Схема метода сопряженных градиентов (из лекции 9):

$$p^0 = -\nabla f(x^0), \qquad x^0 \in E_n$$

$$x^{k+1} = x^k + \alpha_k p^k$$

$f(x^k + \alpha_k p^k) = \min_{\alpha>0} f(x^k + \alpha p^k)$ *или для квадратичной функции*

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

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

$$\beta_k = \frac{(A\nabla f(x^{k+1}), p^k)}{(Ap^k, p^k)}$$

При этом предполагалось, что для вычисления градиента будет использоваться аналитическая формула:

$$\nabla f(x^k) = Ax^k + b.$$

Что плохого будет, если начать программировать эти формулы без предварительной адаптации к минимизации вычислительных ресурсов?

Ответ – три матрично-векторных произведения, что значительно увеличивают время на одну итерацию в методе сопряженных градиентов.

Что необходимо сделать предварительно?

Ответ – провести минимальные подстановки в формулы и получить наиболее оптимальную вычислительную схему.

Начнем. Вспоминаем основные соотношения, которыми оперировали в ходе лекций по градиентным методам (лекции 7-9):

$$\nabla f(x^k) = Ax^k + b,$$

$$\left(\nabla f(x^i), \nabla f(x^{k+1})\right) = 0 \text{ и } \left(\nabla f(x^{k+1}), p^i\right) = 0 \text{ для } i \le k.$$

Далее для $\nabla f(x^{k+1})$:

$$\nabla f(x^{k+1}) = A(x^k + \alpha_k p^k) + b = \nabla f(x^k) + \alpha_k A p^k,$$

из последнего

$$Ap^k = \alpha_k^{-1}\left(\nabla f(x^{k+1}) - \nabla f(x^k)\right).$$

Из условия А-ортогональности $\left(Ap^i, p^j\right) = 0$ и соотношений выше выражение для β_k упрощается:

$$\beta_k = \frac{(A\nabla f(x^{k+1}), p^k)}{(Ap^k, p^k)} = \frac{\left(Ap^k, \nabla f(x^{k+1})\right)}{(Ap^k, p^k)} = \frac{\left(\alpha_k^{-1}\left(\nabla f(x^{k+1}) - \nabla f(x^k)\right), \nabla f(x^{k+1})\right)}{\left(\alpha_k^{-1}\left(\nabla f(x^{k+1}) - \nabla f(x^k)\right), p^k\right)} =$$

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

Подставляя аналогично в выражение для α_k, получим выражение для квадратичной функции:

$$\alpha_k = \frac{\|\nabla f(x^k)\|^2}{(Ap^k, p^k)}$$

В итоге схема для программирования будет такой:

1. выбрать $x^0 \in E_n$, вычислить $\nabla f(x^0) = Ax^0 + b$, задать $p^0 = -\nabla f(x^0)$.

2. далее по итерациям (до условия выхода):

$$\alpha_k = \frac{\|\nabla f(x^k)\|^2}{(Ap^k, p^k)}$$

$$x^{k+1} = x^k + \alpha_k p^k$$

$$\nabla f(x^{k+1}) = \nabla f(x^k) + \alpha_k A p^k$$

$$\beta_k = \frac{\|\nabla f(x^{k+1})\|^2}{\|\nabla f(x^k)\|^2}$$

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

Вычисление Ap^k происходит единоразово на каждой итерации!

Для вычисления градиента не нужно применять численные разностные схемы!

В отчете

1. сравнение методов делаем по итерациям, а не по количеству вычислений функции, как это было в первой лабораторной;
2. для п.2 требование придумать функции, на которых методы будут вести себя по-разному, именно это и означает – нужно придумать функции и задать соответствующие начальные приближения, на которых работа методов будет, действительно, отличаться;
3. для исследований в п.3 не ограничиваемся маленькими пространствами, а размерность пространства задаем $n = 10, 10^2, 10^3, 10^4$. (На вопрос – как же разместить в памяти матрицу большой размерности? Ответ – полагаю, что вы будете генерировать ее диагональной, поэтому проблем с размещением не будет);

   число обусловленности задаем и исследуем в диапазоне от 1 до 1000-2000;
4. в отчете при описании запускаемого теста для каждого метода вы должны указать все параметры запуска (начальное приближение, коэффициенты и

т.п.), указать условия выхода (число итераций, найденный вектор решения, полученную точность), проиллюстрировать работу метода либо графиками, которые получили от собственной разработки, либо от сторонних программ отрисовки.

Если вы реализуете собственную программу отрисовки, то должны быть в качестве инструментария в интерфейсе:

- возможность отображения/скрытия линий уровня функции,
- масшабирования изображения,
- если для указания направления спуска используете стрелочки, то необходима возможность их скрыть/показать;
- подписи к координатным линиям (скрыть/показать);
- координатные оси (скрыть/показать);
- кнопки перехода (вперед/назад) по итерациям;
- выбор метода решения (среди 3х заданных),
- задание начального приближения, точности.

Примеры функций (**для исследования в п.2. можно взять только одну**):

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

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

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

Вопросы к защите

1. Чему равны градиент и гессиан квадратичной функции?
2. Каким свойством обладает квадратичная функция с положительно определенной матрицей A?
3. Чем можете охарактеризовать собственный вектор матрицы A квадратичной функции?
4. Когда говорят, что в итерационном процессе производится исчерпывающий спуск?

5. Какие направления дифференцируемой в точке функции называются направлениями убывания? Каков геометрический смысл направления убывания?
6. Какова скорость сходимости метода градиентного спуска для квадратичной функции с положительно определенной симметричной матрицей A, где – ее l и L наименьшее и наибольшее собственные значения?
7. Когда говорят, что сильно выпуклая функция имеет овражный характер? Какие задачи минимизации называются хорошо обусловленными, а какие – плохо обусловленными?
8. В чем состоят преимущества и недостатки метода наискорейшего спуска по сравнению с методом градиентного спуска?
9. Каков главный недостаток градиентных методов?
10. В чем состоит идея метода сопряженных градиентов? Чем этот метод отличается от методов градиентного и наискорейшего спуска?