Math-Net.Ru

Общероссийский математический портал

О. Н. Переславцева, О вычислении характеристического полинома матрицы, *Дискрет. матем.*, 2011, том 23, выпуск 1, 28–45

DOI: <https://doi.org/10.4213/dm1128>

Использование Общероссийского математического портала Math-Net.Ru подразумевает, что вы прочитали и согласны с пользовательским соглашением
<http://www.mathnet.ru/rus/agreement>

Параметры загрузки:

IP: 3.89.197.203

26 апреля 2023 г., 17:53:41

О вычислении характеристического полинома матрицы

© 2011 г. О. Н. Переславцева

В статье рассмотрены эффективные алгоритмы вычисления характеристических полиномов матриц над коммутативными кольцами. Приведены оценки сложности алгоритмов в числе кольцевых операций, а для кольца целых чисел получены оценки сложности в числе мультипликативных операций над машинными словами. Предложен новый алгоритм вычисления характеристического полинома, имеющий асимптотически лучшую оценку сложности в кольцевых операциях. Даются рекомендации по применению алгоритмов вычисления характеристических полиномов в зависимости от размера матрицы, в частности, предлагаемый алгоритм рекомендуется применять для целочисленных матриц порядка 60 и более.

1. Введение

Пусть R — коммутативное кольцо, $A = (a_j^i)$ — квадратная матрица порядка n с элементами $a_j^i \in R$, $i = 1, \dots, n$, $j = 1, \dots, n$, E_n — единичная матрица порядка n ,

$$F(x) = \det(A - xE_n)$$

— характеристический полином матрицы A .

Мы рассматриваем семь методов вычисления характеристического полинома и предлагаем новый алгоритм. Исследуем сложности алгоритмов и обсуждаем результаты вычислительных экспериментов.

Сложность алгоритмов будем оценивать двумя разными способами. В общем случае будем оценивать число операций над элементами кольца, а в частном случае, когда R — кольцо целых чисел $\mathbf{Z}$ и для записи чисел требуется несколько машинных слов, будем оценивать число операций над машинными словами. Первый способ дает возможность сравнивать алгоритмы в $\mathbf{Z}$ при использовании модулярных вычислений, основанных на китайской теореме об остатках (КТО), а вторая оценка показывает эффективность прямых вычислений в кольце целых чисел.

Для оценки числа операций с машинными словами в алгоритмах принята следующая вычислительная модель. При умножении двух чисел, содержащих l и m слов, получаем число, содержащее $l + m$ слов; при сложении n чисел, каждое из которых содержит m слов, получаем число, содержащее $(m + \lceil \log_w n \rceil)$ слов, где $w = 2^b$, b — число двоичных разрядов в машинном слове. Введем обозначение $\varphi_n = \lceil \log_w n \rceil$, то есть φ_n — число машинных слов, необходимых для записи числа n . Будем считать, что если элементы матрицы A содержат k слов, то элементы матрицы A^2 содержат $2k + \varphi_n$ слов, и элементы матрицы A^t содержат $tk + (t - 1)\varphi_n$ слов.

Для пяти из рассматриваемых алгоритмов мы найдем выражения для числа мультипликативных операций в словах. Для двух алгоритмов мы покажем, что число операций в словах выражается экспоненциальной функцией. Будем считать, что операция деления имеет такую же сложность, что и операция умножения частного от деления на делитель.

В главах 2–4 рассматриваются алгоритм Леверье [1] и его видоизменения — алгоритм Леверье–Фаддеева [2] и алгоритм Леверье–Винограда (см. с. 656 в [3]). Алгоритмы Леверье и Леверье–Фаддеева имеют сложность $2n^4$ кольцевых операций, а алгоритм Леверье–Винограда — $2n^{3.5}$, если использовать стандартное умножение матриц. Если использовать алгоритм Штрассена [4] для умножения матриц, то первые два алгоритма будут иметь сложность порядка $n^{3.81}$, а третий — порядка $n^{3.31}$.

Глава 5 посвящена алгоритму А. Л. Чистова [5], для выполнения которого потребуется порядка $\frac{2}{3}n^4$ кольцевых операций.

В главе 6 приводится алгоритм, предложенный Т. Р. Сейфуллиным [6] для произвольного коммутативного кольца, сложность которого имеет порядок $\frac{1}{2}n^4$ кольцевых операций.

В главе 7 рассматривается алгоритм Берковича [7], в котором число кольцевых операций имеет порядок $\frac{1}{2}n^4$.

В главе 8 дается алгоритм, предложенный Г. И. Малашонком [8], сложность которого имеет порядок $\frac{8}{3}n^3$ кольцевых операций.

В главе 9 предлагается новый метод, который имеет сложность порядка $\frac{7}{3}n^3$ кольцевых операций.

Глава 10 посвящена обсуждению результатов численных экспериментов и сравнению полученных программ с известными математическими пакетами.

2. Метод Леверье

Один из первых методов вычисления характеристического полинома предложил в 1848 г. французский астроном Леверье [1]. Этот метод стал основой для многих других эффективных алгоритмов вычисления характеристического полинома (см. [2, 6, 9]).

Метод Леверье включает три этапа.

- (1) Вычисление для матрицы A всех ее степеней $A^2, A^3, \dots, A^{n-1}$ и вычисление диагональных элементов матрицы A^n .
- (2) Вычисление следов этих матриц $s_i = \text{Tr}(A^i)$, $i = 1, \dots, n$.
- (3) Вычисление коэффициентов f_i , $1 \leq i \leq n$, характеристического полинома $\sum_{i=0}^n f_i x^{n-i}$ матрицы с использованием для этого формулы Ньютона

$$\begin{aligned} s_1 - f_1 &= 0, \\ s_2 - s_1 f_1 - 2f_2 &= 0, \\ &\dots \\ s_n - s_{n-1} f_1 - \dots - s_1 f_{n-1} - n f_n &= 0. \end{aligned} \tag{1}$$

Будем использовать стандартное умножение матриц. Тогда на этапе 1 требуется выполнить $(n-2)n^3 + n^2$ операций умножения и $(n-2)(n-1)n^2 + n(n-1)$ операций сложения; на этапе 2 — $n(n-1)$ операций сложения; на этапе 3 — $\sum_{t=2}^n (t-1) = \frac{1}{2}n^2 - n$ операций умножения и столько же операций сложения, $n-1$ операций деления (сокращения). Следовательно, алгоритм требует выполнения $n^4 - 2n^3 + \frac{3}{2}n^2 - \frac{1}{2}n$ операций умножения, $n^4 - 3n^3 + \frac{9}{2}n^2 - \frac{5}{2}n$ операций сложения и $n-1$ операций деления (сокращения) в кольце.

Если использовать алгоритм Штрассена [4] для умножения матриц, который требует выполнения $n^{2,81}$ операций умножения элементов матриц, то алгоритм Леверье требует выполнения $\Theta(n^{3,81})$ операций умножения в кольце.

Оценим сложность данного алгоритма в кольце целых чисел в количестве операций умножения машинных слов. Если элементы матрицы A содержат k слов, то элементы матрицы A^t содержат $\alpha_t = tk + (t-1)\varphi_n$ слов, числа s_t и $f_t - \beta_t = t(k + \varphi_n)$ слов. Для умножения матриц будем использовать алгоритм Штрассена. Всего на первом этапе выполняется

$$\sum_{t=2}^{n-1} \alpha_1 \alpha_t n^{2,81} + \alpha_1 \alpha_{n-1} n^2 \sim \frac{k}{2} (k + \varphi_n) n^{4,81}$$

умножений слов. На этапе 2 происходит только сложение элементов. При вычислении f_t на 3-м этапе требуется выполнить

$$\sum_{p=1}^{t-1} \beta_p \beta_{t-p} = (k + \varphi_n)^2 \left(\frac{1}{6} t^3 - \frac{2}{3} t^2 + \frac{1}{2} t + \frac{1}{2} \right) + (k + \varphi_n) \varphi_t t$$

умножений слов. Суммируя это выражение по t от 2 до n , получим, что общее число умножений слов на 3-м этапе имеет порядок $n^4(k + \varphi_n)^2/24$. Таким образом, число умножений слов в методе Леверье имеет порядок

$$\frac{k}{2} (k + \varphi_n) n^{4,81}. \quad (2)$$

3. Метод Леверье–Винограда

Для вычисления коэффициентов характеристического полинома с помощью алгоритма Леверье нужно знать лишь диагональные элементы матриц $A, A^2, \dots, A^n$. Для этого, как заметил С. Виноград (см. с. 656 в [3]), достаточно вычислять

(1*) матрицы $A^2, \dots, A^m, m \leq n$;

(2*) матрицы $A^{2m}, A^{3m}, \dots, A^{\lfloor n/m \rfloor m}$;

(3*) диагональные элементы остальных матриц.

Заменив в методе Леверье этап 1 шагами 1*–3*, получаем алгоритм Леверье–Винограда.

В этом случае алгоритм требует выполнения

$$\begin{aligned} (m-1)n^3 + (n/m-1)n^3 + (n/m-1)(m-1)n^2 + n(n-1)/2 \\ = (m + (n-1)/m-1)n^3 - (m-3/2)n^2 - n/2 \end{aligned}$$

операций умножения,

$$\begin{aligned} (m-1)(n-1)n^2 + (n/m-1)(n-1)n^2 + (n/m-1)(m-1)(n-1) + n(n-1)/2 \\ = (m + (n-1)/m-2)n^3 - (m+1/m-3/2)n^2 - (m-1/2)n + (m-1) \end{aligned}$$

операций сложения и $n-1$ операций деления (сокращения) в кольце.

Наименьшее значение выражение $m + (n - 1)/m - 1$ принимает при $m = \sqrt{n - 1}$. Следовательно, алгоритм требует выполнения $2n^{3,5} + O(n^3)$ операций умножения и столько же операций сложения в кольце.

Если для умножения матриц использовать алгоритм Штрассена, то шаги 1*-2* можно выполнить за $\Theta(mn^{2,81})$ операций умножения. Тогда алгоритм Леверье–Винограда вычисления характеристического полинома требует выполнения $\Theta(n^{3,31})$ операций умножения.

Для умножения матриц будем использовать алгоритм Штрассена. Элементы матрицы A^t содержат $\alpha_t = tk + (t - 1)\varphi_n$ слов, числа s_t и $f_t - \beta_t = t(k + \varphi_n)$ слов.

На шаге 1* требуется выполнить

$$\sum_{i=1}^{m-1} (\alpha_1 \alpha_i) n^{2,81} \sim \frac{k}{2} (k + \varphi_n) n^{3,81}$$

умножений слов.

Найдем число умножений слов на шаге 2*. Матрицу $A^{(i+1)m}$ вычисляем следующим образом: $A^{(i+1)m} = A^m A^{im}$. Следовательно, на шаге 2* потребуется

$$\sum_{i=1}^{n/m-1} (\alpha_m \alpha_{im}) n^{2,81} \sim \frac{k}{2} (k + \varphi_n) n^{4,81}$$

умножений слов.

Число умножений слов на шаге 3* на порядок меньше, оно есть $\Theta(n^3)$. Число умножений слов на этапе 3 есть $\Theta(n^4)$.

Таким образом, число умножений слов в методе Леверье–Винограда имеет порядок

$$\frac{k}{2} (k + \varphi_n) n^{4,81}. \quad (3)$$

Зная коэффициенты $f_1, \dots, f_{n-1}$ и матрицы $A^2, A^3, \dots, A^m, A^{2m}, A^{3m}, \dots, A^{\lfloor n/m \rfloor m}$, где $m = \lfloor \sqrt{n - 1} \rfloor$, можно найти присоединенную матрицу

$$A^\perp = f_{n-1} E + \sum_{j=0}^{\lfloor n/m \rfloor} A^{jm} \sum_{i=1}^m f_{n-1-(jm+i)} A^i. \quad (4)$$

При этом требуется выполнить еще $(n/m)n^3 + (n/m)mn^2 = n^{3,5} + n^3$ операций умножения и n операций сложения в кольце. Следовательно, вычисление присоединенной матрицы требует выполнения $3n^{3,5} + O(n^{2,5})$ операций умножения, $2n^{3,5} + O(n^3)$ операций сложения в кольце и $n - 1$ операций деления (сокращения) в кольце. Если для умножения матриц использовать алгоритм Штрассена, то вычисление присоединенной матрицы требует выполнения $\Theta(n^{3,31})$ операций умножения.

Найдем число умножений слов, которое требуется для вычисления присоединенной матрицы по формуле (4).

При вычислении $A^{jm} M_j$, где

$$M_j = \sum_{i=1}^m f_{n-1-(jm+i)} A^i,$$

для каждого $j = 0, 1, \dots, n/m$ необходимо выполнить

$$\begin{aligned} \mu_j &= n^{2,81} \alpha_{jm} (\beta_{n-1-(jm+i)} + \alpha_i + \varphi_m) \\ &= n^{2,81} ((k + \varphi_n)^2 (njm - j^2 m^2) - (k + \frac{1}{2} \varphi_n) jm + \frac{1}{2} \varphi_n^2 - (n - 1)(k + \varphi_n) \varphi_n) \end{aligned}$$

умножений слов. Тогда для вычисления произведений $A^{jm} M_j$, $j = 0, 1, \dots, n/m$, число умножений слов равно

$$\sum_{j=0}^{n/m} \mu_j \sim \frac{1}{6}(k + \varphi_n)^2 n^{5,31}. \quad (5)$$

4. Метод Леверье–Фаддеева

Рассмотрим видоизменение Д. К. Фаддеева метода Леверье. Запишем теорему Гамильтона–Кэли для характеристического полинома

$$F(x) = (-1)^n (x^n - f_1 x^{n-1} - f_2 x^{n-2} - \dots - f_n)$$

матрицы A в схеме Горнера:

$$A(A(\dots(A(A - f_1 E_n) - f_2 E_n) \dots - f_{n-1} E_n) - f_n E_n = 0.$$

Будем обозначать через $B_{(t)}$ выражение в t -х внутренних скобках:

$$B_{(t)} = A(\dots(A(A - f_1 E_n) - f_2 E_n) \dots - f_{t-1} E_n) - f_t E_n.$$

Очевидно, что

$$B_{(1)} = (A - f_1 E_n), \quad B_{(t+1)} = AB_{(t)} - f_{t+1} E_n.$$

Введем обозначение $A_{(t)} = AB_{(t-1)}$.

Заметим, что

$$\begin{aligned} A_{(t)} &= A(\dots(A(A - f_1 E_n) - f_2 E_n) \dots - f_{t-1} E_n) \\ &= A^t - f_1 A^{t-1} - \dots - f_{t-1} A. \end{aligned}$$

Введем обозначение $\text{Tr}(A^i) = s_i$. Отсюда следует, что

$$\text{Tr}(A_{(t)}) = s_t - f_1 s_{t-1} - \dots - f_{t-1} s_1.$$

Поэтому

$$f_t = \frac{1}{t} \text{Tr} A_{(t)}$$

согласно формулам Ньютона (1).

Алгоритм Леверье–Фаддеева можно записать следующим образом:

$$\begin{aligned} B_{(0)} &= E_n; \quad t = 1, \dots, n: \\ \{A_{(t)} &= AB_{(t-1)}; \end{aligned} \quad (6)$$

$$\begin{aligned} f_t &= \frac{1}{t} \text{Tr} A_{(t)}; \\ B_{(t)} &= A_{(t)} - f_t E_n \}. \end{aligned} \quad (7)$$

Отметим, что на последнем шаге $t = n$ по формуле (6) вычисляется только n диагональных элементов матрицы $A_{(n)}$, а формула (7) не используется.

Этот алгоритм вычисляет коэффициенты $f_1, \dots, f_n$ характеристического полинома, определитель $\det(A) = f_n$, присоединенную матрицу $B_{(n-1)}$.

Покажем, что $B_{(n-1)}$ — присоединенная матрица. Так как $B_{(n)} = A_{(n)} - f_n E_n$ и $B_{(n)} = 0$ по теореме Гамильтона–Кэли, $A_{(n)} = f_n E_n$. Поэтому

$$AB_{(n-1)} = f_n E_n.$$

Алгоритм требует выполнения

$$(n-2)n^3 + n^2 = n^4 - 2n^3 + n^2$$

операций умножения,

$$(n-2)(n-1)n^2 + n(n-1) + n^2 + n(n-1) = n^4 - 3n^3 + 5n^2 - 2n$$

операций сложения и $n-1$ операций деления (сокращения) в кольце R .

Если использовать алгоритм Штрассена для умножения матриц, то алгоритм Леверье–Фаддеева требует выполнения $\Theta(n^{3,81})$ операций умножения в кольце.

Оценим сложность данного алгоритма в кольце целых чисел в количестве операций умножения машинных слов. Если элементы исходной матрицы A содержат k слов, то элементы матрицы $B_{(t)}$ содержат $\beta_t = t(k + \varphi_n) - \varphi_n$ слов. Если для умножения матриц будем использовать алгоритм Штрассена, тогда общее число умножений слов будет равно

$$\sum_{t=1}^{n-1} k\beta_{t-1}n^{2,81} + k\beta_{n-1}n^2 + \sum_{t=2}^n (\beta_t + \varphi_n)\varphi_t \sim \frac{k}{2}(k + \varphi_n)n^{4,81}. \quad (8)$$

5. Алгоритм Чистова

Пусть задана матрица A , и пусть A_r обозначает подматрицу размера $r \times r$ в верхнем левом углу матрицы A . Пусть $E_r^{1,r}$ — это r -й столбец единичной матрицы E_r порядка r .

Алгоритм Чистова [5] вычисления характеристического полинома матрицы A состоит из четырех этапов:

- (1) Вычисляются все n степеней для каждой матрицы A_r , $r = 1, 2, \dots, n$,

$$a_{rt} = (E_r^{1,r})^T (A_r)^t E_r^{1,r}$$

— нижний правый элемент матрицы $(A_r)^t$, $r = 1, \dots, n$, $t = 1, \dots, n$.

$$(2) \tilde{Q}(x) = \prod_{r=1}^n \left(\sum_{t=0}^n a_{rt} x^t \right) \bmod x^{n+1}.$$

$$(3) Q(x) = (\tilde{Q}(x))^{-1} \bmod x^{n+1}.$$

$$(4) F(x) = (-1)^n x^n Q(1/x).$$

На первом этапе достаточно считать только последнюю строку матрицы $(A_r)^t$, затем ее умножать на матрицу A_r , а при $t = n$ умножать лишь на последний столбец матрицы A_r . Поэтому необходимо выполнить

$$\sum_{r=1}^n \sum_{t=2}^n r^2 = \frac{1}{3}n^4 + \frac{1}{6}n^3 - \frac{1}{3}n^2 - \frac{1}{6}n$$

операций умножения,

$$\sum_{r=1}^n \sum_{t=2}^n r(r-1) = \frac{1}{3}n^4 - \frac{1}{3}n^3 - \frac{1}{3}n^2 + \frac{1}{3}n$$

операций сложения. На остальных этапах количество кольцевых операций составит $\Theta(n^3)$. Поэтому алгоритм требует выполнения $\frac{1}{3}n^4 + O(n^3)$ операций умножения и столько же операций сложения в кольце R .

Оценим сложность алгоритма Чистова, подсчитывая количество умножений слов. Если элементы входной матрицы содержат k слов, то элементы матрицы A^s содержат $\alpha_s = sk + (s-1)\varphi_n$ слов.

На первом этапе вычисляется нижний правый элемент матрицы $(A_r)^t$, $r = 1, \dots, n$, $t = 1, \dots, n$. Для этого нужно знать только последнюю строку матрицы $(A_r)^{t-1}$ и умножить ее на матрицу A_r . При $t = n$ нужно умножить лишь на последний столбец матрицы A_r . Тогда на первом этапе выполняется

$$\sum_{r=1}^n \left(\sum_{t=3}^{n-1} r^2 k \alpha_{t-1} + k \alpha_{n-1} \right) \sim \frac{k^2 + k\varphi_n}{6} n^5$$

умножений слов.

На втором этапе вычисляется произведение n полиномов степени n по модулю x^{n+1} . Коэффициент при x^t полинома, полученного на s -м шаге, содержит $tk + s\varphi_n$ слов. Таким образом, на следующем шаге при умножении по модулю x^n полученного на s -м шаге полинома на $(s+1)$ -й полином требуется

$$\begin{aligned} \sum_{i=1}^{n/2} \sum_{j=1}^{n/2} (ik + s\varphi_n)(jk + \varphi_n) \\ = \frac{k^2}{32}n^4 + \frac{k}{8}(k + \varphi_n/2)n^3 + \frac{k}{8}(k + \varphi_n)n^2 + s \left(\frac{k\varphi_n}{16}n^3 + \left(\frac{k\varphi_n}{8} + \frac{\varphi_n^2}{4} \right) n^2 \right) \end{aligned}$$

умножений слов.

Всего шагов будет $n-1$, поэтому, суммируя последнее выражение по s от 1 до $n-1$, получим, что число умножений слов имеет порядок $(k/32)(k + \varphi_n)n^5$.

На третьем этапе вычисляется полином

$$Q(x) = b_0x^n + \dots + b_n$$

такой, что

$$\tilde{Q}(x)Q(x) = 1 \bmod x^{n+1}.$$

Запишем $\tilde{Q}(x)$ в виде

$$\tilde{Q}(x) = a_0x^n + \dots + a_n.$$

Приравнявая коэффициенты при одинаковых степенях, получим систему из $n+1$ уравнений с $n+1$ неизвестными $b_0, \dots, b_n$:

$$b_n a_n = 1,$$

$$\sum_{i=n-t}^n b_i a_{2n-t-i} = 0, \quad t = 1, \dots, n.$$

Система имеет треугольную матрицу и решается непосредственно.

Число a_{n-s} содержит $sk + n\varphi_n$ слов. Тогда b_i содержит $(n-i)(k + n\varphi_n)$ слов. Следовательно, при вычислении b_{n-t} необходимо выполнить

$$\begin{aligned} M_t &= \sum_{i=n-(t-1)}^{n-1} ((n-i)(k + n\varphi_n)((t+i-n)k + n\varphi_n)) \\ &= (k + n\varphi_n) \left(\frac{k}{6}t^3 + \frac{n\varphi_n}{2}t^2 - \left(\frac{n\varphi_n}{2} + \frac{k}{6} \right)t \right) \end{aligned}$$

умножений слов.

Тогда на третьем этапе общее число умножений слов равно

$$\sum_{t=2}^n M_t \sim \frac{(k + 4\varphi_n)\varphi_n}{24} n^5.$$

Итого, в алгоритме Чистова число выполняемых умножений слов имеет порядок

$$\frac{19k^2 + 23k\varphi_n + 16\varphi_n^2}{96} n^5. \quad (9)$$

6. Алгоритм Берковича

Обозначим $P_n = ((-1)^n, f_1, f_2, \dots, f_n)$ вектор коэффициентов характеристического полинома. Рассмотрим матрицу размера $(r+2) \times (r+1)$

$$\text{Тоер}(Q_{r+1}) = \begin{pmatrix} -1 & 0 & \dots & 0 & 0 \\ a_{r+1}^{r+1} & -1 & \dots & 0 & 0 \\ (R_r S_r) & a_{r+1}^{r+1} & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ (R_r(A_r)^{r-2} S_r) & (R_r(A_r)^{r-3} S_r) & \dots & a_{r+1}^{r+1} & -1 \\ (R_r(A_r)^{r-1} S_r) & (R_r(A_r)^{r-2} S_r) & \dots & (R_r S_r) & a_{r+1}^{r+1} \end{pmatrix},$$

где $R_r = (a_1^{r+1}, a_2^{r+1}, \dots, a_r^{r+1})$, $S_r = (a_{r+1}^1, a_{r+1}^2, \dots, a_{r+1}^r)^T$. Эта матрица получается из теплицевой после вычеркивания последнего столбца $(0, \dots, 0, -1)^T$.

Берковичем было показано, что

$$P_n = \text{Тоер}(Q_n) \text{Тоер}(Q_{n-1}) \dots \text{Тоер}(Q_1).$$

Для вычисления всех матриц $\text{Тоер}(Q_{r+1})$, $r = 0, 1, \dots, n-1$ необходимо выполнить

$$1 + \sum_{r=2}^{n-1} (r + (r-1)(r^2 + r)) = \frac{1}{4}n^4 - \frac{1}{2}n^3 + \frac{1}{4}n^2$$

операций умножения и

$$\sum_{r=2}^{n-1} (r - 1 + (r-1)r(r-1)) = \frac{1}{4}n^4 - \frac{7}{6}n^3 + \frac{9}{4}n^2 - \frac{7}{3}n + 1$$

операций сложения. Для вычисления P_n требуется выполнить

$$\sum_{r=2}^{n-1} r(r+1) = \frac{1}{3}n^3 - \frac{1}{3}n - 2$$

операций умножения и

$$\sum_{r=2}^{n-1} (r^2 - 1) = \frac{1}{3}n^3 + \frac{1}{2}n^2 - \frac{5}{6}n + 2$$

операций сложения. Следовательно, алгоритм Берковича [7] требует выполнения $\frac{1}{4}n^4 + O(n^3)$ операций умножения и столько же операций сложения в кольце R .

Найдем оценку числа мультипликативных операций.

Для нахождения числа $R_r(A_r)^i S_r$ нужно умножить $i+1$ раз матрицу A_r размера $r \times r$ на столбец и затем умножить слева на строку R_r . Поэтому потребуется выполнить

$$T_r = \sum_{i=0}^{r-1} k r^2 ((i+1)k + i\varphi_n) = \frac{1}{2}k(k + \varphi_n)r^4 + \frac{1}{2}k(k - \varphi_n)r^3$$

умножений слов.

Для вычисления всех матриц $\text{Тоер}(Q_{r+1})$, $r = 1, \dots, n-1$, потребуется

$$\sum_{r=1}^{n-1} T_r \sim \frac{1}{10}k(k + \varphi_n)n^5$$

умножений слов.

При вычислении P_n умножение матриц будем выполнять справа. На шаге t , $t = 2, \dots, n-1$, получим вектор $P_t = (\pm 1, q_1, q_2, \dots, q_t)^T$, где число q_s содержит $sk + (s-1)\varphi_n$ слов. Тогда умножение матрицы $\text{Тоер}(Q_{t+1})$ на вектор P_t для $t = 2, \dots, n-1$ требует

$$S_t = \sum_{i=2}^{t+1} \sum_{j=1}^{i-1} [((i-j)k + (i-j-1)\varphi_n)(jk + (j-1)\varphi_n)]$$

умножений слов.

Для вычисления P_n число умножений слов будет равно

$$\sum_{t=2}^{n-1} S_t \sim \frac{1}{120}(k + \varphi_n)^2 n^5.$$

Итого, алгоритм Берковича требует выполнения порядка

$$\frac{1}{120}(13k^2 + 14k\varphi_n + \varphi_n^2)n^5 \quad (10)$$

умножений слов.

7. Алгоритм Сейфуллина

В основе алгоритма Сейфуллина [6] лежит простой факт: правый нижний элемент матрицы $G_t(x)$, которая является присоединенной для матрицы $x E_t - A_t$, — это характеристический полином

$$\chi_{t-1} = \sum_{p=0}^{t-1} f_{t-1}^p x^{t-1-p}$$

матрицы, получаемой вычеркиванием из матрицы A_t последней строки и последнего столбца:

$$(G_t(x))_t^t = \chi_{t-1}(x), \quad (x E_t - A_t) G_t(x) = E_t \chi_t(x).$$

Запишем $G_t(x)$ как матричный полином

$$\sum_{p=0}^{t-1} \tilde{G}_{t,p} x^{t-1-p}.$$

Введем обозначения

$$G_{t,p} = (-1)^p \tilde{G}_{t,p},$$

$$H_t(x) = A_t G_t(x) = \sum_{p=0}^{t-1} \tilde{H}_{t,p} x^{t-1-p}, \quad H_{t,p} = (-1)^p \tilde{H}_{t,p}.$$

Тогда

$$x G_t(x) = E_t \chi_t(x) + H_t(x).$$

Отсюда следует, что

$$\chi_t(x) = x \chi_{t-1} - (H_t(x))_t^t, \quad (11)$$

$$x G_t(x) = E_t x \chi_{t-1}(x) - E_t (H_t(x))_t^t + H_t(x). \quad (12)$$

Для вычисления $\chi_t(x)$ достаточно знать $(H_t(x))_t^t$ и $(G_t(x))_t^t$.

Приравнявая коэффициенты при одинаковых степенях в равенствах (11) и (12), получаем для всех $t = 1, \dots, n$ и всех $p = 0, 1, \dots, t$ равенства

$$\begin{aligned} f_t^p &= f_{t-1}^p - (\tilde{H}_{t,p})_t^t, \\ (\tilde{G}_{t,p})_t^{1,t} &= E_t^{1,t} f_{t-1}^p + E_{1,t-1}^{1,t} (\tilde{H}_{t,p})_t^{1,t-1}, \\ (\tilde{H}_{t,p})_t^{1,t} &= A_t (\tilde{G}_{t,p})_t^{1,t}. \end{aligned} \quad (13)$$

Равенства (13), с учетом введенных выше обозначений, можно записать в виде следующего алгоритма, который назовем алгоритмом Сейфуллина:

$$f_1^1 = A_1;$$

$$t = 2, \dots, n:$$

$$\{ H_t^{1,t} = A_t^{1,t}; \\ p = 1, \dots, t-1:$$

$$\begin{aligned} \{ f_t^p &= f_{t-1}^p + H_t^t; \\ G_t^t &= f_{t-1}^p; G_t^{1,t-1} = -H_t^{1,t-1}; \\ \text{if } p < t-1 &\text{ then } H_t^{1,t} = A_t G_t^{1,t}; \\ &\text{else } H_t^{1,t} = A_{1,t}^t G_t^{1,t}; \} \end{aligned}$$

$$f_t^t = H_t^t; \}$$

Оценим число умножений и сложений. Для $t > 1$ при каждом $p = 1, \dots, t-2$ число умножений равно t^2 , а число сложений равно $1 + t(t-1)$. При $p = t-1$ число умножений равно t , а число сложений равно $1 + (t-1)$. Всего алгоритм требует выполнения

$$\sum_{t=2}^n \left(\sum_{p=1}^{t-2} t^2 + t \right) = \frac{1}{4}n^4 - \frac{1}{6}n^3 - \frac{1}{4}n^2 + \frac{1}{6}n$$

операций умножения и

$$\sum_{t=2}^n \sum_{p=1}^{t-1} (1 + t(t-1)) = \frac{1}{4}n^4 - \frac{1}{2}n^3 + \frac{3}{4}n^2 - \frac{1}{2}n$$

операций сложения в кольце R .

При вычислении характеристического полинома матрицы число умножений слов равно

$$\frac{k}{10}(k + \varphi_n)n^5 + O(k(k + \varphi_n - 2)n^4). \quad (14)$$

Докажем это утверждение. При умножении матрицы A_t на столбец $G_t^{1,t}$ при каждом $p = 1, \dots, t-2$ выполняется t^2 умножений. Компоненты $G_t^{1,t}$ формируются из компонент столбца $H_t^{1,t}$ и f_t^p . Элементы столбца матрицы H получаются p -кратным умножением элементов матрицы A , кроме того, при умножении строки на столбец происходит сложение t чисел. Таким образом, элементы столбца матрицы H содержат $pk + 1 + (p-1)\varphi_n$ слов, f_{t-1}^p слов получается $(t-2)$ -кратным сложением $2(t-2)$ чисел. Это увеличивает размер элементов матрицы G на $1 + \varphi_n$ слов. Таким образом, компоненты столбца $G_t^{1,t}$ содержат $pk + 1 + p\varphi_n$ слов. При умножении строки матрицы A_t на столбец $G_t^{1,t}$ при $p = t-1$ выполняется t умножений. Тогда число умножений слов будет равно

$$\begin{aligned} \sum_{t=2}^n \left(\sum_{p=1}^{t-2} t^2 k(pk + 1 + p\varphi_n) + tk((t-1)k + 1 + (t-1)\varphi_n) \right) \\ = \frac{k}{10}(k + \varphi_n)n^5 + O(k(k + \varphi_n - 2)n^4). \end{aligned}$$

При вычислении характеристического полинома на n -м шаге мы получили последний столбец присоединенной матрицы $G_{1,n}^{1,n}$. Чтобы найти все остальные столбцы присоединенной матрицы, необходимо при каждом $p = 1, \dots, n-2$ вычислить $H_{1,n-1}^{1,n} = AG_{1,n-1}^{1,n}$. На этом шаге нужно выполнить $n^2(n-1)$ умножений коэффициентов. Число умножений слов будет равно

$$\sum_{p=1}^{n-2} n^2(n-1)k(pk + 1 + p\varphi_n) \sim \frac{k}{2}(k + \varphi_n)n^5.$$

Всего для вычисления присоединенной матрицы с использованием алгоритма Сейфуллина необходимо выполнить порядка

$$\frac{3}{5}k(k + \varphi_n)n^5 \quad (15)$$

умножений слов.

Сравнивая выражения (8) и (15), видим, что при нахождении присоединенной матрицы алгоритм Сейфуллина требует в 1,2 раза больше операций над словами, чем алгоритм Леверье–Фаддеева, если использовать стандартное умножение матриц. Если использовать алгоритм Штрассена для умножения матриц в методе Леверье–Фаддеева, то выигрыш времени вычислений метода Леверье–Фаддеева будет увеличиваться с ростом n .

8. Алгоритм Малашонка

Квазитреугольной называется матрица, у которой $a_j^i = 0$ при $i < j + 1$, то есть равны нулю все элементы, расположенные ниже второй диагонали $\{a_1^2, a_2^3, \dots, a_{n-1}^n\}$. У квазитреугольных матриц характеристический полином вычисляется просто.

В основе алгоритма лежит вычисление квазитреугольной матрицы: сначала вычисляется квазитреугольная матрица $A_u = DL^{-1}AL$, где D — диагональная матрица, а затем вычисляется характеристический полином данной матрицы $\det(xD - A_u)/\det D$.

При вычислении квазитреугольной матрицы на шаге t , $t = 1, \dots, n - 2$, вычисляется матрица, у которой обнуляются элементы t -го столбца под второй диагональю. При этом все вычисления происходят в исходном кольце коэффициентов, и все деления являются точными.

Перейдем к описанию алгоритма Малашонка [8, 10].

1. Вычисление квазитреугольной матрицы. Полагаем $A^{[1]} = A$, $D_0 = \Delta_0 = E_n$.

Для $t = 1, 2, \dots, n - 2$ вычисляются последовательно матрицы

$$A^{[t+1]} = D_{t-1}^{-1}(\tilde{L}_t A^{[t]})L_t.$$

Здесь $\alpha^t = (A^{[t]})_t^{t+1}$, $v^t = (A^{[t]})_t^{t+2, n}$ и

$$D_{t-1} = \begin{pmatrix} E_t & 0 \\ 0 & \alpha^{t-1} E_{n-t} \end{pmatrix},$$

$$\tilde{L}_t = \begin{pmatrix} E_t & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -v^t & \alpha^t E_{n-t-1} \end{pmatrix}, \quad (16)$$

$$L_t = \begin{pmatrix} E_t & 0 & 0 \\ 0 & \alpha^t & 0 \\ 0 & v^t & E_{n-t-1} \end{pmatrix}. \quad (17)$$

Последними вычисляются квазитреугольная матрица $A_u = A^{[n-1]}$ и диагональная матрица

$$D = \text{diag}(1, \alpha^1, \alpha^1 \alpha^2, \alpha^2 \alpha^3, \dots, \alpha^{n-3} \alpha^{n-2}, \alpha^{n-2}). \quad (18)$$

2. Вычисление характеристического полинома матрицы A . Вычисляется детерминант $d^n = \det(xD - A_u)$ квазитреугольной матрицы по формуле Хессенберга:

$$d^n = g_n^n d^{n-1} + \sum_{i=1}^{n-1} g_n^i \left(\prod_{j=i+1}^n (-g_{j-1}^j) \right) d^{i-1}, \quad (19)$$

где начальное значение $d^0 = 1$, $x D - A_u = (g_j^i)$, $i, j = 1, \dots, n$. Эта формула получается простым разложением определителя по последнему столбцу, d^i обозначает определитель подматрицы порядка i , стоящей в левом верхнем углу.

Вычисляется характеристический полином матрицы A

$$F(x) = (-1)^n \det(x D - A_u) / \det(D). \quad (20)$$

Если $\alpha^t = 0$ и $v^t = 0$ на некотором шаге t , то нужно положить $A^{[t+1]} = A^{[t]}$. Это означает, что t -й шаг должен быть пропущен. Если же $\alpha^t = 0$ и $(A^{[t]})_t^i \neq 0$ для некоторого $i \geq t + 2$, то положим $\alpha^t = (A^{[t]})_t^i$ и

$$A^{[t+1]} = D_{t-1}^{-1} \tilde{L}_t P_{i,t+1} A^{[t]} P_{i,t+1} L_t,$$

где $P_{i,j}$ обозначает матрицу перестановки, полученную из единичной матрицы E_n после перестановки строк с номерами i и j .

Приведем оценку сложности алгоритма в числе операций над элементами матриц.

На этапе 1 требуется выполнить порядка $\frac{7}{6}n^3$ операций умножения, $\frac{1}{3}n^3$ операций сокращения и $\frac{5}{6}n^3$ операций сложения.

На этапе 2 требуется порядка $\frac{1}{6}n^3$ операций умножения, n операций сокращения и $\frac{1}{6}n^3$ операций сложения. Всего над элементами матрицы нужно выполнить порядка $\frac{5}{3}n^3$ мультипликативных операций и n^3 аддитивных операций.

Найдем оценку сложности алгоритма в числе мультипликативных операций над словами и покажем, что это экспоненциальная функция.

Обозначим через $A_{r,v}^{s,u}$ подматрицу матрицы A , стоящую на пересечении строк s, u и столбцов r, v , $\|A_{r,v}^{s,u}\|$ — число слов в записи элементов этой матрицы.

Запишем матрицу $A^{[t]}$ размера $n \times n$ в блочном виде

$$A^{[t]} = \begin{pmatrix} h & g & c & d \\ 0 & e & a & b \end{pmatrix},$$

где блоки верхней строки содержат t строк, а нижней — $n - t$ строк, блоки d и b содержат $n - t - 1$ столбец, нулевой блок и блок h содержат $t - 1$ столбец, и блоки g, e, c, a состоят из одного столбца.

Пусть a_t — это число слов на шаге t в элементах α^t и v^t , то есть $\|e\| = a_t$. Пусть $\|a\| = \|b\| = b_t$ и $\|c\| = c_t$. Найдем a_{t+1} , b_{t+1} , c_{t+1} для матрицы $A^{[t+1]}$.

Для вычисления матрицы $A^{[t+1]}$ надо выполнить три матричных умножения.

Первое умножение есть

$$B_1 = \tilde{L}_t A^{[t]} = \begin{pmatrix} h & g & c & d \\ 0 & e' & a' & b' \end{pmatrix}.$$

При этом перемножаются $n - (t + 1)$ строк матрицы $\tilde{L}_t$, в каждой из которых два элемента, на матрицу $(A^{[t]})_{t+1,n}^{t+1,n}$. Число умножений слов будет равно

$$2(n - t - 1)(n - t)a_t b_t. \quad (21)$$

В результате $e' = (\alpha_t, 0, 0, \dots, 0)^T$ и

$$\|a'\| = \|b'\| = a_t + b_t.$$

Второе умножение есть

$$B_2 = D_{t-1}^{-1} B_1 = \begin{pmatrix} h & g & c & d \\ 0 & e' & a'' & b'' \end{pmatrix}.$$

Элементы матрицы $(B_1)_{t+1,n}^{t+1,n}$ сокращаются на α_{t-1} . Для этого надо выполнить

$$(n-t)a_{t-1}b_t + (n-t-1)(n-t)a_{t-1}(a_t + b_t) \quad (22)$$

умножений слов. В результате

$$\|a''\| = \|b''\| = \|b'\| - a_{t-1} = a_t + b_t - a_{t-1}.$$

Третье умножение есть

$$A^{[t+1]} = B_2 L_t = \begin{pmatrix} h & g & c' & d \\ 0 & e' & a''' & b'' \end{pmatrix}.$$

При этом перемножаются n строк матрицы B_2 на $(t+1)$ -й столбец матрицы L_t , в котором $n-t$ ненулевых элементов. Следовательно, число умножений слов будет равно

$$t(n-t)a_t c_t + (n-t)(b_t - a_{t-1})a_t + (n-(t+1))(n-t)(a_t + b_t - a_{t-1}). \quad (23)$$

В результате получим

$$\begin{aligned} \|a'''\| &= \|a''\| + a_t + \varphi_n = 2a_t + b_t - a_{t-1} + \varphi_n, \\ \|c'\| &= \|c\| + a_t = a_t + k. \end{aligned}$$

Таким образом,

$$b_{t+1} = a_t + b_t - a_{t-1}, \quad (24)$$

$$a_{t+1} = a_t + b_{t+1} + \varphi_n, \quad (25)$$

$$c_{t+1} = a_t + k.$$

Подставляя (25) в (24), получим $b_{t+1} = 2b_t + \varphi_n$. Так как $b_1 = a_1 = k$, то

$$\begin{aligned} b_{t+1} &= 2^t(k + \varphi_n) - \varphi_n, \\ a_{t+1} &= 2(2^t - 1)(k + \varphi_n), \\ c_{t+1} &= 2(2^t - 1)(k + \varphi_n) + k. \end{aligned}$$

Следовательно, коэффициенты матриц растут экспоненциально.

С учетом (21), (22), (23), получим число умножений слов на шаге t , именно,

$$U_t \sim (k + \varphi_n)^2 4^{t-1} (7n^2 - 12nt + 5t^2).$$

Суммируя по всем шагам $t = 1, \dots, n-2$, получим, что

$$\sum_{t=1}^{n-2} U_t \sim \frac{7}{18} (k + \varphi_n)^2 n 4^{n-1}. \quad (26)$$

Следовательно, сложность алгоритма описывается экспоненциальной функцией (26).

Отметим, что все предыдущие алгоритмы имели полиномиальную сложность, поэтому данный алгоритм является худшим из них при использовании стандартной арифметики в целых числах.

9. Новый алгоритм

Данный алгоритм является развитием предыдущего алгоритма. В основе алгоритма также лежит разложение данной матрицы A в произведение $A = A_u D^{-1}$ квазитреугольной A_u и диагональной D^{-1} матриц [11].

Вычисление матриц A_u и D происходит за $n - 2$ шага. На каждом шаге раньше выполнялось два матричных умножения и умножение на матрицу D_{t-1}^{-1} , которое сводилось к выполнению $(n - t)^2$ операций сокращения. Однако умножение на матрицу D_{t-1}^{-1} можно на каждом шаге не выполнять, но в конце вычислений нужно использовать новую диагональную матрицу, которая получается произведением всех матриц D_{t-1}^{-1} . Для того, чтобы матрицы $\tilde{L}_t$, L_t и D_t остались прежними, нужно на каждом шаге вычислять те же, что и раньше, элементы α^t и v^t , используя для этого $n - t$ операций сокращения.

Опишем новый алгоритм.

1. Вычисление квазитреугольной матрицы. Положим $B^{[1]} = A$.

Для $t = 1, 2, \dots, n - 2$ вычисляются последовательно матрицы

$$B^{[t+1]} = \tilde{L}_t B^{[t]} L_t,$$

где матрицы $\tilde{L}_t$ и L_t определены формулами (16) и (17), но теперь

$$\alpha^t = \begin{cases} (B^{[t]})_t^{t+1}, & t = 1, 2, \\ (B^{[t]})_t^{t+1} / (B^{[t]})_{t-2}^{t-1}, & t = 3, \dots, n - 2, \end{cases}$$

$$v^t = \begin{cases} (B^{[t]})_t^{t+2,n}, & t = 1, 2, \\ (B^{[t]})_t^{t+2,n} / (B^{[t]})_{t-2}^{t-1}, & t = 3, \dots, n - 2. \end{cases}$$

Вычисляются квазитреугольная матрица $A_u = B^{[n-1]}$ и диагональная матрица

$$D = \text{diag}(1, \alpha^1, \alpha^1 \alpha^2, \alpha^1 \alpha^2 \alpha^3, \dots, \alpha^1 \alpha^2 \dots \alpha^{n-2}, \alpha^1 \alpha^2 \dots \alpha^{n-2}). \quad (27)$$

2. Вычисление характеристического полинома матрицы A . Вычисления производятся по формулам (19) и (20).

Найдем оценку сложности алгоритма. На t -м шаге алгоритма вычисляется матрица $B^{[t+1]} = \tilde{L}_t B^{[t]} L_t$, при этом выполняется $(n - t - 1)(n - t) + n(n - t)$ сложений и вычитаний, $2(n - t - 1)(n - t) + n(n - t)$ умножений и $n - t$ делений. Поэтому асимптотические сложности операций при вычислении матрицы $B^{[n-1]}$ следующие: требуется выполнить $\frac{5}{6}n^3 + O(n^2)$ операций сложения и вычитания, $\frac{7}{6}n^3 + O(n^2)$ операций умножения и $\frac{1}{2}n^2$ операций деления. При нахождении матрицы D необходимо выполнить $n - 2$ операции умножения.

На этапе 2 требуется выполнить $\frac{1}{6}n^3 + O(n^2)$ операций сложения и вычитания, $\frac{1}{6}n^3 + O(n^2)$ операций умножения и n операций деления. Всего необходимо $n^3 + O(n^2)$ аддитивных операций и $\frac{4}{3}n^3 + O(n^2)$ операций умножения и $\Theta(n^2)$ операций сокращения в кольце R .

Таблица 1. Время вычисления характеристического полинома с использованием алгоритмов Берковича, Чистова, Леверье–Винограда, Леверье–Фаддеева по отношению к алгоритму Сейфуллина (в процентах)

Размер матрицы	Алгоритм Берковича	Алгоритм Чистова	Метод Леверье–Винограда	Метод Леверье–Фаддеева
30	90%	160%	510%	540%
40	95%	160%	510%	520%
50	99%	170%	510%	520%
60	98%	170%	500%	500%
70	99%	170%	510%	500%
80	104%	180%	550%	510%
90	101%	180%	530%	510%
100	103%	180%	550%	520%
Теоретические оценки	110%	200%	500%	500%

10. Вычислительные эксперименты

Чтобы экспериментально сравнить рассмотренные алгоритмы и проверить найденные теоретические оценки, были проведены три серии экспериментов.

В первой серии участвовали прямые алгоритмы над целыми числами, которые имеют полиномиальную оценку сложности в операциях над словами. Это алгоритмы Сейфуллина, Берковича, Чистова, Леверье–Винограда и Леверье–Фаддеева.

Во второй серии сравнивался лучший прямой алгоритм с алгоритмом с применением КТО, который имеет лучшую оценку по числу кольцевых операций.

В третьей серии сравнивался лучший из рассмотренных алгоритмов с алгоритмами, реализованными в системах *Mathematica 7.0* и *Maple 9.5*.

В экспериментах вычислялся характеристический полином для целочисленных матриц, элементы которых содержат 50 бит. Размеры матриц менялись в пределах от 30 до 100. В табл. 1 указано отношение времени вычисления характеристического полинома с помощью алгоритмов Берковича, Чистова, Леверье, Леверье–Фаддеева к времени вычисления алгоритма Сейфуллина, выраженное в процентах. Для сравнения с экспериментальными результатами в последней строке приведено отношение коэффициентов при старшем слагаемом $k^2 n^5$ в выражениях для сложности алгоритмов Берковича, Чистова, Леверье–Винограда, Леверье–Фаддеева в машинных словах к этому коэффициенту в выражении для сложности алгоритма Сейфуллина, выраженное в процентах. В алгоритмах Леверье–Винограда и Леверье–Фаддеева для умножения матриц используется стандартное умножение матриц.

Как видно из таблицы, эксперименты хорошо согласуются с выводами, сделанными на основе теоретических оценок.

Среди рассмотренных прямых методов вычисления характеристического полинома для матриц с рациональными коэффициентами лучшим прямым алгоритмом является алгоритм Сейфуллина. Если же используется быстрое матричное умножение со сложностью n^β , $2 < \beta < 3$, то лучшими для больших значений n , $n > 4700$, будут алгоритмы Леверье.

Во второй серии экспериментов использовались плотные матрицы над целыми числами длиной 20 бит и 50 бит, которые выбирались случайным образом. Порядок матрицы менялся в пределах от 10 до 350.

Выражения для сложности в кольцевых операциях показывают, что наименьшую сложность имеет новый алгоритм. Он требует $\frac{7}{3}n^3 + O(n^2)$ операций. Во второй серии экспериментов сравнивались новый алгоритм с применением КТО и прямой алгоритм Сейфуллина. Для программной реализации алгоритмов с применением КТО оценка коэффициентов характеристического полинома была взята из работы [12].

Сравнение алгоритма Сейфуллина с новым алгоритмом с применением КТО показывает, что эти алгоритмы требуют примерно одинаковое время, когда размер матриц $n < 60$, при этом алгоритм Сейфуллина может выигрывать не более 10%. Начиная с $n = 60$, новый алгоритм с применением КТО выигрывает у прямого алгоритма Сейфуллина. Например, при $n = 200$ выигрыш составляет 230%.

В третьей серии экспериментов сравнивался новый алгоритм с применением КТО с алгоритмами, которые реализованы в системах Mathematica 7.0 и Maple 9.5. Эксперименты показали, что новый алгоритм вычисляет характеристический полином существенно быстрее, чем Maple и Mathematica. Так, например, для $n = 50$ Mathematica проигрывает в 3,2 раза, а Maple проигрывает в 6,2 раза. Для $n = 100$ новый алгоритм быстрее, чем Mathematica, в 7 раз и быстрее, чем Maple, в 8 раз, а для $n = 200$ — в 15 и 10 раз соответственно.

11. Заключение

В работе получены выражения для сложности алгоритмов вычисления характеристического полинома, которые дают возможность сравнивать алгоритмы, не проводя экспериментов. Эти выражения можно использовать для того, чтобы найти лучший среди прямых алгоритмов и лучший среди алгоритмов с применением КТО.

Теоретические оценки количества умножений слов показали, что асимптотически лучшими среди рассмотренных прямых методов вычисления характеристического полинома для матриц с рациональными коэффициентами являются методы Леверье (методы Леверье, Леверье–Винограда, Леверье–Фаддеева) при использовании алгоритма Штрассена для умножения матриц. При небольших размерах матриц, меньше, чем 4700, лучшим является алгоритм Сейфуллина.

Оценки сложности в кольцевых операциях позволили выбрать алгоритм, который имеет асимптотически лучшее время при применении КТО. Им оказался новый алгоритм. Новый алгоритм с применением КТО существенно быстрее алгоритмов, реализованных в системах Mathematica 7.0 и Maple 9.5.

Проведенные вычислительные эксперименты позволяют дать следующие рекомендации. При вычислении характеристического полинома целочисленных матриц новый алгоритм с применением КТО является самым эффективным, когда размер матрицы более 60. Если размер менее 60, то в зависимости от разрядности чисел он может проиграть алгоритму Сейфуллина, но не более 10%.

Список литературы

1. Le Verrier U. J. J., Sur les variations séculaires des éléments elliptiques des sept planètes principales: Mercure, Vénus, la Terre, Mars, Jupiter, Saturne et Uranus. *J. de Mathématiques Pures et Appliquées* (1840) 5, 220–254.
2. Фаддеев Д. К., Фаддеева В. Н., *Вычислительные методы линейной алгебры*. ГИФМЛ, Москва–Ленинград, 1963.

3. Кнут Д. Э., *Искусство программирования для ЭВМ*, **2**. Мир, Москва, 1977.
4. Strassen V., Gaussian elimination is not optimal. *Numer. Math.* (1969) **13**, 354–356.
5. Chistov A. L., Fast parallel calculation of the rank of matrices over a field of arbitrary characteristic. *Lecture Notes Computer Sci.* (1985) **199**, 147–150.
6. Сейфуллин Т. Р., Вычисление определителя, присоединенной матрицы и характеристического полинома без деления. *Кибернетика и системный анализ* (2002) **38**, №5, 18–42.
7. Berkowitz S. J., On computing the determinant in small parallel time using a small number of processors. *Inform. Process. Lett.* (1984) **18**, 147–150.
8. Malashonok G. I., A computation of the characteristic polynomial of an endomorphism of a free module. *J. Math. Sci.* (2002) **108**, 966–976.
9. Переславцева О. Н., История и современное состояние теории алгоритмов вычисления характеристического полинома матрицы. В сб.: *Труды Международной научной конференции «Современное математическое образование и проблемы истории и методологии математики»*. Першин, Тамбов, 2006, с. 130–134.
10. Малашенок Г. И., *Матричные методы вычислений в коммутативных кольцах*. Изд-во Тамбовского гос. ун-та, Тамбов, 2002.
11. Переславцева О. Н., Метод вычисления характеристического полинома матрицы. *Вестник Тамбовского Университета. Сер. Естественные и Технические Науки* (2008) **13**, №1, 131–133.
12. Dumas J.-G., Pernet C., Wan Z., Efficient computation of the characteristic polynomial. In: *Proc. ISSAC'05* (M. Kauers, ed.). ACM, New York, 2005, pp. 140–147.

Статья поступила 27.02.2009.

Переработанный вариант поступил 29.01.2011.