Finite Element Method - project report

Aleksandra Słysz, laboratory group no. 6.

1. Theoretical introduction.

The aim of the Finite Element Method classes was to write a program for solving a system of equations in matrix form, resulting directly from the differential Fourier equation through several mathematical transformations:

1. Differential equation

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

2. Integral equation

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

3. The matrix form of a system of equations

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

The solution to this system of equations will be the vector of unknown temperatures that we want to determine at all nodes in the grid - these will be the minima of the functional. The grid, constructed from elements, represents a division of the considered body into smaller parts - this operation allows us to determine the temperature at any point inside the element (i.e., also outside the nodes) as a sum of products of all shape functions and corresponding nodal temperatures. The shape functions themselves can be treated as a factor that is taken from the proportions and determines the share that all nodal temperatures have in the temperature at any point between the nodes. The closer the point is to a given node, the more temperature it receives from it. This is naturally related to the fact that heat is transported in the body between individual points, and they interact with each other.

However, in this software, the target results are the temperatures at the nodes. It is worth noting that the above form of the system applies to a steady state, i.e., for a temperature that does not change, for a single time moment. The boundary conditions that may exist for the problem under consideration are: convection boundary condition, radiation boundary condition, and the condition with a specified temperature on the surface of the body. For the analyzed example, we use the first one and then we use the law of convection, according to the formula:

$$q = \alpha(t_{body} - t_{ambient})$$

The convection boundary condition is divided into two parts:

α * tc - HBC matrix,

α * tot - P vector.

The P vector, which receives the known ambient temperature, will be the vector of the free terms of the system of equations. The part with the unknown temperature of the body is combined with the H matrix, which determines the heat transport in the body.

Description of performing integration in the program using the local system and Gauss quadrature.

In the program, the integration of individual components in the equation (matrices H, HBC, vector P, and also C) is performed using Gauss-Legendre quadrature, which consists of pre-tabulated integration points and their weights. The software allows the use of 2, 3, and 4-point integration schemes - the more points used, the more accurate the results should be. Quadratures apply to the local system, in any dimension: 1D, 2D, 3D... However, in this problem, we focus only on 1D and 2D, as 3D would already reflect an abstraction such as a non-existent fourth dimension in reality. Integration over a surface uses a 1D scheme, and integration over a volume - a 2D scheme.

The one-dimensional local system lies in the interval <-1, 1>, has one coordinate - ξ, and has the following shape functions defined:

N1 = 0.5 * (1- ξ), N2 = 0.5 (1+ ξ)

To calculate the integral in the <-1, 1> system, we calculate the values of the function at integration points ξ and then multiply them by appropriate weights. When in the global system without defined integration points, we use interpolation because the shape function values at integration points in both local and global systems are identical, although the integration points are different. There is a certain proportionality here, and the Jacobian of the transformation is the ratio of the global side length to the local side length in the one-dimensional system. Therefore, to calculate the integral in a system other than the local one, we first need to determine the appropriate integration points from the shape functions, then calculate the function values at those points, multiply them by the weights from the local system, and finally multiply by the Jacobian of the transformation, which ensures the conversion of weight lengths to the global system.

2. **Individual elements of the project, created in order to form the system of equations.**

2.1 Matrix H

This is a matrix that determines the heat transport in the element, determined by the material conductivity. Integrating this matrix is quite complicated because its formula contains derivatives of shape functions with respect to x and y. Having x and y as unknowns, we use derivatives of shape functions with respect to ksi and eta to calculate these appropriate derivatives.

Steps for computing the matrix H

1. Calculation of dN/dksi and dN/deta for the two-point and three-point integration scheme.
2. Then we use the given matrix equation:

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

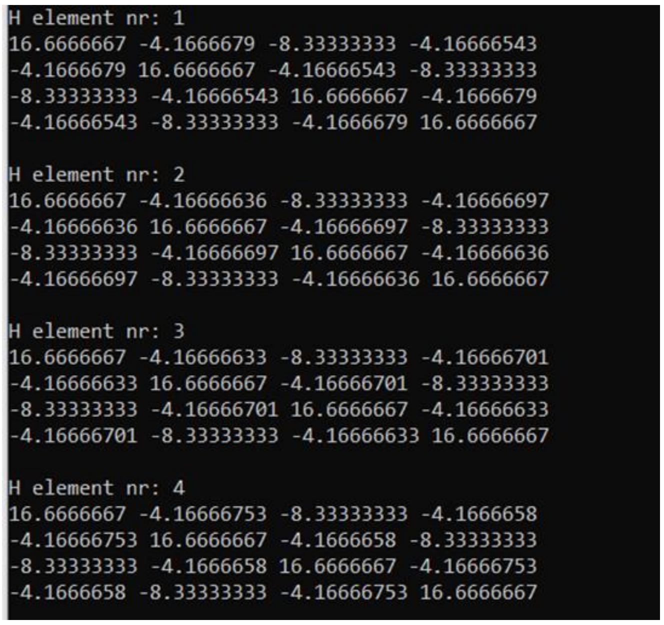
   The central part is the Jacobian matrix, whose components are calculated; we transform the equation so that the unknown part (dN/dx, dN/dy) is on the left side of the equation; for this purpose, we create the inverse Jacobian matrix, in the case of a 2x2 matrix, this is done by changing the signs of the dy/dksi and dx/deta components, as well as swapping the other two components with each other. We then multiply this inverse matrix by 1/detJ, where detJ is the determinant of the Jacobian matrix.
3. We calculate the desired dN/dx and dN/dy (matrix multiplied by vector).
4. To compute the entire matrix at the integration point:

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

   We multiply the shape function derivatives with their corresponding transposes with respect to x and y respectively, and then we sum them up. We multiply the result by the thermal conductivity, and dV is obtained by multiplying it by the previously calculated Jacobian of transformation.

   The local H matrix, which is a matrix of element, is the sum of H matrices from all integration points and regardless of their number, it will always have dimensions of 4x4. The loop over elements contains a loop over integration points; after processing one element and creating the local matrix, it is inserted into a larger array, H_local, which collects local matrices from all elements. It has dimensions of [4 * number of elements] x [4].

   Local H matrices for the 1st grid (fragment):



```
H element nr: 1
16.6666667 -4.1666679 -8.33333333 -4.16666543
-4.1666679 16.6666667 -4.16666543 -8.33333333
-8.33333333 -4.16666543 16.6666667 -4.1666679
-4.16666543 -8.33333333 -4.1666679 16.6666667

H element nr: 2
16.6666667 -4.16666636 -8.33333333 -4.16666697
-4.16666636 16.6666667 -4.16666697 -8.33333333
-8.33333333 -4.16666697 16.6666667 -4.16666636
-4.16666697 -8.33333333 -4.16666636 16.6666667

H element nr: 3
16.6666667 -4.16666633 -8.33333333 -4.16666701
-4.16666633 16.6666667 -4.16666701 -8.33333333
-8.33333333 -4.16666701 16.6666667 -4.16666633
-4.16666701 -8.33333333 -4.16666633 16.6666667

H element nr: 4
16.6666667 -4.16666753 -8.33333333 -4.1666658
-4.16666753 16.6666667 -4.1666658 -8.33333333
-8.33333333 -4.1666658 16.6666667 -4.16666753
-4.1666658 -8.33333333 -4.16666753 16.6666667
```

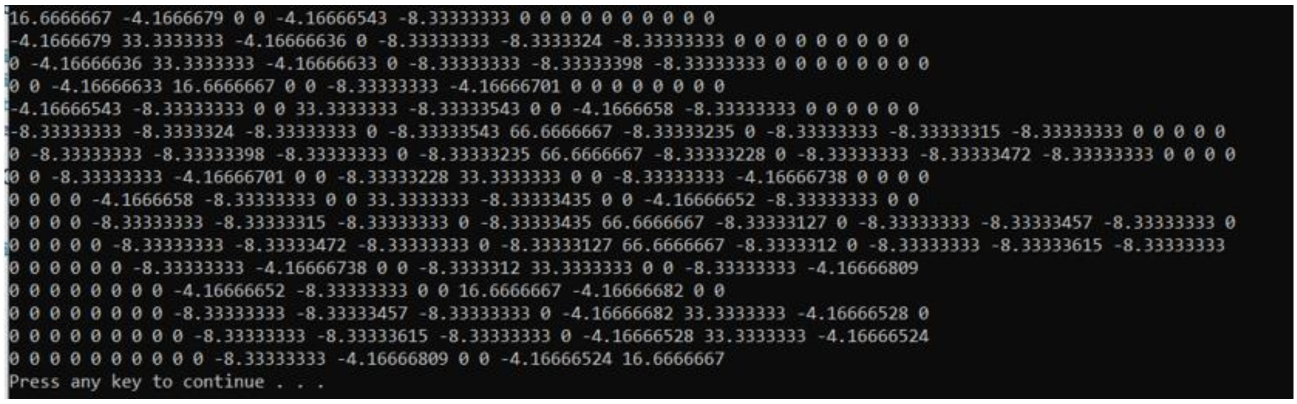
It was noticed that they are the same for each element, because this grid is built of elements that are squares: we only have a change of x and y relative to ksi and eta, which can be seen from the Jacobian matrix.

They are identical for each element. It was different for the second grid, after which the conclusion was drawn that it must be built of quadrilaterals that are neither squares nor rectangles.

2.2 Aggregation

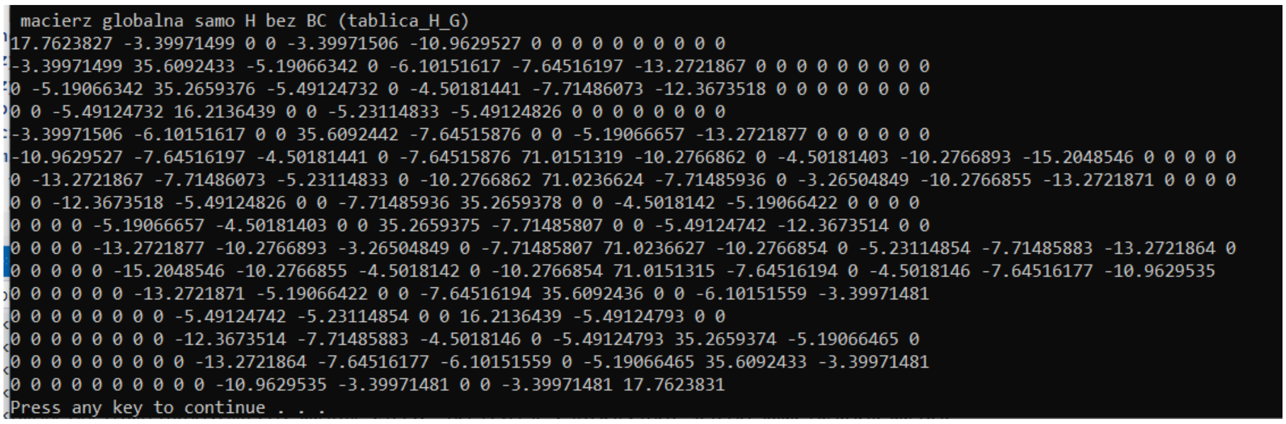
It is a transformation that arranges the local matrices in their appropriate locations in the global matrix (or more precisely, creates this global matrix) of size [number of grid nodes] x [number of grid nodes] using the appropriate mapping. As a result, we obtain a mapping of how heat transport (in the case of matrix H) looks throughout the entire grid, not just in individual elements, through the neighborhood of nodes and their mutual relationships with each other. Where the nodes meet, the corresponding values from H_local were summed up.

Results for the 1st grid:



```
16.6666667 -4.1666679 0 0 -4.16666543 -8.33333333 0 0 0 0 0 0 0 0 0 0
-4.1666679 33.3333333 -4.16666636 0 -8.33333333 -8.3333324 -8.33333333 0 0 0 0 0 0 0 0 0
0 -4.16666636 33.3333333 -4.16666633 0 -8.33333333 -8.33333398 -8.33333333 0 0 0 0 0 0 0 0
0 0 -4.16666633 16.6666667 0 0 -8.33333333 -4.16666701 0 0 0 0 0 0 0 0
-4.16666543 -8.33333333 0 0 33.3333333 -8.33333543 0 0 -4.1666658 -8.33333333 0 0 0 0 0 0
-8.33333333 -8.3333324 -8.33333333 0 -8.33333543 66.6666667 -8.33333235 0 -8.33333333 -8.33333315 -8.33333333 0 0 0 0 0
0 -8.33333333 -8.33333398 -8.33333333 0 -8.33333235 66.6666667 -8.33333228 0 -8.33333333 -8.33333472 -8.33333333 0 0 0 0
0 0 -8.33333333 -4.16666701 0 0 -8.33333228 33.3333333 0 0 -8.33333333 -4.16666738 0 0 0 0
0 0 0 0 -4.1666658 -8.33333333 0 0 33.3333333 -8.33333435 0 0 -4.16666652 -8.33333333 0 0
0 0 0 0 -8.33333333 -8.33333315 -8.33333333 0 -8.33333435 66.6666667 -8.33333127 0 -8.33333333 -8.33333457 -8.33333333 0
0 0 0 0 0 -8.33333333 -8.33333472 -8.33333333 0 -8.33333127 66.6666667 -8.3333312 0 -8.33333333 -8.33333615 -8.33333333
0 0 0 0 0 0 -8.33333333 -4.16666738 0 0 -8.3333312 33.3333333 0 0 -8.33333333 -4.16666809
0 0 0 0 0 0 0 0 -4.16666652 -8.33333333 0 0 16.6666667 -4.16666682 0 0
0 0 0 0 0 0 0 0 -8.33333333 -8.33333457 -8.33333333 0 -4.16666682 33.3333333 -4.16666528 0
0 0 0 0 0 0 0 0 0 -8.33333333 -8.33333615 -8.33333333 0 -4.16666528 33.3333333 -4.16666524
0 0 0 0 0 0 0 0 0 0 -8.33333333 -4.16666809 0 0 -4.16666524 16.6666667
Press any key to continue . . .
```

Results for the 2nd grid:



```
 macierz globalna samo H bez BC (tablica_H_G)
17.7623827 -3.39971499 0 0 -3.39971506 -10.9629527 0 0 0 0 0 0 0 0 0 0
-3.39971499 35.6092433 -5.19066342 0 -6.10151617 -7.64516197 -13.2721867 0 0 0 0 0 0 0 0 0
0 -5.19066342 35.2659376 -5.49124732 0 -4.50181441 -7.71486073 -12.3673518 0 0 0 0 0 0 0 0
0 0 -5.49124732 16.2136439 0 0 -5.23114833 -5.49124826 0 0 0 0 0 0 0 0
-3.39971506 -6.10151617 0 0 35.6092442 -7.64515876 0 0 -5.19066657 -13.2721877 0 0 0 0 0 0
-10.9629527 -7.64516197 -4.50181441 0 -7.64515876 71.0151319 -10.2766862 0 -4.50181403 -10.2766893 -15.2048546 0 0 0 0 0
0 -13.2721867 -7.71486073 -5.23114833 0 -10.2766862 71.0236624 -7.71485936 0 -3.26504849 -10.2766855 -13.2721871 0 0 0 0
0 0 -12.3673518 -5.49124826 0 0 -7.71485936 35.2659378 0 0 -4.5018142 -5.19066422 0 0 0 0
0 0 0 0 -5.19066657 -4.50181403 0 0 35.2659375 -7.71485807 0 0 -5.49124742 -12.3673514 0 0
0 0 0 0 -13.2721877 -10.2766893 -3.26504849 0 -7.71485807 71.0236627 -10.2766854 0 -5.23114854 -7.71485883 -13.2721864 0
0 0 0 0 0 -15.2048546 -10.2766855 -4.5018142 0 -10.2766854 71.0151315 -7.64516194 0 -4.5018146 -7.64516177 -10.9629535
0 0 0 0 0 0 -13.2721871 -5.19066422 0 0 -7.64516194 35.6092436 0 0 -6.10151559 -3.39971481
0 0 0 0 0 0 0 0 -5.49124742 -5.23114854 0 0 16.2136439 -5.49124793 0 0
0 0 0 0 0 0 0 0 -12.3673514 -7.71485883 -4.5018146 0 -5.49124793 35.2659374 -5.19066465 0
0 0 0 0 0 0 0 0 0 -13.2721864 -7.64516177 -6.10151559 0 -5.19066465 35.6092433 -3.39971481
0 0 0 0 0 0 0 0 0 0 -10.9629535 -3.39971481 0 0 -3.39971481 17.7623831
Press any key to continue . . .
```

As a result of aggregation, we obtain global matrices filled with results around the diagonal, with zeros in the remaining places.

2.3. HBC matrix + vector P

I describe them together because they are two parts of a boundary condition, calculated in the same loop but of course with different formulas, although the mechanism is very similar. They use the same values of shape functions and the determinant detJ.

The vector P is the load vector that defines how the ambient temperature affects the nodes. Its characteristic value - the multiplier is the ambient temperature and the heat transfer coefficient, while for HBC only the alpha coefficient is used. We calculate the HBC matrix and P vector for each element face separately, if the condition is met that both nodes forming the face have a Boundary Condition value equal to 1.

In the program, each face is considered separately because each has a different node pair. Choosing the appropriate integration scheme and its appropriate elements (weights and ksi and eta) is done using conditional instructions. Additionally, in the HBC matrix and the P vector, one of the coordinates, either ksi or eta, is always known and equals -1 or 1, resulting from the location of integration points on the element sides.

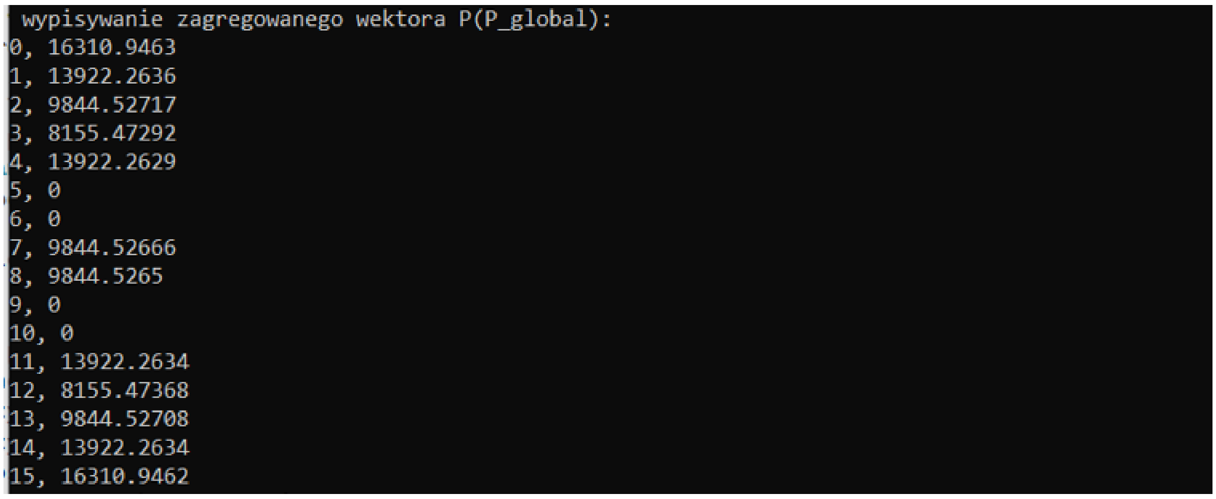
For the top and left faces, I used the reverse order of the shape functions because it turns out that for them, the order of integration points is reversed compared to their position in the table.

To calculate the HBC matrix of one element, we add matrices from integration points on the given face, and then the matrices of all 4 faces, obtaining a matrix of size 4x4.

Similarly, in the case of the P vector, it has a dimension of [1][4]. The local HBC matrix (a set of all elements in one array) is added to the local H matrix, and then their sum is aggregated - because it is the H with the boundary condition that we need in the equation. The P vector is separate and also aggregated.

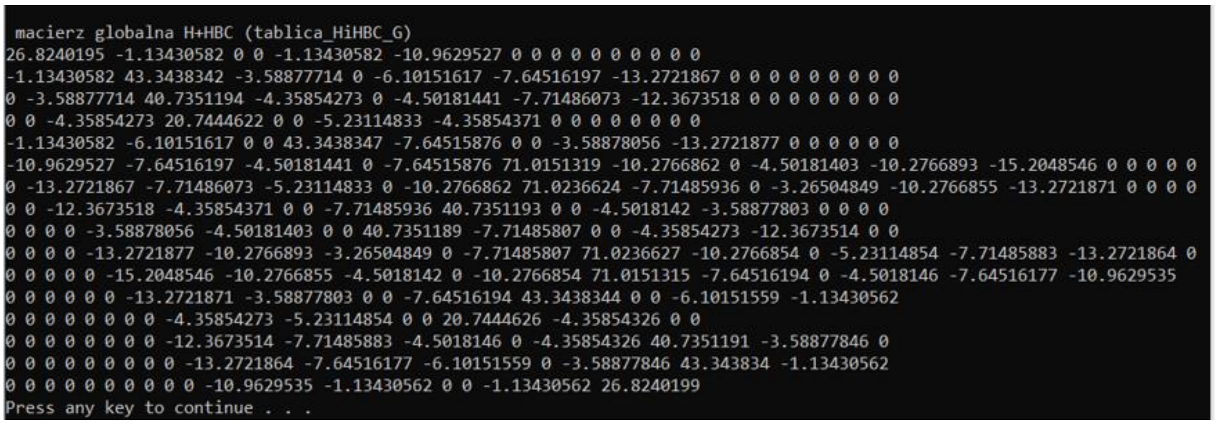
Results for the 2nd grid:

Global P vector:



```
 wypisywanie zagregowanego wektora P(P_global):
0, 16310.9463
1, 13922.2636
2, 9844.52717
3, 8155.47292
4, 13922.2629
5, 0
6, 0
7, 9844.52666
8, 9844.5265
9, 0
10, 0
11, 13922.2634
12, 8155.47368
13, 9844.52708
14, 13922.2634
15, 16310.9462
```

The global HBC matrix combined with the global H matrix:



```
 macierz globalna H+HBC (tablica_HiHBC_G)
26.8240195 -1.13430582 0 0 -1.13430582 -10.9629527 0 0 0 0 0 0 0 0 0 0
-1.13430582 43.3438342 -3.58877714 0 -6.10151617 -7.64516197 -13.2721867 0 0 0 0 0 0 0 0 0
0 -3.58877714 40.7351194 -4.35854273 0 -4.50181441 -7.71486073 -12.3673518 0 0 0 0 0 0 0 0
0 0 -4.35854273 20.7444622 0 0 -5.23114833 -4.35854371 0 0 0 0 0 0 0 0
-1.13430582 -6.10151617 0 0 43.3438347 -7.64515876 0 0 -3.58878056 -13.2721877 0 0 0 0 0 0
-10.9629527 -7.64516197 -4.50181441 0 -7.64515876 71.0151319 -10.2766862 0 -4.50181403 -10.2766893 -15.2048546 0 0 0 0 0
0 -13.2721867 -7.71486073 -5.23114833 0 -10.2766862 71.0236624 -7.71485936 0 -3.26504849 -10.2766855 -13.2721871 0 0 0 0
0 0 -12.3673518 -4.35854371 0 0 -7.71485936 40.7351193 0 0 -4.5018142 -3.58877803 0 0 0 0
0 0 0 0 -3.58878056 -4.50181403 0 0 40.7351189 -7.71485807 0 0 -4.35854273 -12.3673514 0 0
0 0 0 0 -13.2721877 -10.2766893 -3.26504849 0 -7.71485807 71.0236627 -10.2766854 0 -5.23114854 -7.71485883 -13.2721864 0
0 0 0 0 0 -15.2048546 -10.2766855 -4.5018142 0 -10.2766854 71.0151315 -7.64516194 0 -4.5018146 -7.64516177 -10.9629535
0 0 0 0 0 0 -13.2721871 -3.58877803 0 0 -7.64516194 43.3438344 0 0 -6.10151559 -1.13430562
0 0 0 0 0 0 0 0 -4.35854273 -5.23114854 0 0 20.7444626 -4.35854326 0 0
0 0 0 0 0 0 0 0 -12.3673514 -7.71485883 -4.5018146 0 -4.35854326 40.7351191 -3.58877846 0
0 0 0 0 0 0 0 0 0 -13.2721864 -7.64516177 -6.10151559 0 -3.58877846 43.343834 -1.13430562
0 0 0 0 0 0 0 0 0 0 -10.9629535 -1.13430562 0 0 -1.13430562 26.8240199
Press any key to continue . . .
```

Stationary state

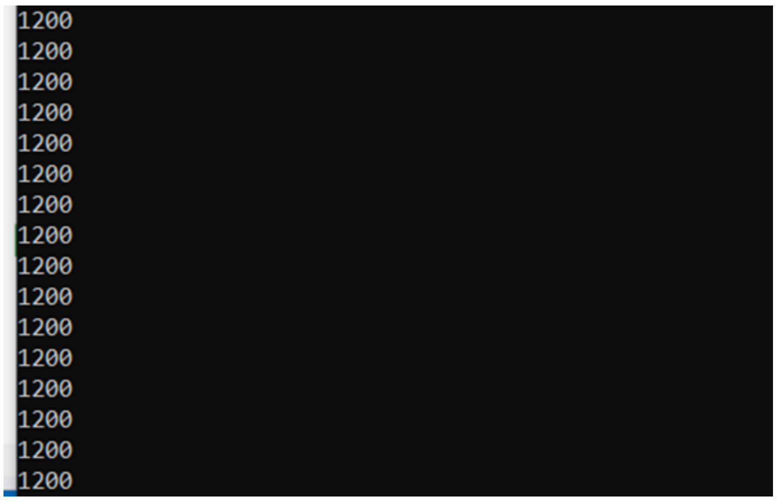
With the above data, it was possible to calculate the solutions of the equation system using the Gaussian elimination method for steady state.

The equation system has the following form:

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

I created an augmented matrix as a preliminary step to the Gauss algorithm - it has dimensions of [number of nodes][number of nodes +1], the last column contains the transcribed P vector, and the rest contain H with HBC. The result obtained is a vector of temperatures {t1}.

Results for the 2nd grid:



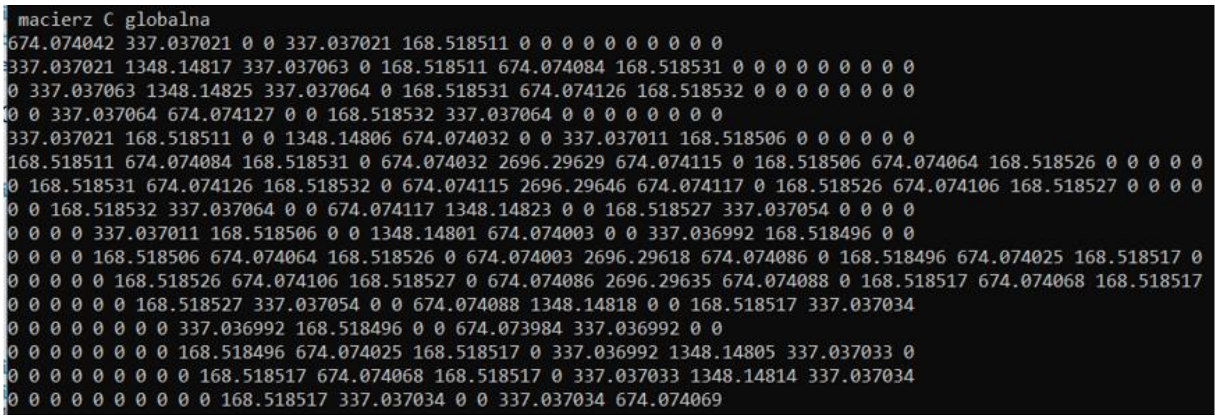
```
1200
1200
1200
1200
1200
1200
1200
1200
1200
1200
1200
1200
1200
1200
1200
1200
```

Transient state. Simulation

Solving the equation system for the unsteady state, i.e., for changes in temperature over time, required first calculating the matrix C, which informs how heat is stored in the element. Its characteristic quantities are the specific heat of the material and the density. The matrix C is calculated in the same loops as the matrix H: they use the same integration points and determinants detJ, operating on the same elements.

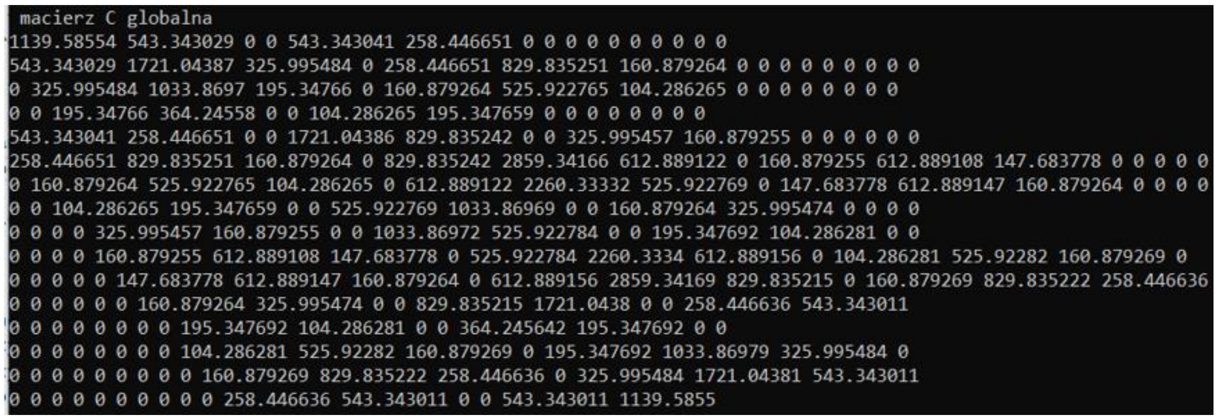
Matrix C.

Results for the 1st grid:



```
 macierz C globalna
674.074042 337.037021 0 0 337.037021 168.518511 0 0 0 0 0 0 0 0 0 0
337.037021 1348.14817 337.037063 0 168.518511 674.074084 168.518531 0 0 0 0 0 0 0 0 0
0 337.037063 1348.14825 337.037064 0 168.518531 674.074126 168.518532 0 0 0 0 0 0 0 0
0 0 337.037064 674.074127 0 0 168.518532 337.037064 0 0 0 0 0 0 0 0
337.037021 168.518511 0 0 1348.14806 674.074032 0 0 337.037011 168.518506 0 0 0 0 0 0
168.518511 674.074084 168.518531 0 674.074032 2696.29629 674.074115 0 168.518506 674.074064 168.518526 0 0 0 0 0
0 168.518531 674.074126 168.518532 0 674.074115 2696.29646 674.074117 0 168.518526 674.074106 168.518527 0 0 0 0
0 0 168.518532 337.037064 0 0 674.074117 1348.14823 0 0 168.518527 337.037054 0 0 0 0
0 0 0 0 337.037011 168.518506 0 0 1348.14801 674.074003 0 0 337.036992 168.518496 0 0
0 0 0 0 168.518506 674.074064 168.518526 0 674.074003 2696.29618 674.074086 0 168.518496 674.074025 168.518517 0
0 0 0 0 0 168.518526 674.074106 168.518527 0 674.074086 2696.29635 674.074088 0 168.518517 674.074068 168.518517
0 0 0 0 0 0 168.518527 337.037054 0 0 674.074088 1348.14818 0 0 168.518517 337.037034
0 0 0 0 0 0 0 0 337.036992 168.518496 0 0 674.073984 337.036992 0 0
0 0 0 0 0 0 0 0 168.518496 674.074025 168.518517 0 337.036992 1348.14805 337.037033 0
0 0 0 0 0 0 0 0 0 168.518517 674.074068 168.518517 0 337.037033 1348.14814 337.037034
0 0 0 0 0 0 0 0 0 0 168.518517 337.037034 0 0 337.037034 674.074069
```

Results for the 2nd grid:



```
 macierz C globalna
1139.58554 543.343029 0 0 543.343041 258.446651 0 0 0 0 0 0 0 0 0 0
543.343029 1721.04387 325.995484 0 258.446651 829.835251 160.879264 0 0 0 0 0 0 0 0 0
0 325.995484 1033.8697 195.34766 0 160.879264 525.922765 104.286265 0 0 0 0 0 0 0 0
0 0 195.34766 364.24558 0 0 104.286265 195.347659 0 0 0 0 0 0 0 0
543.343041 258.446651 0 0 1721.04386 829.835242 0 0 325.995457 160.879255 0 0 0 0 0 0
258.446651 829.835251 160.879264 0 829.835242 2859.34166 612.889122 0 160.879255 612.889108 147.683778 0 0 0 0 0
0 160.879264 525.922765 104.286265 0 612.889122 2260.33332 525.922769 0 147.683778 612.889147 160.879264 0 0 0 0
0 0 104.286265 195.347659 0 0 525.922769 1033.86969 0 0 160.879264 325.995474 0 0 0 0
0 0 0 0 325.995457 160.879255 0 0 1033.86972 525.922784 0 0 195.347692 104.286281 0 0
0 0 0 0 160.879255 612.889108 147.683778 0 525.922784 2260.3334 612.889156 0 104.286281 525.92282 160.879269 0
0 0 0 0 0 147.683778 612.889147 160.879264 0 612.889156 2859.34169 829.835215 0 160.879269 829.835222 258.446636
0 0 0 0 0 0 160.879264 325.995474 0 0 829.835215 1721.0438 0 0 258.446636 543.343011
0 0 0 0 0 0 0 0 195.347692 104.286281 0 0 364.245642 195.347692 0 0
0 0 0 0 0 0 0 0 104.286281 525.92282 160.879269 0 195.347692 1033.86979 325.995484 0
0 0 0 0 0 0 0 0 0 160.879269 829.835222 258.446636 0 325.995484 1721.04381 543.343011
0 0 0 0 0 0 0 0 0 0 258.446636 543.343011 0 0 543.343011 1139.5855
```

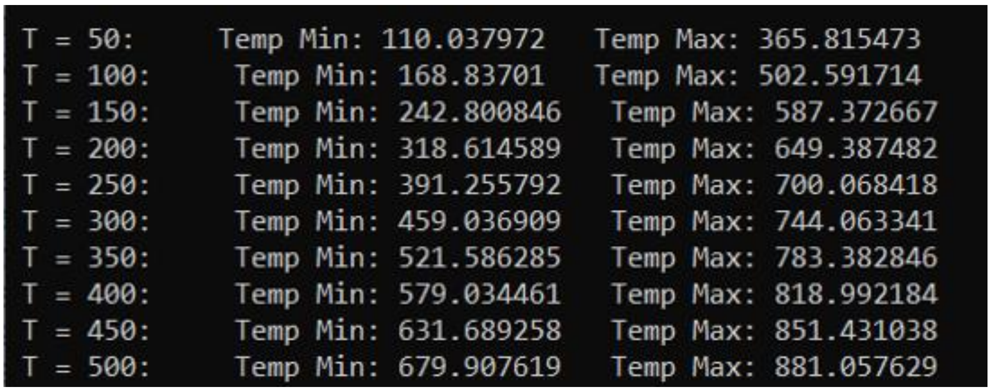
For the non-stationary state, the system of equations has the following form:

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

The right-hand side - with a minus sign, is the entire vector P, which forms the last, 17th column in the augmented matrix. After the first iteration of the loop - the first call of the Gauss method, the calculated temperature vector {t1} is written into the initial temperature vector {t0} of the next iteration; we recalculate both sides of the system (matrix and vector), because both depend on the changing new temperature t0. For each calculated temperature vector, we select the minimum and maximum values.

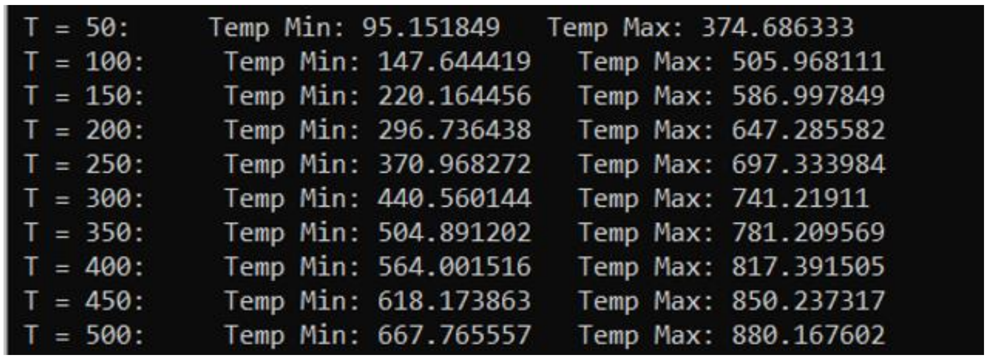
The obtained final temperature results are:

Results for the 1st grid:



```
T = 50:     Temp Min: 110.037972   Temp Max: 365.815473
T = 100:     Temp Min: 168.83701   Temp Max: 502.591714
T = 150:     Temp Min: 242.800846   Temp Max: 587.372667
T = 200:     Temp Min: 318.614589   Temp Max: 649.387482
T = 250:     Temp Min: 391.255792   Temp Max: 700.068418
T = 300:     Temp Min: 459.036909   Temp Max: 744.063341
T = 350:     Temp Min: 521.586285   Temp Max: 783.382846
T = 400:     Temp Min: 579.034461   Temp Max: 818.992184
T = 450:     Temp Min: 631.689258   Temp Max: 851.431038
T = 500:     Temp Min: 679.907619   Temp Max: 881.057629
```

Results for the 2nd grid:



```
T = 50:     Temp Min: 95.151849   Temp Max: 374.686333
T = 100:     Temp Min: 147.644419   Temp Max: 505.968111
T = 150:     Temp Min: 220.164456   Temp Max: 586.997849
T = 200:     Temp Min: 296.736438   Temp Max: 647.285582
T = 250:     Temp Min: 370.968272   Temp Max: 697.333984
T = 300:     Temp Min: 440.560144   Temp Max: 741.21911
T = 350:     Temp Min: 504.891202   Temp Max: 781.209569
T = 400:     Temp Min: 564.001516   Temp Max: 817.391505
T = 450:     Temp Min: 618.173863   Temp Max: 850.237317
T = 500:     Temp Min: 667.765557   Temp Max: 880.167602
```

3. Conclusions

The developed software is a powerful tool that allows not only to solve the system of equations for the steady state, but also to calculate temperature changes over time at nodes, given the initial temperature of the body, ambient temperature, physical parameters of the material, and simulation time data. In principle, this software can model a grid of any size, and even for any shape of the element. The idea behind this method of solving problems is to divide a larger problem into smaller sub-problems and apply interpolation. With the FEM, we can calculate not only temperatures but also other quantities such as stresses and strains.