

1. Ogólny opis metody

Główna idea MES polega na tym, że dowolną ciągłą wartość (np. temperaturę) można zamienić na model dyskretny. Model ten jest oparty na ograniczonej ilości węzłów, które tworzą ograniczoną ilość elementów skończonych.

Algorytm MES:

1. W rozpatrywanym ośrodku bierzemy pod uwagę ograniczoną ilość punktów (węzłów).
2. Wartości temperatury (lub innej funkcji) w każdym węźle definiujemy jako parametr, który musimy wyznaczyć.
3. Strefa wyznaczenia temperatury dzieli się na ograniczoną ilość pod-stref, które nazywamy elementami skończonymi.
4. Temperaturę aproksymuje się na każdym elemencie za pomocą wielomianu, który wyznaczony jest za pomocą węzłowych wartości temperatury. Wyznacza się go w taki sposób, aby zachować warunek ciągłości temperatury na granicach elementów.
5. Węzłowe wartości temperatury muszą być dobrane w taki sposób, aby zapewnić najlepsze do rzeczywistego przybliżenie pola temperatury. Taki dobór wykonywany jest za pomocą minimalizacji funkcjonału, który odpowiada różniczkowemu równaniu przewodzenia ciepła.

2. Opis programu

Program ma na celu obliczenie wartości temperatur we wszystkich węzłach siatki w każdej iteracji czasu.

Żeby tego dokonać potrzebujemy rozwiązać układ równań:

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

gdzie:

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

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

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

$\{t_1\}$ – wektor nieznanymi temperatur, którego szukamy

$\{t_0\}$ – wektor znanych temperatur, wyznaczonych w poprzedniej iteracji

$\Delta\tau$ – krok czasowy

c – ciepło właściwe materiału

ρ – gęstość materiału

α – współczynnik konwekcji

k – współczynnik przewodzenia ciepła

$\{N\}$ – wektor funkcji kształtu

Na początku program wygenerował siatkę, reprezentowaną przez strukturę **Grid**. Składa się ona z elementów(struktura **Element**), które składają się z węzłów(struktura **Node**). Każdy węzeł jest ma określone współrzędne położenia w siatce.

Następnie program wyznacza lokalne macierze H,C oraz wektor P dla każdego elementu.

Wyznaczenie każdej z tych struktur wykorzystuje metodę całkowania Gaussa dla czterech punktów całkowania.

Żeby wyznaczyć macierz [H], najpierw trzeba wyznaczyć macierz [H] po objętości. Algorytm jakim program rozwiązuje ten problem przedstawia się następująco:

Program:

1. Wyznacza współrzędne lokalnych punktów całkowania
2. Oblicza wartość pochodnej funkcji kształtu po ξ i po η w każdym punkcie całkowania
3. Wyznacza składowe jacobianu dla każdego punktu całkowania ze wzorów:

$$\frac{\partial x}{\partial \xi} = \frac{\partial N_1}{\partial \xi} x_1 + \frac{\partial N_2}{\partial \xi} x_2 + \frac{\partial N_3}{\partial \xi} x_3 + \frac{\partial N_4}{\partial \xi} x_4$$

$$\frac{\partial x}{\partial \eta} = \frac{\partial N_1}{\partial \eta} x_1 + \frac{\partial N_2}{\partial \eta} x_2 + \frac{\partial N_3}{\partial \eta} x_3 + \frac{\partial N_4}{\partial \eta} x_4$$

$$\frac{\partial y}{\partial \xi} = \frac{\partial N_1}{\partial \xi} y_1 + \frac{\partial N_2}{\partial \xi} y_2 + \frac{\partial N_3}{\partial \xi} y_3 + \frac{\partial N_4}{\partial \xi} y_4$$

$$\frac{\partial y}{\partial \eta} = \frac{\partial N_1}{\partial \eta} y_1 + \frac{\partial N_2}{\partial \eta} y_2 + \frac{\partial N_3}{\partial \eta} y_3 + \frac{\partial N_4}{\partial \eta} y_4$$

, gdzie x,y to współrzędne węzłów danego elementu w układzie globalnym

4. Liczy wyznacznik każdego jacobianu

5. Odwraca każdy jacobian

6. Oblicza pochodne po x i y każdej funkcji kształtu w każdym punkcie całkowania ze wzoru:

$$\begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{bmatrix} = \frac{1}{\det[J]} \begin{bmatrix} \frac{\partial y}{\partial \eta} & -\frac{\partial y}{\partial \xi} \\ -\frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \xi} \end{bmatrix} \begin{bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{bmatrix}$$

7. Dla każdego punktu całkowania transponuje wektory, mnoży przez współczynnik przewodzenia oraz żeby pozbyć się dV przez wyznacznik jakobianu.

8. Dodaje do siebie powstałe macierze z każdego punktu całkowania w jedną macierz.

Żeby wyznaczyć macierz [H] po powierzchni program:

1. Wyznacza po dwa punkty całkowania dla każdego boku elementu
2. Wyznacza elementy siatki, które stykają się z powierzchnią
3. W zależności od powierzchni liczy wartości niezerowych w danym przypadku funkcji kształtu
4. Dla obu punktów całkowania transponuje wektory i sumuje oba punkty.
5. Otrzymany wynik mnoży przez współczynnik konwekcji i przez jakobian(połowa boku, graniczącego z powierzchnią), żeby pozbyć się ds.

Cała macierz [H] to suma macierzy [H] po dV i [H] po dS .

Algorytm wyznaczania macierzy [C]:

Program:

1. Liczy wartości funkcji kształtu dla wszystkich(czterech) punktów całkowania.
2. Transponuje wektory dla każdego punktu całkowania.
3. Otrzymane z transpozycji macierze mnoży przez wcześniej wyznaczony wyznacznik jakobianu, ciepło właściwe oraz gęstość.
4. Sumuje wartości z każdego punktu całkowania i otrzymuje macierz C.

Algorytm wyznaczania wektora {P}:

Program:

1. Wyznacza po dwa punkty całkowania dla każdego boku elementu
2. Wyznacza elementy siatki, które stykają się z powierzchnią
3. W zależności od powierzchni liczy wartości niezerowych w danym przypadku funkcji kształtu
4. Otrzymany wynik mnoży przez współczynnik konwekcji, temperaturę otoczenia i przez jakobian(połowa boku, graniczącego z powierzchnią), żeby pozbyć się ds.

Gdy mamy wyznaczone powyższe struktury lokalne dla każdego elementu, budujemy z nich struktury globalne korzystając z agregacji.

Algorytm agregacji :

```

for(element = 0 : element = liczba_elementów)
{
    for(i=0:i=4)
    {
        for(j=0:j=4)
        {
            Struktura_globalna[id_wezla[i]][id_wezla[j]] += S_lokalna[i][j]
        }
    }
}

```

Gdy wszystkie struktury są już globalne program wyznacza macierz $[H^{\wedge}]$ oraz wektor $\{P^{\wedge}\}$ dla ułatwienia rozwiązania, gdzie:

$$[H^{\wedge}] = [H] + [C] / \Delta\tau$$

$$\{P^{\wedge}\} = ([C] / \Delta\tau)\{t_0\} + \{P\}$$

Pozostaje rozwiązać układ równań:

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

Program ten układ rozwiązuje za pomocą Gaussa.

Robi to w pętli po iteracjach czasu(krokach czasowych), w skutek czego wyznacza temperatury na każdym węźle w każdej iteracji czasu.

Na końcu oblicza maksymalną i minimalną temperaturę dla każdej iteracji czasu i wypisuje wyniki na ekran.

3. Wyniki dla test_case.pdf:

test_case 1:

Dane:

100 – initial temperature

500 – simulation time [s],

50 – simulation step time [s],

1200 – ambient temperature [C],

300 – alfa $[W/m^2K]$,

0.100 – H [m],

0.100 – B [m],

4 – N_H,

4 – N_B,

700 – specific heat $[J/(kg^{\circ}C)]$,

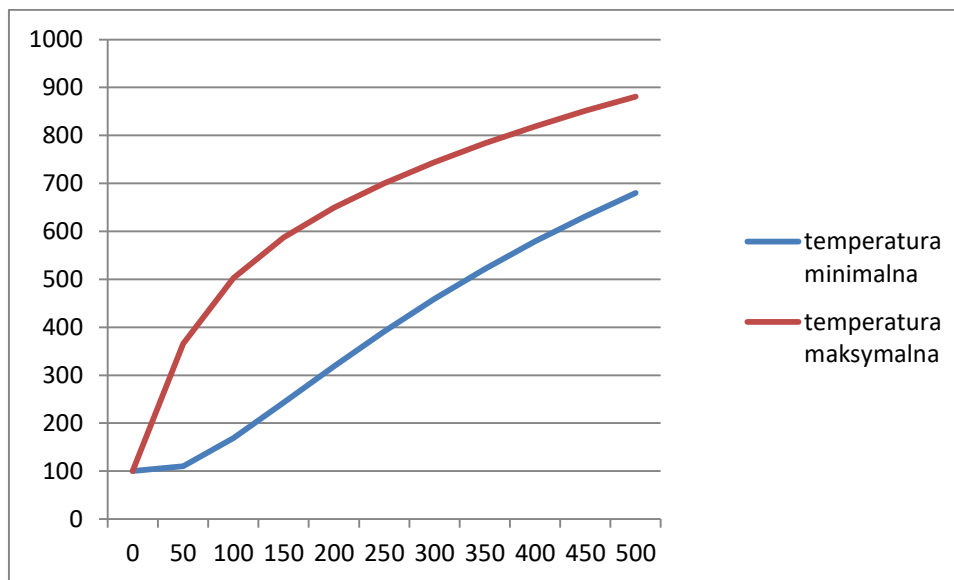
25 – conductivity $[W/(m^{\circ}C)]$,

7800 – density $[kg/m^3]$.

Wyniki wykonania programu:

```
t 1 min: 110.038
t 1 max: 365.815
t 2 min: 168.837
t 2 max: 502.592
t 3 min: 242.801
t 3 max: 587.373
t 4 min: 318.615
t 4 max: 649.387
t 5 min: 391.256
t 5 max: 700.068
t 6 min: 459.037
t 6 max: 744.063
t 7 min: 521.586
t 7 max: 783.383
t 8 min: 579.034
t 8 max: 818.992
t 9 min: 631.689
t 9 max: 851.431
t 10 min: 679.908
t 10 max: 881.058
```

Wykres zależności temperatury od czasu symulacji:



test_case 2:

Dane:

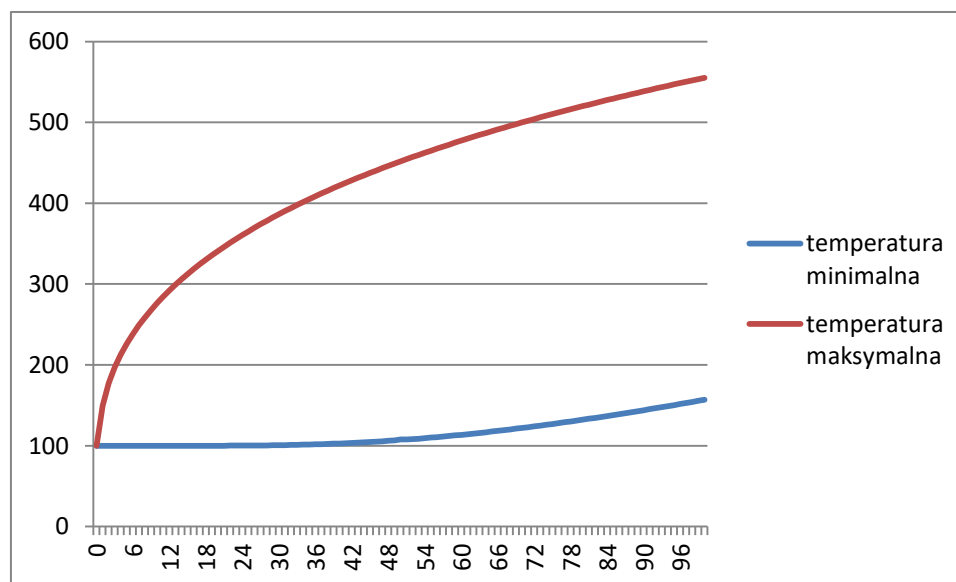
100 – initial temperature
100 – simulation time [s],
1 – simulation step time [s],
1200 – ambient temperature [C],
300 – alfa [$\text{W}/\text{m}^2\text{K}$],
0.100 – H [m],
0.100 – B [m],
31 – N_H,
31 – N_B,
700 – specific heat [$\text{J}/(\text{kg}^\circ\text{C})$],
25 – conductivity [$\text{W}/(\text{m}^\circ\text{C})$],
7800 – density [kg/m^3].

Wyniki wykonania programu:

t 1 min: 100	t 21 min: 100.088	t 45 min: 104.696	t 69 min: 121.105
t 1 max: 149.557	t 21 max: 346.223	t 45 max: 437.52	t 69 max: 497.798
t 2 min: 100	t 22 min: 100.119	t 46 min: 105.13	t 70 min: 122.04
t 2 max: 177.445	t 22 max: 351.199	t 46 max: 440.463	t 70 max: 499.937
t 3 min: 100	t 23 min: 100.157	t 47 min: 105.587	t 71 min: 122.992
t 3 max: 197.267	t 23 max: 356.026	t 47 max: 443.359	t 71 max: 502.052
t 4 min: 100	t 24 min: 100.203	t 48 min: 106.066	t 72 min: 123.962
t 4 max: 213.153	t 24 max: 360.713	t 48 max: 446.211	t 72 max: 504.144
t 5 min: 100	t 25 min: 100.258	t 49 min: 106.567	t 73 min: 124.948
t 5 max: 226.683	t 25 max: 365.269	t 49 max: 449.019	t 73 max: 506.214
t 6 min: 100	t 26 min: 100.325	t 50 min: 107.091	t 74 min: 125.952
t 6 max: 238.607	t 26 max: 369.702	t 50 max: 451.784	t 74 max: 508.262
t 7 min: 100	t 27 min: 100.402	t 51 min: 107.638	t 75 min: 126.971
t 7 max: 249.347	t 27 max: 374.02	t 51 max: 454.509	t 75 max: 510.289
t 8 min: 100	t 28 min: 100.492	t 52 min: 108.206	t 76 min: 128.007
t 8 max: 259.165	t 28 max: 378.229	t 52 max: 457.195	t 76 max: 512.294
t 9 min: 100	t 29 min: 100.596	t 53 min: 108.797	t 77 min: 129.059
t 9 max: 268.241	t 29 max: 382.336	t 53 max: 459.842	t 77 max: 514.28
t 10 min: 100	t 30 min: 100.714	t 54 min: 109.41	t 78 min: 130.126
t 10 max: 276.701	t 30 max: 386.345	t 54 max: 462.452	t 78 max: 516.245
t 11 min: 100.001	t 31 min: 100.847	t 55 min: 110.045	t 79 min: 131.208
t 11 max: 284.641	t 31 max: 390.263	t 55 max: 465.026	t 79 max: 518.191
t 12 min: 100.002	t 32 min: 100.997	t 56 min: 110.701	t 80 min: 132.306
t 12 max: 292.134	t 32 max: 394.093	t 56 max: 467.565	t 80 max: 520.118
t 13 min: 100.003	t 33 min: 101.163	t 57 min: 111.379	t 81 min: 133.417
t 13 max: 299.237	t 33 max: 397.84	t 57 max: 470.07	t 81 max: 522.026
t 14 min: 100.005	t 34 min: 101.347	t 58 min: 112.078	t 82 min: 134.543
t 14 max: 305.997	t 34 max: 401.508	t 58 max: 472.542	t 82 max: 523.916
t 15 min: 100.009	t 35 min: 101.55	t 59 min: 112.799	t 83 min: 135.683
t 15 max: 312.451	t 35 max: 405.101	t 59 max: 474.983	t 83 max: 525.788
t 16 min: 100.014	t 36 min: 101.771	t 60 min: 113.54	t 84 min: 136.836
t 16 max: 318.631	t 36 max: 408.622	t 60 max: 477.392	t 84 max: 527.643
t 17 min: 100.021	t 37 min: 102.012	t 61 min: 114.302	t 85 min: 138.003
t 17 max: 324.564	t 37 max: 412.073	t 61 max: 479.77	t 85 max: 529.48
t 18 min: 100.032	t 38 min: 102.273	t 62 min: 115.084	t 86 min: 139.183
t 18 max: 330.271	t 38 max: 415.459	t 62 max: 482.119	t 86 max: 531.301
t 19 min: 100.046	t 39 min: 102.555	t 63 min: 115.886	t 87 min: 140.375
t 19 max: 335.772	t 39 max: 418.781	t 63 max: 484.44	t 87 max: 533.105
t 20 min: 100.064	t 40 min: 102.858	t 64 min: 116.708	t 88 min: 141.58
t 20 max: 341.085	t 40 max: 422.043	t 64 max: 486.732	t 88 max: 534.893
	t 41 min: 103.182	t 65 min: 117.55	t 89 min: 142.797
	t 41 max: 425.246	t 65 max: 488.997	t 89 max: 536.666
	t 42 min: 103.528	t 66 min: 118.411	t 90 min: 144.026
	t 42 max: 428.393	t 66 max: 491.235	t 90 max: 538.423
	t 43 min: 103.895	t 67 min: 119.29	t 91 min: 145.266
	t 43 max: 431.487	t 67 max: 493.448	t 91 max: 540.165
	t 44 min: 104.285	t 68 min: 120.189	t 92 min: 146.518
	t 44 max: 434.528	t 68 max: 495.635	t 92 max: 541.892

```
t 93 min: 147.781
t 93 max: 543.604
t 94 min: 149.054
t 94 max: 545.302
t 95 min: 150.339
t 95 max: 546.987
t 96 min: 151.633
t 96 max: 548.658
t 97 min: 152.937
t 97 max: 550.315
t 98 min: 154.251
t 98 max: 551.959
t 99 min: 155.575
t 99 max: 553.59
t 100 min: 156.908
t 100 max: 555.209
```

Wykres zależności temperatury od czasu symulacji:



4. Wniosek:

Temperatura minimalna (w środku badanej próbki) z każdą iteracją czasu powiększa się o coraz większą wartość, natomiast temperatura maksymalna (na rogu badanej próbki) z każdą iteracją czasu powiększa się o coraz mniejszą wartość.

