

## **СЦЕНАРИЙ НА ЯЗЫКЕ IRONPYTHON ДЛЯ МОДЕЛИРОВАНИЯ ЯЧЕИСТОГО ТЕЛА**

**Напрасников В.В.<sup>1</sup>, Полозков Ю.В.<sup>1</sup>,  
Соловьев А.Н.<sup>2</sup>, Скалиух А.С.<sup>3</sup>**

- 1) Минск, Белорусский национальный технический университет,  
2) Ростов-на-Дону, Донской государственный технический университет,  
3) Ростов-на-Дону, Южный федеральный университет

При проектировании многих изделий в промышленности существенным требованием является снижение материалоемкости объектов массового производства в условиях сохранения прочности и жесткости. Для выполнения этих требований геометрическая модель объекта должна быть параметрической, поскольку в дальнейшем она должна использоваться в блоке, выполняющем оптимизацию. Построение такой модели часто связано с определенными трудностями, поскольку выбранные геометрические модели имеют ряд ограничений.

Встроенный в среду Ansys Workbench моделиер SpaceClaim обладает рядом преимуществ, одним из которых является наличие инструмента для создания сценариев на языке IronPython, позволяющего исследователю существенно упростить подготовку модели.

В качестве примера использования такого подхода рассматривается задача разделения исходного тела с помощью «фрезы», представленная на рисунке 1 слева. Здесь исходное тело имеет коричневый цвет, а тело «фрезы» зелены.

Контур проекции основного тела предварительно строится по замеренным точкам с помощью сплайна, затем контур заполняется и вытягивается. Отдельный резец «фрезы» представляет собой тело, получаемое вытягиванием задаваемого предварительно сечения вдоль винтовой кривой, которая строится с помощью другого сплайна.

Особенностью предлагаемого алгоритма является то, что при построении «фрезы» необходимо объединить все резцы с предварительно созданной подошвой для образования единого тела.

Это связано с особенностью инструмента, используемого для разделения тела с целью получения ячеистой структуры. Если попытаться выполнить такое разделение предварительно созданным массивом резцов без указанного объединения, то операция выполняется некорректно и прерывается при достижении зуба, не пересекающегося с основным телом. При этом выполнение программы прерывается.

Обратим внимание, что предложенный сценарий может быть легко модифицирован для другого основного тела и другого вида зуба «фрезы».

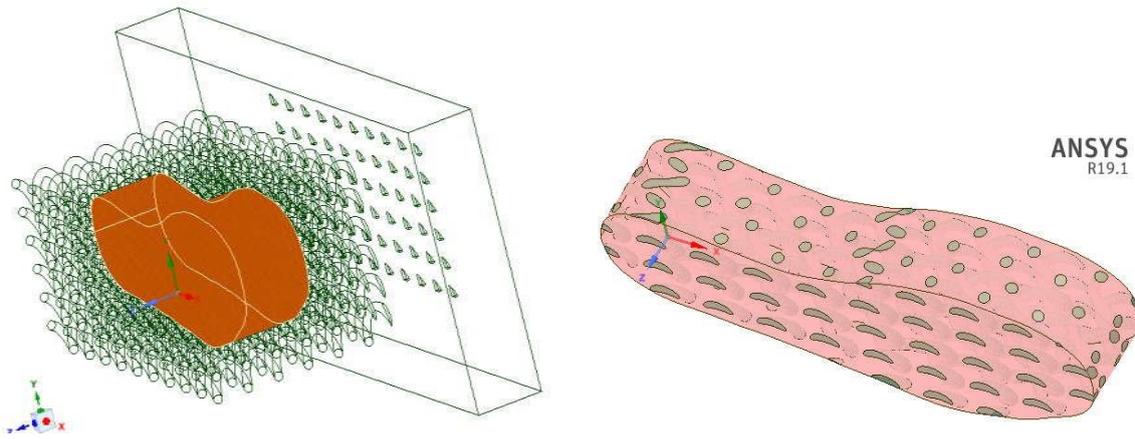


Рисунок 1. Исходное тело и «фреза» (слева) и результат выполнения сценария (справа)

На рисунке 2 показаны фрагменты сценария на языке IronPython с комментариями, запуск которого обеспечивает выполнение нужных операций.

```

1 | # Python Script, API Version = V17
2 |
3 | document = DocumentHelper.CreateNewDocument()
4 |
5 | import math
6 |
7 | def funcWINT(R,H_angle, H,K_point):
8 |     #___ Создаем пустой набор точек с именем ptList
9 |     ptList = List[Point]()
10 |    #___ Добавляем точки в набор ptList
11 |    for i in range(0, K_point):
12 |        ALFA=H_angle*i;
13 |        x1=MM(R*math.cos(ALFA));
14 |        y1=MM(R*math.sin(ALFA));
15 |        z1=MM(i*H)
16 |
17 |        ptList.Add(Point.Create(x1, y1, z1))
18 |    ncurve = NurbsCurve.CreateThroughPoints(False, ptList, 0.0001)
19 |    curveSegment = CurveSegment.Create(ncurve)
20 |    designCurve = DesignCurve.Create(GetRootPart(), curveSegment)
21 |    #_____ Слайн построен
22 |    return ptList
23 |
219 | # Объединить объекты
220 |
221 | for i in range(1, Count_Comp_in_massiv+1):
222 |     print ' i=', i
223 |     telo1=GetRootPart().Bodies[1]
224 |     # Ниже индекс 0 в Components[0] потому,
225 |     #что объединенное тело telo2 удалится после объединения!!!
226 |     telo2=GetRootPart().Components[2].Content.Bodies[0]
227 |     targets = Selection.Create([telo1,telo2])
228 |     result = Combine.Merge(targets, None)
229 | # EndBlock
230 |
231 |
232 | # Переименовать "Твердое тело" в "FREZA"
233 | selection = Selection.Create(GetRootPart().Bodies[1])
234 | result = RenameObject.Execute(selection,"FREZA")
235 | # EndBlock
236 |
237 |
238 | # Создать пересечение объектов
239 | targets = Selection.Create(GetRootPart().Bodies[2])
240 | tools = Selection.Create(GetRootPart().Bodies[1])
241 | options = MakeSolidsOptions()
242 | options.KeepCutter = False
243 | result = Combine.Intersect(targets, tools, options, None)
244 | # EndBlock

```

Рисунок 2. Фрагменты сценария на языке IronPython

На рисунке 1 справа представлен результат выполнения сценария для одного из возможных случаев основного тела и зуба «фрезы». Здесь решена задача о построении облегченного варианта обувной подошвы.

Таким образом в результате выполненной работы обоснован выбор современного геометрического моделиера SpaceClaim, который может работать как из-под Ansys Workbench, так и автономно;

разработан алгоритм и получена его реализация в виде сценария на языке IronPython для параметрического моделирования ячеистого тела; получены результаты моделирования для реального объекта.