

Итерационные методы решения систем линейных алгебраических уравнений (СЛАУ)

Постановка задачи

Возьмём систему: $A\vec{x} = \vec{b}$, где $A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}$, $\vec{b} = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}$.

$$\text{Или } \begin{cases} a_{11}x_1 + \cdots + a_{1n}x_n = b_1 \\ \dots \\ a_{n1}x_1 + \cdots + a_{nn}x_n = b_n \end{cases}.$$

Метод Якоби

Метод Якоби – разновидность метода простой итерации для решения системы линейных алгебраических уравнений. Назван в честь Карла Густава Якоби.

Для того, чтобы построить итеративную процедуру метода Якоби, необходимо провести предварительное преобразование системы уравнений $A\vec{x} = \vec{b}$ к итерационному виду $\vec{x} = B\vec{x} + \vec{g}$. Оно может быть осуществлено по одному из следующих правил:

- 1) $B = E - D^{-1}A = D^{-1}(D - A)$, $\vec{g} = D^{-1}\vec{b}$;
- 2) $B = -D^{-1}(L + U) = -D^{-1}(A - D)$, $\vec{g} = D^{-1}\vec{b}$;
- 3) $D_{ii}^{-1} = 1/D_{ii}$, $D_{ii} \neq 0$, $i = 1, 2, \dots, n$;

где в принятых обозначениях D означает матрицу, у которой на главной диагонали стоят соответствующие элементы матрицы A , а все остальные нули; тогда как матрицы U и L содержат верхнюю и нижнюю треугольные части A , на главной диагонали которых нули, E – единичная матрица.

Тогда процедура нахождения решения имеет вид:

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

Или в виде поэлементной формулы:

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

где k счётчик итерации.

В отличие от метода Гаусса-Зейделя $x_i^{(k)}$ нельзя заменять на $x_i^{(k+1)}$ в процессе итерационной процедуры, так как эти значения понадобятся для остальных вычислений. Это наиболее значимое различие между методом Якоби и методом Гаусса-Зейделя решения СЛАУ. Таким образом на каждой итерации придётся хранить оба вектора приближений: старый и новый.

Условие окончания итерационного процесса при достижении точности ε имеет вид:

$$\|x_i^{(k+1)} - x_i^{(k)}\| \leq \varepsilon.$$

Данный критерий не требует вычисления нормы матрицы и часто используется на практике. При этом точное условие окончания итерационного процесса имеет вид

$$\|Ax^{(k)} - b\| \leq \varepsilon$$

и требует дополнительного умножения матрицы на вектор на каждой итерации, что примерно в два раза увеличивает вычислительную сложность алгоритма.

Метод Гаусса-Зейделя решения системы линейных уравнений

Метод Гаусса-Зейделя (метод Зейделя, процесс Либмана, метод последовательных замещений) является классическим итерационным методом решения системы линейных уравнений.

Назван в честь Зейделя и Гаусса.

Чтобы пояснить суть метода, перепишем задачу в виде

$$\begin{cases} a_{11}x_1 & = -a_{12}x_2 - a_{13}x_3 - \dots - a_{1n}x_n + b_1 \\ a_{21}x_1 + a_{22}x_2 & = -a_{23}x_3 - \dots - a_{2n}x_n + b_2 \\ \dots & \\ a_{(n-1)1}x_1 + a_{(n-1)2}x_2 + \dots + a_{(n-1)(n-1)}x_{n-1} & = -a_{(n-1)n}x_n + b_{n-1} \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{n(n-1)}x_{n-1} + a_{nn}x_n & = b_n \end{cases}$$

Здесь в j -м уравнении в правую часть перенесены все члены, содержащие x_i , для $i > j$. Эта запись может быть представлена как

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

где в принятых обозначениях D означает матрицу, у которой на главной диагонали стоят соответствующие элементы матрицы A , а все остальные нули; тогда как матрицы U и L содержат верхнюю и нижнюю треугольные части A , на главной диагонали которых нули.

Итерационный процесс в методе Гаусса-Зейделя строится по формуле

$$(L + D)\vec{x}^{(k+1)} = -U\vec{x}^{(k)} + \vec{b}, k = 0, 1, 2, \dots$$

после выбора соответствующего начального приближения $\vec{x}^{(0)}$.

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:

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

где

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

Таким образом, i -я компонента $(k + 1)$ -го приближения вычисляется по формуле

$$x_i^{(k+1)} = \sum_{j=1}^{i-1} c_{ij}x_j^{(k+1)} + \sum_{j=i}^n c_{ij}x_j^{(k)} + d_i, i = 1, \dots, n.$$

Условие окончания итерационного процесса Зейделя при достижении точности ε в упрощённой форме имеет вид:

$$\|x_i^{(k+1)} - x_i^{(k)}\| \leq \varepsilon.$$

Более точное условие окончания итерационного процесса имеет вид

$$\|Ax^{(k)} - b\| \leq \varepsilon$$

и требует больше вычислений. Хорошо подходит для разреженных матриц.

Метод релаксации

Метод релаксации – итерационный метод решения систем линейных алгебраических уравнений.

Система линейных уравнений

$$\begin{cases} a_{11}x_1 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + \dots + a_{2n}x_n = b_2 \\ \dots \\ a_{n1}x_1 + \dots + a_{nn}x_n = b_n \end{cases}$$

приводится к виду

$$\begin{cases} P_{11}x_1 + P_{12}x_2 + \dots + P_{1n}x_n + c_1 = 0 \\ \dots \\ P_{n1}x_1 + P_{n2}x_2 + \dots + P_{nn}x_n + c_n = 0 \end{cases},$$

где $P_{ij} = -\frac{a_{ij}}{a_{ii}}$, $c_i = \frac{b_i}{a_{ii}}$. То есть все $P_{ii} = -1$.

Находятся невязки R_j :

$$\begin{cases} R_1^{(0)} = c_1 - x_1^{(0)} + \sum_{j=2}^n P_{1j}x_j^{(0)} \\ R_2^{(0)} = c_2 - x_2^{(0)} + \sum_{j=1, j \neq 2}^n P_{2j}x_j^{(0)} \\ \dots \\ R_n^{(0)} = c_n - x_n^{(0)} + \sum_{j=1}^{n-1} P_{nj}x_j^{(0)} \end{cases}$$

Выбирается начальное приближение $X^{(0)} = 0$. На каждом шаге необходимо обратить в ноль максимальную невязку: , .

Условие остановки: $|R_j^{(k)}| < \varepsilon, \forall j = \overline{1, n}$.

Ответ находится по формуле: $x_i \approx x_i^{(0)} + \sum_j \delta x_i^{(j)}$.

Методы градиентного и наискорейшего спуска

Метод градиентного спуска состоит в нахождении следующего приближения в итерационном процессе из предыдущего, путем смещения в направлении градиента функционала

$$\Phi(\mathbf{u}) = (\mathbf{A}\mathbf{u}, \mathbf{u}) - 2(\mathbf{f}, \mathbf{u}) \quad (2.25)$$

$$\mathbf{u}_{k+1} = \mathbf{u}_k - \alpha_k \cdot \text{grad}\Phi(\mathbf{u}_k), \quad (2.26)$$

где \mathbf{A} — положительно определенная симметричная матрица; α_k — параметр, определяемый из заданных условий; например, из условия минимума величины

$$\Phi[\mathbf{u}_k - \alpha_k \cdot \text{grad}\Phi(\mathbf{u}_k)].$$

В этом случае *итерационный метод* называется **методом наискорейшего спуска**. Так

как $\text{grad}\Phi(\mathbf{u}) = 2(\mathbf{A}\mathbf{u} - \mathbf{f})$, то (2.26) приобретает вид

$$\mathbf{u}_{k+1} = \mathbf{u}_k - \tau_k(\mathbf{A}\mathbf{u}_k - \mathbf{f}), \quad \text{где } \tau_k = 2\alpha_k, \quad (2.27)$$

что соответствует записи *итерационного метода* в форме (2.17). Здесь τ_k является итерационным параметром, который в *методе наискорейшего спуска* определяется из условия минимума

функции $\Phi(\tau_k, \mathbf{u}_{k+1})$ по τ_k . Найдем условие этого минимума:

$$0 = 2(\mathbf{A}\mathbf{u}_{k+1} - \mathbf{f}, (\mathbf{u}_{k+1})'_{\tau_k}) = -2(\mathbf{A}\mathbf{u}_{k+1} - \mathbf{f}, \mathbf{A}\mathbf{u}_k - \mathbf{f}).$$

Здесь учтено соотношение: $(\mathbf{A}\mathbf{u}', \mathbf{u}) = (\mathbf{u}', \mathbf{A}^*\mathbf{u}) = (\mathbf{A}\mathbf{u}, \mathbf{u}')$,

поскольку $\mathbf{A} = \mathbf{A}^*$ и $(\mathbf{v}, \mathbf{A}\mathbf{w}) = (\mathbf{A}\mathbf{w}, \mathbf{v})$ в силу самосопряженности

оператора \mathbf{A} . Подставим в последнее равенство \mathbf{u}_{k+1} из (2.27),

получим $(\mathbf{A}\mathbf{u}_k - \mathbf{f} - \tau_k\mathbf{A}(\mathbf{A}\mathbf{u}_k - \mathbf{f}), \mathbf{A}\mathbf{u}_k - \mathbf{f}) = 0$, откуда следует

$$\mathbf{A}\mathbf{u}_k - \mathbf{f}, \mathbf{A}\mathbf{u}_k - \mathbf{f}) - \tau_k(\mathbf{A}(\mathbf{A}\mathbf{u}_k - \mathbf{f}), \mathbf{A}\mathbf{u}_k - \mathbf{f}) = 0,$$

$$\tau_k = (\mathbf{A}\mathbf{u}_k - \mathbf{f}, \mathbf{A}\mathbf{u}_k - \mathbf{f}) / (\mathbf{A}(\mathbf{A}\mathbf{u}_k - \mathbf{f}), \mathbf{A}\mathbf{u}_k - \mathbf{f}), \text{ или}$$

$$\tau_k = \frac{(\mathbf{r}_k, \mathbf{r}_k)}{(\mathbf{A}\mathbf{r}_k, \mathbf{r}_k)}$$

, где $\mathbf{r}_k = \mathbf{A}\mathbf{u}_k - \mathbf{f}$.

Вектор \mathbf{r}_k называют **вектором невязки**.

Метод минимальных невязок

Этот *итерационный метод* определяется следующим образом. Пусть $\mathbf{u}_{k+1} = \mathbf{u}_k - \tau_k \mathbf{r}_k$, как и ранее, $\mathbf{r}_k = \mathbf{A}\mathbf{u}_k - \mathbf{f}$. Итерационный параметр τ_k на каждой итерации выбирается так, чтобы минимизировать евклидову норму невязки \mathbf{r}_{k+1} . Заметим, что итерационный процесс $\mathbf{u}_{n+1} = \mathbf{u}_n + \tau_n \mathbf{r}_n$ может быть представлен в равносильном виде в терминах невязки $\mathbf{r}_{n+1} = \mathbf{r}_n + \tau_n \mathbf{A}\mathbf{r}_n$. Тогда для квадрата евклидовой (третьей) нормы невязки получаем условие

$$(\mathbf{r}_{k+1}, \mathbf{r}_{k+1}) = (\mathbf{r}_k, \mathbf{r}_k) - 2\tau_k(\mathbf{A}\mathbf{r}_k, \mathbf{r}_k) + \tau_k^2(\mathbf{A}\mathbf{r}_k, \mathbf{A}\mathbf{r}_k).$$

Для отыскания минимума невязки на следующей итерации приравняем нулю производную последнего выражения по итерационному параметру τ_k . Получим равенство

$$-2(\mathbf{A}\mathbf{r}_k, \mathbf{r}_k) + 2\tau_k(\mathbf{A}\mathbf{r}_k, \mathbf{A}\mathbf{r}_k) = 0.$$

Из последнего соотношения находим значение итерационного параметра

$$\tau_k = \frac{(\mathbf{A}\mathbf{r}_k, \mathbf{r}_k)}{(\mathbf{A}\mathbf{r}_k, \mathbf{A}\mathbf{r}_k)}.$$