AGH University of Science and Technology
in Krakow

Automation of Thermal Calculations for Building Enclosures

Python in Engineering and Science

Author:
Szymon Kuczek

Contents

1 Project Aim and Scope

The aim of the project was to create an engineering tool in **Python**, enabling automated analysis of thermal parameters for multi-layer building enclosures. The program performs calculations of the heat transfer coefficient (U) and generates a visualization of the temperature distribution across the wall section ("temperature chart").

A key assumption was **process automation**: from loading data from spreadsheets (Excel), through the symbolic derivation of physical equations, to the generation of graphical reports. The tool allows for rapid verification of enclosure compliance with Technical Conditions (WT 2021).

2 Technologies and Libraries

The project utilized a modern Python technology stack, combining data engineering with symbolic computing:

- **SymPy (Symbolic Python):** Instead of "hard-coding" formulas, this library was used to dynamically create the mathematical model of the enclosure. This allows for calculating walls with any number of layers (n-layers), generating a symbolic total thermal resistance equation R_{tot} in real-time.
- **OpenPyXL:** A library used for integration with the '.xlsx' format commonly used in engineering. The program automatically retrieves material parameters (λ) from an external database and reads enclosure geometry from project files.
- **Matplotlib:** Used for result visualization. It generates professional temperature distribution charts within the enclosure, taking into account actual layer thicknesses and temperature gradients.

3 Mathematical Model

The program bases its calculations on one-dimensional steady-state heat flow.

The total resistance of the enclosure R_{tot} is determined as the sum of surface heat transfer resistances and the conduction resistances of individual layers:

$$R_{tot} = R_{si} + \sum_{i=1}^{n} \frac{d_i}{\lambda_i} + R_{se} \quad \left[\frac{m^2K}{W}\right] \tag{1}$$

Where:

- R_{si}, R_{se} – surface heat transfer resistances (internal/external),
- d_i – thickness of the i-th layer [m],
- λ_i – thermal conductivity coefficient of the i-th layer [W/mK].

The heat transfer coefficient U is calculated from the relationship:

$$U = \frac{1}{R_{tot}} \quad \left[\frac{W}{m^2K}\right] \tag{2}$$

Thanks to the use of the `SymPy` library, this formula is not fixed but is generated dynamically for a list of symbolic variables $d_1 ... d_n$, $\lambda_1 ... \lambda_n$.

4 Case Study - Scenario Analysis

To verify the program's operation, an analysis of three typical construction scenarios was conducted. Boundary conditions were assumed at: $T_{int} = 20°C$, $T_{ext} = -20°C$.

4.1 Scenario 1: Old Tenement House (Uninsulated Wall)

A solid brick wall with a thickness of 50 cm was analyzed.

- **Structure:** Plaster + Solid brick (50cm) + Plaster.
- **Calculation result:** $U \approx 1.25$ W/(m^{2}K).
- **Diagnosis:** The enclosure does not meet contemporary standards. A rapid temperature drop is visible throughout the entire volume of the wall, posing a risk of freezing.

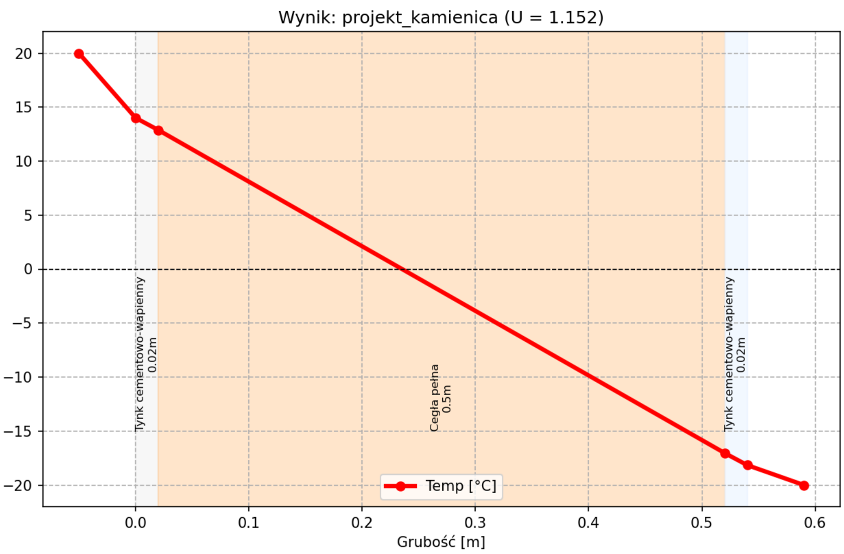

Figure 1: Temperature distribution in an uninsulated wall.

4.2 Scenario 2: Developer Standard

A double-layer wall typical for modern multi-family housing.

- **Structure:** Ceramic block (25cm) + EPS Styrofoam (15cm).
- **Calculation result:** $U \approx 0.21$ W/(m^{2}K).
- **Diagnosis:** Result on the border of the WT 2021 standard ($U_{max} = 0.20$). The chart shows that the main temperature drop (insulation) occurs in the Styrofoam layer, protecting the load-bearing layer from negative temperatures.

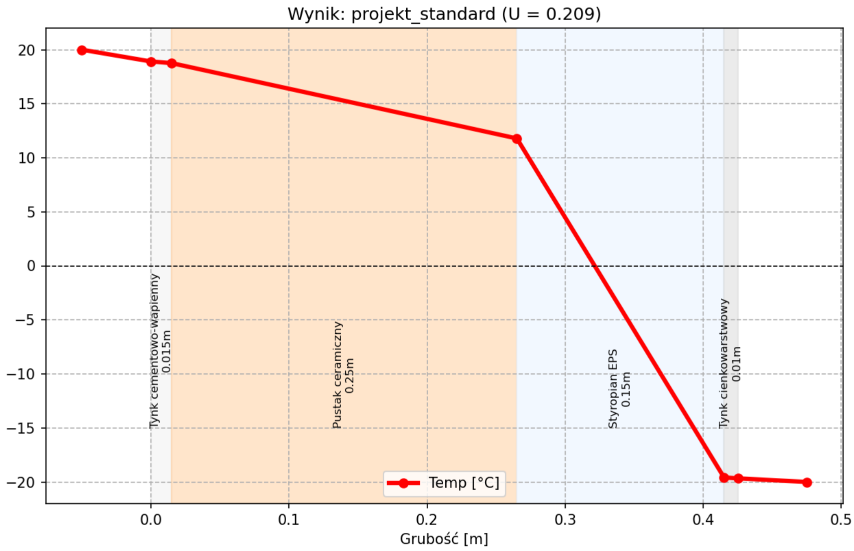

Figure 2: Temperature distribution - Developer Standard.

4.3 Scenario 3: Passive Building

An enclosure with enhanced insulation parameters.

- **Structure:** Reinforced concrete (20cm) + Graphite Styrofoam (25cm, $\lambda = 0.031$).
- **Calculation result:** $U \approx 0.12$ W/(m^{2}K).
- **Diagnosis:** Highly energy-efficient enclosure. The structural layer is located entirely in the positive temperature zone (approx. $19°C$), which eliminates thermal stresses in the structure.

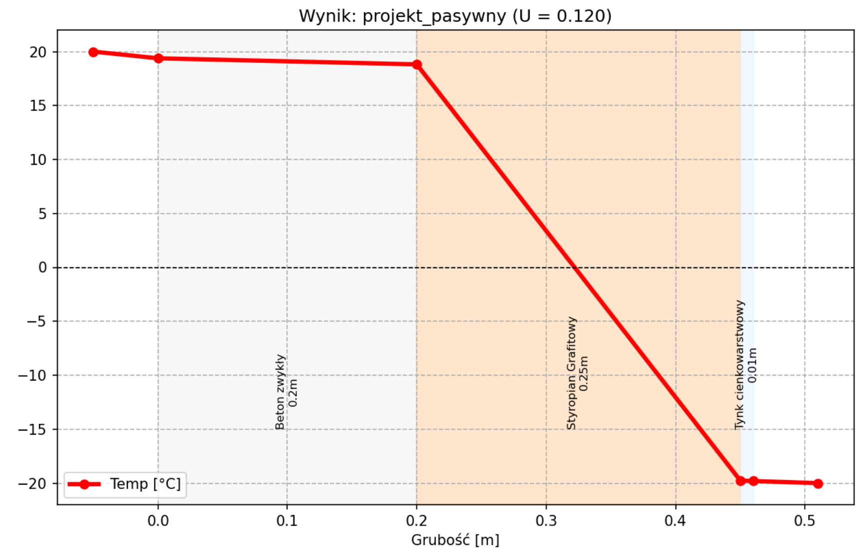

Figure 3: Temperature distribution - Passive Standard.

5 Implementation - Code Snippets

Below is the key code fragment responsible for the symbolic derivation of the thermal resistance equation using the SymPy library.

```python
def oblicz_przegrode_symbolicznie(warstwy_dane):
    # 1. Definition of symbols
    R_tot_sym = sp.symbols('R_tot')
    R_si, R_se = sp.symbols('R_si R_se')

    # Dynamic symbols for n-layers
    n = len(warstwy_dane)
    d_syms = sp.symbols(f'd_1:{n+1}')
    lam_syms = sp.symbols(f'lambda_1:{n+1}')

    # 2. Formula for total resistance (generated automatically)
    suma_oporow = sum(d/l for d, l in zip(d_syms, lam_syms))
    rownanie_R = sp.Eq(R_tot_sym, R_si + suma_oporow + R_se)

    # 3. Substitution of numerical values (Subs)
    wartosci = {R_si: R_SI, R_se: R_SE}
    for i, (nazwa, d_val, lam_val) in enumerate(warstwy_dane):
        wartosci[d_syms[i]] = d_val
        wartosci[lam_syms[i]] = lam_val

    R_wynik = rownanie_R.rhs.subs(wartosci)
    return float(1/R_wynik) # Return U-value
```

Listing 1: Symbolic calculation of enclosure resistance

6 Summary

The developed software constitutes a fully functional prototype of an *Engineering Automation* class tool. The main advantages of the solution are:

1. **Scalability:** Ability to analyze hundreds of project files in a few seconds (batch mode).
2. **Flexibility:** Relying on an external Excel database allows for easy modification of material parameters without interfering with the source code.
3. **Precision:** Application of symbolic calculations eliminates rounding errors at intermediate stages.

This project combines domain knowledge (Building Physics) with programming skills, fitting into the modern trend of digitization in construction (BIM/Automated Design).