

# Metoda elementów skończonych

**Adamczyk Piotr**

25.01.2020

Grupa 1, Informatyka Stosowana, AGH IMIIP

## Spis treści

Wstęp teoretyczny	1
Elementy składowe kodu	2
Opis działania programu	3
Sprawdzenie poprawności – Test 1	4
Sprawdzenie poprawności – Test 2	5
Przykład rozwiązania rzeczywistego problemu	6
Wnioski i uwagi	9

## Wstęp teoretyczny

### Metoda elementów skończonych

(ang. *finite element method*) – najczęściej używana metoda rozwiązywania problemów inżynierskich i modeli matematycznych. Opiera się na dyskretyzacji układów ciągłych na skończoną liczbę podzbiorów.

### Typ problemu

Problemem który będzie rozwiązywany jest nieustalony transfer ciepła

### Równanie

Poniższe równanie przedstawia rozwiązywany problem

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

gdzie:

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

$$+ \int_S \alpha \{N\} \{N\}^T dS,$$

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

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

$\{t_0\}$  – Wektor temperatur początkowych

$\{t_1\}$  – Wektor temperatur wynikowych

## Elementy składowe kodu

Program składa się z klas głównych:

- `class Node` – zawiera parametry dotyczące węzła, pozycję (  $x, y$  ), temperaturę oraz warunek brzegowy
- `class Element` – zawiera węzły tworzące element, ich globalne indeksy oraz ilość krawędzi posiadających warunek brzegowy
- `class GlobalData` – agregującej klasy:
  - `class Size` – zawierającej długość i szerokość obiektu
  - `class Grid` – zawierającej ilość węzłów wzdłuż i wszerz, liczbę węzłów i liczbę elementów
  - `class Temperature` – zawierającej temperatury początkowe ciała oraz otoczenia
  - `class Time` – zawierającej czas procesu oraz krok czasowy
  - `class Material` – zawierającej dane materiału jak i otoczenia – pojemność cieplną, współczynnik przewodzenia, gęstość ciała oraz współczynnik wymiany ciepła
- `class Grid` – zawiera globalne wektory elementów, węzłów, globalne macierze H oraz C, globalny wektor P
- `class Jacobian` – generuje jakobian, wyznacznik jakobianu oraz odwrotny jakobian dla wybranego punktu
- `class Surface` – tworzy powierzchnię na podstawie obiektu klasy `Element`, wyznacza długość wybranej krawędzi, generuje funkcje kształtu oraz wraca wyznacznik powierzchni.
- `class UniversalElement` – klasa reprezentująca element w układzie lokalnym, zawiera współrzędne w układzie (  $\xi, \eta$  ) oraz wartości funkcji kształtu

oraz klas pomocniczych:

- `class Timer` – zapewnia pomiar czasu wybranego obszaru kodu z dokładnością do 1 mikrosekundy
- `class Utils` – zawiera metodę `solver` zarządzającą procesem obliczeń, metodę odpowiedzialną za rozwiązywanie układu równań wykorzystującego metodę Gaussa oraz generyczną metodę znajdującą minimalną oraz maksymalną wartość w wektorze ze złożonością  $O(n)$

## Opis działania programu

1. Tworzenie klasy `GlobalData` ze wszystkimi parametrami potrzebnymi do rozwiązania problemu
2. Tworzenie siatki oraz węzłów i elementów składających się na nią, tworzenie globalnych macierzy [H], [C] oraz wektora {P}
3. Dla każdej iteracji:
  - a. Zerowanie globalnych macierzy [H], [C] oraz wektora {P}
  - b. Dla każdego elementu:
    - i. Tworzenie lokalnych macierzy
    - ii. Dla każdej pary węzłów generowanie jakobianu, macierzy  $\frac{\partial N}{\partial x}$  i  $\frac{\partial N}{\partial y}$ , interpolacja temperatury funkcjonalem
    - iii. Dla każdego pary węzłów obliczanie całek lokalnych wartości [H], [C] i {P} po objętości oraz zamiana wartości na układ globalny
    - iv. Dla każdej krawędzi z warunkiem brzegowym, wyznaczenie funkcji kształtu w zależności od boku/boków na którym/których występuje warunek brzegowy, sumowanie do lokalnej macierzy [H] i wektora {P} wartości całek po powierzchni oraz zamiana wartości na układ globalny
    - v. Agregacja, dla każdego elementu mapujemy lokalny indeks węzła w elemencie na globalny indeks a następnie sumujemy wartości.
  - c. Rozwiązanie układu równań metodą Gaussa oraz podstawienie wyliczonych wartości temperatur jako temperatury początkowe.
  - d. Wypisanie na ekran odpowiednich wiadomości – maksymalna i minimalna temperatura w danej iteracji, siatki elementów\*, macierzy\*\*

\*, \*\* - możliwe niezależne włączenie/wyłączenie wyświetlania parametrów

Sprawdzenie poprawności – Test 1

Wartości początkowe:

- 100 – initial temperature
- 500 – simulation time [s],
- 50 – simulation step time [s],
- 1200 – ambient temperature [C],
- 300 – alfa [W/m2K],
- 0.100 – H [m],
- 0.100 – B [m],
- 4 – N\_H,
- 4 – N\_B,
- 700 – specific heat [J/(kg °C)],
- 25 – conductivity [W/(m °C)],
- 7800 – density [kg/m3].

Wartości [H] oraz {P} po pierwszej iteracji:

H Matrix ([H]+[C])/dT															
0036.815	0004.241	0000.000	0000.000	0004.241	-004.963	0000.000	0000.000	0000.000	0000.000	0000.000	0000.000	0000.000	0000.000	0000.000	0000.000
0004.241	0066.963	0004.241	0000.000	-004.963	0005.148	-004.963	0000.000	0000.000	0000.000	0000.000	0000.000	0000.000	0000.000	0000.000	0000.000
0000.000	0004.241	0066.963	0004.241	0000.000	-004.963	0005.148	-004.963	0000.000	0000.000	0000.000	0000.000	0000.000	0000.000	0000.000	0000.000
0000.000	0000.000	0004.241	0036.815	0000.000	0000.000	-004.963	0004.241	0000.000	0000.000	0000.000	0000.000	0000.000	0000.000	0000.000	0000.000
0004.241	-004.963	0000.000	0000.000	0066.963	0005.148	0000.000	0000.000	0004.241	-004.963	0000.000	0000.000	0000.000	0000.000	0000.000	0000.000
-004.963	0005.148	-004.963	0000.000	0005.148	0120.593	0005.148	0000.000	-004.963	0005.148	-004.963	0000.000	0000.000	0000.000	0000.000	0000.000
0000.000	-004.963	0005.148	-004.963	0000.000	0005.148	0120.593	0005.148	-004.963	0005.148	-004.963	0000.000	0000.000	0000.000	0000.000	0000.000
0000.000	0000.000	-004.963	0004.241	0000.000	0005.148	0066.963	0000.000	0000.000	-004.963	0004.241	0000.000	0000.000	0000.000	0000.000	0000.000
0000.000	0000.000	0000.000	0000.000	0004.241	-004.963	0000.000	0000.000	0066.963	0005.148	0000.000	0000.000	0004.241	-004.963	0000.000	0000.000
0000.000	0000.000	0000.000	0000.000	-004.963	0005.148	-004.963	0000.000	0005.148	0120.593	0005.148	0000.000	-004.963	0005.148	-004.963	0000.000
0000.000	0000.000	0000.000	0000.000	0000.000	-004.963	0005.148	-004.963	0000.000	0005.148	0120.593	0005.148	0000.000	-004.963	0005.148	-004.963
0000.000	0000.000	0000.000	0000.000	0000.000	0000.000	-004.963	0004.241	0000.000	0000.000	0000.000	0066.963	0000.000	-004.963	0004.241	0000.000
0000.000	0000.000	0000.000	0000.000	0000.000	0000.000	0000.000	0000.000	-004.963	0005.148	-004.963	0000.000	0004.241	0066.963	0004.241	0000.000
0000.000	0000.000	0000.000	0000.000	0000.000	0000.000	0000.000	0000.000	0000.000	-004.963	0005.148	-004.963	0000.000	0004.241	0066.963	0004.241
0000.000	0000.000	0000.000	0000.000	0000.000	0000.000	0000.000	0000.000	0000.000	0000.000	-004.963	0004.241	0000.000	0000.000	0004.241	0036.815
Vector ({P})															
20659.699	25552.224	25552.224	20659.699	25552.224	18897.391	18897.391	25552.224	25552.224	18897.391	18897.391	25552.224	20659.699	25552.224	25552.224	20659.699

Wyznaczone wartości:

Iteration [0] after 50.0 s.	Min: 110.038	Max: 365.815
Iteration [1] after 100.0 s.	Min: 168.837	Max: 502.592
Iteration [2] after 150.0 s.	Min: 242.801	Max: 587.373
Iteration [3] after 200.0 s.	Min: 318.615	Max: 649.387
Iteration [4] after 250.0 s.	Min: 391.256	Max: 700.068
Iteration [5] after 300.0 s.	Min: 459.037	Max: 744.063
Iteration [6] after 350.0 s.	Min: 521.586	Max: 783.383
Iteration [7] after 400.0 s.	Min: 579.034	Max: 818.992
Iteration [8] after 450.0 s.	Min: 631.689	Max: 851.431
Iteration [9] after 500.0 s.	Min: 679.908	Max: 881.058

Czas symulacji MES: 6.825 milisekund (dla AMD Ryzen 9 3900X)

## Sprawdzenie poprawności – Test 2

### Wartości początkowe:

100 – initial temperature  
100 – simulation time [s],  
1 – simulation step time [s],  
1200 – ambient temperature [C],  
300 – alfa [W/m2K],  
0.100 – H [m],  
0.100 – B [m],  
31 – N\_H,  
31 – N\_B,  
700 – specific heat [J/(kg °C)],  
25 – conductivity [W/(m °C)],  
7800 – density [kg/m3].

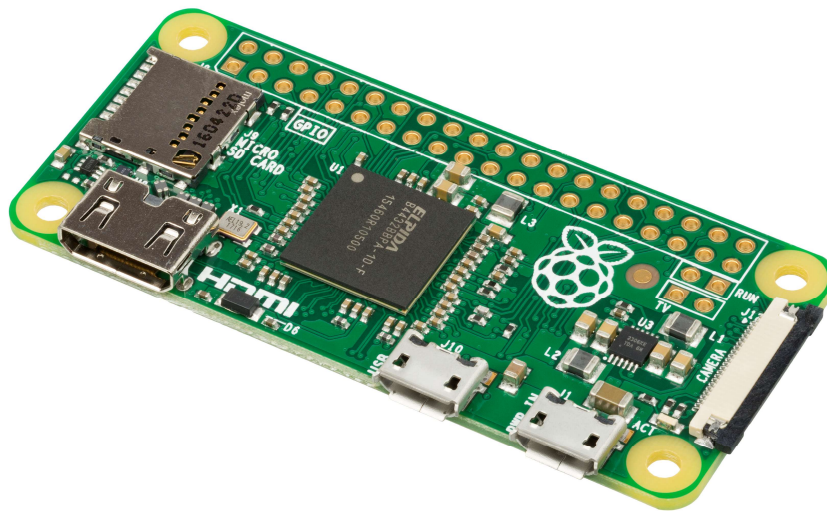
### Wyznaczone wartości:

Iteration [0] after 1.0 s.	Min: 100.000	Max: 149.557
Iteration [1] after 2.0 s.	Min: 100.000	Max: 177.445
Iteration [2] after 3.0 s.	Min: 100.000	Max: 197.267
Iteration [3] after 4.0 s.	Min: 100.000	Max: 213.153
Iteration [4] after 5.0 s.	Min: 100.000	Max: 226.683
Iteration [5] after 6.0 s.	Min: 100.000	Max: 238.607
Iteration [6] after 7.0 s.	Min: 100.000	Max: 249.347
Iteration [7] after 8.0 s.	Min: 100.000	Max: 259.165
Iteration [8] after 9.0 s.	Min: 100.000	Max: 268.241
Iteration [9] after 10.0 s.	Min: 100.000	Max: 276.701
Iteration [10] after 11.0 s.	Min: 100.001	Max: 284.641
Iteration [11] after 12.0 s.	Min: 100.002	Max: 292.134
Iteration [12] after 13.0 s.	Min: 100.003	Max: 299.237
Iteration [13] after 14.0 s.	Min: 100.005	Max: 305.997
Iteration [14] after 15.0 s.	Min: 100.009	Max: 312.451
Iteration [15] after 16.0 s.	Min: 100.014	Max: 318.631
Iteration [16] after 17.0 s.	Min: 100.021	Max: 324.564
Iteration [17] after 18.0 s.	Min: 100.032	Max: 330.271
Iteration [18] after 19.0 s.	Min: 100.046	Max: 335.772
Iteration [19] after 20.0 s.	Min: 100.064	Max: 341.085

Czas symulacji MES: 9.1507 sekund (dla AMD Ryzen 9 3900X)

## Przykład rozwiązania rzeczywistego problemu

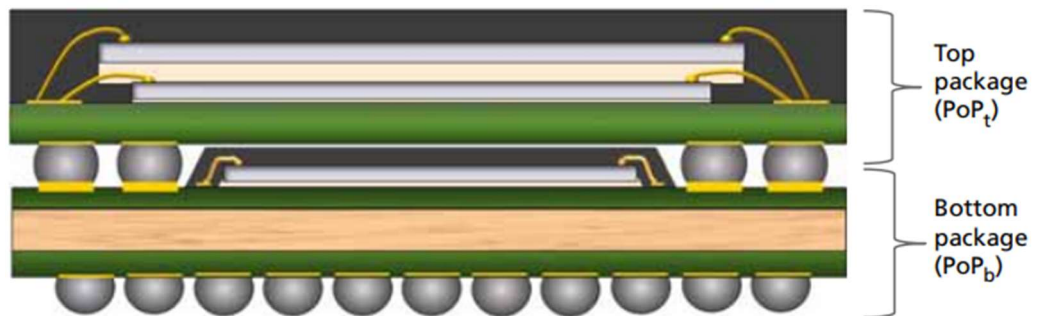
Rzeczywistym problemem który program może rozwiązać jest symulacja procesu przejścia kulek cyny łączących chip metodą PoP (ang. *package-on-package*) w obudowie BGA (ang. *ball grid array*) w stan ciekły przy pomocy ogrzewania chipu za pomocą stacji lutowniczej hot air (strumień gorącego powietrza ogrzewa komponenty na płytce drukowanej). Precyzując problem, zadaniem jest odlutowanie kości pamięci Elpida z płytki Raspberry Pi Zero w celu wymiany jej na model o większej pojemności.



### Raspberry Pi Zero

Ponieważ producent nie zapewnia informacji o typie użytego stopu lutowniczego (domyślna mieszanka  $\text{Sn}_{63}\text{Pb}_{37}$  ma temperaturę topnienia  $183^{\circ}\text{C}$ , natomiast powszechnie używany stop bezołowiowy  $\text{SnAg}_{2,0}$  topi się w temperaturze ok  $225^{\circ}\text{C}$ ), trzeba zatem rozpatrzyć dwa warianty. By unieść chip każda z kulek cyny musi mieć temperaturą nie niższą niż temperatura topnienia, należy jednak pamiętać że przegrzanie układu scalonego może doprowadzić do nieodwracalnego uszkodzenia. Dodatkowo wygrzewając płytkę drukowaną (produkowana jest materiału szklano-epoksydowego) w pojedynczym miejscu jest bardzo wysoka szansa na pojawienie się odkształcenia laminatu co może doprowadzić do uszkodzenia PCB (ścieżki na płytce tworzone są z miedzi która ma inną rozszerzalność termiczną niż włókno szklane / materiał szkalno-epoksydowy).





Metoda łączenia układów scalonych PoP

Dla problemu przyjęliśmy iż ogrzewanie następuje za pomocą stacji lutowniczej Yihua 878d z temperaturą powietrza opuszczającego dyszę równą  $480^{\circ}\text{C}$ . Ponieważ kulka cyny ma średnicę zaledwie 0.5mm (przez co znikomą masę) oraz jest połączona bezpośrednio z układem scalonym możemy przyjąć iż kulki cyny będą miały temperaturę równą temperaturze pamięci.

### Wyznaczone wartości:

Dane dotyczące charakterystyk wafla krzemowego:

<https://www.americanelements.com/silicon-wafer-7440-21-3>

Długość: 12mm

Szerokość: 12mm

nH: 24

nW: 24

Temperatura początkowa:  $25^{\circ}\text{C}$

Temperatura strumienia powietrza:  $480^{\circ}\text{C}$

Pojemność cieplna:  $703.3824 \text{ J}/(\text{kg } ^{\circ}\text{C})$

Współczynnik przewodzenia:  $149 \text{ W}/(\text{m } ^{\circ}\text{C})$

Gęstość:  $2330 \text{ kg}/\text{m}^3$

Współczynnik wymiany ciepła:  $300 \text{ W}/(\text{m}^2\text{K})$

Czas symulacji: 120 s

Krok czasowy: 2s

**Otrzymane wyniki:**

Iteration [0] after 5.0 s.	Min: 29.438	Max: 61.591
Iteration [1] after 10.0 s.	Min: 37.821	Max: 79.448
Iteration [2] after 15.0 s.	Min: 48.328	Max: 93.115
Iteration [3] after 20.0 s.	Min: 59.688	Max: 105.076
Iteration [4] after 25.0 s.	Min: 71.249	Max: 116.158
Iteration [5] after 30.0 s.	Min: 82.713	Max: 126.696
Iteration [6] after 35.0 s.	Min: 93.952	Max: 136.835
Iteration [7] after 40.0 s.	Min: 104.915	Max: 146.643
Iteration [8] after 45.0 s.	Min: 115.586	Max: 156.154
Iteration [9] after 50.0 s.	Min: 125.960	Max: 165.385
Iteration [10] after 55.0 s.	Min: 136.043	Max: 174.350
Iteration [11] after 60.0 s.	Min: 145.840	Max: 183.058
Iteration [12] after 65.0 s.	Min: 155.358	Max: 191.518
Iteration [13] after 70.0 s.	Min: 164.606	Max: 199.736
Iteration [14] after 75.0 s.	Min: 173.590	Max: 207.720
Iteration [15] after 80.0 s.	Min: 182.319	Max: 215.476
Iteration [16] after 85.0 s.	Min: 190.799	Max: 223.011
Iteration [17] after 90.0 s.	Min: 199.037	Max: 230.332
Iteration [18] after 95.0 s.	Min: 207.041	Max: 237.444
Iteration [19] after 100.0 s.	Min: 214.817	Max: 244.354
Iteration [20] after 105.0 s.	Min: 222.371	Max: 251.067
Iteration [21] after 110.0 s.	Min: 229.710	Max: 257.588
Iteration [22] after 115.0 s.	Min: 236.840	Max: 263.924

Możemy zatem wnioskować iż dla tradycyjnej cyny wystarczy ogrzewać układ scalony przez 80 sekund, natomiast dla cyny bezołowiowej należy to robić przez co najmniej 110 sekund.

## Wnioski i uwagi

---

Z uwagi na ilość obliczeń program został napisany w języku C++ by zminimalizować czas wyznaczania wartości. Przy tak dużym projekcie różnice w trybach kompilacji były bardzo widoczne, w trybie Debug obliczanie symulacji problemu rzeczywistego zajmowało średnio 21 sekund, natomiast w trybie Release 600 milisekund (testy dokonane na kompilatorze MSVC19). W metodzie eliminacji Gaussa zastosowano warunek sprawdzający czy współczynnik przez który ma być mnożona część macierzy jest różny od zera, pozwoliło to na redukcję czasów obliczeń nawet 13-krotnie! Przy użyciu biblioteki OpenMP można skorzystać z równoległej wersji algorytmu rozwiązywania układu równań (najbardziej czasochłonna część obliczeń) i przez to skrócić czas obliczeń jeszcze bardziej. Dla siatki o rozmiarze powyżej 200x200 elementów program potrafi zużyć ponad 28GB pamięci RAM, dla sprawnego obliczania siatek o dużych rozmiarach należy zmienić reprezentację macierzy gdyż dużo miejsca jest marnowane na komórki wypełnione zerami.