

Sys 1996137

Universidade de São Paulo
Escola Politécnica
Departamento de Engenharia Mecânica

Programa Computacional para
Cálculo dos Parâmetros Psicrométricos

Elaborado por:
Fernando Luís Couceiro

Orientador:
José Roberto Simões Moreira

São Paulo
1999

Universidade de São Paulo
Escola Politécnica
Departamento de Engenharia Mecânica

Programa Computacional para
Cálculo dos Parâmetros Psicrométricos

Trabalho de formatura apresentado à Escola
Politécnica da Universidade de São Paulo
para obtenção do título de Graduação em
Engenharia.

Orientador:
Prof. Dr. José Roberto Simões Moreira

Área de Concentração:
Engenharia Mecânica
Automação e Sistemas

São Paulo

1999

Couceiro, Fernando Luís

**Programa Computacional para Cálculo dos Parâmetros
Psicrométricos, por F. Couceiro. São Paulo, EPUSP, 1999.**

26 p. + apêndice

**Trabalho de Formatura – Escola Politécnica da Universidade
de São Paulo. Departamento de Engenharia Mecânica.**

**1. Psicrometria. I. Universidade de São Paulo. Escola
Politécnica. Departamento de Engenharia Mecânica. II.t.**

AGRADECIMENTOS

Ao orientador Prof. Dr. José Roberto Simões Moreira pelas orientações seguras sempre presentes, mesmo quando fisicamente distante.

SUMÁRIO

Lista de Símbolos

Resumo

Abstract

<i>1 – Introdução</i>	<i>1</i>
<i>2 – Fundamentação Teórica</i>	<i>2</i>
2.1 - Parâmetros Psicrométricos	2
2.2 – A Equação Virial	4
2.3 - A Abordagem Realista (“fator de intensificação”)	5
2.4 – Métodos Numéricos	7
<i>3 – Desenvolvimento do Programa</i>	<i>9</i>
<i>4 – Características do Programa</i>	<i>12</i>
<i>5 – Conclusões</i>	<i>17</i>
<i>Referências Bibliográficas</i>	<i>18</i>

Apêndice

LISTA DE SÍMBOLOS

B_{ii}	coeficientes viriais
C_{ii}	coeficientes viriais
f	fator de intensificação
h	entalpia específica
P	pressão absoluta
P_v	pressão parcial do vapor de água
P_{vs}	pressão parcial do vapor de água na condição de ar saturado
R	constante universal dos gases
T	temperatura
TBS	temperatura de bulbo seco
TBU	temperatura de bulbo úmido
T_{orv}	temperatura de orvalho
v	volume específico
$\bar{v}$	volume molar
x_a	fração molar do ar seco
x_w	fração molar do vapor de água
x_{as}	fração molar do ar seco na condição de ar saturado
x_{ws}	fração molar do vapor de água na condição de ar saturado
ω	umidade absoluta
ω_s	umidade absoluta do ar saturado
ϕ	umidade relativa
μ	grau de saturação

RESUMO

O estudo da psicrometria tem uma importância vital na análise de diversos processos envolvendo manipulação da mistura ar seco e vapor de água, o ar atmosférico. Para o estudo da psicrometria é de grande utilidade uma ferramenta computacional que realize rapidamente todos os cálculos necessários, para diferentes tipos de combinação de dados de entrada. A fim de aumentar a faixa de utilização deste tipo de programa, foi desenvolvido um novo *software* utilizando-se um modelo termodinâmico mais sofisticado, o modelo para gases reais.

O programa foi desenvolvido com o compilador Delphi versão 4. Para a elaboração dos algoritmos necessários utilizou-se largamente métodos numéricos, especialmente o de Newton-Raphson.

Este trabalho é fundamentalmente inspirado no livro do Prof. Dr. José Roberto Simões Moreira, "*Fundamentos e Aplicações da Psicrometria*".

ABSTRACT

The study of Psychrometrics plays a vital role in the analysis of several processes involving the handling of dry air and water vapor, the moist air. A computational tool which quickly carries out all the necessary calculus for different data entry combinations is highly useful in the psychrometrics. In order to increase the range of usage of this type of software, a new one was developed by making use of a more elaborated thermodynamic model, the real gases model.

The software was developed using the Delphi version 4 compiler. Numeric methods were widely used for the design of the necessary algorithms, specially the Newton-Raphson method.

This project is basically inspired on the book "*Fundamentos e Aplicações da Psicrometria*" by Prof. Dr. José Roberto Simões Moreira.

1 – Introdução

A Psicrometria é o ramo da termodinâmica que se preocupa com o estudo da mistura de ar seco e vapor de água. Desta forma, o estudo da Psicrometria é vital para a análise e entendimento dos processos de manipulação do ar úmido. O conhecimento dos processos de condensação e evaporação da água no ar atmosférico tem importância fundamental nos mais variados campos tecnológicos e científicos. Indústrias de papel, alimento e têxtil, armazenamento e beneficiamento de produtos agrícolas, meteorologia, sistemas de controle ambiental são alguns campos onde a Psicrometria pode ser encontrada. Mas é nas áreas de refrigeração, ventilação, ar condicionado e aquecimento que se encontra a principal e mais freqüente aplicação da Psicrometria.

O projeto de sistemas de refrigeração e ar condicionado é feito com o auxílio de Diagramas Psicrométricos padronizados para determinadas pressões (ou altitude barométricas) e faixas de umidade e temperatura específicos, o que traz pouca flexibilidade uma vez que para pressões diferentes das padronizadas deve ser feita uma interpolação dos dados das tabelas padrão.

Desta forma, este projeto teve por objetivo o desenvolvimento de um programa computacional para o cálculo de todas as propriedades do ar úmido a partir de uma trinca de dados de entrada. O programa é desenvolvido em linguagem Delphi para ambiente Windows 95/98.

Para a elaboração das rotinas de cálculos das propriedades foi utilizada a abordagem realista, que permite que o programa seja utilizado para praticamente qualquer pressão de trabalho (até 50 atm) e temperatura (até 100°C).

2 – Fundamentação Teórica

O *ar atmosférico* é o resultado de um grande número de constituintes gasosos. O *ar seco* é definido como o ar atmosférico de cuja composição se exclui o vapor d'água. Da mistura do ar seco com o vapor d'água temos o *ar úmido*. Desta forma, a ar à nossa volta é o ar úmido.

2.1 - Parâmetros Psicrométricos

Umidade Absoluta (ω): razão entre as massas de vapor de água e de ar seco presentes na mistura.

Umidade Relativa (ϕ): razão entre a fração molar do vapor de água presente na mistura e a fração molar que o vapor de água teria se a mistura estivesse saturada à mesma temperatura e pressão total.

Grau de Saturação (μ): razão entre a umidade absoluta do ar e a umidade absoluta do ar saturado nas mesmas condições de pressão e temperatura.

Nota: as propriedades específicas, como volume e entalpia, são referidas à massa de ar seco e não à massa da mistura, devido ao fato de que o fluxo de ar seco é mantido constante enquanto que o vapor de água é adicionado ou retirado de ar úmido.

Volume específico (v): razão entre o volume ocupado pela mistura e a massa de ar seco presente.

Entalpia específica (h): razão entre a entalpia total da mistura e a massa de ar seco.

Temperatura de Bulbo Seco (TBS): temperatura da mistura indicada por um termômetro.

Temperatura de Orvalho (Torv): temperatura correspondente ao ponto de início da condensação do vapor de água contido no ar úmido, quando o processo de resfriamento é isobárico.

Temperatura de Bulbo Úmido (TBU): temperatura da mistura indicada por um termômetro cujo bulbo é envolvido por uma mecha enbebida em água destilada.

2.2 – A Equação Virial

A Equação Virial de estado é uma das principais correlações usadas no programa para o cálculo das propriedades, juntamente com a sua derivada. A equação virial descreve o comportamento da pressão, temperatura e volume específico de cada componente da mistura *ar seco + vapor de água* e está formulada da seguinte maneira:

$$\frac{P\bar{v}}{RT} = 1 + \frac{B_{ii}}{\bar{v}} + \frac{C_{iii}}{\bar{v}^2}$$

onde i pode ser a para o ar seco ou w para o vapor de água.

Para a mistura (*ar úmido*) temos:

$$B_m = x_a^2 \cdot B_{aa} + 2 \cdot x_a \cdot x_w \cdot B_{aw} + x_w^2 \cdot B_{ww}$$

$$C_m = x_a^3 \cdot C_{aaa} + 3 \cdot x_a^2 \cdot x_w \cdot C_{aaw} + 3 \cdot x_a \cdot x_w^2 \cdot C_{aww} + x_w^3 \cdot C_{www}$$

B_{aa} , B_{aw} , B_{ww} , C_{aaa} , C_{aaw} , C_{aww} , C_{www} são os coeficientes viriais, que são função da temperatura de bulbo seco (TBS) da mistura. x_a e x_w são as frações molares de ar seco e vapor.

2.3 - A Abordagem Realista (“fator de intensificação”)

Na formulação para gás real, a pressão parcial do vapor de água, P_{vs} , é multiplicada pelo fator de intensificação, f . Nesse fator, estão incluídos todos os efeitos não ideais da mistura saturada ar seco e vapor de água. Desta forma, a umidade absoluta do ar úmido saturado, ω_s , é calculada por:

$$\omega_s = 0,62198 * \frac{fP_{vs}}{P - fP_{vs}}$$

A umidade relativa pode ser expressa como:

$$\phi = \frac{\mu}{1 - (1 - \mu)fP_{vs} / P} \quad \text{onde } \mu \text{ é dado por } \mu = \frac{\omega}{\omega_s}$$

O fator de intensificação é função apenas da temperatura e pressão da mistura. A seguir temos o equacionamento do fator de intensificação apresentado por Hyland e Wexler:

$$\begin{aligned}
\ln f = & \left[\frac{(1 + kp)(P - P_{vs}) - \frac{1}{2}k(P^2 - P_{vs}^2)}{RT} \right] \bar{v} + \ln(1 - kx_{as}P) + \left[\frac{x_{as}^2 P}{RT} \right] B_{aa} - \\
& \left[\frac{2x_{as}^2 P}{RT} \right] B_{aw} - \left[\frac{(P - P_{vs} - x_{as}^2 P)}{RT} \right] B_{ww} + \left[\frac{x_{as}^3 P^2}{(RT)^2} \right] C_{aaa} + \left[\frac{3x_{as}^2(1 - 2x_{as})P^2}{2(RT)^2} \right] C_{aaw} - \\
& \left[\frac{3x_{as}^2(1 - x_{as})P^2}{(RT)^2} \right] C_{aww} - \left[\frac{(1 + 2x_{as})(1 - x_{as})^2 P^2 - P_{vs}^2}{2(RT)^2} \right] C_{www} - \\
& \left[\frac{x_{as}^2(1 - 3x_{as})(1 - x_{as})P^2}{(RT)^2} \right] B_{aa}B_{ww} - \left[\frac{2x_{as}^3(2 - 3x_{as})P^2}{(RT)^2} \right] B_{aa}B_{aw} + \\
& \left[\frac{6x_{as}^2(1 - x_{as})^2 P^2}{(RT)^2} \right] B_{ww}B_{aw} - \left[\frac{3x_{as}^2 P^2}{2(RT)^2} \right] B_{aa}^2 - \left[\frac{2x_{as}^2(1 - x_{as})(1 - 3x_{as})P^2}{(RT)^2} \right] B_{aw}^2 - \\
& \left[\frac{P_{vs}^2 - (1 + 3x_{as})(1 - x_{as})^3 P^2}{2(RT)^2} \right] B_{ww}^2
\end{aligned}$$

O gráfico abaixo mostra que para pressões elevadas o fator de intensificação assume valores bastante significativos:

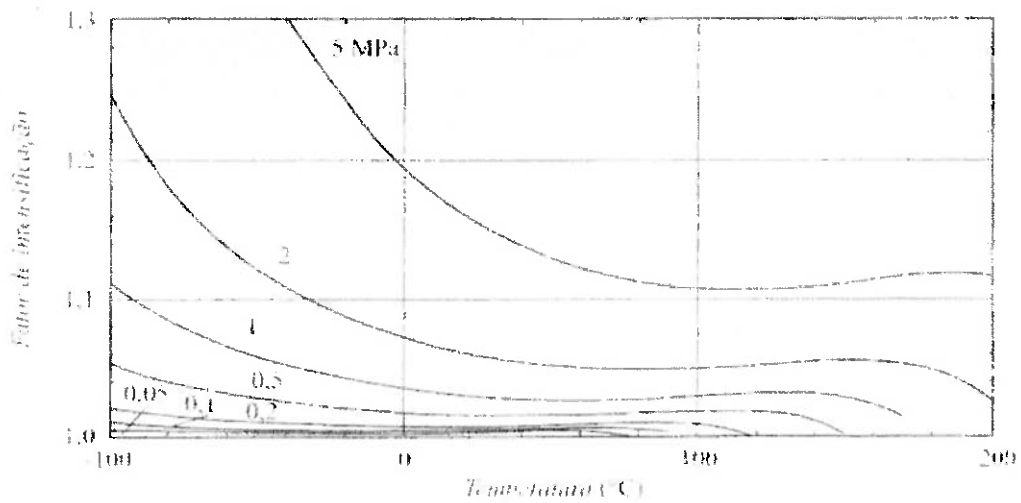


Figura 1

2.4 – Métodos Numéricos

Os algoritmos desenvolvidos para as rotinas de cálculo das propriedades utilizaram basicamente dois métodos numéricos: o *método da Dicotomia* e o *método de Newton-Raphson*. O primeiro apresenta uma confiabilidade muito boa para uma aproximação inicial e convergência garantida porém muito lenta, já o segundo apresenta uma convergência bastante rápida, porém necessita de uma aproximação inicial muito boa para garantir a convergência. Desta forma, várias vezes utilizou-se os dois métodos combinados, usando a Dicotomia para a aproximação inicial e Newton-Raphson para convergência ao valor final.

Método da Dicotomia:

Este método é baseado no Teorema de Bolzano: seja a raiz de uma função f isolada no intervalo $[a,b]$, portanto $f(a)*f(b) < 0$. O método consiste em dividir o intervalo dado ao meio e por aplicação do Teorema de Bolzano nos subintervalos $[a, (a+b)/2]$ e $[(a+b)/2, b]$ descobrir qual deles contém a raiz. O processo é repetido para o novo subintervalo, e assim sucessivamente, até que se obtenha a precisão desejada, ou seja, o intervalo seja menor ou igual a 2 vezes a precisão desejada.

Método de Newton-Raphson:

A idéia deste método é construir uma função $\phi(x)$ para a qual exista um intervalo contendo a raiz onde $|\phi'(x)| < 1$. Para isso impõe-se $\phi'(\bar{x}) = 0$. Como $\phi'(x)$ deve ser uma função contínua, existe sempre uma vizinhança I de $\bar{x}$ onde $\max |\phi'(x)| < 1$.

A forma mais geral de $x = \phi(x)$ equivalente a $f(x) = 0$ é dada por :

$$x = x + A(x)f(x) = \phi(x)$$

onde $A(x)$ é uma função contínua qualquer, tal que $A(\bar{x}) \neq 0$. Escolhe-se $A(x)$ de forma a ter-se $\phi'(\bar{x})=0$. Assim:

$$\phi'(x) = 1 + A(x)f'(x) + A'(x)f(x)$$

No ponto $\bar{x}$:

$$\phi'(\bar{x}) = 1 + A(\bar{x})f'(\bar{x})$$

Supondo $f'(\bar{x}) \neq 0$, como queremos $\phi'(\bar{x})=0$:

$$A(\bar{x}) = -\frac{1}{f'(\bar{x})}$$

Uma escolha satisfatória para $A(x)$ é:

$$A(x) = -\frac{1}{f'(x)}$$

Obtemos então:

$$\phi(x) = x - \frac{f(x)}{f'(x)}$$

Assim o método de Newton-Raphson é definido por:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

3 – Desenvolvimento do Programa

Por se tratar de uma mistura de dois gases (*ar seco* e *vapor de água*), para se definir o estado termodinâmico da mistura necessita-se do conhecimento de três propriedades. Assim, o programa foi desenvolvido de modo aceitar quatro diferentes combinações de dados de entrada:

1. P, TBS e TBU
2. P, TBS e ϕ
3. P, TBS e ω
4. P, TBS e h

Estas combinações foram escolhidas de forma a representar casos práticos de utilização e medição das propriedades psicrométricas: a pressão -P- pode ser facilmente medida com um barômetro ou um transdutor de pressão; a temperatura de bulbo seco -TBS- pode ser obtida com um termopar; a temperatura de bulbo úmido -TBU- é obtida num psicrômetro ou termopar cujo bulbo é envolvido por uma mecha enbebida em água destilada (caso 1); a umidade relativa - ϕ - pode ser lida num transdutor de umidade relativa (caso 2); e a umidade absoluta - ω - é obtida com a ajuda de um higrômetro ou transdutor de umidade absoluta (caso 3). Além disso, as duas primeiras combinações são bastante úteis para o cálculo de sistemas de refrigeração, condicionamento de ar e ventilação, onde dados de projeto são TBS e TBU ou TBS e ϕ . A última combinação é útil para processos de condicionamento de ar, refrigeração e ventilação envolvendo cálculos de carga térmica (caso 4).

As rotinas de cálculos foram baseadas nos algoritmos desenvolvidos pelo Prof. J. R. Simões Moreira em linguagem FORTRAM.

As rotinas desenvolvidas foram as seguintes:

- Pwsat: calcula a pressão parcial do vapor de água saturado para uma dada temperatura de entrada.
- Virial: calcula os coeficientes da equação virial para uma dada temperatura de entrada.
- Dervir: calcula os coeficientes da derivada da equação virial para uma dada temperatura de entrada.
- Mistura: calcula os coeficientes viriais da mistura para uma dada fração molar de ar seco.
- Volume: calcula numericamente (dicotomia) o volume específico da mistura para uma temperatura e pressão de mistura dadas.
- Entalpia: calcula a entalpia específica total da mistura a partir dos seguintes dados de entrada: pressão da mistura, temperatura, fração molar do ar seco e volume específico da mistura.
- Enhance: esta rotina traz o cálculo numérico do fator de intensificação e da fração molar do ar seco saturado, além do fator de compressibilidade e constante da Lei de Henry.
- Hagua: calcula a entalpia da água à uma dada temperatura.
- Tsat: calcula a temperatura de saturação para uma temperatura e pressão de vapor dados.

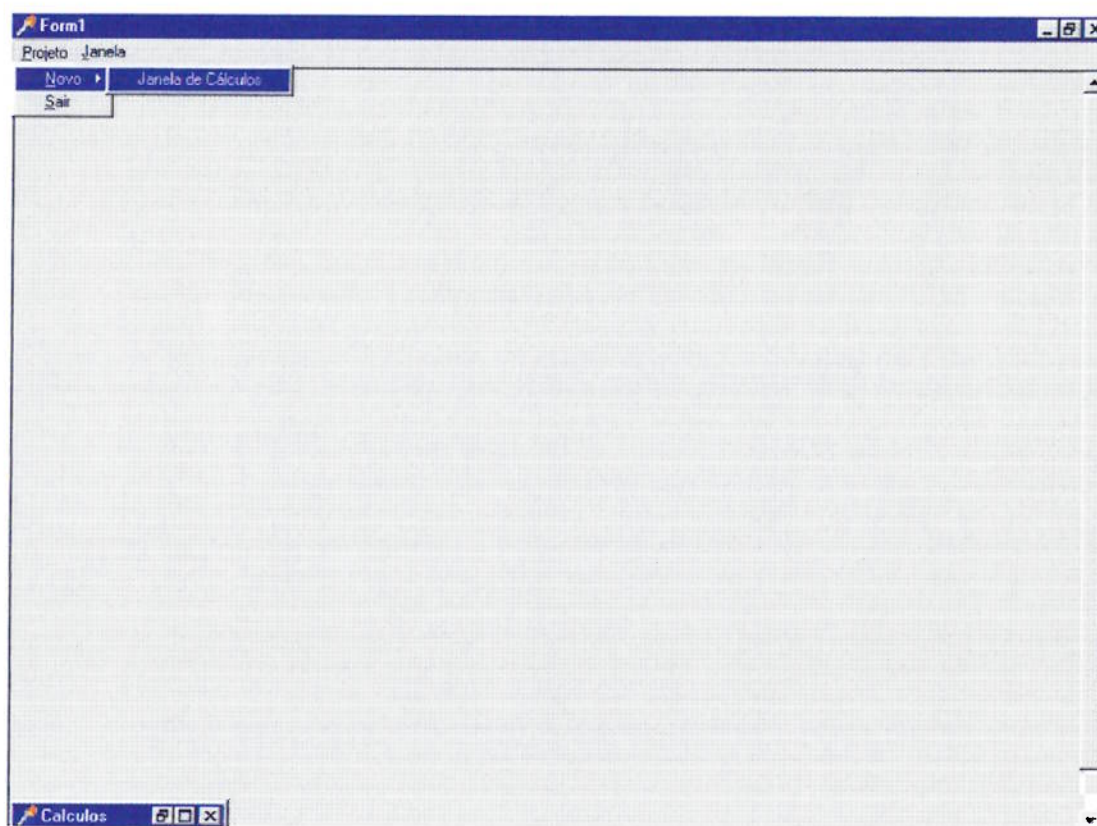
- Fzero: esta rotina calcula o Fzero do método numérico Newton-Raphson.
- FSAD: calcula a função do saturador adiabático para o caso de TBS e TBU fornecidos.
- FSAD2: calcula a função do saturador adiabático para o caso de TBS e phi fornecidos.
- TesteConsistencia: faz uma verificação da consistência dos dados de entrada do programa. (A temperatura não pode ser maior ou muito próxima à temperatura de saturação.)
- Calcula_para_TBSePhi: faz o cálculo de todas as propriedades através do método de Newton-Raphson a partir da pressão, temperatura e umidade relativa como dados de entrada.
- Calcula_para_TBSeTBU: faz o cálculo de todas as propriedades através do método de Newton-Raphson a partir da pressão, temperatura, umidade e temperatura de bulbo úmido como dados de entrada.
- Calcula_para_TBSeW: utilizando as duas rotinas acima faz o cálculo de todas as propriedades a partir da pressão, temperatura e umidade absoluta como dados de entrada.
- Calcula_para_TBSeh: utilizando as duas rotinas supracitadas faz o cálculo de todas as propriedades a partir da pressão, temperatura e entalpia específica como dados de entrada.

4 – Características do Programa

O programa foi desenvolvido no compilador visual Borland Delphi 4.0 com suporte a ambiente Windows 95/98/NT.

O programa está estruturado em dois *forms*: um principal, onde pode-se abrir uma nova janela de cálculos ou sair do programa, ou ainda reorganizar as diversas janelas abertas simultaneamente e o *form* Cálculos onde o usuário insere os dados de entrada e onde são mostrados os resultados. Foi utilizado o conceito de *Multiple Forms* para que várias janelas de cálculos pudessem ser abertas simultaneamente, facilitando ao usuário a comparação de resultados de cálculos diferentes.

Exemplos:



Main Form

Form1 - [Calculos]

Projeto Janela

Pressão = kPa

Temperatura de Bulbo Seco = °C

Temperatura de Bulbo Úmido = °C

☒ TBS e TDU
☐ TBS e ϕ
☐ TBS e W
☐ TBS e h
☐ W e h

Ok

Pressão = kPa
 Temperatura de Bulbo Seco = °C
 Temperatura de Bulbo Úmido = °C
 Temperatura de Orvalho = °C
 Umidade Relativa = %
 Grau de Saturação = %
 Umidade Absoluta = kg vapor/kg ar
 Entalpia Específica = kJ/kg ar
 Volume Específico = m³/kg ar
 Fração Molar de Vapor =
 Fator de Intensificação =
 Coeficiente do Psicrômetro = 1/°C

Form Calculos

Form1

Projeto Janela

Cálculos

- Lado a lado
- Cascata
- 1 Calculos
- 2 Calculos_1
- 3 Calculos_2
- 4 Calculos_3
- 5 Calculos_4
- 6 Calculos_5
- 7 Calculos_6

Calculos_3

Pressão = kPa

Temperatura de Bulbo Seco = °C

Calculos_5

Pressão = kPa

Temperatura de Bulbo Seco = °C

Calculos_4

Pressão = kPa

Temperatura de Bulbo Seco = °C

Calculos_1

Pressão = kPa

Temperatura de Bulbo Seco = °C

Multiple Forms

Na janela de cálculos tomou-se como medida aceitar tanto o ponto como a vírgula como separador decimal. Quando o usuário digita qualquer caractere que não um algarismo ou (apenas um) ponto ou vírgula, o programa não reconhece o caractere evitando-se assim erros de digitação. Quando o usuário seleciona uma opção de combinação de dados de entrada, os títulos e unidades dos campos de entrada são alterados, e os valores dentro dos campos cuja propriedade mudou são apagados. A seguir são mostrados exemplos de utilização para as diversas combinações:

The screenshot shows a software window titled "Form1 - [Cálculos]" with a menu bar containing "Projeto" and "Janela". The window has a green background and contains the following elements:

- Input Fields (Left Column):**
 - Pressão: 101.325 kPa
 - Temperatura de Bulbo Seco: 25 °C
 - Temperatura de Bulbo Úmido: 20 °C
- Radio Buttons (Left Column):**
 - ☒ TBS e TBU
 - ☐ TBS e phi
 - ☐ TBS e W
 - ☐ TBS e h
 - ☐ W e h
- Buttons (Left Column):**
 - Ok
- Output Fields (Right Column):**
 - Pressão = 101.325 kPa
 - Temperatura de Bulbo Seco = 25.00 °C
 - Temperatura de Bulbo Úmido = 20.00 °C
 - Temperatura de Orvalho = 17.73 °C
 - Umidade Relativa = 64.09 %
 - Grau de Saturação = 63.29 %
 - Umidade Absoluta = 0.012765 kg vapor/kg ar
 - Entalpia Específica = 57.655 kJ/kg ar
 - Volume Específico = 0.8617 m³/kg ar
 - Fração Molar de Vapor = 0.0201
 - Fator de Intumescência = 1.0042
 - Constante do Psicrômetro = 0.00001102 1/°C

TBS e TBU

Form1 - [Calculos]

Projeto - Janela

Pressão = 101.325 kPa

Temperatura de Bulbo Seco = 48 °C

Umidade Relativa = 78 %

☐ TBS e TBU
☒ TBS e phi
☐ TBS e W
☐ TBS e h
☐ W e h

Ok

Pressão = 101.325 kPa
 Temperatura de Bulbo Seco = 48.00 °C
 Temperatura de Bulbo Úmido = 43.69 °C
 Temperatura de Orvalho = 43.15 °C
 Umidade Relativa = 77.99 %
 Grau de Saturação = 75.91 %
 Umidade Absoluta = 0.058863 kgvapor/kgar
 Entalpia Específica = 200.648 kJ/kgar
 Volume Específico = 0.9954 m³/kgar
 Fração Molar de Vapor = 0.0865
 Fator de Intermittência = 1.0052
 Constante do Psicrômetro = 0.00000691 1/°C

TBS e ϕ

Form1 - [Calculos]

Projeto - Janela

Pressão = 101.325 kPa

Temperatura de Bulbo Seco = 10 °C

Umidade Absoluta = 0.0012 kgvapor/kgar

☐ TBS e TBU
☐ TBS e phi
☒ TBS e W
☐ TBS e h
☐ W e h

Ok

Pressão = 101.325 kPa
 Temperatura de Bulbo Seco = 10.00 °C
 Temperatura de Bulbo Úmido = 1.92 °C
 Temperatura de Orvalho = -9.62 °C
 Umidade Relativa = 15.84 %
 Grau de Saturação = 15.68 %
 Umidade Absoluta = 0.001201 kgvapor/kgar
 Entalpia Específica = 13.084 kJ/kgar
 Volume Específico = 0.8033 m³/kgar
 Fração Molar de Vapor = 0.0019
 Fator de Intermittência = 1.0039
 Constante do Psicrômetro = 0.00001787 1/°C

TBS e ω

Form1 - [Calculos]

Projeto Janela

Pressão = 101.325 kPa

Temperatura de Bulbo Seco = 15 °C

Entalpia específica = 34.391 kJ/kg

☐ TBS = TBU
☐ TBS = pht
☐ TBS = W
☒ TBS = h
☐ W e h

Ok

Pressão = 101.325 kPa
 Temperatura de Bulbo Seco = 15.00 °C
 Temperatura de Bulbo Úmido = 12.00 °C
 Temperatura de Orvalho = 9.95 °C
 Umidade Relativa = 71.77 %
 Grau de Saturação = 71.43 %
 Umidade Absoluta = 0.007636 kg/m³
 Entalpia Específica = 34.391 kJ/kg
 Volume Específico = 0.8260 m³/kg
 Fração Molar de Vapor = 0.0121
 Fator de Intensificação = 1.0039
 Constante do Psicrômetro = 0.00000640 1/°C

TBS e h

5 – Conclusões

Considero a realização deste projeto muito importante como parte do final de minha graduação. O contato profundo com a área da psicrometria e condicionamento de ar foi valiosíssimo, uma vez que me permitiu entender e entrar em contato com conceitos não vistos a fundo durante o curso e que me serão muito úteis na vida profissional. O utilização de uma ferramenta de programação como o Delphi, também agregou importantes conhecimentos, além da perspectiva futura de utilização desta ferramenta sempre que possível e/ou necessário, pois sabemos que o uso do computador para realização de cálculos complexos com elevado número de iterações é indispensável, além de que uma ferramenta com uma enorme variedade de recursos gráficos como esta, torna bastante fácil a elaboração de programas com uma interface *user friendly*.

A segunda parte do programa proposta inicialmente, o módulo para construção do Diagrama Psicrométrico, não pode ser implementada devido a ausência de tempo hábil por minha parte, uma vez que o projeto foi desenvolvido individualmente e, por estar também trabalhando sem poder dedicar-me integralmente ao projeto. Mas o projeto não só permite como considera futuramente a continuidade do trabalho por outros alunos com a inclusão de vários novos módulos, como Construção do Diagrama Psicrométrico, Análise dos Processos Básicos do Condicionamento de Ar traçados sobre o diagrama, Ciclo completo de Condicionamento de Ar sobre o Diagrama, etc. tornando-se assim uma ferramenta bastante completa e poderosa no auxílio ao projeto de sistemas de Condicionamento do Ar, Ventilação, Controle de Ambiente.

Referências Bibliográficas

Simões Moreira, José R.

Fundamentos e Aplicações da Psicrometria

São Paulo, RPA, 1999

Humes / Melo / Yoshida / Martins

Noções de Cálculo Numérico

São Paulo, McGraw-Hill, 1984

Hyland, R. W. e Wexler, A.

*Formulations for the Thermodynamic Properties of Dry Air from
173.15K to 473.15K, and Saturated Moist Air from 173.15K to 372.15K,
at pressures to 5MPa*

Ashrae Transactions V.89 (2A), 1983

Van Wylen, G., Sonntag, R. e Borgnakke, C.

Fundamentos da Termodinâmica Clássica

São Paulo, Edgard Blücher, 1995

Apêndice – Listagem da Rotina de Cálculos

```
unit CalculosGReal;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,  
StdCtrls, Mask, Math;
```

```
type
```

```
TCalculos = class(TForm)  
    Info2: TEdit;  
    Pressao: TEdit;  
    TBS_TBU: TRadioButton;  
    TBS_phi: TRadioButton;  
    Label1: TLabel;  
    Label2: TLabel;  
    Button1: TButton;  
    Label3: TLabel;  
    Info1: TEdit;  
    Info4: TEdit;  
    Info5: TEdit;  
    Info6: TEdit;  
    Info7: TEdit;  
    Info8: TEdit;  
    Info10: TEdit;  
    Info11: TEdit;  
    Info12: TEdit;  
    Info13: TEdit;  
    Info14: TEdit;  
    Info15: TEdit;  
    Info9: TEdit;  
    TBSLabel: TLabel;  
    PLabel: TLabel;  
    TBULabel: TLabel;  
    TorvLabel: TLabel;
```

```

phiLabel: TLabel;
miLabel: TLabel;
WLabel: TLabel;
HLabel: TLabel;
VLabel: TLabel;
XvLabel: TLabel;
FSLabel: TLabel;
TetaLabel: TLabel;
Label4: TLabel;
Label5: TLabel;
Label6: TLabel;
Label7: TLabel;
Label8: TLabel;
Label9: TLabel;
Label10: TLabel;
Label11: TLabel;
Label12: TLabel;
Label13: TLabel;
Label14: TLabel;
Label15: TLabel;
Label18: TLabel;
TBS_W: TRadioButton;
TBS_h: TRadioButton;
procedure TBS_TBUClick(Sender: TObject);
procedure TBS_phiClick(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure PressaoKeyPress(Sender: TObject; var Key: Char);
procedure Info1KeyPress(Sender: TObject; var Key: Char);
procedure Info2KeyPress(Sender: TObject; var Key: Char);
procedure TBS_WClick(Sender: TObject);
procedure TBS_hClick(Sender: TObject);
procedure W_hClick(Sender: TObject);

private
    { Private declarations }
public
    { Public declarations }
end;

```

const

AJ100: array [1..7] of real = (0.5088496E2,0.6163813,0.1459187E-2,0.2008438E-4,-0.5847727E-7,0.410411E-9,0.1967348E-1);

AJ200: array [1..7] of real = (0.50884917E2,0.62590623,0.13848668E-2,0.21603427E-4,-0.72087667E-7,0.46545054E-9,0.19859983E-1);

AF: array [1..8] of real = (-0.2403360201E4,-0.140758895E1,0.1068287657,-0.2914492351E-3,0.373497936E-6,-0.21203787E-9,-0.3424442728E1,0.1619785E-1);

HEN: array [1..2,1..5] of real = ((-0.0005943,-0.1470,-0.0512,-0.1076,0.8447),(-0.1021,-0.1482,-0.1900,-0.03741,0.8510));

ENTAR: array [1..6] of real = (0.63290874E1,0.28709015E2,0.26431805E-2,-0.10405863E-4,0.18660410E-7,-0.97843331E-11);

ENTVAP: array [1..6] of real = (-0.5008E-2,0.32491829E2,0.65576345E-2,-0.26442147E-4,0.51751789E-7,-0.31541624E-10);

var

Calculos: TCalculos;

P: real;

Pws: real;

Pwstar: real;

Pv: real;

T: real;

Tstar: real;

Torv: real;

T00: real;

phi: real;

V: real;

Vmol: real;

Hmol: real;

Hseco: real;

FS: real;

FStar: real;

Xa,Xas: real;

Xastar: real;

Xv, Xvs: real;

Xvstar: real;

H: real;

HLstar, Hstar: real;

R, RT, RT2: real;

W: real;

```

Win, Hin: real;
Ws, Wstar: real;
mi: real;
ATETA: real;
// coeficientes viriais
baa, baw, bww, caaa, caaw, caww, cwww: real;
// coeficientes viriais (derivadas)
dbaa, dbaw, dbww, dcaaa, dcaaw, dcaww, dcwww: real;
// coeficientes da mistura
bm, cm, Xa2, Xa3, Xw, Xw2, Xw3: real;
teste: boolean;
straux: string;
DecimalSep: boolean;

implementation

{$R *.DFM}

//*****//

function Pwsat(T:real): real;
const
  Gliq : array [1..6] of real = (-0.58002206E4,0.13914993E1,-0.48640239E-1,0.41764768E-4,-
0.14452093E-7,0.65459673E1);
  Gelo : array [1..7] of real = (-0.56745359E4,0.63925247E1,-0.9677843E-2,0.62215701E-
6,0.20747825E-8,-0.9484024E-12,0.41635019E1);
var
  i, j: integer;
  Pws: real;
begin
  Pws:=0;
  if T<273.16 then begin
    i:=1;
    while (i<7) do begin
      j:=i-1;
      Pws:=Pws+Gelo[i]*Power(T,j-1);
      i:=i+1
    end;
    Pws:=Pws+Gelo[7]*Ln(T);
  end;

```

```

end
else begin
    i:=1;
    while (i<6) do begin
        Pws:=Pws+Gliq[i]*Power(T,i-2);
        i:=i+1
    end;
    Pws:=Pws+Gliq[6]*Ln(T)
end;
Pwsat:= Exp(Pws);
end;

//*****//

procedure Virial(T: real);
var
    T2, T3, T4: real;
begin
    T2:=T*T;
    T3:=T2*T;
    T4:=T3*T;
    R:=0.831441E7;
    RT:=R*T;
    RT2:=RT*RT;
    baa:=0.349568E2 - 0.668772E4/T - 0.210141E7/T2 + 0.924746E8/T3;
    baw:=0.32366097E2 - 0.141138E5/T - 0.1244535E7/T2 - 0.2348789E10/T4;
    bww:=(0.7E-8 - 0.147184E-8*Exp(1734.29/T))*RT;
    caaa:=0.125975E4 - 0.190905E6/T + 0.632467E8/T2;
    caaw:=0.482737E3 + 0.105678E6/T - 0.656394E8/T2 + 0.294442E11/T3 - 0.319317E13/T4;
    caww:=Exp(-0.10728876E2 + 0.347802E4/T - 0.383383E6/T2 + 0.33406E8/T3);
    caww:=-1E6*caww;
    cwww:=0.104E-14 - 0.335297E-17*Exp(3645.09/T);
    cwww:=cwww*RT2+bww*bww;
end;

//*****//

procedure Dervir(T: real);
var

```

```

T2, T3, T4, T5: real;
begin
  T2:=T*T;
  T3:=T2*T;
  T4:=T3*T;
  T5:=T4*T;
  RT:=R*T;
  RT2:=RT*RT;
  dbaa:= 0.668772E4/T2 + 0.420282E7/T3 - 0.2774238E9/T4;
  dbaw:= 0.141138E5/T2 + 0.248907E7/T3 + 0.9395156E10/T5;
  dbww:= bww/T + (0.25531861E-5*R*Exp(1734.29/T))/T;
  dcaaa:= 0.190905E6/T2 - 0.1264934E9/T3;
  dcaaw:= -0.105678E6/T2 + 0.1312788E9/T3 - 0.88332600E11/T4 + 0.1277268E14/T5;
  dcaww:=caww*(-0.347802E4/T2 + 0.766766E6/T3 - 0.100218E9/T4);
  dcwww:=2*R*R*T*(0.104E-14 - 0.335297E-17*Exp(3645.09/T));
  dcwww:= dcwww + 0.12221877E-13*R*R*Exp(3645.09/T);
  dcwww:= dcwww+2*bww*dbww;
end;

```

```

//*****//

```

```

procedure Mistura(Xa: real);
begin
  Xa2:=Xa*Xa;
  Xa3:=Xa2*Xa;
  Xw:=1-Xa;
  Xw2:=Xw*Xw;
  Xw3:=Xw2*Xw;
  bm:=Xa2*baa + 2*Xa*Xw*baw + Xw2*bww;
  cm:=Xa3*caaa + 3*Xa2*Xw*caaw + 3*Xa*Xw2*caww + Xw3*cwww;
end;

```

```

//*****//

```

```

procedure Volume (T, P: real);
var
  oldv, newv: real;
begin
  R:=0.831441E7;

```

```

oldv:=R*T/P;
repeat
    newv:=(R*T*(1 + bm/oldv + cm/(oldv*oldv)))/P;
    if (Abs((newv-oldv)/newv)>1E-5) then
        oldv:=(newv+oldv)/2;
    until (Abs((newv-oldv)/newv)<=1E-5);
    Vmol:=(oldv+newv)/2;
end;

//*****//

procedure Entalpia (P, T, Xa, Vmol: real);
var
    enta, entv, Taux: real;
    dbm, dcm: real;
    Hmol1, Hmol2, Hmol3: real;
    i: integer;
begin
    enta:=0;
    entv:=0;
    i:=1;
    while i<7 do begin
        Taux:=Power(T,i-1);
        enta:=enta + ENTAR[i]*Taux;
        entv:=entv + ENTVAP[i]*Taux;
        i:=i+1;
    end;
    dbm:=Xa2*dbaa + 2*Xa*Xw*dbaw + Xw2*dbww;
    dcm:=Xa3*dcaaa + 3*Xa2*Xw*dcaaw + 3*Xa*Xw2*dcaww + Xw3*dcwww;
    Hmol1:=Xa*(enta-7914.1982) + Xw*(entv+35994.17);
    Hmol2:=(RT*(bm - T*dbm))/Vmol;
    Hmol3:=(RT*(cm - 0.5*T*dcm))/(Vmol*Vmol);
    Hmol:=Hmol1 + (Hmol2 + Hmol3)*1E-6;
end;

//*****//

```

```

procedure Enhance (P, Pws, T: real; var Xas: real);
var

```

```

// fator de compressibilidade
betak: real;
// temperatura em Celsius
TC: real;
// constante da Lei de Henry
Khen: real;
// volume espec. molar da agua liquido ou solido
VI: real;
P2, Pws2: real;
Tal: real;
FSA: real;
Xas2: real;
bo, bn, co, cn, ako, akn: real;
i,j: integer;
begin
    betak:=0;
    TC:=T-273.15;
    if TC<=0 then
        betak:=(8.875 + 0.0165*TC)*1E-11
    else
        if TC>=100 then begin
            i:=1;
            while i<7 do begin
                j:=7-i;
                betak:=betak + AJ200[j]*Power(TC,j-1);
                i:=i+1;
            end;
            betak:=betak*1E-11/(1+AJ200[7]*TC);
        end
        else begin
            i:=1;
            while i<7 do begin
                j:=7-i;
                betak:=betak + AJ100[j]*Power(TC,j-1);
                i:=i+1;
            end;
            betak:=betak*1E-11/(1+AJ100[7]*TC);
        end;
    if T<=273.15 then

```

```

    Khen:=0
else begin
    Tal:=1000/T;
    bo:=HEN[1,3]*Tal + HEN[1,4];
    bn:=HEN[2,3]*Tal + HEN[2,4];
    co:=HEN[1,2]*Tal*Tal + HEN[1,5]*Tal - 1;
    cn:=HEN[2,2]*Tal*Tal + HEN[2,5]*Tal - 1;
    ako:=(-Sqrt(bo*bo - 4*HEN[1,1]*co) - bo)/(2*HEN[1,1]);
    akn:=(-Sqrt(bn*bn - 4*HEN[2,1]*cn) - bn)/(2*HEN[2,1]);
    ako:=Power(10,ako);
    akn:=Power(10,akn);
    Khen:=0.22/ako + 0.78/akn;
    Khen:=Khen*1E-4/101325;
end;
VI:=0;
if T<=273.15 then
    VI:=0.1070003E-2 - 0.249936E-7*T + 0.371611E-9*T*T
else begin
    i:=1;
    while i<7 do begin
        j:=i-1;
        VI:=VI+AF[i]*Power(T,J);
        i:=i+1;
    end;
    VI:=(AF[7]+AF[8]*T)/VI;
end;
// transforma para a base molar. O valor acima e m3/kg
VI:=VI*18015.28;

// começa a iterar para buscar os valores de XAS e FS
FS:=1;
Pws2:=Pws*Pws;
P2:=P*P;
repeat
    FSA:=FS;
    Xas:=(P-FSA*Pws)/P;
    Xas2:=Xas*Xas;
    FS:=((1 + betak*Pws)*(P-Pws) - betak*(P2-Pws2)/2)*VI/RT;
    FS:=Ln(1 - Khen*Xas*P)+ FS;

```

```

    FS:=Xas2*P*baa/RT + FS;
    FS:=FS - 2*Xas2*P*baw/RT;
    FS:=FS - bww*(P - Pws - Xas2*P)/RT;
    FS:=Xas2*Xas*P2*caaa/RT2 + FS;
    FS:=3*Xas2*(1 - 2*Xas)*P2*caaw/(2*RT2)+ FS;
    FS:=FS - 3*Xas2*(1 - Xas)*P2*caww/RT2;
    FS:=FS - ((1 + 2*Xas)*Power((1 - Xas),2)*P2 - Pws2)*cwww/(2*RT2);
    FS:=FS - Xas2*(1 - 3*Xas)*(1 - Xas)*P2*baa*bww/RT2;
    FS:=FS - (2*Xas2*Xas*(2 - 3*Xas)*P2)*baa*baw/RT2;
    FS:=6*Xas2*Power((1 - Xas),2)*P2*bww*baw/RT2 + FS;
    FS:=FS -(3*Xas2*Xas2*P2)*baa*baa/(2*RT2);
    FS:=FS - 2*Xas2*(1-Xas)*(1-3*Xas)*P2*baw*baw/RT2;
    FS:=FS - (Pws2 -(1 + 3*Xas)*Power((1 - Xas),3)*P2)*bww*bww/(2*RT2);
    FS:=Exp(FS);
    until(Abs((FS-FSA)/FS)<1E-6);
    Xas:=(P-FS*Pws)/P;
end;

```

//**

```

function Hagua (T: real): real;
begin
    if T<=0.01 then
        Hagua:=3.70E-3*Power(T,2) + 2.103*T - 333.43
    else
        Hagua:=9.637E-5*Power(T,2) + 4.187*T;
    end;
end;

```

//**

```

function Tsat(T, Pv: real): real;
var
    Ta, Taa, Tol, Tn: real;
    Pa, Paa: real;
begin
    Ta:=T;
    Taa:=T - 10;
    Tol:=5e-4;
    Pa:=Pwsat(Ta) - Pv;

```

```

Paa:=Pwsat(Taa) - Pv;
Tn:=Ta - Pa*(Ta - Taa)/(Pa - Paa);
repeat
  if (Abs(Tn-Ta)>=Tol) then begin
    Taa:=Ta;
    Paa:=Pa;
    Ta:=Tn;
    Pa:=Pwsat(Tn) - Pv;
    Tn:=Ta - Pa*(Ta - Taa)/(Pa - Paa);
  end;
until (Tn-Ta)<Tol;
Tsats:=Tn;
end;

//*****//

function FZero (P, T, Hseco: real): real;
var
  Hl: real;
begin
  Pws:=Pwsat(T);
  Virial(T);
  Enhance(P,Pws,T,Xas);
  Dervir(T);
  Mistura(Xas);
  Volume(P,T);
  Entalpia(P,T,Xas,Vmol);
  W:=0.62198*(FS*Pws/(P-FS*Pws));
  H:=Hmol/(Xas*28.9645);
  Hl:=Hagua(T-273.15);
  FZero:=Hseco + W*Hl - H;
end;

//*****//

function FSAD(Xv, P, T, Xvs, WS, FS, Pws, Wstar: real): real;
begin
  Phi:=Xv/Xvs;
  mi:=phi*(P-FS*Pws)/(P-phi*FS*Pws);

```

```

W:=mi*Ws;
Mistura(1-Xv);
Volume(T,P);
Entalpia(P,T,1-Xv,Vmol);
H:=Hmol/((1-Xv)*28.9645);
FSAD:=H + (Wstar-W)*HLstar-Hstar;
end;

```

```

//*****//

```

```

function FSAD2(P, H, W, Tstar: real): real;
begin
    Virial(Tstar);
    Dervir(Tstar);
    Pwstar:=Pwsat(Tstar);
    Enhance(P,Pwstar,Tstar,Xastar);
    Xvstar:=(1-Xastar);
    Wstar:=0.62198*FStar*Pwstar/(P-FStar*Pwstar);
    Mistura(Xastar);
    Volume(P,Tstar);
    Entalpia(P,Tstar,Xastar,Vmol);
    Hstar:=Hmol/(Xastar*28.9645);
    HLstar:=Hagua(Tstar-273.15);
    FSAD2:=H + (Wstar - W)*HLstar - Hstar;
end;

```

```

//*****//

```

```

procedure TesteConsistencia (T, P: real);
begin
    teste:=true;
    Pv:=Pwsat(T);
    if Pv>=0.99*P then begin
        MessageDlg('Parâmetros P e T Incompatíveis !',mtError,[mbOK],1);
        teste:=false;
    end;
    if Tstar>1.001*T then begin
        MessageDlg('TBU não pode ser maior que TBS. Dados de entrada incorretos!',
            mtError,[mbOK],1);
    end;
end;

```

```

    teste:=false;
end;
end;

//*****//

procedure Calcula_para_TBSePhi (P, T, phi: real);
var
    Taux: real;
    DeltaT: real;
    i: integer;
    T0: real;
    F0, F00: real;
    Tseco: real;
    Xv0, Xv00: real;
    Xv1, Xv2: real;
    Xvstar1, Xvstar2: real;
    A1, A2: real;
begin
    Tseco:=150;
    Tstar:=T;
    Virial(T);
    Dervir(T);
    Pws:=Pwsat(T);
    Enhance(P,Pws,T,Xas);
    Xvs:=(1-Xas);
    Ws:=0.62198*FS*Pws/(P-FS*Pws);
    Xv:=phi*Xvs;
    W:=0.62198*Xv/(1-Xv);
    Xa:=1-Xv;
    Mistura(Xa);
    Volume(T,P);
    Entalpia(P,T,Xa,Vmol);
    H:=Hmol/(Xa*28.9645);
// método de N-R disparado com a função FSAD2
// Calcula as grandezas a temperatura T0 admitida (TBU termodinamica)
    T0:=T;
    F0:=FSAD2(P,H,W,T0);
// Calcula as grandezas a temperatura T00 admitida (TBU termodinamica)

```

```

T00:=0.99*T;
F00:=FSAD2(P,H,W,T00);
while (Abs(T00-T0)>1E-4) do begin
    Tstar:=(T0-F0*(T0-T00)/(F0-F00);
// Calcula as grandezas a temperatura TSTAR (TBU termodinamica)
    F0:=F0;
    F0:=FSAD2(P,H,W,Tstar);
    T00:=T0;
    T0:=Tstar;
end;
Tstar:=(T0+T00)/2;
// Nota importante: Uma vez achada Tstar, a subrotina torna-se identica ao caso acima (Case 1).
DeltaT:=1E-4*T;
Taux:=Tstar;
//3 Xv para fazer extrapolação junto com a função ATETA para o caso saturado
i:=1;
while i<=3 do begin
    if i<>3 then
        Tstar:=T - i*DeltaT;
    if i=3 then
        Tstar:=Taux;
    Virial(Tstar);
    Dervir(Tstar);
    Pwstar:=Pwsat(Tstar);
    Enhance(P,Pwstar,Tstar,Xastar);
    Xvstar:=(1-Xastar);
    FStar:=FS;
    Wstar:=0.62198*FStar*PWstar/(P-Fstar*PWstar);
    Mistura(Xastar);
    Volume(P,Tstar);
    Entalpia(P,Tstar,Xastar,Vmol);
    Hstar:=Hmol/(Xastar*28.9645);
    HLstar:=Hagua(Tstar-273.15);
//Atualiza os coeficientes viriais
    Virial(T);
    Dervir(T);
    Pws:=Pwsat(T);
    Enhance(P,Pws,T,Xas);
    Xvs:=(1-Xas);

```

```

Ws:=0.62198*FS*Pws/(P-FS*Pws);
//método de N-R disparado com a função FSAD
//chutes iniciais em Xv
Xv0:=Xvs*(Tstar-Tseco)/(T-Tseco);
F0:=FSAD(Xv0,P,T,Xvs,Ws,FS,Pws,Wstar);
Xv00:=0.9*Xv0;
Xv:=Xv0;
F00:=FSAD(Xv00,P,T,Xvs,Ws,FS,Pws,Wstar);
while Abs(Xv00-Xv0)>1.E-6 do begin
    Xv:=Xv0-F0*(Xv0-Xv00)/(F0-F00);
    F00:=F0;
    F0:=FSAD(Xv,P,T,Xvs,Ws,FS,Pws,Wstar);
    XV00:=XV0;
    Xv0:=XV;
end;
if i=1 then begin
    Xv1:=XV;
    Xvstar1:=XVstar;
end;
if i=2 then begin
    Xv2:=XV;
    Xvstar2:=XVstar;
end;
i:=i+1;
end;
if phi<>0 then
    phi:=Xv/Xvs
else
    Xv:=0;
mi:=phi*(P-FS*Pws)/(P-phi*FS*Pws);
W:=mi*Ws;
Xa:=1-XV;
Mistura(Xa);
Volume(T,P);
Entalpia(P,T,Xa,Vmol);
V:=Vmol/(Xa*28.9645);
H:=Hmol/(Xa*28.9645);
Tstar:=Tstar-273.15;
Pv:=P*Xv/FS;

```

```

    if Pv>0 then begin
        Torv:=T;
        Torv:=Tsat(Torv,Pv);
    end
    else
        Torv:=-200;
//calcula a constante do psicrometro
if (Tstar+273.15)<T-1 then
    ATETA:=(Xvstar-Xv)/(T-Tstar-273.15)
    else begin
        A1:=(Xvstar1-Xv1)/DeltaT;
        A2:=(Xvstar2-Xv2)/(2*DeltaT);
        ATETA:=A1+(A2-A1)/2;
    end;
    Tstar:=Tstar+273.15;
end;

//*****//

procedure Calcula_para_TBSeTBU (P, T, Tstar: real);
var
    Taux: real;
    DeltaT: real;
    i: integer;
    T0: real;
    F0, F00: real;
    Tseco: real;
    Xv0, Xv00: real;
    Xv1, Xv2: real;
    Xvstar1, Xvstar2: real;
    A1, A2: real;
begin
    Virial(T);
    Dervir(T);
    Mistura(1);
    Volume (T,P);
    Entalpia(P, T, 1, Vmol);
    Hseco:=Hmol/28.9645;
//Newton-Raphson para calcular TBU do ar seco

```

```

T00:=T;
if T>273.15 then
    T0:=T-5
else
    T0:=T-1;
F00:=FZero(P,T00,Hseco);
F0:=FZero(P,T0,Hseco);
Tseco:=0;
while (Abs(T00-T0))>0.0001 do begin
    Tseco:=T0 - F0*(T0-T00)/(F0-F00);
    F00:=F0;
    F0:=FZero(P,Tseco,Hseco);
    T00:=T0;
    T0:=Tseco;
end;
if (Tstar<Tseco) then begin
    MessageDlg('Parâmetros P e T Incompatíveis !',mtError,[mbOK],1);
    teste:=false;
end
else begin
//A função FSAD é a usada em conjunção com o método de N-R.
//Tudo fica dependente da fração de vapor Xv.
//Dada uma P, TBS e TBU, procura-se a Xv, via equação do saturador
//adiabático, que satisfaz aquela equação.

//Temperatura TSTAR (TBU termodinmica)

    DeltaT:=1E-4*T;
    Taux:=Tstar;
//3 Xv para fazer extrapolação junto com a função ATETA para o caso saturado
    i:=1;
    while i<=3 do begin
        if i<>3 then
            Tstar:=T - i*DeltaT;
        if i=3 then
            Tstar:=Taux;
        Virial(Tstar);
        Dervir(Tstar);
        Pwstar:=Pwsat(Tstar);

```

```

Enhance(P,Pwstar,Tstar,Xastar);
Xvstar:=(1-Xastar);
Fstar:=FS;
Wstar:=0.62198*FStar*PWstar/(P-Fstar*PWstar);
Mistura(Xastar);
Volume(P,Tstar);
Entalpia(P,Tstar,Xastar,Vmol);
Hstar:=Hmol/(Xastar*28.9645);
HLstar:=Hagua(Tstar-273.15);
//Atualiza os coeficientes viriais
Virial(T);
Dervir(T);
Pws:=Pwsat(T);
Enhance(P,Pws,T,Xas);
Xvs:=(1-Xas);
Ws:=0.62198*FS*Pws/(P-FS*Pws);
//método de N-R disparado com a função FSAD
//chutes iniciais em Xv
Xv0:=Xvs*(Tstar-Tseco)/(T-Tseco);
F0:=FSAD(Xv0,P,T,Xvs,Ws,FS,Pws,Wstar);
Xv00:=0.9*Xv0;
Xv:=Xv0;
F00:=FSAD(Xv00,P,T,Xvs,Ws,FS,Pws,Wstar);
while Abs(Xv00-Xv0)>1.E-6 do begin
    Xv:=Xv0-F0*(Xv0-Xv00)/(F0-F00);
    F00:=F0;
    F0:=FSAD(Xv,P,T,Xvs,Ws,FS,Pws,Wstar);
    XV00:=XV0;
    Xv0:=XV;
end;
if i=1 then begin
    Xv1:=XV;
    Xvstar1:=XVstar;
end;
if i=2 then begin
    Xv2:=XV;
    Xvstar2:=XVstar;
end;
i:=i+1;

```

```

end;
phi:=Xv/Xvs;
mi:=phi*(P-FS*Pws)/(P-phi*FS*Pws);
W:=mi*Ws;
Xa:=1-Xv;
Mistura(Xa);
Volume(T,P);
Entalpia(P,T,Xa,Vmol);
V:=Vmol/(Xa*28.9645);
H:=Hmol/(Xa*28.9645);
Tstar:=Tstar-273.15;
Pv:=P*Xv/FS;
if Pv>0 then begin
    Torv:=T;
    Torv:=Tsat(Torv,Pv);
end
else
    Torv:=-200;

//calcula a constante do psicrometro
if (Tstar+273.15)<T-1 then
    ATETA:=(Xvstar-Xv)/(T-Tstar-273.15)
else begin
    A1:=(Xvstar1-Xv1)/DeltaT;
    A2:=(Xvstar2-Xv2)/(2*DeltaT);
    ATETA:=A1+(A2-A1)/2;
end;
end;
end;

//*****//

procedure Calcula_para_TBSeW (P, T, Win: real);
begin
    Tstar:=T;
    repeat begin
        Tstar:=Tstar-0.005;
        Calcula_para_TBSeTBU (P, T, Tstar);
    end;
end;

```

```
    until (W-Win)< 0.00001;  
end;
```

```
//*****//
```

```
procedure Calcula_para_TBSeH (P, T, Hin: real);  
begin  
    Tstar:=T;  
    repeat begin  
        Tstar:=Tstar-0.005;  
        Calcula_para_TBSeTBU (P, T, Tstar);  
    end;  
    until (H-Hin)< 0.1;  
end;
```

```
//*****//
```

```
procedure TCalculos.TBS_TBUClick(Sender: TObject);  
begin  
    Info2.Text := '';  
    Label1.Caption := '°C';  
    Label2.Caption := '°C';  
    Label5.Caption:= 'Temperatura de Bulbo Seco';  
    Label6.Caption := 'Temperatura de Bulbo Úmido';  
end;
```

```
//*****//
```

```
procedure TCalculos.TBS_phiClick(Sender: TObject);  
begin  
    Info2.Text := '';  
    Label1.Caption := '°C';  
    Label2.Caption := '%';  
    Label5.Caption:= 'Temperatura de Bulbo Seco';  
    Label6.Caption:= 'Umidade Relativa';  
end;
```

```
//*****//
```

```
procedure TCalculos.TBS_WClick(Sender: TObject);
```

```
begin
```

```
    Info2.Text := '';
```

```
    Label1.Caption := '°C';
```

```
    Label2.Caption := 'kgvap/kgas';
```

```
    Label5.Caption:= 'Temperatura de Bulbo Seco';
```

```
    Label6.Caption:= 'Umidade Absoluta';
```

```
end;
```

```
//*****//
```

```
procedure TCalculos.TBS_hClick(Sender: TObject);
```

```
begin
```

```
    Info2.Text := '';
```

```
    Label1.Caption := '°C';
```

```
    Label2.Caption := 'kJ/kgas';
```

```
    Label5.Caption:= 'Temperatura de Bulbo Seco';
```

```
    Label6.Caption:= 'Entalpia específica';
```

```
end;
```

```
//*****//
```

```
procedure TCalculos.W_hClick(Sender: TObject);
```

```
begin
```

```
    Info1.Text := '';
```

```
    Info2.Text := '';
```

```
    Label1.Caption := 'kgvap/kgas';
```

```
    Label2.Caption := 'kJ/kgas';
```

```
    Label5.Caption:= 'Umidade Absoluta';
```

```
    Label6.Caption:= 'Entalpia específica';
```

```
end;
```

```
//*****//
```

```
procedure TCalculos.Button1Click(Sender: TObject);
```

```
begin
```

```
    P:=strtoint(Pressao.Text)*1000;
```

```
    T:=strtoint(Info1.Text)+273.15;
```

```
    if TBS_TBU.Checked then begin
```

```

Tstar:=strtofloat(Info2.Text)+273.15;
TesteConsistencia (T,P);
if Teste=true then begin
    Calcula_para_TBSeTBU (P, T, Tstar);
    str(P/1000:4:3, straux);
    Info4.Text:=straux;
    str(T-273.15:4:2, straux);
    Info5.Text:=straux;
    str(Tstar-273.15:4:2, straux);
    Info6.Text:=straux;
    str(Torv-273.15:4:2, straux);
    Info7.Text:=straux;
    str(phi*100:3:2, straux);
    Info8.Text:=straux;
    str(mi*100:3:2, straux);
    Info9.Text:=straux;
    str(W:2:6, straux);
    Info10.Text:=straux;
    str(H:5:3, straux);
    Info11.Text:=straux;
    str(V/1000:3:4, straux);
    Info12.Text:=straux;
    str(Xv:1:4, straux);
    Info13.Text:=straux;
    str(FS:2:4, straux);
    Info14.Text:=straux;
    str(ATETA:2:8, straux);
    Info15.Text:=straux;
end;
end;
if TBS_phi.Checked then begin
    phi:=strtofloat(Info2.Text)/100;
    TesteConsistencia (T,P);
    if Teste=true then begin
        Calcula_para_TBSePhi (P, T, phi);
        str(P/1000:4:3, straux);
        Info4.Text:=straux;
        str(T-273.15:4:2, straux);
        Info5.Text:=straux;

```

```

    str(Tstar-273.15:4:2, straux);
    Info6.Text:=straux;
    str(Torv-273.15:4:2, straux);
    Info7.Text:=straux;
    str(phi*100:3:2, straux);
    Info8.Text:=straux;
    str(mi*100:3:2, straux);
    Info9.Text:=straux;
    str(W:2:6, straux);
    Info10.Text:=straux;
    str(H:5:3, straux);
    Info11.Text:=straux;
    str(V/1000:3:4, straux);
    Info12.Text:=straux;
    str(Xv:1:4, straux);
    Info13.Text:=straux;
    str(FS:2:4, straux);
    Info14.Text:=straux;
    str(ATETA:2:8, straux);
    Info15.Text:=straux;
end;
end;
if TBS_W.Checked then begin
    Win:=strtofloat(Info2.text);
    TesteConsistencia (T,P);
    if Teste=true then begin
        Calcula_para_TBSeW (P, T, Win);
        str(P/1000:4:3, straux);
        Info4.Text:=straux;
        str(T-273.15:4:2, straux);
        Info5.Text:=straux;
        str(Tstar-273.15:4:2, straux);
        Info6.Text:=straux;
        str(Torv-273.15:4:2, straux);
        Info7.Text:=straux;
        str(phi*100:3:2, straux);
        Info8.Text:=straux;
        str(mi*100:3:2, straux);
        Info9.Text:=straux;

```

```

    str(W:2:6, straux);
    Info10.Text:=straux;
    str(H:5:3, straux);
    Info11.Text:=straux;
    str(V/1000:3:4, straux);
    Info12.Text:=straux;
    str(Xv:1:4, straux);
    Info13.Text:=straux;
    str(FS:2:4, straux);
    Info14.Text:=straux;
    str(ATETA:2:8, straux);
    Info15.Text:=straux;
end;
end;
if TBS_h.Checked then begin
    Hin:=strtofloat(Info2.text);
    TesteConsistencia (T,P);
    if Teste=true then begin
        Calcula_para_TBSeH(P, T, Hin);
        str(P/1000:4:3, straux);
        Info4.Text:=straux;
        str(T-273.15:4:2, straux);
        Info5.Text:=straux;
        str(Tstar-273.15:4:2, straux);
        Info6.Text:=straux;
        str(Torv-273.15:4:2, straux);
        Info7.Text:=straux;
        str(phi*100:3:2, straux);
        Info8.Text:=straux;
        str(mi*100:3:2, straux);
        Info9.Text:=straux;
        str(W:2:6, straux);
        Info10.Text:=straux;
        str(H:5:3, straux);
        Info11.Text:=straux;
        str(V/1000:3:4, straux);
        Info12.Text:=straux;
        str(Xv:1:4, straux);
        Info13.Text:=straux;
    end;
end;

```

```

        str(FS:2:4, straux);
        Info14.Text:=straux;
        str(ATETA:2:8, straux);
        Info15.Text:=straux;
    end;
end;
end;

```

```

//*****//

```

```

procedure TCalculos.PressaoKeyPress(Sender: TObject; var Key: Char);
begin
    if not (key in ['0'..'9',#13,#8,',','.','-']) then
        key:=#0;
    if (key='.') or (key=',') then
        key:=DecimalSeparator;
    if key=#13 then
        Info1.setfocus;
    end;
end;

```

```

//*****//

```

```

procedure TCalculos.Info1KeyPress(Sender: TObject; var Key: Char);
begin
    if not (key in ['0'..'9',#13,#8,',','.','-']) then
        key:=#0;
    if (key='.') or (key=',') then
        key:=DecimalSeparator;
    if key=#13 then
        Info2.setfocus;
    end;
end;

```

```

//*****//

```

```

procedure TCalculos.Info2KeyPress(Sender: TObject; var Key: Char);
begin
    if not (key in ['0'..'9',#13,#8,',','.','-']) then
        key:=#0;
    if (key='.') or (key=',') then

```

```
key:=DecimalSeparator;  
if key=#13 then  
    Button1.setfocus;  
end;
```

```
//*****//
```

```
end.
```