**Алгоритм поиска кратчайшего пути на триангулированной поверхности**

Серебровская Е.А., Марчевский И.К., Ерофеева М.А.

**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.

**1. Введение.** 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. 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» поверхности неприменим аппарат дифференциального исчисления, и для решения поставленной задачи следует применять иные подходы [1–5].

**2. 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 (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, проходящий 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).

|  |  |
| --- | --- |
|  | Развертка пути: |
| Рисунок 1. – Путь, найденный при помощи алгоритма Дейкстры, и его развертка |

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 «достаточно 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 по вершине. 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); мы будем рассматривать процедуру поиска локально-кратчайшего пути, используя в качестве начального приближения результат работы алгоритма Дейкстры.

|  |  |
| --- | --- |
|  | Развертка глобально-кратчайшего пути: |
| Развертка локально-кратчайшего пути: |

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

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.

|  |  |  |
| --- | --- | --- |
|  | Фрагмент пути: | Фрагмент развертки: |
|  |  |

Рисунок 3. – Кратчайший путь в окрестности седловой точки

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.

Прежде чем описать предлагаемый способ решения задачи, отметим, что в хорошо известной библиотеке вычислительной геометрии CGAL имеется модуль Triangulated Surface Mesh Shortest Paths, позволяющий решить задачу поиска глобально-кратчайшего пути на триангулированной поверхности. 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 пути от выбранной точки на поверхности до множества других точек, причем даже значительное увеличение числа «финальных» точек приводит к весьма малому росту вычислительной сложности. 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.

**3. 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, которые представляются довольно очевидными, но их описание заняло бы слишком много места и едва ли способствовало пониманию сути алгоритма.

1. Построить используя алгоритм Дейкстры кратчайший путь на графе – каркасной модели поверхности; он будет иметь вид , где  – номера (индексы) вершин многогранника.

2. Разбить найденный путь на тройки ,  и т.д.; для каждой такой тройки рассмотреть треугольники, имеющие общей вершиной точку с четным индексом  (их совокупность будем называть словом «шатер»). Для каждого шатра рассмотреть две его части, ограниченные ребрами  и ; построить их развертки. Если угол при вершине  (штрих обозначает образ вершины на развертке) меньше развернутого, то кратчайший путь на этом «полушатре» – прямая, соединяющая точки  и , которая не будет проходить через . Если углы на развертках обоих полушатров меньше развернутого, то из двух путей следует выбрать тот, что короче. Если, наоборот, на обеих развертках углы при  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 , при этом положение каждой «промежуточной» точки  удобно задавать, указывая вершины ребра, на котором она лежит, и весовой коэффициент: , где вершины  являются концами ребер шатра с вершиной , которые пересекает путь; , нулевое и единичное значения веса говорят о совпадении точки пути с вершиной  или  соответственно.

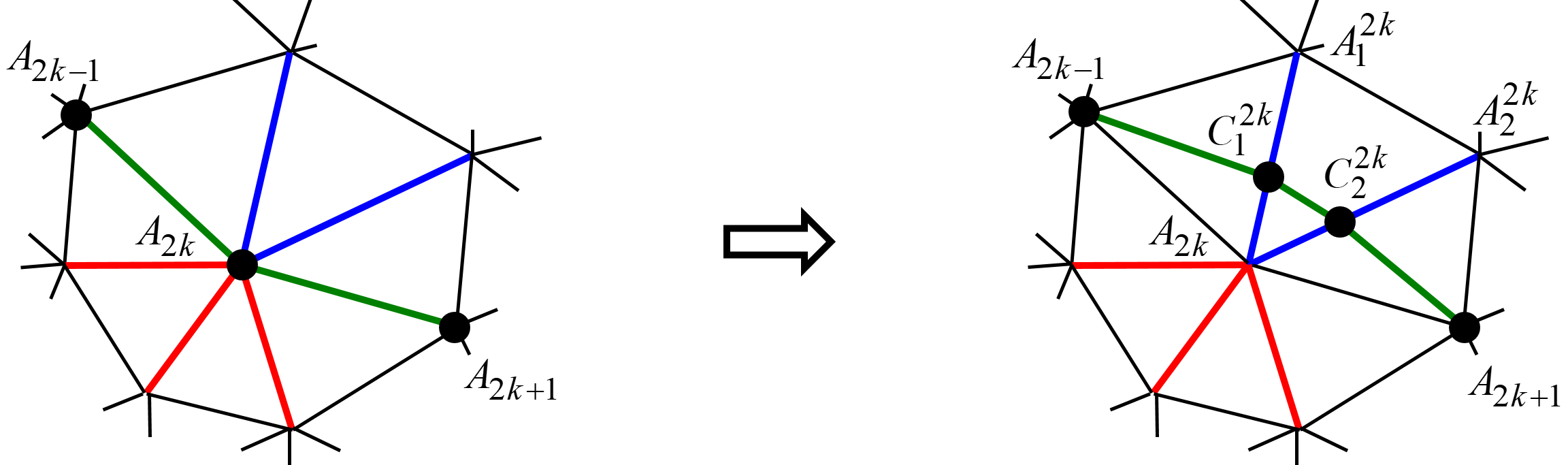


Рисунок 4. – Оптимизация на шатре с четным индексом вершины

4. Проделать аналогичную операцию, но рассматривая шатры с вершинами в точках с нечетными индексами ; в этом случае «разрез» на полушатры будет производиться не по ребрам шатра, а по линиям  и , проходящим в общем случае по граням шатра (рис. 5).

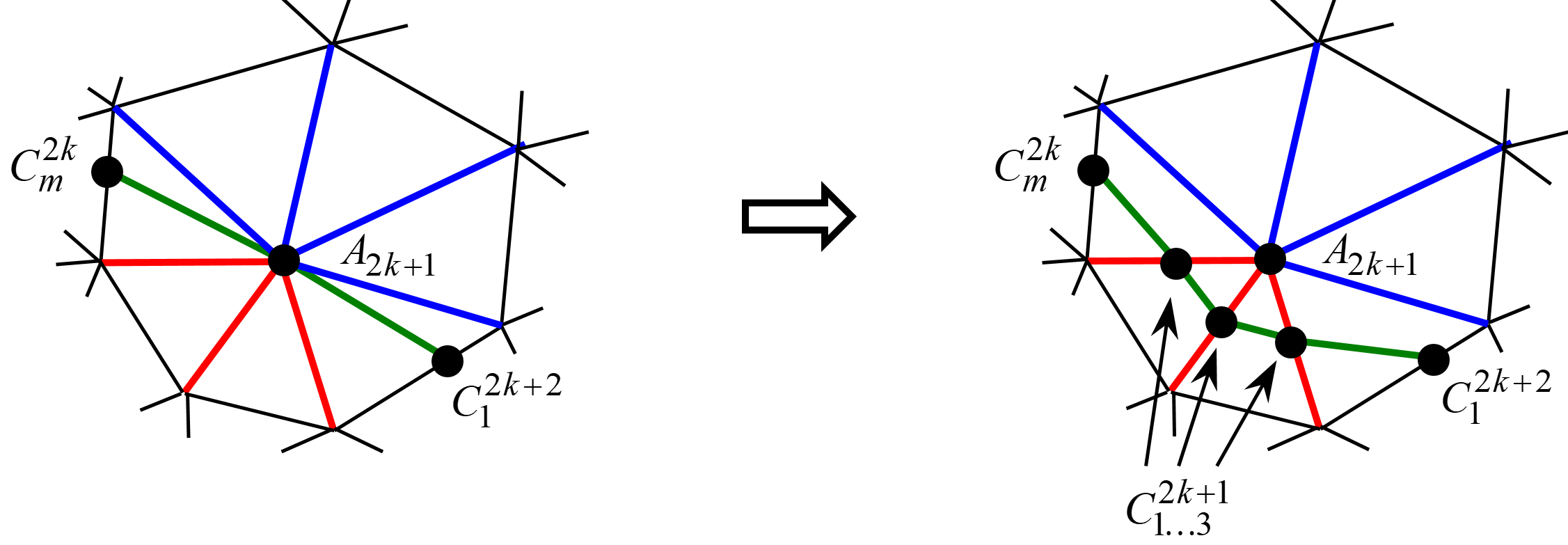


Рисунок 5. – Оптимизация на шатре с нечетным индексом вершины

В итоге полученный путь будет образован точками , а каждая из точек  – задана тройкой указанного в предыдущем пункте вида.

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.

5. «Вытянуть» путь. Для этого нужно построить функцию длины пути на соответствующей развертке, считая веса  переменными (их количество равно числу внутренних точек на пути), и отыскать ее условный минимум в гиперкубе . Несмотря на простоту постановки, эта задача нетривиальна в тех ситуациях, когда некоторые из параметров  выходят на ограничения: градиент функции длины пути в этом случае имеет неопределенность 0/0, а компоненты матрицы Гессе становятся бесконечными. Причина состоит в том, что выход весового параметра на ограничение означает приближение 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  и, следовательно, равенству нулю длины участка пути между этими точками. Механическая аналогия данного шага весьма наглядна: 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 (рис. 7).

6. «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 целесообразно сделать шаг слабой оптимизации, развернув «новый» полушатер и проложив по нему путь, чтобы он образовал отрезок прямой.

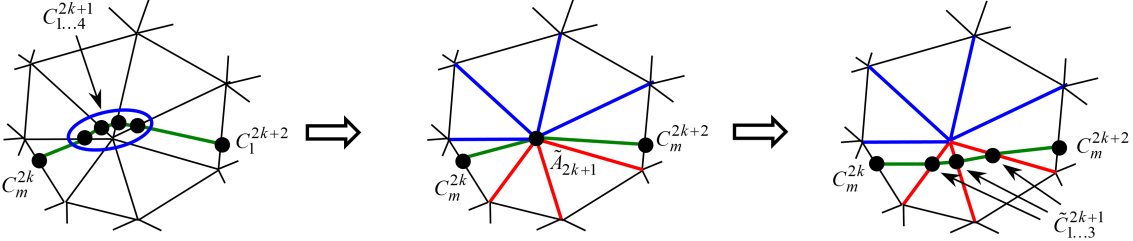


Рисунок 6. – Схема смены развертки

Отметим, что во избежание зацикливания 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.

|  |  |
| --- | --- |
|  | ― алгоритм Дейкстры  ― слабая оптимизация  ― вытянутый путь  Стрелки обозначают вершины шатров, по которым происходит смена развертки; окружностями обведены соответствующие полушатры |

Рисунок 7. – Несколько первых шагов оптимизации пути

**4. 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).

|  |  |  |
| --- | --- | --- |
|  |  |  |
| ― алгоритм Дейкстры ― слабая оптимизация ― кратчайший путь | | |

Рисунок 8. – Результаты оптимизации пути

Путь, найденный с помощью алгоритма Дейскстры, 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 выполнено 4, 7 и 99 итераций вытягивания пути. График изменения длины пути показан на рис. 9.

|  |
| --- |
|  |
| ― сфера ― сложное тело («гиря») ― винт |

Рисунок 9. – Изменение длины в процессе оптимизации пути

Величина  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), представленный на рис. 10, более информативен.

|  |
| --- |
|  |
| ― сфера ― сложное тело («гиря») ― винт |

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

Величина  ­– это сумма углов 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 зацикливания 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.

**5. 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 С++.

**Список литературы**

1. Mitchell J.S.B., Mount D.M., Papadimitriou C.H. The discrete geodesic problem // SIAM J. Comput. 1987. Vol. 16, No. 4, Pp. 647–668.

2. Kapoor S. Efficient computation of geodesic shortest paths // Proc. of 31st ACM Symp. on Theory Computing. Atlanta, 1999. Pp. 770–779.

3. Chen J., Han Y. Shortest paths on a polyhedron // Proc. of 6th ACM Symp. on Comput. Geometry. Berkley, 1990. Pp. 360–369.

4. Marchevsky I.K., Shcheglov G.A., Dergachev S.A. On the algorithms for vertex element evolution modelling in 3D Lagrangian vortex loops method // Topical problems of fluid mech. Prague, 2020, pp. 152–159.

5. Gumirova A., Marchevsky I., Safronov Y. The algorithm of the path length optimization on the polyhedron surface // Lecture Notes in Mech. Engineering. Advanced Problem in Mechanics III. 2023. Pp. 55–68.

**КРАТКИЕ БИОГРАФИЧЕСКИЕ СВЕДЕНИЯ**

**1** Серебровская Екатерина Александровна, студентка кафедры «Математическое моделирование» ФГБОУ ВО «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)», Россия, Москва,   
E-mail: e.a.serebrovskaya@gmail.com

**2** Марчевский Илья Константинович, д.ф.-м.н., доцент, профессор кафедры «Прикладная математика» ФГБОУ ВО «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)», Россия, Москва, E-mail: iliamarchevsky@mail.ru

**3** Ерофеева Мария Александровна, студентка кафедры «Прикладная математика» ФГБОУ ВО «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)», Россия, Москва,   
E-mail: mariya.a.erofeeva@gmail.com