POLITECHNIKA WARSZAWSKA
Wydział Mechatroniki

Praca przejściowa

Przemysław Kuc

Opracowanie narzędzia programistycznego do zapisu danych z symulacji MES w plikach programu Matlab (mat).

Opiekun pracy:
dr inż. Szymon Cygan

Jednostka dyplomująca:
Instytut Metrologii i Inżynierii Biomedycznej

Warszawa, 2014

Spis treści

1. Cel i zakres pracy

Celem poniższej pracy przejściowej jest wykonanie narzędzia programistycznego służącego do zapisu danych pochodzących z symulacji MES, w plikach w formacie programu MATLAB.

Dane pochodzące z symulacji MES pochodzą z programu ABAQUS.

Narzędzie programistyczne zostało opracowane w postaci skryptu w języku programowania PYTHON. Skrypt odczytuje pliki wejściowe (.inp) i wyjściowe (.odb) programu ABAQUS i generuje na ich podstawie plik w formacie programu MATLAB (.mat).

Zakres pracy:

1. Napisanie skryptu odczytującego pliki wejściowe (.inp) i wyjściowe (.odb) programu ABAQUS
2. Napisanie skryptu zapisującego odczytanie dane z symulacji typu MES w formacie programu MATLAB.

2. Metoda elementów skończonych (MES)

Metoda elementów skończonych (MES) jest bardzo popularną komputerową metodą wspomagającą badania naukowe i analizę zagadnień inżynierskich.

Polega ona na tym że każdą wielkość fizyczną, którą możemy opisać za pomocą funkcji ciągłej aproksymuje się modelem dyskretnym.

Model dyskretny dzielony jest na wiele funkcji ciągłych w pewnej skończonej liczbie podobszarów. Takie funkcje nazywa się elementami. Poszczególne funkcję ciągłe definiowane są przez pewną funkcję pierwotną, w skończonej liczbie punktów wewnątrz rozważanego obszaru. Takie punkty nazywa się węzłami. Następnie budowany jest układ równań różniczkowych w poszczególnych węzłach, przebiegających po elementach.

Odpowiedni podział badanego zjawiska na elementy pozwala na zastąpienia problemu analitycznego, który zapisywany jest za pomocą równań różniczkowych, na problem algebraiczny. Pozwala to na znaczne uproszczenie obliczeń, zwłaszcza w przypadku zastosowań inżynierskich. [1]

Metoda elementów skończonych nie jest metodą bardzo dokładną. Poprzez uproszczenie badanego modelu tracimy dokładność badanych zjawisk. Podstawowymi źródłami błędów w metodzie elementów skończonych są :

- błąd podczas modelowania – model dokładnie nie odzwierciedla modelu rzeczywistego
- błąd wartości przyjętych współczynników – błąd związany z niedokładnym określeniem parametrów np. materiałowych modelu
- błąd numeryczny - błąd dyskretyzacji badanego modelu. Metoda aproksymacji modelu może wprowadzać błędy w stosunku do badanego modelu
- błąd zaokrągleń [2]

Popularność metody elementów skończonych spowodowała powstanie wielu programów specjalistycznych, umożliwiających rozwiązywanie zagadnień MES. Najbardziej rozbudowanymi programami dostępnymi na rynku są:

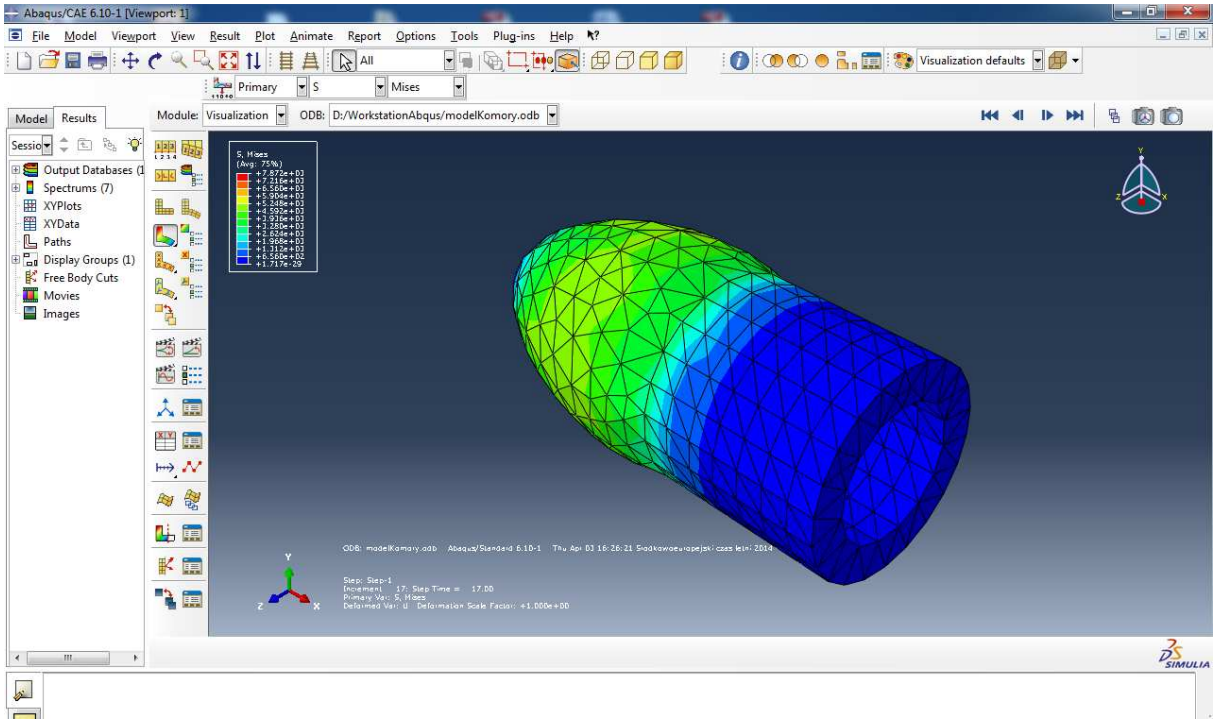
- ANSYS
- ABAQUS
- NASTRAN
- LS DYNA

2.1 Program ABAQUS

Dane pochodzące z symulacji MES, wykorzystywane w tej pracy zostały otrzymane w programie ABAQUS.

Program ABAQUS firmy Dassault Systemes jest jednym z najlepszych programów wykorzystujących metodę elementów skończonych. Pozwala on na dokonywanie bardzo złożonych obliczeń z zakresu mechaniki ciała stałego i przepływów ciepła. Powszechnie wykorzystywany jest w celu oceny wytrzymałościowej elementów maszyn i konstrukcji przemysłowych. Znalazł swoje zastosowanie między innymi w przemyśle samochodowym, maszynowym, wydobywczym i hutniczym. Program ABQAUS umożliwia symulacją złożonych procesów wytwarzania nowych produktów i materiałów.

Na tle innych programów, ABAQUS wyróżnia się przede wszystkim modularną budową. Główny nacisk twórców został położony na koncepcji bibliotek. Zapewnia to użytkownikom możliwość dowolnego tworzenie i łączenia nowych elementów. Program posiada też bogatą bibliotekę umożliwiającą implementowanie różnych materiałów do modeli użytkowników. Biblioteka ta pozwala na modelowanie różnych elementów metalowych, a także posiada modele przeznaczone do analizy np. gum i kompozytów. [3] [4]



Rys. 1 Okno programu ABAQUS. Badany model wykorzystywany w poniższej pracy

3. Opis działania narzędzia programistycznego

Narzędzie programistyczne zostało opracowane w postaci skryptu w języku programowania PYTHON. Skrypt odczytuje pliki wejściowe (.inp) i wyjściowe (.odb) programu ABAQUS i generuje na ich podstawie plik w formacie programu MATLAB (.mat).

3.1 Założenia

W ramach wstępnych założeń plik wynikowy .mat powstały w wyniku działania skryptu powinien zawierać następujące zmienne:

- **Amplitude** – macierz wartości przebiegu amplitudy poszczególnych wymuszenia zadeklarowanych w strukturze Loads. Dane powinny być wyskalowane w przedziale 0-1 [m][n] m – liczba kroków, n – kolejne amplitudy
- **Coordinates** – macierz zawierająca współrzędne węzłów sieci w kolejnych krokach symulacji programu ABAQUS. Macierz wynikowa o wymiarach [L][M][N], gdzie L – liczba węzłów sieci, M – współrzędne kartezjańskie położenia węzłów sieci, N – liczba kroków symulacji
- **CreationTime** – wektor opisujący datę konwersji danych [rok m-c dzień godzina minuta sekunda]
- **CycleTime** – czas trwania symulacji
- **Displacements** – wartości przemieszczeń w węzłach dla kolejnych kroków symulacji
- **Loads** – struktura opisująca wymuszenia zastosowane w symulacji.
- **Strains** – wartości odkształceń w węzłach dla kolejnych kroków symulacji
- **Stresses** – wartości naprężeń w węzłach dla kolejnych kroków symulacji w Pascalach
- **Units** – struktura opisująca jednostki stosowane w symulacji
- **Elements** – tablica zawierająca numery węzłów należących do kolejnych elementów modelu
- **filename** – łańcuch znaków zawierający nazwę pliku wejściowego (.inp) i wyjściowego (.odb)
- **surface_out** – lista węzłów zewnętrznej płaszczyzny modelu
- **surface_in** – lista węzłów wewnętrznej płaszczyzny modelu [5]

3.2 Opis działania skryptu

3.2.1 Moduły dodatkowe – NumPy i SciPy

Skrypt korzysta z dwóch dodatkowych, zewnętrznych modułów NumPy i SciPy.

Moduł NumPy jest modułem dodatkowym języka PYTHON umożliwiającym efektywniejszą pracę z tablicami. W wykonanym skrypcie ułatwia on budowę wielowymiarowych tablic i struktur zgodnych z założeniami projektu.

Moduł SciPy jest modułem dodatkowym języka PYTHON zawierającym pakiet algorytmów i matematycznych narzędzi. W wykonanym skrypcie ułatwia on generację pliku w formacie .mat.

Uprzednie zainstalowanie obu modułów na urządzeniu testującym jest wymagane, do poprawnego działania oprogramowania. Wymagane jest także ustawienie poprawnej ścieżki systemowej programu ABQUS, by program mógł korzystać z dodatkowych modułów.

3.2.2 Opis działania poszczególnych funkcji skryptu

I. countNumberOfElements(partName, stepName)

Dane wejściowe:

partName - nazwa części modelu

stepName - nazwa kroku analizy

Dane wyjściowe:

numberOfElements - liczba elementów

Opis działania funkcji:

Funkcja zlicza liczbę elementów w badanym modelu.

II. countNumberOfNodes(partName, stepName, setName)

Dane wejściowe:

partName - nazwa części modelu

stepName - nazwa kroku analizy

setName - nazwa set'u analizy

Dane wyjściowe:

numberOfElements - liczba węzłów

Opis działania:

Funkcja zlicza liczbę węzłów w badanym modelu.

III. initDispOrCoordTable (stepName, partName, setName)

Dane wejściowe:

partName - nazwa części modelu

stepName - nazwa kroku analizy

setName - nazwa set'u analizy

Dane wyjściowe:

Tablica trójwymiarowa [nodeNumbers][3][1]

Opis działania:

Funkcja generuje wypełniona zerami tablice trójwymiarowa o wymiarach [nodeNumbers – liczba węzłów][3- liczba współrzędnych np. przemieszczeń 'U'][1]

IV. initStressOrStrainTable(stepName, partName, setName)

Dane wejściowe:

partName - nazwa części modelu

stepName - nazwa kroku analizy

setName - nazwa set'u analizy

Dane wyjściowe:

Tablica trójwymiarową [nodeNumbers][6][1]

Opis działania:

Funkcja generuje wypełnioną zerami tablice trójwymiarową o wymiarach [nodeNumbers – liczba węzłów][6-liczba współrzędnych np. odkształceń 'LE'][1]

V. createElementTable(inpFileName, stepName, partName)

Dane wejściowe:

inpFileName - nazwa pliku wejściowego .inp

stepName - nazwa kroku analizy

partName - nazwa części modelu

Dane wyjściowe:

Tablica dwuwymiarowa [elementNumbers][4]

Opis działania:

Funkcja generuje tablice węzłów wchodzących w skład każdego z elementów

VI. createAmplitudeTable(inpFileName, ampName)

Dane wejściowe:

inpFileName - nazwa pliku wejściowego .inp

ampName - nazwa wykorzystywanego zakresu amplitud w symulacji

Dane wyjściowe:

Tablica jednowymiarowa [liczbaAmplitud]

Opis działania:

Funkcja generuje jednowymiarowa tablice z kolejnymi wartościami amplitud wykorzystywanymi w symulacji. Wyniki są przeskalowane z przedziału <0,1>. Wartość 1 oznacza maksymalna amplitudę wykorzystywana w symulacji.

VII. stressAndStrain(stepName, fieldOut, partName, setName)

Dane wejściowe:

stepName - nazwa kroku analizy

fieldOut - nazwa danych wyjściowych, których wyniki chcemy uzyskać

partName - nazwa części modelu

setName - nazwa set'u analizy

Dane wyjściowe:

Tablica trójwymiarowa [nodeNumbers][6][numberOfFrames]

Opis działania:

Funkcja generuje tablice trójwymiarową o wymiarach [nodeNumbers - liczba węzłów][6-liczba współrzędnych np. odkształceń 'LE'][numberOfFrames - liczba frame'ow w symulacji. Funkcja generuje tablice dwuwymiarowa dla każdego frame'u, a następnie łączy tę tablicę tworząc wyjściowa tablice trójwymiarową.

VIII. dispAndCoord(stepName, fieldOut, partName, setName)

Dane wejściowe:

stepName - nazwa kroku analizy

fieldOut - nazwa danych wyjściowych, których wyniki chcemy uzyskać

partName - nazwa części modelu

setName - nazwa set'u analizy

Dane wyjściowe:

Tablica trójwymiarowa [nodeNumbers][6][numberOfFrames]

Opis działania:

Funkcja generuje tablice trójwymiarową o wymiarach [nodeNumbers – liczba węzłów][3-liczba współrzędnych np. przemieszczeń 'U'] [numberOfFrames - liczba frame'ow w symulacji]

Funkcja generuje tablice dwuwymiarowa dla każdego frame'u, a następnie 'skleja' tę tablicę tworząc wyjściowa tablice trójwymiarową.

4. Podsumowanie, wnioski i możliwości rozbudowy projektu

Przedstawione w powyższej pracy narzędzie programistyczne spełnia wymagania postawione w założone w projekcie.

Skrypt napisany w języku programowania PYTHON poprawnie odczytuje pliki wejściowe (.inp) i wyjściowe (.odb) programu ABAQUS i generuje na ich podstawie plik w formacie programu MATLAB (.mat).

Otrzymany plik wyjściowy w formacie programu MATLAB (.mat) zawiera większość zmiennych zawartych w założeniach w podpunkcie 3.1. Nie zawiera jedynie zmiennych surface_in i surface_out, ponieważ powierzchnie te nie zostały zadeklarowane w modelu w programie ABAQUS.

Dalszy rozwój projektu powinien przebiegać drogą optymalizacji kodu skryptu. Czas działania skryptu jest elementem, od którego należałoby rozpocząć dalszy prace nad projektem.

5. Bibliografia

1) Rusiński E., Czmochowski J., Smolnicki T., Zaawansowana metoda elementów skończonych w konstrukcjach nośnych, Wrocław, Oficyna Wydawnicza Politechniki Wrocławskiej, 2000 r.
2) http://pl.wikipedia.org/wiki/Metoda_element%C3%B3w_sko%C5%84czonych
3) Skrzat Andrzej, Modelowanie liniowych i nieliniowych problemów mechaniki ciała stałego i przepływów ciepła w programie ABAQUS, Rzeszów, Oficyna Wydawnicza Politechniki Rzeszowskiej, 2010 r.
4) http://www.cyfronet.krakow.pl/start/13251,artykul,abaqus.html
5) Cygan Szymon, Żmigrodzki Jakub, Opis formatu plików wymiany danych pomiędzy Abaqusem i Matlabem, Warszawa, 2014r.
6) Bressert Eli, SciPy and NumPy , O'Reilly, 2013 r.
7) Abaqus Scripting Reference Manual, SIMULIA, 2010
8) Abaqus Scripting User's Manual, SIMULIA, 2010
9) Abaqus Analysis User's Manual Volume II, SIMULIA, 2010
10) Amar Khennane, Introduction to Finite Element Analysis Using MATLAB® and Abaqus, CRC Press, 2013 r.