

Итерационные методы для решения СЛАУ

План

1. Двухслойные итерационные схемы
 - Метод Якоби
 - Метод Зейделя
 - Методы неполной релаксации
 - Сравнение двухслойных итерационных методов
2. Многослойные итерационные схемы
 - Чебышевское ускорение*
3. Сравнение метода Якоби и метода Гаусса

Двухслойные итерационные схемы

Пусть решаем следующую СЛАУ:

$$Ax = f, \quad A \in \mathbb{R}^{n,n}, \quad x, f \in \mathbb{R}^{n,1}$$

До этого решали её *прямыми методами* - т.е. находили алгоритмами, которые за заранее определенное количество шагов находил решение (тот же метод Гаусса).

Сейчас же хотим решать системку **итерационно** - т.е. постепенно уточнять наше решение в зависимости от шага итерации. Останавливаемся при достижении определённой точности ε .

Как правило, это гораздо более быстрый путь. Самый простой способ реализации - зная текущее приближение (текущий слой), построить из него новое (следующий слой) - так называемая двухслойная схема.

В общем виде **двухслойная итерационная схема** выглядит следующим образом:

$$R_k \frac{x^{(k+1)} - x^{(k)}}{\tau^{(k)}} + Ax^{(k)} = f$$

где $R_k \in \mathbb{R}^{n,n}$, $\tau^{(k)} \in \mathbb{R}$ - ненулевые параметры итерации. Матрицы R_k должны быть невырожденные и положительно определённые.

Очевидно, что если последовательность $x^{(k)}$ является сходящейся, то она сходится к решению исходной системы. При этом на каждом шаге мы всё равно обязаны решить систему относительно $x^{(k+1)}$ - это должно быть проще решения изначальной задачи.

Таким образом, коэффициенты итерации $R_k, \tau^{(k)}$ надо подбирать исходя из:

1. Сходимости последовательности $x^{(k)}$
2. Простоты решения задачи $R_k \cdot x^{(k+1)} = \tau^{(k)} f + (R_k \cdot x^{(k)} - A\tau^{(k)})x^{(k)}$

Понятно, что по сути своей этот метод очень схож с методом простой итерации. За более подробным общим рассмотрением такой задачи смотреть [здесь](#) или большим количеством простых примеров и теоремок по приведенным методам [тута](#)

Приведём некоторые примеры двухслойных итераций.

Метод Якоби

Пусть имеем систему:

$$A\vec{x} = \vec{b}$$

Хотим привести его к итерационному виду:

$$\vec{x} = B\vec{x} + \vec{g}$$

Метод Якоби основан на отделении диагонали матрицы A от других её элементов:

$$A = (L + D + U)$$

где мы разделили нашу матрицу на сумму нижнетреугольной, диагональной и верхнетреугольной. Подставляя в изначальное уравнение:

$$D\vec{x} = -(L + U)\vec{x} + \vec{b}$$

$$\vec{x} = -D^{-1}(L + U)\vec{x} + D^{-1}\vec{b}$$

Получили итерационный вид. Осталось навесить номера итерации и

Тогда итерационная схема выглядит как:

$$\vec{x}^{(k+1)} = B\vec{x}^{(k)} + \vec{g}$$

$$B = -D^{-1}(L + U), \quad \vec{g} = D^{-1}\vec{b}$$

Или, в раскрытом виде:

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j \neq i} a_{ij} x_j^{(k)} \right), \quad i = 1, 2, \dots, n$$

Теорема 1. Критерий сходимости метода Якоби

Метод Якоби сходится тогда и только тогда, когда $|\lambda_i| < 1$, где λ_i - корни следующего уравнения:

$$\det \begin{vmatrix} \lambda a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & \lambda a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & \lambda a_{nn} \end{vmatrix} = 0$$

Теорема 2. Достаточное условие сходимости метода Якоби.

Пусть $\|B\| < 1$. Тогда при любом выборе начального приближения $\vec{x}^{(0)}$:

1. Метод сходится
2. Скорость сходимости метода равна скорости сходимости геометрической прогрессии со знаменателем $q = \|B\|$
3. Верна оценка погрешности: $\|\vec{x}^{(k)} - \vec{x}\| < q^k \|\vec{x}^{(0)} - \vec{x}\|$

Теорема 3. Достаточное условие сходимости метода Якоби.

Метод Якоби сходится, если для матрицы изначального уравнения выполнено **условие диагонального преобладания**:

$$\forall i \rightarrow |a_{ii}| \geq \sum_{j \neq i} |a_{ij}|$$

Притом хотя бы одно из неравенств должно быть строгим.

Метод Зейделя

Пусть имеем систему:

$$A\vec{x} = \vec{b}$$

Хотим привести его к итерационному виду:

$$\vec{x} = B\vec{x} + \vec{g}$$

Опять разобьём матрицу A на сумму нижнетреугольной, диагональной и верхнетреугольной:

$$A = (L + D + U)$$

И построим итерационную формулу из изначальной системы:

$$(L + D)\vec{x}^{(k+1)} = -U\vec{x}^{(k)} + \vec{b}$$

$$\vec{x}^{(k+1)} = -(L + D)^{-1}U\vec{x}^{(k)} + (L + D)^{-1}\vec{b}$$

Метод Гаусса — Зейделя можно рассматривать как модификацию метода Якоби. Основная идея модификации состоит в том, что новые значения $\vec{x}^{(i)}$ используются здесь сразу же по мере получения, в то время как в методе Якоби они не используются до следующей итерации:

$$\begin{cases} x_1^{(k+1)} = c_{12}x_2^{(k)} + c_{13}x_3^{(k)} + \dots + c_{1n}x_n^{(k)} + d_1 \\ x_2^{(k+1)} = c_{21}x_1^{(k+1)} + c_{23}x_3^{(k)} + \dots + c_{2n}x_n^{(k)} + d_2 \\ \dots \\ x_n^{(k+1)} = c_{n1}x_1^{(k+1)} + c_{n2}x_2^{(k+1)} + \dots + c_{nn}x_n^{(k+1)} + d_n \end{cases}$$

где

$$c_{ij} = \begin{cases} -\frac{a_{ij}}{a_{ii}}, & j \neq i, \\ 0, & j = i. \end{cases} \quad d_i = \frac{b_i}{a_{ii}}, \quad i = 1, \dots, n$$

Теорема 4. Критерий сходимости метода Зейделя

Метод Зейделя сходится тогда и только тогда, когда $|\lambda_i| < 1$, где λ_i - корни следующего уравнения:

$$\det \begin{vmatrix} \lambda a_{11} & a_{12} & \dots & a_{1n} \\ \lambda a_{21} & \lambda a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ \lambda a_{n1} & \lambda a_{n2} & \dots & \lambda a_{nn} \end{vmatrix} = 0$$

Теорема 5. Достаточное условие сходимости метода Зейделя.

Метод Зейделя сходится, если для матрицы изначального уравнения выполнено **условие диагонального преобладания**:

$$\forall i \rightarrow |a_{ii}| \geq \sum_{j \neq i} |a_{ij}|$$

Притом хотя бы одно из неравенств должно быть строгим.

Методы неполной релаксации

В целом, метод Зейделя работает лучше метода Якоби. Это наталкивает на мысль о том, что можно ещё чутька улучшить метод, особо не меняя сути. Помните параметр τ в общем определении двухслойной схемы? Сейчас он нам понадобится.

Пусть $z_i^{(k+1)}$ - i -я компонента $k+1$ приближения Зейделя. Пусть теперь следующую итерацию мы будем считать по формуле

$$x_i^{(k+1)} = x_i^{(k)} + \omega(z_i^{(k+1)} - x_i^{(k)})$$

При $\omega = 1$ получается метод Зейделя в чистом виде. Метод при $\omega \in (0, 1)$ называют **методом нижней релаксации**. Если $\omega \in (1, 2)$ - **методом верхней релаксации**. Переписав выражение, видим, что по сути мы "чутька оставляем" прошлый слой при построении следующего:

$$x_i^{(k+1)} = (1 - \omega)x_i^{(k)} + \omega z_i^{(k+1)}$$

Матричная запись данного метода выглядит как:

$$(L + \omega D) \frac{x^{(k+1)} - x^{(k)}}{\omega} + Ax^{(k)} = f$$

Откуда понятно, что параметр $\tau = \omega$ имеет релаксационный смысл (т.е. своей рода "консервативности" метода).

Теорема 6.:

Пусть матрица $A = A^* > 0$, тогда метод релаксации сходится $\forall \omega \in (0, 2)$.

При каких ω метод сходится быстрее? В общем случае на этот вопрос нет универсального ответа, однако математиками было рассмотрено много частных случаев.

Рассмотрим один важный класс задач, для которого найдена такая ω .

Пусть P - матрица смены базиса такая, что : $PAP^T = \begin{bmatrix} D_1 & T_{12} \\ T_{21} & D_2 \end{bmatrix}$, где D_i - диагональные матрицы. Т.е. "разделили переменные на два кластера".

Для таких задач найден оптимальный параметр релаксации

$$\omega_{\text{опт}} = \frac{2}{1 + \sqrt{1 - \rho^2(B)}}$$

Тут B - матрица перехода Якоби итерационного метода ($\rho(B)$ - спектральный радиус, то есть наибольшее по модулю собственное число) в итерационном уравнении вида:

$$\vec{x} = B\vec{x} + \vec{g}$$

Сравнение двухслойных итерационных методов

Сравним скорости сходимости для конкретной системы:

$$\begin{cases} 2x_1 + x_2 = 1 \\ x_1 + 2x_2 = -1 \end{cases}$$

Точное решение системы имеет вид:

$$\vec{x}^* = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

Решим систему уравнений методами Якоби, Зейделя и ПВР.
Тогда для метода Якоби получим:

$$\begin{cases} x_1^{(s+1)} = -0.5x_2^{(s)} + 0.5 \\ x_2^{(s+1)} = -0.5x_1^{(s)} - 0.5 \end{cases}$$

Для метода Зейделя:

$$\begin{cases} x_1^{(s+1)} = -0.5x_2^{(s)} + 0.5 \\ x_2^{(s+1)} = -0.5x_1^{(s+1)} - 0.5 \end{cases}$$

Для метода ПВР:

$$\begin{cases} x_1^{(s+1)} = (1 - \omega)x_1^{(s)} + \omega(-0.5x_2^{(s)} + 0.5) \\ x_2^{(s+1)} = (1 - \omega)x_2^{(s)} + \omega(-0.5x_1^{(s+1)} - 0.5) \end{cases}$$

где значение параметра было выбрано $\omega = 1.1$.

Для каждого метода рассмотрим три итерации. Тогда получим следующие значения:

	$x_1^{(3)}$	$x_2^{(3)}$	$\ \vec{x}^{(3)} - \vec{x}^*\ $
Я к о б и	0.875	-0.875	0.125
З е й д е л ь	0.969	-0.984	0.031
ПВР	1.0008	-1.0009	< 0.0001

Из таблицы видно, что метод ПВР сходится быстрее, чем методы Якоби и Зейделя. Вспомним, что для системы, в которой переменные можно разделить на два класса так, чтобы каждое уравнение связывало элемент одного класса с элементами другого, можно найти оптимальное значение параметра $\omega_{\text{опт}}$. Для систем уравнений с двумя переменными такое разделение выполняется всегда. Таким образом, при условии сходимости метода Якоби для системы второго порядка всегда можно найти оптимальное значение параметра.

Рассмотрим, чему равен спектральный радиус матрицы перехода для метода Якоби.

$$\begin{aligned} \vec{x} &= R\vec{x} + \vec{F} \\ R_{\text{Якоби}} &= \begin{pmatrix} 0 & -0.5 \\ -0.5 & 0 \end{pmatrix} \\ \lambda(R_{\text{Якоби}}) &= \pm \frac{1}{2}, \quad \Rightarrow \quad \rho(R_{\text{Якоби}}) = \frac{1}{2} \end{aligned}$$

Тогда для метода ПВР получим:

$$\omega_{\text{опт}} = \frac{2}{1 + \sqrt{1 - \left(\frac{1}{2}\right)^2}} \approx 1.07$$

Именно поэтому значение ω было выбрано равным 1.1.

Многослойные итерационные схемы

Рассмотрим СЛАУ:

$$Ax = f$$

Пусть СЛАУ привели к виду:

$$x = Bx + g$$

Рассмотрим метод простых итераций, который сходится к точному решению:

$$x^{(k+1)} = Bx^{(k)} + g$$

$$x^{(k)} \rightarrow x^{(*)}$$

Рассмотрим известную последовательность приближений $x^{(0)}, x^{(1)}, x^{(2)} \dots x^{(m)}$. На основе данной последовательности построим новое приближение:

$$y_m = \sum_{i=0}^m \gamma_{im} x^{(i)}$$

Если $x^{(0)} = x^{(1)} = x^{(2)} = \dots = x^{(m)} = x^{(*)}$, т.е. уже сидим в решении и никуда из него не уходим. Тогда необходимое условие того, чтобы эта взвешенная сумма давала нам правильный ответ - единичная нормировка весов слагаемых:

$$\sum_{i=0}^m \gamma_{im} = 1$$

Такой процедурой можно уточнять решения и строить многослойные схемы.

Рассмотрим **невязку** такой процедуры :

$$\begin{cases} x^{(m)} = Bx^{(m-1)} + g \\ x^{(*)} = Bx^{(*)} + g \end{cases}$$

Тогда невязка решения будет:

$$\delta = y_m - x^{(*)} = \sum_{i=0}^m \gamma_{im} x^{(i)} - x^{(*)} = \sum_{i=1}^m \gamma_{im} (x^{(i)} - x^{(*)}) = \sum_{i=1}^m \gamma_{im} (Bx^{(m-1)} - Bx^{(*)}) = P_m(B)(x^{(0)} - x^{(*)})$$

$$\text{где } P_m(B) = \sum_{i=1}^m \gamma_{im} B^i.$$

Чебышевское ускорение*

Многочлены Чебышева.

Многочленами Чебышева называют:

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$$

$$T_0(x) = 1, T_1(x) = x$$

Все многочлены определены на отрезке $[-1; 1]$ и для всех выполняется $\max(|T_n|) = 1$.

Среди всех многочленов, значения которых на отрезке $[-1, 1]$ не превосходят по модулю 1, многочлен Чебышёва имеет наибольший старший коэффициент, наибольшее значение в любой точке за пределами этого отрезка, а нули многочлена Чебышева являются оптимальными узлами во многих вычислительных схемах.

Приведенным многочленом Чебышева назовем $\bar{T}_n(x) = 2^{1-n} T_n(x)$.

Рассмотрим многочлены вида:

$$P_m(x) = \frac{T_m(\frac{x}{\rho})}{T_m(\frac{1}{\rho})}$$

Для такого многочлена выполняется :

- Он определен от $-\rho$ до ρ
- $P_m(1) = 1$

Рассмотрим $\rho = 1 + \varepsilon$, тогда :

$$P_m(x) \leq \frac{1}{T_m(1 + \varepsilon)}$$

$$P_m(x) = \frac{1}{T_m(\frac{1}{\rho})} T_m(\frac{x}{\rho})$$

Обозначим $\mu_m = \frac{1}{T_m(\frac{1}{\rho})}$, тогда

$$\mu_m = \frac{2}{\rho\mu_{m-1}} - \frac{1}{\mu_{m-2}}$$

Рассмотрим, как поменяется решение :

$$y_m - x^* = P_m(B)(x^{(0)} - x^*) = \mu_m T_m\left(\frac{B}{\rho}\right)(x^{(0)} - x^*) = \mu_m \left(\frac{2}{\rho} B(x^{(m-1)} - x^*) \frac{1}{\mu_{m-1}} - (x^{(m-2)} - x^*) \frac{1}{\mu_{m-2}}\right)$$

Приведем выражение к виду:

$$y_m = \frac{2\mu_m}{\mu_{m-1}} \frac{B}{\rho} x^{(m-1)} - \frac{\mu_m}{\mu_{m-2}} x^{(m-2)} + dm$$

$$, \text{ где } dm = x^* \left(1 - \frac{2\mu_m}{\mu_{m-1}} \frac{B}{\rho} + \frac{\mu_m}{\mu_{m-2}}\right)$$

Для x^* выполняется уравнение $x^* = Bx^* + g$, откуда $Bx^* = x^* - g$, тогда:

$$dm = \mu_m \left(\frac{1}{\mu_m} - \frac{2}{\rho\mu_{m-1}} - \frac{1}{\mu_{m-2}}\right) x^* + 2 \frac{\mu_m}{\mu_{m-1}} g$$

В силу ранее полученных выражений первая скобка равна 0.

В итоге получаем итерационный процесс :

$$\begin{cases} \mu_0 = 1, \mu_1 = \rho \\ y_0 = x^{(0)}, y_1 = Bx^{(0)} + g \\ \mu_m = \left(\frac{2}{\rho\mu_{m-1}} - \frac{1}{\mu_{m-2}}\right)^{-1} \\ y_m = \frac{2\mu_m}{\rho\mu_{m-1}} By_{m-1} - \frac{\mu_m}{\mu_{m-2}} By_{m-2} + \frac{2\mu_m}{\rho\mu_{m-2}} g \end{cases}$$

В данном ускорении используются все предыдущие вычисления, но в итоговой формуле остались только 2 предыдущих шага. Т.е., данный алгоритм является трехслойным.

$$\begin{aligned} a_{ij} &= 2, & a_{i,i+1} &= -1 - \alpha, & a_{i,i-1} &= -1 + \alpha \\ f_1 &= 1 - \alpha, & f_i &= 0, & i &= 2, 3, \dots, n-1, & f_n &= 1 + \alpha \end{aligned}$$