Unity. Precision. Perfection.

КОНСПЕКТ ЛЕКЦИЙ
по дисциплине «Методы оптимизации»

Лектор: Балтрашевич Владимир Эдуардович

Страниц: 62
Последнее обновление: 6 сентября 2019 г.

Автор: Корытов Павел, 6304

Санкт-Петербург
2019

Содержание

Замечания

Используемое ПО:

- XƎLaTeX, дистрибутив TeX Live 2018
- Microsoft OneNote
- Jupyter Notebook

Первые две лекции включены как pdf из OneNote.

(MAYBE) TODO: переписать с использованием TeX.

I. Безусловная оптимизация

20 февраля 2019 г. 14:09

Tags used:

Homework

Example

Definitions used

$\vec{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$ - вектор ($\boldsymbol{x}$ в методичке)

В примерах accent обычно опускается, т.к. надо быстро печатать

$x^{(0)}, x^{(1)}, \ldots$ — результаты итерации, если используются x_n как компоненты.

В противном случае, результаты итерации - $x_0, x_1, \ldots$

Безусловная оптимизация

Преподаватель - Балтрашевич В.Э.

$X = \mathbb{R}^n$

Если f — дифференцируема, то в точке минимума $\nabla f(x) = 0$

$$\nabla f(x) = \left(\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n}\right)$$

Релаксационные методы - методы построения последовательности $\{x_i\}, x_i \in X$:

- $f(x_0) > f(x_1) > \cdots$
- $x_i \to x^* = \arg\min\{f(x), x \in X\}$

Порядок метода - порядок старшей производной f, используемой при работе метода.

Общая схема

$$\boxed{\overrightarrow{x_{k+1}} = \overrightarrow{x_k} + t_k \overrightarrow{S_k}}$$

$\overrightarrow{S_k}$ — некоторый вектор, определяющий направление изменения $\overrightarrow{x_k}$

t_k — длина шага

Методы могут быть

- **Одношаговые** $\overrightarrow{S_k} = \varphi(x_k)$
- **Двухшаговые** $\overrightarrow{S_k} = \varphi(x_k, x_{k-1})$
- …

Сходимость релаксационного метода

$d = const$ — наибольшее из чисел, при котором выполняется условие

$$\boxed{\exists c: \forall k \left\|\overrightarrow{x_{k+1}} - \overrightarrow{x^*}\right\| \le c \cdot \|x_k - x^*\|^d}$$

Тогда d — скорость сходимости.

$$\Delta_k := \left\|\overrightarrow{x_k} - \overrightarrow{x^*}\right\|; \Delta_k \xrightarrow[k\to\infty]{} 0 \Rightarrow d = \overline{\lim_{k\to\infty}} \frac{\ln \Delta_{k+1}}{\ln \Delta_k}$$

$\Delta_{k+1} \approx q\Delta_k, 0 < q < q \Rightarrow d = 1 -$ **геометрическия / линейная** скорость
$\Delta'_{k+1} \approx const \cdot \Delta_k^2 \Rightarrow d = 2 -$ **квадратичная** скорость

Методы первого порядка (градиентные методы)

См. [ФОИТ :: Метод градиентного спуска]

Градиентный метод с постоянным шагом

t_k не зависит k, т.е. $t_k = const$
Тогда итерационная формула:

$$\boxed{\overrightarrow{x_{k+1}} = \overrightarrow{x_k} - t\nabla f(\overrightarrow{x_k})}$$

Теорема о сходимости метода с постоянным шагом

Если:

- $f -$ дифференцируема на $\mathbb{R}^n$
- $\nabla f(x)$ удовлетворяет **глобальному условию Липшица**
  $\exists \mathrm{L} > 0: \forall \vec{x}, \vec{y}: \|\nabla f(\vec{x}) - \nabla f(\vec{y})\| \le L \cdot \|\vec{x} - \vec{y}\|$
- $f -$ ограничена снизу, т.е.
  $\exists f^*, \text{т. ч.} \forall \vec{x}: f(\vec{x}) \ge f^* > -\infty$
- $0 < t < \dfrac{2}{L}$

Тогда для последовательности $\{\overrightarrow{x_k}\}$, получаемого по формуле $\vec{x}_{i+1} = \overrightarrow{x_k} - t\nabla f(x_k)$

- $\forall k$: $f(x_{k+1}) \le f(x_k) - t\left(1 - t\frac{L}{2}\right) \cdot \|\nabla f(\overrightarrow{x_k})\|^2$
  $\{f(\overrightarrow{x_k})\}$ монотонно убывает: $f(\overrightarrow{x_{k+1}}) < f(\overrightarrow{x_k})$ при $\nabla f(\overrightarrow{x_k}) \ne 0$
- $\nabla f(\overrightarrow{x_k}) \xrightarrow[k\to\infty]{} 0$

Доказательство

$$\frac{df(\vec{x} + \tau\vec{y})}{d\tau} = (\nabla f(\vec{x} + \tau\vec{y}), y) \Rightarrow$$

$$f(\vec{x} + \vec{y}) = f(\vec{x}) + (\nabla f(\vec{x}), y) + \int_0^1 (\nabla f(\vec{x} + \tau\vec{y}) - \nabla f(\vec{x}), y)\, d\tau$$

Пусть $\vec{x} = \overrightarrow{x_k}; y = -t\nabla f(\overrightarrow{x_k})$
Тогда

$$f(\overrightarrow{x_{k+1}}) = f(\overrightarrow{x_k}) - t \cdot \|\nabla f(x_k)\|^2 + \int_0^1 (\nabla f(\overrightarrow{x_k} + \tau t\nabla f(\overrightarrow{x_k})) - \nabla f(\vec{x}_k), -t\nabla f(\overrightarrow{x_k}))d\tau$$

С учётом $(x, y) \le \|x\| \cdot\ \|y\|$ и неравенства Липшица:

$$\left(\nabla f\left(\overrightarrow{x_k} - t\tau\nabla f(\overrightarrow{x_k})\right) - \nabla f(\overrightarrow{x_k}), -t\nabla f(\overrightarrow{x_k})\right) \le$$
$$\le \left\|\nabla f\left(\overrightarrow{x_k} - t\tau\nabla f(\overrightarrow{x_k})\right) - \nabla f(\overrightarrow{x_k})\right\| \cdot \|-t\nabla f(\overrightarrow{x_k})\| \le$$
$$\le L\|-t\tau\nabla f(\overrightarrow{x_k})\| \cdot \|-t\nabla f(\overrightarrow{x_k})\| = Lt^2\tau\|\nabla f(\overrightarrow{x_k})\|^2$$

Тогда

$$f(\overrightarrow{x_{k+1}}) \le f(\overrightarrow{x_k}) - t\|\nabla f(\overrightarrow{x_k})\|^2 + Lt^2\|\nabla f(\overrightarrow{x_k})\|^2 \int_0^1 \tau d\tau = f(\overrightarrow{x_k}) - \tau\left(1 - \frac{Lt}{2}\right)\|\nabla f(\overrightarrow{x_k})\|^2$$

$a := t\left(1 - \frac{Lt}{2}\right); 0 < \mathrm{t} < \frac{2}{\mathrm{L}} \Rightarrow \mathrm{a} > 0 \Rightarrow \forall k: f(\overrightarrow{x_{k+1}}) \le f(\overrightarrow{x_k}) - a\|\nabla f(\overrightarrow{x_k})\|^2$
Сложим полученные неравенства при $k = 0,1, \dots, s$:

$$f(\overrightarrow{x_{s+1}}) \le f(\overrightarrow{x_0}) - a\sum_{k=0}^{s}\|\nabla f(\overrightarrow{x_k})\|^2$$

$$a > 0 \Rightarrow \sum_{k=0}^{s}\|\nabla f(\overrightarrow{x_k})\|^2 \le a^{-1}\big(f(\overrightarrow{x_0}) - f(\overrightarrow{x_{s+1}})\big)$$

$$f(\vec{x}) \ge f^* > -\infty \Rightarrow \forall s: \sum_{k=0}^{s}\|\nabla f(\overrightarrow{x_k})\|^2 \le a^{-1}(f(\overrightarrow{x_0}) - f^*) \Rightarrow$$

$$\Rightarrow \sum_{k=0}^{s}\|\nabla f(\overrightarrow{x_k})\|^2 < \infty;\ \|\nabla f(\overrightarrow{x_k})\| \underset{k\to\infty}{\longrightarrow} 0 \blacksquare$$

<u>Покоординатный спуск</u>

На каждом этапе выполняется шаг только по одной координате

<u>Пример [ДЗ 27.02.2019]</u>

$f(x_1, x_2) = x_1^2 + (x_2 - 1)^2$

Пусть метод градиентного спуска начинается из точки $(0{,}0)$

Первое направление - $\begin{pmatrix}1\\0\end{pmatrix}$, второе - $\begin{pmatrix}0\\1\end{pmatrix}$

$x^{(0)} = \begin{pmatrix}0\\0\end{pmatrix}$

$x^{(1)} = \begin{pmatrix}0\\0\end{pmatrix} - \gamma\begin{pmatrix}1\\0\end{pmatrix} = \begin{pmatrix}-\gamma\\0\end{pmatrix}$

$f\big(x^{(1)}\big) = \gamma^2$

$f' = 2\gamma$

$f' = 0 \Rightarrow \gamma = 0 \Rightarrow x^{(1)} = \begin{pmatrix}0\\0\end{pmatrix}$

$x^{(2)} = \begin{pmatrix}0\\0\end{pmatrix} - \gamma\begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}0\\-\gamma\end{pmatrix}$

$f\big(x^{(2)}\big) = (-\gamma - 1)^2 = \gamma^2 + 2\gamma + 1 = \psi(\gamma)$

$f' = 2\gamma + 2$

$f' = 0 \Rightarrow \gamma = -1 \Rightarrow x^{(2)} = \begin{pmatrix}1\\0\end{pmatrix}$

<u>Выпуклые функции и множества</u>

$X \subseteq \mathbb{R}^n -$ **выпуклое**, если $\forall \overrightarrow{x_1}, \overrightarrow{x_2} \in X, \lambda \in [0,1]:\ \lambda\vec{x}_1 + (1-\lambda)\overrightarrow{x_2} \in X$

$\vec{z} -$ **выпуклая комбинация** $\overrightarrow{x_1}, \overrightarrow{x_2}, \dots, \overrightarrow{x_m}$, если $\vec{z} = \sum_{i=1}^{m} a_i\overrightarrow{x_i}, a_i \ge 0, \sum a_i = 1$

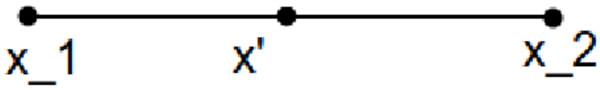
Теорема о выпуклом множестве

Выпуклое множество X содержит все выпуклые комбинации своих точек

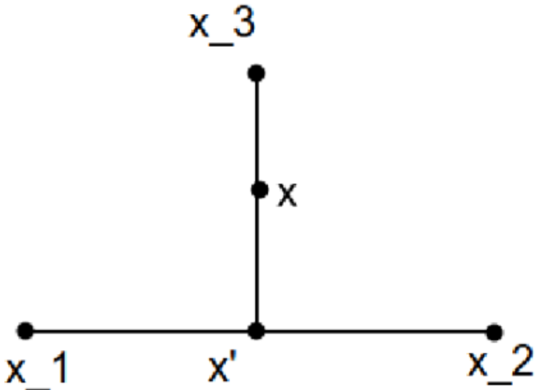
План доказательства

Доказательство методом индукции.

Для двух точек очевидно



$\forall x\ \exists\lambda$: $x_1 \cdot \lambda + x_2 \cdot (1-\lambda) = x$
Добавим ещё одну точку:



$x' = \lambda_1 x_1 + (1-\lambda_1)x_2$
$x = \lambda_2 x' + (1-\lambda_2)x_3 = \lambda_2\lambda_1 x_1 + \lambda_2(1-\lambda_1)x_2 + (1-\lambda_2)x_3$
$\alpha_1 = \lambda_1\lambda_2$
$\alpha_2 = \lambda_2 - \lambda_1\lambda_2$
$\alpha_3 = 1 - \lambda_2$

$$\sum \alpha_i = \lambda_1\lambda_2 + \lambda_2 - \lambda_1\lambda_2 + 1 - \lambda_2 = 1$$

Далее по индукции

Теорема. Пересечение выпуклых множеств выпукло.
Доказательство
Пусть X, Y — выпуклые, т.е.
$Z \coloneqq X \cap Y$ — пересечение
По определению $Z \subseteq X, Z \subseteq Y$. По теореме о выпуклом множестве Z — выпукло ■

Функция f называется **выпуклой**, если:
- $D(f)$ — область определения - выпукла
- $\forall \overrightarrow{x_1}, \overrightarrow{x_2} \in D(f)$: $f(\lambda\overrightarrow{x_1} + (1-\lambda)\overrightarrow{x_2}) \le \lambda f(\overrightarrow{x_1}) + (1-\lambda)f(\overrightarrow{x_2}), \lambda \in (0,1)$

Функция f называется **вогнутой**, если $-f$ выпукла.

Функция f называется **строго выпуклой**, если
$\forall \vec{x} \ne y, 0 < \lambda < 1$: $f(\lambda\vec{x} + (1-\lambda)\vec{y}) < \lambda f(x) + (1-\lambda)f(y)$

Функция f называется **сильновыпуклой** с константой $l > 0$, если
$\forall \lambda \in [0,1]$: $f(\lambda\vec{x} + (1-\lambda)\vec{y}) \le \lambda f(x) + (1-\lambda)f(y) - l\lambda(1-\lambda)\|\vec{x} - \vec{y}\|$

Свойства выпуклых функций
1. Любая точка локального минимума выпуклой функции является точкой её глобального минимума
   Доказательство

Пусть $\vec{x}^*$ — локальный, но не глобальный минимум. Значит
$\exists \vec{y} \in X: f(\vec{y}) < f(\vec{x}^*)$
Пусть $\vec{x} = \lambda\vec{y} + (1-\lambda)\vec{x}^*, \lambda \in (0,1)$
Т.к. X — выпукло, $\vec{x} \in$ X.
Т.к. f — выпукла, $f(\vec{x}) = f(\lambda\vec{y} + (1-\lambda)\vec{x}^*) \leq \lambda f(y) + (1-\lambda) f(\vec{x}^*) + (1-\lambda) f(\vec{x}^*) = f(\vec{x}^*)$
Выходит, что $\forall \lambda \in (0,1): f(\vec{x}) < f(\vec{x}^*)$. Значит, $\vec{x}^*$ — не локальный минимум. ■

2. **Критерий выпуклости дифференцируемых функций**
   Пусть $\exists f''$ — непрерывна.
   f — выпукла $\Leftrightarrow f'' > 0$
   f — сильновыпукла $\Leftrightarrow \exists l, \forall x: \nabla^2 f(\vec{x}) \geq l\vec{I}$
3. **Неравенство Йенсена**
   f — выпукла
   $$\Leftrightarrow \forall m \geq 2, \overrightarrow{x_1}, \overrightarrow{x_2}, \dots, \overrightarrow{x_m} \in X, \lambda_i \geq 0, \sum_{i=1}^{m} \lambda_i = 1:$$
   $$f\left(\sum_{i=1}^{m} \lambda_i \overrightarrow{x_i}\right) \leq \sum_{i=1}^{m} \lambda_i f(\overrightarrow{x_i})$$
   [док-во см. 9-10 мет. опт.]

4. f — выпукла, $X = D(f)$ — выпукло $\Rightarrow f$ — непрерывна в каждой внутренней точке X и имеет в ней производную по любому направлению
5. f — дифференцируемая функция на выпуклом множестве X.
   > Выпуклость эквивалентна
   $\forall \vec{x}, \vec{y} \in X: f(\vec{x} + \vec{y}) \geq f(\vec{x}) + (\nabla f(\vec{x}), \vec{y})$
   > Строгая выпуклость эквивалентна
   $\forall \vec{y} \neq 0, \vec{x} \in X: f(\vec{x} + \vec{y}) \geq f(\vec{x}) + (\nabla f(\vec{x}), \vec{y})$
   > Сильная выпуклость эквивалентна
   $$\forall \vec{x}, \vec{y} \in X, l > 0: f(\vec{x} + \vec{y}) \geq f(\vec{x}) + (\nabla f(\vec{x}), \vec{y}) + \frac{l\|\vec{y}\|^2}{2}$$
6. Для сильновыпуклых функций:
   > $f(\vec{x}) \geq f(\vec{x}^*) + \dfrac{l\|\vec{x} - \vec{x}^*\|^2}{2}$
   > $(\nabla f(\vec{x}), x - \vec{x}^*) \geq l\|\vec{x} - \vec{x}^*\|^2$
   > $\|\nabla f(\vec{x})\| \geq l\|\vec{x} - \vec{x}^*\|$

Теорема. Пусть $\exists f''$. Если для $l, L > 0\ \forall x$ верно:
$l\vec{I} \leq \nabla^2 f(x) \leq L\vec{I}$
То $\left\|\overrightarrow{x_k} - \overrightarrow{x^*}\right\| \leq \|x_0 - x^*\| \cdot q^k$
$q = \max\{|1-\gamma l|, |1-\gamma L|\}$
$$\min q = \frac{L-l}{L+l} < 1$$
[доказательство см. 11 мет. опт.]

Теорема. Если $\varphi(x)$ — выпукла на X, то $f(x) = \max\{\varphi(x), 0\}$ — тоже выпукла [ДЗ

Доказательство.
По определению выпуклой функции
$\lambda \in (0,1), \forall \overrightarrow{x_1}, \overrightarrow{x_2} \in X$:
$\varphi(\lambda\overrightarrow{x_1} + (1-\lambda)\overrightarrow{x_2})) \leq \lambda\varphi(\overrightarrow{x_1}) + (1-\lambda)\varphi(\overrightarrow{x_2})$

Докажем, что: $\max\{a+b, 0\} \leq \max\{a, 0\} + \max\{b, 0\}$

- $a \geq 0, b \geq 0 \Rightarrow \max\{a+b, 0\} = a + b = \max\{a, 0\} + \max\{b, 0\}$
- $a < 0, b \geq 0, a + b > 0 \Rightarrow a + b < b \Rightarrow$
  $\max\{a+b, 0\} = a + b < b = \max\{a, 0\} + \max\{b, 0\}$
- $a < 0, b \geq 0, a + b < 0 \Rightarrow$
  $\max\{a+b, 0\} = 0 < b = \max\{a, 0\} + \max\{b, 0\}$

Очевидно: $\max\{ac, 0\} = c \cdot \max\{a, 0\}$, если $c > 0$
Очевидно: $max\{a, 0\} \leq ma\,x\{b, 0\}$, если $a \leq b$

$f(\lambda\overrightarrow{x_1} + (1-\lambda)\overrightarrow{x_2}) = \max\{\varphi(\lambda\overrightarrow{x_1} + (1-\lambda)\overrightarrow{x_2}), 0\} \leq \max\{\lambda\varphi(\overrightarrow{x_1}) + (1-\lambda)\varphi(\overrightarrow{x_2}), 0\} \leq$
$\leq \max\{\lambda\varphi(\overrightarrow{x_1}), 0\} + \max\{(1-\lambda)\varphi(\overrightarrow{x_2}), 0\} = \lambda \max\{\varphi(\overrightarrow{x_1}, 0\} + (1-\lambda)\max\{\varphi(\overrightarrow{x_2}), 0\} =$
$= \lambda f(\overrightarrow{x_1}) + (1-\lambda) f(\overrightarrow{x_2})$ ■

<u>Метод с дроблением шага, метод наискорейшего спуска</u>
См. ФОИТ :: Метод градиентного спуска

<u>Пример метода наискорейшего спуска</u>
$f(x_1, x_2) = x_1^2 + (x_2 - 1)^2$

$$\nabla f(x) = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{pmatrix} = \begin{pmatrix} 2x_1 \\ 2x_2 - 2 \end{pmatrix}$$

$x^{(0)} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

$x^{(1)} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} - \gamma \begin{pmatrix} 0 \\ -2 \end{pmatrix} = \begin{pmatrix} 0 \\ 2\gamma \end{pmatrix}$

$f\Big|_{x^{(1)}} = (2\gamma - 1)^2 = 4\gamma^2 - 4\gamma + 1$

$f' = 8\gamma - 4 = 0 \Rightarrow \gamma = \frac{1}{2} \Rightarrow x^{(1)} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

Чтобы проверить, что это точка минимума, нужно найти вторую производную

$$\nabla^2 f(x_1, x_2) = \begin{pmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{pmatrix}$$

$\nabla^2 f\Big|_{x^{(1)}} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$

Функция положительно определена, значит $x^{(1)}$ - точка минимума

Масштабирование
Заключается в замене переменных вида
$x_i = \mu_i y_i$
Это позволяет уменьшить вытянутость функции вдоль некоторых осей.

Пример. ДЗ 06.03.2019
$f(x) \coloneqq x_1^2 + 16x_2^2$
Найти минимум с масштабированием и без масштабирования.
$x^{(0)} = \begin{pmatrix} 5 \\ 5 \end{pmatrix}$

Без масштабирования
$$\nabla f(x) = \begin{pmatrix} \dfrac{\partial f}{\partial x_1} \\ \dfrac{\partial f}{\partial x_2} \end{pmatrix} = \begin{pmatrix} 2x_1 \\ 32x_2 \end{pmatrix}$$
$$x^{(0)} = \begin{pmatrix} 5 \\ 5 \end{pmatrix}$$
$$x^{(1)} = \begin{pmatrix} 5 \\ 5 \end{pmatrix} - \gamma \begin{pmatrix} 10 \\ 160 \end{pmatrix} = \begin{pmatrix} 5 - 10\gamma \\ 5 - 160\gamma \end{pmatrix}$$
$$f\Big|_{x^{(1)}} = (5 - 10\gamma)^2 + 16(5 - 160\gamma)^2$$
$$= 25 - 100\gamma + 100\gamma^2 + 400 - 25600\gamma + 409600\gamma^2$$
$$= 409700\gamma^2 - 25700\gamma + 425$$
$$f'\Big|_{x^{(1)}} = 819400\gamma - 25700$$
$$f'\Big|_{x^{(1)}} = 0 \Rightarrow 891400\gamma = 25700 \Rightarrow \gamma = \frac{257}{8194}$$
$$x^{(1)} = \begin{pmatrix} \dfrac{5 \cdot 8194 - 10 \cdot 257}{8194} \\ \dfrac{5 \cdot 8194 - 160 \cdot 257}{8194} \end{pmatrix} = \begin{pmatrix} \dfrac{38400}{8194} \\ \dfrac{-150}{8194} \end{pmatrix} = \begin{pmatrix} \dfrac{19200}{4097} \\ \dfrac{-75}{4097} \end{pmatrix}$$
$$x^{(2)} = x^{(1)} - \gamma \nabla f\left(x^{(1)}\right) = \begin{pmatrix} x_1^{(1)} \\ x_2^{(1)} \end{pmatrix} - \gamma \begin{pmatrix} 2x_1^{(1)} \\ 32x_2^{(1)} \end{pmatrix} = \begin{pmatrix} x_1^{(1)} - 2\gamma x_1^{(1)} \\ x_2^{(1)} - 32\gamma x_2^{(1)} \end{pmatrix} = \begin{pmatrix} x_1^{(1)}(1 - 2\gamma) \\ x_2^{(1)}(1 - 32\gamma) \end{pmatrix}$$
$$= \begin{pmatrix} \dfrac{38400}{8194}\left(1 - 2 \cdot \dfrac{257}{8194} \cdot \gamma\right) \\ -\dfrac{150}{8194}\left(1 - 32 \cdot \dfrac{257}{8194} \cdot \gamma\right) \end{pmatrix} = \begin{pmatrix} 38400\,\dfrac{8194 - 514\gamma}{8194} \\ -150\,\dfrac{8194 - 16864\gamma}{8194} \end{pmatrix}$$
$$= \begin{pmatrix} \dfrac{314649600 - 19737600\gamma}{8194} \\ \dfrac{-1229100 + 2529600\gamma}{8194} \end{pmatrix}$$
$$f\Big|_{x^{(2)}} = \left(\frac{314649600 - 19737600\gamma}{8194}\right)^2 + 16\left(\frac{1229100 - 2529600\gamma}{8194}\right)^2$$
Не получается

С масштабированием

$f(x) = x_1^2 + 16x_2^2$

$$\mu_1 = \frac{1}{\sqrt{\frac{\partial^2 f}{\partial x_1^2}(x^{(0)})}} = \frac{1}{\sqrt{2}}$$

$$\mu_2 = \frac{1}{\sqrt{\frac{\partial^2 f}{\partial x_2^2}(x^{(0)})}} = \frac{1}{\sqrt{32}} = \frac{1}{4\sqrt{2}}$$

$$f(y) = \frac{y_1^2}{2} + \frac{y_2^2}{2}$$

$$\nabla f(y) = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$$

$$x^{(0)} = \begin{pmatrix} \frac{5}{\sqrt{2}} \\ \frac{5}{4\sqrt{2}} \end{pmatrix}$$

$$x^{(1)} = \begin{pmatrix} \frac{5}{\sqrt{2}} \\ \frac{5}{4\sqrt{2}} \end{pmatrix} - \gamma \begin{pmatrix} \frac{5}{\sqrt{2}} \\ \frac{5}{4\sqrt{2}} \end{pmatrix} = \begin{pmatrix} \frac{5}{\sqrt{2}}(1-\gamma) \\ \frac{5}{4\sqrt{2}}(1-\gamma) \end{pmatrix}$$

$$f\Big|_{x^{(1)}} = \frac{\left(\frac{5}{\sqrt{2}}(1-\gamma)\right)^2}{2} + \frac{\left(\frac{5}{4\sqrt{2}}(1-\gamma)\right)^2}{2} = \frac{(1-\gamma)^2}{2}\left(\frac{25}{2} + \frac{25}{32}\right)$$

$$= \frac{425}{64} - \frac{425}{32}\gamma + \frac{425}{64}\gamma^2$$

$$f'\Big|_{x^{(1)}} = \frac{425}{32}\gamma - \frac{425}{32} = 0 \Rightarrow \gamma = 1$$

$$x^{(1)} = \begin{pmatrix} \frac{5}{\sqrt{2}}(1-1) \\ \frac{5}{4\sqrt{2}}(1-1) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

<u>Метод Ньютона</u>

Ряд Тейлора:

$$f(\vec{x}) = f(\vec{x}_k) + (\nabla f(\vec{x}_k), \vec{x} - \vec{x}_k) + \frac{1}{2}(\nabla^2 f(\vec{x}_k)(\vec{x} - \vec{x}_k), \vec{x} - \vec{x}_k) + o(\|\vec{x} - \vec{x}_k\|)$$

Пусть $f_2(\vec{x})$ – квадратичная аппроксимация $f(\vec{x})$

$f_2(\vec{x})$ имеет единственную точку минимума, которая является корнем $\nabla f_2(\vec{x}) = 0$

В данном случае

$$\nabla f_2(\vec{x}) = 0 = \nabla f(\vec{x}_k) + \nabla^2 f(\vec{x}_k)(\vec{x} - \vec{x}_k)$$

$$\vec{x}_{k+1} = \vec{x}_k - [\nabla^2 f(\vec{x}_k)]^{-1}\nabla f(\vec{x}_k)$$

Пример

$$f(\vec{x}) = \frac{1}{2}(A\vec{x},\vec{x}) - \left(\vec{b},\vec{x}\right)$$
$$\nabla f(\vec{x}) = A\vec{x} - \vec{b}$$
$$\nabla f(\vec{x}) = 0 \Rightarrow \vec{x} = A^{-1}\vec{b}$$
$$\nabla^2 f(\vec{x}_k) = A$$
$$\overrightarrow{x_1} = \overrightarrow{x_0} - A^{-1}\left(A\overrightarrow{x_0} - \vec{b}\right) = A^{-1}b$$

<u>Достоинства и недостатки</u>

Градиентный метод	Метод Ньютона
✓ Слабые требования к исходным данным. x_0 может быть далека от x^* ✓ Используется только градиент функции f ✓ Относительная простота вычислений – Медленная скорость сходимости	✓ Быстрая скорость сходимости (квадратичная) – Только локальная сходимость. Начальное приближение должно быть достаточно близким – Жесткие требования к функции (должна быть дважды непрерывно дифференцируема) – Большой объем вычислений, связанный с необходимостью вычисления матрицы вторых производных и её обращений

<u>Метод Пауэлла</u>

- Выбираются две точки и направление
- Через две точки рисуются векторы
- На векторах ищется минимальное значение
- Через точки, в которых на векторе минимальное значение рисуется следующий вектор

Пример

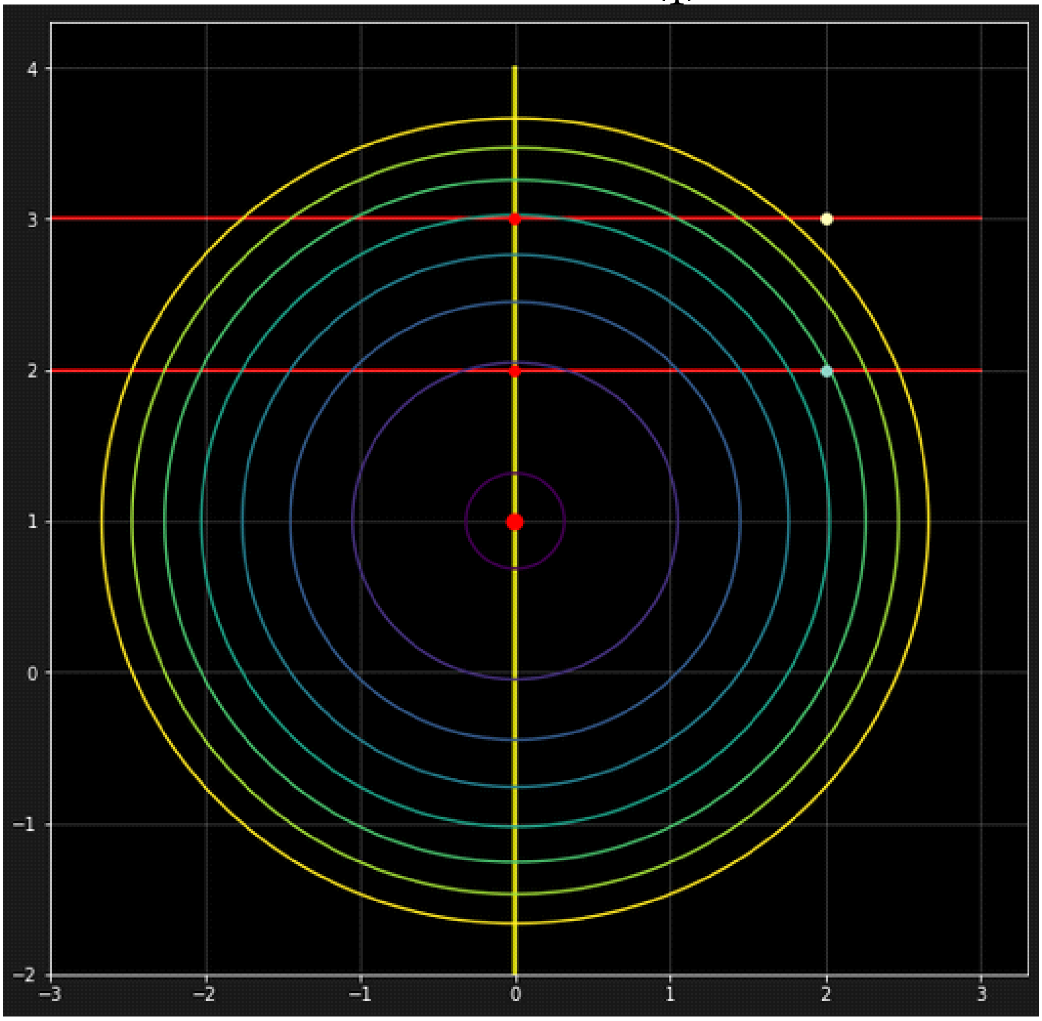
$$f(x_1,x_2) = x_1^2 + (x_2-1)^2$$
$x^{(-1)} = \begin{pmatrix}2\\2\end{pmatrix}; x^{(0)} = \begin{pmatrix}2\\3\end{pmatrix}$; Направление - $\begin{pmatrix}1\\0\end{pmatrix}$
$$x^{(1)} = \begin{pmatrix}2\\2\end{pmatrix} - \gamma\begin{pmatrix}1\\0\end{pmatrix} = \begin{pmatrix}2-\gamma\\2\end{pmatrix}$$
$$f\left(x^{(1)}\right) = (2-\gamma)^2 + 1$$
$$f'(x^{(1)}) = 2\gamma - 4 = 0 \Rightarrow \gamma = 2$$
$$x^{(1)} = \begin{pmatrix}0\\2\end{pmatrix}$$
$$x^{(2)} = \begin{pmatrix}2\\3\end{pmatrix} - \gamma\begin{pmatrix}1\\0\end{pmatrix} = \begin{pmatrix}2-\gamma\\3\end{pmatrix}$$
$$f\left(x^{(2)}\right) = (2-\gamma)^2 + 4$$
$$f'\left(x^{(2)}\right) = 2\gamma + 4 = 0 \Rightarrow \gamma = 2$$
$$x^{(2)} = \begin{pmatrix}0\\3\end{pmatrix}$$

Новое направление - $x^{(2)} - x^{(1)} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

$x^{(3)} = \begin{pmatrix} 0 \\ 2 \end{pmatrix} - \gamma \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 - \gamma \end{pmatrix}$

$f\left(x^{(3)}\right) = (1 - \gamma)^2$

$f'\left(x^{(3)}\right) = 2\gamma - 2 = 0 \Rightarrow \gamma = 1 \Rightarrow x^{(3)} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$



Многошаговые методы

Метод тяжелого шарика

Заключается в учёте "инерции движения"

$\vec{x}_{k+1} = \overrightarrow{x_k} - \alpha \nabla f(\overrightarrow{x_k}) + \beta(\overrightarrow{x_k} - \overrightarrow{x_{k-1}})$

Теорема о скорости сходимости метода тяжелого шарика

Если:

- $0 < l \leq \nabla^2 f(\vec{x}) \leq L$ (сильная выпуклость функции)
- $0 \leq \beta \leq 1; 0 \leq \alpha \leq \dfrac{2(1+\beta)}{L}$

То:

- $\exists c, q, \forall k: \|\vec{x}_k - \vec{x}^*\| \leq cq^k$
- $q_{min} = \dfrac{\sqrt{L} - \sqrt{l}}{\sqrt{L} + \sqrt{l}}$

Получается, что этот метод сходится не быстрее геометрической прогрессии. Поэтому при плохой обусловленности предпочтительно применение одношагового градиентного метода.

Метод сопряженных градиентов

$\vec{x}_{k+1} = \vec{x}_k - \alpha_k \nabla f(\vec{x}_k) + \beta(\vec{x}_k - \vec{x}_{k-1})$

В отличие от метода тяжелого шарика, α, β вычисляются следующим образом:

$(\alpha_k, \beta_k) = \arg\min\left\{f\left(\vec{x}_k - \alpha \nabla f(\vec{x}_k) + \beta(\vec{x}_k - \vec{x}_{k-1})\right) : \alpha > 0, \beta > 0\right\}$

Для квадратичной функции $\frac{1}{2}(A\vec{x}, \vec{x}) - \left(\vec{b}, \vec{x}\right), A > 0$

- Метод сходится за конечное число шагов, не превосходящее размерности пространства состояний
- Градиенты в точках $\vec{x}_k$ попарно ортогональны
  $\forall i \neq k: \left(\nabla f(\vec{x}_i), \nabla f(\vec{x}_k)\right) = 0$
- $\vec{p}_k \coloneqq \vec{x}_k - \vec{x}_{k-1}$
  $\forall i \neq j: \left(A\vec{p}_i, \vec{p}_j\right) = 0$

Для некоторой положительно определенной матрица A векторы $\vec{p}_i$, связанные соотношением $\left(\mathrm{A}\vec{p}_i, \vec{p}_j\right) = 0$, называются **сопряженными** или $\boldsymbol{A}$ **− ортогональными**

Модификация Полака-Ривьера

$\vec{x}_{k+1} = \vec{x}_k + \alpha_k \vec{p}_k$

$\alpha_k = \arg\min f(\vec{x}_k + \alpha \vec{p}_k)\,, \alpha > 0$

$\vec{p}_k = -\nabla f(\vec{x}_k) + \beta \vec{p}_{k-1}$

$$\beta_k = \frac{\left(\left(\nabla f(\vec{x}) - \nabla f(\vec{x}_{k-1})\right), \nabla f(\vec{x}_k)\right)}{\|\nabla f(\vec{x}_{k-1})\|}, \beta_0 = 0$$

Квазиньютоновские методы

Итерационная схема метода имеет вид:

$\vec{x}_{k+1} = \vec{x}_k - \gamma_k H_k \nabla f(\vec{x}_k)$

Если $H_k = 1$, это градиентный метод

Если $H_k = \left(\nabla^2 f(\vec{x}_k)\right)^{-1}, \gamma_k = 1$ − это метод Ньютона

Достоинство такого подхода в том, что не нужно вычислять обратную матрицу вторых производных.

Обозначим:

$p_k \coloneqq -H \nabla f(\vec{x}_k)$

$y_k \coloneqq \nabla f(\vec{x}_{k-1}) - \nabla f(\vec{x}_k)$

$$f(\vec{x}) \coloneqq \frac{A\vec{x}, \vec{x}}{2} + \left(\vec{b}, \vec{x}\right), A > 0$$

Тогда для

$\vec{y}_k = A(\vec{x}_{k+1} - \vec{x}_k) = \gamma_k A \vec{p}_k, \gamma_k \vec{p}_k = A^{-1} \vec{\gamma}_k$

Квазиньютоновское условие: $H_{k+1} \vec{y}_k = \gamma_k \vec{p}_k$

Метод Давидона-Флетчера-Пауэлла

$$H_{k+1} = H_k - \frac{H_k \vec{y}_k \left(H_k \, \vec{y}_j\right)^T}{(H_k \vec{y}_k, \vec{y}_k)} + \gamma_k \frac{\vec{p}_k \vec{p}_k^T}{(\vec{p}_k, \vec{y}_k)}; H_0 > 0$$

На каждом шаге, имея H_k, делается шаг в направлении $\vec{p}_k$. γ_k можно получить, например, по методу наискорейшего спуска.
После чего получается x_{k+1}, вычисляется y_k и пересчитывается H_{k+1}

Метод Бройдена-Флетчера-Шенно

$$H_{k+1} = H_k - \frac{\rho_k \vec{p}_k (\vec{p}_k)^T - \vec{p}_k (\vec{y}_k)^T H_k - H_k \vec{y}_k (\vec{p}_k)}{(\vec{y}_k, \vec{p}_k)}$$

$$\rho_k = \gamma_k + \frac{(H_k \vec{y}_k, \vec{y}_k)}{(\vec{y}_k, \vec{p}_k)}$$

Методы нулевого порядка

Методы аппроксимации

Пусть e_j — орт оси j

$$\varphi\left(\vec{x} + \gamma \vec{e}_j\right) \approx f(\vec{x}) + \frac{\partial f}{\partial \vec{x}} \gamma + o(\gamma)$$

$$\frac{\partial f}{\partial \vec{x}_j} \approx \frac{f\left(\vec{x} + \gamma \vec{e}_j\right) - f(\vec{x})}{\gamma} \approx \frac{f\left(\vec{x} + \gamma \vec{e}_j\right) - f\left(\vec{x} - \gamma \vec{e}_j\right)}{2\gamma}$$

Метод покоординатного спуска

Метод симплексов

Задача - найти экстремум функции

$c_1 x_1 + c_2 x_2 + \dots + c_n x_n \to min$

При ограничениях:

$a_{11} x_1 + a_{12} x_2 + \dots + a_{1n} x_n \le b_1$

$a_{21} x_1 + a_{22} x_2 + \dots + a_{2n} x_n \le b_2$

…

$a_{m1} x_1 + a_{m2} x_2 + \dots + a_{mn} x_n \le b_m$

Симплекс-таблица

	x_1	x_2	…	x_n	
y_1	a_{11}	a_{12}	…	a_{1n}	b_1
y_2	a_{21}	a_{22}	…	a_{2n}	b_2
…	…	…	…	…	…
y_m	a_{m1}	a_{m2}	…	a_{mn}	b_m
	c_1	c_2	…	c_n	0

Алгоритм

1. Выбрать разрешающий элемент
   Для поиска крайней точки:

- Разрешающая строка j — любой, где $b_j < 0$
- Разрешающий столбец i — такой, чтобы $a_{ij} > 0$, и $\frac{b_j}{a_{ij}}$ было максимальным

Для оптимальной точки:

- Разрешающий столбец i — такой, где $c_i < 0$
- Разрешающая строка j — такая, чтобы $a_{ij} < 0$, и $\frac{b_j}{a_{ij}}$ было максимальным

  Оптимальная точка ищется после крайней, поэтому все b_j будут положительными

2. Строится новая симплекс-таблица.
   - x_i меняется местами с y_j
   - Элемент (i, j) возводится в -1 степень
   - Все остальные элементы разрешающего столбца делятся на (i, j)
   - Все остальные элементы разрешающей строки делятся на (i, j) и умножаются на -1.
   - Все остальное вычисляется следующему правилу:
     $a_{qv} \coloneqq \dfrac{a_{ij} \cdot a_{qv} - a_{iv} \cdot a_{qv}}{a_{ij}}$
3. Крайняя точка не существует, если в таблице существует строка, в которой все элементы ≤ 0, а последний - < 0
   Оптимальная точка не существует, если в таблице есть столбец, в котором $c_j < 0$; но все $a_{ij} > 0$
4. Если в столбце b есть отрицательные элементы, обратно на шаг 1. Иначе решение найдено
   Оптимальная точка ищется следующим образом:
   - Если x_j находится на $i - \text{м}$ месте левого столбца, то его значение равно b_i
   - Если x_i находится на j — м месте верхней строки, то его значение равно 0

Пример

Нужно найти минимум функции:

$f(x) = x_1 + x_2$

Ограничения:

$$\begin{cases} 3x_1 + 2x_2 - 6 > 0 \\ x_1 + 4x_2 - 4 > 0 \end{cases}$$

Симплекс-таблица:

	x_1	x_2	b
y_1	3	2	-6
y_2	1	4	-4
	1	1	0

1. Разрешающая строка - 1
   $-\frac{6}{3} > -\frac{6}{2} \Rightarrow a_{11}$ — разрешающий элемент

	y_1	x_2	b

x_1	3^{-1}	$-\frac{2}{3}$	$-\frac{-6}{3}$
y_2	$\frac{1}{3}$	$\frac{3\cdot 4-2\cdot 1}{3}$	$\frac{-4\cdot 3-(-6)\cdot 1}{3}$
	$\frac{1}{3}$	$\frac{3\cdot 1-2\cdot 1}{3}$	$\frac{3\cdot 0-(-6)\cdot 1}{3}$

	y_1	x_2	b
x_1	$\frac{1}{3}$	$-\frac{2}{3}$	2
y_2	$\frac{1}{3}$	$\frac{10}{3}$	-2
	$\frac{1}{3}$	$\frac{1}{3}$	2

2. Разрешающая строка - 2.
   $-\frac{2\cdot 3}{1} < -\frac{2\cdot 3}{10} \Rightarrow a_{22}$ – разрешающий элемент

	y_1	y_2	b
x_1	$\frac{1}{3}$	$-\frac{2}{3}$	2
x_2	$\frac{1}{3}$	$\frac{10}{3}$	-2
	$\frac{1}{3}$	$\frac{1}{3}$	2

	y_1	y_2	b
x_1	$\frac{\frac{1}{3}\cdot\frac{10}{3}+\frac{2}{3}\cdot\frac{1}{3}}{\left(\frac{10}{3}\right)}$	$-\frac{2}{3}\cdot\frac{3}{10}$	$\frac{2\cdot\frac{10}{3}-2\cdot\frac{2}{3}}{\left(\frac{10}{3}\right)}$
x_2	$-\frac{1}{3}\cdot\frac{3}{10}$	$\frac{3}{10}$	$-\left(-2\cdot\frac{3}{10}\right)$
	$\frac{\frac{1}{3}\cdot\frac{10}{3}-\frac{1}{3}\cdot\frac{1}{3}}{\left(\frac{10}{3}\right)}$	$\frac{1}{3}\cdot\frac{3}{10}$	$\frac{2\cdot\frac{10}{3}+2\cdot\frac{1}{3}}{\left(\frac{10}{3}\right)}$

	y_1	y_2	b
x_1	0.4	-0.2	1.6
x_2	-0.1	0.3	0.6

	0.3	0.1	2.2

Поиск ответа
Если x находится вверху, то это свободная переменная и он равен 0. Если x находится слева, он равен b.
Ответ: $x_1 = 1.6; x_2 = 0.6$

Методы прямого поиска

$f -$ унимодальная функция.
Общая схема метода: $\boxed{x^{(k+1)} = x^{(k)} + t_k}$
Нужно определить t_k. Для этого ищется минимум функции $y(t) = f(x^{(k)} + t)$

Метод квадратичной интерполяции

Пусть f задана на прямой. Даны точки $a < b < c$: $f(a) \geq f(b); f(b) \leq f(c)$. На отрезке $[a, c]$ ищется минимум.
Через эти три точки проводится парабола:
$g(t) = g_0 + g_1 t + g_2 t^2$
Коэффициенты находятся из системы:

$$\begin{cases} g(a) = f(a) \\ g(b) = f(b) \\ g(c) = f(c) \end{cases}$$

Решение.

$x_1 \coloneqq a; x_2 \coloneqq b; x_3 \coloneqq c; y_1 \coloneqq f(x_1); y_2 \coloneqq f(x_2); y_3 \coloneqq f(x_3)$

$$g_2 = \frac{y_3 - \dfrac{x_3(y_2 - y_1) + x_2 y_1 - x_1 y_2}{x_2 - x_1}}{x_3(x_3 - x_1 - x_2) + x_1 x_2}$$

$$g_1 = \frac{y_2 - y_1}{x_2 - x_1} - g_2(x_1 + x_2)$$

$$g_0 = \frac{x_2 y_1 - x_1 y_2}{x_2 - x_1} + g_2 x_1 x_2$$

После этого находится значение минимума:

$$t^* = \arg\min g(t) \Rightarrow -\frac{g_1}{2g_2}$$

После этого:

$a \leq t^* \leq b \Rightarrow c \coloneqq b; b \coloneqq t^*$
$b \leq t^* \leq c \Rightarrow a \coloneqq b; b \coloneqq t^*$

Метод дихотомии (половинного деления)

См. ВМ :: метод бисекции

Метод "золотого" сечения

$[a, b] -$ интервал. $t', t'' \in [a, b]$
Пусть минимум функции расположен на интервале.
Тогда выберем точки так:

$$\begin{cases} t' = a + F_1(b-a) \\ t'' = a + F_2(b-a) \end{cases}; F_1, F_2 \in (0,1)$$

Как и прежде, идея состоит в замене $[a, b]$ на $[a, t'']$ или $[t', b]$

Выбор коэффициентов

$\frac{b-t'}{b-a} = \frac{t''-a}{b-a} = F_2 \Rightarrow F_1 + F_2 = 1$, т.к.

$\frac{b-a-F_1(b-a)}{b-a} = 1 - F_1 = F_2$ - после замены отрезок уменьшится в $\frac{1}{F_2} = \tau$

По правилу "золотого" сечения

$\frac{t'-a}{t''-a} = F_2 \Rightarrow F_1 = F_2^2$, т.к. $\frac{F_1(b-a)}{F_2(b-a)} = \frac{F_1}{F_2} = F_2$

$$\begin{cases} F_1 = F_2^2 \\ F_1 + F_2 = 1 \end{cases} \Rightarrow F_2^2 + F_1 = 1 \Rightarrow \begin{cases} F_1 = \dfrac{3-\sqrt{5}}{2} \approx 0.382 \\ F_2 = \dfrac{\sqrt{5}-1}{2} \approx 0.618 \end{cases}$$

Алгоритм

$f(t') \geq f(t'') \Rightarrow a \coloneqq t'; t' = t''; t'' = a + F_2(b-a)$

$f(t') < f(t'') \Rightarrow b \coloneqq t''; t'' = t'; t' = a + F_1(b-a)$

II. Условная оптимизация

6 марта 2019 г. 11:46

Условная минимизация

- задачи поиска экстремума функции на множестве, заданном ограничениями в виде равенств и/или неравенств

$\overline{1,m} \coloneqq 1,\dots,m$

Задача нелинейного программирования

Задача минимизации функции f на множестве x, заданном набором ограничений

$X = \{\vec{x} \in \mathbb{R}^N, g_i(\vec{x}) \le 0; i = \overline{1,m}\}$

$g_1,\dots,g_m \in C^1$ ($\nabla g_1,\dots,\nabla g_m -$ существуют и непрерывны на X)

Ограничения типа равенства

Пусть $X = \{\vec{x} \in \mathbb{R}^2, g(x_1,x_2) = 0\}$. Нужно найти минимум $f \in C^1$ на X.

Пусть $\forall x_1, x_2: \frac{\partial g(x_1,x_2)}{\partial x_1} \ne 0 \Rightarrow g(x_1,x_2) = 0$ **разрешимо относительно $\boldsymbol{x}$**,

т.е. $\forall x_1, x_2: g(x_1,x_2) = 0 \Rightarrow x_1 = \gamma(x_2); \min_X f = \min_{x_2} f(\gamma(x_2), x_2)$

$f, \gamma -$ дифференцируемы. Условие экстремальности:

$$\frac{\partial f}{\partial x_1} \cdot \frac{\partial \gamma}{\partial x_2} + \frac{\partial f}{\partial x_2} = 0$$

$$g(\gamma(x_2), x_2) \equiv 0 \Rightarrow \frac{\partial g}{\partial x_1} \cdot \frac{\partial \gamma}{\partial x_2} + \frac{\partial g}{\partial x_2} = 0 \Rightarrow \frac{\partial \gamma}{\partial x_2} = -\frac{\partial g}{\partial x_2} \cdot \left(\frac{\partial g}{\partial x_1}\right)^{-1}$$

С учётом условия экстремальности:

$$-\frac{\partial f}{\partial x_1} \cdot \frac{\partial g}{\partial x_2} \cdot \left(\frac{\partial g}{\partial x_1}\right)^{-1} + \frac{\partial f}{\partial x_2} = 0$$

Обозначим $\lambda \coloneqq \frac{\partial f}{\partial x_1} \cdot \left(\frac{\partial g}{\partial x_1}\right)^{-1} \Bigg|_{x_1^*, x_2^*}$. Тогда

$$\begin{cases} \dfrac{\partial f}{\partial x_2} + \lambda \dfrac{\partial g}{\partial x_2} = 0 \\ \dfrac{\partial f}{\partial x_1} + \lambda \dfrac{\partial g}{\partial x_1} = 0 \\ g(x_1, x_2) = 0 \end{cases}$$

В точке минимума выполняются эти 3 соотношения. Их можно записать через функцию Лагранжа:

$F(\vec{x}, \lambda) = f + \gamma g$

Тогда необходимое условие минимума может быть записано следующим образом:

$$\begin{cases} \dfrac{\partial F}{\partial x_2} = \dfrac{\partial f}{\partial x_2} + \lambda \dfrac{\partial g}{\partial x_2} = 0 \\ \dfrac{\partial F}{\partial x_1} = \dfrac{\partial f}{\partial x_1} + \lambda \dfrac{\partial g}{\partial x_1} = 0 \\ \quad \dfrac{\partial F}{\partial \lambda} = g(x_1, x_2) \end{cases}$$

Таким образом, задача условной минимизации сведена к задаче безусловной минимизации.

Ограничения типа неравенств

Задача поиска минимума f на $X = \{\vec{x} \in \mathbb{R}^n, g_i(\vec{x}) \le 0; i = \overline{1, m}\}$, где f, g_i имеют непрерывные производные по всем аргументам в некотором открытом подмножестве, содержащем множество X.

Задача - получить необходимые условия экстремума f на X.

Пусть $\vec{x}^*$ — экстремальная точка. Свяжем с $\vec{x}^*$ множество индексов активных ограничений $I(\vec{x}^*) = \big\{i: g_i(\vec{x}^*) = 0; i \in \{1 \dots m\}\big\}$.

Лемма. Пусть $\exists \vec{S}$, $\forall i \in I(\vec{x}^*)$: $\begin{cases} (\nabla g_i(\vec{x}^*), S) < 0 \\ (\nabla f(x^*), S) < 0 \end{cases}$

Тогда $\vec{x}^*$ не является точкой локального минимума.

Доказательство. Пусть $\varepsilon > 0$. Тогда с помощью ряда Тейлора:

$g_i\big(\vec{x}^* + \varepsilon\vec{S}\big) = g_i(\vec{x}^*) + \varepsilon(\nabla g_i(\vec{x}^*), S) + o(\varepsilon)$

$f\big(\vec{x}^* + \varepsilon\vec{S}\big) = f(\vec{x}^*) + \varepsilon\big(\nabla f(\vec{x}^*), \vec{S}\big) + o(\varepsilon)$

Пусть $i \in I(\vec{x}^*) \Rightarrow g_i(\vec{x}^*) = 0, g_i\big(\vec{x}^* + \varepsilon\vec{S}\big) < 0$.

Если $i \notin I(\vec{x}^*)$, то $g_i(\vec{x}^*) < 0 \Rightarrow$ при всех i и достаточно малых положительных ε: $g_i\big(\vec{x}^* + \varepsilon\vec{S}\big) < 0$. Таким образом, при достаточно малых $\varepsilon > 0$ точка $\vec{x}^* + \varepsilon\vec{S}$ допустима и функция f на этом луче убывает. Значит, $\vec{x}^*$ — не экстремальна ■

Лемма Фаркаша

$\forall A_{m\times n}$ - матрицы $m \times n$ справедливо ровно одно из следующих условий:

- $\exists \vec{x} \in R^n$: $A\vec{x} < 0$ — все кординаты вектора $A\vec{x}$ отрицательны
- $\exists \vec{\lambda} \in \mathbb{R}^m, A^T\vec{\lambda} = 0, \lambda_i \ge 0, i \coloneqq \overline{1, m}$

Без доказательства

Теорема Каруша-Джона

Пусть $\vec{x}^*$ — экстремальная точка задачи нелинейного программирования. Пусть в точке $\vec{x}^*$ градиенты функций, соответствующие активным ограничениям, линейно независимы.

Тогда $\exists \lambda_1, \dots, \lambda_m \geq 0$:

$$f(x) = \begin{cases} \nabla f(x^*) + \sum_{i=1}^{m} \lambda_i \nabla g_i(x^*) = 0 \\ \lambda_i g_i(x^*) = 0, i = 1, \dots, m \end{cases}$$

Доказательство

$$\nexists S: \begin{cases} \left(\nabla g_i(\vec{x}^*), \vec{S}\right) < 0 \\ \left(\nabla f(\vec{x}^*), \vec{S}\right) < 0 \end{cases} \forall i \in I(\vec{x}^*)$$

Составим матрицу:

$$A = \begin{pmatrix} \nabla g_i(\vec{x}^*) \\ \dots \\ \nabla f(\vec{x}^*) \end{pmatrix}, i \in I(\vec{x}^*)$$

$\nexists \vec{S}: A\vec{S} < 0$. Значит, по лемме Фаркаша $\exists \lambda_0, \dots, \lambda_m \geq 0$:

$$\lambda_0 \nabla f(\vec{x}^*) + \sum_{i=1}^{m} \lambda_i \nabla g_i(\vec{x}^*) = 0$$

(полагаем $\lambda_i = 0$, если $i \notin I(\vec{x}^*)$)

Для активных ограничений $g_i(\vec{x}^*) = 0$, для неактивных $\lambda_i = 0$. Тогда $\lambda_i g_i(\vec{x}^*) = 0; i = \overline{1, m}$.
$\lambda_0 \neq 0$, т.к. иначе градиенты соответствующих активных ограничений были бы линейно зависимы.
Поделив предыдущее выражение на λ_0, получим требуемое утверждение.

<u>*Пример*</u>
Найти минимум $f(x_1, x_2) = x_1 x_2$ при ограничении $x_1^2 + x_2^2 \leq 1$.

$f(x_1, x_2) = x_1 x_2; g(x) = x_1^2 + x_2^2 \leq 1$

$$\begin{cases} \frac{\partial f}{\partial x_1} = x_2 \\ \frac{\partial f}{\partial x_2} = x_1 \end{cases} \Rightarrow \nabla f(x) = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

$$g(x) = x_1^2 + x_2^2 - 1 \leq 0$$

$$\begin{cases} \frac{\partial g}{\partial x_1} = 2x_1 \\ \frac{\partial g}{\partial x_2} = 2x_2 \end{cases} \Rightarrow \nabla g = \begin{pmatrix} 2x_1 \\ 2x_2 \end{pmatrix}$$

$$\Rightarrow \begin{cases} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \lambda \begin{pmatrix} 2x_1 \\ 2x_2 \end{pmatrix} = 0 \\ \lambda(x_1^2 + x_2^2 - 1) = 0 \end{cases}$$

$$\Rightarrow \begin{cases} x_2 + 2\lambda x_1 = 0 \\ x_1 + 2\lambda x_2 = 0 \\ x_1^2 + x_2^2 - 1 = 0 \end{cases} \Rightarrow$$

$$\Rightarrow \left[\begin{array}{l} \begin{cases} \lambda = -\frac{1}{2} \\ x_1 = x_2 = \pm\frac{1}{\sqrt{2}} \end{cases} \\ \begin{cases} \lambda = \frac{1}{2} \\ x_1 = \pm\frac{1}{\sqrt{2}} \\ x_2 = -x_1 \end{cases} \end{array} \right.$$

Задача выпуклого программирования

Пусть $g_1, \dots, g_m, f$ — выпуклы. Задача - поиск минимума f на $X = \{\vec{x} \in \mathbb{R}^n, g_i(\vec{x}) \leq 0, i = 1, \dots, m\}$.

Утверждение. Допустимое множество в задаче выпуклого программирования выпукло.

Доказательство. Пусть $\vec{x}_1, \vec{x}_2 \in X, \lambda \in [0,1]$.
Рассмотрим $\vec{z} = \lambda\overrightarrow{x_1} + (1-\lambda)x_2 \in X$. Т.к .$\mathbb{R}^n$ — выпукло, то $z \in \mathbb{R}^n$.
Надо проверить $g_i(\vec{z}) < 0$. По свойству выпуклости g:
$g_i(\lambda\vec{x}_1 + (1-\lambda)\vec{x}_1) \leq \lambda g_i(\vec{x}_1) + (1-\lambda)g_i(\vec{x}_2) \leq 0$
Тогда $\lambda\vec{x}_1 + (1-\lambda)\vec{x}_2 \in X$ по определению X.
X — пересечение выпуклых множеств, а значит, выпукло ■.

Функция Лагранжа в ЗВП

$$f(\vec{x}) + \sum_{i=1}^{m} \lambda_i g(\vec{x}) = f(\vec{x}) + \big(\lambda, g(\vec{x})\big)\,; \lambda_\iota \geq 0$$

Теорема Каруша-Джона:

$$\nabla f(\vec{x}^*) + \sum_{i=1}^{m} \lambda_i \nabla g_i(\vec{x}^*) = 0; \lambda_i g_i(\vec{x}^*) = 0; i = 1, \dots, m$$

Если X — выпукло, условие линейной независимости $\nabla g_i(\vec{x})$ можно заменить на **условия регулярности**

a) $\forall i \in 1, \dots, m{:}\ \exists \vec{x}_i \in X$, т.ч. $g_i(\vec{x}_i) < 0$ - **условие регулярности**
b) $\exists \vec{x} \in X$, т.ч. $\forall i \in 1 \dots m{:}\ g_i(\vec{x}) < 0$ - **условие регулярности Слейтера**

Очевидно, что $b \Rightarrow a$.
Пусть верно a. Поскольку X выпукло, можно выбрать $\vec{x}$:

$$\vec{x} := \sum_{i=1}^{m} \lambda_i \vec{x}_j, \sum_{i=1}^{m} \lambda_i = 1; \lambda_i \geq 0, i = 1 \dots m$$

В таком случае из b по неравенству Йенсена следует a.
Получается $a \Leftrightarrow b$.

Седловая точка

$(\vec{x}^*, \vec{y}^*)$ — седловая точка функции ψ аргументов $(\vec{x}, \vec{y})$ на $X \times Y$, если $\psi(\vec{x}^*, y) \leq \psi(\vec{x}^*, \vec{y}^*) \leq \psi(\vec{x}, \vec{y}^*)$ при $\forall \vec{x} \in X, \vec{y} \in Y$

Теорема о седловой точке

Пусть функция Лагранжа ЗВП имеет седловую точку, т.е.
$\forall x \in \mathbb{R}^n, \lambda_i \geq 0, i = 1 \dots m$.

$$f(\vec{x}^*) + \sum_{i=1}^{m} \lambda_i g_i(\vec{x}^*) \leq f(\vec{x}^*) + \sum_{i=1}^{m} \lambda_i^* g_i(\vec{x}^*) \leq f(\vec{x}) + \sum_{i=1}^{m} \lambda_i^* g_i(\vec{x})$$

Тогда $\vec{x}^*$ — оптимальная точка ЗВП.

Доказательство

По левому неравенству:

$$\sum_{i=1}^{m} \lambda_i g_i(\vec{x}^*) \leq \sum_{i=1}^{m} \lambda_i^* g_i(\vec{x}^*)$$

По определению X: $\lambda_i^* \geq 0, g_i(\vec{x}^*) \leq 0$.
Так как λ — любое, при $\lambda = 0$:

$$0 \leq \sum_{i=1}^{m} \lambda_i^* g_i(\vec{x}^*) \Rightarrow \left(\lambda^*, g(\vec{x}^*)\right) = 0$$

Из правого неравенства $\forall \vec{x} \in X$:

$$f(\vec{x}^*) + 0 \leq f(\vec{x}) + \sum_{i=1}^{m} \lambda_i^* g_i(\vec{x}^*) \leq f(\vec{x})$$

По определению оптимальной точки $\vec{x}^*$ — оптимальна.

Теорема Куна-Таккера.

Пусть в ЗВП выполнено условие регулярности Слейтера. Тогда для того, чтобы x^* была оптимальной точкой ЗВП, необходимо и достаточно, чтобы для некоторого вектора λ^* с неотрицательными компонентами точка (x^*, λ^*) была седловой точкой функции Лагранжа.

В частности, если ψ — дифференцируема, то

$$\begin{cases} \dfrac{\partial \psi^*}{\partial \vec{x}} = 0 \\ \dfrac{\partial \psi^*}{\partial \lambda} \le 0 \\ \left(\lambda^*, \dfrac{\partial \psi^*}{\partial \vec{\lambda}}\right) = 0 \\ \lambda^* \ge 0 \end{cases}, \text{где } \begin{matrix} \dfrac{\partial \psi^*}{\partial \vec{x}} \overset{\text{def}}{=} \dfrac{\partial \psi(\vec{x}, \lambda)}{\partial \vec{x}} \Bigg| \vec{x} = \vec{x}^*, \vec{\lambda} = \vec{\lambda}^* \\ \dfrac{\partial \psi^*}{\partial \vec{\lambda}} \overset{\text{def}}{=} \dfrac{\partial \psi(\vec{x}, \lambda)}{\partial \vec{\lambda}} \Bigg| \vec{x} = \vec{x}^*, \vec{\lambda} = \vec{\lambda}^* \end{matrix}$$

В частности, если на $\vec{x}$ наложены координатные органичения ($\vec{x} \ge 0$), то данные условия приобретают вид:

$$\begin{cases} \dfrac{\partial \psi^*}{\partial \vec{x}} \ge 0; \left(\vec{x}^*, \dfrac{\partial \psi^*}{\partial \vec{x}}\right) = 0; \vec{x}^* \ge 0 \\ \dfrac{\partial \psi^*}{\partial \lambda} \le 0; \left(\lambda^*, \dfrac{\partial \psi^*}{\partial \lambda}\right) = 0; \lambda \ge 0 \end{cases}$$

Доказательство

Достаточность - из теоремы о седловой точке

Необходимость - без доказательства

Методы условной минимизации

Метод проекции градиента

Обобщение градиентного метода для задачи условной минимизации с выпуклым допустимым множеством. Так как возможен выход за пределы допустимого множества, то вводится операция проектирования на X (поиск ближайшей точки на X).

$\vec{x}^{(k+1)} = p_x\left(\vec{x}^{(k)} - \gamma \nabla f\left(\vec{x}^{(k)}\right)\right)$, где $p_x -$ проектор на X

Метод условного градиента

В очередной точке $\vec{x}^{(k)}$ линеаризуют функцию $f(\vec{x})$.

Затем решают задачу минимизации линейной функции на X и найденную точку $\vec{x}^{(k)}$ используют для выбора направления движения.

$$\begin{cases} \overline{\vec{x}^{(k)}} = \underset{X}{argmin}\left(\nabla f\left(\vec{x}^{(k)}\right), \vec{x}\right) \\ \vec{x}^{(k+1)} + \gamma_k\left(\overline{\vec{x}^{(k)}} - \vec{x}^{(k)}\right) \end{cases}$$

Предполагается, что:

1. Задача минимизации линейной функции на X имеет решение
2. Это решение может быть найдено достаточно просто, лучше всего в явной форме
3. Нужно указать правило выбора γ_k. Значение γ_k можно определить из условия наискорейшего спуска:

   $\gamma = \underset{0 \le \gamma \le 1}{argmin} f\left(\vec{x}^{(k)} + \gamma\left(\vec{x}^{(k)} - \overline{\vec{x}^{(k)}}\right)\right)$

В этом случае последовательность $\vec{x}^{(k)}$ сходится к стационарной точке. В частности, для гладких функций f верно: $f(\vec{x}^*) - f^* = o\left(\frac{1}{k}\right)$, где $f^* = \min f(x)$ на множестве X.

Метод модифицированной функции Лагранжа

Функция Лагранжа в ЗВП:

$$\psi\left(\vec{x}, \vec{\lambda}\right) = f(\vec{x}) + \sum_{i=1}^{m} \lambda_i g_i(\vec{x}) = f(\vec{x}) + \left(\lambda, g(\vec{x})\right); \lambda_i \geq 0$$

По теореме о седловой точке, если:

$\forall \vec{x} \in \mathbb{R}^n, \lambda \geq 0: \psi\left(\vec{x}^*, \vec{\lambda}\right) \leq \psi\left(\vec{x}^*, \vec{\lambda}^*\right) \leq \psi\left(\vec{x}, \vec{\lambda}^*\right)$

То x^* — оптимальная точка задачи выпуклого программирования.

Это можно записать иначе:

$$\psi(x^*, \lambda^*) = \min_{x \in \mathbb{R}^n} \max_{\lambda \geq 0} \psi(\vec{x}, \vec{\lambda}) = \max_{\lambda \geq 0} \min_{x \in \mathbb{R}^n} \psi(\vec{x}, \vec{\lambda}) = f(\vec{x}^*)$$

Если $\vec{x}$ назвать **прямыми** переменными, а λ — двойственными, то видно, что прямые и двойственные переменные равноправны.

По теореме Куна-Таккера, исходную задачу можно заменить задачей отыскания седловой точки функции Лагранжа, т.е. задачи вида

$$\max_{\lambda \geq 0} \min_{x \in \mathbb{R}^n} \psi(\vec{x}, \vec{\lambda})$$

Модифицированная функция Лагранжа -

$$\mu\left(\vec{x}, \vec{\lambda}, k\right) = f(\vec{x}) + \frac{1}{2k}\left\|\vec{\lambda} + kg(\vec{x})_+\right\|^2 - \frac{\|\lambda\|^2}{2k}$$

Здесь k - некоторый параметр (штраф); + - взятие положительной часть.

Итерационная формула для вычисления $\{\vec{x}^{(k)}, \vec{\lambda}^{(k)}\}$ имеет вид:

$$\begin{cases} \vec{x}^{(k+1)} = \arg\min \mu\left(\vec{x}, \vec{\lambda}^{(k)}, k\right), \vec{x} \in \mathbb{R}^n \\ \lambda^{(k+1)} = \left[\vec{\lambda}_k + \gamma_k \nabla\lambda\mu\left(\vec{x}^{(k+1)}, \vec{\lambda}^{(k)}, k\right)\right]_+ \end{cases}$$

Метод сходится к $(\vec{x}^*, \vec{\lambda}^*)$ со скорость геометрической процессии.

Метод Эрроу-Гурвица

$$\begin{cases} \vec{x}^{(k+1)} = \vec{x}^{(k)} - \gamma\dfrac{\partial\psi}{\partial\vec{x}}\left(\vec{x}^{(k)}, \vec{\lambda}^{(k)}\right) = \vec{x}^{(k)} - \gamma\left(\nabla f\left(\vec{x}^{(k)}\right) + \left(\nabla g\left(\vec{x}^{(k)}\right), \lambda^{(k)}\right)\right) \\ \lambda^{(k+1)} = \lambda^{(k)} + \gamma\dfrac{\partial\psi}{\partial\lambda}\left(\vec{x}^{(k)}, \vec{\lambda}^{(k)}\right) = \vec{\lambda}^{(k)} + \gamma g\left(\vec{x}^{(k)}\right) \end{cases}$$

<u>Метод штрафных функций</u>

Идея метода - сведение задачи с ограничениями к последовательности задач без ограничений

$\min f(\vec{x}), \vec{x} \in X, X = \{\vec{x} \in \mathbb{R}^n, g_i(\vec{x}) \leq 0, i = 1 \dots m\}$

Функция $\psi(\vec{x})$, определённая и непрерывная всюду в $\mathbb{R}^n$, называется штрафной функцией для рассматриваемой задачи с ограничениями, если

$$\begin{cases} \psi(\vec{x}) = 0, \vec{x} \in X \\ \psi(\vec{x}) > 0, \vec{x} \in \dfrac{\mathbb{R}^n}{X} \end{cases}$$

Строится однопараметрическое семейство функций:

$\psi(\vec{x}, \beta) = f(\vec{x}) + \frac{1}{\beta}\psi(\vec{x})$, где $\beta -$ скалярный параметр, принимающий строго положительные значения.

Алгоритм

Выберем такую убывающую последовательность $\{\beta_k\}_{k=1}^{\infty}$ положительных чисел, что $\lim_{k\to\infty} \beta_k = 0$. Сопоставим каждому β_k соответствующую функцию семейства $\psi(\vec{x}, \beta)$. Получаем последовательность функций $\psi(\vec{x}, \beta_1), \dots, \psi(\vec{x}, \beta_k)$.

Пусть для каждой функции этой последовательности может быть решена задача условной минимизации:

$\arg\min \psi(\vec{x}, \beta) = \vec{x}_{\beta}^{*}, \vec{x} \in \mathbb{R}^n$

Оказывается, что при некоторых условиях последовательность оптимальных точек для задач без ограничений к оптимальной точке для исходной задачи с ограничениями:

$\vec{x}_{\beta}^{*} \xrightarrow[\beta\to 0]{} \vec{x}^{*}$

$x^{*} = \arg\min f(\vec{x})$

Наиболее распространенная следующая штрафная функция:

$$\psi(\vec{x}) = \sum_{i=1}^{m} \left(g_i^{+}(\vec{x})\right)^2$$

Здесь $g_i^{+}(\vec{x}) = \max\{0, g_i(\vec{x})\}$

<u>Двойственность задачи выпуклого программирования</u>

В теореме Куна-Таккера прямые и двойственные переменные используются симметричным образом. Поэтому можно ожидать, что аналогичная симметрия существует и для задач оптимизации относительно прямых и двойственных переменных.

Введем функцию: $g(\vec{x}) = \sup \psi(\vec{x}, \lambda)\,, \lambda \geq 0.$

Тогда очевидно, что $g_i(\vec{x}) = f(\vec{x}) \leq 0; i = 1, \dots, m$

$g(\vec{x}) = \infty -$ в противном случае.

Понятно, что $\psi\left(\vec{x}, \vec{\lambda}\right) = f(\vec{x}) + \left(\lambda, g(\vec{x})\right); \lambda \geq 0$

В таком случае исходная ЗВП может быть записана в виде:

$\min g(\vec{x}) -?\ \forall \vec{x} \in \mathbb{R}^n$. Эту задачу принято называть **прямой.**

Поступим аналогичным образом, поменяв роль переменных и операций

max и min. Обозначим $h(\lambda) = \inf \Psi(\vec{x}, \vec{\lambda})$ при $\vec{x} \in \mathbb{R}^n$. Задачу нахождения максимума функции $h\left(\vec{\lambda}\right) -?$ при $\lambda \geq 0$ называю **двойственной.**

Теорема двойственности

Справедливы следующие соотношения двойственности

1) $f(\vec{x}) \geq h\left(\vec{\lambda}\right); \forall \vec{x} \in X, \lambda \geq 0$
2) Если выполнено условие теоремы Куна-Таккера, а пара (x^*, λ^*) есть седловая точка ф-и Лагранжда, то λ^* — решение двойственной задачи: $\lambda^* = \arg\max h(\vec{\lambda})$ при $\vec{\lambda} \geq 0$ и $f(\vec{x}^*) = h\left(\vec{\lambda}^*\right)$
3) Если для допустимых $\vec{x}^*, \vec{\lambda}^*$: $f(\vec{x}^*) = h\left(\vec{\lambda}^*\right)$, то
   - $\vec{x}^* = \arg\min f(\vec{x})$ при $\vec{x} \in X$ — решение прямой задачи
   - $\vec{\lambda}^* = \arg\max h(\vec{\lambda})$ при $\vec{\lambda} \geq 0$ — решение двойственной задачи

Доказательство

1) $f(\vec{x}) \geq f(\vec{x}) + \left(\vec{\lambda}, g(\vec{x})\right) = \Psi\left(\vec{x}, \vec{\lambda}\right) \geq \inf \Psi\left(\vec{x}, \vec{\lambda}\right) = h(\vec{\lambda})$
2) $\forall \lambda \geq 0$:
   $h(\lambda^*) = inf\, \Psi(\vec{x}, \vec{\lambda}^*) = \Psi\left(\vec{\mathrm{x}}^*, \vec{\lambda}^*\right) \geq \Psi\left(\vec{\mathrm{x}}^*, \vec{\lambda}\right) \geq \inf \Psi\left(\vec{\mathrm{x}}, \vec{\lambda}\right) = h(\vec{\lambda})$
   Отсюда $\vec{\lambda}^*$ — решение двойственной задачи. Но $\Psi\left(\vec{x}^*, \vec{\lambda}^*\right) = f(\vec{x}^*) \Rightarrow$
   $h(\lambda^*) = f(x^*)$
3) На основании п.1 $f(\vec{x}) \geq h(\lambda^*) \underset{\text{по п.2}}{=} f(\vec{x}^*) \geq h(\vec{\lambda})$
   Тогда $\vec{x}^*$ — прямое решение, $\vec{\lambda}^*$ — двоственное решение.

Двойственность задачи линейного программирования

Рассмотрим множество $X = \{\vec{x} \in \mathbb{R}, \vec{x} \geq 0, A\vec{x} \geq b\}$.

$\vec{b} = (b_1, b_2, \dots, b_m)^T, A$ — матрица $m \times n$; $f(\vec{x}) = (\vec{c}, \vec{x})$ — целевая функция (линейная).

ЗЛП: $\min f(\vec{x}) -?$ при $\vec{x} \in \mathrm{X}$ — прямая задача линейного программирования

Построим функцию Лагранжа:

$\Psi\left(\vec{x}, \vec{\lambda}\right) = (\vec{c}, \vec{x}) + \left(\vec{\lambda}_1, \vec{b} - A\vec{x}\right) + \left(\vec{\lambda}_2 - \vec{x}\right), \vec{\lambda}_1 \in \mathbb{R}^m, \lambda_2 \in \mathbb{R}^n$

(удовлетворяет требованию $g_i(\vec{x}) \leq 0$).

Тогда $\min(\vec{c}, \vec{x}) = \max\inf((\vec{c}, \vec{x}) + \left(\vec{\lambda}_1, \vec{b} - A\vec{x}\right) + (\lambda_2, -\vec{x}))$ при $\vec{\lambda}_1 \geq 0, \vec{\lambda}_2 \geq 0, x \in \mathbb{R}^n = max\, inf\left(\left(\vec{c} - A^T\vec{\lambda}_1 - \lambda_2, \vec{x}\right) + \left(\vec{b}, \vec{\lambda}_1\right)\right) =$

$$= \max\begin{cases} -\infty, \text{если } \vec{c} - A^T\lambda_1 - \lambda_2 \neq 0 \\ (b, \lambda_1) \text{ если } \vec{c} = \vec{A}^T\vec{\lambda}_1 + \lambda_2 \end{cases} \text{при } \vec{\lambda}_1 \geq 0, \vec{\lambda} \geq 0$$

3. Линейное программирование

3.1. Основные понятия

Задача линейного программирования - поиск минимума линейной функции f на множестве $X = \{\vec{x} \in \mathbb{R} : g_j(\vec{x}) \leq 0, j = 1, ..., m\}$, где g_j - афинные. ЗЛП - частный случай ЗВП и ЗНП.

Функция - **линейная**, если

$$f(\lambda_1 \vec{x_1} + \lambda_2 \vec{x_2} = \lambda_1 f(\vec{x_1}) + \lambda_2 f(\vec{x_2}); \lambda \in \mathbb{R}, \vec{x_i} \in X$$

В n-мерном пространстве линейная функция может быть определена как:

$$f(\vec{x}) = (\vec{c}, \vec{x})$$

$$f(\vec{x}) = c_1 x_1 + ... + c_n x_n$$

Функция g - **афинная**, если $g - b$ - линейна для некоторого числа b. В координатной форме множество X описывается:

$$\begin{cases} a_{11}x_1 + ... + a_{1n}x_n \leq b_1 \\ ... \\ a_{m1}x_1 + ... + a_{mn}x_n \leq b_m \end{cases}$$

Если в системе используются только неравенства, **ЗЛП в стандартной форме**. Если в системе используются только равенства, **ЗЛП в канонической форме**.

ЗЛП в канонической форме эквивалентна некоторой задаче в стандартной форме. Чтобы избавится от нeравеств, нужно добавить в пространство ещё одну координату

$$\begin{cases} a_{11}x_1 + ... + a_{1n}x_n + x_{n+1} = b_1 \\ ... \\ a_{m1}x_1 + ... + a_{mn}x_n + x_{n+m} = b_m \end{cases}$$

Ограничения вида $(\lambda, x) =$ можно представить как совокупность $(\vec{\lambda}, \vec{x}) \leq 0$ и $(\vec{\lambda}, \vec{x}) \geq 0$.

Эти же ограничения можно записать в матричном виде:

$$X = \{A\vec{x} = \vec{b}, x \geq 0, x \in \mathbb{R}\}$$

где A - матрица $m \times n$, x - вектор размерности n, b - вектор размерности m.

Точка $\vec{x}$ выпуклого множества X называется **крайней (угловой)**, если $x = \lambda\vec{x_1} + (1 - \vec{x_2}) = \vec{x} \Rightarrow \vec{x_1} = \vec{x_2} = \vec{x}.$, где $\vec{x}, \vec{x_1}, \vec{x_2} \in X, \lambda \in [0,1].$, т.е. крайнюю точку нельзя выразить с помощью линейной комбинации других точек выпуклого множества.

Теорема (*О представлении*) Всякая точка допустимого ограниченного множества ЗЛП допускает представление в виде выпуклой комбинации его крайних точек. (Без доказательства)

Теорема (*О существовании оптимальной точки*) Если целевая функция на допустимом множестве ЗЛП ограничена снизу, то оптимальная точка существует. (Без доказательства)

Если допустимое множество X не пусто, то справедлива альтернатива:

- $\min\limits_X f(\vec{x}) = -\infty$
- $\exists \vec{x}^* : (\vec{c}, \vec{x}^*) = \min\limits_X \vec{c}, \vec{x}$ - существует оптимальная точка

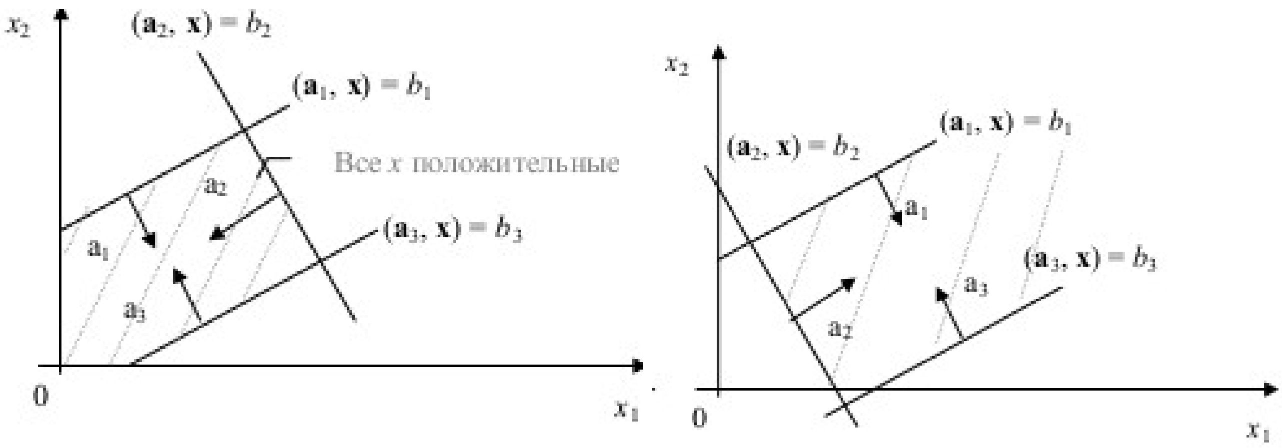
3.2. Геометрическая интерпретация ЗЛП

Пусть $X = \{A\vec{x} \geq \vec{b}, \vec{x} \geq 0\}$; $f(\vec{x}) = (\vec{c}, \vec{x})$ - целевая функция;

$$A = \begin{pmatrix} \vec{a_1} \\ \vdots \\ \vec{a_m} \end{pmatrix}; \vec{b} = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}$$

где $\vec{a_i}$ - строки длины n. В этом случае i-е условие принимает вид $(\vec{a}_i, \vec{x}) \geq b_i$. Множество точек, для которых справедливо это неравенство - полупространство с граничной гиперплоскостью $(\vec{a}_i, \vec{x}) = b_i$ - пересечение конечного числа полупространств.

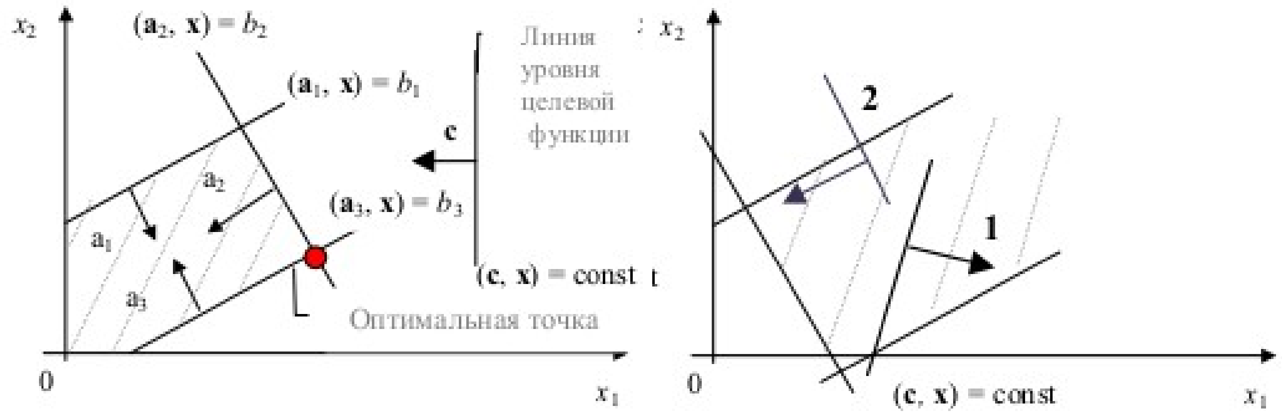
Гиперплоскость в $\mathbb{R}^n$ - пространство размерности $\mathbb{R}^{n-1}$



(a) Ограниченное (b) Неограниченное

Рисунок 1. Допустимые множества

Введем наряду с множествами целевую функцию:



(a) Есть оптимальная точка (b) Минимума нет

Рисунок 2. Поиск оптимальной точки

3.3. Условие оптимальности для ЗЛП

Лемма. Пусть $\vec{x}^*$ - оптимальная точка ЗЛП, пусть существуют такие точки $\vec{x_1}, ..., \vec{x_m} \in X$, что $\sum\limits_{i=1}^{m} \lambda_i \vec{x}_i$, где $\lambda_i > 0$ и $\sum\limits_{i=1}^{m} \lambda_i = 1$. Тогда $(\vec{c}, \vec{x}^*) = (\vec{c}, \vec{x_1}) = ... = (\vec{c}, \vec{x_m})$

Доказательство. По условию, $(\vec{c}, \vec{x}^*) \leq (\vec{c}, \vec{x}_1), ..., (\vec{c}, \vec{x}^*) \leq (\vec{c}, \vec{x_m})$.. По свойству скалярного произведения:

$$(\vec{c}, \vec{x^*}) = (\vec{c}, \sum_{i=1}^{m} \lambda_i \vec{x}_i) = \sum_{i=1}^{m} \lambda_i (\vec{c}, \vec{x}) \Rightarrow (\vec{c}, \vec{x}^*) = (\vec{c}, \vec{x_1}) = ... = (\vec{c}, \vec{x_m})$$

Теорема (*Об угловой точке*) Если для ЗЛП существует оптимальная точка $\vec{x}^*$, то найдется **угловая точка** $\vec{x}^0 \in \vec{X}$ такая, что $(\vec{c}, \vec{x}^*) = (\vec{c}, \vec{x}^0)$

Доказательство

1. Пусть X - ограничено. По теореме о представлении:

$$\vec{x}^* = \sum_{i=1}^{n} \lambda_i \vec{x_i}; \lambda_i > 0, \sum_{i=1}^{n} \lambda_i = 1$$

   x_i - угловые (крайние) точки множества X. В случае необходимости можно перенумеровать λ_i так, чтобы $\lambda_1 \neq 0, ..., \lambda_m \neq 0$, $\lambda_{m+1} = 0, ..., \lambda_n = 0$. Тогда Тогда

$$\vec{x}^* = \sum_{i=1}^{m} \lambda_i \vec{x_i}; \lambda_i > 0, \sum_{i=1}^{m} \lambda_i = 1$$

   По предыдущей лемме $(\vec{c}, \vec{x}^*) = (\vec{c}, \vec{x_1}) = ... = (\vec{c}, \vec{x_m})$. В качестве крайней можно взять любую угловую точку.

2. Пусть X - не ограничено, а $\vec{x}^*$ - оптимальная точка. Выберем некоторое число $\mu > 0$ и построим 2 множества:

$$L = \{\vec{x} \in \mathbb{R} : \sum_{j=1}^{n} \leq \mu\};\ l = \{\vec{x} \in \mathbb{R} : \sum_{j=1}^{n} = \mu\}$$

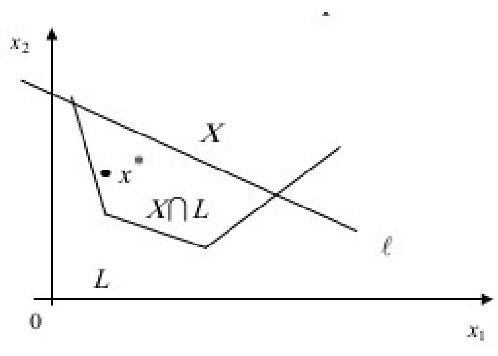


Рисунок 3. Вид множеств L, l

   Число μ выбирается так, что $\vec{x}^* \in L$, но $\vec{x}^* \notin l$. Тогда $\vec{x}^* \in X \cap L$, которое является ограниченным. Поэтому, по теореме о представлении,

$$\vec{x}^* = \sum_{i=1}^{m} \lambda_i \vec{x_i}; \lambda_i > 0, \sum_{i=1}^{m} \lambda_i = 1$$

$\vec{x_i}$ - угловые точки множества $X \cap L$.

Если хотя бы одна $\vec{x_i} \neq l$, теорема доказана. Иначе, из теоремы оп представлении следует, что $\vec{x}^* \in l$, что противоречит выбору числа μ.

3.4. Базис и базисное решение

Пусть $X = \{\vec{x} \in \mathbb{R} : \sum_{i=1}^{n} a_{ij}x_j = \vec{b_i}, i = 1, m,\ x_j \geq 0\}$; Требуется найти $\min\limits_X(\vec{c}, \vec{x}) = \min\limits_X \sum_{j=1}^{n} c_j x_j$

Введем векторы: $\vec{A_j} = (a_{1j}, ..., a_{mj})^T$, $\vec{B} = (b_1, ..., b_m)^T$, которые называются **векторами условий**. Тогда ЗЛП имеет вид:

$$\sum_{j=1}^{n} x_j \vec{A_j} = \vec{B}, x_j \geq 0, j = 1, n;\ \min_X(\vec{c}, \vec{x}) =?$$

Допустимое решение является **базисным**, если система векторов условий, отвечающих его положительным компонентам, линейно независима. При этом упомянутая система векторов(A_j) называется **базисом**.

Теорема (*О допустимом решении ЗЛП*). Допустимое решение является крайней точкой (*планом*) тогда и только тогда, когда оно базисное.

Доказательство. *Достаточность.* Пусть x - базисное решение. Без ограничения общности можно предположить, что первые k компонент x (решения) отличны от нуля, т.е. $\vec{x} = (x_1, ..., x_k, 0, ..., 0)^T$.

По определению базисного решения $\sum\limits_{j=1}^{n} x_j \vec{A_j} = \vec{B}$ и система векторов $\vec{A_1}, ..., \vec{A_k}$ линейно независима. Покажем, что x является крайней точкой. Допустим противное, т.е., что точка x может быть представлена в виде $x = \alpha\vec{x'} + (1-\alpha)\vec{x''}, 0 < \alpha < 1, x', x'' \in X$ - различные точки.

В силу того, что $\alpha > 0, 1 - \alpha > 0$, x' и x'' должны иметь вид:

$$\vec{x}' = (x'_1, ..., x'_k, 0, ..., 0)^T;\ \vec{x}'' = (x''_1, ..., x''_k, 0, ..., 0)^T$$

Так как $\vec{x}'$, $\vec{x}''$ - допустимые точки, то $\sum\limits_{j=1}^{n} x'_j \vec{A}_j = \vec{B}$, $\sum\limits_{j=1}^{n} x''_j \cdot \vec{A}_j = \vec{B}$, следовательно $\sum\limits_{j=1}^{n}(x'_j - x'' - j)\vec{A}_j = 0$

Из линейной независимости $\vec{A}_1, ..., \vec{A}_k$ следует, что $\vec{x}' = \vec{x}''$ - крайняя точка.

Необходимость. Пусть $\vec{x} = (x_1, ..., x_k, 0, ..., 0)^T$ является крайнейточкой X. Покажем, что x является базисным решением. Для этого достаточно показать, что система векторов $\vec{A}_1, ..., \vec{A}_k$ линейно независима.

Предположим противное. Тогда $\exists \alpha_j \neq 0$, такие, что $\sum\limits_{j=1}^{k} \alpha_j \vec{A}_j = 0$. Так как $\vec{x}$ - допустимое решение, то $\sum\limits_{j=1}^{k} x_j \vec{A}_j = \vec{B}$.

Рассмотрим при произвольном $\varepsilon > 0$ следующие выражения:

$$\sum_{j=1}^{k}(x_j + \varepsilon\alpha_j)\vec{A}_j = \vec{B} \ \sum_{j=1}^{k}(x_j - \varepsilon\alpha_j)\vec{A}_j = \vec{B}$$

Ясно, что ε можно подобрать так, чтобы точки

$$\vec{x}' = (x_1 + \varepsilon\alpha_1, ..., x_k + \varepsilon\alpha_k, 0, 0)^T, \ x'' = (x_1 = \varepsilon\alpha_1, ..., x_k - \varepsilon\alpha_k, 0, 0)^T$$

имели положительные компоненты, т.е. были бы допустимыми решениями. В таком случае $\vec{x} = \frac{\vec{x}' + \vec{x}''}{2}$, т.е. x - не крайняя точка. В таком случае предположение неверно, и $\vec{A}_1, ..., \vec{A}_k$ линейно независимы.

Следствие 1. Так как векторы $\vec{A}_1, ..., \vec{A}_n$ имеют размерность m, то крайняя точка, принадлежащая X, имеет не более чем m положительных компонент.

Следствие 2. Каждой угловой точке соответствует не более m ($k \leq m$) линейно независимых векторов системы $A_1, ..., A_n$.

Следствие 3. Число крайних точек множества X конечно.

3.5. Симплекс-метод решения ЗЛП

Задача линейного программирования:

$$X = \{\vec{x} \in \mathbb{R} : A\vec{x} \geq \vec{b}, x \geq 0\} \tag{3.1}$$

Симплекс метод решения ЗЛП состоит из двух этапов:

1. Поиск крайней точки допустимого множества
2. Перебор крайних точек с целью поиска оптимальной

1. Поиск крайней точки допустимого множества.
Расширим пространство состояний. Введем дополнительные координаты

$$X = \{\vec{x} \in \mathbb{R}^n : A\vec{x} - \vec{y} = \vec{b}, x \geq 0, y \geq 0\},\ \min_X(\vec{c}, \vec{x}) =? \tag{3.2}$$

Задачи 3.1, 3.2 эквивалентны, т.к. двойственной к обеим задачам будет $max(\vec{b}, \vec{\lambda})$ при $A^T\vec{\lambda} \leq c, \lambda \geq 0$

$$\begin{gathered}
1y_1 + 0 + ... + 0 = a_{11}x_1 + ... + a_{1n}x_n - b_1 \\
0 + 1y_2 + ... + 0 = a_{21}x_1 + ... + a_{2n}x_n - b_2 \\
.... \\
0 + 0 + ... + 1y_m = a_{m1}x_1 + ... + a_{mn}x_n - b_m
\end{gathered}$$

Переменные $y_1, ..., y_n$ можно рассматривать как базисные переменные, т.к. соответсвующие им векторы условий линейно независимы $(\vec{Y_1}, ..., \vec{Y_m})$.

Есть $n+m$ переменных и m уравнений. Тогда $x_1, ..., x_n$ можно рассматривать как свободные переменные, которые могут принимать произвольные значения. Как было показано ранее, для того, чтобы точка увеличенного пространства $(\vec{x}, \vec{y})^T$ была крайней, достаточно положить $\vec{x} = 0$, поэтому полагаем $x_1 := ...x_n := 0$.

Если все $y_i = -b_i \geq 0$, получим крайнюю точку допустимого множества. Если хотя бы один y_i отрицателен, меняем систему координат.

Переход к новому базису. Обозначим полученную крайнюю точку $\vec{x}' = (y_1 = -b_1, ..., y_m = -b_m, x_1 = 0, ..., x_n = 0)$. Ей соответствует разложение по базису:

$$y_1 + \vec{Y_1} + y_2\vec{Y_2} + ... + y_m\vec{Y_m} = \vec{B} \tag{3.3}$$

Разложим по базису каждый $\vec{A_j},\ j = \overline{1,n} : \sum\limits_{i=1}^{m} a_ij\vec{Y_i},$

Предположим, что для некоторого вектора, не входящего в базис (например, $\vec{A_s}$) положителен хотя бы один из коэффициентов a_is в разложении

$$a_1s + \vec{Y_1} + a_{2s}\vec{Y_2} + ... + ams\vec{Y_m} = \vec{A_s} \tag{3.4}$$

Зададимся значением $\Theta > 0$. Умножим на него обе части равенства 3.4 и вычтем результат почленно из равенства 3.3. Получим:

$$(y_1 - \Theta a_{1s})\vec{Y_1} + (y_2 - \Theta a_{2s})\vec{Y_2} + ... + (y_m - \Theta a_{ms})\vec{Y_m} + \Theta\vec{A_s} = \vec{B} \tag{3.5}$$

Получается, что вектор $\vec{x}'' = (y_1 - \Theta a_{1s}, ..., y_m - \Theta a_{ms}, 0, ..., 0, \Theta_s, 0, ..., 0)$ является допустимой крайней точкой, если его компоненты неотрицательны. Так как $\Theta > 0$, то все компоненты вектора $\vec{x}'$, в которые входят $a_{is} < 0$, положительны. Поэтому надо рассмотреть только компоненты, включающие положительные a_{is}. Т.е. надо определить такое $\Theta > 0$, при котором для всех $a_is > 0$ будет

$$y_i - \Theta a_{is} \geq 0 \Rightarrow 0 \leq \Theta \leq \frac{y_i}{a_is} \Rightarrow \Theta \leq \min_{i:a_{is}>0} \frac{y_i}{a_{is}}$$

Поскольку крайняя точка не может содержать $m + 1$ положительную компоненту, в $\vec{x}''$ нужно обнулить по крайней мере одну положительную компоненту. Поэтому надо положить:

$$\Theta = \Theta_0 = \min_{i:a_is>0} \frac{y_i}{a_{is}} = \frac{y_i}{a_{rs}}$$

И r-я компонента обратится в 0. Полученное значение подставим в 3.5:

$$(y - \frac{y_r}{a_{rs}}a_{1s}\vec{Y_1}) + ... + (y_r - \frac{y_r}{a_{rs}}a_{rs}) + ... + (y_m - \frac{y_r}{a_{rs}}a_{ms}\vec{Y_m}) + ... + \frac{y_r}{a_{rs}}\vec{A_s} = \vec{B}$$

В результате получается новое разложение:

$$y_1'\vec{Y_1} + ... + 0\vec{Y_r} + ... + y\_m\vec{Y_m} + y_s'\vec{A_s} = \vec{B}$$

которому соответствует новая крайняя точка $\vec{x}'' = (y_1', ..., 0, ..., y_m', 0, ..., 0, y_s', 0, ..., 0)$, где $y_i' = y_i - \Theta_0 a_{is}, i = \overline{1, m}, i \neq r, y_s' = \Theta_0$.

Рассмотрен переход от одной крайней точки к другой. Если на первом этапе симплекс-метода получается, что $-b_i < 0$, то нужно пользоваться формулой:

$$\Theta_0 = \max_{i: \frac{-b_i}{a_{is}} < 0} \frac{-b_i}{a_{is}}$$

Нужно осуществить преобразовие для всех векторов. Если при поиску допустимой крайней точке хотя бы один из $y_i < 0$, то необходимо изменить систему координат, т.е. перейти в другую крайнюю точку.

Пусть с помощью предыдущих рассуждений выбрали столбец s и строку r, т.е. будем ввводит в базис x_s и выводить из базиса y_r.

$$y_r = a_{r1}x_1 + ... + a_{rs}x_s + ... + a_{rn}x_n - b_r$$

Так как в базис вводится x_s, коэффициент при нём должен быть 1, а во всех остальных уравнения он должен быть 0. Вычислим x_s:

$$x_s = \frac{b_r}{a_{rs}} - \sum_{j=1; j \neq s}^{n} \frac{a_{rj}x_j}{a_{rs}} + \frac{y_r}{a_{rs}} \tag{3.6}$$

Подставив эту координаты во все остальные ограничения, получается:

$$y_i = \sum_{j=i, j \neq s}^{n} \left(a_{ij} - \frac{a_{is}a_{rj}}{a_{rs}} \right) x_s + \frac{a_{is}}{a_{rs}} y_r - b_i + \frac{a_{is}}{a_{rs}} b_r \tag{3.7}$$

Табличный метод - см. 1.4.1.

3.6. Транспортная задача

Некоторый однородный продукт сосредоточнен у m поставщиков в количестве a_i ($i = \overline{1,m}$) единиц. Необходимо доставить n потребителям B_j в количестве b_j ($j = \overline{1,n}$) единиц этого продукта. Известно, что c_{ij} - стоимость перевозки единицы груза от i-го поставщика к j-му потребителю. Необходимо составить план перевозок, позволяющий вывести все грузы, полностью удовлетворить потребности, имеющий минимальную стоимость.

Пусть x_{ij} - количество единиц груза, запланированного к перевозке от i-го поставщика к j-му потребителю.

Надо найти минимум $\min \sum\limits_{i=1}^{m} \sum\limits_{j=1}^{n} c_{ij}x_{ij}$, причем:

- $\sum\limits_{j=1}^{n} x_{ij} = a_i,\ i = \overline{1,m}$ - все грузы должны быть вывезены,
- $\sum\limits_{i=1}^{m} x_{ij} = b_j,\ j = \overline{1,n}$ - все потребители должны быть удовлетворены
- $x_{ij} \geq 0, i = \overline{1,m}, j = \overline{1,n}$

Если $\sum\limits_{i=1}^{m} a_i = \sum\limits_{j=1}^{n} b_j$, то модель называется **закрытой**. Можно показать, что любая закрытая транспортная задача имеет решение.

Транспортная задача - ЗЛП, но специфичная формой ограничений. Матрица ограничений имеет n+m строк:

1-я группа ограничений:

$$
\begin{array}{llll}
x_{11} + x_{12} + ... + x_{1n} & & & = a_1 \\
 & x_{21} + x_{22} + ... + x_{2n} & & = a_2 \\
\multicolumn{4}{c}{\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots} \\
 & & x_{m1} + x_{m2} + ... + x_{mn} & = a_m
\end{array}
\tag{3.8}
$$

2-я группа ограничений:

$$
\begin{array}{llllllll}
x_{11}+ & & x_{21}+ & & x_{m1} & & & = b_1 \\
 & x_{12}+ & & x_{22}+ & & x_{m2} & & = b_2 \\
\multicolumn{8}{c}{\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots} \\
 & & x_{1n}+ & & x_{2n}+ & & x_{mn} & = b_n
\end{array}
\tag{3.9}
$$

Нетрудно заметить, что матрица сильно разрешена. Например, для $n = 2, m = 3$:

$$A = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}; \; b = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ b_1 \\ b_2 \end{pmatrix}$$

Поэтому симплекс-метод будет работать очень медленно.

3.6.1. Построение первоначального опорного плана

В транспорной задаче $m \times n$ неизвестных и $m + n$ уравнений. Если сложить почленно системы 3.8 и 3.9, то получится 2 одинаковых уравнения, т.е. система линейно зависима.

Отбросим одно из уравнений системы. В общем случае система ограничений должна содержать $m + n - 1$ линейно независимых уравнений, следовательно, невырожденный опорный план транспортной задачи содержит $m + n - 1$ положительных компонент или перевозок (остальные равны 0). Рассмотрим конкретный пример В таблице должно быть

ПОСТАВЩИК	ПОТРЕБИТЕЛЬ				ЗАПАС
	B_1	B_2	B_3	B_4	
A_1	-5 $c_{11} = 5$	**5** $c_{12} = 1$	$+4$ $c_{13} = 4$	-4 $c_{14} = 7$	$a_1 = 5$
A_2	**3** $c_{21} = 2$	-3 $c_{22} = 6$	**− 2** $c_{23} = 10$	**+2** $c_{24} = 3$	$a_2 = 7$
A_3	-3 $c_{31} = 4$	**3** $c_{32} = 2$	$+6$ + $c_{33} = 3$	**−1** $c_{34} = 2$	$a_3 = 4$
ПОТРЕБНОСТЬ	$b_1 = 3$	$b_2 = 8$	$b_3 = 2$	$b_4 = 3$	$\sum a_i = \sum b_j$

цикл

Рисунок 4. Матрица планирования

$m + n - 1$ (здесь - 6) занятых клеток (> 0). Занятые клетки соотвествуют базисным неизвестным. Опорность (допустимость) плана заключается в ацикличности, т.е. в таблице нельзя построить замкнутый цикл, все вершины которого лежат в занятых клетках.

Цикл - набор клеток вида $(i_1, j_1), (i_1, j_2), (i_2, j_2), ..., (i_m, j_1)$, в котором каждые последующие 2 клетки располагаются в одном столбце, причем последняя клетка находится в той же строке или столбце, что и первая. Если цикл не найдет, план является опорным.

Метод минимальной стоимости. Один из методов построения первоначального плана. Из таблицы выбивается наименьшая стоимость, и в нужную клетку поменьшается $\min(a_i, b_j)$. Из рассмотрения исключается строки и столбцы с $a_i = 0$ и $b_j = 0$. Процесс продолжается, пока все запасы не будут распределены, а потребности - удовлетворены.

Для примера матрица стоимостей и результата метода:

	B_1	B_2	B_3	B_4	a
A_1	5	1	4	7	5
A_2	2	6	10	3	7
A_3	4	2	3	2	4
b	3	8	2	3	

;

	B_1	B_2	B_3	B_4	a
A_1		5			5
A_2	3		2	2	7
A_3		3		1	4
b	3	8	2	3	

$$1 \cdot 5 + 3 \cdot 2 + 2 \cdot 10 + 2 \cdot 3 + 3 \cdot 2 + 1 \cdot 2 = 45$$

3.6.2. Метод потенциалов

Лемма. Если при подстановке компонент оптимального плана в систему ограничений исходной задачи i-е ограничение обращается в неравенство, то i-я компонента оптимального плана двойственной задачи равна 0.
Если i-я компонента оптимального плана двойственной задачи положительна, то i-е ограничение исходной задачи удовлетворяется её оптимальным решением, как строгое равенство.

Теорема. Если план x^* является оптимальным, ему соответствует система из $m+n$ чисел u_i и v_j, удовлетворяющая условиям:

$$\begin{cases} u_i + v_j = c_{ij}, \text{ для } x^*_{ij} > 0 \\ u_i + v_j \leq c_{ij}, \text{ для } x^*_{ij} = 0 \end{cases}$$

Доказательство. Вместо решения исходной транспортной задачи решаем двойственную.
ЗЛП: $\min(\vec{c}, \vec{x})-?;\ A\vec{x} = \vec{E},\ \vec{x} \geq 0; \vec{E} = (a_1, ..., a_m, b_1, ..., b_n)^T$

Двойственная: $\min(\vec{E},\vec{\lambda})-?;\ A^T\lambda \leq \vec{c}, \lambda \geq 0$ Так как в строке A^T два ненулевых элемента $u_i = \lambda_i, v_j = \lambda_j$, получаем $u_i + v_j \leq c_{ij}$.

С учетном леммы двойственную задача можно записать в виде:

$$\max\left(\sum_{(}^{m} i = 1)a_iu_i + \sum_{j=1}^{n} b_jv_j\right)$$

где $u_j + v_j = c_{ij}$ для $x^*_{ij} > 0$, $u_i + v_j \leq c_{ij}$ для $x^*_{ij} = 0$. Теорема доказана.

Таким образом, для того, чтобы план был оптимальным, необходимо выполнение следующих условий:

- Для любой занятой клетки $u_i + v_j = c_{ij}$
- Для любой незанятой $u_i + v_j \leq c_{ij}$

Если хотя бы для одной клетки не удовлетворяется условие $u_i + v_j \leq c_{ij}$, то план не является оптимальным и его можно улучшить, введя в базис вектор, соответствующий клетке, для которой нарушается условие оптимальности.

Алгоритм метода потенциалов.

1. Построение системы потенциалов из условия $u_i + v_j = c_{ij}$ для занятых клеток. Поскольку уравнений не хватает, обнулить любую компоненту. Сделав $u_2 = 0$, получим

$$\begin{cases} u_1 + v_2 = 1 \\ u_2 + v_1 = 2 \\ u_2 + v_3 = 10 \\ u_2 + v_4 = 3 \\ u_3 + v_2 = 2 \\ u_3 + v_4 = 2 \end{cases} ; \quad \begin{array}{ccccc|cc} & & & & & u & \\ & & & & & -2 & (1) \\ & & & & & 0 & (2) \\ & & & & & -1 & (3) \\ \hline v & 2 & 3 & 10 & 3 & & \\ & (1) & (2) & (3) & (4) & & \end{array}$$

2. Проверка условия оптимальности для незанятых клеток
   Характеристические разности: $q_{ij} = u_i + v_j - c_{ij}$. Если в любой незанятой клетке

$q_{ij} \leq 0$, то конец.

$Q =$

	B_1	B_2	B_3	B_4	b
A_1	-5		$+4$	-4	5
A_2		-3			7
A_3	-3	$+6$		4	
b	3	8	2	3	

3. Выбор клетки, в которую необходимо послать перевозку. Для этого среди $q_{ij} \geq 0$ выбрать максимальную
4. Построение цикла и определение величины перераспределения груза.
   Найденная клетка отмечается плюсом. Теперь занятых клеток $m+n$, т.е. существует цикл. Нужно пройти по циклу, чередуя плюсы и минусы.

	B_1	B_2	B_3	B_4	b
A_1					5
A_2			$-$	$+$	7
A_3			$+$	$-$	4
b	3	8	2	3	

   По минусам, найденным ранее, значение $\Theta_0 = \min\limits_{(-)} x_{ij}$ записвается в незанятую клетку. Затем, идя по циклу, сделать $x_{ij} + \Theta_0$ там, где $+$ и $x_{ij} - \Theta_0$ там, где i. Для рассматриваемого примера $\Theta_0 = 1$ и новые значения:

	B_1	B_2	B_3	B_4	b
A_1		5			5
A_2	3		1	3	7
A_3		3	1		4
b	3	8	2	3	

$$f = 5 \cdot 1 + 3 \cdot 2 + 1 \cdot 10 + 3 \cdot 3 + 3 \cdot 2 + 1 \cdot 3 = 39$$

   Значение функции уменьшилось.
5. Новый план проверяется на оптимальность, и, в случае необходимости, переход к п. 1.

4. Вариационное исчисление

Задача вариационного исчисления - поиск экстремумов функционалов, аргументами которых являются функции. Требуется найти минимум

$$J(y(x)) = \int_a^b f(x, y(x), y'(x))dx$$

по множеству дважды дифференцируемых функции y с краевыми условия $y(a) = y_0, y(b) = y_1$. Определение нормы на множестве: $\|y(x)\| = \max\limits_{a<x<b}$ { |y(x)| + |y'(x)| }.

Пусть $y^*(x)$ доставляет минимум $J(y(x))$.

Рассмотрим $y(x) = y^*(x) + \varepsilon(x)$, где $\varepsilon(a) = \varepsilon(b) = 0$, и ε - гладкая функция, которая называется **вариацией**.

Пусть $y_\alpha = y^* + \alpha\varepsilon$, тогда

$$J(y_\alpha(x)) = \int_a^b f(x, y^* + \alpha \cdot \varepsilon, y'^* + \alpha \cdot \varepsilon')dx \geq J(y^*(x)),$$

так как y^* - аргумент минимума функционала. Из этого неравенства можно получить необходимые условия экстремума.

4.1. Уравнение Эйлера-Лагранжа

Лемма Дюбуа-Раймона. Пусть для некоторой функции f, непрерывной на $[a, b]$, и для всех функций h, непрерывных на $[a, b]$ вместе со своими производными и таких, что $h(a) = h(b) = 0$, справедливо:

$$\int_a^b f(x)h(x) = 0 \Rightarrow f \equiv 0$$

Разложим $J(y_\alpha(x))$ в ряд Тейлора в окрестности точки $\alpha = 0$:

$$J(y_\alpha(x)) = J(y^*) + \alpha\frac{\partial J}{\partial \alpha} + o(\alpha)$$

$\delta\dfrac{\partial J}{\partial\alpha}$ - первая вариация функционала При фиксированных $y^*(x)$ и $\varepsilon(x)$ функционал зависит от параметра α и необходимым условием экстремума функционала является.

$$\left.\frac{\partial J}{\partial\alpha}\right|_{\alpha=0} = 0 \Rightarrow \delta J = 0$$

Получим необходимое условие экстремума функционала в более конструктивной форме. Функционал J, рассматриваемый как функция от α, может быть записан в виде:

$$J(\alpha) = \int_a^b f(x, y(x) + \alpha\varepsilon(x), y'(x) + \alpha\varepsilon'(x))dx$$

Тогда необходимое условие экстремума:

$$\left.\frac{\partial J(\alpha)}{\partial\alpha}\right|_{\alpha=0} = \int_a^b \left[\frac{\partial f(x,y,y'}{\partial y}\varepsilon(x) + \frac{\partial f(x,y,y')}{\partial y'}\varepsilon'(x)\right]dx = 0$$

$$\int_a^b \frac{\partial f}{\partial y'}\varepsilon'(x)dx = \underbrace{\left.\frac{\partial f}{\partial y'}\varepsilon(x)\right|_a^b}_{=0} - \int_a^b \varepsilon(x)\frac{d}{dx}\frac{\partial f}{\partial y'}dx$$

$= 0$, т.к. $\varepsilon(a) = \varepsilon(b) = 0$

$$\frac{\partial J(\alpha)}{\partial\alpha} = \int_a^b \left(\frac{\partial f(x,y,y')}{\partial y} - \frac{d}{dx}\cdot\frac{\partial f(x,y,y')}{\partial y'}\right)\varepsilon(x)dx = 0$$

Это справедливо с учётом $\varepsilon(a) = \varepsilon(b) = 0$. Отсюда с учётом леммы следует:

$$\frac{\partial f(x,y^*,y'^*)}{\partial y} - \frac{d}{dx}\frac{\partial f(x,y^*y'^*)}{\partial y'} \equiv 0$$

- **уравнение Эйлера-Лагранжа**

Всякое решение уравнения Эйлера-Лагранжа называется **Экстремалью**. Функция, доставляющая экстремум функционалу, находится среди экстремалей. Уравнение Эйлера-Лагранжа - необходимое условие экстремума.

Одно из достаточных условий локального минимума - **условие Лежандра**:

$$\frac{\partial^2 f(x, y^*(x), y'^*(x))}{\partial y'\partial y'} > 0$$

Обозначим $f_y = \frac{\partial f}{\partial y}$, $f_x = \frac{\partial f}{\partial x}$. Тогда уравнение Эйлера-Лагранжа:

$$f_y - \frac{d}{dx} f_{y'} = 0$$

Раскроем второй член по формуле дифференцирования сложной функции:

$$\frac{d}{dx} f_{y'} = \frac{\partial f_{y'}}{\partial x} \cdot \frac{dx}{dx} + \frac{\partial f_{y'}}{\partial y} \cdot \frac{dy}{dx} + \frac{\partial f_{y'}}{\partial y'} \cdot \frac{dy'}{dx}$$

Тогда уравнение:

$$f_y - f_{y'x} - f_{y'y} \cdot y' - f_{y'y'} y'' = 0$$

Это нелинейное дифференциальное уравнение второго порядка, поэтому его решение затруднено.

4.1.1. Частные случаи уравнения Эйлера-Лагранжа

1. $f(x, y)$

$$\frac{\partial f(x, y)}{\partial y'} = 0 \Rightarrow \frac{\partial f(x, y)}{\partial y} \equiv 0$$

2. f(x, y')

$$\frac{f}{y} = 0 \Rightarrow \frac{d}{dx} \cdot \frac{\partial f}{\partial y'} \equiv 0 \Rightarrow \frac{\partial f}{\partial y'} = const$$

3. $f(y, y')$

$$f_y - f_{y'y} y' - f_{y'y'} y'' \equiv 0$$

$$\frac{d}{dx}(f - f_{y'} y') \equiv 0 \Rightarrow f - f_{y'} y' = const$$

4.1.2. Задача о брахистохроне

Как выбрать профиль горки, чтобы шарик скатился с нёё за минимальное время? Трение отсутствует, v - скорость, φ - угол наклона горки, g - ускорение свободного падения.

$$dx = v \cos\varphi dt \tag{4.1}$$

$$\tan\varphi = y'(x) \Rightarrow \frac{1}{\sqrt{1 + (y'(x)^2)}} \tag{4.2}$$

Из закона сохранения энергии:

$$\frac{mv^2}{2} = mgy \Rightarrow v = \sqrt{2gy} \tag{4.3}$$

Подстави 4.2, 4.3 в 4.1, получим

$$\frac{dx}{\sqrt{2gy}}\sqrt{1+y'^2}$$

Проинтегрируем обе части:

$$\int_0^{x_k} \underbrace{\frac{\sqrt{1+y'^2}}{\sqrt{2gy}}}_{F} dx = T$$

Здесь T - время спуска шарика. F зависит от y и y', т.е. это третий частный случай уравнения Эйлера-Лагранжа - $F - y'F_{y'} \equiv const$. $2g$ можно вынести из рассмотрения, т.к. оно не влияет на экстремум.

$$\frac{\sqrt{1+y'^2}}{\sqrt{y}} - y'\frac{1}{2} \cdot \frac{2y'}{\sqrt{1+y'^2}} \cdot \frac{1}{\sqrt{y}} \equiv const$$

Приведем к общему знаменателю:

$$\frac{1}{\sqrt{y(1+y'^2)}} \equiv const$$

Пусть $c = \frac{1}{const^2} \Rightarrow const = \frac{1}{\sqrt{c}}$. Тогда

$$y(1+y'^2) \equiv c$$

Делаем подстановку:

$$y' = \tan t \Rightarrow 1 + y'^2 = \frac{1}{\cos^2 t}$$

Получается $y = c\cos^2 t$. Найдем выражение для x:

$$\left.\begin{aligned} &dy = \tan t t d \\ &BCdy = 2c\cos t(-\sin t)dt \end{aligned}\right\} \Rightarrow dx = 2c(-\cos^2 t)dt$$

Интегрируем:

$$x = -2c \int \cos^2 t dt = -2c \int \frac{1 - \cos 2t}{2} dt = -ct - c/2 \sin 2t$$

Полученная кривая - **циклоида**.

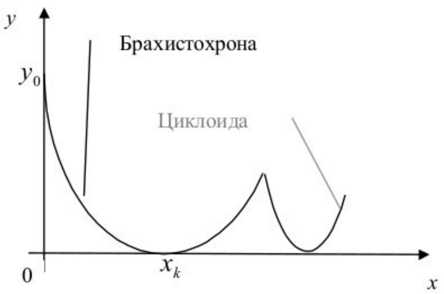


Рисунок 5. Результат решения

4.2. Вариационные задачи на условные экстремум

Будем считать, что y - вектор-функция из m компонент: $y = (y_1, ..., y_m)^T$. Если они независимы, то уравнение Эйлера-Лагранжа надо писать для каждой компоненты y_i отдельно. Если между y_i есть связь, то получаем задачу на условный экстремум.

4.2.1. Модельные задачи на условный экстремум

1. **Геодезическая линия.** Пусть поверхность задана уравнением $\phi(y_1, y_2, x) = 0$, где ϕ - дифференцируемая функция, x - независимая координата, y_1, y_2 - функции от x. Тогда длина пути, соединяющего 2 точки:

$$\int_{x_0}^{x_k} \sqrt{1 + y_1'^2 + y_2'^2} dx$$

   Условия здесь - начало и конец геодезической линии

2. **Изопериметрическая задача.** Требуется найти линию заданной длины, ограничивающую максимальную площадь. Задача может быть поставлена так: на множестве дифференцируемых функций y с заданными граничными точками, удовлющих равенству,

$$\int_{t_0}^{t_k} \sqrt{y'^2(t) + x'^2(t)} dt = l$$

Нужно найти максимум фунционала:

$$S = \int_{t_0}^{t_k} y(t)x'(t)dt$$

Связи типа 1 - **локальные**, 2 - **интегральные**. Эти задачи решаются методом множителей Лагранжа.

4.2.2. Метод множителей Лагранжа

Локальные ограничения: $\varphi_i(x, y, y') = 0, i = \overline{1, k}$ - сильные связи
Интегральные: $\int \psi(x, y, y')dx = l, l = \overline{1, p}$ - слабые связи. Составим функционал F^*:

$$F^* = F + \sum_{i=1}^{k} \lambda_i(x)\varphi(x, y, y') + \sum_{j=1}^{p} \lambda_{k+j}\psi_j(x, y, y')$$

Количество неизвестных функций увеличилось за счёт введения множителей λ_i.

Метод градиентного спуска

Решаемая задача - поиск локального минимума функции $f(x) : \mathbb{R}^n \to \mathbb{R}$

Описание метода

Основная идея - использование градиента для движения в направлении наискорейшего спуска
$\vec{x} = (x_1, \ldots, x_n)$
$\nabla f(\vec{x}) = (\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n})$ - градиент
Шаг метода выглядит следующим образом:
$\vec{x}^{[i+1]} = \vec{x}^{[i]} - \lambda^{[i]} \bullet \nabla f(\vec{x}^{[i]})$
Здесь $\lambda^{[i]}$ - скорость градиентного спуска. Её можно выбрать различными способами:

- $\lambda^{[i]} = \mathrm{const}$. В этом случае алгоритм может расходится
- Скорость убывает в процессе спуска
- Скорость гарантирует наискорейший спуск

Критерий остановки

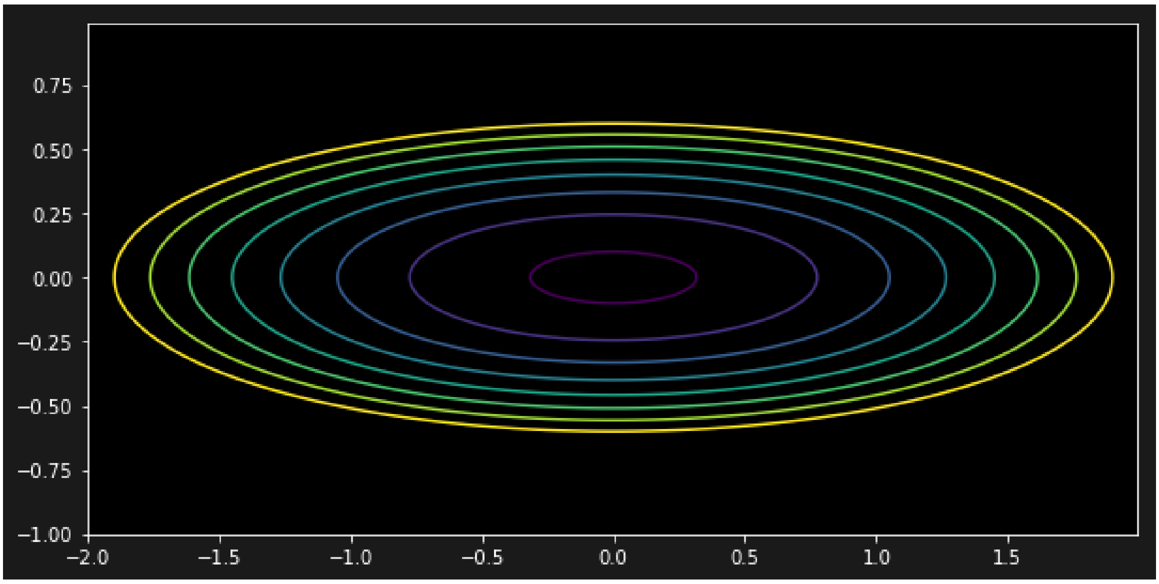
Есть несколько вариантов задания критерия остановки:
$|\vec{x}^{[i+1]} - \vec{x}^{[i]}| < \varepsilon$
$|f(\vec{x}^{[i+1]}) - f(\vec{x}^{[i]})| < \varepsilon$
$|\nabla f(x^{[i+1]})| < \varepsilon$
Очевидная проблема первых двух критериев в том, что при уменьшении шага придется уменьшать ε.
Третий критерий может просто никогда не сработать, если метод не сходится

Метод градиентного спуска с постоянным шагом

В данном случае скорость спуска задаётся константой. Рассмотрим следующую функцию:
$f(x, y) = x^2 + 10y^2$.
Частные производные такой функции:
$\frac{\partial f}{\partial x} = 2x$
$\frac{\partial f}{\partial y} = 20y$
В данном случае трехмерный рисунок будет нерепрезентативен, поэтому построим двумерный график с изолиниями (contour plot)

In [1]:

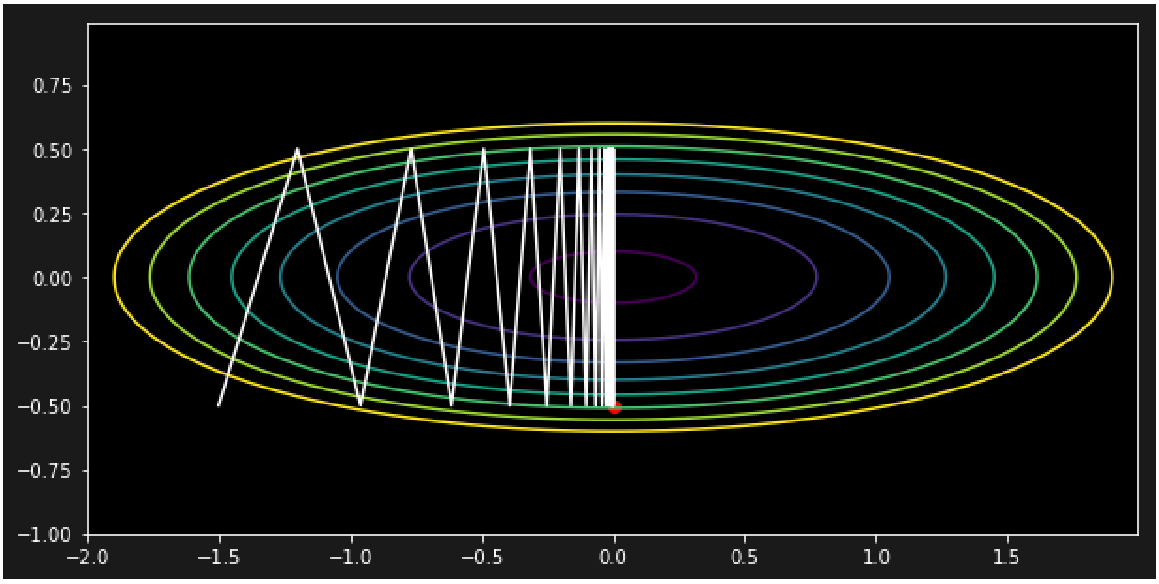
```
from matplotlib import pyplot as plt
import numpy as np
%matplotlib inline
plt.style.use('dark_background')

f = lambda x, y : np.power(x, 2) + 10 * np.power(y, 2)

plt.figure(facecolor='0.1', figsize=(10, 5))
X = np.arange(-2, 2, 0.01)
Y = np.arange(-1, 1, 0.01)
levels = np.arange(0.1, 4, 0.5)
X, Y = np.meshgrid(X, Y)
Z = f(X, Y)
plt.contour(X, Y, Z, levels=levels)
plt.show()
```



Теперь реализуем вычисление корня методом градиентного спуска

In [2]:

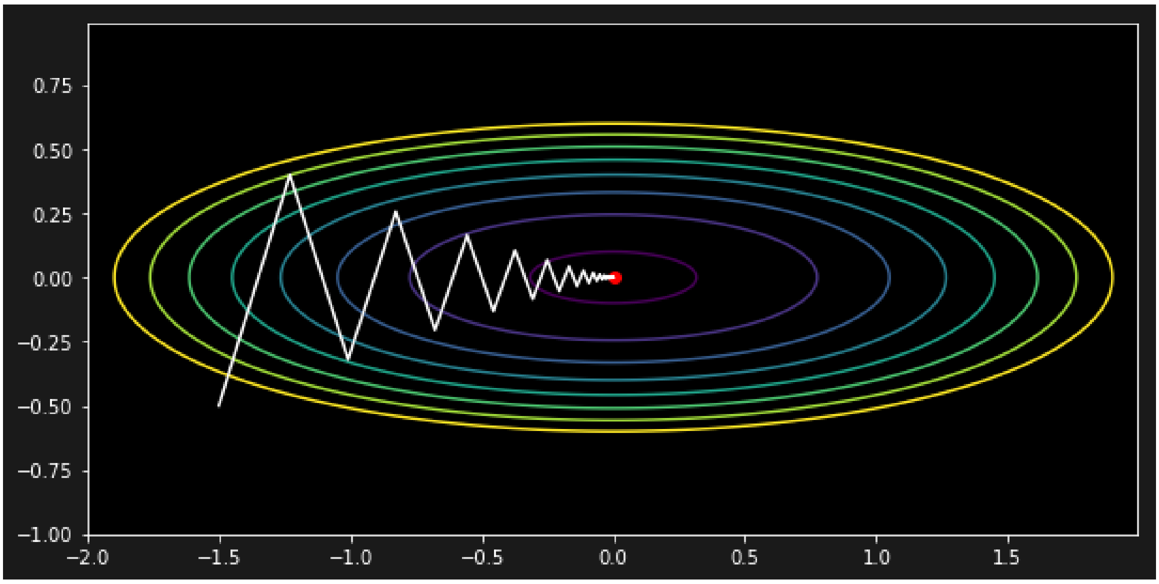
```
from numpy import linalg

def grad_descent_const(grad, x, y, eps, a):
    """
        Градиентный спуск с постоянным коэффициентом
        grad - функция градиента
        x, y - начальная точка
        eps - точность
        a - коэффициент
        Возвращаемые значения - путь, результат, особые точки (ни одной в данном случае)
    """
    X = np.array([x, y])
    cnt = 0
    path = [X]
    while True:
        X1 = X - a * grad(*X)
        path.append(X1)
        if (linalg.norm(grad(*X)) < eps
            or cnt > 1000):
            break
        cnt += 1
        X = X1
    return path, X1, None

def plot_grad_descent(grad_descent_func, grad, x, y, **kwargs):
    path, res, pts = grad_descent_func(grad, x, y, **kwargs)
    p_X, p_Y = [p[0] for p in path], [p[1] for p in path]
    plt.figure(facecolor='0.1', figsize=(10, 5))
    plt.scatter(*res, color='r')
    if pts:
        x, y = [p[0] for p in pts], [p[1] for p in pts]
        plt.scatter(x, y, color='w', marker='o')
    plt.contour(X, Y, Z, levels=levels)
    plt.plot(p_X, p_Y, color='w')
    return path, res, pts

dfdx = lambda x, y: 2 * x
dfdy = lambda x, y: 20 * y
gradient = lambda x, y: np.array([dfdx(x, y), dfdy(x, y)])
plot_grad_descent(grad_descent_const, gradient, -1.5, -0.5, eps=0.1, a=0.1)
pass
```



Как видно, с такими параметрами польза метода сомнительна - нет сходимости. Поробуем немного уменьшить размер шага

In [3]:

```
plot_grad_descent(grad_descent_const, gradient, -1.5, -0.5, eps=0.0001, a=0.09)
pass
```



Сравним различные варианты задания отрезка

In [4]:

```
import pandas as pd
from IPython.display import display
a_r = np.arange(0.1, 0.01, -0.01)
data = [grad_descent_const(gradient, -1.5, -0.5, eps=0.0001, a=a) for a in a_r]
data = [[len(path), linalg.norm(X)] for path, X, pts in data]
df = pd.DataFrame(data, columns=['Iterations', 'Precicion'])
df['eps'] = pd.Series([0.0001] * len(a_r))
df['a'] = pd.Series(a_r)
display(df)
pass
```

	Iterations	Precicion	eps	a
0	1003	0.500000	0.0001	0.10
1	56	0.000027	0.0001	0.09
2	62	0.000036	0.0001	0.08
3	71	0.000039	0.0001	0.07
4	83	0.000042	0.0001	0.06
5	100	0.000044	0.0001	0.05
6	126	0.000045	0.0001	0.04
7	169	0.000046	0.0001	0.03
8	255	0.000047	0.0001	0.02

Как можно заметить, как только метод начинает сходится, точность самая высокая. Таким образом, лучше найти максимальное значение константы, при котором метод начинает сходится

Градиентный метод с дроблением шага

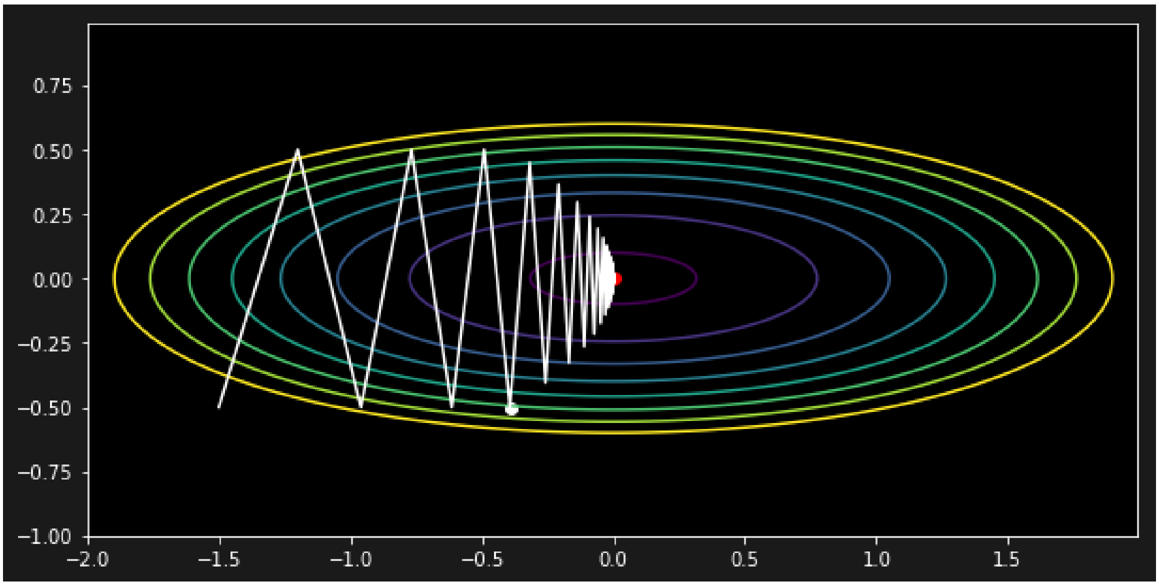
Для каждого шага можно записать условие сходимости
$f(\vec{x}^{[i+1]}) = f(\vec{x}^{[i]} - \lambda^{[i]} \bullet \nabla f(\vec{x}^{[i]})) \leq f(\vec{x}^{[i]}) - \varepsilon\lambda^{[i]} |\nabla f(\vec{x}^{[i]})|^2$
Где $\varepsilon \in (0, 1)$
На каждом шаге выполняется проверка условия. Если условие выполняется, то алгоритм продолжается. Если нет - то шаг делится на число $\delta \in (0, 1)$, пока условие не выполнится. После этого проверка продолжается

In [5]:

```
def grad_descent_splt(grad, x, y, f, eps, e, d, a):
    """
        Градиентный метод с дроблением шага
        grad - функция градиента
        x, y - начальная точка
        eps - точность для условия остановки
        e - коэффициент для проверки условия сходимости
        d - коэффициент дробления шага
        a - начальное значение коэффициента
        f - анализируемая функция
        Возвращаемые значения - путь, результат, особые точки
    """
    X = np.array([x, y])
    cnt = 0
    path = [X]
    pts = []
    while True:
        while True:
            check = f(*(X - a * grad(*X))) <= f(*X) - e * a * f(*X)**2
            if check:
                break
            pts.append(X)
            a *= d
        X1 = X - a * grad(*X)
        path.append(X1)
        if (linalg.norm(grad(*X)) < eps
            or cnt > 1000):
            break
        cnt += 1
        X = X1
    return path, X1, pts
```

In [6]:

```
plot_grad_descent(grad_descent_splt, gradient, -1.5, -0.5, eps=0.0001, e=0.1, d=
0.95, a=0.1, f=f)
pass
```



С коэффициентом 0.1 метод с постоянным шагом бы не сошёлся, но в данном случае в одном месте произведено уменьшение коэффициента, и метод начал сходится.
Выполним вычисление с разными параметрами

In [7]:

```
params = pd.DataFrame({
    "eps": [0.0001] * 5,
    "e": [0.95, 0.1, 0.1, 0.1, 0.9],
    "d": [0.95, 0.95, 0.1, 0.95, 0.95],
    "a": [1, 1, 1, 0.1, 0.1]
})
iteration, precision, split = [], [], []
for index, row in params.iterrows():
    path, res, pts = grad_descent_splt(gradient, -1.5, -0.5, eps=row['eps'],
                                      e=row['e'], d=row['d'], a=row['a'], f=f)
    split.append(len(pts))
    iteration.append(len(path))
    precision.append(linalg.norm(res))

params['Iterations'] = pd.Series(iteration)
params['Precision'] = pd.Series(precision)
params['Split times'] = pd.Series(split)
display(params)
```

	eps	e	d	a	Iterations	Precision	Split times
0	0.0001	0.95	0.95	1.0	58	0.000035	48
1	0.0001	0.10	0.95	1.0	1003	0.000007	45
2	0.0001	0.10	0.10	1.0	453	0.000048	2
3	0.0001	0.10	0.95	0.1	118	0.000004	1
4	0.0001	0.90	0.95	0.1	57	0.000040	3

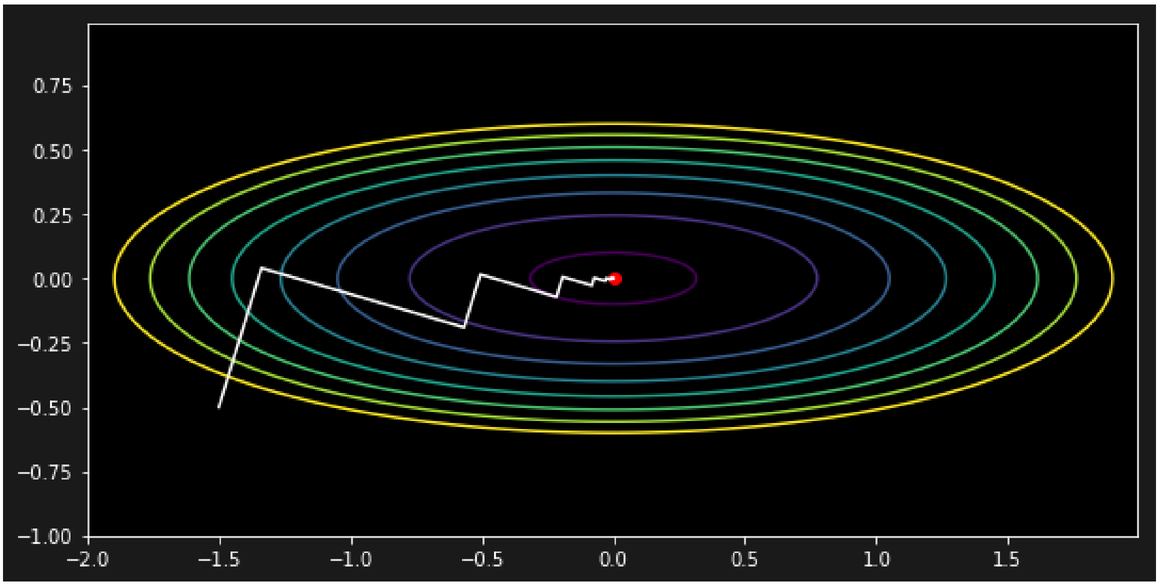
Метод наискорейшего спуска

В этом способе длина шага выбирается таким образом, чтобы движение по лучу заканчивалось в точке минимума функции.
$\operatorname{argmin}_x f(x) \in \{x|\forall y : f(y) \geq f(y)\}$ - аргумент минимизации
С использованием этой функции можно записать следующее условие:
$\alpha^{[i]} = \operatorname{argmin}_{x\in[0,\infty)}(f(\vec{x}^{[i]} - \lambda^{[i]} \bullet \nabla f(\vec{x}^{[i]})))$
Чтобы выбрать значение миниммального аргумента, нужно найти локальный экстремум функции на луче. Это можно сделать более эффективно, чем в данном примере - например, методом золотого сечения.

In [8]:

```
import pdb

def grad_descent_spd(grad, x, y, eps, f):
    """
        Градиентный спуск с выбором коэффициента для наискорейшего спуска
        grad - функция градиента
        x, y - начальная точка
        eps - точность
        f - функция
        Возвращаемые значения - путь, результат, особые точки (ни одной в данном
случае)
    """
    X = np.array([x, y])
    cnt = 0
    path = [X]
    while True:
        a_r = np.arange(0, 1, eps)
        f_r = [f(*(X - a * grad(*X))) for a in a_r]
        a = a_r[np.argmin(f_r)]
        X1 = X - a * grad(*X)
        path.append(X1)
        if (linalg.norm(grad(*X)) < eps
            or cnt > 1000):
            break
        cnt += 1
        X = X1
    return path, X1, None

path, res, pts = plot_grad_descent(grad_descent_spd, gradient, -1.5, -0.5, eps=
0.001, f=f)
print(f"Iterations: {len(path)}, precision: {linalg.norm(res)}")
```

```
Iterations: 21, precision: 9.789814634458175e-05
```



Как видно, в данном случае шаг выбирается действительно наиболее эффективным методом. Также направления шагов ортогональны друг другу. Однако решение задачи оптимизации отнимает существенную часть ресурсов. Также, для некоторых функций, метод наискорейшего спуска может быть не намного эффективнее, чем более простые методы

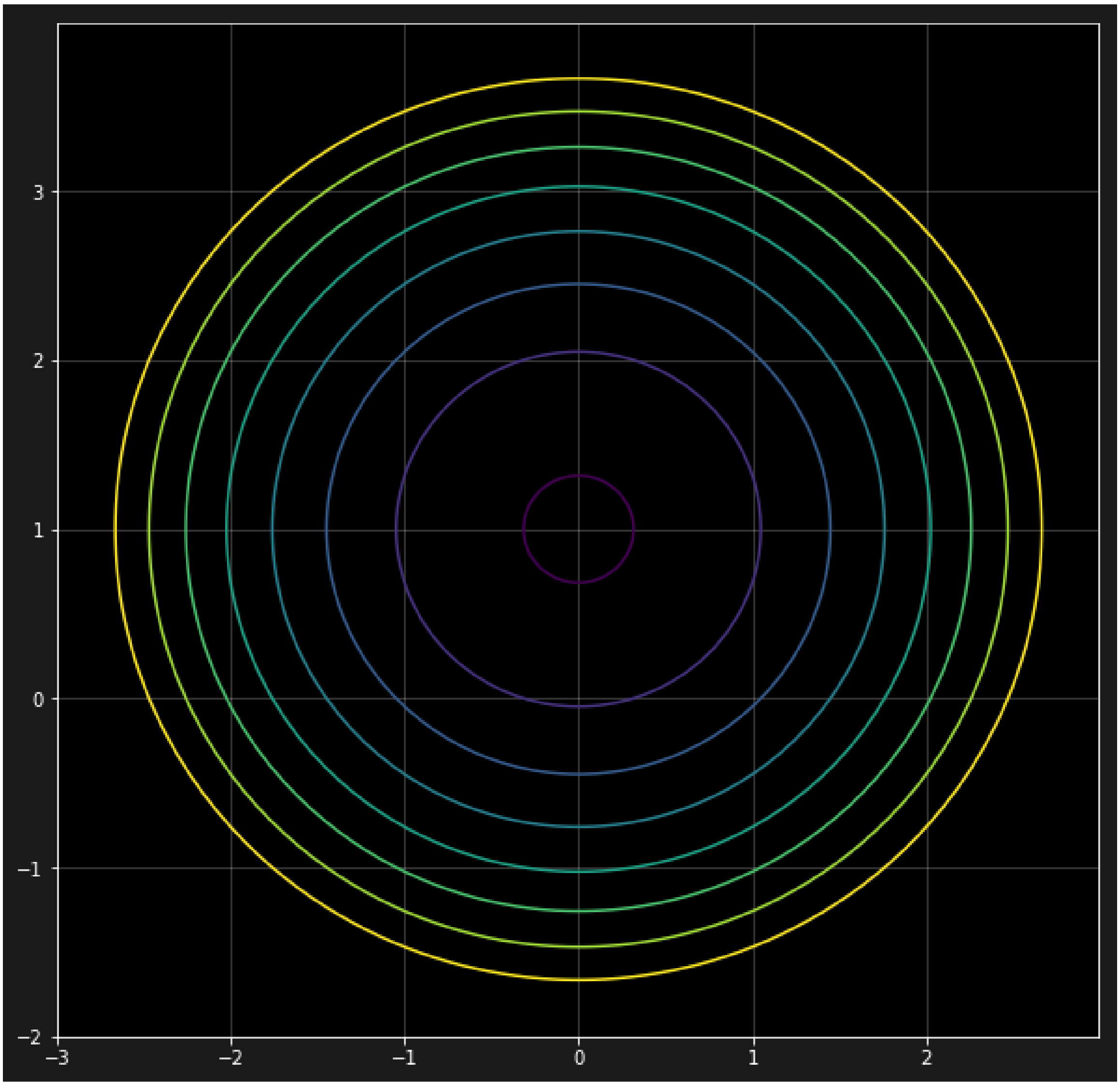
Метод Пауэлла

Решаемая задача - поиск локального минимума функции $f(x) : \mathbb{R}^n \to \mathbb{R}$

Рассмотрим следующую функцию: $f(x, y) = x^2 + (y-1)^2$.

In [1]:

```
from matplotlib import pyplot as plt
import numpy as np
%matplotlib inline
plt.style.use('dark_background')

f = lambda x, y : np.power(x, 2) + (y-1)**2

def plot_func(X, Y, Z):
    plt.figure(facecolor='0.1', figsize=(10, 10))
    plt.grid(True, alpha=0.25)
    plt.contour(X, Y, Z, levels=levels)

X = np.arange(-3, 3, 0.01)
Y = np.arange(-2, 4, 0.01)
levels = np.arange(0.1, 8, 1)
X, Y = np.meshgrid(X, Y)
Z = f(X, Y)

plot_func(X, Y, Z)
plt.show()
```



3
2
1
0
−1
−2
−3
−2
−1
0
1
2

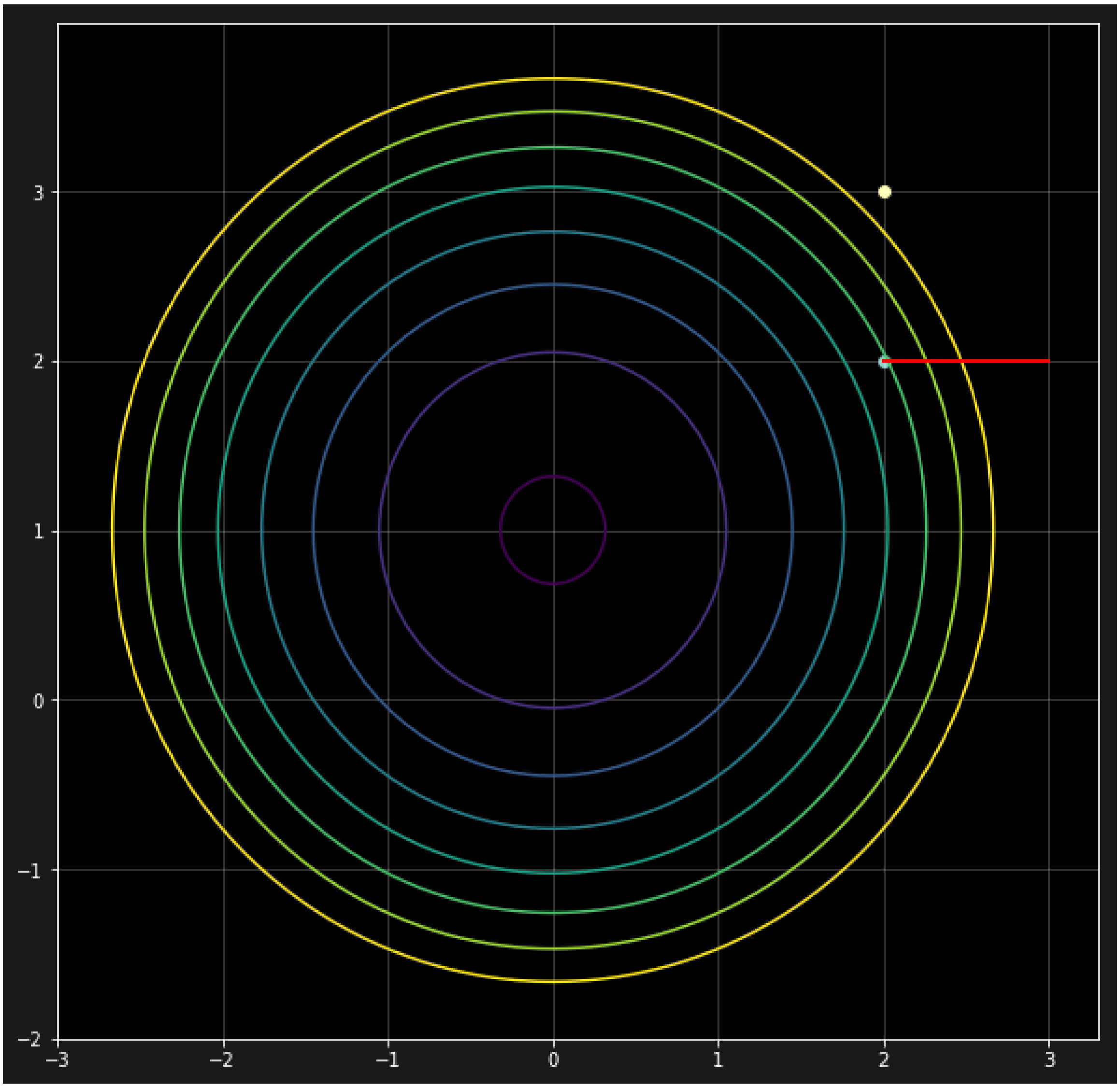
Метод Пауэлла

Выбор начальных точек и направления

$x_1 = (2, 2)$
$x_2 = (2, 3)$
$\Delta x = (1, 0)$

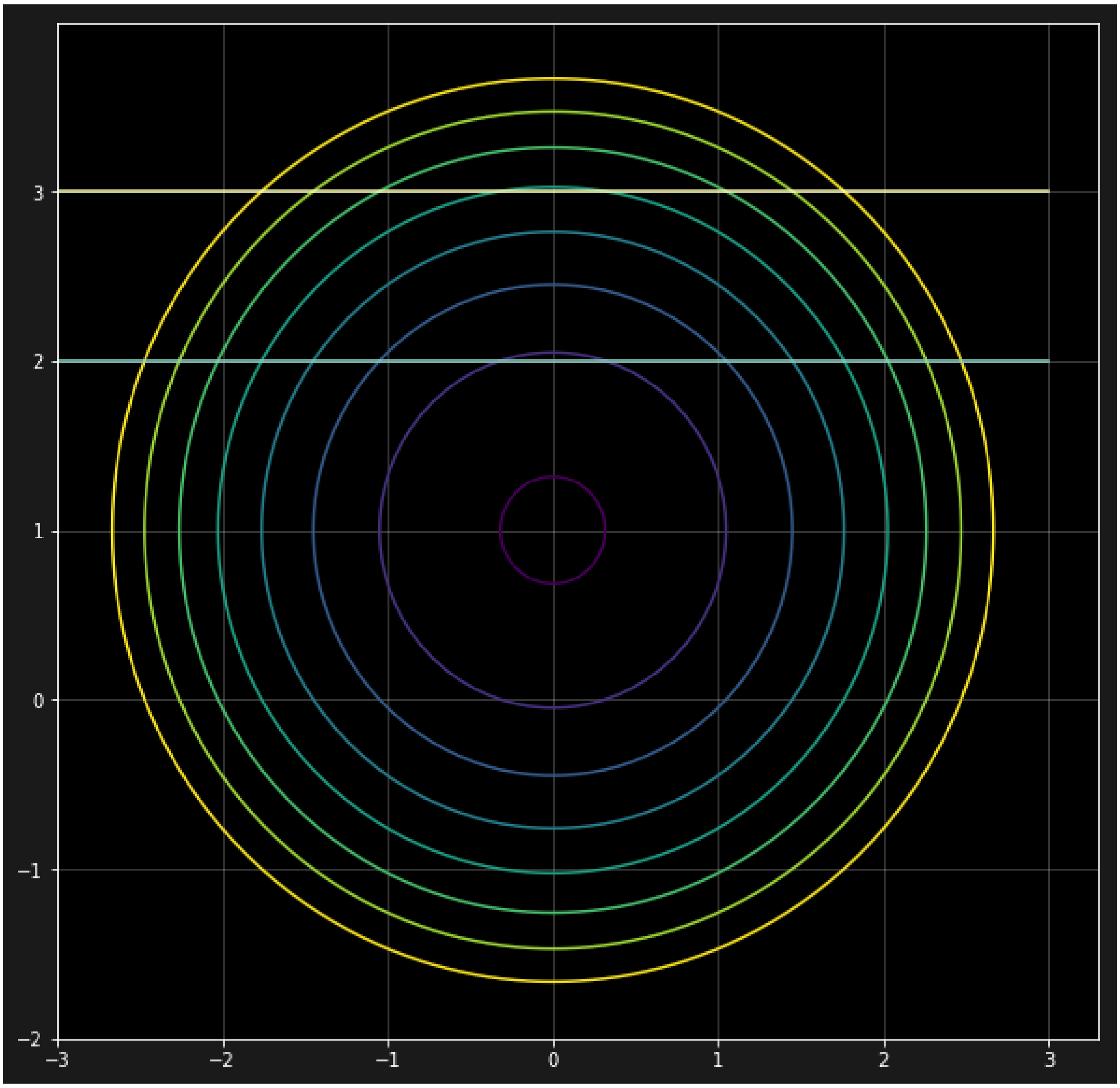
In [2]:

```
x_1 = np.array((2, 2))
x_2 = np.array((2, 3))
dx = np.array((1, 0))

plot_func(X, Y, Z)
plt.scatter(*x_1)
plt.scatter(*x_2)
vec = np.array([x_1, x_1 + dx])
plt.plot(vec[:, 0], vec[:, 1], linewidth=2, color='r')
pass
```



In [3]:

```
def func_mod(f, point, coef):
    return f(*(point * coef))

line1 = np.array([x_1 + a * dx for a in np.linspace(-5, 1, 1000)])
line2 = np.array([x_2 + a * dx for a in np.linspace(-5, 1, 1000)])
plot_func(X, Y, Z)
plt.plot(line1[:, 0], line1[:, 1])
plt.plot(line2[:, 0], line2[:, 1])
pass
```



In [4]:

```
def get_min_on_line(f, line):
    func_vals = np.array([f(*x) for x in line])
    min_ = np.argmin(func_vals)
    min_point = line[min_]
    return min_point

p1 = get_min_on_line(f, line1)
p2 = get_min_on_line(f, line2)
new_vec = p2 - p1
line3 = np.array([p1 + new_vec * c for c in np.linspace(-4, 2, 1000)])
p_min = get_min_on_line(f, line3)

plot_func(X, Y, Z)
plt.plot(line1[:, 0], line1[:, 1], linewidth=2, color='r', zorder=1)
plt.plot(line2[:, 0], line2[:, 1], linewidth=2, color='r', zorder=1)
plt.plot(line3[:, 0], line3[:, 1], linewidth=3, color='y', zorder=1)
plt.scatter(*x_1, zorder=2)
plt.scatter(*x_2, zorder=2)
plt.scatter(*p1, color='r', zorder=2)
plt.scatter(*p2, color='r', zorder=2)
plt.scatter(*p_min, color='r', zorder=2, s=80)
pass
```