

Лекция

ЧИСЛЕННОЕ РЕШЕНИЕ СИСТЕМ ЛИНЕЙНЫХ АЛГЕБРИЧЕСКИХ УРАВНЕНИЙ (СЛАУ) (3 часа)

(краткое изложение)

Постановка задачи и методы ее решения.

Дана СЛАУ

$$Ax = B, \quad (1)$$

где $A_{n \times n}, B_{n \times 1}$ - известные матрица и вектор-столбец, $x_{n \times 1}$ - вектор-столбец неизвестных переменных.

Здесь рассматривается основной случай, когда $B \neq 0, |A| \neq 0$, то есть матрица A не вырожденная и поэтому система (1) имеет единственное решение.

Все методы решения системы (1) можно условно разделить на две группы:

- 1) прямые методы или методы, основанные преобразованиях СЛАУ;
- 2) итерационные методы.

К прямым методам можно отнести Крамера (определителей), методы Гаусса, метод прогонки и др. Эти методы не имеют методической погрешности, так как основаны или на аналитическом решении (метод Крамера, метод прогонки)), или на эквивалентном преобразовании СЛАУ к наиболее простому виду. Однако, как правило, эти методы имеют ограниченное применение, так как могут использоваться для систем со сравнительно небольшой размерностью. Так, например, метод Крамера требует примерно $N = n!n$ элементарных операций. Если $n = 30$, то задача становится очень трудоемкой даже для современных ЭВМ. Меньшего объема операций требует метод Гаусса $N \approx 0.65n(n^2 + 3n + 1)$, однако здесь возникает другая трудность: с увеличением размерности n начинает накапливаться вычислительная погрешность (погрешность округления чисел). Она примерно пропорциональна n .

Для СЛАУ большой размерности (порядка 10^4 и более) необходимо использовать итерационные методы, которые заключаются в использовании разностного уравнения

$$x^{(k+1)} = Cx^{(k)} + F, \quad (2)$$

где k - номер итерации, C, F - матрицы, которые соответствуют применяемому методу. Для начала итерации (2) необходимо задать начальное приближение $x^{(0)}$.

Итерационная процедура должно сходиться, то есть

$$\lim_{k \rightarrow \infty} x^{(k)} = x \text{ (решение)} \quad (3)$$

Свойства сходимости зависят от матрицы C .

Условие сходимости: для сходимости итераций (3) необходимо и достаточно, чтобы все собственные числа матрицы C по модулю были меньше единицы.

Однако для итерационной процедуры (2) существует и более простой критерий сходимости, который определяется непосредственно свойствами матрицы A . Это условие диагонального преобладания

$$|a_{ii}| > \sum_{i \neq j} |a_{ij}|, \quad (4)$$

для всех $i = 1, \dots, n$. Условие (4) является достаточным условием сходимости метода (2).

В качестве условий остановки итераций можно использовать неравенство

$$\|x^{(k+1)} - x^{(k)}\| < \delta, \quad (5)$$

где δ - заданная погрешность определения решения, $\|\cdot\|$ - знак евклидовой нормы.

Можно также проверять непосредственно погрешность решения системы (1)

$$\|Ax^{(k)} - B\| < \delta. \quad (6)$$

Матричный метод Гаусса.

Существует несколько вариантов метода Гаусса [1]. Здесь рассмотрим вариант, основанный на матричных вычислениях. Основные этапы метода Гаусса: 1) прямой ход; 2) обратный ход. Прямой ход служит для приведения матрицы A к верхней треугольной форме. Обратный ход – для непосредственного определения решения, так как из каждого уравнения определяется по одной неизвестной переменной.

Прямой ход основывается на матричных преобразованиях

$$N_{n-1} \dots N_2 N_1 A = A^\Delta. \quad (7)$$

где матрица N_k обнуляет элементы k -ого столбца, стоящие под главной диагональю.

Матрицы N_k определяются правилу

$$N_k = \begin{pmatrix} 1 & 0 & \dots & \dots & \dots & \dots & 0 \\ 0 & \dots & 0 & \dots & \dots & \dots & 0 \\ 0 & 0 & \dots & 0 & \dots & \dots & 0 \\ 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ 0 & \dots & 0 & -\frac{a_{k+1,k}}{a_{k,k}} & \dots & 0 & 0 \\ 0 & \dots & 0 & \dots & 0 & \dots & 0 \\ 0 & \dots & 0 & -\frac{a_{n,k}}{a_{k,k}} & 0 & \dots & 1 \end{pmatrix},$$

причем матрица N_k определяется по элементам предыдущей матрице $N_{k-1} \dots N_1 A$.

После приведения матрицы A к треугольному виду необходимо также преобразовать вектор B : $N_{k-1} \dots N_1 A$.

Обратный ход заключается в решении СЛАУ $A^\Delta x = B^\Delta$, которое не представляет трудность. Так, например, из последнего уравнения находится

$x_n = B_n^\Delta / a_{nn}^\Delta$, из предпоследнего при известном x_n определяется x_{n-1} и т. д. вверх до первого уравнения, из которого определяется x_1 .

Как следует из приведенной матрицы N_k , ее ненулевой столбец вычисляется с использованием знаменателя $a_{k,k}$. Понятно, что этот элемент не может быть нулевым. При реализации алгоритмов вычислений на каждом шаге прямого хода рекомендуется осуществлять перестановку строк или столбцов текущей матрицы (также возможна полная перестановка) так, чтобы модуль $|a_{k,k}|$ был максимален. Такой подход часто называют методом главного элемента. Известно также [1], что выбор главного элемента уменьшает в общем случае накапливающую вычислительную погрешность.

Обусловленность СЛАУ.

Существуют СЛАУ, решение которых вызывает определенные трудности. Эти СЛАУ характеризуются условием $|A| \approx 0$. Если $|A| \approx 0$, то малая погрешность в знании вектора B или компонент матрицы A может существенно изменить искомое решение. Если рассмотреть простейший случай двумерной системы $n = 2$, то условие $|A| \approx 0$ означает, что имеется две почти параллельные прямые. Поэтому любое малое «шевеление» (или возмущение, например, от неизбежно возникающей ошибки округления) может привести к существенному изменению искомого решения. Такие СЛАУ принято называть плохо обусловленными. В общем случае такие задачи называются некорректными [2].

Для сравнения различных СЛАУ обычно вводят так называемые числа обусловленности, по величине которых можно судить о корректности решаемой задачи. Числа обусловленности определяются с использованием различных норм матриц и векторов. Рассмотрим определение числа обусловленности в соответствии с евклидовой нормой. Пусть, например, вектор B известен с некоторой погрешностью. Обозначим приближенное его

значение \tilde{B} . Рассмотрим решение двух систем $Ax = B$ и $A\tilde{x} = \tilde{B}$. Тогда изменение (вариация) решения определится из системы, записанной для отклонений

$$A\delta x = \delta B. \quad (7)$$

где $\delta x = \tilde{x} - x$, $\delta B = \tilde{B} - B$.

Формальное решение системы (7) имеет вид $\delta x = A^{-1}\delta B$. Тогда, используя известные свойства норм матриц, получим неравенства

$$\|\delta x\| \leq \|A^{-1}\| \|\delta B\|, \quad \|B\| \leq \|A\| \|x\|. \quad (8)$$

Перемножая неравенства (8), приведем их к виду

$$\frac{\|\delta x\|}{\|x\|} \leq M_A \frac{\|\delta B\|}{\|B\|}, \quad (9)$$

где $M_A = \|A\| \|A^{-1}\|$ называется числом обусловленности.

Чем больше M_A , тем большее влияние может оказать ошибка δB на величину δx . Наименьшее значение числа $M_A = 1$. Получены оценки, из которых следует, что если $M_A = 10^k$ ($k = 1, 2, \dots$), то можно потерять k значащих цифр в числе, которое определяет результат решения.

Существует и более общая оценка погрешности решения, если имеют место ошибки в исходных данных $\|\delta B\|$ и $\|\delta A\|$

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{M_A}{1 - M_A \|\delta A\| / \|A\|} \left(\frac{\|\delta B\|}{\|B\|} + \frac{\|\delta A\|}{\|A\|} \right), \quad (10)$$

От величины числа обусловленности зависит, например, погрешность решения СЛАУ (1) методом Гаусса. Существует оценка

$$\frac{\|\delta x\|}{\|x\|} \leq CM_A n 2^{-t}, \quad (11)$$

где $C < \infty$ - некоторая константа, t - число разрядов мантиссы числа в двоичной системе исчисления.

Свойства числа обусловленности:

1. $M_A \geq 1$;
2. $M_A \geq \lambda_{\max} / \lambda_{\min}$;
3. $M_A = \lambda_{\max} / \lambda_{\min}$ - для симметричной матрицы.

Здесь $\lambda_{\max}, \lambda_{\min}$ - максимальное и минимальное по модулю собственные значения матрицы A .

Таким образом, оценка числа обусловленности сводится к оценке собственных значений матрицы $\lambda_{\max}, \lambda_{\min}$, то к другой задаче вычислительной математики, которая будет рассмотрена позже.

Итерационные методы решения СЛАУ.

При использовании итерационных методов для СЛАУ используется разностное уравнение (2), причем для начала итерационного процесса необходимо задать начальное приближение $x^{(0)}$. Причем матрица C (2) должна быть построена так, чтобы выполнялись условия сходимости алгоритма, которые заключаются в том, чтобы все собственные числа матрицы $|\lambda_i(C)| < 0$, где $i = 1, 2, \dots, n$.

Наиболее известными итерационными методами решения СЛАУ являются методы Якоби и Зейделя. Для их построения запишем систему (1) в скалярной форме

$$\begin{array}{l} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = B_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = B_2 \\ \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = B_n \end{array}. \quad (11)$$

Перепишем (11) в виде

$$x_i = -\sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} x_j - \sum_{j=i+1}^n \frac{a_{ij}}{a_{ii}} x_j + \frac{B_i}{a_{ii}}. \quad (12)$$

где $i = 1, 2, \dots, n$.

Организуем итерационный процесс по формуле

$$x_i^{(k+1)} = - \sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} x_j^{(k)} - \sum_{j=i+1}^n \frac{a_{ij}}{a_{ii}} x_j^{(k)} + \frac{B_i}{a_{ii}}. \quad (13)$$

Разностное уравнение реализует метод Якоби.

Можно усовершенствовать процесс (13). Можно заметить, что при определении $x_i^{(k+1)}$ значения $x_j^{(k+1)}$, где $j = 1, 2, \dots, i-1$, уже известны из предыдущих уравнений. Поэтому можно записать

$$x_i^{(k+1)} = - \sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} x_j^{(k+1)} - \sum_{j=i+1}^n \frac{a_{ij}}{a_{ii}} x_j^{(k)} + \frac{B_i}{a_{ii}}. \quad (14)$$

Формула (14) соответствует методу Зейделя для СЛАУ.

Из формул (13-14) следует, что также как для метода Гаусса на главной диагонали матрицы не должно быть нулевых элементов. Этого можно добиться также перестановкой строк и столбцов матрицы A , то есть предварительно необходимо сделать тождественные преобразования системы (1).

Для представления итерационных формул (13-14) в матричной форме (2) представим матрицу A в виде суммы $A = A_1 + D + A_2$, где A_1 - нижняя треугольная матрица, D - диагональная матрица (включает главную диагональ матрицы A), A_2 - верхняя треугольная матрица.

Тогда систему (1) можно записать в виде

$$(A_1 + D + A_2)x = B. \quad (15)$$

В этом случае итерационный метод Якоби (13) записывается в виде

$$x^{(k+1)} = -D^{-1}(A_1 + A_2)x^{(k)} + D^{-1}B. \quad (16)$$

Следовательно, для метода Якоби $C = -D^{-1}(A_1 + A_2)$ и $F = D^{-1}B$ (см. (2)).

Соответственно метод Зейделя можно записать в виде

$$x^{(k+1)} = -(D + A_1)^{-1} A_2 x^{(k)} + (D + A_1)^{-1} B. \quad (17)$$

Тогда $C = -(D + A_1)^{-1} A_2$ и $F = (D + A_1)^{-1} B$.

Сходимость матричных алгоритмов обеспечивается, как было сказано выше, свойствами матрицы C . Для этого рекомендуется делать предварительные преобразования исходной матрицы A (перестановка строк и столбцов) таким образом, чтобы сумма абсолютных величин диагональных элементов матрицы принимал наибольшее значение: $\max \sum_{i=1}^n |a_{ii}|$.

Итерационные методы решения СЛАУ с параметром.

Введение дополнительного параметра в итерационные формулы во многих случаях позволяет улучшить свойства сходимости итерационных алгоритмов решения СЛАУ.

Введем дополнительный параметр в методы Якоби и Зейделя. Для этого итерационную формулу для метода Якоби (16) перепишем в виде

$$D(x^{(k+1)} - x^{(k)}) = -(A_1 + D + A_2)x^{(k)} + B. \quad (18)$$

Здесь в правую и левую часть выражения (18) добавлено одинаковое слагаемое $-Dx^{(k)}$. Из (18) следует

$$D(x^{(k+1)} - x^{(k)}) = -Ax^{(k)} + B. \quad (19)$$

Формально введем в (19) параметр $\Delta\tau$, поделив на него приращение $x^{(k+1)} - x^{(k)}$, тогда

$$D \frac{x^{(k+1)} - x^{(k)}}{\Delta\tau} = -Ax^{(k)} + B. \quad (20)$$

Введение таким образом параметра можно объяснить тем, что в этом случае мы получаем разностное уравнение, которое в точности совпадает со случаем, когда для системы обыкновенных уравнений ($\Delta\tau \rightarrow 0$) вида

$$D \frac{dx}{d\tau} = -Ax + B. \quad (21)$$

применяется метод интегрирования Эйлера и решается начальная задача с начальным условием $x(0) = x^{(0)}$. Если существует неподвижная точка

системы $-Ax + B = 0$ и она устойчива, то при достаточно малом шаге интегрирования (параметре $\Delta\tau$) численное решение будет стремиться к решению СЛАУ.

Разностное уравнение (20) при этом можно записать в форме (2)

$$x^{(k+1)} = x^{(k)} + \Delta\tau D^{-1} \left(-Ax^{(k)} + B \right). \quad (22)$$

или

$$x^{(k+1)} = \left(E - \Delta\tau D^{-1} A \right) x^{(k)} + \Delta\tau D^{-1} B. \quad (23)$$

Тогда виду (2) соответствуют матрицы $C = E - \Delta\tau D^{-1} A$, $F = \Delta\tau D^{-1} B$, где E - единичная матрица.

Выбором параметра $\Delta\tau$ можно улучшить свойства сходимости итерационной формулы (23).

Аналогично можно ввести параметр в метод Зейделя. После проведения аналогичных преобразований нетрудно получить систему дифференциальных уравнений

$$(D + A_1) \frac{dx}{d\tau} = -Ax + B. \quad (24)$$

При этом итерационный алгоритм в форме (2) будет иметь вид

$$x^{(k+1)} = \left(E - \Delta\tau (D + A_1)^{-1} A \right) x^{(k)} + \Delta\tau (D + A_1)^{-1} B, \quad (25)$$

то есть $C = \left(E - \Delta\tau (D + A_1)^{-1} A \right)$, $F = \Delta\tau (D + A_1)^{-1} B$.

Читателям предоставляется возможность самим провести необходимые преобразования для получения выражений (24-25).

Здесь необходимо отметить, что существование аналогов решения задач для СЛАУ в виде систем обыкновенных дифференциальных уравнений (21) и (24) позволяет строить другие методы решения СЛАУ. Например, для решения систем (21) и (24) с заданными начальными условиями можно применить любой метод интегрирования систем обыкновенных

дифференциальных уравнений порядка точности $p > 1$ (например, методы Рунге-Кутты), а не только метод Эйлера ($p = 1$).

Метод квадратного корня.

Кроме универсальных методов решения СЛАУ, существуют специальные методы решения СЛАУ, разработанные для матриц A специального вида. Здесь рассмотрим один из таких методов: метод квадратного корня. Это метод применяется, когда матрица A симметрична.

Метод основывается на разложении матрицы A в виде произведения

$$A = S^* D S, \quad (26)$$

где S - верхняя треугольная матрица (включая диагональные элементы), D - диагональная матрица, $(^*)$ - знак транспонирования.

Если такое разложение сделано (26), то решение СЛАУ сводится к решению двух систем с диагональными матрицами, что не вызывает затруднений. Эти системы имеют вид

$$S^* D y = B, \quad S x = y. \quad (27)$$

Сначала решается первая система и определяется y , потом вторая.

В качестве примера рассмотрим разложение матрицы A на множители для случая $n = 2$. Для этого надо перемножить матрицы $S^* D S$, приравняв их к матрице A . Тогда

$$S^* D S = \begin{pmatrix} S_{11} & 0 \\ S_{12} & S_{22} \end{pmatrix} \begin{pmatrix} d_{11} & 0 \\ 0 & d_{22} \end{pmatrix} \begin{pmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{pmatrix}. \quad (28)$$

После перемножения

$$\begin{pmatrix} S_{11}^2 d_{11} & S_{11} S_{12} d_{11} \\ S_{11} S_{12} d_{11} & S_{12}^2 d_{11} + S_{22}^2 d_{22} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{pmatrix}. \quad (29)$$

Приравнивая элементы матриц, получим выражения

$$S_{11}^2 d_{11} = a_{11}, S_{11} S_{12} d_{11} = a_{12}, S_{12}^2 d_{11} + S_{22}^2 d_{22} = a_{22}. \quad (30)$$

Данная система решается следующим образом

$$\begin{aligned} S_{11} &= \sqrt{|a_{11}|}, d_{11} = \text{sgn}(a_{11}), S_{12} = a_{12} / S_{11}d_{11}, \\ S_{22} &= \sqrt{|a_{22} - S_{12}^2 d_{11}|}, d_{22} = \text{sgn}(a_{22} - S_{12}^2 d_{11}), \end{aligned} \quad (31)$$

$$\text{где } \operatorname{sgn}(x) = \begin{cases} 1, \dots \text{если} \dots x > 0 \\ -1, \dots \text{если} \dots x < 0 \\ 0, \dots \text{если} \dots x = 0 \end{cases}$$

Данный алгоритм можно обобщить на квадратную матрицу произвольного размера, тогда

$$S_{ii} = \sqrt{\left| a_{ii} - \sum_{k=1}^{i-1} S_{ik}^2 d_{ii} \right|}, d_{ii} = \text{sgn} \left(a_{ii} - \sum_{k=1}^{i-1} S_{ik}^2 d_{ii} \right), i = 1, 2, \dots, n$$

$$S_{ij} = \frac{a_{ij} - \sum_{k=1}^{i-1} S_{ik} S_{jk} d_{kk}}{d_{ii} S_{ii}}, \quad i < j, j = 2, \dots, n$$

Метод квадратного корня является экономичным алгоритмом решения СЛАУ с симметричной матрицей.

Метод прогонки.

Метод прогонки применяется для СЛАУ, которые имеют ленточную матрицу. Простейшая ленточная матрица имеет вид

$$\begin{aligned} &a_{11}x_1 + a_{12}x_2 = B_1 \\ &a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = B_3 \\ &\dots\dots\dots . \\ &a_{n-1,n-2}x_{n-2} + a_{n-1,n-1}x_{n-1} + a_{n-1,n}x_n = B_{n-1} \\ &a_{n,n-1}x_{n-1} + a_{n,n}x_n = B_n \end{aligned} \tag{32}$$

То есть в матрице A отличны от нуля только компоненты, стоящие на главной диагонали и на двух ей параллельных.

Запишем произвольное уравнение из системы (32) в виде

$$a_{j,j-1}x_{j-1} + a_{j,j}x_j + a_{j,j+1}x_{j+1} = B_j, \quad (33)$$

где $j = 1, 2 \dots n$. Если в уравнении (33) какой-либо индекс для коэффициентов $a_{k,m}$ равен нулю или $n+1$, то данный коэффициент равен нулю (это справедливо для первого и последнего уравнения системы (33)).

Будем искать решение уравнений (33) в виде итерационных соотношений

$$x_j = \gamma_{j+1}x_{j+1} + \beta_{j+1}, \quad j = 1, 2 \dots n, \quad (34)$$

где коэффициенты $\gamma_{j+1}, \beta_{j+1}$ надо определить.

Для определения коэффициентов уравнений (34) подставим (34) в (33), учитывая, что

$$x_{j-1} = \gamma_j x_j + \beta_j = \gamma_j (\gamma_{j+1} x_{j+1} + \beta_{j+1}) + \beta_j.$$

Тогда

$$\begin{aligned} & a_{j,j-1} (\gamma_j (\gamma_{j+1} x_{j+1} + \beta_{j+1}) + \beta_j) + \\ & + a_{j,j} (\gamma_{j+1} x_{j+1} + \beta_{j+1}) + a_{j,j+1} x_{j+1} = B_j \end{aligned}, \quad (35)$$

Преобразуем (35) к виду

$$\begin{aligned} & x_{j+1} [a_{j,j-1} \gamma_j \gamma_{j+1} + a_{j,j} \gamma_{j+1} + a_{j,j+1}] + \\ & + [a_{j,j-1} \gamma_j \beta_{j+1} + a_{j,j-1} \beta_j + a_{j,j} \beta_{j+1} - B_j] = 0 \end{aligned}, \quad (36)$$

Соотношение (36) должно быть тождеством при любом x_{j+1} . Это может быть тогда и только тогда, когда будут равны нулю выражения в (36), стоящие в квадратных скобках. В этом случае

$$\begin{aligned} & a_{j,j-1} \gamma_j \gamma_{j+1} + a_{j,j} \gamma_{j+1} + a_{j,j+1} = 0 \\ & a_{j,j-1} \gamma_j \beta_{j+1} + a_{j,j-1} \beta_j + a_{j,j} \beta_{j+1} - B_j = 0 \end{aligned} \quad (37)$$

Из соотношений получаем итерационные уравнения для определения искомых коэффициентов

$$\gamma_{j+1} = -\frac{a_{j,j+1}}{a_{j,j-1} \gamma_j + a_{j,j}}, \quad j = 1, 2 \dots n-1 \quad (38)$$

$$\beta_{j+1} = \frac{B_j - a_{j,j-1}\beta_j}{a_{j,j-1}\gamma_j + a_{j,j}}, \quad j = 1, 2 \dots n \quad (39)$$

Таким образом, алгоритм метода прогонки включает в себя следующие этапы:

1. Прямой ход, определение коэффициентов по формулам (38-39).
2. Обратный ход, расчет решений СГАУ по формулам (34), начиная с определения x_n .

1. Коварцев А.Н. Вычислительная математика. – Самара: ООО «Офорт», 2011. – 230 с.
2. Тихонов А. Н., Арсенин В. Я. Методы решения некорректных задач. — М.: Наука, 1979. — 283 с.