

Raport: Projekt grupowy „MesTemp”

Autorzy:

Krzysztof Mielczarek

Mateusz Ostrowski

Jędrzej Graczykowski

Spis treści

1. Cele projektu.....	2
2. Wstęp teoretyczny.....	2
2.1 Przepływ ciepła.....	2
2.2 Warunki brzegowe.....	2
2.3 Metoda elementów skończonych.....	3
2.4 Macierze lokalne.....	4
2.5 Rozwiązanie układu równań algebraicznych.....	4
3. Rozwiązanie problemu.....	6
3.1 Założenia programu.....	6
3.2 Wczytanie danych wejściowych.....	6
3.3 Generacja siatki i jej zagęszczanie.....	7
3.4 Obliczenie rozkładu temperatur.....	9
3.5 Graficzne przedstawienie wyników obliczeń.....	10
3.6 Zapis wyników i plik wyjściowy.....	11
4. Przykładowe wyniki.....	13
5. Wnioski.....	15
6. Bibliografia.....	16

1. Cele projektu

1. Zapoznanie się z metodą elementów skończonych.
2. Zbadanie przepływu ciepła i rozkładu temperatur.
3. Wykonanie programu komputerowego w języku C++ (aplikacja MFC), na platformie Microsoft Visual Studio 2019 obliczającego rozkład temperatur w dwuwymiarowym ośrodku, składającym się z różnych materiałów, w którym występować mogą lokalne źródła ciepła.

2. Wstęp teoretyczny

2.1 Przepływ ciepła

Przepływ ciepła jest wynikiem różnic temperatur. Ciepło przepływa z ciała o wyższej temperaturze do ciała o niższej temperaturze. Zatem warunkiem przepływu ciepła jest gradient temperatury. Problem badany w tym projekcie dotyczy przewodzenia ciepła w ciałach stałych, czyli transportu energii za pomocą drgań sieci krystalicznej.

Zbadana zostanie gęstość strumienia ciepła:

$$q = \frac{d\dot{Q}}{dA} \quad (1)$$

gdzie: \dot{Q} - strumień ciepła, A - powierzchnia przepływu.

Powyższa wielkość użyta jest w prawie Fouriera dla ośrodka dwuwymiarowego:

$$q = -\lambda \nabla T \quad (2)$$

gdzie: λ - przewodność cieplna, zależna od danego materiału.

2.2 Warunki brzegowe

Powyższe równanie różniczkowe umożliwia rozwiązanie zagadnienia rozkładu temperatury w danym ośrodku. Równanie to musi być uzupełnione o warunki brzegowe. Wyróżnia się trzy rodzaje takich warunków:

- I rodzaju (warunki Dirichleta) – opisują wartość badanej funkcji na brzegach obszaru.
- II rodzaju (warunki Neumanna) – opisują interakcję z brzegiem obszaru, np. przepływ ciepła.
- III rodzaju (warunki Fouriera, Robina) – połączenie warunków I i II rodzaju.

W badanym problemie użyte zostały tylko warunki I rodzaju.

2.3 Metoda elementów skończonych

Aby rozwiązać zagadnienie przepływu ciepła i rozkładu temperatur posłużono się metodą elementów skończonych. Metoda ta składa się z:

- podziału badanego obszaru na elementy (generacja siatki),
- matematycznego sformułowania metody dla analizowanego warunku brzegowego, w obszarze jednego elementu,
- wyboru funkcji interpolujących rozkład temperatury w danym elemencie,
- wyznaczenia układu równań algebraicznych dla problemu ustalonego lub układu równań różniczkowych zwyczajnych dla zagadnienia nieustalonego dla jednego elementu,
- dodawanie układów równań dla poszczególnych elementów w celu utworzenia jednego globalnego układu równań na wartości pola w węzłach siatki dla całego analizowanego obszaru (agregacja),
- uwzględnienie parametrów występujących w warunkach brzegowych w globalnym układzie równań,
- rozwiązanie układu równań algebraicznych w przypadku zagadnienia ustalonego lub układu równań różniczkowych zwyczajnych w przypadku zagadnienia nieustalonego,
- obliczenie strumieni gęstości pola i innych wielkości wtórnych oraz graficzna prezentacja wyników obliczeń.

Produktem podziału całości na elementy, utworzenia lokalnych macierzy oraz ich agregacji jest:

$$K T = P \quad (3)$$

gdzie: K – macierz utworzona z wszystkich macierzy lokalnych (macierz sztywności), T – wektor rozwiązań, np. wektor temperatur, P – wektor obciążeń, np. wektor przewodności cieplnej.

Rozwiązaniem problemu rozkładu temperatur jest zatem:

$$T = K^{-1} P \quad (4)$$

2.4 Macierze lokalne

Do utworzenia macierzy lokalnej w danym elemencie posłużono się funkcjami kształtu dla elementu prostokątnego:

$$\psi_1 = \frac{1}{4}(1+\xi)(1+\eta) \quad (5)$$

$$\psi_2 = \frac{1}{4}(1-\xi)(1+\eta) \quad (6)$$

$$\psi_3 = \frac{1}{4}(1-\xi)(1-\eta) \quad (7)$$

$$\psi_4 = \frac{1}{4}(1+\xi)(1-\eta) \quad (8)$$

gdzie: ξ i η – współrzędne znormalizowane, czyli przeniesione do lokalnego układu współrzędnych pojedynczego elementu tak, by ów element był kwadratem o przeciwległych rogach w punktach $(-1;-1)$ i $(1;1)$

Dla naszego zagadnienia następujące wzory są prawdziwe:

$$\text{div}(\lambda \text{ grad } T) = -\dot{q}_v \quad (9)$$

$$k_{ij}^e = \int_{\Omega_e} \lambda \left(\frac{\partial \psi_i}{\partial x} \frac{\partial \psi_j}{\partial x} + \frac{\partial \psi_i}{\partial y} \frac{\partial \psi_j}{\partial y} \right) dV \quad (10)$$

$$p_i^e = \int_{\Omega_e} \dot{q}_v \psi_i dV \quad (11)$$

2.5 Rozwiązanie układu równań algebraicznych

Rozwiązanie układu równań algebraicznych nastąpiło przy pomocy metody gradientów sprzężonych. Algorytm rozwiązuje układ równań poprzez szukanie ekstremum związanej z nim formy kwadratowej. Dla każdego kroku liczona jest jego długość α i kierunek \mathbf{p} , obliczany jako liniowa kombinacja wektora gradientu \mathbf{r} i poprzedniego wektora kierunku; przy współczynniku liniowym β . Zachodzą zatem wzory:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k \quad (12)$$

$$\alpha_k = \frac{\mathbf{r}_k^T \mathbf{r}_k}{\mathbf{p}_k^T \mathbf{A} \mathbf{p}_k} \quad (13)$$

$$\mathbf{p}_{k+1} = \mathbf{r}_{k+1} + \beta_{k+1} \mathbf{p}_k \quad (14)$$

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k \mathbf{A} \mathbf{p}_k$$

$$\beta_{k+1} = \frac{\mathbf{r}_{k+1}^T \mathbf{r}_{k+1}}{\mathbf{r}_k^T \mathbf{r}_k} \quad (15)$$

Metoda ta jest szybka ze względu na dokładne wyznaczanie kierunku poszukiwań wykorzystujące również kierunek poprzedni. Obliczenia w dokładnej matematyce powinny zająć tyle ruchów, ile mamy równań, w praktyce zaś prowadzone są do momentu uzyskania zadanej dokładności.

3. Rozwiązanie problemu

3.1 Założenia programu

Program został napisany w języku C++ jako aplikacja MFC typu Single Document. Taki typ charakteryzuje się tym, że nie wykonuje kilku zadań jednocześnie. Następne zadanie zaczyna, gdy poprzednie zostanie skończone. Aplikacja MFC składa się przede wszystkim z klas: doc i view. Klasa doc odpowiedzialna jest za obliczenia i sterowanie programem. Klasa view odpowiedzialna jest za wyświetlanie treści użytkownikowi. Program MesTemp jest zaprogramowany obiektowo.

Zadaniami programu są:

- wczytanie danych wejściowych,
- wygenerowanie siatki i jej zagęszczanie,
- obliczenie rozkładu temperatur na podstawie danych wejściowych i wygenerowanej siatki,
- możliwość uzyskania informacji o danym obszarze,
- zapisanie wyników do pliku.

3.2 Wczytanie danych wejściowych

Dane wejściowe wprowadzane są do programu za pomocą pliku tekstowego .txt. Każdy plik ma określoną strukturę (Rys. 1.)

```
liczba obszarów:
16
skala:
1
czy pr: nr:      x1[mm]: y1[mm]: x4[mm]: y4[mm]: cz  źr:  moc[W/m2]:przew_x:przew_y:[W/K*m] materiał:
1      0      0      0      298      900      0      0      0      10      30      miedz
1      1      298      0      300      500      0      0      0      50      50      miedzxx
1      2      298      500      300      505      1      500      30      30      male_zrodlo
1      3      298      505      300      900      0      0      250      25      pasek2
1      4      300      0      600      400      0      0      100      100      miedz
1      5      600      0      700      200      0      0      50      50      miedz
1      6      700      0      900      300      0      0      100      100      miedz
1      7      600      200      700      300      0      0      100      100      miedz
1      8      600      300      900      500      0      0      100      100      miedz
1      9      300      400      400      500      1      250      100      100      miedz
1      10     400      400      600      500      0      0      50      50      miedz
1      11     300      500      700      600      0      0      100      100      miedz
1      12     700      500      900      900      0      0      100      100      miedz
1      13     300      600      600      900      0      0      100      100      miedz
1      14     600      600      700      700      1      250      100      100      miedz
1      15     600      700      700      900      0      0      50      50      miedz
warunki brzegowe I rodzaju ilość:
2
x1:      y1:      x2:      y2:      T[K]:
0      0      0      900      250
900      0      900      300      25
```

Rys. 1. Przykładowy plik z danymi wejściowymi.

Objaśnienia:

- liczba obszarów – liczba ta wprowadzana jest, aby przekazać do programu granice pętli wczytujących dane,
- skala – skala badanego obszaru względem rzeczywistości,
- czy pr. – znacznik informujący czy dany obszar jest prostokątny,
- nr – numer porządkowy obszaru,
- x1, y1, x4, y4 – współrzędne przeciwległych wierzchołków wprowadzanych obszarów,
- czy źr. – znacznik informujący czy w danym obszarze jest źródło ciepła,
- moc – [W/m²] gęstość mocy grzewczej źródła ciepła,
- przew_x, przew_y [W/m K] – przewodność cieplna w kierunku poziomym i pionowym,
- materiał – nazwa materiału, z którego wykonany jest obszar.

Program zapisuje obszar jako obiekt, którego atrybutami są wszystkie powyższe dane, poza liczbą obszarów i skalą. Te dwie informacje zapisywane są osobno, na początku. Następnie z podanych obiektów tworzony jest wektor, do którego można mieć później dostęp.

Kolejne informacje w pliku wejściowym:

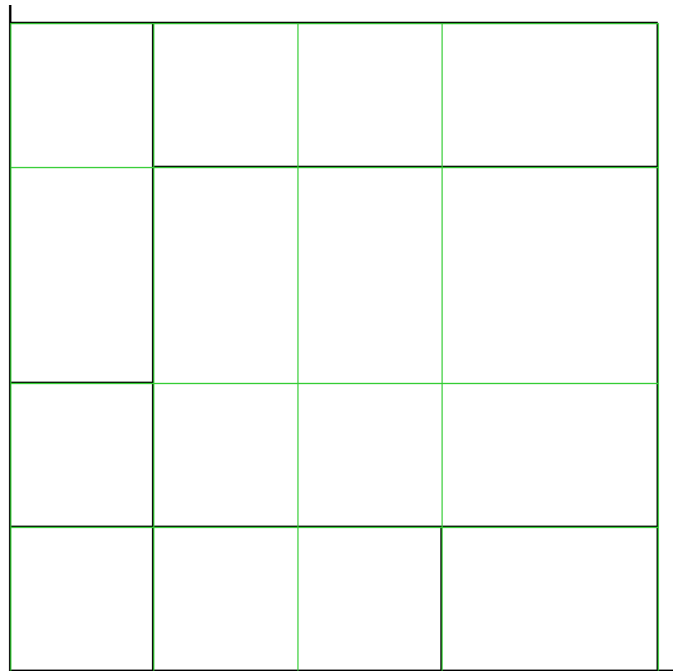
- liczba warunków brzegowych I rodzaju,
- współrzędne końców warunku
- wartość temperatury dla danego warunku

3.3 Generacja siatki i jej zagęszczanie

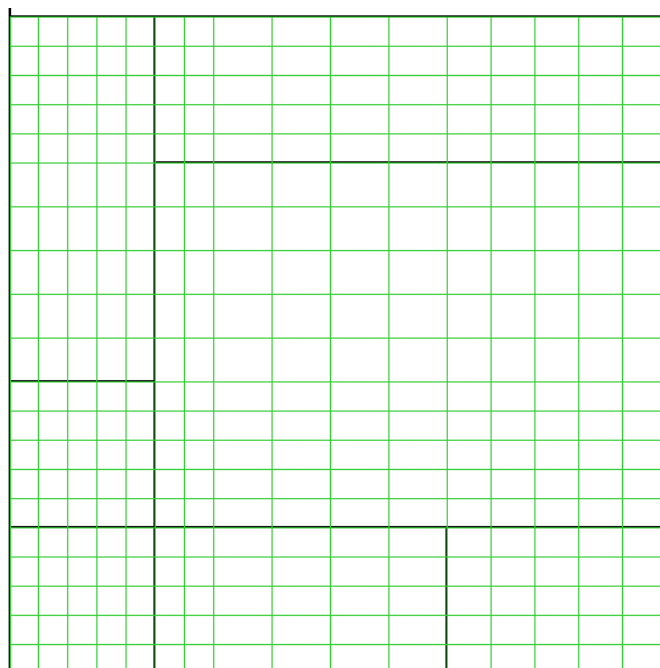
Siatka w programie zapisana jest jako dwa wektory: węzły_x i węzły_y. Każdy wektor zawiera zatem współrzędne węzłów siatki w poziomie lub pionie. Takie rozwiązanie jest możliwe, ponieważ analizowany obszar składa się z elementów prostokątnych.

Pierwsza siatka generowana jest na podstawie granic wprowadzonych obszarów. Następnie dodana została możliwość zagęszczania. Istnieje możliwość ustalenia jak mocno zagęścić siatkę poprzez podanie ile ma powstać nowych elementów na miejscu starego. Dodatkowo program sprawdza czy zagęszczenie siatki nie wygenerowało elementów, które ze sobą sąsiadują i różnią się od siebie znacząco pod względem rozmiarów. Jeśli tak jest, to następuje lokalne zagęszczenie niwelujące te różnice.

Wygenerowana siatka zostaje narysowana w programie (Rys. 2.)



Rys. 2. Niezagęszczona siatka wygenerowana na granicach obszarów.



Rys. 3. Siatka po zagęszczeniu.

3.4 Obliczenie rozkładu temperatur

Dla każdego elementu tworzona jest macierz lokalna według wzoru:

$$\mathbf{k}^e = \frac{\lambda_x b}{6 a} \begin{bmatrix} 2 & -2 & -1 & 1 \\ -2 & 2 & 1 & -1 \\ -1 & 1 & 2 & -2 \\ 1 & -1 & -2 & 2 \end{bmatrix} + \frac{\lambda_y a}{6 b} \begin{bmatrix} 2 & 1 & -1 & -2 \\ 1 & 2 & -2 & -1 \\ -1 & -2 & 2 & 1 \\ -2 & -1 & 1 & 2 \end{bmatrix} \quad (16)$$

gdzie:

- a – szerokość elementu,
- b – wysokość elementu,
- λ_x – przewodność cieplna w kierunku x
- λ_y – przewodność cieplna w kierunku y

Następnie przez funkcję agregującą z macierzy lokalnych tworzona jest globalna macierz sztywności. Macierz ta jest pasmowa i rzadka, więc do efektywnego zapisywania jej wykorzystuje się trzy wektory:

- A – zawiera kolejne niezerowe wartości macierzy,
- $coln$ – indeksy kolumn odpowiadające każdemu z niezerowych elementów,
- $proW$ – indeksy niezerowych elementów (indeksy z macierzy A) dla pierwszego elementu w każdym wierszu macierzy.

Dzięki takiej formie zapisu oszczędzamy pamięć komputera i nie uwzględniamy olbrzymiej ilości zer w obliczeniach.

Z lokalnych danych poszczególnych elementów tworzony jest również globalny wektor ciepła przepływającego przez węzły sieci, wg wzoru:

$$p^e = \frac{QA}{4} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad (17)$$

gdzie:

- Q – gęstość mocy w danym obszarze
- A – pole powierzchni elementu

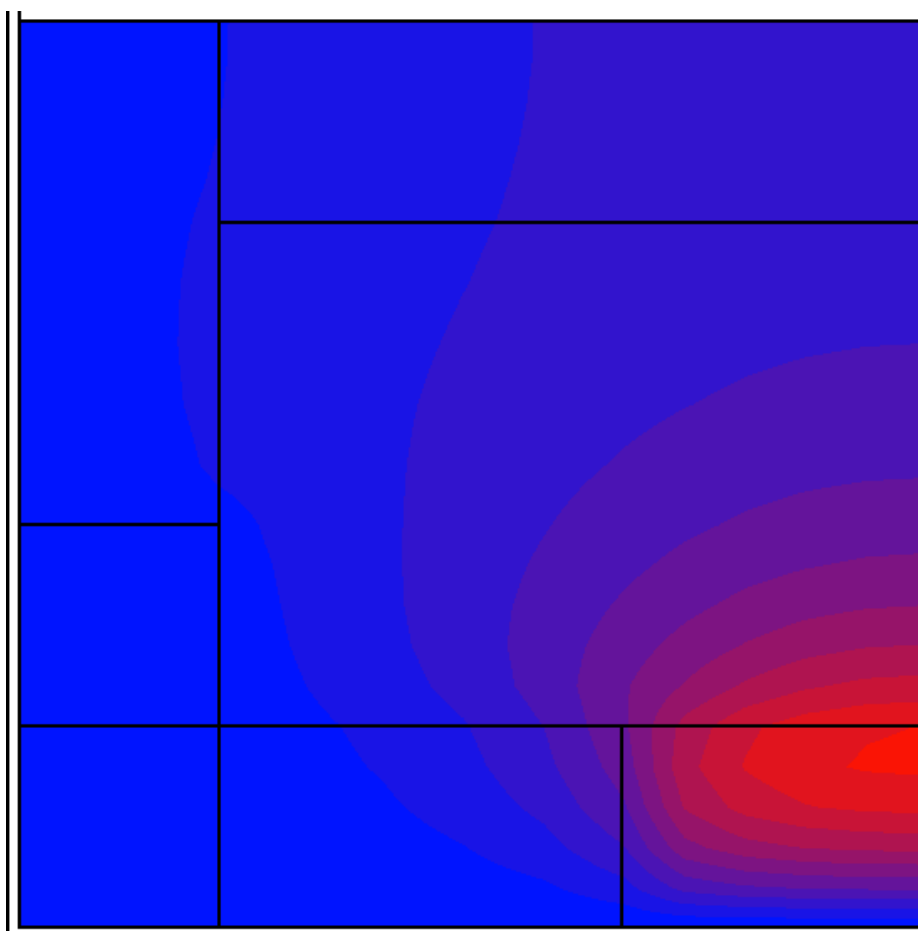
Następnie macierz globalną i wektor ciepła modyfikowane są o zadane warunki brzegowe I rodzaju. Uwzględniając warunki brzegowe posłużono się metodą Payne’a i Ironsa, czyli pomnożeniu odpowiedniego wyrazu macierzy sztywności przez dużą liczbę (10^{15}) i zmodyfikowaniu

odpowiedniego wyrazu macierzy prawych stron. Co w efekcie po wymnożeniu macierzy sztywności i macierzy rozwiązań daje w przybliżeniu same warunki brzegowe.

Układ równań rozwiązywany jest iteracyjną metodą gradientów sprzężonych, która wykazuje dużą użyteczność dla dużych, rzadkich macierzy. Geometria powstałego układu równań spełnia wymogi użycia tego algorytmu (macierz symetryczna i dodatnio określona). Metoda gradientów sprzężonych została opisana we wstępie teoretycznym.

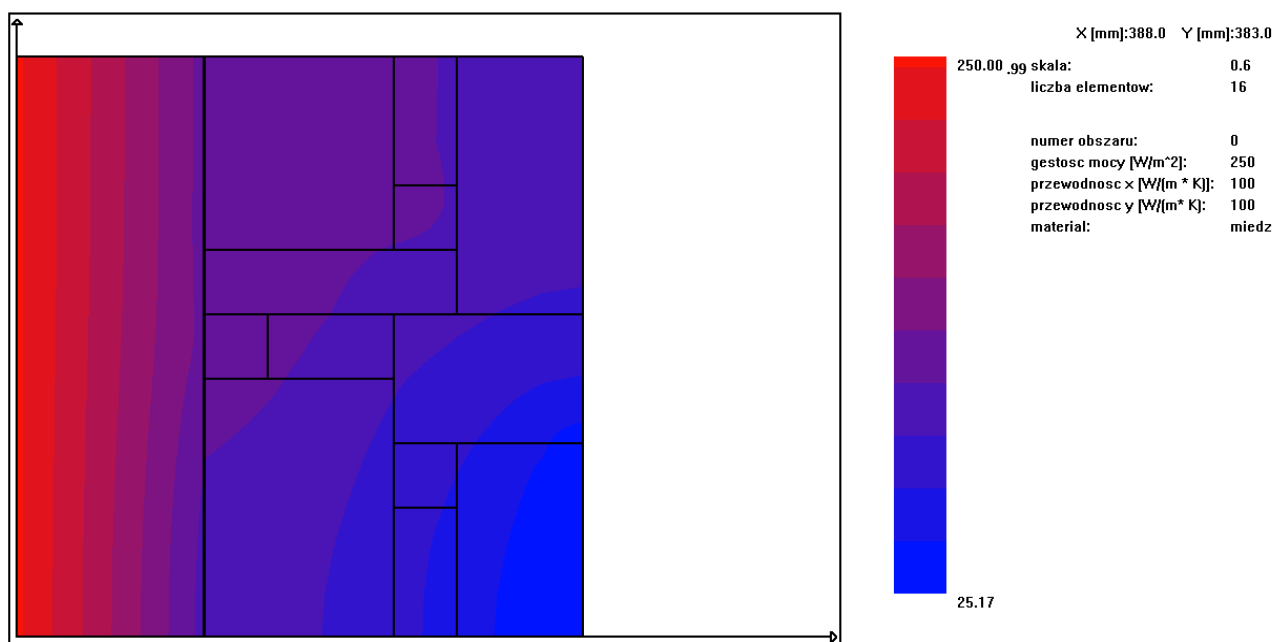
3.5 Graficzne przedstawienie wyników obliczeń

Program rysuje rozkład temperatur sprawdzając piksel po pikselu wartość obliczonej temperatury i koloruje go według odpowiedniej skali. Na izotermy zostają nałożone jeszcze raz granice obszarów, ale linie siatki nie są widoczne.



Rys.4. Narysowany rozkład temperatury w badanym obszarze.

Obok rysunku badanego obszaru narysowana jest także skala informująca o znaczeniu barw. Skala podzielona jest na 11 przedziałów zawartych między temperaturą minimalną i maksymalną.



Rys. 5 Narysowany rozkład temperatury w badanym obszarze wraz ze skalą.

Kolejnym rozwiązaniem jest fakt, że obok skali wypunktowane są współrzędne kliknięcia myszką wraz z podstawowymi informacjami na temat danego obszaru.

W programie istnieje możliwość przybliżania i oddalania rysunku we wskazanym przez użytkownika miejscu.

3.6 Zapis wyników i plik wyjściowy

Plik wyjściowy zawiera informacje o ilości obszarów i węzłów. Kolejną rzeczą zawartą w pliku jest informacja o każdym z węzłów. Dla każdego z nich podane są: numer, współrzędne, wartość temperatury, a także dla obszaru, w którym węzeł się znajduje podane są numer obszaru, przewodność, moc źródła oraz materiał. Wszystkie dane są zapisane w pliku tekstowym. Na koniec pliku podana jest temperatura maksymalna oraz minimalna wraz z informacją, dla których węzłów te ekstremalne temperatury wystąpiły.

Wyniki dla 7 obszarów materiałowych, podzielonych na siatkę o 117 węzłach (9 w osi X i 13 w osi Y):

Nr węzła	X[mm]	Y[mm]	Temperatura[K]	Nr obszaru	Przewodność X[W/(m*K)]	Przewodność Y[W/(m*K)]	Moc źródła[W/m^2]
0	0	0	350	0	100	100	0
Nr węzła	X[mm]	Y[mm]	Temperatura[K]	Nr obszaru	Przewodność X[W/(m*K)]	Przewodność Y[W/(m*K)]	Moc źródła[W/m^2]
1	50	0	350	0	100	100	0
Nr węzła	X[mm]	Y[mm]	Temperatura[K]	Nr obszaru	Przewodność X[W/(m*K)]	Przewodność Y[W/(m*K)]	Moc źródła[W/m^2]
2	141.667	0	350	1	100	100	0
Nr węzła	X[mm]	Y[mm]	Temperatura[K]	Nr obszaru	Przewodność X[W/(m*K)]	Przewodność Y[W/(m*K)]	Moc źródła[W/m^2]
3	233.333	0	350	1	100	100	0
Nr węzła	X[mm]	Y[mm]	Temperatura[K]	Nr obszaru	Przewodność X[W/(m*K)]	Przewodność Y[W/(m*K)]	Moc źródła[W/m^2]
4	416.667	0	350	1	100	100	0
Nr węzła	X[mm]	Y[mm]	Temperatura[K]	Nr obszaru	Przewodność X[W/(m*K)]	Przewodność Y[W/(m*K)]	Moc źródła[W/m^2]
5	600	0	350	1	100	100	0
Nr węzła	X[mm]	Y[mm]	Temperatura[K]	Nr obszaru	Przewodność X[W/(m*K)]	Przewodność Y[W/(m*K)]	Moc źródła[W/m^2]
6	700	0	350	5	100	100	0
Nr węzła	X[mm]	Y[mm]	Temperatura[K]	Nr obszaru	Przewodność X[W/(m*K)]	Przewodność Y[W/(m*K)]	Moc źródła[W/m^2]
7	800	0	350	5	100	100	0
Nr węzła	X[mm]	Y[mm]	Temperatura[K]	Nr obszaru	Przewodność X[W/(m*K)]	Przewodność Y[W/(m*K)]	Moc źródła[W/m^2]
8	900	0	50	5	100	100	0
Nr węzła	X[mm]	Y[mm]	Temperatura[K]	Nr obszaru	Przewodność X[W/(m*K)]	Przewodność Y[W/(m*K)]	Moc źródła[W/m^2]
9	0	66.6667	331.547	0	100	100	0
Nr węzła	X[mm]	Y[mm]	Temperatura[K]	Nr obszaru	Przewodność X[W/(m*K)]	Przewodność Y[W/(m*K)]	Moc źródła[W/m^2]
10	50	66.6667	331.452	0	100	100	0

Rys. 6 Początek przykładowego pliku wyjściowego.

Nr węzła	X[mm]	Y[mm]	Temperatura[K]	Nr obszaru	Przewodność X[W/(m*K)]	Przewodność Y[W/(m*K)]	Moc źródła[W/m^2]
114	700	900	103.592	3	100	100	0
Nr węzła	X[mm]	Y[mm]	Temperatura[K]	Nr obszaru	Przewodność X[W/(m*K)]	Przewodność Y[W/(m*K)]	Moc źródła[W/m^2]
115	800	900	77.3451	3	100	100	0
Nr węzła	X[mm]	Y[mm]	Temperatura[K]	Nr obszaru	Przewodność X[W/(m*K)]	Przewodność Y[W/(m*K)]	Moc źródła[W/m^2]
116	900	900	50	3	100	100	0

Temperatura maksymalna [K]: 350. Wystąpiła dla węzłów nr: 1.

Temperatura minimalna [K]: 50. Wystąpiła dla węzłów nr: 53.

Rys. 7 Koniec przykładowego pliku wyjściowego.

4. Przykładowe wyniki

Poniżej zostało pokazane przykładowe rozwiązanie. Dane wejściowe zostały pokazane na rysunku (Rys. 8)

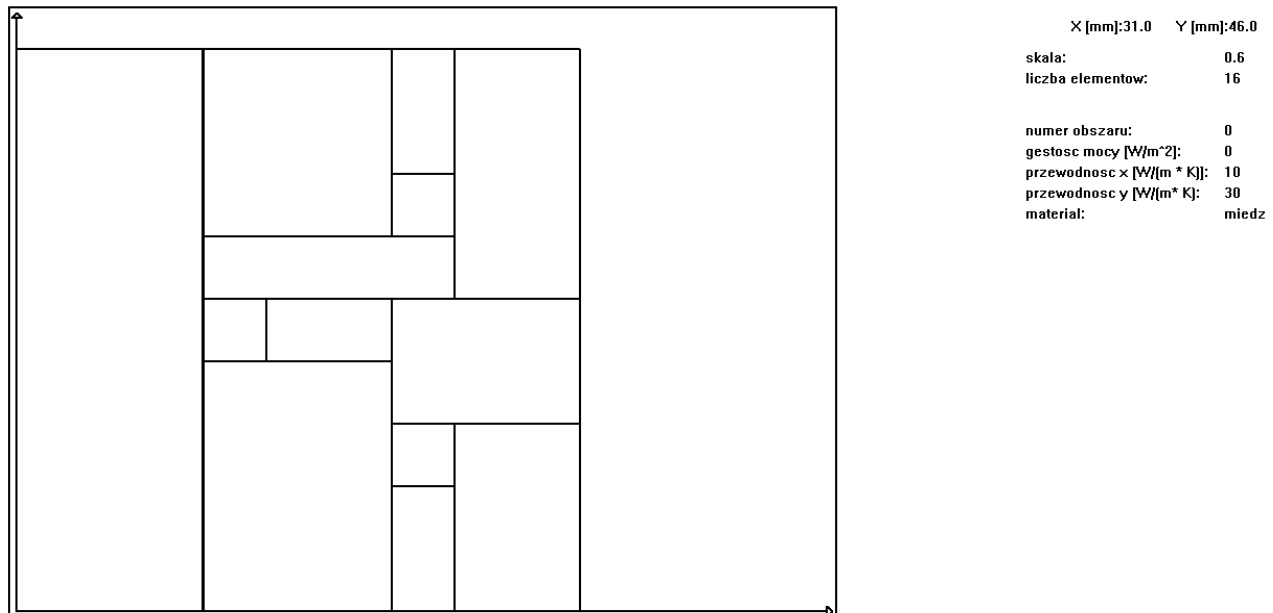
```

liczba obszarów:
16
skala:
1
czy pr: nr:      x1[mm]: y1[mm]: x4[mm]:  y4[mm]: cz  źr:  moc[W/m2]:przew_x:przew_y:[W/K*m] materiał:
1      0        0      0      298    900    0    0    0    10    30    miedz
1      1        298    0      300    500    0    0    0    50    50    miedzxx
1      2        298    500    300    505    1    500    30    30    male_zrodlo
1      3        298    505    300    900    0    0    0    250    25    pasek2
1      4        300    0      600    400    0    0    0    100    100    miedz
1      5        600    0      700    200    0    0    0    50    50    miedz
1      6        700    0      900    300    0    0    0    100    100    miedz
1      7        600    200    700    300    0    0    0    100    100    miedz
1      8        600    300    900    500    0    0    0    100    100    miedz
1      9        300    400    400    500    1    250    100    100    miedz
1     10        400    400    600    500    0    0    0    50    50    miedz
1     11        300    500    700    600    0    0    0    100    100    miedz
1     12        700    500    900    900    0    0    0    100    100    miedz
1     13        300    600    600    900    0    0    0    100    100    miedz
1     14        600    600    700    700    1    250    100    100    miedz
1     15        600    700    700    900    0    0    0    50    50    miedz
warunki brzegowe I rodzaju ilość:
2
x1:    y1:    x2:    y2:    T[K]:
0      0      0      900    250
900    0      900    300    25

```

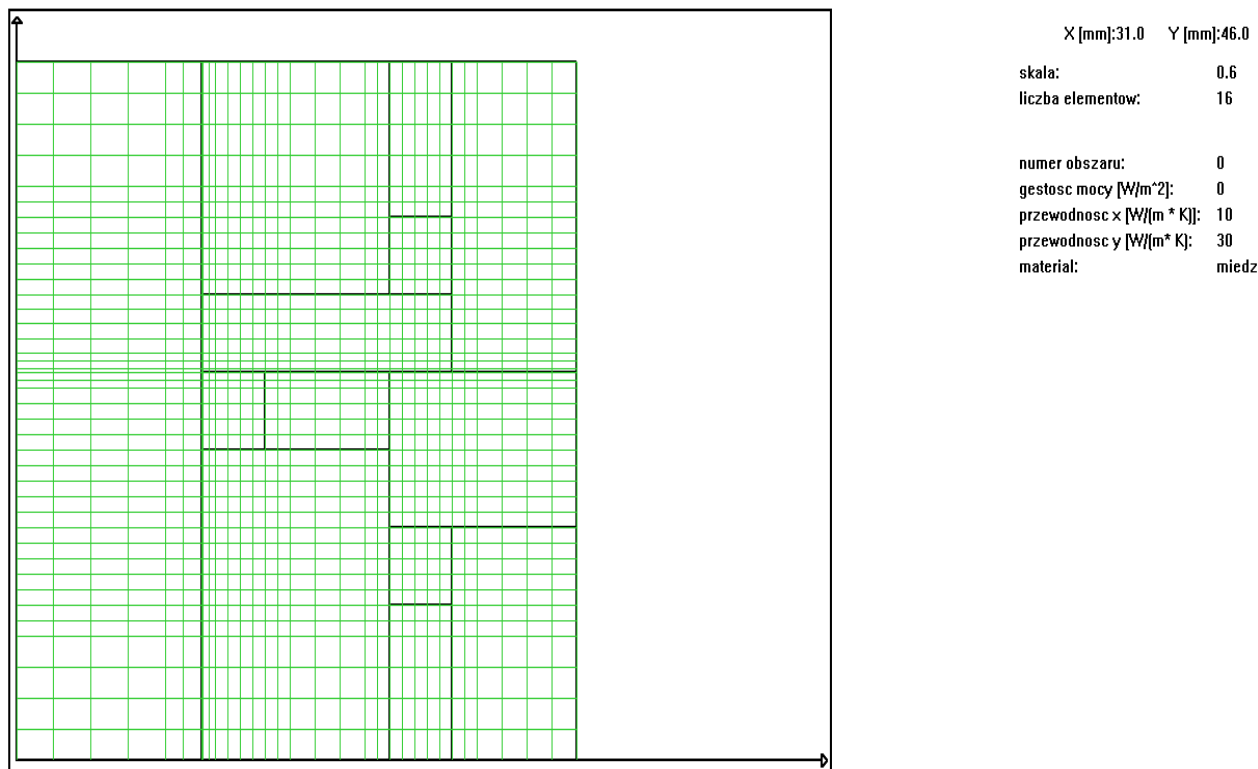
Rys. 8. Dane wejściowe użyte w pokazanym przykładzie.

Po wprowadzeniu danych okno programu wygląda tak, jak poniżej:



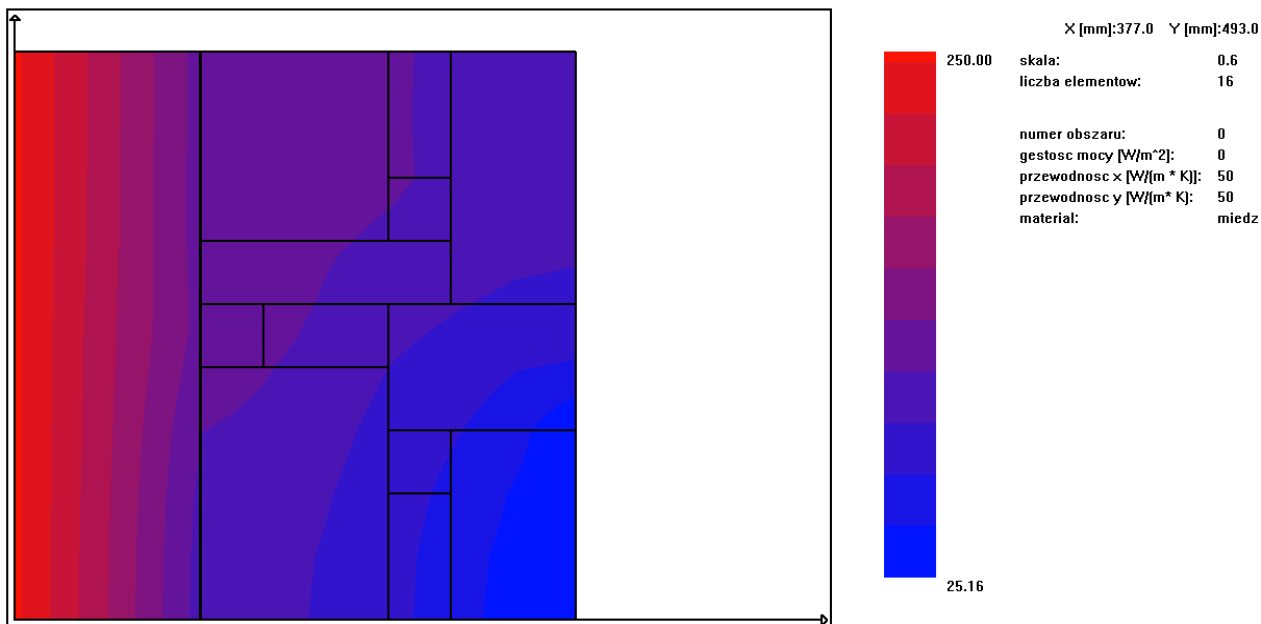
Rys. 9. Okno programu po wczytaniu danych wejściowych.

Poniższy rysunek (Rys 10.) przedstawia wygenerowaną i zagęszczoną siatkę.



Rys. 10. Wygenerowana i zagęszczona siatka.

Po wybraniu opcji „Licz rozkład temperatury” program liczy przez około 30 – 40 sekund. Po czym jest gotowy do dalszego działania. Następnie została wybrana opcja „Rysuj temperatury”. Program zwrócił następujące okno.



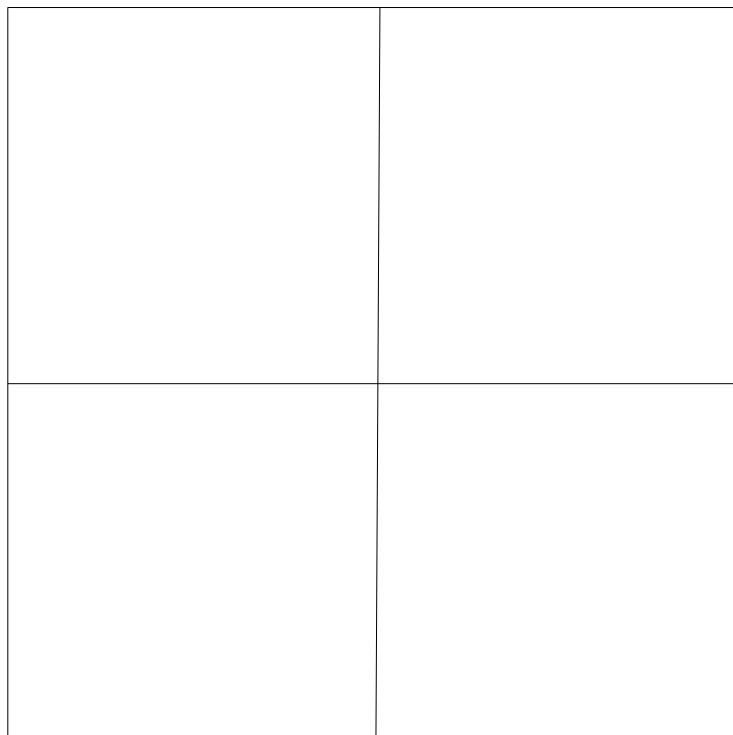
Rys. 11. Narysowany rozkład temperatur.

Program działa poprawnie, ponieważ pokazuje odpowiedni rozkład temperatur dla obszaru z trzema źródłami ciepła i warunkami brzegowymi na dolnym i górnym brzegu.

Następnie zostały wyprodukowane pliki wyjścia, które wyglądają tak jak na rysunku Rys. 6. i Rys. 7.

5. Testy

Przykład: Jednowymiarowy przepływ ciepła:



$$k_1 = k_3 = 100 \text{ W/(K m)}$$

$$k_2 = k_4 = 50 \text{ W/(K m)}$$

$$L_1 = 450 \text{ mm}$$

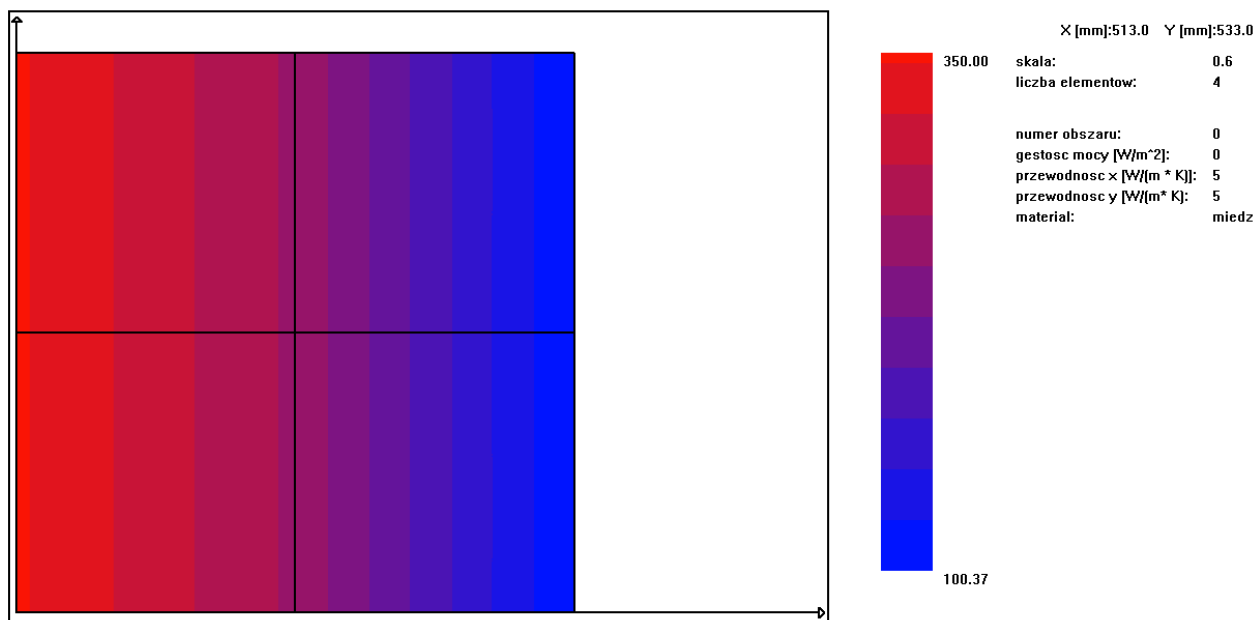
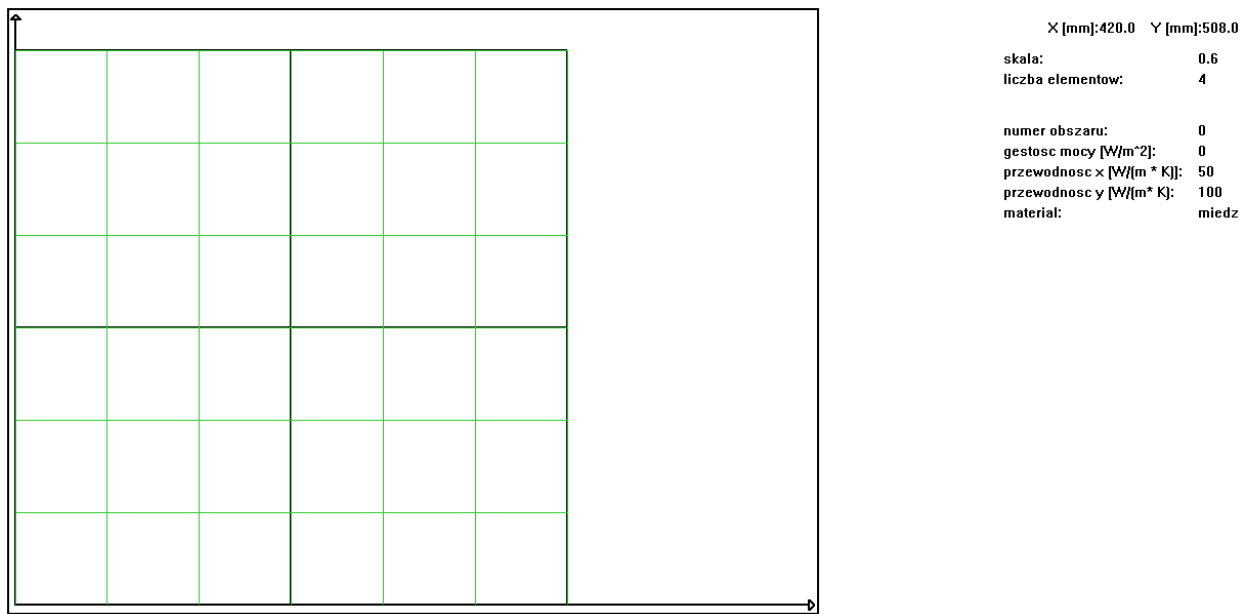
$$L_2 = 450 \text{ mm}$$

$$T_1 = 300 \text{ K}$$

$$T_2 = 150 \text{ K}$$

Wyniki obliczone analitycznie:

Numeryczne rozwiązanie:



Temperatura zmienia się liniowo od 350K do około 100K.

5. Wnioski

1. Program wykorzystuje w pełni metodę elementów skończonych razem z jej wszystkimi elementami.
2. W dość łatwy sposób można obliczyć rozkład temperatury nawet w skomplikowanych strukturach.
3. Rozwiązywane problemy są dosyć realistyczne, a nie są tylko idealnymi modelami fizycznymi.
4. Język programowania C++ bardzo dobrze radzi sobie ze wszystkimi obliczeniami. Najczęściej używanym elementem języka był vector, który działa bardzo podobnie jak tablica, ale jest od niej wydajniejszy i bardziej intuicyjny w użyciu.
5. Program wykazuje cechy podejścia obiektowego. Większość elementów, jak np. siatka czy wprowadzone obszary są osobnymi obiektami, które mają swoje atrybuty i metody.
6. Podczas rozwiązywania problemu i tworzeniu programu grupa projektowa wykorzystywała system kontroli wersji GIT oraz serwis GitHub, który pozwolił na sprawne zarządzanie wprowadzanymi zmianami oraz komunikację.
7. Projekt pozwolił na poszerzenie wiedzy z zakresu fizyki (termodynamiki), informatyki (programowanie) oraz pracy grupowej.
8. Cele projektu zostały zrealizowane.

6. Bibliografia

1. Treść wykładów dla przedmiotu „Numeryczne Metody Fizyki Komputerowej”, dr hab. inż. Robert Sarzała, profesor PŁ, Instytut Fizyki Politechniki Łódzkiej, Łódź 2019.
2. Skrypt do przedmiotu „Podstawy Metod Numerycznych” dr inż. Andrzej Brozi, Instytut Fizyki Politechniki Łódzkiej, Łódź 2019.
3. Microsoft Visual Studio C++ 2008. Praktyczne przykłady, Mariusz Owczarek, Helion, Łódź 2008.