

# COMPUTING THE PERMANENT OF (SOME) COMPLEX MATRICES

ALEXANDER BARVINOK

June 2014

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

## 1. INTRODUCTION AND MAIN RESULTS

The *permanent* of an  $n \times n$  matrix  $A = (a_{ij})$  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  $\{1, \dots, n\}$ . The problem of efficient computation of the permanent has attracted a lot of attention. It is  $\#P$ -hard already for 0-1 matrices [Va79], but a fully polynomial randomized approximation scheme, based on the Markov Chain Monte Carlo approach, is constructed for all non-negative matrices [J+04]. 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 [L+00] and the bound was recently improved to  $2^n$  in [GS13]. An approach based on the idea of “correlation decay” from statistical physics results in a deterministic polynomial time algorithm approximating  $\text{per } A$  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 [GK10].

There is also interest in computing permanents of *complex* matrices [AA13]. The well-known Ryser’s algorithm (see, for example, Chapter 7 of [Mi78]) computes the

---

1991 *Mathematics Subject Classification.* 15A15, 68C25, 68W25.

*Key words and phrases.* permanent, hafnian, algorithm.

This research was partially supported by NSF Grant DMS 0856640.

permanent of a matrix  $A$  over any field in  $O(n2^n)$  time. A randomized approximation algorithm of [Fü00] computes the permanent of a complex matrix within a (properly defined) relative error  $\epsilon$  in  $O(3^{n/2}\epsilon^{-2})$  time. The randomized algorithm of [Gu05], see also [AA13] 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  $|a_{ij} - 1| \leq \gamma$  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  $\text{per } A$  can be computed within relative error  $\epsilon$  in  $n^{O(\ln n - \ln \epsilon)}$  time (we say that  $\alpha \in \mathbb{C}$  approximates  $\text{per } A$  within relative error  $0 < \epsilon < 1$  if  $\text{per } A = \alpha(1 + \rho)$  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

$$(1.1.1) \quad f(z) = \ln \text{per}(J + z(A - J)).$$

Clearly,

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

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

$$(1.1.2) \quad f(1) \approx f(0) + \sum_{k=1}^m \frac{1}{k!} \frac{d^k}{dz^k} f(z) \Big|_{z=0}.$$

It turns out that the right hand side of (1.1.2) can be computed in  $n^{O(m)}$  time. We present the algorithm in Section 2. The quality of the approximation (1.1.2) depends on the location of complex zeros of the permanent.

**(1.2) Lemma.** *Suppose that there exists a real  $\beta > 1$  such that*

$$\text{per}(J + z(A - J)) \neq 0 \quad \text{for all } z \in \mathbb{C} \quad \text{satisfying} \quad |z| \leq \beta.$$

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

$$f(z) = \ln \text{per}(J + z(A - J))$$

*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 (1.1.2) 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  $\text{per } A$  within relative error  $\epsilon$  in  $n^{O(\ln n - \ln \epsilon)}$  time. We prove Lemma 1.2 in Section 2.

Thus we have to identify a class of matrices  $A$  for which the number  $\beta > 1$  of Lemma 1.2 exists. We prove the following result.

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

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

then

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

We prove Theorem 1.3 in Section 3.

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

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

we can choose  $\beta = 195/190$  in Lemma 1.2 and thus obtain an approximation algorithm for computing  $\text{per } A$ .

The sharp value of the constant  $\delta$  in Theorem 1.3 is not known to the author. A simple example of a  $2 \times 2$  matrix

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

for which  $\text{per } A = 0$  shows that in Theorem 1.3 we must have

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

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

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

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

then  $\text{per } Z \neq 0$ ?

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  $A = (a_{ij})$  satisfying

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

within a multiplicative factor of  $n^{\kappa(\epsilon)}$  for some  $\kappa(\epsilon) > 0$  [BS11].

**(1.5) Ramifications.** In Section 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 [Ba14] and graph homomorphisms [BS14]. In each case, the main problem is to come up with a version of Theorem 1.3 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, [SS05].

## 2. THE ALGORITHM

**(2.1) The algorithm for approximating the permanent.** Given an  $n \times n$  complex matrix  $A = (a_{ij})$ , we present an algorithm which computes the right hand side of the approximation (1.1.2) for the function  $f(z)$  defined by (1.1.1).

Let

$$(2.1.1) \quad g(z) = \text{per}(J + z(A - J)),$$

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

$$(2.1.2) \quad \frac{d^k}{dz^k} g(z) \Big|_{z=0} = \sum_{j=0}^{k-1} \binom{k-1}{j} \left( \frac{d^j}{dz^j} g(z) \Big|_{z=0} \right) \left( \frac{d^{k-j}}{dz^{k-j}} f(z) \Big|_{z=0} \right)$$

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

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

$$(2.1.3) \quad \frac{d^k}{dz^k} g(z) \Big|_{z=0} \quad \text{for} \quad k = 1, \dots, m,$$

then the formulas (2.1.2) for  $k = 1, \dots, m$  provide a non-degenerate triangular system of linear equations that allows us to compute

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

Hence our goal is to compute the values (2.1.3).

We have

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

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

**(2.2) Proof of Lemma 1.2.** The function  $g(z)$  defined by (2.1.1) is a polynomial in  $z$  of degree  $d \leq n$  with  $g(0) = n! \neq 0$ , so we factor

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

$\alpha_1, \dots, \alpha_d$  are the roots of  $g(z)$ . By the condition of Lemma 1.2, we have

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

Therefore,

$$(2.2.1) \quad f(z) = \ln g(z) = \ln g(0) + \sum_{i=1}^d \ln \left(1 - \frac{z}{\alpha_i}\right) \quad \text{for } |z| \leq 1,$$

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 (2.2.1) 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

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

□

### 3. PROOF OF THEOREM 1.3

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

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

Thus Theorem 1.3 asserts that  $\text{per } Z \neq 0$  for  $Z \in \mathcal{U}^{n \times n}(\delta)$  and  $\delta = 0.195$ .

First, we establish a simple geometric lemma.

**(3.1) Lemma.** *Let  $u_1, \dots, u_n \in \mathbb{R}^d$  be non-zero vectors such that for some  $0 \leq \alpha < \pi/2$  the angle between any two vectors  $u_i$  and  $u_j$  does not exceed  $\alpha$ . Let  $u = u_1 + \dots + u_n$ . 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 1.3 by induction on  $n$ , using Lemma 3.1 and the following two lemmas.

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

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

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

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

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

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

Using the expansion

$$(3.2.1) \quad \text{per } Z = \sum_{j=1}^n z_{1j} \text{per } Z_j,$$

we conclude that

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

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

$$(3.2.2) \quad \sum_{j=1}^n \left| \frac{\partial}{\partial z_{1j}} \ln \text{per } Z \right| \leq \frac{1}{(1 - \delta)\tau}.$$

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

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

Then

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

Since

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

the proof follows from (3.2.2). □

**(3.3) Lemma.** *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  $\text{per } A$  and  $\text{per } B$ , interpreted as vectors in  $\mathbb{R}^2 = \mathbb{C}$  does not exceed  $\theta$ . Then for any matrix  $Z \in \mathcal{U}^{(n+1) \times (n+1)}(\delta)$ , we have

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

with

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

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

*Proof.* We use the first row expansion (3.2.1) and observe that any two matrices  $Z_j$  and  $Z_k$ , can be obtained from one from another by a 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  $\theta$ . Since

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

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.  $\square$

**(3.4) Proof of Theorem 1.3.** One can see that for a sufficiently small  $\delta > 0$ , the equation

$$(3.4.1) \quad \theta = \frac{2\delta}{(1 - \delta)\sqrt{\cos(\theta + 2 \arcsin \delta)}}$$

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

$$\theta \approx 0.7611025127.$$

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 (3.4.2)–(3.4.4) by induction on  $n$ :

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

(3.4.3) 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  $\theta$ ;

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

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

For  $n = 1$  the statement (3.4.2) 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 (3.4.3) holds as well. The statement (3.4.4) is vacuous.

Lemma 3.3 implies that if the statement (3.4.3) holds for  $n \times n$  matrices then the statement (3.4.4) holds for  $(n+1) \times (n+1)$  matrices.

The statement (3.4.4) for  $(n+1) \times (n+1)$  matrices together with the statement (3.4.2) for  $n \times n$  matrices implies the statement (3.4.2) for  $(n+1) \times (n+1)$  matrices.

Finally, Lemma 3.2 implies that if the statement (3.4.4) holds for  $(n+1) \times (n+1)$  matrices then the statement (3.4.3) holds for  $(n+1) \times (n+1)$  matrices.

This concludes the proof of (3.4.2)–(3.4.4) for all positive integer  $n$ .  $\square$

#### 4. RAMIFICATIONS

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

**(4.1) Hafnians.** Let  $A = (a_{ij})$  be a  $2n \times 2n$  real or complex matrix. The quantity

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

where sum is taken over all  $(2n)!/n!2^n$  unordered partitions of the set  $\{1, \dots, 2n\}$  into  $n$  pairwise disjoint unordered pairs  $\{i_1, j_1\}, \dots, \{i_n, j_n\}$ , is called the *hafnian* of  $A$ , see for example, Section 8.2 of [Mi78]. 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  $c \approx 0.56$  [Ba99]. 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

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

computes  $\text{haf } A$  within a multiplicative factor of  $n^{\kappa(\epsilon)}$  for some  $\kappa(\epsilon) > 0$  [BS11].

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

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

Then

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

and one can use the Taylor polynomial approximation (1.1.2) to estimate  $f(1)$ . As in Section 2, one can compute the right hand side of (1.1.2) in  $n^{O(m)}$  time. The statement and the proof of Theorem 1.3 carries over to hafnians almost verbatim. Namely, let  $\delta > 0$  be a real for which the equation (3.4.1) has a solution  $0 < \theta < \pi/2$  (hence one can choose  $\delta = 0.195$ ). Then  $\operatorname{haf} Z \neq 0$  as long as  $Z = (z_{ij})$  is a  $2n \times 2n$  symmetric complex matrix satisfying

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

Instead of the row expansion of the permanent (3.2.1) used in Lemmas 3.2 and 3.3, one should use the row expansion of the hafnian

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

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

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

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

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

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

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

If  $\nu = 2$  then  $A$  is an  $n \times n$  matrix and  $\operatorname{PER} A = \operatorname{per} A$ . For  $\nu > 2$  it is already an NP-hard problem to tell  $\operatorname{PER} A$  from 0 even if  $a_{i_1 \dots i_\nu} \in \{0, 1\}$  since the problem reduces to detecting a perfect matching in a hypergraph, see, for example, Problem SP1 in [A+99]. 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

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

computes  $\text{PER } A$  within a multiplicative factor of  $n^{\kappa(\epsilon, \nu)}$  for some  $\kappa(\epsilon, \nu) > 0$  [BS11].

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

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

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 (1.1.2) to estimate  $f(1)$ . As in Section 2, one can compute the right hand side of (1.1.2) in  $n^{O(m)}$  time, where the implicit constant in “ $O(m)$ ” depends on  $\nu$ . The proof of Theorem 1.3 carries to multidimensional permanents with some modifications. Namely, for some sufficiently small  $\delta_\nu > 0$  the equation

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

has a solution  $\theta \geq 0$  such that  $(\nu - 1)\theta + 2 \arcsin \delta_\nu < \pi/2$ . For  $\nu = 2$ , we get the equation (3.4.1) 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  $\text{PER } Z \neq 0$  as long as  $Z = (z_{i_1 \dots i_\nu})$  is an array of complex numbers satisfying

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

We proceed as in the proof of Theorem 1.3, only instead of the first row expansion of the permanent (3.2.1) used in Lemmas 3.2 and 3.3, we use the first index expansion

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

where  $Z_{j_2 \dots j_\nu}$  is the  $\nu$ -dimensional array of size  $(n-1) \times \dots \times (n-1)$  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  $j_\nu$ . As in Section 2, we obtain at algorithm of  $n^{O(\ln n - \ln \epsilon)}$  complexity of approximating  $\text{PER } Z$  within relative error  $\epsilon > 0$ , where  $Z$  is a  $\nu$ -dimensional cubic  $n \times \dots \times n$  array of complex numbers satisfying

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

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

## REFERENCES

- [AA13] S. Aaronson and A. Arkhipov, *The computational complexity of linear optics*, Theory of Computing **9** (2013), 143–252.
- [A+99] 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.
- [Ba99] 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.
- [Ba14] A. Barvinok, *Computing the partition function for cliques in a graph*, preprint [arXiv:1405.1974](https://arxiv.org/abs/1405.1974) (2014).
- [BS11] 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.
- [BS14] A. Barvinok and P. Soberón, *Computing the partition function for graph homomorphisms*, preprint [arXiv:1406.1771](https://arxiv.org/abs/1406.1771) (2014).
- [C+13] J.-Y. Cai, X. Chen and P. Lu, *Graph homomorphisms with complex values: a dichotomy theorem*, SIAM Journal on Computing **42** (2013), no. 3, 924–1029.
- [Fü00] 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.
- [GK10] 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.
- [Gu05] 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.
- [GS13] L. Gurvits and A. Samorodnitsky, *Bounds on the permanent and some applications*, preprint available at <http://www.cs.huji.ac.il/~salex/> (2013).
- [J+04] 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.
- [L+00] 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.
- [Mi78] H. Minc, *Permanents*, Encyclopedia of Mathematics and its Applications, Vol. 6, Addison-Wesley Publishing Co., Reading, Mass., 1978.
- [SS05] 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.
- [Va79] L.G. Valiant, *The complexity of computing the permanent*, Theoretical Computer Science **8** (1979), no. 2, 189–201.

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF MICHIGAN, ANN ARBOR, MI 48109-1043, USA

*E-mail address:* [barvinok@umich.edu](mailto:barvinok@umich.edu)