ALGORITHMS USED BY ELECTRONIC LOGBOOKS FOR THE COMPUTATION OF DEW POINT TEMPERATURE

Name of e-logbook	OBSJMA for WIN
Agency	Japan Meteorological Agency
Country	Japan
Contact point (name, e-mail)	Naotaka Hiraishi, vos-office@climar.kishou.go.jp
Web site for the software	http://marine.kishou.go.jp/en/obsjma-en.html
Version number of e-logbook	1.00, 1.01, 2.00
Date of version of e-logbook	March 2002(1.00), March 2004(1.01), March 2009(2.00)
Version number of algorithm	- (All the versions of OBSJMA use same algorithm)
Date of version of algorithm	March 2002
Name of algorithm	-

Variables used as input for the computation of Dew Point Temperature		
Name	**Units**	**Precision required**
air pressure at the sea level	hPa	0.1
wet-bulb temperature	$\mathrm{deg\ C}$	0.1
dry-bulb temperature	deg C	0.1

Variables returned by the algorithm		
Name	**Units**	**Resulted precision**
Dew point temperature	deg C	0.1

Pseudocode of the algorithm[1]

Firstly, OBSJMA calculates a vapor pressure (E) by using the following expressions (1) or (2) and (3).
And then, it substitutes the calculated value to Ew in (1) or (2) and calculates a adequate value of T as the dew point.

In the case of not freezing wet-bulb:

$$\log_{10}Ew = +10.79574(1 - R) - 5.02800\log_{10}(1/R) + 1.50475*10^{-4}(1 - 10^{-8.2969(1/R - 1)}) + 0.42873*10^{-3}(10^{4.76955(1 - R)} - 1) + 0.78614 \quad (1)$$

In the case of freezing wet-bulb:

$$\log_{10}Ew = -9.09685(R - 1) - 3.56654\log_{10}(R) + 0.87682(1 - 1/R) + 0.78614 \quad (2)$$

$$E = Ew - A/755\ P(Td - T) \quad (3)$$

[1] : Possibly based on C++ alike syntax whenever possible; otherwise using original source language that was used

Ew = saturated vapor pressure (hPa)
T = wet-bulb temperature (K)
R = T_1/T (T_1 = 273.16 (K))
E = vapor pressure (hPa)
A = 0.50 (not freezing wet-bulb), 0.44 (freezing wet-bulb)
P = air pressure at the sea level (hPa)
Td = dry-bulb temperature (K)

The following is the source code:

```
 /*************************************************************************/
double vapor(double temp)
{
        double a1,a2,a3,a4,temk,t1=273.16,aa,ee;
        temk=temp+273.15;
        a1=10.79574*(1.0-t1/temk);
        a2=-5.028*log10(temk/t1);
        a3=0.000150475*(1.0-pow(10.0,-8.2969*(temk/t1-1.0)));
        a4=0.00042873*(pow(10.0,4.76955*(1.0-t1/temk))-1.0);
        aa=a1+a2+a3+a4+0.78614;
        ee=pow(10.0,aa);
        return(ee);
}
/*****************************************************************/
double Calc_Dew(double fDry,double fWet,double fPressure)
{
          int i=0;
          double hum,e,ewd,es,dtemp,wtemp,p,dew,t1,t2,t3,delta,d1,d2;
          double hum,e,ewd,es,          dew,t1,t2,t3,delta,d1,d2;
          if (fDry == 27.5 && fWet == 27.5)
          {
            return 27.5;
          }
          if (fDry == 5.0 && fWet == 5.0)
          {
            return 5.0 ;
          }

          es=vapor((double)fDry);
          ewd=vapor((double)fWet);
          e=ewd-0.5*(double)(fDry-fWet)*(double)fPressure/755.0;
          if (e<0) i=1;
          hum=e/es*100.0;
          if(i==0){
                    t1=-40.0;
                    t2=50.0;
                    delta=0.00001;
                    d1=delta*2.0;
                    while(fabs(d1)>delta){
                                        if(i>=100) break;
                                        i=i++;

                                        if(i>=100) return(-999);
                                        d1=vapor(t1)-e;
                                        d2=vapor(t2)-e;
```

```
                                if (d1*d2<0){
                                    t2=t2-(t2-t1)*0.5;
                                }
                                else{
                                        t3=t1;
                                        t1=t2;
                                        t2=t2+(t2-t3)*0.5;
                                }
                        }
                        dew=t1;
                }
                if(i==1) return(-999.0);
                else return(dew);
}
```