

Universidad de Matanzas  
Facultad de Ciencias Técnicas  
Departamento de Química



## Trabajo de Diploma

Trabajo de Diploma en opción al título de Ingeniero Química

*TÍTULO: Modelo para el cálculo de espesor óptimo de aislante en líneas de vapor.*

Autor: Adrián de la Fe Duarte

Tutores: Dr.C. Agustín Benítez Hernández

Ing. María Elena Hernández Serra

Matanzas 2022,

*“La ciencia es un gran juego. Es inspiradora y refrescante, y el campo de juego es el universo entero.”*

*Isidor Isaac Rabi*

*Premio Nobel de Física, 1944*

*“Nunca consideres el estudio como una obligación, sino como una oportunidad de para penetrar en el bello y maravilloso mundo del saber.”*

*Albert Einstein*

*Premio Nobel de Física, 1921*

# Nota de aceptación

---

---

---

---

---

---

---

---

Presidente del Tribunal

---

Miembro del Tribunal

---

Miembro del Tribunal

---

Calificación

---

Ciudad, fecha

# *Dedicatoria*

*A mi madre y abuela.*

*A mi abuelo.*

*A mí porque me lo merezco.*

## Declaración de Autoridad

Yo, Adrián de la Fe Duarte, me declaro como único autor de esta investigación realizada en la Universidad de Matanzas ``Camilo Cienfuegos``, como requisito parcial para optar por el título de Ingeniero Químico y autorizo que la misma sea utilizada por la mencionada institución como material de consulta. Para que así conste, debajo firma:

---

Adrián de la Fe Duarte

## *Agradecimientos*

*No puedo dejar de pasar este momento de significativa importancia en mi vida sin agradecer sinceramente a muchas personas vinculadas a él.*

- ✓ A mi madre Caridad que desde pequeño me enseñó lo importante que es estudiar, por ser mi guía, mi apoyo en cualquier situación, por su confianza siempre en mí.*
- ✓ A mi abuelo Clemente que aunque no está conmigo en estos momentos, fue fundamental en mi vida gracias a sus enseñanzas y puedo decir que soy el hombre que soy ahora gracias a él.*
- ✓ A mi abuela Aniceta por estar pendiente a todo y enseñarme que la familia es muy importante y siempre estar ahí para mí.*
- ✓ A cada miembro de mi familia, por ser incondicionales y estar siempre dispuestos en los que los he necesitado.*
- ✓ A mi novia Hazel por ese gran amor y apoyo en estos años y por estar tan involucrada en mi formación como profesional y mi bienestar.*
- ✓ A todo mi grupo en especial a Agustín y Alexandra, pues gracias a todos ellos han contribuido a ayudarme de alguna manera en este viaje de 4 años por la carrera de Ingeniería Química.*
- ✓ A mis amigos de siempre por cada fiesta, risa, apoyo, momentos compartidos y que sepan que son parte de mí.*
- ✓ A mis tutores Agustín y María Elena por su comprensión y paciencia durante esta dura etapa, la cual siempre les estaré agradecido.*
- ✓ A todo el cuerpo de profesores que gracias a ellos con sus consejos, regaños, alientos hayan hecho en mí ser una mejor persona y ser un buen profesional en el futuro.*
- ✓ A todos*

*Muchas gracias*

## RESUMEN

---

La presente investigación se realiza en la División Territorial de Comercialización de Combustibles, Matanzas, específicamente en el área Crudo y Suministro. Tiene como fin, la determinación del espesor óptimo de aislante en las líneas de vapor de dicha área. Se hace uso del *software* Matlab donde se obtuvo que el espesor óptimo de aislante de 3 mm es el más conveniente en cuanto a la optimización termodinámica como la económica.

## ABSTRACT

---

This research is carried out in the División Territorial de Comercialización de Combustibles, Matanzas, specifically in the Crudo y Suministro area. Its purpose is to determine the optimum thickness of insulation in the steam lines of this area. The Matlab software is used, where it was obtained that the optimum insulation thickness of 3 mm is the most convenient in terms of thermodynamic and economic optimization.



# INTRODUCCIÓN

---

El petróleo es una fuente de energía de gran importancia en el mundo moderno.

El petróleo cuando se trata del sector transporte, la gasolina y el diesel que se obtiene del petróleo, son los combustibles más importantes. Plásticos, fibras plastificantes, colorantes, y más series de productos que son creados en la industria petroquímica se elaboran a base de petróleo. También decir que a lo que concierne a la iluminación, el uso para la generación de electricidad el petróleo no puede dejar de ser mencionado, por lo que aquí apreciamos este combustible tiene una gran importancia en la vida cotidiana.

Las calderas son máquinas que constan de un recipiente a presión cerrado en el que un líquido, generalmente agua, se calienta y/o vaporiza bajo presión o vacío mediante la aplicación de calor derivado de la combustión de combustibles fósiles o electricidad.

El vapor saturado es utilizado como la fuente de calentamiento para fluido en intercambiadores de calor, reactores, rehervidores, y otros tipos de equipos de transferencia de calor.

En un intercambiador de calor, el vapor eleva la temperatura del producto por transferencia de calor, el cual después se convierte en condensado y es descargado a través de una trampa de vapor.

El objetivo del calentamiento del petróleo pesado a través de una caldera es, disminuir su viscosidad y facilitar así su manipulación, evitando presiones excesivas en las líneas de la bomba.

La División Territorial de Comercialización de Combustibles, Matanzas tiene como misión comercializar y brindar servicios especializados asociados al petróleo crudo y sus derivados en el territorio nacional, con estándares de calidad certificados, que asegure la competitividad, seguridad ambiental y satisfacción para los clientes.

Debido al deterioro o inexistencia de aislamiento térmico en la red de distribución de vapor y las pérdidas de calor que trae consigo, resulta difícil alcanzar las temperaturas deseadas en el proceso.

Por lo que se, propone como problema científico de la investigación: ¿Cómo determinar el espesor óptimo de aislante en las líneas de vapor de área Crudo y Suministro en la División Territorial de Comercialización de Combustibles, Matanzas?

Como posible solución al problema se plantea la siguiente hipótesis de trabajo: Si se selecciona correctamente un modelo matemático para determinar el espesor de aislante, se podrá optimizar dicho espesor hasta obtener el más factible económicamente.

Para dar cumplimiento a la hipótesis planteada, se propone el siguiente objetivo general: Determinar el espesor óptimo de aislante en las líneas de vapor del área de Crudo y Suministro en la División Territorial de Comercialización de Combustibles, Matanzas.

Para cumplir el objetivo propuesto anteriormente se plantean los siguientes objetivos específicos:

1. Seleccionar, según la bibliografía consultada el tipo de aislante más adecuado para las condiciones de trabajo del área en cuestión.
2. Realizar un levantamiento del estado, longitud y diámetro de las líneas de vapor del área objeto de estudio.
3. Seleccionar un modelo matemático para el cálculo del espesor de aislante en las tuberías, que responda a las condiciones reales del proceso.
4. Desarrollar la programación de dicho modelo a través de la implementación del *software* MatLab.

# TABLA DE CONTENIDO

---

CAPÍTULO 1: ANÁLISIS BIBLIOGRÁFICO .....	1
1.1    Propiedades y usos del petróleo.....	1
1.2    Riesgos ante derrames de petróleo en el medio ambiente .....	3
1.3    Métodos de obtención de petróleo .....	4
1.3.1 Los métodos más empleados para lograr la extracción del petróleo:.....	5
1.4    Transporte de crudo.....	6
1.5    Calentamiento de crudo .....	9
1.5.1    Calentadores de crudo .....	10
1.6    Líneas de vapor .....	12
1.6.1    Ventajas de usar vapor saturado para calentamiento .....	13
1.7    Aislamiento térmico.....	15
1.7.1    Tipos de aislantes térmicos .....	15
CAPÍTULO 2: METODOLOGÍA DE LA INVESTIGACIÓN.....	19
2.1    Descripción del caso de estudio .....	19
2.2    Caracterización de las líneas de vapor en el área .....	21
2.3    Pérdidas de calor en tuberías no aisladas .....	21
2.4    Pérdidas de calor en tuberías aisladas .....	23
2.5    Implementación en MatLab del modelo matemático .....	24
2.6    Valoración económica sobre el uso del aislante .....	24
CAPÍTULO 3: ANÁLISIS DE LOS RESULTADOS .....	26
3.1    Calor perdido en tubería desnuda .....	26
3.2    Calor perdido en tubería aislada.....	26
3.2.1 Pérdidas de calor en función del espesor de aislante .....	27
3.2.2 Selección del espesor de aislante en función de la termodinámica.....	28
3.3    Selección del espesor de aislante en función de la temperatura superficial ..	29
3.4    Selección del espesor de aislante en función del costo total .....	30
CONCLUSIONES.....	34
RECOMENDACIONES.....	35
REFERENCIAS BIBLIOGRÁFICAS .....	36
ANEXOS .....	1

# CAPÍTULO 1: ANÁLISIS BIBLIOGRÁFICO

---

## 1.1 Propiedades y usos del petróleo

El petróleo es una mezcla compleja de hidrocarburos con pesos y estructuras moleculares diversos que comprenden tres grupos químicos principales: parafínico, nafténico y aromático. Estos hidrocarburos van desde sustancias simples altamente volátiles hasta ceras complejas y compuestos asfálticos que no pueden ser destilados. Pueden combinarse con oxígeno, nitrógeno, azufre, vanadio, níquel, sales minerales, etc. en distintas proporciones. Las características de la mayoría de los crudos están comprendidas dentro de las siguientes gamas: (Ministerio para la Transición Ecológica, 2020)

- Densidad relativa,  $kg/m^3$  a 15 °C: 800 a 980
- Punto de ebullición inicial, °C: 30 a 125
- Viscosidad cinemática, centistokes (cSt) a 40°C: 3 a 100 (pero puede llegar hasta 20 000)
- Punto de fluencia, °C: -30 a +25 (pero puede ser inferior o llegar a 40)
- Punto de inflamación (Abel) °C: -18 a 190
- Azufre, % en peso: 0,08 a 5
- Ceras, % en peso: hasta 15
- Asfaltenos, % en peso: hasta 5
- Vanadio, ppm V: 5 a 170

A menudo las denominaciones del petróleo se derivan o asocian con su origen geográfico. Cada crudo posee propiedades físico-químicas singulares que pueden variar tanto entre hidrocarburos de la misma zona como entre distintas regiones del mundo. Sus propiedades físicas y cómo se comportan cuando se produce un derrame, junto con sus propiedades químicas, determinantes de su toxicidad, son factores importantes que influyen en la respuesta al derrame de hidrocarburos y al impacto que éstos puedan tener en los distintos recursos marinos(Philp, 2020).

En cuanto al uso del petróleo podemos verlo en diferentes sectores como:(Global estaciones de servicio, 2022)

#### 1. En el sector del transporte

Este uso es muy común, el petróleo aún sigue siendo dominante en el sector de los combustibles, aunque cada vez hay más vehículos que funcionan con gas y otro tipo de combustibles alternativos, cuando se trata del sector transporte, la gasolina y el diésel que se obtiene del petróleo, son los combustibles más importantes.

#### 2. Materia prima en la industria petroquímica

Plásticos, fibras plastificantes, colorantes, explosivos, anticongelantes, abonos, detergentes, cauchos sintéticos y otra serie de productos que son creados en la industria petroquímica se elaboran a base de petróleo. En general un 60% de los productos químicos y al menos un 80% de los orgánicos elaborados en esta industria, derivan del petróleo.

#### 3. En el sector farmacéutico

Muchas de las cremas y productos farmacéuticos en general, que de seguro tienes en uso en este momento en casa, están elaborados a base de petrolato y aceite mineral, subproductos que se obtienen del petróleo.

Para afecciones como la psoriasis y la caspa se suelen usar productos a base de alquitrán, que también se obtiene del petróleo.

#### 4. Fabricación de neumáticos

Aunque el petróleo quiera ser eliminado de la industria automotriz, al menos en el sector de los combustibles, cuando se trata de la fabricación de neumáticos, seguirá teniendo una importante presencia.

#### 5. Sector de la iluminación y calefacción

La calefacción es indispensable en casi todos los hogares y la base del funcionamiento de muchos de estos sistemas está en el petróleo. Ya que, de este se obtiene los aceites que se usan en dicho sector, en lo que concierne a

la iluminación, el uso para la generación de electricidad no podía quedar de lado.

Los diferentes usos del petróleo se clasifican en:

Energéticos: combustibles específicos para el transporte, la agricultura, la industria, la generación de corriente eléctrica y para uso doméstico.

Productos especiales: lubricantes, parafinas, asfaltos, grasas para vehículos y productos de uso industrial.

Materias primas para la industria petroquímica básica: plásticos, acrílicos, guantes, pinturas, envases diversos, detergentes, fibras textiles, insecticidas, etc (Wu y Chen, 2019).

## **1.2 Riesgos ante derrames de petróleo en el medio ambiente**

Los invertebrados y macrófitos marinos son sensibles a los efectos tóxicos del petróleo. Dependiendo de la intensidad, la duración y las circunstancias de la exposición, pueden sufrir altos niveles de mortalidad inicial junto con efectos subletales prolongados que pueden actuar a nivel individual, poblacional y comunitario. También puede haber perturbación de ecosistemas y desplazamientos de especies animales; riesgos de contaminación por accidentes y explosiones; además hay más riesgos de alteración de los ecosistemas marinos porque hay mayores posibilidades de dispersión de contaminantes que en el medio terrestre (Keesing *et al.*, 2018).

En algunas circunstancias, la recuperación de estos impactos puede llevar de años a décadas. Sin embargo, los efectos son variables, muchos factores pueden mitigar el grado de exposición, lo que significa que los impactos son moderados en muchos casos, y la recuperación se produce en pocos años. La exposición se ve afectada por un sinnúmero de factores, entre los que se incluyen: el tipo y la cantidad de petróleo, el grado de meteorización, la persistencia de la exposición, la aplicación de dispersantes u otras medidas de limpieza, el tipo de hábitat, la temperatura y la profundidad, las especies presentes y su estado de desarrollo o madurez, y los procesos de recolonización, especialmente el reclutamiento. Casi todos los derrames de hidrocarburos son únicos en cuanto a su impacto debido a los diferentes niveles de exposición y al tipo de hábitats,

comunidades y conjuntos de especies en el entorno receptor (Keesing *et al.*, 2018).

Los derrames de petróleo también pueden afectar a la salud humana. Estos efectos pueden depender de qué tipo de petróleo se derramó y dónde (en tierra, en un río o en el mar). Otros factores incluyen qué tipo de exposición y la cantidad de exposición que hubo. Las personas que limpian derrames están en mayor riesgo. Los problemas pueden incluir irritación cutánea y ocular, problemas neurológicos y respiratorios, y estrés (Wilson *et al.*, 2021).

### **1.3 Métodos de obtención de petróleo**

Los procesos que se emplean para la extracción de petróleo dependen en gran medida de factores como la ubicación del yacimiento o las características que presenta este yacimiento. Una vez que se ha detectado una zona con la presencia de petróleo se procede a hacer una perforación de cierta profundidad para llegar hasta él, en algunos casos estas perforaciones pueden ser de hasta 6 000 metros (Ingenieriaquimica.net, 2017).

A partir de aquí empieza un proceso de perforación-extracción donde se introduce y baja una especie de cañón y se perfora la tubería de revestimiento justo en el punto donde se ha detectado la bolsa. Lo habitual es que el petróleo fluya por los orificios que se han hecho y se pueda extraer mediante una tubería de un diámetro más pequeño. A esta tubería se la denomina tubería de producción (Ingenieriaquimica.net, 2017).

Se pueden dar dos situaciones: (Ingenieriaquimica.net, 2017)

- Que el yacimiento tenga energía propia que habitualmente viene generada por la propia presión subterránea y por los elementos que suelen acompañar al petróleo como son gas y agua. En este caso, el petróleo sale por sí solo en el momento de hacer la perforación. Lo que se suele hacer es instalar en la parte superior del pozo un equipo que se denomina árbol de navidad y que está compuesto por una serie de válvulas que sirven para ir regulando el paso del petróleo.

- En caso de que el yacimiento no tenga energía propia se usan métodos específicos de extracción. El método más habitual es el balancín mediante el cual a través de un balanceo permanente se acciona una bomba en el fondo del pozo que realiza la succión del petróleo hasta la superficie.

### **1.3.1 Los métodos más empleados para lograr la extracción del petróleo:**

- Método a percusión (Ingenieriaquimica.net, 2017)

Este método está ya casi en desuso y consiste en la utilización de trépano pesado que unido a una barra de referencia aumenta su peso. Esta barra se sujeta con un cable de acero conectado a un balancín. A través del balancín se transmite un movimiento alternativo de subida y bajada que es accionado por un motor. Con cierta frecuencia se procede a retirar el trépano para sacar los materiales mediante una herramienta denominada cuchara.

La gran desventaja de este método es que es un proceso muy lento y esto hace que cada vez se use menos. De hecho, su utilización se hace únicamente en yacimientos muy pequeños.

- Método a rotación. (Ingenieriaquimica.net, 2017)

Este método es algo más complejo que el anterior ya que mediante el trépano que es hueco, se atornilla a una serie de caños de acero que forman lo que se llama las barras de sondeo. Estas barras giran impulsadas por una mesa rotativa que se encuentra en la base de la torre y que está unida por una transmisión mediante cadena con los motores del cuadro de maniobras.

La llamada mesa rotativa presenta un agujero cuadrado en el centro en el que se desliza una columna de perforación que desciende a medida que el trépano avanza. En la parte más alta de la torre hay aparejos que hacen posible levantar y bajar los equipos. Es precisamente el movimiento de la mesa rotativa lo que hace que se ponga en marcha la perforación. Cuando es necesario se agregan nuevas barras de sondeo que se enroscan y miden en torno a nueve metros.



Este método es el más usado en la mayoría de las explotaciones de yacimientos. Esta operación descrita se va repitiendo tantas veces como es necesario. Debido al movimiento los materiales se arrastran hasta la superficie.

Una vez que la perforación se ha llevado hasta los 100 o 150 metros se entuba el pozo mediante una cañería metálica y cemento de fraguado rápido (proceso de cementación) de forma que se eviten derrumbes debido a filtraciones de las napas de agua que habitualmente se atraviesan.

- Perforación submarina (Ingenieriaquimica.net, 2017)

En muchos casos para poder aprovechar mayores yacimientos de petróleo se hacen excavaciones y perforaciones sobre el mar. Se trata de equipos que se instalan en una plataforma situada lejos de la costa en aguas de profundidad de hasta varios cientos de metros. Existen diferentes tipos de plataformas, algunas son flotantes y otras están ancladas al fondo marino sobre pilotes. Estas segundas resisten bien a las olas y el viento, así como a los hielos.

#### **1.4 Transporte de crudo**

Existen diversos medios de transporte del petróleo, pero hay dos que son los más utilizados. Nos referimos a los oleoductos (líneas de conducción) de caudal continuo que puede instalarse en el fondo del mar (*off-shore*), en distribuciones aéreas o en la superficie terrestre (*on-shore*). y a los buques petroleros. El resto de transportes, como vagones cisterna, camiones, ferrocarriles o barcos de cabotaje, suelen utilizarse más para distribuir los productos que se derivan del petróleo (Business School Ealde, 2020).

El oleoducto es fundamental para entender cómo se transporta y se distribuye el petróleo. En general, está formado por un conjunto de instalaciones formadas por tuberías para transportar el petróleo. Pero esta instalación no sólo contempla la tubería, sino también las instalaciones necesarias para su explotación. Es decir, depósitos de almacenamiento, estaciones de bombeo, equipos de limpieza, etc (Business School Ealde, 2020).

Estos oleoductos transportan petróleo crudo y en ocasiones comunican los depósitos de almacenamiento del campo de extracción directamente con los depósitos de las refinerías (Business School Ealde, 2020).

En un oleoducto, el petróleo crudo circula por una tubería gracias al impulso que proporciona una estación de bombeo. El crudo parte de los depósitos de almacenamiento, donde por medio de una red de canalizaciones y un sistema de válvulas se pone en marcha la corriente o flujo del producto.

Todo el proceso está dirigido desde un puesto central de control y hay controles situados a lo largo de toda la línea de conducción (Business School Ealde, 2020).

Las características particulares de cada oleoducto, dependen del entorno en el cual se instalen. Por ello para cada condición de instalación, ya sea en el fondo marítimo, o a nivel terrestre o subterráneo, se definen atributos y niveles de exigencias específicos, con el fin de garantizar el buen desempeño de los mismos en dicho entorno (Pérez 2018).

Por ejemplo, los tubos a instalar en el fondo del mar (*off-shore*), deben ser capaces de soportar las condiciones extremas, que por naturaleza posee este entorno, como esfuerzos por fatiga y tensión debido al oleaje y corrientes marítimas, ambientes altamente corrosivos, bajas temperaturas, alta presión externa e interna, entre otros (Pérez 2018).

En la industria petrolera, los tubos de conducción o *line pipe*, son clasificados de acuerdo a su función dentro del circuito de transporte de los fluidos como: *flowline*, líneas de transporte, *gathering line*, *trunk line* y *product pipeline* (Pérez 2018).

*Flowline:*

La función de este tipo de tendidos, es la de transportar los hidrocarburos desde los cabezales de pozos, hasta los tanques de almacenamiento.

En sistemas *off-shore*, el *riser*, conecta el *flowline* (o cabezal de pozo) a la plataforma fija o flotante. Y puede ser rígido, flexible o híbrido (Mezcla de ambos).

Líneas de transporte o Ductos:

En el caso de tubería *offshore*, las líneas de transporte conducen los hidrocarburos procesados, y no procesados, desde la plataforma hasta la costa.

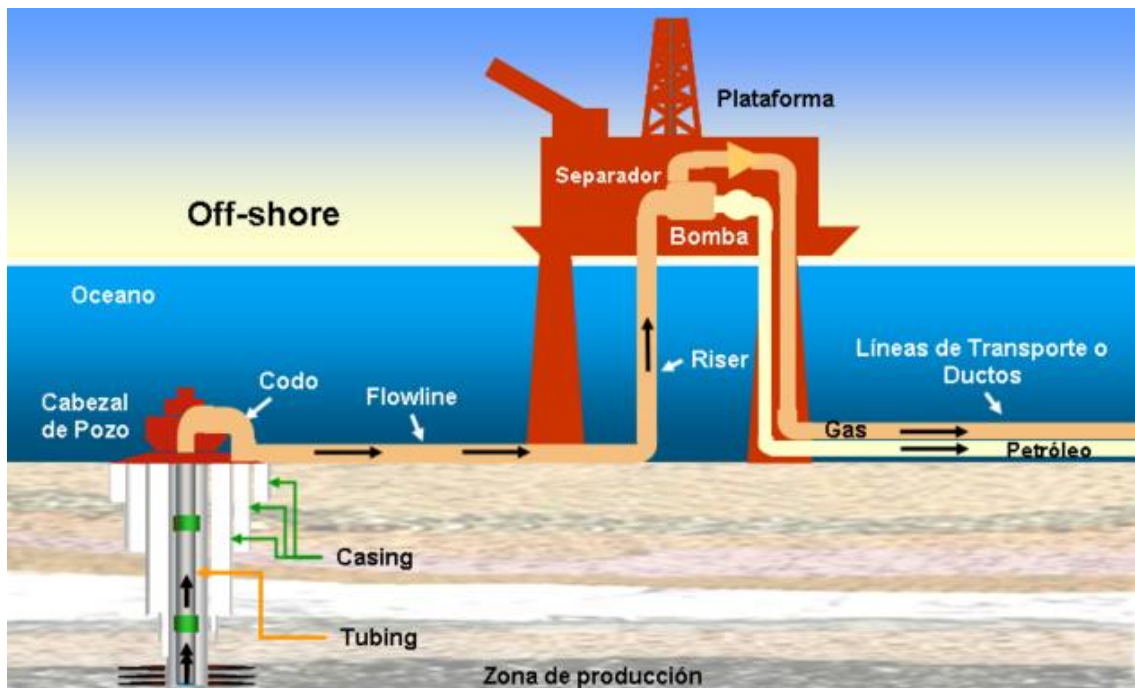


Figura 1: Transporte de crudo

*Gathering Line:*

Conducen el hidrocarburo desde los tanques o puntos de almacenamiento hasta una estación de bombeo, donde confluyen hacia líneas troncales. Usualmente son más largos que los *flowline* y suelen soportar una presión de fluido más alta.

*Trunk line:*

Conduce los fluidos desde las instalaciones de almacenamiento o sistemas de recolección hasta las plantas de procesamiento o refinerías. Usualmente

operan con una mayor presión que los *flowline* y los *gathering line*, son más largos y de mayor diámetro, y suelen estar enterrados.

Estos tendidos suelen recorrer largas distancias a través de carreteras, corrientes de agua o ambientes silvestres, por lo que en la mayoría de los casos, son necesarios permisos ambientales especiales.

#### *Product pipeline:*

Son usados para conducir petróleo o gas, desde las refinerías a los puntos de distribución final o usuarios finales.

### **1.5 Calentamiento de crudo**

Teniendo en cuenta factores como la temperatura ambiente, la radiación solar, la presión del vapor de calentamiento y las propiedades físicas de los productos petrolíferos, puede ser establecido a través de un modelo teórico para el proceso de calentamiento del serpentín del tanque de techo flotante de gran tamaño. Numéricamente se le puede dar solución a estos modelos del campo de temperatura y el campo de flujo del petróleo crudo para revelar las características de acoplamiento de la transferencia de calor y el flujo del petróleo crudo durante el proceso de calentamiento del tanque de almacenamiento. Los resultados de Sun *et al.* (2018) muestran que, de acuerdo con las variaciones del campo de flujo de petróleo crudo en el tanque, el proceso de calentamiento del serpentín de vapor del tanque se ha dividido en cuatro etapas que incluyen la formación de convección natural, la formación de vórtices pequeños, la formación de vórtices grandes y el desarrollo de vórtices grandes. Las estructuras de los serpentines tienen una gran influencia en el campo de temperatura y el campo de flujo del tanque de crudo. La estructura vertical y las bobinas de estructura estereoscópica formarán una zona de baja temperatura de cierto grosor en el fondo del tanque durante el proceso de calentamiento, pero la bobina de estructura serpentina formará una zona muerta calentada en el ángulo del vértice y la esquina de la base del tanque, donde el área es relativamente pequeña. Sobre esta base, comparando el efecto de calentamiento de los tres tipos de bobinas, se encuentra que la velocidad de calentamiento de la bobina de estructura serpentina es la más

rápida y estable. El grado de uniformidad del campo de temperatura se ve afectado por el nivel de líquido del petróleo.

### **1.5.1 Calentadores de crudo**

Los calentadores producen una elevación de la temperatura de un fluido, que puede ser un líquido, un gas o una mezcla de ambos, a expensas de la combustión de un combustible gaseoso en el o los tubos de fuego, ubicados en la parte inferior del cuerpo de presión. En general se trata de corrientes de gas y petróleo. En ese caso, el calentamiento tiene como propósito evitar la formación de hidratos o ceras, facilitar futuras separaciones de fases o disminuir la viscosidad del fluido facilitando su circulación en las líneas o su bombeo (Pirobloc, 2018).

Existen dos tipos de calentadores según Baixardoc (2018)

1. Calentadores directos: el tubo de fuego está rodeado del fluido a ser calentado, son necesarios para suministrar grandes cantidades de energía a corrientes de proceso contenidas en serpentines tubulares a partir de la combustión de combustibles líquidos y gaseosos, para elevar sus niveles de temperatura y modificar sus características.

Sus aplicaciones principales son: (Pirobloc, 2018)

Dentro del sector petroquímico y refinerías:

- Calentamiento de crudos a separadores
- Calentamiento de gaseoductos
- Sistemas de control de puntos de rocío de gases de combustible asociados a centrales de turbinas de gas
- Calefacción de flujos de gases hidrocarburos de alta presión en estaciones de reducción de presión
- Supercalentadores de vapor
- Procesos de cracking
- Hornos aromáticos
- Vaporización de líquidos y gases
- Calentadores de oxígeno

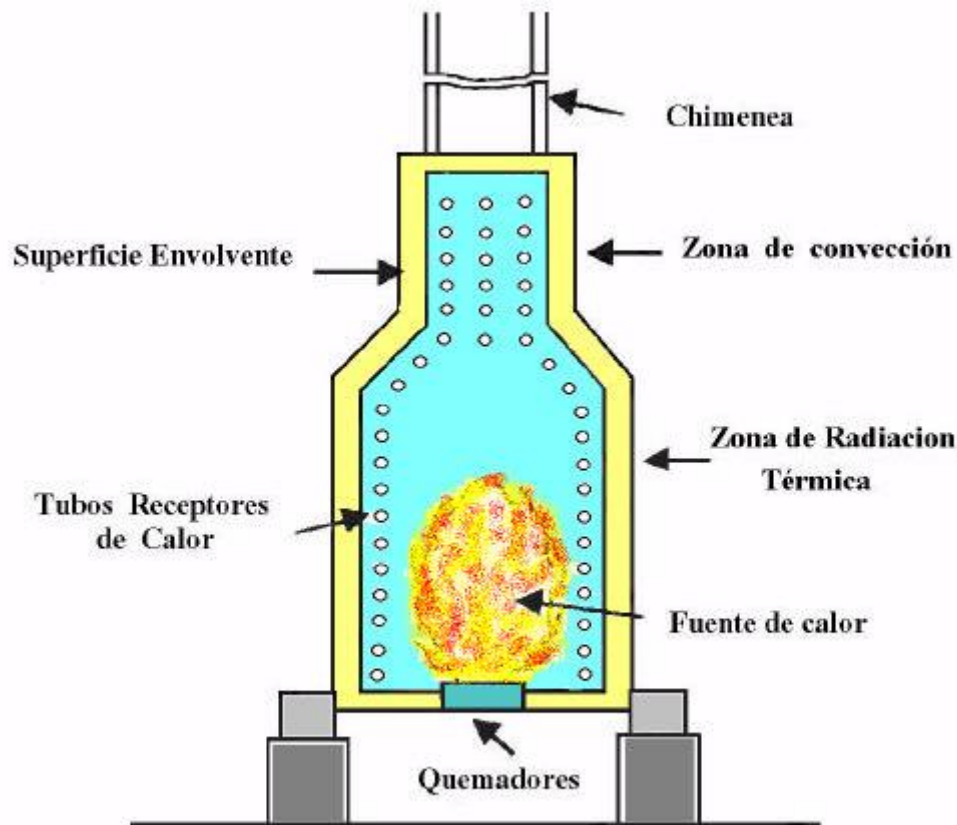


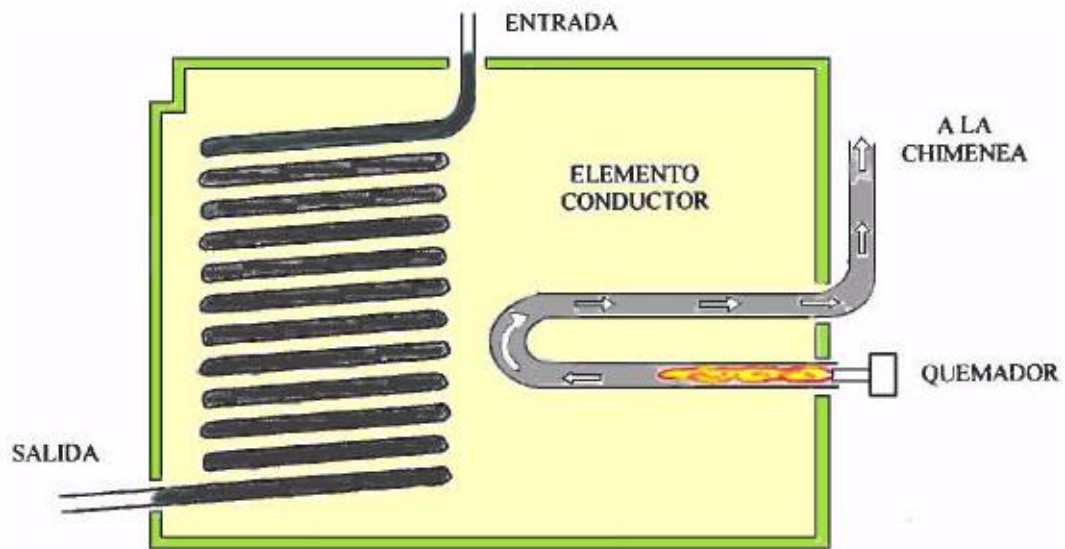
Figura 2: Ejemplo de calentador directo

2. Calentadores indirectos: el tubo de fuego está rodeado de un fluido intermediario, normalmente vapor de agua, el que a su vez cede energía al fluido que se calienta. Estos equipos pueden funcionar con agua líquida o con agua y vapor. En la parte superior del cuerpo de presión se encuentran los serpentines que conducen el fluido a ser calentado. El calentador opera en forma automática produciendo la modulación y/o el corte de gas a quemadores. Son mayormente utilizados en estaciones de almacenamiento de crudo. (Baixardoc, 2018).

Para ello opera los siguientes controles que actúan sobre el sistema de combustión: (Baixardoc, 2018)

1. Temperatura del fluido en el serpentín.
2. Presión del vapor en el cuerpo.
3. Temperatura de vapor en el cuerpo
4. Nivel de agua del cuerpo

## 5. Fallas del sistema de control



*Figura 3: Ejemplo de calentador indirecto*

Los serpentines industriales están elaborados a base de los siguientes materiales: (Inditer, 2022)

1. Acero Inoxidable.
2. Acero al carbón.
3. Cobre
4. Latón almirantazgo.
5. Cuproníquel (CuNi)

Aletas de titanio y tubos de cuproníquel, son los más utilizados en la industria petroquímica.

### **1.6 Líneas de vapor**

Las tuberías que se utilizan más usualmente son de acero al carbono para distribución de vapor y líneas de condensado.

Los sistemas de distribución de vapor conectan a las calderas con el equipo que en realidad utiliza vapor. Estos sistemas de distribución transportan el vapor hasta cualquier sitio en la planta donde se necesita energía calorífica (Kalugin *et al.*, 2021).

El vapor saturado es utilizado como la fuente de calentamiento para fluido de proceso en intercambiadores de calor, reactores, reboilers, precalentadores de aire de combustión, y otros tipos de equipos de transferencia de calor (Gu *et al.*, 2014).

### 1.6.1 Ventajas de usar vapor saturado para calentamiento

El vapor saturado tiene varias propiedades que lo hacen una gran fuente de calor, particularmente a temperaturas de 100 °C y más elevadas. Algunas de estas son (TLV, 2022):

Propiedad	Ventaja
Calentamiento equilibrado a través de la transferencia de calor latente y rapidez	Mejora la productividad y la calidad del producto
La presión puede controlar la temperatura	La temperatura puede establecerse rápida y precisamente
Elevado coeficiente de transferencia de calor	Área de transferencia de calor requerida es menor, permitiendo la reducción del costo inicial del equipo
Se origina del agua	Limpio, seguro y de bajo costo

Habiendo dicho esto, es necesario tener presente lo siguiente cuando se calienta con vapor saturado:

- La eficiencia de calentamiento se puede ver reducida si se usa un vapor diferente al vapor seco para los procesos de calentamiento. Contrario a la percepción común, virtualmente no todo el vapor generado en una caldera es vapor seco, si no vapor húmedo, el cual contiene algunas moléculas de agua no vaporizadas.



- La pérdida de calor por radiación ocasiona que una parte del vapor se condense. Por lo tanto el vapor húmedo generado se vuelve aun más húmedo, y también se forma más condensado, el cual debe ser removido al instalar trampas de vapor en las locaciones apropiadas.
- Condensado el cual es más pesado caerá del flujo de vapor y puede ser removido a través de piernas de condensado y trampas de vapor. Sin embargo, el vapor húmedo que es arrastrado reducirá la eficiencia de calentamiento, y deberá ser removido por medio de estaciones de separación en el punto de uso o en la distribución.
- El vapor que incurre en pérdidas de presión debido a exceso de fricción en la tubería, etc., podría resultar también en su correspondiente pérdida en temperatura.

El vapor húmedo es la forma más común de vapor que se pueda experimentar en plantas. Cuando el vapor se genera utilizando una caldera, generalmente contiene humedad proveniente de las partículas de agua no vaporizadas las cuales son arrastradas hacia las líneas de distribución de vapor. Incluso las mejores calderas pueden descargar vapor conteniendo de un 3% a un 5% de humedad. Al momento en el que el agua se aproxima a un estado de saturación y comienza a evaporarse, normalmente, una pequeña porción de agua generalmente en la forma de gotas, es arrastrada en el flujo de vapor y arrastrada a los puntos de distribución. Este uno de los puntos claves del porque la separación es usada para remover el condensado de la línea de distribución (TLV, 2022).

El condensado acarreado con el vapor es el causante de la mayoría de la erosión. Al golpear repetidamente la tubería en los codos, el condensado puede causar el adelgazamiento gradual de la pared de la tubería, pues su masa y alta velocidad al impactar tienen un efecto similar que las cortadoras industriales de chorro de agua. Este tipo de erosión, causada por las gotas de condensado es típicamente conocida como corrosión por picadura de gotas líquidas (TLV, 2022).

En muchos casos, pero particularmente en la tubería de acero al carbón, la erosión puede remover el tratamiento superficial de la cara interna de la tubería y acelerar el proceso de corrosión electroquímica. De hecho, ambas, la erosión

y la corrosión actúan juntas en el adelgazamiento de las paredes de las tuberías (TLV, 2022).

La resistencia a la erosión varía dependiendo del material. Por razones del costo de instalación, el acero al carbón es típicamente el material estándar para las tuberías de distribución de vapor, aunque es menos resistente que el acero inoxidable (TLV, 2022).

Algunos tratamientos superficiales ofrecen protección contra la corrosión a la tubería de acero al carbón. Los tratamientos pueden proteger temporalmente el acero y disminuyen el proceso de erosión y de corrosión (TLV, 2022).

Sin embargo, una vez que el adelgazamiento por erosión ocurre, la pared perderá su espesor cada vez más rápidamente. Esto es porque el condensado a alta velocidad no sólo causa daño físico a la tubería, sino que acelera la corrosión al remover el tratamiento superficial interno de la tubería (TLV, 2022).

## **1.7 Aislamiento térmico**

Es el conjunto de materiales y técnicas de instalación que se aplican en los elementos constructivos que separan un espacio climatizado del exterior o de otros espacios para reducir la transmisión de calor entre ellos. Asimismo se utiliza para reducir la transmisión de calor desde conducciones que transportan fluidos a distinta temperatura de la ambiente (Villasmil *et al.*, 2019).

### **1.7.1 Tipos de aislantes térmicos**

Existen diversas clasificaciones de los aislantes térmicos, por ejemplo, en cuanto al material aislante en sí o en cuanto al formato en el que se presenta. En cuanto al material como aislante térmico podemos diferenciar entre: (Solerpalau, 2018)

- Aislantes sintéticos. Son aquellos compuestos por materiales sintéticos como el plástico, los polímeros procedentes del petróleo y otros materiales sintéticos. Son muy efectivos térmicamente. Los más comunes son:

- Poliestireno expandido (EPS): Es uno de los aislantes más utilizados por su densidad y baja conductividad térmica.

El es un material plástico espumado utilizado en el sector de la construcción, principalmente como aislamiento térmico y acústico. Los productos de poliestireno expandido, al ser expuestos a temperaturas superiores a 100 °C, empiezan a reblandecerse lentamente y se contraen, si aumenta la temperatura se funden. En ausencia de un foco de ignición los productos de descomposición térmica no se inflaman hasta alcanzar temperaturas del orden de los 400 – 500 °C (Anape, 2020).

- Poliestireno extruido (XPS): Muy similar al anterior, pero con la ventaja de que se puede mojar sin problema ya que es muy absorbente. Se utiliza con frecuencia por sus múltiples aplicaciones.

Son planchas de poliestireno extruido de espuma rígida, aislante, de carácter termoplástico y de estructura de célula cerrada. Por su naturaleza y características técnicas, es la opción más avanzada y recomendable en aislamiento térmico en cubiertas, suelos, muros, bóvedas o depósitos aislados (ChovA, 2020).

- Poliuretano: Tiene un mayor rendimiento térmico que los anteriores, pero se usa generalmente proyectado como espuma.

Es mayormente utilizado como aislante para neveras y congeladores, también como aislante en la construcción, pues tiene muy buen comportamiento térmico soportando una variación amplia de temperaturas. Al tratarse de una espuma, es un material ligero, tiene poco peso, y está especialmente recomendado para el uso como aislamiento térmico en tejados (Eco Green Home, 2021).

- Rollos reflexivos: Son rollos formados por una o varias capas, de grosor variable, de burbujas de polietileno entre varias finas láminas de aluminio utilizados especialmente en zonas climáticas suaves. Son utilizados mayormente en viviendas, furgonetas, fachadas, puertas de garaje, etc.

- Lanas minerales: Son productos aislantes constituidos por un entrelazado de filamentos de materiales pétreos que forman un fieltro que mantiene entre ellos aire en estado inmóvil. Resultan muy versátiles y eficaces ya que además de proporcionar un buen nivel de aislamiento térmico también actúan como aislamiento acústico y ofrecen un elevado nivel de protección contra el fuego. Dentro de las lanas minerales se distinguen varios tipos:
- Lana de roca o lana mineral (SW). Se fabrica a partir de roca volcánica y se presenta en forma de manta, panel no rígido o rollo. Se utiliza en cubiertas, forjados, fachadas, suelos, falsos techos, buhardillas o tabiques interiores.
- Lana de vidrio (GW). Se fabrica fundiendo arena a altas temperaturas y su estructura está formada por finas fibras de vidrio unidas por un aglomerante o resina. Se considera mejor aislante acústico que la lana de roca y resiste mejor a la humedad. Al ser más liviana que otros aislantes y de muy baja conductividad térmica, consigue una mayor eficiencia con el mismo espesor.
- Silicato de calcio. Aislamiento del silicato del calcio es normalmente aplicado en alta temperatura en tuberías y equipos en instalaciones industriales, como fábricas de productos químicos, refinerías, centrales eléctricas de vapor, ya que es un material rígido que tiene una conductividad térmica relativamente baja, extremadamente de alta resistencia a la compresión, alta resistencia a la flexión. Todo eso debido a su alta resistencia a la compresión (mayor que 100 PSI), alta resistencia a la flexión (mayor que 50 PSI), y resistencia al daño de la agitación, además puede soportar vibración inducida por flujo de vapor de alta temperatura alrededor de obstrucciones del interior de tuberías, como el funcionamiento interno de la válvula, aparatos de medición, y orificios de restricción de flujo General Insulation Company (2020). Las propiedades de inhibición de corrosión del silicato de calcio no se disminuyen con cambios de temperatura. Este a su vez no se quema y puede usarse para protección contra fuego en algunas aplicaciones Calorcol Expertos en Ingeniería de Aislamientos (2022). Por todo esto es

que propongo este aislante como el más adecuado para el caso de estudio.

Los diversos materiales aislantes pueden presentarse en diversos formatos que determinarán la forma en la que deben ser aplicados. Como resumen podemos decir que podemos encontrar:

- Placas o paneles. Pueden ser rígidos o semirrígidos y necesitan fijación a un soporte.
- Rollos, mantas y paneles no rígidos. Habitualmente se presentan en estos formatos los aislantes ecológicos o naturales y los de origen mineral. Se suelen colocar sobre perfilería, en la cámara entre el muro y placas de cartón-yeso, y también entre rastreles de suelos de madera o en falsos techos.
- Espuma. Mediante insuflado y con aparatos especiales se inyectan en las cámaras de aire.
- Partículas, bolitas, fibras. Al igual que en el caso anterior se suelen insuflar en las cámaras de aire.

### **Conclusiones parciales**

- En estaciones de almacenamiento y de tratamiento se utilizan los calentadores indirectos.
- Los aislantes que se utilizan en combustibles no pueden ser inflamables.
- Para el aislamiento de tuberías se debe utilizar láminas flexibles o laminas rígidas que se adapte a la forma de la tubería.

## CAPÍTULO 2: METODOLOGÍA DE LA INVESTIGACIÓN

---

En este capítulo se explica la metodología de la investigación que se utiliza para determinar el espesor óptimo de aislante en las líneas de vapor del área de Crudo y Suministro en la División Territorial de Comercialización de Combustibles, Matanzas

La División Territorial de Comercialización de Combustibles, Matanzas (DTCCM) cuenta con cinco aéreas operacionales. En el área de ``Crudo y Suministro`` se lleva a cabo la recepción del crudo proveniente de las productoras (Centro y Occidente). A partir de su almacenamiento se lleva a cabo la comercialización del combustible con los diferentes clientes. El crudo nacional se suministra mediante oleoductos a las centrales térmicas Ernesto Guevara y Antonio Guiteras, debido a su alta viscosidad y las distancias en las que se encuentran las termoeléctricas, es necesario antes del bombeo, someterlo a un calentamiento con el objetivo de lograr temperatura de salida de 80 °C. Además se realiza la entrega a buques con diferentes fines. Esta operación está pactada para que se lleve a cabo con una temperatura de salida del crudo de 65 °C aproximadamente. Los tanques de almacenamiento, equipados con sistemas de calentamiento (serpentes), así como los intercambiadores de calor, emplean vapor como medio de calentamiento.

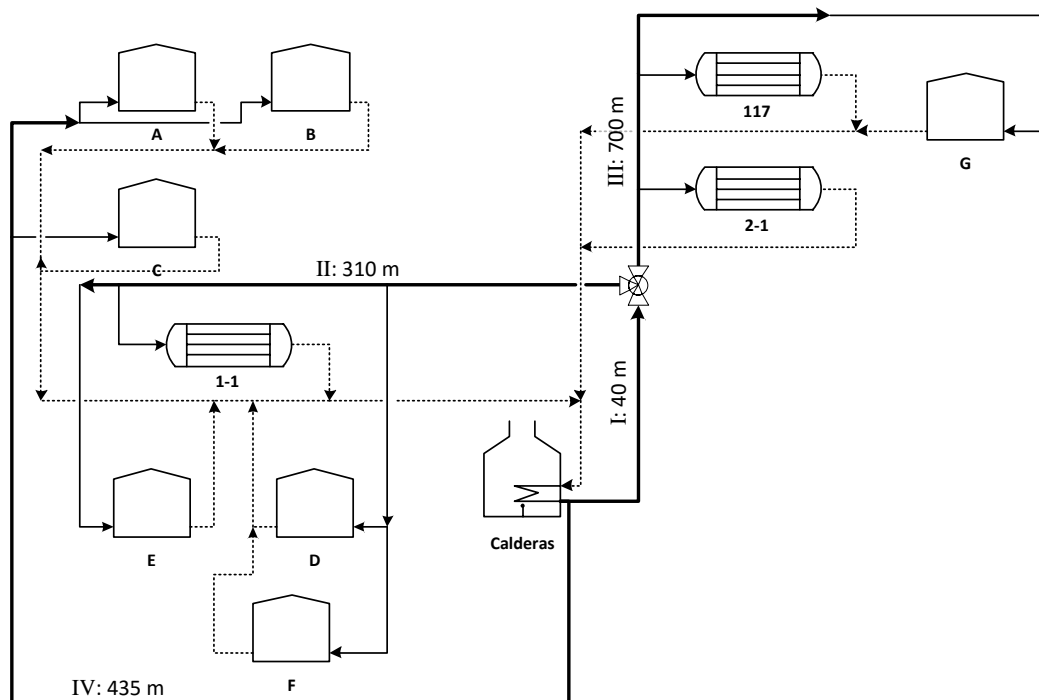
### **2.1 Descripción del caso de estudio**

La División Territorial de Comercialización de Combustibles, Matanzas cuenta con una casa de calderas, con dos calderas para la producción de vapor. La capacidad máxima de generación de cada una es de 16 t/h y el combustible empleado es *fuel oil*. En el momento de realizar la investigación una de las calderas se encuentra en mantenimiento general. Se utilizara para cálculos posteriores, una producción de vapor promedio de 12 t/h, que es la producción de una de las calderas en un día normal de trabajo.

Actualmente funciona una caldera con bajos rendimientos. El vapor producido se distribuye por dos líneas principales hacia diferentes áreas de la empresa, y

en el interior de cada área se distribuyen hacia los equipos que consumen vapor a través de líneas secundarias de menor diámetro.

De la sala de calderas salen dos líneas principales: línea I y línea IV. La primera va hacia una bifurcación. Se derivan las líneas principales II y III. En línea III se instalaron líneas secundarias (línea S1 y S2) para llevar el vapor hacia el intercambiador 117 y la estación 2-1. Para el tanque G se instala una línea secundaria (línea S3). En línea II se instalaron líneas secundarias que van hacia la estación 1-1 (línea S4), y hacia los tanques E (línea S5), D (línea S6) y F (línea S7). Estos dos últimos reciben el vapor mediante una bifurcación en una línea secundaria que proviene de la línea principal II. En cuanto a la línea principal IV va hacia los tanques A, B y C que están conectados a ella por líneas secundarias (línea S8, S9 y S10 respectivamente)



- > Línea principal de vapor (D = 8 in)
- > Línea secundaria de vapor (D = 4 in)
- - -> Retorno de condensado

*Figura 4: Diagrama del proceso*

## 2.2 Caracterización de las líneas de vapor en el área

*Tabla 1: Líneas principales*

Líneas principales	Diámetro	Longitud	Material	Tipo de sección
Línea I	8 in	40 m	Acero	Circular
Línea II	8 in	310 m	Acero	Circular
Línea III	8 in	700 m	Acero	Circular
Línea IV	8 in	435 m	Acero	Circular
Total		1485 m		

*Tabla 2: Líneas secundarias*

Líneas secundarias	Diámetro		Material	Tipo de sección
Línea S1	4 in	8,3 m	Acero	Circular
Línea S2	4 in	66,5 m	Acero	Circular
Línea S3	4 in	46,7 m	Acero	Circular
Línea S4	4 in	33,2 m	Acero	Circular
Línea S5	4 in	32,7 m	Acero	Circular
Línea S6	4 in	60,6 m	Acero	Circular
Línea S7	4 in	70,1 m	Acero	Circular
Línea S8	4 in	7,0 m	Acero	Circular
Línea S9	4 in	16,4 m	Acero	Circular
Línea S10	4 in	44,4 m	Acero	Circular
Total		385,9 m		

## 2.3 Pérdidas de calor en tuberías no aisladas

1. Calcular las pérdidas de calor totales por unidad de longitud por el mecanismo de convección-radiación.



$$q' = q'_{conv} + q'_{rad} \quad (1)$$

2. Calcular las pérdidas de calor por unidad de longitud a causa de la radiación térmica.

$$q'_{rad} = \varepsilon \pi D \sigma (T_i^4 - T_\infty^4) \quad (2)$$

3. Calcular las pérdidas de calor por unidad de longitud a causa de la convección libre.

$$q'_{conv} = \bar{h} \pi D (T_i - T_\infty) \quad (3)$$

4. Calcular el coeficiente de transferencia de calor a partir de la ecuación de *Nusselt*.

$$Nu_D = \frac{\bar{h} D}{k}$$

$$\bar{h} = \frac{k Nu_D}{D} \quad (4)$$

5. Calcular el *Número de Nusselt* mediante la correlación de Churchill y Chu.

$$Nu_D = \left\{ 0,60 + \frac{0,387 Ra_D^{\frac{1}{6}}}{\left[ 1 + \left( \frac{0,559}{Pr} \right)^{\frac{9}{16}} \right]^{\frac{8}{27}}} \right\}^2 \quad Ra_D \leq 10^{12} \quad (5)$$

6. Calcular el *Número de Rayleigh* y, de ser necesario, el *Número de Prandtl*

$$Ra_D = \frac{g\beta(T_s - T_\infty)D^3}{\nu\alpha} \quad (6)$$

$$Pr = \frac{\nu}{\alpha} = \frac{Cp\mu}{k} \quad (7)$$

7. Calcular la temperatura media entre la superficie del aislante y el medio ambiente y determinar todas las propiedades del aire a esta temperatura.

$$T_f = \frac{T_i + T_\infty}{2} \quad (8)$$

8. Asumir la temperatura en la superficie del aislante. Una buena aproximación inicial es el valor medio entre la temperatura del vapor y la temperatura ambiente.

$$T_i = \frac{T_s + T_\infty}{2} \quad (9)$$

## 2.4 Pérdidas de calor en tuberías aisladas

1. Las pérdidas de calor totales son las mismas por mecanismo de convección-radiación al medio (epígrafe 2.3) y por el mecanismo de conducción a través del aislante, calcular la temperatura en la superficie del aislante es:

$$q' = q'_{cond} = \frac{2\pi k(T_s - T_i)}{\ln(r_i / r_s)}$$

$$T_i = T_s - \frac{q' \ln(r_i / r_s)}{2\pi k} \quad (10)$$

2. Si la temperatura en la superficie del aislante asumida en el paso 8 (epígrafe 2.3) difiere significativamente de la calculada en el paso 1 de este epígrafe ( $|T_{i,8} - T_{i,1}| > 0,1^\circ C$ ), utilizar la calculada por la ecuación 10 como nuevo valor asumido y repetir desde el paso 7 (epígrafe 2.3).

## 2.5 Implementación en MatLab del modelo matemático

Se hace uso de la función *xsteam* para la estimación de las propiedades físicas del agua, esta función se encuentra publicada por Holmgren (2006).

En cuanto a las propiedades del aire se diseñó la función equivalente *xair* basada en la tabla de propiedades de Incropera (2006).

Las propiedades físicas del aislante se estimaron por Industrial Insulation Group (2016) mediante la función *insul\_thermo*.

Para calcular el calor por convección se diseñó la función *qconv* y fue necesario utilizar las propiedades del aire (*xair*) para su resolución.

El cálculo del calor por radiación se diseñó la función *grad*.

El cálculo del calor por conducción se diseñó la función *qcond*.

Los códigos de todas estas funciones están descritos en el anexo (5).

## 2.6 Valoración económica sobre el uso del aislante

El costo total del análisis se desglosa en costo por pérdida de calor y costo por la inversión del aislamiento.

El costo por calor perdido se origina por el costo del volumen equivalente de combustible consumido en caldera. La caldera opera con *fuel oil* mediano (*MFO*), con un valor calórico de combustión (Global Combustion, 2022):

$$Q_c = 39,17 \frac{\text{MJ}}{\text{L}} \quad (11)$$

$$V_{comb} \approx \frac{Q_{perd}}{Q_c \cdot efic} \quad (12)$$

El precio unitario del con *fuel oil* mediano en noviembre de 2022 fue:

$$pup_c = 13,66306 \frac{\$}{\text{L}} \quad (13)$$

El costo por instalación del aislante se origina por el precio unitario de este (anexo 3).

Y la combinación de ambos costos origina el costo total:

$$C_T = C_{Qperd} + C_{Aisl} \quad (14)$$

### Conclusiones parciales

- La temperatura en la superficie del tubo sin aislar es aproximadamente igual a la temperatura del vapor que fluye en el tubo.
- El costo total del análisis efectuado esta dado por el costo que conlleva el calor perdido y el costo de instalación del aislante.

## CAPÍTULO 3: ANÁLISIS DE LOS RESULTADOS

---

En el presente capítulo se exponen los resultados de la investigación al aplicar las metodologías para el desarrollo del modelo matemático y la solución del mismo en el *software* MatLab.

### 3.1 Calor perdido en tubería desnuda

Debido a la alta conductividad térmica del acero con que están fabricadas las tuberías de las líneas de vapor y el relativo poco espesor de las mismas se puede despreciar el mecanismo conductivo a través de la pared de la tubería. Solo se consideran los mecanismos combinados de convección y radiación. La temperatura en la superficie de la tubería es aproximadamente igual a la del vapor saturado en el interior.

Al solucionar el modelo bajo estas condiciones se obtienen los siguientes resultados:

Pérdida de calor total:	3171,32 kW
Pérdida de calor en línea principal:	2783,75 kW
Pérdida de calor en línea secundaria:	387,57 kW
Pérdida de vapor equivalente:	5667,47 kg/h

La pérdida de vapor equivalente se refiere a la energía perdida expresada como cantidad de vapor condensado a causa del enfriamiento de la tubería. Como se puede observar aproximadamente la mitad del vapor generado por la caldera condensa sin cumplir su objetivo.

### 3.2 Calor perdido en tubería aislada

El aislante propuesto es *Thermo-12<sup>®</sup> Gold* (silicato de calcio de Industrial InsulationGroup LLC (2010)). Las propiedades físicas de este aislante se muestran en el anexo 2.

La resistencia a la pérdida de calor a través del aislante es significativa por lo que deben ser considerados de manera combinada los mecanismos de convección y radiación desde la superficie y de conducción a través del

aislante. La temperatura en la superficie del aislante debe ser calculada como parte del modelo.

La fuerte dependencia del espesor de aislante sobre las pérdidas de calor, la temperatura superficial y el costo de la inversión hace recomendable que los resultados sean presentados como gráficos en función este parámetro.

### 3.2.1 Pérdidas de calor en función del espesor de aislante

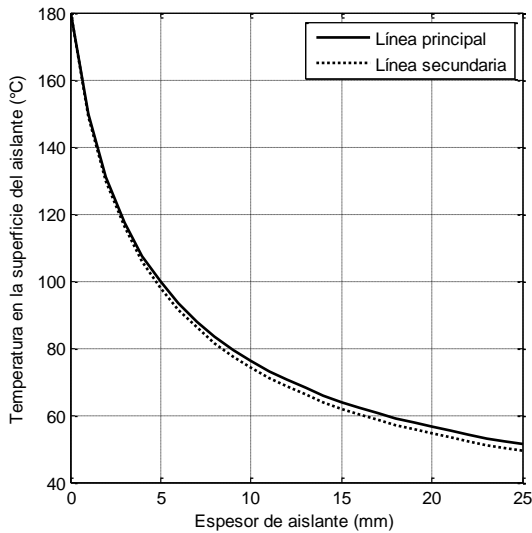


Figura 5: Temperatura en la superficie del aislante (°C) vs Espesor de aislante (mm)

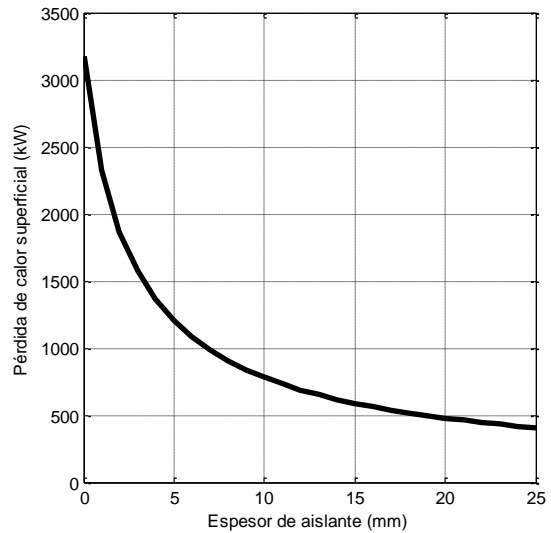
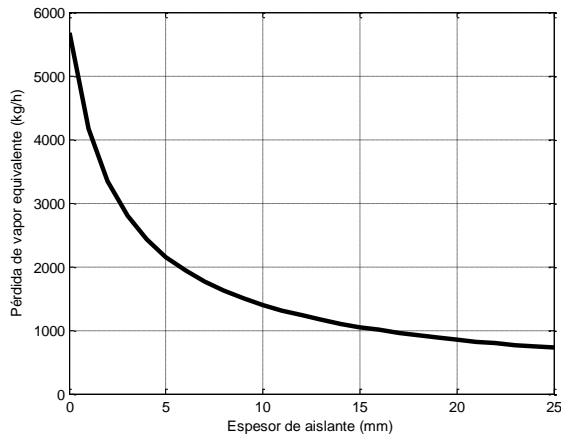


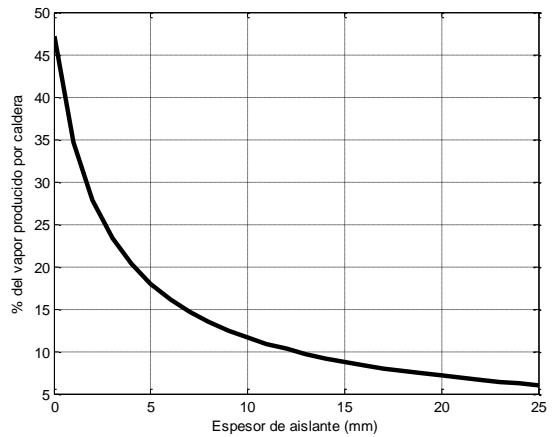
Figura 6: Pérdida de calor superficial (KW) vs Espesor de aislante (mm)

El calor perdido se puede expresar como pérdida del vapor generado por la caldera:

$$m_{vapor} = \frac{Q_{perd}}{\lambda}$$



*Figura 7: Pérdida de vapor equivalente (kg/h) vs Espesor de aislante (mm)*

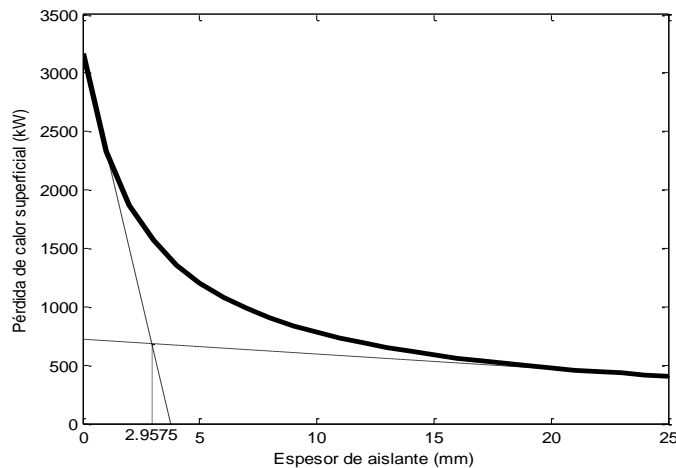


*Figura 8: % del vapor producido por caldera vs Espesor de aislante (mm)*

En las figuras anteriores se aprecia un cambio brusco de pendiente en todas las curvas. Esto significa que para pequeños cambios de espesor de aislante sobre la tubería desnuda, se logran grandes ahorros de energía. Pero para grandes cambios de espesores sobre la tubería ya aislada, la energía ahorrada es insignificante. La zona de cambio de pendiente corresponde con el *criterio termodinámico de optimización* del espesor de aislante.

### **3.2.2 Selección del espesor de aislante en función de la termodinámica**

Para seleccionar el espesor óptimo de aislante desde el punto de vista termodinámico se extienden las rectas tangentes a la curva en la zona de mayor pendiente y en la de menor. El punto de intersección de ambas rectas corresponde con el criterio de optimización.



*Figura 9: Pérdida de calor superficial (KW ) vs Espesor de aislante (mm )*

De la optimización grafica se observa que el espesor ideal de aislante es aproximadamente 3 mm. Para este espesor de aislante se calculan las pérdidas:

Pérdida de calor total:	1570,43 kW
Pérdida de calor en línea principal:	1377,50 kW
Pérdida de calor en línea secundaria:	192,93 kW
Pérdida de vapor equivalente:	2806,53 kg/h

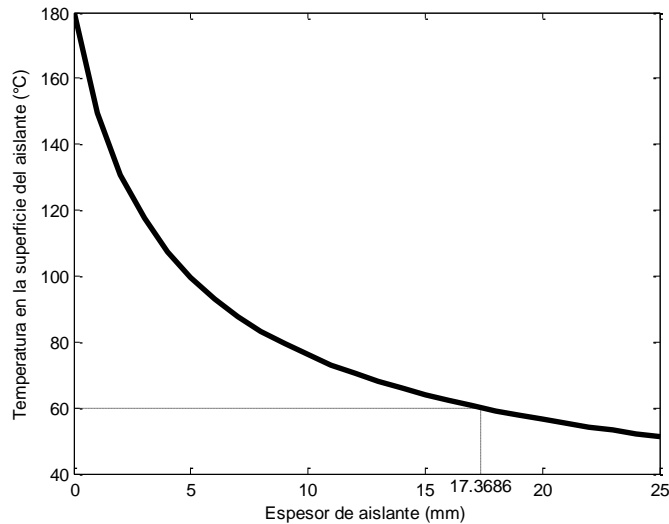
Como se observa la perdida de vapor equivalente se reduce a la mitad de la tubería desnuda.

### **3.3 Selección del espesor de aislante en función de la temperatura superficial**

Por razones de seguridad se recomienda que la temperatura en la superficie del aislante no supere los 60 °C.

La relación entre la temperatura superficial en función del espesor de aislante se muestra en la siguiente figura:





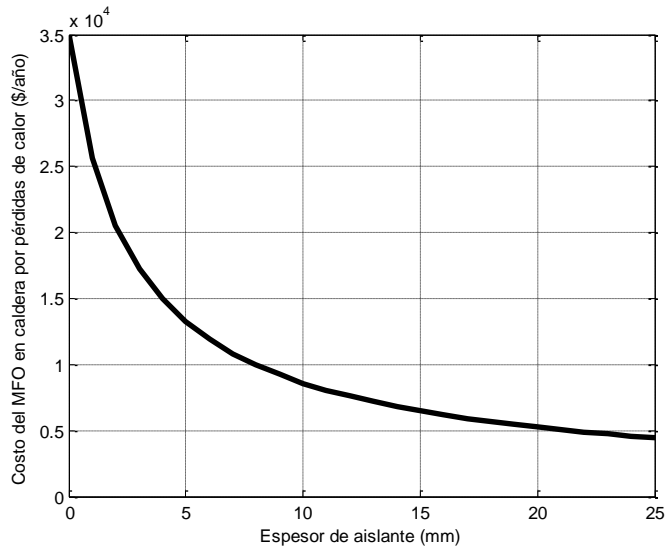
*Figura 10: Temperatura en la superficie del aislante (°C) vs Espesor de aislante (mm)*

Como se observa para garantizar los 60 °C en la superficie se necesita un espesor de aislante de 18 mm o superior.

Pérdida de calor total: 515,65 kW  
 Pérdida de calor en línea principal: 449,76 kW  
 Pérdida de calor en línea secundaria: 65,89 kW  
 Pérdida de vapor equivalente: 921,52 kg/h

### **3.4 Selección del espesor de aislante en función del costo total**

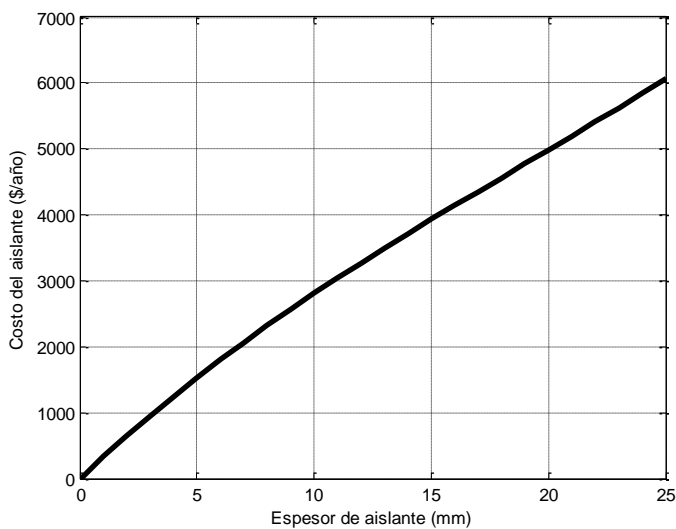
El costo debido a las pérdidas de calor presenta un comportamiento similar a las pérdidas de calor en sí mismas como se muestra en la figura:



*Figura 11: Costo del MFO en caldera por pérdidas de calor (\$/ año) vs Espesor de aislante (mm)*

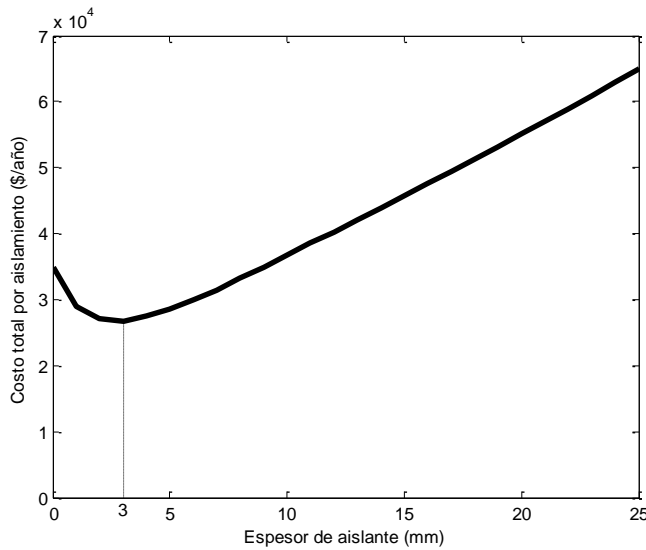
Es decir, en el grafico se nota un cambio brusco de pendiente entre el costo por calor perdido y el espesor de aislante.

El costo por instalación del aislante presenta un comportamiento contrario al grafico anterior, es decir, un aumento en el espesor de aislante origina un aumento en el costo de este.



*Figura 12: Costo del aislante (\$/ año) vs Espesor de aislante (mm)*

El costo total de instalación de aislante y pérdidas de calor presenta un mínimo para 3 mm de espesor.



*Figura 13: Costo total por aislamiento (\$/ año) vs Espesor de aislante (mm)*

Espesor óptimo por costo total de aislamiento: 3 mm.

Como se observa tanto el criterio termodinámico como el económico coinciden con un espesor de aislante de 3 mm.

A partir de los tres criterios de selección del espesor de aislante se puede concluir que, los basados en el diseño óptimo (a partir del criterio termodinámico y del económico) sugieren un espesor de aislante de 3 mm. El criterio basado en la temperatura superficial corresponde a una restricción del proceso de optimización y sugiere un espesor de aislante de 18 mm.

La siguiente tabla resume los resultados para la selección final del espesor de aislante tanto para la condición actual (tubería desnuda) como para los espesores recomendados por los criterios de selección:

*Tabla 3: Características del espesor en cuanto a cantidad de espesor, calor perdido, temperatura superficial y costo total por su implementación.*

Espesor de aislante	0 mm	3 mm	18 mm
Calor perdido	3171,32 kW	1570,43 kW	515,65 kW

Temperatura superficial	180 °C	115 °C	60 °C
Costo total	35 000 \$/año	28 000 \$/año	51 000 \$/año

Según las normas de seguridad industrial ASTM (2020), se recomienda para aislamiento térmico en la industria, cumplir con una temperatura superficial en las paredes del aislante de hasta 60 °C. Sin embargo, para garantizar esta temperatura los costos de inversión e instalación del aislante de 18 mm de espesor hacen que los costos totales asciendan a 51 000 \$/año. Esto significa que después de aislada, los costos totales se incrementan en 16 000 \$/año más que si se mantiene la tubería desnuda.

Por otra parte, si se aísla con un espesor de 3 mm, la temperatura superficial será de 115 °C, lo cual no cumple con lo establecido (60 °C). Aunque este espesor de aislante ofrece un ahorro de 7 000 \$/año con respecto a los costos totales asociados a la tubería desnuda, y de 23 000 \$/año con respecto al espesor de aislante de 18 mm.

Aunque la decisión final para la selección del espesor de aislante es de carácter administrativo y no técnico, el autor recomienda la solución más económica. Teniendo en cuenta que actualmente la temperatura superficial de estas líneas es 180 °C, un espesor de aislante de 3 mm, aunque mantiene el riesgo por quemaduras, reduce significativamente el costo por calor perdido.

### **Conclusiones parciales**

- Las pérdidas de calor actuales corresponden al 50 % del vapor generado.
- Las primeras capas de aislante producen un ahorro de energía significativo.
- Tanto la optimización termodinámica como la económica coinciden en que un espesor de 3 mm es el más conveniente.

## CONCLUSIONES

---

1. El modelo que combina las pérdidas de calor por conducción a través del aislante con las pérdidas de calor por convección-radiación permitió establecer un espesor de aislante de 3 mm como criterio económico y termodinámicamente óptimo.
2. Mediante el análisis bibliográfico se determinó que el tipo de aislante más adecuado para el caso de estudio es el silicato de calcio.
3. El caso de estudio cuenta con dos tipos de líneas de vapor bien definidas, las principales de 8 in y las secundarias de 4 in.
4. La implementación del modelo en MatLab permite valorar diferentes alternativas de espesor de aislante, así como diferentes criterios de optimización.

## RECOMENDACIONES

---

Aplicar la metodología a las líneas de vapor en la División Territorial de Comercialización de Combustibles, Matanzas para validar los resultados obtenidos.

## REFERENCIAS BIBLIOGRÁFICAS

---

1. Anape. (2020). *Poliestireno expandido*. <http://www.anape.es/index.php?accion=producto#:~:text=El%20Poliestireno%20Expandido%20o%20EPS,una%20serie%20de%20aplicaciones%20diversas.>
2. ASTM. (2020). *Standard guide for heated system surface conditions that produce contact burn injuries*. <https://www.astm.org/c1055-20.html>
3. Baixardoc. (2018). *Calentadores de crudo de Petróleo*. <https://baixardoc.com/documents/calentadores-de-crudo-de-petroleo-5c74506212971>
4. Business School Ealde. (2020). *Cómo se transporta el petróleo en un proyecto de Oil & Gas*. <https://www.ealde.es/tipos-transporte-petroleo/>
5. Calorcol Expertos en Ingeniería de Aislamientos. (2022). *Silicato de Calcio*. <https://calorcol.com/productos/silicato-de-calcio/>
6. ChovA. (2020). *POLIESTIRENO EXTRUIDO – ChovAFOAM XPS*. <https://chova.com/producto/xps-poliestireno-extruido/>
7. Eco Green Home. (2021). *Cómo aplicar correctamente espuma de poliuretano*. <https://ecogreenhome.es/como-aplicar-correctamente-espuma-de-poliuretano/>
8. General Insulation Company. (2020). *Accesorios de silicato de calcio*. <https://www.generalinsulation.com/products/marine/marine-insulation/productsmarinemarine-insulationfittings-7/?lang=es#:~:text=Aislamiento%20del%20silicato%20del%20calcio,%20C%20tubo%20preformado%20y%20Junta.>
9. Global Combustion. (2022). *Oil fuel properties*. <http://www.globalcombustion.com/oil-fuel-properties/>
10. Global estaciones de servicio. (2022). *Usos del Petróleo [Actualizado 2022]*. <https://globalestacionesdeservicio.com/usos-del-petroleo/>
11. Gu, H., Cheng, L., Huang, S., Du, B., y Hu, C. (2014). Prediction of thermophysical properties of saturated steam and wellbore heat losses in concentric dual-tubing steam injection wells. *Energy*, 75, 419-429.
12. Holmgren, M. (2006). *Water and steam properties* <http://www.x-eng.com/>

13. Incropera. (1999). *Fundamentos de transferencia de calor* (Vol. 4ta ).
14. Inditer. (2022). *Tipos y usos de serpentines en las industrias*.  
<https://inditer.es/blog/serpentines/#:~:text=Los%20serpentines%20industriales%20est%C3%A1n%20elaborados,Cobre>
15. Industrial Insulation Group. (2016). *Thermo-12® Gold* <http://www.iig-llc.com/>
16. Industrial Insulation Group LLC. (2010). *Thermo-12®Gold : pipe & block insulation*  
[http://www.industrialinsulation.com/calcium\\_silicate\\_insulation.htm](http://www.industrialinsulation.com/calcium_silicate_insulation.htm)
17. Ingenieriaquimica.net. (2017). *Principales métodos de extracción del petróleo*. <https://www.ingenieriaquimica.net/articulos/395-principales-metodos-de-extraccion-del-petroleo>
18. Kalugin, P. N., Anokhov, A. E., Fedina, I. V., Kreitser, K. K., Dudarev, I. D., y Dudka, G. A. (2021). Study of Pipe Bend Metal Prepared by Electroslag Melting of Steel 15KH1M1F-SH and Intended for Thermal Power Plant Steam Lines. *Power Technology and Engineering*, 54(6), 889-895.
19. Keesing, J. K., Gartner, A., Westera, M., Edgar, G. J., Myers, J., Hardman-Mountford, N. J., y Bailey, M. (2018). Impacts and environmental risks of oil spills on marine invertebrates, algae and seagrass: A global review from an Australian perspective.
20. Ministerio para la Transición Ecológica. (2020). *El petróleo y su comportamiento*. [https://www.miteco.gob.es/es/costas/temas/proteccion-medio-marino/plan-ribera/contaminacion-marina-accidental/petroleo\\_y\\_comportamiento.aspx](https://www.miteco.gob.es/es/costas/temas/proteccion-medio-marino/plan-ribera/contaminacion-marina-accidental/petroleo_y_comportamiento.aspx)
21. Pérez , Y. (2018). *Tubería de conducción para uso petrolero*. <https://whaleshares.io/@yreudy/tuberia-de-conduccion-para-uso-petrolero>
22. Philp, R. P. (2020). Composition and Properties of Petroleum. *J Hydrocarbons, Oils Lipids: Diversity, Origin, Chemistry Fate*, 269-310.
23. Pirobloc. (2018). *Calentadores y calderas a fuego directo para calentamiento industrial*. <https://www.pirobloc.com/blog-es/calentadores-y-calderas-a-fuego-directo/>



24. Solerpalau. (2018). *Materiales aislantes térmicos: tipos y aplicaciones*.  
<https://www.solerpalau.com/es-es/blog/materiales-aislantes-termicos/>
25. Steels. (2002). *Insulationprice guide master: calciumsilicate T-12 pipe insulation*. . Insulationprice guide master: calciumsilicate T-12 pipe insulation.
26. Sun, W., Cheng, Q., Zheng, A., Gan, Y., Gao, W., Liu, Y. J. I. J. o. H., y Transfer, M. (2018). Heat flow coupling characteristics analysis and heating effect evaluation study of crude oil in the storage tank different structure coil heating processes. *127*, 89-101.
27. TLV. (2022). *Erosión en tuberías de vapor y condensado*.  
<https://www.tlv.com/global/LA/steam-theory/piping-erosion.html>
28. Villasmil, W., Fischer, L. J., y Worlitschek, J. (2019). A review and evaluation of thermal insulation materials and methods for thermal energy storage systems. *Renewable and Sustainable Energy Reviews*, *103*, 71-84.
29. Wilson, M., Bailey, D., Maung-Douglass, E., Partyka, M., Sempier, S., y Skelton, T. (2021). Oil spill modelling: From the ocean to humans: Integrated modelling of oil spill impacts.
30. Wu, X. F., y Chen, G. Q. (2019). Global overview of crude oil use: From source to sink through inter-regional trade. *Energy Policy*, *128*, 476-486.

## 1. Símbolos

$C_p$  Calor específico a presión constante,  $J/(kgK)$

$g$  Aceleración gravitacional,  $m/s^2$

$h$  Coeficiente de transferencia de calor por convección,  $W/(m^2K)$

$k$  Conductividad térmica,  $W/(mk)$

$r_i$  Radio exterior del aislante,  $m$

$r_s$  Radio interior del tubo,  $m$

$T_f$  Temperatura media entre la superficie del aislante y el medio,  $K$

$T_i$  Temperatura de la superficie del aislante,  $K$

$T_s$  Temperatura de la superficie del tubo  $\approx$  temperatura del vapor,  $K$

$T_\infty$  Temperatura del medio,  $K$

$\alpha$  Difusividad térmica,  $m^2/s$

$\beta$  Coeficiente de expansión térmica volumétrica,  $K^{-1}$

$\varepsilon$  Emisividad

$\mu$  Viscosidad dinámica,  $kg/(sm)$

$\nu$  Viscosidad cinemática,  $m^2/s$

$\rho$  Densidad de masa,  $kg/m^3$

$\sigma$  Constante de Stefan-Bolizmann

$\lambda$  Calor latente

## 2. Determinación de las propiedades físico-químicas del aire

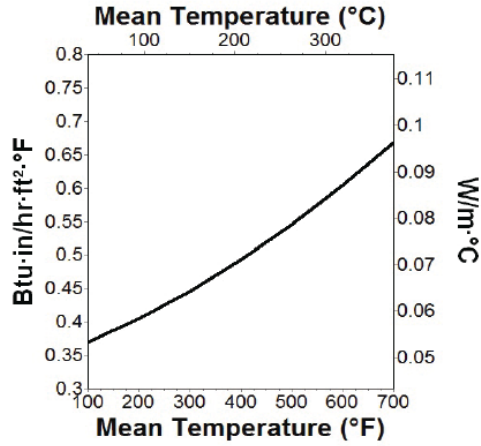
**TABLA A.4** Propiedades termofísicas de gases a presión atmosférica<sup>a</sup>

$T$ (K)	$\rho$ (kg/m <sup>3</sup> )	$c_p$ (kJ/kg · K)	$\mu \cdot 10^7$ (N · s/m <sup>2</sup> )	$\nu \cdot 10^6$ (m <sup>2</sup> /s)	$k \cdot 10^3$ (W/m · K)	$\alpha \cdot 10^6$ (m <sup>2</sup> /s)	$Pr$
Aire							
100	3.5562	1.032	71.1	2.00	9.34	2.54	0.786
150	2.3364	1.012	103.4	4.426	13.8	5.84	0.758
200	1.7458	1.007	132.5	7.590	18.1	10.3	0.737
250	1.3947	1.006	159.6	11.44	22.3	15.9	0.720
300	1.1614	1.007	184.6	15.89	26.3	22.5	0.707
350	0.9950	1.009	208.2	20.92	30.0	29.9	0.700
400	0.8711	1.014	230.1	26.41	33.8	38.3	0.690
450	0.7740	1.021	250.7	32.39	37.3	47.2	0.686
500	0.6964	1.030	270.1	38.79	40.7	56.7	0.684
550	0.6329	1.040	288.4	45.57	43.9	66.7	0.683
600	0.5804	1.051	305.8	52.69	46.9	76.9	0.685
650	0.5356	1.063	322.5	60.21	49.7	87.3	0.690
700	0.4975	1.075	338.8	68.10	52.4	98.0	0.695
750	0.4643	1.087	354.6	76.37	54.9	109	0.702

Fuente: (Incropera, 1999)

### 3. Propiedades del aislante *Thermo-12<sup>®</sup> Goldde Industrial InsulationGroup LLC.*

#### THERMAL CONDUCTIVITY \*



Mean Temperature	°F	100	200	300	400	500	600	700
	°C	38	93	149	204	260	316	371
Btu·in/(hr·ft²·°F)		.37	.41	.45	.49	.55	.60	.67
W/m·°C		.053	.058	.064	.071	.079	.087	.096

\* Thermo-12 Gold Pipe & Block Insulation as tested in accordance with ASTM C177, ASTM C518 and ASTM C335.

SPECIFICATION COMPLIANCE	
ASTM C165 Compressive Strength	>100psi(690kPa) 5% compression
ASTM C203 Flexural Strength	>50psi(450kPa)
ASTM C302 Density (Dry) Average	>14pcf(230kg/m <sup>3</sup> )
ASTM C356 Linear Shrinkage	<2.0% after 24hr Soaking period at 1200°F(650°C)
ASTM C421 Abrasion Resistance Weight Loss by Tumbling	After the first 10min <20% After the second 10min <40%
ASTM C447 Maximum Service Temperature	1200°F(650°C)
ASTM C533, Type I Material Specification	Passes
ASTM C665 Corrosivity to Steel	Passes-Inhibits
ASTM C795/C871/C692 Corrosion	Passes-Inhibits
ASTM C1338 Fungi Resistant	Passes
ASTM C1617 Corrosion	Passes-Inhibits
ASTM E84 Surface Burning Characteristics	Flame Spread -0 Smoke Developed -0
ASTM E119 Building Fire Test	Passes
ASTM E136 Non Combustibility	Passes
BS 476 PART II	Passes
Can/ULC S-102 Surface Burning Characteristics	Flame Spread -0 Smoke Developed -0

Fuente: Industrial InsulationGroup LLC (2010)

#### 4. Precio unitario del aislante *Thermo-12<sup>®</sup> Gold* de *Industrial InsulationGroup LLC*.

##### Calcium Silicate T-12 Pipe Insulation

Date: February 1, 2002 - Superceding: April 16, 2001

PRODUCT CATEGORY: 1130 PRICE TYPE: 1130



List Prices Per Linear Foot - Single Layer

Pipe Size Inches	Wall Thickness (Inches)						
	1"	1 1/2"	2"	2.5	3"	3 1/2"	4"
1/2	6.06	6.40	10.11	12.71	18.33	26.29	35.31
3/4	6.06	6.50	10.66	13.29	18.46	27.95	35.36
1	5.80	7.10	11.23	13.55	18.59	27.95	35.36
1.25	5.90	7.64	12.04	13.70	18.85	27.95	36.66
1.5	5.95	8.22	12.56	14.33	19.01	29.17	39.39
2	6.86	9.05	13.18	14.95	19.60	31.46	39.99
2.5	7.25	9.88	15.86	18.23	23.06	35.13	45.34
3	7.80	10.32	15.91	18.38	23.24	36.37	46.49
3.5	9.62	11.88	18.10	21.84	28.42	41.26	51.51
4	9.75	11.96	18.41	22.75	29.69	43.11	54.60
4.5	10.01	13.26	20.72	28.60	32.06	45.66	58.58
5	10.14	13.47	21.01	29.02	32.19	46.98	59.15
6	10.66	13.75	22.96	30.26	36.22	47.68	63.70
7	-	16.15	24.88	32.34	40.27	55.56	69.21
8	-	18.23	27.35	35.07	43.13	60.61	78.00
9	-	21.87	30.94	39.65	48.91	66.64	80.91
10	-	24.13	34.35	42.82	52.08	75.32	89.70
11	-	27.46	36.22	45.76	55.61	81.90	92.17
12	-	28.63	38.17	47.76	57.54	90.17	98.49

Fuente: Steels (2002)

## 5. Códigos de las funciones en Matlab.

### Función xsteam

```
function Out=xsteam(fun,In1,In2)
%
%XSTEAM provides accurate steam and water properties from 0 - 1000 bar
and from 0 - 2000 °C
%   according to the standard IAPWS IF-97. For accuracy of the
functions in different
%   regions see IF-97 (www.iapws.org).
%
%   OUT = XSTEAM('FUN', IN)           or
%   OUT = XSTEAM('FUN', IN1, IN2)    return the water or steam
propertie. FUN must always be the steam table
%   function you want to use. The other argument(s) is(are) the
input(s) to that function.
%
%   Example: XSTEAM('Tsat_p', 1)      Returns the saturation temperature
of water at 1 bar.
%   Example: XSTEAM('h_pt', 1, 20)   Returns the enthalpy of water at 1
bar and 20 °C
%
%   The FUN syntax is: first the wanted property, then an _ and then
the argument properties.
%   Example: 'T_ph' is temperature as a function of pressure and
enthalpy.
%
%   VALID PROPERTIES:
%
%   T   Temperature      (°C)
%   p   Pressure         (bar)
%   h   Enthalpy         (kJ/kg)
%   rho Density          (kg/m3)
%   v   Specific volume  (m3/kg)
%   s   Specific entropy
%   u   Specific internal energy
%   Cp  Specific isobaric heat capacity (kJ/kg °C)
%   Cv  Specific isochoric heat capacity (kJ/kg °C)
%   w   Speed of sound
%   my  Viscosity        (Pa.s)
%   Pr  Prandtl number
%   tc  Thermal Conductivity
%   st  Surface Tension
%   x   Vapour fraction
%   vx  Vapour Volume Fraction
%
%   VALID FUNCTIONS:
%
%   TEMPERATURE
%   Tsat_p  Saturation temperature
%   T_ph    Temperture as a function of pressure and enthalpy
%   T_ps    Temperture as a function of pressure and entropy
%   T_hs    Temperture as a function of enthalpy and entropy
%
%   PRESSURE
%   psat_T  Saturation pressure
%   p_hs    Pressure as a function of h and s.
```

```

%   p_rho Pressure as a function of h and rho. Very unaccurate for
solid water region since
%       it's almost incompressible!
%
%
%   ENTHALPY
%   hV_p Saturated vapour enthalpy
%   hL_p Saturated liquid enthalpy
%   hV_T Saturated vapour enthalpy
%   hL_T Saturated liquid enthalpy
%   h_pT Entalpy as a function of pressure and temperature.
%   h_ps Entalpy as a function of pressure and entropy.
%   h_px Entalpy as a function of pressure and vapour fraction
%   h_rho Entalpy as a function of pressure and density. Observe
for low temperatures (liquid)
%       this equation has 2 solutions.
%   h_Tx Entalpy as a function of temperature and vapour fraction
%
%
%   SPECIFIC VOLUME
%   vV_p Saturated vapour volume
%   vL_p Saturated liquid volume
%   vV_T Saturated vapour volume
%   vL_T Saturated liquid volume
%   v_pT Specific volume as a function of pressure and temperature.
%   v_ph Specific volume as a function of pressure and enthalpy
%   v_ps Specific volume as a function of pressure and entropy.
%
%
%   DENSITY
%   rhoV_p Saturated vapour density
%   rhoL_p Saturated liquid density
%   rhoV_T Saturated vapour density
%   rhoL_T Saturated liquid density
%   rho_pT Density as a function of pressure and temperature.
%   rho_ph Density as a function of pressure and enthalpy
%   rho_ps Density as a function of pressure and entropy.
%
%
%   SPECIFIC ENTROPY
%   sV_p Saturated vapour entropy.
%   sL_p Saturated liquid entropy.
%   sV_T Saturated vapour entropy.
%   sL_T Saturated liquid entropy.
%   s_pT Specific entropy as a function of pressure and temperature
%       (Returns saturated vapour entalpy if mixture).
%   s_ph Specific entropy as a function of pressure and enthalpy.
%
%
%   SPECIFIC INTERNAL ENERGY
%   uV_p Saturated vapour internal energy.
%   uL_p Saturated liquid internal energy.
%   uV_T Saturated vapour internal energy.
%   uL_T Saturated liquid internal energy.
%   u_pT Specific internal energy as a function of pressure and
temperature.
%   u_ph Specific internal energy as a function of pressure and
enthalpy.
%   u_ps Specific internal energy as a function of pressure and
entropy.
%
%
%   SPECIFIC ISOBARIC HEAT CAPACITY
%   CpV_p Saturated vapour heat capacity.
%   CpL_p Saturated liquid heat capacity.
%   CpV_T Saturated vapour heat capacity.
%   CpL_T Saturated liquid heat capacity.

```

```

% Cp_pT Specific isobaric heat capacity as a function of pressure
and temperature.
% Cp_ph Specific isobaric heat capacity as a function of pressure
and enthalpy.
% Cp_ps Specific isobaric heat capacity as a function of pressure
and entropy.
%
% SPECIFIC ISOCHORIC HEAT CAPACITY
% CvV_p Saturated vapour isochoric heat capacity.
% CvL_p Saturated liquid isochoric heat capacity.
% CvV_T Saturated vapour isochoric heat capacity.
% CvL_T Saturated liquid isochoric heat capacity.
% Cv_pT Specific isochoric heat capacity as a function of
pressure and temperature.
% Cv_ph Specific isochoric heat capacity as a function of
pressure and enthalpy.
% Cv_ps Specific isochoric heat capacity as a function of
pressure and entropy.
%
% SPEED OF SOUND
% wV_p Saturated vapour speed of sound.
% wL_p Saturated liquid speed of sound.
% wV_T Saturated vapour speed of sound.
% wL_T Saturated liquid speed of sound.
% w_pT Speed of sound as a function of pressure and temperature.
% w_ph Speed of sound as a function of pressure and enthalpy.
% w_ps Speed of sound as a function of pressure and entropy.
%
% VISCOSITY
% Viscosity is not part of IAPWS Steam IF97. Equations from:
% "Revised Release on the IAPWS Formulation 1985 for the Viscosity
of Ordinary
% Water Substance", 2003 are used.
% Viscosity in the mixed region (4) is interpolated according to
the density.
% This is not true since it will be two fases.
% my_pT Viscosity as a function of pressure and temperature.
% my_ph Viscosity as a function of pressure and enthalpy.
% my_ps Viscosity as a function of pressure and entropy.
%
% PRANDTL NUMBER
% Pr_pT Prandtl number as a function of pressure and temperature.
% Pr_ph Prandtl number as a function of pressure and enthalpy.
%
% THERMAL CONDUCTIVITY
% Revised release on the IAPS Formulation 1985 for the Thermal
Conductivity of
% ordinary water substance (IAPWS 1998).
% tcL_p Saturated vapour thermal conductivity.
% tcV_p Saturated liquid thermal conductivity.
% tcL_T Saturated vapour thermal conductivity.
% tcV_T Saturated liquid thermal conductivity.
% tc_pT Thermal conductivity as a function of pressure and
temperature.
% tc_ph Thermal conductivity as a function of pressure and
enthalpy.
% tc_hs Thermal conductivity as a function of enthalpy and
entropy.
%
% SURFACE TENSION

```



```

%      st_T  Surface tension for two phase water/steam as a function of
T.
%      st_p  Surface tension for two phase water/steam as a function of
T.
%
%      VAPOUR FRACTION
%      x_ph  Vapour fraction as a function of pressure and enthalpy.
%      x_ps  Vapour fraction as a function of pressure and entropy.
%
%      VAPOUR VOLUME FRACTION
%      vx_ph Vapour volume fraction as a function of pressure and
enthalpy.
%      vx_ps Vapour volume fraction as a function of pressure and
entropy.
%
%      h_prho behöver T_prho för samtliga regioner!!!!
%
%
*****
*****
% * Water and steam properties according to IAPWS IF-97
%
% * By Magnus Holmgren, magnus@x-eng.com, http://www.x-eng.com/
%
% *
%
% * The steam tables are free and provided as is.
%
% * We take no responsibilities for any errors in the code or damage
thereby.
% * You are free to use, modify and distribute the code as long as
authorship is properly acknowledged.
% * Please notify me at magnus@x-eng.com if the code is used in
commercial applications
*****
*****

%*CONTENTS:
%
%*1 Calling functions
%*1.1
%*1.2 Temperature (T)
%*1.3 Pressure (p)
%*1.4 Enthalpy (h)
%*1.5 Specific Volume (v)
%*1.6 Density (rho)
%*1.7 Specific entropy (s)
%*1.8 Specific internal energy (u)
%*1.9 Specific isobaric heat capacity (Cp)
%*1.10 Specific isochoric heat capacity (Cv)
%*1.11 Speed of sound
%*1.12 Viscosity
%*1.13 Prandtl
%*1.14 Kappa
%*1.15 Surface tension
%*1.16 Heat conductivity
%*1.17 Vapour fraction
%*1.18 Vapour Volume Fraction
%
%*2 IAPWS IF 97 Calling functions

```

```

%*2.1 Functions for region 1
%*2.2 Functions for region 2
%*2.3 Functions for region 3
%*2.4 Functions for region 4
%*2.5 Functions for region 5
%
%*3 Region Selection
%*3.1 Regions as a function of pT
%*3.2 Regions as a function of ph
%*3.3 Regions as a function of ps
%*3.4 Regions as a function of hs
%
%4 Region Borders
%4.1 Boundary between region 1 and 3.
%4.2 Region 3. pSat_h and pSat_s
%4.3 Region boundary 1to3 and 3to2 as a functions of s
%
%5 Transport properties
%5.1 Viscosity (IAPWS formulation 1985)
%5.2 Thermal Conductivity (IAPWS formulation 1985)
%5.3 Surface Tension
%
%6 Units
%
%7 Verification
%7.1 Verifiy region 1
%7.2 Verifiy region 2
%7.3 Verifiy region 3
%7.4 Verifiy region 4
%7.5 Verifiy region 5

%*****
%*****
%*1 Calling functions
*
%*****
%*****

%*****
%*****
%*1.1

fun=lower(fun);

switch fun

%*****
%*****
    %*1.2 Temperature
case 'tsat_p'
    p = toSIunit_p(In1);
    if p > 0.000611657 & p < 22.06395
        Out = fromSIunit_T(T4_p(p));
    else
        Out = NaN;
    end
case 'tsat_s'
    s = toSIunit_s(In1);
    if s > -0.0001545495919 & s < 9.155759395

```

```

        ps = p4_s(s);
        Out = fromSIunit_T(T4_p(ps));
    else
        Out = NaN;
    end
case 't_ph'
    p = toSIunit_p(In1);
    h = toSIunit_h(In2);
    Region = region_ph(p, h);
    switch Region
    case 1
        Out = fromSIunit_T(T1_ph(p, h));
    case 2
        Out = fromSIunit_T(T2_ph(p, h));
    case 3
        Out = fromSIunit_T(T3_ph(p, h));
    case 4
        Out = fromSIunit_T(T4_p(p));
    case 5
        Out = fromSIunit_T(T5_ph(p, h));
    otherwise
        Out = NaN;
    end
case 't_ps'
    p = toSIunit_p(In1);
    s = toSIunit_s(In2);
    Region = region_ps(p, s);
    switch Region
    case 1
        Out = fromSIunit_T(T1_ps(p, s));
    case 2
        Out = fromSIunit_T(T2_ps(p, s));
    case 3
        Out = fromSIunit_T(T3_ps(p, s));
    case 4
        Out = fromSIunit_T(T4_p(p));
    case 5
        Out = fromSIunit_T(T5_ps(p, s));
    otherwise
        Out = NaN;
    end
case 't_hs'
    h = toSIunit_h(In1);
    s = toSIunit_s(In2);
    Region = region_hs(h, s);
    switch Region
    case 1
        p1 = p1_hs(h, s);
        Out = fromSIunit_T(T1_ph(p1, h));
    case 2
        p2 = p2_hs(h, s);
        Out = fromSIunit_T(T2_ph(p2, h));
    case 3
        p3 = p3_hs(h, s);
        Out = fromSIunit_T(T3_ph(p3, h));
    case 4
        Out = fromSIunit_T(T4_hs(h, s));
    case 5
        error('functions of hs is not avlaible in region 5');
    otherwise

```

```

        Out = NaN;
    end

%*****
%*****
%*1.3 Pressure (p)
case 'psat_t'
    T = toSIunit_T(In1);
    if T < 647.096 & T > 273.15
        Out = fromSIunit_p(p4_T(T));
    else
        Out = NaN;
    end

case 'psat_s'
    s = toSIunit_s(In1);
    if s > -0.0001545495919 & s < 9.155759395
        Out = fromSIunit_p(p4_s(s));
    else
        Out = NaN;
    end

case 'p_hs'
    h = toSIunit_h(In1);
    s = toSIunit_s(In2);
    Region = region_hs(h, s);
    switch Region
    case 1
        Out = fromSIunit_p(p1_hs(h, s));
    case 2
        Out = fromSIunit_p(p2_hs(h, s));
    case 3
        Out = fromSIunit_p(p3_hs(h, s));
    case 4
        tSat = T4_hs(h, s);
        Out = fromSIunit_p(p4_T(tSat));
    case 5
        error('functions of hs is not avlaible in region 5');
    otherwise
        Out = NaN;
    end

    case 'p_hrho'
        h=In1;
        rho=In2;
%Not valid for water or sumpercritical since water rho does not change
very much with p.
%Uses iteration to find p.
        High_Bound = fromSIunit_p(100);
        Low_Bound = fromSIunit_p(0.000611657);
        ps = fromSIunit_p(10);
        rhos = 1 / XSteam('v_ph',ps, h);
        while abs(rho - rhos) > 0.0000001
            rhos = 1 / XSteam('v_ph',ps, h);
            if rhos >= rho
                High_Bound = ps;
            else
                Low_Bound = ps;
            end
            ps = (Low_Bound + High_Bound) / 2;
        end
end

```

```

Out = ps;

%*****
%*****
%*1.4 Enthalpy (h)
case 'hv_p'
    p = toSIunit_p(In1);
    if p > 0.000611657 & p < 22.06395
        Out = fromSIunit_h(h4V_p(p));
    else
        Out = NaN;
    end

case 'hl_p'
    p = toSIunit_p(In1);
    if p > 0.000611657 & p < 22.06395
        Out = fromSIunit_h(h4L_p(p));
    else
        Out = NaN;
    end

case 'hv_t'
    T = toSIunit_T(In1);
    if T > 273.15 & T < 647.096
        p = p4_T(T);
        Out = fromSIunit_h(h4V_p(p));
    else
        Out = NaN;
    end

case 'hl_t'
    T = toSIunit_T(In1);
    if T > 273.15 & T < 647.096
        p = p4_T(T);
        Out = fromSIunit_h(h4L_p(p));
    else
        Out = NaN;
    end

case 'h_pt'
    p = toSIunit_p(In1);
    T = toSIunit_T(In2);
    Region = region_pT(p, T);
    switch Region
    case 1
        Out = fromSIunit_h(h1_pT(p, T));
    case 2
        Out = fromSIunit_h(h2_pT(p, T));
    case 3
        Out = fromSIunit_h(h3_pT(p, T));
    case 4
        Out = NaN;
    case 5
        Out = fromSIunit_h(h5_pT(p, T));
    otherwise
        Out = NaN;
    end
end

```

```

case 'h_ps'
    p = toSIunit_p(In1);
    s = toSIunit_s(In2);
    Region = region_ps(p, s);
    switch Region
    case 1
        Out = fromSIunit_h(h1_pT(p, T1_ps(p, s)));
    case 2
        Out = fromSIunit_h(h2_pT(p, T2_ps(p, s)));
    case 3
        Out = fromSIunit_h(h3_rhoT(1 / v3_ps(p, s), T3_ps(p, s)));
    case 4
        xs = x4_ps(p, s);
        Out = fromSIunit_h(xs * h4V_p(p) + (1 - xs) * h4L_p(p));
    case 5
        Out = fromSIunit_h(h5_pT(p, T5_ps(p, s)));
    otherwise
        Out = NaN;
    end

case 'h_px'
    p = toSIunit_p(In1);
    x = toSIunit_x(In2);
    if x > 1 | x < 0 | p >= 22.064
        Out = NaN;
        return
    end
    hL = h4L_p(p);
    hV = h4V_p(p);
    Out = hL + x * (hV - hL);

case 'h_prho'
    p = toSIunit_p(In1);
    rho = 1 / toSIunit_v(1 / In2);
    Region = Region_prho(p, rho);
    switch Region
    case 1
        Out = fromSIunit_h(h1_pT(p, T1_prho(p, rho)));
    case 2
        Out = fromSIunit_h(h2_pT(p, T2_prho(p, rho)));
    case 3
        Out = fromSIunit_h(h3_rhoT(rho, T3_prho(p, rho)));
    case 4
        if p < 16.529
            vV = v2_pT(p, T4_p(p));
            vL = v1_pT(p, T4_p(p));
        else
            vV = v3_ph(p, h4V_p(p));
            vL = v3_ph(p, h4L_p(p));
        end
        hV = h4V_p(p);
        hL = h4L_p(p);
        x = (1 / rho - vL) / (vV - vL);
        Out = fromSIunit_h((1 - x) * hL + x * hV);
    case 5
        Out = fromSIunit_h(h5_pT(p, T5_prho(p, rho)));
    otherwise
        Out = NaN;
    end

case 'h_tx'

```

```

T = toSIunit_T(In1);
x = toSIunit_x(In2);
if x > 1 | x < 0 | T >= 647.096
    Out = NaN;
    return
end
p = p4_T(T);
hL = h4L_p(p);
hV = h4V_p(p);
Out = hL + x * (hV - hL);

%*****
%*****
%*1.5 Specific Volume (v)
case {'vv_p', 'rhov_p'}
    p = toSIunit_p(In1);
    if p > 0.000611657 & p < 22.06395
        if p < 16.529
            Out = fromSIunit_v(v2_pT(p, T4_p(p)));
        else
            Out = fromSIunit_v(v3_ph(p, h4V_p(p)));
        end
    else
        Out = NaN;
    end
    if fun(1)=='r'
        Out=1/Out;
    end

case {'vl_p', 'rhol_p'}
    p = toSIunit_p(In1);
    if p > 0.000611657 & p < 22.06395
        if p < 16.529
            Out = fromSIunit_v(v1_pT(p, T4_p(p)));
        else
            Out = fromSIunit_v(v3_ph(p, h4L_p(p)));
        end
    else
        Out = NaN;
    end
    if fun(1)=='r'
        Out=1/Out;
    end

case {'vv_t', 'rhov_t'}
    T = toSIunit_T(In1);
    if T > 273.15 & T < 647.096
        if T <= 623.15
            Out = fromSIunit_v(v2_pT(p4_T(T), T));
        else
            Out = fromSIunit_v(v3_ph(p4_T(T), h4V_p(p4_T(T))));
        end
    else
        Out = NaN;
    end
    if fun(1)=='r'
        Out=1/Out;
    end
end

```

```

case {'v1_t','rho1_t'}
    T = toSIunit_T(In1);
    if T > 273.15 & T < 647.096
        if T <= 623.15
            Out = fromSIunit_v(v1_pT(p4_T(T), T));
        else
            Out = fromSIunit_v(v3_ph(p4_T(T), h4L_p(p4_T(T))));
        end
    else
        Out = NaN;
    end
    if fun(1)=='r'
        Out=1/Out;
    end

case {'v_pt','rho_pt'}
    p = toSIunit_p(In1);
    T = toSIunit_T(In2);
    Region = region_pT(p, T);
    switch Region
    case 1
        Out = fromSIunit_v(v1_pT(p, T));
    case 2
        Out = fromSIunit_v(v2_pT(p, T));
    case 3
        Out = fromSIunit_v(v3_ph(p, h3_pT(p, T)));
    case 4
        Out = NaN;
    case 5
        Out = fromSIunit_v(v5_pT(p, T));
    otherwise
        Out = NaN;
    end
    if fun(1)=='r'
        Out=1/Out;
    end

case {'v_ph','rho_ph'}
    p = toSIunit_p(In1);
    h = toSIunit_h(In2);
    Region = region_ph(p, h);
    switch Region
    case 1
        Out = fromSIunit_v(v1_pT(p, T1_ph(p, h)));
    case 2
        Out = fromSIunit_v(v2_pT(p, T2_ph(p, h)));
    case 3
        Out = fromSIunit_v(v3_ph(p, h));
    case 4
        xs = x4_ph(p, h);
        if p < 16.529
            v4v = v2_pT(p, T4_p(p));
            v4L = v1_pT(p, T4_p(p));
        else
            v4v = v3_ph(p, h4V_p(p));
            v4L = v3_ph(p, h4L_p(p));
        end
        Out = fromSIunit_v((xs * v4v + (1 - xs) * v4L));
    case 5

```



```

        Ts = T5_ph(p, h);
        Out = fromSIunit_v(v5_pT(p, Ts));
    otherwise
        Out = NaN;
    end
    if fun(1)=='r'
        Out=1/Out;
    end

case {'v_ps','rho_ps'}
    p = toSIunit_p(In1);
    s = toSIunit_s(In2);
    Region = region_ps(p, s);
    switch Region
    case 1
        Out = fromSIunit_v(v1_pT(p, T1_ps(p, s)));
    case 2
        Out = fromSIunit_v(v2_pT(p, T2_ps(p, s)));
    case 3
        Out = fromSIunit_v(v3_ps(p, s));
    case 4
        xs = x4_ps(p, s);
        if p < 16.529
            v4v = v2_pT(p, T4_p(p));
            v4L = v1_pT(p, T4_p(p));
        else
            v4v = v3_ph(p, h4V_p(p));
            v4L = v3_ph(p, h4L_p(p));
        end
        Out = fromSIunit_v((xs * v4v + (1 - xs) * v4L));
    case 5
        Ts = T5_ps(p, s);
        Out = fromSIunit_v(v5_pT(p, Ts));
    otherwise
        Out = NaN;
    end
    if fun(1)=='r'
        Out=1/Out;
    end

%*****
%*****
%*1.6 Density (rho)
% Density is calculated as 1/v. Se section 1.5 Volume

%*****
%*****
%*1.7 Specific entropy (s)
case 'sv_p'
    p = toSIunit_p(In1);
    if p > 0.000611657 & p < 22.06395
        if p < 16.529
            Out = fromSIunit_s(s2_pT(p, T4_p(p)));
        else
            Out = fromSIunit_s(s3_rhoT(1 / (v3_ph(p, h4V_p(p))),
T4_p(p)));
        end
    else
        Out = NaN;
    end
end

```

```

case 'sl_p'
    p = toSIunit_p(In1);
    if p > 0.000611657 & p < 22.06395
        if p < 16.529
            Out = fromSIunit_s(s1_pT(p, T4_p(p)));
        else
            Out = fromSIunit_s(s3_rhoT(1 / (v3_ph(p, h4L_p(p))),
T4_p(p)));
        end
    else
        Out = NaN;
    end

case 'sv_t'
    T = toSIunit_T(In1);
    if T > 273.15 & T < 647.096
        if T <= 623.15
            Out = fromSIunit_s(s2_pT(p4_T(T), T));
        else
            Out = fromSIunit_s(s3_rhoT(1 / (v3_ph(p4_T(T),
h4V_p(p4_T(T)))), T));
        end
    else
        Out = NaN;
    end

case 'sl_t'
    T = toSIunit_T(In1);
    if T > 273.15 & T < 647.096
        if T <= 623.15
            Out = fromSIunit_s(s1_pT(p4_T(T), T));
        else
            Out = fromSIunit_s(s3_rhoT(1 / (v3_ph(p4_T(T),
h4L_p(p4_T(T)))), T));
        end
    else
        Out = NaN;
    end

case 's_pt'
    p = toSIunit_p(In1);
    T = toSIunit_T(In2);
    Region = region_pT(p, T);
    switch Region
    case 1
        Out = fromSIunit_s(s1_pT(p, T));
    case 2
        Out = fromSIunit_s(s2_pT(p, T));
    case 3
        hs = h3_pT(p, T);
        rhos = 1 / v3_ph(p, hs);
        Out = fromSIunit_s(s3_rhoT(rhos, T));
    case 4
        Out = NaN;
    case 5
        Out = fromSIunit_s(s5_pT(p, T));
    otherwise
        Out = NaN;
    end
end

```

```

case 's_ph'
    p = toSIunit_p(In1);
    h = toSIunit_h(In2);
    Region = region_ph(p, h);
    switch Region
    case 1
        T = T1_ph(p, h);
        Out = fromSIunit_s(s1_pT(p, T));
    case 2
        T = T2_ph(p, h);
        Out = fromSIunit_s(s2_pT(p, T));
    case 3
        rhos = 1 / v3_ph(p, h);
        Ts = T3_ph(p, h);
        Out = fromSIunit_s(s3_rhoT(rhos, Ts));
    case 4
        Ts = T4_p(p);
        xs = x4_ph(p, h);
        if p < 16.529
            s4v = s2_pT(p, Ts);
            s4L = s1_pT(p, Ts);
        else
            v4v = v3_ph(p, h4V_p(p));
            s4v = s3_rhoT(1 / v4v, Ts);
            v4L = v3_ph(p, h4L_p(p));
            s4L = s3_rhoT(1 / v4L, Ts);
        end
        Out = fromSIunit_s((xs * s4v + (1 - xs) * s4L));
    case 5
        T = T5_ph(p, h);
        Out = fromSIunit_s(s5_pT(p, T));
    otherwise
        Out = NaN;
    end

%*****
%*****
%*1.8 Specific internal energy (u)
case 'uv_p'
    p = toSIunit_p(In1);
    if p > 0.000611657 & p < 22.06395
        if p < 16.529
            Out = fromSIunit_u(u2_pT(p, T4_p(p)));
        else
            Out = fromSIunit_u(u3_rhoT(1 / (v3_ph(p, h4V_p(p))),
T4_p(p)));
        end
    else
        Out = NaN;
    end

case 'ul_p'
    p = toSIunit_p(In1);
    if p > 0.000611657 & p < 22.06395
        if p < 16.529
            Out = fromSIunit_u(u1_pT(p, T4_p(p)));
        else

```

```

        Out = fromSIunit_u(u3_rhoT(1 / (v3_ph(p, h4L_p(p))),
T4_p(p)));
    end
    else
        Out = NaN;
    end

case 'uv_t'
    T = toSIunit_T(In1);
    if T > 273.15 & T < 647.096
        if T <= 623.15
            Out = fromSIunit_u(u2_pT(p4_T(T), T));
        else
            Out = fromSIunit_u(u3_rhoT(1 / (v3_ph(p4_T(T),
h4V_p(p4_T(T)))), T));
        end
    else
        Out = NaN;
    end

case 'ul_t'
    T = toSIunit_T(In1);
    if T > 273.15 & T < 647.096
        if T <= 623.15
            Out = fromSIunit_u(u1_pT(p4_T(T), T));
        else
            Out = fromSIunit_u(u3_rhoT(1 / (v3_ph(p4_T(T),
h4L_p(p4_T(T)))), T));
        end
    else
        Out = NaN;
    end

case 'u_pt'
    p = toSIunit_p(In1);
    T = toSIunit_T(In2);
    Region = region_pT(p, T);
    switch Region
    case 1
        Out = fromSIunit_u(u1_pT(p, T));
    case 2
        Out = fromSIunit_u(u2_pT(p, T));
    case 3
        hs = h3_pT(p, T);
        rhos = 1 / v3_ph(p, hs);
        Out = fromSIunit_u(u3_rhoT(rhos, T));
    case 4
        Out = NaN;
    case 5
        Out = fromSIunit_u(u5_pT(p, T));
    otherwise
        Out = NaN;
    end

case 'u_ph'
    p = toSIunit_p(In1);
    h = toSIunit_h(In2);
    Region = region_ph(p, h);
    switch Region
    case 1
        Ts = T1_ph(p, h);

```

```

        Out = fromSIunit_u(u1_pT(p, Ts));
    case 2
        Ts = T2_ph(p, h);
        Out = fromSIunit_u(u2_pT(p, Ts));
    case 3
        rhos = 1 / v3_ph(p, h);
        Ts = T3_ph(p, h);
        Out = fromSIunit_u(u3_rhoT(rhos, Ts));
    case 4
        Ts = T4_p(p);
        xs = x4_ph(p, h);
        if p < 16.529
            u4v = u2_pT(p, Ts);
            u4L = u1_pT(p, Ts);
        else
            v4v = v3_ph(p, h4V_p(p));
            u4v = u3_rhoT(1 / v4v, Ts);
            v4L = v3_ph(p, h4L_p(p));
            u4L = u3_rhoT(1 / v4L, Ts);
        end
        Out = fromSIunit_u((xs * u4v + (1 - xs) * u4L));
    case 5
        Ts = T5_ph(p, h);
        Out = fromSIunit_u(u5_pT(p, Ts));
    otherwise
        Out = NaN;
    end

case 'u_ps'
    p = toSIunit_p(In1);
    s = toSIunit_s(In2);
    Region = region_ps(p, s);
    switch Region
    case 1
        Ts = T1_ps(p, s);
        Out = fromSIunit_u(u1_pT(p, Ts));
    case 2
        Ts = T2_ps(p, s);
        Out = fromSIunit_u(u2_pT(p, Ts));
    case 3
        rhos = 1 / v3_ps(p, s);
        Ts = T3_ps(p, s);
        Out = fromSIunit_u(u3_rhoT(rhos, Ts));
    case 4
        if p < 16.529
            uLp = u1_pT(p, T4_p(p));
            uVp = u2_pT(p, T4_p(p));
        else
            uLp = u3_rhoT(1 / (v3_ph(p, h4L_p(p))), T4_p(p));
            uVp = u3_rhoT(1 / (v3_ph(p, h4V_p(p))), T4_p(p));
        end
        xs = x4_ps(p, s);
        Out = fromSIunit_u((xs * uVp + (1 - xs) * uLp));
    case 5
        Ts = T5_ps(p, s);
        Out = fromSIunit_u(u5_pT(p, Ts));
    otherwise
        Out = NaN;
    end
end

```

```

%*****
%*****
%*1.9 Specific isobaric heat capacity (Cp)
case 'cpv_p'
    p = toSIunit_p(In1);
    if p > 0.000611657 & p < 22.06395
        if p < 16.529
            Out = fromSIunit_Cp(Cp2_pT(p, T4_p(p)));
        else
            Out = fromSIunit_Cp(Cp3_rhoT(1 / (v3_ph(p, h4V_p(p))),
T4_p(p)));
        end
    else
        Out = NaN;
    end

case 'cpl_p'
    p = toSIunit_p(In1);
    if p > 0.000611657 & p < 22.06395
        if p < 16.529
            Out = fromSIunit_Cp(Cp1_pT(p, T4_p(p)));
        else
            Out = fromSIunit_Cp(Cp3_rhoT(1 / (v3_ph(p, h4L_p(p))),
T4_p(p)));
        end
    else
        Out = NaN;
    end

case 'cpv_t'
    T = toSIunit_T(In1);
    if T > 273.15 & T < 647.096
        if T <= 623.15
            Out = fromSIunit_Cp(Cp2_pT(p4_T(T), T));
        else
            Out = fromSIunit_Cp(Cp3_rhoT(1 / (v3_ph(p4_T(T),
h4V_p(p4_T(T)))), T));
        end
    else
        Out = NaN;
    end

case 'cpl_t'
    T = toSIunit_T(In1);
    if T > 273.15 & T < 647.096
        if T <= 623.15
            Out = fromSIunit_Cp(Cp1_pT(p4_T(T), T));
        else
            Out = fromSIunit_Cp(Cp3_rhoT(1 / (v3_ph(p4_T(T),
h4L_p(p4_T(T)))), T));
        end
    else
        Out = NaN;
    end

case 'cp_pt'
    p = toSIunit_p(In1);
    T = toSIunit_T(In2);
    Region = region_pT(p, T);
    switch Region

```

```

case 1
    Out = fromSIunit_Cp(Cp1_pT(p, T));
case 2
    Out = fromSIunit_Cp(Cp2_pT(p, T));
case 3
    hs = h3_pT(p, T);
    rhos = 1 / v3_ph(p, hs);
    Out = fromSIunit_Cp(Cp3_rhoT(rhos, T));
case 4
    Out = NaN;
case 5
    Out = fromSIunit_Cp(Cp5_pT(p, T));
otherwise
    Out = NaN;
end

case 'cp_ph'
p = toSIunit_p(In1);
h = toSIunit_h(In2);
Region = region_ph(p, h);
switch Region
case 1
    Ts = T1_ph(p, h);
    Out = fromSIunit_Cp(Cp1_pT(p, Ts));
case 2
    Ts = T2_ph(p, h);
    Out = fromSIunit_Cp(Cp2_pT(p, Ts));
case 3
    rhos = 1 / v3_ph(p, h);
    Ts = T3_ph(p, h);
    Out = fromSIunit_Cp(Cp3_rhoT(rhos, Ts));
case 4
    Out = NaN;
case 5
    Ts = T5_ph(p, h);
    Out = fromSIunit_Cp(Cp5_pT(p, Ts));
otherwise
    Out = NaN;
end

case 'cp_ps'
p = toSIunit_p(In1);
s = toSIunit_s(In2);
Region = region_ps(p, s);
switch Region
case 1
    Ts = T1_ps(p, s);
    Out = fromSIunit_Cp(Cp1_pT(p, Ts));
case 2
    Ts = T2_ps(p, s);
    Out = fromSIunit_Cp(Cp2_pT(p, Ts));
case 3
    rhos = 1 / v3_ps(p, s);
    Ts = T3_ps(p, s);
    Out = fromSIunit_Cp(Cp3_rhoT(rhos, Ts));
case 4
    Out = NaN;
case 5
    Ts = T5_ps(p, s);
    Out = fromSIunit_Cp(Cp5_pT(p, Ts));
otherwise

```

```

        Out = NaN;
    end

%*****
%*****
%*1.10 Specific isochoric heat capacity (Cv)
case 'cvv_p'
    p = toSIunit_p(In1);
    if p > 0.000611657 & p < 22.06395
        if p < 16.529
            Out = fromSIunit_Cv(Cv2_pT(p, T4_p(p)));
        else
            Out = fromSIunit_Cv(Cv3_rhoT(1 / (v3_ph(p, h4V_p(p)),
T4_p(p)));
        end
    else
        Out = NaN;
    end

case 'cvl_p'
    p = toSIunit_p(In1);
    if p > 0.000611657 & p < 22.06395
        if p < 16.529
            Out = fromSIunit_Cv(Cv1_pT(p, T4_p(p)));
        else
            Out = fromSIunit_Cv(Cv3_rhoT(1 / (v3_ph(p, h4L_p(p)),
T4_p(p)));
        end
    else
        Out = NaN;
    end

case 'cvv_t'
    T = toSIunit_T(In1);
    if T > 273.15 & T < 647.096
        if T <= 623.15
            Out = fromSIunit_Cv(Cv2_pT(p4_T(T), T));
        else
            Out = fromSIunit_Cv(Cv3_rhoT(1 / (v3_ph(p4_T(T),
h4V_p(p4_T(T))), T));
        end
    else
        Out = NaN;
    end

case 'cvl_t'
    T = toSIunit_T(In1);
    if T > 273.15 & T < 647.096
        if T <= 623.15
            Out = fromSIunit_Cv(Cv1_pT(p4_T(T), T));
        else
            Out = fromSIunit_Cv(Cv3_rhoT(1 / (v3_ph(p4_T(T),
h4L_p(p4_T(T))), T));
        end
    else
        Out = NaN;
    end

case 'cv_pt'

```



```

p = toSIunit_p(In1);
T = toSIunit_T(In2);
Region = region_pT(p, T);
switch Region
case 1
    Out = fromSIunit_Cv(Cv1_pT(p, T));
case 2
    Out = fromSIunit_Cv(Cv2_pT(p, T));
case 3
    hs = h3_pT(p, T);
    rhos = 1 / v3_ph(p, hs);
    Out = fromSIunit_Cv(Cv3_rhoT(rhos, T));
case 4
    Out = NaN;
case 5
    Out = fromSIunit_Cv(Cv5_pT(p, T));
otherwise
    Out = NaN;
end

case 'cv_ph'
p = toSIunit_p(In1);
h = toSIunit_h(In2);
Region = region_ph(p, h);
switch Region
case 1
    Ts = T1_ph(p, h);
    Out = fromSIunit_Cv(Cv1_pT(p, Ts));
case 2
    Ts = T2_ph(p, h);
    Out = fromSIunit_Cv(Cv2_pT(p, Ts));
case 3
    rhos = 1 / v3_ph(p, h);
    Ts = T3_ph(p, h);
    Out = fromSIunit_Cv(Cv3_rhoT(rhos, Ts));
case 4
    Out = NaN;
case 5
    Ts = T5_ph(p, h);
    Out = fromSIunit_Cv(Cv5_pT(p, Ts));
otherwise
    Out = NaN;
end

case 'cv_ps'
p = toSIunit_p(In1);
s = toSIunit_s(In2);
Region = region_ps(p, s);
switch Region
case 1
    Ts = T1_ps(p, s);
    Out = fromSIunit_Cv(Cv1_pT(p, Ts));
case 2
    Ts = T2_ps(p, s);
    Out = fromSIunit_Cv(Cv2_pT(p, Ts));
case 3
    rhos = 1 / v3_ps(p, s);
    Ts = T3_ps(p, s);
    Out = fromSIunit_Cv(Cv3_rhoT(rhos, Ts));
case 4

```

```

        Out = NaN;  %(xs * CvVp + (1 - xs) * CvIp) / Cv_scale -
Cv_offset
    case 5
        Ts = T5_ps(p, s);
        Out = fromSIunit_Cv(Cv5_pT(p, Ts));
    otherwise
        Out = CVErr(xlErrValue);
    end

%*****
%*****
%*1.11 Speed of sound
case 'wv_p'
    p = toSIunit_p(In1);
    if p > 0.000611657 & p < 22.06395
        if p < 16.529
            Out = fromSIunit_w(w2_pT(p, T4_p(p)));
        else
            Out = fromSIunit_w(w3_rhoT(1 / (v3_ph(p, h4V_p(p))),
T4_p(p)));
        end
    else
        Out = NaN;
    end

case 'wl_p'
    p = toSIunit_p(In1);
    if p > 0.000611657 & p < 22.06395
        if p < 16.529
            Out = fromSIunit_w(w1_pT(p, T4_p(p)));
        else
            Out = fromSIunit_w(w3_rhoT(1 / (v3_ph(p, h4L_p(p))),
T4_p(p)));
        end
    else
        Out = NaN;
    end

case 'wv_t'
    T = toSIunit_T(In1);
    if T > 273.15 & T < 647.096
        if T <= 623.15
            Out = fromSIunit_w(w2_pT(p4_T(T), T));
        else
            Out = fromSIunit_w(w3_rhoT(1 / (v3_ph(p4_T(T),
h4V_p(p4_T(T))), T)));
        end
    else
        Out = NaN;
    end

case 'wl_t'
    T = toSIunit_T(In1);
    if T > 273.15 & T < 647.096
        if T <= 623.15
            Out = fromSIunit_w(w1_pT(p4_T(T), T));
        else
    
```

```

        Out = fromSIunit_w(w3_rhoT(1 / (v3_ph(p4_T(T),
h4L_p(p4_T(T)))), T));
    end
    else
        Out = NaN;
    end

case 'w_pt'
    p = toSIunit_p(In1);
    T = toSIunit_T(In2);
    Region = region_pT(p, T);
    switch Region
    case 1
        Out = fromSIunit_w(w1_pT(p, T));
    case 2
        Out = fromSIunit_w(w2_pT(p, T));
    case 3
        hs = h3_pT(p, T);
        rhos = 1 / v3_ph(p, hs);
        Out = fromSIunit_w(w3_rhoT(rhos, T));
    case 4
        Out = NaN;
    case 5
        Out = fromSIunit_w(w5_pT(p, T));
    otherwise
        Out = NaN;
    end

case 'w_ph'
    p = toSIunit_p(In1);
    h = toSIunit_h(In2);
    Region = region_ph(p, h);
    switch Region
    case 1
        Ts = T1_ph(p, h);
        Out = fromSIunit_w(w1_pT(p, Ts));
    case 2
        Ts = T2_ph(p, h);
        Out = fromSIunit_w(w2_pT(p, Ts));
    case 3
        rhos = 1 / v3_ph(p, h);
        Ts = T3_ph(p, h);
        Out = fromSIunit_w(w3_rhoT(rhos, Ts));
    case 4
        Out = NaN;
    case 5
        Ts = T5_ph(p, h);
        Out = fromSIunit_w(w5_pT(p, Ts));
    otherwise
        Out = NaN;
    end

case 'w_ps'
    p = toSIunit_p(In1);
    s = toSIunit_s(In2);
    Region = region_ps(p, s);
    switch Region
    case 1
        Ts = T1_ps(p, s);
        Out = fromSIunit_w(w1_pT(p, Ts));

```

```

    case 2
        Ts = T2_ps(p, s);
        Out = fromSIunit_w(w2_pT(p, Ts));
    case 3
        rhos = 1 / v3_ps(p, s);
        Ts = T3_ps(p, s);
        Out = fromSIunit_w(w3_rhoT(rhos, Ts));
    case 4
        Out = NaN; %(xs * wVp + (1 - xs) * wLp) / w_scale - w_offset
    case 5
        Ts = T5_ps(p, s);
        Out = fromSIunit_w(w5_pT(p, Ts));
    otherwise
        Out = NaN;
    end

%*****
%*****
%*1.12 Viscosity
case 'my_pt'
    p = toSIunit_p(In1);
    T = toSIunit_T(In2);
    Region = region_pT(p, T);
    switch Region
    case 4
        Out = NaN;
    case {1, 2, 3, 5}
        Out = fromSIunit_my(my_AllRegions_pT(p, T));
    otherwise
        Out = NaN;
    end

case {'my_ph'}
    p = toSIunit_p(In1);
    h = toSIunit_h(In2);
    Region = region_ph(p, h);
    switch Region
    case {1, 2, 3, 5}
        Out = fromSIunit_my(my_AllRegions_ph(p, h));
    case {4}
        Out = NaN;
    otherwise
        Out = NaN;
    end

case 'my_ps'
    h = XSteam('h_ps', In1, In2);
    Out = XSteam('my_ph', In1, h);

%*****
%*****
%*1.13 Prandtl
case 'pr_pt'
    Cp = toSIunit_Cp(XSteam('Cp_pT', In1, In2));
    my = toSIunit_my(XSteam('my_pT', In1, In2));
    tc = toSIunit_tc(XSteam('tc_pT', In1, In2));
    Out = Cp * 1000 * my / tc;

    case 'pr_ph'
        Cp = toSIunit_Cp(XSteam('Cp_ph', In1, In2));

```

```

my = toSIunit_my(XSteam('my_ph',In1,In2));
tc = toSIunit_tc(XSteam('tc_ph',In1,In2));
Out = Cp * 1000 * my / tc;

%*****
%*****
%*1.14 Kappa

%*****
%*****

%*****
%*****
%*1.15 Surface tension
case 'st_t'
    T = toSIunit_T(In1);
    Out = fromSIunit_st(Surface_Tension_T(T));

case 'st_p'
    T = XSteam('Tsat_p',In1);
    T = toSIunit_T(T);
    Out = fromSIunit_st(Surface_Tension_T(T));

%*****
%*****
%*1.16 Thermal conductivity
case 'tcl_p'
    T = XSteam('Tsat_p',In1);
    v = XSteam('vL_p',In1);
    p = toSIunit_p(In1);
    T = toSIunit_T(T);
    v = toSIunit_v(v);
    rho = 1 / v;
    Out = fromSIunit_tc(tc_ptrho(p, T, rho));

case 'tcv_p'
    ps = In1;
    T = XSteam('Tsat_p',ps);
    v = XSteam('vV_p',ps);
    p = toSIunit_p(In1);
    T = toSIunit_T(T);
    v = toSIunit_v(v);
    rho = 1 / v;
    Out = fromSIunit_tc(tc_ptrho(p, T, rho));

case 'tcl_t'
    Ts = In1;
    p = XSteam('psat_T',Ts);
    v = XSteam('vL_T',Ts);
    p = toSIunit_p(p);
    T = toSIunit_T(Ts);
    v = toSIunit_v(v);
    rho = 1 / v;
    Out = fromSIunit_tc(tc_ptrho(p, T, rho));

case 'tcv_t'
    Ts = In1;

```

```

    p = XSteam('psat_T',Ts);
    v = XSteam('vV_T',Ts);
    p = toSIunit_p(p);
    T = toSIunit_T(Ts);
    v = toSIunit_v(v);
    rho = 1 / v;
    Out = fromSIunit_tc(tc_ptrho(p, T, rho));

case 'tc_pt'
    Ts = In2;
    ps = In1;
    v = XSteam('v_pT',ps, Ts);
    p = toSIunit_p(ps);
    T = toSIunit_T(Ts);
    v = toSIunit_v(v);
    rho = 1 / v;
    Out = fromSIunit_tc(tc_ptrho(p, T, rho));

case 'tc_ph'
    hs = In2;
    ps = In1;
    v = XSteam('v_ph',ps, hs);
    T = XSteam('T_ph',ps, hs);
    p = toSIunit_p(ps);
    T = toSIunit_T(T);
    v = toSIunit_v(v);
    rho = 1 / v;
    Out = fromSIunit_tc(tc_ptrho(p, T, rho));

case 'tc_hs'
    hs = In1;
    p = XSteam('p_hs',hs, In2);
    ps = p;
    v = XSteam('v_ph',ps, hs);
    T = XSteam('T_ph',ps, hs);
    p = toSIunit_p(p);
    T = toSIunit_T(T);
    v = toSIunit_v(v);
    rho = 1 / v;
    Out = fromSIunit_tc(tc_ptrho(p, T, rho));

%*****
%*****
%*1.17 Vapour fraction
case 'x_ph'
    p = toSIunit_p(In1);
    h = toSIunit_h(In2);
    if p > 0.000611657 & p < 22.06395
        Out = fromSIunit_x(x4_ph(p, h));
    else
        Out = NaN;
    end

case 'x_ps'
    p = toSIunit_p(In1);
    s = toSIunit_s(In2);
    if p > 0.000611657 & p < 22.06395
        Out = fromSIunit_x(x4_ps(p, s));
    else
        Out = NaN;
    end
end

```

```

%*****
%*****
%*1.18 Vapour Volume Fraction
case 'vx_ph'
    p = toSIunit_p(In1);
    h = toSIunit_h(In2);
    if p > 0.000611657 & p < 22.06395
        if p < 16.529
            vL = v1_pT(p, T4_p(p));
            vV = v2_pT(p, T4_p(p));
        else
            vL = v3_ph(p, h4L_p(p));
            vV = v3_ph(p, h4V_p(p));
        end
        xs = x4_ph(p, h);
        Out = fromSIunit_vx((xs * vV / (xs * vV + (1 - xs) * vL)));
    else
        Out = NaN;
    end

case 'vx_ps'
    p = toSIunit_p(In1);
    s = toSIunit_s(In2);
    if p > 0.000611657 & p < 22.06395
        if p < 16.529
            vL = v1_pT(p, T4_p(p));
            vV = v2_pT(p, T4_p(p));
        else
            vL = v3_ph(p, h4L_p(p));
            vV = v3_ph(p, h4V_p(p));
        end
        xs = x4_ps(p, s);
        Out = fromSIunit_vx((xs * vV / (xs * vV + (1 - xs) * vL)));
    else
        Out = NaN;
    end

case 'check'
    err=check;

otherwise
    error(['Unknown calling function to XSteam, ',fun, ' See help
XSteam for valid calling functions']);
end
%*****
%*****
%*2 IAPWS IF 97 Calling functions
*
%*****
%*****
%
%*****
%*****
%*2.1 Functions for region 1
function v1_pT = v1_pT(p, T)
%Release on the IAPWS Industrial formulation 1997 for the
Thermodynamic Properties of Water and Steam, September 1997
%5 Equations for Region 1, Section. 5.1 Basic Equation
%Equation 7, Table 3, Page 6

```

```

I1 = [0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3,
3, 4, 4, 4, 5, 8, 8, 21, 23, 29, 30, 31, 32];
J1 = [-2, -1, 0, 1, 2, 3, 4, 5, -9, -7, -1, 0, 1, 3, -3, 0, 1, 3, 17,
-4, 0, 6, -5, -2, 10, -8, -11, -6, -29, -31, -38, -39, -40, -41];
n1 = [0.14632971213167, -0.84548187169114, -3.756360367204,
3.3855169168385, -0.95791963387872, 0.15772038513228, -
0.016616417199501, 8.1214629983568E-04, 2.8319080123804E-04, -
6.0706301565874E-04, -0.018990068218419, -0.032529748770505, -
0.021841717175414, -5.283835796993E-05, -4.7184321073267E-04, -
3.0001780793026E-04, 4.7661393906987E-05, -4.4141845330846E-06, -
7.2694996297594E-16, -3.1679644845054E-05, -2.8270797985312E-06, -
8.5205128120103E-10, -2.2425281908E-06, -6.5171222895601E-07, -
1.4341729937924E-13, -4.0516996860117E-07, -1.2734301741641E-09, -
1.7424871230634E-10, -6.8762131295531E-19, 1.4478307828521E-20,
2.6335781662795E-23, -1.1947622640071E-23, 1.8228094581404E-24, -
9.3537087292458E-26];
R = 0.461526; %kJ/(kg K)
Pi = p / 16.53;
tau = 1386 / T;
gamma_der_pi = 0;
for i = 1 : 34
    gamma_der_pi = gamma_der_pi - n1(i) * I1(i) * (7.1 - Pi) ^ (I1(i)
- 1) * (tau - 1.222) ^ J1(i);
end
v1_pT = R * T / p * Pi * gamma_der_pi / 1000;

function h1_pT = h1_pT(p, T)
%Release on the IAPWS Industrial formulation 1997 for the
Thermodynamic Properties of Water and Steam, September 1997
%5 Equations for Region 1, Section. 5.1 Basic Equation
%Equation 7, Table 3, Page 6
I1 = [0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3,
3, 4, 4, 4, 5, 8, 8, 21, 23, 29, 30, 31, 32];
J1 = [-2, -1, 0, 1, 2, 3, 4, 5, -9, -7, -1, 0, 1, 3, -3, 0, 1, 3, 17,
-4, 0, 6, -5, -2, 10, -8, -11, -6, -29, -31, -38, -39, -40, -41];
n1 = [0.14632971213167, -0.84548187169114, -3.756360367204,
3.3855169168385, -0.95791963387872, 0.15772038513228, -
0.016616417199501, 8.1214629983568E-04, 2.8319080123804E-04, -
6.0706301565874E-04, -0.018990068218419, -0.032529748770505, -
0.021841717175414, -5.283835796993E-05, -4.7184321073267E-04, -
3.0001780793026E-04, 4.7661393906987E-05, -4.4141845330846E-06, -
7.2694996297594E-16, -3.1679644845054E-05, -2.8270797985312E-06, -
8.5205128120103E-10, -2.2425281908E-06, -6.5171222895601E-07, -
1.4341729937924E-13, -4.0516996860117E-07, -1.2734301741641E-09, -
1.7424871230634E-10, -6.8762131295531E-19, 1.4478307828521E-20,
2.6335781662795E-23, -1.1947622640071E-23, 1.8228094581404E-24, -
9.3537087292458E-26];
R = 0.461526; %kJ/(kg K)
Pi = p / 16.53;
tau = 1386 / T;
gamma_der_tau = 0;
for i = 1 : 34
    gamma_der_tau = gamma_der_tau + (n1(i) * (7.1 - Pi) ^ I1(i) *
J1(i) * (tau - 1.222) ^ (J1(i) - 1));
end
h1_pT = R * T * tau * gamma_der_tau;

function u1_pT = u1_pT(p, T)
%Release on the IAPWS Industrial formulation 1997 for the
Thermodynamic Properties of Water and Steam, September 1997
%5 Equations for Region 1, Section. 5.1 Basic Equation

```



```

%Equation 7, Table 3, Page 6
I1 = [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3,
3, 4, 4, 4, 5, 8, 8, 21, 23, 29, 30, 31, 32];
J1 = [-2, -1, 0, 1, 2, 3, 4, 5, -9, -7, -1, 0, 1, 3, -3, 0, 1, 3, 17,
-4, 0, 6, -5, -2, 10, -8, -11, -6, -29, -31, -38, -39, -40, -41];
n1 = [0.14632971213167, -0.84548187169114, -3.756360367204,
3.3855169168385, -0.95791963387872, 0.15772038513228, -
0.016616417199501, 8.1214629983568E-04, 2.8319080123804E-04, -
6.0706301565874E-04, -0.018990068218419, -0.032529748770505, -
0.021841717175414, -5.283835796993E-05, -4.7184321073267E-04, -
3.0001780793026E-04, 4.7661393906987E-05, -4.4141845330846E-06, -
7.2694996297594E-16, -3.1679644845054E-05, -2.8270797985312E-06, -
8.5205128120103E-10, -2.2425281908E-06, -6.5171222895601E-07, -
1.4341729937924E-13, -4.0516996860117E-07, -1.2734301741641E-09, -
1.7424871230634E-10, -6.8762131295531E-19, 1.4478307828521E-20,
2.6335781662795E-23, -1.1947622640071E-23, 1.8228094581404E-24, -
9.3537087292458E-26];
R = 0.461526; %kJ/(kg K)
Pi = p / 16.53;
tau = 1386 / T;
gamma_der_tau = 0;
gamma_der_pi = 0;
for i = 1 : 34
    gamma_der_pi = gamma_der_pi - n1(i) * I1(i) * (7.1 - Pi) ^ (I1(i)
- 1) * (tau - 1.222) ^ J1(i);
    gamma_der_tau = gamma_der_tau + (n1(i) * (7.1 - Pi) ^ I1(i) *
J1(i) * (tau - 1.222) ^ (J1(i) - 1));
end
ul_pT = R * T * (tau * gamma_der_tau - Pi * gamma_der_pi);

function s1_pT = s1_pT(p, T)
%Release on the IAPWS Industrial formulation 1997 for the
Thermodynamic Properties of Water and Steam, September 1997
%5 Equations for Region 1, Section. 5.1 Basic Equation
%Equation 7, Table 3, Page 6
I1 = [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3,
3, 4, 4, 4, 5, 8, 8, 21, 23, 29, 30, 31, 32];
J1 = [-2, -1, 0, 1, 2, 3, 4, 5, -9, -7, -1, 0, 1, 3, -3, 0, 1, 3, 17,
-4, 0, 6, -5, -2, 10, -8, -11, -6, -29, -31, -38, -39, -40, -41];
n1 = [0.14632971213167, -0.84548187169114, -3.756360367204,
3.3855169168385, -0.95791963387872, 0.15772038513228, -
0.016616417199501, 8.1214629983568E-04, 2.8319080123804E-04, -
6.0706301565874E-04, -0.018990068218419, -0.032529748770505, -
0.021841717175414, -5.283835796993E-05, -4.7184321073267E-04, -
3.0001780793026E-04, 4.7661393906987E-05, -4.4141845330846E-06, -
7.2694996297594E-16, -3.1679644845054E-05, -2.8270797985312E-06, -
8.5205128120103E-10, -2.2425281908E-06, -6.5171222895601E-07, -
1.4341729937924E-13, -4.0516996860117E-07, -1.2734301741641E-09, -
1.7424871230634E-10, -6.8762131295531E-19, 1.4478307828521E-20,
2.6335781662795E-23, -1.1947622640071E-23, 1.8228094581404E-24, -
9.3537087292458E-26];
R = 0.461526; %kJ/(kg K)
Pi = p / 16.53;
tau = 1386 / T;
gamma = 0;
gamma_der_tau = 0;
for i = 1 : 34
    gamma_der_tau = gamma_der_tau + (n1(i) * (7.1 - Pi) ^ I1(i) *
J1(i) * (tau - 1.222) ^ (J1(i) - 1));
    gamma = gamma + n1(i) * (7.1 - Pi) ^ I1(i) * (tau - 1.222) ^
J1(i);
end

```

```

end
s1_pT = R * tau * gamma_der_tau - R * gamma;

function Cp1_pT = Cp1_pT(p, T)
%Release on the IAPWS Industrial formulation 1997 for the
Thermodynamic Properties of Water and Steam, September 1997
%5 Equations for Region 1, Section. 5.1 Basic Equation
%Equation 7, Table 3, Page 6
I1 = [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3,
3, 4, 4, 4, 5, 8, 8, 21, 23, 29, 30, 31, 32];
J1 = [-2, -1, 0, 1, 2, 3, 4, 5, -9, -7, -1, 0, 1, 3, -3, 0, 1, 3, 17,
-4, 0, 6, -5, -2, 10, -8, -11, -6, -29, -31, -38, -39, -40, -41];
n1 = [0.14632971213167, -0.84548187169114, -3.756360367204,
3.3855169168385, -0.95791963387872, 0.15772038513228, -
0.016616417199501, 8.1214629983568E-04, 2.8319080123804E-04, -
6.0706301565874E-04, -0.018990068218419, -0.032529748770505, -
0.021841717175414, -5.283835796993E-05, -4.7184321073267E-04, -
3.0001780793026E-04, 4.7661393906987E-05, -4.4141845330846E-06, -
7.2694996297594E-16, -3.1679644845054E-05, -2.8270797985312E-06, -
8.5205128120103E-10, -2.2425281908E-06, -6.5171222895601E-07, -
1.4341729937924E-13, -4.0516996860117E-07, -1.2734301741641E-09, -
1.7424871230634E-10, -6.8762131295531E-19, 1.4478307828521E-20,
2.6335781662795E-23, -1.1947622640071E-23, 1.8228094581404E-24, -
9.3537087292458E-26];
R = 0.461526; %kJ/(kg K)
Pi = p / 16.53;
tau = 1386 / T;
gamma_der_tautau = 0;
for i = 1:34
    gamma_der_tautau = gamma_der_tautau + (n1(i) * (7.1 - Pi) ^ I1(i)
* J1(i) * (J1(i) - 1) * (tau - 1.222) ^ (J1(i) - 2));
end
Cp1_pT = -R * tau ^ 2 * gamma_der_tautau;

function Cv1_pT = Cv1_pT(p, T)
%Release on the IAPWS Industrial formulation 1997 for the
Thermodynamic Properties of Water and Steam, September 1997
%5 Equations for Region 1, Section. 5.1 Basic Equation
%Equation 7, Table 3, Page 6
I1 = [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3,
3, 4, 4, 4, 5, 8, 8, 21, 23, 29, 30, 31, 32];
J1 = [-2, -1, 0, 1, 2, 3, 4, 5, -9, -7, -1, 0, 1, 3, -3, 0, 1, 3, 17,
-4, 0, 6, -5, -2, 10, -8, -11, -6, -29, -31, -38, -39, -40, -41];
n1 = [0.14632971213167, -0.84548187169114, -3.756360367204,
3.3855169168385, -0.95791963387872, 0.15772038513228, -
0.016616417199501, 8.1214629983568E-04, 2.8319080123804E-04, -
6.0706301565874E-04, -0.018990068218419, -0.032529748770505, -
0.021841717175414, -5.283835796993E-05, -4.7184321073267E-04, -
3.0001780793026E-04, 4.7661393906987E-05, -4.4141845330846E-06, -
7.2694996297594E-16, -3.1679644845054E-05, -2.8270797985312E-06, -
8.5205128120103E-10, -2.2425281908E-06, -6.5171222895601E-07, -
1.4341729937924E-13, -4.0516996860117E-07, -1.2734301741641E-09, -
1.7424871230634E-10, -6.8762131295531E-19, 1.4478307828521E-20,
2.6335781662795E-23, -1.1947622640071E-23, 1.8228094581404E-24, -
9.3537087292458E-26];
R = 0.461526; %kJ/(kg K)
Pi = p / 16.53;
tau = 1386 / T;
gamma_der_pi = 0;
gamma_der_pipi = 0;
gamma_der_pitau = 0;

```

```

gamma_der_tautau = 0;
for i = 1 : 34
    gamma_der_pi = gamma_der_pi - n1(i) * I1(i) * (7.1 - Pi) ^ (I1(i)
- 1) * (tau - 1.222) ^ J1(i);
    gamma_der_pipi = gamma_der_pipi + n1(i) * I1(i) * (I1(i) - 1) *
(7.1 - Pi) ^ (I1(i) - 2) * (tau - 1.222) ^ J1(i);
    gamma_der_pitau = gamma_der_pitau - n1(i) * I1(i) * (7.1 - Pi) ^
(I1(i) - 1) * J1(i) * (tau - 1.222) ^ (J1(i) - 1);
    gamma_der_tautau = gamma_der_tautau + n1(i) * (7.1 - Pi) ^ I1(i) *
J1(i) * (J1(i) - 1) * (tau - 1.222) ^ (J1(i) - 2);
end
Cv1_pT = R * (-tau ^ 2 * gamma_der_tautau + (gamma_der_pi - tau *
gamma_der_pitau) ^ 2 / gamma_der_pipi);

function w1_pT = w1_pT(p, T)
%Release on the IAPWS Industrial formulation 1997 for the
Thermodynamic Properties of Water and Steam, September 1997
%5 Equations for Region 1, Section. 5.1 Basic Equation
%Equation 7, Table 3, Page 6
I1 = [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3,
3, 4, 4, 4, 5, 8, 8, 21, 23, 29, 30, 31, 32];
J1 = [-2, -1, 0, 1, 2, 3, 4, 5, -9, -7, -1, 0, 1, 3, -3, 0, 1, 3, 17,
-4, 0, 6, -5, -2, 10, -8, -11, -6, -29, -31, -38, -39, -40, -41];
n1 = [0.14632971213167, -0.84548187169114, -3.756360367204,
3.3855169168385, -0.95791963387872, 0.15772038513228, -
0.016616417199501, 8.1214629983568E-04, 2.8319080123804E-04, -
6.0706301565874E-04, -0.018990068218419, -0.032529748770505, -
0.021841717175414, -5.283835796993E-05, -4.7184321073267E-04, -
3.0001780793026E-04, 4.7661393906987E-05, -4.4141845330846E-06, -
7.2694996297594E-16, -3.1679644845054E-05, -2.8270797985312E-06, -
8.5205128120103E-10, -2.2425281908E-06, -6.5171222895601E-07, -
1.4341729937924E-13, -4.0516996860117E-07, -1.2734301741641E-09, -
1.7424871230634E-10, -6.8762131295531E-19, 1.4478307828521E-20,
2.6335781662795E-23, -1.1947622640071E-23, 1.8228094581404E-24, -
9.3537087292458E-26];
R = 0.461526; %kJ/(kg K)
Pi = p / 16.53;
tau = 1386 / T;
gamma_der_pi = 0;
gamma_der_pipi = 0;
gamma_der_pitau = 0;
gamma_der_tautau = 0;
for i = 1 : 34
    gamma_der_pi = gamma_der_pi - n1(i) * I1(i) * (7.1 - Pi) ^ (I1(i)
- 1) * (tau - 1.222) ^ J1(i);
    gamma_der_pipi = gamma_der_pipi + n1(i) * I1(i) * (I1(i) - 1) *
(7.1 - Pi) ^ (I1(i) - 2) * (tau - 1.222) ^ J1(i);
    gamma_der_pitau = gamma_der_pitau - n1(i) * I1(i) * (7.1 - Pi) ^
(I1(i) - 1) * J1(i) * (tau - 1.222) ^ (J1(i) - 1);
    gamma_der_tautau = gamma_der_tautau + n1(i) * (7.1 - Pi) ^ I1(i) *
J1(i) * (J1(i) - 1) * (tau - 1.222) ^ (J1(i) - 2);
end
w1_pT = (1000 * R * T * gamma_der_pi ^ 2 / ((gamma_der_pi - tau *
gamma_der_pitau) ^ 2 / (tau ^ 2 * gamma_der_tautau) - gamma_der_pipi))
^ 0.5;

function T1_ph = T1_ph(p, h)
%Release on the IAPWS Industrial formulation 1997 for the
Thermodynamic Properties of Water and Steam, September 1997
%5 Equations for Region 1, Section. 5.1 Basic Equation, 5.2.1 The
Backward Equation T ( p,h )

```

```

%Equation 11, Table 6, Page 10
I1 = [0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 2, 2, 3, 3, 4, 5, 6];
J1 = [0, 1, 2, 6, 22, 32, 0, 1, 2, 3, 4, 10, 32, 10, 32, 10, 32, 32,
32, 32];
n1 = [-238.72489924521, 404.21188637945, 113.49746881718, -
5.8457616048039, -1.528548241314E-04, -1.0866707695377E-06, -
13.391744872602, 43.211039183559, -54.010067170506, 30.535892203916, -
6.5964749423638, 9.3965400878363E-03, 1.157364750534E-07, -
2.5858641282073E-05, -4.0644363084799E-09, 6.6456186191635E-08,
8.0670734103027E-11, -9.3477771213947E-13, 5.8265442020601E-15, -
1.5020185953503E-17];
Pi = p / 1;
eta = h / 2500;
T = 0;
for i = 1 : 20
    T = T + n1(i) * Pi ^ I1(i) * (eta + 1) ^ J1(i);
end
T1_ph = T;

function T1_ps = T1_ps(p, s)
%Release on the IAPWS Industrial formulation 1997 for the
Thermodynamic Properties of Water and Steam, September 1997
%5 Equations for Region 1, Section. 5.1 Basic Equation, 5.2.2 The
Backward Equation T ( p, s )
%Equation 13, Table 8, Page 11
I1 = [0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 4];
J1 = [0, 1, 2, 3, 11, 31, 0, 1, 2, 3, 12, 31, 0, 1, 2, 9, 31, 10, 32,
32];
n1 = [174.78268058307, 34.806930892873, 6.5292584978455,
0.33039981775489, -1.9281382923196E-07, -2.4909197244573E-23, -
0.26107636489332, 0.22592965981586, -0.064256463395226,
7.8876289270526E-03, 3.5672110607366E-10, 1.7332496994895E-24,
5.6608900654837E-04, -3.2635483139717E-04, 4.4778286690632E-05, -
5.1322156908507E-10, -4.2522657042207E-26, 2.6400441360689E-13,
7.8124600459723E-29, -3.0732199903668E-31];
Pi = p / 1;
Sigma = s / 1;
T = 0;
for i = 1 : 20
    T = T + n1(i) * Pi ^ I1(i) * (Sigma + 2) ^ J1(i);
end
T1_ps = T;

function p1_hs = p1_hs(h, s)
%Supplementary Release on Backward Equations for Pressure as a
Function of Enthalpy and Entropy p(h,s) to the IAPWS Industrial
formulation 1997 for the Thermodynamic Properties of Water and Steam
%5 Backward Equation p(h,s) for Region 1
%Equation 1, Table 2, Page 5
I1 = [0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 2, 2, 2, 3, 4, 4, 5];
J1 = [0, 1, 2, 4, 5, 6, 8, 14, 0, 1, 4, 6, 0, 1, 10, 4, 1, 4, 0];
n1 = [-0.691997014660582, -18.361254878756, -9.28332409297335,
65.9639569909906, -16.2060388912024, 450.620017338667,
854.68067822417, 6075.23214001162, 32.6487682621856, -
26.9408844582931, -319.9478483343, -928.35430704332, 30.3634537455249,
-65.0540422444146, -4309.9131651613, -747.512324096068,
730.000345529245, 1142.84032569021, -436.407041874559];
eta = h / 3400;
Sigma = s / 7.6;
p = 0;
for i = 1 : 19

```

```

    p = p + n1(i) * (eta + 0.05) ^ I1(i) * (Sigma + 0.05) ^ J1(i);
end
p1_hs = p * 100;

function T1_prho = T1_prho(p , rho)
    %Solve by iteration. Observe that for low temperatures this equation
    has 2 solutions.
    %Solve with half interval method
    Low_Bound = 273.15;
    High_Bound = T4_p(p);
    rhos=-1000;
    while abs(rho - rhos) > 0.00001
        Ts = (Low_Bound + High_Bound) / 2;
        rhos = 1 / v1_pT(p, Ts);
        if rhos < rho
            High_Bound = Ts;
        else
            Low_Bound = Ts;
        end
    end
    T1_prho = Ts;

%*****
%*****
%*2.2 functions for region 2

function v2_pT = v2_pT(p, T)
%Release on the IAPWS Industrial formulation 1997 for the
Thermodynamic Properties of Water and Steam, September 1997
%6 Equations for Region 2, Section. 6.1 Basic Equation
%Table 11 and 12, Page 14 and 15
J0 = [0, 1, -5, -4, -3, -2, -1, 2, 3];
n0 = [-9.6927686500217, 10.086655968018, -0.005608791128302,
0.071452738081455, -0.40710498223928, 1.4240819171444, -
4.383951131945, -0.28408632460772, 0.021268463753307];
Ir = [1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 4, 4, 4, 5, 6, 6,
6, 7, 7, 7, 8, 8, 9, 10, 10, 10, 10, 16, 16, 18, 20, 20, 20, 21, 22, 23,
24, 24, 24];
Jr = [0, 1, 2, 3, 6, 1, 2, 4, 7, 36, 0, 1, 3, 6, 35, 1, 2, 3, 7, 3,
16, 35, 0, 11, 25, 8, 36, 13, 4, 10, 14, 29, 50, 57, 20, 35, 48, 21,
53, 39, 26, 40, 58];
nr = [-1.7731742473213E-03, -0.017834862292358, -0.045996013696365, -
0.057581259083432, -0.05032527872793, -3.3032641670203E-05, -
1.8948987516315E-04, -3.9392777243355E-03, -0.043797295650573, -
2.6674547914087E-05, 2.0481737692309E-08, 4.3870667284435E-07, -
3.227767723857E-05, -1.5033924542148E-03, -0.040668253562649, -
7.8847309559367E-10, 1.2790717852285E-08, 4.8225372718507E-07,
2.2922076337661E-06, -1.6714766451061E-11, -2.1171472321355E-03, -
23.895741934104, -5.905956432427E-18, -1.2621808899101E-06, -
0.038946842435739, 1.1256211360459E-11, -8.2311340897998,
1.9809712802088E-08, 1.0406965210174E-19, -1.0234747095929E-13, -
1.0018179379511E-09, -8.0882908646985E-11, 0.10693031879409, -
0.33662250574171, 8.9185845355421E-25, 3.0629316876232E-13, -
4.2002467698208E-06, -5.9056029685639E-26, 3.7826947613457E-06, -
1.2768608934681E-15, 7.3087610595061E-29, 5.5414715350778E-17, -
9.436970724121E-07];
R = 0.461526; %kJ/(kg K)
Pi = p;
tau = 540 / T;
g0_pi = 1 / Pi;
gr_pi = 0;

```

```

for i = 1 : 43
    gr_pi = gr_pi + nr(i) * Ir(i) * Pi ^ (Ir(i) - 1) * (tau - 0.5) ^
Jr(i);
end
v2_pT = R * T / p * Pi * (g0_pi + gr_pi) / 1000;

function h2_pT = h2_pT(p, T)
%Release on the IAPWS Industrial formulation 1997 for the
Thermodynamic Properties of Water and Steam, September 1997
%6 Equations for Region 2, Section. 6.1 Basic Equation
%Table 11 and 12, Page 14 and 15
J0 = [0, 1, -5, -4, -3, -2, -1, 2, 3];
n0 = [-9.6927686500217, 10.086655968018, -0.005608791128302,
0.071452738081455, -0.40710498223928, 1.4240819171444, -
4.383951131945, -0.28408632460772, 0.021268463753307];
Ir = [1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 4, 4, 4, 5, 6, 6,
6, 7, 7, 7, 8, 8, 9, 10, 10, 10, 16, 16, 18, 20, 20, 20, 21, 22, 23,
24, 24, 24];
Jr = [0, 1, 2, 3, 6, 1, 2, 4, 7, 36, 0, 1, 3, 6, 35, 1, 2, 3, 7, 3,
16, 35, 0, 11, 25, 8, 36, 13, 4, 10, 14, 29, 50, 57, 20, 35, 48, 21,
53, 39, 26, 40, 58];
nr = [-1.7731742473213E-03, -0.017834862292358, -0.045996013696365, -
0.057581259083432, -0.05032527872793, -3.3032641670203E-05, -
1.8948987516315E-04, -3.9392777243355E-03, -0.043797295650573, -
2.6674547914087E-05, 2.0481737692309E-08, 4.3870667284435E-07, -
3.227767723857E-05, -1.5033924542148E-03, -0.040668253562649, -
7.8847309559367E-10, 1.2790717852285E-08, 4.8225372718507E-07,
2.2922076337661E-06, -1.6714766451061E-11, -2.1171472321355E-03, -
23.895741934104, -5.905956432427E-18, -1.2621808899101E-06, -
0.038946842435739, 1.1256211360459E-11, -8.2311340897998,
1.9809712802088E-08, 1.0406965210174E-19, -1.0234747095929E-13, -
1.0018179379511E-09, -8.0882908646985E-11, 0.10693031879409, -
0.33662250574171, 8.9185845355421E-25, 3.0629316876232E-13, -
4.2002467698208E-06, -5.9056029685639E-26, 3.7826947613457E-06, -
1.2768608934681E-15, 7.3087610595061E-29, 5.5414715350778E-17, -
9.436970724121E-07];
R = 0.461526; %kJ/(kg K)
Pi = p;
tau = 540 / T;
g0_tau = 0;
for i = 1 : 9
    g0_tau = g0_tau + n0(i) * J0(i) * tau ^ (J0(i) - 1);
end
gr_tau = 0;
for i = 1 : 43
    gr_tau = gr_tau + nr(i) * Pi ^ Ir(i) * Jr(i) * (tau - 0.5) ^
(Jr(i) - 1);
end
h2_pT = R * T * tau * (g0_tau + gr_tau);

function u2_pT = u2_pT(p, T)
%Release on the IAPWS Industrial formulation 1997 for the
Thermodynamic Properties of Water and Steam, September 1997
%6 Equations for Region 2, Section. 6.1 Basic Equation
%Table 11 and 12, Page 14 and 15
J0 = [0, 1, -5, -4, -3, -2, -1, 2, 3];
n0 = [-9.6927686500217, 10.086655968018, -0.005608791128302,
0.071452738081455, -0.40710498223928, 1.4240819171444, -
4.383951131945, -0.28408632460772, 0.021268463753307];

```

```

Ir = [1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 4, 4, 4, 5, 6, 6,
6, 7, 7, 7, 8, 8, 9, 10, 10, 10, 16, 16, 18, 20, 20, 20, 21, 22, 23,
24, 24, 24];
Jr = [0, 1, 2, 3, 6, 1, 2, 4, 7, 36, 0, 1, 3, 6, 35, 1, 2, 3, 7, 3,
16, 35, 0, 11, 25, 8, 36, 13, 4, 10, 14, 29, 50, 57, 20, 35, 48, 21,
53, 39, 26, 40, 58];
nr = [-1.7731742473213E-03, -0.017834862292358, -0.045996013696365, -
0.057581259083432, -0.05032527872793, -3.3032641670203E-05, -
1.8948987516315E-04, -3.9392777243355E-03, -0.043797295650573, -
2.6674547914087E-05, 2.0481737692309E-08, 4.3870667284435E-07, -
3.227767723857E-05, -1.5033924542148E-03, -0.040668253562649, -
7.8847309559367E-10, 1.2790717852285E-08, 4.8225372718507E-07,
2.2922076337661E-06, -1.6714766451061E-11, -2.1171472321355E-03, -
23.895741934104, -5.905956432427E-18, -1.2621808899101E-06, -
0.038946842435739, 1.1256211360459E-11, -8.2311340897998,
1.9809712802088E-08, 1.0406965210174E-19, -1.0234747095929E-13, -
1.0018179379511E-09, -8.0882908646985E-11, 0.10693031879409, -
0.33662250574171, 8.9185845355421E-25, 3.0629316876232E-13, -
4.2002467698208E-06, -5.9056029685639E-26, 3.7826947613457E-06, -
1.2768608934681E-15, 7.3087610595061E-29, 5.5414715350778E-17, -
9.436970724121E-07];
R = 0.461526; %kJ/(kg K)
Pi = p;
tau = 540 / T;
g0_pi = 1 / Pi;
g0_tau = 0;
for i = 1 : 9
    g0_tau = g0_tau + n0(i) * J0(i) * tau ^ (J0(i) - 1);
end
gr_pi = 0;
gr_tau = 0;
for i = 1 : 43
    gr_pi = gr_pi + nr(i) * Ir(i) * Pi ^ (Ir(i) - 1) * (tau - 0.5) ^
Jr(i);
    gr_tau = gr_tau + nr(i) * Pi ^ Ir(i) * Jr(i) * (tau - 0.5) ^
(Jr(i) - 1);
end
u2_pT = R * T * (tau * (g0_tau + gr_tau) - Pi * (g0_pi + gr_pi));

function s2_pT = s2_pT(p, T)
%Release on the IAPWS Industrial formulation 1997 for the
Thermodynamic Properties of Water and Steam, September 1997
%6 Equations for Region 2, Section. 6.1 Basic Equation
%Table 11 and 12, Page 14 and 15
J0 = [0, 1, -5, -4, -3, -2, -1, 2, 3];
n0 = [-9.6927686500217, 10.086655968018, -0.005608791128302,
0.071452738081455, -0.40710498223928, 1.4240819171444, -
4.383951131945, -0.28408632460772, 0.021268463753307];
Ir = [1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 4, 4, 4, 5, 6, 6,
6, 7, 7, 7, 8, 8, 9, 10, 10, 10, 16, 16, 18, 20, 20, 20, 21, 22, 23,
24, 24, 24];
Jr = [0, 1, 2, 3, 6, 1, 2, 4, 7, 36, 0, 1, 3, 6, 35, 1, 2, 3, 7, 3,
16, 35, 0, 11, 25, 8, 36, 13, 4, 10, 14, 29, 50, 57, 20, 35, 48, 21,
53, 39, 26, 40, 58];
nr = [-1.7731742473213E-03, -0.017834862292358, -0.045996013696365, -
0.057581259083432, -0.05032527872793, -3.3032641670203E-05, -
1.8948987516315E-04, -3.9392777243355E-03, -0.043797295650573, -
2.6674547914087E-05, 2.0481737692309E-08, 4.3870667284435E-07, -
3.227767723857E-05, -1.5033924542148E-03, -0.040668253562649, -
7.8847309559367E-10, 1.2790717852285E-08, 4.8225372718507E-07,

```

```

2.2922076337661E-06, -1.6714766451061E-11, -2.1171472321355E-03, -
23.895741934104, -5.905956432427E-18, -1.2621808899101E-06, -
0.038946842435739, 1.1256211360459E-11, -8.2311340897998,
1.9809712802088E-08, 1.0406965210174E-19, -1.0234747095929E-13, -
1.0018179379511E-09, -8.0882908646985E-11, 0.10693031879409, -
0.33662250574171, 8.9185845355421E-25, 3.0629316876232E-13, -
4.2002467698208E-06, -5.9056029685639E-26, 3.7826947613457E-06, -
1.2768608934681E-15, 7.3087610595061E-29, 5.5414715350778E-17, -
9.436970724121E-07];
R = 0.461526; %kJ/(kg K)
Pi = p;
tau = 540 / T;
g0 = log(Pi);
g0_tau = 0;
for i = 1 : 9
    g0 = g0 + n0(i) * tau ^ J0(i);
    g0_tau = g0_tau + n0(i) * J0(i) * tau ^ (J0(i) - 1);
end
gr = 0;
gr_tau = 0;
for i = 1 : 43
    gr = gr + nr(i) * Pi ^ Ir(i) * (tau - 0.5) ^ Jr(i);
    gr_tau = gr_tau + nr(i) * Pi ^ Ir(i) * Jr(i) * (tau - 0.5) ^
(Jr(i) - 1);
end
s2_pT = R * (tau * (g0_tau + gr_tau) - (g0 + gr));

function Cp2_pT = Cp2_pT(p, T)
%Release on the IAPWS Industrial formulation 1997 for the
%Thermodynamic Properties of Water and Steam, September 1997
%6 Equations for Region 2, Section. 6.1 Basic Equation
%Table 11 and 12, Page 14 and 15
J0 = [0, 1, -5, -4, -3, -2, -1, 2, 3];
n0 = [-9.6927686500217, 10.086655968018, -0.005608791128302,
0.071452738081455, -0.40710498223928, 1.4240819171444, -
4.383951131945, -0.28408632460772, 0.021268463753307];
Ir = [1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 4, 4, 4, 5, 6, 6,
6, 7, 7, 7, 8, 8, 9, 10, 10, 10, 16, 16, 18, 20, 20, 20, 21, 22, 23,
24, 24, 24];
Jr = [0, 1, 2, 3, 6, 1, 2, 4, 7, 36, 0, 1, 3, 6, 35, 1, 2, 3, 7, 3,
16, 35, 0, 11, 25, 8, 36, 13, 4, 10, 14, 29, 50, 57, 20, 35, 48, 21,
53, 39, 26, 40, 58];
nr = [-1.7731742473213E-03, -0.017834862292358, -0.045996013696365, -
0.057581259083432, -0.05032527872793, -3.3032641670203E-05, -
1.8948987516315E-04, -3.9392777243355E-03, -0.043797295650573, -
2.6674547914087E-05, 2.0481737692309E-08, 4.3870667284435E-07, -
3.227767723857E-05, -1.5033924542148E-03, -0.040668253562649, -
7.8847309559367E-10, 1.2790717852285E-08, 4.8225372718507E-07,
2.2922076337661E-06, -1.6714766451061E-11, -2.1171472321355E-03, -
23.895741934104, -5.905956432427E-18, -1.2621808899101E-06, -
0.038946842435739, 1.1256211360459E-11, -8.2311340897998,
1.9809712802088E-08, 1.0406965210174E-19, -1.0234747095929E-13, -
1.0018179379511E-09, -8.0882908646985E-11, 0.10693031879409, -
0.33662250574171, 8.9185845355421E-25, 3.0629316876232E-13, -
4.2002467698208E-06, -5.9056029685639E-26, 3.7826947613457E-06, -
1.2768608934681E-15, 7.3087610595061E-29, 5.5414715350778E-17, -
9.436970724121E-07];
R = 0.461526; %kJ/(kg K)
Pi = p;
tau = 540 / T;
g0_tautau = 0;

```



```

for i = 1 : 9
    g0_tautau = g0_tautau + n0(i) * J0(i) * (J0(i) - 1) * tau ^ (J0(i)
- 2);
end
gr_tautau = 0;
for i = 1 : 43
    gr_tautau = gr_tautau + nr(i) * Pi ^ Ir(i) * Jr(i) * (Jr(i) - 1) *
(tau - 0.5) ^ (Jr(i) - 2);
end
Cp2_pT = -R * tau ^ 2 * (g0_tautau + gr_tautau);

function Cv2_pT = Cv2_pT(p, T)
%Release on the IAPWS Industrial formulation 1997 for the
Thermodynamic Properties of Water and Steam, September 1997
%6 Equations for Region 2, Section. 6.1 Basic Equation
%Table 11 and 12, Page 14 and 15
J0 = [0, 1, -5, -4, -3, -2, -1, 2, 3];
n0 = [-9.6927686500217, 10.086655968018, -0.005608791128302,
0.071452738081455, -0.40710498223928, 1.4240819171444, -
4.383951131945, -0.28408632460772, 0.021268463753307];
Ir = [1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 4, 4, 4, 5, 6, 6,
6, 7, 7, 7, 8, 8, 9, 10, 10, 10, 16, 16, 18, 20, 20, 20, 21, 22, 23,
24, 24, 24];
Jr = [0, 1, 2, 3, 6, 1, 2, 4, 7, 36, 0, 1, 3, 6, 35, 1, 2, 3, 7, 3,
16, 35, 0, 11, 25, 8, 36, 13, 4, 10, 14, 29, 50, 57, 20, 35, 48, 21,
53, 39, 26, 40, 58];
nr = [-1.7731742473213E-03, -0.017834862292358, -0.045996013696365, -
0.057581259083432, -0.05032527872793, -3.3032641670203E-05, -
1.8948987516315E-04, -3.9392777243355E-03, -0.043797295650573, -
2.6674547914087E-05, 2.0481737692309E-08, 4.3870667284435E-07, -
3.227767723857E-05, -1.5033924542148E-03, -0.040668253562649, -
7.8847309559367E-10, 1.2790717852285E-08, 4.8225372718507E-07,
2.2922076337661E-06, -1.6714766451061E-11, -2.1171472321355E-03, -
23.895741934104, -5.905956432427E-18, -1.2621808899101E-06, -
0.038946842435739, 1.1256211360459E-11, -8.2311340897998,
1.9809712802088E-08, 1.0406965210174E-19, -1.0234747095929E-13, -
1.0018179379511E-09, -8.0882908646985E-11, 0.10693031879409, -
0.33662250574171, 8.9185845355421E-25, 3.0629316876232E-13, -
4.2002467698208E-06, -5.9056029685639E-26, 3.7826947613457E-06, -
1.2768608934681E-15, 7.3087610595061E-29, 5.5414715350778E-17, -
9.436970724121E-07];
R = 0.461526; %kJ/(kg K)
Pi = p;
tau = 540 / T;
g0_tautau = 0;
for i = 1 : 9
    g0_tautau = g0_tautau + n0(i) * J0(i) * (J0(i) - 1) * tau ^ (J0(i)
- 2);
end
gr_pi = 0;
gr_pitau = 0;
gr_pipi = 0;
gr_tautau = 0;
for i = 1 : 43
    gr_pi = gr_pi + nr(i) * Ir(i) * Pi ^ (Ir(i) - 1) * (tau - 0.5) ^
Jr(i);
    gr_pipi = gr_pipi + nr(i) * Ir(i) * (Ir(i) - 1) * Pi ^ (Ir(i) - 2)
* (tau - 0.5) ^ Jr(i);
    gr_pitau = gr_pitau + nr(i) * Ir(i) * Pi ^ (Ir(i) - 1) * Jr(i) *
(tau - 0.5) ^ (Jr(i) - 1);

```

```

    gr_tautau = gr_tautau + nr(i) * Pi ^ Ir(i) * Jr(i) * (Jr(i) - 1) *
    (tau - 0.5) ^ (Jr(i) - 2);
end
Cv2_pT = R * (-tau ^ 2 * (g0_tautau + gr_tautau) - (1 + Pi * gr_pi -
tau * Pi * gr_pitau) ^ 2 / (1 - Pi ^ 2 * gr_pipi));

function w2_pT = w2_pT(p, T)
%Release on the IAPWS Industrial formulation 1997 for the
Thermodynamic Properties of Water and Steam, September 1997
%6 Equations for Region 2, Section. 6.1 Basic Equation
%Table 11 and 12, Page 14 and 15
J0 = [0, 1, -5, -4, -3, -2, -1, 2, 3];
n0 = [-9.6927686500217, 10.086655968018, -0.005608791128302,
0.071452738081455, -0.40710498223928, 1.4240819171444, -
4.383951131945, -0.28408632460772, 0.021268463753307];
Ir = [1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 4, 4, 4, 5, 6, 6,
6, 7, 7, 7, 8, 8, 9, 10, 10, 10, 16, 16, 18, 20, 20, 20, 21, 22, 23,
24, 24, 24];
Jr = [0, 1, 2, 3, 6, 1, 2, 4, 7, 36, 0, 1, 3, 6, 35, 1, 2, 3, 7, 3,
16, 35, 0, 11, 25, 8, 36, 13, 4, 10, 14, 29, 50, 57, 20, 35, 48, 21,
53, 39, 26, 40, 58];
nr = [-1.7731742473213E-03, -0.017834862292358, -0.045996013696365, -
0.057581259083432, -0.05032527872793, -3.3032641670203E-05, -
1.8948987516315E-04, -3.9392777243355E-03, -0.043797295650573, -
2.6674547914087E-05, 2.0481737692309E-08, 4.3870667284435E-07, -
3.227767723857E-05, -1.5033924542148E-03, -0.040668253562649, -
7.8847309559367E-10, 1.2790717852285E-08, 4.8225372718507E-07,
2.2922076337661E-06, -1.6714766451061E-11, -2.1171472321355E-03, -
23.895741934104, -5.905956432427E-18, -1.2621808899101E-06, -
0.038946842435739, 1.1256211360459E-11, -8.2311340897998,
1.9809712802088E-08, 1.0406965210174E-19, -1.0234747095929E-13, -
1.0018179379511E-09, -8.0882908646985E-11, 0.10693031879409, -
0.33662250574171, 8.9185845355421E-25, 3.0629316876232E-13, -
4.2002467698208E-06, -5.9056029685639E-26, 3.7826947613457E-06, -
1.2768608934681E-15, 7.3087610595061E-29, 5.5414715350778E-17, -
9.436970724121E-07];
R = 0.461526; %kJ/(kg K)
Pi = p;
tau = 540 / T;
g0_tautau = 0;
for i = 1 : 9
    g0_tautau = g0_tautau + n0(i) * J0(i) * (J0(i) - 1) * tau ^ (J0(i)
- 2);
end
gr_pi = 0;
gr_pitau = 0;
gr_pipi = 0;
gr_tautau = 0;
for i = 1 : 43
    gr_pi = gr_pi + nr(i) * Ir(i) * Pi ^ (Ir(i) - 1) * (tau - 0.5) ^
Jr(i);
    gr_pipi = gr_pipi + nr(i) * Ir(i) * (Ir(i) - 1) * Pi ^ (Ir(i) - 2)
* (tau - 0.5) ^ Jr(i);
    gr_pitau = gr_pitau + nr(i) * Ir(i) * Pi ^ (Ir(i) - 1) * Jr(i) *
(tau - 0.5) ^ (Jr(i) - 1);
    gr_tautau = gr_tautau + nr(i) * Pi ^ Ir(i) * Jr(i) * (Jr(i) - 1) *
(tau - 0.5) ^ (Jr(i) - 2);
end
w2_pT = (1000 * R * T * (1 + 2 * Pi * gr_pi + Pi ^ 2 * gr_pi ^ 2) /
((1 - Pi ^ 2 * gr_pipi) + (1 + Pi * gr_pi - tau * Pi * gr_pitau) ^ 2 /
(tau ^ 2 * (g0_tautau + gr_tautau)))) ^ 0.5;

```

```

function T2_ph = T2_ph(p, h)
%Release on the IAPWS Industrial formulation 1997 for the
Thermodynamic Properties of Water and Steam, September 1997
%6 Equations for Region 2,6.3.1 The Backward Equations T( p, h ) for
Subregions 2a, 2b, and 2c
if p < 4
    sub_reg = 1;
else
    if p < (905.84278514723 - 0.67955786399241 * h + 1.2809002730136E-
04 * h ^ 2)
        sub_reg = 2;
    else
        sub_reg = 3;
    end
end

switch sub_reg
case 1
    %Subregion A
    %Table 20, Eq 22, page 22
    Ji = [0, 1, 2, 3, 7, 20, 0, 1, 2, 3, 7, 9, 11, 18, 44, 0, 2, 7,
36, 38, 40, 42, 44, 24, 44, 12, 32, 44, 32, 36, 42, 34, 44, 28];
    Ii = [0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2,
2, 2, 2, 3, 3, 4, 4, 4, 5, 5, 5, 6, 6, 7];
    ni = [1089.8952318288, 849.51654495535, -107.81748091826,
33.153654801263, -7.4232016790248, 11.765048724356, 1.844574935579, -
4.1792700549624, 6.2478196935812, -17.344563108114, -200.58176862096,
271.96065473796, -455.11318285818, 3091.9688604755, 252266.40357872, -
6.1707422868339E-03, -0.31078046629583, 11.670873077107,
128127984.04046, -985549096.23276, 2822454697.3002, -3594897141.0703,
1722734991.3197, -13551.334240775, 12848734.66465, 1.3865724283226,
235988.32556514, -13105236.545054, 7399.9835474766, -551966.9703006,
3715408.5996233, 19127.72923966, -415351.64835634, -62.459855192507];
    Ts = 0;
    hs = h / 2000;
    for i = 1 : 34
        Ts = Ts + ni(i) * p ^ (Ii(i)) * (hs - 2.1) ^ Ji(i);
    end
    T2_ph = Ts;
case 2
    %Subregion B
    %Table 21, Eq 23, page 23
    Ji = [0, 1, 2, 12, 18, 24, 28, 40, 0, 2, 6, 12, 18, 24, 28, 40, 2,
8, 18, 40, 1, 2, 12, 24, 2, 12, 18, 24, 28, 40, 18, 24, 40, 28, 2, 28,
1, 40];
    Ii = [0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2,
3, 3, 3, 3, 4, 4, 4, 4, 4, 5, 5, 5, 6, 7, 7, 9, 9];
    ni = [1489.5041079516, 743.07798314034, -97.708318797837,
2.4742464705674, -0.63281320016026, 1.1385952129658, -
0.47811863648625, 8.5208123431544E-03, 0.93747147377932,
3.3593118604916, 3.3809355601454, 0.16844539671904, 0.73875745236695,
-0.47128737436186, 0.15020273139707, -0.002176411421975, -
0.021810755324761, -0.10829784403677, -0.046333324635812,
7.1280351959551E-05, 1.1032831789999E-04, 1.8955248387902E-04,
3.0891541160537E-03, 1.3555504554949E-03, 2.8640237477456E-07, -
1.0779857357512E-05, -7.6462712454814E-05, 1.4052392818316E-05, -
3.1083814331434E-05, -1.0302738212103E-06, 2.821728163504E-07,
1.2704902271945E-06, 7.3803353468292E-08, -1.1030139238909E-08, -
8.1456365207833E-14, -2.5180545682962E-11, -1.7565233969407E-18,
8.6934156344163E-15];

```

```

    Ts = 0;
    hs = h / 2000;
    for i = 1 : 38
        Ts = Ts + ni(i) * (p - 2) ^ (Ii(i)) * (hs - 2.6) ^ Ji(i);
    end
    T2_ph = Ts;
otherwise
    %Subregion C
    %Table 22, Eq 24, page 24
    Ji = [0, 4, 0, 2, 0, 2, 0, 1, 0, 2, 0, 1, 4, 8, 4, 0, 1, 4, 10,
12, 16, 20, 22];
    Ii = [-7, -7, -6, -6, -5, -5, -2, -2, -1, -1, 0, 0, 1, 1, 2, 6, 6,
6, 6, 6, 6, 6, 6];
    ni = [-3236839855524.2, 7326335090218.1, 358250899454.47, -
583401318515.9, -10783068217.47, 20825544563.171, 610747.83564516,
859777.2253558, -25745.72360417, 31081.088422714, 1208.2315865936,
482.19755109255, 3.7966001272486, -10.842984880077, -0.04536417267666,
1.4559115658698E-13, 1.126159740723E-12, -1.7804982240686E-11,
1.2324579690832E-07, -1.1606921130984E-06, 2.7846367088554E-05, -
5.9270038474176E-04, 1.2918582991878E-03];
    Ts = 0;
    hs = h / 2000;
    for i = 1 : 23
        Ts = Ts + ni(i) * (p + 25) ^ (Ii(i)) * (hs - 1.8) ^ Ji(i);
    end
    T2_ph = Ts;
end

function T2_ps = T2_ps(p, s)
%Release on the IAPWS Industrial formulation 1997 for the
Thermodynamic Properties of Water and Steam, September 1997
%6 Equations for Region 2,6.3.2 The Backward Equations T( p, s ) for
Subregions 2a, 2b, and 2c
%Page 26
if p < 4
    sub_reg = 1;
else
    if s < 5.85
        sub_reg = 3;
    else
        sub_reg = 2;
    end
end
switch sub_reg
case 1
    %Subregion A
    %Table 25, Eq 25, page 26
    Ii = [-1.5, -1.5, -1.5, -1.5, -1.5, -1.5, -1.25, -1.25, -1.25, -1,
-1, -1, -1, -1, -0.75, -0.75, -0.5, -0.5, -0.5, -0.5, -0.25, -
0.25, -0.25, 0.25, 0.25, 0.25, 0.25, 0.5, 0.5, 0.5, 0.5, 0.5,
0.5, 0.5, 0.75, 0.75, 0.75, 0.75, 1, 1, 1.25, 1.25, 1.5, 1.5];
    Ji = [-24, -23, -19, -13, -11, -10, -19, -15, -6, -26, -21, -17, -
16, -9, -8, -15, -14, -26, -13, -9, -7, -27, -25, -11, -6, 1, 4, 8,
11, 0, 1, 5, 6, 10, 14, 16, 0, 4, 9, 17, 7, 18, 3, 15, 5, 18];
    ni = [-392359.83861984, 515265.7382727, 40482.443161048, -
321.93790923902, 96.961424218694, -22.867846371773, -449429.14124357,
-5011.8336020166, 0.35684463560015, 44235.33584819, -13673.388811708,
421632.60207864, 22516.925837475, 474.42144865646, -149.31130797647, -
197811.26320452, -23554.39947076, -19070.616302076, 55375.669883164,
3829.3691437363, -603.91860580567, 1936.3102620331, 4266.064369861, -
5978.0638872718, -704.01463926862, 338.36784107553, 20.862786635187,

```

```

0.033834172656196, -4.3124428414893E-05, 166.53791356412, -
139.86292055898, -0.78849547999872, 0.072132411753872, -
5.9754839398283E-03, -1.2141358953904E-05, 2.3227096733871E-07, -
10.538463566194, 2.0718925496502, -0.072193155260427, 2.074988708112E-
07, -0.018340657911379, 2.9036272348696E-07, 0.21037527893619,
2.5681239729999E-04, -0.012799002933781, -8.2198102652018E-06];
    Pi = p;
    Sigma = s / 2;
    teta = 0;
    for i = 1 : 46
        teta = teta + ni(i) * Pi ^ Ii(i) * (Sigma - 2) ^ Ji(i);
    end
    T2_ps = teta;
case 2
    %Subregion B
    %Table 26, Eq 26, page 27
    Ii = [-6, -6, -5, -5, -4, -4, -4, -3, -3, -3, -3, -2, -2, -2, -2,
-1, -1, -1, -1, -1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 2, 2, 2, 3,
3, 3, 4, 4, 5, 5, 5];
    Ji = [0, 11, 0, 11, 0, 1, 11, 0, 1, 11, 12, 0, 1, 6, 10, 0, 1, 5,
8, 9, 0, 1, 2, 4, 5, 6, 9, 0, 1, 2, 3, 7, 8, 0, 1, 5, 0, 1, 3, 0, 1,
0, 1, 2];
    ni = [316876.65083497, 20.864175881858, -398593.99803599, -
21.816058518877, 223697.85194242, -2784.1703445817, 9.920743607148, -
75197.512299157, 2970.8605951158, -3.4406878548526, 0.38815564249115,
17511.29508575, -1423.7112854449, 1.0943803364167, 0.89971619308495, -
3375.9740098958, 471.62885818355, -1.9188241993679, 0.41078580492196,
-0.33465378172097, 1387.0034777505, -406.63326195838, 41.72734715961,
2.1932549434532, -1.0320050009077, 0.35882943516703, 5.2511453726066E-
03, 12.838916450705, -2.8642437219381, 0.56912683664855, -
0.099962954584931, -3.2632037778459E-03, 2.3320922576723E-04, -
0.1533480985745, 0.029072288239902, 3.7534702741167E-04,
1.7296691702411E-03, -3.8556050844504E-04, -3.5017712292608E-05, -
1.4566393631492E-05, 5.6420857267269E-06, 4.1286150074605E-08, -
2.0684671118824E-08, 1.6409393674725E-09];
    Pi = p;
    Sigma = s / 0.7853;
    teta = 0;
    for i = 1 : 44
        teta = teta + ni(i) * Pi ^ Ii(i) * (10 - Sigma) ^ Ji(i);
    end
    T2_ps = teta;
otherwise
    %Subregion C
    %Table 27, Eq 27, page 28
    Ii = [-2, -2, -1, 0, 0, 0, 1, 1, 1, 1, 2, 2, 2, 3, 3, 3, 4, 4,
4, 5, 5, 5, 6, 6, 7, 7, 7, 7, 7];
    Ji = [0, 1, 0, 0, 1, 2, 3, 0, 1, 3, 4, 0, 1, 2, 0, 1, 5, 0, 1, 4,
0, 1, 2, 0, 1, 0, 1, 3, 4, 5];
    ni = [909.68501005365, 2404.566708842, -591.6232638713,
541.45404128074, -270.98308411192, 979.76525097926, -469.66772959435,
14.399274604723, -19.104204230429, 5.3299167111971, -21.252975375934,
-0.3114733441376, 0.60334840894623, -0.042764839702509,
5.8185597255259E-03, -0.014597008284753, 5.6631175631027E-03, -
7.6155864584577E-05, 2.2440342919332E-04, -1.2561095013413E-05,
6.3323132660934E-07, -2.0541989675375E-06, 3.6405370390082E-08, -
2.9759897789215E-09, 1.0136618529763E-08, 5.9925719692351E-12, -
2.0677870105164E-11, -2.0874278181886E-11, 1.0162166825089E-10, -
1.6429828281347E-10];
    Pi = p;
    Sigma = s / 2.9251;

```

```

teta = 0;
for i = 1 : 30
    teta = teta + ni(i) * Pi ^ Ii(i) * (2 - Sigma) ^ Ji(i);
end
T2_ps = teta;
end

function p2_hs = p2_hs(h, s)
%Supplementary Release on Backward Equations for Pressure as a
function of Enthalpy and Entropy p(h,s) to the IAPWS Industrial
formulation 1997 for the Thermodynamic Properties of Water and Steam
%Chapter 6:Backward Equations p(h,s) for Region 2
if h < (-3498.98083432139 + 2575.60716905876 * s - 421.073558227969 *
s ^ 2 + 27.6349063799944 * s ^ 3)
    sub_reg = 1;
else
    if s < 5.85
        sub_reg = 3;
    else
        sub_reg = 2;
    end
end
switch sub_reg
case 1
    %Subregion A
    %Table 6, Eq 3, page 8
    Ii = [0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 3,
3, 3, 3, 3, 4, 5, 5, 6, 7];
    Ji = [1, 3, 6, 16, 20, 22, 0, 1, 2, 3, 5, 6, 10, 16, 20, 22, 3,
16, 20, 0, 2, 3, 6, 16, 16, 3, 16, 3, 1];
    ni = [-1.82575361923032E-02, -0.125229548799536,
0.592290437320145, 6.04769706185122, 238.624965444474, -
298.639090222922, 0.051225081304075, -0.437266515606486,
0.413336902999504, -5.16468254574773, -5.57014838445711,
12.8555037824478, 11.414410895329, -119.504225652714, -
2847.7798596156, 4317.57846408006, 1.1289404080265, 1974.09186206319,
1516.12444706087, 1.41324451421235E-02, 0.585501282219601, -
2.97258075863012, 5.94567314847319, -6236.56565798905,
9659.86235133332, 6.81500934948134, -6332.07286824489, -
5.5891922446576, 4.00645798472063E-02];
    eta = h / 4200;
    Sigma = s / 12;
    Pi = 0;
    for i = 1 : 29
        Pi = Pi + ni(i) * (eta - 0.5) ^ Ii(i) * (Sigma - 1.2) ^ Ji(i);
    end
    p2_hs = Pi ^ 4 * 4;
case 2
    %Subregion B
    %Table 7, Eq 4, page 9
    Ii = [0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 2, 2, 2, 3, 3, 3, 3, 4, 4,
5, 5, 6, 6, 6, 7, 7, 8, 8, 8, 8, 12, 14];
    Ji = [0, 1, 2, 4, 8, 0, 1, 2, 3, 5, 12, 1, 6, 18, 0, 1, 7, 12, 1,
16, 1, 12, 1, 8, 18, 1, 16, 1, 3, 14, 18, 10, 16];
    ni = [8.01496989929495E-02, -0.543862807146111, 0.337455597421283,
8.9055545115745, 313.840736431485, 0.797367065977789, -
1.2161697355624, 8.72803386937477, -16.9769781757602, -
186.552827328416, 95115.9274344237, -18.9168510120494, -
4334.0703719484, 543212633.012715, 0.144793408386013,
128.024559637516, -67230.9534071268, 33697238.0095287, -
586.63419676272, -22140322476.9889, 1716.06668708389, -

```

```

570817595.806302, -3121.09693178482, -2078413.8463301,
3056059461577.86, 3221.57004314333, 326810259797.295, -
1441.04158934487, 410.694867802691, 109077066873.024, -
24796465425889.3, 1888019068.65134, -123651009018773];
    eta = h / 4100;
    Sigma = s / 7.9;
    Pi = 0;
    for i = 1 : 33
        Pi = Pi + ni(i) * (eta - 0.6) ^ Ii(i) * (Sigma - 1.01) ^
Ji(i);
    end
    p2_hs = Pi ^ 4 * 100;
otherwise
    %Subregion C
    %Table 8, Eq 5, page 10
    Ii = [0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 3, 3,
3, 4, 5, 5, 5, 5, 6, 6, 10, 12, 16];
    Ji = [0, 1, 2, 3, 4, 8, 0, 2, 5, 8, 14, 2, 3, 7, 10, 18, 0, 5, 8,
16, 18, 18, 1, 4, 6, 14, 8, 18, 7, 7, 10];
    ni = [0.112225607199012, -3.39005953606712, -32.0503911730094, -
197.5973051049, -407.693861553446, 13294.3775222331, 1.70846839774007,
37.3694198142245, 3581.44365815434, 423014.446424664, -
751071025.760063, 52.3446127607898, -228.351290812417, -
960652.417056937, -80705929.2526074, 1626980172256.69,
0.772465073604171, 46392.9973837746, -13731788.5134128,
1704703926305.12, -25110462818730.8, 31774883083552, 53.8685623675312,
-55308.9094625169, -1028615.22421405, 2042494187562.34,
273918446.626977, -2.63963146312685E+15, -1078908541.08088, -
29649262098.0124, -1.11754907323424E+15];
    eta = h / 3500;
    Sigma = s / 5.9;
    Pi = 0;
    for i = 1 : 31
        Pi = Pi + ni(i) * (eta - 0.7) ^ Ii(i) * (Sigma - 1.1) ^ Ji(i);
    end
    p2_hs = Pi ^ 4 * 100;
end

function T2_prho=T2_prho(p,rho)
    %Solve by iteration. Observe that fo low temperatures this equation
has 2 solutions.
    %Solve with half interval method

    if p < 16.5292
        Low_Bound = T4_p(p);
    else
        Low_Bound = B23T_p(p);
    end
    High_Bound = 1073.15;
    rhos=-1000;
    while abs(rho - rhos) > 0.000001
        Ts = (Low_Bound + High_Bound) / 2;
        rhos = 1 / v2_pT(p, Ts);
        if rhos < rho
            High_Bound = Ts;
        else
            Low_Bound = Ts;
        end
    end
    T2_prho = Ts;
end

```

```
%*****  
*****
```

```
%*2.3 functions for region 3
```

```
function p3_rhoT = p3_rhoT(rho, T)  
%Release on the IAPWS Industrial formulation 1997 for the  
Thermodynamic Properties of Water and Steam, September 1997  
%7 Basic Equation for Region 3, Section. 6.1 Basic Equation  
%Table 30 and 31, Page 30 and 31  
Ii = [0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 3, 3, 3,  
3, 3, 4, 4, 4, 4, 5, 5, 5, 6, 6, 6, 7, 8, 9, 9, 10, 10, 11];  
Ji = [0, 0, 1, 2, 7, 10, 12, 23, 2, 6, 15, 17, 0, 2, 6, 7, 22, 26, 0,  
2, 4, 16, 26, 0, 2, 4, 26, 1, 3, 26, 0, 2, 26, 2, 26, 2, 26, 0, 1,  
26];  
ni = [1.0658070028513, -15.732845290239, 20.944396974307, -  
7.6867707878716, 2.6185947787954, -2.808078114862, 1.2053369696517, -  
8.4566812812502E-03, -1.2654315477714, -1.1524407806681,  
0.88521043984318, -0.64207765181607, 0.38493460186671, -  
0.85214708824206, 4.8972281541877, -3.0502617256965,  
0.039420536879154, 0.12558408424308, -0.2799932969871, 1.389979956946,  
-2.018991502357, -8.2147637173963E-03, -0.47596035734923,  
0.0439840744735, -0.44476435428739, 0.90572070719733,  
0.70522450087967, 0.10770512626332, -0.32913623258954, -  
0.50871062041158, -0.022175400873096, 0.094260751665092,  
0.16436278447961, -0.013503372241348, -0.014834345352472,  
5.7922953628084E-04, 3.2308904703711E-03, 8.0964802996215E-05, -  
1.6557679795037E-04, -4.4923899061815E-05];  
R = 0.461526; %kJ/(KgK)  
tc = 647.096; %K  
pc = 22.064; %MPa  
rhoc = 322; %kg/m3  
delta = rho / rhoc;  
tau = tc / T;  
fidelta = 0;  
for i = 2 : 40  
    fidelta = fidelta + ni(i) * Ii(i) * delta ^ (Ii(i) - 1) * tau ^  
    Ji(i);  
end  
fidelta = fidelta + ni(1) / delta;  
p3_rhoT = rho * R * T * delta * fidelta / 1000;
```

```
function u3_rhoT = u3_rhoT(rho, T)  
%Release on the IAPWS Industrial formulation 1997 for the  
Thermodynamic Properties of Water and Steam, September 1997  
%7 Basic Equation for Region 3, Section. 6.1 Basic Equation  
%Table 30 and 31, Page 30 and 31  
Ii = [0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 3, 3, 3,  
3, 3, 4, 4, 4, 4, 5, 5, 5, 6, 6, 6, 7, 8, 9, 9, 10, 10, 11];  
Ji = [0, 0, 1, 2, 7, 10, 12, 23, 2, 6, 15, 17, 0, 2, 6, 7, 22, 26, 0,  
2, 4, 16, 26, 0, 2, 4, 26, 1, 3, 26, 0, 2, 26, 2, 26, 2, 26, 0, 1,  
26];  
ni = [1.0658070028513, -15.732845290239, 20.944396974307, -  
7.6867707878716, 2.6185947787954, -2.808078114862, 1.2053369696517, -  
8.4566812812502E-03, -1.2654315477714, -1.1524407806681,  
0.88521043984318, -0.64207765181607, 0.38493460186671, -  
0.85214708824206, 4.8972281541877, -3.0502617256965,  
0.039420536879154, 0.12558408424308, -0.2799932969871, 1.389979956946,  
-2.018991502357, -8.2147637173963E-03, -0.47596035734923,  
0.0439840744735, -0.44476435428739, 0.90572070719733,  
0.70522450087967, 0.10770512626332, -0.32913623258954, -  
0.50871062041158, -0.022175400873096, 0.094260751665092,
```



```

0.16436278447961, -0.013503372241348, -0.014834345352472,
5.7922953628084E-04, 3.2308904703711E-03, 8.0964802996215E-05, -
1.6557679795037E-04, -4.4923899061815E-05];
R = 0.461526; %kJ/(KgK)
tc = 647.096; %K
pc = 22.064; %MPa
rhoc = 322; %kg/m3
delta = rho / rhoc;
tau = tc / T;
fitau = 0;
for i = 2 : 40
    fitau = fitau + ni(i) * delta ^ Ii(i) * Ji(i) * tau ^ (Ji(i) - 1);
end
u3_rhoT = R * T * (tau * fitau);

```

```

function h3_rhoT = h3_rhoT(rho, T)
%Release on the IAPWS Industrial formulation 1997 for the
Thermodynamic Properties of Water and Steam, September 1997
%7 Basic Equation for Region 3, Section. 6.1 Basic Equation
%Table 30 and 31, Page 30 and 31
Ii = [0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 3, 3, 3,
3, 3, 4, 4, 4, 4, 5, 5, 5, 6, 6, 6, 7, 8, 9, 9, 10, 10, 11];
Ji = [0, 0, 1, 2, 7, 10, 12, 23, 2, 6, 15, 17, 0, 2, 6, 7, 22, 26, 0,
2, 4, 16, 26, 0, 2, 4, 26, 1, 3, 26, 0, 2, 26, 2, 26, 2, 26, 0, 1,
26];
ni = [1.0658070028513, -15.732845290239, 20.944396974307, -
7.6867707878716, 2.6185947787954, -2.808078114862, 1.2053369696517, -
8.4566812812502E-03, -1.2654315477714, -1.1524407806681, -
0.88521043984318, -0.64207765181607, 0.38493460186671, -
0.85214708824206, 4.8972281541877, -3.0502617256965,
0.039420536879154, 0.12558408424308, -0.2799932969871, 1.389979956946,
-2.018991502357, -8.2147637173963E-03, -0.47596035734923,
0.0439840744735, -0.44476435428739, 0.90572070719733,
0.70522450087967, 0.10770512626332, -0.32913623258954, -
0.50871062041158, -0.022175400873096, 0.094260751665092,
0.16436278447961, -0.013503372241348, -0.014834345352472,
5.7922953628084E-04, 3.2308904703711E-03, 8.0964802996215E-05, -
1.6557679795037E-04, -4.4923899061815E-05];
R = 0.461526; %kJ/(KgK)
tc = 647.096; %K
pc = 22.064; %MPa
rhoc = 322; %kg/m3
delta = rho / rhoc;
tau = tc / T;
fidelta = 0;
fitau = 0;
for i = 2 : 40
    fidelta = fidelta + ni(i) * Ii(i) * delta ^ (Ii(i) - 1) * tau ^
Ji(i);
    fitau = fitau + ni(i) * delta ^ Ii(i) * Ji(i) * tau ^ (Ji(i) - 1);
end
fidelta = fidelta + ni(1) / delta;
h3_rhoT = R * T * (tau * fitau + delta * fidelta);

```

```

function s3_rhoT = s3_rhoT(rho, T)
%Release on the IAPWS Industrial formulation 1997 for the
Thermodynamic Properties of Water and Steam, September 1997
%7 Basic Equation for Region 3, Section. 6.1 Basic Equation
%Table 30 and 31, Page 30 and 31

```

```

Ii = [0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 3, 3, 3,
3, 3, 4, 4, 4, 4, 5, 5, 5, 6, 6, 6, 7, 8, 9, 9, 10, 10, 11];
Ji = [0, 0, 1, 2, 7, 10, 12, 23, 2, 6, 15, 17, 0, 2, 6, 7, 22, 26, 0,
2, 4, 16, 26, 0, 2, 4, 26, 1, 3, 26, 0, 2, 26, 2, 26, 2, 26, 0, 1,
26];
ni = [1.0658070028513, -15.732845290239, 20.944396974307, -
7.6867707878716, 2.6185947787954, -2.808078114862, 1.2053369696517, -
8.4566812812502E-03, -1.2654315477714, -1.1524407806681,
0.88521043984318, -0.64207765181607, 0.38493460186671, -
0.85214708824206, 4.8972281541877, -3.0502617256965,
0.039420536879154, 0.12558408424308, -0.2799932969871, 1.389979956946,
-2.018991502357, -8.2147637173963E-03, -0.47596035734923,
0.0439840744735, -0.44476435428739, 0.90572070719733,
0.70522450087967, 0.10770512626332, -0.32913623258954, -
0.50871062041158, -0.022175400873096, 0.094260751665092,
0.16436278447961, -0.013503372241348, -0.014834345352472,
5.7922953628084E-04, 3.2308904703711E-03, 8.0964802996215E-05, -
1.6557679795037E-04, -4.4923899061815E-05];
R = 0.461526; %kJ/(KgK)
tc = 647.096; %K
pc = 22.064; %MPa
rhoc = 322; %kg/m3
delta = rho / rhoc;
tau = tc / T;
fi = 0;
fitau = 0;
for i = 2 : 40
    fi = fi + ni(i) * delta ^ Ii(i) * tau ^ Ji(i);
    fitau = fitau + ni(i) * delta ^ Ii(i) * Ji(i) * tau ^ (Ji(i) - 1);
end
fi = fi + ni(1) * log(delta);
s3_rhoT = R * (tau * fitau - fi);

function Cp3_rhoT = Cp3_rhoT(rho, T)
%Release on the IAPWS Industrial formulation 1997 for the
Thermodynamic Properties of Water and Steam, September 1997
%7 Basic Equation for Region 3, Section. 6.1 Basic Equation
%Table 30 and 31, Page 30 and 31
Ii = [0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 3, 3, 3,
3, 3, 4, 4, 4, 4, 5, 5, 5, 6, 6, 6, 7, 8, 9, 9, 10, 10, 11];
Ji = [0, 0, 1, 2, 7, 10, 12, 23, 2, 6, 15, 17, 0, 2, 6, 7, 22, 26, 0,
2, 4, 16, 26, 0, 2, 4, 26, 1, 3, 26, 0, 2, 26, 2, 26, 2, 26, 0, 1,
26];
ni = [1.0658070028513, -15.732845290239, 20.944396974307, -
7.6867707878716, 2.6185947787954, -2.808078114862, 1.2053369696517, -
8.4566812812502E-03, -1.2654315477714, -1.1524407806681,
0.88521043984318, -0.64207765181607, 0.38493460186671, -
0.85214708824206, 4.8972281541877, -3.0502617256965,
0.039420536879154, 0.12558408424308, -0.2799932969871, 1.389979956946,
-2.018991502357, -8.2147637173963E-03, -0.47596035734923,
0.0439840744735, -0.44476435428739, 0.90572070719733,
0.70522450087967, 0.10770512626332, -0.32913623258954, -
0.50871062041158, -0.022175400873096, 0.094260751665092,
0.16436278447961, -0.013503372241348, -0.014834345352472,
5.7922953628084E-04, 3.2308904703711E-03, 8.0964802996215E-05, -
1.6557679795037E-04, -4.4923899061815E-05];
R = 0.461526; %kJ/(KgK)
tc = 647.096; %K
pc = 22.064; %MPa
rhoc = 322; %kg/m3
delta = rho / rhoc;

```

```

tau = tc / T;
fitautau = 0;
fidelta = 0;
fidelatau = 0;
fidelatadelta = 0;
for i = 2 : 40
    fitautau = fitautau + ni(i) * delta ^ Ii(i) * Ji(i) * (Ji(i) - 1)
* tau ^ (Ji(i) - 2);
    fidelatau = fidelatau + ni(i) * Ii(i) * delta ^ (Ii(i) - 1) * tau ^
Ji(i);
    fidelatau = fidelatau + ni(i) * Ii(i) * delta ^ (Ii(i) - 1) *
Ji(i) * tau ^ (Ji(i) - 1);
    fidelatadelta = fidelatadelta + ni(i) * Ii(i) * (Ii(i) - 1) * delta
^ (Ii(i) - 2) * tau ^ Ji(i);
end
fidelatau = fidelatau + ni(1) / delta;
fidelatadelta = fidelatadelta - ni(1) / (delta ^ 2);
Cp3_rhoT = R * (-tau ^ 2 * fitautau + (delta * fidelatau - delta * tau *
fidelatadelta) ^ 2 / (2 * delta * fidelatau + delta ^ 2 * fidelatadelta));

function Cv3_rhoT = Cv3_rhoT(rho, T)
%Release on the IAPWS Industrial formulation 1997 for the
Thermodynamic Properties of Water and Steam, September 1997
%7 Basic Equation for Region 3, Section. 6.1 Basic Equation
%Table 30 and 31, Page 30 and 31
Ii = [0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 3, 3, 3,
3, 3, 4, 4, 4, 4, 5, 5, 5, 6, 6, 6, 7, 8, 9, 9, 10, 10, 11];
Ji = [0, 0, 1, 2, 7, 10, 12, 23, 2, 6, 15, 17, 0, 2, 6, 7, 22, 26, 0,
2, 4, 16, 26, 0, 2, 4, 26, 1, 3, 26, 0, 2, 26, 2, 26, 2, 26, 0, 1,
26];
ni = [1.0658070028513, -15.732845290239, 20.944396974307, -
7.6867707878716, 2.6185947787954, -2.808078114862, 1.2053369696517, -
8.4566812812502E-03, -1.2654315477714, -1.1524407806681,
0.88521043984318, -0.64207765181607, 0.38493460186671, -
0.85214708824206, 4.8972281541877, -3.0502617256965,
0.039420536879154, 0.12558408424308, -0.2799932969871, 1.389979956946,
-2.018991502357, -8.2147637173963E-03, -0.47596035734923,
0.0439840744735, -0.44476435428739, 0.90572070719733,
0.70522450087967, 0.10770512626332, -0.32913623258954, -
0.50871062041158, -0.022175400873096, 0.094260751665092,
0.16436278447961, -0.013503372241348, -0.014834345352472,
5.7922953628084E-04, 3.2308904703711E-03, 8.0964802996215E-05, -
1.6557679795037E-04, -4.4923899061815E-05];
R = 0.461526; %kJ/(KgK)
tc = 647.096; %K
pc = 22.064; %MPa
rhoc = 322; %kg/m3
delta = rho / rhoc;
tau = tc / T;
fitautau = 0;
for i = 1 : 40
    fitautau = fitautau + ni(i) * delta ^ Ii(i) * Ji(i) * (Ji(i) - 1)
* tau ^ (Ji(i) - 2);
end
Cv3_rhoT = R * -(tau * tau * fitautau);

function w3_rhoT = w3_rhoT(rho, T)
%Release on the IAPWS Industrial formulation 1997 for the
Thermodynamic Properties of Water and Steam, September 1997
%7 Basic Equation for Region 3, Section. 6.1 Basic Equation
%Table 30 and 31, Page 30 and 31

```

```

Ii = [0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 3, 3, 3,
3, 3, 4, 4, 4, 4, 5, 5, 5, 6, 6, 6, 7, 8, 9, 9, 10, 10, 11];
Ji = [0, 0, 1, 2, 7, 10, 12, 23, 2, 6, 15, 17, 0, 2, 6, 7, 22, 26, 0,
2, 4, 16, 26, 0, 2, 4, 26, 1, 3, 26, 0, 2, 26, 2, 26, 2, 26, 0, 1,
26];
ni = [1.0658070028513, -15.732845290239, 20.944396974307, -
7.6867707878716, 2.6185947787954, -2.808078114862, 1.2053369696517, -
8.4566812812502E-03, -1.2654315477714, -1.1524407806681, -
0.88521043984318, -0.64207765181607, 0.38493460186671, -
0.85214708824206, 4.8972281541877, -3.0502617256965,
0.039420536879154, 0.12558408424308, -0.2799932969871, 1.389979956946,
-2.018991502357, -8.2147637173963E-03, -0.47596035734923,
0.0439840744735, -0.44476435428739, 0.90572070719733,
0.70522450087967, 0.10770512626332, -0.32913623258954, -
0.50871062041158, -0.022175400873096, 0.094260751665092,
0.16436278447961, -0.013503372241348, -0.014834345352472,
5.7922953628084E-04, 3.2308904703711E-03, 8.0964802996215E-05, -
1.6557679795037E-04, -4.4923899061815E-05];
R = 0.461526; %kJ/(KgK)
tc = 647.096; %K
pc = 22.064; %MPa
rhoc = 322; %kg/m3
delta = rho / rhoc;
tau = tc / T;
fitautau = 0;
fidelta = 0;
fideltatau = 0;
fideltadelta = 0;
for i = 2 : 40
    fitautau = fitautau + ni(i) * delta ^ Ii(i) * Ji(i) * (Ji(i) - 1)
* tau ^ (Ji(i) - 2);
    fidelta = fidelta + ni(i) * Ii(i) * delta ^ (Ii(i) - 1) * tau ^
Ji(i);
    fideltatau = fideltatau + ni(i) * Ii(i) * delta ^ (Ii(i) - 1) *
Ji(i) * tau ^ (Ji(i) - 1);
    fideltadelta = fideltadelta + ni(i) * Ii(i) * (Ii(i) - 1) * delta
^ (Ii(i) - 2) * tau ^ Ji(i);
end
fidelta = fidelta + ni(1) / delta;
fideltadelta = fideltadelta - ni(1) / (delta ^ 2);
w3_rhoT = (1000 * R * T * (2 * delta * fidelta + delta ^ 2 *
fideltadelta - (delta * fidelta - delta * tau * fideltatau) ^ 2 / (tau
^ 2 * fitautau))) ^ 0.5;

function T3_ph = T3_ph(p, h)
%Revised Supplementary Release on Backward Equations for the functions
T(p,h), v(p,h) and T(p,s), v(p,s) for Region 3 of the IAPWS Industrial
formulation 1997 for the Thermodynamic Properties of Water and Steam
%2004
%Section 3.3 Backward Equations T(p,h) and v(p,h) for Subregions 3a
and 3b
%Boundary equation, Eq 1 Page 5
h3ab = 2014.64004206875 + 3.74696550136983 * p - 2.19921901054187E-02
* p ^ 2 + 8.7513168600995E-05 * p ^ 3;
if h < h3ab
    %Subregion 3a
    %Eq 2, Table 3, Page 7
    Ii = [-12, -12, -12, -12, -12, -12, -12, -12, -10, -10, -10, -8, -
8, -8, -8, -5, -3, -2, -2, -2, -1, -1, 0, 0, 1, 3, 3, 4, 4, 10, 12];
    Ji = [0, 1, 2, 6, 14, 16, 20, 22, 1, 5, 12, 0, 2, 4, 10, 2, 0, 1,
3, 4, 0, 2, 0, 1, 1, 0, 1, 0, 3, 4, 5];

```

```

    ni = [-1.33645667811215E-07, 4.55912656802978E-06, -
1.46294640700979E-05, 6.3934131297008E-03, 372.783927268847, -
7186.54377460447, 573494.7521034, -2675693.29111439, -
3.34066283302614E-05, -2.45479214069597E-02, 47.8087847764996,
7.64664131818904E-06, 1.28350627676972E-03, 1.71219081377331E-02, -
8.51007304583213, -1.36513461629781E-02, -3.84460997596657E-06,
3.37423807911655E-03, -0.551624873066791, 0.72920227710747, -
9.92522757376041E-03, -0.119308831407288, 0.793929190615421,
0.454270731799386, 0.20999859125991, -6.42109823904738E-03, -
0.023515586860454, 2.52233108341612E-03, -7.64885133368119E-03,
1.36176427574291E-02, -1.33027883575669E-02];
    ps = p / 100;
    hs = h / 2300;
    Ts = 0;
    for i = 1 : 31
        Ts = Ts + ni(i) * (ps + 0.24) ^ Ii(i) * (hs - 0.615) ^ Ji(i);
    end
    T3_ph = Ts * 760;
else
    %Subregion 3b
    %Eq 3, Table 4, Page 7,8
    Ii = [-12, -12, -10, -10, -10, -10, -10, -8, -8, -8, -8, -8, -6, -
6, -6, -4, -4, -3, -2, -2, -1, -1, -1, -1, -1, -1, -1, 0, 0, 1, 3, 5, 6,
8];
    Ji = [0, 1, 0, 1, 5, 10, 12, 0, 1, 2, 4, 10, 0, 1, 2, 0, 1, 5, 0,
4, 2, 4, 6, 10, 14, 16, 0, 2, 1, 1, 1, 1, 1];
    ni = [3.2325457364492E-05, -1.27575556587181E-04, -
4.75851877356068E-04, 1.56183014181602E-03, 0.105724860113781, -
85.8514221132534, 724.140095480911, 2.96475810273257E-03, -
5.92721983365988E-03, -1.26305422818666E-02, -0.115716196364853,
84.9000969739595, -1.08602260086615E-02, 1.54304475328851E-02,
7.50455441524466E-02, 2.52520973612982E-02, -6.02507901232996E-02, -
3.07622221350501, -5.74011959864879E-02, 5.03471360939849, -
0.925081888584834, 3.91733882917546, -77.314600713019,
9493.08762098587, -1410437.19679409, 8491662.30819026,
0.861095729446704, 0.32334644281172, 0.873281936020439, -
0.436653048526683, 0.286596714529479, -0.131778331276228,
6.76682064330275E-03];
    hs = h / 2800;
    ps = p / 100;
    Ts = 0;
    for i = 1 : 33
        Ts = Ts + ni(i) * (ps + 0.298) ^ Ii(i) * (hs - 0.72) ^ Ji(i);
    end
    T3_ph = Ts * 860;
end

function v3_ph = v3_ph(p, h)
%Revised Supplementary Release on Backward Equations for the functions
T(p,h), v(p,h) and T(p,s), v(p,s) for Region 3 of the IAPWS Industrial
formulation 1997 for the Thermodynamic Properties of Water and Steam
%2004
%Section 3.3 Backward Equations T(p,h) and v(p,h) for Subregions 3a
and 3b
%Boundary equation, Eq 1 Page 5
h3ab = 2014.64004206875 + 3.74696550136983 * p - 2.19921901054187E-02
* p ^ 2 + 8.7513168600995E-05 * p ^ 3;
if h < h3ab
    %Subregion 3a
    %Eq 4, Table 6, Page 9

```

```

    Ii = [-12, -12, -12, -12, -10, -10, -10, -8, -8, -6, -6, -6, -4, -
4, -3, -2, -2, -1, -1, -1, -1, 0, 0, 1, 1, 1, 2, 2, 3, 4, 5, 8];
    Ji = [6, 8, 12, 18, 4, 7, 10, 5, 12, 3, 4, 22, 2, 3, 7, 3, 16, 0,
1, 2, 3, 0, 1, 0, 1, 2, 0, 2, 0, 2, 2, 2];
    ni = [5.29944062966028E-03, -0.170099690234461, 11.1323814312927,
-2178.98123145125, -5.06061827980875E-04, 0.556495239685324, -
9.43672726094016, -0.297856807561527, 93.9353943717186,
1.92944939465981E-02, 0.421740664704763, -3689141.2628233, -
7.37566847600639E-03, -0.354753242424366, -1.99768169338727,
1.15456297059049, 5683.6687581596, 8.08169540124668E-03,
0.172416341519307, 1.04270175292927, -0.297691372792847,
0.560394465163593, 0.275234661176914, -0.148347894866012, -
6.51142513478515E-02, -2.92468715386302, 6.64876096952665E-02,
3.52335014263844, -1.46340792313332E-02, -2.24503486668184,
1.10533464706142, -4.08757344495612E-02];
    ps = p / 100;
    hs = h / 2100;
    vs = 0;
    for i = 1 : 32
        vs = vs + ni(i) * (ps + 0.128) ^ Ii(i) * (hs - 0.727) ^ Ji(i);
    end
    v3_ph = vs * 0.0028;
else
    %Subregion 3b
    %Eq 5, Table 7, Page 9
    Ii = [-12, -12, -8, -8, -8, -8, -8, -8, -6, -6, -6, -6, -6, -6, -
4, -4, -4, -3, -3, -2, -2, -1, -1, -1, -1, 0, 1, 1, 2, 2];
    Ji = [0, 1, 0, 1, 3, 6, 7, 8, 0, 1, 2, 5, 6, 10, 3, 6, 10, 0, 2,
1, 2, 0, 1, 4, 5, 0, 0, 1, 2, 6];
    ni = [-2.25196934336318E-09, 1.40674363313486E-08,
2.3378408528056E-06, -3.31833715229001E-05, 1.07956778514318E-03, -
0.271382067378863, 1.07202262490333, -0.853821329075382, -
2.15214194340526E-05, 7.6965608822273E-04, -4.31136580433864E-03,
0.453342167309331, -0.507749535873652, -100.475154528389, -
0.219201924648793, -3.21087965668917, 607.567815637771,
5.57686450685932E-04, 0.18749904002955, 9.05368030448107E-03,
0.285417173048685, 3.29924030996098E-02, 0.239897419685483,
4.82754995951394, -11.8035753702231, 0.169490044091791, -
1.79967222507787E-02, 3.71810116332674E-02, -5.36288335065096E-02,
1.6069710109252];
    ps = p / 100;
    hs = h / 2800;
    vs = 0;
    for i = 1 : 30
        vs = vs + ni(i) * (ps + 0.0661) ^ Ii(i) * (hs - 0.72) ^ Ji(i);
    end
    v3_ph = vs * 0.0088;
end

function T3_ps = T3_ps(p, s)
%Revised Supplementary Release on Backward Equations for the functions
T(p,h), v(p,h) and T(p,s), v(p,s) for Region 3 of the IAPWS Industrial
formulation 1997 for the Thermodynamic Properties of Water and Steam
%2004
%3.4 Backward Equations T(p,s) and v(p,s) for Subregions 3a and 3b
%Boundary equation, Eq 6 Page 11
if s <= 4.41202148223476
    %Subregion 3a
    %Eq 6, Table 10, Page 11
    Ii = [-12, -12, -10, -10, -10, -10, -8, -8, -8, -8, -6, -6, -6, -
5, -5, -5, -4, -4, -4, -2, -2, -1, -1, 0, 0, 0, 1, 2, 2, 3, 8, 8, 10];

```

```

        Ji = [28, 32, 4, 10, 12, 14, 5, 7, 8, 28, 2, 6, 32, 0, 14, 32, 6,
10, 36, 1, 4, 1, 6, 0, 1, 4, 0, 0, 3, 2, 0, 1, 2];
        ni = [1500420082.63875, -159397258480.424, 5.02181140217975E-04, -
67.2057767855466, 1450.58545404456, -8238.8953488889, -
0.154852214233853, 11.2305046746695, -29.7000213482822,
43856513263.5495, 1.37837838635464E-03, -2.97478527157462,
9717779473494.13, -5.71527767052398E-05, 28830.794977842, -
74442828926270.3, 12.8017324848921, -368.275545889071,
6.64768904779177E+15, 0.044935925195888, -4.22897836099655, -
0.240614376434179, -4.74341365254924, 0.72409399912611,
0.923874349695897, 3.99043655281015, 3.84066651868009E-02, -
3.59344365571848E-03, -0.735196448821653, 0.188367048396131,
1.41064266818704E-04, -2.57418501496337E-03, 1.23220024851555E-03];
        Sigma = s / 4.4;
        Pi = p / 100;
        teta = 0;
        for i = 1 : 33
            teta = teta + ni(i) * (Pi + 0.24) ^ Ii(i) * (Sigma - 0.703) ^
Ji(i);
        end
        T3_ps = teta * 760;
    else
        %Subregion 3b
        %Eq 7, Table 11, Page 11
        Ii = [-12, -12, -12, -12, -8, -8, -8, -8, -6, -6, -6, -5, -5, -5, -5,
-5, -4, -3, -3, -2, 0, 2, 3, 4, 5, 6, 8, 12, 14];
        Ji = [1, 3, 4, 7, 0, 1, 3, 0, 2, 4, 0, 1, 2, 4, 6, 12, 1, 6, 2, 0,
1, 1, 0, 24, 0, 3, 1, 2];
        ni = [0.527111701601666, -40.1317830052742, 153.020073134484, -
2247.99398218827, -0.193993484669048, -1.40467557893768,
42.6799878114024, 0.752810643416743, 22.6657238616417, -
622.873556909932, -0.660823667935396, 0.841267087271658, -
25.3717501764397, 485.708963532948, 880.531517490555,
2650155.92794626, -0.359287150025783, -656.991567673753,
2.41768149185367, 0.856873461222588, 0.655143675313458, -
0.213535213206406, 5.62974957606348E-03, -316955725450471, -
6.99997000152457E-04, 1.19845803210767E-02, 1.93848122022095E-05, -
2.15095749182309E-05];
        Sigma = s / 5.3;
        Pi = p / 100;
        teta = 0;
        for i = 1 : 28
            teta = teta + ni(i) * (Pi + 0.76) ^ Ii(i) * (Sigma - 0.818) ^
Ji(i);
        end
        T3_ps = teta * 860;
    end

function v3_ps = v3_ps(p, s)
%Revised Supplementary Release on Backward Equations for the functions
T(p,h), v(p,h) and T(p,s), v(p,s) for Region 3 of the IAPWS Industrial
formulation 1997 for the Thermodynamic Properties of Water and Steam
%2004
%3.4 Backward Equations T(p,s) and v(p,s) for Subregions 3a and 3b
%Boundary equation, Eq 6 Page 11
if s <= 4.41202148223476
    %Subregion 3a
    %Eq 8, Table 13, Page 14
    Ii = [-12, -12, -12, -10, -10, -10, -10, -8, -8, -8, -8, -6, -5, -
4, -3, -3, -2, -2, -1, -1, 0, 0, 0, 1, 2, 4, 5, 6];

```

```

    Ji = [10, 12, 14, 4, 8, 10, 20, 5, 6, 14, 16, 28, 1, 5, 2, 4, 3,
8, 1, 2, 0, 1, 3, 0, 0, 2, 2, 0];
    ni = [79.5544074093975, -2382.6124298459, 17681.3100617787, -
1.10524727080379E-03, -15.3213833655326, 297.544599376982, -
35031520.6871242, 0.277513761062119, -0.523964271036888, -
148011.182995403, 1600148.99374266, 1708023226634.27,
2.46866996006494E-04, 1.6532608479798, -0.118008384666987,
2.537986423559, 0.965127704669424, -28.2172420532826,
0.203224612353823, 1.10648186063513, 0.52612794845128,
0.277000018736321, 1.08153340501132, -7.44127885357893E-02,
1.64094443541384E-02, -6.80468275301065E-02, 0.025798857610164, -
1.45749861944416E-04];
    Pi = p / 100;
    Sigma = s / 4.4;
    omega = 0;
    for i = 1 : 28
        omega = omega + ni(i) * (Pi + 0.187) ^ Ii(i) * (Sigma - 0.755)
    ^ Ji(i);
    end
    v3_ps = omega * 0.0028;
else
    %Subregion 3b
    %Eq 9, Table 14, Page 14
    Ii = [-12, -12, -12, -12, -12, -12, -12, -10, -10, -10, -10, -8, -5, -
5, -5, -4, -4, -4, -4, -3, -2, -2, -2, -2, -2, -2, 0, 0, 0, 1, 1, 2];
    Ji = [0, 1, 2, 3, 5, 6, 0, 1, 2, 4, 0, 1, 2, 3, 0, 1, 2, 3, 1, 0,
1, 2, 3, 4, 12, 0, 1, 2, 0, 2, 2];
    ni = [5.91599780322238E-05, -1.85465997137856E-03,
1.04190510480013E-02, 5.9864730203859E-03, -0.771391189901699,
1.72549765557036, -4.67076079846526E-04, 1.34533823384439E-02, -
8.08094336805495E-02, 0.508139374365767, 1.28584643361683E-03, -
1.63899353915435, 5.86938199318063, -2.92466667918613, -
6.14076301499537E-03, 5.76199014049172, -12.1613320606788,
1.67637540957944, -7.44135838773463, 3.78168091437659E-02,
4.01432203027688, 16.0279837479185, 3.17848779347728, -
3.58362310304853, -1159952.60446827, 0.199256573577909, -
0.122270624794624, -19.1449143716586, -1.50448002905284E-02,
14.6407900162154, -3.2747778718823];
    Pi = p / 100;
    Sigma = s / 5.3;
    omega = 0;
    for i = 1 : 31
        omega = omega + ni(i) * (Pi + 0.298) ^ Ii(i) * (Sigma - 0.816)
    ^ Ji(i);
    end
    v3_ps = omega * 0.0088;
end

```

```

function p3_hs = p3_hs(h, s)
%Supplementary Release on Backward Equations ( ) , p h s for Region 3,
%Equations as a function of h and s for the Region Boundaries, and an
Equation
%( ) sat , T hs for Region 4 of the IAPWS Industrial formulation 1997
for the
%Thermodynamic Properties of Water and Steam
%2004
%Section 3 Backward functions p(h,s), T(h,s), and v(h,s) for Region 3
if s < 4.41202148223476
    %Subregion 3a
    %Eq 1, Table 3, Page 8

```



```

        Ii = [0, 0, 0, 1, 1, 1, 1, 1, 2, 2, 3, 3, 3, 4, 4, 4, 4, 5, 6, 7,
8, 10, 10, 14, 18, 20, 22, 22, 24, 28, 28, 32, 32];
        Ji = [0, 1, 5, 0, 3, 4, 8, 14, 6, 16, 0, 2, 3, 0, 1, 4, 5, 28, 28,
24, 1, 32, 36, 22, 28, 36, 16, 28, 36, 16, 36, 10, 28];
        ni = [7.70889828326934, -26.0835009128688, 267.416218930389,
17.2221089496844, -293.54233214597, 614.135601882478, -
61056.2757725674, -65127225.1118219, 73591.9313521937, -
11664650591.4191, 35.5267086434461, -596.144543825955, -
475.842430145708, 69.6781965359503, 335.674250377312,
25052.6809130882, 146997.380630766, 5.38069315091534E+19,
1.43619827291346E+21, 3.64985866165994E+19, -2547.41561156775,
2.40120197096563E+27, -3.93847464679496E+29, 1.47073407024852E+24, -
4.26391250432059E+31, 1.94509340621077E+38, 6.66212132114896E+23,
7.06777016552858E+33, 1.75563621975576E+41, 1.08408607429124E+28,
7.30872705175151E+43, 1.5914584739887E+24, 3.77121605943324E+40];
        Sigma = s / 4.4;
        eta = h / 2300;
        Pi = 0;
        for i = 1 : 33
            Pi = Pi + ni(i) * (eta - 1.01) ^ Ii(i) * (Sigma - 0.75) ^
Ji(i);
        end
        p3_hs = Pi * 99;
else
    %Subregion 3b
    %Eq 2, Table 4, Page 8
    Ii = [-12, -12, -12, -12, -12, -10, -10, -10, -10, -8, -8, -6, -6,
-6, -6, -5, -4, -4, -4, -3, -3, -3, -3, -2, -2, -1, 0, 2, 2, 5, 6, 8,
10, 14, 14];
    Ji = [2, 10, 12, 14, 20, 2, 10, 14, 18, 2, 8, 2, 6, 7, 8, 10, 4,
5, 8, 1, 3, 5, 6, 0, 1, 0, 3, 0, 1, 0, 1, 1, 1, 3, 7];
    ni = [1.25244360717979E-13, -1.26599322553713E-02,
5.06878030140626, 31.7847171154202, -391041.161399932, -
9.75733406392044E-11, -18.6312419488279, 510.973543414101,
373847.005822362, 2.99804024666572E-08, 20.0544393820342, -
4.98030487662829E-06, -10.230180636003, 55.2819126990325, -
206.211367510878, -7940.12232324823, 7.82248472028153, -
58.6544326902468, 3550.73647696481, -1.15303107290162E-04, -
1.75092403171802, 257.98168774816, -727.048374179467,
1.21644822609198E-04, 3.93137871762692E-02, 7.04181005909296E-03, -
82.910820069811, -0.26517881813125, 13.7531682453991, -
52.2394090753046, 2405.56298941048, -22736.1631268929,
89074.6343932567, -23923456.5822486, 5687958081.29714];
    Sigma = s / 5.3;
    eta = h / 2800;
    Pi = 0;
    for i = 1 : 35
        Pi = Pi + ni(i) * (eta - 0.681) ^ Ii(i) * (Sigma - 0.792) ^
Ji(i);
    end
    p3_hs = 16.6 / Pi;
end

function h3_pT = h3_pT(p, T)
%Not available with if 97
%Solve function T3_ph-T=0 with half interval method.
%ver2.6 Start corrected bug
if p < 22.06395 %Bellow tripple point
    Ts = T4_p(p); %Saturation temperature
    if T <= Ts %Liquid side
        High_Bound = h4L_p(p); %Max h är liauid h.
    end
end

```

```

        Low_Bound = h1_pT(p, 623.15);
    else
        Low_Bound = h4V_p(p); %Min h är Vapour h.
        High_Bound = h2_pT(p, B23T_p(p));
    end
else %Above tripple point. R3 from R2 till R3.
    Low_Bound = h1_pT(p, 623.15);
    High_Bound = h2_pT(p, B23T_p(p));
end
%ver2.6 End corrected bug
Ts = T+1;
while abs(T - Ts) > 0.00001
    hs = (Low_Bound + High_Bound) / 2;
    Ts = T3_ph(p, hs);
    if Ts > T
        High_Bound = hs;
    else
        Low_Bound = hs;
    end
end
h3_pT = hs;

function T3_prho = T3_prho(p, rho)
    %Solve by iteration. Observe that fo low temperatures this equation
    has 2 solutions.
    %Solve with half interval method
    Low_Bound = 623.15;
    High_Bound = 1073.15;
    ps=-1000;
    while abs(p - ps) > 0.00000001
        Ts = (Low_Bound + High_Bound) / 2;
        ps = p3_rhoT(rho, Ts);
        if ps > p
            High_Bound = Ts;
        else
            Low_Bound = Ts;
        end
    end
    T3_prho = Ts;

%*****
%*****
%*2.4 functions for region 4
function p4_T = p4_T(T)
%Release on the IAPWS Industrial formulation 1997 for the
Thermodynamic Properties of Water and Steam, September 1997
%Section 8.1 The Saturation-Pressure Equation
%Eq 30, Page 33
teta = T - 0.23855557567849 / (T - 650.17534844798);
a = teta ^ 2 + 1167.0521452767 * teta - 724213.16703206;
B = -17.073846940092 * teta ^ 2 + 12020.82470247 * teta -
3232555.0322333;
C = 14.91510861353 * teta ^ 2 - 4823.2657361591 * teta +
405113.40542057;
p4_T = (2 * C / (-B + (B ^ 2 - 4 * a * C) ^ 0.5)) ^ 4;

function T4_p = T4_p(p)

```

```

%Release on the IAPWS Industrial formulation 1997 for the
Thermodynamic Properties of Water and Steam, September 1997
%Section 8.2 The Saturation-Temperature Equation
%Eq 31, Page 34
beta = p ^ 0.25;
E = beta ^ 2 - 17.073846940092 * beta + 14.91510861353;
f = 1167.0521452767 * beta ^ 2 + 12020.82470247 * beta -
4823.2657361591;
G = -724213.16703206 * beta ^ 2 - 3232555.0322333 * beta +
405113.40542057;
D = 2 * G / (-f - (f ^ 2 - 4 * E * G) ^ 0.5);
T4_p = (650.17534844798 + D - ((650.17534844798 + D) ^ 2 - 4 * (-
0.23855557567849 + 650.17534844798 * D)) ^ 0.5) / 2;

function h4_s = h4_s(s)
%Supplementary Release on Backward Equations ( ) , p h s for Region
3,Equations as a function of h and s for the Region Boundaries, and an
Equation( ) sat , T hs for Region 4 of the IAPWS Industrial
formulation 1997 for the Thermodynamic Properties of Water and Steam
%4 Equations for Region Boundaries Given Enthalpy and Entropy
% See picture page 14
if (s > -0.0001545495919 & s <= 3.77828134)==1
    %hL1_s
    %Eq 3,Table 9,Page 16
    Ii = [0, 0, 1, 1, 2, 2, 3, 3, 4, 4, 4, 5, 5, 7, 8, 12, 12, 14, 14,
16, 20, 20, 22, 24, 28, 32, 32];
    Ji = [14, 36, 3, 16, 0, 5, 4, 36, 4, 16, 24, 18, 24, 1, 4, 2, 4,
1, 22, 10, 12, 28, 8, 3, 0, 6, 8];
    ni = [0.332171191705237, 6.11217706323496E-04, -8.82092478906822,
-0.45562819254325, -2.63483840850452E-05, -22.3949661148062, -
4.28398660164013, -0.616679338856916, -14.682303110404,
284.523138727299, -113.398503195444, 1156.71380760859,
395.551267359325, -1.54891257229285, 19.4486637751291, -
3.57915139457043, -3.35369414148819, -0.66442679633246,
32332.1885383934, 3317.66744667084, -22350.1257931087,
5739538.75852936, 173.226193407919, -3.63968822121321E-02,
8.34596332878346E-07, 5.03611916682674, 65.5444787064505];
    Sigma = s / 3.8;
    eta = 0;
    for i = 1 : 27
        eta = eta + ni(i) * (Sigma - 1.09) ^ Ii(i) * (Sigma +
0.0000366) ^ Ji(i);
    end
    h4_s = eta * 1700;
elseif (s > 3.77828134 & s <= 4.41202148223476 )==1
    %hL3_s
    %Eq 4,Table 10,Page 16
    Ii = [0, 0, 0, 0, 2, 3, 4, 4, 5, 5, 6, 7, 7, 7, 10, 10, 10, 32,
32];
    Ji = [1, 4, 10, 16, 1, 36, 3, 16, 20, 36, 4, 2, 28, 32, 14, 32,
36, 0, 6];
    ni = [0.822673364673336, 0.181977213534479, -0.011200026031362, -
7.46778287048033E-04, -0.179046263257381, 4.24220110836657E-02, -
0.341355823438768, -2.09881740853565, -8.22477343323596, -
4.99684082076008, 0.191413958471069, 5.81062241093136E-02, -
1655.05498701029, 1588.70443421201, -85.0623535172818, -
31771.4386511207, -94589.0406632871, -1.3927384708869E-06,
0.63105253224098];
    Sigma = s / 3.8;
    eta = 0;
    for i = 1 : 19

```

```

        eta = eta + ni(i) * (Sigma - 1.09) ^ Ii(i) * (Sigma +
0.0000366) ^ Ji(i);
    end
    h4_s = eta * 1700;
elseif (s > 4.41202148223476 & s <= 5.85 )==1
    %Section 4.4 Equations ( ) 2ab " h s and ( ) 2c3b "h s for the
Saturated Vapor Line
    %Page 19, Eq 5
    %hV2c3b_s(s)
    Ii = [0, 0, 0, 1, 1, 5, 6, 7, 8, 8, 12, 16, 22, 22, 24, 36];
    Ji = [0, 3, 4, 0, 12, 36, 12, 16, 2, 20, 32, 36, 2, 32, 7, 20];
    ni = [1.04351280732769, -2.27807912708513, 1.80535256723202,
0.420440834792042, -105721.24483466, 4.36911607493884E+24, -
328032702839.753, -6.7868676080427E+15, 7439.57464645363, -
3.56896445355761E+19, 1.67590585186801E+31, -3.55028625419105E+37,
396611982166.538, -4.14716268484468E+40, 3.59080103867382E+18, -
1.16994334851995E+40];
    Sigma = s / 5.9;
    eta = 0;
    for i = 1 : 16
        eta = eta + ni(i) * (Sigma - 1.02) ^ Ii(i) * (Sigma - 0.726) ^
Ji(i);
    end
    h4_s = eta ^ 4 * 2800;
elseif (s > 5.85 & s < 9.155759395)==1
    %Section 4.4 Equations ( ) 2ab " h s and ( ) 2c3b "h s for the
Saturated Vapor Line
    %Page 20, Eq 6
    Ii = [1, 1, 2, 2, 4, 4, 7, 8, 8, 10, 12, 12, 18, 20, 24, 28, 28,
28, 28, 28, 32, 32, 32, 32, 32, 36, 36, 36, 36, 36];
    Ji = [8, 24, 4, 32, 1, 2, 7, 5, 12, 1, 0, 7, 10, 12, 32, 8, 12,
20, 22, 24, 2, 7, 12, 14, 24, 10, 12, 20, 22, 28];
    ni = [-524.581170928788, -9269472.18142218, -237.385107491666,
21077015581.2776, -23.9494562010986, 221.802480294197, -
5104725.33393438, 1249813.96109147, 2000084369.96201, -
815.158509791035, -157.612685637523, -11420042233.2791,
6.62364680776872E+15, -2.27622818296144E+18, -1.71048081348406E+31,
6.60788766938091E+15, 1.66320055886021E+22, -2.18003784381501E+29, -
7.87276140295618E+29, 1.51062329700346E+31, 7957321.70300541,
1.31957647355347E+15, -3.2509706829914E+23, -4.18600611419248E+25,
2.97478906557467E+34, -9.53588761745473E+19, 1.66957699620939E+24, -
1.75407764869978E+32, 3.47581490626396E+34, -7.10971318427851E+38];
    Sigma1 = s / 5.21;
    Sigma2 = s / 9.2;
    eta = 0;
    for i = 1 : 30
        eta = eta + ni(i) * (1 / Sigma1 - 0.513) ^ Ii(i) * (Sigma2 -
0.524) ^ Ji(i);
    end
    h4_s = exp(eta) * 2800;
else
    h4_s = -99999;
end

function p4_s = p4_s(s)
%Uses h4_s and p_hs for the different regions to determine p4_s
h_sat = h4_s(s);
if (s > -0.0001545495919 & s <= 3.77828134)==1
    p4_s = p1_hs(h_sat, s);
elseif (s > 3.77828134 & s <= 5.210887663)==1
    p4_s = p3_hs(h_sat, s);

```

```

elseif (s > 5.210887663 & s < 9.155759395)==1
    p4_s = p2_hs(h_sat, s);
else
    p4_s = -99999;
end

function h4L_p = h4L_p(p)
if (p > 0.000611657 & p < 22.06395)==1
    Ts = T4_p(p);
    if p < 16.529
        h4L_p = h1_pT(p, Ts);
    else
        %Iterate to find the the backward solution of p3sat_h
        Low_Bound = 1670.858218;
        High_Bound = 2087.23500164864;
        ps=-1000;
        while abs(p - ps) > 0.00001
            hs = (Low_Bound + High_Bound) / 2;
            ps = p3sat_h(hs);
            if ps > p
                High_Bound = hs;
            else
                Low_Bound = hs;
            end
        end
        h4L_p = hs;
    end
else
    h4L_p = -99999;
end

function h4V_p = h4V_p(p)
if (p > 0.000611657 & p < 22.06395)==1
    Ts = T4_p(p);
    if p < 16.529
        h4V_p = h2_pT(p, Ts);
    else
        %Iterate to find the the backward solution of p3sat_h
        Low_Bound = 2087.23500164864;
        High_Bound = 2563.592004+5;
        ps=-1000;
        while abs(p - ps) > 0.000001
            hs = (Low_Bound + High_Bound) / 2;
            ps = p3sat_h(hs);
            if ps < p
                High_Bound = hs;
            else
                Low_Bound = hs;
            end
        end
        h4V_p = hs;
    end
else
    h4V_p = -99999;
end

function x4_ph = x4_ph(p, h)
%Calculate vapour fraction from hL and hV for given p
h4v = h4V_p(p);
h4L = h4L_p(p);

```

```

if h > h4v
    x4_ph = 1;
elseif h < h4L
    x4_ph = 0;
else
    x4_ph = (h - h4L) / (h4v - h4L);
end

function x4_ps = x4_ps(p, s)
if p < 16.529
    ssv = s2_pT(p, T4_p(p));
    ssL = s1_pT(p, T4_p(p));
else
    ssv = s3_rhoT(1 / (v3_ph(p, h4V_p(p))), T4_p(p));
    ssL = s3_rhoT(1 / (v3_ph(p, h4L_p(p))), T4_p(p));
end
if s < ssL
    x4_ps = 0;
elseif s > ssv
    x4_ps = 1;
else
    x4_ps = (s - ssL) / (ssv - ssL);
end

function T4_hs = T4_hs(h, s)
%Supplementary Release on Backward Equations ( ) , p h s for Region 3,
%Chapter 5.3 page 30.
%The if 97 function is only valid for part of region4. Use iteration
outsida.
Ii = [0, 0, 0, 1, 1, 1, 1, 2, 2, 2, 3, 3, 3, 3, 4, 4, 5, 5, 5, 5, 6,
6, 6, 8, 10, 10, 12, 14, 14, 16, 16, 18, 18, 18, 20, 28];
Ji = [0, 3, 12, 0, 1, 2, 5, 0, 5, 8, 0, 2, 3, 4, 0, 1, 1, 2, 4, 16, 6,
8, 22, 1, 20, 36, 24, 1, 28, 12, 32, 14, 22, 36, 24, 36];
ni = [0.179882673606601, -0.267507455199603, 1.162767226126,
0.147545428713616, -0.512871635973248, 0.421333567697984,
0.56374952218987, 0.429274443819153, -3.3570455214214,
10.8890916499278, -0.248483390456012, 0.30415322190639, -
0.494819763939905, 1.07551674933261, 7.33888415457688E-02,
1.40170545411085E-02, -0.106110975998808, 1.68324361811875E-02,
1.25028363714877, 1013.16840309509, -1.51791558000712,
52.4277865990866, 23049.5545563912, 2.49459806365456E-02,
2107964.67412137, 366836848.613065, -144814105.365163, -
1.7927637300359E-03, 4899556021.00459, 471.262212070518, -
82929439019.8652, -1715.45662263191, 3557776.82973575,
586062760258.436, -12988763.5078195, 31724744937.1057];
if (s > 5.210887825 & s < 9.15546555571324)==1
    Sigma = s / 9.2;
    eta = h / 2800;
    teta = 0;
    for i = 1 : 36
        teta = teta + ni(i) * (eta - 0.119) ^ Ii(i) * (Sigma - 1.07) ^
Ji(i);
    end
    T4_hs = teta * 550;
else
    %function psat_h
    if (s > -0.0001545495919 & s <= 3.77828134)==1
        Low_Bound = 0.000611;
        High_Bound = 165.291642526045;
        hL=-1000;

```

```

        while abs(hL - h) > 0.00001 & abs(High_Bound - Low_Bound) >
0.0001
            PL = (Low_Bound + High_Bound) / 2;
            Ts = T4_p(PL);
            hL = h1_pT(PL, Ts);
            if hL > h
                High_Bound = PL;
            else
                Low_Bound = PL;
            end
        end
    elseif s > 3.77828134 & s <= 4.41202148223476
        PL = p3sat_h(h);
    elseif s > 4.41202148223476 & s <= 5.210887663
        PL = p3sat_h(h);
    end
    Low_Bound = 0.000611;
    High_Bound = PL;
    sss=-1000;
    while (abs(s - sss) > 0.000001 & abs(High_Bound - Low_Bound) >
0.0000001)==1
        p = (Low_Bound + High_Bound) / 2;
        %Calculate s4_ph
        Ts = T4_p(p);
        xs = x4_ph(p, h);
        if p < 16.529
            s4v = s2_pT(p, Ts);
            s4L = s1_pT(p, Ts);
        else
            v4v = v3_ph(p, h4V_p(p));
            s4v = s3_rhoT(1 / v4v, Ts);
            v4L = v3_ph(p, h4L_p(p));
            s4L = s3_rhoT(1 / v4L, Ts);
        end
        sss = (xs * s4v + (1 - xs) * s4L);

        if sss < s
            High_Bound = p;
        else
            Low_Bound = p;
        end
    end
    T4_hs = T4_p(p);
end
*****
*****
%*2.5 functions for region 5
function h5_pT = h5_pT(p, T)
%Release on the IAPWS Industrial formulation 1997 for the
Thermodynamic Properties of Water and Steam, September 1997
%Basic Equation for Region 5
%Eq 32,33, Page 36, Tables 37-41
Ji0 = [0, 1, -3, -2, -1, 2];
ni0 = [-13.179983674201, 6.8540841634434, -0.024805148933466,
0.36901534980333, -3.1161318213925, -0.32961626538917];
Iir = [1, 1, 1, 2, 3];
Jir = [0, 1, 3, 9, 3];
nir = [-1.2563183589592E-04, 2.1774678714571E-03, -0.004594282089991,
-3.9724828359569E-06, 1.2919228289784E-07];
R = 0.461526; %kJ/(kg K)
tau = 1000 / T;

```

```

Pi = p;
gamma0_tau = 0;
for i = 1 : 6
    gamma0_tau = gamma0_tau + ni0(i) * Ji0(i) * tau ^ (Ji0(i) - 1);
end
gammar_tau = 0;
for i = 1 : 5
    gammar_tau = gammar_tau + nir(i) * Pi ^ Iir(i) * Jir(i) * tau ^
(Jir(i) - 1);
end
h5_pT = R * T * tau * (gamma0_tau + gammar_tau);

function v5_pT = v5_pT(p, T)
%Release on the IAPWS Industrial formulation 1997 for the
Thermodynamic Properties of Water and Steam, September 1997
%Basic Equation for Region 5
%Eq 32,33, Page 36, Tables 37-41
Ji0 = [0, 1, -3, -2, -1, 2];
ni0 = [-13.179983674201, 6.8540841634434, -0.024805148933466,
0.36901534980333, -3.1161318213925, -0.32961626538917];
Iir = [1, 1, 1, 2, 3];
Jir = [0, 1, 3, 9, 3];
nir = [-1.2563183589592E-04, 2.1774678714571E-03, -0.004594282089991,
-3.9724828359569E-06, 1.2919228289784E-07];
R = 0.461526; %kJ/(kg K)
tau = 1000 / T;
Pi = p;
gamma0_pi = 1 / Pi;
gammar_pi = 0;
for i = 1 : 5
    gammar_pi = gammar_pi + nir(i) * Iir(i) * Pi ^ (Iir(i) - 1) * tau
^ Jir(i);
end
v5_pT = R * T / p * Pi * (gamma0_pi + gammar_pi) / 1000;

function u5_pT = u5_pT(p, T)
%Release on the IAPWS Industrial formulation 1997 for the
Thermodynamic Properties of Water and Steam, September 1997
%Basic Equation for Region 5
%Eq 32,33, Page 36, Tables 37-41
Ji0 = [0, 1, -3, -2, -1, 2];
ni0 = [-13.179983674201, 6.8540841634434, -0.024805148933466,
0.36901534980333, -3.1161318213925, -0.32961626538917];
Iir = [1, 1, 1, 2, 3];
Jir = [0, 1, 3, 9, 3];
nir = [-1.2563183589592E-04, 2.1774678714571E-03, -0.004594282089991,
-3.9724828359569E-06, 1.2919228289784E-07];
R = 0.461526; %kJ/(kg K)
tau = 1000 / T;
Pi = p;
gamma0_pi = 1 / Pi;
gamma0_tau = 0;
for i = 1 : 6
    gamma0_tau = gamma0_tau + ni0(i) * Ji0(i) * tau ^ (Ji0(i) - 1);
end
gammar_pi = 0;
gammar_tau = 0;

```



```

for i = 1 : 5
    gammar_pi = gammar_pi + nir(i) * Iir(i) * Pi ^ (Iir(i) - 1) * tau
    ^ Jir(i);
    gammar_tau = gammar_tau + nir(i) * Pi ^ Iir(i) * Jir(i) * tau ^
    (Jir(i) - 1);
end
u5_pT = R * T * (tau * (gamma0_tau + gammar_tau) - Pi * (gamma0_pi +
gammar_pi));

```

```

function Cp5_pT = Cp5_pT(p, T)
%Release on the IAPWS Industrial formulation 1997 for the
Thermodynamic Properties of Water and Steam, September 1997
%Basic Equation for Region 5
%Eq 32,33, Page 36, Tables 37-41
Ji0 = [0, 1, -3, -2, -1, 2];
ni0 = [-13.179983674201, 6.8540841634434, -0.024805148933466,
0.36901534980333, -3.1161318213925, -0.32961626538917];
Iir = [1, 1, 1, 2, 3];
Jir = [0, 1, 3, 9, 3];
nir = [-1.2563183589592E-04, 2.1774678714571E-03, -0.004594282089991,
-3.9724828359569E-06, 1.2919228289784E-07];
R = 0.461526; %kJ/(kg K)
tau = 1000 / T;
Pi = p;
gamma0_tautau = 0;
for i = 1 : 6
    gamma0_tautau = gamma0_tautau + ni0(i) * Ji0(i) * (Ji0(i) - 1) *
tau ^ (Ji0(i) - 2);
end
gammar_tautau = 0;
for i = 1 : 5
    gammar_tautau = gammar_tautau + nir(i) * Pi ^ Iir(i) * Jir(i) *
(Jir(i) - 1) * tau ^ (Jir(i) - 2);
end
Cp5_pT = -R * tau ^ 2 * (gamma0_tautau + gammar_tautau);

```

```

function s5_pT = s5_pT(p, T)
%Release on the IAPWS Industrial formulation 1997 for the
Thermodynamic Properties of Water and Steam, September 1997
%Basic Equation for Region 5
%Eq 32,33, Page 36, Tables 37-41
Ji0 = [0, 1, -3, -2, -1, 2];
ni0 = [-13.179983674201, 6.8540841634434, -0.024805148933466,
0.36901534980333, -3.1161318213925, -0.32961626538917];
Iir = [1, 1, 1, 2, 3];
Jir = [0, 1, 3, 9, 3];
nir = [-1.2563183589592E-04, 2.1774678714571E-03, -0.004594282089991,
-3.9724828359569E-06, 1.2919228289784E-07];
R = 0.461526; %kJ/(kg K)
tau = 1000 / T;
Pi = p;
gamma0 = log(Pi);
gamma0_tau = 0;
for i = 1 : 6
    gamma0_tau = gamma0_tau + ni0(i) * Ji0(i) * tau ^ (Ji0(i) - 1);
    gamma0 = gamma0 + ni0(i) * tau ^ Ji0(i);
end
gammar = 0;
gammar_tau = 0;
for i = 1 : 5

```

```

    gammar = gammar + nir(i) * Pi ^ Iir(i) * tau ^ Jir(i);
    gammar_tau = gammar_tau + nir(i) * Pi ^ Iir(i) * Jir(i) * tau ^
(Jir(i) - 1);
end
s5_pT = R * (tau * (gamma0_tau + gammar_tau) - (gamma0 + gammar));

function Cv5_pT = Cv5_pT(p, T)
%Release on the IAPWS Industrial formulation 1997 for the
Thermodynamic Properties of Water and Steam, September 1997
%Basic Equation for Region 5
%Eq 32,33, Page 36, Tables 37-41
Ji0 = [0, 1, -3, -2, -1, 2];
ni0 = [-13.179983674201, 6.8540841634434, -0.024805148933466,
0.36901534980333, -3.1161318213925, -0.32961626538917];
Iir = [1, 1, 1, 2, 3];
Jir = [0, 1, 3, 9, 3];
nir = [-1.2563183589592E-04, 2.1774678714571E-03, -0.004594282089991,
-3.9724828359569E-06, 1.2919228289784E-07];
R = 0.461526; %kJ/(kg K)
tau = 1000 / T;
Pi = p;
gamma0_tautau = 0;
for i = 1 : 6
    gamma0_tautau = gamma0_tautau + ni0(i) * (Ji0(i) - 1) * Ji0(i) *
tau ^ (Ji0(i) - 2);
end
gammar_pi = 0;
gammar_pitau = 0;
gammar_pipi = 0;
gammar_tautau = 0;
for i = 1 : 5
    gammar_pi = gammar_pi + nir(i) * Iir(i) * Pi ^ (Iir(i) - 1) * tau
^ Jir(i);
    gammar_pitau = gammar_pitau + nir(i) * Iir(i) * Pi ^ (Iir(i) - 1)
* Jir(i) * tau ^ (Jir(i) - 1);
    gammar_pipi = gammar_pipi + nir(i) * Iir(i) * (Iir(i) - 1) * Pi ^
(Iir(i) - 2) * tau ^ Jir(i);
    gammar_tautau = gammar_tautau + nir(i) * Pi ^ Iir(i) * Jir(i) *
(Jir(i) - 1) * tau ^ (Jir(i) - 2);
end
Cv5_pT = R * (-(tau ^ 2 * (gamma0_tautau + gammar_tautau)) - (1 + Pi *
gammar_pi - tau * Pi * gammar_pitau)^2 / (1 - Pi ^ 2 * gammar_pipi));

function w5_pT = w5_pT(p, T)
%Release on the IAPWS Industrial formulation 1997 for the
Thermodynamic Properties of Water and Steam, September 1997
%Basic Equation for Region 5
%Eq 32,33, Page 36, Tables 37-41
Ji0 = [0, 1, -3, -2, -1, 2];
ni0 = [-13.179983674201, 6.8540841634434, -0.024805148933466,
0.36901534980333, -3.1161318213925, -0.32961626538917];
Iir = [1, 1, 1, 2, 3];
Jir = [0, 1, 3, 9, 3];
nir = [-1.2563183589592E-04, 2.1774678714571E-03, -0.004594282089991,
-3.9724828359569E-06, 1.2919228289784E-07];
R = 0.461526; %kJ/(kg K)
tau = 1000 / T;
Pi = p;
gamma0_tautau = 0;
for i = 1 : 6

```

```

    gamma0_tautau = gamma0_tautau + ni0(i) * (Ji0(i) - 1) * Ji0(i) *
tau ^ (Ji0(i) - 2);
end
gammar_pi = 0;
gammar_pitau = 0;
gammar_pipi = 0;
gammar_tautau = 0;
for i = 1 : 5
    gammar_pi = gammar_pi + nir(i) * Iir(i) * Pi ^ (Iir(i) - 1) * tau
^ Jir(i);
    gammar_pitau = gammar_pitau + nir(i) * Iir(i) * Pi ^ (Iir(i) - 1)
* Jir(i) * tau ^ (Jir(i) - 1);
    gammar_pipi = gammar_pipi + nir(i) * Iir(i) * (Iir(i) - 1) * Pi ^
(Iir(i) - 2) * tau ^ Jir(i);
    gammar_tautau = gammar_tautau + nir(i) * Pi ^ Iir(i) * Jir(i) *
(Jir(i) - 1) * tau ^ (Jir(i) - 2);
end
w5_pT = (1000 * R * T * (1 + 2 * Pi * gammar_pi + Pi ^ 2 * gammar_pi ^
2) / ((1 - Pi ^ 2 * gammar_pipi) + (1 + Pi * gammar_pi - tau * Pi *
gammar_pitau) ^ 2 / (tau ^ 2 * (gamma0_tautau + gammar_tautau)))) ^
0.5;

function T5_ph = T5_ph(p, h)
%Solve with half interval method
Low_Bound = 1073.15;
High_Bound = 2273.15;
hs=h-1;
while abs(h - hs) > 0.00001
    Ts = (Low_Bound + High_Bound) / 2;
    hs = h5_pT(p, Ts);
    if hs > h
        High_Bound = Ts;
    else
        Low_Bound = Ts;
    end
end
T5_ph = Ts;

function T5_ps = T5_ps(p, s)
%Solve with half interval method
Low_Bound = 1073.15;
High_Bound = 2273.15;
ss=s-1;
while abs(s - ss) > 0.00001
    Ts = (Low_Bound + High_Bound) / 2;
    ss = s5_pT(p, Ts);
    if ss > s
        High_Bound = Ts;
    else
        Low_Bound = Ts;
    end
end
T5_ps = Ts;

function T5_prho=T5_prho(p, rho)
%Solve by iteration. Observe that fo low temperatures this equation
has 2 solutions.
%Solve with half interval method

```

```

    Low_Bound = 1073.15;
    High_Bound = 2073.15;
    rhos=-1000;
    while abs(rho - rhos) > 0.000001
        Ts = (Low_Bound + High_Bound) / 2;
        rhos = 1 / v2_pT(p, Ts);
        if rhos < rho
            High_Bound = Ts;
        else
            Low_Bound = Ts;
        end
    end
    T5_prho = Ts;

%*****
%*****
%*3 Region Selection
%*****
%*****
%*3.1 Regions as a function of pT
function region_pT = region_pT(p, T)
if T > 1073.15 & p < 10 & T < 2273.15 & p > 0.000611
    region_pT = 5;
elseif T <= 1073.15 & T > 273.15 & p <= 100 & p > 0.000611
    if T > 623.15
        if p > B23p_T(T)
            region_pT = 3;
            if T < 647.096
                ps = p4_T(T);
                if abs(p - ps) < 0.00001
                    region_pT = 4;
                end
            end
        else
            region_pT = 2;
        end
    else
        ps = p4_T(T);
        if abs(p - ps) < 0.00001
            region_pT = 4;
        elseif p > ps
            region_pT = 1;
        else
            region_pT = 2;
        end
    end
end
else
    region_pT = 0; %**Error, Outside valid area
end
%*****
%*****
%*3.2 Regions as a function of ph
function region_ph = region_ph( p, h)
%Check if outside pressure limits
if p < 0.000611657 | p > 100
    region_ph = 0;
    return
end

%Check if outside low h.

```

```

if h < 0.963 * p + 2.2 %Linear adaption to h1_pt()+2 to speed up
calculations.
    if h < h1_pT(p, 273.15)
        region_ph = 0;
        return
    end
end

if p < 16.5292 %Bellow region 3,Check region 1,4,2,5
    %Check Region 1
    Ts = T4_p(p);
    hL = 109.6635 * log(p) + 40.3481 * p + 734.58; %Approximate
function for hL_p
    if abs(h - hL) < 100 %if approximate is not god enough use real
function
        hL = h1_pT(p, Ts);
    end
    if h <= hL
        region_ph = 1;
        return
    end
    %Check Region 4
    hV = 45.1768 * log(p) - 20.158 * p + 2804.4; %Approximate function
for hV_p
    if abs(h - hV) < 50 %if approximate is not god enough use real
function
        hV = h2_pT(p, Ts);
    end
    if h < hV
        region_ph = 4;
        return
    end
    %Check upper limit of region 2 Quick Test
    if h < 4000
        region_ph = 2;
        return
    end
    %Check region 2 (Real value)
    h_45 = h2_pT(p, 1073.15);
    if h <= h_45
        region_ph = 2;
        return
    end
    %Check region 5
    if p > 10
        region_ph = 0;
        return
    end
    h_5u = h5_pT(p, 2273.15);
    if h < h_5u
        region_ph = 5;
        return
    end
    region_ph = 0;
    return
else %for p>16.5292
    %Check if in region1
    if h < h1_pT(p, 623.15)
        region_ph = 1;
        return
    end
end

```

```

    %Check if in region 3 or 4 (Bellow Reg 2)
    if h < h2_pT(p, B23T_p(p))
        %Region 3 or 4
        if p > p3sat_h(h)
            region_ph = 3;
            return
        else
            region_ph = 4;
            return
        end
    end
    %Check if region 2
    if h < h2_pT(p, 1073.15)
        region_ph = 2;
        return
    end
end
region_ph = 0;

%*****
%*****
%*3.3 Regions as a function of ps
function region_ps = region_ps( p, s)
if p < 0.000611657 | p > 100 | s < 0 | s > s5_pT(p, 2273.15)
    region_ps = 0;
    return
end

%Check region 5
if s > s2_pT(p, 1073.15)
    if p <= 10
        region_ps = 5;
        return
    else
        region_ps = 0;
        return
    end
end

%Check region 2
if p > 16.529
    ss = s2_pT(p, B23T_p(p)); %Between 5.047 & 5.261. Use to speed
    up!
else
    ss = s2_pT(p, T4_p(p));
end
if s > ss
    region_ps = 2;
    return
end

%Check region 3
ss = s1_pT(p, 623.15);
if p > 16.529 & s > ss
    if p > p3sat_s(s)
        region_ps = 3;
        return
    else
        region_ps = 4;
    end
end

```

```

        return
    end
end

%Check region 4 (Not inside region 3)
if p < 16.529 & s > s1_pT(p, T4_p(p))
    region_ps = 4;
    return
end

%Check region 1
if p > 0.000611657 & s > s1_pT(p, 273.15)
    region_ps = 1;
    return
end
region_ps = 1;

%*****
%*****
%*3.4 Regions as a function of hs
function region_hs = region_hs( h, s)
if s < -0.0001545495919
    region_hs = 0;
    return
end
%Check linear adaption to p=0.000611. if bellow region 4.
hMin = (((-0.0415878 - 2500.89262) / (-0.00015455 - 9.155759)) * s);
if s < 9.155759395 & h < hMin
    region_hs = 0;
    return
end

%*****Kolla 1 eller 4. (+liten bit över B13)
if s >= -0.0001545495919 & s <= 3.77828134
    if h < h4_s(s)
        region_hs = 4;
        return
    elseif s < 3.397782955 %100MPa line is limiting
        TMax = T1_ps(100, s);
        hMax = h1_pT(100, TMax);
        if h < hMax
            region_hs = 1;
            return
        else
            region_hs = 0;
            return
        end
    else %The point is either in region 4,1,3. Check B23
        hB = hB13_s(s);
        if h < hB
            region_hs = 1;
            return
        end
        TMax = T3_ps(100, s);
        vmax = v3_ps(100, s);
        hMax = h3_rhoT(1 / vmax, TMax);
        if h < hMax
            region_hs = 3;
            return
        else
            region_hs = 0;
        end
    end
end

```

```

        return
    end
end
end

%*****Kolla region 2 eller 4. (Övre delen av område b23-> max)
if s >= 5.260578707 & s <= 11.9212156897728
    if s > 9.155759395 %Above region 4
        Tmin = T2_ps(0.000611, s);
        hMin = h2_pT(0.000611, Tmin);
        %function adapted to h(1073.15,s)
        hMax = -0.07554022 * s ^ 4 + 3.341571 * s ^ 3 - 55.42151 * s ^
2 + 408.515 * s + 3031.338;
        if h > hMin & h < hMax
            region_hs = 2;
            return
        else
            region_hs = 0;
            return
        end
    end
end

hV = h4_s(s);

if h < hV %Region 4. Under region 3.
    region_hs = 4;
    return
end
if s < 6.04048367171238
    TMax = T2_ps(100, s);
    hMax = h2_pT(100, TMax);
else
    %function adapted to h(1073.15,s)
    hMax = -2.988734 * s ^ 4 + 121.4015 * s ^ 3 - 1805.15 * s ^ 2
+ 11720.16 * s - 23998.33;
end
if h < hMax %Region 2. Över region 4.
    region_hs = 2;
    return
else
    region_hs = 0;
    return
end
end

%Kolla region 3 eller 4. Under kritiska punkten.
if s >= 3.77828134 & s <= 4.41202148223476
    hL = h4_s(s);
    if h < hL
        region_hs = 4;
        return
    end
    TMax = T3_ps(100, s);
    vmax = v3_ps(100, s);
    hMax = h3_rhoT(1 / vmax, TMax);
    if h < hMax
        region_hs = 3;
        return
    else
        region_hs = 0;
        return
    end
end

```



```

end
end

%Kolla region 3 eller 4 från kritiska punkten till övre delen av b23
if s >= 4.41202148223476 & s <= 5.260578707
    hV = h4_s(s);
    if h < hV
        region_hs = 4;
        return
    end
    %Kolla om vi är under b23 giltighetsområde.
    if s <= 5.048096828
        TMax = T3_ps(100, s);
        vmax = v3_ps(100, s);
        hMax = h3_rhoT(1 / vmax, TMax);
        if h < hMax
            region_hs = 3;
            return
        else
            region_hs = 0;
            return
        end
    else %Inom området för B23 i s led.
        if (h > 2812.942061) %Ovanför B23 i h_led
            if s > 5.09796573397125
                TMax = T2_ps(100, s);
                hMax = h2_pT(100, TMax);
                if h < hMax
                    region_hs = 2;
                    return
                else
                    region_hs = 0;
                    return
                end
            else
                region_hs = 0;
                return
            end
        else
            region_hs = 0;
            return
        end
    end
    if (h < 2563.592004) %Nedanför B23 i h_led men vi har redan
kollat ovanför hV2c3b
        region_hs = 3;
        return
    end
    %Vi är inom b23 området i både s och h led.
    Tact = TB23_hs(h, s);
    pact = p2_hs(h, s);
    pBound = B23p_T(Tact);
    if pact > pBound
        region_hs = 3;
        return
    else
        region_hs = 2;
        return
    end
end
end
region_hs = 0;

%*****
*****

```

```

%*3.5 Regions as a function of p and rho
function Region_prho = Region_prho(p,rho)
v = 1 / rho;
if p < 0.000611657 | p > 100
    Region_prho = 0;
    return
end
if p < 16.5292 %Bellow region 3, Check region 1,4,2
    if v < v1_pT(p, 273.15) %Observe that this is not actually min of
v. Not valid Water of 4°C is lighther.
        Region_prho = 0;
        return
    end
    if v <= v1_pT(p, T4_p(p))
        Region_prho = 1;
        return
    end
    if v < v2_pT(p, T4_p(p))
        Region_prho = 4;
        return
    end
    if v <= v2_pT(p, 1073.15)
        Region_prho = 2;
        return
    end
    if p > 10 %Above region 5
        Region_prho = 0;
        return
    end
    if v <= v5_pT(p, 2073.15)
        Region_prho = 5;
        return
    end
else %Check region 1,3,4,3,2 (Above the lowest point of region 3.)
    if v < v1_pT(p, 273.15) %Observe that this is not actually min of
v. Not valid Water of 4°C is lighther.
        Region_prho = 0;
        return
    end
    if v < v1_pT(p, 623.15)
        Region_prho = 1;
        return
    end
    %Check if in region 3 or 4 (Bellow Reg 2)
    if v < v2_pT(p, B23T_p(p))
        %Region 3 or 4
        if p > 22.064 %Above region 4
            Region_prho = 3;
            return
        end
        if v < v3_ph(p, h4L_p(p)) | v > v3_ph(p, h4V_p(p)) %Uses
iteration!!
            Region_prho = 3;
            return
        else
            Region_prho = 4;
            return
        end
    end
    %Check if region 2
    if v < v2_pT(p, 1073.15)

```

```

        Region_prho = 2;
        return
    end
end
Region_prho = 0;

%*****
%*****
%*4 Region Borders
%*****
%*****
%*****
%*****
%*4.1 Boundary between region 2 and 3.
function B23p_T = B23p_T(T)
%Release on the IAPWS Industrial formulation 1997 for the
Thermodynamic Properties of Water and Steam
%1997
%Section 4 Auxiliary Equation for the Boundary between Regions 2 and 3
%Eq 5, Page 5
B23p_T = 348.05185628969 - 1.1671859879975 * T + 1.0192970039326E-03 *
T ^ 2;

function B23T_p = B23T_p(p)
%Release on the IAPWS Industrial formulation 1997 for the
Thermodynamic Properties of Water and Steam
%1997
%Section 4 Auxiliary Equation for the Boundary between Regions 2 and 3
%Eq 6, Page 6
B23T_p = 572.54459862746 + ((p - 13.91883977887) / 1.0192970039326E-
03) ^ 0.5;

%*****
%*****
%*4.2 Region 3. pSat_h & pSat_s
function p3sat_h = p3sat_h(h)
%Revised Supplementary Release on Backward Equations for the functions
T(p,h), v(p,h) & T(p,s), v(p,s) for Region 3 of the IAPWS Industrial
formulation 1997 for the Thermodynamic Properties of Water & Steam
%2004
%Section 4 Boundary Equations psat(h) & psat(s) for the Saturation
Lines of Region 3
%See pictures Page 17, Eq 10, Table 17, Page 18
Ii = [0, 1, 1, 1, 1, 5, 7, 8, 14, 20, 22, 24, 28, 36];
Ji = [0, 1, 3, 4, 36, 3, 0, 24, 16, 16, 3, 18, 8, 24];
ni = [0.600073641753024, -9.36203654849857, 24.6590798594147, -
107.014222858224, -91582131580576.8, -8623.32011700662, -
23.5837344740032, 2.52304969384128E+17, -3.89718771997719E+18, -
3.33775713645296E+22, 35649946963.6328, -1.48547544720641E+26,
3.30611514838798E+18, 8.13641294467829E+37];
hs = h / 2600;
ps = 0;
for i = 1:14
    ps = ps + ni(i) * (hs - 1.02) ^ Ii(i) * (hs - 0.608) ^ Ji(i);
end
p3sat_h = ps * 22;

```

```

function p3sat_s = p3sat_s(s)
Ii = [0, 1, 1, 4, 12, 12, 16, 24, 28, 32];
Ji = [0, 1, 32, 7, 4, 14, 36, 10, 0, 18];
ni = [0.639767553612785, -12.9727445396014, -2.24595125848403E+15,
1774667.41801846, 7170793495.71538, -3.78829107169011E+17, -
9.55586736431328E+34, 1.87269814676188E+23, 119254746466.473,
1.10649277244882E+36];
Sigma = s / 5.2;
Pi = 0;
for i = 1:10
    Pi = Pi + ni(i) * (Sigma - 1.03) ^ Ii(i) * (Sigma - 0.699) ^
Ji(i);
end
p3sat_s = Pi * 22;

%*****
%*****
%4.3 Region boundary 1to3 & 3to2 as a functions of s
function hB13_s = hB13_s(s)
%Supplementary Release on Backward Equations ( ) , p h s for Region 3,
%Chapter 4.5 page 23.
Ii = [0, 1, 1, 3, 5, 6];
Ji = [0, -2, 2, -12, -4, -3];
ni = [0.913965547600543, -4.30944856041991E-05, 60.3235694765419,
1.17518273082168E-18, 0.220000904781292, -69.0815545851641];
Sigma = s / 3.8;
eta = 0;
for i = 1 : 6
    eta = eta + ni(i) * (Sigma - 0.884) ^ Ii(i) * (Sigma - 0.864) ^
Ji(i);
end
hB13_s = eta * 1700;

function TB23_hs = TB23_hs(h, s)
%Supplementary Release on Backward Equations ( ) , p h s for Region 3,
%Chapter 4.6 page 25.
Ii = [-12, -10, -8, -4, -3, -2, -2, -2, -2, 0, 1, 1, 1, 3, 3, 5, 6, 6,
8, 8, 8, 12, 12, 14, 14];
Ji = [10, 8, 3, 4, 3, -6, 2, 3, 4, 0, -3, -2, 10, -2, -1, -5, -6, -3,
-8, -2, -1, -12, -1, -12, 1];
ni = [6.2909626082981E-04, -8.23453502583165E-04, 5.15446951519474E-
08, -1.17565945784945, 3.48519684726192, -5.07837382408313E-12, -
2.84637670005479, -2.36092263939673, 6.01492324973779,
1.48039650824546, 3.60075182221907E-04, -1.26700045009952E-02, -
1221843.32521413, 0.149276502463272, 0.698733471798484, -
2.52207040114321E-02, 1.47151930985213E-02, -1.08618917681849, -
9.36875039816322E-04, 81.9877897570217, -182.041861521835,
2.61907376402688E-06, -29162.6417025961, 1.40660774926165E-05,
7832370.62349385];
Sigma = s / 5.3;
eta = h / 3000;
teta = 0;
for i = 1 : 25
    teta = teta + ni(i) * (eta - 0.727) ^ Ii(i) * (Sigma - 0.864) ^
Ji(i);
end
TB23_hs = teta * 900;

```

```

%*****
%*****
%*5 Transport properties
%*****
%*****
%*5.1 Viscosity (IAPWS formulation 1985, Revised 2003)
%*****
%*****
function my_AllRegions_pT = my_AllRegions_pT( p, T)
h0 = [0.5132047, 0.3205656, 0, 0, -0.7782567, 0.1885447];
h1 = [0.2151778, 0.7317883, 1.241044, 1.476783, 0, 0];
h2 = [-0.2818107, -1.070786, -1.263184, 0, 0, 0];
h3 = [0.1778064, 0.460504, 0.2340379, -0.4924179, 0, 0];
h4 = [-0.0417661, 0, 0, 0.1600435, 0, 0];
h5 = [0, -0.01578386, 0, 0, 0, 0];
h6 = [0, 0, 0, -0.003629481, 0, 0];

%Calcualte density.
switch region_pT(p, T)
case 1
    rho = 1 / v1_pT(p, T);
case 2
    rho = 1 / v2_pT(p, T);
case 3
    hs = h3_pT(p, T);
    rho = 1 / v3_ph(p, hs);
case 4
    rho = NaN;
case 5
    rho = 1 / v5_pT(p, T);
otherwise
    my_AllRegions_pT = NaN;
    return
end

rhos = rho / 317.763;
Ts = T / 647.226;
ps = p / 22.115;

%Check valid area
if T > 900 + 273.15 | (T > 600 + 273.15 & p > 300) | (T > 150 +
273.15 & p > 350) | p > 500
    my_AllRegions_pT = NaN;
    return
end
my0 = Ts ^ 0.5 / (1 + 0.978197 / Ts + 0.579829 / (Ts ^ 2) - 0.202354 /
(Ts ^ 3));
Sum = 0;
for i = 0 : 5
    Sum = Sum + h0(i+1) * (1 / Ts - 1) ^ i + h1(i+1) * (1 / Ts - 1) ^
i * (rhos - 1) ^ 1 + h2(i+1) * (1 / Ts - 1) ^ i * (rhos - 1) ^ 2 +
h3(i+1) * (1 / Ts - 1) ^ i * (rhos - 1) ^ 3 + h4(i+1) * (1 / Ts - 1) ^
i * (rhos - 1) ^ 4 + h5(i+1) * (1 / Ts - 1) ^ i * (rhos - 1) ^ 5 +
h6(i+1) * (1 / Ts - 1) ^ i * (rhos - 1) ^ 6;
end
my1 = exp(rhos * Sum);
mys = my0 * my1;
my_AllRegions_pT = mys * 0.000055071;

```

```

function my_AllRegions_ph = my_AllRegions_ph( p, h)
h0 = [0.5132047, 0.3205656, 0, 0, -0.7782567, 0.1885447];
h1 = [0.2151778, 0.7317883, 1.241044, 1.476783, 0, 0];
h2 = [-0.2818107, -1.070786, -1.263184, 0, 0, 0];
h3 = [0.1778064, 0.460504, 0.2340379, -0.4924179, 0, 0];
h4 = [-0.0417661, 0, 0, 0.1600435, 0, 0];
h5 = [0, -0.01578386, 0, 0, 0, 0];
h6 = [0, 0, 0, -0.003629481, 0, 0];

%Calculalte density.
switch region_ph(p, h)
case 1
    Ts = T1_ph(p, h);
    T = Ts;
    rho = 1 / v1_pT(p, Ts);
case 2
    Ts = T2_ph(p, h);
    T = Ts;
    rho = 1 / v2_pT(p, Ts);
case 3
    rho = 1 / v3_ph(p, h);
    T = T3_ph(p, h);
case 4
    xs = x4_ph(p, h);
    if p < 16.529
        v4v = v2_pT(p, T4_p(p));
        v4L = v1_pT(p, T4_p(p));
    else
        v4v = v3_ph(p, h4V_p(p));
        v4L = v3_ph(p, h4L_p(p));
    end
    rho = 1 / (xs * v4v + (1 - xs) * v4L);
    T = T4_p(p);
case 5
    Ts = T5_ph(p, h);
    T = Ts;
    rho = 1 / v5_pT(p, Ts);
otherwise
    my_AllRegions_ph = NaN;
    return
end
rhos = rho / 317.763;
Ts = T / 647.226;
ps = p / 22.115;
%Check valid area
if T > 900 + 273.15 | (T > 600 + 273.15 & p > 300) | (T > 150 +
273.15 & p > 350) | p > 500
    my_AllRegions_ph = NaN;
    return
end
my0 = Ts ^ 0.5 / (1 + 0.978197 / Ts + 0.579829 / (Ts ^ 2) - 0.202354 /
(Ts ^ 3));

Sum = 0;
for i = 0 : 5
    Sum = Sum + h0(i+1) * (1 / Ts - 1) ^ i + h1(i+1) * (1 / Ts - 1) ^
i * (rhos - 1) ^ 1 + h2(i+1) * (1 / Ts - 1) ^ i * (rhos - 1) ^ 2 +
h3(i+1) * (1 / Ts - 1) ^ i * (rhos - 1) ^ 3 + h4(i+1) * (1 / Ts - 1) ^

```

```

i * (rhos - 1) ^ 4 + h5(i+1) * (1 / Ts - 1) ^ i * (rhos - 1) ^ 5 +
h6(i+1) * (1 / Ts - 1) ^ i * (rhos - 1) ^ 6;
end
my1 = exp(rhos * Sum);
mys = my0 * my1;
my_AllRegions_ph = mys * 0.000055071;

%*****
%*****
%5.2 Thermal Conductivity (IAPWS formulation 1985)
function tc_ptrho = tc_ptrho( p, T, rho)
%Revised release on the IAPWS formulation 1985 for the Thermal
Conductivity of ordinary water
%IAPWS September 1998
%Page 8
%ver2.6 Start corrected bug
if T < 273.15
    tc_ptrho = NaN; %Out of range of validity (para. B4)
    return
elseif T < 500 + 273.15
    if p > 100
        tc_ptrho = NaN; %Out of range of validity (para. B4)
        return
    end
elseif T <= 650 + 273.15
    if p > 70
        tc_ptrho = NaN; %Out of range of validity (para. B4)
        return
    end
else T <= 800 + 273.15
    if p > 40
        tc_ptrho = NaN; %Out of range of validity (para. B4)
        return
    end
end
end
%ver2.6 End corrected bug
T = T / 647.26;
rho = rho / 317.7;
tc0 = T ^ 0.5 * (0.0102811 + 0.0299621 * T + 0.0156146 * T ^ 2 -
0.00422464 * T ^ 3);
tc1 = -0.39707 + 0.400302 * rho + 1.06 * exp(-0.171587 * (rho +
2.39219) ^ 2);
dT = abs(T - 1) + 0.00308976;
Q = 2 + 0.0822994 / dT ^ (3 / 5);
if T >= 1
    s = 1 / dT;
else
    s = 10.0932 / dT ^ (3 / 5);
end
tc2 = (0.0701309 / T ^ 10 + 0.011852) * rho ^ (9 / 5) * exp(0.642857 *
(1 - rho ^ (14 / 5))) + 0.00169937 * s * rho ^ Q * exp((Q / (1 + Q)) *
(1 - rho ^ (1 + Q))) - 1.02 * exp(-4.11717 * T ^ (3 / 2) - 6.17937 /
rho ^ 5);
tc_ptrho = tc0 + tc1 + tc2;

%*****
%*****
%5.3 Surface Tension
function Surface_Tension_T = Surface_Tension_T( T)
%IAPWS Release on Surface Tension of Ordinary Water Substance,
%September 1994

```

```

tc = 647.096; %K
B = 0.2358; %N/m
bb = -0.625;
my = 1.256;
if T < 0.01 | T > tc
    Surface_Tension_T = NaN; %"Out of valid region"
    return
end
tau = 1 - T / tc;
Surface_Tension_T = B * tau ^ my * (1 + bb * tau);

%*****
%*****
%*6 Units
*
%*****
%*****
function toSIunit_p = toSIunit_p( Ins )
%Translate bar to MPa
toSIunit_p = Ins / 10;
function fromSIunit_p = fromSIunit_p( Ins )
%Translate bar to MPa
fromSIunit_p = Ins * 10;
function toSIunit_T = toSIunit_T( Ins )
%Translate degC to Kelvon
toSIunit_T = Ins + 273.15;
function fromSIunit_T = fromSIunit_T( Ins )
%Translate Kelvin to degC
fromSIunit_T = Ins - 273.15;
function toSIunit_h = toSIunit_h( Ins )
toSIunit_h = Ins;
function fromSIunit_h = fromSIunit_h( Ins )
fromSIunit_h = Ins;
function toSIunit_v = toSIunit_v( Ins )
toSIunit_v = Ins;
function fromSIunit_v = fromSIunit_v( Ins )
fromSIunit_v = Ins;
function toSIunit_s = toSIunit_s( Ins )
toSIunit_s = Ins;
function fromSIunit_s = fromSIunit_s( Ins )
fromSIunit_s = Ins;
function toSIunit_u = toSIunit_u( Ins )
toSIunit_u = Ins;
function fromSIunit_u = fromSIunit_u( Ins )
fromSIunit_u = Ins;
function toSIunit_Cp = toSIunit_Cp( Ins )
toSIunit_Cp = Ins;
function fromSIunit_Cp = fromSIunit_Cp( Ins )
fromSIunit_Cp = Ins;
function toSIunit_Cv = toSIunit_Cv( Ins )
toSIunit_Cv = Ins;
function fromSIunit_Cv = fromSIunit_Cv( Ins )
fromSIunit_Cv = Ins;
function toSIunit_w = toSIunit_w( Ins )
toSIunit_w = Ins;
function fromSIunit_w = fromSIunit_w( Ins )
fromSIunit_w = Ins;
function toSIunit_tc = toSIunit_tc( Ins )
toSIunit_tc = Ins;
function fromSIunit_tc = fromSIunit_tc( Ins )

```



```

fromSIunit_tc = Ins;
function toSIunit_st = toSIunit_st( Ins )
toSIunit_st = Ins;
function fromSIunit_st = fromSIunit_st( Ins )
fromSIunit_st = Ins;
function toSIunit_x = toSIunit_x( Ins )
toSIunit_x = Ins;
function fromSIunit_x = fromSIunit_x( Ins )
fromSIunit_x = Ins;
function toSIunit_vx = toSIunit_vx( Ins )
toSIunit_vx = Ins;
function fromSIunit_vx = fromSIunit_vx( Ins )
fromSIunit_vx = Ins;
function toSIunit_my = toSIunit_my( Ins )
toSIunit_my = Ins;
function fromSIunit_my = fromSIunit_my( Ins )
fromSIunit_my = Ins;

%*****
%*****
%*7 Verification
*
%*****
%*****

function err = check()
err=0;
%*****
%*****
%* 7.1 Verifiy region 1
%IF-97 Table 5, Page 9
p=[30/10,800/10,30/10];
T=[300,300,500];
Fun={'v1_pT','h1_pT','u1_pT','sl_pT','Cp1_pT','w1_pT'};
IF97=[0.00100215168,0.000971180894,0.001202418;...
      115.331273,184.142828,975.542239;...
      112.324818,106.448356,971.934985;...
      0.392294792,0.368563852,2.58041912;...
      4.17301218,4.01008987,4.65580682;...
      1507.73921,1634.69054,1240.71337];
R1=zeros(6,3);
for i=1:3
    for j=1:6

R1(j,i)=eval([char(Fun(j)),'(',num2str(p(i))',' ',num2str(T(i))',' ');']);
    end
end
Region1_error=abs((R1-IF97)./IF97)
err=err+sum(sum(Region1_error>1E-8))

%IF-97 Table 7, Page 11
p=[30/10,800/10,800/10];
h=[500,500,1500];
IF97=[391.798509,378.108626,611.041229];
R1=zeros(1,3);
for i=1:3
    R1(i)=T1_ph(p(i),h(i));
end
T1_ph_error=abs((R1-IF97)./IF97)
err=err+sum(sum(T1_ph_error>1E-8))

```

```

%IF-97 Table 9, Page 12
p=[30/10,800/10,800/10];
s=[0.5,0.5,3];
IF97=[307.842258,309.979785,565.899909];
R1=zeros(1,3);
for i=1:3
    R1(i)=T1_ps(p(i),s(i));
end
T1_ps_error=abs((R1-IF97)./IF97)
err=err+sum(sum(T1_ps_error>1E-8))

%Supplementary Release on Backward Equations
%for Pressure as a Function of Enthalpy and Entropy p(h,s)
%Table 3, Page 6
h=[0.001,90,1500];
s=[0,0,3.4];
IF97=[0.0009800980612,91.929547272,58.68294423];
R1=zeros(1,3);
for i=1:3
    R1(i)=p1_hs(h(i),s(i));
end
p1_hs_error=abs((R1-IF97)./IF97)
err=err+sum(sum(p1_hs_error>1E-8))

%*****
%*****
%* 7.2 Verifiy region 2
% IF-97 Table 15, Page 17
p=[0.035/10,0.035/10,300/10];
T=[300,700,700];
Fun={'v2_pT','h2_pT','u2_pT','s2_pT','Cp2_pT','w2_pT'};
IF97=[39.4913866,92.3015898,0.00542946619;...
      2549.91145,3335.68375,2631.49474;...
      2411.6916,3012.62819,2468.61076;...
      8.52238967,10.1749996,5.17540298;...
      1.91300162,2.08141274,10.3505092;...
      427.920172,644.289068,480.386523];
R2=zeros(6,3);
for i=1:3
    for j=1:6
        R2(j,i)=eval([char(Fun(j)),'(',num2str(p(i))',' ',num2str(T(i))',')']);
    end
end
Region2_error=abs((R2-IF97)./IF97)
err=err+sum(sum(Region2_error>1E-8))

%IF-97 Table 24, Page 25
p=[0.01/10,30/10,30/10,50/10,50/10,250/10,400/10,600/10,600/10];
h=[3000,3000,4000,3500,4000,3500,2700,2700,3200];
IF97=[534.433241,575.37337,1010.77577,801.299102,1015.31583,875.279054
,743.056411,791.137067,882.75686];
R2=zeros(1,9);
for i=1:9
    R2(i)=T2_ph(p(i),h(i));
end
T2_ph_error=abs((R2-IF97)./IF97)
err=err+sum(sum(T2_ph_error>1E-8))

%IF-97 Table 29, Page 29
p=[1/10,1/10,25/10,80/10,80/10,900/10,200/10,800/10,800/10];

```

```

s=[7.5,8,8,6,7.5,6,5.75,5.25,5.75];
IF97=[399.517097,514.127081,1039.84917,600.48404,1064.95556,1038.01126
,697.992849,854.011484,949.017998];
R2=zeros(1,9);
for i=1:9
    R2(i)=T2_ps(p(i),s(i));
end
T2_ps_error=abs((R2-IF97)./IF97)
err=err+sum(sum(T2_ps_error>1E-8))

%Supplementary Release on Backward Equations for Pressure as a
Function of Enthalpy and Entropy p(h,s)
%Table 3, Page 6
h=[2800,2800,4100,2800,3600,3600,2800,2800,3400];
s=[6.5,9.5,9.5,6,6,7,5.1,5.8,5.8];
IF97=[1.371012767,0.001879743844,0.1024788997,4.793911442,83.95519209,
7.527161441,94.3920206,8.414574124,83.76903879];
R2=zeros(1,9);
for i=1:9
    R2(i)=p2_hs(h(i),s(i));
end
p2_hs_error=abs((R2-IF97)./IF97)
err=err+sum(sum(p2_hs_error>1E-8))

%*****
%*****
%* 7.3 Verifiy region 3
% IF-97 Table 33, Page 32
T=[650,650,750];
rho=[500,200,500];
Fun={'p3_rhoT','h3_rhoT','u3_rhoT','s3_rhoT','Cp3_rhoT','w3_rhoT'};
IF97=[25.5837018,22.2930643,78.3095639;...
1863.43019,2375.12401,2258.68845;...
1812.26279,2263.65868,2102.06932;...
4.05427273,4.85438792,4.46971906;...
13.8935717,44.6579342,6.34165359;...
502.005554,383.444594,760.696041];
R3=zeros(6,3);
for i=1:3
    for j=1:6
        R3(j,i)=eval([char(Fun(j)),'(',num2str(rho(i))',' ',num2str(T(i))',' ');
]);
    end
end
Region3_error=abs((R3-IF97)./IF97)
err=err+sum(sum(Region3_error>1E-8))

%T3_ph
p=[200/10,500/10,1000/10,200/10,500/10,1000/10];
h=[1700,2000,2100,2500,2400,2700];
IF97=[629.3083892,690.5718338,733.6163014,641.8418053,735.1848618,842.
0460876];
R3=zeros(1,6);
for i=1:6
    R3(i)=T3_ph(p(i),h(i));
end
T3_ph_error=abs((R3-IF97)./IF97)
err=err+sum(sum(T3_ph_error>1E-8))

%v3_ph

```

```

p=[200/10,500/10,1000/10,200/10,500/10,1000/10];
h=[1700,2000,2100,2500,2400,2700];
IF97=[0.001749903962,0.001908139035,0.001676229776,0.006670547043,0.00
28012445,0.002404234998];
R3=zeros(1,6);
for i=1:6
    R3(i)=v3_ph(p(i),h(i));
end
v3_ph_error=abs((R3-IF97)./IF97)
err=err+sum(sum(v3_ph_error>1E-7))

```

```

%T3_ps
p=[200/10,500/10,1000/10,200/10,500/10,1000/10];
s=[3.7,3.5,4,5,4.5,5];
IF97=[620.8841563,618.1549029,705.6880237,640.1176443,716.3687517,847.
4332825];
R3=zeros(1,6);
for i=1:6
    R3(i)=T3_ps(p(i),s(i));
end
T3_ps_error=abs((R3-IF97)./IF97)
err=err+sum(sum(T3_ps_error>1E-8))

```

```

%v3_ps
p=[200/10,500/10,1000/10,200/10,500/10,1000/10];
s=[3.7,3.5,4,5,4.5,5];
IF97=[0.001639890984,0.001423030205,0.001555893131,0.006262101987,0.00
2332634294,0.002449610757];
R3=zeros(1,6);
for i=1:6
    R3(i)=v3_ps(p(i),s(i));
end
v3_ps_error=abs((R3-IF97)./IF97)
err=err+sum(sum(v3_ps_error>1E-8))

```

```

%p3_hs
h=[1700,2000,2100,2500,2400,2700];
s=[3.8,4.2,4.3,5.1,4.7,5];
IF97=[25.55703246,45.40873468,60.7812334,17.20612413,63.63924887,88.39
043281];
R3=zeros(1,6);
for i=1:6
    R3(i)=p3_hs(h(i),s(i));
end
p3_hs_error=abs((R3-IF97)./IF97)
err=err+sum(sum(p3_hs_error>1E-8))

```

```

%h3_pT (Iteration)
p=[255.83702,222.93064,783.09564]./10;
T=[650,650,750];
IF97=[1863.271389,2375.696155,2258.626582];
R3=zeros(1,3);
for i=1:3
    R3(i)=h3_pT(p(i),T(i));
end
h3_pT_error=abs((R3-IF97)./IF97)
err=err+sum(sum(h3_pT_error>1E-6)) %Decimals in IF97

```

```

%*****
*****

```

```

%* 7.4 Verifiy region 4
%Saturation pressure, If97, Table 35, Page 34
T=[300,500,600];
IF97=[0.0353658941, 26.3889776, 123.443146]/10;
R3=zeros(1,3);
for i=1:3
    R4(i)=p4_T(T(i));
end
p4_t_error=abs((R4-IF97)./IF97)
err=err+sum(sum(p4_t_error>1E-7))

T=[1,10,100]/10;
IF97=[372.755919,453.035632,584.149488];
R3=zeros(1,3);
for i=1:3
    R4(i)=T4_p(T(i));
end
T4_p_error=abs((R4-IF97)./IF97)
err=err+sum(sum(T4_p_error>1E-7))

s=[1,2,3,3.8,4,4.2,7,8,9,5.5,5,4.5];
IF97=[308.5509647,700.6304472,1198.359754,1685.025565,1816.891476,1949
.352563,2723.729985,2599.04721,2511.861477,2687.69385,2451.623609,2144
.360448];
R3=zeros(1,12);
for i=1:12
    R4(i)=h4_s(s(i));
end
h4_s_error=abs((R4-IF97)./IF97)
err=err+sum(sum(h4_s_error>1E-7))

%*****
%*****
%* 7.5 Verifiy region 5
% IF-97 Table 42, Page 39
T=[1500,1500,2000];
p=[5,80,80]/10;
Fun={'v5_pT','h5_pT','u5_pT','s5_pT','Cp5_pT','w5_pT'};
IF97=[1.38455354,0.0865156616,0.115743146;...
5219.76332,5206.09634,6583.80291;...
4527.48654,4513.97105,5657.85774;...
9.65408431,8.36546724,9.15671044;...
2.61610228,2.64453866,2.8530675;...
917.071933,919.708859,1054.35806];
R5=zeros(6,3);
for i=1:3
    for j=1:6
        R5(j,i)=eval([char(Fun(j)),'(',num2str(p(i))','(',num2str(T(i))',')');
    end
end
Region5_error=abs((R5-IF97)./IF97)
err=err+sum(sum(Region5_error>1E-8))

%T5_ph (Iteration)
p=[0.5,8,8];
h=[5219.76331549428,5206.09634477373,6583.80290533381];
IF97=[1500,1500,2000];
R5=zeros(1,3);
for i=1:3
    R5(i)=T5_ph(p(i),h(i));

```

```

end
T5_ph_error=abs((R5-IF97)./IF97)
err=err+sum(sum(T5_ph_error>1E-7)) %Decimals in IF97

%T5_ps (Iteration)
p=[0.5,8,8];
s=[9.65408430982588,8.36546724495503,9.15671044273249];
IF97=[1500,1500,2000];
R5=zeros(1,3);
for i=1:3
    R5(i)=T5_ps(p(i),s(i));
end
T5_ps_error=abs((R5-IF97)./IF97)
err=err+sum(sum(T5_ps_error>1E-4)) %Decimals in IF97

%*****
%*****
%* 7.6 Verifiy calling funtions

fun={'Tsat_p','T_ph','T_ps','T_hs','psat_T','p_hs','hV_p','hL_p','hV_T',
'hL_T','h_pT','h_ps','h_px','h_prho','h_Tx','vV_p','vL_p','vV_T','vL_T',
'v_pT','v_ph','v_ps','rhoV_p','rhoL_p','rhoV_T','rhoL_T','rho_pT',
'rho_ph','rho_ps','sV_p','sL_p','sV_T','sL_T','s_pT','s_ph','uV_p','uL_p',
'uV_T','uL_T','u_pT','u_ph','u_ps','CpV_p','CpL_p','CpV_T','CpL_T',
'Cp_pT','Cp_ph','Cp_ps','CvV_p','CvL_p','CvV_T','CvL_T','Cv_pT','Cv_p',
'h','Cv_ps','wV_p','wL_p','wV_T','wL_T','w_pT','w_ph','w_ps','my_pT','m',
'y_ph','my_ps','tcL_p','tcV_p','tcL_T','tcV_T','tc_pT','tc_ph','tc_hs',
'st_T','st_p','x_ph','x_ps','vx_ph','vx_ps'};
In1={'1','1','1','100','100','84','1','1','100','100','1','1','1','1',
'100','1','1','100','100','1','1','1','1','1','1','100','100','1','1','1',
'0.006117','0.0061171','0.0001','100','1','1','1','1','1','100','100','1',
'1','1','1','1','100','100','1','1','1','1','1','1','100','100','1','1','1',
'1','1','100','100','1','1','1','1','1','1','1','1','1','1','1','25','25','1','1',
'100','100','1','1','1','1','1'};
In2={'0','100','1','0.2','0','0.296','0','0','0','0','20','1','0.5','2',
'0.5','0','0','0','0','100','1000','5','0','0','0','0','100','1000',
'1','0','0','0','0','0','20','84.01181117','0','0','0','0','0','100','1000','1',
'0','0','0','0','100','200','1','0','0','0','0','0','100','200','1','0',
'0','0','0','100','200','1','100','100','1','0','0','0','0','0','25','100',
'0.34','0','0','1000','4','418','4'};
Control={'99.60591861','23.84481908','73.70859421','13.84933511','1.01',
4179779','2.295498269','2674.949641','417.4364858','2675.572029','419.',
099155','84.01181117','308.6107171','1546.193063','1082.773391','1547.',
33559210927','1.694022523','0.001043148','1.671860601','0.001043455','1',
1.695959407','0.437925658','1.03463539','0.590310924','958.6368897','0',
.598135993','958.3542773','0.589636754',...
'2.283492601','975.6236788','9.155465556','1.8359E-
05','9.155756716','1.307014328','0.296482921','0.296813845','2505.5473',
89','417.332171','2506.015308','418.9933299','2506.171426','956.207434',
2','308.5082185','2.075938025','4.216149431','2.077491868','4.21664511',
9','2.074108555','4.17913573168802','4.190607038','1.552696979','3.769',
699683','1.553698696','3.76770022','1.551397249',...

'4.035176364','3.902919468','472.0541571','1545.451948','472.2559492',
'1545.092249','472.3375235','1542.682475','1557.8585','1.22704E-
05','0.000914003770302108','0.000384222','0.677593822','0.024753668','0',
0.607458162','0.018326723','0.607509806','0.605710062','0.606283124','0',
0.0589118685876641','0.058987784','0.258055424','0.445397961','0.28849',
3093','0.999233827'};

for i=1:length(fun)

```

```

T=['XSteam('',char(fun(i)),'',',char(In1(i)),',',char(In2(i)),')'];
Res=eval(T);
Error=(Res-(str2num(char(Control(i)))))/str2num(char(Control(i)));
Check=[T,'=',num2str(Res),' -
(Control)',char(Control(i)), '=',num2str(Error)]
if Error>1E-5
    err=err+1;
    error('To large error')
end
end
end

```

## **Función *xair***

```

function Out = xair( fun, In1, In2 )
%
%XAIR provides accurate air properties from -173 to 2727 °C at 1 bar
according to
%   BERGMAN, Th.L. & LAVINE, A.S & INCROPERA, F.P. & DEWITT, D.P.
"Fundamentals
%   of Heat and Mass Transfer", Seventh Edition, John Wiley & Sons.
Appendix A.4
%
%   OUT = XAIR('FUN', Tmean)      or
%   OUT = XAIR('FUN', To, Tinf)   return the air propertie. FUN must
always be the
%   air function you want to use. The other argument(s) is(are) the
temperature(s).
%
%   VALID PROPERTIES:
%
%   T      Temperature                (°C)
%   rho    Density                    (kg/m3)
%   v      Specific volume             (m3/kg)
%   Cp     Specific isobaric heat capacity (kJ/kg °C)
%   mu     Viscosity                  (Pa.s)
%   nu     Kinematic viscosity        (m2/s)
%   Pr     Prandtl number
%   tc     Thermal conductivity       (W/m °C)
%   alpha  Thermal diffusivity        (m2/s)
%   beta   Volumetric thermal expansion coeffcient (1/K)
%
%   VALID FUNCTIONS:
%
%   rho_T  Density as a function of temperature.
%   v_T    Specific volume as a function of temperature.
%   Cp_T   Heat capacity as a function of temperature.
%   mu_T   Viscosity as a function of temperature.
%   nu_T   Kinematic viscosity as a function of temperature.
%   Pr_T   Prandtl number as a function of temperature.
%   k_T    Thermal conductivity as a function of temperature.
%   alpha_T Thermal diffusivity as a function of temperature.
%   beta_T Volumetric thermal expansion as a function of
temperatures.

```

```

T = [-173 -123 -73 -23 27 77 127 177 227 277 327 377 427 477 527 577
627 677 727 827 927 1027 1127 1227 1327 1427 1527 1627 1727 1827 1927
2027 2127 2227 2727];
rho = [3.5562 2.3364 1.7458 1.3947 1.1614 0.995 0.8711 0.774 0.6964
0.6329 0.5804 0.5356 0.4975 0.4643 0.4354 0.4097 0.3868 0.3666 0.3482
0.3166 0.2902 0.2679 0.2488 0.2322 0.2177 0.2049 0.1935 0.1833 0.1741
0.1658 0.1582 0.1513 0.1448 0.1389 0.1135];
cp = [1.032 1.012 1.007 1.006 1.007 1.009 1.014 1.021 1.03 1.04 1.051
1.063 1.075 1.087 1.099 1.11 1.121 1.131 1.141 1.159 1.175 1.189 1.207
1.23 1.248 1.267 1.286 1.307 1.337 1.372 1.417 1.478 1.558 1.665
2.726];
mu = [71.1 103.4 132.5 159.6 184.6 208.2 230.1 250.7 270.1 288.4 305.8
322.5 338.8 354.6 369.8 384.3 398.1 411.3 424.4 449 473 496 530 557
584 611 637 663 689 715 740 766 792 818 955]*1E-7;
nu = [2 4.426 7.59 11.44 15.89 20.92 26.41 32.39 38.79 45.57 52.69
60.21 68.1 76.37 84.93 93.8 102.9 112.2 121.9 141.8 162.9 185.1 213
240 268 298 329 362 396 431 468 506 547 589 841]*1E-6;
k = [9.34 13.8 18.1 22.3 26.3 30 33.8 37.3 40.7 43.9 46.9 49.7 52.4
54.9 57.3 59.6 62 64.3 66.7 71.5 76.3 82 91 100 106 113 120 128 137
147 160 175 196 222 486]*1E-3;
alpha = [2.54 5.84 10.3 15.9 22.5 29.9 38.3 47.2 56.7 66.7 76.9 87.3
98 109 120 131 143 155 168 195 224 257 303 350 390 435 482 534 589 646
714 783 869 960 1570]*1E-6;
Pr = [0.786 0.758 0.737 0.72 0.707 0.7 0.69 0.686 0.684 0.683 0.685
0.69 0.695 0.702 0.709 0.716 0.72 0.723 0.726 0.728 0.728 0.719 0.703
0.685 0.688 0.685 0.683 0.677 0.672 0.667 0.655 0.647 0.63 0.613
0.536];

if nargin < 3,
    Tm = In1;
    Tinf = 100;
else
    Tm = (In1 + In2) / 2;
    Tinf = In2;
end

fun = lower( fun );

switch fun
    case {'rho', 'rho_t'}
        Out = spline( T, rho, Tm );
    case {'v', 'v_t', 'sv', 'sv_t'}
        Out = 1/spline( T, rho, Tm );
    case {'cp', 'cp_t'}
        Out = spline( T, cp, Tm );
    case {'pr', 'pr_t'}
        Out = spline( T, Pr, Tm );
    case {'tc', 'tc_t', 'k', 'k_t'}
        Out = spline( T, k, Tm );
    case {'my', 'my_t', 'mu', 'mu_t'}
        Out = spline( T, mu, Tm );
    case {'ny', 'ny_t', 'nu', 'nu_t'}
        Out = spline( T, nu, Tm );
    case {'alpha', 'alpha_t'}
        Out = spline( T, alpha, Tm );
    case {'beta', 'beta_t'}

```



```

        Out = -1/spline( T, rho, Tm ) * (spline( T, rho, Tinf ) -
spline( T, rho, Tm )) / ( Tinf - Tm );    %Incropera 7th, Eq 9.4, Pg
597
end

```

## **Función *insul\_thermo*.**

```

function Out = insul_thermo( In )
% Thermo-12® Gold
% Industrial Insulation Group, LLC    http://www.iig-llc.com/
%
% Composed of hydrous calcium silicate for use on systems operating up
to
% 1200°F (650 °c). It is inorganic, noncombustible
%
% Insulation thickness: 1½, 2, 2½, 3, 3½, 4 in

    % Thermal conductivity of insulation
    t = [ 38      93      149      204      260      316      371      ];
% °C
    k = [ 0.053  0.058  0.064  0.071  0.079  0.087  0.096 ];
% W/m °C

    Out = spline( t, k, In );

```

## **Función *qconv***

```

function q = qsurf( D, Ts, Tinf, WSpeed )
% Calor perdido por unidad de longitud por el mecanismo combinado
convección+radiación, W/m

% CONSTANTES
g      = 9.8;           % Aceleración de la gravedad, m/s2

% Propiedades del Aire a Tmean
Tmean = ( Ts + Tinf ) / 2;    % °C
k      = xair( 'k', Tmean );   % W/m K
nu     = xair( 'nu', Tmean );  % m2/s
Pr     = xair( 'Pr', Tmean );
alpha  = xair( 'alpha', Tmean ); % m2/s
beta   = xair( 'beta', Tmean ); % 1/K

% Efecto de la velocidad del viento
ws = [ 0  5  10  14  19  26  34 ];    % Velocidad del
viento, km/h
wc = [ 1  1.5  2  2.5  3  3.5  4 ];    % Coeficiente
if nargin < 5; WSpeed = 0; end
kconv = spline( ws, wc, WSpeed );

Ra = ( g*beta*(Ts - Tinf)*D^3 ) / ( nu*alpha );

```

```

    Nu = ( 0.60 + ( 0.387*Ra^(1/6) ) / ( 1 + ((0.559/Pr)^(9/16))
)^(8/27) )^2;
    h = k/D * Nu;
    q = h*pi*D*(Ts - Tinf) * kconv;

```

### **Función *qgrad***

```

function q = qsurf( D, Ts, Tinf, epsilon )
% Calor perdido por unidad de longitud por el mecanismo combinado
convección+radiación, W/m
% CONSTANTES
sigma = 5.67E-8; % Constante de Stefan Boltzmann, W/m2 K4
q = epsilon*pi*D*sigma*((Ts+273)^4 - (Tinf+273)^4);

```

### **Función *qcond.***

```

function q = qcond( D, Da, Ts, Ta, k )
% Calor perdido por unidad de longitud por el mecanismo de conducción, W/m
q = 2*pi*k*(Ts - Ta)/log(Da/D);

```