
ДИСКРЕТНАЯ МАТЕМАТИКА И МАТЕМАТИЧЕСКАЯ КИБЕРНЕТИКА

DISCRETE MATHEMATICS AND MATHEMATICAL CYBERNETICS

УДК 004.925.83

ГЛОБАЛЬНАЯ БАЛАНСИРОВКА ТРИАНГУЛЯЦИОННОЙ СЕТИ

Д. Д. ВАСИЛЬКОВ¹⁾

¹⁾Белорусский государственный университет, пр. Независимости, 4, 220030, г. Минск, Беларусь

Предложен новый алгоритм балансировки триангуляционной сети, содержащей точки Штейнера, основанный на методе наименьших квадратов. Он минимизирует среднеквадратичное отклонение косинусов углов триангуляции от оптимального значения 0,5. Алгоритм не имеет ограничений, поэтому может быть применен к любым триангуляциям, полученным алгоритмами сгущения триангуляционной сети, например алгоритмами Рупперта или Эртена и Унгора, при этом он не увеличивает число точек, а также не нарушает реберных связей. Проведенные эксперименты показали, что предлагаемый алгоритм существенно повышает число углов в диапазоне от 50 до 70° и не приводит к появлению треугольников с существенно меньшими минимальными углами. Алгоритм может быть эффективно реализован с использованием специализированных программных пакетов, предназначенных для быстрого решения разреженных систем линейных уравнений методом наименьших квадратов, например SuiteSparse. Благодаря этому алгоритм является простым в реализации.

Ключевые слова: триангуляция; генерация сети; балансировка триангуляции; точки Штейнера; топология триангуляционной сети; метод наименьших квадратов; погрешность интерполяции.

Образец цитирования:

Васильков Д. Д. Глобальная балансировка триангуляционной сети // Журн. Белорус. гос. ун-та. Математика. Информатика. 2018. № 1. С. 88–94.

For citation:

Vasilkov D. D. Global balancing of a triangular mesh. *J. Belarus. State Univ. Math. Inform.* 2018. No. 1. P. 88–94 (in Russ.).

Автор:

Денис Дмитриевич Васильков – ассистент кафедры дискретной математики и алгоритмики факультета прикладной математики и информатики.

Author:

Denis D. Vasilkov, assistant at the department of discrete mathematics and algorithms, faculty of applied mathematics and computer science.
molibos@gmail.com

GLOBAL BALANCING OF A TRIANGULAR MESH

D. D. VASILKOV^a

^a*Belarusian State University, 4 Niezaliežnasci Avenue, Minsk 220030, Belarus*

New algorithm for Steiner triangular mesh balancing is proposed. The algorithm is based on the least squares method and minimizes the standart deviation of triangulation angles cosines from the optimal value of 0.5. The algorithm has no limitations and therefore can be applied to any triangulations obtained by triangular mesh refinement algorithms, for example Ruppert or Erten and Üngör algorithms, without increasing the resulting number of points and without breaking the edge connections. Experiments indicate that the proposed algorithm significantly increases the number of angles in range from 50 to 70° and doesn't lead to create triangles with significantly smaller minimum angles. The algorithm can be effectively implemented using specialized software packages for quick solving sparse linear systems using the least-squares method, for example SuiteSparse. Therefore the algorithm is easy to implement.

Key words: triangulation; mesh generation; mesh refinement; Steiner points; triangular mesh topology; least squares method; interpolation error.

Рассмотрим задачу геометрического моделирования следующего вида. Дан набор точек $P = \{p_1, p_2, \dots, p_n\}$ на горизонтальной плоскости с известными значениями высот, требуется построить поверхность F , проходящую через эти точки и обладающую определенными свойствами. Наиболее распространенный подход к решению этой общей задачи состоит в следующем [1]: на множестве P строится триангуляция по критерию Делоне, затем над каждым треугольником вычисляется фрагмент поверхности таким образом, чтобы объединение этих фрагментов давало в результате поверхность с необходимыми свойствами. Часто эти свойства формулируются в терминах оптимизационных задач, таких как интегральная кривизна поверхности или максимальный градиент, которые требуется минимизировать. В компьютерной графике в качестве критерия используется визуальная гладкость поверхности.

Несмотря на ряд полезных свойств, триангуляция Делоне, построенная на исходных точках, не всегда может использоваться для моделирования оптимальных поверхностей. Известно [2], что при ограниченной кривизне моделируемой поверхности погрешность линейной интерполяции определяется размером треугольника, тогда как погрешность градиента существенно зависит от его формы: углы, близкие к 60°, дают наименьшую погрешность, а углы, близкие к 0 или 180°, – наибольшую. На рис. 1, *a*, изображена триангуляция Делоне, содержащая множество «плохих» (т. е. остроугольных и тупоугольных) треугольников, что делает невозможным ее использование для построения оптимальных поверхностей.

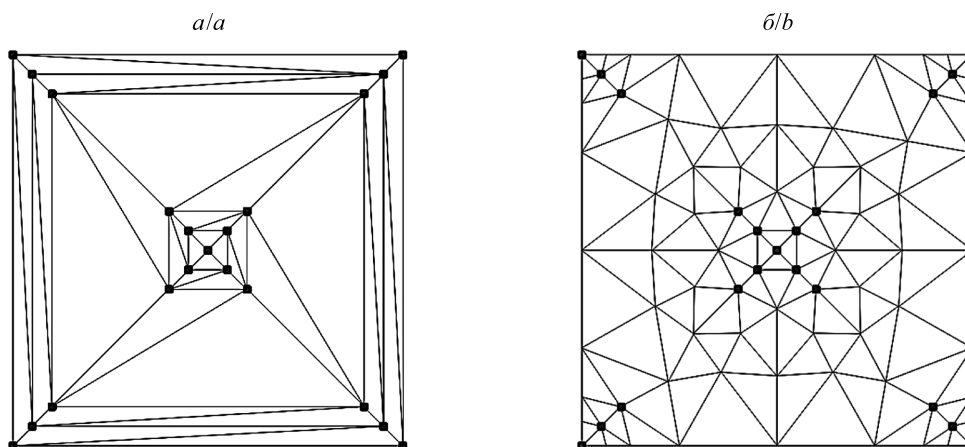


Рис. 1. Исходная триангуляция Делоне (*a*)
и триангуляция Делоне, содержащая точки Штейнера (*b*)

Fig. 1. Original Delaunay triangulation (*a*)
and Delaunay triangulation containing Steiner points (*b*)

Для оптимизации формы поверхности используется процедура сгущения (refinement) сети, которая состоит в добавлении в триангуляцию точек Штейнера (см. рис. 1, б). Предполагается, что в дальнейшем высоты в точках Штейнера должны быть каким-то образом аппроксимированы с учетом свойств, которые должна иметь поверхность. В работах [3; 4] предложены алгоритмы сгущения сети, которые последовательно вставляют точки Штейнера в определенные позиции и производят затем локальное перестроение триангуляции по критерию Делоне, разрушая «плохие» треугольники. Вставка повторяется до тех пор, пока все «плохие» треугольники не будут разрушены. Подобные процедуры, как правило, приводят к появлению большого числа новых точек, что делает трудоемким и неустойчивым процесс аппроксимации высот.

Альтернативный подход называется *балансировкой* сети и заключается в нахождении оптимального положения точек Штейнера уже после их вставки. В [5] предложен алгоритм, добавляющий точки Штейнера и одновременно оптимизирующий их положение в локальной области, что позволяет при вставке достаточно большого числа точек Штейнера построить триангуляцию с минимальным углом, не превосходящим 42° .

Предлагается использовать тот факт, что балансировку можно начинать в любой момент, например сразу после разрушения всех «плохих» треугольников в исходной триангуляции. Таким образом, при рассмотрении числа точек Штейнера в качестве входного параметра требуется найти такое местоположение этих точек, которое минимизирует среднеквадратическое отклонение углов треугольников от оптимального значения 60° . Такой подход позволяет существенно сократить число точек Штейнера и в целом упростить процесс построения поверхности.

В отличие от алгоритмов, основанных на локальных и итерационных схемах, в которых качество балансировки оценивается по значению минимального угла треугольников, в настоящей работе используются статистические оценки метода наименьших квадратов, которые являются оптимальными в классе линейных несмещенных оценок.

Глобальная балансировка

Пусть $S = \{s_1, s_2, \dots, s_m\}$ – множество точек Штейнера триангуляции $T = \{V, E, F\}$, где $V = P \cup S$, E – множество ребер триангуляции; F – множество треугольников. Обозначим через $X(T)$ множество всех углов триангуляции T . Целью алгоритма является минимизация среднеквадратического отклонения углов триангуляции от оптимального значения 60° . Для упрощения заменим углы на их косинусы и рассмотрим функционал

$$A(T) = \sum_{\alpha \in X(T)} \left(\cos \alpha - \cos \frac{\pi}{3} \right)^2. \quad (1)$$

Сведем минимизацию функционала (1) к решению системы линейных уравнений с помощью метода наименьших квадратов. Запишем косинус угла α между вершинами p_1, p_0 и p_2 через скалярное произведение

$$\cos \alpha = \frac{(p_1 - p_0, p_2 - p_0)}{l_{01} l_{02}},$$

где $l_{ij} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$ – длина ребра (i, j) .

Представим векторы координат точек Штейнера в виде $\tilde{p} = p + \Delta p$, где $p = (x, y)$ – вектор координат, полученный в результате работы алгоритма сгущения; $\Delta p = (\Delta x, \Delta y)$ – неизвестная поправка. Для вершин P исходной триангуляции поправки равны нулю. Тогда углу $\angle p_1 p_0 p_2$ соответствует уравнение относительно поправок

$$\frac{p_1 + \Delta p_1 - p_0 - \Delta p_0, p_2 + \Delta p_2 - p_0 - \Delta p_0}{\tilde{l}_{01} \tilde{l}_{02}} = \cos \frac{\pi}{3}, \quad (2)$$

где $\tilde{l}_{ij} = \sqrt{(x_i + \Delta x_i - x_j - \Delta x_j)^2 + (y_i + \Delta y_i - y_j - \Delta y_j)^2}$.

Уравнение (2) не является линейным, поэтому применим линеаризацию: отбросим слагаемые, содержащие квадраты поправок, и проигнорируем зависимость длин ребер от поправок. Перенесем в правую часть свободные члены и получим

$$(a_0, \Delta p_0) + (a_1, \Delta p_1) + (a_2, \Delta p_2) = \cos \frac{\pi}{3} l_{01} l_{02} - (p_1 - p_0, p_2 - p_0), \quad (3)$$

где $a_0 = 2p_0 - p_1 - p_2$; $a_1 = p_2 - p_0$; $a_2 = p_1 - p_0$.

Таким образом, получаем линейную систему относительно поправок Δp , состоящую из уравнений (3). Подобные системы решаются методом наименьших квадратов за несколько итераций. После каждой итерации в точки вносятся найденные поправки и уточняются коэффициенты уравнений (3).

Однако вычислительные эксперименты показали, что при неравномерном распределении точек метод ведет себя неустойчиво. После нескольких итераций разрушается триангуляционная структура, точки Штейнера выходят за пределы своего окружения, происходят пересечения ребер. Причиной этого является внесение больших поправок в вершины больших треугольников, имеющих общие вершины с треугольниками значительно меньшего размера. Иными словами, для приближения угла большого треугольника к оптимальному значению 60° требуется внесение большей поправки, чем для треугольника меньшего размера. Во избежание этого каждому уравнению необходимо сопоставить некоторый вес, который уравнивал бы невязки в уравнениях для подобных треугольников.

Рассмотрим два подобных треугольника $p_1 p_0 p_2$ и $p_3 p_0 p_4$ с коэффициентом подобия s и общей точкой p_0 (рис. 2). Не теряя общности, предположим, что точка p_0 совпадает с началом координат. Тогда

$$p_3 = sR p_1,$$

$$p_4 = sR p_2,$$

где R – матрица поворота, совмещающая соответствующие стороны треугольников. Выпишем оба уравнения вида (3) для вершины p_0 :

$$(\Delta p_0, (-p_1 - p_2)) = \cos \frac{\pi}{3} |p_1| |p_2| - (p_1, p_2), \quad (4)$$

$$(\Delta p_0, sR(-p_1 - p_2)) = \cos \frac{\pi}{3} |sR p_1| |sR p_2| - (sR p_1, sR p_2). \quad (5)$$

Ортогональная матрица R не изменяет длины векторов p_1 и p_2 , поэтому уравнение (5) можно переписать в следующем виде:

$$(\Delta p_0, (-p_3 - p_4)) = s^2 \left(\cos \frac{\pi}{3} |p_1| |p_2| - (p_1, p_2) \right). \quad (6)$$

Таким образом, чтобы сбалансировать невязки уравнений (4) и (6), необходимо разделить уравнение (6) на s^2 – квадрат коэффициента подобия или разделить оба уравнения на площади соответствующих треугольников. С учетом этих весов функционал (1) следует заменить на функционал вида

$$A'(T) = \sum_{\alpha \in X} \frac{1}{S_\alpha} \left(\cos \alpha - \cos \frac{\pi}{3} \right)^2,$$

где S_α – площадь треугольника, содержащего угол α .

Вычислительные эксперименты показали, что в обновленной системе уравнений приемлемый уровень сходимости метода наименьших квадратов достигается после трех-четырёх итераций. Система является сильно разреженной, поэтому решение можно найти при помощи эффективных методов, например LSQR [6]. Использование специализированных пакетов, таких как SuiteSparse [7], позволяет выполнить балансировку сети, содержащей 10 000 точек Штейнера, за одну-две секунды.

Такое увеличение быстродействия алгоритма решения системы линейных уравнений достигается за счет возможности распараллеливания вычислений при помощи многопоточной реализации. Также указанная библиотека позволяет производить эффективную передачу вычислительной нагрузки на современные видеокарты. Предлагаемый метод балансировки является простым в программной реализации и не требует использования сложных структур данных, однако в качестве минуса данного подхода можно отметить дополнительное использование памяти для хранения коэффициентов уравнений системы.

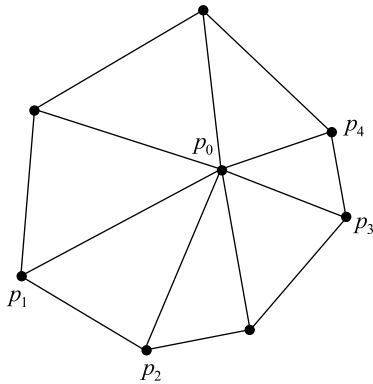


Рис. 2. Подобные треугольники с общей вершиной p_0
Fig. 2. Similar triangles with common vertex p_0

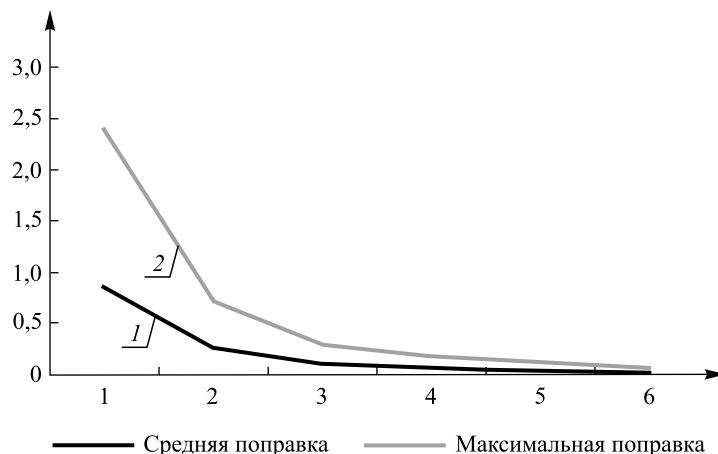


Рис. 3. График зависимости величины поправки от числа итераций на триангуляции из 500 точек в квадрате 100×100 : величины средней поправки (1) и максимальной поправки (2) после k -й итерации

Fig. 3. Graph of the dependence of the correction value on the number of iterations on triangulation of 500 points in a square of 100 per 100: the average (1) and maximum (2) corrections after the k -th iteration

Плюсом данного подхода является сохранение связей в сети: корректируются только координаты вершин, в то время как итеративные локальные подходы изменяют валентности вершин и реберные связи. Предлагаемый метод может быть использован в качестве дополнения к любому алгоритму вставки точек Штейнера без всяких ограничений.

Эксперименты и анализ результатов

Для характеристики результата работы алгоритма предлагается использовать гистограмму углов сети. В отличие от оценки минимального угла, который предлагается в работах [3–5], такой способ является статистическим и отражает качество углов триангуляции в среднем.

Результат работы алгоритма на триангуляции, полученной с помощью алгоритма Рупперта, представлен на рис. 4. Для наглядности был взят набор из малого числа вершин.

Сравнение распределения углов триангуляции до балансировки и после нее на случайных наборах из 300 и 900 точек соответственно приведено на рис. 5 и 6. На гистограммах видно, что распределение углов после балансировки близко к нормальному.

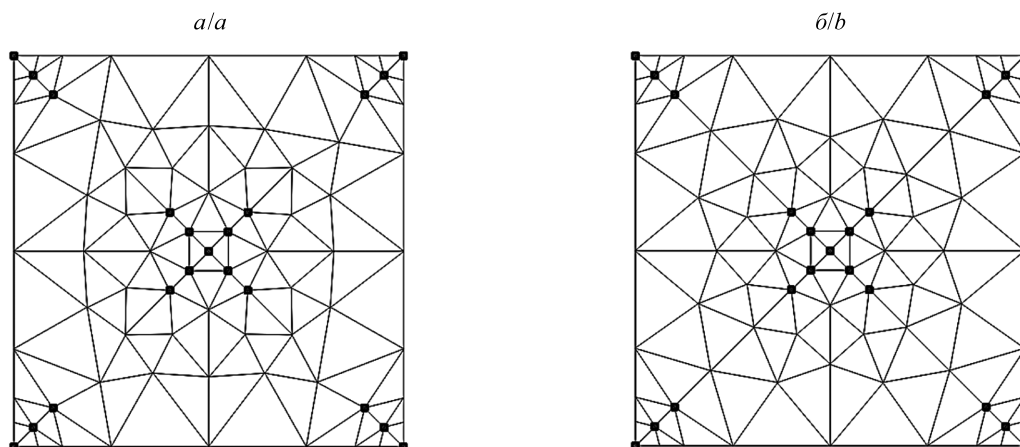


Рис. 4. Триангуляция, полученная алгоритмом Рупперта (а), и триангуляция после балансировки (б)

Fig. 4. Triangulation obtained by the Ruppert algorithm (a), and triangulation obtained after balancing (b)

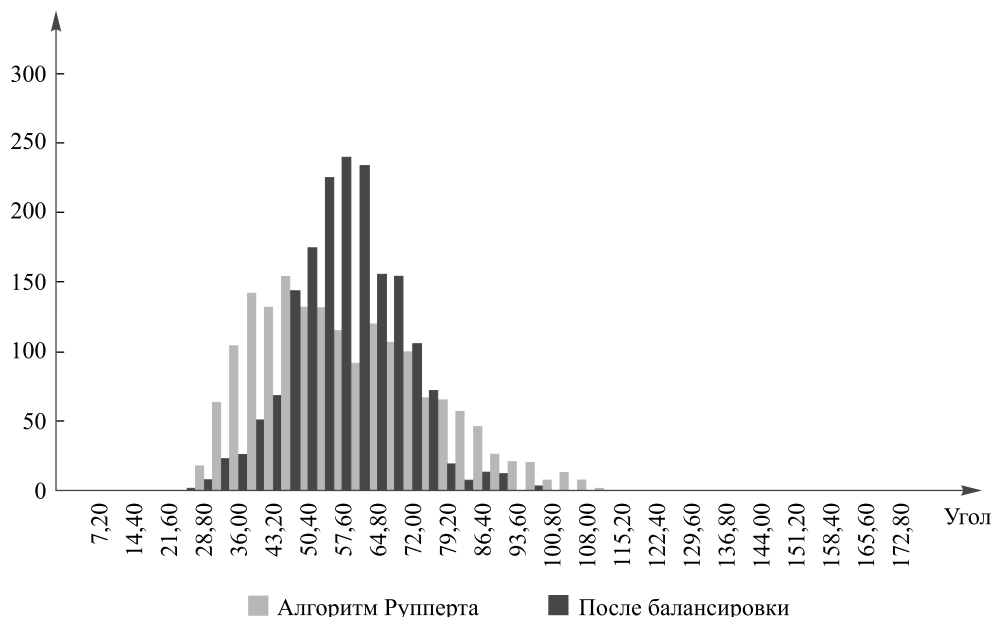


Рис. 5. Гистограмма углов на триангуляции, содержащей 300 точек: углы триангуляции до балансировки на наборе из 300 точек (светлый цвет) и после нее (темный цвет)

Fig. 5. Angle histogram of trinagulation containing 300 points: angles of triangulation before balancing on a set of 300 points (light) and after it (dark)

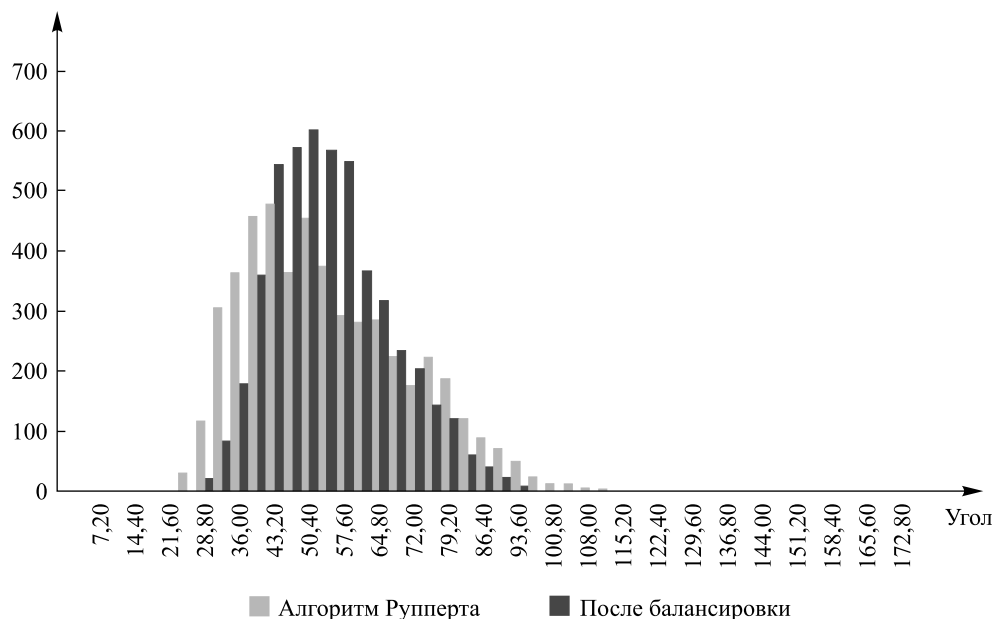


Рис. 6. Гистограмма углов на триангуляции, содержащей 900 точек: углы триангуляции до балансировки на наборе из 900 точек (светлый цвет) и после нее (темный цвет)

Fig. 6. Angle histogram of trinagulation containing 900 points: angles of triangulation before balancing on a set of 900 points (light) and after it (dark)

В экспериментах, проведенных на различных данных, число углов в диапазоне от 50 до 70° в триангуляции после балансировки составляет около $75\text{--}80\%$, в то время как в оригинальных триангуляциях доля таких углов составляла менее 50% . Выявлено также, что алгоритм не приводит к уменьшению минимального угла триангуляции на величины больше 5° .

Библиографические ссылки

1. Farin G. Curves and surfaces for CAGD. 4th ed. San Diego : Acad. Press, 1997.
2. Shewchuk J. R. What is a good linear finite element? // 11th International Meshing Roundtable (New York, 15–18 Sept., 2002). New York : Ithaca, 2002. P. 115–126.
3. Paul Chew L. Guaranteed-quality mesh generation for curved surfaces // Proceedings of the Ninth Annual Symposium on Computational Geometry (San Diego, 18–21 May, 1993). New York : ACM, 1993. P. 274–280.
4. Ruppert J. A Delaunay refinement algorithm for quality 2-dimensional mesh generation // J. Algorithms. 1995. Vol. 18, issue 3. P. 548–585. DOI: 10.1006/jagm.1995.1021.
5. Erten H., Üngör A. Triangulations with locally optimal steiner points // Eurographics Symposium on Geometry Processing (Barcelona, 4–6 July, 2007). Barcelona, 2007. P. 1–10.
6. Paige C. C., Saunders M. A. LSQR: An algorithm for sparse linear equations and sparse least squares // ACM Trans. Math. Soft. 1982. Vol. 8, issue 1. P. 43–71. DOI: 10.1145/355984.355989.
7. Rennich S., Stosic D., Davis T. A. Accelerating sparse cholesky factorization on GPUs // Architectures and Algorithms : IA³ Seventh Workshop on Irregul. Appl. (Denver, 13 Novemb., 2017). New Orleans, 2014.

References

1. Farin G. Curves and surfaces for CAGD. 4th ed. San Diego : Acad. Press, 1997.
2. Shewchuk J. R. What is a good linear finite element? In: 11th International Meshing Roundtable (New York, 15–18 Sept., 2002). New York : Ithaca, 2002. P. 115–126.
3. Paul Chew L. Guaranteed-quality mesh generation for curved surfaces. In: *Proceedings of the Ninth Annual Symposium on Computational Geometry* (San Diego, 18–21 May, 1993). New York : ACM, 1993. P. 274–280.
4. Ruppert J. A Delaunay refinement algorithm for quality 2-dimensional mesh generation. *J. Algorithms*. 1995. Vol. 18, issue 3. P. 548–585. DOI: 10.1006/jagm.1995.1021.
5. Erten H., Üngör A. Triangulations with locally optimal steiner points. In: *Eurographics Symposium on Geometry Processing* (Barcelona, 4–6 July, 2007). Barcelona, 2007. P. 1–10.
6. Paige C. C., Saunders M. A. LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Trans. Math. Soft.* 1982. Vol. 8, issue 1. P. 43–71. DOI: 10.1145/355984.355989.
7. Rennich S., Stosic D., Davis T. A. Accelerating sparse cholesky factorization on GPUs. *Architectures and Algorithms : IA³ Seventh Workshop on Irregul. Appl.* (Denver, 13 Novemb., 2017). New Orleans, 2014.

Статья поступила в редакцию 07.06.2017.
Received by editorial board 07.06.2017.