

Лабораторная работа.

Добавление в узлы массы, зависящей от площади элемента.

Выполнила студент группы 33604

Григорьева Полина

Проверил: Ле-Захаров С.А.

The image shows two handwritten signatures in blue ink on a light-colored background. The top signature is a stylized, cursive mark, possibly initials. The bottom signature is more legible, appearing to be 'Ле-Захаров'.

Целью данной работы является добавление в узлы элементов массы, зависящей от площади, приходящейся на один узел. На рисунке представлена модель с узлами.

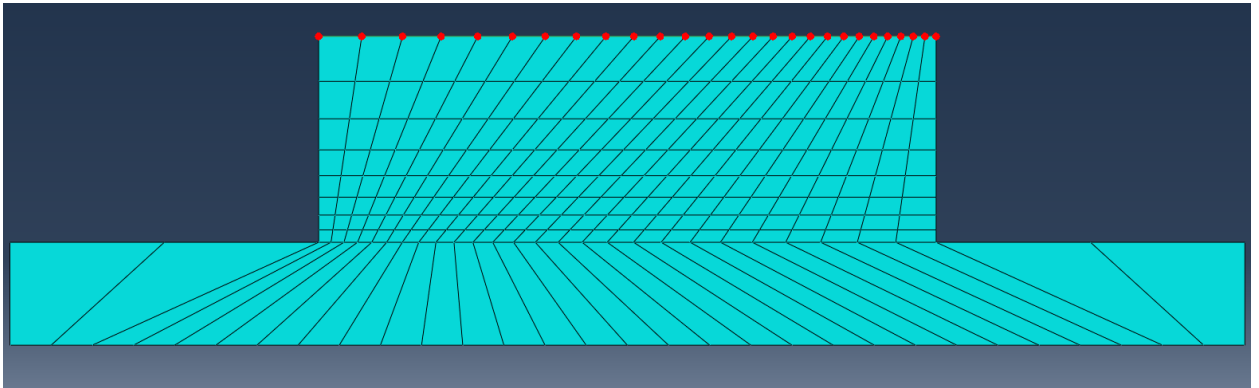


Рис.1 Модель с узлами

Так как количество узлов велико, то для удобной работы сделана запись в файл значений площадей каждого из них. Числовое значение площади равно числовому значению массы элемента, поэтому для информативности будем считать, что в файле записаны массы узлов. Ниже представлен код программы, написанной на языке Python.

```

1 from abaqus import *
2 import visualization
3 import regionToolset
4
5 myMdb = openMdb('Dom.cae')
6 a = myMdb.models['Model-1'].parts['Part-1'].sets['Set-3'].nodes
7 c = myMdb.models['Model-1'].parts['Part-1'].elements
8
9 set_nodes = []
10
11 for node in c:
12     if (str(node.type) == 'B21'):
13         set_nodes.append(node)
14
15 element_square = []
16 x1=0
17 x2=0
18
19 for n in set_nodes:
20     for a0 in a:
21         if ((n.connectivity[0]+1) == a0.label):
22             x1=a0.coordinates[0]
23             if ((n.connectivity[1]+1)== a0.label):
24                 x2=a0.coordinates[0]
25             ds = abs(x1-x2)
26             element_square.append(ds)
27
28 list=[]
29 ds = 0
30 mass_ =[]
31
32 f = open('file.txt','w')
33 for p in a:
34     i=0
35     ds = 0
36     for n in set_nodes:
37         if (p.label== n.connectivity[0]+1) or (p.label== n.connectivity[1]+1):
38             ds = ds+element_square[i]/2
39             i=i+1
40         if (p.coordinates[0]==30) or (p.coordinates[0]==-30):
41             ds = ds*2
42
43         f.write("mass "+ str(ds))
44         f.write("\n")
45         mass_.append(ds)
46 f.close()
47
48 nodeLabelList=[]
49 i=0
50 for a0 in a:
51     m = mass_[i]
52     name_of_inertia = "mass number"+str(i+1)
53     nodeLabelList.append(a0.label)
54     meshNodes=a.sequenceFromLabels(nodeLabelList)
55     s = regionToolset.Region(nodes = meshNodes)
56     myMdb.models['Model-1'].parts['Part-1'].engineeringFeatures.PointMassInertia(name = name_of_inertia.region = s,mass=m)
57     nodeLabelList=[]
58     meshNodes=[]
59     i=i+1
60
61
62

```

Рис.2. Текст программы

```
mass 1.08278465271
mass 1.11286067963
mass 1.17468357086
mass 1.23994159698
mass 1.30882453918
mass 1.38153457642
mass 1.45828342438
mass 1.53929615021
mass 1.62481021881
mass 1.71507358551
mass 1.81035137177
mass 1.91092300415
mass 2.01708221436
mass 2.12913799286
mass 2.24741888046
mass 2.37227094173
mass 2.5040589571
mass 2.64316833019
mass 2.79000556469
mass 2.94500017166
mass 3.10860538483
mass 3.28129935265
mass 3.46358728409
mass 3.65600156784
mass 3.85910511017
mass 4.07349205017
mass 4.18358230591
```

Рис. 3. Значения масс элементов

Видим, что значение масс нелинейно возрастает, что согласуется с распределением узлов.

Вывод:

В данной лабораторной работе была решена задача, которую нельзя было бы решить без использования PDE модуля. Расширенные возможности и большая гибкость являются плюсами данного модуля, что было продемонстрировано при решении задачи.