Exact Error Estimates and Optimal Randomized Algorithms for Integration*

Ivan T. Dimov[1] and Emanouil Atanassov[2]

[1] Institute for Parallel Processing, Bulgarian Academy of Sciences Acad. G. Bonchev Str., bl. 25 A, 1113 Sofia, Bulgaria and ACET Centre, University of Reading Whiteknights, PO Box 217, Reading, RG6 6AH, UK
I.T.Dimov@reading.ac.uk

[2] Institute for Parallel Processing, Bulgarian Academy of Sciences, Acad. G. Bonchev Str., bl. 25 A, 1113 Sofia, Bulgaria
atanasov@parallel.bas.bg

Abstract. Exact error estimates for evaluating multi-dimensional integrals are considered. An estimate is called *exact* if the rates of convergence for the low- and upper-bound estimate coincide. The algorithm with such an exact rate is called *optimal*. Such an algorithm has an *unimprovable* rate of convergence.

The problem of existing exact estimates and optimal algorithms is discussed for some functional spaces that define the regularity of the integrand. Important for practical computations data classes are considered: classes of functions with bounded derivatives and Hölder type conditions.

The aim of the paper is to analyze the performance of two optimal classes of algorithms: deterministic and randomized for computing multidimensional integrals. It is also shown how the smoothness of the integrand can be exploited to construct better randomized algorithms.

1 Introduction: Definitions and Basic Notations

The problem of evaluating integrals of high dimension is an important task since it appears in many important scientific applications of financial mathematics, economics, environmental mathematics and statistical physics. Randomized (Monte Carlo) algorithms have proved to be very efficient in solving multidimensional integrals in composite domains [16], [6].

In this paper we are interested in exact error estimates for evaluating multidimensional integrals. An estimate is called *exact* if the rates of convergence for the low- and upper-bound estimate coincide. An algorithm which reaches such an *unimprovable* rate of convergence is called *optimal*. The class of functions with bounded derivatives and Hölder type conditions are considered. We discuss the

* Partially supported by the NSF of Bulgaria through grant number I-1405/04 and by the Bulgarian IST Centre of Competence in 21 Century – BIS-21++ (contract # INCO-CT-2005-016639).

T. Boyanov et al. (Eds.): NMA 2006, LNCS 4310, pp. 131–139, 2007.

unimprovable limits of complexity for two classes of algorithms: *deterministic–* $\mathcal{A}$ and *randomized–*$\mathcal{A}^{\mathcal{R}}$. Having these unimprovable rates an important question arises: which one of the existing algorithms reaches these unimprovable rates? We analyze the *performance*, i.e., number of operations (or the computational cost) of $\mathcal{A}$ and $\mathcal{A}^{\mathcal{R}}$ classes of algorithms. It should be mentioned here that the performance analysis is connected with complexity that will be defined in Section 2. The complexity characterizes the problem for a given class of algorithms (not the algorithm itself). In Section 2 we present the computational model and show how the computational cost is connected with the complexity. In Section 3 we prove some error estimates for Hölder functions. Performance analysis of algorithms with unimprovable convergence rate is given in Section 4. Complexity of the integration problem for Hölder functions are considered in Section 5. In Section 6 we present some concluding remarks.

Let us introduce some basic notations used in the paper. By $x = (x_1, \ldots, x_d)$ we denote a point in a closed domain $G \subset I\!R^d$, where $I\!R^d$ is d-dimensional Euclidean space. The d-dimensional unite cube is denoted by $E^d = [0,1]^d$.

Definition 1. *Let d, p be integers, and $d, p \geq 1$. Consider the class $\mathbf{W}^p(\alpha; G)$ of real functions f defined over G, possessing all the partial derivatives: $D^r f = \frac{\partial^r f(x)}{\partial x_1^{r_1} \ldots \partial x_d^{r_d}}$, $r_1 + \ldots + r_d = r \leq p$, which are continuous when $r < p$ and bounded in sub norm when $r = p$. The semi-norm $\| \, . \, \|$ on $\mathbf{W}^p(\alpha; G)$ is defined as $\alpha = \| \, f \, \| = \sup \left\{ |D^p f| \, , |r_1, \ldots, r_d| = p, \quad x \equiv (x_1, \ldots, x_d) \in E^d \right\}$.*

Definition 2. *Define the class $H_\lambda^p(\alpha, G), (0 < \lambda \leq 1)$ of functions from W^p, which derivatives of order p satisfy the Hölder condition with a parameter λ:*

$$H_\lambda^p(\alpha, G) \equiv \left\{ f \in W^p : |D^p f(y_1, \ldots, y_d) - D^p f(z_1, \ldots, z_d)| \leq \alpha \sum_{j=1}^{d} |y_j - z_j|^\lambda \right\}.$$

Usually randomized algorithms reduce problems to the approximate calculation of mathematical expectations. The mathematical expectation of the random variable θ is denoted by $E_\mu(\theta)$, where μ denotes some probability measure.(The definition of probability measure is given in [11].) Sometimes $E_\mu(\theta)$ is abbreviated to $E\theta$. We shall further denote the values (realizations) of a random point ξ or random variable θ by $\xi^{(i)}$ and $\theta^{(i)} (i = 1, 2, \ldots, n)$ respectively. If $\xi^{(i)}$ is a d-dimensional random point, then usually it is constructed using d random numbers γ, i.e., $\xi^{(i)} \equiv (\gamma_1^{(i)}, \ldots, \gamma_d^{(i)})$. Let I be the desired value of the integral. Assume for a given random variable θ one can prove that $E\theta = I$. Suppose the mean value of n realizations of θ: $\theta^{(i)}$, $i = 1, \ldots, n$ is considered as a Monte Carlo approximation to the solution: $\bar{\theta}_n = 1/n \sum_{i=1}^{n} \theta^{(i)} \approx I$. One can only state that a certain randomized algorithm can produce the result with a given probability error.

Definition 3. *If I is the exact solution of the problem, then the probability error is the least possible real number R_n, for which $P = Pr \left\{ |\overline{\xi}_n - I| \leq R_n \right\}$, where $0 < P < 1$. If $P = 1/2$, then the probability error is called probable error.*

So, dealing with randomized algorithms one has to accept that the result of the computation can be true only with a certain (even high) probability. In most cases of practical computations it is reasonable to accept an error estimate with a given probability.

2 Computational Model

Consider the following problem of integration:

$$I = \int_{E^d} f(x)dx, \tag{1}$$

where $E^d \equiv [0,1]^d$, $x \equiv (x_1, \ldots, x_d) \in E^d \subset I\!R^d$ and $f \in C(E^d)$ is an integrable function on E^d. The computational problem can be considered as a mapping of function $f : \{[0,1]^d \to I\!R^d\}$ to $I\!R$ [10]: $S(f) : f \to I\!R$, where $S(f) = \int_{E^d} f(x)dx$ and $f \in F_0 \subset C(E^d)$. We will call S the solution operator. The elements of F_0 are the data, for which the problem has to be solved; and for $f \in F_0$, $S(f)$ is the exact solution. For a given f we want to compute (or approximate) $S(f)$. We will be interested to consider subsets F_0 of $C(E^d)$ and try to study how the smoothness of F_0 can be exploited. A similar approach (which is in fact included in the above mentioned consideration) is presented in [18].

We will call a *quadrature formula* any expression $A = \sum_{i=1}^{n} c_i f(x^{(i)})$, which approximates the value of the integral $S(f)$. The real numbers $c_i \in I\!R$ are called weights and d dimensional points $x^{(i)} \in E^d$ are called nodes. It is clear that for fixed weights c_i and nodes x_i the quadrature formula A may be used to define an algorithm. The algorithm A belongs to the class of deterministic algorithms $\mathcal{A}$. We call a *randomized quadrature formula* any formula of the following kind: $A^R = \sum_{i=1}^{n} \sigma_i f(\xi^{(i)})$, where σ_i and $\xi^{(i)}$ are random weights and nodes.

The computational cost of a deterministic algorithm A will be defined as a supremum (over all integrands f from F_0) of the time (number of operations) needed to perform the algorithm A:$\tau(A) = \sup_{f \in F_0} \tau(A, f)$. For a randomized algorithm $A^R \in \mathcal{A}^{\mathcal{R}}$ we will have: $\tau(A^R) = \sup_{f \in F_0} E_\mu\{\tau(A^R, f, \omega)\}$. As a good measure of the cost can be considered

$$\tau(A, f) = kn + c \quad \text{and} \quad \tau(A^R, f, \omega) = k^R n + c^R,$$

where n is the number of nodes and k, k^R are constants depending on the function f, dimensionality d and on the domain of integration (in our case on E^d) and constants c and c^R depend only on d and on the regularity parameter of the problem (in the case of $H^p_\lambda(\alpha, G)$ - on $p + \lambda$). These constants describe the so-called preprocessing operations, i.e., operations that are needed to be performed beforehand.

We assume that one is happy to obtain an ε-approximation to the solution with a probability $0 < P < 1$. For a given positive ε the ε-complexity of the integration problems S and S^R are defined as follows: $C_\varepsilon(S) = \inf_{A \in \mathcal{A}}\{\tau(A) : r(A) \le \varepsilon\}$ and $C_\varepsilon(S^R) = \inf_{A^R \in \mathcal{A}^{\mathcal{R}}}\{\tau(A^R) : r(A^R) \le \varepsilon\}$, where the errors $r(A)$

and $r(A^R)$ are defined in the Section 3. One can see that in our consideration ε-complexity characterizes the problem for a given class of algorithms (not the algorithms itself).

3 Exact Error Estimates in Functional Spaces

Generally, we assume that the problem of integration is not solved exactly, that is $S(f)$ differs from $A(f)$. We define the error as

$$r(A) = \sup_{f \in F_0} |S(f) - A(f)|$$

in the deterministic case and as

$$r(A^R) = \sup_{f \in F_0} E_\mu \left|S(f) - A^R(f,\omega)\right| = \sup_{f \in F_0} \int_{E^d} \left|S(f) - A^R(f,\omega)\right| d\mu(\omega),$$

where $A(f,\omega)$ is Σ-measurable in ω for each f in the randomized case.

Let us now define the subset $F_0 \equiv H_\lambda^p(\alpha, E^d)$. In [2] Bakhvalov proved the following theorem:

Theorem 1. *(Bakhvalov [2]) For any deterministic way of evaluating the integral (1), i.e., for any algorithm from* $\mathcal{A}$

$$\sup_{f \in H_\lambda^p(\alpha, E^d)} r(A) \geq c'(d, p+\lambda) \alpha n^{-\frac{p+\lambda}{d}} \tag{2}$$

and for any randomized way of evaluating the integral (1), i.e., for any algorithm from $\mathcal{A}^{\mathcal{R}}$

$$\sup_{f \in H_\lambda^p(\alpha, E^d)} r(A^R) \geq c''(d, p+\lambda) \alpha n^{-\frac{p+\lambda}{d} - \frac{1}{2}}. \tag{3}$$

The constants $c'(d, p+\lambda)$ and $c''(d, p+\lambda)$ depend only on d and $p+\lambda$. This theorem gives the best possible order for both algorithmic classes $\mathcal{A}$ and $\mathcal{A}^{\mathcal{R}}$.

In our work [1] we construct two randomized algorithms A_1^R and A_2^R, and prove that both have the best possible rate (3) for integrands from $W^p(\alpha, E^d)$. The proposed algorithms allow to extend the estimates for the functional class $H_\lambda^p(\alpha, E^d)$, where $0 < \lambda \leq 1$. Here we give the essential idea of the algorithms (for more details we refer to [1]). In algorithm A_1^R we divide the unit cube $\mathbf{E}^d$ into $n = q^d$ disjoint cubes: $\mathbf{E}^d = \bigcup_{j=1}^{q^d} K_j$. Then we select m random points $\xi(j,s) = (\xi_1(j,s), ..., \xi_d(j,s))$ from each cube K_j, such that all $\xi_i(j,s)$ are uniformly distributed and mutually independent. We consider the Lagrange interpolation polynomial of the function f at the point z: $L_p(f,z)$, which uses the information from the function values at exactly $\binom{p+d-1}{d}$ points satisfying a special property [1]. The second algorithm A_2^R is a modification which calculates the Newton interpolation polynomial. A_2^R involves less operations for the same

number of random nodes. Finally, we use the following randomized quadrature formula:

$$I(f) \approx A_1^R = \frac{1}{q^d m} \sum_{j=1}^{q^d} \sum_{s=1}^{m} (f(\xi(j,s)) - L_p(f, \xi(j,s))) + \int_{K_j} L_p(f,x)dx. \quad (4)$$

Now, for functions from $H_\lambda^p(\alpha, E^d)$ we can prove the following theorem:

Theorem 2. *The quadrature formula (4) satisfies*

$$R_n \leq c^{'}(d, p+\lambda)\frac{1}{m}\alpha n^{-\frac{1}{2}-\frac{p+\lambda}{d}} \quad \textit{and}$$

$$\left(E\left(\int_{\mathbf{E}^d} f(x)dx - I(f) \right)^2 \right)^{1/2} \leq c^{''}(d, p+\lambda)\frac{1}{m}\alpha n^{-\frac{1}{2}-\frac{p+\lambda}{d}},$$

where the constants $c^{'}(d, p+\lambda)$ and $c^{''}(d, p+\lambda)$ depend implicitly on the points $x^{(r)}$, but not on n.

Proof. The proof is a modification of the proof given in [1]. Indeed, taking into account that f belongs to the space $H_\lambda^p(\alpha, E^d)$ one can use the following inequality: $|f(\xi(s,t) - L_p(f, \xi(j,s)| \leq c_{d,p+\lambda}\alpha n^{-p-\lambda}$. Using the above inequality and applying similar technique used in the proof of Theorem 2.1 from [1] we prove the theorem.

Both algorithms A_1^R and A_2^R are unimprovable by rate for all functions from $H_\lambda^p(\alpha, E^d)$. Indeed, $r(A_{I_1}^R) \leq c_1''(d, p+\lambda)\alpha n^{-\frac{p+\lambda}{d}-\frac{1}{2}}$ for the algorithm A_1^R and $r(A_{I_2}^R) \leq c_2''(d, p+\lambda)\alpha n^{-\frac{p+\lambda}{d}-\frac{1}{2}}$ for the algorithm A_2^R.

4 Performance Analysis of Algorithms with Unimprovable Convergence Rate

In this subsection the computational cost of both algorithms A_1^R and A_2^R are presented. The following theorem can be proved:

Theorem 3. *[1] The computational cost of the numerical integration of a function from $H_\lambda^p(\alpha, E^d)$ using randomized algorithm A_i^R $(i = 1, 2)$ can be presented in the following form:*

$$\tau(A_i^R, x, \omega) = k_i^R n + c_i^R,$$

$$k_1^R \leq \left[m + \binom{d+p-1}{d} \right] a_f + m[d(b_r+2)+1] \quad (5)$$

$$+2\binom{d+p-1}{d}\left[m+1+d+\binom{d+p-1}{d} \right], \quad (6)$$

$$k_2^R \leq \left[m + \binom{d+p-1}{d}\right] a_f + m[d(b_r+2+k)+1] \tag{7}$$

$$+2\binom{d+p-1}{d}(d+1+m), \tag{8}$$

where b_r denotes the number of operations used to produce a uniformly distributed random number in $[0,1)$, a_f stands for the number of operations needed for each calculation of a function value, and $c_i^R = c_i^R(d, p+\lambda)$ depends only on d and $p+\lambda$.

Remark 1. The performance analysis results of Theorem 3 shows that the computational cost of both algorithms is linear with the number of nodes n. With such a cost an error of order $n^{-\frac{p+\lambda}{d}-\frac{1}{2}}$ is reached. Such an order is unimprovable in $H_\lambda^p(\alpha, E^d)$.

Optimal algorithms for functions from $W^p(\alpha, E^d)$ are also proposed in [7,12,14,4,15,17,9]. It is not an easy task to construct a unified algorithm with unimprovable rate of convergence for any dimension d and any value of p. Various methods for Monte Carlo integration that achieve the order $O\left(N^{-\frac{1}{2}-\frac{p}{d}}\right)$ are known. While in the case of $p=1$ and $p=2$ these methods are fairly simple and are widely used (see, for example, [17,14,13]), when $p \geq 3$ such methods become much more sophisticated.

Using the same construction as in [1] it is easy to show that for the deterministic case there exists an algorithm for which $r(A) \leq c'_A(d, p+\lambda)\alpha n^{-\frac{p+\lambda}{d}}$. As an example of such an algorithm could be considered the algorithm A_1^R proposed in [1] in which the nodes are fixed points.

5 Complexity of the Integration Problem for Functional Spaces

5.1 Complexity for Hölder Spaces

Now we are ready to formulate a theorem given the estimates of the ε-complexity of the problem.

Theorem 4. *For $F_0 \equiv H_\lambda^p(\alpha, E^d)$ the ε-complexity of the problem of integration S is $C_\varepsilon(S) = k\left(c_A(d, p+\lambda)\alpha\right)^{\frac{d}{p+\lambda}}\left(\frac{1}{\varepsilon}\right)^{\frac{d}{p+\lambda}}$ for the class of deterministic algorithms $\mathcal{A}$, and $C_\varepsilon(S) = k^R\left(c_{A^R}(d, p+\lambda)\alpha\right)^{\frac{d}{p+\lambda+d/2}}\left(\frac{1}{\varepsilon}\right)^{\frac{d}{p+\lambda+d/2}}$ for the class of randomized algorithms $\mathcal{A}^{\mathcal{R}}$.*

Proof. According to the definition of the cost of the algorithm we should take the worst algorithm in sense of $\tau(A, f)$ corresponding to $f \in H_\lambda^p(\alpha, E^d)$. According to the Bakhvalov's theorem [2] one can write:

$$\sup_{f \in H_\lambda^p(\alpha, E^d)} \tau(A, f) = kn + c = k\left(c'_A(d, p+\lambda)\alpha\right)^{\frac{d}{p+\lambda}}\left(\frac{1}{r(A)}\right)^{\frac{d}{p+\lambda}} + c.$$

Now, for a given $\varepsilon > 0$ we should take $\inf\left\{k\left(c'_A(d,p+\lambda)\alpha\right)^{\frac{d}{p+\lambda}}\left(\frac{1}{r(A)}\right)^{\frac{d}{p+\lambda}} : r(A) \le \varepsilon\right\}$. Let us note, that this is a non-uniform complexity notion: for each $\varepsilon > 0$ a separate A can be designed. However, following the remark that the algorithms A are uniform over the set of problems, and the fact that the infimum of the number of preprocessing operations described by c is zero, one can get:

$$C_\varepsilon(S) = k\left(c'_A(d,p+\lambda)\alpha\right)^{\frac{d}{p+\lambda}}\left(\frac{1}{\varepsilon}\right)^{\frac{d}{p+\lambda}},$$

which proves the first part of the theorem concerning the deterministic algorithms. The result for the randomized algorithms can be proved similarly.

Corollary 1. *The ε-complexity of the problem of integration strongly depends on the dimension of the problem for the class of deterministic algorithms. With the increasing of dimensionality the ε-complexity goes* ***exponentially*** *to infinity for the class $F_0 = H^p_\alpha(\lambda, E^d)$.*

Corollary 2. *In the case of randomized algorithms the ε-complexity of the integration problem for functions from $F_0 = H^p_\alpha(\lambda, E^d)$ goes asymptotically to $\left(\frac{1}{\varepsilon}\right)^2$.*

Remark 2. The fact that the ε-complexity exponentially depends on d makes the class of deterministic algorithms infeasible for large dimensions.

Remark 3. In the last case the ε-complexity does not increase exponentially with d. This is why for high-dimensional integration Monte Carlo is a right choice. Nevertheless, the results presented here demonstrate that the smoothness can be exploited to improve the rate of convergence by a factor of $n^{-\frac{p+\lambda}{d}}$ over the rate of standard randomized algorithms $n^{-\frac{1}{2}}$. This fact allows to decrease the ε-complexity from $(1/\varepsilon)^2$ by a factor of $\left(\frac{1}{\varepsilon}\right)^{-\frac{4(p+\lambda)}{2(p+\lambda)+d}}$.

6 Concluding Remarks

As a general remark, one can conclude that as smaller is the order of regularity as simpler randomized algorithm should be used. Even for low dimensions ($d = 1, 2$) Monte Carlo is a right choice if the functional class has no smoothness. It is important to note that the level of confidence P $(0 < P < 1)$ does not reflect on the rate of convergence of the probability error R_n. It reflects only on the constant k^R. That's why the choice of the value of P is not important for the convergence rate (respectively, for the rate of algorithmic complexity). Nevertheless, for practical computations it may be of great importance to have the value of the constant in order to get the number of operations for a given algorithm (as we have done in Section 4).

In case of non-regular input data (discontinues functions and/or singularities) there are special techniques well developed in Monte Carlo algorithms [8,16,3,1]. These techniques allow to *include* the singularity into the density function of special choice (see, for instance [4,5]).

As a general remark it should be emphasized that the randomized algorithms have better convergence rate for the same regularity of the input data. The results can be extended to optimal algorithms for solving integral equations. An important obvious advantage of randomized algorithms is the case of *bad functions*, i.e., functions that do not satisfy some additional conditions of regularity. The main problem with the deterministic algorithms is that normally they need some additional approximation procedure that require additional regularity. The randomized algorithms do not need such procedures. But one should be careful because

- the better convergence rate for randomized algorithms is reached with a given probability less than 1, so the advantage of Monte Carlo algorithms is a matter of definition of the probability error. Such a setting of the problem of error estimation may not be acceptable if one needs a guaranteed accuracy or strictly reliable results. In fact, we see that this is a price paid by randomized algorithms to increase their convergence rate.
- If the nature if the problem under consideration do not allow to use the probability error for estimates or the answer should be given with a guaranteed error then the higher convergence order randomized algorithms are not acceptable.

References

1. E. Atanassov, I. Dimov, *A new Monte Carlo method for calculating integrals of smooth functions*, Monte Carlo Methods and Applications, **Vol. 5**, No 2 (1999), pp. 149-167.
2. N.S. Bachvalov, *On the approximate computation of multiple integrals*, Vestnik Moscow State University, Ser. Mat., Mech., **Vol. 4** (1959), pp. 3–18.
3. Dimov I.T., *Minimization of the probable error for some Monte Carlo methods*, Proc. of the Summer School on Mathematical Modelling and Scientific Computations, Bulgaria, Sofia, Publ. House of the Bulg. Acad. Sci. (1991), pp. 159–170.
4. I. Dimov, *Efficient and Overconvergent Monte Carlo Methods. Parallel algorithms.*, Advances in Parallel Algorithms (I. Dimov, O. Tonev, Eds.), Amsterdam, IOS Press (1994), pp. 100–111.
5. I. Dimov, A. Karaivanova, H. Kuchen, H. Stoltze, *Monte Carlo Algorithms for Elliptic Differential Equations. Data Parallel Functional Approach*, Journal of Parallel Algorithms and Applications, **Vol. 9** (1996), pp. 39–65.
6. I. Dimov, O. Tonev, *Monte Carlo algorithms: performance analysis for some computer architectures*, J. of Computational and Applied Mathematics, **Vol. 48** (1993), pp. 253–277.
7. J.H. Halton, D.C. Handscome, *A method for increasing the efficiency of Monte Carlo integrations*, J. Assoc. comput. machinery, **Vol. 4, No. 3** (1957), pp. 329–340.
8. M.H. Kalos, *Importance sampling in Monte Carlo calculations of thick shield penetration*, Nuclear Sci. and Eng., **2, No. 1** (1959), pp. 34–35.
9. A. Karaivanova, I. Dimov, *Error analysis of an Adaptive Monte Carlo Method for Numerical Integration*, Mathematics and Computers in Simulation, **Vol. 47** (1998), pp. 201-213.

10. Ker-I Ko, *Computational Complexity of Real Functions*, Birkhauser Boston, Boston, MA, 1991
11. A.N. Kolmogorov, *Foundations of the Theory of Probability*, Second English Edition, Chelsea Publishing Company, New York, 1956.
12. E. Novak, K. Ritter, *Optimal stochstic quadrature farmulas for convex functions*, BIT, **Vol. 34** (1994), pp. 288–294.
13. E. Novak, K. Ritter, *High Dimensional Integration of Smooth Functions over cubes*, Numerishche Mathematik (1996), pp. 1–19.
14. Bl. Sendov, A. Andreev, N. Kjurkchiev, *Numerical Solution of Polynomial Equations, Handbook of Numerical Analysis (General Editors: P.G. Ciarlet and J. L. Lions), Vol. 3)*, **Solution of Equations in R^n (Part 2), North-Holland**, Amsterdam, N.Y., 1994.
15. S.A. Smolyak, *Quadrature and inperpolation formulas for tensor products of certain classes of functions*, Soviet Math. Dokl., **Vol. 4** (1963), pp. 240–243.
16. I.M. Sobol, *Monte Carlo numerical methods*, Nauka, Moscow, 1973.
17. I.M. Sobol, *On Quadratic Formulas for Functions of Several Variables Satisfying a General Lipschitz Condition*, USSR Comput. Math. and Math. Phys., **Vol. 29(6)** (1989), pp. 936 – 941.
18. Traub, J.F., Wasilkowski, G.W. and Wozniakowski, H., *Information-Based Complexity*, Acad. Press, INC., New York, 1988.