Эффективные алгоритмы вычисления коэффициентов определителя полиномиальных матриц

Дернин Иван Вадимович

06.11.2026

Полиномиальная матрица

Рассматривается матрица:

$$A(x) \in K[x]^{n \times n}, \quad \deg(A_{ij}) \leq d, \quad d \in \mathbb{N}$$

То есть:

$$A_{ij}(x) = a_{ijd}x^d + a_{ij(d-1)}x^{(d-1)} + \cdots + a_{ij0}$$

Определитель имеет вид:

$$\det(A(x)) = c_m x^m + \cdots + c_k x^k + \cdots + c_1 x + c_0$$

Где $m \leq M$, $M = nd$.

Задача

Найти c_k для заданного k.

Символьный метод Гаусса

$$A(x) = \begin{bmatrix} A_{11}(x) & \cdots & A_{1n}(x) \\ \vdots & \ddots & \vdots \\ A_{n1}(x) & \cdots & A_{nn}(x) \end{bmatrix}$$

$$A'(x) = \begin{bmatrix} A'_{11}(x) & A'_{12}(x) & \cdots & A'_{1n}(x) \\ & A'_{22}(x) & \cdots & A'_{2n}(x) \\ & & \ddots & \vdots \\ 0 & & & A'_{nn}(x) \end{bmatrix}$$

$$\det(A(x)) = \det(A'(x)) = \mathsf{tr}(A'(x))$$

Сложность символьного метода Гаусса

Анализ вычислительных затрат при исключении переменных для шагов $k = 1 \dots (n-1)$:

- **Арифметические операции:** за все шаги алгоритма требуется выполнить $\frac{n(n-1)(2n+5)}{12}$ произведений и $\frac{n(n-1)(2n-1)}{12}$ сложений рациональных функций.
- **Степень функций:** на шаге k степени числителей и знаменателей не превышают $d2^k$.
- **Трудоёмкость базовых операций:** сложность сложения и умножения рациональных функций, степени которых не превышают N, составляет $\mathcal{O}(N^2)$.

Сложность символьного метода Гаусса

Анализ вычислительных затрат при исключении переменных для шагов $k = 1 \dots (n-1)$:

- **Арифметические операции:** за все шаги алгоритма требуется выполнить $\frac{n(n-1)(2n+5)}{12}$ произведений и $\frac{n(n-1)(2n-1)}{12}$ сложений рациональных функций.
- **Степень функций:** на шаге k степени числителей и знаменателей не превышают $d2^k$.
- **Трудоёмкость базовых операций:** сложность сложения и умножения рациональных функций, степени которых не превышают N, составляет $\mathcal{O}(N^2)$.

Из-за экспоненциального роста степеней промежуточных многочленов итоговая сложность:

$$\mathcal{O}\left(\frac{1}{6}n^3 d^2 2^{2n}\right)$$

Критический фактор: экспоненциальная зависимость от размера матрицы 2^{2n}.

Ограничения символьного метода Гаусса

Классическое исключение Гаусса над кольцом многочленов сталкивается с рядом критических проблем:

1. Появление рациональных функций

В процессе деления на ведущие элементы промежуточные коэффициенты становятся дробями вида $P(x)/Q(x)$, что требует постоянного поиска НОД для сокращения.

2. Избыточность вычислений

Символьный метод вынужден последовательно находить абсолютно все промежуточные полиномы, даже если требуется узнать всего один конкретный коэффициент c_k.

3. Вычислительный взрыв степени

Степень числителей и знаменателей на каждом шаге растет экспоненциально (до $d2^k$), приводя к лавинообразному росту объёма требуемой памяти и времени процессора.

Мотивация: Необходим переход к методам без деления или интерполяции.

Алгоритм Барейса (Fraction-Free Gaussian Elimination)

Инициализация:

$$A^{(0)}(x) = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & A_{11}(x) & \cdots & A_{1n}(x) \\ \vdots & \vdots & \ddots & \vdots \\ 0 & A_{n1}(x) & \cdots & A_{nn}(x) \end{bmatrix} = \begin{bmatrix} a_{00}^{(0)} & \cdots & a_{0n}^{(0)} \\ \vdots & \ddots & \vdots \\ a_{n0}^{(0)} & \cdots & a_{nn}^{(0)} \end{bmatrix}$$

Шаг $k = 2, 4, \ldots, 2\left[\frac{n}{2}\right]$:

$$c_0^{(k-2)}(x) = \frac{1}{a_{k-2,k-2}^{(k-2)}(x)} \begin{vmatrix} a_{k-1,k-1}^{(k-2)}(x) & a_{k-1,k}^{(k-2)}(x) \\ a_{k,k-1}^{(k-2)}(x) & a_{k,k}^{(k-2)}(x) \end{vmatrix}$$

$$c_{i1}^{(k-2)}(x) = \frac{1}{a_{k-2,k-2}^{(k-2)}(x)} \begin{vmatrix} a_{k-1,k}^{(k-2)}(x) & a_{k-1,k-1}^{(k-2)}(x) \\ a_{ik}^{(k-2)}(x) & a_{i,k-1}^{(k-2)}(x) \end{vmatrix}$$

$$c_{i2}^{(k-2)}(x) = \frac{1}{a_{k-2,k-2}^{(k-2)}(x)} \begin{vmatrix} a_{k,k-1}^{(k-2)}(x) & a_{k,k}^{(k-2)}(x) \\ a_{i,k-1}^{(k-2)}(x) & a_{ik}^{(k-2)}(x) \end{vmatrix}$$

Алгоритм Барейса (Fraction-Free Gaussian Elimination)

$$a_{ij}^{(k)}(x) = \frac{1}{a_{k-2,k-2}^{(k-2)}(x)} \left(a_{ij}^{(k-2)}(x) c_0^{(k-2)}(x) + a_{kj}^{(k-2)}(x) c_{i1}^{(k-2)}(x) + a_{k-1,j}^{(k-2)}(x) c_{i2}^{(k-2)}(x) \right)$$

для $i = k+1, \ldots, n$; $j = k+1, \ldots, n$

$$a_{kl}^{(k-1)}(x) = \frac{1}{a_{k-2,k-2}^{(k-2)}} \begin{vmatrix} a_{k-1,k-1}^{(k-2)} & a_{k-1,l}^{(k-2)} \\ a_{k,k-1}^{(k-2)} & a_{kl}^{(k-2)} \end{vmatrix} = a_{kl}^{(k)}(x)$$

для $l = k, \ldots, n$

Итог

$$a_{nn}^{(n-1)}(x) = \det(A(x))$$

Пример

$$A^{(0)}(x) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & x & -x^3 & -2x^4 \\ 0 & 1 & -1 & -2x \\ 0 & -3 & x^2+x & 2x^2 \end{bmatrix}$$

Шаг $k = 2$:

$$c_0^{(0)}(x) = \frac{1}{1}\begin{vmatrix} x & -x^3 \\ 1 & -1 \end{vmatrix} = x^3 - x$$

$$c_{31}^{(0)}(x) = \frac{1}{1}\begin{vmatrix} -x^3 & x \\ x^2+x & -3 \end{vmatrix} = 2x^3 - x^2$$

$$c_{32}^{(0)}(x) = \frac{1}{1}\begin{vmatrix} 1 & -1 \\ -3 & x^2+x \end{vmatrix} = x^2 + x - 3$$

$$a_{33}^{(2)}(x) = \frac{1}{1}\left(2x^2(x^3-x) - 2x(2x^3-x^2) - 2x^4(x^2+x-3)\right) = -2x^6 + 2x^4$$

$$a_{22}^{(1)}(x) = \begin{vmatrix} x & -x^3 \\ 1 & -1 \end{vmatrix} = 2x^3 - x = a_{22}^{(2)}(x)$$

$$a_{23}^{(1)}(x) = \begin{vmatrix} x & -2x^4 \\ 1 & -2x \end{vmatrix} = 2x^4 - 2x^2 = a_{23}^{(2)}(x)$$

Сложность алгоритма Барейса

На каждом шаге необходимо сложений/вычитаний, произведений и делений без остатка полиномиальных функций степени не более $2d(k/2+1)$. Если $m = \left[\frac{n}{2}\right]$ и $l = \frac{k}{2}$, то суммарно требуется:

- $\frac{(m-1)(8m^2+5m+6)}{3}$ сложений и вычитаний
- $4(m-1)(m^2+m+1)$ произведений
- $\frac{(m-1)(4m^2+7m+6)}{3}$ делений

Сложность сложения/вычитания $\mathcal{O}(N)$, произведения $M(N) = \mathcal{O}(N \cdot \log N)$, деления $D(N) \geq M(N)$ для двух полиномов длины не более N.

Сложность алгоритма

$$\mathcal{O}\left(\frac{4}{3}m^3 D(2d(m+1)) + 4m^3 M(2d(m+1))\right) = \mathcal{O}\left(\frac{1}{6}n^3 D(nd) + \frac{1}{2}n^3 M(nd)\right)$$

Метод интерполяции

Переход от символьных вычислений с многочленами к работе с числовыми значениями:

Основная идея метода

Вместо сложного раскрытия определителя в общем виде, мы вычисляем числовое значение детерминанта в фиксированных точках x_i, а затем восстанавливаем искомый полином.

Критическое условие

Поскольку максимальная степень определителя ограничена числом $M = nd$, для однозначного восстановления всех коэффициентов строго необходимо вычислить значения матрицы в $M + 1$ точках.

Этапы схемы вычислений

1. **Фиксация сетки:** Выбор точек интерполяции $x_0, x_1, \ldots, x_M$.
2. **Редукция задачи:** Вычисление определителей обычных числовых матриц $A(x_i) \in K^{n \times n}$.
3. **Сбор данных:** Получение набора пар значений (x_i, y_i).
4. **Синтез:** Решение системы или применение быстрых полиномиальных алгоритмов.

Интерполяция

Определитель имеет вид:

$$\det(A(x)) = c_m x^m + \cdots + c_1 x + c_0, \quad m \le M,\ M = nd$$

Система из $M+1$ уравнений с $m+1$ неизвестными:

$$\begin{cases} c_m x_0^m + \cdots + c_1 x_0 + c_0 = y_0 \\ \qquad\qquad \vdots \\ c_m x_M^m + \cdots + c_1 x_M + c_0 = y_M \end{cases}$$

имеет точное решение. Достаточно вычислить значения определителя в $M+1$ точках.
Сложность вычисления определителя: $\mathcal{O}(n^\omega)$.

Если решать систему методом Гаусса: $\mathcal{O}(n^\omega(nd) + (nd)^3)$ операций.
Для оптимизации можно применить интерполяцию (Ньютона или Лагранжа).

Метод усеченной интерполяции

Идея

Для вычисления коэффициентов, соответствующих малым значениям k, достаточно использовать слагаемые в полиномах степени не более k.

Переход от $A_{ij}(x) = a_{ijd}x^d + \cdots + a_{ijk}x^k + \cdots + a_{ij0}$ к $A_{ij}^{(k)}(x) = a_{ijk}x^k + \cdots + a_{ij0}$. Далее интерполяция. В случае субмаксимальных значений k (близких к nd), достаточно

рассматривать $l = nd - k + 1$ старших коэффициентов:

$$A'_{ij}(x) = x^{(d-l+1)}(a_{ijd}x^{l-1} + \cdots + a_{ij(d-l+1)}) = x^{(d-l+1)}A''_{ij}(x)$$

Необходимый коэффициент будет при степени $(n-1)(l-1)$.
В формулах сложности d заменяется на k или l.

Определитель прямой линии

Пусть $MM(n,d) = \mathcal{O}(n^\omega d \log d \log\log d)$ — сложность произведения матриц.

$$MM'(n,d) = \sum_{i=0}^{\log n} 2^i MM(n, 2^{-i}d)$$

Идея

Представить $A^{-1}(x) = B^{-1}(x)U(x)$, где $B(x)$ — диагональная матрица, $\deg B_{ij} = \deg \det(A(x)) \le nd$.

Диагонализируем матрицу $A(x)$:

$$A \to UA = \begin{bmatrix} \bar{U} \\ \underline{U} \end{bmatrix} [A_L \quad A_R] = \begin{bmatrix} \bar{U}A_L & \\ & \underline{U}A_R \end{bmatrix}$$

где $\bar{U}(x), \underline{U}(x)$ — минимальные базисы ядер матриц $A_R(x)$ и $A_L(x)$.

Определитель прямой линии

Вход: $A(x) \in K[x]^{n \times n}$, $\deg A_{ij} \le d$, $\det A(x) \neq 0$, $\log n \in \mathbb{N}$
Выход: $\det A(x)$

1. $B \coloneqq \mathsf{copy}(A)$
2. Шаги от 1 до $\log_2 n$:

$$B \coloneqq \begin{bmatrix} B_i^{(1)} & & 0 \\ & \ddots & \\ 0 & & B_i^{(2^{i-1})} \end{bmatrix}, \quad U_i \coloneqq \begin{bmatrix} \bar{U}_i^{(1)} & & \\ \underline{U}_i^{(1)} & & \\ & \ddots & \\ & & \bar{U}_i^{(2^{i-1})} \\ & & \underline{U}_i^{(2^{i-1})} \end{bmatrix}$$

   где $\bar{U}_i^{(j)}$ и $\underline{U}_i^{(j)}$ — минимальные базисы ядер матриц $B_{i,L}^{(j)}$ и $B_{i,R}^{(j)}$.
3. $B \coloneqq U_i B$
4. Возвращаем $(\det A(C)) b_{n,n} / b_{n,n}(C)$

Пример реализации схемы

$$B = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & x & -x^3 & -2x^4 \\ 0 & 1 & -1 & -2x \\ 0 & -3 & x^2 + x & 2x^2 \end{bmatrix}$$

Шаг 1:

$$\bar{U}_1^{(1)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & x^3 & 0 \end{bmatrix}, \quad \underline{U}_1^{(1)} = \begin{bmatrix} 0 & -1 & x & 0 \\ 0 & 0 & 3 & 1 \end{bmatrix}$$

$$B \coloneqq U_1 B = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & x^3 - x & 0 & 0 \\ 0 & 0 & x^3 - x & 2x^4 - 2x^2 \\ 0 & 0 & x^2 + x - 3 & 2x^2 - 6x \end{bmatrix}$$

Шаг 2:

$$\bar{U}_2^{(1)} = [1 \quad 0], \quad \underline{U}_2^{(1)} = [0 \quad 1], \quad \bar{U}_2^{(2)} = [x - 3 \quad x - x^3], \quad \underline{U}_2^{(2)} = [x^2 + x - 3 \quad x - x^3]$$

$$B \coloneqq U_2 B = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & x^3 - x & 0 & 0 \\ 0 & 0 & x^3 - x^5 & 0 \\ 0 & 0 & 0 & 2x^6 - 2x^4 \end{bmatrix}$$

$$\det A(2) = -96; \quad b_{4,4}(2) = 96 \implies \frac{\det A(2) b_{4,4}}{b_{4,4}(2)} = -2x^6 + 2x^4$$

Сложность алгоритма

$\bar{U}_i^{(j)}$ и $\underline{U}_i^{(j)}$ могут быть подсчитаны за $\mathcal{O}(MM'(2^{-i}n, 2^i d))$ операций.
Матричное произведение: $\mathcal{O}(2^i MM(2^{-i}n, 2^i d))$ операций, где $MM(n,d) \leq MM'(n,d)$.
Сложность вычисления $\det A(0)$ ограничена $\mathcal{O}(n^\omega)$.

Суммарная сложность

$$\mathcal{O}\left(\sum_{i=1}^{\log n} 2^i MM'(2^{-i}n, 2^i d)\right) \quad \text{или} \quad \mathcal{O}(n^\omega d \log n (\log d)^2 \log\log d)$$

Метод Чжоу-Лабана: Смещённые степени

Пусть $\mathbf{p} = [p_1 \ \cdots \ p_n]^T$. Степень вектор-столбца: $\operatorname{cdeg} \mathbf{p} = \max_{1 \le i \le n} \deg p_i$.
Пусть $\vec{\mathbf{s}} = [s_1 \ \cdots \ s_n]^T \in \mathbb{Z}^n$ – вектор смещения.

$$\operatorname{cdeg}_{\vec{\mathbf{s}}} \mathbf{p} = \max_{1 \le i \le n} (\deg p_i + s_i) = \deg(x^{\vec{\mathbf{s}}} \cdot \mathbf{p})$$

где $x^{\vec{\mathbf{s}}} = \operatorname{diag}(x^{s_1}, \ldots, x^{s_n})$.

- Матрица A **столбцово-приведённая**, если матрица старших коэффициентов столбцов $[\operatorname{coeff}(f, x, d_j)]$, где $\vec{\mathbf{s}} = \operatorname{cdeg} A$, имеет полный ранг.
- Матрица A $\vec{\mathbf{s}}$**-столбцово-приведённая**, если $x^{\vec{\mathbf{s}}} A$ столбцово-приведённая.
- N — это $\vec{\mathbf{s}}$**-минимальный ядерный базис** A, если N является $\vec{\mathbf{s}}$-столбцово-приведённой и $AN = 0$.

Метод Чжоу-Лабана: Рекурсивные вычисления

$$A \to AU = \begin{bmatrix} A_U \\ A_D \end{bmatrix} \begin{bmatrix} U_L & U_R \end{bmatrix} = \begin{bmatrix} G_1 & 0 \\ * & G_2 \end{bmatrix} = G$$

G_1 — невырожденный базис столбцов A_U, вычисляется за $\mathcal{O}(n^\omega s \log s \log\log s)$, где $s = \frac{\sum s_i}{n}$.
$U_R = N$ — базис ядра A_U; $G_2 = A_D \cdot N$ вычисляется за $\mathcal{O}(n^\omega s \log s \log\log s)$. Для детерминанта верно: $\det A = \frac{\det G}{\det U} = \frac{\det G_1 \cdot \det G_2}{\det U}$. Пусть $U^* = [U_L^* \quad U_R]$, тогда:

$$\det A = \frac{\det G \cdot \det(V_U U_L^*)}{\det U^*}$$

Для вычисления U_L^* применяется преобразование Гаусса-Жордана (сложность $\mathcal{O}(nm^{\omega-1})$).

Метод Чжоу-Лабана: Алгоритм

Алгоритм Determinant(A)
Вход: $A(x) \in K[x]^{n \times n}, \det A(x) \neq 0$
Выход: $\det A(x)$

1. Если $n = 1$, вернуть A;
2. $\begin{bmatrix} A_U \\ A_D \end{bmatrix} := A$, где $A_U \in K[x]^{n \times \frac{n}{2}}$;
3. $G_1, U_R, V_U :=$ ColumnBasis(A_U) (вычисляем столбцовый базис G_1, ядерный базис U_R и правый фактор V_U);
4. $G_2 := A_D U_R; \quad U_R := U_R \pmod{x}; \quad V_U := V_U \pmod{x};$
5. Вычисляем U_L^*, такую что $U^* = [U_L^* \quad U_R]$ унимодулярна;
6. $d_V := \det(V_U U_L^*) / \det(U^*)$;
7. $d_G :=$ Determinant$(G_1) \cdot$ Determinant(G_2);
8. Возвращаем $d_V \cdot d_G$;

Метод Чжоу-Лабана: Сложность

Сложность на текущем шаге рекурсии алгоритма оценивается следующим соотношением:

$$g(m) = \mathcal{O}(m^{\omega-1} ns \log \frac{ns}{m} \log\log \frac{ns}{m}) + 2g([m/2]) = \mathcal{O}(m^{\omega-1} ns \log \frac{ns}{m} \log\log \frac{ns}{m})$$

Итоговая сложность при $m = n$

$$\mathcal{O}(n^{\omega} s \log s \log\log s)$$

Пример метода Чжоу-Лабана

$$A = \begin{bmatrix} x & -x^3 & -2x^4 & 2x & -x^2 \\ 1 & -1 & -2x & 2 & -x \\ -3 & 3x^2+x & 2x^2 & -x^4+1 & 3x \\ 0 & 1 & x^2+2x-2 & x^3+2x-2 & 0 \\ 1 & -x^2+2 & -2x^3-3x+3 & 2x+2 & 0 \end{bmatrix}$$ в кольце $\mathbb{Z}_7[x]$

$$G_1 = \begin{bmatrix} x & -x^3 & -2x^4 \\ 1 & -1 & -2x \\ -3 & 3x^2+x & 2x^2 \end{bmatrix}; \quad U_R = \begin{bmatrix} -1 & x \\ -x^2 & 0 \\ -3x & 0 \\ -3 & 0 \\ 0 & 1 \end{bmatrix}; \quad V_U = \begin{bmatrix} 1 & 0 & 0 & 2 & -x \\ 0 & 1 & 0 & 2x^2 & 0 \\ 0 & 0 & 1 & -x & 0 \end{bmatrix}$$

После взятия по модулю $(\mathrm{mod}\ x)$:

$$U_R \coloneqq U_R \pmod{x} = \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & 0 \\ -3 & 0 \\ 0 & 1 \end{bmatrix}; \quad V_U \coloneqq V_U \pmod{x} = \begin{bmatrix} 1 & 0 & 0 & 2 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Пример метода Чжоу-Лабана (продолжение)

Дополнение:

$$U_L^* = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \implies \det V = \frac{\det(V_U U_L^*)}{\det(U^*)} = -\frac{1}{3} \equiv 2 \pmod 7$$

Применив алгоритм к G_1 и G_2:

$$\det G_1 = x^6 - x^4, \quad \det G_2 = x^4 - x$$

Результат

$$\det A = \det V \cdot \det G_1 \cdot \det G_2 = 2(x^6 - x^4)(x^4 - x) = 2x^{10} - 2x^8 - 2x^7 + 2x^5$$

Сравнение сложности точных алгоритмов

Сравнительный анализ вычислительной сложности алгоритмов для полиномиальной матрицы размера $n \times n$ и максимальной степени элементов d:

Алгоритм	**Вычислительная сложность**	**Особенности метода**
Барейс	$\mathcal{O}\left(\frac{1}{6}n^3 D(nd) + \frac{1}{2}n^3 M(nd)\right)$	Fraction-Free исключение
Прямая линия	$\mathcal{O}(n^\omega d \log n (\log d)^2 \log\log d)$	Минимальные базисы ядер
Чжоу-Лабан	$\mathcal{O}(n^\omega s \log s \log\log s)$	Смещённые степени ($s = \frac{\sum s_i}{n}$)

* Здесь $2 \leq \omega \leq 3$ — константа быстрого матричного умножения, $M(N)$ и $D(N)$ — сложности умножения и деления полиномов длины N.

Области применения

Разработанные алгоритмы точного вычисления определителя полиномиальных матриц находят широкое применение в прикладных задачах:

1. Компьютерная алгебра

Вычисление характеристических многочленов матриц с символьными или полиномиальными коэффициентами без потери точности.

2. Робототехника

Моделирование динамики сложных кинематических систем, звенья которых описываются дифференциально-геометрическими уравнениями.

3. Теория управления

Анализ устойчивости многомерных систем автоматического регулирования (САР) и вычисление передаточных функций.

Эффективность $\Longrightarrow$ **Скорость анализа**

Литература

1. Bareiss E. H. Sylvester's identity and multistep integer-preserving Gaussian elimination // Math. of Comp. – 1968. – Vol. 22, No. 103. – P. 565-578.
2. Björck Å., Pereyra V. Solution of Vandermonde systems of equations // Math. of Comp. – 1970. – Vol. 24, No. 112. – P. 893-903.
3. Qin X. et al. Computing the determinant of a matrix with polynomial entries by approximation // Journal of Systems Science and Complexity. – 2018. – Vol. 31, No. 2. – P. 508-526.
4. Marco A., Martínez J. J. Parallel computation of determinants of matrices with polynomial entries // Journal of Symbolic Computation. – 2004. – Vol. 37. – №. 6. – С. 749-760.
5. Jeannerod C. P., Villard G. Straight-line computation of the polynomial matrix inverse : дис. – Laboratoire de l'informatique du parallélisme, 2002.
6. Giorgi P., Jeannerod C. P., Villard G. On the complexity of polynomial matrix computations // Proceedings of the 2003 international symposium on Symbolic and algebraic computation. – 2003. – С. 135-142.
7. Storjohann A. Algorithms for matrix canonical forms : дис. – ETH Zurich, 2000.
8. Zhou W., Labahn G. Fast and deterministic computation of the determinant of a polynomial