Jakub Kopeć, IS, GL03  
  
1. Wstęp teoretyczny:

Celem działania programu jest stworzenie symulacji nagrzewania się kwadratowego/prostokątnego (zależnie od ustawień w pliku dane.txt) dwuwymiarowego fragmentu metalu.

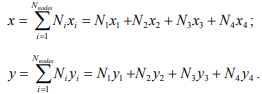
Żeby stworzyć symulację należy rozpatrywany fragment pokryć siatką punktów, z których każdy będzie znajdował się na narożach mniejszego kwadratu/prostokąta, nazywającego się elementem skończonym. W stworzonym programie sprowadza się to do podziału dwuwymiarowej płaszczyzny kwadratowej/prostokątnej na mniejsze kwadratowe/prostokątne elementy skończone, których wielkość zależy od ilości punktów w metodologii MES, nazywanych węzłami.

Jako, że stworzone elementy niekoniecznie muszą być foremne następnie należy je doprowadzić do układu uniwersalnego (kwadratów o wielkości 2x2) przy użyciu Jakobianów oraz funkcji kształtu.

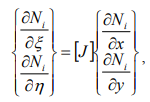
*Wzory funkcji kształtów:*

**

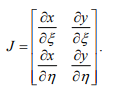
*Transformacja układu współrzędnych:*

**

*Związek funkcji kształtu względem ksi i eta oraz pochodnymi funkcji kształtu po x i y:*

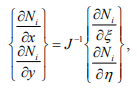
**

*[J] czyli macierz Jakobiego:*

**

Wyznacznik tej macierzy jest Jakobianem transformacji układu współrzędnych.

*Wyznaczenie pochodnych funkcji kształtu x oraz y przy użyciu równania:*

**

*Całkowanie funkcji kształtu z wykorzystaniem metody przybliżonej kwadratury Gauss’a:*

**

Gdzie w to wagi poszczególnych punktów całkowania, a ksi i eta to ich współrzędne Gauss’a.

Wagi oraz współrzędne punktów zależne są od przyjętego schematu całkowania.

(W programie zaimplementowany został schemat dla dwóch,trzech i czterech punktów całkowania).

Ostatnim krokiem w symulacji jest wyznaczenie temperatury elementów skończonych zależnie od czasu. Jako ,że proces zmienia się w czasie jest on niestacjonarny.

*Równanie Fourier’a dla procesu niestacjonarnego:*



Jako ,że w określonej chwili czasu pochodne temperatury mogą być traktowane tak jak funkcje współrzędnych x,y,z, rozwiązanie powyższego równania jest analogicznie jak dla procesu stacjonarnego, gdzie wyrażenie w ostatnim nawiasie jest przyjmowane jako parametr Q.

*Otrzymana forma macierzowa rówania:*

**

Po uwzględnieniu funkcji kształtu i zmian w wektorze temperatur oraz macierzy obciążeń powyższe równanie sprowadza się do jego wersji wykorzystanej w programie:

([H]+[C]/Δt)\*{t}-[C]/Δt\*{t0}+{P}=0

Gdzie:

H- fragment warunku brzegowego, jest to globalna macierz stworzona przez agregację sum macierzy H oraz Hbc dla poszczególnych elementów

C-globalna macierz obciążeń, również powstała przez agregację jak w przypadku macierzy H

t0- wektor temperatur początkowych w danym kroku czasowym

P-wektor obciążeń z warunkami brzegowymi

t-wektor temperatur końcowych w danym kroku czasowym (poszukiwany)

Δt-długość kroku czasowego

*Wzór do wyliczenia temperatur pod koniec kroku czasowego*:

[H^]^(-1)\*[H^]\*{t}-[P^]=0

*Po przekształceniu:*

{t}=[H^]^(-1)\*[P^]

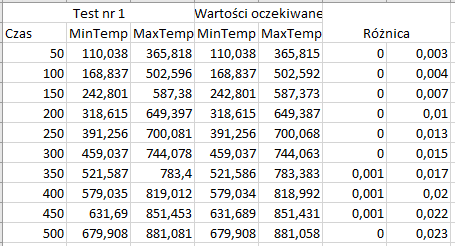
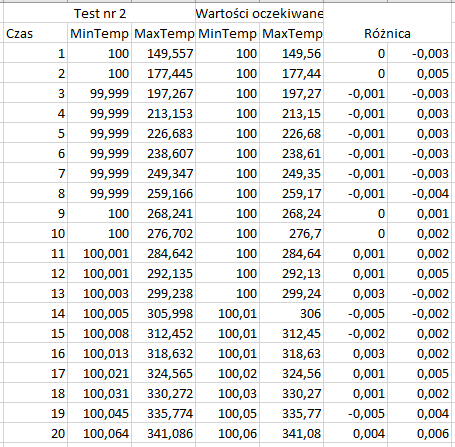
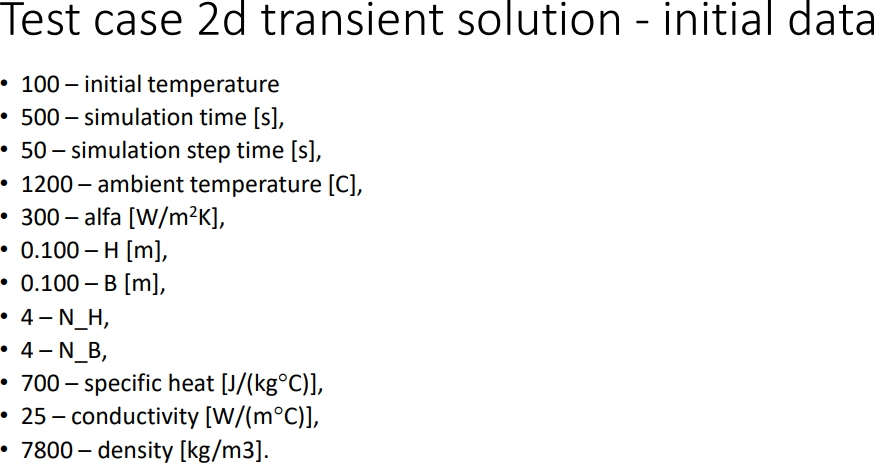
Gdzie:

[H^]=([H]+[C]/Δt)

[P^]=[C]/Δt\*{t0}+{P}

Ten wzór jest wykorzystywany podczas każdego kroku czasowego aż do zakończenia symulacji celem wyznaczenia temperatur w danym przedziale czasu (zależnym od przyjętej długości kroku czasowego).

2. Charakterystyka projektu:  
Działanie programu rozpoczyna obiekt klasy GlobalData, który odczytuje dane z pliku tekstowego niezbędne do utworzenia siatki. Obiekt ten pobiera kolejno: wysokość oraz szerokość siatki (zakładamy, że siatka ma kształt prostokąta), ilość węzłów na jej bokach, współczynnik przewodzenia ciepła, schemat całkowania wykorzystywany przy całkowaniu metodą kwadratur Gaussa, gęstość materiału, ciepło właściwe materiału, temperaturę początkową węzłów siatki, temperaturę otoczenia, współczynnik konwekcji, długość symulacji oraz długość kroku czasowego.  
Na podstawie danych z pliku tworzone są obiekty Node, które wchodzą w skład siatki, by następnie przypisać im odpowiednie identyfikatory, współrzędne, temperaturę początkową oraz zmienną określającą występowanie warunku brzegowego na krawędziach każdego z elementów. Potem węzły o odpowiednich identyfikatorach przypisywane są do elementów. Węzły elementu znajdują się na przecięciach krawędzi elementów. Po stworzeniu struktury siatki wywoływana jest funkcja simulation, która oblicza skok czasowy jednej iteracji symulacji. Każda iteracja pętli symulacji zawiera pętlę wywołań funkcji elemSolve, która oblicza oraz zwraca macierze niezbędne do znalezienia wektora rozwiązań układu równań dla jednego z elementów siatki. Funkcja ta niezależnie od schematu całkowania liczy lokalne macierze H, C i wektor P, które później za pomocą algorytmu agregacji współtworzą globalne macierze H, C, P widoczne w macierzowej formie równania we wstępie teoretycznym. Bazując na operacjach na macierzach oraz na algorytmie odwracania macierzy Gaussa-Jordana, rozwiązywany jest układ równań, z którego wybierana jest minimalna oraz maksymalna wartość temperatury i porównywana z przewidywanymi wartościami. Zbiór rozwiązań jest następnie wykorzystywany w nadpisaniu poprzednich temperatur w węzłach i poszukiwany jest nowy wektor rozwiązań.

3. Porównanie wyników programu z przewidywanymi wartościami:  
Z racji braku różnic pomiędzy wynikami programu dla różnych schematów całkowania, zdecydowałem się nie umieszczać danych dla każdego ze schematów.  
  
  
  
  
  
  
Parametry wejściowe programu dla testu nr. 1:  


Parametry wejściowe programu dla testu nr. 2:  
