

ВЫЧИСЛЕНИЕ КРИВИЗНЫ ДИСКРЕТНЫХ ПОВЕРХНОСТЕЙ В ГЕНЕРАТОРЕ ПОВЕРХНОСТНЫХ ТРИАНГУЛЯЦИОННЫХ СЕТОК ЛОГОС.ПРЕПОСТ

А. И. Белова, А. Н. Лукичев, О. Н. Борисенко

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

Введение

ЛОГОС.ПреПост является программным продуктом для задания начальных данных и генерации сеток в задачах инженерного анализа. Программный продукт обеспечивает единую графическую среду, предоставляющую следующие возможности:

- загрузка и исправление математической модели исходной конструкции, заданной во внешней САД-системе;
- задание граничных условий и начальных данных и их расчет;
- задание параметров сеток и построение сеточных моделей;
- подготовка и запуск инженерных счетных кодов для получения решения;
- визуальный анализ полученных результатов.

Одной из возможностей данного продукта является построение поверхностной триангуляционной сетки для геометрий, заданных в фасеточном представлении. Основная проблема, связанная с фасеточным способом описания поверхностей, заключается в том, что при таком представлении теряется большая часть информации о поверхности: значения производных в каждой точке поверхности, значения кривизны и другие метрики.

В данной работе рассмотрены различные алгоритмы вычисления кривизны дискретных поверхностей и их модификации, используемые в генераторе поверхностных сеток ЛОГОС.ПреПост. Приведены результаты применения рассматриваемых алгоритмов и результаты построения поверхностных сеток с их использованием.

Описание проблемы

При построении поверхностной сетки поверхностным генератором ЛОГОС.ПреПост вычисляются размеры элементов, на основе которых происходит перестроение исходной сетки. Далее эти размеры будем называть метриками. Принцип работы поверхностного генератора таков, что при построении сетки размер треугольников в окрестностях элемента, в котором задана метрика (элементом может быть вершина сетки, ребро, грань), стремится к значению метрики.

Первоначально в поверхностном генераторе ЛОГОС.ПреПост метрики вычислялись на основе размера, заданного пользователем. Для того чтобы

получить хорошую сетку, нужно разбить исходную геометрию на части – патчи, и на каждом патче задать свой размер. Для больших геометрий это трудоемко, так как для некоторых из них придется задавать метрику на каждом элементе. Если задать маленький размер на всей геометрии, то поверхностный генератор выдаст чрезмерно подробную сетку. Если размеры задать достаточно большими, то поверхностная сетка может неточно описывать исходную геометрию в местах ее искривления.

На рис. 1 изображена геометрия cubit: исходная геометрия (рис. 1а) и желаемая поверхностная сетка (рис. 1б).

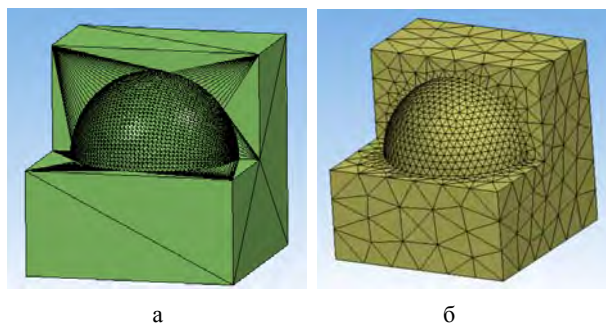


Рис. 1. Геометрия cubit: а – исходная геометрия, б – желаемая поверхностная сетка

На рис. 2 приведены результаты построения поверхностной сетки поверхностным генератором ЛОГОС.ПреПост с разными пользовательскими размерами.

Исходная геометрия не разбита на части (рис. 1а), пользовательские данные заданы для всей геометрии (рис. 2а,б).

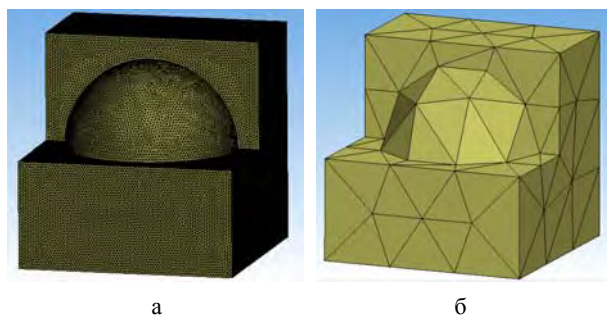


Рис. 2. Результаты построения поверхностной сетки геометрии cubit с разными пользовательскими размерами: а – с размером 0,005; б – с размером 0,003

Из рис. 2,б видно, что, если задан большой размер, теряются основные особенности исходной геометрии в местах ее искривления. Для решения этой проблемы в генератор поверхностных сеток ЛОГОС.ПреПост добавлено вычисление метрик в узлах сетки на основе кривизны. Считается метрика по следующей формуле:

$$s = \frac{1}{k} \sqrt{2 - 2 \cdot \cos\left(\frac{2\pi}{N}\right)},$$

где s – значение метрики, k – вычисленное значение кривизны в точке, N – параметр разбиения окружности на N равных частей.

Для того чтобы размер треугольников на плоских частях был большим, а на искривленных маленьким, кривизна должна удовлетворять следующим требованиям:

- 1) значение кривизны на плоскости должно быть нулевым;
- 2) значения кривизны в точках пересечения плоскостей должны быть нулевыми;
- 3) значения кривизны на сферической, цилиндрической поверхности должны быть одинаковы;
- 4) значения кривизны на стыке искривленных поверхностей и плоскости должны быть близки к значению кривизны в ближайших точках искривленной поверхности;
- 5) не должно быть сильных перепадов значений кривизны в соседних точках.

В следующих разделах приведены алгоритмы вычисления кривизны на дискретных поверхностях, которые были опробованы в поверхностном триангуляционном генераторе ЛОГОС.ПреПост для решения описанной проблемы. Эти алгоритмы могут быть использованы в любом другом генераторе сеток.

Алгоритм Gauss – Bonnet

Алгоритм состоит из двух логических частей: нахождение гауссовой кривизны и нахождение средней кривизны. Для каждой вершины сетки осуществляется обход по первому кольцу треугольников (рис. 3). В первое кольцо входят треугольники, инцидентные вершине v .

Гауссова кривизна находится по формуле:

$$K = \frac{2\pi - \sum_{i=0}^{n-1} \alpha_i}{\frac{1}{3}A},$$

где $\alpha_i = \angle(v_i, v, v_{(i+1) \bmod n})$, $0 \leq i \leq n-1$ – угол между двумя ребрами треугольника, которые инцидентны вершине v , A – площадь треугольников первого кольца.

Нахождение средней кривизны осуществляется по формуле:

$$H = \frac{\frac{1}{4} \sum_{i=0}^{n-1} \|e_i\| \beta_i}{\frac{1}{3}A},$$

где $\|e_i\|$ – длина ребра e_i , $\beta_i = \angle(\overline{N}_i^v, \overline{N}_{(i+1) \bmod n}^v)$.

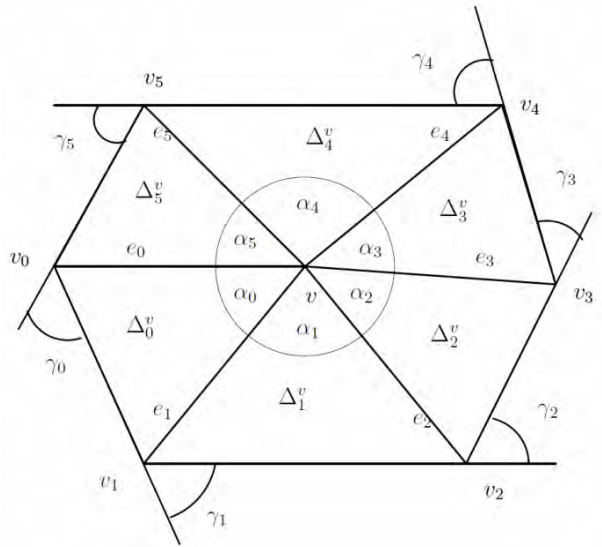


Рис. 3. Первое кольцо треугольников

Угол β_{cur} считается по формуле:

$$\beta_{cur} = \arccos\left(\frac{\overline{N}_{cur} \cdot \overline{N}_{next}}{|\overline{N}_{cur}| \cdot |\overline{N}_{next}|}\right),$$

где \overline{N}_{cur} и \overline{N}_{next} нормали текущего и следующего треугольников в обходе треугольников первого кольца соответственно. Нормаль к треугольнику находится по определению векторного произведения двух векторов, принадлежащих плоскости треугольника.

На рис. 4 и 5 показано распределение значений гауссовой кривизны и средней кривизны соответственно, найденных по алгоритму Gauss – Bonnet.

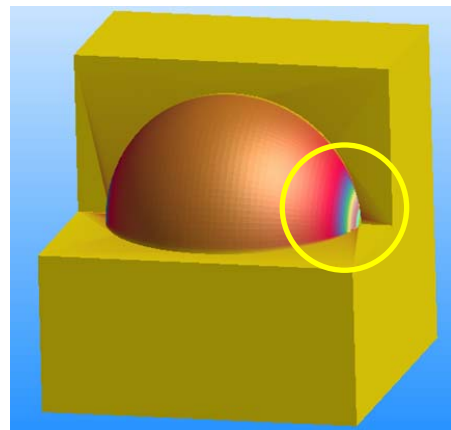


Рис. 4. Распределение значений гауссовой кривизны, полученных в результате работы алгоритма Gauss – Bonnet

На рис. 4 и 5 в выделенных окружностью местах наблюдается сильное изменение значений кривизны. Также в некоторых точках пересечения плоскостей наблюдались ненулевые значения.

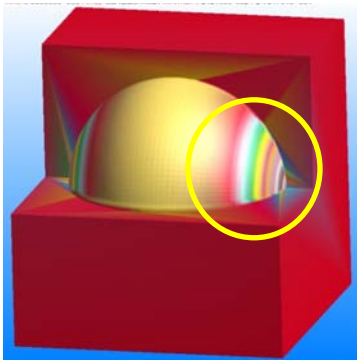


Рис. 5. Распределение значений средней кривизны, полученных в результате работы алгоритма Gauss – Bonnet

Алгоритм Watanabe K., Belyaev A. G.

Для каждой вершины сетки осуществляется обход по первому кольцу треугольников (рис. 3). На каждом шаге обхода требуются два смежных треугольника, назовем их текущий – cur и предыдущий – prev; индексы, соответствующие рассматриваемым треугольникам, обозначим i и $(i-1)$ соответственно.

Для каждой точки сетки нормаль находится по формуле:

$$\overline{N}_v = \frac{\sum_{i=0}^{n-1} N_i^v A_i}{\left\| \sum_{i=0}^{n-1} N_i^v A_i \right\|},$$

где \overline{N}_i^v – нормаль треугольника, A_i – площадь треугольника.

Для каждого треугольника вычисляются значения φ_i и k_n по следующим формулам:

$$\varphi_i = \arccos \left(\frac{\overline{a} \cdot \overline{b}}{\|\overline{a}\| \cdot \|\overline{b}\|} \right),$$

где a и b – вектора соответствующие двум ребрам треугольника, которые образуют угол φ .

$$k_n = \frac{2\overline{N}_v \cdot \overline{vv}_i}{\|\overline{vv}_i\|^2},$$

где \overline{N}_v – нормаль в рассматриваемой вершине v , \overline{vv}_i – вектор, направленный из рассматриваемой вершины, соответствующий общему ребру текущего и предыдущего треугольника.

По окончании обхода вычисляется средняя и гауссова кривизна в рассматриваемой точке сетки.

Подсчет средней кривизны в вершине сетки осуществляется по формуле:

$$H \approx \left(\sum_{i=0}^{n-1} k_n^i \left(\frac{\varphi_i + \varphi_{(i-1) \bmod n}}{2} \right) \right) / 2\pi.$$

Подсчет гауссовой кривизны в вершине сетки происходит по формуле, полученной элементарными алгебраическими преобразованиями из формулы:

$$K \approx 3H^2 - \left(\sum_{i=0}^{n-1} k_n^i \left(\frac{\varphi_i + \varphi_{(i-1) \bmod n}}{2} \right) \right) / \pi.$$

На рис. 6 и 7 показано распределение значений гауссовой кривизны и средней кривизны, соответственно, найденных по алгоритму Watanabe K., Belyaev A. G.

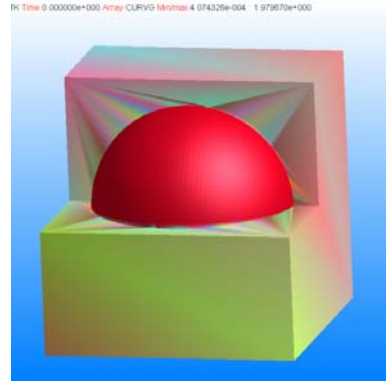


Рис. 6. Распределение значений гауссовой кривизны, полученных в результате работы алгоритма Watanabe K., Belyaev A. G.

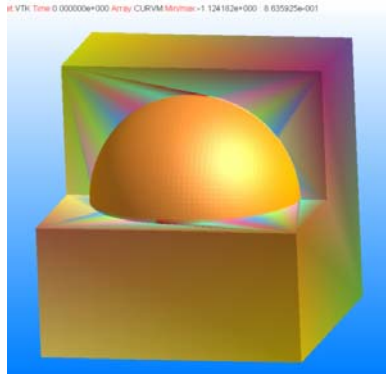


Рис. 7. Распределение значений средней кривизны, полученных в результате работы алгоритма Watanabe K., Belyaev A. G.

На рис. 6 и 7 можно заметить, что в углах куба получены ненулевые значения кривизны. Также на стыке сферы и плоскости можно наблюдать разброс значений кривизны, эти значения отличаются от значений на сферической поверхности.

Практическое применение алгоритмов Watanabe K., Belyaev A. G и Gauss – Bonnet

Результаты вычисления значений кривизны представленных алгоритмов зависят от площади первого кольца, а это значит, что значения кривизны в одной и той же точке геометрической модели при различной тесселяции исходной геометрической модели могут отличаться друг от друга. Значения кривизны

в граничных точках завышаются, так как чаще всего в этих точках не достигается нужная степень гладкости.

Главная максимальная кривизна при вычислении метрик является более приоритетной величиной по сравнению с гауссовой кривизной или средней кривизной. В алгоритмах Gauss – Bonnet и Watanabe K., Belyaev A. G вычисление главных кривизн требует использования дополнительных арифметических операций, которые могут увеличить погрешность вычисления.

Алгоритм Cohen-Steiner D., Morvan J.-M.

На рис. 8 и 9 представлены геометрии cubit и skoba, с выделенными особенностями соответственно, и результаты работы алгоритма Cohen-Steiner D., Morvan J.-M.

Введем следующие понятия:

– *характерная кривая* – набор ребер исходной геометрии (рис. 8а и 9а), которые нельзя перестраивать. Такие кривые обычно выделяют топологические особенности геометрии;

– вершину будем называть *граничной*, если она лежит на характерной кривой, вершина, не лежащая на характерной кривой, является *внутренней* (рис. 8а и 9а).

Алгоритм, описанный в [5] авторами Cohen-Steiner D., Morvan J.-M., основан на статье Taubin [3], в которой используется связь тензора кривизн с собственными векторами и собственными числами.

Для каждой внутренней вершины выполняются следующие действия:

1) проводится поиск области, на которой будет вычисляться кривизна.

Данная область состоит из всех треугольников первого кольца и треугольников, удовлетворяющих следующим условиям:

- треугольник не отделен от найденной области характерной кривой;
- все вершины треугольника лежат на расстоянии меньше заданного от исследуемой точки;
- общее количество треугольников в области ограничено заданным значением;

2) вычисляются матрицы из векторов, соответствующих ребрам каждого треугольника из найденного набора. Каждый вектор умножен на коэффициент a :

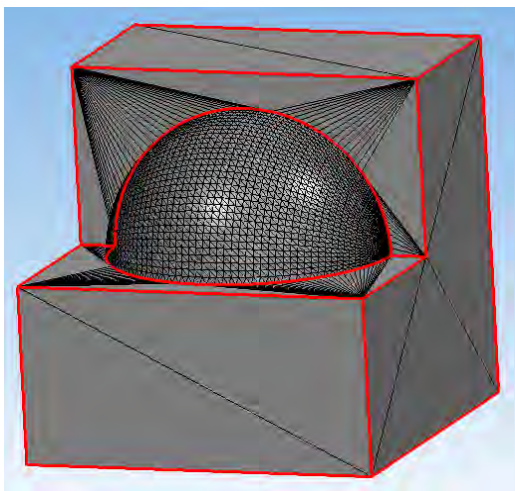
$$a = \text{alfa} \cdot |e|,$$

где alfa – угол между треугольниками инцидентных ребру e , $|e|$ – длина ребра.

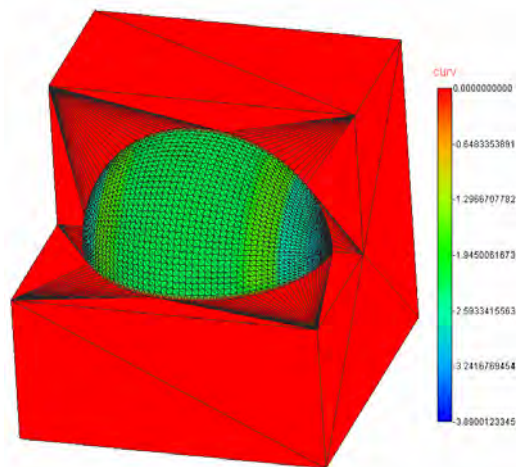
3) для построенной матрицы вычисляются собственные вектора и соответствующие им собственные числа, которые, в свою очередь, являются главными направлениями кривизн и их значениями соответственно.

Результаты работы алгоритма Cohen–Steiner D., Morvan J.-M. представлены на рис. 8б и 9б.

Преимуществом этого алгоритма является то, что он выдает достаточно точные значения, а также в результате работы алгоритма можно получить направления главных кривизн. Недостаток данного алгоритма заключается в том, что вычисление кривизны происходит только во внутренних точках. Если все точки геометрии лежат на характерной кривой, то вычисление значений кривизн не производится (рис. 9б).

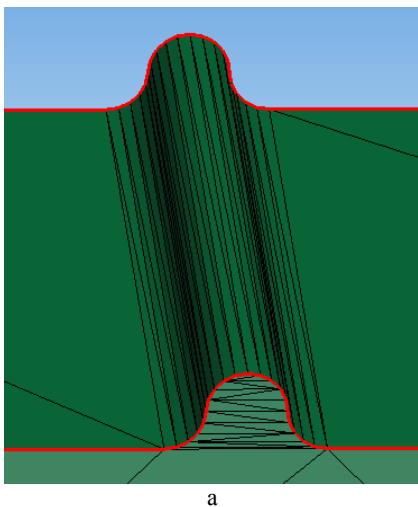


а

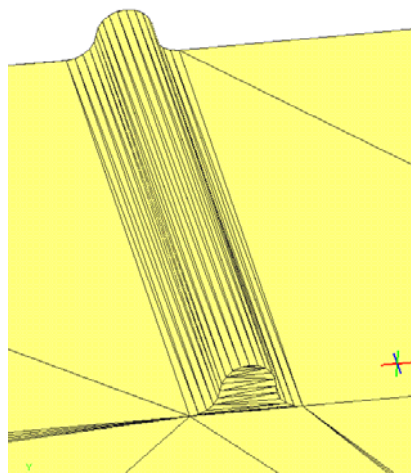


б

Рис. 8. Геометрия cubit: а – красной линией выделены характерные ребра геометрии cubit, б – распределение значений кривизны, полученных в результате работы алгоритма. Красным цветом обозначена нулевая кривизна



а



б

Рис. 9. Геометрия skoba: а – красной линией выделены характерные ребра геометрии skoba, б – распределение значений кривизны, полученных в результате работы алгоритма Cohen–Steiner D., Morvan J.-M. Желтым цветом обозначена нулевая кривизна

Модификации алгоритма Cohen–Steiner D., Morvan J.-M.

В ходе исследования алгоритма Cohen–Steiner D., Morvan J.-M были опробованы несколько его модификаций. Все модификации затронули вычисление кривизны в граничных точках.

Первая модификация. На первом шаге алгоритма в область, на которой будет вычисляться кривизна, добавляется только первое кольцо треугольников, которое, в свою очередь, разбито характерными кривыми на несколько частей.

На каждой части кольца кривизна вычисляется по алгоритму Cohen–Steiner D., Morvan J.-M., начиная со второго шага алгоритма. Итоговое значение кривизны в граничной вершине находится как максимальное из значений, найденных на отдельных частях первого кольца.

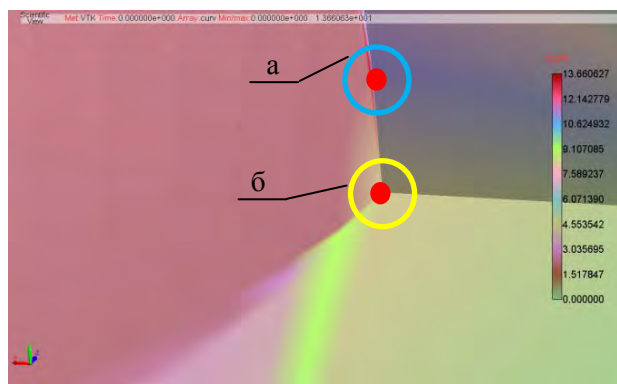


Рис. 10. Распределение значений кривизны, полученных в результате работы первой модификации алгоритма Steiner D: а – вершина, в которой достигается максимальная кривизна на всей геометрии, б – вершина, в которой значение кривизны превышает значение кривизны на сфере в 2 раза

Значения кривизны зависят от площади первого кольца треугольников.

Вторая модификация. Изменен этап поиска области. Из трех ограничений для поиска треугольников остается два:

- все вершины треугольника лежат на расстоянии меньше заданного от исследуемой точки;
- общее количество треугольников в области ограничено заданным значением.

Вся область разбивается характерными кривыми, полученными с предыдущего этапа работы поверхностного генератора.

Главным недостатком стала зависимость этапа вычисления кривизны от результатов предыдущих этапов работы поверхностного генератора.

Третья модификация. Третья модификация отличается от второй этапом получения характерных кривых. Характерные кривые выделяются по углу в 60 градусов между треугольниками. Угол был получен эмпирически.

На рис. 11 приведено распределение значений кривизны для геометрии cubit, полученных в результате работы второй и третьей модификации. На рис. 12 – фрагмент этого распределения в месте стыка сферической поверхности и 2-х плоскостей.

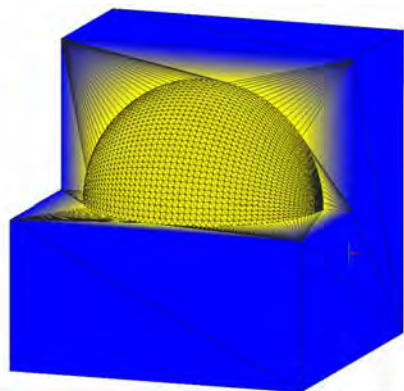


Рис. 11. Распределение значений кривизн для геометрии cubit, полученных в результате работы второй и третьей модификации

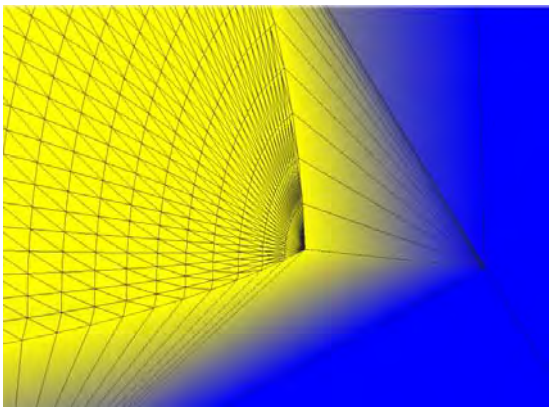


Рис. 12. Фрагмент распределения рис .11

Практическое применение алгоритма Cohen–Steiner D., Morvan J.-M. И его модификаций

При применении первой модификации алгоритма на границе сетки наблюдались скачки значений кривизны из-за характерных особенностей геометрий (длинных вытянутых треугольников или треугольников очень малой площади). Также значение кривизны зависит от размера области, на которой вычисляется кривизна.

Вторая модификация алгоритма зависит от предыдущего этапа работы генератора – разбиения геометрии на патчи. Так как генератор находится на этапе разработки, то после предыдущего этапа данные не всегда корректны, из-за чего возможны ситуации, когда этап вычисления кривизн в вершинах сетки обрабатывает некорректно.

Третья модификация используется в поверхностном генераторе SurfaceRemeshT3.

Заключение

В генераторе поверхностных сеток ЛОГОС.Пре-Пост был добавлен этап вычисления метрик по значениям кривизны, что значительно повысило качество итоговых сеток. Для вычисления кривизны используется третья модификация алгоритма Cohen–Steiner D., Morvan J.-M., так как по сравнению с другими она показала наилучший результат, несмотря на недостатки исходной сетки, такие как значительная разница площадей соседних треугольников, а также низкое качество исходных треугольников.

Результаты работы поверхностного генератора приведены на рис.13–16. Результирующие сетки были получены для одной исходной замкнутой геометрии здания, окруженного габаритной коробкой. Исходная геометрия разбита на два патча: здание и габаритная коробка. На каждом патче заданы одинаковые пользовательские размеры. Итоговые сетки построены в двух версиях поверхностного генератора: в первой версии (рис. 13 и 15) метрика вычисляется по размерам, которые задан пользователь, во второй версии (рис. 14 и 16) метрика вычисляется с учетом кривизны исходной поверхности.

Видно, что на рис.13 и 15 сетка в искривленных местах грубая по сравнению с сетками на рис. 14 и 16.

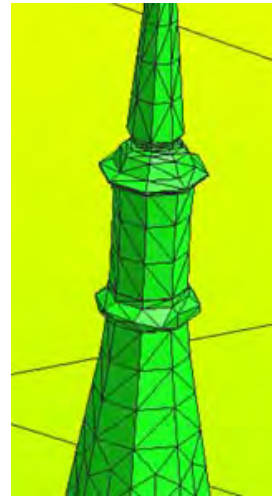


Рис. 13. Поверхностная сетка крыши здания, построенная с метриками, вычисленными по пользовательским данным



Рис. 14. Поверхностная сетка крыши здания, построенная с метриками, вычисленными с учетом искривления поверхности

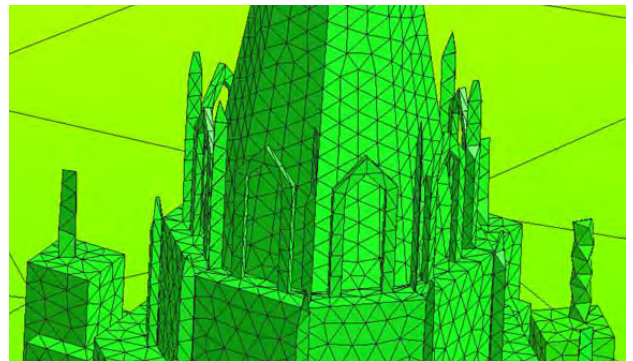


Рис. 15. Фрагмент поверхностной сетки здания, построенной с метриками, вычисленными по пользовательским данным

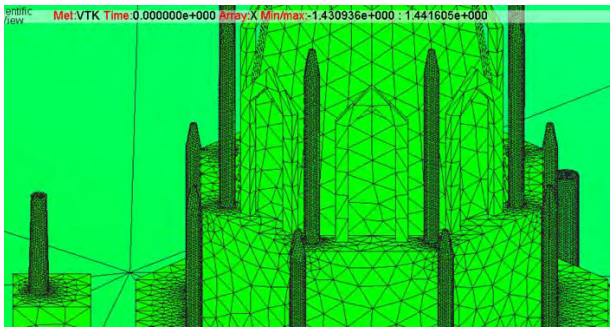


Рис. 16. Фрагмент поверхностной сетки здания, построенной с метриками, вычисленными с учетом искривления поверхности

1. Kim S. J., Kim C.-H., Levin D. Surface simplification using discrete curvature norm // The Third Israel-Korea Binational Conference on Geometric Modeling and Computer Graphics, Seoul, Korea, October 2001.

2. Krsek P., Lukarcs C., Martin R.R. Algorithms for computing curvatures from range data // The Mathematics of Surfaces VIII, Information Geometers, A. Ball et al. (Eds.), 1998, pp. 1–16.

3. Taubin G. Estimating the tensor of curvature of a surface from a polyhedral approximation // The Fifth International Conference on Computer Vision, 1995. P. 902–907.

4. Watanabe K., Belyaev A.G. Detection of salient curvature features on polygonal surfaces // Eurographics 2001. Vol. 20, No 3, 2001.

5. Cohen-Steiner D., Morvan J.-M. Restricted Delaunay Triangulations and Normal Cycle // SoCG'03 California, SD: 2003.