Parallel Importance Separation for Multiple Integrals and Integral Equations

Sofiya Ivanovska and Aneta Karaivanova

IPP – Bulgarian Academy of Sciences
Acad. G. Bonchev St., Bl.25A, 1113 Sofia, Bulgaria
{sofia,anet}@parallel.bas.bg

Abstract. In this paper we present error and performance analysis of a Monte Carlo variance reduction method for solving multidimensional integrals and integral equations. This method, called importance separation, combines the idea of separation of the domain into uniformly small subdomains with the approach of importance sampling. The importance separation method is originally described in our previous works, here we generalize our results and discuss the performance in comparison with crude Monte Carlo and importance sampling. Based on our previous investigation we propose efficient parallelizations of the importance separation method. Numerical tests implemented on PowerPC cluster using MPI are provided. The considered algorithms are carried out using pseudorandom numbers.

1 Introduction

Multidimensional numerical quadratures are of great importance in many practical areas, ranging from atomic physics to finance. The crude Monte Carlo method has rate of convergence $O(N^{-1/2})$ which is independent of the dimension of the integral, and that is why Monte Carlo integration is the only practical method for many high-dimensional problems.

Much of the efforts to improve Monte Carlo method (MCM)are in construction of variance reduction methods which speed up the computation. Importance sampling is probably the most widely used Monte Carlo variance reduction method, [11,6,13,14]. One use of importance sampling is to emphasize rare but important events, i.e., small regions of space in which the integrand is large. One of the difficulties in this method is that sampling from the importance density is required, but this can be performed using acceptance-rejection.

In [8] a method called importance separation (IS) was introduced. This method combines the ideas from importance sampling and stratification. The IS method has the best possible rate of convergence for certain class of functions but its disadvantage is that it gives better accuracy only for low dimensions and its increased computational complexity. This method was applied for evaluation of multidimensional integrals [3] and for solving integral equations [5].

In this paper we consider both problems, solving multiple integrals and integral equations through unified point of view converting the problem of solving of

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3039, pp. 499–506, 2004.

integral equations into approximate calculation of a finite number of integrals (linear functionals of iterative functions), then importance separation is applied to the each of the integrals. We also describe the parallel implementation of the two algorithms based on IS; it has some difficulties due to hierarchical structure of the method.

2 Formulation of the Problem

2.1 Calculation of Multidimensional Integrals

Consider the problem of approximate calculation of the multiple integral

$$I = \int_G f(x)p(x)\,\mathrm{d}x, \;\; G \equiv [0;1]^d \tag{1}$$

where $f(x)$ is an integrable function for any $x \in G \subset \mathbb{R}^d$ and $p(x) \geq 0$ is a probability density function, such that $\int_G p(x)\,\mathrm{d}x = 1$.
The Monte Carlo quadrature formula is based on the probabilistic interpretation of an integral. If $\{x_n\}$ is a sequence in G sampled with density $p(x)$, then the Monte Carlo approximation to the integral is, [12],

$$I \approx I_N[f] = \frac{1}{N}\sum_{n=1}^{N} f(x_n)$$

with the integration error $\varepsilon_N = |I - I_N| \approx \sqrt{\frac{Var(f)}{N}}$.

2.2 Solving Integral Equations

Consider the Fredholm integral equation of the second kind:

$$u(x) = \int_{\Omega} k(x,x')u(x')\,\mathrm{d}x' + f(x)$$

or

$$u = \mathcal{K}u + f \;\; (\mathcal{K} \text{ is an integral operator}), \quad \text{where}$$

$k(x,x') \in L_2(\Omega \times \Omega)$, $f(x) \in L_2(\Omega)$ are given functions and $u(x) \in L_2(\Omega)$ is an unknown function, $x, x' \in \Omega \subset \mathbb{R}^d$ (Ω is a bounded domain).
We are interested in Monte Carlo method for evaluation of linear functionals of the solution of the following type:

$$J(u) = \int \varphi(x)u(x)\,\mathrm{d}x = (\varphi, u). \tag{2}$$

It is assumed that $\varphi(x) \in L_2(\Omega)$. We can apply successive approximation method for solving integral equations:

$$u^{(i)} = \sum_{j=0}^{i} \mathcal{K}^{(j)} f = f + \mathcal{K}f + \ldots + \mathcal{K}^{(i-1)}f + \mathcal{K}^{(i)}f, \quad i = 1, 2, \ldots \tag{3}$$

where $u^{(0)}(x) \equiv f(x)$. It is known that the condition $\|\mathcal{K}\|_{L_2} < 1$ is a sufficient condition for convergence of the Neumann series. Thus, when this condition is satisfied, the following statement holds:

$$u^{(i)} \longrightarrow u \quad \text{as} \quad i \to \infty.$$

Therefore,

$$J(u) = (\varphi, u) = \lim_{i\to\infty} (\varphi, u^{(i)}) = \lim_{i\to\infty} \left(\varphi, \sum_{j=0}^{i} \mathcal{K}^{(j)} f\right) = \lim_{i\to\infty} \sum_{j=0}^{i} \left(\varphi, \mathcal{K}^{(j)} f\right).$$

An approximation of the unknown value (φ, u) can be obtained using a truncated Neumann series (3) for sufficiently large i:

$$(\varphi, u^{(i)}) = (\varphi, f) + (\varphi, \mathcal{K}f) + \ldots + (\varphi, \mathcal{K}^{(i-1)} f) + (\varphi, \mathcal{K}^{(i)} f).$$

So, we transform the problem for solving integral equations into a problem for approximate evaluation of a finite number of multidimensional integrals. We will use the following denotation $(\varphi, \mathcal{K}^{(j)} f) = I(j)$, where $I(j)$ is a value, obtained after integration over $\Omega^{j+1} = \Omega \times \ldots \times \Omega$, $j = 0, \ldots, i$. It is obvious that the calculation of the estimate $(\varphi, u^{(i)})$ can be replaced by evaluation of a sum of linear functionals of iterative functions of the following type $(\varphi, \mathcal{K}^{(j)} f), j = 0, \ldots, i$, which can be presented as:

$$\begin{aligned}(\varphi, \mathcal{K}^{(j)} f) &= \int_{\Omega} \varphi(t_0) \mathcal{K}^{(j)} f(t_0)\, \mathrm{d}t_0 = \\ &= \int_{G} \varphi(t_0) k(t_0, t_1) \ldots k(t_{j-1}, t_j) f(t_j)\, \mathrm{d}t_0 \ldots \mathrm{d}t_j,\end{aligned} \tag{4}$$

where $t = (t_0, \ldots, t_j) \in G \equiv \Omega^{j+1} \subset \mathbb{R}^{d(j+1)}$. If we denote by $F(t)$ the integrand function

$$F(t) = \varphi(t_0) k(t_0, t_1) \ldots k(t_{j-1}, t_j) f(t_j), \quad t \in \Omega^{j+1},$$

then we will obtain the following expression for (4):

$$I(j) = (\varphi, \mathcal{K}^{(j)} f) = \int_{G} F(t)\, \mathrm{d}t, \quad t \in G \subset \mathbb{R}^{d(j+1)}. \tag{5}$$

So, from now on we will consider the problem for approximate calculation of multiple integrals of the type (5). We will first review briefly the most widely used Monte Carlo methods for integrals and integral equations. It is well-known that Monte Carlo methods reduce the problem to the approximate calculation of mathematical expectation which coincides with the unknown functional defined by (2).

3 Importance Separation for Integrals

The importance separation is a Monte Carlo method which combines the idea of separation of the domain of integration into uniformly small subdomains (stratification, [4]) and the Kahn approach to implement more samples in those subdomains where the integrand is large (importance sampling for integrals, [7], and for integrals equations, [2,9]). This method has the best rate of convergence for the class of functions with bounded derivatives (see [1]).

One approach how to make a partition of the given domain into subdomains was studied in [8] where the problem for evaluation of the integral $I(j) = \int_G F(t)\,\mathrm{d}t$ is considered. The suggested there partition scheme of the domain $G = [a; b]$ into M subintervals (one-dimensional case) is the following one:

$$G = \bigcup_{l=1}^{M} G_l, \quad G_l \equiv [x_{l-1}, x_l], \quad l = 1, \dots, M-1,$$
$$C_i = \frac{1}{2}[F(x_{i-1}) + F(x_M)](x_M - x_{i-1}), \quad i = 1, \dots, M-1, \tag{6}$$
$$x_i = x_{i-1} + \frac{C_i}{F(x_{i-1})(M-i+1)}, \qquad x_0 = a, \; x_M = b.$$

It is known (see [12]) that

$$\mathbf{E}\theta_N^*(j) = I(j),$$

where

$$\theta_N^*(j) = \sum_{i=1}^{M} \frac{\mathrm{V}(G_i)}{N_i} \sum_{l=1}^{N_i} F(\xi_l^{(i)}), \quad \sum_{i=1}^{M} N_i = N,$$

and $\xi_l^{(i)}$ is a random point in the i-th subdomain of G.

In the general case of multidimensional integrals ($G \subset \mathbb{R}^n$) the following integration error (the probable error) holds [8]:

$$r_N \le \sqrt{2}n \left[\frac{1}{N} \sum_{i=1}^{N} (\hat{L_i} c_{1_i} \hat{c_{2_i}})^2 \right]^{\frac{1}{2}} N^{-\frac{1}{2}-\frac{1}{n}}, \quad M = N, \tag{7}$$

where n is the dimension of the domain of integration, M is the number of subdomains, the integrand is a positive function $F(t)$, which belongs to $W^{(1)}(L, G)$. This means that $F(t)$ is continuous on G with partially continuous first derivatives and

$$\left| \frac{\partial F}{\partial t_l} \right| \le L_{i_l}, \; l = 1, \dots, d, \; t \in G_i, \; L_i = (L_{i_1}, \dots, L_{i_d}), \; \hat{L_i} = \max_l L_{i_l}.$$

The constants $c_{1_i} (i = 1, \dots, M)$ and the vectors of constants $c_{2_i} \in \mathbb{R}^d$ are determined from the requirement the subdomains $G_i, \; i = 1, \dots, M$ have to be uniformly small in probability and in geometrical size, and it is also assumed that $\hat{c_{2_i}} = \max_l c_{2_{i_l}}$.

From (7) it is clear that the error of the importance separation method which has the order $\mathcal{O}(N^{-1/2-1/n})$ asymptotically goes to $\mathcal{O}(N^{-1/2})$ for large dimensions n. This estimation of integration error shows that importance separation can be considered as a good method for approximate calculation of integrals only if n is not very large. Therefore when we translate this conclusion in the terms of integral equation, it means that the von Neumann series has to converge quickly.

4 Parallel Implementation

In this section we present the parallel importance separation for evaluation of multiple integrals and solving integral equations. The crude Monte Carlo possesses inherent parallelism which is based on the possibility to calculate simultaneously realizations of the random variable on different processors. For our algorithm (importance separation) we have some additional work: partitioning of the domain. We consider a multiprocessor configuration with p nodes.
N uniformly distributed random points $x_i \in [0;1]^d, \ i = 1, \ldots, N$ are used to obtain an approximation with given accuracy of the integral (1). For generation of $d-$dimensional random point we need d random numbers. To estimate the performance of the parallel algorithms we use:

$\mathrm{E}\mathbf{T_p(A)}$ mathematical expectation of time, required for a set of p processing elements to solve the problem using algorithm A

$\mathbf{S_p(A)} = \dfrac{\mathrm{E}\mathbf{T_1(A)}}{\mathrm{E}\mathbf{T_p(A)}}$ speed-up

$\mathbf{E_p(A)} = \dfrac{\mathbf{S_p(A)}}{\mathbf{p}}$ parallel efficiency.

5 Numerical Experiments

We present the numerical results (accuracy, CPU-time in seconds, parallel efficiency, speed-up) for the considered algorithm, importance separation, applied to solving of multidimensional integrals and integral equations using pseudorandom number sequences. The numerical tests are implemented on a cluster of 4 two-processor computers Power Macintosh using MPI.

5.1 Calculation of Integrals

Here we present the results of solving of a multidimensional integral, which is used as a test example in [10].
Example 1. This example is Monte Carlo integration over $I^5 = [0,1]^5$ of the function

$$f_1(x) = exp\left(\sum_{i=1}^{5} a_i\, x_i^2\, \frac{2 + sin(\sum_{j=1, j\neq i}^{5} x_j)}{2}\right),$$

where $\mathbf{a} = (1, \frac{1}{2}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5})$. The numerical results for the accuracy of the described methods for computing the multidimensional quadratures are presented in Table 1. The results are presented as a function of N, number of samples, and as a function of the error, which is computed with respect to the exact solution. The importance separation method leads to smaller errors. The most important fact is that using importance separation we have very good accuracy even using small sample. The superior behavior of importance separation is illustrated also by Figure 1. Table 2 presents the achieved efficiency of the parallel implementation (using MPI) for the considered method. The speed-up is almost linear and the efficiency grows with the increase number of samples.

Table 1. Comparison between Crude MCM, Importance sampling, Importance separation for Example 1 (calculations are implemented on one processor)

N	Crude MCM		Imp. sampling		Imp. separation	
	$\lvert I - I_N \rvert$	T_1	$\lvert I - I_N \rvert$	T_1	$\lvert I - I_N \rvert$	T_1
100	0.009532	0.001	0.081854	0.008	0.000316	6
500	0.092960	0.004	0.007102	0.036	0.000003	31
2500	0.009027	0.020	0.006381	0.175	0.000068	152
10000	0.006611	0.076	0.004673	0.697	0.000061	610
50000	0.008443	0.386	0.003212	3.489	0.000021	3047

Table 2. Implementation of IS for Example 1 using MPI ($I = 2.923651$)

Importance separation					
$N = 1000$			$N = 10000$		
p	I_N	E_p	p	I_N	E_p
1	2.923604	1	1	2.923590	1
2	2.923603	0.979	2	2.923573	0.985
3	2.920636	0.967	3	2.923336	0.983
4	2.923804	0.941	4	2.923638	0.980
5	2.923463	0.934	5	2.923602	0.979
6	2.911825	0.925	6	2.922537	0.977

5.2 Solving Integral Equations

We use the following integral equation as a test **Example 2**:

$$u(x) = \int_{\Omega} k(x, x')u(x')\,\mathrm{d}x' + f(x), \qquad \text{where}$$

$$k(x, x') = \frac{0.055}{1 + \mathrm{e}^{-3x}} + 0.07, \qquad (\|\mathcal{K}\|_{L_2} \approx 0.2) \tag{8}$$

$$f(x) = 0.02(3x^2 + \mathrm{e}^{-0.35x}), \qquad \Omega \equiv [-2; 2].$$

This kind of equation describes some neuron networks procedures. We are interested in an approximate calculation of (φ, u), where $\varphi(x) = 0.7((x + 1)^2 \cos(5x) + 20)$. The results for parallel efficiency are presented as a function of the number of nodes p. The importance separation algorithm is constructed so that only one sample of the random variable is chosen in every subdomain. The number of iterations d is fixed, but it has been chosen in advance according to the L_2-norm of the kernel (8). For the approximate computation of any integral $I(j),\ j = 0, \dots, i$ different number of samples are used in order to have error balancing. The proposed parallel algorithm for importance separation shares the obtained subdomains among the processors. Thus, every node processes the corresponding set of subdomains independently. This fact and insignificant quantity of data that is transferred determine comparatively high parallel efficiency of the algorithm. The results for the achieved efficiency are given in Table 3, which illustrates the inherent parallelism of Monte Carlo methods.

Table 3. Relative error, CPU-time (in seconds) and parallel efficiency in the approximate calculation of (φ, u) for Example 2. The number of samples used for calculation of each of the integrals $I(j),\ j = 0, \ldots, 5$ is denoted by N_j

$N_0 = 240,\ N_1 = 18^2$ $N_2 = 12^3,\ N_3 = 6^4$ $N_4 = 6^5,\ N_5 = 6^6$				$N_0 = 480,\ N_1 = 24^2$ $N_2 = 18^3,\ N_3 = 12^4$ $N_4 = 6^5,\ N_5 = 6^6$				$N_0 = 480,\ N_1 = 30^2$ $N_2 = 24^3,\ N_3 = 12^4$ $N_4 = 12^5,\ N_5 = 6^6$			
p	Rel. error	Time	E_p	p	Rel. error	Time	E_p	p	Rel. error	Time	E_p
1	0.1221	0.63	1	1	0.0014	0.81	1	1	0.0009	3.18	1
2	0.1079	0.31	0.997	2	0.0016	0.40	0.997	2	0.0005	1.59	0.999
3	0.0994	0.21	0.988	3	0.0036	0.27	0.989	3	0.0005	1.06	0.996
4	0.0272	0.16	0.976	4	0.0122	0.21	0.979	4	0.0010	0.80	0.994
6	0.1986	0.11	0.962	6	0.0046	0.14	0.967	6	0.0036	0.53	0.990

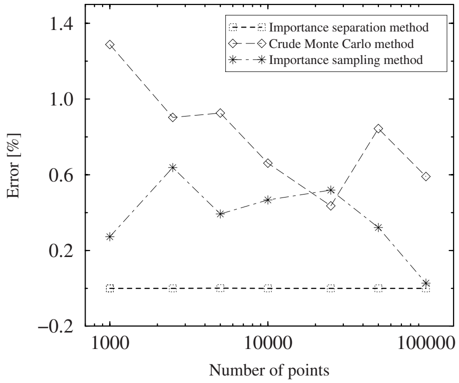

Fig. 1. Comparison of the accuracy of Crude MCM, Importance sampling, and Importance separation for Example 1

Acknowledgments. This work is supported by Center of Excellence BIS-21 Grant ICA1-2000-70016 and by the Ministry of Education and Science of Bulgaria under Grants # I 1201/02 and # MM 902/99.

References

1. N. S. Bahvalov. On the optimal estimations of convergence of the quadrature processes and integration methods, *Numerical Methods for Solving Differential and Integral Equations*, Nauka, Moscow, 5–63, 1964, (in Russian).
2. I. Dimov. Minimization of the probable error for some Monte Carlo methods, Mathematical Modelling and Scientific Computations, Andreev, Dimov, Markov, Ulrich (Eds.), Bulgarian Academy of Sciences, Sofia, 159-170, 1991.
3. I. Dimov, A. Karaivanova, R. Georgieva, and S. Ivanovska, Parallel Importance Separation and Adaptive Monte Carlo Algorithms for Multiple Integrals, Numerical Methods and Applications (I. Dimov, I.Lirkov, S. Margenov, and Z. Zlatev - Eds.), LNCS 2542, 99-107, Springer, 2003.

4. V. Dupach. Stochasticke pocetni metody, Cas. pro pest. mat. 81(1), 55-68, 1956.
5. Rayna Georgieva and Sofiya Ivanovska. Importance Separation for Solving Integral Equations, Large-Scale Scientific Computing (I. Lirkov, S. Margenov, J. Wasniewski, and P. Yalamov - Eds.), LNCS 2907, 144-152, Springer, 2004.
6. T. Hesterberg. Weighted average importance sampling and defensive mixture distributions, *Technometrics*, 37(2), 185–194, 1995.
7. H. Kahn. Random sampling (Monte Carlo) techniques in neutron attenuation problems, Nucleonics 6(5), 27-33, 1950; 6(6), 60-65, 1950.
8. A. Karaivanova. Adaptive Monte Carlo methods for numerical integration, *Mathematica Balkanica*, 11, 391–406, 1997.
9. G. A. Mikhailov. Optimization of the "weight" Monte Carlo methods, Moskow, 1987.
10. B. Moskowitz and R. E. Caflisch. Smoothness and dimension reduction in quasi-Monte Carlo methods, *J. Math. Comput. Modeling*, **23**: 37–54, 1996.
11. A. Owen and Y. Zhou. Safe and effective importance sampling, Technical report, Stanford University, Statistics Department, 1999.
12. I. M. Sobol. *Monte Carlo Numerical Methods*, Nauka, Moscow, 1973, (in Russian).
13. E. Veach and L. J. Guibas. Optimally combining sampling techniques for Monte Carlo rendering, *Computer Graphics Proceedings*, Annual Conference Series, *ACM SIGGRAPH '95*, 419–428, 1995.
14. E. Veach. *Robust Monte Carlo Methods for Light Transport Simulation*, Ph.D. dissertation, Stanford University, 1997.