

при этом уменьшив максимальное значение нагрузки в 1,5 раза по сравнению со случаем использования ячеек равного размера и фиксированной толщиной стенок. Однако, несмотря на сокращение массы детали по сравнению с монолитной ее версией, при использовании адаптивного масштабирования ячеек масса детали увеличилась на 15 % по сравнению с деталью с ячейками равного размера. Последнее объясняется тем, что в рассмотренный алгоритм ориентирован на масштабирование ячеек в наиболее нагруженных областях детали и максимальный размер ячеек равен размеру конечных элементов с учетом минимально заданной толщины стенки между ними. Это ограничивает возможность удаления наибольшего количества материала в наименее нагруженных областях детали. В связи с этим перспективные исследования связываются с совершенствованием представленного алгоритма, за счет объединения и удаления смежных конечных элементов, т.е. увеличения размеров создаваемых полых ячеек в наименее нагруженных областях, что поспособствует более существенному сокращению массы детали.

УДК 004.632.5:004.415.2

АВТОМАТИЗАЦИЯ КОНСТРУИРОВАНИЯ ШЕСТИГРАННЫХ ЯЧЕИСТЫХ СТРУКТУР ДЛЯ ПРОЕКТИРОВАНИЯ ЛЕГКОВЕСНЫХ ДЕТАЛЕЙ

Руцкий В.А.

Научный руководитель – Полозков Ю.В., к.т.н., доцент

Актуальность автоматизации проектирования легковесных деталей с ячеистыми структурами обусловлена требованием к снижению материалоёмкости деталей, которые при этом должны сохранять свою функциональность. В связи с этим рассматривается задача разработки и программной реализации алгоритмов конструирования шестигранных ячеистых структур на основе *API Solid Works*, предназначенных для выработки наиболее рационального варианта интеграции ячеистых структур в процессе многовариантного проектирования легковесных деталей с учетом проведения инженерного анализа.

Согласно общей методике проектирования легковесных деталей на основе инженерного анализа определяются области интеграции ячеистых структур. Найденные области представляют собой параллелепипеды с известными узловыми точками, на основе координат которых вычисляется габариты, объем и другие геометрические параметры. Поэтому в разработанном программном обеспечении в качестве тестового объекта для задания ячеистой структуры была принята модель параллелепипеда.

В математической модели параметрами задания шестигранной ячеистой структуры являются: геометрическая форма основания ячейки (количество углов, в данном случае шесть, основания прямой правильной призмы, которую представляет ячейка); количество рядов и строк, которые занимает структура (степень матрицы ячеистой структуры); высота ячейки; объём, занимаемый ячеистой структурой; расстояние (толщина стенок) между ячейками. Количество углов многоугольника, степень матрицы, высота ячейки являются исходными данными. Объём одной ячейки вычисляется как функция от объёма всей ячеистой структуры и степени матрицы. Расстояния ячеек друг от друга будут определяться индивидуально для всех случаев построения. Если ширина грани больше её длины, горизонтальное и вертикальное расстояние будут разными. Каждая ячейка имеет свой центр, который служит базовой точкой для расчёта координат вершин шестиугольника. Если исходной точкой является (X, Y, Z) , то центр следующей ячейки вычисляется с учётом её смещения относительно предыдущей. – $\Delta X = 1,5 R$, $\Delta Y = \sqrt{3} R$ – смещения центров. Координаты вершин шестиугольника вычисляются на основе центральной точки, радиуса и угловых коэффициентов $(\pi/3 \times i, 0 \leq i \leq 5)$.

Для обеспечения условия постоянства объема, приходящегося на ячеистые структуры, используются разработанные математические модели расчета размеров и положения многоугольников и окружностей, задающих основания многогранных и цилиндрических ячеек в зависимости от заданного количества ячеек в ячеистой структуре, ее объема, высоты ячеек и размеров базового тела детали. Особенность разработанных моделей состоит в обеспечении взаимной непересекаемости ячеек и в непересекаемости ячеек с боковыми гранями модели тела (области интеграции). Выходными параметрами расчета являются радиус описывающей окружности многоугольника, задающего i -ю ячейку и размер квадрата, в котором должна размещаться эта ячейка. В дальнейшем эти параметры используются для определения положения ячеек и передаются в соответствующие функции *API* для отрисовки двумерных примитивов. Фрагмент процедур расчета представлен на следующем листинге кода:

```
private double calculateSideLength(int numberComb) // Вычисляет длину стороны ячейки по размерам
области и числу колонок
{
    double sideLength = (iteration.Lx - distanceBetweenCombs * (numberComb - 1)) / (coefficient * numberComb);
// Формула: доступная длина по X делится на число "коэффициентных" отрезков; coefficient =  $\sqrt{3}$  для корректной
плотности упаковки
    return sideLength; // Возвращаем вычисленную длину стороны
}

int numberOfCumbs = ((6 * (n - 1) + 2) * n) / 2 - (n - 1); // Вычисление ожидаемого числа ячеек для заданной
"степени" n

// Вычисляет координаты шести соседних центров относительно заданного центра centerOld
```

```

private void calculateNewCenterCoordinates(Point3D centerOld, double sideLengthParam, double
distanceBetweenCombsParam)
{
    double distanceBetweenCenters = sideLengthParam * coefficient + distanceBetweenCombsParam; //Расстояние
    между центрами по вертикали (и вертикально-ориентированные смещения):  $R\sqrt{3} + d$ 
    double xDist = (3 * sideLengthParam + distanceBetweenCombsParam * coefficient) / 2; // Горизонтальное
    смещение для диагональных рядов
    double yDist = (sideLengthParam * coefficient + distanceBetweenCombsParam) / 2; // Вертикальное смещение
    для диагональных рядов
    Point3D pointNewTop = new Point3D(centerOld.X, centerOld.Y + distanceBetweenCenters, 0); // Новый центр
    сверху относительно текущего
    Point3D pointNewBottom = new Point3D(centerOld.X, centerOld.Y - distanceBetweenCenters, 0); // Новый
    центр снизу относительно текущего
    Point3D pointNewLeftTop = new Point3D(centerOld.X - xDist, centerOld.Y + yDist, 0); // Левый верхний
    диагональный центр
    Point3D pointNewLeftBottom = new Point3D(centerOld.X - xDist, centerOld.Y - yDist, 0); // Левый нижний
    диагональный центр
    Point3D pointNewRightTop = new Point3D(centerOld.X + xDist, centerOld.Y + yDist, 0); // Правый верхний
    диагональный центр
    Point3D pointNewRightBottom = new Point3D(centerOld.X + xDist, centerOld.Y - yDist, 0); // Правый нижний
    диагональный центр
    List<Point3D> pointsToAdd = new List<Point3D> { pointNewTop, pointNewBottom, pointNewLeftTop,
    pointNewLeftBottom, pointNewRightTop, pointNewRightBottom }; // Список шести потенциальных новых центров
    }.

```

В процессе программной реализации приложения были разработаны следующие основные классы: *Point3D* – структура для задания координат точек (вершин основания шестигранных ячеек) в 3D-пространстве; *Iteration* – класс расчёта габаритов заданной области и создания базовой плоскости с заданными смещениями по осям *Y* и *Z*. На этой плоскости затем создаются эскизы для формирования ячеек. *Honeycomb* и *HexagonalCellBuilder* – основной алгоритмический блок, реализующий построение ячеек, в частности, метод создания эскиза шестиугольника и метод конструирования 3D-призмы путем лофт-выдавливания; *Remover* – для очистки предыдущих вырезов при генерации новых ячеистых структур.

Конструирование ячеистых структур осуществляется с использованием API-функций, которые позволяют в среде *Solid Works* выполнять отрисовку вспомогательной плоскости (*CreatePlaneThru3Points3*), построенной на трех точках-узлах (*SketchManager.CreatePoint*) и трёхмерного эскиза (*SketchManager.Insert3DSketch*), в котором четвертый узел и узлы вспомогательной плоскости соединяются отрезками. Далее в модели посредством операции «Вырезать по траектории» (*FeatureManager.FeatureCut2*) создаются конструктивные элементы, соответствующие полым шестигранным ячейкам. Пример конструирования ячеистых структур с помощью разработанного программного средства представлен на рисунке 1.

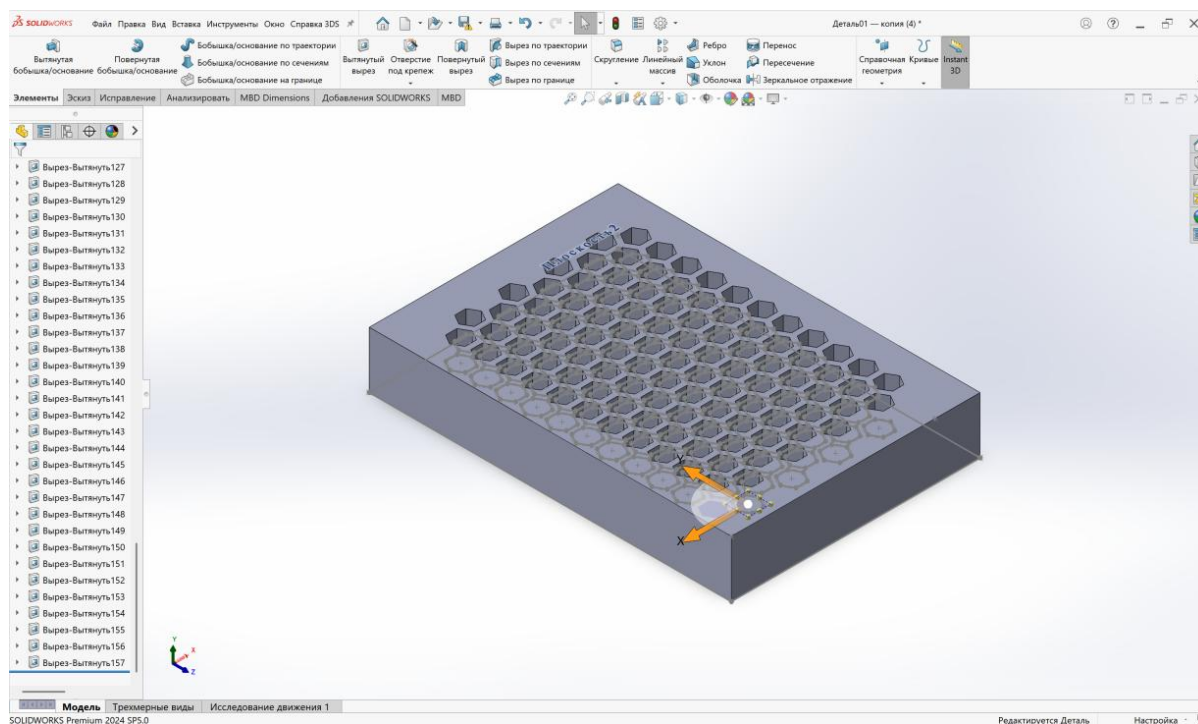


Рис.1. Результат конструирования шестигранных ячеистых структур.

Разработанное программное средство предназначено для автоматизации многократного перепроектирования геометрии исходной модели монолитной детали путем интеграции ячеистых структур, включающих правильные шестигранные отверстия, размеры, относительное положение в области интеграции и толщина стенок, которых должны будут параметрически задаваться в зависимости от результатов проведения многовариантного инженерного анализа проектируемой детали.

УДК 004.921

DESIGN OF A VIRTUAL EXHIBITION HALL FOR ANCIENT CHINESE WEAPONS BASED ON UNITY3D

YuanYiyi

Scientific Supervisor – Martinovich V.A., Ph.D. in Physics and Mathematics,
Associate Professor

There were many wars in ancient Chinese history. Frequent wars also led the ancients to forge a wide variety of weapons, the most well-known of which are "swords, spears, swords, and halberds". Different weapons are suitable for different scenarios and uses, such as knives are suitable for close combat, swords are suitable for stabbing and swinging, and spears are suitable for long-range attacks. In order to better inherit these cultural heritages, this project uses Unity3D to build a virtual weapons exhibition hall. Through its own display and promotion