

УДК 519.6

DOI:10.25729/ESI.2024.34.2.009

Алгоритм сглаживания пути на триангулированной поверхности

Серебровская Екатерина Александровна, Марчевский Илья Константинович,
Ерофеева Мария Александровна

Московский государственный технический университет имени Н.Э. Баумана,
Россия, Москва, *e.a.serebrovskaya@gmail.com*

Аннотация. Предложен и реализован в Wolfram Mathematica алгоритм оптимизации длины пути на триангулированной поверхности. Первые два шага «легковесны», но предполагают вариацию траектории, проходящей по ребрам, лишь в пределах примыкающих к ним треугольников. Последующие шаги позволяют за несколько итераций прийти к кратчайшему в математическом смысле пути. Сходимость алгоритма не доказана строго, но обеспечивается в большом количестве рассмотренных примеров.

Ключевые слова: кратчайший путь, триангулированная поверхность, условная оптимизация, Wolfram Mathematica

Цитирование: Серебровская Е.А. Алгоритм сглаживания пути на триангулированной поверхности / Е.А. Серебровская, И.К. Марчевский, М.А. Ерофеева // Информационные и математические технологии в науке и управлении. – 2024. – № 2(34). – С. 90-98. – DOI:10.25729/ESI.2024.34.2.009.

Введение. В процессе разработки новой модификации бессеточного лагранжева метода вихревых петель [1] для моделирования пространственного обтекания тел потоком несжимаемой среды и вычисления действующих на них аэрогидродинамических нагрузок возникла проблема коррекции пространственной кривой сложной формы, которая определяет форму каждой петли. Подобные петли задают форму вихревых структур, тем самым определяя распределение завихренности в области течения (в вихревых методах именно завихренность является первичной расчетной величиной). Такой подход позволяет, в частности, обеспечить точное выполнение условия бездивергентности.

Вихревые петли генерируются на обтекаемой поверхности, а затем движутся в области течения; для моделирования этих процессов разработано семейство вычислительных алгоритмов, в том числе полуэмпирических. В силу неизбежных погрешностей, связанных, в том числе, с заданием триангулированной формы обтекаемой поверхности вместо гладкой, конечного шага расчета по времени и многими другими допущениями, некоторые участки петель в ходе расчета их движения в потоке «проникают» в обтекаемое тело и поэтому требуется «вынести» их на поверхность тела, обеспечив достаточную гладкость вновь формируемой траектории. Вместо последней задачи можно решить задачу поиска кратчайшего пути на триангулированной поверхности.

Задача поиска кратчайшего пути, лежащего на некоторой поверхности, весьма проста по постановке и может быть отнесена к классическим задачам дифференциальной геометрии кривых и поверхностей – речь идет о построении геодезической. В простом и популярном изложении этот вопрос рассмотрен в [2]. Однако в различных вычислительных алгоритмах поверхность лишь в редких случаях бывает заданной аналитически и обладает необходимой степенью гладкости; как правило, на практике приходится иметь дело с ее упрощенным представлением в виде поверхности некоторого многогранника с плоскими гранями. Далее будем рассматривать наиболее общий случай триангулированной поверхности, поскольку грани с более чем тремя вершинами можно разбить на треугольники. Для такой «кусочно-плоской» (или, по-другому, «граненой») поверхности неприменим аппарат дифференциального исчисления, и для решения поставленной задачи следует применять иные подходы.

1. Кратчайший путь на графе и на триангулированной поверхности. Близкая, на первый взгляд, задача о поиске кратчайшего пути на взвешенном графе давно и успешно решена: наиболее известны алгоритмы Дейкстры [3] (1959 г.), Флойда – Уоршалла (1962 г.), Белмана

– Форда (1969 г.). Не вдаваясь в детали, отметим, что они могут быть применены к графу, ребрами которого выступают ребра рассматриваемой триангулированной поверхности – ее «каркас»: в результате будет найден кратчайший из путей, проходящих по ребрам. Отметим, что если начальная и конечная точки не совпадают с вершинами, то треугольные грани, которым они принадлежат, следует очевидным образом разбить на три треугольника и свести задачу к более простой. Найденный в результате путь, помимо того, что не является кратчайшим на поверхности, обладает особенностью, которая может быть важной для некоторых приложений: он не является гладким (рис. 1).

Естественно, речь идет не о гладкости в математическом смысле: никакой из путей на триангулированной поверхности не может быть гладким, если только он не проходит целиком по граням, лежащим в одной плоскости, или не включает в себя специальным образом построенные сопряжения; в последнем случае он едва ли может быть кратчайшим. Нестрогое, но понятное и наглядное определение «гладкого» пути на триангулированной поверхности можно дать следующим образом: если сделать развертку для тех граней поверхности, по которым проходит путь, то на развертке углы между звеньями полученной кривой должны быть развернутыми, за исключением случаев прохождения пути через вершину. Близкий к оптимальному путь, углы на развертке которого близки к развернутым, будем называть «достаточно гладким».

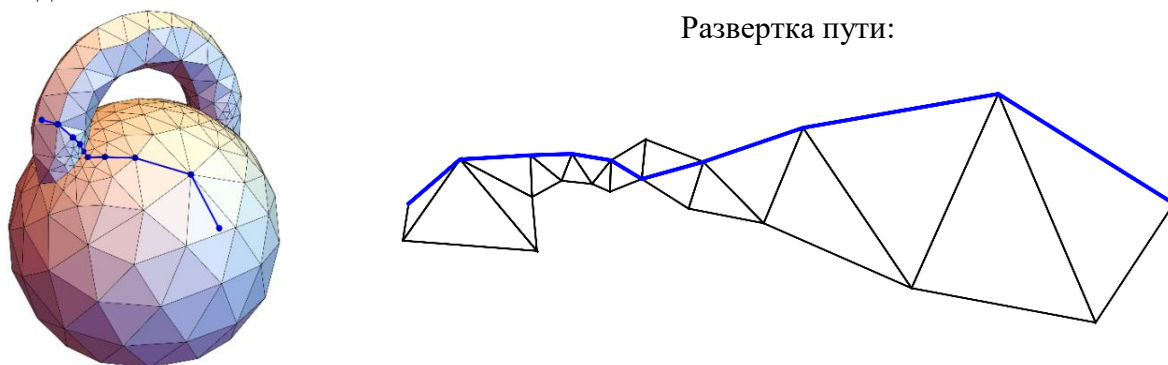


Рис. 1. Путь, найденный при помощи алгоритма Дейкстры, и его развертка

Также следует заметить, что функция длины пути может иметь как глобальный, так и локальные минимумы, соответственно, необходимо различать глобально- и локально-кратчайшие пути. Очевидно, что если путь на развертке образует прямую линию, то он будет, во-первых, локально-кратчайшим, а во-вторых – наиболее гладким из возможных в указанном выше смысле. Далее необходимо сделать три замечания.

1. Локально-кратчайших путей между двумя фиксированными точками может быть несколько, они могут при этом значительно различаться по длине, однако каждый из них на развертке будет представлять собой прямую линию (рис. 2); будем рассматривать процедуру поиска локально-кратчайшего пути, используя в качестве начального приближения результат работы алгоритма Дейкстры.
2. Локально-кратчайший путь на развертке может не быть прямой линией, а оставаться ломаной: последнее характерно для тех случаев, когда исходная поверхность имеет седловые точки, в этом случае искомый путь может проходить через вершину седла (и, соответственно, через вершины примыкающих к седловой точке треугольников на развертке), рис. 3.
3. Задачу поиска локально-кратчайшего (и тем более глобально-кратчайшего) пути на практике во многих случаях, как представляется, можно заменить более «слабой» задачей поиска достаточно гладкого в вышеуказанном смысле пути.

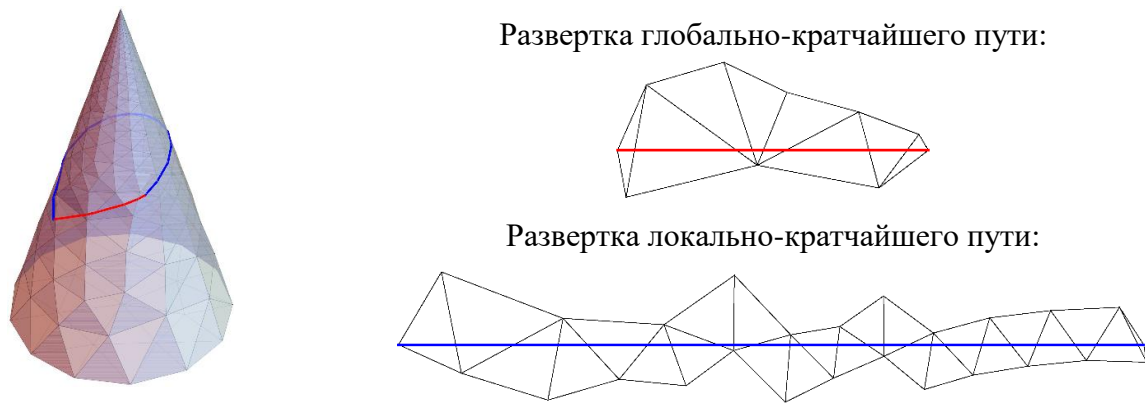


Рис. 2. Два локально-кратчайших пути на конусе и их развертки

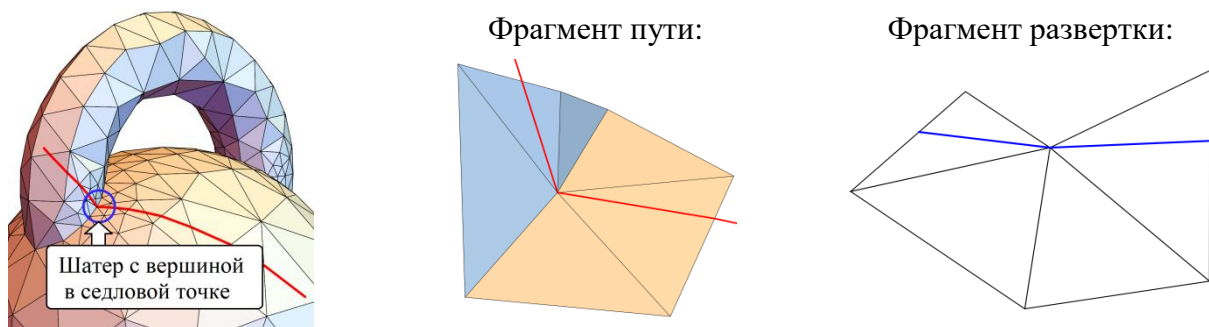


Рис. 3. Кратчайший путь в окрестности седловой точки

Несмотря на кажущуюся простоту задачи поиска кратчайшего пути на триангулированной поверхности, универсальные методы ее эффективного решения до сих пор не разработаны. Публикации на эту тему стали активно появляться еще в 1970-е годы; над смежными проблемами работали такие известные специалисты по алгоритмам вычислительной геометрии, как Д. Киркпатрик, Ф. Препарата, Д. Ли и др. Задача значительно упрощается, если рассматриваемая поверхность выпуклая; краткий обзор соответствующих алгоритмов приведен в [4], однако на практике требуются более универсальные алгоритмы, применимые к невыпуклым телам сложной формы (например, см. рис. 3). Подобные алгоритмы стали появляться в 1980-1990-е годы, выделим среди них работу [5]; обзор таких алгоритмов и их сравнительный анализ по состоянию на 2009 г. приведены в [6]. Вплоть до сегодняшнего дня регулярно появляются новые алгоритмы, среди которых отметим [7, 8]; в абсолютном большинстве из них в той или иной мере использована технология построения разверток, многие алгоритмы предполагают построение некоторого дерева и поиск по нему. Тем не менее, до сих пор остаются актуальными методы «грубой силы», предполагающие сильное дробление поверхности на ячейки очень простой формы, на которых решить задачу, по крайней мере, локально, весьма легко; к таким алгоритмам относится [8].

Отметим, что в хорошо известной библиотеке вычислительной геометрии CGAL [9] имеется модуль *Triangulated Surface Mesh Shortest Paths*, позволяющий решить задачу поиска глобально-кратчайшего пути на триангулированной поверхности. Реализованный в нем алгоритм основан на работе [10], в которой, в свою очередь, разработана усовершенствованная версия достаточно «старого» алгоритма [5].

Сравнительная простота использования библиотеки (вопросы ее предварительной компиляции и настройки опускаем) может в значительной мере нивелироваться не слишком высокой производительностью: с ее помощью наиболее эффективно могут быть решены задачи поиска путей от выбранной точки на поверхности до множества других точек, причем даже

значительное увеличение числа точек назначения приводит к весьма малому росту вычислительной сложности. Последнее связано со структурой алгоритма: сначала производится занимающая значительное время «предобработка» всей поверхности для заданной точки-источника, а затем выполняется прокладка пути в точку назначения, что является намного более легковесной операцией. Если же требуется решать задачу для множества различных пар точек «начало – конец маршрута», и при этом, как было отмечено выше, оптимальность пути не является критичной, а достаточно лишь обеспечить сравнительную его гладкость, указанный алгоритм весьма неэффективен.

2. Алгоритм решения задачи. Предлагаемый алгоритм состоит из нескольких шагов, на каждом из которых происходит сглаживание пути и уменьшение его длины. Приведем их краткое описание, опуская некоторые «технические» детали реализации, которые представляются довольно очевидными, но их описание заняло бы слишком много места и едва ли способствовало пониманию сути алгоритма.

1. Используя алгоритм Дейкстры, построить кратчайший путь на графе – каркасной модели поверхности; он будет иметь вид $\{A_1, A_2, \dots, A_N\}$, где A_i – номера (индексы) вершин многогранника.

2. Разбить найденный путь на тройки $\{A_1, A_2, A_3\}$, $\{A_3, A_4, A_5\}$ и т.д.; для каждой такой тройки рассмотреть треугольники, имеющие общей вершиной точку с четным индексом A_{2k} (их совокупность будем называть словом «шатер»). Для каждого шатра рассмотреть две его части, ограниченные ребрами $A_{2k-1}A_{2k}$ и $A_{2k}A_{2k+1}$; построить их развертки. Если угол при вершине A'_{2k} (штрих обозначает образ вершины на развертке) меньше развернутого, то кратчайший путь на этом «полушатре» – прямая, соединяющая точки A'_{2k-1} и A'_{2k+1} , которая не будет проходить через A'_{2k} . Если углы на развертках обоих полушатров меньше развернутого, то из двух путей следует выбрать тот, что короче. Если, наоборот, на обеих развертках углы при A'_{2k} больше развернутого, то имеем дело с седловой точкой (рис. 3); в этом случае любой из путей по шатру между начальной и конечной точками будет длиннее исходного пути, проходящего через вершину.

3. Заменить вершины в пути, имеющие четные индексы, наборами точек, соответствующих точкам пересечения полученного пути с ребрами шатров (рис. 4); путь в этом случае будет иметь вид $\{A_1, \{C_1^2, C_2^2, \dots, C_{m_2}^2\}, A_3, \{C_1^4, C_2^4, \dots, C_{m_4}^4\}, \dots, A_N\}$, при этом положение каждой «промежуточной» точки C_m^{2k} удобно задавать, указывая вершины ребра, на котором она лежит, и весовой коэффициент: $C_m^{2k} \leftrightarrow \{A_{2k}, A_m^{2k}, \alpha_m^{2k}\}$, где A_m^{2k} – та из вершин поверхности, что лежит на соответствующем месте в основании шатра с вершиной A_{2k} ; $\alpha_m^{2k} \in [0, 1]$ – весовой коэффициент, нулевое и единичное значения веса говорят о совпадении точки пути с вершиной A_{2k} и A_m^{2k} соответственно.

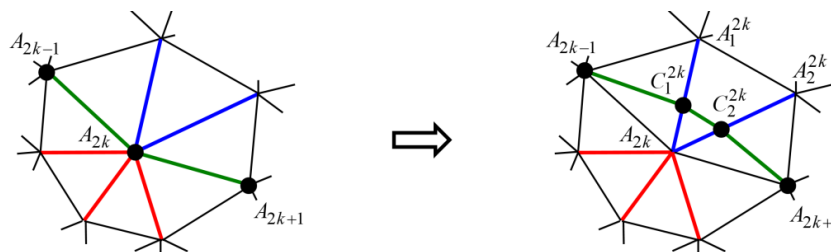


Рис. 4. Оптимизация на шатре с четным индексом вершины

4. Прodelать аналогичную операцию, но рассматривая шатры с вершинами в точках с нечетными индексами A_{2k+1} ; в этом случае «разрез» на полушатры будет производиться не по ребрам шатра, а по линиям $C_{m_{2k}}^{2k}A_{2k+1}$ и $A_{2k+1}C_1^{2k+2}$, проходящим в общем случае по граням шатра (рис. 5). В итоге полученный путь будет образован точками $\{A_1, \{C_1^2, C_2^2, \dots, C_{m_2}^2\}, \{C_1^3,$

$C_2^3, \dots, C_{m_3}^3\}, \{C_1^4, C_2^4, \dots, C_{m_4}^4\}, \dots, A_N\}$, а каждая из точек C_m^k – задана тройкой указанного в предыдущем пункте вида.

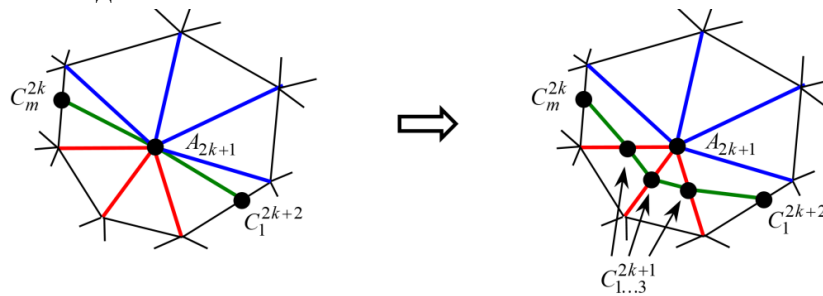


Рис. 5. Оптимизация на шатре с нечетным индексом вершины

Результат выполнения предыдущих операций назовем «слабой оптимизацией» пути: в результате путь сглаживается и укорачивается, однако остается проходящим по тем из треугольных граней, которые примыкают (по ребрам) к исходному пути. В простых случаях найденный таким образом путь может оказаться достаточно гладким, однако в общем случае требуется дальнейшая оптимизация.

4. «Вытянуть» путь. Для этого нужно построить функцию длины пути на соответствующей развертке, считая веса α_i^j переменными (их количество равно числу внутренних точек на пути), и найти ее условный минимум в гиперкубе $0 \leq \alpha_i^j \leq 1$. Несмотря на простоту постановки, эта задача нетривиальна в тех ситуациях, когда некоторые из параметров α_i^j выходят на ограничения: градиент функции длины пути в этом случае имеет неопределенность вида $[0/0]$, а компоненты матрицы Гессе становятся бесконечными. Причина состоит в том, что выход весового параметра α_i^j на ограничение означает приближение соответствующей точки на оптимизируемом пути к одной из вершин. Однако в этом случае и несколько других весовых коэффициентов обязаны принять нулевое или единичное значение, что будет соответствовать сближению вплоть до совпадения нескольких смежных точек C_i^j и, следовательно, равенству нулю длины участка пути между этими точками.

Отметим также, что даже если проблема выхода параметров α_i^j на ограничения не возникает, или она каким-либо аналитическим либо алгоритмическим способом решена, задача безусловной оптимизации пути является весьма непростой для практического решения: если $\Delta\alpha_i^j$ – отклонения параметров от оптимальных значений, то отклонение целевой функции длины пути на развертке от минимального значения имеет вид:

$$\Delta\Psi = o\left(\sum_{i,j} q_{i,j}(\Delta\alpha_i^j)^2\right),$$

где $q_{i,j}$ – некоторые коэффициенты порядка единицы.

При сравнительно небольшом числе звеньев ломаной ее численная минимизация не вызывает существенных сложностей даже при использовании простых методов типа наискорейшего спуска или сопряженных градиентов, однако с увеличением количества звеньев скорость сходимости градиентных методов становится весьма низкой. Разработка эффективных и робастных алгоритмов решения данной задачи представляет собой вопрос для отдельного исследования.

В то же время представляется, что данную задачу можно решить, воспользовавшись простой и наглядной механической аналогией, о которой упомянуто в [2]: путь следует представить в виде нити, которая проходит по развертке и натягивается в начальной и конечной точках. Результирующей формой пути будет кусочно-прямолинейная, «цепляющая» в своих угловых точках некоторые из вершин треугольников развертки (см. далее рис. 7).

5. «Смена развертки». Для точек ломаной, соответствующих нулевым и единичным весовым коэффициентам, т.е. совпадающих с вершинами ячеек, следует рассмотреть шатер с вершиной в этой точке и заменить в развертке треугольники из входящего в нее полушатра треугольниками из второго полушатра (рис. 6). При этом разумно сделать шаг слабой оптимизации, развернув «новый» полушатер и проложив по нему путь, чтобы он образовал отрезок прямой.

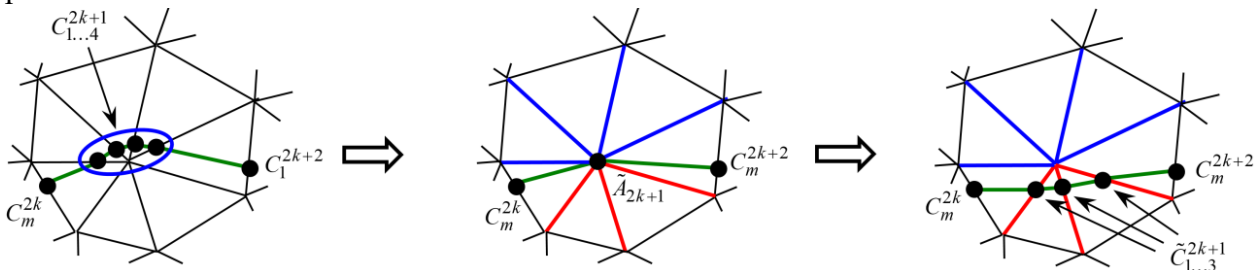


Рис. 6. Схема смены развертки

Отметим, что во избежание заикливания алгоритма целесообразно за один шаг оптимизации, соответствующий данному пункту, изменять развертку лишь в одной «угловой» вершине предыдущего пути. После этого следует повторить п. 5 и вытянуть путь на новой развертке (рис. 7). Алгоритм завершается при получении на развертке прямолинейного пути без изломов, либо пути с изломами в вершинах, соответствующих седловым точкам исходной триангулированной поверхности, когда найти более короткий путь невозможно.

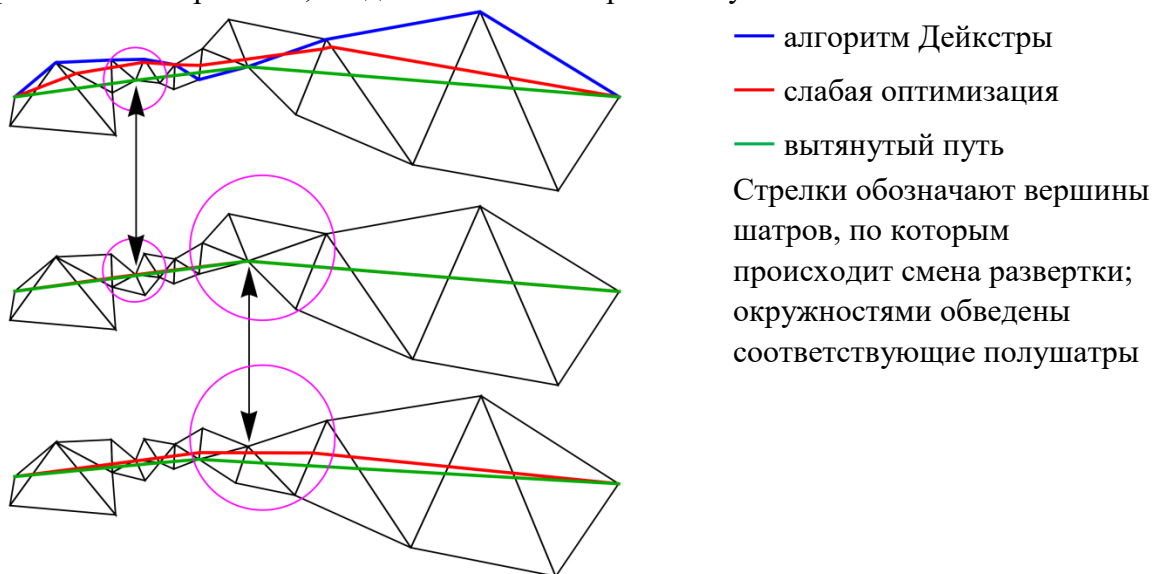


Рис. 7. Несколько первых шагов оптимизации пути

3. Результаты, полученные с помощью реализованного алгоритма. Рассмотрим примеры применения предложенного алгоритма для модели сферы с равномерной поверхностной сеткой, грубой модели тела сложной формы и весьма подробной модели винтового движителя со сгущением сетки на лопастях (рис. 8).

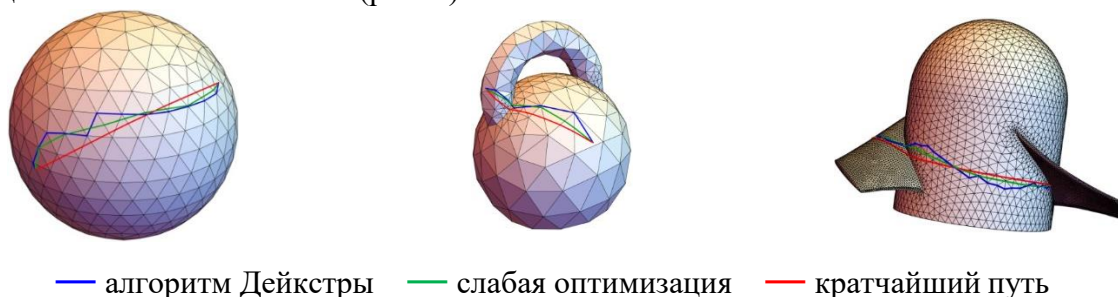


Рис. 8. Результаты оптимизации пути

Путь, найденный с помощью алгоритма Дейкстры, включает для данных тел 13, 9 и 44 сегмента соответственно. Кратчайший путь состоит из 25, 19 и 102 сегментов; для его отыскания после двух шагов слабой оптимизации выполнены 4, 7 и 99 итераций вытягивания пути. График изменения длины пути показан на рис. 9.

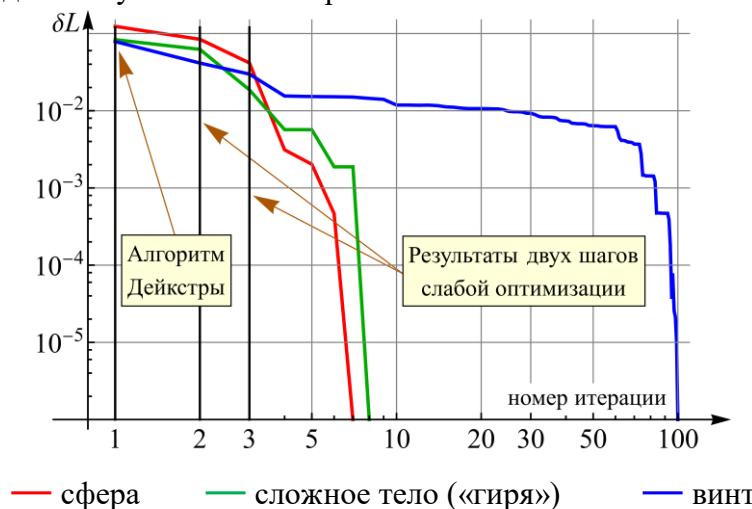


Рис. 9. Изменение длины в процессе оптимизации пути

Величина δL означает отношение превышения длины пути над минимальной длиной к этой минимальной длине. Видно, что путь, найденный с использованием алгоритма Дейкстры, в рассмотренных примерах отличается по длине от оптимального примерно на 10 %; после двух шагов слабой оптимизации отличие составляет 2...4 %. Для модели винта отличие на величину около 1% сохраняется в течение следующих 60 итераций и только затем резко снижается.

График изменения суммарного угла на развертке в зависимости от количества выполненных шагов алгоритма (итераций) представлен на рис. 10.

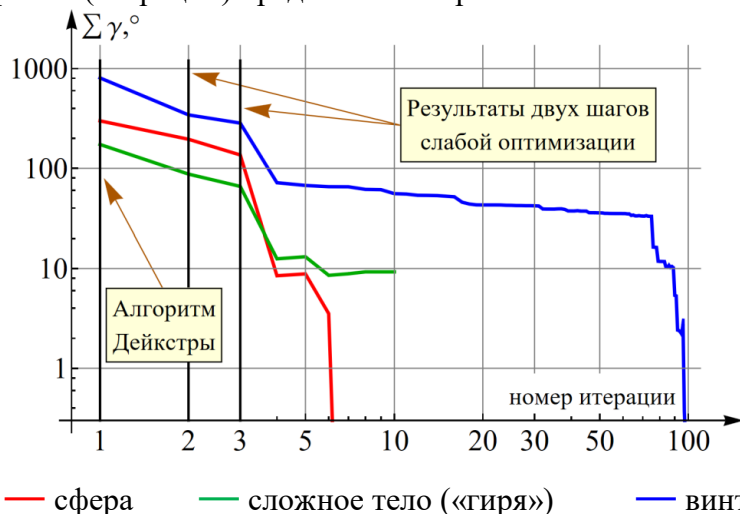


Рис. 10. Изменение суммарного угла (в градусах) на развертке в процессе оптимизации пути

Величина $\sum \gamma$ — это сумма углов (в градусах) между сегментами пути на той развертке, по которой на данном шаге производится вытягивание пути. Видно, что после двух шагов слабой оптимизации суммарный угол уменьшается всего в 2...3 раза, тогда как первый шаг вытягивания пути обеспечивает еще не менее, чем 5-кратное уменьшение суммарного угла (для модели сферы эта величина превышает 20 раз). Далее величина суммарного угла изменяется обычно немонотонно, в ряде случаев при смене развертки наблюдается небольшое ее увеличение. Это не связано с ухудшением самого пути, а объясняется тем фактом, что для одного и того же пути может существовать несколько разверток. Последнее имеет место в тех случаях,

когда путь проходит через «седловую» вершину: в этом случае смена развертки в ней и новая процедура вытягивания приводят к тому же самому пути, но рассматриваемому на другой развертке – т.е. длина пути остается прежней, а угол может и увеличиться (в вычислительном алгоритме реализовано недопущение заикливания в таких ситуациях). Именно эта ситуация и наблюдается во всех рассмотренных примерах; кажущаяся неоптимальность угла на модели гири и винта связана именно с наличием на кратчайшем пути «седловой» вершины.

Заключение. Предложенный алгоритм поиска кратчайшего пути на триангулированной поверхности в рассмотренных модельных задачах позволил построить оптимальные в математическом смысле пути. Алгоритм итерационный; выполнение нескольких первых шагов позволяет существенно улучшить путь. Полная оптимизация пути, состоящего из большого количества сегментов, особенно для неравномерных поверхностных сеток, может требовать выполнения большого числа итераций, что негативно сказывается на времени его исполнения, поэтому задача оптимизации алгоритма остается актуальной. Кроме того, представленный алгоритм реализован в системе компьютерной алгебры Wolfram Mathematica, тогда как для его практического применения требуется создание его эффективной реализации или модификации на языке C++.

Список источников

1. Dergachev S.A., Marchevsky I.K., Shcheglov G.A. Flow simulation around 3D bodies by using Lagrangian vortex loops method with boundary condition satisfaction with respect to tangential velocity components. *Aerospace science and technology*, 2019, vol. 94, DOI: 10.1016/j.ast.2019.105374
2. Люстерник Л.А. Кратчайшие линии. Вариационные задачи. – М.: ГИТТЛ, 1955. – Вып. 19. – 104 с
3. Dijkstra E.W. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1959, vol. 1, pp. 269-271, DOI:10.1007/BF01386390.
4. Hwang Y.H., Chang R.C., Tu H.Y. Finding all shortest path edge sequences on a convex polyhedron. *Lecture notes in computer science. Workshop on algorithms and data structures*, 1989, vol. 382, pp. 251-266, DOI:10.1007/3-540-51542-9_23.
5. Chen J., Han Y. Shortest paths on a polyhedron. *Proc. of 6th ACM symp. on comput. Geometry. Berkley*, 1990, pp. 360-369, DOI:10.1145/98524.98601.
6. Wenk C., Cook A.F. Shortest path problems on a polyhedral surface. *Computational geometry. Dagstuhl seminar proceedings*, 2009, vol. 9111, pp. 1-30, DOI: 10.4230/DagSemProc.09111.5.
7. An P.T., Van Hoai T., Ba Thinh V. The funnel tree algorithm for finding shortest paths on polyhedral surfaces. *Optimization*, 2023, DOI:10.1080/02331934.2023.2241496
8. An P.T., Le N.T., Trang L.H., Wong R.C.-W. Finding shortest gentle paths on polyhedral terrains by the method of multiple shooting. *Journal of computational science*, 2023, vol. 67, DOI:10.1016/j.jocs.2022.101935
9. CGAL: The Computational geometry algorithms library. Available at: <https://www.cgal.org> (accessed: 04/10/2024)
10. Xin S.-Q., Wang G.-J. Improving Chen and Han's algorithm on the discrete geodesic problem. *ACM trans. Graph.* 2009, vol. 28, no. 4, DOI: 10.1145/1559755.1559761.

Серебровская Екатерина Александровна. Студент кафедры Математическое моделирование МГТУ им. Н.Э. Баумана, e.a.serebrovskaya@gmail.com, 105005, Москва, 2-я ул., д. Бауманская 5, к. 1.

Марчевский Илья Константинович. Доцент, д.ф.-м.н., МГТУ им. Н.Э. Баумана, профессор кафедры Прикладная математика, AuthorID: 183544, SPIN: 8988-9507, ORCID: 0000-0003-4899-4828, iliamarchevsky@mail.ru, 105005, Москва, 2-я Бауманская ул., д. 5, к. 1.

Ерофеева Мария Александровна. Студент кафедры Прикладная математика МГТУ им. Н.Э. Баумана, maria.a.erofeeva@gmail.com, 105005, Москва, 2-я Бауманская ул., д. 5, к. 1.

UDC 519.6

DOI:10.25729/ESI.2024.34.2.009

Path smoothing algorithm for the triangulated surface

Ekaterina A. Serebrovskaya, Ilya K. Marchevsky, Mariya A. Erofeeva

Bauman Moscow State Technical University,

Russia, Moscow, *e.a.serebrovskaya@gmail.com*

Abstract. An algorithm for optimizing the path length on a triangulated surface has been proposed and implemented in Wolfram Mathematica. The first two steps are “lightweight” and involve varying the path along the edges only within the triangles adjacent to them. Subsequent steps allow arriving at a shortest path in the mathematical sense in a few iterations. The convergence of the algorithm has not been rigorously proven, but has been ensured in a large number of examples considered.

Keywords: shortest path, triangulated surface, conditional optimization, Wolfram Mathematica

References

1. Dergachev S.A., Marchevsky I.K., Shcheglov G.A. Flow simulation around 3D bodies by using Lagrangian vortex loops method with boundary condition satisfaction with respect to tangential velocity components. *Aerospace science and technology*, 2019, vol. 94, DOI: 10.1016/j.ast.2019.105374
2. Lyusternik L.A. *Kratchayshiye linii. Variatsionnyye zadachi. [Shortest Paths. Variational Problems]*. M.: GITTL, 1964. 102 p.
3. Dijkstra E.W. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1959, vol. 1, pp. 269-271, DOI:10.1007/BF01386390.
4. Hwang Y.H., Chang R.C., Tu H.Y. Finding all shortest path edge sequences on a convex polyhedron. *Lecture notes in computer science. Workshop on algorithms and data structures*, 1989, vol. 382, pp. 251-266, DOI:10.1007/3-540-51542-9_23.
5. Chen J., Han Y. Shortest paths on a polyhedron. *Proc. of 6th ACM symp. on comput. Geometry. Berkley*, 1990, pp. 360-369, DOI:10.1145/98524.98601.
6. Wenk C., Cook A.F. Shortest path problems on a polyhedral surface. *Computational geometry. Dagstuhl seminar proceedings*, 2009, vol. 9111, pp. 1-30, DOI: 10.4230/DagSemProc.09111.5.
7. An P.T., Van Hoai T., Ba Thinh V. The funnel tree algorithm for finding shortest paths on polyhedral surfaces. *Optimization*, 2023, DOI:10.1080/02331934.2023.2241496
8. An P.T., Le N.T., Trang L.H., Wong R.C.-W. Finding shortest gentle paths on polyhedral terrains by the method of multiple shooting. *Journal of computational science*, 2023, vol. 67, DOI:10.1016/j.jocs.2022.101935
9. CGAL: The Computational geometry algorithms library. Available at: <https://www.cgal.org> (accessed: 04/10/2024)
10. Xin S.-Q., Wang G.-J. Improving Chen and Han’s algorithm on the discrete geodesic problem. *ACM trans. Graph.* 2009, vol. 28, no. 4, DOI: 10.1145/1559755.1559761.

Serebrovskaya Ekaterina Alexandrovna. Student of the department of Mathematical modeling, Bauman Moscow state technical university, *e.a.serebrovskaya@gmail.com*, 105005, Moscow, 2nd Baumanskaya St., 5, building 1.

Marchevsky Ilya Konstantinovich. Ph.D., D.Sc. (phys.-math.), associate professor, professor of the department of applied mathematics, Bauman Moscow state technical university, AuthorID: 183544, SPIN: 8988-9507, ORCID: 0000-0003-4899-4828, *iliamarchevsky@mail.ru*, 105005, Moscow, 2nd Baumanskaya St., 5, building 1.

Erofeeva Mariya Alexandrovna. Student of the department of applied mathematics, Bauman Moscow state technical university, *mariya.a.erofeeva@gmail.com*, 105005, Moscow, 2nd Baumanskaya St., 5, building 1.

Статья поступила в редакцию 15.04.2024; одобрена после рецензирования 06.06.2024; принята к публикации 06.06.2024.

The article was submitted 04/15/2024; approved after reviewing 06/06/2024; accepted for publication 06/06/2024.