



doi 10.26089/NumMet.v23r207

УДК 514.8

Алгоритм численного моделирования потока поверхностной диффузии для триангулированных периодических поверхностей

Ю. Д. Ефременко

Новосибирский государственный университет,
Новосибирск, Российская Федерация

ORCID: 0000-0002-6978-1517, e-mail: i.efremenko@ngs.nsu.ru

Аннотация: В настоящей работе предложен алгоритм численного моделирования потока поверхностной диффузии для начальной периодической триангулированной поверхности. Разработаны алгоритмы перестройки триангуляции для обработки особенностей, возникающих при эволюции. Отдельно рассмотрены случаи особенности внутри куба, содержащего поверхность, на его гранях, ребрах и в углах. Работа алгоритма продемонстрирована рядом примеров.

Ключевые слова: вычислительные методы, поверхностная диффузия, триангуляция, изменения топологии, алгоритмы перестройки, средняя кривизна.

Благодарности: Работа выполнена при поддержке Математического Центра в Академгородке, соглашение с Министерством науки и высшего образования Российской Федерации № 075–15–2019–1675.

Для цитирования: Ефременко Ю.Д. Алгоритм численного моделирования потока поверхностной диффузии для триангулированных периодических поверхностей // Вычислительные методы и программирование. 2022. **23**, № 2. 95–116. doi 10.26089/NumMet.v23r207.

Algorithm of numerical modelling of the surface diffusion flow for periodic triangulated surfaces

Yury D. Efremenko

Novosibirsk State University, Novosibirsk, Russia

ORCID: 0000-0002-6978-1517, e-mail: i.efremenko@ngs.nsu.ru

Abstract: The algorithm of numerical modelling of the surface diffusion flow for periodic triangulated initial surface is proposed in the paper. To process the singularities that appear during computations the algorithms of retriangulation were developed. The cases of singularity inside of the cube containing the surface, on its face, edge and corner were considered. The work of the algorithm is illustrated with several examples.

Keywords: numerical methods, surface diffusion, triangulation, topology changes, retriangulation algorithms, mean curvature.

Acknowledgements: The work is supported by Mathematical Center in Akademgorodok, the agreement with Ministry of Science and Higher Education of the Russian Federation N 075–15–2019–1675.

For citation: Yu. D. Efremenko, “Algorithm of numerical modelling of the surface diffusion flow for periodic triangulated surfaces,” Numerical Methods and Programming, **23** (2), 95–116 (2022). doi 10.26089/NumMet.v23r207.



1. Введение. Поток поверхностной диффузии называется семейство $\{S(t) | t \in [0; T]\}$ гладких замкнутых поверхностей, вложенных в трехмерное пространство, удовлетворяющее системе:

$$\begin{cases} V_N = -d\Delta H, & t > 0; \\ S|_{t=0} = S_0, \end{cases} \quad (1)$$

где H — средняя кривизна поверхности $S(t)$, Δ — оператор Лапласа–Бельтрами, $V_N = \frac{\partial \mathbf{X}}{\partial t} \mathbf{N}$ — скорость в направлении нормали \mathbf{N} , $\mathbf{X} : S(t) \ni x \mapsto x \in \mathbb{R}^3$ — вложение поверхности, $d > 0$ — коэффициент диффузии ($\text{м}^2/\text{с}$). Само уравнение (1) называется уравнением поверхностной диффузии. Впервые данное уравнение было получено при рассмотрении деформации поверхностей под действием поверхностной диффузии, вызванной капиллярными силами [1]. Применялось оно в моделировании формы борозд на границе двух зерен нагретого поликристалла и формы катодов под действием напряжения. Модификации данного уравнения впоследствии не единожды появлялись в различных моделях, описывающих поведение границы раздела сред или поверхности вещества во время химической реакции. В первом из этих направлений стоит отметить следующий результат: при близости решения уравнения Кана–Хиллиарда к разделу сред граница раздела движется согласно уравнению поверхностной диффузии [2]. Поэтому данное уравнение применимо к описанию границы раздела сред. Во втором направлении из недавних работ стоит отметить [3], в которой была построена модификация уравнения (1), описывающая эволюцию поверхности сорбента. Кроме того, уравнение поверхностной диффузии хорошо описывает начальные стадии процессов спекания [1]. Были получены и математические результаты: в [4] было доказано, что для достаточно гладкой замкнутой начальной поверхности S_0 существует единственное решение уравнения (1) при $t \in [0; \varepsilon]$ для некоторого $\varepsilon > 0$. К сожалению, глобальное решение существует не всегда — поток поверхностной диффузии способен порождать разрывы и самопересечения поверхности [5]. В эти моменты поверхность, вообще говоря, перестает быть поверхностью в строгом математическом смысле, поэтому решение дальше не существует. Целью настоящей работы является нахождение численных решений уравнения (1), но не для замкнутой, а для периодической начальной поверхности. Разностная схема для решения уравнения (1) была построена и исследована автором в [6], данная же работа посвящена адаптации разностной схемы для случая периодической поверхности и разработке алгоритмов обработки особенностей, позволяющих продолжить решение за моменты возникновения особенностей.

Уравнение (1) — это нелинейное дифференциальное уравнение четвертого порядка с оператором Δ , зависящим от решения $S(t)$ в каждый момент времени. Поэтому аналитически искать точные решения этого уравнения для какой-либо начальной поверхности, отличной от стационарной точки ($H \equiv \text{const}$), не представляется возможным. В такой ситуации особо важными становятся численные методы решения. Для данного уравнения существует два подхода к его решению — метод “level-set” [7], когда S_0 представляется поверхностью уровня $\{\mathbf{x} \in \mathbb{R}^3 | f(\mathbf{x}) = 0\}$ некоторой функции $f : \mathbb{R}^3 \rightarrow \mathbb{R}$, или решение этого уравнения для триангулированной поверхности S_0 [6, 8]. Каждый из двух подходов имеет свои преимущества и недостатки перед другим. При использовании метода “level-set” можно вычислять все величины и операторы в привычных координатах пространства \mathbb{R}^3 , а также немного легче обрабатывать изменения топологии. Но в этом случае во время вычислений необходимо сохранять и изменять информацию не о поверхности, а о функции f , т.е. знать ее на всей области определения. Из-за необходимости дискретизации всего трехмерного пространства данный метод подразумевает существенные ограничения области по размеру либо по степени детализации. В случае же триангулированной поверхности достаточно знать только точки поверхности и связи между ними, но сложнее обстоит дело с вычислением величин и изменениями топологии. Однако для моделирования был выбран именно второй метод, так как при наличии алгоритмов обработки особенностей он гарантирует меньшие затраты памяти и высокую скорость работы. Также стоит отметить, что ни в одном из случаев пока не удалось доказать сходимости или устойчивости существующих разностных схем. Поэтому обращают внимание на выполнение свойств потока — сохранения объема $V(t)$, ограниченного поверхностью, и убывания площади поверхности $A(t)$:

$$\begin{aligned} \frac{d}{dt} V(t) &= \int_{S(t)} V_N ds = - \int_{S(t)} \Delta H ds = - \int_{S(t)} \text{div}(\text{grad } H) ds = 0; \\ \frac{d}{dt} A(t) &= - \int_{S(t)} V_N H ds = - \int_{S(t)} \Delta H H ds = - \int_{S(t)} |\nabla H|^2 ds \leq 0. \end{aligned} \quad (2)$$

Во многих задачах математического моделирования важно моделировать не просто эволюцию некоторой замкнутой поверхности, а поверхности с границей — например, сегмента некоторой периодической поверхности, моделирующей сорбент, катализатор или границу раздела сред. Поэтому далее будут рассматриваться методы решения данного уравнения для поверхностей с условиями симметрии на границе. Для таких поверхностей свойства (2) также будут выполняться.

2. Численное моделирование.

2.1. Представление поверхности. Программа для моделирования потока поверхностной диффузии была написана на языке C++ с использованием библиотек CGAL [9] и Eigen [10]. Для этого были созданы классы поверхности *Surface* и вершины *Vertex*. Поверхность S представляется массивом своих вершин Z и набором собственных методов: вычисление различных величин (средняя кривизна, нормаль, оператор Лапласа–Бельтрами, площадь, объем и т.д.), шаг разностной схемы, перестройки триангуляции, печать поверхности и т.п. В каждой же вершине $v_i \in Z$ хранятся:

- 1) радиус–вектор $X_i \in \mathbb{R}^3$ этой вершины;
- 2) единичный вектор нормали $N_i \in \mathbb{R}^3$;
- 3) значение H_i средней кривизны в данной вершине;
- 4) индикатор $IsBoundary \in \{true, false\}$, равный *true*, если вершина граничная, и *false* — иначе;
- 5) массив I указателей на вершины, соединенные с v_i ребрами. Причем порядок вершин в этом массиве соответствует направлению обхода этой вершины в заданной ориентации.

Такое представление данных оказывается удобным при вычислении вектора нормали, средней кривизны и разностного оператора Лапласа–Бельтрами, а также при численном решении уравнения (1), так как во всех методах вычисления используется обход вершины по ее соседям. К сожалению, при таком подходе, удобном в вычислениях, возникают трудности перестройки триангуляции. В случае локальных перестроек в разделах 3.3 и 3.4 эти трудности удалось преодолеть явно, но в случае глобальных перестроек из раздела 3.1.1 эффективнее оказалось переводить поверхность в другой тип данных, используемый библиотекой CGAL.

В каждом из рассмотренных ниже примеров начальной поверхностью является поверхность уровня $S = \{x \in \mathbb{R}^3 : f(x) = 0\}$ некоторой функции $f : \mathbb{R}^3 \rightarrow \mathbb{R}$. Начальная триангуляция строилась методами библиотеки CGAL пакета *3D Surface Mesh Generation* [11], причем для построенной поверхности задавалась ориентация — направление обхода вершин в каждом треугольнике. Зная ориентацию каждого треугольника, несложно восстановить направление обхода на множестве вершин $\{v_0, v_1, \dots, v_{m-1}\}$, соединенных ребрами с вершиной v . Таким образом, при построении триангуляции также задается ориентация поверхности S .

Для упрощения записи введем некоторые вспомогательные обозначения (рис. 1). Обозначим через e_i ребро vv_i , угол $\angle v_i vv_{(i+1) \bmod m}$ как α_i , углы при стороне $v_i v_{(i+1) \bmod m}$ в треугольнике $v_i vv_{(i+1) \bmod m}$ как β_i и γ_i соответственно. Множество вершин $\{v_i\}_{i=0}^{m-1}$, соединенных ребрами с данной вершиной v , назовем *звездой* вершины v и обозначим $St(v)$.

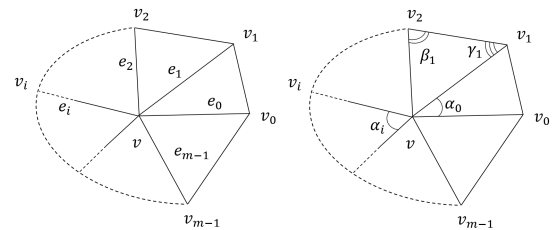


Рис. 1. Обозначения вершин в звезде

Fig. 1. Notation of vertices at the star

2.2. Разностная схема решения уравнения

поверхностной диффузии. Уравнение (1) высоко чувствительно к точности аппроксимации и тяжело поддается численному решению, поэтому построение устойчивой разностной схемы представляет определенную сложность. На данный момент существуют только две разностные схемы для его решения в случае триангулированной поверхности. В работе [8] была предложена полностью неявная схема для случая триангулированной поверхности в \mathbb{R}^3 . На рассмотренных примерах она является устойчивой, однако какое-либо исследование устойчивости представлено не было. Хотя эта схема и позволяет проводить вычисления с большим, по сравнению с явной схемой, шагом, на его величину все равно накладывается ограничение сверху для обратимости матрицы в схеме. Отметим, что в этой работе построение схемы представляет определенную сложность и не является полностью прозрачным. Кроме того, все изменения топологии проводились вручную. Вторая разностная схема предложена автором в работе [6], естественным

способом.

продолжением которой является настоящая статья. Схема является полуявной и задается соотношением

$$\mathbf{N}^n \mathbf{X}^{n+1} = \mathbf{N}^n \mathbf{X}^n - \tau \left(I + \frac{\tau}{2} \Delta_n^2 \right)^{-1} \Delta_n H^n. \quad (3)$$

Здесь τ — шаг по времени, \mathbf{X}^n — координаты вершины на n -м шаге, \mathbf{N}^n — единичный вектор нормали, H^n — значение средней кривизны, Δ_n — разностный оператор Лапласа–Бельтрами. После вычисления $\mathbf{N}^n \mathbf{X}^{n+1}$ поверхность на шаге $n + 1$ находится в следующем виде:

$$\mathbf{X}^{n+1} = \mathbf{X}^n + \tau \mathbf{N}^n V_N = \mathbf{X}^n + \mathbf{N}^n (\mathbf{N}^n \mathbf{X}^{n+1} - \mathbf{N}^n \mathbf{X}^n).$$

Применение повышающего гладкость оператора $\left(I + \frac{\tau}{2} \Delta_n^2 \right)^{-1}$ к явной схеме позволяет добиться устойчивости для больших шагов по времени. Отметим, что оператор $\left(I + \frac{\tau}{2} \Delta_n^2 \right)$ будет обратим для любого $\tau > 0$ при достаточно регулярной триангуляции, поэтому схема построена корректно. Аргументация этого факта, так же как и исследование данной схемы, подробно изложена в [6]. Как будет показано ниже, схема достаточно проста для реализации.

Чтобы применить схему (3), необходимо сначала в каждой вершине вычислить вектор нормали, среднюю кривизну, а затем вычислить матрицу разностного оператора Лапласа–Бельтрами. Для нахождения нормали в каждой вершине $v \in S$ использовался метод

$$\mathbf{N}'_v = \sum_{i=0}^{m-1} \frac{[\mathbf{e}_i \times \mathbf{e}_{(i+1) \bmod m}]}{\|\mathbf{e}_i\|^2 \|\mathbf{e}_{(i+1) \bmod m}\|^2}, \quad \mathbf{N}_v = \frac{\mathbf{N}'_v}{\|\mathbf{N}'_v\|},$$

предложенный в работе [12], так как он показал самые точные результаты во время численных экспериментов. Как уже упоминалось, изначально поверхность ориентирована, т.е. задано направление обхода вершины по ее соседям, поэтому нормали, вычисленные данным методом, оказываются согласованными по направлению. Далее, используя векторы нормали, можно найти в каждой вершине значение средней кривизны H .

Для этой цели был использован метод, основанный на формуле

$$\int_0^{2\pi} \kappa_n(\varphi) d\varphi = 2\pi H.$$

Здесь $\kappa_n(\varphi)$ — нормальная кривизна в вершине v , зависящая от угла поворота φ нормальной плоскости вокруг \mathbf{N}_v . Данный интеграл в вершине v можно находить приближенно в виде суммы по всем ее соседним вершинам:

$$H = \frac{1}{2\pi} \sum_{i=0}^{m-1} \kappa_n^i \left(\frac{\alpha_{(i-1) \bmod m} + \alpha_i}{2} \right).$$

Здесь $\kappa_n^i = \frac{2\mathbf{N}_v \cdot \mathbf{e}_i}{\|\mathbf{e}_i\|^2}$ — приближенное значение нормальной кривизны в направлении ребра \mathbf{e}_i , а $d\varphi(v_i) \approx \frac{1}{2} (\alpha_{(i-1) \bmod m} + \alpha_i)$. Этот метод был предложен в [13] и проанализирован в [14], где показал в среднем лучшие результаты по сравнению с другими известными методами.

На следующем шаге вычисляется матрица разностного оператора Лапласа–Бельтрами модифицированным методом котангенсов [15]:

$$\Delta f(v) = \frac{1}{2A} \sum_{i=0}^{m-1} \omega_i (f(v_i) - f(v)),$$

где

$$\omega_i = \begin{cases} \text{ctg } \beta_i + \text{ctg } \gamma_{(i-1) \bmod m}, & \text{для внутренних вершин;} \\ \text{ctg } \beta_i, & \text{для вершин на границе.} \end{cases}$$

Здесь β_i и $\gamma_{(i-1) \bmod m}$ — углы, противоположные ребру \mathbf{e}_i в звезде вершины v , A — площадь окрестности вершины v . Под окрестностью вершины подразумевается область Вороного — множество точек триангулированной поверхности, находящихся ближе к данной вершине, чем к остальным вершинам триангуляции.



При таком выборе окрестности достигается наилучшее приближение разностным оператором [16]. Легко видеть, что матрица, полученная этим методом, оказывается разреженной — в строке или столбце не более чем 5–8 ненулевых элементов (столько, сколько у вершины соседей). Поэтому и матрица $(I + \frac{\tau}{2}\Delta_n^2)$ будет разреженной, она достаточно быстро обращается методом QR-разложения для разреженных матриц библиотеки Eigen.

Также для проверки выполнения свойств потока необходимо вычислять площадь поверхности и объем, ею ограниченный. Площадь поверхности $A(S)$ вычисляется очевидным образом — это сумма площадей всех треугольников триангуляции. Вычисление объема $V(S)$ также может быть сведено к вычислению интеграла по поверхности. Для этого используется формула Стокса:

$$\int_{\Omega} d\omega = \int_{\partial\Omega} \omega.$$

Положим $\omega = xdy \wedge dz$, тогда

$$V(S) = \int_{\Omega} dx \wedge dy \wedge dz = \int_S xdy \wedge dz = \sum_{j=1}^M \int_{\Delta_j} xdy \wedge dz.$$

Здесь Δ_j — треугольники триангуляции. На произвольном треугольнике интеграл от формы ω вычисляется явно и зависит от координат его вершин. Поэтому весь интеграл можно посчитать одним циклом по треугольникам триангуляции. В случае же периодической поверхности, вообще говоря не являющейся замкнутой, необходимо проделать дополнительное преобразование. Если край поверхности является достаточно короткой замкнутой кривой¹, применяется функция *triangulate_hole* пакета *Polygon Mesh Processing* [17] библиотеки CGAL, триангулирующая дыры в поверхности. Тогда поверхность становится замкнутой и можно вычислить объем методом, изложенным выше. К сожалению, данный метод работает лишь в том случае, если край поверхности лежит на одной грани либо в окрестности ребра или угла куба. Тогда, добавив к объему несколько тетраэдров, его все так же можно посчитать точно.

Для того чтобы отслеживать появления поверхностей постоянной средней кривизны — стационарных точек потока, необходимо следить за поведением отклонения кривизны от ее среднего значения. Среднее значение \tilde{H} вычислялось таким образом:

$$\tilde{H} = \frac{1}{A(S)} \int_S H ds \approx \frac{1}{A(S)} \sum_{i=1}^N H_i A(U_i).$$

Отклонение кривизны от ее среднего значения вычислялось в L_1 -норме:

$$H_{\text{dev}} = \frac{1}{A(S)} \int_S |H - \tilde{H}| ds \approx \frac{1}{A(S)} \sum_{i=1}^N |H_i - \tilde{H}| A(U_i).$$

Для наглядности на графики выводилось не само H_{dev} , а процент, который оно составляет от $|\tilde{H}|$. В обоих интегралах $A(U_i)$ — площадь окрестности Вороного вершины v_i .

2.3. Модификации для периодической поверхности. Как можно было заметить, все вышеприведенные методы применимы лишь для замкнутой поверхности — любую вершину можно обойти по ее соседям. В случае же поверхности с границей для корректной работы методов необходима дополнительная информация о поверхности “за границей”. В настоящей работе этой информацией является условие симметрии поверхности относительно грани куба. В таком случае для граничной вершины v ее соседи, являющиеся внутренними вершинами, отражаются симметрично плоскости. Тогда звезда вершины v достраивается до полной, как у внутренней вершины, и все величины считаются методами, изложенными выше. Для вершин, находящихся в углах куба, такое отражение проделывается два раза. Геометрически этот процесс показан на рис. 2. Отметим отдельно, что разностный оператор Лапласа–Бельтрами оказался хорошо согласованным с этой моделью: его формула для граничных вершин в точности совпадает с формулой, полученной отражением.

¹Пример поверхности, для которой этот метод не работает, приведен в 4.4.

2.4. Алгоритм вычислений. В данном разделе коротко просуммируем весь процесс решения уравнения поверхностной диффузии. По начальной триангулированной поверхности S , шагу по времени $\tau > 0$ и количеству итераций n запускается процесс. На k -м шаге сначала вычисляются векторы нормали N^k для каждой вершины $v \in S$, затем по ним находятся значения средней кривизны H^k поверхности в вершинах. Затем вычисляется матрица разностного оператора Лапласа–Бельтрами Δ_k , решается система линейных уравнений для нахождения $N^k X^{k+1}$, находятся смещения вершин и поверхность перестраивается. Процесс описан для одной поверхности, в общем же случае рассчитывается эволюция массива поверхностей, который пополняется по мере разделения своих поверхностей на несколько компонент. Также на каждом шаге кроме описанных величин вычислялись площадь поверхности, объем, ею ограниченный (в случае, если это можно явно проделать), и отклонение кривизны от среднего значения.

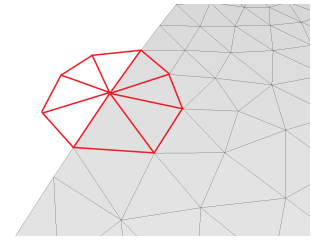


Рис. 2. Вычисление величин в граничных вершинах

Fig. 2. Calculation of the values at boundary vertices

3. Обработка особенностей. Во время эволюции поверхности вершины по-разному меняют свое положение в пространстве, поэтому возможны ситуации, когда треугольники триангуляции растягиваются или сжимаются. Из-за наличия котангенсов в дискретном операторе Лапласа–Бельтрами появление таких треугольников значительно влияет на устойчивость вычислений. Поэтому после одного или нескольких шагов разностной схемы необходимо перестраивать триангуляцию так, чтобы она становилась более регулярной. Это делается с помощью функции *isotropic_remeshing* пакета *Polygon Mesh Processing* библиотеки CGAL. Она перестраивает триангуляцию, делая ее регулярной с длиной ребра, близкой к заданному значению l . Граничные ребра триангуляции остаются лежащими в заданных плоскостях, поэтому условие симметрии не нарушается. Исключением могут быть только вершины, попавшие на ребра куба, о них будет сказано в разделе 4.3 на примере конкретной поверхности.

Кроме того, в [5] было показано, что во время эволюции поверхности возможно появление особенностей — самопересечений или разрывов. В этот момент с геометрической точки зрения поверхность перестает быть поверхностью, классическое решение дальше не существует. Но с физической точки зрения процесс продолжается, необходимо только разделить или склеить поверхность в нужном месте. При моделировании эволюции достаточно сложной поверхности особенности неизбежно возникают, поэтому необходим алгоритм, способный отслеживать их появление и вовремя обрабатывать.

Ниже будут описаны различные особенности и способы их обработки. К сожалению, метода, универсального для всех типов особенностей, не существует, поэтому во время вычислений могут применяться различные методы. Основная цель — смоделировать эволюцию сегмента периодической поверхности в кубе, поэтому возникают особенности как внутри куба, так и на его гранях. Но на практике после одного шага разностной схемы редко появляются особенности различных типов, поэтому, как правило, применяется только один метод.

3.1. Особенности внутри куба.

§3.1.1. Самопересечение поверхности. Самый частый тип особенности, возникающий при вычислениях, — перешеек, который постепенно становится все тоньше, пока в самом узком его месте диаметр не становится нулевым. Тогда происходит разрыв поверхности в точке (рис. 3). Данный тип особенности появляется внутри куба, поэтому на ее вид, геометрию и прохождение никаких ограничений заранее не накладывается (в отличие от особенностей на гранях). Поэтому и алгоритм ее обработки, с одной стороны, не обладает гарантиями — нельзя доказать, что в случае возникновения особенности она точно будет

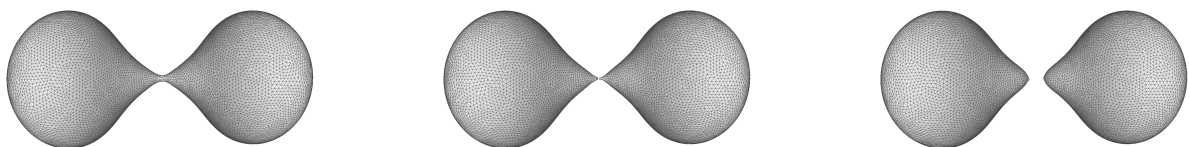


Рис. 3. Пример разрыва перешейка

Fig. 3. An example of a neck pinch-off



обработана, но с другой стороны, алгоритм способен обрабатывать большой спектр одностипных особенностей. В настоящее время существуют работы, посвященные обработке самопересечений поверхности (см., например, [18]), однако предложенные там алгоритмы неприменимы в данном случае, так как они либо убирают самопересечения, подразумевая сохранение топологии поверхности, либо сложны для реализации в качестве вспомогательного шага. Поэтому возникает необходимость в разработке собственных методов.

В настоящей работе алгоритм применялся в том же виде, в котором он описан в работе [6], поэтому здесь он будет приведен кратко в виде псевдокода (алгоритм 1). На очередном шаге на вход подается массив независимо эволюционирующих поверхностей *Surfaces*. Каждая поверхность *S* из этого массива сначала перетриангулируется, а затем проверяется на самопересечение. Если самопересечений нет, то *S* остается неизменной, но если нашлись пересекающиеся треугольники, то запускается процесс обработки.

Алгоритм 1. Обработка особенности внутри куба

Algorithm 1. Processing of the singularity inside of cube

```

1: for S ∈ Surfaces do
2:   ReTriangulate(S, l, n)
3:   if S.SelfIntersects() then
4:     Triangles IntersectTriangles = S.SelfIntersects()
5:     S.Delete(IntersectTriangles)
6:     Points P = S.SplitIntoPoints()
7:     Points UnorientedPoints = P.Orient()
8:     P.Delete(UnorientedPoints)
9:     S = P.Triangulate()
10:    SurfaceArray Components = S.ConnectedComponents()
11:    if Components.Size() > 1 then
12:      Surfaces.Delete(S)
13:      Surfaces.Add(Components)
14:    end if
15:  end if
16: end for
    
```

Сначала все треугольники, по которым поверхность пересеклась, удаляются. Затем *S* разбивается на массив вершин *P*, который на следующем шаге ориентируется — в каждой вершине вычисляется вектор нормали с направлением, согласованным по всем вершинам. Те вершины *UnorientedPoints*, в которых не удалось выбрать направление нормали, удаляются, ведь именно в этих точках и произошло самопересечение.

После этого по оставшимся в массиве *P* вершинам строится триангуляция Делоне функцией *reconstruct_surface* пакета *Scale-Space Surface Reconstruction* [19] библиотеки CGAL. Так как на предыдущем шаге триангуляция делалась регулярной, а самопересечение происходит в окрестности некоторой одной точки, восстановленная триангуляция лишь локально отличается от исходной. На последнем шаге определяется количество компонент связности восстановленной поверхности *S* — находится массив компонент связности *Components*. Если его размер равен единице, то значит, перешеек не разорвал поверхность на несколько частей, тогда *S* остается в *Surfaces*, если же размер массива превосходит единицу, то разрыв перешейка разделил поверхность как минимум на две части. Тогда поверхность *S* убирается из *Surfaces* и заменяется массивом своих компонент связности. Результат работы этого алгоритма можно видеть на рис. 3 или рис. 4.

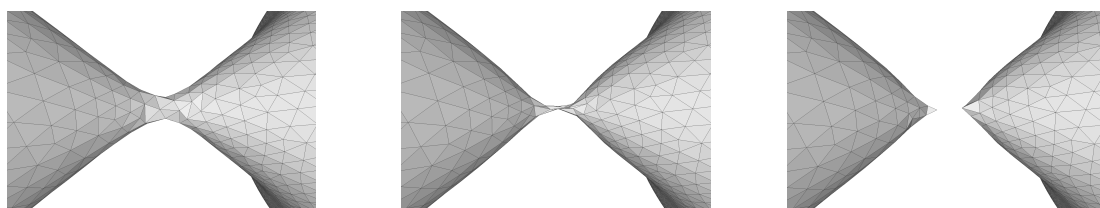


Рис. 4. Пример работы алгоритма обработки особенности внутри куба

Fig. 4. An example of singularity processing inside the cube

§ 3.1.2. *Пересечение двух поверхностей.* Более сложной особенностью, которая может появиться внутри куба, является пересечение двух различных поверхностей. В предыдущем случае эволюция поверхности постепенно приводила к возникновению особенности, поэтому для ее обработки требовалось лишь небольшое изменение триангуляции. В этом же случае триангуляция никак не подготовлена к появлению особенности, поэтому при аналогичном удалении пересекающихся треугольников восстановить триангуляцию уже не удастся. Соответственно, такая особенность требует иного подхода.

Пересечение двух поверхностей было описано и использовано в работе [6], поэтому здесь приведем краткое описание. Для обработки пересечения двух поверхностей используется функция *corefine* пакета *Polygon Mesh Processing*. Она позволяет подразбить триангуляции обеих поверхностей в окрестности их пересечения таким образом, что объединение поверхностей оказывается правильно триангулированным (но не обязательно будет поверхностью в строгом смысле). Результат такого подразбиения можно увидеть на рис. 5. Чтобы получить действительно слияние двух поверхностей, необходимо убрать их пересечение. Для этого используется функция *corefine_and_compute_intersection* того же пакета, которая находит замкнутую поверхность, являющуюся пересечением двух исходных. Затем три эти поверхности объединяются в одну, из которой удаляются все пересекающиеся треугольники. А это в точности поверхность их пересечения. Кратко процесс обработки представлен в виде алгоритма 2, а пример на рис. 5.

Данный алгоритм не претендует на оптимальность, так как здесь, несмотря на локальный характер особенности, используются глобальные свойства поверхности (наличие самопересечений, объединения поверхностей и т.п.). К сожалению, этим необходимо пожертвовать ради того, чтобы особенность обрабатывалась гарантированно. Более тонкий подход требовал бы анализа и учета большого числа частных случаев перестройки триангуляции, так как она заранее не подготовлена к появлению пересечения. Эта работа была бы слишком затратна по сравнению с частотой появления этой особенности — даже для достаточно сложных начальных поверхностей пересечение двух поверхностей происходило не более двух раз за время расчетов.

3.2. Особенности на грани куба. Цель настоящей работы — моделирование сегмента периодической поверхности, находящегося в кубе. Поэтому необходимо учесть не только особенности, возникающие внутри куба, но еще и те, которые возникают на гранях, ребрах или в углах куба. По своей сути это те же самые разрывы перешейков, но из-за их расположения к ним требуется иной подход. В этом случае у перешейка может не быть, например, его половины, с которой он может пересечься.

Сначала рассмотрим обработку разрыва перешейка, центр которого попадает на грань (рис. 6). Эта особенность обрабатывается алгоритмом, представленным в разделе 3.1.1, но сам тип обработки зависит

Алгоритм 2. Алгоритм обработки пересечения двух поверхностей

Algorithm 2. Processing of intersection of two surfaces

```

1: for  $S_1, S_2 \in \text{Surfaces}$  do
2:   Surface  $\Gamma = \text{Union}(S_1, S_2)$ 
3:   if  $\Gamma.\text{SelfIntersects}()$  then
4:     Surface Intersection =  $\text{Corefine}(S_1, S_2)$ 
5:     Surface  $\Sigma = \text{Union}(S_1, S_2, \text{Intersection})$ 
6:     Triangles IntersectTriangles =  $\Sigma.\text{SelfIntersects}()$ 
7:      $\Sigma.\text{Delete}(\text{IntersectTriangles})$ 
8:     Surfaces.Delete( $S_1$ )
9:     Surfaces.Delete( $S_2$ )
10:    Surfaces.Add( $\Sigma$ )
11:  end if
12: end for
    
```

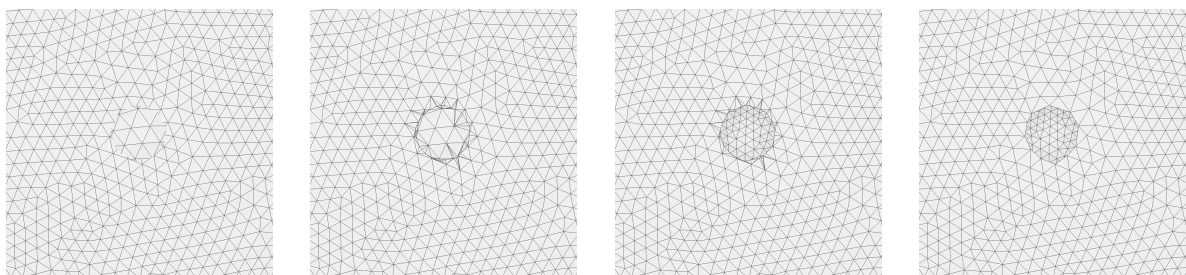


Рис. 5. Пример обработки пересечения двух различных поверхностей

Fig. 5. An example of two surfaces intersection processing

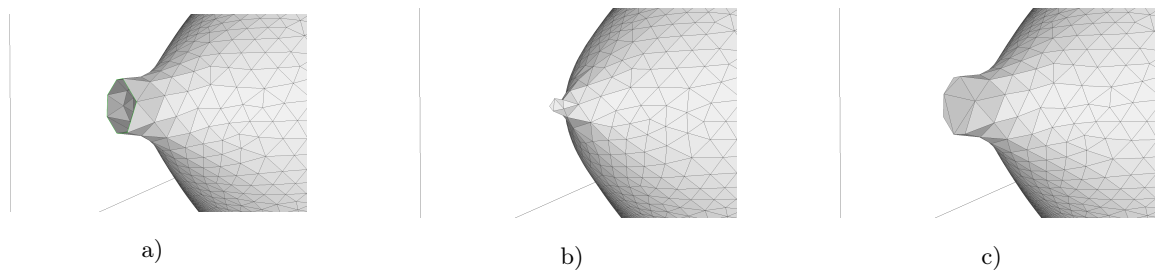


Рис. 6. Обработка разрыва перешейка на грани куба: а) пример сегмента поверхности внутри куба; б) обработка в случае самопересечения поверхности; в) обработка с использованием дополнительной триангуляции Делоне

Fig. 6. Processing of the neck pinch-off on a cube face: a) an example of surface segment inside the cube; b) processing by surface self-intersection; c) processing by additional Delaunay triangulation

от того, в какой момент применялся алгоритм. Здесь возможны две ситуации: либо в какой-то момент перешеек пересек сам себя, либо он стал настолько тонким, что любые две его точки на грани находятся на расстоянии, не превосходящем l . В первом случае все происходит аналогично разделу 3.1.1, во втором же случае самопересечения нет, но если разбить поверхность на точки и по ним построить триангуляцию Делоне, то в ней уже перешеек будет заклеен (рис. 6 в). Это произойдет потому, что расстояние между точками на границе перешейка не превышает среднюю длину ребра l , поэтому для алгоритма эти точки считаются соединенными ребрами.

Для этой цели можно использовать алгоритм *triangulate_hole* пакета *Polygon Mesh Processing*, триангулирующий дыры в поверхности, не превышающие определенный размер. Однако используется именно перестройка триангуляции, так как она находит применение в обработке сразу нескольких типов особенностей, что упрощает вычисления.

Следующие две особенности: перешеек, который плоскость пересекает вдоль (рис. 7 а), и выход поверхности за грань (рис. 7 б) можно обработать одним способом. Выходу поверхности за грань соответствует пересечение частей периодической поверхности, расположенных в разных кубах. Пересечению разных сторон перешейка в момент его разрыва соответствует выход части перешейка за грань. Так как

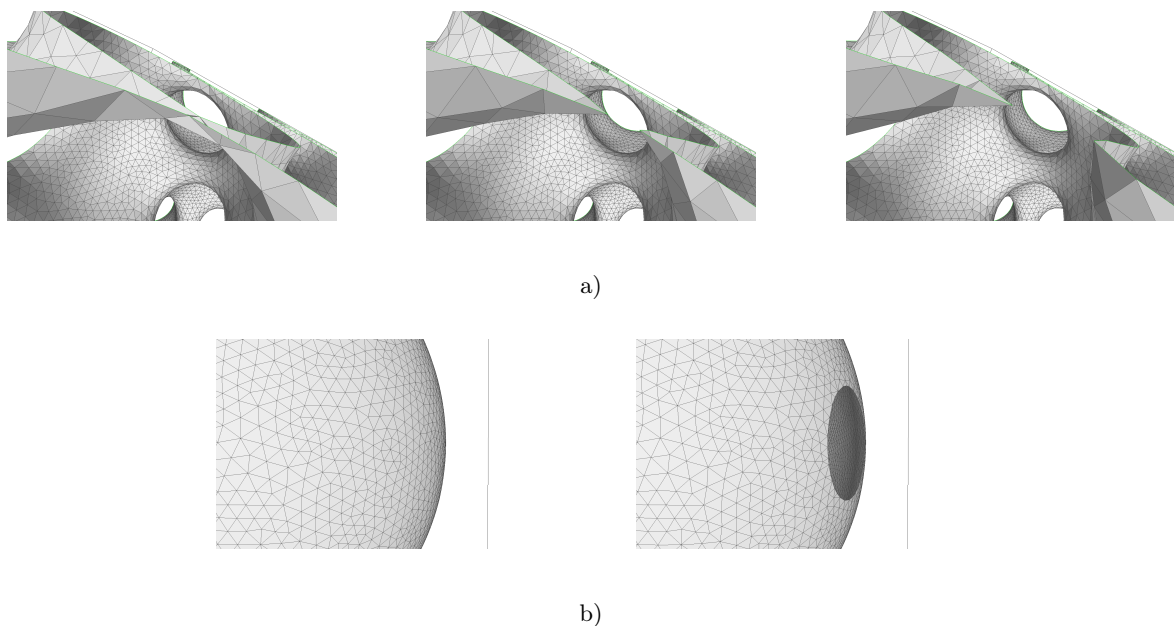


Рис. 7. Пример обработки особенности на грани куба: а) выход перешейка на грань; б) выход поверхности из куба

Fig. 7. An example of singularity processing on a cube face: a) a neck on a cube face; б) an escape of a surface out of cube

обе особенности имеют одинаковые признаки, то и обрабатывать их надо одним и тем же образом — удалением части, находящейся вне куба.

Как только произошел выход некоторой точки за грань куба, этой гранью отсекается вся выпешдая часть поверхности. Это можно сделать, используя функцию *clip* пакета *Polygon Mesh Processing*. При этом на границе триангуляция перестраивается так, чтобы появились новые граничные точки и ребра. К сожалению, на границе триангуляция уже не обязательно регулярна, поэтому необходимо ее еще раз перестроить. Для этой особенности обработку можно представить в виде алгоритма 3. Примеры работы алгоритма для каждой из описанных особенностей представлены на рис. 7.

3.3. Особенности на ребре куба. Следующая особенность — это все тот же перешеек, но на этот раз попавший на ребро куба. В этом случае уже нельзя использовать алгоритмы, представленные в разделах 3.1.1 и 3.2, так как они, если и обработают особенность, не дадут корректного расположения граничных вершин. Для правильного хода вычислений необходимо, чтобы все граничные вершины лежали на гранях куба, а две вершины после разрыва перешейка — на ребре (рис. 8). Поэтому для данного случая предусмотрен отдельный алгоритм.

Индикатором возникновения особенности здесь является ребро между двумя граничными точками на смежных гранях (рис. 8). Наличие такого ребра говорит о том, что в этом месте на следующем шаге схемы возникнет особенность. С точки зрения геометрии триангулированной поверхности в этом месте перешеек имеет минимально возможную толщину, ведь так как триангуляция близка к регулярной, l можно принять за единицу длины. Именно вдоль этого ребра и необходимо разделить поверхность. Далее опишем, как происходит сам процесс разделения. Прежде всего находятся две вершины v и w на разных гранях, соединенные ребром e . Затем обе вершины и ребро дублируются таким образом, чтобы принадлежать разным треугольникам (также меняются и связи между вершинами). После этого на ребре добавляются две точки u_1 и u_2 с соответствующей координатой, равной среднему между v и w . И на полученных тройках вершин u_1, v, w и u_2, v, w строятся треугольники. Таким образом, поверхность оказывается разделенной в самом тонком месте перешейка. Причем граничные вершины остаются на гранях куба, а добавленные вершины — на ребре. Работа алгоритма изображена на рис. 8, а его применение в процессе вычислений — на рис. 9.

Алгоритм 3. Алгоритм обработки особенности на грани куба
 Algorithm 3. Processing of singularity on the facet of the cube

```

1: for S ∈ Surfaces do
2:   for v ∈ S do
3:     if |v| > a then
4:       Plane P = v.DetectEscape()
5:       S = S.Cut(P)
6:     end if
7:   end for
8:   ReTriangulate(S, l, n)
9: end for
    
```

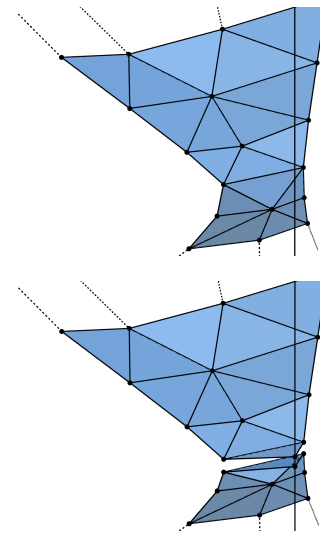


Рис. 8. Обработка особенности на ребре куба

Fig. 8. Singularity processing on the edge of a cube

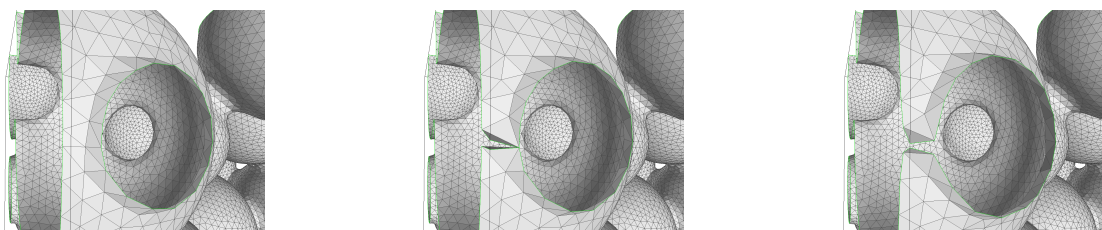


Рис. 9. Пример обработки особенности на ребре куба

Fig. 9. An example of singularity processing on the edge of a cube



Метод обработки кратко изложен в алгоритме 4. Функция *SplitAlongEdge*(v , w , v_1 , w_1) разбивает поверхность на две части вдоль ребра, соединяющего вершины v и w , добавляя вершины v_1 и w_1 , а функция *AddTriangle*(u , v , w) добавляет в триангуляцию треугольник с вершинами u , v , w . Алгоритмы работы данных функций будут опущены, так как их изложение было бы излишне громоздким, несмотря на простую идею работы. Сложности их описания заключаются в том, что необходимо не только добавить в триангуляцию новую вершину со связями, но и прописать связь с ней для каждого ее соседа, причем учтя порядок обхода вершины. Функции же *OnAdjacentFacet* и *AverageOnEdge* не описаны ввиду своей простоты. Первая из них проверяет, что вершина w лежит на смежной грани с гранью вершины v , вторая находит ребро, по которому две эти грани пересекаются, и ставит на него третью вершину в положении среднем между v и w .

3.4. Особенности в углу куба.

Иначе обрабатываются особенности, возникающие в углу куба (рис. 10). Геометрически это все тот же разрыв перешейка, но внутри куба теперь находится лишь его восьмая часть. Здесь уже нет необходимости разбивать поверхность в месте возникновения особенности, но появляются другие проблемы. Эволюция поверхности происходит таким образом, что точка, находящаяся на грани, движется только по грани, точка на ребре — только по ребру. Поэтому при обработке данной особенности необходимо “поднять” две вершины с ребра на грань и “заклеить” получившееся отверстие треугольником. После этого эволюция поверхности продолжится естественным образом.

Идея алгоритма очень проста: для каждой граничной вершины v , попавшей на ребро, проверяются две смежные с ней граничные вершины (см. алгоритм 5). Если одна из них, назовем ее w , также попадает на ребро куба (заметим, что это не может быть ребро, содержащее v), то находится общая для них грань куба. Затем каждая из точек “поднимается” вдоль нормали к этой грани на заданную высоту $\varepsilon > 0$. И, наконец, в триангуляцию

Алгоритм 4. Алгоритм обработки особенности на ребре куба
 Algorithm 4. Processing of singularity on the edge of the cube

```

1: for S ∈ Surfaces do
2:   for v ∈ S.BoundaryVertices() do
3:     for w ∈ v.Neighbours() do
4:       if w.IsBoundary() and w.OnAdjacentFacet(v) then
5:         Vertex v1 = v.Duplicate()
6:         Vertex w1 = w.Duplicate()
7:         S.SplitAlongEdge(v, w, v1, w1)
8:         Vertex u1 = AverageOnEdge(v, w)
9:         Vertex u2 = AverageOnEdge(v, w)
10:        S.AddTriangle(u1, v, w)
11:        S.AddTriangle(u2, v1, w1)
12:      end if
13:    end for
14:  end for
15: end for
    
```

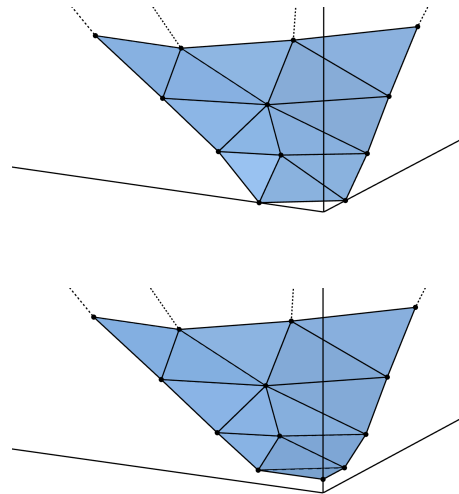


Рис. 10. Обработка особенности в углу куба

Fig. 10. Singularity processing in the corner of a cube

Алгоритм 5. Алгоритм обработки особенности в углу куба
 Algorithm 5. Processing of singularity in the corner of the cube

```

1: for S ∈ Surfaces do
2:   for v ∈ S.BoundaryVertices() do
3:     if v.IsOnEdge()
4:       for w ∈ v.Neighbours() do
5:         if w.IsBoundary() and w.IsOnEdge() then
6:           LiftFromPlane(v, w, ε)
7:           Vertex u = AverageOnEdge(v, w)
8:           S.AddTriangle(u, v, w)
9:         end if
10:      end for
11:    end if
12:  end for
13: end for
    
```

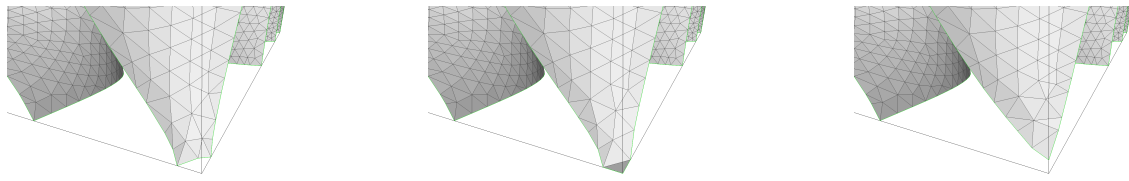


Рис. 11. Пример обработки особенности в углу куба
 Fig. 11. An example of singularity processing in the corner of a cube

добавляется третья точка u , находящаяся на третьем из ребер, образующих угол, на той же высоте ε . Вместе с v и w точка u образует треугольник, “заклеивающий” отверстие, получившееся при поднятии точек. Пример работы алгоритма 5 представлен на рис. 11.

4. Численные эксперименты. В качестве примеров работы алгоритма будет рассмотрена эволюция различных периодических поверхностей, полученных комбинацией некоторого числа сфер. Все описанные особенности вместе появлялись только в случае последней рассмотренной поверхности из 30 сфер. В остальных же случаях будет обращать внимание на какой-то отдельный тип особенности. Во всех численных экспериментах коэффициент диффузии d был взят равным 1.

4.1. Упаковка 4-х сфер в кубе. Первым примером является эволюция периодической поверхности, сегмент которой составлен из одной сферы и трех полусфер радиуса 3, соединенных цилиндрическими перешейками (рис. 12). Начальная поверхность состоит из 17080 вершин и 33748 треугольников. Данная поверхность находится в кубе с ребром 12. Периодические поверхности постоянной средней кривизны с такой топологией неизвестны, поэтому результатом эволюции может быть либо сфера ($H \equiv 1/R$), либо цилиндр ($H \equiv 1/(2R)$), где R — радиус сферы или цилиндра. Каждая из этих поверхностей может быть достигнута лишь с возникновением особенности: на ребре или на грани. Выбор именно такой начальной поверхности обусловлен тем, что здесь можно явно увидеть появление особенности из раздела 3.3.

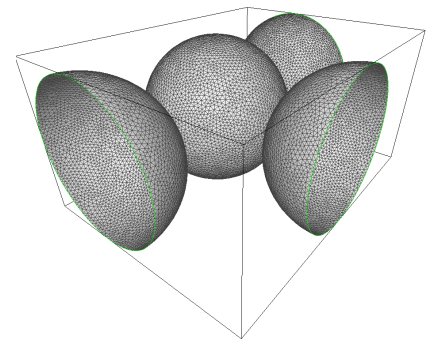


Рис. 12. Упаковка 4-х сфер в кубе
 Fig. 12. Packing of 4 spheres in a cube

Сначала поверхность изменяется только в окрестности перешейков, так как средняя кривизна сфер постоянна. Затем меняться начинает вся поверхность целиком (рис. 13). Можно заметить, что сначала окружность на передней правой грани уменьшается, а затем увеличивается в размерах, причем между этими фазами наблюдается замедление эволюции. Это может быть связано с тем, что в окрестности полученной поверхности есть некоторая поверхность постоянной средней кривизны, к которой, однако, нельзя прийти с данной начальной поверхности. График отклонения средней кривизны от ее среднего значения на рис. 14 не показывает наличие поверхности постоянной средней кривизны, лишь отклонение в среднем на 15–20%. Это небольшое отклонение, но все же недостаточное для утверждения о постоянстве средней кривизны.

По мере расширения уже упомянутой окружности происходит также и расширение окружностей на боковых гранях. Таким образом, поверхность постепенно подходит к ребру куба, на котором должен произойти разрыв поверхности. Если рассматривать поверхность целиком (отразив ее как минимум два раза относительно разных плоскостей), то этот процесс разрыва соответствует заклеиванию отверстий в поверхности. Сам процесс обработки этой особенности представлен на рис. 15.

После обработки особенности поверхность эволюционирует в сегмент цилиндра, средняя кривизна которого постоянна (рис. 16). Вычисления начинались с шагом 0.0002, затем по мере замедления эволюции он увеличивался до 0.2. Такое большое значения шага было необходимо, так как при близости поверхности к цилиндру эволюция практически останавливалась. Отдельно отметим, что при прохождении особенности шаг схемы был достаточно большим — 0.05, но все равно позволил корректно ее обработать. В общем же случае особенности (например, внутри куба) шаг схемы лучше уменьшать, так как в окрестности самопересечения поверхности средняя кривизна резко возрастает. За время вычислений объем изменился не

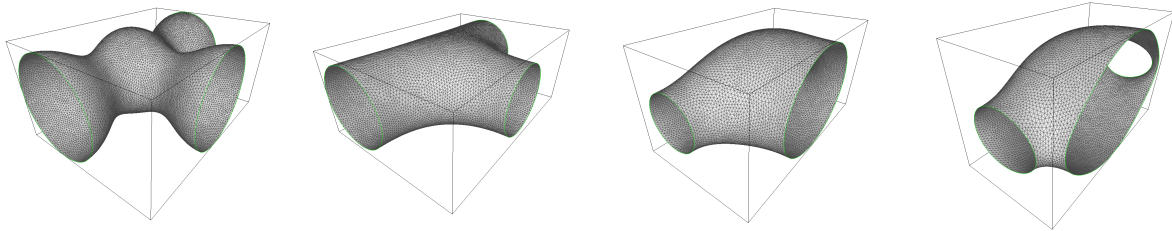


Рис. 13. Эволюция упаковки из 4-х сфер в моменты времени 1.7102, 31.6202, 354.0202 и 391.1702
 Fig. 13. Evolution of a packing of 4 spheres at times 1.7102, 31.6202, 354.0202 and 391.1702

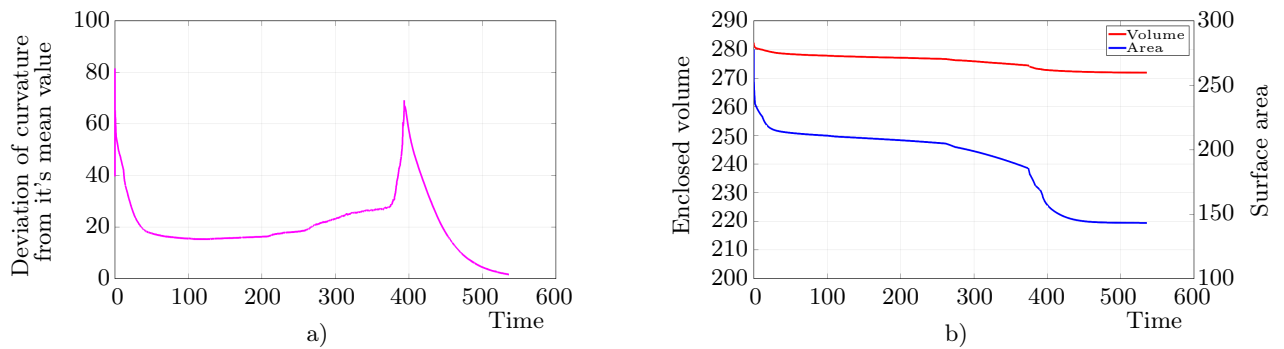


Рис. 14. Графики зависимостей: а) отклонения кривизны от ее среднего значения в процентах;
 б) изменения площади поверхности и объема, ею ограниченного
 Fig. 14. Graphs of dependences: a) deviation of curvature from its average value in percent;
 b) change in surface area and enclosed volume

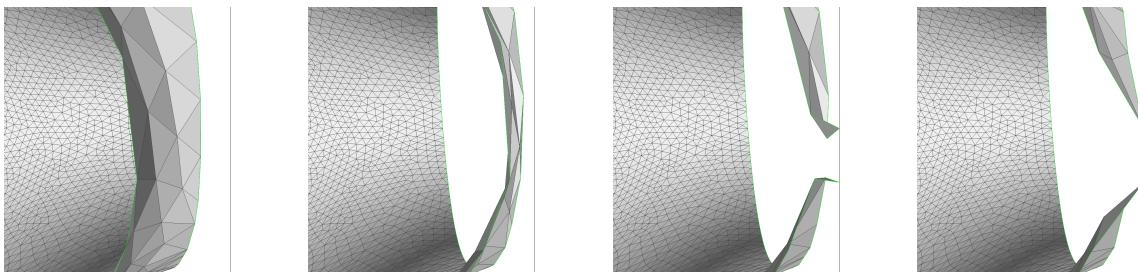


Рис. 15. Процесс обработки особенности на ребре при $t = 391.9702$
 Fig. 15. Singularity processing on the edge at $t = 391.9702$

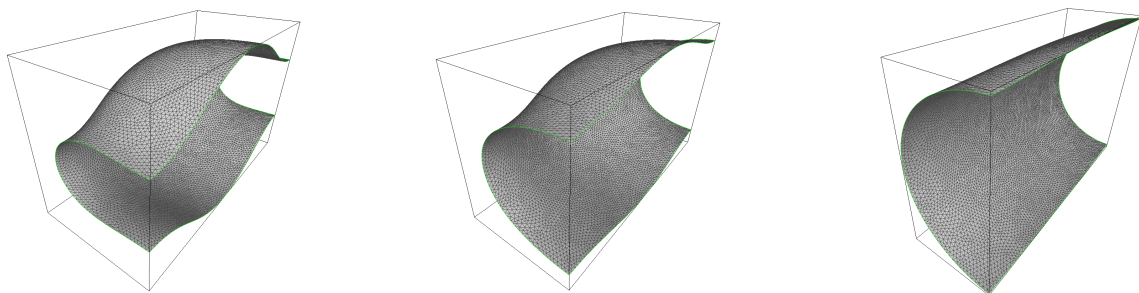


Рис. 16. Эволюция упаковки из 4-х сфер в моменты времени 396.1702, 416.0202 и 536.0202
 Fig. 16. Evolution of a packing of 4 spheres at times 396.1702, 416.0202 and 536.0202

более, чем на 4.5% от изначального, а площадь монотонно убывала с 276.7 до 143.2 (рис. 14). Изломы графиков соответствуют моментам возникновения особенностей.

4.2. Упаковка 7-ми сфер в кубе. Рассмотрим здесь периодическую упаковку сфер следующего вида (рис. 17). Начальная триангуляция содержит 25930 вершин и 51045 треугольников. Как и ранее, радиус сфер равен 3, сторона куба, в котором они находятся, равна 12. Но, в отличие от предыдущей поверхности, данная обладает большим числом симметрий, поэтому результатом эволюции может быть либо сфера, либо некоторая периодическая поверхность постоянной средней кривизны.

По мере эволюции кривизна поверхности выравнивается, поэтому скорость изменения поверхности падает. Из-за этого необходимо увеличивать шаг по времени τ . Поэтому для того, чтобы обнаружить нетривиальные (не плоскости и не сферы) поверхности постоянной средней кривизны, разностная схема должна обладать определенной степенью устойчивости. Несмотря на отсутствие точных оценок, предложенная схема оказывается достаточно устойчивой для нахождения таких поверхностей, что и будет показано ниже. Начальный шаг по времени был равен 0.0002, затем по мере замедления эволюции он увеличивался до 0.05. На рис. 18 представлено поведение поверхности с течением времени. Видно, что постепенно крайние сферы уменьшаются до тех пор, пока эволюция не останавливается при достижении последней поверхности. Если же обратиться к графику отклонения средней кривизны от ее среднего значения на рис. 19 а, можно заметить, что в этот момент кривизна практически постоянна — отклонение

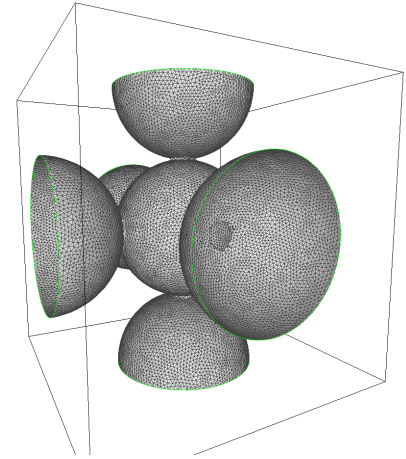


Рис. 17. Упаковка 7-ми сфер в кубе
 Fig. 17. Packing of 7 spheres in a cube

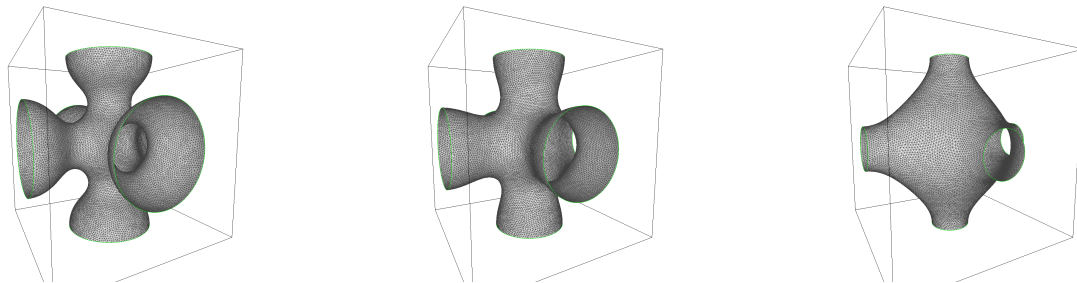


Рис. 18. Эволюция упаковки из 7-ми сфер в моменты времени 0.0472, 0.1462 и 3.5682
 Fig. 18. Evolution of a packing of 7 spheres at times 0.0472, 0.1462 and 3.5682

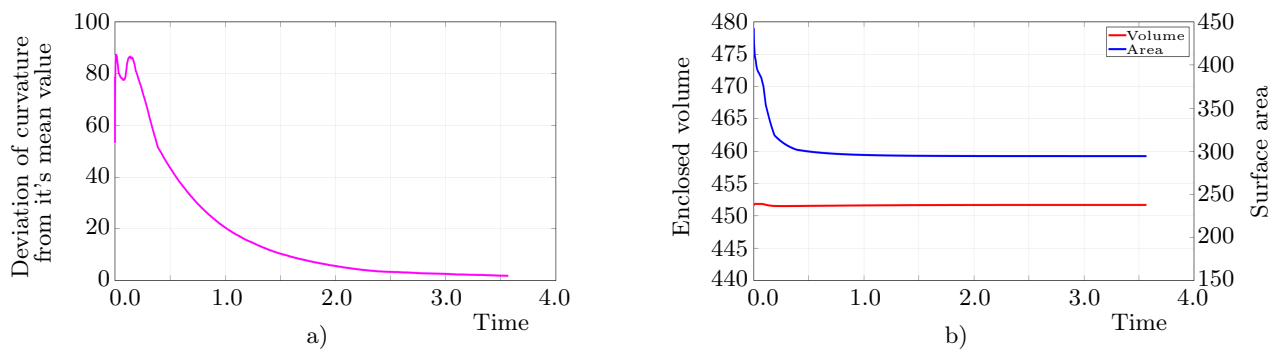


Рис. 19. Графики зависимостей: а) отклонения кривизны от ее среднего значения в процентах;
 б) изменения площади поверхности и объема, ею ограниченного

Fig. 19. Graphs of dependences: a) deviation of curvature from its average value in percent;
 б) change in surface area and enclosed volume

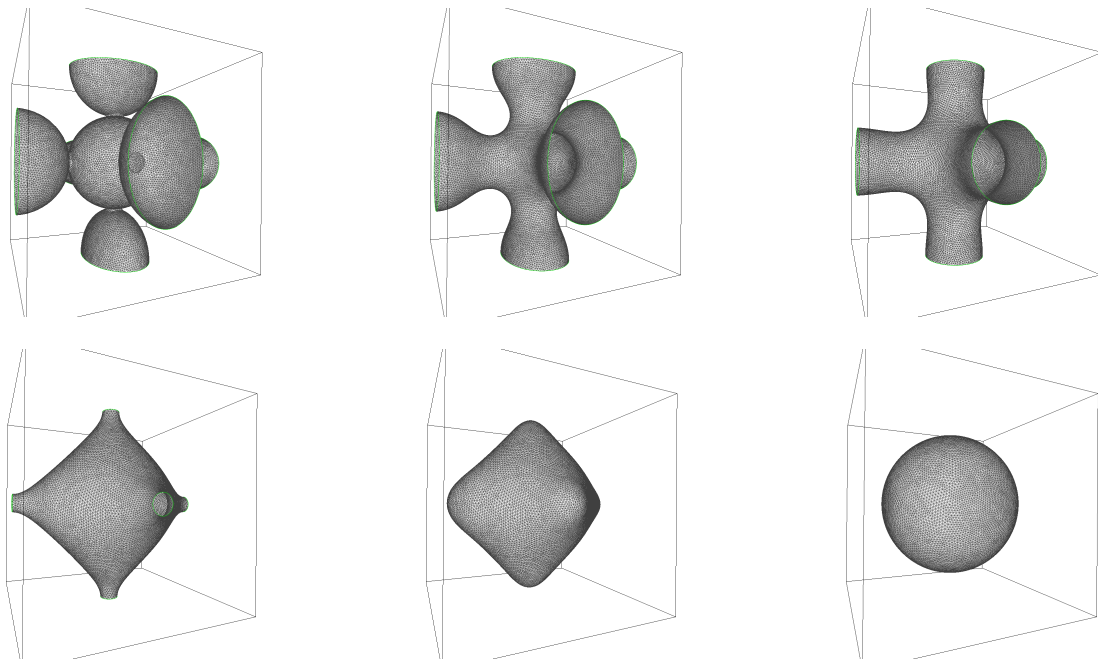


Рис. 20. Эволюция измененной упаковки при $t \in [0, 11.8882]$

Fig. 20. Evolution of the modified packing at $t \in [0, 11.8882]$

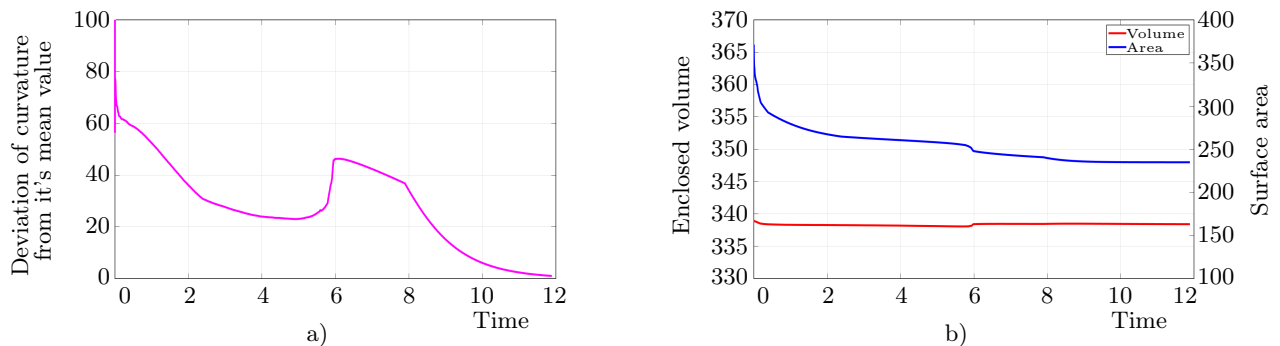


Рис. 21. Графики зависимостей: а) отклонения кривизны от ее среднего значения в процентах; б) изменения площади поверхности и объема, ею ограниченного

Fig. 21. Graphs of dependences: a) deviation of curvature from its average value in percent; b) change in surface area and enclosed volume

лишь в 1.8%. Таким образом, мы видим, что изложенный выше подход позволяет проводить вычисления с большими шагами по времени и при этом не пропускать нетривиальные поверхности постоянной средней кривизны. Объем за время вычислений изменился менее, чем на 0.2%, а площадь уменьшилась с 438.8 до 294.3 — более, чем на 30%.

Но во время эволюции этой поверхности топология поверхности не менялась, соответственно не применялись и алгоритмы перестройки. Поэтому далее изменим начальную поверхность — сожмем каждую из боковых сфер по одной из осей. Тогда поверхность потеряет симметрию и примет вид как на рис. 20. В этом случае объем, ограниченный поверхностью, будет уже меньше, поэтому по мере уменьшения площади процесс уже не остановится до тех пор, пока не произойдет разрыв на границе. Разрывы происходят именно таким образом, как было описано в разделе 3.2, один из них показан на рис. 6. До момента возникновения разрывов отклонение средней кривизны от ее среднего значения составляло не менее 20%, поэтому никакая поверхность постоянной средней кривизны пропущена не была. График отклонения на рис. 21 а имеет пик — это момент возникновения большого числа особенностей. Объем, ограниченный поверхностью, изменился не более, чем на 0.3%, площадь монотонно убывает с 368.2 до 234.9.

4.3. Упаковка 9-ти сфер в кубе. Далее рассмотрим периодическую упаковку, полученную из девяти сфер, одна из которых находится в центре куба с ребром 8, а остальные — в его вершинах (рис. 22). Радиус сфер равен $2\sqrt{3}$, он подбирался таким образом, чтобы все сферы касались друг друга. Начальная триангуляция состоит из 17195 вершин и 33446 треугольников.

Здесь, в отличие от предыдущего случая, граница сегмента поверхности находится не на одной грани, а сразу на нескольких. Поэтому возникают 24 вершины, находящиеся на ребрах куба, и их движение происходит только вдоль ребер. Такое положение вершин вносит определенные сложности, так как во время перетриангуляции эти вершины могут быть удалены. Чтобы этого не происходило, вершины, находящиеся на ребрах, помечаются как ограничения функции *isotropic_remeshing* — она их оставляет на месте. Вершины же, находящиеся на гранях, при перетриангуляции остаются на гранях — при удалении какой-либо вершины ребро все так же остается на грани, а все новые вершины добавляются также на ребра.

В этом случае схема также показывает хорошую устойчивость и приводит поверхность к поверхности постоянной средней кривизны (рис. 23). Особенности в этом случае не возникают. Объем за время вычислений меняется менее чем на 0.3%, а площадь убывает с 286.7 до 212.3. К сожалению, среднее значение \bar{H} средней кривизны H поверхности за время вычисления меняется от положительных значений к отрицательным, а поэтому проходит через ноль. Из-за этого график функции $H_{dev}/|\bar{H}| \cdot 100$ имеет пик и некорректно отражает поведение отклонения кривизны. Поэтому для данной поверхности на рис. 24 а показан график самой H_{dev} . Отклонение кривизны от среднего значения в конце вычислений составляет лишь 5%.

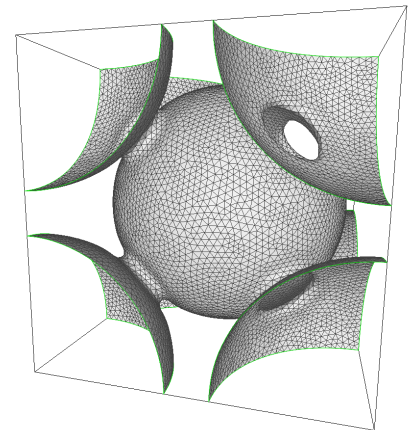


Рис. 22. Упаковка 9-ти сфер в кубе
 Fig. 22. Packing of 9 spheres in a cube

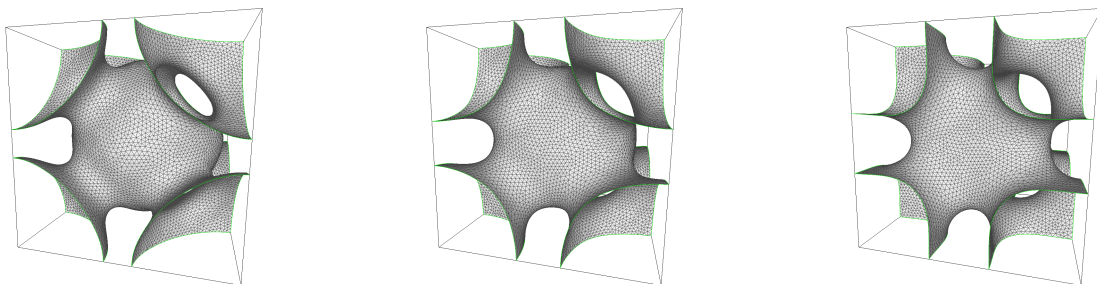


Рис. 23. Эволюция упаковки из 9-ти сфер в моменты времени 0.0302, 0.0962 и 5.7762
 Fig. 23. Evolution of a packing of 9 spheres at times 0.0302, 0.0962 and 5.7762

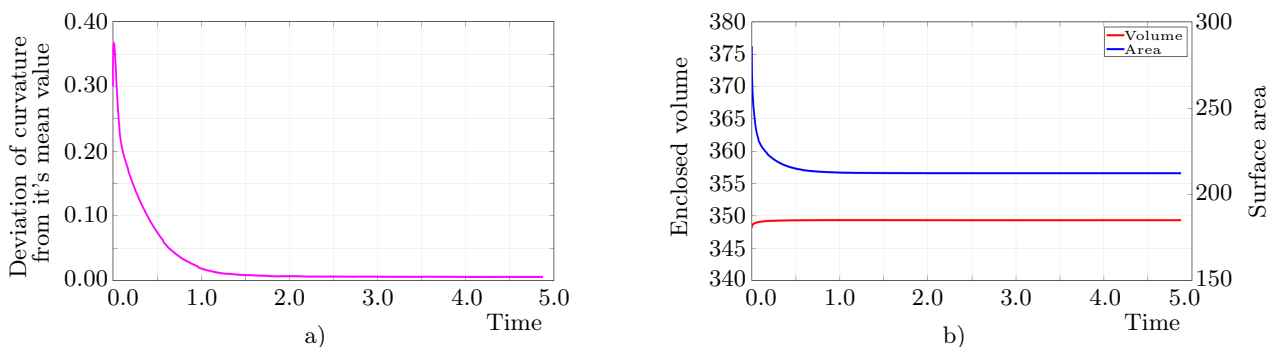


Рис. 24. Графики зависимостей: а) отклонения кривизны от ее среднего значения;
 б) изменения площади поверхности и объема, ею ограниченного
 Fig. 24. Graphs of dependences: a) deviation of curvature from its average value;
 b) change in surface area and enclosed volume



4.4. Упаковка 30-ти сфер в кубе. Все предыдущие поверхности были проверкой работы алгоритмов перед моделированием какой-либо поверхности сложной топологии. В приложениях часто возникают упаковки достаточно большого числа сфер [20], являющиеся моделями, например, некоторых пористых сред. Поэтому для проверки работы алгоритма была выбрана поверхность, составленная из 30-ти сфер радиуса 0.6964 (рис. 25), упакованных в куб со стороной 6 алгоритмом Любачевского–Стиллинжера [21]. В частности, такая поверхность может служить моделью сорбента при химической реакции, но с более сложным уравнением [3].

Начальная триангуляция состоит из 39234 вершин и 77362 треугольников. В этом случае целью не было как можно точнее смоделировать эволюцию данной поверхности, основной интерес представляет сама возможность рассчитать эту эволюцию до конца. Этим объясняется меньшая точность приближения поверхности по сравнению с предыдущими примерами. При эволюции данной упаковки возникло большое число особенностей всех типов (рис. 26), поэтому мы не будем останавливаться на большей части из них. Опишем лишь те, которые приводили к значительным изменениям топологии поверхности либо были интересны с точки зрения работы алгоритма.

В начале вычислений возникало и обрабатывалось большое количество особенностей на гранях куба, внутри же топология поверхности не менялась. Затем, начиная с момента времени $t = 0.00841$, начались разрывы поверхности внутри куба. После одного шага схемы могло возникнуть сразу несколько разрывов перешейков, описанных в 3.1.1. Затем в момент времени 4.15161 поверхность разделяется на две компоненты связности: небольшой сегмент сферы и всю оставшуюся поверхность. Достаточно быстро меньшая компонента становится сферой (рис. 27), и ее эволюция останавливается до тех пор, пока ее не коснется вторая компонента. Слияние происходит в момент времени 119.16161, когда применяется алгоритм из раздела 3.1.2 и две поверхности объединяются в одну. Отдельно отметим, что на практике между собой в таком виде пересекаются только две различные компоненты связности. Внутри же одной компоненты

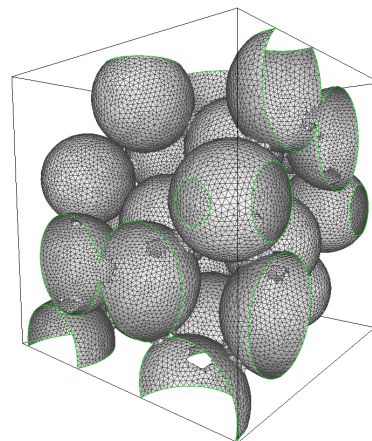


Рис. 25. Упаковка 30-ти сфер в кубе

Fig. 25. Packing of 30 spheres in a cube

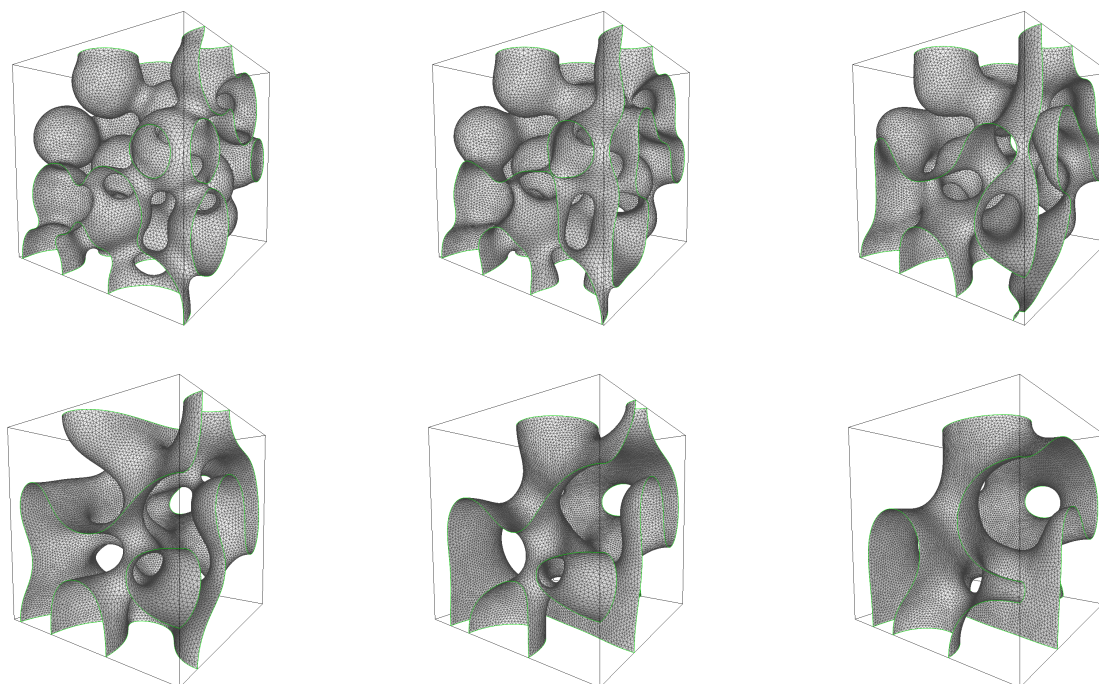


Рис. 26. Эволюция поверхности при $t \in [0.00106, 1.20761]$

Fig. 26. Surface evolution at $t \in [0.00106, 1.20761]$

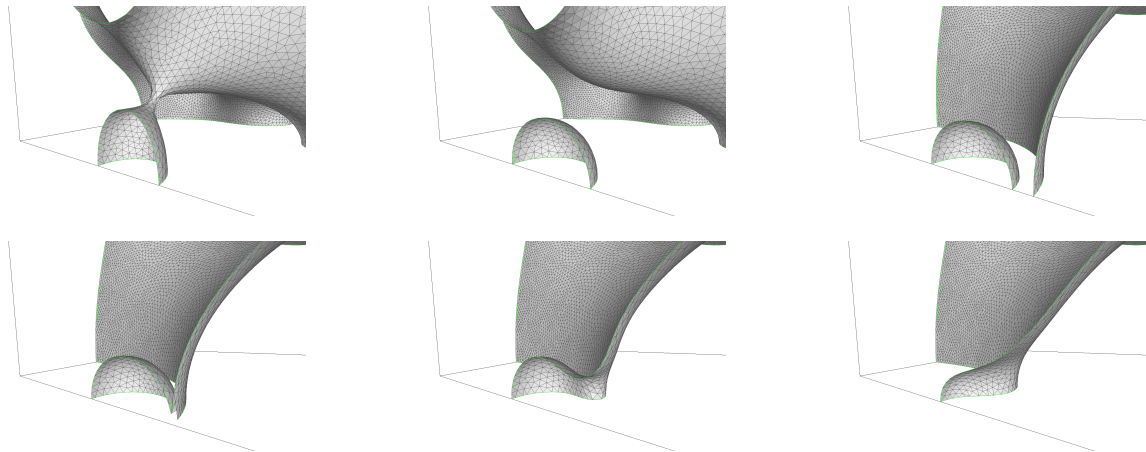


Рис. 27. Разделение поверхности на две компоненты при $t = 4.15161$ и дальнейшее их слияние при $t = 119.16161$
 Fig. 27. Splitting of the surface into two components at $t = 4.15161$ and their further merging at $t = 119.16161$

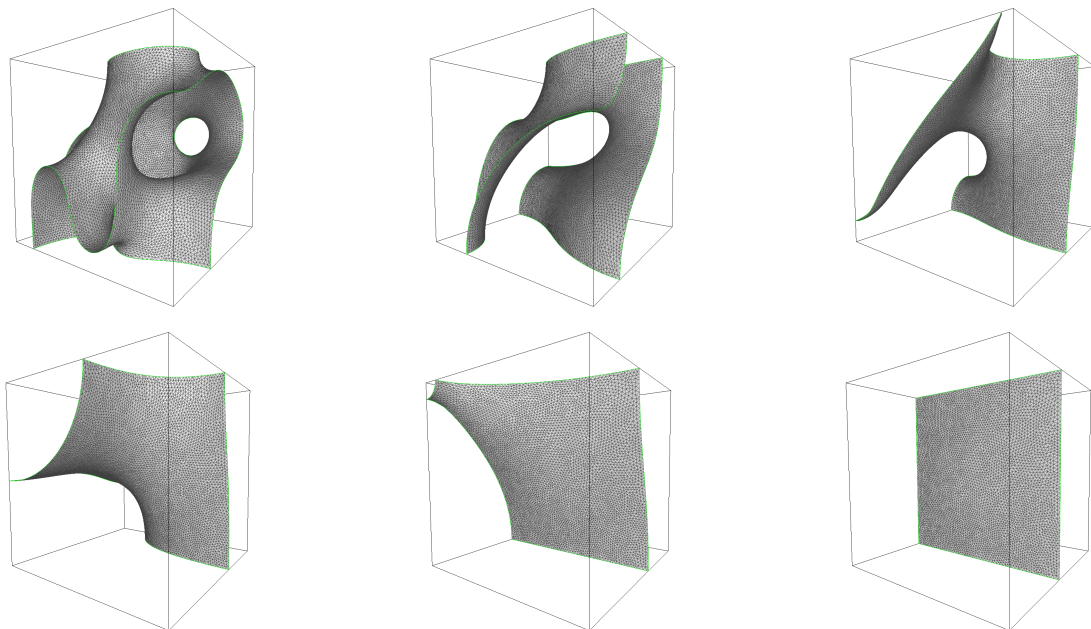


Рис. 28. Эволюция поверхности при $t \in [2.40361, 1129.41111]$
 Fig. 28. Surface evolution at $t \in [2.40361, 1129.41111]$

самопересечение может произойти только в виде перешейка с возникновением особенности. К сожалению, аналитического результата, подтверждающего это наблюдение, пока никем получено не было.

Параллельно с разделением/слиянием поверхностей идет процесс эволюции и изменения топологии, но уже более медленный, чем был в начале вычислений. В момент времени $t = 122.46111$ поверхность принимает вид как на рис. 28 слева снизу, и скорость эволюции существенно замедляется. В это время наблюдается та же ситуация, что и в случае упаковки из четырех сфер — поверхность находится вблизи некоторой поверхности постоянной средней кривизны, но не может в нее попасть. Отклонение средней кривизны от среднего значения колеблется в пределах 20–30%, но ниже не опускается. Затем поверхность выходит из этого состояния, возникают две особенности, и в момент времени $t = 1129.41111$ поверхность достигает стационарной точки потока — плоскости ($H \equiv 0$).

На рис. 29 представлены графики отклонения средней кривизны от ее среднего значения и изменения площади поверхности и объема. Из-за того, что за время вычислений шаг схемы существенно менялся, для наглядности на графиках все величины зависят не от времени, а от номера шага. Среднее значение \bar{H} стремилось к нулю, поэтому на графики выведено именно среднее отклонение H_{dev} , а не его процент

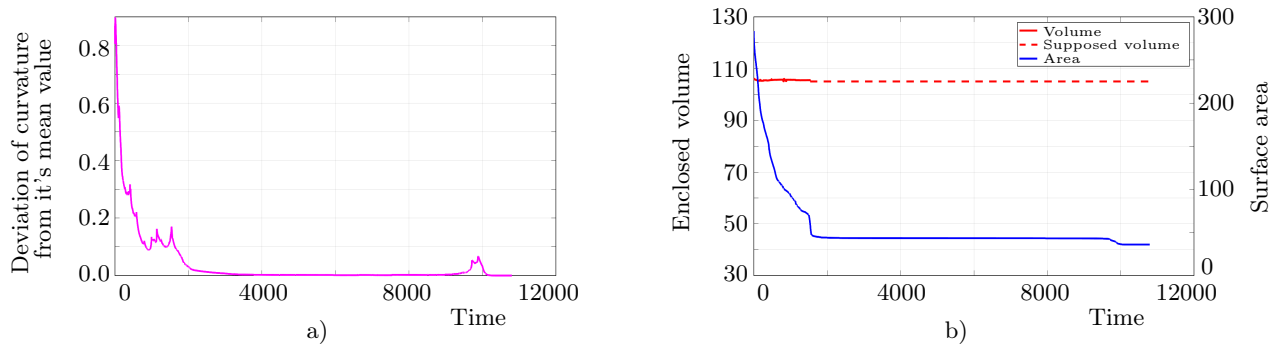


Рис. 29. Графики зависимостей: а) отклонения кривизны от ее среднего значения; б) изменения площади поверхности и объема, ею ограниченного

Fig. 29. Graphs of dependences: a) deviation of curvature from its average value; b) change in surface area and enclosed volume

от \tilde{H} , который бы возрастал по мере приближения поверхности к плоскости. Каждый пик на рис. 29 а — это особенность, возникшая внутри или на грани куба. Площадь поверхности монотонно убывала с 280.1 до 36, объем же, равный в начале 105.9, за первые 1537 шагов изменился не более, чем на 1% (рис. 29 б). После этого поверхность изменилась настолько, что стало невозможным вычислить объем, ограниченный гранями куба и этой поверхностью. Но в конце вычисления была получена плоскость $z \approx 0.082650$, для которой можно вычислить объем области, ограниченный ею и гранями куба: $6 \cdot 6 \cdot (3 - 0.082650) = 105.0246$. Хотя объем в конце вычисления и отличается от изначального лишь на 1%, это еще не гарантирует его сохранение в течение всего времени эволюции, но, безусловно, является хорошим признаком.

5. Поведение объема, ограниченного поверхностью. Во всех численных экспериментах, за исключением упаковки из 4-х сфер (раздел 4.1), отклонение объема было несущественным и позволяло говорить о том, что объем не меняется. Но для упаковки 4-х сфер отклонение в 4.5% является существенным. Для того чтобы оценить изменение объема и понять, почему отклонение оказалось большим, обратимся к формуле его вычисления, более удобной для анализа — основанной на формуле Остроградского–Гаусса:

$$V(S) = \int_{\Omega} d\omega = \frac{1}{3} \int_{\Omega} \operatorname{div}(\mathbf{X}) d\omega = \frac{1}{3} \int_S \mathbf{N} \mathbf{X} ds \approx \frac{1}{3} \sum_{i=1}^N \mathbf{N}_i \mathbf{X}_i A(U_i).$$

С помощью данной формулы можно оценить изменение объема от n -го шага к $(n+1)$ -му. Рассмотрим, при каких предположениях объем будет сохраняться и какие из этих предположений реализуемы на практике. Сначала предположим, что между двумя шагами разностной схемы количество вершин не изменится, следовательно:

$$\begin{aligned} V(S^{n+1}) - V(S^n) &\approx \frac{1}{3} \sum_{i=1}^N (\mathbf{N}_i^{n+1} \mathbf{X}_i^{n+1} A(U_i^{n+1}) - \mathbf{N}_i^n \mathbf{X}_i^n A(U_i^n)) = \\ &= \frac{1}{3} \sum_{i=1}^N (\mathbf{N}_i^{n+1} \mathbf{X}_i^{n+1} - \mathbf{N}_i^n \mathbf{X}_i^n) A(U_i^n) + \frac{1}{3} \sum_{i=1}^N \mathbf{N}_i^{n+1} \mathbf{X}_i^{n+1} (A(U_i^{n+1}) - A(U_i^n)) \approx \\ &\approx -\frac{\tau}{3} \sum_{i=1}^N \left(\left(I + \frac{\tau}{2} \Delta_n^2 \right)^{-1} \Delta_n H^n \right)_i A(U_i^n) + \frac{1}{3} \sum_{i=1}^N \mathbf{N}_i^{n+1} \mathbf{X}_i^{n+1} (A(U_i^{n+1}) - A(U_i^n)). \end{aligned}$$

Далее предположим, что все окрестности вершин имеют одинаковые площади, т.е. $A(U_i^n) = A$. Тогда матрица Δ_n будет симметрической, причем сумма элементов любой ее строки равна нулю. Это значит, что векторы-столбцы с равными элементами (постоянные функции) лежат в ядре оператора Δ_n . Используя перестановочность матриц Δ_n и $\left(I + \frac{\tau}{2} \Delta_n^2 \right)^{-1}$, получаем:

$$\sum_{i=1}^N \left(\left(I + \frac{\tau}{2} \Delta_n^2 \right)^{-1} \Delta_n H^n \right)_i (A)_i = \sum_{i=1}^N \left(\Delta_n \left(I + \frac{\tau}{2} \Delta_n^2 \right)^{-1} H^n \right)_i (A)_i = \sum_{i=1}^N \left(\left(I + \frac{\tau}{2} \Delta_n^2 \right)^{-1} H^n \right)_i (\Delta_n A)_i = 0.$$

Таким образом, получаем оценку на изменение объема, ограниченного поверхностью:

$$V(S^{n+1}) - V(S^n) \approx \frac{1}{3} \sum_{i=1}^N N_i^{n+1} \mathbf{X}_i^{n+1} (A(U_i^{n+1}) - A(U_i^n)).$$

К сожалению, полученное выражение далее упростить уже не удастся, так как нельзя в общем виде оценить изменение площади окрестности каждой вершины. Однако можно заметить, что изменение объема находится в некоторой зависимости от изменения площади поверхности, причем эта зависимость оказывается различной на рассмотренных примерах. Если шаг по времени τ достаточно мал, то площади окрестностей вершин мало изменятся за один шаг. Все произведения $N_i^{n+1} \mathbf{X}_i^{n+1}$ ограничены, так как $\|\mathbf{N}\| = 1$ и поверхность находится в кубе, следовательно уменьшением шага можно добиться и малого изменения объема.

С другой стороны, если не увеличивать шаг τ , то со временем эволюция поверхности будет практически незаметна. В случае близости поверхности к некоторой стационарной точке эволюция замедляется, а поэтому шаг по времени увеличивается. И здесь возможны две ситуации: если стационарная точка достижима, то шагов делается небольшое количество, а если этой точки достичь невозможно, то проводится большое количество шагов со все возрастающим τ . В этом случае с каждым шагом схемы объем будет все сильнее отклоняться от своего изначального значения. Это и отличает эволюцию упаковки 4-х сфер от других примеров — продолжительное время поверхность находилась вблизи некоторой стационарной точки и не могла покинуть это состояние. Это значит, что для поверхностей с такими особенностями эволюции необходимо тщательно подбирать шаг по времени, с одной стороны, позволяющий быстрее пройти моменты “почти стационарности”, а с другой — не допускающий большого изменения объема.

6. Заключение. В статье был представлен метод численного моделирования потока поверхностной диффузии для начальной периодической поверхности. Также предложены алгоритмы, позволяющие обрабатывать различные особенности, возникающие во время вычислений. Работа алгоритмов была продемонстрирована на различных примерах, для которых были выполнены свойства потока (2). Кроме сфер и цилиндров, самых частых стационарных точек, были получены также некоторые периодические поверхности постоянной средней кривизны. Для оценки возможностей алгоритма была рассчитана до конца эволюция упаковки из тридцати сфер в кубе. Данный метод может применяться не только для решения уравнения поверхностной диффузии, но и для других задач моделирования, в которых важную роль играет поведение поверхности вещества или границы раздела сред. Однако, в случае других уравнений эволюции, могут появиться новые особенности, не учитываемые предложенным методом. Открытыми и пока не решенными автором статьи проблемами являются: создание эффективного метода обработки пересечения двух различных поверхностей и вычисление объема, ограниченного гранями куба и периодической поверхностью с границей произвольной сложности.

Список литературы

1. Mullins W.W. Theory of thermal grooving // Journal of Applied Physics. 1957. 28, N 3. 333–339. doi 10.1063/1.1722742.
2. Cahn J.W., Elliott C.M., Novick-Cohen A. The Cahn–Hilliard equation with a concentration dependent mobility: motion by minus the Laplacian of the mean curvature // European Journal of Applied Mathematics. 1996. 7, N 3. 287–301. doi 10.1017/S0956792500002369.
3. Bazaikin Ya.V., Derevschikov V.S., Malkovich E.G., Lysikov A.I., Okunev A.G. Evolution of sorptive and textural properties of CaO-based sorbents during repetitive sorption/regeneration cycles: Part II. Modeling of sorbent sintering during initial cycles // Chemical Engineering Science. 2019. 199. 156–163. doi 10.1016/J.CES.2018.12.065.
4. Escher J., Mayer U.F., Simonett G. The surface diffusion flow for immersed hypersurfaces // SIAM Journal on Mathematical Analysis. 1998 29, N 6. 1419–1433. doi 10.1137/S0036141097320675.
5. Mayer U.F., Simonett G. Self-intersections for the surface diffusion and the volume preserving mean curvature flow // Differential and Integral Equations. 2000. 13, N 7–9. 1189–1199. <https://projecteuclid.org/journals/differential-and-integral-equations/volume-13/issue-7-9>. Cited May 21, 2022.
6. Efremenko Yu.D. On semi-implicit numerical method for surface diffusion equation for triangulated surfaces // Sib. Electron. Math. Rep. 2021. 18, N 2, 1367–1389. doi 10.33048/semi.2021.18.104.



7. *Smereka P.* Semi-implicit level set methods for curvature and surface diffusion motion // Journal of Scientific Computing. 2003. **19**. 439–456. doi [10.1023/A:1025324613450](https://doi.org/10.1023/A:1025324613450).
8. *Mayer U.F.* Numerical solutions for the surface diffusion flow in three space dimensions // Computational and Applied Mathematics. 2001. **20**, N 3. 361–379.
9. The Computational Geometry Algorithms Library (CGAL). <https://www.cgal.org/>. Cited May 21, 2022.
10. Linear Algebra Library (Eigen). <https://eigen.tuxfamily.org>. Cited May 21, 2022.
11. 3D Surface Mesh Generation Package. https://doc.cgal.org/latest/Surface_mesher/index.html. Cited May 21, 2022.
12. *Cenanovic M., Hansbo P., Larson M.G.* Finite element procedures for computing normals and mean curvature on triangulated surfaces and their use for mesh refinement // ArXiv.org. 2017. <https://arxiv.org/pdf/1703.05745.pdf>. Cited May 21, 2022.
13. *Watanabe K., Belyaev A.G.* Detection of salient curvature features on polygonal surfaces // Comput. Graph. Forum. 2001. **20**, N 3. 385–392. doi [10.1111/1467-8659.00531](https://doi.org/10.1111/1467-8659.00531).
14. *Magid E., Soldea O., Rivlin E.* A comparison of Gaussian and mean curvature estimation methods on triangular meshes of range image data // Computer Vision and Image Understanding. 2007. **107**, N 3. 139–159. doi [10.1016/j.cviu.2006.09.007](https://doi.org/10.1016/j.cviu.2006.09.007).
15. *Pinkall U., Polthier K.* Computing discrete minimal surfaces and their conjugates // Exper. Math. 1993. **2**, N 1. 15–36. doi [10.1080/10586458.1993.10504266](https://doi.org/10.1080/10586458.1993.10504266).
16. *Meyer M., Desbrun M., Schröder P., Barr A.H.* Discrete differential-geometry operators for triangulated 2-manifolds // Visualization and Mathematics III. Heidelberg: Springer, 2003. 35–57. doi [10.1007/978-3-662-05105-4_2](https://doi.org/10.1007/978-3-662-05105-4_2).
17. Polygon Mesh Processing Package. https://doc.cgal.org/latest/Polygon_mesh_processing/index.html. Cited May 21, 2022.
18. *Jung W., Shin H., Choi B.K.* Self-intersection removal in triangular mesh offsetting // Comput.-Aided Des. Appl. 2004. **1**, N 1–4. 477–484. doi [10.1080/16864360.2004.10738290](https://doi.org/10.1080/16864360.2004.10738290).
19. Scale-Space Surface Reconstruction Package. https://doc.cgal.org/latest/Scale_space_reconstruction_3/index.html. Cited May 21, 2022.
20. *Torquato S.* Random Heterogeneous Materials: Microstructure and Macroscopic Properties. New York: Springer, 2013.
21. *Lubachevsky B.D., Stillinger F.H.* Geometric properties of random disk packings // J. Stat. Phys. 1990. **60**, N 5–6. 561–583. doi [10.1007/BF01025983](https://doi.org/10.1007/BF01025983).

Поступила в редакцию
30 марта 2022 г.

Принята к публикации
14 мая 2022 г.

Информация об авторе

Юрий Данилович Ефременко — инженер-исследователь, Новосибирский государственный университет, ул. Пирогова, 2, 630090, Новосибирск, Российская Федерация.

References

1. W. W. Mullins, “Theory of Thermal Grooving,” J. Appl. Phys. **28** (3), 333–339 (1957). doi [10.1063/1.1722742](https://doi.org/10.1063/1.1722742).
2. J. W. Cahn, C. M. Elliott, and A. Novick-Cohen, “The Cahn–Hilliard Equation with a Concentration Dependent Mobility: Motion by Minus the Laplacian of the Mean Curvature,” Eur. J. Appl. Math. **7** (3), 287–301 (1996). doi [10.1017/S0956792500002369](https://doi.org/10.1017/S0956792500002369).
3. Ya. V. Bazaikin, V. S. Derevschikov, E. G. Malkovich, et al., “Evolution of Sorptive and Textural Properties of CaO-Based Sorbents during Repetitive Sorption/Regeneration Cycles: Part II. Modeling of Sorbent Sintering during Initial Cycles,” Chem. Eng. Sci. **199**, 156–163 (2019). doi [10.1016/J.CES.2018.12.065](https://doi.org/10.1016/J.CES.2018.12.065).
4. J. Escher, U. F. Mayer, and G. Simonett, “The Surface Diffusion Flow for Immersed Hypersurfaces,” SIAM J. Math. Anal. **29** (6), 1419–1433 (1998). doi [10.1137/S0036141097320675](https://doi.org/10.1137/S0036141097320675).
5. U. F. Mayer and G. Simonett, “Self-Intersections for the Surface Diffusion and the Volume Preserving Mean Curvature Flow,” Differ. Integral Equ. **13** (7–9), 1189–1199 (2000). <https://projecteuclid.org/journals/differential-and-integral-equations/volume-13/issue-7-9>. Cited May 21, 2022.

6. Yu. D. Efremenko, “On Semi-Implicit Numerical Method for Surface Diffusion Equation for Triangulated Surfaces,” *Sib. Electron. Math. Rep.* **18** (2), 1367–1389 (2021). doi [10.33048/semi.2021.18.104](https://doi.org/10.33048/semi.2021.18.104).
7. P. Smereka, “Semi-Implicit Level Set Methods for Curvature and Surface Diffusion Motion,” *J. Sci. Comput.* **19**, 439–456 (2003). doi [10.1023/A:1025324613450](https://doi.org/10.1023/A:1025324613450).
8. U. F. Mayer, “Numerical Solutions for the Surface Diffusion Flow in Three Space Dimensions,” *Comput. Appl. Math.* **20** (3), 361–379 (2001).
9. The Computational Geometry Algorithms Library (CGAL). <https://www.cgal.org/>. Cited May 21, 2022.
10. Linear Algebra Library (Eigen). <https://eigen.tuxfamily.org>. Cited May 21, 2022.
11. 3D Surface Mesh Generation Package. https://doc.cgal.org/latest/Surface_mesher/index.html. Cited May 21, 2022.
12. M. Cenanovic, P. Hansbo, and M. G. Larson, “Finite Element Procedures for Computing Normals and Mean Curvature on Triangulated Surfaces and Their Use for Mesh Refinement,” *ArXiv.org*. (2017). <https://arxiv.org/pdf/1703.05745.pdf>. Cited May 21, 2022.
13. K. Watanabe and A. G. Belyaev, “Detection of Salient Curvature Features on Polygonal Surfaces,” *Comput. Graph. Forum* **20** (3), 385–392 (2001). doi [10.1111/1467-8659.00531](https://doi.org/10.1111/1467-8659.00531).
14. E. Magid, O. Soldea, and E. Rivlin, “A Comparison of Gaussian and Mean Curvature Estimation Methods on Triangular Meshes of Range Image Data,” *Comput. Vis. Image Underst.* **107** (3), 139–159 (2007). doi [10.1016/j.cviu.2006.09.007](https://doi.org/10.1016/j.cviu.2006.09.007).
15. U. Pinkall and K. Polthier, “Computing Discrete Minimal Surfaces and Their Conjugates,” *Exper. Math.* **2** (1), 15–36 (1993). doi [10.1080/10586458.1993.10504266](https://doi.org/10.1080/10586458.1993.10504266).
16. M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr, “Discrete Differential-Geometry Operators for Triangulated 2-Manifolds,” in *Visualization and Mathematics III* (Springer, Heidelberg, 2003), pp. 35–57. doi [10.1007/978-3-662-05105-4_2](https://doi.org/10.1007/978-3-662-05105-4_2).
17. Polygon Mesh Processing Package. https://doc.cgal.org/latest/Polygon_mesh_processing/index.html. Cited May 21, 2022.
18. W. Jung, H. Shin, and B. K. Choi, “Self-Intersection Removal in Triangular Mesh Offsetting,” *Comput.-Aided Des. Appl.* **1** (1–4), 477–484 (2004). doi [10.1080/16864360.2004.10738290](https://doi.org/10.1080/16864360.2004.10738290).
19. Scale-Space Surface Reconstruction Package. https://doc.cgal.org/latest/Scale_space_reconstruction_3/index.html. Cited May 21, 2022.
20. S. Torquato, *Random Heterogeneous Materials: Microstructure and Macroscopic Properties* (Springer, New York, 2013).
21. B. D. Lubachevsky and F. H. Stillinger, “Geometric Properties of Random Disk Packings,” *J. Stat. Phys.* **60** (5–6), 561–583 (1990). doi [10.1007/BF01025983](https://doi.org/10.1007/BF01025983).

Received
March 30, 2022

Accepted for publication
May 14, 2022

Information about the author

Yury D. Efremenko — Research Engineer, Novosibirsk State University, ulitsa Pirogova, 2, 630090, Novosibirsk, Russia.