

УДК 000.00

ПРИМЕНЕНИЕ ПРОЕКТИВНОГО ПОКООРИНАТНОГО СПУСКА В ЗАДАЧЕ ФЕКЕТЕ

И.Ф. Фатхуллин

Московский физико-технический институт

Россия, 141700, Долгопрудный, Институтский пер., 9

E-mail: ilyas.fn979@gmail.com

Ключевые слова: задачи Томсона, Фекете, градиентный спуск, покоординатный спуск, оценка скорости сходимости, быстрый счет функции

Аннотация: Мы исследуем задачу минимизации энергии системы из N точек, с ограничением на поверхности сферы в \mathbf{R}^3 , и взаимодействующих с потенциалом $U = \frac{1}{r^s}$, $s > 0$, где r - евклидово расстояние между парой точек. Мы рассматриваем алгоритм проективного покоординатного спуска, а также предлагаем метод, использующий гессиан минимизируемой функции. Цель - предложить алгоритм оптимизации для данной задачи, который достаточно быстро приближается к известным из литературы минимальным значениям.

1. Введение

Пусть дано натуральное $N \geq 2$ и действительное $s > 0$. Задача Фекете [1] формулируется следующим образом. Найти множество точек $\omega_N = \{x_1, \dots, x_N\}$ с ограничением на единичной сфере $S^2 = \{x \in \mathbf{R}^3 : \|x\|_2 = 1\}$, такое, что оно минимизирует потенциальную энергию системы:

$$(1) \quad E = E(s, \omega_N) = \frac{1}{2} \sum_{i,j=1, i \neq j}^N \frac{1}{\|x_i - x_j\|_2^s}$$

При $s = 1$ получается Кулоновский потенциал и, соответственно, задача Томсона [2]. Последняя задача возникла у Дж. Дж. Томсона в 1904 году при изучении строения атома. Несмотря на то, что от модели атома Томсона вскоре отказались, поставленная задача нашла множество применений в физике, химии и биофизике. Задача Фекете и, в частности, Томсона привлекла особое внимание математиков, потому что аналогичная задача (только с логарифмическим ядром) появляется в списке Смейла - нерешенных задач для 21-го века. Учитывая широкое применение, задача Томсона была хорошо изучена [3, 4]. Известны аналитические решения для $N = 2 - 6$ и 12. Для больших N получены решения с помощью численных методов. В данной статье мы остановимся на обсуждении проективного градиентного метода, использующего быстрый счет функции и итеративный выбор шага. Оценим скорость сходимости

данных алгоритмов, а также обсудим возможности использования методов второго порядка.

2. Покоординатный спуск

Определение точек Фекете на поверхности единичной сферы есть невыпуклая нелинейная задача оптимизации с нелинейными ограничениями. Следует отметить, что здесь можно легко избавиться от ограничений, параметризуя каждую точку на сфере двумя углами [5], и решать задачу нелинейной оптимизации без ограничений от $2N$ переменных вместо $3N$. Однако на практике применение градиентных методов в таких координатах существенно замедляет счет оптимизируемой функции и градиента, поэтому будем рассматривать задачу в исходных координатах. Можно было бы применять градиентные методы, рассматривая зависимость энергии от всех $3N$ переменных, и на каждой итерации перемещать все точки. Однако в таком подходе уже при $N \sim 10^4$ (особенно в случае использования методов второго порядка) у современного персонального компьютера начинаются проблемы с памятью. Поэтому остановимся подробнее на методе покоординатного спуска с ограничениями. В отличие от [4] здесь координата точки сразу после шага проектируется на поверхность сферы, а не раскладывается на нормальную и касательную составляющие.

2.1. Описание метода

Зафиксируем положения $N - 1$ точек (без i -ой) на единичной сфере. Тогда, как известно, $F_i = -\nabla_i E$, где градиент берется по компонентам вектора $x_i \in \mathbf{R}^3$, есть направление невозрастания функции. Сделаем шаг в данном направлении и отнормируем координату i -ой точки, чтобы удовлетворить ограничениям:

$$(2) \quad x_i^n = \frac{x_i^{n-1} + \alpha_n F_i}{\|x_i^{n-1} + \alpha_n F_i\|}$$

Выбор шага α_n зависит от взаимного расположения точек на n -ой итерации. Остановимся подробнее на методе выбора α_n в соответствующем разделе, а здесь опишем структуру алгоритма:

1. Выбрать начальное приближение ω_N^0
 2. Пока не достигнут критерий останова:
 - 2.1. Вычислить шаг α_n
 - 2.2. Выбрать индекс i точки, которую будем двигать
 - 2.3. Вычислить $F_i = -\nabla_i E$
 - 2.4. Сделать шаг (2), обновив i -ую компоненту ω_N^n
- Индекс i на шаге 2.2 выбираем равным по модулю номеру итерации.

2.2. Выбор начального положения точек ω_N^0

Выбор начального приближения играет важную роль в сходимости градиентных методов. Один из способов - метод Монте-Карло, в котором каждый элемент ω_N^0 выбирается из равномерного распределения на поверхности сферы. В предельном случае при больших N метод дает равномерную плотность распределения точек на

сфере, однако для стабильности будем использовать начальное положение, называемое generalized spiral set [6].

2.3. Быстрый счет функции и градиента

В силу специфики задачи можно существенно асимптотически ускорить каждую итерацию алгоритма, в отличие от наивного подхода, с помощью быстрого подсчета функции и его градиента. Как видно из (1), для подсчета энергии нам необходимо знать лишь попарные расстояния между точками. Поэтому будем хранить в памяти матрицу расстояний. Теперь на каждой итерации (для i -ой точки) будем обновлять энергию следующим образом: вычтем $N - 1$ слагаемое (те которые содержат индекс i) до выполнения шага, а затем прибавим соответствующие слагаемые после выполнения шага. Таким образом, если положить время вычисления одного слагаемого в (1) равным τ , то получим время вычисления E на каждой итерации порядка $2N\tau$ вместо $N^2\tau$. Далее поступим аналогично с градиентом, который можно вычислить аналитически:

$$(3) \quad F_k = \sum_{i=1, i \neq k}^N \frac{s}{r_{ik}^{s+2}} (x_k - x_i)$$

где $r_{ik} := \|x_i - x_k\|$

Будем хранить матрицу, содержащую слагаемые вида $\frac{s}{r_{ik}^{s+2}} (x_k - x_i)$ для $i, k = 1, \dots, N, i \neq k$, а также их сумму по i . Пусть на предыдущей итерации двигалась точка с индексом j , тогда для подсчета градиента на данной итерации i достаточно из F_k вычесть слагаемое $\frac{s}{r_{jk}^{s+2}} (x_k - x_j)$ и прибавить $\frac{s}{r_{ik}^{s+2}} (x_k - x_i)$, что снова дает выигрыш по времени в N раз на каждой итерации.

2.4. Выбор шага

На метод покоординатного спуска с ограничением можно посмотреть как на задачу минимизации при фиксированных $N - 1$ точке:

$$(4) \quad x_k^{n+1} = \arg \min_{y \in S^2} \{E(x_k^n) + (\nabla_k E(x_k^n), y - x_k^n) + \frac{\tilde{L}}{2} \|y - x_k^n\|^2\}$$

где \tilde{L} - константа Липшица функции $\nabla_k E(x_k) = -F_k(x_k)$.

Однако легко видеть, что функция F_k не липшицева даже относительно переменных x_k . Если бы нам удалось ограничить \tilde{L} некоторой константой L в процессе алгоритма, то можно было бы явно выписать решение задачи (4):

$$(5) \quad x_k^{n+1} = \frac{x_k^n + (1/L)F_k(x_k)}{\|x_k^n + (1/L)F_k(x_k)\|}$$

и определить в (2) шаг $\alpha = 1/L$. Попробуем оценить константу L , используя градиент (3) в окрестности «хороших» приближений. Например, из свойств начального приближения generalized spiral set [6] ясно, что точки находятся на расстоянии $\delta > 2/(N - 1)$, тогда можно получить оценку:

$$(6) \quad L \leq \max_{x \in S^2} |\nabla_k^2 E(x)| \leq s(N - 1) \frac{(N - 1)^{s+2}}{2^{s+2}} (s + 2 + \sqrt{3}) < \frac{s(s + 2 + \sqrt{3})}{2^{s+2}} (N - 1)^{s+3}$$

Оказывается, что такая оценка для δ достаточно груба и соответственно выбранный шаг слишком мал. Поэтому предлагается оценивать константу L на каждой итерации трехмерным гессианом (7), для посчета которой на каждой итерации требуется константное время, если следовать стратегии быстрого счета как в разделе 2.3.

$$(7) \quad \nabla_k^2 E(x^n) = s \sum_{i=1, i \neq k} \frac{-I r_{ik}^2 + (s+2)(x_k - x_i)^T (x_k - x_i)}{r_{ik}^{s+4}}$$

3. Результаты

В разделе 2.3. описан способ быстрого счета функции и градиента. Предположительно время выполнения алгоритма при фиксированном количестве итераций определяется линейными членами по количеству точек N . Рис. 1 показывает зависимость реального времени t выполнения алгоритма (в секундах) от количества точек N при фиксированном количестве итераций.

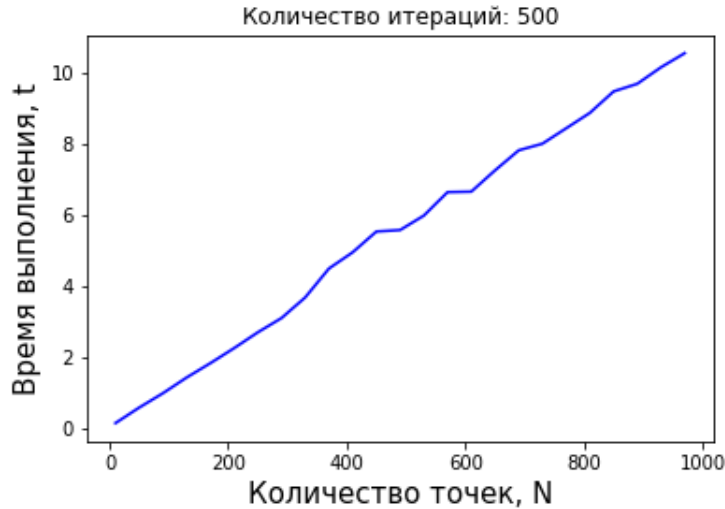


Рис. 1. линейное время выполнения

В градиентных методах стратегия выбора шага, которая обсуждалась в разделе 2.4., определяет скорость сходимости алгоритма по количеству итераций. Мы ожидаем получить сходимость оптимизируемой функции энергии E со скоростью геометрической прогрессии с параметрами a и b :

$$(8) \quad E(n) = aE(n-1) + b$$

или

$$(9) \quad E(n) = a^n E(0) + b \cdot \frac{1 - a^{n-1}}{1 - a}$$

Для подбора параметров a , b и $E(0)$ используется нелинейный метод наименьших квадратов [7]. Графики Рис. 2-5 показывают скорость сходимости метода относительно функции (1). Параметры МНК и минимально найденное значение функции $E(N)$

для различных N указаны в таблице 1. В четвертом столбце рассчитано предельное значение аппроксимации $\frac{b}{1-a}$, к которому асимптотически стремится модель 8. Сравнивая последние два столбца таблицы 1, можно заметить, что алгоритм достаточно близко приближается к минимально известным значениям. Различие: $10^{-4} - 10^{-2}\%$. Результаты многочисленных экспериментов показывают, что данный метод сходится для $N < 1700$. На диаграммах Вороного Рис. 6 и 7 можно заметить насколько улучшилась конфигурация точек при $s = 1, N = 300$ после выполнения 3000 итераций алгоритма. На диаграмме синим обозначены положения точек на единичной сфере, красным - пятиугольники, зеленым - шестиугольники, черным - семиугольники.

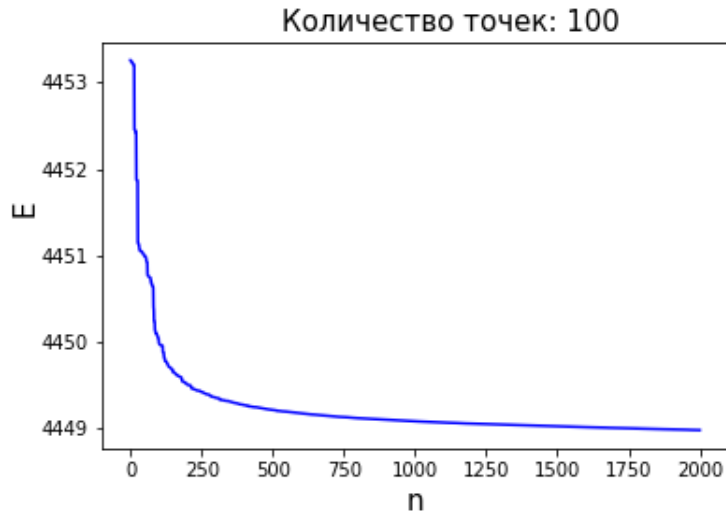


Рис. 2. скорость сходимости



Рис. 3. скорость сходимости

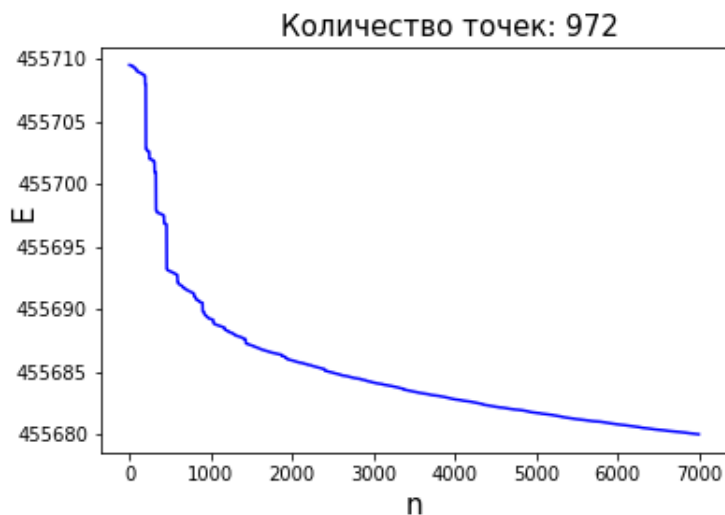


Рис. 4. скорость сходимости

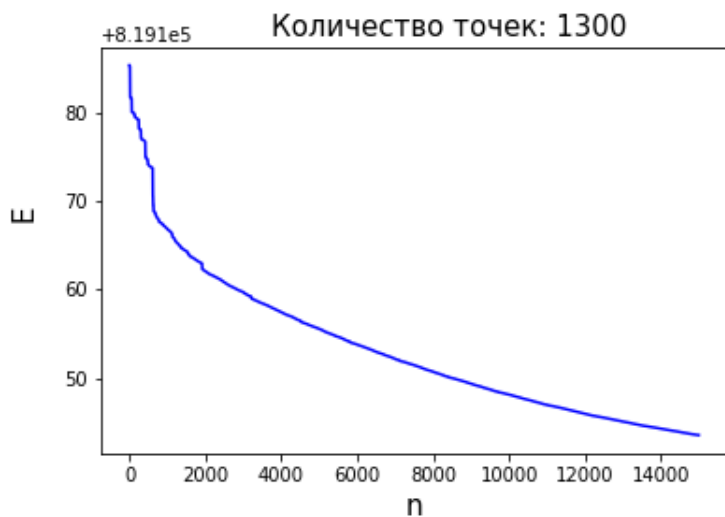


Рис. 5. скорость сходимости

Таблица 1. Параметры МНК

N	a	b	$\frac{b}{1-a}$	$E(N)$	E_{min}
100	0.9869	58.18	4449.10	4448.98	4448.35
470	0.9978	234.97	104831.83	104831.20	104822.89
972	0.9988	526.08	455681.97	455680.04	455651.08
1300	0.9997	279.55	819146.96	819143.51	-

$E(N)$ - результат работы алгоритма, E_{min} - минимально известное значение из литературы [8]

4. Метод второго порядка

Из таблицы 1 можно заметить, что параметр a асимптотически приближается к единице, что означает более медленную сходимость для больших N . С целью уве-

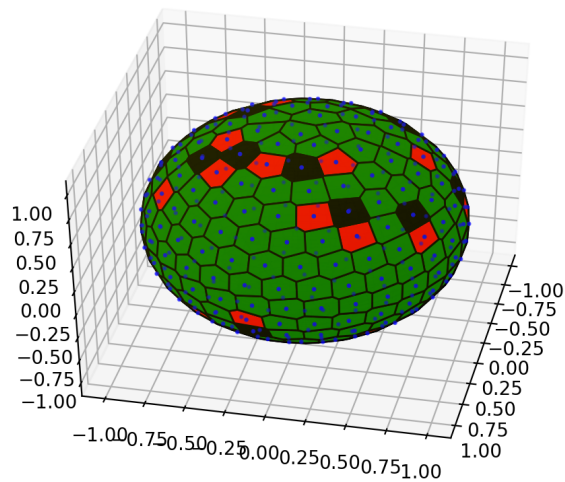


Рис. 6. диаграмма Вороного для начального положения

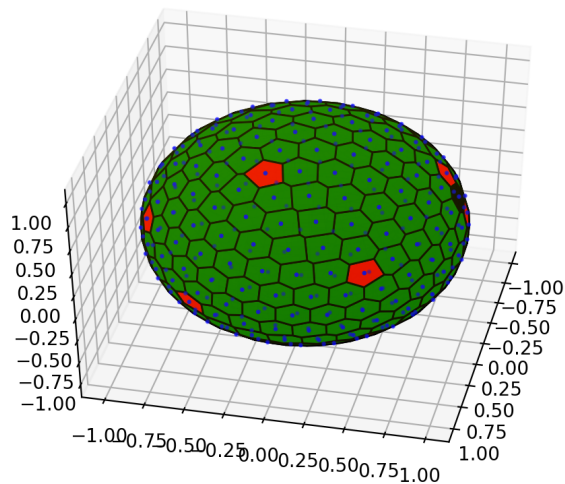


Рис. 7. диаграмма Вороного после оптимизации

личить скорость сходимости можно использовать гессиан функции по переменным $x_i \in \mathbf{R}^3$. Далее на каждой итерации минимизируется квадратичное приближение функции $E(x_i)$ с ограничением на поверхности сферы. К сожалению, применения данного метода еще не дали стабильных положительных результатов.

5. Заключение

В данной работе мы обсудили детали реализации алгоритма покоординатного спуска с ограничением. Показали его вычислительную экономичность для различных N , экспериментально исследовали скорость сходимости и сравнили с известными значениями из литературы. В будущем планируется более детально исследовать сходимость последнего метода, а также обобщить результаты на другие виды поверхностей и функций.

Список литературы

1. M. Fekete. Über die Verteilung der Wurzeln bei gewissen algebraischen Gleichungen mit ganzzahligen Koeffizienten // Mathematische Zeitschrift. Vol. 17, Issue 1, P. 228–249. 1923.
2. of oscillation of a number of corpuscles arranged at equal intervals around the circumference of a circle; with application of the results to the theory of atomic structure // Philosophical Magazine Series 6. 1904. 7:39, 237 — 265.
3. D. J. Wales, H. McKay, E. L. Altschuler. Defect motifs for spherical topologies // Phys. Rev. B. 2009. 79, 224115.
4. E. Bendito, A. Carmona, A. M. Encinas, J. M. Gesto. Estimation of Fekete Points // Journal of Computational Physics. 2007. Vol. 225, Issue 2, P. 2354–2376. .
5. K. J. Nurmela. Constructing spherical codes by global optimization methods // Helsinki University of Technology. Series A: Research Reports. 1995. 32.
6. E. B. Saff, A. B. J. Kuijlaars. Distributing Many Points on a Sphere // The Mathematical Intelligencer. 1997. Vol. 19, Issue 1, P. 5–11.
7. Non-Linear Least-Squares Minimization and Curve-Fitting for Python. <https://lmfit.github.io/lmfit-py/index.html>.
8. D. J. Wales, S. Ulker. Structure and dynamics of spherical crystals characterized for the Thomson problem // Physical Review B. 2006. 74, 212101.