Dajana Kopera
Inżynieria Obliczeniowa, rok 3, gr. 1
Numer indeksu: 286108

SPRAWOZDANIE

Metoda elementów skończonych

Celem laboratorium oraz zajęć projektowych było zapoznanie z algorytmem MES, by samemu stworzyć program opierający się właśnie na tym algorytmie.

Do napisania programu skorzystałam z języka programowania Java oraz ze środowiska JetBrains IntelliJ IDEA. W tym celu stworzyłam 14 klas, które składają się na program. Parametry procesu zostały przeze mnie uwzględnione w jednej z klas, nie wpisywałam pliku do programu, korzystałam z pól klasy Dane.

Na początku utworzyłam klasy:

- **Dane** (zawierająca parametry zadania),
- **Node** (odpowiedzialną za węzeł),
- **Element** (odpowiedzialną za element),
- **Universal Element** (odpowiedzialną za element uniwersalny),
- **Grid** (tworząca siatkę składającą się z elementów oraz węzłów),
- **Jakobian2D** (macierz przekształcenia).

Są to klasy stworzone dzięki wzorowaniu się na pliku Jakobian2D_excel. Po sprawdzeniu wyników z danymi w arkuszach pliku przystąpiłam do pisania dalszej części programu - stworzyłam lokalną macierz [C] oraz [H] ((**MatrixCLocal**, **MatrixHLocal**) macierz H została podzielona na dwie części - taką, która nie korzysta z warunku brzegowego konwekcji oraz taką, która ten warunek brzegowy uwzględnia. Ponadto w klasie lokalnej macierzy H zostały przeze mnie uzupełnione wartości lokalnego wektora {P}).

Dotychczas wszystkie stworzone obiekty klas były przeze mnie wypisywane bezpośrednio w klasie głównej **Main**, jednak w celu większej przejrzystości dla według mnie trudniejszej części zadania stworzyłam osobną klasę - **Proces**, w której zapisywałam agregację do globalnych macierzy ((**MatrixCGlobal**, **MatrixHGlobal**) oraz w późniejszym etapie algorytm Gaussa, który został przeze mnie dodany w postaci nowej klasy - **Gauss**).

Stworzyłam również macierz **VectorPGlobal**, gdzie tworzyłam globalny wektor {P} wraz z metodą potrzebną do symulacji. Cały proces symulacji został zawarty na końcu konstruktora klasy Proces, gdzie za pomocą metody z Grid uzupełniam temperatury w węzłach, stosuję algorytm Gaussa, a następnie wypisuję wyniki.
W celu sprawdzenia poprawności otrzymanych wyników skonfrontowałam je z danymi w **Test cases transient solution**.

Poniżej znajdują się rozszerzone opisy poszczególnych klas oraz zrzuty ekranu dotyczące programu.

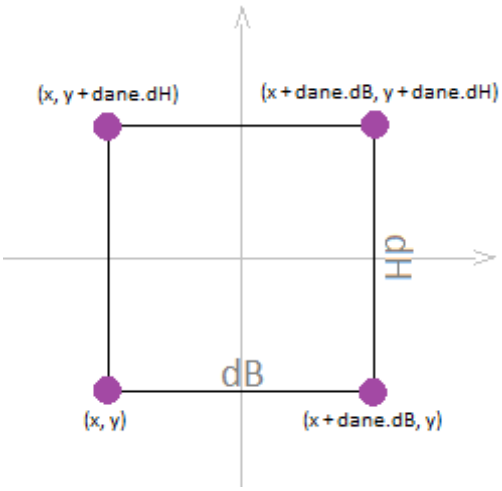
KLASA DANE (Testcase)
Klasa utworzona by korzystać z parametrów.
Zawiera:
- parametry procesu, którym zostają nadane wartości w konstruktorze klasy,
- konstruktor (w konstruktorze znajdują się wzory na ilość wszystkich węzłów, ilość wszystkich elementów oraz odległości pomiędzy węzłami po wysokości (dH) oraz po szerokości (dB)),
- funkcję printData, która wypisuje parametry.

KLASA NODE
Zawiera informacje dotyczące węzłów:
- jego współrzędne,
- status - który jest wykorzystywany jako „flaga" przy sprawdzaniu, czy w danym węźle znajduje się warunek brzegowy czy też nie,
- ID węzła, które jest ułatwieniem przy przejściu pomiędzy kolejnymi węzłami (również przy przejściu z układu lokalnego na globalny - zamiana odpowiednich współrzędnych dla umieszczenia zmiennych w macierzach [C] oraz [H]),
- nodeTemp - zmienna informująca o temperaturze w danych węźle.

Oraz konstruktor, który jako argumenty przyjmuje współrzędne oraz parametry (dane), przypisuje współrzędne, na których operuje, następnie znajduje się warunek if, który sprawdza, czy współrzędne są brzegowe (krawędzie całego badanego obszaru), jeśli tak, status = 1 czyli jest prawdziwe, co oznacza istnienie warunku brzegowego, w przeciwnym przypadku ustawia status = 0 równoznaczne z brakiem warunku brzegowego w węźle.

KLASA ELEMENT
Na początku klasy zostaje utworzona tablica węzłów (która będzie wykorzystywana do tworzenia elementu, każdy z elementów będzie zawierał po 4 węzły) oraz współrzędne.

Zawiera konstruktor, którego argumenty to współrzędne oraz parametry (dane), przypisuje sobie obecne współrzędne, następnie zostaje stworzona tablica czteroelementowa (jedna dla jednego elementu).
Kolejno dla poszczególnych węzłów zostają ustawione współrzędne.



(Współrzędne w elemencie)

KLASA UNIWERSAL_ELEMENT (arkusz Jakobian2D)

Reprezentuje uniwersalny element, który w układzie lokalnym jest wykorzystywany do liczenia Jakobianu (zawiera 3 macierze - macierz funkcji kształtu, funkcje kształtu/ksi, funkcje kształtu/eta)

Na początku zostają zdefiniowane dwie tablice zawierające kolejne wartości ksi oraz eta, następnie zostają zdefiniowane dN, dN/dKsi oraz dN/dEta.

ksi	eta
-0,57735	-0,57735
0,57735	-0,57735
0,57735	0,57735
-0,57735	0,57735

Konstruktor klasy zawiera:

- wypełnianie macierzy o wymiarach 4x4 (cztery funkcje kształtu, ponieważ element ma dokładnie cztery węzły). Macierze zostały wypełnione odpowiadającymi w komórkach wzorami na funkcje kształtu.

Np. dla N1: (macierz 1)

$N1=0.25(1-\varepsilon)(1-\mu)$

		dN			
ksi	eta	N1	N2	N3	N4
-0,57735	-0,57735	0,622008	0,1666667	0,044658	0,166667
0,57735	-0,57735	0,166667	0,6220085	0,166667	0,044658
0,57735	0,57735	0,044658	0,1666667	0,622008	0,166667
-0,57735	0,57735	0,166667	0,0446582	0,166667	0,622008

(macierz 2)

$N1/de=-0.25(1-n)$

			dN/dKsi		
	Eta	-0,57735	-0,57735	0,57735	0,57735
	dN/dksi		pkt całk		
		1	2	3	4
numer	1	-0,39434	-0,39434	-0,10566	-0,10566
funkcji	2	0,394338	0,394338	0,105662	0,105662
lształtu	3	0,105662	0,105662	0,394338	0,394338
	4	-0,10566	-0,10566	-0,39434	-0,39434

(macierz 3)

$N1/dn=-0.25(1-e)$

			dN/dEta		
	Ksi	-0,57735	0,57735	0,57735	-0,57735
	dN/eta		pkt całk		
		1	2	3	4
numer	1	-0,39434	-0,10566	-0,10566	-0,39434
funkcji	2	-0,10566	-0,39434	-0,39434	-0,10566
lształtu	3	0,105662	0,394338	0,394338	0,105662
	4	0,394338	0,105662	0,105662	0,394338

Oprócz tego klasa zawiera dwie funkcje wypisujące - osobno dla tablic ksi/eta oraz wszystkich macierzy (macierze wypisane zostały kolumna po kolumnie ze względu na przejrzystość).

```
--  --  --  --  --  --  --  --   MACIERZ 3  --  --  --  --  --  --  --
                    (funkcje kształtu po ETA)
                              dNdETA
Wartości funkcji kształtu dNdETA dla KSI 1:
dN1/dEta = -0,39434
dN2/dEta = -0,10566
dN3/dEta = 0,10566
dN4/dEta = 0,39434

Wartości funkcji kształtu dNdETA dla KSI 2:
dN1/dEta = -0,10566
dN2/dEta = -0,39434
dN3/dEta = 0,39434
dN4/dEta = 0,10566

Wartości funkcji kształtu dNdETA dla KSI 3:
dN1/dEta = -0,10566
dN2/dEta = -0,39434
dN3/dEta = 0,39434
dN4/dEta = 0,10566

Wartości funkcji kształtu dNdETA dla KSI 4:
dN1/dEta = -0,39434
dN2/dEta = -0,10566
dN3/dEta = 0,10566
dN4/dEta = 0,39434
```

(Przykładowe wypisanie w programie)

KLASA JAKOBIAN2D (arkusz Jakobian2D & Matrix H)

Klasa zawiera:

- obiekt klasy UniwersalElement, ponieważ dN/dKsi oraz dN/dEta są konieczne do jego rozwiązania. Jakobian jest macierzą przekształcenia, która umożliwia przejście z układu lokalnego na globalny (szukane są wartości funkcji kształtu po współrzędnych globalnych, możemy je obliczyć dzięki przekształceniom oraz zależnościom odpowiednio ksi i eta od x, y).
- macierz jakobianu (zawierająca pochodne współrzędnych globalnych po ksi oraz eta),
- wektor wyznaczników dla poszczególnych punktów całkowania,
- macierz, która jest ostateczną wersją jakobianu z arkusza Jakobian2D (połączenie powyższych).

pkt całk	1	2	3	4
J_1_1	1,25E-02	1,25E-02	0,0125	0,0125
J_1_2	0,00E+00	0	0	0
J_2_1	0,00E+00	0	0	0
J_2_2	0,0125	0,0125	0,0125	0,0125

pkt całk	1	2	3	4
Det J	1,56E-04	0,000156	0,000156	0,0001563

pkt całk	1	2	3	4
J-1_1_1	8,00E+01	80	80	80
j-1_1_2	0	0	0	0
J-1_2_1	0	0	0	0
J-1_2_2	80	80	80	80

```
----    ----    ----    ----      JAKOBIAN 2D     ----    ----
0,01250         0,01250         0,01250         0,01250
0,00000         0,00000         0,00000         0,00000
0,00000         0,00000         0,00000         0,00000
0,01250         0,01250         0,01250         0,01250
--  --  --  --  --  --  --  --  --  --  --  --  --  --  --



--  --  --  --  --  --  --  --   Wyznacznik J   --  --  --
detJ1: 0,00016
detJ2: 0,00016
detJ3: 0,00016
detJ4: 0,00016



--  --  --  --  --  --  --  --   Jakobian/det J --  --  --
80,00000        80,00000        80,00000        80,00000
0,00000         0,00000         0,00000         0,00000
0,00000         0,00000         0,00000         0,00000
80,00000        80,00000        80,00000        80,00000
```

Ponadto zdefiniowane zostały dwie macierze, które prezentują wartości funkcji kształtu po współrzędnych globalnych - na rzecz składowych wzoru na macierz [H] (arkusz MatrixH)

Macierze dN/dx oraz dN/dy w klasie Jakobian2d

pc	dN1/dx	dN2/dx	dN3/dx	dN4/dx
1	-31,547	31,54701	8,452995	-8,45299
2	-31,547	31,54701	8,452995	-8,45299
3	-8,453	8,452995	31,54701	-31,547
4	-8,453	8,452995	31,54701	-31,547

pc	dN1/dy	dN2/dy	dN3/dy	dN4/dy
1	-31,547	-8,453	8,453	31,547
2	-8,45299	-31,547	31,547	8,453
3	-8,45299	-31,547	31,547	8,453
4	-31,547	-8,453	8,453	31,547

```
--  --  --  --  --  --    PRZEJŚCIE NA GLOBALNY   --  --  --
--  --  --  --  --  --  --  --    dN/dx   --  --  --  --  --
dN1/dx          dN2/dx          dN3/dx          dN4/dx
-31,54701       31,54701        8,45299         -8,45299
-31,54701       31,54701        8,45299         -8,45299
-8,45299        8,45299         31,54701        -31,54701
-8,45299        8,45299         31,54701        -31,54701
--  --  --  --  --  --  --  --  --  --  --  --  --  --  --



--  --  --  --  --  --  --  --    dN/dy   --  --  --  --  --
dN1/dy          dN2/dy          dN3/dy          dN4/dy
-31,54701       -8,45299        8,45299         31,54701
-8,45299        -31,54701       31,54701        8,45299
-8,45299        -31,54701       31,54701        8,45299
-31,54701       -8,45299        8,45299         31,54701
```

(Zrzut ekranu z wypisania w programie)

Wszystkie macierze uzupełniane są w konstruktorze, który jako argument przyjmuje obiekt klasy Element.

KLASA GRID

Klasa ta definiuje siatkę MES, w konstruktorze przyjmuje parametry (dane) oraz tablicę elementów, gdzie każdy z elementów zawiera tablicę swoich węzłów. Następnie stworzona została zmienna, która była odpowiedzialna za numerację podczas przechodzenia pomiędzy węzłami lub elementami (licznik).

Konstruktor zawiera również dwie pętle, które umożliwiają umiejscowienie odpowiednio w siatce elementów jak i węzłów (na poniższej grafice zostały przedstawione numeracje poszczególnych elementów (zielony) jak i węzłów (fiolet) w układzie współrzędnych).

Ponadto zawiera dwie funkcje - jedna z nich ustawia ID poszczególnych węzłów (by przy przejściu z układu na globalny odpowiednio wysłać wartości do macierzy [C] oraz [H]) oraz funkcję ustawiającą temperaturę w węzłach, która zostaje użyta później w klasie Proces.

KLASA MATRIX H LOCAL (arkusze Matrix H & Matrix H_BC_2d)

Lokalne macierze H (arkusz Matrix H)

Klasa ta zawiera lokalne macierze H utworzone osobno dla każdego punktu całkowania dla współrzędnych x, y osobno np. dN1dNT_dx_1pc/dN2dNT_dy_2pc (oraz kolejne 8 macierzy, gdzie te pierwsze zostają przemnożone odpowiednio przez wyznacznik det[J]).

Te macierze dodatkowo przemnażane są przez współczynnik k = 25 (kXY_detJ_1pc), na końcu są agregowane manualnie według wzorów zawartych w arkuszach do lokalnej macierzy [H] matrixH.

Macierz H			
16,66667	-4,1667	-8,33333	-4,16667
-4,16667	16,6667	-4,16667	-8,33333
-8,33333	-4,1667	16,66667	-4,16667
-4,16667	-8,3333	-4,16667	16,66667

```
==  ==  ==  ==  ==  ====     MACIERZ H LOKALNA  ====
Z excel: arkusz MatrixH
16,667      -4,167      -8,333      -4,167
-4,167      16,667      -4,167      -8,333
-8,333      -4,167      16,667      -4,167
-4,167      -8,333      -4,167      16,667
--  --  --  --  --  --  --  --  --  --  --  --  --  --
```

(Wypisanie części macierzy [H], która nie uwzględnia warunków brzegowych)

Lokalne macierze HBC (arkusz Matrix HBC_2d)

Ta część macierzy [H] uwzględnia warunek brzegowy zawarty w macierzy [H] (część warunku konwekcji zależnego od nieznanej temperatury).

Zostały utworzone trzy macierze (NHBC, NHBC_1pc, NHBC_2pc), które dla każdej sumy są zerowane oraz zapisywane. Dodatkowo na rzecz tworzenia sum dla każdej powierzchni utworzyłam tablicę dwuelementową, w której umieściłam długość boku oraz wartość detJ = długość boku/2 (tablica dBAndDetJ).

Następnie na podstawie wzorów wartości zostały podstawione do macierzy [H] uwzględniając statusy dla konkretnych powierzchni (matrixHCB).

Matrix H	1	2	3	4
1	5	1,25	0	1,25
2	1,25	2,5	0	0
3	0	0	0	0
4	1,25	0	0	2,5

```
--  --  --  --  --  --  --  --   Macierz H_BC   --  --  --  -
Z excel: arkusz Matrix H_BC (uwzględniająca warunek brzegowy)
5,000       1,250       0,000       1,250
1,250       2,500       0,000       0,000
0,000       0,000       0,000       0,000
1,250       0,000       0,000       2,500
```

Oprócz tego klasa MatrixHLocal zawiera **lokalny wektor {P}** (właściwie cztery wektory osobno dla każdego punktu całkowania, które korzystają z wartości funkcji kształtu macierzy NHCB - np. vektorPlokalnie1pc), który przyjmuje swoje globalne wartości w później stworzonej klasie VectorPGlobal. Na końcu zostają sprawdzone poszczególne węzły, czy występują w nich warunki brzegowe, a gdy występują zostaje użyta pętla for, by odpowiednio wymnożyć wektory funkcji kształtu przez współczynnik alfa oraz temperaturę otoczenia.

KLASA MATRIX C LOCAL (arkusz Matrix C)

Klasa ta posiada cztery macierze dla każdego punktu całkowania (które są obliczane na podstawie ciepła właściwego, gęstości, wyznacznika Jakobianu i wartości funkcji kształtu) oraz macierz lokalną [C], na którą składają się wcześniej wspomniane macierze.

	0,622	0,166667	0,044658	0,166667
0,622008	330,07	88,44183	23,69792	88,44183
0,166667	88,442	23,69792	6,349838	23,69792
0,044658	23,698	6,349838	1,701434	6,349838
0,166667	88,442	23,69792	6,349838	23,69792

	0,16666667	0,622008	0,166667	0,044658
0,166667	23,6979167	88,44183	23,69792	6,349838
0,622008	88,441829	330,0694	88,44183	23,69792
0,166667	23,6979167	88,44183	23,69792	6,349838
0,044658	6,34983763	23,69792	6,349838	1,701434

	0,0447	0,166667	0,622008	0,166667
0,044658	1,7014	6,349838	23,69792	6,349838
0,166667	6,3498	23,69792	88,44183	23,69792
0,622008	23,698	88,44183	330,0694	88,44183
0,166667	6,3498	23,69792	88,44183	23,69792

	0,16666667	0,044658	0,166667	0,622008
0,166667	23,6979167	6,349838	23,69792	88,44183
0,044658	6,34983763	1,701434	6,349838	23,69792
0,166667	23,6979167	6,349838	23,69792	88,44183
0,622008	88,441829	23,69792	88,44183	330,0694

Matrix C	379,17	189,5833	94,79167	189,5833
	189,58	379,1667	189,5833	94,79167
	94,792	189,5833	379,1667	189,5833
	189,58	94,79167	189,5833	379,1667

```
Macierze składające się na macierz lokalną [C]:
Dla 1 punktu całkowania:
330,069     88,442      23,698      88,442
88,442      23,698      6,350       23,698
23,698      6,350       1,701       6,350
88,442      23,698      6,350       23,698
--  --  --  --  --  --  --  --  --  --  --  --  -
Dla 2 punktu całkowania:
23,698      88,442      23,698      6,350
88,442      330,069     88,442      23,698
23,698      88,442      23,698      6,350
6,350       23,698      6,350       1,701
--  --  --  --  --  --  --  --  --  --  --  --  -
Dla 3 punktu całkowania:
1,701       6,350       23,698      6,350
6,350       23,698      88,442      23,698
23,698      88,442      330,069     88,442
6,350       23,698      88,442      23,698
--  --  --  --  --  --  --  --  --  --  --  --  -
Dla 4 punktu całkowania:
23,698      6,350       23,698      88,442
6,350       1,701       6,350       23,698
23,698      6,350       23,698      88,442
88,442      23,698      88,442      330,069
--  --  --  --  --  --  --  --  --  --  --  --  -



=   ==  ==  ==  ==  ==  ==  MACIERZ C LOKALNIE  =
379,167     189,583     94,792      189,583
189,583     379,167     189,583     94,792
94,792      189,583     379,167     189,583
189,583     94,792      189,583     379,167
--  --  --  --  --  --  --  --  --  --  --  --  -
```

KLASA MATRIX C GLOBAL (Testcase 2d)

Klasa ta zawiera globalną macierz [C], która powstała na skutek agregacji wszystkich lokalnych macierzy [C]. W konstruktorze zostaje utworzona macierz o wielkości **ilość węzłów x ilość węzłów**, która wypełniona jest zerami (by na etapie jej tworzenia nie była zaśmiecona). Oprócz tego, klasa zawiera funkcję, która dokonuje agregacji oraz funkcję wypisującą. Obie te funkcje zostają użyte w klasie Proces.

Martix [C]

_____ Interation 0 _____
_____ Matrix [C] _____
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
0 337.037 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

=	==	==	==	==	==	==	MACIERZ C GLOBALNIE	==	==	==	==	==	==	=				
674,074	337,037	0,000	0,000	337,037	168,519	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000				
337,037	1348,148	337,037	0,000	168,519	674,074	168,519	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000				
0,000	337,037	1348,148	337,037	0,000	168,519	674,074	168,519	0,000	0,000	0,000	0,000	0,000	0,000	0,000				
0,000	0,000	337,037	674,074	0,000	0,000	168,519	337,037	0,000	0,000	0,000	0,000	0,000	0,000	0,000				
337,037	168,519	0,000	0,000	1348,148	674,074	0,000	0,000	337,037	168,519	0,000	0,000	0,000	0,000	0,000				
168,519	674,074	168,519	0,000	674,074	2696,296	674,074	0,000	168,519	674,074	168,519	0,000	0,000	0,000	0,000				
0,000	168,519	674,074	168,519	0,000	674,074	2696,296	674,074	0,000	168,519	674,074	168,519	0,000	0,000	0,000				
0,000	0,000	168,519	337,037	0,000	0,000	674,074	1348,148	0,000	0,000	168,519	337,037	0,000	0,000	0,000				
0,000	0,000	0,000	0,000	337,037	168,519	0,000	0,000	1348,148	674,074	0,000	0,000	337,037	168,519	0,000				
0,000	0,000	0,000	0,000	168,519	674,074	168,519	0,000	674,074	2696,296	674,074	0,000	168,519	674,074	168,519				
0,000	0,000	0,000	0,000	0,000	168,519	674,074	168,519	0,000	674,074	2696,296	674,074	0,000	168,519	674,074				
0,000	0,000	0,000	0,000	0,000	0,000	168,519	337,037	0,000	0,000	674,074	1348,148	0,000	0,000	168,519				
0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	337,037	168,519	0,000	0,000	674,074	337,037	0,000				
0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	168,519	674,074	168,519	0,000	337,037	1348,148	337,037				
0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	168,519	674,074	168,519	0,000	337,037	1348,148				
0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	168,519	337,037	0,000	0,000	337,037				

KLASA MATRIX H GLOBAL (Testcase 2d)

Klasa ta zawiera globalną macierz [H], która powstała na skutek agregacji wszystkich lokalnych macierzy [H] oraz [H_BC]. W konstruktorze zostaje utworzona macierz o wielkości **ilość węzłów x ilość węzłów**, która wypełniona jest zerami (by na etapie jej tworzenia nie była zaśmiecona). Oprócz tego, klasa zawiera funkcję, która dokonuje agregacji samej macierzy [H], macierzy lokalnych [H_BC], które uwzględniają warunki brzegowe oraz agregację [H] = [H] + [C]/dT, gdzie dT to krok czasowy oraz funkcję wypisującą. Wszystkie te funkcje zostają wywołane w klasie Proces.

Martix [H]

_____ Interation 0 _____
_____ Matrix [H] _____
16.6667 -4.16667 0 0 -4.16667 -8.33333 0 0 0 0 0 0 0 0 0 0
-4.16667 33.3333 -4.16667 0 -8.33333 -8.33333 -8.33333 0 0 0 0 0 0 0 0 0
0 -4.16667 33.3333 -4.16667 0 -8.33333 -8.33333 -8.33333 0 0 0 0 0 0 0 0
0 0 -4.16667 16.6667 0 0 -8.33333 -4.16667 0 0 0 0 0 0 0 0
-4.16667 -8.33333 0 0 33.3333 -8.33333 0 0 -4.16667 -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 -8.33333 -8.33333 0 0 0 0
0 0 -8.33333 -4.16667 0 0 -8.33333 33.3333 0 0 -8.33333 -4.16667 0 0 0 0
0 0 0 0 -4.16667 -8.33333 0 0 33.3333 -8.33333 0 0 -4.16667 -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 -8.33333 -8.33333
0 0 0 0 0 0 -8.33333 -4.16667 0 0 -8.33333 33.3333 0 0 -8.33333 -4.16667
0 0 0 0 0 0 0 0 -4.16667 -8.33333 0 0 16.6667 -4.16667 0 0
0 0 0 0 0 0 0 0 -8.33333 -8.33333 -8.33333 0 -4.16667 33.3333 -4.16667 0
0 0 0 0 0 0 0 0 0 -8.33333 -8.33333 -8.33333 0 -4.16667 33.3333 -4.16667
0 0 0 0 0 0 0 0 0 0 0 -8.33333 -4.16667 0 0 -4.16667 16.6667

```
==  ==  ==  ==  ==  ==  ==  ==  ==  MACIERZ [H] GLOBALNIE   ==  ==  ==  ==  ==  ==  ==  ==  ==
16,667  -4,167  0,000  0,000  -4,167  -8,333  0,000  0,000  0,000  0,000  0,000  0,000  0,000  0,000  0,000  0,000
-4,167  33,333  -4,167  0,000  -8,333  -8,333  -8,333  0,000  0,000  0,000  0,000  0,000  0,000  0,000  0,000  0,000
0,000  -4,167  33,333  -4,167  0,000  -8,333  -8,333  -8,333  0,000  0,000  0,000  0,000  0,000  0,000  0,000  0,000
0,000  0,000  -4,167  16,667  0,000  0,000  -8,333  -4,167  0,000  0,000  0,000  0,000  0,000  0,000  0,000  0,000
-4,167  -8,333  0,000  0,000  33,333  -8,333  0,000  0,000  -4,167  -8,333  0,000  0,000  0,000  0,000  0,000  0,000
-8,333  -8,333  -8,333  0,000  -8,333  66,667  -8,333  0,000  -8,333  -8,333  -8,333  0,000  0,000  0,000  0,000  0,000
0,000  -8,333  -8,333  -8,333  0,000  -8,333  66,667  -8,333  0,000  -8,333  -8,333  -8,333  0,000  0,000  0,000  0,000
0,000  0,000  -8,333  -4,167  0,000  0,000  -8,333  33,333  0,000  0,000  -8,333  -4,167  0,000  0,000  0,000  0,000
0,000  0,000  0,000  0,000  -4,167  -8,333  0,000  0,000  33,333  -8,333  0,000  0,000  -4,167  -8,333  0,000  0,000
0,000  0,000  0,000  0,000  -8,333  -8,333  -8,333  0,000  -8,333  66,667  -8,333  0,000  -8,333  -8,333  -8,333  0,000
0,000  0,000  0,000  0,000  0,000  -8,333  -8,333  -8,333  0,000  -8,333  66,667  -8,333  0,000  -8,333  -8,333  -8,333
0,000  0,000  0,000  0,000  0,000  0,000  -8,333  -4,167  0,000  0,000  -8,333  33,333  0,000  0,000  -8,333  -4,167
0,000  0,000  0,000  0,000  0,000  0,000  0,000  0,000  -4,167  -8,333  0,000  0,000  16,667  -4,167  0,000  0,000
0,000  0,000  0,000  0,000  0,000  0,000  0,000  0,000  -8,333  -8,333  -8,333  0,000  -4,167  33,333  -4,167  0,000
0,000  0,000  0,000  0,000  0,000  0,000  0,000  0,000  0,000  -8,333  -8,333  -8,333  0,000  -4,167  33,333  -4,167
0,000  0,000  0,000  0,000  0,000  0,000  0,000  0,000  0,000  0,000  -8,333  -4,167  0,000  0,000  -4,167  16,667
```

(Wydruk programu po agregacji macierzy [H])

Martix **[H]** = [H]+[C]/dT

```
_____ Interation 0 _____
_____ Matrix ([H]+[C]/dT)_____
36.8148 4.24074 0 0 4.24074 -4.96296 0 0 0 0 0 0 0 0 0 0
4.24074 66.963 4.24074 0 -4.96296 5.14815 -4.96296 0 0 0 0 0 0 0 0 0
0 4.24074 66.963 4.24074 0 -4.96296 5.14815 -4.96296 0 0 0 0 0 0 0 0
0 0 4.24074 36.8148 0 0 -4.96296 4.24074 0 0 0 0 0 0 0 0
4.24074 -4.96296 0 0 66.963 5.14815 0 0 4.24074 -4.96296 0 0 0 0 0 0
-4.96296 5.14815 -4.96296 0 5.14815 120.593 5.14815 0 -4.96296 5.14815 -4.96296 0 0 0 0 0
0 -4.96296 5.14815 -4.96296 0 5.14815 120.593 5.14815 0 -4.96296 5.14815 -4.96296 0 0 0 0
0 0 -4.96296 4.24074 0 0 5.14815 66.963 0 0 -4.96296 4.24074 0 0 0 0
0 0 0 0 4.24074 -4.96296 0 0 66.963 5.14815 0 0 4.24074 -4.96296 0 0
0 0 0 0 -4.96296 5.14815 -4.96296 0 5.14815 120.593 5.14815 0 -4.96296 5.14815 -4.96296 0
0 0 0 0 0 -4.96296 5.14815 -4.96296 0 5.14815 120.593 5.14815 0 -4.96296 5.14815 -4.96296
0 0 0 0 0 0 -4.96296 4.24074 0 0 5.14815 66.963 0 0 -4.96296 4.24074
0 0 0 0 0 0 0 0 4.24074 -4.96296 0 0 36.8148 4.24074 0 0
0 0 0 0 0 0 0 0 -4.96296 5.14815 -4.96296 0 4.24074 66.963 4.24074 0
0 0 0 0 0 0 0 0 0 -4.96296 5.14815 -4.96296 0 4.24074 66.963 4.24074
0 0 0 0 0 0 0 0 0 0 -4.96296 4.24074 0 0 4.24074 36.8148
```

```
[H] = [H] + [C]/dT
==  ==  ==  ==  ==  ==  ==  ==  ==  MACIERZ [H] GLOBALNIE   ==  ==  ==  ==  ==  ==  ==  ==  ==
36,815  4,241  0,000  0,000  4,241  -4,963  0,000  0,000  0,000  0,000  0,000  0,000  0,000  0,000  0,000  0,000
4,241  66,963  4,241  0,000  -4,963  5,148  -4,963  0,000  0,000  0,000  0,000  0,000  0,000  0,000  0,000  0,000
0,000  4,241  66,963  4,241  0,000  -4,963  5,148  -4,963  0,000  0,000  0,000  0,000  0,000  0,000  0,000  0,000
0,000  0,000  4,241  36,815  0,000  0,000  -4,963  4,241  0,000  0,000  0,000  0,000  0,000  0,000  0,000  0,000
4,241  -4,963  0,000  0,000  66,963  5,148  0,000  0,000  4,241  -4,963  0,000  0,000  0,000  0,000  0,000  0,000
-4,963  5,148  -4,963  0,000  5,148  120,593  5,148  0,000  -4,963  5,148  -4,963  0,000  0,000  0,000  0,000  0,000
0,000  -4,963  5,148  -4,963  0,000  5,148  120,593  5,148  0,000  -4,963  5,148  -4,963  0,000  0,000  0,000  0,000
0,000  0,000  -4,963  4,241  0,000  0,000  5,148  66,963  0,000  0,000  -4,963  4,241  0,000  0,000  0,000  0,000
0,000  0,000  0,000  0,000  4,241  -4,963  0,000  0,000  66,963  5,148  0,000  0,000  4,241  -4,963  0,000  0,000
0,000  0,000  0,000  0,000  -4,963  5,148  -4,963  0,000  5,148  120,593  5,148  0,000  -4,963  5,148  -4,963  0,000
0,000  0,000  0,000  0,000  0,000  -4,963  5,148  -4,963  0,000  5,148  120,593  5,148  0,000  -4,963  5,148  -4,963
0,000  0,000  0,000  0,000  0,000  0,000  -4,963  4,241  0,000  0,000  5,148  66,963  0,000  0,000  -4,963  4,241
0,000  0,000  0,000  0,000  0,000  0,000  0,000  0,000  4,241  -4,963  0,000  0,000  36,815  4,241  0,000  0,000
0,000  0,000  0,000  0,000  0,000  0,000  0,000  0,000  -4,963  5,148  -4,963  0,000  4,241  66,963  4,241  0,000
0,000  0,000  0,000  0,000  0,000  0,000  0,000  0,000  0,000  -4,963  5,148  -4,963  0,000  4,241  66,963  4,241
0,000  0,000  0,000  0,000  0,000  0,000  0,000  0,000  0,000  0,000  -4,963  4,241  0,000  0,000  4,241  36,815
```

(Wydruk programu po agregacji funkcją [H] = [H] + [C]/dT)

KLASA VEKTOR P GLOBAL (Testcase 2d)

Ta klasa oprócz konstruktora zawiera funkcje agregującą wektor {P} lokalny do globalnego oraz funkcję uwzględniającą temperaturę początkową oraz krok czasowy: {P} = ({P} + {[C]/dT} * T0). Dodatkowo klasa zawiera metodę operującą na innych temperaturach niż początkowa (za pomocą metody Gaussa).

```
_____ Vector ({{P}+{[C]/dT}*{T0})_____
15033.3 18066.7 18066.7 15033.3 18066.7 12133.3 12133.3 18066.7 18066.7 12133.3 12133.3 18066.7 15033.3 18066.7 18066.7 15033.350 110.04 365.82
```

```
==  ==  ==  ==  ==  ==  ==  ==  ==  WEKTOR {P} GLOBALNIE   ==  ==  ==  ==  ==  ==  ==  ==  ==
15033,3  18066,7  18066,7  15033,3  18066,7  12133,3  12133,3  18066,7  18066,7  12133,3  12133,3  18066,7  15033,3  18066,7  18066,7  15033,3
```

Podobnie jak poprzednio wypisywanie jak i agregacja wektora {P} globalnego znajduje się w klasie Proces.

```
_____ Interation 1 _____
_____H Matrix ([H]+[C]/dT)_____
36.815 4.2407 0 0 4.2407 -4.963 0 0 0 0 0 0 0 0 0 0
4.2407 66.963 4.2407 0 -4.963 5.1481 -4.963 0 0 0 0 0 0 0 0 0
0 4.2407 66.963 4.2407 0 -4.963 5.1481 -4.963 0 0 0 0 0 0 0 0
0 0 4.2407 36.815 0 0 -4.963 4.2407 0 0 0 0 0 0 0 0
4.2407 -4.963 0 0 66.963 5.1481 0 0 4.2407 -4.963 0 0 0 0 0 0
-4.963 5.1481 -4.963 0 5.1481 120.59 5.1481 0 -4.963 5.1481 -4.963 0 0 0 0 0
0 -4.963 5.1481 -4.963 0 5.1481 120.59 5.1481 0 -4.963 5.1481 -4.963 0 0 0 0
0 0 -4.963 4.2407 0 0 5.1481 66.963 0 0 -4.963 4.2407 0 0 0 0
0 0 0 0 4.2407 -4.963 0 0 66.963 5.1481 0 0 4.2407 -4.963 0 0
0 0 0 0 -4.963 5.1481 -4.963 0 5.1481 120.59 5.1481 0 -4.963 5.1481 -4.963 0
0 0 0 0 0 -4.963 5.1481 -4.963 0 5.1481 120.59 5.1481 0 -4.963 5.1481 -4.963
0 0 0 0 0 0 -4.963 4.2407 0 0 5.1481 66.963 0 0 -4.963 4.2407
0 0 0 0 0 0 0 0 4.2407 -4.963 0 0 36.815 4.2407 0 0
0 0 0 0 0 0 0 0 -4.963 5.1481 -4.963 0 4.2407 66.963 4.2407 0
0 0 0 0 0 0 0 0 0 -4.963 5.1481 -4.963 0 4.2407 66.963 4.2407
0 0 0 0 0 0 0 0 0 0 -4.963 4.2407 0 0 4.2407 36.815
_____P_Vector ([{P}+{[C]/dT}*{T0})_____
20660 25552 25552 20660 25552 18897 18897 25552 25552 18897 18897 25552 20660 25552 25552 20660
```

== == == == == == == == == MACIERZ [H] GLOBALNIE == == == == == == == == ==

36,815	4,241	0,000	0,000	4,241	-4,963	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
4,241	66,963	4,241	0,000	-4,963	5,148	-4,963	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
0,000	4,241	66,963	4,241	0,000	-4,963	5,148	-4,963	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
0,000	0,000	4,241	36,815	0,000	0,000	-4,963	4,241	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
4,241	-4,963	0,000	0,000	66,963	5,148	0,000	0,000	4,241	-4,963	0,000	0,000	0,000	0,000	0,000	0,000
-4,963	5,148	-4,963	0,000	5,148	120,593	5,148	0,000	-4,963	5,148	-4,963	0,000	0,000	0,000	0,000	0,000
0,000	-4,963	5,148	-4,963	0,000	5,148	120,593	5,148	0,000	-4,963	5,148	-4,963	0,000	0,000	0,000	0,000
0,000	0,000	-4,963	4,241	0,000	0,000	5,148	66,963	0,000	0,000	-4,963	4,241	0,000	0,000	0,000	0,000
0,000	0,000	0,000	0,000	4,241	-4,963	0,000	0,000	66,963	5,148	0,000	0,000	4,241	-4,963	0,000	0,000
0,000	0,000	0,000	0,000	-4,963	5,148	-4,963	0,000	5,148	120,593	5,148	0,000	-4,963	5,148	-4,963	0,000
0,000	0,000	0,000	0,000	0,000	-4,963	5,148	-4,963	0,000	5,148	120,593	5,148	0,000	-4,963	5,148	-4,963
0,000	0,000	0,000	0,000	0,000	0,000	-4,963	4,241	0,000	0,000	5,148	66,963	0,000	0,000	-4,963	4,241
0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	4,241	-4,963	0,000	0,000	36,815	4,241	0,000	0,000
0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	-4,963	5,148	-4,963	0,000	4,241	66,963	4,241	0,000
0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	-4,963	5,148	-4,963	0,000	4,241	66,963	4,241
0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	-4,963	4,241	0,000	0,000	4,241	66,963
0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	-4,963	4,241	0,000	0,000	4,241	

-- --

== == == == == == == == == WEKTOR {P} GLOBALNIE == == == == == == == == ==

20659,7 25552,2 25552,2 20659,7 25552,2 18897,4 18897,4 25552,2 25552,2 18897,4 18897,4 25552,2 20659,7 25552,2 25552,2 20659,7

-- --

(Wydruk programu dla jednej iteracji)

Metoda Gaussa została przeze mnie wykorzystana w nowej klasie Gauss, a ciało klasy zostało pobrane i odpowiednio zmodyfikowane na rzecz programu.

KLASA PROCES

Jest to klasa agregująca macierze [C] oraz [H] do globalnych, wypisuje wektor {P} globalny oraz dokonuje na nim symulacji - zmiana temperatur Tn za pomocą metody Gaussa (korzysta z metody zdefiniowanej w klasie Grid - ustawiającej wartości temperatur w węzłach). Klasa sama w sobie nie zawiera żadnych metod własnych, korzysta z wcześniej utworzonych przez inne klasy.

Max and min temperature in each step

Time[s]	MinTemp[s]	MaxTemp[s]
50	110.038	365.815
100	168.837	502.592
150	242.801	587.373
200	318.615	649.387
250	391.256	700.068
300	459.037	744.063
350	521.586	783.383
400	579.034	818.992
450	631.689	851.431
500	679.908	881.058

MAKSYMALNE I MINIMALNE TEMPERATURY W KOLEJNYCH KROKACH CZASOWYCH

Time[s]	MinTemp[s]	MaxTemp[s]
50	110,038	365,815
100	168,837	502,592
150	242,801	587,373
200	318,615	649,387
250	391,256	700,068
300	459,037	744,063
350	521,586	783,383
400	579,034	818,992
450	631,689	851,431
500	679,908	881,058