

Министерство образования и науки
Российской Федерации

**Федеральное государственное автономное образовательное
учреждение высшего образования**
«Новосибирский национальный исследовательский государственный
университет»

Механико-математический факультет

Выпускная квалификационная работа на соискание
степени бакалавра

Кафедра математического моделирования

Щербина Полина Алексеевна

**Метод граничных интервалов
в свободной системе компьютерной математики Scilab**

Научный руководитель:
доктор физико-математических
наук С. П. Шарый

Новосибирск, 2016 г.

Содержание

1	Основы метода граничных интервалов	5
1.1	Граничные неравенства и их опоры в \mathbb{R}^n	5
1.2	Граничный интервал в \mathbb{R}^2	6
1.2.1	Определение понятия «граничный интервал»	6
1.2.2	Числовое представление граничного интервала	7
1.3	Применение метода граничных интервалов в пространстве \mathbb{R}^3 .	9
2	Описание пакета <code>lineqs</code>	11
2.1	Псевдокод алгоритма для двумерного случая	11
2.2	Псевдокод алгоритма для трёхмерного случая	17
2.3	Краткое описание функций пакета <code>lineqs</code>	17
3	Примеры работы пакета <code>lineqs</code>	20
3.1	Двумерный случай	20
3.2	Трёхмерный случай	22
4	Сравнение метода граничных интервалов и пакета <code>lineqs</code> с аналогами	23
4.1	Исключающий метод решения системы линейных неравенств Фурье-Моцкина	23
4.2	Применение метода граничных интервалов для решения задачи линейного программирования	23
4.2.1	Постановка задачи линейного программирования	23
4.2.2	Описание работы метода граничных интервалов для за- дачи линейного программирования	24
4.2.3	Симплекс-метод	26
4.3	Функция <code>inequal</code> в MAPLE	27

Введение

Визуализация решений систем линейных неравенств важна как самостоятельная математическая проблема, также потребность в ней может возникнуть при решении разнообразных задач. Например:

- помощь в системах поддержки принятия решений: наглядная презентация всех возможных вариантов решения [7],
- наглядное средство в образовательном процессе при изучении тем «Линейные неравенства», «Линейное программирование» и др.,
- визуализация сложных множеств решений различных математических задач исследования операций и т.п., например, наглядное представление множеств решений интервальных линейных систем уравнений.

У существующих методов решения и визуализации систем линейных неравенств есть недостатки:

1. Плохая обработка неограниченного множества решений.
2. Многие из существующих систем ориентированы на визуализацию только квадратных систем линейных неравенств.
3. В основном, плохая отрисовка тощих (с пустой внутренностью) множеств решений.
4. Значительная часть существующих программ визуализации является платной или требует лицензии.

Метод граничных интервалов — это метод исследования и визуализации полиэдральных множеств в евклидовых пространствах \mathbb{R}^2 и \mathbb{R}^3 , основанный на вычислении и использовании специальной матрицы граничных интервалов для системы линейных неравенств [1].

Метод, указанный выше, был создан Ириной Александровной Шарой и представлен в виде пакета `lineq`, который визуализирует множества решений систем линейных неравенств с двумя и тремя неизвестными. Этот пакет решает проблему пунктов 1–3, но написан в системе MATLAB, которая является платной.

Возникла естественная потребность реализовать метод граничных интервалов в свободных системах компьютерной математики. В качестве первой такой системы был выбран Scilab. В рамках данной работы были поставлены следующие цели:

- Реализация метода граничных интервалов и алгоритма визуализации решений систем линейных неравенств в свободной системе компьютерной математики Scilab.
- Описание реализации написанных алгоритмов.
- Сравнение метода граничных интервалов и его реализации, созданной в рамках данной работы, с существующими аналогами.

1 Основы метода граничных интервалов

В данном разделе приведено краткое описание метода граничных интервалов, сделанное на основе [1]. Более строгое обоснование утверждений ниже можно найти в [1].

1.1 Граничные неравенства и их опоры в \mathbb{R}^n

Пусть задана система линейных неравенств

$$Ax \geq b, \quad A \in \mathbb{R}^{n \times m}, \quad x \in \mathbb{R}^n, \quad b \in \mathbb{R}^m, \quad m, n \in \mathbb{N} \quad (1)$$

относительно неизвестного x . Обозначим множество её решений через H , i -ю строку матрицы A через A_i , а множество решений неравенства $A_i x \geq b_i$, записанного в i -й строке системы, через H_i .

Множество решений системы (1) получается из пересечения множества решений отдельных неравенств $A_i x \geq b_i$, $i = 1, \dots, m$. Поэтому к его внутренним точкам относятся те точки H , которые являются внутренними для множеств решений H_i всех неравенств системы, а к граничным — те точки H , которые лежат на границе хотя бы одного из H_i . Запишем последнее утверждение формально и преобразуем:

$$\partial H = H \cap \left(\bigcup_i \partial H_i \right) = \bigcup_i (H \cap \partial H_i). \quad (2)$$

Таким образом, граница множества решений системы (1) состоит из вкладов $H \cap \partial H_i$, которые вносятся отдельными неравенствами.

Для системы (1) неравенство $A_i x \geq b_i$, записанное в i -ой строке, называется *граничным*, если оно задает опорное полупространство¹ к множеству решений H . Если i -е неравенство граничное, то пересечение гиперплоскости $A_i x = b_i$ с множеством H называется *опорой* i -го неравенства и обозначается S_i . [1]

Обозначим множество индексов граничных неравенств посредством I_b , из

¹Полупространство Ψ называется опорным к замкнутому подмножеству Ω , если Ψ содержит Ω , а граница полупространства $\partial\Psi$ имеет с Ω хотя бы одну общую точку.

(2) получаем:

$$\partial H = \bigcup_{i \in I_b} (H \cap \partial H_i) = \bigcup_{i \in I_b} S_i,$$

то есть граница множества решений состоит из опор граничных неравенств.

1.2 Граничный интервал в \mathbb{R}^2

1.2.1 Определение понятия «граничный интервал»

Понятие «граничного интервала» вводится для системы линейных неравенств вида

$$Ax \geq b, \quad A \in \mathbb{R}^{m \times 2}, \quad x \in \mathbb{R}^2, \quad b \in \mathbb{R}^m, \quad m \in \mathbb{N}, \quad (3)$$

относительно двух неизвестных x_1 и x_2 . Оно включает в себя три составляющих: индекс, опора и направление.

Индекс граничного интервала фиксирует связь интервала с неравенством из системы (3). Граничные интервалы определены только для граничных неравенств системы (3). Каждое граничное неравенство системы (3) порождает один граничный интервал. *Индекс граничного интервала* — это номер строки системы (3), в которой записано граничное неравенство, порождающее данный граничный интервал.

Опора представляет граничный интервал как множество точек плоскости. *Опорой граничного интервала с индексом i* будем называть опору i -го неравенства системы (3), то есть пересечение прямой $A_i x = b_i$ с множеством решений H системы (3). И прямая, и множество H выпуклы и замкнуты, поэтому опорой может служить только выпуклое и замкнутое подмножество прямой. А поскольку опора не бывает пустым множеством, опоры граничных интервалов делятся на четыре типа: точка, отрезок, луч и прямая.

На рисунке 1 приведены примеры всех видов опор для граничного интервала с индексом 1. Множество решений H выделено штриховкой, а опора S_1 показана как утолщённая часть прямой $A_1 x = b_1$

Направление граничного интервала с индексом i задаёт такое движение по прямой $A_i x = b_i$, при котором полуплоскость $A_i x \geq b_i$ остаётся справа. Множество решений H системы (3) удовлетворяет неравенству $A_i x \geq b_i$ и, значит, целиком лежит в полуплоскости, задаваемой этим неравенством.

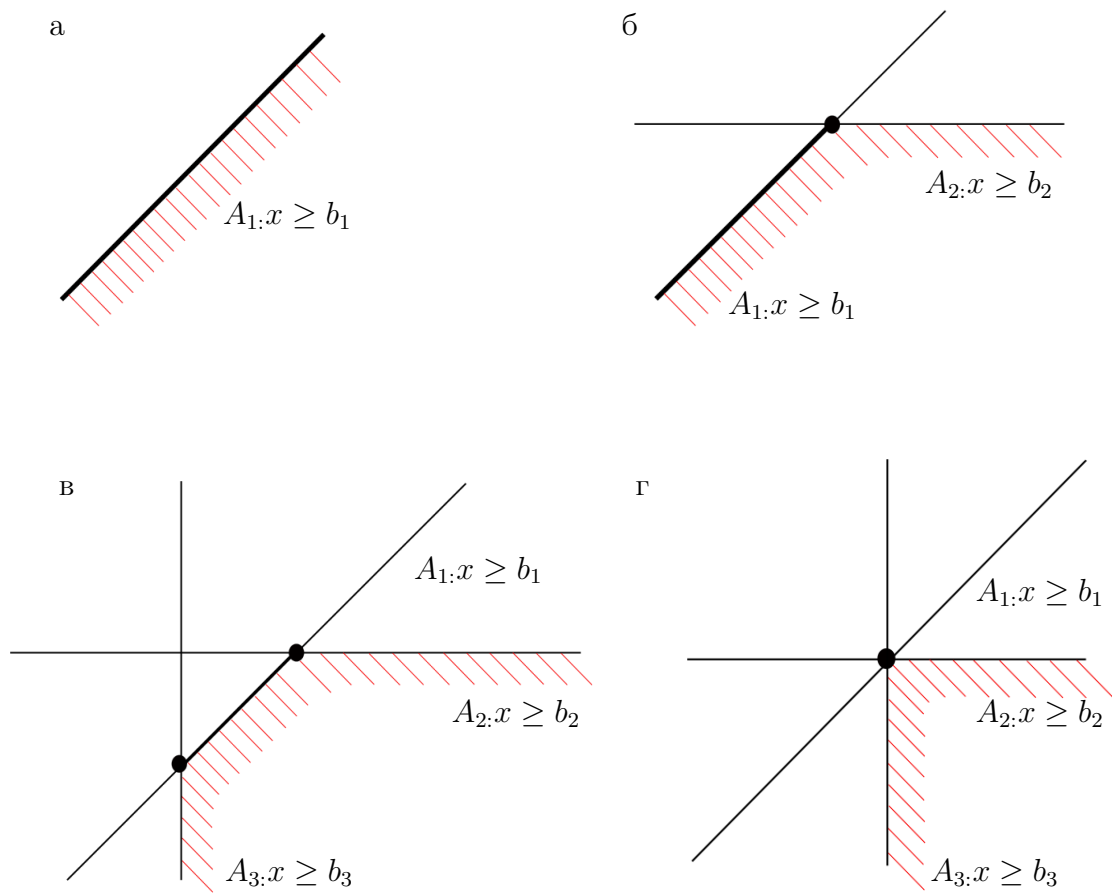


Рис. 1: Типы опор границного интервала: а) прямая б) луч в) отрезок г) точка

Поэтому можно считать: когда мы движемся по прямой $A_i: x = b_i$ в направлении границного интервала с индексом i , множество решений H тоже остаётся справа. Численное описание направления границного интервала задается вектором $(-A_{i2}, A_{i1})^T$.

Когда направление прямой $A_i: x = b_i$ фиксировано, для опоры границного интервала можно естественным образом определить понятия старта и финиша (рис. 2). Старт и финиш границного интервала будем называть его концевыми точками, или концами.

1.2.2 Числовое представление границного интервала

Общий вид числового представления границного интервала. Используя координаты расширенной числовой плоскости (включающие $\pm\infty$), представим границный интервал как упорядоченную пятерку чисел: два пер-

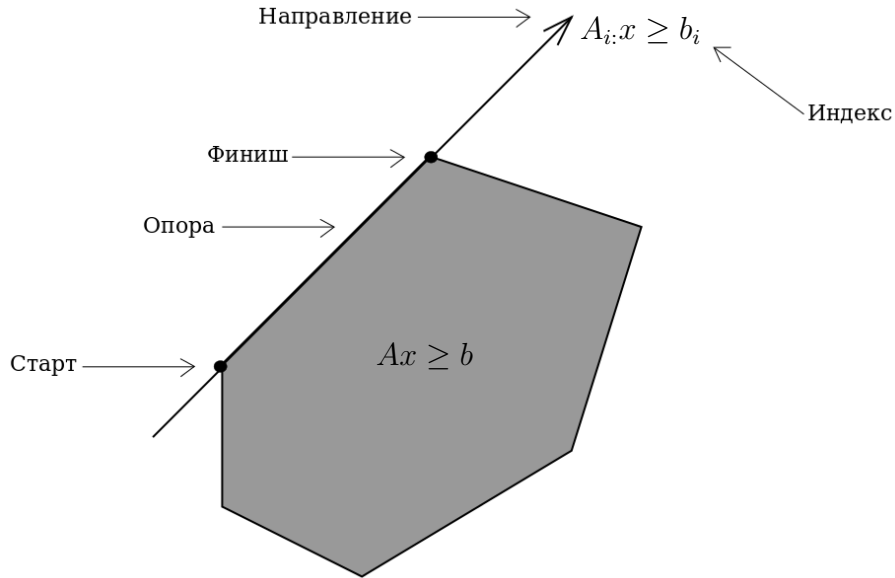


Рис. 2: Наглядное представление граничного интервала

вых числа — абсцисса и ордината старта, затем в том же порядке координаты финиша и, наконец, пятое число — индекс граничного интервала:

$$\left(\underbrace{(*)}_{\text{старт} \in \overline{\mathbb{R}}^2} \quad \underbrace{(*)}_{\text{финиш} \in \overline{\mathbb{R}}^2} \quad i \right) \quad (4)$$

индекс $\in \mathbb{N}$

где $\overline{\mathbb{R}}^2$ — расширенная числовая плоскость.

Вычисление граничных интервалов. Если $A_i = (0, 0)$, то неравенство не порождает граничный интервал (см. [1]).

Рассмотрим случай $A_i \neq (0, 0)$. Для вычисления граничного интервала нам необходимо ввести свою внутреннюю систему координат на прямой $A_i x = b_i$.

- В качестве начала координат \tilde{O} внутренней системы берём проекцию точки $(0, 0)$ исходной системы координат Ox_1x_2 на прямую $A_i x = b_i$. Путём несложных преобразований получаем $\tilde{O} = b_i A_i^\top / \|A_i\|_2^2$, где $\|A_i\|_2 = \sqrt{A_{i1}^2 + A_{i2}^2}$ — 2-норма вектора.
- В качестве единичного вектора внутренней системы координат берём $(-A_{i2}, A_{i1})^\top$ (вектор направления на прямой $A_i x = b_i$).
- Внутреннюю координату обозначим y .

Учитывая сказанное выше, получаем для прямой $A_i x = b_i$ параметрическую

запись:

$$\frac{b_i}{\|A_{i:}\|_2^2} A_{i:}^\top + (-A_{i2}, A_{i1})^\top y, \quad y \in \mathbb{R}. \quad (5)$$

Найдем пересечение множества решений системы $Ax \geq b$ с прямой (5). Для этого в системе выполним замену:

$$x \longrightarrow \frac{b_i}{\|A_{i:}\|_2^2} A_{i:}^\top + (-A_{i2}, A_{i1})^\top y$$

После замены мы получим систему линейных неравенств с одним неизвестным y . Её решение уже не представляет трудностей. Множество решений этой системы представляет собой интервал $[\underline{y}, \bar{y}]$. Если множество решений пусто, то неравенство $A_{i:}x \geq b_i$ — не граничное и не порождает граничного интервала. Если же множество решений не пусто, неравенство $A_{i:}x \geq b_i$ — граничное, оно порождает граничный интервал с индексом i , для которого \underline{y} — координата старта, а \bar{y} — координата финиша в выбранной внутренней системе координат прямой $A_{i:}x = b_i$.

Запишем старт и финиш граничного интервала во внешней системе координат:

$$\text{старт} = \frac{b_i}{\|A_{i:}\|_2^2} A_{i:} + (-A_{i2}, A_{i1}) \underline{y}, \quad (6)$$

$$\text{финиш} = \frac{b_i}{\|A_{i:}\|_2^2} A_{i:} + (-A_{i2}, A_{i1}) \bar{y}. \quad (7)$$

1.3 Применение метода граничных интервалов в пространстве \mathbb{R}^3

Аналогично тому, как случай двумерного пространства сводится к одномерному, трехмерная задача сводится к двумерной, для которой уже известен способ нахождения граничных интервалов.

Рассмотрим систему линейных неравенств:

$$Ax \geq b, \quad A \in \mathbb{R}^{m \times 3}, \quad x \in \mathbb{R}^3, \quad b \in \mathbb{R}^m, \quad m \in \mathbb{N}. \quad (8)$$

В случае \mathbb{R}^3 множество решений i -го неравенства H_i представляет собой уже полупространство, а его граница ∂H_i — плоскость. Обозначим через \tilde{H}_i вклад i -го неравенства в ∂H . H_i — это полиэдр, представляющий собой

пересечение плоскости $A_i:x = b_i$ с множеством решений H :

$$\tilde{H}_i = \{x \in \mathbb{R}^3 \mid A_i:x = b_i, Ax \geq b\} \quad (9)$$

Для того, чтобы определить параметры множества H (ограниченность, множество вершин и т.п.) достаточно определить параметры каждого H_i . А каждое множество H_i можно описать с помощью метода граничных интервалов. Для этого достаточно перейти во внутреннюю систему координат плоскости $A_i:x = b_i$, в которой множество H_i задаётся системой вида (3).

2 Описание пакета `lineqs`

В рамках работы был написан пакет программ `lineqs`, основанный на методе граничных интервалов и реализующий задачу визуализации множеств решений систем линейных неравенств для двумерного и трёхмерного случаев.

2.1 Псевдокод алгоритма для двумерного случая

Алгоритм работы пакета `lineqs` для двумерного случая разделен на четыре логические части:

- формирование матрицы граничных интервалов,
- формирование новой матрицы граничных интервалов с учетом окна отрисовки,
- построение обхода вершин полиэдра решений,
- отрисовка решения с помощью встроенных функций Scilab'a.

Формирование матрицы граничных интервалов.

Задача алгоритма состоит в том, чтобы из системы линейных неравенств сформировать матрицу граничных интервалов в виде (4). В таблице 1 представлен псевдокод реализованного алгоритма решения этой задачи. Ниже приведено пояснение к данному псевдокоду.

На вход подается матрица A , в которой отсутствуют нулевые и повторяющиеся строки, и вектор b . Матрица A должна содержать как минимум одну строчку и 2 или 3 столбца (в зависимости от размерности решаемой задачи). Вектор b должен соответствовать матрице A и иметь такое же количество строк.

На выходе мы хотим получить готовую матрицу граничных интервалов S , которая содержит координаты старта и финиша, а также индексы граничных интервалов, упорядоченных в соответствии с видом (4).

Шаг алгоритма заключается в последовательном рассмотрении неравенств системы. На каждой прямой вида $A_i \cdot x = b_i$ вводится собственная система координат, после чего рассматривается сужение остальных неравенств на эту прямую. Шаг алгоритма включает в себя 4 действия:

Таблица 1: Алгоритм, формирующий матрицу граничных интервалов для случая двух переменных

<p>Вход</p> <p>Система линейных неравенств $Ax \geq b$, которая задаётся матрицей A и вектором правой части b. В матрице A отсутствуют нулевые и повторяющиеся строки.</p>
<p>Выход</p> <p>Матрица граничных интервалов.</p>
<p>Алгоритм</p> <p>$S \leftarrow$ ⟨пустая матрица⟩</p> <p>DO $i = 1$, ⟨число строк в A⟩</p> <p style="padding-left: 2em;">% Рассматриваем прямую $A_i: x = b$</p> <p style="padding-left: 2em;">$x_0 \leftarrow$ произвольная точка на прямой</p> <p style="padding-left: 2em;">$p \leftarrow (-A_{i2}, A_{i1})$ % вектор направления прямой</p> <p style="padding-left: 2em;">% $x(y) = x_0 + py$ — параметризация прямой</p> <p style="padding-left: 2em;">$q \leftarrow [-\infty, \infty]$ % участок активности i-го неравенства</p> <p style="padding-left: 2em;">DO $k = 1$, ⟨число строк в A⟩</p> <p style="padding-left: 4em;">% Рассматриваем сужение неравенства $A_k: x \geq b_k$ на прямую</p> <p style="padding-left: 4em;">$[y, \bar{y}] \leftarrow$ решение неравенства $A_k: (x_0 + py) \geq b_k$</p> <p style="padding-left: 4em;">% $[y, \bar{y}]$ может быть и пустым</p> <p style="padding-left: 4em;">$q \leftarrow q \cap [y, \bar{y}]$</p> <p style="padding-left: 2em;">END DO</p> <p style="padding-left: 2em;">$[q, \bar{q}] \leftarrow q$ % интервал активности на прямой</p> <p style="padding-left: 2em;">% Переходим в исходную систему координат:</p> <p style="padding-left: 2em;">$x_1 \leftarrow x_0 + pq$</p> <p style="padding-left: 2em;">$x_2 \leftarrow x_0 + p\bar{q}$</p> <p style="padding-left: 2em;">$S \leftarrow \begin{pmatrix} S \\ x_1 & x_2 & i \end{pmatrix}$ % Добавляем строчку к матрице S</p> <p>END DO</p> <p>% S — матрица граничных интервалов</p>

1. Рассматриваем текущую строку i матрицы A и вводим на соответствующей прямой $A_i: x = b_i$ собственную систему координат в соответствии с формулой (5).
2. Сужаем систему неравенств на параметризованную i -ю прямую, то есть решаем неравенства вида $A_k: (x_0 + py) \geq b_k$, где x_0 — произвольная точка на прямой $A_i: x = b_i$, $p = (-A_{i2}, A_{i1})$ — направление прямой, а y — новая координата. Решение — координаты старта и финиша граничного интервала, которые находятся по формулам (6) и (7). В случае беско-

нечной границы задаем в качестве координат старта или финиша $\pm\infty$ с соответствующим знаком.

3. Переходим в исходную систему координат.
4. Добавляем новую строку в матрицу S .

Следует обратить внимание, что решение неравенства из пункта 2: $A_k \cdot (x_0 + py) \geq b_k$ может оказаться тривиальным, то есть $A_k \cdot p = 0$. В случае, когда $A_k \cdot x_0 > b_k$ решением является вся числовая плоскость и неравенство не имеет интервала активности на прямой. Если же $A_k \cdot x_0 < b_k$, то может быть два случая:

1. $A_k \cdot A_i > 0$. Этот случай соответствует рисунку 3а и означает, что i -е неравенство не имеет участка активности.
2. $A_k \cdot A_i < 0$. Этот случай соответствует рисунку 3б и означает, что неравенства i и k несовместны.

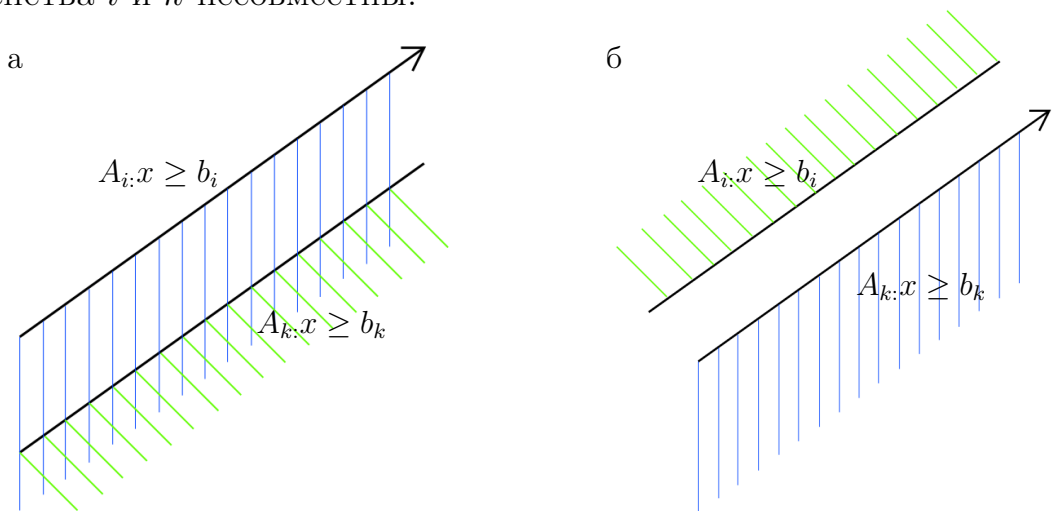


Рис. 3: Тривиальный случай сужения k -го неравенства на i -е: а — при $A_k \cdot A_i > 0$, б — при $A_k \cdot A_i < 0$

Формирование новой матрицы граничных интервалов, с учётом окна отрисовки.

Задача алгоритма: имея матрицу граничных интервалов определить окно отрисовки и, учитывая брус обрезки, сформировать новую матрицу граничных интервалов. Брус обрезки выбирается в соответствии с окном отрисовки.

На вход подается матрица граничных интервалов S , полученная в предыдущей части алгоритма.

Таблица 2: Алгоритм, формирующий новую матрицу граничных интервалов, с учётом окна отрисовки, для двумерного случая.

<p>Вход</p> <p>Матрица граничных интервалов S.</p>
<p>Выход</p> <p>Новая матрица граничных интервалов M, окно отрисовки W.</p>
<p>Алгоритм</p> $PP \leftarrow \begin{pmatrix} S_{11} & S_{12} \\ \vdots & \vdots \\ S_{m1} & S_{m2} \end{pmatrix} \quad \% \text{ Матрица из стартов граничных интервалов}$ <p>Удаляем из PP строки с бесконечными элементами. IF $\langle \text{число строк в } PP = 0 \rangle$ $PP \leftarrow$ проекции $(0, 0)$ на все граничные интервалы. ENDIF $\underline{x} \leftarrow \min_i PP_{i1}$ $\bar{x} \leftarrow \max_i PP_{i1}$ $\underline{y} \leftarrow \min_i PP_{i2}$ $\bar{y} \leftarrow \max_i PP_{i2}$ $C \leftarrow \left(\frac{\underline{x} + \bar{x}}{2}, \frac{\underline{y} + \bar{y}}{2} \right)$ $R \leftarrow 2 \cdot \max \left(\frac{\bar{x} - \underline{x}}{2}, \frac{\bar{y} - \underline{y}}{2} \right)$ IF $R = 0$ $R \leftarrow 1$ ENDIF $W \leftarrow \begin{pmatrix} C_1 - R & C_2 - R \\ C_1 + R & C_2 + R \end{pmatrix} \quad \% W - \text{границы окна отрисовки}$ $B \leftarrow W \quad \% B - \text{брус обрезки}$ IF $\langle \text{в } S \text{ есть бесконечные интервалы} \rangle$ Добавляем в систему $Ax \geq b$ четыре новых неравенства, задающих B: $x \geq B_{11}; \quad y \geq B_{12};$ $-x \geq -B_{21}; \quad -y \geq -B_{22}.$ Выполняем алгоритм из таблицы 1. $M \leftarrow$ новая матрица граничных интервалов. ELSE $M \leftarrow S$ ENDIF</p>

На выходе мы получаем новую матрицу граничных интервалов M , которая сформирована с учетом окна отрисовки W .

Формируем матрицу точек ориентации следующим образом:

1. Создаем матрицу PP из первых двух столбцов матрицы S , то есть из координат стартов граничных интервалов;

2. Удаляем из матрицы PP строки, которые содержат бесконечные элементы;
3. Если в матрице PP после второго пункта не осталось строк, то добавляем в PP проекции точки $(0, 0)$ на i -ый граничный интервал, $i = 1, \dots, m$, по следующей формуле: $b_i A_{i:}^T / \|A_{i:}\|_2^2$.

Находим интервальную оболочку точек ([4], стр. 83-84) из матрицы PP . С помощью координат, задающих интервальную оболочку находим её центр C и радиус R . Если радиус оказался равен нулю (что соответствует случаю одной точки ориентации), то зададем радиус равным 1.

Задаем границы окна отрисовки W : отступаем от центра на расстояние радиуса. Брус обрезки B задаем равным окну отрисовки.

При наличии в матрице граничных интервалов S бесконечных координат, добавляем в систему $Ax \geq b$ новые неравенства задающие брус обрезки B . Повторяем алгоритм, приведенный в таблице 1. Таким образом получаем новую матрицу граничных интервалов, которая уже учитывает границу отрисовки множества решений H .

Построение обхода вершин полиэдра решений.

Задача алгоритма по матрице граничных интервалов построить обход вершин политопа.

На вход алгоритма подается полученная ранее матрица граничных интервалов.

На выходе мы хотим получить матрицу P , строки которой являются координатами вершин обхода политопа.

В качестве начала пути выбираем первую вершину из матрицы M , назовем её «рабочим стартом», и сразу записываем её в матрицу P .

Шаг алгоритма — поиск граничного интервала, старт которого совпадает с «рабочим стартом» и определение следующей вершины обхода. Алгоритм включает в себя 3 действия:

1. Ищем в матрице M строку j , для которой координаты старта совпадают с координатами «рабочего старта». Обозначим финиш (M_{j3}, M_{j4}) «рабочим финишем».

Таблица 3: Алгоритм, строящий обход вершин политопа решений для двумерного случая

<p>Вход</p> <p>Матрица граничных интервалов M.</p>
<p>Выход</p> <p>Матрица P, в которой строки состоят из координат вершин политопа и перечислены в порядке её обхода.</p>
<p>Алгоритм</p> <pre> $W \leftarrow M$ % W — рабочая матрица $B \leftarrow (M_{11}, M_{12})$ % B — начало пути $P \leftarrow B$ % в матрице P накапливается результат алгоритма $S \leftarrow B$ % S — «рабочий старт», текущая вершина пути DO WHILE (количество строк $W > 0$) Ищем такой индекс j, что $S = (W_{j1}, W_{j2})$ % W_j — это граничный интервал, который начинается с рабочего старта S $F \leftarrow (W_{j3}, W_{j4})$ % конец этого интервала, т.е. следующая вершина пути IF $S \neq F$ $P \leftarrow \begin{pmatrix} P \\ F \end{pmatrix}$ IF $F = B$ RETURN % вернулись в начало пути, % потому что обошли обошли вокруг политопа ELSE $S \leftarrow F$ % переходим к следующей вершине пути ENDIF ENDIF Удаляем j-ю строку из W. END DO </pre>

- Если «рабочий старт» и «рабочий финиш» совпадает, то значит опора граничного интервала из j -ой строки матрицы M является вершиной. Так как такой интервал не дает движения к новой вершине, в построении обхода он нам не нужен.
- В дальнейшем построении обхода j -ый граничный интервал нам уже не потребуется, поэтому удаляем j -ю строку из матрицы M .

Отрисовка

На этом этапе работы алгоритма у нас уже есть матрица вершин обхода политопа P и границы окна отрисовки W . Отрисовываем границу, в вершинах множества решения ставим точки и определяем окно отрисовки с помощью встроенной функции Scilab'a `plot2d` [6]. Заливаем область, ограниченную данными вершинами цветом, с помощью функции `xfpoly` [6].

2.2 Псевдокод алгоритма для трёхмерного случая

В рамках работы был реализован алгоритм визуализации решений систем линейных неравенств в трёхмерном пространстве. Краткий псевдокод алгоритма приведен в таблице 4.

Поскольку, как было упомянуто в разделе 1.3, трёхмерные системы сводятся к двумерным, приведенный псевдокод использует ранее описанные алгоритмы для двумерного случая.

2.3 Краткое описание функций пакета `lineqs`

ClearZeroRows

Удаляет нулевые строки во входной матрице A .

NonRepeat

Удаляет строки повторы во входной матрице A .

BoundaryIntervals

Формирует матрицу граничных интервалов. Находит пересечение прямых $A_i x = b_i$ с границей множества решений системы линейных неравенств H . Для каждой прямой такое пересечение либо пусто, либо интервал, который может быть представлен точкой, отрезком, лучом или прямой.

ParticularPoints

Формирует матрицу точек ориентации. Точки ориентации — это либо вершины H , либо проекции нуля на граничные интервалы, которые являются прямыми.

PicWindow

Выбирает окно отрисовки. Находит интервальную оболочку (см. [4]) множества точек ориентации. Выбирает окно отрисовки в соответствии с интервальной оболочкой.

Intervals2Path

Формирование матрицы обхода границы многогранника на основе матрицы граничных интервалов, у которой строки представляют последовательные

Таблица 4: Псевдокод алгоритма визуализации множества решений системы неравенств в R^3 на основе метода граничных интервалов

<p>Вход</p> <p>Система линейных неравенств $Ax \geq b$, которая задаётся матрицей A и вектором правой части b.</p>
<p>Выход</p> <p>Визуализация множества решений системы.</p>
<p>Алгоритм</p> <p>$PP \leftarrow$ пустая матрица точек ориентации. $CS \leftarrow$ исходная система координат. DO $i = 1$, \langleчисло строк в $A$$\rangle$ $CS' \leftarrow$ система координат в плоскости $A_i \cdot x = b_i$ A', b' — матрица и вектор системы A, b в CS' % рассматриваем двумерную систему неравенств на плоскости $A_i \cdot x = b_i$ IF \langleвсе элементы $A' = 0$$\rangle \wedge \langle$все элементы $b' \leq 0$$\rangle$ % решение — вся плоскость Добавить в PP начало CS' ELSE $S' \leftarrow$ матрица граничных интервалов системы $A'y \geq b'$ с помощью алгоритма из таблицы 1. $P' \leftarrow$ точки ориентации S' с помощью алгоритма из таблицы 2. $P \leftarrow$ построчно преобразовать P' из CS' в CS $PP \leftarrow \begin{pmatrix} PP \\ P \end{pmatrix}$ ENDIF END DO $W \leftarrow$ окно отрисовки для точек из P % аналогично таблице 2 % Добавим в систему неравенства, задающие окно отрисовки: $\tilde{A} \leftarrow \begin{pmatrix} A \\ E \\ -E \end{pmatrix} \quad \tilde{b} \leftarrow \begin{pmatrix} b \\ W_{1:}^T \\ W_{2:}^T \end{pmatrix}$ DO $i = 1$, \langleчисло строк в $\tilde{A}$$\rangle$ $CS' \leftarrow$ система координат плоскости $\tilde{A}_i \cdot x = \tilde{b}_i$ $\tilde{A}', \tilde{b}' \leftarrow$ матрица и вектор системы \tilde{A}, \tilde{b} в CS' $S' \leftarrow$ матрица граничных интервалов системы $\tilde{A}'y \geq \tilde{b}'$ IF $\langle S' \text{ не пуста} \rangle$ $L' \leftarrow$ обход S' с помощью алгоритма из таблицы 3. $L \leftarrow$ построчно преобразовать L' из CS' в CS % L задаёт вершины грани, соответствующей i-му неравенству Нарисовать грань, вершины которой задаются строками L. IF $i > \langle$число строк $A$$\rangle$ % «бесконечная грань», заданная неравенством обрезки. Покрасить грань в красный цвет. ELSE Покрасить грань в зелёный цвет. ENDIF ENDIF END DO END DO</p>

вершины обхода и каждая строка есть пара координат соответствующей вершины.

NonRepeatPoints

Удаление из матрицы вершин многогранника отрисовки повторяющихся точек.

SolSetIneq2D

Отрисовка множества решений H системы линейных неравенств $Ax \geq b$, $A \in \mathbb{R}^{m \times 2}$, $b \in \mathbb{R}^m$.

ChangeVariable

Переход от исходных координат к внутренним координатам y плоскости $A_i x = b$. (Трёхмерный случай)

SolSetIneq3D

Отрисовка множества решений H системы линейных неравенств $Ax \geq b$, $A \in \mathbb{R}^{m \times 3}$, $b \in \mathbb{R}^m$.

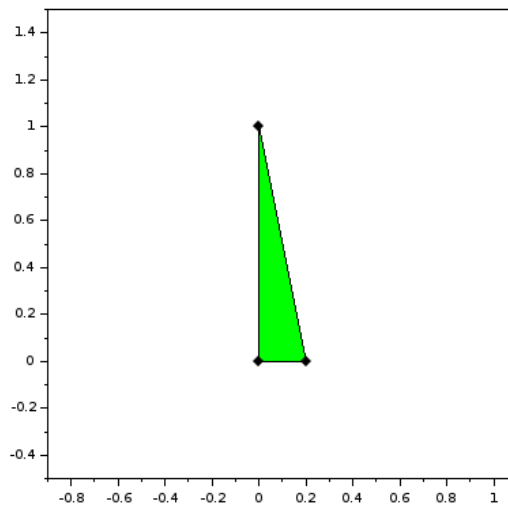
3 Примеры работы пакета `lineqs`

Ниже приведены примеры работы пакета `lineqs` для систем линейных неравенств с различными типами множеств решений в двумерном и трёхмерном случаях.

3.1 Двумерный случай

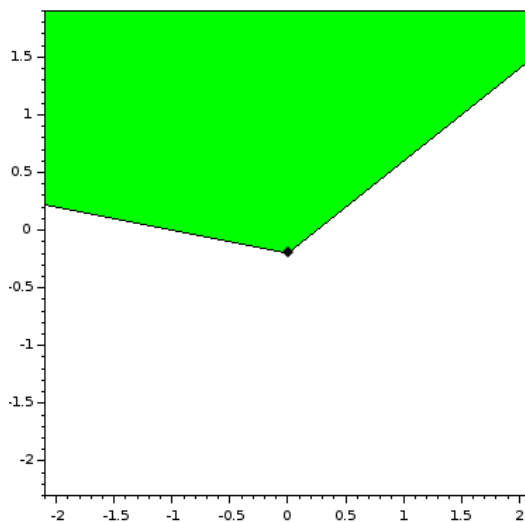
1. Ограниченное множество

$$A = \begin{pmatrix} -1 & 1 \\ -5 & -1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} -1 \\ -1 \\ 0 \\ 0 \end{pmatrix}$$



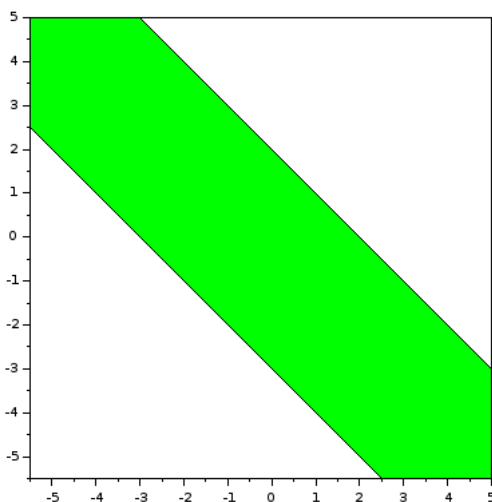
2. Неограниченное множество с одной вершиной

$$A = \begin{pmatrix} 1 & 5 \\ -4 & 5 \end{pmatrix}, \quad b = \begin{pmatrix} -1 \\ -1 \end{pmatrix}$$



3. Неограниченное множество

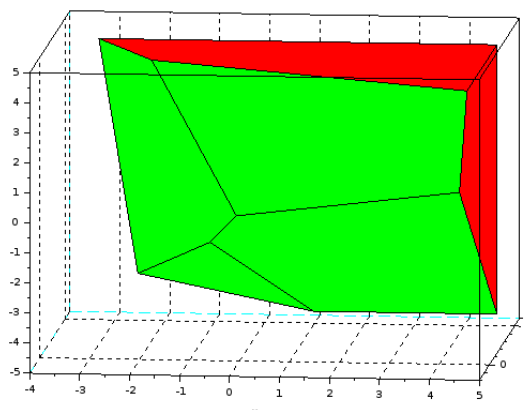
$$A = \begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix}, \quad b = (-3, -2)$$



3.2 Трёхмерный случай

1. Неограниченное множество

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 10 & 5 & 1 \\ 1 & 2 & -1 \\ 0 & 4 & 5 \end{pmatrix}, b = \begin{pmatrix} -2 \\ -1 \\ 0 \\ -1 \end{pmatrix}$$

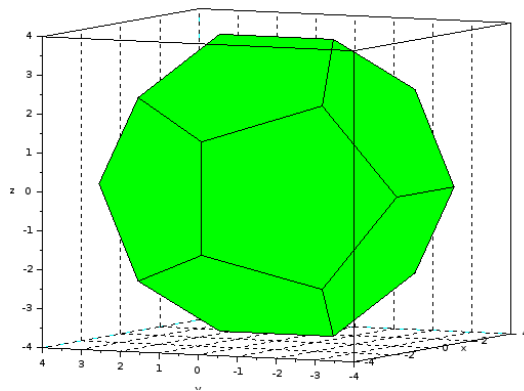


Красным отмечены бесконечные грани, зеленым - конечные.

2. Ограниченное множество

$$A = \begin{pmatrix} \pm\varphi^2 & \pm\varphi & 0 \\ 0 & \pm\varphi^2 & \pm\varphi \\ \pm\varphi & 0 & \pm\varphi^2 \end{pmatrix}, b = \begin{pmatrix} -10 \\ \vdots \\ -10 \end{pmatrix}$$

$$\varphi = \frac{1 + \sqrt{5}}{2}$$



4 Сравнение метода граничных интервалов и пакета `lineqs` с аналогами

В данном разделе приведено неформальное сравнение метода граничных интервалов с естественными конкурентами при решении систем линейных неравенств и смежных линейных задач в маломерных пространствах, такими как метод Фурье-Моцкина и симплекс метод.

4.1 Исключающий метод решения системы линейных неравенств Фурье-Моцкина

Подобно методу исключения для систем линейных уравнений, был предложен аналогичный метод для систем линейных неравенств в работах Фурье (1827) и Моцкина (1936). Алгоритм последовательно исключает по одной переменной, что приводит к системе с единственной неизвестной, решение которой уже не составляет труда. (Подробную историю вопроса смотри в [5])

Однако, в отличие от метода граничных интервалов, данный метод не дает ответ в виде границы геометрического множества решений, что усложняет задачу визуализации. Главным минусом метода Фурье-Моцкина является большая комбинаторная сложность — в худшем случае, на каждом шаге количество неравенств в системе возрастает квадратично, то есть метод имеет неполиномиальную сложность.

4.2 Применение метода граничных интервалов для решения задачи линейного программирования

4.2.1 Постановка задачи линейного программирования

Общая постановка задачи линейного программирования:

Введем обозначения $I = \{1, \dots, m\}$, $J = \{1, \dots, n\}$, $I_1, I_2 \subset I$, $J_1 \subset J$, $I_1 \cup I_2 = I$, $I_1 \cap I_2 = \emptyset$. Пусть заданы вещественные числа c_j, b_i, a_{ij} ($i \in I, j \in J$)

Требуется найти минимум по x функции

$$w(x) = \sum_{j=1}^n c_j x_j \quad (10)$$

при условиях

$$A_i x - b_i \geq 0, i \in I_1; \quad (11)$$

$$A_i x - b_i = 0, i \in I_2; \quad (12)$$

$$x_j \geq 0, j \in J_1. \quad (13)$$

Здесь $x = (x_1, \dots, x_n)$ - вектор переменных задачи.

4.2.2 Описание работы метода граничных интервалов для задачи линейного программирования

По сути, для решения задачи линейного программирования достаточно решить систему ограничений и найти на полученном решении минимум целевой функции.

Для того, чтобы решить задачу методом граничных интервалов, необходимо выполнить следующее:

1. Привести все ограничения к виду: $Ax \geq b$.

Это всегда возможно. Рассмотрим несколько простых методов, позволяющих осуществить такое преобразование:

- Любое ограничение-неравенство вида

$$\sum_{j=1}^n a_{ij} x_j \leq b_i \quad (14)$$

умножением на -1 приводится в неравенству

$$\sum_{j=1}^n a'_{ij} x_j \geq b'_i \quad (15)$$

- Любое ограничение-равенство

$$\sum_{j=1}^n a_{ij} x_j = b_i \quad (16)$$

можно представить в виде двух неравенств

$$\sum_{j=1}^n a_{ij} x_j \geq b_i, \quad \sum_{j=1}^n a_{ij} x_j \leq b_i \quad (17)$$

2. С помощью метода граничных интервалов сгенерировать матрицу граничных интервалов.
3. Из матрицы граничных интервалов выделить матрицу координат вершин. Подробно об алгоритме формирования матрицы можно посмотреть [1].
- 4.1 Если матрица координат не содержит бесконечности. Подставить координаты вершин в целевую функцию и выбрать минимальное значение.
- 4.2 Если матрица координат содержит бесконечность:
 - Необходимо найти расширенную матрицу граничных интервалов, сформированную для системы:

$$\begin{pmatrix} A^0 \\ 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \geq \begin{pmatrix} b^0 \\ \underline{x}_1 \\ \underline{x}_2 \\ -\bar{x}_1 \\ -\bar{x}_2 \end{pmatrix}, \quad (18)$$

где $A^0x \geq b^0$ - исходная система ограничений, а $[\underline{x}_1, \bar{x}_1] \times [\underline{x}_2, \bar{x}_2]$ брус обрезки.

В такой матрице уже отсутствует бесконечность по построению.

- Подставить координаты вершин из расширенной матрицы граничных интервалов в целевую функцию и выбрать минимальное значение.
- Если минимум нашелся в вершине, которая изначально содержалась в матрице вершин, то ответ найден.
- Если же минимальное значение оказалось в вершине, которая присутствует только в расширенной матрице вершин, то ответа нет.
- Если одно и то же минимальное значение нашлось и в случае исходной вершины и в случае "новой" вершины, которая присутствует только в расширенной матрице, тогда считаем, что ответ найден.

4.2.3 Симплекс-метод

Самый известный подход к решению задачи линейного программирования - это симплекс-метод. Он основан на переборе вершин выпуклого многогранника в пространстве заданной размерности.

Симплекс-метод строит базисные решения таким образом, что целевая функция, представленная линейным функционалом, монотонно убывает, пока не будет достигнуто необходимое условие.

Идея симплекс-метода состоит в следующем [2]:

1. Имея систему ограничений, приведенную к общему виду, то есть к системе m линейных уравнений с n переменными ($m < n$), находят любое базисное решение этой системы, заботясь только о том, чтобы найти его как можно проще.
2. Если первое же найденное базисное решение оказалось допустимым, то проверяют его на оптимальность. Если оно не оптимально, то осуществляется переход к другому, обязательно допустимому базисному решению.
3. Если первое найденное базисное решение окажется недопустимым, то с помощью симплекс-метода осуществляется переход к другим базисным решениям, пока на каком-то шаге решения либо базисное решение окажется допустимым и к нему применяют алгоритм симплексного метода, либо мы убеждаемся в противоречивости системы ограничений.

Сравнение метода граничных интервалов и симплекс-метода

Таблица 5: Достоинства и недостатки метода граничных интервалов и симплекс-метода при решении задачи ЛП

Критерий	Метод граничных интервалов	Симплекс-метод
Размерность	Ограничение размерности. Максимум \mathbb{R}^3	Любая размерность
Сложность реализации	Достаточная простая реализация	Сложная реализация
Дополнительные свойства	Возможность на одном и том же наборе ограничений рассматривать различные целевые функции	Решение для конкретной целевой функции

4.3 Функция `inequal` в MAPLE

В системе компьютерной математики MAPLE есть встроенная функция для решения системы линейных неравенств — `inequal`. В качестве аргументов задаются неравенства, пределы для переменных x и y и необязательные опции графической отрисовки.

Сравнение работы функции `inequal` и пакета `lineqs`

В данном разделе проведено сравнение функции `inequal` и пакета `lineqs` по следующим критериям: размерность, формат ввода, корректная отрисовка неограниченного множества решений, корректная отрисовка тощих множеств решений, доступность.

Ниже приведены примеры работ функции `inequal` и пакета `lineqs`.

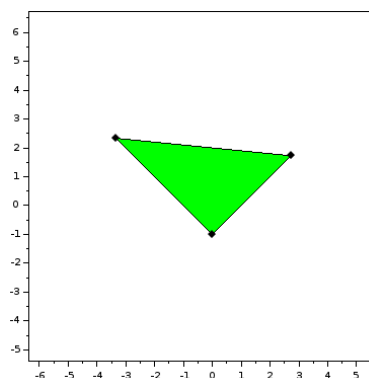
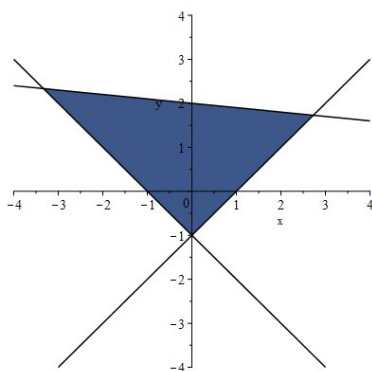
1. Ограниченное множество

Вызов функции `inequal`:

```
inequal ([x+y >= -1, x-y <= 1, y <= 2-.1*x],  
        x = -4..4, y = -4..4)
```

Вызов функции `SolSetIneq2D` из пакета `lineqs`:

```
A=[1 1; -1 1; -0.1 -1]  
b=[-1; -1; -2]  
[cnmtty,V]=SolSetIneq2D(A,b)
```



2. Неограниченное множество с одной вершиной

Вызов функции `inequal`:

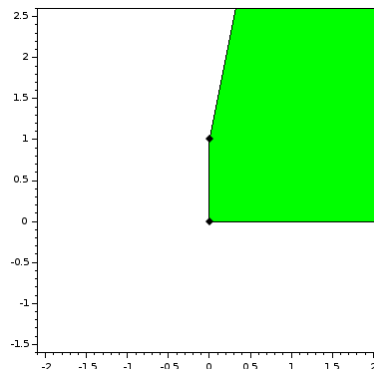
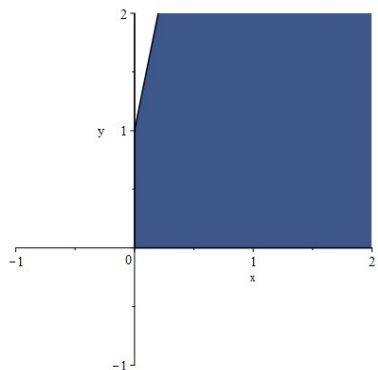
```
inequal ([x+y >= -1, 5*x-y >= -1, x >= 0, y >= 0],  
        x = -1 .. 2, y = -1 .. 2)
```

Вызов функции `SolSetIneq2D` из пакета `lineqs`:

```
A=[1 1; 5 -1; 1 0; 0 1]
```

```
b=[-1; -1; 0; 0]
```

```
[cnmty,V]=SolSetIneq2D(A,b)
```



3. Неограниченное множество

Вызов функции `inequal`:

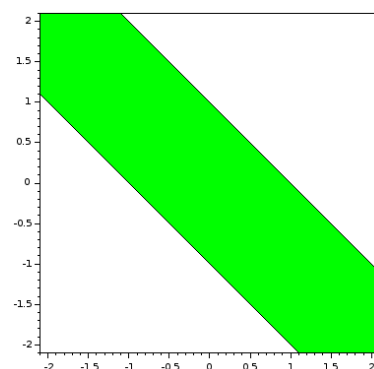
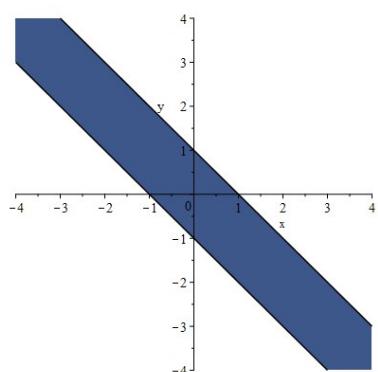
```
inequal([x+y <= 1, x+y >= -1, x = -1 .. 2, y = -1 .. 2])
```

Вызов функции `SolSetIneq2D` из пакета `lineqs`:

```
A=[ 1 1 ; -1 -1 ]
```

```
b=[ -1; -1]
```

```
[cnmty,V]=SolSetIneq2D(A,b)
```



4. Тощее множество

а) Вызов функции `inequal`:

```
inequal([12x+5y >= 1, 12x+5y <= 1, x >= -5, x <= 5],  
x = -4 .. 4, y = -4 .. 4)
```

Вызов функции `SolSetIneq2D` из пакета `lineqs`:

```
A=[12 5; -12 -5; 1 0; -1 0]
```

```
b=[1; -1; -5; -5]
```

```
[cnmty,V]=SolSetIneq2D(A,b)
```

б) Вызов функции `inequal`:

```
inequal([2*x+y>=1, 2*x+y<=1, x-4*y>=2, x-4*y<=2],
```

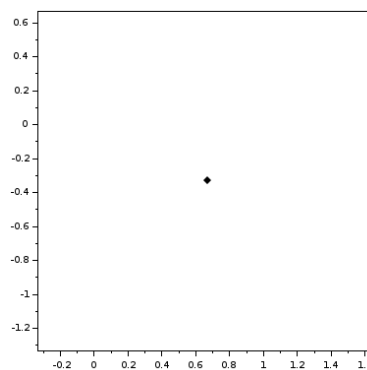
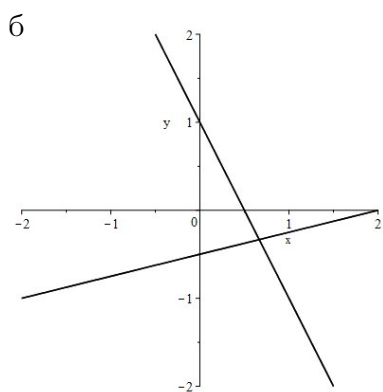
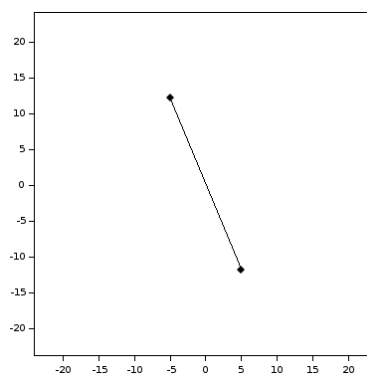
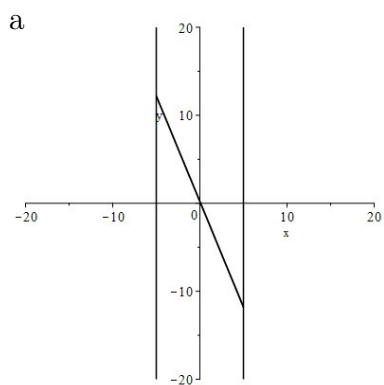
```
x = -2.. 2, y = -2 .. 2)
```

Вызов функции `SolSetIneq2D` из пакета `lineqs`:

```
A=[2 1; -2 -1; 1 -4; -1 4]
```

```
b=[1; -1; 2; -2]
```

```
[cnmty,V]=SolSetIneq2D(A,b)
```



Функция `inequal` в MAPLE работает неочевидным образом, например, все прямые, которые встречаются в качестве границ области решения отдельных неравенств отображаются на графике. Это приводит к неоднозначности при отрисовке тощих множеств решений (точка, отрезок). В примере выше для

случая, когда решением является одна точка, `inequal` выдает две прямые, в пересечении которых и есть искомое решение, но это не очевидно и для пользователя составляет некоторое неудобство выяснение всех нюансов.

При работе это может привести к некорректности отображения. Например, при решении данной системы:

$$\begin{cases} -x + y \geq -1 \\ -5x - y \geq -1 \\ x \geq 0 \\ y \geq 0 \end{cases} \quad (19)$$

на графике в работе `inequal` (рис. 4) отрисовывается часть прямой, которая задает граничное условие. Это вводит заблуждение: считается ли этот луч прямой частью решения? Возможно, при настройке параметров можно сделать так, чтобы линии границ не отображались, но для этого нужно хорошо разбираться со строением функции.

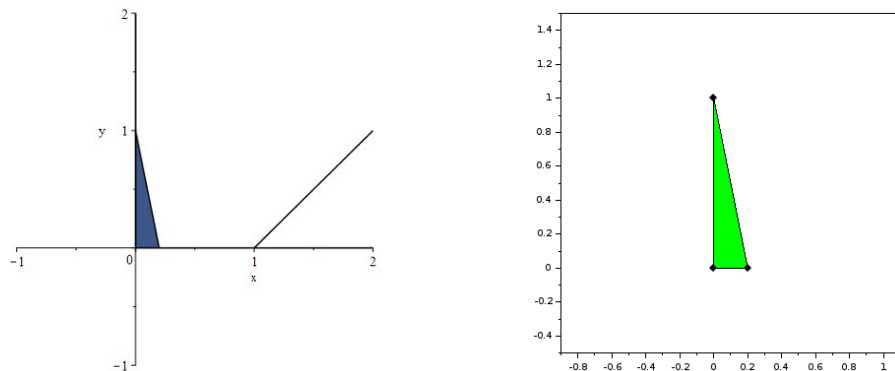


Рис. 4: Пример неоднозначной работы функции `inequal`

Таблица 6: Достоинства и недостатки пакета `lineqs` и функции `inequal`

Критерий	Пакет <code>lineqs</code>	Функция <code>inequal</code>
Размерность	\mathbb{R}^2 и \mathbb{R}^3	\mathbb{R}^2
Формат ввода	Необходимо привести все неравенства к виду: $Ax \geq b$	Любой вид неравенств
Окно отображения	Автоматическая подстройка границ отрисовки под решение	Необходимость задавать границы отрисовки
Обработка неограниченных множеств	Корректная	Корректная
Обработка тощих множеств	Корректная	Некорректная
Доступность	Сделан на базе свободно распространяемого программного обеспечения	Встроенная функция платной платформы

Заключение

В данной работе был рассмотрен метод граничных интервалов и описан подробный алгоритм его реализации для двумерного случая. Итогом является пакет программ `lineqs`, реализующий метод граничных интервалов для визуализации множеств решений систем линейных неравенств в свободной среде компьютерной математики Scilab.

Достоинства данного пакета:

- корректная работа с неограниченными множествами решений,
- корректная работа с тощими множествами решений,
- свободное распространение Scilab и самого пакета `lineqs`,
- простота использования.

Под простотой использования имеется ввиду следующее: чтобы решить задачу с помощью пакета `lineqs`, достаточно задать матрицу левой части системы линейных неравенств A и вектор правой части b , передать их в функцию, после чего вы получите визуализацию решения.

Также была рассмотрена возможность применения метода граничных интервалов для решения задачи линейного программирования в случае двумерной или трехмерной размерности пространства.

Проведено сравнение написанного пакета со встроенной функцией системы MAPLE `inequal`, которая должна решать системы линейных неравенств и равенств, и показана неоднозначность ее работы в общем случае, а также неправильная работа в случае тощих множеств.

Дальнейшим направлением работы будет реализация задачи визуализации множества решений интервальных линейных систем уравнений на основе написанного пакета `lineqs`.

Список литературы

- [1] Шарая И. А. Метод граничных интервалов для визуализации полиэдральных множеств решений // Вычислительные технологии, Том 20, № 1, 2015, страницы 75–102. Адрес доступа: <http://www.nsc.ru/interval/sharaya/Papers/Sharaya-JCT2015.pdf>
- [2] Попова Н. В. Математические методы // электронный учебник, 2005 ВТК. Адрес доступа: <http://matmetod-popova.narod.ru/>
- [3] Ларин Р. М., Плясунов А. В., Пяткин А. В. Методы оптимизации. Примеры и задачи. // учебное пособие, Новосибирск 2003 Адрес доступа: <http://www.math.nsc.ru/LBRT/k5/Plyasunov/met-opt-pr-zad.pdf>
- [4] Шарый С. П. Конечномерный интервальный анализ. // Новосибирск 2015 Адрес доступа: <http://www.nsc.ru/interval/Library/InteBooks/SharyBook.pdf>
- [5] Схрейвер А. Теория линейного и целочисленного программирования: том 1. // Москва, "Мир"1991, страницы 239–242.
- [6] Е. Р. Алексеев, О. В. Чеснокова, Е. А. Рудченко Scilab. Решение инженерных и математических задач // Москва, БИНОМ. Лаборатория знаний. 2008 Адрес доступа: <http://sch169.ru/projekt/Linux/lib/ScilabBook.pdf>
- [7] В.И. Зоркальцев, М.А. Киселева Системы линейных неравенств// – Иркутск: Изд-во Иркутского гос. ун-та, 2007 Адрес доступа: <http://math.isu.ru/ru/chairs/me/files/zorkaltsev/inequalities.pdf>