

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

МОСКОВСКИЙ ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ
(государственный университет)

ФАКУЛЬТЕТ УПРАВЛЕНИЯ И ПРИКЛАДНОЙ МАТЕМАТИКИ
Кафедра математического моделирования и прикладной математики

Базовая организация:

Институт прикладной математики им. М.В. Келдыша РАН

Квалификационная работа на соискание степени бакалавра
по направлению 03.03.01 «Прикладная математика и физика»

Глобальная оптимизация траекторий космических аппаратов с помощью кривых, заполняющих пространство

Выполнил:

студент группы 472

Худайбердиев Юсуф Алишер угли

Научный руководитель:

к.ф.-м.н.

Трофимов Сергей Павлович

Москва, 2018

Оглавление

Введение	3
Глава 1. Оптимизация перелетов с идеально регулируемым двигателем	8
1.1. Постановка модельной задачи. Уравнения движения	8
1.2. Принцип максимума Понтрягина	10
1.3. Необходимые условия оптимальности перелета. Краевая задача принципа максимума	14
Глава 2. Решение краевой задачи принципа максимума методами глобальной оптимизации	17
2.1. Сведение задачи многомерной оптимизации к задаче одномерной оптимизации	17
2.1.1. Постановка задачи глобальной оптимизации	17
2.1.2. Кривые, заполняющие пространство	18
2.1.3. Сведение задачи многомерной глобальной оптимизации к одномерной с помощью кривой Пеано	28
2.2. Алгоритмы одномерной оптимизации	29
2.2.1. Метод Пиявского	30
2.2.2. Геометрический алгоритм	32
2.2.3. Идея для ускорения	37
2.2.4. Связь между многомерной липшицевой функцией и соответствующей ей гёльдеровой функцией	38
Глава 3. Эксперименты и результаты	40
3.1. Оптимизация с помощью кривых Пеано	40
3.2. Глобальный линейный поиск	43

Заключение	50
Список литературы	52

Введение

Космические аппараты (КА) в настоящее время все чаще оснащаются двигателями малой тяги. Благодаря высокому удельному импульсу двигатели малой тяги расходуют меньше топлива, что позволяет увеличить массу выводимой полезной нагрузки, пусть и за счет увеличения продолжительности маневров. Это делает двигатели малой тяги привлекательным средством доставки КА к целям в дальнем космосе [1], но сразу же встает проблема оптимальности перелета: в каких-то случаях необходимо, чтобы маневр был совершен как можно скорее, в каких-то случаях не критично время, но критичны затраты топлива. Решение задачи оптимизации траекторий КА становится все острее с увеличением их распространенности. Эта задача сложна, но существует ряд методов, позволяющих ее решить.

Для оптимизации траектории перелета космического аппарата с одной орбиты на другую используются различные численные методы оптимизации, но специфика задач астродинамики не позволяет использовать многие традиционные методы для решения задачи оптимизации. Эта специфика связана с тем, что функционалы в задачах оптимизации траекторий заданы в многомерном пространстве, существенно нелинейны, многоэкстремальны, могут иметь разрывные экстремали.

Методы оптимизации условно делятся на прямые и непрямые [2]. Прямые методы используют представление траектории и управления в виде конечномерных векторов значений этих вектор-функций в точках-узлах временной сетки. Тем самым задача оптимизации сводится к задаче нелинейного программирования. Непрямые методы опираются на необходимые условия локальной оптимальности. В задачах оптимизации космических траекторий удобно использовать принцип максимума Понтрягина, который в дальнейшем сводится к задаче оптимизации.

К настоящему времени разработано немало методов глобальной оптимизации многоэкстремальных функций, в том числе неградиентных (симплексный метод Нелдера-Мида [3]) и стохастических (метод роя частиц [4]), но эти методы обычно сложны в реализации, а их эффективность сильно зависит от параметров конкретной задачи.

В этой работе будет использоваться непрямой метод, основанный на принципе максимума Понтрягина, благодаря которому можно получить явное выражение для вектора оптимального управления, зависящего от вектора сопряженных переменных. Обычно получающиеся из принципа максимума краевые задачи решаются с помощью модификаций метода Ньютона [5], но сложность этих методов состоит в том, что иногда не удается подобрать достаточно близкое к оптимальному начальное приближение. Основной целью этой работы было исследование на применимость методов, основанных на сведении многомерных задач к одномерным с помощью кривых, заполняющих пространство, таких как кривая Гильберта или кривая Пеано. Но задачи одномерной оптимизации могут быть так же сложны в решении ввиду вида функционалов: они овражистые, сильно нелинейные, область притяжения может быть очень узкой, расчёт значения функционала и его якобиана вычислительно затратен.

В 1877 году Георг Кантор получил поразительный результат, который гласил, что множество точек отрезка имеет ту же мощность, что и множество точек любого конечномерного многообразия, в частности, квадрата, независимо от длины отрезка и размеров квадрата. Он изложил это в своей статье «К учению о многообразиях» [6].

После того, как Кантор получил этот результат, Джузеппе Пеано начал решать задачу о том, может ли кривая заполнить пространство. В 1890 Пеано открыл непрерывную кривую, называемую кривой Пеано, которая проходит через любую точку единичного квадрата [7]. При этом, невозможно существование такого непрерывного взаимнооднозначного

отображения между единичным интервалом и единичным гиперкубом. После открытия Пеано и другие математики, такие как Давид Гильберт [8] и Вацлав Серпинский, предоставили свои варианты построения подобной кривой. Процедуры построения плоских кривых приведены на рисунках 0.1-0.3.

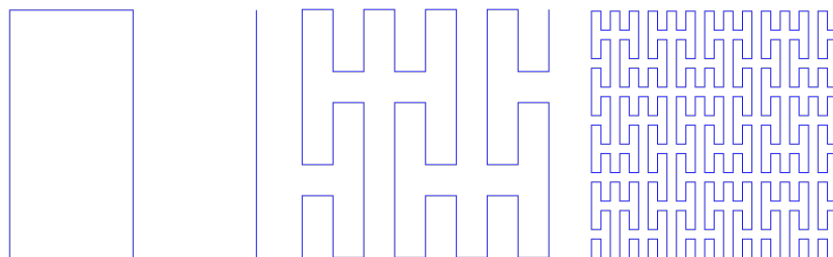


Рис. 0.1. Кривая Пеано на плоскости. ¹

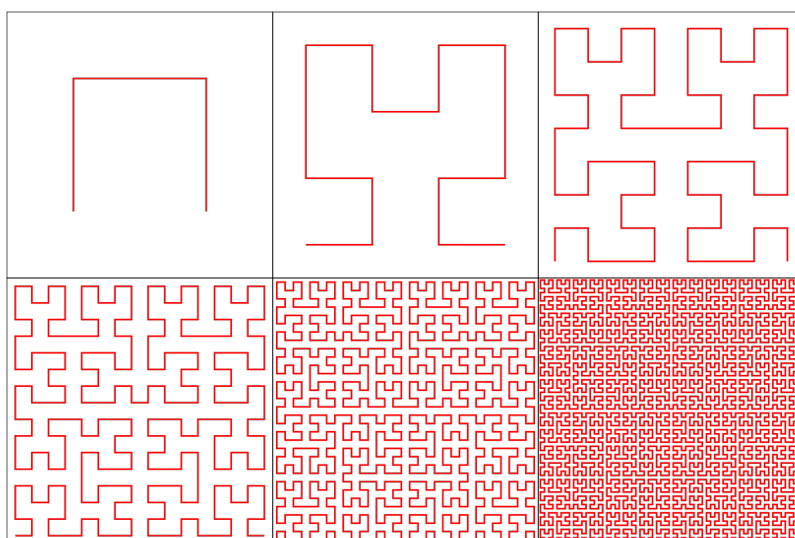


Рис. 0.2. Кривая Гильберта на плоскости. ²

¹ <https://commons.wikimedia.org/w/index.php?curid=2304574>

² <https://commons.wikimedia.org/w/index.php?curid=47570255>

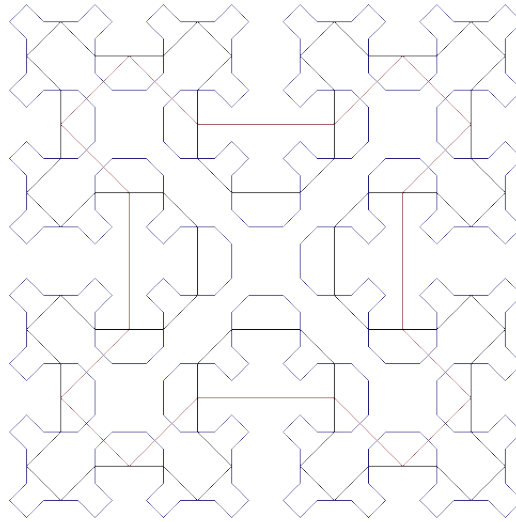
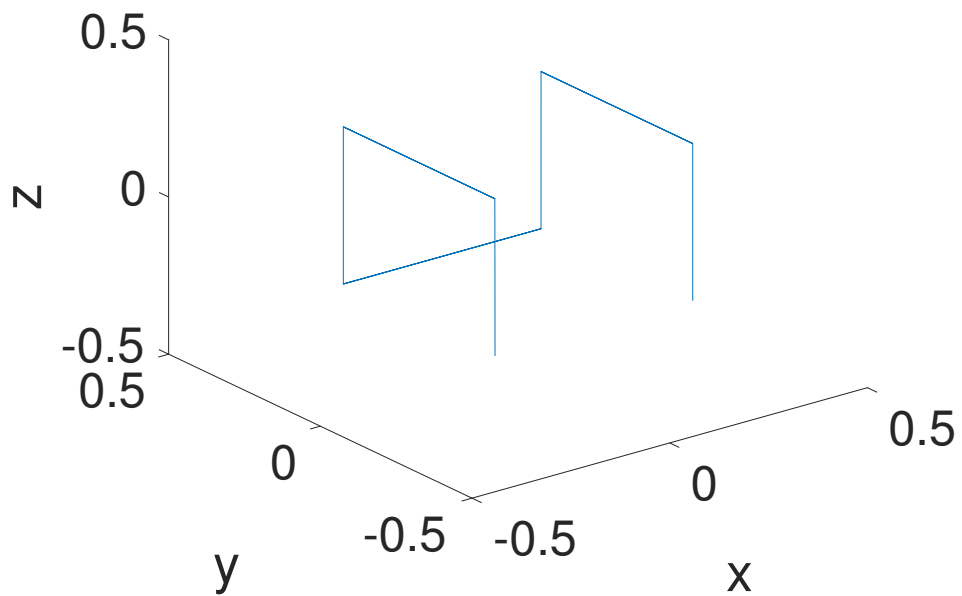


Рис. 0.3. Кривая Серпинского на плоскости.³

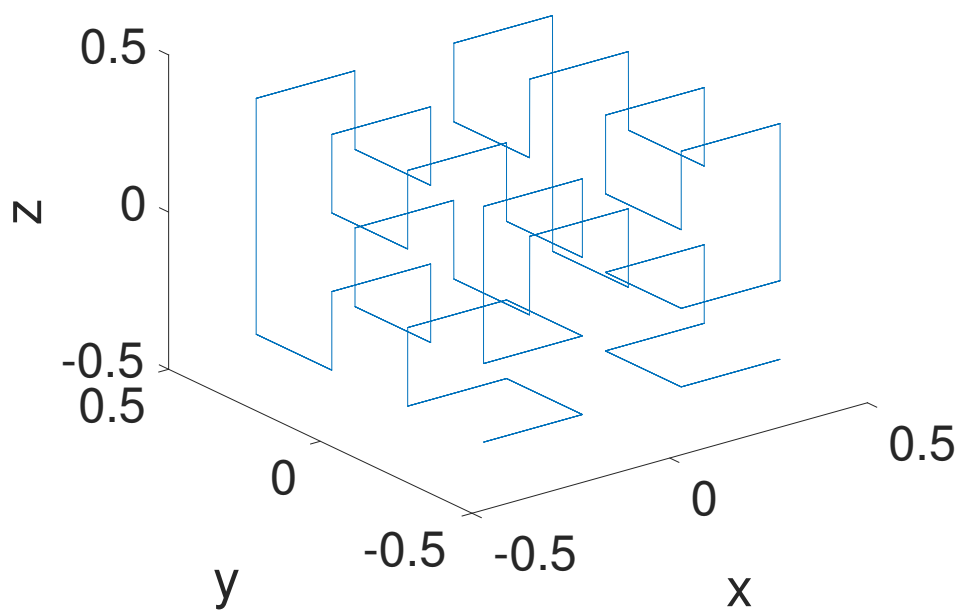
В пространстве \mathbb{R}^3 первых два шага построения кривой Гильберта показаны на рисунке 0.4.

Диссертация имеет следующую структуру: глава 1 посвящена описанию поставленной задачи. В главе 2 будет изложен метод, использованный для решения задачи, и установлена связь между оптимизацией и кривыми Пеано. В главе 3 приводятся результаты расчетов оптимальных траекторий, а в заключении перечисляются полученные выводы.

³ <https://commons.wikimedia.org/w/index.php?curid=208664>



(a) 1 шаг



(б) 2 шага

Рис. 0.4. Построение кривой Гильберта в пространстве.

Оптимизация перелетов с идеально регулируемым двигателем

В этой главе будет сформулирована модельная задача перелета космического аппарата, оснащенного двигателем малой тяги с постоянной мощностью, приведены уравнения движения, а также будут описаны проблемы, связанные с решением этой задачи.

1.1. Постановка модельной задачи. Уравнения движения

Пусть даны круговые компланарные орбиты радиусов R_1 и R_2 , вектор начального положения \mathbf{r}_0 (на орбите отлета), вектор начальной скорости \mathbf{v}_0 , вектор конечного положения \mathbf{r}_f (на орбите прилета), вектор конечной скорости \mathbf{v}_f . Задача состоит в том, чтобы найти траекторию наиболее оптимального по топливу перелета. Иллюстрация задачи приведена на рисунке 1.1 (траектория перелета обозначена дугой).

Уравнения движения космического аппарата (КА) имеют вид

$$\begin{cases} \dot{\mathbf{r}} = \mathbf{v}, \\ \dot{\mathbf{v}} = \mathbf{a}_{\text{тяги}} + \mathbf{a}_{\text{внеш}}, \end{cases} \quad (1.1)$$

где ускорение тяги $\mathbf{a}_{\text{тяги}}$ выражается формулой:

$$\mathbf{a}_{\text{тяги}} = \frac{\mathbf{F}_{\text{тяги}}}{m_{\text{КА}}(t)} = \frac{\dot{m}_{\text{расх}}(t)u}{m_0 - m_{\text{расх}}(t)} \mathbf{e}_{\text{тяги}}.$$

Здесь u – эффективная скорость истечения рабочего тела, $\mathbf{e}_{\text{тяги}}$ – еди-

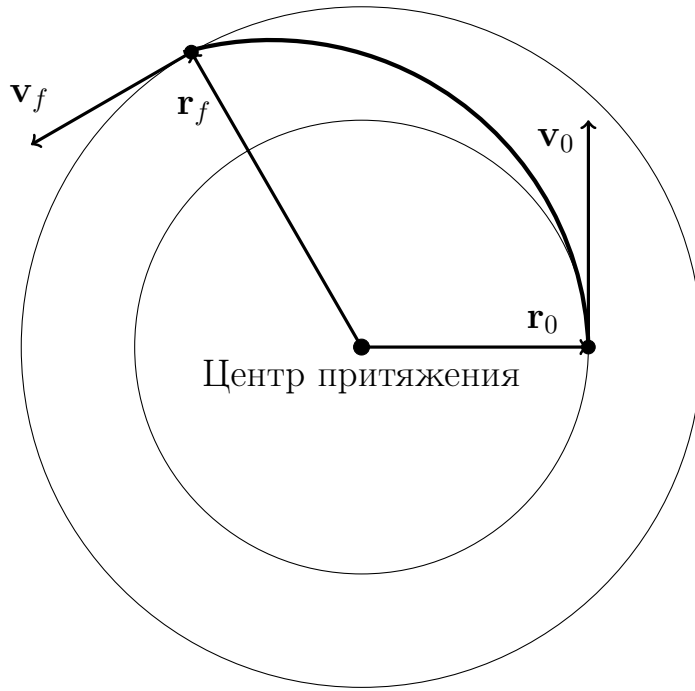


Рис. 1.1. Перелет между круговыми компланарными орбитами.

ничный вектор тяги и

$$\mathbf{a}_{\text{внеш}} = \frac{\mathbf{F}_{\text{внеш}}}{m_{\text{КА}}} = -\mu \frac{\mathbf{r}}{r^3}.$$

Потребуем, чтобы мощность реактивной струи не превышала максимального значения доступной двигателю электрической мощности N_{max} с поправкой на КПД двигателя η :

$$\frac{\dot{m}_{\text{расх}} u^2}{2} \leq N_{\text{max}} \eta. \quad (1.2)$$

Выражая скорость истечения через ускорение тяги получим следующее неравенство:

$$\frac{a_{\text{тяги}}^2 (m_0 - m_{\text{расх}})^2}{2\dot{m}_{\text{расх}}} \leq N_{\text{max}} \eta. \quad (1.3)$$

После интегрирования имеем:

$$\frac{1}{m_0 - m_{\text{расх}}(t)} - \frac{1}{m_0} \geq \frac{1}{2\eta} \int_{t_0}^{t_f} \frac{a_{\text{тяги}}^2(\tau)}{N_{\text{max}}} d\tau. \quad (1.4)$$

Минимальный расход топлива достигается, когда используется вся доступная мощность N_{\max} , то есть когда неравенство (1.4) превращается в равенство и минимизируется целевой функционал

$$J = \frac{1}{2} \int_{t_0}^{t_f} \frac{a_{\text{тяги}}^2(t)}{N_{\max}} dt \rightarrow \min \quad (1.5)$$

Теперь подробнее поговорим про используемые источники энергии. Есть два основных источника энергии: солнечные панели и радиоизотопный термоэлектрический генератор (РИТЭГ). Мощность, которую двигатель получает от солнечных панелей зависит от того, находится ли аппарат в тени, а так же от расстояния до Солнца, а значит, и от времени (так как со временем аппарат передвигается и расстояние зависит от времени). РИТЭГ же лишен этих недостатков и его мощность можно считать постоянной, поэтому для простоты далее полагаем, что источником энергии служит РИТЭГ.

Так как N_{\max} постоянна, то ее можно вынести из-под знака интеграла. Тогда функционал примет вид:

$$J = \frac{1}{2} \int_{t_0}^{t_f} a_{\text{тяги}}^2(t) dt$$

Такая модель электрореактивного двигателя называется идеально регулируемой тягой ограниченной мощности.

1.2. Принцип максимума Понтрягина

Рассмотрим автономную управляемую динамическую систему

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{w}) \quad (1.6)$$

с вектором состояния $\mathbf{x} = (x_1, x_2, \dots, x_n)$ из области $X \subset \mathbb{R}^n$ и вектором управления \mathbf{w} из замкнутого множества допустимых управлений $W \subset \mathbb{R}^m$.

Неавтономную систему можно свести к автономной добавлением времени t к вектору состояния в качестве дополнительной переменной x_{n+1} .

Вектор-функцию $\mathbf{f}(\mathbf{x}, \mathbf{w})$ будем считать непрерывной на $X \times W$, а ее производную $\frac{\partial \mathbf{f}}{\partial \mathbf{x}}$ – непрерывной на X при любом управлении \mathbf{w} из W . От управления $\mathbf{w} = \mathbf{w}(t)$, как функции времени, лишь требуется быть кусочно-непрерывным на интересующем нас интервале времени $t \in [t_0, t_f]$. При таких условиях решение задачи Коши для системы (1.6) существует и единственно.

Для динамической системы (1.6) сформулируем задачу оптимального управления как задачу поиска кривой $\mathbf{w} = \mathbf{w}(t)$, $t \in [t_0, t_f]$ на множестве W , минимизирующей функционал в интегральной форме (форме Лагранжа)

$$J = \int_{t_0}^{t_f} f_0(\mathbf{x}, \mathbf{w}) dt. \quad (1.7)$$

Функцию $f_0(\mathbf{x}, \mathbf{w})$ называют функцией Лагранжа или лагранжианом. К ней будем предъявлять те же требования по непрерывности и гладкости, что и к функциям правых частей системы (1.6).

Граничные условия задачи оптимального управления могут иметь различный вид. В наиболее общем случае начальный и конечный векторы состояния лежат на некоторых дифференцируемых терминальных многообразиях

$$M_0 = \{\mathbf{g}_0(\mathbf{x}(t_0)) = \mathbf{0}\}, M_f = \{\mathbf{g}_f(\mathbf{x}(t_f)) = \mathbf{0}\}.$$

Соответствующая задача носит название задачи со скользящими конца-

ми. В частном случае вырождения этих многообразий в точки получаем задачу с фиксированными концами:

$$\mathbf{x}(t_0) = \mathbf{x}_0, \quad \mathbf{x}(t_f) = \mathbf{x}_f.$$

Систему, задаваемую уравнениями

$$\dot{\mathbf{p}} = - \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right)^T \mathbf{p}, \quad (1.8)$$

называют сопряженной по отношению к системе (1.6), а вектор $\mathbf{p} = (p_1, p_2, \dots, p_n)$ – сопряженным вектором. С помощью сопряженного вектора можно ввести функцию

$$\mathcal{H}(\mathbf{x}, \mathbf{p}, \mathbf{w}) = p_0 f_0 + \mathbf{p} \cdot \mathbf{f} = p_0 f_0 + \sum_{i=1}^n p_i f_i, \quad (1.9)$$

называемую функцией Гамильтона-Понтрягина или просто гамильтонианом. Постоянный множитель p_0 , который может принимать значения 0 или -1 , называется особым множителем. Экстремаль, соответствующая случаю $p_0 = 0$, называется особой.

Принцип максимума является необходимым условием локальной (а значит, и глобальной) оптимальности. Обобщая известные результаты классического вариационного исчисления [9] на более широкий класс задач (например, на случай кусочно-непрерывного управления и на случай недифференцируемости правых частей и/или лагранжиана по \mathbf{w}), он позволяет найти предположительно оптимальные экстремали.

Теорема 1.2.1 (Принцип максимума)

Пусть $\mathbf{w}_{opt}(t)$ – оптимальное управление, переводящее систему из состояния \mathbf{x}_0 в состояние \mathbf{x}_f по оптимальной траектории \mathbf{x}_{opt} за время $t_f - t_0$. Тогда:

1. существует абсолютно непрерывная функция $\mathbf{p}_{opt} : [t_0, t_f] \rightarrow \mathbb{R}^n$, которая удовлетворяет сопряженной системе (1.8), причем $\mathbf{p}_{opt}(t_0) \neq 0$ или $p_0 \neq -1$;
2. $\mathcal{H}_{opt} \equiv \mathcal{H}(\mathbf{x}_{opt}(t), \mathbf{p}_{opt}(t), \mathbf{w}_{opt}(t)) = \sup_{\mathbf{w} \in W} \mathcal{H}(\mathbf{x}(t), \mathbf{p}(t), \mathbf{w}(t)) \Big|_{\substack{\mathbf{x}=\mathbf{x}_{opt}(t) \\ \mathbf{p}=\mathbf{p}_{opt}(t)}}$
почти всюду на $[t_0, t_f]$.

Если управление $\mathbf{w}_{opt}(t)$ ограничено на $[t_0, t_f]$, то $\mathcal{H}_{opt} = const$, $[t_0, t_f]$ (когда t_f не фиксировано, $\mathcal{H}_{opt} = 0$).

Доказательство этой теоремы приведено в [10]. В задаче со скользящими концами к формулировке принципа максимума добавляются дополнительные условия – условия трансверсальности:

$$\mathbf{p}_{opt}(t_0) = \frac{\partial \mathbf{g}_0}{\partial \mathbf{x}(t_0)} \cdot \mathbf{c}_0,$$

$$\mathbf{p}_{opt}(t_f) = \frac{\partial \mathbf{g}_f}{\partial \mathbf{x}(t_f)} \cdot \mathbf{c}_f.$$

Здесь $\mathbf{c}_0, \mathbf{c}_f$ – произвольные постоянные векторы, сумма размерности каждого из которых и размерности соответствующего ему терминального многообразия равна n . Геометрический смысл условий трансверсальности прост: сопряженный вектор в начальный/конечный момент времени должен быть трансверсален начальному/конечному многообразию (ортогонален касательному пространству). Если граничные условия в одном из концов временного интервала не налагаются на какую-либо компоненту вектора состояния x_j , то соответствующая сопряженная переменная p_j на этом конце обращается в нуль. Обобщая наблюдения, стоит отметить, что получающаяся краевая задача для расширенной системы (1.6)-(1.8) всегда включает терминальные многообразия размерности n на обоих концах временного интервала.

В рассматриваемой задаче будем брать $p_0 = -1$, предполагая, что особые экстремали отсутствуют.

1.3. Необходимые условия оптимальности перелета.

Краевая задача принципа максимума

Опираясь на раздел 1.2 применим принцип максимума к решению модельной задачи, сформулированной в разделе 1.1.

Управлением в данном случае удобно считать ускорение тяги, то есть, используя обозначения из раздела (1.2), $\mathbf{w} = \mathbf{a}_{\text{тяги}} \in W \subset \mathbb{R}^3$, где

$$W = \{0 \leq a_{\text{тяги}} \leq \psi, -\frac{\pi}{2} \leq \alpha \leq \frac{\pi}{2}, 0 \leq \beta \leq 2\pi\},$$

а α, β – углы, задающие единичный вектор тяги.

Теперь перепишем гамильтониан (1.9). В нашей задаче

$$p_0 = -1,$$

$$f_0 = \frac{a_{\text{тяги}}^2}{2},$$

$$\mathbf{p} = (\mathbf{p}_r, \mathbf{p}_v) \in \mathbb{R}^4,$$

$$\mathbf{x} = (\mathbf{r}, \mathbf{v}) \in \mathbb{R}^4,$$

$$\dot{\mathbf{x}} = \mathbf{f} = (\mathbf{v}, \mathbf{a}),$$

тогда, учитывая то, что $\mathbf{a} = \mathbf{a}_{\text{тяги}} + \mathbf{a}_{\text{внеш}}$, получаем гамильтониан:

$$\mathcal{H} = -\frac{a_{\text{тяги}}^2}{2} + \mathbf{p}_r \cdot \mathbf{v} + \mathbf{p}_v \cdot (\mathbf{a}_{\text{тяги}} + \mathbf{a}_{\text{внеш}}).$$

Из принципа максимума получаем непрерывное оптимальное управле-

ние:

$$\mathbf{a}_{\text{тяги}} = \mathbf{p}_v.$$

В случае потенциальных внешних сил уравнения оптимального движения имеют вид

$$\begin{aligned}\ddot{\mathbf{r}} &= \frac{\partial U}{\partial \mathbf{r}} + \mathbf{p}_v, \\ \ddot{\mathbf{p}}_v &= \frac{\partial^2 U}{\partial \mathbf{r}^2} \mathbf{p}_v,\end{aligned}$$

где U – потенциал поля. В нашем случае поле ньютоново, то есть, $U = \mu/r$, где $\mu = GM$ – гравитационный параметр центрального тела. Эту задачу будем решать в безразмерных единицах, полагая $\mu = 1$.

С учетом вышеприведенных формул система уравнений оптимального движения (1.6) и (1.8) будет выглядеть так:

$$\left\{ \begin{array}{l} \dot{\mathbf{r}} = \mathbf{v}, \\ \dot{\mathbf{v}} = -\frac{\mathbf{r}}{r^3} + \mathbf{p}_v, \\ \dot{\mathbf{p}}_v = -\mathbf{p}_r, \\ \dot{\mathbf{p}}_r = -\frac{1}{r^3} \left(3 \frac{\mathbf{r} \cdot \mathbf{p}_v}{r^2} \mathbf{r} - \mathbf{p}_v \right), \end{array} \right. \quad (1.10)$$

где $\mathbf{p}_r, \mathbf{p}_v$ – векторы сопряженных переменных.

Если задать значения $\mathbf{p}_r(t_0), \mathbf{p}_v(t_0)$, то получим задачу Коши и численным интегрированием можем вычислить значения $\mathbf{r}(t_f), \mathbf{v}(t_f)$. Краевая задача принципа максимума выглядит так: удовлетворить условиям

$$\left\{ \begin{array}{l} \mathbf{r}(t_0) = \mathbf{r}_0, \\ \mathbf{v}(t_0) = \mathbf{v}_0, \\ \mathbf{r}(t_f) = \mathbf{r}_f, \\ \mathbf{v}(t_f) = \mathbf{v}_f, \end{array} \right. \quad (1.11)$$

с учетом уравнений движения (1.10). Так как заданы $\mathbf{r}_f, \mathbf{v}_f$ – требуе-

мые значения положения и скорости КА в конечный момент времени, то краевая задача принципа максимума сводится к задаче минимизации функционала невязки $I = (\mathbf{r}(t_f) - \mathbf{r}_f)^2 + (\mathbf{v}(t_f) - \mathbf{v}_f)^2$.

Когда найдем $\mathbf{p}_r(t_0), \mathbf{p}_v(t_0)$, дающие значение функционала $I = 0$, двухточечная краевая задача будет считаться решенной.

Глава 2

Решение краевой задачи принципа максимума методами глобальной оптимизации

В этой главе будут описаны методы, с помощью которых решаются задачи глобальной оптимизации.

2.1. Сведение задачи многомерной оптимизации к задаче одномерной оптимизации

2.1.1. Постановка задачи глобальной оптимизации

Функционалы в задачах астродинамики обычно многомерны, существенно нелинейны, очень овражистые, а вычисление матриц частных производных – это порой очень ресурсоемкий процесс, если не вовсе неосуществимый. Задача глобальной оптимизации в общем случае ставится следующим образом:

$$F^* = F(y^*) = \min\{F(y) : y \in D\}, \quad (2.1)$$

где область поиска

$$S = \{w \in \mathbb{R}^N : a_j \leq w_j \leq b_j, 1 \leq j \leq N\} \quad (2.2)$$

представляет собой произвольный параллелепипед. Его можно привести к стандартному гиперкубу D

$$D = \{y \in \mathbb{R}^N : -2^{-1} \leq y_j \leq 2^{-1}, 1 \leq j \leq N\} \quad (2.3)$$

следующим преобразованием:

$$y_j = \frac{(w_j - (a_j + b_j)/2)}{\rho},$$

где

$$\rho = \max\{b_j - a_j : 1 \leq j \leq N\}.$$

На практике все функции принимаются липшицевыми то есть удовлетворяющими следующему неравенству:

$$\forall y', y'' \in D \quad |F(y') - F(y'')| \leq L \|y' - y''\| = L \left\{ \sum_{j=1}^N (y'_j - y''_j)^2 \right\}^{1/2}, \quad (2.4)$$

где $0 < L < \infty$ – так называемая константа Липшица.

2.1.2. Кривые, заполняющие пространство

Теперь рассмотрим кривые, заполняющие пространство, и процедуру их построения.

Определение 2.1.2.1. *Кривая, заполняющая пространство, – это непрерывная функция $y(x)$, отображающая отрезок $[0, 1]$ в стандартный гиперкуб*

$$D = \left\{ y \in \mathbb{R}^n : -\frac{1}{2} \leq y_j \leq \frac{1}{2}, 1 \leq j \leq N \right\}.$$

Далее опишем процедуру построения такой кривой.

Разделим стандартный гиперкуб D (2.3) на 2^N одинаковых гиперкубов «первого деления» пересечением его N ортогональными гиперплоскостями, которые параллельны одной из координатных гиперплоскостей и проходят через средние точки граней гиперкуба, перпендикулярных соответствующей координатной плоскости. У каждого меньшего гиперкуба, образовавшегося после такого деления, длина ребра равна $1/2$. Введем нумерацию этих гиперкубов как $z_1 : 0 \leq z_1 \leq 2^{N-1}$ и

каждый получившийся после первого разделения гиперкуб будем обозначать как $D(z_1)$. Далее, разделим получившиеся гиперкубики еще на 2^N гиперкубиков, которые будем нумеровать индексами $z_2 : 0 \leq z_2 \leq 2^{N-1}$ и обозначать каждый гиперкуб второго разбиения как $D(z_1, z_2)$. Длина ребра гиперкубиков после второго разбиения будет равна $1/2^2$. Пример двух шагов такого разбиения при $N = 2$ приведен на рисунке 2.1. Продолжая этот процесс, то есть каждый раз разбивая гиперкубы предыдущего разбиения на 2^N вложенных гиперкубиков с длиной ребра вдвое меньше, после M -го шага получим гиперкубики $D(z_1, z_2, \dots, z_M)$. Получается, что исходный гиперкуб разделился на 2^{MN} гиперкубиков так, что

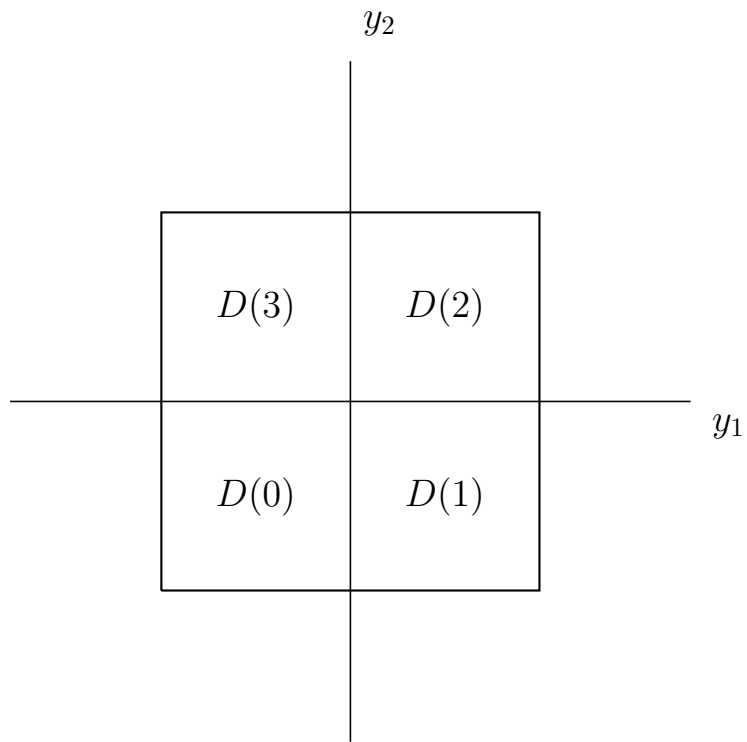
$$D \supset D(z_1) \supset \dots \supset D(z_1, z_2, \dots, z_M), \quad 0 \leq z_j \leq 2^{N-1}, \quad 1 \leq j \leq M.$$

Далее поделим отрезок $[0, 1]$ на оси x на 2^N равных полуинтервалов. Каждую часть обозначим как $d(z_1), 0 \leq z_1 \leq 2^{N-1}$ (нумерация идет слева направо), далее поделим каждую часть на 2^N частей и так далее, как мы делали с гиперкубом. В итоге получим разделение отрезка $[0, 1]$ на 2^{MN} равных частей

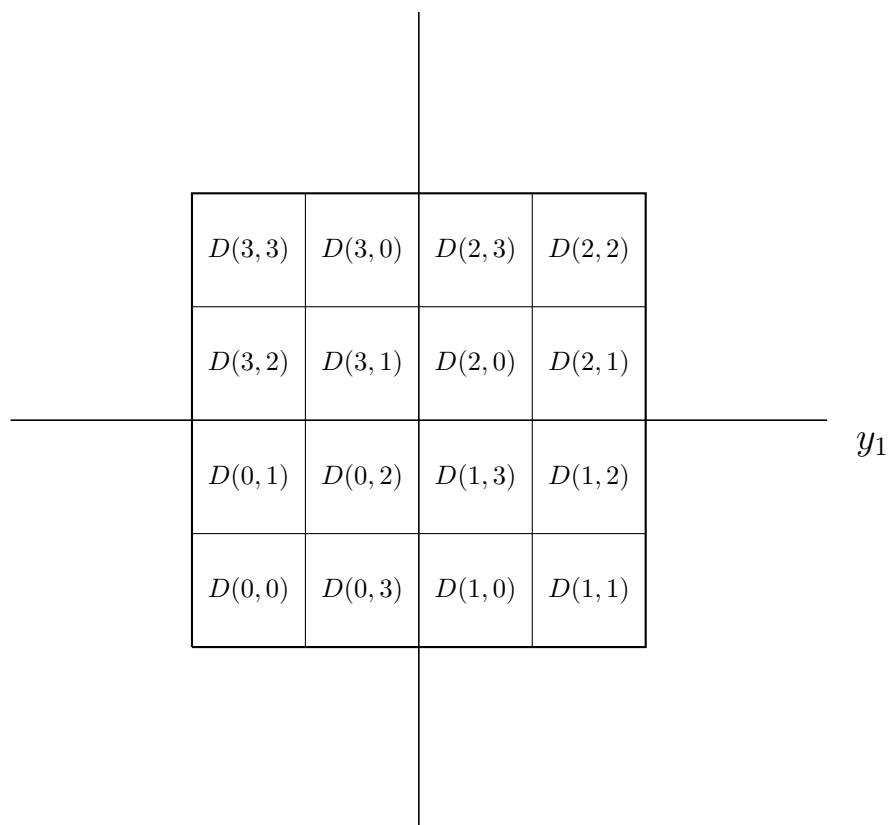
$$[0, 1] \supset d(z_1) \supset \dots \supset d(z_1, z_2, \dots, z_M), \quad 0 \leq z_j \leq 2^{N-1}, \quad 1 \leq j \leq M.$$

Пример такого разделения при $M = 2$ приведен на рисунке 2.2. Представим левую точку ν подинтервала $d(z_1, \dots, z_M) = [\nu, \nu + 2^{-MN})$ в двоичном виде

$$0 \leq \nu = \sum_{i=1}^{MN} \alpha_i 2^{-i} < 1, \quad \alpha_i \in \{0, 1\}, \quad 1 \leq i \leq MN. \quad (2.5)$$



(a) 1 шаг
 y_2



(б) 2 шага

Рис. 2.1. Разбиение квадрата на более мелкие квадраты

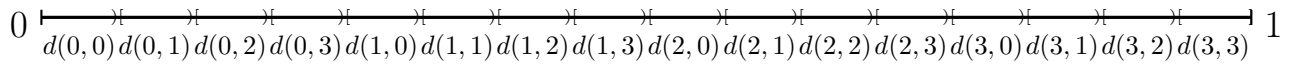


Рис. 2.2. Разбиение отрезка на оси x . Случай $M=2$

Отсюда и из приведенных выше определений легко получить представление индекса $z_j, 1 \leq j \leq M$ в этой же форме

$$z_j = \sum_{i=0}^{N-1} \alpha_{jN-i} 2^i. \quad (2.6)$$

Далее, интервал $d(z_1, \dots, z_M) = [\nu, \nu + 2^{-MN})$ будем обозначать как $d(M, \nu)$. Теперь установим взаимно однозначное соответствие между всеми подинтервалами определенного M -го разбиения и всеми гиперкубами этого же M -го разбиения, полагая, что интервал $d(M, \nu)$ соответствует гиперкубу $D(z_1, \dots, z_M)$ и наоборот. Будем обозначать этот гиперкуб как $D(M, \nu)$. Учитывая вложенность гиперкубов и интервалов друг в друга соотношение удовлетворяет следующим условиям:

1. $D(M+1, \nu') \subset D(M, \nu'')$ тогда и только тогда, когда $d(M+1, \nu') \subset d(M, \nu'')$.
2. Два подинтервала $d(M, \nu')$ и $d(M, \nu'')$ имеют одну конечную точку (ν' или ν'') тогда и только тогда, когда соответствующие субкубики $D(M, \nu')$ и $D(M, \nu'')$ имеют общую грань.

Так как первое условие уже выполнено, то второе условие требует специальной нумерации субкубиков, которое и определяет соответствие этих кубиков интервалам.

Теорема 2.1.1. Пусть $y(x)$ – соответствие, определенное предположением, что для любого $M \geq 1$ образ $y(x) \in D(M, \nu)$ тогда и только тогда, когда прообраз $x \in d(M, \nu)$. Тогда $y(x)$ – это однозначное непрерывное отображение единичного отрезка на стандартный гиперкуб D , а следовательно $y(x)$ – кривая, заполняющая пространство.

Кривая Пеано – предельная кривая, получающаяся в результате какой-либо итерационной процедуры построения. Выбрать приближение с заданной точностью можно всегда. Далее изложим приближение так называемой пеаноподобной неинъективной разверткой $n_M(x)$, но начнем с деления и нумераций.

Введем вспомогательный гиперкуб

$$\Delta = \{y \in \mathbb{R}^N : -2^{-1} \leq y_j \leq 3 \cdot 2^{-1}, 1 \leq j \leq N\} \quad (2.7)$$

и обозначим как $\Delta(s), 0 \leq s \leq 2^N - 1$ гиперкубики, представляющие первое разделение Δ . Исходя из выбора Δ центральные точки $u(s)$ соответствующих кубиков $\Delta(s)$ – N -мерные двоичные вектора. Определим нумерацию центров (а значит, соответствующих субкубиков $\Delta(s)$) соотношениями

$$u_j(s) = (\beta_j + \beta_{j-1}) \pmod{2}, \quad 1 \leq j \leq N, \quad u_N(s) = \beta_{N-1}, \quad (2.8)$$

где $\beta_j, 0 \leq j \leq N$ – цифры в двоичном представлении числа s

$$s = \beta_{N-1}2^{N-1} + \dots + \beta_02^0. \quad (2.9)$$

Можно показать, что все центры разные, любые два смежных центра отличаются только в одной координате и то, что

$$u(0) = (0, \dots, 0, 0), \quad u(2^N - 1) = (0, \dots, 0, 1) \quad (2.10)$$

Предложенная выше схема может быть использована для нумерации кубиков $D(z_1), 0 \leq z_1 \leq 2^N - 1$, представляющих первое разделение стандартного гиперкуба D . Это гарантирует то, что у соседних кубиков

общая грань. Чтобы это сделать, используем линейное отображение

$$g(y) = 2y + p, y \in \mathbb{R}^N, \quad (2.11)$$

$$p = (2^{-1}, \dots, 2^{-1}) \in \mathbb{R}^N, \quad (2.12)$$

удовлетворяющее условию

$$g(D) = \Delta, \quad (2.13)$$

в предположении, что у субкубика $D(z_1)$ номер $z_1 = s$ тогда и только тогда, когда $D(z_1)$ – это прообраз $\Delta(s)$, то есть

$$g(D(z_1)) = \Delta(s). \quad (2.14)$$

Чтобы использовать этот подход к нумерации гиперкубиков второго разбиения, используем линейное отображение

$$g(z_1; y) = 2^2\{y - [u(z_1) - p]2^{-1}\} + p, \quad (2.15)$$

удовлетворяющее условию

$$g(z_1; D(z_1)) = \Delta, 0 \leq z_1 \leq 2^N - 1. \quad (2.16)$$

Пусть $D(z_1, z_2)$ из $D(z_1)$ имеет индекс $z_2 = s$ тогда и только тогда, когда $D(z_1, z_2)$ прообраз $\Delta(s)$, то есть

$$g(z_1; D(z_1, z_2)) = \Delta(s) \quad (2.17)$$

Заметим, что $u(z_1)$ – центр куба $\Delta(z_1)$, сопоставленный $D(z_1)$ соотношением $g(D(z_1)) = \Delta(s)$.

При всем при этом все равно нужно придумать связь в нумерации элементов $D(z_1, z_2), 0 \leq z_2 \leq 2^N - 1$, которые лежат в разных кубиках

$D(z_1)$ разделения, произошедшего раньше.

Для начала воспользуемся фактом (2.10). По сути это означает, что центры $u(0)$ и $u(2^N - 1)$ отличаются только в N -той координате. Перестановка u_N и u_t в $u(s)$, которую обозначим

$$u^t(s) = (u_1(s), \dots, u_{t-1}(s), u_N(s), u_{t+1}(s), \dots, u_{N-1}(s), u_t(s)),$$

$$1 \leq t \leq N \quad (2.18)$$

не меняет начального вектора, то есть, $u^t(0) = u(0)$, а переносит единственную ненулевую координату $u(2^N - 1)$ на t -ую позицию, что значит то, что $u^t(0)$ и $u^t(2^N - 1)$ отличаются только в t -ой координате.

Если прибавить двоичный вектор $q \in \mathbb{R}^N$ к векторам $u^t(s)$, $0 \leq s \leq 2^N - 1$, получим новые вектора $u^{tq}(s)$ такие, что

$$u^{tq}(s) = (u_i^t(s) + q_i) \pmod{2}, \quad 1 \leq i \leq N. \quad (2.19)$$

Эта операция помогает удостовериться в том, что q – центр начального куба $\Delta(0)$, то есть, что

$$u^{tq}(0) = q. \quad (2.20)$$

Эти две операции (перестановка и добавление двоичного вектора) позволяют нам построить нумерацию, которая точно учитывает то, что q – действительно центр начального субкубика $\Delta(0)$ и что центры субкубиков $\Delta(0)$ и $\Delta(2^N - 1)$ различны только в t -ой координате для всех $t : 1 \leq t \leq N$.

Отображение (2.15) можно обобщить на случай гиперкубика M -го разбиения. Он вводится следующим образом:

$$g(z_1, \dots, z_M; y) = 2^{M+1} \left\{ y - \sum_{j=1}^M [u^{tq}(z_j) - p] 2^{-j} \right\} + p, \quad (2.21)$$

в предположении, что субкубик $D(z_1, \dots, z_M, z_{M+1})$ характеризуется индексом z_{M+1} тогда и только тогда, когда

$$g(z_1, \dots, z_M; D(z_1, \dots, z_{M+1})) = \Delta(s). \quad (2.22)$$

Здесь гиперкуб $\Delta(s)$ имеет центр $u^{tq}(s)$ из (2.19), где

$$t = t(z_{M+1}) = \begin{cases} N, & M = 0, \\ l(z_M), & M > 0, \end{cases} \quad (2.23)$$

а

$$q = q(z_{M+1}) = \begin{cases} (0, \dots, 0) \in \mathbb{R}^N, & M = 0, \\ w(z_M), & M > 0. \end{cases} \quad (2.24)$$

Здесь $l(z_M)$ – номер координаты, которая различна у $u(s)$ и $u(s-1)$, где $2 \leq s \leq 2^N - 2$, $l(0) = l(2^N - 1) = 1$, $s \bmod 2 \equiv 0$, а $w(s)$ задается следующей системой:

$$w_i(s+1) = w_i(s) = \begin{cases} 1 - u_i(s), & i = 1, \\ u_i(s), & 2 \leq i \leq N, \end{cases} \quad (2.25)$$

где $s \bmod 2 \equiv 1$, $w(0) = u(0)$. В соответствии с (2.21) и (2.22) точка

$$y(z_1, \dots, z_M) = \sum_{j=1}^M [u^{tq}(z_j) - p] 2^{-j} \quad (2.26)$$

лежит во всех гиперкубиках $M+1$ разбиения $D(z_1, \dots, z_{M+1})$, $0 \leq z_{M+1} \leq 2^N - 1$ $D(z_1, \dots, z_M)$, далее эту точку будем называть центром гиперкуба $D(z_1, \dots, z_M)$

Сейчас перейдем к процессу построения аппроксимации кривой.

Определение 2.1.2.2. *Равномерной сеткой $P(M, N)$ называется сет-*

ка следующего вида:

$$P(M, N) = \{y^i; 0 \leq i \leq 2^{MN} - 1\}. \quad (2.27)$$

Пеаноподобная неинъективная развертка – отображение равномерной сетки в отрезке $[0, 1]$ на сетку $P(M, N)$ в стандартном гиперкубе D с шириной 2^{-M} (по каждой координате), удовлетворяющую условию

$$P(M, N) \subset P(M + 1, N). \quad (2.28)$$

Развертка $n(x)$ аппроксимирует кривую Пеано $y(x)$, а ее точки в D могут иметь несколько прообразов на отрезке $[0, 1]$.

Теперь опишем построение $n(x)$. Пусть множество узлов $P(M, N)$ совпадает со множеством вершин гиперкубов $D(z_1, z_2, \dots, z_M)$ M -го разбиения. Тогда расстояние между узлами такой сетки равно 2^{-M} и количество узлов $P(M, N)$ равно $(2^M + 1)^N$. Так как вершины гиперкубов M -го разбиения так же являются вершинами гиперкубов любого разбиения на уровень ниже, то есть $M + \nu$ -го разбиения, где $\nu \geq 1$, то включение (2.28) верно.

Заметим, что каждая из 2^N вершин на любом гиперкубе M -го разбиения $D(z_1, \dots, z_M)$ в то же время является вершиной только одного гиперкуба $D(z_1, \dots, z_{M+1})$ гиперкуба $D(z_1, \dots, z_M)$. Обозначим как $P(z_1, \dots, z_M)$ общую вершину гиперкубов

$$D(z_1, \dots, z_{M+1}) \subset D(z_1, \dots, z_M). \quad (2.29)$$

Исходя из (2.26) центр гиперкуба из левой части (2.29) и центр гиперкуба из правой части (2.29) связаны соотношением

$$y(z_1, \dots, z_{M+1}) = y(z_1, \dots, z_M) + (u^{tq}(z_{M+1}) - 2^{-1}) \cdot 2^{-(M+1)}, \quad (2.30)$$

откуда следует, что

$$n(z_1, \dots, z_{M+1}) = y(z_1, \dots, z_M) + (u^{tq}(z_{M+1}) - 2^{-1}) \cdot 2^{-M}. \quad (2.31)$$

Перебором z_{M+1} от 0 до $2^N - 1$ получим из (2.31) все 2^N вершин гиперкуба $D(z_1, \dots, z_M)$.

Формула (2.31) устанавливает соответствие между $2^{(M+1)N}$ интервалами $d(z_1, \dots, z_{M+1})$ M -го разбиения отрезка $[0, 1]$ и $(2^M + 1)^N$ узлами $n(z_1, \dots, z_{M+1})$ сетки $P(M, N)$. Это соответствие, очевидно, неинъективно.

Пронумеруем все интервалы $d(z_1, \dots, z_{M+1})$ слева направо с индексом $i : 0 \leq i \leq 2^{(M+1)N} - 1$ и обозначим конечные точки i -го интервала как ν_i, ν_{i+1} , далее введем нумерацию центров $y^i = y(z_1, \dots, z_{M+1})$ из (2.26), полагая, что центру, соответствующему гиперкубу $D(z_1, \dots, z_{M+1})$ присвоен тот же номер i , что и интервалу $d(z_1, \dots, z_{M+1}) = [\nu_i, \nu_{i+1})$. Таким образом, мы определили соответствие узлов

$$\nu_i : 0 \leq i \leq 2^{(M+1)N} - 1, \quad (2.32)$$

составляющих сетку на отрезке $[0, 1]$ и центров y^i , которые в соответствии с (2.31) формируют однозначное соответствие концов интервала и узлов сетки $p \in P(M, N)$. Заметим, что если центры y^i и y^{i+1} из одного гиперкуба M -го разбиения, то эти центры (следовательно и соответствующие точки ν_i и ν_{i+1}) сопоставлены разным узлам сетки $P(M, N)$. Поэтому узел p может быть сопоставлен точкам ν_i, ν_{i+1} тогда и только тогда, когда соответствующие центры y^i и y^{i+1} из разных, (но смежных) гиперкубов M -го разбиения. Так как количество гиперкубов M -го разбиения D равно 2^{MN} , то там ровно $2^{MN} - 1$ пар ν_i и ν_{i+1} , сопоставленных одному и тому же узлу сетки $P(M, N)$.

Чтобы убедиться, что любых два близких узла в отрезке $[0, 1]$ сопоставлены разным узлам $P(M, N)$ мы заменим каждую из пар ν_i, ν_{i+1} на один узел единичного отрезка. Далее мы изменим расположение узлов в $[0, 1]$, чтобы сохранить равномерность сетки. Чтобы это сделать, построим равномерную сетку на отрезке $[0, 1]$ с узлами

$$h_j : 0 \leq j \leq 2^{(M+1)N} - 2^{MN} = q, \quad (2.33)$$

где $h_0 = 0$, $h_q = 1$ и сопоставим узлу h_j сетки (2.33) узел сетки (2.32), где

$$i = j + \lfloor (j - 1)/(2^N - 1) \rfloor. \quad (2.34)$$

Далее мы предполагаем, что узел h_j сопоставлен с узлом сетки $P(M, N)$, порожденной (2.31) для центра y^i с i из (2.34). Это отображение равномерной сетки (2.34) на отрезке $[0, 1]$ на сетку $P(M, N)$ в гиперкубе D и называется пеаноподобной неинъективной разверткой и обозначается $n(x) = n_M(x)$.

2.1.3. Сведение задачи многомерной глобальной оптимизации к одномерной с помощью кривой Пеано

Свести задачу оптимизации (2.1) с помощью кривой Пеано можно следующим образом:

$$F^* = F(\mathbf{y}^*) = F(y(x^*)) = \min\{F(y(x)) : x \in [0, 1]\}, \quad (2.35)$$

где $y(x)$ – кривая Пеано (2.1.3.1). То есть, пусть $F = F(\mathbf{y})$ наш исходный функционал, $\mathbf{y} \in \mathbb{R}^n$. Пусть $y = y(x)$ – кривая Пеано (отображение из \mathbb{R} в \mathbb{R}^n). Тогда составляем новый функционал, но уже одной переменной:

$$f(x) = F(y(x)).$$

Его мы оптимизируем. Получаем x^* — оптимальное решение. Тогда иско-
мое решение

$$y^* = y(x^*).$$

2.2. Алгоритмы одномерной оптимизации

Общая схема всех используемых далее алгоритмов одномерной оп-
тимизации выглядит следующим образом:

1. инициализируем начальные узлы сетки — концы отрезка $[a, b]$
2. перенумеруем узлы сетки так, чтобы $a = x_0 < x_1 < \dots < x_{k-1} < x_k = b$
3. для всех интервалов посчитаем характеристику R — некую скаляр-
ную величину
4. находим интервал с минимальной характеристикой. Тут возможны
два варианта
 - а. если расстояние между концами этого интервала меньше за-
данной точности, то выходим из алгоритма. Находим мини-
мальное значение функции и ее индекс в массиве значений
функций. По индексу находим точку минимума.
 - б. если расстояние между концами этого интервала больше за-
данной точности, то добавляем в этот интервал новый узел и
переходим к шагу 2.

Теперь перейдем к описанию различных методов нахождения ми-
нимума функции одной переменной на отрезке $[a, b]$, то есть методов
решения задачи

$$f^* = f(x^*) = \min\{f(x) : x \in [a, b]\}, \quad (2.36)$$

где целевая функция может быть недифференцируемой, многоэкстремальной. Для начала рассмотрим случай, когда функция липшицева с константой Липшица L . Получим следующее неравенство:

$$f(x) \geq f(y) - L|x - y|, \quad \forall x, y.$$

Если зафиксируем точку $y \in [a, b]$, получим следующую функцию:

$$C(x) = f(y) - L|x - y|, f(x) \geq C(x), \quad x \in [a, b]. \quad (2.37)$$

Эту кусочно-линейную функцию называют минорантой. Целевая функция мажорирует миноранту. Пример функции и ее миноранты приведен на рисунке 2.3.

Методы одномерной оптимизации липшицевых функций можно условно разделить на два вида: методы с априори известной константой Липшица и методы с оценкой константы Липшица на интервале. Проиллюстрируем основные идеи на примере простейшего метода первого класса – метода Пиявского.

2.2.1. Метод Пиявского

Достаточно эффективным для решения задачи (2.36) показал себя метод Пиявского [11], в котором на каждом подинтервале $[x_{i-1}, x_i]$, $2 \leq i \leq k$, где x_1, \dots, x_k – узлы сетки, строится миноранта следующего вида:

$$C^k(x) = \bigcup_{i=2}^k c_i(x), C^k(x) \leq f(x), \quad x \in [a, b], \quad (2.38)$$

где

$$c_i(x) = \max\{f(x_{i-1}) - L(x - x_{i-1}), f(x_i) - L(x - x_i)\}, \quad x_i \in [a, b]. \quad (2.39)$$

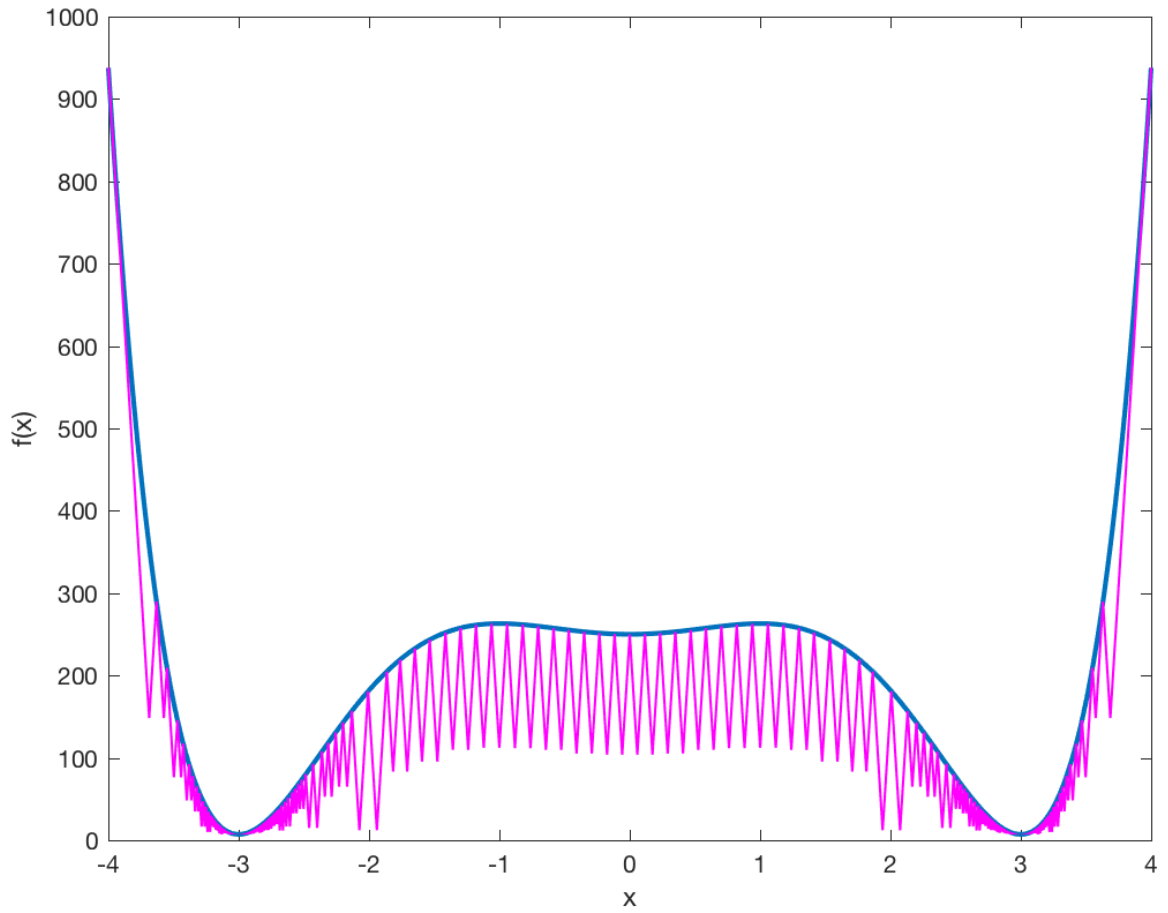


Рис. 2.3. Функция и ее миноранта.

Несложно показать, что минимум «зубца»

$$\tilde{c}_i(\tilde{x}_i) = \frac{f(x_{i-1}) + f(x_i)}{2} - L \frac{x_i - x_{i-1}}{2}$$

достигается в точке

$$\tilde{x}_i = \frac{x_i + x_{i-1}}{2} + \frac{f(x_{i-1}) - f(x_i)}{2L}.$$

Действительно, «зубец» строится из двух прямых

$$f(x_{i-1}) - L(x - x_{i-1}),$$

$$f(x_i) + L(x - x_i).$$

Приравняв их уравнения и перенеся x в правую часть равенства, получим, что их точка пересечения и одновременно, точка минимума «зубца» будет в точке

$$\tilde{x}_i = \frac{x_i + x_{i-1}}{2} + \frac{f(x_{i-1}) - f(x_i)}{2L}.$$

Подставив ее в одно из уравнений прямых, получим минимальное значение зубца.

В методе Пиявского каждый следующий узел x^{k+1} на отрезке выбирается так, что

$$C^k(x^{k+1}) = \min_{x \in [a, b]} C^k(x),$$

то есть узел выбирается там, где зубец наиболее глубокий. Если исходить из терминов общей схемы, представленной выше, то характеристикой в методе Пиявского называется глубина зубца на интервале.

Таким образом, алгоритм продолжается, пока следующий узел не будет меньше заданного заранее значения. Тогда через K итераций глобальный минимум функции f^* и точку x^* , в которой он достигается, можно найти следующим образом:

$$f^* = \min\{f(x_i) : 1 \leq K\},$$

$$x^* = \arg \min\{f(x_i) : 1 \leq K\}.$$

У метода Пиявского есть один существенный недостаток: он требует либо точного знания константы Липшица L , либо знания ее переоцененного значения, однако использование переоцененного значения увеличивает количество итераций алгоритма.

2.2.2. Геометрический алгоритм

Когда функция не липшицева, а гёльдерова с показателем гёльдера $\alpha < 1$ можно построить аналогичный алгоритм, использующий в каче-

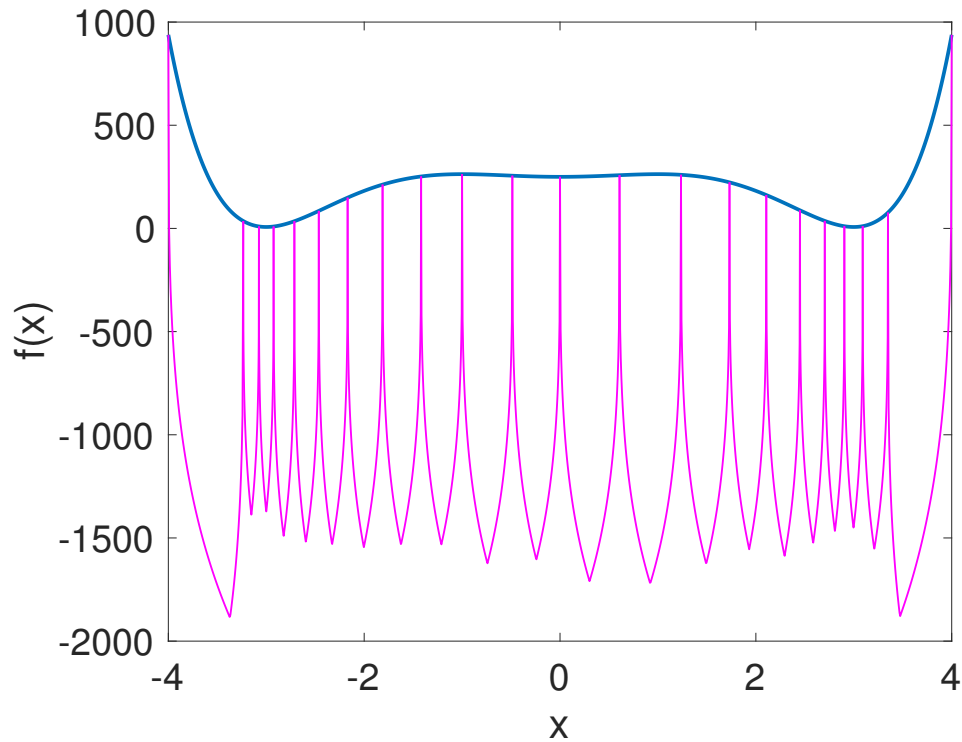


Рис. 2.4. Функция и ее миноранта в гёльдеровой метрике.

стве характеристики значение в зубце миноранты. В этом случае, однако, миноранта представляет собой не кусочно-линейную, а кусочно-гиперболическую функцию. Такой алгоритм называется геометрическим.

В случае геометрического алгоритма миноранты выглядят иначе. Аналитически они строятся так же

$$C^k(x) = \bigcup_{i=2}^k c_i(x),$$

но здесь

$$c_i(x) = \max\{f(x_{i-1}) - rm_i(x - x_{i-1})^{1/N}, f(x_i) - rm_i(x - x_i)^{1/N}\}, x \in [x_{i-1}, x_i].$$

Функция и ее миноранта в этом случае выглядят так, как показано на рисунке 2.4.

Пусть $f(x)$ удовлетворяет условию Гёльдера

$$|f(x) - f(y)| \leq H|x - y|^{1/N}, x, y \in [a, b], \quad (2.40)$$

с константой $0 < H < \infty$.

Рассмотрим два случая:

1. константа Гёльдера или ее переоцененное значение $M \geq H$ дана априори,
2. ничего не известно про константу Гёльдера H .

Для оценки константы Гёльдера также вводится параметр алгоритма r – параметр надежности алгоритма (величина на которую переоцениваем константу Гёльдера). Для начала приведем общую схему алгоритма, который назовем для краткости ГА (геометрический алгоритм), потом представим его модификации

Вход: $a, b, f, \varepsilon, \xi, r > 1$; // концы отрезка, оптимизируемая функция, точность, малый параметр

Выход: f_{opt}, x_{opt} ; // минимум и точка минимума

1: $x_0 = a, x_1 = b$; // инициализация узлов сетки

2: **пока** не достигнется условие выхода

3: перенумеруем все узлы сетки так, чтобы $a = x_0 < x_1 < \dots < x_{k-1} < x_k = b$;

4: Посчитаем rm_i // Значение константы Гёльдера

5: **для всех** интервалов $[x_{i-1}, x_i], 1 \leq i \leq k$

6: $y_i = \frac{1}{2}(x_i + x_{i-1}) + \frac{(z_i - z_{i-1})}{2rm_i(x_i - x_{i-1})^{\frac{1-N}{N}}}$

7: **для всех** интервалов $[x_{i-1}, x_i], 1 \leq i \leq k$

8: $R_i = \min\{z_{i-1} - rm_i(y_i - x_{i-1})^{1/N}, z_i - rm_i(x_i - y_i)^{1/N}\}$ // характеристика интервала

- 9: $t = \arg \min_{1 \leq i \leq k} R_i$ // выберем интервал $[x_{t-1}, x_t]$ с максимальной характеристикой
- 10: **если** $x_t - x_{t-1} > \varepsilon$ **то**
- 11: $x^{k+1} = y_t$ // добавляем новый узел
- 12: **иначе** // выходим
- 13: $f_{opt} = \min\{z_i : 1 \leq i \leq k\}$
- 14: $x_{opt} = \arg \min\{z_i : 1 \leq i \leq k\}$
- 15: **выход**;

Для случая 1 rm_i равно точному значению константы Гёльдера H (модификация алгоритма ГА1).

Для случая 2 rm_i – оценка константы Гёльдера на интервале (модификация алгоритма ГА2). Она вычисляется следующим образом:

$$m_i = \max\{\xi, h^k\},$$

$$h^k = \max\{h_i\}, h_i = \frac{|z_i - z_{i-1}|}{|x_i - x_{i-1}|^{1/N}}, 2 \leq i \leq k$$

В методе ГА, представленном выше на каждом интервале $[x_{i-1}, x_i]$, $2 \leq i \leq k$ мы приближаем точку p_i точкой y_i , которая является точкой пересечения прямых r_{right} и r_{left} , уравнения которых выглядят следующим образом:

$$r_{\text{left}} = -rm_i(x_i - x_{i-1})^{\frac{1-N}{N}}x + rm_i(x_i - x_{i-1})^{\frac{1-N}{N}}x_{i-1} + f(x_{i-1}),$$

$$r_{\text{right}} = rm_i(x_i - x_{i-1})^{\frac{1-N}{N}}x - rm_i(x_i - x_i)^{\frac{1-N}{N}}x_{i-1} + f(x_i).$$

Характеристика R_i тоже имеет геометрический смысл, как и в алгоритме Пиявского:

$$R_i = \min\{c_i^-(y_i), c_i^+(y_i)\},$$

$$c_i^- = f(x_{i-1}) - rm_i(x - x_{i-1})^{1/N},$$

$$c_i^+ = f(x_i) - rm_i(x_i - x)^{1/N}.$$

Стоит отметить, что точку p_i посчитать сложнее, чем ее приближение y_i .

Теперь перейдем к исследованию сходимости алгоритмов ГА1 и ГА2.

Лемма 2.2.1. *Если $rm_i > H_i$, где H_i – локальная оценка константы Гёльдера на интервале $[x_{i-1}, x_i]$, тогда*

$$R_i < f(x).$$

Определение 2.2.2.1. *Сходимость к точке $x' \in (a, b)$ называется двусторонней, если существует две последовательности $\{x^k\}$, одна из которых сходится к $\{x^k\}$ справа, а другая слева.*

Теорема 2.2.1. *Пусть x' любая предельная точка $\{x^k\}$ такая, что $x' \neq a, x' \neq b$. Тогда сходимость к x' двусторонняя.*

Следствие 1. *Для всех узлов отрезка x^k $f(x^k) \geq f(x'), k \geq 1$.*

Следствие 2. *Если существует еще одна предельная точка $x'' \neq x'$, то $f(x'') = f(x')$.*

Следствие 3. *Если у функции $f(x)$ конечное число локальных минимумов на отрезке $[a, b]$, то x' локально оптимальна.*

Теорема 2.2.2 (Достаточное условие глобальной сходимости) *Пусть x^* точка, доставляющая глобальный минимум функции $f(x)$. Если существует итерация с номером k^* такая, что $k > k^*$ неравенство*

$$rm_{j(k)} > H_{j(k)}$$

сохраняется, где $H_{j(k)}$ – константа Гёльдера на интервале $[x_{j(k)-1}, x_{j(k)}]$, то есть,

$$|f(x) - f(y)| \leq H_{j(k)}|x - y|^{1/N}, \quad x, y \in [x_{j(k)-1}, x_{j(k)}]$$

и x^* лежит в отрезке $[x_{j(k)-1}, x_{j(k)}]$, то точка x^* предельная точка последовательности $\{x^k\}$.

Теорема 2.2.3. Для каждой функции $f(x)$, удовлетворяющей условию (2.40) и с константой Гёльдера $H < \infty$ существует такой параметр надежности алгоритма r^* такой, что для всех $r > r^*$ алгоритм ГА2 находит все точки глобального минимума функции $f(x)$ на отрезке поиска $[a, b]$.

Доказательства всех следствий и теорем можно найти в [12].

2.2.3. Идея для ускорения

Чтобы ускорить алгоритм ГА2, можно модернизировать его введением так называемой «локальной настройки». Она заключается в изменении способа оценки константы Гёльдера. Теперь вместо того, чтобы выбирать m_i традиционным образом

$$m_i = \max\{\xi, h^k\}, \quad (2.41)$$

будем выбирать ее как

$$m_i = \max\{\xi, \lambda_i, \gamma_i\}. \quad (2.42)$$

Здесь

$$\lambda_i = \max\{h_{i-1}, h_i, h_{i+1}\}, \quad h_i = \frac{|z_i - z_{i-1}|}{|x_i - x_{i-1}|^{1/N}}, \quad 3 \leq i \leq k - 1. \quad (2.43)$$

Если $i = 2$, берем только h_2, h_3 , если $i = k$, берем h_{k-1}, h_k . Также, значение γ_i вычисляется по формуле

$$\gamma_i = h^k \frac{x_i - x_{i-1}}{X^{max}}, \quad h^k = \max\{h_i\},$$

$$X^{max} = \max\{x_i - x_{i-1} : 2 \leq i \leq k\}. \quad (2.44)$$

Результаты численных экспериментов можно найти в [12].

2.2.4. Связь между многомерной липшицевой функцией и соответствующей ей гёльдеровой функцией

Чтобы использовать описанные выше методы одномерной оптимизации в применении к многомерным липшицевым функциям, нужно понимать, какими свойствами будет обладать одномерная функция, построенная на основе кривой, заполняющей многомерное пространство.

Если метод использует аппроксимацию кривой Пеано $p_M(\cdot)$ порядка M и обеспечивает нижнюю границу U_M^* для одномерной функции $y(x)$, тогда это значение будет нижней границей для функции $F(y)$, но только вдоль кривой $p_M(\cdot)$. Остается вопрос, можно ли установить нижнюю границу для многомерной функции $F(y)$ на всем многомерном интервале поиска D .

Теорема 2.2.4. Пусть U_M^* – нижняя граница вдоль кривой Пеано $p_M(x)$ для многомерной функции $F(y)$, удовлетворяющей условию Липшица с константой Липшица L , то есть

$$U_M^* \leq F(p_M(x)), \quad x \in [0, 1], \quad (2.45)$$

тогда

$$U = U_M^* - 2^{-(M+1)} L \sqrt{N} \quad (2.46)$$

является нижней границей для $F(y)$ на всей области D , то есть, $U \leq F(y)$, $y \in D$.

Теорема 2.2.5. Пусть $y(x)$ – соответствие одномерного отрезка $[0, 1]$ на стандартный гиперкуб, то есть кривая, заполняющая пространство. Пусть функция $F(y)$, $y \in D$ липшицева с константой Липшица L в \mathbb{R}^N , тогда одномерная функция $F(y(x))$, $x \in [0, 1]$ гёльдерова с экспонентной $1/N$ и константой Гёльдера $2L\sqrt{N+3}$, то есть, удовлетворяет условию

$$|F(y(x')) - F(y(x''))| \leq 2L\sqrt{N+3}(|x' - x''|)^{1/N}, \quad x', x'' \in [0, 1]. \quad (2.47)$$

Доказательства теорем 2.2.4 и 2.2.5 приведены в [12].

Глава 3

Эксперименты и результаты

В этой главе будут описаны эксперименты, проведенные в рамках работы, а так же их результаты.

3.1. Оптимизация с помощью кривых Пеано

На языке MATLAB был реализован геометрический алгоритм, описанный в предыдущей главе. С его помощью перебирались начальные векторы сопряженных переменных для решения краевой задачи (1.10). Для построения аппроксимации кривой Пеано была использована пеано-подобная неинъективная развертка и версия геометрического алгоритма ГА2, а также был реализован метод ускорения, описанный также в предыдущей главе. В качестве отрезка поиска был использован стандартный единичный отрезок $[0, 1]$. Для численного интегрирования системы (1.10) использовался слегка модифицированный метод Рунге-Кутты 4 порядка. Модификация в интегратор была введена в силу того, что при некоторых значениях сопряженных переменных траектория аппарата проходила близко к притягивающему центру, что влечет за собой рост вычислительных ошибок, так как модуль радиус-вектора становится очень маленьким, к тому же такое движение с практической точки зрения неосуществимо. В связи с этим, если значение нормы радиус-вектора становилось меньше некоторой заданной величины, интегрирование прекращалось и значения скорости и радиус-вектора искусственно увеличивались для того, чтобы значение невязки становилось очень большим и алгоритм не рассматривал точку, в которой оно достигается.

Стоит отдельно заметить, что при вычислении на компьютере нельзя брать произвольный порядок кривой M . Это связано с ограничен-

ностью бит, выделяемых компьютером для хранения чисел с двойной точностью. Таким образом, M и N связаны соотношением $MN < \Gamma$, где Γ – количество чисел в мантиссе. В 64 - битных процессорах для чисел с двойной точностью $\Gamma = 16$.

В алгоритм была внедрена модификация, которая позволяла получить из точки на отрезке точку на кривой Пеано, которая заполняет стандартный гиперкуб D в пространстве сопряженных переменных. Далее, считался функционал невязки в этой точке. Эксперименты показали, что алгоритм не работает в данном случае (проходит большое количество итераций, но никакой сходимости нет или есть, но в локальный минимум). Также, из-за того, что невязка была большой, нельзя было запустить локальный оптимизационный алгоритм типа `fsolve`.

Далее, другими методами было получено точное решение. Тогда в алгоритм была внедрена следующая модификация: после получения точки в стандартном гиперкубе его ребра сжимались до какого-то небольшого значения, которое было подобрано исходя из порядка точного решения, после этого центр гиперкуба из точки $(0, 0, 0, 0)$ перемещался в точное решение. Мы использовали алгоритм ГА2 с локальной настройкой. Порядок кривой M брался равным 3. В качестве отрезка поиска брался стандартный единичный отрезок $[0, 1]$, точность $\varepsilon = 10^{-6}$.

После проведения ряда экспериментов с различными значениями окрестности решения было получено удовлетворительное решение (значение функционала, выдаваемое алгоритмом было около машинного нуля 10^{-17}). Это связано с тем, что кривая Пеано всегда проходит через точку $(0, 0, 0, 0)$, а соответственно, после перемещения центра она переходила в истинное решение.

Если же кубик не перемещался и не сжимался, то алгоритм просто скатывался в локальный минимум. В связи с этим было введено ограничение на количество итераций, совершаемых алгоритмом: 4000 итераций.

После этого был проведен ряд экспериментов при котором оптимизировался не четырехмерный функционал, а функционал меньшей размерности: две или одна переменные фиксировались (брались соответствующие координаты оптимального решения), а по другим двум шел оптимизационный процесс. Результаты экспериментов были ожидаемы: в пространстве меньшей размерности алгоритм достаточно быстро сходился с перемещением центра гиперкуба в истинное решение, но без перемещения гиперкуба в пространстве не сходился к оптимальному значению.

С практической точки зрения было выгодно определить хотя бы примерную «ширину» области притяжения, в связи с чем был проведен еще ряд экспериментов, в котором истинная точка минимума смещалась на какое-то значение и запускался оптимизационный процесс. Оказалось, что достаточно удовлетворительный ответ (когда величина невязки, выдаваемая алгоритмом после 4000 итераций или меньше становилась примерно 10^{-6}) алгоритм выдавал при отклонении от центра на величину 10^{-3} . При этом нас удовлетворяло и значение невязки равное 10^{-4} . Получается, что примерная ширина области притяжения глобального экстремума равна примерно 10^{-3} . Это значит, что когда кривая проходит в окрестности, равной ширине области притяжения глобального минимума, алгоритм хорошо работает. В таблице 3.1 результаты данного эксперимента. Под расстоянием между решениями подразумевается расстояние между истинным решением и полученным в ходе работы алгоритма. Флаг сходимости показывает, сошелся ли алгоритм к глобальному минимуму. Под невязкой подразумевается ее порядок.

Результаты всех этих экспериментов можно увидеть в таблицах 3.2 - 3.7.

3.2. Глобальный линейный поиск

После проведения экспериментов с использованием кривой Пеано было принято решение попробовать линейный поиск (ЛП), суть которого состоит в нахождении направления вдоль которого функция убывает и вычислении шага вдоль этого направления. В качестве инструмента для подбора шага был использован алгоритм глобальной оптимизации ГА2, что давало возможность сойтись не к локальному минимуму, а к глобальному.

В качестве начальной точки была выбрана точка $(0, 0, 0, 0)$, соответствующая пассивному движению КА по начальной орбите. Схема этого алгоритма следующая:

1. Численно вычисляем градиент функционала ∇I и выбираем направление по антиградиенту.
2. Запускаем ГА2 с локальной настройкой. Минимизируем функционал $I(p - \nabla I \cdot step)$ по размеру шага $step$.
3. Вычисляем следующую точку $p_{new} = p_{old} - \nabla I \cdot opt_step$.
4. Если получили заданную точность на f , выходим.

В этот раз в ГА2 подавался отрезок $[a, b] = [0, 0.2]$, а точность $\varepsilon = 0.2 \cdot 10^{-5}$, точность на $I - 10^{-8}$.

Эксперименты показали, что и данный метод «скатывается» в точку локального минимума и не выходит из нее. В связи с этим, было принято решение вычислять градиент по случайной координате, а после того, как алгоритм совершает какое-то большое число итераций, присваивать значению градиента случайное направление (назовем эту модификацию ЛП М1). Даже после этих двух модификаций получалось, что алгоритм все равно находил только локальную точку минимума, хоть и с меньшим

значением. К тому же, после попадания в эту точку алгоритм выдавал значение `opt_step` равным 0. В связи с этой проблемой так же форсированно менялось и значение оптимального шага (модификация ЛП М2), но и это не привело к удовлетворительному результату. Результаты экспериментов приведены на рисунках 3.1 и 3.2. Сразу отметим, что хоть и алгоритм ЛП показывает лучший результат на последней (60) итерации, на ней ГА2 не сошелся даже после 44362 итераций.

Проанализировав эти графики, можно заключить, что при достаточно долгом времени работы ЛП М1 может сойтись в точку глобального минимума, но неизвестно, сколько итераций ему понадобится для этого.

Таблица 3.1. Результаты поиска глобального минимума функции невязки в четырехмерной области с центром в сдвинутой оптимальной точке в окрестности 0.1.

Смещение оптимальной точки	10^{-8}	10^{-7}	10^{-6}	10^{-5}	10^{-4}	10^{-3}
Невязка	10^{-14}	10^{-12}	10^{-10}	10^{-8}	10^{-6}	10^{-4}
Количество итераций	96	96	96	97	4000	4000
Расстояние между решениями	10^{-8}	10^{-7}	10^{-6}	10^{-5}	10^{-4}	10^{-3}
Флаг сходимости	1	1	1	1	0	0

Таблица 3.2. Результаты поиска глобального минимума функции невязки в четырехмерной области с центром в точном решении.

Диаметр области поиска	0.3	0.31	0.32	0.33	0.34	0.35
Количество итераций	27.0	177.0	94.0	305.0	295.0	87.0
Флаг сходимости	1	1	1	1	1	1

Диаметр области поиска	0.36	0.37	0.38	0.39	0.4	0.41
Количество итераций	651.0	2.0	42.0	124.0	88.0	11.0
Флаг сходимости	1	1	1	1	1	1

Диаметр области поиска	0.42	0.43	0.44	0.45	0.46	0.47
Количество итераций	158.0	275.0	78.0	98.0	175.0	364.0
Флаг сходимости	1	1	1	1	1	1

Таблица 3.3. Результаты поиска глобального минимума функции невязки в четырехмерной области с центром в начале координат.

Диаметр области поиска	0.006	0.007	0.008	0.009	0.01	0.011
Количество итераций	4000.0	4000.0	4000.0	4000.0	4000.0	4000.0
Флаг сходимости	0	0	0	0	0	0

Диаметр области поиска	0.012	0.013	0.014	0.015	0.016	0.017
Количество итераций	4000.0	4000.0	4000.0	4000.0	4000.0	4000.0
Флаг сходимости	0	0	0	0	0	0

Диаметр области поиска	0.018	0.019	0.02	0.021	0.022	0.023
Количество итераций	4000.0	4000.0	4000.0	4000.0	4000.0	4000.0
Флаг сходимости	0	0	0	0	0	0

Таблица 3.4. Результаты поиска глобального минимума функции невязки в трехмерной области с центром в точном решении.

Диаметр области поиска	0.3	0.31	0.32	0.33	0.34	0.35
Количество итераций	1116.0	369.0	540.0	956.0	4000.0	4000.0
Флаг сходимости	1	1	1	1	0	0

Диаметр области поиска	0.36	0.37	0.38	0.39	0.4	0.41
Количество итераций	1430.0	234.0	128.0	4000.0	614.0	4000.0
Флаг сходимости	1	1	1	0	1	0

Диаметр области поиска	0.42	0.43	0.44	0.45	0.46	0.47
Количество итераций	4000.0	1518.0	4000.0	616.0	347.0	3681.0
Флаг сходимости	0	1	0	1	1	1

Таблица 3.5. Результаты поиска глобального минимума функции невязки в трехмерной области с центром в начале координат.

Диаметр области поиска	0.006	0.007	0.008	0.009	0.01	0.011
Количество итераций	52.0	46.0	69.0	48.0	99.0	98.0
Флаг сходимости	0	0	0	0	0	0

Диаметр области поиска	0.012	0.013	0.014	0.015	0.016	0.017
Количество итераций	15.0	55.0	62.0	68.0	68.0	69.0
Флаг сходимости	0	0	0	0	0	0

Диаметр области поиска	0.018	0.019	0.02	0.021	0.022	0.023
Количество итераций	40.0	65.0	77.0	83.0	63.0	56.0
Флаг сходимости	0	0	0	0	0	0

Таблица 3.6. Результаты поиска глобального минимума функции невязки в двумерной области с центром в точном решении.

Диаметр области поиска	0.3	0.31	0.32	0.33	0.34	0.35
Количество итераций	134.0	250.0	29.0	26.0	26.0	26.0
Флаг сходимости	1	1	1	1	1	1

Диаметр области поиска	0.36	0.37	0.38	0.39	0.4	0.41
Количество итераций	26.0	26.0	20.0	158.0	233.0	380.0
Флаг сходимости	1	1	1	1	1	1

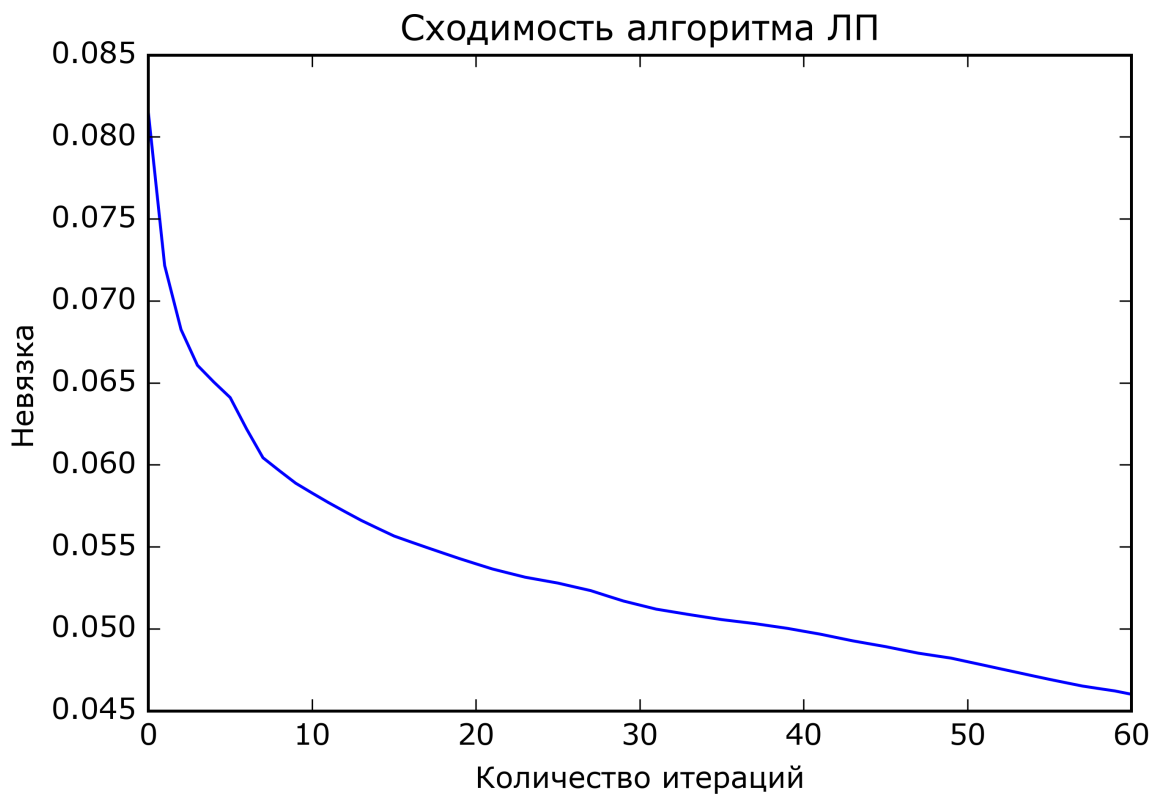
Диаметр области поиска	0.42	0.43	0.44	0.45	0.46	0.47
Количество итераций	231.0	98.0	19.0	281.0	183.0	513.0
Флаг сходимости	1	1	1	1	1	1

Таблица 3.7. Результаты поиска глобального минимума функции невязки в двумерной области с центром в начале координат.

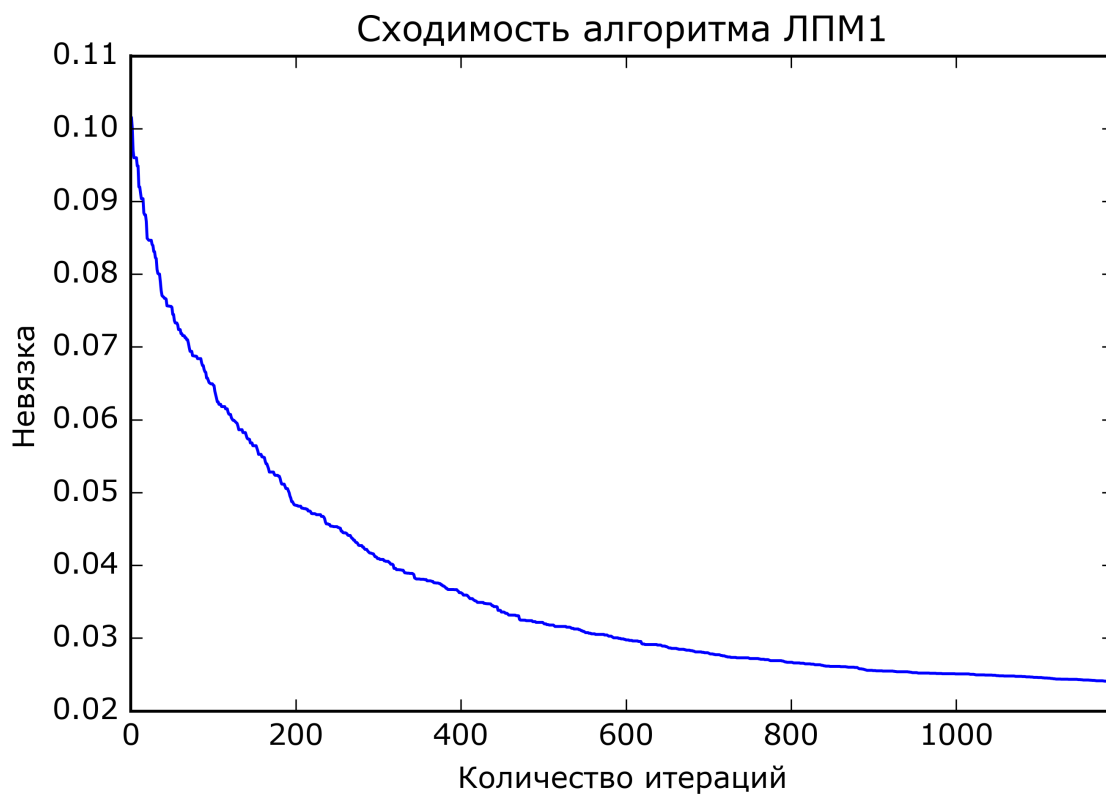
Диаметр области поиска	0.006	0.007	0.008	0.009	0.01	0.011
Количество итераций	4000.0	4000.0	4000.0	4000.0	4000.0	4000.0
Флаг сходимости	0	0	0	0	0	0

Диаметр области поиска	0.012	0.013	0.014	0.015	0.016	0.017
Количество итераций	4000.0	4000.0	4000.0	4000.0	4000.0	4000.0
Флаг сходимости	0	0	0	0	0	0

Диаметр области поиска	0.018	0.019	0.02	0.021	0.022	0.023
Количество итераций	4000.0	4000.0	4000.0	4000.0	4000.0	4000.0
Флаг сходимости	0	0	0	0	0	0



(а) Результаты работы ЛП



(б) Результаты работы ЛПМ1

Рис. 3.1. ЛП и его модификации.

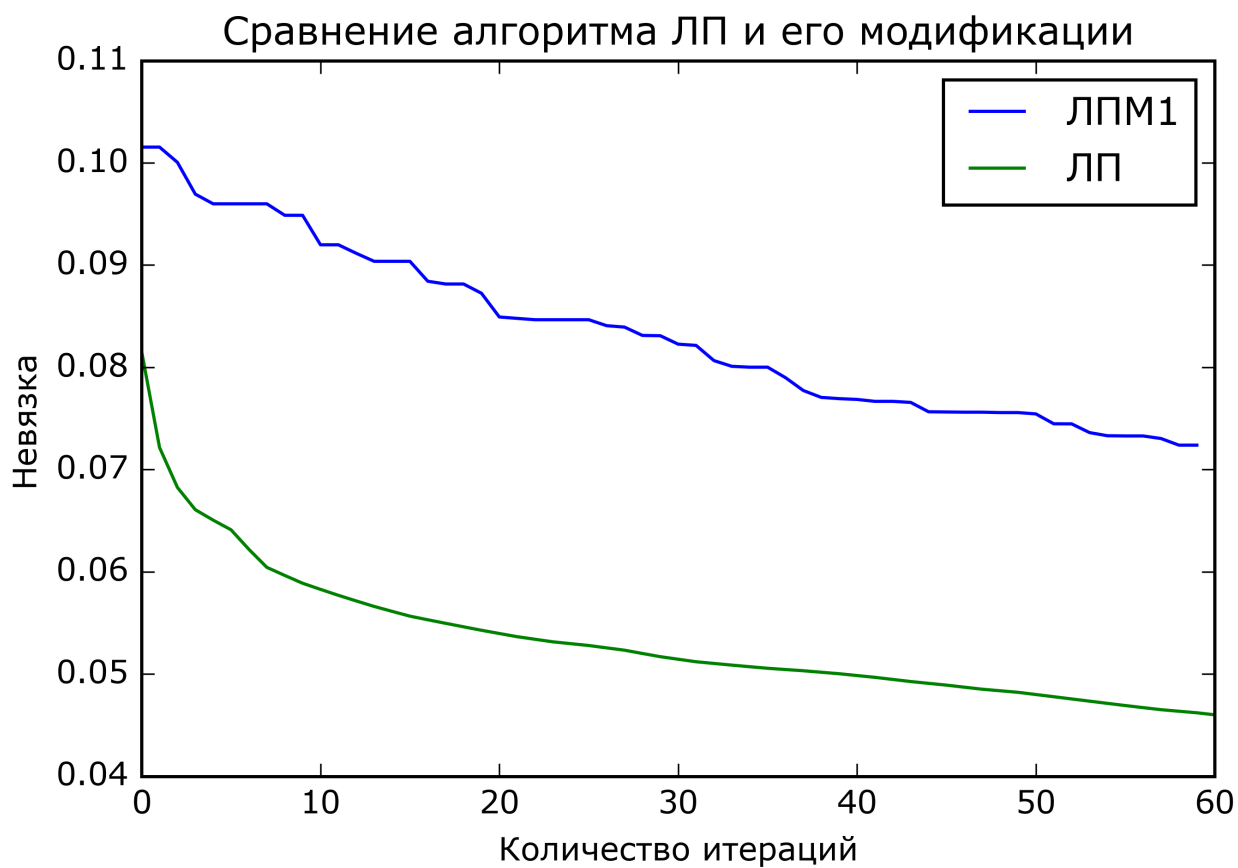


Рис. 3.2. Сравнение работы ЛП и модификации после 60 итераций.

Заключение

Настоящая работа является первой попыткой применения методики глобальной оптимизации с помощью заполняющих пространство кривых к задачам нахождения оптимальных космических траекторий. Рассмотрена проблема перелета космического аппарата между двумя компланарными (лежащими в одной плоскости) орбитами. Необходимые условия оптимальности, дающиеся принципом максимума, сводят задачу поиска оптимальной по затратам топлива траектории к двухточечной краевой задаче. Вектор невязок очень чувствителен к выбору начальных значений сопряжённых переменных. Задача минимизации его нормы в соответствующем четырёхмерном пространстве решается сведением к задаче нахождения глобального минимума функции одной переменной.

В ходе исследований был испробован один из методов сведения многомерной задачи оптимизации к одномерной, использующий кривые, заполняющие пространство, после чего применялся алгоритм решения задачи глобальной одномерной оптимизации. Вдобавок к этому был испробован метод линейного поиска.

При решении нашей задачи, подход, использующий кривые Пеано показал себя достаточно эффективным в случаях, когда кривая проходит через область притяжения максимума, что делает его пригодным для решения подобного типа задач, но ввиду ограничения на порядок кривой M его нужно доработать.

При использовании же линейного поиска значение функционала в ходе работы алгоритма уменьшается, но неизвестно, сколько итераций должно пройти для того, чтобы он сошелся.

На основании проведенного исследования можно сделать следующие выводы:

1. Ограничение на порядок кривой M значительно влияет на сходи-

мость алгоритма, что сильно усложняет задачу. Но и это ограничение можно попробовать обойти распараллеливанием алгоритма. Например, идея разбиения пространства на какое-то количество гиперкубов, в каждом из которых будет запускаться оптимизационный процесс, а далее по всем найденным значениям минимумов в этих кубиках находить минимальное, является достаточно привлекательной для ее реализации на суперкомпьютере.

2. Алгоритм одномерной оптимизации, лежащий в его основе можно еще больше ускорить (в смысле уменьшения количества итераций) и при более сложных экспериментах добиться значительного прироста производительности.
3. Линейный поиск требует значительно больше времени для сходимости, нежели подход, основанный на кривых Пеано, в связи с чем его эффективность под большим вопросом.

В заключении хотелось бы отметить, что исследованный в работе подход сведения задачи многомерной оптимизации к многомерной с помощью кривых, заполняющих пространство достаточно перспективен, хоть и зависит от конкретной задачи, а именно от того, насколько широка область притяжения глобального минимума функционала невязки, но и это, предположительно, можно будет обойти с помощью параллельных вычислений.

Список литературы

1. Важенин Н.А., Обухов В.А., Плохих А.П., Попов Г.А.
Электрические ракетные двигатели космических аппаратов и их влияние на радиосистемы космической связи. М.: ФИЗМАТЛИТ, 2012. 432 с.
2. Петухов В.Г. *Оптимизация траекторий и эволюция движения космических аппаратов с двигательными установками малой тяги*, кандидатская диссертация, Московский государственный авиационный институт (технический университет), 1996.
3. Nelder, J.A. and Mead, R. *A simplex method for function minimization*, Computer Journal, 1965, Vol. 7, No. 4, pp. 308–313.
4. Kennedy J. and Eberhart R. *Particle swarm optimization*, 1995 IEEE International Conference on Neural Networks, Perth, WA, Australia, 27 Nov.-1 Dec. 1995, Vol. 4, pp. 1942–1948.
5. The Accelerated and Modified Newton Methods.
URL: <http://mathfaculty.fullerton.edu/mathews/n2003/newtonacceleratemod.html> (дата обращения: 22.06.2018).
6. Кантор Г. *Труды по теории множеств*: пер. с нем. Ф. А. Медведева и А. П. Юшкевича, отв. редакторы А. Н. Колмогоров и А. П. Юшкевич. — М.: Наука, 1985. с. 22-35.
7. Peano G. *Sur une courbe, qui remplit toute une aire plane*, Mathematische Annalen. — 1890. — Т. 36, вып. 1. — с. 157–160.
8. Hilbert D. *Über die stetige Abbildung einer Line auf ein Flächenstück*, Mathematische Annalen. — 1891. — Т. 38, вып. 3. — с. 459–460.
9. Романко В.К. *Курс дифференциальных уравнений и вариационного исчисления* 2-е изд. — М.: Лаборатория Базовых Знаний, 2001. — 344 с.
10. Понтрягин Л.С., Болтянский В.Г., Гамкрелидзе Р.В., Мищенко

Е.Ф. *Математическая теория оптимальных процессов* 4-е изд., стереотипное. — М.: Наука, 1983. — 393 с.

11. Пиявский С. А. *Один алгоритм отыскания абсолютного экстремума функций*, Журнал вычислительной математики и математической физики, т. 12, №4 (1972), стр. 885—896.
12. Sergeyev, Y.D., Strongin, R.G., and Lera, D. *Introduction to Global Optimization Exploiting Space-Filling Curves*, Springer Briefs in Optimization. Springer Verlag New-York, 2013. 125 p.