

ERRATA SHEET FOR ANSI/ASHRAE STANDARD 55-2013
Thermal Environmental Conditions for Human Occupancy

July 18, 2017

The corrections listed in this errata sheet apply to ANSI/ASHRAE Standard 55-2013. The first printing is identified on the outside back cover as “Product code: 86179 10/13”. The shaded items have been added since the previously published errata sheet dated April 7, 2016 was distributed.

<u>Page</u>	<u>Erratum</u>
--------------------	-----------------------

Inside Cover	ASHRAE Standing Standard Project Committee 55. In the SSPC 55 roster at the top of the inside cover change the committee members name from “Sahar A. Fard” to “Sahar Abbaszadeh”.
---------------------	--

4	3 DEFINITIONS. In the definition of <i>temperature, prevailing mean outdoor air</i> ($t_{pma(out)}$) on page 4 change “Figure 5.4.2.1” to “Figure 5.4.2”.
---	--

9	FIGURE 5.3.1. Graphic Comfort Zone Method: acceptable range of operative temperature (t_o) and humidity for spaces that meet the criteria in Section 5.3.1 ($1.0 \leq met < 1.3$; $0.5 < clo < 1.0$) – (a) I-P and (b) SI. Revise the first sentence in each of the notes for the two figures as shown below.
---	--

(Note: Additions are shown in underline and deletions are shown in ~~striketrough~~.)

When applying this Graphic per Section 5.3.1, use Operative Temperature determined in accordance with Appendix A ~~Appendix C~~.

10	Figure 5.3.3A Acceptable ranges of operative temperature (t_o) and average air speed (V_a) for the 1.0 and 0.5 clo comfort zone presented in Figure 5.3.1.1, at humidity ratio 0.010. In the title to Figure 5.3.3A change “Figure 5.3.1.1” to “Figure 5.3.1”. See attached corrections shown in red .
----	--

(Note: Additions are shown in underline and deletions are shown in ~~striketrough~~.)

14	7.2.2.2 Naturally Conditioned Spaces. In the second sentence of Section 7.2.2.2 change “(Figure 5.3.3B)” to “(Table 5.4.2.4)”.
----	---

14	7.3.2 Physical Measurement Positions within the Building. In Section 7.3.2b, <i>Height above floor</i> , change “(Section 5.2.4.4)” to “(Section 5.3.4.5)” in the first paragraph and change “(Sections 5.2.4.1)” to “(Section 5.3.4.2)” in the second paragraph.
----	--

15	7.4.1 Evaluation Based on Survey Results. Revise Section 7.4.1 as shown below.
----	---

(Note: Additions are shown in underline and deletions are shown in ~~striketrough~~.)

7.4.1 Evaluation Based on Survey Results

- a. The probability of occupants satisfied shall be predicted from seven-point satisfaction survey scores ~~shall be predicted~~ by dividing the number of votes falling between “just satisfied” -1 and “very satisfied,” +3 inclusive, by the total number of votes.

Responses to diagnostic dissatisfaction questions shall be tallied by category.

- b. For point-in-time surveys, comfort shall be evaluated using votes on the acceptability

and/or thermal sensation scales. ~~On each scale, votes between -1 and +3. On the acceptability scale, votes between 0 (neutral) and +3 ('very acceptable'), inclusive, shall be divided by total votes to obtain the probability of comfort acceptability observed during the survey period. On the seven-point thermal sensation scale, votes between -1.5 and +1.5, inclusive, shall be divided by total votes to obtain the probability of comfort acceptability observed during the survey period.~~

- 15 **8. REFERENCES.** Delete references 1, 5, and 6 from Section 8 as shown below. Addendum o to Standard 55-2010 revised Section 7 in 55-2010 and in doing so deleted Section 7.1, which originally cited these references, however the addendum neglected to delete those references from Section 8. These references are no longer cited in the text of the standard. (Note: Deletions are shown in ~~strikethrough~~.)

8. REFERENCES

¹~~ISO 7726:1998, Ergonomics of the Thermal Environment—Instruments for Measuring Physical Quantities.~~

⁵~~ANSI/ASHRAE Standard 70-2006, Method of Testing for Rating the Performance of Air Outlets and Inlets.~~

⁶~~ANSI/ASHRAE Standard 113-2009, Method of Testing for Room Air Diffusion.~~

- 27 **Section G1 Computer Program for Calculation of SET.** See attached computer program for corrections. Changes are shown highlighted in yellow. (Note: Additions are shown in underline and deletions are shown in ~~strikethrough~~.)
- 27 **Section G1 Computer Program for Calculation of SET.** See attached computer program for corrections. Changes are shown highlighted in green. (Note: Additions are shown in underline and deletions are shown in ~~strikethrough~~.)
- 30 **TABLE G1-1 Validation Table for SET Computer Model.** Based on the erratum correction to the core neutral temperature in the computer program for the calculation of SET in Appendix G the values in the validation table need to be corrected as well. Replace the current Table G1-1 with the corrected table attached.
- 34-35 **Informative Appendix I Occupant-Controlled Naturally Conditioned Spaces.** In the fourth and fifth paragraphs, top of right column on page 34, and in the sixth paragraph of Informative Appendix I, bottom of the left column of page 35, change “Figure 5.4.2.1” to “Figure 5.4.2”.
- 39 **K1.2 Environmental and Occupant Measurements.** In the second paragraph of Section K1.2 in Informative Appendix K change “Normative Appendices D and E” to “Informative Appendices D and E”.

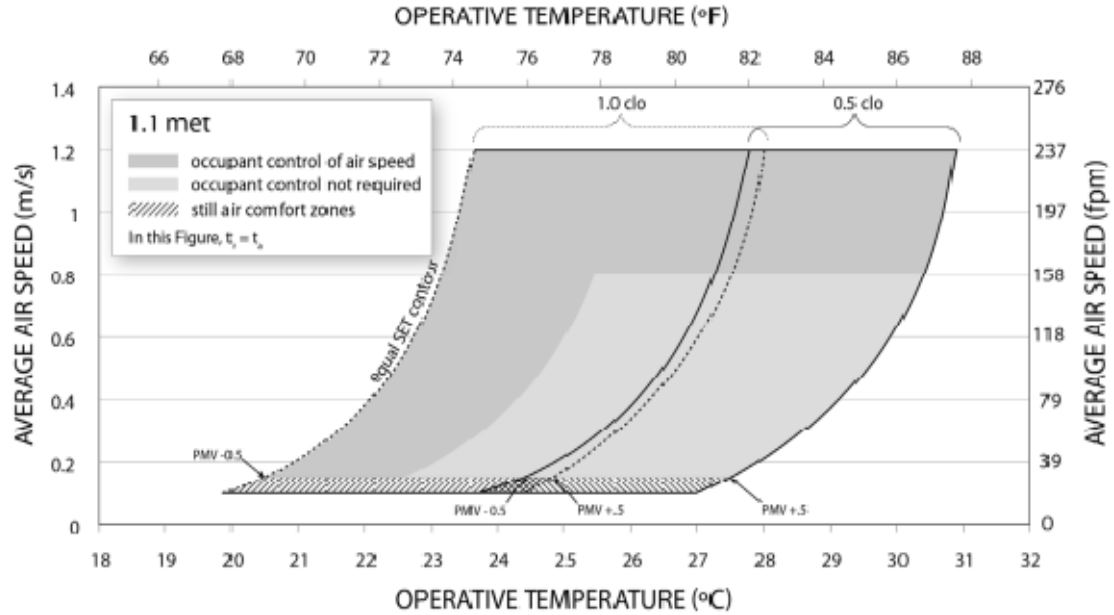


FIGURE 5.3.3A Acceptable ranges of operative temperature (t_o) and average air speed (V_a) for the 1.0 and 0.5 clo comfort zone presented in Figure [5.3.1](#) ~~5.3.1.1~~, at humidity ratio 0.010.

G1. COMPUTER PROGRAM FOR CALCULATION OF SET

For additional technical information and an I-P version of the equations in this appendix, refer to the *ASHRAE Thermal Comfort* Tool referenced in Section 8 of this standard. The Thermal Comfort Tool allows for I-P inputs and outputs, but the algorithm is implemented in SI units

Documentation for the SET model is Fountain and Huizenga (1995). (See Informative Appendix L, “Bibliography and Informative References.”)

Note: For the use of SET in ASHRAE Standard 55, the function for self-generated air speed as a function of met rate has been removed.

```
FindSaturatedVaporPressureTorr = function(T) {  
    // Helper function for pierceSET calculates Saturated Vapor Pressure (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) {  
    // input variables – TA (air temperature): °C, TR (mean radiant temperature): °C, VEL (air velocity): m/s,  
    // RH (relative humidity): %, MET: met unit, CLO: clo unit, WME (external work): W/m², PATM (atmospheric pressure):  
    kPa  
  
    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 = 120.0;  
    var CSTR = 0.5;  
    var LTIME = 60.0;  
  
    var VaporPressure = RH * FindSaturatedVaporPressureTorr(TA) / 100.0;  
    var AirVelocity = Math.max(VEL, 0.1);  
    var TempSkinNeutral = 33.7;  
    var TempCoreNeutral = 36.8; 36.49;  
    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(AirVelocity, -0.29);  
        var ICL = 1.0;  
    } else {  
        var WCRIT = 0.59 * Math.pow(AirVelocity, -0.08);  
        var ICL = 0.45;  
    }  
}
```

```

var CHC = 3.0 * Math.pow(PressureInAtmospheres, 0.53);
var CHCV = 8.600001 * Math.pow((AirVelocity * PressureInAtmospheres), 0.53);
var CHC = Math.max(CHC, CHCV);
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));

// 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;
            CHR = 4.0 * SBC * Math.pow(((TCL + TR) / 2.0 + 273.15), 3.0) * 0.72;
            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/minute
    DTCR = (SCR * BODYSURFACEAREA) / (TCCR * 60.0); //°C/minute
    TempSkin = TempSkin + DTSK;
    TempCore = TempCore + DTCR;
    TB = ALFA * TempSkin + (1 - ALFA) * TempCore;
    SKSIG = TempSkin - TempSkinNeutral;
    WARMS = (SKSIG > 0) * SKSIG;
    COLDS = ((-1.0 * SKSIG) > 0) * (-1.0 * SKSIG);
    if (SKSIG > 0) {
        WARMS = SKSIG;
        COLDS = 0.0;
    } else {
        WARMS = 0.0;
        COLDS = -1.0 * SKSIG;
    }
    CRSIG = (TempCore - TempCoreNeutral);
    WARMC = (CRSIG > 0) * CRSIG;
    COLDC = ((-1.0 * CRSIG) > 0) * (-1.0 * CRSIG);
    if (CRSIG > 0) {
        WARMC = CRSIG;
        COLDC = 0.0;
    } else {

```

```

WARMC = 0.0;
COLDC = -1.0 * CRSIG;
}
BDSIG = TB - TempBodyNeutral;
WARMB = (BDSIG > 0) * BDSIG;
if (BDSIG > 0){
    WARMB = BDSIG;
    COLDB = 0.0;
} else {
    WARMB = 0.0;
    COLDB = -1.0 * 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;
var REA = 1.0 / (LR * FACL * CHC); //evaporative resistance of air layer
var RECL = RCL / (LR * ICL); //evaporative resistance of clothing (icl=.45)
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 + .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 standard environment... denoted "S"
if (MET < 0.85) {
    var CHCS = 3.0;
} else {
    var CHCS = 5.66 * Math.pow(((MET - 0.85)), 0.39);
    CHCS = Math.max(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;
}

```

TABLE G1-1 Validation Table for SET Computer Model

Temperature		MRT		Velocity		RH	Met	Clo	SET	
°C	°F	°C	°F	m/s	fpm	%			°C	°F
25	77	25	77	0.15	29.5	50	1	0.5	23.8	74.9
0	32	25	77	0.15	29.5	50	1	0.5	12.3	54.1
10	50	25	77	0.15	29.5	50	1	0.5	17.0	62.5
15	59	25	77	0.15	29.5	50	1	0.5	19.3	66.7
20	68	25	77	0.15	29.5	50	1	0.5	21.6	70.8
30	86	25	77	0.15	29.5	50	1	0.5	26.4	79.6
40	104	25	77	0.15	29.5	50	1	0.5	34.3	93.7
25	77	25	77	0.15	29.5	10	1	0.5	23.3	74.0
25	77	25	77	0.15	29.5	90	1	0.5	24.9	76.8
25	77	25	77	0.1	19.7	50	1	0.5	24.0	75.2
25	77	25	77	0.6	118.1	50	1	0.5	21.4	70.5
25	77	25	77	1.1	216.5	50	1	0.5	20.3	68.6
25	77	25	77	3	590.6	50	1	0.5	18.8	65.8
25	77	10	50	0.15	29.5	50	1	0.5	15.2	59.3
25	77	40	104	0.15	29.5	50	1	0.5	31.8	89.2
25	77	25	77	0.15	29.5	50	1	0.1	20.7	69.3
25	77	25	77	0.15	29.5	50	1	1	27.3	81.1
25	77	25	77	0.15	29.5	50	1	2	32.5	90.4
25	77	25	77	0.15	29.5	50	1	4	37.7	99.8
25	77	25	77	0.15	29.5	50	0.8	0.5	23.3	73.9
25	77	25	77	0.15	29.5	50	2	0.5	29.7	85.5
25	77	25	77	0.15	29.5	50	4	0.5	36.0	96.7