ERRATA SHEET FOR ANSI/ASHRAE STANDARD 55-2023
Thermal Environmental Conditions for Human Occupancy

September 16, 2024

The corrections listed in this errata sheet apply to ANSI/ASHRAE Standard 55-2023. The first printing is identified on the outside back cover as "Product code: 86897 11/23". Shaded items have been added since the previously published errata sheet dated March 21, 2024 was distributed.

Page	**Erratum**
10	**5.3.3 Solar Radiation Adjustment.** In Section 5.3.3.b.1 change the reference to Section 5.3.5.3 to Section 5.3.5.4 as shown below. *(Note: Additions are shown in underline and deletions are shown in ~~strikethrough~~.)*

5.3.3 Solar Radiation Adjustment. …

[…]

b. Use a mean radiant temperature t_r that is 2.8°C (5°F) higher than average air temperature t_a if all of the following conditions are met:

1. The space has air temperature stratification that meets the requirements of Section 5.3.5.4 ~~5.3.5.3~~.

[…]

37	**Table D-1 Validation Table for SET Computer Model (for a standing person).**
37	**D4. COMPUTER PROGRAM FOR CALCULATION OF SET.** Revise the code in Section D4 as shown in the attached. Changes are highlighted in yellow for clarity. *(Note: Additions are shown in underline and deletions are shown in ~~strikethrough~~.)*

D4. COMPUTER PROGRAM FOR CALCULATION OF SET

The following code is one implementation of the SET calculation using JavaScript in SI units. If the following code is used to calculate the cooling effect as described in Normative Appendix D, the input parameter CALCULATE_CE should be set equal to "true." Alternatively, if it is used to calculate the SET temperature, CALCULATE_CE should be set to "false."

```
FindSaturatedVaporPressureTorr = function(T) {
    /* Helper function for pierceSET calculates Saturated Vapor Pres
    sure(Torr)at Temperature T (°C) */
    return Math.exp(18.6686 4030.183/(T + 235.0));
}
pierceSET = function(TA, TR, VEL, RH, MET, CLO, WME, PATM,
CALCULATE_CE=false, BODY_POSITION) {

    /* Input variables TA (air temperature): °C,
        TR (mean radiant temperature): °C,
        VEL (air speed): m/s, RH (relative humidity): %,
        MET: met unit
        CLO: clo unit
        WME (external work): W/m2,
        PATM (atmospheric pressure): kPa
        BODY_POSITION (body position): "sitting" or "standing"*/
    var KCLO = 0.25;
    var BODYWEIGHT = 69.9;                  //kg
    var BODYSURFACEAREA = 1.8258;           //m2
    var METFACTOR = 58.2;                   //W/m2
    var SBC = 0.000000056697;               //Stefan-Boltzmann constant (W/m2K4)
    var CSW = 170.0;
    var CDIL = 200;
    var CSTR = 0.5;
    var LTIME = 60.0;
    var VaporPressure = RH * FindSaturatedVaporPressureTorr(TA)/100.0;
    var AirSpeed = Math.max(VEL, 0.1);
    var  TempSkinNeutral  =  33.7;
    var  TempCoreNeutral  =  36.8;
    var TempBodyNeutral = 36.49;
    var SkinBloodFlowNeutral = 6.3;
    var TempSkin = TempSkinNeutral;   //Initial values
    var TempCore = TempCoreNeutral;
    var SkinBloodFlow = SkinBloodFlowNeutral;
    var MSHIV = 0.0;
    var ALFA = 0.1;
    var ESK = 0.1 * MET;
    var PressureInAtmospheres = PATM * 0.009869;
    var RCL = 0.155 * CLO;
    var FACL = 1.0 + 0.15 * CLO;
    var LR = 2.2/PressureInAtmospheres; /* Lewis Relation is 2.2 at sea
                                                            level */
    var RM = MET * METFACTOR;
    var M = MET * METFACTOR;
```

```
if (CLO <= 0) {
    var WCRIT = 0.38 * Math.pow(AirSpeed, -0.29);
    var ICL = 1.0;
}
else {
    var WCRIT = 0.59 * Math.pow(AirSpeed, -0.08);
    var ICL = 0.45;
}
let heatTransferConvMet;
if (MET < 0.85) {
    heatTransferConvMet = 3.0;
}
else {
    heatTransferConvMet = Math.max (0.0, 5.66 * Math.pow(MET 0.85, 0.39));
}
let CHC = 3.0 * Math.pow(PressureInAtmospheres, 0.53);
let CHCV = 8.600001 * Math.pow(AirSpeed * PressureInAtmospheres,
  0.53);
CHC = Math.max(CHC, CHCV);
if (!CALCULATE_CE) {
    CHC = Math.max(CHC, heatTransferConvMet);
}
var CHR = 4.7;
var CTC = CHR + CHC;
var RA = 1.0/(FACL * CTC); /* Resistance of air layer to dry heat
                                   transfer*/
var TOP = (CHR * TR + CHC * TA)/CTC;
var TCL = ~~TOP + (TempSkin TOP)/(CTC * (RA + RCL));~~ (RA * TempSkin + RCL * TOP)/(RA + RCL);
/* TCL and CHR are solved iteratively using: H(Tsk TOP) =
CTC(TCL - TOP), where H = 1/(RA + RCL) and RA = 1/FACL*CTC */
var TCL_OLD = TCL;
var flag = true;
var DRY, HFCS, ERES, CRES, SCR, SSK, TCSK, TCCR, DTSK, DTCR, TB, SKSIG, WARMS,
     COLDS, CRSIG, WARMC, COLDC, BDSIG, WARMB, COLDB, REGSW, ERSW, REA,
     RECL, EMAX, PRSW, PWET, EDIF, ESK;
for (var TIM = 1; TIM <= LTIME; TIM++) {   //Begin iteration
  do {
        if (flag) {
          TCL_OLD = TCL;
            if (BODY_POSITION === "sitting") {
                // 0.7 ratio between radiation area of the body and
                // the body area
                CHR = 4.0 * 0.95 * SBC * Math.pow(((TCL + TR)/2.0 + 273.15),
                    3.0) *0.7;
        }else {// if standing
                // 0.73 ratio between radiation area of the body and
                // the body area
            CHR = 4.0 * 0.95 * SBC * Math.pow(((TCL + TR)/2.0 + 273.15),
                3.0) *0.73;
         }
            CTC = CHR + CHC;
            RA = 1.0/(FACL * CTC);*/Resistance of air layer to dry heat
            transfer*/
```

```
            TOP = (CHR * TR + CHC * TA)/CTC;
        }
    TCL = (RA * TempSkin + RCL * TOP)/(RA + RCL);
    flag = true;
    }
    while (Math.abs(TCL TCL_OLD) > 0.01);
        flag = false;
        DRY = (TempSkin TOP)/(RA + RCL);
        HFCS = (TempCore TempSkin) * (5.28 + 1.163 * SkinBloodFlow);
        ERES = 0.0023 * M * (44.0 VaporPressure);
        CRES = 0.0014 * M * (34.0 TA);
        SCR = M HFCS ERES CRES WME;
        SSK = HFCS DRY ESK;
        TCSK = 0.97 * ALFA * BODYWEIGHT;
        TCCR = 0.97 * (1 ALFA) * BODYWEIGHT;
        DTSK = (SSK * BODYSURFACEAREA)/(TCSK * 60.0); //°C/min
        DTCR = SCR * BODYSURFACEAREA/(TCCR * 60.0);        //°C/min
        TempSkin = TempSkin + DTSK;
        TempCore = TempCore + DTCR;
        TB = ALFA * TempSkin + (1 ALFA) * TempCore;
        SKSIG = TempSkin TempSkinNeutral;
            if (SKSIG > 0) {
            WARMS = SKSIG;
            COLDS = 0.0;
            }
            else {
                WARMS = 0.0;
                COLDS = -1.0 * SKSIG;
            }
        CRSIG = (TempCore TempCoreNeutral);
          if (CRSIG > 0) {
                WARMC = CRSIG;
                COLDC = 0.0;
        }
            else {
                WARMC = 0.0;
                COLDC = -1.0 * CRSIG;
        }
        BDSIG = TB - TempBodyNeutral;
        WARMB = (BDSIG > 0) * BDSIG;
        SkinBloodFlow = (SkinBloodFlowNeutral + CDIL * WARMC)/(1 + CSTR
            * COLDS);
        SkinBloodFlow = Math.max(0.5, Math.min(90.0, SkinBloodFlow));
        REGSW = CSW * WARMB * Math.exp(WARMS/10.7);
        REGSW = Math.min(REGSW, 500.0);
        var ERSW = 0.68 * REGSW;
        //Evaporative resistance of air layer
        var REA = 1.0/(LR * FACL * CHC);

        //Evaporative resistance of clothing (icl=.45)
        var RECL = RCL/(LR * ICL);
        var EMAX = (FindSaturatedVaporPressureTorr(TempSkin)
          VaporPressure)/(REA + RECL);
```

```
                var PRSW = ERSW/EMAX;
                var PWET = 0.06 + 0.94 * PRSW;
                var EDIF = PWET * EMAX - ERSW;
                var ESK = ERSW + EDIF;
                if (PWET > WCRIT) {
                    PWET = WCRIT;
                      PRSW = WCRIT/0.94;
                      ERSW = PRSW * EMAX;
                      EDIF = 0.06 * (1.0 - PRSW) * EMAX;
                      ESK = ERSW + EDIF;
                }
                if (EMAX < 0) {
                    EDIF = 0;
                      ERSW = 0;
                      PWET = WCRIT;
                      PRSW = WCRIT;
                      ESK = EMAX;
                }
                ESK = ERSW + EDIF;
                MSHIV = 19.4 * COLDS * COLDC;
                M = RM + MSHIV;
                ALFA = 0.0417737 + 0.7451833/(SkinBloodFlow + 0.585417);
                                 }                                          //End iteration

    var HSK = DRY + ESK;                    //Total heat loss from skin
    var RN = M WME;                         //Net metabolic heat production
    var ECOMF = 0.42 * (RN (1 * METFACTOR));
    if (ECOMF < 0.0) ECOMF = 0.0; //From Fanger
    EMAX = EMAX * WCRIT;
    var W = PWET;
    var PSSK = FindSaturatedVaporPressureTorr(TempSkin);
    var CHRS = CHR; //Definition of ASHRAE std. environment, denoted "S"
    CHCS = 3.0 * Math.pow(PressureInAtmospheres, 0.53);

    if (!CALCULATE_CE && MET > 0.85) {
          CHCS = Math.max(CHCS, heatTransferConvMet);
    }

    if (CHCS < 3.0) CHCS = 3.0;
    var CTCS = CHCS + CHRS;
    var RCLOS = 1.52/((MET WME/METFACTOR) + 0.6944) 0.1835;
    var RCLS = 0.155 * RCLOS;
    var FACLS = 1.0 + KCLO * RCLOS;
    var FCLS = 1.0/(1.0 + 0.155 * FACLS * CTCS * RCLOS);
    var IMS = 0.45;
    var ICLS = IMS * CHCS/CTCS * (1 FCLS)/(CHCS/CTCS FCLS * IMS);
    var RAS = 1.0/(FACLS * CTCS);
    var REAS = 1.0/(LR * FACLS * CHCS);
    var RECLS = RCLS/(LR * ICLS);
    var HD_S = 1.0/(RAS + RCLS);
    var HE_S = 1.0/(REAS + RECLS);

//SET determined using Newton's iterative solution
    var DELTA = .0001;
```

```
        var dx = 100.0;
        var SET, ERR1, ERR2;
        var SET_OLD = TempSkin HSK/HD_S;   //Lower bound for SET
        while (Math.abs(dx) > .01) {
            ERR1 = (HSK HD_S * (TempSkin SET_OLD) W * HE_S *
                (PSSK  0.5  *  FindSaturatedVaporPressureTorr(SET_OLD)));
            ERR2 = (HSK HD_S * (TempSkin (SET_OLD + DELTA)) W * HE_S *
                (PSSK 0.5 * FindSaturatedVaporPressureTorr((SET_OLD + DELTA)));
            SET = SET_OLD DELTA * ERR1/(ERR2 ERR1);
            dx = SET SET_OLD;
            SET_OLD = SET;
        }
    return SET;
    }
```