МИСиС
Национальный исследовательский технологический университет

Методы оптимизации

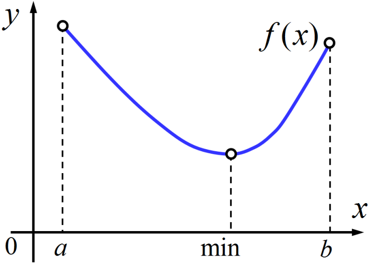
Лекция 2

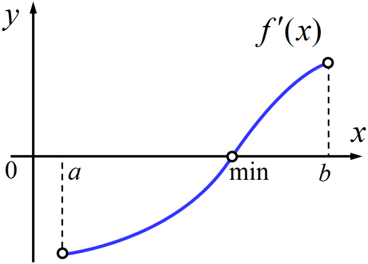
3 марта 2021 г.

v. 1.03

МЕТОДЫ ОДНОМЕРНОЙ МИНИМИЗАЦИИ С ИСПОЛЬЗОВАНИЕМ ПРОИЗВОДНОЙ

Общая идея методов





Метод средней точки

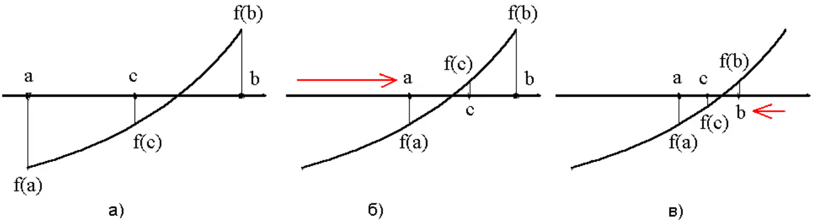
Метод средней точки направлен на повышение эффективности метода деления отрезка пополам при использовании технологии исключения отрезков за счёт замены вычислений функции в трёх точках на операцию вычисления производной в средней точке $\tilde{x} = (a + b)/2$.

Если $f'(\tilde{x}) > 0$, то точка $\tilde{x}$ лежит на участке монотонного возрастания $f(x)$, поэтому $x^* < \tilde{x}$ и точку минимума следует искать на отрезке $[a, \tilde{x}]$.

Если $f'(\tilde{x}) < 0$, то точка $\tilde{x}$ лежит на участке монотонного убывания $f(x)$, поэтому $x^* > \tilde{x}$ и точку минимума следует искать на отрезке $[\tilde{x}, b]$.

Равенство $f'(\tilde{x}) = 0$ означает, что точка минимума найдена точно и $x^* = \tilde{x}$.

Метод средней точки



Такое исключение отрезков требует на каждой итерации только одного вычисления $f'(\tilde{x})$ и уменьшает отрезок поиска точки минимума ровно в два раза.

Поиск заканчивается, если абсолютная величина производной меньше заданной погрешности.

Метод средней точки

Алгоритм.

1. Задаётся начальный интервал неопределённости $L_0 := [a_0, b_0]$ и $\varepsilon > 0$ — требуемая точность.

2. Задать $k = 0$.

3. Вычислить среднюю точку $\tilde{x} := \frac{a_k + b_k}{2}$, $f'(\tilde{x})$.

4. Проверить условие окончания:

- если $|f'(\tilde{x})| \leqslant \varepsilon$, то процесс поиска завершается и $x^* := \tilde{x}$, $f^* := f(x^*)$;
- если $|f'(\tilde{x})| > \varepsilon$, то сравнить $f'(\tilde{x})$ с нулём.
  Если $f'(\tilde{x}) > 0$, то продолжить поиск на отрезке $L_k := [a_k, b_k]$, положив $k := k + 1$, $a_k := a_{k-1}$, $b_k := \tilde{x}_{k-1}$.
  Если $f'(\tilde{x}) \leqslant 0$, то продолжить поиск на отрезке $L_k := [a_k, b_k]$, положив $k := k + 1$, $a_k := \tilde{x}_{k-1}$, $b_k := b_{k-1}$.
  Перейти к п. 3.

Метод хорд

Метод хорд опирается на равенство $f'(x) = 0$, которое является необходимым и достаточным условием глобального минимума выпуклой дифференцируемой функции $f(x)$.

Определение.

Функция f, определённая на выпуклом множестве $X \in \mathbb{R}^n$, называется выпуклой на X, если для произвольных элементов $x^1, x^2 \in X$, и для любого числа $\lambda \in [0, 1]$ выполняется неравенство Йенсена:

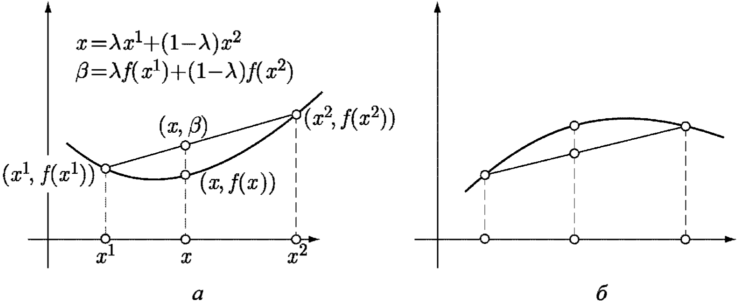
$$f(\lambda x^1 + (1-\lambda)x^2) \leqslant \lambda f(x^1) + (1-\lambda) f(x^2).$$

Определение.

Функция $f \colon X \to \mathbb{R}$ называется квазивыпуклой или унимодальной, если для произвольных элементов $x^1, x^2 \in X$ и $\lambda \in [0, 1]$ выполняется неравенство:

$$f(\lambda x^1 + (1-\lambda)x^2) \leqslant \max(f(x^1), f(x^2)).$$

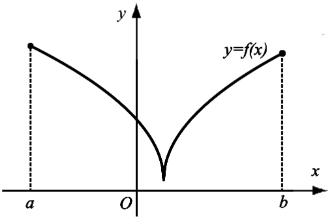
Геометрически выпуклость функции f означает, что любая точка произвольной хорды графика f располагается не ниже соответствующей точки самого графика. Для вогнутой функции взаимное расположение хорды и графика обратно.



Выпуклая функция (*а*), вогнутая функция (*б*)

Произвольная выпуклая функция является квазивыпуклой (унимодальной), произвольная вогнутая функция является квазивогнутой.

Пример унимодальной функции, не являющейся выпуклой:



Метод хорд

Теорема Больцано-Коши.

Пусть дана непрерывная функция на отрезке $f \in C([a,b])$. Пусть также $f(a) \neq f(b)$, и без ограничения общности предположим, что $f(a) = A < B = f(b)$. Тогда для любого $C \in [A,B]$ существует $c \in [a,b]$ такое, что $f(c) = C$.

Если на концах отрезка $L = [a,b]$ производная имеет разные знаки, то на интервале (a,b) найдется точка, в которой $f'(x) = 0$ и поиск точки минимума $f(x)$ на отрезке $[a,b]$ эквивалентен решению уравнения

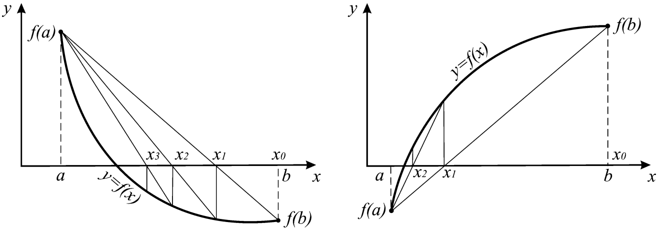
$$f'(x) = 0, \quad x \in [a,b].$$

Метод хорд

Таким образом, любой приближённый метод решения уравнения $f'(x) = 0$, $x \in [a, b]$ можно рассматривать как метод минимизации выпуклой дифференцируемой функции $f(x)$ на отрезке $[a, b]$. Одним из таких методов является метод хорд. Он основан на исключении отрезка путём определения точки

$$\tilde{x} = a - \frac{f'(a)}{f'(a) - f'(b)}(a - b)$$

пересечения с осью Ox хорды графика функции $f'(x)$ на очередном отрезке.



а) б)

Метод хорд

Новыми точками отрезка $[a, b]$ для осуществления следующей итерации являются концы того из отрезков $[a, \tilde{x}]$ и $[\tilde{x}, b]$, который содержит точку x^*. Его определяют по знаку производной $f'(\tilde{x})$.

Если $f'(\tilde{x}) > 0$, то точка $\tilde{x}$ лежит на участке монотонного возрастания $f(x)$, поэтому $x^* < \tilde{x}$ и точку минимума следует искать на отрезке $[a, \tilde{x}]$, то есть $b := \tilde{x}$.

Если $f'(\tilde{x}) < 0$, то точка $\tilde{x}$ лежит на участке монотонного убывания $f(x)$, поэтому $x^* > \tilde{x}$ и точку минимума следует искать на отрезке $[\tilde{x}, b]$, то есть $a := \tilde{x}$.

Равенство $f'(\tilde{x}) = 0$ означает, что точка минимума найдена точно и $x^* := \tilde{x}$.

На каждой итерации, кроме первой, следует вычислять одно новое значение $f'(x)$.

Поиск заканчивается, если абсолютная величина производной меньше заданной погрешности.

Метод хорд

Алгоритм поиска минимума функции методом хорд сводится к выполнению следующих этапов.

1. Задаётся начальный интервал неопределённости $L_0 := [a_0, b_0]$ и $\varepsilon > 0$ — требуемая точность.

2. Задать $k := 0$. Вычислить $f'(a_k), f'(b_k)$.

Если $f'(a_k)f'(b_k) < 0$, то перейти к п. 3, иначе к п. 5.

3. Вычислить $x_k := a_k - \frac{f'(a_k)}{f'(a_k)-f'(b_k)}(a_k - b_k)$ и $f'(x_k)$.

4. Проверить условие окончания:

   – если $|f'(x_k)| \leqslant \varepsilon$, то процесс поиска завершается и $x^* := x_k$, $f^* := f(x^*)$;
   – если $|f'(x_k)| > \varepsilon$, то сравнить $f'(x_k)$ с нулём.

Если $f'(x_k) > 0$, то продолжить поиск на отрезке $L_k = [a_k, b_k]$, положив $k := k+1$, $a_k := a_{k-1}$, $b_k := x_{k-1}$, $f'(b_k) := f'(x_{k-1})$.
Если $f'(x_k) \leqslant 0$, то продолжить поиск на отрезке $L_k = [a_k, b_k]$, положив $k := k+1$, $a_k := x_{k-1}$, $b_k := b_{k-1}$, $f'(a_k) := f'(x_{k-1})$.
Перейти к п. 3.

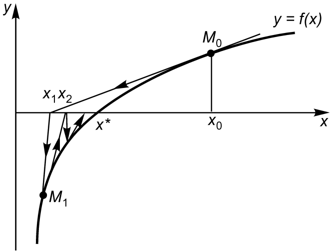
5. Если $f'(a_k) > 0$, $f'(b_k) > 0$, то $f(x)$ возрастает на отрезке $L_k = [a_k, b_k]$ и, следовательно, $x^* := a_k$.

Если $f'(a_k) < 0$, $f'(b_k) < 0$, то $f(x)$ убывает на отрезке $L_k := [a_k, b_k]$ и, следовательно, $x^* := b_k$.

Если $f'(a_k)f'(b_k) = 0$, то $x^* := a_k$ или $x^* := b_k$, в зависимости от того, на каком из концов отрезка $L_k = [a_k, b_k]$ производная $f'(x) = 0$.

Метод касательных (Ньютона-Рафсона)

Основная идея метода заключается в следующем: задаётся начальное приближение вблизи предположительного корня, после чего строится касательная к графику исследуемой функции в точке приближения, для которой находится пересечение с осью абсцисс. Эта точка берётся в качестве следующего приближения. И так далее, пока не будет достигнута необходимая точность.



Метод касательных (Ньютона-Рафсона)

Формула итеративного приближения x_n к x^* может быть выведена из геометрического смысла касательной следующим образом:

$$f'(x_n) = \operatorname{tg}\alpha_n = \frac{\Delta y}{\Delta x} = \frac{f(x_n) - 0}{x_n - x_{n+1}} = \frac{0 - f(x_n)}{x_{n+1} - x_n},$$

где α_n — угол наклона касательной прямой
$y(x) = f(x_n) + (x - x_n) \cdot \operatorname{tg}\alpha_n$ к графику f в точке $(x_n; f(x_n))$.

Следовательно (в уравнении касательной прямой полагаем $y(x_{n+1}) = 0$) искомое выражение для x_{n+1} имеет вид:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

Если $x_{n+1} \in (a, b)$, то это значение можно использовать в качестве следующего приближения к x^*.

Если $x_{n+1} \notin (a, b)$, то имеет место «перелёт» (корень x^* лежит рядом с границей (a, b)). В этом случае надо (воспользовавшись идеей метода половинного деления) заменять x_{n+1} на $\frac{x_n + x_{n+1}}{2}$ до тех пор, пока точка «не вернётся» в область поиска (a, b).

Метод касательных (Ньютона-Рафсона)

Алгоритм.

1. Задаётся начальный интервал неопределённости $L_0 = [a_0, b_0]$ и $\varepsilon > 0$ — требуемая точность.

2. Задать $k := 0$ и начальную точку $x_k \in [a_k, b_k]$.

3. Вычислить $f'(x_k)$. Проверить условие окончания:

   – если $|f'(x_k)| \leqslant \varepsilon$, то процесс поиска завершается и $x^* := x_k$, $f^* := f(x^*)$;

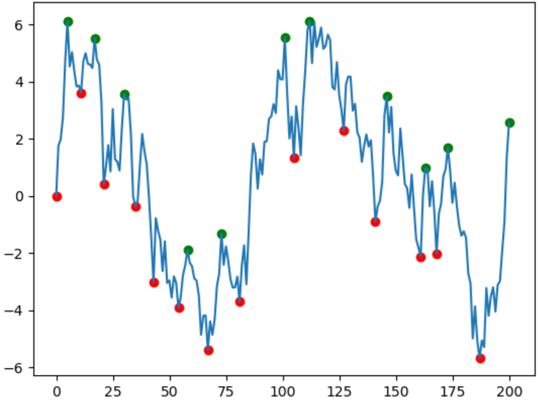
   – если $|f'(x_k)| > \varepsilon$, то вычислить $f''(x_k)$ и если $f''(x_k) > 0$ перейти к п. 4. В противном случае закончить вычисление в связи с нарушением обязательного условия $f''(x_k) > 0$.

4. Вычислить $x_{k+1} := x_k - \dfrac{f'(x_k)}{f''(x_k)}$.

5. Принять $k := k + 1$ и перейти к п. 3.

МЕТОДЫ МИНИМИЗАЦИИ МНОГОЭКСТРЕМАЛЬНЫХ ФУНКЦИЙ

Задача отыскания глобального экстремума многоэкстремальной функции существенно труднее задачи минимизации унимодальной функции.



Постановка задачи

Будем предполагать, что минимизируемая функция $f(x)$ удовлетворяет на отрезке $X = [a, b]$ условию Липшица с известной константой L:

$$|f(x) - f(x')| \leqslant L|x - x'|, \quad x, x' \in X.$$

Очевидно, что функция $f(x)$ непрерывна и поэтому достигает на X своего минимального значения.

Пусть $f(x^*) = \min\limits_{x \in X} f(x)$.

Точка глобального минимума может быть не единственной.

Будем предполагать, что задано либо число N вычислений значений функции $f(x)$, либо точность δ отыскания значения минимума.

Если метод даёт в качестве приближенного к минимальному значение $f(x^0)$, то его погрешностью будем считать величину $f(x^0) - f(x^*)$ (расстояние от точки x^0 до точек глобального минимума во внимание не принимается).

Метод перебора

Пусть задано число вычислений N. Положим

$$x_1^0 = a + \frac{b-a}{2N}, \quad x_2^0 = a + 3\frac{b-a}{2N}, \quad \dots, \quad x_N^0 = a + (2N-1)\frac{b-a}{2N}.$$

Вычислим значения $f(x_1^0), \dots, f(x_N^0)$ и в качестве приближения к минимальному примем значение $\min\limits_{i=1,\dots,N} f(x_i^0)$

Пусть x^* — точка глобального минимума. Тогда, найдётся точка $x_j^0 (j \in \{1, \dots, N\})$ такая, что $\left|x_j^0 - x^*\right| \leqslant \frac{b-a}{2N}$. С учётом условия Липшица имеем

$$0 \leqslant \min_{i=1,\dots,N} f(x_i^0) - f(x^*) \leqslant f(x_j^0) - f(x^*) \leqslant L\left|x_j^0 - x^*\right| \leqslant L\frac{b-a}{2N}.$$

Поэтому погрешность метода не превосходит $L\dfrac{b-a}{2N}$.

МИСиС

Положим $x_1 = a$, $x_2 = b$. Пусть вычисления проведены в точках $x_1, \ldots, x_i$: $y_1 = f(x_1), \ldots, y_i = f(x_i)$.

В силу условия Липшица

$$f(x) \geqslant y_j - L|x - x_j|, \quad x \in X, \quad j = 1, \ldots, i,$$

и поэтому

$$f(x) \geqslant \varphi_i(x) \stackrel{\text{def}}{=} \max_{j=1,\ldots,i} \{y_j - L|x - x_j|\}\,.$$

Функция φ_i является точной минорантой, принимающей в точках $x_1, \ldots, x_i$ значения соответственно $y_1, \ldots, y_i$.

Точка x_{i+1} выбирается по правилу $x_{i+1} = \arg\min\limits_{x \in X} \varphi_i(x)$.

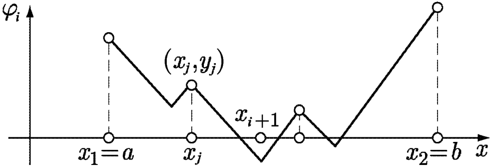


График функции φ_i

После N вычислений погрешность метода не превосходит величины

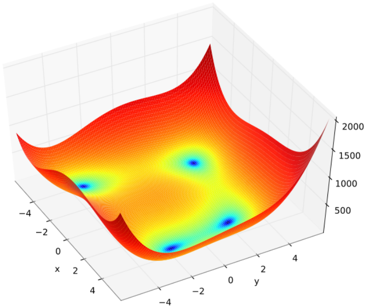
$$\min_{i=1,\dots,N} f(x_i) - \min_{x \in X} \varphi_N(x)$$

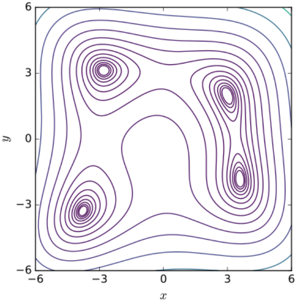
Если же требуется обеспечить отыскание значения минимума с точностью не хуже δ, то следует прекратить вычисления, как только

$$\min_{i=1,\dots,N} f(x_i) - \min_{x \in X} \varphi_N(x) \leqslant \delta$$

Функция Химмельблау

$$f(x, y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2$$





Функции нескольких переменных

Проблемы:

- нет универсальных алгоритмов поиска глобального минимума
- неясно, как выбрать начальное приближение (зависит от задачи и интуиции)

Подходы:

- методы локальной оптимизации (результат зависит от выбора начального приближения)
- случайный поиск (без гарантии)
- методы глобальной оптимизации (для особых классов функций)

МЕТОДЫ БЕЗУСЛОВНОЙ МИНИМИЗАЦИИ ГЛАДКИХ ФУНКЦИЙ

Обобщённый алгоритм

Перейдём к минимизации многомерных дифференцируемых функций. Решается задача

$$\min_{x \in \mathbb{R}^n} f(x), \quad f(x) \in C^1$$

Обобщённый алгоритм

1. $k := 1$, выбрать начальную точку x_1.

2. *Проверка условий останова*. Если условия выполнены, то поиск прекратить, x_k — решение, иначе перейти к п. 3.

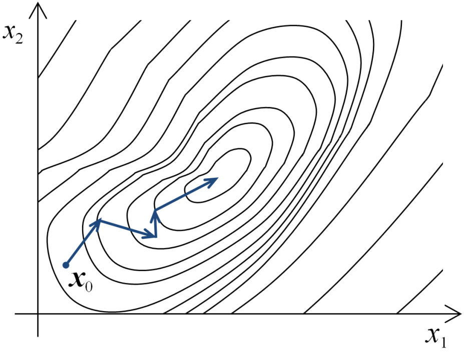
3. *Расчёт направления поиска* p_k.

4. *Расчёт длины шага*. Определить α_k, обеспечивающее, например, $f(x_k + \alpha_k p_k) < f(x_k)$.

5. *Пересчёт оценки решения*. $x_{k+1} := x_k + \alpha_k p_k$, $k := k + 1$, перейти к п. 2.

МИСиС

Процесс функционирования алгоритмов поиска экстремума функции удобно иллюстрировать с помощью изображения траектории поиска экстремума в плоскости линий уровня функции двух переменных $y = f(x_1, x_2)$.



Типовая траектория поиска минимума функции

Градиентный метод. Эвристические соображения

Предположим, что в любой точке x можно вычислить градиент функции $\nabla f(x)$.

Основная идея метода заключается в том, чтобы идти в направлении наискорейшего спуска, а это направление задаётся антиградиентом $-\nabla f(x)$.

В такой ситуации начиная с некоторого начального приближения x^0, строится итерационная последовательность

$$x^{k+1} = x^k - \gamma_k \nabla f(x^k)$$

где параметр $\gamma_k \geqslant 0$ задаёт скорость градиентного спуска.

Этот метод, редко применяется на практике в «чистом виде», но служит моделью для построения более реалистических алгоритмов.

Градиентный метод. Сходимость

Рассмотрим простейший вариант градиентного метода, в котором $\gamma_k \equiv \gamma$:

$$x^{k+1} = x^k - \gamma \nabla f(x^k) \tag{1}$$

Теорема 1.

Пусть $f(x)$ дифференцируема на $\mathbb{R}^n$, градиент $f(x)$ удовлетворяет условию Липшица: $\|\nabla f(x) - \nabla f(y)\| \leqslant L\|x - y\|$

$f(x)$ ограничена снизу: $f(x) \geqslant f^* > -\infty$

и γ удовлетворяет условию $0 < \gamma < 2/L$

Тогда в методе (1) градиент стремится к нулю:

$$\lim_{k \to \infty} \nabla f(x^k) = 0,$$

а функция $f(x)$ монотонно убывает: $f(x^{k+1}) \leqslant f(x^k)$.

Метод градиентного спуска с постоянным шагом

Алгоритм.

1. Задать начальную точку x^0, погрешности расчёта $\varepsilon_1 > 0$, $\varepsilon_2 > 0$, M — предельное число итераций. Найти градиент функции в произвольной точке $\nabla f(x) = \left(\frac{\partial f(x)}{\partial x_1}, \ldots, \frac{\partial f(x)}{\partial x_n}\right)^T$.

2. Принять $k := 0$.

3. Вычислить $\nabla f(x^k)$.

4. Проверить выполнение критерия окончания $\left\|\nabla f(x^k)\right\| < \varepsilon_1$:

   – если критерий выполнен, расчёт закончен $x^* := x^k$;

   – если критерий не выполнен, то перейти к п. 5;

5. Проверить выполнение неравенства $k \geqslant M$:

   – если неравенство выполнено, то расчёт окончен: $x^* := x^k$;

   – если нет, то перейти к п. 6.

МИСиС

Алгоритм.

6. Задать величину шага γ.
7. Вычислить $x^{k+1} := x^k - \gamma \nabla f(x^k)$.
8. Проверить выполнение условия $f(x^{k+1}) - f(x^k) < 0$:
   - если условие выполнено, то перейти к п. 9;
   - если условие не выполнено, принять $\gamma := \dfrac{\gamma}{2}$ и перейти к п. 7.
9. Проверить выполнение условий $\left\|x^{k+1} - x^k\right\| \leqslant \varepsilon_2$, $\left|f(x^{k+1}) - f(x^k)\right| \leqslant \varepsilon_2$:
   - если оба условия выполнены при текущем значении k и при $k-1$, то расчёт окончен и $x^* := x^{k+1}$;
   - если хотя бы одно из условий не выполнено, то принять $k := k + 1$ и перейти к п. 3.

Градиентный метод. Сходимость

В случае, когда минимизируемая функция f не предполагается выпуклой, градиентный метод может обеспечить лишь сходимость к множеству стационарных точек функции f.

Градиентный метод (1) (или даже с произвольным выбором γ_k), начатый из некоторой стационарной точки x^0, останется в этой точке: $x^k = x^0$ для всех k. Иными словами, градиентный метод «застревает» в любой стационарной точке — точке максимума, минимума или седловой.

Градиентный метод «не отличает» точек локального минимума от глобального и никакой гарантии сходимости к глобальному минимуму он не даёт.

В условиях теоремы 1 скорость сходимости $\nabla f(x^k)$ к 0 может быть очень медленной.

МИСиС

Градиентный метод. Сходимость

Определение.

Функция f, определённая на выпуклом множестве $X \in \mathbb{R}^n$, называется строго выпуклой на X, если

$$f(\lambda x^1 + (1-\lambda)x^2) < \lambda f(x^1) + (1-\lambda) f(x^2)$$

при всех $x^1, x^2 \in X$, $\lambda \in [0, 1]$.

Определение.

Функция f, определённая на выпуклом множестве $X \in \mathbb{R}^n$, называется сильно выпуклой с константой $l > 0$, если

$$f(\lambda x^1 + (1-\lambda)x^2) \leqslant \lambda f(x^1) + (1-\lambda) f(x^2) - l\lambda(1-\lambda)\frac{\left\|x^1 - x^2\right\|^2}{2}$$

при всех $x^1, x^2 \in X$, $\lambda \in [0, 1]$.

МИСиС

Рассмотрим поведение градиентного метода для более узкого класса функций — сильно выпуклых. В этом случае градиентный метод сходится к точке минимума со скоростью геометрической прогрессии.

Теорема 2.

Пусть $f(x)$ дифференцируема на $\mathbb{R}^n$, её градиент удовлетворяет условию Липшица с константой L и $f(x)$ является сильно выпуклой функцией с константой l. Тогда при $0 < \gamma < 2/L$ градиентный метод сходится к единственной точке глобального минимума x^* со скоростью геометрической прогрессии:

$$\left\|x^k - x^*\right\| \leqslant cq^k, \quad 0 \leqslant q < 1$$

$$\text{или } \left\|x^{k+1} - x^*\right\| \leqslant q\left\|x^k - x^*\right\|$$

Устойчивость решения

При практическом решении задач оптимизации постоянно приходится сталкиваться со следующей проблемой.

Пусть метод оптимизации приводит к построению минимизирующей последовательности, следует ли отсюда её сходимость к решению? То есть если $f(x)$ близко к $f(x^*)$, можно ли судить о близости точки x к решению x^*?

Такого типа проблемы относятся к области теории экстремальных задач, связанной с понятием устойчивости.

Для количественной оценки устойчивости используют следующий «нормированный» показатель.

Определение.

Назовём *обусловленностью* точки минимума x^* число

$$\mu = \overline{\lim_{\delta \to 0}} \left(\sup_{x \in L_\delta} \|x - x^*\|^2 \Big/ \inf_{x \in L_\delta} \|x - x^*\|^2 \right),$$

$$L_\delta = \{x : f(x) = f(x^*) + \delta\}.$$

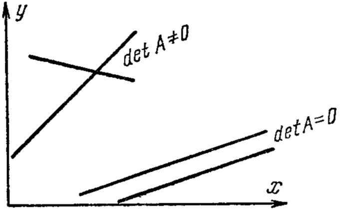
МИСиС

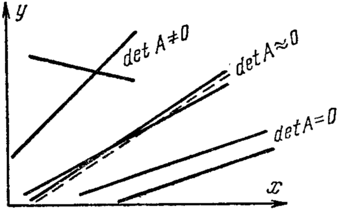
Иначе говоря, μ характеризует степень вытянутости линий уровня $f(x)$ в окрестности x^*. Ясно, что всегда $\mu \geqslant 1$. Если μ велико, то линии уровня сильно вытянуты — функция имеет овражный характер, т. е. резко возрастает по одним направлениям и слабо меняется по другим. В таких случаях говорят о плохо обусловленных задачах минимизации.

Если же μ близко к 1, то линии уровня $f(x)$ близки к сферам; это соответствует хорошо обусловленным задачам. В дальнейшем мы увидим, что число обусловленности μ возникает во многих проблемах, связанных с безусловной минимизацией, и может служить одним из показателей сложности задачи.

Ещё один пример плохо обусловленной задачи

Задача линейной алгебры. Решение системы уравнений $Ax = b$.
A — квадратная матрица, x и b — векторы.



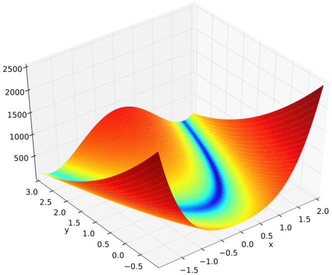


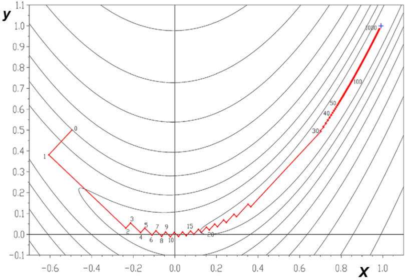
Плохо обусловленная система геометрически соответствует почти

Пример овражной функции

Функция Розенброка

$$f(x, y) = (1 - x)^2 + 100(y - x^2)^2$$



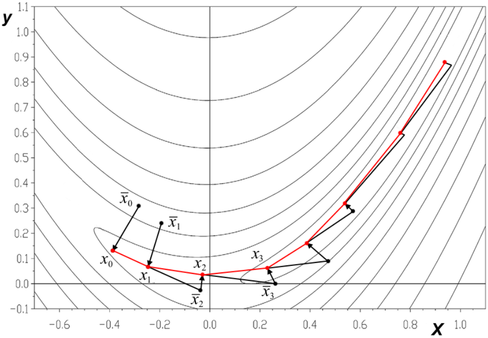


Это типичный пример овражной функции.

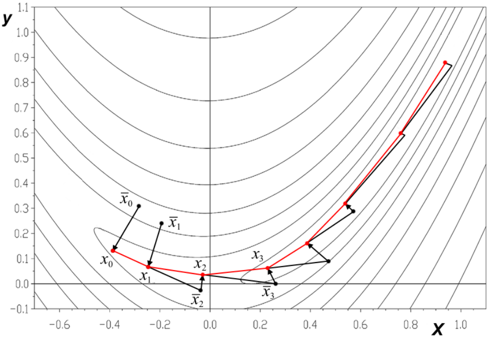
Градиентный метод «прыгает» с одного склона оврага на другой и обратно, иногда почти не двигаясь в нужном направлении, что существенно замедляет сходимость.

Овражный метод

Метод градиентного спуска оказывается очень медленным при движении по оврагу, причём при увеличении числа переменных целевой функции такое поведение метода становится типичным. Для борьбы с этим явлением используется метод оврагов, суть которого очень проста. Сделав два шага градиентного спуска и, получив две точки, третий шаг следует сделать в направлении вектора, соединяющего эти точки вдоль дна оврага.



Овражный метод



При изгибах оврага овражный шаг приводит на склон оврага, с которого осуществляется спуск в район дна оврага.

Отметим, что описанный метод не работает, если дно оврага имеет размерность выше двух.

Овражный метод

Алгоритм.

1. Задать точки $\bar{x}_0$, $\bar{x}_1$.

2. Из точек $\bar{x}_0$, $\bar{x}_1$ совершить один шаг методом наискорейшего спуска, получить точки x_0, x_1.

3. Положить $k := 1$.

4. Произвести овражный шаг, получить точку:

$$\bar{x}_{k+1} = x_k - \frac{x_k - x_{k-1}}{\|x_k - x_{k-1}\|} h \operatorname{sgn}\left(f(x_k) - f(x_{k-1})\right),$$

где h — овражный шаг.

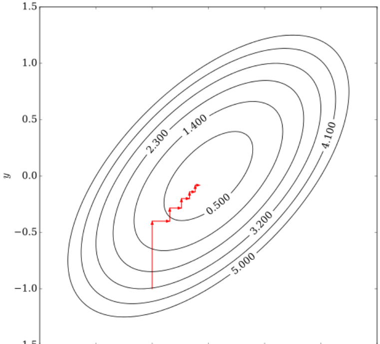
5. Из точки $\bar{x}_{k+1}$ совершить один шаг методом наискорейшего спуска, получить точку x_{k+1}.

6. Проверить условие останова. При выполнении поиск прекратить, запомнить наилучшую точку, иначе $k := k + 1$, и перейти к п. 4.

Метод циклического покоординатного спуска

Этот метод также называют методом Гаусса-Зейделя по аналогии с методом Гаусса-Зейделя для решения системы линейных уравнений.

Улучшает градиентный метод за счёт того, что на очередной итерации спуск осуществляется постепенно вдоль каждой из координат, однако новые γ_k теперь необходимо вычислять n раз за один шаг.



Алгоритм.

1. Задать начальную точку x^0, погрешности расчёта $\varepsilon_1 > 0$, $\varepsilon_2 > 0$, M — предельное число циклов расчёта, кратное n. Найти градиент функции $\nabla f(x)$ в произвольной точке.

2. Задать номер цикла $j := 0$.

3. Проверить условие $j \geqslant M$:

- если $j \geqslant M$, то принять $x^* := x^{jk}$ и расчёт окончен;
- если $j < M$, то перейти к п. 4.

4. Задать $k := 0$.

5. Проверить условие $k \leqslant n - 1$:

- если $k \leqslant n - 1$, то перейти к п. 6;
- если $k = n$, то принять $j := j + 1$ и перейти к п. 3.

6. Вычислить $\nabla f(x^{jk})$.

7. Проверить выполнение критерия окончания $\left\|\nabla f(x^{jk})\right\| < \varepsilon_1$:
   - если критерий выполнен, расчёт закончен $x^* := x^{jk}$;
   - если критерий не выполнен, то перейти к п. 8;
8. Найти t_k^* из условия:
$$t_k^* = \arg\min_{t_k \geqslant 0} f(x^{jk} - t_k \nabla f(x^{jk})_{k+1} e_{k+1})$$
9. Вычислить точку $x^{jk+1} := x^{jk} - t_k^* \nabla f(x^{jk})_{k+1} e_{k+1}$.
10. Проверить выполнение условий
$$\left\|x^{jk+1} - x^{jk}\right\| \leqslant \varepsilon_2, \quad \left|f(x^{jk+1}) - f(x^{jk})\right| \leqslant \varepsilon_2$$
    - если оба условия выполнены при текущем значении цикла j и при $j-1$, то расчёт окончен и $x^* := x^{jk+1}$;
    - если хотя бы одно из условий не выполнено, то принять $k := k+1$ и перейти к п. 5.

МИСиС

Метод наискорейшего спуска

Рассмотрим общий градиентный метод

$$x^{k+1} = x^k - \gamma_k \nabla f(x^k)$$

при различных способах выбора длины шага γ_k.

На первый взгляд, кажется, что можно значительно повысить эффективность градиентного метода, если идти до минимума по направлению антиградиента:

$$\gamma_k = \arg\min_{\gamma \geqslant 0} \varphi_k(\gamma), \quad \varphi_k(\gamma) = f(x^k - \gamma \nabla f(x^k)).$$

При этом мы получаем так называемый метод наискорейшего спуска.

Метод наискорейшего спуска

Алгоритм.

1. Задать начальную точку x^0, погрешности расчёта $\varepsilon_1 > 0$, $\varepsilon_2 > 0$, M — предельное число итераций. Найти градиент функции в произвольной точке $\nabla f(x) = \left(\frac{\partial f(x)}{\partial x_1}, \ldots, \frac{\partial f(x)}{\partial x_n}\right)^T$.

2. Принять $k := 0$.

3. Вычислить $\nabla f(x^k)$.

4. Проверить выполнение критерия окончания $\left\|\nabla f(x^k)\right\| < \varepsilon_1$:

   – если критерий выполнен, расчёт закончен $x^* := x^k$;

   – если критерий не выполнен, то перейти к п. 5;

5. Проверить выполнение неравенства $k \geqslant M$:

   – если неравенство выполнено, то расчёт окончен: $x^* := x^k$;

   – если нет, то перейти к п. 6.

Метод наискорейшего спуска

Алгоритм.

6. Задать величину шага γ^* из условия:

$$\gamma^* := \arg\min_{\gamma \geqslant 0} f(x^k - \gamma \nabla f(x^k))$$

7. Вычислить $x^{k+1} := x^k - \gamma^* \nabla f(x^k)$.

8. Проверить выполнение условий $\left\| x^{k+1} - x^k \right\| \leqslant \varepsilon_2$, $\left| f(x^{k+1}) - f(x^k) \right| \leqslant \varepsilon_2$:

 - если оба условия выполнены при текущем значении k и при $k-1$, то расчёт окончен и $x^* := x^{k+1}$;
 - если хотя бы одно из условий не выполнено, то принять $k := k+1$ и перейти к п. 3.

Вопросы?