ON COMPUTING THE DETERMINANT OF MATRICES WITH POLYNOMIAL ENTRIES

E. Horowitz

S. Sahni

TR 73-180

December 1972

Department of Computer Science
Cornell University
Ithaca, New York 14850

ON COMPUTING THE DETERMINANT OF MATRICES WITH POLYNOMIAL ENTRIES

E. Horowitz

S. Sahni

Abstract:

We consider the problem of computing the determinant of a matrix of polynomials. Four algorithms are compared: expansion by minors, Gaussian elimination over the integers, a method based on evaluation and interpolation, and a procedure which computes the characteristic polynomial of the matrix. Each method is analyzed with respect to its computing time and storage requirements using several models for polynomial growth. The results show which method is preferable for a given computational model. In addition to these asymptotic results, the analysis is exactly done for certain especially small, yet practical and important cases.

CR Categories: 5.14, 5.25.

Key Words: determinants, matrix of polynomials, Gaussian elimination, expansion by minors, characteristic polynomial.

ON COMPUTING THE DETERMINANT OF MATRICES
WITH POLYNOMIAL ENTRIES

E. Horowitz
S. Sahni

Introduction

Though much work has been done on the problem of computing solutions of linear systems with rational number entries, far less has been done for matrices with polynomial entries. However, 2 recent papers have begun to treat this problem, namely Gentleman and Johnson in [8] and McClellan in [7]. The need for such algorithms arise in many areas of application. One such important area is the field of symbolic mathematics via computer. Software systems are developed for performing a variety of operations on symbolic mathematical entities such as polynomials and rational functions. For example one might wish to re-determine the formulas for the inverse of the Vandermonde matrix:

$$\begin{bmatrix} 1, & 1, \ldots, & 1 \\ x_1, & x_2, \ldots, & x_n \\ \vdots & & \vdots \\ x_1^{n-1}, & x_2^{n-1}, \ldots, & x_n^{n-1} \end{bmatrix}$$

(such formulas have already been derived). By using a symbolic mathematical computer system one can produce, for a fixed n, the elements in the inverse which are rational functions in the variables $x_1, \ldots, x_n$. The algebraic manipulation package must solve a linear system with polynomial entries in n variables. Generalizing from this specific example one can imagine the importance of efficient algorithms for performing such operations. In this paper we will specifically consider the problem of determinant calculation for matrices of polynomials.

Historically, McClellan in [7] has presented an algorithm based on evaluation and interpolation. He analyzed his method assuming that the polynomials always remained completely dense (i.e. all possible terms occur) and compared it to using Gaussian elimination. Gentleman and Johnson in [8] consider Gaussian elimination and expansion by minors. Their analysis is done for dense polynomials in one variable and for "completely sparse" polynomials. In this paper we compare four methods: an evaluation — interpolation scheme, Gaussian elimination, expansion by minors and computing the characteristic polynomial. All of these algorithms are analyzed using 3 polynomial models, two of which are generalizations of the completely dense and completely sparse sub-cases. Thus the results of McClellan, Gentleman and Johnson are derived as special cases. In addition to an analysis of the computing time for these methods we also analyze their respective storage requirements. Finally the exact computing time, as opposed to the asymptotic time, is worked out for several important cases.

2. The Algorithms

In this section we shall describe and analyze four different algorithms for computing the determinant of a matrix.

All of these methods have been known before, but their use on matrices with polynomials of different types has only partially been studied.

For all the algorithms the input is an nxn matrix M whose elements are polynomials in the variables $x_1,\ldots,x_r$. The output is an r-variable polynomial $R(x_1,\ldots,x_r)$ such that $R = \det(M)$. The coefficients of the polynomials may be

a) fixed or floating point single precision numbers;

or

b) elements from the finite field, GF(p), where p is a single precision, odd prime.

Recent work studying algorithms with polynomial computations has shown that polynomials with multiple precision integer coefficients are best handled by reducing these coefficients modulo a set of primes p_i and performing the operations over $GF(p_i)$. Thus unless explicitly stated we will always assume that coefficients come from categories (a) or (b) and we note that the time for the arithmetic operations on these elements is bounded by a constant.

While analyzing the computing times of these algorithms only multiplications will be counted and we shall assume that initially all the elements of M have the same number of terms. We shall also assume that polynomial multiplication is carried out using the classical algorithm (i.e. the time required to multiply a p term by a q term polynomial is $p\cdot q$).

The following notation will be used:

1) $|P(x_1,\ldots,x_r)|$ is the number of terms of the polynomial P;

2) $T(P,Q)$ is the number of coefficient multiplications required to multiply polynomials P and Q or $|P|\cdot|Q|$;

3) $e(M)$ is an element of the matrix M.

While analyzing the storage requirements we shall assume that we are not permitted to alter the original matrix M. The storage required to store M will not be explicitly considered. Then

4) $S^{(n)}_{(y)}$ is the storage required by algorithm y using model n;

5) deg(P) in x_i is the degree of P in its i-th variable.

2.1 Homomorphism Method

The use of this method for polynomials was first proposed by G. Collins and more fully investigated by McClellan in [7]. The determinant is computed by evaluating M at a sufficient number of points obtaining related matrices with single precision elements. Then the determinants of these matrices are computed and the results are interpolated to produce the required determinant.

A more precise specification of this algorithm is now presented:

<u>Algorithm Homomorphism(M)</u>

Input: M an nxn matrix, $e(M)$ polynomials in r variables with coefficients in GF(p).

Output: $R(x_1,\ldots,x_r) = \det(M)$.

1) [Initial Case] If $r = 0$ compute $R \leftarrow \det(M)$ using Gaussian Elimination over a field;

2) [Initialize] [Interpolation] $m_r \leftarrow \max\{\deg(e(M))$ in $x_r\} + 1$; $C(x_1,\ldots,x_r) \leftarrow 0$, $D(x_r) \leftarrow 1$; $b \leftarrow -1$

3) [Choose Next Point] $b = b+1$;

4) [Evaluation] [Homomorphism] $\bar{M}(x_1,\ldots,x_{r-1}) \leftarrow M(x_1,\ldots,x_{r-1},b)$ (evaluate every element of M at b for its r-th variable)

5) [Recursion] $R(x_1,\ldots,x_{r-1}) \leftarrow \text{HOMOMORPHISM}(\bar{M})$;

6) [Interpolate] [the determinant] $C(x_1,\ldots,x_r) \leftarrow \{R(x_1,\ldots,x_{r-1}) - C(x_1,\ldots,x_{r-1},b) * \{D(x_r)/D(b)\} + C(x_1,\ldots,x_r)$;

7) [Done?] If $\deg(D) \leq m_r n$ then $(D(x_r) \leftarrow (x_r-b)\, D(x_r)$; go to (3);)

8) [End!] $R(x_1,\ldots,x_r) \leftarrow C(x_1,\ldots,x_r)$; End.

A proof that this algorithm works can be found in [7]. Note that the main loop between steps (3) - (7) is governed by the maximum degree in the main variable of $e(M)$ and the size of the matrix.

Computing Time Analysis

We assume that each element of M has degree m-1 in each of the variables $x_1, \ldots, x_r$ (and hence has at most m^r terms) Thus det(M) has degree at most $(m-1)n$ in each variable and at most $((m-1)n+1)^r$ terms.

Let t_r be the number of multiplications required to compute the determinant of an nxn matrix with r-variable polynomials using the Homomorphism algorithm and let $a = (m-1)n+1$. Then we have:

Step	Number of Multiplications	Comments
(4)	$a[(m-2) + n^2 m^{r-1}(m-1)]$	powers of x_r + multiplications
(5)	$a\, t_{r-1}$	Recursion
(6)	$\sum_{i=3}^{a} (i-2)a^{r-1} + (i-3)$	evaluating $C(x_1,\ldots,x_{r-1},b)$
	$\sum_{i=1}^{a} i + \sum_{i=3}^{a} (i-2)$	$D(x_r)/D(b)$
	$\sum_{i=1}^{a} i\, a^{r-1}$	$(R-C)*D(x_r)/D(b)$

Thus, $t_r = at_{r-1} + a(m-2) + a\, n^2 m^{r-1} (m-1)$

$$+ \frac{(a-1)(a-2)}{2} a^{r-1} + \frac{(a-2)(a-3)}{2} + a(a+1)$$

$$+ \frac{(a-1)(a-2)}{2} + \frac{a(a+1)}{2} a^{r-1}$$

$$= a\, t_{r-1} + a(m-2) + a\, n^2 m^{r-1} (m-1) + (a^2-a+1)a^{r-1} + (2a^2 - 3a + 4).$$

Working out the recurrence relation one gets

$$t_r = a^r t_0 + r(a^2 - a + 1)a^{r-1} + am \frac{a^{r-1}}{a-1} + (2a^2-5a + r) \frac{a^r-1}{a-1} + a\, m^{r-1} n^2 (m-1) \frac{(\frac{a}{m})^r-1}{(\frac{a}{m})-1} + a\, m^{r-1} n^2 (m-1) \frac{(\frac{a}{m})^r-1}{(\frac{a}{m})-1} \quad (1)$$

We have maintained this rather unwieldy expression in order to retain the constant. However we note that

$$t_0 = n(n-1)(n+1)/3 \quad (2)$$

and hence the asymptotic time for the algorithm is

$$t_r = O((mn)^r (n^3 + mn^2)) \quad (3)$$

<u>Storage</u>: Bounded by $r(|R| + n^2|e(M)|)$

$$\therefore \quad S_{Hom} = O(ra^r) \quad (4)$$

In order to use for later comparison we now give the formulas of the exact computing time for the Homomorphism method for the special cases of one and two variables ($r=1$ and $r=2$). We note that these exact formulas are derived assuming elements over a field.

For one variable the time is t_1 =

$$a\left(\frac{n(n-1)(n+1)}{3} + 3a - 6 + m + n^2(m-1)\right) + 5 \qquad (5)$$

and for two variables the time is t_2 =

$$\frac{a^2n(n-1)(n+1)}{3} + an^2(m-1)(a+m) + am(a+1) + 4a^3 - 5a^2 + a + 4 \qquad (6)$$

2.2 Gaussian Elimination with Exact Division

This is the standard algorithm that is used for computing the determinant while remaining over the integers. A proof that this works and that the division in step D below is exact may be found in [1].

Algorithm Exact Division

Let $A_{ij}^{(1)}$ represent the (i,j)th element of matrix M.

Let $A_{0,0}^{(0)} = 1$

For i = 1 TO n-1 DO;

Begin For k = i+1 TO n DO;

Begin For j = i+1 to n DO;

Step D . . . $A_{kj}^{(i+1)} = \dfrac{A_{ki}^{(i)} A_{ij}^{(i)} - A_{ii}^{(i)} A_{kj}^{(i)}}{A_{i-1,i-1}^{(i-1)}}$;

End;

End;

$R(x_1,\ldots,x_r) = A_{nn}^{(n)}$

Computing Time Analysis

Bareiss [1] and Lipson [6] show that each $A_{ij}^{(i)}$ is a determinant (minor) of some i by i submatrix of the original matrix. Since we assumed all entries in the original matrix to be of the same size we may expect that all elements $A_{ij}^{(i)}$, for a given i, are the same size. Though this is not true in general, it does provide a worst case analysis.

For each k we have $(n-i)^2$ executions of step D. We assume that an exact division of A by B to yield C has the same cost as a multiplication of B by C. Then

$$t(\text{Gauss. elim.}) = \sum_{i=1}^{n-1} (n-i)^2 (2T(A_{kj}^{(i)}, A_{kj}^{(i)}) + T(A_{kj}^{(i+1)}, A_{kj}^{(i-1)})) \quad (7)$$

Storage:

$$S(\text{Gauss. elim.}) \leq \max_{0 \leq i \leq n-1} \left\{ (n-i)^2 \ |A_{kj}^{(i+1)}| \right\} \quad (8)$$

2.3 Characteristic Polynomial

Here we compute the characteristic polynomial $P(\lambda) = \det(M-\lambda I)$ of the matrix M. The coefficient of λ^0 is the determinant of M. This method is described in detail in [5]. As may be expected from the results of [2], [3], and [4] the $O(n^4)$ method for matrices with integer elements described in [5] turns out to be better than the $O(n^{3.5})$ method when applied to an M whose entries are polynomials in at least one variable.

Algorithm Characteristic Polynomial

1) [Compute trace of powers of M] $S(1) \leftarrow \text{Trace}(M)$; $A \leftarrow M$;

For $i = 2$ TO r DO;

Begin $A^i \leftarrow M * A^{i-1}$;

$S(i) \leftarrow \text{Trace}(A^i)$; End;

2) [Compute coefficients of characteristic polynomial of M : $P(\lambda) = \Sigma r_i \lambda^{n-i}$] $r(1) \leftarrow -S(1)$

For $i = 2$ TO n DO;

$r(i) = -\sum_{j=1}^{i-1} r(j)\ S(i-j)/i$;

3) [Return determinant] $R(x_1,\ldots,x_r) \leftarrow r(n)$; End.

Computing Time Analysis:

$$t(\text{step 1}) = n^3 \sum_{i=1}^{n-1} T\,(e(m),\, e(A^i))$$

$$t(\text{step 2}) = \sum_{i=2}^{n} \left\{ \sum_{j=1}^{i-1} T(r(j), S(i-j)) + |r(i)| \right\}$$

$$t(\text{charac. poly}) = t(\text{step 1}) + t(\text{step 2}) \tag{9}$$

Storage: We assume $|e(A^i)| \geq |e(A^{i-1})|$ for $i > 1$

Then storage for step 1 $\leq n^2|e(A^n)| + n^2|e(A^{n-1})|$
$$+ \sum_{i=1}^{n} |s(i)|$$

The $r(i)$ of step 2 may be stored in space freed by $A^{(n)}$ and $A^{(n-1)}$. Therefore

$$S_{(\text{charac. poly})} \leq n^2\{|e(A^n)| + |e(A^{n-1})|\} + \sum_{i=1}^{n} |s(i)| \tag{10}$$

2.4 Evaluation by Minors

This is a well-known method and can be found, for example in [5] pp. 440 ex.10.

Here we have n-1 steps, indexed by i from 2 to n. At each step, we compute all of the i by i minors from the first i columns (there are $\binom{n}{i}$ of them), using the $i-1$ by $i-1$ minors from the first $i-1$ columns, computed in the previous step. Gentleman and Johnson consider this method in [8].

Computing Time Analysis

The cost of computing an i x i minor assuming that all required i-1 x i-1 minors are available is i times the time to multiply an element of M by a minor of size i-1. Therefore the total time becomes

$$t(\text{minors}) = \sum_{i=2}^{n} \binom{n}{i} \; i \; T(e(M), \det_{i-1,i-1})$$

$$= n\sum_{i=2}^{n} \frac{(n-1)!}{(i-1)!\,(n-i)!} T(e(M), \det_{i-1,i-1})$$

$$= n\sum_{i=1}^{n-1} \binom{n-1}{i} \; T(e(M), \det_{i,i}) \qquad (11)$$

Storage: At step i we compute the $\binom{n}{i}$ minors of size i x i. Concurrent with this, the $\binom{n}{i-1}$ minors of size i-1 x i-1 have to be retained.

$\therefore$ Storage required at step i $\leq \binom{n}{i}|\det_{i,i}|$

$$+ \binom{n}{i-1}|\det_{i-1,i-1}|$$

$$\therefore \; S_{(\text{minors})} = \max_{2\leq i\leq n} \left\{ \binom{n}{i}|\det_{i,i}| + \binom{n}{i-1}|\det_{i-1,i-1}| \right\} \qquad (12)$$

3. Models and Analysis

In this section we will analyze the times for applying the methods of Section 2 to the problem at hand. But before this can be done it is necessary to make some assumptions about the way in which the polynomials grow as the reductions are carried on. In this case, we are forced to hypothesize a model(s) which hopefully will reasonably reflect reality. In order to provide some motivation for the models we have used, let us look at the 2 extreme possibilities that can occur.

Suppose that every element of M is a polynomial in r variables, degree m-1 in each variable and every term which is possible occurs. Then, each polynomial initially has m^r terms and is said to be dense. Suppose further that as we apply Gaussian elimination the resulting sub-matrices have polynomials which remain dense. In particular at the ith stage, each element has degree $i(m-1)$ in every variable and $(i(m-1)+1)^r$ terms. A generalization of this growth rate for the completely dense case is given in Section 3.1. This model was first used by Horowitz and Sahni in [4] for computing polynomial powers.

Now for the other extreme case suppose that every element of M has t terms and as we apply Gaussian elimination the resulting polynomials have $i!t^i$ terms at the ith stage $1 \leq i \leq n$. This is equivalent to saying that after every iteration every term produced in the multiplication process is distinct from every other term. This gives the maximum rate of growth that could occur without terms overlapping and implies that the polynomials were initially quite sparse. A similar definition for polynomial powers was given by Gentleman in [2]. Section 3.3 contains a generalized model for this completely sparse case.

Then in Section 3.2 a model which is intermediate to the generalized dense and generalized sparse models is presented. Within each of the 3 models we will consider the 4 algorithms: Homomorphism, Gaussian Elimination, Minors and Characteristic Polynomial and compare their relative performances.

However, before we begin let us consider separately the Homomorphism method whose computing time remains the same for all of the models. This curiousity arises from the fact that there is no way for it to take advantage of the sparsity of the polynomials. If for every element of M the degree of $e(M)$ in each variable is $m-1$ then the evaluations and interpolation must be carried on $(m-1)n+1$ times at every level. This is so, even if a polynomial has only t terms where t is much smaller than m^r. Therefore the asymptotic computing time for this algorithm remains $O((mn)^r(n^3+mn^2))$ independent of any of the models. Thus we can already surmise that though the Homomorphism method may work quite well on dense polynomials, other methods will become superior as the matrix's polynomials become more sparse. It is the purpose of the 3 models to include both sparse and dense polynomials which grow at different rates.

3.1 Generalized Dense Model

In this model we assume that a minor of size ixi has $i^k t$ terms where $t = |e(M)|$ and k is a non-negative integer.

We note that for $t = m^r$ and $k = r$ this model corresponds to the case where the $e(M)$ are completely dense polynomials in r-variables and remain completely dense throughout (actually the growth is as $(i(m-1)+1)^r = o(i^r m^r))$. On the other hand, small values of k and t allow for very sparse polynomials which grow at the same rate as the dense polynomials.

3.1.1 Gaussian Elimination

Using equation (7) we get

$$t_{(GE)}^{(GD)} = \sum_{1 \le i \le n-1} (n-i)^2 (2T(A_{kj}^{(i)}, A_{kj}^{(i)}) + T(A_{kj}^{(i+1)}, A_{kj}^{(i-1)}))$$

$$= \sum_{1 \le i \le n-1} (n-i)^2 (2(i^k t)^2 + (i+1)^k t \ (i-1)^k t)$$

$$= 2t^2 \sum_{1 \le i \le n-1} (n-i)^2 \ i^{2k} + t^2 \sum_{1 \le i \le n-1} (n-i)^2 (i^2-1)^k$$

$$= O(t^2 \ n^{2k+3})$$

For the completely dense case $t = m^r$, $k = r$ we get

$t_{(GE)} = O(m^{2r} \ n^{2r+3}) \ge O((mn)^r (n^3+mn^2)) = t_{(HOM)}$. Thus Homomorphism is asymptotically superior to Gaussian elimination on completely dense polynomials for all $r \ge 1$.

From equation (8) we get

$$S_{(GE)}^{(GD)} = \max_{1 \le i \le n-1} \{(n-i)^2 (i+1)^k \ t\}$$

The max occurs for $i = \frac{kn-2}{k+2}$

$$\therefore \ S_{(GE)}^{(GD)} \le \left(\frac{2n+2}{k+2}\right)^2 \left(\frac{kn+k}{k+2}\right)^k t$$

$$= 4k^k \left(\frac{n+1}{k+2}\right)^{k+2} t$$

$$= O((n+1)^{k+2} t/k^2)$$

As special cases we consider dense univariate and dense bivariate polynomials.

Special Case (i) Dense Univariate

If $|e(M)| = t$ then $\deg(e(M)) = t-1$ and $\deg(e(\det_{i,i})) = i(t-1)+1$. Thus,

$$|A^{(i)}| = i(t-1)+1$$

Substituting in equation (7) we get

$$t_{(GE)}^{(\text{dense univariate})} = \frac{n(n-1)}{30}\left((3n^3 + 3n^2 - 7n + 8)(t-1)^2 + 15n(n-1)(t-1)+15(2n-1)\right) \tag{13}$$

Special Case (ii) Dense Bivariate

Let $m^2 = t$, then $\deg(e(M)) = m-1$ and $|A^{(i)}| = (i(m-1)+1)^2$.

Substituting in equation (7) we get

$$\begin{aligned} t_{(GE)}^{(\text{dense bivariate})} &= \sum_{1\le i\le n-1} (n-i)^2[(3i^4-2i^2+1)(m-1)^4 \\ &\qquad + 4i(3i^2-i)(m-1)^3 \\ &\qquad + 2(9i^2-1)(m-1)^2 \\ &\qquad + 12i(m-1)+3] \\ &= \left(\frac{n^7}{35} - \frac{n^5}{15} + \frac{7}{30}n^3 - \frac{n^2}{2} + \frac{32}{105}n\right)(m-1)^4 \\ &+ \left(\frac{n^6}{5} - \frac{n^4}{3} + \frac{2n^2}{15}\right)(m-1)^3 \\ &+ \left(\frac{3}{5}n^5 - \frac{2}{3}n^3 + n^2 - \frac{14}{15}n\right)(m-1)^2 \\ &+ (n^4 - n^2)(m-1) + n^3 - \frac{3}{2}n^2 + \frac{n}{2} \end{aligned} \tag{14}$$

3.1.2 Characteristic Polynomial

Let $|r_i|$, $|s_i|$ and $|e(M^i)|$ = $i^k t$.

Using eq(9) we get:

$$t^{(GD)}_{(\text{charac poly})} = n^3 \sum_{i=1}^{n-1} i^k t^2 + \sum_{i=2}^{n} \sum_{j=1}^{i-1} j^k (i-j)^k t^2 + \sum_{i=2}^{n} i^k t$$

$$= n^3 t^2 \sum_{i=1}^{n-1} i^k + t^2 \sum_{i=2}^{n} \sum_{j=1}^{i-1} [j(i-j)]^k + \sum_{2}^{n} i^k t$$

$$< n^3 t^2 \frac{n^{k+1}}{k+1} + t^2 \sum_{i=2}^{n} (i-1) \left(\frac{i}{2}\right)^{2k} + n^{k+1} t$$

$$< t^2 \frac{n^{k+4}}{k+1} + \frac{1}{2} \cdot \frac{t^2}{4^k} \frac{(n+1)^{2k+2}}{k+1} + n^{k+1} t$$

$$= O(t^2 n^k (4^k n^4 + n^{k+2})/4^k (k+1)).$$

Using (10), for the storage requirements we get

$$S^{(GD)}_{(\text{Charac Poly})} \le n^2 \{n^k t + (n-1)^k t\} + \sum_{1}^{n} i^k t$$

$$\le 2n^{k+2} t + n^{k+1} t$$

$$= O(n^{k+2} t)$$

Special case (i) Dense Univariate:

$|r_i|$, $|s_i|$, and $|e(M^i)|$ = $i(t-1)+1$

$$t^{\text{univariate}}_{\text{charac poly}} = n^3 \sum_{i=1}^{n-1} [i(t-1)+1] t$$

$$+ \sum_{i=2}^{n} \sum_{j=1}^{i-1} (j(t-1)+1)((i-j)(t-1)+1) + \sum_{2}^{n} \left\{ i(t-1)+1 \right\}$$

//

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

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

$$= n(n-1) \ [\frac{n^3(t-1)t}{2} + n^2t + \frac{1}{24}(n+1)(n+2)(t-1)^2$$

$$+ \frac{1}{3}(n+1)(t-1)+1] + \frac{n(n+1)(t-1)}{2} - t + n \qquad (15)$$

Special Case (ii) Dense bivariate

$$|r_i|, |S_i|, |e(M^i)| = (i(m-1)+1)^2$$

$$= i^2(m-1)^2 + 2i(m-1) + 1$$

$$t^{(\text{bivariate})}_{(\text{charac poly})} = t(\text{step1}) + t(\text{step2})$$

$$\text{tstep1} = n^3 \sum_{i=1}^{n-1} (i^2(m-1)^2 + 2i(m-1)+1)t$$

$$= n^3t \ [\frac{n(n-1)(2n-1)}{6}(m-1)^2 + n(n-1)(m-1) + (n-1)]$$

$$= n^3(n-1)t \ [\frac{1}{6}n \ (2n-1)(m-1)^2 + n(m-1) + 1]$$

$$\text{tstep2} = \sum_{i=2}^{n} \sum_{j=1}^{i-1} (j^2(m-1)^2 + 2j(m-1)+1)((i-j)^2(m-1)^2$$

$$+ 2(i-j)(m-1) + 1) + \sum_{i=2}^{n} |r_i|$$

$$= \sum_{i=2}^{n} \sum_{j=1}^{i-1} \Big\{ j^4(m-1)^4 - 2j^3i(m-1)^4$$

$$+ j^2[i^2(m-1)^4 - 2i(m-1)^3 - 2(m-1)^2]$$

$$+ j[2i^2(m-1)^3 + 2i(m-1)^2] + i^2(m-1)^2$$

$$+ 2i(m-1) + 1 \Big\}$$

$$+ \sum_{i=2}^{n} \Big\{ i^2(m-1)^2 + 2i(m-1) + 1 \Big\}$$

$$= (m-1)^4 \left\{\frac{n^2(n+1)(2n^3+4n^2+n-1)}{360} - \frac{n(n+1)}{60}\right\}$$

$$+ (m-1)^3 \left\{\frac{n(n+1)(2n+1)(3n^2+3n-1)}{90} - \frac{n(n+1)(2n+1)}{18}\right\}$$

$$+ (m-1)^2 \left\{\frac{1}{3} n^2(n+1)^2 - \frac{n(n+1)}{6} - 1\right\}$$

$$+ (m-1) \left\{\frac{n(n+1)(2n+1)}{3} - 2\right\}$$

$$+ n(n-1)/2 + (n-1) \tag{16}$$

3.1.3 Minors

$$|\det_{i,i}| = i^k t$$

$$T(e(M)\det_{i,i}) = i^k t^2$$

substituting into eq(11) we get

$$t^{(GD)}_{(minors)} = n \sum_{i=1}^{n-1} \binom{n-1}{i} i^k t^2$$

$$= nt^2 \sum_{i=1}^{n-1} \binom{n-1}{i} i^k$$

$$\leq nt^2 (n-1)^k 2^{n-2}$$

$$= n(n-1)^k 2^{n-2} t^2$$

For the storage requirements we get, from eq (12):

$$S^{(GD)}_{(minors)} \leq \max_{2\leq i\leq n}\left\{\binom{n}{i} i^k t + \binom{n}{i-1}(i-1)^k t\right\}$$

$$< 2tn^k \binom{n}{n/2}$$

Special cases (i) Dense Univariate

$|e(M)| = t$ and therefore $\deg(e(M)) = t-1$ and

$|\det_{i,i}| = i(t-1) + 1.$

Also $T(e(M) \det_{i,i}) = i\, t(t-1) + t$

$$\therefore t_{\substack{\text{(dense univariate)}\\ \text{(minors)}}} = n\, t \sum_{i=1}^{n-1} \binom{n-1}{i} (i(t-1)+1)$$

$$= n\, t[(t-1)\sum_{i=1}^{n-1} \binom{n-1}{i} i + \sum_{i=1}^{n-1} \binom{n-1}{i}]$$

$$= n\, t[(t-1)(n-1)2^{n-2} + 2^{n-1}-1] \tag{17}$$

To see that $\sum_{i=1}^{n-1} \binom{n-1}{i} i = (n-1)2^{n-2}$ observe that

$$\sum_{i=1}^{n-1} \binom{n-1}{i}\, i\, x^i = x \frac{d}{dx} (1+x)^{n-1}$$

$$= (n-1)x(1+x)^{n-2}$$

Set $x = 1$ to get the above result.

This result was first given by Gentleman and Johnson in [8].

Special Case (ii) Dense Bivariate

$|e(M)| = m^2$, $t = m^2$

degree of $e(M) = m-1$

Then $|\det_{i,i}| = (i(m-1)+1)^2$

$= i^2(m-1)^2 + 1 + 2i(m-1)$

$\therefore T(e(M),\det_{i,i}) = i^2(m-1)^2 t + 2\, i\, t(m-1) + t$

$$\begin{Bmatrix}\text{dense bivariate}\\ \text{minors}\end{Bmatrix} = nt \sum_{i=1}^{n-1} \binom{n-1}{i} \{i^2(m-1)^2 + 2i(m-1)+1\}$$

$$= nt\left[\sum_{i=1}^{n-1} \binom{n-1}{i} i^2(m-1)^2 + 2(m-1)\sum\left\{i\binom{n-1}{i}\right\} + (2^{n-1}-1)\right]$$

$$= nt\left[(m-1)^2 \sum_{i=1}^{n-1} \binom{n-1}{i} i^2 + 2(m-1) \sum_{1}^{n-1}\left\{\binom{n-1}{i} i\right\} + 2^{n-1} - 1)\right]$$

$$= nt[(m-1)^2\{(n-1)2^{n-2} + (n-1)(n-2)2^{n-3}\} + 2(m-1)(n-1)2^{n-2} + 2^{n-1}-1]$$

$$= nt(2^{n-1} + 2^{n-2}\{(n-1)[(m-1)^2 + 2(m-1)]\} + 2^{n-3}\{(n-1)(n-2)(m-1)^2\} - 1) \qquad (18)$$

3.2 Average Growth Model

In this model it is assumed that the multiplication of a p term polynomial and a q term polynomial ($p \geq q$) results in a kp ($k \geq 1$) term polynomial, while their addition yields a p term polynomial. Such behavior arises, for instance, when $k = q$, the polynomials P and Q are sparse and the exponents occurring in Q are a subset of those occurring in P. Examples for other values of k may be obtained by varying the degree of sparsity and exponent distribution in P.

The determinant of an $i \times i$ matrix may be viewed as the sum of i! terms, where each term is a product of i polynomials. If each polynomial entry in M has t terms then under our assumptions each of the i! terms in the sum for det(M) has $k^{i-1}t$ terms and det(M) is a polynomial with $k^{i-1}t$ terms (because of the assumption for addition of polynomials). We note that the case when all the entries of M are integers, is a special case of this model ie $k = 1$ and $t = 1$.

3.2.1 Gaussian Elimination

By assumption $|a^{(i)}| = k^{i-1}t$. Using eq(7) we get

$$t_{(GE)}^{(\text{avg growth})} = \sum_{1 \leq i \leq n-1} (n-i)^2 (2(k^{i-1}t)^2 + k^{i-2}t \cdot k^i t)$$

with $k^{-1} = 1$. Then

$$t_{(GE)}^{(avg\ growth)} = \sum_{1\le i\le n-1} (n-i)^2 \ (3k^{2(i-1)}t^2)$$

$$= 3t^2 \sum_{1\le i\le n-1} (n-i)^2 \ k^{2(i-1)}$$

$$< 3t^2 \ k^{2(n-2)} \ n^3$$

$$= O(t^2 \ k^{2(n-2)} \ n^3)$$

For the storage, we get from eq(8):

$$s_{(GE)}^{(avg\ growth)} = \max_{0\le i\le n-1} \left\{ (n-i)^2 \ k^i \ t \right\}$$

which is $\le k^{n-1} t$ for $k^{i-1} t \ge (i+1)^2$

3.2.2 Characteristic Polynomial

Since in the average growth model we assume that the addition of a p term and a q term polynomial ($p\ge q$) results in a p term polynomial while their product has k p terms we obtain:

$$|e(M^i)| = k^{i-1}t$$

and $|s_i| = |e(M^i)| = k^{i-1}t$ as s_i is just the sum of the diagonal entries of M^i. We make the further assumption that $|r_i| = |s_i|$.

From eq(9) we get:

$$t_{\substack{\text{(average growth)}\\ \text{(charac poly)}}} = t_{step1} + t_{step2}$$

$$t_{step1} = n^3 \sum_{i=1}^{n-1} t\cdot k^{i-1} t$$

$$= \begin{cases} n^3 t^2 \dfrac{k^{n-1}-1}{k-1} & \ldots\ k > 1 \\ n^3 (n-1) t^2 & \ldots\ k = 1 \end{cases}$$

$$t_{step2} = \sum_{i=2}^{n} \sum_{j=1}^{i-1} (k^{j-1}t)(k^{i-j-1}t) + \sum_{i=2}^{n} k^{i-1}t$$

$$= t^2 \sum_{i=2}^{n} (i-1)k^{i-2} + \frac{k^n-k}{k-1}\, t$$

$$= t^2 \sum_{i=2}^{n} \frac{d}{dk}(k^{i-1}) + \frac{k^n-k}{k-1}\, t$$

$$= t^2 \frac{d}{dk} \left(\sum_{i=2}^{n} k^{i-1}\right) + \frac{k^n-k}{k-1}\, t$$

$$= \begin{cases} t^2 \dfrac{(n-1)k^n - nk^{n-1}+1}{(k-1)^2} + \dfrac{k^n-k}{k-1} t\ , & k > 1\ ; \\ t^2 \dfrac{n(n-1)}{2} + t(n-1) & k = 1 \end{cases}$$

$$\therefore t_{\text{(charac poly)}}^{\text{(avg growth)}} = \begin{cases} (n^3t^2 \dfrac{k^{n-1}-1}{k-1} + t^2 \dfrac{(n-1)k^n - nk^{n-1}+1}{(k-1)^2} + t\dfrac{k^n-k}{k-1}, & k > 1 \\ n^3(n-1)t^2 + n(n-1)t^2/2 + t(n-1), & k = 1 \end{cases}$$

$$= \begin{cases} O(n^3t^2k^{n-2} + t^2n\, k^{n-2}) = O(n^3t^2k^{n-2}), & k > 1 \\ O(n^4t^2) & k = 1 \end{cases}$$

Substituting in (10) for the storage, we obtain

$$S_{\text{(charac poly)}}^{\text{(avg growth)}} \leq n^2 \{k^{n-1}t + k^{n-2}t\} + \sum_{i=1}^{n} k^{i-1}\, t$$

$$= n^2k^{n-1}t(1+1/k) + \frac{k^n-1}{k-1}\, t \qquad k > 1$$

$$= O(n^2\, k^{n-1}\, t)$$

3.2.3 Minors

$$|\det_{i,i}| = k^{i-1}\, t$$

$$\therefore T(e_{(M)} \det_{i,i}) = k^{i\ 1}t^2$$

substituting into eq (11) we get

$$t_{\text{(minors)}}^{\text{(avg growth)}} = n \sum_{i=1}^{n-1} \binom{n-1}{i} k^{i-1}t^2$$

$$= \frac{nt^2}{k} \left(\sum_{i=0}^{n-1} \binom{n-1}{i} k^i - 1\right)$$

$$= \frac{nt^2}{k} [(1+k)^{n-1} - 1]$$

eq(12) yields:

$$S^{(\text{avg growth})}_{(\text{minors})} = \max_{2 \le i \le n} \left\{ \binom{n}{i} k^{i-1} t + \binom{n}{i-1} k^{i-2} t \right\}$$

$$\le 2 \frac{t}{k} (1+k)^n$$

3.3 Generalized Sparse Model

The assumptions for this model are similar to those for the Average Growth model. As in that model, here too, it is assumed that the product of a p term and a q term ($p \ge q$) polynomial is a kp ($k \ge 1$) term polynomial. However, we now assume that the exponents present in Q are not present in P so that their addition results in a $p + q$ term polynomial. For $k = q$ this means that P,Q are totally sparse and have no exponents in common.

The consequences of this change in assumption for addition is that the determinant of an $i \times i$ matrix now has a number of terms equal to $i! \cdot$(# of terms in the product of i polynomials) $= i!k^{i-1}t$ terms. For $k = t$, this represents the maximum possible growth rate that can occur. It is readily seen that the nxn matrix with $A_{ij} = r_{ij}$, where r_{ij} represents a variable and $r_{ij} = r_{\ell m} \iff i = \ell$ and $j = m$ falls under this model with $k = t = 1$. Examples for other values of k,t can easily be constructed.

3.3.1 Gaussian Elimination

By eq(7) we obtain:

$$t^{(\text{gen sparse})}_{(GE)} = \sum_{1\le i\le n-1} (n-i)^2(2(i!k^{i-1}t)^2$$

$$+ ((i-1)!k^{i-2}t)((i+1)!k^{i}t))$$

$$= 2t^2 \sum_{1\le i\le n-1} (n-i)^2(i!)^2k^{2(i-1)}$$

$$+ \; t^2 \sum_{1\le i\le n-1} (n-i)^2(i-1)!(i+1)!\,k^{2(i-1)}$$

$$\le 3t^2 \sum_{1\le i\le n-1} (i+1)!^2(n-i)^2k^{2(i-1)}$$

$$= O(t^2n!^2k^{2(n-2)}n^3)$$

and eq(8) gives, for the storage,

$$s^{(\text{gen sparse})}_{(GE)} = \max_{1\le i\le n-1} \left\{(n-i)^2(i+1)!k^it\right\}$$

which is $\le n!k^{n-1}t$ for $n, k \ge 2$

For the completely sparse case we get

$$t^{(\text{sparse})}_{(GE)} = O(n!^2n^3t^{2n-2}) \text{ and } s^{(\text{sparse})}_{(GE)} = O(n!t^n).$$

3.3.2 Characteristic Polynomial

Since in this model we assume that multiplication of a t term polynomial by a t term polynomial results in a kt term polynomial and that their addition yields a 2t term polynomial we get $|e(M^i)| = (nk)^{i-1}t$.

Since $s_i = \sum_{j=1}^{n} a^i_{jj}$ where a^i_{jj} is a diagonal entry of M^i, we obtain

$$|s_i| = n\ |e(M^i)| = n^i\ k^{i-1}t\ .$$

In step2 it is assumed that no new terms are generated, so $|r_i| = |s_i| = n^i k^{i-1}t$

Substituting for $|e(M^i)|, |r_i|, |s_i|$ in eq(9) we get:

$$t^{(sparse)}_{(charac\ poly)} = t_{step1} + t_{step2}$$

$$t_{step1} = n^3 \sum_{i=1}^{n-1} t \cdot (nk)^{i-1}t$$

$$= n^3t^2 \sum_{i=1}^{n-1} (nk)^{i-1}$$

$$= n^3t^2 \ \frac{(nk)^{n-1}-1}{nk - 1}$$

$$t_{step2} = \sum_{i=2}^{n} \sum_{j=1}^{i-1} n\ (nk)^{j-1}t \cdot n(nk)^{i-j-1}t + n\sum_{i=2}^{n}(nk)^{i-1}t$$

$$= n^2t^2 \sum_{i=2}^{n} (nk)^{i-2}(i-1) + nt\ \frac{(nk)^n - nk}{nk-1}$$

$$= n^2t^2\ \frac{(n-1)(nk)^n - n(nk)^{n-1}+1}{(nk-1)^2} + nt\ \frac{(nk)^n-nk}{nk-1}\ ,\ nk > 1$$

$$\therefore t^{(gen\ sparse)}_{(charac\ poly)} = n^3t^2\ \frac{(nk)^{n-1}-1}{nk-1} + n^2t^2\ \frac{(n-1)(nk)^n - n(nk)^{n-1}+1}{(nk-1)^2}$$

$$+ \frac{nt(nk)^n - nk}{nk-1}$$

$$= O(n^{n+1}k^{n-2}t^2)$$

The storage required is

$$s^{(gen\ sparse)}_{(charac\ poly)} \leq n^2\{(nk)^{n-1}t + (nk)^{n-2}t\} + \sum_{i=1}^{n} n^i k^{i-1}t$$

$$= O(n^{n+1}k^{n-1}t)$$

and for the completely sparse case we have

$$t^{(sparse)}_{(charac\ poly)} = O(n^{n+1}t^n)$$

$$s^{(sparse)}_{(charac\ poly)} = O(n^{n+1}t^n)$$

3.3.3 Minors

$$|\det_{i,i}| = i!k^{i-1}t$$

$$T(eM, \det_{i,i}) = i!k^{i-1}t^2$$

$$t^{(\text{gen sparse})}_{(\text{minors})} = n \sum_{i-1}^{n-1} \binom{n-1}{i} i!k^{i-1} t^2$$

$$= nt^2 \sum_{i=1}^{n-1} \frac{(n-1)!}{(n-i-1)!} k^{i-1}$$

$$= n!t^2k^{n-2} \sum_{i=1}^{n-1} \frac{k^{-(n-i-1)}}{(n-i-1)!}$$

$$\leq n!\ t^2k^{n-2}\ e^{1/k}$$

For the completely sparse case (k=t) we get

$$t^{(\text{gen sparse})}_{(\text{minors})} \leq n!t^n e^{1/t}$$

Equation (12) yields, for the storage required,

$$S^{(\text{gen sparse})}_{(\text{minors})} = \max_{2\leq i\leq n} \left\{ \binom{n}{i} i!k^{i-1}t + \binom{n}{i-1} (i-1)!k^{i-2}t \right\}$$

$$= \max_{2\leq i\leq n} \left\{ \frac{n!}{(n-i)!} k^{i-1}t + \frac{n!}{(n-i-1)!} k^{i-2}t \right\}$$

$$= n!\ k^{n-1}t\ (1 + 1/k)$$

For the completely sparse case, this gives

$$S^{(\text{sparse})}_{(\text{minors})} = n!t^n(1 + 1/t) \leq 1{\cdot}5n!t^n\ , \quad t \geq 2$$

4. Conclusions

The table below summarizes the asymptotic computing times and storage requirements for each of the 3 methods under each of the 3 models. The methods are listed in increasing order of time

	Method	Time	Space
Generalized Dense:			
	Charac Poly	$t^2(n^{k+4}+n^{2k+2}/4^k)$	$t\ n^{k+2}$
	Gauss. Elim.	$t^2\ n^{2k+3}$	$t(n+1)^{k+2}/k^2$
	Minors	$t^2\ n^{k+1}2^{n-2}$	$t\ n^k\binom{n}{n/2}$
Average Growth:			
	Minors	$t^2\ n\ \ k^{n-2}$	$2t(k+1)^n/k$
	Charac Poly	$t^2\ n^3\ k^{n-2}$ $(k>1)$	$t\ k^{n-1}n^2$
	Gauss. Elim.	$t^2\ n^3\ k^{2(n-2)}$	$t\ k^{n-1}$
Generalized Sparse:			
	Minors	$t^2\ n!\ k^{n-2}e^{1/k}$	$t\ n!\ k^{n-1}(1+1/k)$
	Charac Poly	$t^2\ n^{n+1}\ k^{n-2}$	$t\ n^{n+1}k^{n-1}$
	Gauss. Elim.	$t^2\ n!^2\ k^{2(n-2)}n^3$	$t\ n!\ k^{n-1}$

Thus we can conclude that asymptotically the following is true: for both the average growth and generalized sparse models Minors is the best method for all values of k. As for storage Minors is either better than or no worse than any of the other methods by a constant factor. In fact for the completely sparse polynomial case as considered by Gentlemen and Johnson in [8], where k = t, Minors is optimal to within a constant factor. For the generalized dense model the Characteristic Polynomial method is superior for all $k > 1$, and equivalent to Gaussian Elimination for $k = 1$. In the completely dense case $k = r$ however, Homomorphism is asymptotically better than Characteristic Polynomial.

The fact that Minors is asymptotically the best method on certain models is interesting, but one must note that because of the exponential growth of this method only relatively small problems could ever be successfully computed. Nevertheless the desirableness of using Minors over the other, more commonly used methods has been shown.

Now can it be the case that Minors should ever be used when the entries are completely dense? Gentleman and Johnson [8] have shown that for univariate dense polynomials Minors is better than G.E. for $n \leq 7$. If M has completely dense univariate polynomial entries then, by equations (5), (13), (15) and (17) which give the exact computing times for Homomorphism, Exact Division, Characteristic Polynomial, and Minors. we find that for all $t > 1$ Minors is better on matrices whose size is $n \leq 5$. So in the completely dense univariate case it is better to use Minors over Homomorphism as long as the size of the matrix is ≤ 5 no matter how many terms are in the polynomials.

Similarly examining equations (6), (14), (16) and (18) for the completely dense bivariate case Minors is better to use as long as one of the following conditions hold

a) $n = 2$ & $2 \leq m \leq 28$

b) $n = 3$ & $2 \leq m \leq 11$

c) $n = 4$ & $2 \leq m \leq 4$

For $n > 4$ Homomorphism requires fewer operations than Minors. We further note that the Gaussian Elimination method will be worse than either Minors or Characteristic Polynomial for all $m,n > 1$. Thus Gaussian Elimination is a poor method to use in every case.

For the case where the matrix entries are dense polynomials except for a few that have some high order terms, Homomorphism is not suitable. For such matrices the following may be obtained.

a) In the case of univariate polynomials, Minors is best for $n \leq 7$. For larger n Exact Division is best.

b) For polynomials in more than one variable the choice is between Minors and Characteristic Polynomial, the former being better for small values of n. The value of n at which the two methods break even, increases as the number of variables increase. This may be seen by taking the ratio of their equations listed in this Section which gives

$$\frac{\text{Charac. Poly}}{\text{Minors}} = \frac{n^{k+1}}{4^k 2^{n-2}} + \frac{n^3}{2^{n-2}}$$

So for example in the bivariate case Minors is better for $n < 14$ and as the variables grow the cut-off becomes larger. Therefore, even in this relatively dense model Minors is the best method to use for most practical problems.

Unfortunately no method is a clear winner. Moreover our simplifying assumption that every element has the same structure does not always occur. Thus it is necessary to a priori choose which method to use. The criteria given here should allow one to reasonably approximate the best method for his problem. It is clear that different methods will vary significantly on a mix of problems. Moreover we note that even though we have restricted our analysis to determinant calculation, many of these results also hold true for other computations related to matrices. This is so, since the time for triangularization often bounds many of these tasks and this aspect is essential to determinant calculation.

In conclusion we have shown that for matrices with relatively sparse polynomials the use of Minors for determinant calculation is the most efficient method. In addition, Minors can be successfully applied to matrices with dense polynomials in certain small, but important cases. Finally the Homomorphism method for the completely dense case or else computing the characteristic polynomial should be used for matrices with relatively dense entries and sufficiently large matrix size n.

REFERENCES

[1] Bareiss, G.H., "Sylvester's Identity and Mutistep Integer-Preserving Gaussian Elimination", Math of Comp, Vol. 22, No. 103, July 1968, pp. 565-578.

[2] Gentleman, W.M., "Optimal Multiplication Chains for Computing a Power of a Symbolic Polynomial" SIGSAM Bulletin No. 18, April 1971, and Mathematics of Computation, Oct. 72.

[3] Horowitz, E., "The Efficient Calculation of Powers of a Polynomial", Journal of Computer and System Science, (to appear)

[4] Horowitz, E. and S. Sahni, "On the Computation of Powers of a Class of Polynomials", TR 72-143, Cornell University, August 1972.

[5] Knuth, D.E., "The Art of Computer Programming, Vol. 2 "Seminumerial Algorithms", Addison Wesley, 1969.

[6] Lipson, J.D., "Symbolic Methods for the Computer Solution of Linear Equations with Applications to Flow Graphs", Proceedings of the 1968 Summer Institute on Symbolic Mathematical Computation, IBM, Boston, June 1969, pp. 233-303.

[7] McClellan, M.T., "The Exact Solution of Systems of Linear Equations With Polynomial Coefficients", TR #136, Sept. 71, Computer Science Department, University of Wisconsin (to appear in J.ACM).

[8] Gentleman, W.M. and S.C. Johnson, "Analysis of Algorithms, A Case Study: Determinants of Polynomials", Bell Telephone Laboratories Technical Memorandum, Case 39199-11, October, 1972.