en la simulación del perfil S809 también fueron utilizados en la optimización del perfil, mientras que para ejecutar de manera interna el análisis CFD en la optimización se empleó un nuevo *journal file*, el cual se observa en el ANEXO 7.

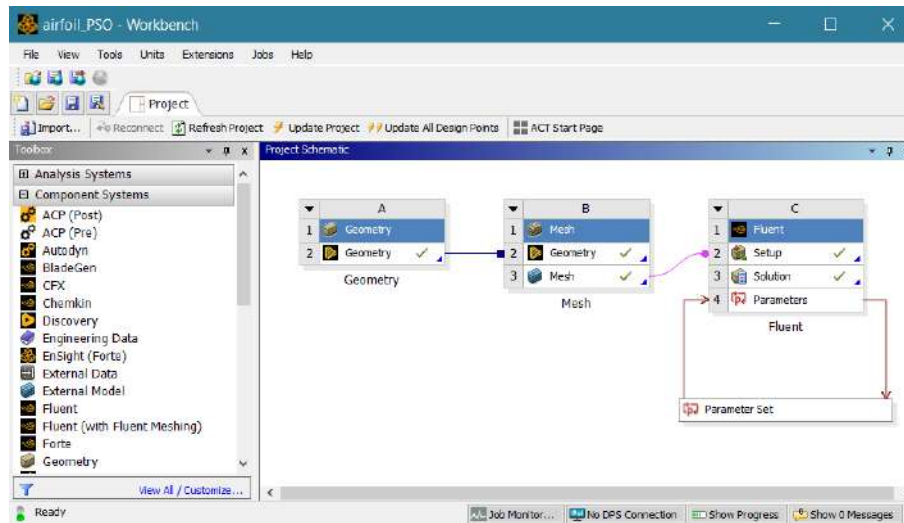


Figura 63. Esquema del proyecto de la optimización del perfil.

Adicionalmente, se emplearon parámetros de entrada (*input parameters*) fijos para establecer los componentes en X y Y de la velocidad del fluido en el sistema de análisis *Fluent*. Debido a que la optimización del perfil se realizó para el ángulo de ataque correspondiente al régimen de máxima sustentación obtenido del análisis numérico del perfil S809 ( $\alpha_s = 15^\circ$ ), se utilizaron valores de los componentes de la velocidad del fluido de  $x\_component = 0.9659$  y  $y\_component = 0.2588$ , mientras que los coeficientes de sustentación y arrastre del perfil aerodinámico se consideraron como parámetros de salida (*output parameters*), como se observa en la Figura 64.

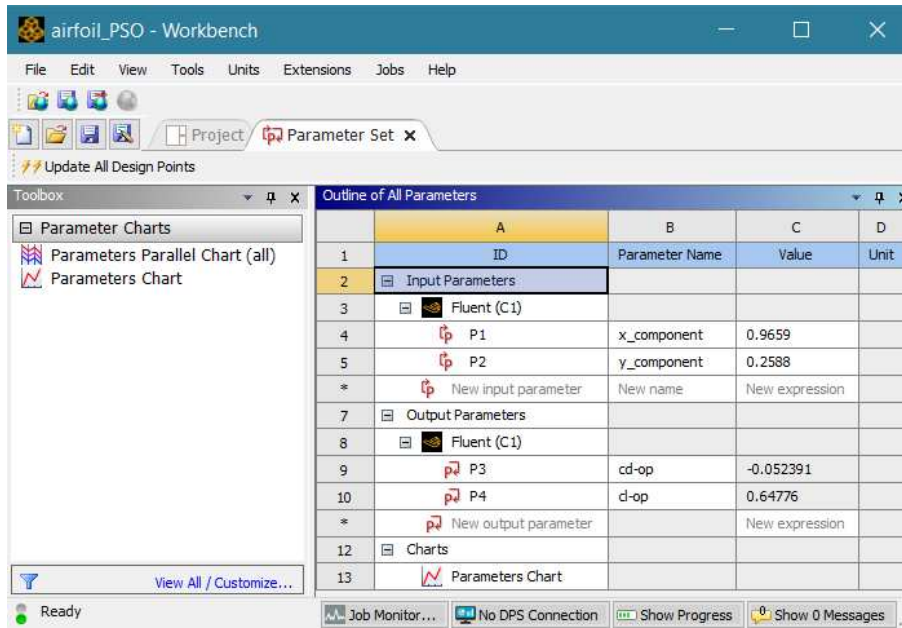


Figura 64. Parámetros de entrada y salida del análisis numérico de la optimización.

Una vez definido el esquema del proyecto en ANSYS Workbench, se realizó el acoplamiento del método CST con el algoritmo PSO y la herramienta de análisis CFD.

### 3.5.2 Acoplamiento del optimizador-analizador con MATLAB y ANSYS Workbench

El acoplamiento del optimizador con el analizador se realizó mediante un *script* en MATLAB en el cual se ejecutó intrínsecamente el esquema de proyecto de ANSYS Workbench, este proceso se muestra detalladamente en el diagrama de la Figura 65.

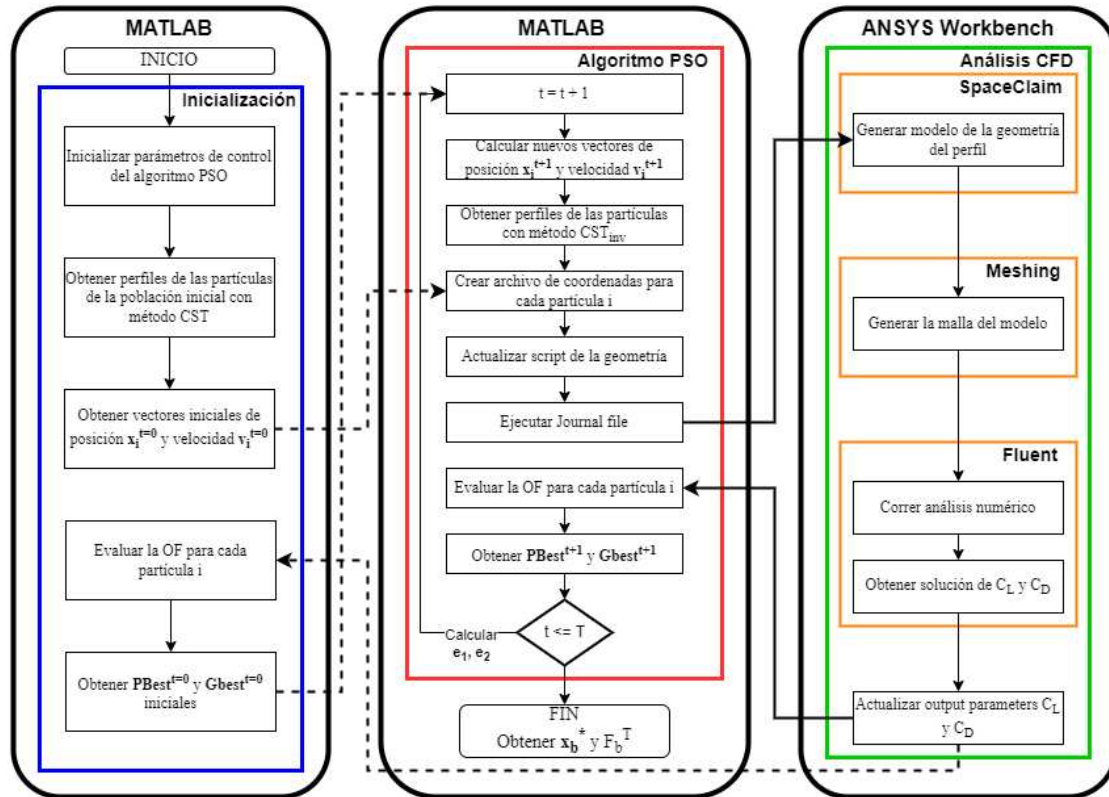


Figura 65. Diagrama del marco de trabajo POA para la optimización del perfil.

En el diagrama se observan tres bloques, dos en MATLAB y uno en ANSYS Workbench, siendo el primero el de la izquierda. Es importante mencionar que las flechas punteadas en el diagrama indican el flujo del algoritmo para el análisis numérico de la población inicial del enjambre, mientras que las flechas continuas gruesas indican el flujo del algoritmo del acoplamiento entre MATLAB y ANSYS Workbench, y las flechas continuas delgadas indican el flujo del algoritmo dentro de cada bloque del diagrama. En el primer bloque se realiza la inicialización del algoritmo PSO en donde se inicializan los parámetros de control, se obtiene la población inicial del enjambre de partículas y los vectores iniciales de posición y velocidad.

Seguidamente, en el segundo bloque se crearon los archivos de coordenadas de cada partícula para actualizar el *script* de la geometría del perfil, y se ejecutó el *journal file* para realizar el análisis CFD en ANSYS Workbench (tercer bloque), en donde mediante las instrucciones del *journal file* se ejecutaron internamente *SpaceClaim*, *Meshing* y *Fluent*

para generar el modelo de la geometría del perfil, discretizarlo y realizar la simulación numérica, respectivamente.

Para el análisis de la población inicial, los valores de los coeficientes  $C_L$  y  $C_D$  se obtuvieron como parámetros de salida y se procesaron en MATLAB para evaluar la OF de cada partícula y así obtener los vectores  $PBest^{t=0}$  y  $GBest^{t=0}$ . Posteriormente, se incrementó el periodo del algoritmo y se calcularon los vectores de posición y velocidad de las partículas de la iteración actual de acuerdo con las ecuaciones (16) y (17), y se obtuvieron los perfiles de cada partícula con el método CST inverso (en el diagrama indicado como  $CST_{inv}$ ) mediante una función propia en MATLAB, la cual devolvió las coordenadas del perfil conociendo los coeficientes de la función de forma.

Finalmente, para cada partícula se realizaron los procesos de crear el archivo de coordenadas, actualizar el *script* de la geometría, ejecutar el *journal file* para efectuar el análisis numérico en ANSYS Workbench y obtener los coeficientes de cada perfil aerodinámico, así como evaluar la OF para obtener los nuevos vectores  $PBest^{t+1}$  y  $GBest^{t+1}$ . Este proceso se realizó hasta que se cumplió la condición del número máximo de iteraciones ( $T$ ), de tal manera que mientras no se alcanzó la condición, se calcularon los factores de aleatoriedad  $e_1$  y  $e_2$  y se aumentó el valor de la iteración actual. Por el contrario, cuando la condición  $t \leq T$  se cumplió, se obtuvieron las variables de diseño  $x_b^*$  del perfil óptimo correspondientes a  $GBest^T$  y el valor fitness de la población (VFP) de  $F_b^T$ , el cual corresponde al valor de la OF del perfil óptimo. En el ANEXO 8 se muestra el *script* en MATLAB empleado para el acoplamiento del optimizador-analizador.

### 3.6 VALIDACIÓN DEL PERFIL ÓPTIMO MEDIANTE ANÁLISIS CFD

Una vez obtenido el perfil óptimo, se realizó la validación de este mediante un análisis numérico con CFD para comparar su curva del coeficiente de sustentación con la del perfil S809. Para ello, se realizó el mismo proceso que en la sección 3.3 de la metodología; se utilizó el mismo modelo computacional, la misma configuración de malla estructurada de tipo C y controles locales, así como las características del análisis numérico en *Fluent*.

Para la generación del modelo computacional, se generó el archivo de coordenadas del perfil óptimo y se implementó en el *script* de la geometría del modelo, en la Figura 66 se muestra en detalle la superficie del perfil óptimo dentro del modelo generado en *SpaceClaim*.

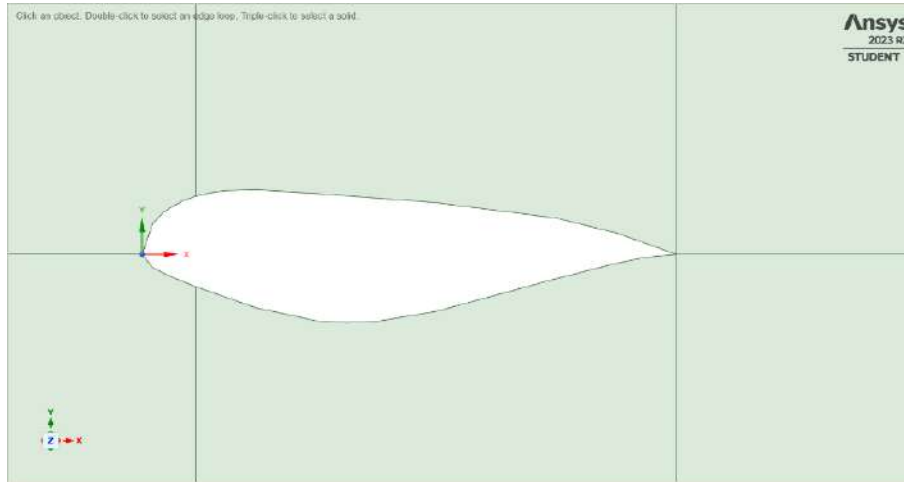


Figura 66. Detalle de la superficie del perfil óptimo en el modelo.

Para la discretización del modelo se empleó también el *script* de la malla, la cual una vez generada, resultó con las mismas características en cantidad y tipo de elementos que en el análisis del perfil S809. En la Figura 67 se observa un acercamiento de la malla próxima a la superficie del nuevo perfil.

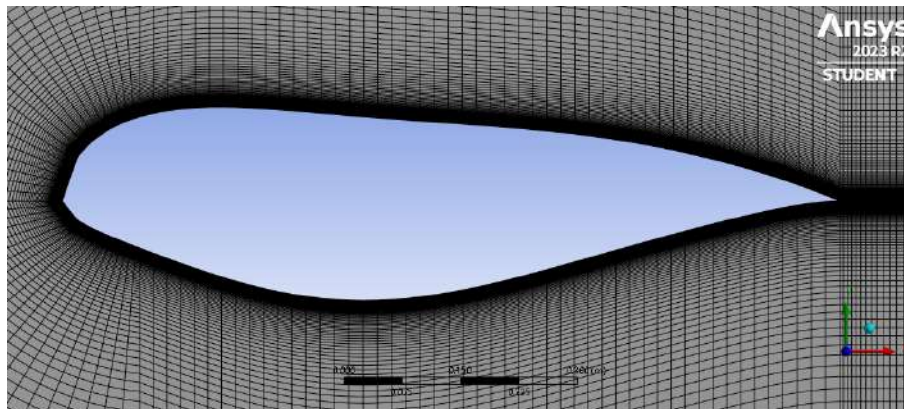


Figura 67. Detalle de la malla próxima a la superficie del perfil óptimo.

Finalmente, para correr de manera automatizada el análisis del perfil óptimo en un rango de  $0^\circ \leq \alpha \leq 30^\circ$ , nuevamente se empleó el *script* en MATLAB del ANEXO 6 y se siguió el diagrama de flujo de la Figura 61.

## IV. RESULTADOS

### 4.1 CURVA DE SUSTENTACIÓN DEL PERFIL S809 CON ANÁLISIS CFD

En la Figura 68 se observa la comparación de la curva de sustentación del perfil S809 de los datos experimentales de la CSU (curva azul) con la curva de sustentación obtenida con el análisis CFD (curva roja). Como se observa en la gráfica, para el análisis CFD se obtuvo un  $C_{L,max} = 0.879$  a un  $\alpha_s = 15^\circ$ , siendo este el  $C_L$  por maximizar. El tiempo de simulación ( $t_s$ ) de este análisis fue de  $t_s = 106.18 \text{ min} = 1.77 \text{ hr}$ , para 31 corridas del perfil aerodinámico.

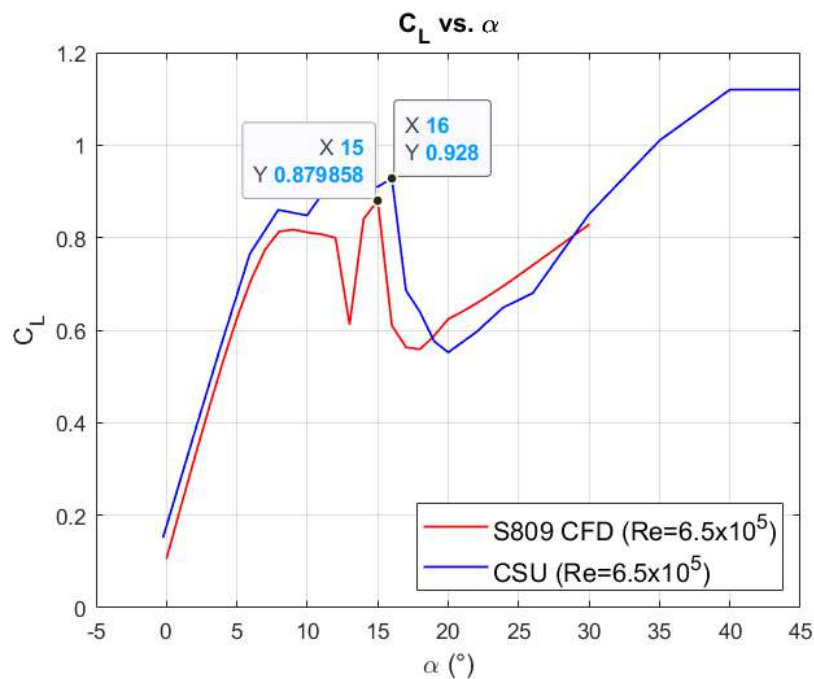


Figura 68. Gráfica de la comparación de la curva de sustentación del perfil S809 con datos experimentales.

De igual manera, en la Figura 68 se observa que el comportamiento de los tres regímenes de flujo de la curva en el análisis CFD corresponde al comportamiento típico del perfil, sin embargo, cabe mencionar que se notó una irregularidad en la curva perteneciente a  $\alpha = 13^\circ$ , lo cual generó un valor incongruente de la curva antes de llegar al régimen de máxima sustentación.

## 4.2 OPTIMIZACIÓN DEL PERFIL AERODINÁMICO

La optimización del perfil se realizó a un ángulo de ataque fijo de  $\alpha = 15^\circ$  correspondiente al ángulo de pérdida del análisis CFD del perfil S809. Una vez finalizada la corrida del algoritmo y obtenida la población inicial, se obtuvieron las geometrías de las partículas en cada uno de los 10 periodos hasta obtener el perfil óptimo, el cual se generó con las variables de diseño de la partícula obtenidas del vector de posición  $\mathbf{x}_b^*$ , correspondientes a la mejor posición encontrada en el enjambre  $\mathbf{GBest}^T$  en periodo final  $T$ . En las Figuras 69-78 se observa la evolución de las geometrías de las 10 partículas en cada iteración, es decir, de  $1 \leq k \leq 10$ . El tiempo de simulación de este análisis fue de  $t_s = 813.63 \text{ min} = 13.56 \text{ hr}$ , para 110 corridas del perfil aerodinámico.

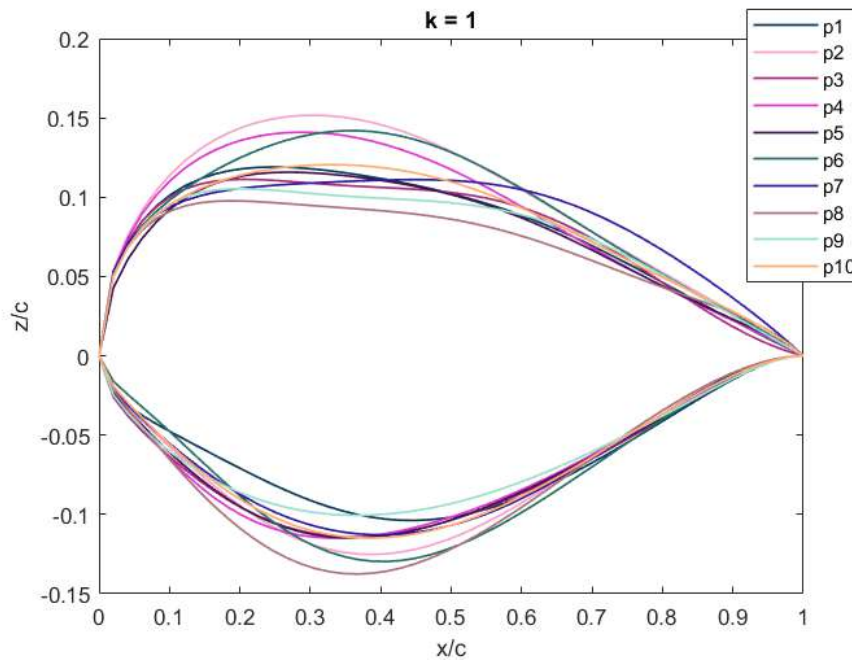


Figura 69. Geometrías de las partículas en la iteración  $k = 1$ .



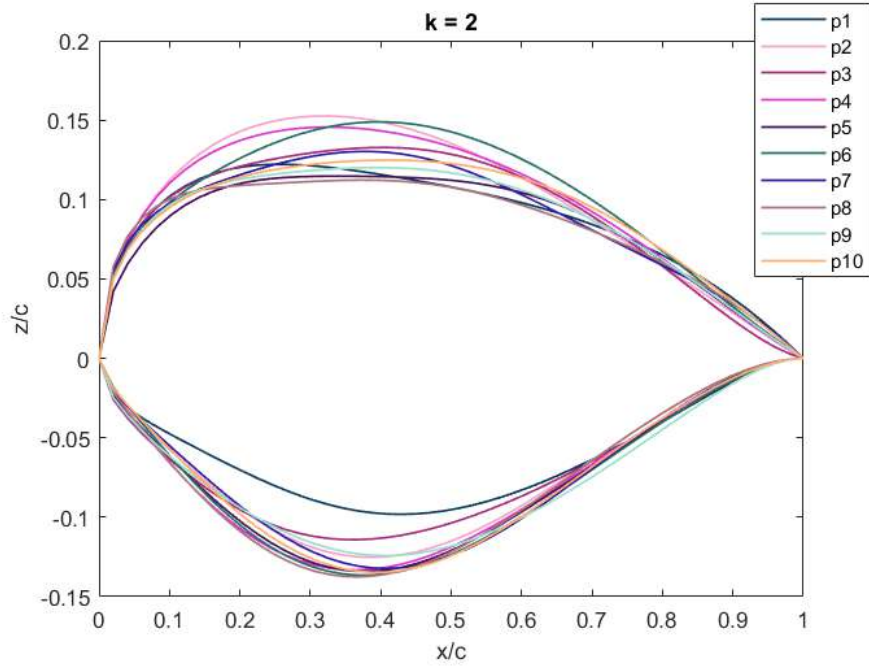


Figura 70. Geometrías de las partículas en la iteración  $k = 2$ .

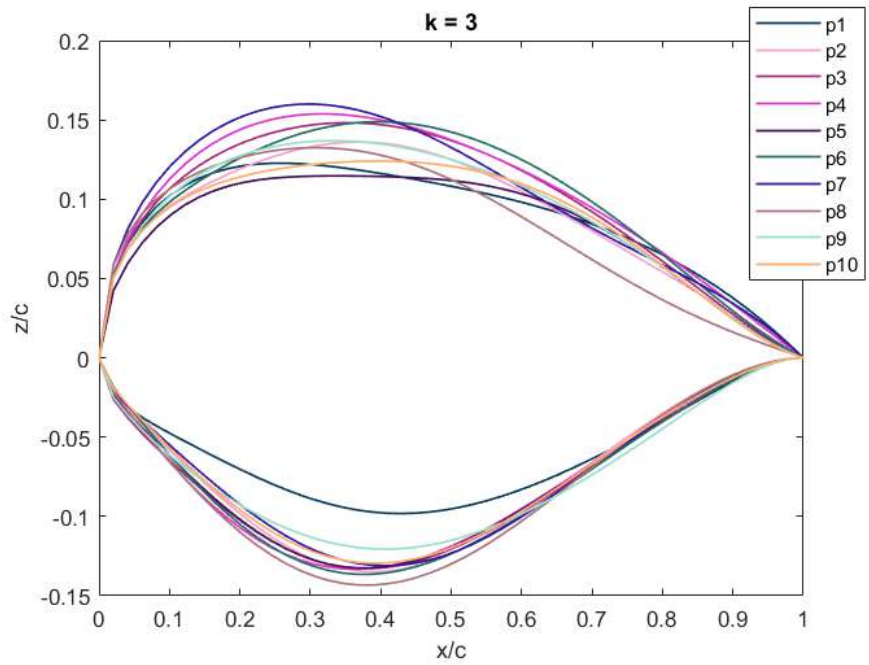


Figura 71. Geometrías de las partículas en la iteración  $k = 3$ .

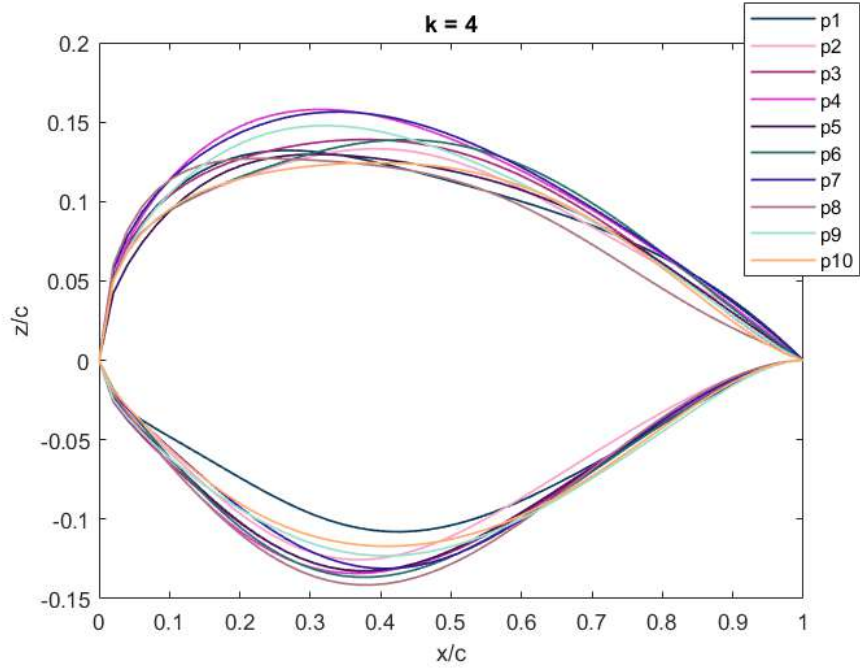


Figura 72. Geometrías de las partículas en la iteración  $k = 4$ .

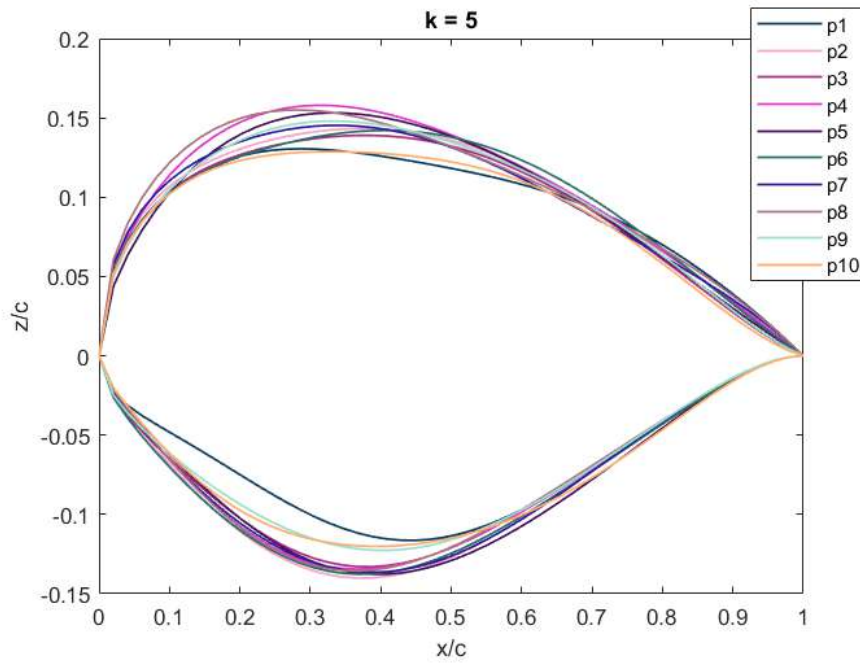


Figura 73. Geometrías de las partículas en la iteración  $k = 5$ .

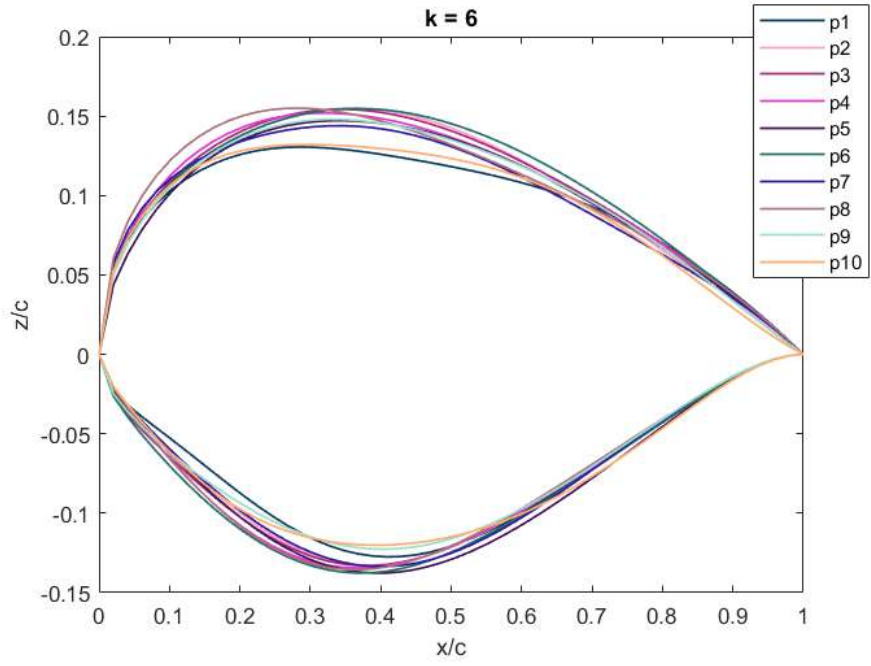


Figura 74. Geometrías de las partículas en la iteración  $k = 6$ .

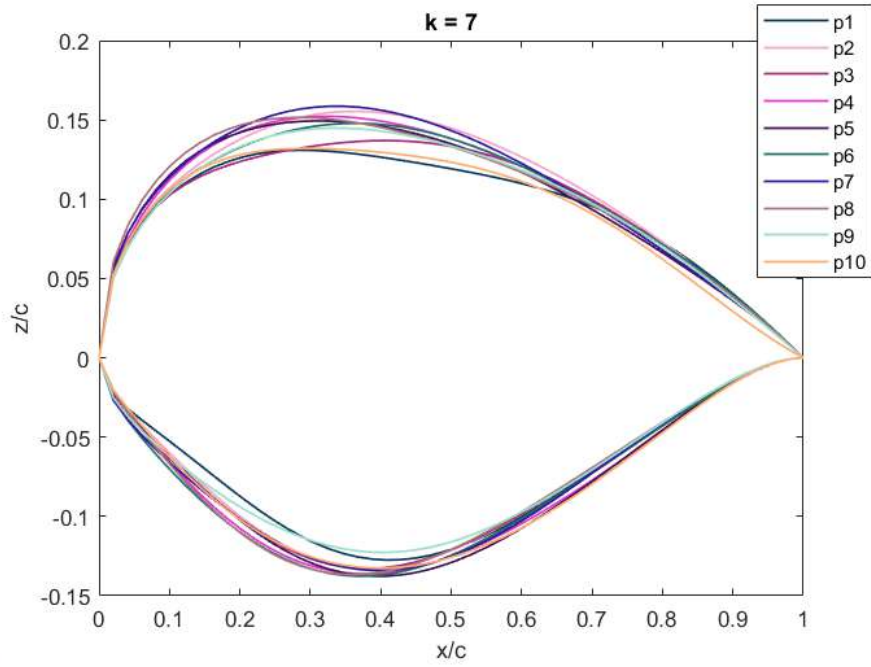


Figura 75. Geometrías de las partículas en la iteración  $k = 7$ .

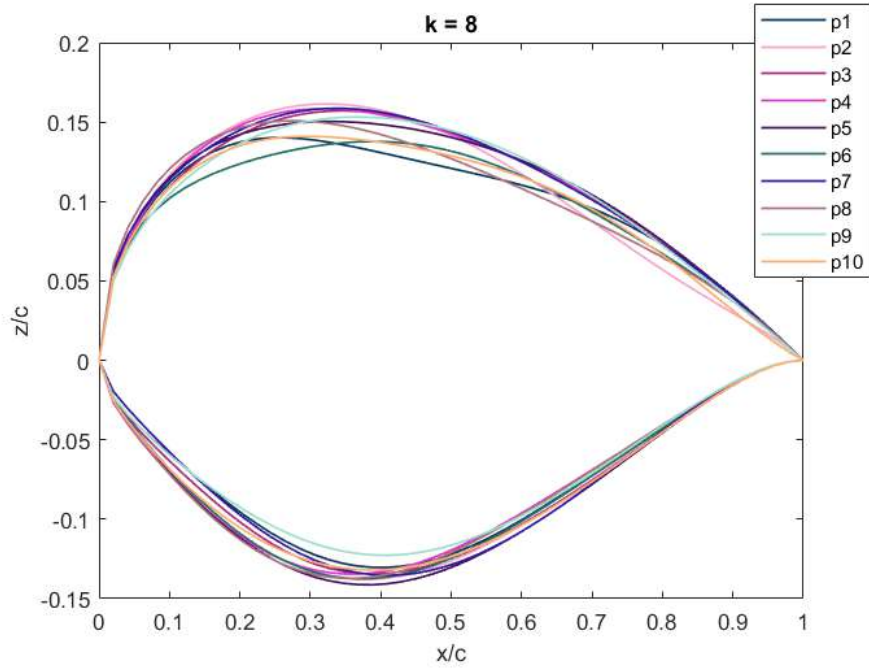


Figura 76. Geometrías de las partículas en la iteración  $k = 8$ .

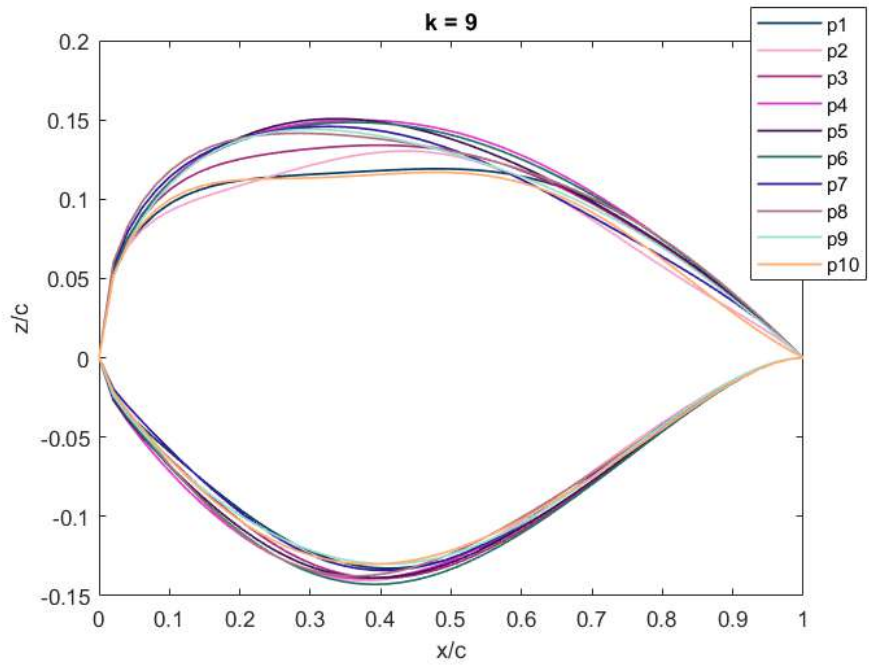


Figura 77. Geometrías de las partículas en la iteración  $k = 9$ .

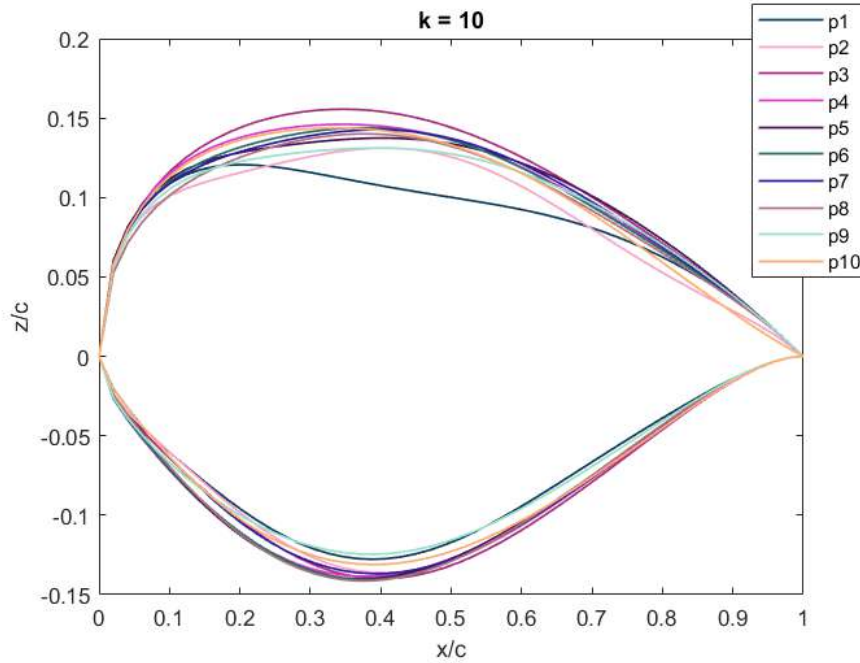


Figura 78. Geometrías de las partículas en la iteración  $k = 10$ .

En la Figura 79 se observa la comparación de las geometrías del perfil óptimo con el perfil S809, siendo el perfil óptimo el correspondiente a la partícula p4 (índice  $b = 4$ ) en el último periodo.

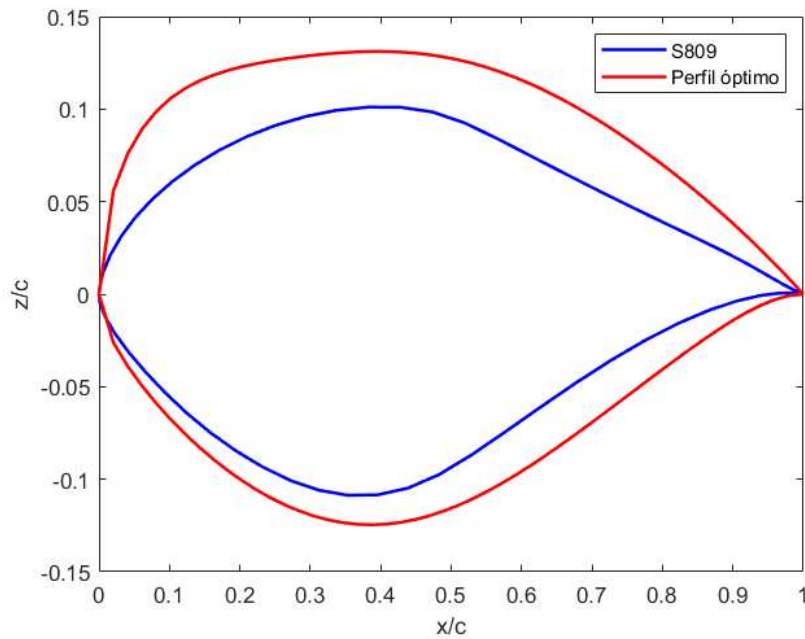


Figura 79. Comparación de la geometría del perfil óptimo con el perfil S809.

La gráfica del Valor Fitness de la Población (VFP) en la optimización se observa en la Figura 80, en donde el *Fitness* corresponde al valor de la partícula con el máximo coeficiente de sustentación en cada periodo. El periodo 0 corresponde a las partículas de la población inicial. En la gráfica del VFP se muestra que el *Fitness* del perfil óptimo fue de  $C_L = 0.932$ , lo que significó un incremento del 6.029 % con relación al  $C_L$  del perfil S809.

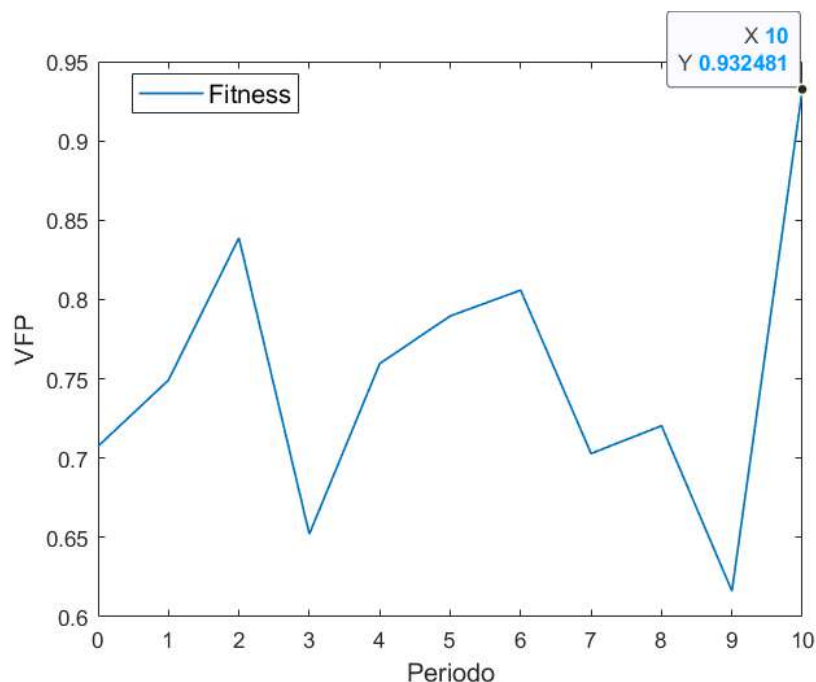


Figura 80. Gráfica del VFP a lo largo de las iteraciones del algoritmo PSO.

Sin embargo, es importante mencionar que de acuerdo con el VFP de la Figura 80, se notó un comportamiento irregular del algoritmo PSO, debido a que del periodo 2 al 3 hubo una disminución del valor *Fitness*, así como en otros intervalos del periodo, lo cual no resultó congruente con el comportamiento típico del algoritmo PSO de mantener en la iteración actual la mejor partícula de la iteración anterior en caso de que en la iteración actual no se encontrara una partícula mejor.

### 4.3 VALIDACIÓN DEL PERFIL ÓPTIMO MEDIANTE CURVAS DE SUSTENTACIÓN

Se realizó el análisis CFD del perfil óptimo obtenido, por lo que en la Figura 81 se muestra la comparación de la curva de sustentación del perfil óptimo (curva verde) con la

curva de sustentación del perfil S809 (curva roja). El tiempo de simulación de este análisis fue de  $t_s = 145.80 \text{ min} = 2.43 \text{ hr}$ , para 31 corridas del perfil aerodinámico.

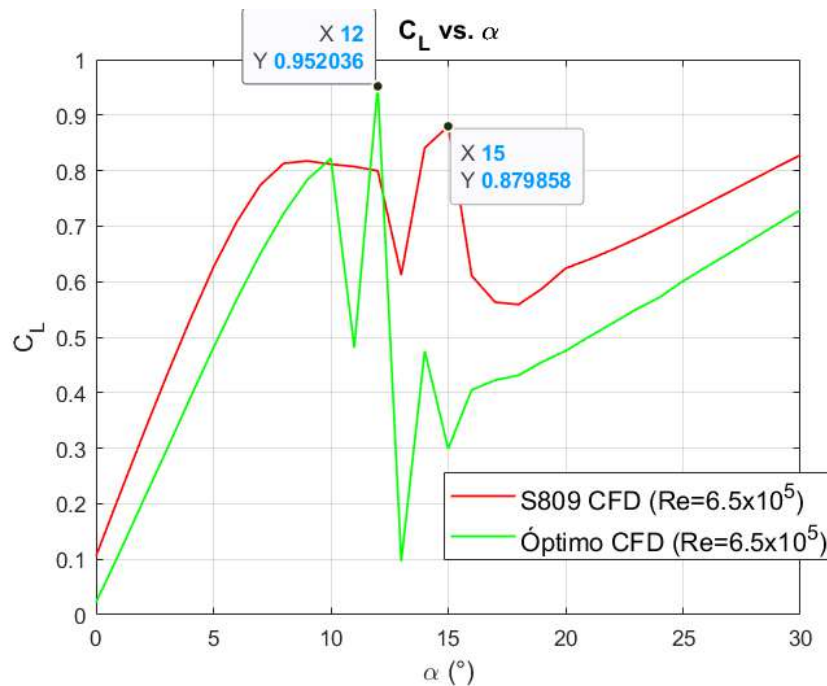


Figura 81. Comparación de las curvas de sustentación del perfil óptimo con el perfil S809.

Los resultados de la gráfica en la Figura 81 muestran algunas incongruencias con la curva de sustentación del perfil óptimo (verde): 1) el valor  $C_{L,max} = 0.952$  no corresponde al valor obtenido del *Fitness* indicado en la Figura 80, además de que no corresponde al ángulo de pérdida de  $\alpha_s = 15^\circ$  que se estableció para la optimización, 2) se observaron irregularidades de los valores  $C_L$  correspondientes a los ángulos de ataque de  $11^\circ$ ,  $13^\circ$  y  $15^\circ$ , siendo el más llamativo el que corresponde al ángulo  $\alpha = 11^\circ$  ya que se presentó antes del régimen de máxima sustentación, y 3) el comportamiento general de la curva no mejoró debido a que el  $C_L$  fue menor en casi todos los puntos de los regímenes de trabajo.

#### 4.4 ANÁLISIS DE PARÁMETROS Y PROBLEMAS DE CONVERGENCIA

De acuerdo con los resultados obtenidos y con la observación de los análisis del diseño del experimento, se encontró que las irregularidades de las curvas de sustentación mostradas en las Figuras 68 y 81, se generaron debido a que las soluciones de los análisis

numéricos realizados a esos AoA no convergieron dentro del número de iteraciones establecido en *Fluent*, lo que generó una variación de los valores de  $C_L$ . Lo mismo se sugiere respecto a la gráfica del VFP de la Figura 80, ya que, al no converger las soluciones de los análisis cuando se evaluó la función objetivo de las partículas de los nuevos periodos, se registraron valores diferentes del *Fitness*.

#### 4.4.1 Redefinición de parámetros del modelo de turbulencia CFD

Para solucionar esta problemática, Menter & Sechner (2021) proponen modificar un parámetro empleado en el cálculo de la viscosidad turbulenta dentro de las ecuaciones de transporte del modelo de turbulencia SST k- $\omega$ , con la finalidad de corregir los efectos indeseados de separación del flujo cercanos a las fronteras de la superficie del perfil, siendo esto lo que genera los resultados no deseables de poca convergencia. En la ecuación (62) se muestra el cálculo de la viscosidad turbulenta.

$$\mu_t = \rho \nu_t = \rho \frac{k}{\max\left(\omega, \frac{F_2 S}{a_1}\right)} \quad (62)$$

La sugerencia de los autores es cambiar el coeficiente  $a_1$  predefinido en 0.31 a un valor dentro del rango de  $0.31 \leq a_1 \leq 1.0$ , por lo que después de una serie de pruebas con el análisis numérico del perfil, se estableció un valor de  $a_1 = 0.7$ . Para mayor detalle de los parámetros involucrados en el cálculo de la viscosidad turbulenta se sugiere revisar el reporte de Menter & Sechner (2021).

#### 4.4.2 Resolución del diseño del experimento

Con el ajuste del coeficiente  $a_1$  en el *setup* dentro del componente *Fluent*, se realizó nuevamente el diseño del experimento de la Figura 35, es decir, se realizaron los tres tipos de análisis numéricos descritos en la metodología, por lo que los resultados finales se muestran a continuación.



#### 4.4.2.1 Curva de sustentación del perfil S809

En la Figura 82 se observa la comparación de la curva de sustentación de los datos de la CSU (curva azul) con la curva de sustentación del análisis CFD (curva roja). Con los ajustes realizados, para el análisis CFD se obtuvo un  $C_{L,max} = 0.928$  a un  $\alpha_s = 10^\circ$ , siendo el mismo valor del  $C_{L,max}$  de los datos experimentales, pero a un ángulo de pérdida menor. El tiempo de simulación de este análisis fue de  $t_s = 105.79 \text{ min} = 1.76 \text{ hr}$ , para 31 corridas del perfil aerodinámico.

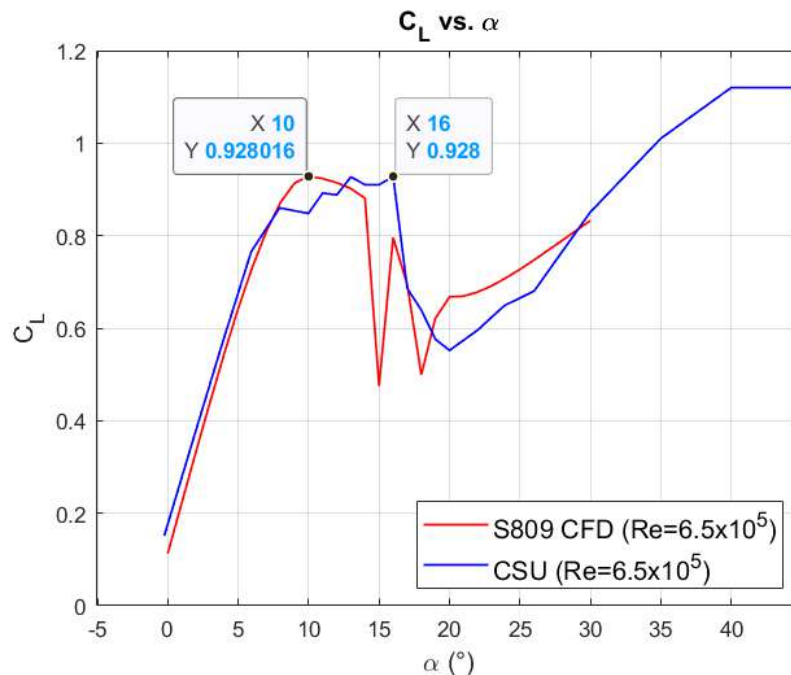


Figura 82. Gráfica de la comparación de la curva de sustentación del perfil S809 con datos experimentales.

Lo que resalta de los resultados de este análisis es que no se encontraron irregularidades en la curva antes del régimen de máxima sustentación.

#### 4.4.2.2 Optimización del perfil aerodinámico

La nueva optimización del perfil se realizó a un ángulo de ataque fijo de  $\alpha = 10^\circ$ . En la Figura 83 se observan las geometrías de las partículas de la población inicial, mientras que en las Figuras 84-93 se muestran las geometrías de las partículas en cada uno de los 10 periodos. El tiempo de simulación de este análisis fue de  $t_s = 435.81 \text{ min} = 7.26 \text{ hr}$ , para 110 corridas del perfil aerodinámico.

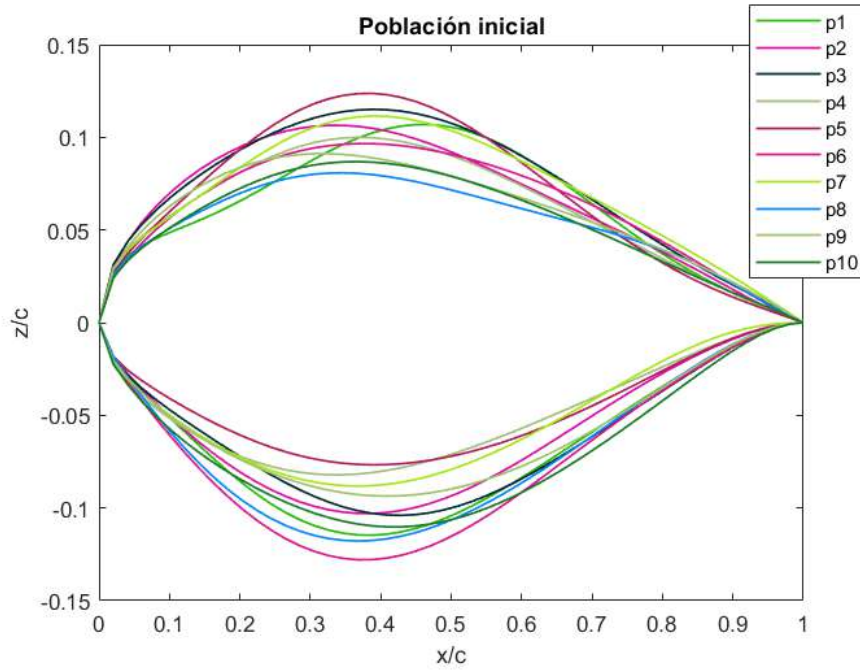


Figura 83. Geometrías de las partículas de la población inicial del enjambre.

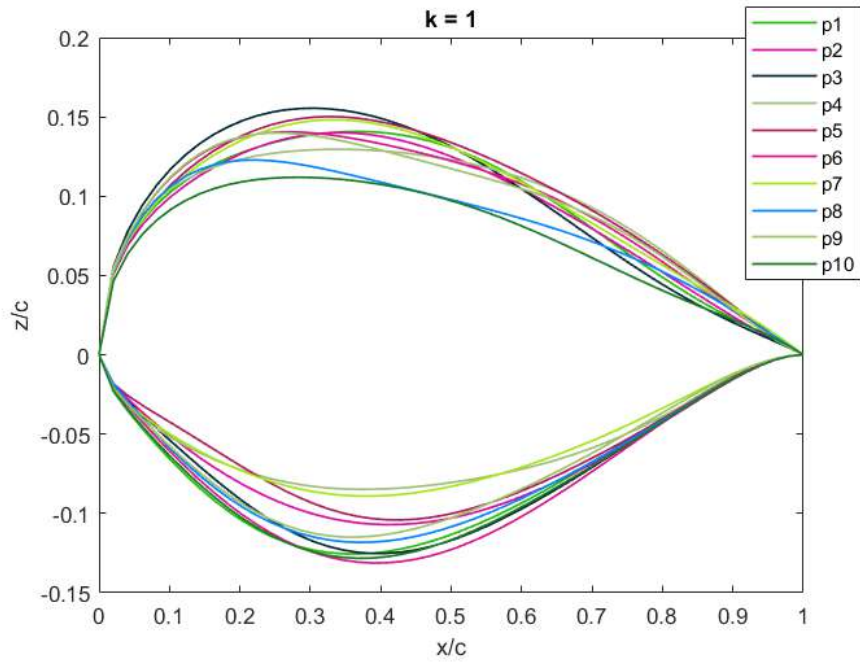


Figura 84. Geometrías de las partículas en la iteración  $k = 1$ .

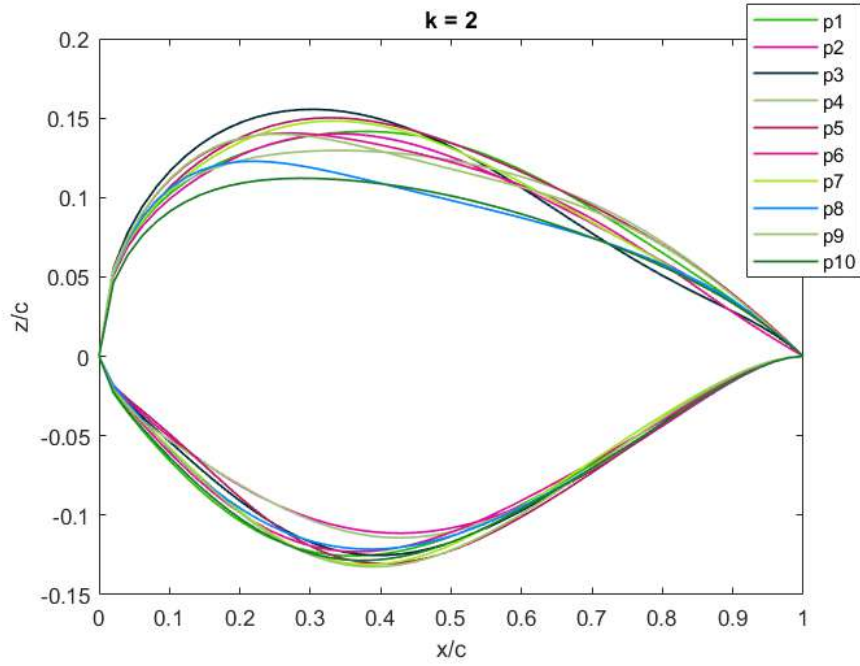


Figura 85. Geometrías de las partículas en la iteración  $k = 2$ .

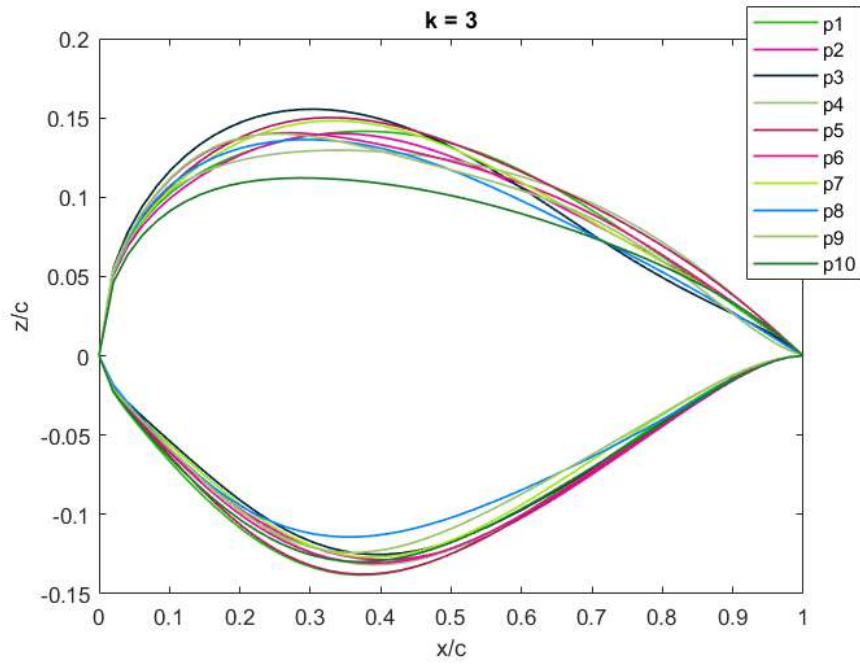


Figura 86. Geometrías de las partículas en la iteración  $k = 3$ .

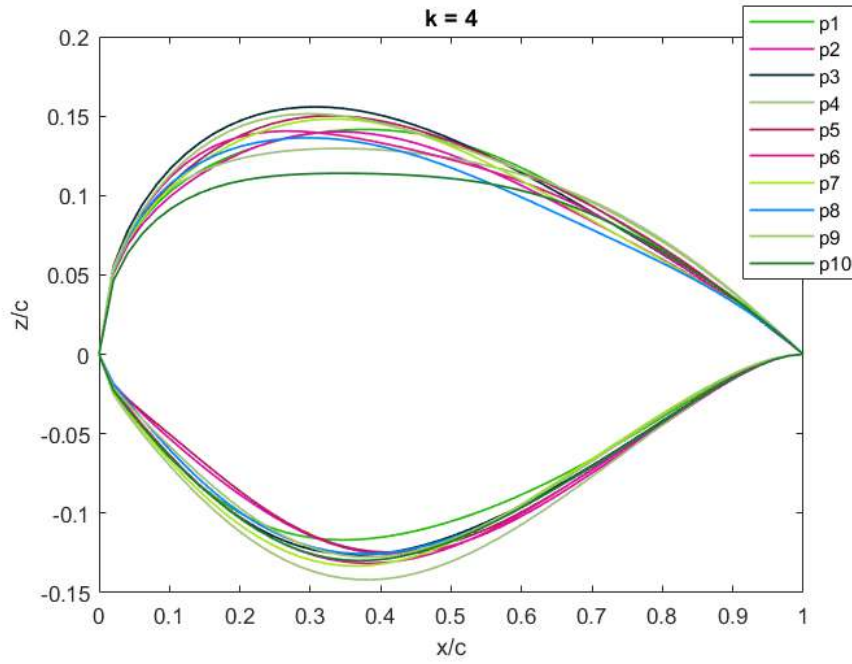


Figura 87. Geometrías de las partículas en la iteración  $k = 4$ .

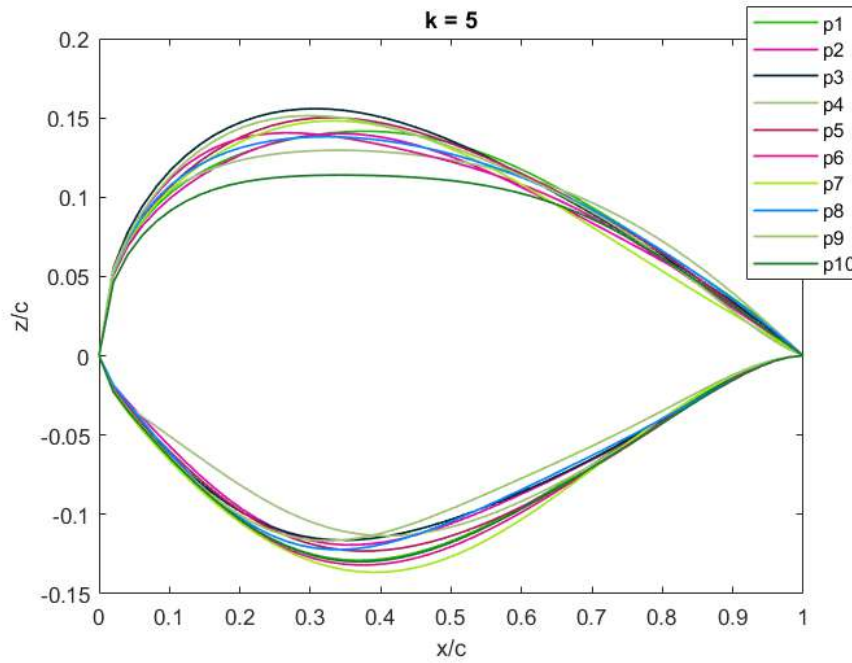


Figura 88. Geometrías de las partículas en la iteración  $k = 5$ .

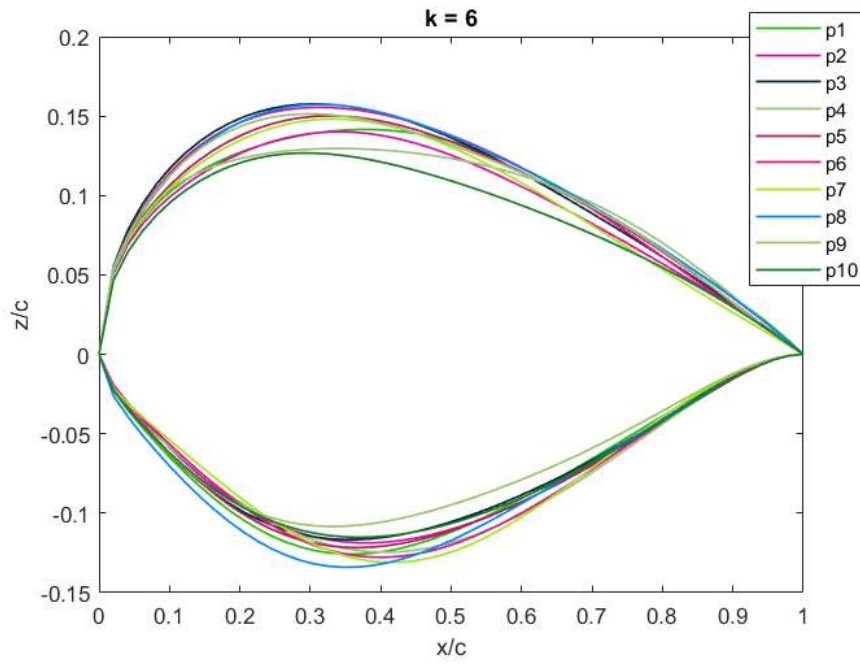


Figura 89. Geometrías de las partículas en la iteración  $k = 6$ .

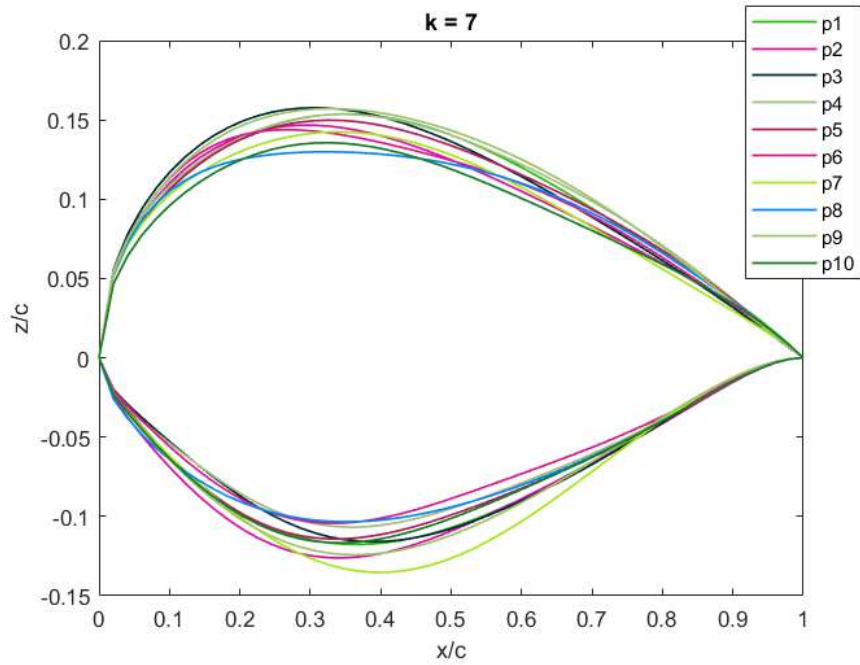


Figura 90. Geometrías de las partículas en la iteración  $k = 7$ .

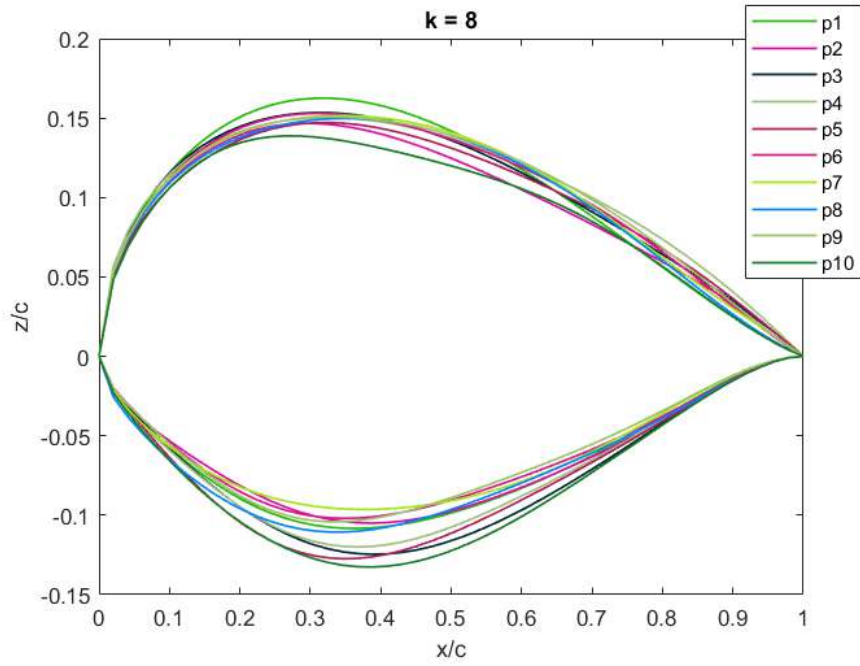


Figura 91. Geometrías de las partículas en la iteración  $k = 8$ .

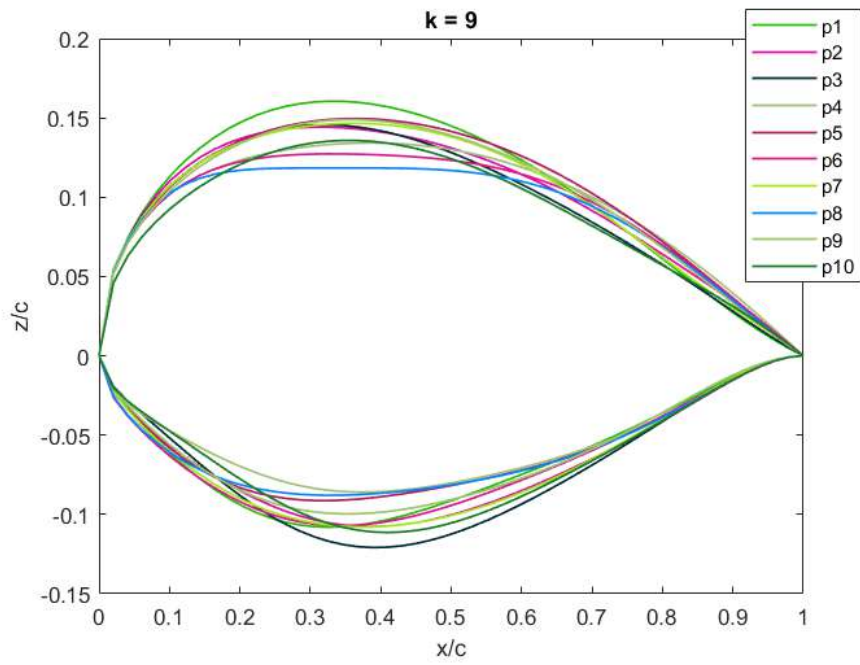


Figura 92. Geometrías de las partículas en la iteración  $k = 9$ .

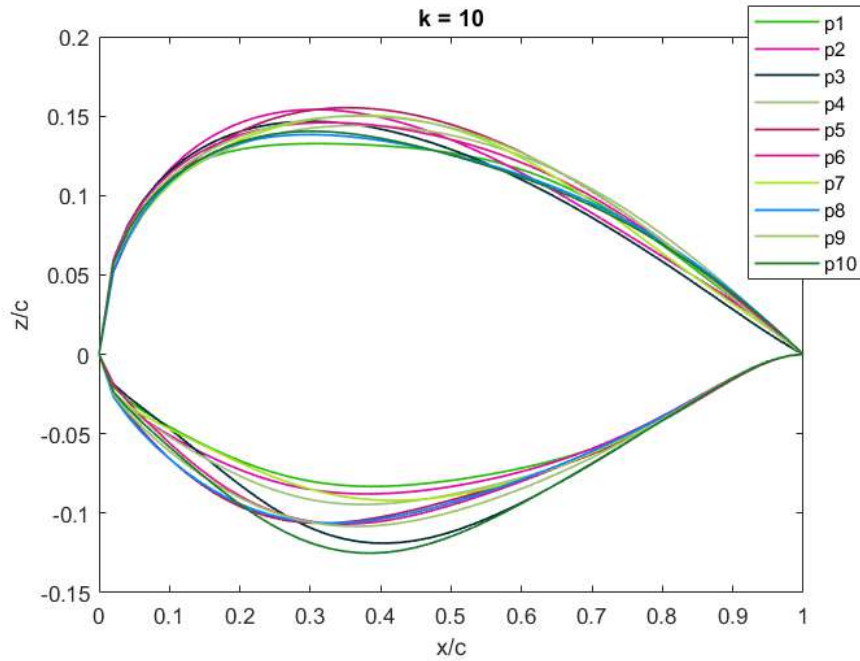


Figura 93. Geometrías de las partículas en la iteración  $k = 10$ .

En la Figura 94 se observa la comparación de las geometrías del perfil óptimo con el perfil S809, siendo el perfil óptimo el correspondiente a la partícula p2 (índice  $b = 2$ ) en el último periodo.

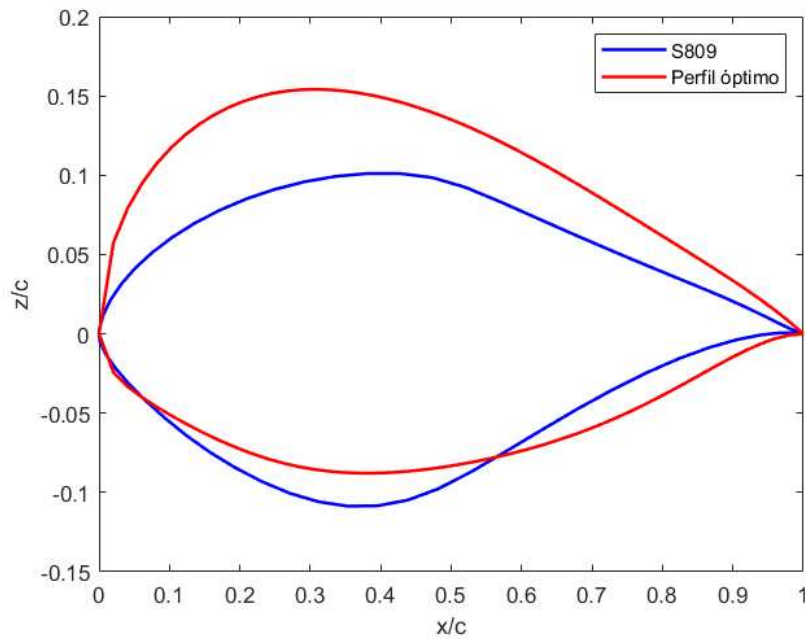


Figura 94. Comparación de la geometría del perfil óptimo con el perfil S809.

En la Figura 95 se muestra la gráfica del VFP de la segunda corrida del algoritmo de optimización. En la gráfica se observa que el comportamiento del algoritmo fue congruente, y que el valor *Fitness* superó el  $C_{L,max}$  del perfil S809 desde el periodo 1 con un valor de  $C_L = 0.953$ , manteniendo el mismo valor hasta el periodo 8, y posteriormente en los periodos 9 y 10 incrementó con valores de  $C_L = 0.954$  y  $C_L = 0.962$ , respectivamente.

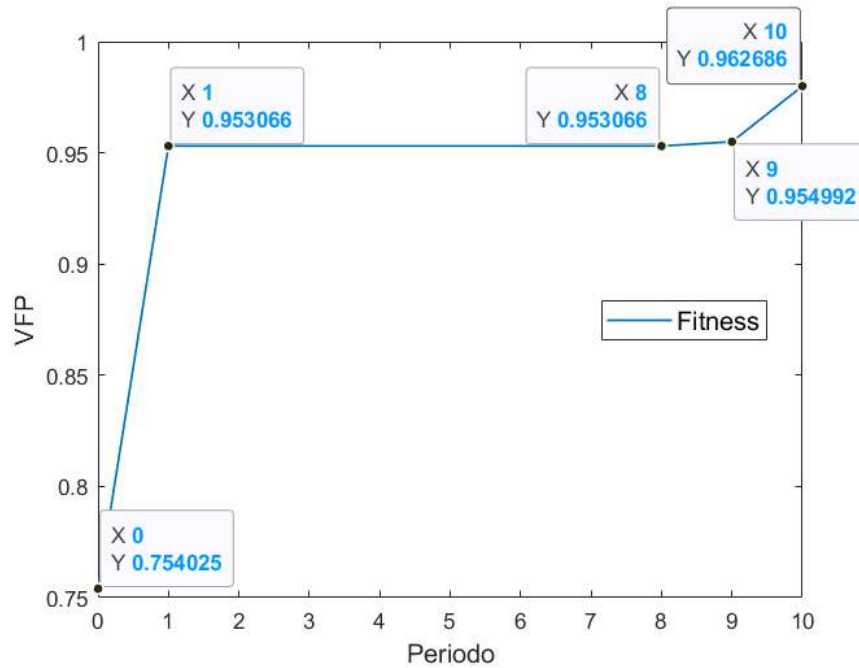


Figura 95. Gráfica del VFP a lo largo de las iteraciones del algoritmo PSO.

De igual manera, en la Figura 96 se muestra la gráfica del VFP promedio de la segunda corrida del algoritmo PSO. En la gráfica se observa un comportamiento creciente del valor promedio del VFP, lo cual representó el comportamiento esperado del algoritmo.



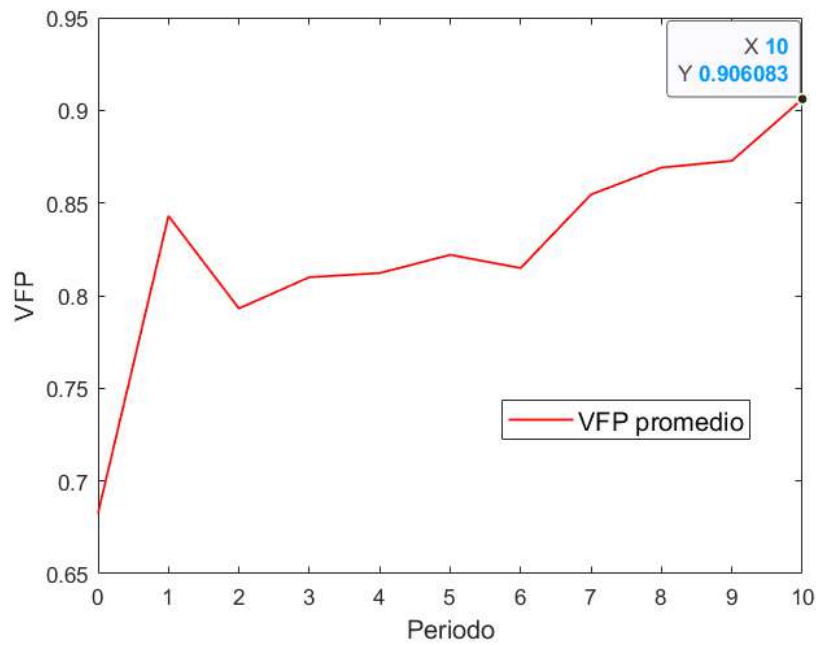


Figura 96. VFP promedio del algoritmo PSO.

#### 4.4.2.3 Validación del perfil óptimo mediante curvas de sustentación

En la Figura 97 se muestra la comparación de las curvas de sustentación del perfil óptimo (curva verde) de la segunda corrida con el perfil S809 (curva roja). En la gráfica se observa que en el ángulo  $\alpha = 10^\circ$ , el perfil óptimo obtuvo un  $C_L = 0.962$ , lo que correspondió a un incremento del 3.736 % respecto al  $C_L$  del perfil S809. El tiempo de simulación de este análisis fue de  $t_s = 84.74 \text{ min} = 1.41 \text{ hr}$ , para 31 corridas del perfil aerodinámico.

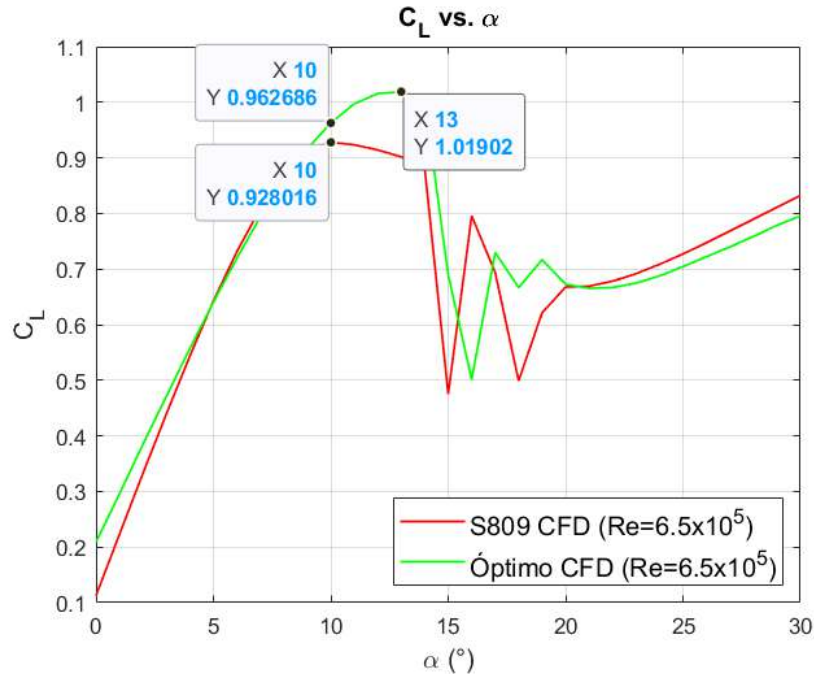


Figura 97. Comparación de las curvas de sustentación del perfil óptimo con el perfil S809.

Por otra parte, se observó que el  $C_{L,max}$  del perfil óptimo en realidad se obtuvo en el ángulo de pérdida de  $\alpha_s = 13^\circ$ , con un valor de  $C_{L,max} = 1.019$ , lo que correspondió a un incremento del 9.8 % respecto al  $C_{L,max}$  del perfil S809, además, no se obtuvieron problemas de convergencia antes del régimen de máxima sustentación, sino hasta el régimen de estancamiento total cuando se experimentan los fenómenos de turbulencia mayor.

Como última observación de los resultados, la curva del perfil óptimo obtuvo valores de  $C_L$  mayores que la curva del perfil S809 tanto en el régimen de flujo adjunto como en el régimen de máxima sustentación, con excepción del rango de ángulos de ataque de  $5^\circ \leq \alpha \leq 9^\circ$ , mientras que en el régimen de estancamiento total se presentaron irregularidades a partir de  $\alpha = 15^\circ$  hasta llegar al ángulo de  $\alpha = 20^\circ$ .

#### 4.4.3 Contornos de velocidad y presión de los perfiles aerodinámicos

En las Figuras 98 y 99 se muestran los contornos de velocidad y presión del perfil S809 en su régimen de máxima sustentación ( $\alpha = 10^\circ$ ), respectivamente. Por otra parte, en

las Figuras 100 y 101 se muestran los contornos de velocidad y presión del perfil óptimo al mismo ángulo de ataque  $\alpha = 10^\circ$ , respectivamente.

La comparación de contornos de velocidad indicó que el perfil óptimo tiene una velocidad del fluido mayor que el perfil S809 en las zonas del borde de ataque de la superficie superior y en las capas de fluido justo después del borde de salida, lo cual se observa con la escala de magnitud. De igual manera, los contornos de presión indicaron que el perfil óptimo tiene una caída de presión mayor que el perfil S809 en la zona del borde de ataque de la superficie superior.

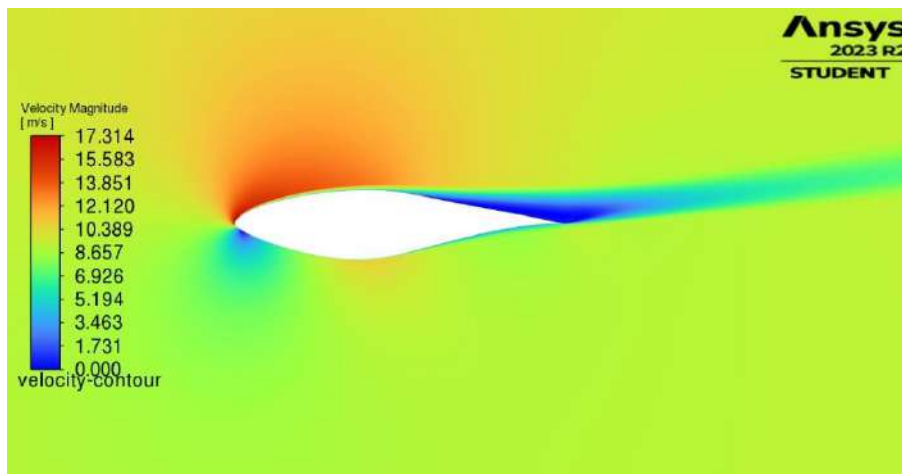


Figura 98. Contornos de velocidad del perfil S809 a un  $\alpha = 10^\circ$ .

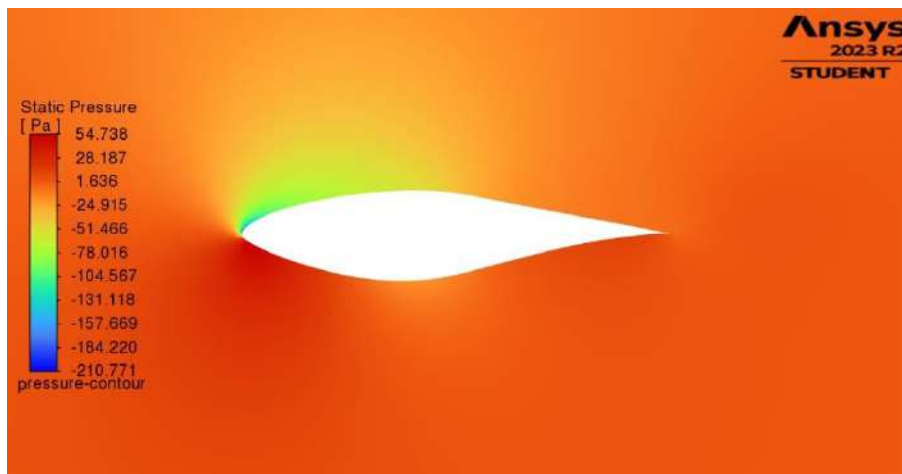


Figura 99. Contornos de presión del perfil S809 a un  $\alpha = 10^\circ$ .

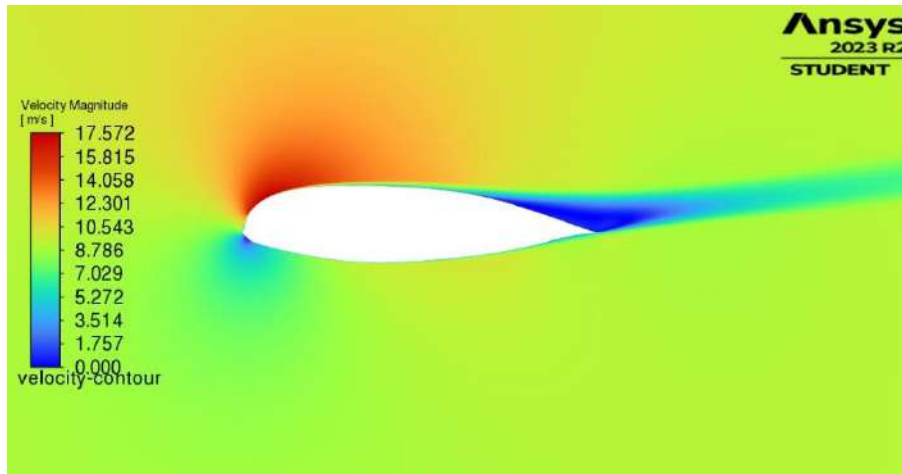


Figura 100. Contornos de velocidad del perfil óptimo a un  $\alpha = 10^\circ$ .

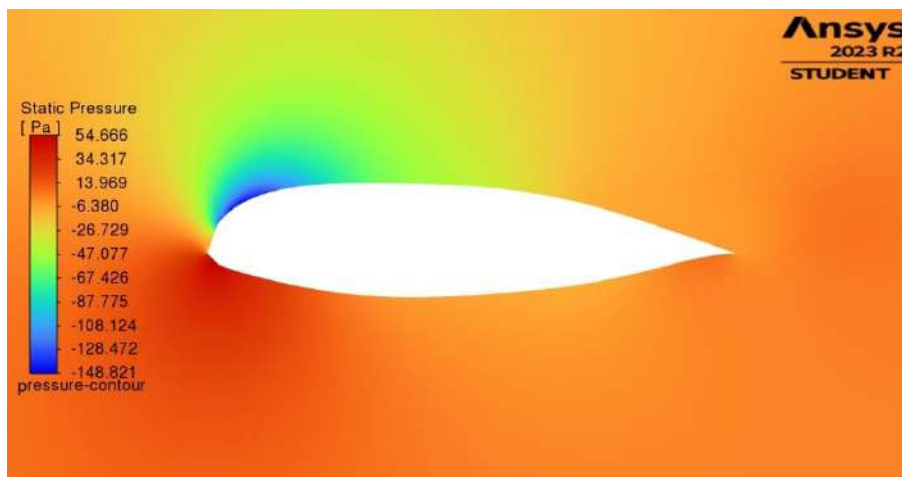


Figura 101. Contornos de presión del perfil óptimo a un  $\alpha = 10^\circ$ .

Por otra parte, en la Figura 102 y 103 se muestran los contornos de velocidad y presión del perfil óptimo en su régimen de máxima sustentación ( $\alpha = 13^\circ$ ), respectivamente. Estos resultados mostraron claramente que el perfil óptimo en su régimen de máxima sustentación tiene un comportamiento similar al de  $\alpha = 10^\circ$  con respecto a la distribución de las capas del fluido cercanas a la superficie y justo detrás del borde de salida, pero con velocidades y caídas de presión mayores.

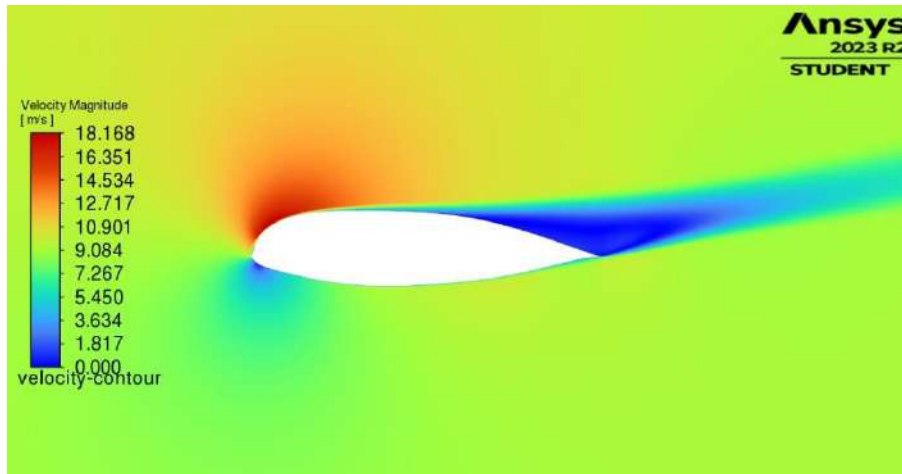


Figura 102. Contornos de velocidad del perfil óptimo a un  $\alpha = 13^\circ$ .

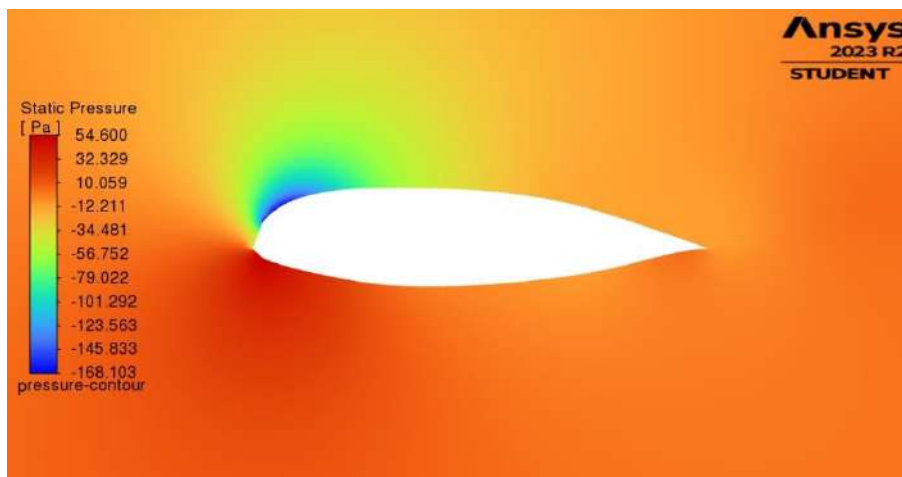


Figura 103. Contornos de presión del perfil óptimo a un  $\alpha = 13^\circ$ .

Para mostrar el comportamiento de los perfiles a un AoA después del régimen de máxima sustentación se definió el ángulo  $\alpha = 18^\circ$ . En las Figuras 104 y 105 se muestran los contornos de velocidad y presión del perfil S809 que corresponden a  $\alpha = 18^\circ$ , respectivamente. Por otra parte, en las Figuras 106 y 107 se muestran los contornos de velocidad y presión del perfil óptimo al mismo ángulo de ataque  $\alpha = 18^\circ$ , respectivamente.

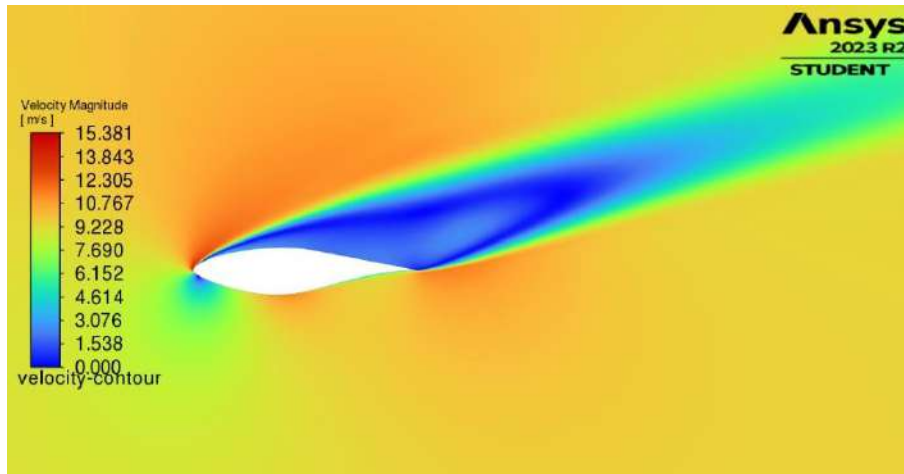


Figura 104. Contornos de velocidad del perfil S809 a un  $\alpha = 18^\circ$ .

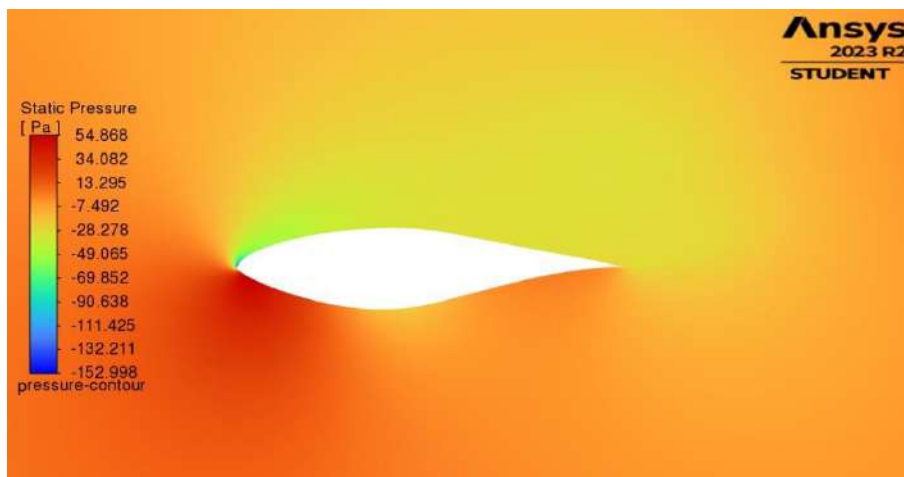


Figura 105. Contornos de presión del perfil S809 a un  $\alpha = 18^\circ$ .

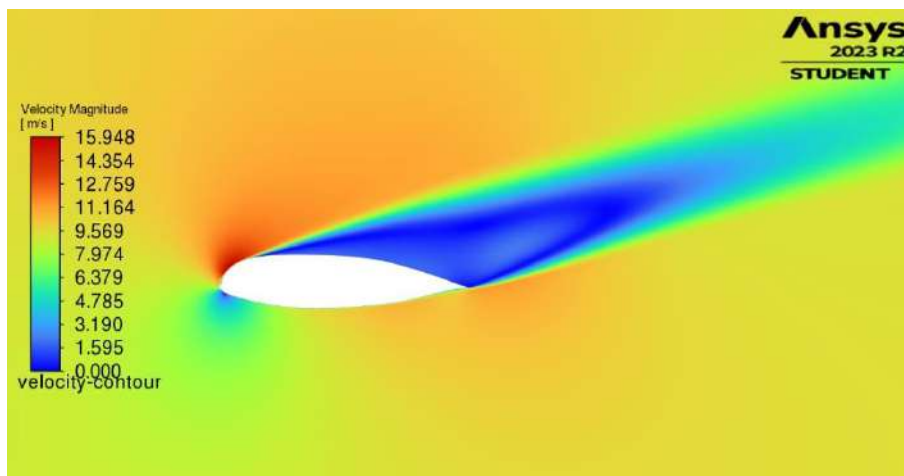


Figura 106. Contornos de velocidad del perfil óptimo a un  $\alpha = 18^\circ$ .

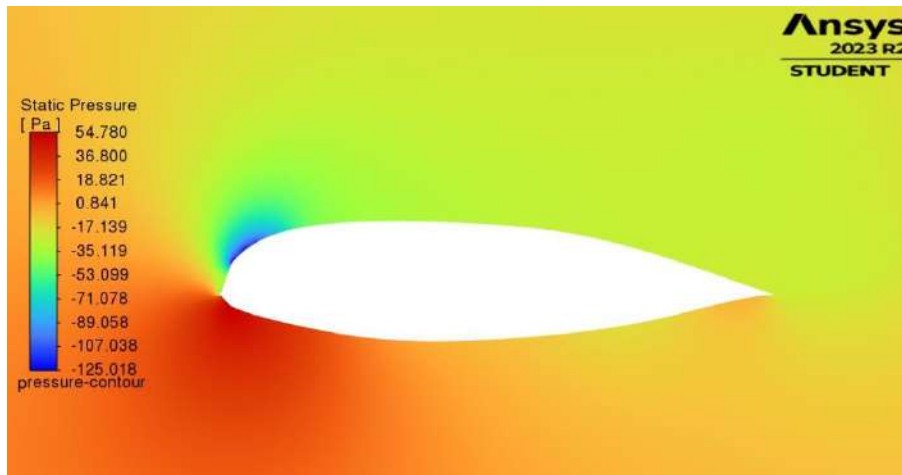


Figura 107. Contornos de presión del perfil óptimo a un  $\alpha = 18^\circ$ .

Esta comparación de los contornos de velocidad en la zona de estancamiento total indicaron que ambos perfiles experimentan turbulencia sobre su superficie superior, lo cual se observa con los contornos de distintas velocidades en la estela formada después del borde de salida, sin embargo, la comparación de los contornos de presión mostraron que el perfil óptimo presenta un caída de presión mayor en el borde de ataque de su superficie superior respecto al resto de su superficie, a diferencia del perfil S809 que presenta una distribución de presión más uniforme sobre su superficie superior, siendo un factor importante en la formación de la capa límite del fluido sobre el perfil.

## V. CONCLUSIONES Y PROSPECTIVAS

De acuerdo con los resultados obtenidos en este estudio, se comprobó que mediante la metodología propuesta basada en el marco de trabajo POA, fue posible optimizar el perfil aerodinámico de la pala de una turbina eólica de baja potencia debido a que se obtuvo un resultado favorecedor en la maximización del coeficiente de sustentación del perfil óptimo a un ángulo  $\alpha = 10^\circ$ , siendo un 3.736 % mayor que el coeficiente del perfil de referencia en su régimen de máxima sustentación. Por otra parte, se obtuvo un incremento del 9.8 % del  $C_{L,max}$  del perfil óptimo respecto al  $C_{L,max}$  del perfil de referencia. Adicionalmente, se observó que optimizar un perfil aerodinámico para un ángulo de ataque específico permitió obtener un comportamiento general mejor que el perfil de referencia, no solamente en el régimen de máxima sustentación, sino en la mayoría de los regímenes de trabajo, lo cual permitiría obtener el diseño de turbina eólica con mayor potencia de salida al emplear el perfil óptimo. De igual manera se concluyó que el uso de métodos CFD para el análisis aerodinámico de perfiles, específicamente en el cálculo del coeficiente de sustentación, es un procedimiento efectivo que permite el diseño de perfiles aerodinámicos y turbinas eólicas desde un aspecto puramente computacional, sin embargo, presenta ventajas importantes respecto al ahorro de tiempo y recursos que, mediante otros procedimientos de diseño, son difíciles de conseguir. Se concluyó también que el método de parametrización utilizado resultó conveniente al emplearse con el algoritmo de optimización, debido a que la definición de las variables de diseño con el método CST se ajustó con el algoritmo PSO para realizar una búsqueda efectiva de la solución en el espacio de diseño del problema. Finalmente, se concluyó que emplear el marco de trabajo POA con las herramientas utilizadas en el diseño del experimento permitió implementar de manera efectiva el acoplamiento del optimizador con el analizador para cumplir con los objetivos de este estudio.

Existen varias prospectivas que se consideran en este trabajo y que son objeto de estudio para investigaciones futuras, las cuales se relacionan con los aspectos a mejorar dentro del marco de trabajo POA implementado y son enlistadas a continuación.

- Realizar un estudio de cómo influye la relación del número de variables de diseño obtenidas con el método CST y el número de partículas definidas en el algoritmo PSO con las características de exploración y explotación en la



optimización, así como con el tiempo en que se obtiene la solución óptima al problema.

- Realizar un estudio comparativo del algoritmo PSO empleando diferentes métodos para obtener el coeficiente del peso de inercia y cómo estos afectan en la obtención de la solución óptima del problema.
- Realizar un estudio comparativo del algoritmo PSO empleando diferentes números de partículas y cómo estos afectan el nivel de convergencia en la obtención de la solución óptima de acuerdo con la gráfica del VFP promedio.
- Realizar un estudio de sensibilidad de la malla del modelo computacional para saber su influencia en la exactitud de los resultados del análisis numérico CFD y si también influye en las incongruencias derivadas de las dificultades de convergencia en algunos ángulos de ataque específicos del análisis.
- Realizar un estudio comparativo del análisis numérico CFD con diferentes modelos de turbulencia tipo RANS, así como con métodos CFD de tipo LES y DES.
- Realizar el diseño de la pala de una turbina eólica de eje horizontal de baja potencia utilizando el perfil óptimo para comprobar la mejora del rotor con relación a su potencia de salida.

## VI. ANEXOS

### 6.1 ANEXO 1. PSEUDOCÓDIGO DEL ALGORITMO PSO.

#### 1) Inicializar la población

Inicializar parámetros de control  $\omega_{max}$ ,  $\omega_{min}$ ,  $c_1$ ,  $c_2$ ,  $e_1$ ,  $e_2$ ,  $T$

Inicializar posiciones  $x_i^{t=0}$  y velocidades  $v_i^{t=0}$  de las  $n$  partículas de la primera población

Establecer la función objetivo  $f(x_i)$  donde  $x_i = (x_{i,1}, \dots, x_{i,d})^{t=0}$  para  $i = 1, 2, \dots, n$

Calcular el fitness  $F_b^{t=0} = \max \{f(x_i^{t=0})\}$  y encontrar el índice de la mejor partícula  $b^{t=0}$

Definir las posiciones iniciales  $Pbest_i^{t=0} = x_i^{t=0}$  y  $Gbest^{t=0} = x_b^{t=0}$

Establecer  $t = 1$

for  $k = t:T$

  for cada  $i$  partícula en las  $d$  dimensiones

    Calcular  $\omega = \omega_{max} - k(\omega_{max} - \omega_{min})/T$

    Calcular las nuevas velocidades  $v_i^{k+1}$

    Calcular las nuevas posiciones  $x_i^{k+1}$

    Calcular  $F_b^{k+1} = \max \{f(x_i^{k+1})\}$

    Encontrar el índice de la mejor partícula  $b^{k+1}$

#### 2) Actualizar la posición del $Pbest$

  if  $f(x_i^{k+1}) > f(x_i^k)$

$Pbest_i^{k+1} = x_i^{k+1}$

  else

$Pbest_i^{k+1} = Pbest_i^k$

#### 3) Obtener la posición del $Gbest$

  if  $F_b^{k+1} > F_b^k$

$Gbest^{k+1} = Pbest_b^{k+1}$

  else

$Gbest^{k+1} = Gbest^k$

$F_b^{k+1} = F_b^k$

  end for

Recalcular  $e_1$ ,  $e_2$

end for

Obtener la solución final  $x_i^*$  de  $Gbest^T$  y el valor del fitness  $F_b^T$

## 6.2 ANEXO 2. COORDENADAS DEL PERFIL AERODINÁMICO S809.

Superficie superior		Superficie inferior	
x/c	z/c	x/c	z/c
0.00000	0.00000	0.00000	0.00000
0.00037	0.00275	0.00140	-0.00498
0.00575	0.01166	0.00933	-0.01272
0.01626	0.02133	0.02321	-0.02162
0.03158	0.03136	0.04223	-0.03144
0.05147	0.04143	0.06579	-0.04199
0.07568	0.05132	0.09325	-0.05301
0.10390	0.06082	0.12397	-0.06408
0.13580	0.06972	0.15752	-0.07467
0.17103	0.07786	0.19362	-0.08447
0.20920	0.08505	0.23175	-0.09326
0.24987	0.09113	0.27129	-0.10060
0.29259	0.09594	0.31188	-0.10589
0.33689	0.09933	0.35328	-0.10866
0.38223	0.10109	0.39541	-0.10842
0.42809	0.10101	0.43832	-0.10484
0.47384	0.09843	0.48234	-0.09756
0.52005	0.09237	0.52837	-0.08697
0.56801	0.08356	0.57663	-0.07442
0.61747	0.07379	0.62649	-0.06112
0.66718	0.06403	0.67710	-0.04792
0.71606	0.05462	0.72752	-0.03558
0.76314	0.04578	0.77668	-0.02466
0.80756	0.03761	0.82348	-0.01559
0.84854	0.03017	0.86677	-0.00859
0.88537	0.02335	0.90545	-0.00370

0.91763	0.01694	0.93852	-0.00075
0.94523	0.01101	0.96509	0.00054
0.96799	0.00600	0.98446	0.00065
0.98528	0.00245	0.99612	0.00024
0.99623	0.00054	1.00000	0.00000
1.00000	0.00000		

### 6.3 ANEXO 3. SCRIPT EN PYTHON PARA LA GEOMETRÍA DEL MODELO.

```
# Python Script, API Version = V232

# Insert From File
importOptions = ImportOptions.Create()
DocumentInsert.Execute(r"C:/Users/shado/Documentos/MATLAB/Tesis_Maestria/airfoils_coordinates/S809
_spaceClaim.txt", importOptions, GetMaps("bde7bd78"))
# EndBlock

# Solidify Sketch
mode = InteractionMode.Solid
result = ViewHelper.SetViewMode(mode, None)
# EndBlock

# Fill
selection = Selection.Create(GetRootPart().Curves[0])
secondarySelection = Selection.Empty()
options = FillOptions()
options.AutoExtendFillArea = False
options.PatchBlend = False
options.ZipLaminarEdges = False
options.GapAngle = DEG(10)
options.GapDistance = M(0.0001)
options.UntrimSingleFace = True
options.AllowMultiFacePatch = False
result = Fill.Execute(selection, secondarySelection, options, FillMode.ThreeD, None)
# EndBlock

# Set Sketch Plane
sectionPlane = Plane.Create(Frame.Create(Point.Create(M(0.5), M(-0.00378500000000001), M(0)),
    -Direction.DirX,
    Direction.DirY))
result = ViewHelper.SetSketchPlane(sectionPlane, None)
# EndBlock

# Sketch Line
start = Point2D.Create(M(0.5), M(0.00378500000000001))
end = Point2D.Create(M(0.4), M(0.00378500000000002))
result = SketchLine.Create(start, end)

curveSelList = Selection.Create(GetRootPart().DatumPlanes[0].Curves[0])
result = Constraint.CreateHorizontal(curveSelList)
# EndBlock

# Sketch Circle
origin = Point2D.Create(M(0.4), M(0.00378500000000002))
result = SketchCircle.Create(origin, M(20))

baseSel = SelectionPoint.Create(GetRootPart().DatumPlanes[0].Curves[1].GetChildren[ICurvePoint]() [0])
targetSel = SelectionPoint.Create(GetRootPart().DatumPlanes[0].Curves[0].GetChildren[ICurvePoint]() [1])

result = Constraint.CreateCoincident(baseSel, targetSel)
# EndBlock

# Sketch Rectangle
```

```

point1 = Point2D.Create(M(0.400000000000001),M(20.003785))
point2 = Point2D.Create(M(-39.6),M(20.003785))
point3 = Point2D.Create(M(-39.6),M(-19.996215))
result = SketchRectangle.Create(point1, point2, point3)

baseSel = SelectionPoint.Create(GetRootPart().DatumPlanes[0].Curves[2], 0)
targetSel = SelectionPoint.Create(GetRootPart().DatumPlanes[0].Curves[1], 1.5707963267949)

result = Constraint.CreateTangent(baseSel, targetSel)

baseSel = SelectionPoint.Create(GetRootPart().DatumPlanes[0].Curves[4], 40)
targetSel = SelectionPoint.Create(GetRootPart().DatumPlanes[0].Curves[1], 4.71238898038469)

result = Constraint.CreateTangent(baseSel, targetSel)
# EndBlock

# Trim Sketch Curve
curveSelPoint = SelectionPoint.Create(GetRootPart().DatumPlanes[0].Curves[1], 2.51602057272735)
options = TrimSketchCurveOptions()
options.InverseTrim = False
options.NormalBoxSelection = False
result = TrimSketchCurve.Execute(curveSelPoint, options)
# EndBlock

# Trim Sketch Curve
curveSelPoint = SelectionPoint.Create(GetRootPart().DatumPlanes[0].Curves[5], 33.779494009956)
options = TrimSketchCurveOptions()
options.InverseTrim = False
options.NormalBoxSelection = False
result = TrimSketchCurve.Execute(curveSelPoint, options)
# EndBlock

# Trim Sketch Curve
curveSelPoint = SelectionPoint.Create(GetRootPart().DatumPlanes[0].Curves[5], 10.085582863705)
options = TrimSketchCurveOptions()
options.InverseTrim = False
options.NormalBoxSelection = False
result = TrimSketchCurve.Execute(curveSelPoint, options)
# EndBlock

# Trim Sketch Curve
curveSelPoint = SelectionPoint.Create(GetRootPart().DatumPlanes[0].Curves[5], 20.0323475060545)
options = TrimSketchCurveOptions()
options.InverseTrim = False
options.NormalBoxSelection = False
result = TrimSketchCurve.Execute(curveSelPoint, options)
# EndBlock

# Trim Sketch Curve
curveSelPoint = SelectionPoint.Create(GetRootPart().DatumPlanes[0].Curves[5], 19.9841682504577)
options = TrimSketchCurveOptions()
options.InverseTrim = False
options.NormalBoxSelection = False
result = TrimSketchCurve.Execute(curveSelPoint, options)
# EndBlock

```

```

# Trim Sketch Curve
curveSelPoint = SelectionPoint.Create(GetRootPart().DatumPlanes[0].Curves[0], 0.08770411829245)
options = TrimSketchCurveOptions()
options.InverseTrim = False
options.NormalBoxSelection = False
result = TrimSketchCurve.Execute(curveSelPoint, options)
# EndBlock

# Solidify Sketch
mode = InteractionMode.Solid
result = ViewHelper.SetViewMode(mode, None)
# EndBlock

# Change Object Visibility
selection = Selection.Create(GetRootPart().DatumPlanes[0])
visibility = VisibilityType.Hide
inSelectedView = False
faceLevel = False
ViewHelper.SetObjectVisibility(selection, visibility, inSelectedView, faceLevel)
# EndBlock

# Intersect Bodies
targets = BodySelection.Create(GetRootPart().Bodies[1])
tools = BodySelection.Create(GetRootPart().Bodies[0])
options = MakeSolidsOptions()
result = Combine.Intersect(targets, tools, options)
# EndBlock

# Delete Selection
selection = FaceSelection.Create(GetRootPart().Bodies[0].Faces[0])
result = Delete.Execute(selection)
# EndBlock

# Delete Selection
selection = FaceSelection.Create(GetRootPart().Bodies[0].Faces[0])
result = Delete.Execute(selection)
# EndBlock

# Split Faces
options = SplitFaceOptions()
options.MakeCurves = False
selection = FaceSelection.Create(GetRootPart().Bodies[0].Faces[0])
point = Selection.Create(GetRootPart().Bodies[0].Edges[4].GetChildren[ICurvePoint]()[0]).Items[0].Position
result = SplitFace.ByParametric(selection, point, FaceSplitType.UV, options)
# EndBlock

# Split Faces
options = SplitFaceOptions()
options.MakeCurves = False
selection = FaceSelection.Create(GetRootPart().Bodies[0].Faces[1])
point = EdgeSelection.Create(GetRootPart().Bodies[0].Edges[1]).Items[0].EvalProportion(0).Point
result = SplitFace.ByParametric(selection, point, FaceSplitType.UV, options)
# EndBlock

# Split Faces
options = SplitFaceOptions()

```

```

options.MakeCurves = False
selection = FaceSelection.Create(GetRootPart().Bodies[0].Faces[0])
point = EdgeSelection.Create(GetRootPart().Bodies[0].Edges[5]).Items[0].EvalProportion(1).Point
result = SplitFace.ByParametric(selection, point, FaceSplitType.UV, options)
# EndBlock

# Create Named Selection Group
primarySelection = EdgeSelection.Create([GetRootPart().Bodies[0].Edges[8],
    GetRootPart().Bodies[0].Edges[7],
    GetRootPart().Bodies[0].Edges[1],
    GetRootPart().Bodies[0].Edges[2],
    GetRootPart().Bodies[0].Edges[5],
    GetRootPart().Bodies[0].Edges[4]])
secondarySelection = Selection.Empty()
result = NamedSelection.Create(primarySelection, secondarySelection)
# EndBlock

# Create Named Selection Group
primarySelection = EdgeSelection.Create([GetRootPart().Bodies[0].Edges[10],
    GetRootPart().Bodies[0].Edges[11]])
secondarySelection = Selection.Empty()
result = NamedSelection.Create(primarySelection, secondarySelection)
# EndBlock

# Create Named Selection Group
primarySelection = EdgeSelection.Create([GetRootPart().Bodies[0].Edges[13],
    GetRootPart().Bodies[0].Edges[14],
    GetRootPart().Bodies[0].Edges[16],
    GetRootPart().Bodies[0].Edges[17]])
secondarySelection = Selection.Empty()
result = NamedSelection.Create(primarySelection, secondarySelection)
# EndBlock

```



## 6.4 ANEXO 4. SCRIPT EN PYTHON PARA LA DISCRETIZACIÓN DEL MODELO.

"NOTE : All workflows will not be recorded, as recording is under development."

```
#region Context Menu Action
mesh_14 = Model.Mesh
sizing_30 = mesh_14.AddSizing()
#endregion
```

```
#region Details View Action
selection = ExtAPI.SelectionManager.CreateSelectionInfo(SelectionTypeEnum.GeometryEntities)
selection.Ids = [7, 16, 9]
sizing_30.Location = selection
#endregion
```

```
#region Details View Action
sizing_30.Type = SizingType.NumberOfDivisions
#endregion
```

```
#region Details View Action
sizing_30.NumberOfDivisions = 100
#endregion
```

```
#region Details View Action
sizing_30.CaptureCurvature = False
#endregion
```

```
#region Details View Action
sizing_30.Behavior = SizingBehavior.Hard
#endregion
```

```
#region Details View Action
sizing_30.BiasType = BiasType.o_o_ooo_ooooo
#endregion
```

```
#region Details View Action
sizing_30.BiasFactor = 500
#endregion
```

```
#region Details View Action
selection = ExtAPI.SelectionManager.CreateSelectionInfo(SelectionTypeEnum.GeometryEntities)
selection.Ids = [9]
sizing_30.ReverseBias = selection
#endregion
```

```
#region UI Action
sizing_30.Name = r""rear-length""
#endregion
```

```
#region Context Menu Action
sizing_33 = mesh_14.AddSizing()
#endregion
```

```
#region Details View Action
```

```
selection = ExtAPI.SelectionManager.CreateSelectionInfo(SelectionTypeEnum.GeometryEntities)
selection.Ids = [18, 8, 26, 11, 21, 5, 17]
sizing_33.Location = selection
#endregion
```

```
#region Details View Action
sizing_33.Type = SizingType.NumberOfDivisions
#endregion
```

```
#region Details View Action
sizing_33.NumberOfDivisions = 200
#endregion
```

```
#region Details View Action
sizing_33.CaptureCurvature = True
#endregion
```

```
#region Details View Action
sizing_33.CurvatureNormalAngle = Quantity(18, "deg")
#endregion
```

```
#region Details View Action
sizing_33.LocalMinimumSize = Quantity(5e-4, "m")
#endregion
```

```
#region Details View Action
sizing_33.BiasType = BiasType.o_o_ooo_ooooo
#endregion
```

```
#region Details View Action
sizing_33.BiasFactor = 500000
#endregion
```

```
#region Details View Action
selection = ExtAPI.SelectionManager.CreateSelectionInfo(SelectionTypeEnum.GeometryEntities)
selection.Ids = [11, 21, 5, 17]
sizing_33.ReverseBias = selection
#endregion
```

```
#region UI Action
sizing_33.Name = r""domain-height""
#endregion
```

```
#region Context Menu Action
sizing_36 = mesh_14.AddSizing()
#endregion
```

```
#region Details View Action
selection = ExtAPI.SelectionManager.CreateSelectionInfo(SelectionTypeEnum.GeometryEntities)
selection.Ids = [22, 6]
sizing_36.Location = selection
#endregion
```

```
#region Details View Action
sizing_36.Type = SizingType.NumberOfDivisions
```

#endregion

#region Details View Action  
sizing\_36.NumberOfDivisions = 50  
#endregion

#region Details View Action  
sizing\_36.CaptureCurvature = False  
#endregion

#region Details View Action  
sizing\_36.Behavior = SizingBehavior.Hard  
#endregion

#region UI Action  
sizing\_36.Name = r""airfoil-top""  
#endregion

#region Context Menu Action  
sizing\_38 = mesh\_14.AddSizing()  
#endregion

#region Details View Action  
selection = ExtAPI.SelectionManager.CreateSelectionInfo(SelectionTypeEnum.GeometryEntities)  
selection.Ids = [28, 10]  
sizing\_38.Location = selection  
#endregion

#region Details View Action  
sizing\_38.Type = SizingType.NumberOfDivisions  
#endregion

#region Details View Action  
sizing\_38.NumberOfDivisions = 50  
#endregion

#region Details View Action  
sizing\_38.CaptureCurvature = False  
#endregion

#region Details View Action  
sizing\_38.Behavior = SizingBehavior.Hard  
#endregion

#region UI Action  
sizing\_38.Name = r""airfoil-bottom""  
#endregion

#region Context Menu Action  
sizing\_40 = mesh\_14.AddSizing()  
#endregion

#region Details View Action  
selection = ExtAPI.SelectionManager.CreateSelectionInfo(SelectionTypeEnum.GeometryEntities)  
selection.Ids = [12, 13, 27, 23]  
sizing\_40.Location = selection

```

#endregion

#region Details View Action
sizing_40.Type = SizingType.NumberOfDivisions
#endregion

#region Details View Action
sizing_40.NumberOfDivisions = 20
#endregion

#region Details View Action
sizing_40.CaptureCurvature = False
#endregion

#region Details View Action
sizing_40.Behavior = SizingBehavior.Hard
#endregion

#region UI Action
sizing_40.Name = r""leading-edge""
#endregion

#region Context Menu Action
face_meshing_42 = mesh_14.AddFaceMeshing()
#endregion

#region Details View Action
selection = ExtAPI.SelectionManager.CreateSelectionInfo(SelectionTypeEnum.GeometryEntities)
selection.Ids = [19, 24, 20, 25, 15, 14]
face_meshing_42.Location = selection
#endregion

#region Details View Action
mesh_14.PhysicsPreference = MeshPhysicsPreferenceType.CFD
#endregion

#region Details View Action
mesh_14AutomaticMeshBasedDefeaturing = 0
#endregion

#region Context Menu Action
mesh_14.GenerateMesh()
#endregion

#region UI Action
named_selection_24 = DataModel.GetObjectById(24)
named_selection_24.Name = r""inlet""
#endregion

#region UI Action
named_selection_26 = DataModel.GetObjectById(26)
named_selection_26.Name = r""outlet""
#endregion

#region UI Action
named_selection_28 = DataModel.GetObjectById(28)

```

```
named_selection_28.Name = r""airfoil-wall""  
#endregion
```

```
#region Context Menu Action  
mesh_14.Update()  
#endregion
```

## 6.5 ANEXO 5. JOURNAL FILE PARA EL ANÁLISIS CFD DEL PERFIL S809 EN ANSYS WORKBENCH.

```
# encoding: utf-8
# 2023 R2
SetScriptVersion(Version="23.2.142")

Open(FilePath="C:/Users/shado/Documentos/ANSYS/Projects/Tesis/S809.wbpj")
Extensions.UnloadExtension(
  Id="5e7e01a6-8601-11e8-9f8c-28f10e13ffe6",
  Version="2022.2",
  Format="Binary")
Extensions.UnloadExtension(
  Id="5f463412-bd3e-484b-87e7-cbc0a665e474",
  Version="2023.2",
  Format="Binary")
Extensions.UnloadExtension(
  Id="7b0e9e84-396d-4099-9602-2ced9dddc253",
  Version="2023.2",
  Format="Binary")
Extensions.UnloadExtension(
  Id="20180725-3f81-49eb-9f31-41364844c769",
  Version="2023.2",
  Format="Binary")
Extensions.UnloadExtension(
  Id="f3e3da52-fb02-4910-8cc9-980efd047bc6",
  Version="2023.1",
  Format="Binary")

designPoint1 = Parameters.GetDesignPoint(Name="0")
parameter1 = Parameters.GetParameter(Name="P1")
designPoint1.SetParameterExpression(
  Parameter=parameter1,
  Expression="0.86603")
parameter2 = Parameters.GetParameter(Name="P2")
designPoint1.SetParameterExpression(
  Parameter=parameter2,
  Expression="0.5")

system1 = GetSystem(Name="FLU")
setupComponent1 = system1.GetComponent(Name="Setup")
setupComponent1.Refresh()

setup1 = system1.GetContainer(ComponentName="Setup")
fluentLauncherSettings1 = setup1.GetFluentLauncherSettings()
fluentLauncherSettings1.SetEntityProperties(Properties=Set(EnvPath={}))
setup1.Edit()
setup1.SendCommand(Command='(cx-gui-do cx-activate-tab-index "NavigationPane*Frame1(TreeTab)"
0)(cx-gui-do cx-activate-tab-index "NavigationPane*Frame1(TreeTab)" 1)')
setup1.SendCommand(Command='/file/set-tui-version "23.2"(cx-gui-do cx-set-list-tree-selections
"NavigationPane*Frame2*Table1*List_Tree2" (list "Setup|Reference Values"))')
setup1.SendCommand(Command='(cx-gui-do cx-set-list-tree-selections
"NavigationPane*Frame2*Table1*List_Tree2" (list "Setup|Reference Values"))(cx-gui-do cx-activate-item
"NavigationPane*Frame2*Table1*List_Tree2")')
```

```

setup1.SendCommand(Command="(cx-gui-do                                     cx-set-list-tree-selections
\NavigationPane*Frame2*Table1*List_Tree2\ (list \Setup|Reference Values\))(cx-gui-do cx-set-list-
selections \Reference Values*DropDownList1(Compute from)\ ( 3))")
setup1.SendCommand(Command="(cx-gui-do cx-activate-item "Reference Values*DropDownList1(Compute
from)")')
setup1.SendCommand(Command="(cx-gui-do                                     cx-set-list-tree-selections
"NavigationPane*Frame2*Table1*List_Tree2" (list "Solution|Initialization"))')
setup1.SendCommand(Command="(cx-gui-do                                     cx-set-list-tree-selections
"NavigationPane*Frame2*Table1*List_Tree2" (list "Solution|Initialization"))(cx-gui-do cx-activate-item
"NavigationPane*Frame2*Table1*List_Tree2")')
setup1.SendCommand(Command="(cx-gui-do                                     cx-set-list-tree-selections
"NavigationPane*Frame2*Table1*List_Tree2" (list "Solution|Initialization"))')
setup1.SendCommand(Command="(cx-gui-do                                     cx-activate-item          "Solution
Initialization*Table1*Frame12*PushButton2(Initialize)")')
setup1.SendCommand(Command="(cx-gui-do                                     cx-set-list-tree-selections
"NavigationPane*Frame2*Table1*List_Tree2" (list "Solution|Run Calculation"))')
setup1.SendCommand(Command="(cx-gui-do                                     cx-set-list-tree-selections
"NavigationPane*Frame2*Table1*List_Tree2" (list "Solution|Run Calculation"))(cx-gui-do cx-activate-item
"NavigationPane*Frame2*Table1*List_Tree2")')
setup1.SendCommand(Command="(cx-gui-do                                     cx-set-list-tree-selections
"NavigationPane*Frame2*Table1*List_Tree2" (list "Solution|Run Calculation"))')
setup1.SendCommand(Command="(cx-gui-do cx-activate-item "Run Calculation*Table1*Table6(Solution
Advancement)*Table1*PushButton1(Calculate)")')
setup1.SendCommand(Command="(cx-gui-do cx-activate-item "Information*OK")')
setup1.SendCommand(Command="(cx-gui-do cx-activate-item "MenuBar*FileMenu*Close Fluent")')

```

```

Parameters.ExportAllDesignPointsData(FilePath="C:/Users/shado/Documents/MATLAB/Tesis_Maestria/parameters.csv")

```

```

ClearMessages()

```

```

Save(Overwrite=True)

```

## 6.6 ANEXO 6. SCRIPT EN MATLAB PARA EL ANÁLISIS NUMÉRICO AUTOMATIZADO DEL PERFIL S809.

```
% Simulación CFD del perfil S809
% M.C. Mecatrónica. UAQ FI, campus SJR
% Autor: Luis Mauricio García Noverola
clear all; close all; clc;

timerVal = tic;

%% Automatización del análisis
for AoA = 0:30
    % Calcular parámetros de entrada
    x_component = cos(AoA*pi()/180);
    y_component = sin(AoA*pi()/180);
    % Extraer texto de journal file
    fid = fopen('S809.wbjn','r'); % Perfil S809
    f = fread(fid,'*char');
    fclose(fid);
    % Reemplazar parámetros de entrada en el texto
    x_component = num2str(x_component);
    y_component = num2str(y_component);
    newf1 = replaceBetween(f,'Expression="',"'",x_component);
    f1 = extractBefore(newf1,'parameter2,');
    newf2 = replaceBetween(f,'Expression="',"'",y_component);
    f2 = extractAfter(newf2,'parameter2,');
    % Crear nuevo journal file
    f = sprintf('%sparameter2,%s',f1,f2);
    fid = fopen('S809.wbjn','w'); % Perfil S809
    fprintf(fid,'%s',f);
    fclose(fid);
    % Ejecutar journal file en Workbench y guardar coeficientes
    system('C:\Program Files\ANSYS Inc\ANSYS
Student\v232\Framework\bin\Win64\RunWB2.exe -B -R S809.wbjn'); % Perfil S809
    Cd(AoA+1,1) = xlsread('parameters.csv','parameters','D8');
    Cl(AoA+1,1) = xlsread('parameters.csv','parameters','E8');
    alpha(AoA+1,1) = AoA;
end

time = toc(timerVal)
```



## 6.7 ANEXO 7. JOURNAL FILE PARA EL ANÁLISIS CFD DEL PROCESO DE OPTIMIZACIÓN EN ANSYS WORKBENCH.

```
# encoding: utf-8
# 2023 R2
SetScriptVersion(Version="23.2.142")

Open(FilePath="C:/Users/shado/Documentos/ANSYS/Projects/Tesis/airfoil_PSO.wbpj")
Extensions.UnloadExtension(
    Id="5e7e01a6-8601-11e8-9f8c-28f10e13ffe6",
    Version="2022.2",
    Format="Binary")
Extensions.UnloadExtension(
    Id="5f463412-bd3e-484b-87e7-cbc0a665e474",
    Version="2023.2",
    Format="Binary")
Extensions.UnloadExtension(
    Id="7b0e9e84-396d-4099-9602-2ced9dddc253",
    Version="2023.2",
    Format="Binary")
Extensions.UnloadExtension(
    Id="20180725-3f81-49eb-9f31-41364844c769",
    Version="2023.2",
    Format="Binary")
Extensions.UnloadExtension(
    Id="f3e3da52-fb02-4910-8cc9-980efd047bc6",
    Version="2023.1",
    Format="Binary")

system1 = GetSystem(Name="FLU")
solutionComponent1 = system1.GetComponent(Name="Solution")
solutionComponent1.Reset()

system2 = GetSystem(Name="Geom")
geometryComponent1 = system2.GetComponent(Name="Geometry")
geometryComponent1.Reset()

geometry1 = system2.GetContainer(ComponentName="Geometry")
geometry1.Edit(IsSpaceClaimGeometry=True)
geometry1.RunScript(ScriptFile="C:/Users/shado/Documentos/MATLAB/Tesis_Maestria/geometry.py")
geometry1.Exit()

system3 = GetSystem(Name="SYS")
meshComponent1 = system3.GetComponent(Name="Mesh")
meshComponent1.Reset()
meshComponent1.Refresh()

mesh1 = system3.GetContainer(ComponentName="Mesh")
mesh1.Edit()
# Send the command to the meshing container to read the macro file for getting all the commands from Meshing
mesh1.SendCommand(Command="WB.AppletList.Applet(\"DSApplet\").App.Script.doToolsRunMacro(\"C:/Users/shado/Documentos/MATLAB/Tesis_Maestria/mesh.py\")")
mesh1.Exit()
```

```

setupComponent1 = system1.GetComponent(Name="Setup")
setupComponent1.Refresh()

setup1 = system1.GetContainer(ComponentName="Setup")
fluentLauncherSettings1 = setup1.GetFluentLauncherSettings()
fluentLauncherSettings1.SetEntityProperties(Properties=Set(EnvPath={}))
setup1.Edit()
setup1.SendCommand(Command='(cx-gui-do cx-activate-tab-index "NavigationPane*Frame1(TreeTab)"
0)(cx-gui-do cx-activate-tab-index "NavigationPane*Frame1(TreeTab)" 1)')
setup1.SendCommand(Command='/file/set-tui-version "23.2"(cx-gui-do cx-set-list-tree-selections
"NavigationPane*Frame2*Table1*List_Tree2" (list "Setup|Reference Values"))')
setup1.SendCommand(Command='(cx-gui-do cx-set-list-tree-selections
"NavigationPane*Frame2*Table1*List_Tree2" (list "Setup|Reference Values"))(cx-gui-do cx-activate-item
"NavigationPane*Frame2*Table1*List_Tree2")')
setup1.SendCommand(Command='(cx-gui-do cx-set-list-tree-selections
"NavigationPane*Frame2*Table1*List_Tree2" (list "Setup|Reference Values"))(cx-gui-do cx-set-list-
selections "Reference Values*DropDownList1(Compute from)" '( 3))')
setup1.SendCommand(Command='(cx-gui-do cx-activate-item "Reference Values*DropDownList1(Compute
from)')')
setup1.SendCommand(Command='(cx-gui-do cx-set-list-tree-selections
"NavigationPane*Frame2*Table1*List_Tree2" (list "Solution|Initialization"))')
setup1.SendCommand(Command='(cx-gui-do cx-set-list-tree-selections
"NavigationPane*Frame2*Table1*List_Tree2" (list "Solution|Initialization"))(cx-gui-do cx-activate-item
"NavigationPane*Frame2*Table1*List_Tree2")')
setup1.SendCommand(Command='(cx-gui-do cx-set-list-tree-selections
"NavigationPane*Frame2*Table1*List_Tree2" (list "Solution|Initialization"))')
setup1.SendCommand(Command='(cx-gui-do cx-activate-item "Solution
Initialization*Table1*Frame12*PushButton2(Initialize)')')
setup1.SendCommand(Command='(cx-gui-do cx-set-list-tree-selections
"NavigationPane*Frame2*Table1*List_Tree2" (list "Solution|Run Calculation"))')
setup1.SendCommand(Command='(cx-gui-do cx-set-list-tree-selections
"NavigationPane*Frame2*Table1*List_Tree2" (list "Solution|Run Calculation"))(cx-gui-do cx-activate-item
"NavigationPane*Frame2*Table1*List_Tree2")')
setup1.SendCommand(Command='(cx-gui-do cx-set-list-tree-selections
"NavigationPane*Frame2*Table1*List_Tree2" (list "Solution|Run Calculation"))')
setup1.SendCommand(Command='(cx-gui-do cx-activate-item "Run Calculation*Table1*Table6(Solution
Advancement)*Table1*PushButton1(Calculate)')')
setup1.SendCommand(Command='(cx-gui-do cx-activate-item "Information*OK")')
setup1.SendCommand(Command='(cx-gui-do cx-activate-item "MenuBar*FileMenu*Close Fluent")')

Parameters.ExportAllDesignPointsData(FilePath="C:/Users/shado/Documents/MATLAB/Tesis_Maestria/pa
rametersPSO.csv")

ClearMessages()

Save(Overwrite=True)

```

## 6.8 ANEXO 8. SCRIPT EN MATLAB PARA EL ACOPLAMIENTO DEL ANALIZADOR-OPTIMIZADOR.

```
% Optimizador-Analizador con PSO-CFD
% M.C. Mecatrónica. UAQ FI, campus SJR
% Autor: Luis Mauricio García Noverola
clear all; close all; clc;

timerVal = tic;

%% Inicialización
% Parámetros de control
d = 16; % Número de variables
p = 10; % Número de partículas
ini = 1; % Iteración inicial
T = 10; % Número de iteraciones totales
wmin = 0.1; % Factor del peso de inercia mínimo
wmax = 0.9; % Factor del peso de inercia máximo
e1 = rand(p,d); % Vector aleatorio
e2 = rand(p,d); % Vector aleatorio
c1 = 2; % Factor de aprendizaje cognitivo
c2 = 2; % Factor de aprendizaje social

%% Generación de la población inicial con método CST
% Obtener perfiles parametrizados con CST
geometry = 'S809'; % Perfil de referencia
n = 7; % Orden del BPN(x)
limPor = 0.5; % Porcentaje de los límites de las variables de diseño
x = 50; % Número de puntos por superficie del perfil
solSPACE = linspace(0,1,x); % Espacio solución
solSPACEInv = flip(solSPACE); % Espacio solución invertido
% Llamar función CST y graficar perfiles
figure
for i = 1:p
    % Obtener parámetros, coeficientes y perfiles para cada partícula
    [parameters(i,:),Au(i,:),Al(i,:),surf_u(i,:),surf_l(i,:)] =
f_CST(geometry,n,x,limPor);
    % Generar la gráfica de los perfiles de cada partícula
    c(i,:) = [rand rand rand];
    plot(solSPACE,surf_u(i,:),solSPACEInv,surf_l(i,:), 'Color',c(i,:), 'LineWidth',1);
    hold on
end
title('Población inicial')
xlabel('x/c')
ylabel('z/c')
legend('p1',' ','p2',' ','p3',' ','p4',' ','p5',' ','p6',' ','p7',' ','p8',' ','p9',' ','
p10',' ') % p = 10
% Inicializar vectores de posición y velocidad de partículas
X = [Au, Al]; % Vector de posiciones inicial
X_inicial = X;
V = zeros(p,d); % Vector de velocidades inicial
% Crear matriz de coordenadas para cada partícula
surf_u = surf_u';
surf_u = flip(surf_u);
```

```

surf_l = surf_l';
coord = zeros(x,2); % Matriz de coordenadas
for i = 1:p
    % Guardar coordenadas del perfil en matriz
    for j = 1:length(solSPACE)
        coord(j,1) = solSPACEInv(j);
        coord(j,2) = surf_u(j,i);
        coord(j+x,1) = solSPACE(j);
        coord(j+x,2) = surf_l(j,i);
    end
    % Crear archivo de coordenadas del perfil
    fname_coord =
sprintf('C:/Users/shado/Documentos/MATLAB/Tesis_Maestria/airfoils_coordinates/p%
d.txt',i);
    fID_coord = fopen(fname_coord,'w');
    fprintf(fID_coord,'polyline=true\n\n');
    for k = 1:length(coord)
        if k == x+1
            else
                fprintf(fID_coord,'1 %f %f\n',coord(k,1),coord(k,2));
            end
        end
    end
    fclose(fID_coord);
    % Extraer texto de geometry.py y reemplazar el archivo de coordenadas en el
texto
    fname_geom = 'C:/Users/shado/Documentos/MATLAB/Tesis_Maestria/geometry.py';
    text = extractFileText(fname_geom);
    fnamestr_coord = sprintf('p%d',i);
    newText =
replaceBetween(text,'airfoils_coordinates/','.txt',fnamestr_coord);
    % Sobrescribir el archivo geometry.py
    fID_geom =
fopen('C:/Users/shado/Documentos/MATLAB/Tesis_Maestria/geometry.py','w');
    fprintf(fID_geom,newText);
    fclose(fID_geom);
    % Ejecutar journal file en Workbench y guardar coeficientes
    system('C:\Program Files\ANSYS Inc\ANSYS
Student\v232\Framework\bin\Win64\RunWB2.exe -B -R airfoil_PSO.wbjn');
    drag(i,1) = xlsread('parametersPS0.csv','parametersPS0','D8');
    lift(i,1) = xlsread('parametersPS0.csv','parametersPS0','E8');
    % Evaluar la función objetivo
    f(i,1) = lift(i,1);
end
% Calcular el fitness de las partículas y encontrar el índice de la mejor
partícula b
[fitness(1,:),index] = max(f(:,1));
% Encontrar PBest y GBest iniciales
pbest = X;
gbest(1,:) = pbest(index,:);

%% PSO
% Límites de los coeficientes
Zu = 0;
Zl = 0;
% Superficie superior

```

```

Ru_le = 0.0181;
beta_u = 10.758;
LB_u = [0.19 0.2 0.28 0.294 0.288 0.201 0.225 0.19].*(1-limPor); % Límites
inferiores
LB_u(1) = sqrt(2*Ru_le*(1-limPor));
LB_u(n+1) = tan(beta_u*(1-limPor)*pi/180)+Zu;
UB_u = [0.19 0.2 0.28 0.294 0.288 0.201 0.225 0.19].*(1+limPor); % Límites
superiores
UB_u(1) = sqrt(2*Ru_le*(1+limPor));
UB_u(n+1) = tan(beta_u*(1+limPor)*pi/180)+Zu;
% Superficie inferior
Rl_le = 0.0112;
beta_l = 5.7295e-4;
LB_l = [0.15 0.19 0.33 0.36 0.235 0.215 0.02 1e-5].*(1-limPor); % Límites
inferiores
LB_l(1) = sqrt(2*Rl_le*(1-limPor));
LB_l(n+1) = tan(beta_l*(1-limPor)*pi/180)+Zu;
UB_l = [0.19 0.2 0.28 0.294 0.288 0.201 0.225 0.19].*(1+limPor); % Límites
superiores
UB_l(1) = sqrt(2*Rl_le*(1+limPor));
UB_l(n+1) = tan(beta_l*(1+limPor)*pi/180)+Zu;
% Actualizar vectores de velocidad y posición de las partículas
for k = ini:T
    w = wmax - k*(wmax-wmin)/T; % Calcular peso de inercia
    V = w*X + c1*e1.*(gbest(k,:)-X) + c2*e2.*(pbest-X); % Actualizar vector de
    velocidades
    X_next = X + V; % Actualizar vector de posiciones
    % Mantener las variables de diseño dentro de los límites
    for i = 1:p
        for j = 1:d
            if j <= d/2
                if X_next(i,j) < LB_u(j)
                    X_next(i,j) = X(i,j);
                elseif X_next(i,j) > UB_u
                    X_next(i,j) = X(i,j);
                end
            else
                if X_next(i,j) < LB_l(j-d/2)
                    X_next(i,j) = X(i,j);
                elseif X_next(i,j) > UB_l(j-d/2)
                    X_next(i,j) = X(i,j);
                end
            end
        end
    end
    end
    % Perfiles de iteración actual con función CST
    surf_u = zeros(p,x);
    surf_l = zeros(p,x);
    figure
    % Obtener perfiles de iteración actual y graficar
    for i = 1:p
        % Obtener perfiles para cada partícula
        [surf_u(i,:),surf_l(i,:)] =
f_CST_inv(n,x,X_next(i,1:d/2),X_next(i,d/2+1:d));

```

```

        % Generar la gráfica de los perfiles de cada partícula
plot(solSPACE,surf_u(i,:),solSPACE,surf_l(i,:), 'Color',c(i,:), 'LineWidth',1);
    hold on
end
title(sprintf('k = %d',k))
xlabel('x/c')
ylabel('z/c')

legend('p1','','p2','','p3','','p4','','p5','','p6','','p7','','p8','','p9','','p10','') % p = 10
% Crear matriz de coordenadas para cada partícula
surf_u = surf_u';
surf_u = flip(surf_u);
surf_l = surf_l';
coord = zeros(x,2); % Matriz de coordenadas
for i = 1:p
% Guardar coordenadas del perfil en matriz
    for j = 1:length(solSPACE)
        coord(j,1) = solSPACEInv(j);
        coord(j,2) = surf_u(j,i);
        coord(j+x,1) = solSPACE(j);
        coord(j+x,2) = surf_l(j,i);
    end
% Crear archivo de coordenadas del perfil
    fname_coord =
sprintf(['C:/Users/shado/Documentos/MATLAB/Tesis_Maestria/airfoils_coordinates/p
%d.txt'],i);
    fID_coord = fopen(fname_coord,'w');
    fprintf(fID_coord,'polyline=true\n\n');
    for j = 1:length(coord)
        if j == x+1
            else
                fprintf(fID_coord,'1 %f %f\n',coord(j,1),coord(j,2));
            end
        end
    end
    fclose(fID_coord);
% Extraer texto de geometry.py y reemplazar el archivo de coordenadas en
el texto
    fname_geom =
'C:/Users/shado/Documentos/MATLAB/Tesis_Maestria/geometry.py';
    text = extractFileText(fname_geom);
    fnamestr_coord = sprintf('p%d',i);
    newText =
replaceBetween(text,'airfoils_coordinates/','.txt',fnamestr_coord);
% Sobrescribir el archivo geometry.py
    fID_geom =
fopen('C:/Users/shado/Documentos/MATLAB/Tesis_Maestria/geometry.py','w');
    fprintf(fID_geom,newText);
    fclose(fID_geom);
% Ejecutar journal file en Workbench y guardar coeficientes
    system('C:\Program Files\ANSYS Inc\ANSYS
Student\v232\Framework\bin\Win64\RunWB2.exe -B -R airfoil_PS0.wbjn');
    drag(i,k+1) = xlsread('parametersPS0.csv','parametersPS0','D8');
    lift(i,k+1) = xlsread('parametersPS0.csv','parametersPS0','E8');
% Evaluar la función objetivo

```

```

        f(i,k+1) = lift(i,k+1);
        % Actualizar PBest
        if f(i,k+1) > f(i,k)
            pbest(i,:) = X_next(i,:);
        else
            pbest(i,:) = pbest(i,:);
        end
    end
    % Calcular el fitness de las partículas y encontrar el índice de la mejor
partícula b
    [fitness(k+1),index] = max(f(:,k+1));
    % Obtener posición del GBest
    if fitness(k+1) > fitness(k)
        gbest(k+1,:) = pbest(index,:);
    else
        gbest(k+1,:) = gbest(k,:);
        fitness(k+1) = fitness(k);
    end
    % Reasignar vector de posición de partículas
    X = X_next;
    % Recalcular vectores aleatorios
    e1 = rand(p,d);
    e2 = rand(p,d);
end

figure
ite = [1:T+1];
plot(ite,fitness)
xlabel('Iteraciones')
ylabel('VFP')
legend('Fitness')

figure
[surf_u_final,surf_l_final] =
f_CST_inv(n,x,gbest(T+1,1:d/2),gbest(T+1,d/2+1:d));
plot(solSPACE,surf_u_final,solSPACE,surf_l_final,'Color','r','Color','r','LineWi
dth',1);
xlabel('x/c')
ylabel('z/c')
legend('Perfil óptimo','')

time = toc(timerVal)

```

## VII. REFERENCIAS

- Abdelwahed, K. S., & Abd El-Rahman, A. I. (2020). Shape optimization of SG6043 airfoil for small wind turbine blades. *Journal of Physics: Conference Series*, 1618(4). <https://doi.org/10.1088/1742-6596/1618/4/042007>
- Akbari, V., Naghashzadegan, M., Kouhikamali, R., Afsharpanah, F., & Yaïci, W. (2022). Multi-Objective Optimization and Optimal Airfoil Blade Selection for a Small Horizontal-Axis Wind Turbine (HAWT) for Application in Regions with Various Wind Potential. *Machines*, 10(8). <https://doi.org/10.3390/machines10080687>
- Akram, M. T., & Kim, M. H. (2021a). Aerodynamic Shape Optimization of NREL S809 Airfoil for Wind Turbine Blades Using Reynolds-Averaged Navier Stokes Model—Part II. *Applied Sciences*, 11(5), 1–26. <https://doi.org/10.3390/app11052211>
- Akram, M. T., & Kim, M. H. (2021b). CFD Analysis and Shape Optimization of Airfoils Using Class Shape Transformation and Genetic Algorithm—Part I. *Applied Sciences*, 11(9). <https://doi.org/10.3390/app11093791>
- AMDEE. (2023). *REPORTE ANUAL 2022*. <https://amdee.org/informes-anauales/>
- Anitha, D., Shamili, G. K., Ravi Kumar, P., & Sabari Vihar, R. (2018). Air foil Shape Optimization Using Cfd and Parametrization Methods. *Materials Today: Proceedings*, 5(2), 5364–5373. <https://doi.org/10.1016/j.matpr.2017.12.122>
- ANSYS Inc. (n.d.). *Ansys Mosaic Meshing Technology Combines Disparate Meshes*. White Paper. Retrieved September 17, 2023, from <https://www.ansys.com/resource-center/white-paper/ansys-fluent-mosaic-technology>
- ANSYS Inc. (2013). *Workbench User's Guide*. <http://www.ansys.com>
- ANSYS Inc. (2016, November 15). *Ansys, HLRS And Cray Set New Supercomputing Record*. Press Release. <https://www.ansys.com/news-center/press-releases/11-15-16-ansys-hlrs-cray-set-new-supercomputing-record>
- ANSYS Inc. (2021a). *ANSYS Fluent Theory Guide 2021 R2*. <http://www.ansys.com>
- ANSYS Inc. (2021b). *ANSYS Meshing User's Guide*. <http://www.ansys.com>
- ANSYS Inc. (2023a). *Ansys Workbench Scripting Guide*. <http://www.ansys.com>
- ANSYS Inc. (2023b). *Discovery SpaceClaim*.
- Belamadi, R., Djemili, A., Ilinca, A., & Mdouki, R. (2016). Aerodynamic performance analysis of slotted airfoils for application to wind turbine blades. *Journal of Wind*



- Engineering and Industrial Aerodynamics*, 151, 79–99.  
<https://doi.org/10.1016/j.jweia.2016.01.011>
- Benim, A. C., Diederich, M., & Pfeiffelmann, B. (2018). Aerodynamic Optimization of Airfoil Profiles for Small Horizontal Axis Wind Turbines. *Computation*, 6(2).  
<https://doi.org/10.3390/computation6020034>
- Bouhelal, A., Smaili, A., Guerri, O., & Masson, C. (2017). Comparison of BEM and Full Navier-Stokes CFD Methods for Prediction of Aerodynamics Performance of HAWT Rotors. *2017 International Renewable and Sustainable Energy Conference (IRSEC)*.  
<https://doi.org/10.1109/IRSEC.2017.8477247>
- Burton, T., Jenkins, N., Sharpe, D., & Bossanyi, E. (2011). *Wind Energy Handbook* (2nd ed.). Wiley. [www.wiley.com](http://www.wiley.com).
- Butterfield, C. P., Musial, W. P., & Simms, D. A. (1992). *Combined Experiment Phase I: Final Report*. chrome-extension://efaidnbnmnnibpcajpcglclefindmkaj/<https://www.nrel.gov/docs/legosti/old/4655.pdf>
- Cengel, Y. A., & Cimbala, J. M. (2006). *Mecánica de Fluidos. Fundamentos y Aplicaciones*. (1st ed.). McGraw-Hill Interamericana.
- Chehoury, A., Younes, R., Ilinca, A., & Perron, J. (2015). Review of performance optimization techniques applied to wind turbines. *Applied Energy*, 142, 361–388.  
<https://doi.org/10.1016/j.apenergy.2014.12.043>
- Chen, W., Chiu, K., & Fuge, M. D. (2020). Airfoil Design Parameterization and Optimization Using Bézier Generative Adversarial Networks. *AIAA Journal*, 58(11), 4723–4735. <https://doi.org/10.2514/1.J059317>
- Costoya, X., deCastro, M., Carvalho, D., & Gómez-Gesteira, M. (2023). Assessing the complementarity of future hybrid wind and solar photovoltaic energy resources for North America. *Renewable and Sustainable Energy Reviews*, 173.  
<https://doi.org/10.1016/j.rser.2022.113101>
- Drela, M. (1989). XFOIL: An Analysis and Design System for Low Reynolds Number Airfoils. In *Low Reynolds Number Aerodynamics*. Springer.

- Du, X., He, P., & Martins, J. R. R. A. (2021). Rapid airfoil design optimization via neural networks-based parameterization and surrogate modeling. *Aerospace Science and Technology, 113*. <https://doi.org/10.1016/j.ast.2021.106701>
- El Khchine, Y., & Sriti, M. (2018). Improved Blade Element Momentum theory (BEM) for Predicting the Aerodynamic Performances of Horizontal Axis Wind Turbine Blade (HAWT). *Technische Mechanik, 38*(2), 191–202. <https://doi.org/10.24352/UB.OVGU-2018-028>
- Fan, M., & Akhter, Y. (2021). A Time-Varying Adaptive Inertia Weight based Modified PSO Algorithm for UAV Path Planning. *2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques, 573–576*. <https://doi.org/10.1109/ICREST51555.2021.9331101>
- García-Noverola, L. M., Domínguez-González, A., Pérez-Cruz, A., & Osornio-Ríos, R. A. (2023, May). Influence study on the variation of the CST method shape function coefficients on the wind turbine airfoil. *CONIIN XIX Congreso Internacional de Ingeniería*. <https://coniinuaq.wordpress.com/memorias-memories/>
- González, J. A., Pérez, R. C., Santos, A. C., & Gil, M.-A. C. (2009). *Centrales de Energías Renovables: Generación Eléctrica con Energías Renovables*. Pearson.
- Grey, Z. J., & Constantine, P. G. (2018). Active Subspaces of Airfoil Shape Parameterizations. *Structures, Structural Dynamics, and Materials Conferences, 56*(5), 2003–2017. <https://doi.org/10.2514/1.J056054>
- Hamlaoui, M. N., Smaili, A., & Fellouah, H. (2018). Improved BEM Method for HAWT Performance Predictions. *2018 International Conference on Wind Energy and Applications in Algeria (ICWEAA)*. <https://doi.org/10.1109/ICWEAA.2018.8605096>
- Hand, M. M., Simms, D. A., Fingersh, L. J., Jager, D. W., Cotrell, J. R., Schreck, S., & Larwood, S. M. (2001). *Unsteady Aerodynamics Experiment Phase VI: Wind Tunnel Test Configurations and Available Data Campaigns*. <https://doi.org/10.2172/15000240>
- Hansen, T. H. (2018). Airfoil optimization for wind turbine application. *Wind Energy, 21*(7), 502–514. <https://doi.org/10.1002/we.2174>
- Hassan, R., Cohanin, B., De Weck, O., & Venter, G. (2005, April 18). A Comparison of Particle Swarm Optimization and the Genetic Algorithm. *46th*

*AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference.*

- Hassanzadeh, A., Hassanabad, A., & Dadvand, A. (2016). Aerodynamic shape optimization and analysis of small wind turbine blades employing the Viterna approach for post-stall region. *Alexandria Engineering Journal*, 55(3), 2035–2043. <https://doi.org/10.1016/j.aej.2016.07.008>
- Heinz, S. (2020). A review of hybrid RANS-LES methods for turbulent flows: Concepts and applications. *Progress in Aerospace Sciences*, 114. <https://doi.org/10.1016/j.paerosci.2019.100597>
- Hussain, K., Mohd Salleh, M. N., Cheng, S., & Shi, Y. (2019a). Metaheuristic research: a comprehensive survey. *Artificial Intelligence Review*, 52(4), 2191–2233. <https://doi.org/10.1007/s10462-017-9605-z>
- Hussain, K., Mohd Salleh, M. N., Cheng, S., & Shi, Y. (2019b). Metaheuristic research: a comprehensive survey. *Artificial Intelligence Review*, 52(4), 2191–2233. <https://doi.org/10.1007/s10462-017-9605-z>
- Johnson, D. A., Gu, M., & Gaunt, B. (2016). Wind Turbine Performance in Controlled Conditions: BEM Modeling and Comparison with Experimental Results. *International Journal of Rotating Machinery*, 2016. <https://doi.org/10.1155/2016/5460823>
- Jureczko, M., Pawlak, M., & Męzyk, A. (2005). Optimisation of wind turbine blades. *Journal of Materials Processing Technology*, 167(2–3), 463–471. <https://doi.org/10.1016/j.jmatprotec.2005.06.055>
- Karthikeyan, N., Kalidasa Murugavel, K., Arun Kumar, S., & Rajakumar, S. (2015). Review of aerodynamic developments on small horizontal axis wind turbine blade. *Renewable and Sustainable Energy Reviews*, 42, 801–822. <https://doi.org/10.1016/j.rser.2014.10.086>
- Kaviani, H., & Nejat, A. (2017). Aeroacoustic and Aerodynamic Optimization of a MW class HAWT using MOPSO Algorithm. *Energy*, 140, 1198–1215. <https://doi.org/10.1016/j.energy.2017.08.011>
- Kaviani, H. R., & Moshfeghi, M. (2023a). Multi-Megawatt Horizontal Axis Wind Turbine Blade Optimization Based on PSO Method. *Aerospace*, 10(2). <https://doi.org/10.3390/aerospace10020158>

- Kaviani, H. R., & Moshfeghi, M. (2023b). Multi-Megawatt Horizontal Axis Wind Turbine Blade Optimization Based on PSO Method. *Aerospace*, *10*(2). <https://doi.org/10.3390/aerospace10020158>
- Kelecy, F. J. (2021, April 8). Adjoint Shape Optimization for Aerospace Applications. *Advanced Modeling & Simulation (AMS) Seminar Series*.
- Khurana, M., & Massey, K. (2015). Swarm algorithm with adaptive mutation for airfoil aerodynamic design. *Swarm and Evolutionary Computation*, *20*, 1–13. <https://doi.org/10.1016/j.swevo.2014.10.001>
- Kim, Y., Bangga, G., & Delgado, A. (2020). Investigations of HAWT Airfoil Shape Characteristics and 3D Rotational Augmentation Sensitivity toward the Aerodynamic Performance Improvement. *Sustainability*, *12*(18). <https://doi.org/10.3390/su12187597>
- Kostas, K. V., Amiralin, A., Sagimbayev, S., Massalov, T., Kalel, Y., & Politis, C. G. (2020). Parametric model for the reconstruction and representation of hydrofoils and airfoils. *Ocean Engineering*, *199*. <https://doi.org/10.1016/j.oceaneng.2020.107020>
- Kulfan, B. M. (2008). Universal Parametric Geometry Representation Method. *Journal of Aircraft*, *45*(1), 142–158. <https://doi.org/10.2514/1.29958>
- Kulfan, B. M., & Bussoletti, J. E. (2006, September 6). “Fundamental” Parametric Geometry Representations for Aircraft Component Shapes. *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference: The Modeling and Simulation Frontier for Multidisciplinary Design Optimization*.
- Lanzafame, R., Mauro, S., & Messina, M. (2015). HAWT Design and Performance Evaluation: Improving the BEM theory Mathematical Models. *Energy Procedia*, *82*, 172–179. <https://doi.org/10.1016/j.egypro.2015.12.015>
- Manwell, J. F., McGowan, J. G., & Rogers, A. L. (2010). *Wind Energy Explained: Theory, Design and Application* (2nd ed.). Wiley.
- Masoumi, H., & Jalili, F. (2019). Inverse and Direct Optimization Shape of Airfoil Using Hybrid Algorithm Big Bang-Big Crunch and Particle Swarm Optimization. *Journal of Theoretical and Applied Mechanics*, *57*(3), 697–711. <https://doi.org/10.15632/jtam-pl/110130>

- Masters, D. A., Taylor, N. J., Rendall, T. C. S., Allen, C. B., & Poole, D. J. (2017). Geometric Comparison of Aerofoil Shape Parameterization Methods. *AIAA Journal*, 55(5), 1575–1589. <https://doi.org/10.2514/1.J054943>
- Mauro, S., Lanzafame, R., Messina, M., & Pirrello, D. (2017). Transition turbulence model calibration for wind turbine airfoil characterization through the use of a Micro-Genetic Algorithm. *International Journal of Energy and Environmental Engineering*, 8(4), 359–374. <https://doi.org/10.1007/s40095-017-0248-2>
- Menter, F. R., & Sechner, R. (2021). *Best Practice: RANS Turbulence Modeling in Ansys CFD*.
- Moallemi, E. A., Eker, S., Gao, L., Hadjidakou, M., Liu, Q., Kwakkel, J., Reed, P. M., Obersteiner, M., Guo, Z., & Bryan, B. A. (2022). Early systems change necessary for catalyzing long-term sustainability in a post-2030 agenda. *One Earth*, 5(7), 792–811. <https://doi.org/10.1016/j.oneear.2022.06.003>
- Moreira Chagas, C. C., Pereira, M. G., Rosa, L. P., da Silva, N. F., Vasconcelos Freitas, M. A., & Hunt, J. D. (2020). From Megawatts to Kilowatts: A Review of Small Wind Turbine Applications, Lessons From The US to Brazil. *Sustainability (Switzerland)*, 12(7). <https://doi.org/10.3390/su12072760>
- Morgado, J., Vizinho, R., Silvestre, M. A. R., & Páscoa, J. C. (2016). XFOIL vs CFD performance predictions for high lift low Reynolds number airfoils. *Aerospace Science and Technology*, 52, 207–214. <https://doi.org/10.1016/j.ast.2016.02.031>
- Moriarty, P. J., & Hansen, A. C. (2005). *AeroDyn Theory Manual*. <http://www.osti.gov/bridge>
- Muhsen, H., Al-Kouz, W., & Khan, W. (2020). Small Wind Turbine Blade Design and Optimization. *Symmetry*, 12(18). <https://doi.org/10.3390/SYM12010018>
- Muratoglu, A., Tekin, R., & Ertuğrul, Ö. F. (2021). Hydrodynamic optimization of high-performance blade sections for stall regulated hydrokinetic turbines using Differential Evolution Algorithm. *Ocean Engineering*, 220. <https://doi.org/10.1016/j.oceaneng.2020.108389>
- Negahban, M. H., Bashir, M., & Botez, R. M. (2023). Free-Form Deformation Parameterization on the Aerodynamic Optimization of Morphing Trailing Edge. *Applied Mechanics*, 4(1), 304–316. <https://doi.org/10.3390/applmech4010017>

- Noronha, N. P., & Krishna, M. (2021). Aerodynamic performance comparison of airfoils suggested for small horizontal axis wind turbines. *Materials Today: Proceedings*, *46*, 2450–2455. <https://doi.org/10.1016/j.matpr.2021.01.359>
- Perez-Cruz, A., Dominguez-Gonzalez, A., Stiharu, I., & Osornio-Rios, R. A. (2012). Optimization of Q-factor of AFM cantilevers using genetic algorithms. *Ultramicroscopy*, *115*, 61–67. <https://doi.org/10.1016/j.ultramic.2012.01.014>
- Petrone, G. (2022, September 29). *Ansys Fluent: A History of Innovations in CFD*. ANSYS Blog. <https://www.ansys.com/blog/ansys-fluent-history-of-innovations>
- Pourrajabian, A., Dehghan, M., & Rahgozar, S. (2021). Genetic algorithms for the design and optimization of horizontal axis wind turbine (HAWT) blades: A continuous approach or a binary one? *Sustainable Energy Technologies and Assessments*, *44*. <https://doi.org/10.1016/j.seta.2021.101022>
- Ram, K. R., Lal, S. P., & Ahmed, M. R. (2018). Design and optimization of airfoils and a 20 kW wind turbine using multi-objective genetic algorithm and HARP\_Opt code. *Renewable Energy*, *56–67*. <https://doi.org/10.1016/j.renene.2018.08.040>
- Ramírez-Ochoa, D., Pérez-Domínguez, L. A., Martínez-Gómez, E. A., & Luviano-Cruz, D. (2022). PSO, a Swarm Intelligence-Based Evolutionary Algorithm as a Decision-Making Strategy: A Review. *Symmetry*, *14*(3). <https://doi.org/10.3390/sym14030455>
- Rodriguez, C. V., & Celis, C. (2022). Design optimization methodology of small horizontal axis wind turbine blades using a hybrid CFD/BEM/GA approach. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, *44*(6). <https://doi.org/10.1007/s40430-022-03561-4>
- Rosas, S. (2024, January 23). *Ya opera el parque eólico de Querétaro; genera 30 megawatts de electricidad*. El Universal Querétaro. <https://www.eluniversalqueretaro.mx/cartera/ya-opera-el-parque-eolico-de-queretaro-genera-30-megawatts-de-electricidad>
- Sadraey, M. H. (2013). *Aircraft design: A Systems Engineering Approach*. Wiley.
- Sale, D. (2010). *HARP\_Opt User's Guide*. <https://www.nrel.gov/wind/nwtc/harp-opt.html>
- Scappatici, L., Bartolini, N., Castellani, F., Astolfi, D., Garinei, A., & Pennicchi, M. (2016). Optimizing the design of horizontal-axis small wind turbines: From the

- laboratory to market. *Journal of Wind Engineering and Industrial Aerodynamics*, 154, 58–68. <https://doi.org/10.1016/j.jweia.2016.04.006>
- Schubel, P. J., & Crossley, R. J. (2012). Wind Turbine Blade Design. *Energies*, 5(9), 3425–3449. <https://doi.org/10.3390/en5093425>
- Sharma, P., Gupta, B., Pandey, M., Sharma, A. K., & Nareliya Mishra, R. (2020). Recent advancements in optimization methods for wind turbine airfoil design: A review. *Materials Today: Proceedings*, 47, 6556–6563. <https://doi.org/10.1016/j.matpr.2021.02.231>
- Shen, X., Yang, H., Chen, J., Zhu, X., & Du, Z. (2016). Aerodynamic shape optimization of non-straight small wind turbine blades. *Energy Conversion and Management*, 119, 266–278. <https://doi.org/10.1016/j.enconman.2016.04.008>
- Simon, D. (2013). *Evolutionary Optimization Algorithms: Biologically-Inspired and Population-Based Approaches to Computer Intelligence*. Wiley.
- Somers, D. M. (1997). *Design and Experimental Results for the S809 Airfoil*.
- Suresh, A., & Rajakumar, S. (2019). Design of small horizontal axis wind turbine for low wind speed rural applications. *Materials Today: Proceedings*, 23, 16–22. <https://doi.org/10.1016/j.matpr.2019.06.008>
- Tang, X., Huang, X., Peng, R., & Liu, X. (2015). A Direct Approach of Design Optimization for Small Horizontal Axis Wind Turbine Blades. *Procedia CIRP 25th Design Conference Innovative Product Creation*, 36, 12–16. <https://doi.org/10.1016/j.procir.2015.01.047>
- Tangler, J., & Kocurek, J. D. (2005, January 10). Wind Turbine Post-Stall Airfoil Performance Characteristics Guidelines for Blade-Element Momentum Methods. *43rd AIAA Aerospace Sciences Meeting and Exhibit*. <http://www.osti.gov/bridgeonlineordering:http://www.ntis.gov/ordering.htm>
- Tangler, J. L., & Somers, D. M. (1995). *NREL Airfoil Families for HAWTs*. <https://doi.org/https://doi.org/10.2172/10106095>
- Tawfiq, K. B., Mansour, A. S., Ramadan, H. S., Becherif, M., & El-Kholy, E. E. (2019). Wind energy conversion system topologies and converters: Comparative review. *Energy Procedia*, 162, 38–47. <https://doi.org/10.1016/j.egypro.2019.04.005>

- The MathWorks Inc. (2023). *MATLAB. El lenguaje del cálculo técnico*. Matlab Centro de Ayuda. [https://la.mathworks.com/help/matlab/index.html?s\\_tid=hc\\_panel](https://la.mathworks.com/help/matlab/index.html?s_tid=hc_panel)
- Timmer, W. A., & Van Rooij, R. P. J. O. M. (2003). Summary of the Delft University Wind Turbine Dedicated Airfoils. *Journal of Solar Energy Engineering, Transactions of the ASME*, 125(4), 488–496. <https://doi.org/10.1115/1.1626129>
- Tortora, G., Concilio, A., & Pecora, R. (2023). Airfoil Shape Morphing through a Novel Parameterization and Fitting Optimization Method Based on Uniform Non-Rational B-Spline Functions. *Designs*, 7(1). <https://doi.org/10.3390/designs7010028>
- Verma, S., Joshi, A., & Paul, A. R. (2022). Design and Multi-Objective Optimization of a Horizontal Axis Wind Turbine. *AIP Conference Proceedings*, 2681. <https://doi.org/10.1063/5.0115991>
- Wang, D., Sun, X., Kang, H., Shen, Y., & Chen, Q. (2022). Heterogeneous differential evolution algorithm for parameter estimation of solar photovoltaic models. *Energy Reports*, 8, 4724–4746. <https://doi.org/10.1016/j.egyr.2022.03.144>
- Wang, K., Yu, S., Wang, Z., Feng, R., & Liu, T. (2019). Adjoint-based airfoil optimization with adaptive isogeometric discontinuous Galerkin method. *Computer Methods in Applied Mechanics and Engineering*, 344, 602–625. <https://doi.org/10.1016/j.cma.2018.10.033>
- White, F. M. (2004). *Mecánica de fluidos* (5th ed.). McGraw-Hill Interamericana.
- Wind Turbines-Part 2: Design Requirements for Small Wind Turbines, Pub. L. No. IEC 61400-2:2006 (2006). [www.iec.ch](http://www.iec.ch)
- Wu, X., Zhang, W., Peng, X., & Wang, Z. (2019). Benchmark aerodynamic shape optimization with the POD-based CST airfoil parametric method. *Aerospace Science and Technology*, 84, 632–640. <https://doi.org/10.1016/j.ast.2018.08.005>
- Xin, J., Chen, G., & Hai, Y. (2009). A Particle Swarm Optimizer with Multi-Stage Linearly-Decreasing Inertia Weight. *2009 International Joint Conference on Computational Sciences and Optimization*, 1, 505–508. <https://doi.org/10.1109/CSO.2009.420>
- Yang, H., Shen, W., Xu, H., Hong, Z., & Liu, C. (2014). Prediction of the wind turbine performance by using BEM with airfoil data extracted from CFD. *Renewable Energy*, 70, 107–115. <https://doi.org/10.1016/j.renene.2014.05.002>



- Yousaf, S., Mughees, A., Khan, M. G., Amin, A. A., & Adnan, M. (2020). A Comparative Analysis of Various Controller Techniques for Optimal Control of Smart Nano-Grid Using GA and PSO Algorithms. *IEEE Access*, 8, 205696–205711. <https://doi.org/10.1109/ACCESS.2020.3038021>
- Zezatti, M., & Castro, L. (2019, September 18). Análisis numérico de la interacción fluido-estructura en el rotor de un generador eólico. *Memorias Del XXV Congreso Internacional Anual de La SOMIM*. <https://somim.org.mx/memorias/memorias2019/thermo.html>
- Zhang, T., Huang, W., Wang, Z., & Yan, L. (2016). A study of airfoil parameterization, modeling, and optimization based on the computational fluid dynamics method. *Journal of Zhejiang University: Science A*, 17(8), 632–645. <https://doi.org/10.1631/jzus.A1500308>
- Zhang, W., Sun, J., Wang, L., Wu, J., & He, L. (2020). Rotor airfoil aerodynamic design method and wind tunnel test verification. *Chinese Journal of Aeronautics*, 33(8), 2123–2132. <https://doi.org/10.1016/j.cja.2020.03.013>