Found Comput Math (2016) 16:329–342
DOI 10.1007/s10208-014-9243-7

FOUNDATIONS OF COMPUTATIONAL MATHEMATICS
The Journal of the Society for the Foundations of Computational Mathematics

Computing the Permanent of (Some) Complex Matrices

Alexander Barvinok

Received: 24 June 2014 / Revised: 22 November 2014 / Accepted: 30 November 2014 /
Published online: 6 January 2015

Abstract We present a deterministic algorithm, which, for any given $0 < \epsilon < 1$ and an $n \times n$ real or complex matrix $A = \left(a_{ij}\right)$ such that $\left|a_{ij} - 1\right| \leq 0.19$ for all i, j computes the permanent of A within relative error ϵ in $n^{O(\ln n - \ln \epsilon)}$ time. The method can be extended to computing hafnians and multidimensional permanents.

Keywords Permanent · Hafnian · Algorithm

Mathematics Subject Classification 15A15 · 68C25 · 68W25

1 Introduction and Main Results

The *permanent* of an $n \times n$ matrix $A = \left(a_{ij}\right)$ is defined as

$$\text{per } A = \sum_{\sigma \in S_n} \prod_{i=1}^{n} a_{i\sigma(i)},$$

where S_n is the symmetric group of permutations of the set $\{1, \ldots, n\}$. The problem of efficient computation of the permanent has attracted a lot of attention. It is $\#P$-hard already for 0–1 matrices [18], but a fully polynomial randomized approximation

Communicated by Stephen Cook.

This research was partially supported by NSF Grants DMS 0856640 and DMS 1361541.

A. Barvinok (✉)
Department of Mathematics, University of Michigan, Ann Arbor, MI 48109-1043, USA
e-mail: barvinok@umich.edu

scheme, based on the Markov Chain Monte Carlo approach, is constructed for all non-negative matrices [14]. A deterministic polynomial time algorithm based on matrix scaling for computing the permanent of non-negative matrices within a factor of e^n is constructed in [15], and the bound was recently improved to 2^n in [13]. An approach based on the idea of "correlation decay" from statistical physics results in a deterministic polynomial time algorithm approximating per A within a factor of $(1+\epsilon)^n$ for any $\epsilon > 0$, fixed in advance, if A is the adjacency matrix of a constant degree expander [11].

There is also interest in computing permanents of *complex* matrices [1]. The well-known Ryser's algorithm (see, for example, [16, Chapter 7]) computes the permanent of a matrix A over any field in $O(n2^n)$ time. A randomized approximation algorithm of Fürer [10] computes the permanent of a complex matrix within a (properly defined) relative error ϵ in $O\left(3^{n/2}\epsilon^{-2}\right)$ time. The randomized algorithm of Gurvits [12], see also [1] for an exposition, computes the permanent of a complex matrix A in polynomial in n and $1/\epsilon$ time within an additive error of $\epsilon\|A\|^n$, where $\|A\|$ is the operator norm of A.

In this paper, we present a new approach to computing permanents of real or complex matrices A and show that if $\left|a_{ij}-1\right| \leq \gamma$ for some absolute constant $\gamma > 0$ (we can choose $\gamma = 0.19$) and all i and j, then, for any $\epsilon > 0$ the value of per A can be computed within relative error ϵ in $n^{O(\ln n - \ln \epsilon)}$ time (we say that $\alpha \in \mathbb{C}$ approximates per A within relative error $0 < \epsilon < 1$ if per $A = \alpha(1+\rho)$ where $|\rho| < \epsilon$). We also discuss how the method can be extended to computing hafnians of symmetric matrices and multidimensional permanents of tensors.

1.1 The Idea of the Algorithm

Let J denote the $n \times n$ matrix filled with 1s. Given an $n \times n$ complex matrix A, we consider (a branch of) the univariate function

$$f(z) = \ln \operatorname{per}\left(J + z(A-J)\right). \tag{1.1}$$

Clearly,

$$f(0) = \ln \operatorname{per} J = \ln n! \quad \text{and} \quad f(1) = \ln \operatorname{per} A.$$

Hence, our goal is to approximate $f(1)$ and we do it by using the Taylor polynomial expansion of f at $z=0$:

$$f(1) \approx f(0) + \sum_{k=1}^{m} \frac{1}{k!}\frac{d^k}{dz^k} f(z)\Big|_{z=0}. \tag{1.2}$$

It turns out that the right hand side of (1.2) can be computed in $n^{O(m)}$ time. We present the algorithm in Sect. 2. The quality of the approximation (1.2) depends on the location of complex zeros of the permanent.

Lemma 1.1 *Suppose that there exists a real $\beta > 1$ such that*

$$\operatorname{per}\left(J + z(A - J)\right) \neq 0 \quad \textit{for all} \quad z \in \mathbb{C} \quad \textit{satisifying} \quad |z| \leq \beta.$$

Then for all $z \in \mathbb{C}$ with $|z| \leq 1$ the value of

$$f(z) = \ln \operatorname{per}\left(J + z(A - J)\right)$$

is well defined by the choice of the branch of the logarithm for which $f(0)$ is a real number, and the right hand side of (1.2) approximates $f(1)$ within an additive error of

$$\frac{n}{(m+1)\beta^m(\beta - 1)}.$$

In particular, for a fixed $\beta > 1$, to ensure an additive error of $0 < \epsilon < 1$, we can choose $m = O(\ln n - \ln \epsilon)$, which results in the algorithm for approximating per A within relative error ϵ in $n^{O(\ln n - \ln \epsilon)}$ time. We prove Lemma 1.1 in Sect. 2.

Thus, we have to identify a class of matrices A for which the number $\beta > 1$ of Lemma 1.1 exists. We prove the following result.

Theorem 1.2 *There is an absolute constant $\delta > 0$ (we can choose $\delta = 0.195$) such that if $Z = \left(z_{ij}\right)$ is a complex $n \times n$ matrix satisfying*

$$\left|z_{ij} - 1\right| \leq \delta \quad \textit{for all} \quad i, j$$

then

$$\operatorname{per} Z \neq 0.$$

We prove Theorem 1.2 in Sect. 3.

For any matrix $A = \left(a_{ij}\right)$ satisfying

$$\left|a_{ij} - 1\right| \leq 0.19 \quad \text{for all} \quad i, j,$$

we can choose $\beta = 195/190$ in Lemma 1.1 and thus obtain an approximation algorithm for computing per A.

The sharp value of the constant δ in Theorem 1.2 is not known to the author. A simple example of a 2×2 matrix

$$A = \begin{pmatrix} \frac{1+i}{2} & \frac{1-i}{2} \\ \frac{1-i}{2} & \frac{1+i}{2} \end{pmatrix}$$

for which per $A = 0$ shows that in Theorem 1.2 we must have

$$\delta < \frac{\sqrt{2}}{2} \approx 0.71.$$

What is also not clear is whether the constant δ can *improve* as the size of the matrix grows.

1.2 Question

Is it true that for any $0 < \epsilon < 1$ there is a positive integer $N(\epsilon)$ such that if $Z = \left(z_{ij}\right)$ is a complex $n \times n$ matrix with $n > N(\epsilon)$ and

$$\left|z_{ij} - 1\right| \leq 1 - \epsilon \quad \text{for all} \quad i, j$$

then per $Z \neq 0$?

In geometric terms, Theorem 1.2 asserts that the ℓ^{∞}-distance from the matrix J of all 1s to the complex hypersurface per $Z = 0$ in $\mathbb{C}^{n \times n}$ is bounded from below by a positive absolute constant, independent on n. The ℓ^2-distance from a point to a complex algebraic variety has been studied recently in [8].

We note that for any $0 < \epsilon < 1$, fixed in advance, a deterministic polynomial time algorithm based on scaling approximates the permanent of a given $n \times n$ real matrix $A = \left(a_{ij}\right)$ satisfying

$$\epsilon \leq a_{ij} \leq 1 \quad \text{for all} \quad i, j$$

within a multiplicative factor of $n^{\kappa(\epsilon)}$ for some $\kappa(\epsilon) > 0$ [6].

1.3 Ramifications

In Sect. 4, we discuss how our approach can be used for computing hafnians of symmetric matrices and multidimensional permanents of tensors. The same approach can be used for computing partition functions associated with cliques in graphs [5] and graph homomorphisms [7], although the most general framework under which our approach works is still not quite clear. In each case, the main problem is to come up with a version of Theorem 1.2 bounding the complex roots of the partition function away from the vector of all 1s. Isolating zeros of complex extensions of real partition functions is a problem studied in statistical physics and also in connection to combinatorics, see, for example, [17].

An anonymous referee asked what “basepoint” matrices other than J can be used in the algorithm. As follows from Sect. 2, such a base matrix (call it X) should have the property that the permanents of its square submatrices are efficiently computable. One candidate for such an X would be a matrix of a small (fixed in advance) rank, cf. [3]. On the other hand, the way we prove Theorem 1.2 in Sect. 3 would require that the arguments of entries of X (as complex numbers) are close to each other. The current choice of J appears to be the easiest to handle and produces the best estimates.

2 The Algorithm

2.1 The Algorithm for Approximating the Permanent

Given an $n \times n$ complex matrix $A = \left(a_{ij}\right)$, we present an algorithm which computes the right hand side of the approximation (1.2) for the function $f(z)$ defined by (1.1).

Let

$$g(z) = \operatorname{per}\big(J + z(A - J)\big), \tag{2.1}$$

so $f(z) = \ln g(z)$. Hence

$$f'(z) = \frac{g'(z)}{g(z)} \quad \text{and} \quad g'(z) = g(z) f'(z).$$

Therefore, for $k \geq 1$ we have

$$\frac{d^k}{dz^k} g(z)\Big|_{z=0} = \sum_{j=0}^{k-1} \binom{k-1}{j} \left(\frac{d^j}{dz^j} g(z)\Big|_{z=0}\right) \left(\frac{d^{k-j}}{dz^{k-j}} f(z)\Big|_{z=0}\right) \tag{2.2}$$

(we agree that the 0th derivative of g is g).

We note that $g(0) = n!$. If we compute the values of

$$\frac{d^k}{dz^k} g(z)\Big|_{z=0} \quad \text{for} \quad k = 1, \ldots, m, \tag{2.3}$$

then the formulas (2.2) for $k = 1, \ldots, m$ provide a non-degenerate triangular system of linear equations that allows us to compute

$$\frac{d^k}{dz^k} f(z)\Big|_{z=0} \quad \text{for} \quad k = 1, \ldots, m.$$

Hence our goal is to compute the values (2.3).

We have

$$\begin{aligned}
\frac{d^k}{dz^k} g(z)\Big|_{z=0} =& \frac{d^k}{dz^k} \sum_{\sigma \in S_n} \prod_{i=1}^{n} \big(1 + z\left(a_{i\sigma(i)} - 1\right)\big)\Big|_{z=0} \\
=& \sum_{\sigma \in S_n} \sum_{1 \leq i_1, \ldots, i_k \leq n} \left(a_{i_1\sigma(i_1)} - 1\right) \cdots \left(a_{i_k\sigma(i_k)} - 1\right) \\
=& (n-k)! \sum_{\substack{1 \leq i_1, \ldots, i_k \leq n \\ 1 \leq j_1, \ldots, j_k \leq n}} \left(a_{i_1 j_1} - 1\right) \cdots \left(a_{i_k j_k} - 1\right),
\end{aligned}$$

where the last sum is over all pairs of ordered k-subsets $(i_1, \ldots, i_k)$ and $(j_1, \ldots, j_k)$ of the set $\{1, \ldots, n\}$. Since the last sum contains $\big(n!/(n-k)!\big)^2 = n^{O(k)}$ terms, the complexity of the algorithm is indeed $n^{O(m)}$.

As an anonymous referee pointed out, the kth number in (2.3) is $k!(n-k)!$ times the sum of permanents of all $k \times k$ submatrices of $A - J$ and hence one can apply the algorithm of Friedland and Gurvits [9] to speed up the computation of (2.3). If one uses the algorithm of Friedland and Gurvits [9], the complexity of computing (2.3) becomes $\binom{n}{m}n^{O(1)}$ provided $m \ll n$, which is still $n^{O(m)}$.

In the bit model of computation (assuming that the input matrix A is complex rational), the complexity of the algorithm is $\mathcal{L}^{O(m)}$, where $\mathcal{L}$ is the length of the input. Indeed, the complexity of computing (2.3) is obviously bounded by $\mathcal{L}^{O(m)}$ and the system (2.2) of linear equations is well conditioned, since the matrix of the system is lower triangular with diagonal entries equal to $g(0) = n!$.

Proof of Lemma 1.1 The function $g(z)$ defined by (2.1) is a polynomial in z of degree $d \leq n$ with $g(0) = n! \neq 0$, so we factor

$$g(z) = g(0)\prod_{i=1}^{d}\left(1 - \frac{z}{\alpha_i}\right),$$

$\alpha_1, \ldots, \alpha_d$ are the roots of $g(z)$. By the condition of Lemma 1.1, we have

$$|\alpha_i| \ \geq \ \beta > 1 \quad \text{for} \quad i = 1, \ldots, d.$$

Therefore,

$$f(z) = \ln g(z) = \ln g(0) + \sum_{i=1}^{d} \ln\left(1 - \frac{z}{\alpha_i}\right) \quad \text{for} \quad |z| \leq 1, \tag{2.4}$$

where we choose the branch of $\ln g(z)$ that is real at $z = 0$. Using the standard Taylor expansion, we obtain

$$\ln\left(1 - \frac{1}{\alpha_i}\right) = -\sum_{k=1}^{m} \frac{1}{k}\left(\frac{1}{\alpha_i}\right)^k + \zeta_m,$$

where

$$|\zeta_m| = \left|\sum_{k=m+1}^{+\infty} \frac{1}{k}\left(\frac{1}{\alpha_i}\right)^k\right| \ \leq \ \frac{1}{(m+1)\beta^m(\beta-1)}.$$

Therefore, from (2.4) we obtain

$$f(1) = f(0) + \sum_{k=1}^{m}\left(-\frac{1}{k}\sum_{i=1}^{d}\left(\frac{1}{\alpha_i}\right)^k\right) + \eta_m,$$

where

$$|\eta_m| \ \leq \ \frac{n}{(m+1)\beta^m(\beta-1)}.$$

It remains to notice that

$$-\frac{1}{k}\sum_{i=1}^{d}\left(\frac{1}{\alpha_i}\right)^k = \frac{1}{k!}\frac{d^k}{dz^k}f(z)\Big|_{z=0}.$$

□

As an anonymous referee pointed out, it follows from the proof of Lemma 1.1 that choosing $m = O(\ln n - \ln\epsilon)$ we achieve an additive error of $\epsilon/(\ln n - \ln\epsilon)$ in (1.2), which is slightly better than just ϵ claimed in Sect. 1.1.

3 Proof of Theorem 1.2

Let us denote by $\mathcal{U}^{n\times n}(\delta) \subset \mathbb{C}^{n\times n}$ the closed polydisc

$$\mathcal{U}^{n\times n}(\delta) = \Big\{Z = \big(z_{ij}\big) : \quad \big|z_{ij} - 1\big| \ \leq\ \delta \quad \text{for all} \quad i, j\Big\}.$$

Thus Theorem 1.2 asserts that per $Z \neq 0$ for $Z \in \mathcal{U}^{n\times n}(\delta)$ and $\delta = 0.195$.

First, we establish a simple geometric lemma.

Lemma 3.1 *Let $u_1, \ldots, u_n \in \mathbb{R}^d$ be nonzero vectors such that for some $0 \leq \alpha < \pi/2$ the angle between any two vectors u_i and u_j does not exceed α. Let $u = u_1 + \ldots + u_n$. Then*

$$\|u\| \ \geq\ \sqrt{\cos\alpha}\sum_{i=1}^{n}\|u_i\|.$$

Proof We have

$$\|u\|^2 = \sum_{1\leq i,j\leq n}\langle u_i, u_j\rangle \ \geq \sum_{1\leq i,j\leq n}\|u_i\|\|u_j\|\cos\alpha = (\cos\alpha)\left(\sum_{i=1}^{n}\|u_i\|\right)^2,$$

and the proof follows. □

We prove Theorem 1.2 by induction on n, using Lemma 3.1 and the following two lemmas.

Lemma 3.2 *For an $n\times n$ matrix $Z = \big(z_{ij}\big)$ and $j = 1, \ldots, n$, let Z_j be the $(n-1)\times(n-1)$ matrix obtained from Z by crossing out the first row and the jth column of Z.*

Suppose that for some $\delta > 0$ and for some $0 < \tau < 1$, for any $Z \in \mathcal{U}^{n\times n}(\delta)$ we have per $Z \neq 0$ *and*

$$|\text{per } Z| \geq \ \tau\sum_{j=1}^{n}\big|z_{1j}\big|\,\big|\text{per } Z_j\big|.$$

Let $A, B \subset \mathcal{U}^{n\times n}(\delta)$ be any two $n \times n$ matrices that differ in one column (or in one row) only. Then the angle between two complex numbers per A *and* per B*, interpreted as vectors in $\mathbb{R}^2 = \mathbb{C}$ does not exceed*

$$\theta = \frac{2\delta}{(1-\delta)\tau}.$$

Proof Since $\operatorname{per} Z \neq 0$ for all $Z \in \mathcal{U}^{n\times n}(\delta)$, we may consider a branch of $\ln \operatorname{per} Z$ defined for $Z \in \mathcal{U}^{n\times n}(\delta)$.

Using the expansion

$$\operatorname{per} Z = \sum_{j=1}^{n} z_{1j} \operatorname{per} Z_j, \tag{3.1}$$

we conclude that

$$\frac{\partial}{\partial z_{1j}} \ln \operatorname{per} Z = \frac{\operatorname{per} Z_j}{\operatorname{per} Z} \quad \text{for} \quad j = 1, \ldots, n.$$

Therefore, since $\left|z_{ij}\right| \geq 1-\delta$ for $j = 1, \ldots, n$, we conclude that for any $Z \in \mathcal{U}^{n\times n}(\delta)$, we have

$$\sum_{j=1}^{n} \left| \frac{\partial}{\partial z_{1j}} \ln \operatorname{per} Z \right| \leq \frac{1}{(1-\delta)\tau}. \tag{3.2}$$

Since the permanent is invariant under permutations of rows, permutations of columns and taking the transpose of the matrix, without loss of generality we may assume that the matrix $B \in \mathcal{U}^{n\times n}(\delta)$ is obtained from $A \in \mathcal{U}^{n\times n}(\delta)$ by replacing the entries a_{1j} by numbers b_{1j} such that

$$\left|b_{1j} - 1\right| \leq \delta \quad \text{for} \quad j = 1, \ldots, n.$$

Then

$$|\ln \operatorname{per} A - \ln \operatorname{per} B| \leq \left(\sup_{Z \in \mathcal{U}^{n\times n}(\delta)} \sum_{j=1}^{n} \left| \frac{\partial}{\partial z_{1j}} \ln \operatorname{per} Z \right| \right) \left(\max_{j=1,\ldots,n} \left|a_{1j} - b_{1j}\right| \right).$$

Since

$$\left|b_{1j} - a_{1j}\right| \leq 2\delta \quad \text{for all} \quad j = 1, \ldots, n,$$

the proof follows from (3.2). □

Lemma 3.3 *Suppose that for some*

$$0 \leq \theta < \frac{\pi}{2} - 2 \arcsin \delta$$

and for any two matrices $A, B \in \mathcal{U}^{n \times n}(\delta)$ which differ in one row (or in one column), the angle between two complex numbers per A *and* per B*, interpreted as vectors in $\mathbb{R}^2 = \mathbb{C}$ does not exceed θ. Then for any matrix $Z \in \mathcal{U}^{(n+1)\times(n+1)}(\delta)$, we have*

$$|\mathrm{per}\ Z| \geq \tau \sum_{j=1}^{n+1} \left|z_{1j}\right| \left|\mathrm{per}\ Z_j\right|$$

with

$$\tau = \sqrt{\cos\left(\theta + 2 \arcsin \delta\right)},$$

where Z_j is the $n \times n$ matrix obtained from Z by crossing out the first row and the jth column.

Proof We use the first row expansion (3.1) and observe that any two matrices Z_j and Z_k, can be obtained from one from another by replacing one column and a permutation of columns. Therefore, the angle between any two complex numbers per Z_j and per Z_k does not exceed θ. Since

$$-\arcsin \delta \leq \arg z_{1j} \leq \arcsin \delta \quad \text{for} \quad j = 1, \ldots, n,$$

the angle between any two numbers z_{1j} per Z_j and z_{1k} per Z_k does not exceed $\theta + 2 \arcsin \delta$. The proof follows by Lemma 3.1. □

Proof of Theorem 1.2 One can see that for a sufficiently small $\delta > 0$, the equation

$$\theta = \frac{2\delta}{(1-\delta)\sqrt{\cos(\theta + 2\arcsin\delta)}} \tag{3.3}$$

has a solution $0 < \theta < \pi/2$. Numerical computations show that we can choose $\delta = 0.195$ and

$$\theta \approx 0.7611025121.$$

Let

$$\tau = \sqrt{\cos(\theta + 2\arcsin\delta)} \approx 0.6365398112.$$

We proceed by induction on n. More precisely, we prove the following three statements (3.4)–(3.6) by induction on n:

(3.4) For every $Z \in \mathcal{U}^{n \times n}(\delta)$, we have per $Z \neq 0$;

(3.5) Suppose $A, B \in \mathcal{U}^{n \times n}(\delta)$ are two matrices which differ by one row (or one column). Then, the angle between two complex numbers per A and per B, interpreted as vectors in $\mathbb{R}^2 = \mathbb{C}$, does not exceed θ;

(3.6) For a matrix $Z \in \mathcal{U}^{n \times n}(\delta)$, $Z = \left(z_{ij}\right)$, let Z_j be the $(n-1) \times (n-1)$ matrix obtained by crossing out the first row and the jth column. Then

$$|\text{per } Z| \geq \tau \sum_{j=1}^{n} \left|z_{1j}\right| \left|\text{per } Z_j\right|.$$

For $n = 1$, the statement (3.4) is obviously true. Moreover, the angle between any two numbers $a, b \in \mathcal{U}^{1 \times 1}(\delta)$ does not exceed

$$2 \arcsin \delta \approx 0.3925149004 < \theta,$$

so (3.5) holds as well. The statement (3.6) is vacuous.

Lemma 3.3 implies that if the statement (3.5) holds for $n \times n$ matrices then the statement (3.6) holds for $(n+1) \times (n+1)$ matrices.

The statement (3.6) for $(n+1) \times (n+1)$ matrices together with the statement (3.4) for $n \times n$ matrices implies the statement (3.4) for $(n+1) \times (n+1)$ matrices.

Finally, Lemma 3.2 implies that if the statement (3.6) holds for $(n+1) \times (n+1)$ matrices then the angle between two complex numbers per A and per B, where $A, B \in \mathcal{U}^{(n+1) \times (n+1)}(\delta)$ are two matrices that differ in one row (or in one column) does not exceed

$$\frac{2\delta}{(1-\delta)\tau} = \frac{2\delta}{(1-\delta)\sqrt{\cos(\theta + 2\arcsin\delta)}} = \theta$$

and hence the statement (3.5) holds for $(n+1) \times (n+1)$ matrices.

This concludes the proof of (3.4)–(3.6) for all positive integer n. □

4 Ramifications

A similar approach can be applied to computing other quantities of interest.

4.1 Hafnians

Let $A = \left(a_{ij}\right)$ be a $2n \times 2n$ symmetric real or complex matrix. The quantity

$$\text{haf } A = \sum_{\{i_1, j_1\}, \ldots, \{i_n, j_n\}} a_{i_1 j_1} \cdots a_{i_n j_n},$$

where sum is taken over all $(2n)!/n!2^n$ unordered partitions of the set $\{1, \ldots, 2n\}$ into n pairwise disjoint unordered pairs $\{i_1, j_1\}, \ldots, \{i_n, j_n\}$, is called the *hafnian* of A, see for example, [16, Section 8.2]. For any $n \times n$ matrix A we have

$$\text{haf} \begin{pmatrix} 0 & A \\ A^T & 0 \end{pmatrix} = \text{per } A$$

and hence computing the permanent of an $n \times n$ matrix reduces to computing the hafnian of a symmetric $2n \times 2n$ matrix. The computational complexity of hafnians is understood less well than that of permanents. Unlike in the case of the permanent, no fully polynomial (randomized or deterministic) polynomial approximation scheme is known to compute the hafnian of a non-negative real symmetric matrix. Unlike in the case of the permanent, no deterministic polynomial time algorithm approximating the hafnian of a $2n \times 2n$ non-negative symmetric matrix within a factor of c^n, where $c > 0$ is an absolute constant, is known. On the other hand, there is a polynomial time randomized algorithm based on the representation of the hafnian as the expectation of the determinant of a random matrix, which approximates the hafnian of a given non-negative symmetric $2n \times 2n$ matrix within a factor of c^n, where $c \approx 0.56$ [4]. Also, for any $0 < \epsilon < 1$ fixed in advance, there is a deterministic polynomial time algorithm based on scaling, which, given a $2n \times 2n$ symmetric matrix $A = (a_{ij})$ satisfying

$$\epsilon \leq a_{ij} \leq 1 \quad \text{for all} \quad i, j,$$

computes haf A within a multiplicative factor of $n^{\kappa(\epsilon)}$ for some $\kappa(\epsilon) > 0$ [6].

With minimal changes, the approach of this paper can be applied to computing hafnians. Namely, let J denote the $2n \times 2n$ matrix filled with 1s and let us define

$$f(z) = \ln \text{haf} \left(J + z(A - J)\right).$$

Then

$$f(0) = \ln \text{haf } J = \ln \frac{(2n)!}{n!2^n} \quad \text{and} \quad f(1) = \ln \text{haf } A$$

and one can use the Taylor polynomial approximation (1.2) to estimate $f(1)$. As in Sect. 2, one can compute the right hand side of (1.2) in $n^{O(m)}$ time. The statement and the proof of Theorem 1.2 carries over to hafnians almost verbatim. Namely, let $\delta > 0$ be a real for which the Eq. (3.3) has a solution $0 < \theta < \pi/2$ (hence one can choose $\delta = 0.195$). Then haf $Z \neq 0$ as long as $Z = (z_{ij})$ is a $2n \times 2n$ symmetric complex matrix satisfying

$$\left|z_{ij} - 1\right| \leq \delta \quad \text{for all} \quad i, j.$$

Instead of the row expansion of the permanent (3.1) used in Lemmas 3.2 and 3.3, one should use the row expansion of the hafnian

$$\text{haf } Z = \sum_{j=2}^{2n} z_{1j} \text{ haf } Z_j,$$

where Z_j is the symmetric $(2n-2) \times (2n-2)$ matrix obtained from Z by crossing out the first and the jth row and the first and the jth column. As in Sect. 2, we obtain an algorithm of $n^{O(\ln n - \ln \epsilon)}$ complexity of approximating haf Z within relative error $\epsilon > 0$, where $Z = \left(Z_{ij}\right)$ is a $2n \times 2n$ symmetric complex matrix satisfying

$$\left|z_{ij} - 1\right| \leq \gamma, \quad \text{for all} \quad i, j.$$

and $\gamma > 0$ is an absolute constant (one can choose $\gamma = 0.19$).

4.2 Multidimensional Permanents

Let us fix an integer $\nu \geq 2$ and let

$$A = \left(a_{i_1 \ldots i_\nu}\right), \quad 1 \leq i_1, \ldots, i_\nu \leq n,$$

be an ν-dimensional cubical $n \times \cdots \times n$ array of real or complex numbers. We define

$$\text{PER } A = \sum_{\sigma_1, \ldots, \sigma_{\nu-1} \in S_n} \prod_{i=1}^{n} a_{i \sigma_1(i) \ldots \sigma_{\nu-1}(i)}.$$

If $\nu = 2$ then A is an $n \times n$ matrix and PER A = per A. For $\nu > 2$, it is already an NP-hard problem to tell PER A from 0 even if $a_{i_1 \ldots i_\nu} \in \{0, 1\}$ since the problem reduces to detecting a perfect matching in a hypergraph, see, for example, [2, Problem SP1]. However, for any $0 < \epsilon < 1$, fixed in advance, there is a polynomial time deterministic algorithm based on scaling, which, given a real array A satisfying

$$\epsilon \leq a_{i_1 \ldots i_\nu} \leq 1 \quad \text{for all} \quad 1 \leq i_1, \ldots, i_\nu \leq n$$

computes PER A within a multiplicative factor of $n^{\kappa(\epsilon, \nu)}$ for some $\kappa(\epsilon, \nu) > 0$ [6].

With some modifications, the method of this paper can be applied to computing this multidimensional version of the permanent. Namely, let J be the array filled with 1s and let us define

$$f(z) = \ln \text{PER} \left(J + z(A - J)\right).$$

Then

$$f(0) = \ln \text{PER } J = (\nu - 1) \ln n! \quad \text{and} \quad f(1) = \ln \text{PER } A$$

and one can use the Taylor polynomial approximation (1.2) to estimate $f(1)$. As in Sect. 2, one can compute the right hand side of (1.2) in $n^{O(m)}$ time, where the implicit constant in "$O(m)$" depends on ν. The proof of Theorem 1.2 carries to multidimensional permanents with some modifications. Namely, for some sufficiently small $\delta_\nu > 0$ the equation

$$\theta = \frac{2\delta_\nu}{(1-\delta_\nu)\sqrt{\cos\bigl((\nu-1)\theta + 2\arcsin\delta_\nu\bigr)}}$$

has a solution $\theta \geq 0$ such that $(\nu-1)\theta + 2\arcsin\delta_\nu < \pi/2$. For $\nu = 2$, we get the Eq. (3.3) with a possible choice of $\delta_2 = 0.195$, while for $\nu = 3$ we can choose $\delta_3 = 0.125$ and for $\nu = 4$ we can choose $\delta_4 = 0.093$. Then PER $Z \neq 0$ as long as $Z = \left(z_{i_1 \ldots i_\nu}\right)$ is an array of complex numbers satisfying

$$\left|z_{i_1\ldots i_\nu} - 1\right| \leq \delta_\nu \quad \text{for all} \quad 1 \leq i_1, \ldots, i_\nu \leq n.$$

We proceed as in the proof of Theorem 1.2, only instead of the first row expansion of the permanent (3.1) used in Lemmas 3.2 and 3.3, we use the first index expansion

$$\text{PER } Z = \sum_{1 \leq j_2, \ldots, j_\nu \leq n} z_{1 j_2 \ldots j_\nu} \text{ PER } Z_{j_2 \ldots j_\nu},$$

where $Z_{j_2\ldots j_\nu}$ is the ν-dimensional array of size $(n-1) \times \cdots \times (n-1)$ obtained from Z by crossing out the section with the first index 1, the section with the second index j_2 and so forth, concluding with crossing out the section with the last index j_ν. As in Sect. 2, we obtain at algorithm of $n^{O(\ln n - \ln \epsilon)}$ complexity of approximating PER Z within relative error $\epsilon > 0$, where Z is a ν-dimensional cubic $n \times \cdots \times n$ array of complex numbers satisfying

$$\left|z_{i_1\ldots i_\nu} - 1\right| \leq \gamma_\nu \quad \text{for all} \quad 1 \leq i_1, \ldots, i_\nu \leq n,$$

and $0 < \gamma_\nu < \delta_\nu$ are absolute constants (one can choose $\gamma_2 = 0.19$, $\gamma_3 = 0.12$ and $\gamma_4 = 0.09$).

Acknowledgments The author is grateful to the anonymous referees for their careful reading of the paper, useful suggestions and interesting questions.

References

1. S. Aaronson and A. Arkhipov, *The computational complexity of linear optics*, Theory of Computing **9** (2013), 143–252.
2. G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A.Marchetti-Spaccamela, and M. Protasi, *Complexity and Approximation.Combinatorial Optimization Problems and their Approximability Properties*, Springer-Verlag, Berlin 1999.
3. A. Barvinok, *Two algorithmic results for the traveling salesman problem*, Mathematics of Operations Research **21** (1996), no. 1, 65–84.
4. A. Barvinok, *Polynomial time algorithms to approximate permanents and mixed discriminants within a simply exponential factor*, Random Structures & Algorithms **14** (1999), no. 1, 29–61.
5. A. Barvinok, *Computing the partition function for cliques in a graph*, preprint arXiv:1405.1974 (2014).
6. A. Barvinok and A. Samorodnitsky, *Computing the partition function for perfect matchings in a hypergraph*, Combinatorics, Probability and Computing **20** (2011), no. 6, 815–835.
7. A. Barvinok and P. Soberón, *Computing the partition function for graph homomorphisms*, preprint arXiv:1406.1771 (2014).
8. J. Draisma, E. Horobet, G. Ottaviani, B. Sturmfels and R.R. Thomas, *The Euclidean distance degree of an algebraic variety*, preprint arXiv:1309.0049 (2013).

9. S. Friedland and L. Gurvits, *Generalized Friedland-Tverberg inequality: applications and extensions*, preprint arXiv:0603410 (2006).
10. M. Fürer, *Approximating permanents of complex matrices*, Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, ACM, New York 2000, pp. 667–669.
11. D. Gamarnik and D. Katz, *A deterministic approximation algorithm for computing the permanent of a 0, 1 matrix*, Journal of Computer and System Sciences **76** (2010), no. 8, 879–883.
12. L. Gurvits, *On the complexity of mixed discriminants and related problems*, Mathematical Foundations of Computer Science 2005, Lecture Notes in Computer Science, vol. 3618, Springer, Berlin, 2005, pp. 447–458.
13. L. Gurvits and A. Samorodnitsky, *Bounds on the permanent and some applications*, preprint arXiv:1408.0976 (2014).
14. M. Jerrum, A. Sinclair and E. Vigoda, *A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries*, Journal of the ACM **51** (2004), no. 4, 671–697.
15. N. Linial, A. Samorodnitsky, and A. Wigderson, *A deterministic strongly polynomial algorithm for matrix scaling and approximate permanents*, Combinatorica **20** (2000), no. 4, 545–568.
16. H. Minc, *Permanents*. Encyclopedia of Mathematics and its Applications, Vol. 6, Addison-Wesley Publishing Co., Reading, Mass., 1978.
17. A.D. Scott and A.D. Sokal, *The repulsive lattice gas, the independent-set polynomial, and the Lovász local lemma*, Journal of Statistical Physics **118** (2005), no. 5–6, 1151–1261.
18. L.G. Valiant, *The complexity of computing the permanent*, Theoretical Computer Science **8** (1979), no. 2, 189–201.