

## МЕТОДЫ МОРФИНГА РАСЧЕТНЫХ СЕТОК ДЛЯ ПОДГОТОВКИ ИНЖЕНЕРНЫХ ЗАДАЧ ПРОЧНОСТИ В ЛОГОС-ПРЕПОСТ

*Тарасова Светлана Сергеевна (SSTarasova@vniief.ru), Кузнецов Михаил Геннадьевич,  
Логинов Дмитрий Викторович, Пятериков Константин Алексеевич,  
Соловьёв Александр Николаевич*

ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской обл.

ЛОГОС-ПРЕПОСТ является системой подготовки начальных данных для моделирующих кодов, входящих в состав пакета программ ЛОГОС.

В настоящее время одним из наиболее трудоемких этапов подготовки математической модели в ЛОГОС-ПРЕПОСТ является этап сеткогенерации. В современных инженерных пакетах реализован ряд автоматических, полуавтоматических и ручных алгоритмов генерации расчетных сеток.

Зачастую использование автоматических методов сеткогенерации становится невозможным при подготовке сеточной модели для сложных геометрий, а использование ручных методов влечет за собой значительное увеличение трудозатрат. Ускорить данный процесс подготовки сеточной модели позволяют методы морфинга (трансформации сетки без изменения ее топологии) узлов расчетной сетки.

Также данные методы позволяют во многом автоматизировать подготовку оптимизационных задач, в которых меняется геометрическая форма исследуемой конструкции или деталей, – т. к. подобные задачи часто требуют генерации и редактирования большого количества различных вариантов расчетной сетки для нахождения оптимального решения.

В препроцессоре ЛОГОС-ПРЕПОСТ в части средств подготовки к решению задач прочности на текущий момент реализованы три базовых алгоритма морфинга – перенос сетки, свободная деформация и морфинг узлов сетки.

В докладе приведено описание реализации данных алгоритмов, общая идея технологии морфинга, базовый математический алгоритм, рассматривается интерфейс пользователя, также приведены примеры использования операций морфинга узлов на практических и тестовых задачах.

**Ключевые слова:** морфинг, деформация, расчетная сетка.

## MORPHING TOOLS IN LOGOS PREPOST PREPROCESSOR FOR SOLVING STRENGTH PROBLEMS

*Tarasova Svetlana Sergeevna (SSTarasova@vniief.ru), Kuznetsov Mikhail Gennadievich,  
Loginov Dmitriy Viktorovich, Pyaterikov Konstantin Alekseevich,  
Soloviov Alexandr Nikolaevich*

FSUE «RFNC-VNIIEF», Sarov Nizhny Novgorod region

LOGOS-PREPOST is a system for preparing initial data for modeling codes included in the LOGOS software package.

Currently, one of the most time-consuming stages of preparing a mathematical model in LOGOS-PREPOST is the stage of mesh generation. A number of automatic, semi-automatic and manual algorithms for generating computational grids are implemented in modern engineering packages.

Often, the use of automatic mesh generation methods becomes impossible when preparing a mesh model for complex geometries, and the use of manual methods entails a significant increase in labor

costs. The methods of morphing the nodes of the computational grid allow to speed up this process of preparing the mesh model.

Also, these methods make it possible to largely automate the preparation of optimization problems in which the geometric shape of the structure or parts under study changes, since such tasks often require the generation and editing of a large number of different variants of the computational grid to find the optimal solution.

In the LOGOS-PREPOST preprocessor, in terms of means of preparation for solving strength problems, three basic morphing algorithms are currently implemented – transferring the mesh to the face, free deformation and morphing of mesh nodes.

The report contains a description of the implementation of these algorithms, the general idea of the morphing technology, the basic mathematical algorithm, the user interface is considered, and examples of using the operations of morphing nodes on practical and test problems are given.

**Key words:** morphing, deformation, computational mesh.

## Математический алгоритм морфинга сетки

Морфинг – это трансформация сетки без изменения ее топологии. Методы морфинга сетки представляют собой алгоритмы адаптации узлов существующей расчетной сетки в соответствии с искажением геометрической формы.

Адаптацию сеточной модели с помощью методов морфинга можно представить следующим образом: имеем исходную замкнутую геометрическую поверхность  $G$ , объемную сетку  $M$ , описывающую данную геометрию, и вариацию исходной геометрии  $G \rightarrow G'$ , полученную изменением  $G$  с сохранением исходной топологии. Методы морфинга «искажают» исходную расчетную сетку  $M$  так, что обновленная версия  $M'$  согласуется с поверхностью  $G'$ . При этом, аналогично геометрическому представлению, сеточная модель сохраняет свою топологию.

Входными данными для алгоритма морфинга, является три набора узлов:  $S$  – поверхностные узлы объемной сетки,  $V$  – внутренние узлы исходной сетки  $M$ , и обновленные координаты поверхностных узлов  $S'$ .  $S$  и  $S'$  располагаются на геометриях  $G$  и  $G'$  соответственно. Цель алгоритма – вычислить обновленные координаты внутренних узлов  $V'$  расчетной сетки  $M'$  наиболее корректно.

В общем случае можно рассматривать задачу морфинга как интерполяционную задачу, решением которой будет функция  $d(x)$  (1), которая точно интерполирует заданное граничное смещение  $d(s_i) = (s'_i - s_i)$  и (2) гладко интерполирует смещение внутренних узлов сетки.

$$d(x) = \sum_{j=1}^m w_j \varphi_j(x) + \sum_{k=1}^4 q_k \pi_k(x), \quad (1)$$

где  $\{\pi_1, \pi_2, \pi_3, \pi_4\} = \{x, y, z, 1\}$  является пространственным базисом трехмерного полинома, взвешенного коэффициентами  $q_k$ ,  $\varphi_j(x)$  – ядро функции, задающее вид интерполанта.

$$\iiint_{IR^3} \left\| \frac{\partial^k d}{\partial x^k} \right\|^2 + \left\| \frac{\partial^k d}{\partial x^{k-1} \partial y} \right\|^2 + \dots + \left\| \frac{\partial^k d}{\partial z^k} \right\|^2 dx dy dz. \quad (2)$$

Метод RBF (радиально базисные функции) является наиболее популярным для решения данного типа задач интерполяции.

Наиболее часто используемые ядра приведены в таблице.

Виды RBF ядер

Gaussian	$\varphi(r) = e^{-(\epsilon r)^2}$
Multiquadratic	$\varphi(r) = \sqrt{1 + (\epsilon r)^2}$
Inverse multiquadratic	$\varphi(r) = 1 / \sqrt{1 + (\epsilon r)^2}$
Polyharmonic spline	$\varphi(r) = \begin{cases} r^{2k-d}, d - odd \\ r^{2k-d} \log(r), d - even \end{cases}$

Весовые коэффициенты  $w_j$  и  $q_k$  вычисляются решением системы линейных уравнений (3) размером  $(m+4) \times (m+4)$ :

$$AX = B, \quad (3)$$

где

$$A = \begin{pmatrix} \varphi_1(s_1) & \dots & \varphi_m(s_1) & \pi_1(s_1) & \dots & \pi_4(s_1) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \varphi_1(s_m) & \dots & \varphi_m(s_m) & \pi_1(s_m) & \dots & \pi_4(s_m) \\ \pi_1(s_1) & \dots & \pi_1(s_m) & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \pi_4(s_1) & \dots & \pi_4(s_m) & 0 & \dots & 0 \end{pmatrix},$$

$$X = (w_1, \dots, w_m, q_1, \dots, q_4)^T,$$

$$B = ((s'_1 - s_1), \dots, (s'_m - s_m), 0, \dots, 0)^T.$$

После решения системы линейных уравнений (3), новые координаты внутренних узлов расчетной сетки  $M'$  рассчитываются применением RBF деформации к каждому из узлов:  $v'_i = v_i + d(v_i)$ . Наиболее вычислительно затратной частью алгоритма является процесс решения системы линейных уравнений (3).

## Морфинг в коммерческих системах инженерного анализа

В настоящее время методы сеточного морфинга являются объектом интенсивных исследований. Большинство ведущих коммерческих систем инженерного анализа имеют в своем распоряжении богатый функционал морфинга расчетной сетки. Это такие программные продукты, как Simulia ABAQUS [1], ANSYS Workbench [2], Altair HyperMesh [3] и др.

Одними из популярных и методов являются такие алгоритмы морфинга как «freehand» – свободная деформация, «map to geom» – перенос сетки и «free form deformation» – морфинг узлов сетки.

Операция свободной деформации позволяет деформировать расчетную сетку, вручную задавая перемещение, два набора узлов: фиксированные (чаще всего на границе тел), перемещаемые (перемещение которых задается пользователем) и зависимые ячейки (перемещение узлов которых рассчитывается алгоритмом морфинга).

Пример использования «freehand» [4] операции морфинга представлен на рис. 1. На рис. 1,а показано исходное состояние сетки, на рис. 1,б – состояние сетки после морфинга.

Операция морфинга узлов сетки позволяет деформировать расчетную сетку, помещенную в пространство – клетку. Если структура клетки деформируется, объект внутри решетки также будет деформирован.

На рис. 2 показан пример использования «free form deformation» [5] операции морфинга узлов сетки. На рис. 2,а показано исходное состояние сетки, а на рис. 2,б – состояние сетки после морфинга.

Операция переноса сетки позволяет отображать (переносить) узлы расчетной сетки на ребро или грань. Данный алгоритм позволяет исказить исходную сетку в соответствии с видоизмененной геометрией, а также позволяет упростить и значительно ускорить процесс построения расчетных сеток в задачах с большим количеством смежных деталей без изменения исходной топологии расчетной сетки.

На рис. 3 приведен пример использования операции «map to geom» [6] для переноса сетки с грани на грань. На рис. 3,а показано исходное состояние сетки, на рис. 3,б – результат переноса сетки на грань с помощью операции «map to geom». Перенесенная сетка является искаженной копией исходной: имеет такое же количество узлов и ячеек соответствующих типов.

Получая обратную связь по возможностям препроцессора ЛОГОС-ПРЕПОСТ [7] в части построения сеток для подготовки начальных данных для решения задач прочности, описанный сценарий использования данного метода морфинга был отмечен как самый востребованный.

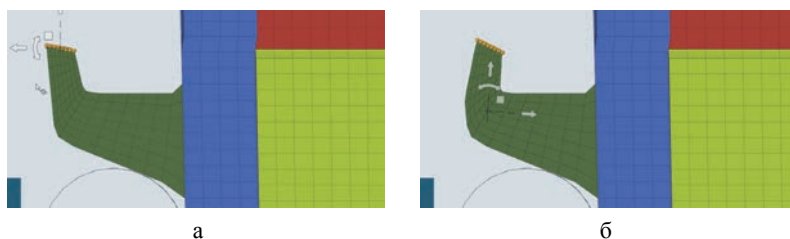


Рис. 1. Пример использования «freehand» операции морфинга:  
а – исходное состояние сетки, б – состояние сетки после морфинга

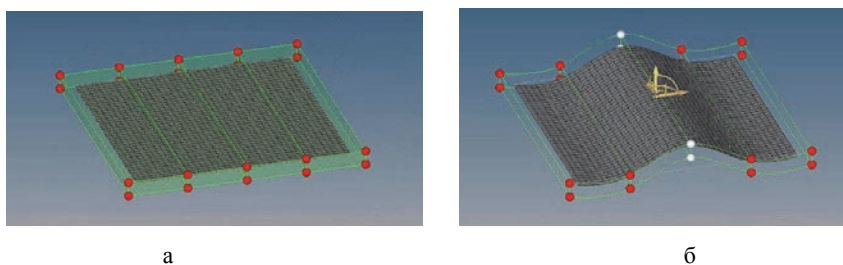


Рис. 2. Пример использования «free form deformation»[5] операции морфинга:  
а – исходное состояние сетки, б – состояние сетки после морфинга

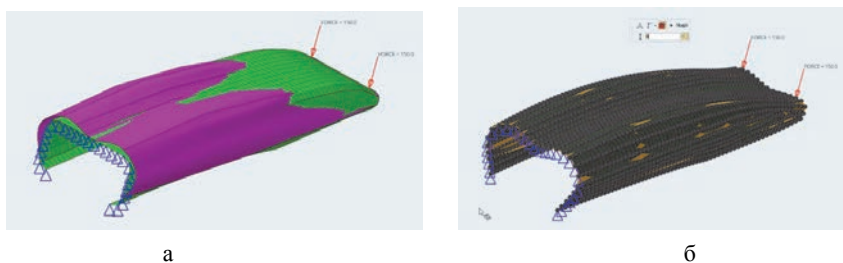


Рис. 3. Пример использования «map to geom» операции морфинга:  
а – исходное состояние сетки, б – состояние сетки после выполнения операции

## Перенос сетки

Для реализации операции переноса сетки на грань был разработан собственный алгоритм. Основной задачей которого является подготовка входных данных для модуля решения задачи морфинга сетки. Входные данные представляют собой два набора узлов – фиксированные и зависимые, и карта векторов смещения фиксированных узлов.

На первом этапе алгоритма выполняется выделение наборов узлов, образующих ребра, виртуальной грани, описываемой исходной поверхностной сеткой. Каждое сеточное ребро представляет собой упорядоченный набор идентификаторов узлов; первый и последний узлы являются вершинами. Критерием того, что  $i$ -ый узел является вершиной и разделяет два сеточных ребра является превышение порогового значения угла, построенного на трех узлах –  $i-1; i; i+1$ .

На рис. 4 узлы сеточных ребер выделены зеленым, желтым, фиолетовым и красным цветом. Узлы, являющиеся вершинами выделены синим цветом. Узлы, входящие в сеточные ребра, передаются в алгоритм морфинга в качестве фиксированного набора. Далее выполняется расчет вектора между центрами габаритных коробок исходной сетки и целевой грани для выполнения грубого переноса. Данный шаг позволяет избежать появления некоторых неоднозначностей в сопоставлении ребер, вызванных взаимным расположением исходной поверхностной сетки и целевой грани.

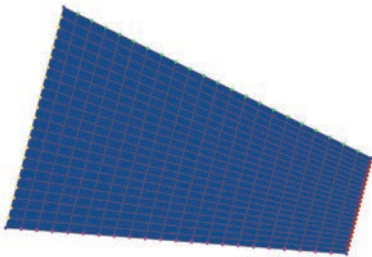


Рис. 4. Сеточные ребра

На втором этапе производится сопоставление выделенных сеточных ребер с ребрами целевой грани и перенос узлов на соответствующие ребра. Первым шагом данного этапа производится сопоставление вершин сеточных и геометрических ребер: составляется карта расстояний между каждой парой узел – вершина; далее итерационно выделяются пары с наименьшим расстоянием; в случае равных расстояний вводится дополнительный критерий – наименьший средний квадрат суммы расстояний между всеми возможными комбинациями пар узел – вершина, отмеченными как неоднозначные. Далее необходимо учесть возможность различия в топологии между поверхностной сеткой и целевой гранью.

В случае, если сеточных вершин, а значит и ребер, было больше чем геометрических, необходимо объединить несколько сеточных ребер в одно, вершины которого будут совпадать с вершинами соот-

ветствующего геометрического ребра. В противном случае, когда сеточных ребер меньше геометрических, необходимо дополнительно разбить сеточные ребра.

Согласно полученным соответствиям узлы сеточных ребер переносятся на геометрические ребра.

Третьим этапом выполняется вычисление векторов перемещения узлов сеточных ребер из исходного положения на соответствующие геометрические ребра. И выделение набора зависимых узлов – внутренних узлов исходной поверхностной сетки. Эти данные необходимы для алгоритма морфинга сетки.

Внешний вид диалога приведен на рис. 5. Данный диалог содержит:

- виджет отбора переносимых ячеек сетки;
- виджет отбора целевой грани;
- выпадающий список с выбором направления переноса (к геометрии, нормаль к геометрии, вдоль вектора);
- блок задания параметров процедуры (ассоциация сетки с целевой гранью, удаление исходной сетки, выбор целевого компонента).

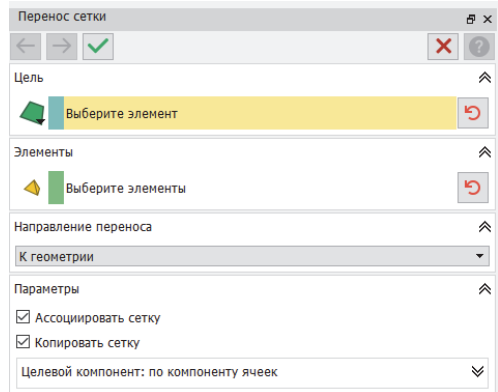


Рис. 5. Внешний вид диалога процедуры переноса сетки

На рис. 6 показан пример использования процедуры переноса сетки на грань при построении расчетной сетки для одной из практических задач: рис. 6,а – задание исходных данных для операции переноса сетки, рис. 6,б – результат выполнения операции переноса сетки на грань.

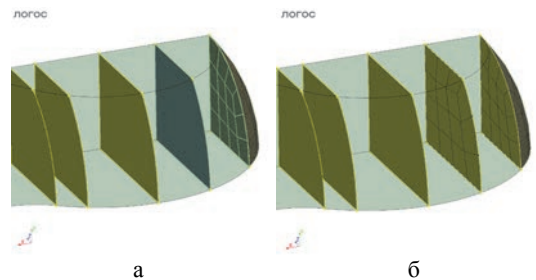


Рис. 6. Пример использования процедуры переноса сетки на грань

Следующим шагом производится построение объемной сетки интерполяцией между двумя наборами поверхностных сеток.

## Свободная деформация

Данная операция основана на реализации алгоритма RBF, математические основы которого описаны выше.

В качестве ядра функции используется

$$\varphi(r) = e^{-(\epsilon r)^2}.$$

Внешний вид диалога приведен на рис. 7. Данный диалог содержит:

- виджет отбора перемещаемых узлов сетки;
- виджет отбора фиксированных узлов сетки;
- виджет отбора ячеек – изменяемых элементов;
- виджет задания параметров трансформации.

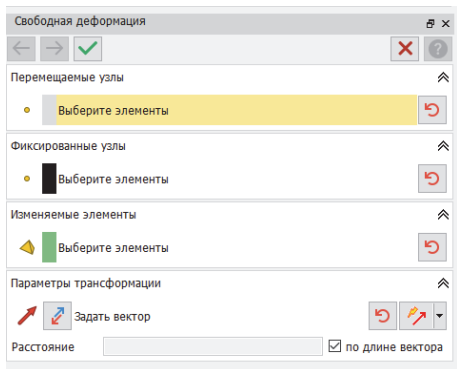


Рис. 7. Внешний вид диалога процедуры свободной деформации

На рис. 8 показан пример использования процедуры свободной деформации для одной из тестовых задач: рис. 8,а – исходное состояние сетки, рис. 8,б – результат выполнения операции свободной деформации сетки.

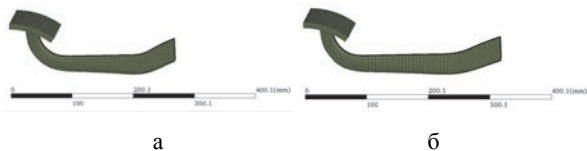


Рис. 8. Пример использования процедуры свободной деформации

## Морфинг узлов сетки

Основная идея техники деформации произвольной формы заключается в том, что существует взаимно однозначное сопоставление между сеткой и решеткой – для каждой точки исходной сетки существует соответствующая точка в деформированной клетке.

Первым шагом необходимо построить локальную параметрическую систему координат внутри клетки, называемую пространством решетки. Пространство решетки задается формулой:

$$X(s, t, u) = X_0 + s \cdot S + t \cdot T + u \cdot U, \quad (4)$$

где  $X_0$  – начало локальной системы координат,  $S, T, U$  – расположены вдоль краев клетки, при этом  $0 < s < 1, 0 < t < 1, 0 < u < 1$ .

Координаты  $X(s, t, u)$  определяются следующим образом:

$$\begin{aligned} s &= \frac{T \times U (X - X_0)}{T \times U \cdot S}, \\ t &= \frac{S \times U (X - X_0)}{S \times U \cdot T}, \\ u &= \frac{S \times T (X - X_0)}{S \times T \cdot U}. \end{aligned} \quad (5)$$

Дальнейшим шагом является нахождение контрольных точек – узлов на решетке  $P_{ijk}$ . Пусть  $l+1$  – число точек в направлении  $S$ ,  $m+1$  – число точек в направлении  $T$ , а  $n+1$  – число точек в направлении  $U$ , тогда:

$$P_{ijk} = X_0 + \frac{i}{l} S + \frac{j}{m} T + \frac{k}{n} U. \quad (6)$$

После того, как сетка параметризована, деформация может быть достигнута путем перемещения точек решетки. Новое положение узлов сетки определяется формулой ниже:

$$\begin{aligned} X_{ijk} &= \sum_{i=0}^l \binom{l}{i} (1-s)^{l-i} s^i \times \\ &\times \left[ \sum_{j=0}^m \binom{m}{j} (1-t)^{m-j} t^j \left[ \sum_{k=0}^n \binom{n}{k} (1-u)^{n-k} u^k P_{ijk} \right] \right]. \end{aligned} \quad (7)$$

Внешний вид диалога приведен на рис. 9. Данный диалог содержит:

- виджет отбора узлов сетки, которые необходимо деформировать;
- блок задания набора разбиений решетки;
- виджет отбора контрольных точек решетки;
- блок демонстрации изменения перемещения узлов решетки при ее деформации.

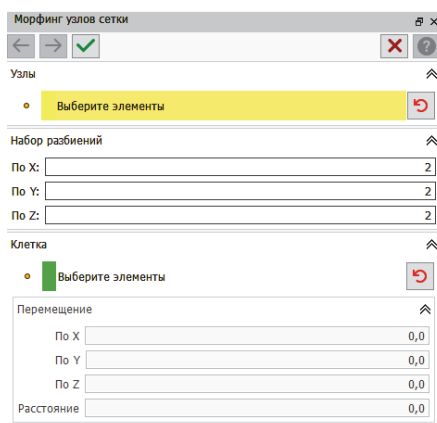


Рис. 9. Внешний вид диалога процедуры морфинга узлов сетки

На рис. 10 показан пример использования процедуры морфинга узлов сетки для одной из тестовых задач: на рис. 10,а – исходное состояние сетки, на рис. 10,б – результат выполнения операции морфинга узлов сетки.

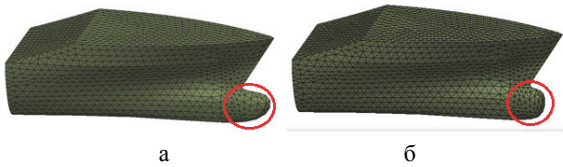


Рис. 10. Пример использования процедуры морфинга узлов сетки: а – исходное состояние сетки, б – результат выполнения операции морфинга узлов сетки

### Макроязык

Одной из ключевых особенностей современных САЕ-пакетов является поддержка макроязыка (наборов макрокоманд) для обработки и подготовки расчетных сеток. Реализация такой поддержки различна в разных коммерческих пакетах. В препроцессоре ЛОГОС-ПРЕПОСТ макрокоманда, описанная в сценарии на языке Python [8], вызывается с помощью утилиты SWIG [9].

Операциям морфинга сетки, выполняемым через графический пользовательский интерфейс (ГПИ), соответствует макрокоманда, которой в качестве аргументов передается информация об объектах, участвующих в операции, и другие необходимые параметры. В качестве аргументов в макрокоманды передаются не сами объекты, а их идентификаторы. Это позволяет контролировать время жизни объектов модели данных и избегать аварийных остановок ЛОГОС-ПРЕПОСТ при попытках доступа к уже несуществующим объектам.

Место подсистемы макроязыка схематично изображено на рис. 11.



Рис. 11. Место подсистемы макроязыка, реализующего средства макроязыка в компоненте ЛОГОС-ПРЕПОСТ

Использование макрокоманд позволяет создавать необходимые инженеру сценарии автоматического задания параметров и выполнения цепочек операций без использования их ГПИ.

В качестве примера рассмотрим автоматизацию алгоритма построения сетки с помощью макрокоманд на языке Python.

```
Python 2.6.5 (r265:79063, Feb 2 2016, 11:03:55) [MSC v.1800 64 bit (AMD64)] on win32
>>> m=mdl.GetModel(1)
>>> nodes=[]
>>> for x in range(0,1800): nodes.append(x)
...
>>> partitions=(2,2,2)
>>> offsets=[ 0:[3, 0, 0], 1:[3, 0, 0], 2:[3, 0, 0], 3:[3, 0, 0] ]
>>> MorphNodes(m, nodes, partitions, offsets)
0
>>> offsets=[ 0:[0, 3, 0], 1:[0, 3, 0], 4:[0, 3, 0], 5:[0, 3, 0] ]
>>> MorphNodes(m, nodes, partitions, offsets)
0
>>> |
```

Рис. 12. Сценарий операции морфинга узлов сетки на языке Python

В сценарии, представленном на рис. 12, приведен пример следующего алгоритма:

- задание набора узлов, над которыми будет совершаться операция;
  - задание числа разбиений на решетке в каждом направлении;
  - задание пар: номер перемещаемого узла решетки, вектор перемещения;
  - выполнение операции морфинга узлов сетки.
- Результат работы сценария приведен на рис. 13.

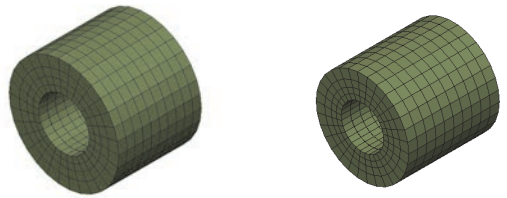


Рис. 13. Результат работы сценария на макроязыке

Для сравнения: время исполнения данного сценария занимает не более 2 секунд, тогда как выполнение данного сценария инженером с использованием ГПИ операций потребует не менее 5 минут.

### Заключение

Одним из ключевых и наиболее трудозатратных этапов подготовки расчетной модели является процесс сеткогенерации. Современные расчетные сетки задач прочности могут включать сотни тысяч и миллионы элементов. Создание расчетных сеток такой размерности вручную не представляется возможным, поэтому в современных инженерных пакетах реализован ряд автоматических, полуавтоматических и ручных алгоритмов и методов генерации расчетных сеток.

Ускорить процесс подготовки сеточной модели для сложных геометрических моделей и значительно автоматизировать подготовку оптимизационных задач, в которых невозможно использование методов автоматической сеткогенерации, позволяют методы морфинга.

В рамках данной работы описаны методы деформации расчетных сеток в препроцессоре ЛОГОС-ПРЕПОСТ. Описана общая идея технологии морфинга, базовый математический алгоритм, приведены примеры использования разработанных средств. Ключевыми направлениями развития методов морфинга расчетной сетки являются: реализация новых возможностей, оптимизация, повышение стабильности и доработка уже существующих возможностей. Реализованные возможности внесены в основную ветвь разработки ЛОГОС-ПРЕПОСТ и будут доступны пользователям в очередном релизе ЛОГОС.

### Список литературы

1. Simulia ABAQUS [Электронный ресурс]: – URL: <http://www.3ds.com/ru/produkty-i-uslugi/simula/produkty/abaqus> (дата обращения: 12.05.2021).

2. ANSYS Workbench: Engineering Solutions is a modeling and visualization environment [Электронный ресурс] – URL: <http://ansys.com/Products/Platform> (дата обращения: 21.05.2021).

3. Altair HyperMesh: Engineering Solutions is a modeling and visualization environment [Электронный ресурс] – URL: <http://www.altairengineering.ru/hyperMesh> (дата обращения: 03.06.2021).

4. YouTube [Электронный ресурс]: – URL: <https://www.youtube.com/watch?v=JzpiJuguXPo> (дата обращения: 20.07.2021).

5. YouTube [Электронный ресурс]: – URL: <https://www.youtube.com/watch?v=fjgfNy3M7vc> (дата обращения: 20.07.2021).

6. YouTube [Электронный ресурс]: – URL: <https://www.youtube.com/watch?v=glJ0хуJExl8> (дата обращения: 20.07.2021).

7. ЛОГОС-ПРЕПОСТ [Электронный ресурс]: – URL: <http://logos.vniief.ru/products/prepost> (дата обращения: 14.07.2021).

8. Python: About Python [Электронный ресурс] – URL: <https://www.python.org/about> (дата обращения: 02.07.2021).

9. SWIG: Simplified Wrapper and Interface Generator [Электронный ресурс] – URL: <http://swig.org/exec.html> (дата обращения: 02.07.2021).