



**UNIVERSIDAD MICHOACANA DE SAN
NICOLÁS DE HIDALGO**

**FACULTAD DE INGENIERIA MECANICA
DIVISIÓN DE ESTUDIOS DE POSGRADO**

**BANCO DE PRUEBAS VIRTUAL BASADO EN
HPC PARA PRUEBAS DE MODELO DE LA
TURBINA FRANCIS 99**

TESIS

**QUE PARA OBTENER EL GRADO DE DOCTOR EN CIENCIAS EN
INGENIERÍA MECÁNICA**

PRESENTA:

M.C. DANIEL MOLINERO HERNÁNDEZ

ASESOR:

DR. SERGIO RICARDO GALVÁN GONZÁLEZ

COASESOR:

DR. NICOLAS DAVID HERRERA SANDOVAL

Morelia, Michoacán. Agosto 2024



Dedicatorias.

A mí amada madre Lourdes y mi hermano Antonio.

Agradecimientos.

A mis amigos, compañeros y profesores, quienes brindaron consejos y asesorías para el desarrollo y conclusión de este trabajo.

A la Universidad Michoacana de San Nicolás de Hidalgo y su Facultad de Ingeniería Mecánica, así como al Posgrado en Ciencias en Ingeniería Mecánica por las facilidades otorgadas para la conclusión de este proyecto.

Al Consejo Nacional de Humanidades, Ciencias y Tecnologías (Conahcyt) por el apoyo económico para la realización de este proyecto.

Resumen.

En la actualidad, durante las etapas de investigación y desarrollo de nuevas turbinas hidráulicas se realizan procesos de optimización del diseño por simulación numérica y verificación del rendimiento de las turbinas mediante pruebas físicas. Dado que las pruebas de modelo forman una gran parte del costo total del proyecto hidroeléctrico, se podría lograr una reducción significativa si las verificaciones de desempeño se sustituyeran por simulación numérica empleando cálculo de alto desempeño. Esta investigación abarca el desarrollo e implementación de un banco de pruebas virtual para turbinas hidráulicas tipo Francis. Para lograr el desarrollo del banco de pruebas virtual se hizo uso del código libre de código abierto OpenFOAM y un par de modificaciones del mismo capaces de acoplar tarjetas gráficas con el fin de acelerar el cálculo. En una etapa inicial se analizó el tubo difusor del caso de estudio de la turbina T99 con el fin de establecer la mejor configuración para aprovechar de mejor manera los recursos computacionales, se llevaron a cabo una serie de pruebas para determinar la influencia que tiene la arquitectura computacional de las tarjetas gráficas, el tipo solucionador algebraico y la memoria disponible en el tiempo de cálculo y exactitud de los resultados. Finalmente se realizaron análisis adicionales en estado estacionario y transitorio tomando como referencia el caso de estudio de la turbina Francis 99 con el fin de corroborar el comportamiento del banco de pruebas en un amplio rango de condiciones de operación.

Palabras clave: Turbina, tubo difusor, cálculo de alto desempeño, tarjetas gráficas, simulación.

Abstract.

Currently, during the research and development stages of new hydraulic turbines, design optimization processes are carried out by numerical simulation and performance verification of the turbines by physical tests. Since model tests form a large part of the total cost of the hydroelectric project, a significant reduction could be achieved if performance verifications were replaced by numerical simulation using high-performance calculation. This research covers the development and implementation of a virtual test bench for Francis-type hydraulic turbines. To achieve the development of the virtual test bench, the free and open source code OpenFOAM was used, as well as a couple of modifications of it capable of coupling graphic cards in order to speed up the calculation. In an initial stage, the diffuser tube of the T99 turbine case study was analyzed in order to establish the best configuration to make the best use of the computational resources. A series of tests were carried out to determine the influence that the computational architecture of the graphic cards, the algebraic solver type and the available memory have on the calculation time and accuracy of the results. Finally, additional steady-state and transient analyses were performed using the Francis 99 turbine case study as a reference in order to corroborate the behavior of the test bench in a wide range of operating conditions.

Key words: Turbine, diffuser tube, high-performance computing, graphics cards, simulation.

Índice General.

Dedicatorias.....	i
Agradecimientos.....	ii
Resumen.....	iii
Abstract.....	iv
Índice General.....	v
Índice de Tablas.....	viii
Índice de Figuras.....	ix
Objetivos.....	1
General:	1
Específicos:	1
Planteamiento del Problema.....	2
Justificación.....	3
Hipótesis.....	4
Capítulo I. Aprovechamiento Hidroeléctrico.....	5
Recursos Hídricos.	5
Energía Hidroeléctrica.	6
Potencial Hidroeléctrico.	7
Capítulo II. Pruebas de Modelo a Turbomáquinas Hidráulicas.....	9
Normatividad Aplicable.	9
Laboratorios y Bancos de Pruebas.	10
Turbomaquinaria Hidráulica y CFD.	11
Capítulo III. Dinámica de Fluidos Computacional.....	17
Ecuaciones Gobernantes.	18
Turbulencia.	23
Propiedades Promedio.	23
Modelos de Turbulencia.	27
Método de Volumen Finito.	29
Solucionadores de Ecuaciones Lineales.	34

Métodos Iterativos.	35
Métodos Multigrid.	37
Capítulo IV. Descripción del Software y Hardware.	39
Ubuntu.	39
GPUs en Ubuntu.	39
Software en Ubuntu.	40
OpenFOAM.	40
foam-extend.	41
Entorno de OpenFOAM.	41
Paralelización en OpenFOAM.	43
CFD en OpenFOAM.	43
RapidCFD.	46
cufflink.	46
CUDA.	47
Programación en CUDA.	48
GPUs en CUDA.	49
Capítulo V. Casos de Estudio.	52
Turbina T-99.	52
Turbina Francis-99.	53
Capítulo VI. Resultados.	56
Aceleración de Calculo.	56
Solucionadores Lineales y Velocidad de Calculo.	64
Solucionadores Lineales y Uso de Memoria.	68
Exactitud de los Cálculos.	76
RapidCFD vs. cufflink.	84
Simulaciones RANS.	87
Cálculos en estado estable utilizando MRF	94
Perfiles de velocidad en estado estable	99
Simulaciones URANS.	105
Cálculos en estado transitorio utilizando mallas deslizantes	105

Solución transitoria utilizando el modelo de turbulencia $k-\omega$ SST	110
Solución transitoria utilizando el modelo de turbulencia $k-\varepsilon$ Realizable	116
Solución transitoria utilizando el modelo de turbulencia $k-\varepsilon$ RNG	122
Perfiles de velocidad en estado transitorio	126
Conclusiones.....	131
Referencias.	134
Anexos.....	138

Índice de Tablas.

Tabla 1. Capacidad de almacenamiento y uso de las principales presas de México, 2014. Datos del estado de Michoacán [2].....	5
Tabla 2. Principales presas de Michoacán [3].....	6
Tabla 3. Volúmenes declarados para el pago de derechos por la producción de energía hidroeléctrica, 2005-2014 [4].	6
Tabla 4. Participación de las Energías Renovables en la Generación Total (GWh) [5].....	7
Tabla 5. Potencial Hidroeléctrico Nacional con pequeñas Centrales [7].	8
Tabla 6. Modelos de Turbulencia según el número de ecuaciones	28
Tabla 7. Unidades base para SI y USCS [27].....	42
Tabla 8. Solvers lineales [27].	44
Tabla 9. Opciones para pre acondicionador [27].....	45
Tabla 10. Opciones de Smother [27].....	45
Tabla 11. Parámetros observados a las condiciones de operación PL, BEP and HL [12]. ..	54
Tabla 12. Características de las Workstations WSPAC y WSGAL	57
Tabla 13. Descomposición de dominio	58
Tabla 14. Características de la Workstation WSMOL	61
Tabla 15. Parámetros de simulación de RapidCFD en Workstations usando GPUs.....	64
Tabla 16 . Parámetros para generación de dominios computacionales.	68
Tabla 17. Cálculo de error contra valores experimentales de Cpwall	83
Tabla 18. Datos de las condiciones de funcionamiento de la serie de talleres Francis 99. ..	88
Tabla 19. Comparación de datos experimentales frente a CFD para BEP.	90

Índice de Figuras.

Figura 1. Capacidad Instalada y Generacion Bruta de Centrales Hidroelectricas, 2004-2014 (MW y GWh/Año [5].	7
Figura 2. Distribución de sitios potenciales para centrales mini hidroeléctricas [7].	8
Figura 3. Instalaciones Centro Hidroeléctrico Porjus en Suecia [8].	10
Figura 4. Equipo de prueba turbina / bomba-turbina Francis. Vista completa de la carcasa en espiral, el cono del tubo de aspiración, el codo y la descarga de la turbina conectada al tanque aguas abajo. Las mediciones de presión y velocidad (LDV y PIV) pueden realizarse en la sección transparente del cono del tubo de aspiración [12].	11
Figura 5. Comparación de los valores experimentales y numéricos de la eficiencia hidráulica en la carga parcial, BEP, y las condiciones de funcionamiento de alta carga [13].	12
Figura 6. Estimación de la capa limite en los lados de succión y presión de los alabes guía con diferentes valores de y^+ y modelos de turbulencia [14].	13
Figura 7. Contornos del coeficiente de presión sobre los lados de succión y presión de los alabes operando a sobre velocidad ($\alpha = 12.4^\circ$) [15].	13
Figura 8. Comparación de la eficiencia de turbina entre la predicción numérica y los resultados experimentales [16].	14
Figura 9. Comparación entre la eficiencia calculada y la eficiencia medida [17].	15
Figura 10. Magnitudes de velocidad media en la sección media del difusor. CFD (Izquierda), LDV (Derecha) para el BEP (60 m ³ /h) con el modelo k-e Realizable [18].	16
Figura 11. Aglomeración de un nivel de malla fina para formar un nivel de malla grueso [22]	38
Figura 12. Comparación del tiempo de ejecución usando CPUs (OpenFOAM) y GPUs (RapidCFD) para un caso de 4 millones de celdas [28].	46
Figura 13. Speedup mediante el uso de GPUs (RapidCFD) para un caso de 4 millones de celdas [28].	46
Figura 14. Operaciones de punto flotante por segundo para CPU y GPU [36].	49
Figura 15. Ancho de banda para memoria de CPU y GPU [36].	50
Figura 16. Comparación de asignación de transistores para procesamiento de datos en CPU y GPU [36].	50
Figura 17. Esquemático del tubo de aspiración [37].	52
Figura 18. Localización de sensores en el banco de pruebas de la turbina Francis [12].	53
Figura 19. Turbina Tokke [12].	54
Figura 20. Diagrama de colina para la eficiencia [12] [38].	55
Figura 21. Residuales de ecuaciones de momento y continuidad en OpenFOAM v1706 utilizando CPU.	58
Figura 22. Residuales de ecuaciones de momento y continuidad en RapidCFD utilizando GPU.	58

Figura 23. Execution time del benchmark de la turbina T99.	59
Figura 24. Wall clock time del benchmark de la turbina T99.	59
Figura 25. Speedup del modelo computacional.....	59
Figura 26. Fracción de paralelismo del modelo computacional.....	60
Figura 27. Execution time del benchmark de la turbina T99 empleando diferentes modelos de GPU (WSPAC/C1060, WSGAL/M2090, WSMOL/K40).	61
Figura 28. Wall clock time del benchmark de la turbina T99 empleando diferentes modelos de GPU (WSPAC/C1060, WSGAL/M2090, WSMOL/K40).	61
Figura 29. Wall clock time vs execution time del benchmark de la turbina T99 empleando diferentes modelos de GPU (WSPAC/C1060, WSGAL/M2090, WSMOL/K40).	62
Figura 30. Speedup por work station del modelo computacional de la turbina T99 empleando diferentes modelos de GPU (WSPAC/C1060, WSGAL/M2090, WSMOL/K40).	62
Figura 31. Speedup global del modelo computacional de la turbina T99 empleando diferentes modelos de GPU (WSPAC/C1060, WSGAL/M2090, WSMOL/K40).	63
Figura 32. Wall clock time de primer time step (iteración) en WSPAC con RapidCFD y GPUs C1060.	65
Figura 33. Wall clock time de primer time step (iteración) en WSGAL con RapidCFD y GPUs M2090.	65
Figura 34. Wall clock time de ultimo time step (iteración) en WSPAC con RapidCFD y GPUs C1060.	65
Figura 35. Wall clock time de ultimo time step (iteración) en WSGAL con RapidCFD y GPUs M2090.	65
Figura 36. Tiempo por time step (iteración) en WSPAC con RapidCFD y GPUs C1060.	65
Figura 37. Tiempo por time step (iteración) en WSGAL con RapidCFD y GPUs M2090.	65
Figura 38. Wall clock time de primer time step (iteración) en WSMOL con RapidCFD y GPUs K40.	66
Figura 39. Wall clock time de ultimo time step (iteración) en WSMOL con RapidCFD y GPUs K40.	66
Figura 40. Tiempo por time step (iteración) en WSMOL con RapidCFD y GPUs K40.	66
Figura 41. Wall clock time de ultimo time step (iteración) con RapidCFD empleando diferentes modelos y numero de GPU. Solucionador lineal GAMG, suavizador GaussSeidel.	67
Figura 42. Wall clock time de ultimo time step (iteración) con RapidCFD empleando diferentes modelos y numero de GPU. Solucionador lineal PCG, pre acondicionador diagonal.	67
Figura 43. Wall clock time de ultimo time step (iteración) con RapidCFD empleando diferentes modelos y numero de GPU. Solucionador lineal GAMG, suavizador Jacobi.	67
Figura 44. Wall clock time de ultimo time step (iteración) con RapidCFD empleando diferentes modelos y numero de GPU. Solucionador lineal PCG, pre acondicionador AINV.	67

Figura 45. Discretización de los ejes de los grupos <i>A</i> y <i>a</i> .	69
Figura 46. Discretización de los ejes del grupo <i>B</i> .	69
Figura 47. Discretización de los ejes del grupo <i>C</i> .	69
Figura 48. Discretización de los ejes del grupo <i>D</i> .	70
Figura 49. Discretización de los ejes del nivel 1-5.	70
Figura 50. Discretización de los ejes del nivel 6-16 (izquierda) y del dominio computacional completo (derecha).	70
Figura 51. Memoria RAM requerida para la solución de dominios computacionales.	71
Figura 52. Memoria RAM requerida para la solución de dominios computacionales y regresión lineal para cada solucionador con 1 GPU. A) PCG/PBiCG, $M=83.79+0.001148S$ B) GAMG, $M=64.47+0.001424S$. Donde M es la memoria requerida y S el tamaño del dominio computacional.	72
Figura 53. Memoria RAM requerida para la solución de dominios computacionales y regresión lineal para cada solucionador con 2 GPUs. A) PCG/PBiCG, $M=203.69+0.001151S$ B) GAMG, $M=115.35+0.001416S$. Donde M es la memoria requerida y S el tamaño del dominio computacional.	72
Figura 54. Memoria RAM requerida para la solución de dominios computacionales y regresión lineal para cada solucionador con 3 GPUs. A) PCG/PBiCG, $M=293.60+0.001152S$ B) GAMG, $M=303.80+0.001437S$. Donde M es la memoria requerida y S el tamaño del dominio computacional.	73
Figura 55. Memoria RAM requerida para la solución de dominios computacionales y regresión lineal para cada solucionador con 4 GPUs. A) PCG/PBiCG, $M=379.09+0.001159S$ B) GAMG, $M=407.87+0.001434S$. Donde M es la memoria requerida y S el tamaño del dominio computacional.	73
Figura 56. Memoria RAM requerida para la solución de dominios computacionales y regresión lineal utilizando el solucionador PCG/PBiCG con 1, 2, 3 y 4 GPUs.	74
Figura 57. Memoria RAM requerida para la solución de dominios computacionales y regresión lineal utilizando el solucionador GAMG con 1, 2, 3 y 4 GPUs.	74
Figura 58. Memoria RAM requerida para la solución de dominios computacionales y propuesta de línea de ajuste para cada solucionador. A) PCG/PBiCG, $M=150N+0.001165S$ B) GAMG, $M=150N+0.001469S$. Donde M es la memoria requerida y S el tamaño del dominio computacional y N el número de GPUs empleadas que corresponde al número de dominios en paralelo.	75
Figura 59. Contornos de velocidad con RapidCFD y solucionador GAMG con suavizador GasussSeidel.	76
Figura 60. Contornos de velocidad con RapidCFD y solucionador GAMG con suavizador GasussSeidel.	76
Figura 61. Contornos de velocidad con RapidCFD y solucionador GAMG con suavizador Jacobi.	76

Figura 62. Contornos de velocidad con RapidCFD y solucionador GAMG con suavizador <i>Jacobi</i> .	76
Figura 63. Contornos de velocidad con RapidCFD y solucionador PCG/PBiCG con preconditionador AINV.	77
Figura 64. Contornos de velocidad con RapidCFD y solucionador PCG/PBiCG con preconditionador AINV.	77
Figura 65. Contornos de velocidad con RapidCFD y solucionador PCG/PBiCG con preconditionador <i>diagonal</i> .	77
Figura 66. Contornos de velocidad con RapidCFD y solucionador PCG/PBiCG con preconditionador <i>diagonal</i> .	77
Figura 67. Contornos de velocidad con OpenFOAMv1706 y solucionador GAMG con suavizador GasussSeidel.	78
Figura 68. Contornos de velocidad con OpenFOAMv1706 y solucionador GAMG con suavizador GasussSeidel.	78
Figura 69. Contornos de velocidad con OpenFOAMv1706 y solucionador PCG/PBiCG con preconditionador <i>diagonal</i> .	78
Figura 70. Contornos de velocidad con OpenFOAMv1706 y solucionador PCG/PBiCG con preconditionador <i>diagonal</i> .	78
Figura 71. Líneas centrales superior e inferior del tubo difusor.	79
Figura 72. Secciones de medición del tubo difusor.	79
Figura 73. Contornos de velocidad axial U_x y vectores de la componente de velocidad horizontal U_y en la sección CsII (el flujo se mueve hacia el lector). De izquierda a derecha: 1) Experimental, 2) T99 CFD de referencia y 3) malla de 1 millón de celdas.	79
Figura 74. Comparación de la recuperación de presión en la línea central inferior y superior del tubo de aspiración utilizando solucionadores GAMG con una malla de 1 millón de celdas.	81
Figura 75. Comparación de la recuperación de presión en la línea central inferior y superior del tubo de aspiración utilizando solucionador PCG/PBiCG con una malla de 1 millón de celdas.	81
Figura 76. Contornos de velocidad normal U_x y vectores del componente de velocidad horizontal U_y en la sección CsII (el flujo se mueve hacia el lector). De izquierda a derecha: 1) Experimental, 2) T99 CFD de referencia y 3) malla refinada de 4 millones de celdas.	82
Figura 77. Comparación de la recuperación de presión en la línea central inferior del tubo de aspiración con diferentes tamaños de rejilla.	82
Figura 78. Comparación de la recuperación de presión en la línea central superior del tubo de aspiración con diferentes tamaños de rejilla.	82
Figura 79. Gráficos de C_p : extrapolaciones de primer ϕ (1) y segundo ϕ (2) orden, soluciones CFD ϕ y valores experimentales ϕ (E) para el análisis de convergencia de malla en la línea central superior.	83

Figura 80. Gráficos de C_p : extrapolaciones de primer ϕ (1) y segundo ϕ (2) orden, soluciones CFD ϕ y valores experimentales ϕ (E) para el análisis de convergencia de malla en la línea central inferior.	83
Figura 81. Tiempo de cálculo por time/step para cada dominio computacional utilizando diferentes software (RapidCFD, cufflink) y hardware (Tesla C1060, M2090, K40).....	85
Figura 82. Tiempo de cálculo total para cada dominio computacional utilizando diferentes software (RapidCFD, cufflink) y hardware (Tesla C1060, M2090, K40).	85
Figura 83. Memoria usada en GPUs para cada dominio computacional utilizando diferentes software (RapidCFD, cufflink) y hardware (Tesla C1060, M2090, K40).	86
Figura 84. Memoria usada en GPUs vs tiempo de cómputo para cada dominio computacional utilizando diferentes software (RapidCFD, cufflink) y hardware (Tesla C1060, M2090, K40).	86
Figura 85. Vistas isométricas de la turbina Francis 99 (izquierda), de cerca (centro) y de los alabes del rodete (izquierda).....	87
Figura 86. <i>Speedup</i> en la condición de BEP usando varios subprocesos.	89
Figura 87. Tiempo del reloj de pared para la condición BEP usando varios subprocesos... ..	89
Figura 88. Contornos de valores de presión (izquierda) y velocidad (derecha) en la turbina Francis 99 para la condición BEP.	91
Figura 89. Contornos de valores de presión (izquierda) y velocidad (derecha) en carcasa, distribuidor y rodete de la turbina Francis 99 para la condición BEP.	91
Figura 90. Contornos de valores de fuerza (izquierda) y torque (derecha) en el rodete de la turbina Francis 99 para la condición BEP.	91
Figura 91. Tiempo del reloj de pared para diferentes condiciones de operación usando RapidCFD.....	92
Figura 92. Tiempo por paso de tiempo para diferentes condiciones de operación usando RapidCFD.....	92
Figura 93. Valores de presión diferencial para diferentes condiciones de operación usando RapidCFD vs. datos experimentales.....	93
Figura 94. Valores de torque del rodete para diferentes condiciones de operación usando RapidCFD vs. datos experimentales.....	93
Figura 95. Valores de altura neta para diferentes condiciones de operación usando RapidCFD vs. datos experimentales.	93
Figura 96. Valores de eficiencia hidráulica para diferentes condiciones de operación usando RapidCFD vs. datos experimentales.....	93
Figura 97. Valores de presión diferencial para diferentes modelos de turbulencia usando RapidCFD.....	94
Figura 98. Valores de torque para diferentes modelos de turbulencia usando RapidCFD... ..	94
Figura 99. Valores de altura neta para diferentes modelos de turbulencia usando RapidCFD.	94

Figura 100. Valores de eficiencia hidráulica para diferentes modelos de turbulencia usando RapidCFD.....	94
Figura 101. Valores de presión diferencial para diferentes modelos de turbulencia usando RapidCFD.....	95
Figura 102. Valores de torque para diferentes modelos de turbulencia usando RapidCFD.	95
Figura 103. Valores de altura neta para diferentes modelos de turbulencia usando RapidCFD.	95
Figura 104. Valores de eficiencia hidráulica para diferentes modelos de turbulencia usando RapidCFD.....	95
Figura 105. Valores de presión diferencial para diferentes modelos de turbulencia usando RapidCFD.....	96
Figura 106. Valores de torque para diferentes modelos de turbulencia usando RapidCFD.	96
Figura 107. Valores de altura neta para diferentes modelos de turbulencia usando RapidCFD.	96
Figura 108. Valores de eficiencia hidráulica para diferentes modelos de turbulencia usando RapidCFD.....	96
Figura 109. Valores de presión diferencial para diferentes modelos de turbulencia usando RapidCFD.....	97
Figura 110. Valores de torque para diferentes modelos de turbulencia usando RapidCFD.	97
Figura 111. Valores de altura neta para diferentes modelos de turbulencia usando RapidCFD.	97
Figura 112. Valores de eficiencia hidráulica para diferentes modelos de turbulencia usando RapidCFD.....	97
Figura 113. Valores de presión diferencial para diferentes modelos de turbulencia usando RapidCFD.....	98
Figura 114. Valores de torque para diferentes modelos de turbulencia usando RapidCFD.	98
Figura 115. Valores de altura neta para diferentes modelos de turbulencia usando RapidCFD.	98
Figura 116. Valores de eficiencia hidráulica para diferentes modelos de turbulencia usando RapidCFD.....	98
Figura 117. Valores de presión diferencial para diferentes modelos de turbulencia usando RapidCFD.....	99
Figura 118. Valores de torque para diferentes modelos de turbulencia usando RapidCFD.	99
Figura 119. Valores de altura neta para diferentes modelos de turbulencia usando RapidCFD.	99
Figura 120. Valores de eficiencia hidráulica para diferentes modelos de turbulencia usando RapidCFD.....	99
Figura 121. Coordenadas globales para las ubicaciones de medición, geometría y malla.	100
Figura 122. Secciones de medición mediante PIV en el cono del tubo de desfogue.	100

Figura 123. Valores de velocidad horizontal en PIV Line 1 con diferentes modelos de turbulencia a carga parcial PL2.	101
Figura 124. Valores de velocidad vertical en PIV Line 1 con diferentes modelos de turbulencia a carga parcial PL2.	101
Figura 125. Valores de velocidad horizontal en PIV Line 2 con diferentes modelos de turbulencia a carga parcial PL2.	101
Figura 126. Valores de velocidad vertical en PIV Line 2 con diferentes modelos de turbulencia a carga parcial PL2.	101
Figura 127. Valores de velocidad horizontal en PIV Line 3 con diferentes modelos de turbulencia a carga parcial PL2.	101
Figura 128. Valores de velocidad vertical en PIV Line 3 con diferentes modelos de turbulencia a carga parcial PL2.	101
Figura 129. Valores de velocidad horizontal en PIV Line 1 con diferentes modelos de turbulencia en eficiencia máxima BEP*.....	102
Figura 130. Valores de velocidad vertical en PIV Line 1 con diferentes modelos de turbulencia en eficiencia máxima BEP*.....	102
Figura 131. Valores de velocidad horizontal en PIV Line 2 con diferentes modelos de turbulencia en eficiencia máxima BEP*.....	102
Figura 132. Valores de velocidad vertical en PIV Line 2 con diferentes modelos de turbulencia en eficiencia máxima BEP*.....	102
Figura 133. Valores de velocidad horizontal en PIV Line 3 con diferentes modelos de turbulencia en eficiencia máxima BEP*.....	102
Figura 134. Valores de velocidad vertical en PIV Line 3 con diferentes modelos de turbulencia en eficiencia máxima BEP*.....	102
Figura 135. Valores de velocidad horizontal en PIV Line 1 con diferentes modelos de turbulencia a carga máxima HL*.	103
Figura 136. Valores de velocidad vertical en PIV Line 1 con diferentes modelos de turbulencia a carga máxima HL*.	103
Figura 137. Valores de velocidad horizontal en PIV Line 2 con diferentes modelos de turbulencia a carga máxima HL*.	103
Figura 138. Valores de velocidad vertical en PIV Line 2 con diferentes modelos de turbulencia a carga máxima HL*.	103
Figura 139. Valores de velocidad horizontal en PIV Line 3 con diferentes modelos de turbulencia a carga máxima HL*.	104
Figura 140. Valores de velocidad vertical en PIV Line 3 con diferentes modelos de turbulencia a carga máxima HL*.	104
Figura 141. Valores de presión diferencial para diferentes modelos de turbulencia usando RapidCFD.....	105
Figura 142. Valores de torque para diferentes modelos de turbulencia usando RapidCFD.	105

Figura 143. Valores de altura neta para diferentes modelos de turbulencia usando RapidCFD.	106
Figura 144. Valores de eficiencia hidráulica para diferentes modelos de turbulencia usando RapidCFD.	106
Figura 145. Valores de presión diferencial para diferentes modelos de turbulencia usando RapidCFD.	106
Figura 146. Valores de torque para diferentes modelos de turbulencia usando RapidCFD.	106
Figura 147. Valores de altura neta para diferentes modelos de turbulencia usando RapidCFD.	106
Figura 148. Valores de eficiencia hidráulica para diferentes modelos de turbulencia usando RapidCFD.	106
Figura 149. Valores de presión diferencial para diferentes modelos de turbulencia usando RapidCFD.	107
Figura 150. Valores de torque para diferentes modelos de turbulencia usando RapidCFD.	107
Figura 151. Valores de altura neta para diferentes modelos de turbulencia usando RapidCFD.	107
Figura 152. Valores de eficiencia hidráulica para diferentes modelos de turbulencia usando RapidCFD.	107
Figura 153. Valores de presión diferencial para diferentes modelos de turbulencia usando RapidCFD.	108
Figura 154. Valores de torque para diferentes modelos de turbulencia usando RapidCFD.	108
Figura 155. Valores de altura neta para diferentes modelos de turbulencia usando RapidCFD.	108
Figura 156. Valores de eficiencia hidráulica para diferentes modelos de turbulencia usando RapidCFD.	108
Figura 157. Valores de presión diferencial para diferentes modelos de turbulencia usando RapidCFD.	109
Figura 158. Valores de torque para diferentes modelos de turbulencia usando RapidCFD.	109
Figura 159. Valores de altura neta para diferentes modelos de turbulencia usando RapidCFD.	109
Figura 160. Valores de eficiencia hidráulica para diferentes modelos de turbulencia usando RapidCFD.	109
Figura 161. Valores de presión diferencial para diferentes modelos de turbulencia usando RapidCFD.	110
Figura 162. Valores de torque para diferentes modelos de turbulencia usando RapidCFD.	110

Figura 163. Valores de altura neta para diferentes modelos de turbulencia usando RapidCFD. 110

Figura 164. Valores de eficiencia hidráulica para diferentes modelos de turbulencia usando RapidCFD. 110

Figura 165. Valores de presión diferencial para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega SST$ en condición de carga parcial PL1. 111

Figura 166. Valores de torque para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega SST$ en condición de carga parcial PL1. 111

Figura 167. Valores de altura neta para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega SST$ en condición de carga parcial PL1. 111

Figura 168. Valores de eficiencia hidráulica para para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega SST$ en condición de carga parcial PL1. 111

Figura 169. Valores de presión diferencial para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega SST$ en condición de carga parcial PL2. 112

Figura 170. Valores de torque para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega SST$ en condición de carga parcial PL2. 112

Figura 171. Valores de altura neta para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega SST$ en condición de carga parcial PL2. 112

Figura 172. Valores de eficiencia hidráulica para para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega SST$ en condición de carga parcial PL2. 112

Figura 173. Valores de presión diferencial para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega SST$ en condición de eficiencia máxima BEP. 113

Figura 174. Valores de torque para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega SST$ en condición de eficiencia máxima BEP. 113

Figura 175. Valores de altura neta para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega SST$ en condición de eficiencia máxima BEP. 113

Figura 176. Valores de eficiencia hidráulica para para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega SST$ en condición de eficiencia máxima BEP. 113

Figura 177. Valores de presión diferencial para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega SST$ en condición de eficiencia máxima BEP*. 114

Figura 178. Valores de torque para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega SST$ en condición de eficiencia máxima BEP*. 114

Figura 179. Valores de altura neta para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega SST$ en condición de eficiencia máxima BEP*. 114

Figura 180. Valores de eficiencia hidráulica para para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega SST$ en condición de eficiencia máxima BEP*. 114

Figura 181. Valores de presión diferencial para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega SST$ en condición de carga máxima HL. 115

Figura 182. Valores de torque para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega SST$ en condición de carga máxima HL. 115

Figura 183. Valores de altura neta para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega$ SST en condición de carga máxima HL..... 115

Figura 184. Valores de eficiencia hidráulica para para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega$ SST en condición de carga máxima HL..... 115

Figura 185. Valores de presión diferencial para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega$ SST en condición de carga máxima HL*..... 116

Figura 186. Valores de torque para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega$ SST en condición de carga máxima HL*..... 116

Figura 187. Valores de altura neta para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega$ SST en condición de carga máxima HL*..... 116

Figura 188. Valores de eficiencia hidráulica para para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega$ SST en condición de carga máxima HL*..... 116

Figura 189. Valores de presión diferencial para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ Realizable en condición de carga parcial PL1. 117

Figura 190. Valores de torque para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ Realizable en condición de carga parcial PL1. 117

Figura 191. Valores de altura neta para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ Realizable en condición de carga parcial PL1. 117

Figura 192. Valores de eficiencia hidráulica para para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ Realizable en condición de carga parcial PL1. 117

Figura 193. Valores de presión diferencial para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ Realizable en condición de carga parcial PL2. 118

Figura 194. Valores de torque para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ Realizable en condición de carga parcial PL2. 118

Figura 195. Valores de altura neta para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ Realizable en condición de carga parcial PL2. 118

Figura 196. Valores de eficiencia hidráulica para para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ Realizable en condición de carga parcial PL2. 118

Figura 197. Valores de presión diferencial para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ Realizable en condición de eficiencia máxima BEP..... 119

Figura 198. Valores de torque para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ Realizable en condición de eficiencia máxima BEP. 119

Figura 199. Valores de altura neta para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ Realizable en condición de eficiencia máxima BEP. 119

Figura 200. Valores de eficiencia hidráulica para para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ Realizable en condición de eficiencia máxima BEP. 119

Figura 201. Valores de presión diferencial para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ Realizable en condición de eficiencia máxima BEP*..... 120

Figura 202. Valores de torque para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ *Realizable* en condición de eficiencia máxima BEP*. 120

Figura 203. Valores de altura neta para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ *Realizable* en condición de eficiencia máxima BEP*. 120

Figura 204. Valores de eficiencia hidráulica para para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ *Realizable* en condición de eficiencia máxima BEP*. 120

Figura 205. Valores de presión diferencial para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ *Realizable* en condición de carga máxima HL..... 121

Figura 206. Valores de torque para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ *Realizable* en condición de carga máxima HL. 121

Figura 207. Valores de altura neta para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ *Realizable* en condición de carga máxima HL. 121

Figura 208. Valores de eficiencia hidráulica para para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ *Realizable* en condición de carga máxima HL..... 121

Figura 209. Valores de presión diferencial para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ *Realizable* en condición de carga máxima HL*..... 122

Figura 210. Valores de torque para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ *Realizable* en condición de carga máxima HL*. 122

Figura 211. Valores de altura neta para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ *Realizable* en condición de carga máxima HL*. 122

Figura 212. Valores de eficiencia hidráulica para para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ *Realizable* en condición de carga máxima HL*..... 122

Figura 213. Valores de presión diferencial para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ *RNG* en condición de eficiencia máxima BEP..... 123

Figura 214. Valores de torque para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ *RNG* en condición de eficiencia máxima BEP. 123

Figura 215. Valores de altura neta para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ *RNG* en condición de eficiencia máxima BEP. 123

Figura 216. Valores de eficiencia hidráulica para para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ *RNG* en condición de eficiencia máxima BEP..... 123

Figura 217. Valores de presión diferencial para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ *RNG* en condición de eficiencia máxima BEP*..... 124

Figura 218. Valores de torque para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ *RNG* en condición de eficiencia máxima BEP*. 124

Figura 219. Valores de altura neta para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ *RNG* en condición de eficiencia máxima BEP*. 124

Figura 220. Valores de eficiencia hidráulica para para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ *RNG* en condición de eficiencia máxima BEP*..... 124

Figura 221. Valores de presión diferencial para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon RNG$ en condición de carga máxima HL. 125

Figura 222. Valores de torque para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon RNG$ en condición de carga máxima HL. 125

Figura 223. Valores de altura neta para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon RNG$ en condición de carga máxima HL. 125

Figura 224. Valores de eficiencia hidráulica para para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon RNG$ en condición de carga máxima HL. 125

Figura 225. Valores de presión diferencial para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon RNG$ en condición de carga máxima HL*. 126

Figura 226. Valores de torque para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon RNG$ en condición de carga máxima HL*. 126

Figura 227. Valores de altura neta para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon RNG$ en condición de carga máxima HL*. 126

Figura 228. Valores de eficiencia hidráulica para para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon RNG$ en condición de carga máxima HL*. 126

Figura 229. Valores de velocidad horizontal en PIV Line 1 con diferentes modelos de turbulencia a carga parcial PL2. 127

Figura 230. Valores de velocidad vertical en PIV Line 1 con diferentes modelos de turbulencia a carga parcial PL2. 127

Figura 231. Valores de velocidad horizontal en PIV Line 2 con diferentes modelos de turbulencia a carga parcial PL2. 127

Figura 232. Valores de velocidad vertical en PIV Line 2 con diferentes modelos de turbulencia a carga parcial PL2. 127

Figura 233. Valores de velocidad horizontal en PIV Line 3 con diferentes modelos de turbulencia a carga parcial PL2. 128

Figura 234. Valores de velocidad vertical en PIV Line 3 con diferentes modelos de turbulencia a carga parcial PL2. 128

Figura 235. Valores de velocidad horizontal en PIV Line 1 con diferentes modelos de turbulencia en eficiencia máxima BEP*. 128

Figura 236. Valores de velocidad vertical en PIV Line 1 con diferentes modelos de turbulencia en eficiencia máxima BEP*. 128

Figura 237. Valores de velocidad horizontal en PIV Line 2 con diferentes modelos de turbulencia en eficiencia máxima BEP*. 129

Figura 238. Valores de velocidad vertical en PIV Line 2 con diferentes modelos de turbulencia en eficiencia máxima BEP*. 129

Figura 239. Valores de velocidad horizontal en PIV Line 3 con diferentes modelos de turbulencia en eficiencia máxima BEP*. 129

Figura 240. Valores de velocidad vertical en PIV Line 3 con diferentes modelos de turbulencia en eficiencia máxima BEP*. 129

Figura 241. Valores de velocidad horizontal en PIV Line 1 con diferentes modelos de turbulencia a carga máxima HL*. 130

Figura 242. Valores de velocidad vertical en PIV Line 1 con diferentes modelos de turbulencia a carga máxima HL*. 130

Figura 243. Valores de velocidad horizontal en PIV Line 2 con diferentes modelos de turbulencia a carga máxima HL*. 130

Figura 244. Valores de velocidad vertical en PIV Line 2 con diferentes modelos de turbulencia a carga máxima HL*. 130

Figura 245. Valores de velocidad horizontal en PIV Line 3 con diferentes modelos de turbulencia a carga máxima HL*. 130

Figura 246. Valores de velocidad vertical en PIV Line 3 con diferentes modelos de turbulencia a carga máxima HL*. 130

Objetivos.

General:

- Crear un banco de pruebas virtual basado en HPC para pruebas de modelo a turbinas hidráulicas tipo Francis que incluya la tubería de presión, carcasa espiral, distribuidor, alabes guía, rodete y difusor.

Específicos:

- Acoplar las tarjetas GPU para acelerar el cálculo en CFD.
- Encontrar el mallado óptimo de acuerdo a los recursos computacionales disponibles y realizar el análisis de convergencia.
- Establecer las condiciones del modelo numérico para la simulación del caso de estudio (modelo de turbulencia, funciones de pared, método de solución, discretización espacial, etc.).
- Validar los resultados obtenidos en la simulación con datos experimentales de la turbina Francis 99 usando leyes de similitud.
- Analizar el efecto fluido dinámico obtenido en CFD para verificar el desempeño adecuado de los componentes del banco de pruebas virtual.

Planteamiento del Problema.

Actualmente los recursos hídricos nacionales no son aprovechados para la generación eléctrica, ya que en algunos casos se requiere que tanto la turbina como equipos auxiliares e instalaciones sean rehabilitados y/o modernizados, y en algunos otros casos no existe una infraestructura previa.

Sin embargo, buscando el aprovechamiento de estos recursos, el re-diseño de estas unidades requeriría de ensayos y pruebas para la nueva turbina hidráulica. Pero a lo largo de los años, esta ha sido una actividad que la mayoría de las veces ha sido contratada a empresas privadas extranjeras. La razón de lo anterior es que en México no existen laboratorios dedicados especialmente a realizar ensayos y pruebas en turbinas hidráulicas, ya sea del tipo Francis, Kaplan o Pelton. Como consecuencia de esto, no se cuenta con el conocimiento de cómo realizar estas pruebas de operación, es decir el “*know how*”, siendo necesario calificar personal para realizar tales actividades. Sumado a lo anterior, las pruebas de desempeño de nuevos diseños de turbinas hidráulicas o re-potenciadas es una actividad que consume gran parte del tiempo y costo de un proyecto hidroeléctrico. Lo anterior se debe a la necesidad de construir modelos físicos a escala en bancos de pruebas basados en leyes de similitud, lo que conlleva a diseñar y construir también la instalación que proporcione el caudal y carga requeridos para la operación de dichos modelos de pruebas.

Por todo lo anterior, es necesaria la construcción de un banco de pruebas virtual basado en HPC que nos permita evaluar diseños de turbinas tipo Francis en forma rápida y lo más exacta posible.

Justificación.

Debido a que en México no existen laboratorios especializados para evaluar el desempeño de turbinas hidráulicas capaces de aprovechar los recursos hídricos disponibles, es necesario el desarrollo y construcción de un banco de pruebas virtual basado en HPC para turbinas hidráulicas tipo Francis. El banco de pruebas incluirá los siguientes componentes: tubería de presión, carcasa espiral, distribuidor, alabes guía, rodete y difusor; sin considerar los equipos auxiliares como: sistema de lubricación, sistema de enfriamiento, sistemas eléctricos de control, regulación y excitación, así como el generador eléctrico en el modelo virtual. Se tomara como base la turbina Francis 99.

Los resultados de esta investigación aportarán el “know how” para la evaluación de turbinas hidráulicas tipo Francis en diferentes condiciones de operación. Además, se obtendrá experiencia en el desarrollo de pruebas de modelo, las cuáles son indispensables para evaluar el desempeño de cada turbina.

Hipótesis.

Un banco de ensayo virtual basado en HPC para la turbina Francis 99, permitirá la realización de pruebas de modelo a diferentes condiciones de operación de manera rápida.

Capítulo I. Aprovechamiento Hidroeléctrico.

Recursos Hídricos.

De acuerdo a la CONAGUA (Comisión Nacional del Agua), en México se tienen 5,163 presas y bordos con varios usos, entre ellos la generación de energía eléctrica [1]. En la Tabla 1 se muestran las presas más grandes del estado de Michoacán [2].

Tabla 1. Capacidad de almacenamiento y uso de las principales presas de México, 2014. Datos del estado de Michoacán [2].

Nombre Oficial	Capacidad NAMO (hm ³)	Altura de Cortina (m)	Capacidad Efectiva (MW)	Corriente	Volumen útil 2014 (hm ³)
Infiernillo	9340	151.5	1000	Río Balsas	8753.95
José María Morelos	540.8	73	300	Río Balsas	501.8
Tepuxtepec	425.2	47	80	Río Lerma	365.72
Solidaridad	220.81	87		Ríos Coahuayana y Barreras	222.82
El Bosque	202.4	70		Río Zitácuaro	176.4
Melchor Ocampo	200	34		Río Angulo	192.53
Cointzio	84.8	46		Río Grande de Morelia	84.34
SanJuanico	60.48	5.7		Río Cotija	50.43
Zicuirán	50	46		Río Zicuirán	47.61
Guaracha	38.2	8		Arroyo de Las Liebres	26.7
Malpaís	23.74	6.1		Río Queréndaro	24.46
Los Olivos	21.75	37		Río Otates	19.67
Aristeo	18.34	9.2		Arroyo Seco	13.83
Mercado Agostitlán	15.95	25		Río Agostitlán	16.14
Tercer Mundo	15.58	30		Río Cachivi	14.43
Urepetiro	13	31		Río Tlazazalca	11.48
Laguna del Fresno	12.08	8.8		El Fresno	11.1
Jaripo	10.2	20		Río Jaripo	9.82
Pucuat	9.58	15		Río Pucuat	9.65
Copándaro	6.5	5.7		Canal La Estancia	6.19
Sabaneta	5.19	17		Arroyo Sabaneta	5.19

Tabla 2. Principales presas de Michoacán [3].

Subregión de planeación	Cuenca		Nombre oficial	Utilización	Capacidad Total (hm ³)	Capacidad útil (hm ³)
	Clave	Nombre				
Alto Lerma	12-D	Río Lerma 2	Tepuxtepec	Generación energía y riego	537.70	298.40
	12-E	Río Lerma 3	Laguna del Fresno	Riego	14.70	13.70
			Tercer Mundo	Riego	23.20	18.50
	12-R	C. C. Lago de Cuitzeo	Cointzio	Riego y Agua Potable	84.84	69.84
Medio Lerma	12-L	Río Angulo	Aristeo Mercado	Generación energía y riego	19.03	15.00
			Melchor Ocampo	Riego	201.38	185.00
	12-M	Río Lerma 5	Los Ángeles	Riego	11.25	5.00
	12-N	Río Lerma 6	Antonio Rodríguez L.	Riego	8.70	7.30
Bajo Lerma	12-Ñ	Río Duero	Urepetiro	Riego	10.30	7.50
			Barraje de Ibarra	Control de avenidas	14.70	11.00
	12-P	Río Lerma 7	Guaracha	Riego	19.27	22.00
			Jaripo	Riego	41.20	39.20
Medio Balsas	18-G	Río Cutzamala	Agostitlán	Riego	14.00	9.70
			Pucuto	Riego	16.55	14.95
			Tuxpan	Derivación	11.30	11.30
			Del Bosque	Riego y agua potable	20.00	1.00
	18-H	Medio Río Balsas	El Pejo	Riego	200.00	160.00
	18-I	Río Cupatitzio	Cupatitzio	Generación energía y riego	8.50	6.51
Bajo Balsas	18-K	Río Tepalcatepec	San Juanico	Riego	4.20	3.80
			Const. de Apatzingán	Riego	60.00	30.00
			Los Olivos	Riego	600.00	450.00
	18-L	Bajo Río Balsas	El Infiernillo	Gen. Hid.	21.75	21.00
			La Villita	Generación energía, riego y agua potable.	12 000.00	4 957.35
		Zicuairan	Riego	710.00	180.00	
		El Cuernamal	Riego	50.00	40.00	
				16.50	13.00	
Total					14 747.30	6 609.94

En Michoacán se tienen contabilizadas 118 presas y 143 bordos, donde el 85.7% de la capacidad de almacenamiento se encuentra en la cuenca 18-L Bajo Río Balsas [3] como se muestra en la Tabla 2.

Energía Hidroeléctrica.

Las Regiones Hidrológico-Administrativas (RHA) XI y IV tienen la concesiones más grande para generación eléctrica, donde el volumen total empleado para este uso es variable [4] según se muestra en la Tabla 3.

Tabla 3. Volúmenes declarados para el pago de derechos por la producción de energía hidroeléctrica, 2005-2014 [4].

Número de RHA	Volumen de agua declarado (hm ³)									
	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014
I	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
II	3 250.8	2 928.6	3 350.7	3 404.7	3 127.7	4 140.6	3 416.5	3 032.7	2 627.2	2 456.3
III	11 598.4	10 747.0	11 183.9	13 216.7	11 405.1	11 912.1	11 100.3	5 176.6	6 127.9	7 475.4
IV	32 141.0	21 820.3	31 099.4	30 572.8	28 059.6	34 487.9	35 539.9	32 177.7	28 126.2	29 688.3
V	1 890.3	1 949.1	2 139.6	2 244.7	2 063.4	3 528.0	16 313.8	2 028.2	1 716.9	26.3
VI	2 073.6	2 262.7	2 889.6	1 967.7	2 960.4	2 987.7	3 350.1	3 771.8	2 556.8	2 125.5
VII	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
VIII	7 361.0	4 657.8	10 516.6	13 516.9	9 030.9	11 764.6	7 741.4	5 733.5	5 598.0	10 693.3
IX	1 487.8	809.7	1 105.3	2 912.1	1 441.0	1 525.9	1 243.0	1 312.4	1 273.5	1 225.7
X	13 978.5	17 835.0	14 279.1	14 040.5	13 673.7	15 029.1	4 254.6	17 286.7	16 463.1	12 319.4
XI	41 573.3	77 245.7	46 256.8	68 793.3	64 304.7	49 406.9	81 813.4	85 197.3	48 325.9	67 007.6
XII	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
XIII	31.0	39.1	10.6	0.0	18.8	0.5	0.0	0.0	0.3	0.5
Total	115 385.8	140 294.9	122 831.6	150 669.4	136 085.3	134 783.3	164 773.0	155 716.9	112 815.9	133 018.3

Según datos de la SENER, durante 2014 la generación de energía eléctrica a partir de fuentes renovables alcanzó 55,002 GWh. Sin embargo, en 2015 se apreció una disminución en la participación de la generación hidroeléctrica como se puede observar en la Tabla 4 [5].

Tabla 4. Participación de las Energías Renovables en la Generación Total (GWh) [5].

Tecnología	2014	(%)	Ene-jun 2015	(%)
Hidroeléctrica	38,822	12.9	17,885	11.9
Bagazo	629	0.2	362	0.2
Biogás	148	<0.1	86	0.1
Geotermia	6,000	2.0	3,100	2.1
Fotovoltaica ¹	85	<0.1	91	0.1
Eólica	6,426	2.1	4,242	2.8
Sistemas Híbridos ²	N.D.	N.D.	0.1	<0.1
Cogeneración Eficiente	2,892	1.0	1,527	1.0
Sistemas rurales no interconectados ³	N.D.	N.D.	14	<0.1
SUBTOTAL DE GENERACIÓN A PARTIR DE RENOVABLES	55,002	18.2	27,307.1	18.2
SUBTOTAL DE GENERACIÓN A PARTIR DE NO RENOVABLES	246,460	81.8	122,473.4	81.8
TOTAL DE GENERACIÓN DE ELECTRICIDAD	301,462	100.0	149,780.5	100.0

Como se muestra en la Figura 1, en 2014 la capacidad instalada para generación hidroeléctrica llegó a 12,429 MW. En 10 años previos el crecimiento promedio había sido cercano al 2% anual [5].

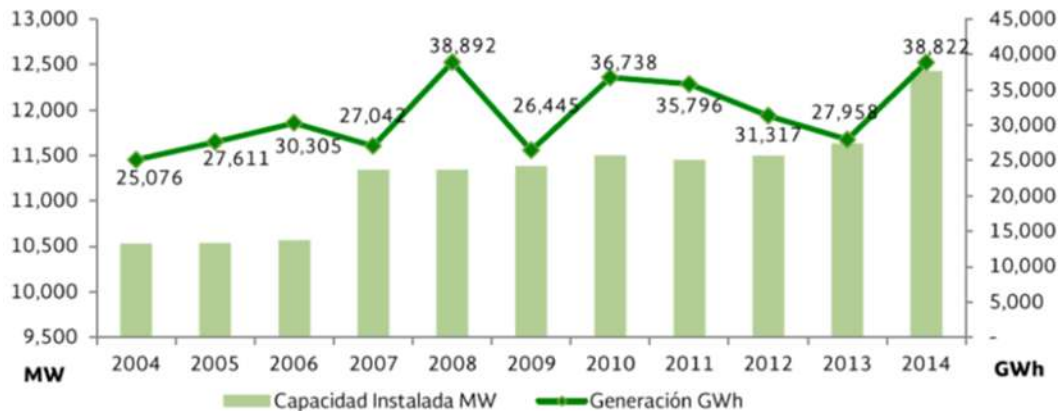


Figura 1. Capacidad Instalada y Generación Bruta de Centrales Hidroeléctricas, 2004-2014 (MW y GWh/Año) [5].

Potencial Hidroeléctrico.

Según datos de la IHA (*International Hydropower Association*) en México se cuenta con un potencial hidroeléctrico estimado cercano a 53,000 MW, de los cuales 27,000 MW se han catalogado como económicamente viables. Por su parte, CFE ha estudiado alrededor de 100 cuencas adecuadas para explotación hidroeléctricas [6].

En 2011 mediante un estudio independiente, el Centro Mario Molina estimó el potencial hidroeléctrico para centrales con capacidad instalada menor a 30 MW en las cuencas del Pacífico, Veracruz, Puebla, Oaxaca y Chiapas [7]. Los datos pueden apreciarse en la Figura 2.



Figura 2. Distribución de sitios potenciales para centrales mini hidroeléctricas [7].

De las 26 cuencas estudiadas, se identificaron 489 sitios con un alto potencial para producir más de 23 mil GWh [7] como se puede apreciar en la Tabla 5.

Tabla 5. Potencial Hidroeléctrico Nacional con pequeñas Centrales [7].

Zona	Número de Proyectos	Potencia instalable (MW)	Generación (GWh)
Soto la Marina	1	4	35
Pánuco	24	48	419
Costa de Veracruz	49	89	780
Papaloapan	42	141	1,233
Coatzacoalcos	11	125	1,100
Tonalá	4	6	56
Tacotalpa	16	77	662
Grijalva	38	257	2,253
Usumacinta	38	1,157	10,134
Costa de Chiapas	4	4	37
Costa de Oaxaca	27	107	941
Costa de Michoacán	8	21	182
Costa de Guerrero	32	96	844
Balsas	45	200	1,752
Costa de Michoacán	8	21	182
Lerma	8	19	167
Costa de Jalisco	27	49	431
Santiago	17	59	513
San Pedro	6	19	163
Acaponeta	6	15	136
Baluartel Elota	16	27	235
San Lorenzo	21	37	328
Río Fuerte	19	29	251
Yaquí	15	77	674
Conchos	4	10	92
Nazas	3	6	48
Total	489	2,700	23,645

Capítulo II. Pruebas de Modelo a Turbomáquinas Hidráulicas.

La energía hidroeléctrica todavía enfrenta desafíos científicos y técnicos complejos para asegurar la disponibilidad y la confiabilidad de las centrales eléctricas, a pesar de contar con más de un siglo de desarrollo. El mayor problema está relacionado con la dinámica del rotor que involucra varios campos: hidráulica, ingeniería de potencia y mecánica. Por otro lado, el gran y creciente mercado mundial de energía hidroeléctrica representa una oportunidad para empresas técnicamente avanzadas que ofrezcan una mayor eficiencia en sus equipos. Sin embargo, la dificultad para escalar adecuadamente cualquier mejora en los diseños desde los modelos a escala hace que los experimentos a gran escala sean una necesidad.

Normatividad Aplicable.

A nivel nacional se cuenta con normas que establecen características técnicas de diseño y control de calidad que deben cubrir las turbinas hidráulicas tipo Francis, ya sea de eje vertical (CFE W8300-11) [8] u horizontal (CFE W8300-13) [9], que se adquieren por parte de la Comisión Federal de Electricidad (CFE). En estas normas se indica que al proveedor le corresponde, entre otras actividades, diseñar, fabricar y realizar las pruebas en fábrica y en sitio, así como los informes de las pruebas realizadas, incluyendo las pruebas de modelo.

De acuerdo a [8], como parte de las pruebas de modelo se debe construir y probar un modelo homólogo al prototipo con el fin de verificar el diseño hidráulico y las características de operación del prototipo, de tal manera que los resultados de las pruebas de modelo se utilicen para verificar el diseño del rodete y se consideren en la fabricación del prototipo. El modelo de la turbina debe ser completo, con carcasa espiral extensión de la tubería a presión, ante distribuidor, paletas reguladoras, rodete, tubo de aspiración, pilas y ranuras para compuertas a la salida del tubo. Se deben instalar tomas piezométricas en la entrada y salida del modelo, para obtener lecturas de presión y otros datos. Una sección del tubo de aspiración directamente abajo del rodete debe ser transparente para permitir la visualización del comportamiento del flujo.

Las pruebas deben realizarse en un laboratorio independiente del proveedor y aprobado por un organismo certificador, con el objeto de garantizar la imparcialidad de las mismas y son las siguientes [8]:

- Eficiencia.
- Cavitación.
- Velocidad de desboque e identificación de posición de marcha en vacío.
- Pulsaciones de presión en el tubo de aspiración.
- Variación del par motriz en la flecha.
- Empuje axial.
- Par hidráulico actuante sobre las paletas directrices.

- Calibración de tomas "Winter – Kennedy".

El banco de pruebas, el modelo, la calibración de los instrumentos, la verificación dimensional, la realización de los ensayos, la transposición de valores del modelo al prototipo y la presentación del reporte, deben ser de acuerdo a las normas nacionales [8] e internacionales (IEC 60193) [10]. Sin embargo, se debe aclarar que la norma IEC-60193 aplica a modelos de máquinas prototipo con potencia unitaria superior a 5 MW y la aplicación completa de los procedimientos prescritos generalmente no está justificada para máquinas de menor potencia y tamaño. No obstante, la norma podrá utilizarse para dichas máquinas mediante común acuerdo [10].

Adicional a lo anterior CFE indica que se debe efectuar la simulación de flujo por medio de Computacional Fluid Dynamics (CFD) considerando el trayecto del agua desde la obra de toma hasta el desfogue [8] [9].

Laboratorios y Bancos de Pruebas.

Se debe mencionar que son pocos los laboratorios que existen para la realización de experimentos, entre ellos está el Centro Hidroeléctrico Porjus en Suecia, compuesto por una turbina Francis y una Kaplan de 10 MW cada una dedicada exclusivamente a la educación, la investigación y el desarrollo. Este centro es una fundación de empresas privadas, en el cual se han llevado a cabo varios experimentos desde su inauguración en 2006 por parte de miembros de la fundación y universidades suecas como la LTU (*Luleå Tekniska Universitet*) [11].



Figura 3. Instalaciones Centro Hidroeléctrico Porjus en Suecia [8].

Un laboratorio especializado para pruebas de modelos a escala es el que existe en la NTNU (*Norges Teknisk-Naturvitenskapelige Universitet*) en Noruega. Este laboratorio llamado NVKS (*Norsk VannKraftSenter*) participa activamente en la investigación y desarrollo de turbinas hidráulicas desde 1917. Consiste en una infraestructura flexible, que ofrece la posibilidad de operar plataformas de prueba en configuraciones de ciclo abierto y cerrado. Hay dos plataformas de prueba de turbinas, turbina/bomba, turbina Francis y turbina Pelton, así como varios sistemas de tuberías para investigaciones de golpe de ariete, vibraciones, etc. El sistema de tuberías cerrado puede presurizarse hasta un máximo de 100 m y el sistema abierto tiene una carga máxima de 16 m. La potencia de bombeo disponible es de 700 kW y el caudal máximo es de 1.1 m³/s [12].



Figura 4. Equipo de prueba turbina / bomba-turbina Francis. Vista completa de la carcasa en espiral, el cono del tubo de aspiración, el codo y la descarga de la turbina conectada al tanque aguas abajo. Las mediciones de presión y velocidad (LDV y PIV) pueden realizarse en la sección transparente del cono del tubo de aspiración [12].

Turbomaquinaria Hidráulica y CFD.

En la actualidad, la producción de las turbinas hidráulicas implica la optimización del diseño por simulación numérica y la verificación de rendimiento de la turbina mediante pruebas físicas. Dado que las pruebas requieren la fabricación del modelo de turbina, esto forma una gran parte del costo. Si la verificación del desempeño por pruebas convencionales se sustituye por simulación numérica se podría lograr una reducción significativa de los costos. Sin embargo, el modelado numérico de turbinas hidráulicas es aún un desafío, ya que, el modelado específico para investigar una condición de operación no siempre funciona para otras, además de que la simulación de una turbina completa requiere recursos computacionales sustanciales.

A partir de los resultados en [13] se observó que globalmente la eficiencia se captura bien en las simulaciones, así como la presión media y los perfiles de velocidad. Sin embargo, este resultado es engañoso ya que el torque y la carga pueden presentar una fuerte desviación de los resultados experimentales. Esta dificultad es el resultado de una mala estimación de las

pérdidas en la turbina se atribuye al uso extensivo de funciones de pared para resolver el flujo de las paredes, ya que asumen el equilibrio entre la producción y la disipación de la turbulencia. Por otro lado, el estudio de la capa límite en las turbinas hidráulicas no está completamente desarrollado, debido a que la geometría cambia continuamente y a los grandes gradientes de presión existentes [14]. En la Figura 5 se puede apreciar la exactitud que tienen diversos modelos de turbulencia respecto a los resultados experimentales.

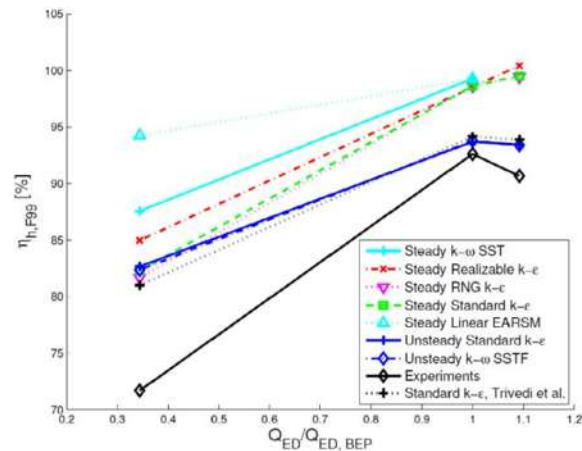


Figura 5. Comparación de los valores experimentales y numéricos de la eficiencia hidráulica en la carga parcial, BEP, y las condiciones de funcionamiento de alta carga [13].

Esta situación es algo común en la simulación de turbinas hidráulicas mediante CFD. Es por ello, que existe una necesidad de desarrollar funciones de pared que permitan la estimación de pérdidas viscosas para simulaciones más precisas, ya que simulaciones y resultados mejorados permiten una estimación fiable de las cargas en los alabes del rotor. En [14], tomando como base el caso de estudio Francis 99, se realizaron investigaciones numéricas sobre el flujo de las fugas a través de los sellos de laberinto y se determinó la eficiencia volumétrica y las pérdidas. Se determinó que las pérdidas por fugas en el sello formuladas mediante técnicas analíticas fueron suficientes para llegar a resultados aproximados al fenómeno real. En la Figura 6 se puede observar la variación en la estimación de la capa límite dependiendo del modelo de turbulencia empleado en la simulación.

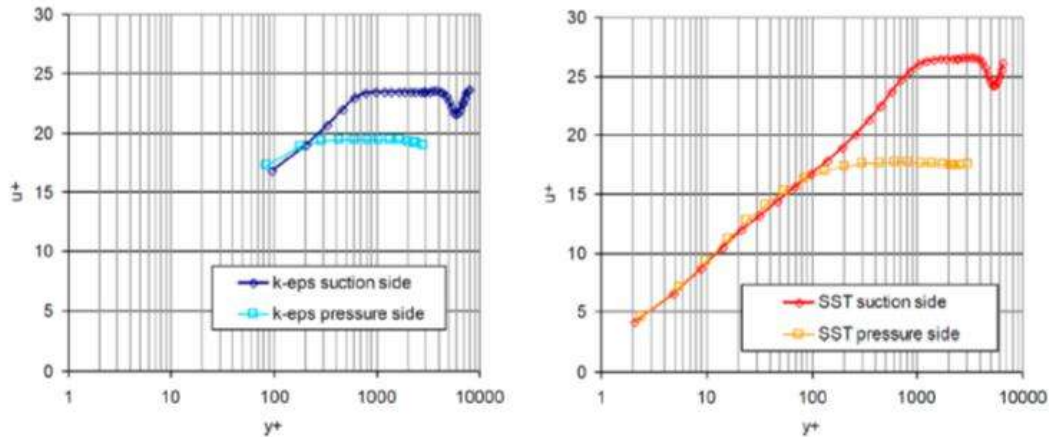


Figura 6. Estimación de la capa limite en los lados de succión y presión de los alabes guía con diferentes valores de y^+ y modelos de turbulencia [14].

El caso de estudio Francis 99 ha servido no solo para desarrollar el estado del arte, sino para aplicaciones más específicas, por ejemplo en [15] se investigaron las consecuencias de la condición de sobre velocidad, que es un fenómeno en el cual turbinas hidráulicas presentan rechazo de carga total durante el funcionamiento debido a las altas fluctuaciones en los parámetros de la red de distribución y en consecuencia, el rodete de la turbina acelera a alta velocidad en segundos. Las mediciones y simulaciones se realizaron en tres puntos de operación. Las simulaciones numéricas se realizaron utilizando el modelo de turbulencia k -estándar, k - ω *shear stress transport* (SST) y *scale-adaptive simulation* (SAS). Un total de 12.8 millones de elementos de malla hexaédrica fueron creados en la turbina completa, desde la entrada de la carcasa en espiral hasta la salida del tubo de aspiración. El análisis experimental y numérico mostró que el rotor fue sometido a una carga de presión inestable hasta tres veces la carga de presión observada en el mejor punto de eficiencia.

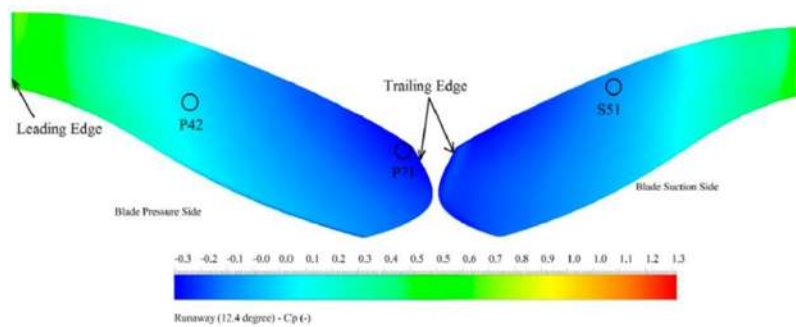


Figura 7. Contornos del coeficiente de presión sobre los lados de succión y presión de los alabes operando a sobre velocidad ($\alpha = 12.4^\circ$) [15].

La simulación numérica se ha convertido en una herramienta fundamental para el análisis de turbinas hidráulicas debido a la precisión en los resultados que se ha alcanzado en los últimos

años. En [16] se presenta un método de predicción del rendimiento de una turbina Francis completa basado en el análisis de flujo mediante un banco de pruebas virtual. El rendimiento se predice resolviendo las ecuaciones promedio de Navier-Stokes (RANS) en estado transitorio combinadas con el modelo de esfuerzos de Reynolds (*Reynolds-Stress*). La precisión de la predicción con CFD se evaluó comparando los resultados de las pruebas con pruebas de modelo para tres turbinas con velocidades específicas diferentes. Al comparar los resultados de la simulación con los resultados de las pruebas del modelo para la eficiencia, fluctuación de presión y cavitación, se demostró que el modelo numérico presentado podría predecir características importantes de las turbinas Francis con alta precisión, no sólo cuantitativa sino también cualitativamente. Los resultados obtenidos en cuanto a eficiencia demuestran que las pérdidas hidráulicas se sobrestiman en el cálculo en estado estable, por lo tanto, es necesario predecir la eficiencia utilizando estado transitorio. Sin embargo, la simulación en estado transitorio necesita un enorme tiempo computacional para calcular muchos de los puntos de operación deseados para el desarrollo del rendimiento hidráulico. Actualmente, con el aumento de la velocidad de procesamiento, las pruebas numéricas son una herramienta más precisa para la estimación del desempeño hidráulico turbinas Francis.

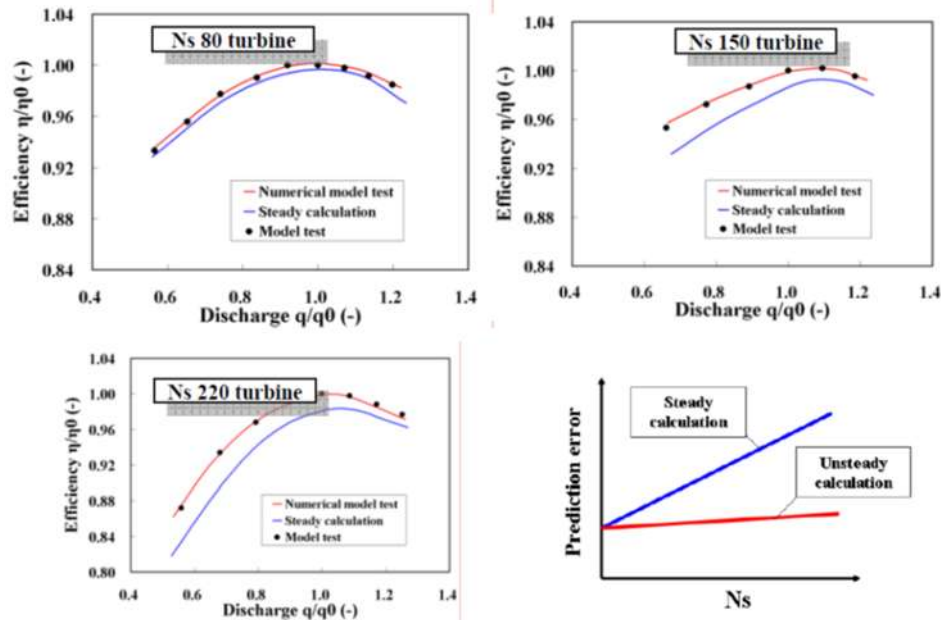


Figura 8. Comparación de la eficiencia de turbina entre la predicción numérica y los resultados experimentales [16].

De manera similar, en [17] se utiliza totalmente la simulación en CFD para desarrollar un proyecto de re-potenciación, donde diseño inverso y simulaciones de flujo totalmente tridimensionales se realizaron para fabricar un rotor de turbina Francis nuevo. En el trabajo, los análisis de flujo se llevaron a cabo con cuatro posiciones de apertura de los alabes del distribuidor manteniendo la velocidad de rotación constante. Para obtener características de

flujo, se calcularon los valores medios de parámetros como velocidades y ángulos de flujo en la entrada y salida del distribuidor, los alabes guía y los alabes de la turbina. El objetivo fue analizar el comportamiento del flujo y la distribución de la presión para afinar aún más el experimento numérico completo y lograr el nivel de precisión necesario para el diseño conceptual de la nueva turbina.

La combinación exitosa de la optimización del diseño basada en CFD con las pruebas físicas resultó en un nuevo modelo que proporciona una mejora de aproximadamente un 9,93% en la eficiencia máxima. Además, el modelo proporciona una característica de cavitación perfectamente mejorada con un rendimiento extremadamente estable en una gama mucho más amplia de condiciones de operación en comparación con el diseño existente. Ya que el sistema de análisis en CFD ayuda en la reducción del costo de pruebas físicas y ahorro en tiempo, este puede ser útil para mejorar las técnicas existentes de medición de eficiencia y evaluar el rendimiento de turbinas hidráulicas [17].

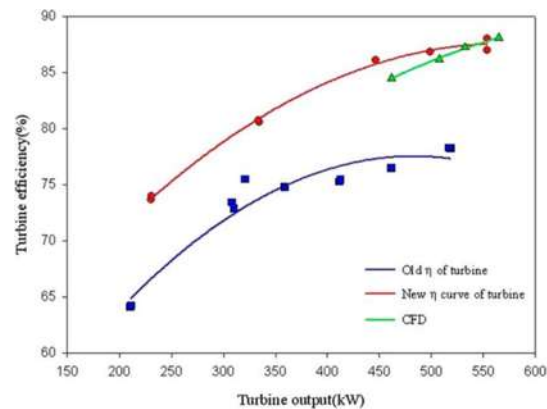


Figura 9. Comparación entre la eficiencia calculada y la eficiencia medida [17].

Ya que la simulación mediante CFD requiere de un gran poder de cómputo, la llegada de los GPUs (*Graphical Processor Units*) ha permitido resolver problemas numéricos de más intensidad. En [18] se presentan los cálculos de flujo tridimensional estable en una bomba centrífuga, lo cual es una muestra de las tendencias que se pueden implementar, a fin de reducir costos de inversión. El dominio computacional considerado fue una de las tres etapas de la bomba, ya que cada una tiene exactamente el mismo diseño. Las simulaciones se realizaron en BEP (*Best Efficiency Point*) con cuatro diferentes modelos de turbulencia. Los resultados se compararon con el informe de rendimiento proporcionado por el fabricante. La comparación mostró que las simulaciones en estado estacionario presentan una buena aproximación a los datos LDV (*Laser Doppler Velocimetry*). Los resultados demuestran también que RapidCFD, basado en OpenFOAM, presenta un incremento de velocidad de procesamiento (alrededor de 2.6-3.5) con las tarjetas gráficas GeForce Titan X de NVIDIA frente a las unidades CPU Intel Xeon E5-2670 para cálculos en doble precisión. Sin embargo,

la memoria de las tarjetas GPU es limitada y puede ser un cuello de botella para simular casos más grandes, por lo que la distribución de la carga de trabajo a varios nodos de cálculo se vuelve un punto de vital importancia.

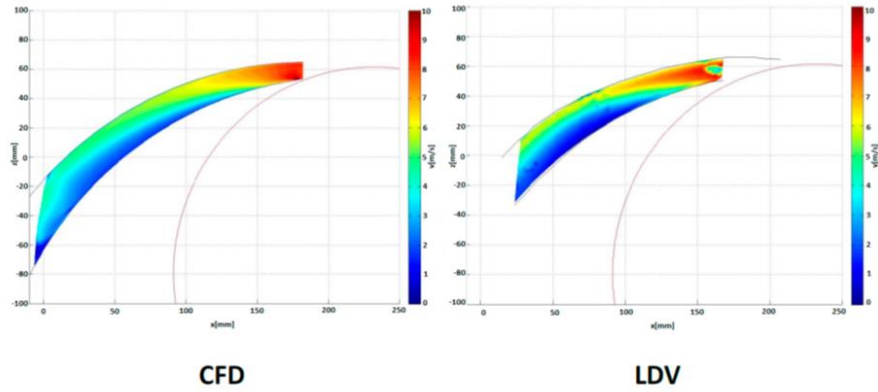


Figura 10. Magnitudes de velocidad media en la sección media del difusor. CFD (Izquierda), LDV (Derecha) para el BEP ($60 \text{ m}^3/\text{h}$) con el modelo k-e Realizable [18].

Capítulo III. Dinámica de Fluidos Computacional.

Según [19] la dinámica de fluidos computacionales o CFD (*Computational Fluid Dynamics*) se enfoca en el análisis de sistemas que involucran flujo de fluidos, transferencia de calor y fenómenos asociados mediante simulación numérica. Hasta hace pocos años el CFD se desarrolló más lentamente respecto a otras herramientas empleadas en los paquetes CAE (Ingeniería Asistida por Computadora), como análisis por elementos finitos (FEA), debido a que su complejidad limita una solución del flujo de fluidos que sea tanto económica como completa y exacta.

Así mismo, la inversión para realizar simulaciones mediante CFD suficientemente exactas tiende a ser elevada (*hardware y software*), sin embargo no es tan grande como la inversión requerida para un laboratorio bien instrumentado. Además, el CFD ofrece algunas ventajas sobre la experimentación física [19]:

- reducción de los plazos de entrega y los costos de nuevos modelos
- capacidad para analizar sistemas en los que los experimentos controlados son complicados
- capacidad para analizar sistemas en condiciones peligrosas

Los códigos comerciales de CFD y algunos de código abierto incluyen interfaces de usuario (GUI) para ingresar condiciones iniciales y de frontera de los problemas y examinar los resultados. Sin embargo, algunos códigos de acceso abierto no cuentan con una GUI, sino que la comunicación en varias etapas es mediante línea de comandos [19]. Aun así, todos los códigos contienen tres elementos principales:

- un pre procesador,
- un solucionador y
- un pos procesador.

El pre procesamiento abarca lo siguiente:

- Definición de la geometría del dominio computacional
- Generación de la malla
- Definición de las propiedades de los fluidos
- Especificación de las condiciones iniciales y de frontera

La etapa de solución incluye siguientes actividades:

- Integración de las ecuaciones gobernantes del flujo en el dominio
- Discretización de las ecuaciones integrales resultantes en un sistema de ecuaciones algebraicas
- Solución de las ecuaciones algebraicas por métodos iterativos

Finalmente, en el pos procesamiento se pueden realizar las siguientes tareas:

- Visualización de malla y geometría de dominio
- Gráficos vectoriales
- Gráficos de líneas y contornos sombreados
- Gráficos de superficie 2D y 3D
- Seguimiento de partículas

Ecuaciones Gobernantes.

Las ecuaciones que gobiernan el flujo de fluidos representan las leyes de conservación de la física [20]:

- Conservación de la materia
- Conservación de cantidad de movimiento lineal y angular
- Conservación de la energía

La ecuación de continuidad describe la velocidad de variación respecto al tiempo de la densidad del fluido en un punto del espacio, se puede escribir en forma diferencial como:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x} \rho v_x + \frac{\partial}{\partial y} \rho v_y + \frac{\partial}{\partial z} \rho v_z = 0$$

En notación vectorial puede abreviarse como:

$$\frac{\partial \rho}{\partial t} + (\nabla \cdot \rho \mathbf{v}) = 0$$

Como se detalla en [20], el primer término de la ecuación anterior es la tasa de incremento de masa por unidad de volumen, mientras que el segundo término es la tasa de adición de masa por convección por unidad de volumen. El término $\nabla \cdot \rho \mathbf{v}$ se conoce como "divergencia de $\rho \mathbf{v}$ ", y algunas veces se escribe como "div $\rho \mathbf{v}$ ". El vector $\rho \mathbf{v}$ es la densidad de flujo de materia y su divergencia es la velocidad neta con que sale o se emite la densidad de flujo de materia por unidad de volumen.

La ecuación de movimiento, la cual parte de la formulación diferencial de la ley de conservación de cantidad de movimiento, incluye las fuerzas superficiales y de cuerpo. En forma diferencial se puede escribir mediante componentes de la siguiente manera:

$$\frac{\partial}{\partial t} \rho v_x = - \left(\frac{\partial}{\partial x} \rho v_x v_x + \frac{\partial}{\partial y} \rho v_y v_x + \frac{\partial}{\partial z} \rho v_z v_x + \frac{\partial p}{\partial x} + \frac{\partial}{\partial x} \tau_{xx} + \frac{\partial}{\partial y} \tau_{yx} + \frac{\partial}{\partial z} \tau_{zx} \right) + \rho g_x$$

$$\frac{\partial}{\partial t} \rho v_y = - \left(\frac{\partial}{\partial x} \rho v_x v_y + \frac{\partial}{\partial y} \rho v_y v_y + \frac{\partial}{\partial z} \rho v_z v_y + \frac{\partial p}{\partial y} + \frac{\partial}{\partial x} \tau_{xy} + \frac{\partial}{\partial y} \tau_{yy} + \frac{\partial}{\partial z} \tau_{zy} \right) + \rho g_y$$

$$\frac{\partial}{\partial t} \rho v_z = - \left(\frac{\partial}{\partial x} \rho v_x v_z + \frac{\partial}{\partial y} \rho v_y v_z + \frac{\partial}{\partial z} \rho v_z v_z + \frac{\partial p}{\partial z} + \frac{\partial}{\partial x} \tau_{xz} + \frac{\partial}{\partial y} \tau_{yz} + \frac{\partial}{\partial z} \tau_{zz} \right) + \rho g_z$$

La forma general conservativa de la ecuación de movimiento puede escribirse usando notación vectorial como:

$$\frac{\partial}{\partial t} \rho \mathbf{v} = -[\nabla \cdot \rho \mathbf{v} \mathbf{v}] - \nabla p - [\nabla \cdot \boldsymbol{\tau}] + \rho \mathbf{g}$$

Donde, como se detalla en [20], el término del lado izquierdo representa la tasa de incremento de cantidad de movimiento por unidad de volumen, el primer término del lado derecho es la tasa de adición de cantidad de movimiento por convección por unidad de volumen, el segundo y tercer término representan la tasa de adición de cantidad de movimiento por transporte molecular por unidad de volumen y el ultimo termino agrupa la fuerzas externas sobre el fluido por unidad de volumen. El termino, ∇p es un vector denominado "gradiente del (escalar) p ", y algunas veces se escribe como "grad p ". El símbolo $[\nabla \cdot \boldsymbol{\tau}]$ es un vector denominado "divergencia del (tensor) $\boldsymbol{\tau}$ ", y $[\nabla \cdot \rho \mathbf{v} \mathbf{v}]$ es un vector denominado "divergencia del (producto diádico) $\nabla \cdot \rho \mathbf{v} \mathbf{v}$ ".

Para fluidos que siguen la ley de Newton de la viscosidad, el tensor de esfuerzo $\boldsymbol{\tau}$ es una función lineal de la tasa de deformación que se puede escribir en forma expandida como:

$$\boldsymbol{\tau} = \begin{bmatrix} -2\mu \frac{\partial v_x}{\partial x} + \lambda(\nabla \cdot \mathbf{v}) & -\mu \left(\frac{\partial v_y}{\partial x} + \frac{\partial v_x}{\partial y} \right) & -\mu \left(\frac{\partial v_x}{\partial z} + \frac{\partial v_z}{\partial x} \right) \\ -\mu \left(\frac{\partial v_y}{\partial x} + \frac{\partial v_x}{\partial y} \right) & -2\mu \frac{\partial v_y}{\partial y} + \lambda(\nabla \cdot \mathbf{v}) & -\mu \left(\frac{\partial v_y}{\partial z} + \frac{\partial v_z}{\partial y} \right) \\ -\mu \left(\frac{\partial v_x}{\partial z} + \frac{\partial v_z}{\partial x} \right) & -\mu \left(\frac{\partial v_y}{\partial z} + \frac{\partial v_z}{\partial y} \right) & -2\mu \frac{\partial v_z}{\partial z} + \lambda(\nabla \cdot \mathbf{v}) \end{bmatrix}$$

Donde:

$$\lambda(\nabla \cdot \mathbf{v}) = \frac{2}{3} \mu \left(\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z} \right)$$

Si se emplea notación tensorial, el tensor de esfuerzo $\boldsymbol{\tau}$ se puede expresar como:

$$\boldsymbol{\tau} = -\mu(\nabla \mathbf{v} + (\nabla \mathbf{v})^T) + \lambda(\nabla \cdot \mathbf{v})\boldsymbol{\delta}$$

Por lo tanto, su divergencia es un vector que se puede escribir como:

$$[\nabla \cdot \boldsymbol{\tau}] = \begin{bmatrix} \frac{\partial}{\partial x} \left[-2\mu \frac{\partial v_x}{\partial x} + \lambda(\nabla \cdot \mathbf{v}) \right] + \frac{\partial}{\partial y} \left[-\mu \left(\frac{\partial v_y}{\partial x} + \frac{\partial v_x}{\partial y} \right) \right] + \frac{\partial}{\partial z} \left[-\mu \left(\frac{\partial v_x}{\partial z} + \frac{\partial v_z}{\partial x} \right) \right] \\ \frac{\partial}{\partial x} \left[-\mu \left(\frac{\partial v_y}{\partial x} + \frac{\partial v_x}{\partial y} \right) \right] + \frac{\partial}{\partial y} \left[-2\mu \frac{\partial v_y}{\partial y} + \lambda(\nabla \cdot \mathbf{v}) \right] + \frac{\partial}{\partial z} \left[-\mu \left(\frac{\partial v_y}{\partial z} + \frac{\partial v_z}{\partial y} \right) \right] \\ \frac{\partial}{\partial x} \left[-\mu \left(\frac{\partial v_x}{\partial z} + \frac{\partial v_z}{\partial x} \right) \right] + \frac{\partial}{\partial y} \left[-\mu \left(\frac{\partial v_z}{\partial y} + \frac{\partial v_y}{\partial z} \right) \right] + \frac{\partial}{\partial z} \left[-2\mu \frac{\partial v_z}{\partial z} + \lambda(\nabla \cdot \mathbf{v}) \right] \end{bmatrix}$$

Que al utilizar notación tensorial se puede escribir como:

$$[\nabla \cdot \boldsymbol{\tau}] = \nabla \cdot [-\mu(\nabla \mathbf{v} + (\nabla \mathbf{v})^T)] + \nabla(\lambda(\nabla \cdot \mathbf{v}))$$

Así, al sustituir en la forma conservativa de la ecuación de movimiento se puede obtener la siguiente ecuación:

$$\frac{\partial}{\partial t} \rho \mathbf{v} = -[\nabla \cdot \rho \mathbf{v} \mathbf{v}] - \nabla p + \nabla \cdot [\mu(\nabla \mathbf{v} + (\nabla \mathbf{v})^T)] - \nabla(\lambda(\nabla \cdot \mathbf{v})) + \rho \mathbf{g}$$

Cuando se desarrolla el tercer término de la ecuación se llega a:

$$\frac{\partial}{\partial t} \rho \mathbf{v} = -[\nabla \cdot \rho \mathbf{v} \mathbf{v}] - \nabla p + \nabla \cdot [\mu \nabla \mathbf{v}] + \nabla \cdot [\mu(\nabla \mathbf{v})^T] - \nabla(\lambda(\nabla \cdot \mathbf{v})) + \rho \mathbf{g}$$

Al agrupar en un único “termino fuente”, se llega a la forma conservativa de la ecuación de movimiento para fluidos newtonianos:

$$\frac{\partial}{\partial t} \rho \mathbf{v} = -[\nabla \cdot \rho \mathbf{v} \mathbf{v}] - \nabla p + \nabla \cdot [\mu \nabla \mathbf{v}] + \mathbf{S}_v$$

La ecuación de energía, la cual es una derivación de la primera ley de la termodinámica, establece que la tasa de cambio de energía de una partícula de fluido es igual a la tasa de adición de calor a la partícula de fluido más la tasa de trabajo realizado sobre la partícula [20]. Considerando los flujos de calor, la ecuación toma la siguiente forma:

$$\frac{\partial}{\partial t} \rho c_p T + \nabla \cdot \rho c_p T \mathbf{v} = -\nabla \cdot \mathbf{q} - \left(\frac{\partial \ln \rho}{\partial \ln T} \right)_p \frac{Dp}{Dt} + [\boldsymbol{\tau} : \nabla \mathbf{v}] + q_{gen}$$

Donde:

$$\nabla \cdot \mathbf{q} = \frac{\partial}{\partial x} q_x + \frac{\partial}{\partial y} q_y + \frac{\partial}{\partial z} q_z$$

Mediante el uso de la ley de Fourier se puede obtener:

$$\frac{\partial}{\partial t} \rho c_p T + \nabla \cdot \rho c_p T \mathbf{v} = \nabla \cdot k \nabla T - \left(\frac{\partial \ln \rho}{\partial \ln T} \right)_p \frac{Dp}{Dt} + [\boldsymbol{\tau} : \nabla \mathbf{v}] + q_{gen}$$

En la ecuación anterior, los divergentes de $\rho c_p T \mathbf{v}$ y $k \nabla T$ están dados por:

$$\begin{aligned} \nabla \cdot \rho c_p T \mathbf{v} &= \frac{\partial}{\partial x} \rho c_p T v_x + \frac{\partial}{\partial y} \rho c_p T v_y + \frac{\partial}{\partial z} \rho c_p T v_z \\ \nabla \cdot k \nabla T &= \frac{\partial}{\partial x} \left(k \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left(k \frac{\partial T}{\partial y} \right) + \frac{\partial}{\partial z} \left(k \frac{\partial T}{\partial z} \right) \end{aligned}$$

Mientras que la derivada material de p se puede expresar como:

$$\frac{Dp}{Dt} = \frac{\partial p}{\partial t} + (\mathbf{v} \cdot \nabla p) = \frac{\partial p}{\partial t} + v_x \frac{\partial p}{\partial x} + v_y \frac{\partial p}{\partial y} + v_z \frac{\partial p}{\partial z}$$

Además, la tasa irreversible de aumento de energía interna por unidad de volumen por disipación viscosa se puede definir como:

$$\begin{aligned} [\boldsymbol{\tau} : \nabla \mathbf{v}] &= -\frac{2}{3} \mu \left(\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z} \right)^2 \\ &+ \mu \left[2 \left(\frac{\partial v_x}{\partial x} \right)^2 + 2 \left(\frac{\partial v_y}{\partial y} \right)^2 + 2 \left(\frac{\partial v_z}{\partial z} \right)^2 + \left(\frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x} \right)^2 + \left(\frac{\partial v_x}{\partial z} + \frac{\partial v_z}{\partial x} \right)^2 + \left(\frac{\partial v_y}{\partial z} + \frac{\partial v_z}{\partial y} \right)^2 \right] \end{aligned}$$

Finalmente, se puede agregar una ecuación de estado a la ecuación de la energía como:

$$\frac{\partial}{\partial t} \rho c_p T + \nabla \cdot \rho c_p T \mathbf{v} = \nabla \cdot k \nabla T + \rho T \frac{Dc_p}{Dt} - \left(\frac{\partial \ln \rho}{\partial \ln T} \right)_p \frac{Dp}{Dt} + [\boldsymbol{\tau} : \nabla \mathbf{v}] + q_{gen}$$

Donde:

$$\frac{Dc_p}{Dt} = \frac{\partial c_p}{\partial t} + (\mathbf{v} \cdot \nabla c_p) = \frac{\partial c_p}{\partial t} + v_x \frac{\partial c_p}{\partial x} + v_y \frac{\partial c_p}{\partial y} + v_z \frac{\partial c_p}{\partial z}$$

Los últimos cuatro términos de la ecuación de la energía pueden agruparse en un nuevo “termino fuente”, dando lugar a la siguiente ecuación:

$$\frac{\partial}{\partial t} \rho c_p T = -\nabla \cdot \rho c_p T \mathbf{v} + \nabla \cdot k \nabla T + \mathbf{S}_T$$

Ahora bien, si se introduce una variable general ϕ , la forma conservativa de las ecuaciones de conservación anterior, se puede generalizar y escribir de la siguiente forma:

$$\frac{\partial}{\partial t}(\rho\phi) + \nabla \cdot (\rho\phi\mathbf{v}) = \nabla \cdot (\Gamma\nabla\phi) + S_\phi$$

Esta ecuación se conoce como “ecuación de transporte para la propiedad ϕ ”. Atendiendo la descripción dada por [19], en la ecuación se pueden observar los diversos procesos de transporte presentes: el término de tasa de cambio y el término convectivo del lado izquierdo y el término difusivo ($\Gamma =$ coeficiente de difusión) y el término fuente respectivamente del lado derecho. Esta ecuación se utiliza como punto de partida para los procedimientos de cálculo en el método de volumen finito. La clave del método de volumen finito para el CFD es la integración de la ecuación de transporte sobre un volumen de control (CV):

$$\int_{CV} \frac{\partial}{\partial t}(\rho\phi)dV + \int_{CV} \nabla \cdot (\rho\phi\mathbf{v})dV = \int_{CV} \nabla \cdot (\Gamma\nabla\phi)dV + \int_{CV} S_\phi dV$$

Las integrales de volumen en el segundo término del lado izquierdo, el término convectivo, y en el primer término del lado derecho, el término difusivo, se pueden reescribir como integrales de superficie usando el teorema de divergencia de Gauss [19] de la siguiente manera:

$$\frac{\partial}{\partial t} \left(\int_{CV} \rho\phi dV \right) + \int_A \mathbf{n} \cdot (\rho\phi\mathbf{v}) dA = \int_A \mathbf{n} \cdot (\Gamma\nabla\phi) dA + \int_{CV} S_\phi dV$$

Siguiendo el procedimiento dado por [19], el primer término del lado izquierdo determina la tasa de cambio de la cantidad total de la propiedad del fluido ϕ en el volumen de control. El segundo término del lado izquierdo, donde el producto $\mathbf{n} \cdot (\rho\phi\mathbf{v})$ expresa el componente de la propiedad ϕ debido al flujo del fluido a lo largo del vector normal hacia afuera \mathbf{n} , es el término convectivo, que determina la tasa neta de disminución de la propiedad ϕ del elemento de fluido debido a la convección. El primer término del lado derecho, el término difusivo, representa la tasa neta de aumento de la propiedad ϕ del elemento fluido debido a la difusión. El término final proporciona la tasa de aumento de la propiedad ϕ como resultado de las fuentes dentro del elemento fluido.

En estado estacionario, el término de tasa de cambio es igual a cero. Lo cual conduce a la forma integral de la ecuación de transporte estacionaria:

$$\int_A \mathbf{n} \cdot (\rho\phi\mathbf{v}) dA = \int_A \mathbf{n} \cdot (\Gamma\nabla\phi) dA + \int_{CV} S_\phi dV$$

En estado transitorio es necesario integrar con respecto al tiempo t en un pequeño intervalo Δt , desde t hasta $t + \Delta t$. Esto produce la forma integral más general de la ecuación de transporte:

$$\int_{\Delta t} \frac{\partial}{\partial t} \left(\int_{CV} \rho \phi dV \right) dt + \int_{\Delta t} \int_A \mathbf{n} \cdot (\rho \phi \mathbf{v}) dA dt = \int_{\Delta t} \int_A \mathbf{n} \cdot (\Gamma \nabla \phi) dA dt + \int_{\Delta t} \int_{CV} S_\phi dV dt$$

Turbulencia.

En la naturaleza y en aplicaciones de ingeniería la mayoría de los flujos son turbulentos y dar una definición precisa de turbulencia muy difícil. Sin embargo, lo que se puede hacer es proporcionar las características de los flujos turbulentos como se describe en [21]:

- Una característica es la irregularidad o aleatoriedad de todos los flujos turbulentos, esto hace imposible un enfoque determinista de los problemas de turbulencia; en cambio se emplean métodos estadísticos.
- La difusividad de la turbulencia provoca una mezcla rápida y mayores tasas de transferencia de cantidad de movimiento, calor y masa.
- Los flujos turbulentos siempre ocurren con números de Reynolds altos. La turbulencia se origina como una inestabilidad de los flujos laminares si el número de Reynolds se vuelve demasiado grande.
- La turbulencia es rotacional y tridimensional, y se caracteriza por altos niveles de vorticidad fluctuante. Las fluctuaciones aleatorias de vorticidad no podrían mantenerse si las fluctuaciones de velocidad fueran bidimensionales, ya que el mecanismo de estiramiento de vórtice está ausente en el flujo bidimensional.
- Los flujos turbulentos son siempre disipativos, ya que los esfuerzos cortantes viscosos realizan un trabajo de deformación que aumenta la energía interna del fluido a expensas de la energía cinética de la turbulencia; por lo tanto se necesita un suministro continuo de energía para compensar estas pérdidas viscosas.
- La turbulencia es un fenómeno continuo, regido por las ecuaciones de la mecánica de fluidos. Incluso a escalas muy pequeñas en un flujo turbulento, las perturbaciones son mucho más grandes que cualquier escala de longitud molecular.
- La turbulencia no es una característica de los fluidos sino de los flujos de fluidos, debido que la mayor parte de la dinámica de la turbulencia es la misma en todos los fluidos, ya sean líquidos o gases. Si el número de Reynolds es grande las principales características de los flujos turbulentos no están controladas por las propiedades moleculares del fluido.

Propiedades Promedio.

Si se considera un flujo turbulento dentro de un tubo sometido a un gradiente de presión constante, en un punto cualquiera dentro del seno del fluido se puede considerar el vector

velocidad como una función del tiempo fluctuando de manera caótica, donde las fluctuaciones son desviaciones irregulares respecto a un valor medio [20]. Por lo tanto, la velocidad puede representarse como la suma del valor promedio $\bar{\mathbf{v}}$ y su fluctuación \mathbf{v}' :

$$\mathbf{v} = \bar{\mathbf{v}} + \mathbf{v}'$$

Esta formulación se denomina comúnmente como descomposición de Reynolds. Cuando se obtiene un promedio en el tiempo agrupando un gran número de fluctuaciones a partir de \mathbf{v} se obtiene:

$$\bar{\mathbf{v}} = \frac{1}{t_0} \int_{t-\frac{1}{2}t_0}^{t+\frac{1}{2}t_0} \mathbf{v} dt$$

En la ecuación, el periodo de tiempo t_0 debe ser suficientemente grande para generar una función promedio que capture la mayor cantidad de fluctuaciones. De esta manera, la velocidad promedio $\bar{\mathbf{v}}$ es independiente del tiempo, aunque aún sigue dependiendo de la posición [19]. Cuando la velocidad promedio no depende del tiempo, se considera un flujo turbulento estable.

De acuerdo con [22], en definición de valores promedio y fluctuantes para la velocidad se tienen las relaciones siguientes:

$$\overline{\mathbf{v}'} = 0$$

$$\overline{\bar{\mathbf{v}}} = \bar{\mathbf{v}}$$

$$\overline{\bar{\mathbf{v}}\mathbf{v}'} = 0$$

$$\overline{\nabla \mathbf{v}} = \nabla \bar{\mathbf{v}}$$

Además, según menciona [20], se debe notar que la cantidad $\overline{\mathbf{v}'^2}$ no es cero, y que la relación $\sqrt{\overline{\mathbf{v}'^2}} / \bar{\mathbf{v}}$ se considera una medida de la magnitud de las fluctuaciones turbulentas. Esta cantidad, conocida como intensidad de la turbulencia. Cantidades como $\overline{u'_x u'_y}$ también son poseen valores distintos de cero. Esto es debido a que los movimientos locales en las direcciones x y y están correlacionados. En otras palabras, las fluctuaciones en la dirección x no son independientes de las fluctuaciones en la dirección y .

Una vez definidas las cantidades promedio y establecido las propiedades de las cantidades fluctuantes, se puede desarrollar la ecuación de continuidad promedio para un fluido de densidad ρ y viscosidad μ constantes.

Al sustituir $\mathbf{v} = \bar{\mathbf{v}} + \mathbf{v}'$ en la ecuación de continuidad se tiene:

$$\overline{\frac{\partial \rho}{\partial t} + \nabla \cdot \rho(\bar{\mathbf{v}} + \mathbf{v}')} = 0$$

Después de reducir términos se obtiene la ecuación de continuidad promedio en forma vectorial:

$$\nabla \cdot \rho \bar{\mathbf{v}} = 0$$

O bien, en forma desarrollada se puede expresar como:

$$\frac{\partial}{\partial x} \rho \bar{u}_x + \frac{\partial}{\partial y} \rho \bar{u}_y + \frac{\partial}{\partial z} \rho \bar{u}_z = 0$$

Adicionalmente, si se considera el flujo turbulento en un tubo sometido a un gradiente de presión dependiente del tiempo, es posible definir cantidades promedio. Se debe tener en cuenta nuevamente que el periodo de tiempo t_0 debe ser grande respecto a los periodos de las fluctuaciones del gradiente de presión [20]. Por lo tanto se tiene:

$$p = \bar{p} + p'$$

Al sustituir $\mathbf{v} = \bar{\mathbf{v}} + \mathbf{v}'$ y $p = \bar{p} + p'$ en la ecuación de cantidad de movimiento, esta se transforma en:

$$\overline{\frac{\partial}{\partial t} \rho(\bar{\mathbf{v}} + \mathbf{v}')} = -[\nabla \cdot \rho(\bar{\mathbf{v}} + \mathbf{v}')(\bar{\mathbf{v}} + \mathbf{v}')] - \nabla(\bar{p} + p') \\ + \nabla \cdot \left\{ \mu \left[\nabla(\bar{\mathbf{v}} + \mathbf{v}') + (\nabla(\bar{\mathbf{v}} + \mathbf{v}'))^T \right] \right\} - \nabla \{ \lambda [\nabla \cdot (\bar{\mathbf{v}} + \mathbf{v}')] \} + \rho \mathbf{g}$$

Posterior a reducir términos se obtiene:

$$\frac{\partial}{\partial t} \rho \bar{\mathbf{v}} = -[\nabla \cdot \rho \overline{\mathbf{v}\mathbf{v}'}] - \nabla \bar{p} + [\nabla \cdot (\bar{\boldsymbol{\tau}}^{(v)} - \bar{\boldsymbol{\tau}}^{(t)})] + \rho \mathbf{g}$$

Donde $\bar{\boldsymbol{\tau}}^t$ es el tensor de densidad de flujo de cantidad de movimiento turbulento, cuyas componentes se denominan esfuerzos de Reynolds [22] y tiene la forma siguiente:

$$\bar{\boldsymbol{\tau}}^{(t)} = \rho \begin{bmatrix} \overline{u'_x u'_x} & \overline{u'_x u'_y} & \overline{u'_x u'_z} \\ \overline{u'_x u'_y} & \overline{u'_y u'_y} & \overline{u'_y u'_z} \\ \overline{u'_x u'_z} & \overline{u'_y u'_z} & \overline{u'_z u'_z} \end{bmatrix}$$

Los esfuerzos de Reynolds no están relacionados con los gradientes de velocidad de una forma simple, como es el caso de los esfuerzos viscosos promedio, ya que son funciones de

la posición y la intensidad de la turbulencia [22]. El tensor de densidad de flujo de cantidad de movimiento viscoso promedio $\bar{\boldsymbol{\tau}}^{(v)}$ tiene las mismas componentes que las expresiones para flujo laminar, aunque las componentes de la velocidad promedio aparecen en ellas:

$$\bar{\boldsymbol{\tau}}^{(v)} = \begin{bmatrix} -2\mu \frac{\partial \bar{v}_x}{\partial x} & -\mu \left(\frac{\partial \bar{v}_y}{\partial x} + \frac{\partial \bar{v}_x}{\partial y} \right) & -\mu \left(\frac{\partial \bar{v}_x}{\partial z} + \frac{\partial \bar{v}_z}{\partial x} \right) \\ -\mu \left(\frac{\partial \bar{v}_y}{\partial x} + \frac{\partial \bar{v}_x}{\partial y} \right) & -2\mu \frac{\partial \bar{v}_y}{\partial y} & -\mu \left(\frac{\partial \bar{v}_y}{\partial z} + \frac{\partial \bar{v}_z}{\partial y} \right) \\ -\mu \left(\frac{\partial \bar{v}_x}{\partial z} + \frac{\partial \bar{v}_z}{\partial x} \right) & -\mu \left(\frac{\partial \bar{v}_z}{\partial y} + \frac{\partial \bar{v}_y}{\partial z} \right) & -2\mu \frac{\partial \bar{v}_z}{\partial z} \end{bmatrix}$$

Si el flujo no es isotérmico, es decir, si el fluido se somete a un gradiente de temperatura dependiente del tiempo, también es posible obtener una temperatura promedio \bar{T} y su fluctuación T' . Para un punto arbitrario en el fluido se puede definir entonces que:

$$T = \bar{T} + T'$$

Se observa que $\overline{T'} = 0$, pero cantidades como $\overline{v'_x T'}$, $\overline{v'_y T'}$, $\overline{v'_z T'}$ no son iguales a cero debido a la correlación entre las fluctuaciones de velocidad y temperatura en cualquier punto [20]. Para un fluido con densidad ρ , viscosidad μ , calor específico c_p y conductividad térmica k constantes, al sustituir $\mathbf{v} = \bar{\mathbf{v}} + \mathbf{v}'$ y $T = \bar{T} + T'$ en la ecuación de la energía se obtiene:

$$\overline{\frac{\partial}{\partial t} \rho c_p (\bar{T} + T')} = -\nabla \cdot \rho c_p (\bar{T} + T') (\bar{\mathbf{v}} + \mathbf{v}') + \nabla \cdot k \nabla (\bar{T} + T') + [\boldsymbol{\tau} : \nabla (\bar{\mathbf{v}} + \mathbf{v}')]$$

Cuando se usa la definición de valores promedio y fluctuantes la ecuación se escribe como:

$$\frac{\partial}{\partial t} \rho c_p \bar{T} = -\nabla \cdot \rho c_p \bar{T} \bar{\mathbf{v}} + \nabla \cdot (\bar{\mathbf{q}}^{(v)} + \bar{\mathbf{q}}^{(t)}) + \mu (\bar{\Phi}_v^{(v)} + \bar{\Phi}_v^{(t)})$$

Donde, el vector de flujo de calor turbulento $\bar{\mathbf{q}}^{(t)}$ se define como:

$$\bar{\mathbf{q}}^{(t)} = \rho c_p \begin{bmatrix} \overline{v'_x T'} \\ \overline{v'_y T'} \\ \overline{v'_z T'} \end{bmatrix} = \rho c_p \overline{\mathbf{v}' T'}$$

y la función de disipación de energía turbulenta $\bar{\Phi}_v^{(t)}$ puede expresarse de la siguiente forma:

$$\bar{\Phi}_v^{(t)} = 2 \overline{\left(\frac{\partial v'_x}{\partial x} \right) \left(\frac{\partial v'_x}{\partial x} \right)} + \overline{\left(\frac{\partial v'_x}{\partial y} \right) \left(\frac{\partial v'_x}{\partial y} \right)} + \overline{\left(\frac{\partial v'_x}{\partial z} \right) \left(\frac{\partial v'_x}{\partial z} \right)}$$

$$\begin{aligned}
 & + \overline{\left(\frac{\partial v'_y}{\partial x}\right)\left(\frac{\partial v'_y}{\partial x}\right)} + 2 \overline{\left(\frac{\partial v'_y}{\partial y}\right)\left(\frac{\partial v'_y}{\partial y}\right)} + \overline{\left(\frac{\partial v'_y}{\partial z}\right)\left(\frac{\partial v'_y}{\partial z}\right)} \\
 & + \overline{\left(\frac{\partial v'_z}{\partial x}\right)\left(\frac{\partial v'_z}{\partial x}\right)} + \overline{\left(\frac{\partial v'_z}{\partial y}\right)\left(\frac{\partial v'_z}{\partial y}\right)} + 2 \overline{\left(\frac{\partial v'_z}{\partial z}\right)\left(\frac{\partial v'_z}{\partial z}\right)} \\
 & + 2 \overline{\left(\frac{\partial v'_x}{\partial y}\right)\left(\frac{\partial v'_y}{\partial x}\right)} + 2 \overline{\left(\frac{\partial v'_x}{\partial z}\right)\left(\frac{\partial v'_z}{\partial x}\right)} + 2 \overline{\left(\frac{\partial v'_y}{\partial z}\right)\left(\frac{\partial v'_z}{\partial y}\right)}
 \end{aligned}$$

Además se tiene que el vector de flujo de calor promedio $\bar{\mathbf{q}}^{(v)}$ está dado por:

$$\bar{\mathbf{q}}^{(v)} = k \begin{bmatrix} \frac{\partial \bar{T}}{\partial x} \\ \frac{\partial \bar{T}}{\partial y} \\ \frac{\partial \bar{T}}{\partial z} \end{bmatrix} = k \nabla \bar{T}$$

y que la función de disipación viscosa $\bar{\Phi}_v^{(v)}$ está definida como:

$$\begin{aligned}
 \bar{\Phi}_v^{(v)} &= \mu \left[2 \left(\frac{\partial \bar{v}_x}{\partial x}\right)^2 + 2 \left(\frac{\partial \bar{v}_y}{\partial y}\right)^2 + 2 \left(\frac{\partial \bar{v}_z}{\partial z}\right)^2 \right] \\
 &+ \left[\left(\frac{\partial \bar{v}_x}{\partial y} + \frac{\partial \bar{v}_y}{\partial x}\right)^2 + \left(\frac{\partial \bar{v}_x}{\partial z} + \frac{\partial \bar{v}_z}{\partial x}\right)^2 + \left(\frac{\partial \bar{v}_y}{\partial z} + \frac{\partial \bar{v}_z}{\partial y}\right)^2 \right]
 \end{aligned}$$

El conjunto de ecuaciones anterior no es un conjunto cerrado y para resolverlo se requieren ecuaciones adicionales para las componentes desconocidas de los esfuerzos de Reynolds [22]. El proceso de cálculo se puede definir como modelado de turbulencia, donde cualquier modelo de turbulencia tiene que cerrar el sistema de ecuaciones expresando las componentes de esfuerzos fluctuantes no lineales sólo en términos de las componentes promedio.

Modelos de Turbulencia.

Como se ha vislumbrado, la turbulencia provoca la aparición de remolinos con una amplia gama de escalas de longitud y tiempo que interactúan de forma dinámicamente compleja. Debido a que ciertas aplicaciones requieren, ya sea evitar o promover la turbulencia, se han desarrollado métodos numéricos para capturar sus efectos. De acuerdo a [19], los métodos se pueden agrupar en tres categorías:

- Modelos de turbulencia para ecuaciones de Navier-Stokes con Promedio de Reynolds (RANS): El análisis se enfoca en el flujo promedio y los efectos de la turbulencia en las

propiedades promedio del flujo. Previo a la aplicación de cualquier métodos numérico, las ecuaciones de Navier-Stokes se promedian en el tiempo, donde aparecen términos adicionales debido a las interacciones entre las fluctuaciones turbulentas. Estos términos adicionales se modelan en lugar de solucionarse, por lo tanto, se denominan modelos de turbulencia.

- Simulación de remolinos grandes (LES): En lugar de promediar el tiempo, LES usa una operación de filtrado espacial para separar los remolinos más grandes y más pequeños. El método realiza el filtrado espacial de las ecuaciones no estacionarias de Navier-Stokes, con lo que los remolinos más grandes son resueltos y los remolinos más pequeños son modelados. Los efectos sobre el flujo resuelto (flujo promedio y grandes remolinos) debido a los remolinos más pequeños no resueltos se incluyen por medio del llamado modelo a escala de sub malla o esfuerzos SGS.
- Simulación numérica directa (DNS): Estas simulaciones resuelven el flujo promedio y todas las fluctuaciones de velocidad turbulenta. Las ecuaciones no estacionarias de Navier-Stokes se resuelven en mallas espaciales que son lo suficientemente finas como para resolver las escalas de longitud de Kolmogorov en las que tiene lugar la disipación de energía con pasos de tiempo pequeños para resolver el período de las fluctuaciones más rápidas.

Hasta ahora no se cuenta con un modelo de turbulencia de propósito general adecuado para una todos los casos de flujo turbulento, según [19] esto se debe a las diferencias en el comportamiento de los remolinos grandes y pequeños, donde los remolinos pequeños son casi isotrópicos y tienen un comportamiento universal, por otro lado, los remolinos grandes, que interactúan y extraen energía del flujo medio, son anisotrópicos y su comportamiento está dictado por la geometría del dominio del problema, las condiciones de contorno y las fuerzas externas o de cuerpo.

En la mayoría de los casos de análisis CFD los detalles de las fluctuaciones turbulentas son innecesarios y la información sobre las propiedades promedio es suficiente, por lo tanto, los cálculos de flujo turbulento se realizan con modelos de turbulencia RANS, ya que son computacionalmente más económicos [19]. Los modelos de turbulencia RANS generalmente se clasifican de acuerdo al número de ecuaciones de transporte adicionales que deben resolverse junto con las ecuaciones de flujo como se muestra en la Tabla 6.

Tabla 6. Modelos de Turbulencia según el número de ecuaciones

Ecuaciones de transporte adicionales	Modelo de Turbulencia
0	Longitud de Mezclado
1	Spalart-Almaraz
2	$k-\varepsilon$
	$k-\omega$
7	RSM

Método de Volumen Finito.

Encontrar una solución exacta para las ecuaciones RANS es muy complicado, sin embargo, es posible encontrar soluciones aproximadas en puntos específicos si se supone división o discretización espacial mediante lo que se conoce como mallado del dominio computacional, donde los puntos de intersección de la malla, denominados nodos de malla, funcionan como ubicaciones de medición, así los valores de cualquier propiedad ϕ se pueden colocar en el centro o en las esquinas de cada uno de los elementos de la malla [19].

El principio fundamental del Método de Volumen Finito (MVF), como se describe en [19], consiste en reemplazar la solución exacta y continua de las ecuaciones RANS por un conjunto de valores individuales, esto implica que se está aproximando y no resolviendo la forma en que la propiedad ϕ cambia en el área al convertir la ecuación original en un conjunto de ecuaciones algebraicas. Estas ecuaciones se basan en los principios físicos que gobiernan a la propiedad ϕ y relacionan el valor de ϕ en cada punto con los valores de los nodos adyacentes. Al resolver este sistema de ecuaciones, se obtienen los valores aproximados de ϕ en todos los nodos de la malla.

Se debe tener en cuenta que el dominio físico se somete a un proceso de discretización geométrica, que da como resultado una malla utilizada para resolver las ecuaciones de conservación de masa, cantidad de movimiento y energía; esto implica subdividir el dominio en una colección de celdas o elementos no superpuestos conformaran el dominio computacional, ya sea en forma bidimensional o tridimensional. Para facilitar la discretización de las ecuaciones gobernantes en la malla, se requiere información sobre la topología del dominio y ciertas propiedades geométricas. Más adelante, durante el proceso de discretización matemática, se integran las ecuaciones gobernantes sobre cada elemento dentro de la malla, esto genera un sistema de ecuaciones algebraicas, donde cada ecuación relaciona el valor de la variable ϕ en un elemento específico con los valores de sus elementos vecinos [19].

La integración de las ecuaciones sobre cada elemento se denomina ensamble local, mientras que la construcción de todo el sistema de ecuaciones a partir de estas contribuciones se denomina ensamble global [19], donde la conectividad de elementos establece la correspondencia entre la matriz de ensamble local y la matriz global. De esta manera, las ecuaciones algebraicas ensambladas en una matriz global y vectores se pueden expresar en la forma:

$$\mathbf{A}\phi = \mathbf{B}$$

donde la variable ϕ se define en cada elemento interior y en la frontera del dominio computacional. Los valores de frontera para ϕ se obtienen a partir de las condiciones de frontera especificadas.

En términos generales, como lo describe [19], el valor de una variable ϕ en un punto de la malla sólo influye en la distribución de ϕ dentro de su vecindad inmediata, a medida que aumenta el número de elementos de la malla, se espera que la solución obtenida de las ecuaciones discretizadas converja hacia la solución exacta de la ecuación diferencial original. Este comportamiento se debe a que a medida que los elementos se vuelven más pequeños, los cambios en ϕ entre elementos vecinos se vuelven insignificantes, haciendo que los detalles del supuesto del perfil sean menos significativos.

Si se analiza el flujo de un fluido considerando únicamente la difusión pura en estado estacionario, al utilizar el MVF se puede derivar la ecuación de difusión a partir de la ecuación de transporte general para la propiedad ϕ eliminando los términos transitorio y convectivo. Esto es:

$$\nabla \cdot (\Gamma \nabla \phi) + S_\phi = 0$$

La integración del volumen de control produce la siguiente ecuación:

$$\int_{CV} \nabla \cdot (\Gamma \nabla \phi) dV + \int_{CV} S_\phi dV = \int_A \mathbf{n} \cdot (\Gamma \nabla \phi) dA + \int_{CV} S_\phi dV = 0$$

De esta manera la difusión en estado estacionario de una propiedad ϕ en un dominio unidimensional delimitado espacialmente entre los puntos A y B se rige por:

$$\frac{d}{dx} \left(\Gamma \frac{d\phi}{dx} \right) + S = 0$$

donde Γ es el coeficiente de difusión y S es el término fuente.

Ahora, como lo detalla [19], se debe dividir el dominio en volúmenes de control discretos, por lo tanto se crean una serie de nodales en el espacio entre A y B. Los límites (o caras) de los volúmenes de control se ubican en una zona media entre los nodos adyacentes. Así, cada nodo está rodeado por un volumen o celda de control. Para facilitar la descripción, un punto nodal se identifica con P y sus vecinos, dentro de una geometría unidimensional, se identifican con W y E , los nodos al oeste y al este respectivamente. La cara del lado oeste del volumen de control se denomina w y la cara del volumen de control del lado este se denomina e . Las distancias entre los nodos W y P , y entre los nodos P y E , se identifican mediante δx_{WP} y δx_{PE} respectivamente. De manera similar, las distancias entre la cara w y el punto P y entre P y la cara e se denotan por δx_{wP} y δx_{Pe} respectivamente. Donde el ancho del volumen de control es $\Delta x = \delta x_{we}$.

Considerando lo anterior, para un volumen de control definido se tiene

$$\int_{\Delta V} \frac{d}{dx} \left(\Gamma \frac{d\phi}{dx} \right) dV + \int_{\Delta V} S dV = \left(\Gamma A \frac{d\phi}{dx} \right)_e - \left(\Gamma A \frac{d\phi}{dx} \right)_w + \bar{S} \Delta V = 0$$

Donde A es el área de la sección transversal de la cara del volumen de control, ΔV es el volumen y \bar{S} es el valor promedio de la fuente S sobre el volumen de control. La ecuación establece que el flujo difusivo de ϕ que sale de la cara ubicada en e menos el flujo que entra en la cara ubicada en w es igual a la generación de ϕ , es decir, se tiene una ecuación balance para ϕ sobre el volumen de control. Para llegar a formas útiles de las ecuaciones discretizadas, se requieren el coeficiente de difusión de la interfaz Γ y el gradiente $\frac{d\phi}{dx}$ en las caras e y w . Los valores de la propiedad ϕ y el coeficiente de difusión Γ se definen y evalúan en los puntos nodales. Para calcular los gradientes en las caras del volumen de control se utiliza una distribución lineal de propiedades entre puntos nodales, denominada diferenciación central [19]. En una malla uniforme, los valores interpolados linealmente para Γ_w y Γ_e están dados por:

$$\Gamma_w = \frac{\Gamma_W + \Gamma_P}{2}$$

$$\Gamma_e = \frac{\Gamma_P + \Gamma_E}{2}$$

Y los términos de flujo difusivo se evalúan como

$$\left(\Gamma A \frac{d\phi}{dx} \right)_w = \Gamma_w A_w \left(\frac{\phi_P - \phi_W}{\delta x_{WP}} \right)$$

$$\left(\Gamma A \frac{d\phi}{dx} \right)_e = \Gamma_e A_e \left(\frac{\phi_E - \phi_P}{\delta x_{PE}} \right)$$

En varios casos de flujo, el término fuente S puede ser función de la variable dependiente, por lo tanto, en el MVF el término fuente se aproxima mediante una forma lineal:

$$\bar{S} \Delta V = S_u + S_p \phi_P$$

Al sustituir las ecuaciones anteriores en la ecuación para un volumen de control se tiene:

$$\Gamma_e A_e \left(\frac{\phi_E - \phi_P}{\delta x_{PE}} \right) - \Gamma_w A_w \left(\frac{\phi_P - \phi_W}{\delta x_{WP}} \right) + S_u + S_p \phi_P = 0$$

Al reorganizar los términos se llega a:

$$\left(\frac{\Gamma_e}{\delta x_{PE}} A_e + \frac{\Gamma_w}{\delta x_{WP}} A_w - S_p \right) \phi_P = \left(\frac{\Gamma_w}{\delta x_{WP}} A_w \right) \phi_W + \left(\frac{\Gamma_e}{\delta x_{PE}} A_e \right) \phi_E + S_u$$

Finalmente se pueden agrupar varios términos en nuevas variables y escribir la ecuación como:

$$a_P \phi_P = a_W \phi_W + a_E \phi_E + S_u$$

Para plantear un problema se deben establecer ecuaciones discretizadas en cada uno de los puntos nodales, sin embargo, en los volúmenes de control adyacentes a los límites del dominio, la ecuación general discretizada se modifica para incorporar condiciones de frontera. Para resolver el sistema algebraico de ecuaciones lineales se puede utilizar una técnica de solución matricial con el fin de obtener la distribución de la propiedad ϕ en los puntos nodales [19].

Si se amplía el MVF para cubrir problemas donde el movimiento del fluido es significativo, se deben tener en cuenta los efectos de la convección. La ecuación de convección-difusión en estado estacionario se puede derivar de la ecuación de transporte para una propiedad ϕ eliminando el término transitorio como sigue:

$$\nabla \cdot (\rho \phi \mathbf{v}) = \nabla \cdot (\Gamma \nabla \phi) + S_\phi$$

La integración sobre un volumen de control produce:

$$\int_A \mathbf{n} \cdot (\rho \phi \mathbf{v}) dA = \int_A \mathbf{n} \cdot (\Gamma \nabla \phi) dA + \int_{CV} S_\phi dV$$

Atendiendo la descripción de [19], el lado izquierdo proporciona el flujo convectivo neto y el lado derecho contiene el flujo difusivo neto y la generación o destrucción de la propiedad ϕ dentro del volumen de control. Ahora, el principal problema en la discretización de los términos convectivos es el cálculo del valor de la propiedad transportada ϕ en las caras del volumen de control y su flujo convectivo a través de estas fronteras. En este punto, el método de diferenciación central utilizado previamente para obtener ecuaciones discretizadas para los términos de difusión y fuente podría parecer una buena aproximación para los términos convectivos, sin embargo, el proceso de difusión afecta la distribución de una cantidad transportada ϕ a lo largo de los gradientes en todas las direcciones, mientras que la convección influye solo en la dirección del flujo.

En ausencia de fuentes, la convección y difusión estables de una propiedad ϕ en un campo de flujo unidimensional dado el vector \mathbf{v} se rige por:

$$\frac{d}{dx} (\rho u \phi) = \frac{d}{dx} \left(\Gamma \frac{d\phi}{dx} \right)$$

El flujo también debe satisfacer la ecuación de continuidad, por lo que:

$$\frac{d(\rho u)}{dx} = 0$$

Si se considera un volumen de control unidimensional, la integración de la ecuación de transporte sobre el volumen de control produce:

$$(\rho u A \phi)_e - (\rho u A \phi)_w = \left(\Gamma A \frac{d\phi}{dx} \right)_e - \left(\Gamma A \frac{d\phi}{dx} \right)_w$$

De igual manera, de la integración de la ecuación de continuidad se obtiene:

$$(\rho u A)_e - (\rho u A)_w = 0$$

Para obtener las ecuaciones discretizadas, se deben obtener aproximaciones de los términos de convección-difusión de las ecuaciones anteriores. Si se definen dos nuevas variables F y D para representar el flujo de masa convectivo por unidad de área y la conductancia de difusión en las caras de las celdas se tiene:

$$F = \rho u$$

$$D = \frac{\Gamma}{\delta x}$$

Los valores de las variables F y D en las caras de las celdas se pueden escribir como:

$$F_w = (\rho u)_w$$

$$F_e = (\rho u)_e$$

$$D_w = \frac{\Gamma_w}{\delta x_{WP}}$$

$$D_e = \frac{\Gamma_e}{\delta x_{PE}}$$

Suponiendo una malla rectangular estructurada donde $A_w = A_e = A$, se puede dividir ambos lados de la ecuación por el área A . Así, al emplear el método de diferenciación central para representar la contribución de los términos de difusión en el lado derecho. La ecuación integrada de convección-difusión se puede escribir de la siguiente manera:

$$F_e \phi_e - F_w \phi_w = D_e (\phi_E - \phi_P) - D_w (\phi_P - \phi_W)$$

Mientras que la ecuación de continuidad queda definida como:

$$F_e - F_w = 0$$

Hasta el momento, se supone que el campo de velocidades es conocido, lo que define los valores de F_e y F_w . Para resolver la ecuación de convección-difusión se necesita calcular la propiedad transportada ϕ en las caras e y w .

Para una malla uniforme, se pueden escribir los valores en las caras de las celdas de la propiedad ϕ como:

$$\phi_e = \frac{(\phi_P + \phi_E)}{2}$$

$$\phi_w = \frac{(\phi_W + \phi_P)}{2}$$

La sustitución de las expresiones anteriores en los términos convectivos produce:

$$\frac{F_e}{2}(\phi_P + \phi_E) - \frac{F_w}{2}(\phi_W + \phi_P) = D_e(\phi_E - \phi_P) - D_w(\phi_P - \phi_W)$$

Se pueden reorganizar los términos para llegar a:

$$\left[\left(D_w - \frac{F_w}{2} \right) + \left(D_e + \frac{F_e}{2} \right) \right] \phi_P = \left(D_w + \frac{F_w}{2} \right) \phi_W + \left(D_e - \frac{F_e}{2} \right) \phi_E$$

$$\left[\left(D_w + \frac{F_w}{2} \right) + \left(D_e - \frac{F_e}{2} \right) + (F_e - F_w) \right] \phi_P = \left(D_w + \frac{F_w}{2} \right) \phi_W + \left(D_e - \frac{F_e}{2} \right) \phi_E$$

Si se identifican los coeficientes de ϕ_W y ϕ_E como a_W y a_E , las expresiones de diferenciación centrales para la ecuación de convección-difusión discretizada son:

$$a_P \phi_P = a_W \phi_W + a_E \phi_E$$

Para resolver el problema unidimensional de convección-difusión se deben escribir ecuaciones discretizadas de la forma anterior para todos los nodos de la cuadrícula. Esto produce un conjunto de ecuaciones algebraicas que se resuelven para obtener la distribución de la propiedad transportada ϕ [19].

Solucionadores de Ecuaciones Lineales.

Como se ha comentado, el resultado del proceso de discretización es un sistema de ecuaciones lineales de la forma $\mathbf{A}\phi=\mathbf{B}$ donde las incógnitas ϕ , ubicadas en los centroides de los elementos de malla, son los valores buscados. En este sistema, los coeficientes de las variables desconocidas que constituyen la matriz \mathbf{A} son el resultado del procedimiento de linearización y la geometría de la malla, mientras que el vector \mathbf{b} contiene todas las fuentes, constantes, condiciones de frontera y componentes no linearizables. De acuerdo a [22], las técnicas para resolver sistemas lineales de ecuaciones generalmente se agrupan en métodos

directos e iterativos, con muchos subgrupos en cada categoría. Como los problemas de flujo son altamente no lineales, los coeficientes resultantes de su proceso de linearización generalmente dependen de la solución. Por este motivo y dado que no se necesita una solución precisa en cada iteración, los métodos directos se han utilizado únicamente en las aplicaciones de CFD. Los métodos iterativos, por otro lado, han sido más populares porque son más adecuados para este tipo de aplicaciones que requieren un menor costo computacional por iteración y menor memoria.

El punto de partida para cualquier solucionador lineal es el conjunto de ecuaciones generadas por el proceso de discretización, que se escriben matemáticamente como:

$$\mathbf{A}\boldsymbol{\phi} = \mathbf{B}$$

Donde \mathbf{A} es la matriz de coeficientes de elementos a_{ij} , $\boldsymbol{\phi}$ el vector de variables desconocidas ϕ_i , y \mathbf{b} el vector de fuentes b_i . Usando la forma matricial, la forma expandida viene dada por:

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N-1} & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N-1} & a_{2N} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN-1} & a_{NN} \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_N \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix}$$

Según [22], cada fila en la matriz representa una ecuación definida sobre un elemento del dominio computacional, y los coeficientes distintos de cero son aquellos relacionados con los vecinos de ese elemento. Como una celda está conectada solo a unos pocos vecinos, y su número depende de la conectividad de los elementos en el dominio discretizado, muchos de los coeficientes son ceros y la matriz \mathbf{A} resultante es dispersa (es decir, los coeficientes distintos de cero son una pequeña fracción de la matriz). Si además el método utiliza un sistema de mallado estructurado, la matriz \mathbf{A} se agrupará con todos los elementos que no sean cero alineados a lo largo de algunas diagonales.

Como se mencionó anteriormente, las técnicas para resolver sistemas de ecuaciones algebraicas se dividen en dos categorías: métodos directos e iterativos. En un método directo, la matriz \mathbf{A} se invierte y la solución $\boldsymbol{\phi}$ se calcula en un paso como $\boldsymbol{\phi} = \mathbf{A}^{-1}\mathbf{B}$, sin embargo cuando la matriz \mathbf{A} es grande, calcular su inversa es computacionalmente costoso, por otro lado, con los solucionadores algebraicos iterativos, el algoritmo de la solución se aplica repetidamente hasta que se alcanza un nivel de convergencia predefinido [22].

Métodos Iterativos.

Los métodos iterativos son más atractivos para problemas que involucran matrices dispersas, ya que la solución del sistema linearizado se convierte en parte del proceso de la solución iterativa. Dos métodos iterativos populares son Jacobi y Gauss-Seidel. Como se menciona en [22], la tasa de convergencia de los métodos iterativos depende de las propiedades de la

matriz de iteración, que depende de la matriz de coeficientes. En base a esto, un método iterativo realiza una transformación del sistema de ecuaciones en un equivalente que tenga la misma solución, pero de mejores características. De esta manera, los valores propios del sistema equivalente están más agrupados, lo que permite que la solución iterativa se obtenga más rápido.

La baja tasa de convergencia de los métodos de Gauss-Seidel y Jacobi incentivo el desarrollo de metodologías para acelerar la tasa de convergencia de los solucionadores iterativos. Como se menciona en [22], se comenzó a emplear pre acondicionadores avanzados, en donde se realiza una factorización incompleta de la matriz original de coeficientes \mathbf{A} . Esta implementación es substancial y remarcable, ya que una factorización completa de \mathbf{A} en una matriz triangular inferior \mathbf{L} y una superior \mathbf{U} es equivalente a una solución directa, la cual es costosa computacionalmente. De esta manera, se tiene que el pre acondicionador más común está basado en la factorización Incompleta LU (ILU), con versiones simplificadas como el pre acondicionador diagonal LU (DILU) y el diagonal IC (DIC) basado en la factorización incompleta de Cholesky.

Dentro de los métodos iterativos para resolver sistemas de ecuaciones algebraicos lineales existen también los métodos de gradiente, los cuales incluyen los métodos de pendiente pronunciada y de gradiente conjugado. Como lo establece [22], el método de pendiente pronunciada garantiza la convergencia, su funcionamiento es lento debido a oscilaciones alrededor de los mínimos locales, lo que ocasiona una búsqueda repetida en la misma dirección. Para evitar este comportamiento, cada nueva búsqueda debe ir en una dirección distinta mediante la selección un conjunto de direcciones de búsqueda ortogonales a la matriz \mathbf{A} , creando así el método de Gradiente Conjugado (CG). La tasa de convergencia del método CG puede aumentarse mediante el pre acondicionamiento, para lo cual se han desarrollado pre-acondicionadores con un amplio rango de sofisticación que varía desde una matriz diagonal, cuyos elementos son los elementos diagonales de la matriz original \mathbf{A} , hasta los más sofisticados que utilizan la factorización de incompleta de Cholesky. Por lo tanto, el método CG siempre debe usarse con un acondicionador cuando se resuelven grandes sistemas de ecuaciones (PCG).

La matriz de coeficientes resultante de la discretización de la ecuación de difusión y algunas otras ecuaciones como la presión incompresible o la ecuación de corrección de presión son simétricas, lo que lleva a sistemas simétricos que pueden resolverse utilizando el método CG. Sin embargo, como lo menciona [22], la matriz \mathbf{A} obtenida a partir de la discretización de la ecuación de conservación general que surge en las aplicaciones de CFD es asimétrica, lo que produce un sistema asimétrico de ecuaciones. Para poder resolver este sistema utilizando el método CG, debe transformarse en uno simétrico. Una manera para lograr esto es mediante la siguiente operación:

$$\begin{bmatrix} 0 & \mathbf{A} \\ \mathbf{A}^T & 0 \end{bmatrix} \begin{bmatrix} \hat{\phi} \\ \phi \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ 0 \end{bmatrix}$$

Cuando se aplica a este procedimiento, el método CG da como resultado dos secuencias de vectores similares a CG, debido esto el método se conoce como gradiente bi-conjugado (BiCG). El método BiCG necesita una multiplicación con la matriz de coeficientes y con su transpuesta en cada iteración, lo que resulta en casi el doble del esfuerzo computacional requerido por el método CG por iteración [22].

Métodos Multigrid.

La tasa de convergencia de los métodos iterativos se deteriora drásticamente a medida que aumenta el tamaño del sistema algebraico, lo cual ha constituido una limitación para los solucionadores iterativos. Según [22], los métodos multigrid mejoran la eficiencia de los solucionadores iterativos al garantizar que los errores de baja frecuencia resultantes que surgen de la aplicación de un suavizado en cualquier nivel de malla se transformen en errores de mayor frecuencia en un nivel de malla más grueso. Al utilizar una jerarquía de malla gruesas, los métodos multigrid pueden superar la degradación de la convergencia.

Los métodos multigrid son altamente complicados y computacionalmente más costoso que su contraparte, los métodos de gradiente conjugado. Una descripción de estos métodos según [22] a grandes rasgos es que la malla gruesa se puede formar utilizando la topología y la geometría de la malla más fina, este enfoque se conoce como el método *Geometric-Algebraic Multi-Grid* (GAMG). Por el contrario, el método *Algebraic Multi-Grid* (AMG) no se necesita ni se usa directamente información geométrica y el proceso es puramente algebraico, con las ecuaciones reconstruidas en cada nivel a partir de aquellas del nivel más fino, a través del proceso de aglomeración. Este enfoque se puede utilizar para crear solucionadores lineales altamente eficientes y robustos para mallas altamente anisótropas y/o problemas con grandes cambios en los coeficientes de sus ecuaciones. En cualquiera de los dos enfoques, se utiliza un procedimiento de ciclo multigrid para guiar el recorrido de las distintas jerarquías de malla. Cada recorrido de una malla fina a una gruesa implica: (i) un procedimiento de restricción, (ii) la configuración o actualización del sistema de ecuaciones para el nivel de la malla gruesa, y (iii) la aplicación de varias iteraciones más suaves. Un recorrido de una malla gruesa a uno más fino requiere: (i) un procedimiento de prolongación, (ii) la corrección de los valores de campo en el nivel más fino, y (iii) la aplicación de varias iteraciones más suaves en las ecuaciones construidas durante restricción.

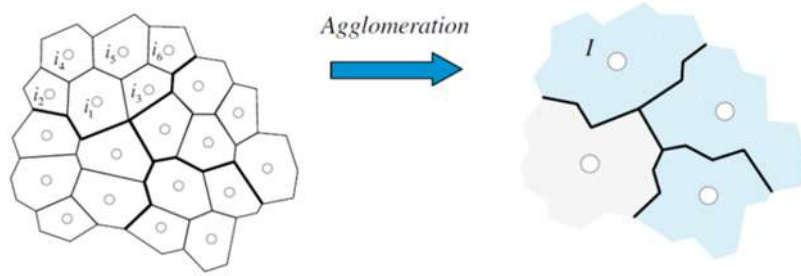


Figura 11. Aglomeración de un nivel de malla fina para formar un nivel de malla grueso [22]

Capítulo IV. Descripción del Software y Hardware.

Como se ha declarado el presente proyecto involucra la creación de un banco virtual basado en HPC para pruebas de modelo de turbinas hidráulicas tipo Francis empleando GPUs. Esta sección presenta información que ayudara a la mejor comprensión del software y hardware empleado, así como su importancia y la complejidad de su implementación.

Ubuntu.

Primeramente comenzaremos con el sistema operativo empleado, ya que este ofrece características útiles, como es la implementación de software de CFD especializado desarrollado nativamente para este sistema operativo.

Ubuntu [23] está construido sobre la base de Linux, miembro de la familia Unix. En un inicio, Unix estuvo completamente basado en línea de comandos hasta que aparecieron las GUIs (*Graphical User Interface*) en 1973 [24].

A continuación se enlistan las especificaciones que una computadora debe cumplir como mínimo para instalar Ubuntu [24]:

- Procesador de 1 GHz x86 (Pentium 4 o superior)
- 1 Gb de memoria del sistema (RAM)
- 8.6 Gb de espacio en disco (se recomienda al menos 15 Gb)
- Soporte de video con una resolución de 1024×768
- Soporte de audio (recomendado, pero no obligatorio)
- Una conexión a Internet (muy recomendada, pero no obligatoria)

Ubuntu y sus derivados están disponibles en dos versiones: 32 y 64 bits. Esta diferencia se refiere a la forma en que las computadoras procesan la información. Las computadoras capaces de ejecutar software de 64 bits pueden procesar más información que las computadoras que ejecutan software de 32 bits; sin embargo, los sistemas de 64 bits requieren más memoria para poder hacerlo, aunque obtienen mejoras de rendimiento [24].

GPUs en Ubuntu.

La mayoría de las tarjetas gráficas son fabricadas por tres compañías: Intel, AMD/ATI y NVIDIA. Ubuntu tiene soporte para dispositivos gráficos fabricados por las compañías anteriores, esto significa que no se tiene que instalar ningún controlador. Los controladores que se utilizan por defecto para potenciar dispositivos gráficos son de código abierto. Sin embargo, en algunos casos, un controlador propietario (controlador restringido) proporcionado por la empresa fabricante de la GPU puede proporcionar un mejor rendimiento o características que no están presentes en el controlador de código abierto. En esos casos, es posible que se requiera instalar el controlador restringido proporcionado por el fabricante, ya que Ubuntu no instala controladores restringidos por defecto, sino que permite

al usuario hacerlo según sus necesidades [24]. Lo anterior es de vital importancia para la implementación de las GPUs en HPC.

Software en Ubuntu.

Muchas aplicaciones diseñadas para Microsoft Windows no se ejecutarán en Ubuntu. Para la gran mayoría de las tareas informáticas cotidianas, se encuentran aplicaciones alternativas disponibles en Ubuntu. Alternativamente, algunas aplicaciones desarrolladas para Windows funcionan en Ubuntu con el programa Wine [24].

Ubuntu y varias otras variantes de Linux utilizan una colección de herramientas de software llamadas sistema de gestión de paquetes o gestor de paquetes. Un gestor de paquetes es una colección de herramientas que facilita la instalación, eliminación, actualización y configuración del software. Un sistema de administración de paquetes tiene una base de datos de software llamada repositorio (*repository*) donde el software individual se organiza en una colección llamada paquete. De forma predeterminada, Ubuntu proporciona un punto centralizado con dos formas diferentes de explorar los repositorios para buscar, instalar y eliminar software [24]:

- La aplicación de software de Ubuntu
- Línea de comando apt-get

Aunque Ubuntu proporciona una gran librería de aplicaciones, puede presentarse de caso de requerir una aplicación no disponible en estos repositorios. Ubuntu facilita la adición de repositorios de terceros adicionales a su lista de fuentes de software. Los repositorios más comunes agregados a Ubuntu se llaman PPAs. Los PPA (*Personal Package Archive*) son repositorios en línea utilizados para alojar las últimas versiones de paquetes de software, proyectos digitales y otras aplicaciones que no están disponibles en los repositorios oficiales [24].

Ya que la implementación de GPUs para uso en CFD dentro de Ubuntu requiere de un software especializado y flexible, se empleó OpenFOAM y algunas librerías o derivaciones, las cuales no siempre están disponibles en los repositorios de Ubuntu, por lo cual estas últimas alternativas fueron de gran utilidad.

OpenFOAM.

OpenFOAM es un software CFD gratuito y de código abierto lanzado y desarrollado por OpenCFD Ltd desde 2004. OpenFOAM tiene una amplia gama de características para resolver desde complejos flujos de fluidos que involucran reacciones químicas, turbulencia y transferencia de calor, hasta acústica, mecánica sólida y electromagnética [25].

Las versiones de OpenFOAM fueron originalmente lanzadas por OpenCFD, hasta la versión 2.0.1. Una nueva entidad, OpenFOAM Foundation, se creó en 2011 tras la adquisición de

OpenCFD por parte de Silicon Graphics International (SGI). OpenCFD continuó lanzando OpenFOAM a través de OpenFOAM Foundation hasta la versión 2.3.1. En 2012, OpenCFD se convirtió en parte del ESI Group. A partir de 2016, OpenCFD lanzó OpenFOAM directamente una vez más, comenzando con el lanzamiento de OpenFOAM v3.0+. OpenFOAM se lanza cada seis meses y es probado de forma independiente por especialistas en aplicaciones, socios de desarrollo y los clientes seleccionados de ESI-OpenCFD, y respaldado por la infraestructura de ESI a nivel mundial [25].

foam-extend.

Una gran cantidad de características de OpenFOAM se han perdido desde la versión 1.3, ya que la base de código se ha reducido en más del 40%. De este modo, el proyecto foam-extend es una bifurcación de OpenFOAM. Es un proyecto abierto que acoge e integra contribuciones de todos los usuarios y desarrolladores. Anteriormente conocido como OpenFOAM-dev y OpenFOAM-extend, contiene correcciones de errores y mejoras de rendimiento, así como extensiones y características adicionales proporcionadas por los contribuyentes de la comunidad, como soporte para mallas dinámicas y cambios topológicos, extensiones de turbo maquinaria, incluida la interpolación general de malla (GGI), GGI cíclica y plano de mezclado, soporte para malla acoplada en bloque, transferencia de calor conjugada implícitamente acoplada y otros acoplamientos físicos, método de área finita, capacidad integral de malla dinámica (movimiento y cambios topológicos) y soporte para GPUs. La versión 4.0, apodada "Guimaraes", es la versión actual de foam-extend [26].

Entorno de OpenFOAM.

En el nivel más básico OpenFOAM es una librería C++ utilizada para crear ejecutables, conocidos como aplicaciones. OpenFOAM se distribuye con un gran conjunto de aplicaciones pre compiladas, pero los usuarios tienen la libertad de crear las propias o modificar las existentes. Las aplicaciones se dividen en dos categorías principales [27]:

- *solvers* que están diseñados para resolver un problema específico en mecánica del continuo computacional;
- utilidades que realizan tareas simples de pre y pos procesamiento, principalmente con manipulación de datos y cálculos algebraicos.

OpenFOAM se divide en un conjunto de librerías pre compiladas que se vinculan dinámicamente durante la compilación de los *solvers* y las utilidades. Las librerías, como las de modelos físicos, se suministran como código fuente (*source code*) para que los usuarios puedan agregar sus propios modelos a las librerías. OpenFOAM se suministra también con entornos de pre y post procesamiento. La interfaz para pre y post procesamiento son en sí mismas utilidades de OpenFOAM, lo que garantiza un manejo coherente de datos en todos los entornos [27].

OpenFOAM necesita leer un rango de estructuras de datos tales como cadenas, escalares, vectores, tensores, listas y campos. El formato de entrada/salida (I/O) de los archivos está diseñado para ser flexible con el fin de permitir al usuario modificar las I/O en las diversas aplicaciones. De esta manera OpenFOAM usa diccionarios como el medio más común de especificar datos. Un diccionario es una entidad que contiene un conjunto de entradas de datos que las I/O pueden recibir por medio de palabras clave. La mayoría de los archivos de datos de OpenFOAM son en sí mismos diccionarios que contienen un conjunto de entradas en forma de palabras clave [27].

Como protección contra la implementación de una operación sin sentido, OpenFOAM agrega dimensiones a los datos de campo y las propiedades físicas y además realiza la verificación de dimensiones en cualquier operación de tensores. El formato de I/O para un conjunto de dimensiones es de 7 escalares delimitados por corchetes, donde cada uno de los valores corresponde a la potencia de cada una de las unidades base de medida enumeradas en la Tabla 7 [27].

Tabla 7. Unidades base para SI y USCS [27].

No.	Property	SI unit	USCS unit
1	Mass	kilogram (kg)	pound-mass (lbm)
2	Length	metre (m)	foot (ft)
3	Time	— — — —	second (s) — — — —
4	Temperature	Kelvin (K)	degree Rankine (°R)
5	Quantity	kilogram-mole (kgmol)	pound-mole (lbmol)
6	Current	— — — —	ampere (A)
7	Luminous intensity	— — — —	candela (cd) — — — —

Durante cualquier ejecución, OpenFOAM comprueba que la malla cumpla un conjunto bastante estricto de restricciones de validez y dejará de funcionar si no se satisfacen las restricciones. De forma predeterminada, OpenFOAM define una malla de celdas poliédricas arbitrarias en 3-D, delimitadas por caras poligonales arbitrarias, es decir, las celdas pueden tener un número ilimitado de caras donde, para cada cara, no hay límite en el número de bordes ni ninguna restricción en su alineación. Una malla con esta estructura general se conoce en OpenFOAM como un *polyMesh*. Este tipo de malla ofrece una gran libertad en la generación y manipulación de malla, en particular cuando la geometría del dominio es compleja o cambia con el tiempo. El precio de la generalidad de malla absoluta es, sin embargo, que puede ser difícil convertir mallas generadas con herramientas convencionales. Es posible generar mallas utilizando otros paquetes y convertirlos al formato que utiliza OpenFOAM. Existen numerosas utilidades de conversión de malla. Algunos de los convertidores de malla más populares se enumeran a continuación [27]:

- `fluentMeshToFoam` lee un archivo de malla Fluent `.msh`, que funciona para casos 2D y 3D;

- `star4ToFoam` lee los archivos de malla STAR-CD/PROSTAR
- `gambitToFoam` lee un archivo neutro GAMBIT .neu
- `ansysToFoam` lee una malla I-DEAS escrita en formato ANSYS .ans
- `cfx4ToFoam` lee una malla CFX escrita en formato .geo

Lo anterior fue extremadamente útil para trabajar con OpenFOAM, debido a que OpenFOAM no cuenta con herramientas de mallado flexibles cuando se trabaja con geometrías complejas.

Paralelización en OpenFOAM.

El método de cómputo paralelo utilizado por OpenFOAM se conoce como descomposición de dominio, en el cual la geometría y los campos asociados se dividen en bloques y se asignan a los procesadores para la solución. El proceso de cálculo paralelo implica: descomposición de malla y campos, ejecutar la aplicación en paralelo y finalmente post procesamiento del caso descompuesto [27].

La malla y los campos se descomponen usando la utilidad *decomposePar*. El objetivo es dividir el dominio con un mínimo esfuerzo pero de tal manera que se garantice una solución económica computacionalmente. La geometría y los campos se dividen de acuerdo con un conjunto de parámetros especificados en un diccionario llamado *decomposeParDict* [27].

Un caso OpenFOAM descompuesto se puede ejecutar en paralelo utilizando la implementación de dominio público *openMPI*. Esta se puede ejecutar en una máquina de multiprocesador de forma muy simple, pero cuando se ejecuta en máquinas a través de una red, se debe crear un archivo que contenga los nombres de host de las máquinas. El archivo puede recibir cualquier nombre y ubicarse en cualquier ruta [27].

Después de que se ha solucionado un caso en paralelo, se puede reconstruir para el post procesamiento. El caso se reconstruye fusionando los conjuntos de directorios de tiempo de cada directorio de procesador en un único conjunto de directorios de tiempo. Se puede post procesar casos descompuestos usando el post procesador *paraFoam*. La simulación completa puede procesarse posteriormente al reconstruir el caso, aunque también es posible post procesar un segmento del dominio descompuesto de forma individual simplemente tratando el directorio de procesador individual como un caso en sí mismo [27].

CFD en OpenFOAM.

OpenFOAM incluye una amplia gama de *solvers*, cada uno diseñado para una clase de problema específico. Las ecuaciones y los algoritmos difieren de un *solver* a otro, de modo que la selección de alguno implica conocimiento básico del modelado y fenómeno físico para su caso particular.

Como en la mayoría de los problemas que involucran flujo de fluidos se deben considerar los efectos viscosos y la turbulencia. En OpenFOAM el diccionario *turbulenceProperties* es leído por cualquier *solver* que incluya modelado de turbulencia. Dentro de ese archivo la palabra clave *simulationType* controla el tipo de modelado de turbulencia que se utilizará:

- *laminar* no utiliza modelos de turbulencia;
- *RAS* usa el modelo de estrés promediado de Reynolds (RAS);
- *LES* utiliza la simulación de grandes remolinos (LES) o la simulación de remolino separado (DES).

Como se ha discutido previamente la realización de análisis CFD requiere del discretizado de las ecuaciones gobernantes del fenómeno. En OpenFOAM el diccionario *fvSchemes* establece los esquemas numéricos se deben asignar a los términos que aparecen en las ecuaciones gobernantes. Aunque la interpolación lineal es efectiva en muchos casos, OpenFOAM ofrece una amplia selección de esquemas de interpolación para todos los términos [27].

Adicionalmente los *solvers*, tolerancias y algoritmos se controlan desde el diccionario *fvSolution*, el cual contiene un conjunto de sub diccionarios que son específicos de cada *solver* [27].

El primer sub diccionario especifica cada *solver* lineal que se usa para cada ecuación discretizada; el término *linear solver* (solucionador lineal) se refiere al método de cálculo numérico para resolver el conjunto de ecuaciones lineales, distinto al *solver* de aplicaciones que describe el conjunto de ecuaciones y algoritmos para resolver un problema particular. La sintaxis para cada entrada dentro del apartado *solvers* usa una palabra clave relacionada con la variable que se está resolviendo. El *solver* se selecciona a través de palabras clave en OpenFOAM según la Tabla 8 [27].

Tabla 8. Solvers lineales [27].

Solver	Keyword
Preconditioned (bi-)conjugate gradient	PCG/PBiCG†
Solver using a smoother	smoothSolver
Generalised geometric-algebraic multi-grid	GAMG
Diagonal solver for explicit systems	diagonal

†PCG for symmetric matrices, PBiCG for asymmetric

Los *solvers* distinguen entre matrices simétricas y matrices asimétricas. La simetría de la matriz depende de la estructura de la ecuación que se esté resolviendo y, aunque el usuario pueda determinar esto, no es esencial ya que OpenFOAM producirá un mensaje de error para avisar al usuario si se ha seleccionado un *solver* inapropiado. Existe una gama de opciones

para el pre acondicionamiento de matrices en los solucionadores PCG/PBiCG. Los pre acondicionadores se enumeran en la Tabla 9 [27].

Tabla 9. Opciones para pre acondicionador [27].

Preconditioner	Keyword
Diagonal incomplete-Cholesky (symmetric)	DIC
Faster diagonal incomplete-Cholesky (DIC with caching)	FDIC
Diagonal incomplete-LU (asymmetric)	DILU
Diagonal	diagonal
Geometric-algebraic multi-grid	GAMG
No preconditioning	none

Adicional a los *solvers* PCG/PBiCG existe el método GAMG, el cual es más rápido que los métodos estándar cuando el aumento de velocidad resolviendo primero mallas más gruesas compensa los costos adicionales de refinamiento de malla y mapeo de datos de campo [27].

Los *solvers* que usan un *smoother* (suavizador), como GAMG, requieren que este se especifique, las opciones se enumeran en la Tabla 10. En general, *GaussSeidel* es la opción más confiable, pero para matrices malas, DIC puede ofrecer una mejor convergencia [27].

Tabla 10. Opciones de Smother [27].

Smoother	Keyword
Gauss-Seidel	GaussSeidel
Diagonal incomplete-Cholesky (symmetric)	DIC
Diagonal incomplete-Cholesky with Gauss-Seidel (symmetric)	DICGaussSeidel

Los *solvers* de matrices son iterativos, es decir, se basan en la reducción del *residual* de la ecuación sobre una sucesión de soluciones. El *residual* es una medida del error en la solución de modo que cuanto más pequeño es, más precisa es la solución [27].

Un segundo sub diccionario de *fvSolution* en OpenFOAM es *relaxationFactors* que controla la relajación, una técnica utilizada para mejorar la estabilidad de cálculo, particularmente en la resolución de problemas de estado estacionario. El funcionamiento consiste en limitar la cantidad que una variable cambia de una iteración a la siguiente, ya sea modificando la matriz de solución y la fuente antes de resolver un campo o modificando el campo directamente [27].

La mayoría de las aplicaciones de resolución de dinámica de fluidos en OpenFOAM utilizan los algoritmos PISO o SIMPLE. Estos algoritmos son procedimientos iterativos para resolver ecuaciones de velocidad y presión, PISO se usa para problemas transitorios y SIMPLE para estado estacionario. Ambos algoritmos se basan en evaluar algunas soluciones iniciales y luego corregirlas [27].

RapidCFD.

OpenFOAM no está desarrollado para hacer uso de GPUs en la solución de problemas. Lo que se ha realizado por parte de grupos de investigación y compañías ha sido desarrollar librerías que se compilan adicionalmente a OpenFOAM y de esta manera hacer uso de las GPUs en el cálculo, sin embargo durante la ejecución de las mismas surge un cuello de botella que no permite alcanzar el desempeño máximo del hardware. En este trabajo se hace uso de RapidCFD, el cual es un *toolbox* de CFD ejecutándose en *Compute Unified Device Architecture* (CUDA), donde los *solvers* de OpenFOAM han sido portados a Nvidia CUDA. Todo el cálculo se hace completamente en la GPU y no es necesario copiar datos entre la CPU y la GPU durante los cálculos. Todo esto para hacer el trabajo más rápido. Sus principales características son [28]:

- la mayoría de los *solvers* incompresibles y compresibles para mallas estáticas están disponibles
- todos los cálculos se hacen en la GPU
- sin sobrecarga para la copia de memoria entre GPU y CPU
- puede ejecutarse en paralelo en múltiples GPU

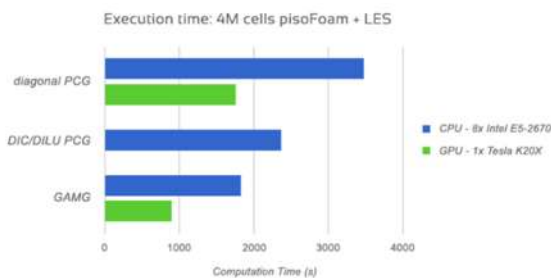


Figura 12. Comparación del tiempo de ejecución usando CPUs (OpenFOAM) y GPUs (RapidCFD) para un caso de 4 millones de celdas [28].

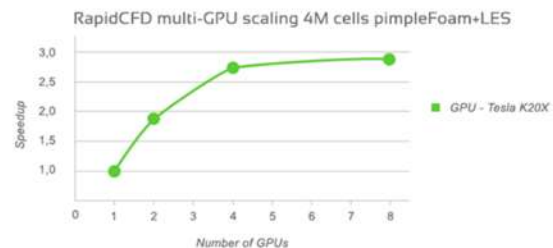


Figura 13. Speedup mediante el uso de GPUs (RapidCFD) para un caso de 4 millones de celdas [28].

RapidCFD se puede compilar usando el sistema de compilación estándar de OpenFOAM. Para simulaciones multi GPU, se requiere OpenMPI. Todos los archivos fuente de RapidCFD se compilan utilizando nvcc, por lo tanto, el proceso de compilación lleva mucho tiempo en comparación con la compilación estándar [29].

cufflink.

Cuda For FOAM Link (cufflink) [30] es una biblioteca de código abierto para vincular métodos numéricos basados en el lenguaje de programación CUDA de Nvidia y OpenFOAM. Actualmente, la biblioteca utiliza los solucionadores lineales Cusp [31] y los métodos de Thrust [32] para resolver el sistema lineal derivado de OpenFOAM y devolver el vector solución. Cufflink está diseñado para utilizar el paralelismo completo de OpenFOAM (a

través de la descomposición del dominio) para permitir el paralelismo multi-GPU al nivel del solucionador lineal del sistema. Sus principales características son [30]:

- Actualmente solo es compatible con foam-extend.
- Se dispone de un solucionador de gradiente conjugado basado en CUSP para matrices simétricas con opción de Pre acondicionador *diagonal* y AMG basado en el pre acondicionador de agregación suavizada.
- Se dispone de un solucionador de gradiente bi-conjugado estabilizado basado en CUSP para matrices asimétricas con opción de: Pre acondicionador *diagonal* y disperso aproximado inverso.
- Compatibilidad con GPU única.
- Soporte multi-GPU a través del paralelismo de OpenFOAM logrado mediante la descomposición de dominio.
- Compatibles con precisión sencilla y precisión doble. Los solucionadores de doble precisión se recomiendan sobre la precisión sencilla debido a los errores conocidos encontrados en el pre acondicionador de agregación suavizada en precisión simple.

La aceleración de cálculo depende en gran medida de la GPU utilizada y el CPU con el que se comparen los resultados [30].

CUDA.

CUDA es una plataforma de cómputo paralela y un modelo de programación desarrollado por NVIDIA para cómputo general en GPU, con la cual se pueden acelerar drásticamente las aplicaciones informáticas mediante el aprovechamiento de la potencia de las GPU. En las aplicaciones aceleradas por GPU, la parte secuencial de la carga de trabajo se ejecuta en el CPU, que está optimizado para el rendimiento de un único subproceso, mientras que la parte intensa de cálculo de la aplicación se ejecuta en miles de núcleos GPU en paralelo. Al usar CUDA, se programa en lenguajes como C, C ++, Fortran, Python y MATLAB y se logra acelerar el paralelismo [33].

CUDA fue desarrollado con varios objetivos [33]:

- Proporcionar un conjunto de extensiones a los lenguajes de programación estándar, como C, que permiten una implementación directa de algoritmos paralelos.
- Soportar el cálculo heterogéneo donde las aplicaciones usan tanto la CPU como la GPU. Las porciones seriales de las aplicaciones se ejecutan en la CPU y las porciones paralelas se descargan a la GPU, donde la CPU y la GPU se tratan como dispositivos separados que tienen sus propios espacios de memoria. Esta configuración también permite el cálculo simultáneo en el CPU y la GPU.

Las GPUs compatibles con CUDA tienen cientos de núcleos que pueden ejecutar colectivamente miles de subprocesos. Estos núcleos tienen recursos compartidos que

incluyen un archivo de registro y una memoria compartida. La memoria compartida en el chip permite que las tareas paralelas que se ejecutan en estos núcleos compartan datos sin enviarlos a través del bus de memoria del sistema [34].

Las librerías NVIDIA aceleradas por GPU proporcionan funciones altamente optimizadas que funcionan de 2x a 10x más rápido que las alternativas de CPU. Las librerías aceleradas por GPU para álgebra lineal, procesamiento de señales, procesamiento de imágenes y video sientan las bases para aplicaciones de cómputo intensivo en áreas tales como dinámica molecular, química computacional, imágenes médicas y exploración sísmica [33].

Para usar CUDA en un sistema, se necesita lo siguiente [34]:

- GPU con capacidad para soportar CUDA
- Una versión compatible de Linux con un compilador gcc y herramientas relacionadas
- NVIDIA CUDA Toolkit

CUDA Toolkit puede instalarse utilizando cualquiera de los dos mecanismos de instalación disponibles: paquetes específicos de distribución o un paquete independiente de la distribución linux. El paquete de distribución independiente tiene la ventaja de trabajar en un conjunto más amplio de distribuciones de Linux, pero no actualiza el sistema de administración de paquetes nativo de la distribución [34].

Programación en CUDA.

CUDA Toolkit se dirige a una clase de aplicaciones cuya parte de control se ejecuta como un proceso en una computadora de propósito general, ya sea en Linux o Windows, y que utiliza una o más GPU NVIDIA como coprocesadores para acelerar los trabajos paralelos. Tales trabajos son autónomos y pueden ser ejecutados y completados por un lote de subprocesos de GPU sin intervención del *host*. CUDA Toolkit admite el envío de trabajos de GPU por el *host* en forma de llamadas a procedimientos remotos. El código de GPU se implementa como una colección de funciones en lenguaje C, pero con anotaciones para distinguirlas del código del *host*, más anotaciones para distinguir los diferentes tipos de datos de memoria que existen en la GPU. Dichas funciones pueden tener parámetros, y pueden llamarse usando una sintaxis que es muy similar a la función de llamada C normal, pero extendida para poder especificar la matriz de subprocesos de GPU que deben ejecutar la función de llamada [35].

Los archivos fuente para aplicaciones CUDA consisten en una mezcla de código C++ convencional, más funciones de dispositivo GPU. La compilación con CUDA Toolkit separa las funciones del *device* del código del *host* y compila las funciones del *device* usando compiladores/ensambladores NVIDIA, mientras que el código del *host* se compila utilizando un compilador C/C++ de uso general. En la etapa de vinculación, se agregan librerías de ejecución de CUDA específicas para admitir llamadas a procedimientos remotos y para proporcionar manipulación explícita de la GPU, como la asignación de almacenamientos de

memoria de GPU y la transferencia de datos entre la GPU y el *host*. Esta trayectoria de compilación implica varios pasos de división, compilación, pre procesamiento y fusión para cada archivo fuente CUDA, y varios de estos pasos son diferentes para cada modo de compilación CUDA. El propósito del controlador del compilador CUDA *nvcc* es ocultar los detalles intrincados de la compilación CUDA. Además, en lugar de ser un controlador de compilación CUDA específico, *nvcc* imita el comportamiento del compilador GNU *gcc*, y acepta una variedad de opciones de compilación convencionales para definir macros y rutas para dirigir el proceso de compilación [35].

GPUs en CUDA.

Las primeras GPU se diseñaron como aceleradores gráficos, volviéndose más programables en los 90, culminando en la primera GPU de NVIDIA en 1999. Los investigadores y científicos rápidamente comenzaron a aplicar el excelente rendimiento de punto flotante de estas GPUs para cálculo de propósito general. En 2003, un equipo de investigadores dirigido por Ian Buck dio a conocer Brook, el primer modelo de programación ampliamente adoptado para extender C a construcciones de datos paralelos. Ian Buck luego se unió a NVIDIA y dirigió el lanzamiento de CUDA en 2006, la primera solución del mundo para informática general en GPU [33].

Impulsada por la insaciable demanda del mercado para gráficos en 3D de alta definición en tiempo real, la Unidad procesadora gráfica programable o GPU se ha convertido en un procesador altamente paralelo, multiproceso y multi-core con una gran potencia de cálculo y un ancho de banda de memoria muy alto, como se ilustra en la Figura 14 y la Figura 15 [36].

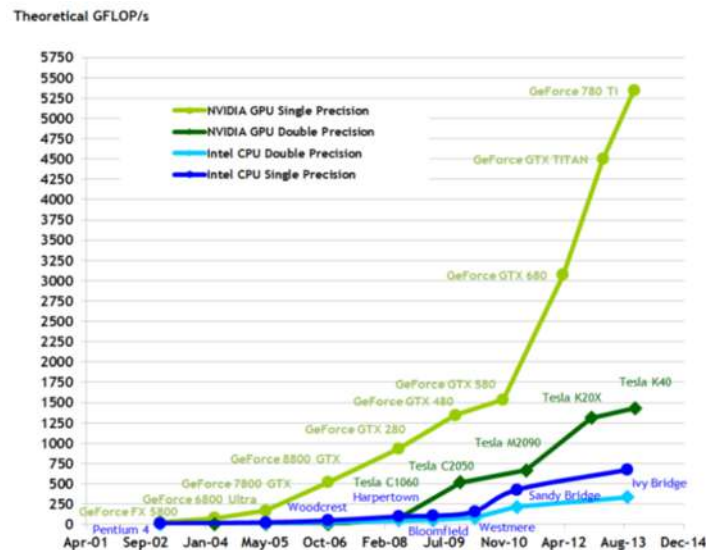


Figura 14. Operaciones de punto flotante por segundo para CPU y GPU [36].



Figura 15. Ancho de banda para memoria de CPU y GPU [36].

La razón detrás de la discrepancia en la capacidad de punto flotante entre la CPU y la GPU es que la GPU está especializada en computación altamente paralela y además está diseñada para que más transistores se dediquen al procesamiento de datos en lugar de caché de datos y control de flujo, como se ilustra esquemáticamente en la Figura 16 [36].



Figura 16. Comparación de asignación de transistores para procesamiento de datos en CPU y GPU [36].

Más específicamente, la GPU es especialmente adecuada para abordar problemas que pueden expresarse como cálculos paralelos de datos con alta intensidad aritmética. Debido a que se ejecuta el mismo programa para cada elemento de datos, existe un requisito menor para un sofisticado control de flujo, y debido a que se ejecuta en muchos elementos de datos y tiene una alta intensidad aritmética, la latencia de acceso a memoria puede ocultarse con cálculos en lugar de cachés de datos grandes. El procesamiento de datos paralelos mapea los elementos de datos a los hilos de procesamiento paralelos [36].

El advenimiento de las CPU multi núcleo y las GPU significa que los chips procesadores convencionales ahora son sistemas paralelos. Además, su paralelismo continúa escalando con la ley de Moore. El desafío es desarrollar software de aplicaciones que escale de manera transparente su paralelismo para aprovechar el creciente número de núcleos de procesadores. El modelo de programación en paralelo de CUDA está diseñado para superar este desafío, manteniendo una curva de aprendizaje baja para los programadores familiarizados con los lenguajes de programación estándar, como C [36].

Capítulo V. Casos de Estudio.

El caso de estudio de la turbina Francis-99, que consiste en un modelo de turbina Francis de alta carga es el objeto de estudio en este proyecto, sin embargo, se tomó la decisión de utilizar el caso de estudio de la turbina T-99 en la etapa inicial, ya que es un caso más sencillo.

Turbina T-99.

La recuperación de la presión en un tubo de aspiración, que es el difusor curvado corriente abajo del rodete en una turbina de agua, tiene lugar en una geometría muy complicada. Cuando se restauran antiguas centrales hidroeléctricas, existe la posibilidad de mejorar la recuperación de presión en el tubo de aspiración modificando la geometría del difusor. Para utilizar este potencial, se necesitan herramientas validadas de CFD [37].

El flujo aguas abajo del corredor en una turbina de agua, que es el flujo de entrada al tubo de aspiración, es complicado con un flujo medio en remolino, un componente de velocidad periódica y turbulencia. El flujo en el tubo de aspiración en sí está compuesto por varios fenómenos de flujo o regímenes de flujo coexistentes: capas límite en gradiente de presión positiva, curvatura de flujo, rotación de flujo y posiblemente separación de flujo (2D o 3-D), creando así un verdadero desafío de la aplicación [37].

El caso de la Turbina T-99 consiste en un estudio experimental del flujo en un tubo de aspiración con tacón afilado para una turbina Kaplan. Las mediciones se realizaron en la instalación de pruebas de turbinas en Vattenfall Utveckling AB en Älvkarleby, Suecia. El tubo de aspiración es un difusor curvado aguas abajo del rotor en una turbina de agua, y el propósito del tubo de aspiración es recuperar la mayor cantidad posible de la energía cinética del flujo después del rotor. La pérdida de eficiencia en el tubo de aspiración es particularmente importante para las estaciones de energía Kaplan de baja altura y alto flujo. El modelo completo consiste en una copia a escala 1:11 de la estación de energía Hölleforsen que incluye el rotor tipo Kaplan, que tiene un diámetro de 0.5 m. La Figura 17 muestra el tubo de aspiración [37].

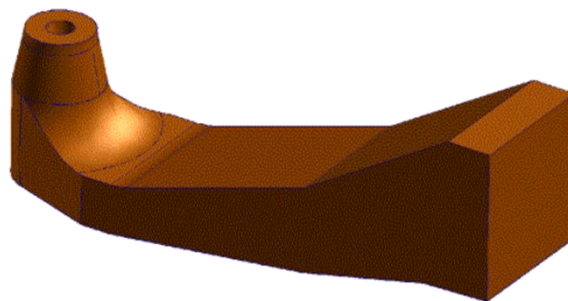


Figura 17. Esquemático del tubo de aspiración [37].

El caso de prueba presentado aquí está diseñado para evaluar la capacidad de los códigos CFD para predecir el aumento de presión y el campo de flujo para tales problemas [37].

Turbina Francis-99.

En el caso de estudio la Turbina Francis-99, se realizaron experimentos en una turbina tipo Francis modelo Tokke con escala 1:5.1, considerando un sistema hidráulico de ciclo abierto. La carga de operación fue de aproximadamente 12 m y la descarga fue de $0.2 \text{ m}^3/\text{s}$ en el BEP (*Best Efficiency Point*). Se utilizó un sistema de registro para obtener datos relacionados con el diagrama de colina y otro para adquirir valores de presión de los sensores de presión adicionales.

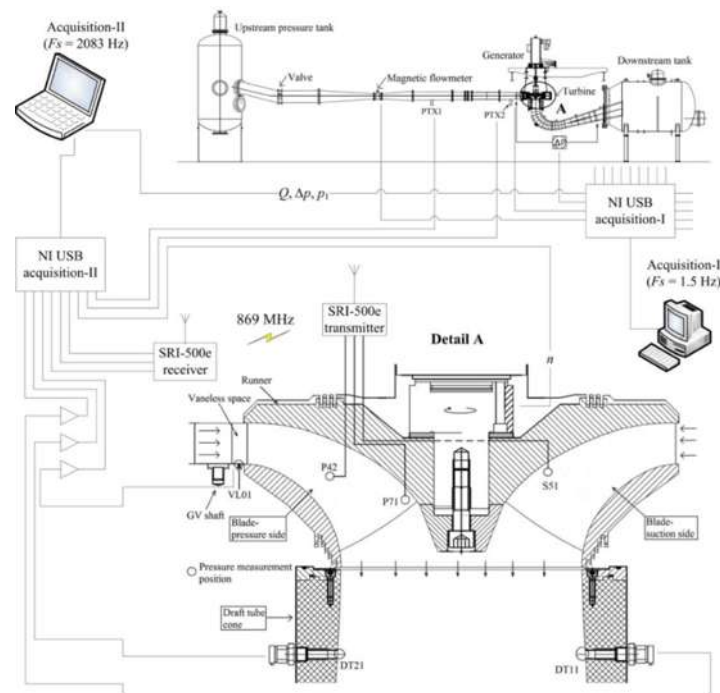


Figura 18. Localización de sensores en el banco de pruebas de la turbina Francis [12].

Se instalaron seis sensores de presión en la turbina, un sensor (VL01) en el espacio sin alabes, tres sensores (P42, S51 y P71) en el rotor y dos sensores (DT11 y DT21) en el tubo de aspiración. Las ubicaciones de los sensores y el sistema de registro se muestran en la Figura 18. Los valores de presión se adquirieron a la frecuencia de muestreo de 2,083 Hz. Se utilizó un sistema de telemetría inalámbrico para obtener datos de los sensores del rotor [12].

La turbina prototipo correspondiente (carga = 377 m, potencia = 110 MW y diámetro de rodete = 1.779 m) está en operación en la central hidroeléctrica Tokke, Noruega. El modelo y el prototipo están integrados con 14 alabes estacionarios unidos dentro de la carcasa de la voluta, 28 alabes guía, un rotor con 15 divisores y 15 alabes de longitud completa, y un tubo

de aspiración tipo codo. La geometría del modelo comienza en la entrada de la carcasa en espiral y termina en la salida del tubo de aspiración [38] (ver Figura 19).

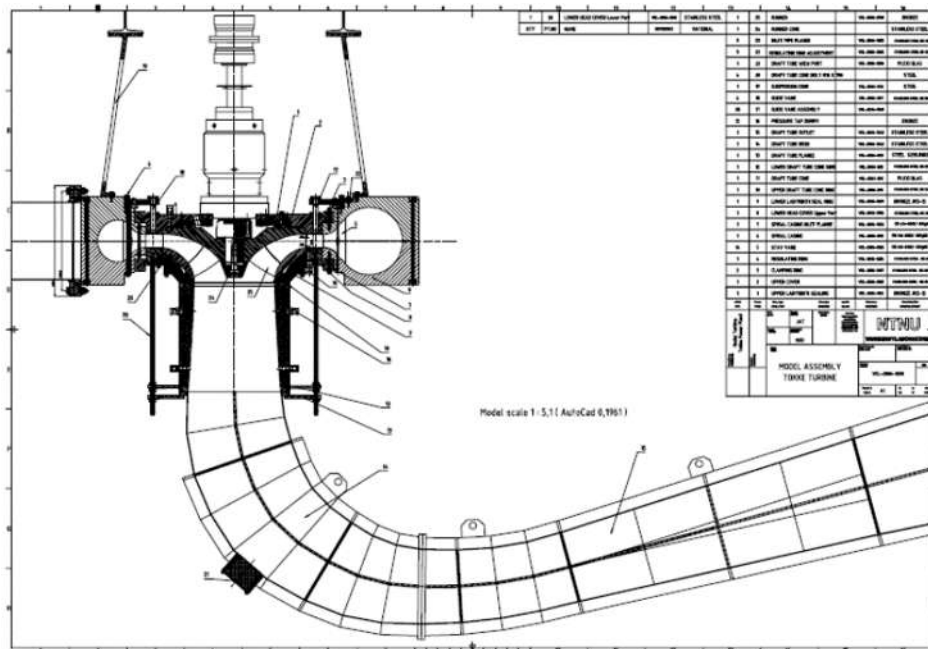


Figura 19. Turbina Tokke [12].

Antes de las mediciones de presión y velocidad, se llevó a cabo un análisis detallado del rendimiento de la turbina. Se seleccionaron 10 ángulos diferentes para los alabes guía y se seleccionaron 15 valores de velocidad diferentes para cada ángulo. Así, se realizaron mediciones de eficiencia en un total de 150 puntos. La Tabla 11 y la Figura 20 muestran los parámetros observados en estos puntos de operación.

Tabla 11. Parámetros observados a las condiciones de operación PL, BEP and HL [12].

Parameter	PL	BEP	HL	Remark
Turbine inlet pressure absolute (kPa)	219.93	216.54	210.01	Pressure was measured just before the casing inlet using ring type manifold.
Differential pressure across the turbine (kPa)	120.39	114.98	114.03	Turbine outlet pressure = Inlet pressure - Differential pressure.
Water density (kg m ⁻³)	999.23	999.19	999.20	Water density was computed using measured temperature value.
Net head (m)	12.29	11.91	11.24	Head was calculated using equations given in IEC 60193.
Discharge (m ³ s ⁻¹)	0.071	0.203	0.221	Discharge was measured using a magnetic flow meter.
Generator input torque (Nm)	137.52	619.56	597.99	Torque was measured in-between the thrust block just and generator.
Friction torque (Nm)	6.54	8.85	7.63	Torque developed by runner = Generator input torque + Friction torque.
Runner speed (rpm)	406.2	335.4	369.6	
Hydraulic efficiency (%)	71.69	92.61	90.66	Hydraulic efficiency is computed using equations given in IEC 60193
Guide vane angle (degree)	3.91	9.84	12.44	14 degree is the maximum opening position (full load).

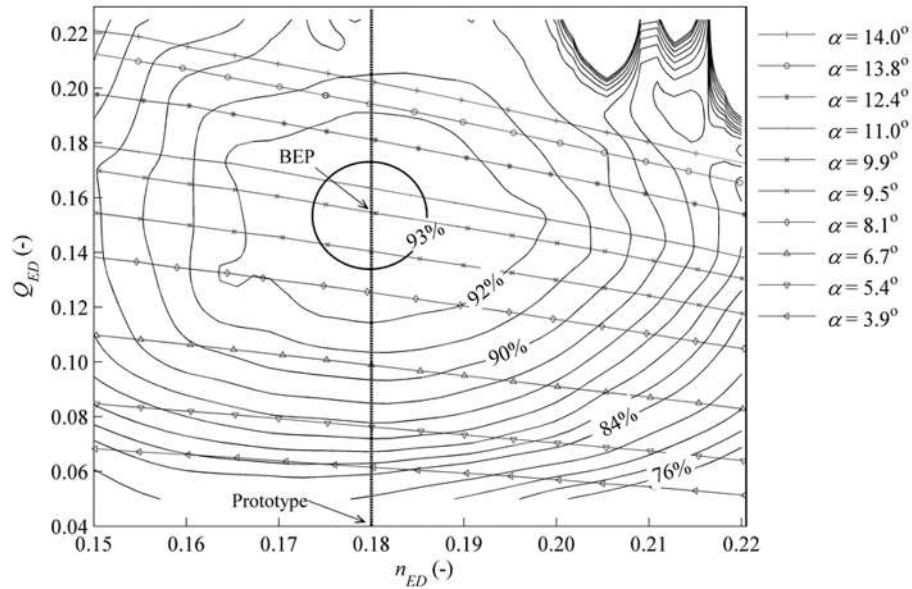


Figura 20. Diagrama de colina para la eficiencia [12] [38].

Las secciones transversales de medición de la presión de entrada y salida de la turbina son de 0.0872 m^2 y 0.2360 m^2 , respectivamente. La constante gravitacional en el laboratorio es de 9.821 m/s^2 . El diámetro de referencia del modelo Francis es 0.349 m . La viscosidad cinemática es $9.57 \times 10^{-7} \text{ m}^2/\text{s}$. Los números de Reynolds en BEP son 1.8×10^6 [17].

Capítulo VI. Resultados.

Aceleración de Calculo.

El flujo turbulento desarrollado en un tubo de aspiración de una turbina hidráulica tiene una estructura compleja, y así, para lograr la exactitud necesaria en una predicción numérica, el dominio del problema debe estar suficientemente discretizado. Sin embargo, a medida que aumenta el número de puntos de la malla, también aumenta el tiempo computacional requerido para obtener la simulación. Sin embargo, estudiar el flujo en detalle exige un esfuerzo computacional significativo para acelerar el tiempo de solución [39], ya que se requieren modelos de turbulencia robustos, nuevos esquemas de discretización, mallas densas y pasos de tiempo más cortos. Adicional a la solución numérica, el uso eficiente del hardware continúa siendo un factor importante detrás de las decisiones de ingeniería para expandir el CFD como una tecnología para apoyar el diseño del producto. En el cálculo de alto rendimiento (HPC), el paralelismo se considera el futuro de la informática, ya que los esfuerzos en el desarrollo de microprocesadores se han concentrado en agregar núcleos en lugar de aumentar el rendimiento de un solo subproceso. En consecuencia, los cálculos de CFD se han realizado distribuyendo el proceso en muchos nodos en un entorno paralelo distribuido o utilizando CPU multi núcleo en un solo nodo. Las características de diseño de la GPU dan como resultado un poder de cómputo y un ancho de banda de memoria que excede el de las CPU más rápidas de varios núcleos en casi un orden de magnitud. Sin embargo, problemas que no tienen una solución paralela no se benefician del paralelismo en absoluto y se resuelven mejor usando un CPU. Por otro lado, los problemas que pueden dividirse en muchos sub problemas independientes. Tales problemas son masivamente paralelos y se resuelven mejor usando una GPU. Por lo tanto, este tremendo potencial solo puede explotarse en cálculos que se pueden paralelizar en al menos decenas de miles de subprocesos, con un patrón de acceso a la memoria muy regular. Afortunadamente, los algoritmos CFD contienen muchos pasos que satisfacen estas condiciones, por lo que vale la pena probar su aceleración en GPU [40].

Para cuantificar la aceleración de cálculo se tomó como caso de estudio de la turbina T99. Esta geometría ha sido utilizada previamente en dos talleres de la Comunidad Europea de Investigación sobre Turbulencia de Flujo y Combustión (ERCOFTAC) [41] [42] [43]. Las pruebas se desarrollaron inicialmente utilizando dos *workstations* distintas para un tamaño de malla estructurada de un millón de celdas hexaédricas.

La Tabla 12 muestra las características de las *workstations* WSPAC y WSMOL utilizadas en los análisis.

Tabla 12. Características de las Workstations WSPAC y WSGAL

Workstation WSPAC	Workstation WSGAL
<i>CPU</i>	
2 x Intel Xeon E5504, 2.0 GHz	2 x Intel Xeon L5639, 2.13 GHz
4 cores per processor / 4 threads	6 cores per processor / 12 threads
14 GB Memory DDR3, 1060 Hz	24 GB Memory DDR3, 1060 Hz
<i>GPU</i>	
4 x Nvidia TESLA C1060	2 x Nvidia TESLA M2090
240 CUDA cores, 1.296 GHz	512 CUDA cores, 1.3 GHz
4 GB Memory GDDR3, 800 MHz	6 GB Memory GDDR5, 1.85 GHz

Los programas/software utilizados fueron las siguientes:

- OpenFOAM v1706, que utiliza la librería de interfaz de paso de mensajes (MPI) para proporcionar una funcionalidad de paralelización multiprocesador al CPU, que se adapta bien a sistemas homogéneos pero que no utiliza al máximo el rendimiento potencial por nodo en sistemas híbridos.
- RapidCFD, que es una implementación de código abierto de OpenFOAM capaz de ejecutar simulaciones en GPUs de Nvidia, mediante paralelización de la GPU con la librería de arquitectura de dispositivo unificado de cálculo (CUDA), lo que conduce a mejoras de rendimiento [28].

El dominio computacional se resolvió con doble precisión de punto flotante utilizando la aplicación *simpleFoam* en estado estacionario, lo cual implica que los términos temporales no se resuelven; además se emplearon esquemas de discretización de segundo orden. Debido a que los datos experimentales para las condiciones de frontera solo proporcionan información de k y ε [41] [42] [43], el modelo de turbulencia $k-\varepsilon$ *standard* se implementó en las simulaciones. El solucionador lineal utilizado para la ecuación de presión p fue PCG y para la velocidad U , k y ε se usó PBiCG. Para el pre acondicionamiento de las matrices de utilizo el pre acondicionador *diagonal*. La tolerancia del solucionador se estableció en 10^{-12} para todas las ecuaciones. Para acoplar el sistema de ecuaciones $p-U$, se utilizó el algoritmo del método semi implícito de ecuaciones vinculadas a la presión (SIMPLE).

El dominio computacional se descompuso usando el método de descomposición *scotch*. Al usar OpenFOAM, el dominio se descompuso proporcionalmente al número de núcleos/hilos del CPU, sin embargo, en el caso de RapidCFD, el dominio no se descompuso de acuerdo con la cantidad de núcleos en la GPU, sino con el número de GPU utilizados. El procedimiento descrito se resume en la Tabla 13.

Tabla 13. Descomposición de dominio

Workstation WSPAC			Workstation WSGAL		
Configuration	Cores	Domains	Configuration	Cores	Domains
1 cpu	1	1	1 cpu	1	1
2 cpu	2	2	2 cpu	2	2
4 cpu	4	4	4 cpu	4	4
6 cpu	6	6	6 cpu	6	6
8 cpu	8	8	8 cpu	8	8
1 cpu + 1 gpu	240	1	10 cpu	10	10
2 cpu + 2 gpu	480	2	12 cpu	12	12
3 cpu + 3 gpu	720	3	24 cpu	24	24
4 cpu + 4 gpu	960	4	1 cpu + 1 gpu	512	1
			2 cpu + 2 gpu	1024	2

En todas las simulaciones se utilizó la misma configuración numérica, donde el modelo computacional alcanzó el criterio de convergencia de 10^{-03} para los residuales de las ecuaciones de momento y continuidad después de 557 iteraciones al usar OpenFOAM v1706+ y 599 iteraciones para RapidCFD. Este comportamiento se puede observar claramente en la Figura 21 y Figura 22.

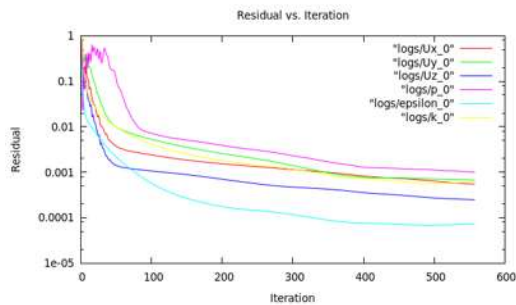


Figura 21. Residuales de ecuaciones de momento y continuidad en OpenFOAM v1706 utilizando CPU.

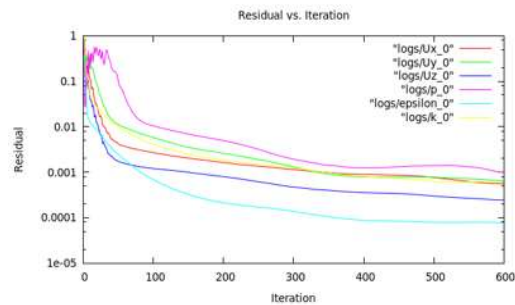


Figura 22. Residuales de ecuaciones de momento y continuidad en RapidCFD utilizando GPU.

La diferencia en el número de iteraciones necesarias para llegar a la solución podría atribuirse a que RapidCFD se desarrolló en base a OpenFOAM 2.3.1 y se incluyen muchas mejoras en las versiones de software más recientes.

El *execution time* mide el tiempo durante el cual un procesador está trabajando activamente en una determinada tarea y el *wall clock time* es el tiempo transcurrido desde el inicio del proceso hasta el final. Si el *wall clock time* es menor que el *execution time*, entonces un programa se ejecuta perfectamente en paralelo; por el contrario, si este es mayor, el sistema está consumiendo tiempo en acceso de datos en memoria, red u otros dispositivos entre cada iteración. Los resultados del tiempo de cómputo se presentan en la Figura 23 y Figura 24

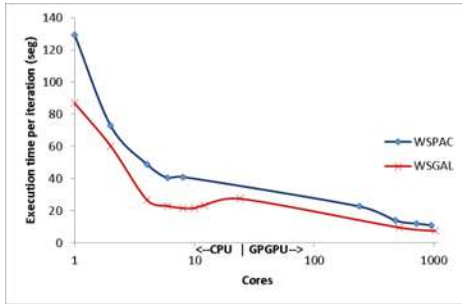


Figura 23. Execution time del benchmark de la turbina T99.

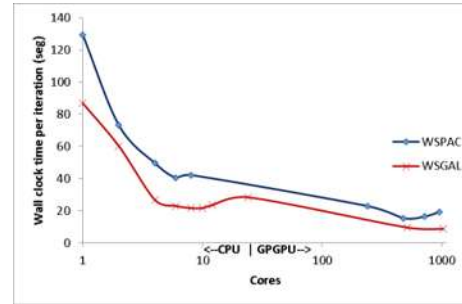


Figura 24. Wall clock time del benchmark de la turbina T99.

El *speedup* mide cuánto más rápido corre un algoritmo en paralelo con respecto al mejor algoritmo secuencial. La aceleración se comparó mediante la Ecuación 1 [44]. Para un problema de tamaño n , la expresión para el speedup es:

$$S_p = \frac{T_s(n, 1)}{T_p(n, N)} \quad \text{Ecuación 1}$$

Donde $T_s(n, 1)$ es el tiempo del mejor algoritmo secuencial y $T_p(n, N)$ es el tiempo del algoritmo en paralelo con los procesadores N , ambos resolviendo el mismo problema. La Figura 25 muestra la aceleración dada por la ecuación anterior usando el tiempo de ejecución por iteración obtenido a partir de las simulaciones CFD. Con ambas *worksations*, las GPUs fueron significativamente más rápidas (hasta alrededor de 9.32 veces). Esto demuestra que las GPUs son dispositivos masivamente paralelos que necesitan decenas de miles de subprocesos para una ejecución eficiente del programa. En [45] se reporta hasta 4.29x de *speedup*, mientras que en [40] reportan hasta 6x de *speedup*. En muchos casos, obtener una aceleración de 5 o 10 es más que adecuado, especialmente si el esfuerzo involucrado en el desarrollo del programa paralelo no fue muy grande [46].

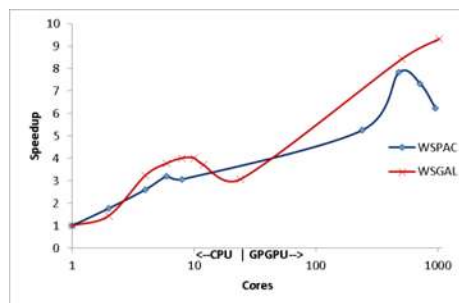


Figura 25. Speedup del modelo computacional

La ley de Amdahl [44] establece que, para un problema de tamaño fijo, el *speedup* esperado está dada por:

$$S_p = \frac{1}{(1 - c) + \frac{c}{N}}$$

Ecuación 2

Donde c es la fracción paralela del programa, $(1 - c)$ la fracción secuencial y N la cantidad de procesadores. Si se tiene una gran cantidad de procesadores, $N \approx \infty$, entonces el *speedup* máximo está limitada por la parte secuencial del algoritmo $(1 - c)$. La Figura 26 compara la fracción de paralelismo c lograda por el hardware en ambas workstations. El mayor paralelismo alcanzado fue 0.92 y depende de la cantidad de núcleos utilizados (CPU y GPU).

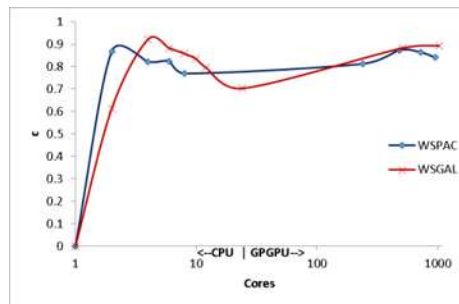


Figura 26. Fracción de paralelismo del modelo computacional

La paralelización determinada por el número de núcleos de la GPU desempeña un papel clave en las implementaciones multi GPU. Como se muestra en la Figura 25 y Figura 26 cuando se usan alrededor de 500 núcleos, el *speedup* y el paralelismo son similares en ambas *workstations*, sin embargo, cuanto más se descompone el dominio, el *speedup* y paralelismo caen en WSPAC. Sin embargo, cuando se usan cerca de 1000 núcleos, el *speedup* en WSPAC es 6.32 mientras que en WSGAL continúa creciendo hasta 9.32.

Los análisis de tiempo de cálculo y *speedup* anteriores se realizaron con GPUs de primera y segunda generación de la familia Tesla de NVIDIA (C1060 y M2090), que a pesar de soportar cálculos con doble precisión, el tamaño de memoria RAM y número de núcleos es limitado con respecto a modelos posteriores.

Debido a estas características y la necesidad de contar con equipos más potentes, fue necesario el uso de una tercera *workstation* denominada WSMOL, cuyas características se detallan en la Tabla 14.

Tabla 14. Características de la Workstation WSMOL

Workstation WSMOL	
<i>CPU</i>	
2 x Intel Xeon E5-2640 v2, 2.0 GHz	
8 cores per processor / 16 threads	
64 GB Memory DDR3, 1600 MHz	
<i>GPU</i>	
4 x Nvidia Tesla K40	
2880 CUDA cores, 745 MHz	
12 GB Memory GDDR5, 3GHz	

Para complementar los análisis previos y evaluar el desempeño de las GPUs K40, se analizó el tiempo de cálculo y el *speedup* para la solución del caso de estudio de la Turbina T99 utilizando únicamente RapidCFD, considerando el mismo dominio computacional de un millón de celdas utilizado en análisis previos, pero ahora utilizando la *workstation* WSMOL.

La Figura 27 y Figura 28 muestran el *execution time* y *wall clock time* respectivamente. Se puede observar en ambas graficas que las GPUs K40 en la WSMOL son más rápidas que las C1060 y M2090.

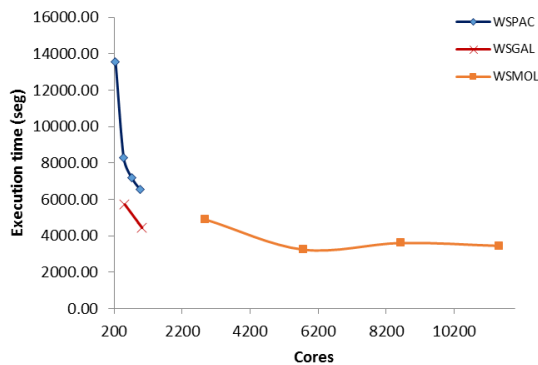


Figura 27. Execution time del benchmark de la turbina T99 empleando diferentes modelos de GPU (WSPAC/C1060, WSGAL/M2090, WSMOL/K40).

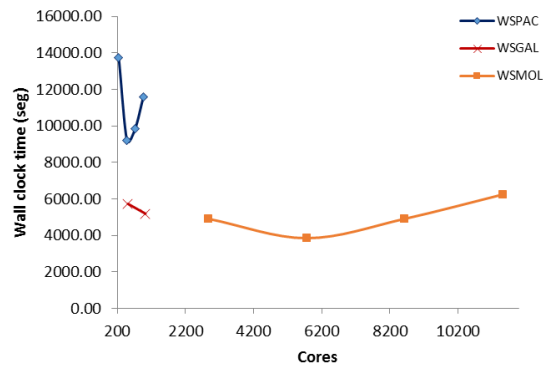


Figura 28. Wall clock time del benchmark de la turbina T99 empleando diferentes modelos de GPU (WSPAC/C1060, WSGAL/M2090, WSMOL/K40).

Al incrementar el número de GPUs y en consecuencia el número de núcleos para solucionar el mismo dominio computacional, el *wall clock time* comienza a incrementarse, mientras que el *execution time* sigue disminuyendo. Esto es debido principalmente al intercambio de información que existe entre las GPUs. A medida que se paraleliza el mismo tamaño de dominio computacional, el tamaño de problema asignado a cada GPU disminuye, y con ello la ventaja de las GPUs de disponer de canales de memoria asignados a cada núcleo para acelerar el cálculo no se aprovecha, ya que ahora la información tiene que transmitirse a través del bus de comunicación en la tarjeta madre de la estación de trabajo. La Figura 29

muestra lo anteriormente descrito mediante una comparación entre *execution time* y *wall clock time*.

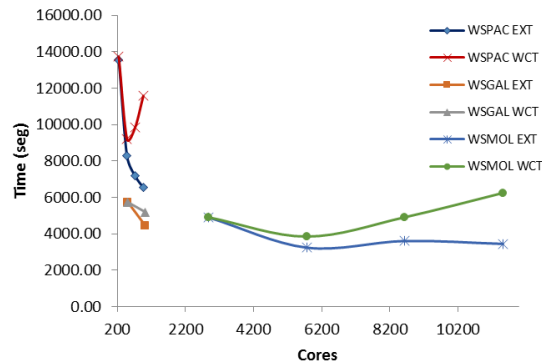


Figura 29. Wall clock time vs ejecución time del benchmark de la turbina T99 empleando diferentes modelos de GPU (WSPAC/C1060, WSGAL/M2090, WSMOL/K40).

El intercambio de información que se asemeja a un “cuello de botella”, es la razón de que el tiempo de cálculo para la solución de un dominio computacional no se reduzca linealmente al emplear más núcleos en la solución. Esto influye directamente en el *speedup* como se muestra en la Figura 30. Se puede observar que el *speedup* aumenta al aumentar de una GPU a dos, sin embargo a partir de este punto el *speedup* comienza a descender, llegando incluso a ser más lento el proceso de cálculo que usando una sola GPU.

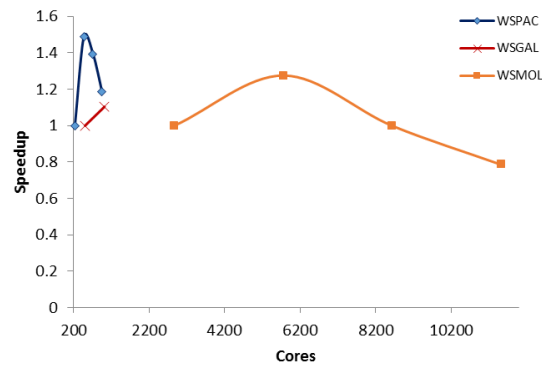


Figura 30. Speedup por workstation del modelo computacional de la turbina T99 empleando diferentes modelos de GPU (WSPAC/C1060, WSGAL/M2090, WSMOL/K40).

También se puede observar que el *speedup* más alto lo obtienen las GPUs C1060 (1.5 aproximadamente) a pesar de ser las más limitadas computacionalmente. Este resultado es debido a la forma en que se calcula y presentan los datos. En la Figura 30 el *speedup* se calculó utilizando la Ecuación 1. En los tres casos el cociente se realizó entre los mismos

datos de cada modelo de GPU, es decir $T_s(n, 1)$ de la GPU C1060 se dividió entre $T(n, N)$ de 1, 2, 3 y 4 GPUs C1060.

La Figura 31 muestra el *speedup* utilizando la misma Ecuación 1, sin embargo en este caso $T_s(n, 1)$ de la GPU C1060 se dividió entre $T(n, N)$ de 1, 2, 3 y 4 GPUs C1060, M2090 y K40. Ya que el valor de $T_s(n, 1)$ de la GPU C1060 es el más grande cuando se emplea una sola GPU. De esta manera se puede observar que el mayor *speed up* se obtiene con las GPUs K40 (aproximadamente 3.5), lo cual está de acuerdo con los valores de *wall clock time* presentados previamente. Por lo que, cuando se compare equipo de cómputo en base al *speedup* se debe considerar y analizar la forma en que se calculó.

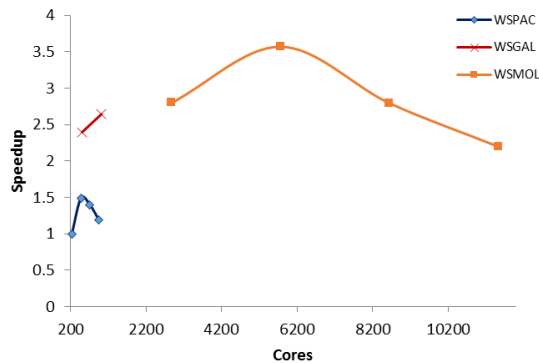


Figura 31. Speedup global del modelo computacional de la turbina T99 empleando diferentes modelos de GPU (WSPAC/C1060, WSGAL/M2090, WSMOL/K40).

Solucionadores Lineales y Velocidad de Calculo.

Para determinar el *set up* más adecuado que explote de manera más eficiente el hardware (GPUs NVIDIA C1060, M2060 y K40), se realizaron pruebas utilizando varios solucionadores lineales, pre acondicionadores y suavizadores, así como dos configuraciones para el parámetro *tolerance* entre *time step* (iteración) durante la solución del caso de la Turbina T99 utilizando un dominio computacional de un millón de celdas.

Para el análisis se empleó el solucionador lineal PCG/PBiCG en conjunto con los pre acondicionadores *diagonal* y AINV. También se utilizó el solucionador lineal GAMG, lo cual incluyó el uso de dos diferentes suavizadores; *Gauss-Seidel* y *Jacobi*. Además, se realizaron pruebas con dos valores del parámetro *tolerance*. El algoritmo de solución SIMPLE y el modelo de turbulencia *k-ε standard* se utilizaron nuevamente. Es importante recalcar que el esfuerzo principal se encamino a obtener el mejor *set up* empleando una sola GPU. A continuación, en la Tabla 15 se detalla el esquema de todas las pruebas realizadas.

Tabla 15. Parámetros de simulación de RapidCFD en Workstations usando GPUs.

algorithm	SIMPLE							
	PCG/PBiCG				GAMG			
solver	diagonal		AINV		GaussSeidel		Jacobi	
preconditioner/smoothen								
RAS model					k-ε standard			
tolerance	1x10 ⁻¹²	1x10 ⁻⁰⁶	1x10 ⁻¹²	1x10 ⁻⁰⁶	1x10 ⁻¹²	1x10 ⁻⁰⁶	1x10 ⁻¹²	1x10 ⁻⁰⁶
residuals					1x10 ⁻⁰³			

Se analizaron 3 parámetros: *wall clock* time del primer y último *time step* y *time per time step*. De la Figura 32 a la Figura 40 se observa que el solucionador PCG/PBiCG en combinación con el pre acondicionador AINV es más rápido que el resto cuando se utiliza un valor de *tolerance* de 1x10⁻¹². Por otro lado, cuando se utiliza un valor de *tolerance* de 1x10⁻⁰⁶ el solucionador GAMG en combinación con el suavizador *Jacobi* es más rápido que el resto. Estos resultados concuerdan para los diferentes modelos de GPUs utilizados (C1060, M2090 y M2090), con lo que se podría generalizar esta tendencia a modelos de GPU más recientes.

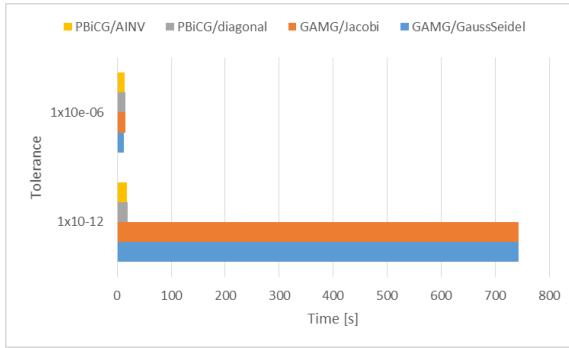


Figura 32. Wall clock time de primer time step (iteración) en WSPAC con RapidCFD y GPUs C1060.

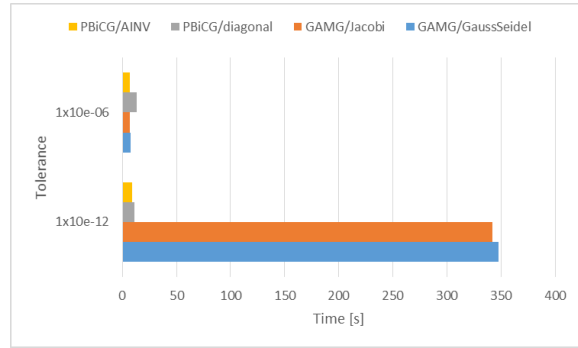


Figura 33. Wall clock time de primer time step (iteración) en WSGAL con RapidCFD y GPUs M2090.

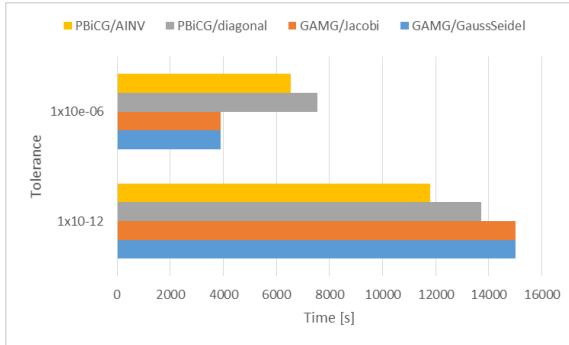


Figura 34. Wall clock time de ultimo time step (iteración) en WSPAC con RapidCFD y GPUs C1060.

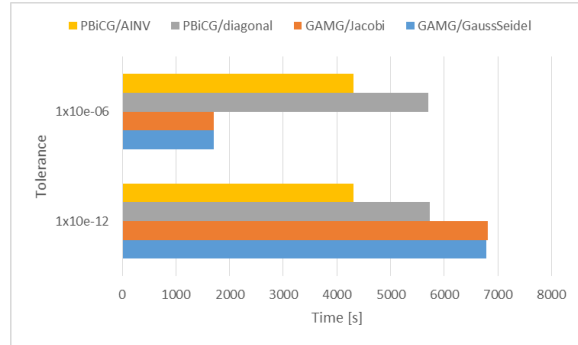


Figura 35. Wall clock time de ultimo time step (iteración) en WSGAL con RapidCFD y GPUs M2090.

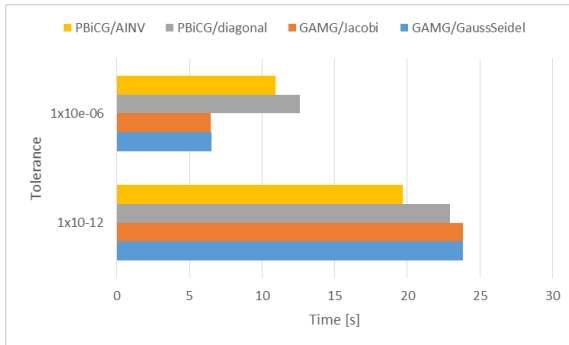


Figura 36. Tiempo por time step (iteración) en WSPAC con RapidCFD y GPUs C1060.

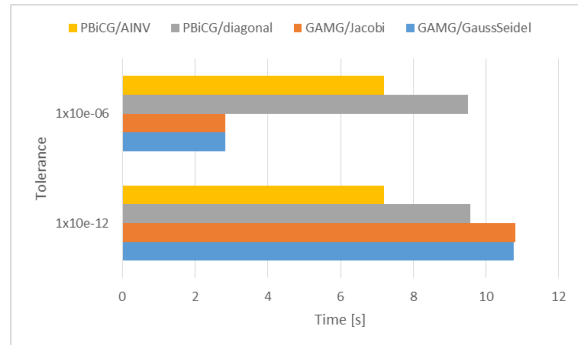


Figura 37. Tiempo por time step (iteración) en WSGAL con RapidCFD y GPUs M2090.

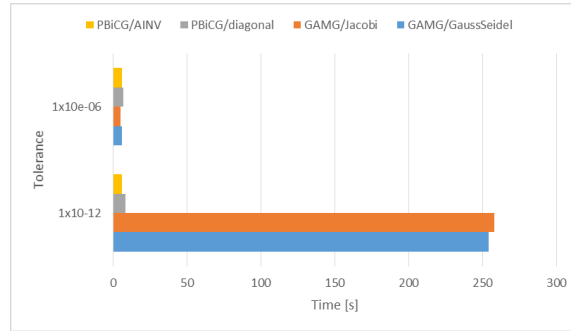


Figura 38. Wall clock time de primer time step (iteración) en WSMOL con RapidCFD y GPUs K40.

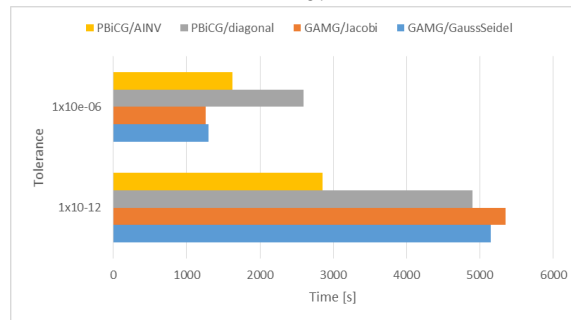


Figura 39. Wall clock time de ultimo time step (iteración) en WSMOL con RapidCFD y GPUs K40.

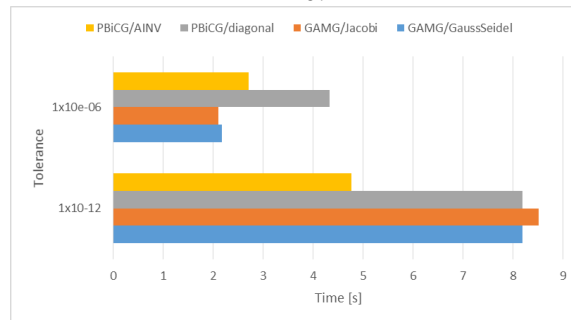


Figura 40. Tiempo por time step (iteración) en WSMOL con RapidCFD y GPUs K40.

Al hacer una comparación entre los resultados obtenidos de los diversos modelos de GPUs empleando 1, 2, 3 y 4 GPUs se observa el mismo comportamiento antes descrito y se confirma nuevamente que las GPUs NVIDIA Tesla K40 son más rápidas como anteriormente también se había demostrado y que el tipo de solucionador lineal no influye en su desempeño (ver Figura 41 a Figura 44).

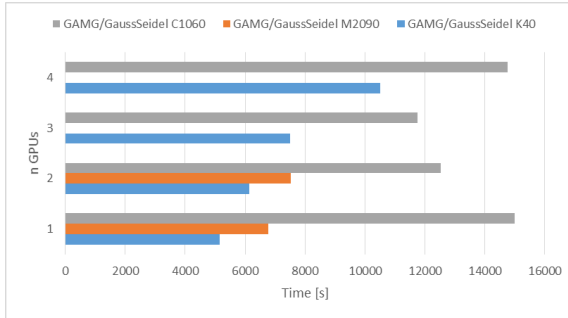


Figura 41. Wall clock time de ultimo time step (iteración) con RapidCFD empleando diferentes modelos y numero de GPU. Solucionador lineal GAMG, suavizador GaussSeidel.

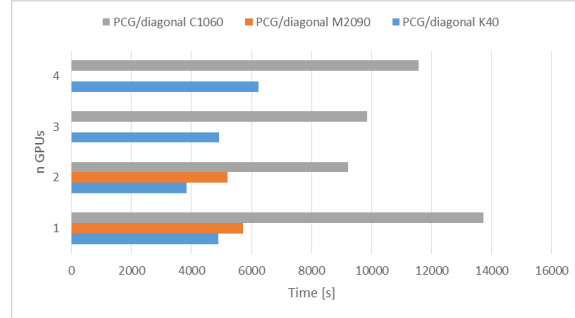


Figura 42. Wall clock time de ultimo time step (iteración) con RapidCFD empleando diferentes modelos y numero de GPU. Solucionador lineal PCG, pre acondicionador diagonal.

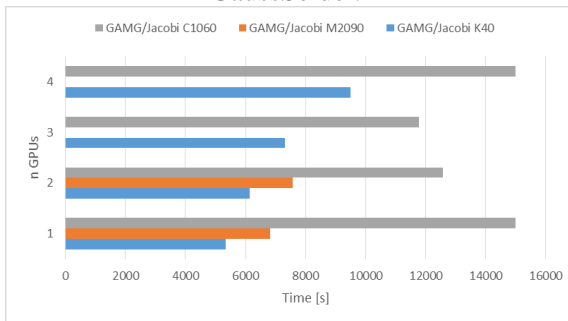


Figura 43. Wall clock time de ultimo time step (iteración) con RapidCFD empleando diferentes modelos y numero de GPU. Solucionador lineal GAMG, suavizador Jacobi.

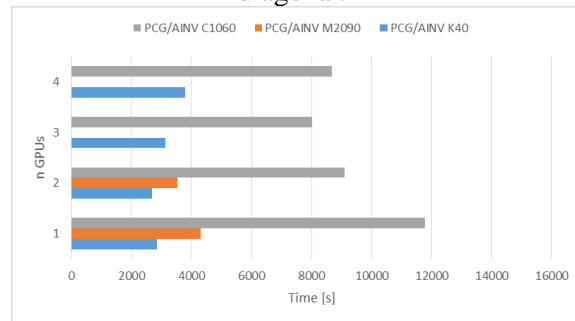


Figura 44. Wall clock time de ultimo time step (iteración) con RapidCFD empleando diferentes modelos y numero de GPU. Solucionador lineal PCG, pre acondicionador AINV.

Los resultados en cuanto al desempeño de las combinaciones software/hardware, mostraron que la mejor estrategia es utilizar el solucionador GAMG con valores de *tolerance* bajos, ya que de esta manera se logra una reducción en los tiempos de cálculo si afectar los resultados en CFD.

Solucionadores Lineales y Uso de Memoria.

En esta sección se describen las pruebas realizadas para encontrar el tamaño del problema que se podrá resolver, es decir el número máximo de celdas que se emplearan para encontrar la mejor resolución de flujo en CFD utilizando RapidCFD y GPUs Nvidia Tesla C1060, M2090 y K40. Hasta el momento se ha empleado un tamaño fijo de problema para los análisis, los cuales incluyen tanto *speedup* empleando GPUS, como análisis de velocidad de solución con distintos solucionadores lineales (PCG/PBiCG y GAMG).

Aun así, la memoria RAM (Random Access Memory) requerida por una GPU para la solución de un problema o dominio sin tener en cuenta el tipo de solucionador lineal y/o la arquitectura empleada es desconocida. De esta manera fue creado un grupo de dominios computacionales con un número de celdas que fue escalándose hasta alcanzar la memoria RAM máximo disponible en las GPUs.

Los dominios fueron creados empleados utilizando la parametrización de la Tabla 16.

Tabla 16 . Parámetros para generación de dominios computacionales.

Domain	1	2	3	4	5	6	7	8	9	10	11	12	13
Size	446820	739584	1154832	1412992	2051712	2594120	2815200	3179080	3433600	3799840	4215904	4624896	5096736
A	18	21	24	27	30	33	33	36	36	36	39	39	42
a	8	10	11	12	13	14	14	15	15	15	16	16	17
B	10	12	14	16	18	20	20	20	20	22	22	24	24
C	15	18	21	24	27	30	30	30	30	33	33	36	36
D	5	6	7	8	9	10	10	10	10	11	11	12	12
1	15	18	21	21	24	27	27	27	30	30	30	30	30
2	6	6	7	7	8	8	9	10	10	10	11	11	12
3	15	18	21	21	24	27	27	27	30	30	30	30	30
4	6	6	7	7	8	8	9	10	10	10	11	11	12
5	42	48	56	56	64	70	72	74	80	80	82	82	84
6	33	36	39	45	51	51	51	57	57	57	60	60	63
7	4	4	6	6	8	8	10	12	12	12	14	14	16
8	6	8	9	9	10	10	11	12	12	12	13	13	14
9	6	8	9	9	10	10	11	12	12	12	13	13	14
10	20	26	30	30	34	34	38	42	42	42	46	46	50
11	14	18	21	21	24	24	27	30	30	30	33	33	36
12	20	26	30	30	34	34	38	42	42	42	46	46	50
13	6	8	9	9	10	10	11	12	12	12	13	13	14
14	2	2	3	3	4	4	5	6	6	6	7	7	8
15	6	8	9	9	10	10	11	12	12	12	13	13	14
16	4	4	6	6	8	8	10	12	12	12	14	14	16

Los valores de las variables A y a corresponden a las líneas radiales de la entrada del tubo de aspiración de la turbina T-99, así como de los niveles inferiores como se muestra en la Figura 45.

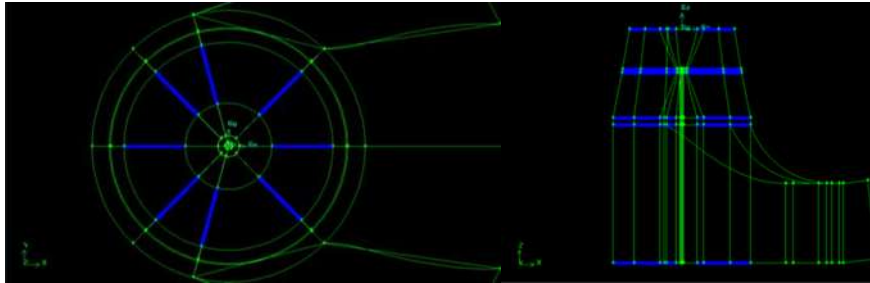


Figura 45. Discretización de los ejes de los grupos A y a .

Los valores de las variables B , C y D corresponden a los arcos que forman la sección cilíndrica del tubo difusor y líneas en la dirección Y de la sección rectangular como se muestra de la Figura 46 a Figura 48.

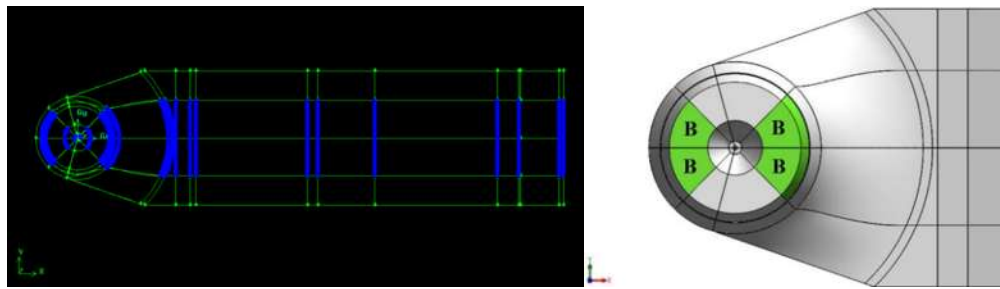


Figura 46. Discretización de los ejes del grupo B .

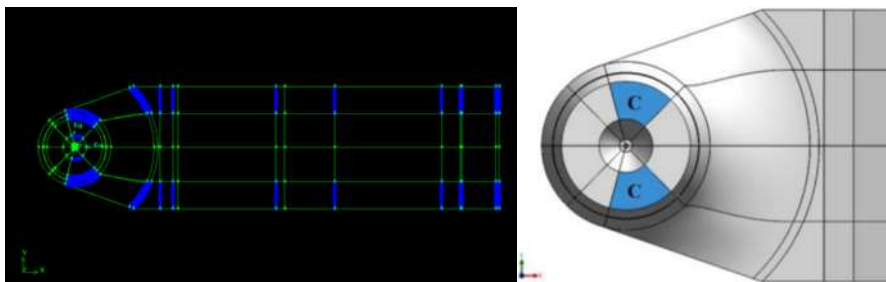


Figura 47. Discretización de los ejes del grupo C .

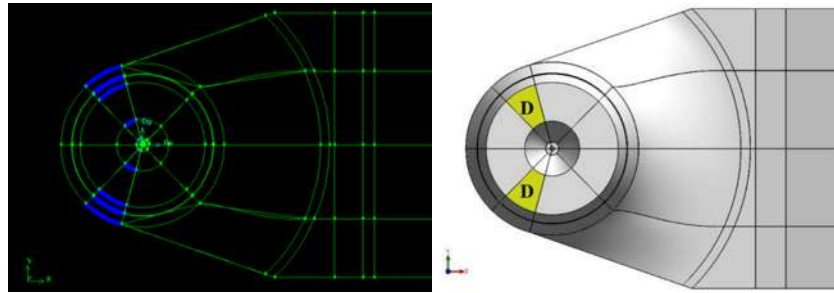


Figura 48. Discretización de los ejes del grupo D .

Los valores de la variable $I-5$ corresponden a los valores de las líneas en la dirección Z de los niveles superior a inferior de la sección cilíndrica del tubo de aspiración como se muestra en la Figura 49.

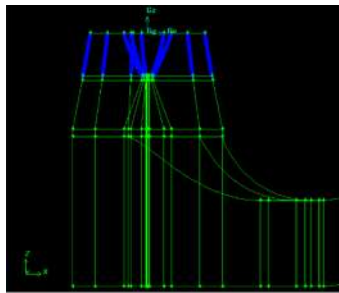


Figura 49. Discretización de los ejes del nivel $I-5$.

Los valores de las variables $6-16$ corresponden las líneas restantes en la dirección X de la sección rectangular del tubo difusor, con lo que queda establecida la discretización del dominio computacional (Ver Figura 50).

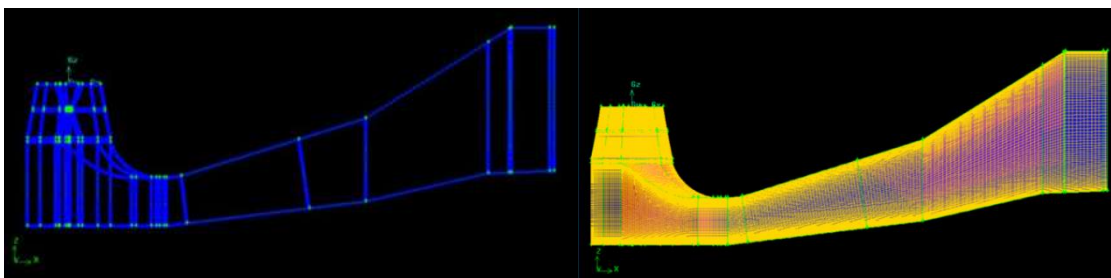


Figura 50. Discretización de los ejes del nivel $6-16$ (izquierda) y del dominio computacional completo (derecha).

Finalmente se insertó el valor de la capa límite, para lograr una resolución de flujo durante las simulaciones más aproximada al fenómeno físico. Teóricamente un valor más pequeño para la altura las primeras celdas de la capa límite favorece a la exactitud de los resultados y convergencia de la simulación, pero durante el proceso de discretización se debe cuidar la

calidad de la malla, por lo que existe un límite para el valor de la altura de la celda más pequeña respecto a la pared. Un valor extremadamente producirá dominios computacionales de mala calidad, de esta manera se tuvieron que utilizar valores diferentes para la capa limite en el mismo dominio computacional con el fin de asegurar la calidad de malla.

Con los dominios computacionales creados, se realizaron simulaciones en tres *workstations* (WSPAC, WSGAL y WSMOL) para medir la memoria RAM requerida para la solución de cada uno de ellos, empleando los solucionadores lineales PCG/PBiCG y GAMG en conjunto con el pre acondicionador AINV suavizador *Gauss-Seidel* respectivamente, ya que se ha comprobado que son las combinaciones más rápidas cuando se utiliza un valor de *tolerance* de 10^{-06} . El algoritmo de solución SIMPLE y el modelo de turbulencia *k-ε standard* se utilizaron nuevamente para alcanzar un valor de *residuals* de 10^{-03} como criterio de convergencia. Las simulaciones se realizaron en serie empleando Rapid CFD y GPUs, es decir cada dominio computacional fue resuelto empleando únicamente una GPU. La memoria RAM requerida para la solución de cada dominio computacional se muestra en la Figura 51.

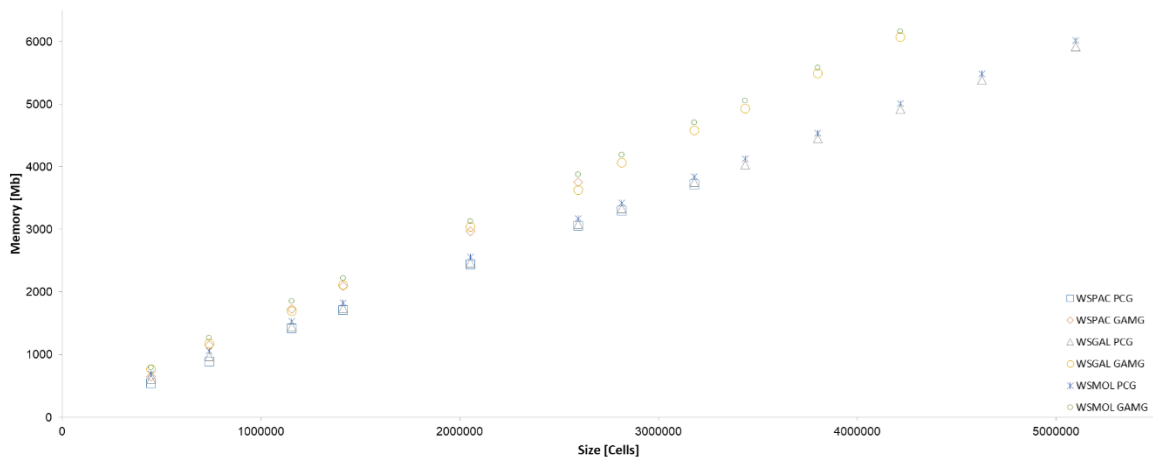


Figura 51. Memoria RAM requerida para la solución de dominios computacionales.

Se puede observar que el solucionador GAMG requiere mayor memoria RAM para la solución de cualquier dominio computacional, independientemente de la arquitectura empleada. En promedio el solucionador GAMG requiere 22% más de memoria RAM que el PCG/PBiCG. Por lo que, se puede concluir que la mayor velocidad en tiempo de cálculo empleando el solucionador GAMG implica una penalización relacionada con el uso de la memoria RAM.

Ya que, en la mayoría de ocasiones se pretende emplear toda la capacidad de cómputo disponible para realizar CFD, se realizaron también mediciones de la memoria RAM requerida para los mismos tamaños de dominio, pero ahora en paralelo en cada una de las en

tres *workstations* (WSPAC, WSGAL y WSMOL). Cada dominio computacional en WSPAC y WSMOL fue evaluado empleando 2, 3 y 4 GPUs, mientras que en WSGAL solo en 2 GPUs.

Usando regresión lineal simple se obtuvieron los valores del intercepto y la pendiente de las líneas que se ajusten de mejor manera a los datos obtenidos. En cada caso se obtuvieron dos ecuaciones, una para el solucionador PCG/PBiCG y otra para el GAMG, usando datos de WSPAC, WSGAL y WSMOL. La Figura 52 muestra la memoria RAM requerida para la solución en serie de los dominios computacionales con 1 GPU y las líneas que se ajustan a cada grupo de datos.

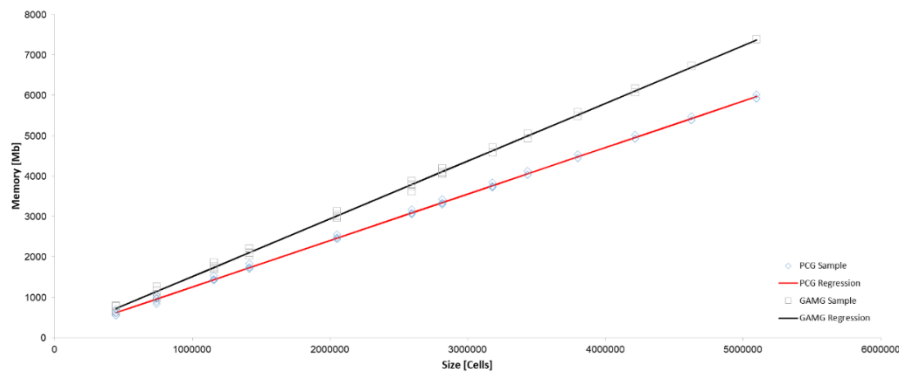


Figura 52. Memoria RAM requerida para la solución de dominios computacionales y regresión lineal para cada solucionador con 1 GPU. A) PCG/PBiCG, $M=83.79+0.001148S$ B) GAMG, $M=64.47+0.001424S$. Donde M es la memoria requerida y S el tamaño del dominio computacional.

De la Figura 53 a la Figura 55 se muestra la memoria RAM requerida para la solución en paralelo de los dominios computacionales con 2, 3 y 4 GPUs y las líneas que se ajustan a cada grupo de datos.

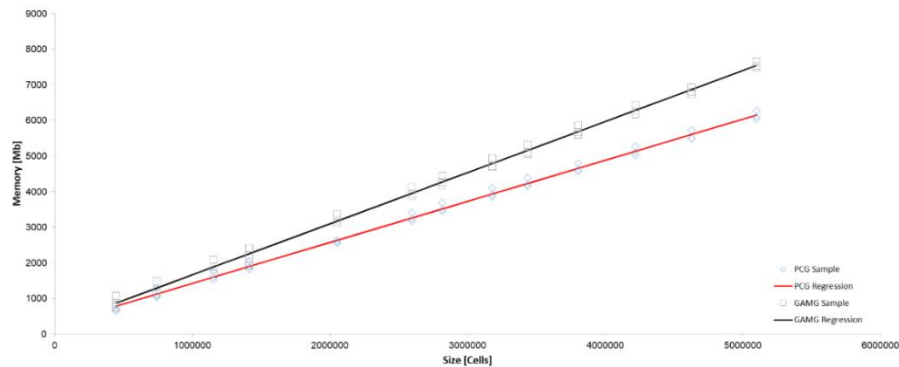


Figura 53. Memoria RAM requerida para la solución de dominios computacionales y regresión lineal para cada solucionador con 2 GPUs. A) PCG/PBiCG, $M=203.69+0.001151S$ B) GAMG, $M=115.35+0.001416S$. Donde M es la memoria requerida y S el tamaño del dominio computacional.

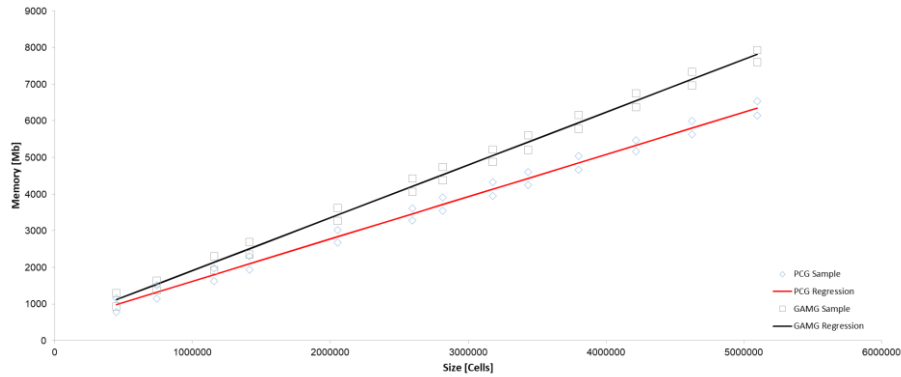


Figura 54. Memoria RAM requerida para la solución de dominios computacionales y regresión lineal para cada solucionador con 3 GPUs. A) PCG/PBiCG, $M=293.60+0.001152S$ B) GAMG, $M=303.80+0.001437S$. Donde M es la memoria requerida y S el tamaño del dominio computacional.

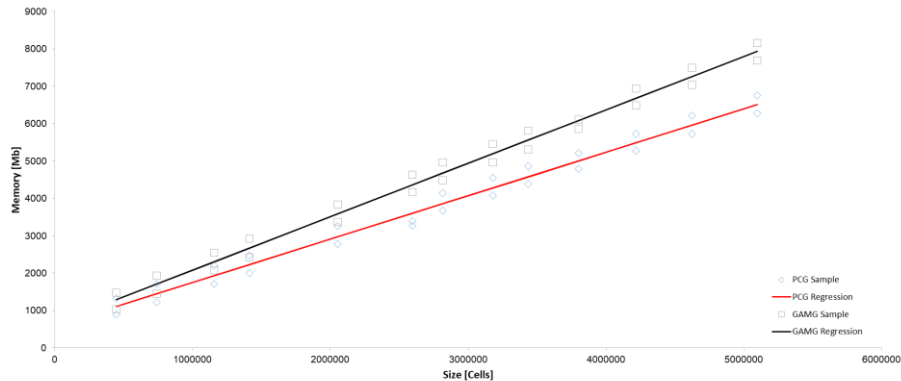


Figura 55. Memoria RAM requerida para la solución de dominios computacionales y regresión lineal para cada solucionador con 4 GPUs. A) PCG/PBiCG, $M=379.09+0.001159S$ B) GAMG, $M=407.87+0.001434S$. Donde M es la memoria requerida y S el tamaño del dominio computacional.

Las Figura 56 a Figura 57 muestran las mediciones de RAM requerida en serie y paralelo para la solución de los dominios computacionales. Se observa que tanto al emplear el solucionador lineal PCG/PBiCG como GAMG los datos obtenidos siguen la misma tendencia, es decir, al paralelizar un dominio computacional dado se requiere más memoria RAM que al resolverse en serie.

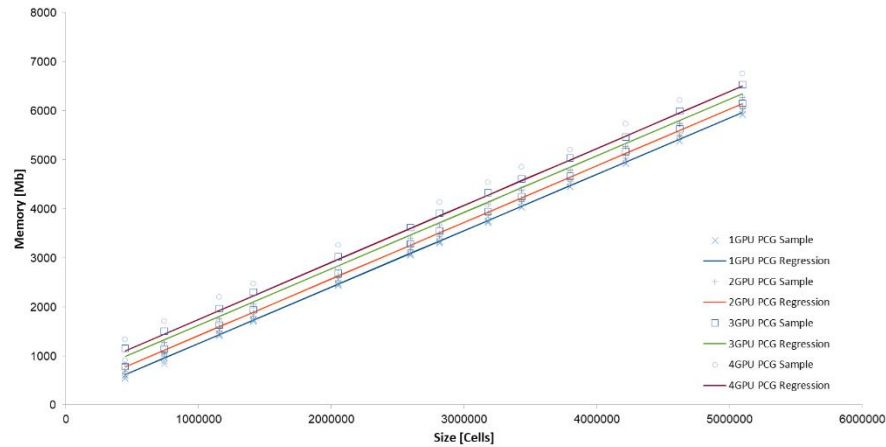


Figura 56. Memoria RAM requerida para la solución de dominios computacionales y regresión lineal utilizando el solucionador PCG/PBiCG con 1, 2, 3 y 4 GPUs.

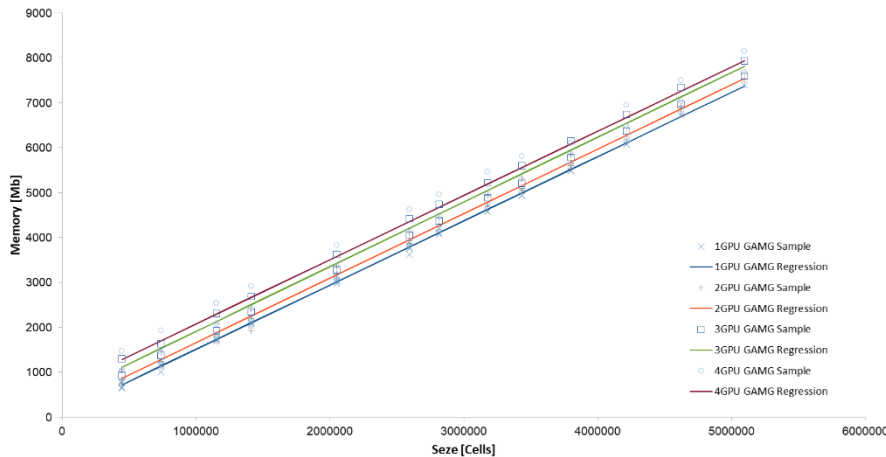


Figura 57. Memoria RAM requerida para la solución de dominios computacionales y regresión lineal utilizando el solucionador GAMG con 1, 2, 3 y 4 GPUs.

Al realizar un análisis de las 8 ecuaciones obtenidas en total (4 para PCG/PBiCG y 4 para GAMG), se observa claramente que para cada dominio computacional el intercepto va cambiando en función del número de GPUs empleadas en la solución, ya sea en serie con 1 GPU o paralelo con 2, 3 y 4 GPUs, ya que las pendientes de las ecuaciones de cada solucionador son prácticamente las mismas. Esta variación del intercepto en función del número de GPUs presenta un patrón, el cual consiste en que cada vez que se paraleliza el dominio computacional el intercepto aumenta en promedio 150 unidades respecto al caso previo. La Figura 58 muestra los datos obtenidos de las mediciones de memoria RAM requerida contra las líneas de ajuste propuestas basadas en las observaciones anteriores.

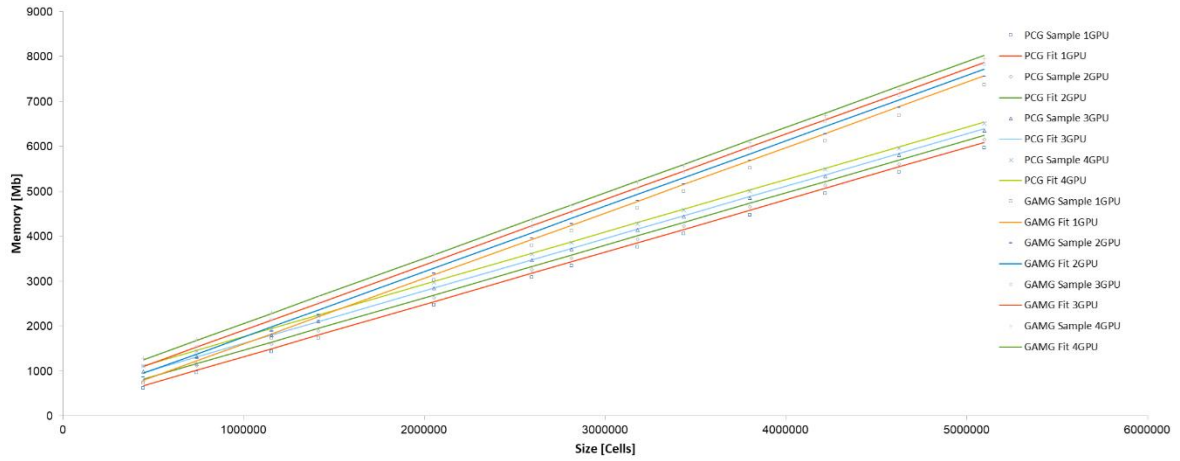


Figura 58. Memoria RAM requerida para la solución de dominios computacionales y propuesta de línea de ajuste para cada solucionador. A) PCG/PBiCG, $M=150N+0.001165S$ B) GAMG, $M=150N+0.001469S$. Donde M es la memoria requerida y S el tamaño del dominio computacional y N el número de GPUs empleadas que corresponde al número de dominios en paralelo.

Exactitud de los Cálculos.

Los resultados en secciones anteriores utilizando Rapid CFD y GPUs Nvidia Tesla C1060, M2090 y K40 para la solución del dominio de un millón de elementos correspondiente al caso de estudio de la Turbina T99, mostraron que el solucionador PCG/PBiCG en conjunto con el pre acondicionador AINV es más eficiente con valores de *tolerance* de 10^{-12} , sin embargo con valores de *tolerance* de 10^{-06} GAMG es más veloz. Al hacer el postprocesamiento cualitativo de los resultados de CFD, se observa que la diferencia entre usar distintos valores de *tolerance* es casi inexistente.

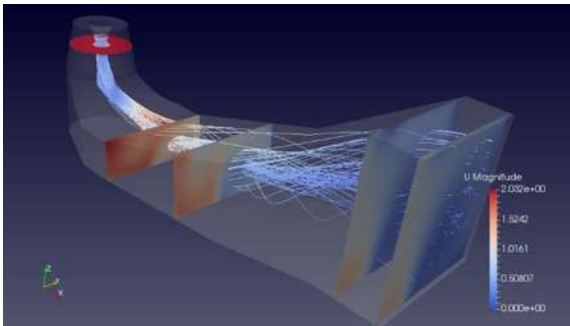


Figura 59. Contornos de velocidad en m/s con RapidCFD y solucionador GAMG con suavizador GasussSeidel.
Tolerance 10^{-12} .

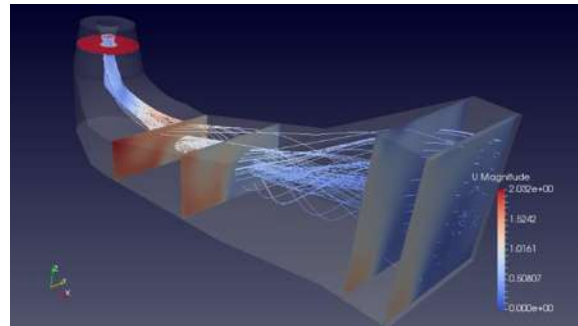


Figura 60. Contornos de velocidad en m/s con RapidCFD y solucionador GAMG con suavizador GasussSeidel.
Tolerance 10^{-06} .

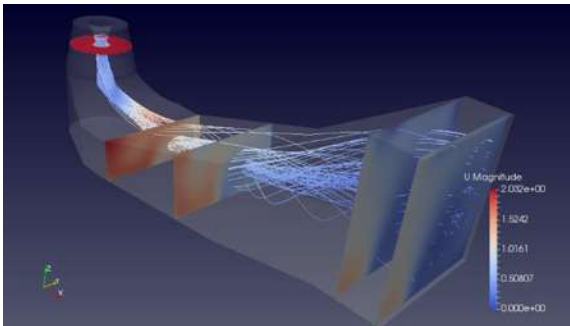


Figura 61. Contornos de velocidad en m/s con RapidCFD y solucionador GAMG con suavizador Jacobi.
Tolerance 10^{-12} .

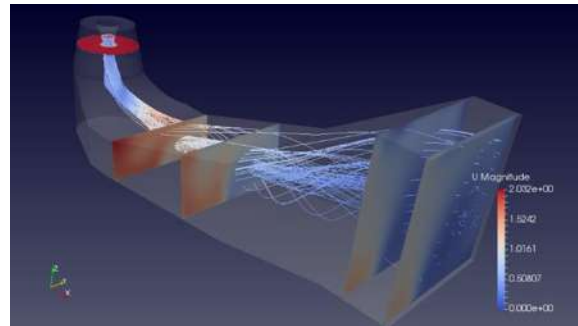


Figura 62. Contornos de velocidad en m/s con RapidCFD y solucionador GAMG con suavizador Jacobi.
Tolerance 10^{-06} .

Así mismo, la diferencia entre el solucionador GAMG y PCG/PBiCG con sus distintos suavizadores y pre acondicionadores es muy pequeña.

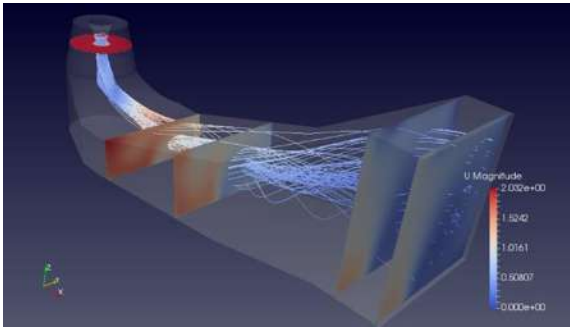


Figura 63. Contornos de velocidad en m/s con RapidCFD y solucionador PCG/PBiCG con preconditionador AINV.
Tolerance 10^{-12} .

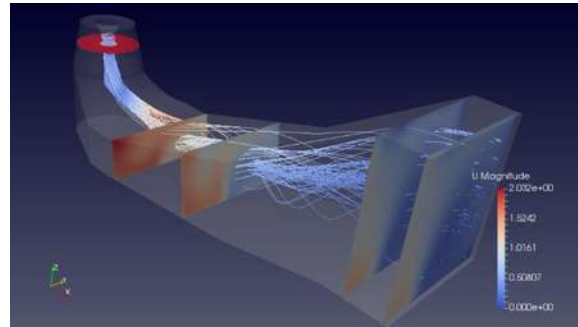


Figura 64. Contornos de velocidad en m/s con RapidCFD y solucionador PCG/PBiCG con preconditionador AINV.
Tolerance 10^{-12} .

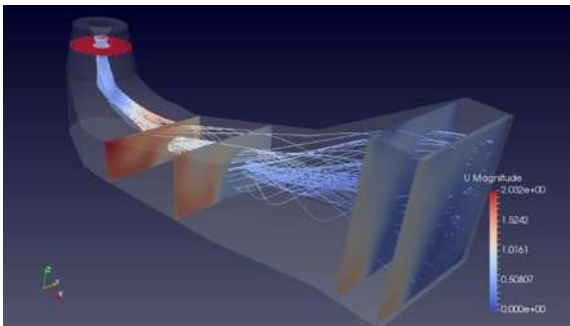


Figura 65. Contornos de velocidad en m/s con RapidCFD y solucionador PCG/PBiCG con preconditionador *diagonal*.
Tolerance 10^{-12} .

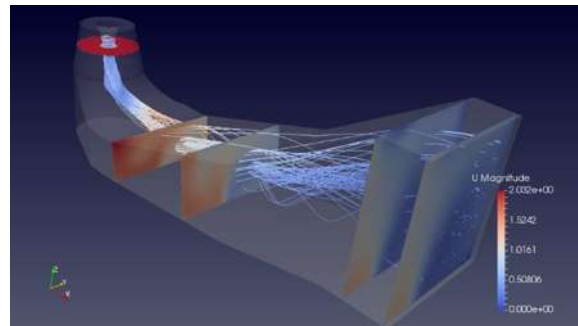


Figura 66. Contornos de velocidad en m/s con RapidCFD y solucionador PCG/PBiCG con preconditionador *diagonal*.
Tolerance 10^{-06} .

Únicamente se muestran los resultados de WSPAC ya que los resultados son idénticos a los obtenidos en WSGAL y WSMOL.

Adicionalmente se realizaron pruebas empleando únicamente OpenFOAMv1706 con CPUs en WSPAC, WSGAL y WSMOL. Al realizar el postprocesamiento de los resultados de CFD se observó un comportamiento similar al que se presentó empleando RapidCFD y GPUs, esto es, que tampoco existe una diferencia apreciable entre usar PCG/PBiCG y GAMG con distintos valores de *tolerance*.

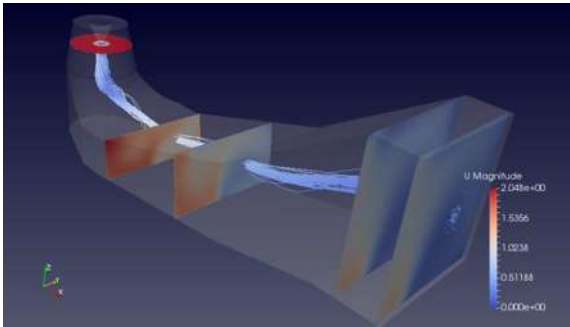


Figura 67. Contornos de velocidad en m/s con OpenFOAMv1706 y solucionador GAMG con suavizador GasussSeidel.
Tolerance 10^{-12} .

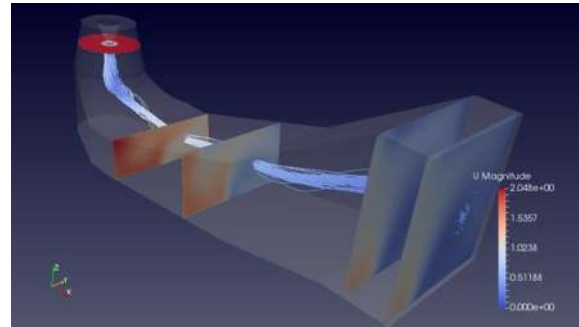


Figura 68. Contornos de velocidad en m/s con OpenFOAMv1706 y solucionador GAMG con suavizador GasussSeidel.
Tolerance 10^{-06} .

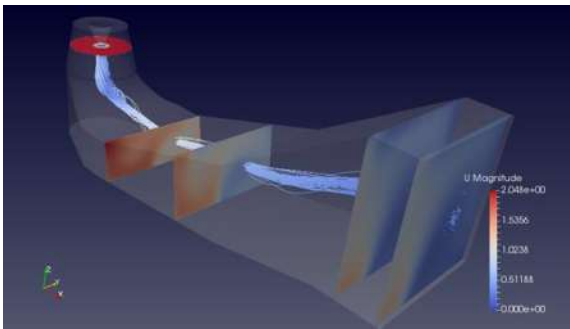


Figura 69. Contornos de velocidad en m/s con OpenFOAMv1706 y solucionador PCG/PBiCG con preconditionador *diagonal*.
Tolerance 10^{-12} .

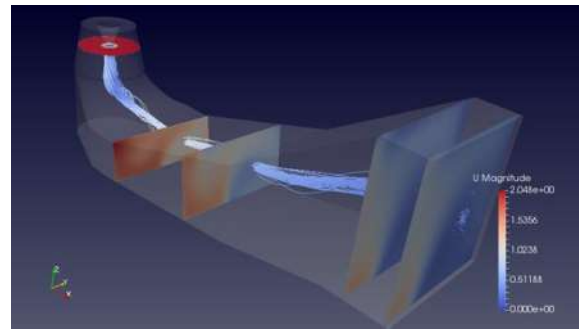


Figura 70. Contornos de velocidad en m/s con OpenFOAMv1706 y solucionador PCG/PBiCG con preconditionador *diagonal*.
Tolerance 10^{-06} .

La comparación entre RapidCFD y OpenFOAMv1706 con estos últimos resultados muestran una pequeña diferencia tanto en los valores de velocidad como de presión. Es probable que esta diferencia se deba a la diferencia entre los software, ya que RapidCFD está basado en OpenFOAM v2.3.1 y la versión OpenFOAMv1706 es equivalente a OpenFOAM v5.0, entre cada versión se van modificando algunos algoritmos y funciones. Otro factor que podría influir en los resultados en la transferencia de información entre los núcleos de procesamiento, ya sean GPGPU cores o CPU cores, y la memoria RAM en cada aplicación.

Mediciones detalladas de velocidad y presión fueron realizadas por [47] en varias secciones del tubo difusor de la turbina T99. Estas mediciones se utilizan tanto para establecer las condiciones de frontera, como para validar los resultados de las simulaciones numéricas. Una vista en perspectiva del tubo de aspiración se muestra en la Figura 71. La Figura 72 presenta un plano simétrico vertical que incluye seis secciones transversales. Para dar certeza a los resultados de la solución obtenida a través de CFD, se realizó una comparación con los datos experimentales disponibles, donde se tomó como referencia el plano CsII.

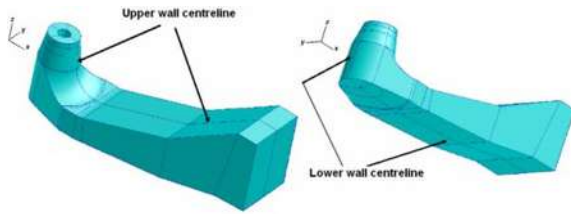


Figura 71. Líneas centrales superior e inferior del tubo difusor.

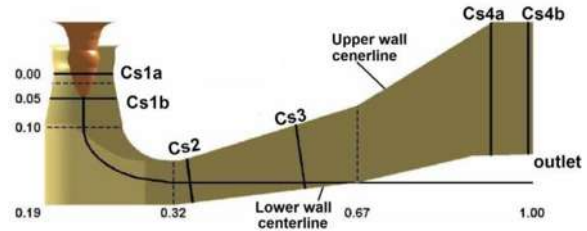


Figura 72. Secciones de medición del tubo difusor.

Tomando como punto de comparación un dominio computacional de 1 millón de celdas aproximadamente, en la Figura 73 se puede observar que los resultados de CFD para el componente de velocidad axial (U_x) son diferentes de los datos experimentales, ya que en estos últimos la región de baja velocidad está en la esquina inferior del lado izquierdo y el CFD lo muestra bastante más centrado. Sin embargo, los datos generados por CFD en el centro y el lado derecho de la sección de prueba CsII presentan una buena aproximación a los datos experimentales. Los datos disponibles del cálculo en CFD de Vattenfall Utveckling AB [37] se incluyen como referencia para ilustrar resultados computacionales típicos para el caso obtenidos en Fluent con una malla de aproximadamente 700,000 celdas.

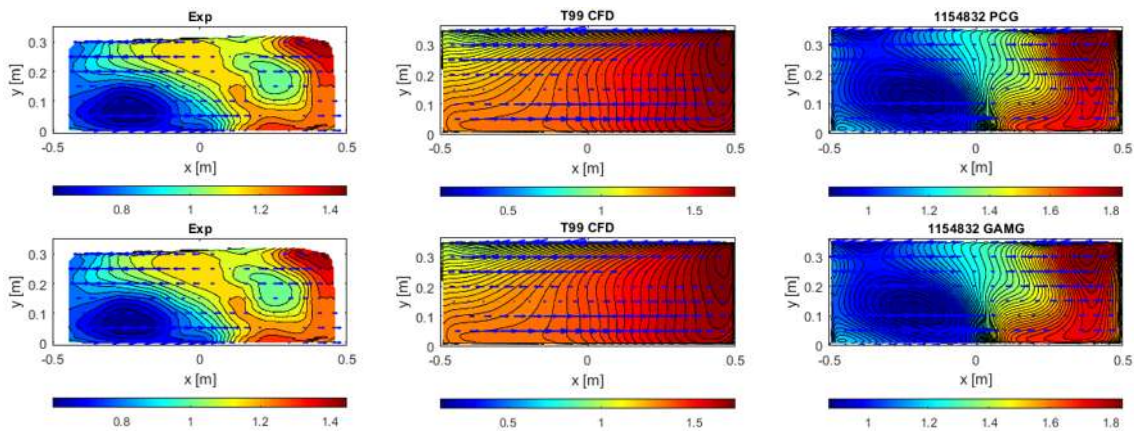


Figura 73. Contornos de velocidad axial U_x y vectores de la componente de velocidad horizontal U_y en m/s en la sección CsII (el flujo se mueve hacia el lector). De izquierda a derecha: 1) Experimental, 2) T99 CFD de referencia y 3) malla de 1 millón de celdas.

Cualitativamente, los resultados de CFD podrían considerarse una buena aproximación al flujo real, sin embargo, se realizó la verificación a través del $C_{p_{wall}}$ (coeficiente de recuperación de presión).

El coeficiente de recuperación de presión en la pared obtenido mediante CFD con diferentes tamaños de malla se comparó con los datos experimentales disponibles. La Figura 72 presenta la línea central superior e inferior del tubo de aspiración a lo largo de la cual el

coeficiente de recuperación de presión estática se midió experimental y computacionalmente usando la siguiente ecuación:

$$Cp_{wall} = \frac{P_{out} - P_{in}}{\frac{1}{2}\rho U^2} \quad \text{Ecuación 3}$$

Donde P_{out} es la presión estática en la pared en diferentes puntos a lo largo de la línea central de la pared, P_{in} es la presión estática en CsIa, ρ es la densidad y U_{in} es la velocidad media en CsIa.

El factor de recuperación de presión indica el grado de conversión de la energía cinética en presión estática, donde un valor más alto significa una mayor eficiencia para el tubo de aspiración. El valor exacto del factor de recuperación de presión depende de la solución del campo completo y puede verse como una propiedad integral de la solución. Debido a que la presión de pared estática promedio de entrada, P_{in} , se ha encontrado muy sensible a las condiciones de entrada y la geometría en la entrada, se ha utilizado un $Cp_{average}$ alternativo definido como [37]:

$$Cp_{average} = \frac{P_{out:mean} - P_{in:mean}}{\frac{1}{2}\rho U^2} \quad \text{Ecuación 4}$$

Donde $P_{out:mean}$ y $P_{in:mean}$ son las presiones estáticas promedio en estas secciones. El $Cp_{average}$ es un resultado directo de toda la solución de campo para todas las variables (si la presión se establece en la salida) y mucho menos sensible a los diferentes cambios en la entrada.

Si bien los contornos de velocidad muestran un acuerdo aceptable, la Figura 74 y Figura 75 muestran que los valores de Cp calculados a partir de CFD no concuerdan con los valores experimentales calculados en la línea central superior e inferior del tubo de aspiración. Nuevamente en los gráficos de Cp se han incluido los resultados CFD disponibles de uno de los participantes en el taller T99 como referencia.

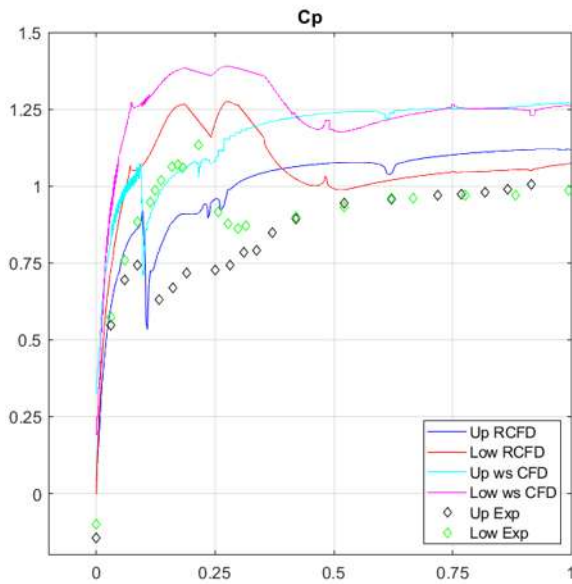


Figura 74. Comparación de la recuperación de presión en la línea central inferior y superior del tubo de aspiración utilizando solucionadores GAMG con una malla de 1 millón de celdas.

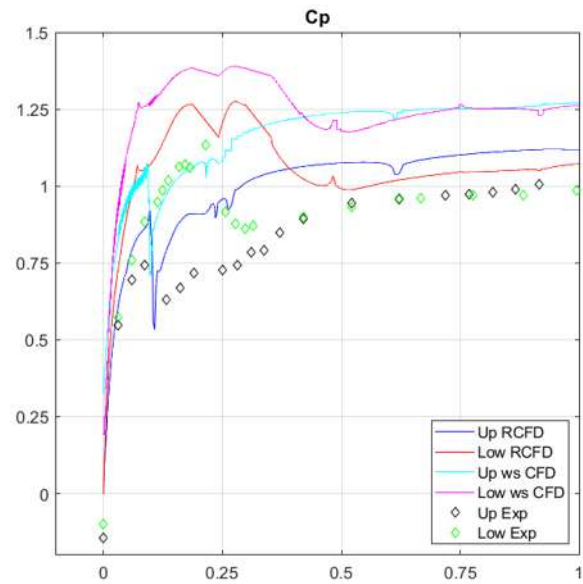


Figura 75. Comparación de la recuperación de presión en la línea central inferior y superior del tubo de aspiración utilizando solucionador PCG/PBiCG con una malla de 1 millón de celdas.

Teóricamente, un mayor número de celdas mejora la precisión de los resultados y la convergencia de la solución. Por lo tanto, se desarrollaron pruebas para estimar el tamaño de problema que se deberá resolver a fin de alcanzar un nivel de precisión y resolución de flujo aceptables en CFD, es decir el tamaño del dominio computacional en términos del número de elementos. Se solucionaron cinco dominios computacionales, desde el tamaño de malla más grueso al más fino: 0.5, 1, 2, 3 y 4 millones de celdas aproximadamente utilizando el solucionador GAMG, que ha mostrado tiempos de simulación más bajos y un resultado igual al PCG/PBiCG.

Para la malla más fina de 4 millones de celdas, los gráficos de contorno de los resultados de CFD en la Figura 76 muestran una mejor aproximación en las partes central y derecha del plano de prueba CsII. Sin embargo, el lado izquierdo de la gráfica no coincide con los datos experimentales como en la prueba con la malla de 1 millón de celdas.

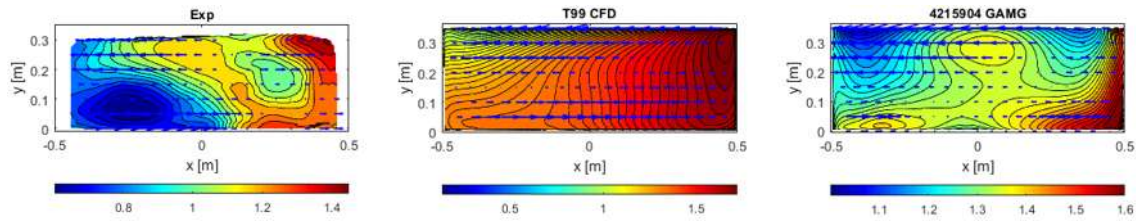


Figura 76. Contornos de velocidad normal U_x y vectores del componente de velocidad horizontal U_y en m/s en la sección CsII (el flujo se mueve hacia el lector). De izquierda a derecha: 1) Experimental, 2) T99 CFD de referencia y 3) malla refinada de 4 millones de celdas.

A pesar de esto, la Figura 77 y la Figura 78 muestran un mejor acuerdo entre los resultados de CFD y los datos experimentales a medida que aumenta el tamaño de malla, lo que indica que un refinamiento de malla mayor conducirá a una mejor solución, incluso sin implementar modelos de turbulencia más robustos o esquemas de discretización precisos como en [48] [49].

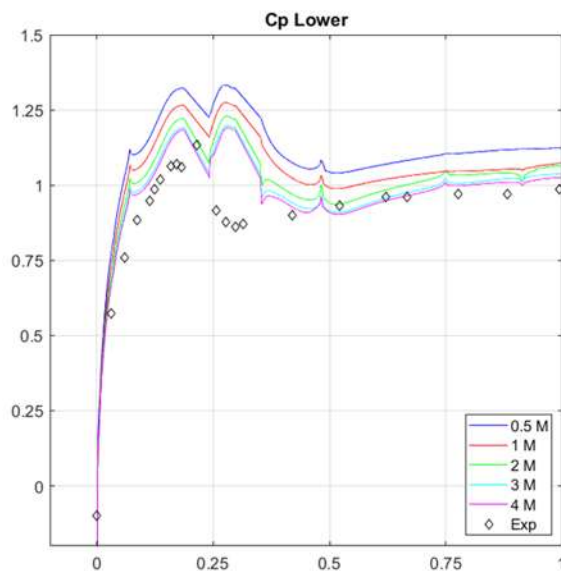


Figura 77. Comparación de la recuperación de presión en la línea central inferior del tubo de aspiración con diferentes tamaños de rejilla.

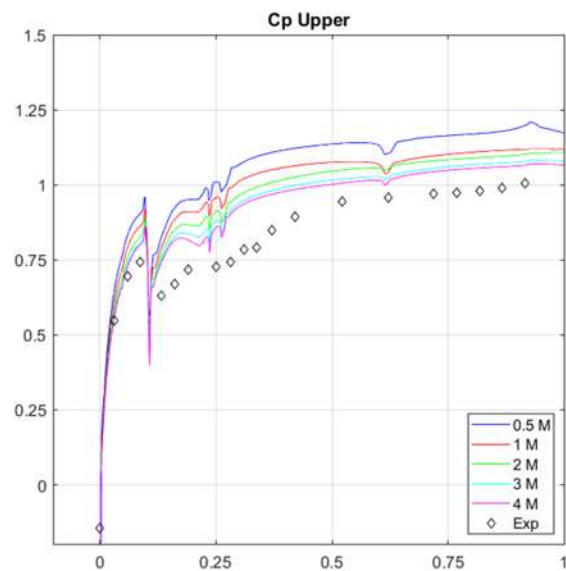


Figura 78. Comparación de la recuperación de presión en la línea central superior del tubo de aspiración con diferentes tamaños de rejilla.

Para verificar la solución en CFD, como se describe en [50], los valores de las variables de rendimiento como $C_{p_{wall}}$, se pueden extrapolar en función de los valores iniciales obtenidos en CFD y a partir de la solución de malla escalados utilizando los métodos de extrapolación de Richardson. Los valores de extrapolación se pueden calcular mediante dos enfoques, primer y segundo orden, que dependen del método utilizado en los métodos numéricos para resolver el problema estudiado, como se mencionó antes, el FVM es un esquema de segundo orden, por lo tanto, se utilizó un enfoque de segundo orden.

En Figura 79 para la línea central superior y la Figura 80 para la línea inferior, se pueden observar los resultados del análisis de convergencia. Los valores de extrapolaciones de primer orden $\varphi(1)$ y de segundo orden $\varphi(2)$ del Cp_{wall} total se muestran junto con los valores obtenidos de las soluciones CFD, φ , y los valores experimentales, $\varphi(E)$. En los gráficos, α indica el nivel de refinamiento, cuanto menor es el valor, más fino es el tamaño de malla, donde un valor cero corresponde a un continuo. Para el Cp_{wall} , solo las tres mallas más finas están dentro del rango de convergencia asintótico. Por lo tanto, la malla que tiene 2 millones de celdas parece estar dentro del rango que minimizaría los cálculos de CFD.

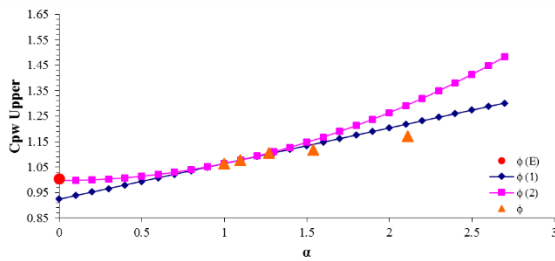


Figura 79. Gráficos de Cp: extrapolaciones de primer $\varphi(1)$ y segundo $\varphi(2)$ orden, soluciones CFD φ y valores experimentales $\varphi(E)$ para el análisis de convergencia de malla en la línea central superior.

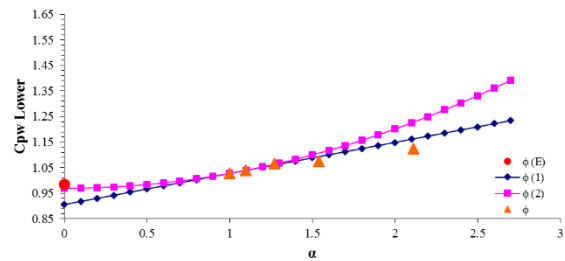


Figura 80. Gráficos de Cp: extrapolaciones de primer $\varphi(1)$ y segundo $\varphi(2)$ orden, soluciones CFD φ y valores experimentales $\varphi(E)$ para el análisis de convergencia de malla en la línea central inferior.

El error del CFD y los valores extrapolados de Cp_{wall} frente a los experimentales se presentan en la Tabla 17.

Tabla 17. Cálculo de error contra valores experimentales de Cp_{wall}

Source	Error %	
	Upper wall	Lower wall
0.5 M cells	14.36	12.28
1 M cells	10.21	8.34
2 M cells	9.25	7.51
3 M cells	6.83	5.12
4 M cells	5.62	4.01
1st Order	8.67	8.88
2 nd Order	0.69	1.73

Después de concluir la etapa de validación de las simulaciones numéricas del caso de estudio de la turbina T99, se logra dar confiabilidad a los resultados obtenidos de las simulaciones numéricas para la solución en CFD de modelos complejos de turbo maquinaria. Sumado a los datos de aceleración de cálculo, demuestra la viabilidad del proyecto para la creación de un banco de pruebas virtual basado en HPC para turbinas tipo Francis.

RapidCFD vs. cufflink.

Una de las actividades que más tiempo consume en HPC es la realización de simulaciones mediante CFD. Las arquitecturas híbridas que incluyen configuraciones de CPU + GPU han demostrado acelerar los procesos de computación en diferentes aplicaciones. Sin embargo, las opciones confiables de código abierto para aprovechar este tipo de combinación de hardware para CFD son pocas. Dos alternativas que se mantienen vigentes son cufflink y RapidCFD. Sin embargo, aún son desconocidos parámetros como el uso de memoria y el tiempo de cálculo al elegir alguno de los software para aplicaciones de turbomaquinaria. Por lo tanto, se desarrollaron varias pruebas para resolver el caso de la turbina T99. El *wall clock time* y la RAM requerida se midieron a lo largo de un análisis de convergencia con el fin de ofrecer pautas al usuario de CFD sobre los criterios más representativos e importantes para elegir una aplicación basada en CUDA para aprovechar la potencia computacional de las GPU en el análisis de turbomaquinaria.

Se solucionaron 4 diferentes tamaños de dominio de la turbina T99, estos fueron: 1) 446,820 celdas, 2) 1,154,832 celdas, 3) 2,051,712 celdas y 4) 3,179,080 celdas. Cada dominio fue resuelto usando RapidCFD y cufflink, en las tres *workstations* WSPAC, WSGAL y WSMOL. De esta manera se realizaron 24 simulaciones en total para obtener datos de tiempo de cálculo y uso de RAM en las GPUs. En las simulaciones se empleó el solucionador PCG/PBiCG en conjunto con el pre acondicionador *diagonal* para RapidCFD, mientras que en cufflink se empleó el solucionador *cudaCG/cudaPBiCGStab* con el pre acondicionador *diagonal*. El algoritmo de solución SIMPLE y el modelo de turbulencia *k-ε standard* se utilizaron nuevamente para alcanzar un valor de *residuals* de 10^{-03} como criterio de convergencia, así como un valor de *tolerance* de 10^{-06} .

En la Figura 81 se muestra como el tiempo de cálculo por iteración aumenta a medida que se incrementa el número de celdas dentro del dominio computacional, así como la influencia del tipo de hardware empleado. También se puede observar la influencia que tiene el tipo de implementación usada para la solución del caso. Es claro que RapidCFD es más rápido que cufflink para la solución del caso.

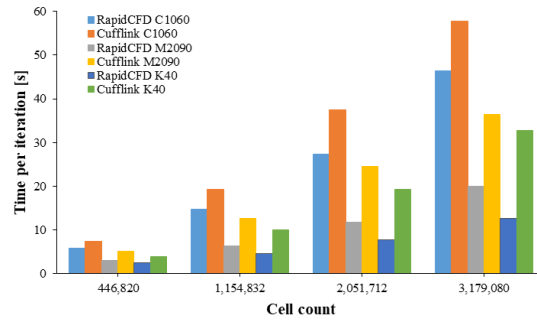


Figura 81. Tiempo de cálculo por time/step para cada dominio computacional utilizando diferentes software (RapidCFD, cufflink) y hardware (Tesla C1060, M2090, K40).

La Figura 82 muestra el tiempo total de cálculo de todos los casos solucionado, en promedio cufflink requiere un 22.82% más de tiempo para la solución de cada caso cuando se emplean GPUs C1060, un 79.8% más de tiempo cuando se emplean M2090 y un 107% más de tiempo cuando se usan K40, respecto al uso de RapidCFD. Aun con esta marcada diferencia en el tiempo extra requerido para la solución de los casos, las GPUs K40 solucionaron todos los casos más rápido utilizando tanto RapidCFD como cufflink.

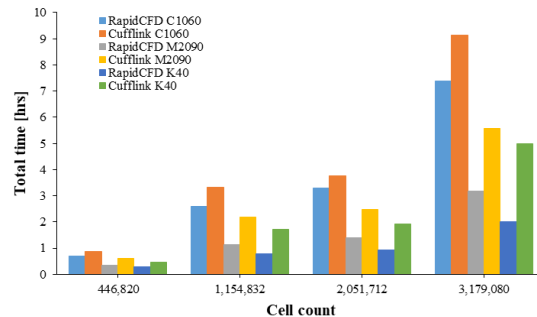


Figura 82. Tiempo de cálculo total para cada dominio computacional utilizando diferentes software (RapidCFD, cufflink) y hardware (Tesla C1060, M2090, K40).

También se obtuvieron datos del uso de RAM empleando RapidCFD y cufflink en la solución de cada caso. En la Figura 83 se puede observar que el uso de RAM es prácticamente el mismo en con las tres distintas GPUs. Sin embargo, al usar cufflink con GPUs C1060 se requiere en promedio 77.20% menos de RAM, mientras que al usar GPUs M2090 este valor disminuye a 76.85% y finalmente al utilizar GPUs K40 el uso de memoria RAM disminuye hasta 73.3% menos.

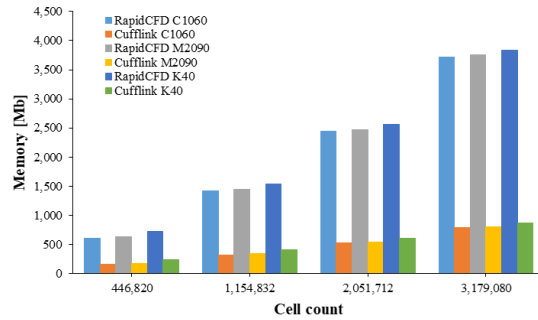


Figura 83. Memoria usada en GPUs para cada dominio computacional utilizando diferentes software (RapidCFD, cufflink) y hardware (Tesla C1060, M2090, K40).

Los datos anteriormente descritos se resumen en la Figura 84. De esta grafica se puede tomar la decisión de qué tipo de hardware y software usar la momento de realizar simulaciones numéricas en CFD dependiendo de las necesidades y recursos disponibles, ya que cuando se requiera velocidad de cálculo sobre exactitud de los resultados, se podría optar por usar RapidCFD, en caso de dar mayor importancia a la exactitud que a la velocidad de cálculo se puede optar por cufflink.

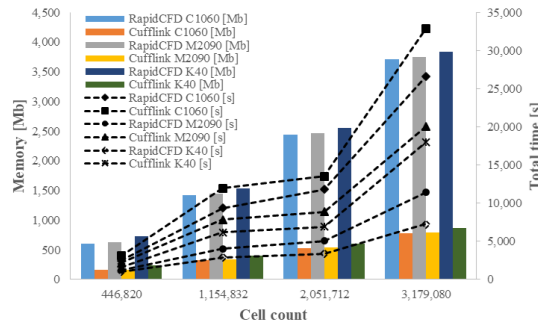


Figura 84. Memoria usada en GPUs vs tiempo de cómputo para cada dominio computacional utilizando diferentes software (RapidCFD, cufflink) y hardware (Tesla C1060, M2090, K40).

Simulaciones RANS.

Hoy en día CFD es una herramienta confiable que se puede utilizar en la predicción del rendimiento de turbomáquinas como lo demuestra [16], donde se realizaron una serie de pruebas para una turbina Francis a diferentes velocidades específicas y se comparó con datos experimentales, en general se observó una buena concordancia para las cantidades analizadas. El despliegue de CFD en aplicaciones de turbomáquinas es de hecho bastante importante en la optimización y el diseño de nuevos equipos para la modernización de plantas de energía.

A medida que la potencia de cálculo ha aumentado, la complejidad y precisión de los fenómenos resueltos y analizados en CFD ha seguido el mismo camino, comenzando con el flujo potencial para seguir con la solución de las ecuaciones promedio de Reynolds Navier-Stokes (RANS) en estado estable y transitorio con el fin de incluir efectos viscosos [51]. Por lo tanto, se ha llegado a analizar geometrías más complejas dentro de un umbral de precisión aún más fino, aquí es donde las GPU han surgido como un nuevo paradigma para resolver problemas computacionales complejos.

Recientemente, se ha abierto el caso de estudio Francis 99 [52] para mejorar las capacidades de CFD y aplicar herramientas y técnicas modernas a las aplicaciones de turbomáquinas. El caso de prueba Francis 99 es una excelente oportunidad para probar esta implementación ya que el modelado numérico de turbinas hidráulicas es todo un desafío. Primero, porque el modelado específico de un problema para investigar una condición operativa no siempre funciona para otros; segundo, la simulación de una turbina completa requiere recursos computacionales sustanciales.

El caso de prueba Francis 99 consiste en un modelo de turbina tipo Francis (1:5.1) de la turbina que opera en la planta de energía Tokke en Noruega. La Figura 85 muestra el modelo completo de la turbina Francis y los alabes del rodete.

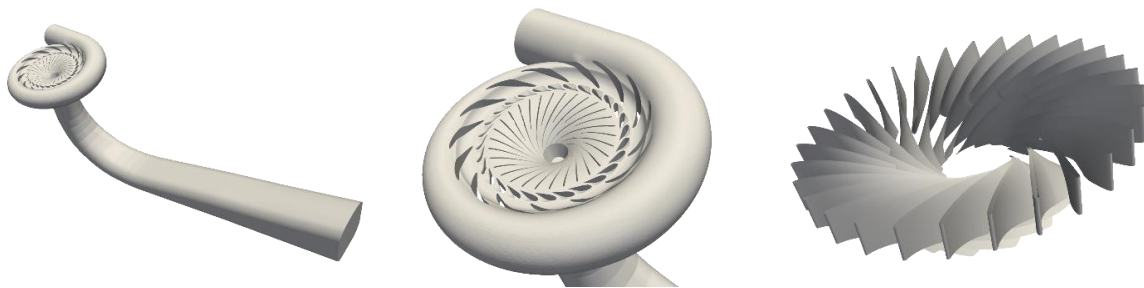


Figura 85. Vistas isométricas de la turbina Francis 99 (izquierda), de cerca (centro) y de los alabes del rodete (izquierda).

Las condiciones de operación analizadas fueron carga parcial (PL), mejor punto de eficiencia (BEP) y carga alta (HL). La Tabla 18 muestra los datos disponibles que se utilizaron como

condiciones de frontera y validación en este trabajo; se muestran condiciones de operación duplicadas, ya que se realizaron 2 talleres con algunas diferencias en los datos experimentales.

Tabla 18. Datos de las condiciones de funcionamiento de la serie de talleres Francis 99.

Parameter	PL1	PL2 ^a	BEP	BEP ^a	HL	HL ^a
Turbine inlet pressure absolute (kPa)	219.93	218.08	216.54	215.57	210.01	212.38
Differential pressure across the turbine (kPa)	120.39	104.91	114.98	104.44	114.03	102.79
Water density (kg/m³)	999.23	999.80	999.19	999.80	999.20	999.80
Kinematic Viscosity (m²/s)	9.57E-07	9.57E-07	9.57E-07	9.57E-07	9.57E-07	9.57E-07
Gravity (m/s²)	9.82	9.82	9.82	9.82	9.82	9.82
Net head (m)	12.29	11.87	11.91	11.94	11.24	11.88
Discharge (m³/s)	0.07100	0.13962	0.20300	0.19959	0.22100	0.24246
Runner torque (N-m)	144.06	420.79	628.41	620.65	605.62	744.3900
Runner speed (rpm)	406.20	332.84	335.40	332.59	369.60	332.5900
Hydraulic efficiency (%)	71.6900	90.1300	92.6100	92.3900	90.6600	91.7100
Guide vane angle (degree)	3.9100	6.7200	9.8400	9.8400	12.4400	12.4400

^a Second workshop

Para calcular el *speedup* y la exactitud de las simulaciones utilizando OpenFOAM y RapidCFD, ya que esta última es una implementación más veloz respecto a cufflink, se realizaron varias pruebas en la *worstation* WSMOL, cuyas características se detallan en la Tabla 14. Se utilizó el algoritmo del SIMPLE para alcanzar el nivel de convergencia de *residuals* de 10^{-03} . Todas las simulaciones se resolvieron con doble precisión flotante utilizando la aplicación *simpleFoam* en estado estable bajo la aproximación de Marco de Referencia Multiple (MRF). A fin de asegurar la estabilidad y la convergencia para los casos de prueba, se utilizó un esquema de discretización de primer orden *upwind* para los términos convectivos, incluso cuando es sabido que este esquema tiende a ser difusivo, se ha demostrado que puede ser utilizado como una buena aproximación inicial en aplicaciones de turbomáquinas [48]. Los términos restantes de gradiente y laplaciano utilizaron un esquema de discretización de segundo orden *lineal*. El solucionador lineal utilizado fue GAMG y se incluyó en el proceso el suavizador *GaussSeidel*. En todas las simulaciones se utilizó el modelo de turbulencia *k- ω SST*.

La configuración descrita anteriormente se utilizó para resolver un dominio de 22,984,188 celdas para cada condición de operación. La malla fue proporcionada por NTNU [52] en dominios separados para cada componente que se fusionaron en un solo dominio utilizando herramientas de pre procesamiento OpenFOAM. Para transferir información entre dominios se utilizó la condición de frontera de Interfaz de Malla Arbitraria (AMI). Con el fin de conservar concordancia en la comparación de la velocidad y la precisión de la simulación, se utilizaron la misma configuración y condiciones de frontera en OpenFOAM y RapidCFD, o al menos en opciones equivalentes

La condición BEP del primer taller se seleccionó como punto de referencia de comparación para la aceleración. El dominio computacional se resolvió usando 1, 4 y 32 hilos en OpenFOAM v1912 mientras que en RapidCFD se resolvió usando 4 GPUs equivalentes a 11,520 núcleos. En la Figura 86 se muestra la aceleración calculada a partir de la Ecuación 1. La Figura 87 muestra el tiempo del reloj de pared medido durante las simulaciones, que es el tiempo transcurrido entre el inicio del proceso y su finalización.

Como se puede observar, el speed up máxima usando OpenFOAM con la capacidad de CPU de la máquina completa fue de 9.78 mientras que usando RapidCFD con 4 subprocesos de CPU más 4 GPUs fue de 20.53. Este valor supera los reportados anteriormente en trabajos de investigación relacionados con aplicaciones de GPUs en CFD [53] [45] [40]. Esto podría ser consecuencia del tamaño del caso de prueba, ya que las CPU y GPU funcionan mejor en paralelo a medida que aumenta el tamaño de problema.

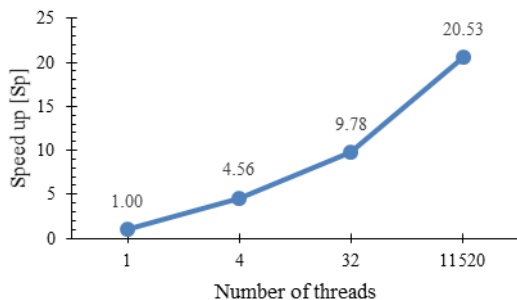


Figura 86. *Speedup* en la condición de BEP usando varios subprocesos.

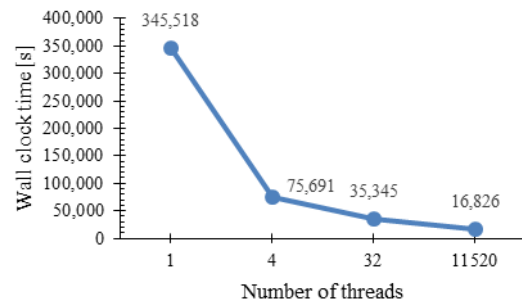


Figura 87. Tiempo del reloj de pared para la condición BEP usando varios subprocesos.

La validación de los resultados de CFD tiene una gran relevancia si se desea tener un modelo confiable en futuros procesos de optimización. Por lo tanto, se realizó una validación con datos experimentales para la diferencia de presión en la turbina, el par generado por el rodete, la carga neta y la eficiencia hidráulica.

Los valores de presión diferencial y torque se extrajeron directamente de las simulaciones CFD en entrada, salida y rodete, respectivamente. Con respecto a la altura neta h_{net} y la eficiencia hidráulica η_H , se calcularon utilizando datos adicionales de la siguiente manera:

$$h_{net} = \frac{p_{in} - p_{out}}{\rho g} + \frac{v_{in}^2 - v_{out}^2}{2g} + (z_1 - z_2) \quad \text{Ecuación 5}$$

$$\eta_H = \frac{T\omega}{\rho g h_{net} Q} \quad \text{Ecuación 6}$$

Donde $p_{in} - p_{out}$ representa la presión diferencial a través de la turbina, $v_{in}^2 - v_{out}^2$ representa la diferencia en la magnitud de la velocidad en la entrada y la salida, $z_1 - z_2$ es la diferencia de altura entre la entrada y la salida, $T\omega$ es la potencia producida por el rodete y $\rho g h_{net} Q$ es la potencia disponible en el sistema.

Como se puede observar en la Tabla 19, la presión diferencial, el torque del rodete y la altura neta están sobreestimados por OpenFOAM y RapidCFD; sin embargo, la eficiencia hidráulica está subestimada. Este comportamiento podría ser consecuencia de la naturaleza difusiva del esquema de discretización de primer orden utilizado en los términos convectivos. Sin embargo, los datos de CFD pueden considerarse una buena aproximación inicial. También se puede observar que no hay gran diferencia entre los resultados de RapidCFD y OpenFOAM a pesar de la cantidad de hilos utilizados para la solución en ambos casos.

Tabla 19. Comparación de datos experimentales frente a CFD para BEP.

	Experiment	RapidCFD	Diff. (%)	OpenFOAM	Diff. (%)
Differential pressure (kPa)	114.9800	122.4897	6.13%	123.4399	6.85%
Runner torque (N-m)	628.4100	680.0770	7.60%	679.3510	7.50%
Net head (m)	11.9100	13.8103	13.76%	13.9072	14.36%
Hydraulic efficiency (%)	92.61	87.04	-6.40%	86.34	-7.26%

La variación de presión a través de la turbina calculada mediante CFD se puede observar en la Figura 88. Se aprecia que la transformación de energía potencial (presión) a cinética ocurre en la zona de los alabes guía (distribuidor), la cual es aprovechada en el rodete. Además, se muestra una zona de baja velocidad en la zona del hub asociada con el flujo rotatorio presente. La Figura 89 presenta una mejor visualización de los cambios de presión y velocidad en el conjunto carcasa, distribuidor y rodete.

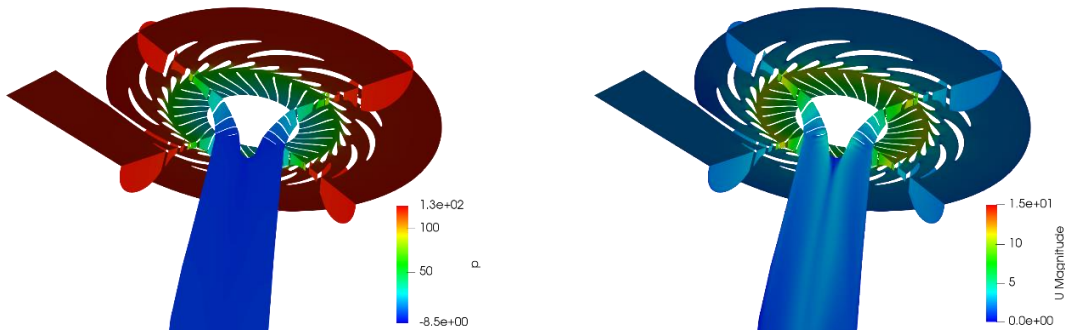


Figura 88. Contornos de valores de presión en Pa (izquierda) y velocidad en m/s (derecha) en la turbina Francis 99 para la condición BEP.

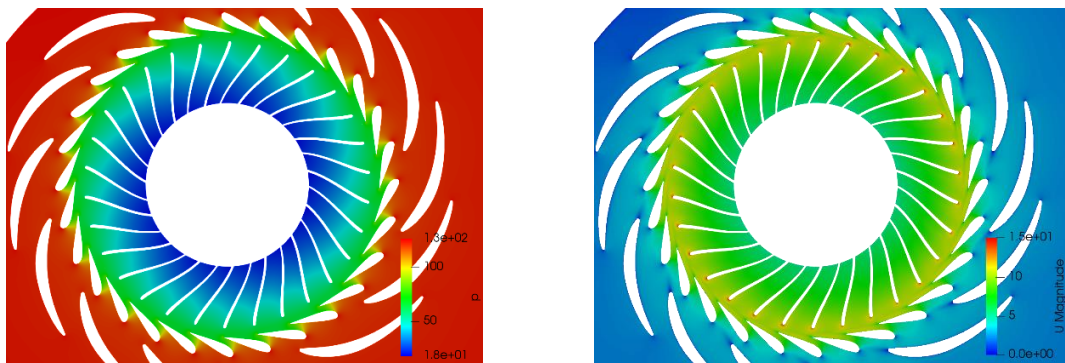


Figura 89. Contornos de valores de presión en Pa (izquierda) y velocidad en m/s (derecha) en carcasa, distribuidor y rodete de la turbina Francis 99 para la condición BEP.

La interacción entre el fluido y el rodete en forma de una diferencia de fuerzas producto de variaciones de presión se puede apreciar en la Figura 90, así mismo, el momento producto del desequilibrio de fuerzas es visible, el cual produciría una aceleración constante y velocidad angular infinita teóricamente de no ser por la extracción de energía que equilibra el sistema.

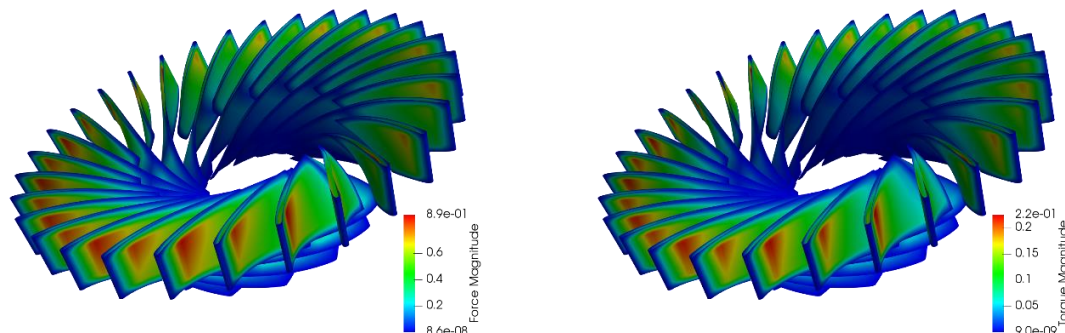


Figura 90. Contornos de valores de fuerza en N (izquierda) y torque en N-m (derecha) en el rodete de la turbina Francis 99 para la condición BEP.

Las simulaciones CFD para el resto de condiciones de operación y sus variaciones para el 1° y 2° taller se realizaron únicamente en RapidCFD. El tiempo requerido y el número de pasos de tiempo (iteraciones) para alcanzar la convergencia fueron cercanos entre sí y se esperaba un comportamiento similar en relación con la aceleración. En la Figura 91 se muestra el tiempo de simulación requerido para cada condición de operación. Incluso cuando la condición operativa de HL tomó más tiempo para resolverse; no es el tiempo máximo por paso de tiempo, que pertenece a la condición de operación PL1 (Figura 92), el caso resuelto más rápido. Esto muestra cómo los cálculos de CFD se ven afectados por la geometría, la malla, las condiciones de frontera y los esquemas de discretización cuando ocurren pequeños cambios; en consecuencia, una configuración para una condición no siempre puede funcionar para otras, lo que se mostrará más adelante.

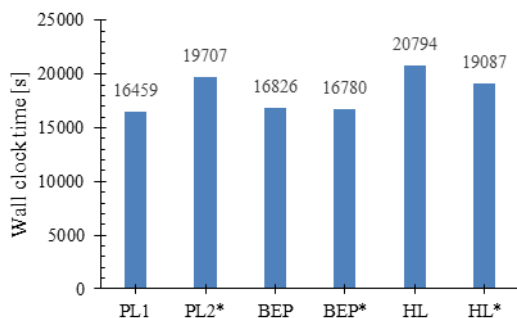


Figura 91. Tiempo del reloj de pared para diferentes condiciones de operación usando RapidCFD.

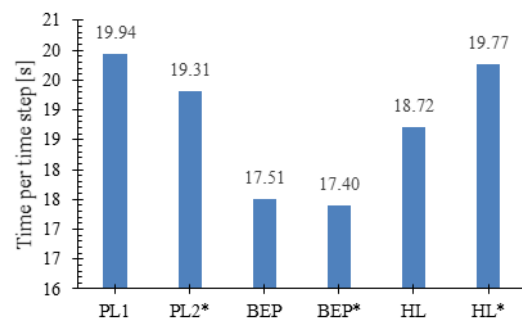


Figura 92. Tiempo por paso de tiempo para diferentes condiciones de operación usando RapidCFD.

Al igual que se hizo anteriormente, los datos experimentales se compararon con los resultados de CFD para todas las condiciones de operación. Se evaluaron la presión diferencial, el par, la altura neta y la eficiencia hidráulica. La Figura 93 muestra cómo la presión diferencial en la mayoría de las condiciones de operación es mayor que lo estimado por el CFD a excepción de PL1 y HL.

Se podría suponer que los valores de torque del rodete seguirán la misma tendencia que los datos de CFD de presión diferencial, es decir, serán sobre predichos para todas las condiciones de operación, pero no es el caso. Como se muestra en la Figura 94, el par para la condición PL1 está dramáticamente subestimado, el valor experimental es 144.06 N-m, mientras que el valor CFD es 4.59 N-m. Esto podría atribuirse a múltiples causas, una de ellas es el esquema de discretización para los términos convectivos, y otra podría ser el modelo de turbulencia o incluso las condiciones de frontera.

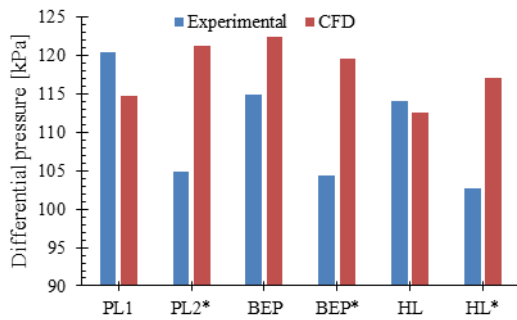


Figura 93. Valores de presión diferencial para diferentes condiciones de operación usando RapidCFD vs. datos experimentales.

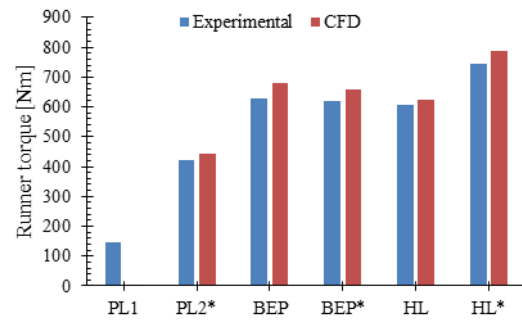


Figura 94. Valores de torque del rodete para diferentes condiciones de operación usando RapidCFD vs. datos experimentales.

Debido a que los valores netos de carga están relacionados con la presión diferencial entre otras variables, los valores del CFD son cercanos a los experimentales, sin embargo están sobre estimados para todas las condiciones de operación como se puede observar en la Figura 95. La influencia combinada del torque del rodete y los cálculos de altura neta por el CFD producen que los valores de eficiencia hidráulica caen por debajo de los experimentales en todos los casos. Esta tendencia es más visible para la condición PL1 en la Figura 96, donde el valor experimental es 71.69% y 2.20% para el CFD, la razón principal es la mala aproximación del torque del rodete.

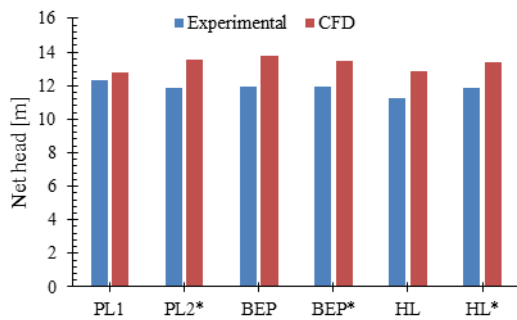


Figura 95. Valores de altura neta para diferentes condiciones de operación usando RapidCFD vs. datos experimentales.

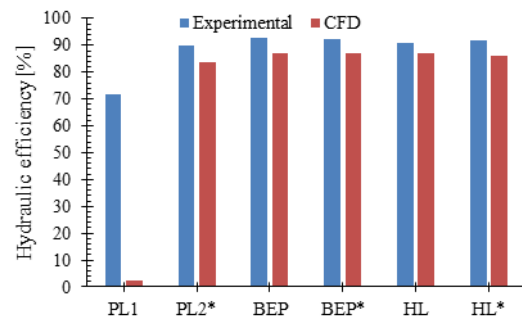


Figura 96. Valores de eficiencia hidráulica para diferentes condiciones de operación usando RapidCFD vs. datos experimentales.

Los artículos de investigación relacionados [13] [14] [54] [55] informaron discrepancias significativas con respecto a los datos experimentales para los valores de eficiencia hidráulica obtenidos de CFD para la condición PL1 (ángulo alabe guía = 3.91 °), mientras que los otros valores de las condiciones de operación de eficiencia hidráulica presentan mejor concordancia con los valores experimentales.

Cálculos en estado estable utilizando MRF

Debido a que hasta ahora únicamente se realizaron pruebas en estado estacionario utilizando la aproximación MRF y el modelo de turbulencia $k-\omega SST$, se añadieron pruebas adicionales para todas las condiciones de operación previamente utilizadas con el fin de determinar cuál modelo de turbulencia proporciona los resultados más aproximados a los datos experimentales al realizar la validación de los mismos. Los modelos adicionales que se incluyeron en las pruebas fueron: $k-\varepsilon Realizable$ y $k-\varepsilon RNG$. La elección de estos tres modelos se basa en el hecho de que son los de uso más común en CFD de turbomaquinaria y que dan como resultado las mejores aproximaciones.

Desde la Figura 97 a la Figura 100 se muestran los resultados para la condición de operación de carga parcial PL1, se observa que la presión diferencial es calculada de mejor manera por el modelo $k-\varepsilon RNG$, sin embargo la carga neta se calcula mejor con el modelos $k-\omega SST$. En cuanto al torque y la eficiencia ninguno de los modelos de turbulencia genera resultados aceptables.

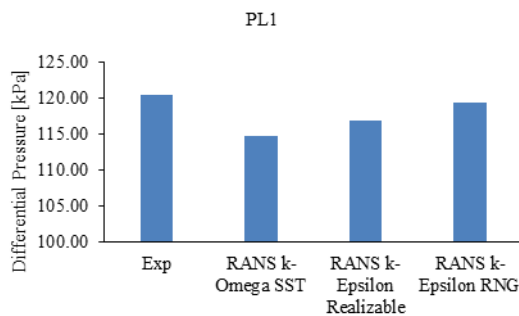


Figura 97. Valores de presión diferencial para diferentes modelos de turbulencia usando RapidCFD.

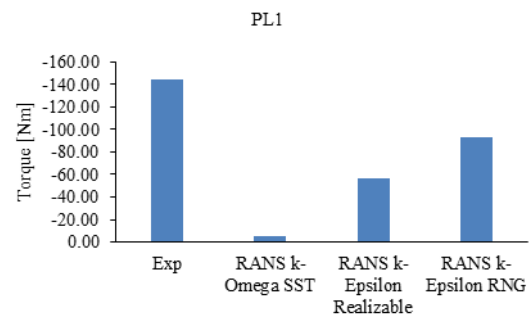


Figura 98. Valores de torque para diferentes modelos de turbulencia usando RapidCFD.

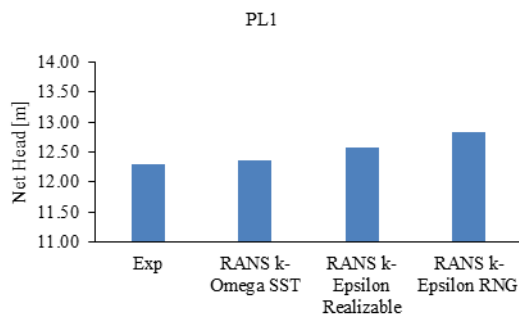


Figura 99. Valores de altura neta para diferentes modelos de turbulencia usando RapidCFD.

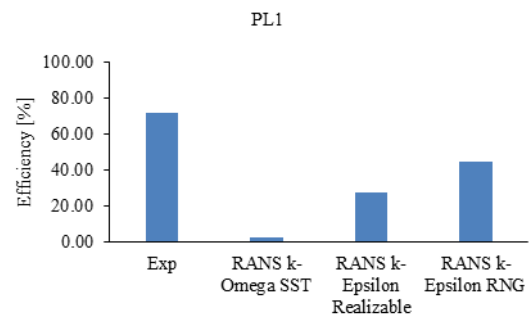


Figura 100. Valores de eficiencia hidráulica para diferentes modelos de turbulencia usando RapidCFD.

En la Figura 101 a Figura 104 solo se muestran los resultados de los modelos $k-\omega$ SST y $k-\varepsilon$ Realizable para la condición de operación de carga parcial PL2, ya que el modelo $k-\varepsilon$ RNG no presento convergencia a la solución. En general se puede observar que el modelo $k-\varepsilon$ Realizable presenta mejores aproximaciones.

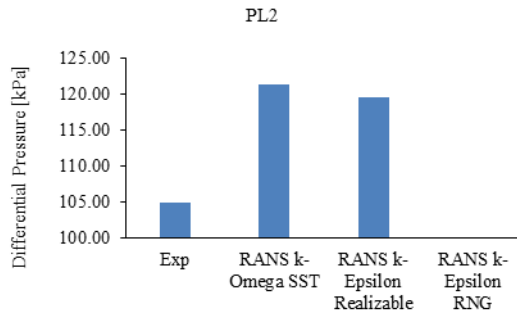


Figura 101. Valores de presión diferencial para diferentes modelos de turbulencia usando RapidCFD.

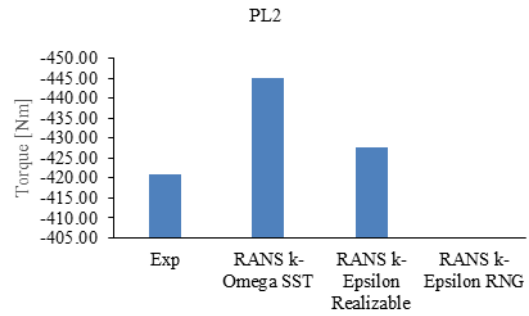


Figura 102. Valores de torque para diferentes modelos de turbulencia usando RapidCFD.

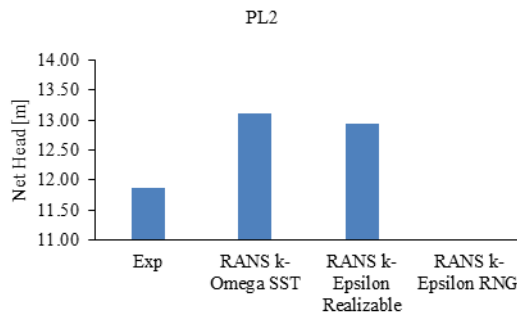


Figura 103. Valores de altura neta para diferentes modelos de turbulencia usando RapidCFD.

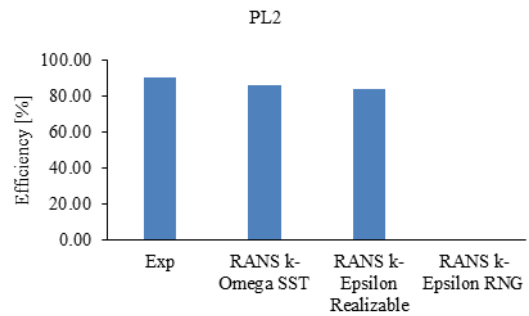


Figura 104. Valores de eficiencia hidráulica para diferentes modelos de turbulencia usando RapidCFD.

Los resultados para la condición de operación de eficiencia máxima BEP se muestran en la Figura 105 a Figura 108. Se observa que para esta condición de operación los mejores resultados se obtienen con el modelo $k-\varepsilon$ Realizable.

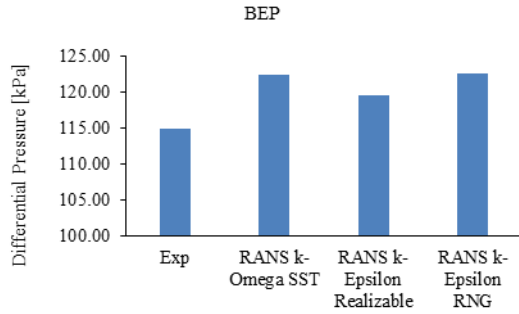


Figura 105. Valores de presión diferencial para diferentes modelos de turbulencia usando RapidCFD.

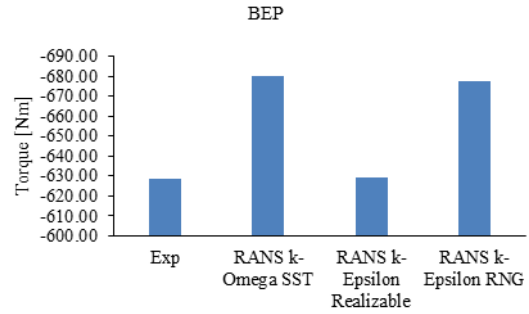


Figura 106. Valores de torque para diferentes modelos de turbulencia usando RapidCFD.

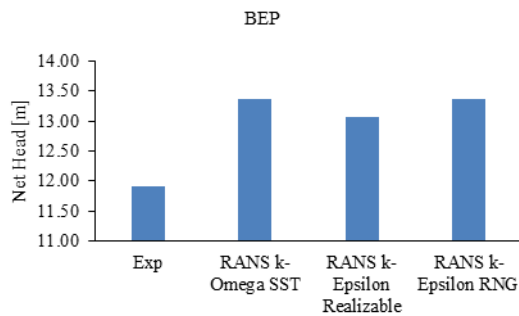


Figura 107. Valores de altura neta para diferentes modelos de turbulencia usando RapidCFD.

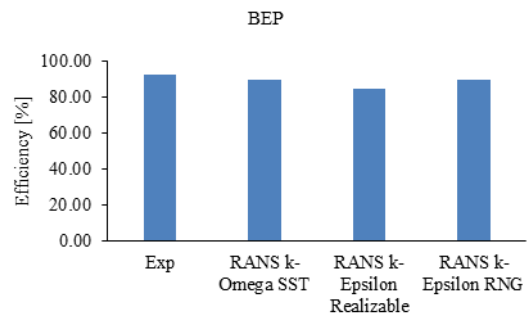


Figura 108. Valores de eficiencia hidráulica para diferentes modelos de turbulencia usando RapidCFD.

Para la condición de operación de eficiencia máxima BEP* se observa nuevamente que se obtienen resultados ligeramente mejores con el modelo $k-\epsilon$ Realizable como se muestra en la Figura 109 a Figura 112.

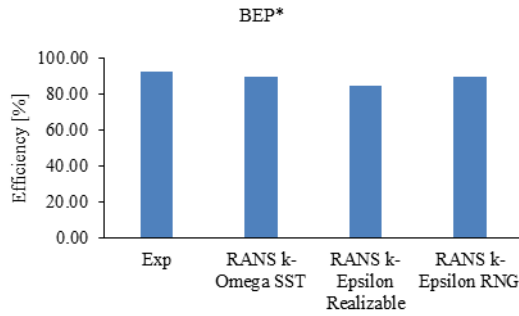


Figura 109. Valores de presión diferencial para diferentes modelos de turbulencia usando RapidCFD.

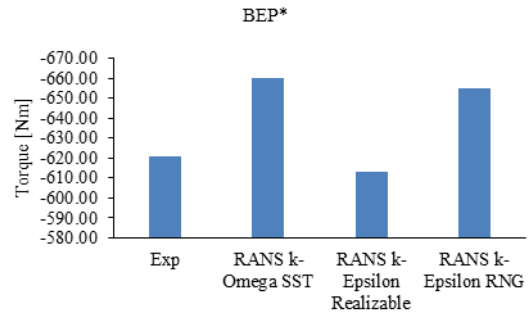


Figura 110. Valores de torque para diferentes modelos de turbulencia usando RapidCFD.

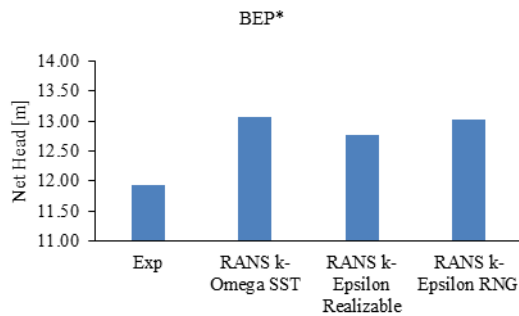


Figura 111. Valores de altura neta para diferentes modelos de turbulencia usando RapidCFD.

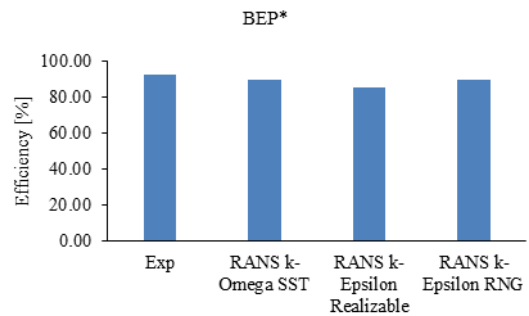


Figura 112. Valores de eficiencia hidráulica para diferentes modelos de turbulencia usando RapidCFD.

Como se muestra en la Figura 113 a Figura 116, el modelo $k-\epsilon$ *Realizable* genera resultados ligeramente mejores que sus contrapartes para la condición de operación de carga máxima HL, como ha sucedido en condiciones de operación previas.

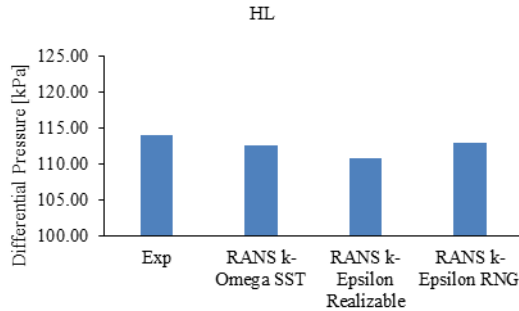


Figura 113. Valores de presión diferencial para diferentes modelos de turbulencia usando RapidCFD.

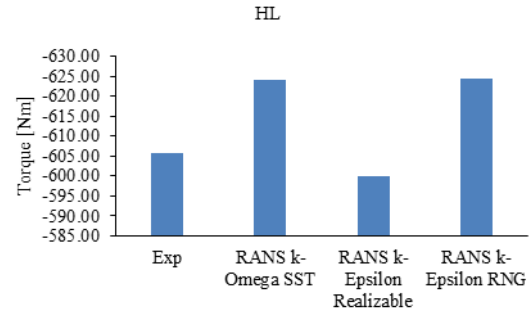


Figura 114. Valores de torque para diferentes modelos de turbulencia usando RapidCFD.

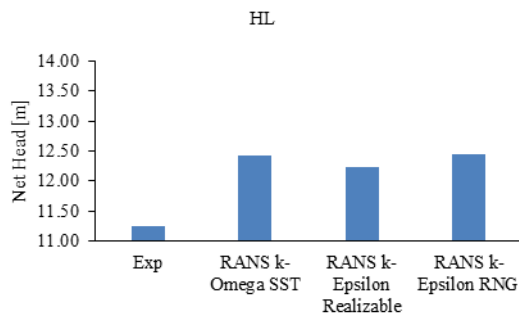


Figura 115. Valores de altura neta para diferentes modelos de turbulencia usando RapidCFD.

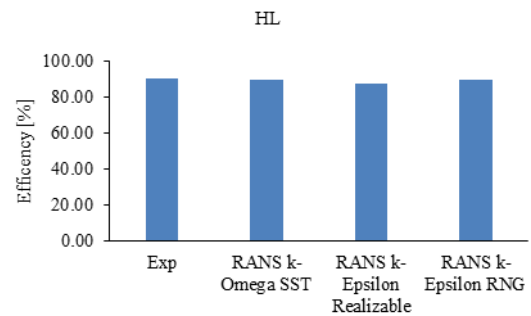


Figura 116. Valores de eficiencia hidráulica para diferentes modelos de turbulencia usando RapidCFD.

Finalmente la Figura 117 a Figura 120 muestran los resultados para la condición de operación de carga máxima HL*, donde el modelo $k-\epsilon$ Realizable ofrece resultados ligeramente mejores.

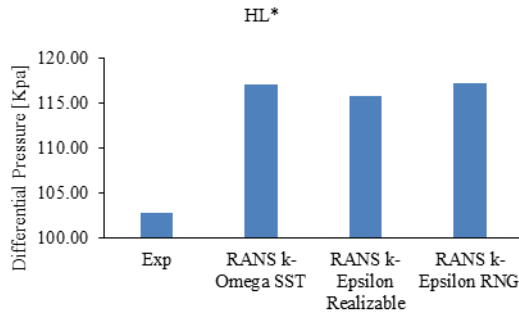


Figura 117. Valores de presión diferencial para diferentes modelos de turbulencia usando RapidCFD.

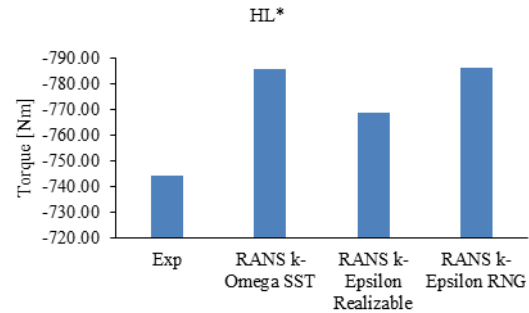


Figura 118. Valores de torque para diferentes modelos de turbulencia usando RapidCFD.

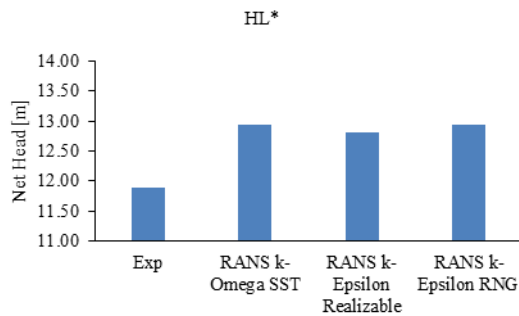


Figura 119. Valores de altura neta para diferentes modelos de turbulencia usando RapidCFD.

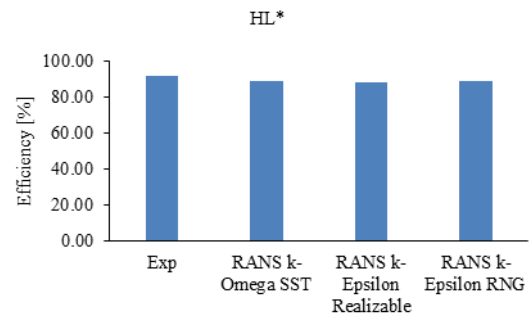


Figura 120. Valores de eficiencia hidráulica para diferentes modelos de turbulencia usando RapidCFD.

Perfiles de velocidad en estado estable

Una vez evaluados los resultados de las cantidades globales como lo son presión diferencial, torque, carga neta y eficiencia, se procedió a la validación de los perfiles de velocidad del fluido a la salida del rodete, justamente en el cono del tubo aspirador.

Los datos experimentales de velocidad axial (V) y horizontal (U) se obtuvieron mediante PIV en un plano situado a la salida del rodete, donde valores discretos se obtuvieron a lo largo de 3 líneas, dos de ellas horizontales paralelas y otra vertical. La Figura 121 y Figura 122 muestran los detalles y ubicación del plano utilizado para el PI, así como las líneas a lo largo de las cuales se realizaron las mediciones.

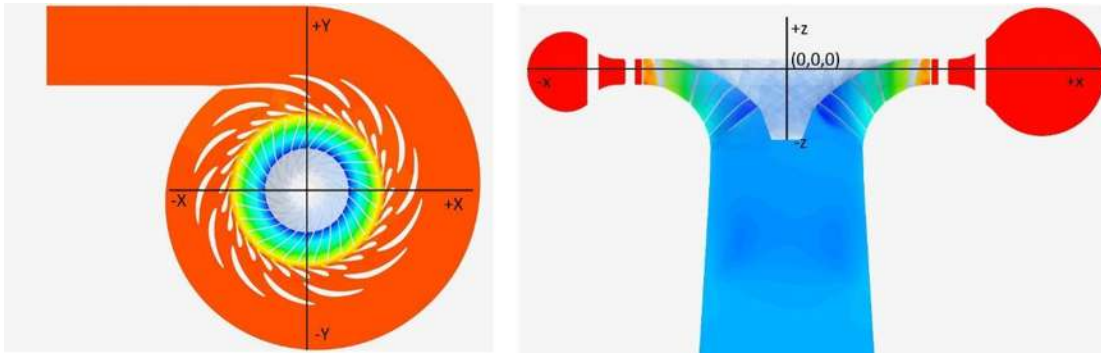


Figura 121. Coordenadas globales para las ubicaciones de medición, geometría y malla.

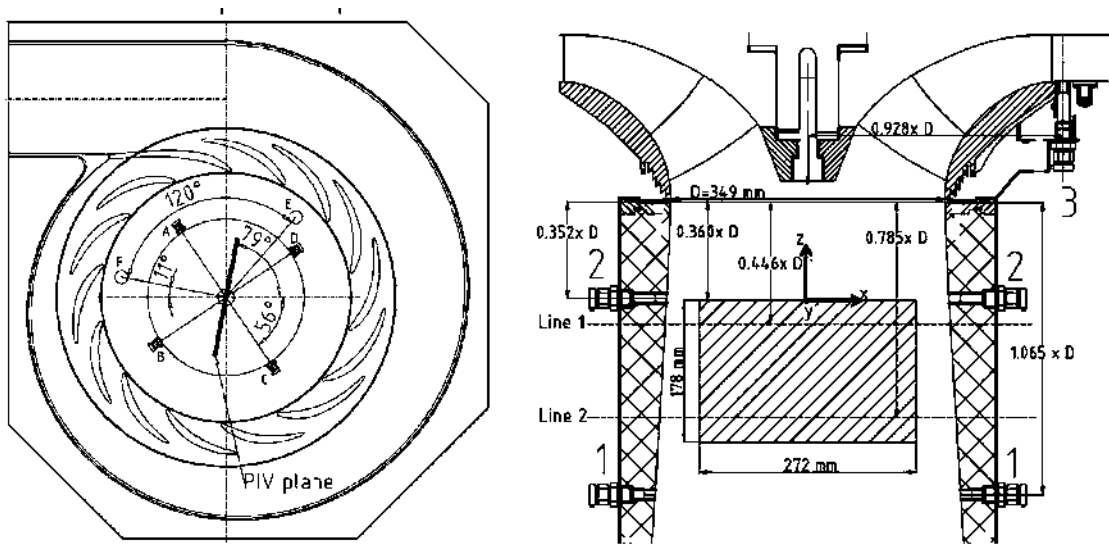


Figura 122. Secciones de medición mediante PIV en el cono del tubo de desfogeo.

La Figura 123 a la Figura 128 muestran los perfiles de velocidad obtenidos con los modelos de turbulencia $k-\omega$ SST y $k-\varepsilon$ Realizable para la condición de operación a carga parcial PL2, el modelo $k-\varepsilon$ RNG no produjo resultados. Se puede observar que entre ambos modelos existe muy poca diferencia en los perfiles de velocidad obtenidos, aunque ninguno de los dos logra una buena aproximación de la velocidad vertical en la parte media del cono. Se observa además que hay una marcada diferencia en la predicción de la velocidad horizontal en las zonas cercana a las paredes del cono.

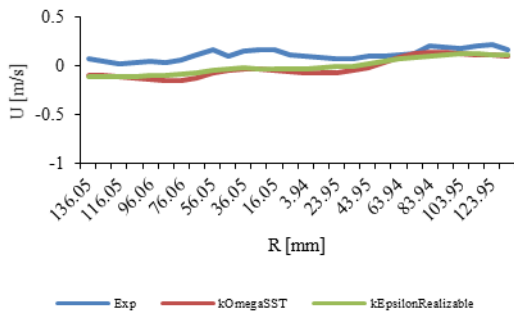


Figura 123. Valores de velocidad horizontal en PIV Line 1 con diferentes modelos de turbulencia a carga parcial PL2.

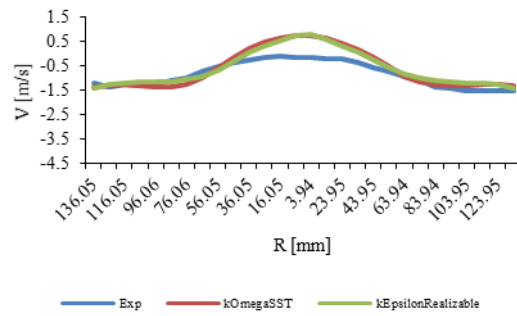


Figura 124. Valores de velocidad vertical en PIV Line 1 con diferentes modelos de turbulencia a carga parcial PL2.

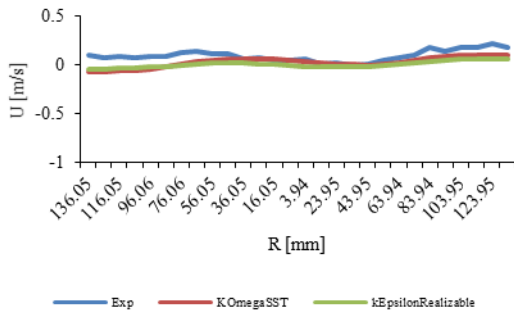


Figura 125. Valores de velocidad horizontal en PIV Line 2 con diferentes modelos de turbulencia a carga parcial PL2.

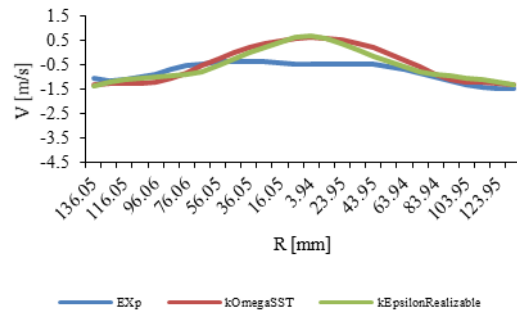


Figura 126. Valores de velocidad vertical en PIV Line 2 con diferentes modelos de turbulencia a carga parcial PL2.

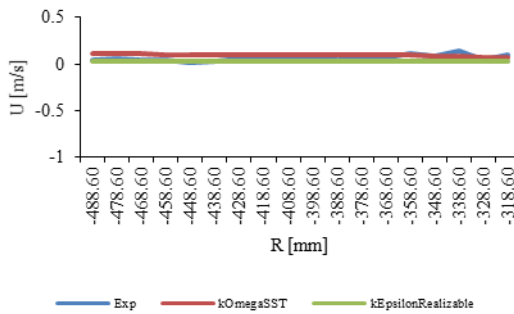


Figura 127. Valores de velocidad horizontal en PIV Line 3 con diferentes modelos de turbulencia a carga parcial PL2.

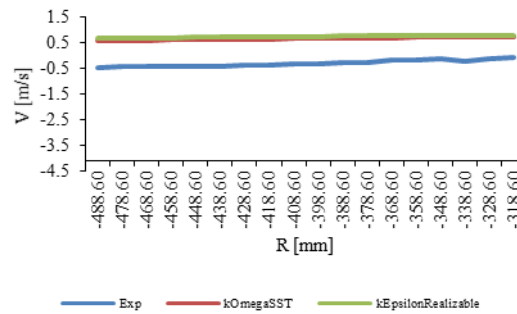


Figura 128. Valores de velocidad vertical en PIV Line 3 con diferentes modelos de turbulencia a carga parcial PL2.

Los perfiles de velocidad obtenidos por los tres modelos de turbulencia para la condición de operación de máxima eficiencia BEP* presentan una buena aproximación tanto para la

velocidad horizontal como vertical. Así mismo, existe poca diferencia entre las predicciones de los tres modelos como se muestra en la Figura 129 a la Figura 134.

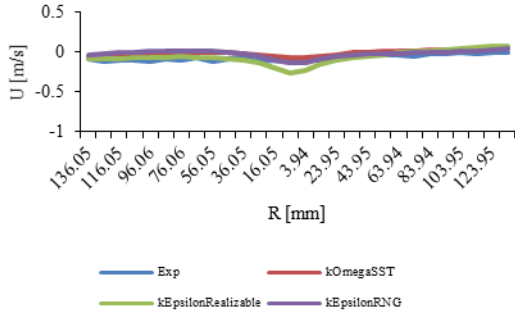


Figura 129. Valores de velocidad horizontal en PIV Line 1 con diferentes modelos de turbulencia en eficiencia máxima BEP*.

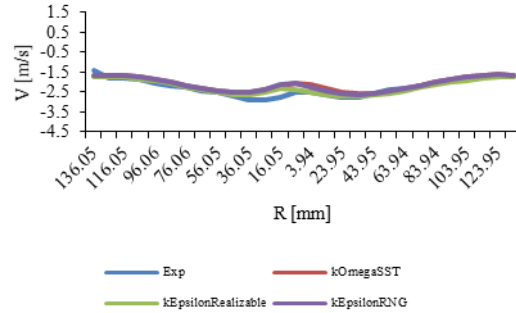


Figura 130. Valores de velocidad vertical en PIV Line 1 con diferentes modelos de turbulencia en eficiencia máxima BEP*.

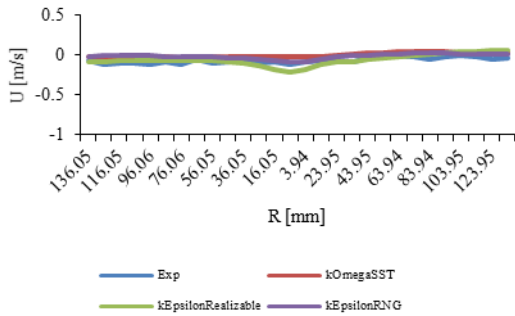


Figura 131. Valores de velocidad horizontal en PIV Line 2 con diferentes modelos de turbulencia en eficiencia máxima BEP*.

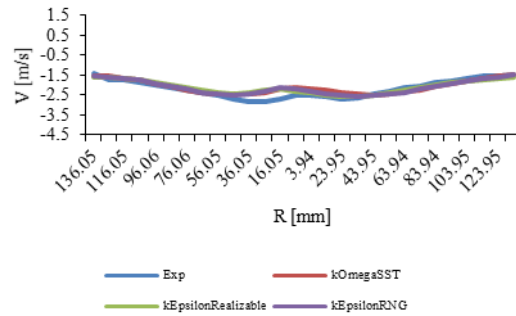


Figura 132. Valores de velocidad vertical en PIV Line 2 con diferentes modelos de turbulencia en eficiencia máxima BEP*.

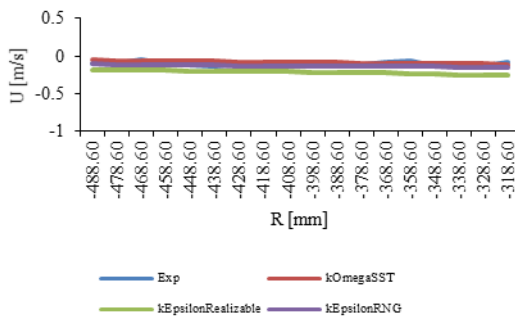


Figura 133. Valores de velocidad horizontal en PIV Line 3 con diferentes modelos de turbulencia en eficiencia máxima BEP*.

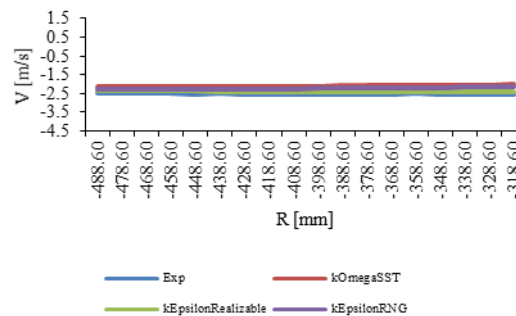


Figura 134. Valores de velocidad vertical en PIV Line 3 con diferentes modelos de turbulencia en eficiencia máxima BEP*.

En cuanto a la condición de operación de carga máxima HL* los perfiles obtenidos por los diferentes modelos de turbulencia presentan una buena aproximación cualitativa, sin embargo, ninguno de ellos genera buenas aproximaciones cuantitativas en las zonas centrales del cono tanto para la velocidad vertical como par la horizontal. El mismo comportamiento inexacto en la predicción de la velocidad horizontal se puede observar para las zonas cercanas la pared como muestran la Figura 135 a la Figura 140.

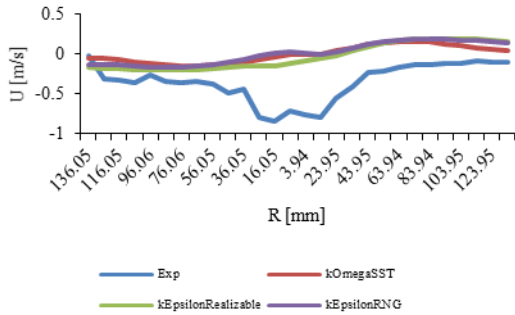


Figura 135. Valores de velocidad horizontal en PIV Line 1 con diferentes modelos de turbulencia a carga máxima HL*.

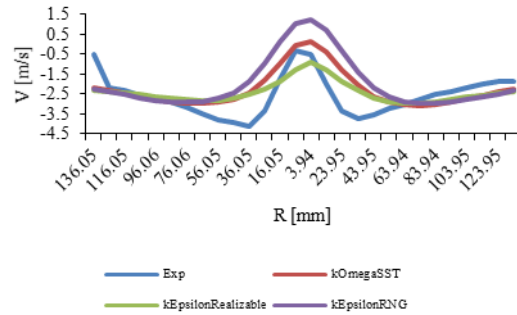


Figura 136. Valores de velocidad vertical en PIV Line 1 con diferentes modelos de turbulencia a carga máxima HL*.

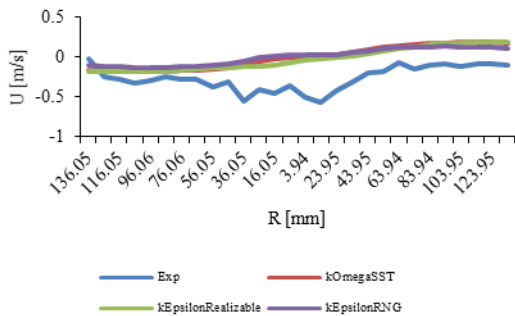


Figura 137. Valores de velocidad horizontal en PIV Line 2 con diferentes modelos de turbulencia a carga máxima HL*.

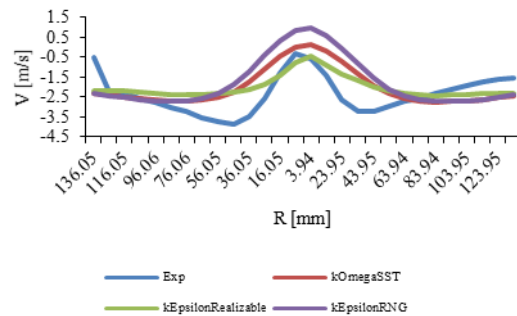


Figura 138. Valores de velocidad vertical en PIV Line 2 con diferentes modelos de turbulencia a carga máxima HL*.

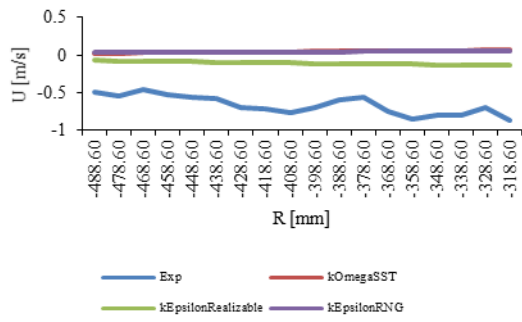


Figura 139. Valores de velocidad horizontal en PIV Line 3 con diferentes modelos de turbulencia a carga máxima HL*.

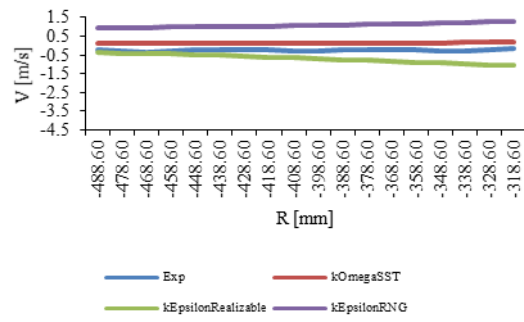


Figura 140. Valores de velocidad vertical en PIV Line 3 con diferentes modelos de turbulencia a carga máxima HL*.

Simulaciones URANS.

Cálculos en estado transitorio utilizando mallas deslizantes

Buscando obtener resultados más exactos, se realizaron cálculos adicionales utilizando esquemas de discretización de segundo orden además de incluir los términos temporales, es decir, se realizaron cálculos en estado transitorio con los mismos tres modelos de turbulencia empleado en la sección anterior. En estos nuevos cálculos no se utilizó la aproximación MRF, ya que este solo puede ser utilizada en estado estable, por lo que se hizo uso de la aproximación *sliding mesh* (mallas deslizantes) para considerar el giro del rodete. Se utilizó la aplicación de RapidCFD denominada *pimpleDymFoam*, la cual combina dentro de su programación los algoritmos de solución SIMPLE y *Pressure-Implicit with Splitting of Operators* (PISO) con el fin de utilizar intervalos de tiempo más grandes y mantener estabilidad en el proceso de cálculo. Para todas las condiciones de operación únicamente se calcularon 24° de giro del rodete, con un *time step* (intervalo de tiempo) promedio de aproximadamente 0.00005 segundos (varía de acuerdo a cada condición de operación) que corresponde a 1° de giro del rodete, en total se calcularon 24 intervalos de tiempo. Se utilizaron los resultados de las simulaciones en estado estable como condiciones iniciales con el fin de reducir el tiempo de cálculo, en cuanto a las condiciones de frontera estas permanecieron iguales que en los cálculos de la sección anterior.

Los resultados para la condición de operación de carga parcial PL1 una vez concluidos los 24 intervalos de tiempo se muestran en la Figura 141 a la Figura 144. Se observa que ninguno de los modelos de turbulencia genera buenas aproximaciones para la el torque y la eficiencia, además de que uno de los modelos de turbulencia no genero ninguna predicción.

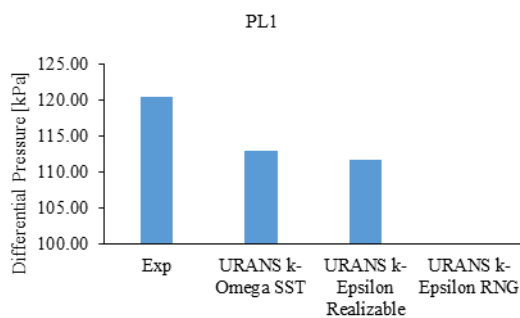


Figura 141. Valores de presión diferencial para diferentes modelos de turbulencia usando RapidCFD.

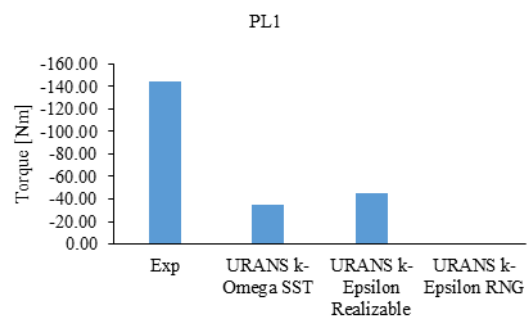


Figura 142. Valores de torque para diferentes modelos de turbulencia usando RapidCFD.

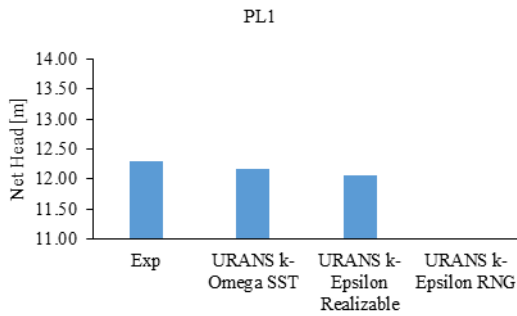


Figura 143. Valores de altura neta para diferentes modelos de turbulencia usando RapidCFD.

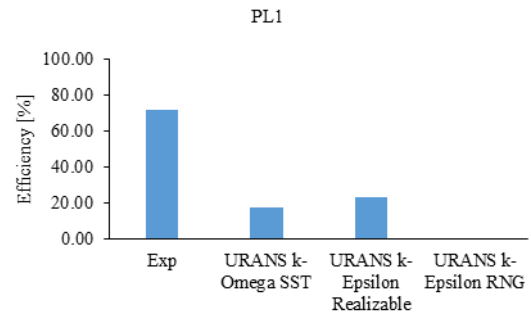


Figura 144. Valores de eficiencia hidráulica para diferentes modelos de turbulencia usando RapidCFD.

En cuanto a la condición de operación de carga parcial PL2, las predicciones son ligeramente mejores con el modelo $k-\epsilon$ *Realizable* como se muestra en la Figura 145 a la Figura 148, mismo comportamiento que los cálculos en estado estable.

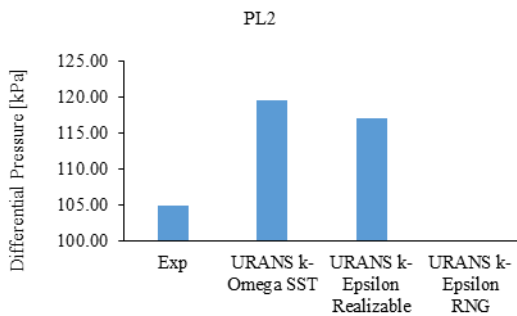


Figura 145. Valores de presión diferencial para diferentes modelos de turbulencia usando RapidCFD.

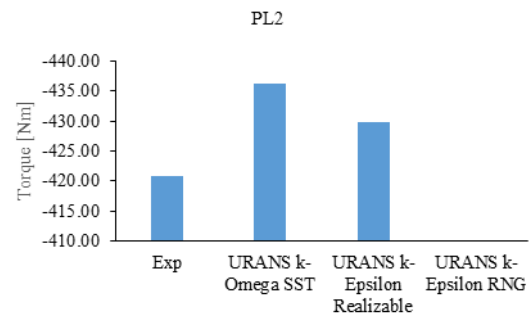


Figura 146. Valores de torque para diferentes modelos de turbulencia usando RapidCFD.

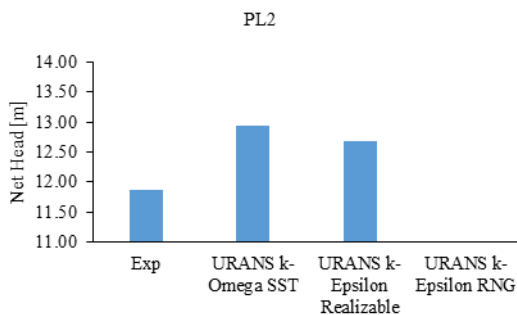


Figura 147. Valores de altura neta para diferentes modelos de turbulencia usando RapidCFD.

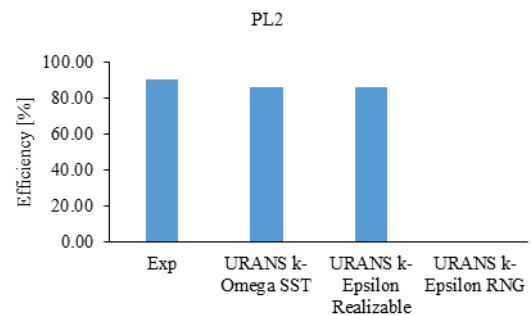


Figura 148. Valores de eficiencia hidráulica para diferentes modelos de turbulencia usando RapidCFD.

La Figura 149 a la Figura 152 muestran que el modelo $k-\epsilon$ *Realizable* genera las mejores aproximaciones para la condición de operación de eficiencia máxima BEP, comportamiento similar al obtenido en estado estable.

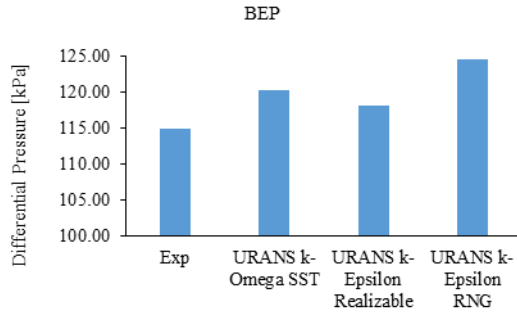


Figura 149. Valores de presión diferencial para diferentes modelos de turbulencia usando RapidCFD.

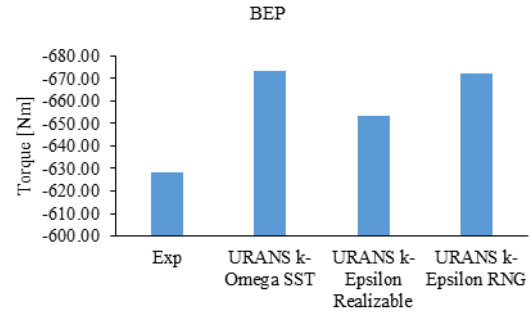


Figura 150. Valores de torque para diferentes modelos de turbulencia usando RapidCFD.

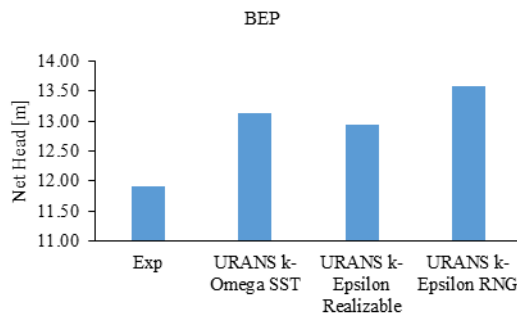


Figura 151. Valores de altura neta para diferentes modelos de turbulencia usando RapidCFD.

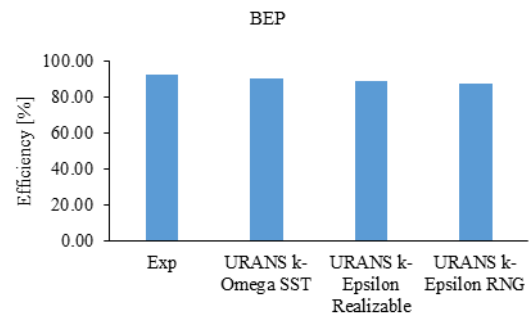


Figura 152. Valores de eficiencia hidráulica para diferentes modelos de turbulencia usando RapidCFD.

En cuanto a la condición de operación de eficiencia máxima BEP*, se puede observar en la Figura 153 a la Figura 156 que al igual que en estado estable, las mejores aproximaciones se obtienen con el modelo $k-\varepsilon$ Realizable.

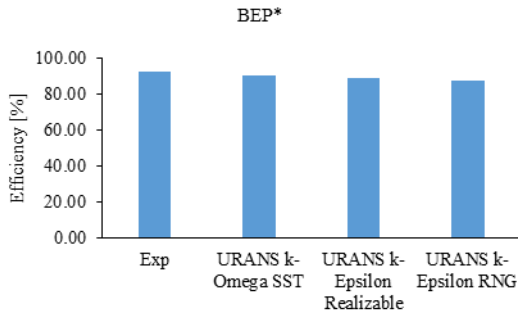


Figura 153. Valores de presión diferencial para diferentes modelos de turbulencia usando RapidCFD.

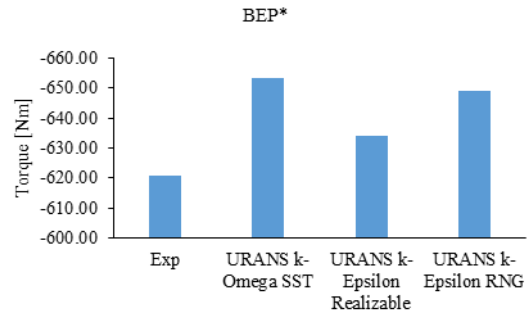


Figura 154. Valores de torque para diferentes modelos de turbulencia usando RapidCFD.

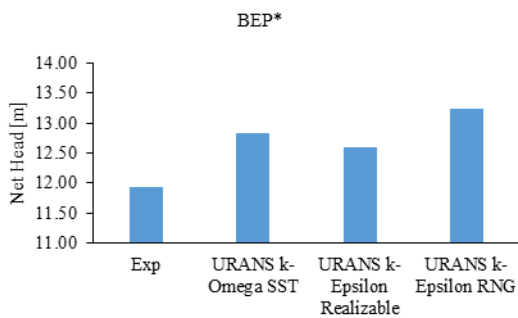


Figura 155. Valores de altura neta para diferentes modelos de turbulencia usando RapidCFD.

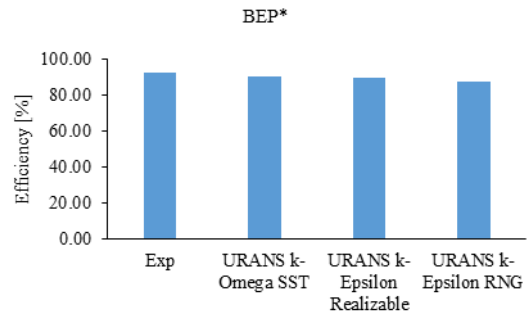


Figura 156. Valores de eficiencia hidráulica para diferentes modelos de turbulencia usando RapidCFD.

En la condición de operación de carga máxima HL en estado transitoria, al igual que estado estable, se observa que los resultados son ligeramente mejores con el modelo $k-\varepsilon$ Realizable como se muestra en la Figura 157 a la Figura 160.

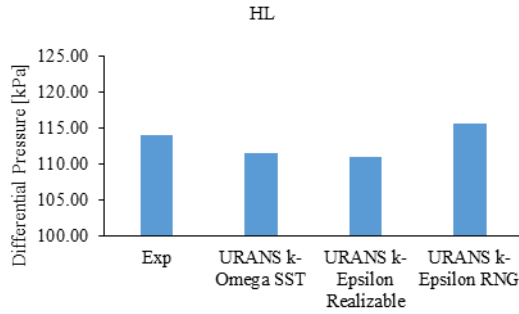


Figura 157. Valores de presión diferencial para diferentes modelos de turbulencia usando RapidCFD.

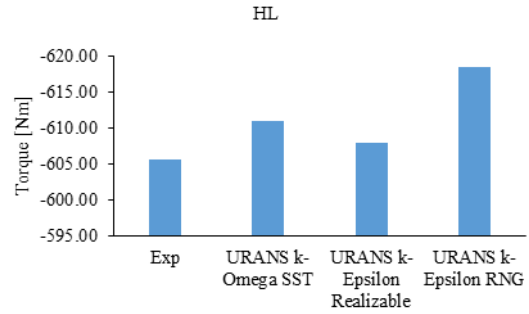


Figura 158. Valores de torque para diferentes modelos de turbulencia usando RapidCFD.

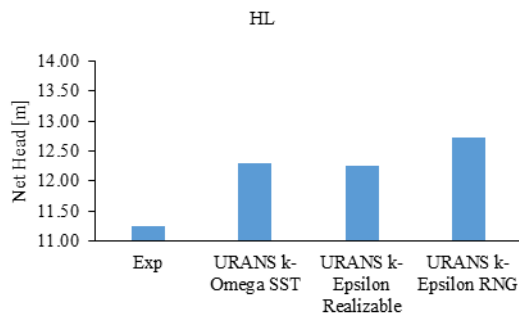


Figura 159. Valores de altura neta para diferentes modelos de turbulencia usando RapidCFD.

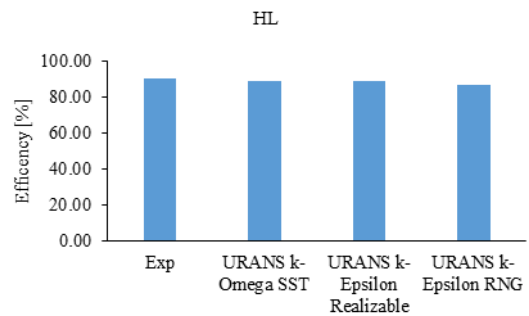


Figura 160. Valores de eficiencia hidráulica para diferentes modelos de turbulencia usando RapidCFD.

Finalmente para la condición de operación de carga máxima HL*, se observa que el modelo $k-\omega$ SST presenta aproximaciones ligeramente mejores al resto de los modelos de turbulencia, a diferencia de los que se había observado hasta el momento. Lo antes descrito se muestra en la Figura 161 a la Figura 164.

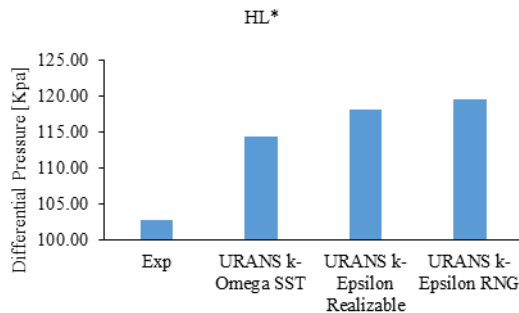


Figura 161. Valores de presión diferencial para diferentes modelos de turbulencia usando RapidCFD.

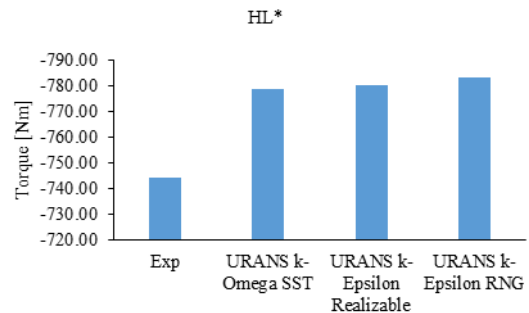


Figura 162. Valores de torque para diferentes modelos de turbulencia usando RapidCFD.

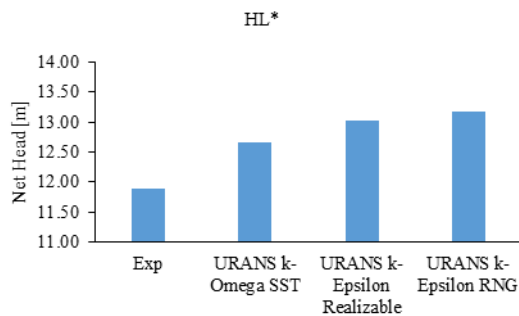


Figura 163. Valores de altura neta para diferentes modelos de turbulencia usando RapidCFD.

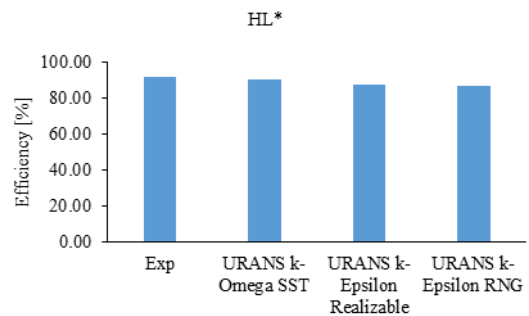


Figura 164. Valores de eficiencia hidráulica para diferentes modelos de turbulencia usando RapidCFD.

El utilizar los resultados de los cálculos en estado transitorio después de solo haber girado 24° el rodete podría parecer inapropiado, ya que los cálculos en estado transitorio generalmente requieren que los elementos rotativos realicen dos o tres giros completos para alcanzar un comportamiento cíclico. Sin embargo como se muestra más adelante, el haber inicializado los cálculos con los resultados obtenidos en estado estable permitió reducir el proceso de cálculo.

Solución transitoria utilizando el modelo de turbulencia $k-\omega$ SST

Las Figura 165 a la Figura 168 muestran el comportamiento de los resultados para cada variable a lo largo de cada intervalo de tiempo calculado correspondiente a la condición de operación de carga parcial PL1 utilizando el modelo $k-\omega$ SST. Se puede observar que tanto

la presión diferencial como la carga neta alcanzaron de inmediato valores oscilatorios cercanos a los datos experimentales, sin embargo el torque y la eficiencia presentan una tendencia ascendente después de 24° de giro, por lo que para esta condición probablemente se requiera un tiempo extra de cálculo.

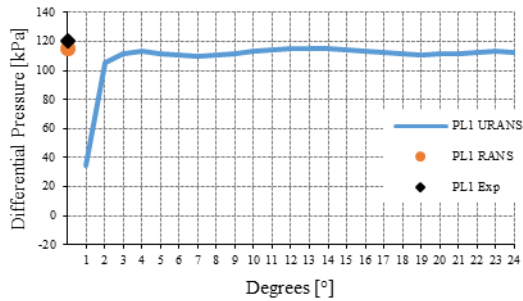


Figura 165. Valores de presión diferencial para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega$ SST en condición de carga parcial PL1.

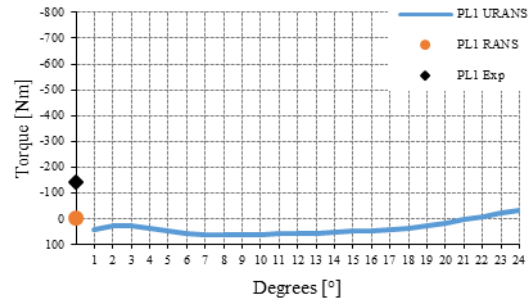


Figura 166. Valores de torque para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega$ SST en condición de carga parcial PL1.

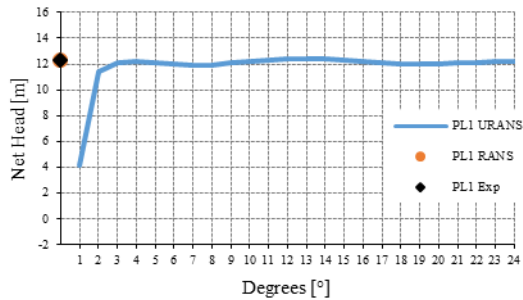


Figura 167. Valores de altura neta para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega$ SST en condición de carga parcial PL1.

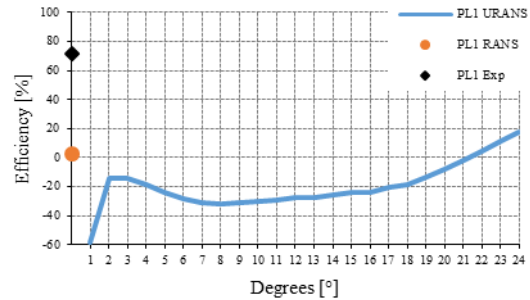


Figura 168. Valores de eficiencia hidráulica para para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega$ SST en condición de carga parcial PL1.

En la condición de operación de carga parcial PL2 con el modelo $k-\omega SST$ se observa que todas las variables mantienen un comportamiento oscilatorio después de algunos intervalos de tiempo como muestran la Figura 169 a la Figura 172.

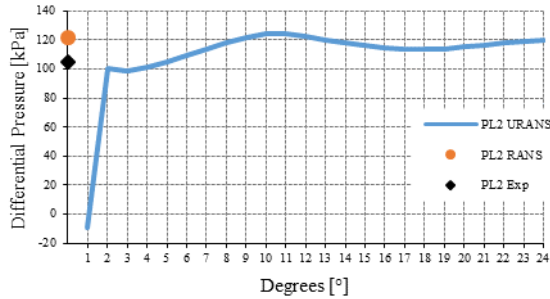


Figura 169. Valores de presión diferencial para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega SST$ en condición de carga parcial PL2.

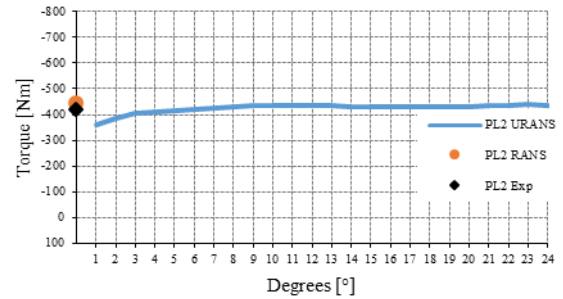


Figura 170. Valores de torque para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega SST$ en condición de carga parcial PL2.

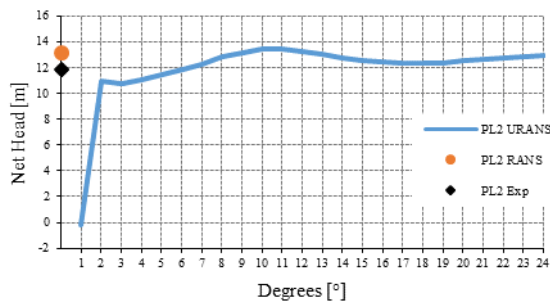


Figura 171. Valores de altura neta para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega SST$ en condición de carga parcial PL2.

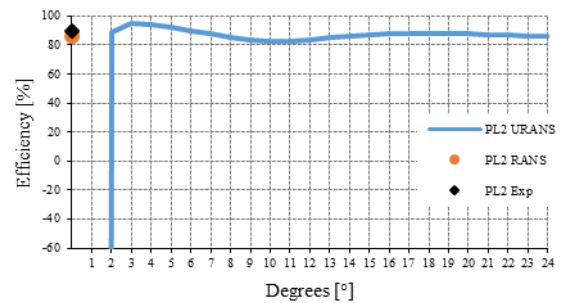


Figura 172. Valores de eficiencia hidráulica para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega SST$ en condición de carga parcial PL2.

En la condición de operación de eficiencia máxima BEP según se muestra en la Figura 173 a la Figura 176 prácticamente después de dos intervalos de tiempo se llega a valores cercanos a los experimentales utilizando el modelo $k-\omega$ SST.

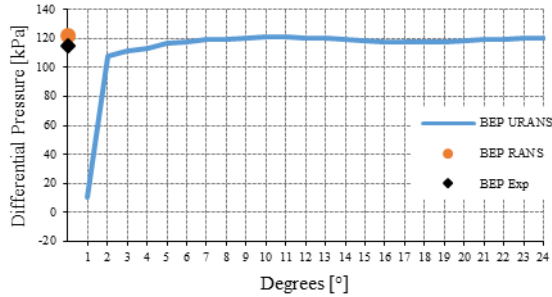


Figura 173. Valores de presión diferencial para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega$ SST en condición de eficiencia máxima BEP.

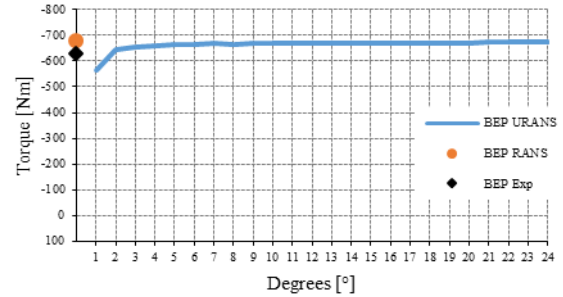


Figura 174. Valores de torque para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega$ SST en condición de eficiencia máxima BEP.

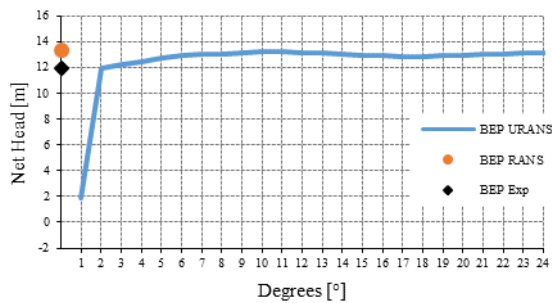


Figura 175. Valores de altura neta para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega$ SST en condición de eficiencia máxima BEP.

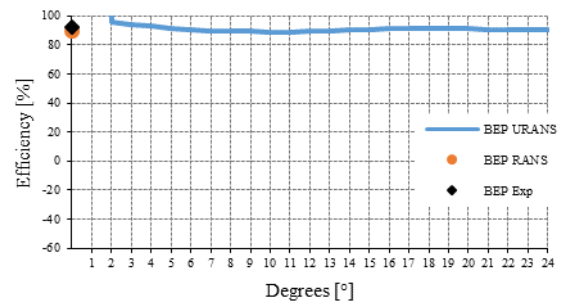


Figura 176. Valores de eficiencia hidráulica para para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega$ SST en condición de eficiencia máxima BEP.

Un comportamiento similar se observa para la condición de carga máxima BEP* al utilizar el modelo de turbulencia $k-\omega SST$ en la Figura 177 a la Figura 180.

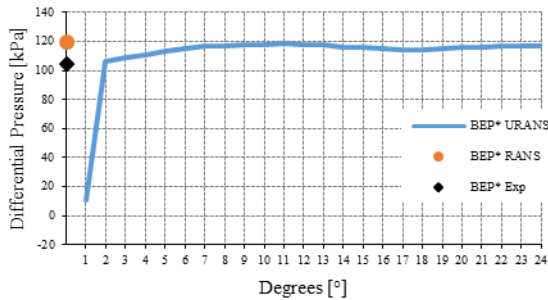


Figura 177. Valores de presión diferencial para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega SST$ en condición de eficiencia máxima BEP*.

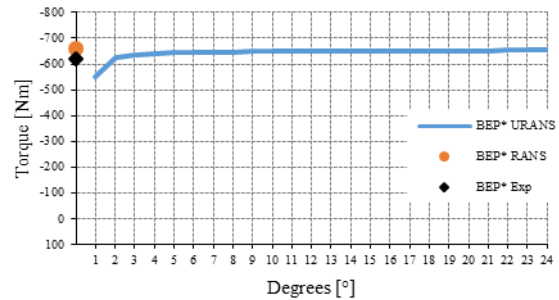


Figura 178. Valores de torque para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega SST$ en condición de eficiencia máxima BEP*.

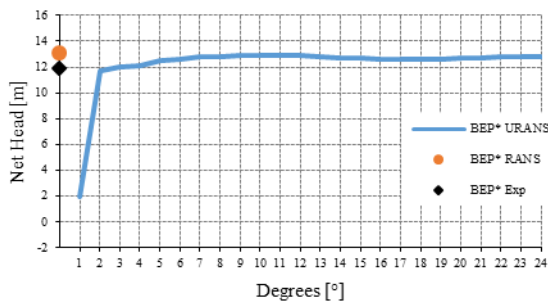


Figura 179. Valores de altura neta para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega SST$ en condición de eficiencia máxima BEP*.

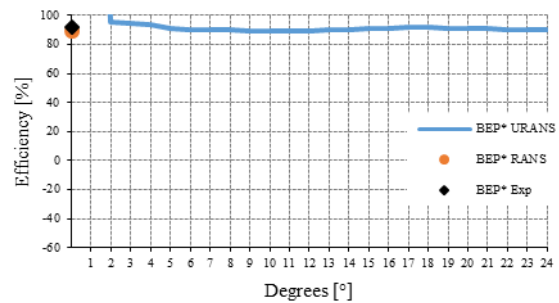


Figura 180. Valores de eficiencia hidráulica para para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega SST$ en condición de eficiencia máxima BEP*.

En la condición de carga máxima HL se observa que para alcanzar valores cercanos a los experimentales deben pasar en seis y siete intervalos de tiempo ya que la presión diferencial no es calculada de manera precisa al inicio de los cálculos como se muestra en Figura 181 a la Figura 184.

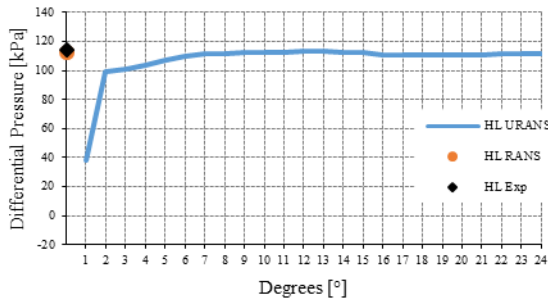


Figura 181. Valores de presión diferencial para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega$ SST en condición de carga máxima HL.

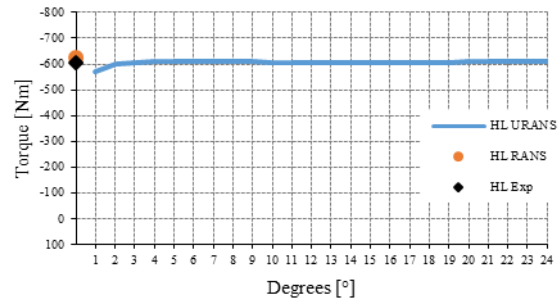


Figura 182. Valores de torque para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega$ SST en condición de carga máxima HL.

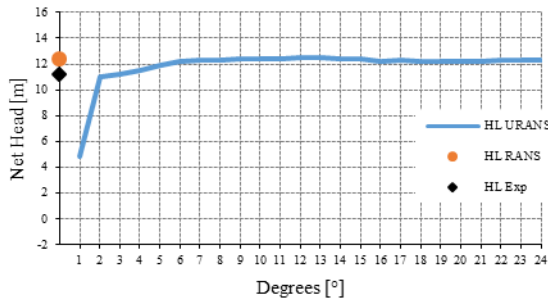


Figura 183. Valores de altura neta para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega$ SST en condición de carga máxima HL.

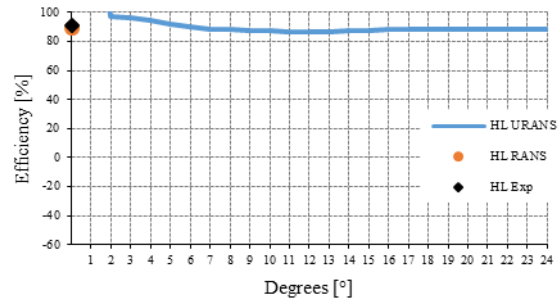


Figura 184. Valores de eficiencia hidráulica para para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega$ SST en condición de carga máxima HL.

En el caso de la condición de carga máxima HL* el cálculo de la presión diferencial presenta un comportamiento un poco más estable como se muestra en la Figura 185 a la Figura 188.

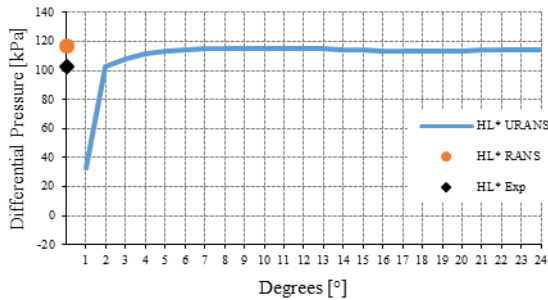


Figura 185. Valores de presión diferencial para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega$ SST en condición de carga máxima HL*.

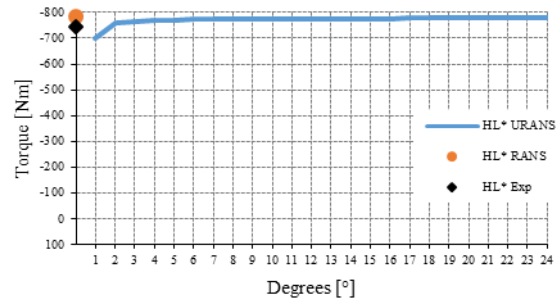


Figura 186. Valores de torque para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega$ SST en condición de carga máxima HL*.

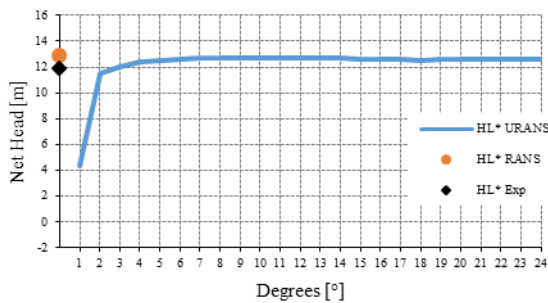


Figura 187. Valores de altura neta para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega$ SST en condición de carga máxima HL*.

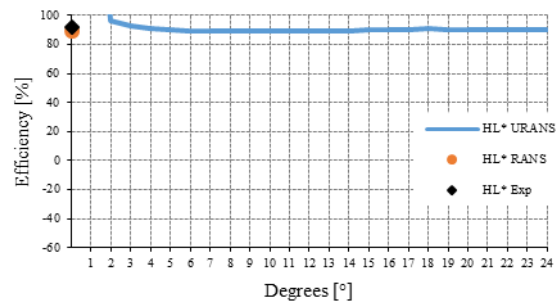


Figura 188. Valores de eficiencia hidráulica para para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\omega$ SST en condición de carga máxima HL*.

Solución transitoria utilizando el modelo de turbulencia $k-\epsilon$ Realizable

El segundo modelo de turbulencia utilizado fue el modelo épsilon Realizable, el cual ha demostrado generar resultados ligeramente más aproximados a datos experimentales que el resto de los modelos probados. En la Figura 189 a la Figura 192 se puede observar que al igual que con el uso del modelo $k-\omega$ SST, los valores de torque son calculados de manera deficiente, lo que impacta en los valores de eficiencia hidráulica.

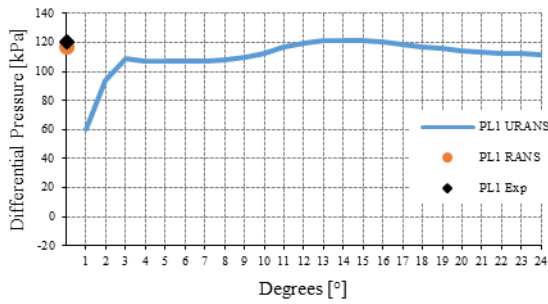


Figura 189. Valores de presión diferencial para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ Realizable en condición de carga parcial PL1.

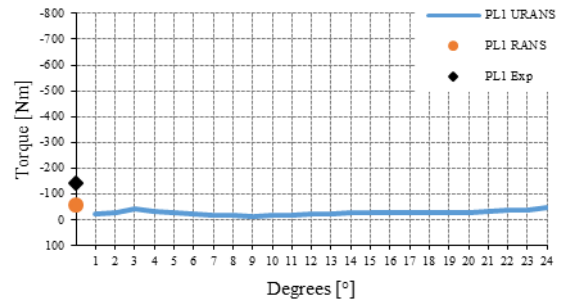


Figura 190. Valores de torque para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ Realizable en condición de carga parcial PL1.

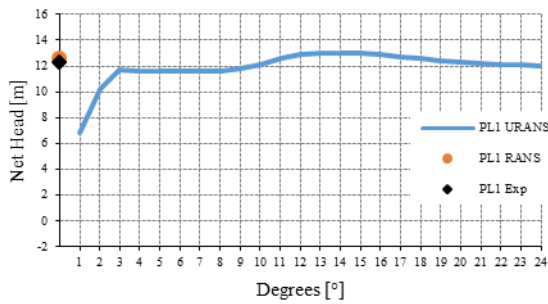


Figura 191. Valores de altura neta para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ Realizable en condición de carga parcial PL1.

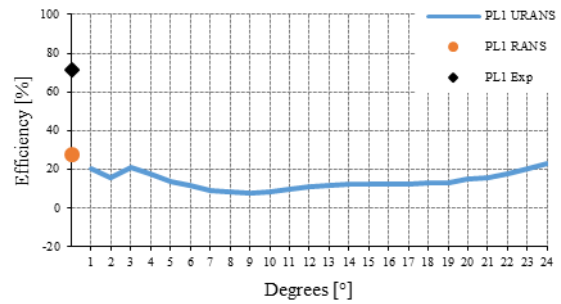


Figura 192. Valores de eficiencia hidráulica para para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ Realizable en condición de carga parcial PL1.

Al utilizar el modelo de turbulencia $k-\varepsilon$ *Realizable* para la condición de operación de carga parcial PL2 se observa un comportamiento inestable en los resultados al inicio del proceso de cálculo, sin embargo después de seis o siete intervalos de tiempo se estabilizan como se muestra en la Figura 193 a la Figura 196.

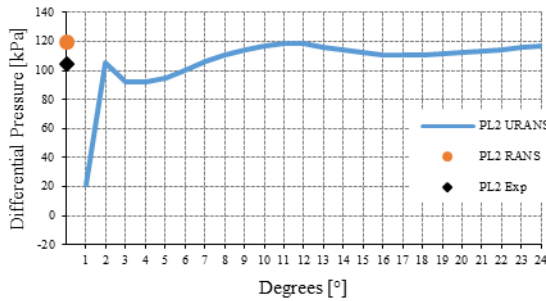


Figura 193. Valores de presión diferencial para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ *Realizable* en condición de carga parcial PL2.

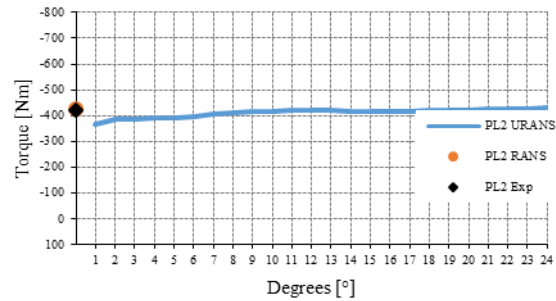


Figura 194. Valores de torque para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ *Realizable* en condición de carga parcial PL2.

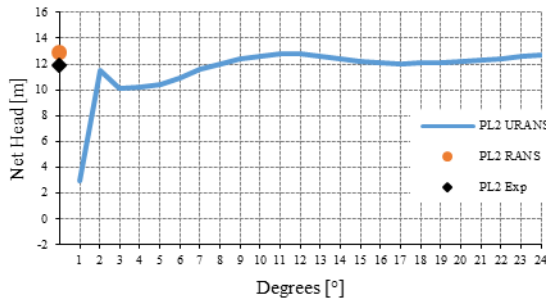


Figura 195. Valores de altura neta para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ *Realizable* en condición de carga parcial PL2.

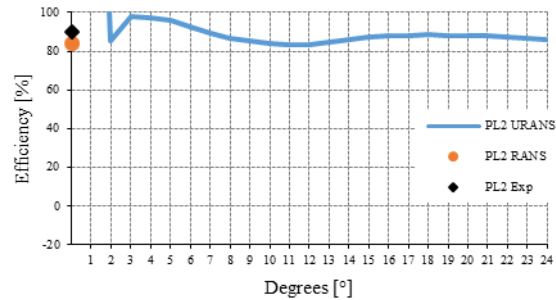


Figura 196. Valores de eficiencia hidrulica para para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ *Realizable* en condición de carga parcial PL2.

En la condición de máxima eficiencia se observa también un inicio inestable en los resultados sin embargo después de 4 o 5 intervalos de tiempo se estabilizan como puede apreciarse en la Figura 197 a la Figura 200.

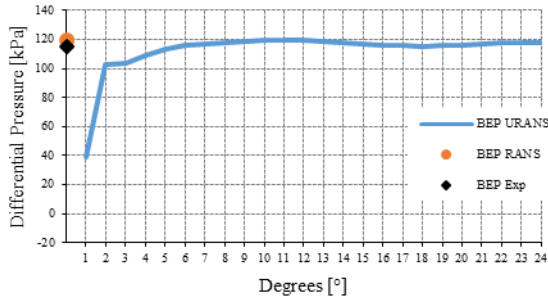


Figura 197. Valores de presión diferencial para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ Realizable en condición de eficiencia máxima BEP.

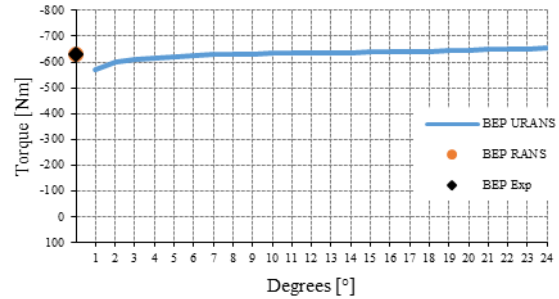


Figura 198. Valores de torque para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ Realizable en condición de eficiencia máxima BEP.

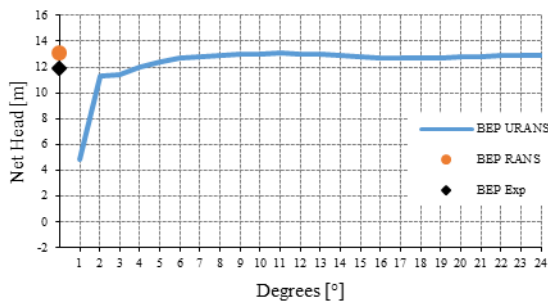


Figura 199. Valores de altura neta para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ Realizable en condición de eficiencia máxima BEP.

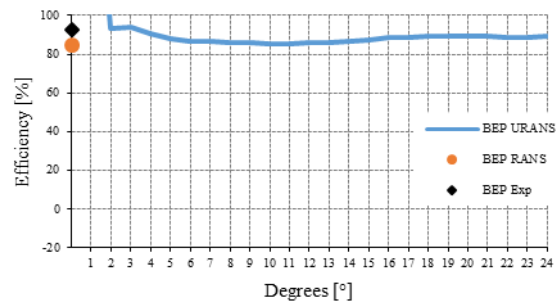


Figura 200. Valores de eficiencia hidráulica para para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ Realizable en condición de eficiencia máxima BEP.

Los resultados para la condición de operación de eficiencia máxima BEP* se muestran en la Figura 201 a la Figura 204 donde se observa un comportamiento similar a la condición BEP (de parámetros similares).

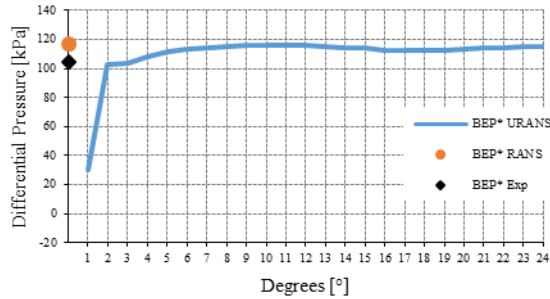


Figura 201. Valores de presión diferencial para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ Realizable en condición de eficiencia máxima BEP*.

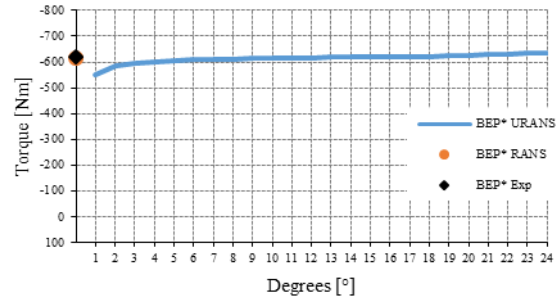


Figura 202. Valores de torque para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ Realizable en condición de eficiencia máxima BEP*.

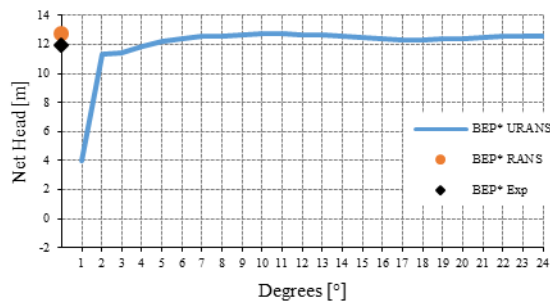


Figura 203. Valores de altura neta para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ Realizable en condición de eficiencia máxima BEP*.

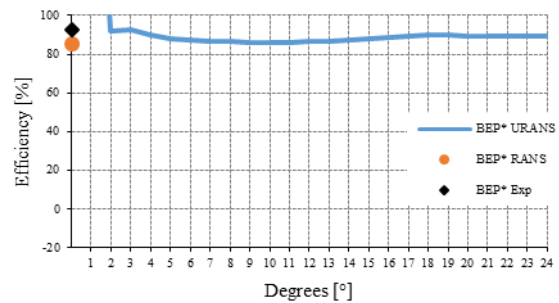


Figura 204. Valores de eficiencia hidráulica para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ Realizable en condición de eficiencia máxima BEP*.

Los resultados mostrados en la Figura 205 a la Figura 208 para la condición de operación de carga máxima HL utilizando el modelo $k-\varepsilon$ *Realizable* presenta un comportamiento parecido al modelo $k-\omega$ *SST*, sin embargo como se mostró previamente, los resultados del primero son mejores.

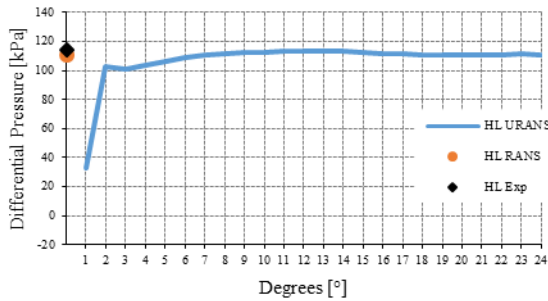


Figura 205. Valores de presión diferencial para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ *Realizable* en condición de carga máxima HL.

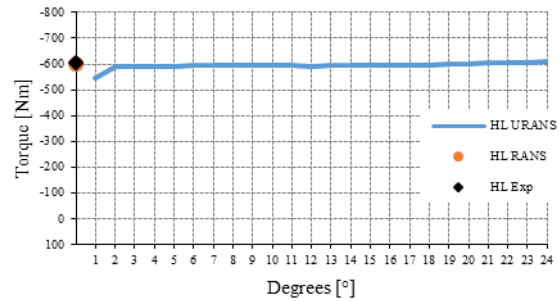


Figura 206. Valores de torque para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ *Realizable* en condición de carga máxima HL.

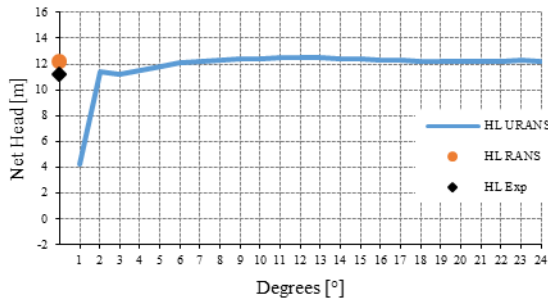


Figura 207. Valores de altura neta para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ *Realizable* en condición de carga máxima HL.

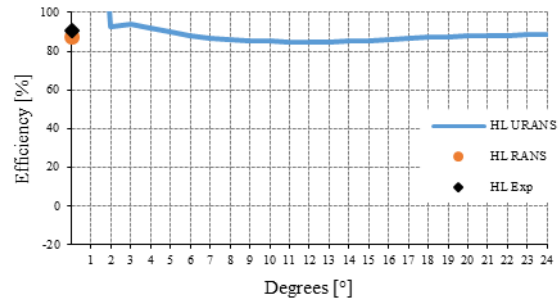


Figura 208. Valores de eficiencia hidráulica para para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ *Realizable* en condición de carga máxima HL.

Para finalizar con el modelo $k-\varepsilon$ *Realizable*, la Figura 209 a la Figura 212 muestran los resultados para la condición de operación de carga máxima HL*. Se puede observar un comportamiento un poco más estable que la condición de operación previa.

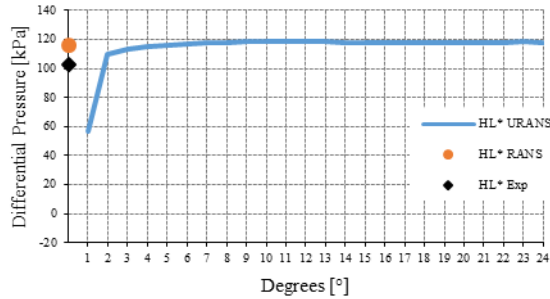


Figura 209. Valores de presión diferencial para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ *Realizable* en condición de carga máxima HL*.

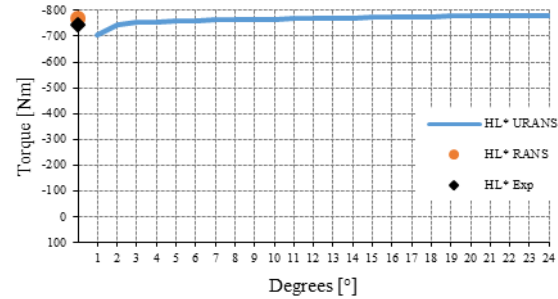


Figura 210. Valores de torque para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ *Realizable* en condición de carga máxima HL*.

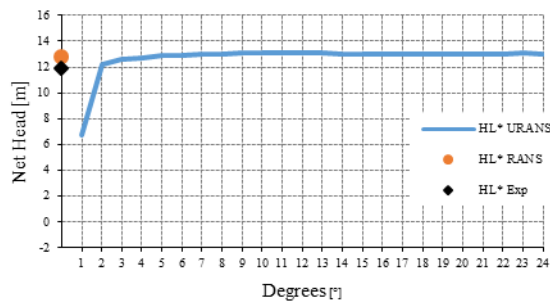


Figura 211. Valores de altura neta para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ *Realizable* en condición de carga máxima HL*.

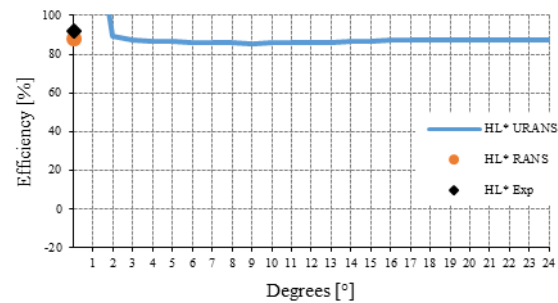


Figura 212. Valores de eficiencia hidrúlica para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ *Realizable* en condición de carga máxima HL*.

Solución transitoria utilizando el modelo de turbulencia $k-\varepsilon$ *RNG*

Los resultados obtenidos con el tercer modelo de turbulencia, $k-\varepsilon$ *RNG*, se muestran a continuación. Se debe mencionar que este modelo de turbulencia no generó resultados para las condiciones de operación de carga parcial PL1 y PL2. En el caso de la primera condición de operación mencionada a pesar de que se tenía una solución inicial en estado estable, esta no logró convergencia en ninguno de los intervalos de tiempo (posición del rodete) establecidos. Para la segunda condición de operación no se contó con datos para inicializar el cálculo por lo que este no se realizó.

La Figura 213 a la Figura 216 muestran los resultados obtenidos para la condición de operación de eficiencia máxima, donde se puede observar que después de un par de intervalos de tiempo los resultados se estabilizan.

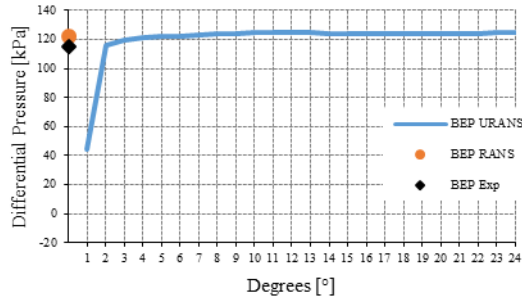


Figura 213. Valores de presión diferencial para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ RNG en condición de eficiencia máxima BEP.

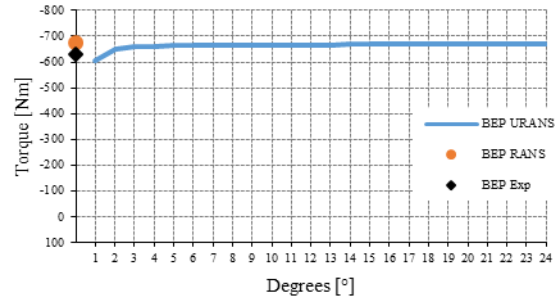


Figura 214. Valores de torque para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ RNG en condición de eficiencia máxima BEP.

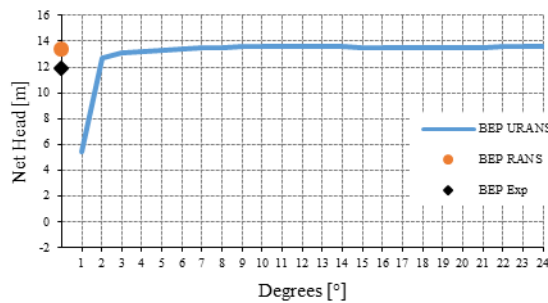


Figura 215. Valores de altura neta para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ RNG en condición de eficiencia máxima BEP.

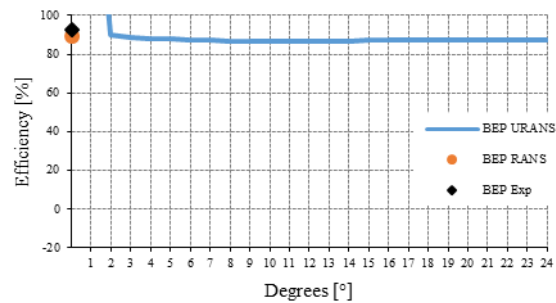


Figura 216. Valores de eficiencia hidráulica para para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ RNG en condición de eficiencia máxima BEP.

Al evaluar los resultados de la condición de operación de eficiencia máxima BEP^* con el modelo $k-\varepsilon RNG$, se puede observar en la Figura 217 a la Figura 220 que los valores obtenidos se estabilizan después de dos o tres intervalos de tiempo.

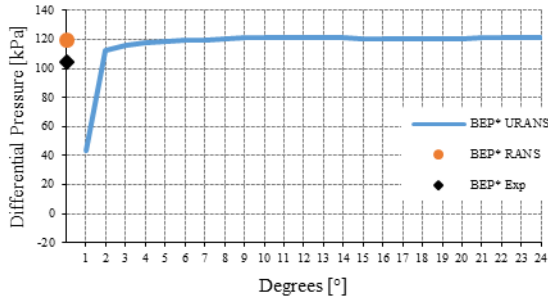


Figura 217. Valores de presión diferencial para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon RNG$ en condición de eficiencia máxima BEP^* .

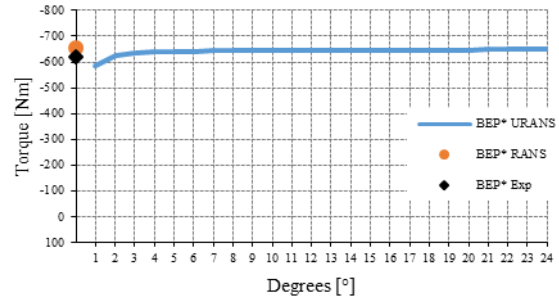


Figura 218. Valores de torque para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon RNG$ en condición de eficiencia máxima BEP^* .

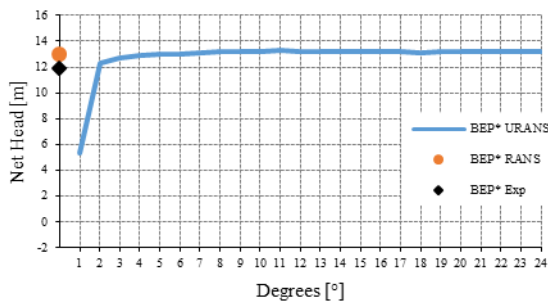


Figura 219. Valores de altura neta para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon RNG$ en condición de eficiencia máxima BEP^* .

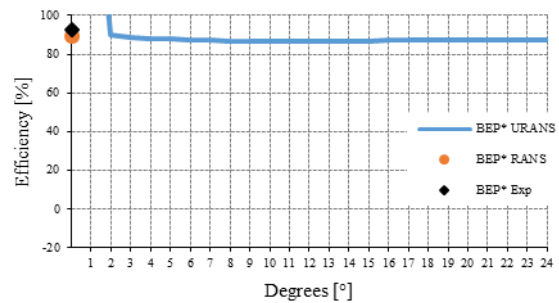


Figura 220. Valores de eficiencia hidráulica para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon RNG$ en condición de eficiencia máxima BEP^* .

Para la condición de operación de carga máxima HL se observa en la Figura 221 a la Figura 224 que los resultados son prácticamente estables después de seis o siete intervalos de tiempo al usar el modelo de turbulencia $k-\varepsilon RNG$.

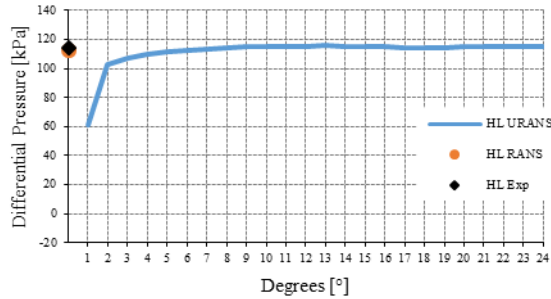


Figura 221. Valores de presión diferencial para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon RNG$ en condición de carga máxima HL.

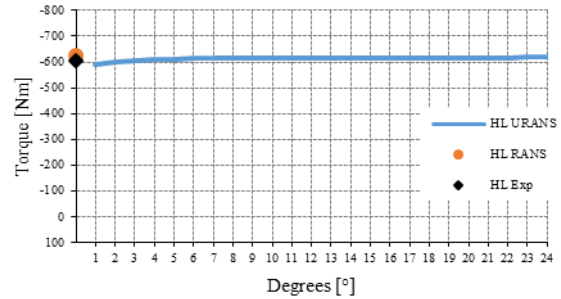


Figura 222. Valores de torque para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon RNG$ en condición de carga máxima HL.

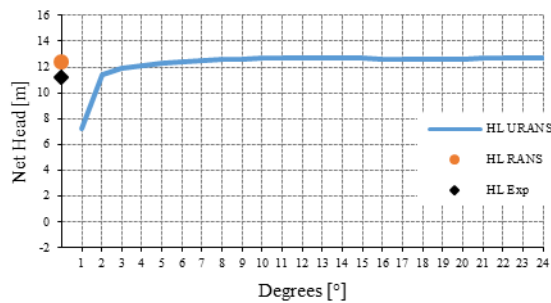


Figura 223. Valores de altura neta para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon RNG$ en condición de carga máxima HL.

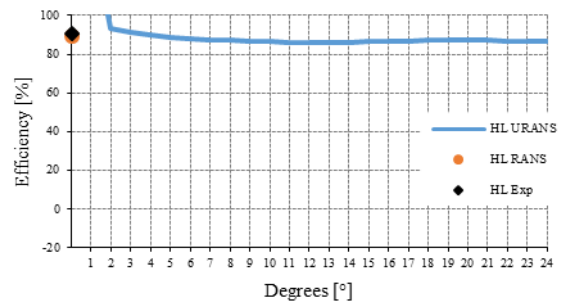


Figura 224. Valores de eficiencia hidráulica para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon RNG$ en condición de carga máxima HL.

Finalmente los resultados obtenidos utilizando el modelo $k-\varepsilon$ RNG para la condición de operación de carga máxima HL* se muestran en la Figura 225 a la Figura 228, donde se observa comportamiento similar a la condición de operación previa y al resto de los modelos de turbulencia.

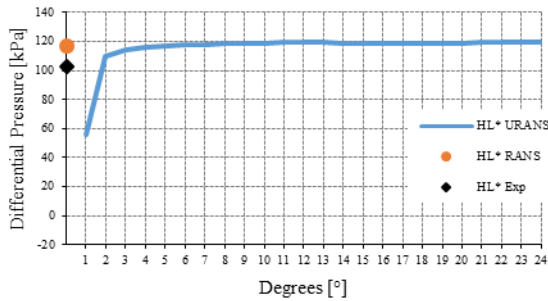


Figura 225. Valores de presión diferencial para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ RNG en condición de carga máxima HL*.

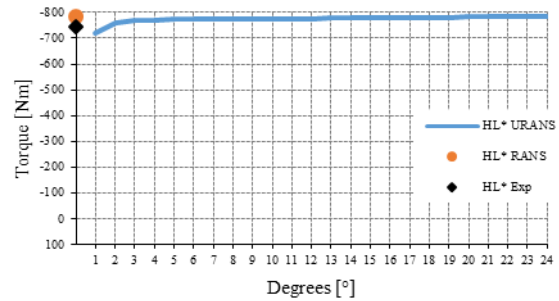


Figura 226. Valores de torque para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ RNG en condición de carga máxima HL*.

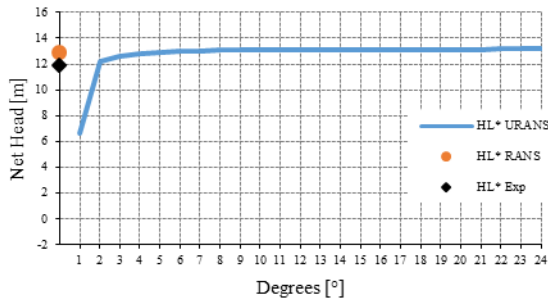


Figura 227. Valores de altura neta para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ RNG en condición de carga máxima HL*.

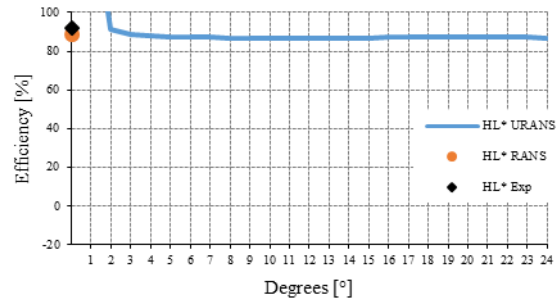


Figura 228. Valores de eficiencia hidráulica para diferentes posiciones del rodete utilizando modelo de turbulencia $k-\varepsilon$ RNG en condición de carga máxima HL*.

Perfiles de velocidad en estado transitorio

Al igual que se hizo en el análisis en estado estacionario, se realizó una comparación de los resultados obtenidos en estado transitorio para los perfiles de velocidad utilizando diferentes modelos de turbulencia. Los resultados que se muestran en todos los casos la solución obtenida después de 24 intervalos de tiempo, es decir después de 24° de giro del rodete.

La Figura 229 a la Figura 234 muestran los perfiles de velocidad obtenidos con los modelos de turbulencia $k-\omega$ SST y $k-\varepsilon$ Realizable para la condición de operación a carga parcial PL2, el modelo $k-\varepsilon$ RNG no produjo resultados, ya que como se mencionó anteriormente no se

contaba con una solución en estado estable para inicializar el cálculo. Se observa que entre ambos modelos existe poca diferencia en los perfiles de velocidad obtenidos, aunque ninguno logra una buena aproximación de la velocidad vertical en la parte media del cono, además hay una marcada diferencia en la predicción de la velocidad horizontal en las zonas cercana a las paredes del cono. Por otro lado, los resultados son muy similares a los obtenidos en estado estable.

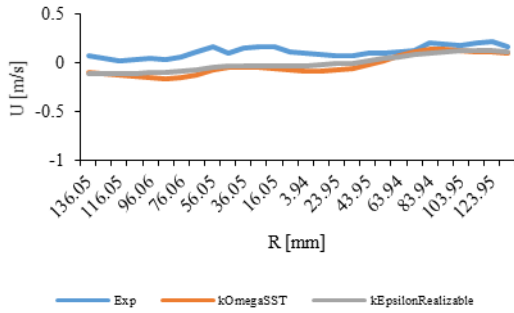


Figura 229. Valores de velocidad horizontal en PIV Line 1 con diferentes modelos de turbulencia a carga parcial PL2.

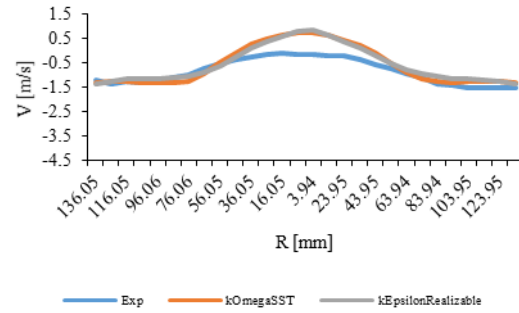


Figura 230. Valores de velocidad vertical en PIV Line 1 con diferentes modelos de turbulencia a carga parcial PL2.

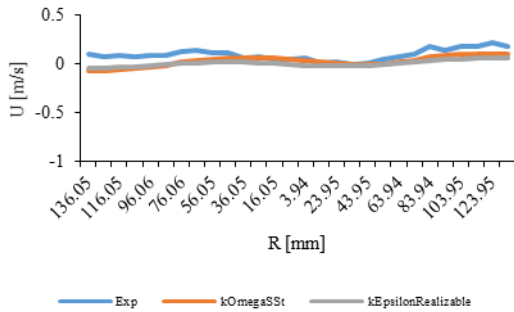


Figura 231. Valores de velocidad horizontal en PIV Line 2 con diferentes modelos de turbulencia a carga parcial PL2.

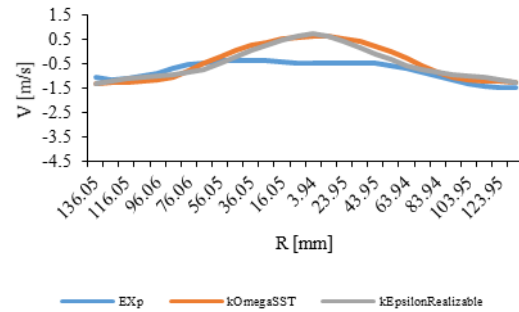


Figura 232. Valores de velocidad vertical en PIV Line 2 con diferentes modelos de turbulencia a carga parcial PL2.

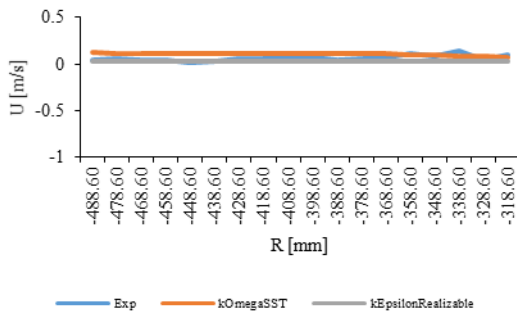


Figura 233. Valores de velocidad horizontal en PIV Line 3 con diferentes modelos de turbulencia a carga parcial PL2.

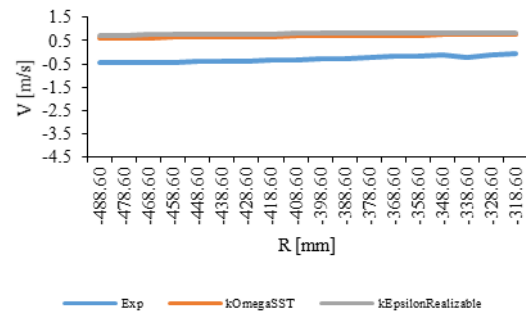


Figura 234. Valores de velocidad vertical en PIV Line 3 con diferentes modelos de turbulencia a carga parcial PL2.

Los perfiles de velocidad obtenidos por los tres modelos de turbulencia para la condición de operación de máxima eficiencia BEP* presentan en general una buena aproximación tanto para la velocidad horizontal como vertical. Así mismo, existe poca diferencia entre las predicciones de los tres modelos como se muestra en la Figura 235 a la Figura 240. Un comportamiento similar al observado en los cálculos en estado estable.

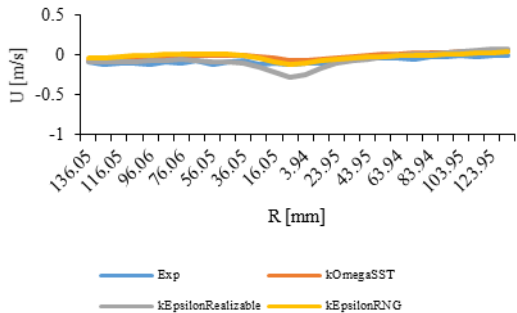


Figura 235. Valores de velocidad horizontal en PIV Line 1 con diferentes modelos de turbulencia en eficiencia máxima BEP*.

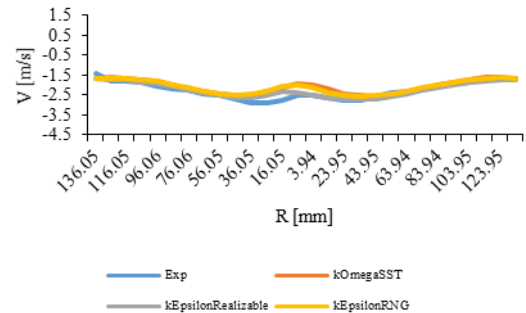


Figura 236. Valores de velocidad vertical en PIV Line 1 con diferentes modelos de turbulencia en eficiencia máxima BEP*.

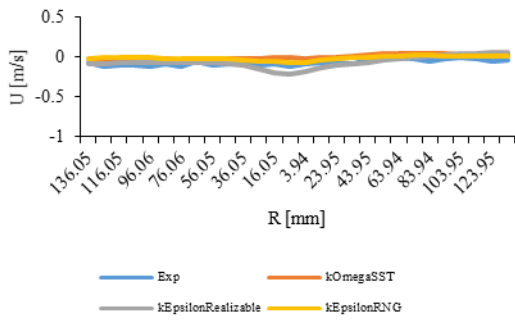


Figura 237. Valores de velocidad horizontal en PIV Line 2 con diferentes modelos de turbulencia en eficiencia máxima BEP*.

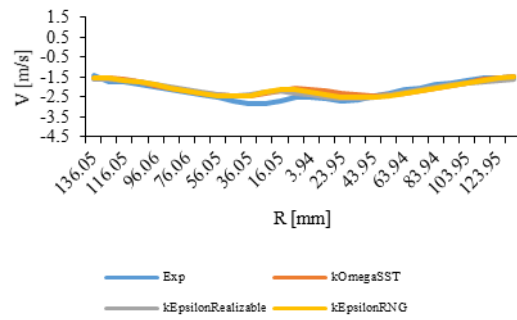


Figura 238. Valores de velocidad vertical en PIV Line 2 con diferentes modelos de turbulencia en eficiencia máxima BEP*.

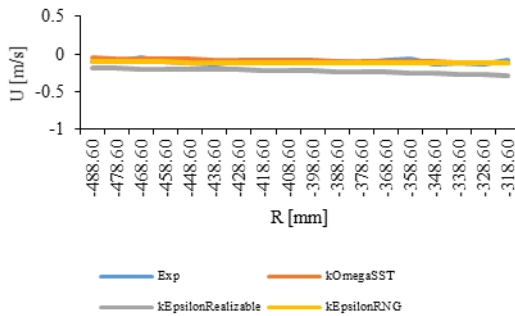


Figura 239. Valores de velocidad horizontal en PIV Line 3 con diferentes modelos de turbulencia en eficiencia máxima BEP*.

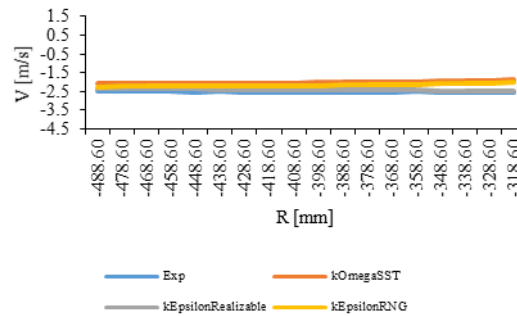


Figura 240. Valores de velocidad vertical en PIV Line 3 con diferentes modelos de turbulencia en eficiencia máxima BEP*.

Finalmente en la condición de operación de carga máxima HL* los perfiles obtenidos por los diferentes modelos de turbulencia presentan una aproximación cualitativamente buena, sin embargo, ninguno de ellos presenta aproximaciones cuantitativas exactas en zonas centrales del cono tanto para la velocidad vertical como par la horizontal. El mismo desfase de la velocidad horizontal se puede observar para las zonas cercanas a la pared como muestran la Figura 241 a la Figura 246, esto a pesar de que se están empleando esquemas de discretización de segundo orden y términos temporales.

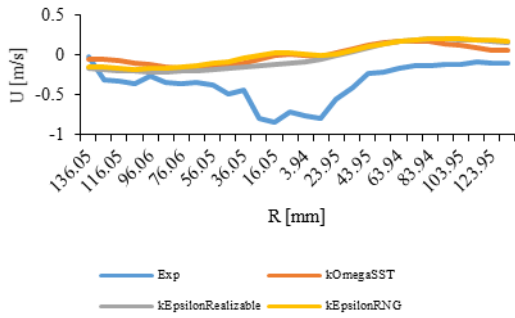


Figura 241. Valores de velocidad horizontal en PIV Line 1 con diferentes modelos de turbulencia a carga máxima HL*.

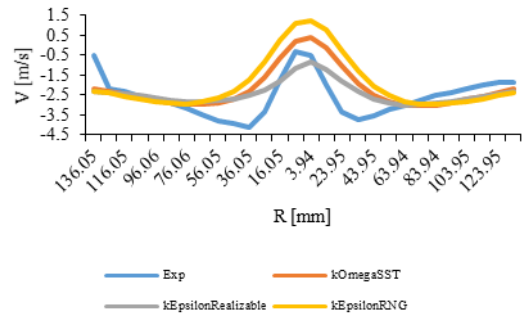


Figura 242. Valores de velocidad vertical en PIV Line 1 con diferentes modelos de turbulencia a carga máxima HL*.

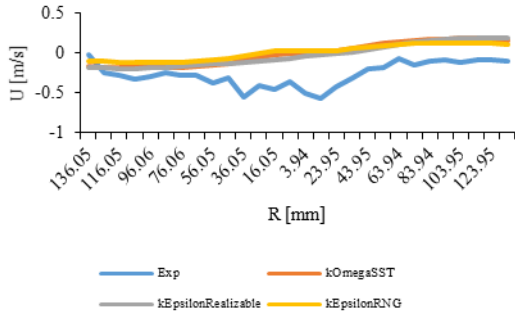


Figura 243. Valores de velocidad horizontal en PIV Line 2 con diferentes modelos de turbulencia a carga máxima HL*.

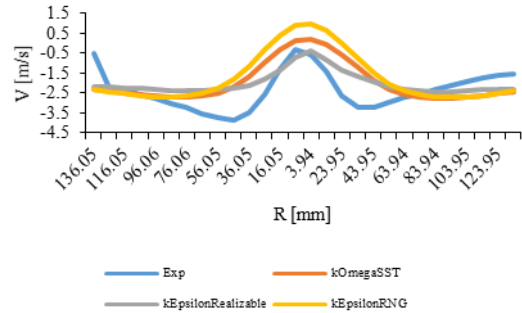


Figura 244. Valores de velocidad vertical en PIV Line 2 con diferentes modelos de turbulencia a carga máxima HL*.

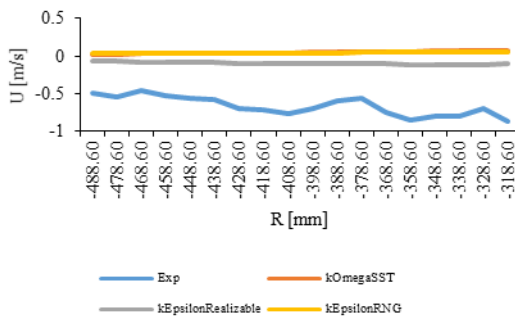


Figura 245. Valores de velocidad horizontal en PIV Line 3 con diferentes modelos de turbulencia a carga máxima HL*.

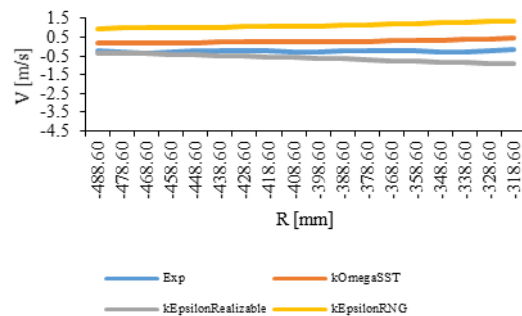


Figura 246. Valores de velocidad vertical en PIV Line 3 con diferentes modelos de turbulencia a carga máxima HL*.

Conclusiones.

Simular el flujo en turbomáquinas es un desafío para las aplicaciones industriales de CFD, sin embargo, OpenFOAM ha demostrado obtener con buenas aproximaciones numérica en algunos casos de estudio. Una gran ventaja de usar este software gratuito es que el código fuente completo es de acceso abierto. Por lo tanto, se ha investigado la posibilidad de acelerar las simulaciones de CFD mediante el uso de GPUs y un par de implementaciones de CUDA (RapidCFD y cufflink), debido a que las GPU pueden proporcionar un mayor rendimiento que los CPU para los solucionadores de CFD.

De los análisis realizados sobre el caso de estudio de la turbina T99, una vez que se realizó una comparación entre el uso de CPUs mediante OpenFOAM y GPUs acopladas con RapidCFD, se demostró que incluso usando GPUs de baja capacidad como la Nvidia Tesla C1060, se logra una reducción en los tiempos de cálculo, sin importar los diferentes parámetros que se utilicen en el *set up* para la simulación numérica. Además, los resultados mostraron que la mejor estrategia para acelerar el cálculo es utilizar el solucionador GAMG con valores de *tolerance* alrededor de 10^{-6} , ya que de esta manera se logra una reducción en los tiempos de cálculo si afectar la exactitud de los resultados en CFD.

Por otro lado, mediante el análisis de los registros de memoria RAM consumida para la solución de varios dominios computacionales del caso de estudio de la turbina T99, se demostró que el solucionador GAMG requiere aproximadamente un 22% más de memoria RAM que el PCG/PBiCG. Esto podría considerarse como el costo a cubrir por la diferencia en aceleración de cálculo para llegar a la solución, ya que el solucionador GAMG en promedio es hasta 70% más rápido el solucionador PCG/PBiCG.

Se debe remarcar que el tipo de pre acondicionador o suavizador asociado con cada uno de los solucionadores lineales mencionados tiene poco influencia en la exactitud de los resultados y velocidad de cálculo de cada caso.

Un análisis de la ley de Amdah y el *speedup* de la solución de los dominios computacionales de la turbina T99, demuestra que los recursos computacionales son aprovechados de manera más eficiente cuando se resuelven problemas de gran tamaño, es decir, que cada GPU debe resolver el tamaño de dominio que más se acerque al límite de su capacidad en memoria RAM. Así mismo, se ha demostrado que la paralelización para la solución de un dominio computacional dado tiene un costo asociado a cambio de la reducción en el tiempo de cálculo. Por lo tanto, de los datos obtenidos se propuso un conjunto de ecuaciones que determinan la memoria RAM requerida para cada solucionador lineal.

En la revisión bibliográfica hasta el momento no se ha encontrado un análisis similar en cuanto a la relación existente entre el tipo de solucionador lineal, la velocidad de solución, la

memoria RAM requerida y la eficiencia del uso del hardware aplicado a la solución de dominios computacionales en CFD.

Al comparar el desempeño de cufflink y RapidCFD, se han encontrado diferencias significativas. Por un lado, cufflink tiene la desventaja de ser más lento para la solución de dominios computacionales respecto a RapidCFD, aunque tiene la ventaja de hacer un uso menor de la memoria de las GPUs comparado con RapidCFD, lo cual es muy importante cuando se tienen recursos limitados para la solución de modelos de gran tamaño. Además, cufflink tiene la ventaja de trabajar en conjunto con foam-extend, el cual cuenta con más herramientas desarrolladas para la solución de problemas de turbo maquinaria en comparación con RapidCFD. Así que, no se puede descartar su uso de cufflink en pro de RapidCFD debido a su mayor rapidez de cálculo. Sin embargo, existe un área de oportunidad para trasladar algunas herramientas de foam-extend y cufflink a RapidCFD o incluso desarrollar la integración de ambos *software* en conjunto para la solución de algunos problemas más complicados y demandantes.

Para el caso de estudio de la turbina Francis 99, se llevó a cabo un análisis de aceleración para verificar el potencial de las GPU en la solución de un caso de turbomaquinaria más complejo y grande en CFD. Se demostró que se puede lograr una aceleración de hasta 20x, respecto a OpenFOAM, cuando el código RapidCFD se usa en combinación con GPUs Nvidia Tesla K40.

La validación de los resultados de CFD demostró que la presión diferencial, el par motor y la altura neta son sobre estimados, sin embargo, la eficiencia hidráulica está por debajo de lo previsto. Los resultados de los cálculos computacionales en estado estacionario y transitorio utilizando diferentes modelos de turbulencia mostraron que el modelos *k-ε Realizable* produce los mejores resultados en la mayoría de las condiciones de operación.

En cuanto a la precisión de los resultados obtenidos para los perfiles de velocidad se observó que los tres modelos de turbulencia utilizados, tanto en estado estacionario como transitorio, producen perfiles prácticamente idénticos, al menos para las condiciones de operación analizadas. Sin embargo, los resultados en estado transitorio, son un poco más precisos que los obtenidos en estado estacionario, ya se emplearon esquemas de discretización de segundo orden y en estado estable se utilizaron esquemas de primer orden.

Otro punto importante a mencionar es el referente al tiempo de cálculo, ya que se observó que para la mayoría de las condiciones de operación el modelo de turbulencia *k-ε RNG* fue el más rápido, tanto en estado estable como en estado transitorio.

De lo anterior se puede inferir que la mejor opción para realizar análisis de CFD en aplicaciones de turbomaquinaria de flujo incompresible, es utilizar esquemas de discretización de primer orden en estado estacionario con MRF en conjunto con el modelo

de turbulencia $k-\varepsilon$ *Realizable*, ya que esta combinación ofrece menores tiempos de cálculo, menor uso de memoria y aproximaciones aceptables.

Sin embargo, el uso de simulaciones en estado estacionario no ofrece la posibilidad de analizar fluctuaciones en el torque producido durante la rotación completa del rodete, así como detalles de interacción fluido estructura (vibraciones) y pulsaciones de presión en el tubo difusor. Esta clase de análisis conllevaría un costo computacional asociado al uso de mallas deslizantes y el tiempo requerido para los cálculos de cada intervalo de tiempo en función del tamaño de este, ya que entre menor tamaño de intervalo de tiempo se requiere mayor número de estos para evaluar un giro completo del rodete, pero se podrían apreciar diversos espectros de frecuencias de vibración y pulsaciones de presión.

Los resultados de aceleración de cálculo obtenidos son prometedores, ya que sugieren que trasladar la parte de cálculo intensiva de las simulaciones CFD a GPUs conduce a un ahorro significativo en el tiempo de cálculo. Al añadir a esto último el hecho de que la validación de las simulaciones numéricas de los caso de estudio de la turbina T99 y Francis 99 da certidumbre a los datos obtenidos. De esta manera se demuestra la viabilidad del proyecto para la creación de un banco de pruebas virtual basado en HPC para turbinas tipo Francis.

Referencias.

- [1] CONAGUA, «Numeragua,» Comisión Nacional del Agua, Mexico City, 2015.
- [2] CONAGUA, «Atlas del Agua en México,» Comisión Nacional del Agua, Mexico City, 2015.
- [3] CONAGUA, «Programa Hídrico Visión 2030 del Estado de Michoacán de Ocampo,» Comisión Nacional del Agua, Mexico City, 2009.
- [4] CONAGUA, «Estadísticas del Agua en México,» Comisión Nacional del Agua, Mexico City, 2015.
- [5] SENER, «Prospectiva de Energías Renovables 2015-2029,» Secretaria de Energía, Mexico City, 2015.
- [6] IHA, «Hydropower Status Report,» International Hydropower Association, 2016.
- [7] Centro Mario Molina, «Estrategias Regionales y Sectoriales para Lograr un Desarrollo Sustentable y de Baja Intensidad de Carbono en México. Parte II Potencial de Generación Minihidroeléctrica en México,» Centro Mario Molina para Estudios Estratégicos Sobre Energía y Medio Ambiente A.C., Mexico City, 2012.
- [8] CFE, *Turbina Hidraulica Tipo Francis de Eje Vertical. CFE W8300-11*, Comisión Federal de Electricidad, 2015.
- [9] CFE, *Unidad Auxiliar con Turbina Francis de Eje Horizontal. CFE W8300-13*, Comisión Federal de Electricidad, 2007.
- [10] IEC, *Hydraulic turbines, storage pumps and pump-turbines - Model acceptance tests. IEC 60193*, International Electrotechnical Commission, 1999.
- [11] M. Cervantes, I. Jansson, A. Jourak, S. Glavatskikh y J. Aidanpää, «PORJUS U9 A Full-Scale Hydropower Research Facility,» de *24th IAHR Symposium on Hydraulic Machinery and Systems*, Foz Do Iguassu, Brazil, 2008.
- [12] Norwegian University of Science and Technology, «NVKS Norwegian Hydropower Centre,» [En línea]. Available: <https://www.ntnu.edu/nvks/francis-99>.
- [13] C. Trivedi y M. J. Cervantes, «State of the art in numerical simulation of high head Francis turbines,» *Renew. Energy Environ. Sustain.*, vol. 1, n° 20, pp. 1-5, 2016.
- [14] C. Trivedi, M. J. Cervantes y O. G. Dahlhaug, «Experimental and Numerical Studies of a High-Head Francis Turbine: A Review of the Francis-99 Test Case,» *Energies*, vol. 9, n° 74, pp. 1-24, 2016.
- [15] C. Trivedi, M. J. Cervantes y B. K. Gandhi, «Investigation of a High Head Francis Turbine at Runaway Operating Conditions,» *Energies*, vol. 9, n° 149, pp. 1-22.

- [16] S. Kurosawa, S. M. Lim y Y. Enomoto, «Virtual model test for a Francis turbine,» de *25th IAHR Symposium on Hydraulic Machinery and Systems*, 2010.
- [17] H.-J. Choi, M. A. Zullah, H.-W. Roh, P.-S. Ha, S.-Y. Oh y Y.-H. Lee, «CFD validation of performance improvement of a 500 kW Francis turbine,» *Renewable Energy*, vol. 54, pp. 111-123, 2013.
- [18] A. Nocente, T. Arslan, D. Jasinski y T. K. Nielsen, «A Study of Flow inside a Centrifugal Pump: High Performance Numerical Simulations Using GPU cards,» de *International Symposium on Transport Phenomena and Dynamics of Rotating Machinery*, Hawaii, Honolulu, 2016.
- [19] H. K. Versteeg and W. Malalasekera, *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*, 2nd ed., Harlow, Essex: Pearson Education Limited, 2007.
- [20] R. B. Bird, W. E. Stewart, E. N. Lightfoot y D. J. Klingenberg, *Introductory Transport Phenomena*, Hoboken, NJ: JohnWiley & Sons, Inc, 2015.
- [21] H. Tennekes y J. L. Lumley, *A First Course in Turbulence*, Cambridge, MA: The MIT Press, 1972.
- [22] F. Moukalled, L. Mangani y M. Darwish, *The Finite Volume Method in Computational Fluid Dynamics - An Advanced Introduction with OpenFOAM and Matlab*, Switzerland: Springer International Publishing, 2016.
- [23] Canonical Ltd, «Ubuntu,» [En línea]. Available: https://www.ubuntu.com/?utm_expid=.92kIXiqDSpuSF35IE8FS1Q.0&utm_referrer=.
- [24] The Ubuntu Manual Team, «Getting Started with Ubuntu 16.04,» 2016.
- [25] OpenCFD Ltd, «OpenFOAM The Open Source CFD Toolbox,» [En línea]. Available: <https://www.openfoam.com>.
- [26] «foam-extend,» SourceForge, [En línea]. Available: <https://sourceforge.net/p/foam-extend/foam-extend-4.0/ci/master/tree/ReleaseNotes.txt#l52>.
- [27] OpenCFD Limited, «OpenFOAM. The Open Source CFD Tool Box. User Guide v1706,» 2017.
- [28] SimFlow CFD software, «RapidCFD GPU,» [En línea]. Available: <https://sim-flow.com/rapid-cfd-gpu>.
- [29] simFlow CFD Software, «Atizar/RapidCFD-dev,» GitHub, Inc., [En línea]. Available: <https://github.com/Atizar/RapidCFD-dev>.
- [30] «cufflink-library,» Google Code Archive, [En línea]. Available: <https://code.google.com/archive/p/cufflink-library/>.
- [31] «cusp-library,» Google Code Archive, [En línea]. Available: <https://code.google.com/archive/p/cusp-library/>.
- [32] «thrust,» Google Code Archive, [En línea]. Available: <https://code.google.com/archive/p/thrust/>.

- [33] NVIDIA Corporation, «CUDA Zone,» [En línea]. Available: <https://developer.nvidia.com/cuda-zone>.
- [34] NVIDIA Corporation, «Nvidia Cuda Getting Started Guide For Linux v6.5,» 2014.
- [35] NVIDIA Corporation, «Cuda Compiler Driver NVCC v6.5,» 2014.
- [36] NVIDIA Corporation, «Cuda C Programming Guide v6.5,» 2014.
- [37] ERCOFTAC, «ERCOFTAC QNET,» [En línea]. Available: http://www.kbwiki.ercoftac.org/w/index.php/Abstr:Draft_tube.
- [38] Lulea University of Technology, «Francis-99,» [En línea]. Available: <https://www.ltu.se/research/subjects/Stromningslara/Konferenser/Francis-99/Test-Case-1.111520?l=en>.
- [39] H. Nilsson, «Some Experiences on the Accuracy and Parallel Performance of OpenFOAM for CFD in Water Turbines,» de *Applied Parallel Computing. State of the Art in Scientific Computing*, Springer, Berlin, Heidelberg, 2007, pp. 168-176.
- [40] Z. Malecha, Ł. Mirosław, T. Tomczak, Z. Koza, M. Matyka, W. Tarnawski y D. Szczerba, «GPU-based simulation of 3D blood flow in abdominal aorta using OpenFOAM,» *Archives of Mechanics*, vol. 63, n° 2, pp. 137 - 161, 2011.
- [41] B. Gebart, L. Gustavsson y R. Karlsson, «Proceedings of Turbine 99 Workshop on draft tube flow in Porjus, Sweden,» Luleå University of Technology, 2000.
- [42] T. Engström, L. Gustavsson y R. Karlsson, «Turbine-99 Workshop 2 on Draft Tube Flow,» de *Proceedings of 21st IAHR Symposium on Hydraulic Machinery and Systems*, 2005, Lausanne, Switzerland.
- [43] M. Cervantes, T. Engström y L. Gustavsson, «Proceedings of the third IAHR/ERCOFTAC Workshop on draft tube flows,» Luleå University of Technology, Porjus, Sweden, 2005.
- [44] C. Navarro, N. Hitschfeld-Kahler y L. Mateu, «A survey on parallel computing and its applications in data-parallel problems using GPU architectures,» *Communications in Computational Physics*, vol. 15, n° 2, pp. 285-329, 2014.
- [45] Q. He, C. Hongli y F. Jingchao, «Acceleration of the OpenFOAM-based MHD solver using graphics processing units,» *Fusion Engineering and Design*, vol. 101, pp. 88-93, 2015.
- [46] P. Pacheco, *An introduction to parallel programming*, Elsevier, 2011.
- [47] U. Andersson, «Test case T- Some news results and updates since Workshop 1,» de *Proceedings of Turbine 99-WS2, The second ERCOFTAC Workshop on draft tubeflow*, Alvkärleby, Sweden, 2001.
- [48] S. Galván, M. Reggio y F. Guibault, «Assessment Study of k-ε Turbulence Models and Near-Wall Modeling for Steady State Swirling Flow Analysis in Draft Tube Using Fluent,» *Engineering Applications of Computational Fluid Mechanics*, vol. 5, n° 4, pp. 459-478, 2011.

- [49] J. M. Marra, L. M. Gramani, L. F. Zubeldía y E. Kaviski, «Prediction of Pressure Pulsation in Francis Turbines Using Rans Solution: An Applied Study of Effects of the Turbulence Models, Mesh, Domain Extension and Multiphase Flow,» *American Journal of Hydropower, Water and Environment Systems*, vol. 6, pp. 33-43, 2018.
- [50] N. Herrera, S. Galván, J. Camacho, G. Solorio and A. Aguilar, "Automatic shape optimization of a conical-duct diffuser using a distributed computing algorithm," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 39, no. 11, p. 4367–4378, 2017.
- [51] H. Keck y M. Sick, «Thirty years of numerical flow simulation in hydraulic turbomachines,» *Acta Mechanica*, n° 201, p. 211–229, 2008.
- [52] «Norwegian Hydropower Centre,» [En línea]. Available: <https://www.ntnu.edu/nvks/francis-99>. [Último acceso: 15 October 2020].
- [53] D. Molinero, S. Galván, J. Pacheco y N. Herrera, «Multi GPU Implementation to Accelerate the CFD Simulation of a 3D Turbo-Machinery Benchmark Using the RapidCFD Library,» de *10th International Conference on Supercomputing in Mexico, ISUM 2019*, Monterrey, Mexico, 2019.
- [54] C. Trivedi, M. J. Cervantes, B. K. Gandhi and O. G. Dahlhaug, "Experimental and Numerical Studies for a High Head Francis Turbine at Several Operating Points," *Journal of Fluids Engineering*, vol. 135, no. 11, 2013.
- [55] C. Trivedi, M. J. Cervantes and O. G. Dahlhaug, "Numerical Techniques Applied to Hydraulic Turbines: A Perspective Review," *Applied Mechanics Reviews*, vol. 68, no. 1, 2016.

Anexos.

NOMBRE DEL TRABAJO

**Banco de Pruebas Virtual Basado en HP
C para Pruebas de Modelo de la Turbina
Francis 99**

AUTOR

Daniel Molinero Hernández

RECUENTO DE PALABRAS

43060 Words

RECUENTO DE CARACTERES

245490 Characters

RECUENTO DE PÁGINAS

152 Pages

TAMAÑO DEL ARCHIVO

7.2MB

FECHA DE ENTREGA

Aug 14, 2024 7:48 AM GMT-6

FECHA DEL INFORME

Aug 14, 2024 7:50 AM GMT-6

● **20% de similitud general**

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para cada base de datos.

- 20% Base de datos de Internet
- Base de datos de Crossref
- 0% Base de datos de trabajos entregados
- 7% Base de datos de publicaciones
- Base de datos de contenido publicado de Crossref

Formato de Declaración de Originalidad y Uso de Inteligencia Artificial

Coordinación General de Estudios de Posgrado
Universidad Michoacana de San Nicolás de Hidalgo



A quien corresponda,

Por este medio, quien abajo firma, bajo protesta de decir verdad, declara lo siguiente:

- Que presenta para revisión de originalidad el manuscrito cuyos detalles se especifican abajo.
- Que todas las fuentes consultadas para la elaboración del manuscrito están debidamente identificadas dentro del cuerpo del texto, e incluidas en la lista de referencias.
- Que, en caso de haber usado un sistema de inteligencia artificial, en cualquier etapa del desarrollo de su trabajo, lo ha especificado en la tabla que se encuentra en este documento.
- Que conoce la normativa de la Universidad Michoacana de San Nicolás de Hidalgo, en particular los Incisos IX y XII del artículo 85, y los artículos 88 y 101 del Estatuto Universitario de la UMSNH, además del transitorio tercero del Reglamento General para los Estudios de Posgrado de la UMSNH.

Datos del manuscrito que se presenta a revisión		
Programa educativo	Doctorado en Ciencias en Ingeniería Mecánica	
Título del trabajo	Banco de Pruebas Virtual Basado en HPC para Pruebas de Modelo de la Turbina Francis 99	
	Nombre	Correo electrónico
Autor/es	Daniel Molinero Hernández	daniel.molinero@umich.mx
Director	Sergio Ricardo Galván González	sergio.galvan@umich.mx
Codirector	Nicolás David Herrera Sandoval	nicolas.hs@morelia.tecnm.mx
Coordinador del programa	Georgina Carbajal de la Torre	georgina.carbajal@umich.mx


Uso de Inteligencia Artificial		
Rubro	Uso (sí/no)	Descripción
Asistencia en la redacción	No	

Formato de Declaración de Originalidad y Uso de Inteligencia Artificial

Coordinación General de Estudios de Posgrado
Universidad Michoacana de San Nicolás de Hidalgo



Uso de Inteligencia Artificial		
Rubro	Uso (sí/no)	Descripción
Traducción al español	No	
Traducción a otra lengua	No	
Revisión y corrección de estilo	No	
Análisis de datos	No	
Búsqueda y organización de información	No	
Formateo de las referencias bibliográficas	No	
Generación de contenido multimedia	No	
Otro	No	

Datos del solicitante	
Nombre y firma	Daniel Molinero Hernández 
Lugar y fecha	Morelia, Michoacán a 14 de agosto de 2024.