Applied Chemical Engineering (2025) Volume 8 Issue 3 ISSN: 2578-2010 (O)
doi: 10.59429/ace.v8i3.5695

ORIGINAL RESEARCH ARTICLE

Integration of python programming in renewable energy studies: A flat plate collector model

Choon Kit Chan[1], Pankaj Dumka[2], Pradeep Boka[3], Pallavi Shrikrushnarao Thakare[4], Ghanshyam G Tejani[5,6], Feroz Shaik[7], L. Syam Sundar[7]

[1] *Faculty of Engineering and Quantity Surveying, INTI International University, Putra Nilai, Negeri Sembilan, 71800, Malaysia.*

[2] *Department of Mechanical Engineering, Jaypee University of Engineering and Technology, A.B. Road, Raghogarh, Guna, Madhya Pradesh, 473226, India*

[3] *Department of Mechanical Engineering, Government Engineering College Palanpur, Gujarat, 380001, India.*

[4] *Department of Information technology, Dr. D. Y. Patil Institute of Technology, Pimpri, Pune, Maharashtra, 411018, India*

[5] *Department of Research Analytics, Saveetha Dental College and Hospitals, Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai, 600077 India*

[6] *Applied Science Research Center, Applied Science Private University, Amman, 11937, Jordan*

[7,8] *Department of Mechanical Engineering, College of Engineering, Prince Mohammad Bin Fahd University, Al-Khobar, 31952, Saudi Arabia*

***Corresponding author:** Choon kit Chan ; choonkit.chan@newinti.edu.my

ABSTRACT

This paper studies the application of Python programming in renewable energy research through the development of a model for a flat plate solar collector. These collectors plays a very important role in solar thermal systems especially in in low to medium-temperature applications. Common modelling methods frequently involve the manual calculations (which are sometimes error prone) or the use of commercial software which are often costly. By utilizing Python's open-source environment along with the efficient libraries like NumPy and Matplotlib, this article presents an iterative, reproducible, and computationally efficient construct. The developed Python model integrates key heat transfer principles which governs the flat plate collectors and accounting for thermal and radiative interactions between the absorber plate, glass cover, and surrounding environment. The developed Python functions are used to calculate the essential parameters such as Rayleigh and Nusselt numbers along with the heat transfer coefficients and loss factors. The developed model offers a versatile educational tool for renewable energy studies and will provide a robust foundation for improving the research in solar thermal system optimization.

Keywords: Flat plate collector; heat transfer; python programming; solar energy, energy efficiency

ARTICLE INFO

Received: 6 July 2025
Accepted: 15 September 2025
Available online: 19 September 2025

1. Introduction

Renewable energy has come up as an alternate source of energy to address the climate change problem by reducing the global reliance on fossil fuels [1-3]. Among all the sources of renewable technologies, the solar energy stands out for its extensive availability and potential to give clean and sustainable power [4]. Flat plate collectors (FPC) are one of the most used solar thermal devices which finds application in water heating, space heating, and industrial preheating processes [5-7]. The performance of these collectors is affected by the complex physics which encompasses thermal, optical, and environmental engineering. Therefore, a precise modelling is very much need to enhance their design and efficiency [8].

Generally, the flat plate collectors are modelled by using either manual calculations or specialized simulation software. However, both of these methods present limitations in terms of the cost and accessibility for the students and researchers [9-11]. With the increasing number of programming languages, the way the scientific modelling was done previously has changed a lot and Python being a versatile tool, is doing wonders in the field of scientific computation [12,13]. It gives an open-source and user-friendly environment for building mathematical models, simulations, and creating visual descriptions. Its strong library network viz. NumPy [14-16], Matplotlib [17-19], and Pandas [20,21] can seamlessly handle complex datasets and advanced numerical computations, thus making it good aid for the applications in the renewable energy field.

This article focuses on the development of a Python-based model to know the performance of FPC. By using Python programming, this study establishes a reproducible, simple, and computationally effective method to analyse the performance of a FPC under different base plate thicknesses. The objective of the article is to gain insights into the thermodynamic behaviour of FPC and showcasing the potential of Python programming language as a powerful tool in the field of renewable energy modelling. Beyond serving as an educational resource, the developed Python model sets a foundation for the future developments in the study of solar thermal systems.

2. Methodology

2.1. FPC fundamentals

FPC are among the most used solar thermal technologies known for their simple design, stability, and efficiency in catching the solar energy for the applications which requires low to medium temperature. The core parts of a FPC are [5]:

Absorber Plate (AP): Absorber plate is normally made up of metal and absorbs solar radiation which are then converted it into heat. To improve its radiation absorption efficiency, it is coated with black material that enhances.

Transparent Cover: It is the top covering of FPC, and it is normally made up of glass or plastic. It allows the sunlight to enter the FPC cavity while reducing heat loss by trapping thermal radiation within.

Insulation: Insulation is placed on the side boundaries and the underside of the FPC to limit the heat loss to the surrounding environment.

Piping System: Embedded within or attached to the absorber plate, the piping system facilitates the flow of the working fluid; commonly water or a water-glycol mixture, which absorbs and transports the heat generated.

Casing: The external structure houses all components, offering mechanical protection and ensuring the collector's structural integrity.

The operation of a FPC involves the transparent cover permitting the solar radiation, which is subsequently absorbed by the AP. The generated heat is then transferred to the working fluid, which is circulating through the piping system, effectively harnessing solar energy for various applications. A diagrammatic representation of a FPC is given in **Figure 1**.

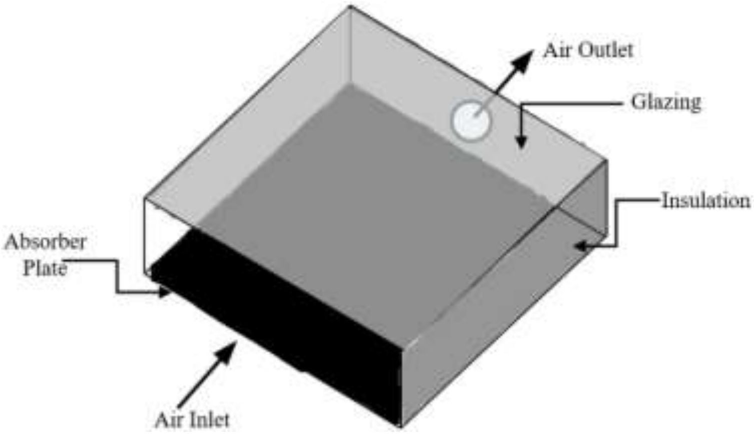

Figure 1. Schematic of FPC

When it comes to the theoretical modelling of FPC the following approach is normally followed:

- Heat transfer from plate to glass:

First the Rayleigh number (Ra) is evaluated with the help of Eqn. 1 [22].

$$Ra = \frac{g \times \beta_v \times \Delta T \times L_c^3}{\alpha \nu} \quad (1)$$

where, β_v is the volumetric expansivity, ΔT is the difference of temperature between mean plate temperature (T_{pm}) and ambient air (T_a), L_c is the specific length of the FPC, α is the thermal diffusivity, and ν is the kinematic viscosity. The thermophysical properties of are estimated by the relations given by the Toyama [23,24] which is evaluated at the mean air temperature $\left(T_v = \frac{T_{in}+T_{pm}}{2}\right)$. The inlet air temperature (T_{in}) is taken same as the T_a. The T_{pm} can be taken as $T_{in} + 20$.

Once Ra is known then Nusselt number (Nu) is evaluated based on the inclination of plate and Ra by using the Eqn. 2 [10].

$$Nu = 1 + 1.44\left[1 - \frac{1708}{Ra\cos\beta}\right]^+ \left(1 - \sin^{1.6}(1.8\beta) \times \frac{1708}{\mathrm{Ra}\cos\beta}\right)\left[\left(\frac{Ra\cos\beta}{5830}\right)^{1/3} - 1\right]^+ \quad (2)$$

Here, $[\quad]^+$ means that only positive value will be taken if negative value comes then it should be taken as zero.

Then the convective heat transfer coefficient (h_{1c}) from the AP and glass is evaluated as follows [10]:

$$h_{1c} = Nu \times \frac{k_a}{L_c} \quad (3)$$

where, k_a is the airs thermal conductivity which is evaluated at T_v.

The radiative heat transfer coefficient (h_{1r}) from plate to glass is evaluated with the help of Eqn. 4 [5].

$$h_{1r} = \varepsilon_{ff}\sigma\left(\frac{(T_{pm}+273.15)^4-(T_g+273.15)^4}{T_{pm}-T_g}\right) \tag{4}$$

where, T_g is the glass temperature, σ is Stefan Boltzmann's constant (5.67×10^{-8} W/m^{2}K^4), and ε_{ff} is the overall emissivity of AP and the glass cover (Eqn. 5) which is obtained by using the individual emissivity of plate (ε_p) and glass (ε_g).

$$\varepsilon_{ff} = \frac{1}{\varepsilon_p} + \frac{1}{\varepsilon_g} \tag{5}$$

The overall heat transfer coefficient (h_1) from plat to glass will be the sum of radiative and convective coefficients which will be obtained as follows [10]:

$$h_1 = h_{1c} + h_{1r} \tag{6}$$

- Heat transport from glass to surrounding ambient:

The convective heat transfer coefficient (h_{2c}) between glass and surrounding atmosphere is obtained with the help of Eqn. 7 as follows [10]:

$$h_{2c} = 2.8 + 3V_a \tag{7}$$

where, V_a is the local wind speed.

The radiative heat transfer coefficient (h_{2r}) between glass and surrounding ambient (sky) is obtained by using Eqn. 8 [10].

$$h_{2r} = \varepsilon_g\sigma\left(\frac{(T_g+273.15)^4-(T_a+273.15)^4}{T_g-T_s}\right) \tag{8}$$

where, T_s is the sky temperature which is obtained as follows [10]:

$$T_s = T_a - 6 \tag{9}$$

The overall heat transfer coefficient (h_2) from glass to ambient will be the sum of radiative and convective coefficients which will be obtained as follows [5]:

$$h_2 = h_{2c} + h_{2r} \tag{10}$$

- Evaluation of top loss coefficients:

Once, h_1 and h_2 are known then the coefficient of top loss (U_T) can be obtained as:

$$U_T = \frac{1}{\left(\frac{1}{h_1}+\frac{1}{h_2}\right)} \tag{11}$$

- Iterative approach to evaluate glass temperature (T_g)

All the Eqn's 4-11 cannot be solved until T_g is unknown. Therefore, the evaluation of T_g is the most important part of calculation. This can be done by using iterative procedure i.e. first some initial guess of T_g is taken then based on this guess U_T is obtained. Then based on this U_T again the $T_{g,new}$ is obtained by using Eqn. 12. If the difference between new and old T_g is less than sum suitable valued (say 10^{-2}) then the $T_{g,new}$ is the final value of T_g otherwise, $T_g = T_{g,new}$ and the process is repeated. Once the iteration gets over then obtain the final value of U_T form the converged vale of T_g.

- Evaluation of bottom, edge, and overall loss coefficients:

The bottom (U_b), edge (U_e), and overall (U_L) loss coefficients are obtained as follows [10]:

$$U_b = k_i/\Delta_i \tag{12}$$

$$U_e = \frac{k_i}{\Delta_i} \times \frac{A_e}{A_c} \tag{13}$$

$$U_L = U_T + U_b + U_e \tag{14}$$

where, k_i is the thermal conductivity of insulation,Δ_i is the insulation thickness (which has been assumed uniform in this manuscript), and A_e & A_c are the edge and collector plate areas, respectively.

- Evaluation of heat removal factor for Collector (F_R):

F_R is the ratio of the actual energy received by the collector to that of the energy take-up by the air. Eqn. 15 shows the mathematical formulation of F_R [25].

$$F_R = \frac{\dot{m}c_p}{A_c U_L}\left(1 - e^{-\left(\frac{A_c U_L F'}{\dot{m}c_p}\right)}\right) \tag{15}$$

where, $\dot{m}$, c_p, and F' are the mass flow rate of air, specific heat at constant pressure of air, and collector efficiency factor, respectively. The F' is evaluated with the help of Eqn. 16.

$$F' = \frac{1}{1 + \frac{U_L}{\left(h_1 + \frac{1}{\left(\frac{1}{h_r} + \frac{1}{h_2}\right)}\right)}} \tag{16}$$

where, h_r is the radiative heat transfer coefficient which is evaluated as follows:

$$h_r = \delta_t\left(\frac{(T_{pm}^2 + T_a^2)\times(T_a + T_{pm})}{\frac{1}{\varepsilon p} + \frac{1}{\varepsilon g} - 1}\right) \tag{17}$$

In Eqn. 17, δ_t is the thickness of absorber plant.

2.2. Python programming approach

- First modules for array and data plotting will be imported.

```
from numpy import *
from pylab import *
```

- Then functions are developed for the evaluation of h_{1c}, h_{2c}, h_{1r}, h_{2r}, Nu, Ra, h_1, h_2, U_T, U_b, U_e, U_T, T_{pm}, T_s, and T_w. **Table. 1** shows the functions.

Table 1. Functions for different parameters

T_s	`# The sky temperature (Tsky)` `Tsky = lambda Ta: Ta-6`

h_{2r}	`# rad. heat trnsfr. coeff. glass --> ambient`
	
	`def h2_rad(Tg,Ts,ϵg,σ):`
	
	`    A = (Tg+273)**4 - (Ts+273)**4`
	
	`    return ϵg*σ*A/(Tg-Ts)`

h_{1r}	`# rad. heat trnsfr. coeff. abs. plate --> glass`
	`def h1_rad(ϵp,ϵg,σ,Tp,Tg):`
	
	`    ϵ_eff = (1/ϵp+1/ϵg)**-1`
	
	`    A = (Tp+273)**4 - (Tg+273)**4`
	
	`    return ϵ_eff*σ*A/(Tp-Tg)`

h_{2c}	`# conv. heat trnsfr. coeff. glass --> ambient air`

```
h2_con = lambda va: 2.8+3.0*va #Va is the wind speed
```

Ra

```
# Rayleigh number (Ra)
Rayleigh = lambda g,β_v,ΔT,Lc,α,ν: g*β_v*ΔT*Lc**3/(α*ν)
```

Nu

```
# Nusselt number for air
def Nusselt(Ra, β):

    A =1708/(Ra*cos(radians(β)))
    B = sin(1.8*(radians(β)))**1.6*1708/(Ra*cos(radians(β)))
    C = (Ra*cos(radians(β))/5830)**(1/3)

    if (1-A)<0:
        x = 0
    else:
        x = 1-A

    if (C-1)<0:
        y = 0
    else:
        y = C01

    return 1+1.44*x*(1-B)+y
```

h_{1c}

```
# Top heat loss coefficient
def h1_con(Nu, ka, Lc):

        return Nu*ka/Lc
```

T_{pm}

```
# Estimation of Tpm (deg C)
Tp_mean = lambda Tin: Tin+20 # Tin: fluid entery Temp.

    return δt*(Tpm**2+Ta**2)*(Tpm+Ta)
```

h_r

```
#radiative coefficient (hr) for F_prime
def h_rad(δt,Tpm,Ta,ϵp,ϵg):

        X = 1/ϵp+1/ϵg-1
```

h_1

```
# total top loss coeff. coll. plate-->glass

h1_Total = lambda h1c,h1r: h1c+h1r
```

h_2

```
# total top loss coeff. glass --> ambient
h2_Total = lambda h2c,h2r: h2c+h2r
```

U_T

```
# effective total top loss coefficient (UT)
UTop = lambda h1,h2: (1/h1+1/h2)**(-1)
```

F'

```
#collector efficiency factor (F_prime)
def Collector_eff_fact(UL,h1,h2,hr):

    A = 1/h2+1/hr

    B = h1+1/A

    return 1/(1+UL/B)
```

F_R

```
# Collector heat removal factor (FR)
def Collector_heat_removal_fact(m_dot, cp, Ac, UL, F_prime):

    X = exp(-Ac*UL*F_prime/(m_dot*cp))

    Y = m_dot * cp/(Ac*UL)

    return Y*(1-X)
```

U_b

```
# Back loss coefficient (Ub)
def Ubottom(ki,Δi):

        return ki/Δi
```

U_e

```
# Edge loss coeff. (Ue)
def Uedge(ki,Ae,Δi,Ac):
```

	`return (ki/Δi)*(Ae/Ac)`
U_L	`# Overall heat loss coeff (UL)` `UL_Total = lambda UT,Ub,Ue: UT+Ub+Ue`
T_w	(see code below)

```
# Function to evaluate Glass temeprature
def T_glass(β,ϵp,ϵg,σ,Tpm,g,β_v,ΔT,Lc,α,ν,ka,Ts,va):

    # Assumed value of Tg
    Tg = 0
    error = 1
    count =0
    while error>1.E-3:

        # Heat transfer: plate --> Glass
        h1r = h1_rad(ϵp,ϵg,σ,Tpm,Tg)
        Ra  = Rayleigh(g,β_v,ΔT,Lc,α,ν)
        Nu  = Nusselt(Ra, β)
        h1c = h1_con(Nu, ka, Lc)
        h1  = h1_Total (h1c,h1r)

        # Heat transfer: Glass --> Ambient
        h2r = h2_rad(Tg,Ts,ϵg,σ)
        h2c = h2_con(va)
        h2 = h2_Total(h2c,h2r)

        # Top heat loss coeff.
        UT = UTop(h1,h2)

        # New value of Tg
        Tg_new = Tpm -(UT/h1)*(Tpm - Ta)

        # Error calculaiton
        error = abs(Tg_new-Tg)

        # Updating Guess
        Tg = Tg_new

        # updating counter
        count = count+1

    return Tg, h1, h2, UT
```

- Thermophysical properties of air are evaluated using the following code snippet

```
# Thermophysical properties of air
cp = 999.2+0.1434*Tv + 1.101E-4*Tv**2-6.7581E-8*Tv**3

ρa = 353.44/(Tv+273)

ka = 0.0244+0.7673E-4*Tv

μa = 1.718E-5+4.62E-8*Tv

α  = ka/ρa

ν  = μa/ρa

β_v = 1/(Tv+273)
```

- Following Flow chart is used to evaluate the impact of plate thickness on the performance of FPC viz. on F_R.

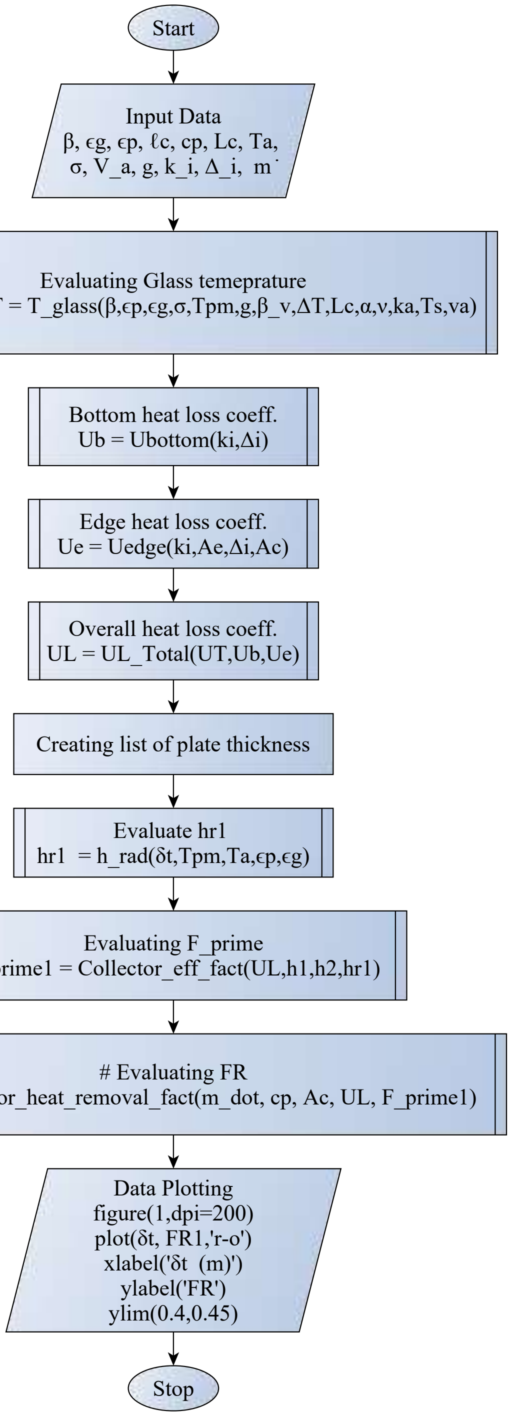

Figure 2. Flowchart to evaluate the impact of plate thickness on the performance of FPC

3. Case study

The input data for the calculations is taken as follows:

Table 2. Input parameters

Category	Parameter	Symbol	Value	Unit
Angle of FPC	Angle	β	30	°
Input Parameters	Emissivity of glass	ϵg	0.88	-
	Emissivity of plate	ϵp	0.95	-

Category	Parameter	Symbol	Value	Unit
	Collector length	ℓc	0.8	m
Geometric Dimensions	Collector breadth	𝕓c	0.6	m
	Box channel height	Lc	0.1	m
Temperatures	Ambient air temperature	Ta	26	°C
Constants	Stefan-Boltzmann const.	σ	5.67×10^{-08}	W/m²-K⁴
	Gravity	g	9.81	m/s²
Insulation	Thermal conductivity	k_i	0.03	W/m-K
	Insulation thickness	Δ_i	0.01	m
Air Flow	Mass flow rate	$\dot{m}$	0.0024	kg/s

Table 2. (*Continued*)

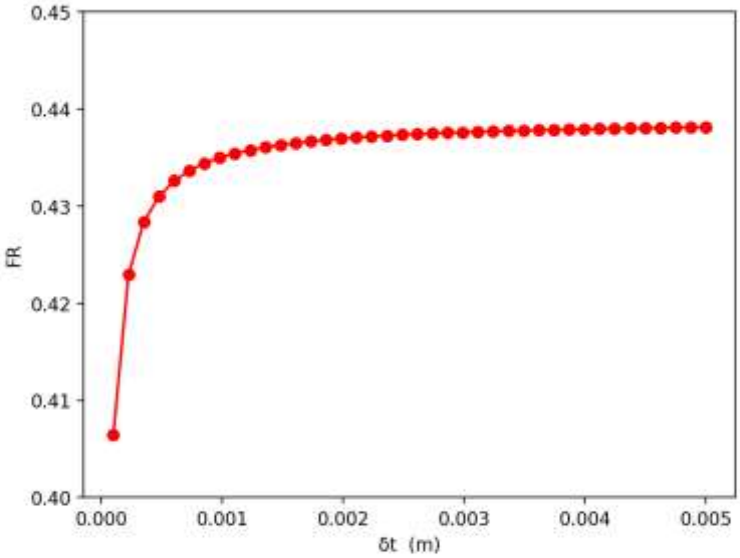

Figure 3. Variation of FR with δ_t

Based on the procedure mentioned in the **Figure 2**, the variation of F_R as a function of δ_t is presented in **Figure 3**. At the lower end of the insulation thickness, F_R starts at a relatively low value. This is because thin insulation allows significant heat losses from the absorber plate to the surroundings, reducing the efficiency of heat transfer to the working fluid. As the insulation thickness increases, the ability of the insulation to minimize these losses improves, resulting in a sharp rise in F_R. This initial region is characterized by a noticeable improvement in thermal performance, as more heat is retained and effectively transferred to the working fluid. However, as the insulation thickness continues to increase, the graph exhibits a flattening trend. This asymptotic behaviour indicates that further increases in insulation thickness led to diminishing returns in the improvement of F_R . Once a critical thickness is reached, most of the heat loss from the absorber plate is already mitigated, and additional insulation has little impact on the collector's thermal performance. Beyond this point, F_R saturates, signifying that the heat transfer efficiency has reached its maximum practical value.

This shows the importance of selecting an optimal insulation thickness during the design of flat plate collectors. While increasing insulation thickness initially enhances the system's efficiency by reducing heat losses, excessive insulation results in unnecessary material costs with minimal gains in performance. This balance between cost and efficiency is critical for designing an effective and economical solar collector system.

4. Conclusion and future work

The use of Python programming in renewable energy research offers many advantages viz. including ease of access, repeatability, and computational efficiency. This article shows the potential of Python programming for simulating the performance of FPC. Through the application of heat transfer concepts and iterative numerical methods, the proposed Python-based model provides a strong outline for understanding the thermal behaviour and energy efficiency in different operating conditions. The study highlights the Python's ability to manage complex computations and present results in a clear visual format. This methodology not only increases the understanding for students and researchers but also will aid in the advancing renewable energy technologies by enabling in-depth performance evaluations.

In future, this model can be further enhanced by incorporating additional features and exploring new scenarios viz. integration of real-time weather data and geographical parameters that will improve the scope of the simulations. The program could also be developed to include other solar collector types, such as evacuated tube collectors which can help in comparative performance analyses. Moreover, Python's increasing ecosystem of machine learning libraries suggestions opportunities for optimizing system design and forecasting long-term performance trends. Improvements like interactive dashboards and graphical user interfaces (GUIs) could further increase the model's reach which will make it more accessible to policymakers and non-technical users. These improvements would strengthen the role of Python programming in driving innovation and dissemination in renewable energy research.

Conflict of interest

The authors declare no conflict of interest

References

1. Kanase-Patil, A. B., Kaldate, A. P., Lokhande, S. D., Panchal, H., Suresh, M., & Priya, V. (2020). A review of artificial intelligence-based optimization techniques for the sizing of integrated renewable energy systems in smart cities. Environmental Technology Reviews, 9(1), 111–136. https://doi.org/10.1080/21622515.2020.1836035
2. Mevada, D., Panchal, H., Nayyar, A., Sharma, K., Manokar, A. M., El-Sebaey, M. S., & Hussien, A. G. (2023). Experimental performance evaluation of solar still with zig-zag shape air-cooled condenser: An energy–exergy analysis approach. Energy Reports, 10, 1198–1210. https://doi.org/10.1016/j.egyr.2023.07.053
3. Yong, Y. K., Lee, H. P., Gan, Z. S., Tan, K. S., Chong, J. W., Lee, C. X., Hashma, S., Hassan, M. J., Latheef, U. A., & Low, W. P. (2024). Catalyzing biomass concrete as new renewable energy source for green building. International Journal of Integrated Engineering, 16(8), 153–161. https://doi.org/10.30880/ijie.2024.16.08.016
4. C. M. S. Kumar et al., "Solar energy: A promising renewable source for meeting energy demand in Indian agriculture applications," Sustain. Energy Technol. Assessments, vol. 55, p. 102905, 2023, doi: https://doi.org/10.1016/j.seta.2022.102905.
5. Panchal, H. N., Patel, M. I., Patel, B., Goswami, R., & Doshi, M. (2011). A comparative analysis of single slope solar still coupled with flat plate collector and passive solar still. International Journal of Research and Reviews in Applied Sciences, 7(2), 111–116.
6. O. Bamisile et al., "A brief review and comparative evaluation of nanofluid application in solar parabolic trough and flat plate collectors," Energy Reports, vol. 8, pp. 156–166, 2022, doi: https://doi.org/10.1016/j.egyr.2022.08.078.
7. Y. Kumar, M. Verma, H. K. Ghritlahre, S. Kumar, P. Verma, and S. Shekhar, "A review of performance improvements in design features of liquid flat-plate solar collector," Int. J. Green Energy, vol. 21, no. 5, pp. 1072–1106, Apr. 2024, doi: 10.1080/15435075.2023.2234035.
8. F. Chichango, L. Cristóvão, J. Nhambiu, F. Cumbe, and G. G. Mavanga, "Literature review of potential materials for the construction of an alternative flat-plate solar collector," Res. Soc. Dev., vol. 13, no. 5, p. e0513545674, May 2024, doi: 10.33448/rsd-v13i5.45674.
9. L. A. Tagliafico, F. Scarpa, and M. De Rosa, "Dynamic thermal models and CFD analysis for flat-plate thermal solar collectors – A review," Renew. Sustain. Energy Rev., vol. 30, pp. 526–537, 2014, doi: https://doi.org/10.1016/j.rser.2013.10.023.
10. L. Geovo, G. D. Ri, R. Kumar, S. K. Verma, J. J. Roberts, and A. Z. Mendiburu, "Theoretical model for flat plate solar collectors operating with nanofluids: Case study for Porto Alegre, Brazil," Energy, vol. 263, p. 125698, 2023, doi: https://doi.org/10.1016/j.energy.2022.125698.

11. A. Seddaoui, M. Z. Dar Ramdane, and R. Noureddine, “Performance investigation of a new designed vacuum flat plate solar water collector: A comparative theoretical study,” Sol. Energy, vol. 231, pp. 936–948, 2022, doi: https://doi.org/10.1016/j.solener.2021.12.038.
12. J. Ranjani, A. Sheela, and K. Pandi Meena, “Combination of NumPy, SciPy and Matplotlib/Pylab-A good alternative methodology to MATLAB-A Comparative analysis,” in Proceedings of 1st International Conference on Innovations in Information and Communication Technology, ICIICT 2019, 2019, pp. 1–5. doi: 10.1109/ICIICT1.2019.8741475.
13. G. Van Rossum and others, “Python Programming Language.,” in USENIX annual technical conference, 2007, pp. 1–36.
14. S. Van Der Walt, S. C. Colbert, and G. Varoquaux, “The NumPy array: A structure for efficient numerical computation,” Comput. Sci. Eng., vol. 13, no. 2, pp. 22–30, 2011, doi: 10.1109/MCSE.2011.37.
15. C. Bauckhage, “NumPy / SciPy Recipes for Data Science: Subset-Constrained Vector Quantization via Mean Discrepancy Minimization,” no. February, pp. 1–4, 2020.
16. P. Dumka, R. Chauhan, A. Singh, G. Singh, and D. Mishra, “Implementation of Buckingham ’ s Pi theorem using Python,” Adv. Eng. Softw., vol. 173, no. July, p. 103232, 2022, doi: 10.1016/j.advengsoft.2022.103232.
17. V. Porcu, “Matplotlib,” in Python for Data Mining Quick Syntax Reference, Berkeley, CA: Apress, 2018, pp. 201–234. doi: 10.1007/978-1-4842-4113-4_10.
18. E. Bisong, “Matplotlib and Seaborn,” in Building Machine Learning and Deep Learning Models on Google Cloud Platform, Berkeley, CA: Apress, 2019, pp. 151–165. doi: 10.1007/978-1-4842-4470-8_12.
19. P. Dumka, A. Deo, K. Gajula, V. Sharma, R. Chauhan, and D. R. Mishra, “Load and Load Duration Curves Using Python,” Int. J. All Res. Educ. Sci. Methods, vol. 10, no. 8, pp. 2127–2134, 2022.
20. W. McKinney, Python for data analysis: Data wrangling with Pandas, NumPy, and IPython. “ O’Reilly Media, Inc.,” 2012.
21. C. Fuhrer, O. Verdier, J. E. Solem, C. Führer, O. Verdier, and J. E. Solem, Scientific Computing with Python. High-performance scientific computing with NumPy, SciPy, and pandas. Packt Publishing Ltd, 2021.
22. T. L. Bergman, A. S. Lavine, F. P. Incropera, and D. P. DeWitt, Fundamentals of Heat and Mass Transfer, 8th ed. Hoboken, NJ: John Wiley & Sons, 2018. doi: 10.1016/j.applthermaleng.2011.03.022.
23. S. Toyama, T. Aragak, H. M. Salah, K. Murase, and M. Sando, “Simulation of a Multieffect Solar Still and the Static Characteristics,” J. Chem. Eng. Japan, vol. 20, no. 5, pp. 473–478, 1987, doi: 10.1252/jcej.20.473.
24. P. Dumka and D. R. Mishra, “Experimental investigation of modified single slope solar still integrated with earth (I) &(II):Energy and exergy analysis,” Energy, vol. 160, pp. 1144–1157, Oct. 2018, doi: 10.1016/j.energy.2018.07.083.
25. A. K. Tiwari and G. N. Tiwari, “Effect of Cover Inclination and Water Depth on Performance of a Solar Still for Indian Climatic Conditions,” J. Sol. Energy Eng., vol. 130, no. 2, p. 024502, 2008, doi: 10.1115/1.2844450.