

# Sprawozdanie Metoda Elementów Skończonych

Imię i nazwisko:  
Łukasz Sawina

WSTĘP TEORETYCZNY	2
OPIS MODELU	5
MES - OPIS	7
OPIS KODU	8
PORÓWNANIE WYNIKÓW	15
WNIOSKI	23

## Wstęp teoretyczny

### Transport ciepła

Jeden ze sposobów przekazywania energii pomiędzy układami termodynamicznymi. Siłą napędową dla wymiany jest różnica temperatur występująca pomiędzy obiektami. Zachodzi zawsze z ciała o temperaturze wyższej do ciała o temperaturze niższej, zgodnie z drugą zasadą termodynamiki. Wymiany ciepła między sąsiadującymi ciałami nie można zatrzymać, można ją jedynie spowolnić. Odbywać się będzie aż do momentu osiągnięcia stanu równowagi termicznej.

Można wymienić trzy mechanizmy odpowiedzialne za przepływ ciepła:

1. Przewodnictwo cieplne, jest to wymiana ciepła między ciałami o różnej temperaturze, która pozostają ze sobą w bezpośrednim kontakcie, polega na przekazywaniu energii kinetycznej bezładnego ruchu cząsteczek w wyniku ich zderzeń.
2. Konwekcja, wymiana ciepła przez makroskopowy ruch materii w gazie, cieczy lub plazmie. Proces konwekcji polega na unoszeniu ciepła na skutek przemieszczania się mas cieczy lub gazów.
3. Wymiana ciepła przez promieniowanie, pojawia się wtedy, gdy np. mikrofałe, promieniowanie podczerwone, światło widzialne lub inny rodzaj promieniowania elektromagnetycznego jest wysyłany lub pochłaniany przez ciało. Promieniowanie cieplne emitowane jest przez każdą materię o temperaturze wyższej od zera bezwzględnego.

### Ustalony wymiana ciepła

Znany również jako przewodzenie cieplne w warunkach ustalonych, odnosi się do sytuacji, w której ilość ciepła przekazywana przez daną substancję jest stała w czasie. Temperatura w danym obszarze pozostaje stała, a proces przewodzenia ciepła osiąga stan równowagi termicznej.

### Nieustalony wymiana ciepła

Zwany także, przewodzeniem ciepła w warunkach nieustalonych, występuje, gdy temperatura w danym obszarze zmienia się w czasie. W przeciwieństwie do ustalonego transportu ciepła, w przypadku nieustalonego temperatura w danym punkcie może się zmieniać w zależności od czasu.

### Warunki brzegowe

- Stała temperatura na powierzchni - temperatura na danej powierzchni utrzymuje się na stałym poziomie i nie zmienia się w czasie
- Stały strumień ciepła - na danej powierzchni utrzymywany jest stały strumień cieplny, który nie zmienia się w czasie

$$q = \text{const}$$

- Strumień ciepła według prawa konwekcji - opisuje, jak ciepło przemieszcza się między powierzchnią ciała a otoczeniem za pomocą konwekcji

$$q = \alpha(t - t_{\infty})$$

- Strumień ciepła według prawa radiacji - opisuje sposób, w jaki ciepło przemieszcza się między ciałem a otoczeniem poprzez promieniowanie elektromagnetyczne. Prawo to jest szczególnie ważne w przypadku wymiany ciepła w sytuacjach, gdzie przewodnictwo cieplne i konwekcja są ograniczone lub nie odgrywają głównej roli, na przykład w próżni

## Równanie Fouriera

Równanie w postaci:

$$\operatorname{div}(k(t)\operatorname{grad}(t)) + Q = 0$$

Gdzie:

- $k(t)$  współczynnik przewodzenia ciepła
- $Q$  prędkość generowania ciepła

Opisuje zjawiska cieplne zachodzące w stanie ustalonym. Równanie mówi nam, że suma dywergencji iloczynu  $k$  i gradientu temperatury oraz ilości wydzielanego ciepła jest równa zero w stanie ustalonym.

Równanie Fouriera dla procesu niestacjonarnego (nieustalonego) ma postać:

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

Gdzie:

- $c$  termiczna pojemność masowa
- $\rho$  gęstość materiału

## Prawo konwekcji

Opisuje sposób w jaki ciepło przemieszcza się między powierzchnią ciała a otoczeniem za pomocą konwekcji. Prawo konwekcji można przedstawić w postaci równania:

$$q = \alpha(t - t_{\infty})$$

Gdzie:

- $q$  ilość przekazywanego ciepła
- $\alpha$  współczynnik wnikania ciepła
- $t_{\infty}$  temperatura otoczenia

## Całkowanie Gaussa

Znane również jako całkowanie kwadratury Gaussa, to technika numerycznego całkowania, która wykorzystuje specjalnie dobrane węzły i wagi, aby obliczyć przybliżoną całkę o określonej funkcji.

W przestrzeni jednowymiarowej, całkowanie Gaussa może być zapisane jako:

$$\int_a^b f(x)dx \approx \sum_{i=1}^n w_i \times f(x_i)$$

W przestrzeni dwuwymiarowej:

$$\iint f(x,y)dA \approx \sum_{i=1}^n \sum_{j=1}^m w_{ij} \times f(x_i, y_j)$$

## Interpolacja

Proces estymowania wartości funkcji dla punktów znajdujących się pomiędzy znanymi wartościami funkcji. Pojawia się problem opisanego przestrzeni międzywęzłowych, dlatego wykorzystuje się Funkcje kształtu, w której wartości międzywęzłowe przybliża się do wielomianu interpolacyjnego trzeciego stopnia. Takie działanie daje wystarczająco dobre przybliżenie.

Interpolacja 1D za pomocą funkcji kształtu:

$$\begin{aligned}N_1 &= 0.5(1 - \xi) \\N_2 &= 0.5(1 + \xi) \\x &= \sum_{i=1}^{np} (N_i x_i) = N_1 x_1 + N_2 x_2\end{aligned}$$

Interpolacja 2D za pomocą funkcji kształtu:

$$\begin{aligned}N_1 &= 0.25(1 - \xi)(1 - \eta) \\N_2 &= 0.25(1 + \xi)(1 - \eta) \\N_3 &= 0.25(1 + \xi)(1 + \eta) \\N_4 &= 0.25(1 - \xi)(1 + \eta) \\x &= \sum_{i=1}^{np} (N_i x_i) = N_1 x_1 + N_2 x_2 + N_3 x_3 + N_4 x_4 = \{N\}^t \{x\} \\y &= \sum_{i=1}^{np} (N_i y_i) = N_1 y_1 + N_2 y_2 + N_3 y_3 + N_4 y_4 = \{N\}^t \{y\}\end{aligned}$$

## Opis modelu

Do symulacji nieustalonego procesu cieplnego wykorzystujemy postać równanie Fouriera dla procesów niestacjonarnych:

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

Jednak, aby wykonać symulację potrzebujemy to równanie zapisać w postaci możliwej do obliczenia przez nasz program, dlatego po przekształceniach z powyższego równania otrzymujemy równanie:

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

Z powyższego równania kolejne składowe to:

$\{t_0\}$

Wektor reprezentujący temperatury węzłowe w chwili  $\tau=0$ .

$\{t_1\}$

Wektor reprezentujący temperatury wynikowe po przedziale czasowym  $\Delta\tau$ . Ten wektor jest wynikiem naszej symulacji, czyli tym czego poszukujemy.

$\Delta\tau$

Krok czasowy w symulacji.

$[H]$

Macierz opisująca współczynniki wymiany ciepła przez przewodzenie między punktami w układzie. Przechowuje współczynniki układu równań opisującego wpływ przewodzonego ciepła na temperatury w każdym z węzłów układu. W tym równaniu jest to zagregowana wersja, czyli przechowuje w sobie wszystkie Macierze  $[H]$  elementów oraz Macierze  $[HBC]$  zsumowanie.

Macierz  $[HBC]$  to zbiór wartości opisujących wpływ konwekcji na układ powierzchni elementu, jest również zależny od zmiennej temperatury tej powierzchni.

$$[HBC] = \int \alpha \{N\} \{N\}^T dS$$

Gdzie:

- $\alpha$  współczynnik wnikania ciepła
- $\{N\}$  wektor funkcji kształtu elementu

Macierz  $[H]$  ma postać:

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

Jak widać drugi człon tego wyrażenia to macierz  $[HBC]$ .

$\{P\}$

Wektor reprezentujący strumień ciepła, jego składowe opisują ilość ciepła przepływającego przez powierzchnię w jednostce czasu i jednostce powierzchni. Składowe są związane z przewodnictwem cieplnym materiału i temperaturami w elemencie skończonym.

Wektor  $\{P\}$  przyjmuje postać

$$[P] = - \int \alpha \{N\} t_{\infty} dS - \int Q \{N\} dV + \int q \{N\} dS$$

Gdzie:

- $\alpha$  współczynnik wnikania ciepła
- $\{N\}$  wektor funkcji kształtu elementu
- $t_{\infty}$  temperatura otoczenia
- $Q$  prędkość generowania ciepła
- $q$  gęstość strumienia ciepła

$[C]$

Macierz przechowująca informacje o pojemności cieplnej materiałów w elemencie. Związana jest z informacją ile energii cieplnej jest w stanie przechować dany materiał. Macierz ta uwzględnia wpływ zmiany temperatury na stan termiczny struktury i sposób w jaki materiał przewodzi i przechowuje ciepło.

Macierz  $[C]$  przyjmuje postać:

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

Gdzie:

- $c$  ciepło właściwe, mówi o tym, ile energii materiał jest w stanie zmagazynować. Jeżeli może przyjąć więcej ciepła to także dłużej będzie to ciepło przyjmować.
- $\rho$  gęstość, materiał bardziej porowaty ma mniejszą gęstość i będzie mógł przechować mniej energii niż materiał o tej samej objętości i większej gęstości.
- $\{N\}$  wektor funkcji kształtu elementu

Do obliczeń wszystkich składowych równania potrzeba policzyć całkę powierzchniową/objętościową. Do tego wykorzystujemy w naszym programie całkowanie metoda Gaussa.

Do obliczania układów równań możemy wykorzystać jedną z kilku metod Metoda faktoryzacji LU, Metoda Eliminacji Gaussa itp.

## MES - Opis

Metoda rozwiązywania równań różniczkowych, w której na obiekt nakładana jest siatka elementów skończonych. Główną ideą MES jest, że dowolną ciągłą wartość (np. temperaturę) można zmienić na model dyskretny. Model opiera się na ograniczonej ilości węzłów, które tworzą ograniczoną ilość elementów skończonych. Dzięki tej metodzie możemy obliczyć różne procesy spotykanych w fizyce i technice. Wartość temperatury lub innej funkcji w każdym węźle definiujemy jako parametr, który poszukujemy.

Na obiekt nakładana jest siatka elementów skończonych. Siatka może być nakładana na obiekt przestrzenny jak również na płaską powierzchnię, każdy element jest prostym geometrycznie kształtem, takim jak trójkąt, czworokąt czy sześciąt.

Temperaturę aproksymuje się na każdym elemencie za pomocą wielomianu, który wyznaczony jest za pomocą węzłów wartości temperatury. Dodatkowo wyznacza się go w taki sposób, aby zachować warunek ciągłości temperatury na granicach elementów. Wartości temperatur w węzłach muszą być tak dobrane, aby zapewniały jak najlepsze odwzorowanie rzeczywistego pola temperatury, można to osiągnąć przy pomocy minimalizacji funkcjonu, który odpowiada różnicowemu równaniu przewodzenia ciepła.

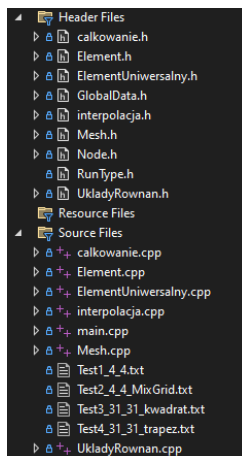
Dodatkowo z wyników otrzymanych możemy wykonywać wizualizację, aby lepiej zrozumieć zachowania analizowanych obiektów. Można do tego wykorzystać już istniejące programy jak np. ParaView.

Zaletami MES są:

- Własności materiału elementów nie muszą być jednakowe, daje to możliwość wykorzystania MES do materiałów wielofazowych oraz takich, których własności zależą od temperatury,
- Wymiary elementów mogą być objętościowo różne, daje to możliwość powiększania lub zmniejszania wymiarów elementów w pewnych strefach rozpatrywanej objętości, a co za tym idzie lepsze odwzorowanie rzeczywistego obiektu,
- Z wykorzystaniem MES można uwzględniać nieliniowe warunki brzegowe

## Opis kodu

Program został napisany w języku C++, wszystkie modele danych jak węzły, elementy itp. zostały napisane z wykorzystaniem struktur. Struktura plików w programie wygląda następująco:



Fot. 1 Struktura plików

Głównym modelem w kodzie jest struktura Mesh, która przechowuje w sobie wszystkie informacje o siatce oraz zawiera wszystkie funkcje do obliczeń poszczególnych elementów programu oraz temperatur.

```
struct Mesh
{
    GlobalData* globalData;
    Node* nodes;
    Element* elements;

    void readMeshFile(std::string fileSrc);
    void showGlobalData();
    void showNodes();
    void showElements();

    void calcHForElements(const ElementUniwersalny& elUni);
    void calcHBCForElements(const ElementUniwersalny& elUni);
    void calcVectorPForElements(const ElementUniwersalny& elUni);
    void calcCForElements(const ElementUniwersalny& elUni);

    static double* calcTemperature(Mesh& mesh, const ElementUniwersalny& elUni);

    ~Mesh();
private:
    GlobalData* readMeshGlobalData(std::string fileSrc);
    Node* readMeshNodes(std::string fileSrc);
    Element* readMeshElements(std::string fileSrc);
};
```

Fot.2 Struktura Mesh – główna struktura w programie

Do przeprowadzenia symulacji zmiany temperatury potrzebne jest uruchomienie tylko 2 funkcji ze struktury Mesh oraz utworzenie elementu uniwersalnego, który jest przekazywany jako parametr do wszystkich funkcji liczących.



Przykładowe wykonanie symulacji:

```
Mesh siatka;  
siatka.readMeshFile("Test2_4_4_MixGrid.txt");  
  
ElementUniwersalny elementUniwersalny;  
elementUniwersalny.init(3);  
  
Mesh::calcTemperature(siatka, elementUniwersalny);
```

Fot.3 Przykładowe wywołanie symulacji

Funkcja calcTemperature() przyjmuje jako parametry strukturę siatki, do której zostały wczytane dane z pliku .txt oraz zainicjalizowaną strukturę elementu uniwersalnego. Przy inicjalizacji elementu uniwersalnego potrzebne jest podanie ilość punktów w układzie całkowania, na podstawie tego parametru będą dopasowywane wszystkie układy całkowania dla całego programu.

Wewnątrz funkcji calcTemperature() wykonywane są kolejno obliczenia składowych modelu, takie jak: macierze H, HBC, C oraz wektor P. Dodatkowo dla opcji sprawdzania wyników poszczególnych składowych modelu możliwe jest wywołanie z osobna każdej z tych funkcji. Po obliczeniu każdej ze składowych tworzony jest agregat H oraz wektor temperatur początkowych. Następnie w pętli obliczane są temperatury dla danego kroku, po czym w konsoli wyświetlane są dane odnośnie minimalnej/maksymalnej temperatury dla danego kroku oraz generowany jest plik .vtk, do dalszego wykorzystania w programie ParaView.

```
for (int i = 0; i < nstep; i++)  
{  
    tempV = calcTemperatureForStep(mesh, agregatH, agregatC, tempV);  
  
    std::cout << std::endl;  
    std::cout << "Time: " << mesh.globalData->SimulationStepTime * (i + 1) << std::endl;  
  
    std::pair<double, double> wyniki = znajdz_min_i_max(tempV, mesh.globalData->NodesNumber);  
    std::cout << "Min: " << wyniki.first << "\t" << wyniki.second;  
  
    GenerateOutputFile(mesh, tempV, i+1);  
#ifdef DEBUG_CALC_TEMP  
    std::cout << "Temperatures: " << std::endl;  
    for (int j = 0; j < mesh.globalData->NodesNumber; j++)  
    {  
        std::cout << tempV[j] << " ";  
    }  
    std::cout << std::endl;  
#endif  
}  
  
return tempV;
```

Fot.4 Fragment calcTemperature() – obliczanie temperatur, wyświetlanie w konsoli oraz generowanie pliku

Jak można zauważyć pojawiają się w kodzie również instrukcje preprocesora, odpowiadają one za dodatkowe wyświetlenie danych przy obliczeniach, w celu kontroli wyników dla poszczególnych elementów. Do uruchomienia jednego z takich trybów w pliku RunType.h wystarczy odkomentować wybraną definicję, każda z nich odpowiada poszczególnym obliczeniom, co można odczytać z ich nazw.

```
//#define DEBUG_ELEMENT_UNIwersalny  
//#define DEBUG_ELEMENT_H  
//#define DEBUG_ELEMENT_HBC  
//#define DEBUG_ELEMENT_VP  
//#define DEBUG_ELEMENT_C  
//#define DEBUG_CALC_TEMP
```

Fot.5 Tryby uruchomienia

Dla zmniejszenia ilości opisu kodu opiszę tylko najważniejsze części poszczególnych funkcji liczących składowe modelu.

### Element uniwersalny

Do uruchomienia wszystkich funkcji w strukturze Mesh potrzebna jest zainicjalizowana struktura elementu uniwersalnego, przechowuje ona w sobie następujące informacje:

```
struct ElementUniwersalny {  
    double** matdEta;  
    double** matdKsi;  
    std::vector<double**> matNPktForEdges;  
    double** N;  
  
    int nPkt;  
    void init(int n);  
  
    ~ElementUniwersalny();  
};
```

Fot.6 Deklaracja struktury ElementUniwersalny

Jak widać przechowuje ona pochodne funkcji kształtu względem ksi oraz eta zapisane w macierzy (matdKsi/matdEta), funkcje kształtu dla warunków brzegowych (matNPktForEdges), funkcje kształtu dla punktów (N) oraz ilość punktów schematu całkowania.

Przy uruchomieniu funkcji init() obliczane są wszystkie powyżej wymienione dane, np.:

```
for (int i = 0, k = 0; i < n; i++)  
{  
    for (int j = 0; j < n; j++, k++)  
    {  
        matdKsi[k][0] = -0.25 * (1 - G_X[i]);  
        matdKsi[k][1] = 0.25 * (1 - G_X[i]);  
        matdKsi[k][2] = 0.25 * (1 + G_X[i]);  
        matdKsi[k][3] = -0.25 * (1 + G_X[i]);  
  
        matdEta[k][0] = -0.25 * (1 - G_X[j]);  
        matdEta[k][1] = -0.25 * (1 + G_X[j]);  
        matdEta[k][2] = 0.25 * (1 + G_X[j]);  
        matdEta[k][3] = 0.25 * (1 - G_X[j]);  
    }  
}
```

Fot.7 obliczanie pochodnych funkcji kształtu ksi/eta

```
for (int j = 0; j < nPktCalc; j++)  
{  
#ifdef DEBUG_ELEMENT_UNIERSALNY  
    std::cout << std::endl << "ksi: " << pkt[j][0] << " eta: " << pkt[j][1] << std::endl;  
#endif  
  
    N[j][0] = 0.25 * (1 - pkt[j][0]) * (1 - pkt[j][1]);  
    N[j][1] = 0.25 * (1 + pkt[j][0]) * (1 - pkt[j][1]);  
    N[j][2] = 0.25 * (1 + pkt[j][0]) * (1 + pkt[j][1]);  
    N[j][3] = 0.25 * (1 - pkt[j][0]) * (1 + pkt[j][1]);  
  
#ifdef DEBUG_ELEMENT_UNIERSALNY  
    std::cout << "N1: " << N[j][0];  
    std::cout << " N2: " << N[j][1];  
    std::cout << " N3: " << N[j][2];  
    std::cout << " N4: " << N[j][3] << std::endl;  
#endif  
}
```

Fot. 8 obliczanie funkcji kształtu

## Macierz H

Do obliczenia macierzy H dla elementów w siatce trzeba uruchomić funkcję `calcHForElements()`, pobiera ona współrzędne węzłów i przekazuje je do funkcji `calcH`, która liczy macierz H dla danego elementu.

```
void Mesh::calcHForElements(const ElementUniwersalny& elUni)
{
    for (int i = 0; i < globalData->ElementsNumber; i++)
    {
#ifdef DEBUG_ELEMENT_H
        std::cout << std::endl << "Element: " << i+1 << std::endl;
#endif
        double* x = new double[4];
        double* y = new double[4];

        for (int j = 0; j < 4; j++)
        {
            int n = elements[i].ID_wezlow[j];
            x[j] = nodes[n-1].x;
            y[j] = nodes[n-1].y;
#ifdef DEBUG_ELEMENT_H
            std::cout << "PC " << i+1 << "(x: " << x[j] << " ; " << y[j] << ")" << std::endl;
#endif
        }

        elements[i].calcH(x, y, elUni);

        delete[] x;
        delete[] y;
    }
}
```

Fot.9 Wywołanie obliczeń macierzy H dla każdego elementu

Wewnątrz funkcji `calcH` obliczane są kolejno macierz Jakobiego, Jakobian, odwrotność Jakobianu, pochodne funkcji kształtu względem X oraz Y, macierze H punktów oraz macierz H całego elementu.

```
for (int i = 0; i < pow(elUni.nPkt, 2); i++)
{
    double** macierzJakobiego = new double* [2];

    for (int i = 0; i < 2; i++)
    {
        macierzJakobiego[i] = new double[2];
    }

    //Liczenie macierzy jakobiego dla pc[i]
    macierzJakobiego[0][0] = elUni.matdEta[i][0] * y[0] + elUni.matdEta[i][1] * y[1] + elUni.matdEta[i][2] * y[2] + elUni.matdEta[i][3] * y[3];
    macierzJakobiego[0][1] = -1 * (elUni.matdKsi[i][0] * y[0] + elUni.matdKsi[i][1] * y[1] + elUni.matdKsi[i][2] * y[2] + elUni.matdKsi[i][3] * y[3]);
    macierzJakobiego[1][0] = -1 * (elUni.matdEta[i][0] * x[0] + elUni.matdEta[i][1] * x[1] + elUni.matdEta[i][2] * x[2] + elUni.matdEta[i][3] * x[3]);
    macierzJakobiego[1][1] = elUni.matdKsi[i][0] * x[0] + elUni.matdKsi[i][1] * x[1] + elUni.matdKsi[i][2] * x[2] + elUni.matdKsi[i][3] * x[3];

    macierzeJakobiegoPunktow.push_back(macierzJakobiego);

    double detJ = macierzJakobiego[0][0] * macierzJakobiego[1][1] - (macierzJakobiego[1][0] * macierzJakobiego[0][1]);
    detJPunktow.push_back(detJ);
}
```

Fot.10 Obliczenia macierzy Jakobiego oraz Jakobian

```
for (int i = 0; i < 4; i++)
{
    Hpkt[i][0] = (matdx[p][i] * matdx[p][0] + matdy[p][i] * matdy[p][0]) * detJPunktow[p] * globalData->Conductivity;
    Hpkt[i][1] = (matdx[p][i] * matdx[p][1] + matdy[p][i] * matdy[p][1]) * detJPunktow[p] * globalData->Conductivity;
    Hpkt[i][2] = (matdx[p][i] * matdx[p][2] + matdy[p][i] * matdy[p][2]) * detJPunktow[p] * globalData->Conductivity;
    Hpkt[i][3] = (matdx[p][i] * matdx[p][3] + matdy[p][i] * matdy[p][3]) * detJPunktow[p] * globalData->Conductivity;
}

macierzeHPunktow.push_back(Hpkt);
```

Fot.11 Obliczenia macierzy H dla punktu

```

for (int i = 0, p = 0; i < elUni.nPkt; i++)
{
    for (int j = 0; j < elUni.nPkt; j++, p++)
    {
        for (int n = 0; n < 4; n++)
            for (int m = 0; m < 4; m++)
            {
                macierzeHPunktow[p][n][m] *= (G_W[i] * G_W[j]);
                outputH[n][m] += macierzeHPunktow[p][n][m];
            }
    }
}

```

Fot.12 Obliczenia macierzy H dla elementu

## Macierz HBC

W przypadku macierzy HBC sytuacja jest podobna jak dla macierzy H, funkcja calcHBCForElements() pobiera współrzędne elementu oraz flagi warunku brzegowego i na tej podstawie oblicza macierz HBC dla elementów.

```

for (int i = 0; i < 4; i++)
{
    for (int j = 0; j < 4; j++)
        HBCEdge[i][j] = 0;

    for (int j = 0; j < elUni.nPkt; j++)
    {
        for (int k = 0; k < 4; k++)
        {
            HBCEdge[i][k] += elUni.matNPktForEdges[p][j][i] * elUni.matNPktForEdges[p][j][k] * G_W[j] * globalData->Alfa;
        }

        for (int j = 0; j < 4; j++)
            HBCEdge[i][j] *= detJ[p];
    }

    HBCEdges.push_back(HBCEdge);
}

```

Fot.13 Obliczenia macierzy HBC dla krawędzi

## Wektor P

Tutaj identycznie jak w wyżej w calcVectorPForElements() pobierane są współrzędne X, Y oraz flagi warunku brzegowego i dla każdego elementu z osobna liczony jest wektor P.

```

for (int i = 0; i < 4; i++)
{
    for (int j = 0; j < elUni.nPkt; j++)
    {
        vectorPForEdge[i] += elUni.matNPktForEdges[e][j][i] * globalData->Tot * G_W[j];
    }

    for (int j = 0; j < 4; j++)
        vectorPForEdge[j] = vectorPForEdge[j] * detJ[e] * globalData->Alfa;

    vectorPForEdges.push_back(vectorPForEdge);
}

```

Fot.14 Obliczenia wektora P dla krawędzi

## Macierz C

W funkcji calcCForElements() ponownie pobierane są współrzędne X oraz Y elementu i na ich podstawie liczone są macierze C.

```
for (int i = 0; i < 4; i++)
{
    Cpkt[i][0] = (elUni.N[p][i] * elUni.N[p][0]) * detJPunktow[p] * globalData->Density * globalData->SpecificHeat;
    Cpkt[i][1] = (elUni.N[p][i] * elUni.N[p][1]) * detJPunktow[p] * globalData->Density * globalData->SpecificHeat;
    Cpkt[i][2] = (elUni.N[p][i] * elUni.N[p][2]) * detJPunktow[p] * globalData->Density * globalData->SpecificHeat;
    Cpkt[i][3] = (elUni.N[p][i] * elUni.N[p][3]) * detJPunktow[p] * globalData->Density * globalData->SpecificHeat;
}

macierzeCPunktow.push_back(Cpkt);
```

Fot.15 Obliczenia macierzy C dla punktów

```
for (int i = 0, p = 0; i < elUni.nPkt; i++)
{
    for (int j = 0; j < elUni.nPkt; j++, p++)
    {
        for (int n = 0; n < 4; n++)
            for (int m = 0; m < 4; m++)
            {
                macierzeCPunktow[p][n][m] *= (G_W[i] * G_W[j]);
                outputC[n][m] += macierzeCPunktow[p][n][m];
            }
    }
}
```

Fot. 16 Obliczenia macierzy C dla elementu

## Kwadratura Gaussa oraz układy równań

W wielu miejscach w kodzie potrzebne są do obliczeń punkty, w których obliczamy wartości funkcji oraz współczynniki wagowe przypisane dla tych punktów. W programie zdefiniowane są one dla schematu całkowania 2, 3, 4, oraz 5 punktowego. Wartości ich są obliczane ze wzorów, dlatego dla różnych maszyn może pojawiać się pewna niewielka różnica w wynikach, spowodowana różnymi przybliżeniami.

```
double* G_X = new double[n];

switch (n)
{
case 2:
    G_X[0] = -sqrt(1./3.);
    G_X[1] = sqrt(1./3.);
    break;
case 3:
    G_X[0] = -sqrt(3./5.);
    G_X[1] = 0;
    G_X[2] = sqrt(3./5.);
    break;
case 4:
    G_X[0] = -sqrt((3. / 7.) + (2. / 7.) * sqrt(6. / 5.));
    G_X[1] = -sqrt((3. / 7.) - (2. / 7.) * sqrt(6. / 5.));
    G_X[2] = sqrt((3. / 7.) - (2. / 7.) * sqrt(6. / 5.));
    G_X[3] = sqrt((3. / 7.) + (2. / 7.) * sqrt(6. / 5.));
    break;
case 5:
    G_X[0] = -(1. / 3.) * sqrt(5. + 2. * (10. / 7.));
    G_X[1] = -(1. / 3.) * sqrt(5. - 2. * (10. / 7.));
    G_X[2] = 0;
    G_X[3] = (1. / 3.) * sqrt(5. - 2. * (10. / 7.));
    G_X[4] = (1. / 3.) * sqrt(5. + 2. * (10. / 7.));
    break;
default:
    return 0;
}

return G_X;
```

```
double* G_W = new double[n];

switch (n)
{
case 2:
    G_W[0] = G_W[1] = 1;
    break;
case 3:
    G_W[0] = G_W[2] = (5. / 9.);
    G_W[1] = (8. / 9.);
    break;
case 4:
    G_W[0] = G_W[3] = (18. - sqrt(30.)) / 36.;
    G_W[1] = G_W[2] = (18. + sqrt(30.)) / 36.;
    break;
case 5:
    G_W[0] = G_W[4] = (322. - 13. * sqrt(70.)) / 900.;
    G_W[1] = G_W[3] = (322. + 13. * sqrt(70.)) / 900.;
    G_W[2] = (128. / 225.);
    break;
default:
    return 0;
}

return G_W;
```

Fot.17,18 Współrzędne punktów całkowania oraz ich wagi

Do obliczania układów równań wykorzystana jest Metoda Eliminacji Gaussa, która została napisana przeze mnie na przedmiocie Metody Numeryczne.

Przykładowe wyniki programu wyglądają następująco:

```
Time: 1
Min: 100      166.936
Time: 2
Min: 100      207.233
Time: 3
Min: 100      236.287
Time: 4
Min: 100      259.465
Time: 5
Min: 100      279.031
Time: 6
Min: 100      296.121
Time: 7
Min: 100.001  311.385
Time: 8
Min: 100.001  325.235
Time: 9
Min: 100.003  337.951
Time: 10
Min: 100.005  349.731
Time: 11
Min: 100.01   360.723
Time: 12
Min: 100.018  371.04
Time: 13
Min: 100.03   380.771
```

Fot.19 Fragment podstawowego wyniku programu, widoczne minimalne i maksymalne temperatury dla kroków

```
Agregat BC
29793.4
30615.7
20556.4
13249.9
30615.7
19946.4
16170.7
20556.4
20556.4
16170.7
19946.4
30615.7
13249.9
20556.4
30615.7
29793.4

Time: 100
Min: 147.656  505.954Temperatures:
488.557 329.292 381.084 505.954 329.292 147.656 186.878 381.085 381.085 186.878 147.656 329.292 505.954 381.085 329.292 488.557
```

Fot.20 Fragment wyniku programu z trybem debugowania dla obliczania temperatury, widoczne minimalne i maksymalne temperatury dla kroków oraz poszczególne wartości w węzłach.

```
---Macierze H punktow---
PC1
10.2331 -4.46668 -1.29978 -4.46668
-4.46668 4.95499 0.567342 -1.05565
-1.29978 0.567342 0.165093 0.567342
-4.46668 -1.05565 0.567342 4.95499

PC2
6.94126 -2.95992 -2.61948 -1.36186
-2.95992 5.69277 -0.816903 -1.91594
-2.61948 -0.816903 1.83267 1.60371
-1.36186 -1.91594 1.60371 1.67409

PC3
4.74912 -2.83592 -2.27341 0.360209
-2.83592 8.16915 -4.29561 -1.03762
-2.27341 -4.29561 6.02341 0.545615
0.360209 -1.03762 0.545615 0.131795

PC4
6.94126 -1.36186 -2.61948 -2.95992
-1.36186 1.67409 1.60371 -1.91594
-2.61948 1.60371 1.83267 -0.816903
-2.95992 -1.91594 -0.816903 5.69277
```

Fot.20 Fragment wyniku programu z trybem debugowania dla obliczania macierzy H, widoczny fragment pokazuje macierze H dla punktów całkowania w elemencie

## Porównanie wyników

Do otrzymania wszystkich wyników został użyty trójpunktowy schemat całkowania.

Siatka 4x4

W przypadku pliku Test1\_4\_4.txt ponieważ siatka jest kwadratem Macierze H Lokalne są zbliżone do siebie z niewielkimi różnicami po przecinku dla wszystkich elementów, dlatego przedstawiam zestawienie tylko jednej.

Element							
Wyniki testu				Wyniki programu			
16.6667	-4.16667	-8.33333	-4.16667	16.6667	-4.16667	-8.33333	-4.16667
-4.16667	16.6667	-4.16667	-8.33333	-4.16667	16.6667	-4.16667	-8.33333
-8.33333	-4.16667	16.6667	-4.16667	-8.33333	-4.16667	16.6667	-4.16667
-4.16667	-8.33333	-4.16667	16.6667	-4.16667	-8.33333	-4.16667	16.6667

W przypadku macierzy HBC oraz wektora P pojawiają się różne wartości dla różnych elementów, dlatego zostaną one zestawione dla porównania.

Macierz HBC								
Wyniki testu					Wyniki programu			
Element 1								
6.66667	1.66667	0	1.66667		6.66667	1.66667	0	1.66667
1.66667	3.33333	0	0		1.66667	3.33333	0	0
0	0	0	0		0	0	0	0
1.66667	0	0	3.33333		1.66667	0	0	3.33333
Element 2								
3.33333	1.66667	0	0		3.33333	1.66667	0	0
1.66667	3.33333	0	0		1.66667	3.33333	0	0
0	0	0	0		0	0	0	0
0	0	0	0		0	0	0	0
Element 3								
3.33333	1.66667	0	0		3.33333	1.66667	0	0
1.66667	6.66667	1.66667	0		1.66667	6.66667	1.66667	0
0	1.66667	3.33333	0		0	1.66667	3.33333	0
0	0	0	0		0	0	0	0
Element 4								
3.33333	0	0	1.66667		3.33333	0	0	1.66667
0	0	0	0		0	0	0	0
0	0	0	0		0	0	0	0
1.66667	0	0	3.33333		1.66667	0	0	3.33333
Element 5								
0	0	0	0		0	0	0	0
0	0	0	0		0	0	0	0
0	0	0	0		0	0	0	0
0	0	0	0		0	0	0	0

Element 6							
0	0	0	0	0	0	0	0
0	3.33333	1.66667	0	0	3.33333	1.66667	0
0	1.66667	3.33333	0	0	1.66667	3.33333	0
0	0	0	0	0	0	0	0
Element 7							
3.33333	0	0	1.66667	3.33333	0	0	1.66667
0	0	0	0	0	0	0	0
0	0	3.33333	1.66667	0	0	3.33333	1.66667
1.66667	0	1.66667	6.66667	1.66667	0	1.66667	6.66667
Element 8							
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	3.33333	1.66667	0	0	3.33333	1.66667
0	0	1.66667	3.33333	0	0	1.66667	3.33333
Element 9							
0	0	0	0	0	0	0	0
0	3.33333	1.66667	0	0	3.33333	1.66667	0
0	1.66667	6.66667	1.66667	0	1.66667	6.66667	1.66667
0	0	1.66667	3.33333	0	0	1.66667	3.33333

Wektor P								
Wyniki testu					Wyniki programu			
Element 1								
12000	6000	0	6000		12000	6000	0	6000
Element 2								
6000	6000	0	0		6000	6000	0	0
Element 3								
6000	12000	6000	0		6000	12000	6000	0
Element 4								
6000	0	0	6000		6000	0	0	6000
Element 5								
0	0	0	0		0	0	0	0
Element 6								
0	6000	6000	0		0	6000	6000	0
Element 7								
6000	0	6000	12000		6000	0	6000	12000
Element 8								
0	0	6000	6000		0	0	6000	6000
Element 9								
0	6000	12000	6000		0	6000	12000	6000



W przypadku macierzy C jest podobnie jak dla macierzy H, wartości dla wszystkich elementów są zbliżone do siebie, z nielicznymi różnicami po przecinku, dlatego zestawię wyniki dla jednego elementu.

Macierz C								
Wyniki testu					Wyniki programu			
Element								
674.074	337.037	168.519	337.037		674.074	337.037	168.519	337.037
337.037	674.074	337.037	168.519		337.037	674.074	337.037	168.519
168.519	337.037	674.074	337.037		168.519	337.037	674.074	337.037
337.037	168.519	337.037	674.074		337.037	168.519	337.037	674.074

Jak można zobaczyć po zagregowaniu danych macierzy H oraz HBC do macierzy globalnej H otrzymujemy identyczne macierze dla danych testowych oraz danych otrzymanych z programu

Macierz globalna H															
Wyniki testu															
23.3333	-2.5	0	0	-2.5	-8.33333	0	0	0	0	0	0	0	0	0	0
-2.5	40	-2.5	0	-8.33333	-8.33333	-8.33333	0	0	0	0	0	0	0	0	0
0	-2.5	40	-2.5	0	-8.33333	-8.33333	-8.33333	0	0	0	0	0	0	0	0
0	0	-2.5	23.3333	0	0	-8.33333	-2.5	0	0	0	0	0	0	0	0
-2.5	-8.33333	0	0	40	-8.33334	0	0	-2.5	-8.33333	0	0	0	0	0	0
-8.33333	-8.33333	-8.33333	0	-8.33334	66.6667	-8.33333	0	-8.33333	-8.33333	-8.33333	0	0	0	0	0
0	-8.33333	-8.33333	-8.33333	0	-8.33333	66.6667	-8.33333	0	-8.33333	-8.33333	-8.33333	0	0	0	0
0	0	-8.33333	-2.5	0	0	-8.33333	40	0	0	-8.33333	-2.5	0	0	0	0
0	0	0	0	-2.5	-8.33333	0	0	40	-8.33333	0	0	-2.5	-8.33333	0	0
0	0	0	0	-8.33333	-8.33333	-8.33333	0	-8.33333	66.6667	-8.33333	0	-8.33333	-8.33333	-8.33333	0
0	0	0	0	0	-8.33333	-8.33333	-8.33333	0	-8.33333	66.6667	-8.33333	0	-8.33333	-8.33334	-8.33333
0	0	0	0	0	0	-8.33333	-2.5	0	0	-8.33333	40	0	0	-8.33333	-2.5
0	0	0	0	0	0	0	0	-2.5	-8.33333	0	0	23.3333	-2.5	0	0
0	0	0	0	0	0	0	0	-8.33333	-8.33333	-8.33333	0	-2.5	40	-2.5	0
0	0	0	0	0	0	0	0	0	-8.33333	-8.33334	-8.33333	0	-2.5	40	-2.5
0	0	0	0	0	0	0	0	0	0	-8.33333	-2.5	0	0	-2.5	23.3333
Wyniki programu															
23.3333	-2.5	0	0	-2.5	-8.33333	0	0	0	0	0	0	0	0	0	0
-2.5	40	-2.5	0	-8.33333	-8.33333	-8.33333	0	0	0	0	0	0	0	0	0
0	-2.5	40	-2.5	0	-8.33333	-8.33333	-8.33333	0	0	0	0	0	0	0	0
0	0	-2.5	23.3333	0	0	-8.33333	-2.5	0	0	0	0	0	0	0	0
-2.5	-8.33333	0	0	40	-8.33334	0	0	-2.5	-8.33333	0	0	0	0	0	0
-8.33333	-8.33333	-8.33333	0	-8.33334	66.6667	-8.33333	0	-8.33333	-8.33333	-8.33333	0	0	0	0	0
0	-8.33333	-8.33333	-8.33333	0	-8.33333	66.6667	-8.33333	0	-8.33333	-8.33333	-8.33333	0	0	0	0
0	0	-8.33333	-2.5	0	0	-8.33333	40	0	0	-8.33333	-2.5	0	0	0	0
0	0	0	0	-2.5	-8.33333	0	0	40	-8.33333	0	0	-2.5	-8.33333	0	0
0	0	0	0	-8.33333	-8.33333	-8.33333	0	-8.33333	66.6667	-8.33333	0	-8.33333	-8.33333	-8.33333	0
0	0	0	0	0	-8.33333	-8.33333	-8.33333	0	-8.33333	66.6667	-8.33333	0	-8.33333	-8.33334	-8.33333
0	0	0	0	0	0	-8.33333	-2.5	0	0	-8.33333	40	0	0	-8.33333	-2.5
0	0	0	0	0	0	0	0	-2.5	-8.33333	0	0	23.3333	-2.5	0	0
0	0	0	0	0	0	0	0	-8.33333	-8.33333	-8.33333	0	-2.5	40	-2.5	0
0	0	0	0	0	0	0	0	0	-8.33333	-8.33334	-8.33333	0	-2.5	40	-2.5
0	0	0	0	0	0	0	0	0	0	-8.33333	-2.5	0	0	-2.5	23.3333

Podobna sytuacja jest dla globalnej macierzy C

Macierz globalna C															
Wyniki testu															
674.074	337.037	0	0	337.037	168.519	0	0	0	0	0	0	0	0	0	0
337.037	1348.15	337.037	0	168.519	674.074	168.519	0	0	0	0	0	0	0	0	0
0	337.037	1348.15	337.037	0	168.519	674.074	168.519	0	0	0	0	0	0	0	0
0	0	337.037	674.074	0	0	168.519	337.037	0	0	0	0	0	0	0	0
337.037	168.519	0	0	1348.15	674.074	0	0	337.037	168.519	0	0	0	0	0	0
168.519	674.074	168.519	0	674.074	2696.3	674.074	0	168.519	674.074	168.519	0	0	0	0	0
0	168.519	674.074	168.519	0	674.074	2696.3	674.074	0	168.519	674.074	168.519	0	0	0	0
0	0	168.519	337.037	0	0	674.074	1348.15	0	0	168.519	337.037	0	0	0	0
0	0	0	0	337.037	168.519	0	0	1348.15	674.074	0	0	337.037	168.519	0	0
0	0	0	0	168.519	674.074	168.519	0	674.074	2696.3	674.074	0	168.519	674.074	168.519	0
0	0	0	0	0	168.519	674.074	168.519	0	674.074	2696.3	674.074	0	168.519	674.074	168.519
0	0	0	0	0	0	168.519	337.037	0	0	674.074	1348.15	0	0	168.519	337.037
0	0	0	0	0	0	0	0	337.037	168.519	0	0	674.074	337.037	0	0
0	0	0	0	0	0	0	0	168.519	674.074	168.519	0	337.037	1348.15	337.037	0
0	0	0	0	0	0	0	0	0	168.519	674.074	168.519	0	337.037	1348.15	337.037
0	0	0	0	0	0	0	0	0	0	168.519	337.037	0	0	337.037	674.074
Wyniki programu															
674.074	337.037	0	0	337.037	168.519	0	0	0	0	0	0	0	0	0	0
337.037	1348.15	337.037	0	168.519	674.074	168.519	0	0	0	0	0	0	0	0	0
0	337.037	1348.15	337.037	0	168.519	674.074	168.519	0	0	0	0	0	0	0	0
0	0	337.037	674.074	0	0	168.519	337.037	0	0	0	0	0	0	0	0
337.037	168.519	0	0	1348.15	674.074	0	0	337.037	168.519	0	0	0	0	0	0
168.519	674.074	168.519	0	674.074	2696.3	674.074	0	168.519	674.074	168.519	0	0	0	0	0
0	168.519	674.074	168.519	0	674.074	2696.3	674.074	0	168.519	674.074	168.519	0	0	0	0
0	0	168.519	337.037	0	0	674.074	1348.15	0	0	168.519	337.037	0	0	0	0
0	0	0	0	337.037	168.519	0	0	1348.15	674.074	0	0	337.037	168.519	0	0
0	0	0	0	168.519	674.074	168.519	0	674.074	2696.3	674.074	0	168.519	674.074	168.519	0
0	0	0	0	0	168.519	674.074	168.519	0	674.074	2696.3	674.074	0	168.519	674.074	168.519
0	0	0	0	0	0	168.519	337.037	0	0	674.074	1348.15	0	0	168.519	337.037
0	0	0	0	0	0	0	0	337.037	168.519	0	0	674.074	337.037	0	0
0	0	0	0	0	0	0	0	168.519	674.074	168.519	0	337.037	1348.15	337.037	0
0	0	0	0	0	0	0	0	0	168.519	674.074	168.519	0	337.037	1348.15	337.037
0	0	0	0	0	0	0	0	0	0	168.519	337.037	0	0	337.037	674.074

Zestawiając ze sobą wyniki minimalnych oraz maksymalnych temperatur dla danego kroku zauważyć można, że dane testowe zgadzają się z wynikami programu, co pokazuje, że dla siatki kwadratowej 4x4 udało się obliczyć temperatury poprawnie.

Wyniki temperatur			
Wyniki testu		Wyniki programu	
110.03797659406167	365.8154705784631	110.038	365.815
168.83701715655656	502.5917120896439	168.837	502.592
242.80085524391868	587.372666691486	242.801	587.373
318.61459376004086	649.3874834542602	318.615	649.388
391.2557916738893	700.0684204214381	391.256	700.068
459.03690325635404	744.0633443187048	459.037	744.063
521.5862742337766	783.382849723737	521.586	783.383
579.0344449687701	818.9921876836681	579.034	818.992
631.6892368621455	851.4310425916341	631.689	851.431
679.9075931513394	881.057634906017	679.908	881.058

Siatka 4x4 mix

W przypadku pliku Test2\_4\_4\_MixGrid nie mamy do czynienia z kwadratem, dlatego wartości lokalne macierzy H będą różne dla elementów.

Macierz H								
Wyniki testu					Wyniki programu			
Element 1								
17.7624	-3.39971	-10.963	-3.39972		17.7702	-3.40951	-10.9511	-3.40951
-3.39971	14.6508	-5.14961	-6.10152		-3.40951	14.6632	-5.16446	-6.08919
-10.963	-5.14961	21.2622	-5.14961		-10.9511	-5.16446	21.2801	-5.16445
-3.39972	-6.10152	-5.14961	14.6508		-3.40951	-6.08919	-5.16445	14.6632
Element 2								
20.9584	-5.19066	-13.2722	-2.49555		20.9637	-5.19733	-13.2652	-2.5012
-5.19066	14.138	-4.44553	-4.50181		-5.19733	14.1464	-4.4543	-4.49474
-13.2722	-4.44553	23.3769	-5.65921		-13.2652	-4.4543	23.3861	-5.66663
-2.49555	-4.50181	-5.65921	12,6566		-2.5012	-4.49474	-5.66663	12,6626
Element 3								
21.1279	-5.49125	-12.3674	-3.26933		21.1474	-5.5156	-12.3479	-3.28383
-5.49125	16.2136	-5.49125	-5.23115		-5.5156	16.2442	-5.5156	-5.21297
-12.3674	-5.49125	21.1279	-3.26933		-12.3479	-5.5156	21.1474	-3.28383
-3.26933	-5.23115	-3.26933	11,7698		-3.28383	-5.21297	-3.28383	11,7698
Element 4								
20.9584	-2.49555	-13.2722	-5.19067		20.9637	-2.50119	-13.2652	-5.19734
-2.49555	12,6566	-5.65921	-4.50181		-2.50119	12,6626	-5.66663	-4.49474
-13.2722	-5.65921	23.3769	-4.44552		-13.2652	-5.66663	23.3861	-4.45429
-5.19067	-4.50181	-4.44552	14.138		-5.19734	-4.49474	-4.45429	14.1464

Element 5							
24.4398	-4.61748	-15.2049	-4.61748		24.4398	-4.61748	-15.2049
-4.61748	12,5	-4.61748	-3.26505		-4.61748	12,5	-4.61748
-15.2049	-4.61748	24.4398	-4.61748		-15.2049	-4.61748	24.4398
-4.61748	-3.26505	-4.61748	12,5		-4.61748	-3.26505	-4.61748
							12,5
Element 6							
23.3769	-4.44553	-13.2722	-5.65921		23.3861	-4.4543	-13.2652
-4.44553	14.138	-5.19066	-4.50181		-4.4543	14.1464	-5.19733
-13.2722	-5.19066	20.9584	-2.49555		-13.2652	-5.19733	20.9637
-5.65921	-4.50181	-2.49555	12,6566		-5.66663	-4.49474	-2.5012
							12,6626
Element 7							
21.1279	-3.26933	-12.3674	-5.49125		21.1474	-3.28383	-12.3479
-3.26933	11,7698	-3.26933	-5.23115		-3.28383	11,7698	-3.28383
-12.3674	-3.26933	21.1279	-5.49125		-12.3479	-3.28383	21.1474
-5.49125	-5.23115	-5.49125	16.2136		-5.5156	-5.21297	-5.5156
							16.2442
Element 8							
23.3769	-5.65921	-13.2722	-4.44553		23.3861	-5.66663	-13.2652
-5.65921	12,6566	-2.49555	-4.50181		-5.66663	12,6626	-2.5012
-13.2722	-2.49555	20.9584	-5.19066		-13.2652	-2.5012	20.9637
-4.44553	-4.50181	-5.19066	14.138		-4.45429	-4.49474	-5.19734
							14.1464
Element 9							
21.2622	-5.14961	-10.963	-5.14961		21.2801	-5.16446	-10.9511
-5.14961	14.6508	-3.39971	-6.10152		-5.16446	14.6632	-3.40951
-10.963	-3.39971	17.7624	-3.39971		-10.9511	-3.40951	17.7702
-5.14961	-6.10152	-3.39971	14.6508		-5.16446	-6.08919	-3.40951
							14.6632

Macierz HBC								
Wyniki testu					Wyniki programu			
Element 1								
9.06164	2.26541	0	2.26541		9.06164	2.26541	0	2.26541
2.26541	4.53082	0	0		2.26541	4.53082	0	0
0	0	0	0		0	0	0	0
2.26541	0	0	4.53082		2.26541	0	0	4.53082
Element 2								
3.20377	1.60189	0	0		3.20377	1.60189	0	0
1.60189	3.20377	0	0		1.60189	3.20377	0	0
0	0	0	0		0	0	0	0
0	0	0	0		0	0	0	0
Element 3								
2.26541	1.1327	0	0		2.26541	1.1327	0	0
1.1327	4.53082	1.1327	0		1.1327	4.53082	1.1327	0
0	1.1327	2.26541	0		0	1.1327	2.26541	0
0	0	0	0		0	0	0	0

Element 4								
3.20377	0	0	1.60189		3.20377	0	0	1.60189
0	0	0	0		0	0	0	0
0	0	0	0		0	0	0	0
1.60189	0	0	3.20377		1.60189	0	0	3.20377
Element 5								
0	0	0	0		0	0	0	0
0	0	0	0		0	0	0	0
0	0	0	0		0	0	0	0
0	0	0	0		0	0	0	0
Element 6								
0	0	0	0		0	0	0	0
0	3.20377	1.60189	0		0	3.33333	1.66667	0
0	1.60189	3.20377	0		0	1.66667	3.33333	0
0	0	0	0		0	0	0	0
Element 7								
2.26541	0	0	1.1327		2.26541	0	0	1.1327
0	0	0	0		0	0	0	0
0	0	2.26541	1.1327		0	0	2.26541	1.1327
1.1327	0	1.1327	4.53082		1.1327	0	1.1327	4.53082
Element 8								
0	0	0	0		0	0	0	0
0	0	0	0		0	0	0	0
0	0	3.20377	1.60189		0	0	3.20377	1.60189
0	0	1.60189	3.20377		0	0	1.60189	3.20377
Element 9								
0	0	0	0		0	0	0	0
0	4.53082	2.26541	0		0	4.53082	2.26541	0
0	2.26541	9.06164	2.26541		0	2.26541	9.06164	2.26541
0	0	2.26541	4.53082		0	0	2.26541	4.53082

Wektor P								
Wyniki testu					Wyniki programu			
Element 1								
16310.9	8155.47	0	8155.47		16310.9	8155.47	0	8155.47
Element 2								
5766.79	5766.79	0	0		5766.79	5766.79	0	0
Element 3								
4077.74	8155.47	4077.74	0		4077.74	8155.47	4077.74	0
Element 4								
5766.79	0	0	5766.79		5766.79	0	0	5766.79
Element 5								
0	0	0	0		0	0	0	0

Element 6							
0	5766.79	5766.79	0	0	5766.79	5766.79	0
Element 7							
4077.74	0	4077.74	8155.47	4077.74	0	4077.74	8155.47
Element 8							
0	0	5766.79	5766.79	0	0	5766.79	5766.79
Element 9							
0	8155.47	16310.9	8155.47	0	8155.47	16310.9	8155.47

Macierz C								
Wyniki testu					Wyniki programu			
Element								
1139.59	543.343	258.447	543.343		1139.59	543.343	258.447	543.343
543.343	1033.79	490.444	258.447		543.343	1033.79	490.444	258.447
258.447	490.444	927.988	490.444		258.447	490.444	927.988	490.444
543.343	258.447	490.444	1033.79		543.343	258.447	490.444	1033.79
Element 2								
687.257	325.995	160.879	339.392		687.257	325.995	160.879	339.392
325.995	616.725	304.125	160.879		325.995	616.725	304.125	160.879
160.879	304.125	599.777	317.522		160.879	304.125	599.777	317.522
339.392	160.879	317.522	670.309		339.392	160.879	317.522	670.309
Element 3								
417.145	195.348	104.286	221.797		417.145	195.348	104.286	221.797
195.348	364.246	195.348	104.286		195.348	364.246	195.348	104.286
104.286	195.348	417.145	221.797		104.286	195.348	417.145	221.797
221.797	104.286	221.797	470.045		221.797	104.286	221.797	470.045
Element 4								
687.257	339.392	160.879	325.995		687.257	339.392	160.879	325.995
339.392	670.309	317.522	160.879		339.392	670.309	317.522	160.879
160.879	317.522	599.777	304.125		160.879	317.522	599.777	304.125
325.995	160.879	304.125	616.725		325.995	160.879	304.125	616.725
Element 5								
590.735	295.368	147.684	295.368		590.735	295.368	147.684	295.368
295.368	590.735	295.368	147.684		295.368	590.735	295.368	147.684
147.684	295.368	590.735	295.368		147.684	295.368	590.735	295.368
295.368	147.684	295.368	590.735		295.368	147.684	295.368	590.735
Element 6								
599.777	304.125	160.879	317.522		599.777	304.125	160.879	317.522
304.125	616.725	325.995	160.879		304.125	616.725	325.995	160.879
160.879	325.995	687.257	339.392		160.879	325.995	687.257	339.392
317.522	160.879	339.392	670.309		317.522	160.879	339.392	670.309

Element 7							
417.145	221.797	104.286	195.348		417.145	221.797	104.286
221.797	470.045	221.797	104.286		221.797	470.045	221.797
104.286	221.797	417.145	195.348		104.286	221.797	417.145
195.348	104.286	195.348	364.246		195.348	104.286	195.348
Element 8							
599.777	317.522	160.879	304.125		599.777	317.522	160.879
317.522	670.309	339.392	160.879		317.522	670.309	339.392
160.879	339.392	687.257	325.995		160.879	339.392	687.257
304.125	160.879	325.995	616.725		304.125	160.879	325.995
Element 9							
927.988	490.444	258.447	490.444		927.988	490.444	258.447
490.444	1033.79	543.343	258.447		490.444	1033.79	543.343
258.447	543.343	1139.59	543.343		258.447	543.343	1139.59
490.444	258.447	543.343	1033.79		490.444	258.447	543.343

Jak widać w poszczególnych macierzach pojawiają się większe różnorodności niż w przypadku pierwszego pliku, jest to przede wszystkim spowodowane różnicą w wyglądzie siatki, w tym przypadku nasza siatka nie jest już prostą figurą.

Ponownie zestawiając ze sobą wyniki minimalnych oraz maksymalnych temperatur dla danego kroku zauważyć można, że program radzi sobie również z obliczeniami dla bardziej skomplikowanych figur.

Wyniki temperatur			
Wyniki testu		Wyniki programu	
95.15184673458245	374.6863325385064	95.1591	374.668
147.64441665454345	505.96811082245307	147.656	505.954
220.1644549730314	586.9978503916302	220.178	586.989
296.7364399006366	647.28558387732	296.751	647.28
370.968275802604	697.3339863103786	370.983	697.33
440.5601440058566	741.2191121514377	440.574	741.216
504.8911996551285	781.209569726045	504.904	781.241
564.0015111915015	817.3915065469778	564.014	817.42
618.1738556427995	850.2373194670416	618.185	850.264
667.7655470268747	880.1676054000437	667.776	880.192

### Siatka 31x31 kwadrat

W tym pliku testowym pojawia się nam dużo więcej węzłów oraz elementów w siatce, co może sprawić trudność dla naszego programu, jednak porównując wyniki minimalne oraz maksymalne dla danego kroku możemy zauważyć, że wyniki są bardzo zbliżone do siebie, różnice po przecinku mogą się pojawić przez przybliżenia, które mogą być różne na różnych maszynach.

Wyniki temperatur			
Wyniki testu		Wyniki programu	
99.99969812978378	149.5566275788947	100	149.557
100.00053467957446	177.44482649738018	100	177.445
100.00084733335379	197.2672291500534	100	197.267
100.00116712763896	213.15348263983788	100	213.153
100.00150209858216	226.6837398631218	100	226.683
100.001852708951	238.60869878203812	100	238.607
100.00222410506852	249.34880985057373	100	249.347
100.00263047992797	259.1676797521773	100	259.165
100.00310216686808	268.24376548847937	100	268.241
100.00369558647527	276.70463950306436	100	276.701
100.00450560745507	284.64527660833346	100.001	284.641
100.00567932588369	292.1386492100023	100.002	292.134
100.00742988613344	299.242260871447	100.003	299.237
100.01004886564658	306.00237684844643	100.005	305.997
100.01391592562979	312.4568735346492	100.009	312.451
100.01950481085419	318.637221302136	100.014	318.631
100.02738525124852	324.56990275925733	100.021	324.564
100.0382207726261	330.27745133351596	100.032	330.271
100.05276279329537	335.77922748329735	100.046	335.772
100.07184163487159	341.0920092636545	100.064	341.085

### Siatka 31x31 trapez

W tym pliku testowym pomimo zwiększonej ilości węzłów oraz elementów pojawia się dodatkowo inny kształt, program musi sobie dodatkowo poradzić z bardziej skomplikowanymi przekształceniami niż w przypadku kwadratu. Ponownie porównując wyniki można zauważyć zbliżone wyniki, z pewnym błędem przybliżenia po przecinku.

Wyniki temperatur			
Wyniki testu		Wyniki programu	
99.99911415177323	166.9362149651147	100	166.936
99.99873673857435	207.23241215252318	100	207.233
99.99913622824398	236.28484836489392	100	236.287
99.99886349097673	259.4615454293437	100	259.465
99.99856368769406	279.02621207925847	100	279.031
99.99833307076041	296.11440465672047	100	296.121
99.99828777060439	311.37745804357013	100.001	311.385
99.9986389011151	325.22693428231423	100.001	325.235

99.999733155009	337.94169376607823	100.003	337.951
100.00209185746898	349.72071897716853	100.005	349.731
100.0064419788215	360.71173731817754	100.01	360.723
100.01373418237196	371.02790901578555	100.018	371.04
100.02514547372809	380.7581368613588	100.03	380.771
100.04206658948135	389.9737401779163	100.047	389.987
100.06607630236047	398.7329444084316	100.072	398.747
100.09890604703071	407.0839999228044	100.105	407.099
100.14239869485296	415.06740851188135	100.149	415.083
100.19846510735009	422.71755059301114	100.205	422.734
100.2690415128534	430.063898923412	100.276	430.081
100.35604999410768	437.13194022620934	100.364	437.15
100.46136360581374	443.9438861744528	100.47	443.962
100.58677695817498	450.51922965500603	100.596	450.538
100.73398255211671	456.8751855104281	100.743	456.895
100.90455274455185	463.02704374695475	100.914	463.047
101.09992694675618	468.9884555278435	101.11	469.009
101.32140349391727	474.77166692761165	101.332	474.793
101.57013554447936	480.38771163654076	101.582	480.41
101.84713035050619	485.8465710811325	101.859	485.869
102.15325126484217	491.1573084394593	102.166	491.18
102.48922190137662	496.328181562518	102.502	496.352
102.85563192917733	501.3667387153518	102.869	501.391
103.25294405113215	506.2799002224972	103.267	506.304
103.68150178722517	511.07402846921707	103.696	511.099
104.14153774807718	515.7549882221302	104.157	515.78
104.63318214384935	520.3281988536518	104.649	520.354
105.15647132609867	524.7986797574265	105.173	524.825
105.71135620545562	529.171090007135	105.728	529.198
106.29771042636658	533.4497631244197	106.315	533.477
106.91533821210513	537.6387376719981	106.933	537.666
107.56398181948711	541.741784267583	107.582	541.77
108.24332856395998	545.7624295164479	108.262	545.791
108.9530173926518	549.7039772807208	108.973	549.733
109.69264499625561	553.5695276382133	109.713	553.598
110.46177146086819	557.3619938296619	110.482	557.391
111.259925468644	561.0841174486852	111.281	561.114
112.08660906181586	564.738482091664	112.108	564.769
112.94130198867555	568.3275256537182	112.964	568.358
113.82346565282369	571.8535514309826	113.846	571.884
114.7325466886738	575.3187381674231	114.756	575.35
115.66798018705074	578.7251491659451	115.692	578.757
116.62919259496024	582.074740567812	116.654	582.107
117.61560431334811	585.3693688909699	117.641	585.402
118.62663201608873	588.6107979064777	118.652	588.643

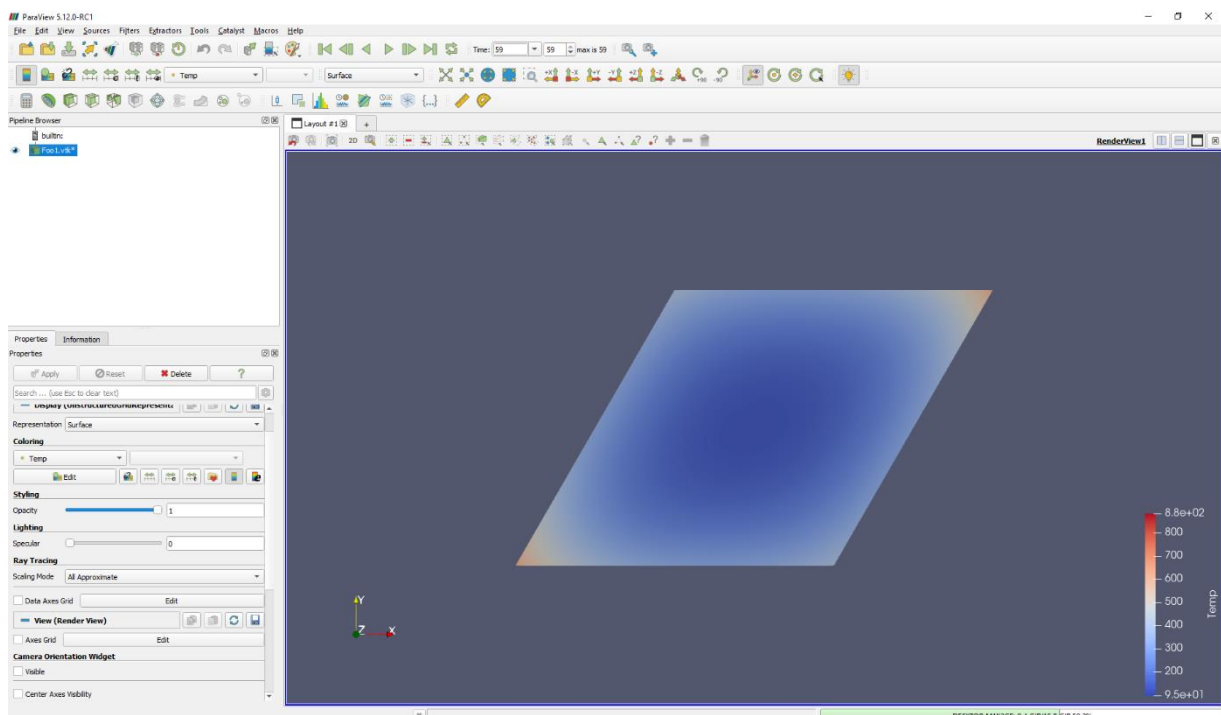
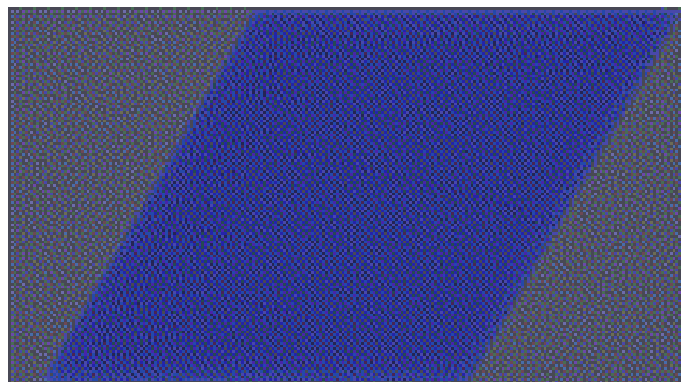
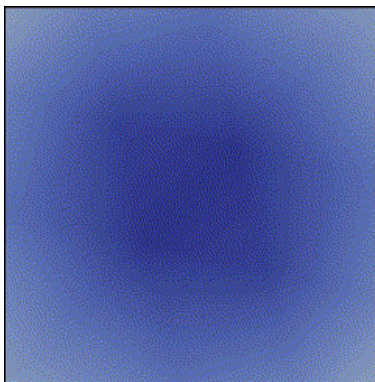


119.66169071257625	591.8007049224022	119.688	591.834
120.72019557527858	594.9406865361285	120.747	594.974
121.801563552473	598.0322639087257	121.829	598.066
122.90521478518018	601.0768876087712	122.933	601.111
124.0305738460866	604.0759420675566	124.059	604.11
125.17707081701873	607.0307496828312	125.206	607.065
126.34414222032	609.9425746041862	126.374	609.977

## Wnioski

Jak można zobaczyć z powyższego porównania wyniki otrzymane z napisanego programu są bardzo zbliżone do danych testowych, co świadczy o poprawności działania programu. Dodatkowo program radzi sobie z bardziej skomplikowanymi siatkami, co dobrze pokazuje przykład siatki trapezu 31x31.

Dzięki opcji generowania plików do programu ParaView możemy zilustrować naszą symulację. Pomimo podglądu rozkładu temperatury w elemencie możemy również ją zilustrować przy pomocy animacji.



Realizacja programu pokazuje w jaki sposób można rozwiązać równania różniczkowe przez użyciu metod numerycznych. Obecny program ma możliwość dalszego rozwoju, możliwe są np. dodanie różnych rodzajów materiałów w siatce o różnych własnościach, różne warunki brzegowe na różnych ścianach, współpraca z innymi programami zewnętrznymi. Dodatkowo program ten rozwiązuje jeden specyficzny problem, wykorzystując wiedzę jaką się uzyskało oraz umiejętności matematyczne oraz metod numerycznych, można napisać inne programu rozwiązujące inne zagadnienia fizyki jak np., inne rodzaje transportu ciepła itp.

Dzięki dodatkowym mechanikom debugowania wyników w kodzie mamy możliwość lepszego podglądu naszej symulacji oraz sprawdzenia co się dzieje w poszczególnych węzłach, jakie wartości posiada poszczególny fragment naszego równania. Dodatkowo sam program jest tak zbudowany, że nie musimy zawsze liczyć wszystkie, kod pozwala nam na własne obliczanie tylko fragmentów, zamiast całej temperatury. Dzięki temu, gdybyśmy chcieli poznać tylko wartości macierzy  $H$  wystarczy użyć odpowiedniej metody zamiast uruchamiać całej symulacji.