

2 Численные методы решения СЛАУ

Численные методы решения систем линейных алгебраических уравнений делятся на прямые и итерационные. К прямым методам относят те вычислительные алгоритмы, которые дают точное решение системы при условии, что все арифметические операции выполняются точно. Итерационные методы строят последовательность векторов, которая теоретически сходится с точному решению.

2.1 Верхне- и нижнетреугольные системы

Определение. Матрица A размерности $n \times n$ называется верхнетреугольной, если $\forall i > j \quad a_{ij} = 0$. Матрица A размерности $n \times n$ называется нижнетреугольной, если $\forall i < j \quad a_{ij} = 0$.

Теорема 2.1 Пусть матрица системы $Ax = b$ верхнетреугольная и $a_{ii} \neq 0, \quad i = 1, \dots, n$. Тогда существует единственное решение системы, которое задается формулой

$$x_n = \frac{b_n}{a_{nn}},$$
$$x_i = \frac{b_i - \sum_{j=i+1}^n a_{ij}x_j}{a_{ii}}, \quad i = n-1, \dots, 1.$$

Пример. Исследовать системы

$$\left\{ \begin{array}{l} 4x_1 - x_2 + 2x_3 + 3x_4 = 20 \\ -2x_2 + 7x_3 - 4x_4 = -7 \\ 6x_3 + 5x_4 = 4 \\ 3x_4 = 6 \end{array} \right. \quad \left\{ \begin{array}{l} 4x_1 - x_2 + 2x_3 + 3x_4 = 20 \\ 0x_2 + 7x_3 - 4x_4 = -7 \\ 6x_3 + 5x_4 = 4 \\ 3x_4 = 6 \end{array} \right.$$
$$\left\{ \begin{array}{l} 4x_1 - x_2 + 2x_3 + 3x_4 = 20 \\ 0x_2 + 7x_3 + 0x_4 = -7 \\ 6x_3 + 5x_4 = 4 \\ 3x_4 = 6 \end{array} \right.$$

Задача 2.1.1. Решить системы уравнений, реализовав алгоритм на языке \mathcal{L}

$$\left\{ \begin{array}{l} 3x_1 - 2x_2 + x_3 - x_4 = 8 \\ 4x_2 - x_3 + 2x_4 = -3 \\ 2x_3 + 3x_4 = 11 \\ 5x_4 = 15 \end{array} \right. \quad \left\{ \begin{array}{l} 5x_1 - 3x_2 - 7x_3 + x_4 = -14 \\ 11x_2 + 9x_3 + 5x_4 = 22 \\ 3x_3 - 13x_4 = -11 \\ 7x_4 = 14 \end{array} \right.$$

Задача 2.1.2. Решить системы уравнений, реализовав алгоритм на языке \mathcal{L}

$$\left\{ \begin{array}{l} 4x_1 - x_2 + 2x_3 + 2x_4 - x_5 = 4 \\ -2x_2 + 6x_3 + 2x_4 + 7x_5 = 0 \\ x_3 - x_4 - 2x_5 = 3 \\ -2x_4 - x_5 = 10 \\ 3x_5 = 6 \end{array} \right. \quad \left\{ \begin{array}{l} 7x_4 = 14 \\ 3x_3 - 13x_4 = -11 \\ 11x_2 + 9x_3 + 5x_4 = 22 \\ 5x_1 - 3x_2 - 7x_3 + x_4 = -14 \end{array} \right.$$

Задача 2.1.3. Будет ли произведение треугольных матриц треугольной матрицей? Ответ обосновать.

Задача 2.1.4. Подсчитать количество арифметических операций в алгоритме решения треугольных систем.

2.2 Метод исключения переменных Гаусса

Рассмотренный выше частный случай может помочь при решении систем общего (не треугольного вида). На этом основан метод исключения (элиминации) переменных Гаусса. Используя элементарные преобразования систем уравнений, он сводит (при некоторых ограничениях) систему общего вида к эквивалентной треугольной системе.

Напомним, что элементарными преобразованиями систем уравнений являются:

- перестановка двух уравнений местами;
- умножение одного из уравнений на ненулевой множитель;
- замена одного из уравнений его линейной комбинацией с другим уравнением.

Таким образом, используя элементарные преобразования, можно построить матрицу системы (точнее, расширенную матрицу системы, т.е. $(A|b)$), которая будет эквивалентна исходной, однако иметь нужную «конфигурацию».

Теорема 2.2 Пусть матрица A системы $Ax = b$ невырожденная. Тогда существует верхнетреугольная система $Ux = y$, эквивалентная ей, причем $u_{ii} \neq 0, i = \overline{1, n}$.

Алгоритм метода Гаусса применительно к невырожденным системам можно описать следующим образом:

Шаг 1. Создать расширенную матрицу системы $\hat{A} = (A|b)$.

Шаг 2. Если необходимо, поменять первую строчку расширенной матрицы с какой-нибудь другой ниже нее так, чтобы $\hat{a}_{11} \neq 0$. Затем исключить переменную x_1 из строк со 2-ой по n -ую:

$$\hat{a}_{rc} := \hat{a}_{rc} - \frac{\hat{a}_{r1}}{\hat{a}_{11}} \hat{a}_{1c}, \quad r = \overline{2, n}, \quad c = \overline{2, n+1}.$$

Шаг $p+1$. Если необходимо, поменять p -ую строчку расширенной матрицы с какой-нибудь другой ниже нее так, чтобы $\hat{a}_{pp} \neq 0$. Затем исключить переменную x_p из строк с $(p+1)$ -ой по n -ую:

$$\hat{a}_{rc} := \hat{a}_{rc} - \frac{\hat{a}_{rp}}{\hat{a}_{pp}} \hat{a}_{pc}, \quad r = \overline{p+1, n}, \quad c = \overline{p+1, n+1}.$$

Шаг n . Если необходимо, поменять $(n-1)$ -ую строчку расширенной матрицы с последней строкой так, чтобы $\hat{a}_{n-1, n-1} \neq 0$. Затем исключить переменную x_{n-1} из последней строки.

Шаг $n+1$. Решить получившуюся треугольную систему известным методом.

Замечание. Если $\hat{a}_{pp} = 0$, то этот ведущий элемент не позволит исключить переменную в p -ом столбце ниже главной диагонали. Поэтому необходима проверка в алгоритме выше и, в случае необходимости, перемена местами двух строк матрицы \hat{A} .

Замечание. Выбор ведущего элемента также влияет на точность вычислений с конечной разрядной сеткой. Следующий пример показывает, как простейшая стратегия выбора ведущего элемента приводит к значительной погрешности решения.

Пример. Решить методом Гаусса, используя арифметику с четырьмя значащими цифрами, следующую систему, имеющую точное решение $x_1 = 1, x_2 = 1$:

$$\begin{cases} 1.133x_1 + 5.281x_2 = 6.414 \\ 24.14x_1 - 1.210x_2 = 22.93 \end{cases}$$

(Ответ: $x_1 = 0.9956, x_2 = 1.001$.)

Сравните этот результат с решением тем же методом в тех же условиях следующей системы:

$$\begin{cases} 24.14x_1 - 1.210x_2 = 22.93 \\ 1.133x_1 + 5.281x_2 = 6.414 \end{cases}$$

(Ответ: $x_1 = 1.000, x_2 = 1.000$.)

Чтобы исправить ситуацию, часто используют следующий простой способ, называемый частным выбором ведущего элемента. При исключении p -ой переменной сравнивают между собой элементы столбца под номером p , лежащие под главной диагональю и выбирают k -ую строчку так, что

$$|a_{kp}| = \max \{ |a_{pp}|, |a_{p+1p}|, \dots, |a_{np}| \}.$$

Если $k > p$, то k -ую и p -ую строчки меняют местами.

Другая стратегия носит название масштабированного выбора ведущего элемента. При этом вначале для строчек с r -ой по n -ую находят максимальный в строке элемент:

$$s_r = \max \{ |a_{rp}|, |a_{rp+1}|, \dots, |a_{rn}| \}, \quad r = \overline{p, n}.$$

Ведущая строка определяется следующим образом:

$$\frac{|a_{kp}|}{s_r} = \max \left\{ \frac{|a_{pp}|}{s_p}, \frac{|a_{p+1p}|}{s_{p+1}}, \dots, \frac{|a_{np}|}{s_n} \right\}.$$

Задача 2.2.1. Найти точное решение системы $Ax = b$, где A — матрица Гильберта 4×4 :

$$\begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \end{pmatrix}$$

а столбец правых частей $b = (1, 0, 0, 0)^T$. Вычисления проводите вручную, сохраняя все выражения в виде дробей.

Решите ту же систему в арифметике с четырьмя значащими цифрами (в первую очередь надо записать приближенную матрицу Гильберта). Сравните результаты и объясните их.

Задача 2.2.2. Решить систему уравнений $Ax = b$, где A — матрица $n \times n$ с элементами $a_{ij} = i^{j-1}$, а b — вектор $n \times 1$ с элементами $b_1 = n, b_i = \frac{i^n - 1}{i - 1}, i = \overline{2, n}$. Точное решение ее — вектор $x^* = (1, 1, \dots, 1)^T$. Рассмотрите случаи $n = 3, 7$ и 11 . Объясните результаты.

2.3 Итерационные методы решения СЛАУ

Напомним идею простейших итерационных методов, рассмотрев пример.

Пример. Пусть необходимо решить систему

$$\begin{cases} 4x_1 - x_2 + x_3 = 7 \\ 4x_1 - 8x_2 + x_3 = -21 \\ -2x_1 + x_2 + 5x_3 = 15 \end{cases}$$

Выразим из каждого уравнения по одной переменной:

$$\begin{cases} x_1 = \frac{7 + x_2 - x_3}{4} \\ x_2 = \frac{21 + 4x_1 + x_3}{8} \\ x_3 = \frac{15 + 2x_1 - x_2}{5} \end{cases}.$$

Представим, что мы знаем приближенное решение системы уравнений, вектор $x^0 = (x_1^0, x_2^0, x_3^0)^T$. Подставив эти значения в последние соотношения, мы получим некоторый новый вектор x_1 . Его мы снова подставим в формулы и т.д. При определенных условиях (о них — позже) построенная последовательность векторов будет сходиться (по норме) к точному решению системы. Пусть мы начинаем с $x^0 = (1, 2, 2)^T$. Проведем вычисления и сохраняя девять цифр, на 19-ой итерации этого процесса получим точное решение $x^* = (2, 4, 3)^T$.

Рассмотренный метод решения носит название метода итераций Якоби или метода простых итераций. Более формально его можно сформулировать так. Пусть уже известно k -ое приближение решения $p^k = (x_1^k, x_2^k, \dots, x_n^k)$. Координаты следующего приближения вычисляются поочередно из уравнений системы по формулам:

$$x_j^{k+1} = \frac{b_j - a_{j1}x_1^k - \dots - a_{jj-1}x_{j-1}^k - a_{jj+1}x_{j+1}^k - \dots - a_{jn}x_n^k}{a_{jj}}, \quad j = \overline{1, n}.$$

Этот процесс останавливают, когда норма разницы между p^k и p^{k+1} становится меньше заданной малой величины или когда достигнут установленный предел количества итераций.

Метод итераций Якоби сходится не всегда, как это легко можно увидеть из примера.

Пример. Если в предыдущем примере переставить местами первую и третью строки (это одно из элементарных преобразований, так что решение системы останется тем же), то в процессе итераций норма p^k неограниченно растет, так что последовательность приближений не сходится вовсе.

Чтобы сформулировать условия сходимости метода итераций Якоби, необходимо ввести новое понятие.

Определение. Матрица A размерности $n \times n$ называется строго доминированной, если

$$|a_{kk}| > \sum_{j=1, j \neq k}^n |a_{kj}|, \quad k = \overline{1, n}.$$

Теорема 2.3 Пусть A — строго доминированная матрица. Тогда система $Ax = b$ имеет единственное решение $x = p$ и последовательность векторов $\{p^k\}$ в методе итераций Якоби сходится к точному решению при любом начальном приближении p_0 .

Нетрудно заметить, что при вычислении нового приближения мы использовали информацию только предыдущей итерации. В то же самое время, когда мы вычисляем x_j^{k+1} , нам уже известны первые $j - 1$ координаты нового приближения. Метод решения СЛАУ, который использует эту информацию, называется методом Гаусса-Зейделя. Его расчетная формула имеет вид:

$$x_j^{k+1} = \frac{b_j - a_{j1}x_1^{k+1} - \dots - a_{jj-1}x_{j-1}^{k+1} - a_{jj+1}x_{j+1}^k - \dots - a_{jn}x_n^k}{a_{jj}}, \quad j = \overline{1, n}.$$

Можно показать, что метод Гаусса-Зейделя сходится при тех же условиях, что и метод итераций Якоби. Численные эксперименты показывают, что его скорость сходимости, как правило, выше, однако это не всегда так.

Задача 2.3.1. Реализовать на языке программирования \mathcal{N} алгоритм обоих методов. Программа должна проверять условие теоремы сходимости.

Задача 2.3.2. Привести примеры систем, которые быстрее решаются методом итераций Якоби, чем методом Гаусса-Зейделя при одном и том же начальном приближении. Порядок систем не меньше чем 4×4 .

Задача 2.3.3. Решить методом Гаусса-Зейделя и методом итераций Якоби следующие системы, взяв за начальное приближение нулевой вектор. Для каждой системы построить график значений $\|p^{k+1} - p^k\|$ и $\|p^k - p^*\|$, где p^* — точное решение системы:

$$\begin{cases} 5x_1 - x_2 + x_3 = 10 \\ 2x_1 + 8x_2 - x_3 = 11 \\ -x_1 + x_2 + 4x_3 = 3 \end{cases} \quad \begin{cases} 2x_1 + 8x_2 - x_3 = 11 \\ 5x_1 - x_2 + x_3 = 10 \\ -x_1 + x_2 + 4x_3 = 3 \end{cases}$$

$$\begin{cases} x_1 - 5x_2 - x_3 = -8 \\ 4x_1 + x_2 - x_3 = 13 \\ 2x_1 - x_2 - 6x_3 = -2 \end{cases} \quad \begin{cases} 4x_1 + x_2 - x_3 = 13 \\ x_1 - 5x_2 - x_3 = -8 \\ 2x_1 - x_2 - 6x_3 = -2 \end{cases}$$

Задача 2.3.4. Следующую систему решить методом Гаусса-Зейделя и методом итераций Якоби при нескольких различных начальных приближениях. Проанализируйте сходимость обоих методов. Является ли условие строгой доминированности матрицы системы необходимым?

$$\begin{cases} x_1 & + & x_3 & = & 2 \\ -x_1 + x_2 & & & = & 0 \\ x_1 + 2x_2 - 3x_3 & = & 0 \end{cases}$$

Задача 2.3.5. Пусть имеется система уравнений следующего вида

$$\begin{cases} d_1x_1 & + & c_1x_2 & & & & & & & & = & b_1 \\ a_1x_1 & + & d_2x_2 & + & c_2x_3 & & & & & & = & b_2 \\ & & a_2x_2 & + & d_3x_3 & + & c_3x_4 & & & & = & b_3 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \\ & & & & & & a_{n-1}x_{n-1} & + & d_nx_n & = & b_n \end{cases}$$

Напишите и реализуйте на языке \mathcal{N} итерационный алгоритм решения этой системы, эффективно использующий разреженность матрицы системы.

Решите две системы, используя этот алгоритм:

$$\left\{ \begin{array}{cccccccc} 4x_1 & + & x_2 & & & & & = 3 \\ x_1 & + & 4x_2 & + & x_3 & & & = 3 \\ & & x_2 & + & 4x_3 & + & x_4 & = 3 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ & & x_{48} & + & 4x_{49} & + & x_{50} & = 3 \\ & & & & x_{49} & + & 4x_{50} & = 3 \end{array} \right. \quad \text{и} \quad \left\{ \begin{array}{cccccccc} 4x_1 & + & x_2 & & & & & = 1 \\ x_1 & + & 4x_2 & + & x_3 & & & = 2 \\ & & x_2 & + & 4x_3 & + & x_4 & = 1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ & & x_{48} & + & 4x_{49} & + & x_{50} & = 1 \\ & & & & x_{49} & + & 4x_{50} & = 2 \end{array} \right.$$

Задача 2.3.6. Напишите варианты обоих алгоритмов, в которых критерием останова является не достаточно малая норма разницы $\|p^{k+1} - p^k\|$, а малость невязки $\|Ap^k - b\|$. Сравните число итераций этого и первоначальных алгоритмов.