SoftwareX 12 (2020) 100578

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

pythermalcomfort: A Python package for thermal comfort research

Federico Tartarini [a,*], Stefano Schiavon [b]

[a] Berkeley Education Alliance for Research in Singapore, Singapore
[b] Center for the Built Environment, University of California Berkeley, USA

ARTICLE INFO

Article history:
Received 9 June 2020
Received in revised form 16 July 2020
Accepted 30 July 2020

Keywords:
Thermal comfort
ASHRAE
Predicted mean vote
Adaptive comfort

ABSTRACT

We developed `pythermalcomfort`, a Python package that allows users to calculate the most common thermal comfort indices in compliance with the main thermal comfort standards. For example, `pythermalcomfort` can be used to calculate: whole body thermal comfort indices (e.g., Predicted Mean Vote, adaptive models, Standard Equivalent Temperature), local discomfort, clothing insulation, and psychrometric properties of air. All `pythermalcomfort` functions have been validated against the reference tables provided in the corresponding thermal comfort standards. We have developed documentation, examples and tutorial videos to guide users on how to use our package. `pythermalcomfort` allows researchers and professionals to accurately perform complex thermal comfort calculations without the need of re-writing the programming code. With Python being among the most widely utilized programming languages and `pythermalcomfort` being the only Python library which includes a comprehensive list of thermal comfort functions, we believe that `pythermalcomfort` will have a significant impact on both the research and industrial communities.

Code metadata

Current code version	v1.2.0
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX_2020_250
Legal Code License	MIT
Code versioning system used	git
Software code languages, tools, and services used	Python
If available Link to developer documentation/manual	https://pythermalcomfort.readthedocs.io/en/latest/readme.html
Support email for questions	cbecomforttool@gmail.com

1. Motivation and significance

People spend the great majority of their time indoors. Indoor thermal environmental conditions significantly affect people well-being, performances and their overall satisfaction with the built environment [1–3]. Consequently, a series of indices had been developed, tested and implemented to assess the quality of the indoor thermal environment. Among these models, the Predicted Mean Vote (*PMV*) [4], the adaptive thermal comfort models [5,6], and the Standard Effective Temperature (*SET*) [7] have been widely adopted by researchers and practitioners worldwide. These models are now included in many National Building Codes and in International Standards such as the ASHRAE 55–2017, EN 16798–1:2019 and ISO 7730:2005 [8–10]. Both the *PMV* and the adaptive are aggregate models, which means that they aim to predict how a group of people would perceive their thermal environment in terms of given environmental (e.g., relative humidity (RH), dry-bulb air temperature (t_{db})) and personal (e.g., total clothing insulation (I_{cl}) and metabolic rate (M)) parameters. *PMV* is the reference index to assess thermal comfort conditions in mechanically conditioned buildings, while the EN and ASHRAE adaptive models are used to assess the thermal environment in naturally conditioned buildings [11].

Programming codes to calculate the above mentioned indices are available both in the ASHRAE 55 [8] and the ISO 7730 [10] Standards. These Standards do also provide procedure and methods to calculate other thermal comfort indices (e.g., *SET*). However, these programming codes are either written in JavaScript or BASIC and the above mentioned standards are not freely available. Consequently even for a skilled Python user it may take a significant amount of time to access the code, re-write it and validate it

* Corresponding author.
E-mail address: federicotartarini@berkeley.edu (F. Tartarini).

https://doi.org/10.1016/j.softx.2020.100578

Nomenclature

CE	Cooling Effect (K)
I_{cl}	Total clothing insulation (clo)
M	Metabolic rate (met)
PMV	Predicted Mean Vote (–)
PPD	Predicted Percentage of Dissatisfied (%)
RH	Relative Humidity (%)
SET	Standard Effective Temperature (°C)
t_{db}	Dry-bulb air temperature (°C)
t_g	Globe temperature (°C)
$\overline{t_r}$	Mean radiant temperature (°C)
UTCI	Universal Thermal Climate Index (°C)
v	Average air speed (m/s)
v_r	Relative average air speed (m/s)

against reference tables to ensure that it does not contain errors. Making it an inefficient, long and error-prone process.

Python [12] is a general purpose programming language which is gaining popularity not only among programmers but also among researchers. In 2020, it was one of the top five most used programming languages worldwide [13]. PyPI is the official third-party software repository for Python which contains more than 200,000 packages [14]. However, only two packages available on PyPI, i.e., `comfort` [15] and `ladybug-comfort` [16], allow users to calculate thermal comfort indices. The `comfort` package is released under the MIT licence and its last update was released in 2018. It can only be used to calculate the *PMV* and Predicted Percentage of Dissatisfied (*PPD*) indices and provides very limited documentation on how to use the package [15]. The `ladybug-comfort` package is specifically designed to add thermal comfort functionalities to Ladybug. It has been released under GPLv3 licence by Ladybug Tools [17] and it is frequently updated. Ladybug is an open source plugin for Grasshopper that performs detailed analysis of climate data to produce customized and interactive visualizations for environmentally-informed architectural design [16]. Since we did not find a comfort package specifically designed for thermal comfort research on PyPI, we expanded our research and looked for packages developed for other programming languages. We found an R package named `comf` which comprises a comprehensive list of functions to calculate all commonly used thermal comfort indices [18]. However, it cannot be compiled and used in Python.

We, therefore, decided to develop the `pythermalcomfort` package to allow researchers and professionals who use Python to calculate all commonly used thermal comfort indices. We released the code open-source so other users can contribute to our project or use our code in other applications. We validated all functions against reference tables to ensure that results are accurate; and, developed supporting documentation and tutorials to help and guide users who either have limited programming skills or limited knowledge about thermal comfort.

2. Software description

The `pythermalcomfort` package can be freely downloaded from the Python Project Inventory (PyPI) using the following command: "`pip3 install pythermalcomfort`". The package can also be used in multiple platforms and operative systems inside a Python interpreter or in script mode.

2.1. Software architecture

The `pythermalcomfort` package contains three main modules: `models.py`, `psychrometrics.py` and `utilities.py`. `models.py` contains functions (see Section 2.2 for a detailed description) to calculate the thermal comfort indices. `psychrometrics.py` contains functions to calculate thermodynamic properties of air and other parameters such as the running mean temperature. `utilities.py` contains reference dictionaries (see Section 2.2 for more information) and functions which are needed to perform some background calculations.

2.2. Software functionalities

The list below gives an overview of the main functions that we have included in this package. We, however, recommend to consult the official project documentation available at https://pythermalcomfort.readthedocs.io for more and up-to-date information since new features will possibly be released in the future. Moreover, we do not report here all the necessary details to understand the calculated indices, these information should be found in the standards and relevant literature. All the functions below can be imported from the module `pythermalcomfort.models`.

- `pmv_ppd(tdb, tr, vr, rh, met, clo)` - returns *PMV* and *PPD* values in accordance with either the ASHRAE 55–2017 [8] or the ISO 7730:2005 Standards [10]. It also allows users to choose between the SI or IP systems of units. The *PMV* is an index that predicts the mean value of the thermal sensation votes of a large group of people on a sensation scale expressed from −3 to +3 corresponding to the categories "cold", "cool", "slightly cool", "neutral", "slightly warm", "warm", and "hot", respectively. The *PPD* is an index that establishes a quantitative prediction of the percentage of thermally dissatisfied people based on the PMV [8] value.
- `pmv(tdb, tr, vr, rh, met, clo)` - returns only the *PMV* value.
- `set_tmp(tdb, tr, v, rh, met, clo)` - returns *SET* calculated in compliance with the ASHRAE 55–2017 Standard. SET is the temperature of an imaginary environment at RH = 50%, average air speed (v) < 0.1 m/s, and mean radiant temperature $\overline{t_r} = t_{db}$, in which the total heat loss from the skin of an imaginary occupant with a M = 1.0 met and I_{cl} = 0.6 clo is the same as that from a person in the actual environment with actual clothing and activity level [8].
- `cooling_effect(tdb, tr, vr, rh, met, clo)` - returns the Cooling Effect (*CE*) calculated in compliance with the ASHRAE 55–2017 Standard. *CE* is the value that, when subtracted equally from both the average t_{db} and $\overline{t_r}$, yields the same *SET* under still air as in the first *SET* calculation under elevated air speed [8].
- `adaptive_ashrae(tdb, tr, t_running_mean, v)` - returns the acceptable indoor operative temperature ranges based on the prevailing mean outdoor temperature [6,8]. Values are calculated in compliance with the ASHRAE 55–2017.
- `adaptive_en(tdb, tr, t_running_mean, v)` - returns the acceptable indoor operative temperature ranges based on the prevailing mean outdoor temperature. Values are calculated in compliance with the EN 16798–1:2019 [5,9].
- `solar_gain(sol_altitude, sol_azimuth, sol_radiation_dir, sol_transmittance, f_svv, f_bes, asw=0.7, posture=''seated'')` - returns the Effective Radiant Field (*ERF*) and $\Delta\overline{t_r}$. The *ERF* is a measure of the net energy flux to or from the human body, which it is

expressed in power per body surface area [W/m^2]. The $\Delta\overline{t_r}$ is the amount by which the mean radiant temperature of the space should be increased if no solar radiation was to be present [8,19].

- `utci(tdb, tr, v, rh)` - returns the Universal Thermal Climate Index (UTCI) value. UTCI is an outdoor thermal comfort index [20]. It is defined as the air temperature of the reference environment which produces the same strain index value in comparison with the reference individual's response to the real environment.
- `clo_tout(tout)` - returns the total clothing insulation I_{cl}. It estimates the representative I_{cl} as a function of the outdoor air temperature at 06:00 a.m. [21].
- `vertical_tmp_grad_ppd(tdb, tr, vr, rh, met, clo, vertical_tmp_grad)` - returns *PPD* with vertical temperature gradient between feet and head [8,22].
- `ankle_draft(tdb, tr, vr, rh, met, clo, v_ankle)` - returns *PPD* with the ankle draft at 0.1 m above floor level [8,23].

In addition, this package offers functions to calculate thermodynamic properties of air (e.g., enthalpy, vapor pressure of water, etc.), the prevailing mean outdoor temperature (input for the adaptive models), the relative air velocity and the dynamic clothing insulation. The package also includes dictionaries to estimate M of typical activities (met_typical_tasks), the I_{cl} values of typical ensembles (clo_typical_ensembles) and clo value of individual garments (clo_individual_garments). These dictionaries have been included in the `pythermalcomfort.utilities` module.

2.3. Software exceptions and warnings

We implemented warnings to alert users if input variables do not comply with the standard applicability limits. For example the ISO 7730:2005 [10] Standard states that the *PMV* index is only valid when t_{db} ranges from 10 °C to 30 °C, while the ASHRAE 55–2017 Standard [8] states that the applicability limits are from 10 °C to 40 °C. Hence, different warning are raised based on the standard selected.

Python exceptions are used to throw errors when the user specifies input values that are not allowed. For example, when the user specifies an units system which is neither SI nor IP in the SET equation. The official documentation contains a list of all the exceptions that each function may raise.

2.4. Software validation

`pytest` was used to write tests for all the functions included in `pythermalcomfort`. When available, data from validation tables published in the relevant standards were used to verify the output of the functions. All the tests we performed can be found in the 'test' directory. Tables 1 and 2 show side-by-side the PMV values calculated using `pythermalcomfort` and the values from the validation tables provided in the ISO 7730:2005 and ASHRAE 55–2017 Standards, respectively. Table 3 contains the *SET* values calculated using `pythermalcomfort` and the values from the validation Table D3 of the ASHRAE 55–2017 Standard. Based on the results obtained we concluded that the *PMV* and *SET* functions comply with the above mentioned standards since our results matched the reference values, with the only exception of row 6 in Table 1. In this manuscript, we decided not to report the *PPD* values since it is calculated using a relatively simple equation. However, in our package the *PPD* equation is also tested.

Table 1
PMV values calculated with the ISO 7730:2005 Standard and pythermalcomfort.

	Input data						ISO	pythermalcomfort
	t_{db}	$\overline{t_r}$	RH	v_r	M	I_{cl}	PMV	PMV
0	22.0	22.0	60	0.1	1.2	0.5	−0.75	−0.75
1	27.0	27.0	60	0.1	1.2	0.5	0.77	0.77
2	27.0	27.0	60	0.3	1.2	0.5	0.44	0.43
3	23.5	25.5	60	0.1	1.2	0.5	−0.01	−0.01
4	23.5	25.5	60	0.3	1.2	0.5	−0.55	−0.55
5	19.0	19.0	40	0.1	1.2	1.0	−0.60	−0.60
6	23.5	23.5	40	0.1	1.2	1.0	0.50	0.36
7	23.5	23.5	40	0.3	1.2	1.0	0.12	0.12
8	23.0	21.0	40	0.1	1.2	1.0	0.05	0.05
9	23.0	21.0	40	0.3	1.2	1.0	−0.16	−0.17
10	22.0	22.0	60	0.1	1.6	0.5	0.05	0.05
11	27.0	27.0	60	0.1	1.6	0.5	1.17	1.17
12	27.0	27.0	60	0.3	1.6	0.5	0.95	0.95

Table 2
PMV values calculated with the ASHRAE 55–2017 Standard and pythermalcomfort.

	Input data						ASHRAE	pythermalcomfort
	t_{db}	$\overline{t_r}$	RH	v_r	M	I_{cl}	PMV	PMV
0	19.6	19.6	86	0.1	1.1	1.0	−0.5	−0.5
1	23.9	23.9	66	0.1	1.1	1.0	0.5	0.5
2	25.7	25.7	15	0.1	1.1	1.0	0.5	0.5
3	21.2	21.2	20	0.1	1.1	1.0	−0.5	−0.5
4	23.6	23.6	67	0.1	1.1	0.5	−0.5	−0.5
5	26.8	26.8	56	0.1	1.1	0.5	0.5	0.5
6	27.9	27.9	13	0.1	1.1	0.5	0.5	0.5
7	24.7	24.7	16	0.1	1.1	0.5	−0.5	−0.5

Table 3
SET values calculated with the ASHRAE 55–2017 Standard and pythermalcomfort.

	Input data						ASHRAE	pythermalcomfort
	t_{db}	$\overline{t_r}$	RH	v_r	M	I_{cl}	SET	SET
0	25	25	10	0.15	1	0.5	23.3	23.3
1	0	25	50	0.15	1	0.5	12.3	12.4
2	10	25	50	0.15	1	0.5	17.0	17.0
3	15	25	50	0.15	1	0.5	19.3	19.3
4	20	25	50	0.15	1	0.5	21.6	21.6
5	30	25	50	0.15	1	0.5	26.4	26.4
6	40	25	50	0.15	1	0.5	34.3	34.3
7	25	25	10	0.15	1	0.5	23.3	23.3
8	25	25	90	0.15	1	0.5	24.9	24.9
9	25	25	50	0.10	1	0.5	24.0	24.0
10	25	25	50	0.60	1	0.5	21.4	21.4
11	25	25	50	1.10	1	0.5	20.3	20.4
12	25	25	50	3.00	1	0.5	18.8	18.8

3. Illustrative examples

The following examples demonstrate how to use the `pythermalcomfort` package to calculate some widely used thermal comfort indices. More examples can be found in the package official documentation at https://pythermalcomfort.readthedocs.io/en/latest/usage.html which also contains a link to the online video tutorials.

3.1. PMV and PPD calculation

Source Code 1 shows step by step how a user can calculate the *PMV* and *PPD* indices using `pythermalcomfort`. We assume that Python version 3.6 or higher has already been installed on the user's machine. The `pythermalcomfort` package can be downloaded and installed from PyPI by running the following command in the terminal: "`pip3 install pythermalcomfort`". More information on how to install a package with Python Package Installer (`pip`) can be found on `pip` official web-site.

Firstly, in Source Code 1 is shown that the user needs to import the necessary functions and dictionaries (see line 1 to 4 in Source Code 1). Secondly, the following parameters: t_{db}, $\overline{t_r}$, v, RH, I_{cl} and M have to be specified by the user. In this example, we assume that the user has measured the environmental parameters (i.e., t_{db}, $\overline{t_r}$, v, RH, see lines 6 to 10), logged what the occupants were wearing (list of garments, line 12) and their activity levels (line 11). The user can determine the clothing insulation (I_{cl}) and metabolic rate (M), respectively, using the `clo_individual_garments` and `met_typical_tasks` dictionaries as shown in lines 14 and 15. I_{cl} can be determined by adding together the clo value of all the garments that the representative occupant is wearing. Finally, the relative air velocity and the dynamic clothing insulation which are both a function of the body movement can be calculated using the respective functions as shown in line 18 and 21. The *PMV* and *PPD* values can then be computed, in accordance with the ASHRAE 55–2017 Standard [8], by entering the six input parameters in the `pmv_ppd()` function.

```
1 from pythermalcomfort.models import pmv_ppd
2 from pythermalcomfort.psychrometrics import v_relative
3 from pythermalcomfort.utilities import met_typical_tasks
4 from pythermalcomfort.utilities import clo_individual_garments
5
6 # input variables
7 tdb = 27  # dry-bulb air temperature, [°C]
8 tr = 25  # mean radiant temperature, [°C]
9 v = 0.1  # average air velocity, [m/s]
10 rh = 50  # relative humidity, [%]
11 # partcipant's activity description
12 activity = "Typing"
13 garments = ["Sweatpants", "T-shirt", "Shoes or sandals"]
14
15 # activity met, [met]
16 met = met_typical_tasks[activity]
17 # calculate total clothing insulation
18 icl = sum([clo_individual_garments[item] for item in garments])
19
20 # calculate the relative air velocity
21 vr = v_relative(v=v, met=met
22
23 # calculate dynamic clothing insulation
24 icl = clo_dynamic(clo=icl, met=met)
25
26 # calculate PMV in accordance with the ASHRAE 55
27 results = pmv_ppd(tdb=tdb, tr=tr, vr=vr, rh=rh, met=met, clo=icl, standard="ASHRAE")
28
29 # print the results
30 print(results)
31 # {"pmv": -0.11, "ppd": 5.2}
32
33 # print PMV value
34 print(results["pmv"])
35 # -0.11
36
37 # for users who wants to use the IP system
38 results_ip = pmv_ppd(tdb=80.6, tr=77, vr=0.426, rh=50, met=1.1, clo=0.38, units="IP")
39
40 print(results_ip)
41 # {"pmv": -0.11, "ppd": 5.2}
```

Source Code 1: Code to calculate the PMV and PPD indices using the pythermalcomfort Python package. The example also shows how to change the system of units.

The `pythermalcomfort` package also allows users to specify their preferred system of units between the international system (SI) and the imperial system (IP). Line 38 in Source Code 1 shows how the user can change the units in the `pmv_ppd` equation. It should be noted that all equations by default perform calculations in SI units.

3.2. Solar gain on people

The code shown in Source Code 2 shows how use the `solar_gain` function to calculate the *ERF* and $\Delta\overline{t_r}$ [8,19]. Firstly the user needs to import the function, then needs to specify the input values needed to calculate the solar gain on the human body.

```
from pythermalcomfort.models import solar_gain

# Solar altitude from horizontal, [deg]
s_alt = 0
# Solar azimuth clockwise from North, [deg]
s_az = 120
# Direct-beam solar radiation, [W/m2]
rad = 800
# Total solar transmittance of window systems
tr = 0.5
# Fraction of sky vault exaposed to body
f_svv = 0.5
# Fraction of body surface exposed to sun
f_bes = 0.5
# Average short-wave absorptivity of the occupant
asw = 0.7
posture = "seated"

results = solar_gain(sol_altitude=s_alt, sol_azimuth=s_az, sol_radiation_dir=rad, sol_transmittance=tr, f_svv=f_svv, f_bes=f_bes, asw=asw, posture=posture)

print(results)
# {"erf": 42.9, "delta_mrt": 10.3}
```

Source Code 2: Code to calculate solar gain on an occupant using the pythermalcomfort Python package.

The value of $\Delta\overline{t_r}$ calculated in Source Code 2 shall be added to the value of $\overline{t_r}$ and the resulting value is used as input to calculate *PMV*.

4. Impact

The building sector alone accounts for approximately one third of the total final energy consumption in the world [24]. The International Energy Agency (IEA) estimates that the energy used for heating and cooling accounts for 50% and 10% of the energy consumption in buildings, respectively [24]. The heating and cooling systems in buildings aim to provide comfortable indoor environmental conditions for occupants. However, a survey of more than 90,000 occupants in 900 office buildings found that 39% of the respondents were dissatisfied with their thermal environment [25]. A quantitative assessment of thermal comfort conditions indoors can be done by calculating thermal comfort indices to identify where and when building are thermally uncomfortable. However, calculating these thermal comfort indices could be a complex task, even for engineers and architects with extensive working experience in the building sector. The open-sourced `pythermalcomfort` package developed by us could significantly help building professionals and researchers in perform detailed thermal comfort analyses, in terms of cost-effectiveness, reliability and simplicity. Our is the only Python package available on PyPI which contains an exhaustive list of validated thermal comfort models. We have also generated documentation, video tutorials and source code examples to guide users with limited programming or thermodynamic skills to use this package.

With Python being one of the most widely used programming languages worldwide, we foresee that pythermalcomfort will be used by many users. Our package can be used for industrial, educational and product development purposes. For example, educators can use it to show how each environmental parameter affects how humans perceive their thermal environment. Architects and engineers can use it to calculate thermal comfort indices from time-series data collected by building management systems or estimated by building performance simulation softwares. This package can also be embedded in control algorithms of building management systems. pythermalcomfort together with the CBE Thermal Comfort Tool [26] and the R comf package [18] are free open-source tools specifically designed to translate research findings into practical industrial applications. These tools may help to curb the overall energy consumption of the building sector while improving occupants well-being, productivity, and health. When using these thermal comfort indices users should be aware of their limitations and their applicability limits.

5. Conclusions

We developed pythermalcomfort an open-source Python package that allows users to calculate the most common thermal comfort indices in compliance with the main thermal comfort standards. In addition, we also developed detailed documentation and tutorial videos to guide users on how to install and use of our package. pythermalcomfort is a powerful tool to conduct thermal comfort analyses. Our package can be used in educational settings to raise awareness regarding thermal comfort; by architects and engineers to improve the design and optimize the operation of naturally and mechanically conditioned buildings; and, by researchers to calculate thermal comfort indices in scientific publications.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We would like to acknowledge the work of all the authors who contributed to the development of the PyPI package https://pythermalcomfort.readthedocs.io/en/latest/authors.html; the GitHub repository https://github.com/CenterForTheBuiltEnvironment/pythermalcomfort/graphs/contributors; and, the CBE Thermal Comfort Tool, since pythermalcomfort is a derivative work of this project. This research has been supported by the Republic of Singapore's National Research Foundation through a grant to the Berkeley Education Alliance for Research in Singapore (BEARS) for the Singapore-Berkeley Building Efficiency and Sustainability in the Tropics (SinBerBEST) Program.

References

[1] de Dear RJ, Akimoto T, Arens EA, Brager G, Candido C, Cheong KWD, et al. Progress in thermal comfort research over the last twenty years. Indoor Air 2013;23(6):442–61. http://dx.doi.org/10.1111/ina.12046.

[2] Zhang F, de Dear R, Hancock P. Effects of moderate thermal environments on cognitive performance: A multidisciplinary review. Appl Energy 2019;236(November 2018):760–77. http://dx.doi.org/10.1016/j.apenergy.2018.12.005.

[3] Wyon D, Wargocki P. Room temperature effects on office work. In: Creating the productive workplace. Taylor and Francis; 2006, Ch. 11.

[4] Fanger PO. Thermal comfort: analysis and applications in environmental engineering. New York: McGraw-Hill; 1970.

[5] Nicol J, Humphreys M. Adaptive thermal comfort and sustainable thermal standards for buildings. Energy Build 2002;34(6):563–72. http://dx.doi.org/10.1016/S0378-7788(02)00006-3.

[6] de Dear R, Brager G, Cooper D. Developing an adaptive model of thermal comfort and preference - Final Report on RP-884. Australia: Macquarie University; 1997.

[7] Gagge, A. P., Fobelets, A. P. and Berglund LG. A standard predictive index of human response to thermal environment. ASHRAE Trans 1986;(92(2B)):709–31.

[8] ANSI, ASHRAE. Standard 55 - Thermal environmental conditions for human occupancy. 2017.

[9] CEN. EN 16798-1 - Energy performance of buildings - Ventilation for buildings. Part 1: Indoor environmental input parameters for design and assessment of energy performance of buildings addressing indoor air quality, thermal environment, lighting and acoustics. 2019.

[10] ISO. ISO 7730 - Ergonomics of the thermal environment - Analytical determination and interpretation of thermal comfort using calculation of the PMV and PPD indices and local thermal comfort criteria. 2005.

[11] Kim J, Tartarini F, Parkinson T, Cooper P, Dear RD. Energy & Buildings Thermal comfort in a mixed-mode building : Are occupants more adaptive ?. Energy Build 2019;203:109436. http://dx.doi.org/10.1016/j.enbuild.2019.109436.

[12] van Rossum G. Python tutorial. Tech. Rep. CS-R9526, Amsterdam: Centrum voor Wiskunde en Informatica (CWI); 1995.

[13] IEEE Spectrum. The top programming languages 2019. 2019, https://spectrum.ieee.org/computing/software/the-top-programming-languages-2019, accessed: 2020-03-04.

[14] Python Software Fundation. Pypi. 2020, https://pypi.org, accessed: 2020-03-19.

[15] Chuan-Che (Jeff) Huang. Comfort. 2018, https://pypi.org/project/comfort, accessed: 2020-03-19.

[16] Ladybug Tools. Ladybug. 2020, https://www.ladybug.tools/ladybug.html, accessed: 2020-03-19.

[17] Ladybug Tools. Ladybug-comfort. 2019, https://pypi.org/project/ladybug-comfort, accessed: 2020-03-19.

[18] Schweiker M. Comf: An R Package for Thermal Comfort Studies. R J 2016;8(2):341. http://dx.doi.org/10.32614/RJ-2016-050.

[19] Arens E, Hoyt T, Zhou X, Huang L, Zhang H, Schiavon S. Modeling the comfort effects of short-wave solar radiation indoors. Build Environ 2015;88:3–9. http://dx.doi.org/10.1016/j.buildenv.2014.09.004.

[20] Fiala D, Havenith G, Brode P, Kampmann B, Jendritzky G. Deriving the operational procedure for the Universal Thermal Climate Index (UTCI). Int J Biometeorol 2012;56:429–41. http://dx.doi.org/10.1007/s00484-011-0454-1.

[21] Schiavon S, Lee KH. Dynamic predictive clothing insulation models based on outdoor air and indoor operative temperatures. Build Environ 2013;59:250–60. http://dx.doi.org/10.1016/j.buildenv.2012.08.024.

[22] Liu S, Wang Z, Schiavon S, He Y, Luo M, Zhang H, et al. Predicted percentage dissatisfied with vertical temperature gradient. Energy Build 2020;220:110085. http://dx.doi.org/10.1016/j.enbuild.2020.110085.

[23] Liu S, Schiavon S, Kabanshi A, Nazaroff WW. Predicted percentage dissatisfied with ankle draft. Indoor Air 2017;27(4):852–62. http://dx.doi.org/10.1111/ina.12364.

[24] International Energy Agency. Buildings a source of enormous untapped efficiency potential. 2020, https://www.iea.org/topics/buildings, accessed: 2020-03-19.

[25] Graham LT, Parkinson T, Schiavon S. Where do we go now? Lessons learned from 20 years of CBE's Occupant Survey. CBE Report; 2020, URL https://escholarship.org/uc/item/8k20v82j.

[26] Tartarini F, Schiavon S, Cheung T, Hoyt T. CBE Thermal Comfort Tool: online tool for thermal comfort calculations and visualizations. SoftwareX 2020;12:100563. http://dx.doi.org/10.1016/j.softx.2020.100563.