Лабораторная работа 3.

Градиентные методы минимизации функций многих переменных.

Постановка задачи: Требуется найти безусловный минимум функции n переменных $f(x) = f(x_1, x_2, ..., x_n)$, т.е. такую точку $x^* \in E_n$, что $f(x^*) = \min\limits_{x \in E_n} f(x)$.

Предполагается, что целевая функция $f(x)$ − дифференцируема в E_n и возможно вычисление ее производных в произвольной точке E_n.

В данной работе рассматриваются итерационные процедуры минимизации вида

$$x^{k+1} = x^k + \alpha_k p^k, \quad k = 0,1,... \qquad (1)$$

где направление убывания p^k определяется тем или иным способом с учетом информации о частных производных функции $f(x)$, а величина шага $\alpha_k > 0$ такова, что

$$f(x^{k+1}) < f(x^k), \quad k = 1,2,... \qquad (2)$$

Метод градиентного спуска.

Стратегия поиска: Положим в (1) на каждом шаге $p^k = -\nabla f(x^k)$. Если $\nabla f(x^k) \neq 0$, то условие $(\nabla f(x^k), p^k) < 0$, очевидно, выполнено. Следовательно, направление вектора p^k является направлением убывания функции $f(x)$, причем в малой окрестности точки x^k направление p^k обеспечивает наискорейшее убывание этой функции. Поэтому найдется такое $\alpha_k > 0$, что будет выполняться условие (2).

Для сильно выпуклых функций метод градиентного спуска гарантирует сходимость последовательности $\{x^k\}$ к точке минимума x^* функции $f(x)$. Скорость сходимости метода − линейная.

Алгоритм:

1. Задать параметр точности $\varepsilon > 0$, начальный шаг $\alpha > 0$, выбрать $x \in E_n$. Вычислить $f(x)$.
2. Вычислить $\nabla f(x)$ и проверить условие достижения точности: $\|\nabla f(x)\| < \varepsilon$. Если оно выполнено, вычисления завершить, положив $x^* = x$, $f^* = f(x)$. Иначе – перейти к шагу 3.
3. Найти $y = x - \alpha \cdot \nabla f(x)$ и $f(y)$. Если $f(y) < f(x)$, то положить $x = y$, $f(x) = f(y)$ и перейти к шагу 2, иначе − к шагу 4.
4. Положить $\alpha = \alpha / 2$ и перейти к шагу 3.

Замечание. Вблизи стационарной точки функции $f(x)$ величина $\|\nabla f(x)\|$ становится малой. Это часто приводит к замедлению сходимости последовательности $\{x^k\}$. Поэтому в основной формуле (1) иногда полагают $p^k = -\nabla f(x^k) / \|\nabla f(x^k)\|$, используя вместо антиградиента вектор единичной длины в этом же направлении.

Метод наискорейшего спуска.

Стратегия поиска: Метод является модификацией метода градиентного спуска. Здесь также полагают $p^k = -\nabla f(x^k)$, но величина шага α_k из (1) находится в результате решения задачи одномерной оптимизации

$$\Phi_k(\alpha) \to \min, \quad \Phi_k(\alpha) = f(x^k - \alpha \nabla f(x^k)), \quad \alpha > 0, \qquad (3)$$

т.е. на каждой итерации в направлении антиградиента $-\nabla f(x^k)$ совершается исчерпывающий спуск.

Для сильно выпуклых функций метод наискорейшего спуска гарантирует сходимость последовательности $\{x^k\}$ к точке минимума x^* функции $f(x)$. Скорость сходимости метода – линейная.

Алгоритм:

1. Задать параметр точности $\varepsilon > 0$, выбрать $x \in E_n$. Вычислить $f(x)$.
2. Вычислить $\nabla f(x)$ и проверить условие достижения точности: $\|\nabla f(x)\| < \varepsilon$. Если оно выполнено, вычисления завершить, полагая $x^* = x, \quad f^* = f(x)$. Иначе – перейти к шагу 3.
3. Решить задачу одномерной оптимизации (3) для $x^k = x$, т.е. найти α^*. Положить $x = x - \alpha^* \cdot \nabla f(x)$ и перейти к шагу 2.

Замечание. Решение задачи (3) можно получить одним из рассмотренных для одномерной минимизации способов. Проверять выполнение условия (2) в этом методе необязательно. Если функция $f(x)$ квадратична, т.е. $f(x) = \frac{1}{2}(Ax, x) + (b, x) + c$, то для величины шага исчерпывающего спуска можно получить точную формулу $\alpha_k = -\frac{(\nabla f(x^k), p^k)}{(Ap^k, p^k)} = -\frac{(Ax^k + b, p^k)}{(Ap^k, p^k)}$, положив $p^k = -\nabla f(x^k)$.

На практике, значения многих исследуемых функций вдоль некоторых направлений изменяются гораздо быстрее (иногда на несколько порядков), чем в других направлениях. Их поверхности уровня сильно вытягиваются, а в более сложных случаях изгибаются и представляют собой овраги. Направление антиградиента у овражных функций существенно отклоняется от направления в точку минимума, что приводит к замедлению скорости сходимости.

Скорость сходимости градиентных методов существенно зависит также от точности вычислений градиента. Потеря точности, а это обычно происходит в окрестности точек минимума или в овражной ситуации, может вообще нарушить сходимость процесса градиентного спуска.

Вследствие перечисленных причин градиентные методы зачастую используются в комбинации с другими, более эффективными методами на начальной стадии решения задачи. В этом случае точка x^0 находится далеко от точки минимума, и шаги в направлении антиградиента позволяют достичь существенного убывания функции.

Метод сопряженных градиентов.

Стратегия поиска: В методах градиентного спуска в итерационной процедуре (1) в качестве направления убывания функции $f(x)$ использовалось направление антиградиента: $p^k = -\nabla f(x^k)$. Однако такой выбор направления убывания не всегда бывает удачным. В частности, для плохо обусловленных задач минимизации направление антиградиента в точке x^k может значительно отличаться от направления к точке минимума x^*. В результате, траектория приближения к точке минимума имеет зигзагообразный характер. В методе сопряженных градиентов применяется другой подход. Используется итерационный процесс

$$x^{k+1} = x^k + \alpha_k \cdot p^k, \quad k = 0,1,...; \quad x^0 \in E_n, \quad p^0 = -\nabla f(x^0),$$

в котором величина шага α_k находится из условия исчерпывающего спуска по направлению p^k:

$$f(x^k + \alpha_k\, p^k) = \min_{\alpha > 0} f(x^k + \alpha\, p^k), \quad k = 0,1,... .$$

Далее, после вычисления очередной точки x^{k+1}, $k = 0,1,...$, новое направление поиска p^{k+1} находится по формуле, отличной от антиградиента:

$$p^{k+1} = -\nabla f(x^{k+1}) + \beta_k\, p^k, \quad k = 0,1,...,$$

где коэффициенты β_k выбираются так, чтобы при минимизации квадратичной функции $f(x)$ с положительно определенной матрицей A получалась последовательность A- ортогональных векторов $p^0, p^1, ...$.

Одна из возможных формул для коэффициента β_k имеет вид

$$\beta_k = \frac{\left\|\nabla f(x^{k+1})\right\|^2}{\left\|\nabla f(x^k)\right\|^2}, \quad k = 1,2,...$$

Выражение для коэффициента β_k не содержит в явном виде матрицу A квадратичной формы. Поэтому метод сопряженных градиентов применяют для минимизации любых, а не только квадратичных функций.

Если функция $f(x)$ квадратична, т.е. $f(x) = \frac{1}{2}(Ax, x) + (b, x) + c$, то для величины шага исчерпывающего спуска можно получить точную формулу

$$\alpha_k = -\frac{(\nabla f(x^k), p^k)}{(Ap^k, p^k)}.$$

При реализации метода сопряженных градиентов применяется практический прием – через каждые N шагов производят обновление метода, полагая $\beta_{m \cdot N} = 0$, $m = 1,2,...$. Номера $m \cdot N$ называют моментами рестарта. Часто полагают $N = n$ - размерности пространства E_n. Если $N = 1$, то получается частный случай метода сопряженных градиентов – метод наискорейшего спуска.

Алгоритм:

1.Задать точность вычислений ε, выбрать начальное приближение x^0 и частоту обновления N.

2.Положить $k = 0$ (k - номер итерации). Вычислить значение $p^0 = -\nabla f(x^0)$.

3.Вычислить значение α_k из решения задачи одномерного исчерпывающего спуска $f(x^k + \alpha_k\, p^k) = \min\limits_{\alpha>0} f(x^k + \alpha\, p^k)$, либо другим методом

4.Вычислить точку $x^{k+1} = x^k + \alpha_k \cdot p^k$ и градиент $\nabla f(x^{k+1})$.

5.Проверить критерий окончания поиска $\left|\nabla f(x^{k+1})\right| < \varepsilon$. Если критерий выполнен, перейти к шагу 8.

6.Если $k+1 = N$, то положить $x^0 = x^{k+1}$, и перейти к шагу 2 (рестарт).

7.Вычислить коэффициент β_k по выбранной формуле и найти новое направление поиска $p^{k+1} = -\nabla f(x^{k+1}) + \beta_k\, p^k$. Положить $k = k+1$ и перейти к шагу 3.

8.Выбрать приближенно $x^* = x^{k+1}$, $f(x^*) = f(x^{k+1})$. Поиск завершен.

Дополнительные сведения о методе сопряженных градиентов

1.Минимизация квадратичных функций

Рассмотрим задачу поиска минимума квадратичной функции: $f(x) = c^T x + \frac{1}{2} x^T A x$, где A – симметричная положительно определенная матрица.

Обозначим для краткости $g^i = \nabla f(x^i)$ - вектор градиента, $y^i = g^{i+1} - g^i$, p^i - направление спуска в формуле $x^{i+1} = x^i + \alpha_i p^i$.

Рассмотрим набор $A-$ сопряженных ненулевых векторов, то есть: $(p^i)^T A p^j = 0$ для всех $i \neq j$. Так как эти вектора $A-$ сопряжены, то они являются линейно независимыми. Действительно, если представить один из этих векторов p^k как линейную комбинацию остальных: $p^k = a_0 p^0 + \ldots + a_{k-1} p^{k-1}$, то при умножении этого равенства на $(p^i)^T A$, получим, что $(p^i)^T A p^k = a_0 (p^i)^T A p^0 + \ldots + a_{k-1} (p^i)^T A p^{k-1} = 0$, что невозможно, так как $p^k \neq 0$ и A – положительно определенная матрица.

Для рассмотренного набора $A-$ сопряженных направлений соответствующий метод сопряженных направлений выглядит так: $x^{k+1} = x^k + \alpha_k p^k$. (Можно доказать, что последовательный исчерпывающий спуск по $A-$ ортогональным направлениям приводит к точке минимума квадратичной формы не более чем за n шагов)

По свойствам квадратичной функции $f(x)$ для любых точек x^{i+1} и x^i, связанных равенством $x^{i+1} = x^i + \alpha_i p^i$, имеем: $y^i = \nabla f(x^{i+1}) - \nabla f(x^i) = A(x^{i+1} - x^i) = \alpha_i A p^i$.

Получаем, что условие сопряженности $(p^i)^T A p^j = 0$ эквивалентно условию ортогональности: $(y^i)^T p^j = 0$.

Докажем теперь, что направление p^k для очередной итерации можно вычислять как линейную комбинацию градиента в текущей точке x^k и предыдущего направления p^{k-1}. Доказываем методом математической индукции.

Пусть p^0 совпадает в начальной точке с антиградиентом $-g^0$ в начальной точке x^0. Пусть k шагов метода спуска по взаимно сопряженным направлениям $p^0, \ldots, p^{k-1}$ уже выполнены, причем каждый из векторов p^i был суммой антиградиента $-g^i$ и линейной комбинации $p^0, \ldots, p^{i-1}$. Направление p^k для очередного шага тоже будем искать в таком виде: $p^k = -g^k + \sum\limits_{i=0}^{k-1} \beta_{ki} p^i$. Из этой формулы следует, что g^k - это линейная комбинация направлений $p^0, \ldots, p^k$. Учитывая $(g^i)^T p^i = 0$, можно утверждать $(g^k)^T g^i = 0, i < k$.

Чтобы выяснить, какие коэффициенты β_{kj} обеспечивают сопряженность p^k по всем предыдущим направлениям p^i, умножим $p^k = -g^k + \sum_{i=1}^{k-1} \beta_{kj} p^j$ на величину $(p^i)^T A$ слева:

$$(p^i)^T A p^k = -(p^i)^T A g^k + \sum_{j=0}^{k-1} \beta_{kj} (p^i)^T A p^j = -\frac{1}{\alpha_k} (y^i)^T g^k + \beta_{ki} (p^i)^T A p^i .$$

При преобразовании были использованы равенство $y^i = \alpha_i A p^i$ и предположение о взаимной сопряженности направлений $p^0,..., p^{k-1}$. В силу $(g^k)^T g^i = 0$ при $i < k-1$ следует положить соответствующие β_{ki} нулевыми. Таким образом, среди множителей β_{kj} ненулевым может быть только $\beta_{k,k-1}$. Обозначим его через β_{k-1}. Этот множитель надо подобрать так, чтобы вектор p^k был сопряжен к вектору p^{k-1}. Для этого достаточно удовлетворить условию ортогональности $(y^{k-1})^T p^k = 0$. Умножим $p^k = -g^k + \sum_{i=1}^{k-1} \beta_{ki} p^i$ на $(y^{k-1})^T$ слева и приравняем левую часть полученного равенства к нулю. Отсюда находим, что искомый множитель β_{k-1} должен быть решением уравнения $0 = -(y^{k-1})^T g^k + \beta_{k-1} (y^{k-1})^T p^{k-1}$, то есть $\beta_{k-1} = \dfrac{(y^{k-1})^T g^k}{(y^{k-1})^T g^{k-1}}$. Вывод: в качестве векторов p^k нужно брать $p^k = -y^k + \beta_{k-1} p^{k-1}$.

Можно показать, что метод, примененный к квадратичной функции, сходится не более чем за n шагов.

Можно также показать, что для квадратичной функции шаг исчерпывающего спуска по направлению p^k равен $\alpha_k = -\dfrac{(\nabla f(x^k), p^k)}{(Ap^k, p^k)}$.

2.Минимизация неквадратичных функций

Полученный метод минимизации квадратичных функций можно обобщить на задачи минимизации нелинейных функций общего вида. Для этого нужно изменить точную формулу расчета α_k на какую-нибудь процедуру одномерного поиска и решить, будет ли направление p^k всегда определяться по выведенной формуле. Выражение для коэффициента β_k, как оказывается, не содержит в явном виде матрицу A квадратичной формы, поэтому метод сопряженных градиентов может применяться для минимизации неквадратичных функций.

Вблизи точки минимума дважды дифференцируемая функция с положительно определенной матрицей Гессе, как правило, достаточно хорошо аппроксимируется квадратичной функцией. Поэтому можно надеяться на хороший результат применения метода сопряженных градиентов для функций такого вида.

3.Рестарт и частота рестарта

Построенный выше итерационный процесс может не приводить к точке минимума неквадратичной функции за конечное число итераций. Более того, точное определение величины α_k возможно только в редких случаях, а вектора p^k не образуют, вообще говоря, A–ортогональную систему относительно какой-либо матрицы A. Поэтому реализация каждой итерации метода будет сопровождаться неизбежными погрешностями. Эти погрешности, накапливаясь, могут привести к тому, что вектора p^k перестанут формировать направление убывания функции, и сходимость метода может нарушиться.

Поэтому на практике предлагается делать некоторые отступления от формулы $p^k = -\nabla f(x^k) + \beta_{k-1} p^{k-1}$, $k = 1,2,...$ – рестарты через каждые N шагов. Принимая во внимание n-шаговую теоретическую сходимость в квадратичном случае, можно через каждые $N = n$ шагов брать в качестве очередного вектора p^k антиградиент $-g^k$. Метод в такой реализации называется традиционным. Если $N = 1$, то получается частный случай метода сопряженных градиентов – метод наискорейшего спуска. Часто N берется немного большим, чем n.

4.Точность одномерной минимизации и направление убывания

Определение точного (в машинном смысле) минимума целевой функции $f(x)$ вдоль вектора p^k в общем случае требует больших усилий. Поэтому многие реализации традиционного метода сопряженных градиентов допускают приближенный одномерный поиск. Тогда необходимо заботиться о том, чтобы вектор p^k всегда являлся направлением спуска. При точном одномерном поиске это условие соблюдается автоматически. Из условия $p^k = -\nabla f(x^k) + \beta_{k-1} p^{k-1}$ следует: $(g^k)^T p^k = -(g^k)^T g^k + \beta_{k-1}(g^k)^T p^{k-1}$. Если минимум вдоль вектора p^{k-1} найден точно, то скалярное произведение $(g^k)^T p^{k-1}$ равно нулю, и поэтому величина $(g^k)^T p^k$, совпадающая с величиной $-(g^k)^T g^k$, будет отрицательной. Если же одномерный поиск ведется приближенно, то произведение $\beta_{k-1}(g^k)^T p^{k-1}$ может оказаться положительным и превосходящим $-(g^k)^T g^k$. Тогда вектор p^k будет указывать в сторону возрастания целевой функции $f(x)$. Эту проблему можно решить с помощью организации внеочередного рестарта, заменяя неудачное направление антиградиентом. Однако, если одномерный поиск организован непродуманно, то такие замены могут стать слишком частыми, и метод практически превратится в метод наименьшего спуска, то есть станет неэффективным. Также можно применять другой подход к выбору α_k, введя критерий приемлемости шага.

5.Точность одномерной минимизации методом поразрядного поиска

Метод поразрядного поиска, несмотря на простоту и меньшую эффективность, чем некоторые другие методы одномерной минимизации, имеет значительную область применимости в методах многомерной минимизации. Это связано с тем, что при реализации этого метода не обязательно задавать границы отрезка области поиска, роль которого может играть вся бесконечная прямая или любая ее часть (например, луч $\alpha_k > 0$).

6.Точность одномерной минимизации и точность решения основной задачи

Одним из важных практических вопросов является точность одномерной минимизации, необходимая для эффективных вычислений. Определение точного (в машинном смысле) минимума $F(x)$ вдоль p^k в общем случае требует больших усилий. Поэтому многие реализации традиционного метода сопряженных градиентов допускают приближенный одномерный поиск. Тогда на каждой итерации метода будут возникать неизбежные погрешности. Необходимо соблюдать баланс между трудоемкостью итераций и точностью поиска. С одной стороны, точный одномерный поиск необходим для того, чтобы не потерять сопряженность направлений и не понизить скорость сходимости. С другой стороны, слишком точная одномерная минимизация может оказаться вычислительно очень дорогой. На практике, таким образом, может быть применен метод проб и ошибок.

7.Способы нахождения шага спуска

Для решения на каждом шаге одномерной задачи оптимизации могут использоваться различные методы. Кроме того, можно пользоваться менее трудоемкими способами выбора параметра длины шага, например, правилом Армихо, правилом постоянного параметра, правилом Вулфа, правилом Голдстейна.

Наиболее известным является правило Вулфа, которое мы и рассмотрим. В тех случаях, когда вычисление производной функции не слишком трудоемко, правило Вулфа признается одним из самых эффективных правил решения задачи одномерного поиска.

Параметр длины шага должен удовлетворять неравенствам:

$$f(x^k+\alpha p^k)\le f(x^k)+c_1\alpha(\nabla f(x^k),p^k), \qquad \text{(V1)}$$

$$(\nabla f(x^k+\alpha p^k),p^k)\ge c_2(\nabla f(x^k),p^k) \qquad \text{(V2)}$$

где $0<c_1<c_2<1$.

Неравенство (V1),также известное как условие «Армихо», показывает, что уменьшение функции вдоль направления p^k должно быть пропорционально длине шага α и производной функции $f(x)$ по направлению p^k в точке x^k. На рисунке 1 показана графическая интерпретация неравенства (V1) и отмечена область, удовлетворяющая данному условию.

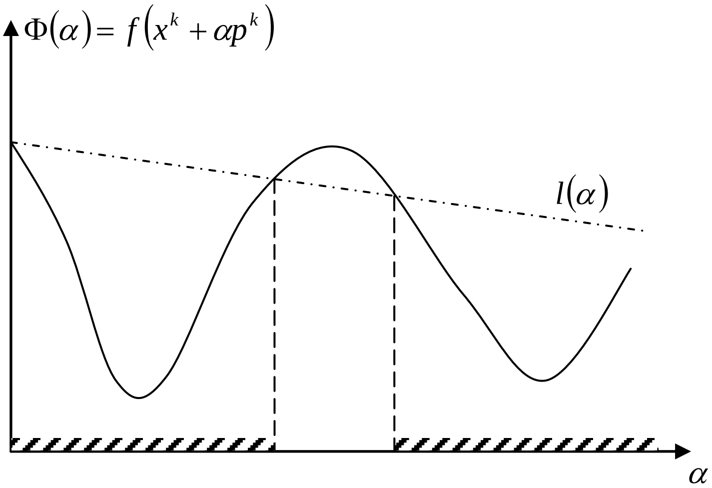

Рисунок 1. Иллюстрация условия (V1)

График функции $l(\alpha)$ имеет отрицательный наклон $c_1(\nabla f(x^k),p^k)$. Так как $0<c_1<1$ то прямая $l(\alpha)$ лежит выше, чем график функции $\Phi(\alpha)$, только при небольшом значении шага α. Поэтому условие (V1) выполняется тогда, когда $\Phi(\alpha)\le l(\alpha)$. На практике, c_1 выбирается достаточно малым, например $c_1=10^{-4}$.

Успешного выполнение условия (V1) не достаточно, чтобы алгоритмы многомерной минимизации имели разумное количество итераций, потому что условие (V1) выполняется для любого достаточно малого α. Для того, чтобы исключить малые шаги спуска в алгоритмах многомерной оптимизации, вводится второе условие (V2), известное как условие «изгиба». Правая часть этого условия представляет собой произведение $c_2\in(c_1,1)$ и производной функции $f(x)$ по направлению p^k в точке x^k. Выполнение условия «изгиба» гарантирует, что наклон функции $\Phi(\alpha)$ в c_2 раз больше чем $\Phi'(0)$ (рисунок 2). Обычно, при использовании условия в методе сопряжённых градиентов параметр $c_2=0.1$. На рисунке 2 отмечены значения шага α, при которых выполнено условие (V2).

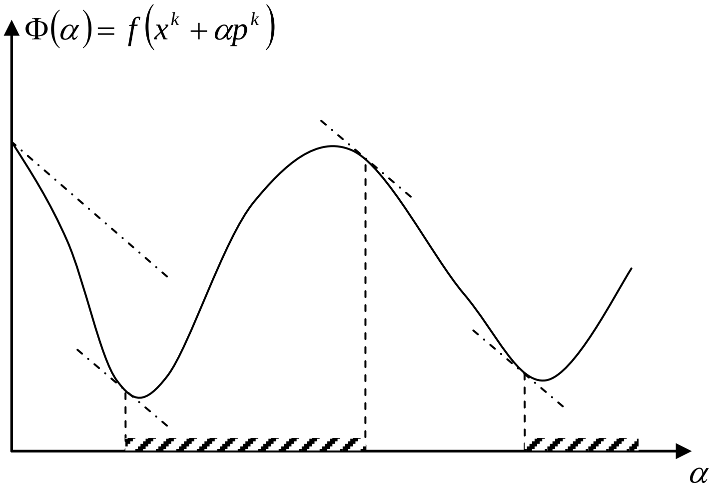

Рисунок 2. Иллюстрация условия (V2)

Алгоритм нахождения шага спуска, удовлетворяющий правилам Вульфа:
1. Положить $\alpha_0 = 0$, выбрать $\alpha_{\max} > 0$ и $0 < \alpha_1 < \alpha_{\max}$.
2. Положить $i = 1$.
3. Вычислить $\Phi(\alpha_i) = f(x^k - \alpha_i\ p^k)$. Если $\Phi(\alpha_i) > f(x^k) + c_1 \alpha_i\ (\nabla f(x^k), p^k)$ или $\Phi(\alpha_i) \ge \Phi(\alpha_{i-1})$, если $i > 1$, то выполнить алгоритм уточнения шага спуска с параметрами $\alpha_L = \alpha_{i-1}$ и $\alpha_H = \alpha_i$ (см. ниже) и завершить алгоритм.
4. Вычислить $\Phi'(\alpha_i) = \left(\nabla f(x^k + \alpha_i\ p^k), p^k\right)$.
5. Если $\left|\Phi'(\alpha_i)\right| \le -c_2(\nabla f(x^k), p^k)$, то алгоритм завершен, величина шага спуска равна α_i .
6. Если $\Phi'(\alpha_i) \ge 0$, то выполнить алгоритм уточнения шага спуска с параметрами $\alpha_L = \alpha_i$ и $\alpha_H = \alpha_{i-1}$ (см. ниже) и завершить алгоритм.
7. Выбрать $\alpha_{i+1} \in (\alpha_{i,} \alpha_{\max,})$, положить $i = i + 1$, перейти к шагу 3.

Последовательность пробных шагов $\{\alpha_i\}$ монотонно возрастает, но порядок аргументов, которые передаются в алгоритм уточнения может меняться. Алгоритм использует то факт, что интервал (α_{i-1}, α_i) содержит шаг, удовлетворяющий неравенствам (V1) и (V2), если выполняются следующие условия:

- α_i противоречит условию уменьшения значения функции (неравенству (V1));
- $\Phi(\alpha_i) \ge \Phi(\alpha_{i-1})$;
- $\Phi'(\alpha_i) \ge 0$.

Последний шаг алгоритма выполняет экстраполяцию, для нахождения следующего пробного шага.

Рассмотрим алгоритм уточнения шага. Параметры алгоритма α_L и α_H должны удовлетворять следующим условиям:

- интервал, ограничиваемый α_L и α_H, должен содержать шаг α удовлетворяющий правилу Вульфа;
- среди всех выполненных пробных шагов, удовлетворяющих условию уменьшения(V1), α_L является шагом с наименьшим значением функции $\Phi(\alpha) = f(x^k - \alpha\ p^k)$;
- α_H выбран так, что $\Phi'(\alpha_H)(\alpha_H - \alpha_L) < 0$

Каждый шаг алгоритма создает новый пробный шаг спуска α_j между α_L и α_H, затем переопределяет одну из граничных точек на α_j.

Алгоритм уточнения шага спуска:
1. В качестве начального интервала поиска задается интервал, ограниченный α_L и α_H.
2. Положить $j = 1$.
3. Выбрать α_j между α_L и α_H.
4. Вычислить $\Phi(\alpha_j) = f(x^k - \alpha_j\ p^k)$.
5. Если $\Phi(\alpha_j) > f(x^k) + c_1\alpha_j(\nabla f(x^k), p^k)$ или $\Phi(\alpha_j) \ge \Phi(\alpha_L)$, то положить $\alpha_H = \alpha_j$, $j = j+1$ и перейти к шагу 3, иначе перейти к шагу 6
6. Если $\left|\Phi'(\alpha_j)\right| \le -c_2(\nabla f(x^k), p^k)$, то алгоритм завершен, величина шага спуска равна α_j.
7. Если $\left|\Phi'(\alpha_j)\right|(\alpha_H - \alpha_L) \ge 0$, то положить $\alpha_H = \alpha_L$.
8. Положить $\alpha_L = \alpha_j$ и перейти к шагу 3.

8.Виды коэффициента сопряжённости направлений

При определении стратегии поиска в п.1 дополнительных сведений была выведена следующую формулу для расчета β_{k-1}:

$$\beta_{k-1} = \frac{(y^{k-1})^T g^k}{(y^{k-1})^T g^{k-1}} \qquad \text{(В1).}$$

Взаимная ортогональность градиентов и определение величины p^{k-1} позволяют предположить еще формулы, внешне отличные, но теоретически эквивалентные в случае квадратичной функции для расчета β_{k-1}. Действительно, преобразуем знаменатель формулы (В1): $(y^{k-1})^T g^{k-1} = [g^k - g^{k-1}]g^{k-1} = -\left\|g^{k-1}\right\|^2$. Получим формулу Полака-Рибьера:

$$\beta_{k-1} = \frac{(y^{k-1})^T g^k}{\left\|g^{k-1}\right\|^2} \qquad \text{(В2).}$$

Проведя аналогичные преобразования с числителем, и получим формулу Флетчера-Ривса:

$$\beta_{k-1} = \frac{\left\|g^k\right\|^2}{\left\|g^{k-1}\right\|^2} \qquad \text{(В3).}$$

Формулу (В1) практически никто не использует, две оставшиеся вызывают споры. В литературе можно найти и другие формулы для коэффициента β_{k-1}.

Приведенные формулы эквивалентны в случае минимизации квадратичной функции, но для неквадратичной функции значения β_{k-1} будут, вообще говоря, различными. Поэтому использование каждой из этих формул при минимизации неквадратичной функции может привести к построению своей системы направлений спуска. При этом, вообще говоря, точка минимума не будет найдена за конечное число шагов.

В случае минимизации неквадратичной функции считается, что метод (В2) предпочтительнее метода (В3). Частичное объяснение этому мнению может состоять в следующем. В связи с неквадратичностью задачи и возможными неточностями в одномерной минимизации, сопряженность полученных направлений может постепенно теряться, что может привести к тому, что метод «застрянет», в том смысле, что генерируемые направления p^k будут почти ортогональны градиенту g^k. Когда это случится, получится, что $g^{k+1} \cong g^k$. В этом случае число β_k, получаемое по формуле (В2), будет практически равно нулю, и следующее направление $p^{k+1} = -y^{k+1} + \beta_k p^k$ будет близко к $-g^{k+1}$, таким образом не допуская «застревания». На практике алгоритм Флетчера-Ривса может и не «застревать», а для некоторых задач работать даже быстрее алгоритма Полака-Рибьера.

9.Эффективность метода сопряженных градиентов в задачах с большой размерностью

В случае, если размерность n аргумента целевой функции становится очень большой, при обращении к обычным методам минимизации могут возникнуть значительные трудности. Во-первых, время вычисления может вырасти настолько, что решение такой задачи станет бессмысленным. Во-вторых, что не менее важно, в памяти машины может не найтись места, достаточного для хранения матриц, используемых при расчете направлений поиска.
Метод сопряженных градиентов относится к классу алгоритмов минимизации, в которых вычисление направлений поиска не предполагает решения каких-либо систем линейных уравнений. Поэтому метод сопряженных градиентов можно применять для решения таких задач, к которым другие методы неприменимы из-за того, что фигурирующие в них матрицы оказываются слишком большими или слишком сильно заполненными.
В настоящее время схема метода сопряженных градиентов является наиболее приемлемым универсальным средством решения задач безусловной минимизации с очень большим числом переменных (сотни и тысячи переменных).

10.Память метода сопряженных градиентов

Из формулы $p^k = -\nabla f(x^k) + \beta_{k-1} p^{k-1}$, $k = 1,2,...$ следует, что для формирования вектора p^k не требуется операций с матрицами. Метод сопряженных градиентов требует хранения всего лишь 3-х векторов размерности n.
Каждая итерация метода сопряженных градиентов требует вычисления функции и ее градиента (возможно, несколько раз в процессе одномерной минимизации) и $O(n)$ операций для вычисления сопряженного направления p^k и следующей точки x^{k+1}.

11.Скорость сходимости

Так как метод, примененный к квадратичной функции, сходится не более чем за n шагов, можно было бы предположить, что для неквадратичной функции скорость сходимости будет сравнима со скоростью сходимости метода Ньютона. Действительно, для достаточно широкого класса функций теоретическая скорость его сходимости при точной одномерной минимизации оказывается n-шаговой сверхлинейной. Доказательство этого утверждения существенно опирается на наличие рестартов. Алгоритмы без рестартов для большинства функций сходятся линейно. Однако это – теоретические оценки, а на практике из-за ошибок округления реальная скорость сходимости метода сопряженных градиентов почти всегда линейна независимо от того, применяются рестарты или нет.

12.Итог

Метод сопряженных градиентов имеет 3 важных свойства: а) минимизирует квадратичную функцию от n переменных за n шагов; б) не требует вычисления второй производной; в) не требует хранения или инвертирования матриц.
Последнее свойство делает метод применимым к задачам с большой размерностью. Один из недостатков метода сопряженных градиентов – при его реализации требуется высокая точность одномерного поиска.

Задания

1. Реализовать в среде MATLAB метод градиентного спуска, метод наискорейшего спуска и метод сопряженных градиентов.

В методе наискорейшего спуска и в методе сопряженных градиентов наряду с аналитическим выражением для величины шага исчерпывающего спуска для квадратичной функции реализовать решение задач одномерной минимизации методом поразрядного поиска. Обратить внимание, что при выборе оптимального метода наиболее важным критерием является количество вычислений функций и ее производной.

2. Протестировать работу реализованных методов на примере овражной функции

$$f(x) = x_1^2 + a\,x_2^2,$$

при a = 1, 250, 1000. При $\varepsilon = 10^{-3}$ и $\varepsilon = 10^{-5}$ сравнить скорость работы методов при различных значениях параметра a по числу итераций и по числу вызовов совокупности значений функций и производных.

3. Выбрать для выполнения работы тестовую функцию, номер которой соответствует номеру Вашего компьютера. Например, для компьютера №3 это будет функция 3), для компьютера №13 – функция 4): 13-9=4; для компьютера №23 это будет функция 5): 23-9$\times$2=5.

1) $f(x) = 64x_1^2 + 126x_1x_2 + 64x_2^2 - 10x_1 + 30x_2 + 13$

2) $f(x) = 129x_1^2 - 256x_1x_2 + 129x_2^2 - 51x_1 - 149x_2 - 27$

3) $f(x) = 254x_1^2 + 506x_1x_2 + 254x_2^2 + 50x_1 + 130x_2 - 111$

4) $f(x) = 151x_1^2 - 300x_1x_2 + 151x_2^2 + 33x_1 + 99x_2 + 48$

5) $f(x) = 85x_1^2 + 168x_1x_2 + 85x_2^2 + 29x_1 - 51x_2 + 83$

6) $f(x) = 211x_1^2 - 420x_1x_2 + 211x_2^2 - 192x_1 + 50x_2 - 25$

7) $f(x) = 194x_1^2 + 376x_1x_2 + 194x_2^2 + 31x_1 - 229x_2 + 4$

8) $f(x) = 45x_1^2 - 88x_1x_2 + 45x_2^2 + 102x_1 + 268x_2 - 21$

9) $f(x) = 99x_1^2 + 196x_1x_2 + 99x_2^2 - 95x_1 - 9x_2 + 91$

4. Сравнить эффективность методов градиентного и наискорейшего спуска, а так же метода сопряженных градиентов для задачи п.2 при a=250 и тестовой функции п.3. *Объяснить полученные результаты. Результаты расчетов сохранить для сравнения с результатами, полученными прямыми методами в лабораторной работе №5.*

5. Минимизировать функцию Розенброка

$$f(x) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2$$

с точностью $\varepsilon = 10^{-3}$ и $\varepsilon = 10^{-5}$, выбрав начальную точку $x^0 = (-1,\ 1)^T$.

Для решения задачи использовать метод сопряженных градиентов. Параметр точности одномерного поиска установить постоянным, например, 10^{-6}. Определить, сколько вычислений функций и ее производной потребуется методу для того, чтобы разность между численным решением и точным решением $x^* = (1,\ 1)^T$ была меньше ε. *Результаты расчетов сохранить для сравнения с результатами, которые будут получены квазиньютоновскими методами в Л.Р. №4 и прямыми методами в Л.Р. №5.*

6. Исследовать работу метода сопряженных градиентов для функции Розенброка в зависимости от частоты рестарта N (ее варьировать, например, в диапазоне от 1 до 5). *Какое значение можно назвать оптимальным?*

7. Выбрать формулу для расчета величины β_k. Установить постоянной частоту рестарта (можно взять равной $N = n$, либо воспользоваться результатами предыдущего задания).

Для решения задачи одномерной минимизации использовать методы: поразрядного поиска, золотого сечения. Параметр точности одномерного поиска менять (например, задавать 10^{-4}, 10^{-6}, 10^{-8}). Как зависит точность нахождения решения основной задачи от точности одномерной минимизации?

Реализовать нахождение шага спуска с помощью правила Вулфа. Определить значения c_1 и c_2, при которых минимум находится за меньшее число итераций.

Сравнить работу программы (точность решения основной задачи, количество итераций, количество вычислений функций и ее производной) при использовании методов поразрядного поиска, золотого сечения, правила Вулфа (для каждого метода рассматривайте его наилучший результат).

8. На примере функции Химмельблау

$$f(x) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$$

рассмотреть особенности применения градиентных методов для минимизации многомодальных функций. В качестве начального приближения взять точки $(0, 0)$ и $(-5, 0)$. Как зависит работа рассматриваемых алгоритмов от выбора начального приближения?

9. Сдать лабораторную работу преподавателю, *ответив предварительно на все следующие контрольные вопросы.*

Контрольные вопросы к Лабораторной работе 3

1).Какие функции называются квадратичными функциями n переменных?

2). Чему равен градиент, гессиан квадратичной функции?

3).Каким свойством обладает квадратичная функция с положительно определенной матрицей A?

4).При каких значениях a, b, c функция $f(x) = ax_1^2 + bx_1x_2 + cx_2^2$ будет выпукла?

5).Выписать матрицу A квадратичной функции $f(x) = x_1^2 + 3x_3^2 + 2x_1x_2 - x_2x_3$ $+ 2x_2 + x_3$.

6).Какая последовательность $\{x^k\}$, $k = 0,1,2...$ называется минимизирующей?

7).Привести пример минимизирующей последовательности, не сходящейся к точке минимума.

8).Что такое скорость сходимости минимизирующей последовательности? Какие скорости сходимости Вы знаете?

9).Когда говорят, что в итерационном процессе $x^{k+1} = x^k + \alpha_k p^k$, $k = 0,1,...$ производится исчерпывающий спуск?

10).Какие направления дифференцируемой в точке x^k функции $f(x)$ называются направлениями убывания? Каков геометрический смысл направления убывания?

11).Какова скорость сходимости метода градиентного спуска для квадратичной функции $f(x)$ с положительно определенной симметрической матрицей A, где $0 < l < L$ - ее наименьшее и наибольшее собственные значения?

12).Когда говорят, что сильно выпуклая функция $f(x)$ имеет "овражный характер"? Какие задачи минимизации называются хорошо обусловленными, а какие – плохо обусловленными?

13). С какой целью в методе градиентного спуска иногда полагают $p^k = -\frac{\nabla f(x^k)}{\|\nabla f(x^k)\|}$, используя вместо антиградиента вектор единичной длины в этом же направлении?

14).В чем состоят преимущества и недостатки метода наискорейшего спуска по сравнению с методом градиентного спуска?

15). Каков главный недостаток градиентных методов?

16).В чем состоит идея метода сопряженных градиентов? Чем этот метод отличается от методов градиентного и наискорейшего спуска?

17).Эквивалентны ли различные формулы, применяющиеся для расчета β_k?

18).Зачем в традиционном варианте метода сопряженных градиентов производится рестарт?

19).Что можно сказать о скорости сходимости метода сопряженных градиентов?

20).Назовите основные преимущества и недостатки метода сопряженных градиентов.

21).Назовите те области применения, где метод сопряженных градиентов наиболее эффективен по сравнению с другими методами оптимизации.