空気状態計算ツール説明書

Shigeru Suzuki

Apr. 1, 2024 (original release Oct. 30, 2021)

Version02 変更点

このリリースについて、下記の変更、追加を実施しました。

1. 相対湿度 100%のときに計算結果が出力されなくなるバグを修正した。
2. 計算結果は、計算シートの下側部分 A15:K15 に自動追記されるよう変更した。
3. 計算結果記録部分の小数点以下桁数を２桁とするようフォーマットを変更した(表示のみ)。
4. 入力変数間の矛盾を検証し、プログラムを停止するロジックを追加した。

   (追記した判定条件)

   (1) 管球温度≥質球温度≥露点温度、の条件を満さない場合は入力値を自動修正して表示する。

   (2) 0%RH≤相対湿度≤100%RH、の条件を追加し、相対湿度<0%RH の場合は 0%、相対湿度>100%RH の場合は 100%として計算する。

   (3) -10°C≥管球温度≥50°C、の条件を追加し、この範囲の外側は計算せずにプログラムを打ち切り表示する。
5. V2 から V2.1 への変更点
   湿球温度指定の場合に、湿球温度が 0.00 と表示されるバグを修正。

[1]はじめに

このプログラムは、空気の乾球温度と、その他の条件として、湿球温度、相対湿度、露点、絶対湿度をそ指定して、空気の状態を計算します。
計算式は、OpenOffice BASIC で記述されており、ソースコードを参照することができます。
この文書は、ソースコードを読むための情報をまとめたものです。

OpenOffice BASIC は、LibreOffice Calc のシートの情報を読み、その情報に基いて計算するプロシージャを実行させることを可能とするプログラム言語です。
各プロシージャ(Sub)や函数(Function)の計算式は、一般的に用いられているもので、式のリファレンスは、この文書に記載しています。

計算式や記述の誤りに気付いた場合には、一報頂けましたら修正して公開します。また、このプログラムは、General Public License (GPL)に基いて配布するため、誰でも自由にこのプログラムを使用することができますが、商用利用することはできず、また、ユーザーが自分で修正、改造して使用する場合でも、元の作成者を含む第三者から開示請求があった場合には、ソースコードを公開する義務があります。
また、GPL 以外に著作権法による制限もあります（計算式にはプログラム作者の著作権はありませんが、プログラムの構成やプログラムそのものには、プログラム作者の著作権があります）。
また、いかなるバグがあろうとも、プログラムの著作者は、その使用による結果の責任を負いませんので、ユーザーは自己責任で使用しなければなりません。
このプログラムを使用したことは、計算結果に誤りのないことの裏付けにはなりませんので、重要な計算に使用する場合には、他のサイトのツール等で、結果を確認してください。
インターネットの計算サイトとの結果の比較は、いくつかの場合について実施していますが、小数点以下の差異が認められています。これは、内部のアルゴリズムの違いによるものと推定され、インターネットサイトのツールにおいても、同じ数字で、選択する変数を変更して検証すると、小数点以下が同じにならないことがあります。
元の計算式そのものが、近似式を含んでおり、また、その中で使用されている数式の定数の有効桁数の設定に無理があるので、やむを得ない誤差と考えることもできますが、小数点より上の桁で誤差が生じた場合には、プログラムのバグが疑われます。

[2]使用方法

プログラムは、１枚のスプレッドシートに BASIC プログラムを埋め込んだファイルで構成されており、シートにはプログラム実行用のボタンが付いています。
計算したい条件は、セルの C4:C8 に入力します。うちの C4 セルは、乾球温度の指定値であり必須入力です。
C5:C8 のセルは、それぞれ湿球温度、相対湿度、露点、絶対湿度を入力する欄であり、このうち一つのみを入力し、その他は、必ず空欄としてください。
入力が終了したら、"CALCULATE"ボタンを押すと、A15:I15 セルに、入力した条件と、計算結果を書き込みます。計算結果は小数点２桁まで表示されます。
表示桁数を変更したい場合には、プログラム内の Spostare モジュールの該当部分を書き換えるか、計算後に表示桁数を変更してください。
前回までに計算した結果は、１行下にシフトしてそのまま残ります。
J15:K15 セルには任意のコメントを書くことができます。
J16:K16 以下に後からコメントを書くこともできます。
"CLEAR"ボタンをクリックすると、計算結果とコメントが全て消去されます。

[3]プログラムの解説

プログラムは、プロシージャ（Sub)と函数（Function）で構成され、入出力は、Calc のスプレッドシート上で行います。
函数とプロシージャの一覧を、表 1 に示します。
使用されている主要な変数名の一覧を表２に示します。
各プロシージャの解説を表３に示します。

表１　プロシージャと函数

Procedure or Function	機能	出力内容（()内は、変数名）
Main	入力変数の選択とエラー処理 計算結果出力	絶対湿度(AHI で処理する場合を除く) 空気の比エンタルピ, 不快指数
RHI	相対湿度指定時の計算	飽和水蒸気圧(ps), 水蒸気分圧(pw) 露点(dpe), 湿球温度(tw)
DWI	湿球温度指定時の計算	飽和水蒸気圧(ps), 水蒸気分圧(pw) 相対湿度(rh), 露点(dpe)
DPI	露点指定時の計算	飽和水蒸気圧(ps), 水蒸気分圧(pw) 相対湿度(rh), 湿球温度(tw)
AHI	絶対湿度指定時の計算	飽和水蒸気圧(ps), 水蒸気分圧(pw) 相対湿度(rh), 湿球温度(tw)
WH (Function)	飽和水蒸気圧計算 Wexler-Hyland 式の蒸気圧公式	飽和水蒸気圧函数(ps)
WHD (Function)	飽和水蒸気圧の差分値計算	飽和水蒸気の温度による差分値（露点計算に使用）
TWEST	湿球温度計算（誤差の最小値を順次計算し、誤差が最小から増えたときに、その一つ前の値を湿球温度とする）	湿球温度(tw)
DPEST	露点計算（ニュートン法）	露点(dpe)
Param	パラメータ値読込	Wexler-Hyland 式の定数 {wha0(0) - wha0(6), whb0(0) - whb0(6)}

		乾湿係数(cdw)、大気圧(p0) 水の分子量(mw)、空気の分子量相当値(ma)
Cancellare	計算結果記録部分の消去	入力値は消去しない
Spostare	セルを挿入して計算結果を挿入し、数値のセルを小数点以下 2 桁に揃える	A15:K15 に空白のセルを挿入する 前回までの計算結果と注釈は下にシフトして残る

表２　主要変数一覧

tc	気温（乾球温度）[°C]	h	比エンタルピ@tk[K](tc[°C]) [kJ/kg]
td	乾球温度[°C]	mw	水の分子量=18.0153[g/mol]
tk	乾球温度[K]	ma	空気の分子量相当値=28.9645[g/mol]
tw	湿球温度[°C]	p0	大気圧 = 101.3 [kPa]
rh	相対湿度 [-] （プログラム中では%値ではない）	cdw	乾湿係数(tc<0.01 の場合 0.000583、それ以外は 0.00062)
ah	絶対湿度[g/kg]	dp	ユーザ指定の露点値[°C]
ps	飽和水蒸気圧@tk [kPa]	dpe	露点の推定値[°C]
psl_wk	ln(ps)	twe	湿球温度推定値[°C]
ps	飽和水蒸気圧@tk[K](tc[°C])	ps_error	湿球温度計算用誤差値
pw	水蒸気圧@tk[K](tc[°C]) [kPa]		

各プロシージャの解説

表３−１　メインプログラム

プログラム	説明
Dim dbs, wbs, rhs, dps, ahs As String	入力の各項目を文字として宣言
errore1="" errore2=""	これらは、メッセージを保存する変数で初期値は空白
cbook=ThisComponent csheet=cbook.CurrentController.ActiveSheet	Object 変数の定義
dbs=csheet.getCellByPosition(2,3).string wbs=csheet.getCellByPosition(2,4).string rhs=csheet.getCellByPosition(2,5).string dps=csheet.getCellByPosition(2,6).string ahs=csheet.getCellByPosition(2,7).string	入力変数選択処理のための文字の読込 入力値を文字として読込んで、記入の有無を判別するための変数を定義した。
td=csheet.getCellByPosition(2,3).value	乾球温度を読込む

If td<-10 or td>50 Then
　errore1="Range error."
　foglio_a.getCellByPosition(9,14).string=errore1
　Goto Z9999
EndIf
	乾球温度∈[-10,50]のレンジ外の場合のエラー処理

Call Param	パラメータの値を読み込む
If dbs="" Then	下記の場合には、メッセージを出し

MsgBox("DBT shall not be blank.",1)
Goto Z9999
End If
If wbs="" AND rhs="" AND dps="" AND ahs="" Then
MsgBox("Input one of WBT, RH, DewPoint or Absolute Humidity.",1)
Goto Z9999
End If
	て、処理を終了する。
	
	(1)乾球温度が入力されていない
	(2)条件不足の場合（湿球温度、相対湿度、露点温度、絶対湿度、のいずれも空白の場合）

If wbs="" AND rhs<>"" Then
Call RHI
Goto Z9980
End If
	Case#1 相対湿度指定の場合
	プロシージャ RHI を実行し、絶対湿度計算ステップに戻る。

If wbs<>"" Then
Call DWI
Goto Z9980
End If
	Case#2 湿球温度指定の場合
	プロシージャ DWI を実行し、絶対湿度計算ステップに戻る。

If wbs="" AND rhs="" AND dps<>"" Then
CALL DPI
GOTO Z9980
End If
	Case#3 露点指定の場合
	プロシージャ DPI を実行し、絶対湿度計算ステップに戻る。

If wbs="" AND rhs="" AND dps="" AND ahs<>"" Then
CALL AHI
GOTO Z9970
End If
	Case#4 絶対湿度の場合
	プロシージャ AHI を実行し、絶対湿度計算ステップをスキップして、空気の比エンタルピ計算式に戻る。

Z9980: ah=mw/ma*pw/(p0-pw)	絶対湿度[g/kg]計算ステップ
Z9970: h=1.006*td+(1.86*td+2501)*ah di=0.81*td+0.01*rh*100*(0.99*td-14.3)+46.3	空気の比エンタルピ計算式 不快指数計算式

foglio_a.getCellByPosition(0,14).value=td
foglio_a.getCellByPosition(1,14).value=twe
foglio_a.getCellByPosition(2,14).value=rh*100
foglio_a.getCellByPosition(3,14).value=ah*1000
foglio_a.getCellByPosition(4,14).value=tc
foglio_a.getCellByPosition(5,14).value=h
foglio_a.getCellByPosition(6,14).value=di
foglio_a.getCellByPosition(7,14).value=ps
foglio_a.getCellByPosition(8,14).value=pw
foglio_a.getCellByPosition(9,14).string=errore1
foglio_a.getCellByPosition(10,14).string=errore2
If wbs="" then
foglio_a.getCellByPosition(1,14).value=twe
Else
Goto Z9999
End If
	計算結果出力
	A15:K15 に計算結果等を出力する。
	errore1, errore2: 計算判定条件のエラーがった場合に J15:K16 にエラーメッセージを出力する。
	J15:K15 には、計算後にユーザーが自由にコメントを書き込んでも良い。
	
	If ブロック：
	湿球温度が空白の場合、湿球温度は計算値を書き込む

Z9999:	エラー処理の終了位置

表３−２　RHI：相対湿度を指定した場合の計算プロシージャ

Sub RHI'Case#1 相対湿度指定	
rh=foglio_a.getCellByPosition(2,5).value/100	相対湿度を読み込む

If rh>1.0 Then
rh=1.0
errore2="Humidity set to 100%"
ElseIf rh<0.0 Then
rh=0.0
errore2="Humidity set to 0%"
End If
	相対湿度が[0%,100%]の範囲に入らない場合は、下限または上限として計算する。
	Humidity modified：湿度修正

If rh=100.0 Then
dpe=td
tw=td
Goto Z0100
End If
	相対湿度 100%の場合、湿球温度、露点ともに乾球温度に一致するので、計算処理をスキップする。

If rh=1.0 Then
tc=td
twe=td
pw=WH(td)
ps=pw
Goto Z0100
End If
	相対湿度が 100%のときは、湿球温度、露点ともに乾球温度に一致する

ps=WH(td) pw=ps*rh	ps　:飽和水蒸気圧 pw　:指定の相対湿度での水蒸気圧

If rh=0.0 Then
tc=-273.15
Goto Z0110
EndIf
	相対湿度が 0%のときは、露点計算モジュールをスキップして露点を絶対 0 度(-273.15°C)と表示する

Call DPEST Z0110: Call TWEST Z0100:	DPEST　:露点計算モジュール TWEST　:湿球温度計算モジュール

表３−３　DWI　'Case#2 湿球温度指定

tw=csheet.getCellByPosition(2,4).value
If tw>td Then
twe=td
tw=td
errore2="WBT is set to DBT"
End If
ps=WH(td)
pw=WH(tw)-cdw*p0*(td-tw)
rh=pw/ps
csheet.getCellByPosition(2,16).value=tw
twe=tw
Call DPEST
	湿球温度を読込む
	湿球温度が乾球温度よりも高い場合には、
	湿球温度＝乾球温度
	として処理する。

表３−４ DPI 'Case#3 露点指定

dp=csheet.getCellByPosition(2,6).value
 	
If dp>td Then
　dp=td
　errore2="dewpoint is set to DBT"
End If
 	
rh=WH(dp)/WH(td)
dpe=dp
ps=WH(td)
pw=WH(dp)
	露点値を読込む
	
	露点が乾球温度よりも高い場合は、指定値を乾球温度と同じ(相対湿度100%)とし、『露点を乾球温度に設定した』というメッセージを表示する。

Call TWEST	

表３−５ AHI 'Case#4 絶対湿度指定

ah=csheet.getCellByPosition(2,7).value/1000	絶対湿度を読込む
ps=WH(td) pw=ma*p0*ah/(mw+ma*ah) rh=pw/ps	ps　: 飽和水蒸気圧 pw　:指定温度 DBT での水蒸気圧 rh　:相対湿度

If rh>1.0 Then
　rh=1.0
　ah=mw/ma*WH(td)/(p0-WH(td))
　errore2="input value modified"
EndIf
	RH>100%となる場合の処理

Call DPEST	露点温度計算モジュール
Call TWEST	湿球温度計算モジュール

表３−６ DPEST '露点計算

If rh=0 Then
　tc=-273.15
　GOTO Z0020
End If
pwdp=WH(td)*rh
spwmpw=WH(td)-pwdp
	相対湿度=0 のときは、露点=絶対 0 度 (−273.15°C)とする
	spwmpw: 飽和水蒸気圧と実水蒸気圧との差

tc=td	計算する温度を乾球温度とする
i=0 Z0010: 　i=i+1 　dpe=tc-(WH(tc)-pwdp)/WHD(tc)	i: 反復計算のためのカウンタ Z0010: 反復計算の起点 dpe: ニュートン法（後述）による露点推定値
ps_error=WH(dpe)-pwdp	露点推定値での蒸気圧の誤差

If i>100 OR ps_error<1.0E-6 Then
　goto Z0020
Else
　tc=dpe
　goto Z0010
End If
Z0020:
	反復の終了条件
	(1)反復 102 回
	(2)打切り誤差(1/1000000)以下に収束
	ニュートン法は、通常数回で収束するため 102 回は、プログラムの誤りによるものと考える

表３−７　TWEST　'湿球温度計算

Dim delta_t As Double	温度上昇の刻み幅
twe=dpe ps_error_p=(rh*WH(td)-WH(twe)+cdw*p0*(td-twe))^2	湿球温度は露点以上乾球温度以下なので、露点から始めて徐々に温度を上昇させ、誤差函数を計算する ps_error_p: 誤差函数の前回値
i=0 delta_t=0.02 Z0030:	順次計算のカウンタ初期値 刻み幅は、最終計算値を小数点以下 1 桁として、0.04 以下である 0.02 に設定

i=i+1
twe=twe+delta_t
ps_error=(rh*WH(td)-WH(twe)+cdw*p0*(td-twe))^2
 	
If ps_error>ps_error_p Then
foglio_a.getCellByPosition(2,16).value=twe-delta_t
Goto Z0040
End If
 	
If twe>td Then
Goto Z0040
End If
 	
ps_error_p = ps_error
 	
Goto Z0030
 	
Z0040:
	湿球温度推定値に刻み幅を加算する。新しい推定値での誤差函数値を計算し、新しい誤差函数値とする。
	
	誤差函数値が誤差函数の前回値を下回れば、湿球温度が、推定値より高いことになるが、前回値を上回れば直前の推定値を実際値とする。
	
	推定値が乾球温度を超えたら計算異常で打切る。
	前回の誤差値に今回の誤差値を代入して最初に戻る。

その他の函数とプロシージャについては、説明する内容がないので、ソースコードとコメントを参照ください。

[4] 計算式とアルゴリズム

凡例

	単位		プログラム中の記号
p_s	kPa	T_d での飽和蒸気圧	ps
p_w	kPa	T_d での蒸気分圧	pw
r_H	無単位	相対湿度(結果は%表示)	rh
T_d	°C	乾球温度	td
T_w	°C	湿球温度	tw
T_{dp}	°C	露点温度	dpe, dp
x	g/kg(DA)	絶対湿度	ah
h	kJ/kg(DA)	空気の比エンタルピ	h
T	K	乾球温度	tk
I_d	無単位	不快指数	di

飽和蒸気圧

Wexler-Hyland(ウェクスラーハイランド)の式を使用した。式は空気調和衛生工学便覧参照。

Ps=飽和水蒸気圧(Pa)、T = 273.15 + 乾球温度(℃)[K]

$$\ln(P_S)=a_{-1}T^{-1}+a_0+a_1T+a_2t^2+a_3T^3+a_4T^4+a_L\ln(T)$$

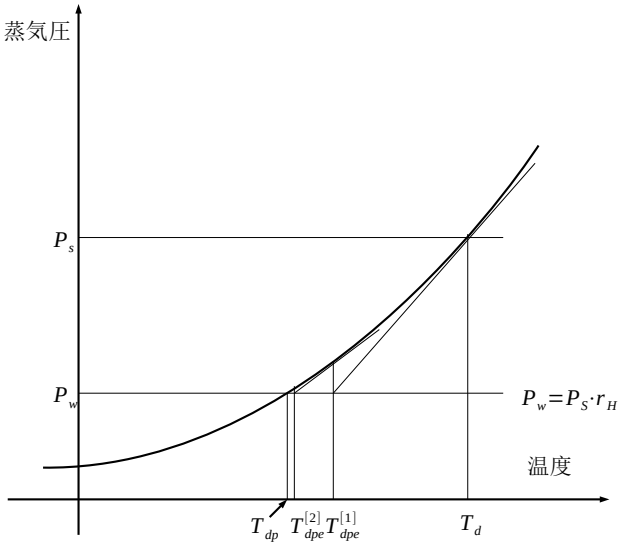
$$P_S=\exp[a_{-1}T^{-1}+a_0+a_1T+a_2t^2+a_3T^3+a_4T^4+a_L\ln(T)]$$

係数表

係数	値	
	T_d<0.01	T_d≥0.01
a_{-1}	-0.56745359E4	-0.58002206E4
a_0	6.3925247	1.3914993
a_1	-0.96778430E-2	-0.48640239E-1
a_2	0.62215701E-6	0.41764768E-4
a_3	0.20747825E-8	-0.14452093E-7
a_4	-0.94840240E-12	0
a_L	4.1635019	6.5459673

露点

露点は、ニュートン法のアルゴリズムを使用して計算する。
乾球温度を第１推定値として誤差を計算し、誤差が、打切り誤差を下回るまで、推定を繰り返す。
詳細は下記の通りとなる。



ニュートン法

飽和水蒸気圧曲線において、温度 T_d における接線が、 P_w と交わる温度を露点の第１推定値 $T_{dpe}^{[1]}$ とし、その点での飽和水蒸気圧の接線が P_w と交わる点を露点の第２推定値 $T_{dpe}^{[2]}$ とする。
同様に第３、第４と推定値を更新し、露点での飽和水蒸気圧の誤差が指定値を下回ったときの温度が露点となる。
飽和水蒸気圧と飽和水蒸気圧の微分値（差分値）は、Function としてプログラムに組込んだ。

Function WH　飽和水蒸気圧計算
Function WHD　飽和水蒸気圧の差分値

露点を指定して空気の状態を計算する場合には、次式による。

$$p_s(T_{dp})=r_H p_s(T_d)$$

$$r_H=\frac{p_s(T_{dp})}{p_s(T_d)}$$

湿球温度

Sprung の式を使用より

P_w：水蒸気分圧(kPa)、P_s：湿球温度における飽和水蒸気圧(kPa)、C_{dw}：乾湿計数、p_0：気圧(kPa)、T_d：乾球温度(℃)、T_w：湿球温度(℃)

※C_{dw}：乾湿計数は Tw が 0.01 以上の時に 0.000662、0.01 未満の時に 0.000583。

$$P_w = P_s(T_w) - C_{dw} \cdot p_0 (T_d - T_w)$$

$$P_w = r_H P_s (T_d)$$

Sprung の式を変形し、誤差函数を下記のように定義する

$$E_p = \left[P_w - P_s(T_w) + C_{dw} p_0 (T_d - T_w)\right]^2$$

湿球温度は、露点と乾球温度の間にある（ $T_{dp} \leq T_w \leq T_d$ ）。このため、湿球温度の第１推定値を露点として誤差函数を計算し、推定値に＋ δt ℃を加算して、再度誤差函数を計算する。この処理を繰り返すと誤差が漸減し、湿球温度推定値＝湿球温度付近で、誤差函数の値がが最小となる。したがって、前推定値での誤差函数値と、現推定値での誤差管数値を比較し、誤差函数値が増加に転じたところで処理を打ち切り、前推定値を真の湿球温度値 T_w とする。

ここで、例えば、湿球温度を小数点第１桁まで表示したい場合には、δt<0.04 とする。プログラム中では、 δt=0.02 としてあり、これで十分だが、桁数を更に大きく表示したい場合には、 プログラム中の Sub TWEST の変数 delta_t の値を変更して δt を変える。

相対湿度 100%の場合には、 $T_{dp} = T_w = T_d$ となるので、このプロシージャの最初に条件処理を行う。

空気の状態の計算式には、理論式と近似式（実験式や理論式の近似式）が混在しているので、むやみに有効桁数を増やすことに意味はない。

絶対湿度

空気調和衛生工学便覧から次式により絶対湿度を算出する。

$$x = \frac{m_w}{m_a} \frac{p_w}{p_0 - p_w}$$

但し、

x　絶対湿度[kg/kg]

mw　水の分子量(=18.0153[g/mol])

ma　空気の分子量(=28.9645[g/mol])

p0　空気の全圧[kPa]

pw　水蒸気分圧[kPa]

絶対湿度を指定して空気の状態を計算する場合には、この式を変形して相対湿度を求める。

$$p_w = \frac{m_a p_0 x}{m_w + m_a x}$$

$p_w = r_H p_s$ により

$$r_H = \frac{p_0}{p_s} \frac{m_a x}{m_w + m_a x}$$

不快指数

J.F.Bosen の式により計算する。

T_d：乾球温度(℃)、r_H：相対湿度(%)

$$I_d = 0.81 T_d + 0.01 r_H (0.99 T_d - 14.3) + 46.3$$

比エンタルピ

空気調和衛生工学便覧から次式により絶対湿度を算出する。

h：比エンタルピ(kJ/kg(DA))、Td：乾球温度(℃)、x：絶対湿度(kg/kg')

$$h = 1.006 T_d + (1.86 T_d + 2501) x$$

以上