1 Методы градиентного спуска

1.1 Градиентный спуск с дроблением шага.

a. Новая точка $x^{k+1} = x^k + \varkappa_{k+1}(-grad\ f(x^k))$

b. Направление. Антиградиент $\omega^{k+1} = -grad\ f(x^k)$.

c. Шаги алгоритма.

1. На k-ой итерации находимся в точке x^k.
   Вычисляем $\omega^{k+1} = -grad\,f(x^k)$. Если выполненно условие $|\omega^{k+1}| < \epsilon$, то x^k берем за минимум, иначе переходим к пункту 2 и устанавливаем $\varkappa_k := \varkappa_0$(первоначальная длина шага спуска для k-й итерации).
2. Определяем следующую точку $x^{k+1} = x^k + \varkappa_{k+1}\omega^{k+1}$.
   Если $f(x^{k+1}) \leq f(x^k)$, то принимаем эту точку. $k := k+1$ переходим к пункту 1.
   Если $f(x^{k+1}) \geq f(x^k)$, то нужно искать другую точку в этом направлении, уменьшая длину шага спуска. Переходим к пункту 3.
3. $\varkappa_k := \nu\varkappa_k$, где ν - коэффициент дробления. Переходим к пункту 2.

1.2 Метод наискорейшего спуска спуска.

a. Новая точка $x^{k+1} = x^k + \varkappa_{k+1}(-grad\ f(x^k))$

b. Направление. Антиградиент $\omega^{k+1} = -grad\ f(x^k)$.

c. Длина шага соответсвует минимальному значению целевой функции $f(x)$ в направлении ω^{k+1} и отсчитывается от точки x^k. Для этого находим минимум вспомагательной функции $\varphi_k(\alpha) = f(x^k + \alpha\omega^{k+1})$ по формуле $\alpha_{min} = \frac{(\omega^{k+1},\omega^{k+1})}{(Q\omega^{k+1},\omega^{k+1})}$. Где $f(x) = \frac{1}{2}(Qx,x) + (b,x) + c$ - минимизируемая функция. Тогда минимум α_{min} функции $\varphi_k(\alpha)$ берется за длину шага.

d. Шаги алгоритма.

1. На k-ой итерации находимся в точке x^k.
   Вычисляем $\omega^{k+1} = -grad\ f(x^k)$. Если выполненно условие $|\omega^{k+1}| < \epsilon$, то x^k берем за минимум, иначе переходим к пункту 2.
2. Определяем следующую точку $x^{k+1} = x^k + \varkappa_{k+1}\omega^{k+1}$. где $\varkappa_{k+1} = \frac{(\omega^{k+1},\omega^{k+1})}{(Q\omega^{k+1},\omega^{k+1})}$

1.3 Метод сопряженных направлений.

a. Новая точка $x^{k+1} = x^k + \varkappa_{k+1}p^{k+1}$.

b. Направление. Вектора, определяющие направления спуска на каждой итераци, являются сопряженными относительно матрицы Q, т.е. $(Qp^i, p^j) = (Qp^j, p^i) = 0,\ i,j = 1...$ Где $f(x) = \frac{1}{2}(Qx,x)+(b,x)+c$ - минимизируемая функция. На первой итерации направление спуска совпадает с направлением антиградиента $p^1 = grad\ f(x^0)$.

c. Длина шага соответсвует минимальному значению целевой функции $f(x)$ в направлении p^{k+1} и отсчитывается от точки x^k. Для этого находим минимум $\varphi_k(\alpha) = f(x^k + \alpha p^{k+1})$ по формуле $\alpha_{min} = \frac{(\omega^{k+1},p^{k+1})}{(Qp^{k+1},p^{k+1})}$. Где $f(x) = \frac{1}{2}(Qx,x) + (b,x) + c$ - минимизируемая функция. Тогда минимум α_{min} функции $\varphi_k(\alpha)$ берется за длину шага.

d. Алгоритм сходится не более чем за m итераций(колличество переменных).

e. Шаги алгоритма.

1. На k-ой итерации находимся в точке x^k.
   Вычисляем $\omega^{k+1} = -grad\ f(x^k)$. Если выполненно условие $|\omega^{k+1}| < \epsilon$, то x^k берем за минимум, иначе переходим к пункту 2.
2. Для новой точки нужно определить направление спуска.
   $p^{k+1} = \gamma_{k+1} p^k + \omega^{k+1}$, где $\gamma_{k+1} = \frac{|\omega^{k+1}|^2}{|\omega^k|^2}$. Переходим к пункту 3.
3. Длина шага $\varkappa_{k+1} = \frac{(\omega^{k+1}, p^{k+1})}{(Qp^{k+1}, p^{k+1})}$. Перехдим к пункту 4.
4. $x^{k+1} = x^k + \varkappa_{k+1} p^{k+1}$. k:=k+1. Переходим к 1.

2 Квазиньютоновские методы

2.1 ДФП(Давидона-Флетчера-Пауэлла)

a. Новая точка $x^{k+1} = x^k + \varkappa_{k+1} p^{k+1}$.

b. Направление.
В методе Ньютона $p^{k+1} = -H^{-1}(x^k) grad\ f(x^k)$. Где $H(x^k)$ - значение матрицы Гессе в точке x^k.
В ДФП $p^{k+1} = -A_{k+1} grad\ f(x^k) = A_{k+1}\omega^{k+1}$. Матрицы $\{A_k\} \to H^{-1}(x^*)$ при $k \to \infty$, x^*- точный минимум целевой функции $f(x)$.

c. Длина шага определяется исчерпывающим спуском в направлении p^{k+1}.(см. предыдущие методы)

d. На первом шаге A_1-единичная матрица, поэтому первая итерация совпадает с 1-й итерацией наискорейшего спуска.

e. Шаги алгоритма.

1. На k-ой итерации находимся в точке x^k.
   Вычисляем $\omega^{k+1} = -grad\ f(x^k)$. Если выполненно условие $|\omega^{k+1}| < \epsilon$, то x^k берем за минимум, иначе переходим к пункту 2.
2. Определяем направление на новую точку $p^{k+1} = A_{k+1}\omega^{k+1}$.
3. Матрица $A_{k+1} = A_k + \Delta A_k$. ΔA_k - поправочная матрица.
   $\Delta A_k = -\frac{\Delta x^k, (\Delta x^k)^T}{(\Delta\omega^k, \Delta x^k)} - \frac{A_k \Delta\omega^k (\omega_k)^T (A_k)^T}{(A^k \Delta\omega^k, \Delta\omega^k)}$.
   $\Delta x^k = x^k - x^{k-1}$
   $\Delta\omega^k = \omega^{k+1} - \omega^k$
4. Длина шага спуска
   $\varkappa_{k+1} = \frac{(\omega^{k+1}, p^{k+1})}{(Qp^{k+1}, p^{k+1})}$
5. Следующая точка $x^{k+1} = x^k + \varkappa_{k+1} p_{k+1}$.

3 Методы прямого поиска

3.1 Циклический покоординатный спуск

a. Каждая итерация состит из последовательной минимизации функции по каждому из направлений в установленном базисе.

b. Базис естественный в R^n.

c. Длина шага в каждом направлении определяется исчерпывающим спуском. При минимизации целевой функции по i-му направлению базиса $\{e_i\}_{i=1}^n$ используется информация после минимизации по первым $(i-1)$ направлениям. Т.е. в выражении для вспомогательной функции $\varphi_i^k(\alpha) = (\tilde{x} + \alpha e_i)$ $\tilde{x}$ - это результат минимизации по направлению e_{i-1}. Имеются сходства с методом Зейделя.

d. Шаги алгоритма.

   1. На k-ой итерации находимся в точке x^k.
   Проверяем условие $|x^{k-1} - x^k| < \epsilon_1$, или $|f(x^k) - f(x^{k-1})| < \epsilon_2$ или сразу оба. В случае их выполнения считаем точку x^k за минимум функции, иначе переходим к пункту 2.
   2. Выполняем одномерную минимизацию по каждому направлению базиса $\{e_i\}_{i=1}^n$.
   $i := 1$ - счетчик направлений.
   $\tilde{x} = x^k$ - задаем начальное значение временной переменной, хранящей в себе результат одномерной минимизации на прошлой итерации. Переходим к пункту 3.
   3. Минимизация по i-му направлению. Находим минимум вспомогательной функции $\varphi_i^k(\alpha) = (\tilde{x} + \alpha e_i)$ $\tilde{x}$ по формуле $\alpha_{min} = \frac{(\tilde{\omega}, e_i)}{(Qe_i, e_i)}$, где $\tilde{\omega}$- антиградиент из точки $\tilde{x}$.
   $\tilde{x} = \tilde{x} + \alpha_{min} e_i$. Переходим к 4.
   4. Если $i = n$(перебрали все направления), то принимаем $x^{k+1} = \tilde{x}$, $k = k+1$ и переходим к 1.
   Иначе $i = i+1$ и переходим к 3.

3.2 Метод Хука-Дживса

a. Каждая итерация состит из двух этапов. Первы - это обычный циклический покоординатный спуск. Второй - это минимизация оп направлению соответсвующему вектору перемещения $p = \tilde{x} - x^k$. Где x^k - начальная точка для k - ой итерации, $\tilde{x}$ - результат циклическокого покоординаного спуска на данной итерации.

b. Базис естественный в R^n.

c. Длина шага в каждом направлении определяется исчерпывающим спуском. При минимизации целевой функции по i-му направлению базиса $\{e_i\}_{i=1}^n$ используется информация после минимизации по первым $(i-1)$ направлениям. Т.е. в выражении для вспомогательной функции $\varphi_i^k(\alpha) = (\tilde{x} + \alpha e_i)$ $\tilde{x}$ - это результат минимизации по направлению e_{i-1}. Имеются сходства с методом Зейделя.

d. Шаги алгоритма.

   1. На k-ой итерации находимся в точке x^k.
   Проверяем условие $|x^{k-1} - x^k| < \epsilon_1$, или $|f(x^k) - f(x^{k-1})| < \epsilon_2$ или сразу оба. В случае их выполнения считаем точку x^k за минимум функции, иначе переходим к пункту 2.
   2. Выполняем одномерную минимизацию по каждому направлению базиса $\{e_i\}_{i=1}^n$.
   $i := 1$ - счетчик направлений.
   $\tilde{x} = x^k$ - задаем начальное значение временной переменной, хранящей в себе результат одномерной минимизации на прошлой итерации. Переходим к пункту 3.
   3. Минимизация по i-му направлению. Находим минимум вспомогательной функции $\varphi_i^k(\alpha) = (\tilde{x} + \alpha e_i)$ $\tilde{x}$ по формуле $\alpha_{min} = \frac{(\tilde{\omega}, e_i)}{(Qe_i, e_i)}$, где $\tilde{\omega}$- антиградиент из точки $\tilde{x}$.
   $\tilde{x} = \tilde{x} + \alpha_{min} e_i$. Переходим к 4.
   4. Если $i = n$(перебрали все направления), то переходим к 5. Иначе $i = i+1$ и переходим к 3.

5. Вычисляем $p = \tilde{x} - x^k$ и проводим исчерпывающий спуск в этом направении. Находим минимум функции $\varphi_i^k(\alpha) = (\tilde{x} + \alpha p)$ по формуле $\alpha_{min} = \frac{(\tilde{\omega},p)}{(Qp,p)}$, где $\tilde{\omega}$-антиградиент из точки $\tilde{x}$.
Принимаем $x^{k+1} = \tilde{x} + \alpha_{min}p$, $k = k+1$ и переходим к пункту 1.

3.3 Метод Розенброка

a. Каждая итерация состоит из последовательной минимизаии по каждому из направлений базиса

b. После каждой итераци проводять процесс ортогонализации Грамма - Шмидта, результатом которой является новый базис для следующей итерации. На первой итерации базис естественный.

c. Длина шага в каждом направлении определяется исчерпывающим спуском. При минимизации целевой функции по i-му направлению базиса $\{p_i^k\}_{i=1}^n$ используется информация после минимизации по первым $(i-1)$ направлениям. Т.е. в выражении для вспомогательной функции $\varphi_i^k(\alpha) = (\tilde{x} + \alpha p_i^k)$, где $\tilde{x}$ - это результат минимизации по направлению p_{i-1}^k. Имеются сходства с методом Зейделя.

d. Шаги алгоритма.

1. На k-ой итерации находимся в точке x^k и $\{p_i^k\}_{i=1}^n$ -система векторов, определяющих базис на этой итерации.
Проверяем условие $|x^{k-1} - x^k| < \epsilon_1$, или $|f(x^k) - f(x^{k-1})| < \epsilon_2$ или сразу оба. В случае их выполнения считаем точку x^k за минимум функции, иначе переходим к пункту 2.
2. Выполняем одномерную минимизацию по каждому направлению базиса $\{p_i^k\}_{i=1}^n$.
$i := 1$ - счетчик направлений.
$\tilde{x} = x^k$ - задаем начальное значение временной переменной, хранящей в себе результат одномерной минимизации на прошлой итерации. Переходим к пункту 3.
3. Минимизация по i-му направалению. Находим минимум вспомогательной функции $\varphi_i^k(\alpha) = (\tilde{x} + \alpha p_i^k)$ по формуле $\alpha_{min}^i = \frac{(\tilde{\omega},p_i^k)}{(Qp_i^k,p_i^k)}$, где $\tilde{\omega}$- антиградиент из точки $\tilde{x}$.
$\tilde{x} = \tilde{x} + \alpha_{min}^i p_i^k$. Все α_{min}^i - нужно запомнить, для ортоганализации. Переходим к 4.
4. Если $i = n$(перебрали все направления), то переходим к 5. Иначе $i = i+1$ и переходим к 3.
5. Проводим ортоганализацию Грамма-Шмидта. (см учебник стр 294-295).
Для случаю $n = 2$.

$$a_1^{k+1} = \begin{cases} p_1^k, & \alpha_{min}^1 = 0, \\ \alpha_{min}^1 p_1^k + \alpha_{min}^2 p_2^k, & \alpha_{min}^1 \neq 0. \end{cases} \tag{1}$$

$$a_2^{k+1} = \begin{cases} p_2^k, & \alpha_{min}^2 = 0, \\ \alpha_{min}^2 p_2^k, & \alpha_{min}^2 \neq 0. \end{cases} \tag{2}$$

$b_1^{k+1} = \alpha_1^{k+1} \implies p_1^{k+1} = \frac{b_1^{k+1}}{|b_1^{k+1}|}$
$b_2^{k+1} = \alpha_2^{k+1} - (\alpha_2, p_1^{k+1})p_2^{k+1} \implies p_2^{k+1} = \frac{b_2^{k+1}}{|b_2^{k+1}|}$
p_1^k - сонаправлен с перемещением из x^k в $\tilde{x}$

3.4 Метод Пауэлла

a. Каждая итерация состоит из многократной минимизации по направлениям базиса. После минимизации по всем направлениям, базис меняется некоторым образом, и снова проводим минимизацию по всем направлениям. Так происходит n раз.

b. Пусть$\{p_i^k\}_{i=1}^n$ - базис на начало к-ой итерации. Пусть мы провели одномерную минимизацию по всем направлениям базиса и получили точку $\tilde{x}_1$. Теперь проводим замены: $p_i^k = p_{i+1}^k$ для $i = 1..n-1$, а $p_n^k = \tilde{x}_1 - x^k$ - перемещение из x^k в $\tilde{x}_1$. Теперь мы еще раз проводим одномерную минимизацию по всем направлениям нового базиса и начальная точка теперь $\tilde{x}_1$. Эти действия повторяются n раз. Все это считается за одну итарацию.

c. Шаги алгоритма.

1. На k-ой итерации находимся в точке x^k и $\{p_i^k\}_{i=1}^n$ -система векторов, определяющих базис на k-й итерации.
   Проверяем условие $|x^{k-1} - x^k| < \epsilon_1$, или $|f(x^k) - f(x^{k-1})| < \epsilon_2$ или сразу оба. В случае их выполнения считаем точку x^k за минимум функции, иначе устанавливаем $j := 1$ и переходим к пункту 2.
2. Одномерная минимизаци по каждому направлению базиса $\{p_i^k\}_{i=1}^n$.
   $i := 1$.
   $\tilde{x} := x^k$.
3. Минимизация по i-му направлению.
   Находими минимум вспомогательной функции $\varphi_i^k(\alpha) = f(\tilde{x} + \alpha p_i^k)$ по формуле $\alpha_{min} = \frac{(\tilde{\omega}, p_i^k)}{(Qp_i^k, p_i^k)}$, где $\tilde{\omega}$- антиградиент из точки $\tilde{x}$.
   $\tilde{x} = \tilde{x} + \alpha_{min} p_i^k$.
4. Если $i < n$, то $i = i + 1$ и переходим к пункту 3.
   Иначе переходим к 5.
5. Вычисляем $p = \tilde{x} - x^k$. Проводим минимизацию в этом направлении. Находим минимум функции $\varphi(\alpha) = f(\tilde{x} + \alpha p)$ по формуле $\alpha_{min} = \frac{(\tilde{\omega}, p)}{(Qp, p)}$, где $\tilde{\omega}$- антиградиент из точки $\tilde{x}$.
   $z = \tilde{x} + \alpha_{min} p$.
   Если $j = n$, то $x^{k+1} = z$, $k = k + 1$ и переходим к 1.
   Иначе меняем базис: $p_r = p_{r+1}$, $r = 1..(n-1)$, $p_n = p$. Устанавливаем $i = 1$, $j = j + 1$ и переходим к пункту 3.

4 Симплексные методы.

4.1 Регулярный симплекс.

a. Каждая итерация - это отражение симплекса относительно одной из его сторон либо сжатие симплекса к одной из его из его вершин. На каждой итерации необходимо поддерживать строгую нумерацию вершин. Пусть в симплексе S^k имеются вершины $x^{x,1}$, $x^{k,2}, \ldots, x^{k,n+1}$. Тогда $f(x^{k,1}) \leq f(x^{k,2}) \leq \ldots f(x^{k,n+1})$. На плоскости(n=2) симплексом S^k является треугольник с тремя вершинами: $x^{k,1}$, $x^{k,2}$, $x^{k,3}$.

b. Отражение происходит относитело гиперплоскости, содержащей вершины $x^{k,1} \ldots x^{k,n}$ (вершины с наименьшим значением функции). При этом вместо точки $x^{k,n+1}$ получаем точку $x^{k+1,n+1}$. И новый симплекс S^{k+1} будет состоять из n точек $x^{k,1} \ldots x^{k,n}$ симплекса S^k и новой точки $x^{k+1,n+1}$.
   Если $f(x^{k+1,n+1}) \leq f(x^{k,n})$, то отражение считаем неудачным полученный симплекс

S^{k+1} не рассматривают. Строят новый, сжимая симплекс S^k к вершине $x^{k,1}$, значение функции в которой наименшее среди других вершин.

c. Шаги алгоритма.

1. На k-ой итерации имеем симплекс S^k с вершинами $x^{k,1} \dots x^{k,n+1}$.
   Проверяем условие $(\frac{1}{n+1}\sum_{i=1}^{n}(f(x^{k,i}) - f(x^k))^2)^{\frac{1}{2}} \le \epsilon$, где x^k - центр симплекса. Если оно выполеннно, то за минимум можно брать точку x_k, иначе переходим к пункту 2.
2. Отражение происходит по формуле $x^{x+1,n+1} = 2x_c^k - x^{k,n+1} = \frac{2}{n}\sum_{i=1}^{n} x^{k,i} - x^{k,n+1}$, где x_c^k - центр гиперплоскости, относительно которой проводим отражение. Получим новый симплекс S^{k+1}. Переходим к пункту 3.
3. Редукция проводится при условии, что $f(x^{k+1,n+1}) \le f(x^{k,n+1})$(значение в новой вершине получилось больше, чем наибольшее значение для старого симплекса).
   При его выполнении симплекс S^{k+1} полученный в пункте 2 отбрасывают и сроят другой. Первая вершина будет совпадать с вершиной $x^{k,1}$ старого симплекса S^k, остальные получаются по формуле $x^{k+1,j} = x^{k.1} + \delta(x^{k,j} - x^{k,1})$, $j = 2 \dots n+1$. В новом симплексе нужно перенумероват вершины в соответсвии с правилом: $f(x^{k+1,1}) \le f(x^{k+1,2}) \le \dots f(x^{k+1,n+1})$. Увеличим $k = k+1$ и перейдем к пункту 1.

4.2 Нерегулярный симплекс. Алгоритм Нелдера-Мида.

a. Шаги алгоритма.

1. На k-ой итерации имеем симплекс S^k с вершинами $x^{k,1} \dots x^{k,n+1}$.
   Проверяем условие $(\frac{1}{n+1}\sum_{i=1}^{n}(f(x^{k,i}) - f(x^k))^2)^{\frac{1}{2}} \le \epsilon$, где x^k - центр симплекса. Если оно выполеннно, то за минимум можно брать точку x_k, иначе переходим к пункту 2.
2. Отражение происходит по формуле $x^{x+1,n+1} = 2x_c^k + \alpha(x_c^k - x^{k,n+1})$, где $x_c^k = \frac{1}{n}\sum_{i=1}^{n} x^{k,i}$, α - коэффициент отражения, при его помощи можно менять расстояния от первых n верши до новой.
   Рисунок в учебнике на странице 271 (6.9 a).
   Получим новый симплекс S^{k+1}. Теперь из того как соотносятся значения функции в новой вершин со значениями в старых вершинах, возможны три ситуации.
   При выполнении $f(x^{k+1,n+1}) \le f(x^{k,1})$. Новое значение получилось меньше, чем старые. Попробуем растянуть симплекс, что бы нйти еще меньшее значение. Переходим к пункту 3 с симплнксом S^{k+1}.
   При выполнении $f(x^{k,1}) \le f(x^{k+1,n+1}) \le f(x^{k,n})$ переходим к пункту 1 с симплексом S^{k+1}, $k = k+1$.
   При выполнении $f(x^{k,n}) \le f(x^{k+1,n+1})$ переходим к пункту 4(сжатие) с симплексом S^{k+1}.
3. Растяжение симплекса S^{k+1} происходит по формуле $x_*^{k+1,n+1} = x_c^k + \beta(x^{k+1,n+1} - x_c^k)$.
   Получили новый симплекс S_*^{k+1}.
   Рисунок в учебнике на странице 271 (6.9 б).
   При выполнении $f(x_*^{k+1,n+1}) < f(x^{k,1})$, считаем растяжение удачным и переходим к пункту 1 с новым симплексом $S^{k+1} = S_*^{k+1}$, $k = k+1$
   При выполнении $f(x_*^{k+1,n+1}) \ge f(x^{k,1})$ переходим к пункту 1 с симплексом S^{k+1}(полученным во 2-м пункте), $k = k+1$.

4. Сжатие симплекса S_{k+1}.
   При выполнении условия $f(x^{k+1,n+1}) \leq f(x^{k,n+1})$ новая вешина получается по формуле $x_{**}^{k+1,n+1} = x_c^k + \gamma(x^{k+1,n+1} - x_c^k)$.
   При выполнении условия $f(x^{k+1,n+1}) > f(x^{k,n+1})$ новая вешина получается по формуле $x_{**}^{k+1,n+1} = x_c^k + \gamma(x^{k,n+1} - x_c^k)$.
   Рисунки в учебнике на странице 271 (6.9 в,г).
   В итоге получаем новый симплекс S_{**}^{k+1}.
   При выполении условия $f(x_{**}^{k+1,n+1}) < f(x^{k,n+1})$ переходим к пункту 1 с новым симплексом S_{**}^{k+1}.
   При выполении условия $f(x_{**}^{k+1,n+1}) \geq f(x^{k,n+1})$ переходим к пункту 5 с первоначальным симплексом S^k.

5. Редукция симплекса S^k проходи по формуле $x^{k+1,i} = x^{k,1} + \delta(x^{k,i} - x^{k,1})$, $i = 2 \ldots (n+1)$. Длины всех ребер.
   Рисунок в учебнике на странице 271 (6.9 д).
   Получаем новый симплекс S_{***}^{k+1}.
   уменьшаются в $\frac{1}{\delta}$ раз. $S^{k+1} = S_{***}^{k+1}$, $k = k+1$ и переходим к пункту 1.

5 Методы нелинейного программирования.

5.1 Метод условного градиента.

a. Новая точка $x^{k+1} = x^k + \varkappa_{k+1} p^{k+1}$.

b. Направление спуска зависит от свойств границы, оно может не совпадать с направлением антиградиента. В двумерном случае с прямоугольной границей направление спуска будет совпадать с направлением на один из угол границы. Данный угол выбирается в соответвии с направлением антиградиента в данной точке.

c. Длина шага спуска определяется одномерной минимизацией в выбранном направлении.

d. Шаги алгоритма.

1. На k-ой итерации находимся в точке x^k.
   Проверяем условие $|x^{k-1} - x^k| < \epsilon_1$, или $|f(x^k) - f(x^{k-1})| < \epsilon_2$ или сразу оба. В случае их выполнения считаем точку x^k за условный минимум функции, иначе переходим к пункту 2.
2. Вычисляем $\omega^{k+1} = -grad\, f(x^k)$. Находим проекции ω_{x1} и ω_{x2} на оси Ox1(горизонтальная) Ox2(вертикальная). В зависимости от знаков проекции выбираем нужный угол прямоугольной границы.
   Например: $\omega_{x1} \leq 0$ и $\omega_{x2} \leq 0$, что соответвует направлению антиградиента вниз и влево, значит выбираем нижний левый угол $\tilde{x}$ нашей границы.
   Получили направление спуска $p^{k+1} = \tilde{x} - x^k$
3. Проведем минимизацию по направлению p^{k+1}.
   Находим минимум функции $\varphi(\alpha) = f(x^k + \alpha p^{k+1})$ по формуле $\alpha_{min} = \frac{(\omega^{k+1}, p^{k+1})}{(Qp^{k+1}, p^{k+1})}$.
   $x^{k+1} = x^k + \alpha_{min} p^{k+1}$. Устанавливаем k=k+1 и переходим к пункту 1.

5.2 Метод проекции антиградиента.

a. Каждая итерация состоит из спуска в некотором направлении и проецирования полученной точки на множестово, задающее ограничение.

b. Направление спуска ортогонально градиентам активных ограничений. Пусть область ограничения $\Omega = \{x \in R : g_i(x) \leq 0, i = 0 \dots m\}$. Для итерации k с точкой x^k активными ограничениями будут те $g_i(x)$, для которых $g_i(x^k) = 0$.
В нашем случае есть единственное ограничение $g(x)$. И направление спуска будет сонаправленно с касательной к границе в точке x^k.

c. Проекцию точки на множество Ω, заданного одним неравенством $g(x) \leq 0$, находим методом последовательных приближений, т.е. движение в направлении антиградиента границы $g(x)$ с длиной шага соответвующему $g(x^*)$ - значеню функции в текущей точке.

d. Шаги алгоритма.

1. На k-ой итерации находимся в точке x^k.
   Проверяем условие $|x^{k-1} - x^k| < \epsilon_1$, или $|f(x^k) - f(x^{k-1})| < \epsilon_2$ или сразу оба. В случае их выполнения считаем точку x^k за условный минимум функции, иначе переходим к пункту 2.
2. Вычислим направление спуска. Для этого составим проекционную матрицу
   $P_{k+1} = I_n - A_{k+1}^T(A_{k+1}A_{k+1}^T)^{-1}A_{k+1}$, где I_n-единничная матрица, строками матрицы A_{k+1} будут матрицы-строки $(grad\ g_i(x_k))^T$. В нашем случае ограничение одно $g(x)$. Эта матрица спроектирует антиградиент $\omega^{k+1} = -grad\ f(x^k)$ на нужное направление.
   Получим нужное направление $p^{k+1} = P\omega^{k+1}$. Переходим к пункту 3.
3. Проведем минимизацию по направлению p^{k+1}.
   Находим минимум функции $\varphi(\alpha) = f(x^k + \alpha p^{k+1})$ по формуле $\alpha_{min} = \frac{(\omega^{k+1}, p^{k+1})}{(Qp^{k+1}, p^{k+1})}$.
   $\tilde{x} = x^k + \alpha_{min} p^{k+1}$. Переходим к пункту 4.
4. Точка $\tilde{x}$ лежит за пределами области Ω, поэтому необходимо спроецировать ее на это множество.
   $x_0^k = \tilde{x}$, $i = 0$ перейдем к пункту 5.
5. Находимся в x_i^k точке. Новую точку получим из соотношения $x_{i+1}^k = x_i^k - g(x_i^k)\frac{a_i}{|a_i|^2}$, где $a_i = grad\ g(x_i^k)$. Переходим к пункту 6.
6. Если выполннно условие условие $|x_{i-1}^k - x_i^k| < \epsilon_1$, или $|f(x_i^k) - f(x_{i-1}^k)| < \epsilon_2$, то считаем точку x_i^k за нужную проекцию точки $\tilde{x}$. Присваиваем $x^{k+1} = x_i^k$, $k = k+1$ и переходим к 1. Иначе $i = i+1$ и возвращаемся к 5.

5.3 Метод внутренних штрафных функций.

a. В данном методе вместо нахождения условного минимума функции $f(x)$ в области Ω проводят безусловную минимизацию функций $f_k(x) = f(x) + \delta_k(x)$, где функция штрафа

$$\delta_k(x) = \begin{cases} \geq 0, & x \text{ внутри области } \Omega \\ +\infty, & x \text{ на границе области } \Omega. \end{cases} \tag{3}$$

Эта функция создает барьер вдоль границы $g(x)$.

b. В роли такой функции может быть $\delta_k(x) = -r_k \sum_{i=1}^{m} \frac{1}{g_i(x)}$.
Рассмотрим случай границы заданной одним неравенством $g(x) \leq 0$. В этом случае $\delta_k(x) = \frac{-r_k}{g(x)}$. r_k - параметр штрафа. С каждой итерацией $k \to \infty$ и $r_k \to 0$.

d. Шаги алгоритма.

1. На k-ой итерации находимся в точке x^k.
   Если условие $|f(x^k) - f(x^{k-1})| < \epsilon_2$,то считаем точку x^k за условный минимум функции, иначе переходим к пункту 2.

2. Любым из методов безусловной минимизации находим минимум функции $f_k(x) = f(x) + \frac{-r_k}{g(x)}$, за начальное приближение берем x^k. Полученным минимум берем за начальную точку следующей итерации x^{k+1}.
   Устанавливаем $r^{k+1} = \frac{r^k}{2}$, $k = k + 1$ и переходим к пункту 1.