

Исследование операций

Нелинейное программирование

Постановка и классификация задач. Условия оптимальности

Нелинейное программирование.

Постановка задачи

Пример: Построить прямоугольную картонную коробку заданного объема V^* с заданной площадью сторон S^*).

Варьируемые параметры: длины сторон x_1, x_2, x_3 .

Целевая функция: $F = (V - V^*)^2 = (x_1 x_2 x_3 - V^*)^2$

Ограничения: $S = 2(x_1 x_2 + x_1 x_3 + x_2 x_3) = S^*$, длины сторон неотрицательны

Математическая формулировка:

$$\begin{cases} \min F = (x_1 x_2 x_3 - V^*)^2 \\ 2(x_1 x_2 + x_1 x_3 + x_2 x_3) = S^* \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0 \end{cases}$$

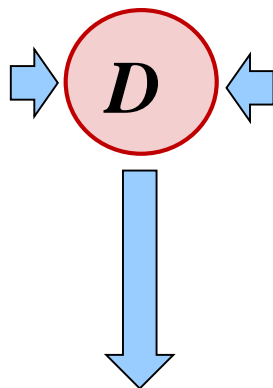


Нелинейное программирование.

Постановка задачи

$$\begin{cases} \min F(\vec{x}), & \vec{x} = (x_1, x_2, \dots, x_n) \\ \vec{x} \in D \end{cases}$$

$$\begin{aligned} & x_i \leq b_i^+; \\ & x_i \geq b_i^-; \\ & x_i = b_i \Leftrightarrow \begin{cases} x_i \leq b_i \\ x_i \geq b_i \end{cases} \\ & i = 1, 2, \dots, n \end{aligned}$$



$$\begin{aligned} & \phi_j(\vec{x}) \leq c_j^+; \\ & \phi_j(\vec{x}) \geq c_j^-; \\ & \phi_j(\vec{x}) = c_j \Leftrightarrow \begin{cases} \phi_j(\vec{x}) \leq c_j \\ \phi_j(\vec{x}) \geq c_j \end{cases} \\ & j = 1, 2, \dots, m \end{aligned}$$

$$\begin{cases} g_k(\vec{x}) \geq 0; \\ k = 1, 2, \dots, 2n + 2m \end{cases}$$

Нелинейное программирование.

Постановка задачи

$$\vec{g}_k = \vec{g}_k(\vec{x}) = \vec{g}_k(\vec{x}, \vec{\phi}(\vec{x}))$$

$$\phi_j(\vec{x}) \geq c_j^- \Rightarrow g_j = \phi_j - c_j^-; \quad g_j \geq 0$$

$$\phi_j(\vec{x}) \leq c_j^+ \Rightarrow g_{m+j} = c_j^+ - \phi_j; \quad g_{m+j} \geq 0$$

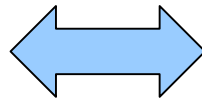
$$x_i \leq b_i^+ \Rightarrow g_{2m+i} = b_i^+ - x_i; \quad g_{2m+i} \geq 0$$

$$x_i \geq b_i^- \Rightarrow g_{2m+n+i} = x_i - b_i^-; \quad g_{2m+n+i} \geq 0$$

Нелинейное программирование. Постановка задачи



$$\begin{cases} \min F = (x_1 x_2 x_3 - V^*)^2, \\ 2(x_1 x_2 + x_1 x_3 + x_2 x_3) = S^*, \\ x_1 \geq 0, \\ x_2 \geq 0, \\ x_3 \geq 0 \end{cases}$$



$$\begin{cases} \min F = (x_1 x_2 x_3 - V^*)^2, \\ 2(x_1 x_2 + x_1 x_3 + x_2 x_3) - S^* \geq 0, \\ S^* - 2(x_1 x_2 + x_1 x_3 + x_2 x_3) \geq 0, \\ x_1 \geq 0, \\ x_2 \geq 0, \\ x_3 \geq 0 \end{cases}$$

Нелинейное программирование.

Постановка задачи

$$\begin{cases} \min F(\vec{x}) & \vec{x} = (x_1, x_2, \dots, x_n)^T; \\ \vec{g}(\vec{x}) \geq 0 & \vec{g} = (g_1, g_2, \dots, g_k, \dots, g_p)^T; \end{cases}$$

Здесь и далее используется обозначение

$$\vec{g} \geq 0 \Leftrightarrow \forall k = 1, 2, \dots, 2n + 2m : g_k(x) \geq 0.$$

Нелинейное программирование.

Классификация задач

- **По числу варьируемых параметров**
 - Задачи одномерной минимизации
 - Задачи многомерной минимизации
- **По наличию (отсутствию ограничений)**
 - Задачи условной минимизации
 - Задачи безусловной минимизации
- **В зависимости от количества экстремумов в области допустимых значений вектора варьируемых параметров**
 - Задачи минимизации одноэкстремальной (унимодальной) целевой функции
 - Задачи минимизации многоэкстремальной целевой функции
- **В зависимости от структуры целевой функции и ограничений**
 - Задачи квадратичного программирования, выпуклого программирования и т.д.

Нелинейное программирование.

Классификация задач

- Локальный и глобальный минимум в многоэкстремальных задачах
 - Локальный минимум:

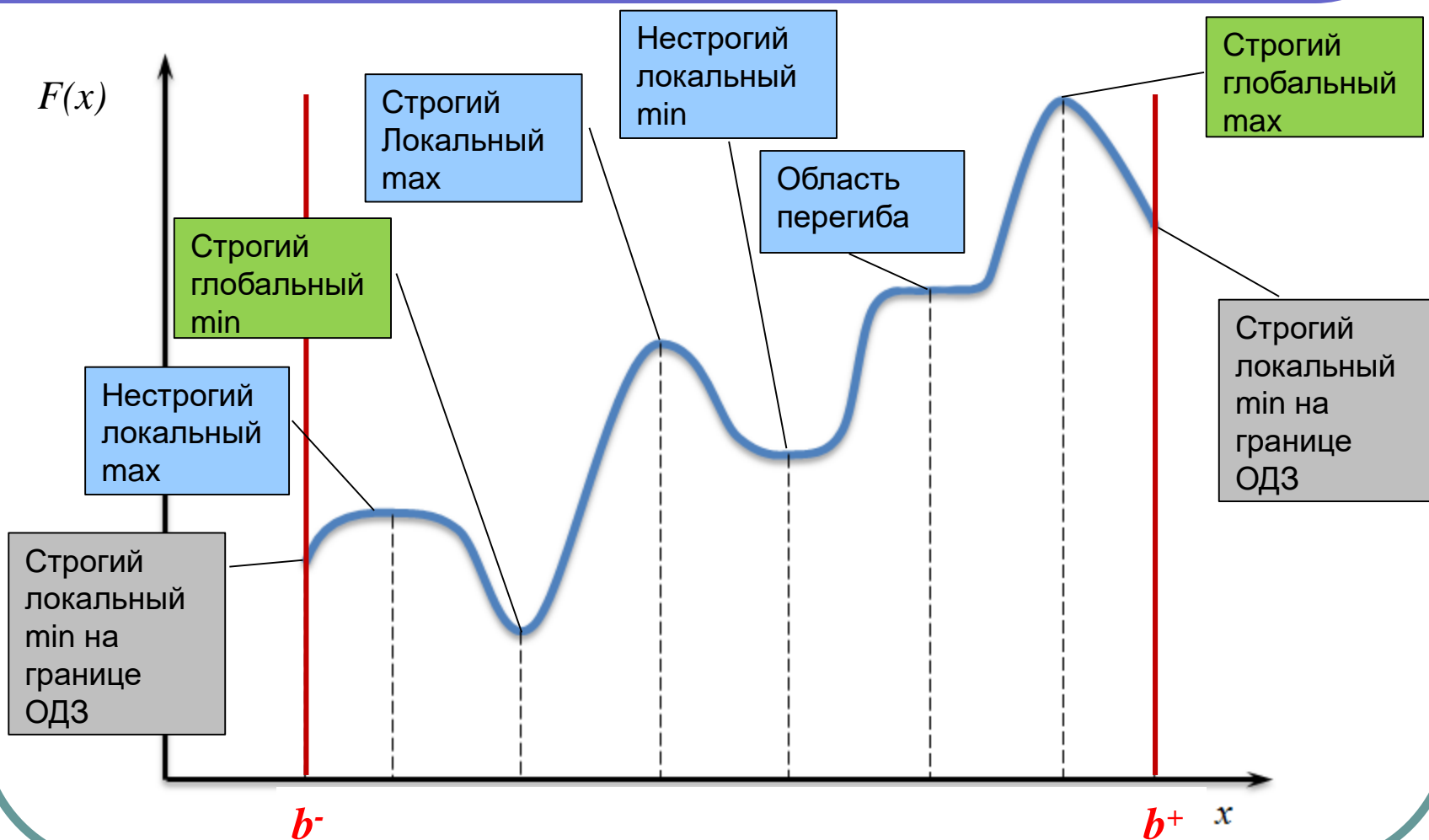
\vec{x}^* , если $\exists r > 0, F(\vec{x}^*) \leq F(\vec{x}) \quad \forall \vec{x} : |\vec{x} - \vec{x}^*| \leq r, x \in D$

- Глобальный минимум:

\vec{x}^* , если $F(\vec{x}^*) \leq F(\vec{x}) \quad \forall \vec{x} : \vec{x} \in D$

Нелинейное программирование.

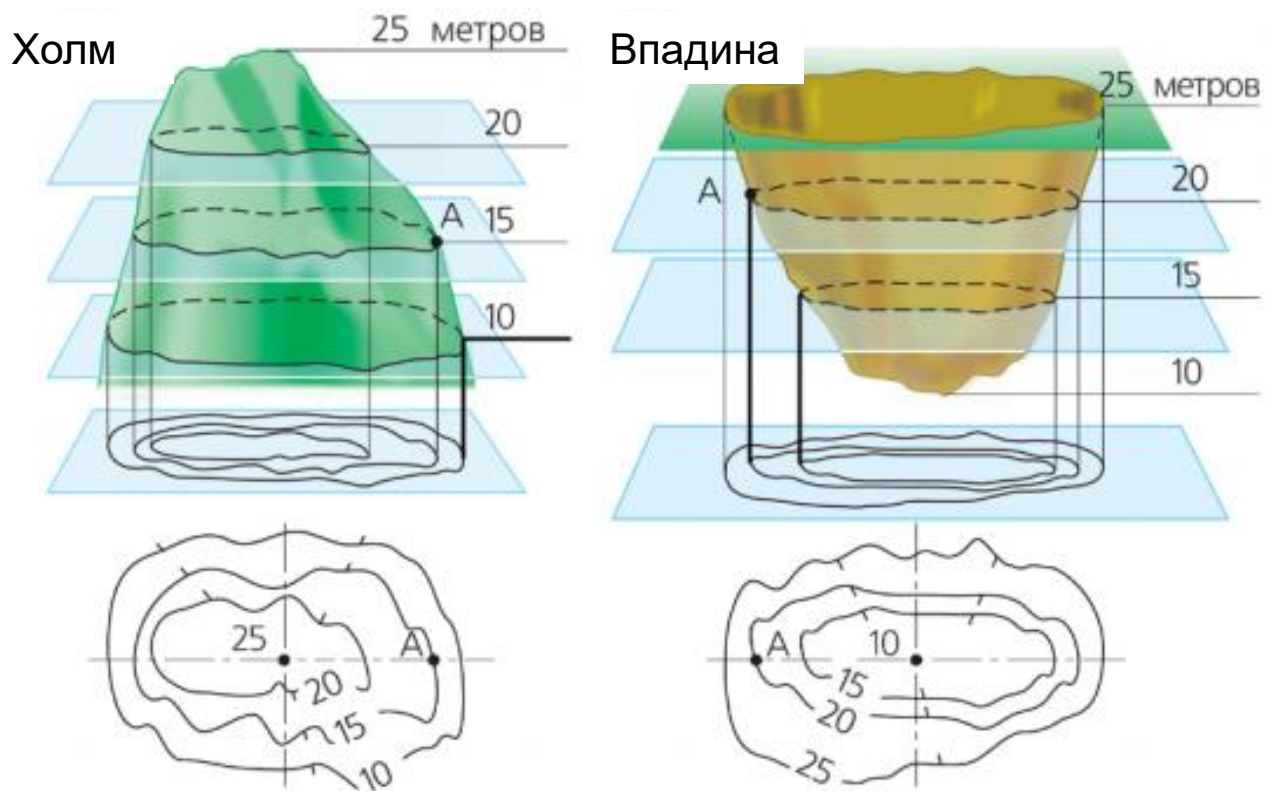
Классификация задач



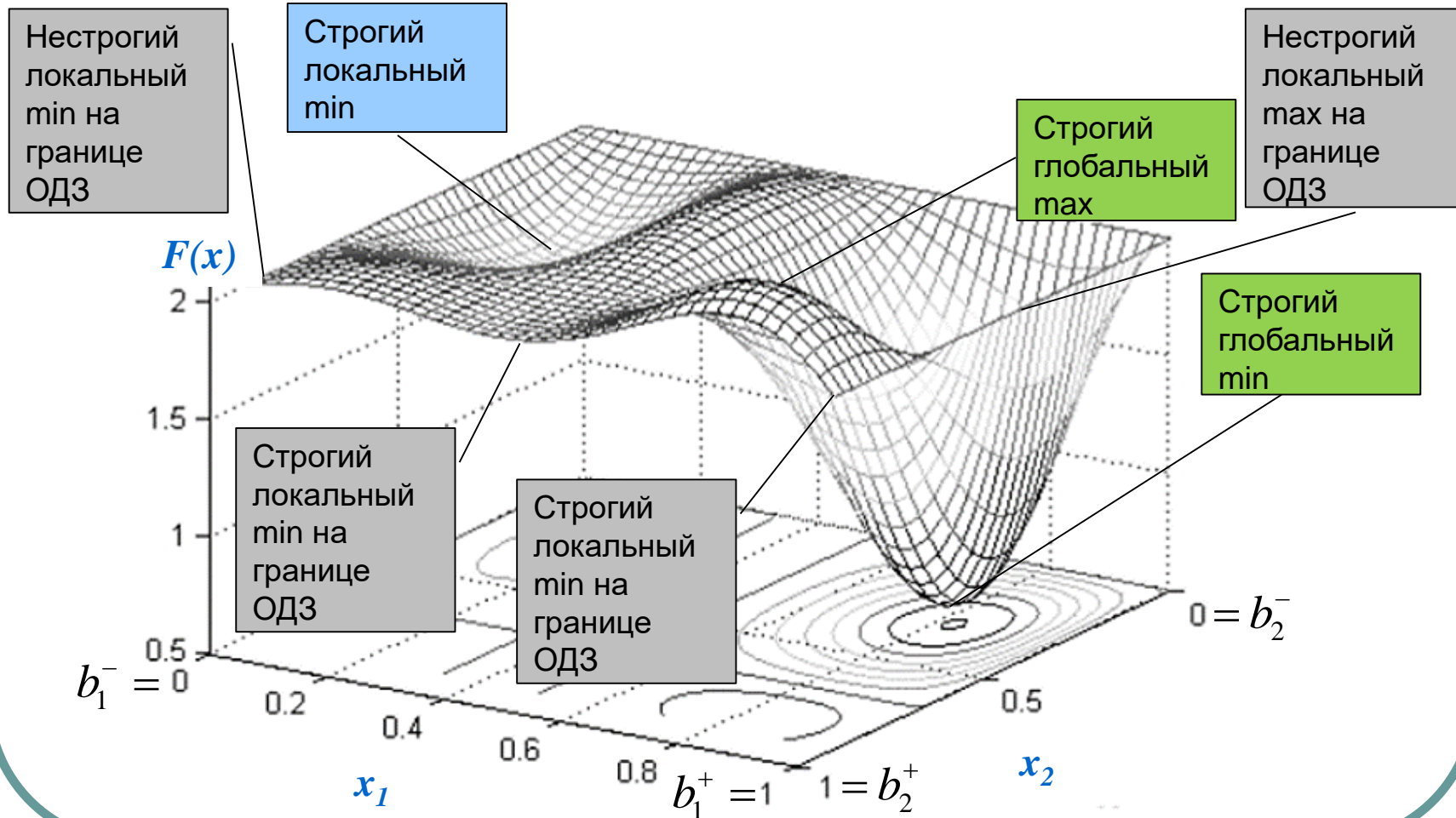
Нелинейное программирование.

Классификация задач

Линии постоянного уровня на топографической карте

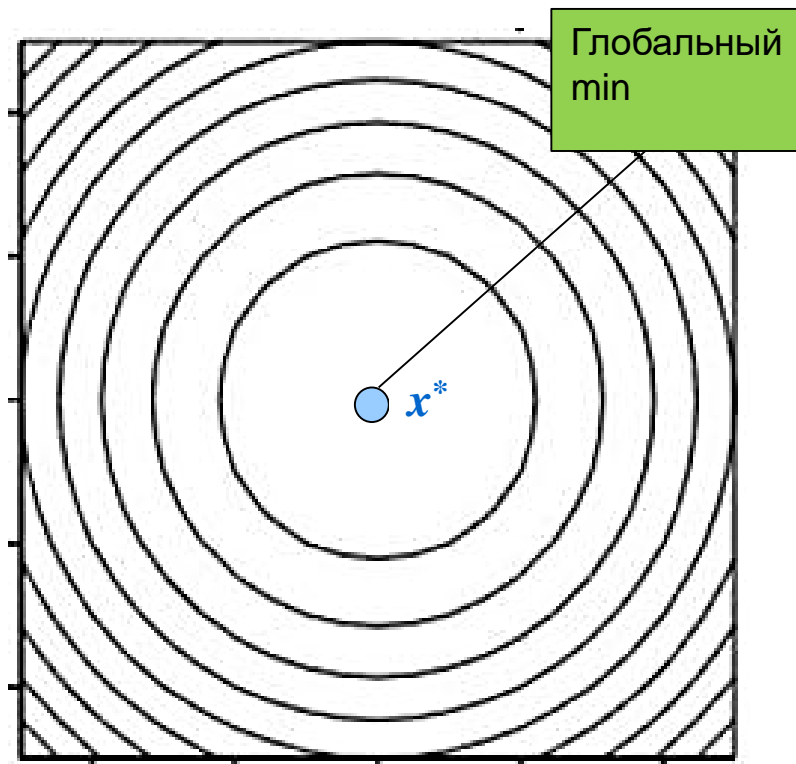
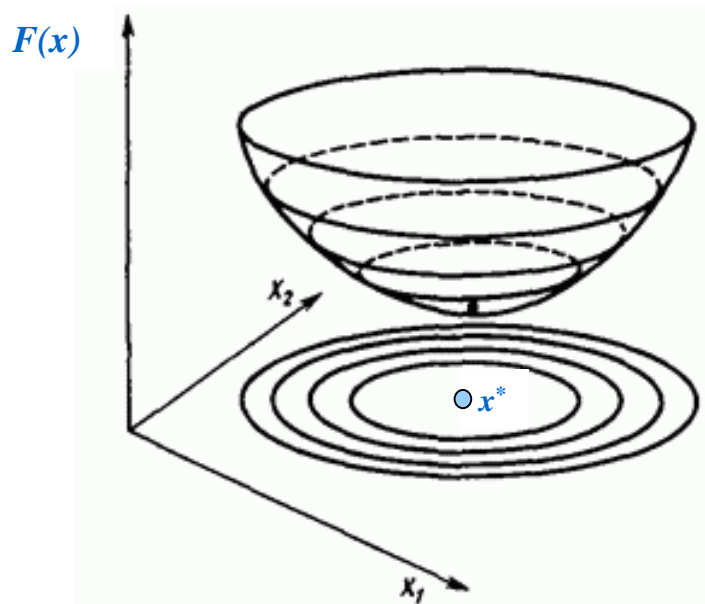


Нелинейное программирование. Классификация задач



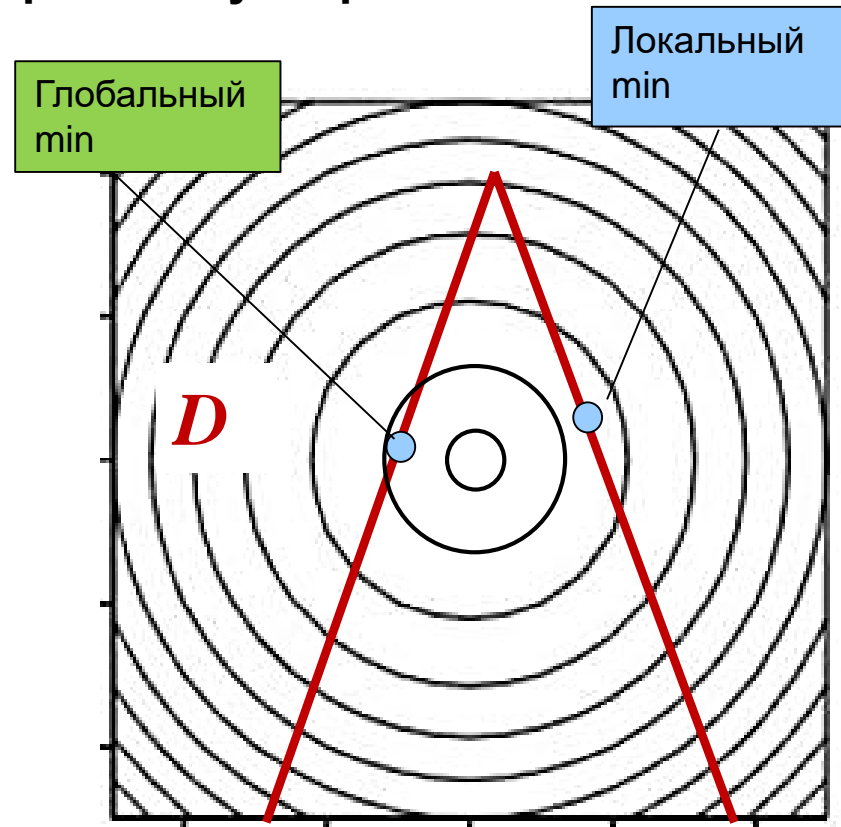
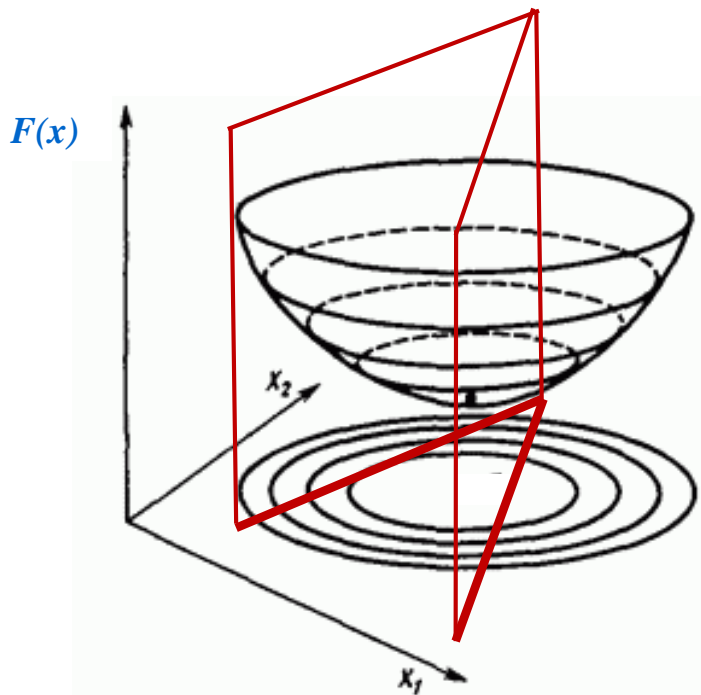
Нелинейное программирование. Классификация задач

Одноэкстремальная задача без ограничений...



Нелинейное программирование. Классификация задач

Одноэкстремальная задача без ограничений может превратиться в многоэкстремальную при их наличии



Нелинейное программирование.

Условия оптимальности для некоторых классов задач

- Безусловная минимизация функций одной переменной

$$\begin{array}{l} \min F(x); \\ x \in R \end{array} \quad \left\{ \begin{array}{l} \frac{dF(x)}{dx} = 0 \\ \frac{d^2 F(x)}{dx^2} > 0 \end{array} \right.$$

Нелинейное программирование.

Условия оптимальности для некоторых классов задач

- Безусловная минимизация функций n переменных

$$\begin{array}{l} \min F(\vec{x}); \\ \vec{x} \in R^n \end{array} \quad \left\{ \begin{array}{l} \frac{\partial F(\vec{x})}{\partial x_i} = 0 \\ H_{ij} = \frac{\partial^2 F(\vec{x})}{\partial x_i \partial x_j} \text{ П.О.} \end{array} \right.$$

Нелинейное программирование.

Условия оптимальности для некоторых классов задач

- Условная минимизация функций n переменных: условия Куна-Таккера

$$L = F(\vec{x}) + \sum_{j=1}^m \lambda_j g_j(\vec{x})$$

$$\begin{array}{l} \min F(\vec{x}); \\ g(\vec{x}) \geq 0 \end{array} \quad \longleftrightarrow \quad \left\{ \begin{array}{l} \frac{\partial L(\vec{x})}{\partial x_i} = 0; \\ \lambda_j g_j = 0; \\ g_j \geq 0; \\ \lambda_j \leq 0; \end{array} \right. \quad \begin{array}{l} i = 1, 2, \dots, n; \\ j = 1, 2, \dots, m; \end{array}$$

Нелинейное программирование.

Условия оптимальности для некоторых классов задач

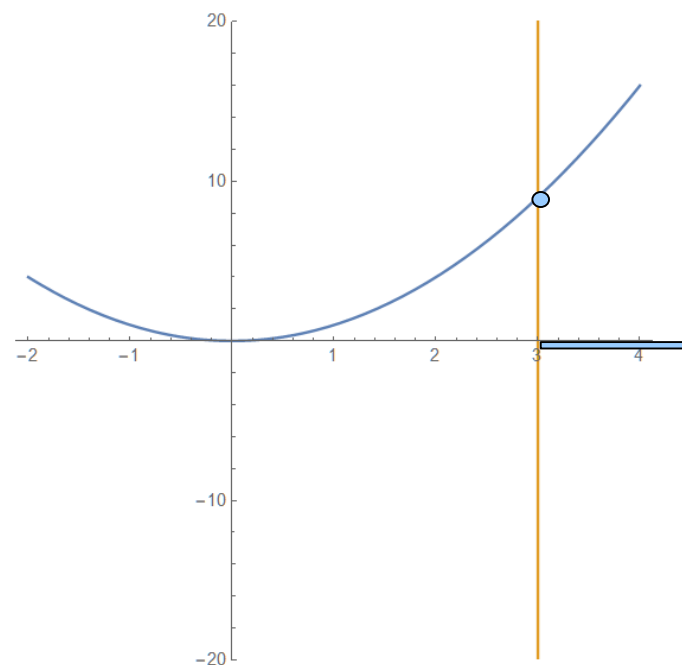
- Рассмотрим совсем простую задачу:

$$\begin{cases} \min F(x) = x^2; \\ x \geq 3. \end{cases} \iff \begin{cases} \min F(x) = x^2; \\ x - 3 \geq 0. \end{cases} \quad L = F(\vec{x}) + \sum_{j=1}^m \lambda_j g_j(\vec{x}) = x^2 + \lambda(x - 3)$$

$$\begin{cases} \frac{\partial L(\vec{x})}{\partial x_i} = 0; \\ \lambda_j g_j = 0; \\ g_j \geq 0; \\ \lambda_j \leq 0; \end{cases} \Rightarrow \begin{cases} 2x + \lambda = 0; \\ \lambda(x - 3) = 0; \\ \underline{x - 3 \geq 0}; \\ \lambda \leq 0. \end{cases}$$

$$\Rightarrow \begin{cases} x = 3; \\ \lambda = -6. \end{cases}$$

~~$\begin{cases} \lambda = 0; \\ x = 0. \end{cases}$~~



Нелинейное программирование.

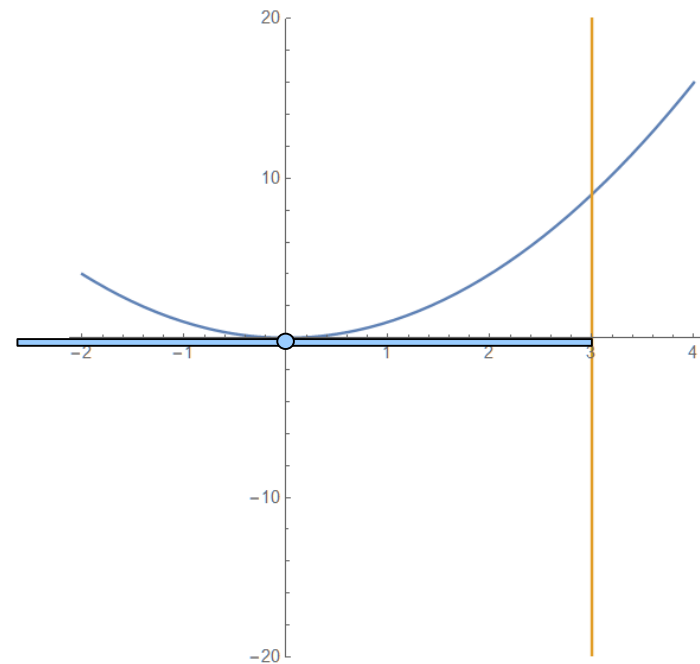
Условия оптимальности для некоторых классов задач

- Еще одна совсем простая задача:

$$\begin{cases} \min F(x) = x^2; \\ x \leq 3. \end{cases} \iff \begin{cases} \min F(x) = x^2; \\ 3 - x \geq 0. \end{cases} \quad L = F(\vec{x}) + \sum_{j=1}^m \lambda_j g_j(\vec{x}) = x^2 + \lambda(3 - x)$$

$$\begin{cases} \frac{\partial L(\vec{x})}{\partial x_i} = 0; \\ \lambda_j g_j = 0; \\ g_j \geq 0; \\ \lambda_j \leq 0; \end{cases} \Rightarrow \begin{cases} 2x - \lambda = 0; \\ \lambda(3 - x) = 0; \\ 3 - x \geq 0; \\ \underline{\lambda \leq 0.} \end{cases}$$

$$\Rightarrow \begin{cases} \cancel{x = 3;} \\ \cancel{\lambda = 6.} \\ \lambda = 0; \\ x = 0. \end{cases}$$



Безусловная минимизация функций одной переменной

Нелинейное программирование.

Методы
последовательного
сужения интервала
неопределенности

Метод
дихотомии

Метод золотого
сечения

Метод
Фибоначчи

Методы
аппроксимации

Метод Пауэлла
(квадратичной
аппроксимации)

Градиентные
методы

Метод Ньютона

Метод секущих

Метод средней
точки

Нелинейное программирование.

Безусловная минимизация функций одной переменной.

Метод Ньютона

$$\begin{cases} \min F(x); \\ x \in R \end{cases} \Rightarrow \frac{dF(x)}{dx} \equiv \Phi(x) = 0$$

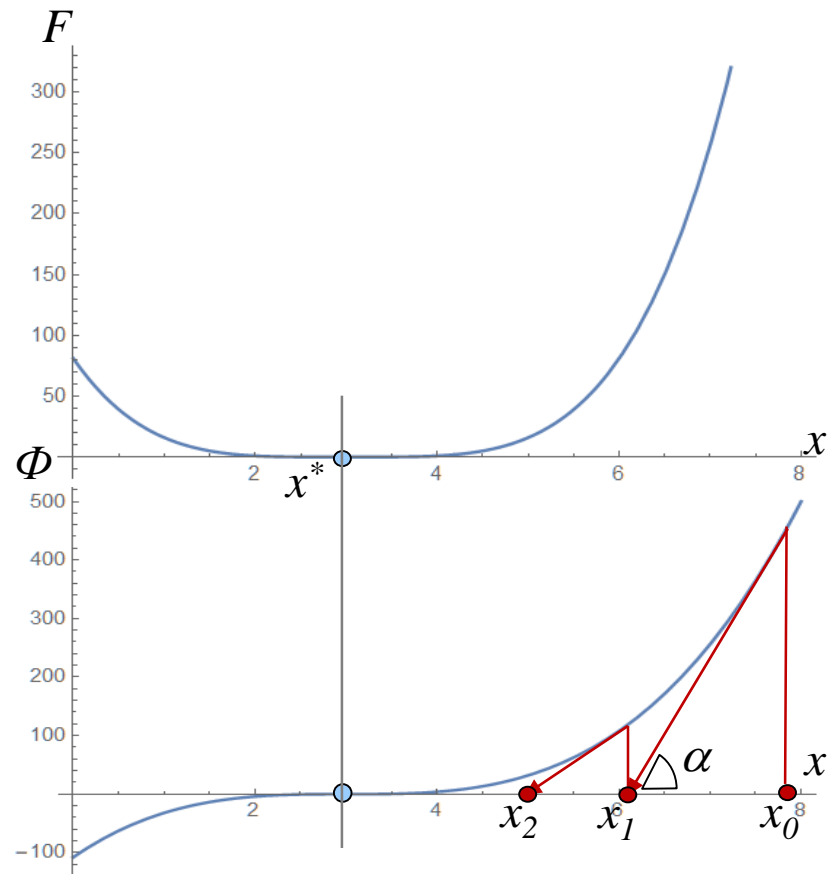
$$\operatorname{tg} \alpha = \frac{d\Phi(x_k)}{dx} = \frac{\Phi(x_k)}{x_k - x_{k+1}} \Rightarrow$$

$$x_{k+1} = x_k - \frac{\Phi(x_k)}{\Phi'(x_k)}$$

Критерии остановки:

$$|x_{k+1} - x_k| < \varepsilon$$

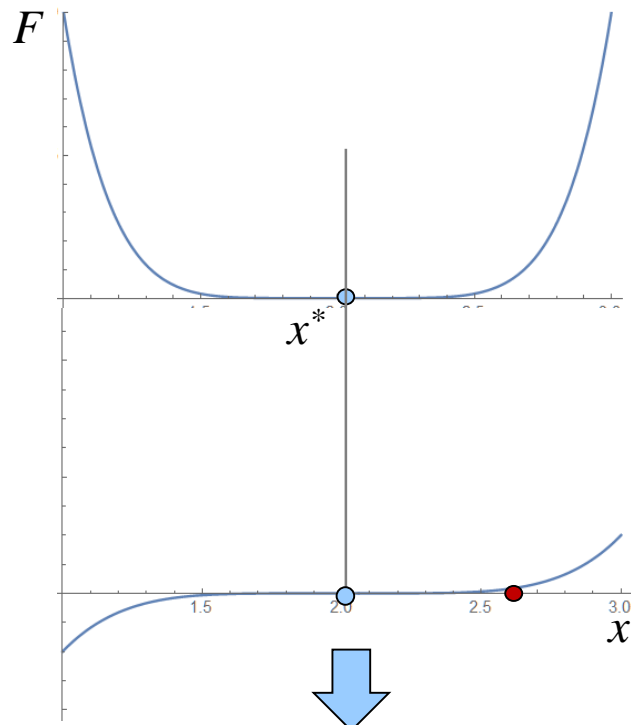
$$|\Phi(x_{k+1})| < \varepsilon$$



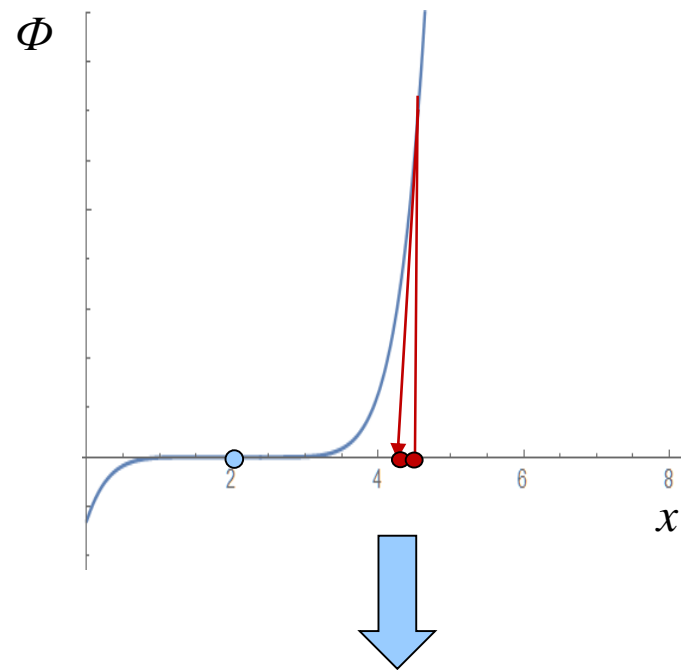
Нелинейное программирование.

Безусловная минимизация функций одной переменной.

Метод Ньютона



$$|\Phi(x_{k+1})| < \varepsilon$$



$$|x_{k+1} - x_k| < \varepsilon$$

Нелинейное программирование.

Безусловная минимизация функций одной переменной.

Метод Ньютона

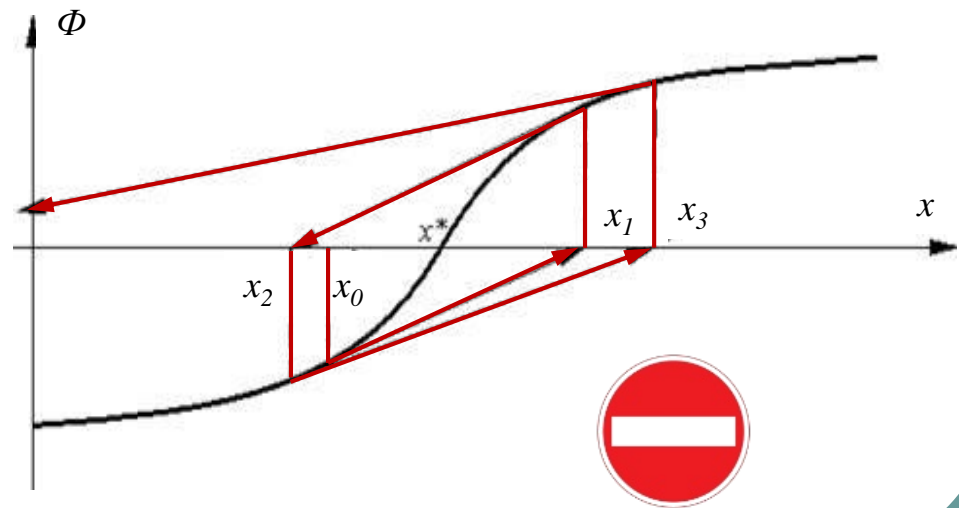
Ограничения:

- Первая производная должна быть ограничена во всей области поиска решения
- Вторая производная целевой функции должна существовать и быть непрерывной во всей области, в которой ведется поиск, включая точку минимума.
- Начальное приближение должно быть достаточно близко к точке корня

$$\left| x^* - x_k \right| \leq \frac{M}{2m} (x_k - x_{k+1})^2$$

$$M = \max_{x \in [a, b]} |\Phi'(x)|$$

$$m_1 = \min_{x \in [a, b]} |\Phi'(x)|$$



Нелинейное программирование.
Безусловная минимизация функций одной переменной.
Метод Ньютона

Нелинейное программирование.
Безусловная минимизация функций одной переменной.
Метод Ньютона

Алгоритм метода Ньютона

1. *Выбрать x_0 , задать $\varepsilon, \delta > 0$, положить $k = 0$*

3. *Положить $\Delta := 2\delta$*

4. *Пока $|F'(x_k)| > \varepsilon, \Delta > \delta$ выполнить*

$$4.1 \quad x_{k+1} := x_k - \frac{F'(x_k)}{F''(x_k)};$$

$$4.2 \quad \Delta := |x_{k+1} - x_k|$$

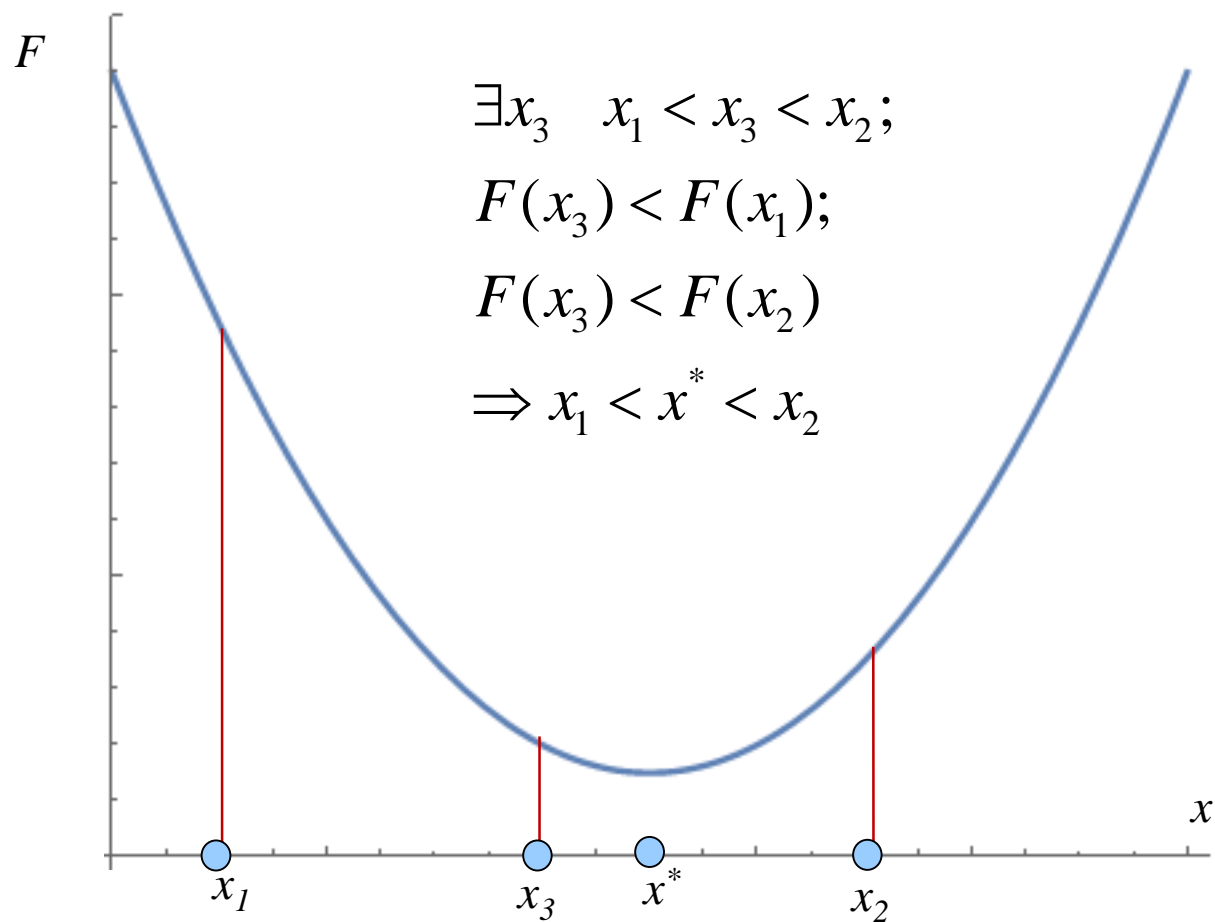
5. *Положить $x^* = x_{k+1}$.*

6. *Завершить работу.*

Нелинейное программирование.

Безусловная минимизация функций одной переменной.

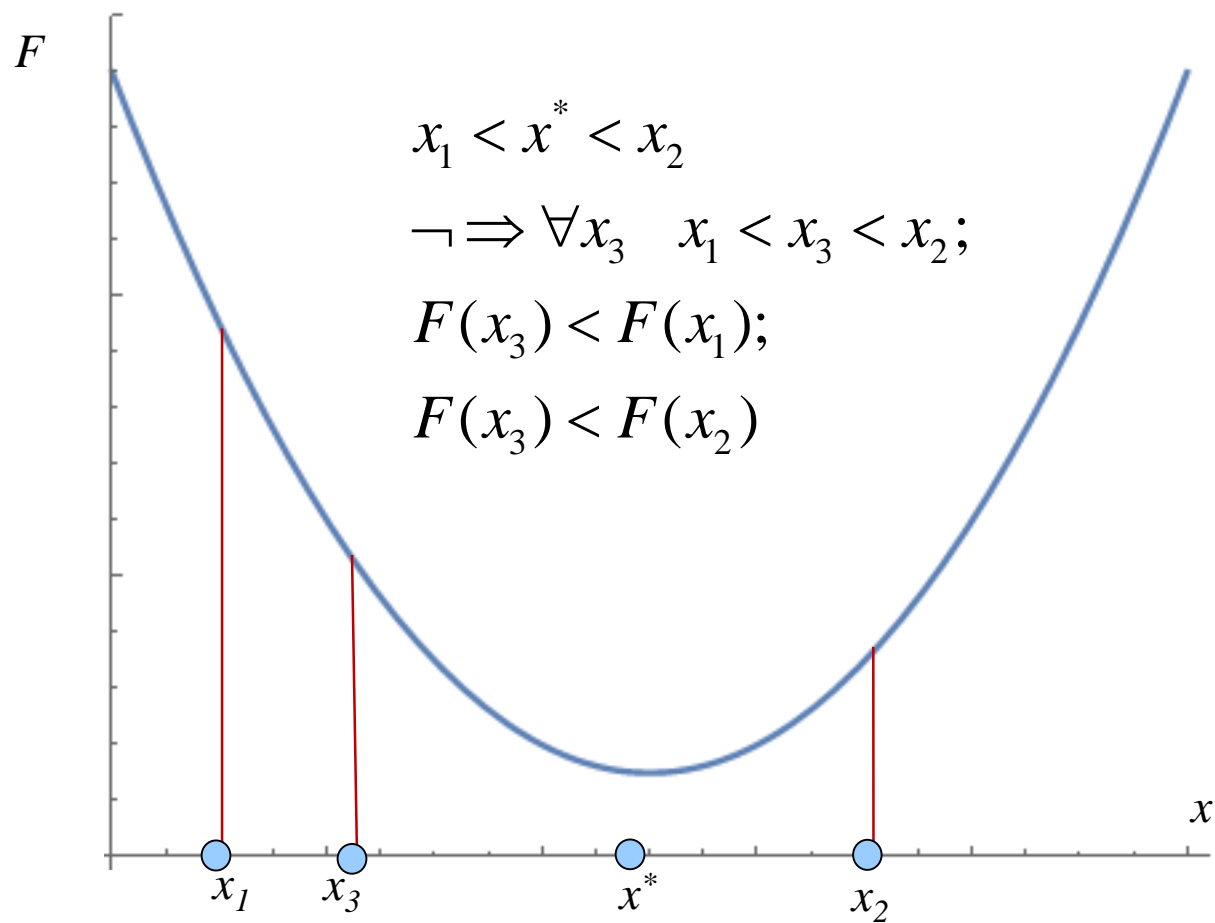
Метод золотого сечения



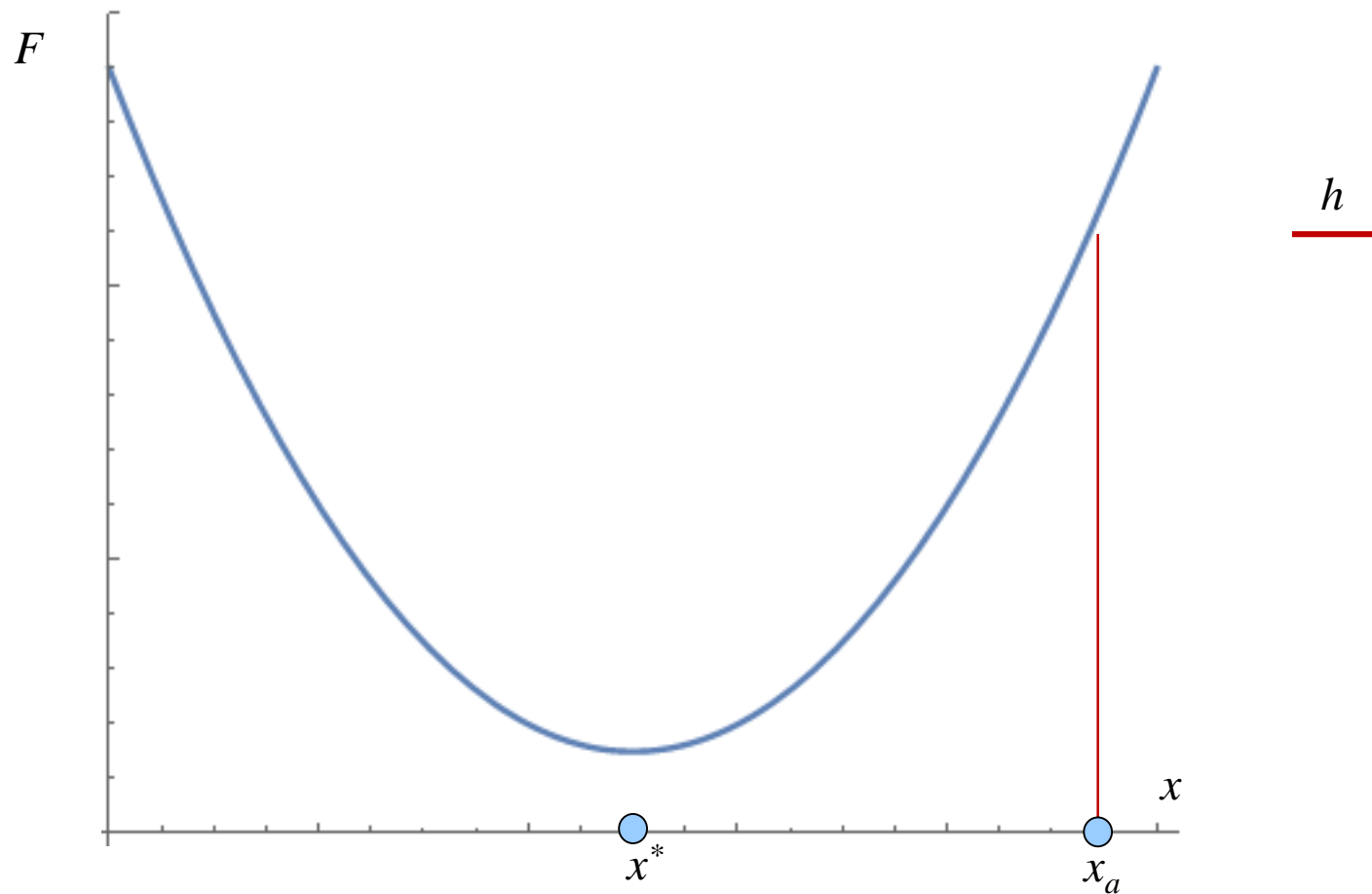
Нелинейное программирование.

Безусловная минимизация функций одной переменной.

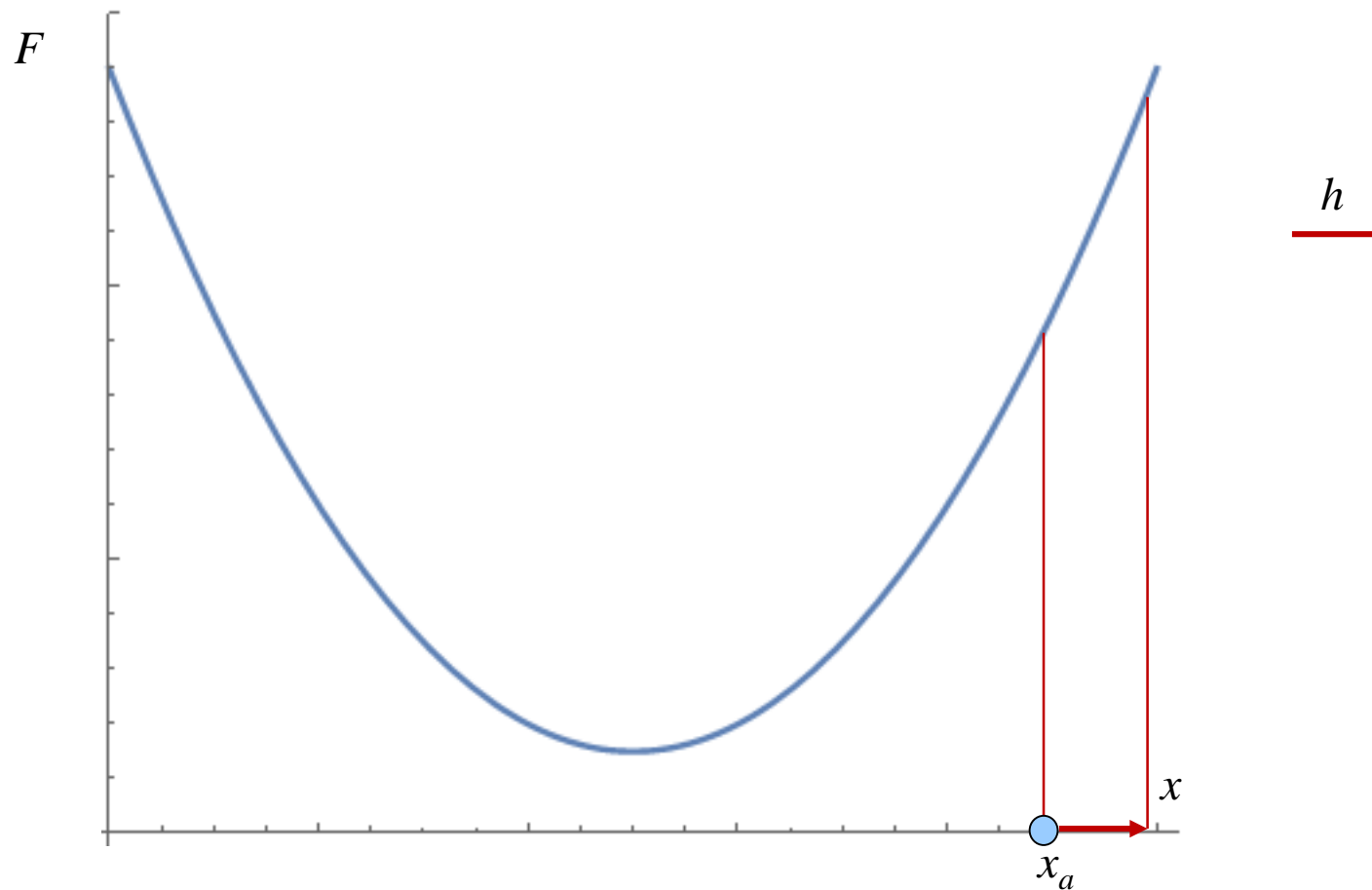
Метод золотого сечения



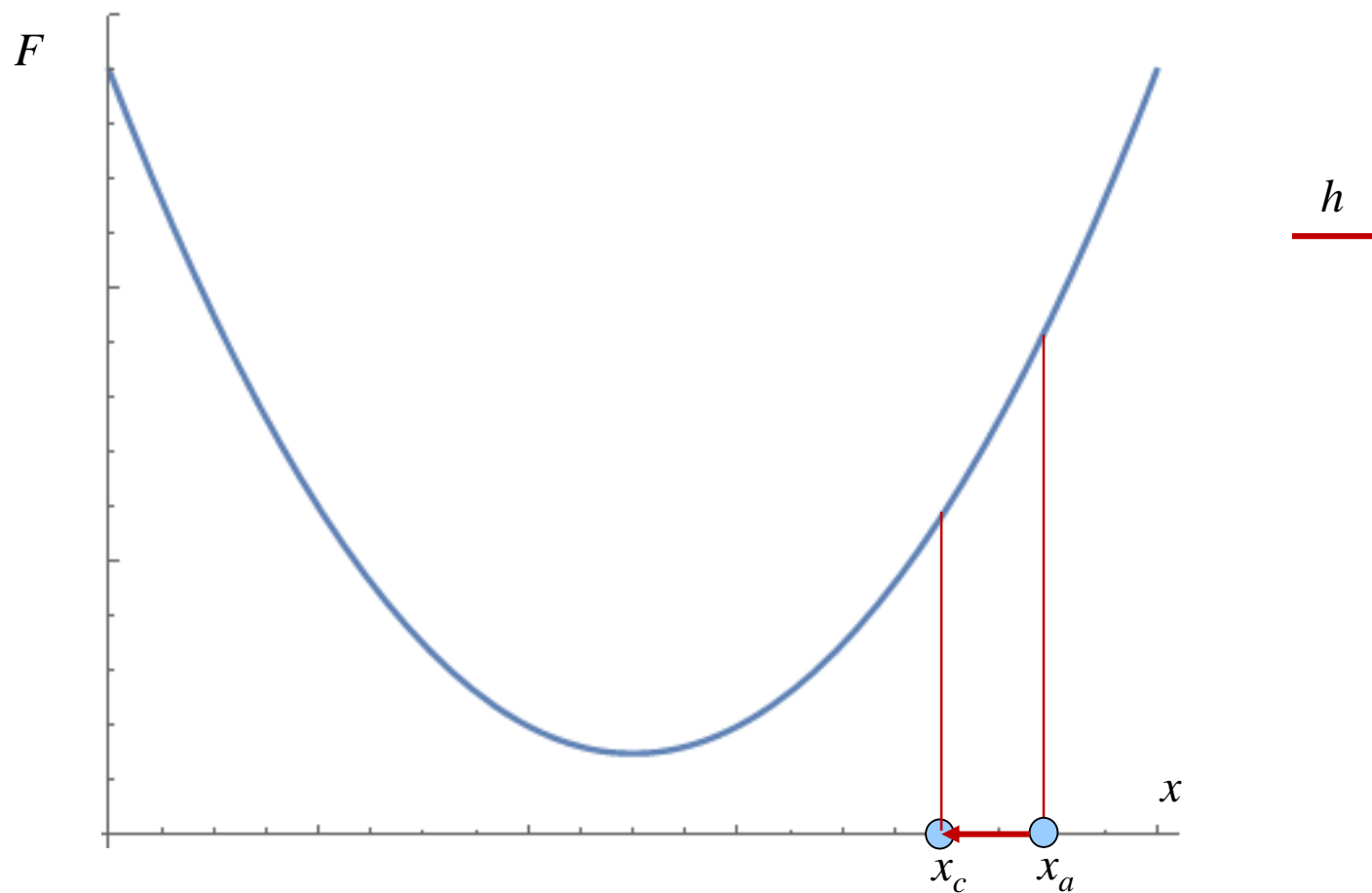
Нелинейное программирование.
Безусловная минимизация функций одной переменной.
Метод золотого сечения.



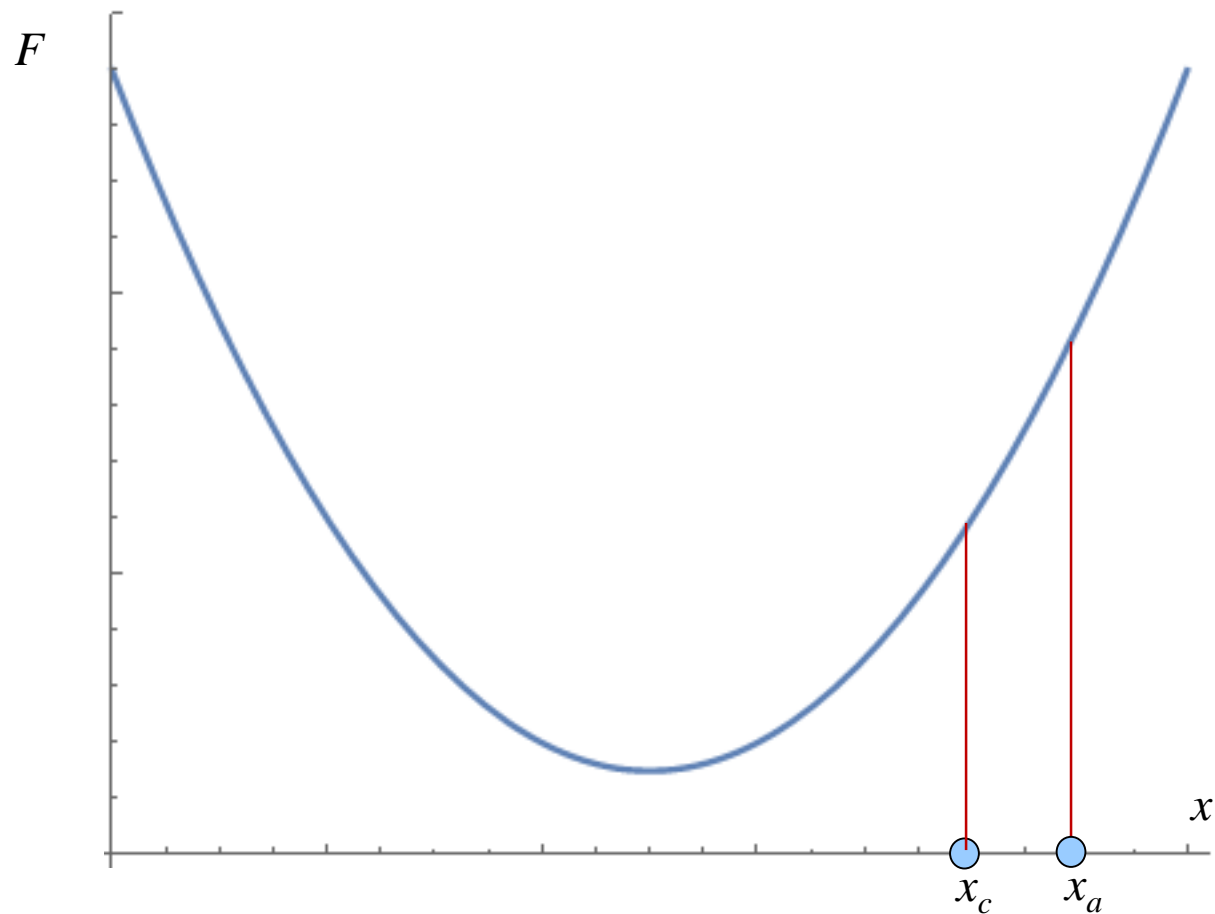
Нелинейное программирование.
Безусловная минимизация функций одной переменной.
Метод золотого сечения.



Нелинейное программирование.
Безусловная минимизация функций одной переменной.
Метод золотого сечения.

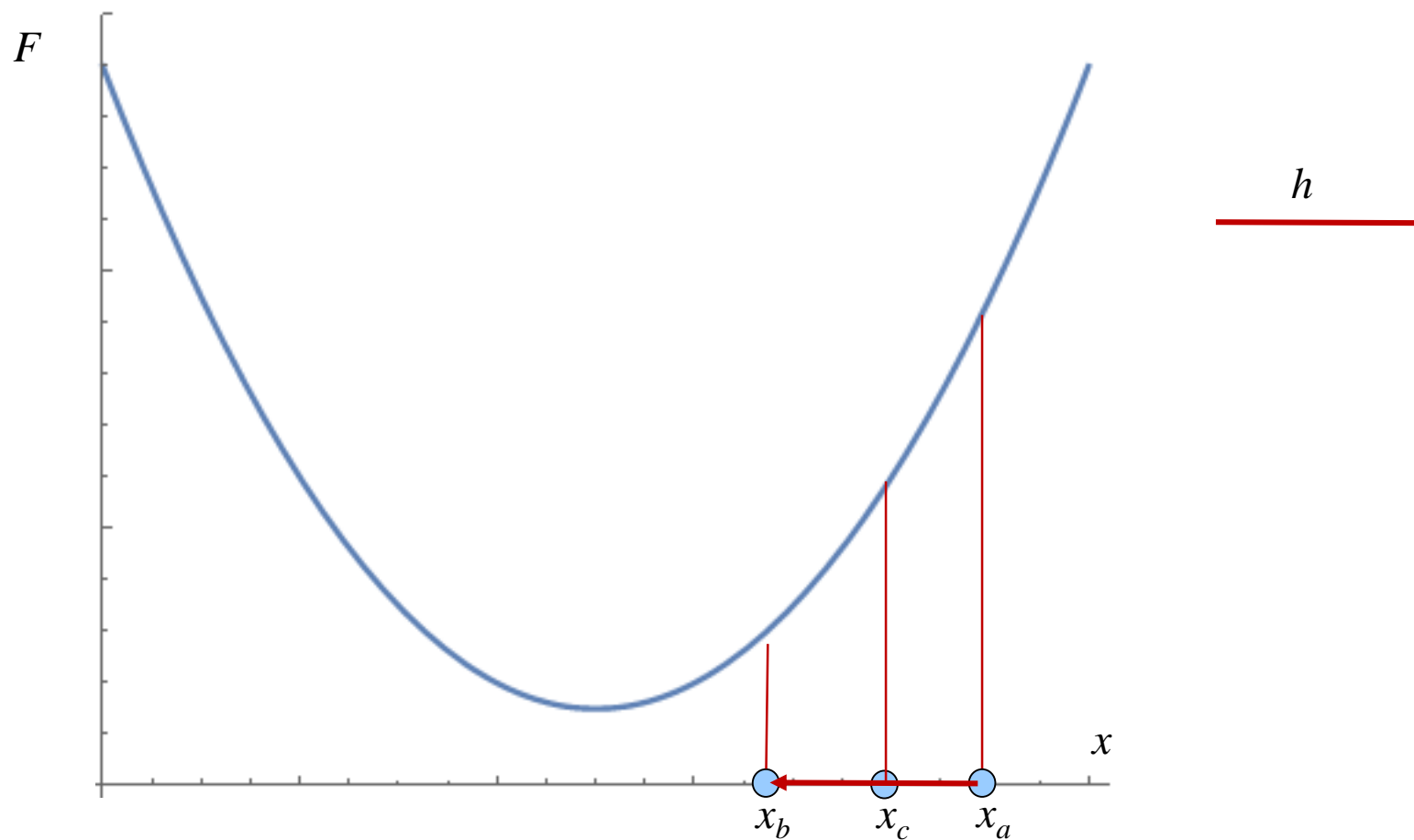


Нелинейное программирование.
Безусловная минимизация функций одной переменной.
Метод золотого сечения.

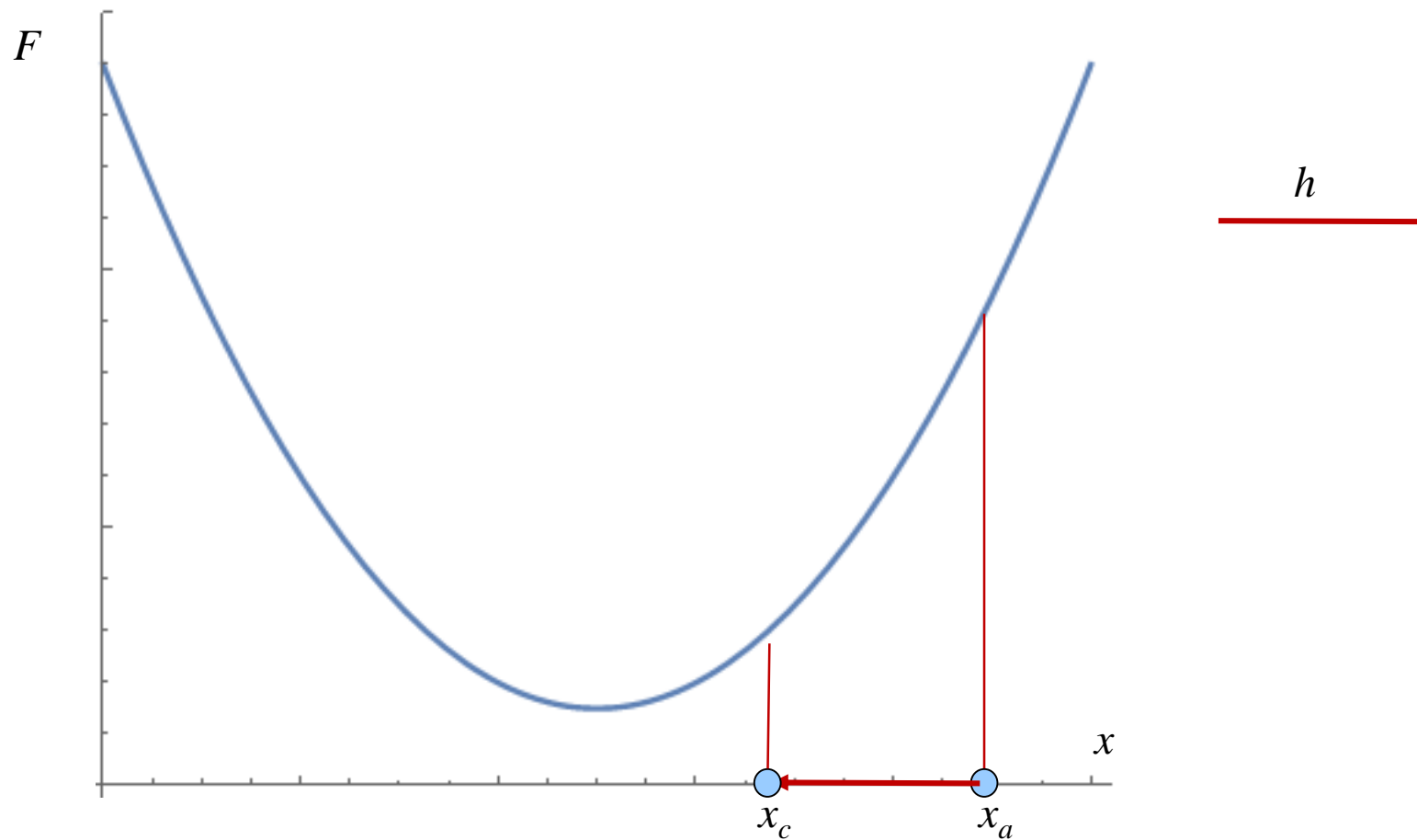


$$h = \alpha h$$

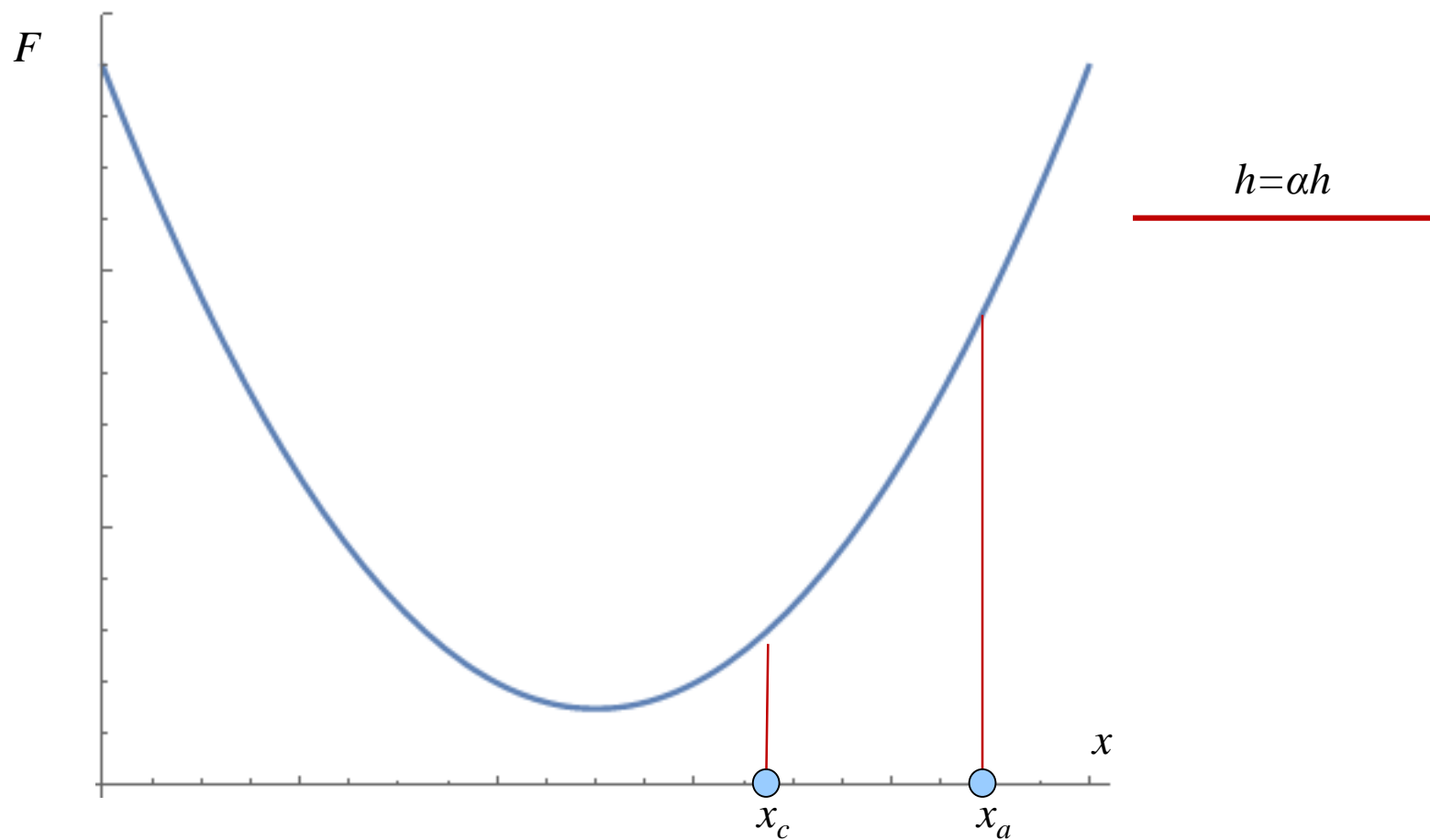
Нелинейное программирование.
Безусловная минимизация функций одной переменной.
Метод золотого сечения.



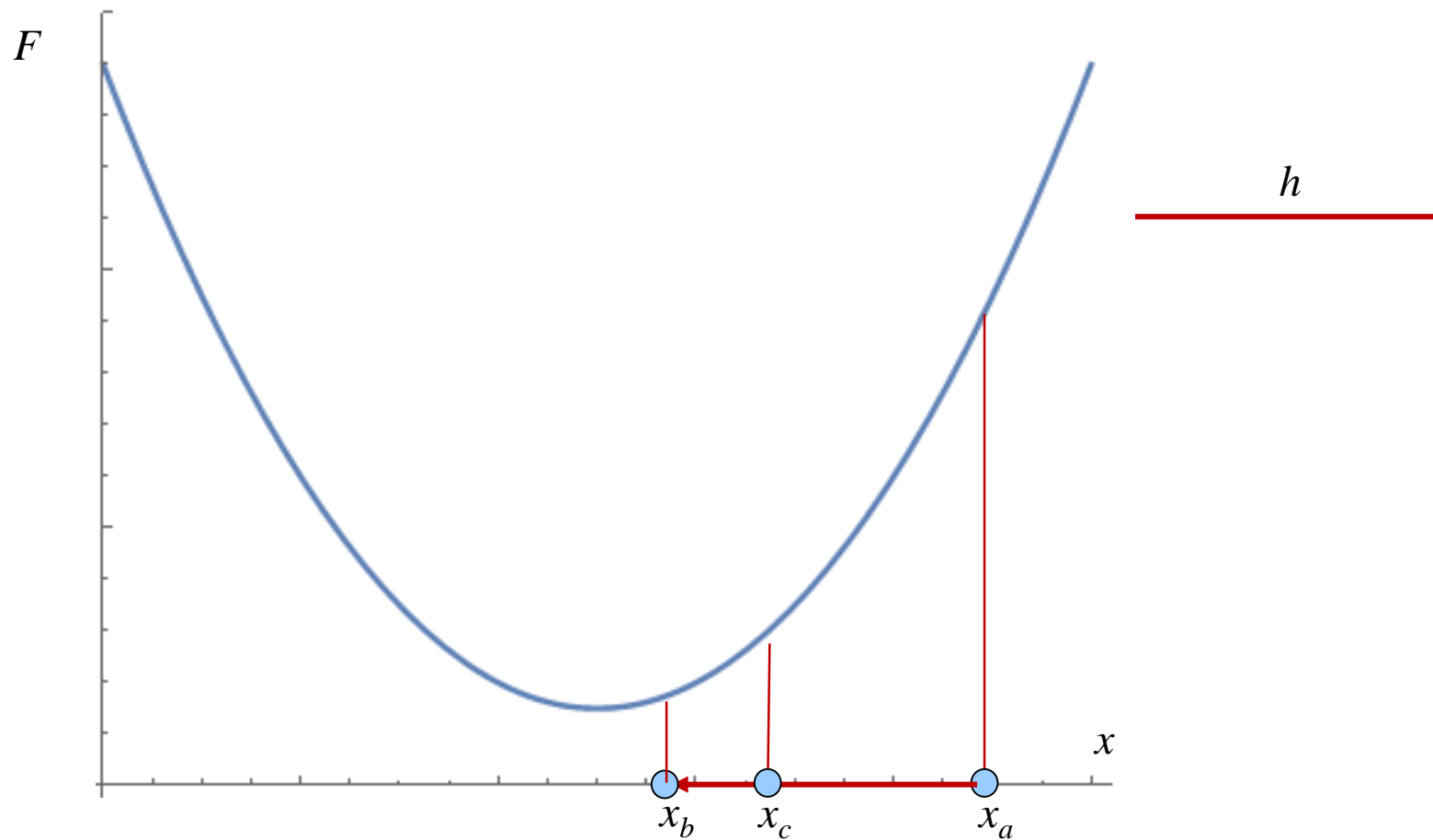
Нелинейное программирование.
Безусловная минимизация функций одной переменной.
Метод золотого сечения.



Нелинейное программирование.
Безусловная минимизация функций одной переменной.
Метод золотого сечения.



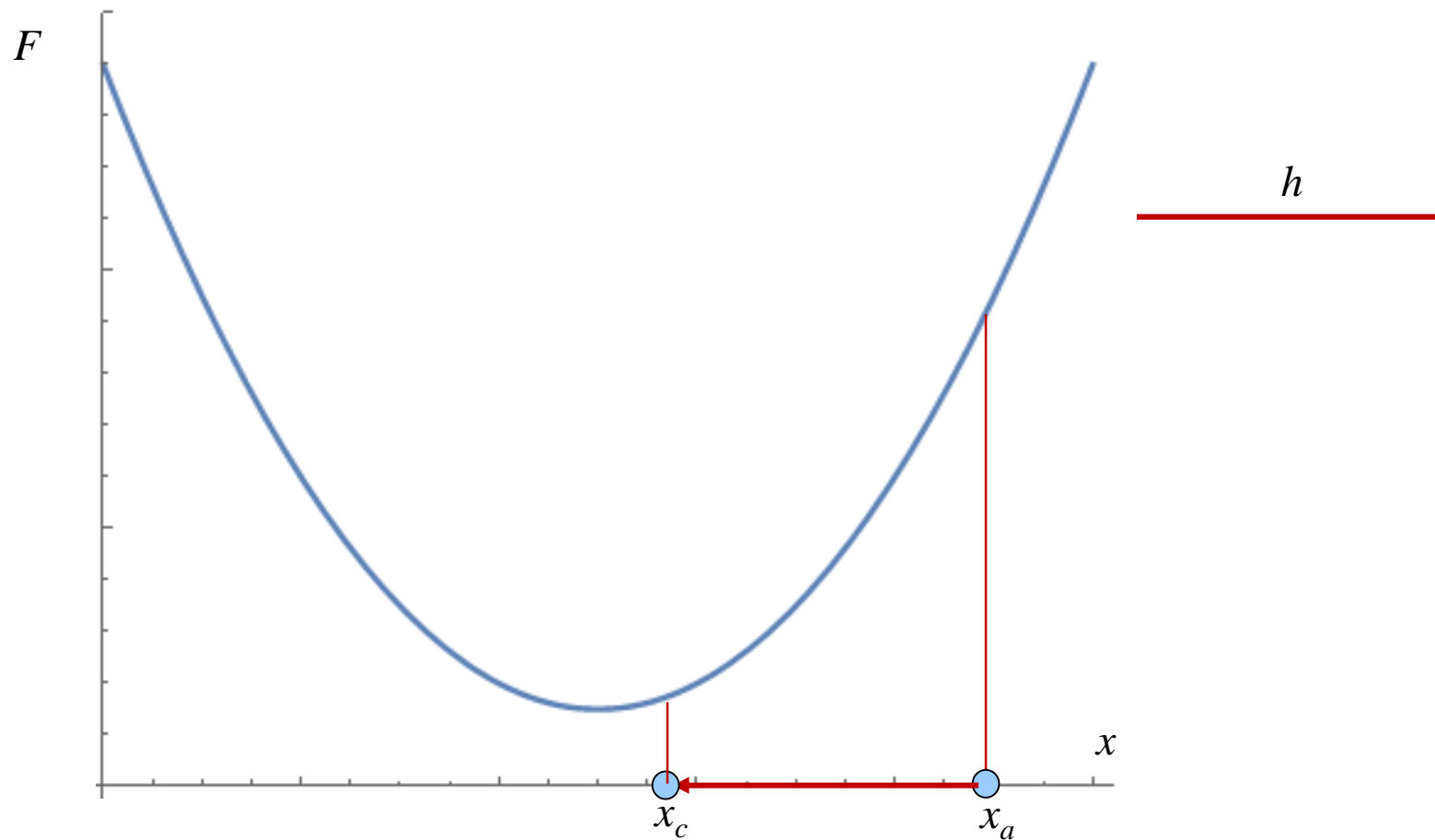
Нелинейное программирование.
Безусловная минимизация функций одной переменной.
Метод золотого сечения.



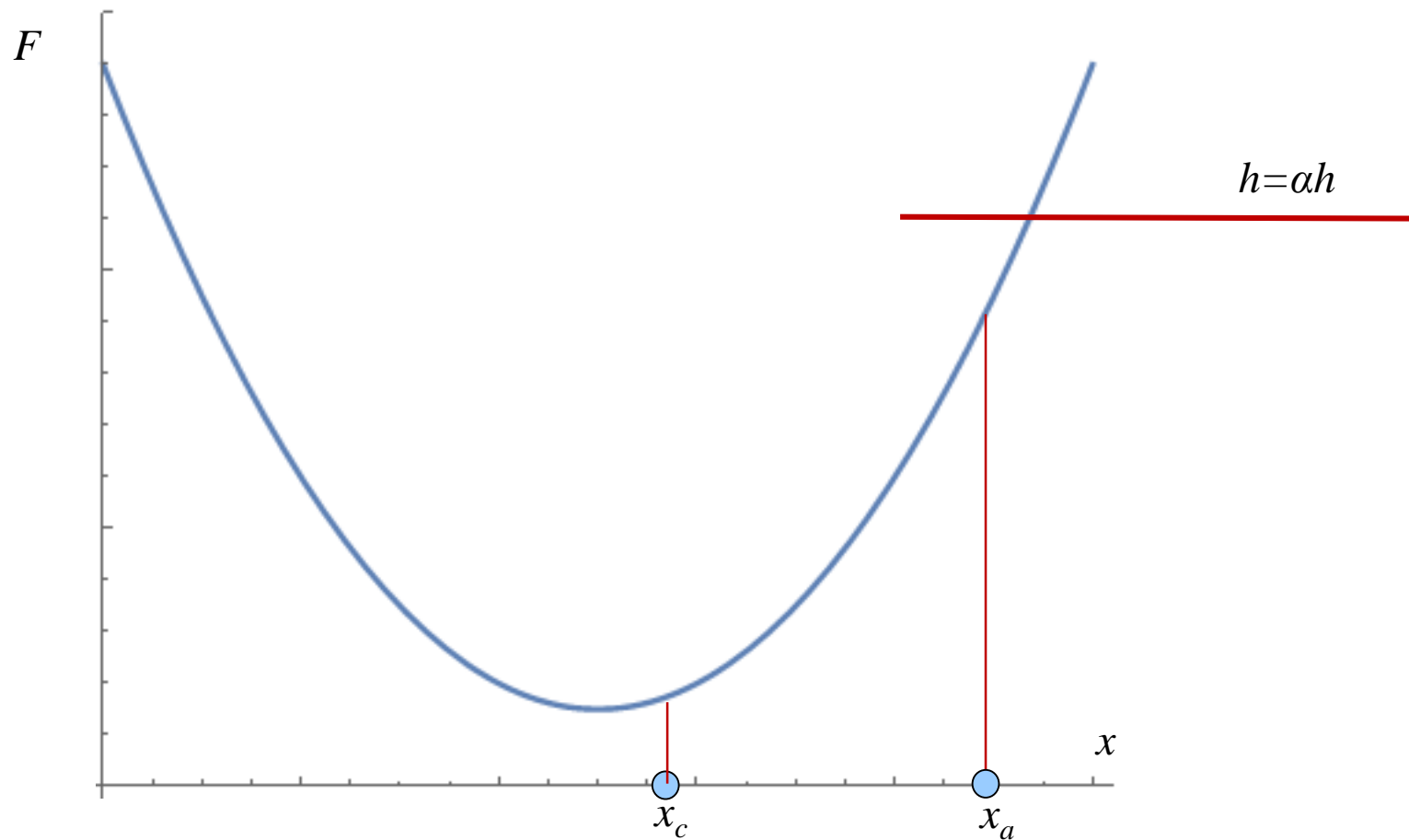
Нелинейное программирование.

Безусловная минимизация функций одной переменной.

Метод золотого сечения.



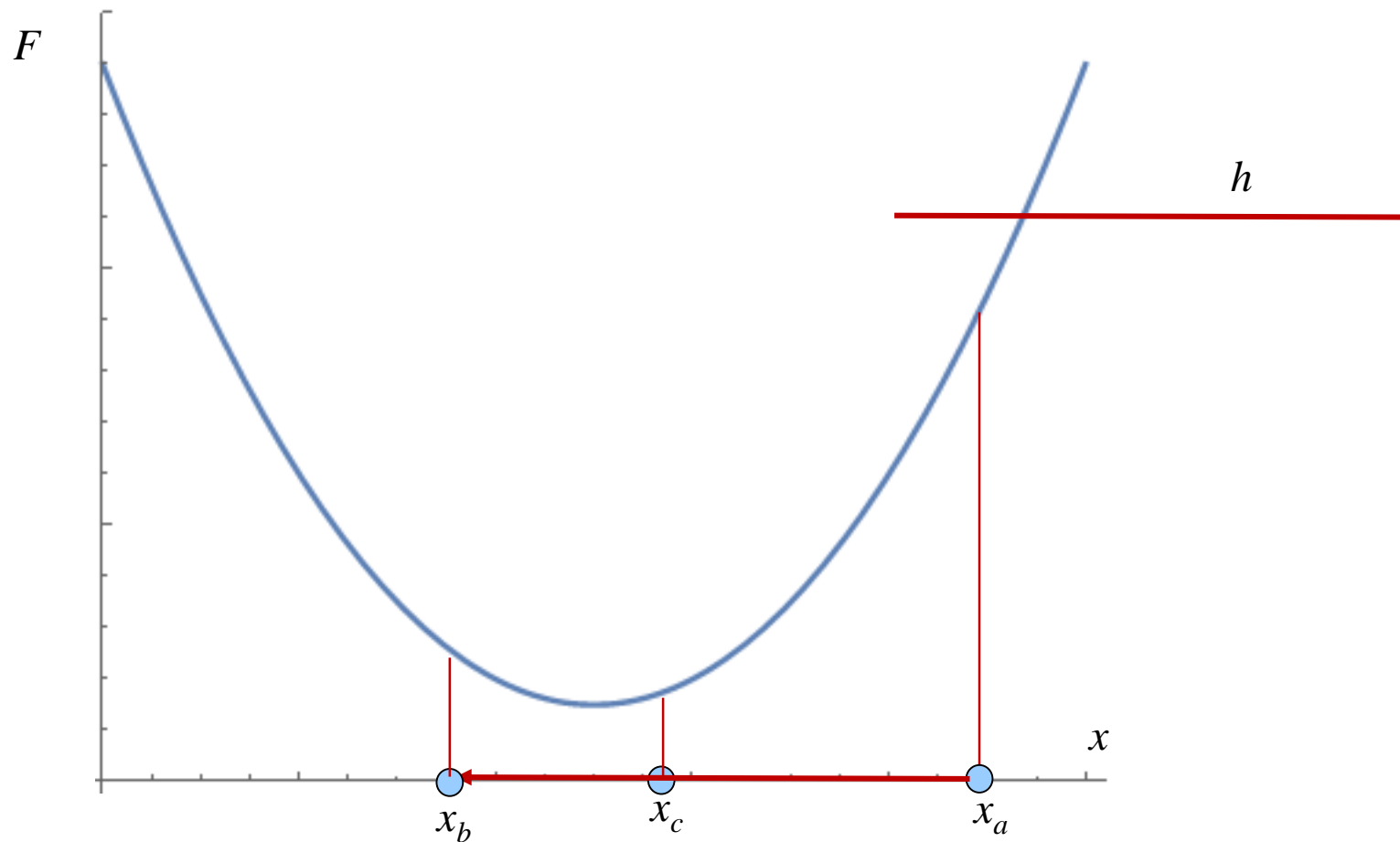
Нелинейное программирование.
Безусловная минимизация функций одной переменной.
Метод золотого сечения.



Нелинейное программирование.

Безусловная минимизация функций одной переменной.

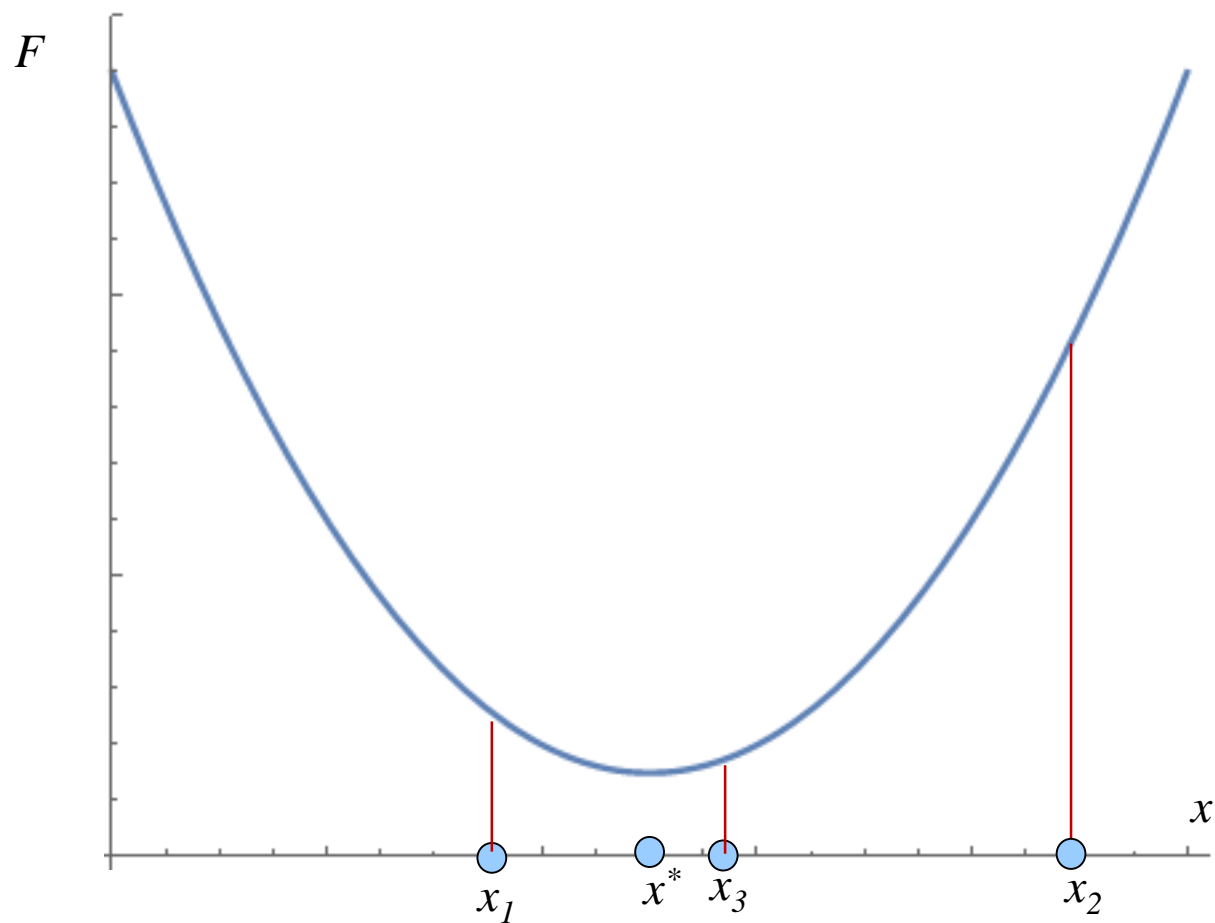
Метод золотого сечения.



Нелинейное программирование.

Безусловная минимизация функций одной переменной.

Метод золотого сечения.



Нелинейное программирование.

Безусловная минимизация функций одной переменной.

Метод золотого сечения

- Алгоритм локализации минимума

$$\frac{|x_2 - x_1|}{|x_3 - x_1|} = \alpha$$

1. Выбрать произвольное x_a , $h > 0$, $1 < \alpha < 2$.
2. $x_c = x_a + h$
3. Если $F(x_c) > F(x_a)$ то $h = -h$,
4. $x_c = x_a + h$
3. Если $F(x_c) > F(x_a)$ то $h = h/10$, перейти к шагу 2.
5. $h = \alpha h$
6. $x_b = x_a + h$
7. Если $F(x_b) < F(x_c)$, то $x_c = x_b$ и перейти к шагу 4.
8. $x_3 = x_c$.
9. Если $x_a < x_b$, то $x_1 = x_a$, $x_2 = x_b$, иначе $x_1 = x_b$, $x_2 = x_a$

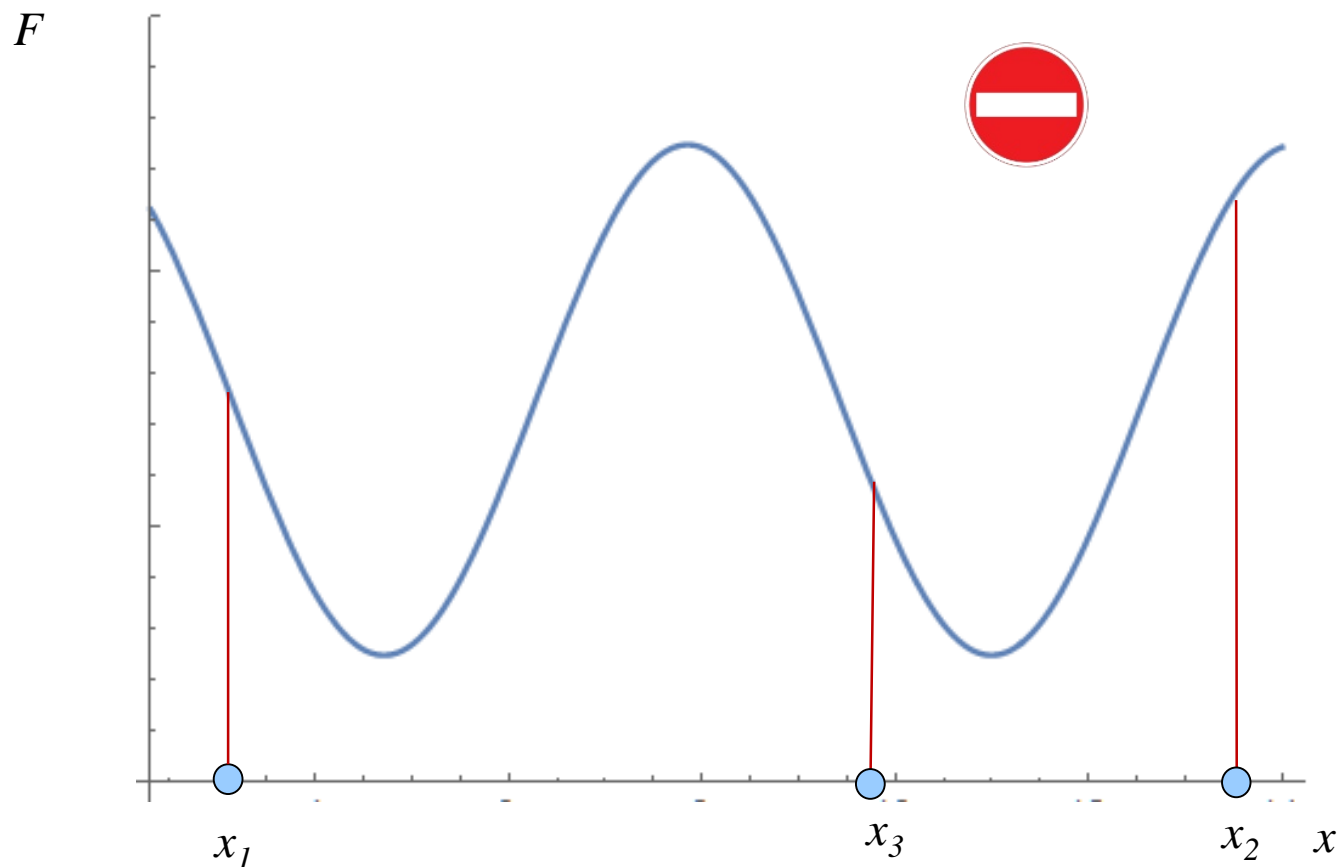
Нелинейное программирование.

Безусловная минимизация функций одной переменной.

Метод золотого сечения.

Ограничение:

- Не гарантирована минимизация многоэкстремальной функции



Нелинейное программирование.

Безусловная минимизация функций одной переменной.

Метод золотого сечения

- Алгоритм метода золотого сечения (1)

1. Взять точки x_1, x_2, x_3 , такие что

$$x_1 < x_3 < x_2, F(x_3) < F(x_1) \text{ и } F(x_3) < F(x_2).$$

Задать точность нахождения минимума ε .

2. Положить $\delta = |x_2 - x_1|$

3. Пока $\delta > \varepsilon$ выполнить

3.1. Выбрать следующую точку симметрично x_3 относительно середины интервала (x_1, x_2) :

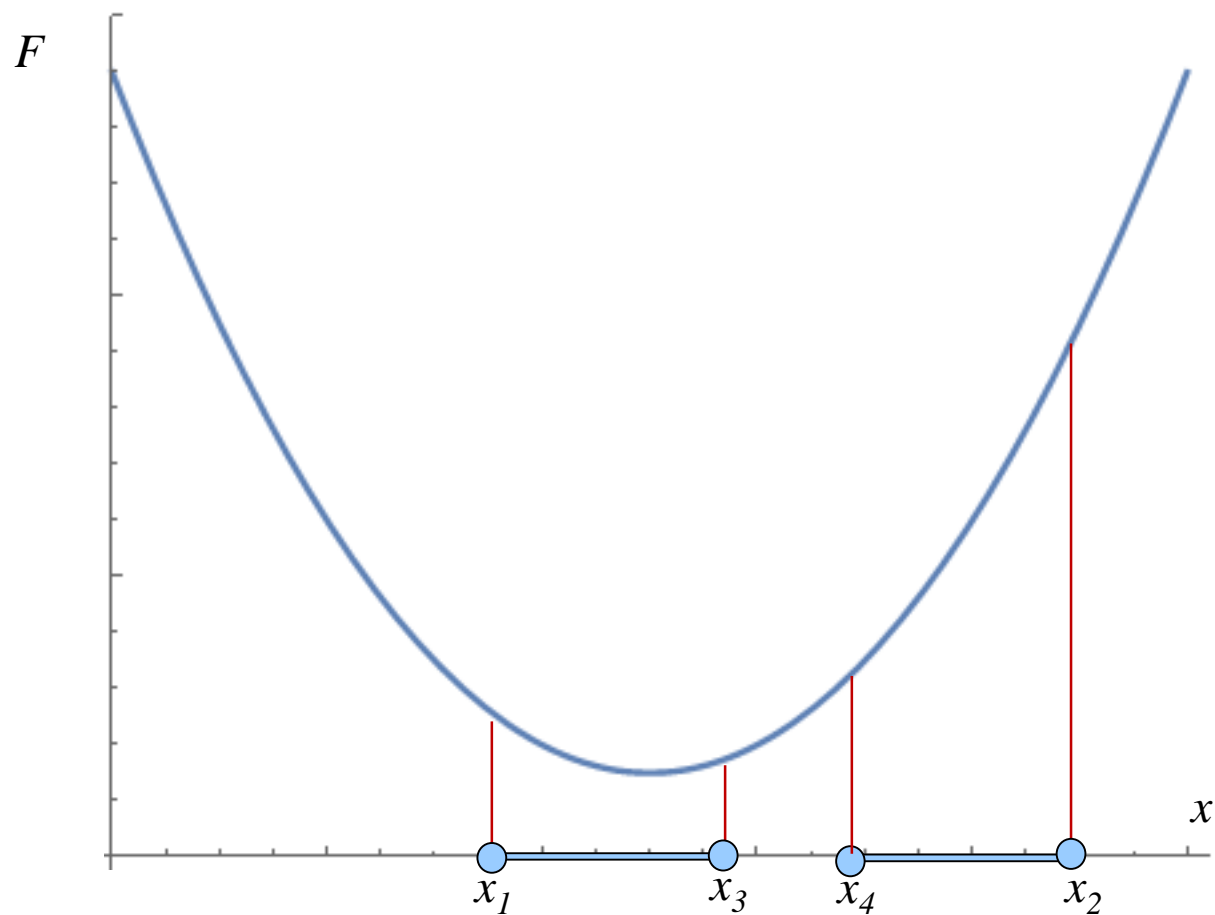
$$x_4 - x_1 = x_2 - x_3,$$

$$x_4 = x_1 + x_2 - x_3.$$

Нелинейное программирование.

Безусловная минимизация функций одной переменной.

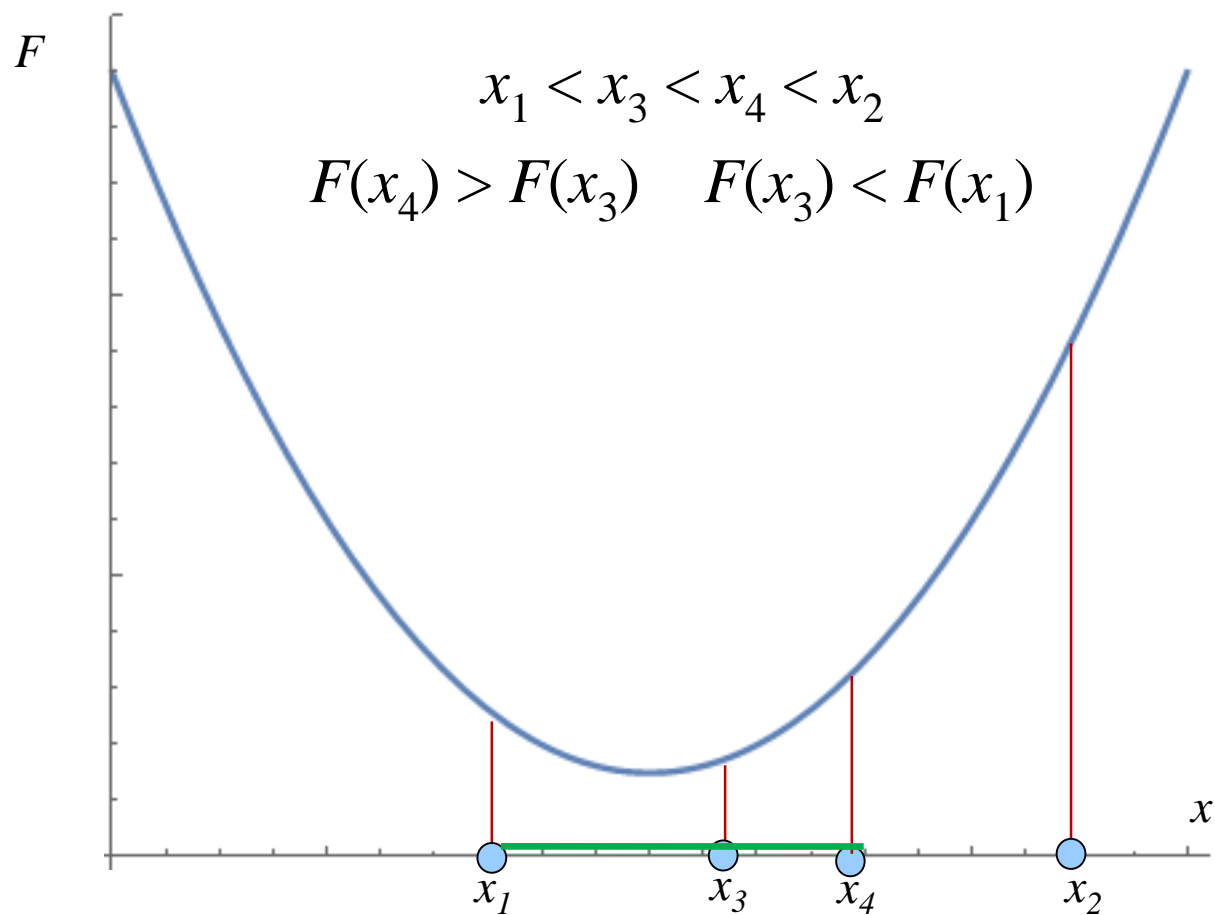
Метод золотого сечения.



Нелинейное программирование.

Безусловная минимизация функций одной переменной.

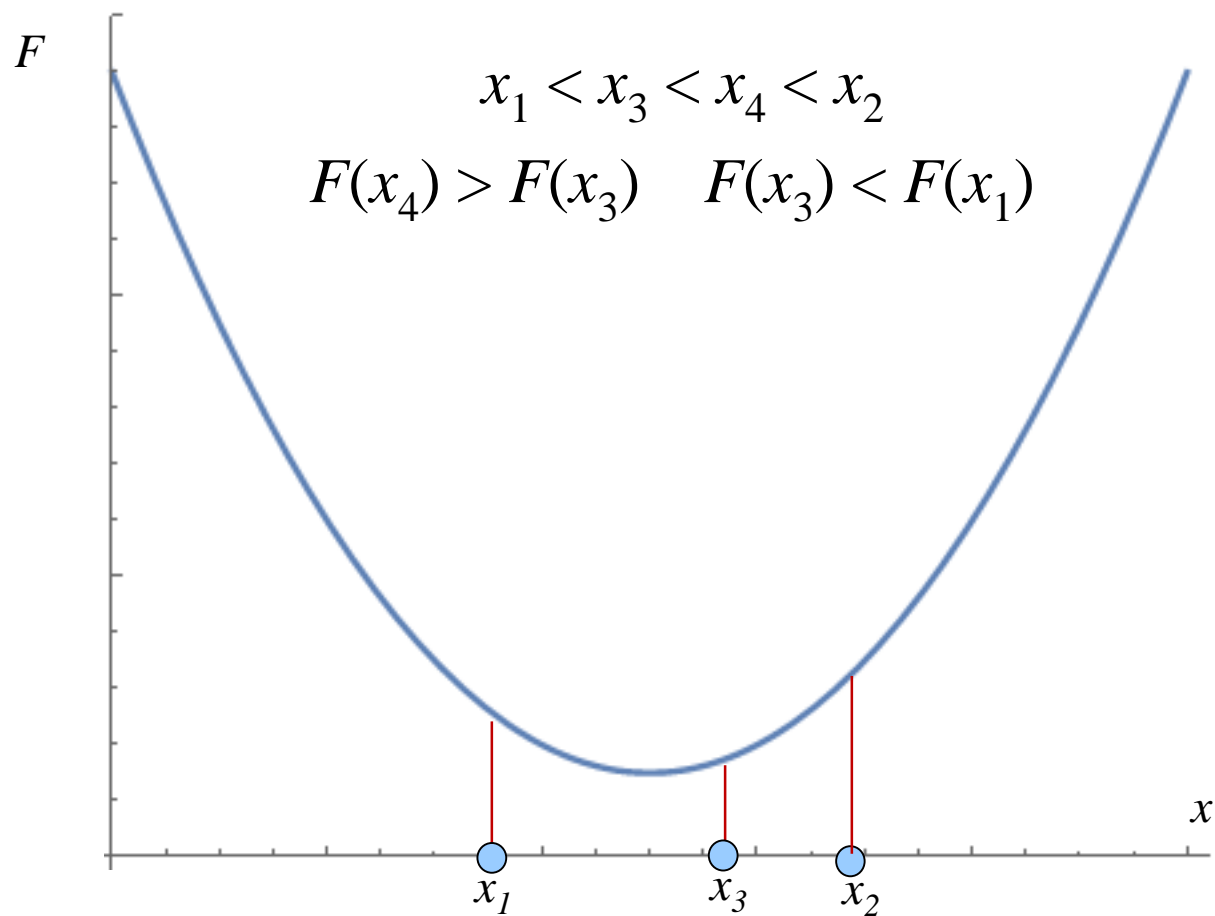
Метод золотого сечения.



Нелинейное программирование.

Безусловная минимизация функций одной переменной.

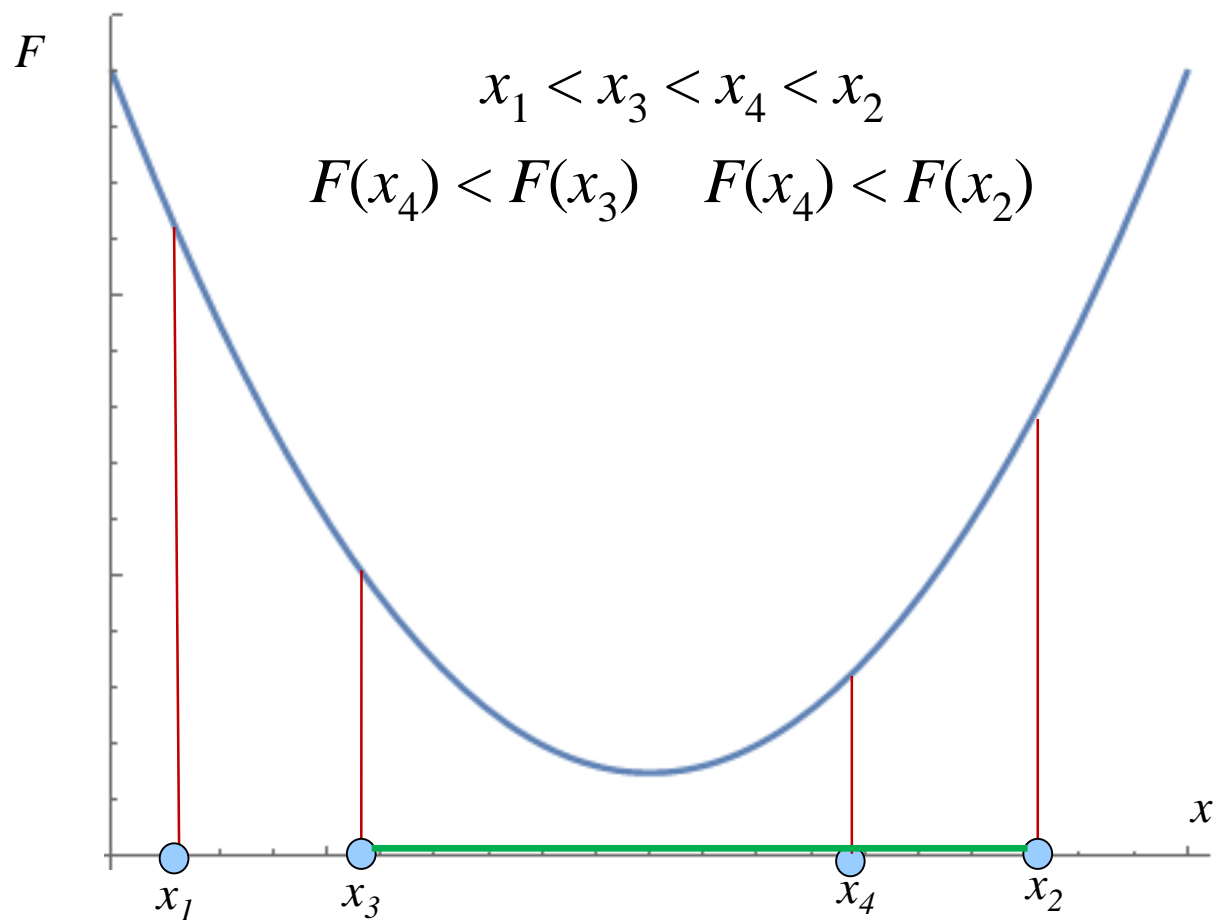
Метод золотого сечения.



Нелинейное программирование.

Безусловная минимизация функций одной переменной.

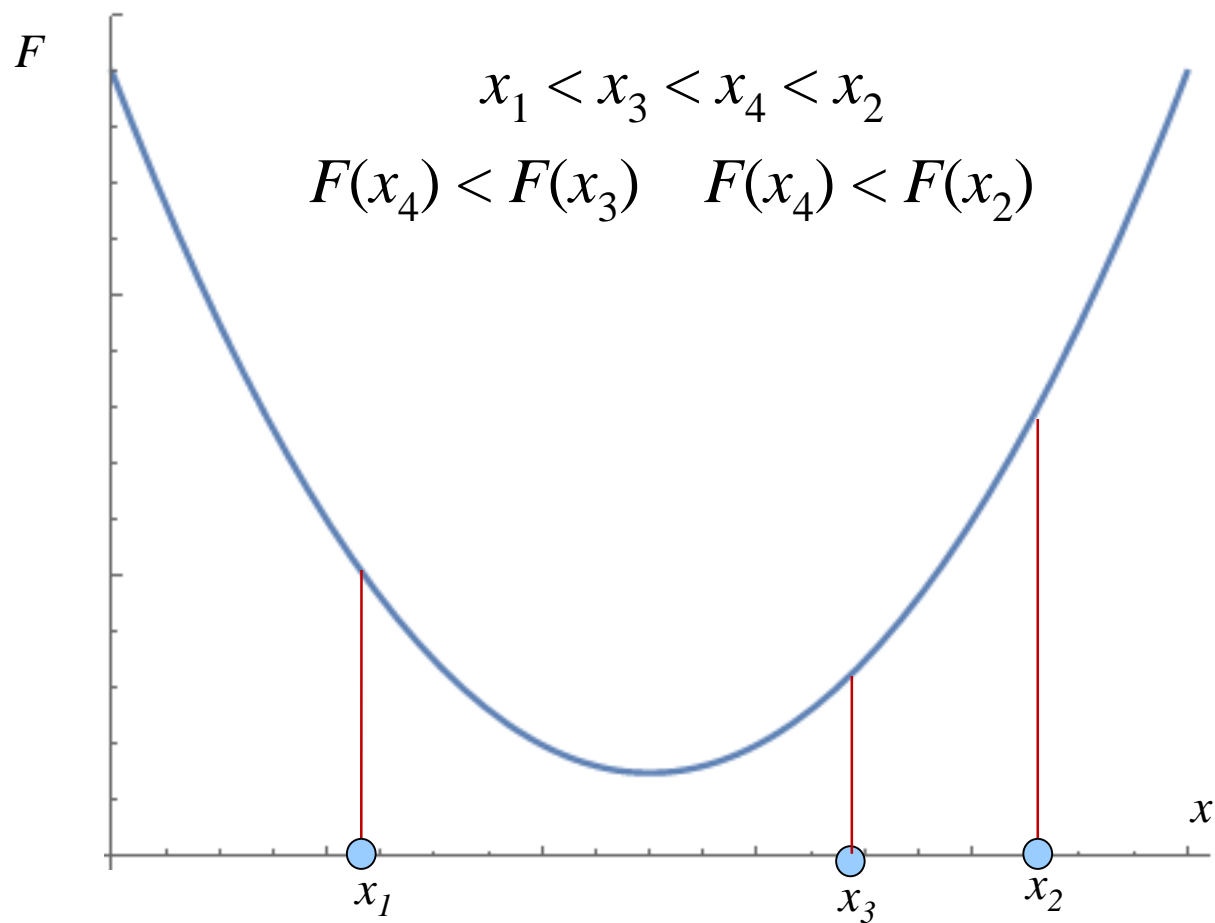
Метод золотого сечения.



Нелинейное программирование.

Безусловная минимизация функций одной переменной.

Метод золотого сечения.

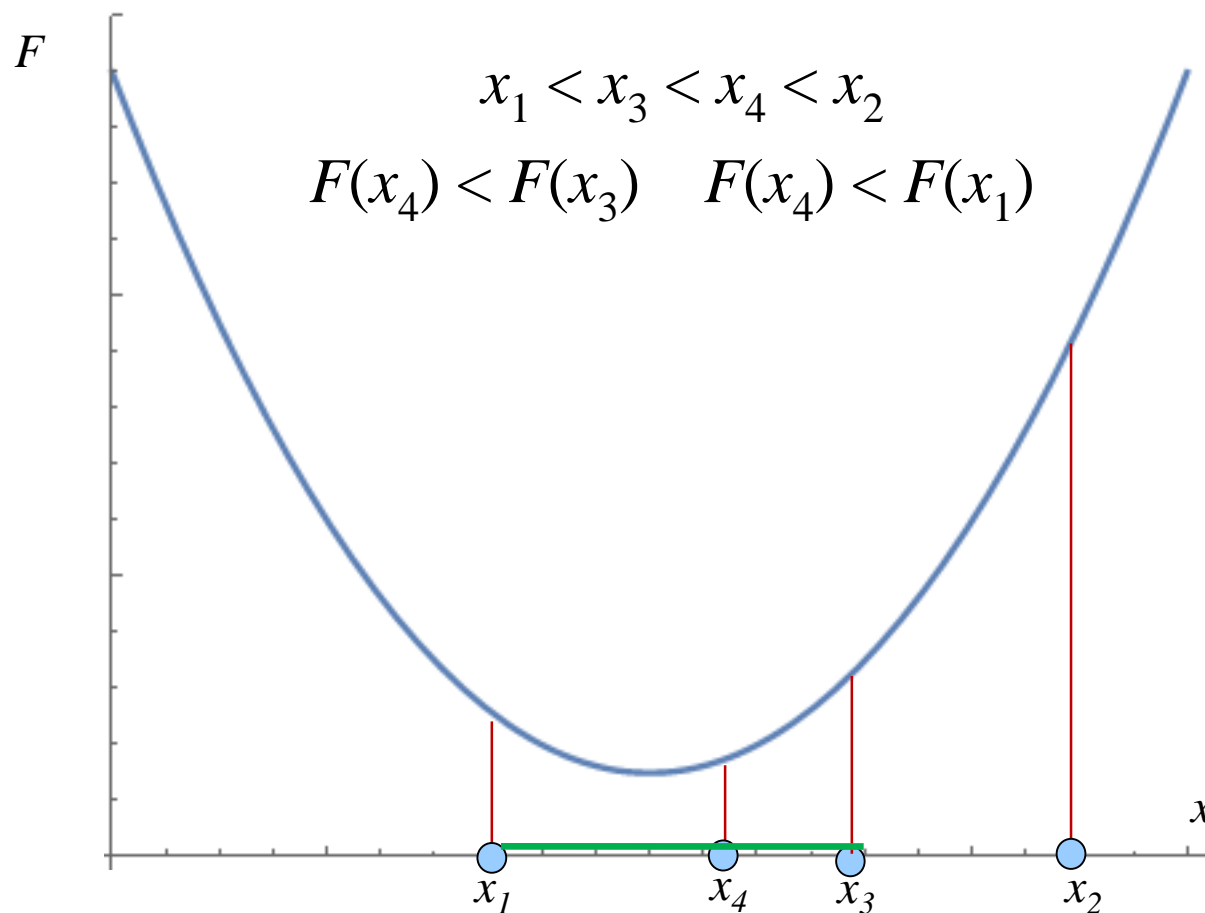


Нелинейное программирование.

Безусловная минимизация функций одной переменной.

Метод золотого сечения.

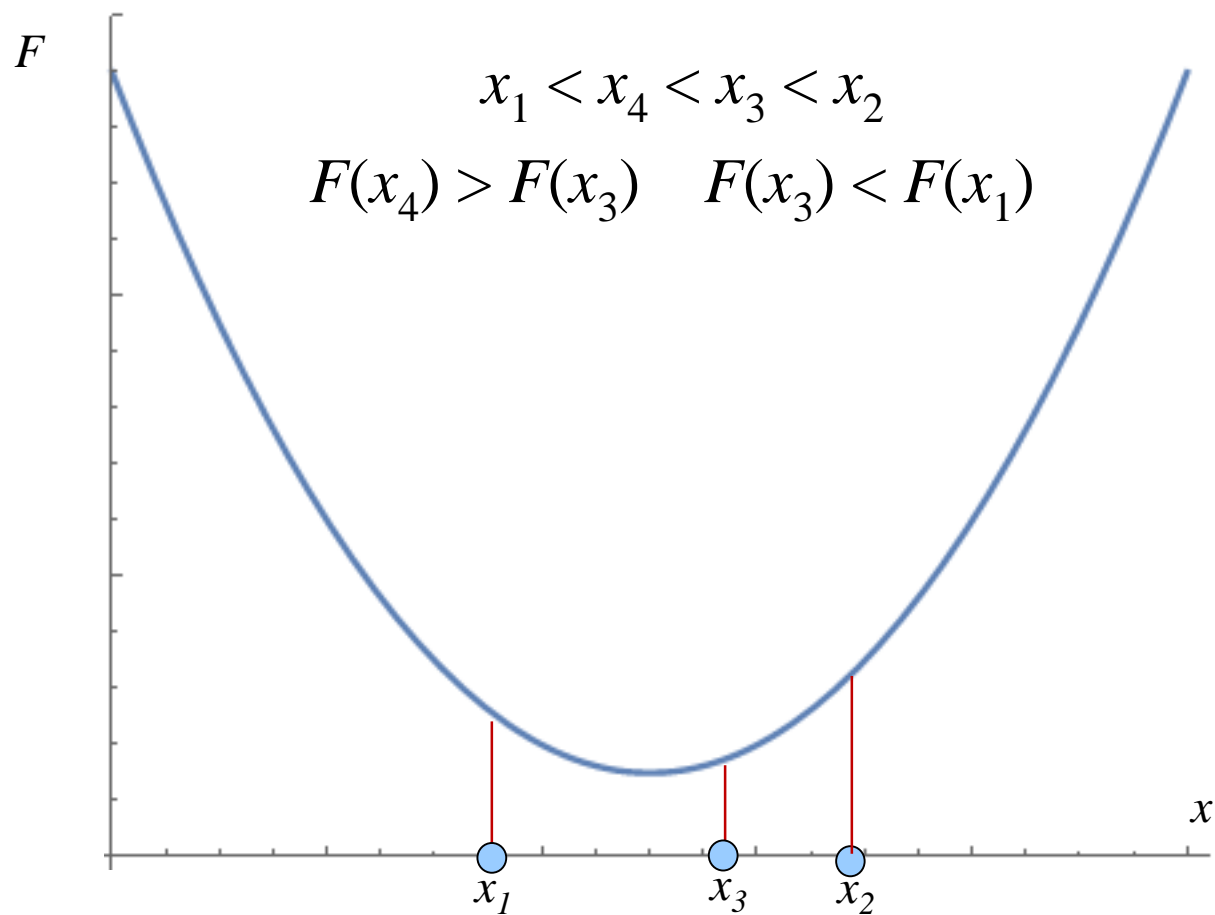
3



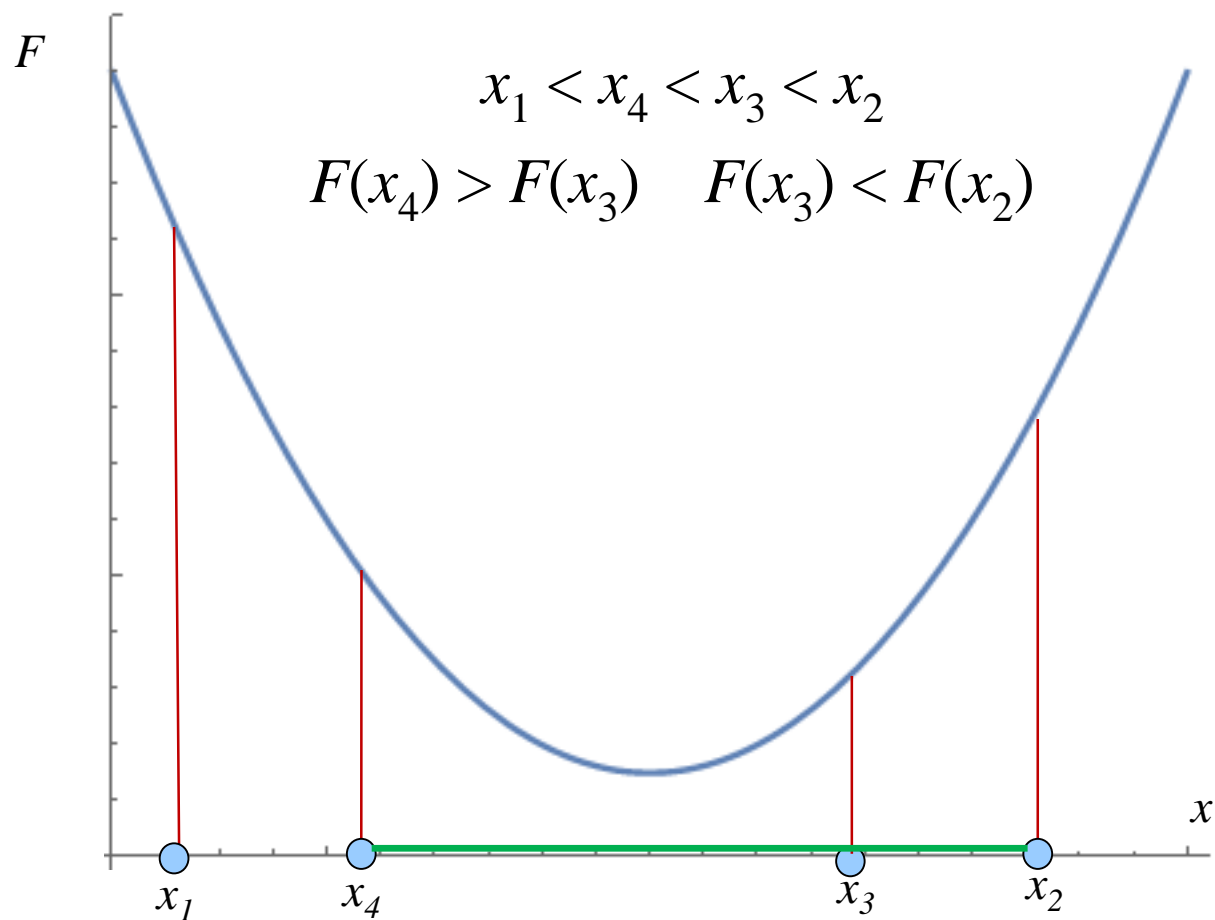
Нелинейное программирование.

Безусловная минимизация функций одной переменной.

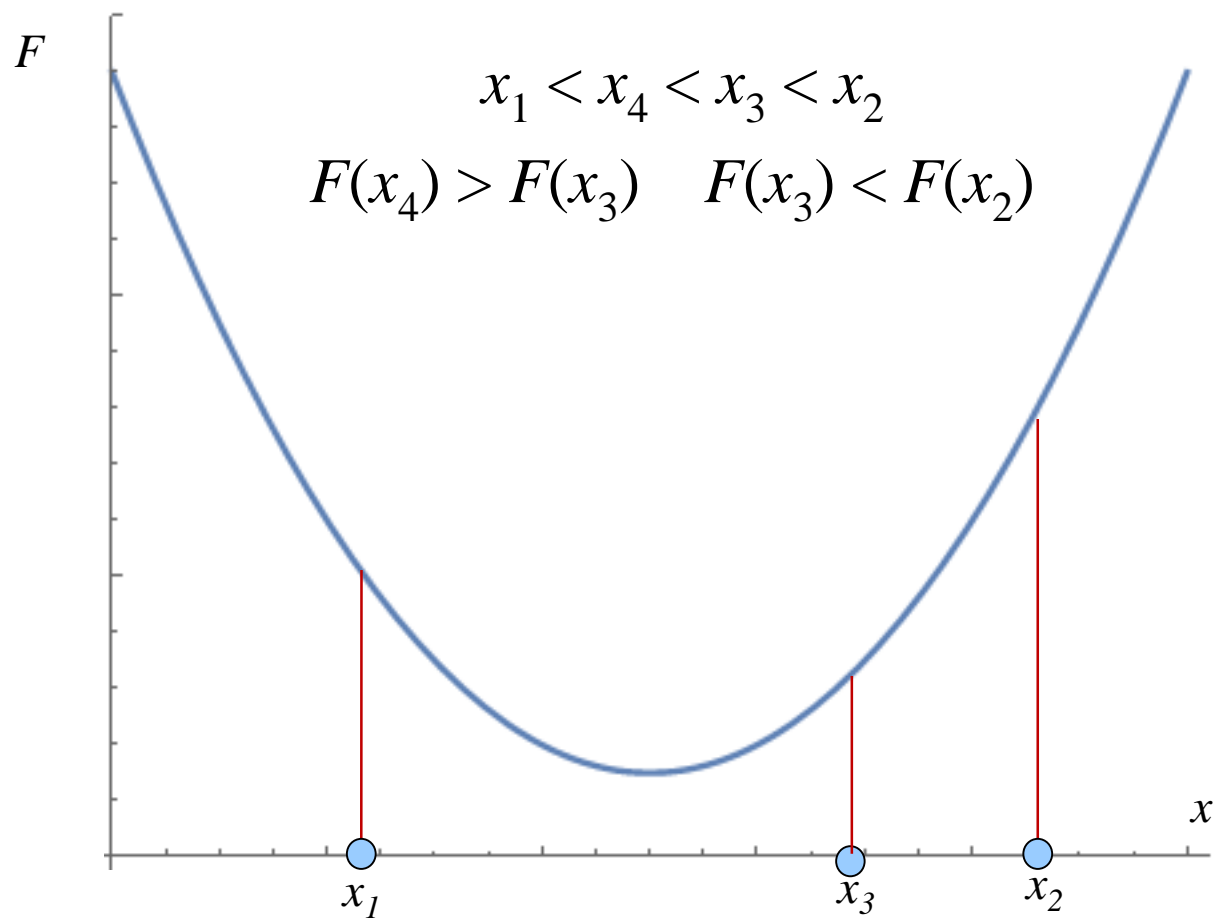
Метод золотого сечения.



Нелинейное программирование. Безусловная минимизация функций одной переменной. Метод золотого сечения.



Нелинейное программирование. Безусловная минимизация функций одной переменной. Метод золотого сечения.



Нелинейное программирование.

Безусловная минимизация функций одной переменной.

Метод золотого сечения

- Алгоритм метода золотого сечения (2)

3.2. Если $x_4 < x_3$, то

если $F(x_4) < F(x_3)$, то $x_2 = x_3$, $x_3 = x_4$

иначе $x_1 = x_4$

иначе

если $F(x_4) < F(x_3)$, то $x_1 = x_3$, $x_3 = x_4$

иначе $x_2 = x_4$

3.3. $\delta = |x_2 - x_1|$.

4. $x^* = (x_2 + x_1)/2$.

5. Завершить работу.

Нелинейное программирование.

Безусловная минимизация функций одной переменной.

Метод золотого сечения

- Алгоритм метода золотого сечения

1. Взять точки x_1, x_2, x_3 , такие что
 $x_1 < x_3 < x_2$, $F(x_3) < F(x_1)$ и $F(x_3) < F(x_2)$.
Задать точность нахождения минимума ε .
2. Положить $\delta = |x_2 - x_1|$
3. Пока $\delta > \varepsilon$ выполнить
 - 3.1. Выбрать следующую точку симметрично x_3 относительно середины интервала (x_1, x_2) :
 $x_4 - x_1 = x_2 - x_3$,
 $x_4 = x_1 + x_2 - x_3$.
 - 3.2. Если $x_4 < x_3$, то
если $F(x_4) < F(x_3)$, то $x_2 = x_3, x_3 = x_4$
иначе $x_1 = x_4$
иначе
если $F(x_4) < F(x_3)$, то $x_1 = x_3, x_3 = x_4$
иначе $x_2 = x_4$
 - 3.3. $\delta = |x_2 - x_1|$.
4. $x^* = (x_2 + x_1)/2$.
5. Завершить работу.

Нелинейное программирование.

Безусловная минимизация функций одной переменной.

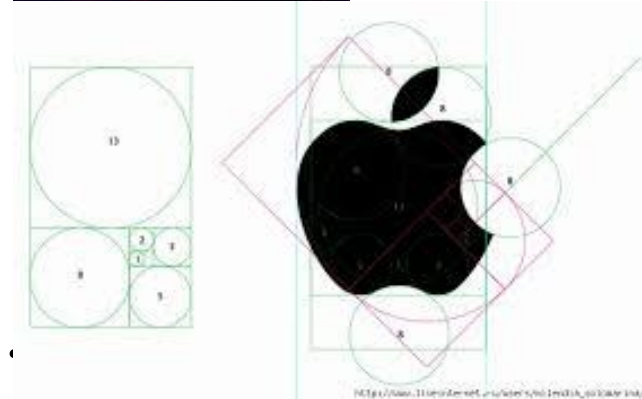
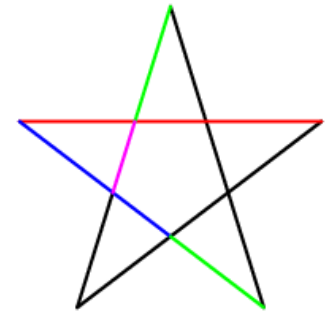
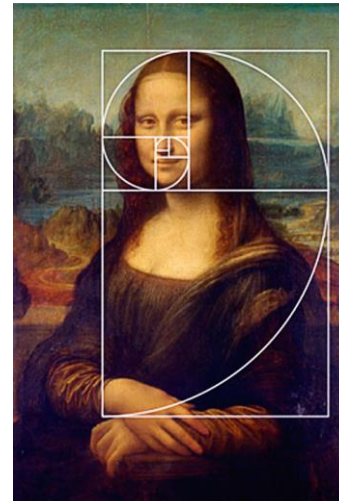
Метод золотого сечения

- Алгоритм метода золотого сечения

$$\frac{x_2 - x_1}{x_3 - x_1} = \frac{x_3 - x_1}{x_2 - x_3} = \alpha$$

$$\alpha = \frac{\frac{x_2 - x_1}{\alpha}}{x_2 - x_1 - \frac{x_2 - x_1}{\alpha}} = \frac{1}{1 - \alpha}$$

$$\alpha^2 - \alpha - 1 = 0 \Rightarrow \alpha = 1,618...$$

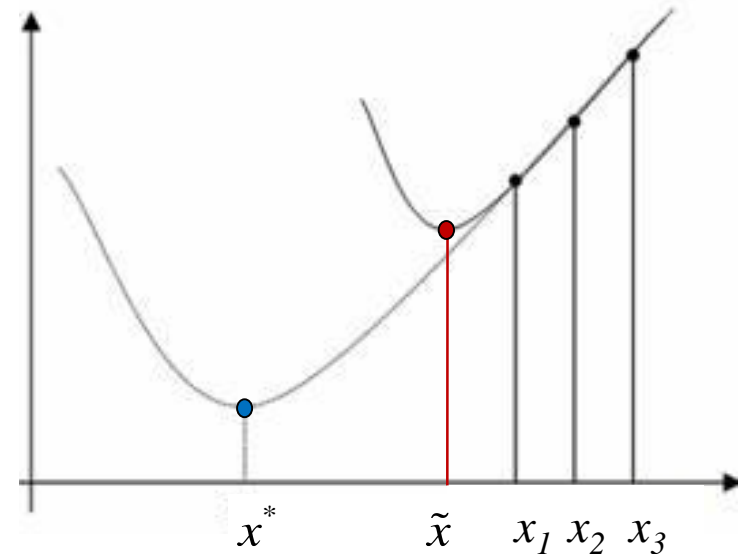


Нелинейное программирование.
Безусловная минимизация функций одной переменной.
Метод Пауэлла (квадратичной аппроксимации)

$$F \approx y(x) = ax^2 + bx + c$$

$$\begin{cases} ax_1^2 + bx_1 + c = F(x_1) \\ ax_2^2 + bx_2 + c = F(x_2) \\ ax_3^2 + bx_3 + c = F(x_3) \end{cases}$$

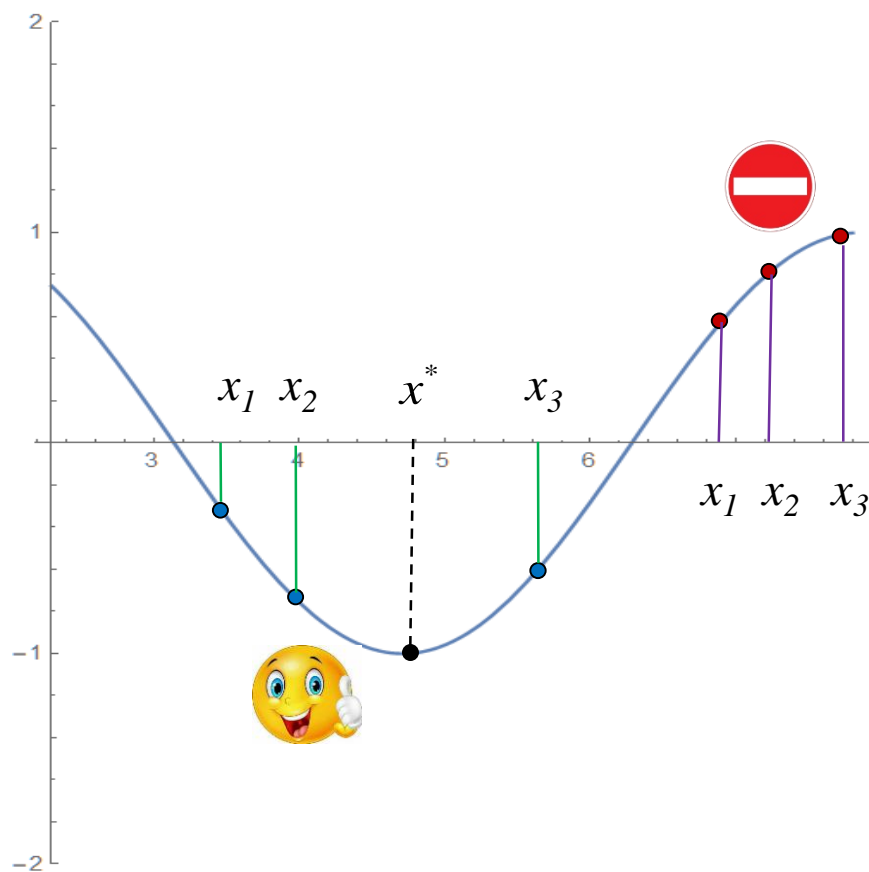
$$x^* \approx \tilde{x} = -b / 2a$$



Нелинейное программирование.

Безусловная минимизация функций одной переменной.

Метод Пауэлла (квадратичной аппроксимации)



Нелинейное программирование.

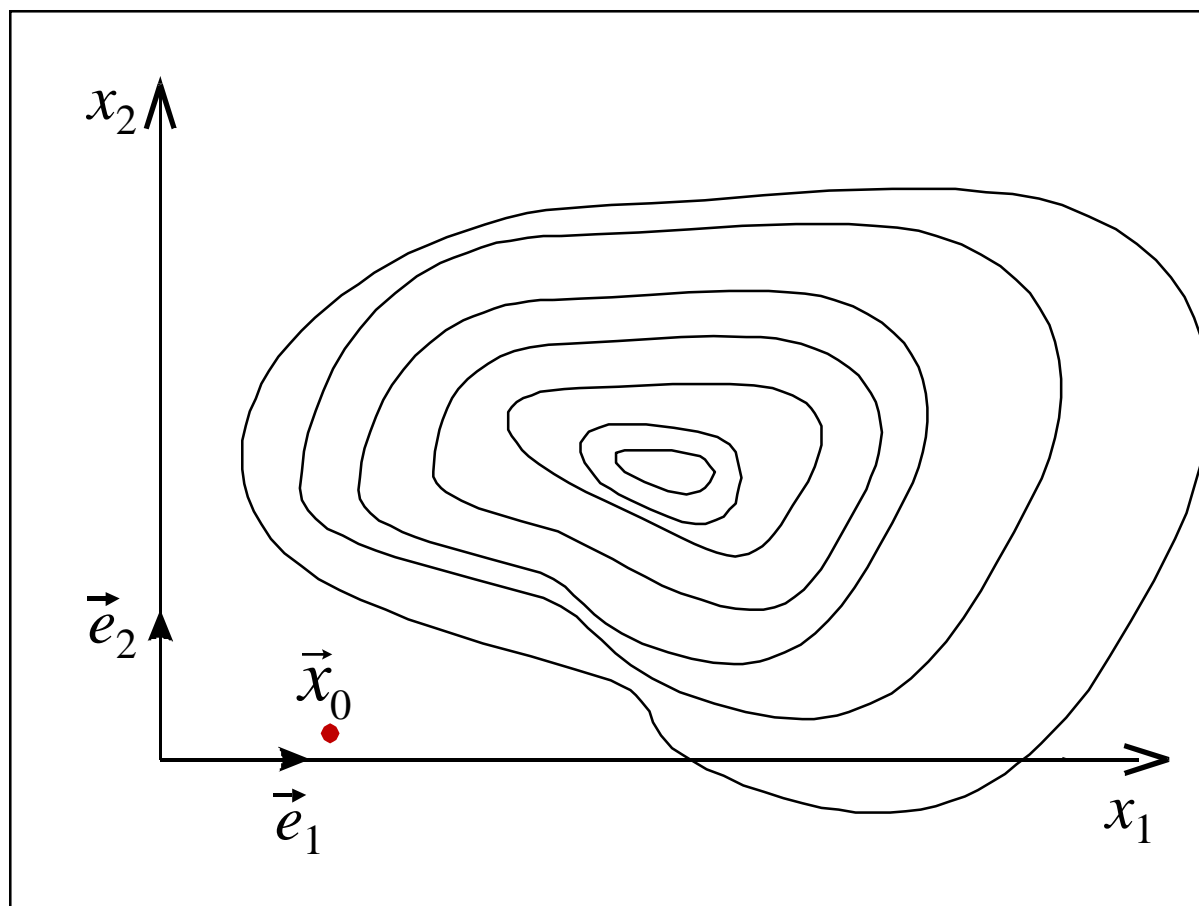
Безусловная минимизация функций одной переменной.

Метод Пауэлла (квадратичной аппроксимации)

1. Выбрать x_1, x_2, x_3 , локализующие $\min F(x)$, задать ε
2. $\Delta = |x_1 - x_2|$
3. Пока $\Delta > \varepsilon$ выполнить
 - 3.1 Найти a, b, c по значениям $x_1, x_2, x_3, F(x_1), F(x_2), F(x_3)$;
 - 3.2 Положить $x_4 = -b / 2a$;
 - 3.3 Выбрать из точек x_1, x_2, x_3 точку x_k , такую что $F(x_k) = \min(F(x_1), F(x_2), F(x_3))$;
 - 3.4 Из точек x_4 и x_k выбрать точку x_{\min} такую, что $F(x_{\min}) = \min(F(x_k), F(x_4))$;
 - 3.5 Если точка x_{\min} не является "крайней",
положить $x_3 = x_{\min}$; x_1 - точка справа, x_2 - слева от x_{\min}
иначе
отбросить точку с максимальным значением целевой функции,
оставшиеся обозначить x_1, x_2, x_3 ;
 - 3.6 $\Delta = |x_1 - x_2|$
4. Положить $x^* = (x_1 + x_2) / 2$.
5. Завершить работу.

Безусловная минимизация функций n переменных. Методы прямого поиска

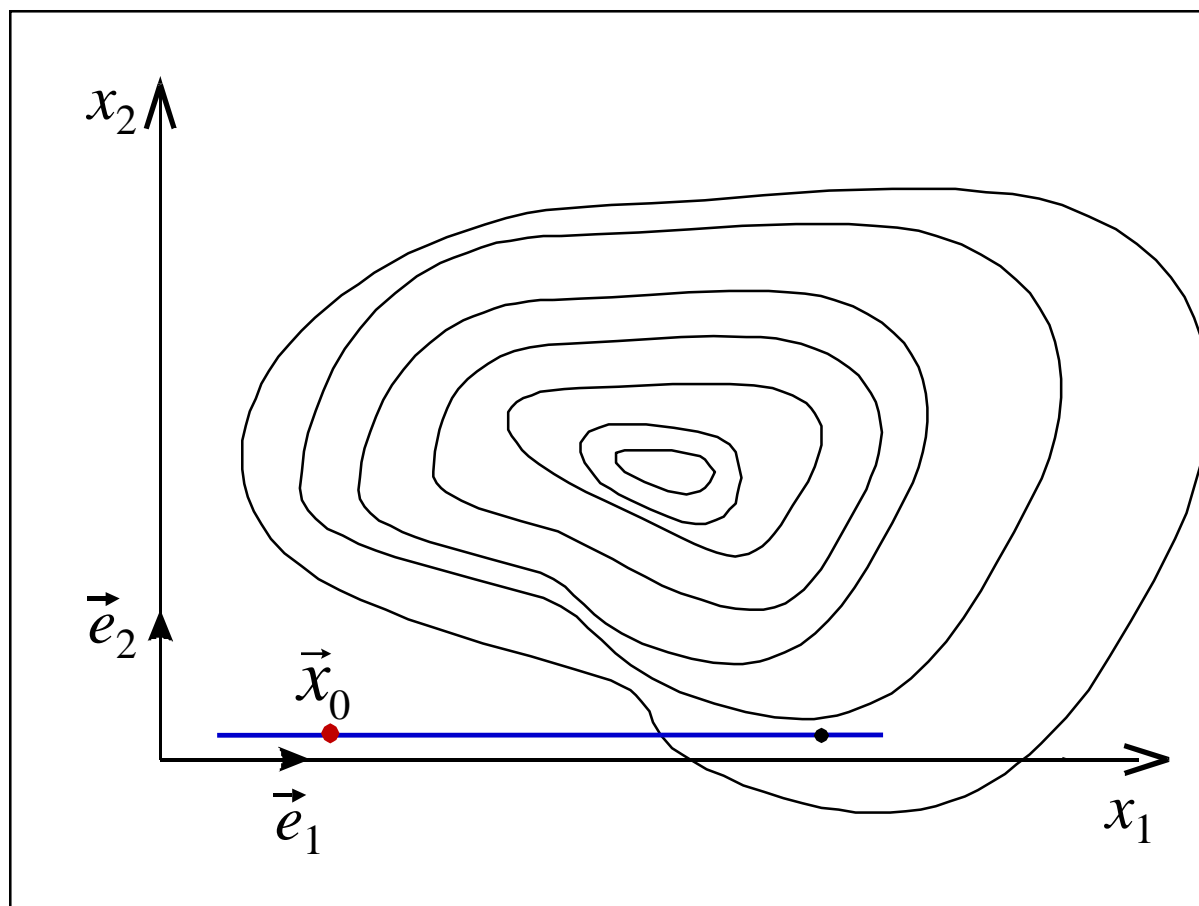
Нелинейное программирование.
Безусловная минимизация функций n переменных.
Метод покоординатного спуска



Нелинейное программирование.

Безусловная минимизация функций n переменных.

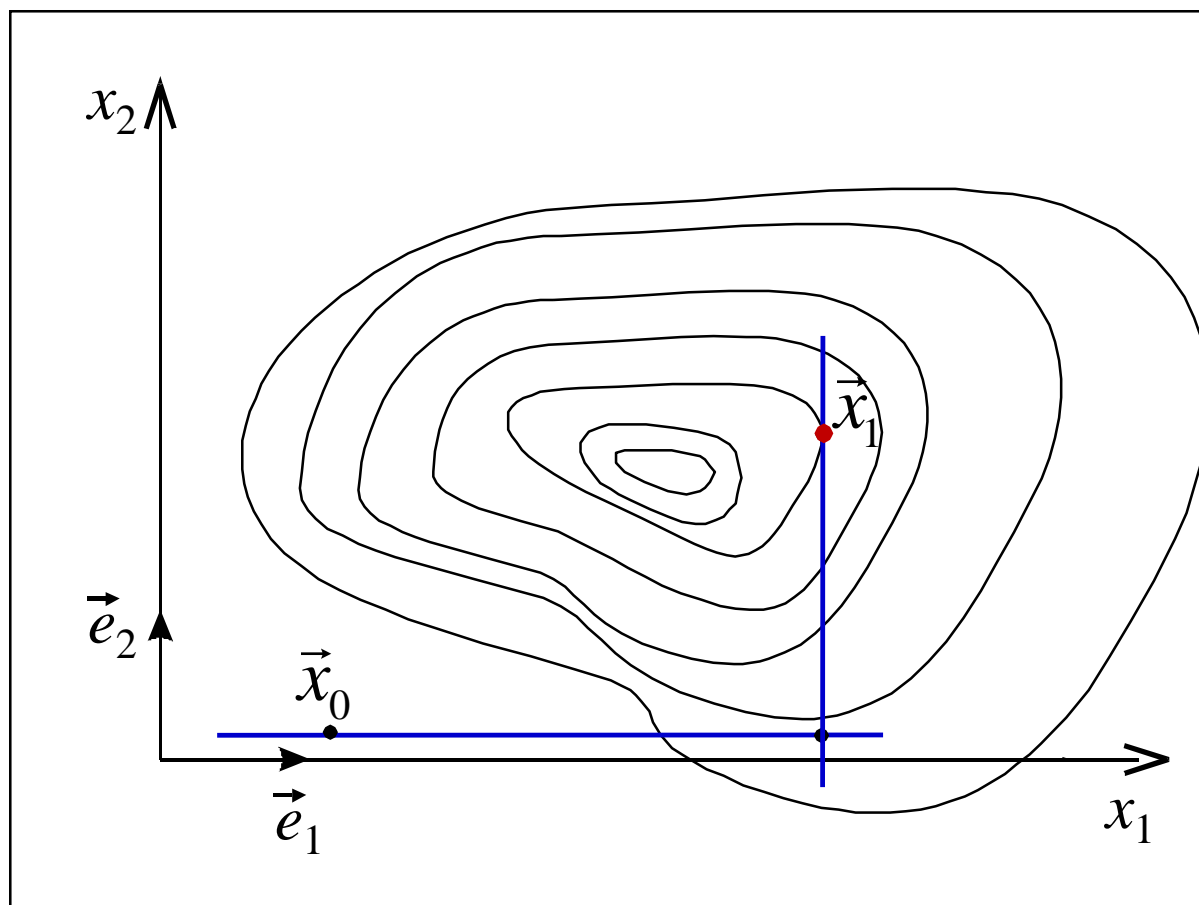
Метод покоординатного спуска



Нелинейное программирование.

Безусловная минимизация функций n переменных.

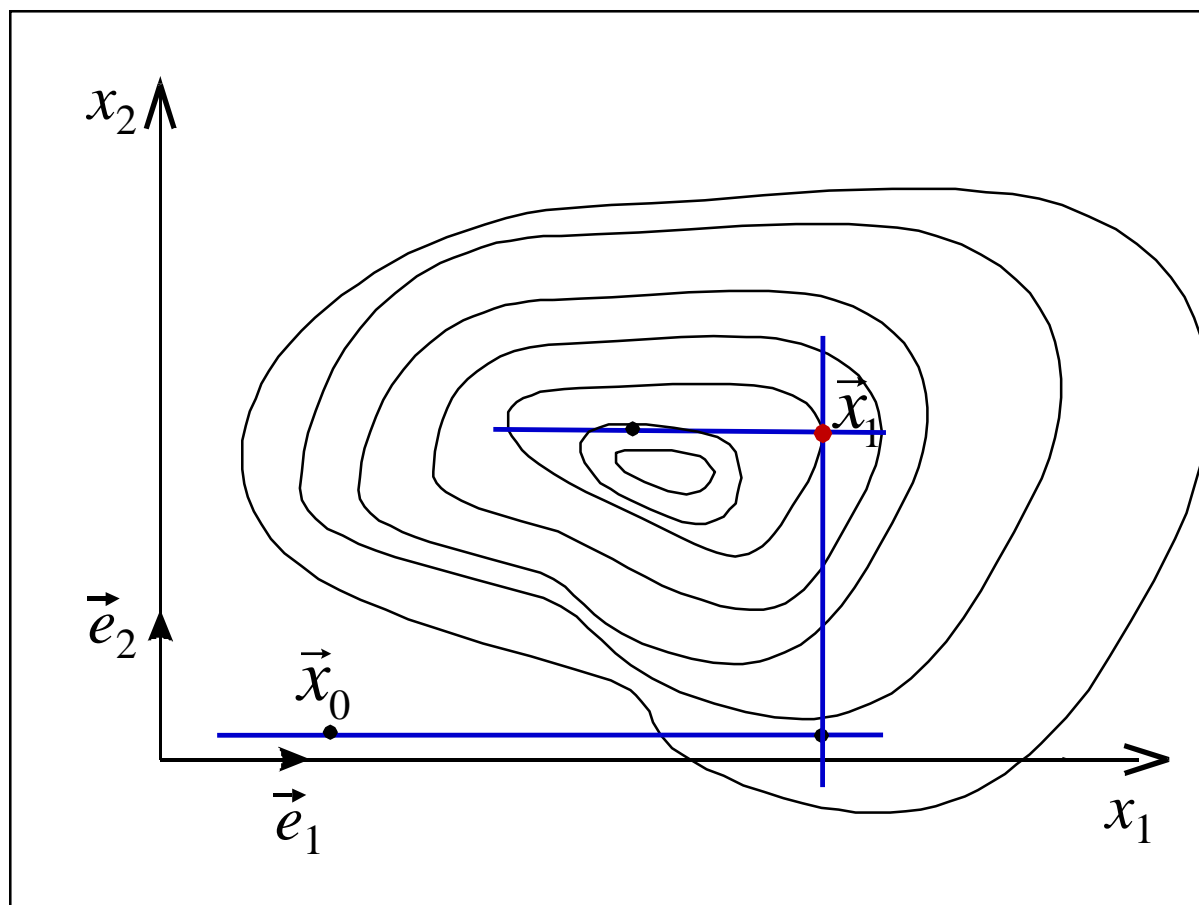
Метод покоординатного спуска



Нелинейное программирование.

Безусловная минимизация функций n переменных.

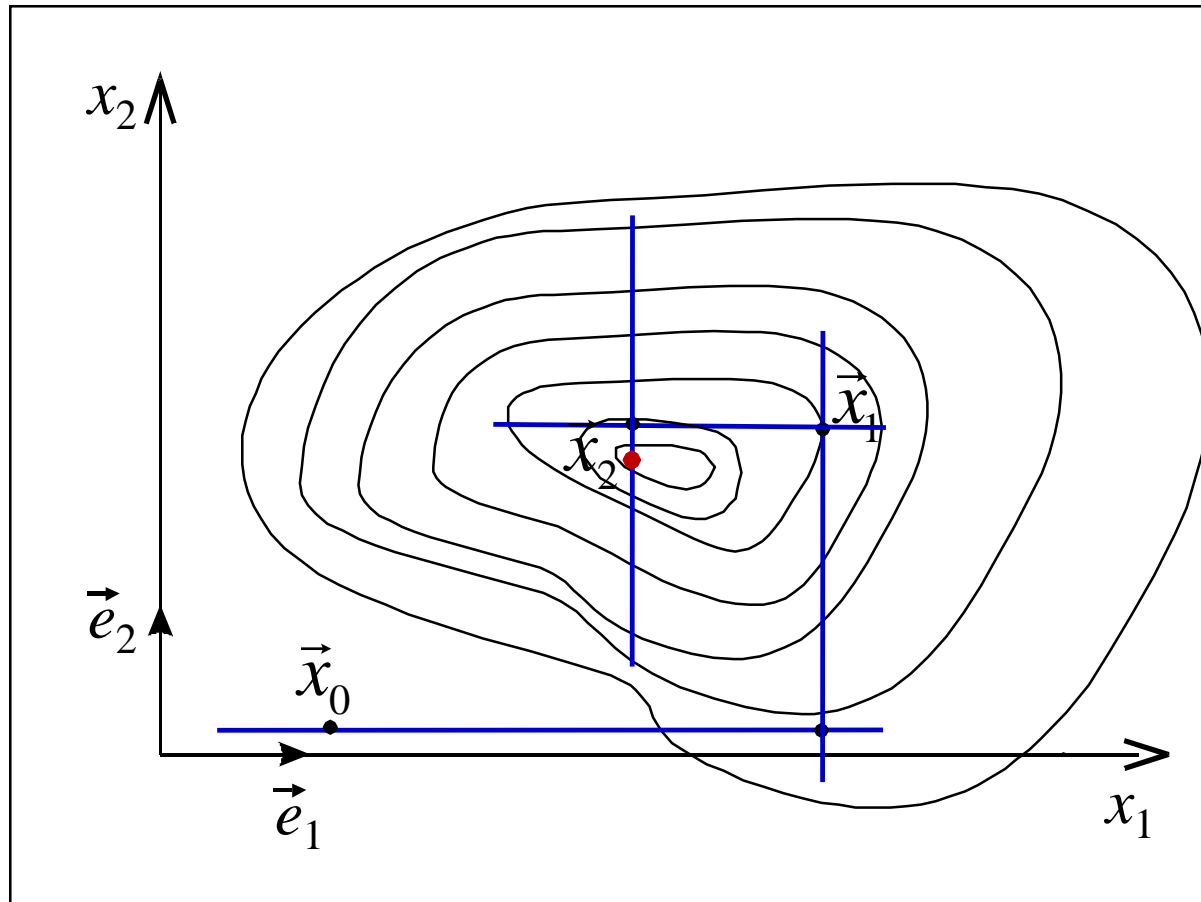
Метод покоординатного спуска



Нелинейное программирование.

Безусловная минимизация функций n переменных.

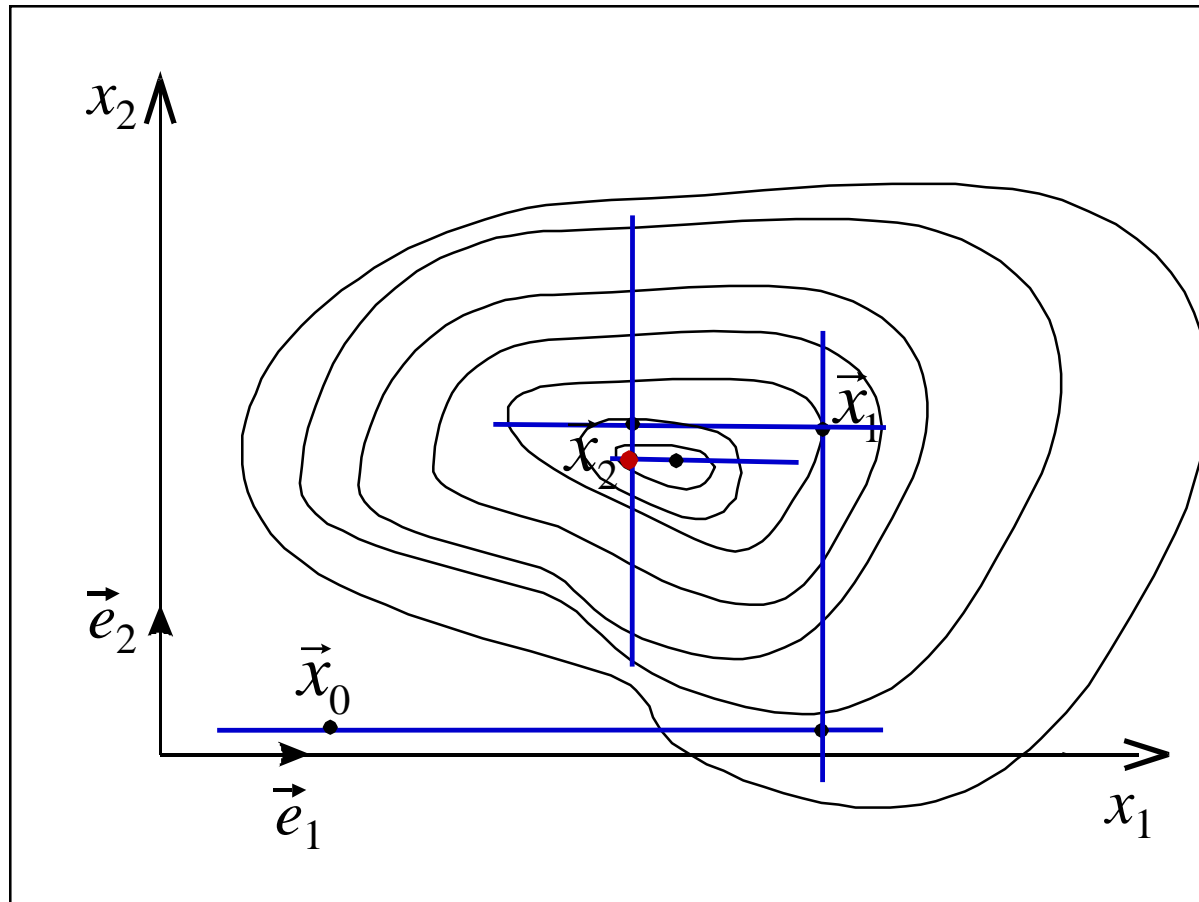
Метод покоординатного спуска



6. Нелинейное программирование.

6.8. Безусловная минимизация функций n переменных.

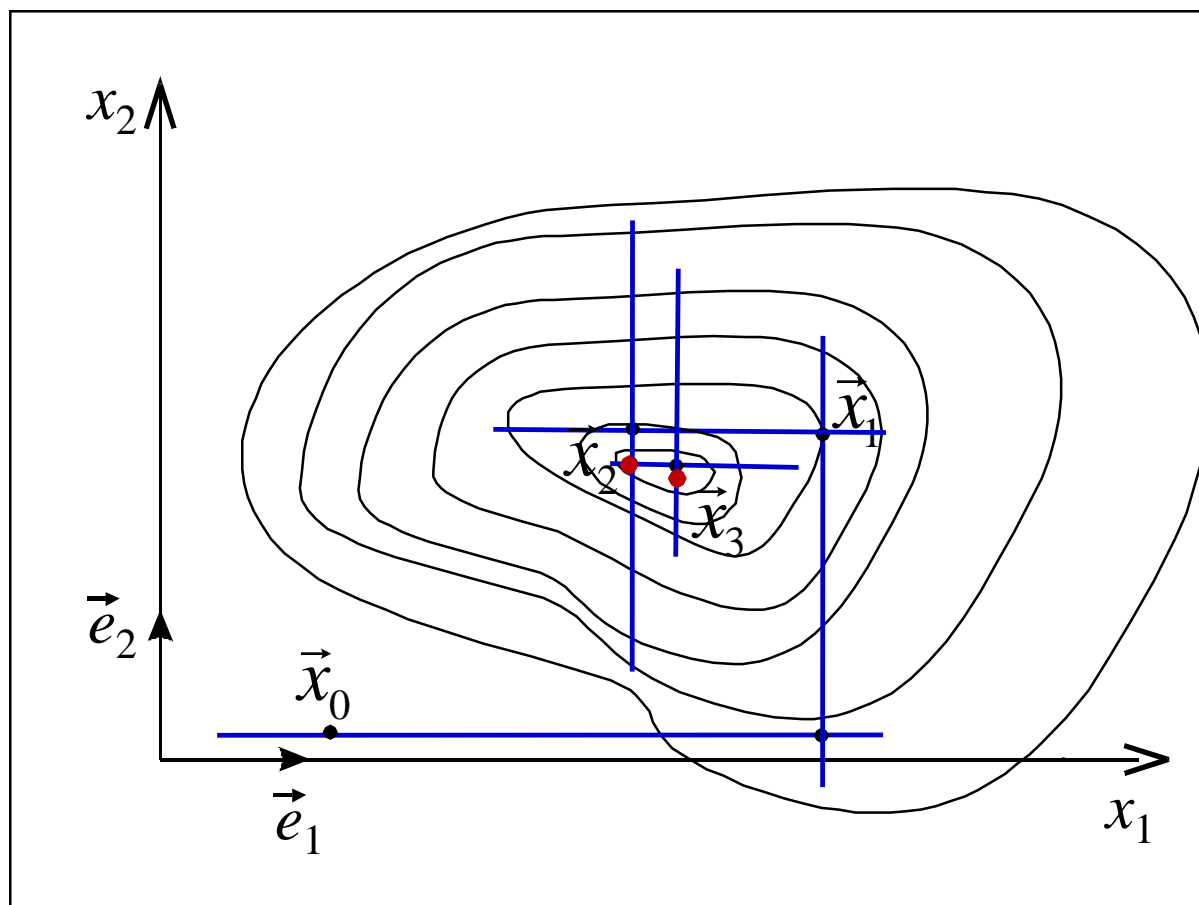
Метод покоординатного спуска



Нелинейное программирование.

Безусловная минимизация функций n переменных.

Метод покоординатного спуска



Нелинейное программирование.

Безусловная минимизация функций n переменных.

Метод покоординатного спуска

1. Выбрать \vec{x}_0 , задать ε

2. $\vec{x}_b = \vec{x}_0$; $\Delta := 2\varepsilon$;

3. Пока $\Delta > \varepsilon$ выполнить

Для всех i от 1 до n выполнить

$$\vec{x}_s = \vec{x}_b;$$

$$\text{Найти } \lambda^* : \min_{\lambda} F(\vec{x}_s + \lambda \vec{e}_i);$$

$$\text{Положить } \vec{x}_s = \vec{x}_s + \lambda \vec{e}_i;$$

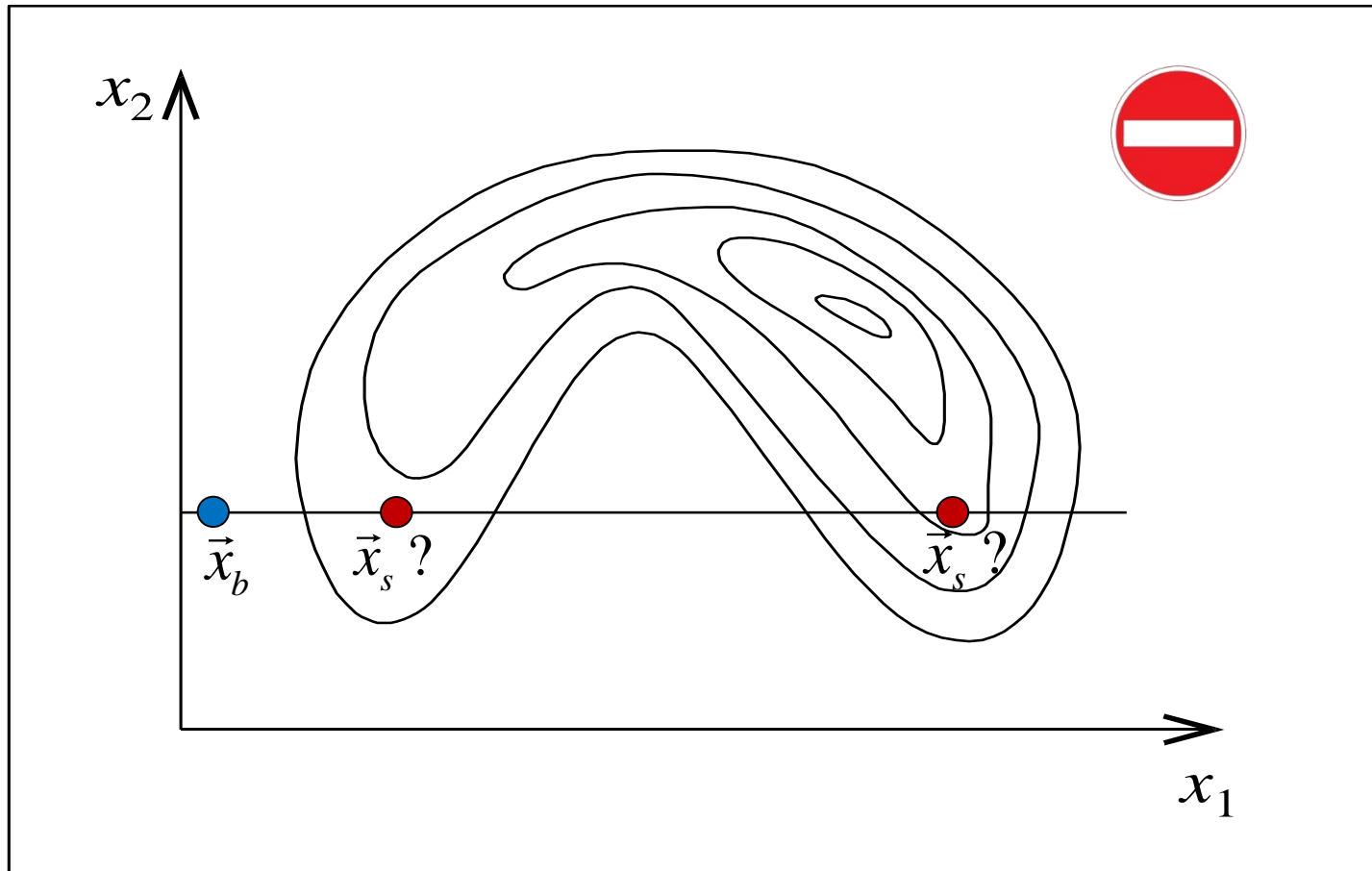
$$\text{Положить } \Delta = |\vec{x}_b - \vec{x}_s|; \quad \vec{x}_b = \vec{x}_s$$

4. Положить $\vec{x}^* = \vec{x}_s$ Завершить работу.

Нелинейное программирование.

Безусловная минимизация функций n переменных.

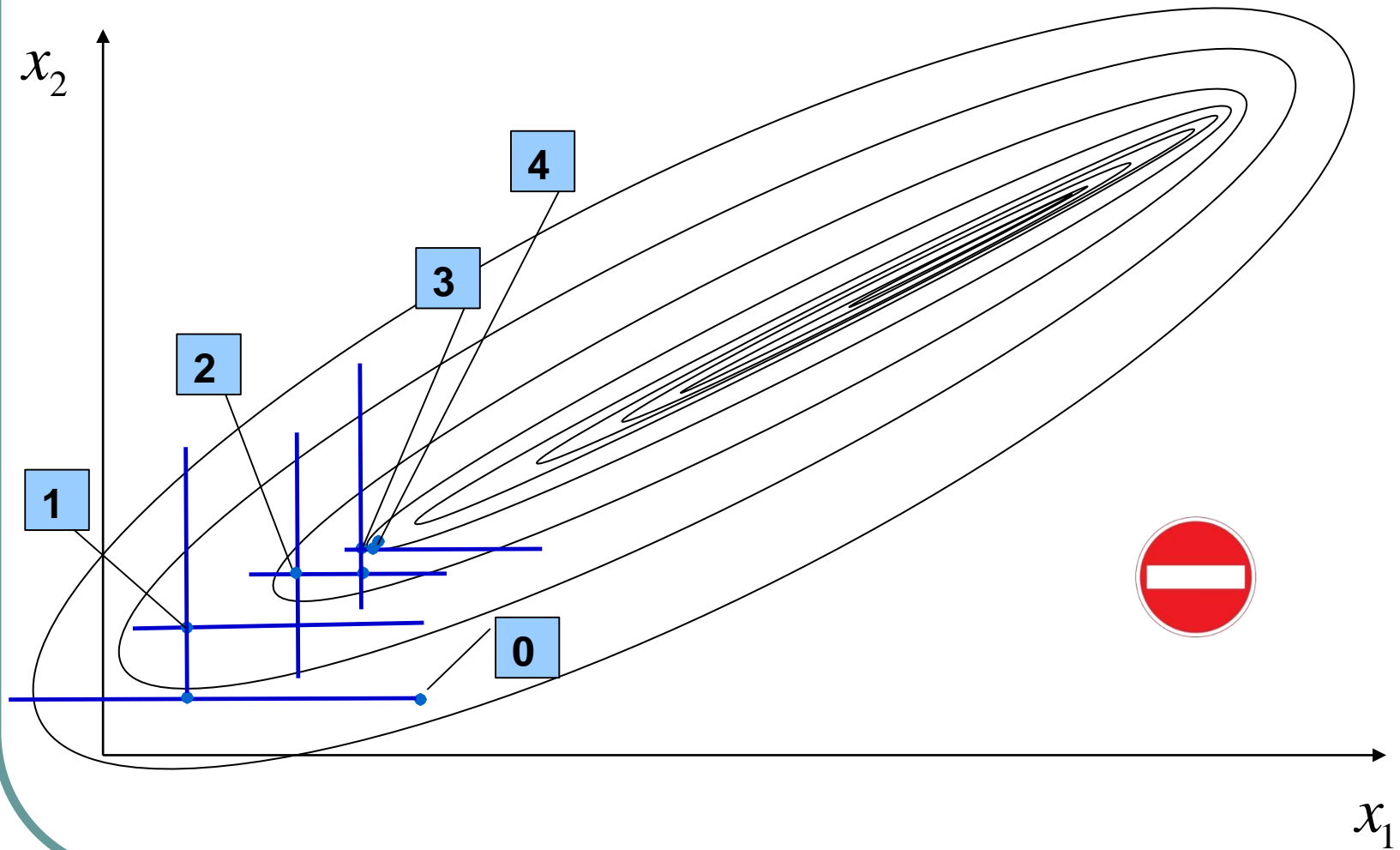
Метод покоординатного спуска



Нелинейное программирование.

Безусловная минимизация функций n переменных.

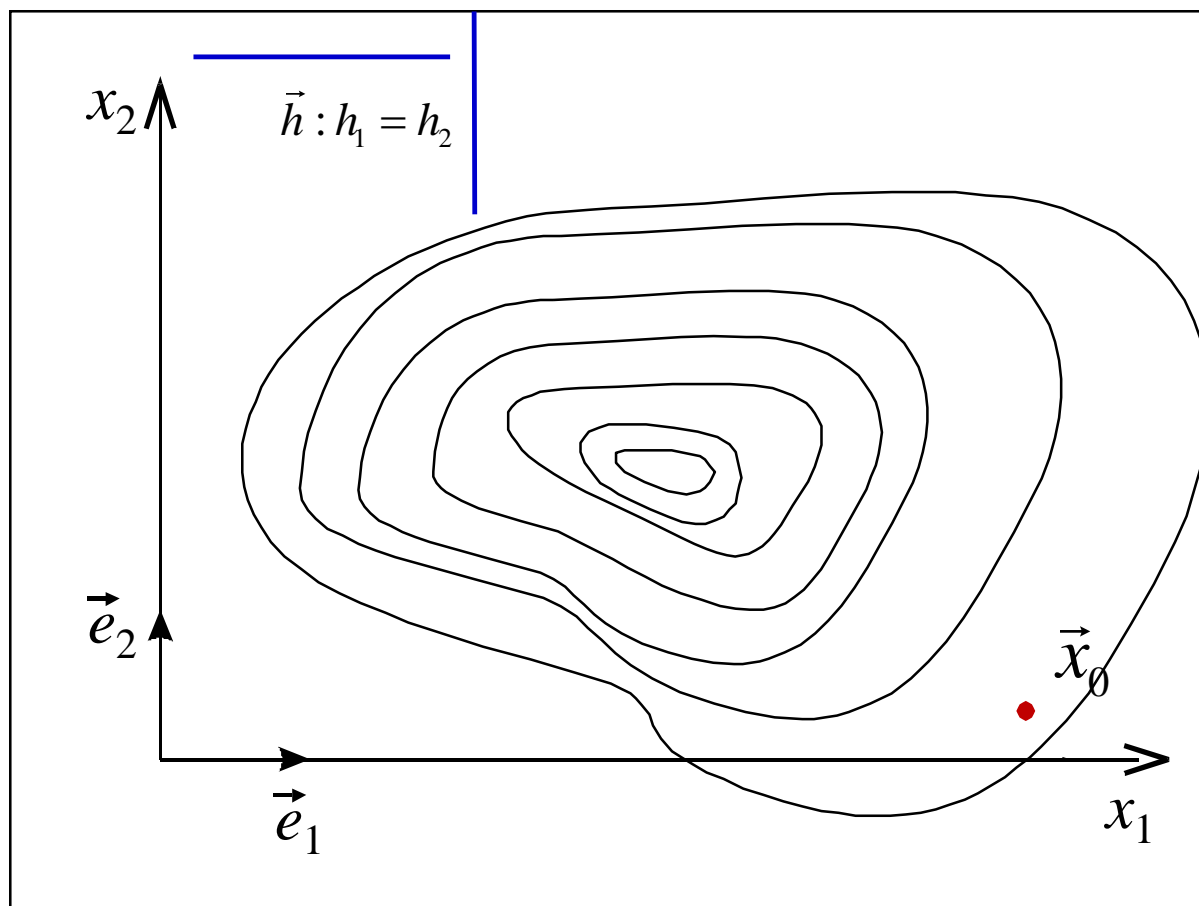
Метод покоординатного спуска



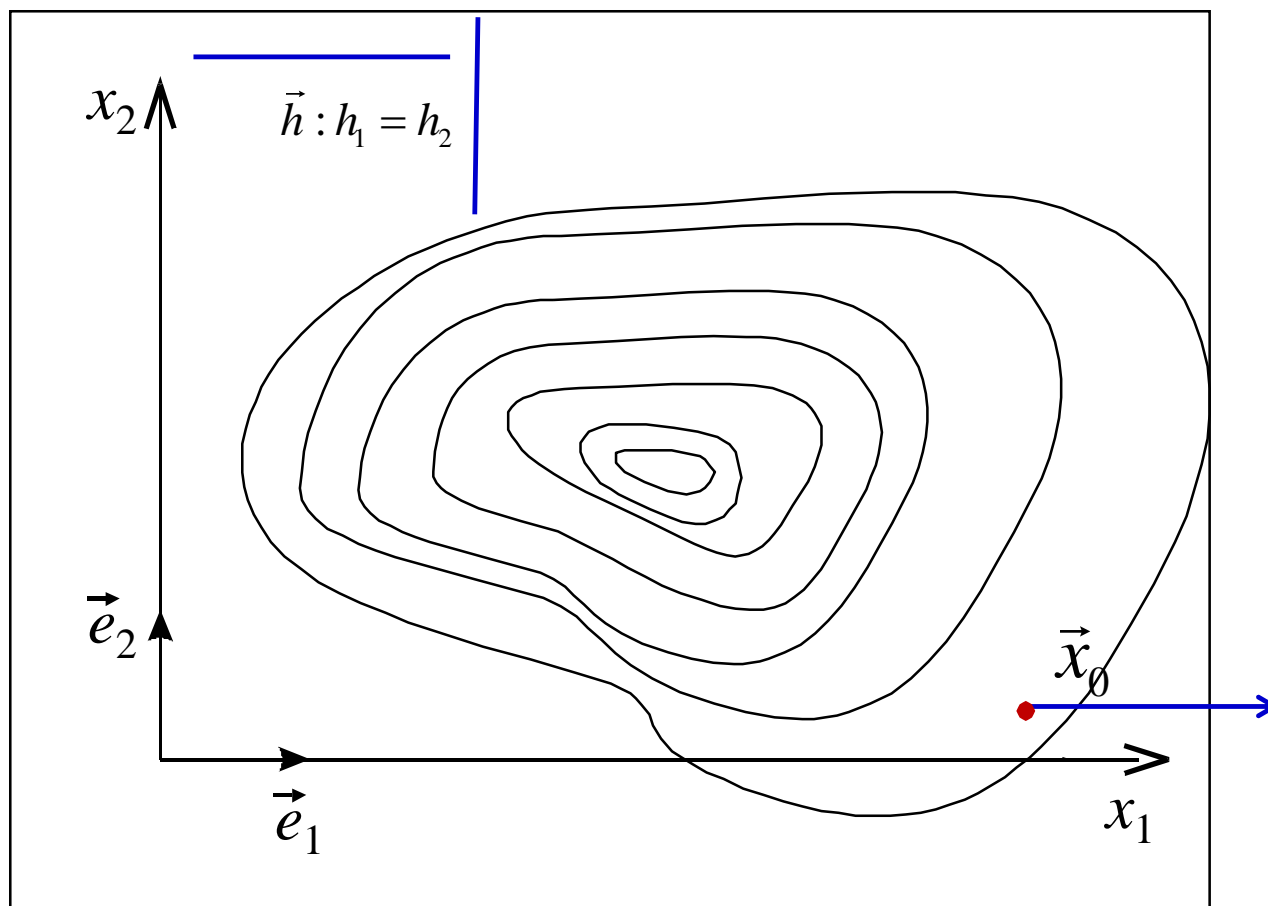
Нелинейное программирование.

Безусловная минимизация функций n переменных.

Метод локальных вариаций



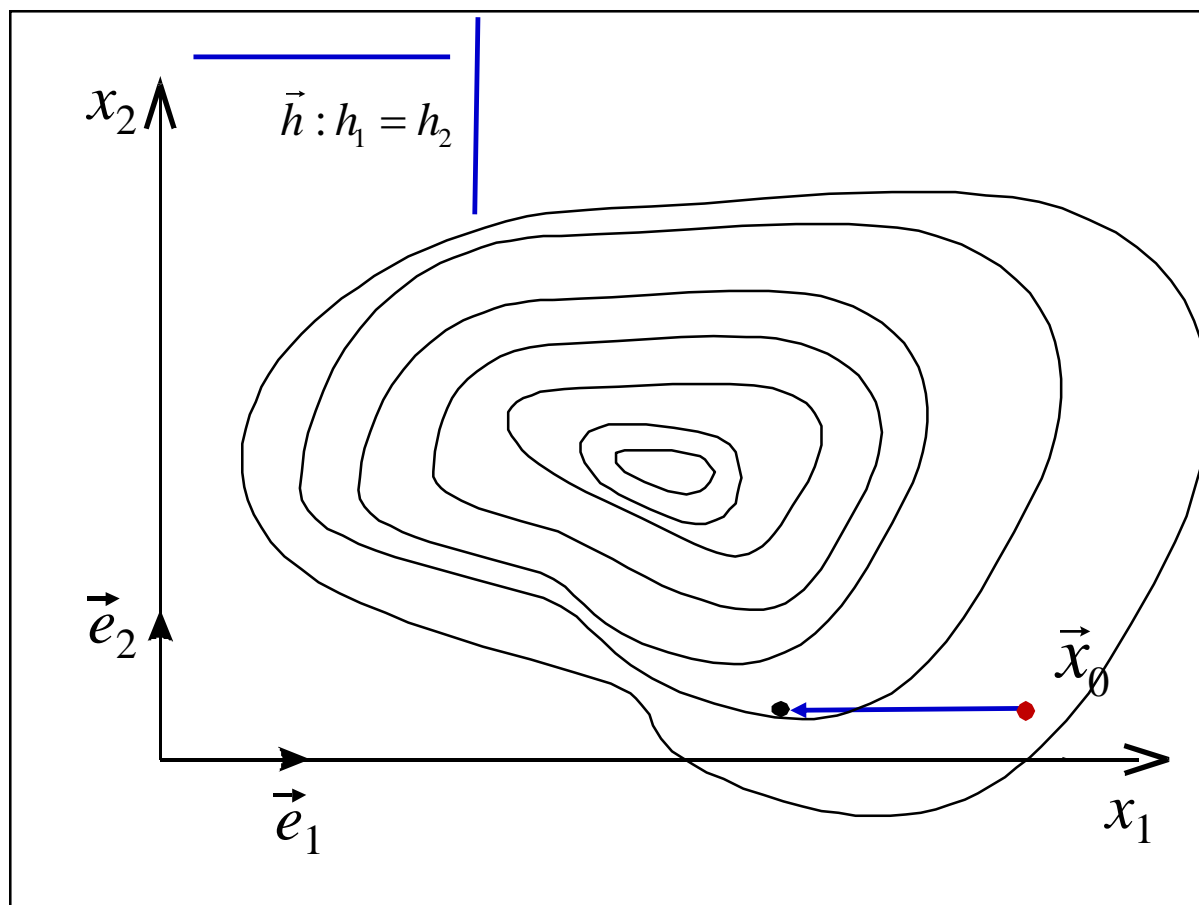
Нелинейное программирование.
Безусловная минимизация функций n переменных.
Метод локальных вариаций



Нелинейное программирование.

Безусловная минимизация функций n переменных.

Метод локальных вариаций

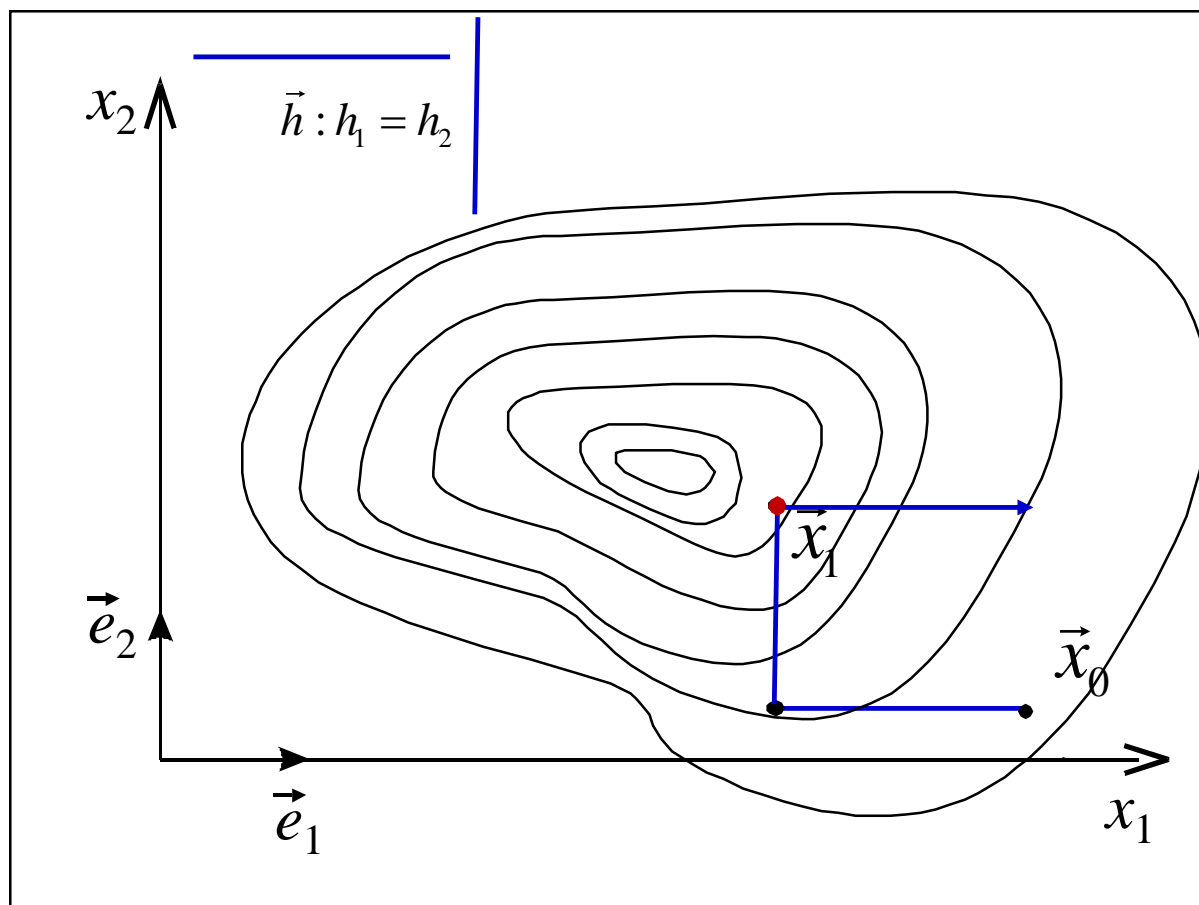


Нелинейное программирование.
Безусловная минимизация функций n переменных.
Метод локальных вариаций

Нелинейное программирование.

Безусловная минимизация функций n переменных.

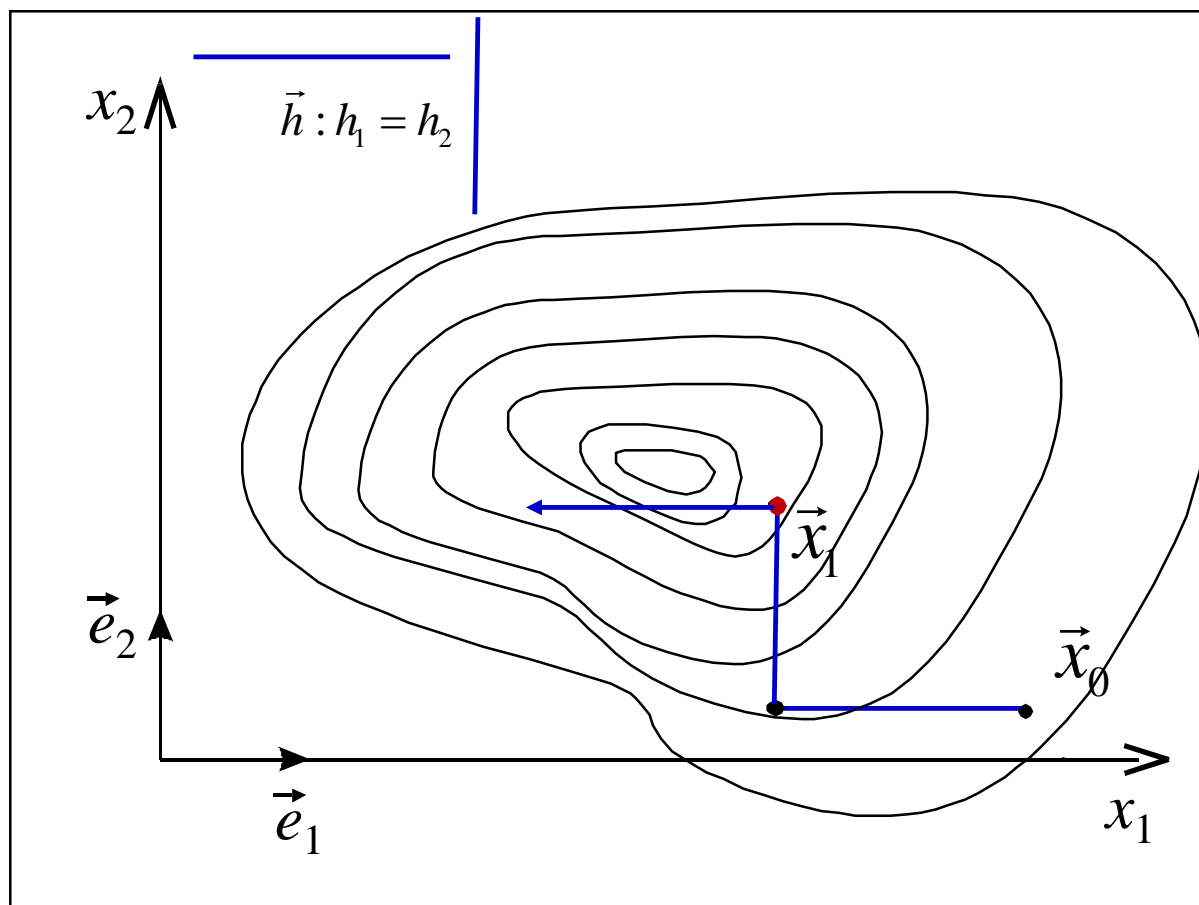
Метод локальных вариаций



Нелинейное программирование.

Безусловная минимизация функций n переменных.

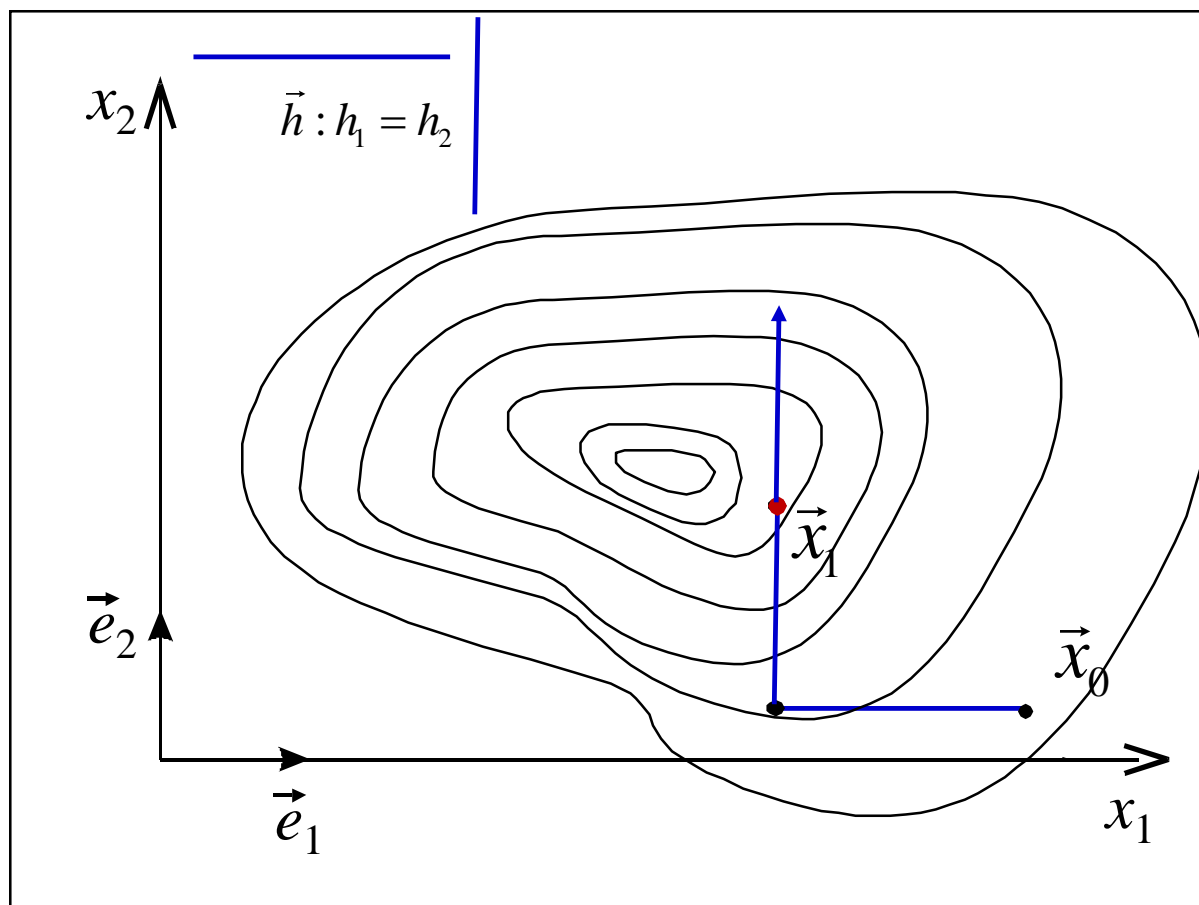
Метод локальных вариаций



Нелинейное программирование.

Безусловная минимизация функций n переменных.

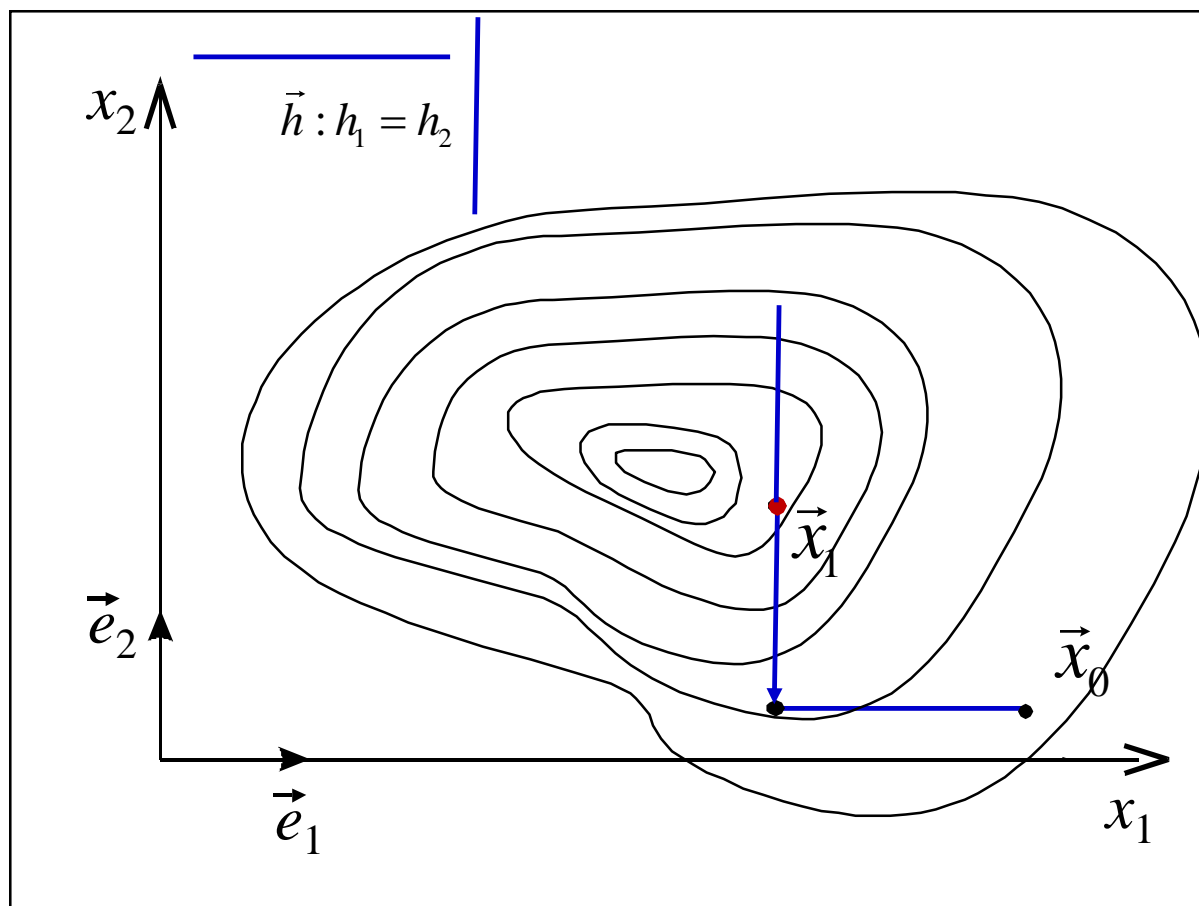
Метод локальных вариаций



Нелинейное программирование.

Безусловная минимизация функций n переменных.

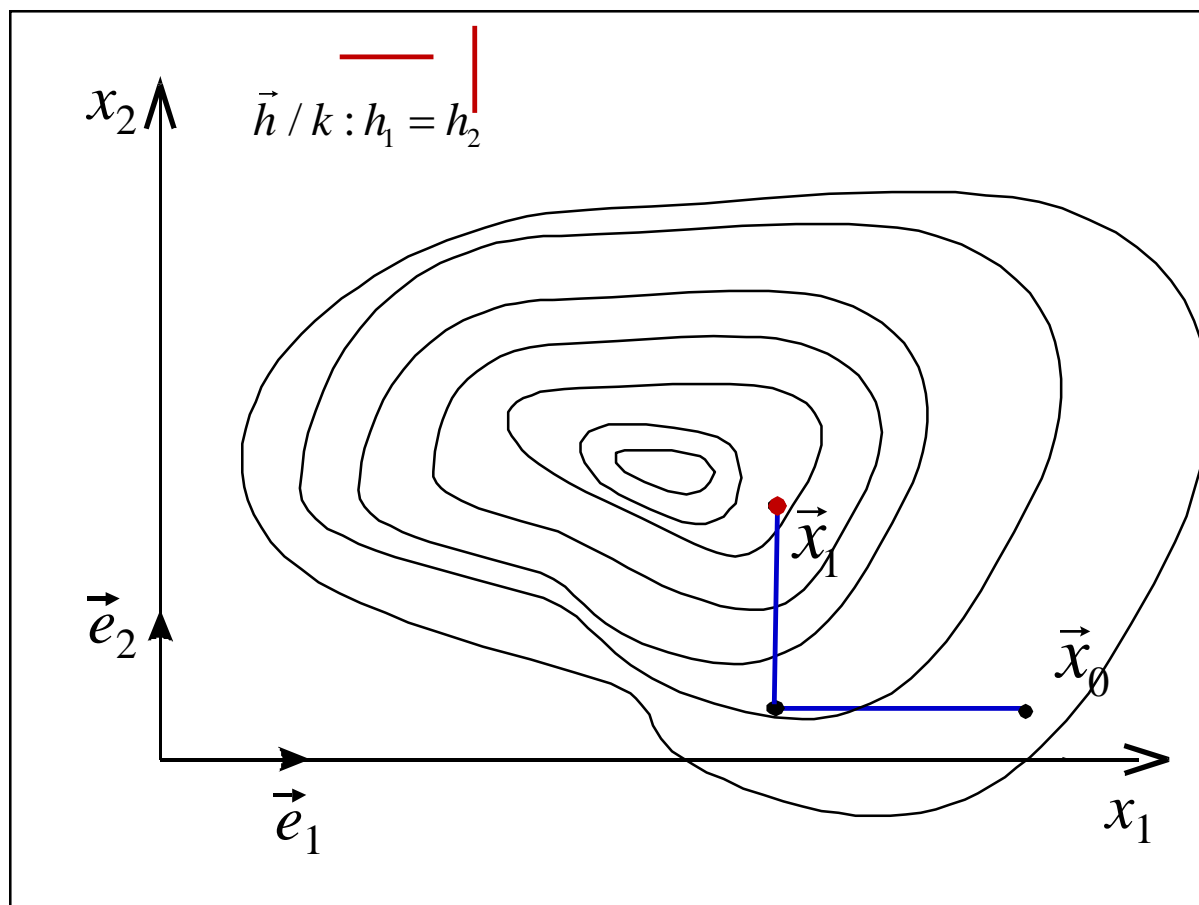
Метод локальных вариаций



Нелинейное программирование.

Безусловная минимизация функций n переменных.

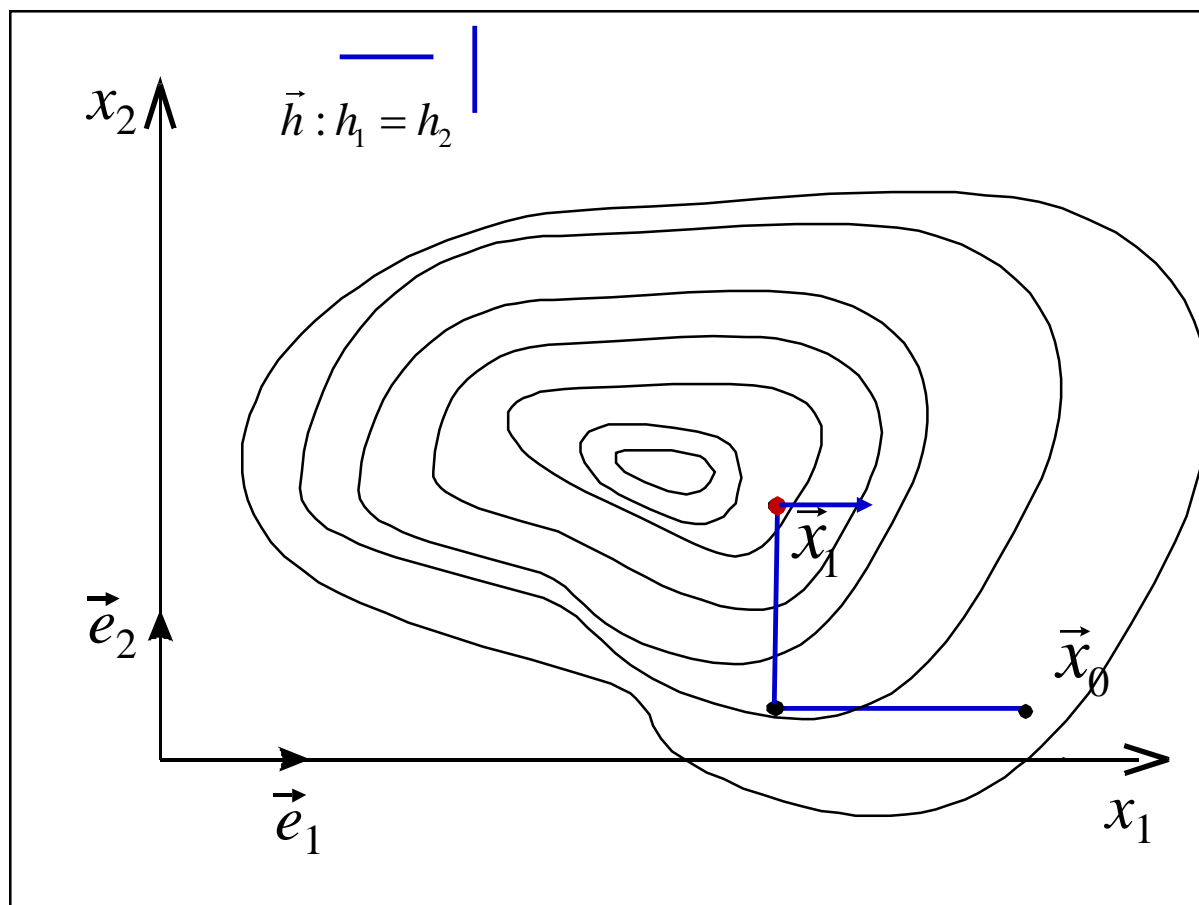
Метод локальных вариаций



Нелинейное программирование.

Безусловная минимизация функций n переменных.

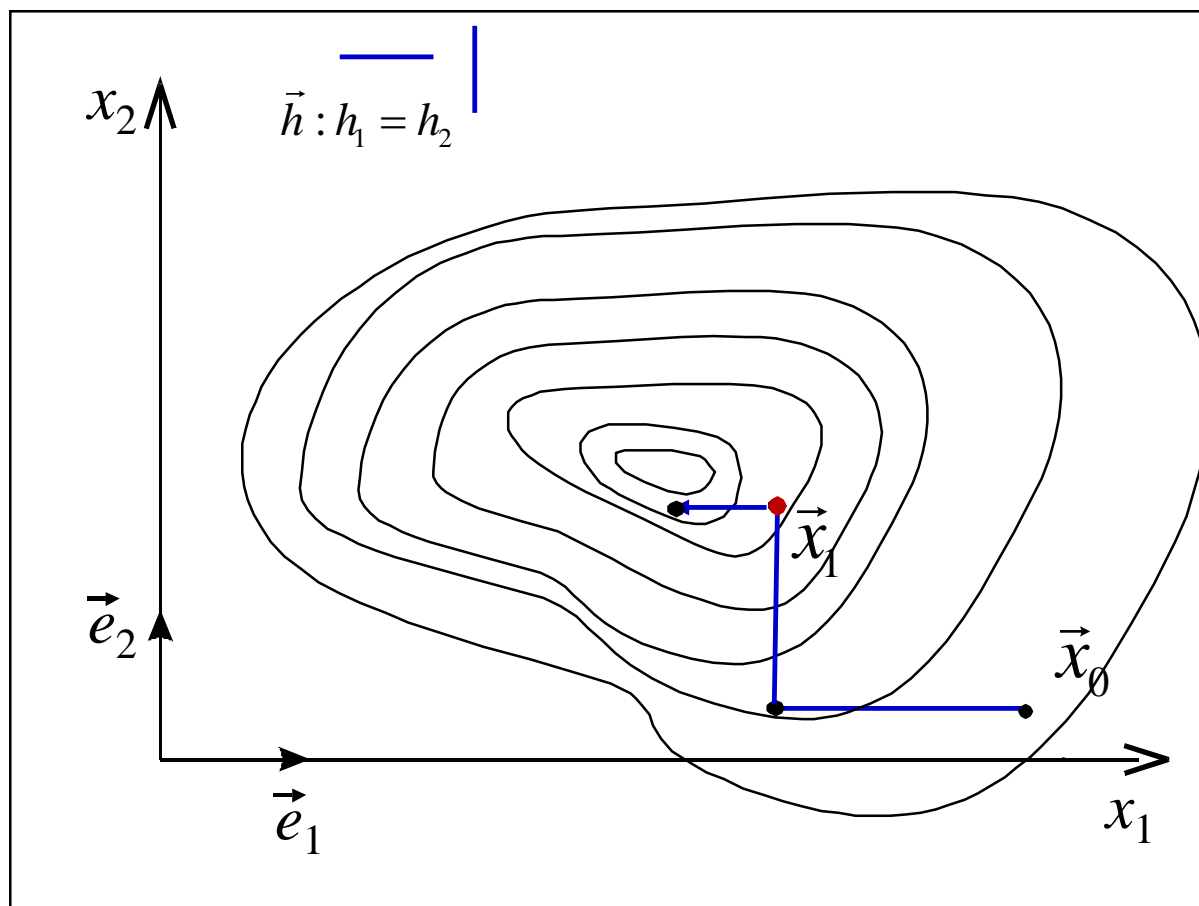
Метод локальных вариаций



Нелинейное программирование.

Безусловная минимизация функций n переменных.

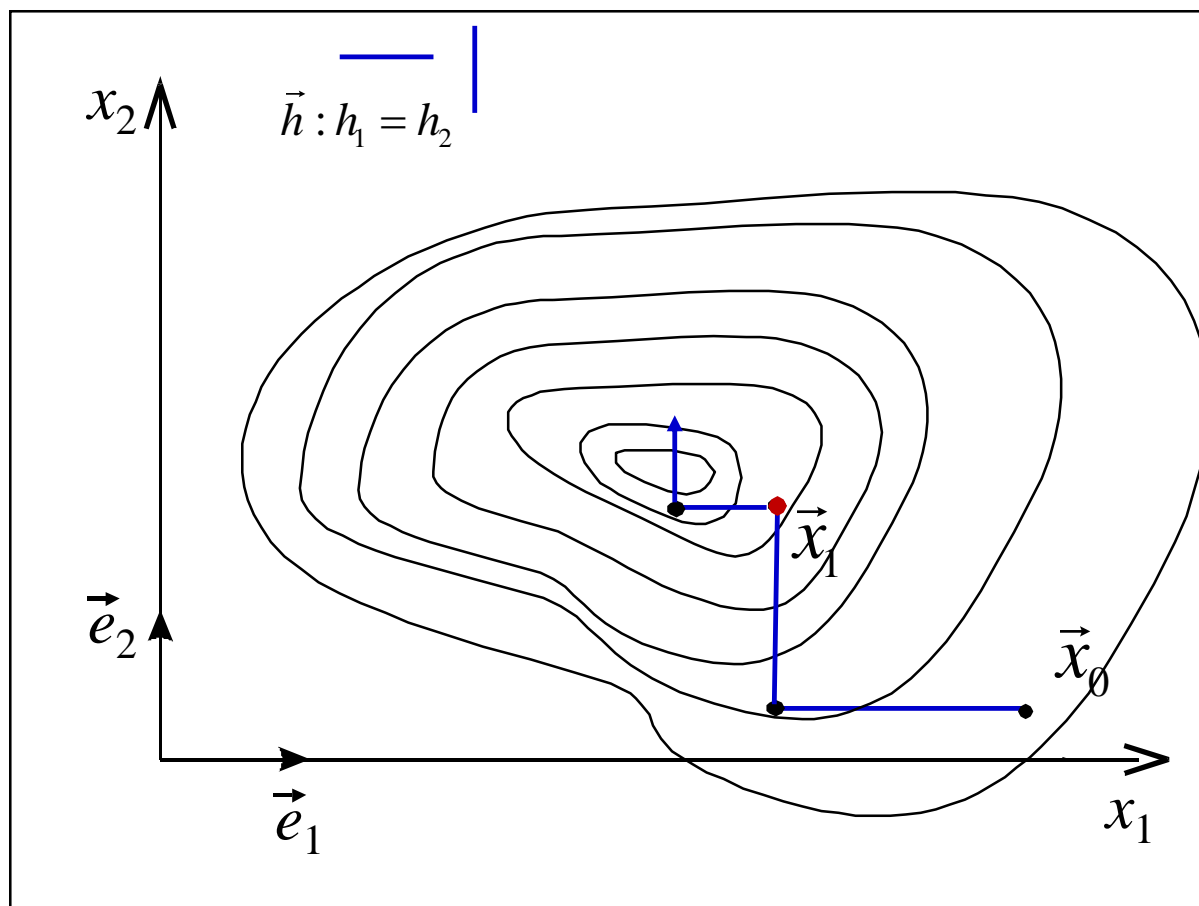
Метод локальных вариаций



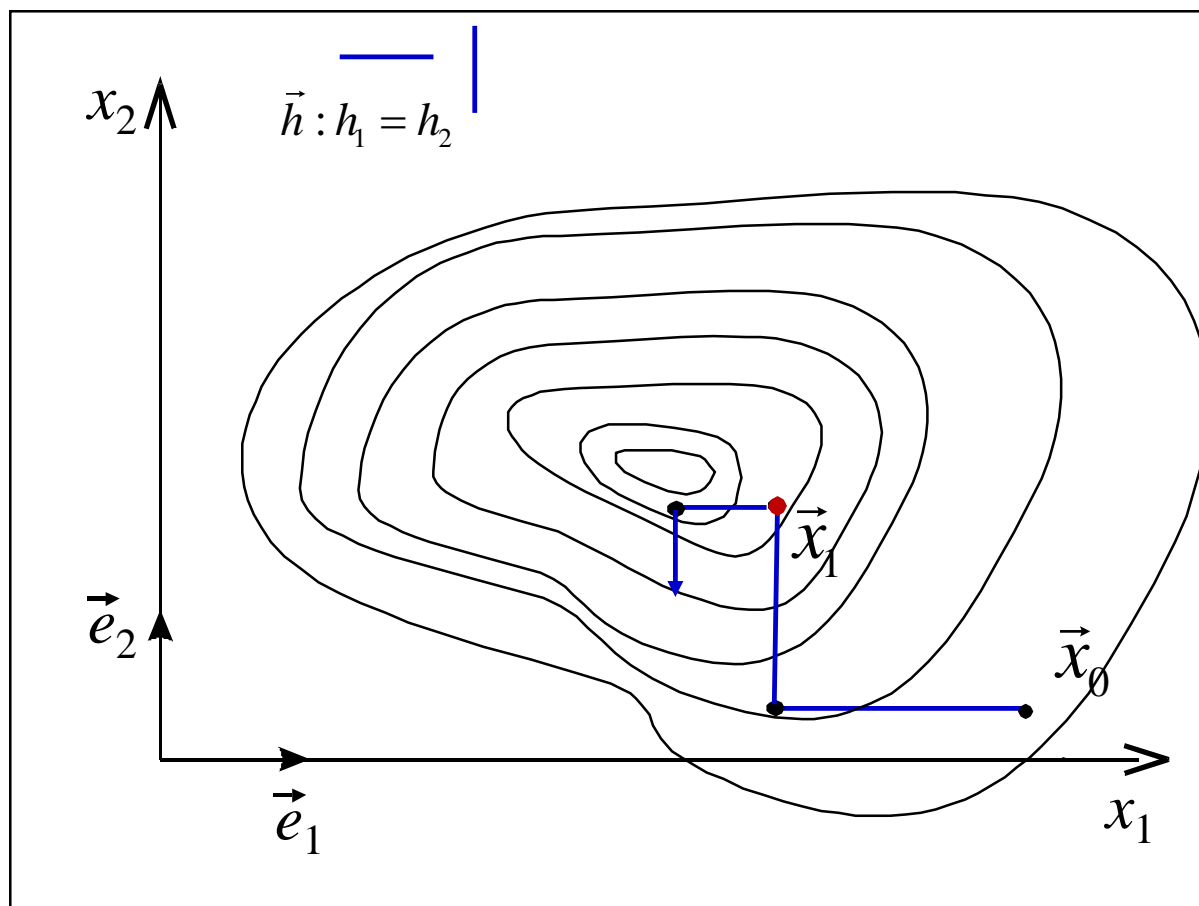
Нелинейное программирование.

Безусловная минимизация функций n переменных.

Метод локальных вариаций



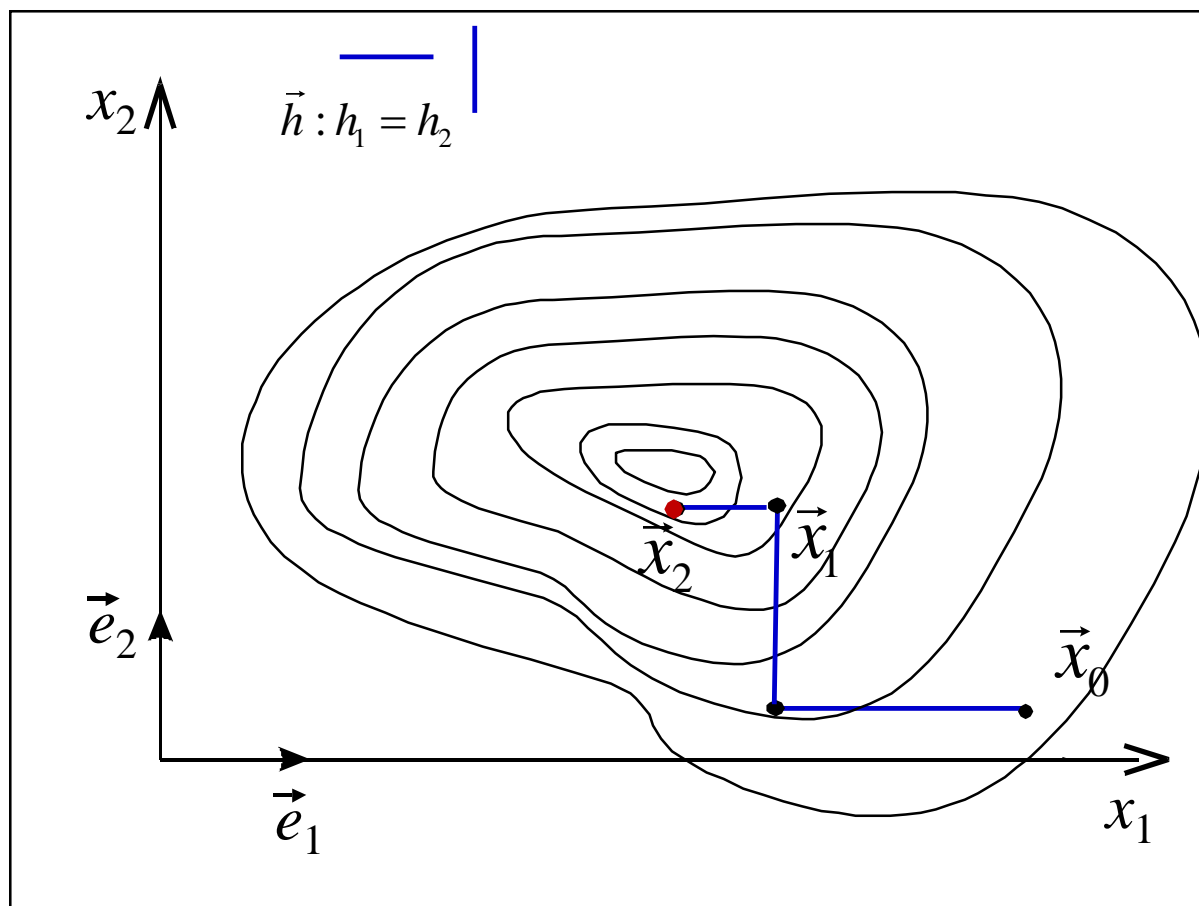
Нелинейное программирование.
Безусловная минимизация функций n переменных.
Метод локальных вариаций



Нелинейное программирование.

Безусловная минимизация функций n переменных.

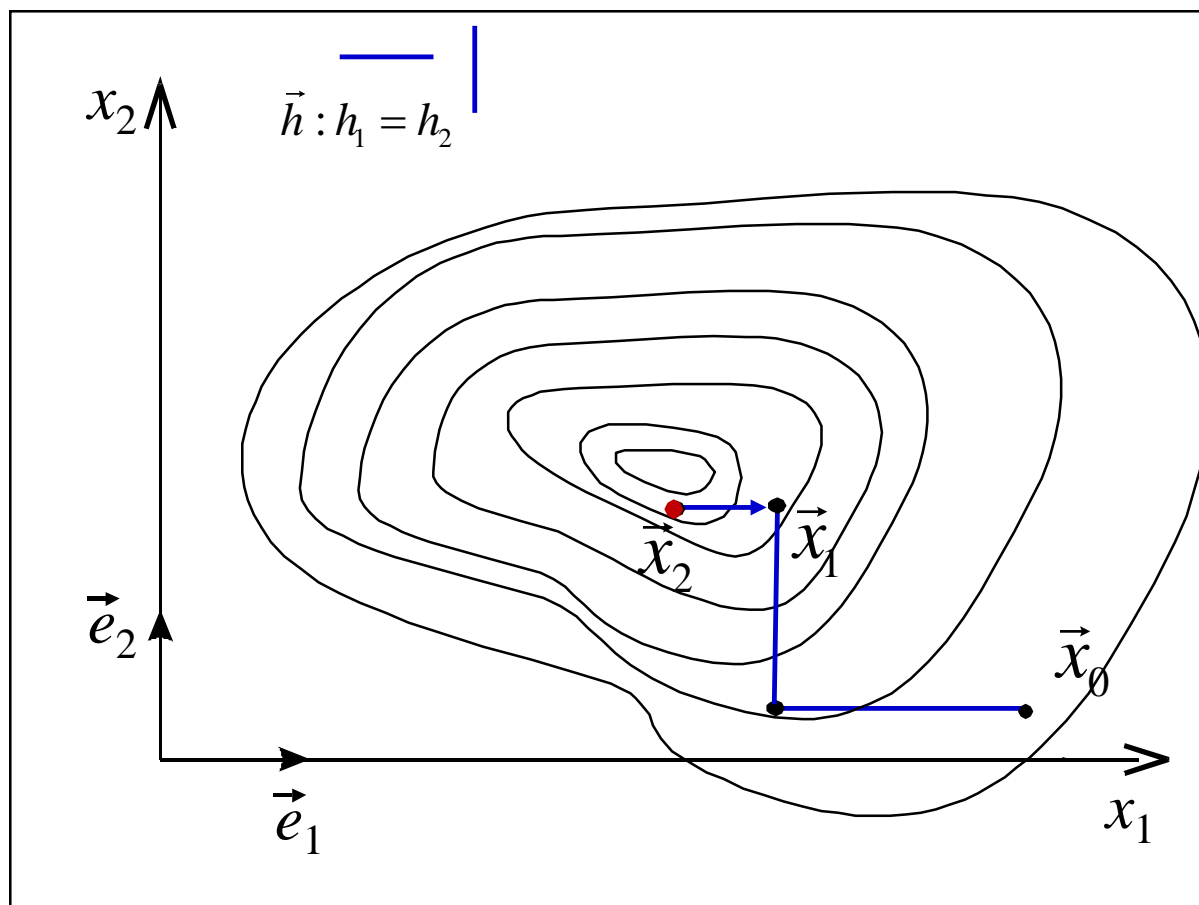
Метод локальных вариаций



Нелинейное программирование.

Безусловная минимизация функций n переменных.

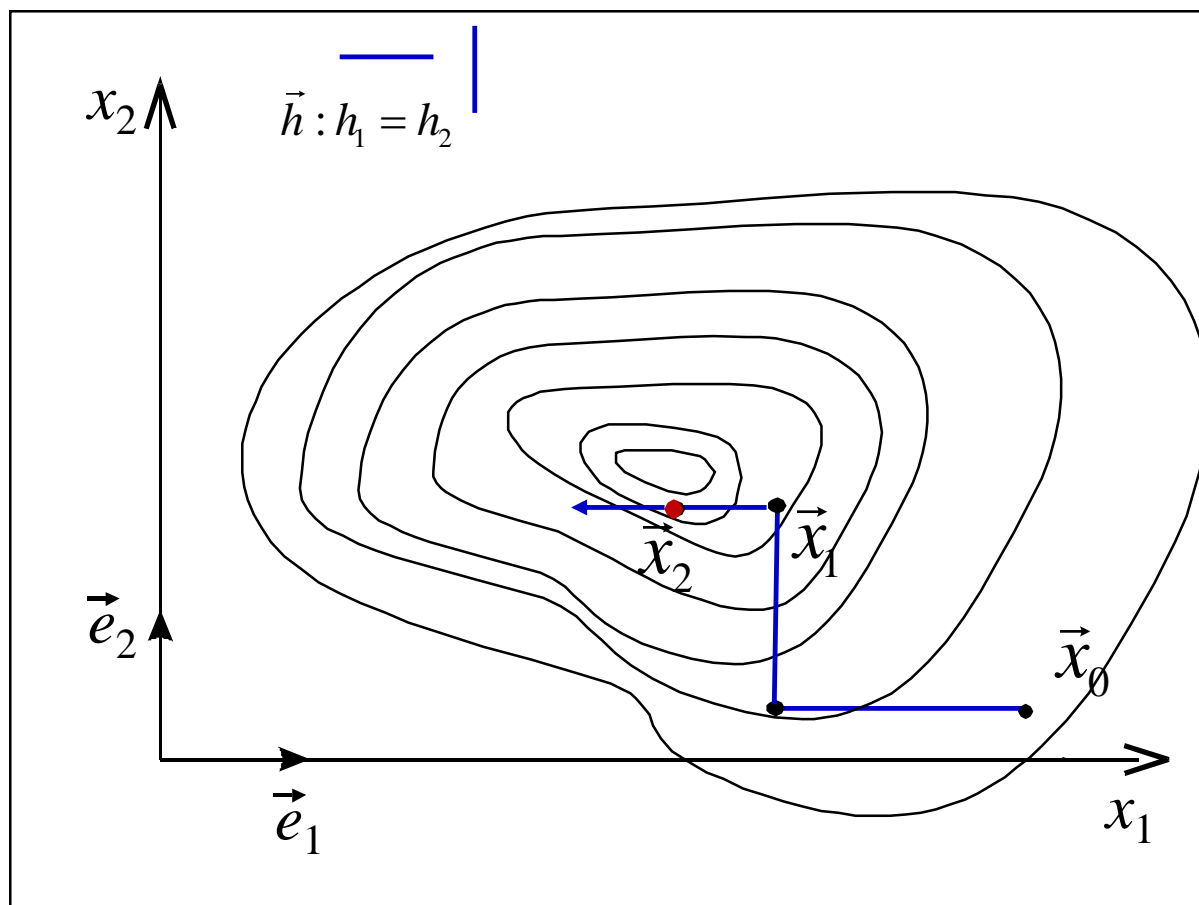
Метод локальных вариаций



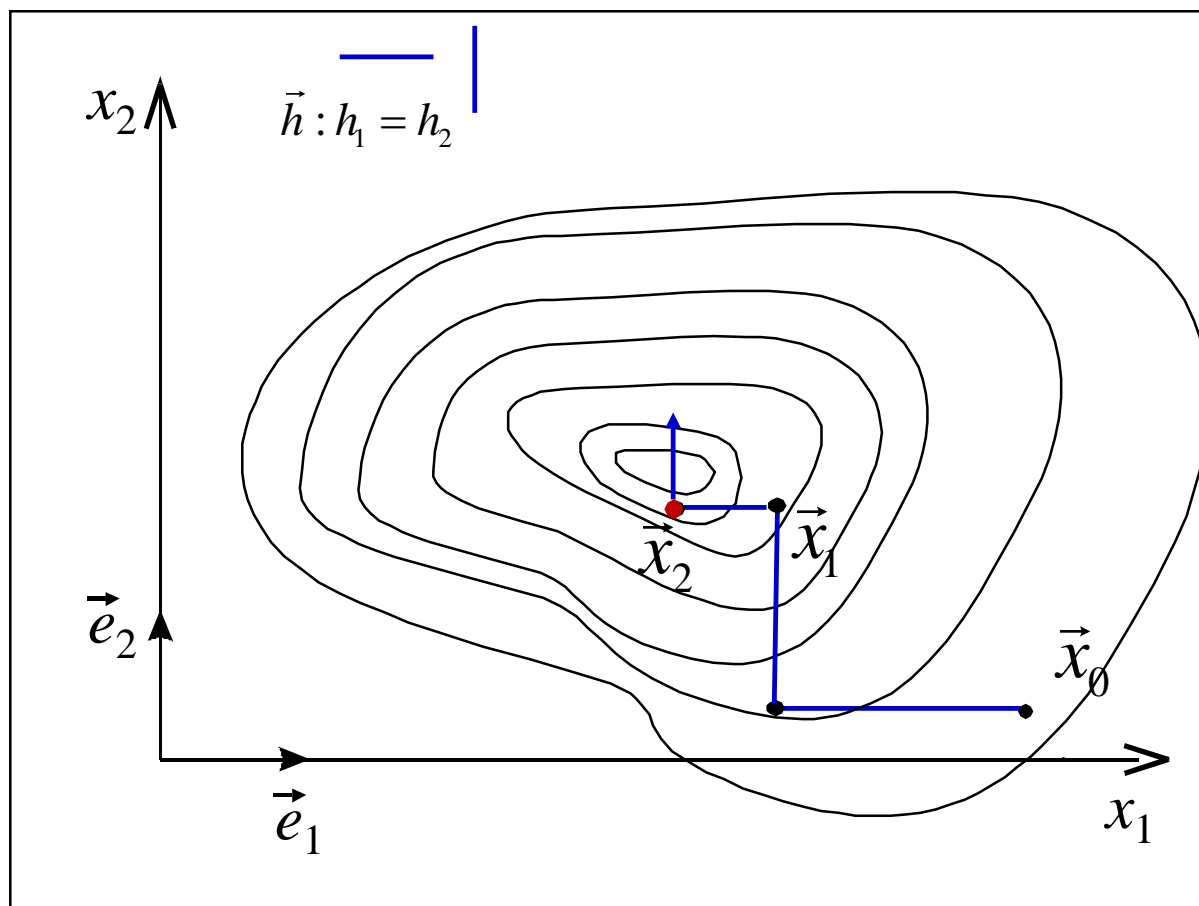
Нелинейное программирование.

Безусловная минимизация функций n переменных.

Метод локальных вариаций



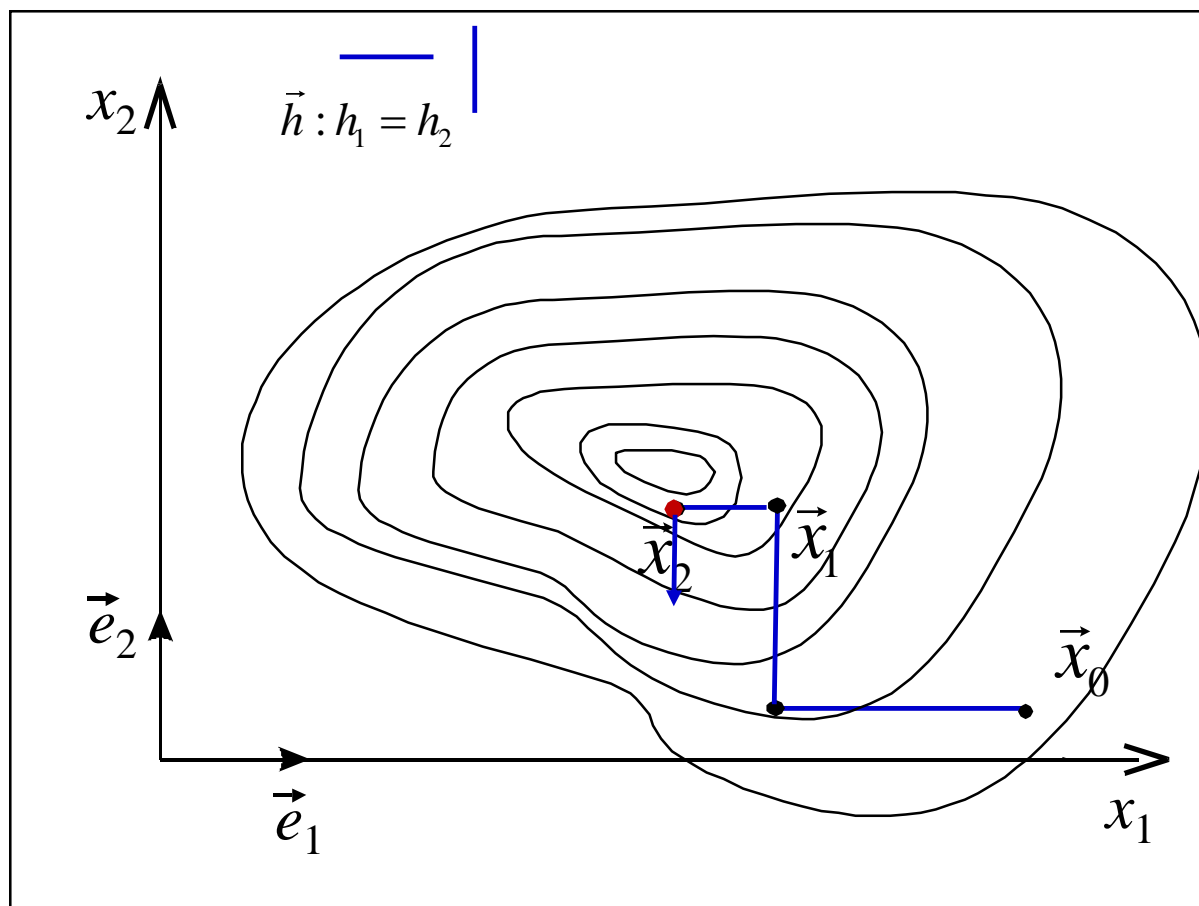
Нелинейное программирование.
Безусловная минимизация функций n переменных.
Метод локальных вариаций



Нелинейное программирование.

Безусловная минимизация функций n переменных.

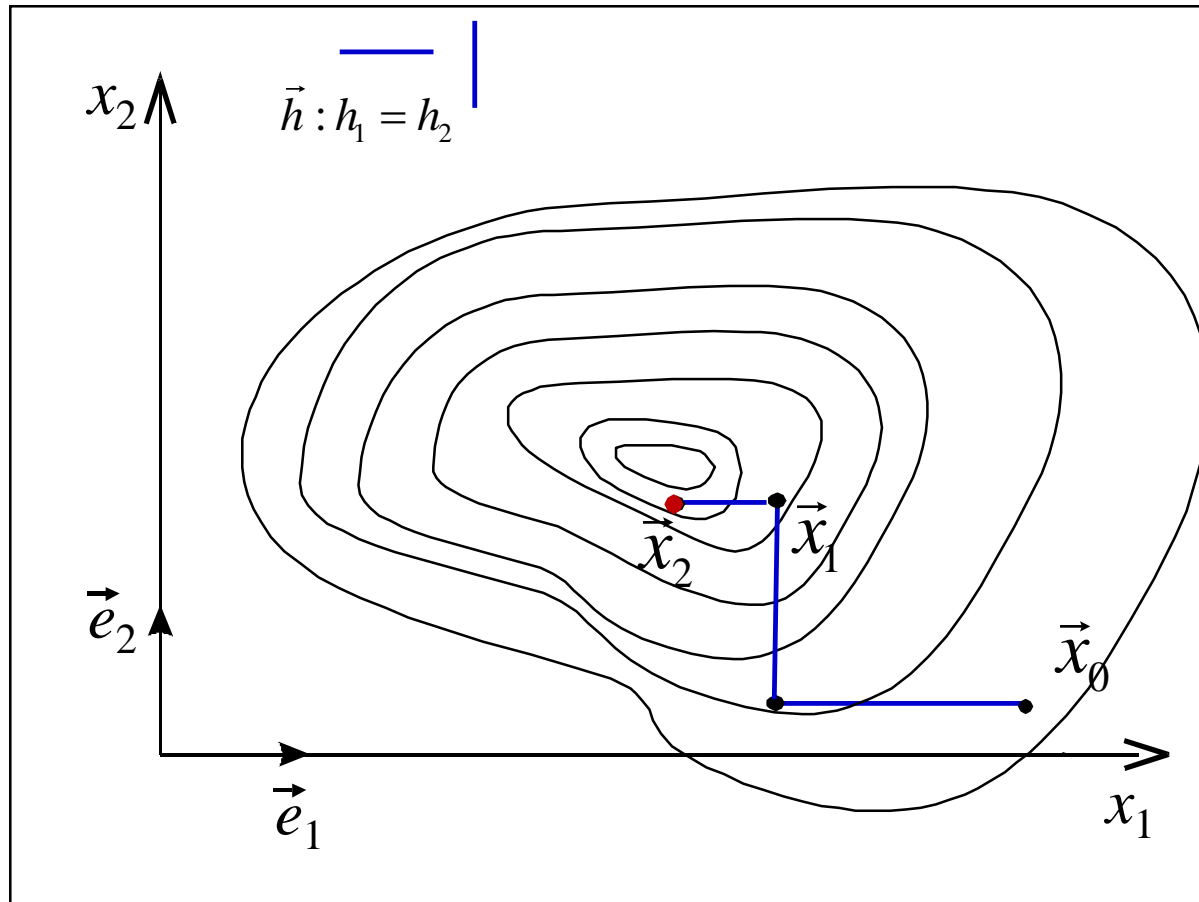
Метод локальных вариаций



6. Нелинейное программирование.

6.8. Безусловная минимизация функций n переменных.

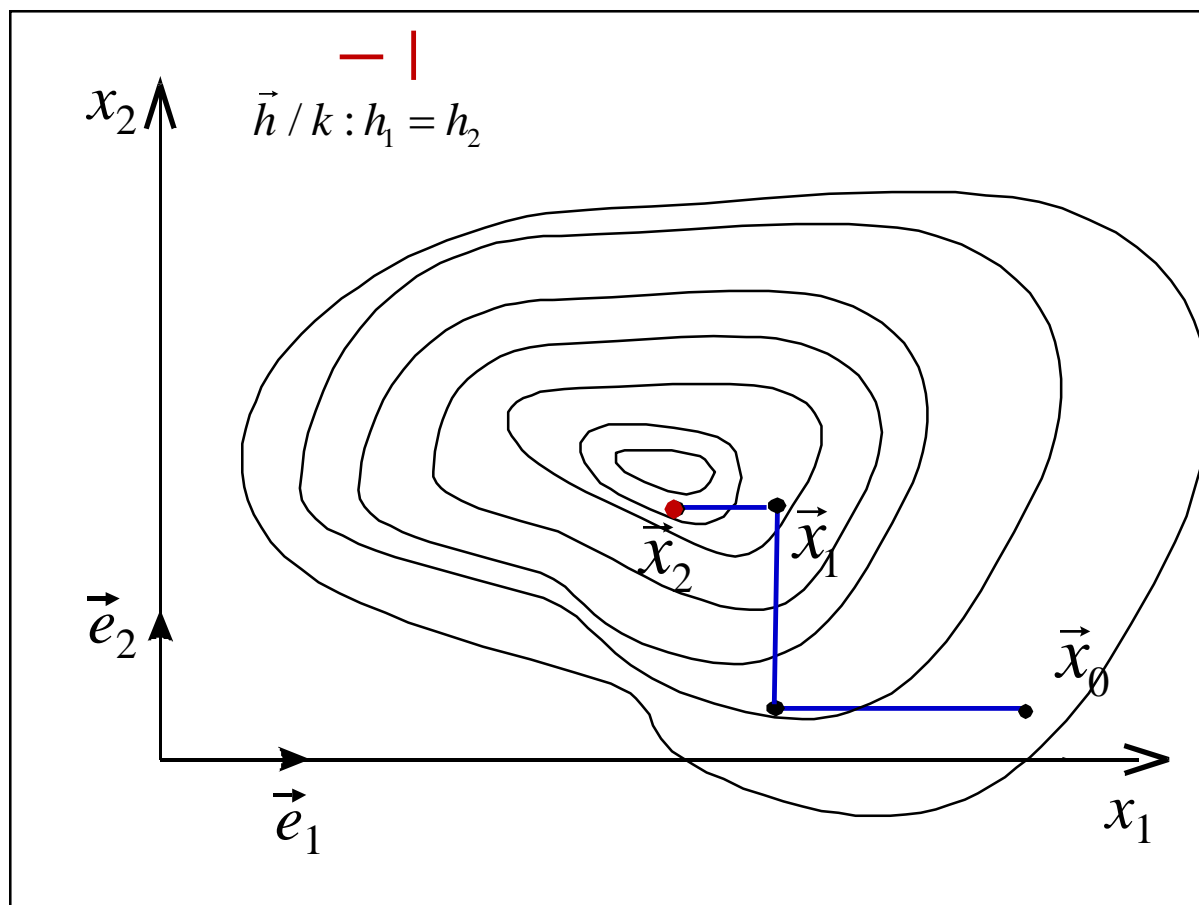
Метод локальных вариаций



Нелинейное программирование.

Безусловная минимизация функций n переменных.

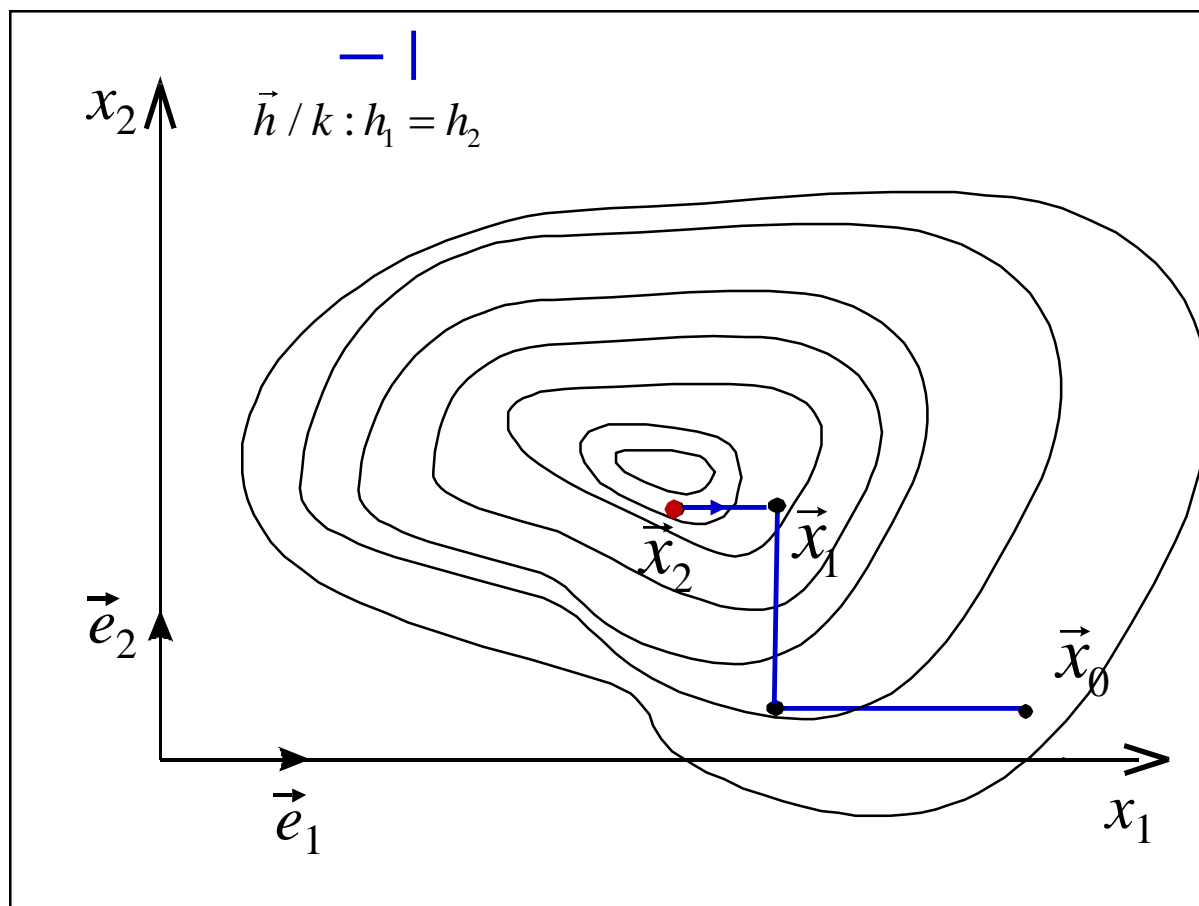
Метод локальных вариаций



Нелинейное программирование.

Безусловная минимизация функций n переменных.

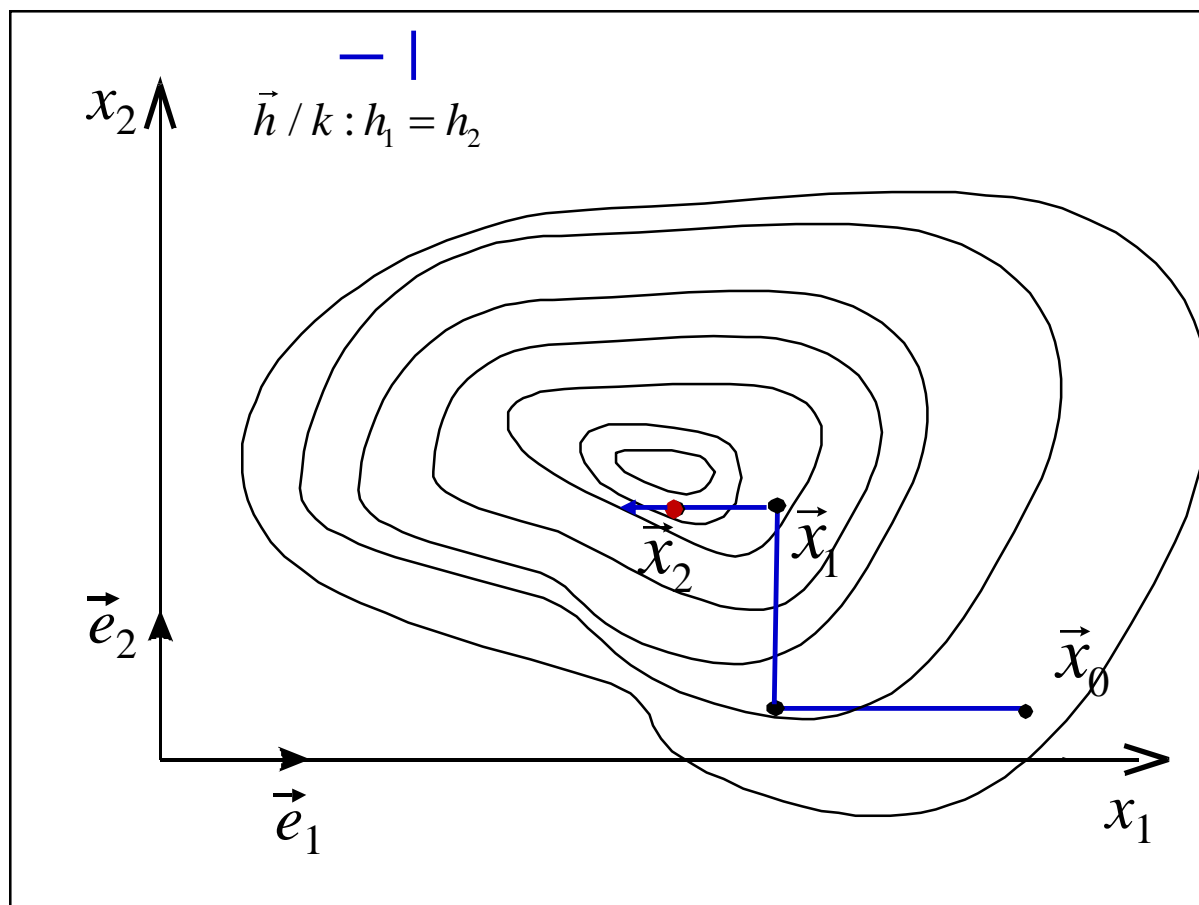
Метод локальных вариаций



Нелинейное программирование.

Безусловная минимизация функций n переменных.

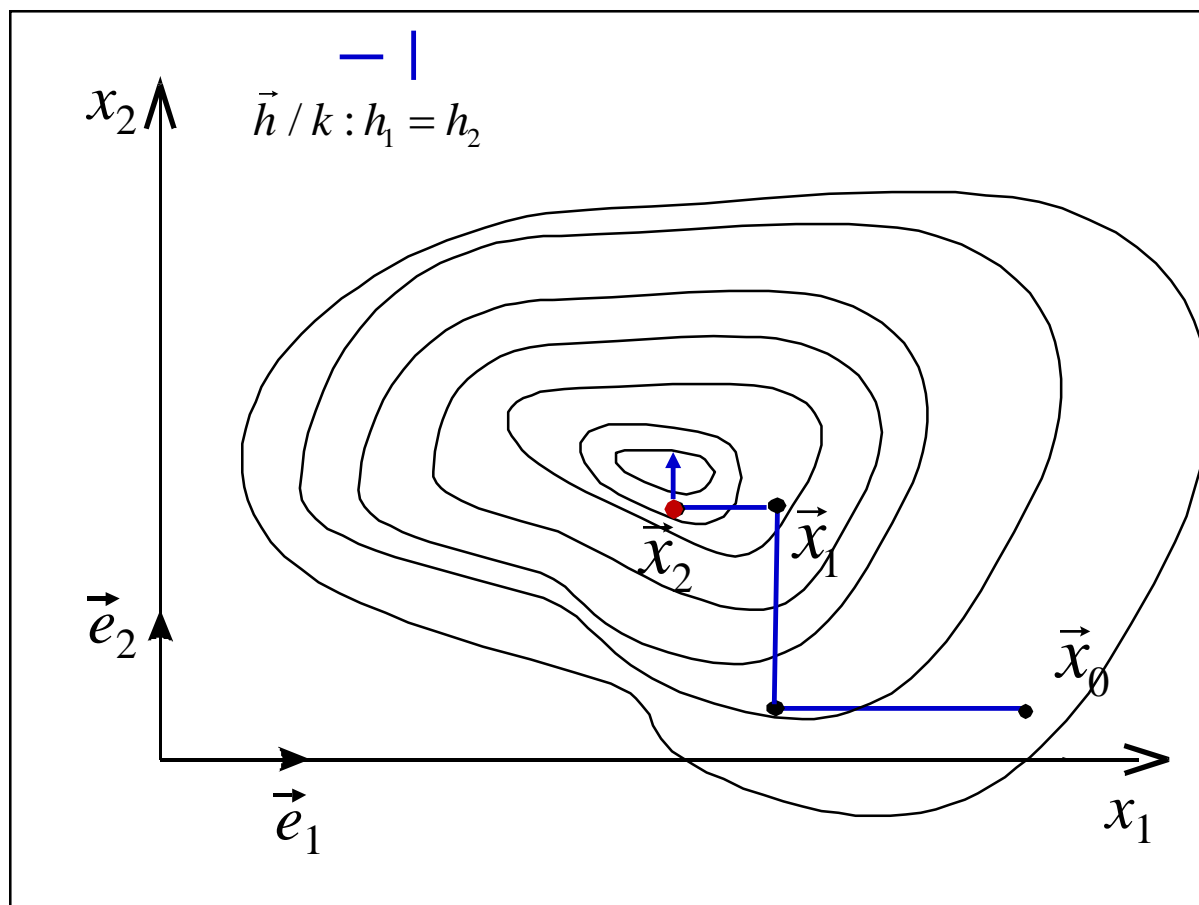
Метод локальных вариаций



Нелинейное программирование.

Безусловная минимизация функций n переменных.

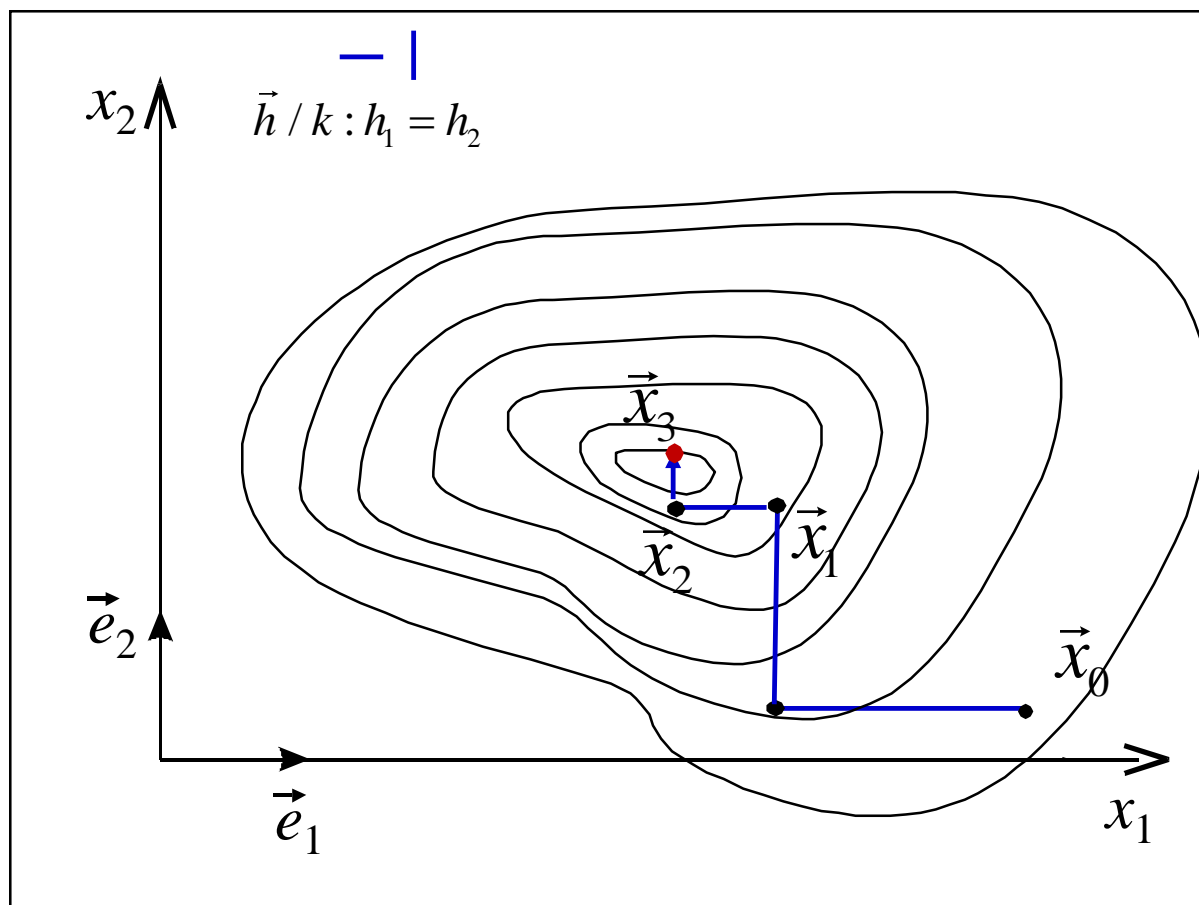
Метод локальных вариаций



Нелинейное программирование.

Безусловная минимизация функций n переменных.

Метод локальных вариаций



Нелинейное программирование.

Безусловная минимизация функций n переменных.

Метод локальных вариаций

1. *Выбрать \vec{x}_0 , задать \vec{h}, ε ;*

2. $\vec{x}_b = \vec{x}_0$; $\Delta := |\vec{h}|$;

3. *Пока $\Delta > \varepsilon$ выполнить*

Покоординатный поиск

из \vec{x}_b с шагом \vec{h} и результатом \vec{x}_s

Если $\vec{x}_s \neq \vec{x}_b$, то $\vec{x}_b = \vec{x}_s$

иначе $\vec{h} = \vec{h} / 10$; $\Delta = \Delta / 10$;

4. *Положить $\vec{x}^* = \vec{x}_s$; Завершить работу.*

Нелинейное программирование.

Безусловная минимизация функций n переменных.

Метод локальных вариаций

Покоординатный поиск из $\vec{x}_b \rightarrow \vec{x}_s$ с шагом \vec{h}

1. $\vec{x}_s = \vec{x}_b$;

2. Для всех i от 1 до n с шагом 1 выполнить

$$\vec{x}_r = \vec{x}_s + h_i \vec{e}_i;$$

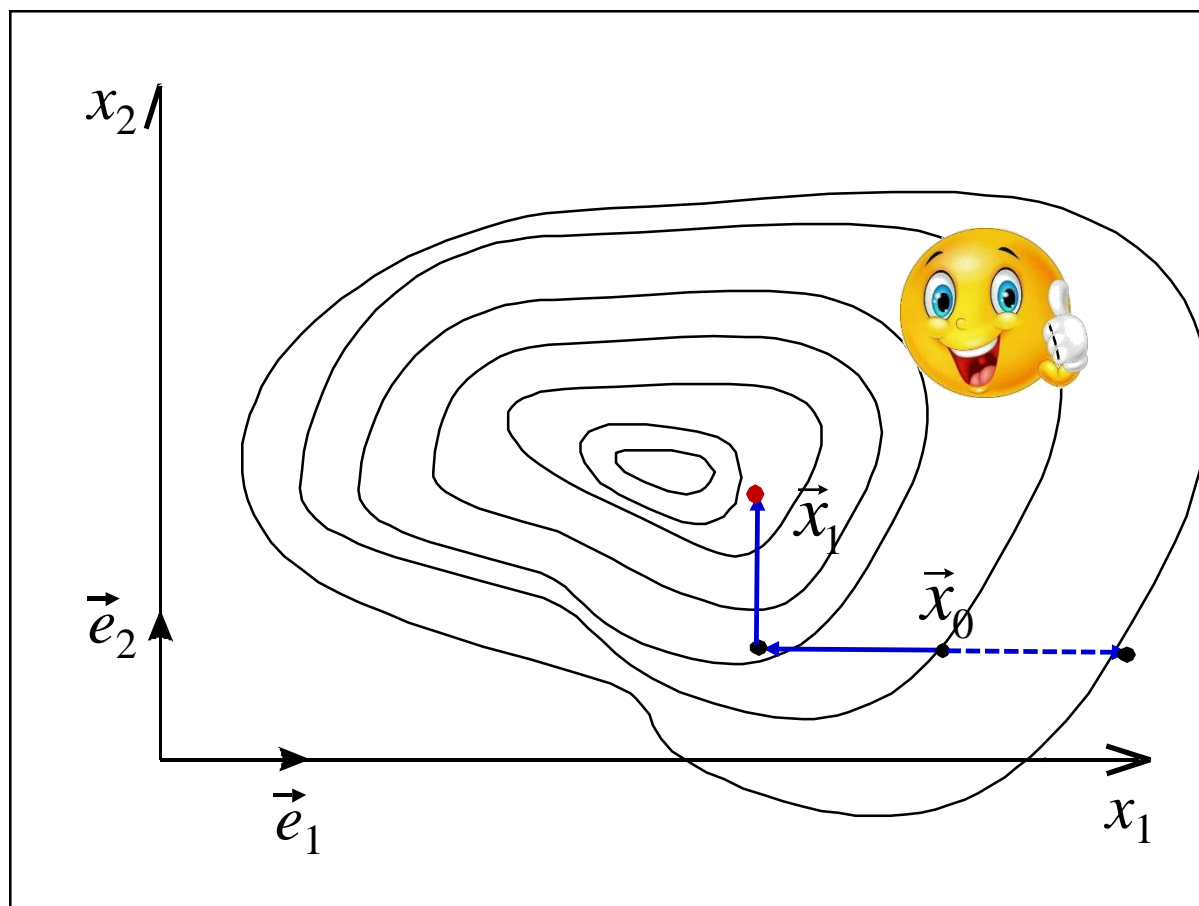
Если $F(\vec{x}_r) \geq F(\vec{x}_s)$, то $\vec{x}_r = \vec{x}_s - h_i \vec{e}_i$;

Если $F(\vec{x}_r) < F(\vec{x}_s)$, то $\vec{x}_s = \vec{x}_r$;

Нелинейное программирование.

Безусловная минимизация функций n переменных.

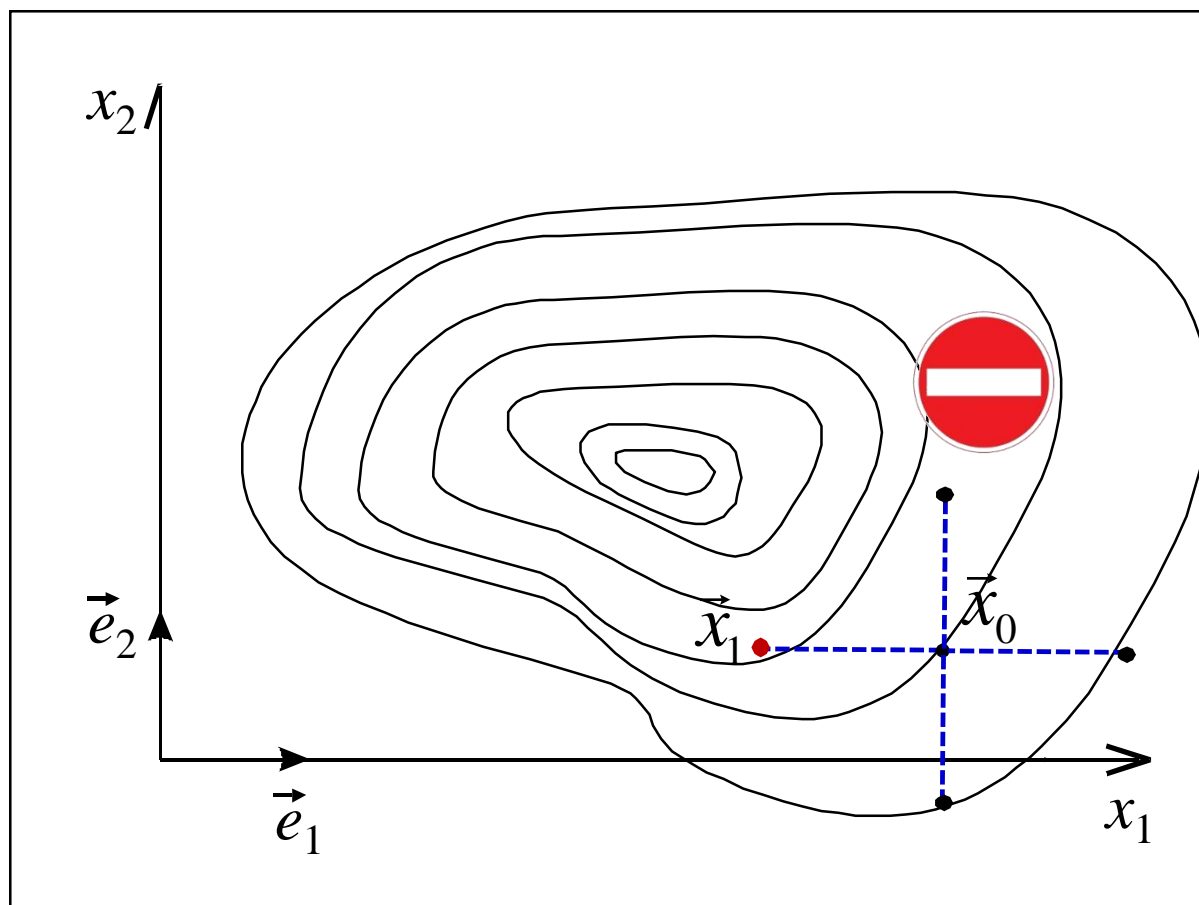
Метод локальных вариаций



Нелинейное программирование.

Безусловная минимизация функций n переменных.

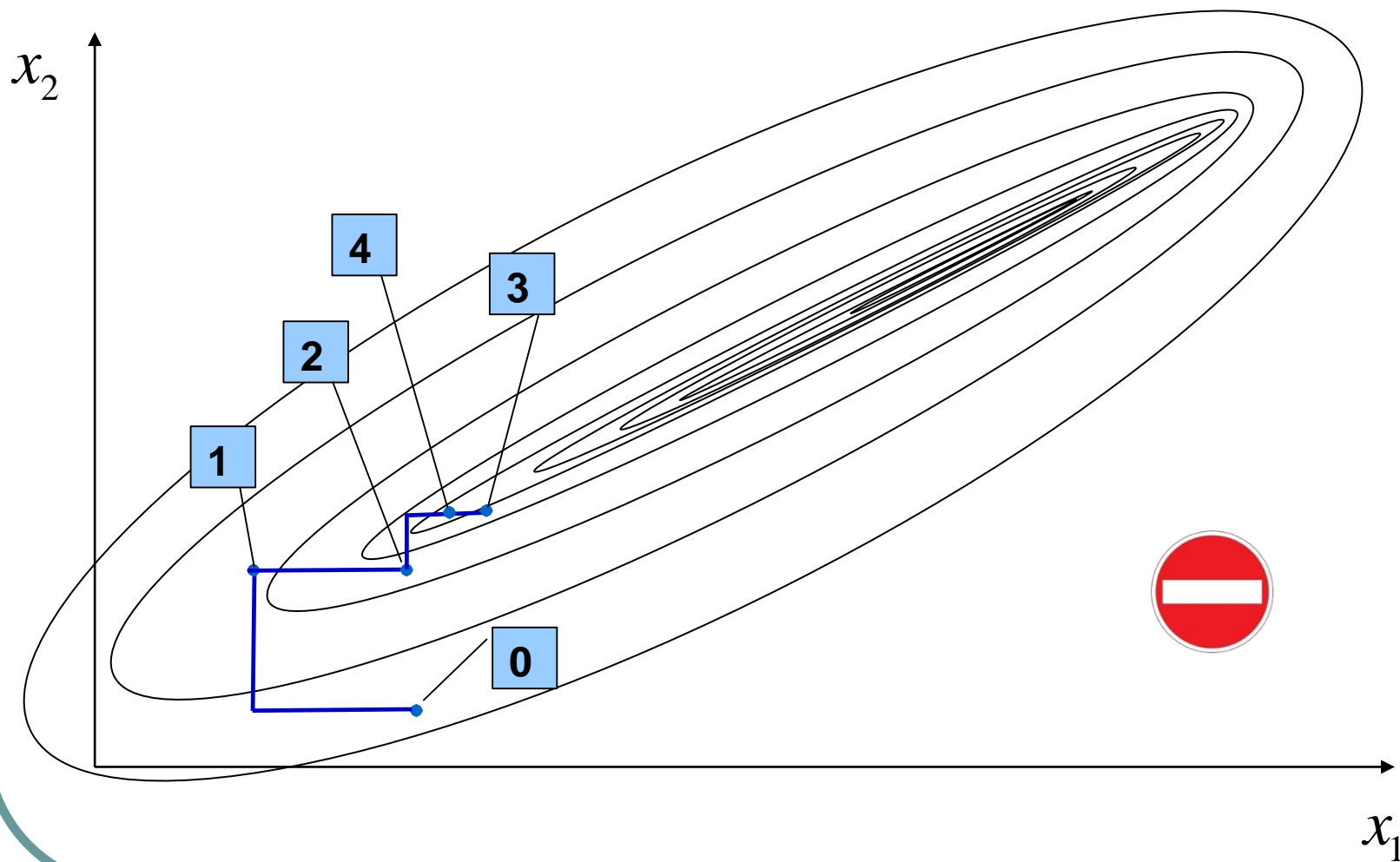
Метод локальных вариаций



Нелинейное программирование.

Безусловная минимизация функций n переменных.

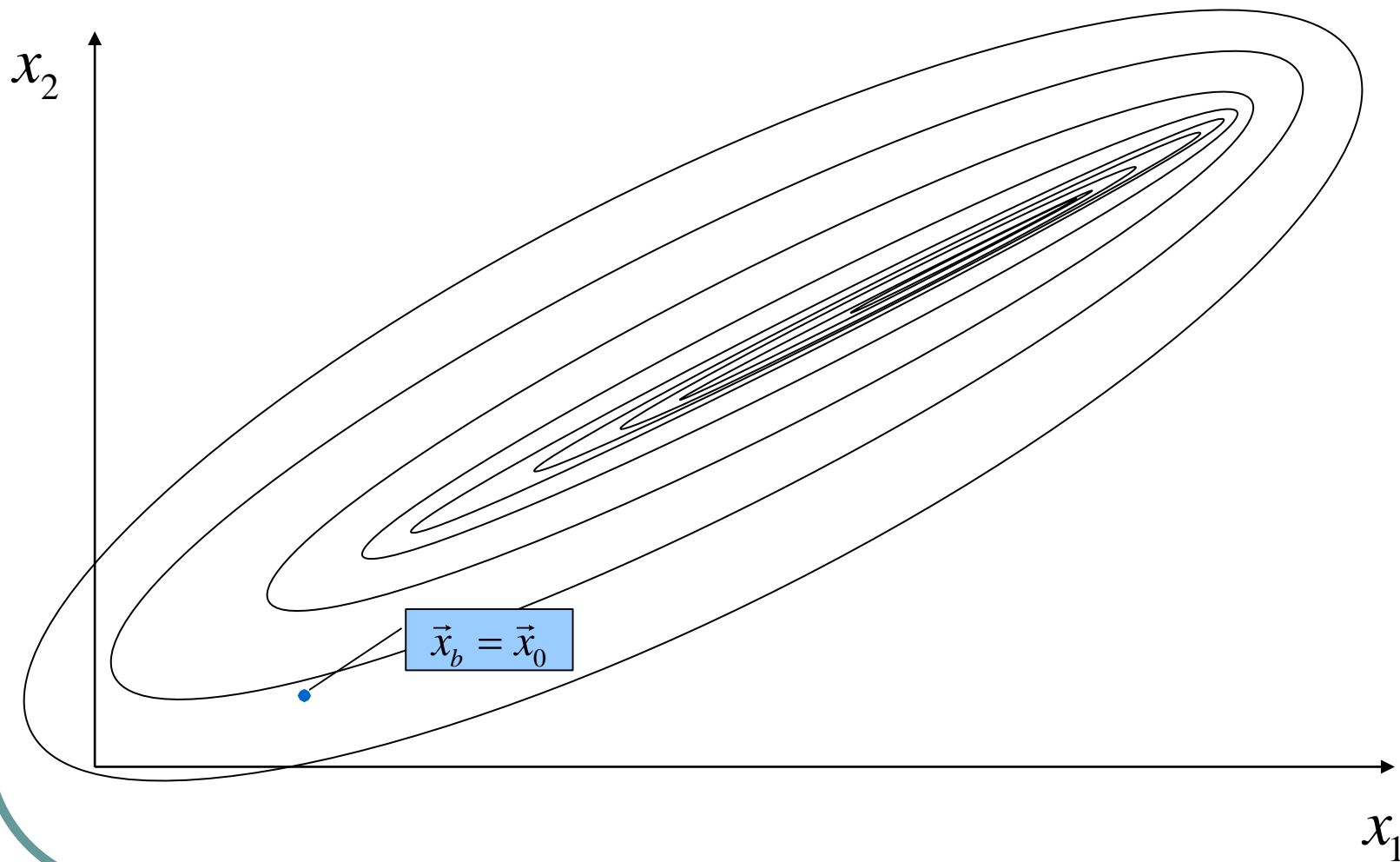
Метод локальных вариаций



Нелинейное программирование.

Безусловная минимизация функций n переменных.

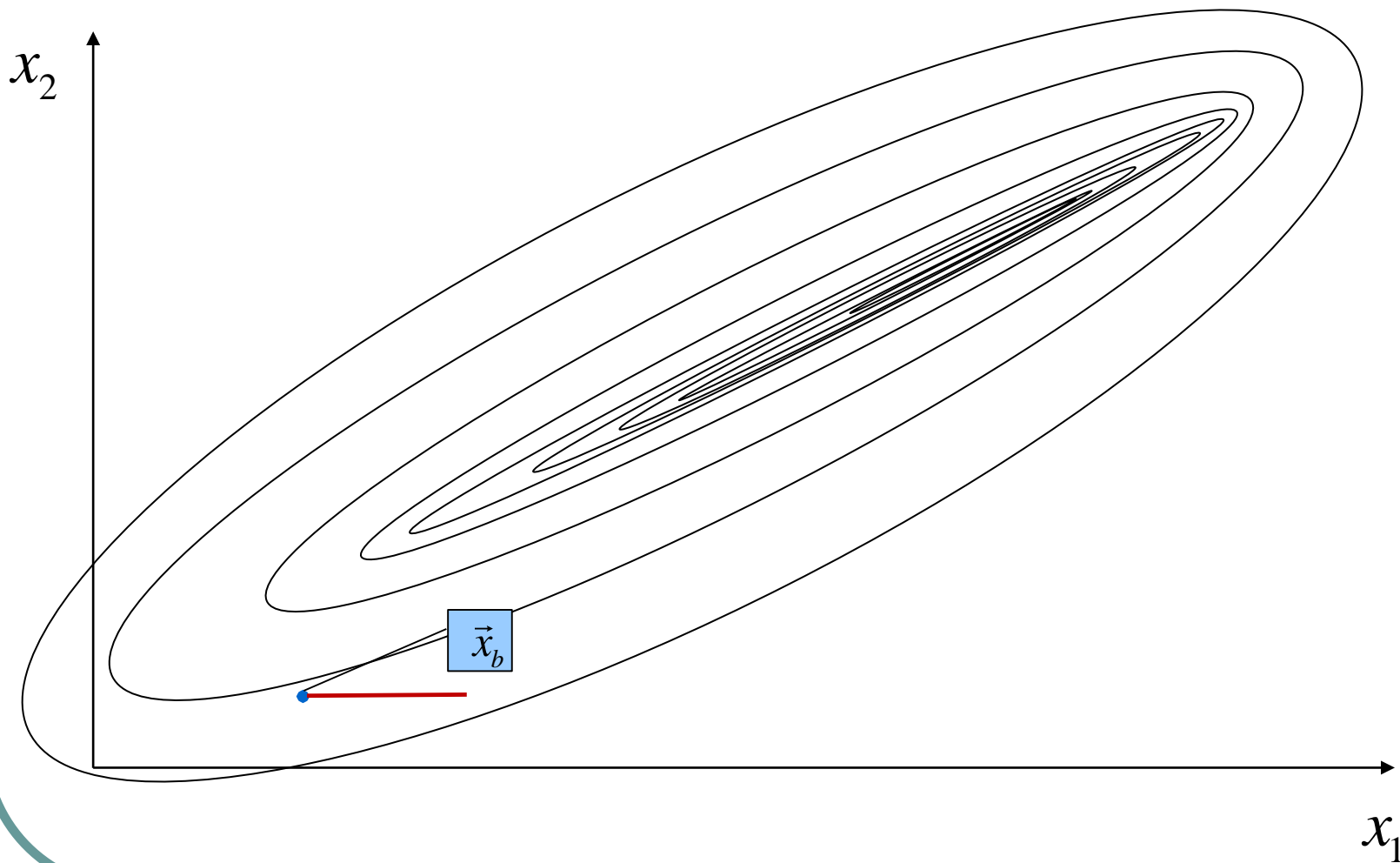
Метод Хука-Дживса



Нелинейное программирование.

Безусловная минимизация функций n переменных.

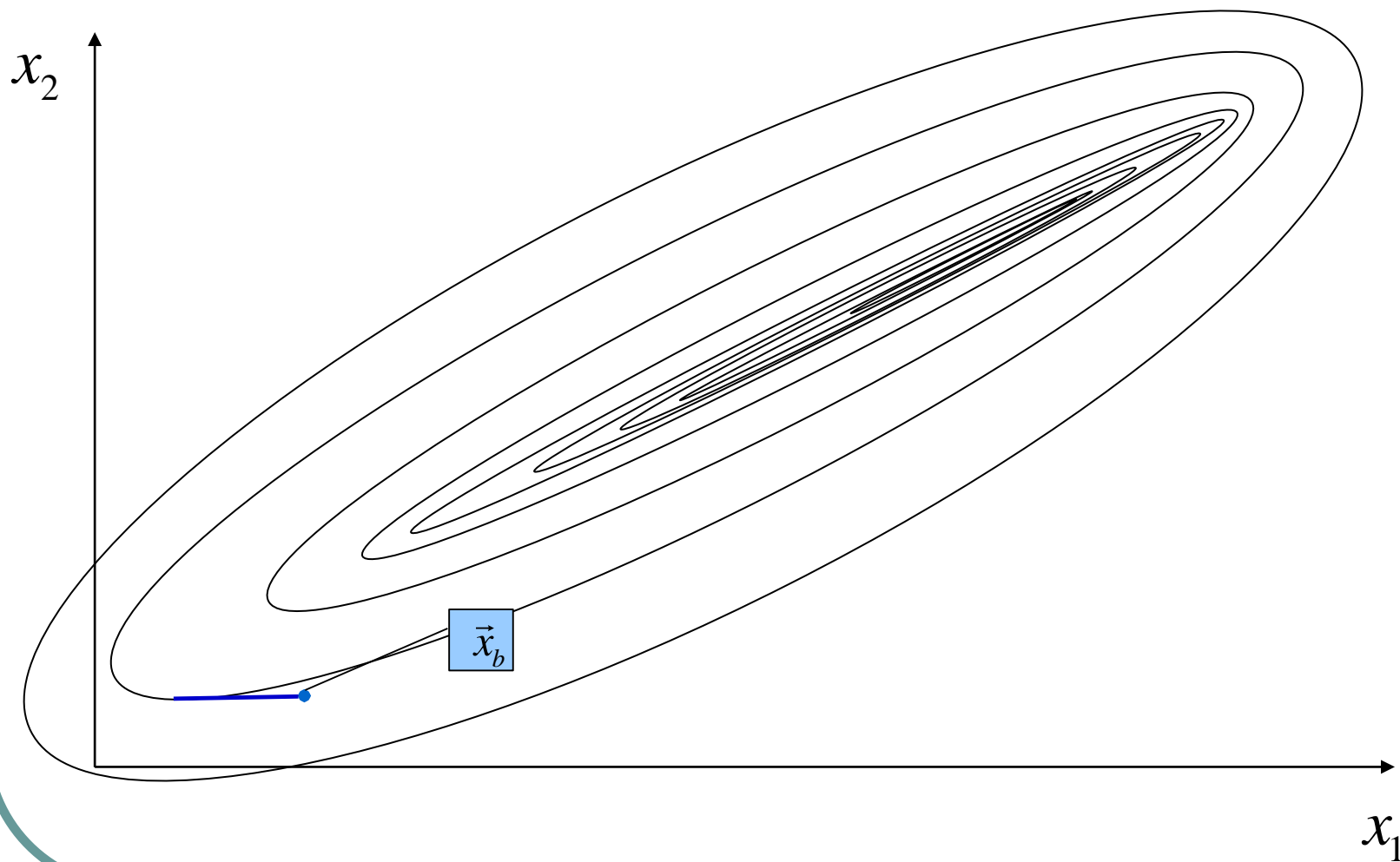
Метод Хука-Дживса



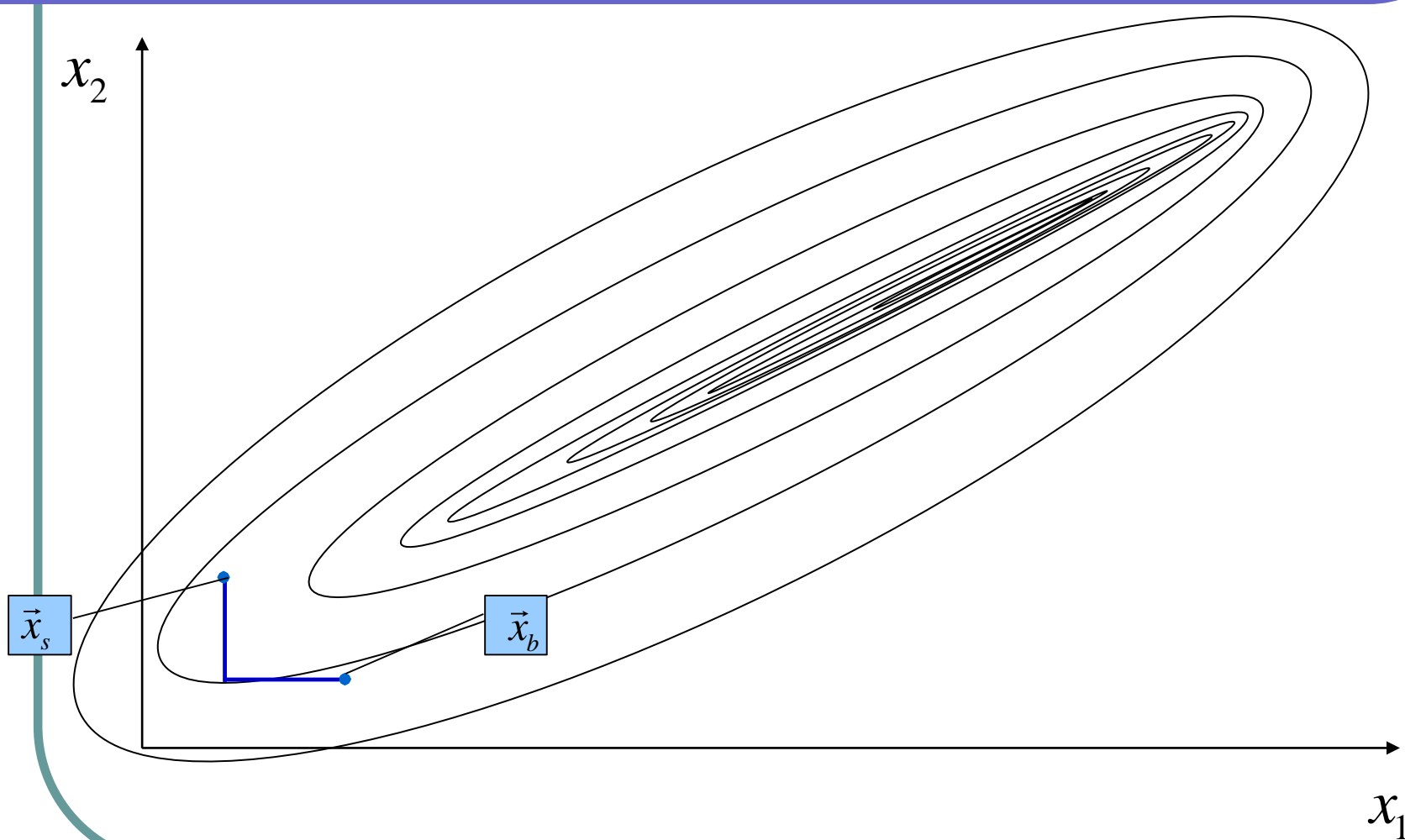
Нелинейное программирование.

Безусловная минимизация функций n переменных.

Метод Хука-Дживса



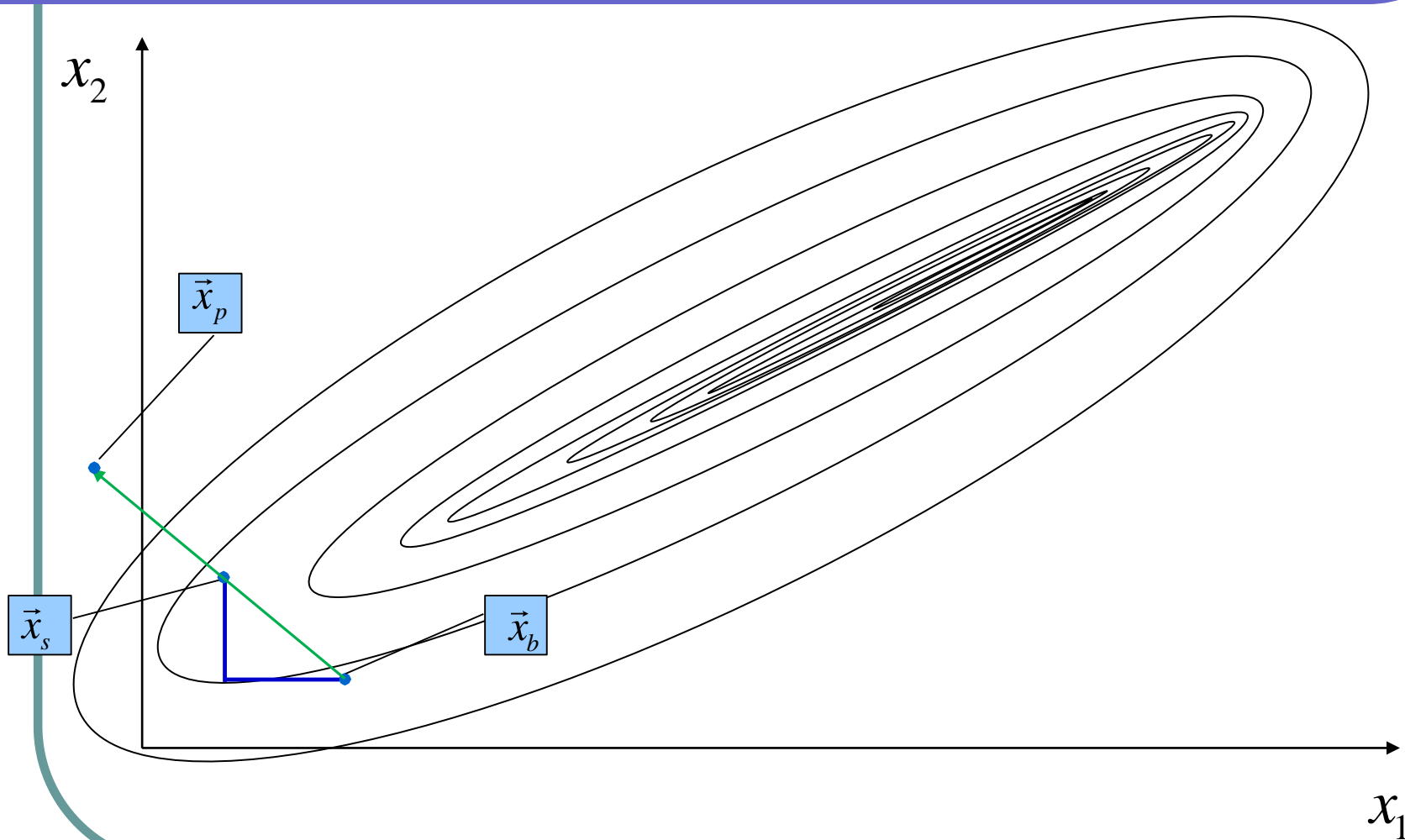
Нелинейное программирование. Безусловная минимизация функций n переменных. Метод Хука-Дживса



Нелинейное программирование.

Безусловная минимизация функций n переменных.

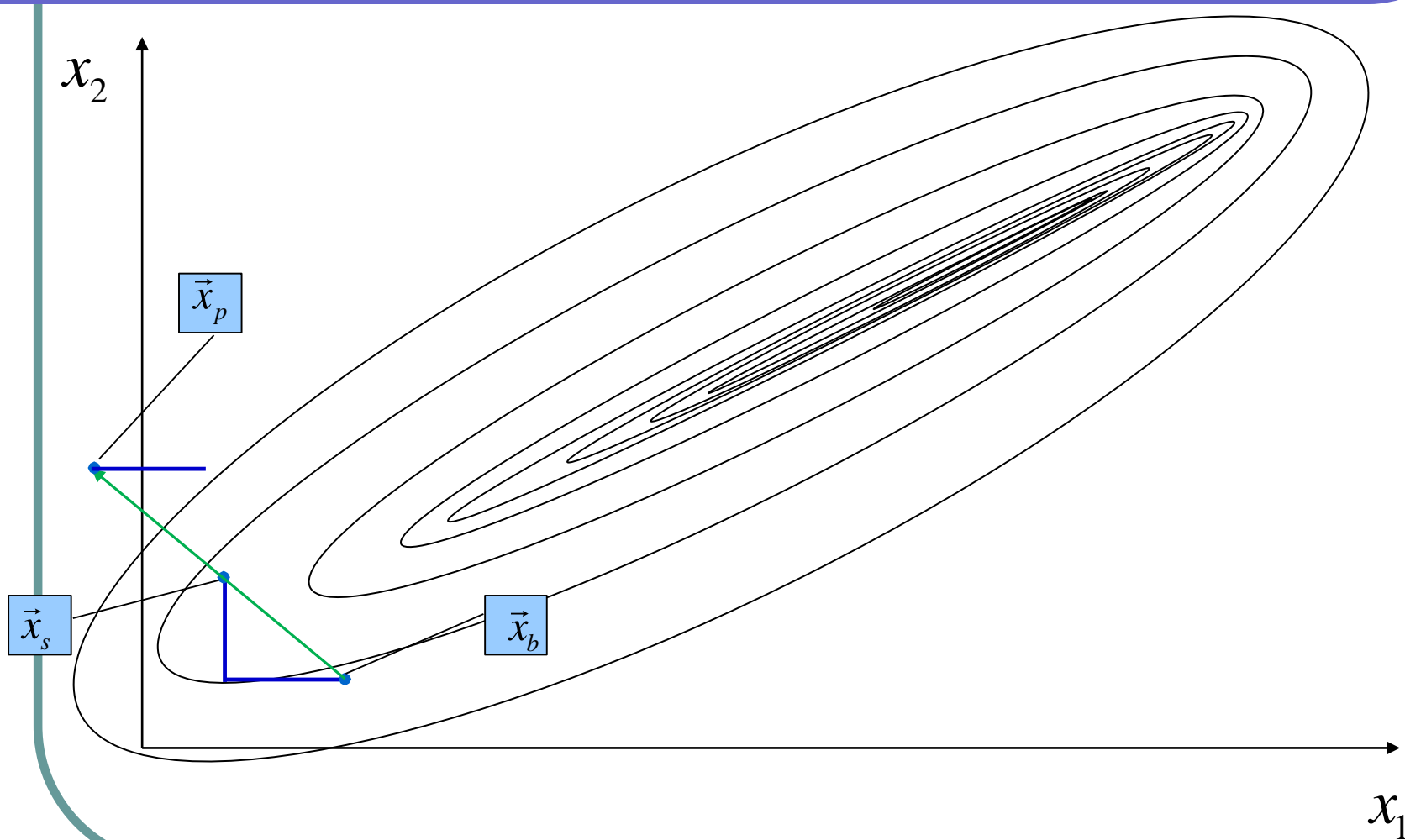
Метод Хука-Дживса



Нелинейное программирование.

Безусловная минимизация функций n переменных.

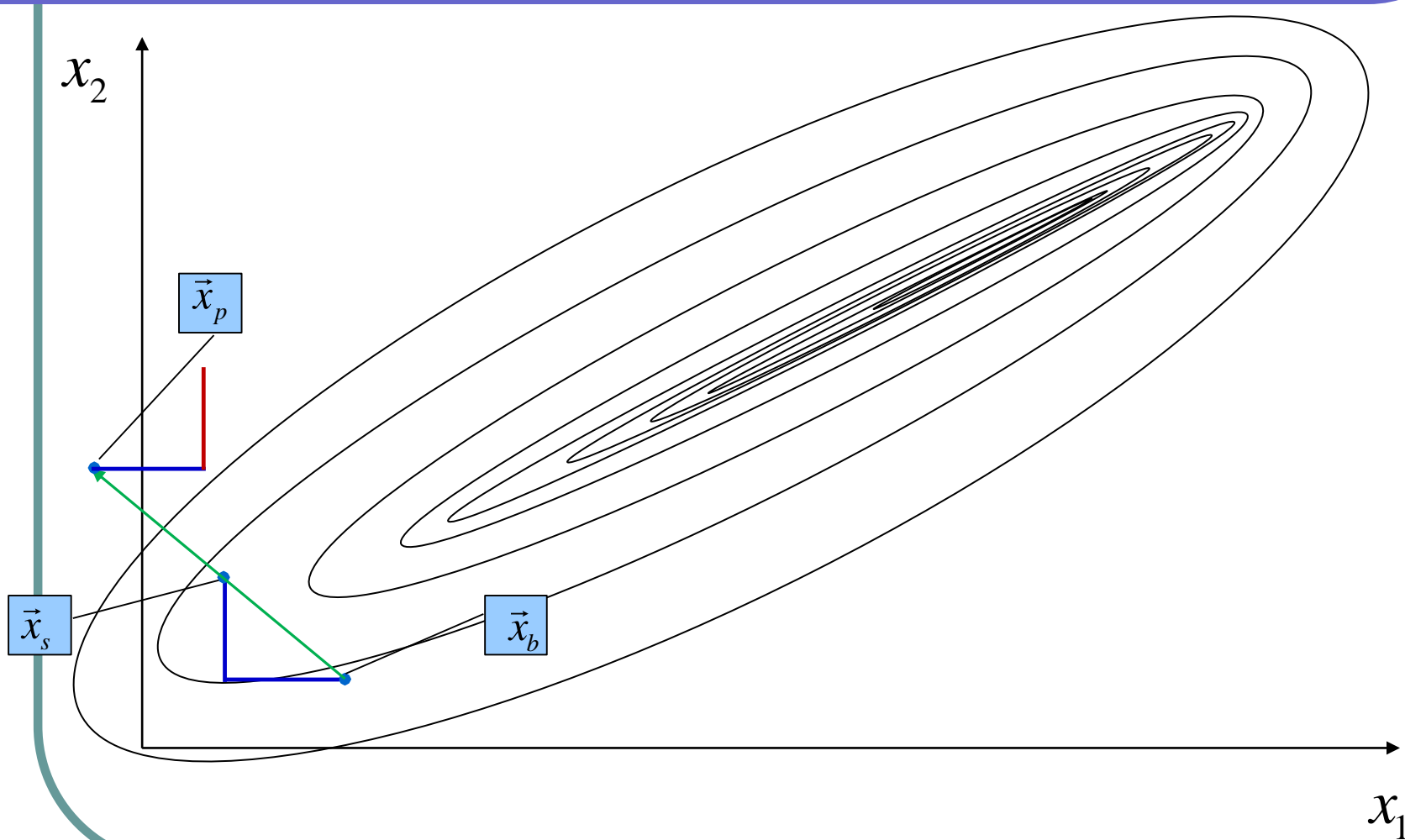
Метод Хука-Дживса



Нелинейное программирование.

Безусловная минимизация функций n переменных.

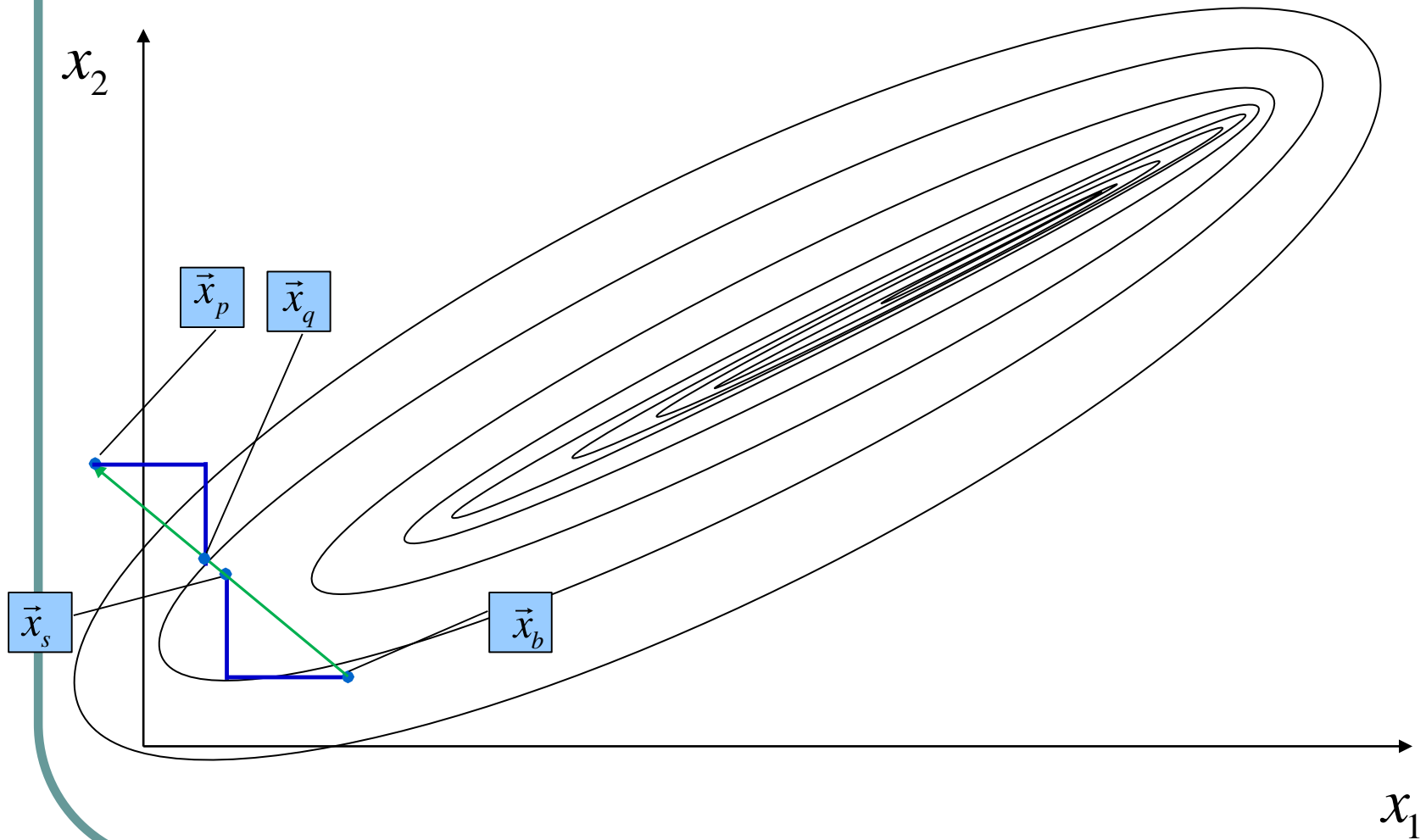
Метод Хука-Дживса



Нелинейное программирование.

Безусловная минимизация функций n переменных.

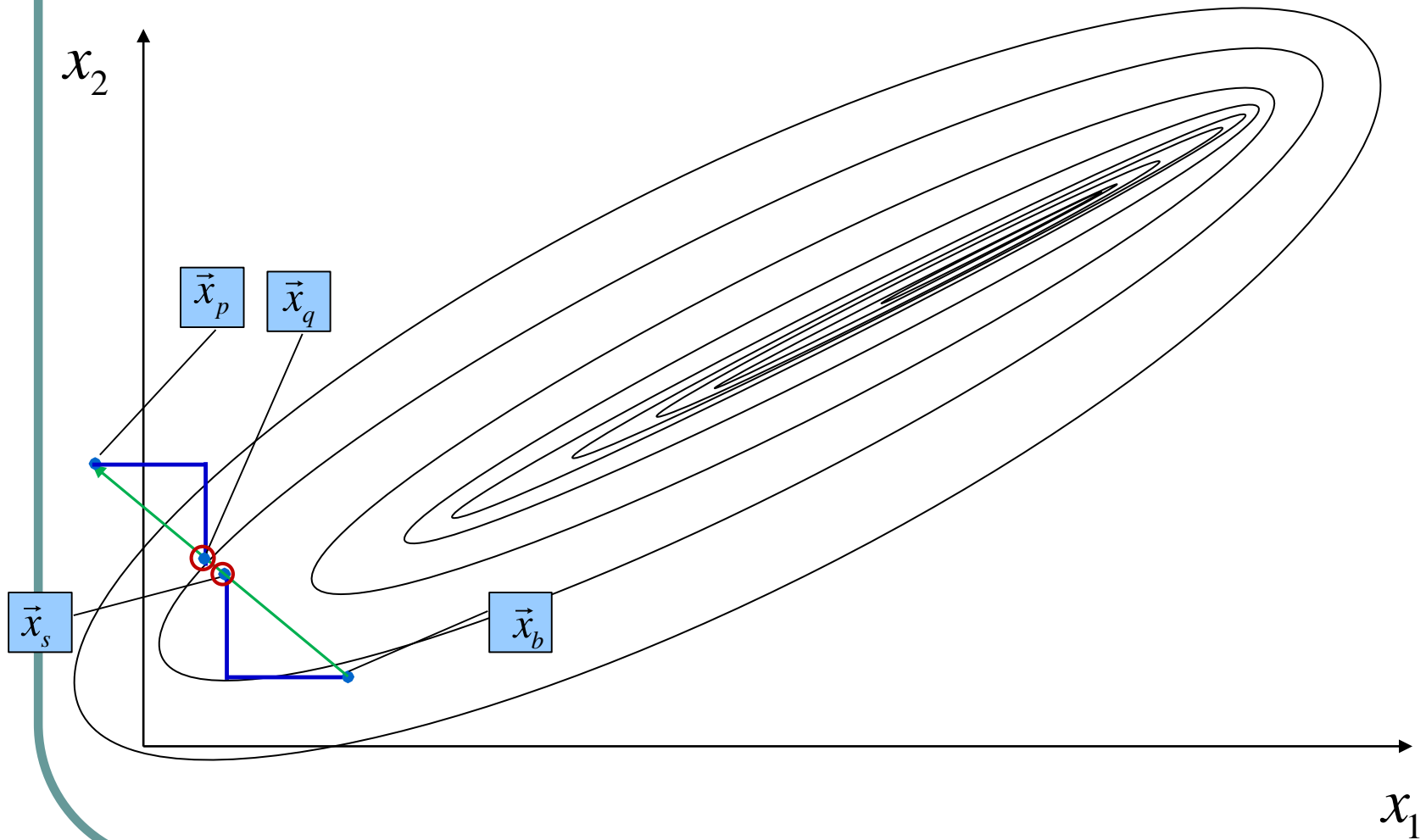
Метод Хука-Дживса



Нелинейное программирование.

Безусловная минимизация функций n переменных.

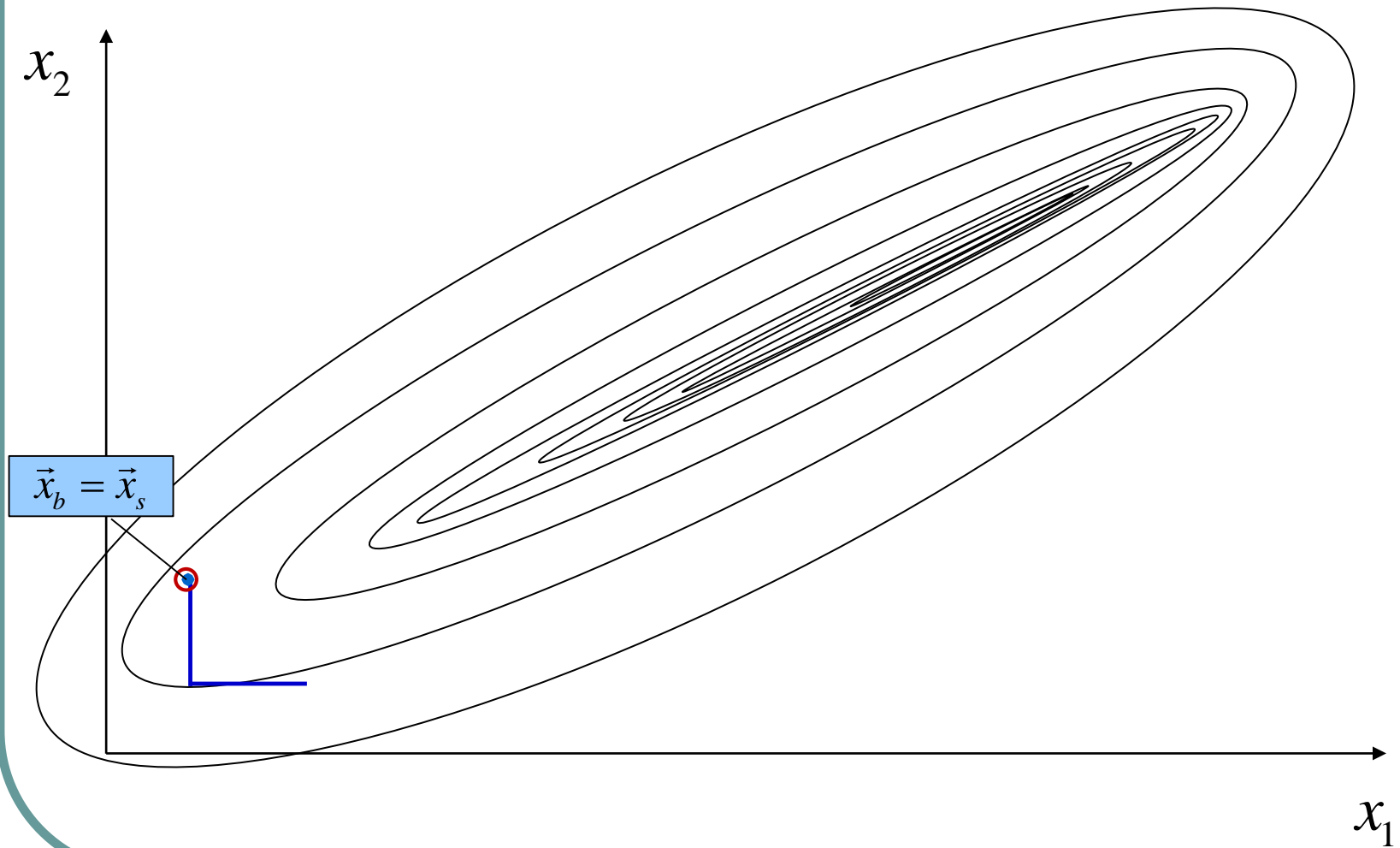
Метод Хука-Дживса



Нелинейное программирование.

Безусловная минимизация функций n переменных.

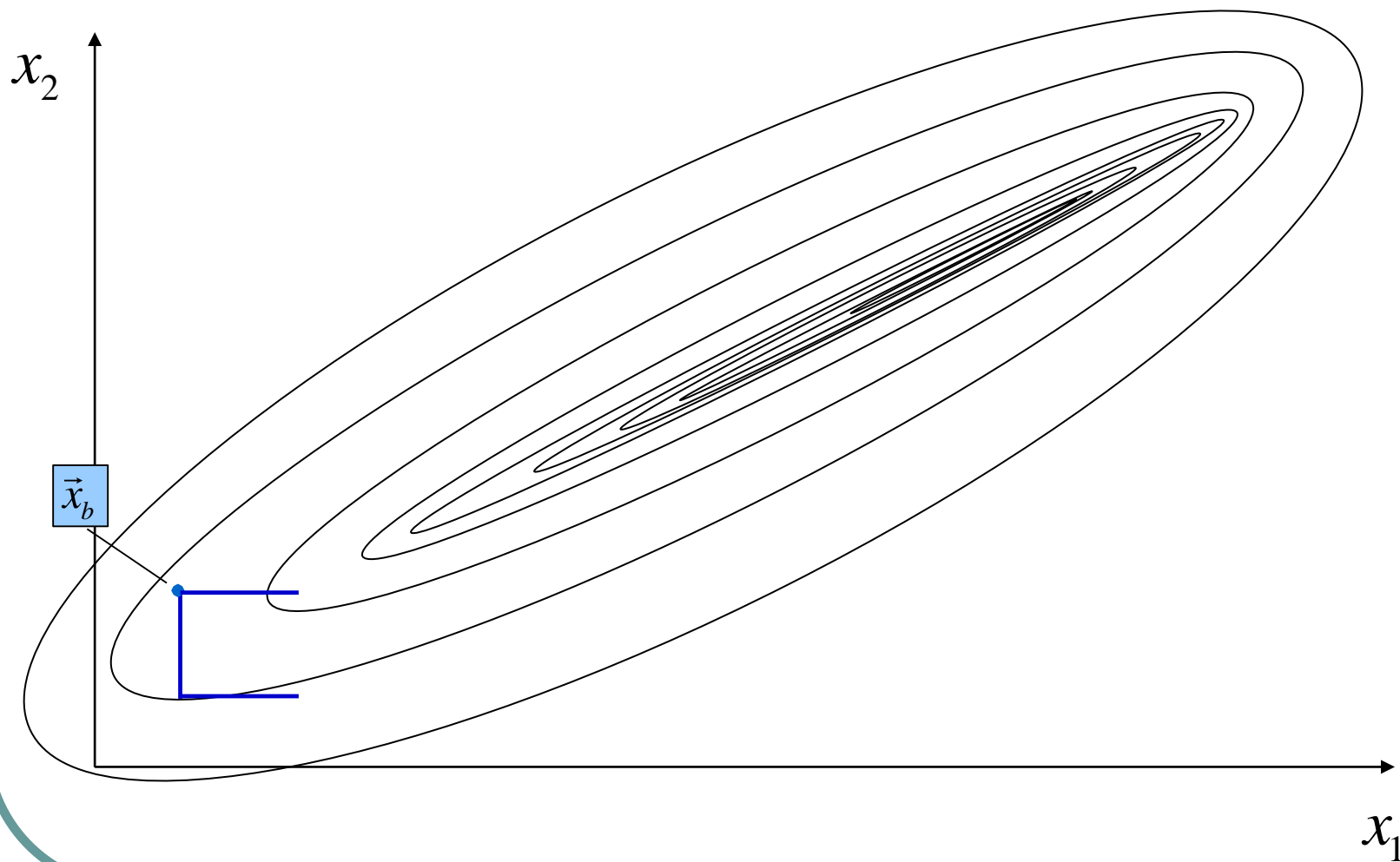
Метод Хука-Дживса



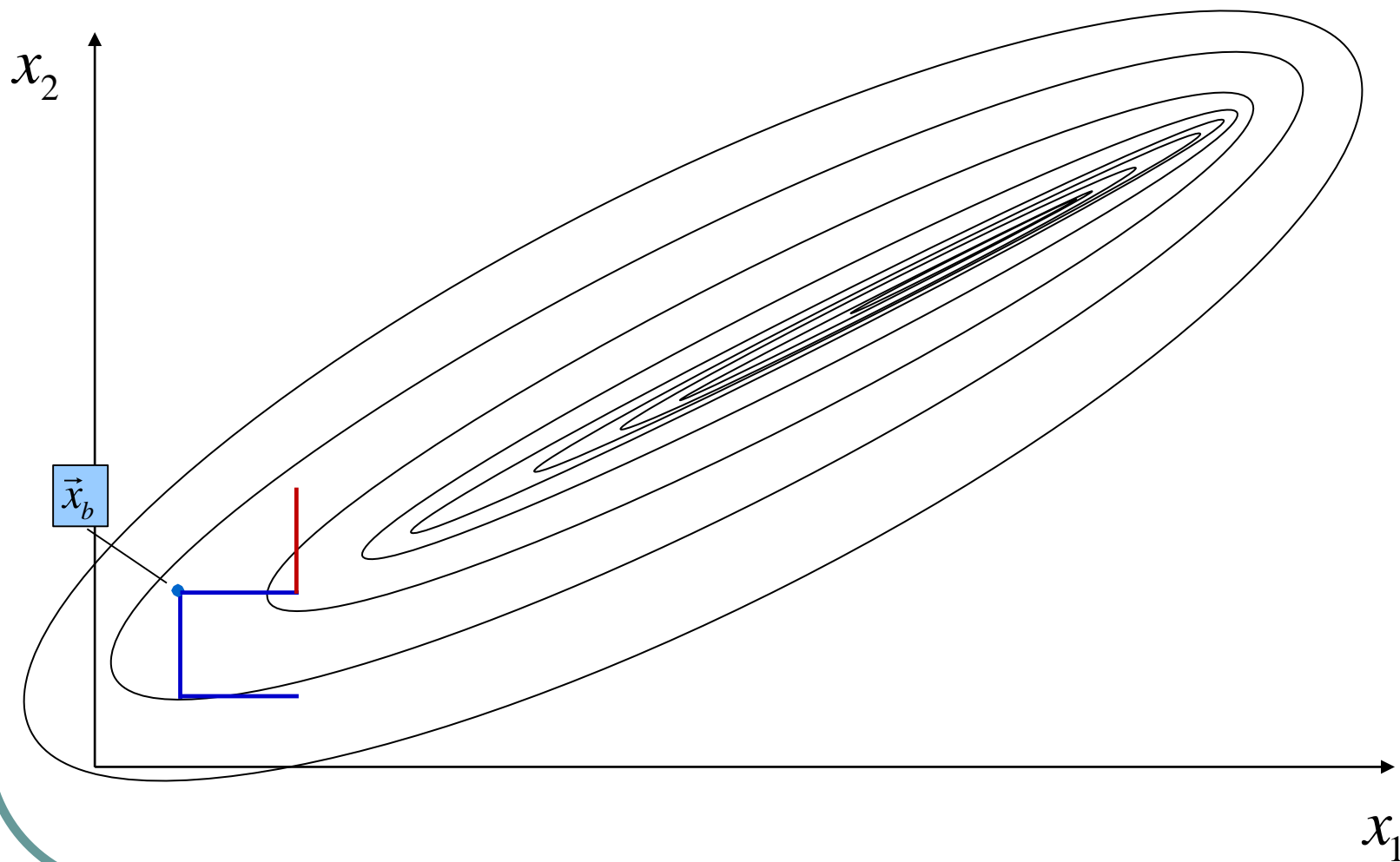
Нелинейное программирование.

Безусловная минимизация функций n переменных.

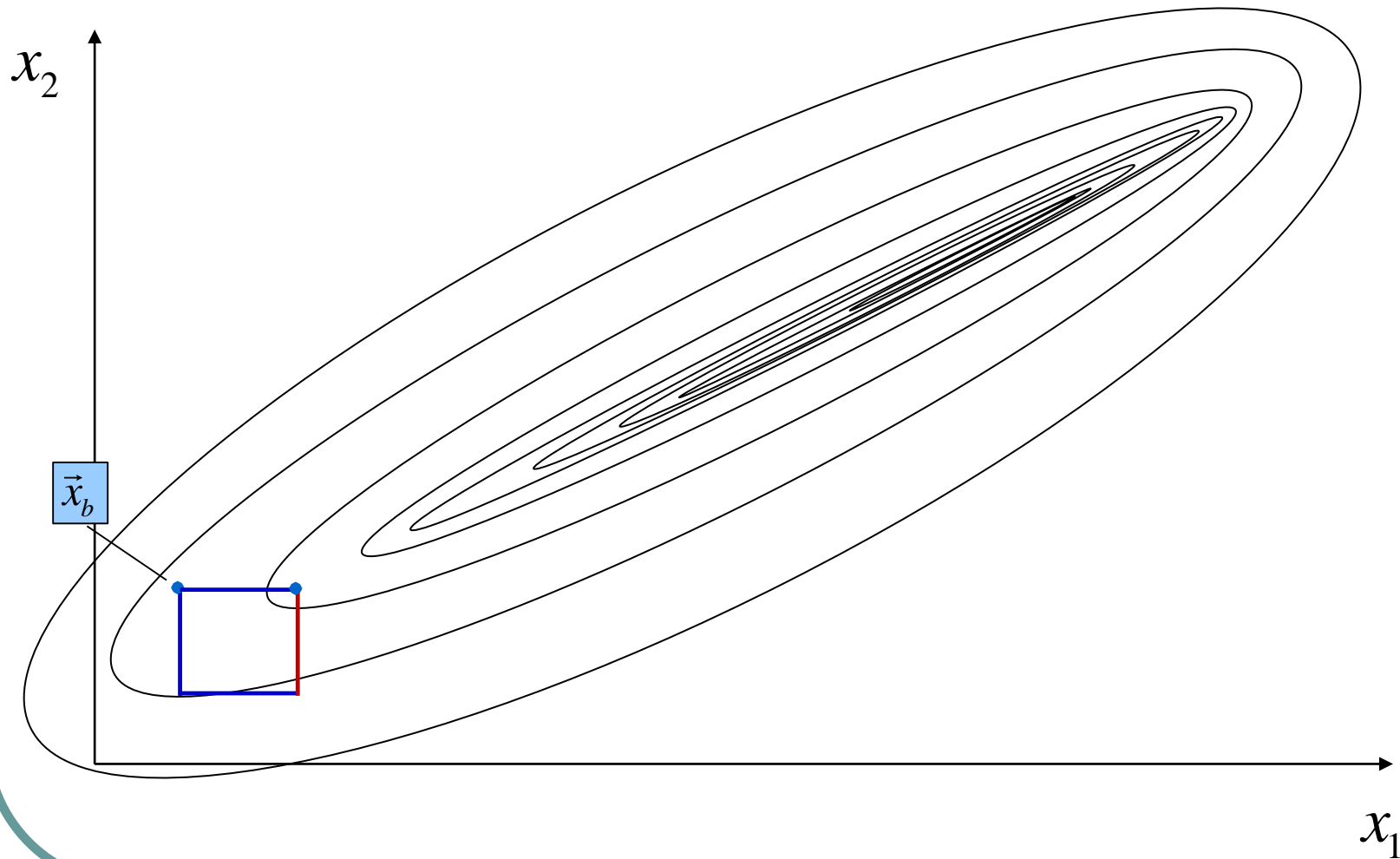
Метод Хука-Дживса



Нелинейное программирование. Безусловная минимизация функций n переменных. Метод Хука-Дживса



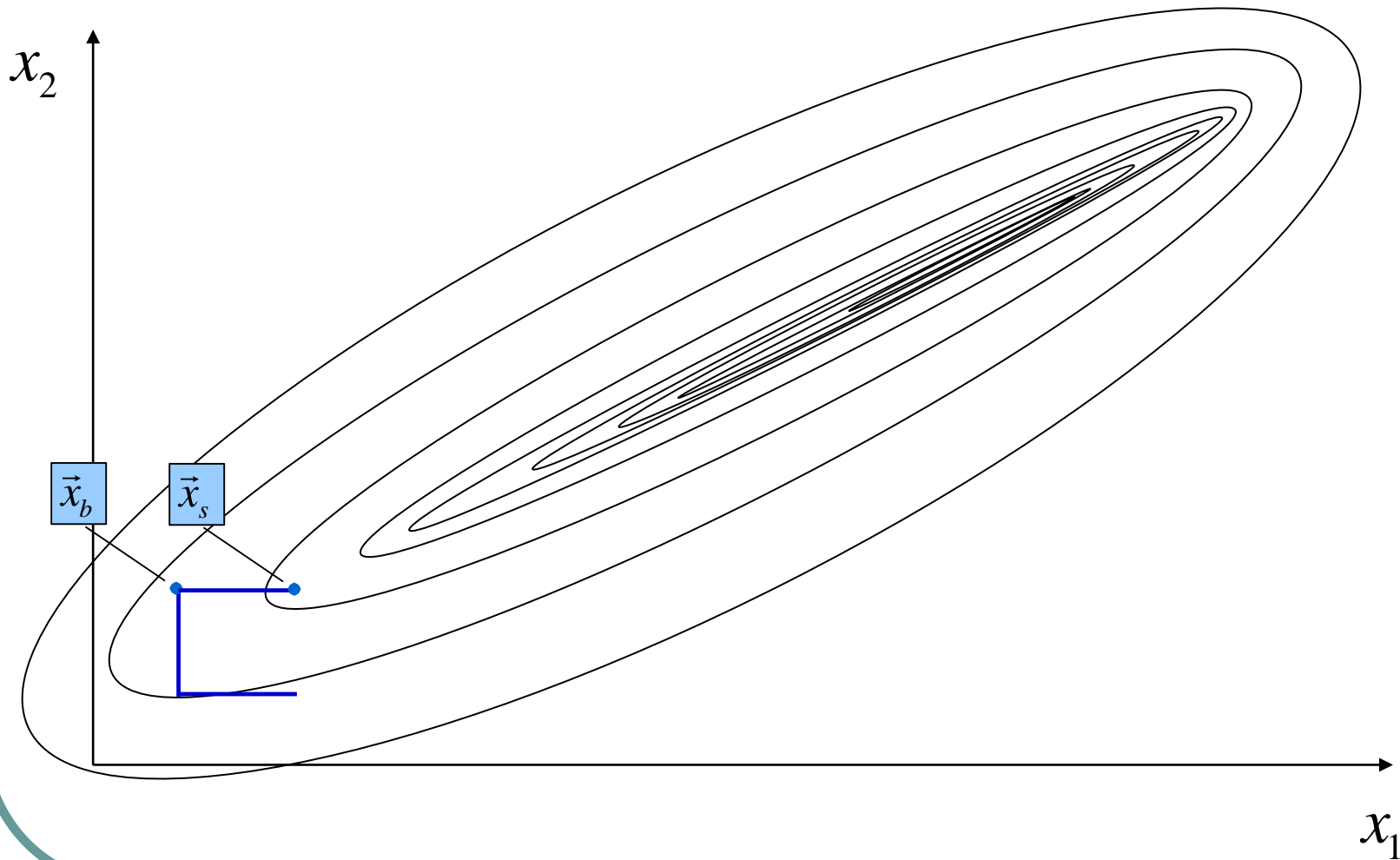
Нелинейное программирование. Безусловная минимизация функций n переменных. Метод Хука-Дживса



Нелинейное программирование.

Безусловная минимизация функций n переменных.

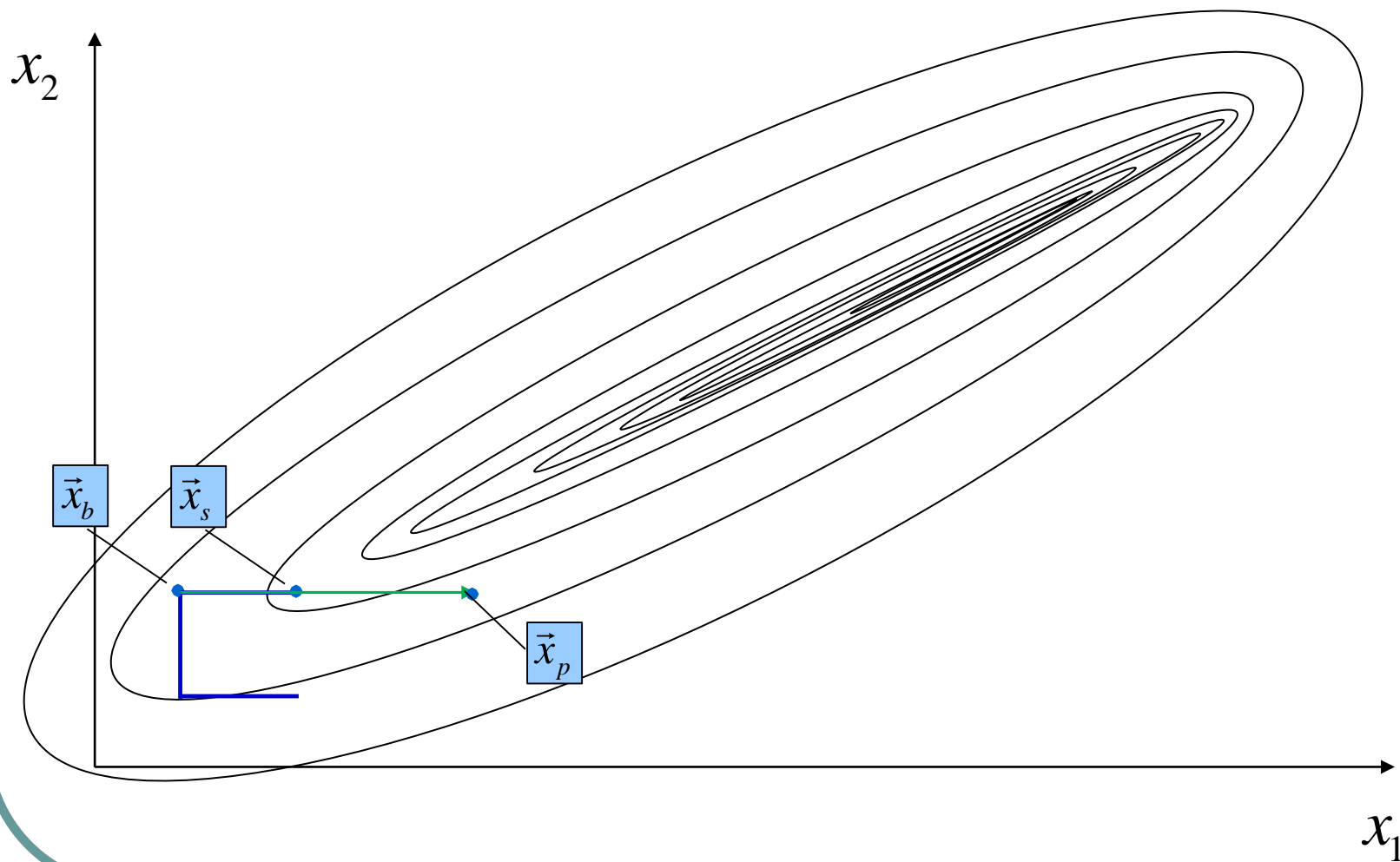
Метод Хука-Дживса



Нелинейное программирование.

Безусловная минимизация функций n переменных.

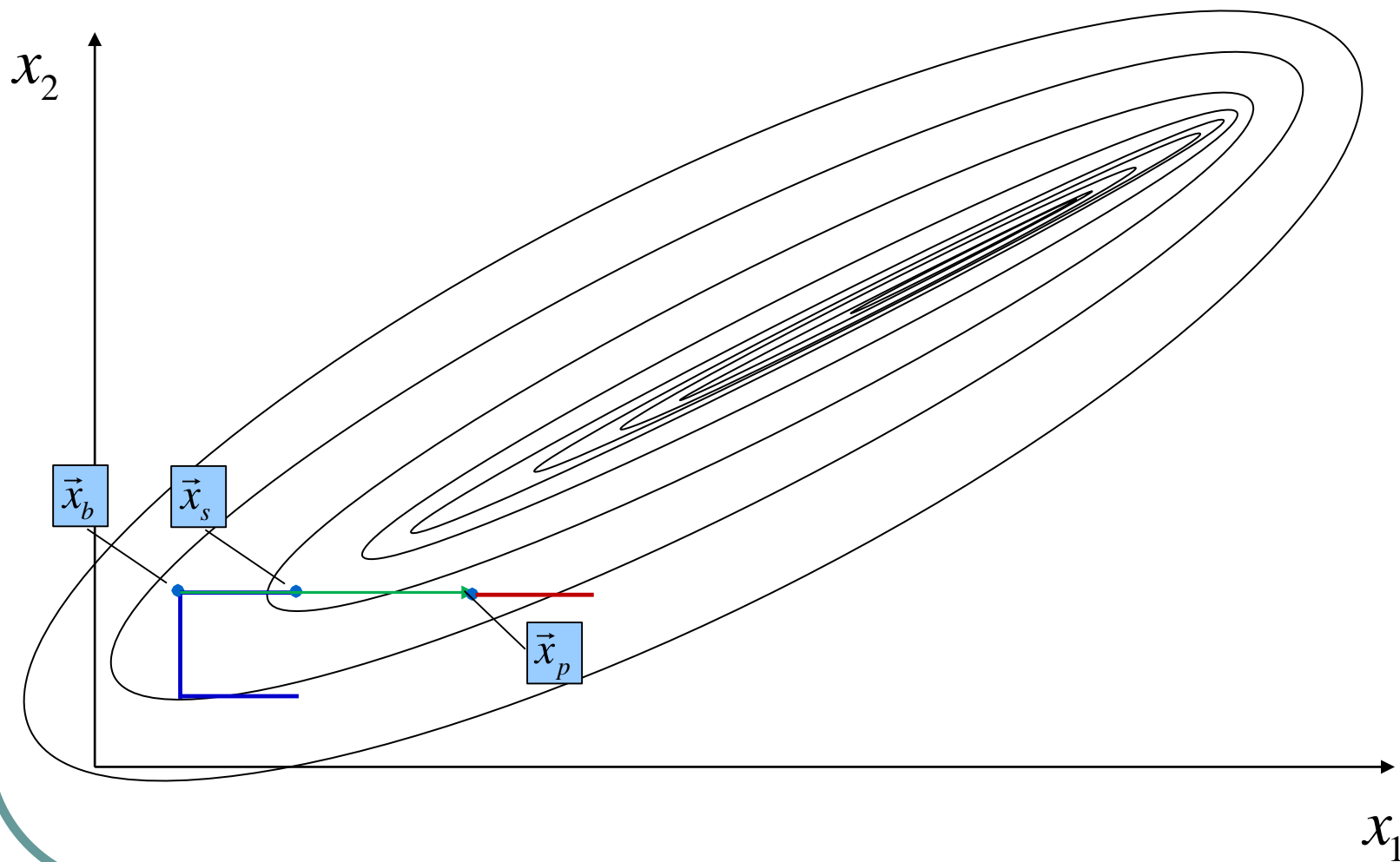
Метод Хука-Дживса



Нелинейное программирование.

Безусловная минимизация функций n переменных.

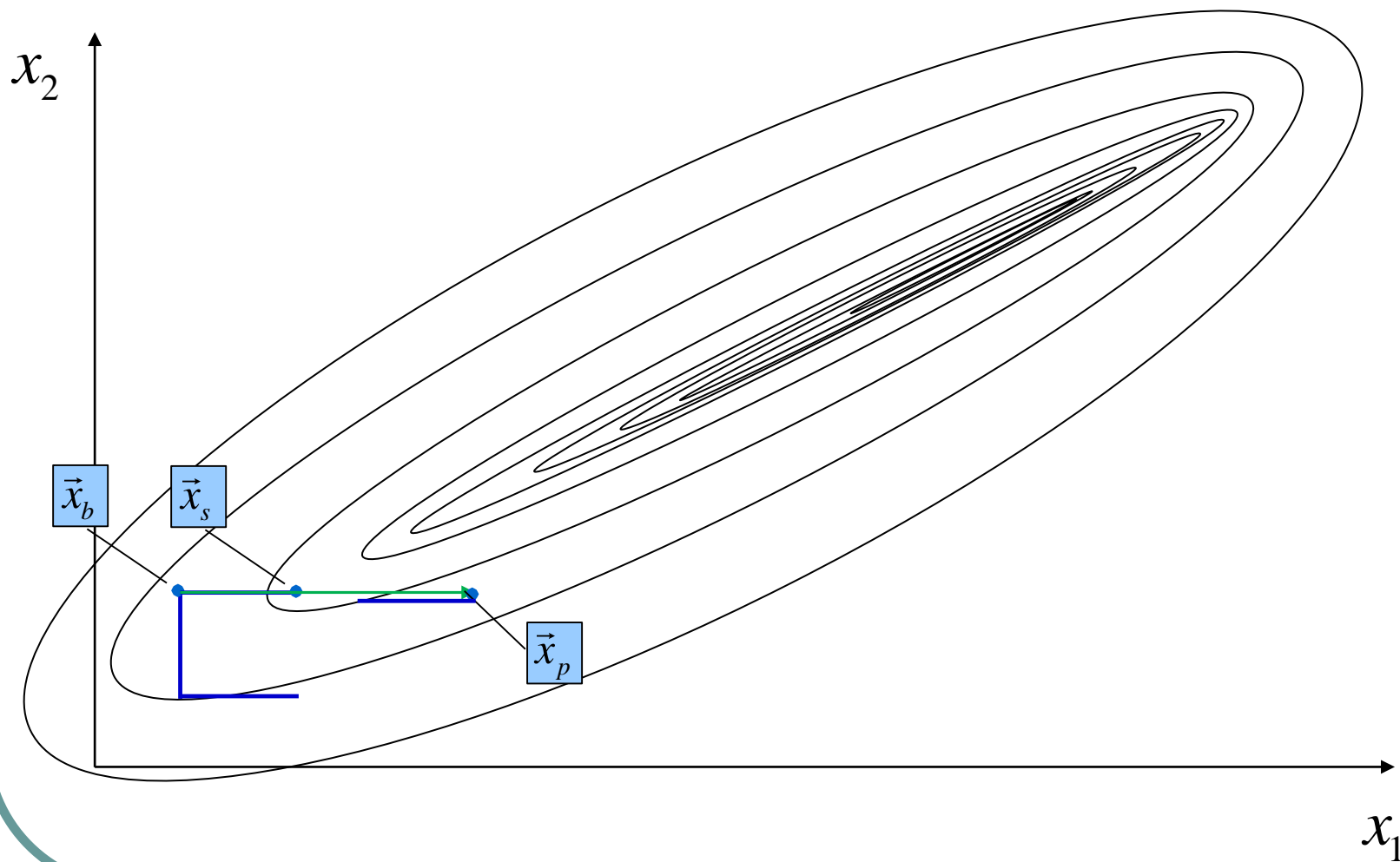
Метод Хука-Дживса



Нелинейное программирование.

Безусловная минимизация функций n переменных.

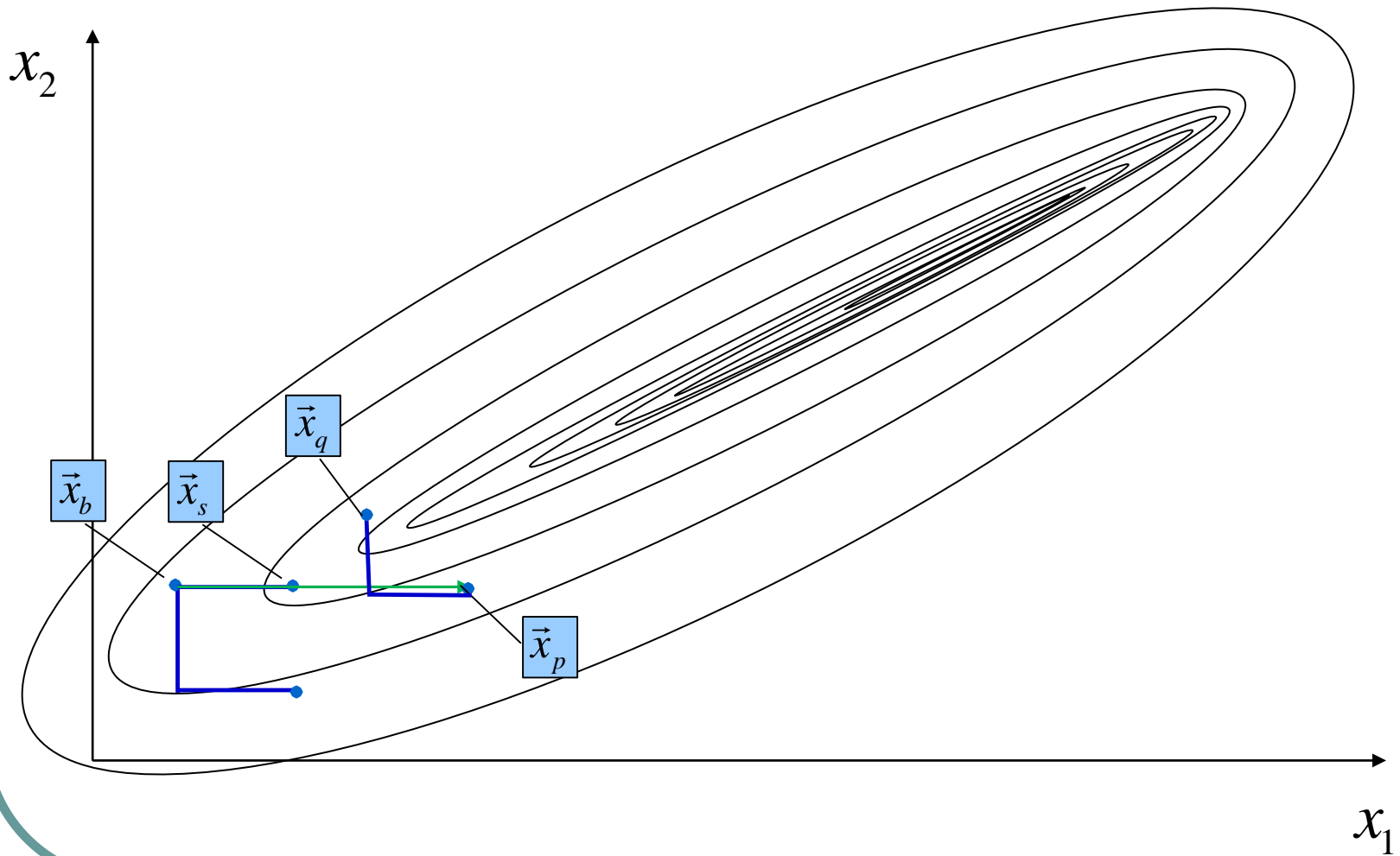
Метод Хука-Дживса



Нелинейное программирование.

Безусловная минимизация функций n переменных.

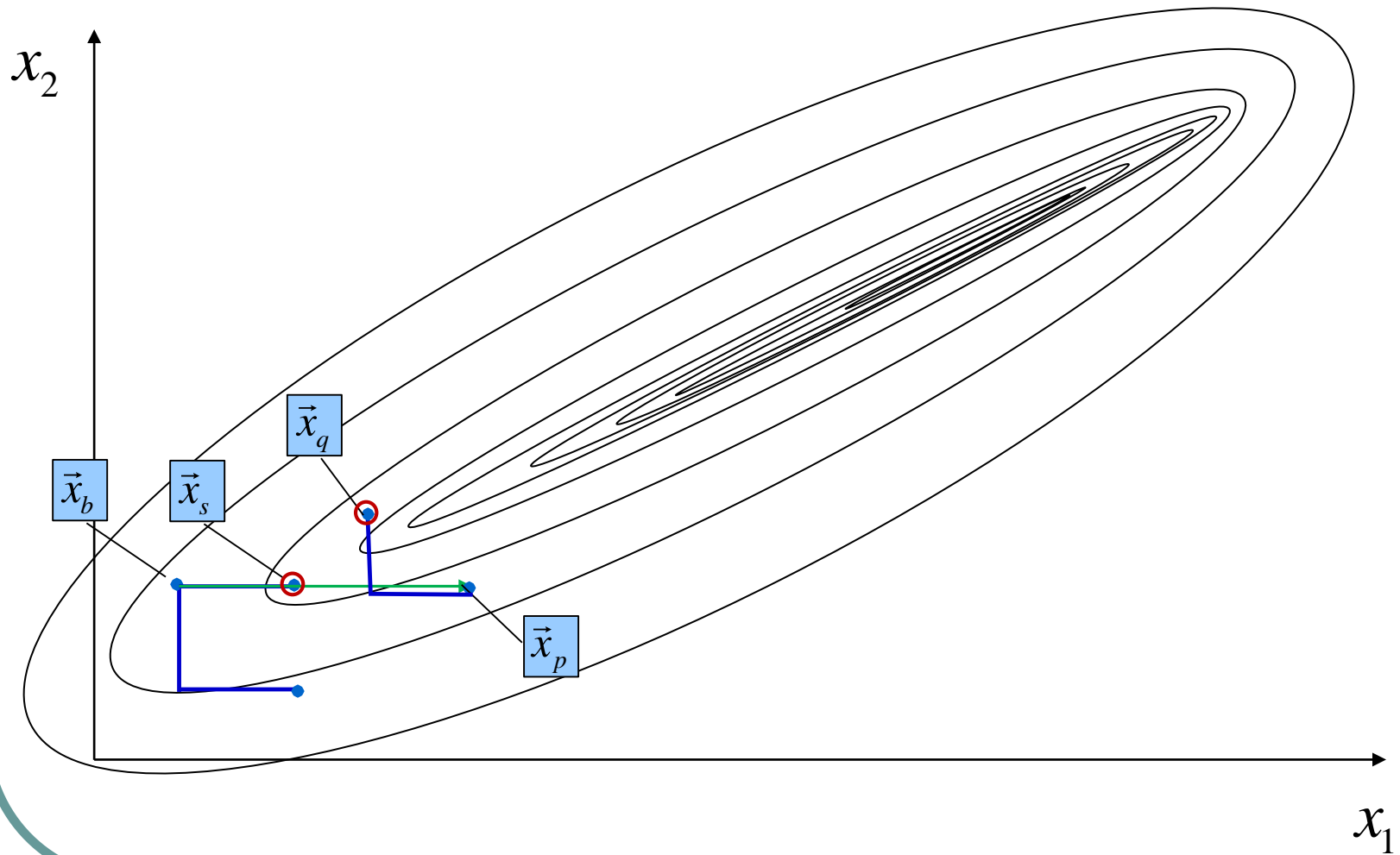
Метод Хука-Дживса



Нелинейное программирование.

Безусловная минимизация функций n переменных.

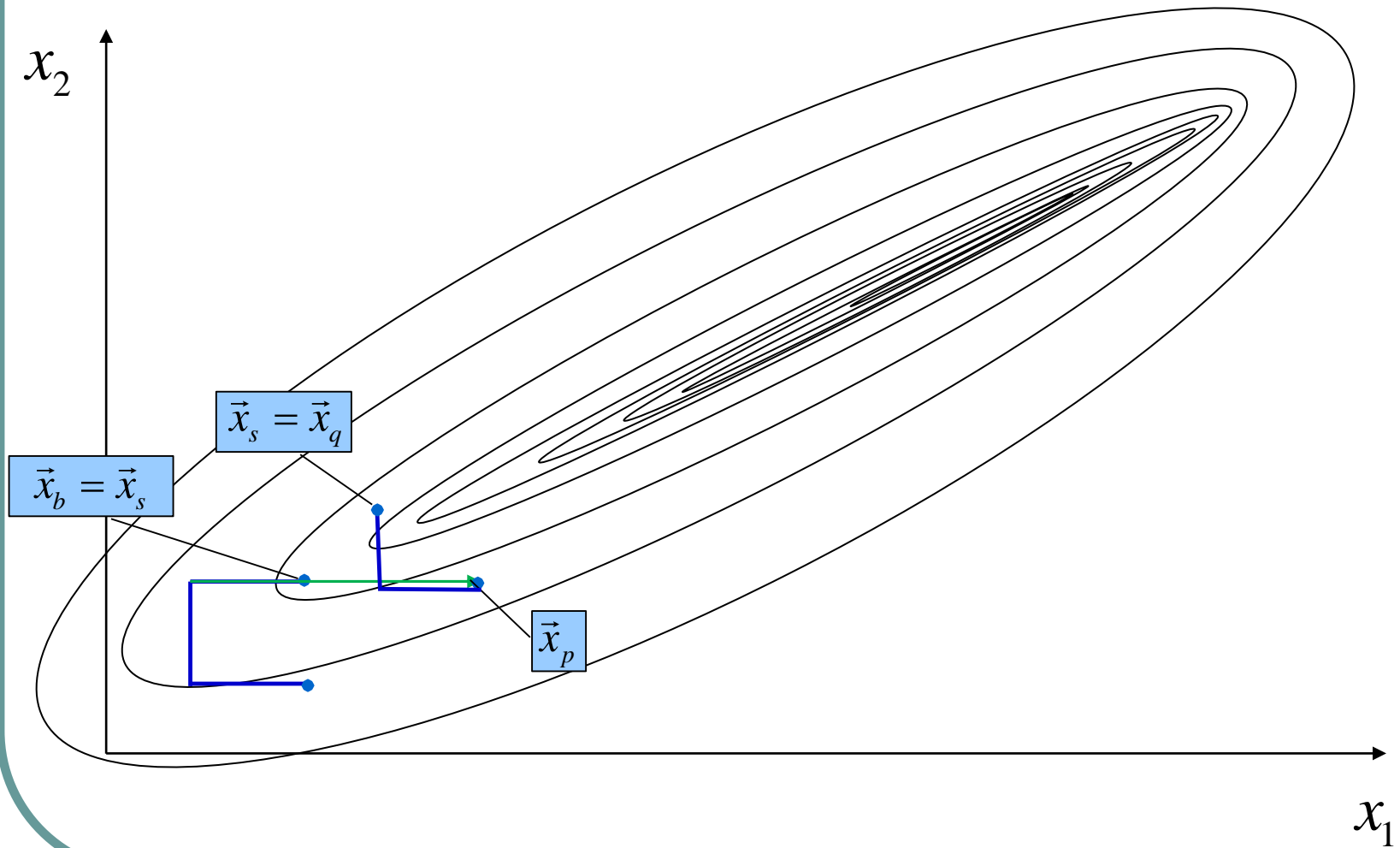
Метод Хука-Дживса



Нелинейное программирование.

Безусловная минимизация функций n переменных.

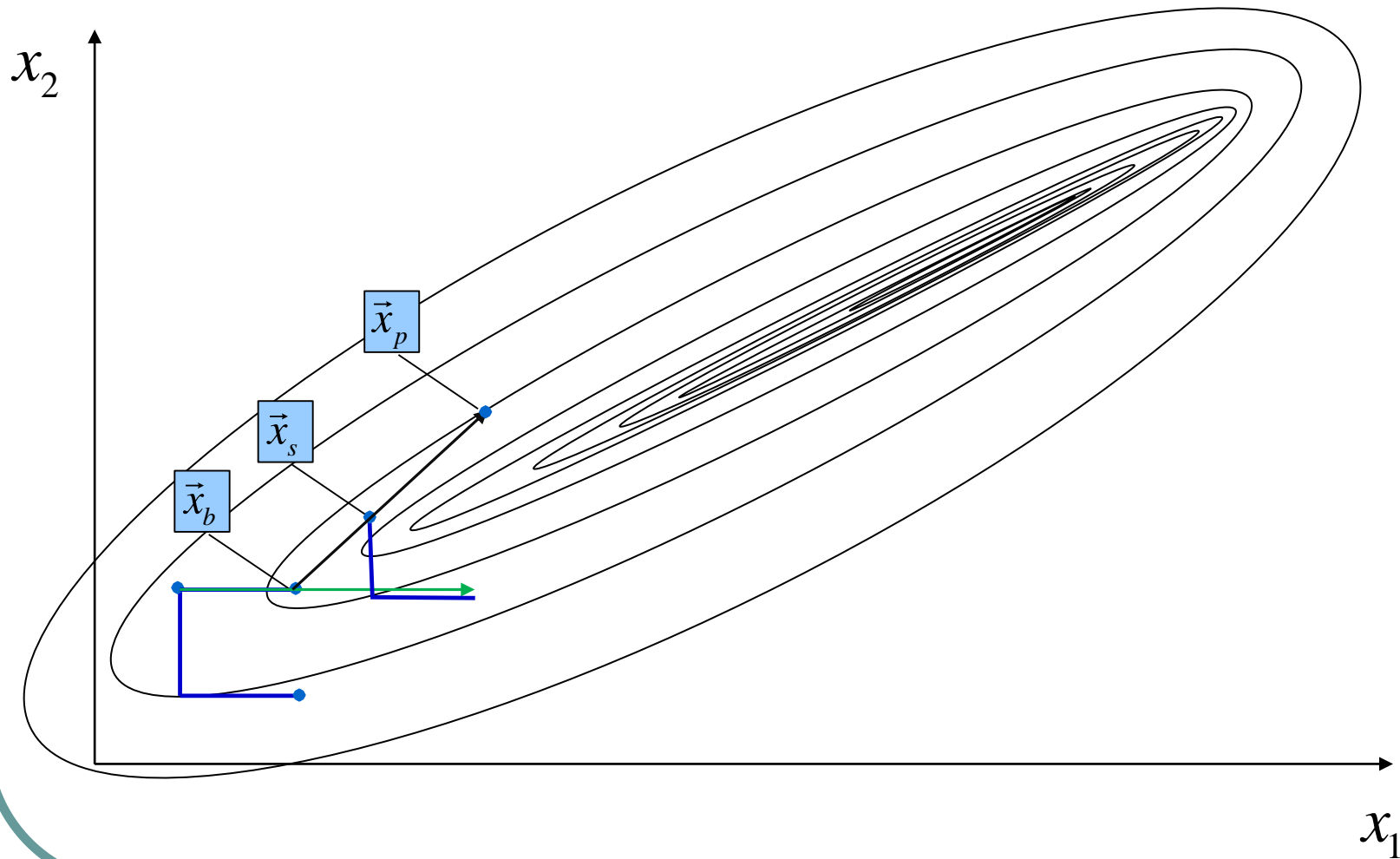
Метод Хука-Дживса



Нелинейное программирование.

Безусловная минимизация функций n переменных.

Метод Хука-Дживса

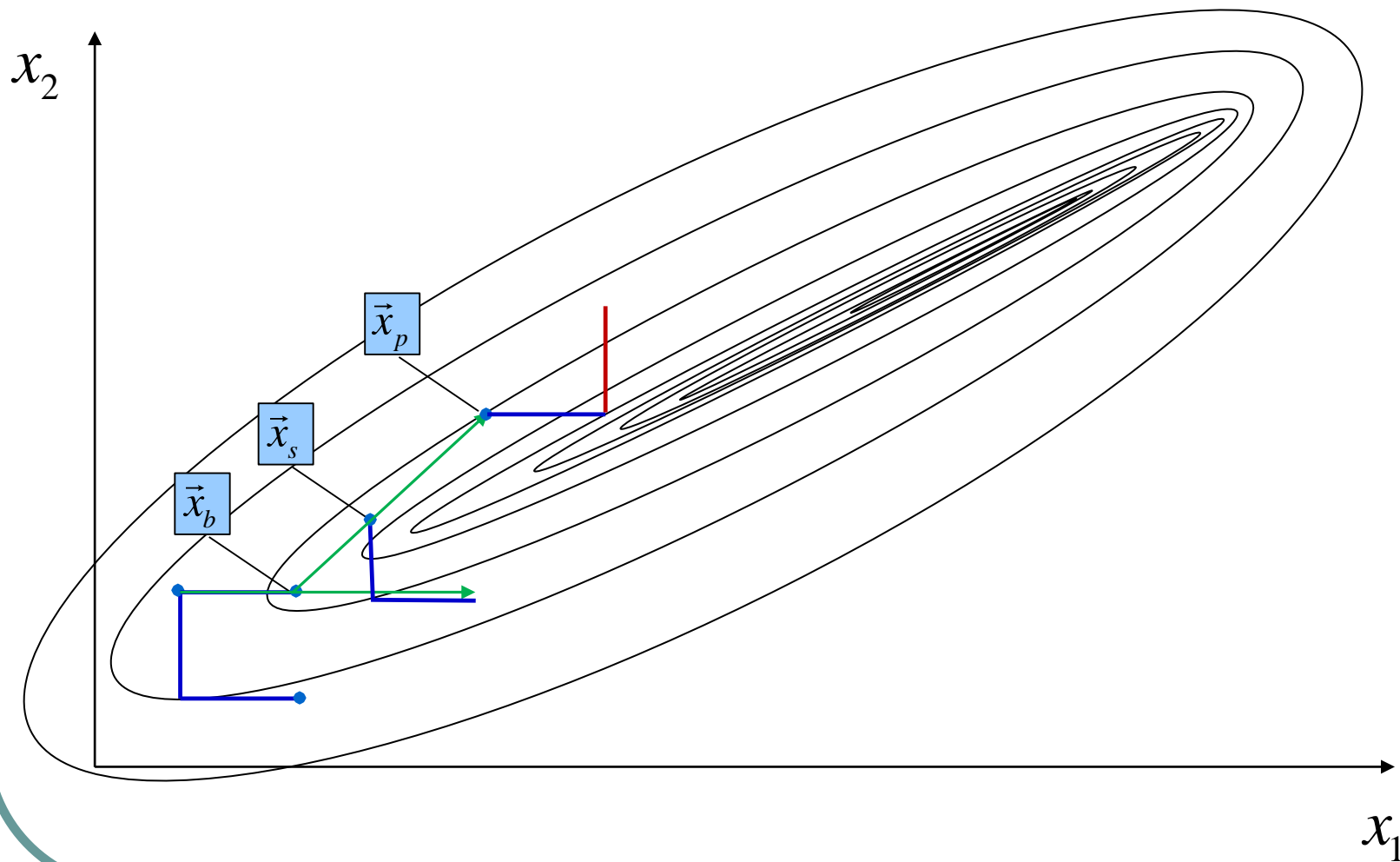


Нелинейное программирование.
Безусловная минимизация функций n переменных.
Метод Хука-Дживса

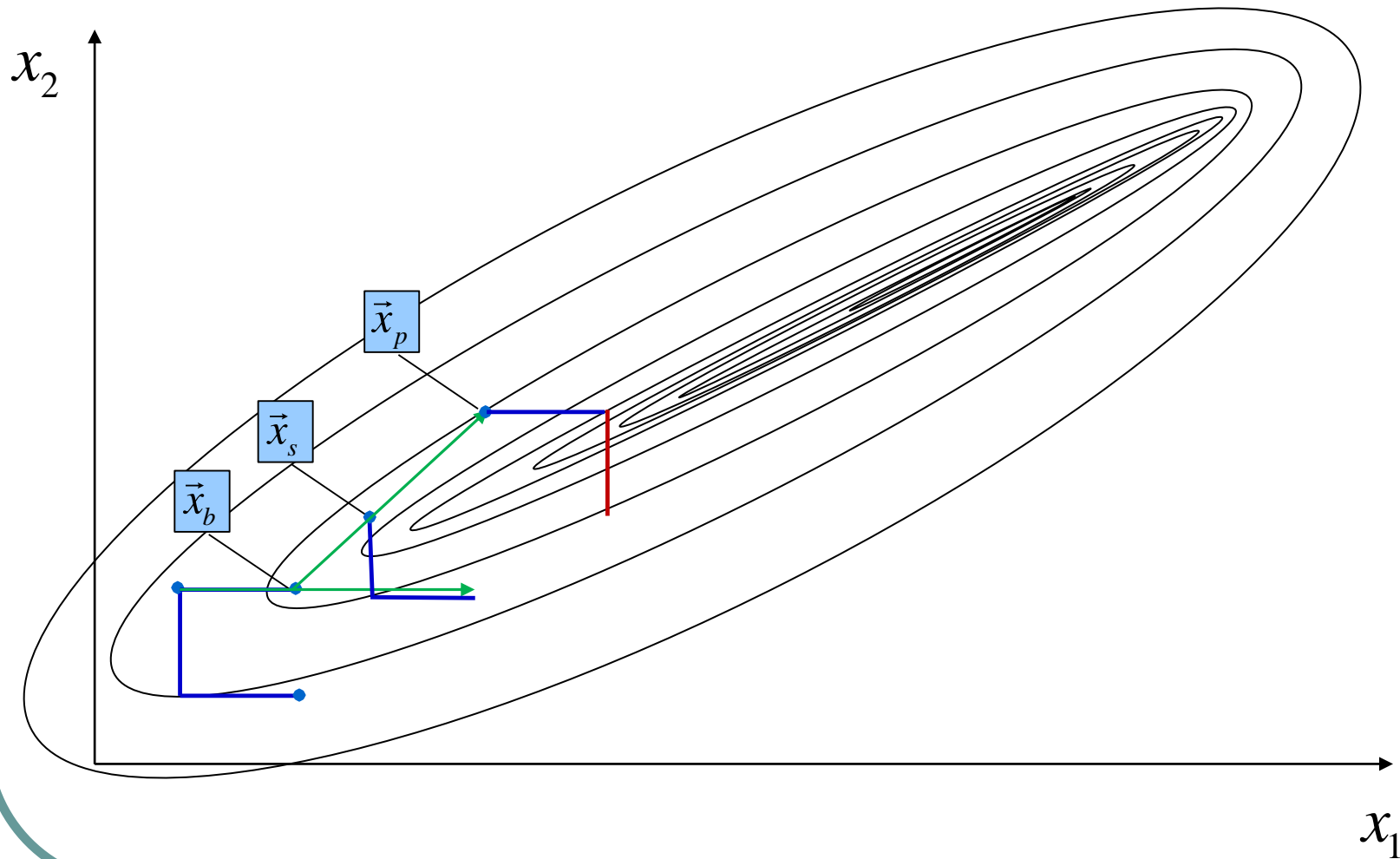
Нелинейное программирование.

Безусловная минимизация функций n переменных.

Метод Хука-Дживса



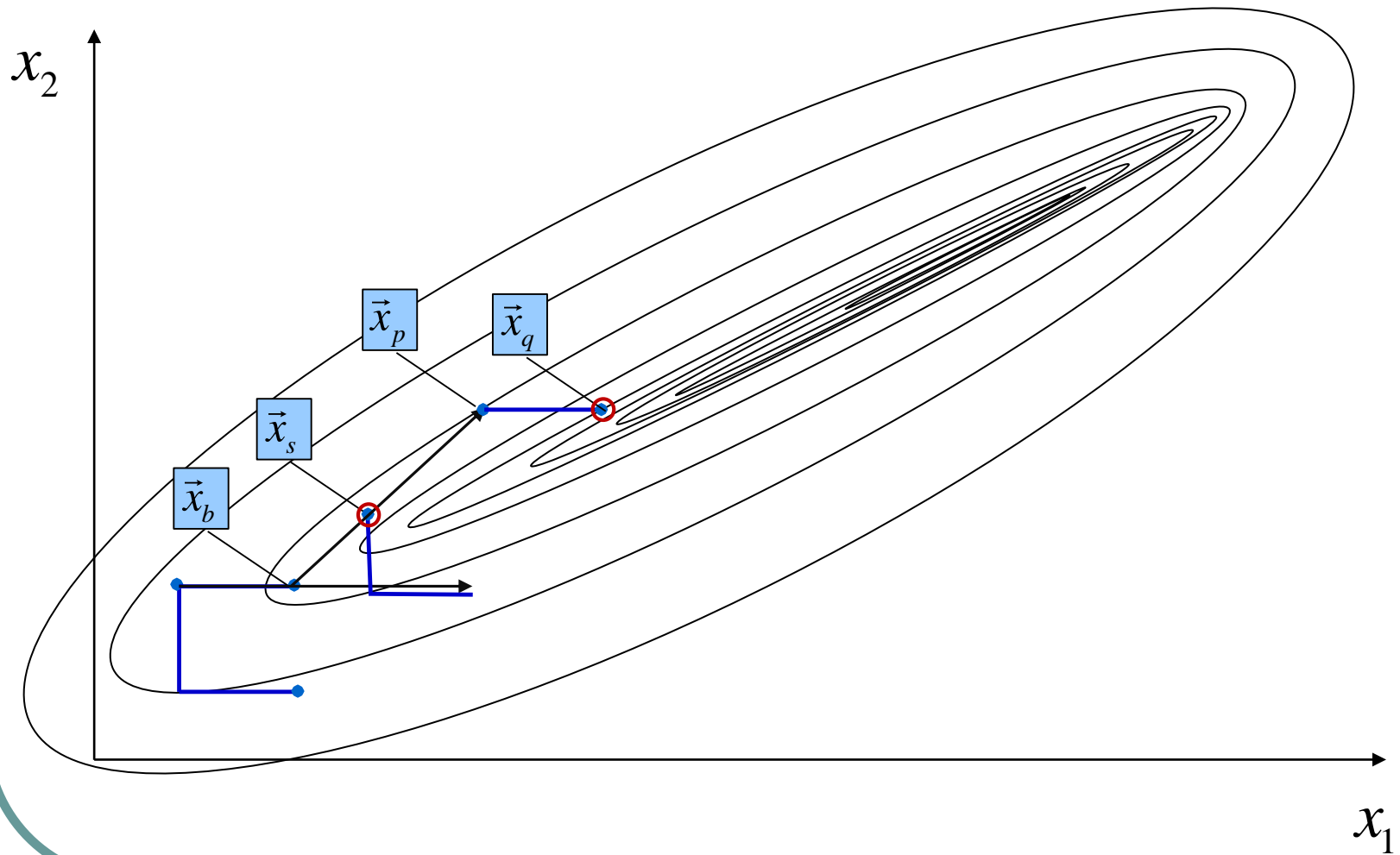
Нелинейное программирование. Безусловная минимизация функций n переменных. Метод Хука-Дживса



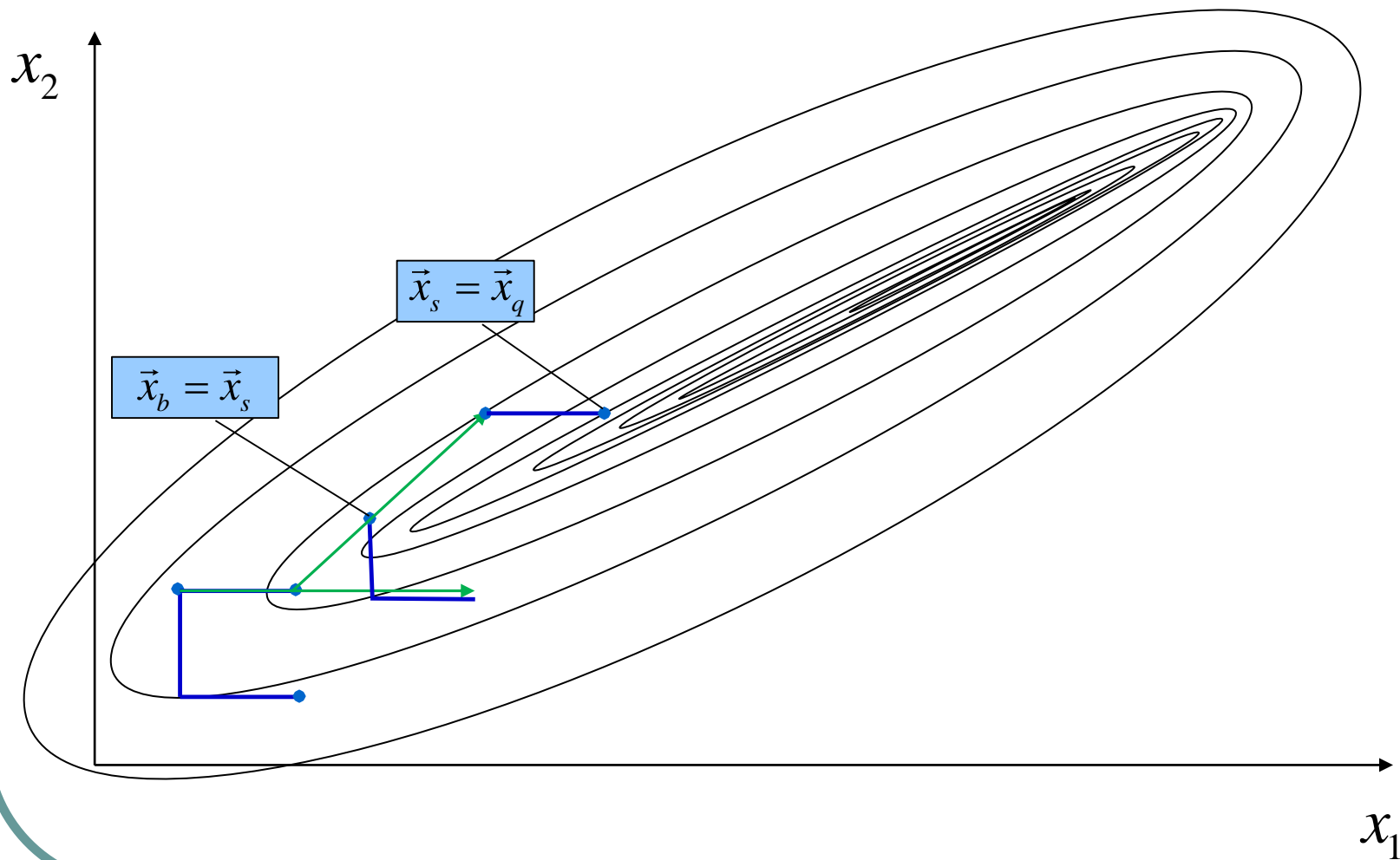
Нелинейное программирование.

Безусловная минимизация функций n переменных.

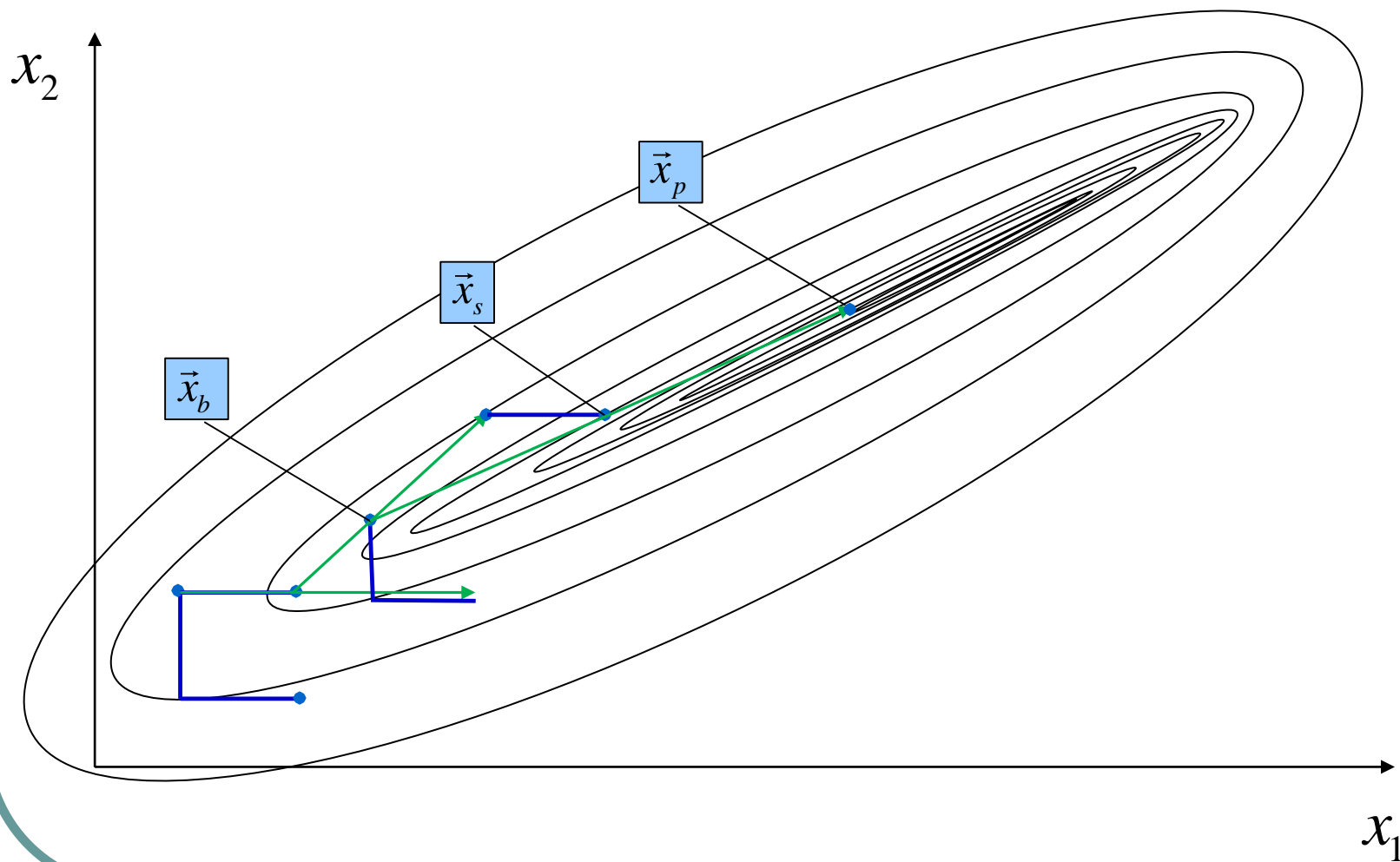
Метод Хука-Дживса



Нелинейное программирование.
Безусловная минимизация функций n переменных.
Метод Хука-Дживса



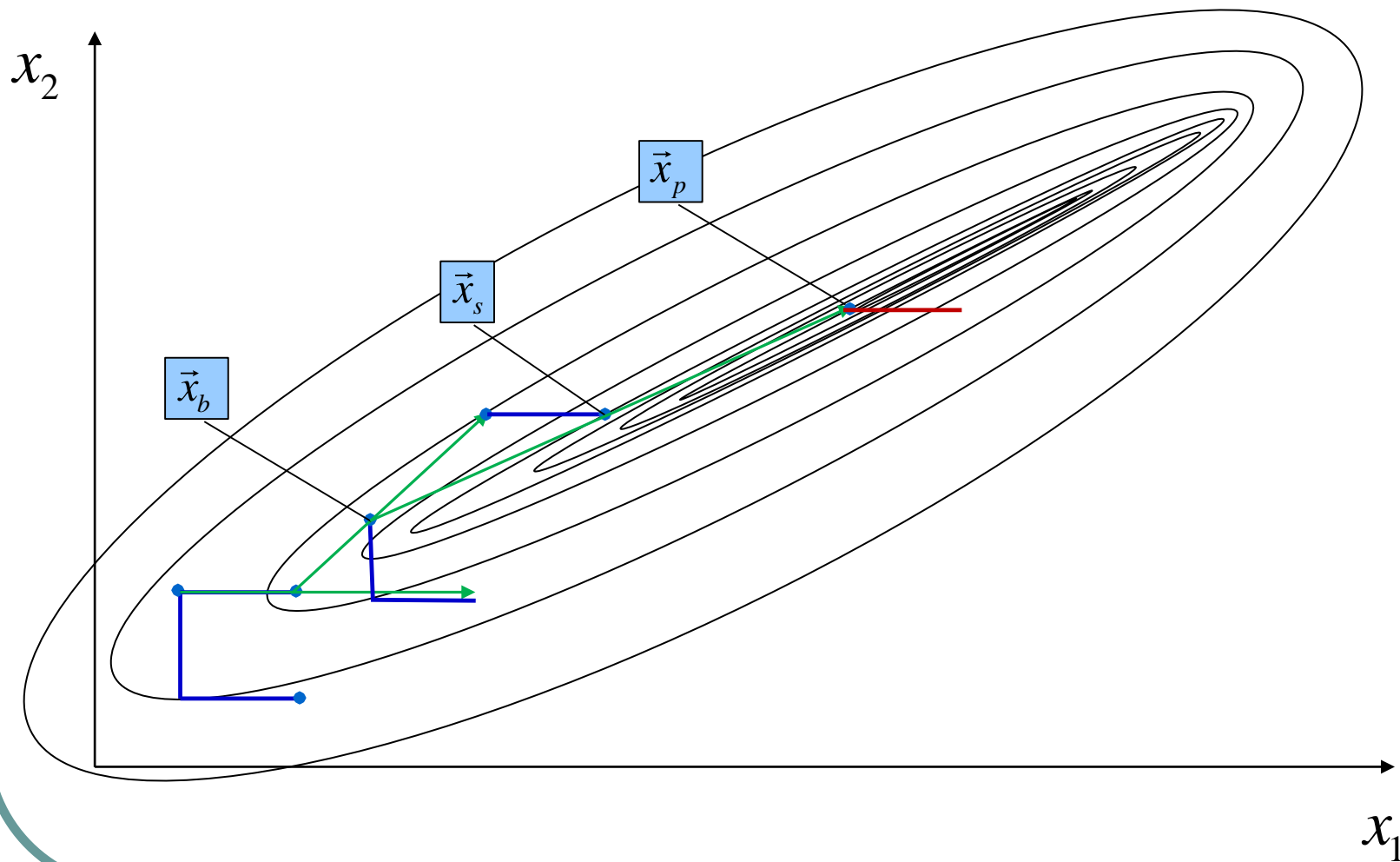
Нелинейное программирование.
Безусловная минимизация функций n переменных.
Метод Хука-Дживса



Нелинейное программирование.

Безусловная минимизация функций n переменных.

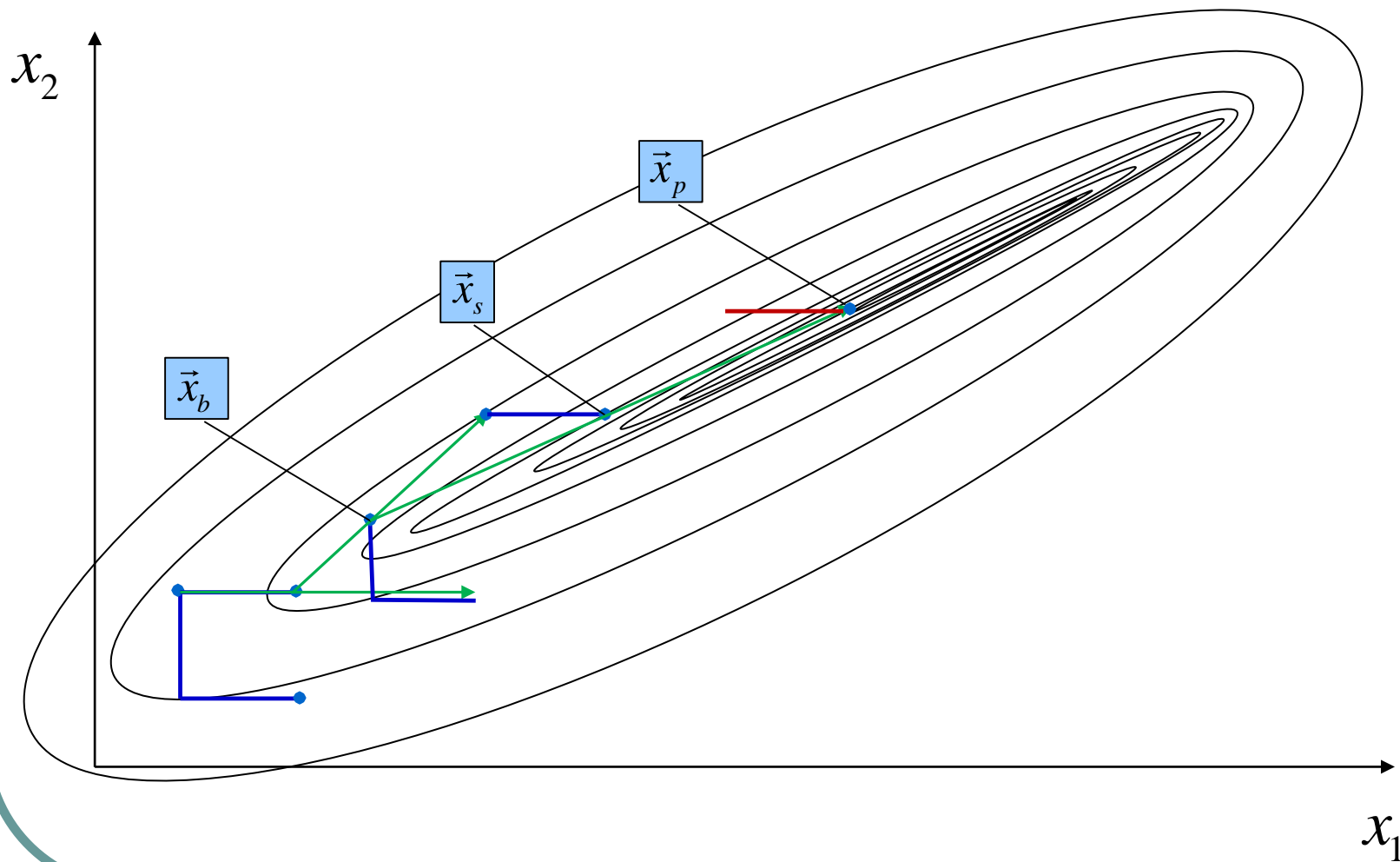
Метод Хука-Дживса



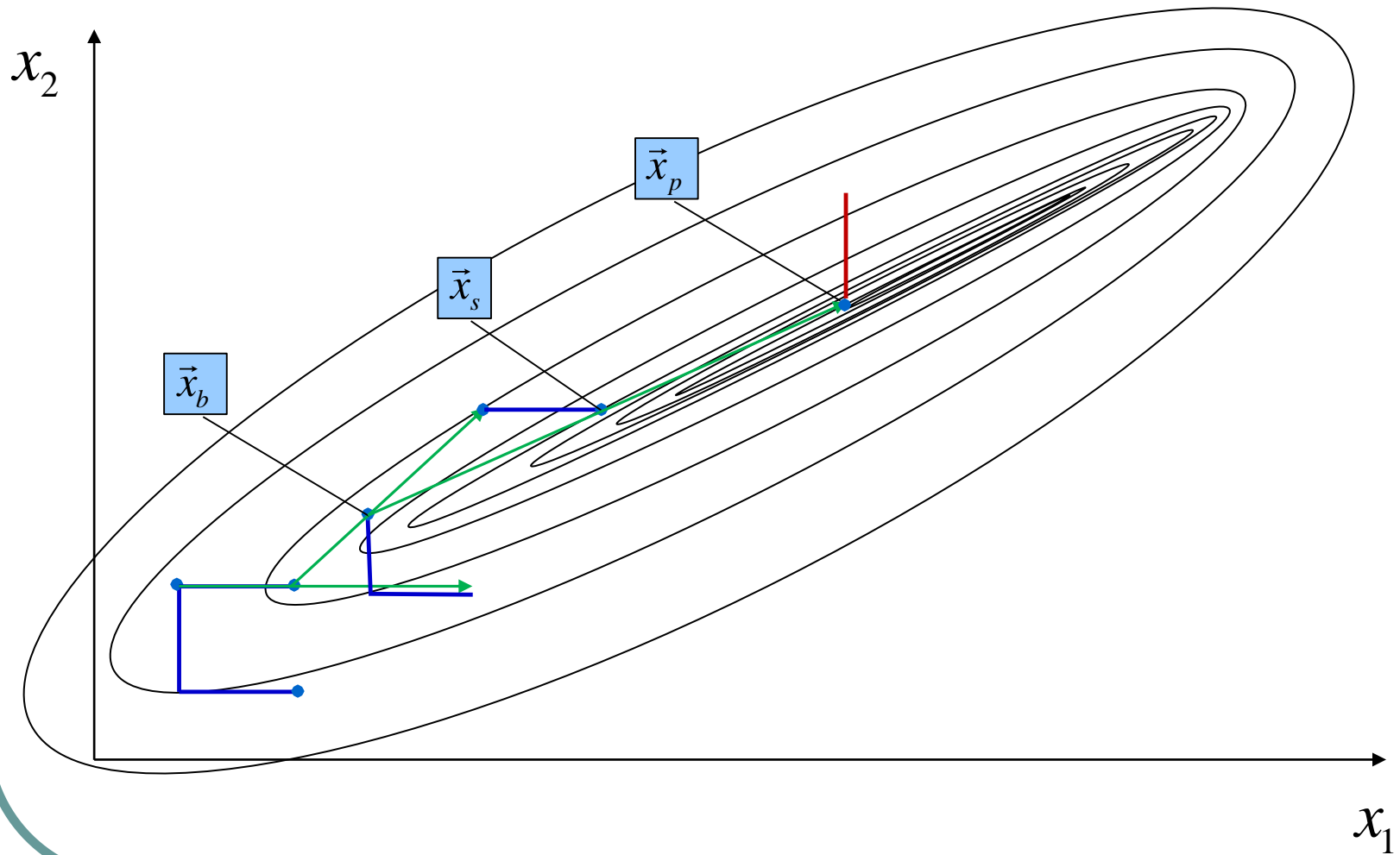
Нелинейное программирование.

Безусловная минимизация функций n переменных.

Метод Хука-Дживса



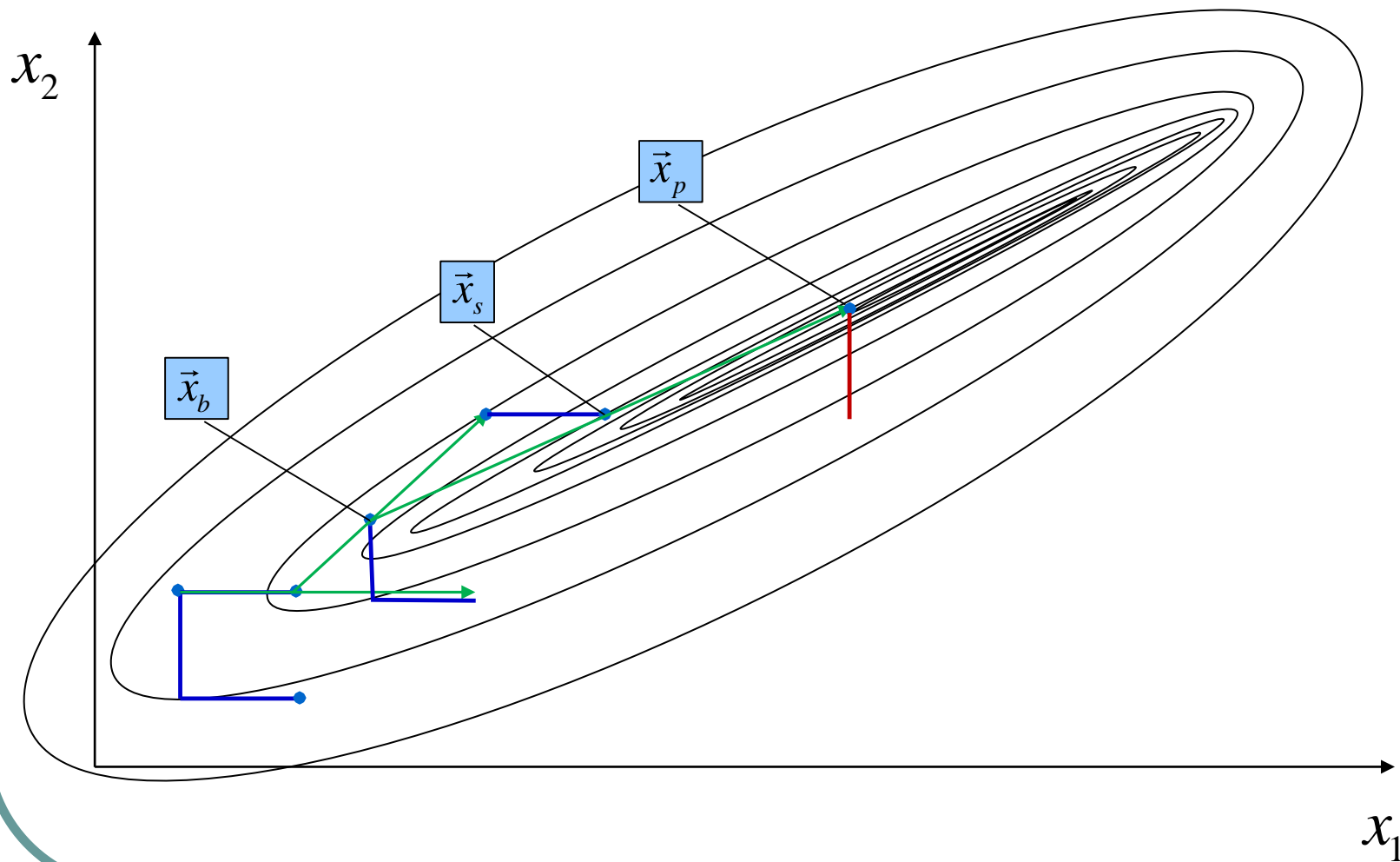
Нелинейное программирование.
Безусловная минимизация функций n переменных.
Метод Хука-Дживса



Нелинейное программирование.

Безусловная минимизация функций n переменных.

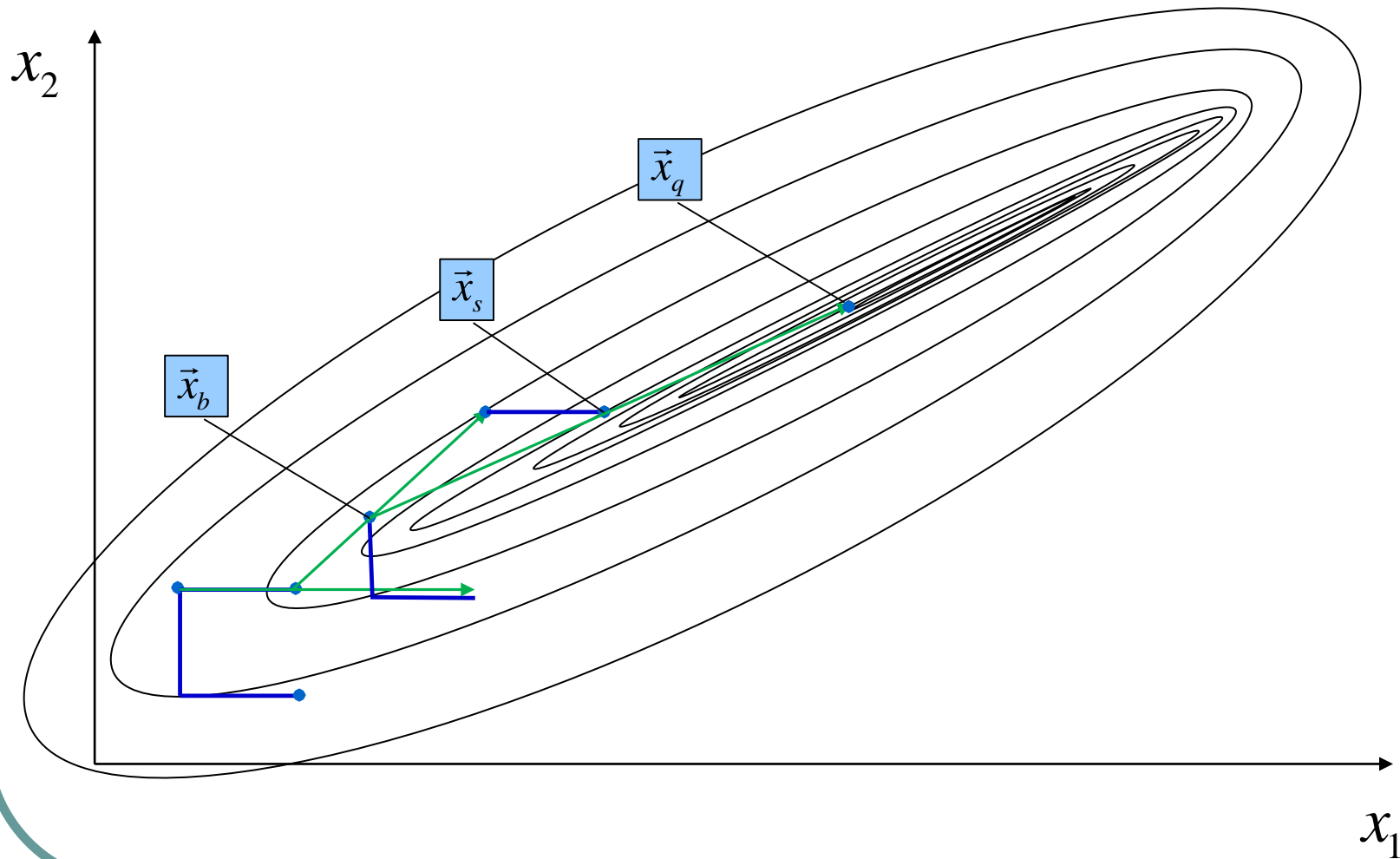
Метод Хука-Дживса



Нелинейное программирование.

Безусловная минимизация функций n переменных.

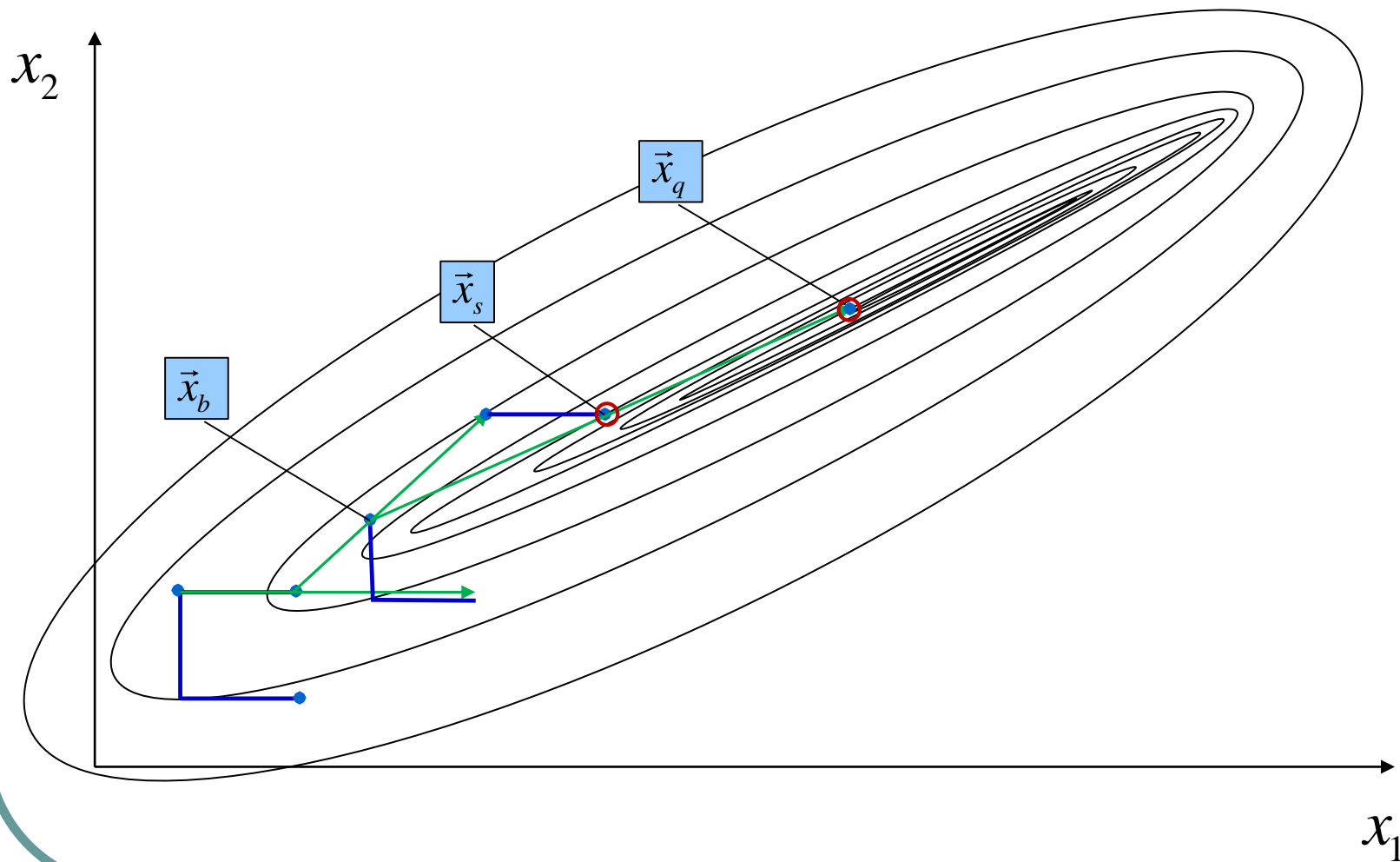
Метод Хука-Дживса



Нелинейное программирование.

Безусловная минимизация функций n переменных.

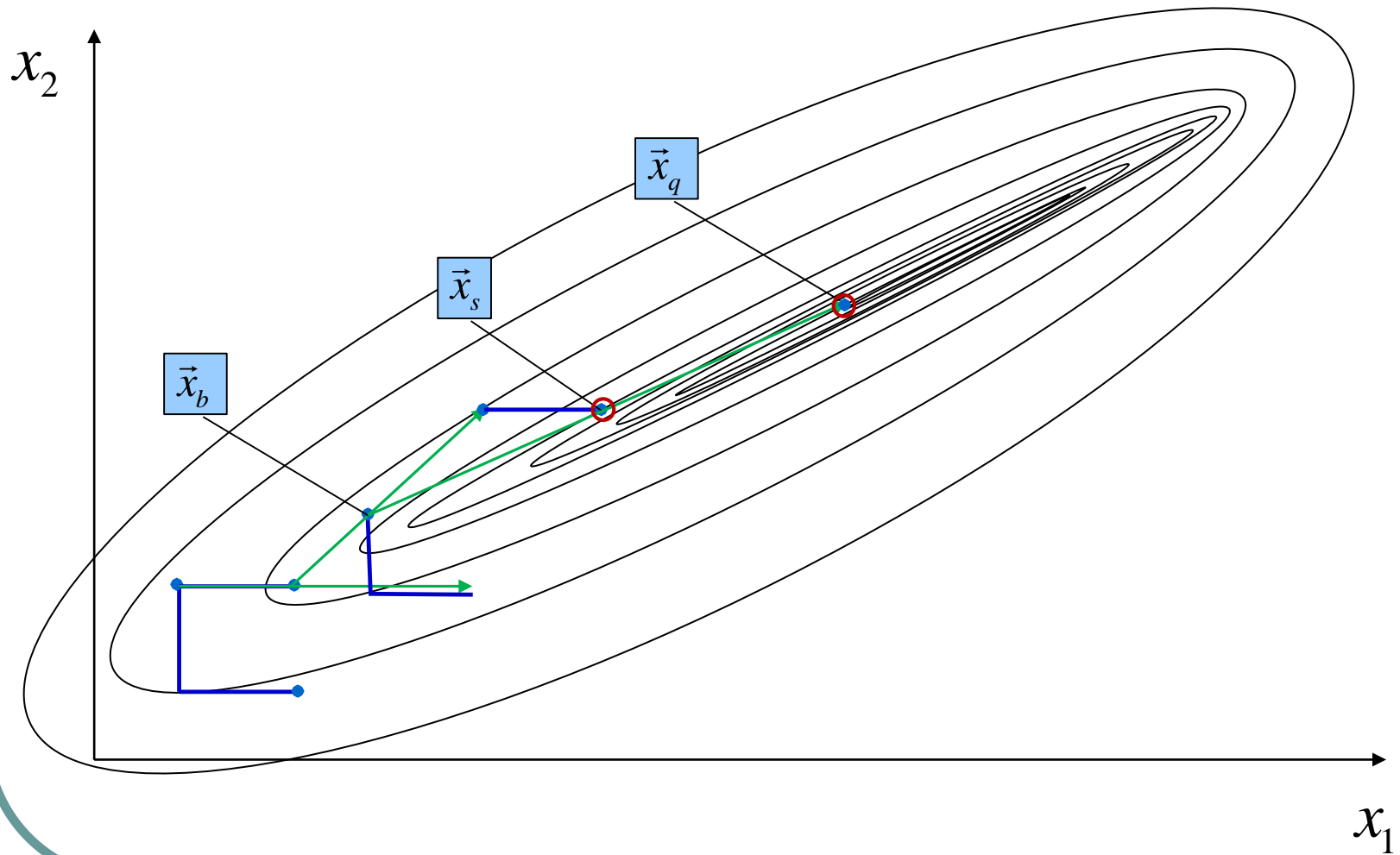
Метод Хука-Дживса



Нелинейное программирование.

Безусловная минимизация функций n переменных.

Метод Хука-Дживса



Нелинейное программирование.

Безусловная минимизация функций n переменных.

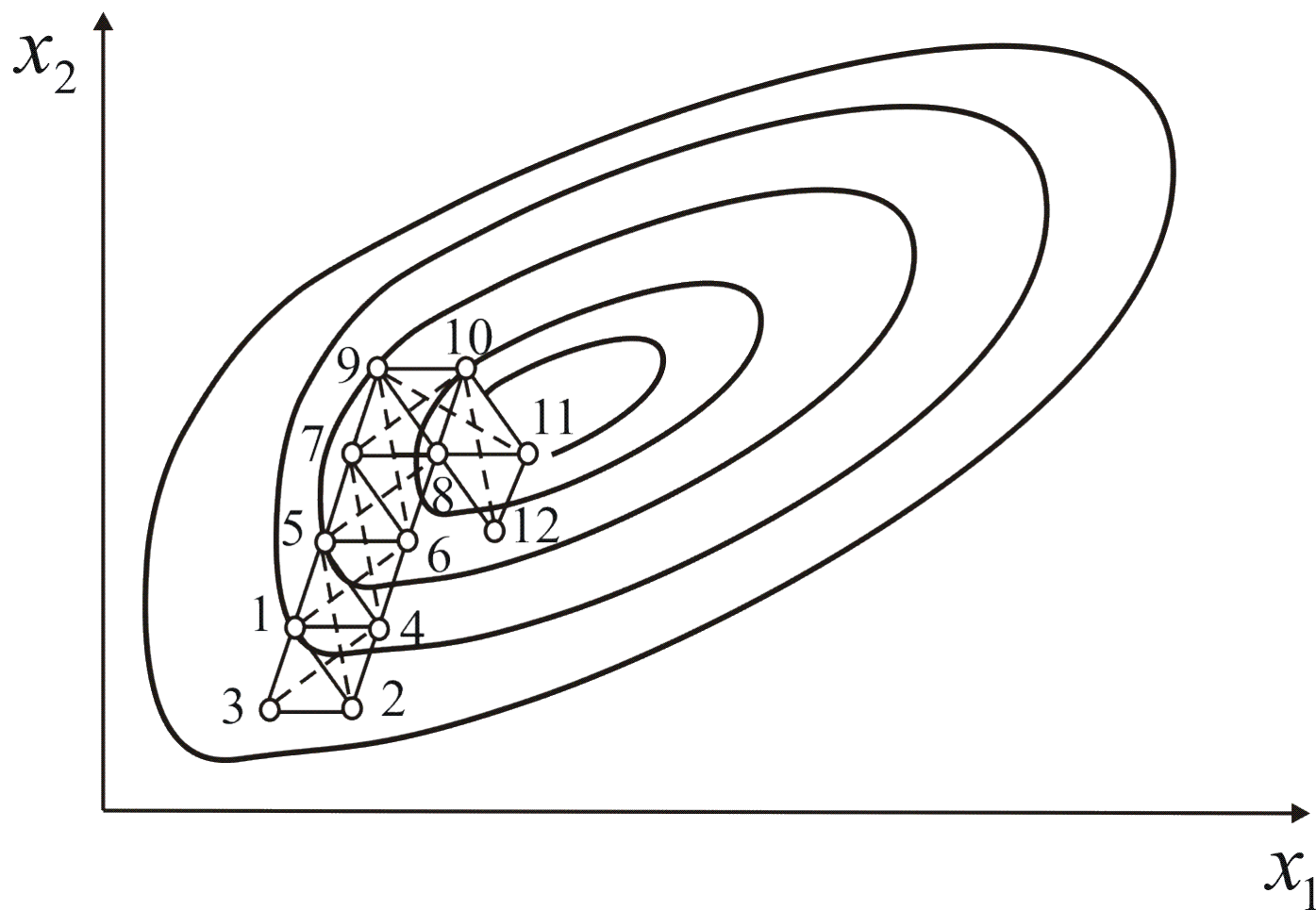
Метод Хука-Дживса

1. Выбрать \vec{x}_0 , задать \vec{h}, ε ;
2. $\vec{x}_b = \vec{x}_0$; $\Delta := |\vec{h}|$;
3. Покоординатный поиск из \vec{x}_b с шагом $\vec{h} \rightarrow \vec{x}_s$
4. Если $\vec{x}_s = \vec{x}_b$, то
 - 4.1 $\vec{h} = \vec{h} / 10$; $\Delta = \Delta / 10$;
 - 4.2 Если $\Delta < \varepsilon$, то
 - 4.2.1 $\vec{x}^* = \vec{x}_s$; Завершить работу;
 - Иначе
 - 4.2.2 перейти к шагу 3;
- Иначе
 - 4.3 $\vec{x}_p = \vec{x}_b + \lambda(\vec{x}_s - \vec{x}_b)$; $\lambda \sim 2,3-2,5$; $\vec{x}_b = \vec{x}_s$
 - 4.4 Покоординатный поиск из \vec{x}_p с шагом $\vec{h} \rightarrow \vec{x}_q$;
 - 4.5 Если $F(\vec{x}_q) < F(\vec{x}_s)$, то
 - 4.5.1 $\vec{x}_s = \vec{x}_q$, перейти к шагу 6
 - иначе
 - 4.5.2 перейти к шагу 3.

Нелинейное программирование.

Безусловная минимизация функций n переменных.

Симплекс-метод нелинейного программирования

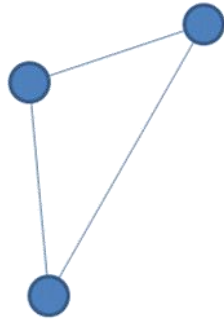


Нелинейное программирование.

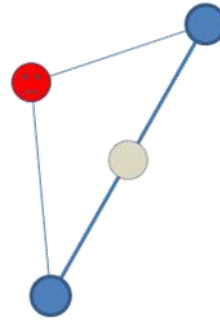
Безусловная минимизация функций n переменных.

Метод Нелдера-Мида

b - best
 g - good
 w - worst
 m - mid
 r - reflected
 e – expanded
 c – contracted
 s - shrunk

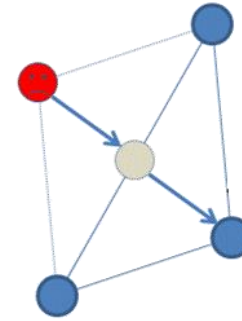


1: Initial Simplex



2: Center of gravity
(without the worst point)

$$\vec{m} = \left(\frac{b_1 + g_1}{2}, \frac{b_2 + g_2}{2} \right)$$



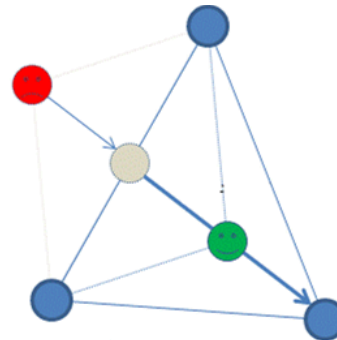
3: Reflection

$$\vec{r} = \vec{m} + \alpha (\vec{m} - \vec{w})$$



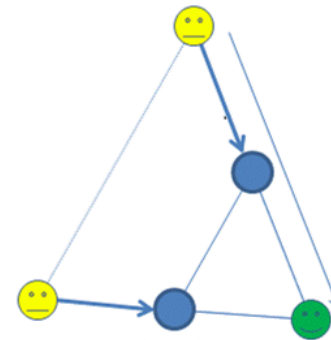
4a: Contraction

$$\vec{c} = \vec{m} + \beta (\vec{w} - \vec{m})$$



4b: Expansion

$$\vec{e} = \vec{m} + \gamma (\vec{r} - \vec{w})$$



5: Shrinkage

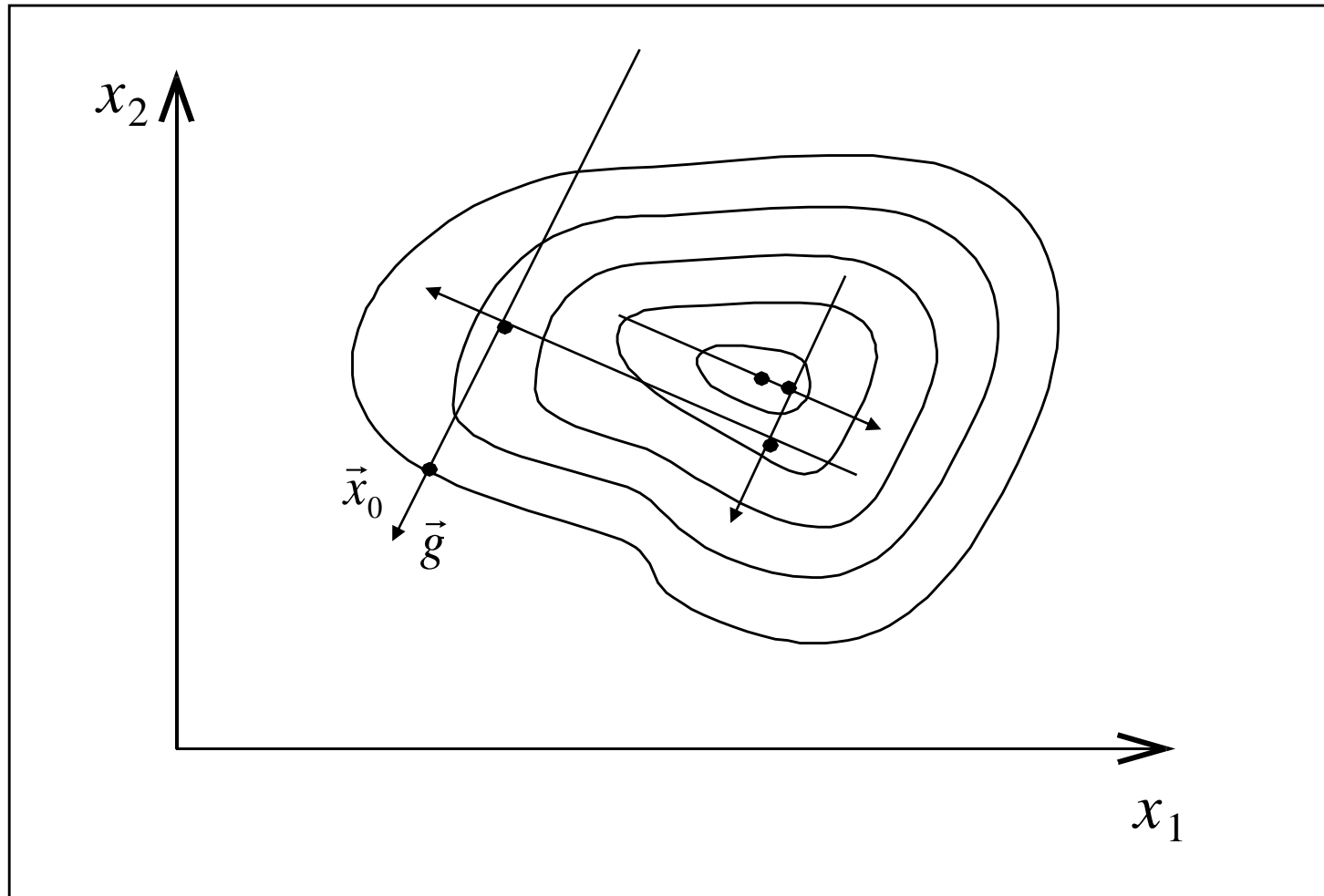
$$\vec{s} = \vec{b} + \delta (\vec{x}^j - \vec{m})$$

Безусловная минимизация функций n переменных. Градиентные методы

Нелинейное программирование.

Безусловная минимизация функций n переменных.

Метод наискорейшего спуска



Нелинейное программирование.

Безусловная минимизация функций n переменных.

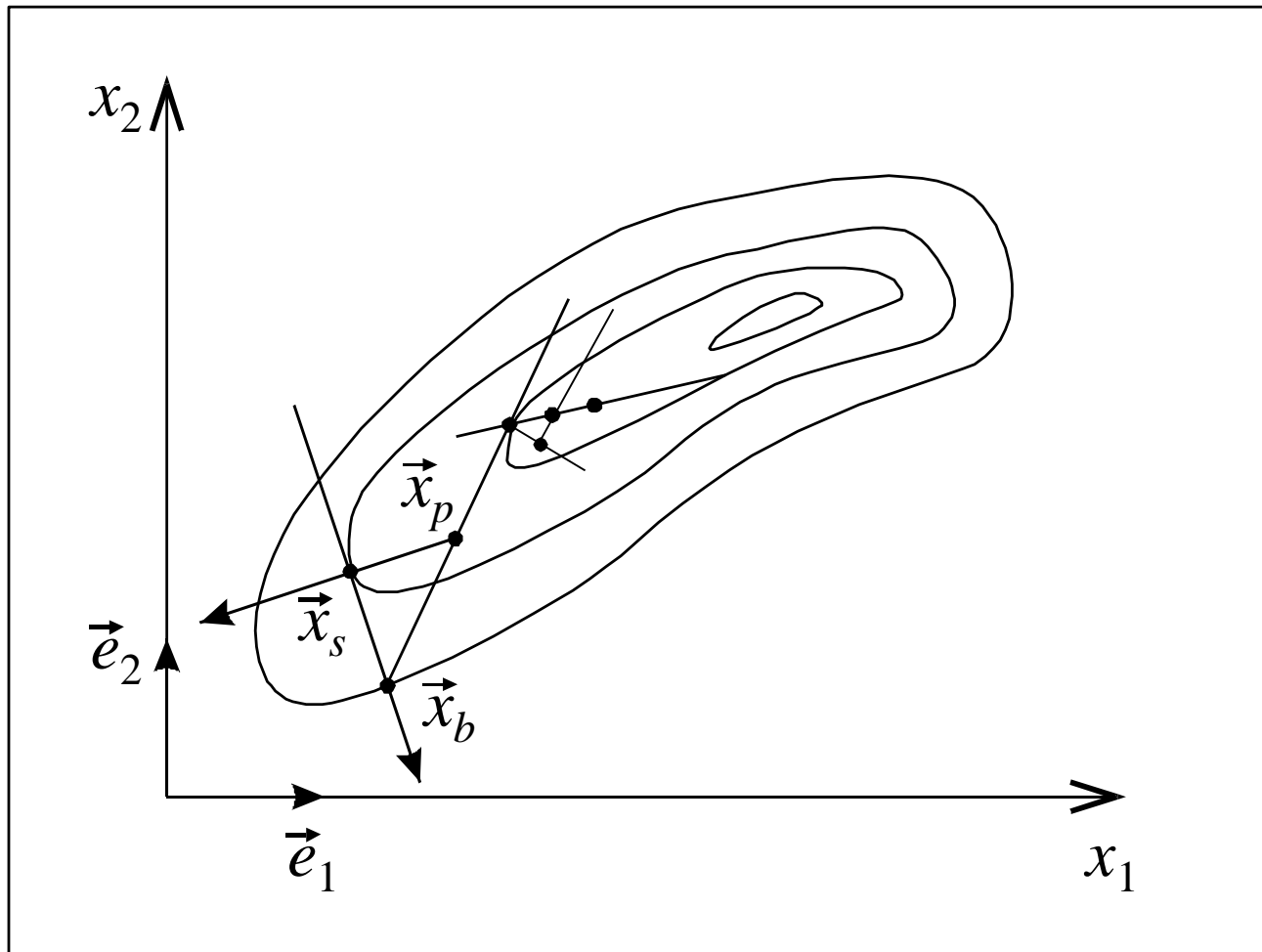
Метод наискорейшего спуска

1. *Выбрать \vec{x}_0 , задать ε , ;*
2. $\vec{x}_b = \vec{x}_0$; $\Delta = 2\varepsilon$;
3. *Пока $\Delta > \varepsilon$ выполнить*
 4. $\vec{g} = \nabla F(x)|_{\vec{x}=\vec{x}_b}$; $\vec{h} = -\vec{g}$; $\Delta = |\vec{g}|$;
 5. $\min_{\lambda} F(\vec{x}_b + \lambda \vec{h}) \rightarrow \lambda^*$;
 6. $\vec{x}_s = \vec{x}_b + \lambda^* \vec{h}$;
 7. $\vec{x}_b = \vec{x}_s$;
8. $\vec{x}^* = \vec{x}_s$; *Завершить работу;*

Нелинейное программирование.

Безусловная минимизация функций n переменных.

Метод градиентного спуска с ускорением



Нелинейное программирование.

Безусловная минимизация функций n переменных.

Метод градиентного спуска с ускорением

1. Выбрать \vec{x}_0 , задать ε , ; $\vec{x}_b = \vec{x}_0$; $\Delta = 2\varepsilon$;

2. Пока $\Delta > \varepsilon$ выполнить

$$4. \vec{g} = \nabla F(x) \Big|_{\vec{x}=\vec{x}_b}; \vec{h} = -\vec{g};$$

$$5. \min_{\lambda} F(\vec{x}_b + \lambda \vec{h}) \rightarrow \lambda^*; \vec{x}_s = \vec{x}_b + \lambda^* \vec{h};$$

$$7. \vec{g} = \nabla F(x) \Big|_{\vec{x}=\vec{x}_s}; \vec{h} = -\vec{g};$$

$$8. \min_{\lambda} F(\vec{x}_s + \lambda \vec{h}) \rightarrow \lambda^*; \vec{x}_p = \vec{x}_s + \lambda^* \vec{h};$$

$$9. \vec{h} = \vec{x}_p - \vec{x}_b; \Delta = \max\{|\vec{g}|, |\vec{h}|\};$$

$$10. \min_{\lambda} F(\vec{x}_p + \lambda \vec{h}) \rightarrow \lambda^*; \vec{x}_b = \vec{x}_p + \lambda^* \vec{h};$$

11. $\vec{x}^* = \vec{x}_b$; Завершить работу;

Нелинейное программирование.

Безусловная минимизация функций n переменных.

Что есть еще?

Метод тяжелого шарика

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

Метод Нестерова

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

**Стохастический
градиентный спуск**

$$\nabla F \rightarrow \vec{G}(x, \theta) : M[\vec{G}(x, \theta)] = \nabla F$$

$$\vec{x}_{k+1} = \vec{x}_k - \alpha_k \vec{G}(\vec{x}_k, \theta_k)$$

Для дифференцируемой выпуклой функции

Субградиентный спуск

$$F(\vec{x}_j) \geq \vec{x}_k + \nabla F(\vec{x}_k)^T (\vec{x}_j - \vec{x}_k) \quad \forall x_k, x_j$$

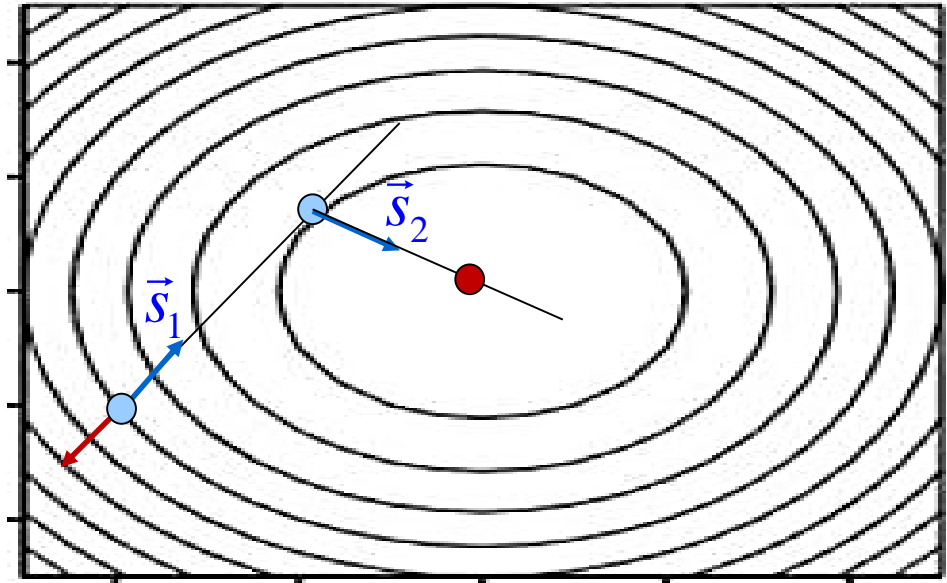
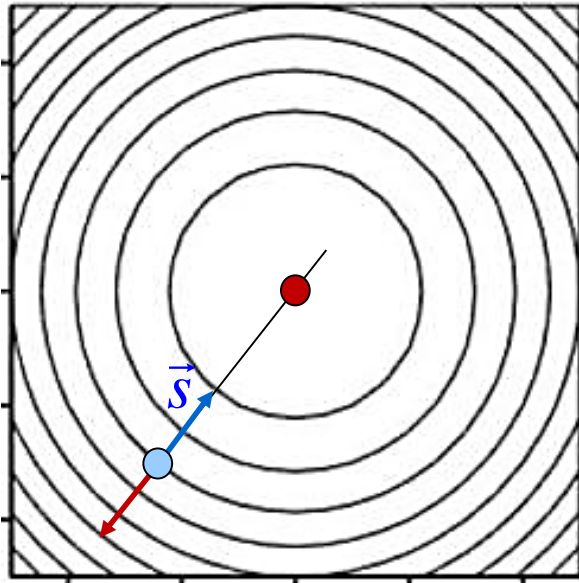
Для недифференцируемой выпуклой функции

$$\forall x_k \quad \exists \vec{G}, \quad F(\vec{x}_j) \geq \vec{x}_k + \vec{G}(\vec{x}_k)^T (\vec{x}_j - \vec{x}_k) \quad \forall x_j$$

Нелинейное программирование.

Безусловная минимизация функций n переменных.

Развитие градиентных методов



$$F(\vec{x}) = a + \vec{b}^T \vec{x} + \frac{1}{2} \vec{x}^T \mathbf{G} \vec{x} \quad \mathbf{G}_{ij} = \mathbf{H}_{ij} = \frac{\partial}{\partial x_i \partial x_j} \left(a + \vec{b}^T \vec{x} + \frac{1}{2} \vec{x}^T \mathbf{G} \vec{x} \right)$$

$$\vec{s}_i^T \mathbf{G} \vec{s}_j = 0, \quad i \neq j, \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, n;$$

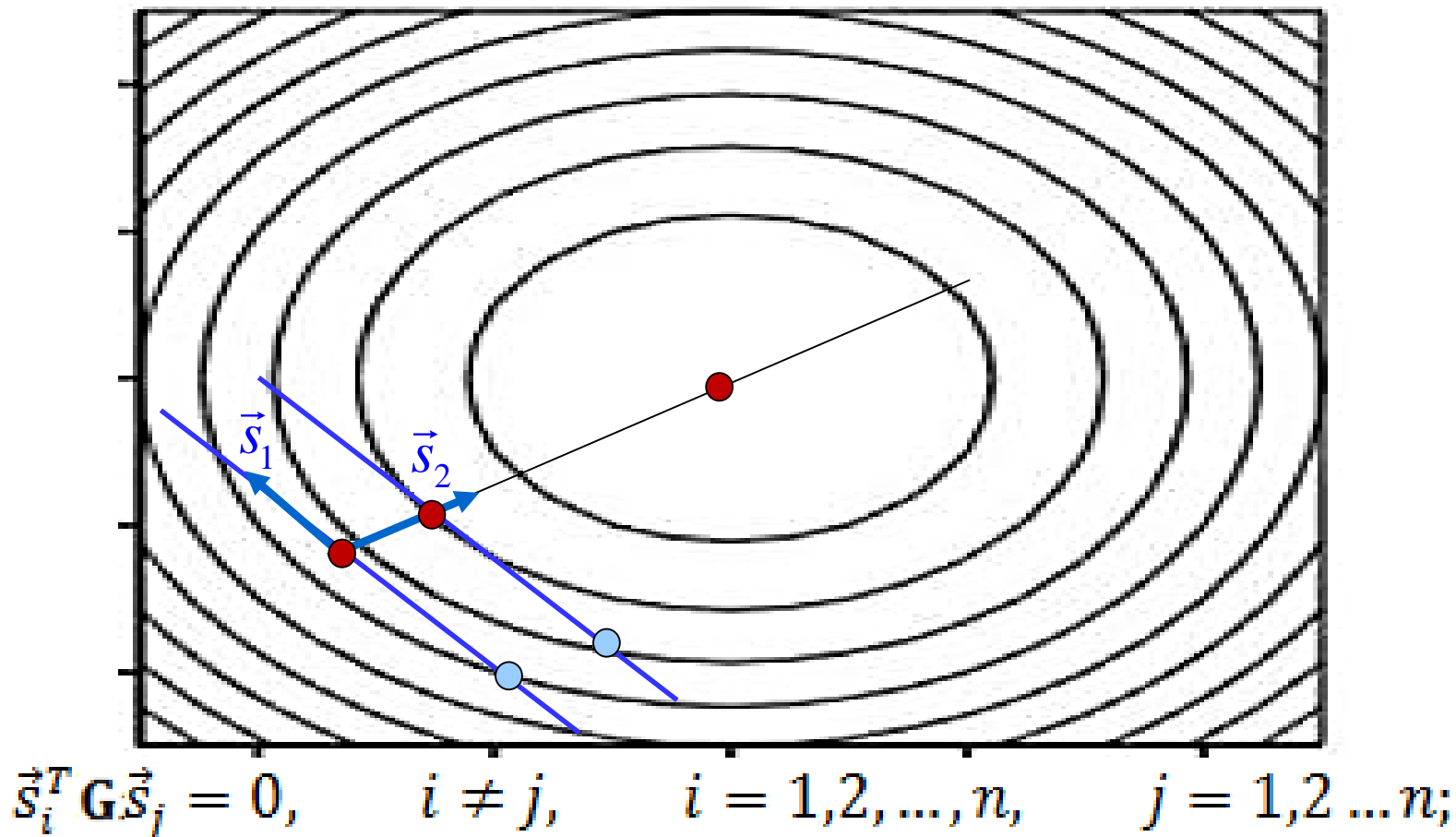
$$\vec{s}_i^T \mathbf{G} \vec{s}_i \geq 0, \quad i = 1, 2, \dots, n;$$

Нелинейное программирование.

Безусловная минимизация функций n переменных.

Развитие градиентных методов

Идея метода параллельных касательных (метода Пауэлла)



Нелинейное программирование.

Безусловная минимизация функций n переменных.

Развитие градиентных методов

$$F(\vec{x}) = a + \vec{b}^T \vec{x} + \frac{1}{2} \vec{x}^T \mathbf{G} \vec{x}$$

$$\mathbf{G}_{ij} = \mathbf{H}_{ij} = \frac{\partial}{\partial x_i \partial x_j} \left(a + \vec{b}^T \vec{x} + \frac{1}{2} \vec{x}^T \mathbf{G} \vec{x} \right)$$

$$\vec{s}_i^T \mathbf{H} \vec{s}_j = 0, \quad i \neq j, \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, n;$$

$$\vec{s}_i^T \mathbf{H} \vec{s}_i \geq 0, \quad i = 1, 2, \dots, n;$$

Можно доказать, что для произвольной квадратичной формы с положительно определенной матрицей \mathbf{G} , задав n сопряженных относительно \mathbf{G} направлений $\vec{s}_1, \vec{s}_2, \dots, \vec{s}_n$, можно найти минимум данной формы, последовательно выполнив n одномерных поисков вдоль этих направлений.

Нелинейное программирование.

Безусловная минимизация функций n переменных.

Метод сопряженных градиентов

$$\vec{s}_1 = -\vec{g}(\vec{x}_1)$$

$$\min_{\lambda} F(\vec{x}_1 + \lambda \vec{s}_1) \text{ с результатом } \lambda_1 \Rightarrow \vec{x}_2 = \vec{x}_1 + \lambda_1 \vec{s}_1$$

$$\vec{s}_2 = -\vec{g}(\vec{x}_2) + \omega_2 \vec{s}_1 \quad \vec{s}_1^T \mathbf{H} \vec{s}_2 = 0 \text{ (матрица } \mathbf{H} \text{ вычисляется в точке } \vec{x}_1 \text{)}$$

$$\vec{g}(\vec{x}_2) \approx \vec{g}(\vec{x}_1) + \mathbf{H}(\vec{x}_2 - \vec{x}_1) \Rightarrow \vec{g}(\vec{x}_2) - \vec{g}(\vec{x}_1) \approx \mathbf{H}(\vec{x}_2 - \vec{x}_1) = \lambda_1 \mathbf{H} \vec{s}_1$$

$$(\vec{g}(\vec{x}_2) - \vec{g}(\vec{x}_1))^T \mathbf{H}^{-1} \approx \lambda_1 \vec{s}_1^T \mathbf{H}^T \mathbf{H}^{-1} \quad \mathbf{H}^T = \mathbf{H} \quad \mathbf{H}^T \mathbf{H}^{-1} = \mathbf{I} \Rightarrow$$

$$\vec{s}_1 = \frac{(\vec{g}(\vec{x}_2) - \vec{g}(\vec{x}_1))^T \mathbf{H}^{-1}}{\lambda_1} \xrightarrow{\vec{s}_1^T \mathbf{H} \vec{s}_2 = 0} \frac{(\vec{g}(\vec{x}_2) - \vec{g}(\vec{x}_1))^T \mathbf{H}^{-1} \mathbf{H} \vec{s}_2}{\lambda_1} = 0$$

$$\xrightarrow{\substack{\vec{s}_2 = -\vec{g}(\vec{x}_2) + \omega_2 \vec{s}_1 \\ \vec{s}_1 = -\vec{g}(\vec{x}_1)}} (\vec{g}(\vec{x}_2) - \vec{g}(\vec{x}_1))^T (-\vec{g}(\vec{x}_2) - \omega_2 \vec{g}(\vec{x}_1)) = 0 \Rightarrow \omega_2 = \frac{|\vec{g}(\vec{x}_2)|^2}{|\vec{g}(\vec{x}_1)|^2}$$

Нелинейное программирование.

Безусловная минимизация функций n переменных.

Метод сопряженных градиентов

$$\vec{s}_2 = -\vec{g}(\vec{x}_2) + \frac{|\vec{g}(\vec{x}_2)|^2}{|\vec{g}(\vec{x}_1)|^2} \vec{s}_1$$

$$\vec{s}_{k+1} = -\vec{g}(\vec{x}_{k+1}) + \omega_{k+1} \vec{s}_k$$

$$\omega_{k+1} = \frac{|\vec{g}(\vec{x}_{k+1})|^2}{|\vec{g}(\vec{x}_k)|^2} \quad (\text{Метод Флетчера - Ривса})$$

$$\vec{g}(\vec{x}_2) \approx \vec{g}(\vec{x}_1) + \mathbf{H}(\vec{x}_2 - \vec{x}_1) \quad (!!!!)$$

$$\omega_{k+1} = \max \left(0, \frac{-g(\vec{x}_{k+1}) \cdot (-g(\vec{x}_{k+1}) + g(\vec{x}_k))}{|\vec{g}(\vec{x}_k)|^2} \right) \quad (\text{Метод Полака - Райбера})$$

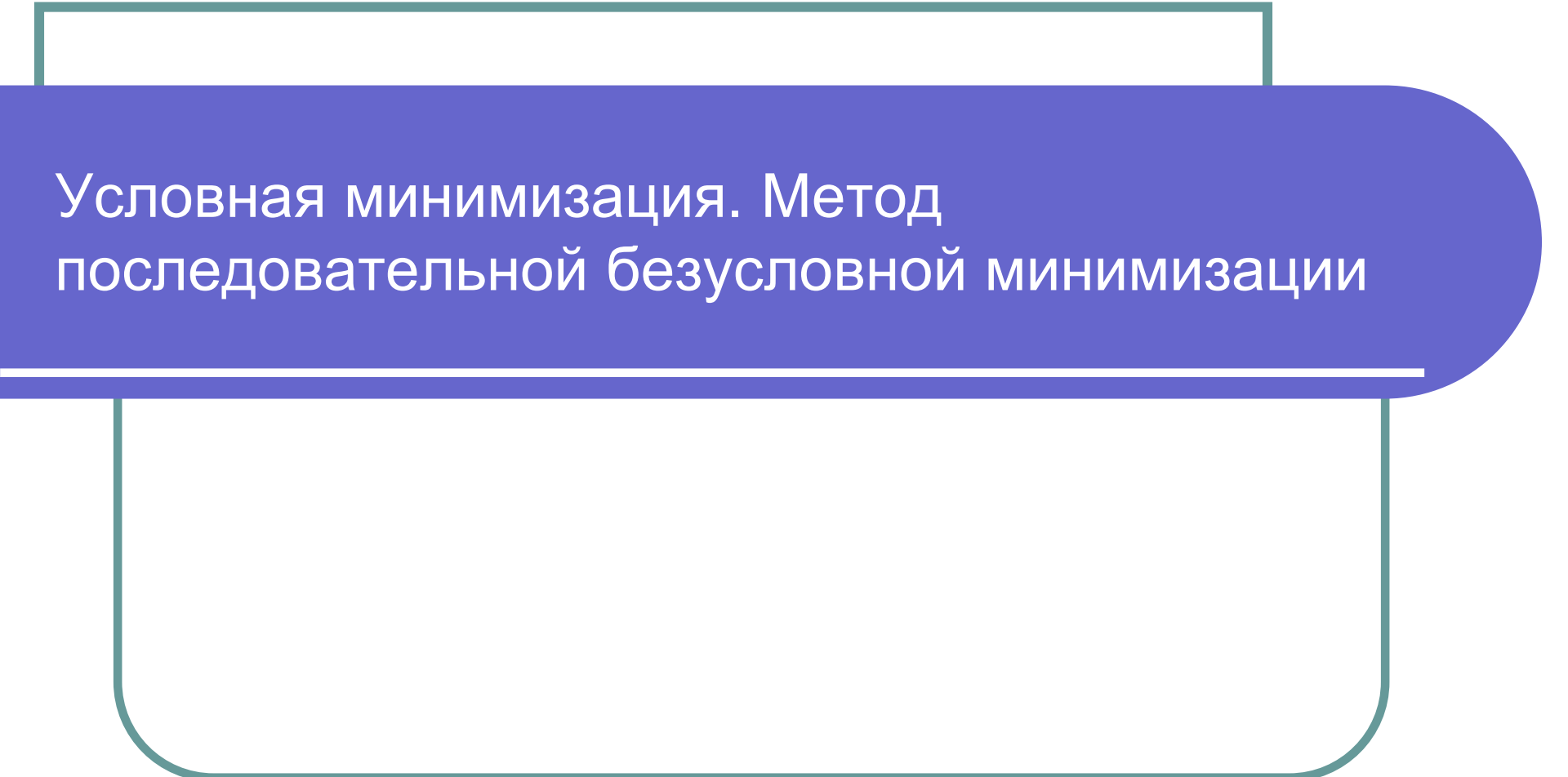
Нелинейное программирование.

Безусловная минимизация функций n переменных.

Метод сопряженных градиентов

Алгоритм метода Полака - Райбера)

1. Задать начальную точку \vec{x}^0 , значения $\varepsilon_1, \varepsilon_2$, предельное число шагов поиска N .
2. Положить $\vec{x}_b := \vec{x}^0$; $\vec{x}_p = \vec{x}_b$; $\vec{g}_p = \nabla F(\vec{x})_{\vec{x}=\vec{x}_p}$
3. Для r от 1 до N с шагом +1 выполнять цикл
 1. Положить $\vec{s} = -\vec{g}_p$
 2. Для i от 1 до n с шагом +1 выполнять цикл
 1. Решить задачу $\min_{\lambda} F(\vec{x}_p + \lambda \vec{s})$ с результатом λ^* .
 2. Положить $\vec{x}_p := \vec{x}_p + \lambda^* \vec{s}$; $\vec{g}_q = \nabla F(\vec{x})_{\vec{x}=\vec{x}_p}$
 3. Положить $\omega = \max\left(0, \frac{-\vec{g}_q \cdot (-\vec{g}_q + \vec{g}_p)}{|\vec{g}_p|^2}\right)$
 4. Положить $\vec{s} = -\vec{g}_q + \omega \vec{s}$; $\vec{g}_p = \vec{g}_q$
 3. Если $|\vec{x}_p - \vec{x}_b| < \varepsilon_1$ и $|\vec{g}_p| < \varepsilon_2$, то выйти из цикла
 4. Положить $\vec{x}_b := \vec{x}_p$.
4. Если $k = N$, то вывести предупреждение, что минимум не достигнут
5. Положить $\vec{x}^* := \vec{x}_p$.
6. Завершить работу



Условная минимизация. Метод
последовательной безусловной минимизации

Нелинейное программирование.

Условная минимизация

$$\begin{cases} \min F(\vec{x}), & \vec{x} = (x_1, x_2, \dots, x_n)^T \\ \vec{g}(\vec{x}) \geq 0, & \vec{g}(\vec{x}) = (g_1(\vec{x}), g_2(\vec{x}), \dots, g_m(\vec{x}))^T \end{cases}$$

Основные подходы к решению:

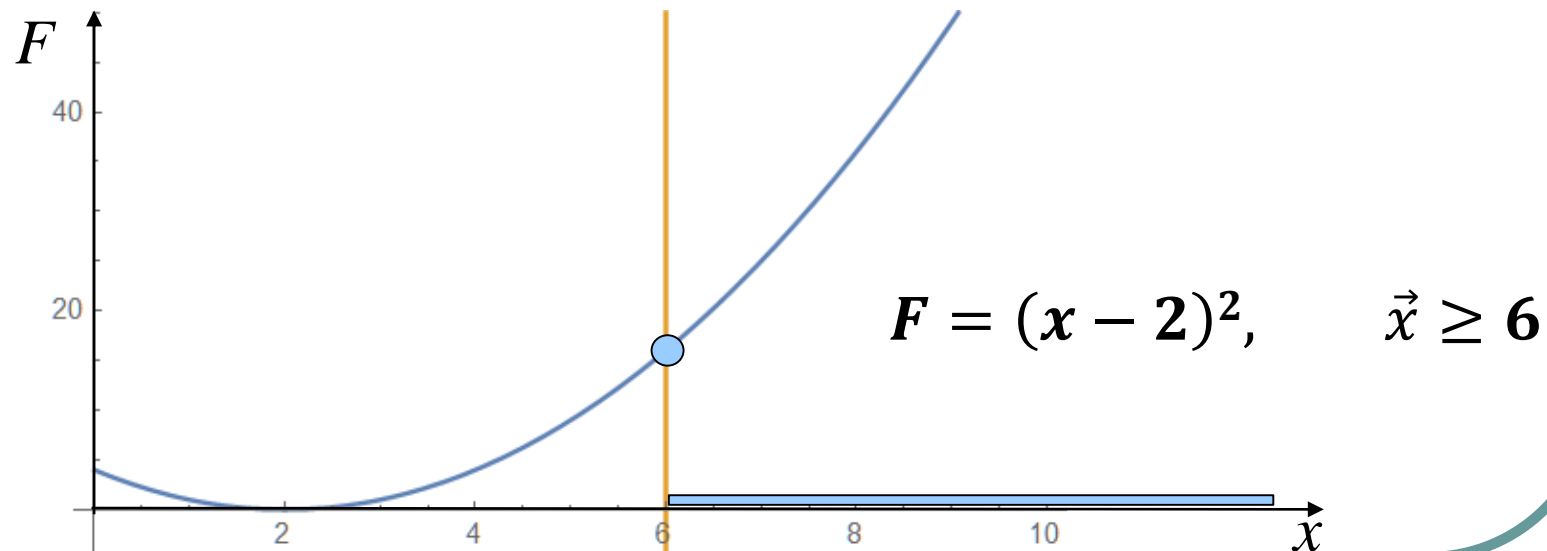
- Сведение задачи условной минимизации к последовательности задач безусловной минимизации путем использования барьерных или штрафных функций (метод последовательной безусловной минимизации).
- Применение метода скользящего допуска, позволяющего оперировать недопустимыми, но близкими к допустимым, векторами в пространстве поиска.
- Применение методов случайного поиска.

Нелинейное программирование.

Условная минимизация

$$\begin{cases} \min F(\vec{x}), & \vec{x} = (x_1, x_2, \dots, x_n)^T \\ \vec{g}(\vec{x}) \geq 0, & \vec{g}(\vec{x}) = (g_1(\vec{x}), g_2(\vec{x}), \dots, g_m(\vec{x}))^T \end{cases}$$

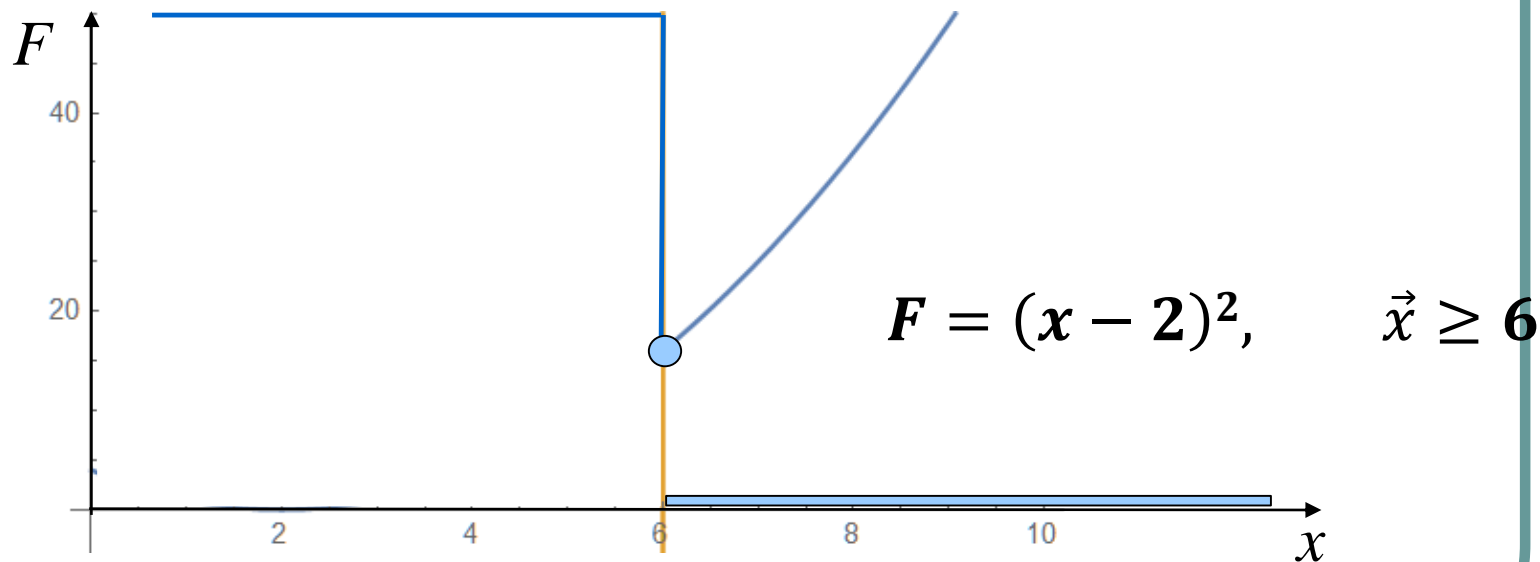
$$\hat{F}(\vec{x}) = \begin{cases} F(\vec{x}), & \text{если } \vec{g}(\vec{x}) \geq 0; \\ P, & \text{если } \exists j, g_j(\vec{x}) < 0. \end{cases} \quad P \gg F(\vec{x}), \quad \vec{x} \in D$$



Нелинейное программирование.

Условная минимизация.

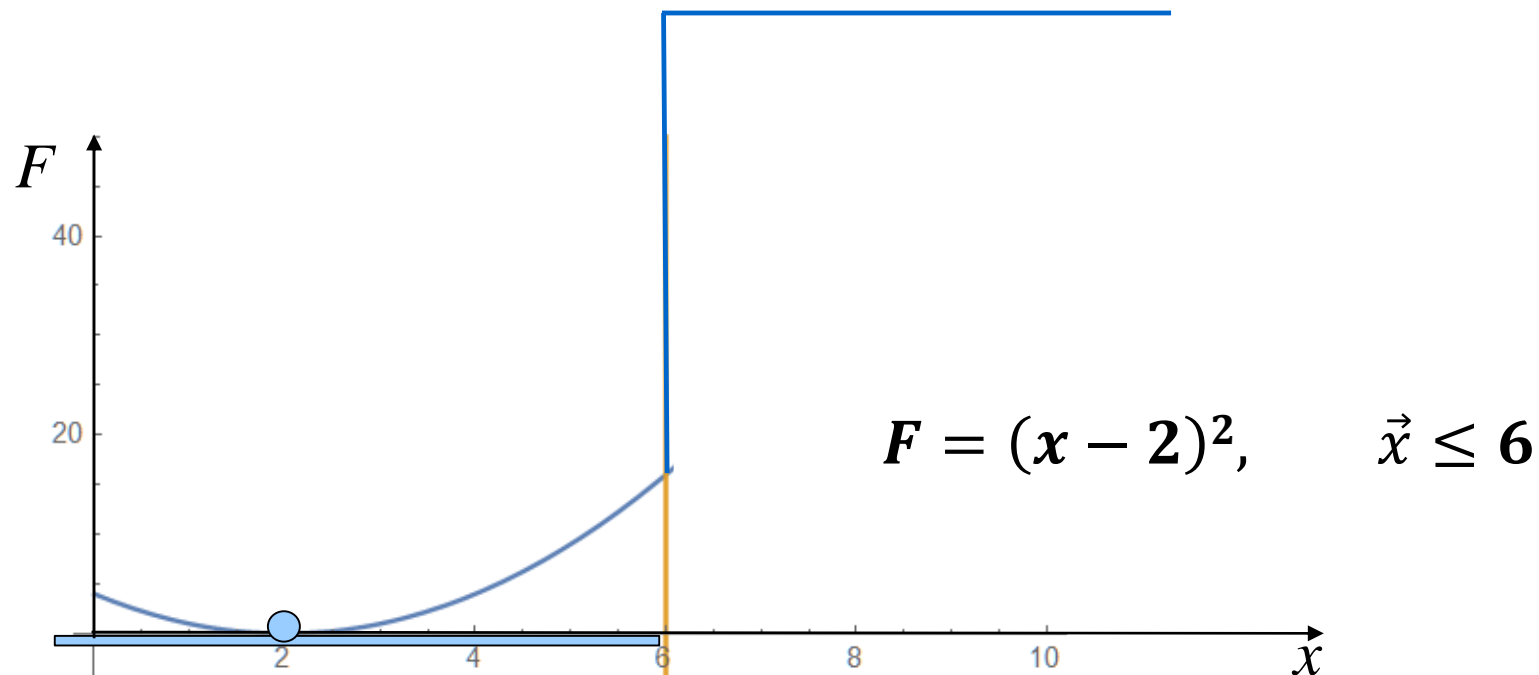
$$\hat{F}(\vec{x}) = \begin{cases} F(\vec{x}), & \text{если } \vec{g}(\vec{x}) \geq 0; \\ P, & \text{если } \exists j, g_j(\vec{x}) < 0. \end{cases} \quad P \gg F(\vec{x}), \quad \vec{x} \in D$$



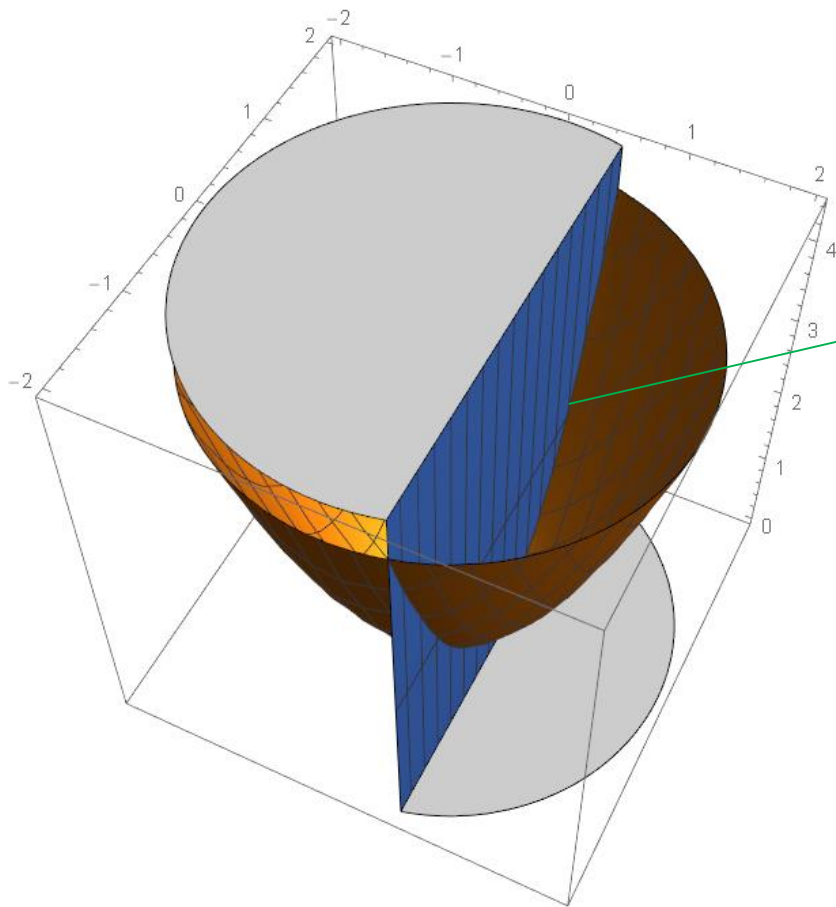
Нелинейное программирование.

Условная минимизация.

$$\hat{F}(\vec{x}) = \begin{cases} F(\vec{x}), & \text{если } \vec{g}(\vec{x}) \geq 0; \\ P, & \text{если } \exists j, g_j(\vec{x}) < 0. \end{cases} \quad P \gg F(\vec{x}), \quad \vec{x} \in D$$



Нелинейное программирование. Условная минимизация

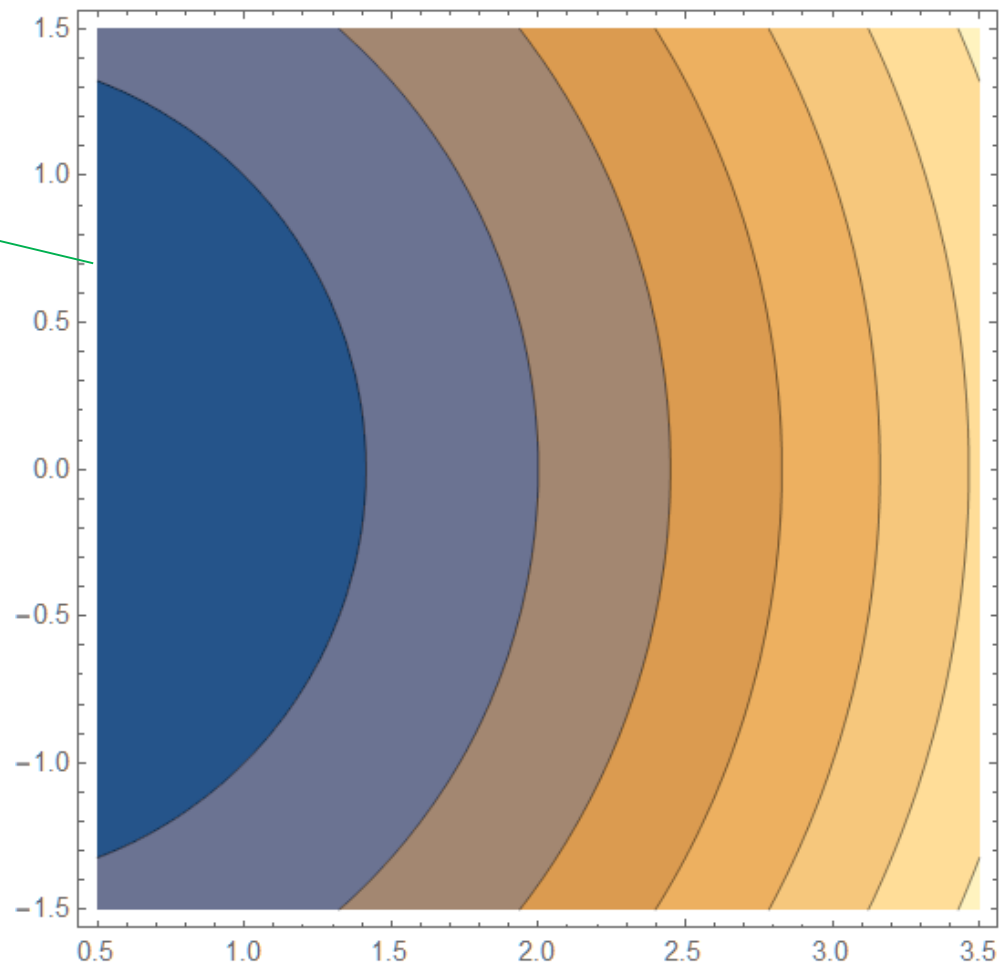


**Овраг бесконечной
крутизны**

Нелинейное программирование.

Условная минимизация

**Овраг
бесконечной
крутизны**



Нелинейное программирование.

Условная минимизация

Метод последовательной безусловной минимизации

$$\begin{cases} \min F(\vec{x}), \\ \vec{g}(\vec{x}) \geq 0, \end{cases} \longrightarrow \begin{cases} \min \hat{F}_k(\vec{x}), & k = 1, 2, \dots; \\ \vec{x} \in R^n; \\ \hat{F}_k(\vec{x}) = \hat{F}_k(F(\vec{x}), \vec{g}(\vec{x})). \end{cases}$$

$$\lim_{k \rightarrow \infty} \min_{\vec{x} \in R^n} \hat{F}_k(\vec{x}) = \min_{\vec{x} \in D} F(\vec{x})$$

$$|\min \hat{F}_{k+1}(\vec{x}) - \min \hat{F}_k(\vec{x})| < \varepsilon ,$$

$$\hat{F}_k(\vec{x}) = F(\vec{x}) + r_k \sum_{j=1}^m v_j (g_j(\vec{x}))$$

Нелинейное программирование.

Условная минимизация.

Барьерные функции

$$\hat{F}_k(\vec{x}) = F(\vec{x}) + r_k \sum_{j=1}^m v_j(g_j(\vec{x}))$$

$$r_k \sum_{j=1}^m v_j(g_j(\vec{x})); \quad \lim_{c_j \rightarrow +0} v_j(g_j(\vec{x})) = +\infty$$

$$v_j(\vec{x}) = \frac{1}{g_j(\vec{x})}$$

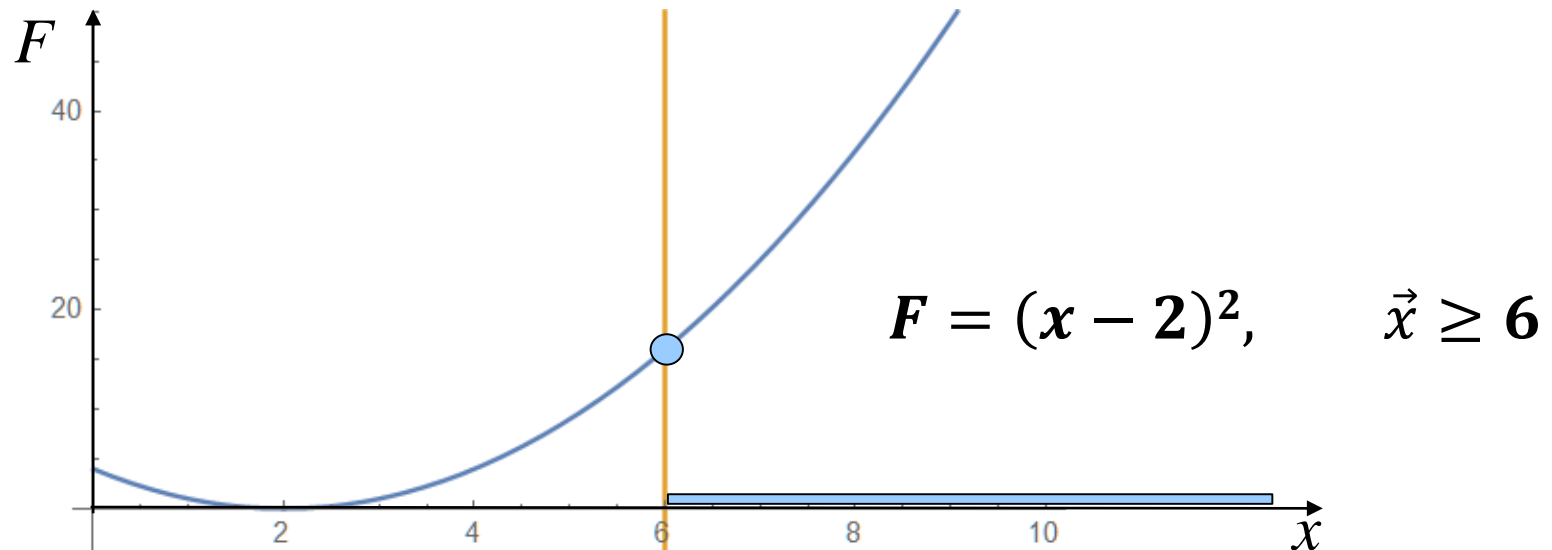
$$r_k = 1/10^{k-1}$$

Нелинейное программирование.

Условная минимизация.

Барьерные функции

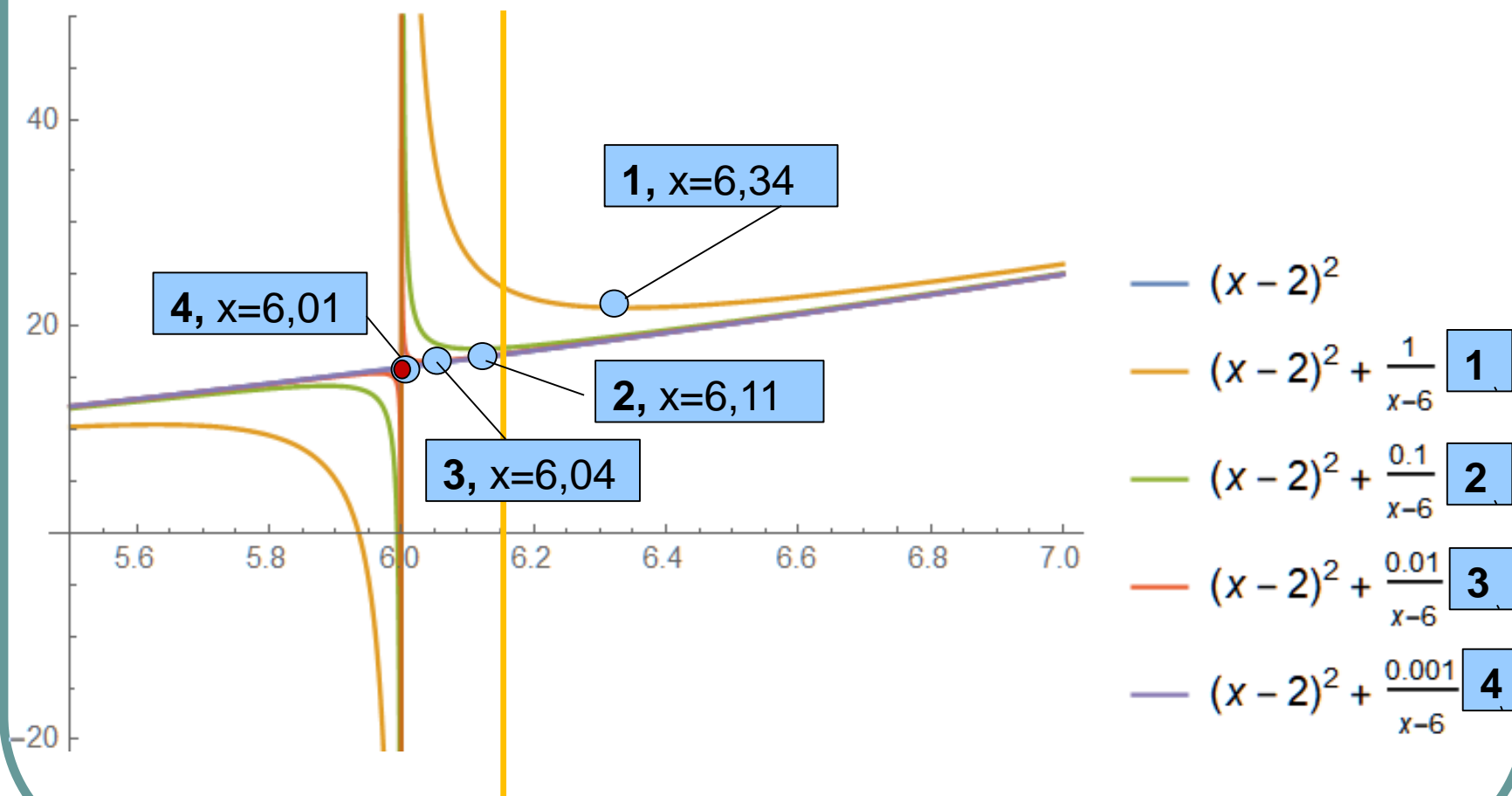
$$\begin{cases} \min F(\vec{x}), & \vec{x} = (x_1, x_2, \dots, x_n)^T \\ \vec{g}(\vec{x}) \geq 0, & \vec{g}(\vec{x}) = (g_1(\vec{x}), g_2(\vec{x}), \dots, g_m(\vec{x}))^T \end{cases}$$



Нелинейное программирование.

Условная минимизация.

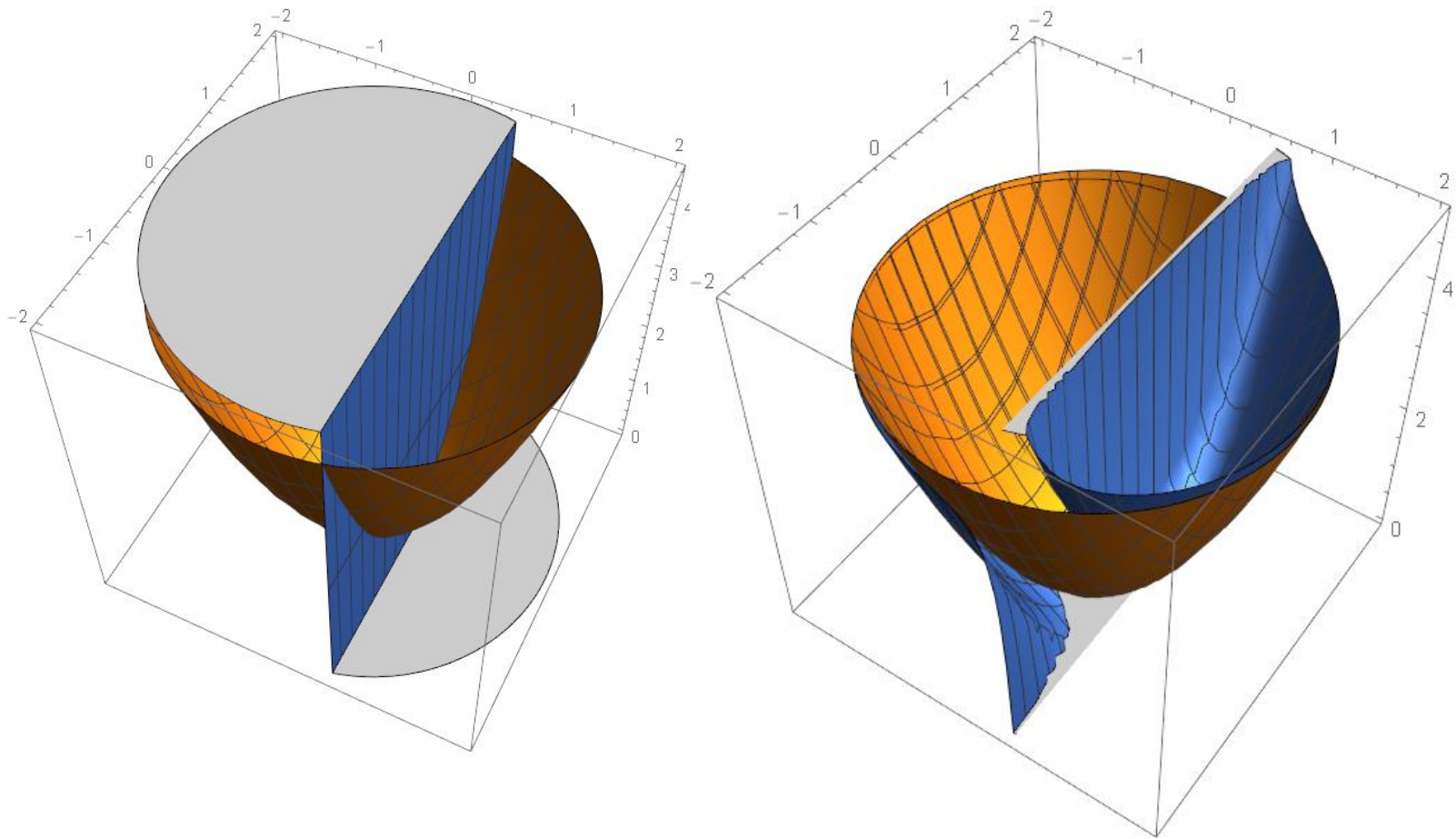
Барьерные функции



Нелинейное программирование.

Условная минимизация.

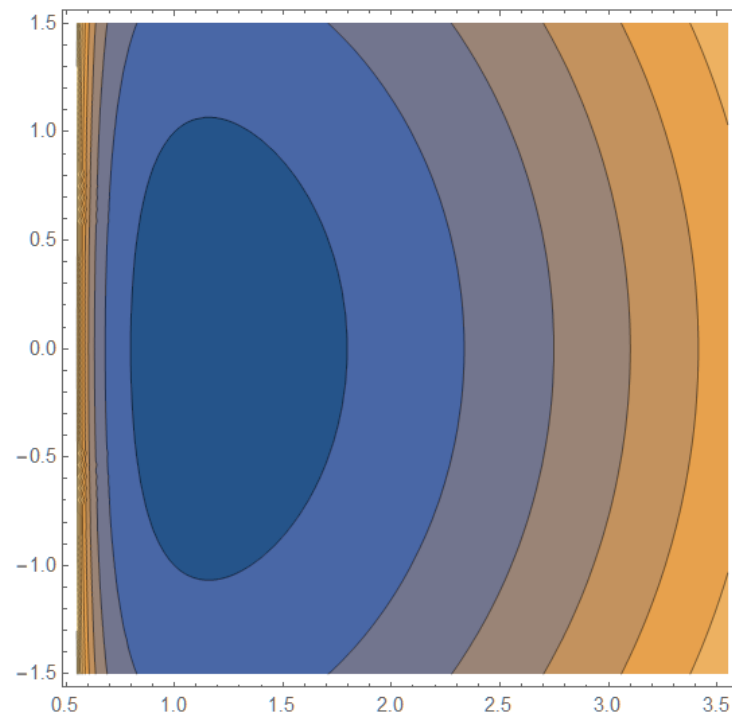
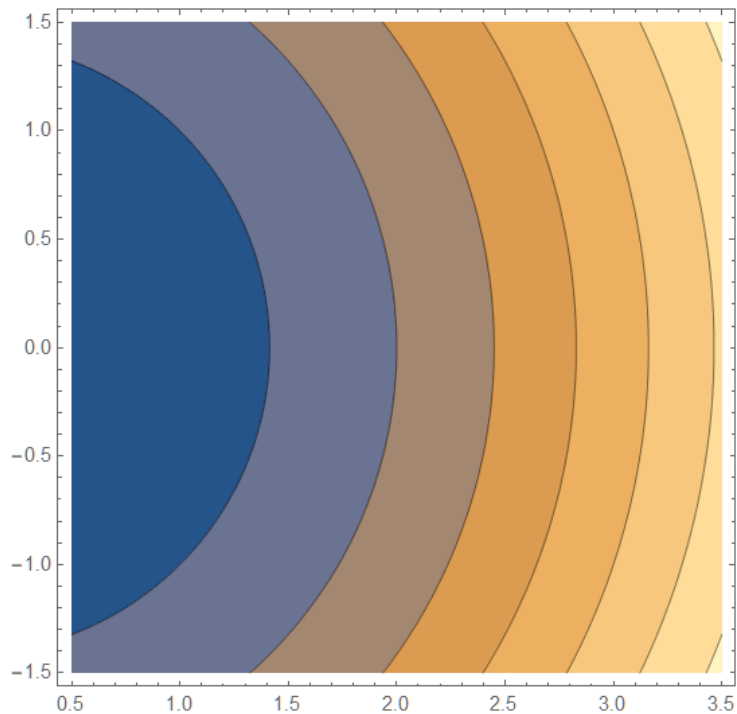
Барьерные функции



Нелинейное программирование.

Условная минимизация.

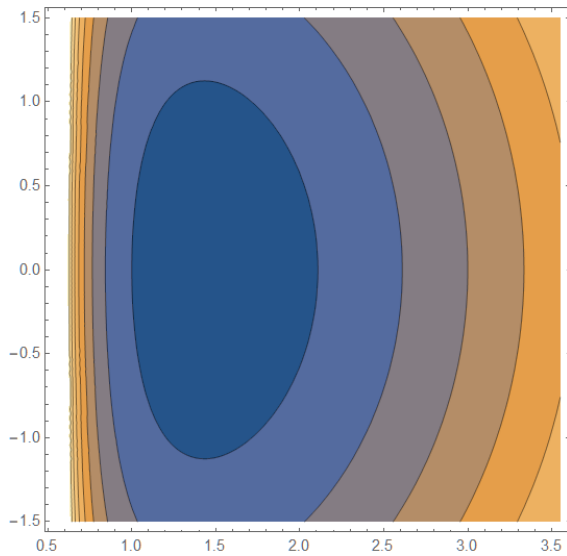
Барьерные функции



6. Нелинейное программирование.

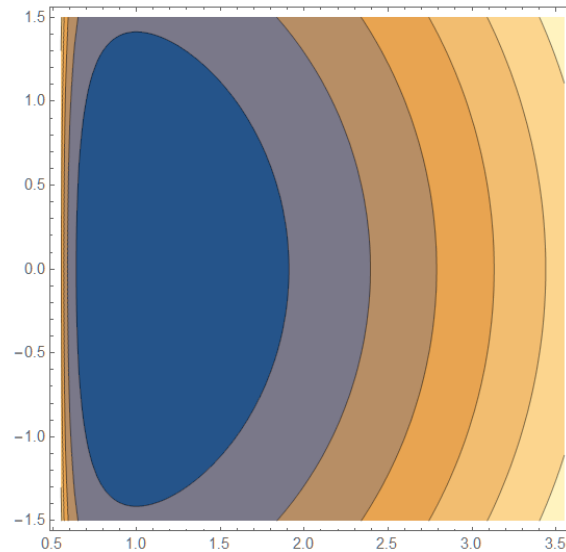
6.15. Условная минимизация функций n переменных.

Барьерные функции



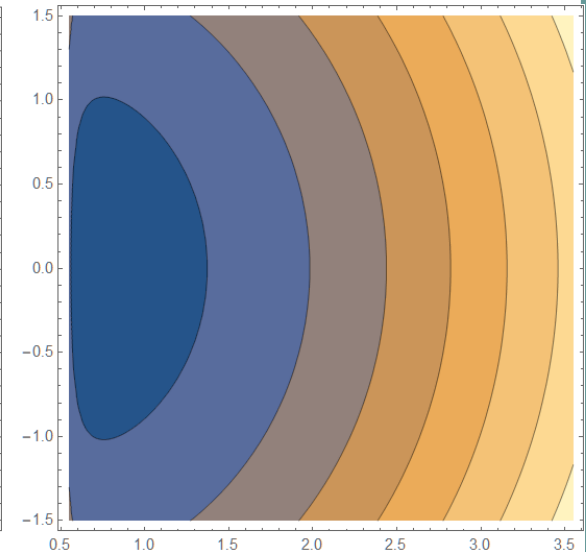
r_1

>



r_2

>



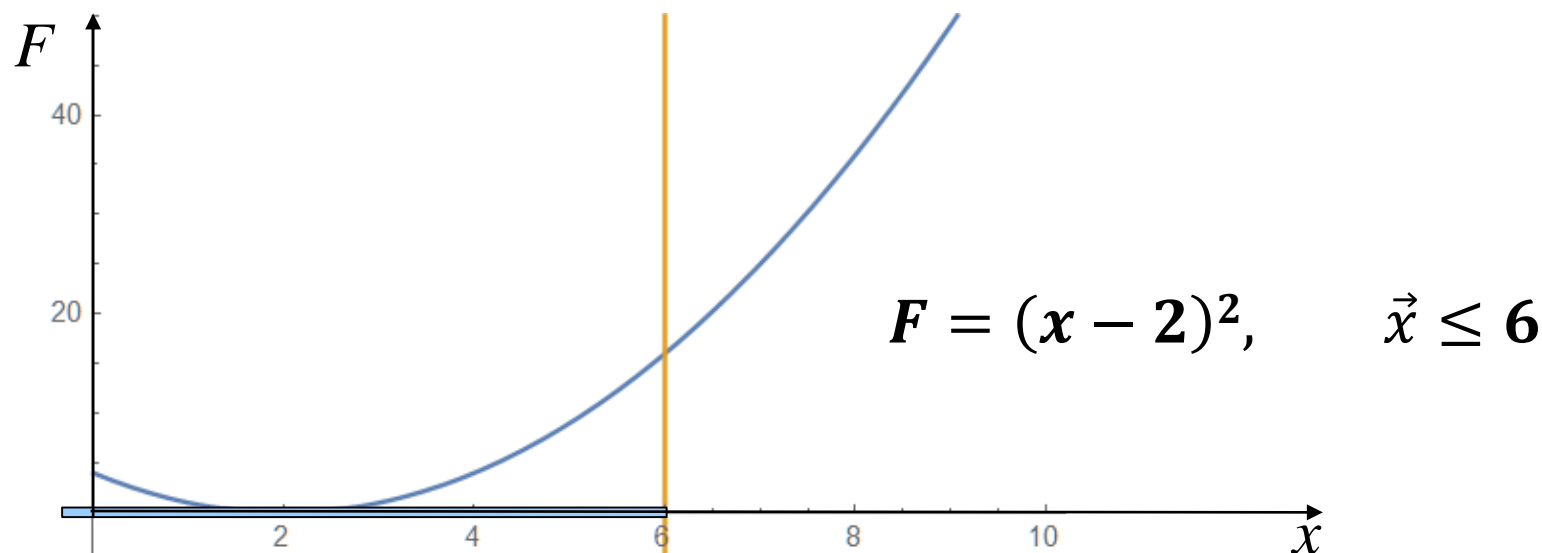
r_3

Нелинейное программирование.

Условная минимизация.

Барьерные функции

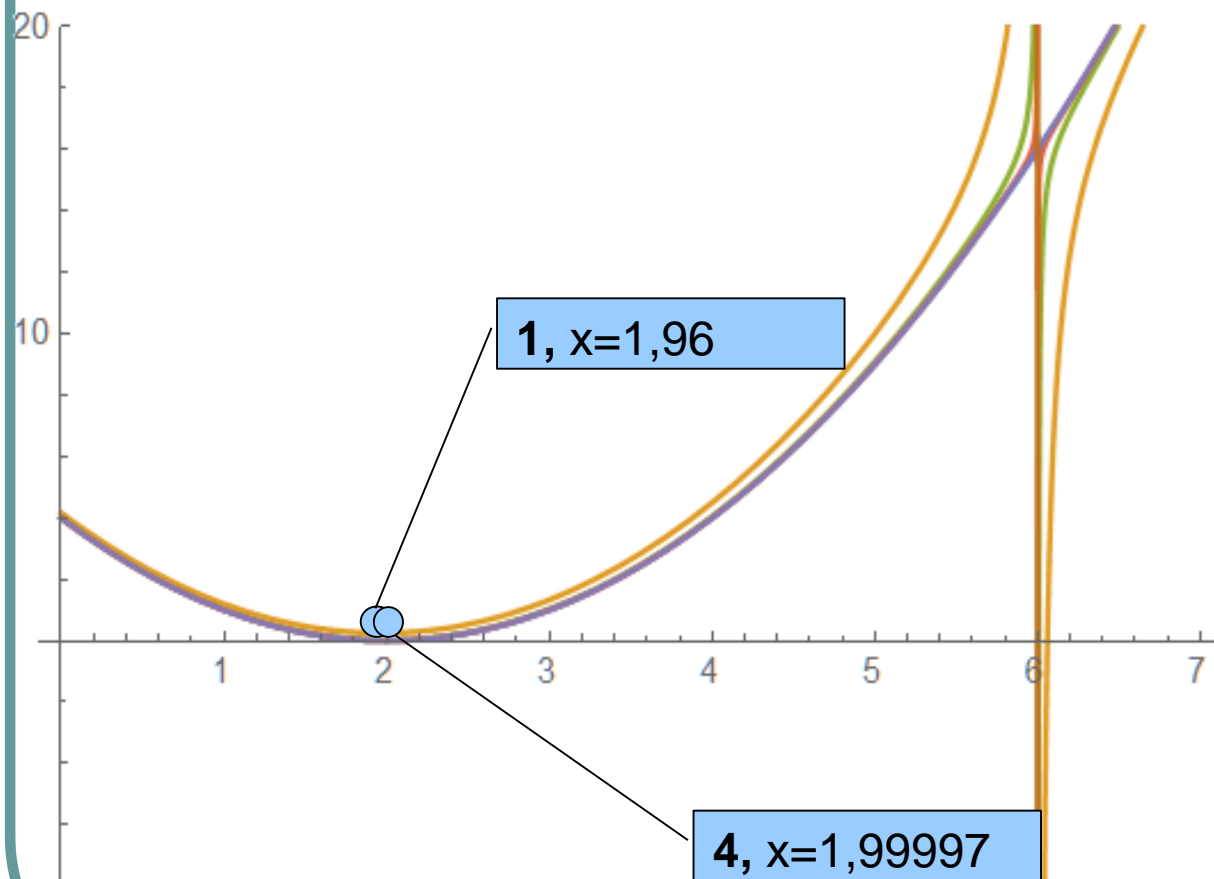
$$\begin{cases} \min F(\vec{x}), & \vec{x} = (x_1, x_2, \dots, x_n)^T \\ \vec{g}(\vec{x}) \geq 0, & \vec{g}(\vec{x}) = (g_1(\vec{x}), g_2(\vec{x}), \dots, g_m(\vec{x}))^T \end{cases}$$



Нелинейное программирование.

Условная минимизация.

Барьерные функции



- $(x - 2)^2$
- $(x - 2)^2 + \frac{1}{6-x}$ **1**
- $(x - 2)^2 + \frac{0.1}{6-x}$ **2**
- $(x - 2)^2 + \frac{0.01}{6-x}$ **3**
- $(x - 2)^2 + \frac{0.001}{6-x}$ **4**

Нелинейное программирование.

Условная минимизация.

Штрафные функции

$$\hat{F}_k(\vec{x}) = F(\vec{x}) + r_k \sum_{j=1}^m v_j(g_j(\vec{x}))$$

$$v_j(g_j(\vec{x})) = 0, g_j(\vec{x}) \geq 0;$$

$$v_j(g_j(\vec{x})) > 0, g_j(\vec{x}) < 0;$$

$$v_j(g_j(\vec{x})) = [\max\{-g_j(\vec{x}), 0\}]^2$$

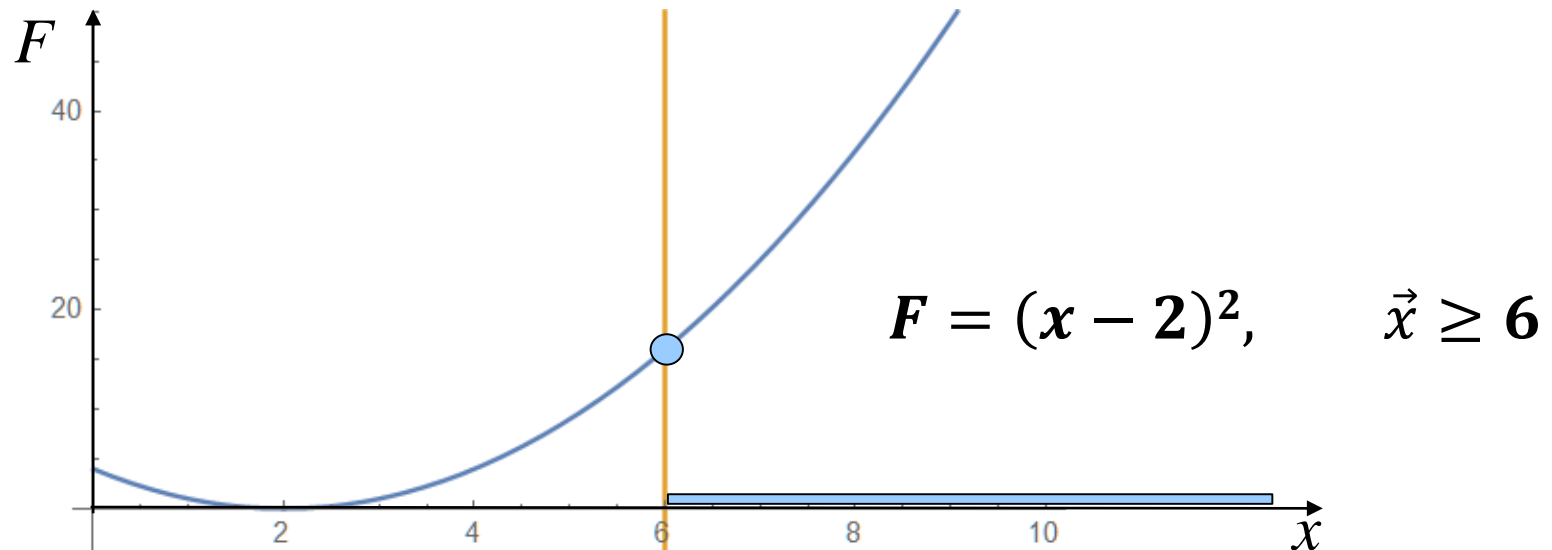
r_k — МОНОТОННО
возрастающая
числовая
последовательность

Нелинейное программирование.

Условная минимизация.

Штрафные функции

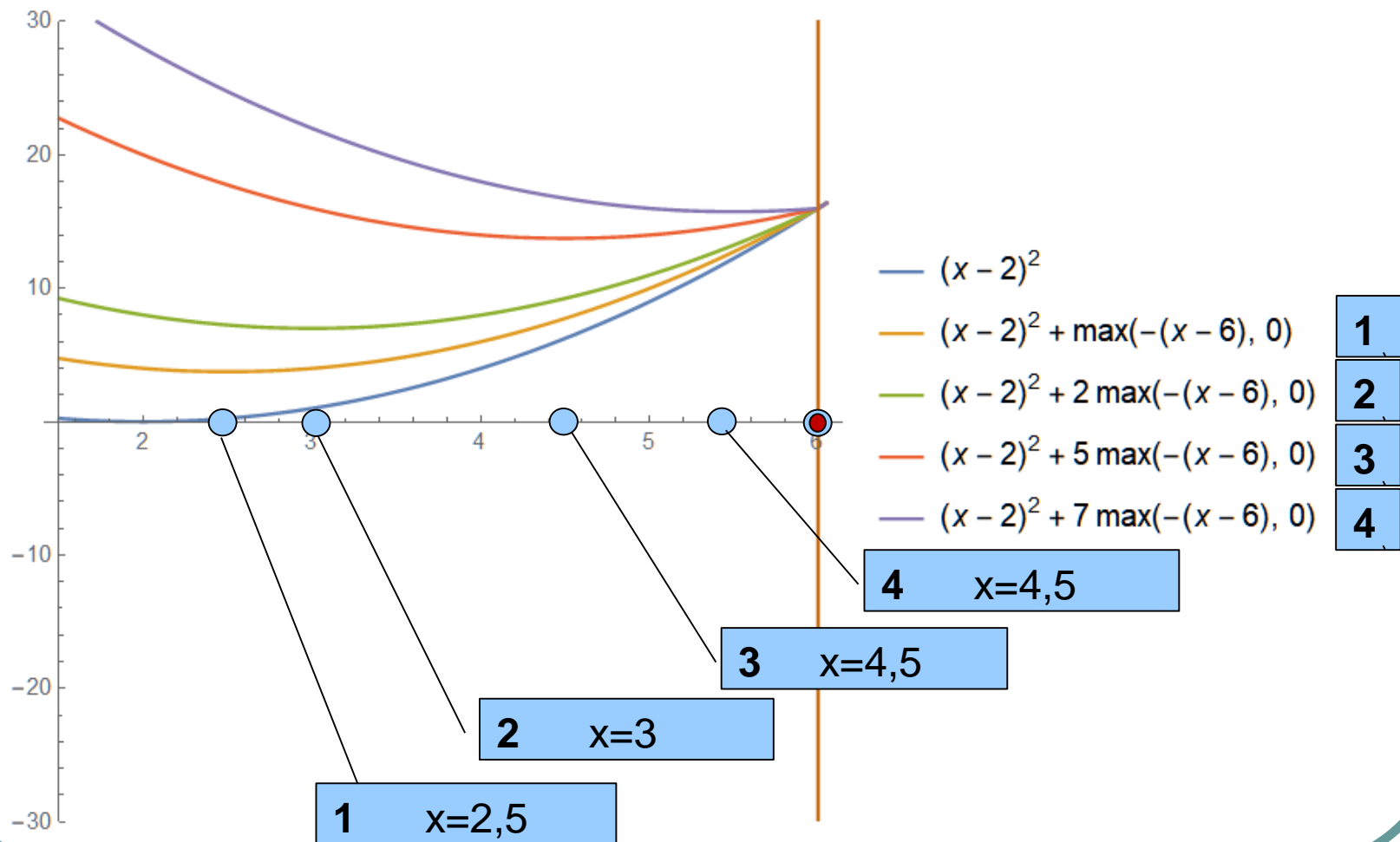
$$\begin{cases} \min F(\vec{x}), & \vec{x} = (x_1, x_2, \dots, x_n)^T \\ \vec{g}(\vec{x}) \geq 0, & \vec{g}(\vec{x}) = (g_1(\vec{x}), g_2(\vec{x}), \dots, g_m(\vec{x}))^T \end{cases}$$



Нелинейное программирование.

Условная минимизация.

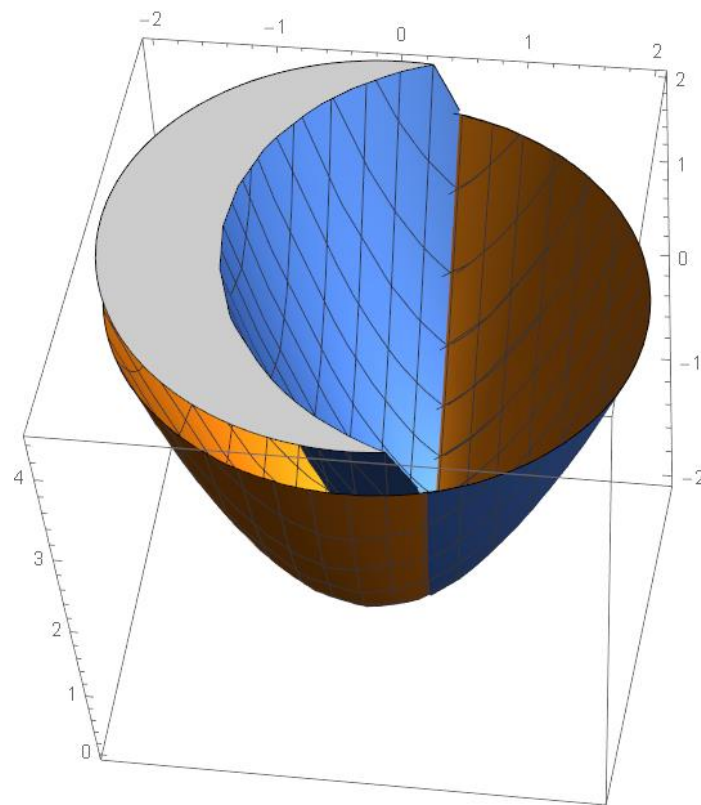
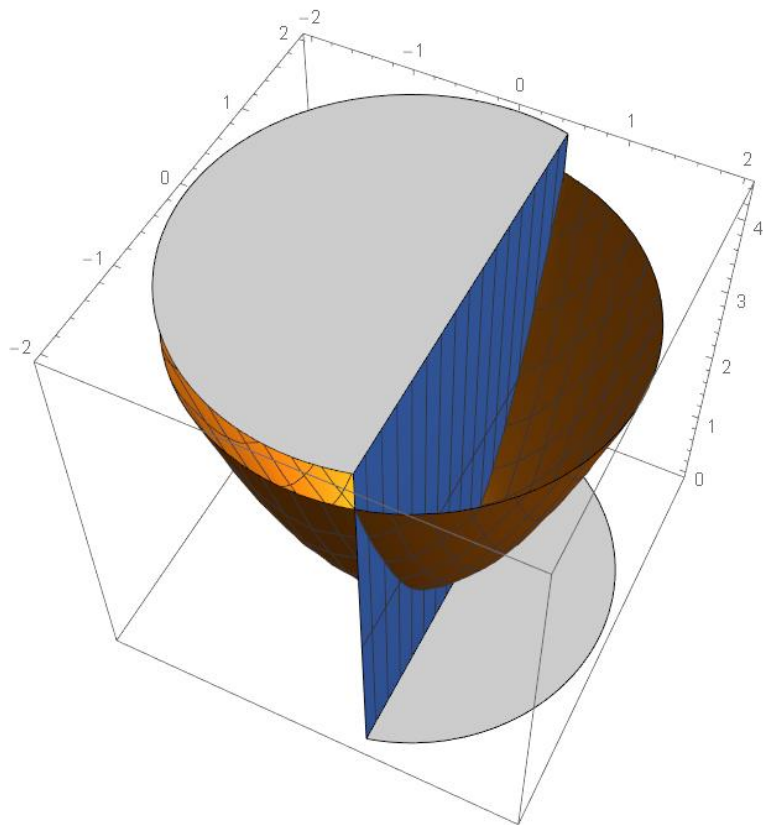
Штрафные функции



Нелинейное программирование.

Условная минимизация.

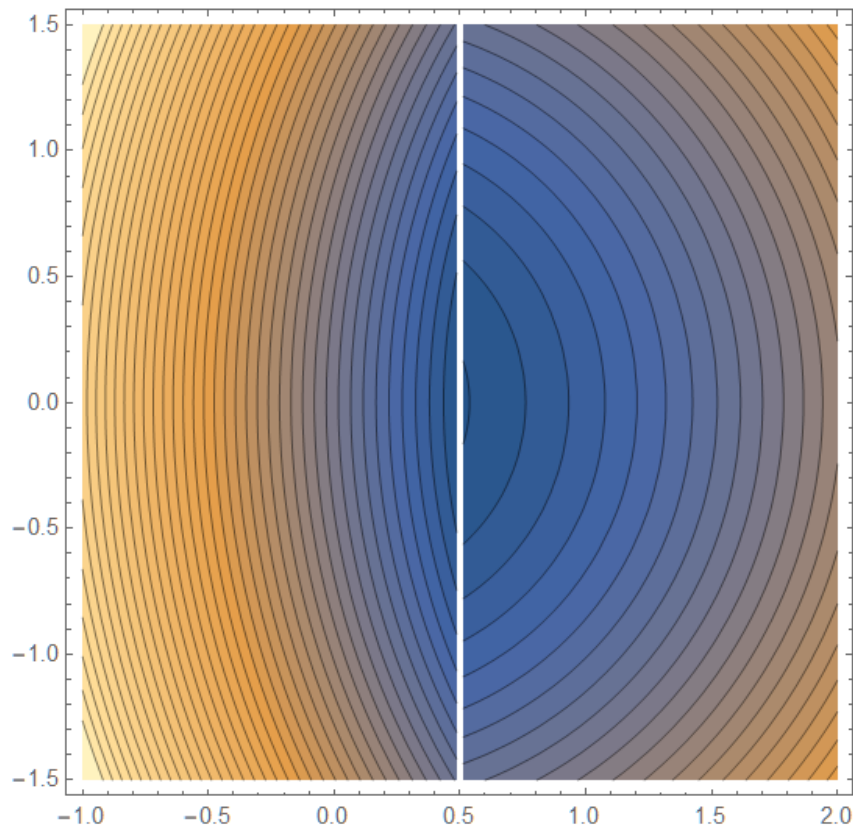
Штрафные функции



Нелинейное программирование.

Условная минимизация.

Штрафные функции



Нелинейное программирование.

Условная минимизация функций n переменных.

Алгоритм метода последовательной безусловной минимизации

1. Задать начальную точку \vec{x}^0 , значение ε , предельное число итераций M ;
2. Положить $k=1$
3. Выполнить безусловный поиск минимума целевой функции $\hat{F}_k(\vec{x}) = F(\vec{x}) + r_k \sum_{j=1}^m v_j (c_j(\vec{x}))$ из начальной точки \vec{x}^0 с результатом в точке \vec{x}^b и значением целевой функции F^b
4. Для k от 2 до M с шагом +1 выполнять цикл
 - 4.1. Выполнить безусловный поиск минимума целевой функции $\hat{F}_k(\vec{x}) = F(\vec{x}) + r_k \sum_{j=1}^m v_j (c_j(\vec{x}))$ из начальной точки \vec{x}^b с результатом в точке \vec{x}^c и значением целевой функции F^c
 - 4.2. Если $|F^b - F^c| < \varepsilon$, выйти из цикла
 - 4.3. Положить $\vec{x}^b = \vec{x}^c, F^b = F^c$;
5. Если $k = M$, то вывести предупреждение, что минимум не достигнут
6. Положить результат решения задачи условной минимизации $\vec{x}^* := \vec{x}^c$;
7. Завершить работу