

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «МЭИ»

ВЕСТФАЛЬСКИЙ А.Е.

ВВЕДЕНИЕ В ПРОЕКЦИОННЫЕ МЕТОДЫ РЕШЕНИЯ ЛИНЕЙНЫХ СИСТЕМ

Учебное пособие
по курсу «Методы вычислительной математики»
для студентов направления подготовки
«Прикладная математика и информатика»

Москва
2019

УДК 519.61
ББК 22.193
В 387

*Утверждено учебным управлением МЭИ
в качестве учебного пособия*

Подготовлено на кафедре Математического моделирования

Рецензенты: доктор физ.-мат. наук, проф. А.А. Амосов,
доктор физ.-мат. наук, доц. А.В. Перескоков

Вестфальский, А.Е.

В 387 Введение в проекционные методы решения линейных систем: учебное пособие / А.Е. Вестфальский. — М.: МЭИ, 2019. — 48 с.

Пособие содержит основы общей теории построения современных проекционных методов решения несимметричных линейных систем, а также подробный вывод наиболее известных из них: обобщенного метода наименьших невязок (GMRES), метода бисопряженных градиентов (BSG) и их модификаций. Отдельно рассмотрен случай симметричных систем. Все методы сопровождаются алгоритмами на псевдокоде, удобными для программной реализации.

Для студентов, обучающихся по направлению подготовки «Прикладная математика и информатика».

Предисловие

Методы решения разреженных несимметричных линейных систем большого порядка являются актуальным, перспективным и интересным разделом современной вычислительной линейной алгебры. Однако в русскоязычной специализированной учебной литературе они представлены очень слабо. Полные описания методов сопряженных (и даже бисопряженных) направлений в отечественной литературе встречаются давно (например, [3, 8]), но они подаются с иных методических предпосылок. Переводные издания, претендующие на широту охвата (например, [5, 7]), часто ограничиваются обсуждением темы на уровне сравнения алгоритмов. Прекрасная монография [4], энциклопедичная для симметричных матриц, почти не затрагивает проекционную постановку задачи. Единственным глубоким описанием проекционного подхода на русском языке, по-видимому, является перевод первой части монографии [6]. К ней можно добавить пособие [2], в котором изложение ведется на более простом уровне и где опущены некоторые важные технические детали.

Данная работа имеет своей целью полное и подробное описание вывода двух наиболее известных проекционных методов (GMRES и BCG). По содержанию она примерно соответствует половине лекционного спецкурса, читаемого для избранных студентов направления Прикладная математика и информатика.

Для освоения материала требуется хорошее владение линейной алгеброй. Предполагается также знакомство с классическими итерационными процедурами решения СЛАУ (методами Якоби, Зейделя), а также с методами декомпозиции матриц (LU, QR разложения). При необходимости соответствующий материал можно найти в [1].

1. ОРТОГОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ И ЛИНЕЙНЫЕ СИСТЕМЫ

1.1. Постановка задачи

Рассмотрим систему линейных алгебраических уравнений (СЛАУ)

$$Ax = b.$$

Здесь A – квадратная невырожденная матрица порядка n , x – столбец неизвестных, b – столбец правых частей; $x, b \in \mathbb{R}^n$.

Как хорошо известно, невырожденность матрицы необходима и достаточна для однозначной разрешимости СЛАУ.

Пусть в \mathbb{R}^n заданы подпространства \mathcal{K} и \mathcal{L} . Через (x, y) будем обозначать стандартное скалярное произведение в \mathbb{R}^n . Рассмотрим следующую задачу: найти $x \in \mathbb{R}^n$ такой, что

$$(Ax, l) = (b, l) \quad \forall l \in \mathcal{L}. \quad (1)$$

Для произвольного $x \in \mathbb{R}^n$ величина $r = b - Ax$ называется невязкой. Перенося в (1) все в правую часть, получаем эквивалентное условие

$$(r, l) = 0 \quad \forall l \in \mathcal{L}.$$

Другими словами, требуется найти такой $x \in \mathcal{K}$, для которого невязка ортогональна подпространству \mathcal{L} . Эту задачу принято называть задачей проектирования на \mathcal{K} ортогонально \mathcal{L} . Важно отметить, что подпространства \mathcal{K} и \mathcal{L} могут как совпадать, так и не совпадать друг с другом.

1.2. Разрешимость задачи ортогонального проектирования

Обозначим точное решение СЛАУ через \bar{x} . Пусть известно начальное приближение к решению x_0 . Величина $\delta = \bar{x} - x_0$ называется поправкой. Поскольку

$$r = b - A(x_0 + \delta) = (b - Ax_0) - A\delta = r_0 - A\delta,$$

где $r_0 = b - Ax_0$, условие ортогональности можно переписать в виде

$$(r_0 - A\delta, l) = 0 \quad \forall l \in \mathcal{L}. \quad (2)$$

Рассмотрим подпространства \mathcal{K} и \mathcal{L} одинаковой размерности m . Пусть в \mathcal{K} задан произвольный базис v_1, v_2, \dots, v_m . Обозначим через V матрицу, столбцами которой являются эти базисные векторы. Пусть в

\mathcal{L} задан произвольный базис w_1, w_2, \dots, w_m . Через W обозначим матрицу, столбцами которой являются базисные векторы w_k . В общем случае матрицы V и W прямоугольные и имеют n строк и m столбцов каждая.

Ортогональность вектора подпространству \mathcal{L} эквивалентна ортогональности этого вектора каждому вектору из базиса W . Поэтому условие (2) переписывается в виде

$$(r_0 - A\delta, w_j) = 0, \quad j = \overline{1, m}.$$

Поскольку w_j есть j -й столбец матрицы W , последняя совокупность условий может быть записана в виде эквивалентного матрично-векторного соотношения

$$W^T(r_0 - A\delta) = 0.$$

Разложим искомую поправку $\delta \in \mathcal{K}$ по базису V :

$$\delta = y_1 v_1 + y_2 v_2 + \dots + y_m v_m = Vy.$$

Подставляя δ в предыдущее равенство, получаем новую СЛАУ для нахождения вектора неизвестных коэффициентов $y \in \mathbb{R}^m$

$$W^T AVy = W^T r_0. \quad (3)$$

Полученная система замкнута, она содержит m уравнений и m неизвестных, что является следствием равенства размерностей подпространств \mathcal{K} и \mathcal{L} .

Из соотношения (3) можно формально выразить y и получить уточнение решения исходной системы

$$x = x_0 + \delta = x_0 + Vy = x_0 + V(W^T AV)^{-1} W^T r_0.$$

Важно отметить, что если $m < n$, то полученный по последней формуле x не будет, вообще говоря, являться точным решением исходной системы.

Замечание. Система (3) определяет важные требования к выбору подпространств \mathcal{K} , \mathcal{L} и базисов в них.

- а) Матрица $W^T AV$ должна быть невырожденной и легко обратимой.
- б) Трудоемкость решения СЛАУ (3) должно быть существенно меньше, чем у исходной системы $Ax = b$.

Приведем классический **пример** [6], иллюстрирующий первое условие. Пусть матрица

$$A = \begin{pmatrix} O & I \\ I & I \end{pmatrix},$$

где O и I – нулевая и единичная матрицы порядка m . Выберем совпадающие базисы V и W , состоящие из m первых декартовых ортов. Тогда $W^T AV = O$.

Существует несколько простых достаточных условий, гарантирующих невырожденность матрицы метода.

Теорема 1.1. Пусть

- либо (1) $A > 0$ и $\mathcal{K} = \mathcal{L}$,
- либо (2) $\det A \neq 0$ и $\mathcal{L} = A\mathcal{K}$.

Тогда матрица $B = W^T AV$ невырожденная при любом выборе базисов V и W .

Доказательство.

1) Начнем со случая $\mathcal{K} = \mathcal{L}$. В этом случае V и W – два базиса в одном и том же пространстве. Следовательно, существует невырожденная матрица перехода G (порядка m) такая, что $W = VG$. Тогда

$$B = W^T AV = G^T V^T AV.$$

Из положительной определенности матрицы A следует положительная определенность матрицы $V^T AV$, которая влечет ее невырожденность.

Теперь невырожденность B следует из невырожденности двух составляющих ее множителей.

2) Если $\mathcal{L} = A\mathcal{K}$, то невырожденная матрица перехода G связывает базисы AV и W , то есть $W = AVG$. Следовательно,

$$B = W^T AV = G^T (AV)^T (AV).$$

Матрица AV имеет полный ранг (равный $m \leq n$) как произведение полноранговых матриц. Следовательно, квадратная матрица $(AV)^T (AV)$ также имеет полный ранг, что эквивалентно ее невырожденности.

Снова невырожденность B следует из невырожденности двух составляющих ее множителей. Теорема доказана.

Выписанные соотношения и наличие достаточных условий разрешимости позволяют записать общую схему метода.

Прототип алгоритма проекционного метода

1. Задать начальное приближение x
2. **While** не достигнута точность
3. Выбрать подпространства \mathcal{K} и \mathcal{L}
4. Построить в них базисы V и W
5. Вычислить невязку $r = b - Ax$
6. Решить СЛАУ $(W^T AV)y = W^T r$
7. Уточнить решение $x = x + Vy$
8. **EndWhile**

Приведем без доказательства два факта, связывающие проекционную постановку с задачей оптимизации.

Теорема 1.2. Если $A^T = A > 0$ и $\mathcal{L} = \mathcal{K}$, то задача ортогонального проектирования эквивалентна задаче минимизации функционала $\Phi(x) = \|x - \bar{x}\|_A^2$ на множестве $\{x \in \mathcal{K}\}$.

Теорема 1.3. Если матрица A невырожденная и $\mathcal{L} = A\mathcal{K}$, то задача проектирования на \mathcal{K} ортогонально \mathcal{L} эквивалентна задаче минимизации функционала $\Phi_r(x) = \|r_x\|_2^2 \equiv \|b - Ax\|_2^2$ на множестве $\{x \in \mathcal{K}\}$.

Дальнейшее изложение будет посвящено исследованию различных способов выбора подпространств \mathcal{K} и \mathcal{L} и удобных методов построения базисов в них.

2. ОДНОМЕРНЫЕ ПРОЕКЦИИ

2.1. Проектирование на одномерное подпространство

Построение методов начнем со случая, когда базисы подпространств \mathcal{K} и \mathcal{L} состоят из одного вектора, т.е.

$$\mathcal{K} = \text{span}\{v_k\}, \quad \mathcal{L} = \text{span}\{w_k\}.$$

Индекс k показывает, что на разных итерациях можно выбирать разные базисные векторы. Как правило, эти векторы связаны с текущим приближением к точному решению системы.

Пусть задано начальное приближение к решению x_0 и выполнено k шагов метода. Согласно общей теории (см. шаг (7) протоалгоритма), следующее уточнение находится по формуле

$$x_{k+1} = x_k + Vy.$$

В данном случае матрица V состоит из одного столбца v_k , следовательно y – скаляр. Обозначив его другой буквой γ , перепишем соотношение в виде

$$x_{k+1} = x_k + \gamma v_k.$$

Условие ортогональности подпространству \mathcal{L} имеет вид

$$(r_{k+1}, w_k) = 0.$$

Выполним преобразования

$$\begin{aligned} r_{k+1} &= b - Ax_{k+1} = b - A(x_k + \gamma v_k) = \\ &= (b - Ax_k) - \gamma Av_k = r_k - \gamma Av_k. \end{aligned}$$

Полученное соотношение представляет собой рекуррентную формулу пересчета невязки и позволяет, помимо прочего, экономить вычисления на каждой итерации.

Подставив выражение невязки в условие ортогональности

$$(r_{k+1}, w_k) = (r_k, w_k) - \gamma (Av_k, w_k) = 0,$$

получим формулу для коэффициента

$$\gamma_k = \frac{(r_k, w_k)}{(Av_k, w_k)}. \quad (4)$$

Дальше в качестве примеров рассмотрим конкретные способы выбора векторов v_k и w_k . Первые три метода интересны тем, как усложняя выбор базисов, уменьшаются требования к матрице A . Четвертый метод вписывает в картину проекционных методов хорошо известный классический алгоритм.

2.2. Метод наискорейшего спуска

В англоязычной литературе для данного метода принято использовать аббревиатуру SD (от англ. steepest descent).

Выберем в качестве базисного вектора как подпространства \mathcal{K} , так и подпространства \mathcal{L} вектор текущей невязки

$$v_k = w_k = r_k.$$

Подставляя v_k и w_k в формулу (4), получаем

$$\gamma_k = \frac{(r_k, r_k)}{(Ar_k, r_k)}.$$

Поскольку данный способ проектирования относится к случаю $\mathcal{K} = \mathcal{L}$, то в соответствии с теоремой 1.1 нужно требовать, чтобы матрица A была симметричной и положительно определенной. Заметим, что последнего свойства достаточно, чтобы знаменатель γ не обращался в ноль.

Подставляя все формулы в прототип алгоритма, получаем

Алгоритм SD

1. Задать начальное приближение x
2. **While** не достигнута точность
3. $\gamma := \frac{(r, r)}{(Ar, r)}$
4. $x := x + \gamma r$
5. $r := r - \gamma Ar$
6. **EndWhile**

В записи алгоритма не использован индекс k , поскольку метод является одношаговым и не требует хранения предыдущих приближений для своей работы. Необходимо также отметить, что введение дополнительной переменной $p = Ar$, вычисляемой перед шагом (3) и используемой в шагах (3) и (5) позволяет существенно уменьшить трудоемкость каждой итерации (для плотных матриц – почти вдвое).

Согласно теореме 1.2 каждая итерация метода SD соответствует процессу минимизации функции $\Phi(\gamma) = \|x_k + \gamma r_k - \bar{x}\|_A^2$.

Приведем информацию о скорости сходимости метода.

Теорема 2.1. Пусть $A^T = A > 0$. Тогда верна оценка

$$\|x_{k+1} - \bar{x}\|_A \leq \frac{\lambda_{\max} - \lambda_{\min}}{\lambda_{\max} + \lambda_{\min}} \|x_k - \bar{x}\|_A,$$

где λ_{\max} и λ_{\min} – максимальное и минимальное собственные числа матрицы A .

2.3. Метод минимальных невязок

В англоязычной литературе для данного метода принято использовать аббревиатуру MR (от англ. minimal residuals).

Выберем в качестве базисного вектора подпространства \mathcal{K} вектор текущей невязки и возьмем $\mathcal{L} = A\mathcal{K}$ так, что

$$v_k = r_k, \quad w_k = Av_k.$$

Подставляя v_k и w_k в формулу (4), получаем

$$\gamma_k = \frac{(r_k, Ar_k)}{(Ar_k, Ar_k)}.$$

Поскольку данный способ проектирования относится к случаю $\mathcal{K} = A\mathcal{L}$, то в соответствии с теоремой 1.1 нужно требовать только, чтобы матрица A была невырожденной. Заметим, что тогда знаменатель γ не обращается в ноль при лобом $r \neq \bar{0}$.

Подставляя все выведенные формулы в прототип алгоритма, получаем

Алгоритм MR

1. Задать начальное приближение x
2. **While** не достигнута точность
3. $\gamma := \frac{(r, Ar)}{(Ar, Ar)}$
4. $x := x + \gamma r$
5. $r := r - \gamma Ar$
6. **EndWhile**

Все сказанное об индексации и трудоемкости для предыдущего алгоритма остается верным.

Согласно теореме 1.3 каждая итерация метода MR соответствует процессу минимизации функции $\Phi_r(\gamma) = \|b - A(x_k + \gamma r_k)\|_2^2$.

Приведем (снова без доказательства) информацию о скорости сходимости метода.

Теорема 2.2. Пусть $A > 0$. Тогда верна оценка

$$\|r_{k+1}\|_2 \leq \sqrt{1 - \frac{\mu^2}{\sigma^2}} \|r_k\|_2,$$

в которой $\sigma = \|A\|_2$, а μ равно минимальному собственному числу матрицы $\frac{A^T + A}{2}$.

2.4. Метод наискорейшего уменьшения невязки

В отечественной литературе нет устоявшегося названия для данного метода. В англоязычной принято использовать аббревиатуру RnSD (от англ. residual norm steepest descent).

Выберем в качестве базисов в \mathcal{K} и $\mathcal{L} = A\mathcal{K}$ вектора

$$v_k = A^T r_k, \quad w_k = Av_k.$$

Подставляя v_k и w_k в формулу (4), после преобразований получаем

$$\gamma_k = \frac{(r_k, AA^T r_k)}{(AA^T r_k, AA^T r_k)} = \frac{(A^T r_k, A^T r_k)}{(A[A^T r_k], A[A^T r_k])} = \frac{(v_k, v_k)}{(Av_k, Av_k)}.$$

Поскольку данный способ проектирования относится к случаю $\mathcal{K} = A\mathcal{L}$, то в соответствии с теоремой 1.1 нужно требовать только, чтобы матрица A была невырожденной. Заметим, что тогда знаменатель γ не обращается в ноль при любом $v \neq \bar{0}$. Вектор v , в свою очередь, отличен от нуля если и только если $r \neq \bar{0}$.

В приведенном ниже алгоритме проведена оптимизация, описанная в предыдущих пунктах.

Алгоритм RnSD

1. Задать начальное приближение x
2. **While** не достигнута точность
3. $v := A^T r$
4. $p := Av$
5. $\gamma := \frac{(v, v)}{(p, p)}$
6. $x := x + \gamma v$
7. $r := r - \gamma p$
8. **End While**

В качестве плюса метода RnSD по сравнению с SD и MR следует отметить отсутствие каких-либо условий на матрицу системы A (кроме естественного условия невырожденности).

Обратной стороной медали является необходимость работы с транспонированной матрицей A^T , что не всегда бывает удобным и даже возможным. Каждая итерация метода RnSD требует двух матрично-векторных произведений (с разными матрицами), в то время как в предыдущих методах достаточно одной такой операции.

Если дополнительно преобразовать выражение для параметра

$$\gamma = \frac{(A^T r_k, A^T r_k)}{(AA^T r_k, AA^T r_k)} = \frac{(A^T r_k, A^T r_k)}{([A^T A] A^T r_k, A^T r_k)},$$

то можно заметить, что оно соответствует формулам метода SD, примененного для решения СЛАУ

$$A^T Ax = A^T b.$$

Отсюда несложно получить оценку скорости сходимости метода.

2.5. Декартовы проекции

Вернемся к случаю совпадающих подпространств $\mathcal{K} = \mathcal{L} \subset \mathbb{R}^n$ с одним и тем же базисом в них. В качестве базисного вектора $v_k = w_k$ будем использовать декартовы орты e_k (в которых на k -й позиции стоит единица, остальное – нули). После орта e_n будем брать e_1 и так далее.

В такой ситуации

$$x_{k+1} = x_k + \gamma_k e_k,$$

следовательно, на каждом шаге будет изменяться только один элемент приближенного решения.

Из формулы (4) получаем

$$\gamma_k = \frac{(r_k, e_k)}{(Ae_k, e_k)}.$$

Знаменатель равен диагональному элементу матрицы $A_{k,k}$. Рассмотрим числитель

$$\begin{aligned} (r_k, e_k) &= (b - Ax_k, e_k) = (b, e_k) - (Ax_k, e_k) = \\ &= b_k - [Ax_k]_k = b_k - \sum_{j=1}^n A_{k,j} x_j^{(k)}. \end{aligned}$$

В последнем выражении номер шага (итерации) перенесен в верхний индекс, чтобы использовать нижний индекс для номера элемента вектора (такие обозначения будут использованы только в данном пункте). Используя индексы таким же образом, запишем полученные расчетные формулы

$$x^{(k+1)} = \left[b_k - \sum_{j=1}^n A_{k,j} x_j^{(k)} \right] \frac{e^{(k)}}{A_{k,k}}.$$

Если теперь назвать одной итерацией перебор всех ортов (от e_1 до e_n), то полученные формулы будут в точности описывать хорошо известный метод Зейделя.

3. ОРТОГОНАЛИЗАЦИЯ В ПОДПРОСТРАНСТВАХ КРЫЛОВА

3.1. Подпространства Крылова. Ортогонализация Арнольди

Проекционные методы, использующие одномерные подпространства, имеют существенный недостаток, заключающийся в том, что на каждом следующем шаге все приходится делать заново. Информация из выбранного одномерного подпространства позволяет сделать шаг в сторону точного решения, но затем она безвозвратно теряется. Это может приводить к большому количеству итераций (существенно превышающему порядок матрицы n).

Два подхода, рассматриваемые ниже, лишены указанного недостатка. В них подпространство \mathcal{K} (и, следовательно, \mathcal{L}) плавно расширяется так, что $\mathcal{K}_k \subset \mathcal{K}_{k+1} \forall k \geq 1$. Отсюда, в частности, следует, что в точной арифметике метод сойдется не более, чем за n итераций.

Очень удобными являются подпространства Крылова (названные в честь выдающегося русского советского механика и математика Крылова А.Н.), в построении которых участвует матрица решаемой системы A .

Определение. Линейная оболочка

$$\text{span}\{v, Av, A^2v, \dots, A^{m-1}v\} \subset \mathbb{R}^n$$

называется подпространством Крылова и обозначается $\mathcal{K}_m(v, A)$.

Важно отметить, что размерность подпространства Крылова может быть меньше m . Фактически, эта размерность зависит от расположения начального вектора v относительно инвариантных подпространств матрицы A . Чтобы представить такую картину на наиболее простом примере, рекомендуется рассмотреть случай, когда v является собственным вектором матрицы A .

Для построения ортонормированного базиса в $\mathcal{K}_m(v, A)$ можно использовать известную процедуру ортогонализации Грама-Шмидта. Пусть некоторое количество j взаимно ортогональных и нормированных векторов уже построено. Обозначим их v_1, v_2, \dots, v_j .

Умножим последний полученный вектор v_j на матрицу A и будем искать следующий вектор \tilde{v}_{j+1} в виде

$$\tilde{v}_{j+1} = Av_j - \sum_{i=1}^j h_{i,j} v_i. \quad (5)$$

Коэффициенты $h_{1,j}, h_{2,j}, \dots, h_{j,j}$ подберем так, чтобы новый вектор \tilde{v}_{j+1} был ортогонален всем предыдущим векторам v_1, v_2, \dots, v_j . Обозначив $w = Av_j$, запишем эти условия ортогональности

$$(\tilde{v}_{j+1}, v_k) = (w - \sum_{i=1}^j h_{i,j} v_i, v_k) = 0, \quad k = \overline{1, j}.$$

Поскольку по предположению $(v_i, v_j) = \delta_{i,j}$ (где $\delta_{i,j}$ – символ Кронекера), то

$$\left(\sum_{i=1}^j h_{i,j} v_i, v_k \right) = (h_{k,j} v_k, v_k) = h_{k,j}.$$

Отсюда получаем формулы для коэффициентов ортогонализации

$$h_{i,j} = (w, v_i), \quad i = \overline{1, j}.$$

После этого остается отнормировать полученный вектор

$$v_{j+1} = \frac{\tilde{v}_{j+1}}{\|\tilde{v}_{j+1}\|_2}.$$

Если на некотором шаге j после всех преобразований норма вектора \tilde{v}_{j+1} оказывается равной нулю, это означает, что очередной вектор $A^j v$ линейно выражается через предыдущие. Несложно показать, что в этом случае все вектора вида $A^k v$ при $k \geq j$ также будут линейно зависеть от предыдущих. Следовательно процесс построения ортонормированного базиса будет исчерпан и размерность соответствующего подпространства Крылова будет равна j .

Описанная процедура получила название ортогонализации Арнольди. Запишем ее в виде алгоритма.

Алгоритм ортогонализации Арнольди

1. Выбрать v_1 , такой что $\|v_1\|_2 = 1$
2. **For** $j = 1, 2, \dots, n$
3. $w := Av_j$
4. **For** $i = 1, \dots, j$
5. $h_{i,j} := (w, v_i)$
6. $w := w - h_{i,j} v_i$
7. **EndFor**
8. $h_{j+1,j} := \|w\|_2$
9. **If** $h_{j+1,j} = 0$ **then** положить $m := j$ и выйти из цикла **EndIf**
10. $v_{j+1} := w/h_{j+1,j}$
11. **EndFor**

Ситуация, связанная с преждевременным исчерпанием базиса, отслеживается в строке (9) алгоритма. Если же выход из цикла по условию строки (9) не произведен, то будет построен базис из n векторов, и подпространство Крылова будет совпадать со всем пространством \mathbb{R}^n .

Пусть выполнено m шагов рассматриваемого алгоритма. Введем в рассмотрение матрицу V_{m+1} , столбцами которой являются элементы построенного ортонормированного базиса v_1, v_2, \dots, v_{m+1} (возможно, $v_{m+1} = \bar{0}$), а также матрицу \bar{H}_m , составленную из коэффициентов ортогонализации

$$V_{m+1} = \begin{array}{|c|c|c|c|} \hline & & & \\ \hline v_1 & v_2 & \cdots & v_{m+1} \\ \hline \end{array}, \quad \bar{H}_m = \begin{array}{|cccc|} \hline h_{1,1} & h_{1,2} & \cdots & h_{1,m} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,m} \\ 0 & h_{3,2} & \cdots & h_{3,m} \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & h_{m+1,m} \\ \hline \end{array}.$$

Ясно, что матрица V состоит из n строк и $m+1$ столбца, а матрица \bar{H}_m имеет $m+1$ строк и m столбцов, причем все элементы \bar{H}_m , лежащие ниже первой поддиагонали, равны нулю. Матрицы такой структуры принято называть хессенберговыми.

Таким образом, после выполнения описанного алгоритма ортогонализации Арнольди будут получены

во-первых, набор ортонормированных векторов v_i (которые обычно хранят как столбцы объединяющей их матрицы V);

во-вторых, матрица коэффициентов ортогонализации $\bar{H}_m = \{h_{i,j}\}$, у которой будут заполнены только верхний треугольник и первая поддиагональ;

в-третьих, размерность m крыловского подпространства $\mathcal{K}_m(v, A)$.

Зафиксируем несколько свойств построенных матриц.

Утверждение 3.1. Справедливо равенство

$$AV_m = V_{m+1}\bar{H}_m.$$

Доказательство.

Используя формулу (5) и учитывая, что $h_{j+1,j} = \|\tilde{v}_{j+1}\|_2$, имеем

$$v_{j+1} = \frac{\tilde{v}_{j+1}}{\|\tilde{v}_{j+1}\|_2} = \left(Av_j - \sum_{i=1}^j h_{i,j} v_i \right) / h_{j+1,j},$$

откуда

$$Av_j = \sum_{i=1}^j h_{i,j}v_i + h_{j+1,j}v_{j+1} = \sum_{i=1}^{j+1} h_{i,j}v_i. \quad (6)$$

Заметим, что выражение

$$\sum_{i=1}^{j+1} h_{i,j}v_i$$

описывает j столбец матрицы, являющейся произведением $V_{m+1} \cdot \bar{H}_m$. Следовательно, соотношение (6), справедливое для любого j от 1 до m , есть постолбцовое выражение искомого матричного равенства.

Свойство доказано.

Обозначим через H_m квадратную матрицу, получающуюся отбрасыванием последней строки в матрице \bar{H}_m

Утверждение 3.2. Справедливо равенство

$$AV_m = V_m H_m + h_{m+1,m}v_{m+1}e_m^T.$$

Доказательство.

Воспользуемся равенством

$$AV_m = V_{m+1}\bar{H}_m$$

и разобьем в нем матрицу V_{m+1} на два слагаемых

$$V_{m+1} = V_{m+1}^0 + \tilde{V}_{m+1}$$

следующим образом.

Первые m столбцов попадают в V_{m+1}^0 и дополняются нулевым столбцом. В матрице \tilde{V}_{m+1} , наоборот, все столбцы, кроме последнего, нулевые, а последний совпадает с v_{m+1} .

$$\begin{array}{|c|c|c|c|} \hline v_1 & \cdots & v_m & v_{m+1} \\ \hline \end{array} = \begin{array}{|c|c|c|c|} \hline v_1 & \cdots & v_m & \bar{0} \\ \hline \end{array} + \begin{array}{|c|c|c|c|} \hline \bar{0} & \cdots & \bar{0} & v_{m+1} \\ \hline \end{array}$$

Очевидно, что

$$\begin{array}{|c|c|c|c|} \hline v_1 & \cdots & v_m & \bar{0} \\ \hline \end{array} \cdot \begin{array}{|c|c|c|c|} \hline h_{1,1} & h_{1,2} & \cdots & h_{1,m} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,m} \\ 0 & h_{3,2} & \cdots & h_{3,m} \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & h_{m,m} \\ \hline 0 & 0 & \cdots & h_{m+1,m} \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline v_1 & \cdots & v_m \\ \hline \end{array} \cdot \begin{array}{|c|c|c|c|} \hline h_{1,1} & h_{1,2} & \cdots & h_{1,m} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,m} \\ 0 & h_{3,2} & \cdots & h_{3,m} \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & h_{m,m} \\ \hline \end{array},$$

Данное равенство соответствует матричной записи

$$V_{m+1}^0 \bar{H}_m = V_m H_m.$$

Второе слагаемое – произведение $\tilde{V}_{m+1} \bar{H}_m$ – представляет собой одноранговую матрицу, имеющую единственный ненулевой столбец (последний). Ее можно представить как произведение вектора-столбца на вектор-строку e_m^T , получаемую транспонированием декартова орта с единицей в последней позиции ($m+1$). Сказанное иллюстрируется схемой

$$\begin{array}{|c|c|c|c|c|} \hline \bar{0} & \cdots & \bar{0} & \bar{0} & v_{m+1} \\ \hline \end{array} \cdot \begin{array}{|c|c|c|c|} \hline h_{1,1} & h_{1,2} & \cdots & h_{1,m} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,m} \\ 0 & h_{3,2} & \cdots & h_{3,m} \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & h_{m,m} \\ \hline 0 & 0 & \cdots & h_{m+1,m} \\ \hline \end{array} = \begin{array}{|c|c|c|c|} \hline \bar{0} & \cdots & \bar{0} & h_{m+1,m} v_{m+1} \\ \hline \end{array} = \begin{array}{|c|} \hline h_{m+1,m} v_{m+1} \\ \hline \end{array} \cdot \begin{array}{|c|c|c|c|c|} \hline 0 & 0 & \cdots & 0 & 1 \\ \hline \end{array}$$

и соответствует матричной записи

$$\tilde{V}_{m+1}\bar{H}_m = h_{m+1,m}v_{m+1}e_m^T.$$

Складывая оба равенства, получаем доказываемое утверждение.

Утверждение 3.3. Справедливо равенство

$$V_m^T AV_m = H_m.$$

Доказательство.

Согласно предыдущему утверждению,

$$V_m^T AV_m = V_m^T V_{m+1} \bar{H}_m$$

Рассмотрим произведение $V_m^T V_{m+1}$ и воспользуемся ортонормированностью базиса v_1, \dots, v_{m+1}

$$V_m^T V_{m+1} = \begin{array}{|c|} \hline v_1 \\ \hline v_2 \\ \hline \vdots \\ \hline v_m \\ \hline \end{array} \cdot \begin{array}{|c|c|c|c|} \hline & & & \\ \hline v_1 & v_2 & \cdots & v_{m+1} \\ \hline \end{array} = \begin{array}{|cccc|} \hline 1 & 0 \cdots 0 & 0 & 0 \\ \hline 0 & 1 \cdots 0 & 0 & 0 \\ \hline \vdots & \vdots & \vdots & \vdots \\ \hline 0 & 0 \cdots 1 & 0 & 0 \\ \hline \end{array}.$$

Полученная матрица представляет собой единичную матрицу порядка m с приписанным к ней справа нулевым столбцом. Умножая эту матрицу справа на \bar{H}_m , получаем H_m , что и требовалось.

Утверждение 3.4. Для любого номера k от 1 до $m+1$ справедливо равенство

$$V_m^T v_k = e_k,$$

где e_k – декартов орт с единицей в позиции k .

Доказательство следует непосредственно из ортонормированности столбцов матрицы V_m . Для наглядности можно воспользоваться иллюстрацией к предыдущему утверждению.

Процесс ортогонализации Арнольди лежит в основе многих проекционных методов решения СЛАУ. Его несложно увидеть в качестве первого этапа описанных далее методов.

3.2. Метод полной ортогонализации

В англоязычной литературе для данного метода принято использовать аббревиатуру FOM (от англ. full orthogonalization method).

Выберем в качестве подпространств \mathcal{K} и \mathcal{L} подпространство Крылова

$$\mathcal{K} = \mathcal{L} = \mathcal{K}_m(v_r, A),$$

для построения которого используется орт начальной невязки

$$v_1 = r_0 / \|r_0\|_2.$$

Для построения базиса в \mathcal{K} (и также в \mathcal{L}) будем использовать ортогонализацию Арнольди с начальным вектором v_1 .

В соответствии с общим проекционным подходом (см. пп. 7 и 8 протоалгоритма) решение должно уточняться по формуле

$$x = x_0 + Vy,$$

где y является решением системы

$$(W^T AV)y = W^T r_0,$$

в которой V и W – матрицы, составленные из базисных векторов подпространств \mathcal{K} и \mathcal{L} соответственно.

В нашем случае эти матрицы равны

$$V = W = V_m.$$

Следовательно, (см. утверждение 3.3)

$$W^T AV = V_m^T AV_m = H_m$$

Обозначим для краткости $\beta = \|r_0\|_2$. По утверждению 3.4

$$W^T r_0 = V_m^T r_0 = V_m^T (\beta v_1) = \beta e_1,$$

где e_1 – единичный декартов орт, первая координата которого равна единице, остальные – нулю.

Таким образом, вспомогательная СЛАУ принимает вид

$$H_m y = \beta e_1.$$

Отметим, что оба основных требования выполнены: матрица H_m невырождена, а также легко обратима, поскольку имеет хессенбергову форму. Для решения системы с такой матрицей требуется занулить одну поддиагональ. Это можно сделать гауссовыми исключениями, либо ортогональными вращениями Гивенса.

Подведем итоги на псевдокоде.

Алгоритм FOM

1. $r_0 := b - Ax_0$, $\beta := \|r_0\|_2$, $v_1 := r_0/\beta$, $m = n$
2. **For** $j = 1, 2, \dots, n$
3. Вычислить $w_j := Av_j$
4. **For** $i = 1, \dots, j$
5. $h_{i,j} := (w_j, v_i)$
6. $w_j := w_j - h_{i,j}v_i$
7. **EndFor**
8. Вычислить $h_{j+1,j} := \|w_j\|_2$
9. **If** $h_{j+1,j} = 0$ **then** положить $m := j$ и выйти из цикла **EndIf**
10. Вычислить $v_{j+1} := w_j/h_{j+1,j}$
11. **EndFor**
12. Преобразовать матрицу H_m , исключив из нее поддиагональ.
 Выполнить те же преобразования с вектором $g = \beta e_1$
13. Решить треугольную СЛАУ $Hy = g$ (порядка m)
14. Вычислить $x = x_0 + \sum_{i=1}^m y_i v_i$

Замечание. Размерность пространства \mathcal{K} зависит от выбора начального вектора v_1 и может оказаться меньше порядка матрицы n . В этом случае процесс ортогонализации нет необходимости продолжать, поскольку он будет обнулять все последующие векторы.

Таким образом, после завершения полного процесса ортогонализации возможны два случая.

Либо подпространство $\mathcal{K}_m(v_r, A)$ совпадет со всем пространством \mathbb{R}^n (получим $m = n$).

Либо произойдет исчерпание базиса и процесс завершится досрочно при $m < n$.

Важно, что в любом из этих случаев будет получен $v_{m+1} = \bar{0}$ и $h_{m+1,m} = 0$ (в точной арифметике).

Утверждение 3.5. На каждом шаге m метода FOM для невязки r_m справедливо свойство

$$\|r_m\|_2 = h_{m+1,m}|y_m|$$

(здесь y_m означает последний, m -й, элемент вектора y).

Доказательство.

Согласно утверждению 3.2

$$AV_m = V_m H_m + h_{m+1,m} v_{m+1} e_m^T$$

Выполним преобразования

$$\begin{aligned}
r_m &= b - A(x_0 + V_my) = (b - Ax_0) - AV_my = \\
&= r_0 - AV_my = \beta e_1 - (V_m H_m + h_{m+1,m} v_{m+1} e_m^T) y = \\
&= \beta v_1 - V_m(H_my) - h_{m+1,m} v_{m+1} (e_m^T y)
\end{aligned}$$

Так как вектор y является решением системы $H_my = \beta e_1$, то

$$\beta v_1 - V_m(H_my) = \beta v_1 - V_m \beta e_1 = \beta v_1 - \beta v_1 = \bar{0}.$$

Остается заметить, что произведение $e_m^T y$ есть обычное скалярное произведение и равно y_m , а $\|v_{m+1}\|_2 = 1$. Утверждение доказано.

Важным недостатком метода FOM является необходимость хранения всех векторов $\{v_j\}$, получаемых при выполнении итераций основного цикла. Для их хранения может потребоваться слишком большой (возможно, недоступно большой) объем памяти.

Следующие два подхода частично устраняют этот недостаток.

3.3. Алгоритм полной ортогонализации с рестартами

Один из способов уменьшения затрат памяти в алгоритме FOM состоит в том, чтобы периодически обновлять алгоритм, а именно, цикл в строке (2) выполнять не до $j = n$ (порядка матрицы A), а останавливать раньше при некотором заданном значении m (как правило, небольшом). Поскольку в такой ситуации решение системы, скорее всего, найдено не будет, то процесс нужно повторять, используя в качестве нового начального приближения вектор x_m , полученный после неполного выполнения цикла (2). Указанный подход принято называть методом FOM с рестартами или FOM(m).

Алгоритм FOM(m)

1. Вычислить $r_0 := b - Ax_0$, $\beta := \|r_0\|_2$, $v_1 := r_0/\beta$
2. **For** $j = 1, 2, \dots, m$
 выполнить действия (3) – (10) алгоритма FOM
EndFor
3. Преобразовать матрицу H_m , исключив из нее поддиагональ.
 Выполнить те же преобразования с вектором $g = \beta e_1$
4. Решить треугольную СЛАУ $H_y = g$ порядка m
5. **If** качество приближения удовлетворительное **then** выход
6. Положить $x_0 = x_m$, вернуться в шаг 1.

3.4. Алгоритм неполной ортогонализации

Другим вариантом экономии ресурсов памяти в алгоритме FOM является проведение неполной ортогонализации. Идея заключается в том, что очередной вектор v_{j+1} (получаемый на j -ом шаге основного цикла) делается ортогональным только к k предыдущим векторам, а не ко всем.

Такой подход принято называть методом ИОМ (от англ. incomplete orthogonalization method).

Для получения алгоритма метода достаточно заменить строку (4) алгоритма FOM на следующую

4'. **For** $i = \max\{1, j - k + 1\}, \dots, j$

Подобно методу FOM(m), алгоритм ИОМ также можно дополнить рестартами, ограничивая m и повторяя основные действия циклически до выполнения критерия окончания.

3.5. Обобщенный метод минимальных невязок

В литературе данный метод широко известен под аббревиатурой GMRES (от англ. generalized minimal residuals).

Выберем в качестве подпространства \mathcal{K} подпространство Крылова

$$\mathcal{K} = \mathcal{K}_m(v_r, A),$$

для построения которого используется орт начальной невязки

$$v_1 = r_0 / \|r_0\|_2.$$

Для построения базиса в \mathcal{K} будем использовать ортогонализацию Арнольди с начальным вектором v_1 .

В отличие от предыдущего метода, в качестве подпространства \mathcal{L} выберем

$$\mathcal{L} = A\mathcal{K}.$$

Для вывода расчетных формул вместо явного использования условий ортогональности привлечем результат об оптимальном свойстве метода. По соответствующей теореме проекционный процесс при указанном выборе подпространств соответствует процессу минимизации 2-нормы невязки $\|r\|_2$.

Обозначим, как и раньше, $\beta = \|r_0\|_2$. Тогда начальный вектор базиса $v_1 = r_0 / \beta$.

Преобразуем выражение невязки

$$\begin{aligned}
r &= b - Ax = b - A(x_0 + V_m y) = \\
&= (b - Ax_0) - AV_m y = \\
&= r_0 - V_{m+1} \bar{H}_m y = \beta v_1 - V_{m+1} \bar{H}_m y = \\
&= \beta V_{m+1} e_1 - V_{m+1} \bar{H}_m y = V_{m+1} (\beta e_1 - \bar{H}_m y).
\end{aligned}$$

Здесь, как и выше, использовано обозначение e_1 для первого декартового орта.

Пользуясь тем, что ортогональные матрицы сохраняют 2-норму, получаем

$$\|r\|_2 = \|V_{m+1}(\beta e_1 - \bar{H}_m y)\|_2 = \|\beta e_1 - \bar{H}_m y\|_2.$$

Таким образом, задача минимизации величины $\|r\|_2$ эквивалентна минимизации величины

$$\|\beta e_1 - \bar{H}_m y\|_2,$$

что, в свою очередь, равносильно задаче решения переопределенной СЛАУ

$$\bar{H}_m y = \beta e_1$$

в смысле наименьших квадратов.

Применим для этого QR разложение. Разлагая

$$\bar{H}_m = QR$$

(где Q – матрица размера $(m+1) \times m$, R – верхнетреугольная матрица $m \times m$) и выполняя одновременно преобразование правой части, приходим к системе

$$Ry = \tilde{g}.$$

Выше, в методе FOM, процесс решения СЛАУ $Hu = \beta e_1$ производился строго после построения полного базиса Арнольди. Однако его можно совместить с процессом построения этого базиса. В этом случае получится полностью итерационная процедура, которую можно прерывать досрочно, если невязка очередного приближения окажется достаточно малой.

Для этого исключение (зануление) элементов первой поддиагонали матрицы H следует производить по мере их появления, т.е. на первом шаге внешнего цикла занулить элемент $h_{2,1}$, на втором шаге – элемент $h_{3,2}$ и так далее. Ниже этот процесс будет подробно пояснен матричными схемами.

В результате возникает алгоритм метода. Приведем его на псевдокоде.

Алгоритм GMRES

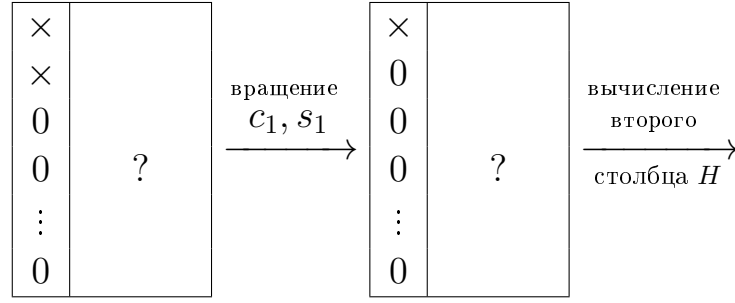
1. $r_0 := b - Ax_0$, $\beta := \|r_0\|_2$, $g := \beta e_1$, $v_1 := r_0/\beta$
2. **For** $j = 1, \dots, n$
3. $w := Av_j$
4. **For** $i = 1, \dots, j$
5. $h_{ij} := (w, v_i)$
6. $w := w - h_{ij}v_i$
7. **EndFor**
8. $h_{j+1j} := \|w\|_2$
9. **If** $h_{j+1j} = 0$ **then** $p := j$; выход из цикла **EndIf**
10. $v_{j+1} := w/h_{j+1j}$
11. **For** $i = 1, \dots, j - 1$
12. Применить к вектору $(h_{1j}, \dots, h_{j+1j})^T$
 плоское вращение $\{c_i, s_i\}$
13. **EndFor**
14. Вычислить параметры нового плоского вращения $\{c_j, s_j\}$:

$$\begin{pmatrix} h_{j,j} \\ h_{j+1,j} \end{pmatrix} \xrightarrow{\{c_j, s_j\}} \begin{pmatrix} \tilde{h}_{j,j} \\ 0 \end{pmatrix}$$

и применить его
15. Применить к вектору g плоское вращение $\{c_j, s_j\}$
16. **If** $|g_{j+1}| < \varepsilon$ **then** $p := j$; выход из цикла **EndIf**
17. **EndFor**
18. Решить треугольную СЛАУ $Hu = g$ порядка p
19. $x = x_0 + \sum_{i=1}^p y_i v_i$

Поясним инкорпорирование плоских вращений в процедуру ортогонализации следующей схемой. В ней крестиками (\times , $\bar{\times}$, $\tilde{\times}$) отмечены ненулевые элементы матрицы H , знаком вопроса обозначена не вычисленная к рассматриваемому моменту часть H .

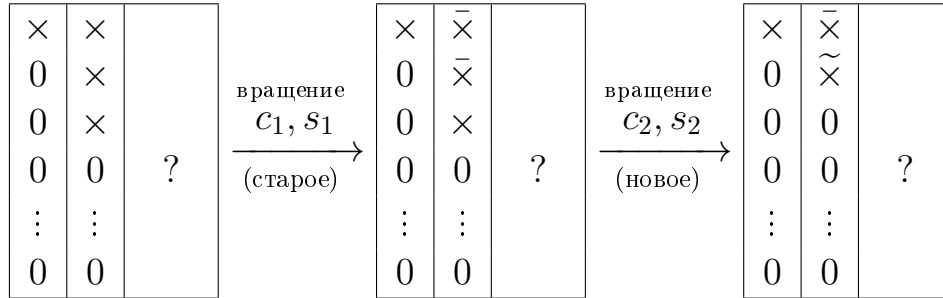
На первом шаге цикла из строки (2) при $j = 1$ формируется первый столбец матрицы H , состоящий из двух ненулевых элементов. По этим элементам вычисляются параметры первого плоского вращения c_1, s_1 (которые следует запомнить) и производится это вращение. В результате второй элемент первого столбца матрицы H зануляется. Отметим, что при $j = 1$ цикл в строках (11) – (13) не выполняется.



Далее, при $j = 2$ (второй проход цикла) формируется второй столбец матрицы H , состоящий из трех ненулевых элементов.

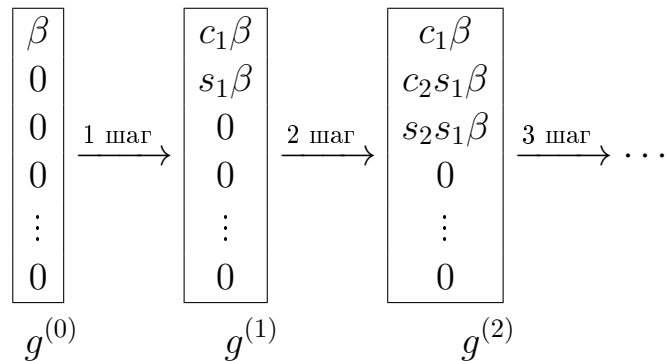
Сначала к полученному новому столбцу применяется сохраненное ранее вращение c_1, s_1 . Это происходит в цикле в строках (11) – (13).

Затем по элементам $\bar{h}_{2,2}$ и $h_{3,2}$ вычисляются параметры второго плоского вращения c_2, s_2 (которые также следует запомнить) и производится это вращение. В результате третий (поддиагональный) элемент второго столбца матрицы H зануляется.



На очередном проходе цикла (при $j = 3$) будет сформирован третий столбец H из четырех ненулевых элементов. Сначала к нему применяются сохраненные вращения c_1, s_1 и c_2, s_2 , затем будет сформировано и применено новое плоское вращение c_3, s_3 и так далее.

Приведенные выше схемы описывают изменения матрицы H . Те же самые действия (вращения) одновременно производятся над вектором правых частей $g = \beta e_1$. Изобразим их схематически.



Таким образом, для последнего ненулевого элемента $g_{j+1}^{(j)}$ верна формула

$$g_{j+1}^{(j)} = s_j s_{j-1} \dots s_1 \beta,$$

которая гарантирует невозрастание g_{j+1}^j (так как $|s_k| \leq 1 \quad \forall k$). Этим обосновывается критерий останова в строке (16) алгоритма.

Для полноты изложения приведем без доказательства информацию о сходимости метода в одном частном случае.

Пусть матрица A – диагонализируемая, то есть существует невырожденная матрица B такая, что

$$B^{-1}AB = \Lambda = \text{diag}\{\lambda_1, \dots, \lambda_n\}.$$

Обозначим через P_k множество всех многочленов степени k и введем

$$\varepsilon_m = \min_{p \in P, p(0)=1} \max_{1 \leq i \leq n} |p(\lambda_i)|.$$

Теорема 3.1. Для невязки m -го шага метода GMRES справедлива оценка

$$\|r_m\|_2 \leq \text{cond}_2(B) \varepsilon_m \|r_0\|_2.$$

Несмотря на то, что сформулированная теорема обосновывает убывание невязки, использовать ее для построения критерия окончания метода не представляется возможным.

4. СИММЕТРИЧНАЯ ОРТОГОНАЛИЗАЦИЯ

4.1. Ортогонализация Ланцоша

Рассмотрим построенные алгоритмы при дополнительном предположении симметричности матрицы A порядка n .

Полная ортогонализация Арнольди приводит к матричному разложению

$$AV_n = V_{n+1}\bar{H}_n = V_n H_n.$$

Второе равенство объясняется тем, что последний столбец матрицы V_{n+1} — нулевой, поскольку n совпадает с размерностью всего пространства.

Отсюда получаем, что $H_n = V_n^T AV_n$, следовательно

$$H_n^T = (V_n^T AV_n)^T = V_n^T A^T V_n = V_n^T AV_n = H_n.$$

Поскольку матрица H_n имеет хессенбергову форму и симметрична, она должна быть трехдиагональной. Обозначим для удобства $\alpha_i = h_{i,i}$, $\beta_i = h_{i-1,i}$, $T_n = H_n$. Тогда

$$T_n = H_n = \begin{bmatrix} \alpha_1 & \beta_2 & 0 & \cdots & 0 \\ \beta_2 & \alpha_2 & \beta_3 & \cdots & 0 \\ 0 & \beta_3 & \alpha_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & \alpha_n \end{bmatrix}$$

По аналогии с матрицами \bar{H}_m и H_m можно ввести матрицу \bar{T}_m (для произвольного $0 < m < n$), имеющую $(m+1)$ строк и m столбцов, и квадратную $m \times m$ матрицу T_m , которые стоят в левом верхнем углу матрицы T_n .

Общая формула процедуры ортогонализации крыловского базиса

$$w_j = h_{j+1,j}v_{j+1} = Av_j - \sum_{i=1}^j h_{i,j}v_i$$

в новых обозначениях примет вид

$$w_j = \beta_{j+1}v_{j+1} = Av_j - \alpha_j v_j - \beta_j v_{j-1},$$

где коэффициенты вычисляются по формулам

$$\alpha_j = (w_j, v_j), \quad \beta_j = \|w_j\|_2.$$

Полученную процедуру принято связывать с именем Ланцоша. Запишем ее в виде алгоритма.

Симметричный алгоритм ортогонализации Ланцоша

1. Выбрать v_1 , такой что $\|v_1\|_2 = 1$
2. Положить (формально) $v_0 = 0$, $\beta_1 = 0$
3. **For** $j = 1, 2, \dots, n$
4. $w := Av_j - \beta_j v_{j-1}$
5. $\alpha_j := (w, v_j)$
6. $w := w - \alpha_j v_j$
7. $\beta_{j+1} := \|w\|_2$
8. **If** $\beta_{j+1} = 0$ **then** положить $m := j$ и выйти из цикла **EndIf**
9. $v_{j+1} := w/\beta_{j+1}$
10. **EndFor**

Важным отличием симметричной ортогонализации Ланцоша от ортогонализации Арнольди является трехчленность соотношения между базисными векторами на каждом шаге. Поэтому в симметричном случае необходимо хранить только два предыдущих вектора базиса (в несимметричном нужны все предыдущие). Это приводит к существенной экономии ресурсов памяти.

Полученный алгоритм позволяет построить метод решения симметричных систем, аналогичный методу ФОМ. Поскольку дальнейшие действия будут направлены на доработку этого алгоритма, запишем его без детализации.

Симметричный метод Ланцоша

1. Вычислить $r_0 = b - Ax_0$, $\beta = \|r_0\|_2$, $v_1 = r_0/\beta$.
2. Построить ортогональный базис Ланцоша в $\mathcal{K}(v_1, A)$.
3. Решить трехдиагональную СЛАУ $T_m y = \beta e_1$.
4. Вычислить $x = x_0 + V_m y$.

Второе преимущество, которое дает симметрия, заключается в структуре матрицы T_m , а именно, в ее трехдиагональности. По этой причине решение системы в строке (3) осуществляется более экономично, чем в методе ФОМ, как по затратам памяти, так и по количеству действий.

4.2. Прямой метод Ланцоша

Дальнейшие действия будут направлены на объединение процесса решения трехдиагональной системы с процессом построения базиса, в результате чего должен получиться полностью итерационный алгоритм. Для удобства разобьем эти действия на несколько шагов.

1. Процесс решения трехдиагональной системы можно объединить с процессом построения базиса. Для этого воспользуемся LU разложением матрицы T_m и представим ее в виде

$$T_m = L_m U_m.$$

Из трехдиагональности T_m следует, что матрица L_m будет нижне-двухдиагональной, а U_m – верхне-двухдиагональной. Введем обозначения для их элементов

$$L_m = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ \mu_2 & 1 & 0 & \cdots & 0 \\ 0 & \mu_3 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}, \quad U_m = \begin{bmatrix} \eta_1 & \beta_2 & 0 & \cdots & 0 \\ 0 & \eta_2 & \beta_3 & \cdots & 0 \\ 0 & 0 & \eta_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & \eta_n \end{bmatrix}$$

Очевидно, что элементы наддиагонали β_k не будут изменяться при построении LU разложения, поэтому достаточно выразить элементы μ_k и η_k .

В соответствии с общим алгоритмом гауссова исключения (без выбора главного элемента)

$$\begin{aligned} \eta_1 &= \alpha_1, \\ \mu_k &= \beta_k / \eta_{k-1}, \quad \eta_k = \alpha_k - \mu_k \beta_k, \quad k = 2, 3, \dots, n \end{aligned}$$

2. Заметим, что формулу для нахождения решения можно переписать в виде

$$\begin{aligned} x = x_0 + V_m y &= x_0 + V_m T_m^{-1} \beta e_1 = x_0 + V_m (L_m U_m)^{-1} \beta e_1 = \\ &= x_0 + (V_m U_m^{-1}) (L_m^{-1} \beta e_1) = x_0 + P_m z_m. \end{aligned}$$

Здесь введены новые обозначения

$$P_m = V_m U_m^{-1}, \quad z_m = L_m^{-1} \beta e_1.$$

Обозначим элементы вектора z_m через ζ_k .

Записывая второе равенство в эквивалентном виде $L_m z_m = \beta e_1$, получаем

$$L_m z_m = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ \mu_2 & 1 & 0 & \cdots & 0 & 0 \\ 0 & \mu_3 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \mu_m & 1 \end{bmatrix} \cdot \begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \zeta_3 \\ \vdots \\ \zeta_{m-1} \\ \zeta_m \end{bmatrix} = \begin{bmatrix} \beta \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} = \beta e_1.$$

Расписывая последнюю строку произведения, имеем

$$\mu_m \zeta_{m-1} + \zeta_m = 0,$$

откуда немедленно следуют рекуррентные формулы

$$\zeta_1 = \beta, \quad \zeta_m = -\mu_m \zeta_{m-1}.$$

Перейдем к первому обозначению, которое эквивалентно соотношению $P_m U_m = V_m$. Рассмотрим его.

$$\begin{bmatrix} p_1 & \cdots & p_{m-1} & p_m \end{bmatrix} \cdot \begin{bmatrix} \eta_1 & \beta_2 & 0 & \cdots & 0 \\ 0 & \eta_2 & \beta_3 & \cdots & 0 \\ 0 & 0 & \eta_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & \beta_m \\ 0 & 0 & 0 & \cdots & \eta_m \end{bmatrix} = \begin{bmatrix} v_1 & \cdots & v_{m-1} & v_m \end{bmatrix}.$$

Как и везде раньше, через p_k обозначены столбцы матрицы P_m . Рассмотрим последний столбец произведения

$$v_m = \sum_{j=1}^m u_{j,m} p_j = \beta_m p_{m-1} + \eta_m p_m.$$

Отсюда получаем рекуррентную формулу пересчета

$$p_m = (v_m - \beta_m p_{m-1}) / \eta_m.$$

3. Распишем подробнее итоговую формулу алгоритма

$$x = x_0 + P_m z_m = x_0 + \zeta_1 p_1 + \zeta_2 p_2 + \cdots + \zeta_m p_m.$$

В совокупности с рекуррентными соотношениями для p_m и ζ_m они позволяют переписать алгоритм в итерационной форме с уточнением решения сразу после вычисления очередного вектора p_m .

Алгоритм D-Ланцош

1. Вычислить $r_0 = b - Ax_0$, $\beta = \|r_0\|_2$, $v_1 = r_0/\beta$
2. Положить (формально) $v_0 = 0$, $\beta_1 = 0$, $\mu_1 = 0$, $p_0 = 0$
3. **While** не достигнута точность
4. $w := Av_j - \beta_j v_{j-1}$
5. $\alpha_j := (w, v_j)$
6. **If** $j > 1$ **then** $\mu_j = \beta_j/\eta_{j-1}$, $\zeta_j = -\mu_j \zeta_{j-1}$ **EndIf**
6. $\eta_j := \alpha_j - \mu_j \beta_j$
6. $p_j := (v_j - \beta_j p_{j-1})/\eta_j$
6. $x_j := x_{j-1} + \zeta_j p_j$
6. $w := w - \alpha_j v_j$
7. $\beta_{j+1} := \|w\|_2$
8. **If** $\beta_{j+1} = 0$ **then** выйти из цикла **EndIf**
9. $v_{j+1} := w/\beta_{j+1}$
10. **EndWhile**

Построенный алгоритм обладает несколькими важными свойствами. Изучим их.

Утверждение 4.1. После m шагов алгоритма D-Ланцош невязки r_0, r_1, \dots, r_m взаимно ортогональны.

Доказательство.

Согласно утверждению 3.2

$$AV_m = V_m H_m + h_{m+1,m} v_{m+1} e_m^T.$$

В новых («симметричных») обозначениях оно принимает вид

$$AV_m = V_m T_m + \beta_{m+1} v_{m+1} e_m^T,$$

что позволяет записать выражение невязки в виде

$$\begin{aligned} r_m &= b - Ax_m = b - A(x_0 + V_m y_m) = (b - Ax_0) - AV_m y_m = \\ &= r_0 - V_m T_m y_m + \beta_{m+1} v_{m+1} e_m^T y_m. \end{aligned}$$

Заметим, что

$$V_m T_m y_m = V_m \beta e_1 = \beta v_1 = r_0,$$

а величина $e_m^T y_m = (e_m, y_m)$ есть скаляр. Тогда

$$r_m = r_0 - r_0 + \sigma v_{m+1}, \quad \text{где} \quad \sigma = \beta_{m+1} (e_m, y_m).$$

Таким образом, каждая очередная невязка r_m ($m = 1, 2, \dots$) коллинеарна каждому очередному вектору v_{m+1} , которые в совокупности образуют ортогональный базис. Утверждение доказано.

Утверждение 4.2. После m шагов алгоритма D-Ланцош векторы p_1, p_2, \dots, p_m взаимно A -ортогональны, то есть

$$(Ap_k, p_j) = 0 \quad \text{при } k \neq j.$$

Доказательство.

По определению $P_m = V_m U_m^{-1}$. Будем далее использовать обозначение $U_m^{-T} = (U_m^{-1})^T$.

Выполним преобразования:

$$\begin{aligned} P_m^T A P_m &= U_m^{-T} V_m^T A V_m U_m^{-1} = U_m^{-T} T_m U_m^{-1} = \\ &= U_m^{-T} L_m U_m U_m^{-1} = U_m^{-T} L_m. \end{aligned}$$

Матрица $U_m^{-T} L_m$ является нижнегребугольной как произведение двух нижнетреугольных матриц. Таким образом, матрица $P_m^T A P_m$ нижнетреугольна.

С другой стороны, матрица $P_m^T A P_m$ симметрична. Следовательно, она является диагональной.

Матричное произведение $P_m^T A P_m$ можно схематически изобразить в виде

$$P_m^T A P_m = P_m^T (A P_m) = \begin{array}{|c|} \hline p_1 \\ \hline p_2 \\ \hline \vdots \\ \hline p_m \\ \hline \end{array} \cdot \begin{array}{|c|c|c|c|} \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline \end{array} \cdot Ap_1 Ap_2 \cdots Ap_m$$

Равенство нулю внедиагональных элементов изображенного произведения описывается условием $(p_j, Ap_k) = 0$ при $j \neq k$, которое требовалось доказать.

4.3. Метод сопряженных градиентов

Знаменитый метод сопряженных градиентов, CG (от англ. conjugate gradients), можно вывести из метода Ланцоша, оптимизируя его вычислительный процесс.

Установленный выше факт взаимной ортогональности невязок и рекуррентные формулы для A -сопряженных направлений p_m позволяют полностью исключить из вычислительной схемы нахождение векторов базиса Ланцоша v_m . Разобьем преобразования на несколько этапов.

1. Выведем рекуррентное соотношение для невязок. Для этого соответствующее соотношение для приближений

$$x_j = x_{j-1} + \zeta_j p_j$$

умножим на матрицу A и вычтем вектор b .

$$Ax_j - b = Ax_{j-1} - b + \zeta_j Ap_j.$$

Изменяя знак левой и правой частей получим нужную формулу

$$r_j = r_{j-1} - \zeta_j Ap_j.$$

Коэффициент ζ_j найдем из условия ортогональности невязок, для чего домножим последнее равенство скалярно на r_{j-1} .

$$0 = (r_j, r_{j-1}) = (r_{j-1}, r_{j-1}) - \zeta_j (Ap_j, r_{j-1}),$$

откуда

$$\zeta_j = \frac{(r_{j-1}, r_{j-1})}{(Ap_j, r_{j-1})}.$$

2. Найдем новое рекуррентное соотношение для векторов p_j , не содержащее элементов базиса v_j . Как было установлено выше,

$$p_j = (v_j - \beta_j p_{j-1}) / \eta_j.$$

При доказательстве утверждения 4.1 было установлено, что вектора v_j и r_{j-1} коллинеарны. Поэтому вектор p_j линейно выражается через вектора r_{j-1} и p_{j-1} .

Так как длины векторов p_j не имеют большого значения, то можно положить один из коэффициентов линейной зависимости равным единице и искать новое направление в виде

$$p_j = r_{j-1} + \gamma_j p_{j-1}. \tag{7}$$

Коэффициент γ_j найдем из условия A -ортогональности, для чего домножим последнее равенство скалярно на Ap_{j-1} .

$$0 = (p_j, Ap_{j-1}) = (r_{j-1}, Ap_{j-1}) + \gamma_j (p_{j-1}, Ap_{j-1}),$$

откуда выводим

$$\gamma_j = -\frac{(Ap_{j-1}, r_{j-1})}{(Ap_{j-1}, p_{j-1})}.$$

3. Преобразуем полученные выражения коэффициентов. Во-первых, домножая (7) скалярно на Ap_j , получаем

$$(Ap_j, p_j) = (Ap_j, r_{j-1}) + \gamma_j (Ap_j, p_{j-1}) = (Ap_j, r_{j-1}).$$

Таким образом,

$$\zeta_j = \frac{(r_{j-1}, r_{j-1})}{(Ap_j, r_{j-1})} = \frac{(r_{j-1}, r_{j-1})}{(Ap_j, p_j)}.$$

Во-вторых, из связи невязок (на предыдущем шаге) имеем

$$Ap_{j-1} = -\frac{1}{\zeta_{j-1}}(r_{j-1} - r_{j-2}).$$

Поэтому

$$\gamma_j = \frac{1}{\zeta_{j-1}} \frac{(r_{j-1} - r_{j-2}, r_{j-1})}{(Ap_{j-1}, p_{j-1})} = \frac{(Ap_{j-1}, p_{j-1})}{(r_{j-2}, r_{j-2})} \frac{(r_{j-1}, r_{j-1})}{(Ap_{j-1}, p_{j-1})} = \frac{\|r_{j-1}\|_2^2}{\|r_{j-2}\|_2^2}.$$

4. В формулах предыдущего метода формально полагалось $p_0 = 0$. Заметим, что в этом случае

$$p_1 = r_0 + \gamma_1 p_0 = r_0$$

и коэффициент γ_1 вычислять нет необходимости.

Запишем все полученные формулы в виде алгоритма.

Алгоритм CG

1. Вычислить $r_0 := b - Ax_0$,
2. **For** $j = 1 \dots n$
3. **If** $j = 1$ **Then** $p_1 := r_0$
4. **Else**
5. $\gamma := \frac{\|r_{j-1}\|_2^2}{\|r_{j-2}\|_2^2}$
6. $p_j := r_{j-1} + \gamma p_{j-1}$
7. **EndIf**
8. $\zeta := \frac{\|r_{j-1}\|_2^2}{(Ap_j, p_j)}$
9. $x_j := x_{j-1} + \zeta p_j$
10. $r_j := r_{j-1} - \zeta Ap_j$
11. **If** достигнута точность **Then** выйти из цикла
12. **EndFor**

На каждом шаге основного цикла записаного метода вычисляются ровно два скалярных произведения. Кроме того, хранения требуют один вектор предыдущего направления p_{j-1} , предыдущее приближение x_{j-1} и его невязка r_{j-1} . Все вместе это приводит к существенному снижению затрат памяти и вычислительного времени по сравнению с алгоритмом D-Ланцош.

Отметим еще раз важный факт, который доказан при построении метода.

Утверждение 3.3. После выполнения k шагов метода CG справедливо свойство

$$\text{span}\{r_0, r_1, \dots, r_{k-1}\} = \text{span}\{p_1, p_2, \dots, p_k\} = \mathcal{K}_{k-1}(r_0, A).$$

Таким образом, проекционный процесс метода сопряженных градиентов приводит к построению ортогонального базиса из невязок. Отсюда, в частности, следует, что для систем порядка n будет получено точное решение не более, чем за n шагов (в точной арифметике).

С другой стороны, можно получить оценку, гарантирующую монотонное убывание погрешности (это будет сделано ниже), что позволяет завершать процесс гораздо раньше (при $k \ll n$) и рассматривать метод как итерационный. Приведем соответствующий факт без доказательства.

Теорема 4.1. Пусть $A^T = A > 0$. Тогда после m шагов метода CG верна оценка

$$\|x_m - \bar{x}\|_A \leq 2 \left(\frac{\sqrt{\rho} - 1}{\sqrt{\rho} + 1} \right)^m \|x_0 - \bar{x}\|_A,$$

где $\rho = \text{cond}_2(A)$, \bar{x} – точное решение системы.

5. ДВОЙСТВЕННАЯ ОРТОГОНАЛИЗАЦИЯ

5.1. Биортогонализация Ланцоша

Существенным отличием методов, основанных на процессе ортогонализации Ланцоша, от методов, построенных на процессе ортогонализации Арнольди, является объем информации, необходимый для построения очередного базисного вектора или, что то же, очередной поправки к решению. В то время, как процедура Арнольди требует использования всех предыдущих базисных векторов, алгоритм Ланцоша на каждом шаге применяет всего лишь трехчленные соотношения. Ясно, что такая грандиозная экономия достигается за счет симметричности матрицы системы.

Возникает вопрос, нельзя ли в несимметричном случае получить подобные экономичные формулы. Оказывается, на него можно дать положительный ответ, если изменить требования к базису. Точнее говоря, если строить не один, но сразу два базиса, связанные друг с другом.

Определение. Системы векторов $\{v_j\}$ и $\{w_j\}$ ($j = 1, \dots, n$) называются биортогональными, если

$$(v_j, w_k) = 0, \quad \text{при } j \neq k.$$

Выведем формулы, аналогичные процедуре Арнольди. Выберем два начальных вектора v_1 и w_1 , такие что $(v_1, w_1) \neq 0$. Предположим, что уже построены j векторов в каждой из биортогональных систем, и будем искать следующие вектора в виде

$$v_{j+1} = Av_j - \alpha_j v_j - \beta_j v_{j-1}, \quad (8)$$

$$w_{j+1} = A^T w_j - \alpha_j w_j - \beta_j w_{j-1}. \quad (9)$$

При этом формально положим $v_0 = 0$, $w_0 = 0$.

Умножим равенство (8) скалярно на w_j .

$$0 = (Av_j, w_j) - \alpha_j (v_j, w_j).$$

Откуда получаем, что

$$\alpha_j = \frac{(Av_j, w_j)}{(v_j, w_j)}.$$

Теперь умножим равенство (8) скалярно на w_{j-1} .

$$0 = (Av_j, w_{j-1}) - \beta_j (v_{j-1}, w_{j-1}).$$

Из этого равенства можно получить формулу для β_j , однако она будет содержать новое скалярное произведение с матрицей A . Его можно исключить, выполнив следующие преобразования. Перебросим матрицу в скалярном произведении на второй множитель и воспользуемся равенством (9).

$$\begin{aligned} 0 &= (v_j, A^T w_{j-1}) - \beta_j (v_{j-1}, w_{j-1}) = \\ &= (v_j, w_j) + \alpha_{j-1} (v_j, w_{j-1}) + \beta_{j-1} (v_j, w_{j-2}) - \beta_j (v_{j-1}, w_{j-1}). \end{aligned}$$

Поскольку второе и третье слагаемые в последнем выражении равны нулю, получаем

$$\beta_j = \frac{(v_j, w_j)}{(v_{j-1}, w_{j-1})}.$$

Запишем полученные соотношения в виде алгоритма.

Алгоритм биортогонализации Ланцоша

1. Выбрать вектора v_1, w_1 , такие что $(v_1, w_1) \neq 0$,
положить (формально) $v_0 = 0, w_0 = 0, \beta_1 = 0$
2. **For** $j = 1, \dots, n$
3. $\alpha_j = \frac{(Av_j, w_j)}{(v_j, w_j)}$
4. $v_{j+1} = Av_j - \alpha_j v_j - \beta_j v_{j-1}$
5. $w_{j+1} = A^T w_j - \alpha_j w_j - \beta_j w_{j-1}$
6. $\beta_{j+1} = \frac{(v_{j+1}, w_{j+1})}{(v_j, w_j)}$
7. **If** $\beta_{j+1} = 0$ **Then** Выйти из цикла **EndIf**
8. **EndFor**

Накопление вычислительных ошибок может приводить к тому, что получаемые системы векторов перестают быть взаимно ортогональными. Эта ситуация отслеживается в строке (7) алгоритма. Отметим принципиальное различие преждевременного прекращения процесса в двух сравниваемых подходах. В процедуре ортогонализации Арнольди преждевременный выход из цикла является кхорошимъ и свидетельствует об исчерпании размерности подпространства. В биортогонализации Ланцоша срыв вычислений является следствием критического накопления ошибок. Существуют и продолжают разрабатываться методы, направленные на преодоление такой ситуации. Однако наиболее простым выходом из положения является перезапуск, аналогичный рестартам в методах, обсуждавшихся выше.

Зафиксируем некоторые свойства нового алгоритма.

Утверждение 5.1. Алгоритм биортогонализации Ланцоша строит взаимно ортогональные базисы в пространствах Крылова $\mathcal{K}_n(A, v_1)$ и $\mathcal{K}_n(A^T, w_1)$ соответственно.

Доказательство этого факта следует непосредственно из формул (8), (9).

Введем трехдиагональную матрицу

$$T_m = \begin{bmatrix} \alpha_1 & \beta_2 & 0 & \cdots & 0 \\ 1 & \alpha_2 & \beta_3 & \cdots & 0 \\ 0 & 1 & \alpha_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & \alpha_n \end{bmatrix}$$

Утверждение 5.2. Справедливо равенство

$$AV_m = V_m T_m + v_{m+1} e_m^T,$$

где e_m – единичный орт с единицей в позиции m .

Доказательство.

Перепишем равенство (8) в виде

$$Av_j = \beta_j v_{j-1} + \alpha_j v_j + v_{j+1}.$$

Дополним матрицу V_m справа нулевым столбцом. Выполним перемножение

$$\begin{bmatrix} v_1 & \cdots & v_m & \bar{0} \end{bmatrix} \cdot \begin{bmatrix} \alpha_1 & \beta_2 & 0 & \cdots & 0 \\ 1 & \alpha_2 & \beta_3 & \cdots & 0 \\ 0 & 1 & \alpha_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & \alpha_m \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} =$$

$$= \begin{bmatrix} v_1 & \cdots & v_m \end{bmatrix} \cdot \begin{bmatrix} \alpha_1 & \beta_2 & 0 & \cdots & 0 \\ 1 & \alpha_2 & \beta_3 & \cdots & 0 \\ 0 & 1 & \alpha_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & \alpha_m \end{bmatrix} = V_m T_m$$

Рассмотрим далее нулевую матрицу порядка m , дополненную справа вектором-столбцом v_{m+1} . Выполним перемножение

$$\begin{array}{|c|c|c|c|c|} \hline \bar{0} & \cdots & \bar{0} & \bar{0} & v_{m+1} \\ \hline \end{array} \cdot \begin{array}{|c|c|c|c|c|} \hline \alpha_1 & \beta_2 & 0 & \cdots & 0 \\ 1 & \alpha_2 & \beta_3 & \cdots & 0 \\ 0 & 1 & \alpha_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & \alpha_m \\ \hline 0 & 0 & 0 & \cdots & 1 \\ \hline \end{array} = \\
 = \begin{array}{|c|c|c|c|} \hline \bar{0} & \cdots & \bar{0} & v_{m+1} \\ \hline \end{array} = v_{m+1} \cdot \begin{array}{|c|c|c|c|c|} \hline 0 & 0 & \cdots & 0 & 1 \\ \hline \end{array} = v_{m+1} e_m^T.$$

Таким образом,

$$V_m T_m + v_{m+1} e_m^T = \begin{array}{|c|c|c|c|} \hline v_1 & \cdots & v_m & v_{m+1} \\ \hline \end{array} \cdot \begin{array}{|c|c|c|c|c|} \hline \alpha_1 & \beta_2 & 0 & \cdots & 0 \\ 1 & \alpha_2 & \beta_3 & \cdots & 0 \\ 0 & 1 & \alpha_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & \alpha_m \\ \hline 0 & 0 & 0 & \cdots & 1 \\ \hline \end{array}.$$

Правая часть равенства является матрицей, j -й столбец которой вычисляется как

$$\beta_j v_{j-1} + \alpha_j v_j + v_{j+1}.$$

Согласно (8) это выражение равно Av_j , то есть совпадает с j -м столбцом матрицы AV_m . Свойство доказано.

Утверждение 5.3. Справедливо равенство

$$A^T W_m = W_m T_m^T + c w_{m+1} e_m^T,$$

где c – некоторая константа.

Доказательство полностью аналогично предыдущему с той лишь разницей, что следует рассматривать матричную интерпретацию равенства (9).

Утверждение 5.4. Справедливо равенство

$$W_m^T AV_m = D_m T_m = \tilde{T}_m,$$

где D_m – диагональная матрица.

Доказательство.

Умножим равенство

$$AV_m = V_m T_m + v_{m+1} e_m^T,$$

слева на матрицу W_m^T и заметим, что $W_m^T V_m$ есть диагональная матрица, поскольку составлена из скалярных произведений векторов w_i на v_j . По той же причине произведение $W_m^T v_{m+1}$ дает нулевой вектор. Свойство доказано.

5.2. Метод бисопряженных градиентов

В англоязычной литературе для данного метода принято использовать аббревиатуру BCG или BiCG (от англ. Biconjugate Gradients).

Превращение процедуры биортогонализации в итерационный метод решения СЛАУ (метод BCG) полностью аналогично построению метода CG на основе симметричной ортогонализации Ланцоша.

В соответствии с общей схемой проекционных методов решение уточняется по формуле

$$x = x_0 + V_m y,$$

где y является решением системы

$$W_m^T AV_m y = \beta e_1.$$

В нашем случае

$$W_m^T AV_m = \tilde{T}_m.$$

Как и в симметричном случае, воспользуемся LU разложением матрицы \tilde{T}_m и представим ее в виде

$$\tilde{T}_m = L_m U_m.$$

Из трехдиагональности \tilde{T}_m следует, что матрицы L_m и U_m будут двухдиагональными нижне- и верхнетреугольными матрицами.

Введем обозначения для их элементов, идентичные симметричному случаю

$$L_m = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ \mu_2 & 1 & 0 & \cdots & 0 \\ 0 & \mu_3 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}, \quad U_n = \begin{bmatrix} \eta_1 & \beta_2 & 0 & \cdots & 0 \\ 0 & \eta_2 & \beta_3 & \cdots & 0 \\ 0 & 0 & \eta_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & \eta_n \end{bmatrix}.$$

Введем также обозначения

$$P_m = V_m U_m^{-1}, \quad P_m^* = W_m L_m^{-T} = W_m (L_m^T)^{-1}.$$

Через p_k и p_k^* будем (как и выше) обозначать столбцы матриц P_m и P_m^* соответственно.

Рассмотрим первое равенство, предварительно переписав его в эквивалентном виде $P_m U_m = V_m$.

$$\begin{bmatrix} p_1 & \cdots & p_{m-1} & p_m \end{bmatrix} \cdot \begin{bmatrix} \eta_1 & \beta_2 & 0 & \cdots & 0 \\ 0 & \eta_2 & \beta_3 & \cdots & 0 \\ 0 & 0 & \eta_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & \beta_m \\ 0 & 0 & 0 & \cdots & \eta_m \end{bmatrix} = \begin{bmatrix} v_1 & \cdots & v_{m-1} & v_m \end{bmatrix}.$$

Распишем последний столбец произведения

$$v_m = \sum_{j=1}^m p_j u_{j,m} = \beta_m p_{m-1} + \eta_m p_m.$$

Отсюда получаем рекуррентную формулу

$$p_m = (v_m - \beta_m p_{m-1}) / \eta_m.$$

Рассмотрим теперь второе равенство, переписанное в эквивалентном виде $P_m^* L_m^T = W_m$.

$$\begin{bmatrix} p_1^* & \cdots & p_{m-1}^* & p_m^* \end{bmatrix} \cdot \begin{bmatrix} 1 & \mu_2 & 0 & \cdots & 0 \\ 0 & 1 & \mu_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & \mu_m \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} = \begin{bmatrix} w_1 & \cdots & w_{m-1} & w_m \end{bmatrix}.$$

Распишем последний столбец произведения

$$w_m = \sum_{j=1}^m p_j^* L_{j,m} = \mu_m p_{m-1}^* + p_m^*.$$

Отсюда получаем рекуррентную формулу

$$p_m^* = w_m - \mu_m p_{m-1}^*.$$

В качестве начального вектора при построении базиса V_m следует выбирать вектор невязки $r_0 = b - Ax_0$. Обозначим через r_0^* начальный вектор для построения базиса W_m . Поскольку матрица A^T невырождена, всегда найдется такой вектор b^* , для которого r_0^* является невязкой решения системы $A^T x^* = b^*$. Фактически, метод VCG, который будет сейчас построен, генерирует две последовательности приближений к решениям двух систем – с матрицей A и с матрицей A^T . Часто на практике решение второй системы не требуется и соответствующие ей последовательности векторов носят вспомогательный характер.

Утверждение 5.5. Пусть в качестве первого базисного вектора выбрана невязка $v_1 = r_0 = b - Ax_0$. Тогда каждая следующая невязка r_m коллинеарна соответствующему базисному вектору v_{m+1} .

Доказательство. Согласно утверждению 5.2

$$AV_m = V_m T_m + v_{m+1} e_m^T,$$

что позволяет записать выражение невязки в виде

$$\begin{aligned} r_m = b - Ax_m &= b - A(x_0 + V_m y_m) = (b - Ax_0) - AV_m y_m = \\ &= r_0 - V_m T_m y_m + v_{m+1} e_{m+1}^T y_m. \end{aligned}$$

Заметим, что

$$V_m T_m y_m = V_m \beta e_1 = \beta v_1 = r_0,$$

а величина $\sigma = e_{m+1}^T y_m = (e_{m+1}, y_m)$ есть скаляр. Тогда

$$r_m = r_0 - r_0 + \sigma v_{m+1}.$$

Свойство доказано.

Утверждение 5.6. Пусть первый двойственный базисный вектор $w_1 = r_0^* = b^* - A^T x_0^*$ является невязкой системы $A^T x^* = b^*$ с некоторой правой частью b^* . Тогда каждая следующая невязка r_m^* коллинеарна соответствующему базисному вектору w_{m+1} .

Доказательство аналогичным образом получается из соотношения

$$A^T W_m = W_m T_m^T + c w_{m+1} e_m^T.$$

Утверждение 5.7. После m шагов алгоритма невязки r_0, r_1, \dots, r_{m-1} и вспомогательные вектора $r_0^*, r_1^*, \dots, r_{m-1}^*$ (невязки системы с транспонированной матрицей) образуют биортогональные системы.

Доказательство. Согласно утверждениям 5.5 и 5.6,

$$r_j = \alpha_j v_{j+1}, \quad r_k^* = \beta_k w_{k+1},$$

где α_j и β_k – некоторые скаляры. Тогда

$$(r_j, r_k^*) = \alpha_j \beta_k (v_j, w_k) = 0 \quad \text{при} \quad j \neq k,$$

что и требовалось.

Вернемся к формуле уточнения решения и перепишем ее в новых обозначениях.

$$x = x_0 + V_m \tilde{T}_m^{-1} \beta e_1 = x_0 + V_m U_m^{-1} L_m^{-1} \beta e_1 = P_m z_m,$$

где $z_m = L_m^{-1} \beta e_1$. Обозначая (как и выше) через $\zeta_1, \zeta_2, \dots, \zeta_m$ компоненты вектора z_m , получаем, что

$$x = x_0 + P_m z_m = x_0 + \zeta_1 p_1 + \zeta_2 p_2 + \dots + \zeta_m p_m.$$

Полученное равенство можно переписать в форме, удобной для итерационного алгоритма

$$x_j = x_{j-1} + \zeta_j p_j.$$

Утверждение 5.8. После m шагов алгоритма векторы p_1, p_2, \dots, p_m и $p_1^*, p_2^*, \dots, p_m^*$ образуют A -сопряженные системы, то есть

$$(Ap_k, p_j^*) = 0 \quad \text{при} \quad k \neq j.$$

Доказательство. Напомним, что $P_m = V_m U_m^{-1}$, $P_m^* = W_m L_m^{-T}$. Выполним преобразования:

$$(P_m^*)^T A P_m = L_m^{-1} W_m^T A V_m U_m^{-1} = L_m^{-1} T_m U_m^{-1} = L_m^{-1} L_m U_m U_m^{-1} = I.$$

Матричное произведение $(P_m^*)^T A P_m$ можно схематически изобразить в виде

$$(P_m^*)^T A P_m = \begin{array}{|c|} \hline p_1^* \\ \hline p_2^* \\ \hline \vdots \\ \hline p_m^* \\ \hline \end{array} \cdot \begin{array}{|c|c|c|c|} \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline \end{array} \cdot Ap_1 Ap_2 \dots Ap_m$$

Равенство нулю внедиагональных элементов изображенного произведения описывается условием $(Ap_k, p_j^*) = 0$ при $j \neq k$, которое требовалось доказать.

Дальнейшие действия, как и раньше, оформим в виде последовательности небольших шагов.

1. Выведем рекуррентное соотношение для невязок. Для этого соответствующее соотношение для приближений

$$x_j = x_{j-1} + \zeta_j p_j$$

умножим на матрицу A и вычтем вектор b :

$$Ax_j - b = Ax_{j-1} - b + \zeta_j Ap_j.$$

Изменяя знак левой и правой частей получаем нужную формулу

$$r_j = r_{j-1} - \zeta_j Ap_j.$$

Коэффициент ζ_j найдем из условия биортогональности невязок, для чего домножим последнее равенство скалярно на r_{j-1}^* .

$$0 = (r_j, r_{j-1}^*) = (r_{j-1}, r_{j-1}^*) - \zeta_j (Ap_j, r_{j-1}^*),$$

откуда

$$\zeta_j = \frac{(r_{j-1}, r_{j-1}^*)}{(Ap_j, r_{j-1}^*)}.$$

2. Найдем рекуррентные соотношения для векторов p_j и p_j^* . Как было установлено выше,

$$p_j = (v_j - \beta_j p_{j-1}) / \eta_j$$

$$p_j^* = w_j - \mu_j p_{j-1}.$$

Согласно утверждениям 5.5 и 5.6, пары векторов v_j и r_{j-1} , а также w_j и r_{j-1}^* коллинеарны. Поэтому вектор p_j линейно выражается через вектора r_{j-1} и p_{j-1} , а вектор p_j^* линейно выражается через вектора r_{j-1}^* и p_{j-1} .

Так как длины векторов p_j и p_j^* не имеют большого значения, то можно положить один из коэффициентов в каждой линейной зависимости равным единице и искать новые направления в виде

$$p_j = r_{j-1} + \gamma_j p_{j-1}. \quad (10)$$

$$p_j^* = r_{j-1}^* + \gamma_j^* p_{j-1}^*. \quad (11)$$

Коэффициент γ_j найдем из условия A -биортогональности, для чего домножим последнее равенство скалярно на Ap_{j-1} .

$$0 = (Ap_{j-1}, p_j^*) = (Ap_{j-1}, r_{j-1}^*) + \gamma_j (Ap_{j-1}, p_{j-1}^*),$$

откуда выводим

$$\gamma_j^* = -\frac{(Ap_{j-1}, r_{j-1}^*)}{(Ap_{j-1}, p_{j-1}^*)}.$$

3. Преобразуем полученные выражения коэффициентов. Во-первых, домножая (11) скалярно на Ap_j , получаем

$$(Ap_j, p_j^*) = (Ap_j, r_{j-1}^*) + \gamma_j(Ap_j, p_{j-1}^*) = (Ap_j, r_{j-1}^*).$$

Таким образом,

$$\zeta_j = \frac{(r_{j-1}, r_{j-1}^*)}{(Ap_j, r_{j-1}^*)} = \frac{(r_{j-1}, r_{j-1}^*)}{(Ap_j, p_j^*)}.$$

Во-вторых, из связи невязок (на предыдущем шаге) имеем

$$Ap_{j-1} = -\frac{1}{\zeta_{j-1}}(r_{j-1} - r_{j-2}).$$

Поэтому

$$\gamma_j^* = \frac{1}{\zeta_{j-1}} \frac{(r_{j-1} - r_{j-2}, r_{j-1}^*)}{(Ap_{j-1}, p_{j-1}^*)} = \frac{(Ap_{j-1}, p_{j-1}^*)}{(r_{j-2}, r_{j-2}^*)} \frac{(r_{j-1}, r_{j-1}^*)}{(Ap_{j-1}, p_{j-1}^*)} = \frac{(r_{j-1}, r_{j-1}^*)}{(r_{j-2}, r_{j-2}^*)}.$$

4. Остается найти коэффициенты γ_j и ζ_j^* в рекуррентных соотношениях для p_j и r_j^* . Покажем, что

$$\zeta_j = \zeta_j^*, \quad \gamma_j = \gamma_j^*.$$

Будем искать сопряженную невязку в виде

$$r_j^* = r_{j-1}^* - \zeta_j^* A^T p_j^*.$$

Умножив это равенство скалярно на r_{j-1} и воспользовавшись биортогональностью невязок, получим

$$0 = (r_j^*, r_{j-1}) = (r_{j-1}^*, r_{j-1}) - \zeta_j^* (A^T p_j^*, r_{j-1}),$$

откуда

$$\zeta_j^* = \frac{(r_{j-1}, r_{j-1}^*)}{(A^T p_j^*, r_{j-1})}.$$

Заметим, что

$$(A^T p_j^*, p_k) = (p_j^*, Ap_k) = 0 \quad \text{при } j \neq k.$$

Поэтому умножение равенства (10) скалярно на $A^T p_j^*$ дает

$$(A^T p_j^*, p_j) = (A^T p_j^*, r_{j-1}).$$

Таким образом,

$$\zeta_j^* = \frac{(r_{j-1}, r_{j-1}^*)}{(A^T p_j^*, r_{j-1})} = \frac{(r_{j-1}, r_{j-1}^*)}{(A^T p_j^*, p_j)} = \frac{(r_{j-1}, r_{j-1}^*)}{(Ap_j, p_j^*)} = \zeta_j.$$

Первое равенство обосновано.

Для поиска γ_j умножим равенство (10) скалярно на $A^T p_{j-1}^*$.

$$0 = (A^T p_{j-1}^*, p_j) = (A^T p_{j-1}^*, r_{j-1}) + \gamma_j (A^T p_{j-1}^*, p_{j-1}),$$

откуда

$$\gamma_j = -\frac{(A^T p_{j-1}^*, r_{j-1})}{(A^T p_{j-1}^*, p_{j-1})}.$$

Величину $A^T p_{j-1}^*$ выразим из рекуррентной формулы для сопряженных невязок

$$A^T p_{j-1}^* = -\frac{1}{\zeta_{j-1}^*} (r_{j-1}^* - r_{j-2}^*).$$

Поэтому

$$\gamma_j = \frac{1}{\zeta_{j-1}^*} \frac{(r_{j-1} - r_{j-2}, r_{j-1}^*)}{(A^T p_{j-1}^*, p_{j-1})} = \frac{1}{\zeta_{j-1}^*} \frac{(r_{j-1}, r_{j-1}^*)}{(Ap_{j-1}, p_{j-1}^*)} = \gamma_j^*.$$

Вывод формул завершен. Запишем их в виде алгоритма.

Алгоритм BiCG

1. Вычислить $r_0 := b - Ax_0$,
выбрать произвольный вектор $r_0^* : (r_0, r_0^*) \neq 0$
2. $p_1 := r_0, \quad p_1^* := r_0^*$
3. **For** $j = 1 \dots n$
4. $\zeta := \frac{(r_{j-1}, r_{j-1}^*)}{(Ap_{j-1}, p_{j-1}^*)}$
5. $x_j := x_{j-1} + \zeta p_{j-1}$
6. $r_j := r_{j-1} - \zeta Ap_{j-1}$
7. $r_j^* := r_{j-1}^* - \zeta A^T p_{j-1}^*$
8. **If** достигнута точность **then** выйти из цикла
9. $\gamma := \frac{(r_j, r_j^*)}{(r_{j-1}, r_{j-1}^*)}$
10. $p_{j+1} := r_j + \gamma p_j$
11. $p_{j+1}^* := r_j^* + \gamma p_j^*$
12. **EndFor**

Построение правильного критерия окончания метода ВСС является сложным и до конца не решенным вопросом. Большую трудность создает отсутствие простых апостериорных оценок. По этой причине на практике, как правило, отслеживают поведение невязки. Один из простейших вариантов условия выхода – уменьшение величины невязки в заданное количество раз относительно ее начального значения.

СПИСОК ЛИТЕРАТУРЫ

1. Амосов, А.А., Вычислительные методы / А.А. Амосов, Ю.А. Дубинский, Н.В. Копченова. – М.: Издательский дом МЭИ, 2008.
2. Баландин, М.Ю. Методы решения СЛАУ большой размерности / М.Ю. Баландин, Э.П. Шурина. – Новосибирск: Издательство НГТУ, 2000.
3. Воеводин, В.В. Матрицы и вычисления / В.В. Воеводин, Ю.А. Кузнецов. – М.: Наука, 1984.
4. Голуб, Дж. Матричные вычисления / Дж. Голуб, Ч. Ван Лоун. – М.: Мир. 1999.
5. Деммель, Дж. Вычислительная линейная алгебра / Дж. Деммель. – М.: Мир, 2001.
6. Саад, Ю. Итерационные методы для разреженных линейных систем / Ю. Саад. – М.: Издательство Московского университета, 2013.
7. Уоткинс, Д. Основы матричных вычислений / Д. Уоткинс. – М.: Бином, 2006.
8. Фаддеев, Д.К. Вычислительные методы линейной алгебры / Д.К. Фаддеев, В.Н. Фаддеева. – СПб.: Лань, 2002.

Оглавление

Предисловие 3

1. Ортогональное проектирование и линейные системы	4
1.1. Постановка задачи	4
1.2. Разрешимость задачи ортогонального проектирования ...	4
2. Одномерные проекции	8
2.1. Проектирование на одномерное подпространство	8
2.2. Метод наискорейшего спуска	9
2.3. Метод минимальных невязок	10
2.4. Метод наискорейшего уменьшения невязки	11
2.5. Декартовы проекции	12
3. Ортогонализация в подпространствах Крылова	13
3.1. Подпространства Крылова. Ортогонализация Арнольди .	13
3.2. Метод полной ортогонализации	19
3.3. Алгоритм полной ортогонализации с рестартами	21
3.4. Алгоритм неполной ортогонализации	22
3.5. Обобщенный метод минимальных невязок	22
4. Симметричная ортогонализация	27
4.1. Ортогонализация Ланцоша	27
4.2. Прямой метод Ланцоша	29
4.3. Метод сопряженных градиентов	32
5. Двойственная ортогонализация	36
5.1. Биортогонализация Ланцоша	36
5.2. Метод бисопряженных градиентов	40
Список литературы	47