

**CALCULOS DE CARGA PARCIAL EN CICLOS DE VAPOR
CONVENCIONAL**

AUDES HERNÁNDEZ MORALES

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA MECÁNICA
BUCARAMANGA**

2004

CALCULOS DE CARGA PARCIAL EN CICLOS DE VAPOR CONVENCIONAL

AUDES HERNÁNDEZ MORALES

Trabajo de grado para optar el título de
Ingeniero Mecánico

Director

DAVID ALFREDO FUENTES DIAZ

Ingeniero Mecánico

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA MECÁNICA
BUCARAMANGA

2004

DEDICATORIA

Al Señor Dios Todopoderoso.

A mis padres Audes y Anastasia por su inmenso amor y apoyo permanente e incondicional.

AUDES

AGRADECIMIENTO

Quiero expresar mi profundo agradecimiento al Ingeniero DAVID ALFREDO FUENTES DIAZ, por su orientación y permanente colaboración sin la cual no hubiera sido posible culminar con éxito este proyecto.

CONTENIDO

	pág.
INTRODUCCIÓN	17
1. CÁLCULOS A CARGA PARCIAL	19
1.1 TURBINA DE VAPOR	19
1.1.1 Generalidades	19
1.1.2 Cálculos	20
1.2 INTERCAMBIADOR, CONDENSADOR Y CALDERA	21
1.2.1 Generalidades	21
1.2.2 Cálculos	22
1.3 VÁLVULAS	25
1.3.1 Generalidades	25
1.3.2 Cálculos	25
1.4 BOMBAS	29
1.4.1 Generalidades	29
1.4.2 Cálculos	30
2. MANUAL DEL USUARIO	31
2.1 PROGRAMA TURBINA_CARGA_PARCIAL.EXE	31
2.1.1 Interfaz de usuario	31
2.1.2 Datos de entrada	36
2.1.3 Funcionamiento	40
2.2 PROGRAMA INTERCAMBIADOR_CARGA_PARCIAL.EXE_	49
2.2.1 Interfaz de usuario	49
2.2.2 Datos de entrada	53
2.2.3 Funcionamiento	55
2.3 PROGRAMA CONDENSADOR_CARGA_PARCIAL.EXE_	61
2.3.1 Interfaz de usuario	61

2.3.2	Datos de entrada	65
2.3.3	Funcionamiento	67
2.4	PROGRAMA CALDERA_CARGA_PARCIAL.EXE	73
2.4.1	Interfaz de usuario	73
2.4.2	Datos de entrada	79
2.4.3	Funcionamiento	81
2.5	PROGRAMA VALVULA_CARGA_PARCIAL.EXE	86
2.5.1	Interfaz de usuario	86
2.5.2	Datos de entrada	91
2.5.3	Funcionamiento	93
2.6	PROGRAMA BOMBA_CARGA_PARCIAL.EXE	99
2.6.1	Interfaz de usuario	99
2.6.2	Datos de entrada	103
2.6.3	Funcionamiento	105
3.	EJEMPLOS DE APLICACIÓN	111
3.1	PROGRAMA TURBINA_CARGA_PARCIAL.EXE	111
3.2	PROGRAMA INTERCAMBIADOR_CARGA_PARCIAL.EXE	113
3.3	PROGRAMA VALVULA_CARGA_PARCIAL.EXE	114
3.4	PROGRAMA BOMBA_CARGA_PARCIAL.EXE	117
4.	CONCLUSIONES	120
5.	RECOMENDACIONES	122
	BIBLIOGRAFÍA	123

LISTA DE FIGURAS

	pág.
Figura 1. Características de flujo Lineal a través de las válvulas	27
Figura 2. Características de flujo Equal (Exponencial) a través de las válvulas	27
Figura 3. Presentación Turbina_Carga_Parcial.exe	32
Figura 4. Elementos del comando Archivo de la turbina	33
Figura 5. Elementos del comando Entrar de la turbina	33
Figura 6. Elementos del comando Ver... de la turbina	34
Figura 7. Control del número de secciones de la turbina	34
Figura 8. Controles de datos en los flujos de la turbina	35
Figura 9. Control de selección de entrada de la presión de salida en condiciones de operación.	35
Figura 10. Opciones de cálculo del programa de la turbina.	35
Figura 11. Controles de los flujos de la turbina.	36
Figura 12. Datos termodinámicos de los flujos de la turbina.	37
Figura 13. Datos de las secciones de la turbina.	38
Figura 14. Datos mecánicos de una sección de la turbina	39
Figura 15. Datos del generador de la turbina.	39
Figura 16. Creando archivo de datos de la turbina	40
Figura 17. Guardar como... de la turbina	41
Figura 18. Seleccionando las secciones de la turbina	42
Figura 19. Seleccionando las condiciones de flujo de la turbina	43
Figura 20. Seleccionando presión de salida en operación habilitada	43
Figura 21. Seleccionando los parámetros a calcular en la turbina	44
Figura 22. Entrando datos generales de las secciones de la turbina	44
Figura 23. Entrando datos mecánicos de las secciones de la turbina	45
Figura 24. Entrando datos del generador	46

Figura 25. Comando calcular de la turbina	47
Figura 26. Fin de los cálculos de la turbina	47
Figura 27. Revisando los resultados de la turbina	48
Figura 28. Archivo de texto con los resultados de la turbina	48
Figura 29. Presentación de Intercambiador_Carga_Parcial.exe	49
Figura 30. Elementos del comando Archivo del menú del intercambiador	50
Figura 31. Elementos del comando de menú Entrar Datos del intercambiador	50
Figura 32. Elementos del comando del menú Ver... del intercambiador	51
Figura 33. Flujos del intercambiador	52
Figura 34. Tipo de datos de entrada del intercambiador	52
Figura 35. Controles para la selección del tipo de intercambiador	53
Figura 36. Datos termodinámicos de los flujos del intercambiador	53
Figura 37. Datos mecánicos del intercambiador	54
Figura 38. Exponentes para el análisis del intercambiador	54
Figura 39. Creando archivo de datos del intercambiador	55
Figura 40. Guardar como... del intercambiador	56
Figura 41. Seleccionando el tipo de intercambiador	56
Figura 42. Entrando datos en condiciones de diseño y de operación del intercambiador	57
Figura 43. Entrando datos mecánicos del intercambiador	58
Figura 44. Entrando datos de análisis del intercambiador	58
Figura 45. Comando calcular del intercambiador	59
Figura 46. Fin de los cálculos del intercambiador	59
Figura 47. Revisando los resultados del intercambiador	60
Figura 48. Archivo de texto con los resultados del intercambiador	61
Figura 49. Presentación Condensador_Carga_Parcial.exe	62
Figura 50. Elementos del menú Archivo del condensador	63
Figura 51. Elementos del menú Entrar Datos del condensador	63
Figura 52. Comando Calcular del condensador	64

Figura 53. Elementos del menú Ver... del condensador	64
Figura 54. Controles de los flujos del condensador	65
Figura 55. Tipo de datos de entrada del condensador	65
Figura 56. Datos termodinámicos de los flujos del condensador	66
Figura 57. Datos mecánicos del condensador	66
Figura 58. Datos de análisis del condensador	67
Figura 59. Creando archivo de datos del condensador	68
Figura 60. Guardar como... del condensador	68
Figura 61. Seleccionando datos de diseño y de operación del condensador	69
Figura 62. Entrando datos mecánicos del condensador	70
Figura 63. Entrando datos de análisis del condensador	70
Figura 64. Oprimiendo el comando Calcular del condensador	71
Figura 65. Fin de los cálculos del condensador	71
Figura 66. Revisando los resultados del condensador	72
Figura 67. Archivo de texto con los resultados del condensador	73
Figura 68. Presentación Caldera_Carga_Parcial.exe	74
Figura 69. Elementos del menú Archivo de la caldera	74
Figura 70. Comando Calcular de la caldera	76
Figura 71. Elementos del menú Ver... de la caldera	76
Figura 72. Controles de los componentes de la caldera	77
Figura 73. Controles para seleccionar los componentes de la caldera	77
Figura 74. Control para entrar la composición de los gases de combustión	77
Figura 75. Ventana del componente hogar de la caldera	78
Figura 76. Elementos del menú de cada componente de la caldera	78
Figura 77. Datos mecánicos por cada componente de la caldera	80
Figura 78. Exponentes para el análisis de cada componente de la caldera	80
Figura 79. Datos de los gases de combustión de la caldera	81
Figura 80. Creando archivo de datos de la caldera	82
Figura 81. Guardar como... de la caldera	82
Figura 82. Seleccionando los componentes de la caldera	83

Figura 83. Entrando composición de los gases de combustión de la caldera	84
Figura 84. Fin de los cálculos de la caldera	84
Figura 85. Revisando los resultados de la caldera	85
Figura 86. Archivo de texto con los resultados de la caldera	86
Figura 87. Presentación Válvula_Carga_Parcial.exe	87
Figura 88. Elementos del menú Archivo de la válvula	88
Figura 89. Elementos del menú Entrar Datos de la válvula	89
Figura 90. Comando Calcular de la válvula	90
Figura 91. Elementos del menú Ver... de la válvula	90
Figura 92. Opciones de cálculo del programa de la válvula	90
Figura 93. Datos de diseño de la válvula	91
Figura 94. Dato de operación de la válvula	91
Figura 95. Características generales de la válvula	92
Figura 96. Datos de la curva del comportamiento del flujo a través de la válvula	92
Figura 97. Creando archivo de datos de la válvula	93
Figura 98. Guardar como... de la válvula	94
Figura 99. Seleccionando la opción de cálculo de la válvula	94
Figura 100. Entrando los datos de diseño de la válvula	95
Figura 101. Entrando los datos de operación de la válvula	96
Figura 102. Entrando los datos generales de la válvula	96
Figura 103. Entrando los datos de la curva del comportamiento del flujo a través de la válvula	97
Figura 104. Fin de los cálculos de la válvula	98
Figura 105. Revisando los resultados de la válvula	98
Figura 106. Archivo de texto con los resultados de la válvula	99
Figura 107. Presentación de la Bomba_Carga_Parcial.exe	100
Figura 108. Elementos del menú Archivo de la bomba	100
Figura 109. Comando calcular de la bomba	101
Figura 110. Elementos del menú Ver... de la bomba	102
Figura 111. Opción de cálculo de la bomba	102

Figura 112. Tipo de datos de la bomba	102
Figura 113. Controles de los flujos de la bomba	103
Figura 114. Botón para entrar los datos de la curva del comportamiento de la bomba	103
Figura 115. Datos de diseño de los flujos de la entrada y salida de la bomba	104
Figura 116. Otros datos de diseño de la bomba	104
Figura 117. Dato de operación de la bomba	105
Figura 118. Datos de la curva del funcionamiento de la bomba	105
Figura 119. Creando un archivo de datos de la bomba	106
Figura 120. Guardar como... de la bomba	107
Figura 121. Seleccionando la opción de cálculo de la bomba	107
Figura 122. Entrando datos de la curva del comportamiento de la bomba	108
Figura 123. Fin los cálculos de la bomba	109
Figura 124. Revisando los resultados de la bomba	110
Figura 125. Archivo de texto con los resultados de la bomba	110
Figura 126. Informe de resultados del programa Turbina_Carga_Parcial.exe	112
Figura 127. Informe de resultados del Intercambiador_Carga_Parcial.exe	115
Figura 128. Resultados del programa Válvula_Carga_Parcial.exe	116
Figura 129. Resultados Bomba_Carga_Parcial.exe	119

LISTA DE ANEXOS

	Pag.
Anexo A. Diagrama de flujo para el cálculo a carga parcial de turbinas y generadores a vapor de 16500 KW y mayores	125
Anexo B. Diagrama de flujo del cálculo a carga parcial para el intercambiador de calor	137
Anexo C. Diagrama de flujo del cálculo a carga parcial para la válvula de control	141
Anexo D. Diagrama de flujo del cálculo a carga parcial para las bombas	143
Anexo E. Archivos fuentes del programa Turbina_Carga_Parcial.exe	144
Anexo F. Archivos fuentes del programa Intercambiador_Carga_Parcial.exe	180
Anexo G. Archivos fuentes del programa Caldera_Carga_Parcial.exe	209
Anexo H. Archivos fuentes del programa Válvula_Carga_Parcial.exe	232
Anexo I. Archivo fuentes del programa Bomba_Carga_Parcial.exe	236
Anexo J. Artículo ASME No. 62-WA-209	240

RESUMEN

TÍTULO: CALCULOS DE CARGA PARCIAL EN CICLOS DE VAPOR CONVENCIONAL.*

AUTOR: HERNÁNDEZ MORALES, Audes.**

PALABRAS CLAVES:

Simulador, equipos de plantas térmicas a vapor, carga parcial.

DESCRIPCIÓN:

Los Cálculos de Carga Parcial en Ciclos de Vapor Convencional son seis (6) programas diseñados y desarrollados pensando en la posibilidad de incluirlos en el Simulador Térmico SB-2000. Dentro de estos programas se utilizan procedimientos, en lo que se refiere a cálculos a carga parcial, para modelar el comportamiento de algunos de los equipos de las plantas de vapor que ya contenía el SB-2000.

Los Cálculos de Carga Parcial en Ciclos de Vapor Convencional considera los siguientes equipos de plantas de vapor convencional para el análisis a carga parcial: Turbinas de Vapor, Calderas, Intercambiadores de calor, Condensadores, Válvulas y Bombas. Cada equipo es analizado en forma separada de otro, en un programa que contiene su propia interfaz de usuario; y además se ejecuta independientemente del SB-2000.

* Proyecto de grado.

** Facultad de Ingenierías Físico-mecánicas, Escuela de Ingeniería Mecánica, Ing. David Fuentes.

SUMMARY

TITLE: OFF-DESIGN CALCULATIONS IN CONVENTIONAL STEAM CYCLES.*

AUTHOR: HERNÁNDEZ MORALES, Audes.**

KEY WORDS:

Simulator, equipments of steam thermal plants, off-design.

DESCRIPTION:

Off-design Calculations in Conventional Steam Cycles are six (6) computer programs designed and developed thinking of the possibility of including them in the SB-2000 Thermal Simulator. Inside these programs procedures are used, in what concerning to off-design calculations, to model the behavior of some of the equipments of the steam plants that the SB-2000 already contents.

The Off-design Calculations in Conventional Steam Cycles considers the following equipments of conventional steam plants for the analysis to off-design conditions: Steam Turbines, Boilers, Heat Exchanger, Condensers, Valves and Pumps. Each equipment is analyzed separately of another in an individual program that contains its own user interface; and it is also executed independently of the SB-2000.

* Degree project.

** Faculty of Physical-mechanical Engineering, Mechanical Engineering, David Fuentes.

INTRODUCCIÓN

La Escuela de Ingeniería Mecánica de la Universidad Industrial de Santander ha venido desarrollando una serie de proyectos de investigación que buscan simular plantas de generación de energía eléctrica, para lo cual se han ejecutado dos trabajos de grado; el primero titulado “SIMULADOR DE UNA PLANTA ELECTRICA CON CICLO DE VAPOR CONVENCIONAL”¹ que permite simular el comportamiento en condiciones de diseño de plantas de generación de energía eléctrica que funcionan de acuerdo con un ciclo Ranking. Posteriormente, se desarrolló otro proyecto, el cual es una actualización del primero, para simular en condiciones de diseño y de carga parcial ciclos de generación de energía basados en el ciclo Brayton de gas, titulado SIMULADOR DE PLANTAS TÉRMICAS A GAS². En ambos se permite incluir los elementos del sistema de una manera flexible y se realizan cálculos termodinámicos de balances de masa y de energía en cada uno de los componentes.

Las plantas de generación de energía eléctrica tienen establecidas unas condiciones de diseño para una operación de máxima potencia; sin embargo, dicha operación no es posible ejecutarla permanentemente a dichas condiciones, ya que a menudo se requiere de variaciones, debido a circunstancias como la necesidad de enfrentarse a una demanda variable, requisitos prácticos de despacho de potencia, y cambios en las condiciones externas, verbigracia la temperatura ambiente. Un modelo del desempeño de estos ciclos a carga parcial es por consiguiente una herramienta importante para el diseño o selección de nuevas plantas de potencia, e igualmente se puede usar para analizar el funcionamiento de plantas existentes, ayudando a encontrar deficiencias operacionales precisas y evaluando las oportunidades de corrección.

¹ JIMÉNEZ GALLARDO, Edwing Yovany y MORENO BAYONA, Luis Eduardo. Simulador de una planta térmica con ciclo de vapor convencional. Proyecto de grado UIS, 2000.

² AMAYA, Edward. Simulador de plantas térmicas a gas. Proyecto de grado UIS, 2003.

Para analizar el funcionamiento de plantas de potencia sobre rangos típicos de operación, es necesario desarrollar modelos que puedan predecir el comportamiento bajo (i) condiciones de diseño y (ii) condiciones fuera de diseño o a carga parcial. En el modelo bajo condiciones de diseño, los datos de entrada son las especificaciones de operación deseadas para cada equipo, para determinar las características físicas que permitan estos requisitos. En el modelo a carga parcial, los datos de entrada son las características físicas y las condiciones de flujo las que determinan el comportamiento de los equipos bajo las condiciones dadas.

El presente trabajo se refiere a la implementación del modelo a carga parcial de equipos básicos en una planta de potencia con ciclo de vapor convencional. Por consiguiente, los datos de entrada son las características físicas del equipo y las condiciones de flujo de diseño y de operación.

En este trabajo, los equipos de plantas de vapor convencional que se analizarán son: Turbinas de Vapor, Calderas, Intercambiadores de calor, Condensadores, Válvulas y Bombas.

En este documento se condensa toda la información concerniente al desarrollo de este proyecto. En el capítulo uno, el procedimiento de cálculo a carga parcial que se utilizó para modelar cada equipo. En el capítulo dos, el funcionamiento de las interfaces de usuario de los programas desarrollados para cada equipo que utiliza el modelo a carga parcial. En el capítulo tres, un ejemplo de aplicación de las mismas. Por último se anexan los elementos de programación en lenguaje C++, los cuales contienen los algoritmos para los cálculos a carga parcial de cada equipo.

1. CÁLCULOS A CARGA PARCIAL

En este capítulo se presentan los modelos matemáticos que se implementaron para conocer el funcionamiento a carga parcial de los equipos anteriormente nombrados.

Estos modelos toman como regla general la relación entre los valores a carga nominal y los valores a carga parcial de un equipo en particular. Lo anterior, junto con algunas consideraciones, da como resultado la reducción de la cantidad de datos de entrada y la simplificación de los procedimientos de cálculos involucrados, puesto que los términos multiplicativos relacionados con el diseño físico permanecen constantes bajo condiciones de diseño, como en el caso a carga parcial, por lo tanto se cancelan. Los términos que no se cancelan son aquellos que están relacionados con las condiciones de flujo, los cuales no permanecen constantes bajo condiciones de operación a carga parcial.

1.1 TURBINA DE VAPOR

1.1.1 Generalidades. El modelamiento del comportamiento a carga parcial de las turbinas de vapor es complejo y se requiere de una habilidad considerable, además del conocimiento acerca del diseño y operación de estas unidades. Para analizar el comportamiento de las características de las turbinas de vapor, operando a condiciones de diseño como a carga parcial, es útil considerar por separado las características externas e internas del equipo.

Las características externas de la turbina se refieren a la habilidad del paso del flujo de la turbina en respuesta a los cambios en carga, temperatura y presión. Las características internas se refieren a las variaciones de la eficiencia de la sección con cambios en las condiciones de operación de la turbina. La labor de predecir el comportamiento de las turbinas de vapor a carga parcial puede llegar a ser una tarea

considerablemente fácil, considerando los ajustes de los parámetros anteriormente nombrados y las simplificaciones por suposiciones que se pueden hacer, tales como asumir que la turbina opera a una velocidad y a una contrapresión relativamente constante.

La ASME ha publicado dos reportes técnicos en 1967 y 1974³, los cuales ayudan a la comprensión del cálculo del comportamiento de las turbinas trabajando a carga parcial y a carga total. En lugar de realizar cálculos etapa por etapa, requerida por los métodos de diseño detallados, las metodologías utilizadas en estos reportes analizan la operación de una turbina, combinando grupos individuales de etapas dentro de secciones. Las correlaciones se desarrollaron para predecir el comportamiento de estas secciones usando datos recopilados en pruebas de flujo de calor, en unidades de turbinas reales y de la experiencia acumulada en cálculos etapa por etapa. El resultado fue una serie de curvas que pueden ser utilizadas para la predicción del desempeño de las turbinas tanto a carga total como a carga parcial. Es importante notar que estas correlaciones son aplicadas sección por sección, ya que no existe una correlación única para evaluar cómo se comporta una turbina desde su entrada hasta el extremo de condensación, a menos que el usuario esté analizando una turbina de una sola sección.

1.1.2 Cálculos. Los cálculos a carga parcial de las turbinas de vapor se basan en el procedimiento expuesto en el documento de la ASME⁴; este procedimiento está orientado hacia la determinación de la eficiencia y el cálculo de la energía disponible en la expansión del vapor, analizando sólo secciones de turbina de alta presión, turbina intermedia, turbina de recalentamiento, y turbinas con condensación y sin condensación.

³ SPENCER, R.C. COTTON, K.C. y CANNON, C.N. A method for predicting the performance of steam turbine-generators... 16,500 kw and larger. ASME, 1974.

⁴ Ibid.

Este método ofrece la posibilidad de predecir el desempeño de las turbinas de centrales de potencia de 17 MW o mayores, con descarga a un condensador. Las eficiencias se calculan a partir de determinados datos de diseño de una turbina como son: presiones, temperaturas a la admisión, relaciones de presión, caudales, diámetros pitch de la primera etapa (de gobierno) y la longitud de alabes de la última rueda, de acuerdo con la tabla 1 del documento de la ASME⁵.

Después de determinar la eficiencia de la sección se calcula la potencia (Kw.) generada por la misma, luego se determinan las pérdidas mecánicas en la turbina y en el generador, para así determinar la potencia neta.

Para la determinación de la presión de salida de cada sección en condiciones reales o de operación se utiliza el modelo de comportamiento basado en la elipse de Stodola anexo al documento de la ASME⁶.

El procedimiento seguido para la programación de los cálculos es el **Diagrama de Flujo para el Cálculo de Eficiencias en Turbinas** expuesto en el Informe General del Proyecto Clíope⁷ (ver anexo A).

1.2 INTERCAMBIADOR, CONDENSADOR Y CALDERA

1.2.1 Generalidades. El cálculo que se realiza para analizar los intercambiadores de calor se utiliza también en las calderas y condensadores. Las calderas se modelan como un equipo compuesto por diferentes elementos o secciones que realizan una transferencia de calor entre dos fluidos. Por lo tanto estos elementos se analizan como intercambiadores de calor por separado para la simplificación de los cálculos en dichas unidades. Los diferentes elementos tomados en cuenta son: el hogar, el sobrecalentador, el recalentador, el economizador y el calentador de aire. Para

⁵ Ibid., p. 12.

⁶ Ibid.

realizar este análisis existen muchos modelos matemáticos propuestos en la literatura, pero estos modelos son por lo general muy complejos y engorrosos para ser incorporados en un código que permita predecir el comportamiento de estas unidades. Es mucho más útil una serie de correlaciones normalizadas basadas en los principios de ingeniería, los cuales toman lo esencial de las variaciones del funcionamiento a carga parcial de intercambiadores de calor. Las correlaciones codificadas en este proyecto predicen las variaciones en las caídas de presión en el intercambiador de calor y en los coeficientes de transferencia de calor, variando las condiciones de operación de la planta.

1.2.2 Cálculos⁸. Para los cálculos fuera de diseño del lado tubo y lado casco se asume que los flujos son totalmente turbulentos; por lo tanto el factor de fricción f puede ser definido por la siguiente correlación:

$$f = \frac{-\Delta p / \rho g}{(l/D)(V^2 / 2g)} \quad (1)$$

Tomando la relación de caída de presión a condiciones de diseño y a condiciones de carga parcial, y usando además:

$$V = m / \rho A \quad (2)$$

y cancelando los términos comunes se tiene:

$$\frac{\Delta p}{\Delta p_{des}} = \left(\frac{f}{f_{des}} \right) \left(\frac{m^2 / \rho}{(m^2 / \rho)_{des}} \right) \quad (3)$$

⁷ LARA, Adriana, Informe general del proyecto Clíope.

⁸ ENTERSOFTWARE. Gate Cycle User's Guide: Appendix G, p.3.

Existen muchas correlaciones para expresar el factor de fricción en función de las condiciones de flujo, muchas de ellas tienen implícitamente el mismo factor, así que se tiene que resolver por iteración, pero para evitar tales cálculos se utiliza una expresión empírica de Karman-Nikuradse donde el factor f está de forma explícita:

$$f = 0.046 * Re^{-0.2} \quad (4)$$

Al sustituir la fórmula 4 y cancelar términos comunes, se simplifica la expresión a una de varias formas posibles tal como:

$$\frac{\Delta p}{\Delta p_{des}} = \left(\frac{m}{m_{des}} \right)^x \left(\frac{T}{T_{des}} \right)^y \left(\frac{P}{P_{des}} \right)^z \quad (5)$$

Donde los exponentes son definidos por el usuario para determinar el comportamiento de la caída de presión a carga parcial general del equipo; estos pueden tomar los valores entre 0 y 1.

Ya que la transferencia de calor es calculada con la ecuación general $Q = U * A * LMTD$, es necesario calcular un coeficiente de transferencia de calor a condiciones de diseño y luego calcular el coeficiente a carga parcial.

Para el cálculo del coeficiente a carga parcial se asume que el intercambiador de calor tiene una resistencia térmica balanceada, esto es, que la resistencia térmica del lado caliente es similar a la resistencia térmica del lado frío y puesto que los flujos son turbulentos, el coeficiente de transferencia de calor puede ser calculado por la siguiente ecuación:

$$Nu = C * Re^n * Pr^m \quad (6)$$

Donde C, n y m son constantes. La formula 6 es válida para un amplio rango de valores de número de Reynolds (4 a 400.000), dependiendo de los coeficientes y exponentes usados. Al realizar la relación entre valores de diseño y los valores a carga parcial, y puesto que U es proporcional a U_n , se obtiene:

$$\frac{U}{U_{des}} = \frac{Nu}{Nu_{des}} = \left(\frac{Re}{Re_{des}} \right)^n \left(\frac{Pr}{Pr_{des}} \right)^m \quad (7)$$

Sabiendo que el número de Reynolds depende del flujo másico y de la temperatura, y que el número de Prandtl depende de propiedades como el C_p y k , que a su vez dependen de la temperatura y la presión, se reduce la ecuación 7 a la siguiente expresión de una manera general para el cálculo del coeficiente de transferencia de calor en condiciones de carga parcial:

$$U = U_{des} \left(\frac{m}{m_{des}} \right)^x \left(\frac{T}{T_{des}} \right)^y \left(\frac{P}{P_{des}} \right)^z \quad (8)$$

Para el análisis de la transferencia de calor en estos equipos se utiliza la metodología de la LMTD (Diferencia de la temperatura media logarítmica), pero para ello se necesita que el usuario introduzca las condiciones del fluido caliente y frío en condiciones de operación a la entrada; con estos datos se suponen temperaturas y se calcula el calor transferido; luego se procede a calcular el calor por medio de la LMTD para verificar el valor de las temperaturas, y así sucesivamente hasta obtener una convergencia de la solución. Este procedimiento tiene una limitante al introducirse valores incoherentes por parte del usuario, por lo cual nunca se llegaría a una solución convergente o lógica.

Se debe tener en cuenta que aunque los cálculos de transferencia de calor son los mismos para los intercambiadores, condensadores y calderas, cada equipo funciona

de una forma diferente; por ejemplo el intercambiador puede tener la configuración de flujos en contracorrientes o unidireccionales, en cambio el condensador solo tiene la configuración de flujos contracorriente; y en las calderas se manejan otros tipos de fluidos además del agua, como los gases de combustión y aire real. Lo anterior se toma en cuenta para la implementación de diferentes elementos de programación que tienen los mismos tipos de cálculos de acuerdo con el mismo procedimiento explicado anteriormente.

El diagrama de flujo seguido para la programación de los cálculos es el **Diagrama de flujo para intercambiadores de calor** expuesto en el informe general del proyecto Clíope⁹ (ver anexo B).

1.3 VÁLVULAS

1.3.1 Generalidades. El único cálculo necesario para el análisis a carga parcial de estos equipos es el calculo de la caída de presión, basándonos en la teoría termodinámica, la cual declara que en una válvula la entalpía de entrada es la misma que a la de salida, o sea, que conociendo la caída de presión a través de ella y las condiciones de entrada, se calculan las propiedades termodinámicas a la salida.

1.3.2 Cálculos. La expresión utilizada en el código es la correlación que se usa en hidráulica¹⁰ para conocer la caída de presión en válvulas:

$$C_v = Q\sqrt{\Delta p} \quad (9)$$

Donde C_v es el coeficiente de capacidad, Q el caudal y Δp es la caída de presión a través de la válvula. Con la anterior expresión también se relacionan las variables de

⁹ LARA, Adriana, Op. cit.

¹⁰ CRANE, Flow of fluids Through valves, fittings, and pipe, McGraw Hill.

flujo en las condiciones de diseño y de carga parcial, lo cual da una expresión que calcula la caída de presión con un comportamiento cuadrático.

$$\Delta p_{real} = \Delta p_{diseño} * \left(\frac{Q_{real}}{Q_{diseño}} \right)^2 \quad (10)$$

Se debe anotar que con la anterior expresión se predice el comportamiento en las válvulas que se encuentran totalmente abiertas y no cuando se han cerrado hasta cierto punto para restringir el flujo. Para pronosticar cómo se comportan las válvulas cuando se han cerrado, se debe cambiar de modelo. De acuerdo con la literatura¹¹ existen dos conductas características del flujo en la válvula cuando está entre abierta, las cuales son: de tipo Linear y Equal (Exponencial); las figuras 1 y 2 muestran el comportamiento del flujo contra el porcentaje de apertura de la válvula.

La fórmula matemática para predecir el comportamiento de una válvula con característica de flujo Linear es:

$$Q_o = K * L \quad (11)$$

Donde Q_o es el caudal de operación, K constante de proporcionalidad y L es el porcentaje de apertura de la válvula.

La fórmula matemática para una característica de flujo Exponencial es:

$$Q_o = Q_{mc} * e^{K*L} \quad (12)$$

¹¹ PHILIP, L. Skousen. Valve Handbook

Donde Q_0 es el caudal de operación, K constante, L es el porcentaje de apertura de la válvula, e es el número de Euler igual a 2.718 y Q_{mc} es el caudal mínimo controlable.

Fig. 1 Característica de flujo lineal a través de las válvulas.

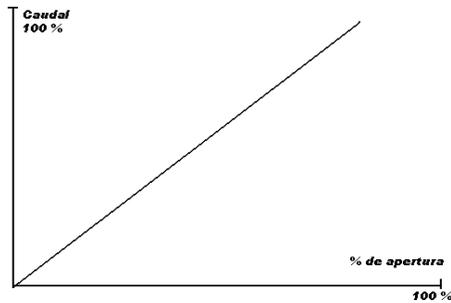
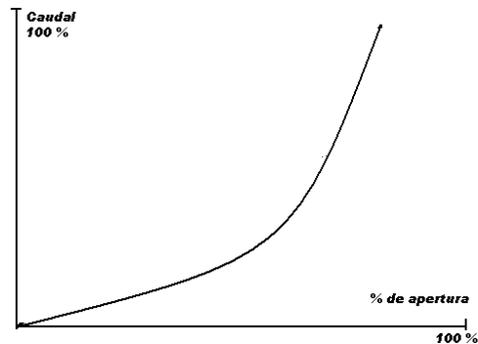


Fig. 2 Característica de flujo Equal (Exponencial) a través de las válvulas.



En la teoría de válvulas¹² se ha desarrollado una correlación donde se calcula el coeficiente de capacidad de acuerdo a dos características de la válvula en operación: la rangeabilidad y la apertura. Luego de conocer el coeficiente de capacidad en la nueva posición se procede a calcular la caída de presión. Las siguientes son las

¹² J.W. Hutchison. Handbook of control valves

ecuaciones que calculan la característica C_v de la válvula cuando se encuentra en esta condición:

$$C_{v,x} = C_v(0.96X + 0.03) \quad (13) \text{ Para el comportamiento Linear.}$$

$$C_{v,x} = C_v R^{X+1} \quad (14) \text{ Para el comportamiento Exponencial.}$$

Donde C_v es el coeficiente de capacidad de la válvula totalmente abierta, X el porcentaje de apertura de la válvula, R la rangeabilidad de la válvula y C_{vx} el coeficiente de capacidad a una apertura dada.

El procedimiento para realizar los cálculos es el siguiente:

- Se define que tipo de comportamiento tiene la válvula, si esta totalmente abierta o tiene un comportamiento lineal o exponencial.
- Si tiene un comportamiento de válvula totalmente abierta, se calcula directamente la caída de presión con la expresión (10).
- Si la válvula esta entre abierta, se calcula primero el coeficiente de capacidad a la apertura en que se encuentra con la expresión (13) o (14) dependiendo del comportamiento elegido por el usuario.
- Se calcula la caída de presión con la expresión: $\Delta p_o = \left(\frac{C_{vx}}{Q_o} \right)^2$, pero con el caudal de operación y el coeficiente de capacidad calculado anteriormente.

El diagrama de flujo seguido para la programación de los cálculos se muestra en el anexo C.

1.4 BOMBAS

1.4.1 Generalidades. Las bombas aquí analizadas son del tipo centrífugo, ya que estas son las que se encuentran en las plantas de potencia con ciclo de vapor convencional. Por lo tanto, para el análisis del comportamiento son usadas las leyes de la afinidad que gobierna este tipo de bombas.

Las leyes de afinidad¹³ son derivadas de análisis adimensional de máquinas rotatorias. Estas leyes muestran que para ciertas condiciones relativamente comunes, ciertos parámetros permanecían constantes. Estas leyes declaran que cuando la velocidad cambia, la capacidad es directamente proporcional a ella; la cabeza o carga es proporcional al cuadrado de la velocidad; y la potencia es proporcional al cubo de la ella. Las relaciones matemáticas entre estas variables son las siguientes:

$$Q_2 = Q_1 \left(\frac{N_2}{N_1} \right) \quad (15)$$

$$H_2 = H_1 \left(\frac{N_2}{N_1} \right)^2 \quad (16)$$

$$BHP_2 = BHP_1 \left(\frac{N_2}{N_1} \right)^3 \quad (17)$$

Donde Q_1 , H_1 , BHP_1 , y N_1 son la capacidad, la cabeza, la potencia y velocidad en condiciones de diseño y Q_2 , H_2 , BHP_2 , y N_2 son la capacidad, la cabeza, la potencia y velocidad en condiciones de operación.

Usando estas leyes pueden ser construidas las curvas de funcionamiento a cualquier velocidad dada. Para cualquier punto sobre la curva estándar pueden ser encontrados puntos equivalentes sobre la nueva curva de velocidad los cuales tienen condiciones

hidráulicas idénticas; flujo de fluido más exacto, balance del empuje del impulsor y la eficiencia del bombeo.

1.4.2 Cálculos. El procedimiento usado para los cálculos es el siguiente:

- Se calcula la relación que hay entre la velocidad inicial o de diseño de la bomba con la velocidad de operación. $\left(\frac{N_2}{N_1}\right)$.
- Se calcula directamente el caudal de operación utilizando la expresión (15).
- Se calcula directamente la carga (cabeza) de la bomba de operación utilizando la expresión (16).
- Se calculan directamente la potencia de la bomba de operación utilizando la expresión (17).

El diagrama de flujo seguido para la programación de los cálculos se muestra en el anexoD.

¹³ KARASSIK KRUTZSCH, Fraser Messina. Pump Handbook. second edition. McGraw Hill.

2. MANUAL DEL USUARIO

En este capítulo se explica el funcionamiento de las interfaces de usuario de los programas desarrollados para cada equipo que utiliza el modelo a carga parcial. Por consiguiente se enseñan sus componentes, controles y su funcionalidad dando a conocer como se introducen los datos, como se realizan los cálculos y como se revisan los resultados.

La primera parte del manual de usuario de cada programa llamada interfaz de usuario muestra todos los comandos de la ventana principal y define su funcionalidad. La segunda parte tiene por objetivo indicar cuales son los datos de entrada requeridos. La tercera parte enumera de una manera secuencial los pasos para llevar a cabo los cálculos a carga parcial.

2.1 PROGRAMA TURBINA_CARGA_PARCIAL.EXE.

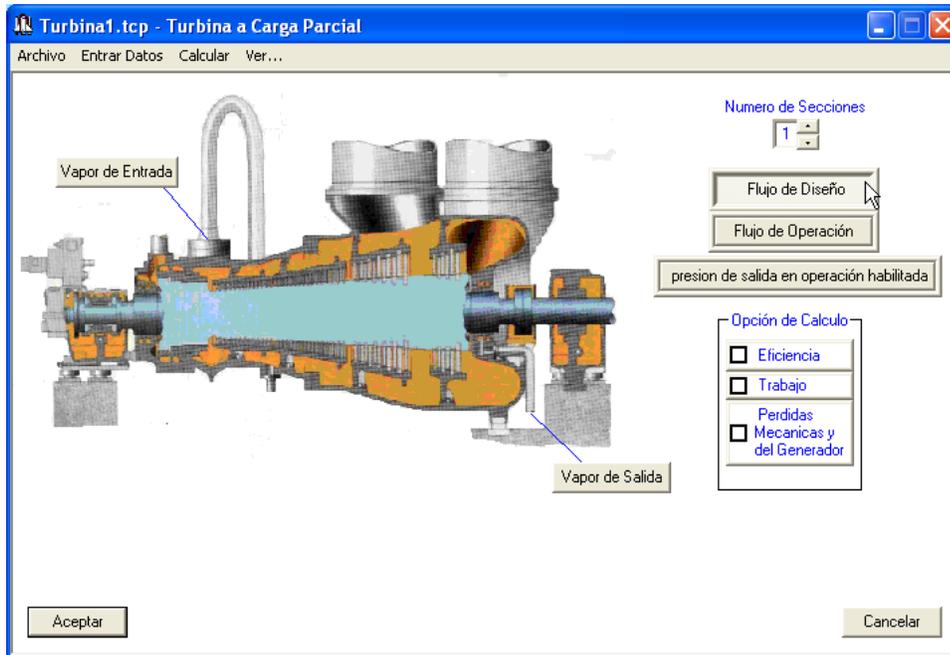
El programa Turbina_Carga_Parcial.exe es un archivo ejecutable que tiene la capacidad de calcular la eficiencia, potencia y pérdidas mecánicas de una turbina a vapor y además las pérdidas en el generador en condiciones de operación.

2.1.1 Interfaz de Usuario. La figura número 3, muestra la ventana principal del programa; en ella se observan la barra de título, el icono y los comandos minimizar y cerrar comunes de Windows, debajo se encuentra la barra de menú y el área de trabajo.

En la figura número 4 se pueden observar los elementos del menú **Archivo**, los cuales son: **Crear Archivo de Datos**, **Leer Archivo de Datos**, **Guardar Archivo**, **Remover Archivo** y **Salir**.

- **Crear Archivo de Datos:** Abre la ventana de Windows *Guardar como* para la creación de un archivo de datos de la turbina con extensión “.tcp”.

Fig. 3 Presentación Turbina_Carga_Parcial.exe

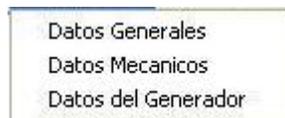


- **Leer Archivo de Datos:** Abre la ventana de Windows *Abrir* para la búsqueda de un archivo ya creado.
- **Guardar Archivo:** Guarda el archivo de datos en el cual se esta trabajando para poder crear o abrir otro.
- **Remove Archivo:** Borra o elimina todos los archivos de datos generados por el programa que tenga el nombre del archivo que se encuentre cargado.
- **Salir:** Se sale del programa.

Fig. 4 Elementos del menú Archivo de la turbina



Fig. 5 Elementos del menú Entrar Datos de la turbina.



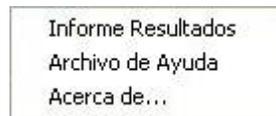
En el menú **Entrar Datos** se encuentran tres elementos que son: **Datos Generales**, **Datos Mecánicos** y **Datos del Generador**.

- **Datos Generales:** Abre una ventana para la introducción de las características generales de cada sección de la turbina.
- **Datos Mecánicos:** Abre una ventana para la introducción de algunos datos del diseño mecánico de las secciones de la turbina.
- **Datos del Generador:** Abre una ventana por medio de la cual se introducen datos tanto de la turbina como del generador.

El comando **Calcular** es el elemento del menú que lleva a cabo los cálculos a carga parcial seleccionados, los cuales pueden ser: cálculo de las eficiencias, la potencia generada y las pérdidas mecánicas de la turbina y las pérdidas mecánicas en el generador.

El menú **Ver...** (Ver Fig. 6) contiene también tres elementos, los cuales son: **Informe Resultados**, **Archivo de Ayuda** y **Acerca de...**

Fig. 6 Elementos del menú Ver... de la turbina



- **Informe Resultados:** Abre el archivo de texto donde se encuentran los resultados.
- **Archivo de Ayuda:** Abre el documento de ayuda.
- **Acerca de...:** Abre la ventana donde aparece el nombre y la versión del programa, quien lo desarrolló y quien dirigió su realización.

Fig. 7 Control del número de secciones de la turbina



En la figura 7 se muestra el control que maneja la selección del número de secciones de la turbina, siendo el valor mínimo 1 y el máximo 5, advirtiendo que no se puede cambiar directamente sino solo por el uso de las flechas.

La figura 8 muestra dos botones de selección única, los cuales al operarlos se define si los datos que entran en los flujos de la turbina son datos termodinámicos en condiciones de diseño o de operación.

Fig. 8 Controles de datos en los flujos de la turbina

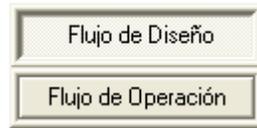
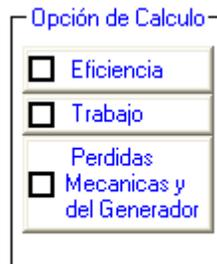


Fig. 9 Control de selección de entrada de la presión de salida en condiciones de operación.



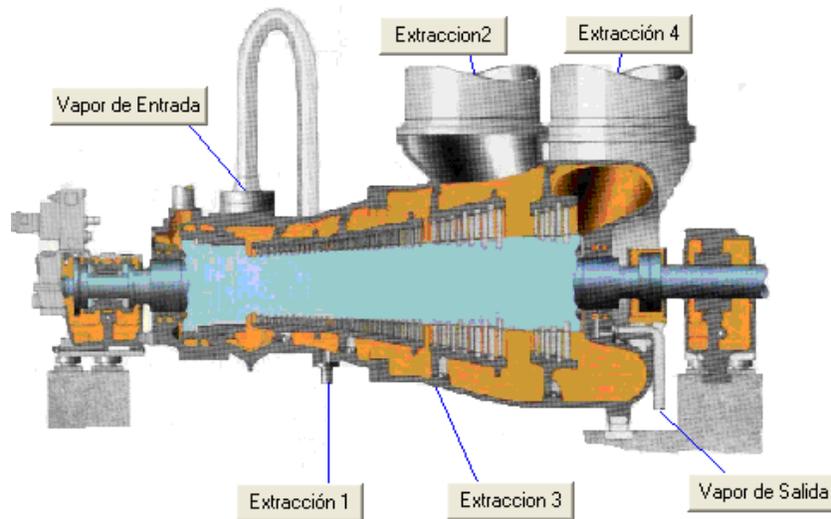
En la figura 9 se muestra el control para indicar si se quiere entrar la presión de salida en una sección determinada de la turbina en condiciones de operación o si se requiere que sea calculada por el programa.

Fig. 10 Opciones de cálculo del programa de la turbina.



En la figura 10 se muestran las tres opciones de cálculo del programa como son: la eficiencia de cada sección de la turbina, la potencia generada y las pérdidas mecánicas tanto en la turbina como en el generador. Se puede seleccionar cualquiera de estas opciones indistintamente una de la otra.

Fig. 11 Controles de los flujos de la turbina.



En la figura 11 se muestran los botones que abren las ventanas para entrar los datos termodinámicos de los diferentes flujos de la turbina de acuerdo al número de secciones.

2.1.2 Datos de entrada. Para llevar a cabo los cálculos a carga parcial en una turbina a vapor con ciclo convencional se requieren los siguientes datos: Datos de flujos de vapor, datos generales y mecánicos de las secciones y datos del generador.

Los datos de los flujos de vapor en la turbina, tanto en condiciones de diseño como a carga parcial son la temperatura, la presión y el flujo másico (ver Fig. 12).

Los datos generales de las secciones son el tipo de turbina (ver Fig. 13), número de flujos paralelos a la entrada, número de válvulas de control, diámetro pitch de la etapa de gobierno, presión de salida de la etapa de gobierno, presión transversal en una sección a 1800 rpm; estos datos son requeridos dependiendo del tipo de turbina. En cuanto a los tipos de turbina son 7, a saber:

Fig. 12 Datos termodinámicos de los flujos de la turbina.



- Turbina tipo A: sin recalentamiento, 3600 rpm, con condensación, dos etapas de gobierno.
- Turbina tipo B: alta presión, 3600 rpm, sin condensación, una etapa de gobierno.
- Turbina tipo C: alta presión, 3600 rpm, sin condensación, dos etapas de gobierno.
- Turbina tipo D: sección intermedia, 3600 rpm, sin condensación, sin etapa de gobierno.
- Turbina tipo E: sección recalentada, 3600 rpm, con condensación, sin etapa de gobierno.
- Turbina tipo F: sección recalentada, 3600/1800 rpm, con condensación, sin etapa de gobierno.
- Turbina tipo G: sección recalentada, 1800 rpm, con condensación, sin etapa de gobierno.

Fig. 13 Datos de las secciones de la turbina.

The image shows a software window titled "Entre datos para calcular la eficiencia de la turbina por secciones". It contains five rows, each labeled "Sección 1" through "Sección 5". Each row has a dropdown menu with the text "Sin recalentamiento-3600 rpm-condensación-2 etapas de gobierno" and a button labeled "Entre datos generales". At the bottom of the window are "OK" and "Cancel" buttons.

Los datos mecánicos de la turbina (ver Fig. 14) son la altura del último alabe de la turbina, el diámetro de este alabe, el área anular de la última etapa y las rpm.

Los datos del generador (ver Fig. 15) son los KVAs, los Kilowatios que puede generar, la presión del hidrógeno tanto en condiciones de diseño como en operación, el tipo de enfriamiento utilizado, las rpm y una característica mecánica del generador.

Fig. 14 Datos mecánicos de una sección de la turbina.

Datos Mecánicos de la Sección 1

Datos termodinámicos a carga parcial

Presión de entrada: 500.23 psia

Temperatura de entrada: 950 F

Flujo másico: 2.15e+006 Lb/hr

Presión de salida: 415.728 psia

características geométricas

Altura del alabe última etapa: 0 pulg

Díametro del alabe última etapa: 0 pulg

Area anular de la última etapa: 32.56 ft²

Con condensación Entrar Area

Eficiencia de la sección: 85.8166 %

sistema unidades

internacional

inglesas

métrico

RPM del eje de la turbina

3600 rpm

1800 rpm

OK Cancel

Fig. 15 Datos del generador de la turbina.

Turbina1.gnd - Turbina a Carga parcial

KVA del Generador: 204000

KW's del Generador: 170000

Presión del hidrogeno en condiciones de diseño: 420 psia

Presión del hidrogeno en condiciones de operación: 420 psia

Tipo de enfriamiento

con conductor

convencional

Factor de Potencia: 0.85

Velocidad del eje

1800 rpm

3600 rpm

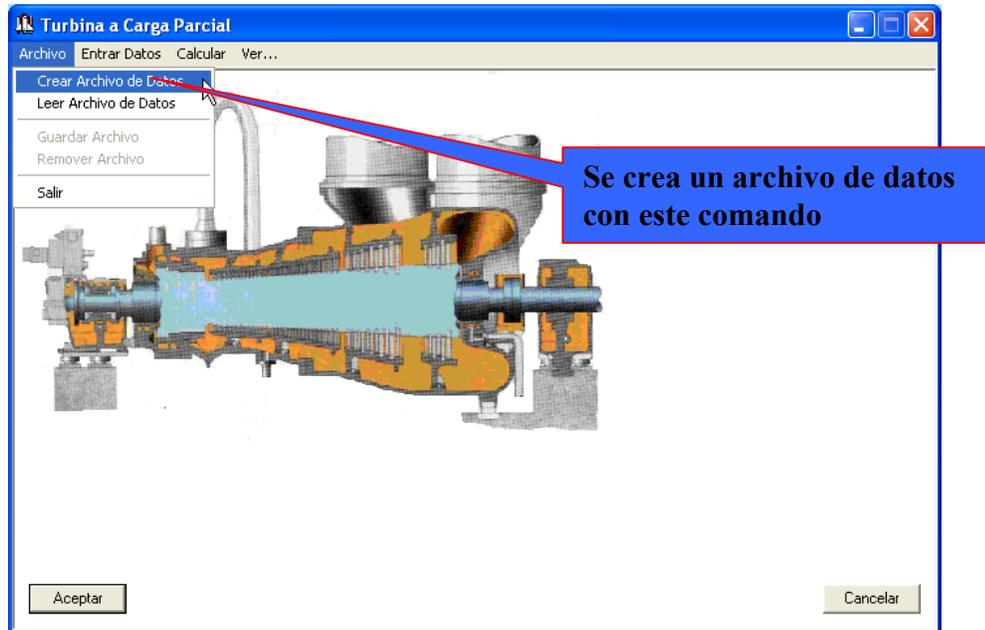
carcasa unica y TCDF a 3600 rpm

OK Cancel

2.1.3 Funcionamiento. Pasos para llevar a cabo los cálculos a carga parcial con el programa Turbina_Carga_Parcial.exe.

1. Cree un archivo de datos: Oprima el comando **Crear Archivo de Datos** en el menú **Archivo** como se muestra en la figura 16. Al oprimir el comando se abre la ventana **Guardar como** de Windows (ver Fig. 17), donde se le da un nombre al archivo de datos y se selecciona el directorio donde se guarda. Después de oprimir el botón **Guardar** aparecen los comandos del programa.

Fig. 16 Creando archivo de datos de la turbina



2. Seleccione el número de secciones de la turbina: Opere las flechas del control de la figura 7, después de lo cual aparecerán los diferentes botones correspondientes a las extracciones que la turbina puede tener (ver Fig. 11 y 18).

Fig. 17 Guardar como... de la turbina



3. Entre los datos de los flujos de vapor en condiciones de diseño (ver Fig. 19): Para esto oprima el comando **Flujo de Diseño** de la figura 8 y luego oprima cada botón de los flujos de la turbina que ha determinado anteriormente (paso 2); después de lo cual aparecerá la ventana de la figura 12.

4. Entre los datos de los flujos de vapor en condiciones de operación (ver Fig. 19): Para esto oprima el comando **Flujo de Operación** de la figura 8 y luego oprima cada botón de los flujos de la turbina que ha determinado anteriormente (paso 2); después de lo cual aparecerá la ventana de la figura 12. Si desea entrar la presión de salida en operación para cada sección oprima el botón **presión de salida en operación habilitada** (ver Fig. 20).

5. Seleccione los parámetros que desea calcular a carga parcial (ver Fig. 21): Seleccione uno o varios elementos de la figura 10.

6. Entre los datos generales de las secciones de la turbina (ver Fig. 22): (Si se ha seleccionado como parámetro a calcular a carga parcial la **eficiencia** en el paso 5, entonces se requiere introducir estos datos, de lo contrario no son requeridos) Oprima

el elemento **Datos Generales** del menú **Entrar Datos** de la barra de menú (ver Fig. 5), después de lo cual aparecerá la ventana de la figura 13. En esta ventana se elige el tipo de turbina por cada sección y luego se oprime el botón de enfrente llamado **Entre datos generales** para entrar otros datos necesarios dependiendo del tipo de turbina escogido. Si se opta por una turbina tipo D se deshabilita el botón, ya que no es necesario entrar más datos.

Fig. 18 Seleccionando las secciones de la turbina.

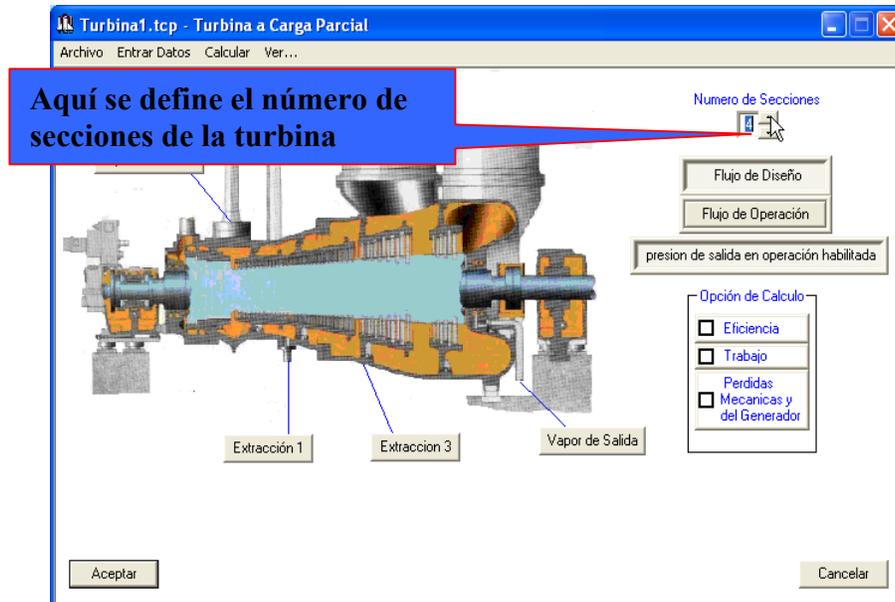


Fig. 19 Seleccionando las condiciones de flujo de la turbina.

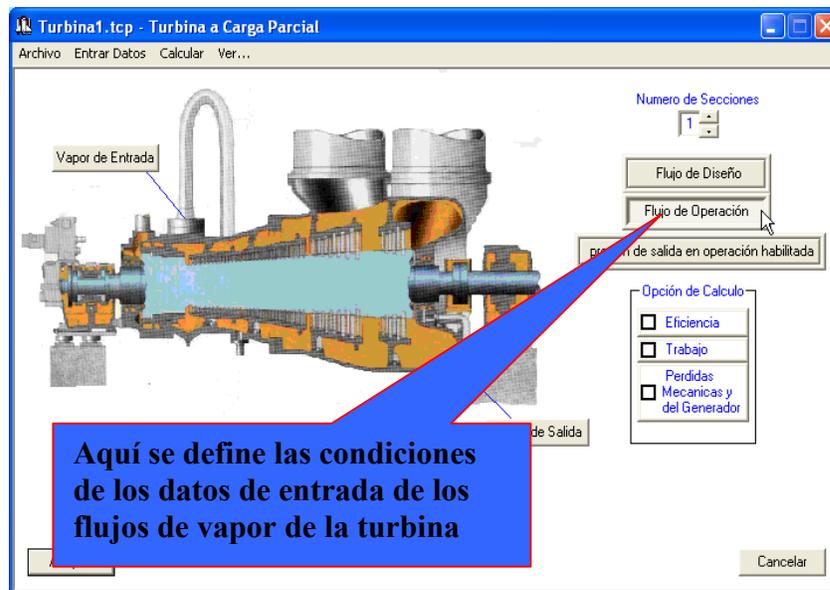


Fig. 20 Seleccionando presión de salida en operación habilitada.

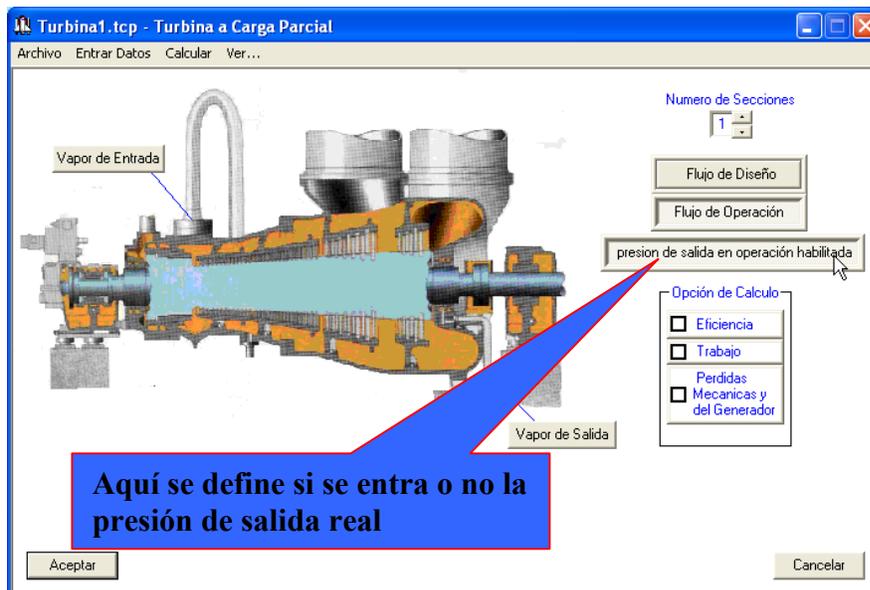


Fig. 21 Seleccionando los parámetros a calcular en la turbina.

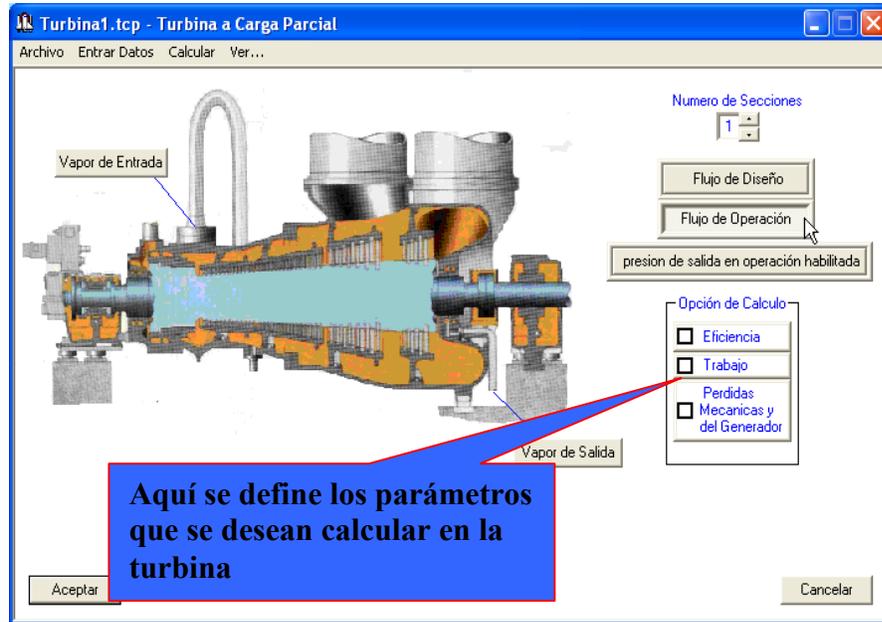
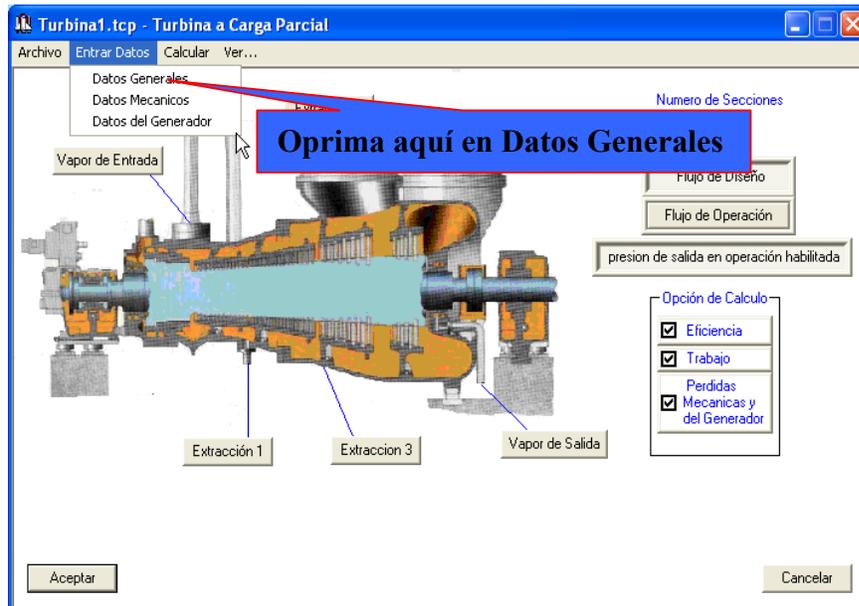
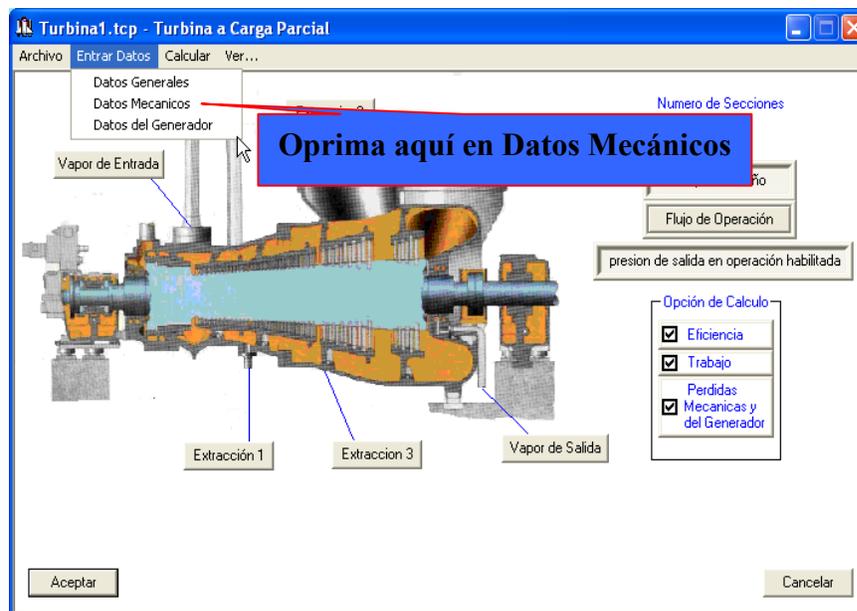


Fig. 22 Entrando datos generales de las secciones de la turbina.



7. Entre los datos mecánicos de las secciones de la turbina (ver Fig. 23): (Si se ha seleccionado como parámetro a calcular a carga parcial el **trabajo** o potencia generada en la turbina en el paso 5 es necesario introducir estos datos, de lo contrario no son necesarios) Oprima el elemento **Datos Mecánicos** del menú **Entrar Datos** de la barra de menú (ver Fig. 5); después de lo cual aparece la ventana de la figura 15, por medio de la cual se introducen los datos mecánicos de la sección al oprimir el botón aceptar de la ventana. Por medio de esta ventana es posible también entrar datos termodinámicos y de eficiencia conocidos en el caso de que se desee calcular solamente el trabajo pero no la eficiencia.

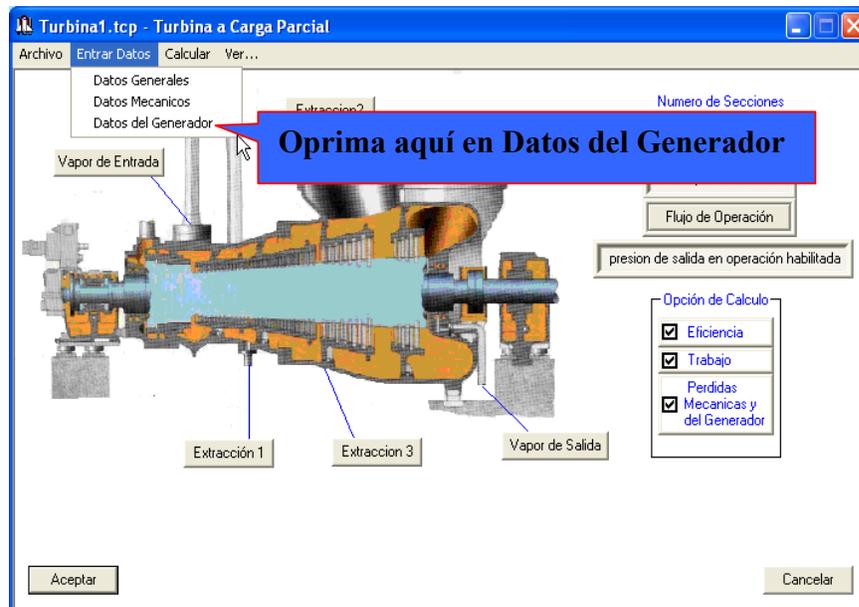
Fig. 23 Entrando datos mecánicos de las secciones de la turbina.



8. Entre los datos del generador (ver Fig. 24): (Si se ha seleccionado como parámetro a calcular a carga parcial las pérdidas mecánicas del generador y de la turbina en el paso 5 es necesario introducir estos datos, de lo contrario no son necesarios) Oprima el elemento **Datos del Generador** del menú **Entrar Datos** de la barra de menú (ver Fig. 5); después de lo cual aparece la ventana de la figura 16, por

medio de la cual se introducen los datos del generador al oprimir el botón aceptar de la ventana.

Fig. 24 Entrando datos del generador.



9. Realice los cálculos a carga parcial: Oprima el comando **Calcular** del menú (ver Fig. 25). Cuando el programa ya ha terminado los cálculos informa con la ventana de la figura 26 si se desea ver los resultados o no.

10. Guarde o elimine los datos introducidos: Por último se puede conservar todos los datos introducidos anteriormente en archivos que tienen el mismo nombre del archivo de datos creado al principio accionando el comando **Guardar Archivo** del menú **Archivo** o eliminarlos accionando el comando **Remover Archivo**.

Fig. 25 Comando calcular de la turbina.

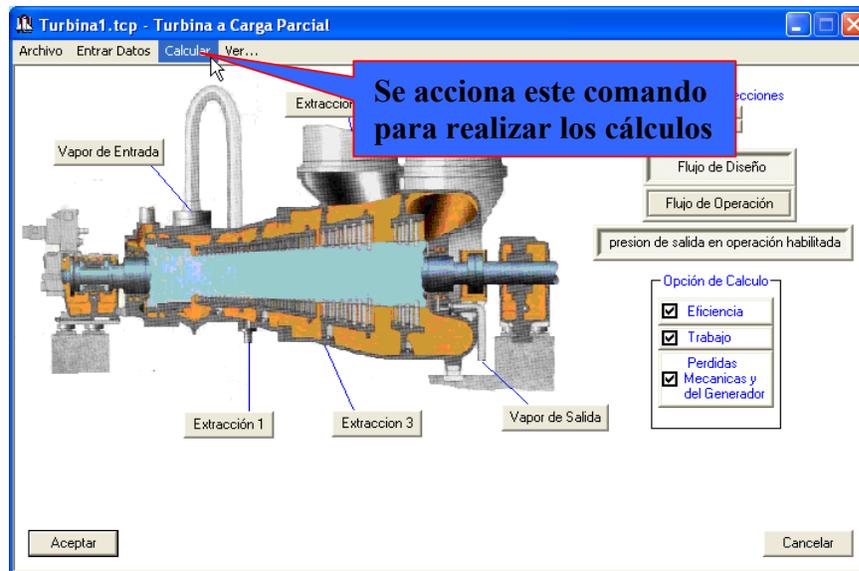
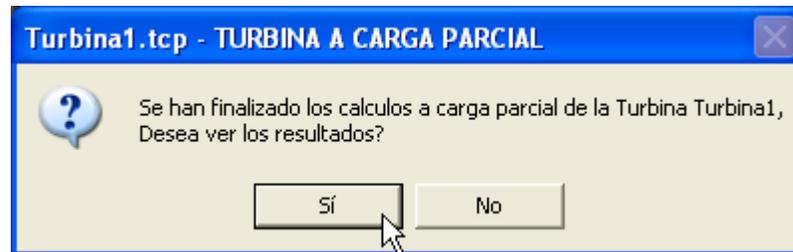


Fig. 26 Fin de los cálculos de la turbina.



Los resultados de los cálculos a carga parcial de la turbina son mostrados por medio del programa WordPad de Windows; para verlos se oprime el elemento **Informe Resultados** del comando **Ver...** del menú (ver Fig. 6 y 27); después de lo cual se abre la ventana de la figura 28.

Fig. 27 Revisando los resultados de la turbina.

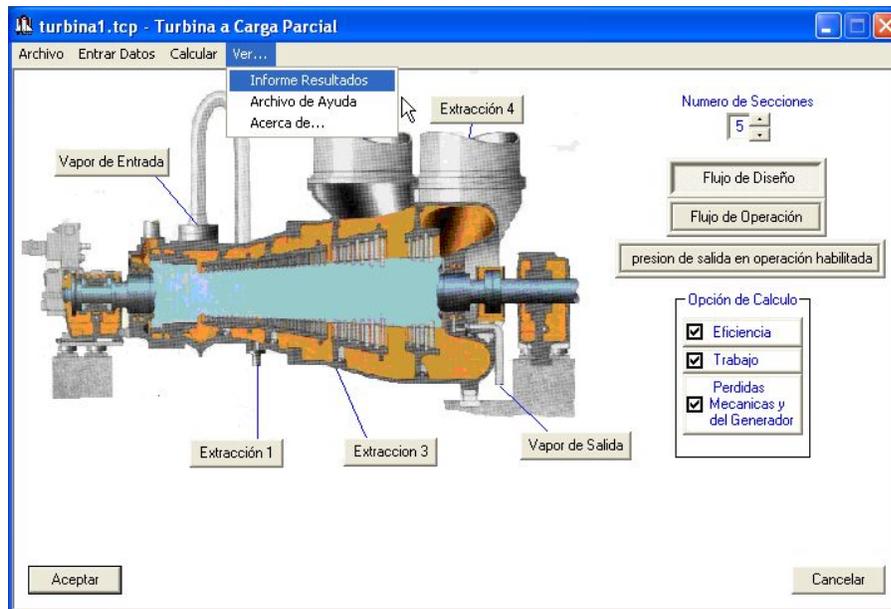
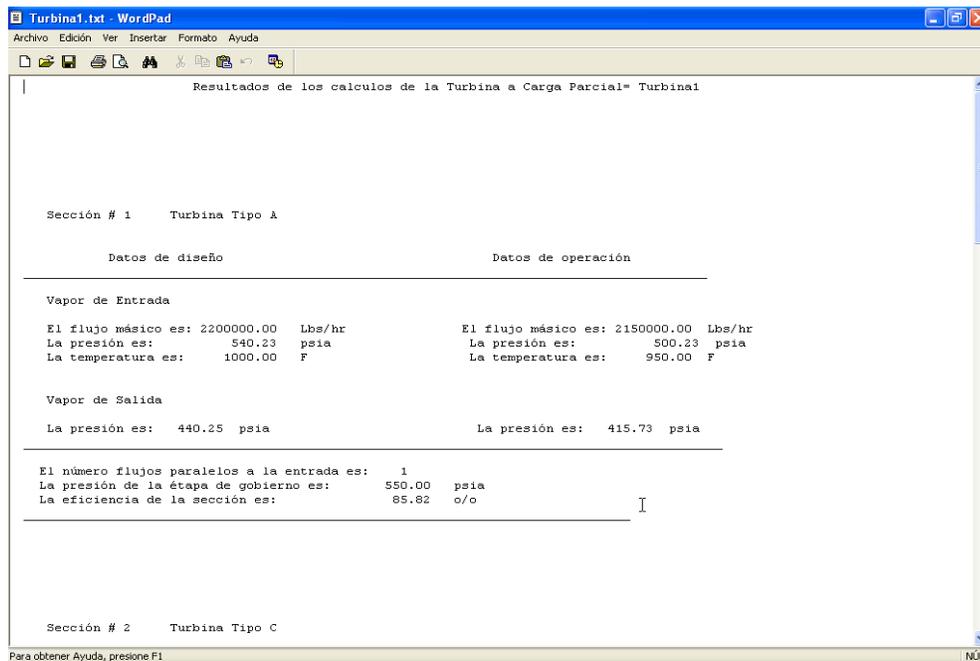


Fig. 28 Archivo de texto con los resultados de la turbina.

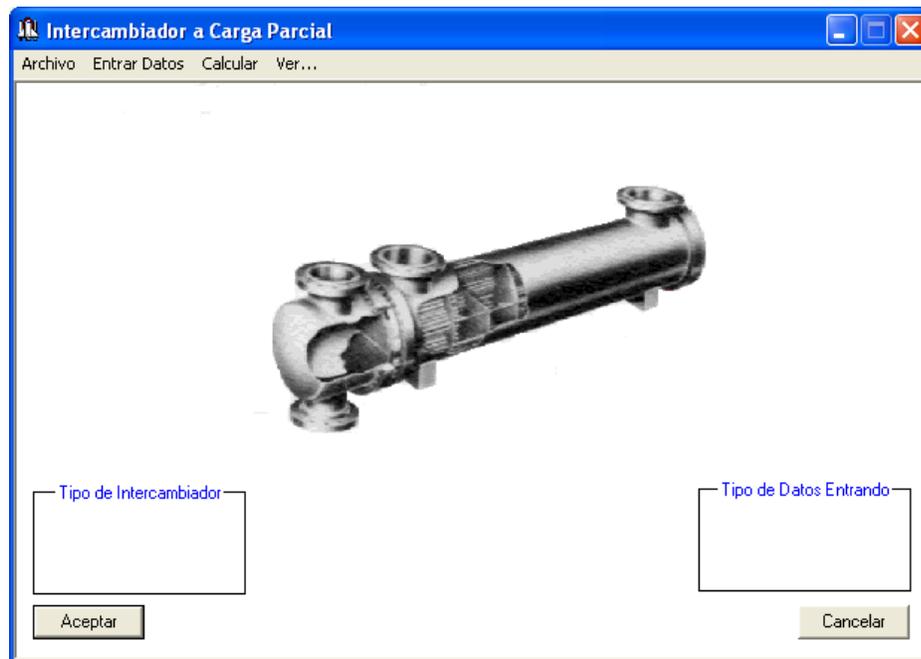


2.2 PROGRAMA INTERCAMBIADOR_CARGA_PARCIAL.EXE.

El programa Intercambiador_Carga_Parcial.exe es un archivo ejecutable que tiene la capacidad de calcular las caídas de presión y las temperaturas de salida en el lado casco y tubo en condiciones de operación de intercambiadores de calor; siendo tanto el fluido caliente como el fluido frío el agua.

2.2.1. Interfaz de Usuario. La figura 29 muestra la ventana principal del programa; en ella se observan la barra de título, el icono y los comandos minimizar y cerrar comunes de Windows; debajo se encuentra la barra de menú y el área de trabajo.

Fig.29 Presentación de Intercambiador_Carga_Parcial.exe



En la figura 30 se pueden ver los elementos del menú **Archivo** que son: **Crear Archivo de Datos, Leer Archivo de Datos, Guardar Archivo, Remover Archivo y Salir.**

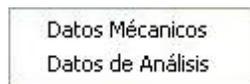
- **Crear Archivo de Datos:** Abre la ventana de Windows *Guardar como* para la creación de un archivo de datos de intercambiador de calor con extensión “.int”.
- **Leer Archivo de Datos:** Abre la ventana de Windows *Abrir* para la búsqueda de un archivo ya creado.
- **Guardar Archivo:** Guarda el archivo de datos en el cual se está trabajando para poder crear o abrir otro.

Fig. 30 Elementos del menú Archivo del intercambiador



- **Remover Archivo:** Borra o elimina todos los archivos de datos generados por el programa que tenga el nombre del archivo que se encuentre cargado.
- **Salir:** Termina la ejecución del programa.

Fig. 31 Elementos del menú Entrar Datos del intercambiador.



En el menú **Entrar Datos** se encuentran dos elementos que son: **Datos Mecánicos** y **Datos de Análisis** (ver Fig. 31).

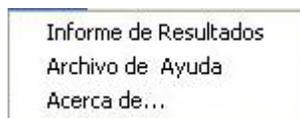
- **Datos Mecánicos:** Abre una ventana en la cual se introducen algunos datos del diseño mecánico del intercambiador.
- **Datos de Análisis:** Abre una ventana en la cual se introducen los exponentes x, y, z de análisis para determinar el comportamiento del equipo.

El comando **Calcular** del menú se oprime para realizar los cálculos a carga parcial.

El menú **Ver...** (Ver Fig. 32) contiene también tres elementos que son: **Informe de Resultados**, **Archivo de Ayuda** y **Acerca de...**

- **Informe de Resultados:** Abre el archivo de texto donde se encuentran los resultados.
- **Archivo de Ayuda:** Abre el documento de ayuda.
- **Acerca de...:** Abre la ventana donde aparece el nombre y la versión del programa, quien lo desarrolló y quien dirigió su realización.

Fig. 32 Elementos del menú Ver... del intercambiador.



La figura 33 muestra los botones que abren las ventanas para introducir los datos termodinámicos de los diferentes flujos del intercambiador.

La figura 34 muestra los controles de selección única para la escogencia del tipo de datos termodinámicos en los flujos del equipo: si son datos en condiciones de diseño o en condiciones de operación.

Fig. 33 Flujos del intercambiador.

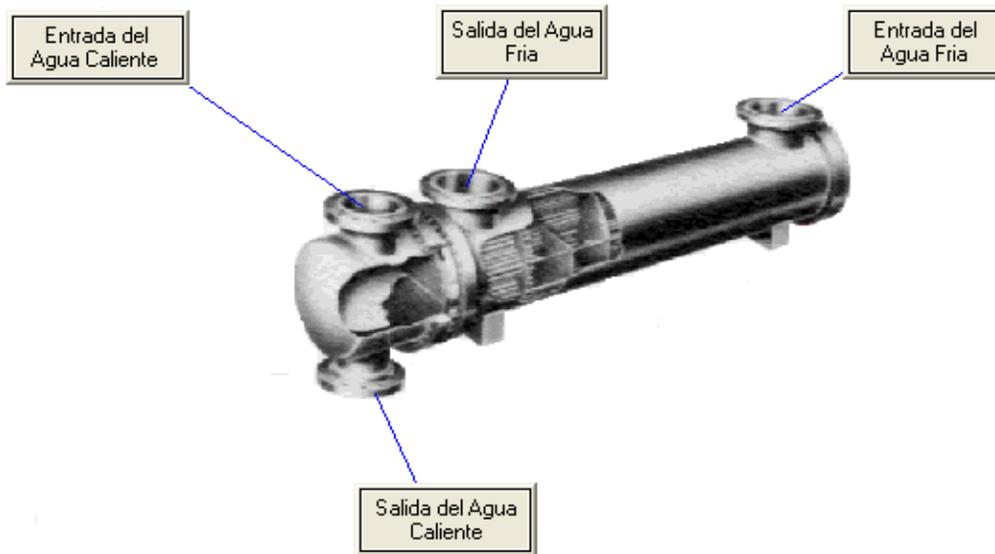


Fig.34 Tipo de datos de entrada del intercambiador.



La figura 35 muestra los controles de selección única para la elección del tipo de intercambiador: si es un intercambiador en contracorriente o uno en flujo paralelo.

Fig. 35 Controles para la selección del tipo de intercambiador.



2.2.2 Datos de entrada. Para llevar a cabo los cálculos a carga parcial en un intercambiador se requieren los siguientes datos: Datos de los flujos del agua fría y del agua caliente, datos mecánicos del intercambiador y exponentes para determinar el comportamiento del equipo.

Los datos de los flujos del agua fría y del agua caliente, tanto en condiciones de diseño como a carga parcial son la temperatura, la presión y el flujo másico (ver Fig. 36).

Fig. 36 Datos termodinámicos de los flujos del intercambiador.



El único dato mecánico requerido es el área total de transferencia de calor, pero en el caso de que se conozca el coeficiente de transferencia de calor global en condiciones de diseño es posible introducirlo al programa como dato mecánico conocido (ver Fig. 37).

Los datos para determinar el comportamiento del equipo (ver Fig. 38) son los exponentes x , y y z que se usan para calcular la caída de presión en el lado casco, la caída de presión en el lado tubo y el coeficiente de transferencia de calor global en condiciones de operación (ver ecuaciones 5 y 8). Los exponentes x toman los valores de a_0 , a_1 y a_2 . Los exponentes y toman los valores de b_0 , b_1 y b_2 . Los exponentes z toman los valores de c_0 , c_1 y c_2 .

Fig. 37 Datos mecánicos del intercambiador.

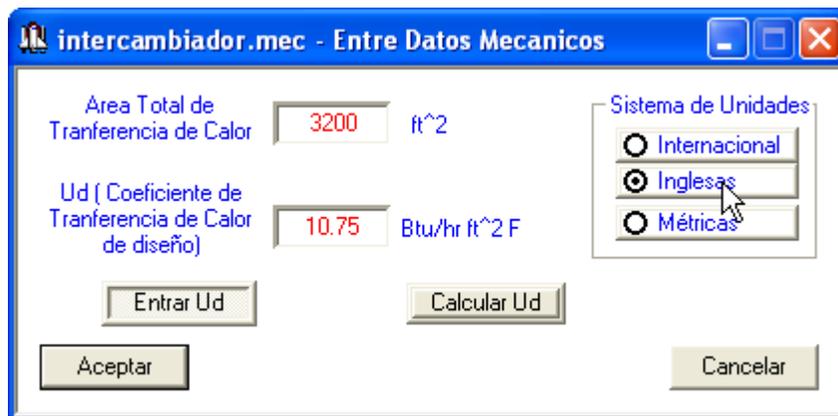
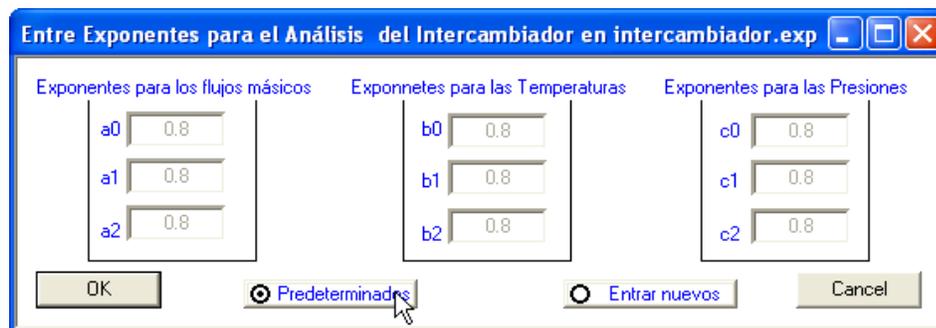


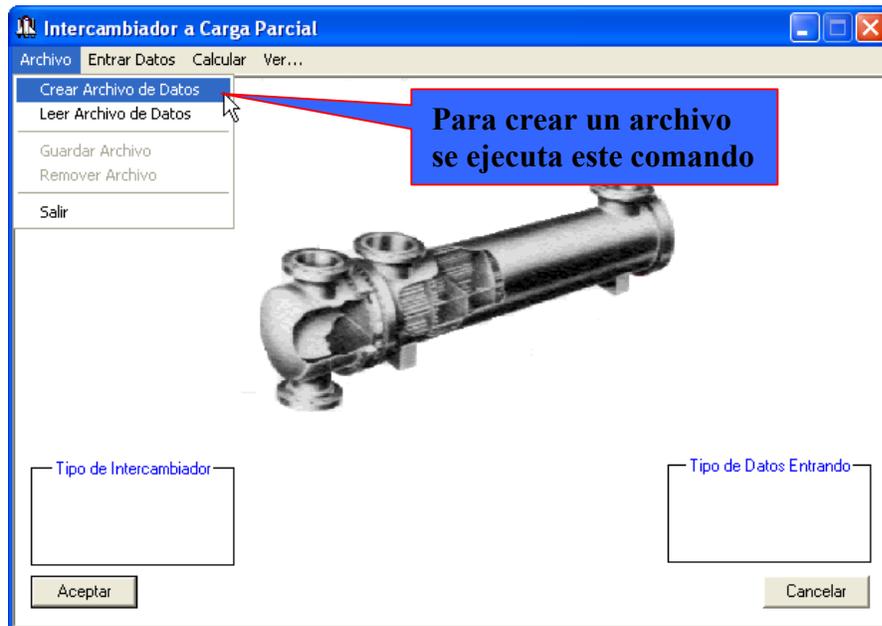
Fig. 38 Exponentes para el análisis del intercambiador.



2.2.3 Funcionamiento. Pasos para llevar acabo los cálculos a carga parcial con el programa Intercambiador_Carga_Parcial.exe.

1. Cree un archivo de datos: Oprima el comando **Crear Archivo de Datos** en el menú **Archivo** como se muestra en la figura 39. Al oprimir el comando se abre la ventana **Guardar como** de Windows (ver Fig. 40), donde se le da un nombre al archivo de datos y se selecciona el directorio donde se guarda. Después de oprimir el botón **Guardar** aparecerán los controles del programa.

Fig. 39 Creando archivo de datos del intercambiador.



2. Seleccione el tipo de intercambiador a analizar (ver Fig. 41): Esto se realiza operando uno de los controles de la figura 35.

3. Entre los datos de los flujos en condiciones de diseño (ver Fig. 42): Para esto oprima el comando **Datos de Diseño** de la figura 34 y luego oprima cada botón de los

flujos del intercambiador mostrados (ver Fig. 33); después de lo cual se abre la ventana de la figura 36 donde se recogen los datos al oprimir el botón OK.

Fig. 40 Guardar como... del intercambiador.

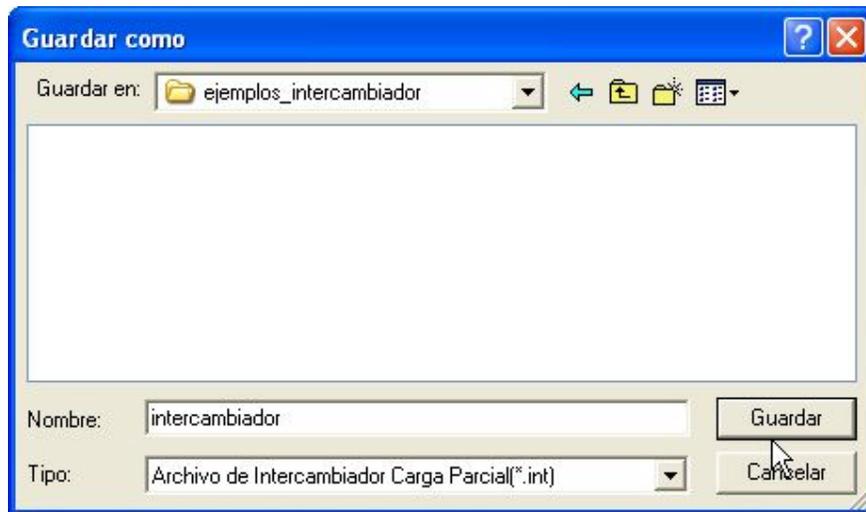


Fig. 41 Seleccionando el tipo de intercambiador.

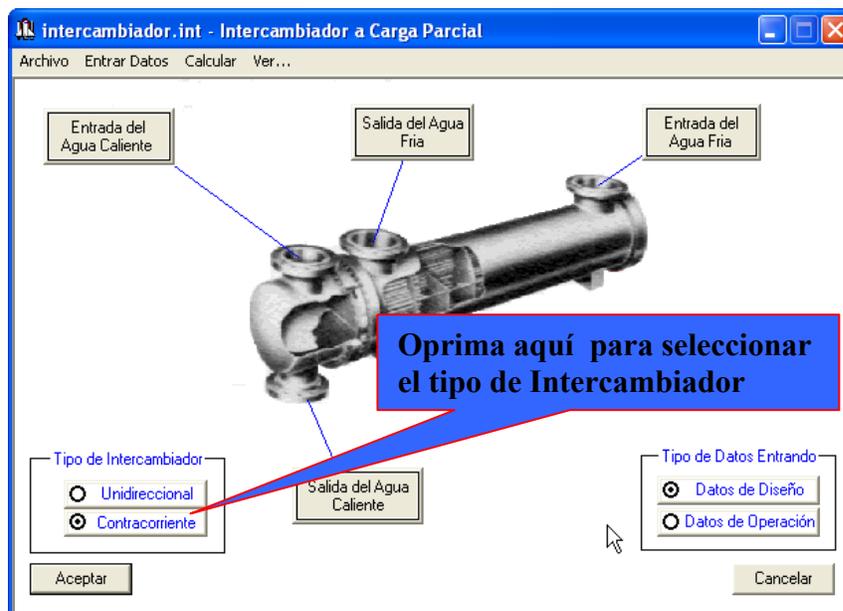
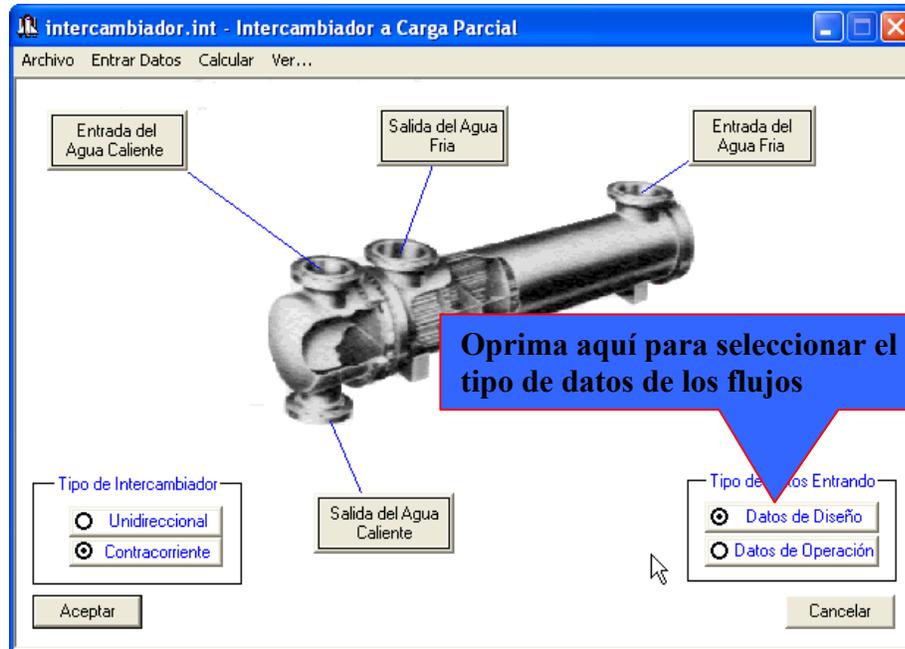


Fig. 42 Entrando datos en condiciones de diseño y de operación del intercambiador.



4. Entre los datos de los flujos en condiciones de operación (ver Fig. 42): Para esto oprima el comando **Datos de Operación** de la figura 34 y luego oprima cada botón de los flujos del intercambiador mostrados (ver Fig. 33); después de lo cual se abre la ventana de la figura 36 donde se recogen los datos al oprimir el botón OK.

5. Entre los datos mecánicos (ver Fig. 43): Oprima el elemento **Datos Mecánicos** del menú **Entrar Datos** de la barra de menú (ver Fig. 31); después de lo cual se abre la ventana de la figura 37. En esta ventana se recogen los datos después de oprimir el botón aceptar.

6. Entre los valores de los exponentes para analizar el intercambiador (ver Fig. 44): Oprima el elemento **Datos de Análisis** del menú **Entrar Datos** de la barra de menú (ver Fig. 31); después de lo cual se abre la ventana de la figura 38.

Fig. 43 Entrando datos mecánicos del intercambiador.

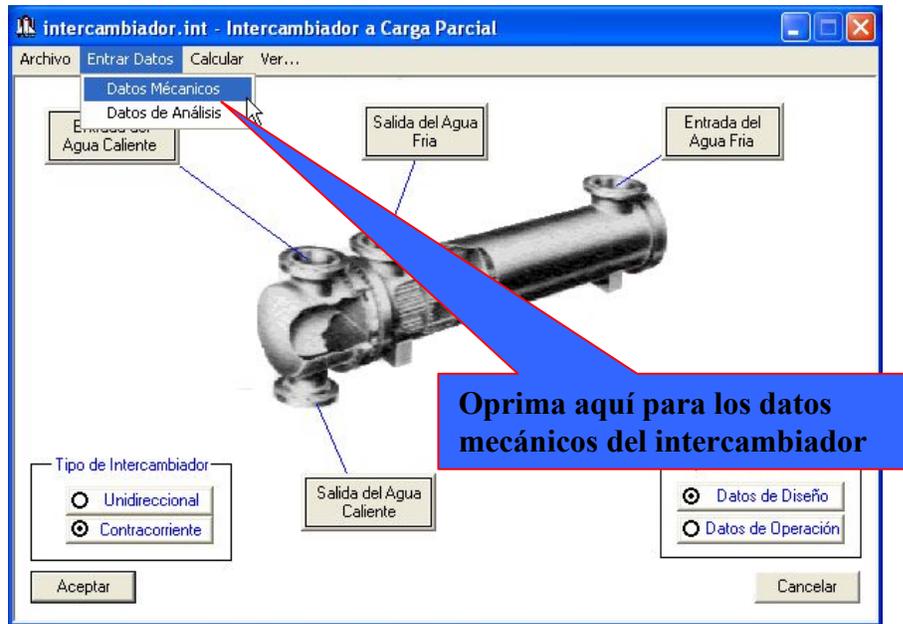
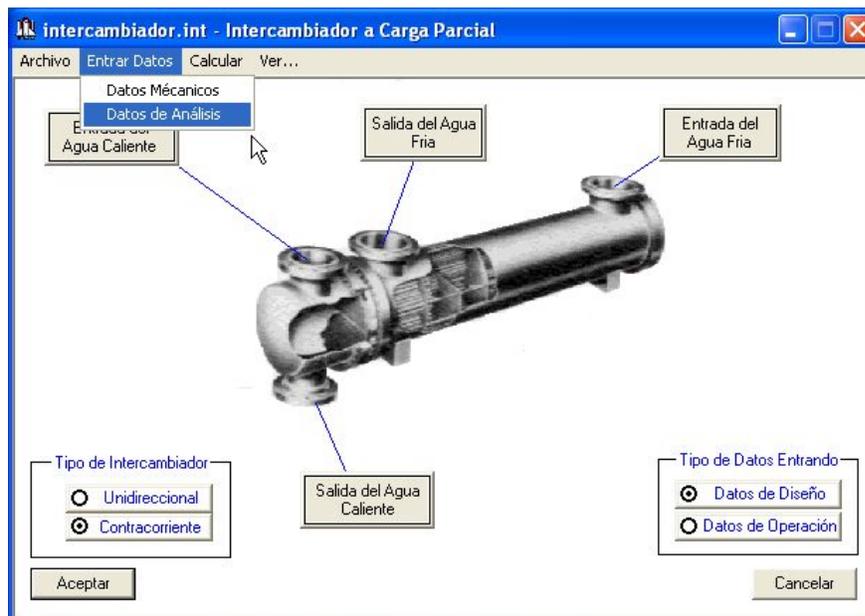


Fig. 44 Entrando datos de análisis del intercambiador.



7. Realice los cálculos a carga parcial: Oprima el comando **Calcular** del menú (ver Fig. 45). Cuando el programa ya ha terminado los cálculos pregunta con la ventana de la figura 46 si se desea ver los resultados o no.

Fig. 45. Comando calcular del intercambiador.

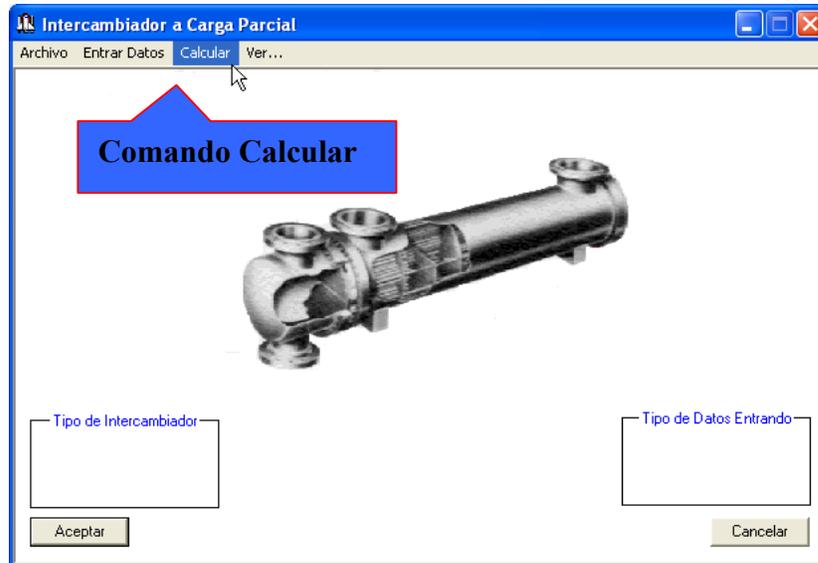
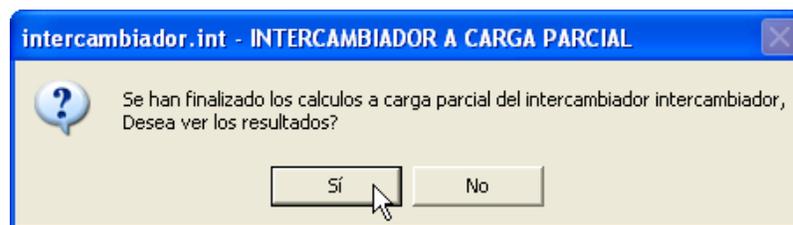


Fig. 46 Fin de los cálculos del intercambiador.



11. Guarde o elimine los datos introducidos: Por último se puede conservar todos los datos introducidos anteriormente en archivos que tienen el mismo nombre del archivo de datos creado al principio accionando el comando **Guardar Archivo** del menú **Archivo** o eliminarlos accionando el comando **Remover Archivo**.

Los resultados de los cálculos a carga parcial del intercambiador son mostrados por medio del programa WordPad de Windows; para verlos se oprime el elemento **Informe de Resultados** del menú **Ver...** (ver Fig. 32 y 47); después de lo cual se abre la ventana de la figura 48.

Fig. 47 Revisando los resultados del intercambiador.

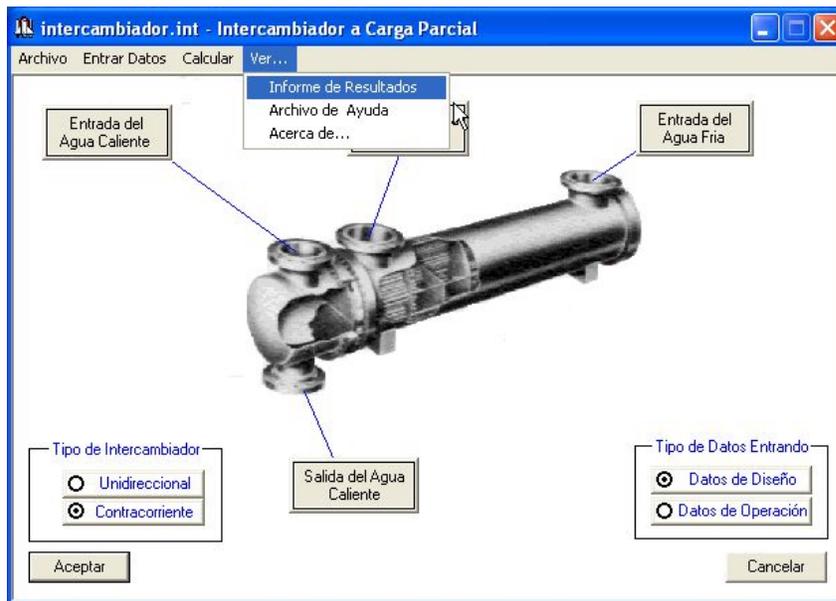
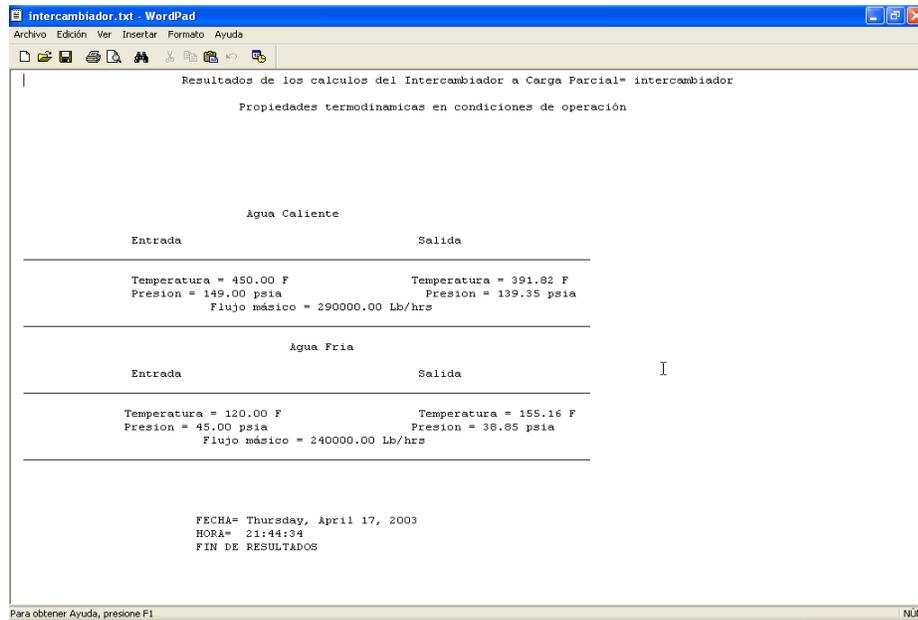


Fig. 48 Archivo de texto con los resultados del intercambiador.



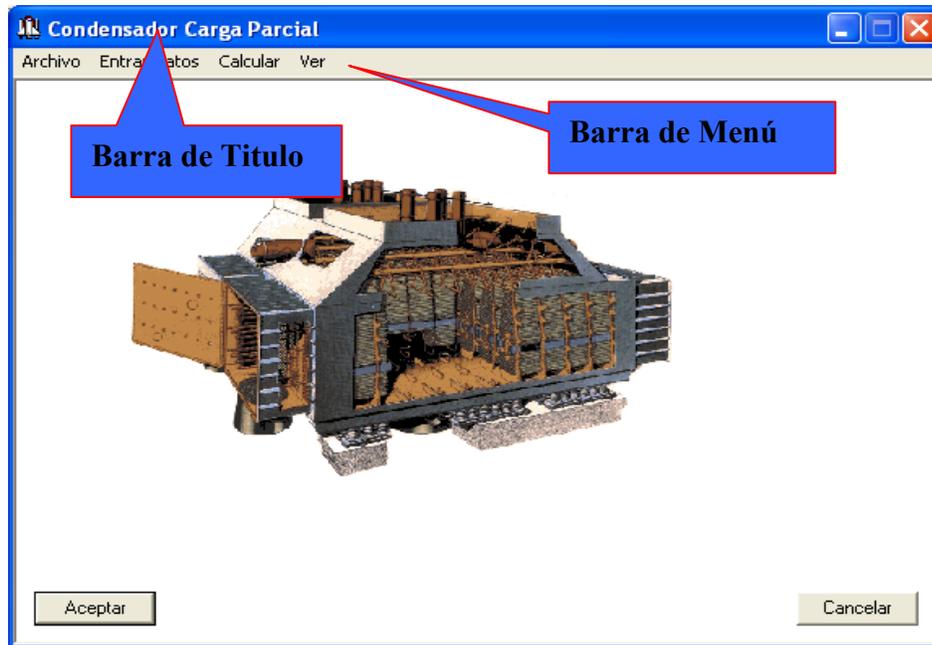
2.3 PROGRAMA CONDENSADOR_CARGA_PARCIAL.EXE.

El programa Condensador_Carga_Parcial.exe es un archivo ejecutable que tiene la capacidad de calcular las caídas de presión y las temperaturas de salida en el lado casco y tubo a condiciones de operación de los condensadores; siendo tanto el fluido caliente como el fluido frío el agua.

2.3.1 Interfaz de Usuario. La figura 49 muestra la ventana principal del programa; en ella se observan la barra de título, el ícono y los comandos minimizar y cerrar comunes de Windows; debajo se encuentra la barra de menú y el área de trabajo.

En la figura 50 se observan los elementos del menú **Archivo** que son: **Crear Archivo de Datos**, **Leer Archivo de Datos**, **Guardar Archivo**, **Remover Archivo** y **Salir**.

Fig.49 Presentación Condensador_Carga_Parcial.exe.



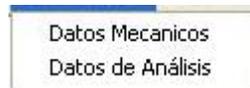
- **Crear Archivo de Datos:** Abre la ventana de Windows *Guardar como* para la creación de un archivo de datos del condensador con extensión “.cdr”.
- **Leer Archivo de Datos:** Abre la ventana de Windows *Abrir* para la búsqueda de un archivo ya creado.
- **Guardar Archivo:** Guarda el archivo de datos en el cual se esta trabajando para poder crear o abrir otro.
- **Remove Archivo:** Borra o elimina todos los archivo de datos generados por el programa que tenga el nombre del archivo que se encuentre cargado.
- **Salir:** Finaliza la ejecución del programa.

En el menú **Entrar Datos** se encuentran dos elementos que son: **Datos Mecánicos** y **Datos de Análisis** (ver Fig. 51).

Fig. 50 Elementos del menú Archivo del condensador.



Fig. 51 Elementos del menú Entrar Datos del condensador.



- **Datos Mecánicos:** Abre una ventana en la cual se introducen algunos datos del diseño mecánico del condensador.
- **Datos de Análisis:** Abre una ventana en la cual se introducen los exponentes x , y , z para el análisis para determinar el comportamiento del equipo.

El comando **Calcular** del menú es el control que se acciona para realizar los cálculos a carga parcial.

El menú **Ver...** (Ver Fig. 53) contiene también tres elementos que son: **Informe de Resultados**, **Archivo de Ayuda** y **Acerca de...**

Fig. 52 Comando Calcular del condensador.

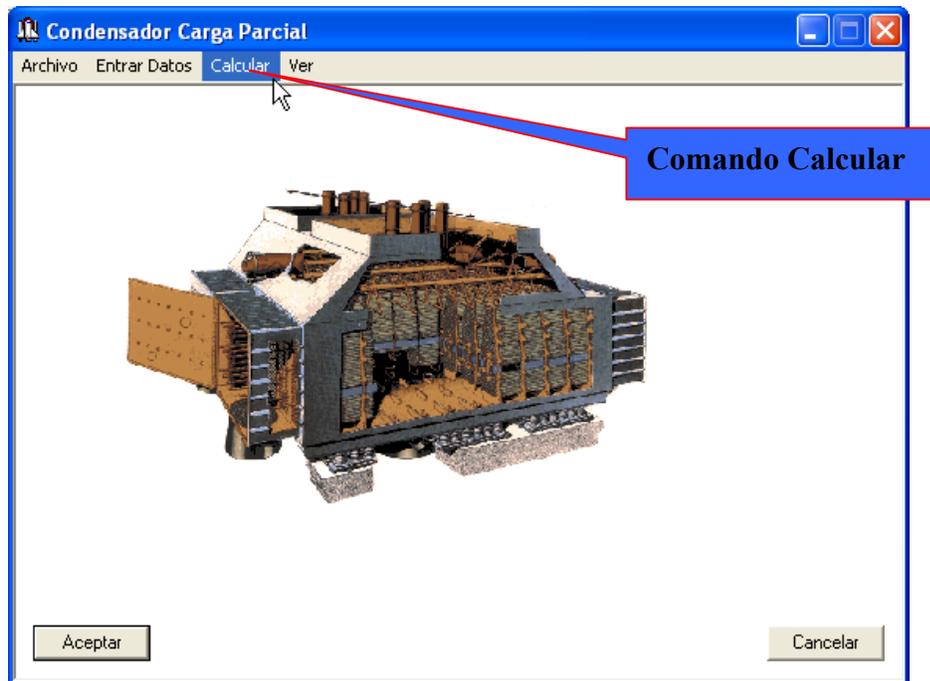
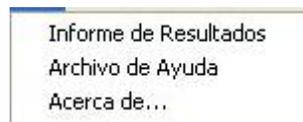
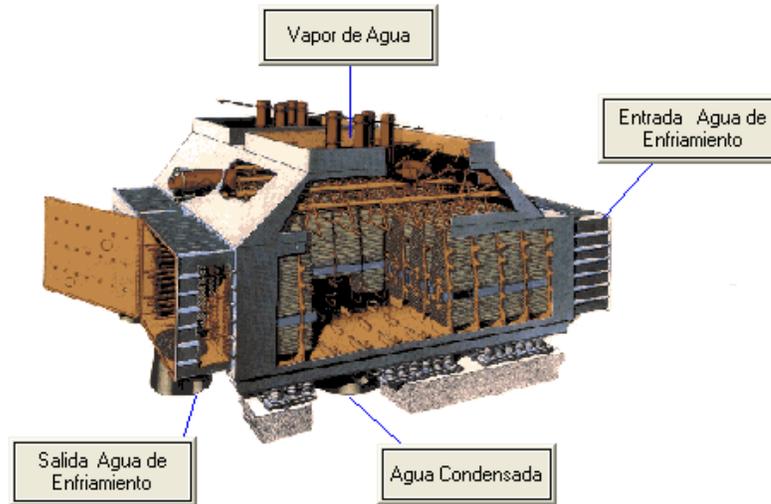


Fig. 53 Elementos del menú Ver... del condensador.



- **Informe de Resultados:** Abre el archivo de texto de los resultados de los cálculos.
- **Archivo de Ayuda:** Abre el documento de ayuda.
- **Acerca de...:** Abre la ventana donde aparece el nombre y la versión del programa, quien lo desarrolló y quien dirigió su realización.

Fig. 54 Controles de los flujos del condensador.



La figura 54 muestra los botones que abren las ventanas para entrar los datos termodinámicos de los diferentes flujos del condensador.

La figura 55 muestra los controles de selección única para escoger el tipo de datos termodinámicos que se introducen en los flujos del condensador.

Fig.55 Tipo de datos de entrada del condensador.



2.3.2 Datos de entrada. Para llevar a cabo los cálculos a carga parcial en un condensador se requieren los siguientes datos: Datos de los flujos del agua fría y del agua caliente, datos mecánicos y datos para determinar el comportamiento del equipo.

Los datos de los flujos del agua fría y del agua caliente, tanto en condiciones de diseño como a carga parcial son la temperatura, la presión y el flujo másico (ver Fig. 56).

Fig. 56 Datos termodinámicos de los flujos del condensador.

Datos de Diseño del Vapor de Agua

Temperatura: 455 F

Presión: 60 psia

Flujo Másico: 300000 Lb/hrs

Sistema de Unidades:

- Internacional
- Inglesas
- Métricas

OK Cancel

Fig. 57 Datos mecánicos del condensador.

Datos Mecánicos

Area Total Transferencia de Calor: 3200 ft²

Ud (Coeficiente de Transferencia de Calor de diseño): 10.75 Btu/hr*ft²*F

Sistema de Unidades:

- Internacional
- Inglesas
- Métricas

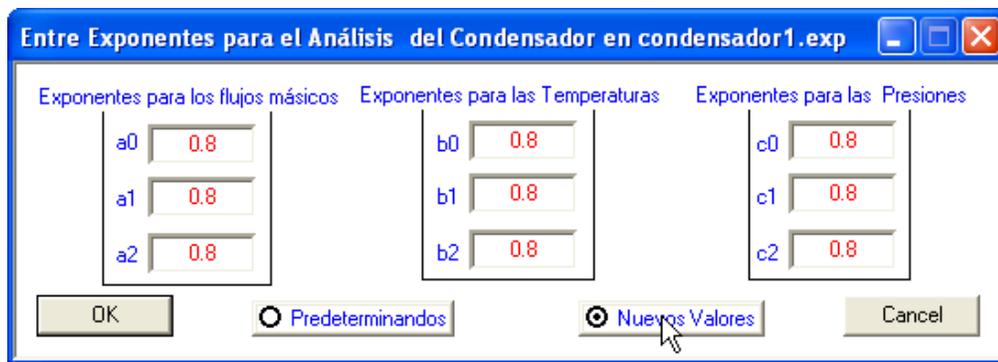
Entrar Ud Calcular Ud

OK Cancel

El único dato mecánico requerido es el área total de transferencia de calor, pero en el caso de que se conozca el coeficiente de transferencia de calor global en condiciones de diseño es posible introducirlo al programa como dato mecánico conocido (ver Fig. 57).

Los datos para determinar el comportamiento del equipo (ver Fig. 58) son los exponentes x , y y z que se usan para calcular la caída de presión en el lado casco, la caída de presión en el lado tubo y el coeficiente de transferencia de calor global en condiciones de operación (ver ecuaciones 5 y 8). Los exponentes x toman los valores de a_0 , a_1 y a_2 . Los exponentes y toman los valores de b_0 , b_1 y b_2 . Los exponentes z toman los valores de c_0 , c_1 y c_2 .

Fig. 58 Datos de análisis del condensador.



2.3.3 Funcionamiento. Pasos para llevar a cabo los cálculos a carga parcial con el programa Condensador_Carga_Parcial.exe.

1. Cree un archivo de datos: Oprima el comando **Crear Archivo de Datos** en el menú **Archivo** como se muestra en la figura 59. Al oprimir el comando se abre la ventana **Guardar como** de Windows (ver Fig. 60), donde se le da un nombre al archivo de datos y se selecciona el directorio donde se guarda. Después de oprimir el botón **Guardar** aparecerán los controles del programa.

2. Entre los datos de los flujos en condiciones de diseño (ver Fig. 61): Para esto oprima el comando **Datos de Diseño** de la figura 55 y luego oprima cada botón de los flujos del condensador mostrados (ver Fig. 54); después de lo cual se abre la ventana de la figura 56 donde se recogen los datos al oprimir el botón **OK**.

Fig. 59 Creando archivo de datos del condensador.

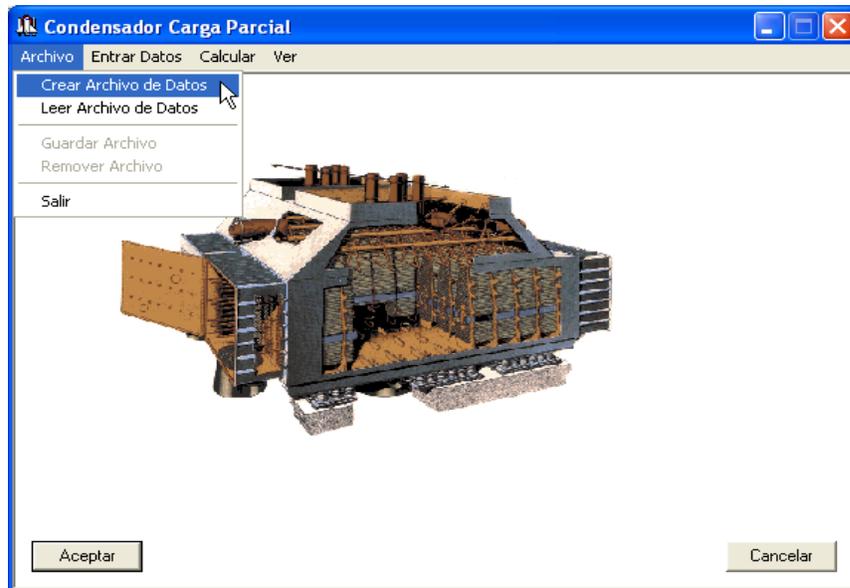
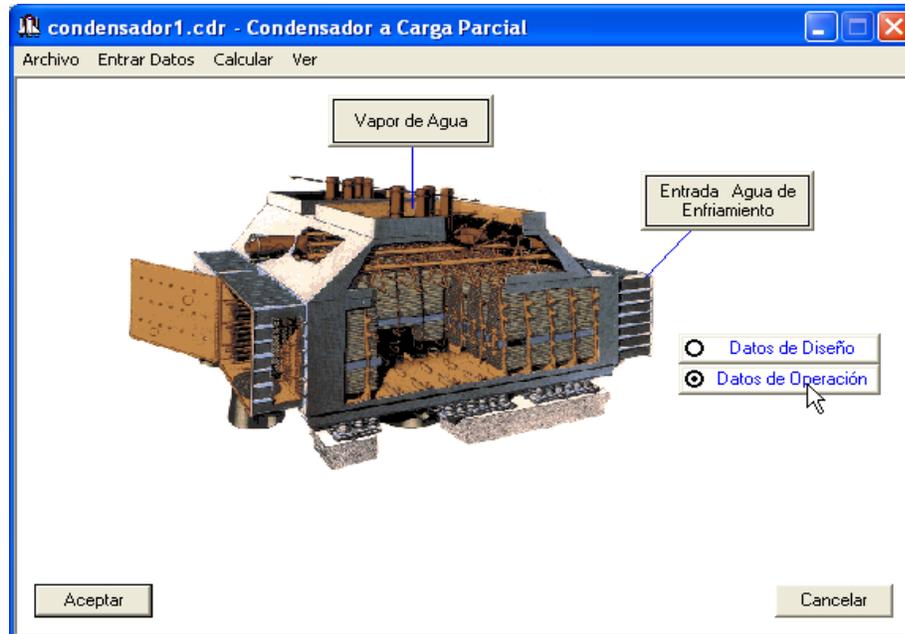


Fig. 60 Guardar como... del condensador.



3. Entre los datos de los flujos en condiciones de operación (ver Fig. 61): Para esto oprima el comando **Datos de Operación** de la figura 55 y luego oprima cada botón de los flujos del intercambiador mostrados (ver Fig. 54); después de lo cual se abre la ventana de la figura 56 donde se recogen los datos al oprimir el botón OK.

Fig. 61 Seleccionando datos de diseño y de operación del condensador.



4. Entre los datos mecánicos (ver Fig. 62): Oprima el elemento **Datos Mecánicos** del menú **Entrar Datos** de la barra de menú (ver Fig. 51); después de lo cual se abre la ventana de la figura 57. En esta ventana se recogen los datos después de oprimir el botón OK.

5. Entre los valores de los exponentes para analizar el intercambiador (ver Fig. 63): Oprima el elemento **Datos de Análisis** del menú **Entrar Datos** de la barra de menú (ver Fig. 51); después de lo cual se abre la ventana de la figura 58. En esta ventana se recogen los datos después de oprimir el botón OK.

Fig. 62 Entrando datos mecánicos del condensador.

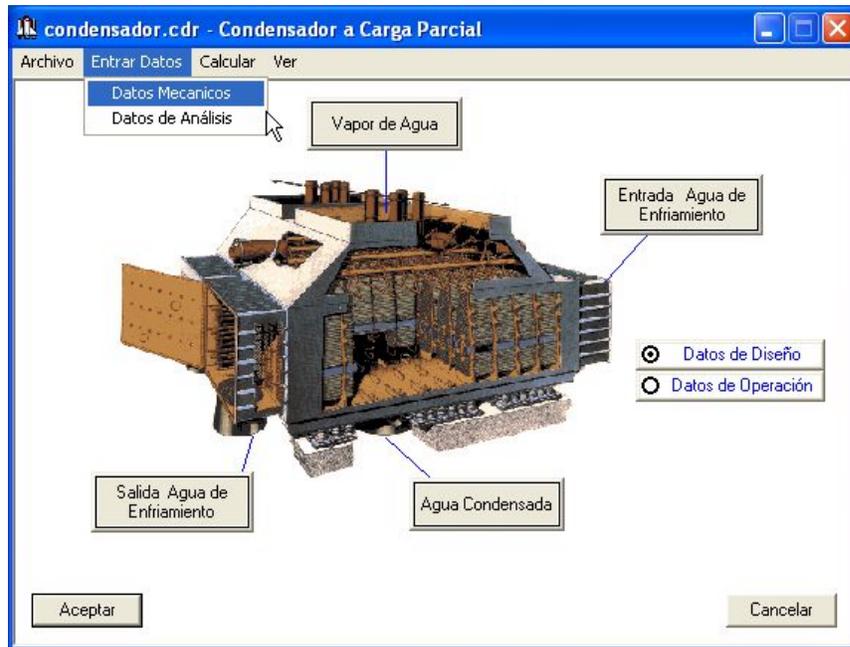
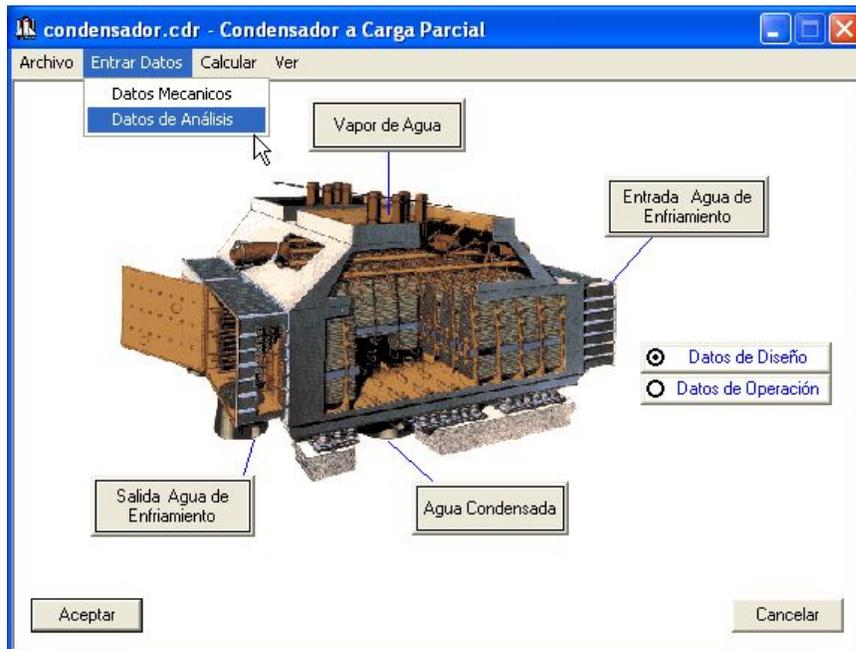


Fig. 63 Entrando datos de análisis del condensador.



6. Realice los cálculos a carga parcial: Oprima el comando **Calcular** del menú (ver Fig. 64). Cuando el programa ya ha terminado los cálculos pregunta con la ventana de la figura 65 si se desea ver los resultados o no.

Fig. 64 Oprimiendo el comando Calcular del condensador.

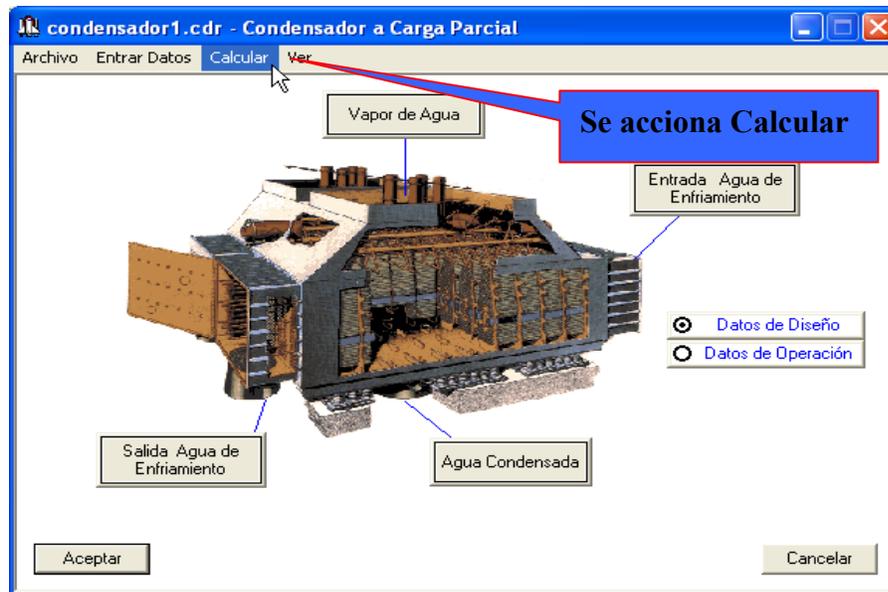
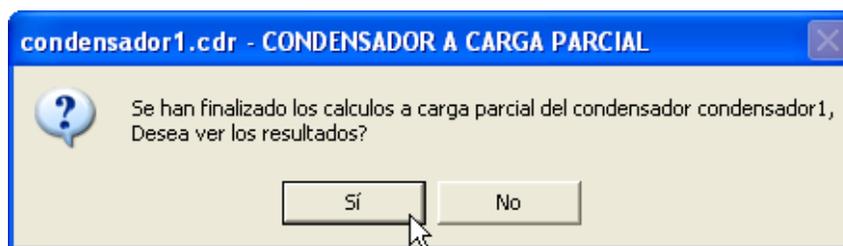


Fig. 65 Fin de los cálculos del condensador.



12. Guarde o elimine los datos introducidos: Por último se puede conservar todos los datos introducidos anteriormente en archivos que tienen el mismo nombre del archivo de datos creado al principio accionando el comando **Guardar Archivo** del menú **Archivo** o eliminarlos accionando el comando **Remove Archivo**.

Los resultados de los cálculos a carga parcial del condensador son mostrados por medio del programa WordPad de Windows; para verlos se oprime el elemento **Informe de Resultados** del menú **Ver...** (ver Fig. 53 y 66); después de lo cual se abre la ventana de la figura 67.

Fig. 66 Revisando los resultados del condensador.

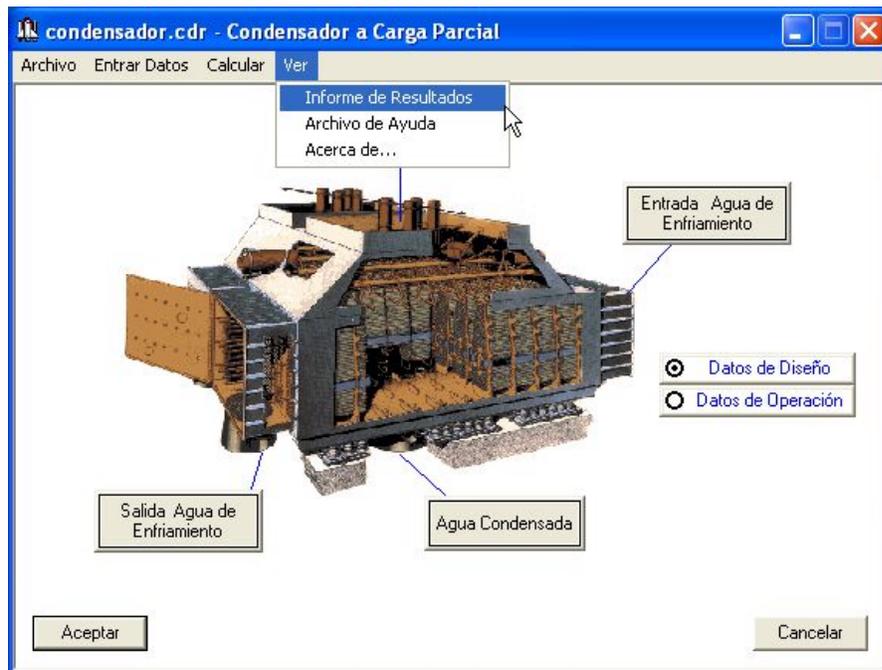
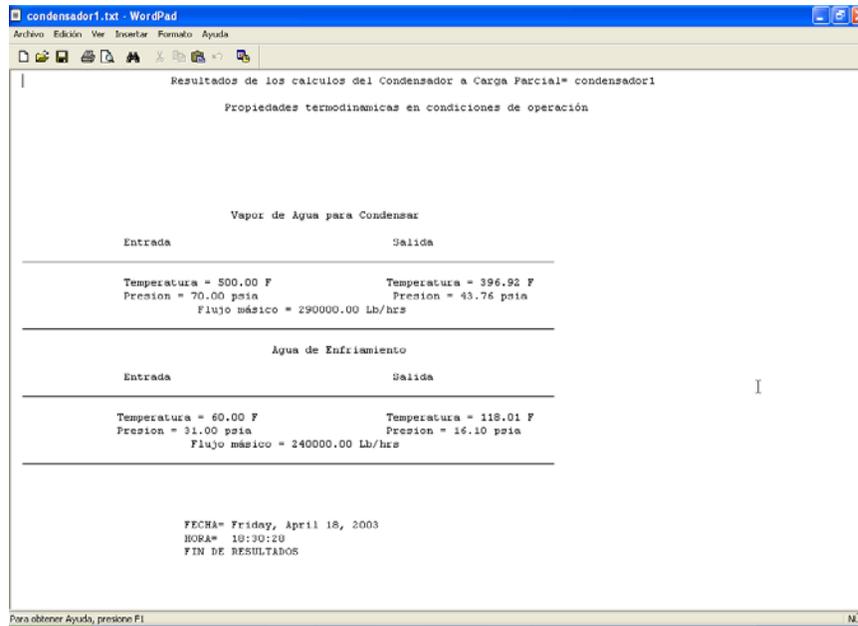


Fig. 67 Archivo de texto con los resultados del condensador.



2.4 PROGRAMA CALDERA_CARGA_PARCIAL.EXE

El programa Caldera_Carga_Parcial.exe es un archivo ejecutable que tiene la capacidad de calcular las caídas de presión y las temperaturas de salida en el lado casco y tubo a condiciones de operación de los equipos independientes que constituyen una caldera, los cuales son: el hogar, el sobrecalentador, el recalentador, el economizador y el calentador de aire.

2.4.1 Interfaz de usuario. La figura 68 muestra la ventana principal del programa; en ella se observan la barra de título, el ícono y los comandos minimizar y cerrar comunes de Windows; debajo se encuentra la barra de menú y la zona de trabajo.

En la figura 69 pueden observarse los elementos del menú **Archivo**, que son: **Crear Archivo de Datos**, **Leer Archivo de Datos**, **Guardar Configuración Actual**, **Remover Archivo de Datos** y **Salir**.

- **Crear Archivo de Datos:** Abre la ventana de Windows *Guardar como* para crear un archivo de datos de caldera con extensión “.cal”.
- **Leer Archivo de Datos:** Abre la ventana de Windows *Abrir* para la búsqueda de un archivo ya creado.

Fig.68 Presentación Caldera_Carga_Parcial.exe.

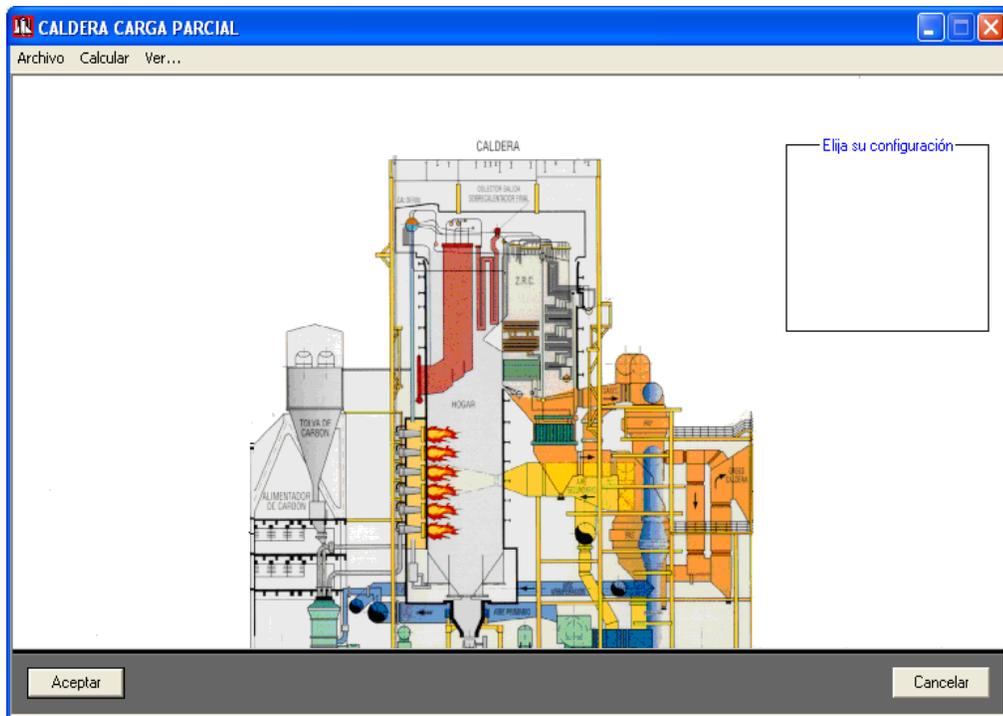


Fig. 69 Elementos del menú Archivo de la caldera.



- **Guardar Configuración Actual:** Guarda el archivo con la configuración de la caldera para poder crear o abrir otro.
- **Remove Archivo de Datos:** Elimina todos los archivo de datos generados por el programa que tengan el nombre del archivo que se encuentra cargado.
- **Salir:** Finaliza la ejecución del programa.

El comando **Calcular** del menú es el comando que se oprime para realizar todos los cálculos (ver figura 70).

En la figura 71 se observan los elementos del menú **Ver...**, que son: **Informe de Resultados**, **Archivo de Ayuda** y **Acerca de...**

- **Ver Informe de Resultados:** Abre el archivo de texto de los resultados de los cálculos.
- **Archivo de Ayuda:** Abre el documento de ayuda.
- **Acerca de...:** Abre la ventana donde aparece el nombre y la versión del programa, quien lo desarrolló y quien dirigió su realización.

En la figura 72 se muestran los botones de cada uno de los componentes que una caldera puede contener.

En la figura 73 se muestran los controles para seleccionar los componentes con que cuenta la caldera que se esta analizando.

En la figura 74 se muestra el botón que abre la ventana donde se introduce la composición de los gases de combustión en porcentaje.

Fig. 70 Comando Calcular de la caldera.

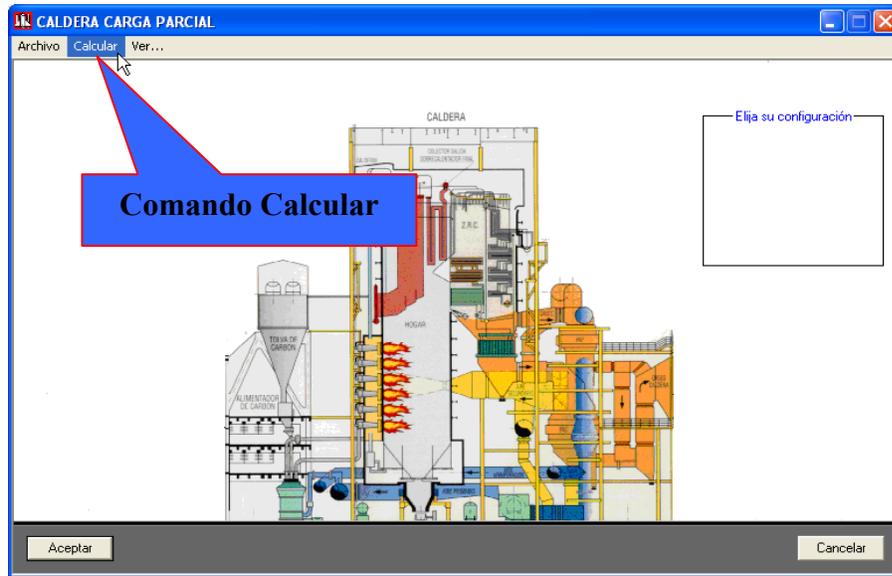


Fig. 71 Elementos del menú Ver... de la caldera.

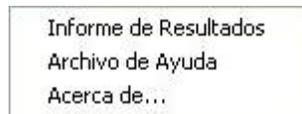


Fig. 72 Controles de los componentes de la caldera.

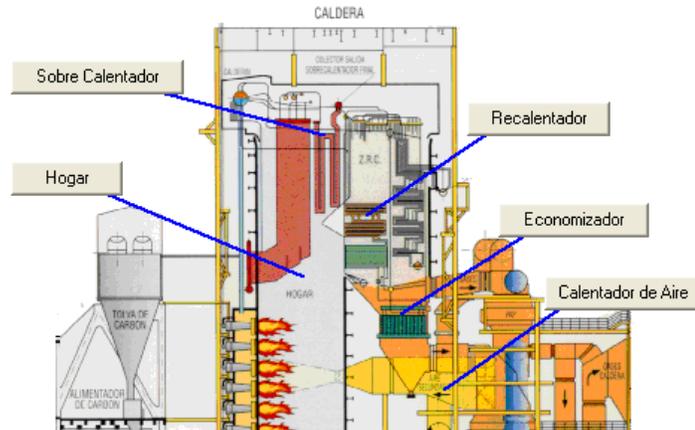


Fig.73 Controles para seleccionar los componentes de la caldera.

Elija su configuración

<input checked="" type="checkbox"/>	Con Hogar
<input type="checkbox"/>	Con SobreCalentador
<input type="checkbox"/>	Con Recalentador
<input checked="" type="checkbox"/>	Con Economizador
<input type="checkbox"/>	Con Calentador

Fig. 74 Control para entrar la composición de los gases de combustión.

Composicion de los Gases

La figura 75 muestra la ventana del componente hogar, donde se encuentran los diferentes controles para entrar propiedades termodinámicas de diseño o de operación de los flujos, elección del sistema de unidades y una barra de menú con tan solo un comando llamado **Entre Otros Datos del Equipo Oprimiendo Aquí**.

La Fig. 76 muestra la lista de elementos del comando del menú de la ventana de componente **Entre Otros Datos del Equipo Oprimiendo Aquí**, los cuales son: **Datos mecánicos y Datos de análisis.**

- **Datos Mecánicos:** Abre una ventana en la cual se introducen algunos datos del diseño mecánico del componente.
- **Datos de Análisis:** Abre una ventana en la cual se introducen los exponentes x, y, z de análisis para determinar el comportamiento del equipo.

Fig. 75 Ventana del componente hogar de la caldera.

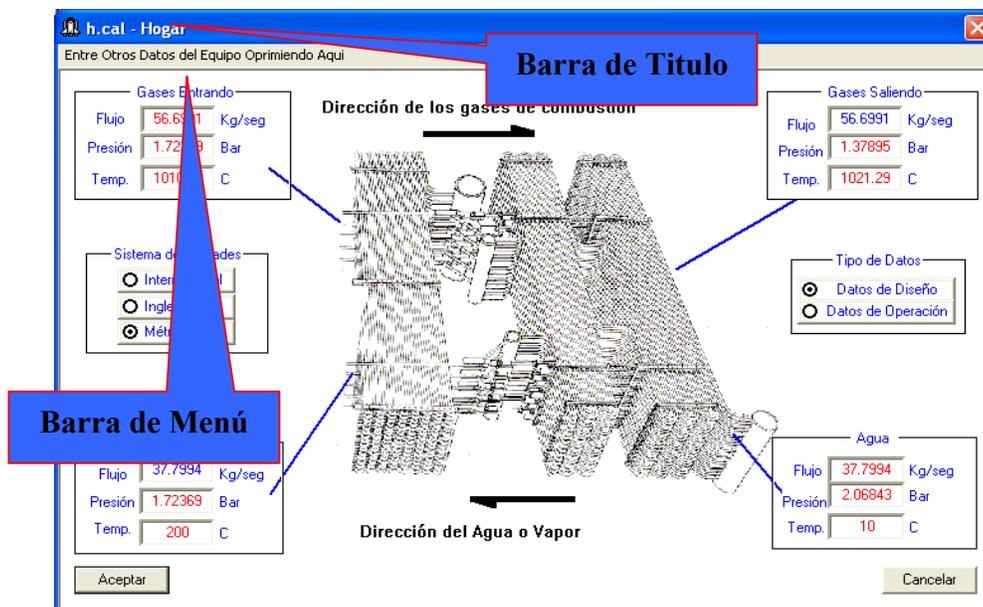
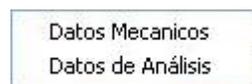


Fig. 76 Elementos del menú de cada componente de la caldera.



2.4.2 Datos de entrada. Para llevar a cabo los cálculos a carga parcial en una caldera se requieren los siguientes datos: Los datos para cada componente, que son los datos del flujo frío y del flujo caliente tanto en condiciones de diseño como de operación, datos mecánicos y datos para determinar su comportamiento y los datos de los gases de combustión para la caldera en general.

Los datos de los flujos frío y caliente, tanto en condiciones de diseño como a carga parcial por cada componente son la temperatura, la presión y el flujo másico (ver Fig. 75).

El único dato mecánico requerido es el área total de transferencia de calor, pero en el caso de que se conozca el coeficiente de transferencia de calor global en condiciones de diseño es posible introducirlo al programa como dato mecánico conocido (ver Fig. 77).

Los datos para determinar el comportamiento del componente (ver Fig. 78) son los exponentes x , y y z que se usan para calcular la caída de presión en el lado casco, la caída de presión en el lado tubo y el coeficiente de transferencia de calor global en condiciones de operación (ver ecuaciones 5 y 8). Los exponentes x toman los valores de a_0 , a_1 y a_2 . Los exponentes y toman los valores de b_0 , b_1 y b_2 ; y los exponentes z toman los valores de c_0 , c_1 y c_2 .

Fig. 77 Datos mecánicos por cada componente de la caldera.

Datos Mecánicos

Area Total de Transferencia de Calor: 3200 ft²

Ud (Coeficiente de transferencia de calor de diseño): 10.75 Btu/hr*ft²*F

Sistema de Unidades:

- Internacional
- Inglesas
- Métricas

Entrar Ud Calcular Ud

OK Cancel

Fig. 78 Exponentes para el análisis de cada componente de la caldera.

Datos de Análisis

Exponentes para los flujos másicos Exponentes para las Temperaturas Exponentes para las Presiones

a0	0.8	b0	0.8	c0	0.8
a1	0.8	b1	0.8	c1	0.8
a2	0.8	b2	0.8	c2	0.8

OK Predefinidos Entrar Nuevos Cancel

Los datos de los gases de combustión son las proporciones en porcentaje de cada gas que se encuentra en ellos (ver Fig. 79).

Fig. 79 Datos de los gases de combustión de la caldera.

Gas	Porcentaje (%)
CO2	80
SO2	0
N2	0
CO	0
SOx	0
H2	0
H2O	20
O2	0
NOx	0

Humedad del Aire: 0 Absoluta

2.4.3 Funcionamiento. Pasos para llevar a cabo los cálculos a carga parcial con el programa Caldera_Carga_Parcial.exe.

1. Cree un archivo de datos: Oprima el comando **Crear Archivo de Datos** en el menú **Archivo** como se muestra en la figura 80. Al oprimir el comando se abre la ventana **Guardar como** de Windows (ver Fig. 81), donde se le da un nombre al archivo de datos y se selecciona el directorio donde se guarda. Después de oprimir el botón **Guardar** aparecerán los controles del programa.

2. Seleccione o elija los componentes de la caldera para ser analizados (ver Fig. 82): Para esto oprima cualquier control de la figura 73; después de lo cual aparecerán los respectivos botones (ver Fig. 72) de los componentes en la ventana principal del programa.

3. Entre los datos de cada componente: Para esto oprima sobre los botones que aparecen en la ventana principal luego de elegir la configuración de la caldera (paso 2); al hacer esto aparecerá la ventana de cada componente (ver Fig. 75), en esta ventana entre los datos termodinámicos de cada flujo, tanto en condiciones de diseño como en operación, los datos mecánicos y de análisis requeridos. Luego de oprimir el botón aceptar se recogerán los datos.

Fig. 80 Creando archivo de datos de la caldera.

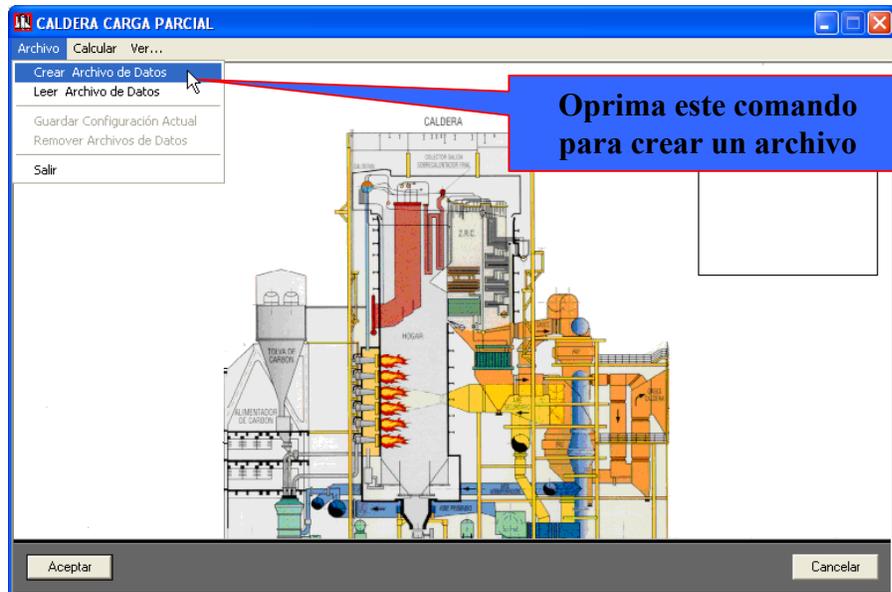


Fig. 81 Guardar como... de la caldera.

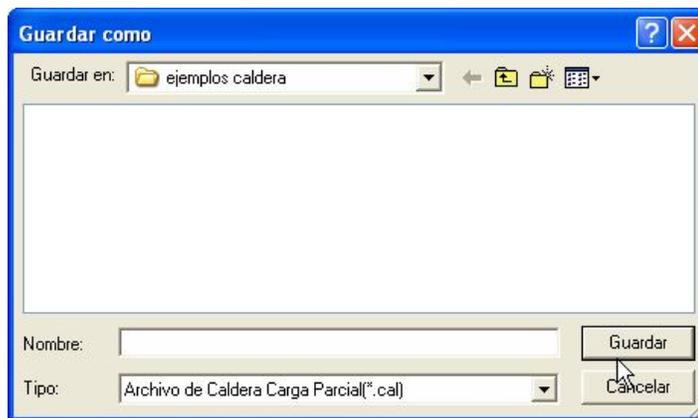
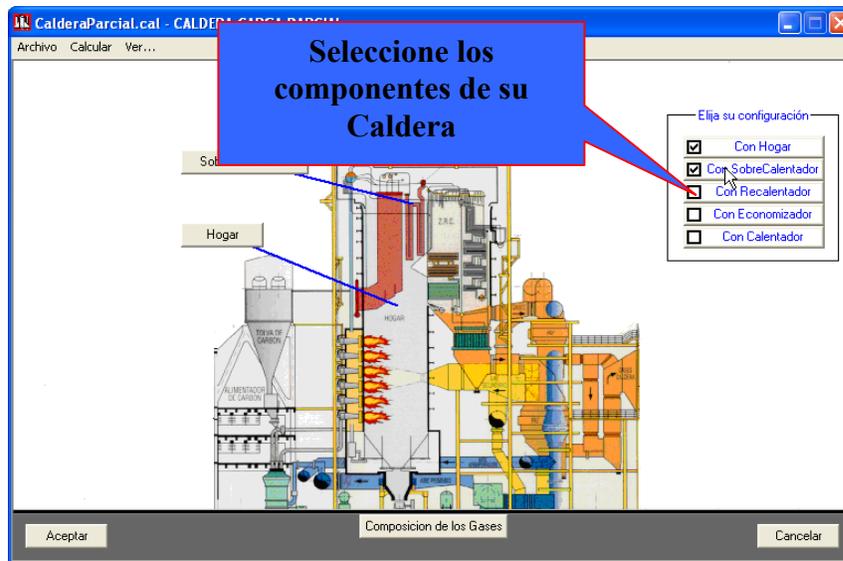


Fig. 82 Seleccionando los componentes de la caldera.



Al oprimir sobre un control de componente como el Hogar o Sobrecalentador se abre la ventana de la unidad seleccionada, como se muestra en la figura 75. Como se ha dicho anteriormente cada componente de la caldera se analiza como un intercambiador de calor en contra flujo, solo que el fluido frío es agua o el vapor de agua y el fluido caliente son los gases de combustión; en el componente Calentador de aire el fluido frío es el aire real y el fluido caliente son los gases de combustión. Por lo tanto en cada componente seleccionado, se debe entrar los datos termodinámicos en condiciones de diseño y de operación, los datos mecánicos y de análisis. Como se puede notar (ver Fig. 75) la ventana tiene por título el nombre del archivo que el usuario ha definido y el nombre del componente.

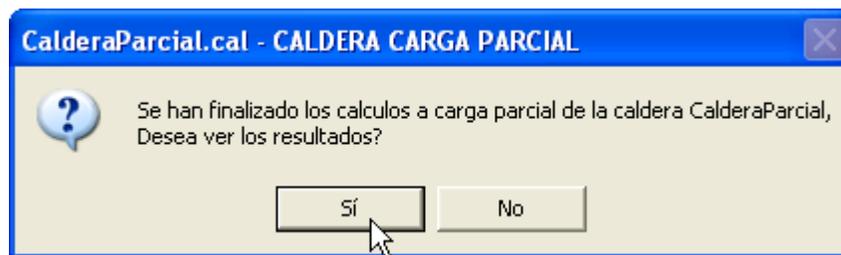
4. Entre los datos de la composición de los gases de combustión: Oprima el botón de la ventana principal llamado **Composición de Gases** (ver Fig. 83); después de lo cual aparece la ventana 78, en donde se recogen los datos al oprimir el botón OK.

Fig. 83 Entrando composición de los gases de combustión de la caldera.



5. Realice los cálculos a carga parcial: Oprima el comando del menú Calcular (ver Fig. 70). Cuando el programa ya ha terminado los cálculos pregunta con la ventana de la figura si se desea ver los resultados o no (ver Fig. 84).

Fig. 84 Fin de los cálculos de la caldera.



6. Guarde o elimine los datos introducidos: Por último se puede conservar todos los datos introducidos anteriormente en archivos que tienen el mismo nombre del archivo de datos creado al principio accionando el comando **Guardar Configuración Actual**

del menú **Archivo** o eliminarlos accionando el comando **Remove Archivo de Datos**.

Los resultados de los cálculos a carga parcial de la caldera son mostrados por medio del programa WordPad de Windows; para verlos se oprime el elemento **Informe de Resultados** del menú **Ver...** (ver Fig. 85 y 86).

Fig. 85 Revisando los resultados de la caldera.

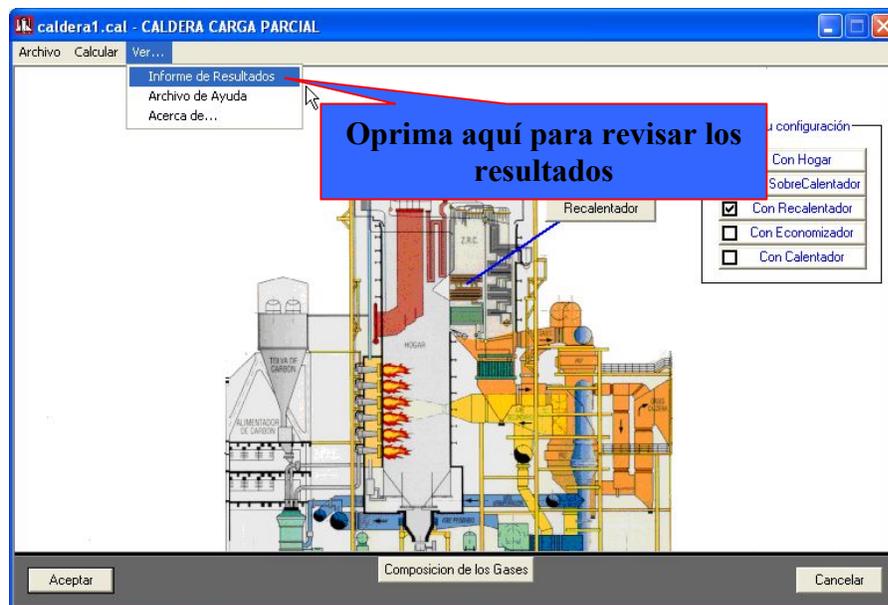
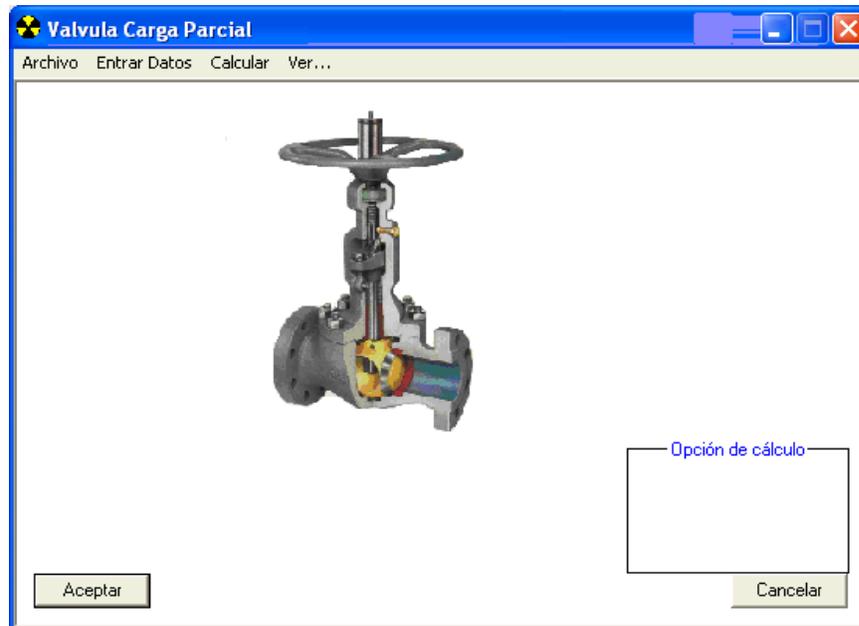
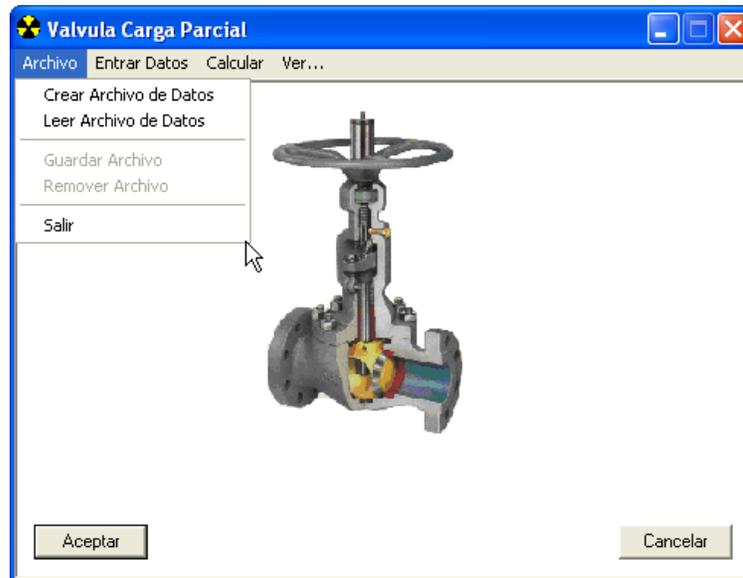


Fig. 87 Presentación Válvula_Carga_Parcial.exe.



- **Crear Archivo de Datos:** Abre la ventana de Windows *Guardar como* para la creación de un archivo de datos de la válvula con extensión “.vlv”.
- **Leer Archivo de Datos:** Abre la ventana de Windows *Abrir* para la búsqueda de un archivo ya creado.
- **Guardar Archivo:** Guarda el archivo de datos en el cual se está trabajando para poder crear o abrir otro.
- **Remover Archivo:** Borra o elimina todos los archivos de datos generados por el programa que tenga el nombre del archivo que se encuentre cargado.
- **Salir:** Finaliza la ejecución del programa.

Fig. 88 Elementos del menú Archivo de la válvula.

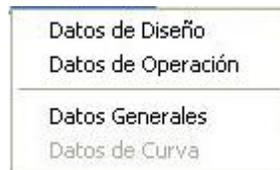


En el menú **Entrar Datos** (ver Fig. 89) se encuentran cuatro elementos que son: **Datos de Diseño**, **Datos de Operación**, **Datos Generales** y **Datos de Curva**.

- **Datos de Diseño:** Abre una ventana en la cual se introducen algunos datos de diseño de la válvula, como el flujo máximo o de diseño y la pérdida de presión generada al pasar tal flujo por ella.
- **Datos de Operación:** Abre una ventana en la cual se introduce el flujo en condiciones de operación.
- **Datos Generales:** Abre una ventana en la cual se introducen algunos datos que involucran características de análisis de la válvula en operación como su apertura, y el tipo de comportamiento de la válvula que se desee analizar.
- **Datos de Curva:** Abre una ventana donde se introducen los valores de la curva que predice el comportamiento de una válvula en condiciones de operación al

restringir el caudal cerrándola un porcentaje de su apertura total, estos valores se introducen cuando no se conoce el flujo de operación y se quiere predecir.

Fig. 89 Elementos del menú Entrar Datos de la válvula.

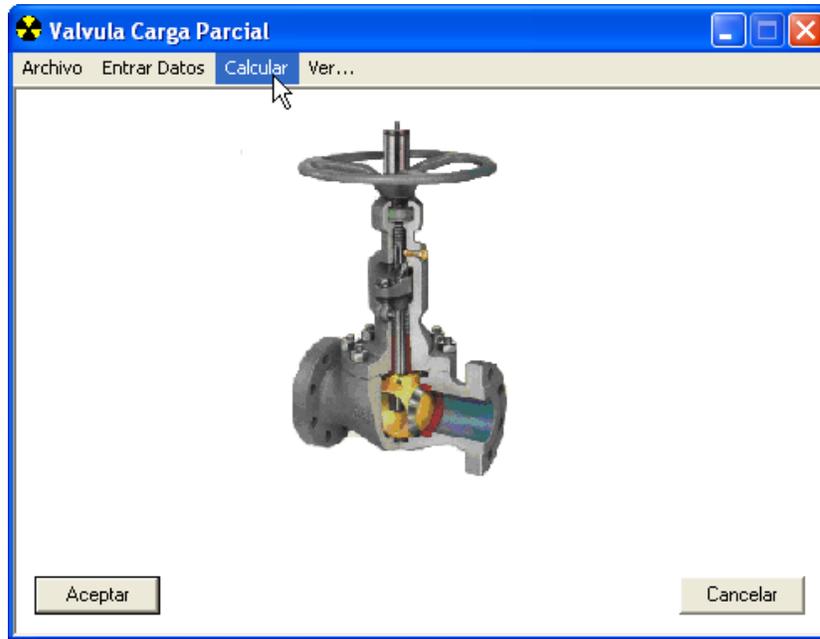


El comando **Calcular** del menú es el control que se acciona para realizar el cálculo del programa (ver figura 90).

El menú **Ver...** (Ver Fig. 91) contiene también tres elementos que son: **Ver Informe de Resultados**, **Archivo de Ayuda** y **Acerca de...**

- **Informe de Resultados:** Abre el archivo de texto generado al terminar los cálculos donde se encuentran los datos en condiciones de diseño y de operación de la válvula analizada a carga parcial.
- **Informe de Resultados:** Abre el archivo de texto generado al terminar los cálculos donde se encuentran los datos en condiciones de diseño y de operación de la válvula analizada a carga parcial.
- **Archivo de Ayuda:** Abre el archivo de ayuda.
- **Acerca de...:** Abre la ventana donde aparece el nombre, la versión del programa, quien lo desarrolló y quien dirigió su realización y el año.

Fig. 90 Comando Calcular de la válvula.



La figura 92 muestra los controles para seleccionar la forma de cálculo del programa Válvula_Carga_Parcial.exe dependiendo de los datos conocidos.

Fig. 91 Elementos del menú Ver... de la válvula.

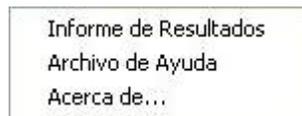
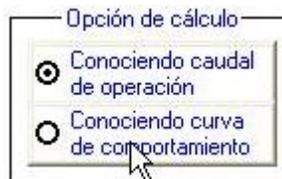


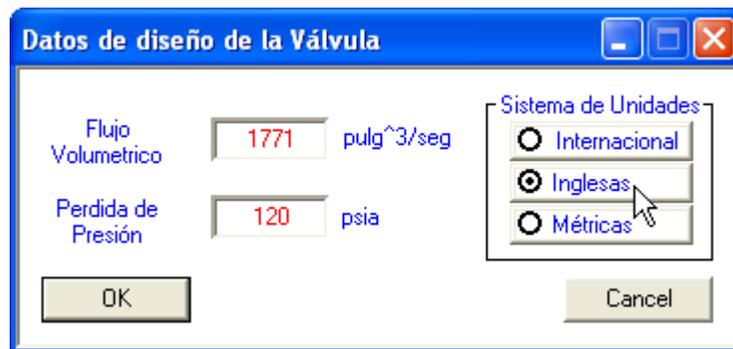
Fig. 92 Opciones de cálculo del programa de la válvula.



2.5.2 Datos de entrada. Para llevar a cabo los cálculos a carga parcial en una válvula se requieren los siguientes datos: Datos de diseño, un dato de operación, datos de las características generales de la válvula y datos de la curva del comportamiento del flujo a través de la válvula.

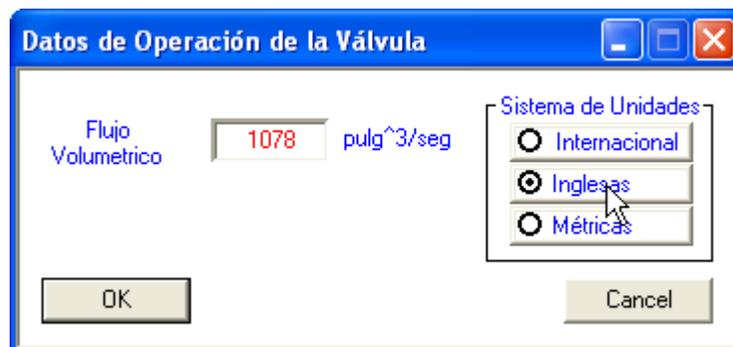
Los datos de diseño son el flujo volumétrico y la pérdida de presión a través de la válvula cuando está totalmente abierta.

Fig. 93 Datos de diseño de la válvula.



El dato de operación requerido es el flujo volumétrico que pasa a través de la válvula en condiciones de operación.

Fig. 94 Dato de operación de la válvula.

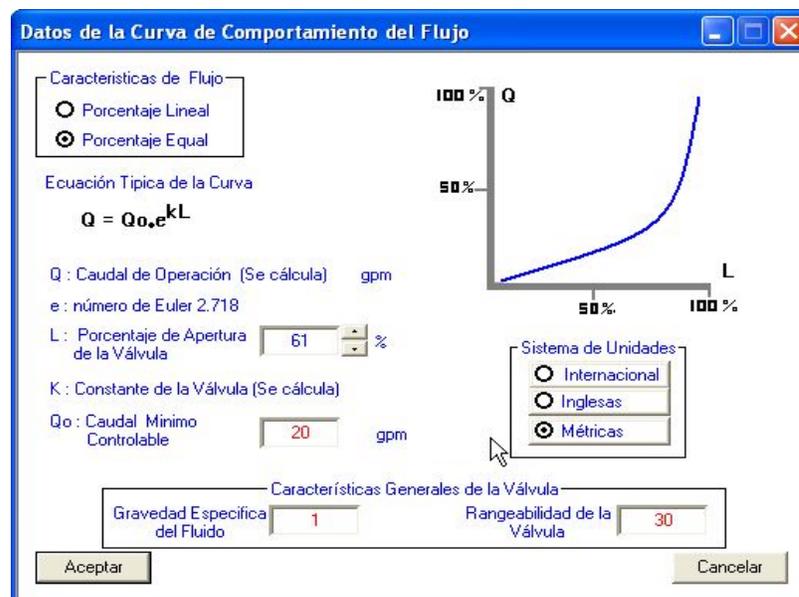


Los datos de las características generales de la válvula (ver Fig. 95) se requieren cuando se ha elegido la opción de cálculo de **Conociendo caudal de operación** (ver Fig. 92); estos son: gravedad específica del fluido, apertura de la válvula, tipo de comportamiento del flujo a través de la válvula y una característica mecánica de diseño llamada la rangeabilidad, definida por fabricante.

Fig. 95 Características generales de la válvula.



Fig. 96 Datos de la curva del comportamiento del flujo a través de la válvula.



Los datos de la curva del comportamiento del flujo a través de la válvula (ver Fig. 96) se requieren cuando se ha elegido la opción de cálculo **Conociendo curva de comportamiento** (ver Fig. 92); estos son: característica del flujo a través de la válvula, apertura de la válvula, caudal mínimo controlable, la gravedad específica del fluido y la rangeabilidad.

2.5.3. Funcionamiento. Pasos para llevar a cabo los cálculos a carga parcial con el programa Valvula_Carga_Parcial.exe.

1. Cree un archivo de datos: Oprima el comando **Crear Archivo de Datos** en el menú **Archivo** como se muestra en la figura 97. Al oprimir el comando se abre la ventana **Guardar como** de Windows (ver Fig. 98), donde se le da un nombre al archivo de datos y se selecciona el directorio donde se guarda. Después de oprimir el botón **Guardar** aparecerán los comandos del programa.

Fig. 97 Creando archivo de datos de la válvula.

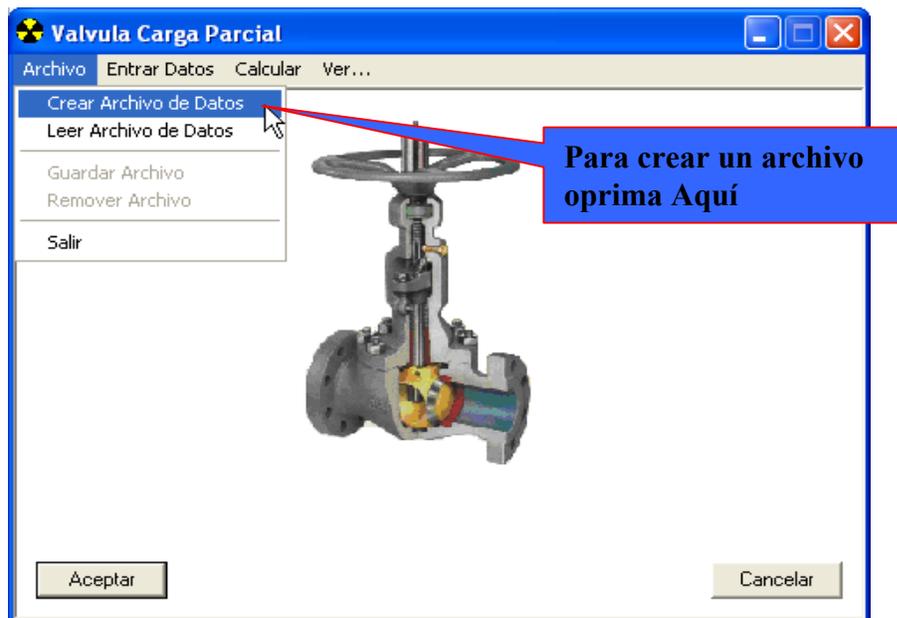
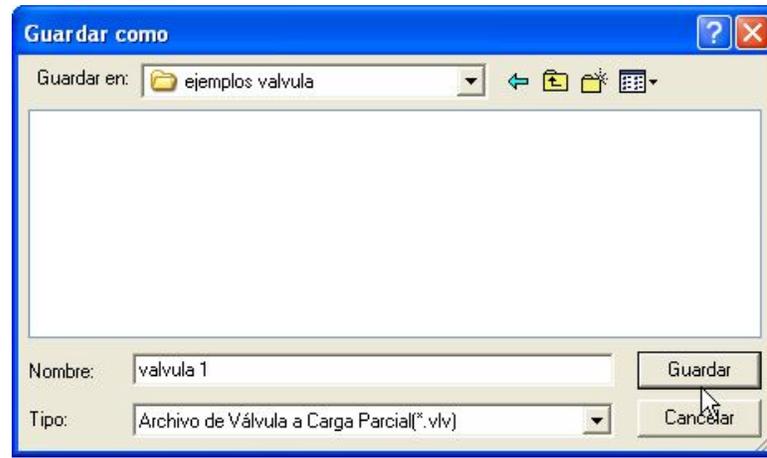
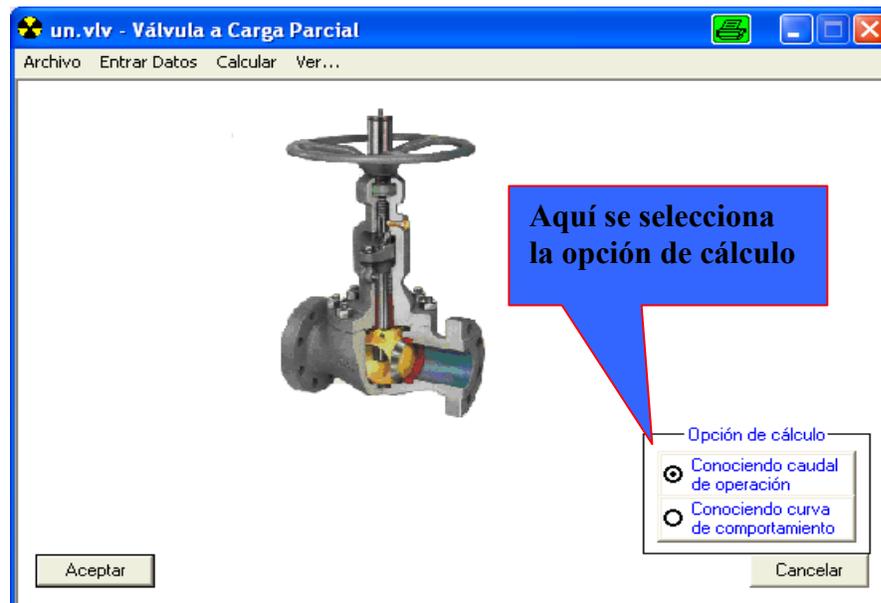


Fig. 98 Guardar como... de la válvula.



2. Seleccione la opción de cálculo para el programa (ver Fig. 99): Oprima sobre uno de los controles de la figura 92.

Fig. 99 Seleccionando la opción de cálculo de la válvula.



3. Entre los datos de diseño de la válvula (ver Fig. 100): Oprima el elemento **Datos de Diseño** del comando **Entrar Datos**; después de lo cual se abre la ventana de la figura 93 donde se recogen los datos después de oprimir el botón OK.

4. Entre el dato de operación de la válvula(ver Fig. 101): Si se ha elegido la opción de cálculo **Conociendo caudal de operación**; oprima el elemento **Datos de Operación** del comando **Entrar Datos**; después de lo cual se abre la ventana de la figura 94 donde se recogen los datos después de oprimir el botón OK.

5. Entre los Datos Generales (ver Fig. 102): Si se ha elegido la opción de cálculo **Conociendo caudal de operación**; oprima el elemento **Datos Generales** del comando **Entrar Datos**; después de lo cual se abre la ventana de la figura 95 donde se recogen los datos después de oprimir el botón OK.

Fig. 100 Entrando los datos de diseño de la válvula.



Fig. 101 Entrando los datos de operación de la válvula.



Fig. 102 Entrando los datos generales de la válvula.



Fig. 103 Entrando los datos de la curva del comportamiento del flujo a través de la válvula.

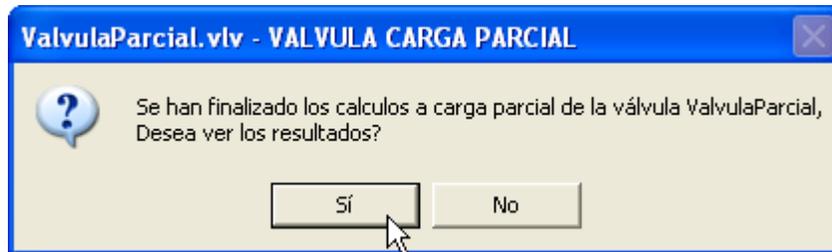


6. Entre los datos de la curva del comportamiento del flujo a través de la válvula (ver Fig. 103): Si se ha elegido la opción de cálculo **Conociendo curva de comportamiento**; oprima el elemento **Datos de Curva** del comando **Entrar Datos**; después de lo cual se abre la ventana de la figura 96 donde se recogen los datos después de oprimir el botón OK.

7. Realice los cálculos a carga parcial: Oprima el comando del menú **Calcular** (ver Fig. 90). Cuando el programa ya ha terminado los cálculos pregunta con la ventana de la figura 104 si se desea ver los resultados o no.

8. Guarde o elimine los datos introducidos: Por último se puede conservar todos los datos introducidos anteriormente en archivos que tienen el mismo nombre del archivo de datos creado al principio accionando el comando **Guardar archivo** del menú **Archivo** o eliminarlos accionando el comando **Remove Archivo**.

Fig. 104 Fin de los cálculos de la válvula.



Los resultados de los cálculos a carga parcial de la válvula son mostrados por medio del programa WordPad de Windows; para verlos se oprime el elemento **Informe de Resultados** del comando **Ver...** del menú (ver Fig. 105 y 106).

Fig. 105 Revisando los resultados de la válvula.

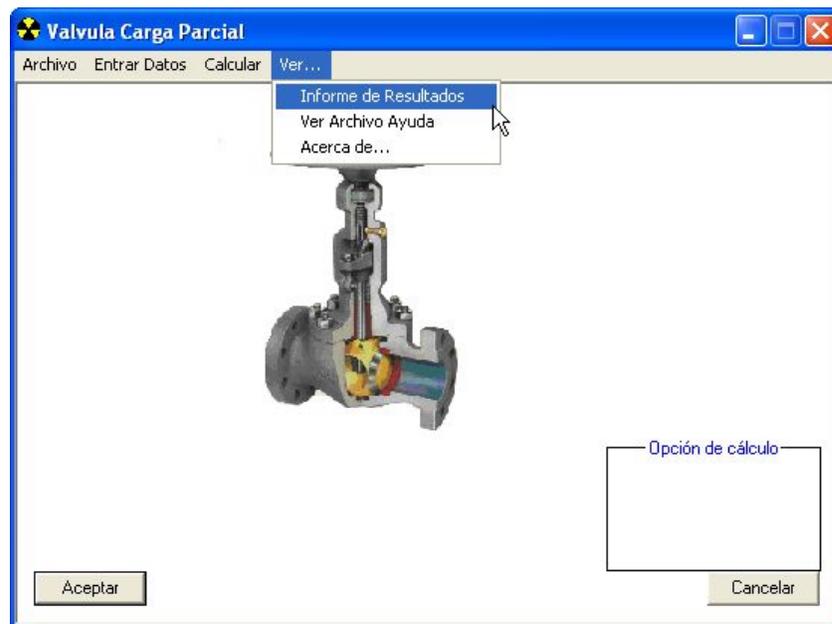
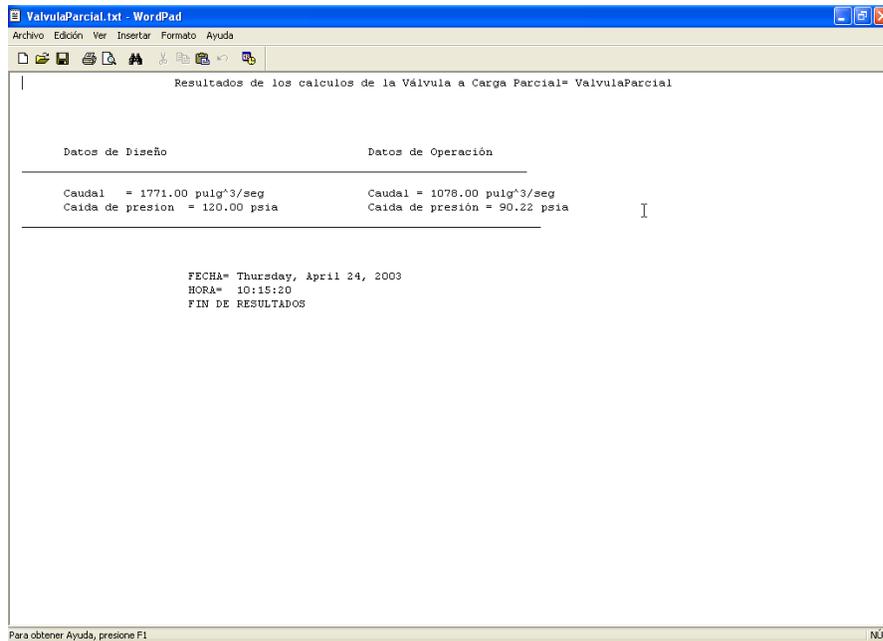


Fig.106 Archivo de texto con los resultados de la válvula.



2.6 PROGRAMA BOMBA_CARGA_PARCIAL.EXE.

El programa Bomba_Carga_Parcial.exe es un archivo ejecutable que tiene la capacidad de calcular el flujo volumétrico, potencia y cabeza de la bomba en condiciones de operación; estos cálculos se realizan de acuerdo al procedimiento expuesto por las leyes de afinidad de las bombas centrifugas.

2.6.1 Interfaz de Usuario. La figura 107 muestra la ventana principal del programa; en ella se observan la barra del titulo de la ventana con el nombre del programa, el icono y los comandos minimizar y cerrar comunes de Windows; debajo se encuentra la barra de menú y el área de trabajo.

En figura 108 se pueden ver los elementos del menú **Archivo**, los cuales son: **Crear Archivo de Datos, Leer Archivo de Datos, Guardar Archivo, Remover Archivo y Salir.**

- **Crear Archivo de Datos:** Abre la ventana de Windows *Guardar como* para la creación de un archivo de datos del condensador con extensión “.bmb”.
- **Leer Archivo de Datos:** Abre la ventana de Windows *Abrir* para la búsqueda de un archivo ya creado.
- **Guardar Archivo:** Guarda el archivo de datos en el cual se esta trabajando para poder crear o abrir otro.

Fig.107 Presentación de la Bomba_Carga_Parcial.exe.

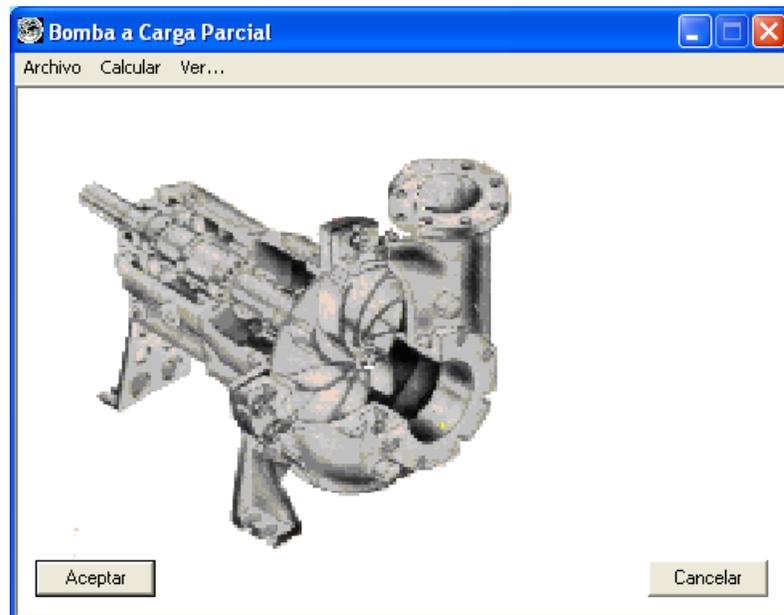


Fig. 108 Elementos del menú Archivo de la bomba.

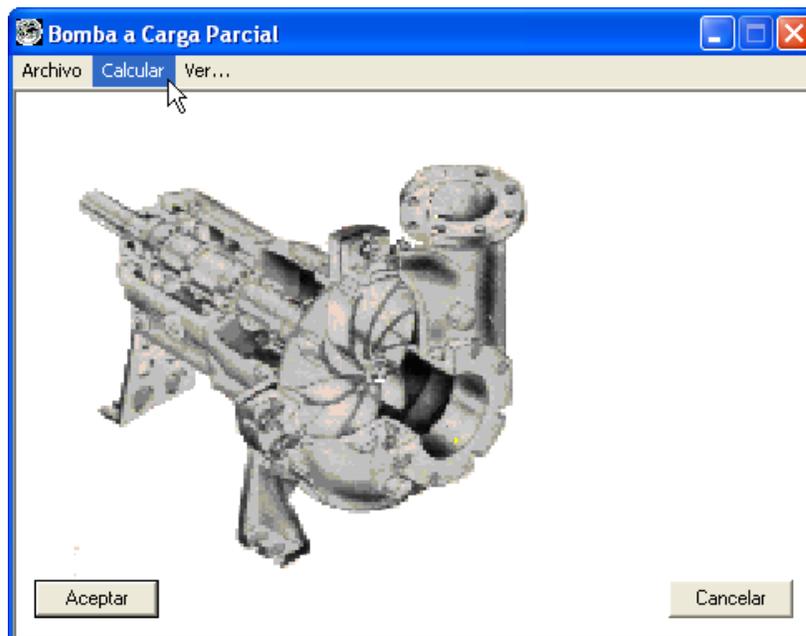


- **Remove Archivo:** Borra o elimina todos los archivos de datos generados por el programa que tenga el nombre del archivo que se encuentre cargado.
- **Salir:** Finaliza la ejecución del programa.

El comando **Calcular** (ver Fig. 109) del menú es el control que se acciona para realizar las operaciones de cálculo de la bomba en condiciones de operación.

El menú **Ver...** (Ver Fig. 110) contiene también tres elementos que son: **Informe de Resultados**, **Archivo de Ayuda** y **Acerca de...**

Fig. 109 Comando Calcular de la bomba.



- **Informe de Resultados:** Abre el archivo de texto generado al terminar los cálculos donde se encuentra los datos de entrada en condiciones de diseño y en condiciones de operación.

- **Archivo de Ayuda:** Abre el archivo de ayuda.
- **Acerca de...:** Abre la ventana donde aparece el nombre y la versión del programa, quien lo desarrolló y quien dirigió su realización.

Fig. 110 Elementos del menú Ver... de la bomba.

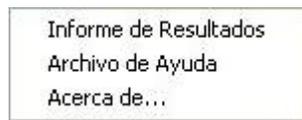
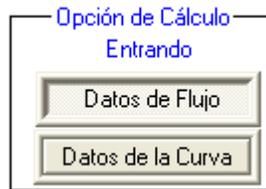
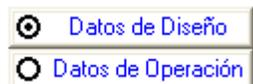


Fig. 111 Opción de cálculo de la bomba.



Los controles de la figura 111 sirven para definir el tipo de cálculo dependiendo si se conoce los datos de los flujos o se conoce los datos de la curva de comportamiento de la bomba.

Fig. 112 Tipo de datos de la bomba.



La figura 112 anterior muestra dos botones de selección única, con los cuales se define si los datos que se introducen son de diseño o de operación.

La figura 113 muestra los botones de los flujos de la bomba para introducir los datos de diseño y de operación; estos botones aparecen cuando se ha elegido la opción de calculo **Entrando Datos de Flujo** (ver Fig. 111).

Fig. 113 Controles de los flujos de la bomba.

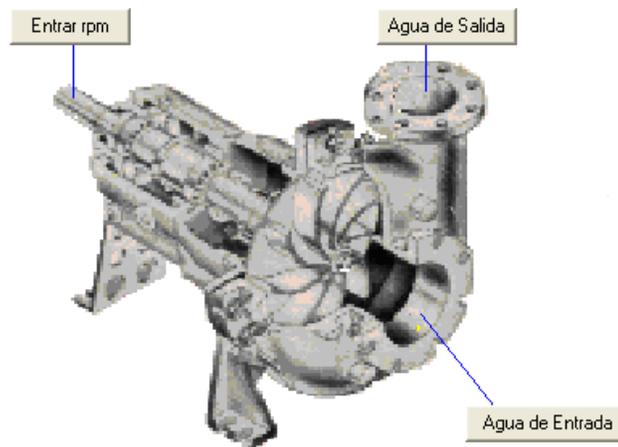


Fig. 114 Botón para entrar los datos de la curva del comportamiento de la bomba.

Entre Datos de la Curva

2.6.2 Datos de entrada. Para llevar a cabo los cálculos a carga parcial en una bomba se requieren los siguientes datos: Datos de diseño, datos de operación y datos de la curva de la bomba.

Los datos de diseño son la temperatura, la presión y el flujo volumétrico de la entrada y salida de la bomba; de la diferencia de presiones se calcula la cabeza o carga de la

bomba en condiciones de diseño. Otros datos de diseño necesarios son la velocidad del eje en rpms y la eficiencia.

El único dato de operación necesario es una nueva velocidad del eje de la bomba, puesto que los demás datos en condiciones de operación son calculados a partir de los datos de diseño.

Fig. 115 Datos de diseño de los flujos de la entrada y salida de la bomba.



Fig. 116 Otros datos de diseño de la bomba.

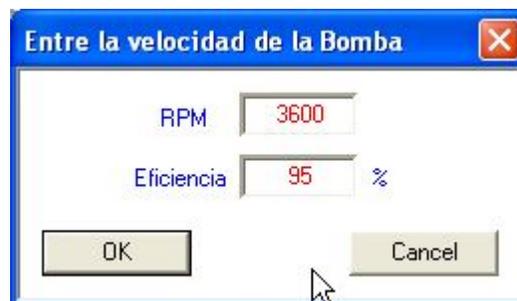


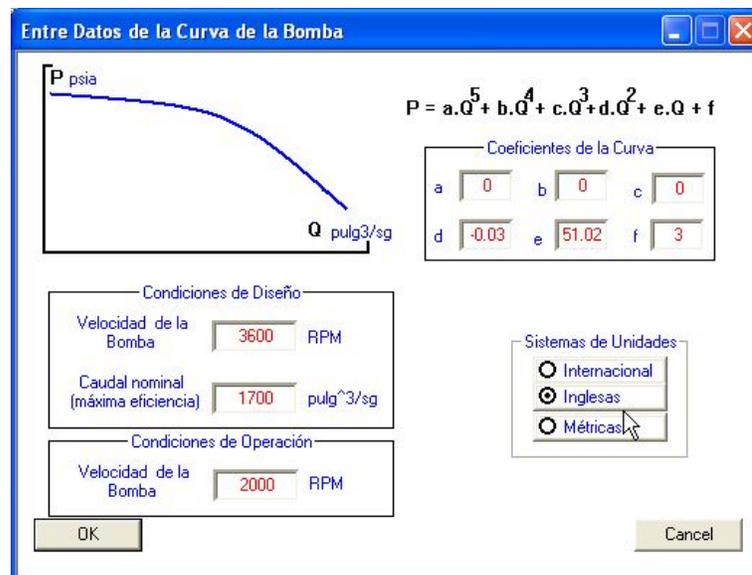
Fig. 117 Dato de operación de la bomba.



Los datos de la curva de la bomba son requeridos cuando se ha elegido la opción de cálculo, **Entrando Datos de la Curva** (ver Fig. 111); se introducen de manera directa el caudal o flujo volumétrico de diseño, la velocidad en rpms de diseño y la velocidad en rpms en operación y los coeficientes de la curva (ver Fig. 118).

2.6.3 Funcionamiento. Pasos para llevar a cabo los cálculos a carga parcial con el programa Bomba_Carga_Parcial.exe.

Fig. 118 Datos de la curva del funcionamiento de la bomba.



1. Cree un archivo de datos: Oprima el comando **Crear Archivo de Datos** en el menú **Archivo** como se muestra en la figura 119. Al oprimir el comando se abre la ventana **Guardar como** de Windows (ver Fig. 120), donde se le da un nombre al archivo de datos y se selecciona el directorio donde se guarda. Después de oprimir el botón **Guardar** aparecerán los comandos del programa.

2. Seleccione la opción de cálculo para el programa (ver Fig. 121): Oprima sobre uno de los controles de la figura 111.

Fig. 119 Creando un archivo de datos de la bomba.

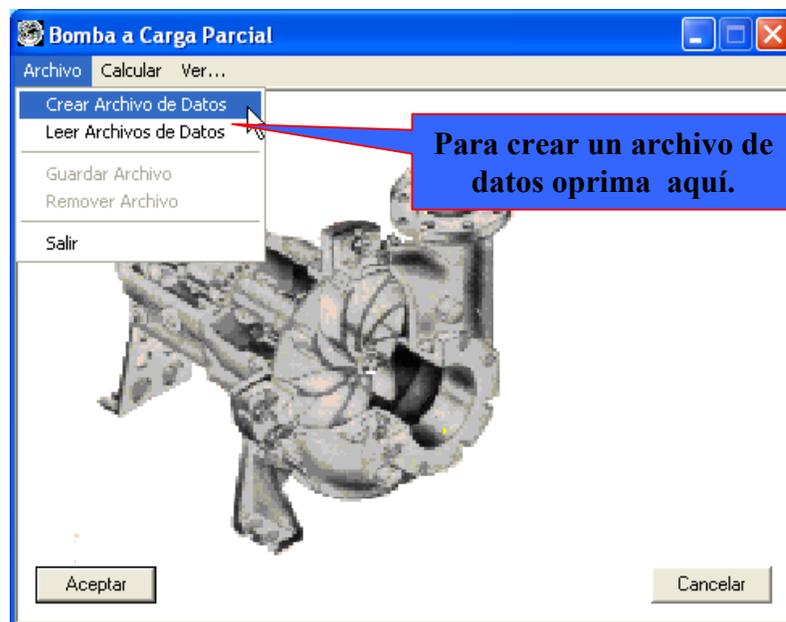
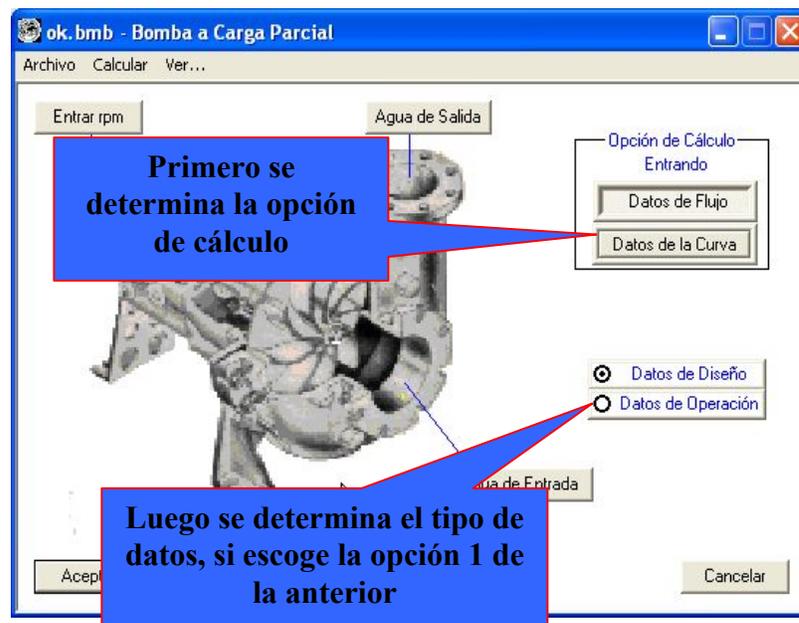


Fig. 120 Guardar como... de la bomba.



Fig. 121 Seleccionando la opción de cálculo de la bomba.



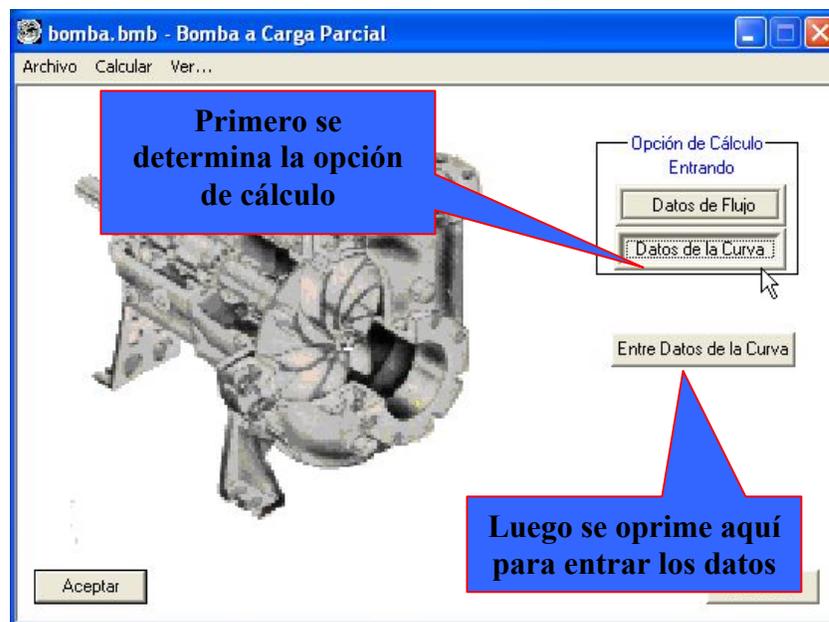
3. Si se ha elegido la opción de cálculo, **Entrando Datos de Flujo**, entre los datos de diseño: Oprima sobre el control **Datos de Diseño** de la figura 112; después oprima sobre los botones de los flujos; y a continuación se abren las ventanas

respectivas donde se recogen los datos al oprimir el botón OK (ver Fig. 115 y Fig. 116).

4. Entre el dato de operación: Oprima sobre el control **Datos de Operación** de la figura 112; después oprima sobre el botón del eje de la bomba; y a continuación se abre la ventana respectiva donde se recoge la nueva velocidad al oprimir el botón OK (ver Fig. 117).

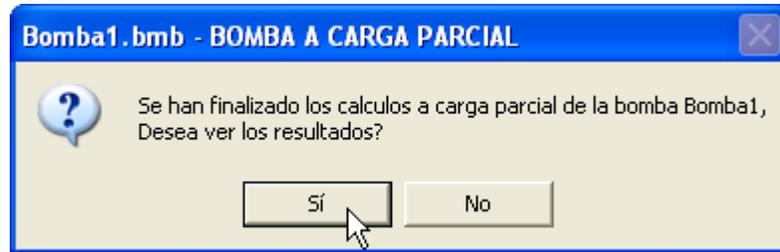
5. Si se ha elegido la opción de cálculo, **Entrando Datos de la Curva**, entre los datos de la curva del comportamiento de la bomba (ver Fig. 122): Oprima el botón **Entre Datos de la Curva** de la figura 114; y a continuación se abren la ventana de la figura 118 donde se recogen los datos al oprimir el botón OK.

Fig. 122 Entrando datos de la curva del comportamiento de la bomba.



6. Realice los cálculos a carga parcial: Oprima el comando del menú Calcular (ver Fig. 109). Cuando el programa ya ha terminado los cálculos pregunta con la ventana de la figura 123 si se desea ver los resultados o no.

Fig. 123 Fin de los cálculos de la bomba.



7. Guarde o elimine los datos introducidos: Por último se puede conservar todos los datos introducidos anteriormente en archivos que tienen el mismo nombre del archivo de datos creado al principio accionando el comando **Guardar archivo** del menú **Archivo** o eliminarlos accionando el comando **Remove Archivo**.

Los resultados de los cálculos a carga parcial de la bomba son mostrados por medio del programa WordPad de Windows; para verlos se oprime el elemento **Informe de Resultados** del menú **Ver...** (ver Fig. 124 y 125).

Fig. 124 Revisando los resultados de la bomba.

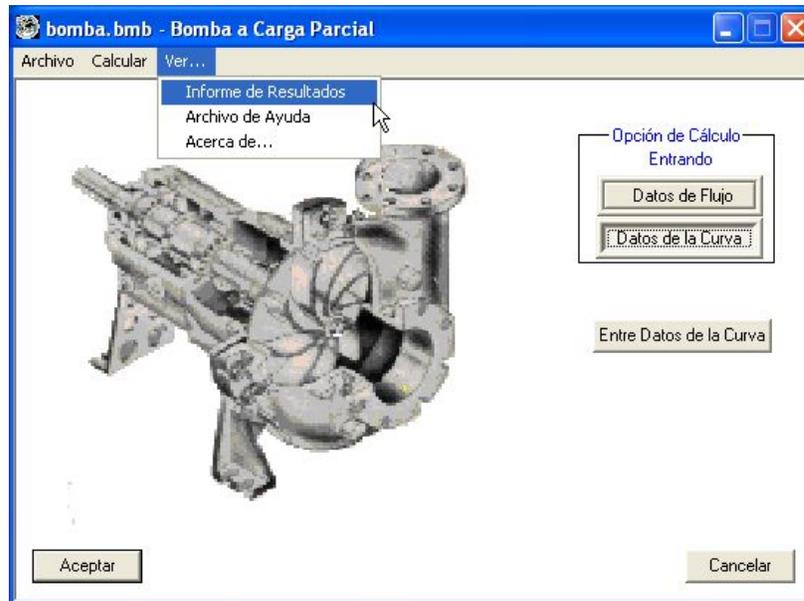
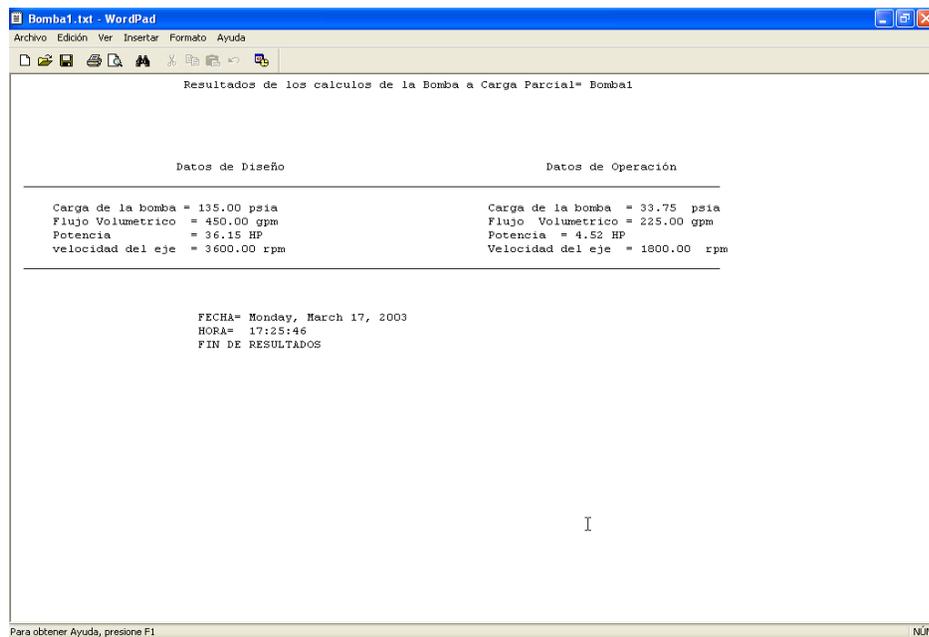


Fig. 125 Archivo de texto con los resultados de la bomba.



3. EJEMPLOS DE APLICACIÓN

En este capítulo se mostrará un ejemplo de cálculo por cada programa desarrollado. Para el ejemplo de la turbina se utiliza un ejercicio desarrollado en la bibliografía, los ejemplos de la bomba y la válvula se resolverán de forma manual usando los medios convencionales para ello, y luego se harán los cálculos utilizando el programa desarrollado para el equipo; para el intercambiador, condensador y la caldera se usa un ejemplo supuesto directamente en el programa. Posteriormente se examinan los resultados obtenidos por ambos procedimientos y se comprueba el grado de precisión.

3.1 PROGRAMA TURBINA_CARGA_PARCIAL.EXE.

En el artículo de la ASME¹⁴ mencionado anteriormente se encuentra un pequeño ejemplo para el cálculo de la eficiencia de una turbina; los datos de dicho ejemplo son usados en el programa para comparar sus resultados. Se utiliza este ejemplo ya dispuesto con sus resultados puesto que el proceso de cálculo es muy extenso como para realizarlo en estas paginas de nuevo.

Los datos son los siguientes:

Vapor de entrada en condiciones de diseño:	{ Flujo másico = 2115387 lbm/hrs. Presión = 2415 psia. Temperatura = 1000 F.
Vapor de salida en condiciones de diseño:	{ Flujo másico = 2115387 lbm/hrs. Presión = 604.1 psia. Temperatura = No es necesario este dato para el calculo de la eficiencia
	{ Flujo másico = 1846913 lbm/hrs

Vapor de entrada en condiciones de operación:

Los demás datos de flujos no son necesarios para el cálculo de la eficiencia pero si se quiere calcular la presión de salida en condiciones de operación, entonces se debe introducir la presión del vapor de entrada en condiciones de operación; los demás datos del ejemplo son:

Tipo de Turbina: Alta Presión-3600 rpm-sin condensación-1 etapa de gobierno.

Diámetro Pitch de la etapa de gobierno = 41.1 pulg.

Número de válvulas = 4.

Número de flujos paralelos de la sección de Expansión: 1.

Con los datos anteriores la eficiencia de la turbina es, según el ejemplo: 82.07 % y según el programa la eficiencia es de 82.08%, que se puede observar en la figura 126, siendo el resultado del programa muy cercano al del ejemplo.

Fig. 126 Informe de resultados del programa Turbina_Carga_Parcial.exe.

Sección # 1 Turbina Tipo B	
Datos de diseño	Datos de operación
Vapor de Entrada	
El flujo másico es: 2115390.00 Lbs/hr	El flujo másico es: 1846913.00 Lbs/hr
La presión es: 2415.00 psia	La presión es: 500.25 psia
La temperatura es: 1000.00 F	La temperatura es: 1000.00 F
Vapor de Salida	
La presión es: 604.10 psia	La presión es: 230.50 psia
El número flujos paralelos a la entrada es: 1	
El número de válvulas de control es: 4	
El diametro Pitch de la etapa de gobierno es: 41.10 pulg	
La eficiencia de la sección es: 82.08 o/o	

3.2 PROGRAMA INTERCAMBIADOR_CARGA_PARCIAL.EXE.

¹⁴ Spencer, R.C. Cotton, K.C. Cannon, C.N. Op. cit., p.49

Como en la bibliografía recopilada no se encuentra un ejemplo para la comparación de resultados se asumen datos de entrada para una operación lógica de equipos de intercambiador de calor de acuerdo a la termodinámica; y como los programas de intercambiadores, condensadores y calderas a carga parcial realizan los mismos cálculos solo es necesario exponer un ejemplo por los tres programas para observar los resultados. Los datos del ejemplo son los siguientes:

Datos en condiciones de diseño.

Flujo Caliente (Agua) Entrando al Equipo: {
 Flujo másico = 300000 lbm/hr.
 Presión = 150 psia.
 Temperatura = 455 F.

Flujo Caliente (Agua) Saliendo del Equipo: {
 Flujo másico = el mismo anterior.
 Presión = 140 psia.
 Temperatura = 200 F.

Flujo Frío (Agua) Entrando del Equipo: {
 Flujo másico = 250000.
 Presión = 40 psia.
 Temperatura = 100 F.

Flujo Frío (Agua) Saliendo del Equipo: {
 Flujo másico = El mismo anterior.
 Presión = 38 psia.
 Temperatura = 300 F.

Datos en condiciones de operación.

Flujo Caliente (Agua) Entrando del Equipo: {
 Flujo másico = 270000 lbm/hr.
 Presión = 149 psia.
 Flujo másico = El mismo anterior.

Flujo Caliente (Agua) Saliendo del Equipo: {
 Presión = Se calcula.
 Temperatura = Se calcula.

Flujo Frío (Agua) Entrando del Equipo:	{ <ul style="list-style-type: none"> Flujo másico = 200000 lbm/hr. Presión = 45 psia. Temperatura = 110 F.
Flujo Frío (Agua) Saliendo del Equipo:	{ <ul style="list-style-type: none"> Flujo másico = El mismo anterior. Presión = Se calcula. Temperatura = Se calcula.

Los datos del diseño mecánico son los siguientes: 5200 pies cuadrados de área total de transferencia de calor y se selecciona: ***calcular el coeficiente de transferencia global de calor***. Los exponentes para el análisis del comportamiento del equipo se dejan predeterminados con el valor de 0.8. Usando los anteriores datos en el programa Intercambiador_Carga_Parcial.exe y eligiendo una configuración de intercambiador a contra flujo los resultados son los mostrados en la figura 127.

Observando la figura se halla que los valores obtenidos después de realizar los cálculos a carga parcial están en rangos lógicos de acuerdo a la termodinámica de estos equipos.

3.3 PROGRAMA VALVULA_CARGA_PARCIAL.EXE.

En la literatura recopilada no existe un ejemplo del comportamiento de la válvula a carga parcial pero a fin de comparar resultados es posible realizar cálculos manualmente y con el programa.

Fig. 127 Informe de resultados del Intercambiador_Carga_Parcial.exe

Agua Caliente

Entrada	Salida
Temperatura = 450.00 F Presion = 149.00 psia Flujo másico = 270000.00 Lb/hrs	Temperatura = 352.63 F Presion = 139.89 psia

Agua Fria

Entrada	Salida
Temperatura = 110.00 F Presion = 45.00 psia Flujo másico = 200000.00 Lb/hrs	Temperatura = 183.88 F Presion = 43.02 psia

Los siguientes son los datos de entrada al programa válvula carga parcial.exe:

Datos de Diseño: { Caudal = 460 gpm.
Caída de presión = 120 psia o 8.27 bares

Datos de Operación: { Caudal = 300 gpm.
Caída de presión = Se calcula

Los datos de entrada anteriores se introducen al programa habiéndose seleccionado la opción de cálculo: *conociendo caudal de operación.*

Se asume que la válvula se encuentra totalmente abierta, por lo tanto tiene un tipo de comportamiento cuadrático.

Cálculos:

1. Se calcula el coeficiente de flujo de la válvula en condiciones de diseño:

$$C_v = Q * \sqrt{\frac{G_f}{\Delta P}} = 460 * \sqrt{\frac{1}{120}} = 41.99.$$

2. Se calcula la nueva caída de presión con el nuevo caudal:

$$\Delta P = G_f * \left(\frac{Q}{C_v}\right)^2 = 1 * \left(\frac{300}{41.99}\right)^2 = 51.04 \text{ psia.}$$

Al introducir los anteriores datos al programa también da como resultado una caída de presión de 51.04 psia o 3.52 bares como se observa en la figura 128.

Fig. 128 Resultados del programa Válvula_Carga_Parcial.exe.

Resultados de los calculos de la Válvula a Carga Parcial= val

Datos de Diseño	Datos de Operación
Caudal = 460.00 gpm	Caudal = 300.00 gpm
Caída de presión = 8.27 Bar	Caída de presión = 3.52 Bar

FECHA= Thursday, August 14, 2003
 HORÀ= 14:49:39
 FIN DE RESULTADOS

3.4 PROGRAMA BOMBA_CARGA_PARCIAL.EXE.

A fin de comprobar los resultados del programa Bomba_Carga_Parcial.exe también es posible realizar cálculos de forma manual, ya que las ecuaciones utilizadas son las leyes de afinidad de las bombas centrífugas expuestas en la bibliografía. Los siguientes datos son asumidos como ejemplo:

Condiciones de diseño: $\left\{ \begin{array}{l} \text{Fluido} = \text{Agua.} \\ \text{Caudal} = 450 \text{ gpm.} \\ \text{Presión de entrada} = 72.51 \text{ psia.} \\ \text{Temperatura de entrada} = 301 \text{ F.} \\ \text{Presión de salida} = 290 \text{ nsia.} \end{array} \right.$

Condición de operación: $\left\{ \begin{array}{l} \text{Velocidad del Eje} = 1800 \text{ rpm.} \end{array} \right.$

Se asume como eficiencia de la bomba = 95%.

Los datos de entrada anteriores se introducen al programa habiendo seleccionado la opción de cálculo: *entrando datos de flujo*.

De las presiones de entrada y salida se calcula la carga de la bomba que es 217.5 psia; con la temperatura y presión de entrada se comprueba si la bomba está en peligro de cavitación.

Para conocer el nuevo caudal en la condición de operación se utiliza la siguiente ley de afinidad:

$$Q_2 = Q_1 * \left(\frac{N_2}{N_1}\right) = 450 * (1800/3600) = 225 \text{ gpm.}$$

Para conocer la nueva carga de bomba se utiliza la siguiente ley de afinidad:

$$H_2 = H_1 * \left(\frac{N_2}{N_1}\right)^2 = 217.5 * (0.5)^2 = 54.37 \text{ psia.}$$

Para calcular la nueva potencia se utiliza la siguiente expresión:

$$Potencia2 = Potencia1 * \left(\frac{N_2}{N_1}\right)^3, \text{ pero primero se debe calcular la potencia1 por}$$

medio de la siguiente ecuación:

$$Potencia1 = \frac{Q1 * H1(pies) * gravedadespecifica}{3960 * Eficiencia}$$

$$H1(pies) = \frac{psia * 2.31}{gravedadespecifica} = 217.5 * 2.31 / 1 = 502.425 \text{ pies.}$$

$$Potencia1 = \frac{450 * 502.425 * 1}{3960 * 0.95} = 60 \text{ HP.}$$

$$Potencia2 = 60 * 0.5^3 = 7.5 \text{ HP.}$$

Al introducir los datos de entrada al programa da como resultado los siguientes valores como se puede observar en la figura 117:

Caudal #2 = 225 gpm.

Carga de la bomba #2 = 54.39 psia.

Potencia #2 = 7.51 HP.

Los anteriores parámetros son valores muy cercanos o los mismos a los calculados manualmente.

Fig. 129 Resultados Bomba_Carga_Parcial.exe.

Resultados de los calculos de la Bomba a Carga Parcial= bomb

Datos de Diseño	Datos de Operación
Carga de la bomba = 217.49 psia	Carga de la bomba = 54.37 psia
Flujo Volumetrico = 450.00 gpm	Flujo Volumetrico = 225.00 gpm
Potencia = 60.07 HP	Potencia = 7.51 HP
velocidad del eje = 3600.00 rpm	Velocidad del eje = 1800.00 rpm

4. CONCLUSIONES

- Se han alcanzados los objetivos propuestos al comienzo del presente proyecto, puesto que se han implementado por medio de un software cálculos de carga parcial para: Turbinas de vapor, intercambiadores de calor, bombas, calderas, válvulas y condensadores; se han provisto de una interfaz de usuario amigable con filosofía Windows para cada uno de los anteriores equipos y se han integrado al software SB2000.
- Los cálculos a carga parcial de los equipos nombrados anteriormente se realizan de una forma más fácil y rápida utilizando estos software, ya que estos cálculos son muy extensos y en ocasiones es necesario realizar varias iteraciones.
- Teniendo en cuenta la teoría termodinámica y la hidráulica, estos programas dan como resultados valores dentro de rangos coherentes.
- Los programas le dan al usuario la libertad de escoger o seleccionar la configuración de cada uno de los equipos, el entrar o calcular ciertos datos, guardar los datos de entrada en archivos generados en cualquier directorio del computador, eliminar tales archivos, manejar diferentes ejemplos de un mismo equipo sin necesidad de perder datos de uno de ellos ni de cerrar el software y de escoger con cierta limitación un sistema de unidades de trabajo.

- Los informes de los resultados obtenidos de los diferentes programas pueden ser impresos desde un editor de texto de libre distribución como lo es el WordPad.exe.
- Al desarrollar estos programas con los modelos expuestos anteriormente no se busca la exactitud de los resultados, sino acercarse lo más posible al comportamiento que tendrían estos equipos al operar a carga parcial en una planta real.

5. RECOMENDACIONES.

- Se recomienda introducir valores coherentes por parte del usuario a los programas del presente proyecto, por que cuando estos datos no lo son nunca se llega a una solución convergente o lógica de acuerdo a la teoría termodinámica e hidráulica.
- Se recomienda revisar estos programas en casos de equipos reales operando en plantas térmicas para ajustar los modelos de calculo a carga parcial que son utilizados dentro de ellos, puesto que tales modelos son teóricos y no necesariamente predicen con gran precisión el comportamiento de los equipos en dichos casos.
- Se recomienda implementar más herramientas y funcionalidad a cada software de acuerdo a las necesidades futuras.
- Se recomienda el estudio del procedimiento para predecir el comportamiento de las turbinas a vapor expuesto en el artículo de la ASME mencionado anteriormente.

BIBLIOGRAFIA

- **INSTITUTION OF MECHANICAL ENGINEERS.** International Conference, Proceedings of the Institution of Mechanical Engineers, Steam Plants for the 1990s. Mechanical Engineering Publications Limited.
- **Spencer, R.C.; Cotton, K.C. y Cannon, C.N.** A Method for Predicting the Performance of Steam Turbine-Generators... 16,500 KW And Larger. ASME, 1974.
- **COOKE, D.H.** On Prediction of Off-Design Multistage Turbine Pressures by Stodola's Ellipse.
- **KNOEBEL, John Edmard.** A computer model for power plant performance calculations. The University of Texas austin.
- **ENTERSOFTWARE.** Gate Cycle User's Guide: Appendix G.
- **KAKAS, John Willy.** Boilers, Evaporators, and Condensers.
- **BABCOKB AND WILLCOX.** Steam, Its generation and Use.
- **HEMISPHERE PUBLISHING CORPORATION.** Heat Exchanger design Handbook. 1 Heat exchanger Theory.
- **GARCIA-BORRAS, Thomas.** Manual for Improving Boiler and Furnace performance. Gulf Publishing Company.

- **KOHAN, Anthony L. & M. SPRING, Harry.** Boiler Operator's Guide. Third edition. McGraw Hill.
- **KARASSIK KRUTZSCH, Fraser Messina.** Pump Handbook. Second edition. McGraw Hill.
- **BENAROYA, Alfred.** Fundamentals and application of centrifugal pumps. Petroleum publishing Company.
- **WARRING, R. H.** Pumps: Selection, systems and applications, 2nd Edition. Gulf publishing Company.
- **MCNAUGHTON, Kenneth J.** Bombas: Selección, uso y mantenimiento. McGraw Hill.
- **Ing. D'CROZ, Néstor Raúl.** Curso de turbo maquinas Hidráulicas. Universidad Industrial de Santander.
- **HUTCHISON, J.W...** Handbook of Control Valves, 2nd Edition, McGraw Hill.
- **SKOUSEN, Philip L...** Valve Handbook, McGraw Hill.
- **ZAPPE, R.W...** Valve Selection Handbook, McGraw Hill.
- **CRANE.** Flow of fluids through valves, fittings, and pipe, McGraw Hill.
- **KRUGLINSKI, David J.** Programación avanzada con visual C++, McGraw Hill.

Anexo A. Diagrama de flujo para el cálculo a carga parcial de turbinas y generadores a vapor de 16500 KW y mayores.¹⁵

Definiciones.

Ted = Temperatura de entrada del fluido en condiciones de diseño.

Ter = Temperatura de entrada del fluido en condiciones de reales.

T = Tipo de turbina. (A-B-C-D-E-F-G).

Ped = Presión de entrada del fluido en condiciones de diseño.

Psd = Presión de salida del fluido en condiciones de diseño.

Per = Presión de entrada del fluido en condiciones reales.

Pg = Presión de la etapa de gobierno.

md = Flujo másico en condiciones de diseño.

mr = Flujo másico en condiciones reales.

N = Número de flujos paralelos a la entrada.

NV = Número de válvulas.

PD = Diámetro pitch de la etapa de gobierno.

M = Con condensación o sin condensación.

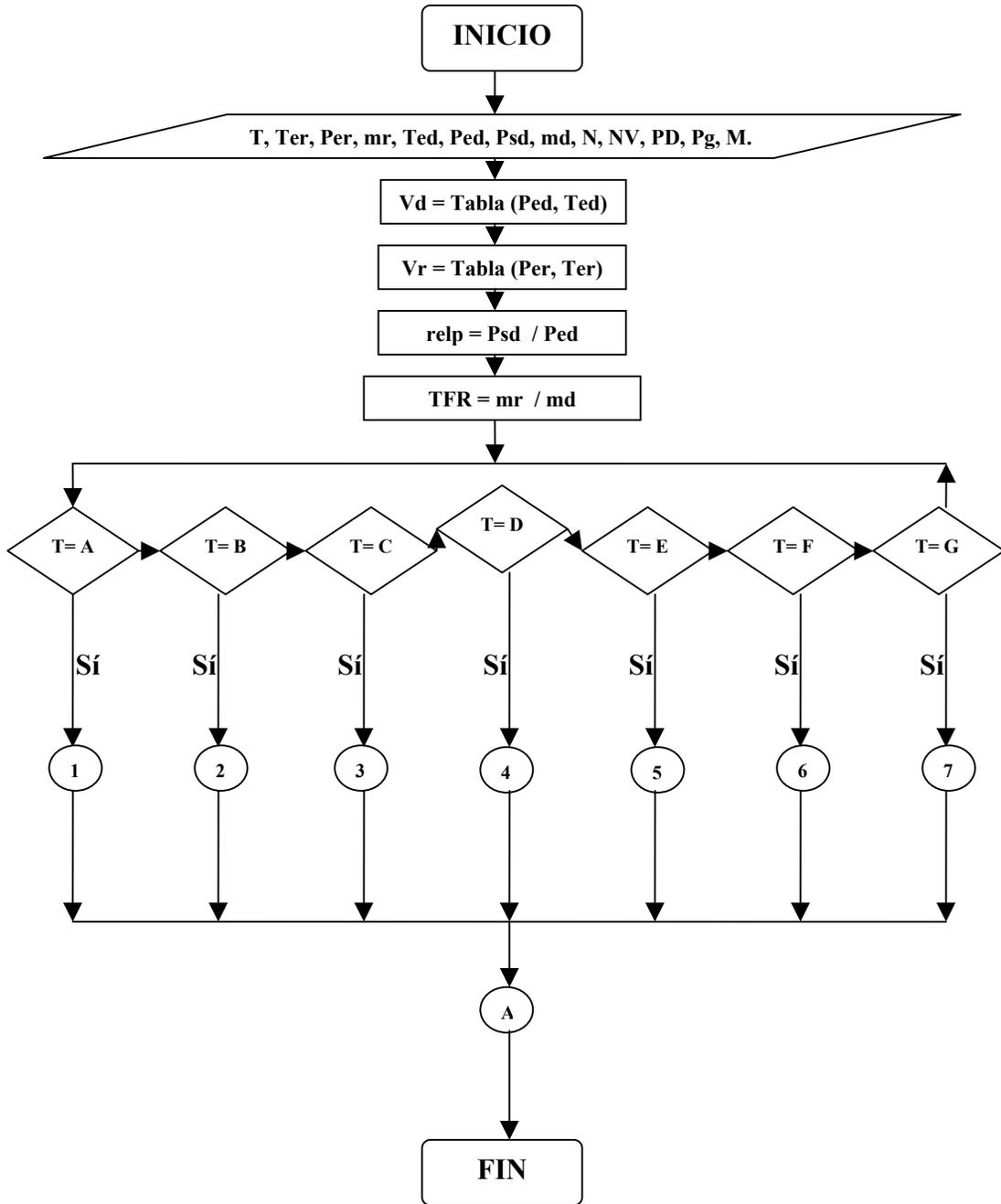
Ax,y = Coeficientes de las ecuaciones.

Vd = Volumen en condiciones de diseño.

Vr = Volumen en condiciones reales.

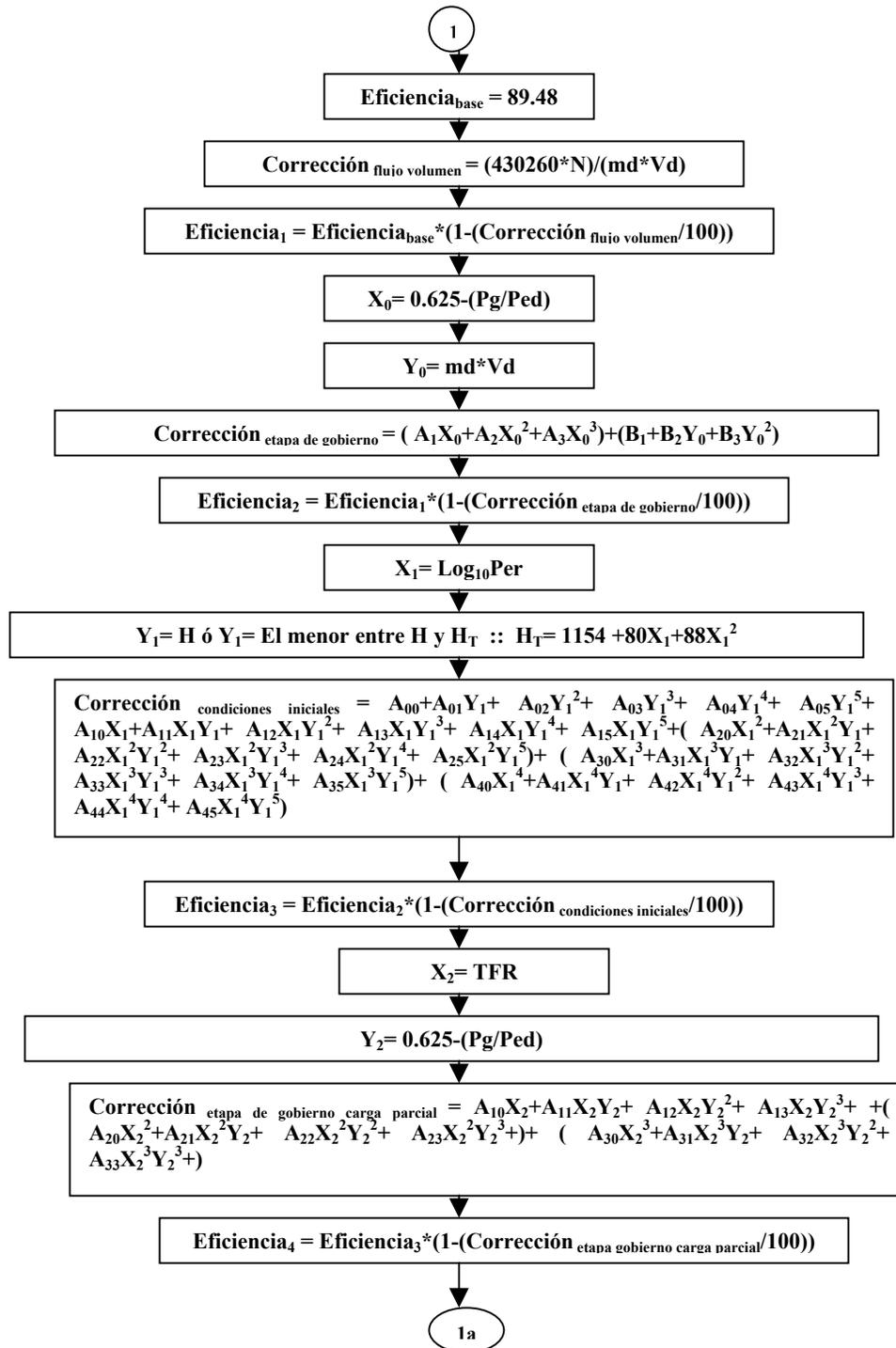
Corrección(x,y) = Corrección de la eficiencia.

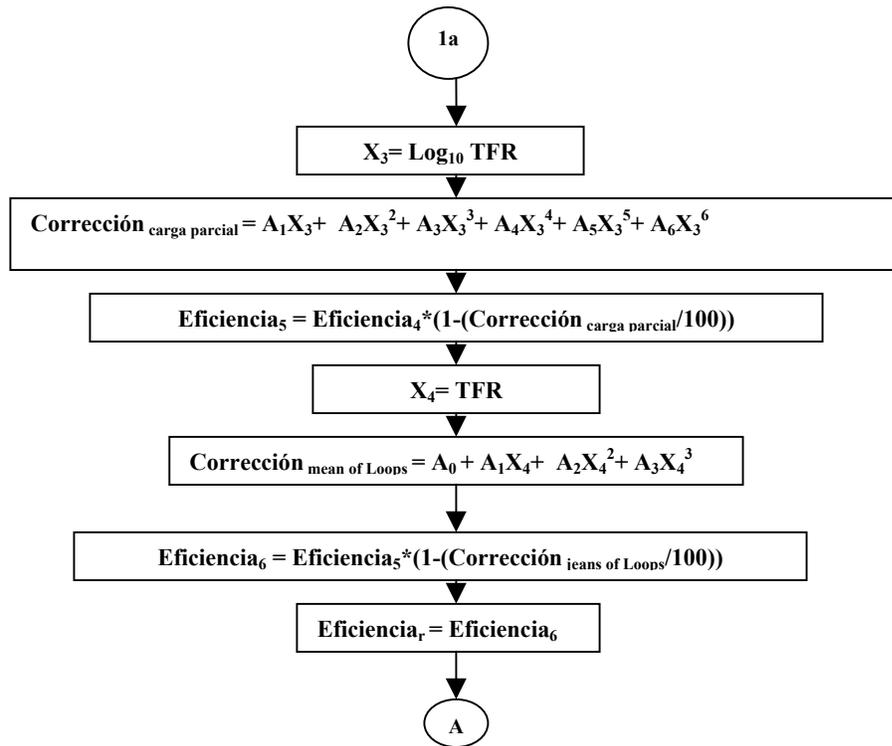
Diagrama de flujo para el calculo de las eficiencias en turbinas.



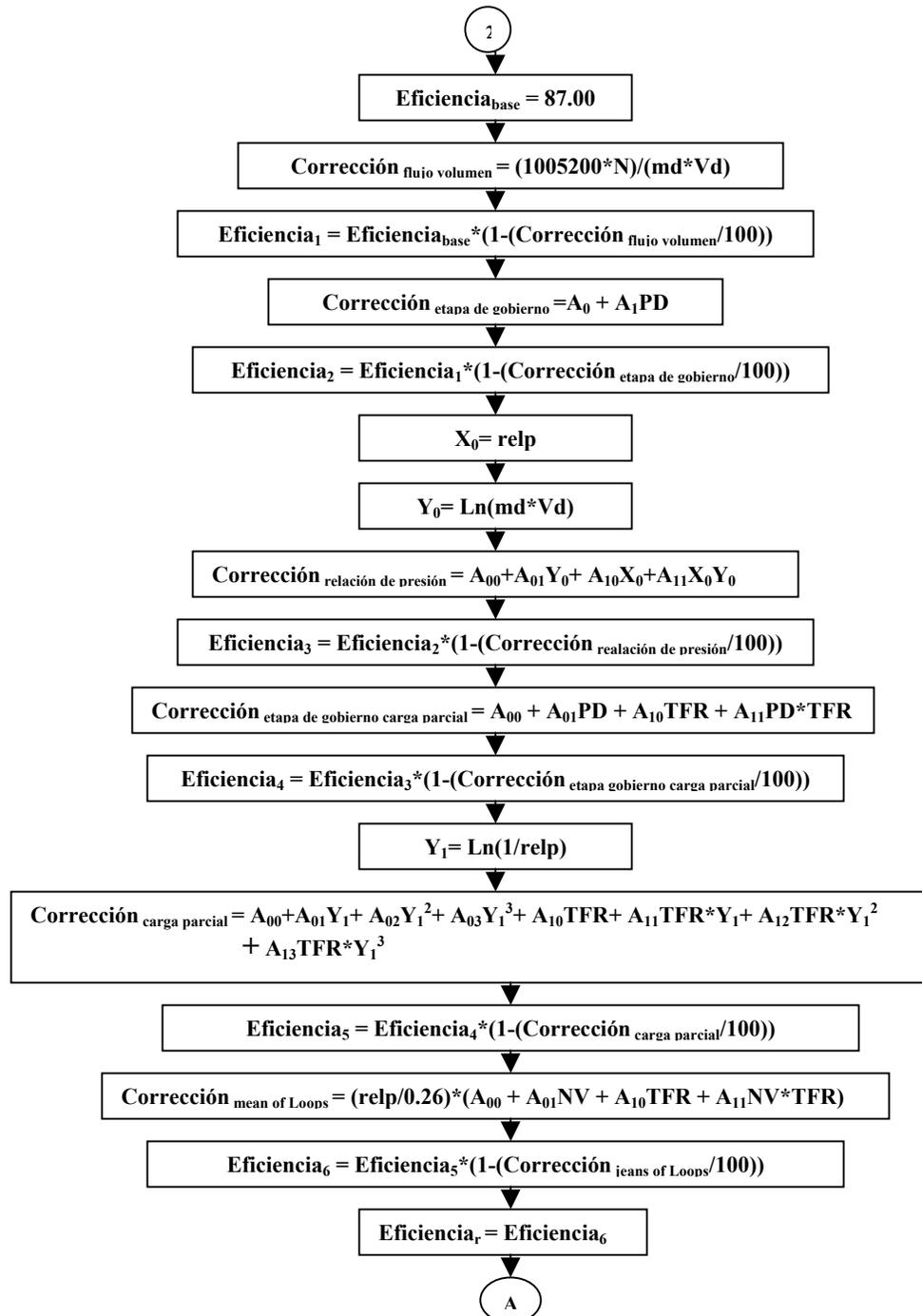
¹⁵ LARA, Adriana, Op. cit

Procedimiento para calcular la eficiencia de una turbina tipo A (sin recalentamiento, 3600 rpm, con condensación, 2 etapas de gobierno).

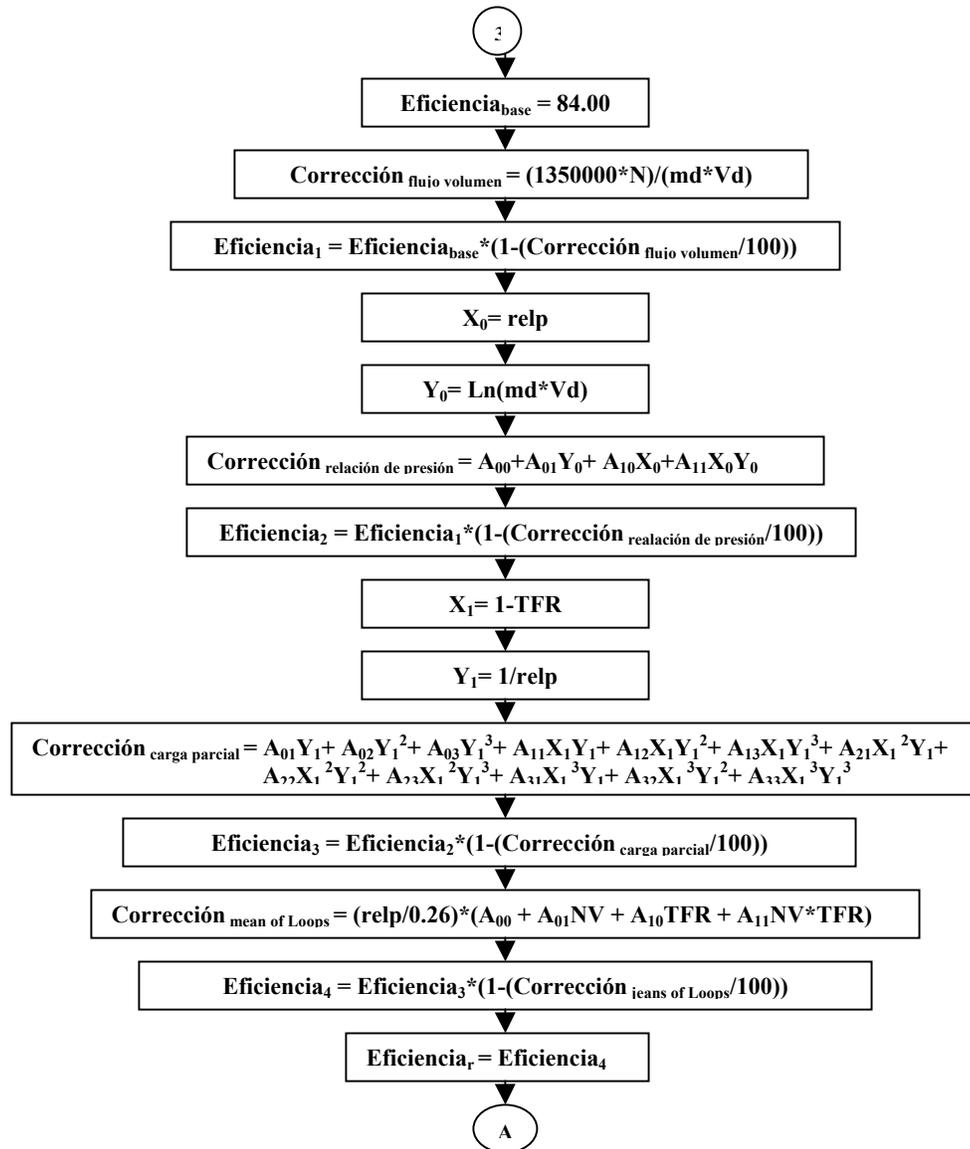




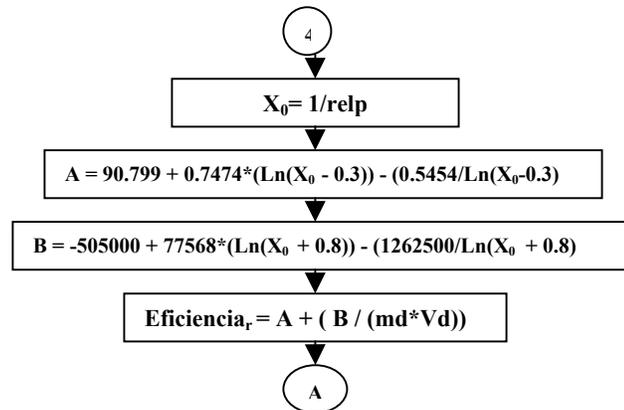
Procedimiento para calcular la eficiencia de una turbina tipo B (Secciones de alta presión, 3600 rpm, sin condensación, 1 etapa de gobierno)



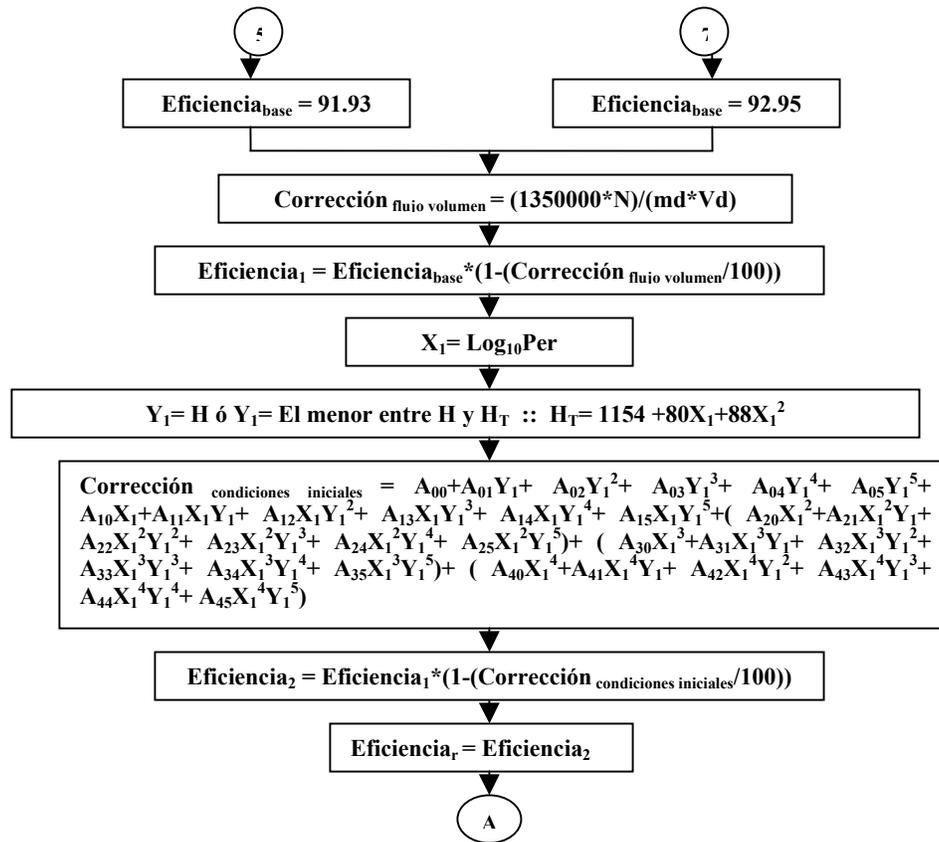
Procedimiento para calcular la eficiencia de una turbina tipo C (Secciones de alta presión, 3600 rpm, sin condensación, 2 etapas de gobierno)



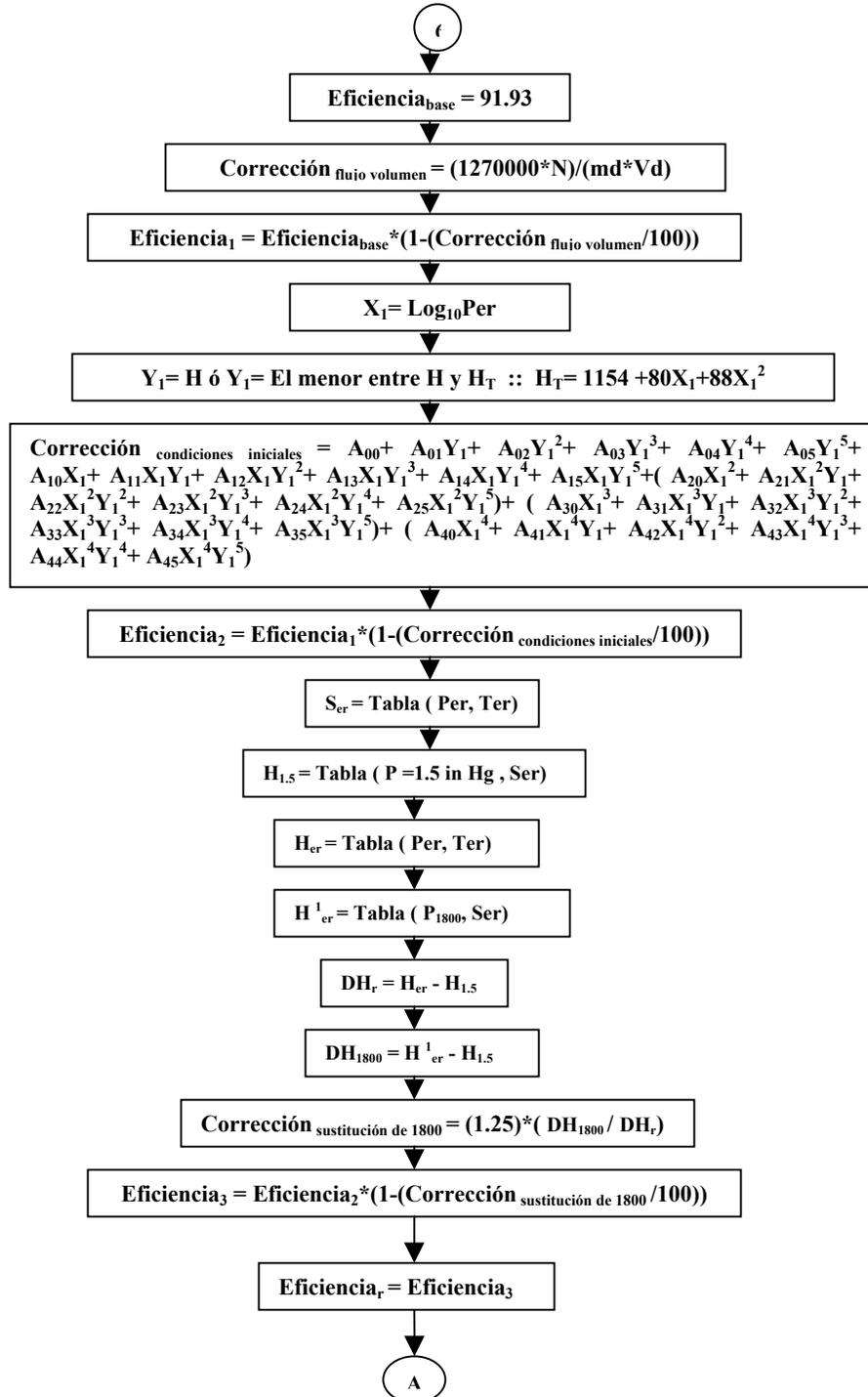
Procedimiento para calcular la eficiencia de una turbina tipo D (Secciones de presión intermedia, 3600 rpm, sin condensación, sin etapa de gobierno)



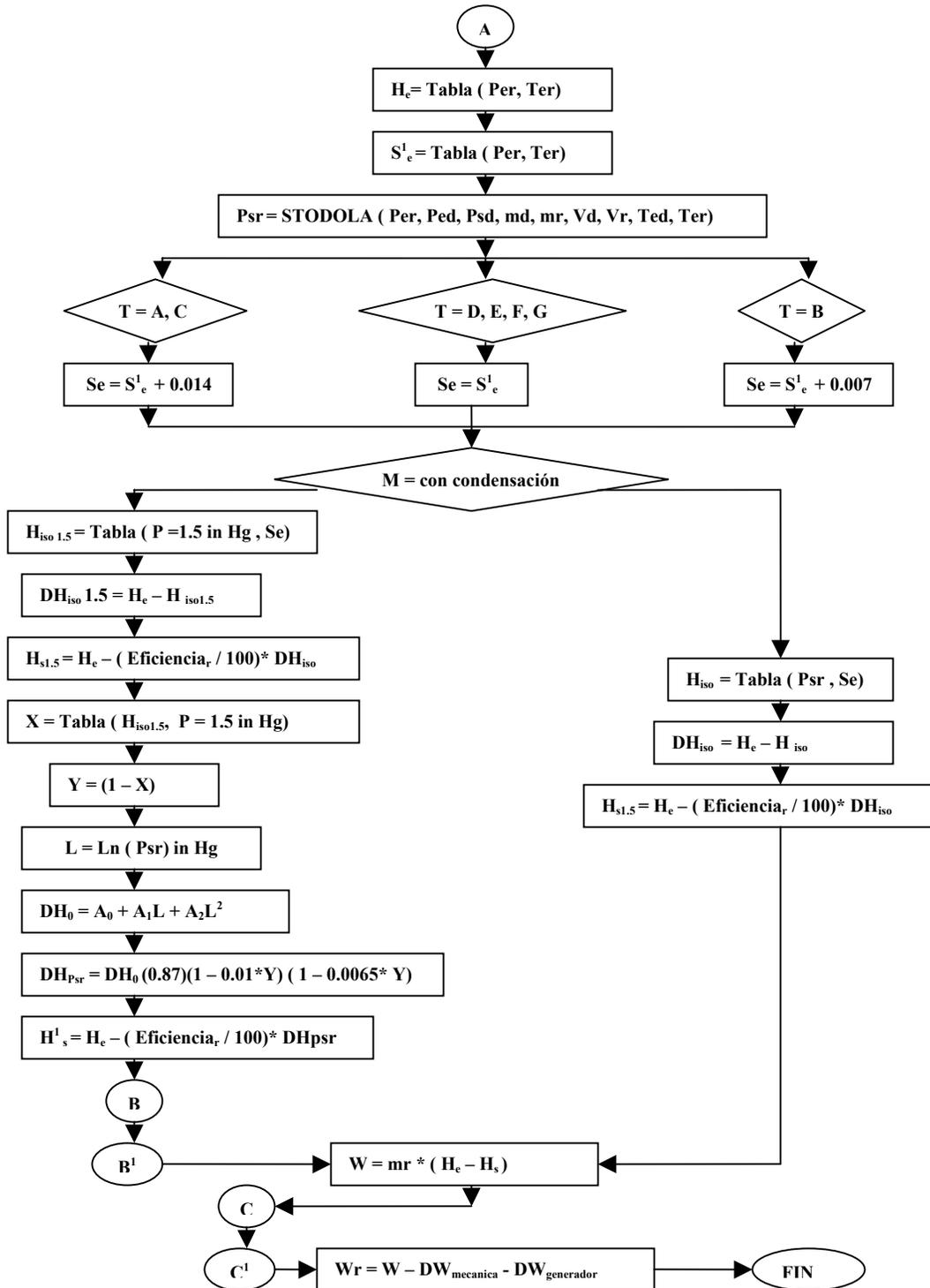
Procedimiento para calcular la eficiencia de una turbina tipo E (Secciones recalentadas, 3600 rpm, con condensación, sin etapa de gobierno) y tipo G (Secciones recalentadas, 1800 rpm, con condensación, sin etapa de gobierno)



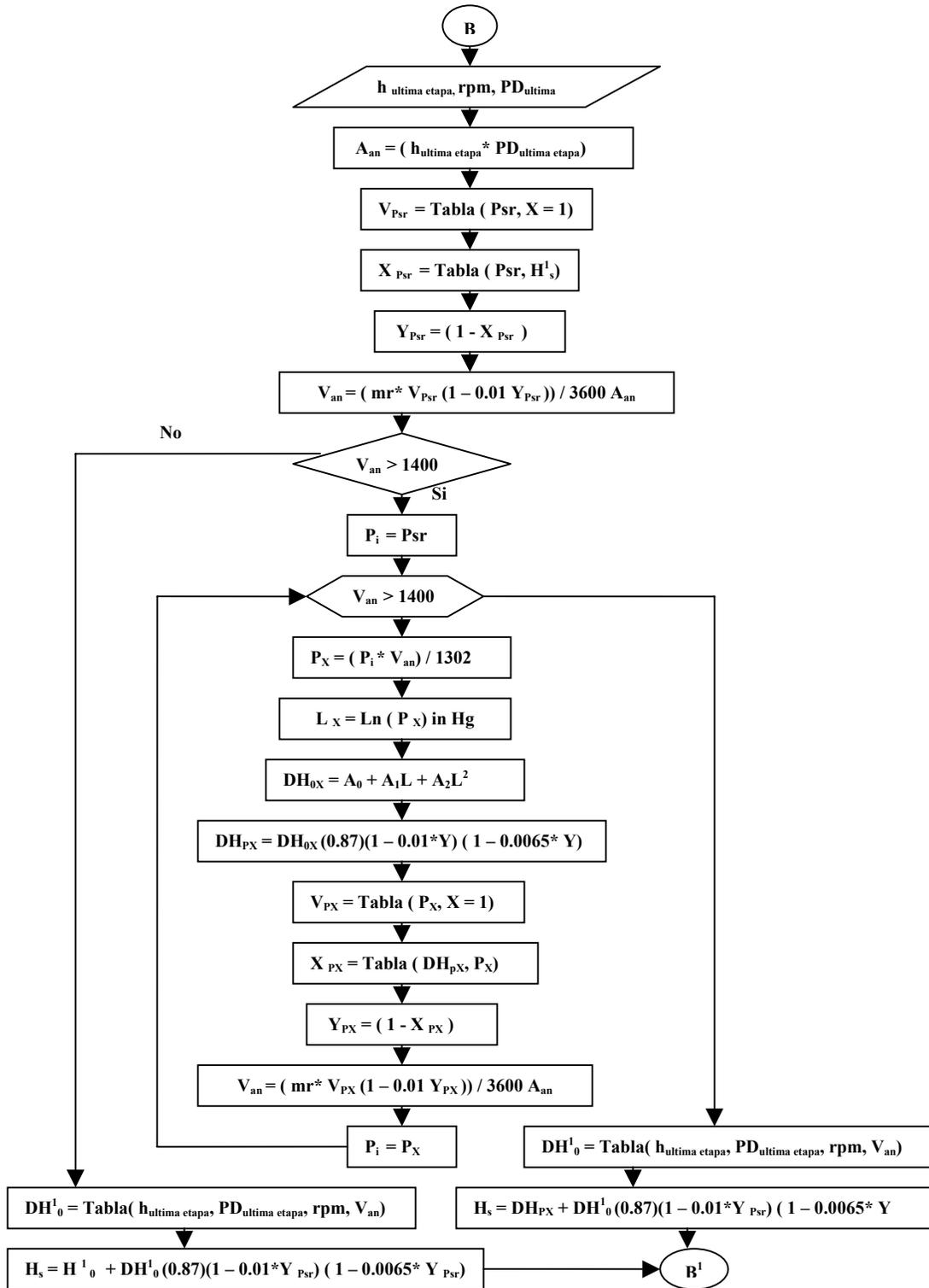
Procedimiento para calcular la eficiencia de una turbina tipo F (Secciones recalentadas, 3600/1800 rpm, con condensación, sin etapa de gobierno).



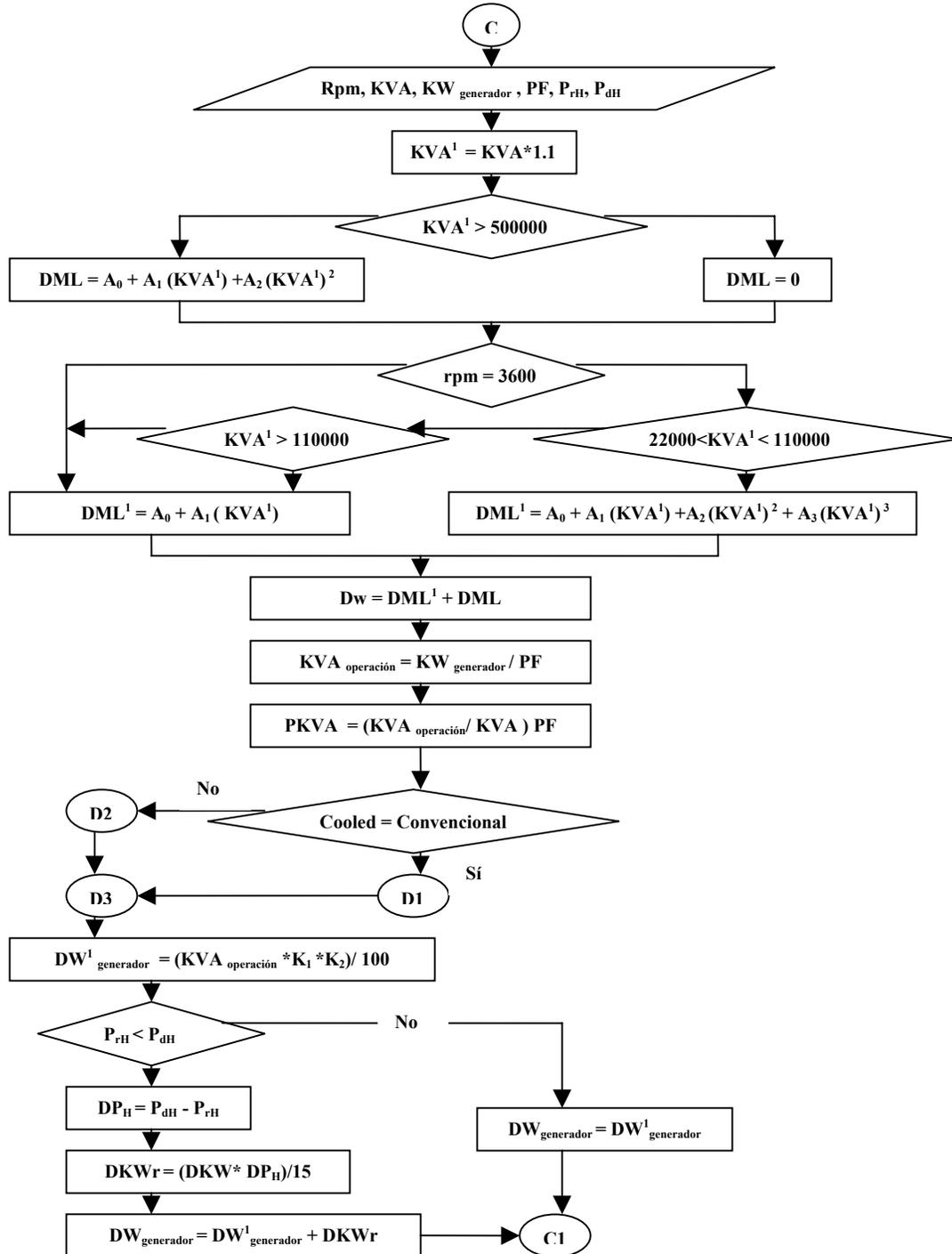
Procedimiento para el cálculo del punto final de la línea de expansión corregido y del trabajo de una turbina.

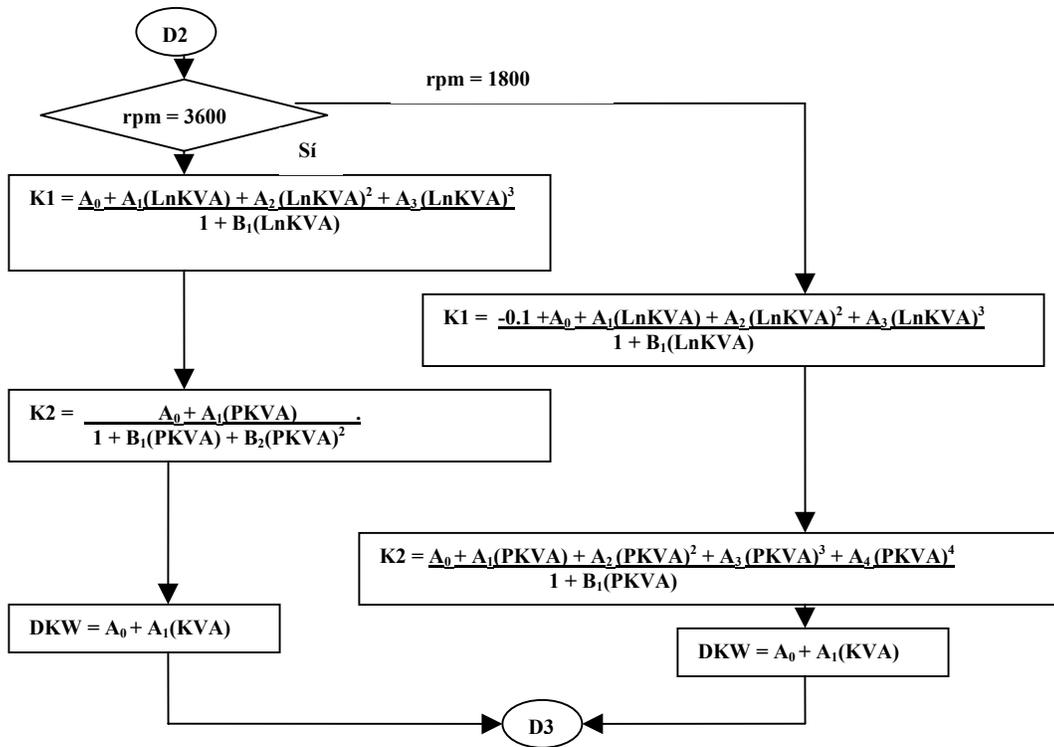
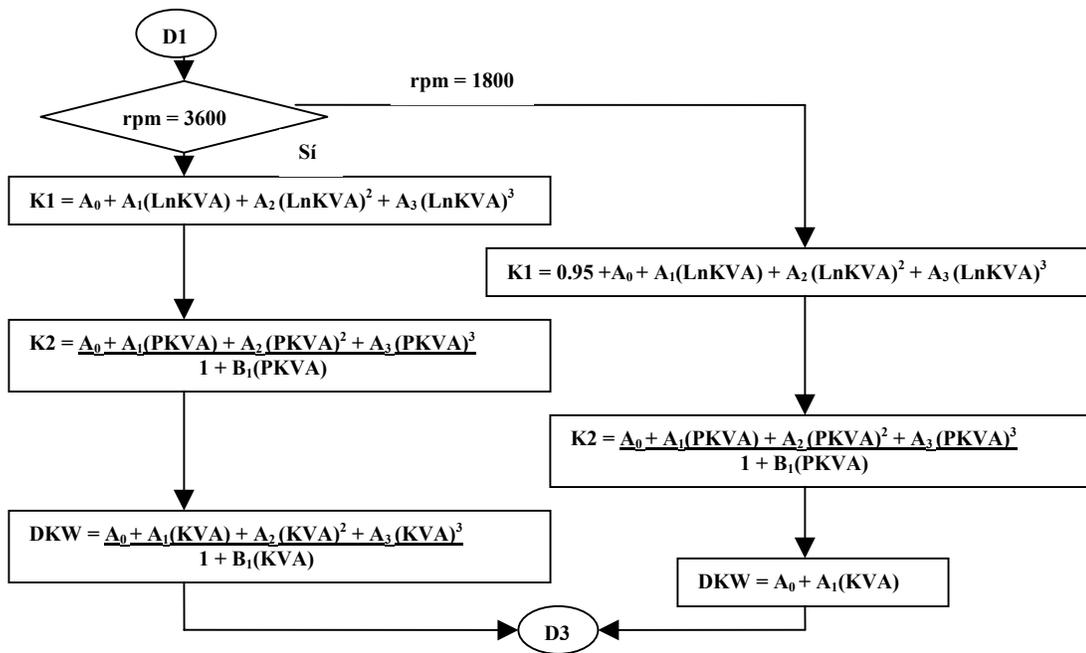


Procedimiento para el cálculo de la corrección por fugas del punto final de la línea de expansión.



Procedimiento para el cálculo de la corrección del trabajo en una turbina por pérdidas mecánicas y pérdidas en el generador.





Anexo B. Diagrama de flujo del cálculo a carga parcial para el intercambiador de calor¹⁶

Definiciones.

$T_{e_{cd}}$ = temperatura de entrada del fluido caliente en condiciones de diseño.

$T_{e_{co}}$ = temperatura de entrada del fluido caliente en condiciones de operación.

$T_{s_{cd}}$ = temperatura de salida del fluido caliente en condiciones de diseño.

$T_{s_{co}}$ = temperatura de salida del fluido caliente en condiciones de operación.

$T_{m_{cd}}$ = temperatura media del fluido caliente en condiciones de diseño.

$T_{m_{co}}$ = temperatura media del fluido caliente en condiciones de operación.

$T_{e_{fd}}$ = temperatura de entrada del fluido frío en condiciones de diseño.

$T_{e_{fo}}$ = temperatura de entrada del fluido frío en condiciones de operación.

$T_{s_{fd}}$ = temperatura de salida del fluido frío en condiciones de diseño.

$T_{s_{fo}}$ = temperatura de salida del fluido frío en condiciones de operación.

$T_{m_{fd}}$ = temperatura media del fluido frío en condiciones de diseño.

$T_{m_{fo}}$ = temperatura media del fluido frío en condiciones de operación.

$LMTD_{und}$ = LMTD en flujo unidireccional en condiciones de diseño.

$LMTD_{ccd}$ = LMTD en flujo contracorriente en condiciones de diseño.

$LMTD_{uno}$ = LMTD en flujo unidireccional en condiciones de operación.

$LMTD_{cco}$ = LMTD en flujo contracorriente en condiciones de operación.

U_d = Coeficiente global de transferencia de calor en condiciones de diseño.

U_o = Coeficiente global de transferencia de calor en condiciones de operación.

Dp_{fd} = Caída de presión por el lado frío en condiciones de diseño.

Dp_{fo} = Caída de presión por el lado frío en condiciones de operación.

Dp_{cd} = Caída de presión por el lado caliente en condiciones de diseño.

Dp_{co} = Caída de presión por el lado caliente en condiciones de operación.

Q_o = Calor transferido en condiciones de operación.

Q_o^1 = Calor transferido en condiciones de operación calculado por transferencia de calor.

x, y = temperaturas supuestas.

Pe_{cd} = Presión de entrada del fluido caliente en condiciones de diseño.

Pe_{co} = Presión de entrada del fluido caliente en condiciones de operación.

Ps_{co} = Presión de salida del fluido caliente en condiciones de operación.

Pe_{fd} = Presión de entrada del fluido frío en condiciones de diseño.

Pe_{fo} = Presión de entrada del fluido frío en condiciones de operación.

Ps_{fo} = Presión de salida del fluido frío en condiciones de operación.

Cp_{fd} = Poder calorífico del fluido frío a su temperatura media en condiciones de diseño.

Cp_{fo} = Poder calorífico del fluido frío a su temperatura media en condiciones de operación.

Cp_{cd} = Poder calorífico del fluido caliente a su temperatura media en condiciones de diseño.

Cp_{co} = Poder calorífico del fluido caliente a su temperatura media en condiciones de operación.

A = Área total de transferencia de calor.

m_{fd} = Masa de fluido frío en condiciones de diseño.

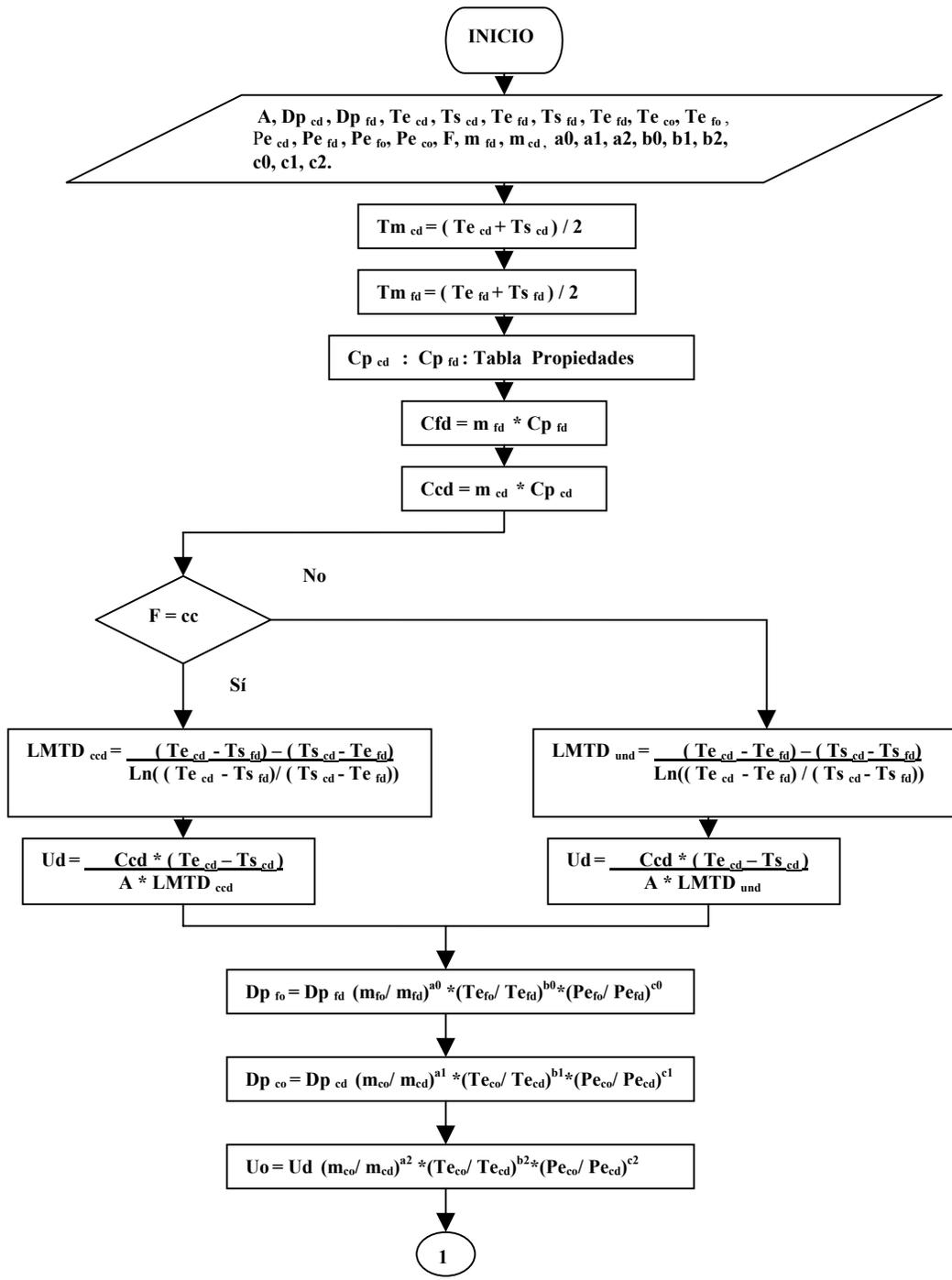
m_{fo} = Masa de fluido frío en condiciones de operación.

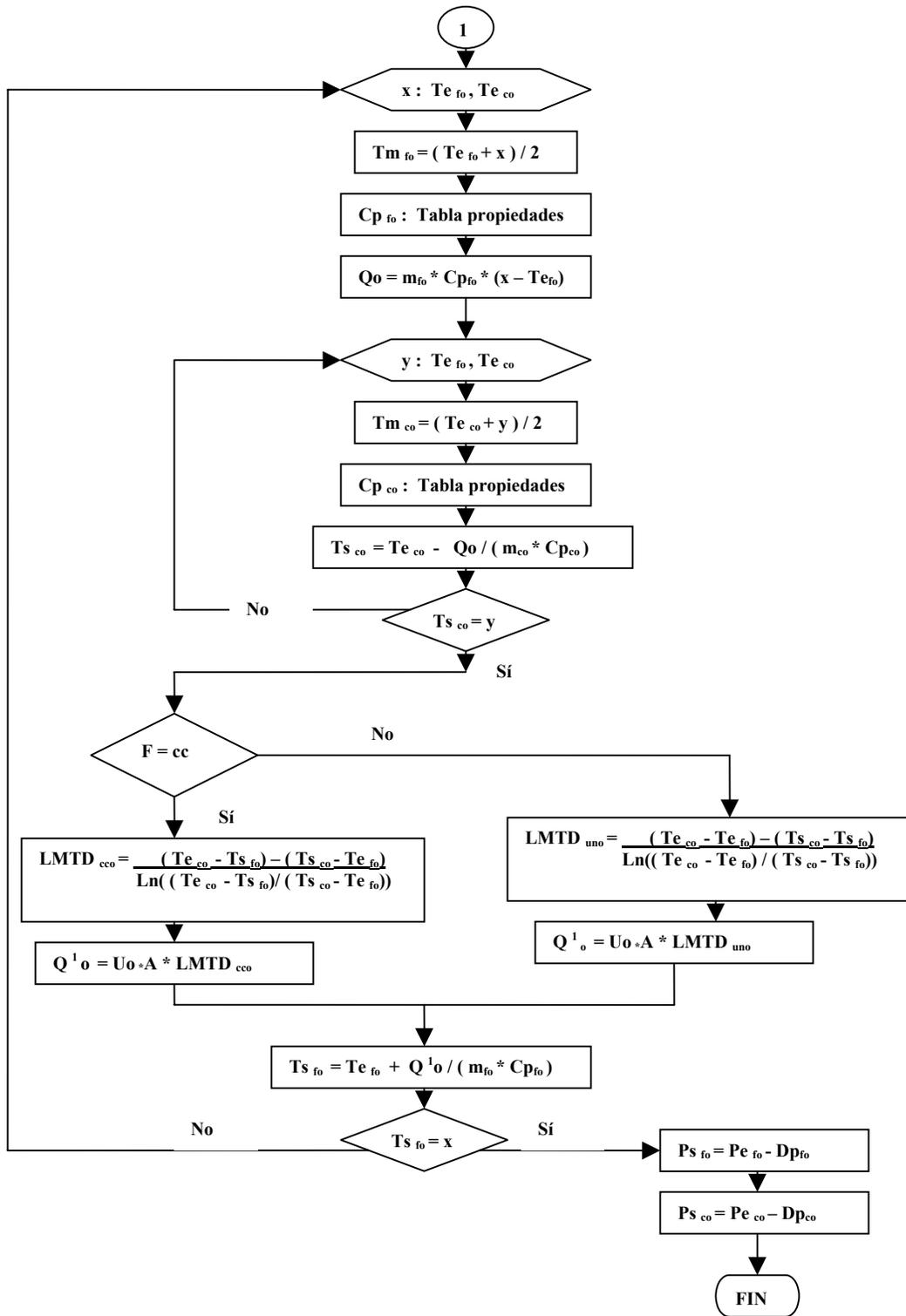
m_{cd} = Masa de fluido caliente en condiciones de diseño.

m_{co} = Masa de fluido caliente en condiciones de operación.

F = Tipo de intercambiador (cc: contracorriente ó un: unidireccional)

¹⁶ *Ibíd.*





Anexo C. Diagrama de flujo del cálculo a carga parcial para la válvula de control

Definiciones.

T = Tipo de condición de la válvula.

TA = Totalmente abierta.

FL = Válvula con flujo lineal.

FE = Válvula con flujo exponencial o equal.

Cv = coeficiente de capacidad de la válvula totalmente abierta.

Cvx = coeficiente de capacidad de la válvula entre abierta.

X = Apertura de la válvula en porcentaje.

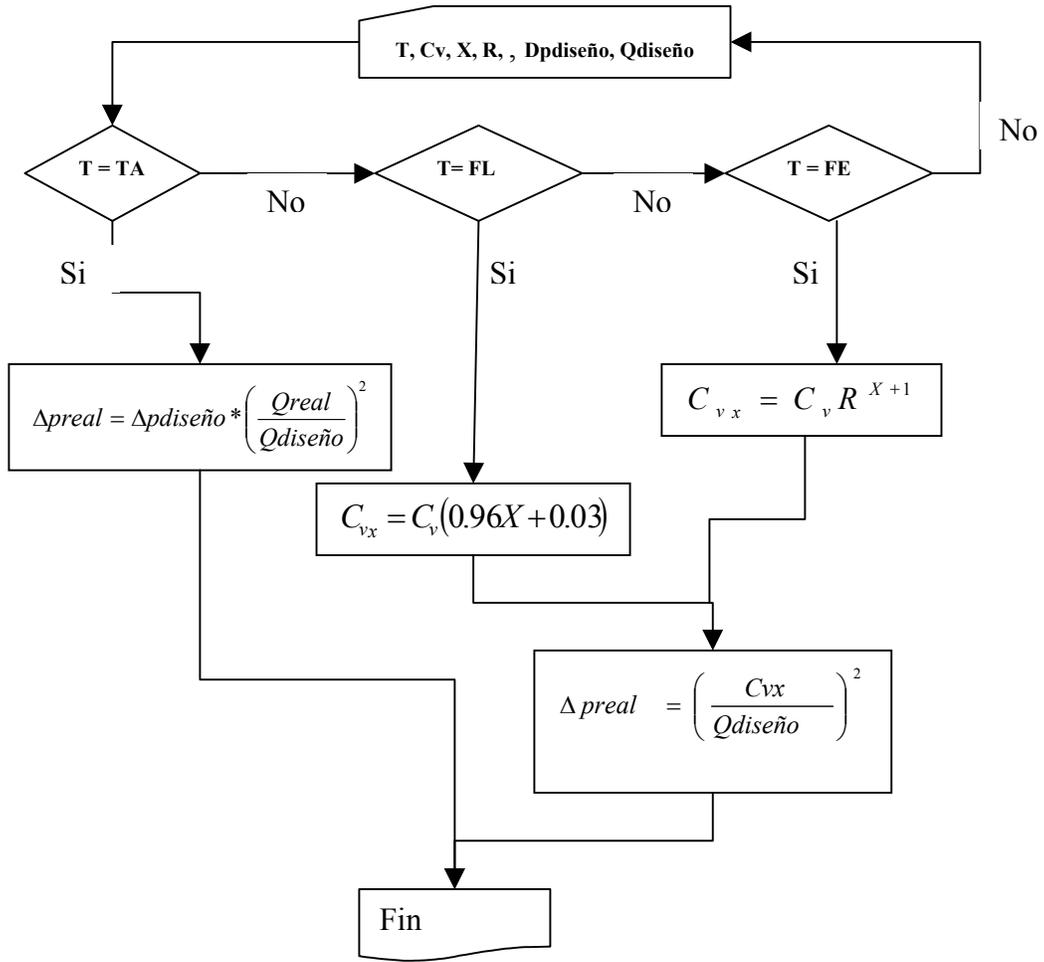
R = Rangeabilidad de la válvula.

Δp_{real} = caída de presión en condiciones de operación.

$Dp_{diseño} = \Delta p_{diseño}$ = caída de presión en condiciones de diseño.

$Q_{diseño} = Q_{diseño}$ = caudal máximo de diseño de la válvula.

Q_{real} = caudal de operación de la válvula



Anexo D. Diagrama de flujo del cálculo a carga parcial para las bombas.

Definiciones.

N_1 = velocidad del eje de la bomba en condiciones de diseño.

N_2 = velocidad del eje de la bomba en condiciones de operación.

Q_1 = Caudal de la bomba en condiciones de diseño.

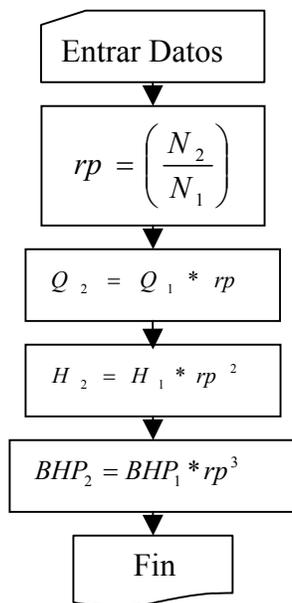
Q_2 = Caudal de la bomba en condiciones de operación.

H_1 = Carga o cabeza de la bomba en condiciones de diseño.

H_2 = Carga o cabeza de la bomba en condiciones de operación.

BHP_1 = Potencia de la bomba en condiciones de diseño.

BHP_2 = Potencia de la bomba en condiciones de operación.



Anexo E. Archivos fuentes del programa Turbina_Carga_Parcial.exe

La implementación de las funciones y de los elementos de programación en C++ (Clases) se realiza en dos archivos, uno con extensión “.h”, en donde se hacen las declaraciones de los datos y funciones necesarias. En otro archivo de extensión “.cpp”, se realizan las definiciones de cada una de las funciones declaradas en el archivo “.h”.

1. Eficiencia_Perdidas.h.

*/*Archivo de declaración de las funciones para calcular la eficiencia y las perdidas de la turbinas a vapor de acuerdo al articulo de la Asme llamado: A method for predicting the performance of steam turbine-generators 16.500 Kw and large realizado por R.C.Spencer, K.C.Cotton y C.N.Cannon*/*

```
float PU(int ent, float X, const double array[]);
float figura2(float X,float Y);
float figura3(float X);
float figura4(float X,float Y);
float figura5(float X);
float figura6(float X,float Y);
float figura7(float X);
float figura8(float X,float Y);
float figura9(float X,float Y);
float figura10(float X,float Y);
float figura11(float X,float Y);
float figura12(int X,float Y,float rp);
float figura13(float X,float Y);
float figura14(float X,float Y);
float figura15(float X);
float figura16_17_18(float Van, float Aan,int rpm);///Interpolacion
en tabla III
```

```

float función_3600rpm(float Van,float Aan);////función auxiliar de
la tabla III
float función_1800rpm(float Van,float Aan);////función auxiliar de
la tabla III
float figura19(int KVA,BOOL convencional,int rpm,BOOL sc_tcdf_3600);
float figura20(int KVA,BOOL convencional,int rpm);
float figura21(float KVA,BOOL convencional,int rpm);
float figura23(int KVA,BOOL convencional,int rpm);

```

2. Eficiencia_Perdidas.cpp.

/*Archivo de definición de las funciones para el cálculo de eficiencia y perdidas de turbinas a vapor de acuerdo al artículo de la Asme llamado: A method for predicting the performance of steam turbine-generators 16.500 Kw and large realizado por R.C.Spencer, K.C.Cotton y C.N.Cannon.*/

```

#include "stdafx.h"
#include "Eficiencia_Perdidas.h" // incluye el archivo anterior
#include <math.h> // incluye la libreria de funciones matematicas
#include <iostream.h> //incluye la libreria de funciones de archivos

/*Primero se define arreglos con los coeficientes para cada figura
del artículo de la Asme.*/

//Coeficientes figura 2
const double f2_A[4]={0,-1.6649986,-22.538964,19.464851};
const double f2_B[4]={0,798267.5,-7540.7,-154269.4};

//Coeficientes figura 3
const double f3_AB[7]={0,2.4462684,-41.90457,-164.13062,-485.99735,
-674.41251,-342.16474};

//Coeficientes figura 4
const double f4_A[4][4]={{0,0,0,0},{-24.899722,63.299521,-
75.518221,37.084109},
{-0.71362812,-23.404163,126.772447,-102.17991},
{-17.632581,-308.12317,594.36987,-269.89024}};

//Coeficiente figura 5
const double f5_A[4]={-1.771,3.475,-3.389,1.445};
//Coeficientes figura 6
const double f6_A[2][2]={{11.151,-63},{-0.50091,2.83}};
//Coeficientes figura 7
const double f7_A[2]={4.37,-0.115};
//Coeficientes figura 8
const double f8_A[2][2]={{-21.8085,21.8085},{0.573908,-0.573908}};

```

```

//Coeficientes figura 9
const double f9_A[2][4]={{-60.75,66.85,29.75,-35.85},{17.5,-20.02,-
0.525,3.045}};
//Coeficientes figura 10
const double f10_A[2][2]={{25.665,-145},{-1.33281,7.53}};

//Coeficientes figura 11
const double f11_A[4][4]={{0,42.676909,-89.391147,9.0376638},
{0,-26.221836,25.549385,8.8283868},{0,4.047955,-1.4725197,-
4.0183332},
{0,-0.14502211,-0.18580363,0.42657518}};

//Coeficientes figura 12
const double f12_A[2][2]={{-5.4,4.395},{0.45,-0.36625}};

//Coeficiente figura 14
const double f14_A[5][6]={{28.232252,-92.390491,-
625.7959,207.2301,70.251642,-22.516388},{-
0.047796308,1.2844571,0.38556961,-0.039652999,-
0.27180357,0.064869467},{-0.69791427*pow(10,-3),-0.17037268*pow(10,-
2),0.86563845*pow(10,-3),-0.5951066*pow(10,-3),0.39705804*pow(10,-
3),-0.73533255*pow(10,-4)},{0.12050837*pow(10,-
5),0.26826382*pow(10,-6),-0.67887771*pow(10,-6),0.52886157*pow(10,-
6),-0.24106229*pow(10,-6),0.37881801*pow(10,-7)},{-
0.50719109*pow(10,-9),0.26393497*pow(10,-9),0.380219111*pow(10,-
10),-0.10149993*pow(10,-9),0.47757232*pow(10,-10),-
0.70989561*pow(10,-11)}};

//Coeficientes figura 15
const double f15_A[3]={{-23.984811,57.86244,3.1849404}};

//Valores de la velocidad anular Tabla III en ft/sec
const int Van[21]=
{128,150,175,200,250,300,350,400,450,500,550,600,650,700,800,900,100
0,1100,1200,1300,1400};

//Area anular TablaIII a rpm 3600 en ft^2
const float Aan_rpm3600[6]={{26.2,32.9,41.1,55.6,66.1,95.3}};

//Perdidas en la TablaIII a rpm de 3600
const double Perdidas_3600[21][6]=
{{25.61,53.65,64.09,99.21,106.7,42.85},{20.64,46.5,56.15,88.9,95.43,
37.8},
{16.1,40.08,47.58,77.88,84.28,32.63},{12.75,34.35,40.55,67.65,74.38,
28.4},
{8.33,24.95,29.75,50.85,56.72,21.6},{5.53,18.68,22.48,38.38,43.00,16
.6},
{4.09,14.23,17.28,29.83,32.82,13.13},{3.73,10.9,13.55,23.15,25.4,10.
55},
{3.95,8.65,10.82,17.8,19.4,9.00},{4.9,7.45,8.85,13.6,14.66,8.13},
{6.57,7.12,8.3,11.2,11.93,7.89},{8.65,7.55,8.5,10.45,10.8,8.40},
{10.95,8.70,9.50,10.75,10.75,9.57},{13.56,10.45,11.25,12.00,11.65,11
.40},

```

```

{19.35,15.45,15.82,16.2,15.2,16.25},{25.62,21.07,21.13,21.17,19.8,21
.97},
{32.00,26.95,26.82,26.42,25.22,28.00},{38.35,33.05,32.82,32.10,31.06
,34.00},
{44.35,38.75,38.5,37.55,36.7,39.80},{49.9,44.15,43.8,42.7,42.05,45.1
8},
{54.7,48.9,48.3,47.35,46.69,49.9}};

//Area anular TablaIII a rpm 1800 en ft^2
const float Aan_rpm1800[5]={66.1,95.3,105.7,123.8,172.4};

//Perdidas en la Tabla III a rpm de 1800
const double Perdidas_1800[21][4]={{42.85,51.63,51.63,75.69},
{37.8,45.5,45.5,65.64},{32.63,38.38,38.38,55.68},{28.40,32.65,32.65,
48.05},
{21.60,23.8,23.8,36.00},{16.6,17.43,17.43,27.80},{13.13,12.91,12.91,
21.45},
{10.55,9.80,9.80,16.3},{9.00,7.97,7.97,12.5},{8.13,7.07,7.07,9.60},
{7.89,7.02,7.02,8.55},{8.40,7.72,7.72,8.47},
{9.57,9.00,9.00,9.05},
{11.4,11.00,11.00,10.38},{16.25,16.25,16.25,14.45},{21.97,21.97,21.9
7,19.56},
{28.00,27.92,27.92,25.07},{34.00,34.00,34.00,30.82},{39.80,39.92,39.
92,36.40},
{45.18,45.25,45.25,41.78},{49.9,49.85,49.85,46.5}};

//Coeficientes figura 19
single casing and TCDF at 3600 rpm
const double f19_A3600[4]={-2.78061,3.69116*pow(10,-
3),6.63261*pow(10,-9),
-6.98336*pow(10,-14)};
const double f19_B3600[2]={105.1185,2.5948815*pow(10,-3)};

//Otros tipos
const double f19_C[2]={220.0,2.82*pow(10,-3)};

//figura 19 continuacion
const double f19_cont[3]={570.0,-3.04*pow(10,-3),3.8*pow(10,-9)};

//Coeficiente figura 20
//Cooled Convencional
const double f20_Convencional[4]={-0.34293488,0.89126466,
-0.10877851,0.36908686*pow(10,-2)};
//Cooled Conductor
const double f20_Conductor[4]={0.18479050*pow(10,1),0.56681377*pow(10,-1),-
0.26652558*pow(10,-1),0.10759713*pow(10,-2)};
const double f20_B[2]={1.0,-0.49564581*pow(10,-1)};

//Coeficientes figura 21
//cooled convencional a 3600 y a 1800 rpm
const double f21_AConvencional[4]={-0.20525271*pow(10,2),
-0.35019203*pow(10,2),0.25514027*pow(10,2),-0.25182400*pow(10,2)};
const double f21_BConvencional[2]={1.0,-0.56218495*pow(10,2)};

```

```

//cooled conductor a 3600 rpm
const double f21_AConductor3600[2]={-0.17535943*pow(10,2)
,-0.15515139*pow(10,2)};
const double f21_BConductor3600[3]={1.0,
-0.65338483*pow(10,2),0.31286714*pow(10,2)};

//cooled conductor a 1800 rpm
const double f21_AConductor1800[5]={-0.22727744,
-0.24939976*pow(10,2),0.34283283*pow(10,2),
-0.35808901*pow(10,2),0.90544169*pow(10,1)};
const double f21_BConductor1800[2]={1.0,-0.18638959*pow(10,2)};

//Coeficientes figura 23
//Cooled convencional a 3600
const double f23_AConvencional3600[4]={0.10949623*pow(10,2),
0.12755560*pow(10,-3),0.71452351*pow(10,-8),-0.58333495*pow(10,-
14)};
const double f23_BConvencional3600[2]={1.0,0.56299996*pow(10,-5)};
//cooled convencional a 1800
const double f23_AConvencional1800[2]={0.33807252*pow(10,2),
0.25207663*pow(10,-3)};
//cooled conductor a 3600
const double f23_AConductor3600[2]={0.12*pow(10,3),0.96*pow(10,-4)};
//cooled conductor a 1800
const double f23_AConductor1800[2]={0.40*pow(10,2),0.96*pow(10,-4)};

float PU(int ent,float x,const double array[])
{
    /// Código para el cálculo de polinomios univariable
    // para los cálculos de este //procedimiento de la Asme
    float Z=0.0;
    for(int i=0;i<ent;i++)
    {
        float c=array[i]*pow(x,i);
        Z+=c;
    }
    return Z;
}

float figura2( float X, float Y)
{
    //función que hace los cálculos de la figura 2 del
    ///procedimiento de la Asme
    float Z=0.0;
    if(X>2.0)
    {
        AfxMessageBox("presion de gobierno muy grande\n"
        "en comparacion a la presion de diseño\n"
        "el resultado sera falso");
        return Z;
    }
    float p=0.625-X;
    for(int i=1;i<4;i++)
    {
        Z+=(f2_A[i]+(f2_B[i]/pow(Y,i)))*pow(p,i);
    }
}

```

```

        return Z;
    }

float figura3(float X)
{    //función que hace los cálculos de la figura 3 del
////procedimiento de la Asme
    float Z=0.0;
    float L=log10(X);
    Z=PU(7,L,f3_AB);
    return Z;
}

float figura4(float rp,float Q)
{    //función que hace los cálculos de la figura 4 del
////procedimiento de la Asme
    float Z=0.0;
    if(rp<0.5 )
    {    AfxMessageBox(" la presion de la etapa de gobierno es\n"
        " mucho mayor que la entrada de diseño\n"
        "el resultado sera falso");
        return Z;
    }
    float x=0.625-(1.0/rp);
    for(int i=1;i<4;i++)
    {
        for(int j=0;j<4;j++)
        {
            Z+=(f4_A[i][j])*(pow(x,i))*(pow(Q,j));
        }
    }
    return Z;
}

float figura5(float X)
{    //función que hace los cálculos de la figura 5 del
////procedimiento de la Asme
    float Z=0.0;
    Z=PU(4,X,f5_A);
    return Z;
}

float figura6(float X, float Y)
{    //función que hace los cálculos de la figura 6 del
////procedimiento de la Asme
    float Z=0.0;
    float c=log(Y);
    for(int i=0;i<2;i++)
    {
        for(int j=0;j<2;j++)
        {
            Z+=(f6_A[i][j])*(pow(X,j))*(pow(c,i));
        }
    }
    return Z;
}

```

```

}

float figura7(float X)
{ //función que hace los cálculos de la figura 7 del
///procedimiento de la Asme
float Z=0.0;
Z=PU(2,X,f7_A);
return Z;
}

float figura8(float X,float TFR)
{ //función que hace los cálculos de la figura 8 del
///procedimiento de la Asme
float Z=0.0;
for(int i=0;i<2;i++)
{
for(int j=0;j<2;j++)
{
Z+=(f8_A[i][j])*(pow(X,i))*(pow(TFR,j));
}
}
return Z;
}

float figura9(float TFR,float Y)
{ //función que hace los cálculos de la figura 9 del
///procedimiento de la Asme
float Z=0.0;
float y=log(Y);
for(int i=0;i<2;i++)
{
for(int j=0;j<4;j++)
{
Z+=(f9_A[i][j])*(pow(TFR,j))*(pow(y,i));
}
}
return Z;
}

float figura10(float rp, float Q)
{ //función que hace los cálculos de la figura 10 del
///procedimiento de la Asme
float Z=0.0;
float q=log(Q);
for(int i=0;i<2;i++)
{
for(int j=0;j<2;j++)
{
Z+=(f10_A[i][j])*(pow(rp,j))*(pow(q,i));
}
}
return Z;
}

```

```

float figura11(float rp, float Q)
{ //función que hace los cálculos de la figura 11 del
//procedimiento de la Asme
float Z=0.0;
float q=1.0-Q;
for(int i=0;i<4;i++)
{
for(int j=0;j<4;j++)
{
Z+=(f11_A[i][j])*(pow(rp,i))*(pow(q,j));
}
}
return Z;
}

float figura12(int nv,float TFR,float rp)
{ //función que hace los cálculos de la figura 12 del
//procedimiento de la Asme
float Z=0.0;
float r=rp/0.26;
float a=0.0;
for(int i=0;i<2;i++)
{
for(int j=0;j<2;j++)
{
a+=(f12_A[i][j])*(pow(TFR,j))*(pow(nv,i));
}
}
if(rp>=0.24 && rp <=0.27)
{
Z=a;
return Z;
}
else
{
Z=r*a;
return Z;
}
}

float figura13(float rp,float Qv)
{ //función que hace los cálculos de la figura 13 del
//procedimiento de la Asme
float Z=0.0;
float log_rp1=log(rp-0.3);
float log_rp2=log(rp-0.8);
float A=90.799+(0.7474*log_rp1)-(0.5454/log_rp1);
float B=-505000+(77568*log_rp2)-(1262500/log_rp2);
Z=A+(B/Qv);
return Z;
}

```

```

float figura14(float P, float H)
{ //función que hace los cálculos de la figura 14 del
//procedimiento de la Asme
float Z=0.0;
if(P<0 || H<0)
{AfxMessageBox(" la presion real no es valida\n"
                " ó la temperatura real de entrada\n"
                "el resultado sera falso");
}
float p=log10(P);
for(int i=0;i<5;i++)
{
    for(int j=0;j<6;j++)
    {
        Z+=(f14_A[i][j])*(pow(p,j))*(pow(H,i));
    }
}
return Z;
}

float figura15(float P)
{ //función que hace los cálculos de la figura 15 del
//procedimiento de la Asme
// P es presion (debe estar en inHg abs)
float Z=0.0;
float p=log(P);
Z=PU(3,p,f15_A);
return Z;
}

float figura16_17_18(float Van, float Aan,int rpm)
{ //función que hace los cálculos de las figuras 16,17 y 18 del
//procedimiento de la Asme
float Z=0.0;
if(Van<=0 || Aan<=0)
{AfxMessageBox("Velocidad anular o los valores de area
anular,\n"
                "son igual o menor que cero, el resultado sera
incorrecto");
return Z;
}
switch(rpm)
{
case 3600:
Z=función_3600rpm(Van,Aan);
break;
case 1800:
Z=función_1800rpm(Van,Aan);
break;
default:
Z=0.0;
AfxMessageBox("No has definido las rpm\n"
                "el resultado sera incorrecto");
break;
}
}

```

```

    }
    return Z;
}

float función_3600rpm(float Vel,float Aan)
{ //función que hace los cálculos de las perdidas para una
//velocidad de eje de 3600 rpm de la tabla III del
//procedimiento de la Asme por medio de una interpolación
    float Z1=0.0;
    //// i no puede ser mayor de 20 y j no mayor de 4 ////
    int i=0;
    int j=0;
    while ( Vel >Van[j+1])
    {
        j++;
    }
    while (Aan >Aan_rpm3600[i+1] )
    {
        i++;
    }
    double N1;
    if(Vel==Van[j])
    {
        N1=Perdidas_3600[j][i];
    }
    else
    {
        N1=Perdidas_3600[j+1][i];
    }
    double N2=Perdidas_3600[j+1][i+1];
    double N3=Perdidas_3600[j][i+1];
    double N4=Perdidas_3600[j][i];
    double a=Aan_rpm3600[i+1]-Aan_rpm3600[i];
    double b=Van[j+1]-Van[j];
    double Xc=(Aan_rpm3600[i+1]+Aan_rpm3600[i])/2;
    double Yc=(Van[j+1]+Van[j])/2;
    double Z=(Aan-Xc)/a;
    double N=(Vel-Yc)/b;
    double S1=(1-(Z/a))*(1-(N/b));
    double S2=(Z/a)*(1-(N/b));
    double S3=(Z/a)*(N/b);
    double S4=(N/b)*(1-(Z/a));
    Z1=(N1*S1)+(N2*S2)+(N3*S3)+(N4*S4);
    return Z1;
}

float función_1800rpm(float Vel,float Aan)
{ //función que hace los cálculos de las pérdidas para una
//velocidad de eje de 1800 rpm de la tabla III del
//procedimiento de la Asme por medio de una interpolación
    float Z1=0.0;
    //// i no puede ser mayor de 20 y j no mayor de 4 ////
    int i=0;
    int j=0;
    while ( Vel >Van[j+1])

```

```

        {
            j++;
        }
    while (Aan >Aan_rpm1800[i+1] )
    {
        i++;
    }
    double N1;
    if (Vel==Van[j])
    {
        N1=Perdidas_1800[j][i];
    }
    else
    {
        N1=Perdidas_1800[j+1][i];
    }
    double N2=Perdidas_1800[j+1][i+1];
    double N3=Perdidas_1800[j][i+1];
    double N4=Perdidas_1800[j][i];
    double a=Aan_rpm1800[i+1]-Aan_rpm1800[i];
    double b=Van[j+1]-Van[j];
    double Xc=(Aan_rpm1800[i+1]+Aan_rpm1800[i])/2;
    double Yc=(Van[j+1]+Van[j])/2;
    double Z=(Aan-Xc)/a;
    double N=(Vel-Yc)/b;
    double S1=(1-(Z/a))*(1-(N/b));
    double S2=(Z/a)*(1-(N/b));
    double S3=(Z/a)*(N/b);
    double S4=(N/b)*(1-(Z/a));
    Z1=(N1*S1)+(N2*S2)+(N3*S3)+(N4*S4);
    return Z1;
}

float figura19(int KVA,BOOL convencional,int rpm,BOOL sc_tcdf_3600)
{
    //función que hace los cálculos de la figura 19 del
    //procedimiento de la Asme
    float Z=0.0;
    int kva=0.0;
    if(KVA<0)
    {AfxMessageBox(" entre un numero positivo en KVA,\n"
        "el resultado sera falso");
        return Z;
    }
    if(convencional)
    {
        kva=KVA*1.1;
    }
    else
    {
        kva=KVA;
    }
    float Dml=0.0;
    if(kva>=500000)
    {
        Dml=PU(3,kva,f19_cont);
    }
}

```

```

    }
else
{
    Dml=0.0;
}
float ML=0.0;
if(sc_tcdf_3600)
{
    switch(rpm)
    {
        case 3600:
            if( kva>=22000 && kva<110000)
            {
                ML=PU(4,kva,f19_A3600);
            }
            else if(kva>=110000)
            {
                ML=PU(2,kva,f19_B3600);
            }
            else{
                ML=PU(2,kva,f19_C);
            }
            break;
        case 1800:
            ML=PU(2,kva,f19_C);
            break;
        default:
            AfxMessageBox("Defina solo rpm 3600 ó
1800");
            return Z;
            break;
    }
}
else
{
    ML=PU(2,kva,f19_C);
}
Z=ML+Dml;
return Z;
}

float figura20(int KVA,BOOL convencional,int rpm)
{
    //función que hace los cálculos de la figura 20 del
    //procedimiento de la Asme
    float Z=0.0;
    if(KVA<0)
    {AfxMessageBox(" entre un numero positivo en KVA,\n"
        "el resultado sera falso");
        return Z;
    }
    float kva=log(KVA);
    switch(rpm)
    {
        case 3600:

```

```

        if(convencional)
        {
            Z=PU(4,kva,f20_Convencional);
        }
        else
        { ////// cooled conductor //////////
            float a=PU(4,kva,f20_Conductor);
            float b=PU(2,kva,f20_B);
            Z=a/b;
        }
        break;
case 1800:
    if(convencional)
    {
        Z=-0.095+PU(4,kva,f20_Convencional);
    }
    else
    { ////// cooled conductor //////////
        float a=PU(4,kva,f20_Conductor);
        float b=PU(2,kva,f20_B);
        Z=-0.10+a/b;
    }
    break;
default:
    AfxMessageBox("El resultado sera incorrecto\n"
        "Defina rpm: 3600 ó 1800 ");
    Z=0.0;
    break;
}
return Z;
}

float figura21(float PKVA,BOOL convencional,int rpm)
{ //función que hace los cálculos de la figura 21 del
//procedimiento de la Asme
float Z=0.0;
float a,b;
a=b=0.0;
if(PKVA<=0.0)
{AfxMessageBox(" entre un numero positivo en KVA,\n"
    "el resultado sera falso");
    return Z;
}
if(PKVA>1.0)
{AfxMessageBox(" entre un KWs del generador menor,\n"
    "o un mayor factor de potencia\n" "o un mayor KVA\n"
    "el resultado sera falso");
    return Z;
}
switch(rpm)
{
case 3600:
    if(convencional)
    {
        a=PU(4,PKVA,f21_AConvencional);
    }
}
}

```

```

        b=PU(2,PKVA,f21_BConvencional);
        Z=a/b;
    }
    else
    { ////// cooled conductor //////////
        a=PU(2,PKVA,f21_AConductor3600);
        b=PU(3,PKVA,f21_BConductor3600);
        Z=a/b;
    }
    break;
case 1800:
    if(convencional)
    {
        a=PU(4,PKVA,f21_AConvencional);
        b=PU(2,PKVA,f21_BConvencional);
        Z=a/b;
    }
    else
    { ////// cooled conductor //////////
        a=PU(5,PKVA,f21_AConductor1800);
        b=PU(2,PKVA,f21_BConductor1800);
        Z=a/b;
    }
    break;
default:
    AfxMessageBox("El resultado sera incorrecto\n"
        "Defina rpm: 3600 ó 1800 ");
    Z=0.0;
    break;
}
return Z;
}

float figura23(int KVA,BOOL convencional,int rpm)
{ //función que hace los cálculos de la figura 23 del
//procedimiento de la Asme
float Z=0.0;
if(KVA<0)
{AfxMessageBox(" entre un numero positivo en KVA,\n"
    "el resultado sera falso");
return Z;
}
switch(rpm)
{
case 3600:
    if(convencional)
    {
        float a=PU(4,KVA,f23_AConvencional3600);
        float b=PU(2,KVA,f23_BConvencional3600);
        Z=a/b;
    }
    else
    { ////// cooled conductor //////////
        Z=PU(2,KVA,f23_AConductor3600);
    }
    break;
}
}

```

```

case 1800:
    if(convencional)
    {
        Z=PU(2,KVA,f23_AConvencional1800);
    }
    else
    { ////// cooled conductor //////////
        Z=PU(2,KVA,f23_AConductor1800);
    }
    break;
default:
    AfxMessageBox("El resultado sera incorrecto\n"
        "Defina rpm: 3600 ó 1800 ");
    Z=0.0;
    break;
}
return Z;      }

```

3. EficienciaTurbina.h

/*Archivo de declaración de la clase CEficienciaTurbina, la cual se usa en el programa para calcular la eficiencia y perdidas de turbinas a vapor de acuerdo al articulo de la Asme llamado: A method for predicting the performance of steam turbine-generators 16.500 Kw and large realizado por R.C.Spencer, K.C.Cotton y C.N.Cannon.*/

```

Interface for the CEficienciaTurbina class.
#if
!defined(AFX_CEficienciaTurbina_H__398DE4B0_9E78_468C_9BA7_7C14492ADBDC__INCLUDED_)
#define
AFX_CEficienciaTurbina_H__398DE4B0_9E78_468C_9BA7_7C14492ADBDC__INCLUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 100
// se define una estructura para recoger los datos termodinamicos de
//los flujos
typedef struct{
    float presion;
    float temp;
    float flujomasico;
    int sistema_unidades;
}datostermo;

// se define una estructura para recoger los datos generales de la
//turbina
typedef struct{
    int tipoturbina;

```

```

        int NumFlujos;
        int NumValvulas;
}datosgenerales;

// se define una estructura para recoger los datos de la etapa de
//gobierno
typedef struct{
    float PDg;
    float presion_gobierno;
}datosgobierno;

// se declara la clase para el cálculo de la eficiencia de la
//turbina

class CEficienciaTurbina
{
public:
    float calcular_presion_salida_real();//calcula la presion de
//salida en condiciones de operación según la ecuacion de Stodola

    float calcular_eficiencia();//calcula la eficiencia de la
//turbina de acuerdo al tipo de turbina seleccionada por el usuario
//según //procedimiento de la Asme

    //////////////// Funciones de entrada de datos ////////////////
    int entrar_presion_1800rpm(float p1800);
    int datosgobiernoentrada(datosgobierno dgob);
    int entrar_datosgenerales( datosgenerales dg);
    int datosdisenosalida(float psd);
    int datosrealentrada(datostermo der);
    int datosdisenoentrada(datostermo ded);
    CEficienciaTurbina();//constructor
    virtual ~CEficienciaTurbina();//destructor
private:
    //define funciones y variables privadas
    int comprobar_entrada_datos();//comprueba la entrada de datos
    //////////////// Funciones de trabajo de la clase ////////////////

    float Calcular_Entalpia(float p, float s);//calcula la entalpia
//del agua entrando presion y entropia
    float Tabla_PT(float p, float t, int s);// se calcula entropia
//entrando presión y temperatura pero se elige entre la calculada o
//la entropia que entra s

    void calcular_volumen();//calcula volumen especifico

    //funciones que calculan la eficiencia de la turbina de acuerdo
//a su tipo segun Spencer , Cotton y Cannon: método de la Asme
    float EficienciaTurbinaTipoF();
    float EficienciaTurbinaTipoE();
    float EficienciaTurbinaTipoD();
    float EficienciaTurbinaTipoC();
    float EficienciaTurbinaTipoB();
    float EficienciaTurbinaTipoA();

```

```

float EficienciaTurbinaTipoG();
float fY1();//calcula entalpía haciendo una elección según
/// Diagrama de flujo Cliope realizado por Adriana
float fEficiencia(float d,float correcion);// primera
//corrección de la eficiencia base según Spencer , Cotton y Cannon:
//metodo de la Asme
float correcion_flujo_volumetrico(int tipo,int N,float md);
float tfr();//calcula un parametro que es la relacion entre
//flujo masico real y de diseño
float relp();// calcula otro parametro requerido por los
//cálculos de la eficiencia es la relacion entre la presion de
//salida de diseño y la presion de entrada de diseño variables de
//trabajo de la clase
float m_volumen_real;
int m_TipoTurbina;
float m_Diametro_etapa_gobierno;
float m_temp_entrada_real;
float m_temp_entrada_diseno;
float m_presion_salida_diseno;
float m_flujo_real;
float m_presion_entrada_real;
float m_presion_entrada_diseno;
float m_presion_etapa_gobierno;
float m_volumen_diseno;
float m_flujo_diseno;
int m_flujos_entrada;
float m_presion_seccion_1800rpm;
int m_numero_valvulas;
BOOL a,b,c,d,e,f;
};

#endif
!defined(AFX_EficienciaTurbina_H__398DE4B0_9E78_468C_9BA7_7C14492AD
BDC__INCLUDED_)

```

4. EficienciaTurbina.cpp

```

/* Archivo de definición de la clase CEficienciaTurbina, la cual se usa en el
programa para calcular la eficiencia y perdidas de turbinas a vapor de acuerdo al
articulo de la Asme llamado: A method for predicting the performance of steam
turbine-generators 16.500 Kw and large realizado por R.C.Spencer, K.C.Cotton y
C.N.Cannon.*/

```

```

implementation of the CEficienciaTurbina class.
//

#include "stdafx.h"
#include "EficienciaTurbina.h"
#include "Eficiencia_Perdidas.h"
#include "Propiedad.h"
#include <math.h>

```

```

#ifdef _DEBUG
#undef THIS_FILE
static char THIS_FILE[]=__FILE__;
#define new DEBUG_NEW
#endif

// se define constantes con nombres convenientes
#define TurbinaTipoA 0
#define TurbinaTipoB 1
#define TurbinaTipoC 2
#define TurbinaTipoD 3
#define TurbinaTipoE 4
#define TurbinaTipoF 5
#define TurbinaTipoG 6
#define s1 0
#define h1 1
// se define un arreglo con los valores de las eficiencias base para
//cada tipo de turbina
const float
EficienciaBase[7]={89.48,87.00,84.00,0.0,91.93,91.93,92.95};
const int
Constante[7]={430260,1005200,1350000,0,1270000,1270000,1270000};

// Construction/Destruction

CEficienciaTurbina::CEficienciaTurbina()
{
    //constructor general de la clase
    m_TipoTurbina=0;
    m_temp_entrada_real=0.0;
    m_temp_entrada_diseno=0.0;
    m_presion_salida_diseno=0.0;
    m_flujo_real=0.0;
    m_presion_entrada_real=0.0;
    m_presion_entrada_diseno=0.0;
    m_presion_etapa_gobierno=0.0;
    m_volumen_diseno=0.0;
    m_volumen_real=0.0;
    m_flujo_diseno=0.0;
    m_flujos_entrada=0;
    m_Diametro_etapa_gobierno=0.0;
    m_numero_valvulas=0.0;
    m_presion_seccion_1800rpm=0.0;
    a=b=c=d=e=f=0;
}

CEficienciaTurbina::~~CEficienciaTurbina()
{
    //destructor
}

int CEficienciaTurbina::datosdisenoentrada(datostermo ded)
{
    //función que entra los datos de diseño del flujo que
    // entra a la turbina
    m_temp_entrada_diseno=ded.temp;
    m_presion_entrada_diseno=ded.presion;
    m_flujo_diseno=ded.flujomasico;
}

```

```

        a=1;
        return 1;
    }

int CEficienciaTurbina::datosrealentrada(datostermo der)
{
    //función que entra los datos de operación del flujo que
    // entra a la turbina
    m_temp_entrada_real=der.temp;
    m_presion_entrada_real=der.presion;
    m_flujo_real=der.flujomasico;
    b=1;
    return 1;
}

int CEficienciaTurbina::datosdisenosalida(float psd)
{
    //función que entra los datos de diseño del flujo que
    // sale de la turbina
    // solo es necesario la presion de diseño del flujo que sale
    m_presion_salida_diseno=psd;
    c=1;
    return 1;
}

int CEficienciaTurbina::entrar_datosgenerales(datosgenerales dg)
{
    //función que entra los datos de generales de la turbina
    // el tipo de turbina, el numero de flujos paralelos de
//entrada
    // el número de valvulas de control
    m_TipoTurbina=dg.tipoturbina;
    m_flujos_entrada=dg.NumFlujos;
    m_numero_valvulas=dg.NumValvulas;
    d=1;
    return 1;
}

int CEficienciaTurbina::datosgobiernoentrada(datosgobierno dgob)
{
    //función que entra los datos de la etapa de gobierno
    // de la turbina que tenga tal etapa
    // como la presion en esa etapa y el diametro picth
    m_presion_etapa_gobierno=dgob.presion_gobierno;
    m_Diametro_etapa_gobierno=dgob.PDg;
    e=1;
    return 1;
}

int CEficienciaTurbina::entrar_presion_1800rpm(float p1800)
{
    //función que entra la presion transversal
    //de una turbina tipo F de baja presion a 1800 rpm
    m_presion_seccion_1800rpm=p1800;
    f=1;
    return 1;
}

void CEficienciaTurbina::calcular_volumen()
{
    //función que calcula el volumen especifico

```

```

//tanto en condiciones de diseño como en
//condiciones de operacion
CPropiedad prop;
prop.unidades=inglesas;
prop.presion=m_presion_entrada_diseno;
prop.temp=m_temp_entrada_diseno;
prop.tipo_cal=_PT;
prop.calcular();
m_volumen_diseno=prop.volumen;

prop.presion=m_presion_entrada_real;
prop.temp=m_temp_entrada_real;
prop.tipo_cal=_PT;
prop.calcular();
m_volumen_real=prop.volumen;
}

float CEficienciaTurbina::relp()
{
//calcula la relacion que existe entre
//la presion de diseño del flujo de salida
// y la presion de diseño del flujo de entrada
// de la turbina
float Z=0.0;
Z=m_presion_salida_diseno/m_presion_entrada_diseno;
return Z;
}

float CEficienciaTurbina::tfr()
{
//calcula la relacion que existe entre
//el flujo masico de operación del flujo de entrada
// y el flujo masico de diseño del flujo de entrada
// de la turbina
float Z=0.0;
if(m_flujo_real>m_flujo_diseno)
{AfxMessageBox(" el flujo real debe ser menor ó\n"
" igual al de diseño\n"
"El resultado sera falso");
return Z;
}
Z=m_flujo_real/m_flujo_diseno;
return Z;
}

float CEficienciaTurbina::correccion_flujo_volumetrico(int tipo,int
N, float md)
{
//función que calcula la primera correccion
//de la eficiencia base de acuerdo al tipo de turbina
//ver tabla I del articulo de la Asme llamado //////////////////////////////////
///// A method for predicting the performance of steam turbine-
//generators 16.500 Kw and large realizado por R.C.Spencer,
//K.C.Cotton y C.N.Cannon
float Z=0.0;
Z=- (Constante[tipo]*N)/(md*m_volumen_diseno);
return Z;
}

```

```

}

float CEficienciaTurbina::fEficiencia(float d, float correccion)
{
    //función que hace la correccion de la eficiencia una vez
    //obtenida la correccion de la misma, se repite para todas
    //las correcciones que se le hagan a la eficiencia de la
//turbina
    float Z=0.0;
    Z=d*(1.0+(correccion/100));
    return Z;
}

float CEficienciaTurbina::fY1()
{
    //Busqueda y eleccion de una entropia de acuerdo
    // a unas condiciones expresadas en el articulo de la Asme
//llamado A method for predicting the performance of steam turbine-
//generators 16.500 Kw and large realizado por R.C.Spencer,
//K.C.Cotton y C.N.Cannon para mejor comprension ver diagrma de
//flujo del proyecto cliope realizado por Adriana en el
//procedimiento de la Asme del cálculo de la eficiencia de una
//turbina tipo A para el cálculo de una variable llamada Y1 para
//entrar en la correccion de la eficiencia en la figura 14
    float Z=0.0;
    CPropiedad Busqueda;
    Busqueda.unidades=inglesas;
    Busqueda.presion= m_presion_entrada_real;
    Busqueda.temp= m_temp_entrada_real;
    Busqueda.tipo_cal=PT;
    Busqueda.calcular();
    float S= Busqueda.entropia;
    float H= Busqueda.entalpia;
    float X1,Y1;
    X1=m_presion_entrada_real;
    if( S > 2.0041 )
    {
        float Ht= 1154+(80*X1)+(88*pow(X1,2));
        if( H< Ht)
            Y1= H;
        Else
            Y1= Ht;
    }
    else
    {
        Y1=H;
    }
    Z=Y1;
    return Z;
}

float CEficienciaTurbina::EficienciaTurbinaTipoA()
{
    //función que calcula la eficiencia de una turbina tipo A
    //en condiciones de operación diferentes a la de diseño
    ///definicon de variables necesarias
    float Z=0.0;

```

```

        float corr[6],Eficiencia[6];
        calcular_volumen();//se calcula el volumen especifico
        //a condicones de entrada de diseño y de operación
        corr[0]=correccion_flujo_volumetrico(TurbinaTipoA,m_flujos_entrada,m_flujo_diseño);//se calcula la corrección por parte del flujo
        //volumétrico
        Eficiencia[0]=fEficiencia(EficienciaBase[TurbinaTipoA],corr[0]);
        //se hace la correccion a al eficidad base
        float Xo=(m_presion_etapa_gobierno/m_presion_entrada_diseño);
        float Yo=m_flujo_diseño*m_volumen_diseño;
        corr[1]=figura2(Xo,Yo);//se calcula la correccion de acuerdo a
        //la figura 2 del procedimiento de la Asme ver archivo
        //Eficiencia_Perdidas.cpp
        Eficiencia[1]=fEficiencia(Eficiencia[0],corr[1]);//se corrige
        //la eficiencia
        float X1,Y1;
        X1=m_presion_entrada_real;
        Y1=fY1();
        corr[2]=figura14(X1,Y1);//se calcula la correccion de acuerdo a
        //la figura 14 ver archivo Eficiencia_Perdidas.cpp
        Eficiencia[2]=fEficiencia(Eficiencia[1],corr[2]);//se corrige la
        //eficiencia
        float X2=tfr();
        corr[3]=figura4(1.0/Xo,X2);//se calcula la correccion de acuerdo
        //a la figura 4 ver archivo Eficiencia_Perdidas.cpp
        Eficiencia[3]=fEficiencia(Eficiencia[2],corr[3]);//se corrige la
        //eficiencia
        corr[4]=figura3(X2); //se calcula correccion de acuerdo a la
        //figura 3 ver archivo Eficiencia_Perdidas.cpp
        Eficiencia[4]=fEficiencia(Eficiencia[3],corr[4]); //se corrige
        //la eficiencia
        corr[5]=figura5(X2); //se calcula la correccion de acuerdo a la
        //figura 5 ver archivo Eficiencia_Perdidas.cpp
        Eficiencia[5]=fEficiencia(Eficiencia[4],corr[5]); //se corrige
        //la eficiencia
        Z=Eficiencia[5]; //se devuelve la eficiencia corregida
        return Z;
    }

float CEficienciaTurbina::EficienciaTurbinaTipoB()
{
    //función que calcula la eficiencia de una turbina tipo B
    //en condicones de operacion diferentes a la de diseño
    //se realiza de la misma forma que la anterior pero con
    //diferentes figuras para calcular las correcciones de la
    //eficiencia
    float Z=0.0;
    float corr[6],Eficiencia[6];
    calcular_volumen();
    corr[0]=correccion_flujo_volumetrico(TurbinaTipoB,m_flujos_entrada,m_flujo_diseño);
    Eficiencia[0]=fEficiencia(EficienciaBase[TurbinaTipoB],corr[0]);
    float Xo=m_Diametro_etapa_gobierno;
    corr[1]=figura7(Xo); //ver archivo Eficiencia_Perdidas.cpp
    Eficiencia[1]=fEficiencia(Eficiencia[0],corr[1]);

```

```

float X1=relp();
float Y1=m_flujo_diseno*m_volumen_diseno;
corr[2]=figura6(X1,Y1); //ver archivo Eficiencia_Perdidas.cpp
Eficiencia[2]=fEficiencia(Eficiencia[1],corr[2]);
float X2=tfrc();
corr[3]=figura8(Xo,X2); //ver archivo Eficiencia_Perdidas.cpp
Eficiencia[3]=fEficiencia(Eficiencia[2],corr[3]);
float X3=1.0/X1;
corr[4]=figura9(X2,X3); //ver archivo Eficiencia_Perdidas.cpp
Eficiencia[4]=fEficiencia(Eficiencia[3],corr[4]);
int NV=m_numero_valvulas;
corr[5]=figura12(NV,X2,X1); //ver archivo Eficiencia_Perdidas.cpp
Eficiencia[5]=fEficiencia(Eficiencia[4],corr[5]);
Z=Eficiencia[5];
return Z;
}

float CEficienciaTurbina::EficienciaTurbinaTipoC()
{
//función que calcula la eficiencia de una turbina tipo C
//en condiciones de operacion diferentes a la de diseño
//se realiza de la misma forma que la anterior pero con
//diferentes figuras para calcular las correcciones de la
//eficiencia
float Z=0.0;
float corr[4],Eficiencia[4];
calcular_volumen();
corr[0]=correccion_flujo_volumetrico(TurbinaTipoC,m_flujos_entra
da,m_flujo_diseno);
Eficiencia[0]=fEficiencia(EficienciaBase[TurbinaTipoC],corr[0])
;
float Xo=relp();
float Yo=m_flujo_diseno*m_volumen_diseno;
corr[1]=figura10(Xo,Yo); //ver archivo Eficiencia_Perdidas.cpp
Eficiencia[1]=fEficiencia(Eficiencia[0],corr[1]);
float X1=1.0/Xo;
float Y1=tfrc();
corr[2]=figura11(X1,Y1); //ver archivo Eficiencia_Perdidas.cpp
Eficiencia[2]=fEficiencia(Eficiencia[1],corr[2]);
int NV=m_numero_valvulas;
corr[3]=figura12(NV,Y1,Xo); //ver archivo
Eficiencia_Perdidas.cpp
Eficiencia[3]=fEficiencia(Eficiencia[2],corr[3]);
Z=Eficiencia[3];
return Z;
}

float CEficienciaTurbina::EficienciaTurbinaTipoD()
{
//función que calcula la eficiencia de una turbina tipo D
//en condiciones de operacion diferentes a la de diseño
//se realiza de la misma forma que la anterior pero con
//diferentes figuras para calcular las correcciones de la
//eficiencia
float Z=0.0;
float Xo=relp();

```

```

        float X1=1.0/Xo;
        calcular_volumen();
        float Y1=m_flujo_diseno*m_volumen_diseno;
        Z=figura13(X1,Y1); //aquí solo se necesita la figura 13 para
//realizar todos los cálculos de eficiencia ver archivo
//Eficiencia_Perdidas.cpp
        return Z;
    }

float CEficienciaTurbina::EficienciaTurbinaTipoE()
{
    //función que calcula la eficiencia de una turbina tipo E
    //en condiciones de operación diferentes a la de diseño
    //se realiza de la misma forma que la anterior pero con
    //diferentes figuras para calcular las correcciones de la
//eficiencia
    float Z=0.0;
    float corr[2],Eficiencia[2];
    int N=m_flujos_entrada;
    calcular_volumen();
    corr[0]=correccion_flujo_volumetrico(TurbinaTipoE,N,m_flujo_dise
no);
    Eficiencia[0]=fEficiencia(EficienciaBase[TurbinaTipoE],corr[0])
;
    float X1=m_presion_entrada_real;
    float Y1=fY1();
    corr[1]=figura14(X1,Y1); //ver archivo Eficiencia_Perdidas.cpp
    Eficiencia[1]=fEficiencia(Eficiencia[0],corr[1]);
    Z=Eficiencia[1];
    return Z;
}

float CEficienciaTurbina::EficienciaTurbinaTipoF()
{
    //función que calcula la eficiencia de una turbina tipo F
    //en condiciones de operación diferentes a la de diseño
    //se realiza de la misma forma que la anterior pero con
    //diferentes figuras para calcular las correcciones de la
//eficiencia
    float Z=0.0;
    float corr[3],Eficiencia[3];
    int N=m_flujos_entrada;
    calcular_volumen();
    corr[0]=correccion_flujo_volumetrico(TurbinaTipoF,N,m_flujo_dise
no);
    Eficiencia[0]=fEficiencia(EficienciaBase[TurbinaTipoF],corr[0])
;
    float X1=m_presion_entrada_real;
    float t1=m_temp_entrada_real;
    float p=m_presion_seccion_1800rpm;
    float Y1=fY1();
    corr[1]=figura14(X1,Y1); //ver archivo Eficiencia_Perdidas.cpp
    Eficiencia[1]=fEficiencia(Eficiencia[0],corr[1]);
    float Ser=Tabla_PT(X1,t1,s1);
    float Her=Tabla_PT(X1,t1,h1);
    const double P1_5=0.7368;

```

```

float H1_5=Calcular_Entalpia(P1_5,Ser);
float He=Calcular_Entalpia(p,Ser);
float DHr=Her-H1_5;
float DH1800=He-H1_5;
corr[2]=(1.25)*(DH1800/DHr);
Eficiencia[2]=fEficiencia(Eficiencia[1],corr[2]);
Z=Eficiencia[2];
return Z;
}
float CEficienciaTurbina::Tabla_PT(float p, float t, int pr)
{
float Z=0.0;
CPropiedad prop;
prop.unidades=inglesas;
prop.presion=p;
prop.temp=t;
prop.tipo_cal=_PT;
prop.calcular();
float s=prop.entropia;
float h=prop.entalpia;
if(pr==s1)
{
Z=s;
return Z;
}
else
if(pr==h1)
{
Z=h;
return Z;
}
return Z;
}

float CEficienciaTurbina::Calcular_Entalpia(float p, float s)
{
float Z=0.0;
CPropiedad prop;
prop.unidades=inglesas;
prop.presion=p;
prop.entropia=s;
prop.tipo_cal=_PS;
prop.calcular();
Z=prop.entalpia;
return Z;
}

float CEficienciaTurbina::EficienciaTurbinaTipoG()
{
//función que calcula la eficiencia de una turbina tipo G
//en condiciones de operación diferentes a la de diseño
//se realiza de la misma forma que la Turbina tipo E pero con
//una entrada diferente: la eficiencia base
//se usan las mismas figuras para el cálculo de las correcciones
float Z=0.0;
float corr[2],Eficiencia[2];
int N=m_flujos_entrada;
calcular_volumen();

```

```

    corr[0]=correccion_flujo_volumetrico(TurbinaTipoG,N,m_flujo_dise
no);
    Eficiencia[0]=fEficiencia(EficienciaBase[TurbinaTipoE],corr[0])
;
    float X1=m_presion_entrada_real;
    float Y1=fY1();
    corr[1]=figural4(X1,Y1); //ver archivo Eficiencia_Perdidas.cpp
    Eficiencia[1]=fEficiencia(Eficiencia[0],corr[1]);
    Z=Eficiencia[1];
    return Z;
    }

float CEficienciaTurbina::calcular_eficiencia()
{
    //función que de acuerdo a al tipo de turbina entrada
    //por el usuario busca su función para calcularla
    float Z=0.0;
    if(comprobar_entrada_datos())
    {
        switch(m_TipoTurbina)
        {
            case TurbinaTipoA:
                Z=EficienciaTurbinaTipoA();
                break;
            case TurbinaTipoB:
                Z=EficienciaTurbinaTipoB();
                break;
            case TurbinaTipoC:
                Z=EficienciaTurbinaTipoC();
                break;
            case TurbinaTipoD:
                Z=EficienciaTurbinaTipoD();
                break;
            case TurbinaTipoE:
                Z=EficienciaTurbinaTipoE();
                break;
            case TurbinaTipoF:
                if(!f)
                { AfxMessageBox(" Falta presion de la seccion
a 1800 rpm");
                    return Z;
                }
                else
                {
                    Z=EficienciaTurbinaTipoF();
                }
                break;
            case TurbinaTipoG:
                Z=EficienciaTurbinaTipoG();
                break;
            default:
                AfxMessageBox(" No ha definido el tipo de
turbina ");
                break;
        }
    }
}

```

```

        }
    }
    return Z;
}

/*int CEficienciaTurbina::comprobar_entrada_datos()    la inhabilite
por innecesaria en el ruedo!!!*/

float CEficienciaTurbina::calcular_presion_salida_real()
{
    ///Función que calcula la presion en condiciones de operacion
//del flujo a la salida de la turbina de acuerdo a función
//planteada por la Elipse de Stodola
    float Z=0.0;
    if( comprobar_entrada_datos())
    {
        calcular_volumen();
        float a1=m_flujo_real/m_flujo_diseno;
        float b1=m_presion_entrada_real/m_presion_entrada_diseno;
        float c1=m_volumen_real/m_volumen_diseno;
        float d1=m_presion_salida_diseno/m_presion_entrada_diseno;
        if(a1>1.0)
            { AfxMessageBox(" flujo masico real mayor que el de
diseño\n"
                            "resultado falso");
              return Z;
            }
        if(b1>1.0)
            {AfxMessageBox(" presin de entrada real mayor que el
de diseño\n"
                            "resultado falso");
              return Z;
            }
        if(d1>1.0)
            { AfxMessageBox(" presion de salida de diseño mayor que
el
                            de
                            la
entrada\n"
                            "resultado falso");
              return Z;
            }
        float A2=pow(a1,2);
        float rp2=pow(d1,2);
        float A=(A2*b1*c1)*(1.0-rp2);
        float Rr=1.0-A;
        Z=sqrt(Rr)*m_presion_entrada_real;
    }
    return Z;
}

```

5. TrabajoTurbina.h

/* Archivo de declaración de la clase CTrabajoTurbina que se usa para calcular el trabajo o potencia generadas por las turbinas a vapor de acuerdo al artículo de la

Asme llamado: A method for predicting the performance of steam turbine-generators
 16.500 Kw and large realizado por R.C.Spencer, K.C.Cotton y C.N.Cannon */

```
// TrabajoTurbina.h: interface for the CTrabajoTurbina class.
#if
!defined(AFX_TRABAJOTURBINA_H__86AD31F4_3BBB_420F_8BF4_770435570EA7_
_INCLUDED_)
#define
AFX_TRABAJOTURBINA_H__86AD31F4_3BBB_420F_8BF4_770435570EA7_INCLUDED

_
#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000
#include "EficienciaTurbina.h"
#include "Turbina_Carga_Parcial.h"

class CTrabajoTurbina
{
public:
    //función que calcula la potencia generada
    // por la turbina en Kws
    float Calcular_trabajo_turbina();
    //funciones que entran datos termodinamicos de los flujos y
    // geometricos de las turbina
    int datosreales_entrada( datostermo der);
    int datosgeometricos_entrada(datosgeometricos dg);
    //////////////////////////////////////
    CTrabajoTurbina(); //constructor por defecto
    CTrabajoTurbina( float eficiencia,float psr) //
    {
        //constructor para la asignación de datos
        // necesarios para los cálculos
        m_eficiencia_turbina=eficiencia;
        m_presion_salida_real=psr;
    }
    virtual ~CTrabajoTurbina(); //destructor
private:
    //funciones de trabajo dentro de la clase
    float calcular_Hs(float f1, float f2, float f3);
    void calcular_area();
    float Volumeng_PX(float p,float x);
    float Calidad_PH(float p, float h);
    float Entalpia_PS(float p,float s);
    float Entropia_PT(float p, float t);
    float Entalpia_PT(float p, float t);
    //////////////////////////////////////
    BOOL x,y;
    ////////////////////////////////////// datos de entrada de trabajo //////////////////////////////////////
    BOOL entrar_area;
    int m_rpm;
    float m_area_anular_ultima_etapa;
    float m_diametro_alabe_ultima_etapa;
    float m_altura_alabe_ultima_etapa;
    float m_eficiencia_turbina;
```

```

        BOOL con_condensacion;
        int m_TipoTurbina;
        float m_presion_salida_real;
        float m_flujo_real;
        float m_temp_entrada_real;
        float m_presion_entrada_real;
        //////////////////////////////////////
};
#endif
#ifndef (AFX_TRABAJOTURBINA_H__86AD31F4_3BBB_420F_8BF4_770435570EA7_
_INCLUDED_)

```

6. TrabajoTurbina.cpp.

/* Archivo de definición de la clase CTrabajoTurbina, la cual se usa en el programa para calcular el trabajo o potencia generada por las turbinas a vapor de acuerdo al artículo de la Asme llamado: A method for predicting the performance of steam turbine-generators 16.500 Kw and large realizado por R.C.Spencer, K.C.Cotton y C.N.Cannon:*/

```

// TrabajoTurbina.cpp: implementation of the CTrabajoTurbina class.
//
////////////////////////////////////
//
#include "stdafx.h"
#include "TrabajoTurbina.h"
#include "Eficiencia_Perdidas.h"
#include "Propiedad.h"
#include <math.h>
#ifdef _DEBUG
#undef THIS_FILE
static char THIS_FILE[]=__FILE__;
#define new DEBUG_NEW
#endif
#define PI 3.141516
////////////////////////////////////
//
// Construction/Destruction
////////////////////////////////////
//

CTrabajoTurbina::CTrabajoTurbina() // constructor
{
    entrar_area=0;
    m_rpm=0;
    m_area_anular_ultima_etapa=0.0;
    m_diametro_alabe_ultima_etapa=0.0;
    m_altura_alabe_ultima_etapa=0.0;
    m_eficiencia_turbina=0.0;
    con_condensacion=0;
}

```

```

        m_TipoTurbina=0;
        m_presion_salida_real=0.0;
        m_flujo_real=0.0;
        m_temp_entrada_real=0.0;
        m_presion_entrada_real=0.0;
        x=y=0;
    }

CTrabajoTurbina::~CTrabajoTurbina()
{
    // destructor
}

int CTrabajoTurbina::datosreales_entrada(datostermo der)
{
    //función que se encarga de entrar los siguientes datos
    //de carga parcial de la turbina
    m_presion_entrada_real=der.presion;
    m_temp_entrada_real=der.temp;
    m_flujo_real=der.flujomasico;
    x=1;
    return 1;
}

int CTrabajoTurbina::datosgeometricos_entrada(datosgeometricos dg)
{
    //función que entra los valores de los datos
    //geometricos de la turbina
    entrar_area=dg.entrar_area;
    m_rpm=dg.rpm;
    m_area_anular_ultima_etapa=dg.Area_Anular_Ultima_Etapa;
    m_diametro_alabe_ultima_etapa=dg.Diametro_Alabe_Ultima_Etapa;
    m_altura_alabe_ultima_etapa=dg.Altura_Alabe_Ultima_Etapa;
    con_condensación=dg.con_condensacion;
    y=1;
    return 1;
}

float CTrabajoTurbina::Calcular_trabajo_turbina()
{
    //Función que calcula la potencia generada por la turbina
    // de acuerdo al articulo de la Asme
    float Z=0.0;
    if(x!=1 || y!=1)
        {AfxMessageBox("Falta entrar datos: geometricos ó
        termodinamicos");
        return Z;
        }
    const double P1_5=0.7368;
    float
He=Entalpia_PT(m_presion_entrada_real,m_temp_entrada_real);
float
Se=Entropia_PT(m_presion_entrada_real,m_temp_entrada_real);
float Hs=0.0;
float Hisol_5=Entalpia_PS(P1_5,Se);
float DHisol_5=He-Hisol_5;
float Hs1_5=He-(m_eficiencia_turbina/100)*DHisol_5;
float X=Calidad_PH(P1_5,Hs1_5);
}

```

```

        float Y=(1.0-X)*100;
float Psringh=m_presion_salida_real*2.036;
float DHo=figural5(Psringh);
//ver archivo Eficiencia_Perdidas.cpp
float DHpsr=DHo*(0.87)*(1.0-(0.01*Y))*(1.0-0.0065*Y);
float Hls=Hsl_5+DHpsr;
float
calidad_prueba=Calidad_PH(m_presion_salida_real,Hls);
if(calidad_prueba<1.0)      {
        con_condensacion=1;
}else
{
        con_condensacion=0;
}
if(con_condensacion)
{
        Hisol_5=Entalpia_PS(P1_5,Se);
        DHisol_5=He-Hisol_5;
        Hsl_5=He-(m_eficiencia_turbina/100)*DHisol_5;
        X=Calidad_PH(P1_5,Hsl_5);
        Y=(1.0-X)*100;
        Psringh=m_presion_salida_real*2.036;
        DHo=figural5(Psringh);
        //ver archivo Eficiencia_Perdidas.cpp
        DHpsr=DHo*(0.87)*(1.0-(0.01*Y))*(1.0-0.0065*Y);
        Hls=Hsl_5+DHpsr;
        Hs=calcular_Hs(Hsl_5,Hls,Y);
}
else
{
        float Hiso=Entalpia_PS(m_presion_salida_real,Se);
        float DHiso=He-Hiso;
        Hs=He-(m_eficiencia_turbina/100)*DHiso;
}

Z=m_flujo_real*(He-Hs)*0.0002928104; //cambio de unidades para
salir como KWs
return Z;
}

float CTrabajoTurbina::Entalpia_PT(float p, float t)
{
        float Z=0.0;
        CPropiedad prop;
        prop.unidades=inglesas;
        prop.presion=p;
        prop.temp=t;
        prop.tipo_cal=_PT;
        prop.calcular();
        Z=prop.entalpia;
        return Z;
}

float CTrabajoTurbina::Entropia_PT(float p, float t)
{
        float Z=0.0;
        CPropiedad prop;
        prop.unidades=inglesas;

```

```

        prop.presion=p;
        prop.temp=t;
        prop.tipo_cal=_PT;
        prop.calcular();
        Z=prop.entropia;
        return Z;
    }

float CTrabajoTurbina::Entalpia_PS(float p, float s)
{
    float Z=0.0;
    CPropiedad prop;
    prop.unidades=inglesas;
    prop.presion=p;
    prop.entropia=s;
    prop.tipo_cal=_PS;
    prop.calcular();
    Z=prop.entalpia;
    return Z;
}

float CTrabajoTurbina::Calidad_PH(float p, float h)
{
    float Z=0.0;
    CPropiedad prop;
    prop.unidades=inglesas;
    prop.presion=p;
    prop.entalpia=h;
    prop.tipo_cal=_PH;
    prop.calcular();
    Z=prop.calidad;
    return Z;
}

float CTrabajoTurbina::Volumeng_PX(float p, float x)
{
    float Z=0.0;
    CPropiedad prop;
    prop.unidades=inglesas;
    prop.presion=p;
    prop.calidad=x;
    prop.tipo_cal=_PX;
    prop.calcular();
    Z=prop.volumen;
    return Z;
}

void CTrabajoTurbina::calcular_area()
{
    if(entrar_area)
    {
        return;
    }
    float
    Dext=(m_diametro_alabe_ultima_etapa+m_altura_alabe_ultima_etapa)/12;
    float
    Dint=(m_diametro_alabe_ultima_etapa-m_altura_alabe_ultima_etapa)/12;

    m_area_anular_ultima_etapa=(PI/4)*(pow(Dext,2)-pow(Dint,2));
}

```

```

float CTrabajoTurbina::calcular_Hs(float Hs1_5, float H1s, float Y)
{
    //Este el cálculo de una entalpia de acuerdo al articulo de la
    //Asme
    float Z=0.0;
    calcular_area();
    float g=m_presion_salida_real/2.036;
    float Vpsr=Volumeng_PX(m_presion_salida_real,1.0);
    float Xpsr=Calidad_PH(m_presion_salida_real,H1s);
    float Ypsr=(1.0-Xpsr)*100;
    float Van=(m_flujo_real*Vpsr*
                (1.0-
0.01*Ypsr))/(3600*m_area_anular_ultima_etapa);
    float DHo=0.0;
    if(Van>1400)
    { float DHpx;
      float Pi=m_presion_salida_real*2.036;
      while ( Van>1400 )
      { float Px= (Pi*Van)/1302;
        ////// se pasa Px de bares a inHg //////
        float Px1= Px;
        //////////////////////////////////////
        //float Lx= log(Px1);
        float DHox= figura15(Px1);
        DHpx=Hs1_5+DHox*(0.87)*(1.0-0.01*Y)*(1.0-0.0065*Y);
        float p1=Px*0.4912;
        float Vpx=Volumeng_PX(p1,1.0);
        float Xpx=Calidad_PH(p1,DHpx);
        float Ypx=(1.0-Xpx)*100;
        Van=(m_flujo_real*Vpx*(1.0-0.01*Ypx))/
(3600*m_area_anular_ultima_etapa);
        Pi=Px;
      }
    }
    DHo=figura16_17_18(Van,m_area_anular_ultima_etapa,m_rpm); //ver
    //archivo Eficiencia_Perdidas.cpp
    Z=H1s+DHo*(0.87)*(1.0-0.01*Ypsr)*(1.0-0.0065*Ypsr);
    return Z;
}

```

7. PerdidasTurbina.h

```

/* Archivo de declaración de la clase CPerdidasTurbina que se usa en el programa
para el cálculo de las perdidas mecánicas en las turbinas a vapor y en el generador de
acuerdo al articulo de la Asme llamado: A method for predicting the performance of
steam turbine-generators 16.500 Kw and large realizado por R.C.Spencer, K.C.Cotton
y C.N.Cannon:*/

```

```

// PerdidasTurbina.h: interface for the CPerdidasTurbina class.
#if
!defined(AFX_PERDIDASTURBINA_H__F84AE8F5_6D31_4E82_9299_668F49D6D19B
__INCLUDED_)
#define
AFX_PERDIDASTURBINA_H__F84AE8F5_6D31_4E82_9299_668F49D6D19B__INCLUDE
D_
#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000
/// se define una estructura para entrar datos del generador
typedef struct{
    BOOL cooled;
    int rpm;
    float Factor_Potencia;
    float presion_hidrogeno_diseno;
    float presion_hidrogeno_real;
    BOOL sc_tcdf_3600;
    int KVA;
    int KWSgen;
}datos_generador;

class CPerdidasTurbina
{
public:
    ///funciones que calculan perdidas
    float Perdidas_Generador();
    float Perdidas_Mecanicas();
    ///función que entra datos
    int entrar_datos_generador(datos_generador d);
    CPerdidasTurbina();
    /*CPerdidasTurbina(int KVA,float KWgenerador)
    {
        m_KVA=KVA;
        m_KWgenerador=KWgenerador;
    }*/
    virtual ~CPerdidasTurbina();
private:
    ///datos de la clase
    BOOL sc_tcdf_3600;
    BOOL cooled_convencional;
    float m_presion_hidrogeno_diseno;
    float m_presion_hidrogeno_real;
    float m_factor_potencia;
    float m_KWgenerador;
    int m_KVA;
    int m_rpm;
    bool x;

};

#endif
!defined(AFX_PERDIDASTURBINA_H__F84AE8F5_6D31_4E82_9299_668F49D6D19B
__INCLUDED_)

```

8.PerdidasTurbina.cpp.

/* Archivo de definición de la clase CPerdidasTurbina que se usa en el programa para el cálculo de las perdidas mecánicas en las turbinas a vapor y en el generador de acuerdo al artículo de la Asme llamado: A method for predicting the performance of steam turbine-generators 16.500 Kw and large realizado por R.C.Spencer, K.C.Cotton y C.N.Cannon:*/

```
// PerdidasTurbina.cpp: implementation of the CPerdidasTurbina
//class.
#include "stdafx.h"
#include "PerdidasTurbina.h"
#include "Eficiencia_Perdidas.h"
#include <math.h>

#ifdef _DEBUG
#undef THIS_FILE
static char THIS_FILE[]=__FILE__;
#define new DEBUG_NEW
#endif

////////////////////////////////////
//
// Construction/Destruction
////////////////////////////////////
//
CPerdidasTurbina::CPerdidasTurbina()
{
    //constructor
        cooled_convencional=0;
        m_presion_hidrogeno_diseno=0.0;
        m_presion_hidrogeno_real=0.0;
        m_factor_potencia=0.0;
        m_KWgenerador=0.0;
        m_KVA=0;
        m_rpm=0;
        x=0;
        sc_tcdf_3600=0;
}

CPerdidasTurbina::~CPerdidasTurbina()
{
    destructor
}

int CPerdidasTurbina::entrar_datos_generador(datos_generador d)
{
    cooled_convencional=d.cooled;
    m_presion_hidrogeno_diseno=d.presion_hidrogeno_diseno;
    m_presion_hidrogeno_real=d.presion_hidrogeno_real;
    m_factor_potencia=d.Factor_Potencia;
    m_rpm=d.rpm;
    sc_tcdf_3600=d.sc_tcdf_3600;
}
```

```

    m_KVA=d.KVA;
    m_KWgenerador=d.KWSgen;
    x=1;
    return 1;
}

float CPerdidasTurbina::Perdidas_Mecanicas()
{
    float Z=0.0;
    Z=figura19(m_KVA,cooled_convencional,m_rpm,sc_tcdf_3600);
    //ver archivo Eficiencia_Perdidas.cpp
    return Z;
}

float CPerdidasTurbina::Perdidas_Generador()
{
    float Z=0.0;
    float K1=figura20(m_KVA,cooled_convencional,m_rpm);
    //ver archivo Eficiencia_Perdidas.cpp
    float KVAoperacion=m_KWgenerador/m_factor_potencia;
    float PKVA=(KVAoperacion/m_KVA);
    float K2=figura21(PKVA,cooled_convencional,m_rpm);
    //ver archivo Eficiencia_Perdidas.cpp
    float DWlgen=(KVAoperacion*K1*K2)/(100);
    float DWgen=0.0;
    if(m_presion_hidrogeno_real<m_presion_hidrogeno_diseno)
    {
        float
        DPh=m_presion_hidrogeno_diseno<m_presion_hidrogeno_real;
        float DKW=figura23(m_KVA,cooled_convencional,m_rpm);
        //ver archivo Eficiencia_Perdidas.cpp
        float DKWr=(DKW*DPh)/15;
        DWgen=DWlgen+DKWr;
    }
    else
    {
        DWgen=DWlgen;
    }

    Z=DWgen;
    return Z;
}

```

Anexo F. Archivos fuentes del programa Intercambiador_Carga_Parcial.exe

La implementación de los elementos de programación en C++ (Clases) se realiza en dos archivos, uno con extensión “.h”, en donde se hacen las declaraciones de los datos y funciones necesarias. En otro archivo de extensión “.cpp”, se realizan las definiciones de cada una de las funciones declaradas en el archivo “.h”.

1. Inter.h.

/*Este es el archivo de declaración de la clase CInter usada en el programa Intercambiador_Carga_Parcial.exe y Condensador_Carga_Parcial.exe para realizar los cálculos a carga parcial.*/

```
// Inter.h: interface for the CInter class.

#if
!defined(AFX_INTER_H__D0611E72_401C_4366_B6AF_340EB25DA046__INCLUDED_)
#define AFX_INTER_H__D0611E72_401C_4366_B6AF_340EB25DA046__INCLUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000
//// se define constantes con nombres convenientes
#define total_flujos      4
#define contracorriente  1
#define unidireccional   0
#define lado_frio        0
#define lado_caliente    1
#define agua_caliente    0
#define agua_domo        1
#define vapor            2
#define flujo_caliente   0
#define flujo_frio       1

/// se define una estructura para los datos termodinámicos de los
//flujos del intercambiador
typedef struct{
    float temp;
    float presión;
    float flujomasico;
    int sistema_unidades;
}termodatos;
```

```

/// se define una estructura para los datos mecánicos del
//intercambiador
typedef struct{
    float Ud;
    BOOL calcular_Ud;
    float Area_total_transferencia_calor;
    int sistema_unidades;
    //BOOL tipo_flujo;
}datosmecanicos;
/// se define una estructura para los exponentes de análisis del
//intercambiador
typedef struct{
    float a0,a1,a2,b0,b1,b2,c0,c1,c2;
    bool pre;
}exponentes_comportamiento;

class CInter
{
public:
    //Funciones que entran los datos para los cálculos
    int asignar_tipo_flujo(bool tipoflujo);
void entrar_exponentes_comportamiento(exponentes_comportamiento cp);
    void asignar_datos_mecanicos(datosmecanicos dm);

    ////funciones que llama el usuario para extraer los resultados de
    //los cálculos
    float perdidas_presión_diseno_lado_frio();
    float perdidas_presión_diseno_lado_caliente();
    float Temp_salida_lado_caliente();
    float Temp_salida_lado_frio();
    float Delta_presión_lado_caliente();
    float Delta_presión_lado_frio();

    //función que realiza los cálculos de transferencia de calor por
    //iteración
    void iteracion_intercambiador();

    CInter();

    CInter(termodatos pd[4],termodatos pr[4])
    {////constructor que asigna valores de flujos al objeto de la
//clase
        prop_diseno[FLUJO_CALIENTE_ENTRANDO].flujomasico=
            pd[FLUJO_CALIENTE_ENTRANDO].flujomasico;
        prop_diseno[FLUJO_CALIENTE_ENTRANDO].presión=
            pd[FLUJO_CALIENTE_ENTRANDO].presión;
        prop_diseno[FLUJO_CALIENTE_ENTRANDO].temp=
            pd[FLUJO_CALIENTE_ENTRANDO].temp;
        prop_diseno[FLUJO_CALIENTE_SALIENDO].flujomasico=
            pd[FLUJO_CALIENTE_SALIENDO].flujomasico;
        prop_diseno[FLUJO_CALIENTE_SALIENDO].presión=
            pd[FLUJO_CALIENTE_SALIENDO].presión;
        prop_diseno[FLUJO_CALIENTE_SALIENDO].temp=
            pd[FLUJO_CALIENTE_SALIENDO].temp;
    }
}

```

```

        prop_real[FLUJO_CALIENTE_ENTRANDO].flujomasico=
        pr[FLUJO_CALIENTE_ENTRANDO].flujomasico;
prop_real[FLUJO_CALIENTE_ENTRANDO].presión=
        pr[FLUJO_CALIENTE_ENTRANDO].presión;
prop_real[FLUJO_CALIENTE_ENTRANDO].temp=
        pr[FLUJO_CALIENTE_ENTRANDO].temp;

        ///// las siguientes propiedades se calculan
prop_real[FLUJO_CALIENTE_SALIENDO].flujomasico=
        pr[FLUJO_CALIENTE_SALIENDO].flujomasico=0.0;
prop_real[FLUJO_CALIENTE_SALIENDO].presión=
        pr[FLUJO_CALIENTE_SALIENDO].presión=0.0;
prop_real[FLUJO_CALIENTE_SALIENDO].temp=
        pr[FLUJO_CALIENTE_SALIENDO].temp=0.0;

prop_diseño[FLUJO_FRÍO_ENTRANDO].flujomasico=
        pd[FLUJO_FRÍO_ENTRANDO].flujomasico;
prop_diseño[FLUJO_FRÍO_ENTRANDO].presión=
        pd[FLUJO_FRÍO_ENTRANDO].presión;
prop_diseño[FLUJO_FRÍO_ENTRANDO].temp=
        pd[FLUJO_FRÍO_ENTRANDO].temp;

prop_diseño[FLUJO_FRÍO_SALIENDO].flujomasico=
        pd[FLUJO_FRÍO_SALIENDO].flujomasico;
prop_diseño[FLUJO_FRÍO_SALIENDO].presión=
        pd[FLUJO_FRÍO_SALIENDO].presión;
prop_diseño[FLUJO_FRÍO_SALIENDO].temp=
        pd[FLUJO_FRÍO_SALIENDO].temp;

prop_real[FLUJO_FRÍO_ENTRANDO].flujomasico=
        pr[FLUJO_FRÍO_ENTRANDO].flujomasico;
prop_real[FLUJO_FRÍO_ENTRANDO].presión=
        pr[FLUJO_FRÍO_ENTRANDO].presión;
prop_real[FLUJO_FRÍO_ENTRANDO].temp=
        pr[FLUJO_FRÍO_ENTRANDO].temp;

        ///// las siguientes propiedades se calculan
prop_real[FLUJO_FRÍO_SALIENDO].flujomasico=
        pr[FLUJO_FRÍO_SALIENDO].flujomasico=0.0;
prop_real[FLUJO_FRÍO_SALIENDO].presión=
        pr[FLUJO_FRÍO_SALIENDO].presión=0.0;
prop_real[FLUJO_FRÍO_SALIENDO].temp=
        pr[FLUJO_FRÍO_SALIENDO].temp=0.0;

        zona[0]=0;
        zona[1]=0;
}

virtual ~CInter(); //destructor

private:
        /////funciones de trabajo de la clase/////
void cálculos_flujoparalelo();
void cálculos_contraflujo();

```

```

float Entalpia_PT_aguasaturada(float p,float t);
float Entalpia_PT_vaporsaturado(float p, float t);
float Buscar_Temp_saturación(float p);
int zona[2];
float Entalpia_PX(float p, float x);
float Calidad_PT(float p, float t);
void calculo_presión_salida();
float Temp_PH(float p, float h);
float Entalpia_PT(float p, float t);
float calculo_Uo();
float perdidas_presión_lado_caliente();
float elevar(float x, float y);
float perdidas_presión_lado_frío();
float calculo_volumen(float p, float t);
float calor_maximo_diseno();
float calculo_Ud();
float Temp_media( float t1, float t2);
float Lmtd_flujo_ebullición(float Tec, float Tsc, float Ts);
float Lmtd_contraflujo(float Tec,float Tef, float Tsc, float
Tsf);
float Lmtd_flujo_paralelo(float Tec,float Tef, float Tsc, float
Tsf);
float presión_media(float p1,float p2);
float calculo_Cp(float p, float t);

        //////////////////////////////////////////////////variables de trabajo de la clase
int y;
float a0,a1,a2,b0,b1,b2,c0,c1,c2,d0;
int x;
float Ud;
BOOL calcular_Ud;
float Area_total_transferencia_calor;
float perdidas_presión_real[2];
float perdidas_presión_diseno[2];
BOOL TipoFlujo;
float LMTD_diseno;
termodatos_prop_diseno[total_flujos];
termodatos_prop_real[total_flujos];

};

#endif
#ifndef(AFX_INTER_H__D0611E72_401C_4366_B6AF_340EB25DA046__INCLUDED
_)

```

2. Inter.cpp.

```

/* Este es el archivo de definición de la clase CInter usada en el programa
Intercambiador_Carga_Parcial.exe y Condensador_Carga_Parcial.exe para realizar
los cálculos a carga parcial.*/

```

```

// Inter.cpp: implementation of the CInter class.
//
#include "stdafx.h"
#include "Intercambiador_Carga_Parcial.h"
#include "Inter.h"
#include "propiedad.h"
#include "unidades.h"
#include <math.h>

#ifdef _DEBUG
#undef THIS_FILE
static char THIS_FILE[]=__FILE__;
#define new DEBUG_NEW
#endif

// Construction/Destruction
////////////////////////////////////
//

CInter::CInter()
{
    prop_diseno[FLUJO_CALIENTE_ENTRANDO].flujomasico=0.0;
    prop_diseno[FLUJO_CALIENTE_ENTRANDO].presión=0.0;
    prop_diseno[FLUJO_CALIENTE_ENTRANDO].temp=0.0;

    prop_diseno[FLUJO_CALIENTE_SALIENDO].flujomasico=0.0;
    ///no es necesaria
    prop_diseno[FLUJO_CALIENTE_SALIENDO].presión=0.0;/// se calcula
    prop_diseno[FLUJO_CALIENTE_SALIENDO].temp=0.0;

    prop_real[FLUJO_CALIENTE_ENTRANDO].flujomasico=0.0;
    prop_real[FLUJO_CALIENTE_ENTRANDO].presión=0.0;
    prop_real[FLUJO_CALIENTE_ENTRANDO].temp=0.0;

    prop_real[FLUJO_CALIENTE_SALIENDO].flujomasico=0.0;
    ///no es necesaria
    prop_real[FLUJO_CALIENTE_SALIENDO].presión=0.0;/// se calcula
    prop_real[FLUJO_CALIENTE_SALIENDO].temp=0.0;
    //////////////////////////////////////
    prop_diseno[FLUJO_FRÍO_ENTRANDO].flujomasico=0.0;
    prop_diseno[FLUJO_FRÍO_ENTRANDO].presión=0.0;
    prop_diseno[FLUJO_FRÍO_ENTRANDO].temp=0.0;

    prop_diseno[FLUJO_FRÍO_SALIENDO].flujomasico=0.0;
    ///no es necesaria
    prop_diseno[FLUJO_FRÍO_SALIENDO].presión=0.0;///se calcula
    prop_diseno[FLUJO_FRÍO_SALIENDO].temp=0.0;

    prop_real[FLUJO_FRÍO_ENTRANDO].flujomasico=0.0;
    prop_real[FLUJO_FRÍO_ENTRANDO].presión=0.0;
    prop_real[FLUJO_FRÍO_ENTRANDO].temp=0.0;

    prop_real[FLUJO_FRÍO_SALIENDO].flujomasico=0.0;

```

```

        ///no es necesaria
        prop_real[FLUJO_FRÍO_SALIENDO].presión=0.0;//se calcula
        prop_real[FLUJO_FRÍO_SALIENDO].temp=0.0;
    }

CInter::~CInter()
{
}

void CInter::asignar_datos_mecanicos(datosmecanicos dm)
{
    Ud=dm.Ud;
    calcular_Ud=dm.calcular_Ud;

    Area_total_transferencia_calor=dm.Area_total_transferencia_calor;
    perdidas_presión_real[lado_frío]=0.0;//se calcula
    perdidas_presión_diseno[lado_frío]=0.0;
    perdidas_presión_real[lado_caliente]=0.0;//se calcula
    perdidas_presión_diseno[lado_caliente]=0.0;
    TipoFlujo=dm.tipo_flujo;
    x=1;
}

void
CInter::entrar_exponentes_comportamiento(exponentes_comportamiento
cp)
{
    a0=cp.a0;
    a1=cp.a1;
    a2=cp.a2;
    b0=cp.b0;
    b1=cp.b1;
    b2=cp.b2;
    c0=cp.c0;
    c1=cp.c1;
    c2=cp.c2;
    y=1;
}

float CInter::Temp_media( float t1, float t2)
{
    float z=0.0;
    z=(t1+t2)/2;
    return z;
}

float CInter::presión_media(float p1, float p2)
{
    float z=0.0;
    z=(p1+p2)/2;
    return z;
}

float CInter::Lmtd_flujo_paralelo(float Tec, float Tef, float Tsc,
float Tsf)
{
    float z=0.0;
    if(Tsc<Tsf)

```

```

        {AfxMessageBox("La temp de salida mas fria es mayor que la
caliente\n"
        " el resultado sera erroneo:");
        return z;
    }
    float a=(Tec-Tef)-(Tsc-Tsf);
    float b=log((Tec-Tef)/(Tsc-Tsf));
    z=a/b;
    return z;
}

float CInter::Lmtd_contraflujo(float Tec, float Tef, float Tsc,
float Tsf)
{
    float z=0.0;
    if(Tec<Tsf)
    {
        return z;
    }
    if(Tsc<Tef)
    {
        return z;
    }
    float a=(Tec-Tsf)-(Tsc-Tef);
    float b=log((Tec-Tsf)/(Tsc-Tef));
    z=a/b;
    return z;
}

float CInter::Lmtd_flujo_ebullendo(float Tec, float Tsc, float Ts)
{
    float z=0.0;
    if(Ts>Tec || Ts>Tsc)
    {AfxMessageBox(" Temp de ebullición debe ser menor que la \n"
        "temperatura de entrada y de salida del fluido
caliente\n"
        "el resultado sera erroneo");
        return z;
    }

    if(Tec<Tsc)
    {AfxMessageBox(" la temp entrada caliente debe \n"
        "ser mayor que la de salida\n"
        "el resultado sera erroneo");
        return z;
    }

    float a=(Tec-Tsc);
    float b=log((Tec-Ts)/(Tsc-Ts));
    z=a/b;
    return z;
}

float CInter::calculo_Cp(float p, float t)
{
    float z=0.0;

```

```

        if(t>1400)
        { AfxMessageBox(" la temp es mayor de 1400 K\n"
            " no se puede calcular, el resultado sera
erroneo");
            return z;
        }
        CPropiedad prop;
        prop.unidades=inglesas;
        double t1=t-1;
        double t2=t+1;
        prop.presión= p;
        prop.temp= t1;
        prop.tipo_cal=_PT;
        prop.calcular();
        double h1=prop.entalpia;
        prop.temp= t2;
        prop.tipo_cal=_PT;
        prop.calcular();
        double h2=prop.entalpia;
        double C = (h2-h1)/((t+0.1)-(t-0.1));
        z=C/10;
        return z;
    }

float CInter::calculo_Ud()
{
    float z=0.0;
    float Tec=prop_diseno[FLUJO_CALIENTE_ENTRANDO].temp;
    float Tsc=prop_diseno[FLUJO_CALIENTE_SALIENDO].temp;
    float Tef=prop_diseno[FLUJO_FRÍO_ENTRANDO].temp;
    float Tsf=prop_diseno[FLUJO_FRÍO_SALIENDO].temp;
    float Delta_temp=prop_diseno[FLUJO_CALIENTE_ENTRANDO].temp-
prop_diseno[FLUJO_CALIENTE_SALIENDO].temp;

    if(TipoFlujo==contracorriente)
    {
        LMTD_diseno=Lmtd_contraflujo(Tec,Tef,Tsc,Tsf);
    }
    else if(TipoFlujo==unidireccional)
    {
        LMTD_diseno=Lmtd_flujo_paralelo(Tec,Tef,Tsc,Tsf);
    }
    if(calcular_Ud==0)
    {
        z=Ud;
        return z;
    }
    float Cmax=calor_maximo_diseno();
    z=(Cmax)/(Area_total_transferencia_calor*LMTD_diseno);
    return z;
}

float CInter::calor_maximo_diseno()
{
    flota z=0.0;
    prop_diseno[FLUJO_CALIENTE_SALIENDO].presión=

```

```

        prop_diseño[FLUJO_CALIENTE_ENTRANDO].presión-
perdidas_presión_diseño[lado_caliente];

        prop_diseño[FLUJO_FRÍO_SALIENDO].presión=
        prop_diseño[FLUJO_FRÍO_ENTRANDO].presión-
perdidas_presión_diseño[lado_frío];

        float pmcd=
presión_media(prop_diseño[FLUJO_CALIENTE_ENTRANDO].presión,prop_dise-
ño[FLUJO_CALIENTE_SALIENDO].presión);

        float pmfd=
presión_media(prop_diseño[FLUJO_FRÍO_ENTRANDO].presión,prop_diseño[F-
LUJO_FRÍO_SALIENDO].presión);

        float tmcd=
Temp_media(prop_diseño[FLUJO_CALIENTE_ENTRANDO].temp,prop_diseño[FLU-
JO_CALIENTE_SALIENDO].temp);

        float tmfd=
Temp_media(prop_diseño[FLUJO_FRÍO_ENTRANDO].temp,prop_diseño[FLUJO_F-
RÍO_SALIENDO].temp);
        float Cpfd=calculo_Cp(pmfd,tmfd);
        float Cpcd=calculo_Cp(pmcd,tmcd);
        float Cmfd=
prop_diseño[FLUJO_FRÍO_ENTRANDO].flujomasico*Cpfd;

        float Cmcd=
prop_diseño[FLUJO_CALIENTE_ENTRANDO].flujomasico*Cpcd;

        if(Cmcd>Cmfd)
        {
                                z=
Cmfd*(prop_diseño[FLUJO_FRÍO_SALIENDO].temp-
prop_diseño[FLUJO_FRÍO_ENTRANDO].temp);
        }
        else if( Cmcd<Cmfd)
        {
                z=Cmcd*(prop_diseño[FLUJO_CALIENTE_ENTRANDO].temp-
prop_diseño[FLUJO_CALIENTE_SALIENDO].temp);
        }
        return z;
}

float CInter::calculo_volumen(float p, float t)
{
    float z=0.0;
    CPropiedad prop;
    prop.unidades=inglesas;
    prop.presión= p;
    prop.temp= t;
    prop.tipo_cal=_PT;
    prop.calcular();
    z=prop.volumen;
    return z;
}

```

```

}

float CInter::perdidas_presión_lado_frío()
{
    float z=0.0;
    perdidas_presión_diseno[lado_frío]=
    prop_diseno[FLUJO_FRÍO_ENTRANDO].presión-
    prop_diseno[FLUJO_FRÍO_SALIENDO].presión;

    float rflujomasico=
    prop_real[FLUJO_FRÍO_ENTRANDO].flujomasico/prop_diseno[FLUJO_FRÍO_EN
    TRANDO].flujomasico;

    float rtemp=
    prop_real[FLUJO_FRÍO_ENTRANDO].temp/prop_diseno[FLUJO_FRÍO_ENTRANDO]
    .temp;

    float rpresión=
    prop_real[FLUJO_FRÍO_ENTRANDO].presión/prop_diseno[FLUJO_FRÍO_ENTRAN
    DO].presión;

    z=perdidas_presión_diseno[lado_frío]*elevar(rflujomasico,a0)*el
    evar(rtemp,b0)*elevar(rpresión,c0);//*elevar(rvelocidad,d0);
    return z;
}

float CInter::elevar(float x, float y)
{
    float z=0.0;
    z=pow(x,y);
    return z;
}

float CInter::perdidas_presión_lado_caliente()
{
    float z=0.0;
    perdidas_presión_diseno[lado_caliente]=
    prop_diseno[FLUJO_CALIENTE_ENTRANDO].presión-
    prop_diseno[FLUJO_CALIENTE_SALIENDO].presión;

    float rflujomasico=
    prop_real[FLUJO_CALIENTE_ENTRANDO].flujomasico/prop_diseno[FLUJO_CAL
    IENTE_ENTRANDO].flujomasico;

    float rtemp=
    prop_real[FLUJO_CALIENTE_ENTRANDO].temp/prop_diseno[FLUJO_CALIENTE_E
    NTRANDO].temp;

    float rpresión=
    prop_real[FLUJO_CALIENTE_ENTRANDO].presión/prop_real[FLUJO_CALIENTE_
    ENTRANDO].presión;

    z=perdidas_presión_diseno[lado_caliente]*elevar(rflujomasico,a1)*ele
    var(rtemp,b1)*elevar(rpresión,c1);
    return z;
}

```

```

float CInter::calculo_Uo()
{
    float z=0.0;
    float U_diseno=calculo_Ud();
    float rflujomasico=
prop_real[FLUJO_CALIENTE_ENTRANDO].flujomasico/prop_diseno[FLUJO_CAL
IENTE_ENTRANDO].flujomasico;

    float rtemp=
prop_real[FLUJO_CALIENTE_ENTRANDO].temp/prop_diseno[FLUJO_CALIENTE_E
NTRANDO].temp;

    float rpresión=
prop_real[FLUJO_CALIENTE_ENTRANDO].presión/prop_diseno[FLUJO_CALIENT
E_ENTRANDO].presión;
z=U_diseno*elevar(rflujomasico,a2)*elevar(rtemp,b2)*elevar(rpresión,
c2);
    return z;
}

void CInter::iteracion_intercambiador()
{
    float Tec=prop_diseno[FLUJO_CALIENTE_ENTRANDO].temp;
    float Tsc=prop_diseno[FLUJO_CALIENTE_SALIENDO].temp;
    float Tef=prop_diseno[FLUJO_FRÍO_ENTRANDO].temp;
    float Tsf=prop_diseno[FLUJO_FRÍO_SALIENDO].temp;

    if(Tsc<=Tef)
        {AfxMessageBox("Las temperaturas de diseño de salida condensada
no pueden ser ni igual ni menor \n"
"de la temperatura de diseño de entrada del agua de
enfriamiento");
        return;
    }
    float difencia=Tec-Tsf;
    if(difencia<50)
        {AfxMessageBox(" La temperatura de diseño de salida del agua de
enfriamiento esta muy cerca\n"
" a la temperatura de entrada del vapor, no es
creible!");
        return;
    }
    calculo_presión_salida();
    switch(TipoFlujo)
    {
        case unidireccional:
            cálculos_flujoparalelo();
            break;

        case contracorriente:
            cálculos_contraflujo();
            break;
        default:
            AfxMessageBox("Defina tipo de flujo");
            break;
    }
}

```

```

}

float CInter::Entalpia_PT(float p, float t)
{
    float z=0.0;
    CPropiedad prop;
    prop.unidades=inglesas;
    prop.presión=p;
    prop.temp=t;
    prop.tipo_cal=_PT;
    prop.calcular();
    z=prop.entalpia;
    return z;
}

float CInter::Temp_PH(float p, float h)
{
    float z=0.0;
    CPropiedad prop;
    prop.unidades=inglesas;
    prop.presión=p;
    prop.entalpia=h;
    prop.tipo_cal=_PH;
    prop.calcular();
    z=prop.temp;
    return z;
}

void CInter::calculo_presión_salida()
{
    perdidas_presión_real[lado_caliente]=perdidas_presión_lado_caliente();
    perdidas_presión_real[lado_frío]=perdidas_presión_lado_frío();
    float Dpco=perdidas_presión_real[lado_caliente];
    float Dpfo=perdidas_presión_real[lado_frío];
    prop_real[FLUJO_CALIENTE_SALIENDO].presión=
        prop_real[FLUJO_CALIENTE_ENTRANDO].presión-
    perdidas_presión_real[lado_caliente];
    prop_real[FLUJO_FRÍO_SALIENDO].presión=
        prop_real[FLUJO_FRÍO_ENTRANDO].presión-
    perdidas_presión_real[lado_frío];
}

float CInter::Delta_presión_lado_frío()
{
    return perdidas_presión_real[lado_frío];
}

float CInter::Delta_presión_lado_caliente()
{
    return perdidas_presión_real[lado_caliente];
}

float CInter::Temp_salida_lado_frío()
{
    return prop_real[FLUJO_FRÍO_SALIENDO].temp;
}

float CInter::Temp_salida_lado_caliente()
{
    return prop_real[FLUJO_CALIENTE_SALIENDO].temp;
}

```

```

}

float CInter::perdidas_presión_diseno_lado_caliente()
{
    return perdidas_presión_diseno[lado_caliente];
}

float CInter::perdidas_presión_diseno_lado_frío()
{
    return perdidas_presión_diseno[lado_frío];
}

float CInter::Calidad_PT(float p, float t)
{
    float z=0.0;
    CPropiedad prop;
    prop.unidades=inglesas;
    prop.presión=p;
    prop.temp=t;
    prop.tipo_cal=_PT;
    prop.calcular();
    z=prop.calidad;
    return z;
}

float CInter::Entalpia_PX(float p, float x)
{
    float z=0.0;
    CPropiedad prop;
    prop.unidades=inglesas;
    prop.presión=p;
    prop.calidad=x;
    prop.tipo_cal=_PX;
    prop.calcular();

    z=prop.entalpia;
    return z;
}

float CInter::Buscar_Temp_saturación(float p)
{
    float z=0.0;
    float t=500;
    CPropiedad prop;
        prop.unidades=inglesas;
    prop.calidad=0.0;
    prop.presión=p;
    prop.tipo_cal=_PX;
    prop.calcular();
    z=prop.temp;
    return z;
}

float CInter::Entalpia_PT_vaporsaturado(float p, float t)
{
    float z=0.0;
    CPropiedad prop;
    prop.unidades=inglesas;
    prop.presión=p;
    prop.temp=t;
}

```

```

        prop.calidad=1.0;
        prop.tipo_cal=_PT;
        prop.calcular();
        z=prop.entalpia;
        return z;
    }

float CInter::Entalpia_PT_aguasaturada(float p, float t)
{
    float z=0.0;
    CPropiedad prop;
    prop.unidades=inglesas;
    prop.presión=p;
    prop.temp=t;
    prop.calidad=0.0;
    prop.tipo_cal=_PT;
    prop.calcular();
    z=prop.entalpia;
    return z;
}

void CInter::cálculos_contraflujo()
{
    float U_real=calculo_Uo();
    float Teco=prop_real[FLUJO_CALIENTE_ENTRANDO].temp;
    float Tefo=prop_real[FLUJO_FRÍO_ENTRANDO].temp;
    float Peco=prop_real[FLUJO_CALIENTE_ENTRANDO].presión;
    float Pefo=prop_real[FLUJO_FRÍO_ENTRANDO].presión;
    float Psco=prop_real[FLUJO_CALIENTE_SALIENDO].presión;
    float Psfo=prop_real[FLUJO_FRÍO_SALIENDO].presión;
    float mco=prop_real[FLUJO_CALIENTE_ENTRANDO].flujomasico;
    float mfo=prop_real[FLUJO_FRÍO_ENTRANDO].flujomasico;
    float Heco=0.0;
    float Tsco=0.0;
    float Tsfo=0.0;
    float Hsco=0.0;
    float Hsfo=0.0;
    float Hefo=0.0;
    float x1=Calidad_PT(Peco,Teco);
    if(x1==0.0)
    {
        zona[flujo_caliente]=agua_domo;
        AfxMessageBox("Entre nuevos valores de entrada, en
el flujo más caliente\n"
"en operación, ya que este flujo esta dentro
del domo");
        Tsco=0.0;
        Tsfo=0.0;
        return;
    }
    else if(x1==100)
    {
        zona[flujo_caliente]=vapor;
        Heco=Entalpia_PT(Peco,Teco);
    }
    else if(x1==-100)
    {
        zona[flujo_caliente]=agua_caliente;
    }
}

```

```

        Heco=Entalpia_PT(Peco,Teco);
    }

    float x2=Calidad_PT(Pefo,Tefo);
    if(x2==0.0)
    {
        zona[flujo_frío]=agua_domo;
        AfxMessageBox("Entre nuevos valores de entrada, en el flujo más
frío\n"
                    "en operación o entre la calidad, es agua
dentro del domo");
        Tsco=0.0;
        Tsfo=0.0;
        return;
    }
    else if(x2==100)
    {
        zona[flujo_frío]=vapor;
        Hefo=Entalpia_PT(Pefo,Tefo);
    }
    else if(x2==-100)
    {
        zona[flujo_frío]=agua_caliente;
        Hefo=Entalpia_PT(Pefo,Tefo);
    }
    float Q0=0.0;
    float Q1=0.0;
    float Tsu=(Teco+Tefo)/2;
    float LMTD_real=0.0;
    int contador=0;
    double limite1=0.0;
    double limite2=0.0;
    float Tx=0.0;
    float Ty=0.0;
    float Qx=0.0;
    //////////////////////////////////////////// Cálculos por zona
    ////////////////////////////////////////////
    if(zona[flujo_caliente]==vapor && zona[flujo_frío]==vapor)
    {
        float valor=Teco-Tefo;
        if(valor<35)
        {
            AfxMessageBox("Con los valores dados no se pueden hacer
cálculos");
            Tsfo=0.0;
            Tsco=0.0;
            return;
        }
        float Tx=0.0;
        float Ty=0.0;
        float Ts_vapor_flujo_caliente_salida=
            Buscar_Temp_saturación(Psco);
        ///calculat temps
        do{
            if(Tsco==0.0)
            {
                Tx=Teco-10;
                Hsco=Entalpia_PT_vaporsaturado(Psco,Tx);
            }
        }
    }

```

```

    }

    else
    {
        Tx=TSCO;
        HSCO=Entalpia_PT(PSCO,TX);
    }
    Q0=mco*(HECO-HSCO);
    limite2=10;
    do{

        if(Tsfo==0.0)
        {Ty=Tefo+5;
        }
        else
        {Ty=Tsfo;
        }

LMTD_real=Lmtd_contraflujo(Teco,Tefo,Tx,Ty);

Q1=U_real*Area_total_transferencia_calor*LMTD_real;
    if(Q1<Q0)
    {

Hsfo=Hefo+(Q1/mfo);

                                Tsfo=Temp_PH(Psfo,Hsfo);
        }
        else
        {
            Hsfo=Hefo+(Q0/mfo);
            Tsfo=Temp_PH(Psfo,Hsfo);
        }
        limite2=fabs(Ty-Tsfo);
    }while(limite2>5);

    HSCO=HECO-(Q1/mco);
    TSCO=Temp_PH(PSCO,HSCO);
    limite1=fabs(TX-TSCO);
}while(limite1>5);

if(TSCO<=Ts_vapor_flujo_caliente_salida)
{
    TSCO=Ts_vapor_flujo_caliente_salida;
    HSCO=Entalpia_PT_vaporsaturado(PSCO,TSCO);
    do{
        if(Tsfo==0.0)
        {
            Ty=Tefo+5;
        }
        else
        {
            Ty=Tsfo;
        }
    }
}

```

```

float
LMTD=Lmtd_contraflujo(Teco,Tefo,Tsco,Ty);
Q1=U_real*Area_total_transferencia_calor*LMTD;
if(Q1<Q0)
{
    Hsfo=Hefo+(Q1/mfo);
    Tsfo=Temp_PH(Psfo,Hsfo);
}
else
{
    Hsfo=Hefo+(Q0/mfo);
    Tsfo=Temp_PH(Psfo,Hsfo);
}
limite2=fabs(Ty-Tsfo);
}while(limite2>10);
}
prop_real[FLUJO_CALIENTE_SALIENDO].temp=Tsco;
prop_real[FLUJO_FRÍO_SALIENDO].temp=Tsfo;
return;
}

if(zona[flujo_caliente]==vapor &&
zona[flujo_frío]==agua_caliente)
{
    ///calcular temps
    float Temp_saturación_flujo_frío_salida=
        Buscar_Temp_saturación(Psfo);
    float Temp_saturación_flujo_caliente_entrada=
        Buscar_Temp_saturación(Peco);
    float Temp_saturación_flujo_caliente_salida=
        Buscar_Temp_saturación(Psco);

    if(Tefo>Temp_saturación_flujo_frío_salida)
    {AfxMessageBox("Con los valores dados del flujo frío en
operación no se pueden hacer cálculos.....\n"
        "cambie el valor de la presión de salida,
aumentandola o disminuya la temperatura de entrada");
    ///el valor de Tefo es mayor que la temp de saturación a la presión
    //de salida
        Tsfo=0.0;
        Tsco=0.0;
        return;
    }
    float valor=Teco-Tefo;
    if(valor<35)
    {AfxMessageBox("Con los valores dados no se pueden hacer
cálculos");
        Tsfo=0.0;
        Tsco=0.0;
        return;
    }
    float Tx=0.0;
    float Ty=0.0;
    ///calcular temps
    do{

```

```

if(Tsco==0.0)
{
    Tx=Temp_saturación_flujo_caliente_entrada;
    Hsco=Entalpia_PT_vaporsaturado(Psco,Tx);
}
else
{
    Tx=Tsco;
    Hsco=Entalpia_PT(Psco,Tx);
}
float Q0=0.0;
float Q1=0.0;
Q0=mco*(Heco-Hsco);
limite2=10;
do{
    if(Tsfo==0.0)
    {Ty=Temp_saturación_flujo_frio_salida;
    }
    else
    {Ty=Tsfo;
    }
    float LMTD=Lmtd_contraflujo(Teco,Tefo,Tx,Ty);

Q1=U_real*Area_total_transferencia_calor*LMTD;
    if(Q1<Q0)
    { Hsfo=Hefo+(Q1/mfo);
      Tsfo=Temp_PH(Psfo,Hsfo);
    }
    else
    { Hsfo=Hefo+(Q0/mfo);
      Tsfo=Temp_PH(Psfo,Hsfo);
    }
    limite2=fabs(Ty-Tsfo);
    if(Tsfo>=Temp_saturación_flujo_frio_salida)
    {Tsfo=Temp_saturación_flujo_frio_salida;
      limite2=3;
    }
}while(limite2>5);
Hsco=Heco-(Q1/mco);
Tsco=Temp_PH(Psco,Hsco);

limite1=fabs(Tx-Tsco);
}while(limite1>5);

if(Tsco<=Temp_saturación_flujo_caliente_salida)
{
    Tsco=Temp_saturación_flujo_caliente_salida;

    Hsco=Entalpia_PT_vaporsaturado(Psco,Tsco);
    Q0=mco*(Heco-Hsco);
    do{

if(Tsfo==0.0)

        {
            Ty=Tefo+5;
        }
        else
        {

```

```

        Ty=Tsfo;
        }

        float
LMTD=Lmtd_contraflujo (Teco, Tefo, Tsco, Ty);

Q1=U_real*Area_total_transferencia_calor*LMTD;
        if (Q1<Q0)
        {Hsfo=Hefo+(Q1/mfo);
          Tsfo=Temp_PH (Psfo, Hsfo);
        }
        else
        {   Hsfo=Hefo+(Q0/mfo);
          Tsfo=Temp_PH (Psfo, Hsfo);
        }
        limite2=fabs (Ty-Tsfo);
    }while (limite2>5);

prop_real [FLUJO_CALIENTE_SALIENDO].temp=Tsco;
        prop_real [FLUJO_FRÍO_SALIENDO].temp=Tsfo;
    return;
}
prop_real [FLUJO_CALIENTE_SALIENDO].temp=Tsco;
        prop_real [FLUJO_FRÍO_SALIENDO].temp=Tsfo;
    return;
}

if (zona [flujo_caliente]==agua_caliente&&
zona [flujo_frío]==agua_caliente)
{
    //calcular temps
    float Temp_saturación_flujo_frío_salida=
        Buscar_Temp_saturación (Psfo);
    float Temp_saturación_flujo_caliente_entrada=
        Buscar_Temp_saturación (Peco);
    float Temp_saturación_flujo_caliente_salida=
        Buscar_Temp_saturación (Psco);

        if (Tefo>Temp_saturación_flujo_frío_salida)
        {AfxMessageBox ("Con los valores dados del flujo frío en
operación no se pueden hacer cálculos");
        /////el valor de Tefo es mayor que la temp de saturación a la
//presión de salida
            Tsfo=0.0;
            Tsco=0.0;
            return;
        }
        float valor=Teco-Tefo;
        if (valor<35)
        {AfxMessageBox ("Con los valores dados no se pueden hacer
cálculos");
            Tsfo=0.0;
            Tsco=0.0;
            return;
        }
}

```

```

float Tx=0.0;
float Ty=0.0;
float
Ts_vapor_flujo_frio_salida=Buscar_Temp_saturación(Psfo);
///calcular temps
do{
    if(Tsco==0.0)
    {
        Tx=Teco-10;
        Hsco=Entalpia_PT_vaporsaturado(Psco,Tx);
    }
    else
    {
        Tx=Tsco;
        Hsco=Entalpia_PT(Psco,Tx);
    }
    float Q0=0.0;
                                float Q1=0.0;
    Q0=mco*(Heco-Hsco);
    limite2=10;
    do{
        if(Tsfo==0.0)
        {
                                                    Ty=Tefo+5;
        }
        else
        {
            Ty=Tsfo;
        }
        float LMTD=Lmtd_contraflujo(Teco,Tefo,Tx,Ty);
        Q1=U_real*Area_total_transferencia_calor*LMTD;

        if(Q1<Q0)
        {
            Hsfo=Hefo+(Q1/mfo);
            Tsfo=Temp_PH(Psfo,Hsfo);
        }
        else
        {
            Hsfo=Hefo+(Q0/mfo);
            Tsfo=Temp_PH(Psfo,Hsfo);
        }
        limite2=fabs(Ty-Tsfo);
    }while(limite2>5);
    Hsco=Heco-(Q1/mco);
    Tsco=Temp_PH(Psco,Hsco);
    limite1=fabs(Tx-Tsco);
}while(limite1>5);

if(Tsfo>=Ts_vapor_flujo_frio_salida)
{
    Tsfo=Ts_vapor_flujo_frio_salida;
    Hsfo=Entalpia_PT(Psfo,Tsfo);
    Q0=mfo*(Hsfo-Hefo);
    do{
        if(Tsco==0.0)
        {
            Tx=Teco+5;
        }
        else
        {
            Tx=Tsco;
        }
    }
}

```

```

float
LMTD=Lmtd_contraflujo(Teco,Tefo,Tx,Tsfo);

Q1=U_real*Area_total_transferencia_calor*LMTD;

if(Q1<Q0)
{ Hsfo=Hefo+(Q1/mfo);
  Tsfo=Temp_PH(Psfo,Hsfo);
}
else
{ Hsfo=Hefo+(Q0/mfo);
  Tsfo=Temp_PH(Psfo,Hsfo);
}
limite2=fabs(Ty-Tsfo);
}while(limite2>5);
}
prop_real[FLUJO_CALIENTE_SALIENDO].temp=Tsco;
prop_real[FLUJO_FRÍO_SALIENDO].temp=Tsfo;
return;
}
AfxMessageBox("No se hacen cálculos con los valores dados");
prop_real[FLUJO_CALIENTE_SALIENDO].temp=Tsco;
prop_real[FLUJO_FRÍO_SALIENDO].temp=Tsfo;
}

void CInter::cálculos_flujoparalelo()
{
float U_real=calculo_Uo();
float Teco=prop_real[FLUJO_CALIENTE_ENTRANDO].temp;
float Tefo=prop_real[FLUJO_FRÍO_ENTRANDO].temp;
float Peco=prop_real[FLUJO_CALIENTE_ENTRANDO].presión;
float Pefo=prop_real[FLUJO_FRÍO_ENTRANDO].presión;
float Psco=prop_real[FLUJO_CALIENTE_SALIENDO].presión;
float Psfo=prop_real[FLUJO_FRÍO_SALIENDO].presión;
float mco=prop_real[FLUJO_CALIENTE_ENTRANDO].flujomasico;
float mfo=prop_real[FLUJO_FRÍO_ENTRANDO].flujomasico;
float Heco=0.0;
float Tsco=0.0;
float Tsfo=0.0;
float Hsco=0.0;
float Hsfo=0.0;
float Hefo=0.0;
float x1=Calidad_PT(Peco,Teco);
if(x1==0.0)
{
zona[flujo_caliente]=agua_domo;
AfxMessageBox("Entre nuevos valores de entrada, en el
flujo más caliente\n"
"en operación, ya que este flujo esta dentro
del domo");
Tsco=0.0;
Tsfo=0.0;
return;
}
else if(x1==100)
{
zona[flujo_caliente]=vapor;
}
}

```

```

        Heco=Entalpia_PT(Peco,Teco);
    }
    else if(x1==100)
    {
        zona[flujo_caliente]=agua_caliente;
        Heco=Entalpia_PT(Peco,Teco);
    }
    float x2=Calidad_PT(Pefo,Tefo);
    if(x2==0.0)
    {
        zona[flujo_frío]=agua_domo;
        AfxMessageBox("Entre nuevos valores de entrada, en el flujo más
frío\n"
                    "en operación o entre la calidad, es agua
dentro del domo");
        Tsco=0.0;
        Tsfo=0.0;
        return;
    }
    else if(x2==100)
    {
        zona[flujo_frío]=vapor;
        Hefo=Entalpia_PT(Pefo,Tefo);
    }
    else if(x2==100)
    {
        zona[flujo_frío]=agua_caliente;
        Hefo=Entalpia_PT(Pefo,Tefo);
    }
    float Q0=0.0;
    float Q1=0.0;
    float Tsu=(Teco+Tefo)/2;
    float LMTD_real=0.0;
    int contador1=0;
    int contador2=0;
    double limite1=0.0;
    double limite2=0.0;
    float Tx=0.0;
    float Ty=0.0;
    float Tsm=(Teco+Tefo)/2;
    float Tlimite_inferior=Tsm-10;
    float Tlimite_superior=Tsm+10;
    float Qx=0.0;

    //////////// Cálculos por zona
    if(zona[flujo_caliente]==vapor && zona[flujo_frío]==vapor)
    {
        float valor=Teco-Tefo;
        if(valor<120)
        {AfxMessageBox("Con los valores dados no se pueden hacer
cálculos");
            Tsfo=0.0;
            Tsco=0.0;
        }
        return;
    }
    float Ts_vapor_flujo_caliente_salida=

    Buscar_Temp_saturación(Psco);
    ///calculat temps

```

```

do{
    if(Tsco==0.0)
    {
        Tx=Tlimite_superior;
        Hsco=Entalpia_PT(Psco,Tx);
    }
    else
    {
        Tx=Tsco;
        Hsco=Entalpia_PT(Psco,Tx);
    }
    Q0=mco*(Heco-Hsco);
    limite2=10;
    do{
        if(Tsfo==0.0)
        {
            Ty=Tlimite_inferior;
        }
        else
        {
            Ty=Tsfo;
        }
        LMTD_real=Lmtd_flujo_paralelo(Teco,Tefo,Tx,Ty);
        Q1=U_real*Area_total_transferencia_calor*LMTD_real;
        if(Q1<Q0)
        {
            Hsfo=Hefo+(Q1/mfo);
            Tsfo=Temp_PH(Psfo,Hsfo);
        }
        else
        {
            Hsfo=Hefo+(Q0/mfo);
            Tsfo=Temp_PH(Psfo,Hsfo);
        }
        if(Tsfo>Tlimite_inferior)
        {
            Tlimite_inferior=Tlimite_inferior-5;
        }

        Tsfo=Tlimite_inferior;

        Hsfo=Entalpia_PT(Psfo,Tsfo);
        Q1=mfo*(Hsfo-Hefo);
    }

    contador1++;
    limite2=fabs(Ty-Tsfo);
    if(contador1>10)
    {
        limite2=1;
    }
}while(limite2>5);
Hsco=Heco-(Q1/mco);
Tsco=Temp_PH(Psco,Hsco);
if(Tsco<Tlimite_superior)
{
    Tlimite_superior=Tlimite_superior+5;
    Tsco=Tlimite_superior;
}
contador2++;
limite1=fabs(Tx-Tsco);
if(contador2>10)
{
    limite1=1;
}

```

```

}while(limitel>5);

if(Tsco<=Ts_vapor_flujo_caliente_salida)
{
    Tsco=Ts_vapor_flujo_caliente_salida;
    Hsco=Entalpia_PT_vaporsaturado(Psco,Tsco);
    do{
        if(Tsfo==0.0)
        {
            Ty=Tefo+5;
        }
        else
        {
            Ty=Tsfo;
        }
        float LMTD=Lmtd_flujo_paralelo(Teco,Tefo,Tsco,Ty);
        Q1=U_real*Area_total_transferencia_calor*LMTD;
        if(Q1<Q0)
        {
            Hsfo=Hefo+(Q1/mfo);
            Tsfo=Temp_PH(Psfo,Hsfo);
        }
        else
        {
            Hsfo=Hefo+(Q0/mfo);
            Tsfo=Temp_PH(Psfo,Hsfo);
        }
        limite2=fabs(Ty-Tsfo);
    }while(limite2>10);
}
prop_real[FLUJO_CALIENTE_SALIENDO].temp=Tsco;
prop_real[FLUJO_FRÍO_SALIENDO].temp=Tsfo;
return;
}

if(zona[flujo_caliente]==vapor                                &&
zona[flujo_frío]==agua_caliente)
{
    ///calcular temps
    float Temp_saturación_flujo_frío_salida=

Buscar_Temp_saturación(Psfo);
    float Temp_saturación_flujo_caliente_entrada=

Buscar_Temp_saturación(Peco);
    float Temp_saturación_flujo_caliente_salida=

Buscar_Temp_saturación(Psco);
    if(Tefo>Temp_saturación_flujo_frío_salida)
    {AfxMessageBox("Con los valores dados del flujo frio en
operación no se pueden hacer cálculos.....\n"
    "cambie el valor de la presión de salida,
aumentandola o disminuya la temperatura de entrada");
    /////el valor de Tefo es mayor que la temp de saturación a la
//presión de salida
    Tsfo=0.0;
}

```

```

        Tsco=0.0;
        return;
    }

    float valor=Teco-Tefo;
    if(valor<35)
    {AfxMessageBox("Con los valores dados no se pueden hacer
cálculos");
        Tsfo=0.0;
        Tsco=0.0;
        return;
    }
    float Tx=0.0;
    float Ty=0.0;

    ///calculat temps
    do{
        if(Tsco==0.0)
        {
            Tx=Temp_saturación_flujo_caliente_entrada;
            Hsco=Entalpia_PT_vaporsaturado(Psco,Tx);
        }
        else
        {
            Tx=Tsco;
            Hsco=Entalpia_PT(Psco,Tx);
        }
        float Q0=0.0;
            float Q1=0.0;
        Q0=mco*(Heco-Hsco);
        limite2=10;
        do{
            if(Tsfo==0.0)
            {
                Ty=Temp_saturación_flujo_frio_salida;
            }
            else
            {
                Ty=Tsfo;
            }
            float LMTD=

Lmtd_flujo_paralelo(Teco,Tefo,Tx,Ty);

Q1=U_real*Area_total_transferencia_calor*LMTD;
            if(Q1<Q0)
            {

Hsfo=Hefo+(Q1/mfo);

Tsfo=Temp_PH(Psfo,Hsfo);
            }
            else
            {
                Hsfo=Hefo+(Q0/mfo);
                Tsfo=Temp_PH(Psfo,Hsfo);
            }
            limite2=fabs(Ty-Tsfo);
            if(Tsfo>=Temp_saturación_flujo_frio_salida)

```

```

        {
Tsfo=Temp_saturación_flujo_frío_salida;
        limite2=3;
        }

        if(Tsfo>Tlimite_inferior)
        {
            Tlimite_inferior=Tlimite_inferior-5;
            Tsfo=Tlimite_inferior;
            Hsfo=Entalpia_PT(Psfo,Tsfo);
            Q1=mfo*(Hsfo-Hefo);
        }
    }while(limite2>5);
Hsco=Heco-(Q1/mco);
TSCO=Temp_PH(Psco,Hsco);
if(TSCO<Tlimite_superior)
{
    Tlimite_superior=Tlimite_superior+5;
    TSCO=Tlimite_superior;
}
limite1=fabs(Tx-TSCO);
}while(limite1>5);
if(TSCO<=Temp_saturación_flujo_caliente_salida)
{
    TSCO=Temp_saturación_flujo_caliente_salida;

    Hsco=Entalpia_PT_vaporsaturado(Psco,TSCO);
    Q0=mco*(Heco-Hsco);
    do{
        if(Tsfo==0.0)
        {
            Ty=Tefo+5;
        }
        else
        {
            Ty=Tsfo;
        }
        float LMTD=

Lmtd_flujo_paralelo(Teco,Tefo,TSCO,Ty);
        Q1=U_real*Area_total_transferencia_calor*LMTD;
        if(Q1<Q0)
        {
            Hsfo=Hefo+(Q1/mfo);
            Tsfo=Temp_PH(Psfo,Hsfo);
        }
        else
        {
            Hsfo=Hefo+(Q0/mfo);
            Tsfo=Temp_PH(Psfo,Hsfo);
        }
        limite2=fabs(Ty-Tsfo);
    }while(limite2>5);

prop_real[FLUJO_CALIENTE_SALIENDO].temp=TSCO;
prop_real[FLUJO_FRÍO_SALIENDO].temp=Tsfo;
return;

```

```

    }

    prop_real[FLUJO_CALIENTE_SALIENDO].temp=TSCO;
    prop_real[FLUJO_FRÍO_SALIENDO].temp=TSFO;

    return;
}

if(zona[flujo_caliente]==agua_caliente &&
zona[flujo_frío]==agua_caliente)
{
    //calcular temps
    float Temp_saturación_flujo_frío_salida=

    Buscar_Temp_saturación(Psfo);
    float Temp_saturación_flujo_caliente_entrada=

    Buscar_Temp_saturación(Peco);
    float Temp_saturación_flujo_caliente_salida=

    Buscar_Temp_saturación(Psco);
    if(Tefo>Temp_saturación_flujo_frío_salida)
        {AfxMessageBox("Con los valores dados del flujo frío en
operación no se pueden hacer cálculos");
        //el valor de Tefo es mayor que la temp de saturación a la
//presión de salida
            Tsfo=0.0;
            TSCO=0.0;
            return;
        }
    float valor=Teco-Tefo;
    if(valor<35)
        { AfxMessageBox("Con los valores dados no se pueden
hacer cálculos");
            Tsfo=0.0;
            TSCO=0.0;
            return;
        }
    float Tx=0.0;
    float Ty=0.0;
    float
Ts_vapor_flujo_frío_salida=Buscar_Temp_saturación(Psfo);
    //calcular temps
    do{
        if(TSCO==0.0)
        {
            Tx=Teco-10;
            HSCO=Entalpia_PT_vaporsaturado(Psco,Tx);
        }
        else
        {
            Tx=TSCO;
            HSCO=Entalpia_PT(Psco,Tx);
        }
        float Q0=0.0;
        float Q1=0.0;

```

```

Q0=mco*(Heco-Hsco);
limite2=10;
do{  if(Tsfo==0.0)
    {    Ty=Tefo+5;
    }
    else
    {    Ty=Tsfo;
    }

    float LMTD=

Lmtd_flujo_paralelo(Teco,Tefo,Tx,Ty);
Q1=U_real*Area_total_transferencia_calor*LMTD;
if(Q1<Q0)
{    Hsfo=Hefo+(Q1/mfo);
    Tsfo=Temp_PH(Psfo,Hsfo);
}
else
{    Hsfo=Hefo+(Q0/mfo);
    Tsfo=Temp_PH(Psfo,Hsfo);
}
if(Tsfo>Tlimite_inferior)
{    Tlimite_inferior=Tlimite_inferior-5;
    Tsfo=Tlimite_inferior;
    Hsfo=Entalpia_PT(Psfo,Tsfo);
    Q1=mfo*(Hsfo-Hefo);
}
limite2=fabs(Ty-Tsfo);
}while(limite2>5);
Hsco=Heco-(Q1/mco);
Tsco=Temp_PH(Psco,Hsco);
if(Tsco<Tlimite_superior)
{    Tlimite_superior=Tlimite_superior+5;
    Tsco=Tlimite_superior;
}
limite1=fabs(Tx-Tsco);
}while(limite1>5);
if(Tsfo>=Ts_vapor_flujo_frio_salida)
{    Tsfo=Ts_vapor_flujo_frio_salida;
    Hsfo=Entalpia_PT(Psfo,Tsfo);
    Q0=mfo*(Hsfo-Hefo);
do{
    if(Tsco==0.0)
    {    Tx=Teco+5;
    }

    else
    {    Tx=Tsco;
    }
    float LMTD=

Lmtd_flujo_paralelo(Teco,Tefo,Tx,Tsfo);

Q1=U_real*Area_total_transferencia_calor*LMTD;

```

```

        if (Q1<Q0)
        {
            Hsfo=Hefo+(Q1/mfo);
            Tsfo=Temp_PH(Psfo,Hsfo);
        }
        else
        {
            Hsfo=Hefo+(Q0/mfo);
            Tsfo=Temp_PH(Psfo,Hsfo);
        }
        limite2=fabs(Ty-Tsfo);
    }while(limite2>5);
    }
    prop_real[FLUJO_CALIENTE_SALIENDO].temp=TSCO;
    prop_real[FLUJO_FRÍO_SALIENDO].temp=Tsfo;
    return;
}
    AfxMessageBox("No se hacen cálculos con los valores dados");
    prop_real[FLUJO_CALIENTE_SALIENDO].temp=TSCO;
    prop_real[FLUJO_FRÍO_SALIENDO].temp=Tsfo;
}
int CInter::asignar_tipo_flujo(bool tipoflujo)
{
    TipoFlujo=tipoflujo;
    return 1;
}

```

Anexo G. Archivos fuentes del programa Caldera_Carga_Parcial.exe

Aunque en el programa Caldera_Carga_Parcial.exe se realizan los mismos cálculos a carga parcial que en los programas del intercambiador y condensador a carga parcial, los fluidos que intervienen en la transferencia de calor son diferentes, por tal motivo fue necesario construir otra clase que realizara los cálculos en los componentes de la caldera. La diferencia de esta nueva clase es que tiene la capacidad de calcular las propiedades termodinámicas tanto del agua como del aire en condiciones reales y las de los gases de combustión, que en la caldera es el fluido caliente.

1. InterGasAguaAire.h.

/* El archivo InterGasAguaAire.h es el archivo de declaración de la clase CInterGasAguaAire.*/

```
// InterGasAguaAire.h: interface for the CInterGasAguaAire class.

#if
!defined(AFX_INTERGASAGUAAIRE_H__6770869C_E771_401B_BDAE_FD753B5D15BE__INCLUDED_)
#define
AFX_INTERGASAGUAAIRE_H__6770869C_E771_401B_BDAE_FD753B5D15BE__INCLUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

#include "Caldera_Carga_Parcial.h"

////defino valores para los componentes///
#define HOGAR 1
#define SOBRECALENTADOR 2
#define RECALENTADOR 3
#define ECONOMIZADOR 4
#define CALENTADOR_AIRE 5

// defino 4 puntos de los flujos en el componente del intercambiador
#define flujocaliente_entrada 2
#define flujocaliente_salida 3
#define flujofrío_entrada 0
#define flujofrío_salida 1
#define total_flujos 4 ///numero total de flujos
```

```

// defino los valores para identificar a que lado del
//intercambiador me refiero en otras variables
#define lado_frío 0
#define lado_caliente 1

/* Estructura que guarda valores termodinámicos necesarios para los
cálculos por cada punto del intercambiador*/
typedef struct{
    float temp;
    float presión;
    float flujo;
}termodatos;

/* Estructura que guarda valores mecánicos necesarios para los
cálculos del intercambiador*/
typedef struct{
    float Ud;
    BOOL calcular_Ud;
    float área_transversal_tubo;
    float Área_total_transferencia_calor;
}datosmecánicos;

/* Estructura que guarda valores de los exponentes que el usuario
decide, para analizar el comportamiento del intercambiador en cuanto
el coeficiente de transferencia de calor universal y las perdidas de
presión en él*/
typedef struct{
    float a0,a1,a2,b0,b1,b2,c0,c1,c2;
}coeficientes_comportamiento;

/* defino valores para saber a que flujo me refiero dentro del
intercambiador*/
#define flujo_caliente 0
#define flujo_frío 1

/* Se incluye funciones de la librería dinámica prop_gases.dll para
buscar ciertas propiedades termodinámicas de los gases de
combustión y del aire real*/
extern "C" __declspec(dllimport) double Calor_especifico_aire(double
p,double t,double humedad);
extern "C" __declspec(dllimport) double Entalpía_PT_Aire(double
p,double t,double humedad);
extern "C" __declspec(dllimport) double Temp_PH_Aire(double p,double
h,double humedad);
extern "C" __declspec(dllimport) double Temp_PH_Gas(double p,double
h,porcentaje_gas pg);
extern "C" __declspec(dllimport) double Entalpía_PT_Gas(double
p,double t,porcentaje_gas pg);

class CInterGasAguaAire /* declaración de la clase que realiza los
cálculos de intercambiador por cada componente de la caldera*/
{
public:

```

```

        CString texto; //guarda cualquier texto proporcionado en los
//cálculos
        float presión_salida_real_lado_frío(); //devuelve la presión
//de salida de operación del fluido frío
        float presión_salida_real_lado_caliente(); // devuelve la
//presión de salida de operación del fluido caliente
        int componente; //define el componente que se va a calcular
        int Calcular(); // función general para hacer cálculos de
//intercambiador en la caldera
        float perdidas_presión_diseno_lado_frío(); // función que
//devuelve el valor de la pérdida de presión del lado frío o flujo
//frío en condiciones de diseño
        float perdidas_presión_diseno_lado_caliente(); // función que
//devuelve el valor de la pérdida de presión del lado caliente o
//flujo caliente pero en condiciones de diseño
        float Temp_salida_lado_caliente(); // función que devuelve el
//valor de la temperatura de salida del flujo mas caliente en
//condiciones de operación
        float Temp_salida_lado_frío(); // función que devuelve el
//valor de la temperatura de salida del flujo mas frío en
//condiciones de operación
        float Delta_presión_lado_caliente(); // función que devuelve
//el valor de la pérdida de presión del lado del flujo mas caliente
//en condiciones de operación
        float Delta_presión_lado_frío(); // función que devuelve el
//valor de la pérdida de presión del lado del flujo mas frío en
//condiciones de operación
        void entrar_coeficientes_comportamiento(
exponentes_comportamiento cp); // función que asigna los valores a
//los exponentes para analizar el comportamiento del componente
        void asignar_datos_mecánicos(datosmecánicos dm); // función
//que asigna los valores mecánicos del componente
        int asignar_porcentaje_gas(porcentaje_gas pg); // función que
//asigna los valores el porcentaje de los gases de combustión
        CInterGasAguaAire(); // constructor normal
        CInterGasAguaAire(termodatos pd[4],termodatos pr[4])
/* constructor de trabajo se asigna las propiedades termodinámicas
a cada flujo en condiciones de diseño y de operación que el usuario
a querido entrar*/
{prop_diseno[flujocaliente_entrada].flujo=pd[flujocaliente_entrada].
flujo;
prop_diseno[flujocaliente_entrada].presión=pd[flujocaliente_entrada]
.presión;
prop_diseno[flujocaliente_entrada].temp=pd[flujocaliente_entrada].te
mp;
prop_diseno[flujocaliente_salida].flujo=pd[flujocaliente_salida].flu
jo;
prop_diseno[flujocaliente_salida].presión=pd[flujocaliente_salida].p
resión;
prop_diseno[flujocaliente_salida].temp=pd[flujocaliente_salida].temp
;
prop_real[flujocaliente_entrada].flujo=pr[flujocaliente_entrada].flu
jo;

```

```

prop_real[flujocaliente_entrada].presión=pr[flujocaliente_entrada].p
resión;
prop_real[flujocaliente_entrada].temp=pr[flujocaliente_entrada].temp
;

///// las siguientes propiedades se calculan
prop_real[flujocaliente_salida].flujo=pr[flujocaliente_salida].flujo
=0.0;
prop_real[flujocaliente_salida].presión=pr[flujocaliente_salida].pre
sión=0.0;
prop_real[flujocaliente_salida].temp=pr[flujocaliente_salida].temp=0
.0;

prop_diseno[flujofrío_entrada].flujo=pd[flujofrío_entrada].flujo;
prop_diseno[flujofrío_entrada].presión=pd[flujofrío_entrada].presión
;
prop_diseno[flujofrío_entrada].temp=pd[flujofrío_entrada].temp;

prop_diseno[flujofrío_salida].flujo=pd[flujofrío_salida].flujo;
prop_diseno[flujofrío_salida].presión=pd[flujofrío_salida].presión;
prop_diseno[flujofrío_salida].temp=pd[flujofrío_salida].temp;

prop_real[flujofrío_entrada].flujo=pr[flujofrío_entrada].flujo;
prop_real[flujofrío_entrada].presión=pr[flujofrío_entrada].presión;
prop_real[flujofrío_entrada].temp=pr[flujofrío_entrada].temp;

///// las siguientes propiedades se calculan
prop_real[flujofrío_salida].flujo=pr[flujofrío_salida].flujo=0.0;
prop_real[flujofrío_salida].presión=pr[flujofrío_salida].presión=0.0
;
prop_real[flujofrío_salida].temp=pr[flujofrío_salida].temp=0.0;
/// se inicializa a ceros los porcentaje de los gases y la función
//de asignación de los porcentajes les da los valores que el usuario
//entro al programa
    pm.CO2c=0.0;
    pm.COc=0.0;
    pm.SO2c=0.0;
    pm.SOxc=0.0;
    pm.NOxc=0.0;
    pm.H2Oc=0.0;
    pm.O2c=0.0;
    pm.N2c=0.0;
    pm.H2c=0.0;

    texto="";
}
virtual ~CInterGasAguaAire();// destructor
private:
    int cálculos_equipos(); //hace cálculos de intercambiador de
//contracorriente para los componente de: sobrecalentador,
//economizador, recalentador y algun otro tipo
    int cálculos_calentador(); // hace cálculos de intercambiador
//de contracorriente para el calentador de aire

```

```

    int cálculos_hogar(); // hace cálculos de intercambiador de
//contracorriente para el hogar
    porcentaje_gas pm; // variable que guarda los porcentajes de
//los gases de combustión
    float Temp_media( float t1, float t2); // función que calcula
//la temperatura media
    float Lmtd_flujo_ebullendo(float Tec, float Tsc, float Ts);
// función que calcula la LMTD para tipo de intercambiador donde un
//flujo cambia de fase
    float Lmtd_contraflujo(float Tec,float Tef, float Tsc, float
Tsf); // función que calcula la LMTD para tipo de intercambiador
//de contracorriente
    float presión_media(float p1,float p2); // función que calcula
//la presión media
    float perdidas_presión_lado_caliente(); // función que calcula
//las perdidas de presión del lado del flujo mas caliente en
//condiciones de operación
    float elevar(float x, float y); // función para elevar un
//numero a otro
    float perdidas_presión_lado_frío(); // función que calcula
//las perdidas de
// presión del lado del flujo mas frío en condiciones de
//operación
    int calculo_presión_salida(); // función que calcula la
//presión de salida
// de ambos lados (caliente y frío) en condiciones de operación
    float Buscar_Temp_saturación(float p); //función que calcula
//la temperatura de saturación del agua a una presión dada.
    float Temp_PH(float p, float h); //función que calcula la
//temperatura del agua a una presión y entalpía dada.
    float Entalpía_PT(float p, float t); //función que calcula la
//entalpía del agua a una presión y temperatura dada.
    int y; // variable para comprobar si fueron asignados los
//valores de los exponentes de análisis
    float a0,a1,a2,b0,b1,b2,c0,c1,c2,d0; // exponentes de
//análisis
    int x; // variable para comprobar si fueron asignados los
//valores mecánicos
    float Ud; // coeficiente universal de transferencia de calor
    BOOL calcular_Ud; // variable para decidir si se calcula el
//coeficiente o fue asignado por el usuario
    float Área_total_transferencia_calor; // área total de
//transferencia de calor
    float perdidas_presión_real[2]; // arreglo que guarda las
//perdidas de presión de ambos lados (caliente y frío) en
//condiciones de operación
    float perdidas_presión_diseno[2]; // arreglo que guarda las
//perdidas de presión de ambos lados (caliente y frío) en
//condiciones de diseño
    float LMTD_diseno; // LMTD de diseño
    termodatos_prop_diseno[total_flujos]; // arreglo que guarda
//los valores termodinámicos de cada uno de los cuatro puntos de los
//flujos dentro del intercambiador en condiciones de diseño

```

```

        termodatos_prop_real[total_flujos]; // arreglo que guarda
//los valores termodinámicos de cada uno de los cuatro puntos de los
//flujos dentro del intercambiador en condiciones de operación.
        float calculo_Cp(float p, float t); // función que calcula
//el calor específico del agua o del aire dependiendo del componente
//que se este analizando.
        float calculo_Ud(); // función que calcula y devuelve el valor
//del coeficiente de transferencia calor universal en condiciones
//de diseño
        float calor_máximo_diseno(); // función que calcula y devuelve
//el valor del calor máximo en condiciones de diseño.
        float calculo_Uo(); // función que calcula y devuelve el valor
//del coeficiente de transferencia calor universal en condiciones
//de operación.
};

#endif //
#ifndef(AFX_INTERGASAGUAIRE_H__6770869C_E771_401B_BDAE_FD753B5D15B
E__INCLUDED_)

```

2. InterGasAguaAire.cpp.

/*El archivo InterGasAguaAire.cpp es el archivo de definición de la clase CInterGasAguaAire.*/

```

//InterGasAguaAire.cpp: implementation of the
CInterGasAguaAireGasAguaAire class.
//
#include "stdafx.h"
#include "Caldera_Carga_Parcial.h"
#include "InterGasAguaAire.h"
#include "Propiedad.h"

#include <math.h>

#ifdef _DEBUG
#undef THIS_FILE
static char THIS_FILE[]=__FILE__;
#define new DEBUG_NEW
#endif

////////////////////////////////////
//
// Construction/Destruction
////////////////////////////////////
//

CInterGasAguaAire::CInterGasAguaAire()
{ // constructor
}

```

```

CInterGasAguaAire::~CInterGasAguaAire()
{
    // destructor
}

int CInterGasAguaAire::asignar_porcentaje_gas(porcentaje_gas pg)
{
    // Asigna valores de los porcentaje de los gases de combustión
    pm.COc=pg.COc;
    pm.CO2c=pg.CO2c;
    pm.H2c=pg.H2c;
    pm.H2Oc=pg.H2Oc;
    pm.N2c=pg.N2c;
    pm.NOxc=pg.NOxc;
    pm.O2c=pg.O2c;
    pm.SO2c=pg.SO2c;
    pm.SOxc=pg.SOxc;
    pm.Habsc=pg.Habsc;
    return 1;
}

int CInterGasAguaAire::Calcular()
{
    // Realiza los cálculos de intercambiador de calor dependiendo
    //del componente elegido por el usuario del programa
    switch(componente)
    {
        case HOGAR:
            return cálculos_hogar();
            break;
        case CALENTADOR_AIRE:
            return cálculos_calentador();
            break;
        default :
            return cálculos_equipos();
            break;
    }
    return 1;
}

int CInterGasAguaAire::cálculos_hogar()
{
    // Realiza los cálculos de intercambiador de calor
    // de acuerdo al informe general del proyecto cliope
    // realizado por Adriana Lara para el hogar
    // Aquí el fluido caliente son los gases de combustión
    // y el fluido frío es agua
    CString titulo;
    titulo.Format("HOGAR");
    CString mensaje;
    if(!calcula_presión_salida())
    {
        return 0;
    }
    // Se calcula las presiones de salida en condiciones de operación
    float U_real=calcula_Uo(); //se calcula el coeficiente de
    //transferencia de calor en condiciones de operación se asignan los
    //valores de operación de los flujos de entrada
    float Teco=prop_real[flujocaliente_entrada].temp;
}

```

```

float Tefo=prop_real[flujofrío_entrada].temp;
float Peco=prop_real[flujocaliente_entrada].presión;
float Pefo=prop_real[flujofrío_entrada].presión;
float Psco=prop_real[flujocaliente_salida].presión;
float Psfo=prop_real[flujofrío_salida].presión;
float mco=prop_real[flujocaliente_entrada].flujo;
float mfo=prop_real[flujofrío_entrada].flujo;

float Tsat_efo=Buscar_Temp_saturación(Pefo);
// Temperatura de saturación del agua a la presión de entrada
float Tsat_sfo=Buscar_Temp_saturación(Psfo);
// temperatura de saturación del agua a la presión de salida
CString mens1;
mens1.Format("La temperatura del agua de entrada al hogar es
%.2f F debe estar por debajo\n"
"de la temperatura de saturación %.2f F a la presión de entrada;
estamos en el Hogar!!",Tefo,Tsat_efo);

if(Tsat_efo<Tefo)
{
// la temperatura de entrada a el hogar debe estar por debajo
// de la de saturación a la presión de entrada
MessageBox(NULL,mens1,titulo,MB_ICONERROR);
return 0;
}

// como estamos haciendo cálculos en el hogar el agua debe salir
// como vapor saturado o con calidad 1, por eso la temperatura
// de salida del agua(flujo frío) es la temperatura de saturación
// a la presión de salida
float Tsco=0.0;
float Tsfo=Buscar_Temp_saturación(Psfo); // temperatura de
saturación
// a la presión de salida
float Hefo=Entalpía_PT(Pefo,Tefo); // se calcula la entalpía del
//agua en las condiciones de entrada
float Heco=Entalpía_PT_Gas(Peco,Teco,pm); // se calcula la
//entalpía de los gases en las condiciones de entrada
mens1.Format("A la temperatura dada de entrada de los \n"
" gases de combustión en el hogar %.2f F\n"
"no se puede hallar la entalpía ", Teco);
CString titulo1;
titulo1.Format(" HOGAR...La Entalpía es= %.2f Btu/lbm",Heco);
if(Heco<=0)
{
MessageBox(NULL,mens1,titulo1,MB_ICONERROR);
return 1;
}
float Hsco=0.0; //Entalpía de los gases a las condiciones de
//salida de operación se calcula
CString mens2;
mens2.Format("La Energía del agua de entrada al hogar es %.2f
Btu/lbm\n"
"mayor que la energia de los gases de combustión %.2f
Btu/lbm\n"
"Aumente la temperatura de entrada de los gases\n")

```

```

        "No se realiza ninguna transferencia de
calor", Hefo, Heco);
        if(Heco<Hefo)
            { // me aseguro que el agua de entrada no tenga mas energia que
//los gases
                MessageBox(NULL, mens2, titulo, MB_ICONERROR);
                return 2;
            }
        CString mens3;
        float Area=Area_total_transferencia_calor;
        float limite=0.0; // defino esta variable para saber
//cuando para la siguiente iteración
        float limite2=0.0;
        float Ty=0.0; // temperatura del fluido caliente de salida en
//condiciones de operación supuesta
        float Tx=0.0;
        float Qx=0.0; // calor transferido
        float LMTD=0.0;
        float ts=(Teco+Tefo)/2;
        int contador=1;
        float Hsfo=0.0;
        do{ if(Tsfo==0)
            { // supongo temperatura
                Tx=ts;
            }
            else
            { // asigno temperatura calculada
                Tx=Tsfo;
            }
        }do{ if(Tsco==0)
            { // se supone una temperatura inicial
                Ty=(Tsfo*0.5)+Tsfo;
            }
            else
            { // se toma la temperatura antes calculada
                Ty=Tsco;
            }
        }
        LMTD=Lmtd_contraflujo(Teco, Tefo, Ty, Tx); // se calcula la
LMTD
        if(LMTD==0)
            { //reportarlo en el archivo de texto
                return 3;
            }
        Qx=U_real*Area*LMTD; // se calcula el calor
        Hsco=Heco-(Qx/mco); // con el calor se halla la
//entalpía a la salida
        Tsco=Temp_PH_Gas(Psco, Hsco, pm); // y con esa entalpía
//con la presión de salida se halla la temperatura de salida de los
//gases de combustión
        if(Tsco<=Tsfo)
            { //reportar en el archivo de texto
                mens3.Format("Suba la temperatura de entrada de
los gases %.2f F en el Hogar\n"
                "La temperaruta de salida de los gases %.2f F \n"

```

```

        "es menor que la salida del agua %.2f F\n"
        "No hay transferencia de calor",Teco,Tsco,Tsfo);
        MessageBox(NULL,mens3,titulo,MB_ICONINFORMATION);
        return 4;
    }
    limite2=fabs(Ty-Tsco);        // se compara las
temperatura supuestas y calculadas
    if(contador>20)
    { limite2=1;
    }
}while(limite2>5);    // se decide si seguir o no con la
//iteración
Hsfo=Hefo+(Qx/mfo);    // se halla la entalpía de salida
//del agua
Tsfo=Temp_PH(Psfo,Hsfo);    //se calcula la temperatura de
//salida
limite=fabs(Tx-Tsfo);    // se compara las temperaturas
contador++;
if(contador>20)
{ limite=1;
}
// se saca la diferencia en valor absoluto entre la
//temperatura supuesta y la calculada
}while(limite>5);    // se mira la aproximacion entre las dos
if(Tsfo<=Tefo)
{
    Tsfo=Tsfo+1;
    texto.Format("La temperatura de salida %.2f F es menor
que la entrada %.2f F porque\n"
        "el agua entra a la temperatura de saturación a la
presión de entrada de %.2f psia\n"
        "y como hay perdidas de presión en la caldera el
agua sale como vapor\n"
        "con calidad 1 pero a la presión de salida de %.2f
psia",Tsfo,Tefo,Pefo,Psfo);
}

// se asigna los valores encontrados a las variables respectivas
prop_real[flujocaliente_salida].temp=Tsco;
prop_real[flujofrío_salida].temp=Tsfo;
return 5;
}

int CInterGasAguaAire::cálculos_calentador()
{
    //En esta funcion se realizan los cálculos de
    // intercambiador de calor en el calentador del aire
    // Aqui el fluido frío es el aire y el caliente son
    //los gases de combustión
    if(!calcula_presión_salida())
    {
        return 0;
    }
    // se calcula las presiones de salida en condiciones de
//operación se calculan las presiones de salida en condiciones de
//operación
    float U_real=calcula_Uo();    // se calcula el coeficiente de
//transferencia de calor universal en condiciones de operaciones

```

```

/*asignan los valores termodinamicos de entrada en condiciones de
operación a variables locales*/
float Teco=prop_real[flujocaliente_entrada].temp;
float Tefo=prop_real[flujofrío_entrada].temp;
float Peco=prop_real[flujocaliente_entrada].presión;
float Pefo=prop_real[flujofrío_entrada].presión;
float Psco=prop_real[flujocaliente_salida].presión;
float Psfo=prop_real[flujofrío_salida].presión;
float mco=prop_real[flujocaliente_entrada].flujo;
float mfo=prop_real[flujofrío_entrada].flujo;

/*se define variables locales para ser calculadas las
temperaturas de salida en condiciones de operación*/
float Tsco=0.0;
float Tsfo=0.0;
float Hsco=0.0;
float Hsfo=0.0;
CString titulo;
titulo.Format("CALENTADOR DE AIRE");
CString message;
if(Psfo<=1)
{message.Format("La presión de salida del aire calculada\n"
"es = %.2f psia, por lo tanto no se puede seguir
los cálculos",Psfo);
MessageBox(NULL,message,titulo,MB_ICONERROR);
return 0;
}
float Hefo=Entalpía_PT_Aire(Pefo,Tefo,pm.Habsc); //calculo entalpía
//del aire en las condiciones de entrada

message.Format("La entalpía del aire a la entrada no se puede
calcular\n"
"con esta temperatura %.2f F, aumentela por favor",Tefo);

if(Hefo<=0)
{
MessageBox(NULL,message,titulo,MB_ICONERROR);
return 0;
}
float Heco=Entalpía_PT_Gas(Peco,Teco,pm); // calculo entalpía
//de los gases en las condiciones de entrada

if(Heco<=0)
{ message.Format("La entalpía de los gases a la entrada no se
puede calcular\n"
"con esta temperatura de %.2f F, aumentela por
favor",Teco);
MessageBox(NULL,message,titulo,MB_ICONERROR);
return 0;
}
message.Format("El Aire tiene más energia que el gas en la
entrada:\n"
"Entalpía del gas= %.3f Btu/lbm,\n"
"Entalpía del agua= %.3f Btu/lbm, \n"

```

```

        "por lo tanto no se pueden hacer cálculos",Heco,Hefo);
if(Hefo>Heco)//reportarlo en el archivo de texto como resultado
{ // me aseguro que el aire no tenga mayor energia que los gases
    MessageBox(NULL,message,titulo,MB_ICONERROR);
    return 1;
}
//establesco unas variables para hacer calculo iterativos
float Q0=0.0;
float Q1=0.0;
float Area=Area_total_transferencia_calor;
float limite=0.0;
float limite2=0.0;
float Ty=0.0;
float Tx=0.0;
float Qx=0.0;
float LMTD=0.0;
float ts=(Tefo+Teco)/2;

//////////comienzo la iteración
do{
    if(Tsfo==0)
    { // supongo temperatura inicial
        Tx=ts;
    }
    else
    { // introduzco temperatura calculada
        Tx=Tsfo;
    }
    do{
        if(Tsco==0)
        { // supongo temperatura inicial
            Ty=(Tefo*0.5)+Tefo;
        }
        else
        { // introduzco temperatura calculada
            Ty=Tsco;
        }
        LMTD=Lmtd_contraflujo(Teco,Tefo,Ty,Tx);// calculo la LMTD
        // para un intercambiador de tipo contracorriente
        if(LMTD==0)
        { //reportarlo en el archivo de texto
            message.Format("LMTD = 0, Suba la temperatura de entrada
\n"
                " de los gases, actualmente es de %.2f
F",Teco);
            MessageBox(NULL,message,titulo,MB_ICONERROR);
            return 2;
        }
        Qx=U_real*Area*LMTD; // calculo el calor
        Hsco=Heco-(Qx/mco); // hallo la entalpía de salida de los
//gases
        Tsco=Temp_PH_Gas(Psco,Hsco,pm); // hallo la temperatura
//de los gases a la salida del intercambiador
        if(Tsco<Tefo)

```

```

        {message.Format("No se puede hacer una transferencia de
calor\n")
        "Aumente la temperatura de entrada de los gases que es
%.2f F\n"
"La temperatura de salida de los gases %.2f F es menor que la
temperatura \n"
        "del aire de entrada %.2f F",Teco,Tsco,Tefo);
        MessageBox(NULL,message,titulo,MB_ICONERROR);
        return 3;
    }
    limite2=fabs(Ty-Tsco); // comparo la temperatura supuesta
//con la calculada
    }while(limite2>5); // se decide si seguir iterando o
//no
Hsfo=Hefo+(Qx/mfo); // hallo la entalpía del aire a la salida en
//el intercambiador
Tsfo=Temp_PH_Aire(Psfo,Hsfo,pm.Habsc); // hallo la temperatura del
//aire a la salida
    limite=fabs(Tx-Tsfo); //comparo la temperatura supuesta con
//la calculada
    }while(limite>5); // se decide si seguir la iteración o no

    // se asigna los valores calculados de la temperatura a las
//propiedades respectivas
    prop_real[flujocaliente_salida].temp=Tsco;
    prop_real[flujofrío_salida].temp=Tsfo;
    return 4;
}

int CInterGasAguaAire::cálculos_equipos()
{ /* En esta función se realizan los cálculos de iteración para
establecer o hallar las temperaturas de salida en condiciones de
operación en los componentes de caldera como el sobrecalentador, el
recalentador, y el economizador y algún otro equipo que se pueda
analizar como un intercambiador de calor en contraflujo*/
    CString equipo;
    if(componente==SOBRECALENTADOR)
    {
        equipo.Format("SOBRECALENTADOR");
    }
    else if(componente==ECONOMIZADOR)
    {
        equipo.Format("ECONOMIZADOR");
    }
    else if(componente==RECALENTADOR)
    {
        equipo.Format("RECALENTADOR");
    }
    else
    {
        equipo.Format("EQUIPO NO DEFINIDO");
    }

    CString mensaje;
    if(!calcula_presión_salida())
    {
        return 0;
    } // se calcula las presiones de salida en condiciones de
//operación

```

```

float U_real=calculo_Uo(); // calculo el coeficiente de
//transferencia de calor en condiciones de diseño
// se asignan los valores termodinamicos de entrada en condiciones
// de operación a variables locales
float Teco=prop_real[flujocaliente_entrada].temp;
float Tefo=prop_real[flujofrío_entrada].temp;
float Peco=prop_real[flujocaliente_entrada].presión;
float Pefo=prop_real[flujofrío_entrada].presión;
float Psco=prop_real[flujocaliente_salida].presión;
float Psfo=prop_real[flujofrío_salida].presión;
float mco=prop_real[flujocaliente_entrada].flujo;
float mfo=prop_real[flujofrío_entrada].flujo;
float Tsa_efo=Buscar_Temp_saturación(Pefo); // se busca la
//temperatura de saturación del agua en las condiciones de entrada
if(componente!=ECONOMIZADOR)
{
    if(Tsa_efo>Tefo) //reportalo en el archivo de texto
//si ocurre esta posibilidad
    {
        mensaje.Format("El agua debe entrar al equipo
como minimo como vapor con calidad 1\n"
"Entre una temperatura del agua de entrada mayor que %.2f F\n"
"pues la temperatura de sautaracion es %.2f F",Tefo,Tsa_efo);
        MessageBox(NULL,mensaje,equipo,MB_ICONERROR);
        return 0;
    }
}

float Tsa_sfo=Buscar_Temp_saturación(Psfo); // se busca la
//temperatura de saturación del agua a la presión de salida
float Heco=Entalpía_PT_Gas(Peco,Teco,pm); // se calcula la
//entalpía de los gases de combustión a las condiciones de entrada
//se define variables para la iteración
float Tsco=0.0; // temperatura de salida del lado caliente en
//condiciones de operación
float Tsfo=0.0; // temperatura de salida del lado frío en
//condiciones de operación
float Hsco=0.0;
float Hsfo=0.0;
float Hefo=Entalpía_PT(Pefo,Tefo); // se calcula la entalpía
//del agua a las
// condiciones de entrada
mensaje.Format("La entalpía del agua a la entrada no se puede
calcular con esta temperatura %.2f F,\n"
" solo se puede calcular hasta 1400 F",Tefo);

if(Hefo<=0)
{
    MessageBox(NULL,mensaje,equipo,MB_ICONERROR);
    return 0;
}
if(Heco<=0)
{
    mensaje.Format("La entalpía de los gases a la entrada no se
puede calcular\n"
"con esta temperatura de %.2f F, aumentela por favor",Teco);
}

```

```

        MessageBox(NULL,message,equipo,MB_ICONERROR);
        return 0;
    }
    if(Hefo>Heco)    //reportarlo en el archivo de texto como
resultado
    {
        // me aseguro que el agua no tenga mas energia que los
gases
        message.Format("El agua tiene más energia que el gas en
la entrada:\n"
        "Entalpía del gas= %.2f Btu/lbm,\n"
        "Entalpía del agua= %.2f Btu/lbm, \n"
        "por lo tanto no hay tranferencia de calor hacia el
agua",Heco,Hefo);
        MessageBox(NULL,message,equipo,MB_ICONERROR);
        return 1;
    }
    float Q0=0.0;
    float Q1=0.0;
    float Area=Area_total_transferencia_calor;
    float limite=0.0;
    float limite2=0.0;
    float Ty=0.0;
    float Tx=0.0;
    float Qx=0.0;
    float LMTD=0.0;
    float ts=(Teco+Tefo)/2;
    int contador=1;
    //desde aqui empieza la iteración
    do{
        if(Tsfo==0)
        {
            // supongo temperatura
            Tx=ts;
        }
        else
        {
            // asigno temperatura calculada
            Tx=Tsfo;
        }

        do{
            if(Tsco==0)
            {
                // supongo temperatura
                Ty=Tefo+(Tefo*0.2);
            }
            else
            {
                // asigno temperatura calculada
                Ty=Tsco;
            }

            LMTD=Lmtd_contraflujo(Teco,Tefo,Ty,Tx); // se calcula la
LMTD

            Qx=LMTD*Area*U_real; // se calcula el calor
            Hsco=Heco-(Qx/mco); // se halla la entalpía de los
//gases a la salida
            Tsco=Temp_PH_Gas(Psco,Hsco,pm); // temperatura se salida
//de los gases
            if(Tsco<Tefo)

```

```

        {   mensaje.Format("No se puede hacer una
transferencia de calor\n"
    "Aumente la temperatura de entrada de los gases que es %.2f
F\n"
    "La temperatura de salida de los gases %.2f F es menor que la
temperatura \n"
    "del agua de entrada %.2f F",Teco,Tsco,Tefo);
    MessageBox(NULL,mensaje,equipo,MB_ICONERROR);
        return 3;
    }
    limite2=fabs(Ty-Tsco); // se compara las
//temperatura supuestas y calculadas
    if(contador>20)
    {   limite2=1;
    }
    }while(limite2>5); // se decide si seguir o no con la
//iteración
    Hsfo=Hefo+(Qx/mfo); // se halla la entalpía de salida
//del agua
    Tsfo=Temp_PH(Psfo,Hsfo); //se calcula la temperatura de
//salida
    limite=fabs(Tx-Tsfo); // se compara las temperaturas
    contador++;
    if(contador>20)
    {   limite=1;
    }

    }while(limite>5); // se decide si seguir o no la iteración

    float q=mfo*(Hsfo-Hefo);
    float qe=mco*(Heco-Hsco);
    float eficiencia=q/qe;
    /// se asigna los valores de la stemperaturas calculadas a las
//propiedades respectivas
    prop_real[flujocaliente_salida].temp=Tsco;
    prop_real[flujofrío_salida].temp=Tsfo;
    return 1;
}

void CInterGasAguaAire::asignar_datos_mecanicos(datosmecanicos dm)
{   // funcion que asigna los valores de las caracteristicas
mecanicas
    Ud=dm.Ud;
    calcular_Ud=dm.calcular_Ud;

    Area_total_transferencia_calor=dm.Area_total_transferencia_calor;
    perdidas_presión_real[lado_frío]=0.0;///se calcula
    perdidas_presión_diseño[lado_frío]=0.0;
    perdidas_presión_real[lado_caliente]=0.0;///se calcula
    perdidas_presión_diseño[lado_caliente]=0.0;
    x=1;
}

```

```

void
CInterGasAguaAire::entrar_coeficientes_comportamiento(exponentes_com
portamiento cp)
{
    // se asigna los valores de los exponentes para el analisis del
//comportamiento
        a0=cp.a0;
        a1=cp.a1;
        a2=cp.a2;
        b0=cp.b0;
        b1=cp.b1;
        b2=cp.b2;
        c0=cp.c0;
        c1=cp.c1;
        c2=cp.c2;
        y=1;
}

float CInterGasAguaAire::Temp_media( float t1, float t2)
{
    // se calcula la temperatura media
    float z=0.0;
    z=(t1+t2)/2;
    return z;
}

float CInterGasAguaAire::presión_media(float p1, float p2)
{
    // se halla la presión media
    float z=0.0;
    z=(p1+p2)/2;
    return z;
}

float CInterGasAguaAire::Lmtd_contraflujo(float Tec, float Tef,
float Tsc, float Tsf)
{
    float z=0.0;
    /// se calcula la LMTD de acuerdo al caso 3 de la figura 1 de
//la pagina 21-1 del capitulo 21 del libro de Babcock & Wilcox
    float a=(Tec-Tsf)-(Tsc-Tef);
    float b=log((Tec-Tsf)/(Tsc-Tef));
    z=a/b;
    return z;
}

float CInterGasAguaAire::Lmtd_flujo_ebullendo(float Tec, float Tsc,
float Ts)//puedo usar para gases
{
    float z=0.0;
    /// se calcula la LMTD de acuerdo al caso 1 de la figura 1
// de la pagina 21-1 del capitulo 21 del libro de Babcock & Wilcox
    float a=(Tec-Tsc);
    float b=log((Tec-Ts)/(Tsc-Ts));
    z=a/b;
    return z;
}

```

```

float CInterGasAguaAire::calculo_Cp(float p, float t)//solo sirve
para agua
{
    float z=0.0;
    // se calcula el calor especifico del aire o agua a unas
//condiciones dadas
    if(componente==CALENTADOR_AIRE)
    {
        // si se esta analizando un calentador de aire el fluido
//frío es aire, por eso se busca el calor especifico del aire
        z=Calor_especifico_aire( p, t, pm.Habsc);
        return z;
    }
    if(t>1400)
    { AfxMessageBox(" la temp es mayor de 1400 F\n"
        " no se puede calcular el calor especifico\n"
        " el resultado sera erroneo");
        return z;
    }
    /// de lo contrario el fluido frío sera el agua
    CPropiedad prop;
    prop.unidades=inglesas;
    double t1=t-1;
    double t2=t+1;
    prop.presión= p;
    prop.temp= t1;
    prop.tipo_cal=_PT;
    prop.calcular();
    double h1=prop.entalpía;
    prop.temp= t2;
    prop.tipo_cal=_PT;
    prop.calcular();
    double h2=prop.entalpía;
    double C = (h2-h1)/((t+0.1)-(t-0.1));
    z=C/10;
    return z;
}

float CInterGasAguaAire::calculo_Ud()
{
    float z=0.0;
    // En esta funcion calculo si es necesario el coeficiente de
//transferencia de calor en condiciones de diseño
    float Tec=prop_diseno[flujocaliente_entrada].temp;
    float Tsc=prop_diseno[flujocaliente_salida].temp;
    float Tef=prop_diseno[flujofrío_entrada].temp;
    float Tsf=prop_diseno[flujofrío_salida].temp;
    /// para el programa de la caldera esta funcion esta cambiada
//porque en la caldera todos los componentes se comportan como tipo
//de contracorriente o contraflujo
    LMTD_diseno=Lmtd_contraflujo(Tec,Tef,Tsc,Tsf);
    if(calcular_Ud==0)
    {
        // si el usuario a introducido el valor no se calcula
//sino que se toma directamente
        z=Ud;
        return z;
    }
}

```

```

        float Cmax=calor_maximo_diseño(); //se calcula el calor maximo
//de transferencia a condiciones de diseño
        z=(Cmax)/(Area_total_transferencia_calor*LMTD_diseño); // se
//calcula el coeficiente
        return z;
}

float CInterGasAguaAire::calor_maximo_diseño() //solo para agua
{
    float z=0.0;
    // en esta funcion se calcula el calor maximo de diseño se
//hallan las presiones medias y las temperaturas medias entre la
//entrada y salida del intercambiador
    float pmcd=
presión_media(prop_diseño[flujocaliente_entrada].presión,prop_diseño
[flujocaliente_salida].presión);
    float pmfd=
presión_media(prop_diseño[flujofrío_entrada].presión,prop_diseño[flu
jofrío_salida].presión);
    float tmcd=
Temp_media(prop_diseño[flujocaliente_entrada].temp,prop_diseño[flujoc
aliente_salida].temp);
    float tmfd=
Temp_media(prop_diseño[flujofrío_entrada].temp,prop_diseño[flujofrío
_salida].temp);
    // se calcula el calor especifico del fluido frío
    float Cpfd=calculo_Cp(pmfd,tmfd);
    float
        Cmfd=prop_diseño[flujofrío_entrada].flujo*Cpfd;
//variable transitoria
    z=Cmfd*(prop_diseño[flujofrío_salida].temp-
prop_diseño[flujofrío_entrada].temp); //se calcula
// el calor transmitido maximo d ediseño
    return z;
}

float CInterGasAguaAire::perdidas_presión_lado_frío()
{
    float z=0.0;
    // En esta funcion se halla las perdidas de presión del lado
//del fluido frío en condiciones de operación, de acuerdo al
//apendice G de las Notas Tecnicas del GateCycle la ecuacion 4a

perdidas_presión_diseño[lado_frío]=prop_diseño[flujofrío_entrada].pr
esión-prop_diseño[flujofrío_salida].presión;
    float rflujo=
prop_real[flujofrío_entrada].flujo/prop_diseño[flujofrío_entrada].fl
ujo;
    float rtemp=
prop_real[flujofrío_entrada].temp/prop_diseño[flujofrío_entrada].tem
p;
    float rpresión=
prop_real[flujofrío_entrada].presión/prop_diseño[flujofrío_entrada].
presión;
    z=perdidas_presión_diseño[lado_frío]*elear(rflujo,a0)*elear(rtemp,
b0)*elear(rpresión,c0);
    return z;
}

```

```

}
float CInterGasAguaAire::elevar(float x, float y)
{
    // funcion para elevar un numero a otro
    float z=0.0;
    z=pow(x,y);
    return z;
}

float CInterGasAguaAire::perdidas_presión_lado_caliente()
{
    float z=0.0;
    // En esta funcion se halla las perdidas de presión del lado
    //del fluido caliente en condiciones de operación, de acuerdo al
    //apendice G de las Notas Tecnicas del GateCycle la ecuacion 4a

    perdidas_presión_diseño[lado_caliente]=prop_diseño[flujocaliente_
e_entrada].presión-prop_diseño[flujocaliente_salida].presión;
float rflujo=
prop_real[flujocaliente_entrada].flujo/prop_diseño[flujocaliente_entr
ada].flujo;
float rtemp=
prop_real[flujocaliente_entrada].temp/prop_diseño[flujocaliente_entr
ada].temp;
float rpresión=
prop_real[flujocaliente_entrada].presión/prop_real[flujocaliente_entr
ada].presión;

z=perdidas_presión_diseño[lado_caliente]*elevar(rflujo,a1)*elevar(rtemp,b1)*elevar(rpresión,c1);
return z;
}

float CInterGasAguaAire::calculo_Uo()
{
    float z=0.0;
    // En esta funcion se halla el coeficiente de transferencia de
    //calor en condiciones de operación, de acuerdo al apendice G de las
    //Notas Tecnicas del GateCycle la ecuacion 13
    float U_diseño=calculo_Ud();
    float rflujo=
prop_real[flujocaliente_entrada].flujo/prop_diseño[flujocaliente_entr
ada].flujo;
float rtemp=
prop_real[flujocaliente_entrada].temp/prop_diseño[flujocaliente_entr
ada].temp;
float rpresión=
prop_real[flujocaliente_entrada].presión/prop_diseño[flujocaliente_entr
ada].presión;

    z=U_diseño*elevar(rflujo,a2)*elevar(rtemp,b2)*elevar(rpresión,c
2);
    return z;
}

float CInterGasAguaAire::Entalpia_PT(float p, float t)
{
    float z=0.0;

```

```

// calculo de entalpía del agua con la presión y temperaturas dadas
float tsat=Buscar_Temp_saturación(p);
if(t==tsat)
{
    CPropiedad prop;
        prop.unidades=inglesas;
        prop.presión=p;
        prop.calidad=1;
        prop.tipo_cal=_PX;
        prop.calcular();
        z=prop.entalpía;
        return z;
}
CPropiedad prop;
prop.unidades=inglesas;
prop.presión=p;
prop.temp=t;
prop.tipo_cal=_PT;
prop.calcular();
z=prop.entalpía;
return z;
}

float CInterGasAguaAire::Temp_PH(float p, float h)
{
    float z=0.0;
    // se calcula la temperatura del agua con presión y entalpía
//dadas
    CPropiedad prop;
    prop.unidades=inglesas;
    prop.presión=p;
    prop.entalpía=h;
    prop.tipo_cal=_PH;
    prop.calcular();
    z=prop.temp;
    return z;
}

int CInterGasAguaAire::calculo_presión_salida()
{
    // calculo de las presiones de salida en condiciones de
//operación en ambos lados (frío y caliente)
    perdidas_presión_real[lado_caliente]=perdidas_presión_lado_caliente();
    perdidas_presión_real[lado_frío]=perdidas_presión_lado_frío();
    float Dpco=perdidas_presión_real[lado_caliente];
    float Dpfo=perdidas_presión_real[lado_frío];
    prop_real[flujocaliente_salida].presión=prop_real[flujocaliente_entrada].presión-perdidas_presión_real[lado_caliente];
    prop_real[flujofrío_salida].presión=prop_real[flujofrío_entrada].presión-perdidas_presión_real[lado_frío];
    if(prop_real[flujocaliente_salida].presión<=0.0)
    {
        AfxMessageBox("El valor de la presión de entrada del flujo caliente\n"
            "en operación es menor que las perdidas generadas en el equipo");
        return 0;
    }
}

```

```

    }
    if(prop_real[flujofrío_salida].presión<=0.0)
    { AfxMessageBox("El valor de la presión de entrada del flujo
de enfriamiento o frío\n"
    "en operación es menor que las perdidas generadas
en el equipo");
    return 0;
    }
    return 1;
}

float CInterGasAguaAire::Delta_presión_lado_frío()
{ // devuelve el valor de las perdidas de presión en el lado frío
//en condiciones de operación
    return perdidas_presión_real[lado_frío];
}

float CInterGasAguaAire::Delta_presión_lado_caliente()
{ // devuelve el valor de las perdidas de presión en el lado
//caliente en condiciones de operación
    return perdidas_presión_real[lado_caliente];
}

float CInterGasAguaAire::Temp_salida_lado_frío()
{ // devuelve el valor de la temperatura de salida en el lado
//frío en condiciones de operación
    return prop_real[flujofrío_salida].temp;
}
float CInterGasAguaAire::Temp_salida_lado_caliente()
{ // devuelve el valor de la temperatura de salida en el lado
//caliente en condiciones de operación
    return prop_real[flujocaliente_salida].temp;
}

float CInterGasAguaAire::perdidas_presión_diseno_lado_caliente()
{ // devuelve el valor de las perdidas de presión en el lado
//caliente en condiciones de diseño
    return perdidas_presión_diseno[lado_caliente];
}

float CInterGasAguaAire::perdidas_presión_diseno_lado_frío()
{ // devuelve el valor de las perdidas de presión en el lado frío
//en condiciones de diseño
    return perdidas_presión_diseno[lado_frío];
}

float CInterGasAguaAire::Buscar_Temp_saturación(float p)
{ float z=0.0;
// calcula la temperatura de saturación del agua a una presión dada
CPropiedad prop;
    prop.unidades=inglesas;
    prop.calidad=0.0;
    prop.presión=p;
    prop.tipo_cal=_PX;
}

```

```
        prop.calcular();
        z=prop.temp;
        return z;
    }

float CInterGasAguaAire::presión_salida_real_lado_caliente()
{
    return prop_real[flujocaliente_salida].presión;
}

float CInterGasAguaAire::presión_salida_real_lado_frío()
{
    return prop_real[flujofrío_salida].presión;
}
```

Anexo H. Archivos fuentes del programa Válvula_Carga_Parcial.exe

1. Válvula.h.

/* Este es el archivo de declaración de la clase CVálvula que se utiliza en el programa Válvula_Carga_Parcila.exe para realizar los cálculos a carga parcial.*/

```
// Válvula.h: interface for the CVálvula class.
//
#if
!defined(AFX_VÁLVULA_H__C969ACB5_691E_4A0D_B079_4BF863CF6005__INCLUD
ED_)
#define
AFX_VÁLVULA_H__C969ACB5_691E_4A0D_B079_4BF863CF6005__INCLUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

#define operacion_1 0
#define operacion_2 1
#define Linear 0
#define Equal 1
#define diseno 0
#define real 1
#define Totalmente_abierta 0
#define Entre_abierta_linear 1
#define Entre_abierta_equal 2

typedef struct{
    float Pérdida_presion[2];
    float Caudal[2];
    int sistema_unidades[2];
    float gravedad_especifica;
    float Rangeabilidad;
    int TipoComp;
    float apertura;
}datos;

class CVálvula
{
public:
    float pérdida_presion_real();    ///función que devuelve la
//pérdida de presión a través de la válvula en condiciones de
//operación que es calculada
    void calcular(); //función que hace los cálculos de la válvula

    CVálvula(); //constructor por defecto
```

```

CVálvula(datos dt) // constructor que inicializa datos
{
    Pérdida_presion[diseño]=dt.Pérdida_presion[diseño];
    Caudal[diseño]=dt.Caudal[diseño];
    gravedad_especifica=dt.gravedad_especifica;
    Rangeabilidad=dt.Rangeabilidad;
    Caudal[real]=dt.Caudal[real];
    TipoComportamiento=dt.TipoComp;
    apertura=dt.apertura;
}
virtual ~CVálvula(); //destructor
private:
    int TipoComportamiento; //define que tipo de comportamiento
//tiene la válvula de acuerdo al tipo de comportamiento de la
//válvula se hacen los siguientes cálculos para conocer la pérdida
//de presión en condiciones de operación
    float calculo_equal();// comportamiento definido por la
//bibliografía
    float calcular_Cvd(); // comportamiento definido por la
//bibliografía
    float calculo_lineal(); // comportamiento definido por la
//bibliografía
    float calculo_cuadratico(); // comportamiento que relaciona
//los parametros de operacion con los de diseño variables usadas
//para el calculo de la pérdida de presión en la,válvula
    bool Flujo_caracteristico;
    float Pérdida_presion[2];
    float Caudal[2];
    float gravedad_especifica; //del liquido
    float Rangeabilidad; // de la válvula
    float Coeficiente_Flujo[2];// el coeficiente de flujo de la
//válvula o Cv
    float Coeficiente_Flujo_diseño;
    float apertura; // de la válvula
};
#endif //
#ifndef AFX_VÁLVULA_H__C969ACB5_691E_4A0D_B079_4BF863CF6005__INCLUD
ED_

```

2.Válvula.cpp.

/*Este es el archivo de definición de la clase CVálvula que se utiliza en el programa Válvula_Carga_Parcila.exe para realizar los cálculos a carga parcial.*/

```

// Valvula.cpp: implementation of the CValvula class.
//
#include "stdafx.h"
#include "Valvula_Carga_Parcial.h"
#include "Valvula.h"
#include <math.h>
#ifdef _DEBUG

```

```

#undef THIS_FILE
static char THIS_FILE[]=__FILE__;
#define new DEBUG_NEW
#endif
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//
// Construction/Destruction
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//
CValvula::CValvula()
{
    Pérdida_presion[diseño]=0.0;
    Pérdida_presion[real]=0.0;
    Caudal[diseño]=0.0;
    Caudal[real]=0.0;
    gravedad_especifica=0.0;
    Rangeabilidad=0.0;
    Coeficiente_Flujo_diseño=0.0;
    apertura=0.0;
    Flujo_caracteristico=0;
}

CValvula::~CValvula()
{
}

float CValvula::calcular_Cvd()
{
    // calcula el coeficiente de flujo de la valvula cuando esta
    //totalmente abierta,
    float z=0.0;
    float i=(gravedad_especifica/Pérdida_presion[diseño]);
    z=Caudal[diseño]*sqrt(i);
    return z;
}

void CValvula::calcular()
{
    /// de acuerdo al tipo de comportamiento de la válvula definido
    //por el usuario se calcula la pérdida de presión en condiciones de
    //diseño
    switch(TipoComportamiento)
    {case Totalmente_abierta:
        Pérdida_presion[real]=calculo_cuadratico();
        break;
    case Entre_abierta_lineal:
        Pérdida_presion[real]=calculo_lineal();
        break;
    case Entre_abierta_equal:
        Pérdida_presion[real]=calculo_equal();
        break;
    default:
        AfxMessageBox("Mire que va hacer");
        break;
    }
}
}

```

```

float CValvula::calculo_cuadratico()
{
    /// relacion de parámetros de operación con los de diseño según
    //la filosofía o modelo planteado para calculos a carga parcial
    float z=0.0;
    float qr=(Caudal[real]/Caudal[diseño]);
    z=Pérdida_presion[diseño]*pow(qr,2);
    return z;
}

float CValvula::calculo_lineal()
{
    ///función que calcula las pérdidas de presión en la válvula
    //en condiciones de operación con comportamiento lineal según la
    //ecuación 2 de § J.W. Hutchison. Handbook of Control Valves, 2nd
    //Edition , McGraw Hill
    float z=0.0;
    Coeficiente_Flujo_diseño=calcular_Cvd();
    double X=apertura/100.0;
    float Cvo=Coeficiente_Flujo_diseño*((0.96*X)+0.03); //se
    //calcula el coeficiente de flujo de la válvula a la apertura actual
    //de ella
    float Qo=Caudal[real];
    float r=Qo/Cvo;
    z=pow(r,2)*gravedad_especifica;
    return z;
}

float CValvula::calculo_equal()
{
    ///función que calcula las pérdidas de presión en la válvula
    //en condiciones de operación con comportamiento lineal según la
    //ecuación 3 de § J.W. Hutchison. Handbook of Control Valves, 2nd
    //Edition , McGraw Hill
    float z=0.0;
    Coeficiente_Flujo_diseño=calcular_Cvd();
    double X=apertura/100.0;
    float R=Rangeabilidad;
    float Cvo=Coeficiente_Flujo_diseño*pow(R,X-1); // se calcula el
    //coeficiente de flujo de la válvula a la apertura actual de ella
    float Qo=Caudal[real];
    float r=Qo/Cvo;
    z=pow(r,2)*gravedad_especifica;
    return z;
}

float CValvula::pérdida_presion_real()
{
    return Pérdida_presion[real];
}

```

Anexo I. Archivo fuentes del programa Bomba_Carga_Parcial.exe

1. Bomba.h

/*Este es el archivo de declaración de la clase CBomba con la cual se realizan los datos en condiciones de operación en el programa Bomba_Carga_Parcial.exe.*/

```
// Bomba.h: interface for the CBomba class.
//
#if
!defined(AFX_BOMBA_H__8A97F3FE_3A9F_480D_A962_3AEBB92B193D__INCLUDED_
_)
#define AFX_BOMBA_H__8A97F3FE_3A9F_480D_A962_3AEBB92B193D__INCLUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000
#include "Bomba_Carga_ParcialDlg.h"

#define Agua_entrada 0
#define Agua_salida 1

typedef struct{
    float temp;
    float presion;
    float flujo;
}termo;

class CBomba
{
public:
    float potencia_entrada();    ///devuelve la potencia en la
//primera condición o de diseño
    float Potencia();    //devuelve la potencia calculada o de
//operación
    float flujo();    //devuelve el caudal o flujo volumetrico en
//condiciones de operación
    float cabeza_operacion();    //función que devuelve la carga de
//la bomba o cabeza en psia en condiciones reales o de operación
    void calcular();///función que llama el usuario para calcular

    CBomba();///constructor
    CBomba(datos_diseno p,float rpm)
    {    ///constructor que entra valores de diseño o de entrada
        rpm_entrada=p.rpm_diseno;
        Eficiencia_entrada=p.Eficiencia_diseno;
        prop_entrada[Agua_entrada].flujo=p.prop_diseno[Agua_entrada].flujo;
    }
};
```

```

prop_entrada[Agua_entrada].presion=p.prop_diseño[Agua_entrada].presion;
prop_entrada[Agua_entrada].temp=p.prop_diseño[Agua_entrada].temp;prop
p_entrada[Agua_salida].flujo=p.prop_diseño[Agua_salida].flujo;

prop_entrada[Agua_salida].presion=p.prop_diseño[Agua_salida].presion
;
prop_entrada[Agua_salida].temp=p.prop_diseño[Agua_salida].temp;
    rpm_real=rpm;
}
    virtual ~CBomba();    ///destructor
private:
    void comprobar();    ///función que comprueba si hay peligro de
cavitación en las condiciones de entrada de la bomba variables de
trabajo de la clase
    float potencial;
    float potencia_operacion;
    float Eficiencia_salida;
    float Cabeza_real_psia;
    float Cabeza_real_feet;
    float flujo_volumetrico_real;
    bool h;
    termo prop_entrada[2];
    float rpm_entrada;
    float rpm_real;
    float Eficiencia_entrada;
    //////////////////////////////////////
};
#endif
#ifndef AFX_BOMBA_H__8A97F3FE_3A9F_480D_A962_3AEBB92B193D__INCLUDED_

```

2. Bomba.cpp.

/* Este es el archivo de definición de la clase CBomba, con la cual se realizan los datos en condiciones de operación en el programa Bomba_Carga_Parcial.exe.*/

```

// Bomba.cpp: implementation of the CBomba class.

#include "stdafx.h"
#include "Bomba Carga Parcial.h"
#include "Bomba.h"
#include <math.h>
#include "Propiedad.h"

#ifdef _DEBUG
#undef THIS_FILE
static char THIS_FILE[]=__FILE__;
#define new DEBUG_NEW
#endif

```

```

////////////////////////////////////
//
// Construction/Destruction
////////////////////////////////////
//

CBomba::CBomba()
{
    h=0;
    prop_entrada[Agua_entrada].flujo=0.0;
    prop_entrada[Agua_entrada].presion=0.0;
    prop_entrada[Agua_entrada].temp=0.0;
    prop_entrada[Agua_salida].flujo=0.0;
    prop_entrada[Agua_salida].presion=0.0;
    prop_entrada[Agua_salida].temp=0.0;
    rpm_entrada=0.0;
    Eficiencia_entrada=0.0;
    rpm_real=0.0;
    flujo_volumetrico_real=0.0;
}

CBomba::~~CBomba()
{
}

void CBomba::calcular()
{
    // esta es la función principal de la clase
    comprobar(); //se comprueba la entrada de datos
    float m=prop_entrada[Agua_entrada].flujo; //flujo
volumetrico no necesario
    // pues es la misma de la salida
    float p1=prop_entrada[Agua_entrada].presion; //presion del
//agua a la entrada de la bomba
    float t1=prop_entrada[Agua_entrada].temp; //temp. del agua a
//la entrada de la bomba
    float Q1=prop_entrada[Agua_salida].flujo; //flujo volumetrico
//de la bomba
    float p2=prop_entrada[Agua_salida].presion; //presion del agua
//a la salida de la bomba
    float t2=prop_entrada[Agua_salida].temp; //temperatura del
//agua a la salida de la bomba
    if(p1>=p2)
    { AfxMessageBox(" la presion de entrada de diseño debe ser
menor!!!");
        return;
    }
    // el flujo de la bomba es Q en unidades gpm
    float N=(rpm_real/rpm_entrada); //se calcula la relación de
//velocidades de la bomba en rpm entre la de operación o la de
//diseño
    float Q2=Q1*N; //por ley de afinidad de la bomba se calcula
//el flujo volumetrico para la velocidad de operación
    flujo_volumetrico_real=Q2;
    // El delta de presión es p2-p1 en psia se convierte en
//feet(pies)
}

```

```

        float H1=2.309*(p2-p1);    /convierto la cabeza de la bomba de
//psia a pies de agua
        float H2=H1*pow(N,2);    //por ley de afinidad de la bomba se
//calcula la cabeza de la bomba para la velocidad de operación
        Cabeza_real_feet=H2;      //// calculo la potencia en HPs
//////////
        potencial=(Q1*H1*100)/(3960*Eficiencia_entrada); //cálculo la
//potencia en HPs con parametros con unidades inglesas
        potencia_operacion=potencial*pow(N,3); //por ley de afinidad
//de la bomba se calcula la potencia de la bomba para la velocidad
//de operación
        Eficiencia_salida=(Q2*H2*100)/(3960*potencia_operacion);
        ////calculo el incremento de presion en psia
        Cabeza_real_psia=H2/2.309; //convierto la cabeza de la bomba de
//pies de agua a la velocidad de operación a psia
    }

float CBomba::cabeza_operacion()
{
    return Cabeza_real_psia;
}

float CBomba::flujo()
{
    return flujo_volumetrico_real;
}

float CBomba::Potencia()
{
    return potencia_operacion;
}

void CBomba::comprobar()
{
    CPropiedad prop;
    prop.unidades=inglesas;
    float p1=prop_entrada[Agua_entrada].presion;
    float t1=prop_entrada[Agua_entrada].temp;
    prop.temp=t1;
    prop.calidad=1.0;
    prop.tipo_cal=_TX;
    prop.calcular();
    if(p1<=prop.presion)
    {
        AfxMessageBox(" ; Peligro de Cavitación ! ");
        return;
    }
}

float CBomba::potencia_entrada()
{
    ///devuelve la potencia en la primera condicion o de diseño
    return potencial;
}

```

Anexo J. Artículo ASME No. 62-WA-209

**A method for predicting the performance of steam turbine-generators...
16.500 Kw and large.**

A METHOD FOR PREDICTING THE
PERFORMANCE OF STEAM TURBINE-GENERATORS . . .
16,500 KW AND LARGER

by

R. C. Spencer K. C. Cotton

*Large Steam Turbine-Generator Department
General Electric Company, Schenectady, New York*

C. N. Cannon

*Medium Steam Turbine, Generator, and Gear Department
General Electric Company, West Lynn, Massachusetts*

REVISED

JULY 1974

Based on ASME Paper No. 62-WA-209, contributed by the Power Division of the American Society of Mechanical Engineers for presentation at the Winter Annual Meeting, New York City, New York, November 25-30, 1962.

Reprinted by The General Electric Company

FOREWORD TO JULY 1974 REVISION

This paper was presented at the 1962 ASME Winter Annual Meeting and has been used since then to calculate the performance of General Electric steam turbine-generators with high initial superheat for operation at 60 hertz. Since 1962 there has been a continual evolution of these turbine-generators to larger sizes, different applications, etc. For this reason the paper has become somewhat out of date in certain areas. To overcome this problem, some minor revisions and extensions have been made.

The text of the original paper is reproduced exactly as presented. Where a section is no longer valid, a vertical line has been added in the adjacent margin. Where extensions to the method have been made, they are contained entirely within this foreword. Where changes have been made, the appropriate figure, table or appendix has been revised and a revision date has been added to each revised page.

A summary of these revisions is included in this foreword.

REVISIONS

- Appendix I - Definition of generator rating changed to capability at maximum hydrogen pressure.
- The definitions of "design flow" and "throttle flow ratio" have been made more explicit.
- In the list of symbols, dropped P_{min} , added T_m and VWO, and modified S_m .
- Figures 6, 9, 10, 11: Added the term "Rated" to Throttle Pressure.
- Figure 12: Added the qualifying terms "Rated" and "(PSIA)".
- Figure 14: Added explanatory notes.
- Figure 15: Range of back pressure extended from 3.5 in. Hg. Abs. to 15 in. Hg. Abs.
- Figure 16: Exhaust loss curve for the 33.5 in. active length on 90.5 in. pitch diameter bucket at 3600 rpm added.
- Figure 17: Exhaust loss curve for 35 in. active length on 110 in. pitch diameter at 1800 rpm deleted. 38 in. active length on 127.5 in. pitch diameter bucket at 1800 rpm added.
- Figure 18: Exhaust loss curves for 1800 rpm axial flow exhausts deleted. Exhaust loss curve for 3600 rpm "high back-pressure" applications substituted.
- Table II: In box C, reference to "3500 psig, single reheat" has been changed to "supercritical." In box "L" reference to "Single Reheat" and "Double Reheat" changed to "Reheat-subcritical and "Reheat-supercritical."
- Figure 19: New curves of mechanical losses substituted. These new curves better represent actual mechanical losses as unit size increases.
- Figures 20, 21: Curves of K_1 & K_2 based on maximum hydrogen pressure substituted for curves based on minimum hydrogen pressure. Definition of generator rating changed as noted above.
- Figure 22: Curve of minimum hydrogen pressure as a function of percent capability, deleted.
- Figure 23: New curve of reduction in generator losses with reduction in hydrogen pressure based on new definition of generator rating.
- Figure 27: Generator Losses, Fixed Losses, Hydrogen Pressure and Heat Rates revised.
- Figures 28, 29, 30: Deleted. These curves were relative heat rates for a number of different machines at 1.0, 1.5 and 2.0 in. Hg. Abs. respectively. The curves were not consistent with the latest General Electric product line and were made redundant by the publication "Heat Rates for Fossil Reheat Cycles using General Electric Steam Turbine-Generators 150,000 KW and Larger" available as GET-2050.
- Appendix II (Sample Calculation) sections F and G: Mechanical and generator losses recalculated consistent with above.
- Table III and Appendix IV: Modified and extended as necessary for the changes listed above. Mathematical expressions for expansion lines added.

EXTENSIONS

"TWO-SHIFT" OR "PEAKING" UNITS

Some units are designed for rapid starting and loading. They are described in the paper "The Design of Turbines for Frequent Starting" by P. G. Ipsen and D. P. Timo, which was presented at the 1969 American Power Conference. Reprints of this paper are available as GER-2539.

These units have "full-throttling" or "single admission" governing stages and the efficiencies calculated by Table I are made a little poorer to allow for the modifications necessary to give improved starting and loading. For units up to a rating of 265 MW, each section is made poorer by the following:

$$\% \Delta \eta = 0.395 \left(\frac{\text{Throttle pressure} - \text{PSIA}}{\text{Rating} - \text{MW}} \right)^{1/3}$$

For units rated 265 MW and greater, each section efficiency will be made poorer by 0.75 percent.

HIGH BACK PRESSURE UNITS

Conventional General Electric condensing units are suitable for a maximum back pressure of 5 in. Hg. Abs. For operation with dry cooling towers, special units suitable for operation up to 15 in. Hg. Abs. are available at 3600 rpm. These "High Back Pressure" units have special last-stage buckets as indicated on Fig. 18. The reheat section expansion efficiency (down to 1.5 in. Hg. Abs.) is multiplied by 0.99.

SIDE EXHAUST

Some units have had the exhaust to the condensers from either side of the low-pressure sections rather than underneath. The efficiency and exhaust loss for these "Side Exhaust" units are calculated exactly the same as the Normal "Down Exhaust" machines.

SERIES CONDENSERS

Very often the cooling water is directed through sections of the condenser in series. This results in there being a different back pressure for each low-pressure section. The expansion line end point to 1.5 in. Hg. Abs. is taken the same for each hood but the correction to end point (Fig. 15) and the exhaust loss (Fig. 16, 17 or 18) are calculated separately for each hood.

COOLING FLOWS ON DOUBLE REHEAT UNITS

To reduce wheel space temperatures on some double reheat units, cooling steam can be introduced into the intermediate or reheat sections. The flows shown in the following table are at VWO rated pressure; the flows at other conditions should be assumed proportional to throttle flow.

SECTION TO BE COOLED	REHEAT TEMP °F	COOLING FLOW LB/HR	UNIT RATING MW
IP	1000 and less	0	Up to 500
IP	1000 and less	20,000	Above 500
IP	Above 1000	0	Up to 400
IP	Above 1000	20,000	Above 400
RH	1000 and less	0	Up to 400
RH	1000 and less	70,000	Above 400
RH	Above 1000	0	Up to 300
RH	Above 1000	70,000	Above 300

The 20,000 lb/hr cooling flow is from the first-stage shell of the HP turbine and is introduced into the shell of the second stage of the intermediate section.

The 70,000 lb/hr flow is from the third-stage shell of the intermediate section and is distributed into the reheat section as follows:

- first-stage shell - 63,000 lb/hr
- second-stage shell - 7,000 lb/hr

Cooling flows are divided equally between all parallel flows of the section being cooled.

When shell pressure of the second and third stages in the intermediate pressure section and the first and second stage of the reheat section are not known, they can be estimated by assuming a stage pressure ratio of 1.25.

These cooling flows are in addition to the normal packing leakage flows which are shown on Table II.

When cooling flow is introduced into a section, there is a loss in efficiency associated with this steam flow entering the steam path. For simplicity, efficiency margin will be applied only to the reheat condensing expansion as follows:

- no cooling - 0 correction
- cooling to reheat section only - 0.06%
- cooling to both reheat and intermediate sections - 0.10%

CORRECTION TO EXPANSION LINE FOR THE EFFECT OF COOLING STEAM ENTERING THE STEAM PATH

In addition to the efficiency correction, the mixing of the cooling steam in the steam path causes a jog in the expansion line. Since this affects the downstream extractions, the expansion line must show the discontinuity.

A. Condensing Sections

Enter Fig. 14 with the shell pressure and temperature before the mix (point 1) to obtain correction E1. Re-enter Fig. 14 with the pressure and temperature after the mix (point 2) to obtain correction E2. Correct the efficiency calculated from point 1 to the exhaust by the ratio $\frac{100 + E2}{100 + E1}$. Apply the new efficiency to the available energy from point 2 to the exhaust to obtain the new end point. A new stateline will be drawn from point 2 to the new end point.

B. Non-Condensing Sections

The method is the same as for condensing sections except that the expansion efficiency from point 2 to the exhaust pressure is assumed to be the same as for the expansion from point 1.

EFFICIENCY AT OFF-RATED STEAM CONDITIONS

A. HP Sections

For the purposes of this method, the value for "design flow" is always measured at rated initial steam conditions. The value for "exhaust pressure at design flow" is measured at rated initial and rated reheat steam conditions. Volume flow is always calculated using design flow and the specific volume corresponding to rated initial steam conditions.

The efficiency calculation procedure can be extended to include operation at non-rated steam conditions by the following:

1. Calculate an equivalent Throttle Flow Ratio (TFR) at rated steam conditions using the current flow at non-rated conditions.

$$\text{Equivalent TFR} = \frac{\text{Current Flow}}{\text{Design Flow}} \sqrt{\frac{(P/\nu) \text{ rated}}{(P/\nu) \text{ non-rated}}}$$

where P and ν are pressure and specific volume at throttle steam conditions.

2. Use this equivalent TFR in all figures requiring TFR.

B. Intermediate Sections

The current values of Flow, Pressure and Enthalpy at the bowl of the IP Section are used in Fig. 13.

C. Reheat Sections

The current values of Flow and Enthalpy at the bowl of the Reheat Section and (Bowl Pressure/0.98) are used to determine volume flow and to enter Fig. 14.

D. Nonreheat Sections

The values for "design flow" and "pressure at the exit of the governing stage at design flow" are always measured at rated initial steam conditions.

Volume flow is determined as for the HP turbine.

Enter Fig. 14 with the current values of initial pressure and temperature.

Note: These procedures represent an extrapolation of the original method. As with all extrapolations, the degree of uncertainty increases with the amount of extrapolation. This extension is primarily intended for calculations at 5 percent overpressure and initial and reheat temperature variations of ± 50 F.

"MOISTURE FUNCTION" CURVE

The expression "0.87(1-0.01Y)(1-0.0065Y)" is used both in the correction of expansion line end point (Fig. 15) and in the calculation of exhaust loss (Figs. 16, 17, 18). To help in hand calculations, a curve of this function plotted against % Moisture (Y) is given as Fig. F1.

"FEEDBACK" CURVE

If a change in cycle conditions results in a change in the pressure distribution in the high pressure section, the valves-wide-open throttle flow can be affected. This situation usually results when the highest pressure feedwater heater is removed from service.

Fig. F2 is a curve from which the effect on valves-wide-open throttle flow can be derived.

Changes to the pressure distribution in a turbine will normally result in a loss in stage efficiency. However, for small changes such as those resulting from the removal of the top-heater, these losses in stage efficiency can be ignored. The efficiency of the high pressure section would therefore be calculated as if it had been designed for this non-standard pressure distribution.

NOTES

The following notes are not changes or additions to the original paper; rather, they represent clarifications or comments pertaining to that paper.

STEAM PROPERTIES

The sample calculation in Appendix II and the corresponding heat balance (Fig. 27) were prepared using Keenan and Keys steam properties. Small differences should be expected if the 1967 ASME properties are used.

REHEAT SECTION "INITIAL CONDITIONS"

Fig. 14 is used to correct the efficiency of reheat sections for initial steam conditions.

For reheat sections, it is assumed that there is a drop of 2 percent from the initial pressure down to the bowl pressure. If this pressure drop differs from 2 percent for any reason, Fig. 14 should be entered at an equivalent initial pressure of bowl pressure divided by 0.98.

If cooling or leakage steam mixes with the main steam at the reheat section bowl, then Fig. 14 should be entered at an equivalent initial temperature which is based on the "mixed" bowl enthalpy.

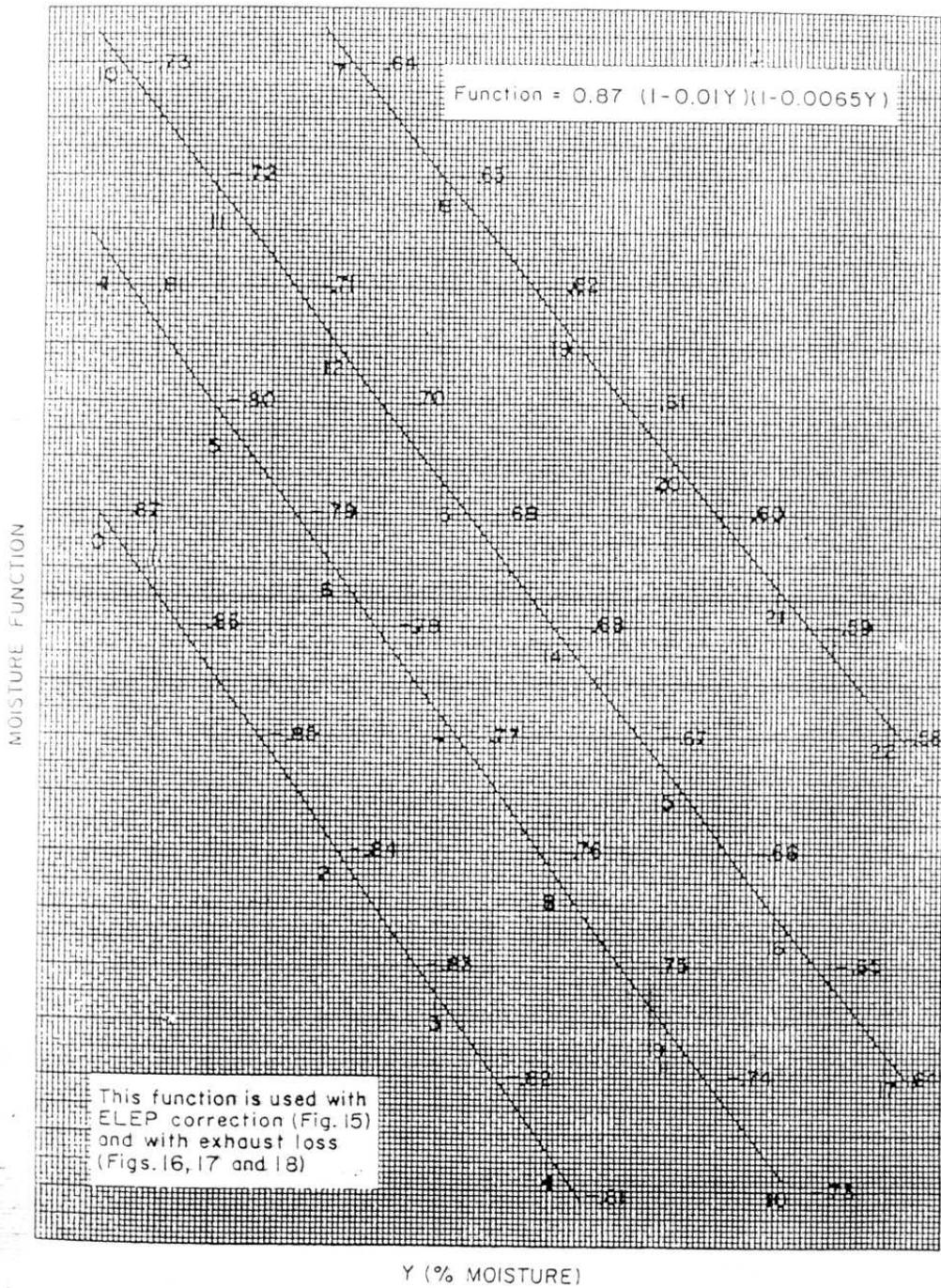


Fig. F1. Moisture function

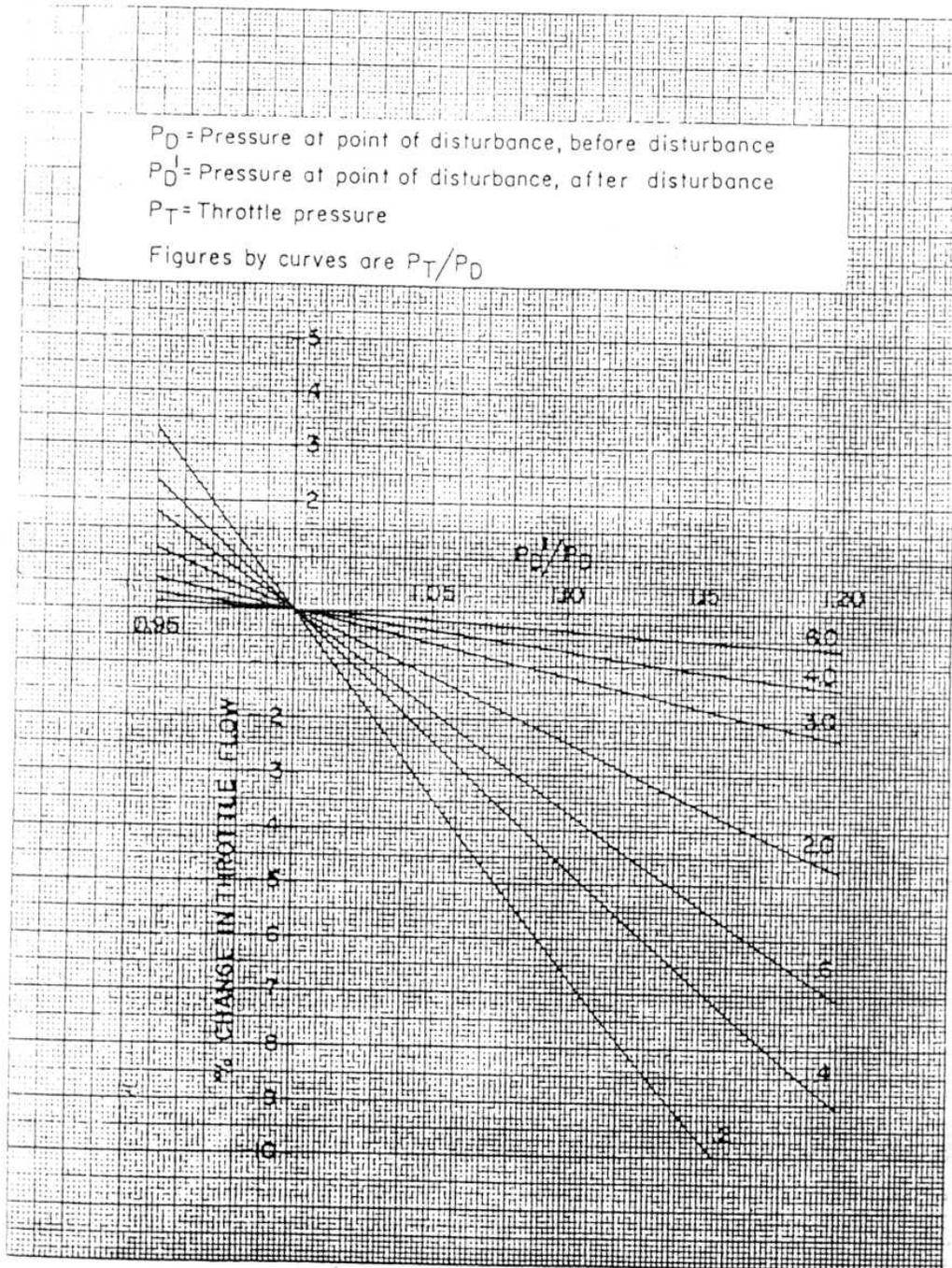


Fig. F2. Effects of pressure feedback on valves-wide-open throttle flow

A METHOD FOR PREDICTING THE PERFORMANCE OF STEAM TURBINE-GENERATORS . . . 16,500 KW AND LARGER

ABSTRACT

A method is presented for predicting the performance of modern steam turbine-generator units designed for high efficiency levels. This method is based on recent developmental and analytical results.

The necessary curves, tables, and instructions are provided for the application of the method to a large variety of units of various kilowatt ratings, types, and steam conditions. Relative heat rate curves are presented and mathematical expressions are provided for computer applications.

I. INTRODUCTION

Since 1940, General Electric Company engineers have published three ASME papers on the performance of steam turbine-generator units. The first was the Warren-Knowlton paper⁽¹⁾ showing the over-all performance of nonreheat turbine-generator units. In 1952, the Elston-Knowlton paper⁽²⁾ extended the area of application and included reheat units. The Hegetschweiler-Bartlett paper⁽³⁾ in 1956 presented full- and part-load performance of nonreheat and reheat turbine sections for calculating over-all performance.

This paper is a modification and extension of the aforementioned Hegetschweiler-Bartlett paper, and is based on the results of tests of groups of stages in the Turbine-Generator Development Laboratory of the authors' company, the experience gained from stage-by-stage calculations, and the results of twenty-four heat rate tests conducted during the past several years in power plants according to the ASME Power Test Code-Committee No. 6 requirements.

Improved testing accuracy and more detailed test measurements of such factors as total exhaust loss, crossover pressure drop, and valve pressure drop during recent years have led to an exceptionally good correlation of turbine component performance. The knowledge and understanding obtained from these results serves as the basis for the calculation methods and performance level presented in this paper.

⁽¹⁾Numbers in parentheses refer to similarly numbered references in "Bibliography" at end of paper.

⁽²⁾Relative heat rate curves deleted from this revision; please see Foreword

This performance method is applicable to reheat and nonreheat, 60-cycle units equipped with hydrogen-cooled generators and operating with conventional steam conditions and normal cycle operation. In addition, it presupposes turbines that are designed to a high performance level by using up-to-date technology in the fields of fluid mechanics and thermodynamics. Figure 1 illustrates a cross-section of a typical unit.

II. EXPLANATION OF THE METHOD

In developing this method it was possible to follow the general procedures of the Hegetschweiler-Bartlett method even though some of the concepts which establish the efficiency levels are different.

The over-all steam turbine performance depends on the efficiency of the individual turbine stages, and can be determined by calculating the performance of each individual stage, or by combining the performance of groups of individual stages into sections. A stage-by-stage calculation system can be used to evaluate existing turbine designs; however, it is not generally practical for predicting the performance of new designs in the study and proposition stage because the detailed knowledge of the particular turbine design is usually unavailable at that time.

Turbine-generator performance at design flow can be predicted accurately from the following:

1. Expansion Line Efficiency, which is determined for each section by the:
 - (a) Volume flow.
 - (b) Pressure ratio.
 - (c) Initial pressure and temperature.
 - (d) Governing stage design, if any.
2. Exhaust Loss
3. Packing and Valve Stem Leakage Flows
4. Mechanical Losses
5. Generator Losses

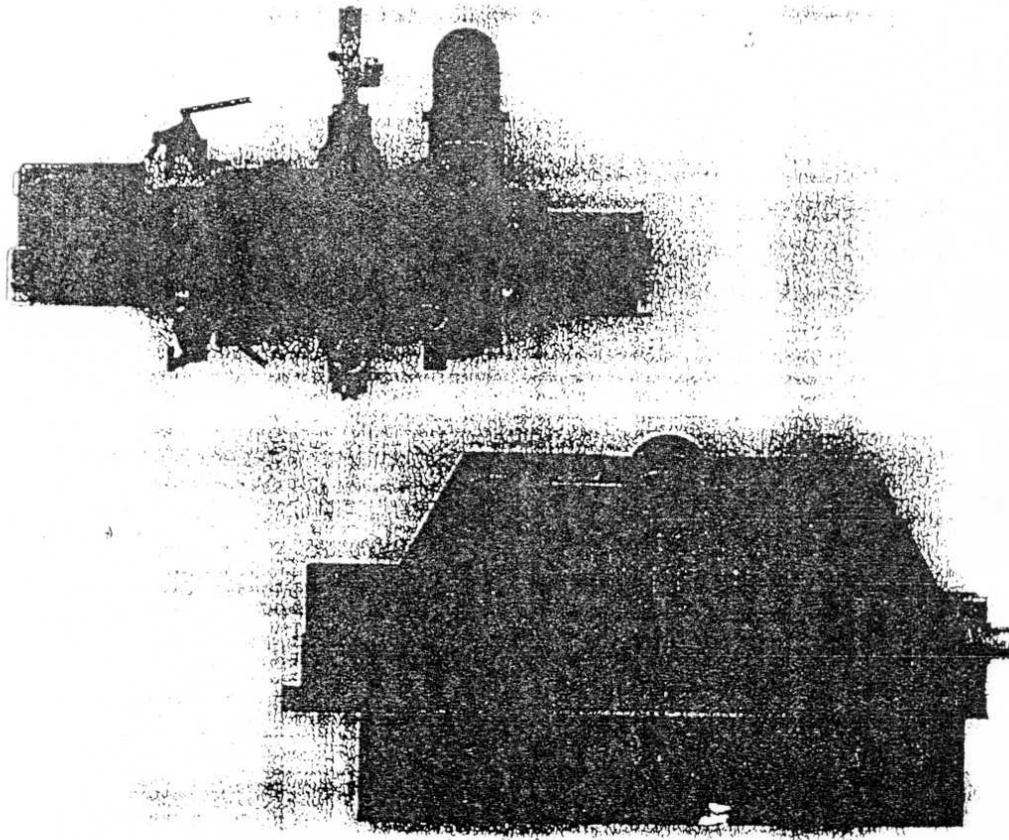


Fig. 1. Typical turbine; 3600/1800-rpm, cross-compound, double-flow reheat

At flows other than design flow, the efficiency of stages between the first stage and a few stages preceding the last is the same as at design flow, since their pressure ratios and volume flows do not change. Hence, the efficiency of the intermediate-pressure turbine section is independent of changes in throttle flow. The efficiency of the high-pressure turbine section is primarily a function of throttle flow ratio, because the efficiency of the first stage decreases with decreasing throttle flow as the pressure ratio across it increases and the velocity ratio decreases below the design value. The efficiency of the condensing section is primarily a function of the annulus velocity of the last stage. Leaving loss for a given last-stage bucket design operating at constant speed is a unique function of annulus velocity. Likewise, hood loss and the change in efficiency of the last stage due to variations in pressure ratio may

be expressed in terms of annulus velocity. Thus, the efficiency characteristics of each section of a reheat turbine can be expressed in terms of one prime variable.

EXPANSION LINE EFFICIENCY

The expansion line efficiency represents the internal efficiency of a group of stages. The section efficiency for one design may vary from another, depending on the initial volume flow and design pressure ratio across the group of stages. As the volume flow increases, the height of the nozzles and buckets increases proportionately; consequently, stage leakages, root and tip losses, and rotation loss are a smaller percentage of the stage energy. As the design pressure ratio increases, the section

IP → efficiency independent of change in throttle flow

efficiency improves due to an increase in average volume flow and an increase in reheat factor. Test efficiencies of turbines with the same pressure ratio and superheat plotted against volume flow have a characteristically hyperbolic shape. This hyperbolic shape is represented by deducing base efficiencies corresponding to infinite volume flow and subtracting from them quotients of constants divided by volume flow.

The effect of the governing stage performance on section efficiency depends on the ratio of governing stage energy to the total energy of the section. These effects can be expressed in terms of the section pressure ratio, and either the diameter or the pressure ratio of the governing stage.

The efficiency of a high-pressure turbine section at design throttle flow is inversely affected by governing stage diameter as shown in Fig. 7. For a properly designed governing stage, an increase in diameter is accompanied by an increase in energy across the stage to keep the velocity ratio at design flow at 0.5 for one-row governing stages and 0.31 for two-row governing stages. Since the stages following the governing stage are generally more efficient than the governing stage, increasing the energy on the governing stage results in a less efficient turbine section. However, as the throttle flow ratio decreases, the efficiency of a high-pressure turbine section with a large diameter governing stage does not decrease as rapidly as a section with a small diameter governing stage, as shown in Fig. 8. The reason is that the percent increase in energy on the governing stage is not as great for a large diameter governing stage.

For nonreheat turbines, the proper governing stage pressure ratio for a condensing section increases as the section initial volume flow is decreased. This is because of the increased parasitic losses at the decreased volume flows. The effect on design flow expansion line efficiency for changing the governing stage pressure ratio is shown in Fig. 2. As the throttle flow ratio is decreased, the energy on the governing stage increases. This results in a reduced velocity ratio of the governing stage and an increase in the fraction of available energy involved. The net result is that changes to governing stage pressure ratio generally invert and magnify changes to part-load expansion line efficiencies. These changes are shown in Fig. 4.

Condensing turbine sections normally operate with their latter stages in the moisture region. Operation in the moisture region is substantially less efficient than in the superheat region. Analysis of the results obtained from recent laboratory tests indicates a supersaturation loss down to the Wilson Line with no moisture loss, and a moisture loss of about 3/4 percent per 1 percent average moisture be-

low the Wilson Line with no supersaturation loss.⁽⁴⁾ Using the Wilson Line established by Yellott⁽⁵⁾ at about 3.5-percent moisture gives a better correlation of test results than when using the original Wilson Line at about 4.5-percent moisture. The correction curves for initial pressure and temperature, Fig. 14, are based on the losses due to operation in the wet region and changes in the reheat factor⁽⁶⁾ to 1.5-in. Hg abs. For exhaust pressures other than 1.5-in. Hg abs, Fig. 15 must be applied. Included in this figure is a correction to account for changes in moisture loss and reheat factor from the 1.5-in. Hg abs condition.

The speed of the low-pressure turbine element of a cross-compound unit affects efficiency because 1800-rpm turbine sections are more efficient than 3600-rpm sections. Table I indicates the procedure to be used when 1800-rpm low-pressure sections are employed in 3600/1800-rpm units.

In the past, small performance differences have been shown between various types and arrangements of turbines. They are not included in this paper because all turbine types and arrangements have been improved to substantially the same performance level.

It should be noted that performance calculated by these methods assumes the turbine-generator is operated in its cycle under normal conditions. Operation under other conditions may affect the performance in varying degrees and may also affect the flow-passing ability of the turbine. For instance, operation with the highest pressure heater out of service somewhat upsets the normal energy distribution of the high-pressure turbine section, and may reduce its performance. It also reduces slightly the flow-passing ability of the governing stage due to pressure feedback from an increased pressure at the extraction point. Such effects must be considered when calculating performance at other than normal conditions.

This method may be used to predict the performance of units designed for 50-cycle operation if differences in design parameters are properly taken into account. In most cases, these differences are small.

EXHAUST LOSS

The exhaust loss curves, Fig. 16, 17 and 18, were obtained by combining calculated leaving loss with exhaust hood loss data obtained from model and field tests, and off-design pressure ratio loss, from results of tests conducted according to ASME PTC-6.

As the exhaust pressure decreases, the axial component of the discharge velocity from the last-stage bucket increases until it reaches sonic ve-

velocity, which is about 1300 feet/second. Any further decrease in exhaust pressure does not increase this velocity above sound velocity. Therefore, there is no further increase in energy available to the turbine stages. The exhaust loss curves cover the normal operating subsonic region. Appendix III describes the procedure for calculating exhaust loss for conditions beyond the point where sonic velocity exists at the discharge of the last-stage bucket.

PACKING AND VALVE STEM LEAKAGES

As in the past, the method assumes that all leakage flows to heaters, or around groups of stages, must be treated as external flows in the heat balance calculations.

Martin's formula⁽⁷⁾ for leakage of steam through labyrinth seals has been used to calculate the leakage flows for representative designs with normal running clearances.

The information necessary to apply the procedure outlined in Appendix V to the various turbine types built by the authors' Company is shown in Table II. A building block approach has been used in order to allow the calculation of the many possible combinations of the turbine components. Factors are also supplied in Table II for the calculation of valve stem leakages, and the steam seal requirements are specified.

MECHANICAL LOSSES

The mechanical losses, Fig. 19, include all the mechanical losses of the turbine such as bearing and oil pump loss, plus the bearing losses of the generator, when operating with conventional lubricants. The mechanical losses associated with a shaft-driven boiler feed pump must be considered separately.

GENERATOR LOSSES

The generator losses, Fig. 20 through 23, include all mechanical and electrical losses of the generator except the bearing losses which are included in the turbine mechanical losses.

The procedure shown in Fig. 20 establishes the generator loss at minimum hydrogen pressure, and may be corrected to any higher hydrogen pressure by the factors provided in Fig. 22 and 23.

III. PROCEDURE

The values of section efficiency obtained, using the procedures of Table I, include losses such as stop and intercept valve pressure drops, crossover pressure drop and the exhaust loss of noncondensing

⁶ Generator losses in this revision based on maximum hydrogen pressure; please see Foreword

sections. The intermediate-pressure section efficiency from Fig. 13 does not include the pressure drop of 2 percent across the intercept valves. The items not included in the Table I procedures for section efficiencies are condensing section exhaust loss and pressure drop from stage shell to extraction flange . . . normally about 1 percent to 3 percent. Packing leakages and other external flows entering or leaving the turbine must be accounted for by heat balance calculations.

The remainder of this portion of the paper on procedures essentially duplicates and amplifies that described in the 1956 Hegetschweiler-Bartlett paper.

For high-pressure sections with governing stages, the expansion line end point can be obtained by multiplying the internal efficiency by the available energy from ahead of the main stop valves to the section exhaust flange, and subtracting this value from the enthalpy of the steam ahead of the main stop valves.

For the larger turbine ratings and the more advanced steam conditions, it becomes increasingly more difficult to design multi-admission governing stage nozzles and buckets to withstand the loadings encountered at partial load because of the large energy and pressure drops associated with multi-valve operation. Therefore, many such units are designed to operate with a smaller number of admissions - in fact, the largest capacity units may be designed for full throttling operation.

For reheat units, when operating at flows less than that for the first admission, the high-pressure turbine efficiency calculation procedure of Table I should be modified as follows:

1. At the first admission point, the high-pressure section efficiency and end point should be determined in the normal manner. Using this end point, determine the efficiency from the first-stage bowl to the high-pressure section exhaust, based on a 4-percent pressure drop between throttle and bowl.

2. At all smaller throttle flows, this same bowl to exhaust efficiency will be applied to the isentropic available energy between the appropriate bowl and exhaust pressures. The bowl pressure can be determined by assuming that $Q/\sqrt{P/v}$ is a constant.

This efficiency remains constant at all flows less than the first-admission flow because the first-stage pressure ratio as well as the following single-stage pressure ratios remain constant. Of course, the throttle-to-exhaust efficiency decreases with decreasing flow.

The method presented herein predicts performance based upon operation at valve points. Because of the current trend to evaluate performance and to operate at or near valve points, performance will

generally be calculated on a locus of valve-best-points basis. However, efficiency correction curves are provided, Fig. 5 and 12, to establish the mean-of-valve-loop performance of sections with governing stages.

For intermediate-pressure turbine sections such as the first-reheat section of a double-reheat unit, the expansion line end point is found by multiplying the internal efficiency by the available energy from the bowl of the section to the section exhaust flange, and subtracting this value from the bowl enthalpy. This efficiency will remain constant over the load range. The pressure drop of 2 percent from the intercept valves to the bowl must be accounted for in the calculation.

For condensing sections, the available energy should be taken from ahead of the stop valves, reheat intercept valves, or equivalent inlet point down to 1.5-in. Hg abs exhaust pressure. Due to the treatment of exhaust loss, the expansion line end point obtained is more or less fictitious, since no such measurable condition exists. To obtain the expansion line end point for exhaust pressures other than 1.5-in. Hg abs, a Btu/lb difference in enthalpy is applied as shown in Fig. 15. This difference depends upon the incremental efficiency of the turbine and the available energy in that part of the moisture region.

Expansion lines are drawn on Mollier charts from the initial conditions ahead of the turbine section to the exhaust pressure and expansion line end point.

Expansion lines for nonreheat units are drawn with Keuffel and Esser curve No. 1864-41. The convex edge of the curve is placed with the rounded end of the curve projecting about 3.5 inches below the 1.5-in. Hg abs expansion line end point. The upper end of the curve is positioned at the throttle enthalpy and at an entropy greater than that at throttle conditions by 0.014 entropy units at design throttle flow. Part-load expansion lines are drawn from the calculated 1.5-in. Hg abs expansion line end points, keeping a constant entropy difference between the expansion lines. Figure 26 shows the construction of expansion lines for nonreheat turbines with a governing stage.

Expansion lines for the high-pressure section of reheat units are drawn as straight lines. The upper end of the expansion line for the high-pressure turbine section may be taken at throttle enthalpy, and at an entropy greater than that at throttle conditions by 0.007 and 0.014 entropy units for one- and two-row governing stages, respectively, at design throttle flow. Partial-flow expansion lines are drawn parallel to the design flow expansion line through the calculated expansion line end point. Fig-

ure 24a shows the construction of these high-pressure section expansion lines.

The expansion lines for the intermediate-pressure section also are drawn as straight lines. The upper end of the expansion line is drawn from the appropriate bowl enthalpy and entropy, and through the calculated expansion line end point as shown in Fig. 24b. Partial-flow expansion lines are drawn in a similar manner.

Expansion lines for condensing sections of reheat units are drawn with Keuffel and Esser curve No. 1864-31. The concave edge of the curve is positioned with the tip of the small end of the curve at the upper end of the expansion line and the lower end passing through the 1.5-in. Hg abs exhaust-pressure expansion line end point. Partial-flow expansion lines are drawn in the same manner. Figure 25 shows the construction of expansion lines for reheat sections.

For condensing sections, both nonreheat and reheat, the used energy end point (UEEP) is obtained by applying the appropriate exhaust loss found as indicated in Fig. 15, 17 or 18, to the expansion line end point at the desired exhaust pressure.

Packing leakage factors and steam seal requirements may be obtained from Table II and applied using the method shown in Appendix V.

Mechanical and generator losses are obtained as outlined in the appropriate figures.

A heat balance, calculated according to these procedures, is shown in Fig. 27. Appendix II gives a detailed numerical example of the use of the method to calculate the efficiencies, end points and losses for this heat balance.

IV. RELATIVE HEAT RATES

The application of the method, described in this paper, to determine heat rate information, even with the aid of a high-speed computer, is a laborious, time consuming process. Therefore, the authors have determined relative heat rates for a large number of reheat units in the range of 50,000 to 1,000,000 kilowatts. This information, along with the assumed steam and cycle conditions, is shown in Fig. 28, 29 and 30 for 1.0, 1.5 and 2.0-in. Hg abs exhaust pressures respectively. The base of the relative heat rates is the unit and cycle shown in Fig. 27. Thus, absolute heat rates for any unit may be obtained by multiplying its relative heat rate by the base unit heat rate.

It should be noted that these curves have been smoothed in the regions where generator cooling and type of turbine arrangement changes. It should also be recognized that the design of the largest

*Relative heat rate curves deleted from this revision; please see Foreword

capacity units represented by these curves has not in all cases proceeded to the point where their performance is definitely established. However, the authors believe the curves represent levels of performance that can be achieved by the careful design of the turbine components involved.

V. MATHEMATICAL EXPRESSIONS

To aid users in the application of this method in calculating heat balances using high-speed computers, Appendix IV contains the equations and tabular values required to represent properly the curves shown in this paper.

VI. CONCLUSIONS

This method accurately predicts full- and part-load performance of a wide range of type and rating

of electric utility steam turbine-generator units operating with various steam conditions and exhaust pressures when designed to a high performance level using modern technology in the fields of fluid mechanics and thermodynamics.

VII. ACKNOWLEDGEMENTS

This paper represents the combined efforts of a large number of steam turbine engineers in the General Electric Company. The authors are especially grateful for the contributions of D. M. Dickinson to the reheat prediction system; of G. F. Kelley for his work on the initial condition factor; of W. O. Printup and P. W. Richardson for the development and analysis of the nonreheat methods; to the programming and evaluation work by R. B. McClintock, R. S. Couchman and S. Chesmejeff; and to E. H. Miller for his constructive comments and encouragement.

BIBLIOGRAPHY

- (1) "Relative 'Engine Efficiencies' Realizable from Large Modern Steam Turbine-Generator Units" by G. B. Warren and P. H. Knowlton, Jr. TRANSACTIONS ASME, 1941, Vol. 63, p 125
- (2) "Comparative Efficiencies of Central Station Reheat and Non-Reheat Steam Turbine-Generator Units" by C. W. Elston and P. H. Knowlton, Jr. TRANSACTIONS ASME, 1952, Vol. 74, p 1389
- (3) "Predicting Performance of Large Steam Turbine-Generator Units for Central Stations" by H. Hegetschweiler and R. L. Bartlett TRANSACTIONS ASME, 1957, Vol. 79, p 1085
- (4) "Supersaturation in Steam Expanding from a Low Initial Quality" by D. J. Ryley THE ENGINEER, May 1957, p 708
- (5) "Supersaturated Steam" by John I. Yellott, Jr. TRANSACTIONS ASME, 1934, Vol. 56, p 411
- (6) "Reheat Factors for Expansion of Superheated and Wet Steam" by C. G. Thatcher TRANSACTIONS ASME, 1938, Vol. 60, pg. A-161
- (7) "Articles on Leakage of Steam through Dummy Pistons" by H. M. Martin ENGINEERING, Jan 10, 1908 and Jan 3, 1919 Also, see Kent's Mechanical Engineering Handbook, Vol. II

APPENDIX I
SYMBOLS, ABBREVIATIONS AND SUBSCRIPTS

A_{an}	Last-stage annulus area, square feet
AE	Available energy or isentropic enthalpy difference, Btu/lb
AE1800	Available energy of 1800-rpm low-pressure section, Btu/lb
AERht. Sect.	Available energy of the reheat section, Btu/lb
BFP	Boiler feed pump
EL	Exhaust loss, Btu/lb
ELEP	Expansion line end point, Btu/lb
Δ ELEP	Change in expansion line end point with exhaust pressure, Btu/lb
H	Enthalpy, Btu/lb
H_{IV}	Enthalpy ahead of intercept valves, Btu/lb
H_m	Mixed enthalpy, the enthalpy resulting when leakage steam is mixed with H_{IV} , Btu/lb
HP	High-pressure section
H_T	Enthalpy at the throttle, Btu/lb
H_{xs}	Isentropic exhaust enthalpy, Btu/lb
IP	Intermediate-pressure section (1st-reheat turbine section)
K	High-pressure packing coefficient for interlocking teeth
K_1	Generator loss factor 1
K_2	Generator loss factor 2
KVA	Generator kilovolt-amperes
%KVA	Percent of generator rating
KW	Kilowatts
Δ KW	Change in generator loss with hydrogen pressure, kw
LP	Low-pressure section
LSB	Low-pressure section last-stage bucket
ML	Mechanical losses, kw
MU	Make-up, %
MVA	Generator megavolt-amperes
N	Number of parallel sections at beginning of an expansion
NV	Number of control valves
P	Pressure, psia
P_B	Pressure at section bowl, psia
PD	Pitch diameter of governing stage, inches
PF	Power factor
PH2	Operating hydrogen pressure, psig
PH2R	Rated hydrogen pressure, psig
P_1	Pressure at exit of governing stage, psia
P_{IV}	Pressure ahead of intercept valves, psia
P_T	Pressure at throttle, psia
P_x	Pressure at exit of section, psia
P_{xD}	Exhaust pressure of section at design flow, psia
Q	Flow at beginning of an expansion, lb/hr
Q_a	Condenser flow, lb/hr
RHTB	Reheat section
S_B	Entropy at bowl, Btu/lb F
S_{IV}	Entropy at intercept valves, Btu/lb F
S_m	Entropy corresponding to H_m and $(P_B/0.98)$ or P_B , Btu/lb F
SPE	Steam packing exhauster
SSR	Steam seal regulator
ST	Entropy at throttle, Btu/lb F
T_{IV}	Temperature at intercept valves, F
T_m	Temperature corresponding to H_m and $(P_B/0.98)$ or P_B , F
T_T	Temperature at throttle, F
UEEP	Used energy end point, Btu/lb
v	Specific volume, cu ft/lb
V	Velocity, ft/sec
V_{an}	Velocity in exhaust annulus, ft/sec
VSLO	Valve stem leak off flow, lb/hr
VWO	Valves wide open
Y	Moisture, %
η	Efficiency, %
$\Delta\eta$	Change in efficiency, %

Revised July 1974

9

APPENDIX I (Cont'd)

DEFINITIONS

Bivariate Polynomial	$Z = \sum_{j=0}^J \sum_{i=0}^I A_{ji} x^i y^j$
Bowl	The space immediately ahead of the nozzles of a turbine stage or section.
Conductor-cooled Generator	Generator utilizing gas or liquid circulation inside the stator bars and/or rotor conductors for cooling purposes.
Conventionally Cooled Generator	Generator utilizing a hydrogen circulation system, for cooling the exterior only, of the stator bars and rotor conductors.
Crossover	A connecting pipe which conducts the steam from one turbine section to another.
Design Flow	The flow which the section is expected to pass with the control valves wide open and rated initial steam conditions.
Expansion Line End Point	The enthalpy of the steam upon completion of its expansion through a turbine section.
Generator Rating	Generator capability at maximum hydrogen pressure.
Governing Stage	A stage which controls the flow of steam into the turbine. Usually each segment of the first-stage nozzle arc is directly connected to individual control valves.
High-pressure Section	The turbine casing into which steam from the boiler is first admitted. The section or sections through which the steam passes prior to reheating.
Intermediate-pressure Section	The turbine section(s) between reheaters on multiple reheat turbines.
Nameplate Rating	The generator output of the turbine-generator when operating at rated initial steam conditions, 3-1/2-in. Hg abs. exhaust pressure, and with 3 percent evaporated feedwater makeup, and with a margin to design flow of 5 percent to assure that shop tolerances on drawing areas, variations in flow coefficients from expected values, etc., will not prevent the turbine from meeting its capability commitments.
Pitch Diameter	The diameter of a turbine stage measured to the midpoint of the buckets.

$$Z = \frac{\sum_{i=0}^I A_i x^i}{1 + \sum_{j=1}^J B_j x^j}$$

APPENDIX I (Cont'd)

DEFINITIONS

Reheat Section(s)	A condensing turbine section(s) through which the steam expands after reheating.
Throttle Flow Ratio	Throttle flow divided by VWO throttle flow at the same initial steam conditions.
Univariate Polynomial	$Z = \sum_{i=0}^I A_i x^i$
Used Energy End Point	The enthalpy of the steam at the exhaust flange of a condensing section including the exhaust loss.
Velocity Ratio	The ratio of the blade speed at the pitch line to the steam velocity corresponding to the available energy of the stage.
Volume Flow	Weight flow (lb/hr) times specific volume (cu ft/lb) at beginning of a section.
1-row Governing Stage	A governing stage consisting of one set of nozzles and a single row of buckets.
2-row Governing Stage	A governing stage consisting of one set of nozzles, two rows of moving buckets, and a row of stationary buckets for reversing the flow.

Revised July 1974

TABLE I
EFFICIENCY CALCULATION PROCEDURES
 (All corrections are in percent)

TURBINE TYPE	NON-REHEAT		HIGH-PRESSURE SECTIONS		INTERMEDIATE-PRESSURE SECTIONS	REHEAT SECTIONS	
	$\frac{E_{loss}}{\phi}$ 3600-rpm Condensing 2-row Governing Stage 0	3600-rpm Noncondensing 1-row Governing Stage 1	3600-rpm Noncondensing 2-row Governing Stage 2	3600-rpm Noncondensing Without Governing Stage 3	3600-rpm Condensing Without Governing Stage 4	3600/1800-rpm Condensing Without Governing Stage 5	1800-rpm Condensing Without Governing Stage 6
Base Efficiency	89.48	87.00	84.00	Fig. 13	91.93	92.95	
Efficiency Correction for Volume Flow	$\frac{430,260}{Q_v} N$ Poorer Fig. 2	$\frac{1,005,200}{Q_v} N$ Poorer Fig. 7	$\frac{1,350,000}{Q_v} N$ Poorer Fig. 10		$\frac{1,270,000}{Q_v} N$ Poorer Fig. 14	$\frac{1,270,000}{Q_v} N$ Poorer Fig. 14	
Efficiency Correction for Governing Stage		Fig. 6					
Efficiency Correction for Pressure Ratio							
Efficiency Correction for Initial Conditions							
Efficiency Correction for Governing Stage at Part Load							
Efficiency Correction for Part Load							
Efficiency Correction for Substitution of 1800-rpm, Low-pressure Section							$1.25 \left(\frac{AE_{1800}}{AE_{\text{tht. sect.}}} \right)$ Better
Efficiency Correction for Mean-of-Loops (Optional)							

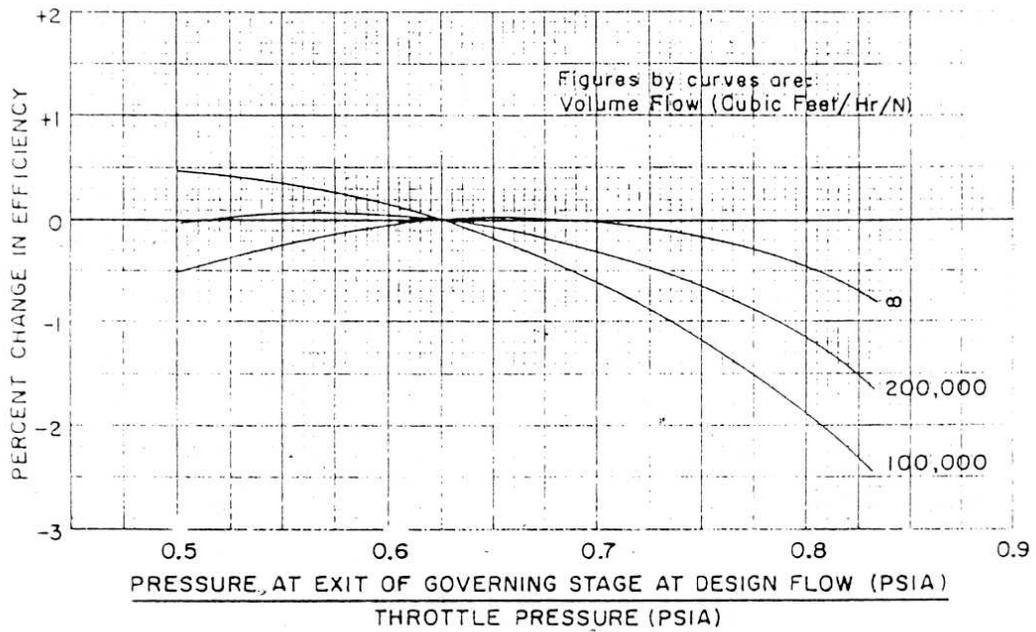


Fig. 2. Nonreheat condensing, 2-row governing stage, design flow and efficiency correction for governing stage pressure ratio

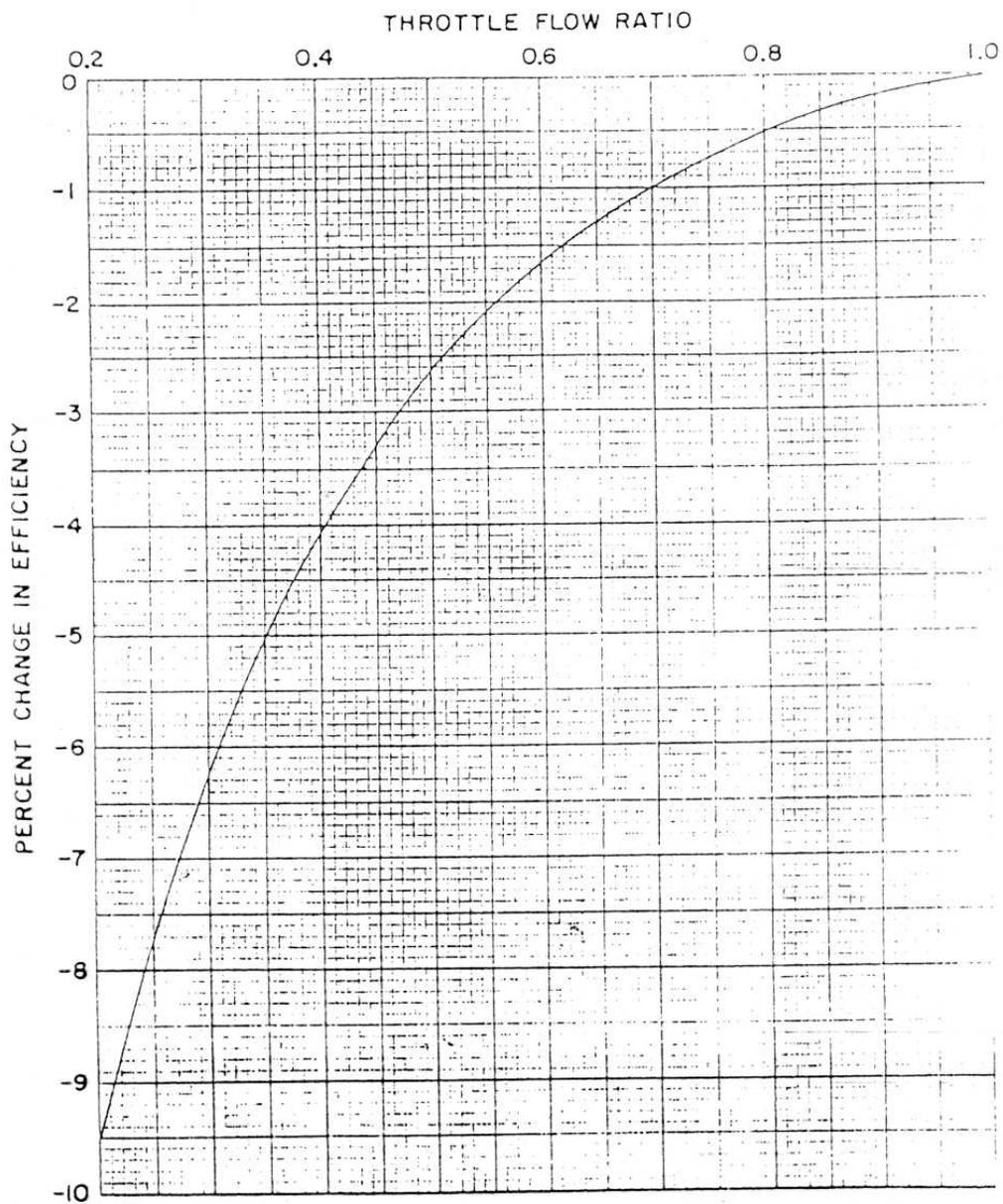


Fig. 3. Nonreheat condensing, 2-row governing stage, part-load efficiency correction for throttle flow ratio

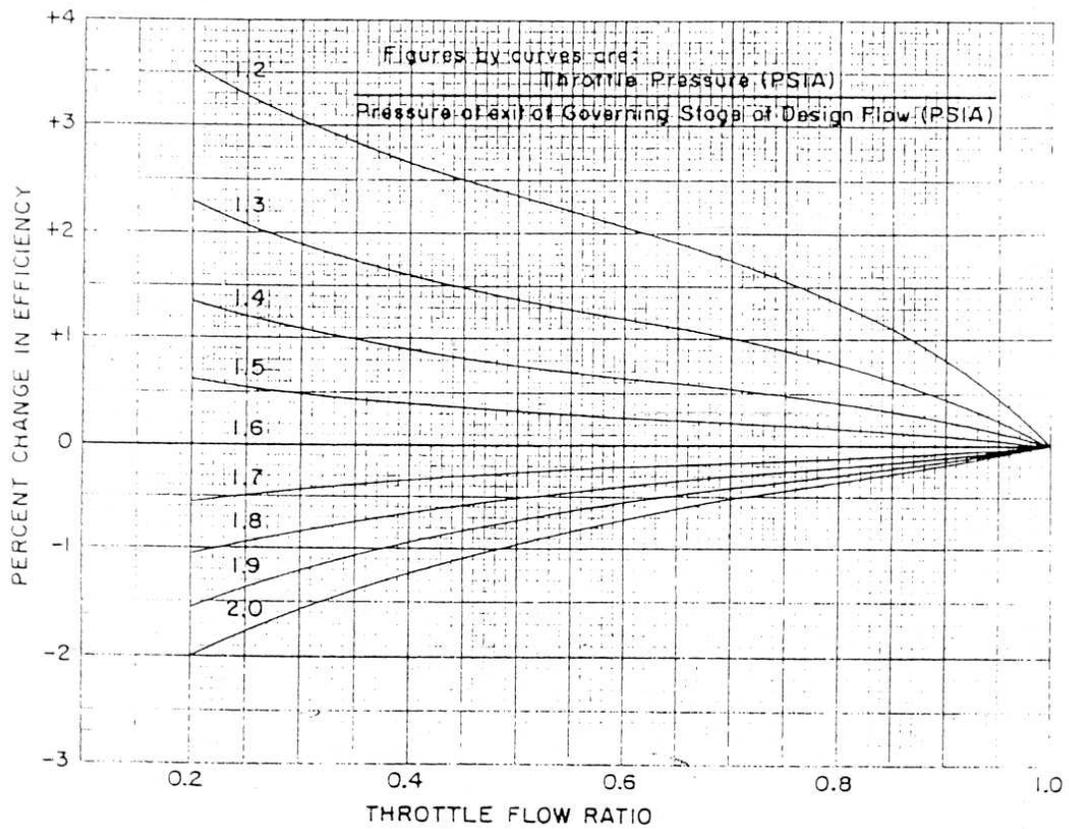


Fig. 4. Nonreheat condensing, 2-row governing stage, part-load efficiency correction for governing stage pressure ratio

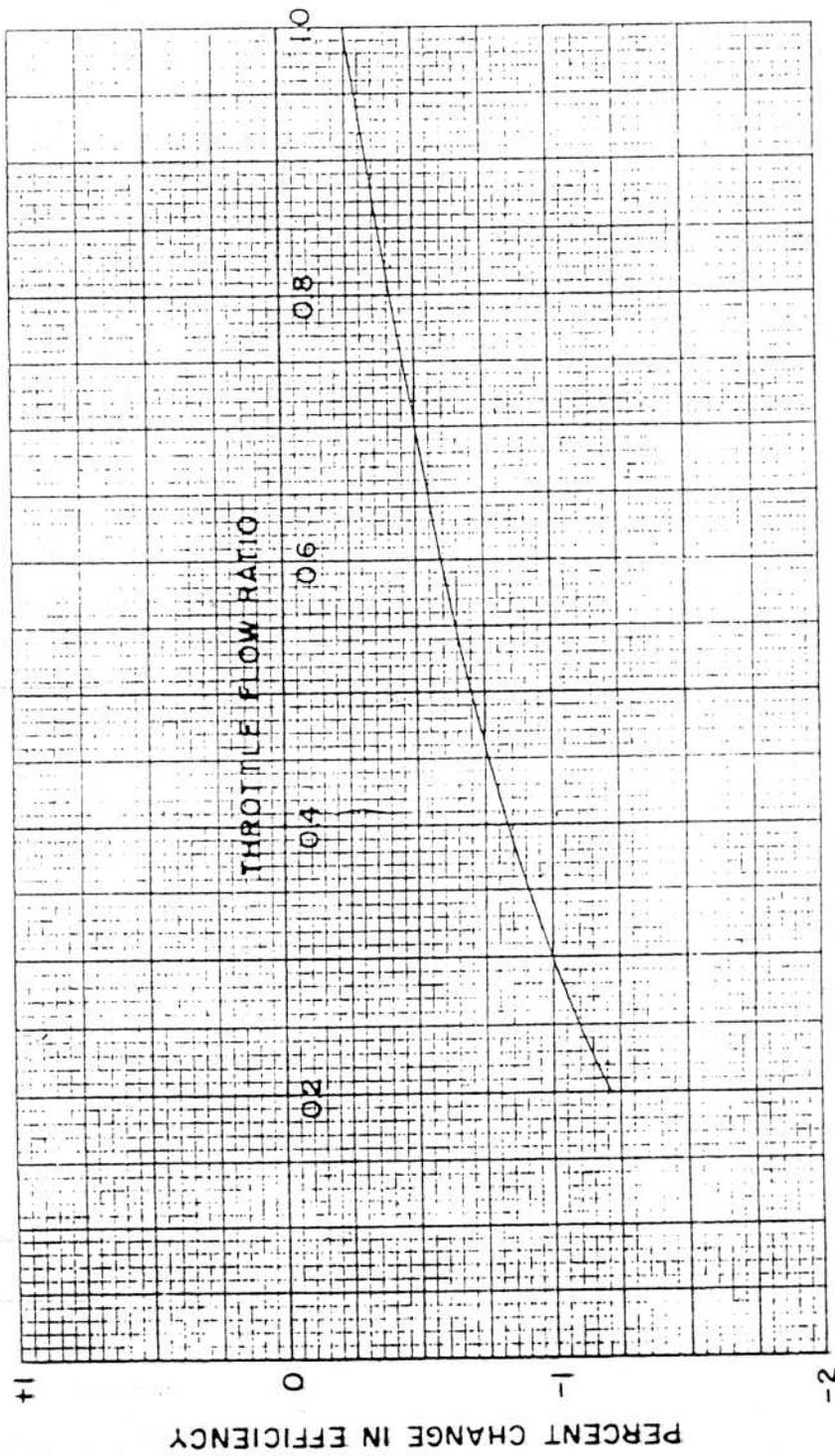


Fig. 5. Nonreheat condensing, 2-row governing stage, efficiency correction for mean-of-valve loops

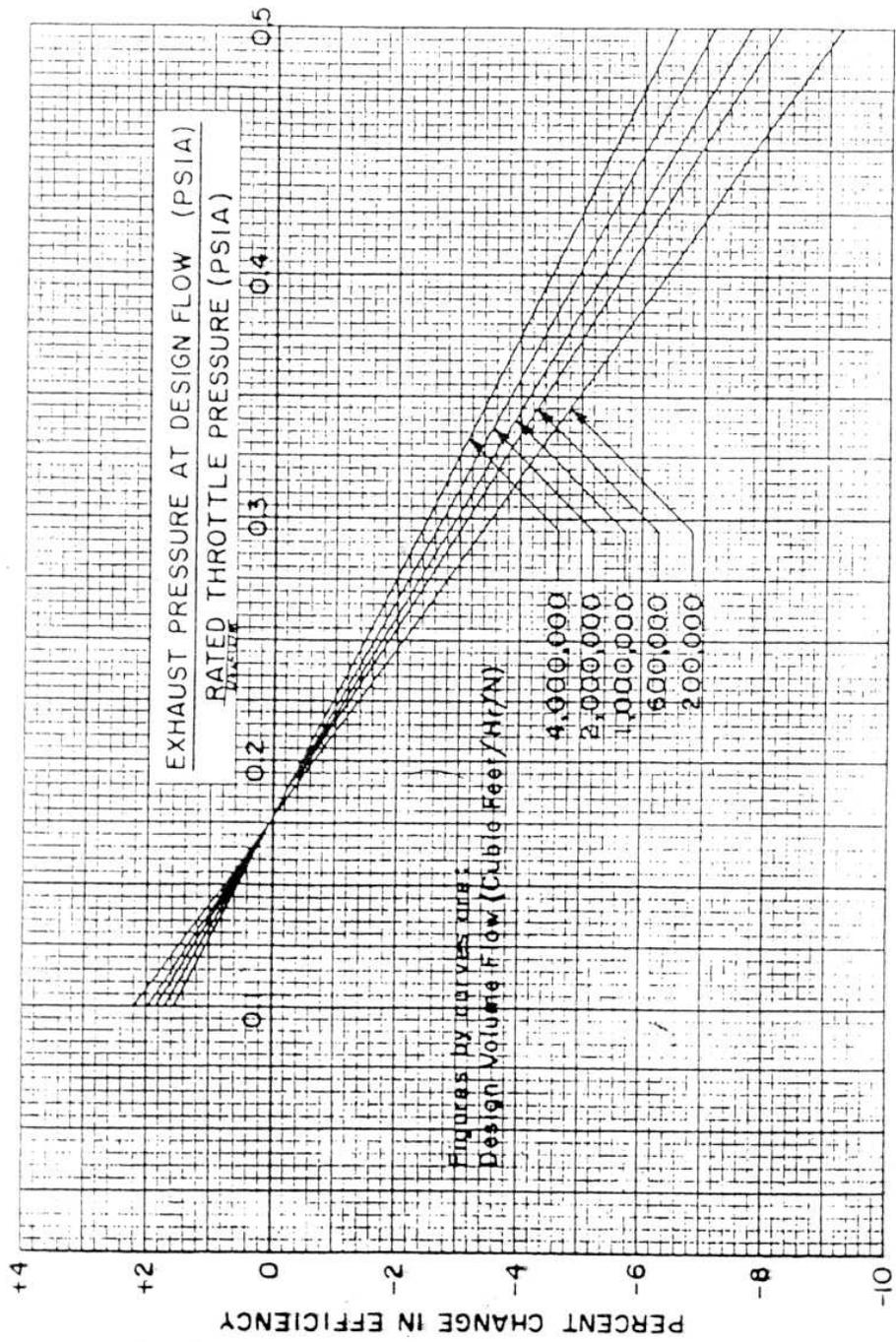


Fig. 6. 3600-rpm high-pressure turbine section, 1-row governing stage, design flow efficiency correction for pressure ratio

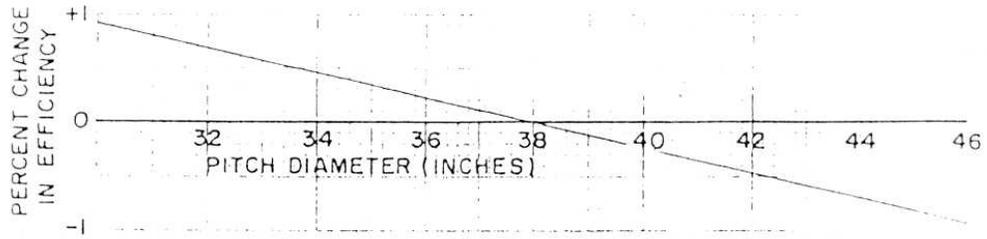


Fig. 7. 3600-rpm high-pressure turbine section, 1-row governing stage, design flow efficiency correction for governing stage pitch diameter

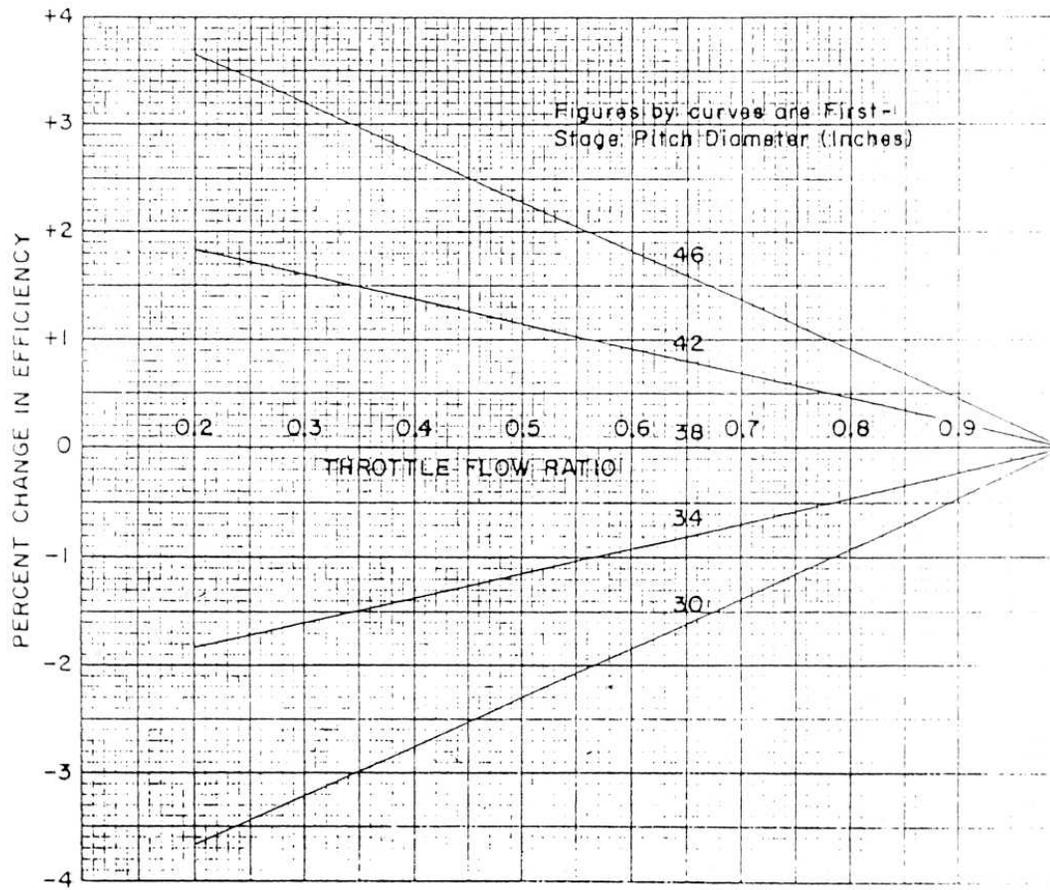


Fig. 8. 3600-rpm high-pressure turbine section, 1-row governing stage, part-load efficiency correction for governing stage pitch diameter

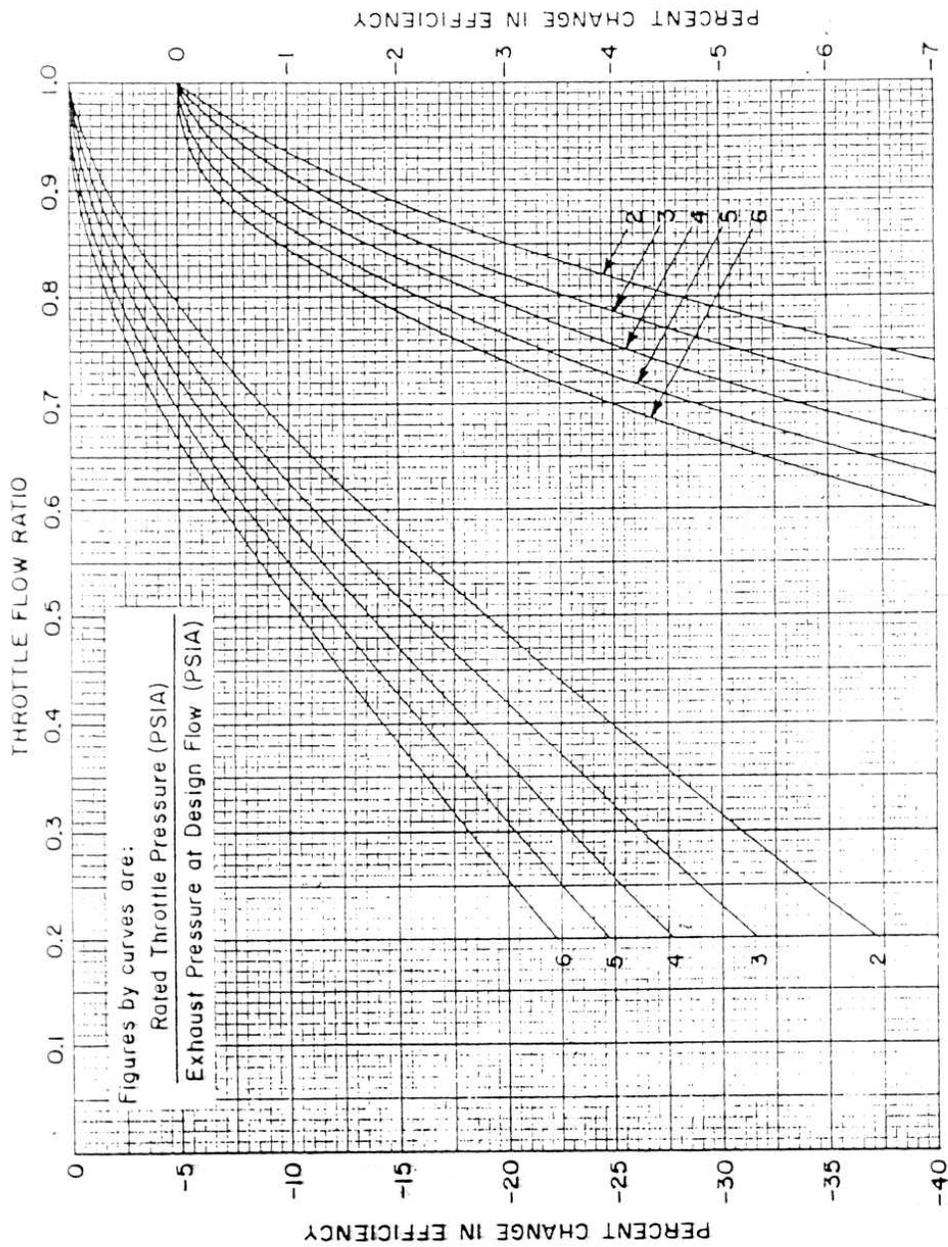


Fig. 9 3000-rpm high-pressure turbine section, 1 row governing stage, part-load efficiency correction for throttle flow ratio

Revised July, 1974

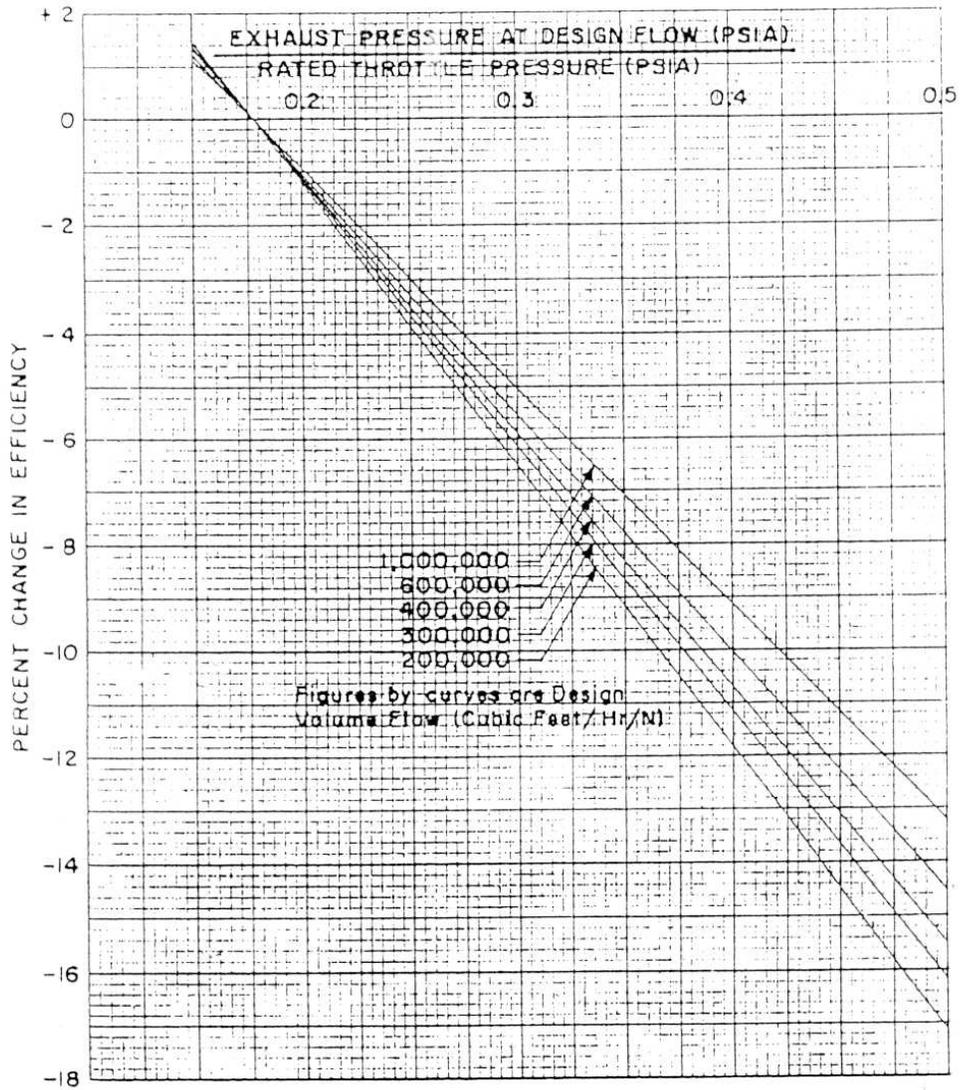


Fig. 10. 3600-rpm high-pressure turbine section, 2-row governing stage, design flow efficiency correction for pressure ratio

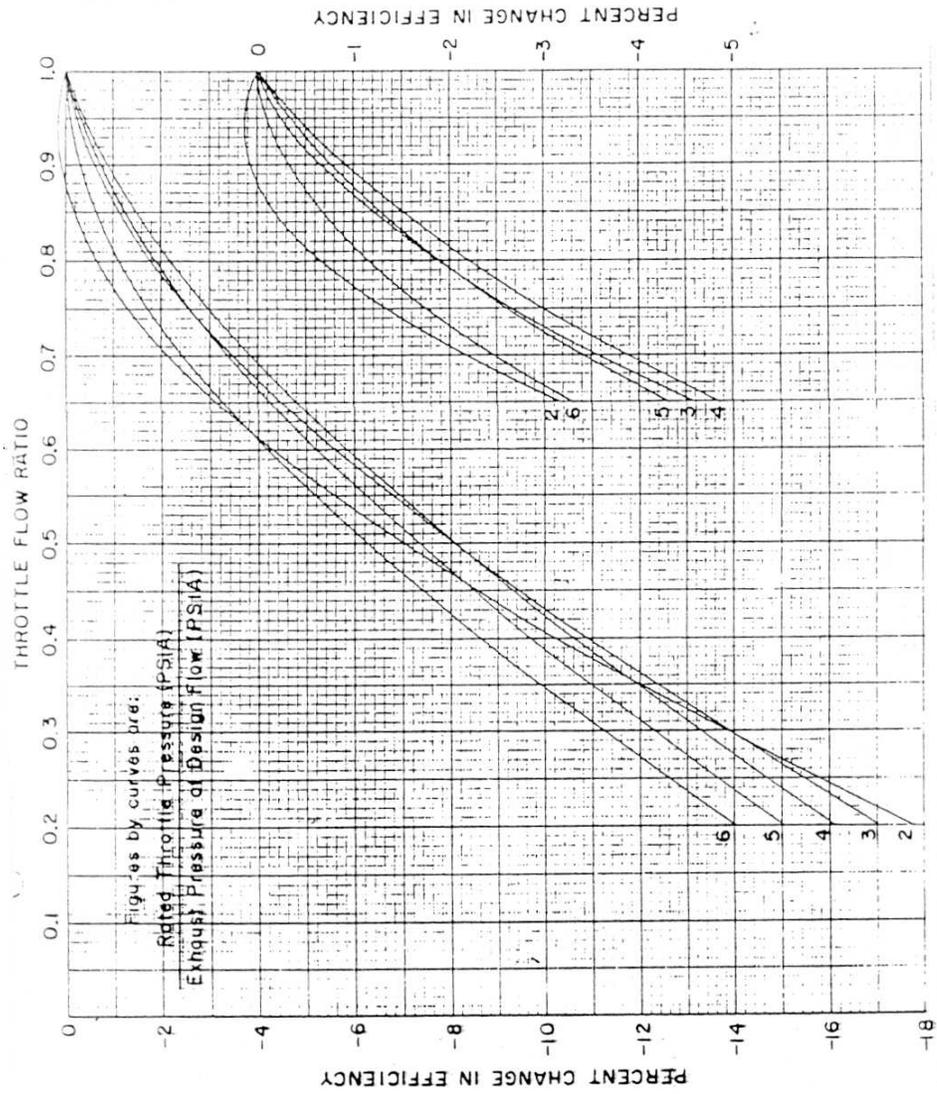


Fig 11 3600-rpm high-pressure turbine section, 2-row governing stage, part load efficiency correction for throttle flow ratio

Revised July 1974

12

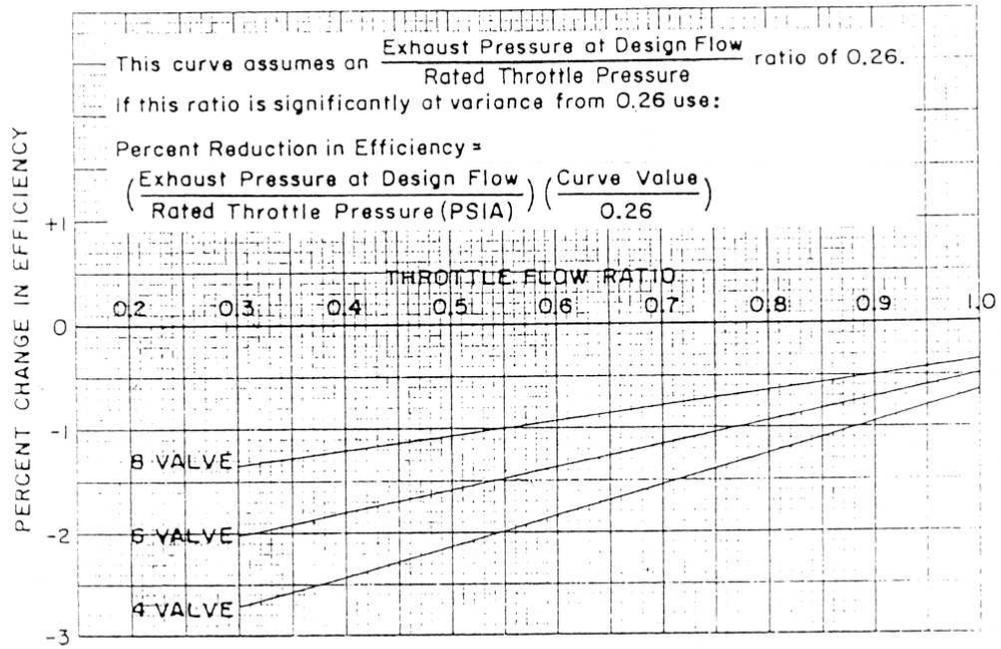
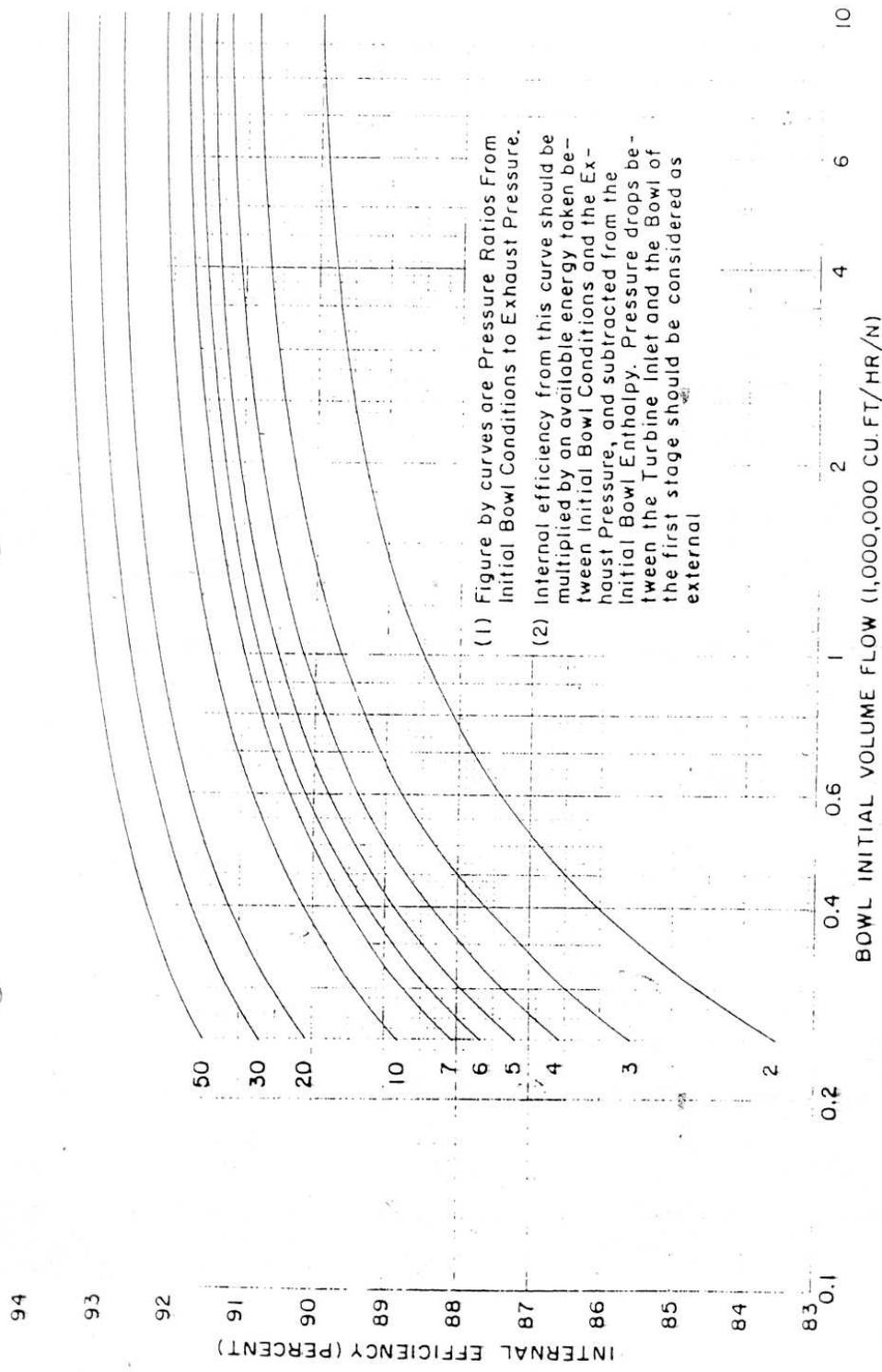


Fig. 12. 3600-rpm high-pressure turbine section, 1- or 2-row governing stage, efficiency correction for mean-of-valve loops



(1) Figure by curves are Pressure Ratios From Initial Bowl Conditions to Exhaust Pressure.
 (2) Internal efficiency from this curve should be multiplied by an available energy taken between Initial Bowl Conditions and the Exhaust Pressure, and subtracted from the Initial Bowl Enthalpy. Pressure drops between the Turbine Inlet and the Bowl of the first stage should be considered as external

Fig. 13. 3600-rpm intermediate-pressure turbine section, without governing stage, section efficiency

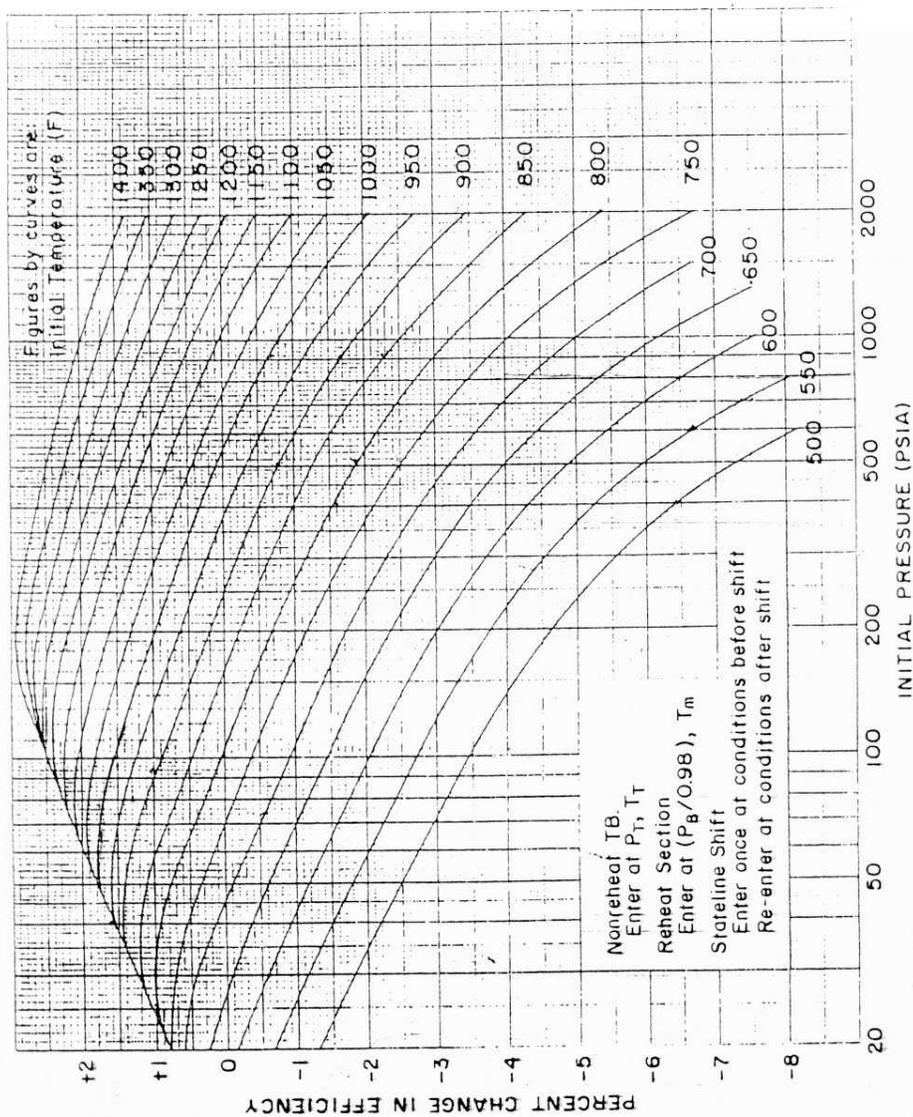
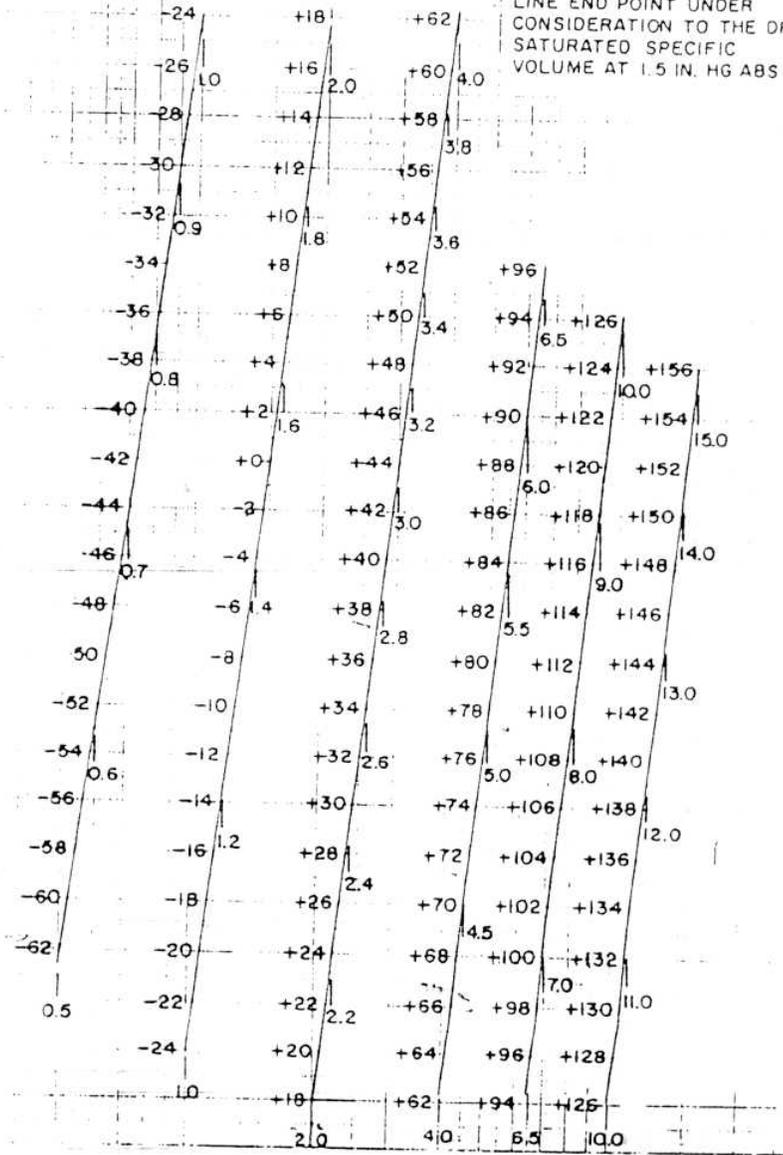


Fig 14 3600- and 1800-rpm, reheat and nonreheat condensing section, efficiency correction for initial steam conditions

(1) WHEN EXPANSION LINE END POINT AT 1.5 IN. HG ABS IS IN THE MOISTURE REGION, USE THE FOLLOWING EQUATION:
 $\Delta ELEP = \Delta ELEP_{Y=0} (0.87)(1-0.01Y)(1-0.0065Y)$
 Y = PERCENT MOISTURE AT ELEP AT 1.5 IN. HG ABS

(2) WHEN THE EXPANSION LINE END POINT AT 1.5 IN. HG ABS IS IN THE SUPERHEAT REGION, MULTIPLY THE CHANGE IN EXPANSION LINE END POINT AT 0 PERCENT MOISTURE BOTH BY THE FACTOR 0.87 AND THE RATIO OF SPECIFIC VOLUME AT THE EXPANSION LINE END POINT UNDER CONSIDERATION TO THE DRY SATURATED SPECIFIC VOLUME AT 1.5 IN. HG ABS

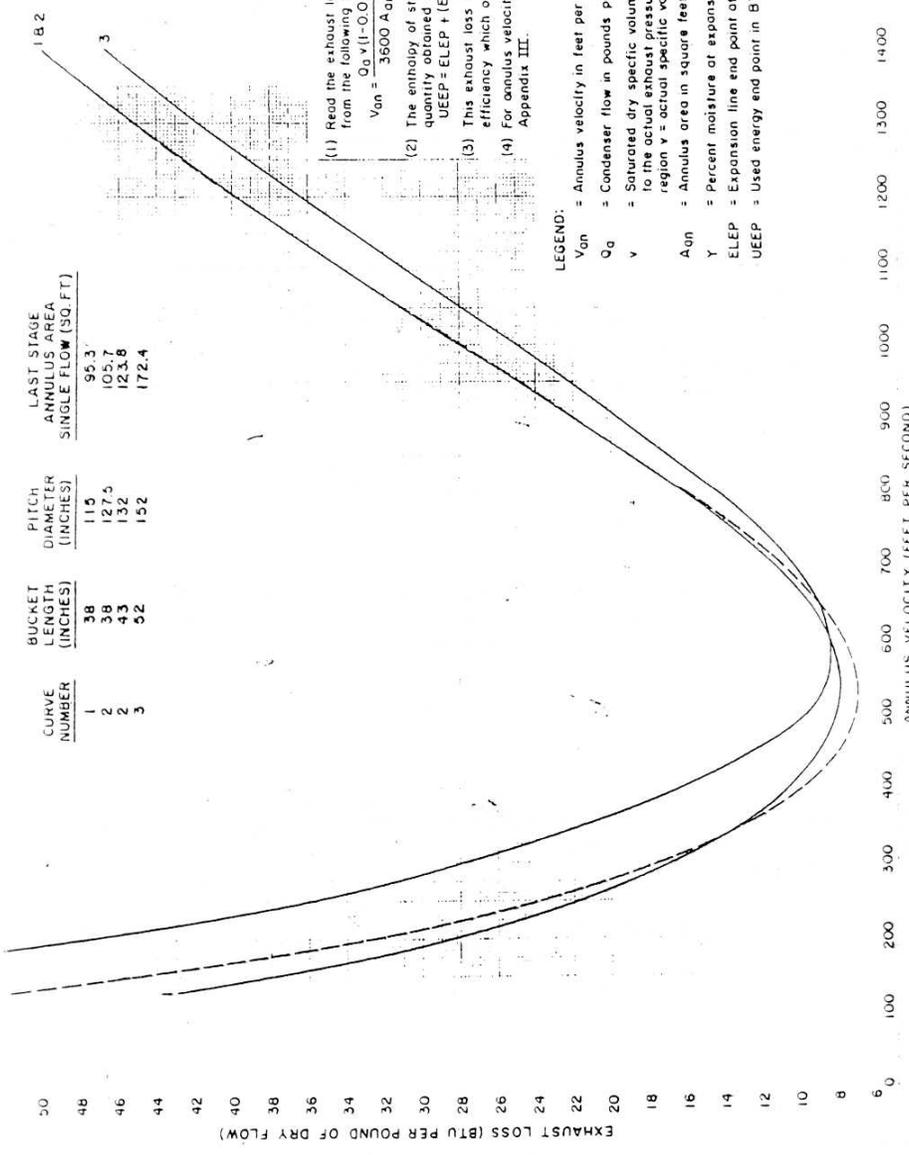
CHANGE IN EXPANSION LINE END POINT WITH 0 PERCENT MOISTURE (BTU/LB)



EXHAUST PRESSURE (IN HG ABS)

Fig. 15. 3600- and 1800-rpm, reheat and nonreheat condensing section, correction to expansion line end point for exhaust pressure

CURVE NUMBER	BUCKET LENGTH (INCHES)	PITCH DIAMETER (INCHES)	LAST STAGE ANNULUS AREA SINGLE FLOW (SQ. FT)
1	38	115	95.3
2	38	127.5	105.7
3	43	132	123.8
5	52	152	172.4



- (1) Read the exhaust loss at the annulus velocity v from the following expression

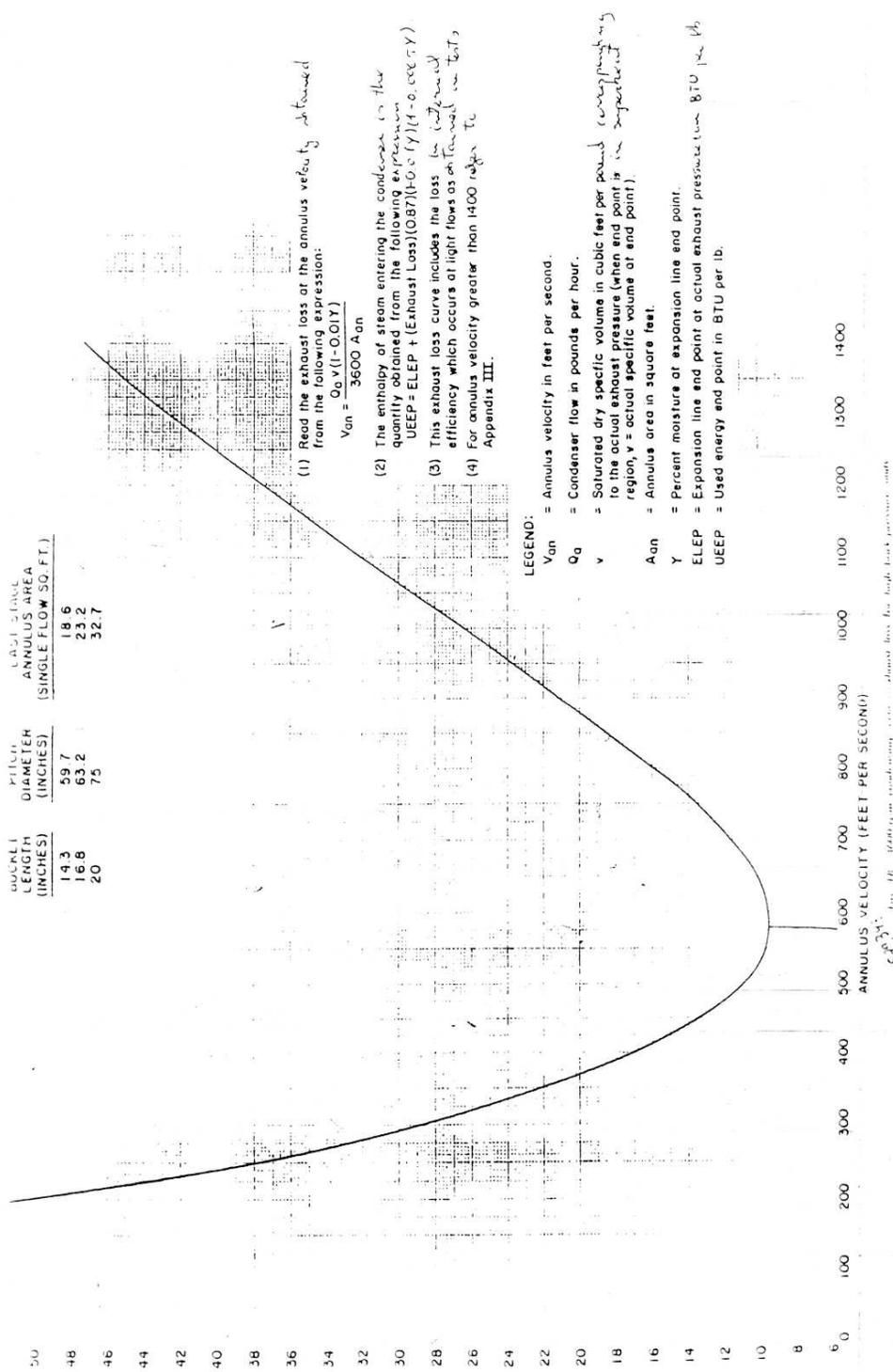
$$Q_a v (1 - 0.01 Y)$$

$$V_{an} = \frac{Q_a v (1 - 0.01 Y)}{3600 A_{an}}$$
- (2) The enthalpy of steam entering the condenser is the quantity obtained from the psychrometric chart. UEEP = ELEP + (Exhaust Loss) \times Efficiency
- (3) This exhaust loss curve includes the loss in exhaust efficiency which occurs at high flow rates.
- (4) For annulus velocity greater than 1400 ft/s see Appendix III.

- LEGEND:
- V_{an} = Annulus velocity in feet per second.
 - Q_a = Condenser flow in pounds per hour.
 - v = Saturated dry specific volume in cubic feet per pound expanded to the actual exhaust pressure (when end point is in superheated region v = actual specific volume at end point)
 - A_{an} = Annulus area in square feet.
 - Y = Percent moisture at expansion line end point.
 - ELEP = Expansion line end point of actual exhaust pressure in BTU/lb
 - UEEP = Used energy end point in BTU per lb

FIG. 17. Exhaust loss curves for various bucket sizes.

BUCKET LENGTH (INCHES)	PISTON DIAMETER (INCHES)	LAST STAGE ANNUSS AREA (SINGLE FLOW SQ. FT.)
14.3	59.7	18.6
16.8	63.2	23.2
20	75	32.7



(1) Read the exhaust loss at the annulus velocity obtained from the following expression:

$$V_{an} = \frac{Q_a v}{3600 A_{an} Y}$$

(2) The enthalpy of steam entering the condenser is the quantity obtained from the following expression

$$UEEP = ELEP + (Exhaust Loss) \{ (0.87)(1 - 0.001 Y) \} + 0.0001 Y$$

(3) This exhaust loss curve includes the loss for interstage efficiency which occurs at light flows as obtained in tests.

(4) For annulus velocity greater than 1400 refer to Appendix III.

LEGEND:

- V_{an} = Annulus velocity in feet per second.
- Q_a = Condenser flow in pounds per hour.
- v = Saturated dry specific volume in cubic feet per pound corresponding to the actual exhaust pressure (when end point is in superheated region, v = actual specific volume at end point).
- A_{an} = Annulus area in square feet.
- Y = Percent moisture at expansion line end point.
- ELEP = Expansion line end point of actual exhaust pressure in BTU per lb.
- UEEP = Used energy end point in BTU per lb.

S.F.P. 31. Exhaust loss curve for high back pressure units.

TABLE II
PACKING LEAKAGE CONSTANTS "C" AND STEAM SEAL FLOWS

Diagram	Diagram	Diagram	Diagram	Diagram	Diagram	Diagram	Diagram	Diagram	Diagram	Diagram	Diagram	Diagram
A	B	C	D	E	F	G	H	I	J	K	L	M
<p>Non-reheat (A) 1450 psig and under --- 225 250 (B) Over 1450 psig 480 480 670</p>	<p>Single Reheat (A) HP-RHTB up to 200 MW 340 --- 450 670 (B) HP-RHTB 200 MW B over 500 --- 520 970 (C) HP-RHTB 1050 RH temp. 430 --- 550 1060 350 650 Double Reheat (D) HP-IP 430 720 870 1200 550 740 (E) HP-RHTB 450 720 870 1200 --- 670 (F) IP-RHTB 390 --- 770 1100 --- 670</p>	<p>Single Reheat (A) HP up to 250 MW 320 --- 370 480 (B) HP 250 MW to 400 MW 450 --- 450 700 --- 520 980 (C) HP 400 MW B over 540 --- 500 800 --- 560 980 Double Reheat (D) HP 430 450 750 450 480 950 720 (E) IP 340 --- 460 530 --- 560 700 (F) IP-RHTB 340 --- 460 530 --- 560 700</p> <p>For supercritical turbines lags 3 and 4 are led from first stage shell, dashed line omitted.</p>	<p>Single Reheat 1 2 3 (A) HP --- 620 970 Double Reheat (B) HP 720 870 1200</p> <p>Equal flows are shown from each end for heat balance purposes only.</p>	<p>Single Reheat (A) RHTB 450 650 Double Reheat (B) RHTB 720 770</p>	<p>Single Reheat (A) RHTB 540 770 540 770 (B) IP --- 550 --- 550 (C) RHTB --- 550 --- 550</p> <p>Equal flows are shown from each end for heat balance purposes only.</p>	<p>Single Reheat (A) RHTB 700 800 600 Double Reheat (B) IP --- 700 --- 800 (C) RHTB --- 700 --- 800</p> <p>See C (E)</p>	<p>Single or Double Reheat (A) RHTB 1400 1600 1520</p>	<p>(A) To L-1 Stage 580</p>	<p>(A) 1000</p>	<p>(A) Crossover: inlet from BIC 670 600 500 (B) Crossover: inlet from F (A) ---</p>	<p>Non-reheat Total VSLO flow = 40(P₁/V₁) 2nd VSLO flow = 50(P₂/V₂) Reheat-subcritical Total VSLO flow = 56(P₁/V₁) 2nd VSLO flow = 50(P₂/V₂) Reheat-supercritical Total VSLO flow = 70(P₁/V₁) 2nd VSLO flow = 80(P₂/V₂) 3rd VSLO flow = 60(P₃/V₃)</p> <p>P₁ = Inlet pressure P₂ or P₃ = Pressure ahead of 2nd or 3rd VSLO V = Specific volume at P₁, P₂, P₃ & turbine enthalpy</p>	<p>STEAM SEAL FLOWS (LB/HR) 1000 MW and above</p> <p>ID TO TOTAL SPE COMD 250 400 650</p> <p>Under 1000 MW Per exhaust end @ 3600 RPM 250 400 650</p> <p>TO SPE COMD TOTAL 700 600 1300 TO CONO TOTAL 1000 600 2500</p>

Notes (1) The first packing leakage in a series will be the sum of the flows to the packing leakoffs following it.
 (2) Leakage 1 in Diagrams B, C, E, G, H, I, J, K is from the shell of the first stage of the section.

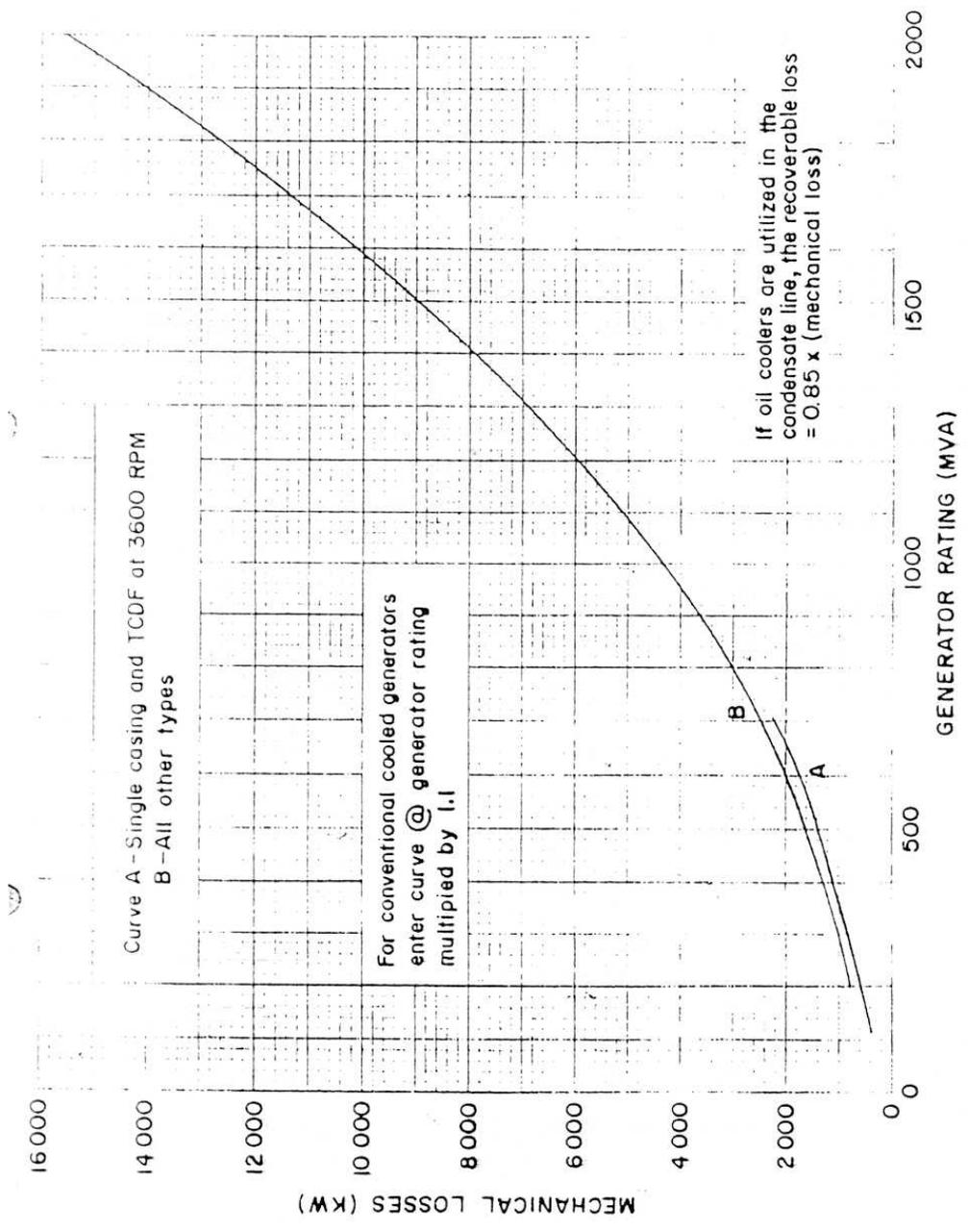
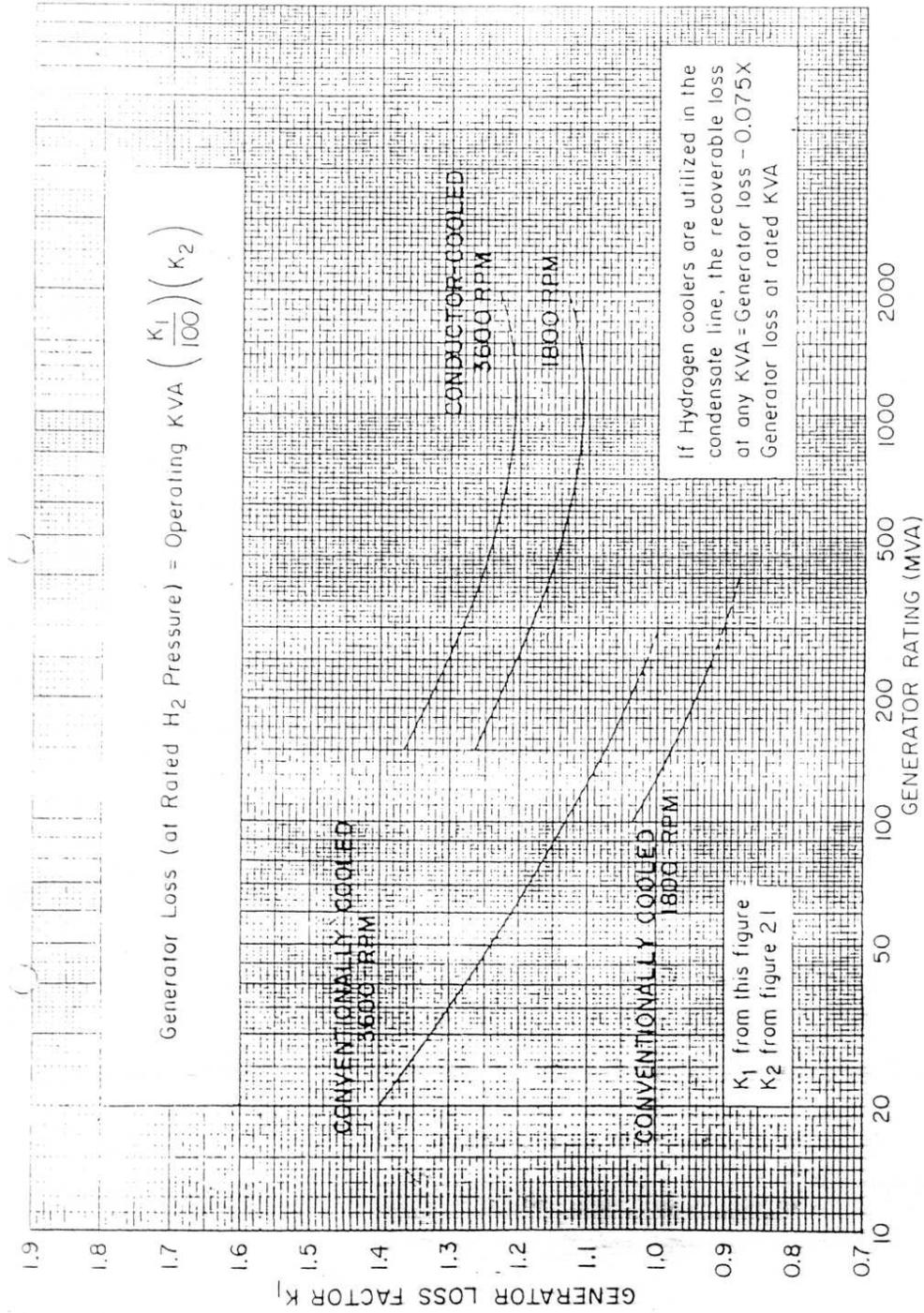


Fig. 19 Mechanical losses



Revised July 1974

Fig. 20 Generator loss factor K_1

37

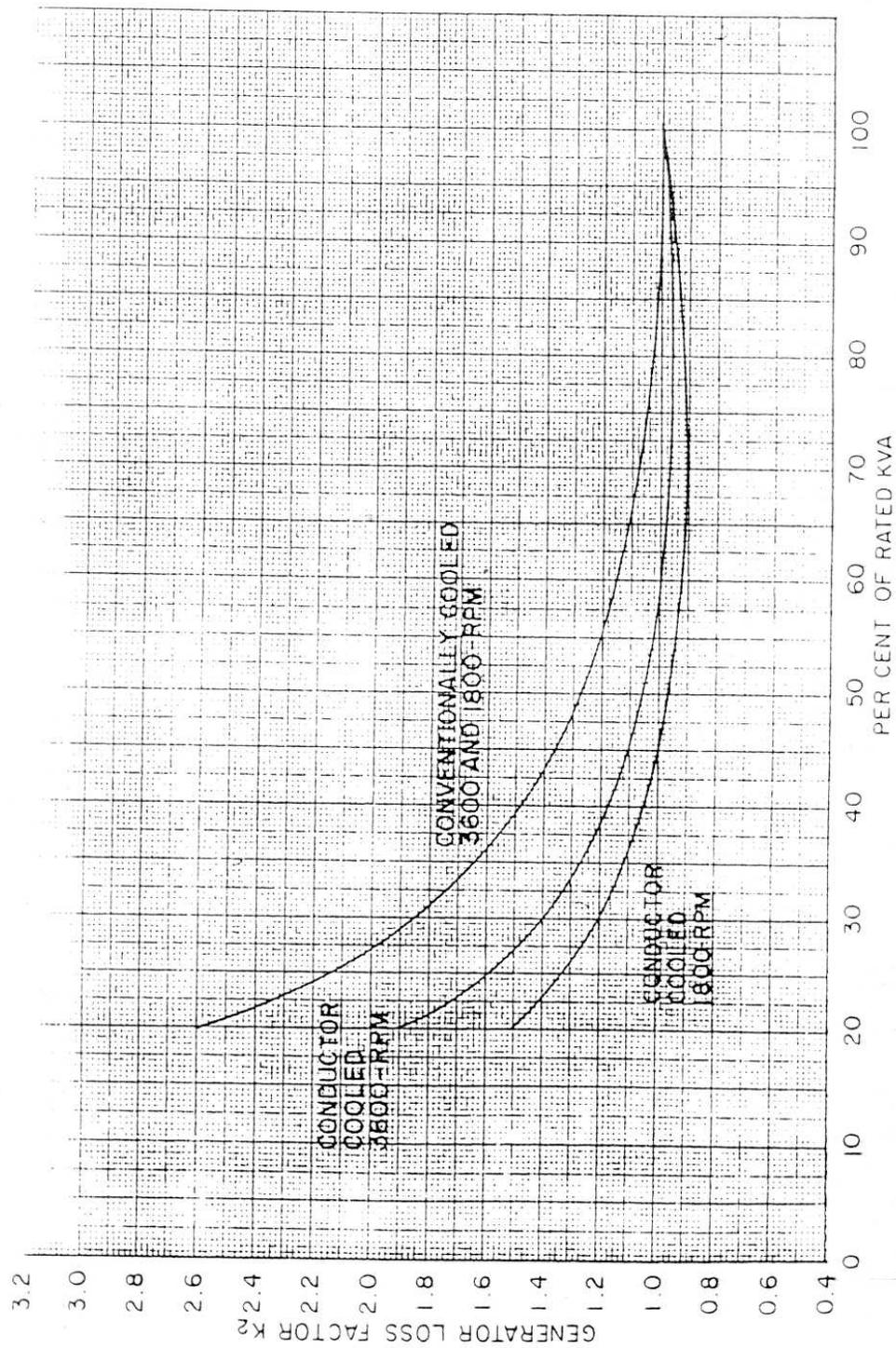


Fig 21. Generator loss factor K_2

Revised July 1971

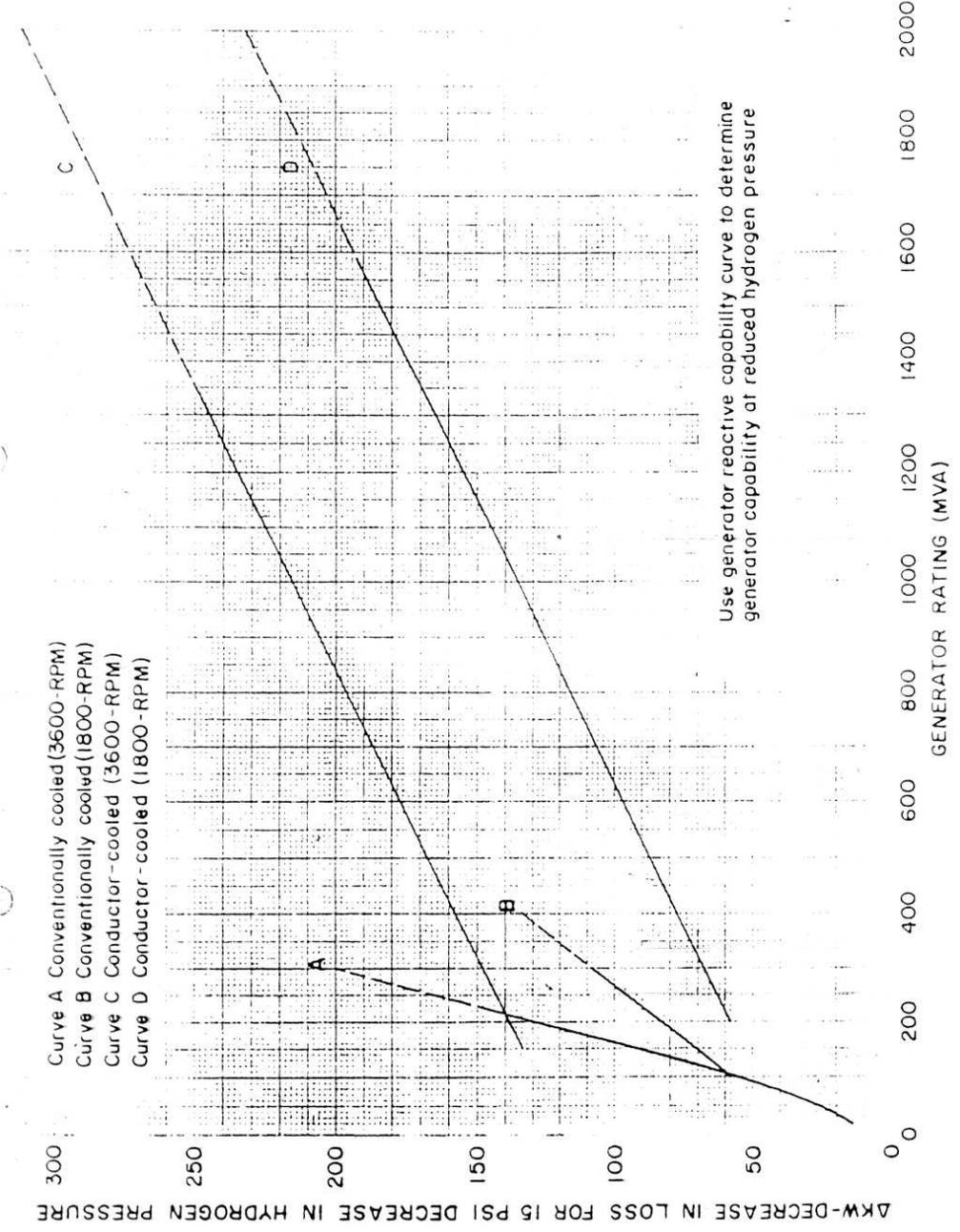


Fig. 23. Change in generator loss with hydrogen pressure, at constant kva

Revised July 1974

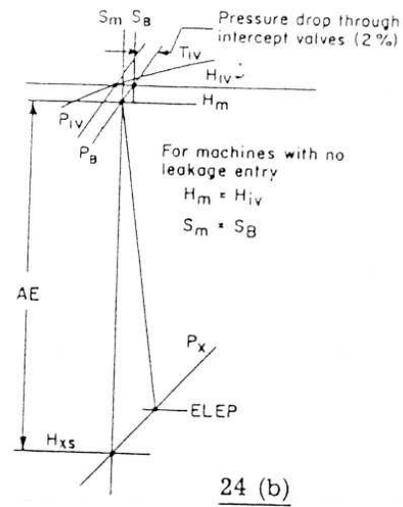
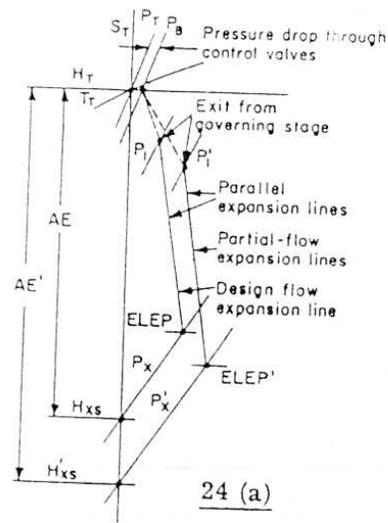


Fig. 24. Construction of expansion lines for high-pressure and intermediate-pressure sections

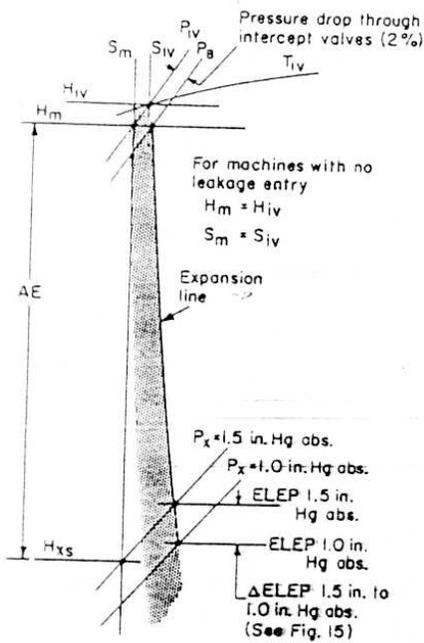


Fig. 25. Construction of expansion lines for reheat sections

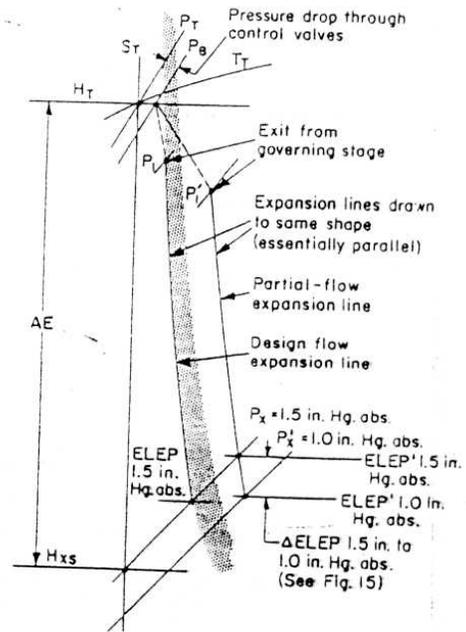
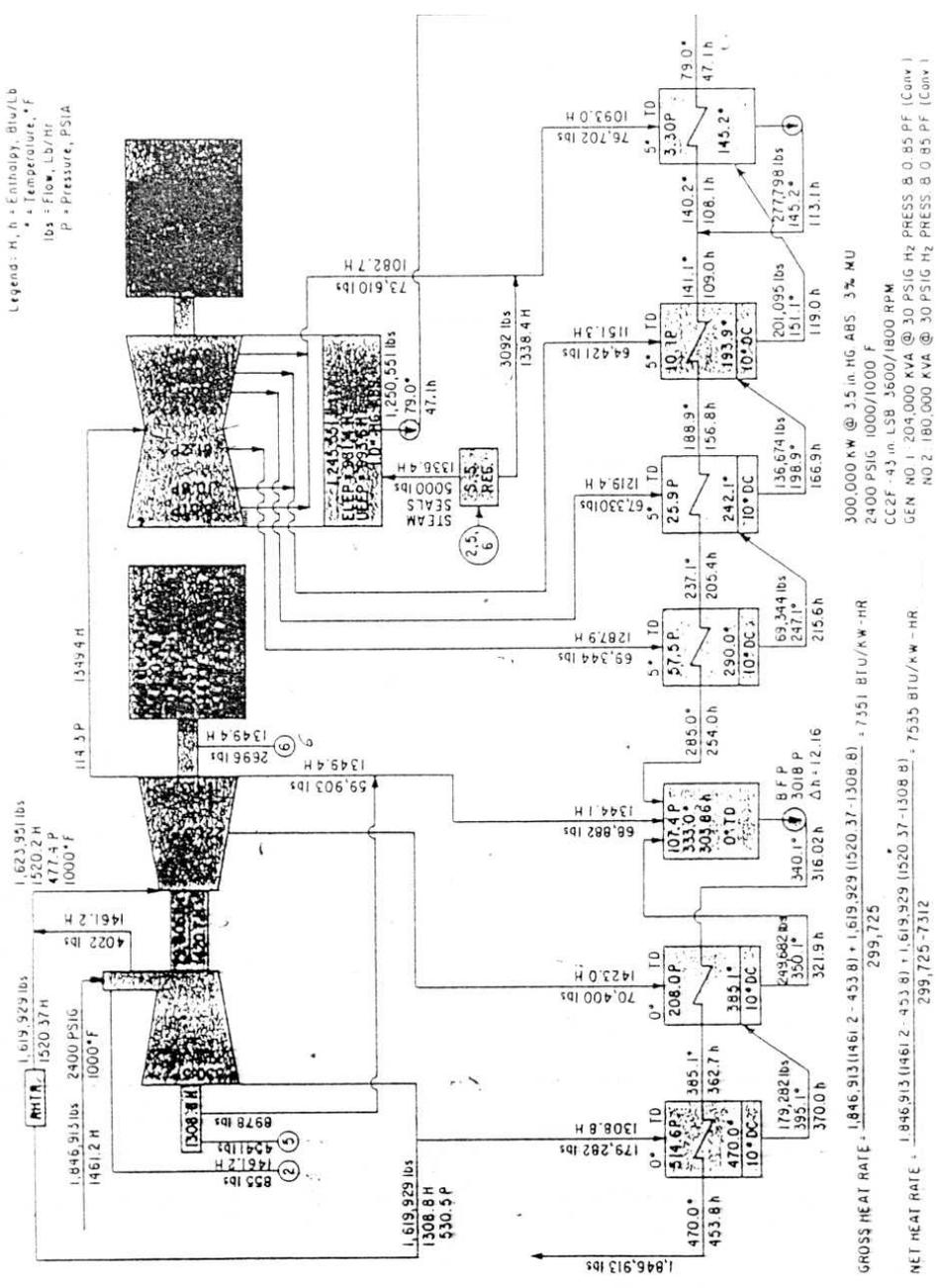


Fig. 26. Construction of expansion lines for nonreheat sections

Legend: H, h = Enthalpy, Btu/Lb
 T = Temperature, °F
 \dot{m} = Flow, Lb/HR
 P = Pressure, PSIA



GROSS HEAT RATE = $1,846,913(1461.2 - 453.8) + 1,619,929(1520.37 - 1308.8) = 7551 \text{ BTU/KW-HR}$
 $299,725$

NET HEAT RATE = $1,846,913(1461.2 - 453.8) + 1,619,929(1520.37 - 1308.8) = 7535 \text{ BTU/KW-HR}$
 $299,725 - 7312$

300,000 KW @ 3.5 in. HG. ABS. 3% MU
 2400 PSIG 1000/1000 F
 CCF - 43 in. LSB 3600/1800 RPM
 GEN NO 1 204,000 KVA @ 30 PSIG H₂ PRESS. B 0.85 PF (Conv)
 NO 2 180,000 KVA @ 30 PSIG H₂ PRESS. B 0.85 PF (Conv)

Fig. 27 Heat balance diagram

Revised July 1974

APPENDIX II

ILLUSTRATIVE EXAMPLE OF CALCULATION PROCEDURE

The calculation of turbine-generator performance and heat balances is an iterative process. The final result may require several sets of efficiency calculations with accompanying heat balances. The calculations for each iteration are based on values obtained from the previous iteration. Initially, these values are assumed. The following set of calculations is assumed to be the final iteration used in preparing the heat balance shown in Fig. 27.

A. BASIC ASSUMPTIONS

1. Turbine Nameplate Rating: 300,000 kw at 3.5 in. Hg Abs - 3% MU
2. Turbine Type: Cross-compound double-flow, 3600/1800-rpm, 43-inch last-stage buckets with 247.7 sq ft total last-stage annulus area.
3. Steam Conditions: 2400-psig, 1000/1000 F, 1.0-in. Hg Abs
4. Feedwater Heating Cycle: Spen heaters; topheater at cold reheat point; heater terminal differences and heater arrangement as shown in Fig. 27; pressure drop from extraction stage to turbine flange is 3 percent and from turbine flange to heater is 3 percent; final feedwater temperature is approximately 470 F; no feedwater makeup; 10-percent reheater pressure drop; extraction stage pressures selected for optimum cycle performance.
5. Efficiency calculation on valve-best-point basis with a 41.1-inch pitch diameter, one-row governing stage, having 4 control valves.
6. Generator Rating: 3600 rpm 204,000 kva at 30 psig H₂
1800 rpm 180,000 kva at 30 psig H₂
Both conventionally cooled.
7. Heat balance to be calculated at 300,000 kw; estimated throttle flow required is 1,846,913 lb/hr.
8. Known values from a design flow heat balance (or assumed):
Design throttle flow is 2,115,387 lb/hr.
Intercept valve pressure at design flow is 543.7 psia.
High-pressure section exhaust pressure at design flow is 504.1 psia (10-percent reheater pressure drop).
Design reheat section flow is 1,884,956 lb/hr (includes packing flow).
9. Estimated values from previous iteration:
Pressure ahead of intercept valves = 477.4 psia
High-pressure section exhaust pressure is
477.4/(1-.1) = 530.5 psia (10-percent reheater pressure drop)
Crossover pressure to 1800-rpm section = 114.3 psia

B. CALCULATION OF HIGH-PRESSURE SECTION EFFICIENCY AND EXPANSION LINE END POINT

$$1. \text{ Throttle Flow Ratio (TFR)} = \frac{\text{Throttle Flow}}{\text{Design Throttle Flow}} = \frac{1,846,913}{2,115,387} = 0.873$$

Revised July 1974

49

APPENDIX II (Cont'd)

2. Base Efficiency - Table I, 3600-rpm Noncondensing, one-row governing stage
 $\eta_b = 87\%$

3. Correction for Volume Flow (Table I)

$$\% \Delta \eta = \frac{1,005,200}{Q_v} N_{\text{poorer}}$$

$$= \frac{1,005,200}{(2,115,387)(0.3184)} 1$$

$$= -1.49\%$$

$$\eta_{\text{corrected}} = 87.00 - 0.0149 (87.00) = 85.70\%$$

Where:

- Q = Design Throttle Flow = 2,115,387 lb/hr
- v = Specific Volume at Throttle Conditions = 0.3184 at 2415 psia, 1000 F
- N = Number of parallel flow sections at beginning of Expansion = 1

4. Correction for Governing Stage (Table I indicates Fig. 7)

Pitch diameter of governing stage = 41.1 inches

From Fig. 7 at 41.1 inches, $\% \Delta \eta = -0.36\%$

$$\eta_{\text{corrected}} = 85.70 - 0.0036 (85.70) = 85.39\%$$

5. Correction for Pressure Ratio (Table I indicates Fig. 5)

$$\frac{\text{Exhaust Pressure at Design}}{\text{Throttle Pressure}} = \frac{604.1}{2415} = 0.25$$

Design Volume Flow = 2,115,387 (0.3184) = 673,000 cu ft/hr (See Step 3)

From Fig. 6 at 0.25 and 673,000 $\% \Delta \eta = -1.33\%$

$$\eta_{\text{corrected}} = 85.39 - 0.0133 (85.39) = 83.83\%$$

6. Correction for Governing Stage at Partial Load (Table I indicates Fig. 8)

Pitch Diameter of Governing Stage = 41.1 inches

Throttle Flow Ratio = 0.873 (Step 1)

From Fig. 8 at 0.873 and 41.1 $\% \Delta \eta = 0.23\%$

$$\eta_{\text{corrected}} = 83.83 + 0.0023 (83.83) = 84.02\%$$

7. Correction for Partial Load Operation (Table I indicates Fig. 9)

$$\frac{\text{Throttle Pressure}}{\text{Exhaust Pressure at Design}} = \frac{2415}{604.1} = 4$$

Throttle Flow Ratio = 0.873 (Step 1)

From Fig. 9 at 0.873 and 4 $\% \Delta \eta = -1.28\%$

$$\eta_{\text{corrected}} = 84.02 - 0.0128 (84.02) = 82.94\%$$

APPENDIX II (Cont'd)

8. Correction for Mean of Loops (Table I indicates Fig. 12)

This correction is optional and shown here only for illustrative purposes. It will not be used further in this example.

Number of Valves = 4
Throttle Flow Ratio = 0.873 (Step 1)

From Fig. 12 at 0.873 and 4 $\% \Delta \eta = -1.05 \%$

$\eta_{\text{corrected}} = 82.94 - 0.0105 (82.94) = 82.07 \%$

9. Expansion Line End Point, ELEP (Text and Fig. 24a)

Throttle Enthalpy (H_T) at 2415 psia and 1000 F = 1461.2 Btu/lb ($S_T = 1.5326$)

High-pressure Section Exhaust Pressure is 530.5 psia

Available Energy from Throttle to Exhaust = 1461.2 - 1277.5 = 183.7 Btu/lb

ELEP = $H_T - \eta_{\text{corrected}} (AE) = 1461.2 - 0.8294 (183.7) = 1308.8 \text{ Btu/lb}$

C. CALCULATION OF REHEAT SECTION EFFICIENCY AND EXPANSION LINE END POINT

1. Design Reheat Section flow including packing leakage from High-pressure Section is 1,884,956 lb/hr. Pressure ahead of intercept valves = 543.7 psia at design.

2. Base Efficiency, Table I 3600/1800-rpm Condensing, without Governing Stage, $\eta = 91.93\%$

3. Correction for Volume Flow (Table I)

$$\% \Delta \eta = \frac{1,270,000}{Q_v} N \text{ poorer}$$

$$= \frac{1,270,000}{1,884,956 (1.561)^1}$$

$$= 0.43\% \text{ poorer}$$

Where:

$Q = \text{Design Reheat Section Flow} = 1,884,956 \text{ lb/hr}$

$v = \text{Specific Volume at Design Pressure and Temperature ahead of Intercept Valves} = 1.561 \text{ cu ft/lb at } 543.7 \text{ and } 1000 \text{ F}$

$N = \text{Number of parallel flow sections at beginning of expansion} = 1$

$$\eta_{\text{corrected}} = 91.93 - 0.0043 (91.93) = 91.53\%$$

4. Correction for Initial Pressure and Temperature (Table I indicates Fig. 14)

Pressure ahead of Intercept Valves = 477.4 psia $\neq 543.7 \text{ psia at design}$

Temperature ahead of Intercept Valves = 1000 F

From Fig. 14 at 477.4 and 1000, $\% \Delta \eta = 0.15 \%$

$$\eta_{\text{corrected}} = 91.53 + 0.0015 (91.53) = 91.66 \%$$

APPENDIX II (Cont'd)

5. Correction for Substitution of 1800-rpm Low-pressure Section (Table I and Fig. 25)

$$\begin{aligned} \% \Delta \eta &= 1.25 \left[\frac{AE_{1800}}{AE_{Rht Sect}} \right] \text{ better} \\ &= 1.25 \frac{369.2}{562.2} \text{ 1.5 in. hg} \\ &= 0.82\% \end{aligned}$$

Where:

$$AE_{Rht Sect} = \text{Available energy from bowl mixed enthalpy } (H_m) \text{ and intercept valve pressure to 1.5 in. Hg abs } (S_m = 1.7404) = 1518.5 - 956.3 = 562.2 \text{ Btu/lb}$$

$$AE_{1800} = \text{Available Energy from crossover pressure of 114.3 PSIA to 1.5 in. Hg abs along } S_m \text{ entropy line} = 1325.5 - 956.3 = 369.2 \text{ Btu/lb}$$

$$\eta_{corrected} = 91.66 + 0.0082 (91.66) = 92.41\%$$

6. Expansion Line End Point (ELEP) at 1.5 in. Hg abs (Text and Fig. 25)

$$\begin{aligned} \boxed{ELEP_{1.5}} &= H_m - \eta_{corrected} (AE_{Rht Section}) \\ &= 1518.5 - 0.9241 (562.2) = 999.0 \text{ Btu/lb} \end{aligned}$$

*x = Quality (h=970, p=1.5)
↳ y = 9.85%*

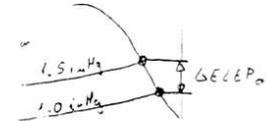
7. Change in Expansion Line End Point ($\Delta ELEP$) to 1.0 in. Hg abs (Fig. 15)

$$\text{Moisture } (Y) \text{ at } ELEP_{1.5} \text{ and 1.5 in. Hg Abs} = 9.87\%$$

$$\text{From Fig. 15 at 1.0 in. Hg Abs, } \Delta ELEP_0 = -23.9 \text{ Btu/lb}$$

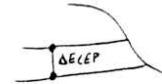
$$\begin{aligned} \Delta ELEP &= \Delta ELEP_0 (.87) (1 - .01Y) (1 - .0065Y) \\ &= -23.9 (.87) (1 - .0987) (1 - .0641) \\ &= -17.6 \text{ Btu/lb} \end{aligned}$$

*calculado para flujo seco (en la línea de saturación)
es para llevarlo al punto "real", no en la línea de saturación*



8. Expansion Line End Point (ELEP) at 1.0 in. Hg abs Exhaust Pressure

$$\begin{aligned} ELEP_{1.0} &= ELEP_{1.5} + \Delta ELEP \\ &= 999.0 + (-17.6) \\ &= 981.4 \text{ Btu/lb} \end{aligned}$$

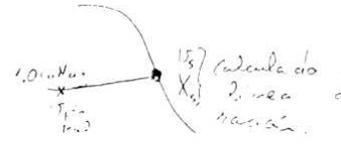


D. PACKING LEAKAGES, TABLE II

Refer to Appendix V for Calculation of Packing Leakages.

At this point the heat balance flows should be calculated. The condenser flow is found to be 1,245,551 lb/hr.

APPENDIX II (Cont'd)



E. EXHAUST LOSS AND USED ENERGY END POINT, FIG. 17

$$\text{Annulus Velocity, } V_{an} = \frac{Q_a v (1 - .01Y)}{3600 A_{an}}$$

$$V_{an} = \frac{1,245,551 (652.3) (1 - .1096)}{3600 (247.7)}$$

$$= 812 \text{ Ft/Sec}$$

Where:

- Q_a = Condenser flow = 1,245,551 Lb/hr
- v = Saturated ^{Seco}Dry Specific Volume at the desired exhaust pressure = 652.3 Cu ft/lb at 1.0 in. Hg Abs ($X = 1$)
- Y = Percent moisture at ELEP_{1.0} = 10.96% $f = 1.07$
 $n = 78.1$
- A_{an} = Annulus Area = 247.7 Sq Ft

From Fig. 17, Curve 3 at 812 Ft/sec; Exhaust Loss = 16.9 Btu/lb

$$\text{UEEP} = \text{ELEP}_{1.0} + (\text{Exhaust Loss}) (.87) (1 - .01Y) (1 - .0065Y)$$

$$= 981.4 + (16.9) (.87) (1 - .1096) (1 - .0711)$$

$$= 993.6 \text{ Btu/lb}$$

calcula das e de saturação
para transmissões al pto real

F. MECHANICAL LOSSES, FIG. 19

3600-rpm Shaft, Curve B at 204,000 x 1.1 = 224,400 kva; Loss = 854 kw
 1800-rpm Shaft, Curve B at 180,000 x 1.1 = 198,000 kva; Loss = 779 kw

At this point the heat balance power output should be calculated. The generator losses are estimated to be 3600 rpm: 1944 kw; 1800 rpm: 1588 kw; and the generator output estimated to be 3600 rpm: 158,890 kw at 0.85 PF; 1800 rpm: 140,835 kw at 0.85 PF.

G. GENERATOR LOSSES, FIGS. 20, 21 AND 23

Both generators are assumed to be operating at maximum hydrogen pressure

Operating KVA:

$$3600 \text{ rpm} = 158,890 / .85 = 186,929 \text{ kva}$$

$$1800 \text{ rpm} = 140,835 / .85 = 165,688 \text{ kva}$$

Percent of rated KVA:

$$3600 \text{ rpm} = 186,929 / 204,000 = 91.63\%$$

$$1800 \text{ rpm} = 165,688 / 180,000 = 92.05\%$$

Losses at rated hydrogen pressure, from equation on Fig. 20:

$$\text{Loss} = \frac{\text{Operating KVA} (K_1) (K_2)}{(100)}$$

From Fig. 20:

3600 rpm, Conventionally Cooled at 204 mva; $K_1 = 1.039$
 1800 rpm, Conventionally Cooled at 180 mva; $K_2 = 0.958$

APPENDIX II (Cont'd)

From Fig. 21:

3600 rpm, Conventionally Cooled at 91.63%; $K_2 = 1.0010$
 1800 rpm, Conventionally Cooled at 92.05%; $K_2 = 1.0006$

Losses at rated hydrogen pressure:

$$3600 \text{ rpm} = \frac{186,929 (1.039) (1.0010)}{(100)} = 1944 \text{ kw}$$

$$1800 \text{ rpm} = \frac{165,688 (0.958) (1.0006)}{(100)} = 1588 \text{ kw}$$

H. GENERATOR OUTPUT

Generator Output = Turbine Output - Mechanical Losses - Generator Losses
 Turbine Output Calculated by Heat Balance after Step E or F.

$$\text{Generator Output}_{3600} = 161,688 - 854 - 1944 = 158,890 \text{ kw}$$

$$\text{Generator Output}_{1800} = 143,202 - 779 - 1588 = 140,835 \text{ kw}$$

Note: These outputs should agree closely with those assumed in step G, or Step G and H repeated, until agreement is reached.

I. BOILER FEED PUMP POWER

Isentropic Boiler Feed Pump Enthalpy Rise from K and K Steam Tables at

$$333 \text{ F and } 107.4 \text{ psia to } 3018 \text{ psia} = 9.485 \text{ Btu/lb}$$

$$\text{Actual Boiler Feed Pump Enthalpy Rise} = \frac{\text{Isentropic Rise}}{\text{Pump Efficiency}} = \frac{9.485}{0.78} = 12.16 \text{ Btu/lb}$$

$$\text{Boiler Feed Pump Power} = \frac{\text{BFP Flow (BFP Rise)}}{\text{BFP Drive Efficiency (3412.75)}} = \frac{1,346,913 (12.16)}{0.90 (3412.75)} = 7312 \text{ kw}$$

APPENDIX III

EXHAUST LOSS AND USED ENERGY END POINT CALCULATION

WHEN v_{an} IS GREATER THAN 1400 FT/SEC

$\approx 426'22 \text{ m/s}$

METHOD	EXAMPLE
GIVEN:	
P = EXHAUST PRESSURE, In. HG abs.	P = 0.5 in. HG abs
ELEP _{1.5} = EXPANSION LINE END POINT AT 1.5 in. HG abs.	ELEP _{1.5} = 1028.8 BTU/lb
ELEP _p = EXPANSION LINE END POINT AT P Inches HG abs.	ELEP _{0.5} = 980.5 BTU/lb
Q _a = CONDENSER FLOW, LB/HR	Q _a = 750,000 lb/hr
A _{an} = ANNULUS AREA, SQ. FT.	A _{an} = 82.2
Y = EXPANSION LINE END POINT MOISTURE AT ELEP _{1.5} AND 1.5 in. HG abs., %	Y = 7% AT 1028.8 BTU/lb AND 1.5 in. HG abs.
LAST-STAGE BUCKET AND NO. ENDS	DF-26
FIND:	
1. Y _p = EXPANSION LINE END POINT MOISTURE AT ELEP _p AND P	Y _p = 10.0% AT 980.5 BTU/lb AND 0.5 in. HG abs.
2. V _{an} = ANNULUS VELOCITY AT P	$V_{an} = \frac{Q_a v_p (1 - .01Y_p)}{3600 A_{an}} = \frac{750,000 (1256.4) (.900)}{3600 (82.2)} = 2861 \text{ FT/SEC}$
3. P ¹ = EXHAUST PRESSURE TO GIVE SONIC VELOCITY (APPROX.), In. HG abs.	$P^1 = P \left(\frac{V_{an}}{1302} \right) = 0.5 \left(\frac{2861}{1302} \right) = 1.099 \text{ in. HG abs.}$
4. ΔELEP _{op} ¹ = CHANGE IN ELEP FROM 1.5 in. TO P ¹ (FIG. 15)	ΔELEP _{op} ¹ = -18.4 BTU/lb
5. ELEP _p ¹ = EXPANSION LINE END POINT AT P ¹ , BTU/lb	$\begin{aligned} \text{ELEP}_{p^1} &= \text{ELEP}_{1.5} + \Delta \text{ELEP}_{op}^1 (0.87) (1 - .01Y) (1 - .0065Y) \\ &= 1028.8 + (-18.4) (0.87) (.93) (.955) = 1014.8 \text{ BTU/LB} \end{aligned}$
6. v _p ¹ = SATURATED DRY SPECIFIC VOLUME AT P ¹ , CU. FT/LB.	v _p ¹ = 598 CU FT/lb
7. Y _p ¹ = EXPANSION LINE END POINT MOISTURE AT ELEP _p ¹ AND P ¹	Y _p ¹ = 7.9% AT 1014.8 BTU/lb AND 1.099 in. HG abs.
8. V _{an} ¹ = ANNULUS VELOCITY AT P ¹ NOTE: IF V _{an} ¹ IS GREATER THAN 1400, RECALCULATE STEPS 4 THROUGH 8 USING $P_{new}^1 = P_{old}^1 \left(\frac{V_{an}^1}{1302} \right)$	$V_{an}^1 = \frac{Q_a v_p^1 (1 - .01Y_p^1)}{3600 A_{an}} = \frac{750,000 (598) (.921)}{3600 (82.2)} = 1396 \text{ FT/SEC}$
9. EL = EXHAUST LOSS FROM APPROPRIATE EXHAUST LOSS CURVE AT V _{an} ¹	EL = 48.2 BTU/LB FROM 26-IN. EXHAUST LOSS CURVE
10. UEPP = USED ENERGY END POINT AT P	$\begin{aligned} \text{UEEP} &= \text{ELEP}_{p^1} - \text{EL} (0.87) (1 - .01Y_p^1) (1 - .0065Y_p^1) \\ &= 1014.8 - 48.2 (0.87) (.921) (.949) = 1051.2 \text{ BTU/lb} \end{aligned}$

TABLE III
TABULATED VALUES OF EXHAUST LOSS

LAST-STAGE BUCKET LENGTH, (IN.)	3600 RPM						1800 RPM			
	20*	23	26	30	33.5	20#	38	38	43	52
PITCH DIAMETER (IN.)	60	65.5	72.125	85	90.5	75	115	127.5	132	152
ANNULUS AREA/END, (SQ. FT.)	26.2	32.9	41.1	55.6	66.1	32.7	95.3	105.7	123.8	172.4
ANNULUS VELOCITY, FT/SEC										
128	25.81	53.65	64.09	99.21	106.7	76.3	42.85	51.63		75.69
150	20.84	46.50	56.15	88.90	95.43	67.90	37.80	45.50		65.64
175	16.10	40.08	47.58	77.88	84.28	58.70	32.63	38.38		55.68
200	12.75	34.35	40.55	67.65	74.38	50.50	28.40	32.65		48.05
250	8.33	24.95	29.75	50.85	56.72	37.70	21.60	23.80		36.00
300	5.53	18.68	22.48	38.38	43.00	28.75	16.60	17.43		27.80
350	4.09	14.23	17.28	29.83	32.82	22.25	13.13	12.91		21.45
400	3.73	10.90	13.55	23.15	25.40	17.10	10.55	9.80		16.30
450	3.95	8.65	10.82	17.80	19.40	13.45	9.00	7.97		12.50
500	4.90	7.45	8.85	13.60	14.66	11.00	8.13	7.07		9.60
550	6.57	7.12	8.30	11.20	11.93	9.70	7.89	7.02		8.55
600	8.65	7.55	8.50	10.45	10.80	9.60	8.40	7.72		8.47
650	10.95	8.70	9.50	10.75	10.75	10.23	9.57	9.00		9.05
700	13.56	10.45	11.25	12.00	11.65	11.75	11.40	11.00		10.38
800	19.35	15.45	15.82	16.20	15.20	15.95	16.25	16.25		14.45
900	25.62	21.07	21.13	21.17	19.80	21.17	21.97	21.97		19.56
1000	32.00	26.95	26.82	26.42	25.22	26.42	28.00	27.92		25.07
1100	38.35	33.05	32.82	32.10	31.06	32.10	34.00	34.00		30.82
1200	44.35	38.75	38.50	37.55	36.70	37.55	39.80	39.92		36.40
1300	49.90	44.15	43.80	42.70	42.05	42.70	45.18	45.25		41.78
1400**	54.70	48.90	48.30	47.35	46.69	47.35	49.90	49.85		46.50

*Also use for 14.3 X 52.4; 16.5 X 57.5; 17 X 52-inch last-stage buckets

**For V_{an} beyond 1400 ft/sec, see Appendix III

#Also use for 14.3 X 59.7; 16.8 X 63.2-inch last-stage buckets; for high back pressure units only.

APPENDIX IV
 MATHEMATICAL EQUATIONS AND DATA REPRESENTING
 THE FIGURES REQUIRED TO DETERMINE EFFICIENCIES AND LOSSES
 FOR NOMENCLATURE, SEE APPENDIX I

FIGURE NUMBER	DEPENDENT VARIABLE Z	INDEPENDENT VARIABLE X Y	TYPE OF FUNCTION	COEFFICIENTS
2	$\% \Delta \eta$	$0.625 - \frac{P_1}{P_T}$ Q_v	$\left(\begin{matrix} \text{UNIV.} \\ \text{POLY.} \\ \text{(A)} \end{matrix} \right) + \left(\begin{matrix} \text{UNIV.} \\ \text{POLY.} \\ \text{(B)} \\ \frac{1}{Q_v} \end{matrix} \right)$	$A_0 = 0$ $A_1 = -1.6649986$ $A_2 = -22.538964$ $A_3 = 19.464851$ $B_0 = 0$ $B_1 = 798267.50$ $B_2 = -7540.7$ $B_3 = -154269.4$
3	$\% \Delta \eta$	$\log_{10}(\text{TFR})$	UNIVARIATE POLYNOMIAL	$A_0 = 0$ $A_1 = 2.4462684$ $A_2 = -41.904570$ $A_3 = -164.13062$ $A_4 = -485.99735$ $A_5 = -674.41251$ $A_6 = -342.16474$
4	$\% \Delta \eta$	TFR $0.625 - \frac{P_1}{P_T}$	BIVARIATE POLYNOMIAL	$A_{01} = A_{02} = A_{03} = A_{04} = 0$ $A_{10} = -24.899722$ $A_{11} = 63.299521$ $A_{12} = -75.518221$ $A_{13} = 37.084109$ $A_{14} = -17.632581$ $A_{30} = -308.12317$ $A_{31} = -269.89024$ $A_{20} = -0.71362812$ $A_{21} = -23.404163$ $A_{22} = 126.77247$ $A_{23} = -102.17991$ $A_{24} = 594.36987$ $A_{32} = -269.89024$ $A_{33} = -269.89024$
5	$\% \Delta \eta$	TFR	UNIVARIATE POLYNOMIAL	$A_0 = -1.771$ $A_1 = 3.475$ $A_2 = -3.389$ $A_3 = 1.445$
6	$\% \Delta \eta$	$\frac{P_{xD}}{P_T}$ $\ln Q_v$	BIVARIATE POLYNOMIAL	$A_{00} = 11.151$ $A_{01} = -63.0$ $A_{10} = -0.50091$ $A_{11} = 2.83$
7	$\% \Delta \eta$	PD	UNIVARIATE POLYNOMIAL	$A_0 = 4.37$ $A_1 = -0.115$
8	$\% \Delta \eta$	TFR PD	BIVARIATE POLYNOMIAL	$A_{00} = -21.8085$ $A_{01} = 21.8085$ $A_{10} = 0.573908$ $A_{11} = -0.573908$
9	$\% \Delta \eta$	TFR $\ln \left(\frac{P_T}{P_{xD}} \right)$	BIVARIATE POLYNOMIAL	$A_{00} = -60.75$ $A_{01} = 66.35$ $A_{02} = 29.75$ $A_{03} = -35.85$ $A_{10} = 17.50$ $A_{11} = -20.02$ $A_{12} = -0.525$ $A_{13} = 3.045$
10	$\% \Delta \eta$	$\frac{P_{xD}}{P_T}$ $\ln Q_v$	BIVARIATE POLYNOMIAL	$A_{00} = 25.665$ $A_{01} = -145.0$ $A_{10} = -1.33281$ $A_{11} = 7.53$
11	$\% \Delta \eta$	$1 - \text{TFR}$ $\frac{P_T}{P_{xD}}$	BIVARIATE POLYNOMIAL	$A_{00} = A_{10} = A_{20} = A_{30} = 0$ $A_{01} = 42.676909$ $A_{02} = -89.391147$ $A_{03} = 9.0376638$ $A_{11} = -26.221836$ $A_{12} = 25.549385$ $A_{13} = 8.8283868$ $A_{21} = 40479550$ $A_{22} = -1.4725197$ $A_{23} = -4.0183332$ $A_{31} = -0.14502211$ $A_{32} = -0.18580363$ $A_{33} = 0.42657518$
12	$\% \Delta \eta$	TFR NV	$\frac{P_{xD}}{.26 P_T}$ (BIV. POLY.)	$A_{00} = -5.4$ $A_{01} = 4.395$ $A_{10} = 0.45$ $A_{11} = -0.36625$

APPENDIX IV (Cont'd)

FIGURE NUMBER	DEPENDENT VARIABLE Z	INDEPENDENT VARIABLE X Y	TYPE OF FUNCTION	COEFFICIENTS	
13	η	$\frac{Q_v}{P_B/P_{xD}}$	$A + \left(\frac{B}{Q_v}\right)$	$A = 90.799 + 0.7474 \left[\ln \left(\frac{P_B}{P_{xD}} - 0.3 \right) \right] - \left[\frac{0.5454}{\ln \left(\frac{P_B}{P_{xD}} - 0.3 \right)} \right]$ $B = -505000 + 77568 \left[\ln \left(\frac{P_B}{P_{xD}} + 0.8 \right) \right] - \left[\frac{1262500}{\ln \left(\frac{P_B}{P_{xD}} + 0.8 \right)} \right]$	
14	$\frac{3}{8}\Delta\eta$ (1150 < H < 1750)	P H	BIVARIATE POLY-NOMIAL FUNCTION OF X AND H*, WHERE X = LOG ₁₀ P; H* = H UNLESS S > 2.0041 THEN H* = Smaller OF H OR H _T , WHERE H _T = 1154 + 80X + 88X ²	A ₀₀ = 28.232252 A ₀₁ = -92.390491 A ₀₂ = -625.79590 A ₀₃ = 207.23010 A ₀₄ = 70.251642 A ₀₅ = -22.516388 A ₁₀ = -0.047796308 A ₁₁ = 1.2844571 A ₁₂ = 0.38556961 A ₁₃ = -0.039652999 A ₁₄ = -0.27180357 A ₁₅ = 0.064869467 A ₄₀ = -0.50719109 X 10 ⁻⁹ A ₄₁ = 0.26393497 X 10 ⁻⁹ A ₄₂ = 0.38021911 X 10 ⁻¹⁰	A ₂₀ = -0.69791427 X 10 ⁻³ A ₂₁ = -0.17037268 X 10 ⁻² A ₂₂ = 0.86563845 X 10 ⁻³ A ₂₃ = -0.59510660 X 10 ⁻³ A ₂₄ = 0.39705804 X 10 ⁻³ A ₂₅ = -0.73533255 X 10 ⁻⁴ A ₃₀ = 0.12050837 X 10 ⁻⁵ A ₃₁ = 0.26826382 X 10 ⁻⁶ A ₃₂ = -0.67887771 X 10 ⁻⁶ A ₃₃ = 0.52886157 X 10 ⁻⁶ A ₃₄ = -0.24106229 X 10 ⁻⁶ A ₃₅ = 0.37881801 X 10 ⁻⁷ A ₄₃ = -0.10149993 X 10 ⁻⁹ A ₄₄ = 0.47757232 X 10 ⁻¹⁰ A ₄₅ = -0.70989561 X 10 ⁻¹¹
15	$\Delta E L E P_0$	ln (INCHES HG. ABS.)	UNI-VARIATE POLY-NOMIAL	A ₀ = -23.984811 A ₁ = 57.862440.	A ₂ = 3.1849404
16 17 18	EL	V _{an}	TABULAR VALUES USE INTER-POLATION ROUTINE	SEE TABLE III FOR VALUES	
19	ML	KVA **	UNIVARIATE POLYNOMIAL + ΔML	SINGLE CASING AND TCDF AT 3600 rpm 22,000 to 110,000 KVA** A ₀ = -2.78061 A ₁ = +3.69116 X 10 ⁻³ A ₂ = +6.63281 X 10 ⁻⁹ A ₃ = -6.98336 X 10 ⁻¹⁴ SINGLE CASING AND TCDF AT 3600 rpm 110,000 KVA** AND ABOVE A ₀ = +105.1185 A ₁ = +2.5948815 X 10 ⁻³ ALL OTHER TYPES A ₀ = +220.0 A ₁ = +2.82 X 10 ⁻³	

** For conventionally cooled generators use generator rating multiplied by 1.1

APPENDIX IV (Cont'd)

FIGURE NUMBER	DEPENDENT VARIABLE Z	INDEPENDENT VARIABLE X Y	TYPE OF FUNCTION	COEFFICIENTS
19 (cont'd)	ΔML	KVA **	UNIVARIATE POLYNOMIAL	BELOW 500,000 KVA ** $\Delta ML = 0.0$ 500,000 KVA** AND ABOVE $A_0 = +570.0$ $A_1 = -3.04 \times 10^{-3}$ $A_2 = +3.8 \times 10^{-9}$
20	K_1	ln (KVA)	$\Delta +$ UNIVARIATE POLYNOMIAL	CONVENTIONALLY COOLED 3600 rpm $\Delta = 0.0$ 1800 rpm $\Delta = -0.095$ $A_0 = -0.34293488 \times 10^0$ $A_1 = 0.89126466 \times 10^0$ $A_2 = -0.10877851 \times 10^0$ $A_3 = 0.36908686 \times 10^{-2}$
			$\Delta +$ RATIONAL FUNCTION	CONDUCTOR COOLED 3600 rpm $\Delta = 0.0$ 1800 rpm $\Delta = -0.10$ $A_0 = 0.18479050 \times 10^1$ $A_1 = 0.56681377 \times 10^{-1}$ $A_2 = -0.26652558 \times 10^{-1}$ $A_3 = 0.10759713 \times 10^{-2}$ $B_1 = -0.49564581 \times 10^{-1}$
21	K_2	$\frac{\% KVA}{100}$	RATIONAL FUNCTION	CONVENTIONALLY COOLED 3600 AND 1800 rpm $A_0 = -0.20525271 \times 10^2$ $A_1 = -0.35019203 \times 10^2$ $A_2 = 0.25514027 \times 10^2$ $A_3 = -0.25182400 \times 10^2$ $B_1 = -0.56218495 \times 10^2$
			CONDUCTOR COOLED 3600 rpm	1800 rpm $A_0 = -0.17535943 \times 10^2$ $A_1 = -0.15515139 \times 10^2$ $B_1 = -0.65338483 \times 10^2$ $B_2 = 0.31286714 \times 10^2$ $A_0 = -0.22727744 \times 10^0$ $A_1 = -0.24939976 \times 10^2$ $A_2 = 0.34283283 \times 10^2$ $A_3 = -0.35808901 \times 10^2$ $A_4 = 0.90544189 \times 10^1$ $B_1 = -0.18638959 \times 10^2$
23	ΔKW	KVA	RATIONAL FUNCTION	CONVENTIONALLY COOLED 3600 rpm $A_0 = 0.10949623 \times 10^2$ $A_1 = 0.12755560 \times 10^{-3}$ $A_2 = 0.71452351 \times 10^{-8}$ $A_3 = -0.58333495 \times 10^{-14}$ $B_1 = 0.56299996 \times 10^{-5}$
			UNIVARIATE POLYNOMIAL	CONVENTIONALLY COOLED 1800 rpm $A_0 = 0.33807252 \times 10^2$ $A_1 = 0.25207663 \times 10^{-3}$
			CONDUCTOR COOLED 3600 rpm	1800 rpm $A_0 = 0.12 \times 10^3$ $A_1 = 0.96 \times 10^{-4}$ $A_0 = 0.40 \times 10^2$ $A_1 = 0.96 \times 10^{-4}$

**For conventionally cooled generators use generator rating multiplied by 1.1

Revised July 1974

59

APPENDIX IV (Cont'd)

FIGURE NUMBER	DEPENDENT VARIABLE Z	INDEPENDENT VARIABLE X Y	TYPE OF FUNCTION	COEFFICIENTS
24a, 24b, 25, 26	s	h	COMPLEX	<p>THE EXPANSION LINE IS DEFINED BY THE ENTHALPY AND ENTROPY AT THE TWO END CONDITIONS, A AND B. THE EXPRESSION FOR THE ENTROPY, s, ON THE EXPANSION LINE AT THE ENTHALPY, h, IS:</p> <p>HIGH PRESSURE AND INTERMEDIATE SECTIONS</p> $s = s_B + (s_A - s_B)(h - h_B)/(h_A - h_B)$ <p>REHEAT AND NONREHEAT SECTIONS</p> $s = \frac{1}{10} \left(\frac{h_B - (h + H_0)}{371.0} \right) + R_O(h - h_B) + s_B - (0.177 \times 10^{-1})$ <p>WHERE:</p> <p>$H_0 = 650.0$ for reheat sections $H_0 = 450.0$ for nonreheat sections</p> $R_O = \frac{(s_A - s_B) + (0.177 \times 10^{-1}) - \left[\frac{1}{10} \left(\frac{h_B - (h_A + H_0)}{371.0} \right) \right]}{(h_A - h_B)}$ <p>AN ITERATIVE PROCEDURE AND STEAM PROPERTIES FORMULATIONS MAY BE USED TO CONVERGE ON THE DESIRED POINT WHEN PRESSURE IS THE KNOWN VARIABLE.</p>

APPENDIX V

CALCULATION OF SHAFT END PACKING LEAKAGES

To simplify the calculation of leakage flows, the information for evaluating Martin's formula(7) for typical packings used by the authors' company is given in the form of a packing leakage constant, C, in Table II. The C factor is represented by:

$$F = 25KA \sqrt{\frac{1-(P_2/P_1)^2}{N-\ln(P_2/P_1)}} \sqrt{\frac{P_1}{v_1}} = C \sqrt{\frac{P_1}{v_1}}$$

where:

- F = Packing flow in pounds per hour
- K = Factor for packing type (see Fig. 31)
- A = Leakage Area in square inches =
 η (Packing diameter, inches) (clearance, inches)
- P₁ = Initial pressure in packing in psia
- P₂ = Final pressure in packing in psia
- v₁ = Initial specific volume in cubic feet per pound
- N = Number of stationary packing teeth
- C = Packing Leakage Constant from Table II

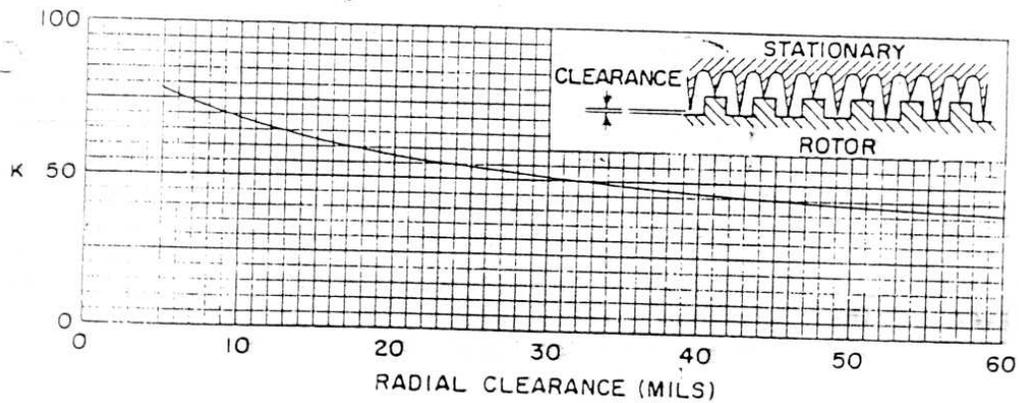
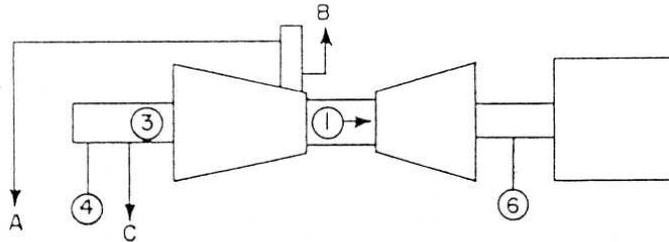


Fig. 31. High-pressure packing coefficient, K

Using heat balance diagram, Fig. 27, as an example, the packing leakages may be found as follows:



VALVE STEM LEAKAGES (Table II - Reheat Subcritical)

Leakage	Packing Leakage Constant "C"	P ₁ Psia	H ₁ Btu/lb	v ₁ Cu ft/lb	$\sqrt{P_1/v_1}$	F Lb/hr
Total Flow	56	2415	1461.2	0.3184	87.09	4877
A 2nd LO	50	477.4	1461.2	1.6327	17.10	855
B 1st LO	By Difference					4022

SHAFT END PACKINGS (Table II, Diagram B, Line B)

Leakage	Packing Leakage Constant "C"	P ₁ Psia	H ₁ Btu/lb	v ₁ Cu ft/lb	$\sqrt{P_1/v_1}$	F Lb/hr
1	500	1576	1420.7	0.4588	58.612	29306
3	620	530.5	1308.8	1.1158	21.805	13519
4	970	114.3	1308.8	5.215	4.6814	4541
C	By Difference	3 4				8978
6	600	114.3	1349.4	5.662	4.493	2696

STEAM SEAL REQUIREMENTS (Table II, 100 mw and above; 1800 rpm)

Total Available = 855 + 4541 + 2696 = 8092

Total Required 2 x 2500 = 5000

To Low-pressure Heater (Available - Required) = 3092 Lb/hr

