

# Metoda Elementów Skończonych

## Projekt - Nieustalona wymiana ciepła w układzie dwuwymiarowym na przykładzie okna drzwi piekarnika.

Mikołaj Stępniewski  
286247, IS (WIMIP)

---

### WSTĘP

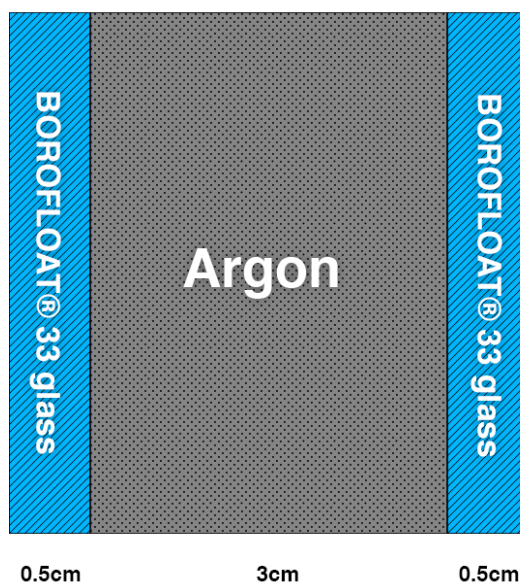
Program MES został użyty do przeprowadzenia symulacji ogrzewania okna drzwi piekarnika. Głównym celem było zbadanie jak zmienia się temperatura na zewnętrznej powierzchni okna - a tym samym jakie straty ciepła zachodzą podczas korzystania z piekarnika. W dalszej części przedstawione będą wykresy rozkładu ciepła w czasie oraz animacje, które w najlepszy sposób zobrazują rozkład gradientu temperatur w badanej strukturze.

### ZAŁOŻENIA

Przyjąłem, że badane okno będzie składać się z trzech warstw (od lewej):

1. Borokrzemianowa płaska szyba (typu float) BOROFLOAT® 33 produkowana przez firmę Schott o grubości 5 mm.
2. Przestrzeń o szerokości 3 cm, wypełniona argonem.
3. Tak jak w punkcie 1.

Strukturę okna przedstawia poniższy rysunek:



Całość badanego obszaru ma wymiary 4 cm x 4 cm i jest reprezentowane przez siatkę MES o wielkości 41 x 41 węzłów (40 x 40 elementów), gdzie każdy element ma szerokość 1 mm. Ważne dla stabilności obliczeń było, aby każda z warstw składała się z co najmniej pięciu węzłów.

Parametry fizyczne oraz termiczne dla szkła BOROFLOAT® 33 zostały zaczerpnięte z oficjalnej strony producenta. Co do argonu, posiłkowałem się arkuszem (*cieplo\_sc.xls*) znalezionym w internecie. Pliki pdf pobrane ze strony [schott.com](http://schott.com) oraz arkusz z parametrami [m.in.](#) dla argonu znajdują się w katalogu *docs*. Całość została przedstawiona w poniższej tabeli:

	Ciepło właściwe [J/kgK]	Współczynnik przewodzenia ciepła [W/mK]	Gęstość [kg/m3]
BOROFLOAT® 33	830	1,2	2230
Argon	520	0,017	1,7

Aby policzyć współczynnik konwekcji  $\alpha \left[ \frac{W}{m^2K} \right]$  posłużyłem się wzorem znalezionym w “Zbiór zadań z techniki cieplnej”, Wydawnictwo Politechniki Warszawskiej rok. 1973:

Powierzchnie wewnętrzne:

przy  $t_{f1} - \vartheta_1 < 5^\circ C$

$$\alpha = 3,49 + 0,093(t_{f1} - \vartheta_1)$$

przy  $t_{f1} - \vartheta_1 > 5^\circ C$

$$\alpha = \varphi \sqrt[4]{t_{f1} - \vartheta_1}$$

gdzie:  $t_{f1}$  - temperatura otoczenia

$\vartheta_1$  - temperatura powierzchni

przy czym:  $\varphi = 2,32$  - dla powietrza w zamkniętym pomieszczeniu (po lewej stronie okna oraz przy wyłączonym termoobiegu po prawej stronie okna)

$\varphi = 3,2$  - dla powietrza w pomieszczeniach produkcyjnych z wirującymi elementami maszyn.

Przyjąłem zatem, że dla włączonego termoobiegu:

$$\varphi = 5$$

Dla  $\varphi = 2,32$  uzyskałem współczynnik konwekcji: (*left, right* - oznacza dla której powierzchni)

$$\alpha_{\text{left}} = 3,49$$

$$\alpha_{\text{right}} = 9,03$$

Dla  $\varphi = 5$  (termoobieg) uzyskałem współczynnik konwekcji:

$$\alpha_{\text{right}} = 19,45$$

**W pliku GlobalData.swift, linijka 158:**

```

1      static func calculateAlpha(t_surf:Double, t_ambient:Double, phi:Double? = nil) -> Double {
2          if t_ambient - t_surf < 5 {
3              return 3.49 + 0.093 * (t_ambient - t_surf)
4          } else {
5              return (phi ?? 2.32) * pow(t_ambient - t_surf, 0.25)
6          }
7      }

```

Powyższa metoda wywoływana jest przez każdy element znajdujący się na powierzchni. Dla reszty parametr  $\alpha$  równy jest  $nil$ . Współczynnik ten będzie wykorzystywany podczas liczenia całki po powierzchni.

Ustaliłem, że po lewej stronie (od strony pokoju) temperatura otoczenia wynosić będzie **21°C**, zaś po prawej stronie (wewnątrz piekarnika) **250°C**. Temperatura początkowa całości układu wynosić będzie **21°C** (wyrównanie z temperaturą pokojową).

Kolejnym elementem do policzenia jest krok czasowy, który powinien być dobrany pod względem badanego materiału w celu zachowania stabilności rozwiązania. Do obliczenia kroku czasowego posłużyłem się poniższym wzorem:

$$\Delta\tau = \frac{2c\rho\left(\frac{B}{nB}\right)^2}{k}$$

gdzie:  $c$  - ciepło właściwe materiału

$k$  - współczynnik przewodzenia ciepła

$\rho$  - gęstość materiału

**W pliku GlobalData.swift, linijka 148:**

```
1 func setStableTimeStep(forMaterial material:ElementMaterial) {
2     let params = GlobalData.getParameters(for: material)
3     let k = params["k"] as! Double, c = params["c"] as! Double, ro = params["ro"] as! Double
4
5     let Asr = k/(c*ro)
6     self.d_tau = round(pow(B/Double(nB), 2.0)/(0.5*Asr))
7 }
```

Dla szyby  $\Delta\tau$  wyniosło **3 sekundy** a dla argonu 2 sekundy. Uznałem, że użyję kroku czasowego dla szyby, gdyż to ona odgrywa najważniejszą rolę w układzie, poza tym różnica między krokiem czasowym dla szyby i argonu nie jest duża i nie wpłynie negatywnie na wyniki.

Podsumowując:

	Wielkość siatki MES	Temperatura początkowa układu	Temperatura otoczenia (strona lewa)	Temperatura otoczenia (strona prawa)	Współczynnik konwekcji (strona lewa)	Współczynnik konwekcji (strona prawa)	Krok czasowy
<b>termoobieg</b>	41 x 41 węzłów	21°C	21°C	250°C	3,49 W/m2K	19,45 W/m2K	3 sekundy
<b>brak termoobiegu</b>	↑	↑	↑	↑	↑	<b>9,03 W/m2K</b>	↑

## KRÓTKI WSTĘP TEORETYCZNY

Rozwiązanie zagadnienia nieustalonej wymiany ciepła w układzie dwuwymiarowym sprowadza się do uzyskania układu równań MES na podstawie poniższego wzoru:

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

gdzie poszczególne wyrażenia obliczono za pomocą równań:

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

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

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

gdzie:  $\{t_1\}$  - wektor szukanych temperatur węzłowych

$\{t_0\}$  - wektor temperatur węzłowych na początku danego kroku czasowego

$V$  - objętość

$k$  - współczynnik przewodzenia ciepła

$\{N\}$  - wektor kolejnych węzłowych funkcji kształtu (określone lokalnie)

$x, y$  - globalne współrzędne

$T$  - oznacza wektor transponowany

$\alpha$  - współczynnik konwekcji

$S$  - powierzchnia

$t_\infty$  - temperatura otoczenia

$c$  - ciepło właściwe

$\rho$  - gęstość

---

### WAŻNE!

W programie przez  $[H]$  rozumiemy  $\left([H] + \frac{[C]}{\Delta\tau}\right)$  a przez  $\{P\}$  rozumiemy  $\left(\frac{[C]}{\Delta\tau}\right)\{t_0\} + \{P\}$ .

Temperatury na początku każdego kroku czasowego wyznaczone są za pomocą interpolacji w danym punkcie całkowania:

$$t_0 = \sum_{i=1}^4 N_{pc_i} t_{start_i}$$

gdzie:  $N_{pc}$  - zbiór funkcji kształtu dla danego punktu całkowania

$t_{start}$  - zbiór temperatur węzłowych na początku kroku czasowego

Funkcje kształtu dla układu 2D wyglądają następująco:

$$N_1 = \frac{1}{4}(1 - \xi)(1 - \eta), \quad N_2 = \frac{1}{4}(1 + \xi)(1 - \eta), \quad N_3 = \frac{1}{4}(1 + \xi)(1 + \eta), \quad N_4 = \frac{1}{4}(1 - \xi)(1 + \eta)$$

Natomiast punkty całkowania Gaussa (2p):

pc	$\xi$	$\eta$	waga
1	$-1/\sqrt{3}$	$-1/\sqrt{3}$	1
2	$1/\sqrt{3}$	$-1/\sqrt{3}$	1
3	$1/\sqrt{3}$	$1/\sqrt{3}$	1
4	$-1/\sqrt{3}$	$1/\sqrt{3}$	1

Całkowanie za pomocą kwadratury Gaussa przeprowadzane jest na podstawie wzoru:

$$f(x) = \sum_{i=1}^{npc} (f(x_{pc_i}) * w_i) * \det J$$

gdzie:  $f(x_{pc_i})$  - wartość funkcji całkowanej dla interpolowanego x

## ALGORYTM I IMPLEMENTACJA

Wymieniony w powyższej sekcji układ równań musimy obliczyć, dla każdego kroku czasowego a następnie rozwiązać go za pomocą metody eliminacji Gaussa (w pliku *Solver.swift*). Natomiast dla uproszczenia, przedstawię algorytm obliczania macierzy  $[H]$  wraz z wektorem  $\{P\}$  dla danego kroku czasowego (metoda *compute* w pliku *GlobalData.swift*):

**Dla każdego elementu w siatce:**

1. Wyzeruj  $[H]$  oraz  $\{P\}$ .
2. Dla każdego punktu całkowania:
  - 2.1. Oblicz macierz Jakobiego 2D.
  - 2.2. Oblicz wyznacznik macierzy Jakobiego 2D.
  - 2.3. Wyzeruj  $t_0$ .
  - 2.4. Oblicz  $\frac{\delta N}{\delta x}$ ,  $\frac{\delta N}{\delta y}$  oraz  $t_0$ .
  - 2.5. Oblicz i dołącz elementy macierzy  $[H]$  oraz wektora  $\{P\}$  (całka po  $V$ ).
3. Dla każdej powierzchni zewnętrznej siatki:
  - 3.1. Oblicz wyznacznik macierzy Jakobiego 1D.
  - 3.2. Dla obu węzłów w powierzchni:
    - 3.2.1. Oblicz i dołącz elementy macierzy  $[H]$  oraz wektora  $\{P\}$  (całka po  $S$ ).
4. Dokonaj agregacji lokalnej macierzy  $[H]$  oraz lokalnego wektora  $\{P\}$  do ich globalnych odpowiedników.

Punkt 2. z powyższego algorytmu został zaimplementowany w następujący sposób:

2.1., 2.2., 2.3. Oblicz macierz Jakobiego 2D, oblicz wyznacznik tej macierzy oraz wyzeruj  $t_0$   
**W pliku GlobalData, linijka 279:**

```
1 // ipi - index of a current integration point
2 let jacobian = Jacobian(intPoint: ipi, xs: coordsX, ys: coordsY, dNdKsis:
  ips_dNdKsi, dNdEtas: ips_dNdEta)
3 detJ = abs(jacobian.det)
4 let dNdKsi = ips_dNdKsi[ipi], dNdEta = ips_dNdEta[ipi]
5 t0 = 0.0
```

2.4. Oblicz  $\frac{\delta N}{\delta x}$ ,  $\frac{\delta N}{\delta y}$  oraz  $t_0$ .

**W pliku GlobalData, linijka 285:**

```
1 for i in 0 ..< 4 {
2   dNdx[i] = jacobian.matrixInverted[0][0] * dNdKsi[i] + jacobian.matrixInverted[0]
   [1] * dNdEta[i]
3
4   dNdy[i] = jacobian.matrixInverted[1][0] * dNdKsi[i] + jacobian.matrixInverted[1]
   [1] * dNdEta[i]
5
6   //MARK: - Interpolating temperature of current IP, from every other element's IPs.
7   t0 += temperatures_start[i] * shapeFunctionsVals[ipi][i]
8 }
```

2.5. Oblicz i dołącz elementy macierzy  $[H]$  oraz wektora  $\{P\}$  (całka po  $V$ )

**W pliku GlobalData, linijka 297:**

```
1 for i in 0..<4 {
2   for j in 0..<4 {
3     C_ij = c * ro * shapeFunctionsVals[ipi][i] * shapeFunctionsVals[ipi][j] * detJ
4     H_current[i][j] += k * (dNdx[i] * dNdx[j] + dNdy[i] * dNdy[j]) * detJ + (C_ij /
d_tau)
5     P_current[i] += (C_ij / d_tau) * t0
6   }
7 }
```

3.1. Oblicz wyznacznik macierzy Jakobiego 1D ( $\det J = \frac{x_2 - x_1}{2}$ ).

**W pliku GlobalData, linijka 314:**

```
1 detJ = 0.5 * sqrt(pow(surface.ND[0].x - surface.ND[1].x, 2) +
2                  pow(surface.ND[0].y - surface.ND[1].y, 2))
```

3.2.1. Oblicz i dołącz elementy macierzy  $[H]$  oraz wektora  $\{P\}$  (całka po  $S$ ).

**W pliku GlobalData, linijka 318:**

```
1 let shapeFunc = localElement.localSF[siid].shapeFunctionsVals
2 for i in 0..<2 {
3   for j in 0..<4 {
4     for k in 0..<4 {
5       H_current[j][k] += alpha * shapeFunc![i][j] * shapeFunc![i][k] * detJ
6     }
7     P_current[j] += alpha * t_ambient * shapeFunc![i][j] * detJ
8   }
9 }
```

4. Dokonaj agregacji lokalnej macierzy  $[H]$  oraz lokalnego wektora  $\{P\}$  do ich globalnych odpowiedników.

**W pliku GlobalData, linijka 330:**

```
1 for i in 0..<4 {
2   let first = element.ND[i].iid
3   for j in 0..<4 {
4     let next = element.ND[j].iid
5     H_global[first][next] += H_current[i][j]
6   }
7   P_global[first] += P_current[i]
8 }
```

Oprócz tego, obliczanie macierzy Jakobiego zostało zaimplementowane na podstawie obliczeń z pliku *Jakobian.pdf* na stronie prowadzącego i wygląda w następujący sposób:

**W pliku Jakobian.swift, linijka 35:**

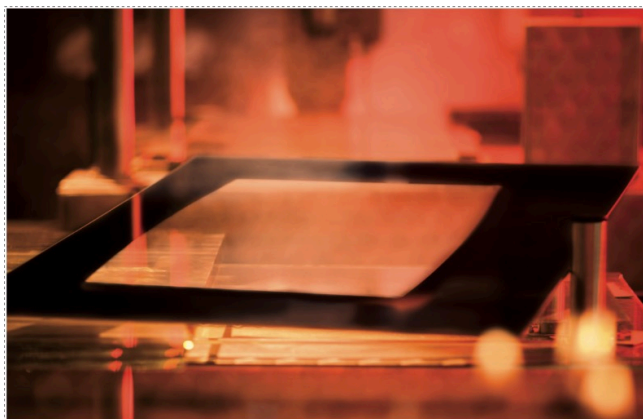
```
1 for i in 0..<4 {
2   matrix[0][0] += self.ips_dNdKsi[ipi][i] * xs[i] // dxdKsi
3   matrix[0][1] += self.ips_dNdKsi[ipi][i] * ys[i] // dydKsi
4   matrix[1][0] += self.ips_dNdEta[ipi][i] * xs[i] // dxdEta
5   matrix[1][1] += self.ips_dNdEta[ipi][i] * ys[i] // dydEta
6 }
```

Macierz Jakobiego w matematycznym zapisie:

$$\begin{bmatrix} \frac{\partial y}{\partial \eta} & -\frac{\partial y}{\partial \xi} \\ -\frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \xi} \end{bmatrix}$$

Wszelkie całkowanie w programie jest przeprowadzane za pomocą wspomnianej już kwadratury Gaussa.

# WYNIKI



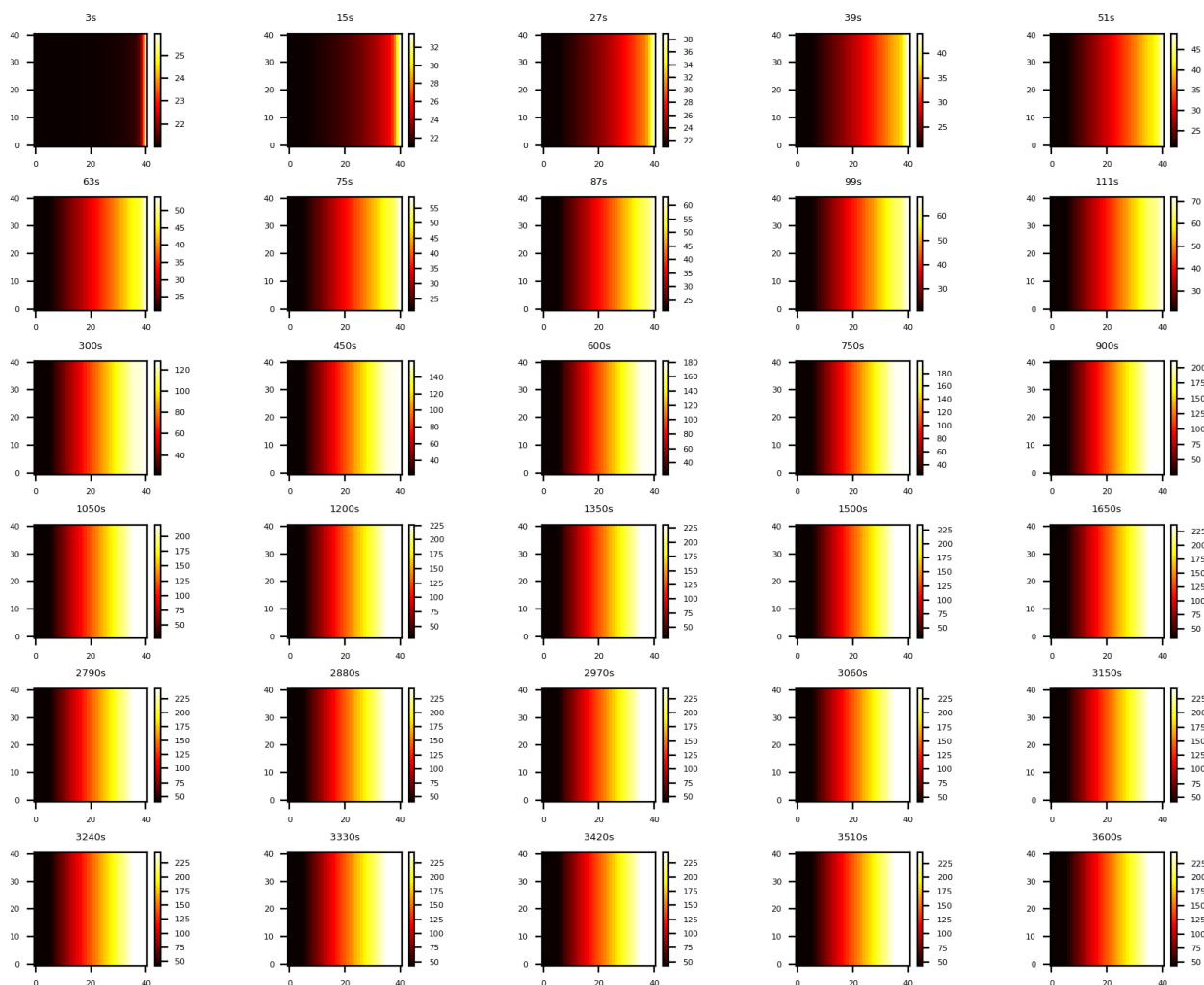
Thermally resistant oven door made with BOROFLOAT® 33.

## 1. Ogrzewanie z termoobiegiem

Symulacja została przeprowadzona przy zastosowaniu parametrów opisanych w sekcji *Założenia* i trwała **60 minut**. Gradient temperatur dla każdego kroku czasowego został zapisany w katalogu [Results/borofloat\\_simulations/fan\\_forced/csv/](#).

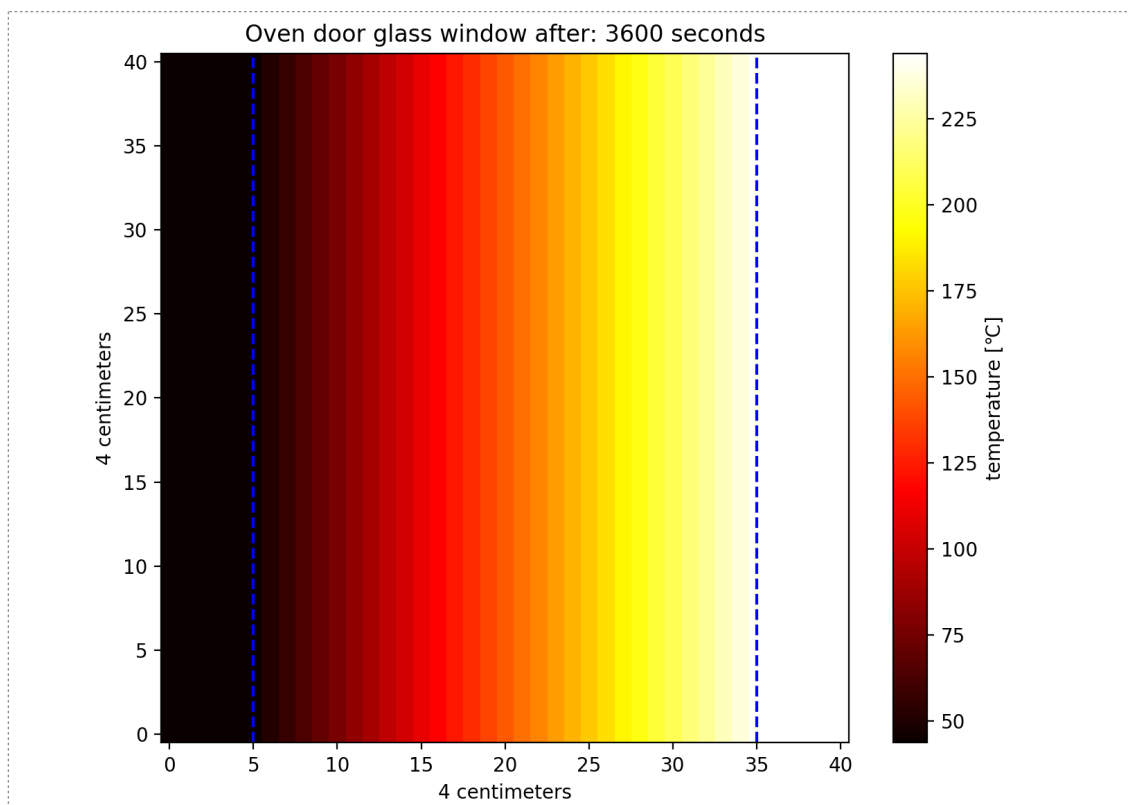
Z całości wyników zostały wybrane tylko niektóre, tak aby dostrzegalna była zmiana gradientu temperatur. Na ich podstawie sporządzone zostały mapy temperatur, które zostały umieszczone w jednym pliku [fan-forced\\_combined\\_1h.png](#).

W pliku tym umieszczone zostały gradienty temperatur dla fazy początkowej (0-111 sekund z krokiem 12 sekund), fazy środkowej (300 - 1650 sekund z krokiem 50 sekund) oraz fazy końcowej (2790 - 3600 sekund z krokiem 90 sekund). Każda faza to dwa rzędy:

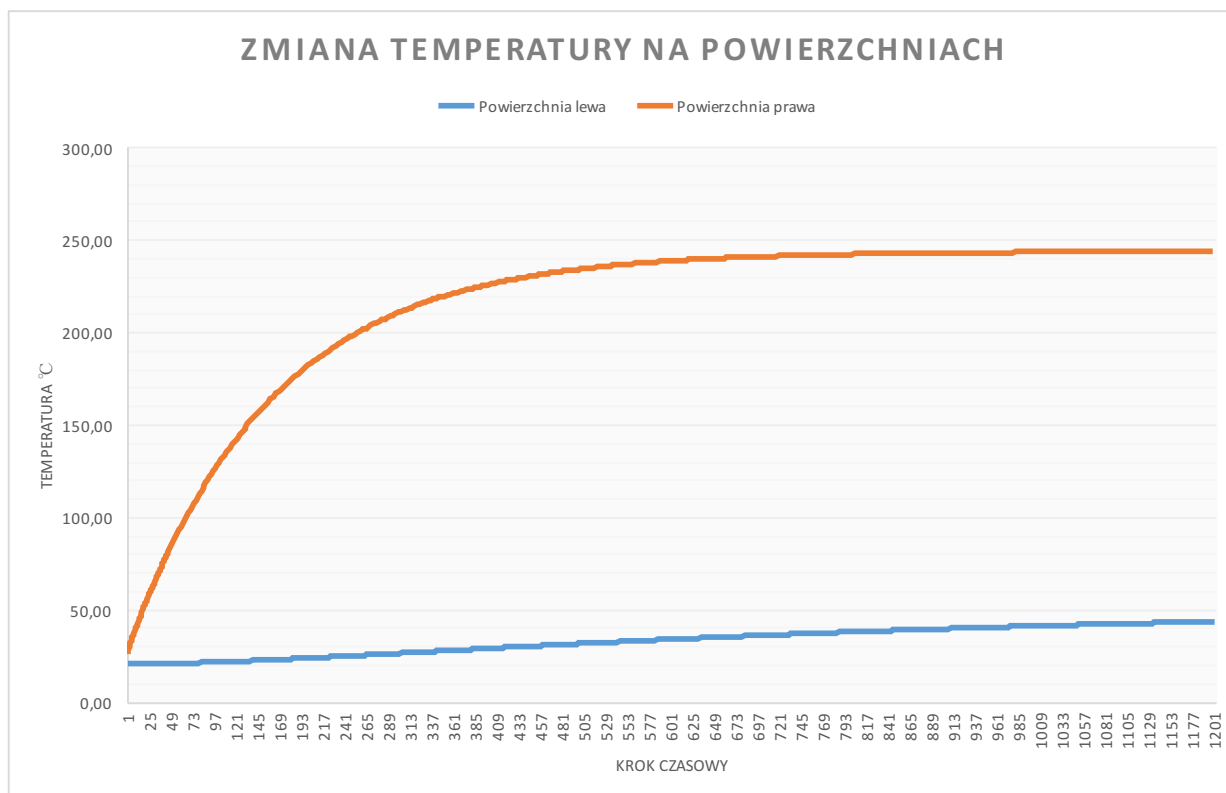




Poniżej znajduje się ostatni z wykresów w większym rozmiarze, gdzie niebieskie linie oznaczają granice materiałów:

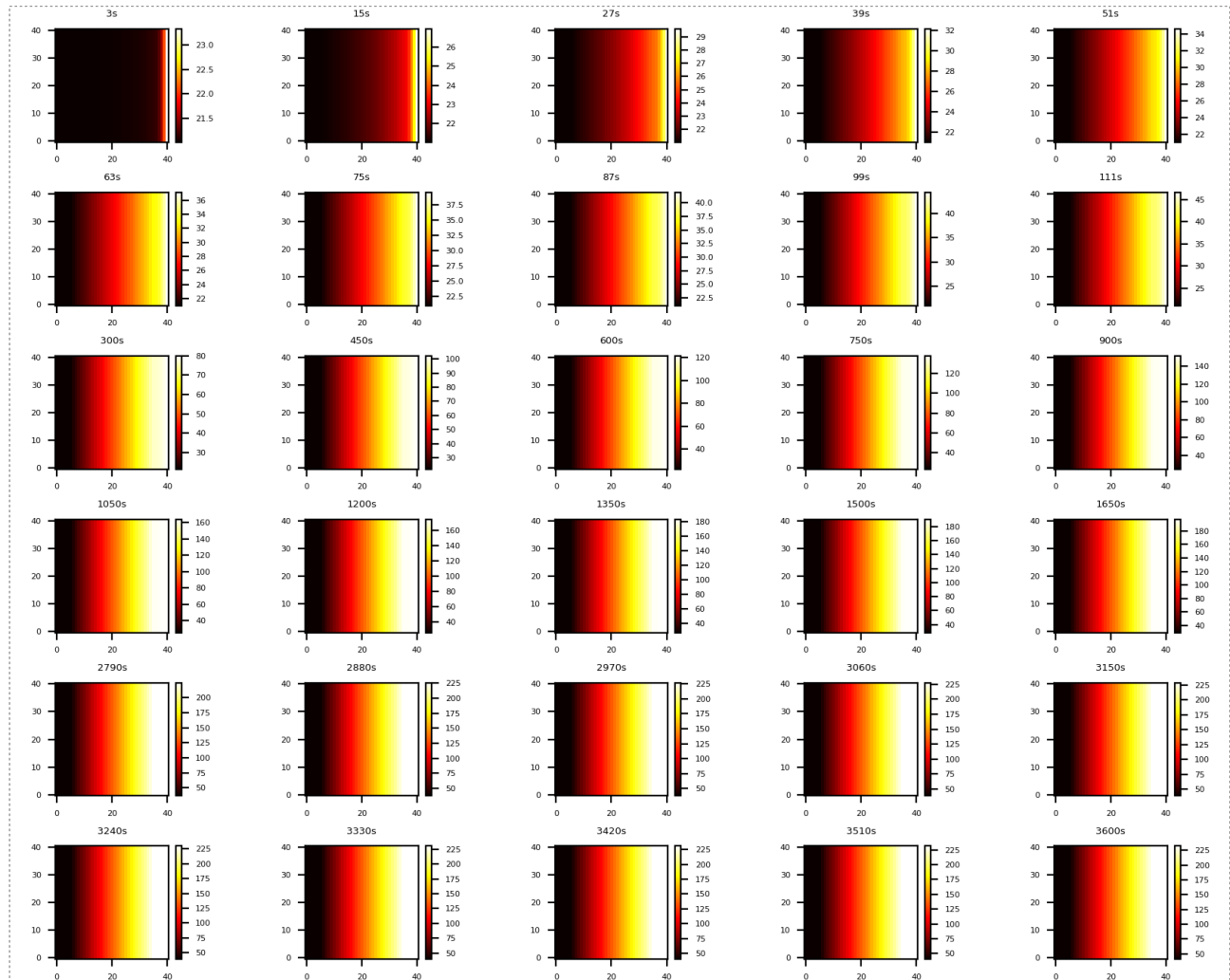


Na podstawie wyników sporządziłem także wykres zmiany temperatur na powierzchni lewej (od strony pokoju) i na powierzchni prawej (wewnątrz piekarnika) na przestrzeni całego procesu ogrzewania:

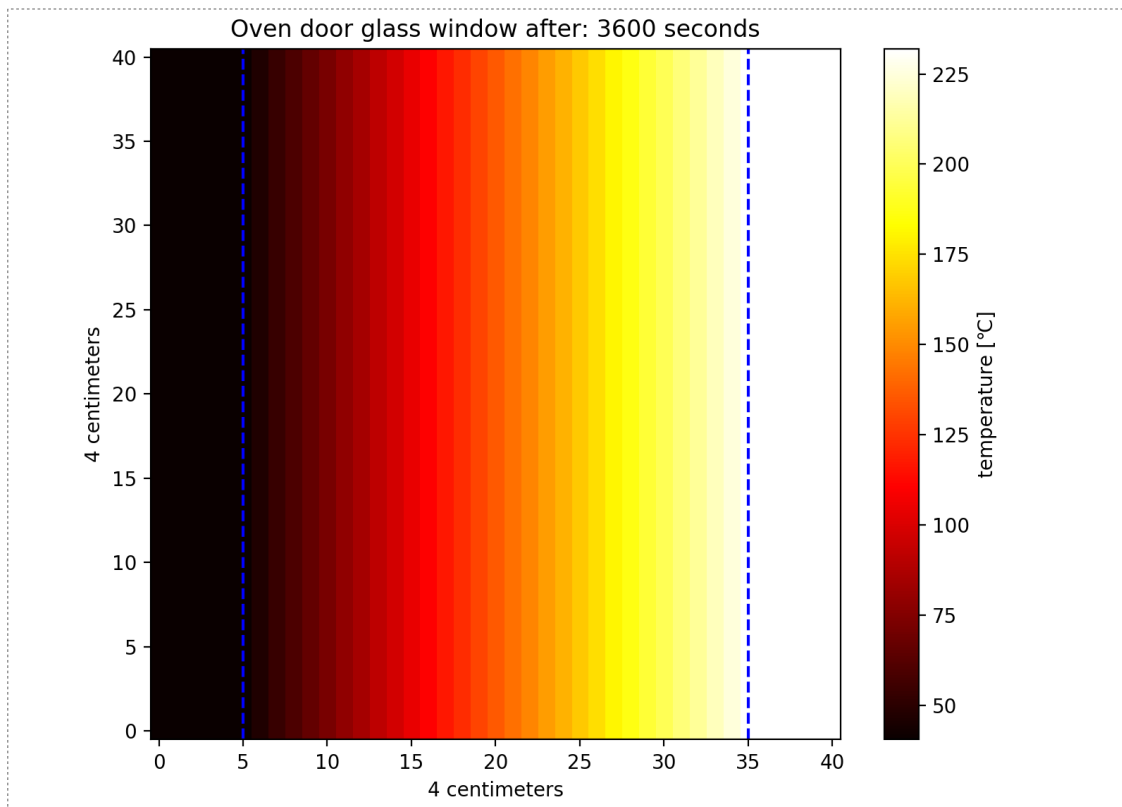


## 2. Ogrzewanie bez termoobiegu

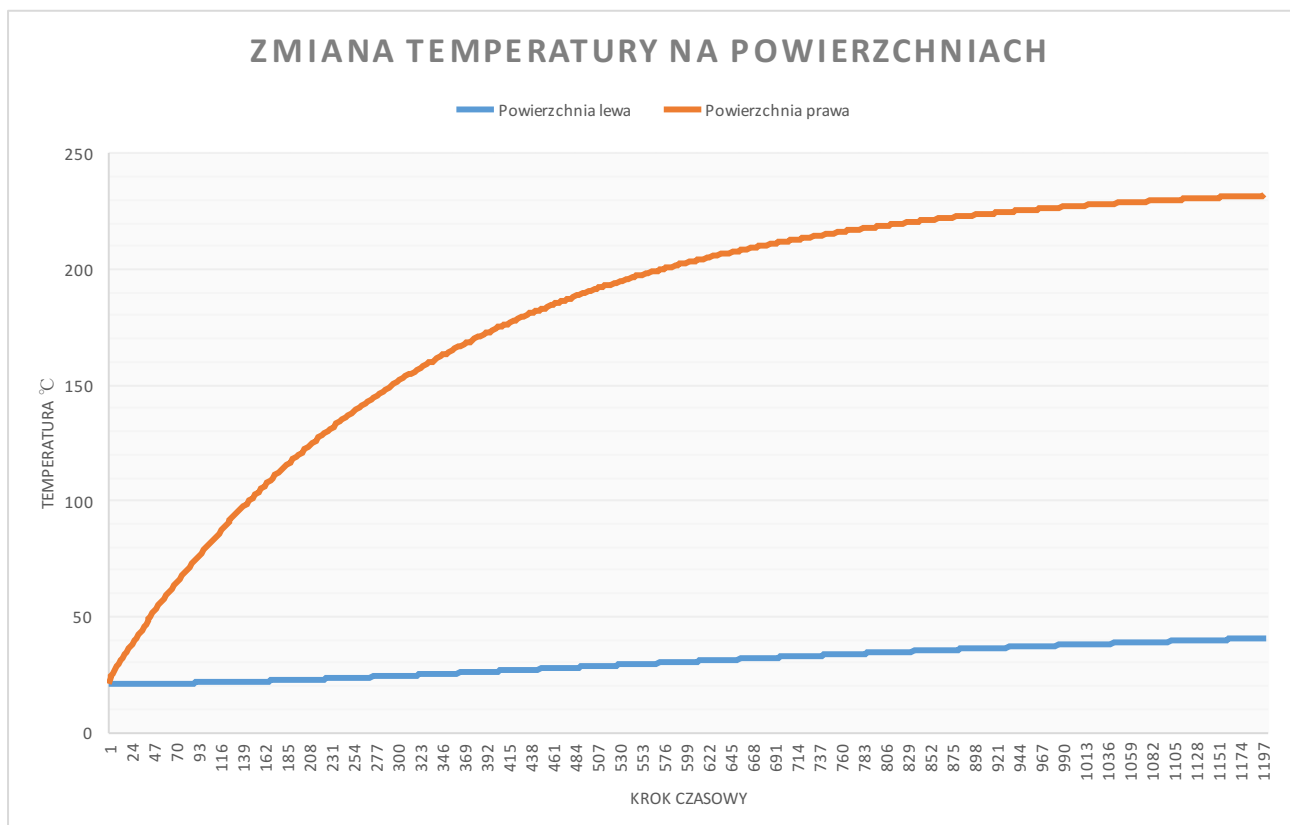
Ogrzewanie zostało przeprowadzone dokładnie w ten sam sposób co poprzednie z tym, że współczynnik konwekcji wynosił przyjęte wcześniej  $19,45 \frac{W}{m^2 K}$ . Tak jak poprzednio w pliku [normal\\_combined\\_1h.png](#) zawarte zostały wykresy gradientów temperatur siatki z podziałem na fazy (czasowe):



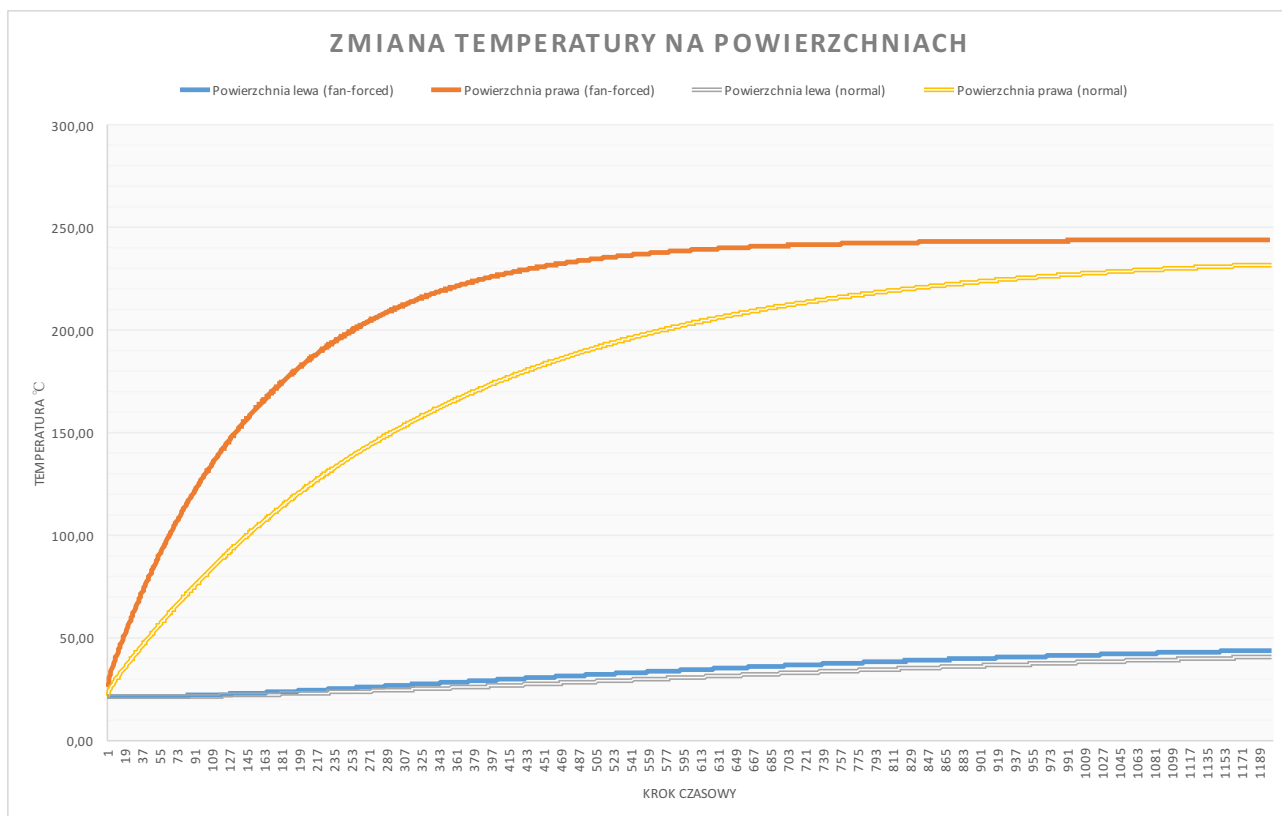
Również zapisany został ostatni wykres (po godzinie ogrzewania):



oraz wykres zmiany temperatury na powierzchniach okna piekarnika:



Dla porównania stworzyłem także wykres przedstawiający wszystkie zmiany temperatur na powierzchniach - dla ogrzewania z włączonym jak i wyłączonym termoobieganiem:



Oprócz tego tabela przedstawiająca końcowe temperatury na powierzchniach:

	Powierzchnia lewa [°C]	Powierzchnia prawa [°C]	$\Delta t$ [°C]
<b>z termoobieganiem</b>	43,80	244,01	200,21
<b>bez termoobiegu</b>	40,63	232,02	191,39

Zmiany temperatur w ciągu trwania procesu ogrzewania:

1. Z termoobieganiem:

$$\Delta t_{left} = 22,8^{\circ}C,$$

$$\Delta t_{right} = 223,01^{\circ}C$$

2. Bez termoobiegu:

$$\Delta t_{left} = 19,63^{\circ}C,$$

$$\Delta t_{right} = 211,02^{\circ}C$$

## SPOSTRZEŻENIA

1. Zmiana temperatury w fazie początkowej przebiega dużo szybciej niż w dalszej części trwania wymiany ciepła.
2. Temperatura powierzchni lewej (od strony pokoju) zmienia się prawie liniowo.
3. Temperatura powierzchni prawej (wewnątrz piekarnika) zmienia się logarytmicznie.
4. Na końcowym wykresie gradientu temperatur uwidaczniają się granice między warstwami układu.
5. Różnice temperatur na węzłach szyby lewej są minimalne, natomiast między węzłami argonu są znacznie większe.
6. Wymiana ciepła z włączonym termoobiegem postępuje znacznie szybciej, niż przy wyłączonym termoobiegu.
7. Temperatura na powierzchni lewej (od strony pokoju) przy włączonym termoobiegu w ciągu godziny zmieniła się o **TODO**
8. Temperatura na powierzchni lewej (wewnątrz piekarnika) przy wyłączonym termoobiegu w ciągu godziny zmieniła się o **TODO**

## WNIOSKI

1. ddddd
2. ssd
3. did