

Some Lessons learned from the AWEX-G field campaign regarding calculations relating to water vapor calculations and analysis

Considerable time was spent during the analysis of the AWEX data to insure that all parties were doing calculations the same way and, hopefully, that the most accurate formulas were being used. This document is intended to convey some of the lessons learned.

Lesson 1: Use the same formula for RH

Relative Humidity has been defined in different ways. Bohren and Albrecht (Atmospheric Thermodynamics, 1998, p. 186) describe some of the confusion regarding the definition. The traditional definition is that RH is the ratio of vapor pressure, e , to saturation vapor pressure, e_s

$$\text{RH} = \frac{e}{e_s} \quad (1)$$

Bohren and Albrecht report, however, that at some point the WMO defined RH as a ratio of the mixing ratio, w , to the saturation mixing ratio, w_s

$$\text{RH} = \frac{w}{w_s} \quad (2)$$

These are not equivalent formulations. In fact,

$$\frac{w}{w_s} = \frac{e}{e_s} \left(\frac{p-e}{p-e_s} \right) \quad (3)$$

We agreed to use the traditional formulation of RH shown in equation (1), which involves the ratio of vapor pressures.

Lesson 2: Use the same vapor pressure formulation

There is considerable disagreement on what the most accurate formulation is for vapor pressure of liquid at temperatures below freezing. It is hard to measure and, as has been pointed out in the literature (Murphy and Koop, 2005), the concept is unphysical below about -40C. Because of these problems, Miloshevich et. al. have suggested that calculations of RH be done with respect to ice and not water since the ice formulations are all in close agreement. But the standard is to report RH with respect to water. So at least we should agree on what formulation to use.

Vaisala uses the Wexler (1976) formulation of vapor pressure over water. (Holger Vomel has an excellent summary of many formulations at <http://cires.colorado.edu/~voemel/vp.html>. The equations for some of those formulations are shown below in *Mathematica* code.) So in AWEX we agreed to use the Wexler formulation over water for the vapor pressure calculations. It should be pointed out that, for cold temperatures, there is a considerable difference between this formulation and the more recent reference of Murphy and Koop. We will be studying the influence of this different formulation of vapor pressure on the WAVES comparisons.

```
In[612]:=
SatVaporPressLiq[TempC_, Choice_] := Module[{TempK},
  TempK = TempC + 273.15;
  Switch[Choice,
    "GoffGratch", 10 ^ (-7.90298 (373.16 / TempK - 1) +
      5.02808 Log[10, 373.16 / TempK] - 1.3816 10^-7 (10^11.344 (1 - TempK/373.16) - 1) +
      8.1328 10^-3 (10^-3.49149 (373.16/TempK - 1) - 1) + Log[10, 1013.246]),
    "HylandWexler", Exp[(-0.58002206 10^4 / TempK + 0.13914993 10^1 - 0.48640239 10^-1 TempK +
      0.41764768 10^-4 TempK^2 - 0.14452093 10^-7 TempK^3 + 0.65459673 10^1 Log[TempK])] 10^-2,
    "Wexler", Exp[-2.9912729 10^3 TempK^-2 - 6.0170128 10^3 TempK^-1 + 1.887643854 10^1 -
      2.8354721 10^-2 TempK + 1.7838301 10^-5 TempK^2 - 8.4150417 10^-10 TempK^3 +
      4.4412543 10^-13 TempK^4 + 2.858487 Log[TempK]] 10. / 1000.,
    "MagnusTeten", 10 ^ ((7.5 TempC) / (TempC + 237.3) + 0.7858),
    "Buck81", 6.1121 Exp[17.502 TempC / (240.97 + TempC)],
    "Buck96", 6.1121 Exp[(18.678 - TempC / 234.5) TempC / (257.14 + TempC)],
    "WMO", 10 ^ (10.79574 (1 - 273.15 / TempK)
      - 5.02800 Log[10, TempK / 273.15] + 1.50475 10^-4 (1 - 10^-8.2969 (TempK/273.15-1)) +
      0.42873 10^-3 (10^-4.769955 (1-273.15/TempK) - 1) + 0.78614),
    "MurphyKoop", Exp[(54.842763 - 6763.22 / TempK - 4.21 Log[TempK] +
      0.000367 TempK + Tanh[0.0415 (TempK - 218.8)]
      (53.878 - 1331.22 / TempK - 9.44523 Log[TempK] + 0.014025 TempK))] 10^-2
  ]
]
```

Lesson 3: Convert from RH to Mixing Ratio the same way

There are some simpler formulas for this conversion that neglect the influence of the relative humidity in the calculation of saturation mixing ratio. This introduces significant error under warm, moist conditions as we likely will have in the boundary layer during WAVES. The full conversion from T, P, RH to mixing ratio is given in equation (4) and shown below in *Mathematica* code.

$$\text{MR (g/kg)} = 10 \text{ RH } 0.62197 \frac{e_s}{P - \frac{\text{RH}}{100} e_s} \quad (P, e_s \text{ in hPa (i.e. mb)}) \quad (4)$$

In[613] :=

```
GetMixRatioWater[{TempC_, PressMB_, RH_}] :=
  10 RH SaturationMixingRatioWater2[{TempC, PressMB, RH}] (* output in g/kg *)

SaturationMixingRatioWater2[{TempC_, PressMB_, RHLiqPercent_}] :=
  Module[{SatVapPress},
    SatVapPress = SatVaporPressLiq[TempC, "Wexler"];
    0.62197 SatVapPress / (PressMB - (RHLiqPercent / 100) SatVapPress)
    (* output in units of kg/kg *)
  ]
```

Lesson 4: When comparing radiosondes launched on the same balloon, use time as the independent variable

Pressure sensors can disagree, but the clocks in sondes are highly precise. If the sondes are on the same balloon, time after launch is the reliable independent variable.

Lesson 5: When comparing radiosondes launched on different balloons, use pressure/altitude as the independent variable

When comparing sensors launched on different balloons, there is no guarantee that the ascent rate of the two balloons is the same so pressure/altitude is the independent variable to use.

Plot up the some of the vapor pressure (over water) formulations with respect to Goff-Gratch

Create Temperature table

In[615] :=

```
TempTable = Table[-i, {i, 1, 100}];
```

Get the vapor pressure over liquid.

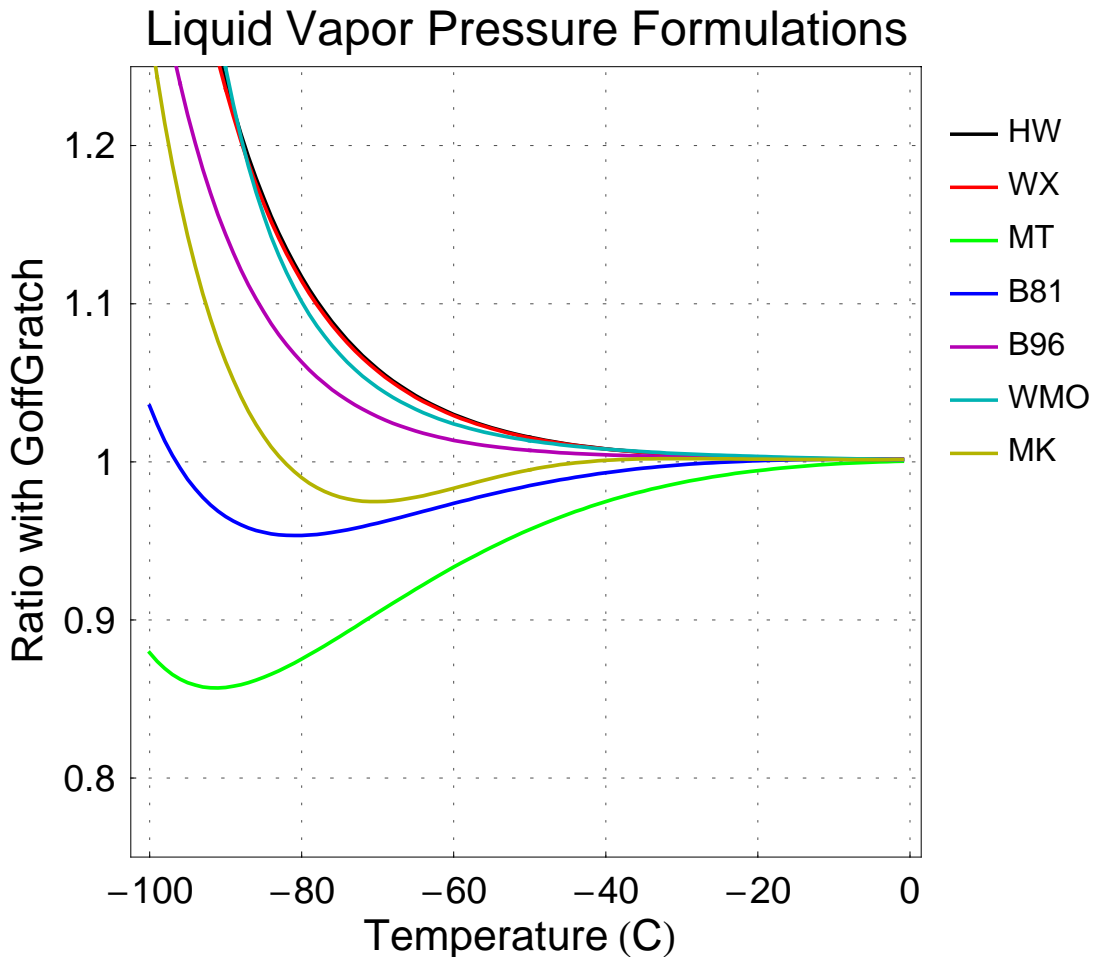
```
In[616]:=
GG = SatVaporPressLiq[TempTable, "GoffGratch"];
HW = SatVaporPressLiq[TempTable, "HylandWexler"];
WX = SatVaporPressLiq[TempTable, "Wexler"];
MT = SatVaporPressLiq[TempTable, "MagnusTeten"];
B81 = SatVaporPressLiq[TempTable, "Buck81"];
B96 = SatVaporPressLiq[TempTable, "Buck96"];
WMO = SatVaporPressLiq[TempTable, "WMO"];
MK = SatVaporPressLiq[TempTable, "MurphyKoop"];
```

Collect the data for plotting

```
In[624]:=
HWPlot = {HW / GG, HW / GG, TempTable};
WXPlot = {WX / GG, WX / GG, TempTable};
MTPlot = {MT / GG, MT / GG, TempTable};
B81Plot = {B81 / GG, B81 / GG, TempTable};
B96Plot = {B96 / GG, B96 / GG, TempTable};
WMOPlot = {WMO / GG, WMO / GG, TempTable};
MKPlot = {MK / GG, MK / GG, TempTable};
```

In[630]:=

```
MyPlotNew3[ {HWPlot, WXPlot, MTPlot, B81Plot, B96Plot, WMOPlot, MKPlot},
  ErrorBarInterval = {5, 5, 5, 5, 5, 5, 5, 5, 5},
  DropBefore = {-100, -100, -100, -100, -100, -100, -100, -100, -100},
  DropAfter = {0, 0, 0, 0, 0, 0, 0, 0, 0}, BarsYN = {0, 0, 0, 0, 0, 0, 0, 0, 0}, RangeXY = 0,
  DLog = {0, 0}, PLog = {1, 1, 0, 0}, SmoothPoints = {0, 0, 0, 0, 0, 0, 0, 0, 0},
  LegendText = {"HW", "WX", "MT", "B81", "B96", "WMO", "MK"}, PRange = {All, {0.75, 1.25}},
  XLabel = "Temperature (C) ", YLabel = "Ratio with GoffGratch",
  PltTitle = "Liquid Vapor Pressure Formulations", DFont = "Helvetica",
  DSize = 14, XSize = 16, YSize = 16, PSize = 18, LSize = 12, ColorOrBW = 1,
  LineThickness = 0.005, AspectRat = 1.0, LegPos = {1, 0}, Symbols =
  {{0, 5}, {0, 5}, {0, 5}, {0, SSize}, {0, SSize}, {0, SSize}, {0, SSize}, {0, SSize}},
  WhichFunc = 2, BarThick = 0.003, Grid = 1, Disp = 1, ColorDash = 0,
  PJoined = {1, 1, 1, 1, 1, 1, 1, 1, 1}];
```



For comparison look at vp over ice with respect to Goff-Gratch. The good agreement of most of the formulations (note the different scales for the liquid and

ice plots) is the reason for the suggestion in Miloshevich et. al. (2006) to use RH with respect to ice as the standard unit of relative humidity.

```
In[631]:=
SatVaporPressIce[TempC_, Choice_] := Module[{TempK},
  TempK = TempC + 273.15;
  Switch[Choice,
    "GoffGratch", 10 ^ (-9.09718 (273.15 / TempK - 1) -
      3.56654 Log[10, 273.15 / TempK] + 0.876793 (1 - TempK / 273.15) + Log[10, 6.1071]),
    "HylandWexler", Exp[(-0.56745359 104 / TempK + 0.63925247 101 -
      0.96778430 10-2 TempK + 0.62215701 10-6 TempK2 + 0.20747825 10-8 TempK3 -
      0.94840240 10-12 TempK4 + 0.41635019 101 Log[TempK])] 10-2,
    "MagnusTeten", 10 ^ ((9.5 TempC) / (TempC + 265.5) + 0.7858),
    "Buck81", 6.1115 Exp[22.452 TempC / (272.55 + TempC)],
    "Buck96", 6.1115 Exp[(23.036 - TempC / 333.7) TempC / (279.82 + TempC)],
    "WMO", 10 ^ (-9.09685 (273.15 / TempK - 1)
      - 3.56654 Log[10, 273.15 / TempK] + 0.87682 (1 - TempK / 273.15) + 0.78614),
    "MurphyKoop",
    Exp[9.550426 - 5723.265 / TempK + 3.53068 Log[TempK] - 0.00728332 TempK] 10-2
  ]
]
```

```
In[646]:=
GG = SatVaporPressIce[TempTable, "GoffGratch"];
HW = SatVaporPressIce[TempTable, "HylandWexler"];
MT = SatVaporPressIce[TempTable, "MagnusTeten"];
B81 = SatVaporPressIce[TempTable, "Buck81"];
B96 = SatVaporPressIce[TempTable, "Buck96"];
WMO = SatVaporPressIce[TempTable, "WMO"];
MK = SatVaporPressIce[TempTable, "MurphyKoop"];
```

```
In[653]:=
HWPlot = {HW / GG, HW / GG, TempTable};
MTPlot = {MT / GG, MT / GG, TempTable};
B81Plot = {B81 / GG, B81 / GG, TempTable};
B96Plot = {B96 / GG, B96 / GG, TempTable};
WMOPlot = {WMO / GG, WMO / GG, TempTable};
MKPlot = {MK / GG, MK / GG, TempTable};
```


Here is some of Larry Miloshevich's code in IDL

Below is IDL code for calculating saturation vapor pressure using the Wexler (1976) formulation.

```
*****
function few_wexhy, T
;Saturation vapor pressure over water.Wexler (1976) formulation (used by;Vaisala). Input temperature in degrees K.
;Returns e_w in dyne/cm^2 (microbars).(1 Pa=10 dyne/cm^2)
;Range of validity:273.15<T<473.15K

td=double(T)

c0=-2.9912729e3& c1=-6.0170128e3& c2=1.887643854e1
c3=-2.8354721e-2& c4=1.7838301e-5& c5=-8.4150417e-10
c6=4.4412543e-13& D=2.858487

term=(c0*td^(-2))+c1*td^(-1))+c2*td^0)+c3*td^1)+c4*td^2)+
$(c5*td^3)+(c6*td^4)+(D*log(td))
few=exp(term) ;Pa

return,few*10.
end
*****
```

Below is IDL code for calculating mixing ratio given relative humidity

```
*****
function calc_Xv,RHin,TC,Pmb
;Calculate mixing ratio.Output in g/kg.Inputs in %,C,mb.
;Xv=rho_vapor/rho_dry=(epsilon*e)/(p-e)

RH=double(RHin)& TK=double(TC)+273.15& Pcgs=double(Pmb) *1000.
eps=0.62197 ;epsilon=rd/rv

e=(RH/100.)*few_wexhy(TK) ;vapor pressure,cgs=dynes/cm^2
Xv=(eps*e)/(Pcgs-e) ;mixing ratio (g/g)

return,float(Xv*1000.) ;g/kg
end
*****
```