

Sprawozdanie

I Wstęp

SYMULACJA USTALONYCH PROCESÓW CIEPLNYCH

ZASADY OGÓLNE

Celem programu było danie możliwości obliczania temperatury w wybranych węzłach badanego przedmiotu. Przykładowym obiektem był pręt ogrzewany na jednym końcu, w którym zachodziła konwekcja czyli rozchodzenie się ciepła. W zależności od tego jaki proces zachodził takie były użyte warunki do obliczenia temperatury. Zjawiska cieplne, które zachodziły można opisać za pomocą równania Fouriera:

$$\operatorname{div}(k(t)\operatorname{grad}(t)) + Q = 0 \text{ lub } \frac{\partial}{\partial x}\left(k_x(t)\frac{\partial t}{\partial x}\right) + \frac{\partial}{\partial y}\left(k_y(t)\frac{\partial t}{\partial y}\right) + \frac{\partial}{\partial z}\left(k_z(t)\frac{\partial t}{\partial z}\right) + Q = 0$$

Prędkość generowania ciepła

Współczynniki przewodzenia ciepła zależne od temperatury

Aby uzyskać wynik za pomocą powyższych wzorów należy znaleźć minimum takiego funkcjonału, dla którego drugie równanie będzie równaniem Eulera. Ten funkcjonał wygląda następująco:

$$J = \int_V \frac{1}{2} \left(k_x(t) \left(\frac{\partial t}{\partial x} \right)^2 + k_y(t) \left(\frac{\partial t}{\partial y} \right)^2 + k_z(t) \left(\frac{\partial t}{\partial z} \right)^2 - 2Qt \right) dV$$

Jeśli współczynniki przewodzenia ciepła są sobie równe $k(t)$ to funkcjonał wynosi

$$J = \int_V \left(\frac{k(t)}{2} \left(\frac{\partial t}{\partial x} \right)^2 + \left(\frac{\partial t}{\partial y} \right)^2 + \left(\frac{\partial t}{\partial z} \right)^2 - Qt \right) dV$$

Funkcja $t(x, y, z)$ musi spełniać wcześniej wspomniane warunki brzegowe. Istnieją trzy możliwości, na powierzchnię jest zadana temperatura t , strumień ciepła q z konwekcji lub promieniowania.

$$\text{Prawo konwekcji } k(t) \left(\frac{\partial t}{\partial x} \alpha_x + \frac{\partial t}{\partial y} \alpha_y + \frac{\partial t}{\partial z} \alpha_z \right) = \alpha_{konw} (t - t_\infty)$$

Temperatura otoczenia

Kosinusy kierunkowe wektora normalnego do powierzchni

Prawo wymiany przez promieniowanie $k(t) \left(\frac{\partial t}{\partial x} \alpha_x + \frac{\partial t}{\partial y} \alpha_y + \frac{\partial t}{\partial z} \alpha_z \right) = \sigma_{rad}(t^4 - t_{\infty}^4)$

Gdy zachodzi konwekcja i promieniowanie to można skorzystać z prawa konwekcji

$$k(t) \left(\frac{\partial t}{\partial x} \alpha_x + \frac{\partial t}{\partial y} \alpha_y + \frac{\partial t}{\partial z} \alpha_z \right) = \alpha(t - t_{\infty})$$

Gdzie α to efektywny współczynnik wymiany ciepła

$$\alpha = \alpha_{konw} + \sigma_{rad}(t^2 + t_{rad}^2)(t + t_{\infty})$$

Nie da się wprowadzić w obecnej formie warunków brzegowych do funkcjonału. Aby to zrobić należy je dodać jako całki wyglądające następująco

$$\int_S \frac{\alpha}{2} (t - t_{\infty})^2 dS + \int_S q t dS$$

powierzchnia

Po dodaniu warunków brzegowych otrzymujemy

$$J = \int_V \left(\frac{k(t)}{2} \left(\frac{\partial t}{\partial x} \right)^2 + \left(\frac{\partial t}{\partial y} \right)^2 + \left(\frac{\partial t}{\partial z} \right)^2 - Q t \right) dV + \int_S \frac{\alpha}{2} (t - t_{\infty})^2 dS + \int_S q t dS$$

Korzystając z następującej zależności temperatury będą przedstawione jako funkcja wartości węzłowych

$$t = \sum_{i=1}^n N_i t_i = \{N\}^T \{t\}$$

Tą zależność uwzględniamy w funkcjonałe wzbogaconym o warunki brzegowe

$$J = \int_V \left(\frac{k}{2} \left(\left\{ \frac{\partial \{N\}}{\partial x} \right\}^T \{t\} \right)^2 + \left(\left\{ \frac{\partial \{N\}}{\partial y} \right\}^T \{t\} \right)^2 + \left(\left\{ \frac{\partial \{N\}}{\partial z} \right\}^T \{t\} \right)^2 - Q \{N\}^T \{t\} \right) dV + \int_S \frac{\alpha}{2} (\{N\}^T \{t\} - t_{\infty})^2 dS + \int_S q \{N\}^T \{t\} dS$$

Należy wykonać minimalizację funkcjonału czyli obliczyć pochodne cząstkowe względem temperatury. Otrzymamy

$$\frac{\partial J}{\partial \{t\}} = \int_V \left(k \left\{ \frac{\partial \{N\}}{\partial x} \right\} \left\{ \frac{\partial \{N\}}{\partial x} \right\}^T + \left\{ \frac{\partial \{N\}}{\partial y} \right\} \left\{ \frac{\partial \{N\}}{\partial y} \right\}^T + \left\{ \frac{\partial \{N\}}{\partial z} \right\} \left\{ \frac{\partial \{N\}}{\partial z} \right\}^T \right) \{t\} - Q \{N\} dV + \int_S \alpha (\{N\}^T \{t\} - t_{\infty}) \{N\} dS + \int_S q \{N\} dS = 0$$

Można zapisać jako macierz

$$[H] \{t\} + \{P\} = 0$$

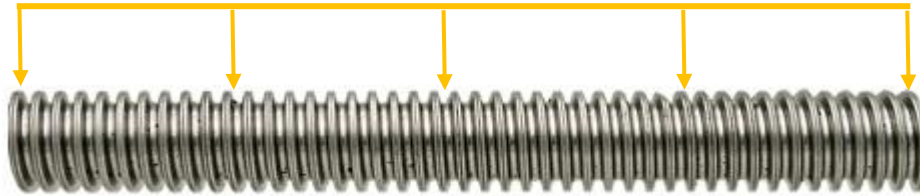
Pojemność cieplna Temperatura Macierz obciążeń

$$[H] = \int_V \left(k(t) \left(\left\{ \frac{\partial \{N\}}{\partial x} \right\} \left\{ \frac{\partial \{N\}}{\partial x} \right\}^T + \left\{ \frac{\partial \{N\}}{\partial y} \right\} \left\{ \frac{\partial \{N\}}{\partial y} \right\}^T + \left\{ \frac{\partial \{N\}}{\partial z} \right\} \left\{ \frac{\partial \{N\}}{\partial z} \right\}^T \right) dV + \int_S \alpha \{N\} \{N\}^T dS$$

$$\{P\} = - \int_S \alpha \{N\} t_{\infty} dS - \int_V Q \{N\} dV + \int_S q \{N\} dS$$

II Excel

W programie Excel zadaniem było obliczenie temperatur i wartości funkcjonałów za pomocą dodatku SOLVER. Poprawność obliczeń była sprawdzana poprzez obliczenie temperatury dla 2 i 4 elementów. Temperatuury w odpowiednich węzłach powinny być takie same.



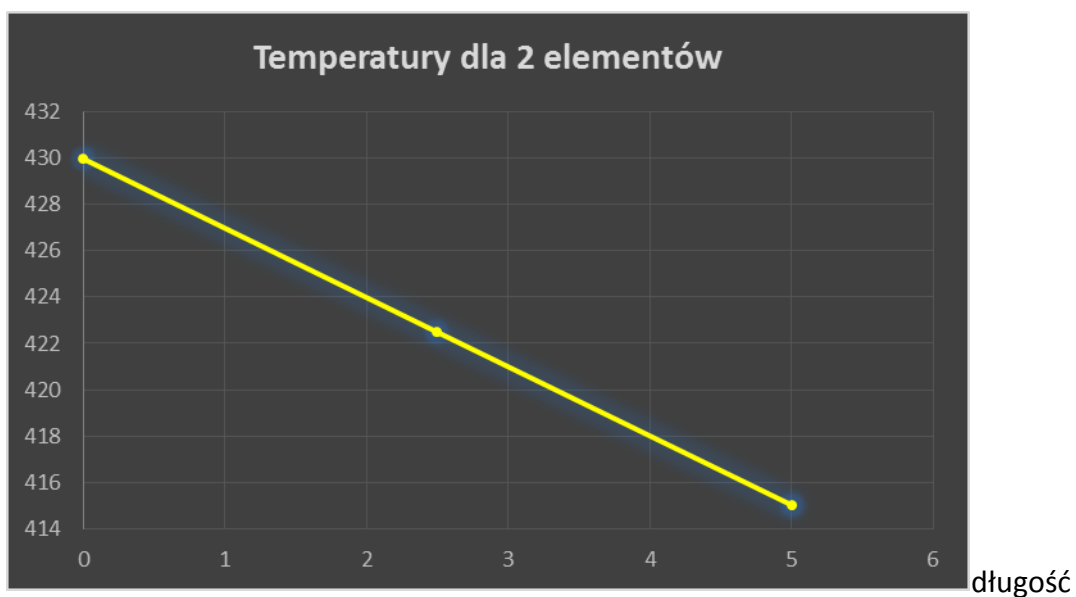
Dane

α	20
qS	-300
t_{∞}	400
s	2
k	50
l	2,5
C	40

Szukane dla 2 elementów

t_1	430
t_2	422,5
t_3	415

J1	1125,017
J2	1125,002
J11	-129000,009
J12	2249,990
J	-124500



J1 węzeł	$\frac{C}{2}(t_1^2 + 2t_1t_2 + t_2^2)$
J2 węzeł	$\frac{C}{2}(t_2^2 + 2t_2t_3 + t_3^2)$
J11 warunek związany z strumieniem	qSt_1
J12 warunek związany z konwekcją	$\frac{\alpha S}{2}(t_3^2 + 2t_3t_\infty + t_\infty^2)$
J suma	$J1+J2+J3+J4$

Dane dla 4 elementów

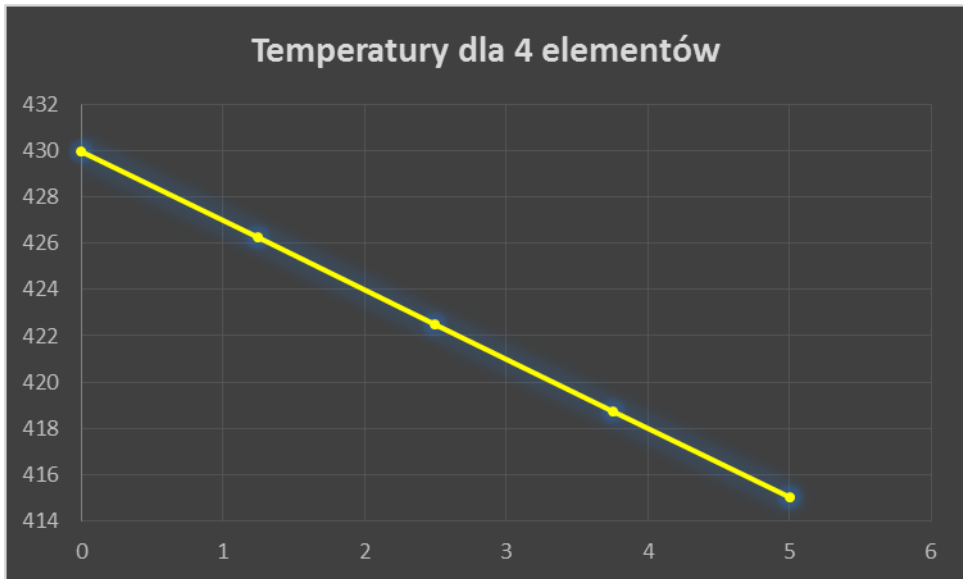
α	20
qS	-300
t_∞	400
s	2
k	50
l	1,25
C	80

Szukane

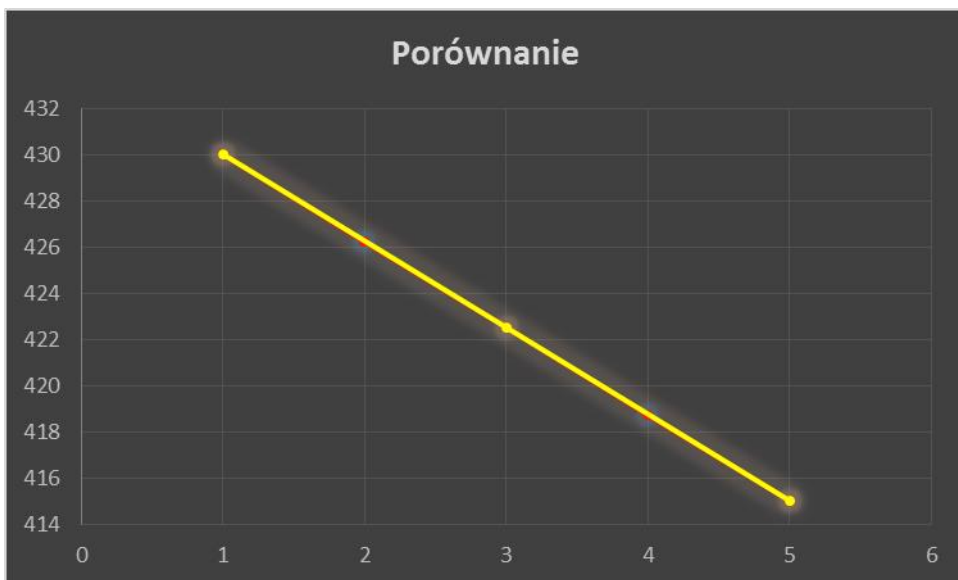
t_1	430
t_2	426,25
t_3	422,50
t_4	418,75
t_5	415

J1	562,495
J2	562,498
J3	562,505
J4	562,505
J11	-128999,962
J12	2249,958
J	-124500

J1 węzeł	$\frac{C}{2}(t_1^2 + 2t_1t_2 + t_2^2)$
J2 węzeł	$\frac{C}{2}(t_2^2 + 2t_2t_3 + t_3^2)$
J3 węzeł	$\frac{C}{2}(t_3^2 + 2t_3t_4 + t_4^2)$
J4 węzeł	$\frac{C}{2}(t_4^2 + 2t_4t_5 + t_5^2)$
J11 warunek związany z strumieniem	qSt_1
J12 warunek związany z konwekcją	$\frac{\alpha S}{2}(t_4^2 + 2t_4t_\infty + t_\infty^2)$
J suma	$J1+J2+J3+J4+J5+J6$



długość



nr indeksu

Porównując wyniki można zauważyć, że temperatury w pierwszym, środkowym i ostatnim węźle są takie same jak im odpowiadające w drugim przypadku. Na wykresie porównawczym widać, że jedna seria całkowicie zasłoniła drugą (żółta i czerwona seria). Wyniki są zgodne w całym modelu. Oznacza to, że temperatury zostały poprawnie wyliczone. Gdy korzystamy z większej ilości punktów możemy zbadać jak zmienia się temperatura. Bardzo przydatne to będzie gdy model będzie się składał z różnych materiałów, które w różnym stopniu przepuszczają ciepło. W jednorodnym modelu zmiana temperatury jest stała.

III Własne zagadnienie

Wyprowadzenie metody

WYZNACZANIE USTALONEGO POLA TEMPERATURY W PRĘCIE

Zakładamy, że wymiana ciepła zachodzi tylko przez końce pręta, przez ogrzewanie i konwekcję. Do jednego końca dokładamy strumień ciepła q , na drugim zachodzi wspomniana konwekcja. Aby wyznaczyć ustalone pole temperatur w pręcie najpierw należy rozpatrzyć równanie różniczkowe Fouriera, które przedstawia się następująco

$$k \frac{d^2}{dx^2} = 0$$

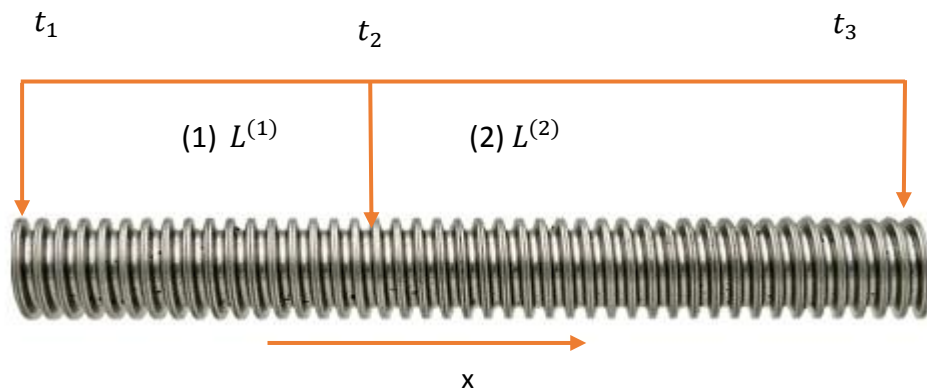
A warunki brzegowe

$$k \frac{dt}{dx} + g = 0 \quad \text{Gdy koniec jest ogrzewany}$$

$$k \frac{dt}{dx} + \alpha (t - t_\infty) = 0 \quad \text{Gdy zachodzi konwekcja}$$

Dla pręta funkcjonal będzie wyglądał tak

$$J = \int_V \frac{1}{2} \left(k \left(\frac{\partial t}{\partial x} \right)^2 \right) dV + \int_S \frac{\alpha}{2} (t - t_\infty)^2 dS + \int_S t q dS$$



Minimalizacja funkcjonatu.

Temperatury wynoszą kolejno

$$t^{(1)} = N_1^{(1)} t_1 + N_2^{(1)} t_2$$

$$t^{(2)} = N_2^{(2)} t_2 + N_3^{(2)} t_3$$

$$N_1^{(1)} = \frac{x_2 - x}{L^{(1)}}$$

$$N_2^{(1)} = \frac{x - x_1}{L^{(1)}}$$

$$N_2^{(2)} = \frac{x_3 - x}{L^{(2)}}$$

$$N_2^{(1)} = \frac{x - x_2}{L^{(2)}}$$

Całki objętościowe

$$\int_{S_1} q t dS = q t_1 S_1$$

$$\int_{S_3} \frac{\alpha}{2} (t - t_\infty)^2 dS = \frac{\alpha S_3}{2} (t_3^2 - 2t_3 t_\infty + t_\infty^2)$$

Należy wyznaczyć pochodne temperatury względem x

$$\frac{dt^{(1)}}{dx} = \frac{(-t_1 + t_2)}{L^{(1)}}$$

$$\frac{dt^{(2)}}{dx} = \frac{(-t_2 + t_3)}{L^{(2)}}$$

Po przekształceniach otrzymano

$$\int_V \frac{k}{2} \left(\frac{dt}{dX} \right)^2 dV = \frac{k^{(1)} S^{(1)}}{2L^{(1)}} (t_2 - t_1)^2 + \frac{k^{(2)} S^{(2)}}{2L^{(2)}} (t_3 - t_2)^2$$

Należy uwzględnić, że elementy mogą być z różnych materiałów i będą mieć różne współczynniki przewodzenia ciepła. Po dodaniu powyższych zależności do siebie otrzymujemy funkcjonal w postaci funkcji węzłowych wartości temperatury

$$J = \frac{C^{(1)}}{2} (t_1^2 - 2t_1 t_2 + t_2^2) + \frac{C^{(2)}}{2} (t_2^2 - 2t_2 t_3 + t_3^2) + q S_1 t_1 + \frac{\alpha S_3}{2} (t_3^2 - 2t_3 t_\infty + t_\infty^2)$$

$$k = \frac{C}{2}$$

$$C = \frac{Sk}{L}$$

Teraz można zastosować bezpośrednią minimalizację funkcji przez wybranie odpowiednich temperatur w węźle lub skorzystać z warunku ekstremum funkcji. Druga metoda polega na zróżniczkowaniu wzoru względem węzłowych zmiennych i przyrównania do zera. Układ jako macierz wygląda następująco

$$\begin{bmatrix} C^{(1)} & -C^{(1)} & 0 \\ -C^{(1)} & C^{(1)} + C^{(2)} & -C^{(2)} \\ 0 & -C^{(2)} & C^{(2)} + \alpha S t_3 \end{bmatrix} \begin{Bmatrix} t_1 \\ t_2 \\ t_3 \end{Bmatrix} + \begin{Bmatrix} qS \\ 0 \\ -\alpha S t_\infty \end{Bmatrix} = 0$$

$$\text{lub } [H]\{t\} + \{P\} = 0$$

$$[H] = \int_V k \left\{ \frac{\partial \{N\}}{\partial x} \right\} \left\{ \frac{\partial \{N\}}{\partial x} \right\}^T dV + \int_S \alpha \{N\} \{N\}^T dS$$

$$\{P\} = - \int_S \alpha \{N\} t_\infty dS + \int_S q \{N\} dS$$

Funkcja kształtu

$$\{N\} = \begin{Bmatrix} N_i \\ N_j \end{Bmatrix} = \begin{Bmatrix} \frac{x_j - x}{L} \\ \frac{x - x_i}{L} \end{Bmatrix}$$

Następnie wykonujemy transpozycję, obliczamy pochodną cząstkową po x i kolejny raz transponujemy. Otrzymane przekształcenia wykorzystujemy w wzorze i po całkowaniu mamy

$$[H] = k \begin{Bmatrix} \frac{-1}{L} \\ \frac{1}{L} \end{Bmatrix} \begin{Bmatrix} \frac{-1}{L} & \frac{1}{L} \end{Bmatrix} + \alpha \begin{Bmatrix} N_i \\ N_j \end{Bmatrix} \begin{Bmatrix} N_i & N_j \end{Bmatrix} S$$

Druga macierz będzie mieć postać

$$\{P\} = - \int_S \alpha \begin{Bmatrix} N_i \\ N_j \end{Bmatrix} t_\infty dS + \int_S q \begin{Bmatrix} N_i \\ N_j \end{Bmatrix} dS$$

Po uwzględnieniu warunków należy stworzyć globalne macierze z powyższych. Numer węzła określa gdzie dana wartość będzie się znajdować. Ostatnim krokiem jest rozwiązanie układu gdzie niewiadomą jest temperatura.

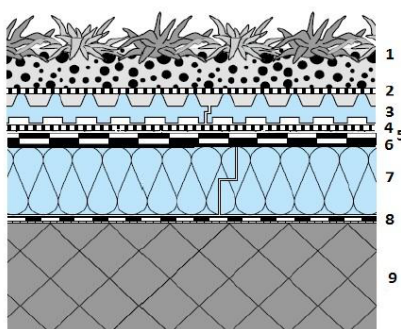
Parametry dachów zielonych

swisspor informacje techniczne

Dach zielony ekstensywny

swisspor GREEN System ekstensywny o układzie tradycyjnym

swisspor BITERM BT 150



Warstwy	Grubość [mm]	Współczynnik przewodzenia ciepła λ [W/m·K]
1. substrat + roślinność *	80	-
2. geowłóknina filtracyjna	-	-
3. swisspor AKUMULATOR wody	53	0,058
4. warstwa ochronna włóknina 285 g/m ²	0,25	-
5. swisspor BIKUTOP EP4 WF flam	4,0	0,18
6. swisspor BIKUTOP podkładowa 200	4,0	0,18
7. swisspor BITERM BT 150	dowolna	0,035
8. swisspor BIKUTOP V24	2,4	0,18
9. konstrukcja zagruntowana gruntem swisspor PRIMER	200,0	1,7

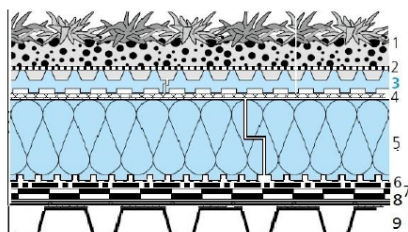
* substrat do zazielenienia ekstensywnego, roślinność typu SEDUM

swisspor informacje techniczne

Dach zielony ekstensywny

swisspor GREEN System ekstensywny o układzie odwróconym

swisspor DREN dach



Warstwy	Grubość [mm]	Współczynnik przewodzenia ciepła λ [W/m·K]
1. substrat + roślinność *	80	-
2. geowłóknina filtracyjna	-	-
3. swisspor AKUMULATOR wody	53	0,058
4. mata dyfuzyjna DELTA VENT S PLUS	-	-
5. styropian DREN dach	dowolna	0,038
6. warstwa separacyjna folia PE 02	0,2	-
7. swisspor BIKUTOP EP4 WF flam	4,0	0,18
8. swisspor BIKUTOP podkładowa 200	4,0	0,18
9. blacha trapezowa zagruntowana gruntem swisspor PRIMER	1,0	50

* substrat do zazielenienia ekstensywnego, roślinność typu SEDUM

Tematem własnego zadania jest porównanie dwóch typów dachów zielonych ekstensywnych, gdy świeci na nie słońce. Różnią się one warstwami, z których są zbudowane. Aby obliczyć temperatury w węzłach należy znać grubość warstwy, pole powierzchni badanej, jej współczynnik przewodzenia ciepła, strumień ciepła, współczynnik konwekcji i temperaturę otoczenia (pomieszczenia). Grubość podana jako „dowolna” ustalono na 0.08m.



Wyniki dla modelu-dach zielony ekstensywny o układzie tradycyjnym

Gt 1	300,9249051 K	27,92°C
Gt 2	299,6455948 K	26,64°C
Gt 3	299,6144837 K	26,61°C
Gt 4	299,5833725 K	26,58°C
Gt 5	296,3833725 K	23,38°C
Gt 6	296,3647059 K	23,36°C
Gt 7	296,2 K	23,20°C

Wyniki dla modelu-dach zielony ekstensywny o układzie odwróconym

Gt 1	300,4889290 K	27,49°C
Gt 2	299,2096187 K	26,21°C
Gt 3	296,2622502 K	23,26°C
Gt 4	296,2311391 K	23,23°C
Gt 5	296,2000280 K	23,20°C
Gt 6	296,2 K	23,20°C

Wnioski

Porównując te dwa modele dachów widać na pierwszy rzut oka różnicę w temperaturach. Na początku jest ona znikoma i warstwy mają około 27°C, w I cztery pierwsze warstwy pochłaniają ciepło, przekazują je dalej i utrzymują temperaturę na poziomie około 26°C. W II pierwsze odstają od średniej temperatury a kolejne warstwy są chłodniejsze o 3°C. Taki sam spadek temperatury można zaobserwować w I, ale dopiero po minięciu czterech warstw. Za zatrzymanie dalszego ogrzewania poddasza przez słońce odpowiada warstwa występująca w obu przypadkach o nazwie BIKUTOP różnego typu dla tych modeli. Następne warstwy utrzymują temperaturę i ostatnie zbliżają się do określonej komfortowej temperatury 23°C. Modele się znacznie różnią ostatnią warstwą. W I jest to konstrukcja zagruntowana gruntem o grubości 200mm, a w drugim blacha trapezowa zagruntowana gruntem 1mm. Porównując je można dojść do wniosku, że nie zawsze im grubsza izolacja to tym lepsza. II model ostatnią warstwę ma 200 razy cieńszą, a temperaturę osiągnięto taką samą już w przedostatniej warstwie. Obserwując zmiany ciepła, ilość istotnych warstw i grubość można stwierdzić, że lepiej wypada model dachu zielonego ekstensywnego o układzie odwróconym. Jego układ warstw oraz moment, w którym badana temperatura była bliska 23°C przekonuje, że lepszy jest cieńszy dach zbudowany z tych warstw o danej grubości i współczynniku.

IV Program

```
#include <iostream>
#include <cstdlib>

using namespace std;

struct wezel
{
    double x; //wspolrzeczna
    int stan; //0 brak, 1 poczatek grzany, 2 koniec
    int ID; //indeks elementu w siatce
};

struct element
{
    double L; //dlugosc odcinka
    double S; //pole
    double k; //wspolczynnik
    double alfa; //warunek 2
    double q; // strumien warunek 1
    double tot; //temperatura otoczenia
    wezel wezly[2]; //wezly dwa
    double H[2][2]; //macierz pojemnosci cieplnej
    double P[2]; // macierz obciazen

    void oblicz_macierze_lokalne()
    {
        //macierz H
        double C = (S*k) / L;
        for (int i = 0; i < 2; i++)
        {
            for (int j = 0; j < 2; j++)
            {
                H[i][j] = C;
                if (i != j)
                {
                    H[i][j] *= -1;
                }
            }
        }
        //macierz P
        for (int i = 0; i < 2; i++)
            P[i] = 0;

        for (int i = 0; i < 2; i++)
        {
            if (wezly[i].stan == 1)
            {
                P[0] = q*S;
            }
            else if (wezly[i].stan == 2)
            {
                P[1] = (-alfa)*tot*S;
                H[1][1] += alfa*S;
            }
        }
    }
};
```

```

struct siatka
{
    int ne; //liczba elementow
    int nh; //liczba wierzchołkow
    double **GH; //globalna
    double *GP;
    double *Gt; //wynik
    element *elementy;

    void stworz_macierze()
    {
        //macierz GH
        GH = new double*[nh];
        for (int i = 0; i < nh; i++)
        {
            GH[i] = new double[nh];
        }

        for (int i = 0; i < nh; i++)
        {
            for (int j = 0; j < nh; j++)
            {
                GH[i][j] = 0;
            }
        }

        //macierz GP Gt
        GP = new double[nh];
        Gt = new double[nh];
        for (int i = 0; i < nh; i++)
        {
            GP[i] = 0;
            Gt[i] = 0;
        }

        //sumowanie macierz
        //GH
        for (int i = 0; i < ne; i++)
        {
            //ustalanie miejsca w duzej macierzy
            GH[elementy[i].wezly[0].ID - 1][elementy[i].wezly[0].ID - 1] +=
elementy[i].H[0][0];
            GH[elementy[i].wezly[1].ID - 1][elementy[i].wezly[1].ID - 1] +=
elementy[i].H[1][1];
            GH[elementy[i].wezly[1].ID - 1][elementy[i].wezly[0].ID - 1] +=
elementy[i].H[1][0];
            GH[elementy[i].wezly[0].ID - 1][elementy[i].wezly[1].ID - 1] +=
elementy[i].H[0][1];
        }

        //GP
        for (int i = 0; i < ne; i++)
        {
            GP[elementy[i].wezly[0].ID - 1] += elementy[i].P[0];
            GP[elementy[i].wezly[1].ID - 1] += elementy[i].P[1];
        }
        cout.precision(15);
        cout << "Macierz GH" << endl;
        for (int i = 0; i < nh; i++)
        {

```

```

        for (int j = 0; j < nh; j++)
        {
            cout << GH[i][j] << " ";
        }
        cout << endl;
    }
    cout << endl;
    cout << "Macierz GP" << endl;
    for (int i = 0; i < nh; i++)
    {
        cout << GP[i] << " ";

        cout << endl;
    }
    cout << endl;
}

void gauss()
{
    for (int i = 0; i < nh - 1; i++)
    {
        for (int j = i + 1; j < nh; j++)
        {
            double pom = -GH[j][i] / GH[i][i];

            for (int k = i; k <= nh; k++)
            {
                GH[j][k] += pom* GH[i][k];
            }
            GP[j] += pom*GP[i];
        }
    }

    Gt[nh] = GP[nh - 1] / GH[nh - 1][nh - 1];

    for (int i = nh - 1; i >= 0; i--)
    {
        double pom = 0;
        for (int j = i + 1; j < nh; j++)
        {
            pom += GH[i][j] * Gt[j];
        }
        Gt[i] = (GP[i] - pom) / GH[i][i];
    }
    for (int i = 0; i < nh; i++)
    {
        Gt[i] *= -1;
    }
    cout << "Macierz Gt" << endl;
    for (int i = 0; i < nh; i++)
    {
        cout << "Gt " << i + 1 << " = " << Gt[i] << endl;
    }
}

};

```

```

int main()
{
    int ilosc_elementow = 0;
    int ilosc_wierzchołkow = 0;

    cout << "Wyznaczanie ustalonego pola temperatury w precie" << endl;
    cout << "Podaj ilosc elementow" << endl;
    cin >> ilosc_elementow;
    ilosc_wierzchołkow = ilosc_elementow + 1;

    element * elementyG = new element[ilosc_elementow];
    wezel * wezlyG = new wezel[ilosc_wierzchołkow];

    cout << "Podaj dla kazdego wierzchołka wspolrzadna x, stan i ID" << endl;
    for (int i = 0; i < ilosc_wierzchołkow; i++)
    {
        cin >> wezlyG[i].x;
        cin >> wezlyG[i].stan;
        cin >> wezlyG[i].ID;
    }
    cout << endl;

    cout << "Podaj dla kazdego elementu dlugosc odcinka L, pole S,
wspolczynnik k, alfa(jesli nie dotyczy to 1, warunek 2(koniec)), strumien q(jesli nie
dotyczy 1, warunek 1(grzanie)), temperatura otoczenia too i numery jego wezlow" <<
endl;
    for (int i = 0; i < ilosc_elementow; i++)
    {
        int id_wezel1 = 0;
        int id_wezel2 = 0;
        cin >> elementyG[i].L;
        cin >> elementyG[i].S;
        cin >> elementyG[i].k;
        cin >> elementyG[i].alfa;
        cin >> elementyG[i].q;
        cin >> elementyG[i].tot;
        cin >> id_wezel1;
        cin >> id_wezel2;
        for (int j = 0; j < ilosc_wierzchołkow; j++)
        {
            if (wezlyG[j].ID == id_wezel1)
            {
                elementyG[i].wezly[0] = wezlyG[j];
            }
            else if (wezlyG[j].ID == id_wezel2)
            {
                elementyG[i].wezly[1] = wezlyG[j];
            }
        }
        elementyG[i].oblicz_macierze_lokalne();
    }
    cout << endl;
    siatka siatka = { ilosc_elementow, ilosc_wierzchołkow };

    siatka.elementy = elementyG;
    siatka.stworz_macierze();
    siatka.gauss();

    system("PAUSE");
    return 0;
}

```