

Symulacja niestabilnych procesów cieplnych

I Ogólny wstęp teoretyczny

Kolejnym tematem projektu z MES jest symulacja niestabilnych procesów cieplnych. Różni się od procesu stacjonarnego tym, że uwzględniany jest upływ czasu i zmiany jakie to zjawisko niesie. Przez dodanie zależności czasu równanie Fouriera wygląda następująco:

$$\text{div}(k(t) \text{grad}(t)) + Q = c \rho \frac{\partial t}{\partial \tau}$$

Lub gdy występują anizotropowe własności cieplne:

$$\frac{\partial}{\partial x} \left(k_x(t) \frac{\partial t}{\partial x} \right) + \frac{\partial}{\partial y} \left(k_y(t) \frac{\partial t}{\partial y} \right) + \frac{\partial}{\partial z} \left(k_z(t) \frac{\partial t}{\partial z} \right) + (Q - c \rho \frac{\partial t}{\partial \tau}) = 0$$

Można dokonać kilka założeń takich jak uznanie pochodnych temperatury w danej chwili jako funkcje współrzędnych oraz zastąpienie całego ostatniego wyrażenia parametrem Q. Wtedy rozwiązanie równania wykonuje się tak samo jak w ustalonym procesie cieplnym omawianym w poprzednim sprawozdaniu. Otrzymujemy wtedy:

$$[H] \{t\} + [C] \frac{\partial}{\partial \tau} \{t\} + \{P\} = 0$$

$$[C] = \int_V c \rho \{N\} \{N\}^T dV$$

Wartości temperatury w węzłach zależą od czasu. Gdy założymy, że wektor $\{t_0\}$ przedstawia dane w chwili 0 to będzie on wynosił:

$$\{t\} = \{N_0, N_1\} \begin{Bmatrix} \{t_0\} \\ \{t_1\} \end{Bmatrix}$$

$\{N_0\}\{N_1\}$ to funkcje kształtu zależne od czasu

$\{t_1\}$ temperatury węzłowe po czasie ΔT

Zakładamy, że dla małych kroków czasowych zależność temperatur od czasu jest liniowa to funkcje kształtu przyjmują wzór:

$$N_0 = \frac{\Delta T - T}{\Delta T} \quad N_1 = \frac{T}{\Delta T}$$

Gdy dodamy zależność $\{t_0\}$ to pochodne wyglądają następująco:

$$\frac{\partial \{t\}}{\partial T} = \left\{ \frac{\partial N_0}{\partial T}, \frac{\partial N_1}{\partial T} \right\} \begin{Bmatrix} \{t_0\} \\ \{t_1\} \end{Bmatrix} = \frac{1}{\Delta T} \begin{Bmatrix} -1, 1 \end{Bmatrix} \begin{Bmatrix} \{t_0\} \\ \{t_1\} \end{Bmatrix} = \frac{\{t_1\} - \{t_0\}}{\Delta T}$$

Przyjeliśmy, że funkcje kształtu są liniowe to wynik jest zbieżny z wynikiem, który otrzymalibyśmy za pomocą metody różnic skończonych. Tu pojawia się kilka dróg rozwiązania układu równań za pomocą interpolacji, w zależności od chwili, w której będziemy rozpatrywać wektor $\{t\}$ i macierz $[H]$.

1. Przyjmujemy że $\{t\} = \{t_0\}$ i wzór wygląda tak:

$$[H] \{t_0\} + [C] \frac{\{t_1\} - \{t_0\}}{\Delta T} + \{p\} = 0$$

Otrzymaliśmy jawny schemat wyznaczania temperatury $\{t_1\}$ w czasie T:

$$\{t_1\} = \{t_0\} - \frac{\Delta T}{[C]} ([H] \{t_0\} + \{p\})$$

Stosowanie tego schematu jest ograniczone, ponieważ ma słabą stabilność przy różnych ΔT .

2. Przyjmujemy $\{t\} = \{t_1\}$ i wzór jest opisany następująco:

$$[H] \{t_1\} + [C] \frac{\{t_1\} - \{t_0\}}{\Delta T} + \{p\} = 0$$

Jest to niejawny schemat wyznaczania temperatury $\{t_1\}$:

$$\left([H] + \frac{[C]}{\Delta T} \right) \{t_1\} - \left(\frac{[C]}{\Delta T} \right) \{t_0\} + \{p\} = 0$$

Musimy rozwiązać ten układ równań.

3. Przyjmujemy $\{t\}$ jako średnią z temperatur w chwili $T = 0$

$$\text{w } T = \Delta T : \{t\} = \frac{1}{2} (\{t_0\} + \{t_1\})$$

Należy rozwiązać dany układ równań:

$$\{p\} = \frac{1}{2} (\{p_1\} + \{p_0\})$$

$$[H] \frac{1}{2} (\{t_0\} + \{t_1\}) + [C] \frac{\{t_1\} - \{t_0\}}{\Delta T} + \{p\}^* = 0$$

Drugie równanie zostaje przekształcone do postaci:

$$\left(\frac{[H]}{2} + \frac{[C]}{\Delta T} \right) \{t_1\} + \left(\frac{[H]}{2} - \frac{[C]}{\Delta T} \right) \{t_0\} + \{p\}^* = 0$$

$$\left([H] + \frac{2[C]}{\Delta T} \right) \{t_1\} + \left([H] - \frac{2[C]}{\Delta T} \right) \{t_0\} + 2\{p\}^* = 0$$

4. Czas można uwzględnić również przez metodę reszt ważonych. Znamy wektor temperatur węzłowych $\{t_0\}$ to aby scałkować wyrażenie

$$\frac{\partial \{t\}}{\partial T} = \left[\frac{\partial N_0}{\partial T}, \frac{\partial N_1}{\partial T} \right] \begin{Bmatrix} \{t_0\} \\ \{t_1\} \end{Bmatrix} = \frac{1}{\Delta T} \begin{Bmatrix} -1, 1 \end{Bmatrix} \begin{Bmatrix} \{t_0\} \\ \{t_1\} \end{Bmatrix} = \frac{\{t_1\} - \{t_0\}}{\Delta T}$$

względem czasu wprowadzamy jedną ważoną rezydualną $N_1 = \frac{T}{\Delta T}$ z zależnością:

$$\int_0^{\Delta T} \frac{T}{\Delta T} \left([H] \{N_0, N_1\} \begin{Bmatrix} \{t_0\} \\ \{t_1\} \end{Bmatrix} + [C] \left[\frac{\partial N_0}{\partial T}, \frac{\partial N_1}{\partial T} \right] \begin{Bmatrix} \{t_0\} \\ \{t_1\} \end{Bmatrix} + \{p\} \right) dT = 0$$

Uzupełniając wzór o funkcje kształtu otrzymujemy:

$$\int_0^{\Delta T} \frac{T}{\Delta T} \left([H] \left(\frac{\Delta T - T}{\Delta T} \{t_0\} + \frac{T}{\Delta T} \{t_1\} \right) + \frac{[C]}{\Delta T} (-\{t_0\} + \{t_1\}) + \{p\} \right) dT = 0$$

Po obliczeniu całki i przekształceniu mamy:

$$\left(2[H] + \frac{3}{\Delta T} [C] \right) \{t_1\} + \left([H] - \frac{3}{\Delta T} [C] \right) \{t_0\} + 3\{p\} = 0$$

Jest to układ liniowych równań, który pozwala na obliczenie temperatur węzłowych po czasie przy zadanych temperaturach w chwili $T=0$.

Przedstawiono tutaj cztery metody na uzyskanie temperatur węzłowych.

II Wyznaczanie nieustalonego rozkładu temperatury we wsadzie o przekroju okrągłym.Wstęp teoretyczny

Będziemy analizować proces nieustalonego przewodnictwa ciepła we wsadzie o przekroju okrągłym przy założeniu, że wymiana ciepła zachodzi w sposób osiowo-symetryczny. Na powierzchni zewnętrznej zachodzi konwekcja. Wsad jest w cylindrycznym układzie współrzędnych (później nastąpi zmiana). Dla takiego przypadku funkcjonał wygląda następująco:

$$J = \int_V \frac{k}{2} \left(\frac{dt}{dr} \right)^2 dV - \int_V q t dV + \int \frac{d}{2} (t - t_w)^2 dS$$

Proces minimalizacji funkcjonału został rozpatrzony dla jednego wybranego elementu. Temperatura wewnątrz elementów jest opisana wzorem:

$$t = \{N\}^T \{t\} = N_i t_i + N_j t_j = \frac{r_j - r}{\Delta r} t_i + \frac{r - r_i}{\Delta r} t_j$$

Δr krok siatki

N_i, N_j funkcje kształtu

t_i, t_j temperatury w węzłach elementu

Wyznaczamy pochodną temperatury po r :

$$\frac{dt}{dr} = \frac{\partial N_i}{\partial r} t_i + \frac{\partial N_j}{\partial r} t_j$$

Obliczono całki objętościowe funkcjonału. Aby to zrobić wyznaczono parametry całkowania:

$$dV = 2\pi r L dr$$

$$\int_S dS = 2\pi r_{max} L$$

L długość wsadu

r_{max} promień wsadu

Obliczamy pierwszą całkę:

$$\int_V \frac{k}{2} \left(\frac{dt}{dr} \right)^2 dV = \int_{r_i}^{r_j} \frac{k}{2} \left(\frac{\partial N_i}{\partial r} t_i + \frac{\partial N_j}{\partial r} t_j \right)^2 2\pi r L dr = \pi L \int_{r_i}^{r_j} k \left(\frac{t_j - t_i}{\Delta r} \right)^2 r dr = \pi L k \left(\frac{t_j - t_i}{\Delta r} \right)^2 \sum_{p=1}^{np} (r_p w_p \det[J])$$

Det[J] wyznacznik macierzy Jacobiego czyli Jakobian transformacji układu współrzędnych Ox do $O\xi$, gdzie element ma długość 2

w współczynniki wagi w punktach całkowania Gaussa r_p

Dany przypadek nie jest skomplikowany i możemy analitycznie obliczyć całkę. Jeśli problem się rozszerzy musimy użyć całkowania numerycznego.

Transformacja układu współrzędnych jest opisana wzorem:

$$r = N_i r_i + N_j r_j = \frac{1}{2}(1 - \xi)r_i + \frac{1}{2}(1 + \xi)r_j$$

Powyższe funkcje kształtu są zapisane w układzie lokalnym.

Wartość wyznacznika macierzy Jacobiego obliczymy ze wzoru:

$$\det[J] = \det\left[\frac{dr}{d\xi}\right] = \frac{\partial N_i}{\partial \xi} r_i + \frac{\partial N_j}{\partial \xi} r_j = \frac{r_j - r_i}{2} = \frac{\Delta r}{2}$$

Po wprowadzeniu zależności do wzoru obliczania całki otrzymujemy:

$$\int_V \frac{k}{2} \left(\frac{dt}{dx} \right)^2 dV = \Pi L K \left(\frac{t_j - t_i}{\Delta r} \right)^2 \sum_{p=1}^{np} (r_p w_p d\xi) = \Pi L K \frac{\Delta r}{2} \left(\frac{t_j - t_i}{\Delta r} \right)^2 \sum_{p=1}^{np} (r_p w_p)$$

W pierwszych założeniach ostatni człon równania Fouriera został zastąpiony parametrem Q . Teraz z tego skorzystamy i będziemy mogli uwzględnić niestacjonarność procesu zmiany temperatury.

$$Q = c_p \frac{dt}{dT}$$

Gdy to uwzględnimy, wariacyjne sformułowanie:

$$J = \int_V \frac{k}{2} \left(\frac{dt}{dx} \right)^2 dV - \int_V Q t dV + \int \frac{d}{2} (t - t_w)^2 dS$$

będzie równoważne z równaniem Fouriera:

$$\frac{\partial}{\partial x} \left(k_x(t) \frac{\partial t}{\partial x} \right) + \frac{\partial}{\partial y} \left(k_y(t) \frac{\partial t}{\partial y} \right) + \frac{\partial}{\partial z} \left(k_z(t) \frac{\partial t}{\partial z} \right) + \left(Q - c_p \frac{\partial t}{\partial T} \right) = 0$$

Otrzymujemy po uwzględnieniu wcześniejszych wyprowadzeń:

$$\int_V Q_t dV = \int_V c_p \frac{dt}{dT} t dV = 2\pi L c_p \int_{r_i}^{r_j} \frac{dt}{dT} t r dr =$$

$$\pi L c_p \sum_{p=1}^{n_p} \left(\frac{dt}{dT} (N_i t_i + N_j t_j) \right) r_p w_p \Delta r$$

Całka liczona po powierzchni zewnętrznej wynosi:

$$\int_S \frac{1}{2} d(t - t_\infty)^2 dS = \pi L r_{max} d(t_j - t_\infty)^2$$

Po podstawieniu wcześniejszych wzorów do funkcjonału dla elementu e otrzymujemy:

$$J_e = \pi L k \frac{\Delta r}{2} \left(\frac{t_j - t_i}{\Delta r} \right)^2 \sum_{p=1}^{n_p} (r_p w_p) + \pi L c_p \sum_{p=1}^{n_p} \left(\frac{dt}{dT} (N_i t_i + N_j t_j) \right) r_p w_p \Delta r$$

$$+ \pi L r_{max} d(t_j - t_\infty)^2$$

Można zapisać też w postaci:

$$J_e = k \frac{\Delta r}{2} \left(\frac{t_j - t_i}{\Delta r} \right)^2 \sum_{p=1}^{n_p} (r_p w_p) + c_p \Delta r \sum_{p=1}^{n_p} \left(\frac{dt}{dT} (N_i t_i + N_j t_j) \right) r_p w_p \Delta r$$

$$+ r_{max} d(t_j - t_\infty)^2$$

Aby wykonać minimalizację funkcjonału korzystamy z warunku ekstremum funkcji:

$$\begin{cases} \frac{\partial J_e}{\partial t_i} = 0 \\ \frac{\partial J_e}{\partial t_j} = 0 \end{cases}$$

Gdy zróżniczkujemy równania względem temperatury w węzłach otrzymamy:

$$\frac{\partial J_e}{\partial t_i} = k \Delta r \left(\frac{t_j - t_i}{\Delta r} \right) \left(-\frac{1}{\Delta r} \right) \sum_{p=1}^{n_p} r_p w_p + c_p \Delta r \sum_{p=1}^{n_p} \left(\frac{dt}{dT} N_i r_p w_p \right) = 0$$

$$\frac{dt}{dT} = \frac{N_i t_i + N_j t_j - N_i t_{i0} - N_j t_{j0}}{\Delta T}$$

$t_{i0} t_{j0}$ temperatury w danym elemencie w jego węzłach w poprzedniej chwili czasu

ΔT przyrost czasu

Po przekształceniu powyższego równania otrzymujemy:

$$\frac{\partial J_c}{\partial t_i} = k \Delta v \left(\frac{t_j - t_i}{\Delta v} \right) \left(-\frac{1}{\Delta v} \right) \sum_{p=1}^{np} v_p w_p +$$

$$\frac{c p \Delta v}{\Delta T} \sum_{p=1}^{np} \left((N_i t_i + N_j t_j - N_i t_{i0} - N_j t_{j0}) N_i v_p w_p \right) = 0$$

Kolejne przekształcenia:

$$\frac{\partial J_c}{\partial t_i} = t_i \left(\frac{k}{\Delta v} \sum_{p=1}^{np} v_p w_p + \frac{c p \Delta v}{\Delta T} \sum_{p=1}^{np} N_i^2 v_p w_p \right) +$$

$$+ t_j \left(-\frac{k}{\Delta v} \sum_{p=1}^{np} v_p w_p + \frac{c p \Delta v}{\Delta T} \sum_{p=1}^{np} N_i N_j v_p w_p \right) -$$

$$+ \frac{c p \Delta v}{\Delta T} \sum_{p=1}^{np} (N_i t_{i0} + N_j t_{j0}) N_i v_p w_p = 0$$

To samo wykonujemy dla t_j :

$$\frac{\partial J_c}{\partial t_j} = t_i \left(-\frac{k}{\Delta v} \sum_{p=1}^{np} v_p w_p + \frac{c p \Delta v}{\Delta T} \sum_{p=1}^{np} N_i N_j v_p w_p \right) +$$

$$+ t_j \left(\frac{k}{\Delta v} \sum_{p=1}^{np} v_p w_p + \frac{c p \Delta v}{\Delta T} \sum_{p=1}^{np} N_j^2 v_p w_p + 2 d v_{max} \right) -$$

$$+ \frac{c p \Delta v}{\Delta T} \sum_{p=1}^{np} (N_i t_{i0} + N_j t_{j0}) N_j v_p w_p - 2 d v_{max} t_{j0} = 0$$

Otrzymane dwa równania można zapisać w formie macierzowej:

$$[K_e] \{t\} + q \{F_e\} = 0$$

Elementy macierzy $[K]$ (lub $[H]$) i wektora $\{F\}$ (lub $\{P\}$) obliczamy ze wzorów:

$$k_{11} = \frac{k}{\Delta v} \sum_{p=1}^{np} v_p w_p + \frac{c p \Delta v}{\Delta T} \sum_{p=1}^{np} N_i^2 v_p w_p$$

$$k_{12} = -\frac{k}{\Delta v} \sum_{p=1}^{np} v_p w_p + \frac{c p \Delta v}{\Delta T} \sum_{p=1}^{np} N_i N_j v_p w_p$$

$$k_{21} = -\frac{k}{\Delta v} \sum_{p=1}^{np} v_p w_p + \frac{c p \Delta v}{\Delta T} \sum_{p=1}^{np} N_i N_j v_p w_p$$

$$k_{22} = \frac{k}{\Delta v} \sum_{p=1}^{np} v_p w_p + \frac{c p \Delta v}{\Delta T} \sum_{p=1}^{np} N_j^2 v_p w_p + 2 d v_{max}$$

$$F_1 = -\frac{c p \Delta v}{\Delta T} \sum_{p=1}^{np} (N_i t_{i0} + N_j t_{j0}) N_i v_p w_p$$

$$F_2 = -\frac{c p \Delta v}{\Delta T} \sum_{p=1}^{np} (N_i t_{i0} + N_j t_{j0}) N_j v_p w_p - 2 d v_{max} t_{j0}$$

Aby otrzymać układ równań dla całego obszaru dodajemy odpowiednie elementy z macierzy lokalnej każdego elementu do siebie.

$$[K] = \sum_{e=1}^{ne} [K_e]$$

$$\{F\} = \sum_{e=1}^{ne} \{F_e\}$$

Drugie rozwiązanie, które opiera się o rozwiązanie ogólne.

$$([H] + \frac{[C]}{\Delta T}) \{t_i\} - \left(\frac{[C]}{\Delta T} \right) \{t_o\} + \{P\} = 0$$

Dla dowolnego elementu wcześniej wyprowadzone wzory możemy zapisać tak razem z analogicznymi zależnościami:

$$[C] = \int_V c p \{N\} \{N\}^T dV$$

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

$$\{P\} = - \int_S \alpha \{N\} t_o dS$$

$$dV = 2\pi r L dr \quad \int_S dS = 2\pi r_{max} L$$

Całkujemy numerycznie po r:

$$\int_{r_i}^{r_j} r dr = \sum_{p=1}^{np} r_p w_p dt[j] = \frac{\Delta r}{2} \sum_{p=1}^{np} (r_p w_p)$$

Wektor funkcji kształtu zapisujemy następująco:

$$\{N\} = \begin{Bmatrix} N_i \\ N_j \end{Bmatrix} = \begin{Bmatrix} \frac{r_j - r}{\Delta r} \\ \frac{r - r_i}{\Delta r} \end{Bmatrix} \quad \{N\}^T = \begin{Bmatrix} N_i & N_j \end{Bmatrix} = \begin{Bmatrix} \frac{r_j - r}{\Delta r} & \frac{r - r_i}{\Delta r} \end{Bmatrix}$$

Macierz pochodnych funkcji kształtu wyrażamy przez wzór:

$$\left\{ \frac{\partial \{N\}}{\partial r} \right\} = \begin{Bmatrix} -\frac{1}{\Delta r} \\ \frac{1}{\Delta r} \end{Bmatrix} \quad \left\{ \frac{\partial \{N\}}{\partial r} \right\}^T = \begin{Bmatrix} -\frac{1}{\Delta r} & \frac{1}{\Delta r} \end{Bmatrix}$$

Zapisujemy macierz [C] i [H] i [P] według wcześniejszych wzorów:

$$\begin{aligned}
 [C] &= \int_V c_p \{N\} \{N\}^T dV = \int_V 2\pi L c_p \{N\} \{N\}^T r dr = \\
 &= 2\pi L c_p \int_V \begin{Bmatrix} N_i \\ N_j \end{Bmatrix} \begin{Bmatrix} N_i & N_j \end{Bmatrix} r dr = \\
 &= \pi L c_p \sum_{p=1}^{np} \left(r_p w_p \begin{bmatrix} N_i N_i & N_i N_j \\ N_i N_j & N_j N_j \end{bmatrix} \right)
 \end{aligned}$$

$$\begin{aligned}
 [H] &= \int_V k \left\{ \frac{\partial \{N\}}{\partial r} \right\} \left\{ \frac{\partial \{N\}}{\partial r} \right\}^T dV + \int_S \alpha \{N\} \{N\}^T dS = \\
 &= \int_V 2\pi L k \begin{Bmatrix} -\frac{1}{\Delta r} \\ \frac{1}{\Delta r} \end{Bmatrix} \begin{Bmatrix} -\frac{1}{\Delta r} & \frac{1}{\Delta r} \end{Bmatrix} r dr + 2\pi L \alpha \begin{bmatrix} N_i N_i & N_i N_j \\ N_i N_j & N_j N_j \end{bmatrix} r_{\text{max}} = \\
 &= 2\pi L k \begin{bmatrix} \frac{1}{\Delta r^2} & -\frac{1}{\Delta r^2} \\ -\frac{1}{\Delta r^2} & \frac{1}{\Delta r^2} \end{bmatrix} \int_{N_i}^{N_j} r dr + 2\pi L \alpha \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} r_{\text{max}} = \\
 &= \pi L k \begin{bmatrix} \frac{1}{\Delta r} & -\frac{1}{\Delta r} \\ -\frac{1}{\Delta r} & \frac{1}{\Delta r} \end{bmatrix} \sum_{p=1}^{np} (r_p w_p) + 2\pi L \alpha \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} r_{\text{max}}
 \end{aligned}$$

$$\{P\} = - \int_S \alpha \begin{Bmatrix} N_i \\ N_j \end{Bmatrix} t_{\infty} dS = -2\pi L r_{\text{max}} t_{\infty} \alpha \begin{Bmatrix} N_i \\ N_j \end{Bmatrix} = -2\pi L r_{\text{max}} t_{\infty} \alpha \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Otrzymane wzory wstawiamy do równania na niejawnym schemacie wyznaczania temperatury:

$$\begin{aligned}
 &\left(\pi L k \begin{bmatrix} \frac{1}{\Delta r} & -\frac{1}{\Delta r} \\ -\frac{1}{\Delta r} & \frac{1}{\Delta r} \end{bmatrix} \sum_{p=1}^{np} (r_p w_p) + 2\pi L \alpha \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} r_{\text{max}} + \frac{\pi L c_p \Delta r}{\Delta T} \sum_{p=1}^{np} (r_p w_p) \right. \\
 &\left. \begin{bmatrix} N_i N_i & N_i N_j \\ N_i N_j & N_j N_j \end{bmatrix} \right) \{t_1\} - \left(\frac{\pi L c_p \Delta r}{\Delta T} \sum_{p=1}^{np} (r_p w_p \begin{bmatrix} N_i N_i & N_i N_j \\ N_i N_j & N_j N_j \end{bmatrix}) \right) \{t_0\} - 2\pi L r_{\text{max}} t_{\infty} \alpha \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\
 &= 0
 \end{aligned}$$

Po uproszczeniu:

$$\left(\frac{k}{\Delta v} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \sum_{p=1}^{np} (v_p \omega_p) + 2d \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} v_{max} + \frac{cp\Delta v}{\Delta T} \sum_{p=1}^{np} \left(v_p \omega_p \begin{bmatrix} N_i N_i & N_i N_j \\ N_i N_j & N_j N_j \end{bmatrix} \right) \right) \begin{bmatrix} t_i \\ t_j \end{bmatrix} - \left(\frac{cp\Delta v}{\Delta T} \sum_{p=1}^{np} \left(v_p \omega_p \begin{bmatrix} N_i N_i & N_i N_j \\ N_i N_j & N_j N_j \end{bmatrix} \right) \right) \begin{bmatrix} t_0 \\ 1 \end{bmatrix} - 2 v_{max} + \omega_d \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 0$$

Powyższy wzór możemy zapisać jako:

$$[K] \begin{bmatrix} t_i \\ t_j \end{bmatrix} + \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = 0$$

$$\begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} \begin{bmatrix} t_i \\ t_j \end{bmatrix} + \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = 0$$

Składowe macierzy i wektora wyliczamy ze wzoru:

$$k_{11} = \frac{k}{\Delta v} \sum_{p=1}^{np} v_p \omega_p + \frac{cp\Delta v}{\Delta T} \sum_{p=1}^{np} N_i^2 v_p \omega_p$$

$$k_{12} = -\frac{k}{\Delta v} \sum_{p=1}^{np} v_p \omega_p + \frac{cp\Delta v}{\Delta T} \sum_{p=1}^{np} N_i N_j v_p \omega_p$$

$$k_{21} = -\frac{k}{\Delta v} \sum_{p=1}^{np} v_p \omega_p + \frac{cp\Delta v}{\Delta T} \sum_{p=1}^{np} N_i N_j v_p \omega_p$$

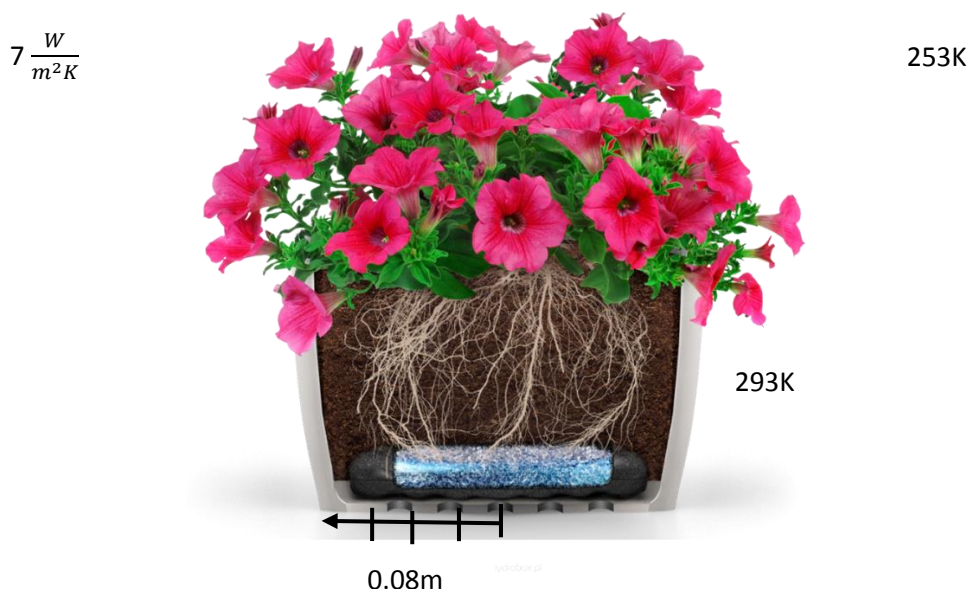
$$k_{22} = \frac{k}{\Delta v} \sum_{p=1}^{np} v_p \omega_p + \frac{cp\Delta v}{\Delta T} \sum_{p=1}^{np} N_j^2 v_p \omega_p + 2d v_{max}$$

$$F_1 = -\frac{cp\Delta v}{\Delta T} \sum_{p=1}^{np} (N_i t_{i0} + N_j t_{j0}) N_i v_p \omega_p$$

$$F_2 = -\frac{cp\Delta v}{\Delta T} \sum_{p=1}^{np} (N_i t_{i0} + N_j t_{j0}) N_j v_p \omega_p - 2d v_{max} + \omega_d$$

III Własne zagadnienie

Jako własne zagadnienie będę badać rozkład temperatury w doniczkach z różnych mniej lub bardziej popularnych materiałów. Rośliny potrzebują odpowiedniej temperatury, aby nie uległy uszkodzeniu, więc doniczki powinny zapewniać pewną ochronę przed ujemnymi temperaturami otoczenia. Doniczkę wynosimy z pomieszczenia o temperaturze 293K na zewnątrz, gdzie panuje temperatura 253K. Zostawiamy na około 5 i pół godziny.



Dane stałe:

| | |
|--|--------------------|
| Minimalny promień | 0m |
| Maksymalny promień | 0.08m |
| Współczynnik konwekcyjnej wymiany ciepła | $7 \frac{W}{m^2K}$ |
| Temperatura początkowa | 293K |
| Temperatura otoczenia | 253K |
| Czas procesu | ~5,5h |

Dane materiałów:

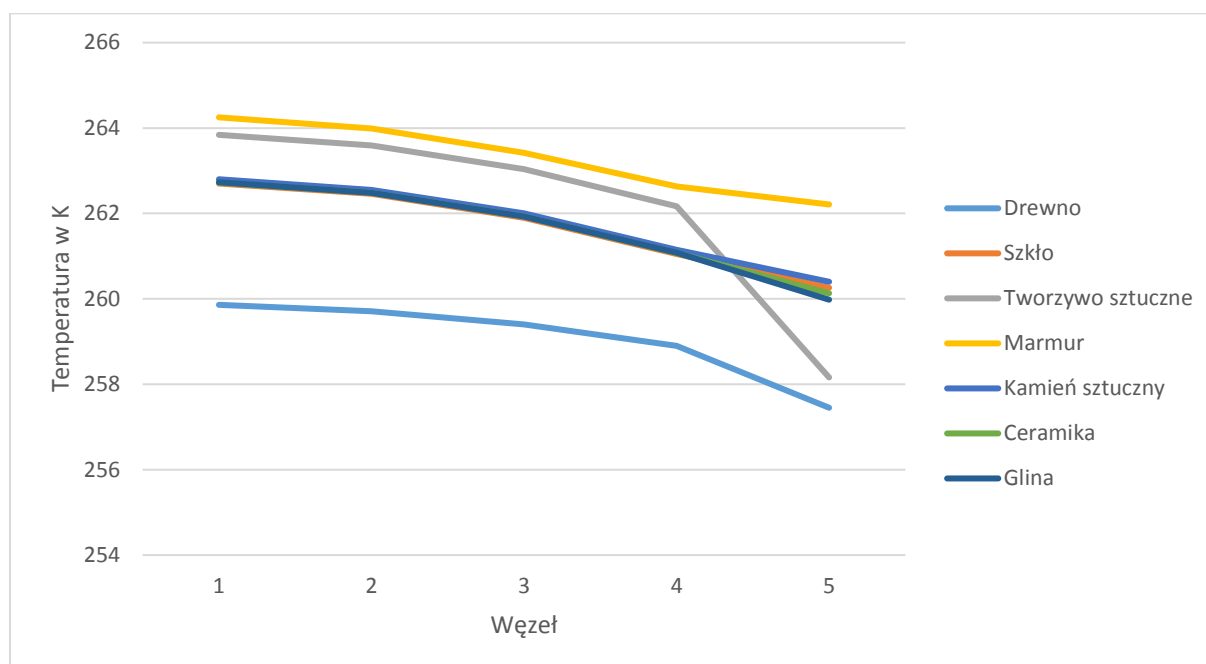
| Nazwa | Współczynnik przewodzenia ciepła $\frac{W}{mK}$ | Gęstość $\frac{kg}{m^3}$ | Efektywne ciepło właściwe $\frac{J}{kgK}$ |
|-----------------------|---|--------------------------|---|
| Drewno dąb | 0.4 | 800 | 2510 |
| Szkło mozaika szklana | 1.2 | 2000 | 840 |
| Tworzywo sztuczne | 0.2 | 1000 | 250 |
| Marmur | 3.5 | 2800 | 920 |
| Kamień sztuczny | 1.3 | 1750 | 1000 |
| Ceramika | 1.0 | 2000 | 800 |
| Gлина | 0.85 | 1800 | 840 |
| Gleba | 0.9 | 1800 | 1200 |

Węzły będą się znajdować co 2cm. Będziemy mogli zobaczyć jak zmienia się temperatura na powierzchni doniczki i w głąb gleby.

0.08m

Wyniki I

| Nazwa | Ziemia | | | | Doniczka |
|-----------------------|--------|--------|--------|--------|----------|
| | t1 | t2 | t3 | t4 | t5 |
| Drewno dąb | 259,86 | 259,71 | 259,40 | 258,90 | 257,45 |
| Szkło mozaika szklana | 262,70 | 262,46 | 261,90 | 261,05 | 260,26 |
| Tworzywo sztuczne | 263,84 | 263,59 | 263,04 | 262,17 | 258,16 |
| Marmur | 264,25 | 263,99 | 263,42 | 262,63 | 262,21 |
| Kamień sztuczny | 262,80 | 262,55 | 262,00 | 261,15 | 260,40 |
| Ceramika | 262,72 | 262,48 | 261,93 | 261,07 | 260,13 |
| Glina | 262,73 | 262,48 | 261,93 | 261,08 | 259,98 |



IV Wnioski I

Zbadaliśmy siedem rodzajów materiałów, z których można stworzyć doniczkę. Na wykresie możemy zauważyć 4 elementy, które warto omówić. Pierwsza z nich to doniczka marmurowa, która utrzymała najwyższą temperaturę wewnątrz oraz sama nie przejęła w tak dużym stopniu jak pozostałe temperaturę otoczenia. 264K to najwyższa temperatura, którą udało się uzyskać, ale to nadal ujemna. Kolejnym punktem jest doniczka z tworzywa sztucznego. Wyraźnie widać skok na wykresie w miejscu przejścia z doniczki do ziemi. To tworzywo przejęło niską temperaturę 256K bliską temperaturze otoczenia-253K. Chłód nie przeszedł dalej i w rdzeniu jest temperatura 263K. Jest to drugi wynik w najwyższych temperaturach. Doniczki marmurowe i z tworzywa sztucznego są dobrym pomysłem dla ogrodników. Ze względu na cenę marmuru i jego wagę wygodniejsze są doniczki plastikowe, które są najczęściej spotykane

i używane. Teraz kolej na omówienie grupy kilku materiałów z prawie takimi samymi wynikami. Szkło, sztuczny kamień, ceramika i glina uzyskały wewnątrz doniczki temperaturę 263K, a na powierzchni 260K. Temperatura łagodnie spada w kierunku środka. Nie jest to najlepszy wynik, ale można uznać za zadowalający w porównaniu do następnego materiału. Najgorzej wypadła drewniana doniczka. Doniczka miała temperaturę 257K a wewnątrz 259K. Ten materiał nie nadaje się na doniczkę, którą chcemy trzymać w ogródku przez cały rok. Lepiej wybrać doniczkę z tworzywa sztucznego lub marmuru. Drewniana doniczka może przebywać na zewnątrz latem.

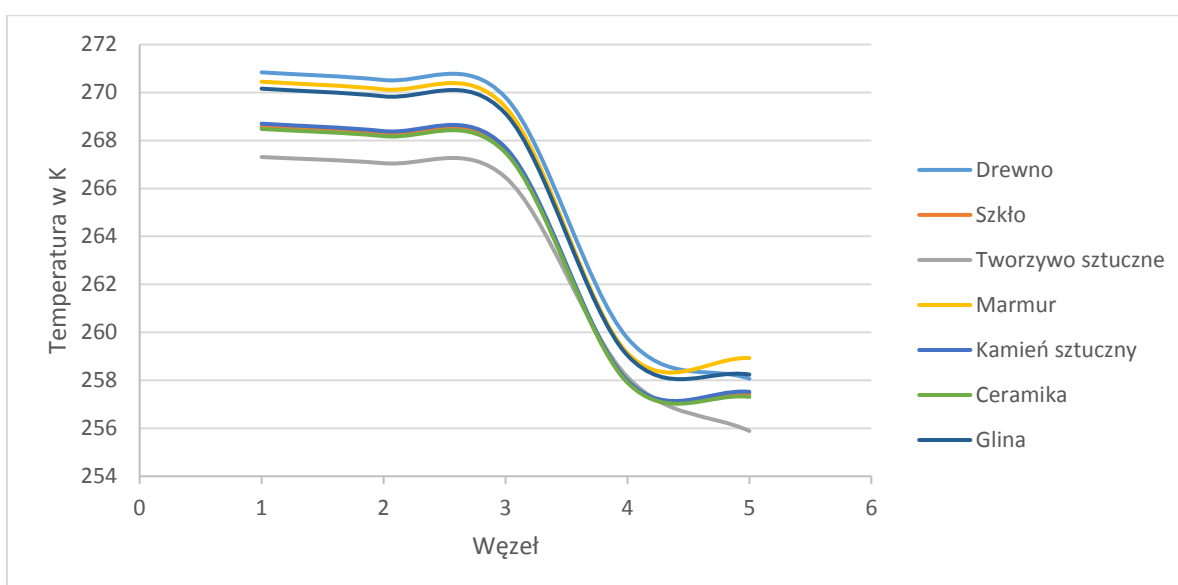
Spróbujmy dodać izolację do doniczek. Wykorzystanym materiałem będzie płyta z trzciny.

| Nazwa | Współczynnik przewodzenia ciepła $\frac{W}{mK}$ | Gęstość $\frac{kg}{m^3}$ | Efektywne ciepło właściwe $\frac{J}{kgK}$ |
|-----------------|--|-----------------------------|--|
| Płyta z trzciny | 0.07 | 250 | 1460 |

0.08m

Wyniki II

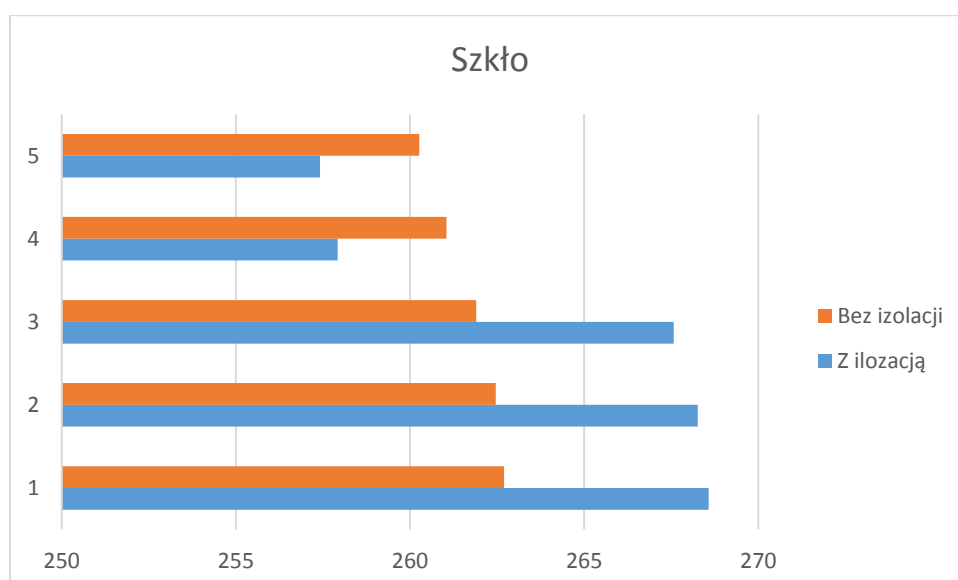
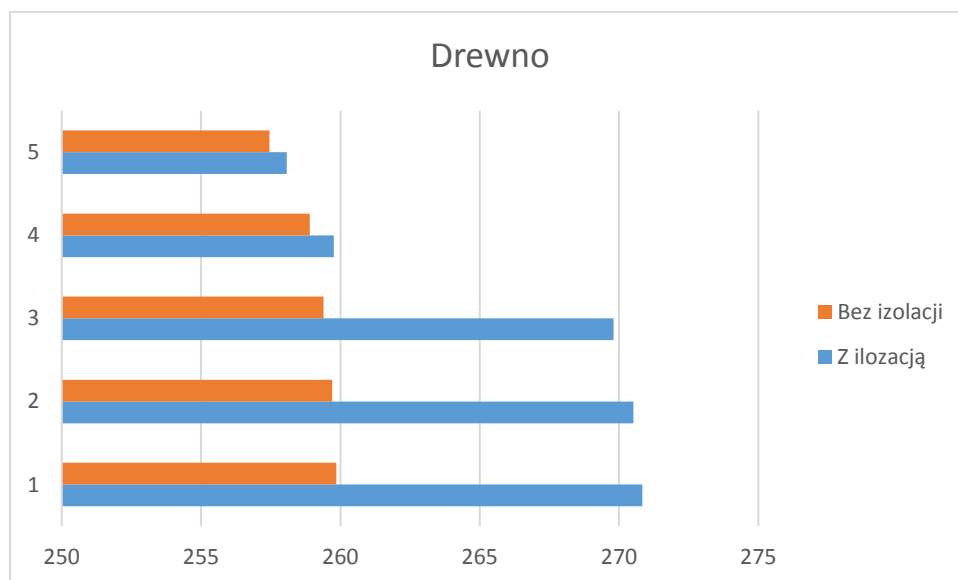
| Nazwa | Ziemia | | | Izolacja | Doniczka |
|-----------------------|--------|--------|--------|----------|----------|
| | t1 | t2 | t3 | t4 | t5 |
| Drewno dąb | 270,84 | 270,52 | 269,80 | 259,76 | 258,07 |
| Szkło mozaika szklana | 268,57 | 268,26 | 267,57 | 257,92 | 257,42 |
| Tworzywo sztuczne | 267,31 | 267,05 | 266,46 | 258,14 | 255,89 |
| Marmur | 270,45 | 270,13 | 269,39 | 259,14 | 258,93 |
| Kamień sztuczny | 268,70 | 268,39 | 267,70 | 257,99 | 257,52 |
| Ceramika | 268,48 | 268,18 | 267,49 | 257,91 | 257,31 |
| Glina | 270,16 | 269,84 | 269,12 | 259,04 | 258,24 |

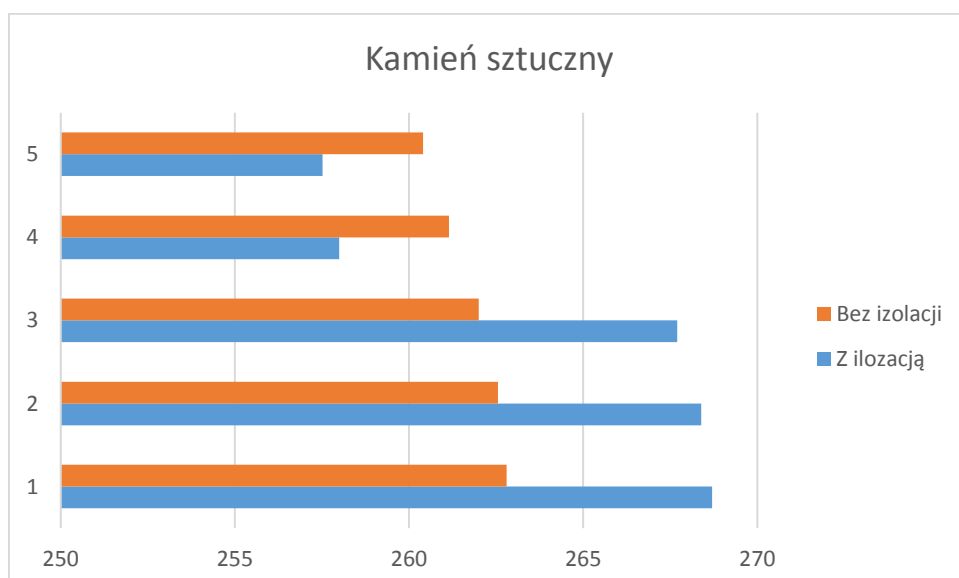
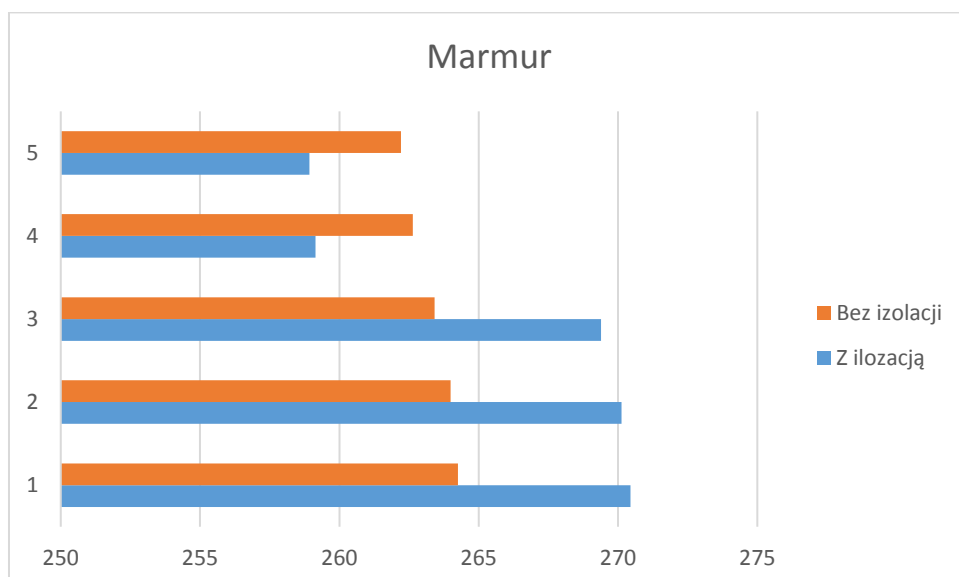
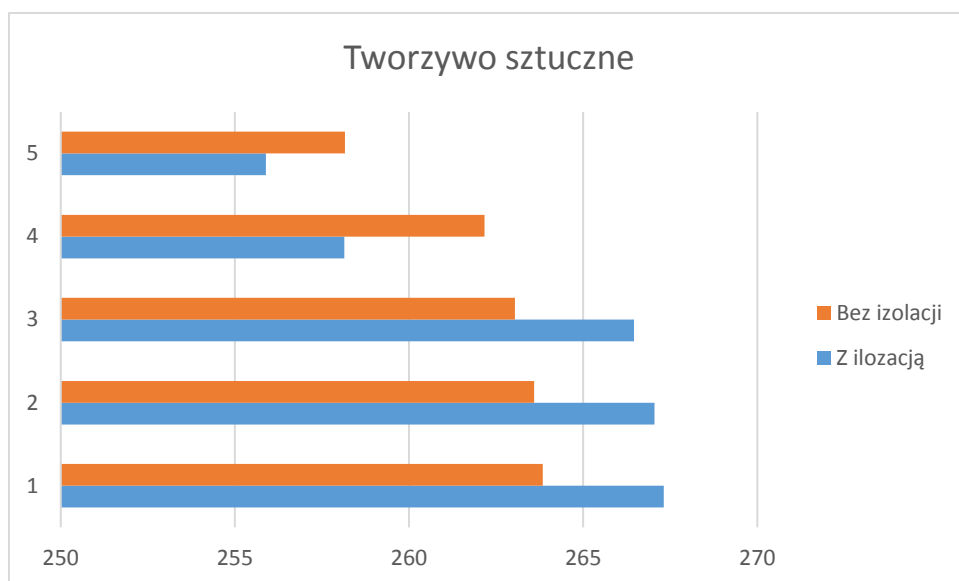


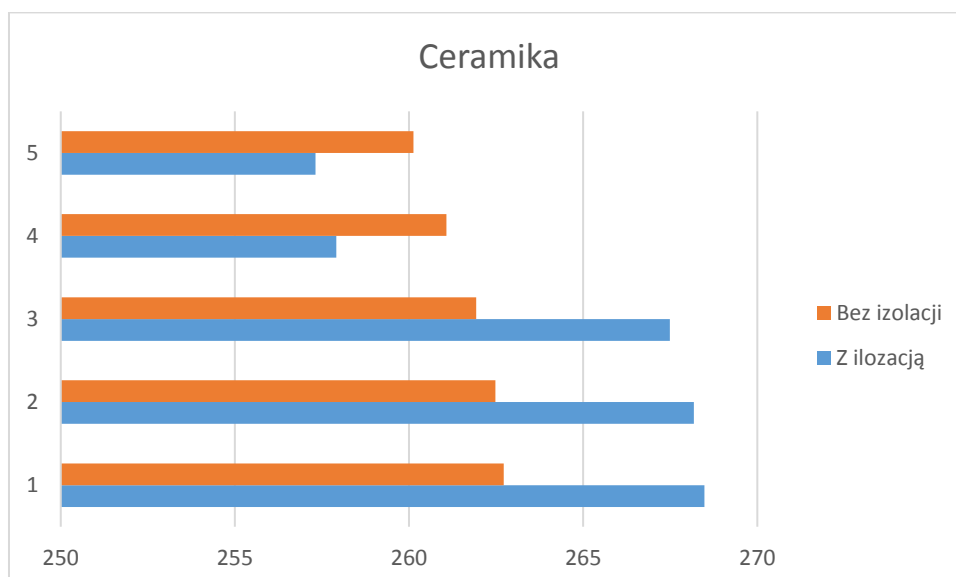
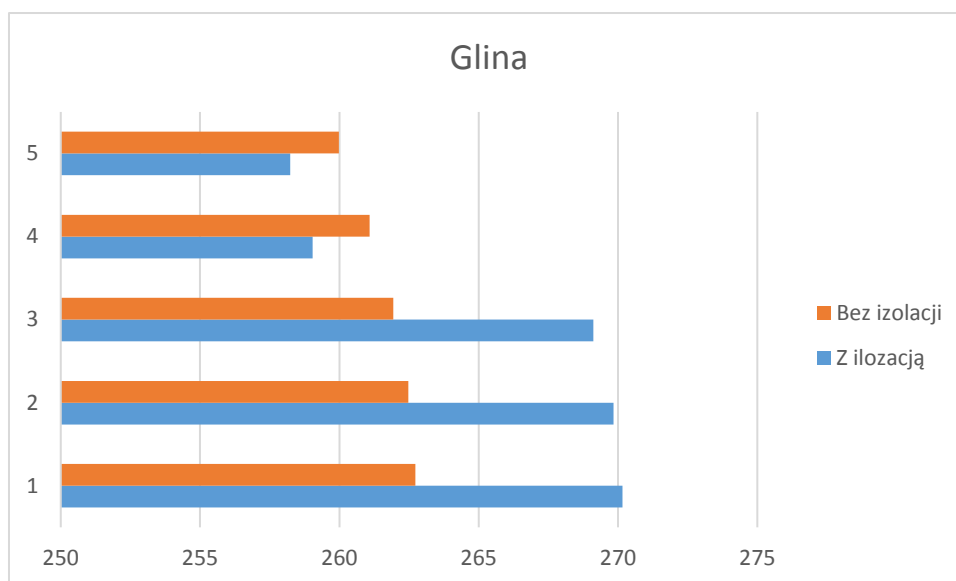
V Wnioski II

Gdy zastosujemy izolację wszystkie materiały zachowują się podobnie- temperatura wewnątrz doniczki wzrosła. Zastosowanie płyty z trzciny spowodowało, że doniczka może być wykonana z każdego materiału. Drewno, które wypadło najgorzej teraz jest najlepszym materiałem w połączeniu z izolacją. Wewnątrz panuje temperatura 270K czyli niewiele poniżej 0°C. W takim układzie jest to idealna doniczka do ogrodu. Marmur utrzymał swoje drugie miejsce, natomiast tworzywo sztuczne z izolacją nie dało dużej zmiany wartości temperatury. Widać, że wystarczające jest użycie samego tworzywa. Gлина uzyskała podobne wartości jak marmur, a wcześniej była w grupie materiałów, które można używać, ale są lepsze zamienniki. Kamień sztuczny i ceramika zwiększyły temperaturę wewnątrz o 7K. Na wykresach porównano temperatury poszczególnych materiałów z i bez izolacji. W każdym przypadku widać wzrost temperatury wewnątrz.

Porównanie wyników







VI Wnioski- podsumowanie

Sprawdzono dwa warianty doniczek- z i bez izolacji. W kategorii bez izolacji najlepszym materiałem na doniczki, które znajdują się w temperaturze poniżej zera, jest marmur. Z spadkiem o 1K drugim materiałem jest tworzywo sztuczne. Utrzymały one najwyższą temperaturę w rdzeniu. Jednak te wyniki były niewystarczające i zbadano co się stanie gdy dodamy warstwę izolacji. Wyniki były zaskakujące i drewniana doniczka stała się najlepszą. W środku była temperatura bliska 0°C. Doniczka z tworzywa sztucznego nie wykazała dużych zmian jak inne. Można stwierdzić, że jest to dobry materiał na doniczkę bez korzystania z kolejnych warstw izolacji. Być może dlatego plastikowe doniczki są najpopularniejsze.

VII Program

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

using namespace std;

struct wezel
{
    double r0; //wspolrzedna
    int stan; //0 brak, lub tylko 2 koniec
    int ID; //indeks elementu w siatce
    int waga; //waga wezla =1
};

struct element
{
    double k; //wspolczynnik
    double alfa; //warunek 2
    double tot; //temperatura otoczenia

    double c; //cieplo wlasciwe
    double deltar; //skok r
    double rmax; //maksymalny promien
    double ro; //gestosc
    double deltatau;
    double temperaturapoczatkowa1;
    double temperaturapoczatkowa2;

    wezel wezly[2]; //wezly dwa
    double H[2][2]; //macierz pojemnosci cieplnej
    double P[2]; // macierz obciazen

    void oblicz_macierze_lokalne()
    {
        //macierz H
        for (int i = 0; i < 2; i++)
            for (int j = 0; j < 2; j++)
                H[i][j] = 0;

        double ksi = 0.5573;
        //funkcje ksztaltu
        double Ni[2];
        double Nj[2];

        Ni[0] = 0.5*(1 - ksi);
        Ni[1] = 0.5*(1 - (-ksi));

        Nj[0] = 0.5*(1 + ksi);
        Nj[1] = 0.5*(1 + (-ksi));

        //r
        double r[2];
        r[0] = (Ni[0] * wezly[0].r0) + (Nj[0] * wezly[1].r0);
        r[1] = (Ni[1] * wezly[0].r0) + (Nj[1] * wezly[1].r0);
    }
};
```

```

        H[0][0] = ((k / deltar)*((r[0] * wezly[0].waga) + (r[1] *
wezly[1].waga))) + ((c*ro*deltar) / deltatau)*((Ni[0] * Ni[0] * r[0] * wezly[0].waga)
+ (Ni[1] * Ni[1] * r[1] * wezly[1].waga));
        H[0][1] = ((-(k / deltar))*((r[0] * wezly[0].waga) + (r[1] *
wezly[1].waga))) + ((c*ro*deltar) / deltatau)*((Ni[0] * Nj[0] * r[0] * wezly[0].waga)
+ (Ni[1] * Nj[1] * r[1] * wezly[1].waga));
        H[1][0] = ((-(k / deltar))*((r[0] * wezly[0].waga) + (r[1] *
wezly[1].waga))) + ((c*ro*deltar) / deltatau)*((Ni[0] * Nj[0] * r[0] * wezly[0].waga)
+ (Ni[1] * Nj[1] * r[1] * wezly[1].waga));

        if( wezly[1].stan==2)
            H[1][1] = ((k / deltar)*((r[0] * wezly[0].waga) + (r[1] *
wezly[1].waga))) + ((c*ro*deltar) / deltatau)*((Nj[0] * Nj[0] * r[0] * wezly[0].waga)
+ (Nj[1] * Nj[1] * r[1] * wezly[1].waga))+2*alfa*rmax;
        else
            H[1][1] = ((k / deltar)*((r[0] * wezly[0].waga) + (r[1] *
wezly[1].waga))) + ((c*ro*deltar) / deltatau)*((Nj[0] * Nj[0] * r[0] * wezly[0].waga)
+ (Nj[1] * Nj[1] * r[1] * wezly[1].waga));

        //macierz P
        for (int i = 0; i < 2; i++)
            P[i] = 0;

        P[0] = ((-(c*ro*deltar)) / deltatau)*(((Ni[0] * temperaturapoczkowa1 +
Nj[0] * temperaturapoczkowa2)*Ni[0] * r[0] * wezly[0].waga) + ((Ni[1] *
temperaturapoczkowa1 + Nj[1] * temperaturapoczkowa2)*Ni[1] * r[1] *
wezly[1].waga));
        if (wezly[1].stan == 2)
            P[1] = ((-(c*ro*deltar)) / deltatau)*(((Ni[0]
*temperaturapoczkowa1 + Nj[0] *temperaturapoczkowa2)*Nj[0] * r[0] * wezly[0].waga)
+ ((Ni[1] *temperaturapoczkowa1 + Nj[1] * temperaturapoczkowa2)*Nj[1] * r[1] *
wezly[1].waga)) - 2 * alfa*rmax*tot;
        else
            P[1] = ((-(c*ro*deltar)) / deltatau)*(((Ni[0]
*temperaturapoczkowa1 + Nj[0] *temperaturapoczkowa2)*Nj[0] * r[0] * wezly[0].waga)
+ ((Ni[1] * temperaturapoczkowa1 + Nj[1] *temperaturapoczkowa2)*Nj[1] * r[1] *
wezly[1].waga));

    }
};

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 q = -GH[j][i] / GH[i][i];
            for (int k = i; k <= nh; k++)
            {
                GH[j][k] += q* GH[i][k];
            }
            GP[j] += q*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()
{
    wezel * listawezlow = new wezel[5];
    for (int i = 0; i < 5; i++)
    {
        listawezlow[i].r0 = 0.02*i;
        listawezlow[i].stan = 0;
        listawezlow[i].ID = i+1;
        listawezlow[i].waga = 1;
    }
    listawezlow[4].stan = 2;

    element *listaelementow = new element[4];

    for (int i = 0; i < 4; i++)
    {

```

```

        listaelementow[i].wezly[0] = listawezlow[i];
        listaelementow[i].wezly[1] = listawezlow[i+1];
        listaelementow[i].rmax = 0.08;
        listaelementow[i].deltar = 0.02;
        listaelementow[i].alfa = 7;
        listaelementow[i].deltatau = 50;
        listaelementow[i].tot = 253;
        listaelementow[i].temperaturapoczkowa1 = 293;
        listaelementow[i].temperaturapoczkowa2 = 293;
    }
    //ziemia
    for (int i = 0; i < 2; i++)
    {
        listaelementow[i].ro = 1800;
        listaelementow[i].c = 1200;
        listaelementow[i].k = 0.9;
    }
    //izolacja
    listaelementow[2].ro = 250;
    listaelementow[2].c = 1460;
    listaelementow[2].k = 0.07;

    //doniczka
    listaelementow[3].ro = 1800;
    listaelementow[3].c = 1200;
    listaelementow[3].k = 0.85;

    for (int i = 0; i < 4; i++)
        listaelementow[i].oblicz_macierze_lokalne();

    Siatka Siatka;
    Siatka.ne = 4;
    Siatka.nh = 5;
    Siatka.elementy = listaelementow;
    Siatka.stworz_macierze();
    Siatka.gauss();

    for (int j = 0; j < 400; j++)
    {
        for (int i = 0; i < 4; i++)
        {
            Siatka.elementy[i].temperaturapoczkowa1 =
Siatka.Gt[Siatka.elementy[i].wezly[0].ID - 1];
            Siatka.elementy[i].temperaturapoczkowa2 =
Siatka.Gt[Siatka.elementy[i].wezly[1].ID - 1];
            Siatka.elementy[i].oblicz_macierze_lokalne();
        }

        Siatka.stworz_macierze();
        Siatka.gauss();
    }

    system("PAUSE");
    return 0;
}

```

VIII Załączniki

Tablice z danymi do programu.