РЕШЕНИЕ СИСТЕМ ЛИНЕЙНЫХ УРАВНЕНИЙ В КОММУТАТИВНЫХ ОБЛАСТЯХ *

© Г.И. Малашонок

Malaschonok G.I. Solving systems of linear equations in commutative domains. There are considered fast methods of linear systems solving in commutative domains and in their fields of fractions. There is suggested the realization of these methods for the rings of principal ideals with the help of Hensel lifting techniques. The complexity of the deterministic algorithm to find all the rational solutions for linear systems with coefficient matrix A of size $n \times m$ and rank n with p-adic lifting is $O(n^2 m(m-n+1)(\log n + \log \|A\| + \log p)^2)$ bit operations. The randomized algorithm to find all the Diophantine solutions for this system has essentially the same complexity.

*Вестник ТГУ, Т.5, N.1, 2000, 147–154.

ВВЕДЕНИЕ

Пусть $\mathbf{R}$ – коммутативная область с единицей, $\mathbf{K}$ – поле частных $\mathbf{R}$. Одной из важных задач вычислительной коммутативной алгебры является задача решения систем линейных уравнений $Ax = b$, коэффициенты которых берутся из коммутативной области R: $A \in R^{n\times m}$, $b \in R^n$.

При этом могут ставиться разные задачи: (1^o) найти одно решение в $\mathbf{K}^m$, (2^o) найти все решения в $\mathbf{K}^m$, (3^o) найти одно решение в $\mathbf{R}^m$, (4^o) найти все решения $\mathbf{R}^m$. Рассмотрим каждую из этих задач для разных областей.

Область $\mathbf{R}$ *является полем.* Основным вычислительным методом здесь является метод исключения Гаусса и его модификации со сложностью $O(n^2m)$ операций в поле $\mathbf{R}$. Быстрые методы решения систем линейных уравнений над полем ведут свое начало от работы Штрассена [1] и имеют такую же сложность, как и матричное умножение – $O(n^{\beta-1}m)$. $O(n^\beta)$ – это число операций, необходимое для умножения квадратных матриц n-го порядка. Для обычного умножения матриц $\beta = 3$, для алгоритма Штрассена $\beta = \log_2 7$, а для лучшего на сегодня алгоритма $\beta < 2.376$ [2].

Произвольная коммутативная область $\mathbf{R}$ *с единицей.* Оценку алгоритмов будем производить по количеству операций над элементами $\mathbf{R}$. При этом предполагаем, что в $\mathbf{R}$ существует алгоритм сокращения, то есть если $a = bc$ и известны a и c, $c \neq 0$, то существует алгоритм вычисления b. Основным вычислительным методом здесь является метод Доджсона, иногда называемый методом Гаусса с точным делением, со сложностью $O(n^2m)$ операций в $\mathbf{R}$. Впервые такой метод был опубликован в работе Доджсона [3], а его улучшения получены в работах [4], [5], [6]. Быстрый метод решения систем линейных уравнений над коммутативной областью опубликован в работе [7]. Сложность этого метода $O(n^{\beta-1}m)$ такая же, как и сложность матричного умножения. И теперь дальнейший прогресс в задачах (1^o) и (2^o) целиком связан с улучшениями алгоритма матричного умножения.

Вероятностный алгоритм для решения задач (3^o) и (4^o) при самых общих предположениях относительно коммутативной области обсуждается в настоящей работе. Предполагается, что в области $\mathbf{R}$ имеется алгоритм, который может установить, является ли конечнопорожденный идеал $I = (a_1, a_2, \ldots, a_s)$ единичным. Если I является единичным, то этот алгоритм вычисляет коэффициенты $k_i \in \mathbf{R}$ в разложении единицы $1 = \sum_{i=1}^{s} k_i a_i$. Например, достаточно иметь алгоритм вычисления в области $\mathbf{R}$ базиса Гребнера.

Коммутативная область главных идеалов $\mathbf{R}$. Это основная прикладная область. Лучшим методом для задачи (1^o) в случае решения определенных систем в поле рациональных чисел ($\mathbf{R} = \mathbf{Z}$, $m = n$, $\det A \neq 0$) остается метод Диксона [9], использующий линейный p-адический подъем. Его сложность $O(n^3(\log n + \log\|A\| + \log p)^2)$ бит-операций.

Если применять метод Гаусса с точным делений для $\mathbf{R} = \mathbf{Z}$, то для решения задачи (2^o) в обычной арифметике потребуется $O(mn^4(\log n + \log\|A\|)^2)$ бит-операций. $\|A\|$ обозначает модуль наибольшего из коэффициентов системы. Если применить китайскую теорему об остатках, то сложность можно уменьшить до величины $O(mn^3(\log n + \log\|A\|)^2)$ бит-операций [8].

Мой подход к решению задачи (2^o) в области главных идеалов состоит в применении p-адического подьема, как и в методе Диксона [9]. Предлагаемый в настоящей работе метод имеет сложность $O(n^2 m(m-n+1)(\log m + \log \|A\| + \log p)^2)$ бит-операций.

В работе [10] предлагается вероятностный алгоритм для задачи (3^o) – получения одного решения в кольце целых чисел $\mathbf{Z}$ и кольце $F[x]$ многочленов над полем F, который опирается на алгоритм Диксона.

Я предлагаю вероятностный алгоритм для задачи (4^o), который основан на p-адическом подьеме и сложность которого имеет существенно кубическую зависимость от порядка системы, как и [10], но количество вычислений обратных матриц становится в $m-n+1$ раз меньше.

Общий план всей работы следующий. Во втором параграфе приводится обоснование метода нахождения всех решений в поле частных и соответствующий алгоритм для коммутативной области с сокращением. В третьем параграфе приводится алгоритм решения систем линейных уравнений в поле частных с помощью линейного p-адического подъема. В четвертом параграфе обсуждается вероятностный метод получения всех решений системы линейных уравнений в коммутативной области и соответствующий алгоритм. Пятый параграф посвящен оценкам сложности этих алгоритмов. В заключении сравниваются предлагаемые алгоритмы получения точных решений систем линейных уравнений со стандартными приближенными методами решения систем линейных уравнений, использующими представление чисел с плавающей точкой.

РЕШЕНИЕ СИСТЕМЫ В ПОЛЕ ЧАСТНЫХ

Пусть $\mathbf{R}$ – коммутативная область с единицей, $\mathbf{K}$ – поле частных области $\mathbf{R}$, $A \in \mathbf{R}^{n\times m}$, $c \in \mathbf{R}^n$,

$$Ax = c \qquad (1)$$

– система линейных уравнений над $\mathbf{R}$. Пусть $\operatorname{rank} A = r$, S и T – матрицы перестановок, такие, что $\det A_0 \neq 0$, где A_0 верхний левый блок размера $r\times r$ матрицы $SAT = \begin{pmatrix} A_0 & A_1 \\ A_2 & A_3 \end{pmatrix}$ и пусть $Sc = \begin{pmatrix} c_0 \\ c_1 \end{pmatrix}$, $c_0 \in \mathbf{R}^r$. $\mathcal{M} = \{x | x \in \mathbf{K}^m, Ax = c\}$ – множество решений системы (1). Единичную матрицу порядка r будем обозначать I_r, а также обозначим $E_{i,j}$ – квадратные матрицы, у которых только один ненулевой элемент – (i,j), и этот элемент равен единице. Будем пользоваться этими обозначениями на протяжении всей статьи.

Нам потребуются некоторые факты из теории систем линейных уравнений.

1. Если $\operatorname{rank}(A,c) \neq r$, то $\mathcal{M} = \emptyset$. Если $\operatorname{rank}(A,c) = r$, то $\mathcal{M}$ - это гиперплоскость размерности $m-r$ в пространстве размерности m. Она определяется $m-r+1$ точками, которые не лежат в одной гиперплоскости меньшей размерности.

2. Если система (1) однородная ($c = 0$) и $x_1, x_2, \dots, x_{m-r}$ ее линейно независимые решения, то $\mathcal{M} = \{\sum_{i=1}^{m-r+1} x_i u_i | u_i \in \mathbf{K}\}$.

3. Если система (1) неоднородная ($c \neq 0$) и $x_1, x_2, \dots, x_{m-r+1}$ ее линейно независимые решения, то $\mathcal{M} = \{\sum_{i=1}^{m-r+1} x_i u_i | u_i \in \mathbf{K}, \sum_{i=1}^{m-r+1} u_i = 1\}$.

Определение. *Базисным множеством решений однородной системы линейных уравнений* (1) называется множество, состоящее из $m-r$ линейно независимых решений системы (1).
Базисным множеством решений неоднородной системы линейных уравнений (1) называется множество, состоящее из $m-r+1$ линейно независимых решений системы (1).

В следующих двух теоремах задача вычисления базисного множества решений системы (1) сводится к нескольким задачам решения определенных систем. В первой теореме рассматриваются однородные системы, во второй теореме – неоднородные системы.

Теорема 1 *Пусть (1) – однородная система линейных уравнений и $A_1 = (a_1, a_2, \dots, a_{m-r})$, $a_j \in \mathbf{R}^r$. Тогда системы*

$$A_0 x_j = -a_j, \quad j = 1, \dots, m-r \qquad (2)$$

определены, и их решения $x_j \in \mathbf{K}^r$ задают базисное множество решений системы (1):

$$T\begin{pmatrix} x_j \\ e_j \end{pmatrix}, \quad j = 1, \dots, m-r, \qquad (3)$$

где $e_j \in \mathbf{R}^{m-r}$ — это столбцы единичной матрицы $I_{m-r} = (e_1, e_2, \dots, e_{m-r})$.

Доказательство. Обозначим $y = T^{-1}x$. По условию теоремы $(A_0, A_1)y = 0$. Если искать решение в виде $y = \begin{pmatrix} x_j \\ e_j \end{pmatrix}$, то придем к системе (2).

Следствие 1 *Пусть $B = A_0^{-1}A_1 = (b_1, b_2, \dots, b_{m-r})$, $b_j \in \mathbf{K}^r$, тогда $T\begin{pmatrix} -b_1 \\ e_1 \end{pmatrix}$,*

$T\begin{pmatrix} -b_2 \\ e_2 \end{pmatrix}, \ldots, T\begin{pmatrix} -b_{m-r} \\ e_{m-r} \end{pmatrix}$ *базисное множество решений однородной системы линейных уравнений (1).*

Доказательство. В (3) подставим решения систем (2).

Теорема 2 *Пусть (1) – неоднородная система линейных уравнений, P – матрица перестановок, такая, что последний элемент вектора $b = PA_0^{-1}c_0$ не равен* 0. *Пусть $B = PA_0^{-1}A_1$, $J \subset \{1, \ldots, m-r\}$ – номера столбцов матрицы B с нулевыми элементами в последней строке и $U = I_{m-r+1} + \sum_{j \in J} E_{1,j+1}$. Пусть $PA_0P = (A'_0, a_0)$, $(a_0, PA_1)U = (a_0, a_1, a_2, \ldots, a_{m-r})$. Тогда системы*

$$(A'_0, a_j)x_j = Pc_0, \quad j = 0, 1, \ldots, m-r, \qquad (4)$$

определены и их решения $x_j = \begin{pmatrix} x'_j \\ \xi_j \end{pmatrix}$, $x'_j \in \mathbf{K}^{r-1}$, $\xi_j \in \mathbf{K}$ задают базисное множество решений системы (1):

$$TV\begin{pmatrix} x'_j \\ \xi_j e'_j \end{pmatrix} \quad j = 0, 1, \ldots, m-r, \qquad (5)$$

где $e'_j \in \mathbf{R}^{m-r+1}$ — это столбцы единичной матрицы $I_{m-r+1} = (e'_0, e'_1, \ldots, e'_{m-r})$, $Q = \mathrm{diag}(\mathrm{P}, \mathrm{I_{m-r}})$, $W = \mathrm{diag}(\mathrm{I_{r-1}}, \mathrm{U})$, $V = QW$.

Доказательство. Обозначим $y = V^{-1}T^{-1}x$. По условию теоремы $P(A_0, A_1)Vy = Pc_0$ и $P(A_0, A_1)V = (A'_0, a_0, a_1, \ldots, a_{m-r})$. Если искать решение в виде $y = \begin{pmatrix} x'_j \\ \xi_j e'_j \end{pmatrix}$, то придем к системам (4).

Покажем, что системы (4) определены и $\xi_j \neq 0$. Для этого умножим их слева на $PA_0^{-1}P$. Так как $P = P^{-1}$, то получим $PA_0^{-1}P(A'_0, a_j)x_j = PA_0^{-1}c_0$, $j = 0, 1, \ldots, m-r$.

Так как $b = PA_0^{-1}c_0$ и $PA_0^{-1}P(A'_0, a_0) = I_r$, то первая из систем (4) принимает вид $I_r x_0 = b$. Обозначим $I_r = (I', e)$, $e \in \mathbf{R}^r$, тогда $PA_0^{-1}PA'_0 = I'$, $PA_0^{-1}Pa_0 = e$.

Так как по определению $PA_0^{-1}P(a_0, \ldots, a_{m-r}) = PA_0^{-1}P(a_0, PA_1)U = (e, B)U$, то обозначим $d_j = PA_0^{-1}Pa_j$, $j = 1, \ldots, m-r$ столбцы матрицы $(e, B)U$. Так как $U = I_{m-r+1} + \sum_{j \in J} E_{1,j+1}$ и J – номера столбцов матрицы B с нулевыми элементами в последней строке, то последние элементы всех столбцов d_j матрицы $(e, B)U$ не равны нулю. Системы (4) принимают вид

$$(I', d_j)x_j = b, \quad j = 0, 1, \ldots, m-r, \qquad (6)$$

при этом $\det(I', d_j) \neq 0$. Так как у вектора b последний элемент не равен нулю, то решения систем $x_j = \begin{pmatrix} x'_j \\ \xi_j \end{pmatrix}$ таковы, что $\xi_j \neq 0$. Поэтому векторы $\xi_j e'_j$, $j = 1, \ldots, m-r$ линейно независимы, следовательно, и векторы (5) линейно независимы.

Algorithm RationalBasis$[\mathbf{A}, \mathbf{c}]$
Input: $A \in \mathbf{R}^{n \times m}, c \in \mathbf{R}^n$
Output: Решения системы $Ax = c$ или NIL
  if $c = 0$
  then
    $\{B, S, T, r\} := DiagonalFormH(A)$;
    Comment : $B = (b_1, b_2, \ldots, b_{m-r})$
    Comment : $I_{m-r} = (e_1, e_2, \ldots, e_{m-r})$.
    return $\left(T\begin{pmatrix} -b_1 \\ e_1 \end{pmatrix}, \ldots, T\begin{pmatrix} -b_{m-r} \\ e_{m-r} \end{pmatrix}\right)$
  else
    $\{B, b, S, T, P, r\} := DiagonalForm(A, c)$;
    Comment : $B = (b_1, b_2, \ldots, b_{m-r})$,
    Comment : $b = \begin{pmatrix} b' \\ \beta \end{pmatrix}$, $b_j = \begin{pmatrix} b'_j \\ \beta_j \end{pmatrix}$
    Comment : $Q = \mathrm{diag}(\mathrm{P}, \mathrm{I_{m-r}})$,
    Comment : $\mathrm{I_{m-r+1}} = (\mathrm{e'_0}, \mathrm{e'_1}, \ldots, \mathrm{e'_{m-r}})$.
    if $\mathrm{r} < \mathrm{rank(A, c)}$ **then return** NIL;
    else $\mathrm{J} := \emptyset$
      for $\mathrm{j} := 1$ **to** $\mathrm{m - r}$ **do**
        if $\beta_\mathrm{j} = 0$ **then** $\mathrm{J} := \mathrm{J} \cup \{\mathrm{j}\}$;
      for $\mathrm{j} := 1$ **to** $\mathrm{m - r}$ **do**
        if $\mathrm{j} \notin \mathrm{J}$
        then $\xi_\mathrm{j} := \beta / \beta_\mathrm{j}$, $\mathrm{f_j} := \mathrm{e'_j}$
        else $\xi_\mathrm{j} := \beta$, $\mathrm{f_j} := \mathrm{e'_j} + \mathrm{e'_0}$
      return $\left(\mathrm{TQ}\begin{pmatrix} b' \\ \beta e'_0 \end{pmatrix},\right.$
        $\left.\mathrm{TQ}\begin{pmatrix} b' - \xi_j b'_j \\ \xi_j f_j \end{pmatrix}, \ \mathrm{j} = 1, \ldots, \mathrm{m - r}\right)$

Рис. 1. Алгоритм "Рациональный базис"

Следствие 2 *Пусть $B = (b_1, b_2, \ldots, b_{m-r})$, $b = \begin{pmatrix} b' \\ \beta \end{pmatrix}$, $b_j = \begin{pmatrix} b'_j \\ \beta_j \end{pmatrix}$. $\xi_j = \beta / \beta_j$, $f_j = e'_j$ для $j \notin J$. $\xi_j = \beta$, $f_j = e'_j + e'_0$ для $j \in J$. Тогда*

$$TQ\begin{pmatrix} b' \\ \beta e'_0 \end{pmatrix}, \quad TQ\begin{pmatrix} b' - \xi_j b'_j \\ \xi_j f_j \end{pmatrix}, \qquad (7)$$

j=1,…,m-r,

– базисное множество решений неоднородной системы линейных уравнений (1).

Доказательство. В (5) подставим решения систем (6). При этом будем учитывать, что, по построению $d_j = \begin{pmatrix} b'_j \\ \delta_j \end{pmatrix}$, $\delta_j = 1$ при $j \in J$ и $\delta_j = \beta_j$ при $j \notin J$. Затем умножим на матрицу W.

Следствия теорем 1 и 2 позволяют записать алгоритм построения базисного множества решений системы линейных уравнений для произвольной коммутативной области $\mathbf{R}$.

Алгоритм приведен на рис. 1. В этом алгоритме полагается следующее.

$\mathbf{R}$ – коммутативная область с единицей, в которой существует алгоритм сокращения (вычисляющий сомножитель b, по известным a и c, $a = bc, c \neq 0$), (1) – система линейных уравнений над $\mathbf{R}$.

$DiagonalForm(A, c)$ – алгоритм диагонализации матрицы (A, c) для неоднородной системы (1). Он вычисляет $\{B, b, T, S, P, r\}$, при этом $r = \mathrm{rank} A$, а матрицы B, b, T, S, P описаны выше.

$DiagonalFormH(A)$ – алгоритм диагонализации матрицы A, который вычисляет $\{B, T, S, r\}$ для однородной системы (1).

Для этих алгоритмов можно взять любой из методов [3], [4], [5], [6] со сложность $O(n^2 m)$ или метод [7] со сложностью $O(n^{\beta-1} m)$.

РЕШЕНИЕ СИСТЕМЫ С ПОМОЩЬЮ ПОДЪЕМА ПО ГЕНЗЕЛЮ

Теоремы 1 и 2 сводят задачу построения базисного множества решений системы (1) к задаче решения нескольких определенных систем. Для вычисления базисного множества решений неоднородной системы необходимо решить $m - r + 1$ систем с матрицей коэффициентов размера $r \times r$ (для однородных систем нужно решить $m - r$ систем). Для решения определенных систем можно воспользоваться p-адическим подъемом.

Напомним общую схему p-адических методов. Выбирается подходящий простой элемент p кольца $\mathbf{R}$. Для решения системы необходимо, чтобы элемент p не делил определитель матрицы коэффициентов. Например, в кольце $\mathbf{Z}$ этот выбор может оказаться неудачным с вероятностью не более $1/p$. Если проверка покажет, что решение неверное, то выбирается другой простой элемент p. Кольцо вычетов по простому модулю p является полем, и решение в этом поле можно искать, например, методом Гаусса. С помощью неравенства Адамара находится верхняя оценка для числителей и знаменателей решения системы, а по ним находится оценка для величины p^k – границы подъема. Затем решение поднимается по модулю p до p^k, и затем реконструируется рациональное решение. Для решения одной определенной системы используем алгоритм, приведенный

Algorithm RationalBasisH[A]
Input: $A \in \mathbf{R}^{n\times m}$
Output: Решения системы $Ax = 0$.
$p := FirstPrime;$
repeat
  Initialization:
  $\{\tilde{A}_0^{-1}, S, T, r\} := DiagonalFormH[mod_p\ A];$
  if $m = r$ **then return** $\{\emptyset\}$;
  $\begin{pmatrix} A_0 & A_1 \\ A_2 & A_3 \end{pmatrix} := SAT;$
  $(A'_0, a_0) := A_0;$
  $(a_1, \ldots, a_{m-r}) := A_1;$
  $j := 1; Test := TRUE;$
  repeat
    Lifting:
    $l := 1; x := 0; c^* := -a_j;$
    repeat
      $\tilde{x} := mod_p \tilde{A}_0^{-1} c^*;$
      $c^* := (c^* - A_0\tilde{x})/p;$
      $x := x + l\tilde{x};\ l := lp;$
    until $\|l\| \geq L_j^H[A];$
    $\begin{pmatrix} x'_j \\ \xi_j \end{pmatrix} :=$
      $RationalReconstruction[x, N_j^H, D];$
    Test:
    Comment : $(e_1, \ldots, e_{m-r}) = I_{m-r}$
    $x_j := T \begin{pmatrix} x'_j \\ \xi_j e_j \end{pmatrix};$
    if $Ax_j \neq 0$ **then** $Test := FALSE;$
    $j := j + 1;$
  until $j > m - r$ **or** $Test = FALSE;$
  $p := NextPrime[p];$
until $Test = TRUE;$
return $\{x_1, \ldots, x_{m-r}\}$.

Рис. 2. Алгоритм "Вычисление рационального базиса однородной системы линейных уравнений с помощью p-адического подъема".

Диксоном [9] и использующий линейный p-адический подъем.

Алгоритмы построения базисного множества решений системы (1), использующие линейный p-адический подъем, приведены на рис. 2 и 3. Алгоритм для решения однородных систем приведен на рис. 2, алгоритм для решения неоднородных систем — на рис. 3.

В алгоритмах используются следующие функции.

Функция $\|\ \ \|$ имеет следующее значение: $\|\alpha\| = |\alpha|$ для $\alpha \in \mathbf{Z}$, $\|\alpha\| = \deg \alpha$ для $\alpha \in F[x]$, для матрицы A $\|A\| = \max_{\alpha\in A} \|\alpha\|$, для столбца k матрицы A $\|a_k\| = \max_i \|a_{i,k}\|$.

Из неравенства Адамара следуют оценки для числителя и знаменателя решения системы.

Для кольца Z получим:

Algorithm RationalBasis$[\mathbf{A},\mathbf{c}]$
Input: $A \in \mathbf{R}^{n\times m}, c \in \mathbf{R}^n$
Output: Решения системы $Ax = c$ или NIL
$p := FirstPrime; i := 1;$
repeat
  Initialization:
  $\{\tilde{A}_0^{-1}, \tilde{B}, \tilde{b}, S, T, P, r, r_c\} :=$
    $DiagonalForm[mod_p(A,c)];$
  if $r < r_c$ **then return** $NIL;$
  Comment : $\tilde{B} = (\tilde{b}_1, \tilde{b}_2, \ldots, \tilde{b}_{m-r})$
  Comment : $\tilde{b}_j = \begin{pmatrix} \tilde{b}'_j \\ \tilde{\beta}_j \end{pmatrix}$ $Q = \mathrm{diag}(\mathrm{P}, \mathrm{I}_{\mathrm{m-r}})$
  Comment : $\mathrm{I}_{\mathrm{m-r+1}} = (\mathrm{e'}_0, \mathrm{e'}_1, \ldots, \mathrm{e'}_{\mathrm{m-r}}).$
  $\begin{pmatrix} A_0 & A_1 \\ A_2 & A_3 \end{pmatrix} := \mathrm{SAT};$ $\begin{pmatrix} c_0 \\ c_1 \end{pmatrix} := \mathrm{Sc};$
  $\hat{\mathrm{c}} := \mathrm{Pc_0};$ $(\mathrm{A'}_0, \mathrm{a}_0) := \mathrm{PA_0P};$
  $(\mathrm{a}_1, \ldots, \mathrm{a}_{\mathrm{m-r}}) := \mathrm{PA}_1;$ $\mathrm{J} := \emptyset;$
  for $\mathrm{j} := 1$ **to** $\mathrm{m - r}$ **do**
    if $\beta_{\mathrm{j}} = 0$ **then** $\mathrm{J} := \mathrm{J} \cup \{\mathrm{j}\};$
  $\mathrm{j} := 0;$ $\mathrm{Test} = \mathrm{TRUE};$
  repeat
    if $\mathrm{j} \in \mathrm{J}$ **then** $\mathrm{A_j} := (\mathrm{A'}_0, \mathrm{a_j} + \mathrm{a}_0)$
     else $\mathrm{A_j} := (\mathrm{A'}_0, \mathrm{a_j})$
    Lifting:
    $\tilde{\mathrm{A}}_{\mathrm{j}}^{-1} := \begin{pmatrix} I_{r-1} & -\tilde{b}'_j\tilde{\beta}_j^{-1} \\ 0 & \tilde{\beta}_j^{-1} \end{pmatrix} \tilde{\mathrm{A}}_0^{-1}$
    $\mathrm{l} := 1; \mathrm{x} := 0; \mathrm{c}^* := \hat{\mathrm{c}};$
    repeat
      $\tilde{\mathrm{x}} := \mathrm{mod_p}\tilde{\mathrm{A}}_{\mathrm{j}}^{-1}\mathrm{c}^*;$
      $\mathrm{c}^* := (\mathrm{c}^* - \mathrm{A_j}\tilde{\mathrm{x}})/\mathrm{p};$
      $\mathrm{x} := \mathrm{x} + \mathrm{l}\tilde{\mathrm{x}};$ $\mathrm{l} := \mathrm{lp};$
    until $\|\mathrm{l}\| \geq \mathrm{L_j}[\mathrm{A}, \mathrm{c}]$
    $\begin{pmatrix} x'_j \\ \xi_j \end{pmatrix} :=$
      $\mathrm{RationalReconstruction}[\mathrm{x}, \mathrm{N_j}, \mathrm{D_j}];$
    Test:
    if $\mathrm{j} = 0$ **then** $\mathrm{x}_0 := \mathrm{T}\begin{pmatrix} P\begin{pmatrix} x'_0 \\ \xi_0 \end{pmatrix} \\ 0 \end{pmatrix}$
    else if $\mathrm{j} \in \mathrm{J}$
      then $\mathrm{x_j} := \mathrm{T}\begin{pmatrix} P\begin{pmatrix} x'_j \\ \xi_j \end{pmatrix} \\ \xi_j e_j \end{pmatrix}$
       else $\mathrm{x_j} := \mathrm{T}\begin{pmatrix} P\begin{pmatrix} x'_j \\ 0 \end{pmatrix} \\ \xi_j e_j \end{pmatrix};$
    if $\mathrm{Ax_j} \neq \mathrm{c}$ **then** $\mathrm{Test} := \mathrm{FALSE};$
    $\mathrm{j} := \mathrm{j} + 1;$
  until $\mathrm{j} > \mathrm{m - r}$ **or** $\mathrm{Test} = \mathrm{FALSE}$
  $\mathrm{i} := \mathrm{i} + 1; \mathrm{p} := \mathrm{NextPrime}[\mathrm{p}];$
until $\mathrm{Test} = \mathrm{TRUE}$ **or** $\mathrm{i} = \mathrm{NumberOfPrimes}$
if $\mathrm{Test} = \mathrm{FALSE}$ **then return** NIL
else return $\{\mathrm{x}_0, \mathrm{x}_1, \ldots, \mathrm{x}_{\mathrm{m-r}}\}.$

Рис. 3. Алгоритм "Вычисление рационального базиса неоднородной системы линейных уравнений с помощью p-адического подъема".

$\gamma = \prod_{k=1}^{n-1} \|a_k\|$, $\lambda = \min_{k=1,\ldots,n-1} \|a_k\|$, $\mu_j = \min(\lambda, \|a_j\|)$,

$D_j = n^{n/2}\gamma\|a_j\|$ — оценка для знаменателя решения, $N_j = n^{n/2}\|a_j\|\gamma\|c\|\mu_j^{-1}$ — оценка для числителя, $L_j[A,c] = 2N_jD_j$ – оценка для границы p-адического подъема.

Для $F[x]$ получим: $D_j = \|a_j\| + \sum_{k=1}^{n-1}\|a_k\|$ — оценка для знаменателя, $N_j = D_j + \|c\| + \min_{k=1,\ldots,n-1,j}\|a_k\|$ — оценка для числителя, $L_j[A,c] = N_j + D_j$ — оценка для границы p-адического подъема.

$NextPrime[p]$ — генератор простых элементов кольца $\mathbf{R}$, который возвращает следующий простой элемент, после p. $FirstPrime$ и $MinPrime$ обозначают первый простой элемент и наименьший по норме простой элемент.

$NumberOfPrimes$ — количество простых элементов $\lfloor(\max\limits_{j=n,n+1,\ldots,m} D_j)\|MinPrime\|^{-1}\rfloor$.

Пусть $\tilde{\mathbf{R}} = \mathbf{R}/p\mathbf{R}$ фактор-кольцо $\mathbf{R}$ по модулю $p\mathbf{R}$.

$DiagonalForm[mod_p(A,c)]$ алгоритм диагонализации в кольце $\tilde{\mathbf{R}}$. Результатом вычислений будет $\{\tilde{A}_0^{-1}, \tilde{B}, \tilde{b}, S, T, P, r, r_c\}$. Здесь все обозначения имеют прежний смысл. Знак тильды обозначает матрицы над фактор-кольцом $\tilde{\mathbf{R}}$. r и r_c — это ранги матриц mod_pA и $mod_p(A,c)$. Так как $\tilde{\mathbf{R}}$ является полем, то для вычислений можно воспользоваться методом Гаусса или любым другим методом решения систем линейных уравнений над полем.

$DiagonalFormH[mod_p\ A]$ алгоритм диагонализации в $\tilde{\mathbf{R}}$, вычисляет $\{\tilde{A}_0^{-1}, S, T, r\}$.

$RationalReconstraction[\tilde{x}, N_i, D_i]$ – восстановление вектора решения в поле частных кольца $\mathbf{R}$ по решению $\tilde{x}$ в поле $\mathbf{R}/p^k\mathbf{R}$ когда числитель и знаменатель ограничены числами N_i и D_i. Алгоритм описан в работе [11].

РЕШЕНИЕ СИСТЕМЫ В КОММУТАТИВНОЙ ОБЛАСТИ

Теперь рассмотрим задачу решения системы линейных уравнений в коммутативной области. Отметим, что для однородных систем эта задача не представляет трудности, так как всякое решение в поле частных, после умножения на подходящий множитель, дает решение в области. Поэтому дальше будем рассматривать только неоднородные системы.

Пусть α — непустое конечное подмножество в области $\mathbf{R}$. Пересечение главных идеалов, порожденных элементами множества α, является главным идеалом $\cup_{p\in\alpha}(p)$. Порождающий элемент этого идеала обозначим (α), т. е. $((\alpha)) = \cup_{\mathrm{p}\in\alpha}(\mathrm{p})$.

Пусть $\mathbf{K}$ – поле частных области $\mathbf{R}$, x – вектор из пространства $\mathbf{K}^m$, $\tilde{\chi} \subset \mathbf{R}$ – множество знаменателей компонент вектора x.

Определение. *Знаменателем вектора x* будем называть $\chi = \text{НОК}(\tilde{\chi})$.

Вектор x будем записывать в виде произведения $x = \mathbf{x}\chi^{-1}$, $\mathbf{x} \in \mathbf{R}^m$, $\chi \in \mathbf{R}$. Знаменатель вектора x будем обозначать $\text{DEN}(x)$.

Пусть $\mathcal{M} = \{x | x \in \mathbf{K}^m, Ax = b\}$ – множество всех решений системы (1) в $\mathbf{K}^m$. Обозначим $\mathcal{M}_D = \mathcal{M} \cup \mathbf{R}^m$ – множество решений системы, лежащих в модуле $\mathbf{R}^m$. Будем называть $\mathcal{M}_D$ *множеством диофантовых решений.*

Для $x = (x_1, x_2, \ldots, x_s)$ и $y = (y_1, y_2, \ldots, y_s)$ будем обозначать $\langle x, y\rangle = \sum_{i=1}^{s} x_i y_i$.

Теорема 3 *Пусть $\{x_i = \mathbf{x}_i\chi_i^{-1} | i = 1, 2, \ldots, h\}$ – базисное множество решений неоднородной системы линейных уравнений (1) и пусть $\hat{\mathbf{x}} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_h)$, $\chi = (\chi_1, \chi_2, \ldots, \chi_h)$. Тогда*

$$\mathcal{M} = \{\frac{\langle \hat{\mathbf{x}}, q\rangle}{\langle \chi, q\rangle} | q \in \mathbf{R}^h, \langle \chi, q\rangle \neq 0\}$$

Доказательство: Пусть $u = (u_1, u_2, \ldots, u_h) \in \mathbf{K}^h$ такой, что $\sum_{i=1}^{h} u_i = 1$. Обозначим $s_i = u_i\chi_i^{-1}$, g – знаменателей всех чисел $\{s_i | i = 1, 2, \ldots, h\}$, $q_i = gs_i, q_i \in \mathbf{R}$, $i = 1, 2, \ldots, h$, $q = (q_1, q_2, \ldots, q_h)$.

Пусть $\hat{x} = (x_1, x_2, \ldots, x_h)$ и $z = \langle \hat{x}, u\rangle$, следовательно, $z \in \mathcal{M}$. Тогда получим

$$z = \langle \hat{x}, u\rangle = \sum_{i=1}^{h} \mathbf{x}_i\chi_i^{-1}u_i = \sum_{i=1}^{h} \mathbf{x}_i s_i = \langle \hat{\mathbf{x}}, q\rangle g^{-1}, \tag{7}$$

$$1 = \sum_{i=1}^{h} u_i = \sum_{i=1}^{h} \chi_i\chi_i^{-1}u_i = \sum_{i=1}^{h} \chi_i s_i = \langle \chi, q\rangle g^{-1}. \tag{8}$$

Поэтому

$$\frac{\langle \hat{\mathbf{x}}, q\rangle}{\langle \chi, q\rangle} = z \in \mathcal{M}.$$

Наоборот, если $q \in \mathbf{R}^h$, $\langle \chi, q\rangle = g \neq 0$ и $z = \frac{\langle \hat{\mathbf{x}}, q\rangle}{\langle \chi, q\rangle}$, то, обозначая $u_i = \chi_i q_i g^{-1}$, получим $z = \langle x, u\rangle$ и $\sum_{i=1}^{h} u_i = 1$. Поэтому $z \in \mathcal{M}$.

Следствие 3 *Множество всех знаменателей $\{DEN(x) | Ax = c, x \in \mathbf{K}^m\}$ решений системы $Ax = c$ вместе с нулевым элементом образует идеал I_A в кольце $\mathbf{R}$.*

Следствие 4 *Для того чтобы система (1) имела диофантовы решения, необходимо и достаточно, чтобы $I_A = \mathbf{R}$.*

Следствие 5 *Пусть идеал, порожденный знаменателями базисного множества решений $x_i = \mathbf{x}_i\chi_i^{-1}$, $i = 1, 2, \ldots, h$ системы (1), является единичным, $\chi = (\chi_1, \chi_2, \ldots, \chi_h)$. Тогда найдется такой вектор $q = (q_1, q_2, \ldots, q_h) \in \mathbf{R}^h$, что $\langle \chi, q\rangle = 1$ и $z = \langle \mathbf{x}, q\rangle$ – диофантово решение системы (1). Если при этом $q_s \neq 0$, то $x_1, \ldots, x_{s-1}, z, x_{s+1}, \ldots, x_h$ – базисное множество решений этой системы.*

Следствие 6 *Система (1) имеет диофантовы решения, если идеал, порожденный знаменателями базисного множества решений, является единичным.*

Следствие 7 *Пусть идеал, порожденный знаменателями базисного множества решений $x_i = \mathbf{x}_i\chi_i^{-1}$, $i = 1, 2, \ldots, h$ системы (1), является единичным, $\chi = (\chi_1, \chi_2, \ldots, \chi_h)$. Тогда найдется такой вектор $q = (q_1, q_2, \ldots, q_h) \in \mathbf{R}^h$, что $\langle \chi, q\rangle = 1$ и $z = \langle \mathbf{x}, q\rangle$ – диофантово решение системы (1). Если при этом $q_s \neq 0$, то $x_1, \ldots, x_{s-1}, z, x_{s+1}, \ldots, x_h$ – базисное множество решений этой системы.*

Следствие 8 *Пусть $x_i = \mathbf{x}_i\chi_i^{-1}$, $i = 1, 2, \ldots, h$ базисное множество решений системы (1) и $\chi_1 = 1$. Тогда $\mathbf{x}_1$, $\mathbf{x}_i - \mathbf{x}_1(\chi_i - 1)$, $i = 2, 3, \ldots, h$ – линейно независимые диофантовы решения этой системы.*

Доказательство: Так как все $z_i = \mathbf{x}_i - \mathbf{x}_1(\chi_i - 1)$, $i = 2, 3, \ldots, h$ принадлежат $\mathbf{R}^m$ и являются линейно независимыми вместе с x_1, остается показать, что z_i – решения системы (1). Так как $\chi_1 = 1$, то

$$z_i = \mathbf{x}_i - \mathbf{x}_1(\chi_i - 1) = \frac{\mathbf{x}_1(-\chi_i + 1) + \mathbf{x}_i \cdot 1}{\chi_1(-\chi_i + 1) + \chi_i \cdot 1}.$$

Следовательно, по теореме, z_i, $i = 2, 3, \ldots, h$, – решения системы $Ax = b$.

Определение. Будем называть *диофантовым базисом решений системы $Ax =$* – базисное множество решений этой системы, которое целиком лежит в области $\mathbf{R}$.

Другими словами – это $m + 1 - r$ линейно независимых решений неоднородной ($m - r$ – для однородной) системы, принадлежащих $\mathbf{R}^m$.

Следствия последней теоремы позволяют записать алгоритм вычисления диофантова базиса решений системы. Алгоритм приведен на рис. 4.

В этом алгоритме используются следующие функции.

Функция $RationalBasis[A, c]$ вычисляет базисное множество $\{\mathbf{x}_1\chi_1^{-1}, \mathbf{x}_2\chi_2^{-1}, \ldots, \mathbf{x}_h\chi_h^{-1}\}$

Algorithm DiophantineBasis$[\mathbf{A}, \mathbf{c}]$
Input: $A \in \mathbf{R}^{n\times m}, c \in \mathbf{R}^n$
Output: Диофантовы решения или NIL
$\{h, \mathbf{x}_1\chi_1^{-1}, \mathbf{x}_2\chi_2^{-1}, \ldots, \mathbf{x}_h\chi_h^{-1}\} :=$
  $RationalBasis[A, c]$;
if $h = 0$ **then return** NIL;
$(gcd, u_1, u_2, \ldots, u_h) :=$
  $ExdendedGCD[(\chi_1, \chi_2, \ldots, \chi_h)]$;
if $gcd \neq 1$ **then return** NIL
else
  $k := 1$; **while** $u_k = 0$ **do** $k := k + 1$;
  $z_k := \sum_{i=1}^{h} x_i u_i$;
  for $i = 1, 2, \ldots, k-1, k+1 \ldots, h$ **do**
    $z_i := x_i - x_k(\chi_i - 1)$;
return $\{z_1, z_2, \ldots, z_h\}$.

Рис. 4. Алгоритм "Диофантов базис".

решений системы $Ax = c$ и количество h векторов в нем.

Функция $ExdendedGCD[(\chi_1, \chi_2, \ldots, \chi_h)]$ по заданным h числам $\chi_1, \chi_2, \ldots, \chi_h$ вычисляет порождающий элемент gcd идеала $(\chi_1, \chi_2, \ldots, \chi_h)$, если этот идеал является главным, а также множители $u_1, u_2, \ldots, u_h \in \mathbf{R}$, такие, что $gcd = \sum_{i=1}^{h} u_i\chi_i$. Если этот идеал не является главным, то gcd принимает значение NIL.

Если результатом является NIL, то необходимо выполнять следующую итерацию. Для следующей итерации нужно выбрать случайным образом новые матрицы перестановок S и T и вычислить новый рациональный базис. Алгоритм $ExdendedGCD$ нужно применять к знаменателям всех рациональных решений, полученным ранее. Как доказано в [10], ожидаемое число случайных рациональных решений, которое необходимо для получения одного диофантова решения, будет $O(\log n + \log\log\|(A, c)\|)$. Если $\mathrm{rank}A = n$, то за одну итерацию, будет получено $m - n + 1$ рациональных решений. Поэтому ожидаемое количество итераций будет $O((m - n + 1)^{-1}(\log n + \log\log\|(A, c)\|))$.

ОЦЕНКА СЛОЖНОСТИ АЛГОРИТМОВ

Сложность вычисления диофантового базиса определяется количеством операций в алгоритмах 1 и 4. Сложность первого из них составляет $O(mn^{\beta-1})$, а второго $O(m(m-r)+C_G)$ операций в кольце $\mathbf{R}$. C_G — количество операций, необходимых для разложения единицы по $m - r$ образующим единичного идеала в кольце $\mathbf{R}$. Для такого разложения единицы можно воспользоваться, например, алгоритмом вычисления базиса Гребнера. Оценка сложности такого алгоритма в общем случае выходит за рамки настоящей статьи. Отметим только, что для евклидовых колец C_G — это количество операций в алгоритме Евклида, который вычисляет наибольший общий делитель для $m - r$ чисел.

Более определенные оценки можно получить для алгоритмов, использующих линейный p-адический подъем.

Известны оценки сложности алгоритма Диксона [10]. Для целых чисел $\mathbf{Z}$ сложность алгоритма Диксона для решения системы (1) при $n = m$ и $\det A \neq 0$ ограничена числом $\mathcal{D}_Z = O(n^3(\log n + \log\|A\| + \log p)^2 + n(\log\|\|)^2)$ бит-операций. Для кольца многочленов $F[x]$ над полем F сложность алгоритма Диксона ограничена числом $\mathcal{D}_{F[x]} = O(n^3(\deg A + \deg p)^2 + n(\deg b)^2)$ операций в поле F.

Для вычисления базиса, состоящего из $m-r$ решений потребуется не более, чем $\mathcal{C}_Z = (m - r)\mathcal{D}_Z$ и $\mathcal{C}_{F[x]} = (m - r)\mathcal{D}_{F[x]}$ операций, соответственно для каждого из этих случаев.

Перейдем к оценке сложности вычисления диофантова базиса при использовании p-адического подъема.

Количесто операций при вычислении диофантова базиса определяется главным образом количеством операций необходимых для вычисления рационального базиса и количеством итераций. Ожидаемое количество итераций равно $O((m - n + 1)^{-1}(\log n + \log\log\|(A, c)\|))$, а количество операций будет соответственно $\mathcal{C}_Z$ и $\mathcal{C}_{F[x]}$ для $\mathbf{Z}$ и $F[x]$ соответственно. Следовательно, при $m - n << n$ количество операций для вычисления диофантова базиса от размеров системы зависит существенно по кубическому закону.

По сравнению с алгоритмом [10], среднее число вычислений обратной матрицы теперь в $m - r$ раз меньше.

ЗАКЛЮЧЕНИЕ

В заключение сравним по количеству операций приведенные алгоритмы со стандартными приближенными методами решения систем линейных уравнений при использовании чисел с плавающей точкой.

С помощью масштабирования числа из формата с плавающей точкой переводятся в целые числа. А затем решение ищется в поле частных. Такой перенос в область точных решений позволяет избавиться от проблемы неустойчивости решения, накопления погрешности и т.п. Вопрос состоит в том, как при этом увеличивается вычислительная сложность.

Если использовать арифметику с s-битными словами, то метод Гаусса для решения системы с матрицей размера $n \times m$ ранга n потребует $O(n^2ms^2)$ бит-операций.

Приведенный алгоритм построения рационального базиса решений требует для $\mathbf{R} = \mathbf{Z}$ примерно $O((m-n+1)n^2mw^2)$ бит-операций. Где $w = \log n + \log \|A\| + \log p$. Их отношение равно $O((m-n+1)(w/s)^2$. В реальных задачах $\log n + \log \|A\|$ меньше s, а p обычно выбирается так, что $\log p$ меньше s. Поэтому w/s – величина меньше 1. А искомое отношение – это число $(m-n+1)$.

Отсюда видно, что при $m-n << n$ алгоритм вычисления рационального решения имеет такую же сложность, что и метод Гаусса для приближенной арифметики при использовании чисел с плавающей точкой. Однако с ростом разности $m-n$ отношение растет линейно, как $m-n$.

ЛИТЕРАТУРА

1. *Strassen. V.* Gaussian Elimination is not optimal // Numerische Mathematik. 1969. V. 13. P. 354–356.

2. *Coppersmith D. and Winograd S.* Matrix multiplication via arithmetic progressions // Journal of Symbolic Computation. 1990. V. 9. P. 251–280.

3. *Dodgson C.L.* Condensation of determinants, being a new and brief method for computing their arithmetic values // Proc. Royal Soc. Lond. 1866. V. A.15. P. 150–155.

4. *Bareiss E.H.* Sylvester's Identity and Multistep Integer-Preserving Gaussian Elimination // Math. Comp. 1968. V. 22 (103). P. 565–578.

5. *Malaschonok G.I.* Solution of a system of linear equations in an intergal domain // USSR J. of computational Mathematics and Mathematical Physics. 1983. V. 23. P. 1497-1500.

6. *Malaschonok G.I.* Algorithms for the solution of systems of linear equations in commutative rings. // Effective Methods in Algebraic Geometry / Ed. by T. Mora and C. Traverso, Progress in Mathematics 94, Birkhauser, 1991. P. 289–298.

7. *Malaschonok G.I.* Recursive Method for the Solution of systems of Linear Equations // Computational Mathematics (Sydow A. Ed, Proceedings of the 15th IMACS World Congress, Vol. I, Berlin, August 1997). Wissenschaft & Technik Verlag. Berlin, 1997. P. 475–480.

8. *Bareiss E.H.* Computational solutions of matrix problems over an integral domain // J. Inst. Maths Applics. 1972. V. 10. P. 68–104.

9. *Dixon J.* Exact solution of linear equations using p-adic expansions // Numer. Math. 1982. V. 40. P. 137–141.

10. *Mulders T. and Storjohann A.* Diophantine Linear System Solving // Proceedings of ISSAC'99: ACM International Symposium on Symbolic and Algebraic Computation. July 1999, Vancouver, Canada.

11. *Collins G.E. and Encarnacion M.J.* Efficient rational number reconstruction. Journal of Symbolic Computation. 1995. V. 20. P. 287–297.

Поступила в редакцию 1 февраля 2000 г.