

МОДЕЛИРОВАНИЕ И СПОСОБЫ РЕШЕНИЯ СИСТЕМ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ

В данной статье дан сравнительный анализ существующих методов решения СЛАУ. Разработана программа, которая реализует методов Гаусса и Зейделя для решения СЛАУ.

Ключевые слова: линейные алгебраические уравнения, моделирование, способы решения, арифметическое действие.

MODELING AND METHODS OF SOLVING SYSTEMS OF LINEAR ALGEBRAIC EQUATION

This article presents a comparative analysis of existing methods for solving linear systems. The program, which implements the Gauss-Seidel methods and for solving linear systems.

Keywords: linear algebraic equations, modeling, solution processes, arithmetic operation.

Введение. Решение систем линейных алгебраических уравнений (СЛАУ) является одной из основных задач линейной алгебры. Эта задача имеет важное прикладное значение при решении научных и технических проблем. Кроме того, является вспомогательной при реализации многих алгоритмов вычислительной математики, математической физики, обработки результатов экспериментальных исследований.

Применяемые на практике численные методы решения СЛАУ делятся на две группы - *прямые и итерационные*.

В *прямых* (или точных) методах решение системы получают за конечное число арифметических действий. К ним относятся известное правило Крамера, нахождения решения с помощью определителей, метод последовательного исключения неизвестных (метод Гаусса) и его модификации, метод прогонки и другие. Сопоставление различных прямых методов проводится обычно по числу арифметических действий, необходимых для получения решения. Прямые методы являются универсальными и применяются для решения систем до порядка 10^3 . Отметим, что вследствие погрешностей округления при решении задач на ЭВМ прямые методы на самом деле не приводят к точному решению системы.

Итерационные (или приближенные) методы являются бесконечными и находят решение системы как предел при $k \rightarrow \infty$ последовательных приближений $x^{(k)}$, где k - номер итерации. Обычно задается точность и вычисления итерации проводятся до тех пор, пока не будет выполнена оценка $|x^k - x^{k-1}| < \varepsilon$. Число итераций n , которое необходимо провести для получения заданной точности, для многих методов можно найти из теоретических рассуждений. Качество различных итерационных методов можно сравнивать по необходимому числу итераций n . Эти методы особенно предпочтительны для систем с матрицами специального вида - симметричными, трехдиагональными, ленточными и большими разреженными матрицами.

1. Анализ существующих методов решения задачи

Прямые методы решения СЛАУ. К *прямым* (или точным) методам решения СЛАУ относятся алгоритмы, которые в предположении, что вычисления ведутся без округлений, позволяют получить точное решение системы за конечное число арифметических действий.

Чаще всего решение задач такими методами осуществляется поэтапно: на первом этапе систему преобразуют к тому или иному простому виду, на втором - решают упрощенную систему и получают значения неизвестных.

Запишем систему линейных алгебраических уравнений в развернутом виде:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \dots + a_{2n}x_n = b_2 \\ \dots \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \dots + a_{nn}x_n = b_n, \end{cases}$$

где x_1, x_2, \dots, x_n - неизвестные величины, b_1, b_2, \dots, b_n - элементы правой части. Если определитель системы отличен от нуля, то она имеет единственное решение. Для удобства дальнейших преобразований обозначим элементы правой части $a_{i(n+1)}$ и запишем расширенную матрицу размерами $n \times (n+1)$, которая содержит всю информацию о системе:

$$A = \left[\begin{array}{cccc|c} a_{11} & a_{12} & a_{13} & \dots & a_{1n} & a_{1(n+1)} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} & a_{2(n+1)} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} & a_{n(n+1)} \end{array} \right].$$

С этой матрицей можно обращаться так же, как и с системой - переставлять строки, прибавлять кратное одной строки к другой, исключая неизвестные и приводя матрицу к треугольному или диагональному виду.

Приведем формальное описание схем некоторых прямых методов.

Метод Гаусса (схема единственного деления). Алгоритм метода состоит из двух этапов. Первый этап называется *прямым ходом* метода и заключается в последовательном исключении неизвестных из уравнений, т.е. в приведении матрицы A к верхнему треугольному виду (ниже главной диагонали все нули). Для этого на первых шагах разделим первое уравнение системы на a_{11} (предположим, что коэффициент $a_{11} \neq 0$, в противном случае осуществляем перестановку уравнений системы). Обозначим коэффициенты полученного *приведенного* уравнения $U_{1j}^{(1)} = a_{1j} / a_{11}$, ($j = 2, \dots, n+1$), домножим его на коэффициент a_{21} и вычтем из второго уравнения системы, исключая тем самым x_1 из второго уравнения (обнуляя коэффициент a_{21} матрицы). Поступим аналогично с остальными уравнениями и получим новую систему, матрица которой в первом столбце, кроме первого элемента, содержит только нули, т.е.

$$\left[\begin{array}{cccc|c} 1 & U_{12}^{(1)} & U_{13}^{(1)} & \dots & U_{1n}^{(1)} & U_{1(n+1)}^{(1)} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & \dots & a_{2n}^{(1)} & a_{2(n+1)}^{(1)} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & a_{n2}^{(1)} & a_{n3}^{(1)} & \dots & a_{nn}^{(1)} & a_{n(n+1)}^{(1)} \end{array} \right].$$

Первое уравнение в дальнейших преобразованиях не участвует. Описанный выше процесс исключения неизвестных применим к матрице $[a_{ij}^{(1)}]$ размерами $(n-1) \times n$. После k аналогичных шагов получим k приведенных уравнений с коэффициентами

$$U_{lj}^{(l)} = a_{lj}^{(l-1)} / a_{ll}^{(l-1)}, \quad (l = 2, \dots, k, \quad j = k+1, \dots, n+1)$$

и матрицу $[a_{ij}^{(k)}]$ размерами $(n-k) \times (n-k+1)$, элементы которой вычисляются по

формулам

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} - a_{ik}^{(k-1)} \times U_{kj}^{(k)}, \quad (i = k+1, \dots, n, \quad j = k+1, \dots, n+1).$$

Элементы $a_{kk}^{(k-1)}$, на которые осуществляется деление, называются *ведущими*

элементами метода Гаусса и не должны равняться нулю. Прямой ход метода Гаусса заканчивается после n шагов определением $U_{n(n+1)}^{(n)}$.

Обратный ход метода Гаусса заключается в последовательном определении компонент решения, начиная с x_n и заканчивая x_1 , по следующим формулам:

$$x_n = U_{n(n+1)}^{(n)}, \quad x_k = U_{k(n+1)}^{(k)} - \sum_{i=k+1}^n U_{ki}^{(k)} x_i, \quad k = n-1, \dots, 1.$$

Метод Гаусса с выбором главного элемента. Метод заключается в том, что при прямом ходе в алгоритме метода Гаусса на каждом шаге исключения производится выбор наибольшего по модулю элемента в качестве *ведущего*. Этого достигают перестановкой строк или столбцов матрицы коэффициентов. Наиболее распространённой в вычислительной практике является стратегия выбора *главного элемента столбца* - нахождение максимального по модулю элемента k -го столбца матрицы $[a_{ij}^{(k)}]$ и использование его в качестве ведущего элемента на k -м шаге исключения.

В этом случае для невырожденных систем гарантируется, что ведущие элементы не равны нулю, и уменьшается погрешность при делении и последующем вычитании при преобразованиях. Рекомендуется также масштабировать предварительно каждое уравнение исходной системы, разделив на его наибольший по абсолютной величине коэффициент. Это делает рост элементов промежуточных матриц ограниченным.

Метод оптимального исключения. В целях экономии оперативной памяти (примерно в 4 раза) операции прямого и обратного хода метода Гаусса выполняются попеременно. На первом шаге после приведения первого уравнения исключается неизвестное x_1 из второго уравнения, а затем с помощью приведенного второго уравнения - неизвестное x_2 из первого. После $(k-1)$ таких шагов матрица системы имеет вид

$$\left[\begin{array}{cccc|ccc} 1 & \dots & 0 & a_{1(k+1)}^{(k)} & \dots & a_{1n}^{(k)} & a_{1(n+1)}^{(k)} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 1 & a_{k(k+1)}^{(k)} & \dots & a_{kn}^{(k)} & a_{k(n+1)}^{(k)} \\ a_{(k+1)1}^{(k)} & \dots & a_{(k+1)k}^{(k)} & a_{(k+1)(k+1)}^{(k)} & \dots & a_{(k+1)n}^{(k)} & a_{(k+1)(n+1)}^{(k)} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n1}^{(k)} & \dots & a_{nk}^{(k)} & a_{n(k+1)}^{(k)} & \dots & a_{nn}^{(k)} & a_{n(n+1)}^{(k)} \end{array} \right].$$

На k -м шаге, используя первые k уравнений, исключаем неизвестные x_1, \dots, x_k из $(k+1)$ -го уравнения. Затем посредством этого уравнения исключается неизвестное x_{k+1} из первых k уравнений и т.д. В результате прямого хода матрица системы приводится к диагональному виду с единицами на главной диагонали. При этом отпадает необходимость обратного хода, поскольку столбец правой части приведенной матрицы $[a_{i(n+1)}^{(n)}]$ и является вектором решения.

Метод простых итераций (Якоби).

Для решения итерационным методом система линейных алгебраических уравнений $Ax = b$ должна быть приведена к виду $x = Gx + f$, где G - некоторая матрица, f - преобразованный вектор свободных членов. Затем выбирается начальное приближение - произвольный вектор $x^{(0)}$ - и строится рекуррентная последовательность векторов $x^{(1)}, x^{(2)}, \dots, x^{(k)}, \dots$ по формуле

$$x^{(k)} = Gx^{(k-1)} + f.$$

Для сходимости этой последовательности при любом начальном приближении *необходимо и достаточно*, чтобы все собственные значения матрицы G были по абсолютной величине меньше единицы. На практике это трудно проверить, и обычно пользуются *достаточными условиями сходимости* - итерации сходятся, если какая-нибудь норма матрицы меньше единицы, т.е.

$$\|G\|_1 = \max_{1 \leq i \leq n} \sum_{j=1}^n |g_{ij}| = \alpha < 1 \quad \text{или} \quad \|G\|_2 = \max_{1 \leq j \leq n} \sum_{i=1}^n |g_{ij}| = \beta < 1.$$

Чем меньше норма матрицы G , тем быстрее сходится итерационный процесс.

Преобразование системы можно осуществить, просто решая каждое i -е уравнение относительно x_i :

$$x_i = -\frac{1}{a_{ii}} \left[\sum_{j=1, j \neq i}^n a_{ij} x_j - b_i \right].$$

Метод Якоби использует следующий алгоритм построения приближений:

$$x_i^{(k)} = -\frac{1}{a_{ii}} \left[\sum_{j=1, j \neq i}^n a_{ij} x_j^{(k-1)} - b_i \right].$$

Если A - матрица с доминирующей диагональю, т.е. $|a_{ii}| > \sum_{j \neq i}^n |a_{ij}|$, то метод Якоби

сходится при любом начальном приближении $x^{(0)}$.

Метод Якоби относится к *одношаговым итерационным методам*, когда для нахождения $x^{(k+1)}$ требуется помнить только одну предыдущую итерацию $x^{(k)}$. Для исследования сходимости удобнее записывать итерационные методы не в координатной, а в матричной форме, придерживаясь стандартной формы записи итерационных методов.

Канонической формой одношагового итерационного метода решения СЛАУ называется его запись в виде

$$B_{k+1} \frac{x^{(k+1)} - x^{(k)}}{\tau_{k+1}} + Ax^{(k)} = b,$$

где B_{k+1} - матрица, задающая тот или иной итерационный метод, τ_{k+1} - итерационный параметр. Числовые параметры τ_k вводят для ускорения сходимости. Способ выбора итерационных параметров определяется при исследовании сходимости метода, когда выясняется при каких значениях параметров метод сходится и когда сходимость будет наиболее быстрой (соответствующие параметры называются оптимальными).

Итерационный метод называют *явным*, если B_{k+1} - единичная матрица. *Неявные* итерационные методы имеет смысл применять лишь в том случае, когда решение системы уравнений с матрицей B_k требует меньше машинной памяти или времени или алгоритмически проще, чем решение исходной системы.

Методом простой итерации называют явный метод с постоянным параметром

$$\frac{x^{(k+1)} - x^{(k)}}{\tau} + Ax^{(k)} = b, \quad \text{или} \quad x^{(k+1)} = x^{(k)} - \tau \cdot r^{(k)},$$

где $r^{(k)} = Ax^{(k)} - b$ - вектор невязки. Метод сходится для симметричных положительно определенных матриц при $0 < \tau < \frac{2}{\|A\|}$.

Для *окончания итерационного процесса* используют три способа. При первом определяют величину стабилизации и прекращают вычисления, если она меньше \mathfrak{M} , т.е.

$$\frac{\|x^{(k+1)} - x^{(k)}\|}{\max\{\|x^{(k)}\|, \|x^{(k+1)}\|\}} \leq \varepsilon.$$

Недостатком этого способа является то, что при медленно сходящихся итерациях величина стабилизации может быть малой, хотя приближенное решение сильно отличается от точного решения.

При втором способе вычисляют нормы невязки до начала итераций и на каждой

итерации. Итерации прекращают при выполнении неравенства $\frac{\|Ax^{(k)} - b\|}{\|Ax^{(0)} - b\|} \leq \varepsilon$.

При третьем способе предварительно оценивается число итераций, необходимое для получения заданной точности \mathbb{M} . Если для погрешности итерационного метода выполняются оценки

$$\|x^{(k)} - x\| \leq q^k \|x^{(0)} - x\|,$$

где $q \in (0,1)$, то метод сходится со скоростью геометрической прогрессии со знаменателем q . Можно определить, потребовав, чтобы $q^n < \mathbb{M}$, число итераций n , достаточное для того, чтобы начальная погрешность уменьшилась в заданное число раз:

$$n \geq n_0(\varepsilon) = \frac{\ln(1/\varepsilon)}{\ln(1/q)}.$$

Целая часть числа n_0 является *минимальным числом итераций, необходимым для получения заданной точности*.

Величина $\ln(1/q)$ является *скоростью сходимости итерационного метода*.

2. Описание используемого метода

Для решения **методом Зейделя** система линейных алгебраических уравнений $Ax = b$ должна быть приведена к виду $x = Gx + f$, где G - некоторая матрица, f - преобразованный вектор свободных членов. Затем выбирается начальное приближение - произвольный вектор $x^{(0)}$ - и строится рекуррентная последовательность векторов $x^{(1)}, x^{(2)}, \dots, x^{(k)}, \dots$ по формуле

$$x^{(k)} = Gx^{(k-1)} + f.$$

Для сходимости этой последовательности при любом начальном приближении *необходимо и достаточно*, чтобы все собственные значения матрицы G были по абсолютной величине меньше единицы. На практике это трудно проверить, и обычно пользуются *достаточными условиями сходимости* - итерации сходятся, если какая-нибудь норма матрицы меньше единицы, т.е.

$$\|G\|_1 = \max_{1 \leq i \leq n} \sum_{j=1}^n |g_{ij}| = \alpha < 1 \quad \text{или} \quad \|G\|_2 = \max_{1 \leq j \leq n} \sum_{i=1}^n |g_{ij}| = \beta < 1.$$

Чем меньше норма матрицы G , тем быстрее сходится итерационный процесс.

Преобразование системы можно осуществить, просто решая каждое i -е уравнение относительно x_i :

$$x_i = -\frac{1}{a_{ii}} \left[\sum_{j=1, j \neq i}^n a_{ij} x_j - b_i \right].$$

Метод Зейделя использует следующий алгоритм построения приближений:

$$x_i^{(k)} = -\frac{1}{a_{ii}} \left[\sum_{j=1}^{i-1} a_{ij} x_j^{(k)} + \sum_{j=i+1}^n a_{ij} x_j^{(k-1)} - b_i \right]$$

Если A - матрица с доминирующей диагональю, т.е. $|a_{ii}| > \sum_{j \neq i}^n |a_{ij}|$, то метод Зейделя

сходится при любом начальном приближении $x^{(0)}$.

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

Метод Зейделя сходится примерно так же, как геометрическая прогрессия со знаменателем $\|G\|$. Если норма матрицы G близка к 1, то скорость сходимости очень медленная. Для ускорения сходимости используется *метод релаксации*. Суть его в том, что полученное по методу Зейделя очередное значение пересчитывается по формуле:

Здесь $0 < \omega \leq 2$ - параметр релаксации. Если $\omega < 1$ - *нижняя релаксация*, если $\omega > 1$ - *верхняя релаксация*. Параметр ω подбирают так, чтобы сходимость метода достигалась за

минимальное число итераций.

Метод Зейделя является *одношаговым итерационным методом*, когда для нахождения $x^{(k+1)}$ требуется помнить только одну предыдущую итерацию $x^{(k)}$.

Погрешность итерации вычисляется по формуле:

$$\delta = \max_{1 \leq i \leq n} |x_i^k - x_i^{k-1}|$$

n - порядок матрицы A.

Если δ меньше заданной точности ε , то итерационный процесс прекращают.

Элементы главной диагонали называются главными. Заметим, что если в ходе расчётов по данному алгоритму на главной диагонали окажется нулевой элемент, то произойдет сбой программы. Для того, чтобы избежать этого, следует перестановку строк таким образом, чтобы на главной диагонали находились максимальные элементы строк. Т. е., если в k-й строке максимальным является i-й элемент, необходимо поменять местами k-ю и i-ю строки, и поменять местами соответствующие элементы вектора b. Такой выбор главного элемента необходим для сходимости итерационного процесса.

3. Анализ результатов

Скорость сходимости итерационного процесса зависит от заданной матрицы коэффициентов. В зависимости от вида исходных данных (матрицы коэффициентов и матрицы b) программа подбирает оптимальный параметр релаксации ω (при котором решение достигается за минимальное число итераций).

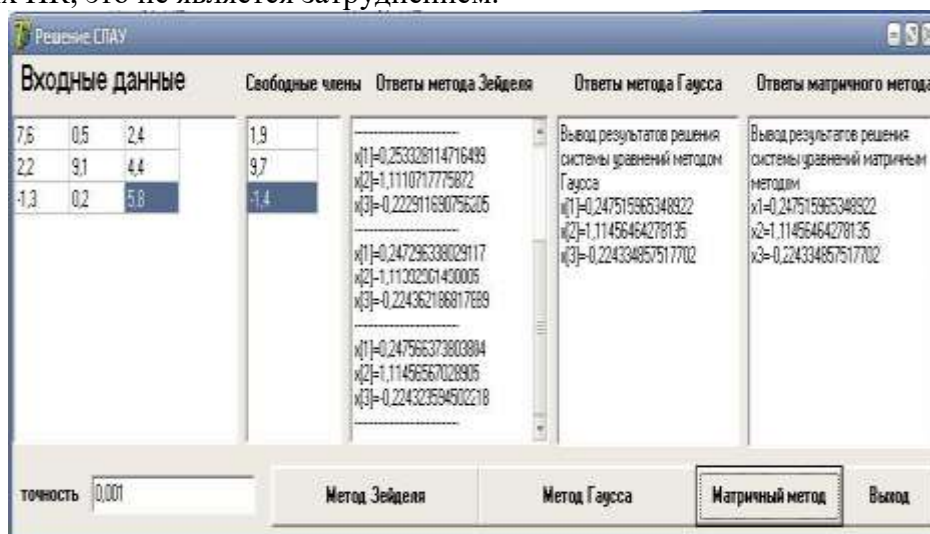
Для достижения наивысшей скорости сходимости итерационного процесса для уравнения, заданного программой был выбран параметр релаксации $\omega=1,26$. Таким образом, была применена верхняя релаксация. Заданная точность $\varepsilon=0,0001$ была достигнута за 40 итераций.

Для достижения наивысшей скорости сходимости итерационного процесса для уравнения, заданного программой был выбран параметр релаксации $\omega=0,98$. Таким образом, была применена нижняя релаксация. Заданная точность $\varepsilon=0,0001$ была достигнута за 17 итераций.

Правильность решения СЛАУ была проверена с помощью программного пакета Delphi. Отметим, что программа даёт правильное решение СЛАУ почти во всех случаях, когда каждый элемент главной диагонали является максимальным в своей строке.

Программа проста в эксплуатации и нетребовательна к ресурсам. Реализованная в современной среде разработки Delphi, она без труда может быть доработана или исправлена.

Недостатки программы: 1) применима не для всех систем линейных уравнений; 2) оптимальный параметр релаксации ω вычисляется методом подбора, и, поэтому, количество итераций, требуемое для его отыскания достаточно велико (около 18000), однако, для современных ПК, это не является затруднением.



Литература:

1. Волков Е.А. Численные методы. - М.: Наука, 1987. - 254 с.
2. Калиткин Н.Н. Численные методы. - М.: Наука, 1978. - 512 с.
3. Мудров А.Е. Численные методы для ПЭВМ на языках БЕЙСИК, ФОРТРАН и ПАСКАЛЬ.
4. Самарский А.А., Гулин А.В. Численные методы. - М.: Наука, 1989. -432 с.
5. Кэнту М. Delphi 4 для профессионалов. -СПб: «Питер», 1999.- 1200 с.