QSPEEDUP: KVAZI-MONTE KARLO IMPLEMENTACIJA SPEEDUP KODA ZA RAČUNANJE FUNKCIONALNIH INTEGRALA
QSPEEDUP: QUASI-MC IMPLEMENTATION OF THE SPEEDUP PATH INTEGRAL CODE

Dušan Vudragović, Antun Balaž, Aleksandar Belić, Aleksandar Bogojević
Scientific Computing Laboratory, Institute of Physics Belgrade, Serbia

Sadržaj – *Značajno ubrzanje Monte Karlo algoritma za izračunavanje funkcionalnih integrala u kvantnoj mehanici postignuto je upotrebom novog analitičkog metoda koji sistematski poboljšava numeričku konervenciju diskretizovanih amplituda ka njihovim kontinuum vrednostima. Ovakav optimizovani Monte Karlo algoritam je implementiran u SPEEDUP Monte Karlo kodu. U ovom radu opisujemo značajno poboljšanje ovog algoritma upotrebom kvazi-Monte Karlo metoda za efikasno generisanje relevantnih trajektorija za nekoliko jednostavnih kvantnih modela.*

Abstract - *Significant speedup of the Monte Carlo algorithm for calculation of path integrals of a generic quantum mechanical theory is achieved using the new analytic method, which systematically improves numerical convergence of discretized amplitudes to their continuum values. This optimized Monte Carlo algorithm is implemented in the SPEEDUP code. In this paper we describe further significant improvements of the algorithm through the use of the quasi-Monte Carlo method for efficient generation of relevant trajectories for several simple quantum models.*

1. INTRODUCTION

Exact solution of a given many-body model in non-relativistic quantum theory is usually expressed in terms of eigenvalues and eigenfunctions of the corresponding Hamiltonian

$$\hat{H}=\sum_{i=1}^{M}\frac{\hat{\mathbf{p}}_i^2}{2m_i}+\hat{V}(\hat{\mathbf{q}}_1,...,\hat{\mathbf{q}}_M)\,, \qquad (1)$$

where M represents the number of particles. The complete analytic solution of the model can be also expressed in terms of general transition amplitudes $A(\mathbf{a},\mathbf{b};T)=\langle\mathbf{b}|e^{-iT\hat{H}/\hbar}|\mathbf{a}\rangle$ from the initial state $|\mathbf{a}\rangle$ to the final state $|\mathbf{b}\rangle$ during the time of propagation T. However, exact solutions can be found only in a very limited number of cases. Therefore, use of various analytic approximation techniques or numerical treatment is necessary for detailed understanding of the behavior of almost all models of interest.

Recently introduced effective action approach [1-5] provides an ideal framework for exact numerical calculation of such quantum amplitudes. It gives systematic short-time expansion of transition amplitudes for a general potential, thus allowing accurate calculation of relevant short-time properties of quantum systems directly, as has been demonstrated in Refs. [6-8]. For numerical calculations that require long times of propagation to be considered, relying on the use of Monte Carlo method, the effective action approach provides improved discretized actions leading to the speedup in the convergence of numerically calculated discretized quantities to their exact continuum values. This has been demonstrated not only for the amplitudes, but also in Monte Carlo calculations of energy expectation values using the improved energy estimators [9, 10].

From inception of the path integral formalism, expansion of short-time amplitudes in the time of propagation was used for the definition of path integrals through the time-discretization procedure [11, 12]. This is also straightforwardly implemented in the Path Integral Monte Carlo approaches [13], where one usually relies on the naive discretization of the action. Several improved discretized actions, mainly based on the Trotter formula and its generalizations, were developed and used in the past [14-16].

The effective action approach is based on the concept of ideal discretization [4]. It was introduced first for single-particle 1D models [1-3] and later extended to general many-body systems in arbitrary number of spatial dimensions [10, 5]. This approach allows systematic derivation of higher-order terms to a chosen order p in the short time of propagation. Recursive method for deriving the discretized effective actions, established in Ref. [5], is based on solving the underlying Schrödinger equation for the amplitude. It represents the most efficient tool to analytically calculate higher-order effective actions.

We will illustrate this approach on the example of one-dimensional quantum theory. In this case, the transition amplitudes are expressed in terms of the ideal discretized action S^* in the form

$$A(a,b;T)=\frac{1}{\sqrt{2\pi T}}e^{-S^*(a,b;T)}\,, \qquad (2)$$

which can be also seen as a definition of the ideal action []. Therefore, by definition, the above expression is correct not only for short times of propagation, but also for arbitrary large T. The ideal effective potential W is introduced by

$$S^*(a,b;T)=T\left[\frac{1}{2}\left(\frac{b-a}{T}\right)^2+W\right], \qquad (3)$$

as a reminiscent of the naive discretized action, with the arguments usually written in the form $W\left(\frac{a+b}{2},\frac{b-a}{2};T\right)$, to emphasize that we will be using the mid-point prescription. As was shown earlier, the effective potential is symmetric in its second argument, and allows systematic and hierarchic double expansion in the form

$$W(x,\bar{x};\varepsilon)=\sum_{m=0}^{\infty}\sum_{k=0}^{m}c_{m,k}(x)\varepsilon^{m-k}\bar{x}^{2k}\,. \qquad (4)$$

If we restrict the above sum over m to $p-1$, the obtained truncated level p effective potential $W_p(x,\bar{x};\varepsilon)$ gives the expansion of the effective action S_p^* to order ε^p, and hence

the level designation p for both the effective action and the corresponding potential W_p.

As shown previously [1-3], when used in Path Integral Monte Carlo simulations for calculation of long time amplitudes, the use of level p effective action leads to the convergence of discretized amplitudes proportional to ε^p, i.e. as $1/N^p$, where N is the number of time steps $\varepsilon = T/N$ used in the discretization. This was implemented in the SPEEDUP code [17] and used in several numerical studies [6-8].

In this paper we present QSPEEDUP code, which implement the effective action approach using the SPEEDUP algorithm, and quasi-Monte Carlo method to efficiently generate relevant trajectories. We verify the correctness of the new code by comparison with the standard SPEEDUP MC implementation and study in detail the performance and behavior or errors of quasi-MC algorithm. In Section 2 we first give brief overview of the algorithm used in the SPEEDUP and QSPEEDUP codes. Section 3 introduces quasi-MC method and low-discrepancy sequences, and tests the performance of this method on simple but instructive examples of low-dimensional integrals of the Gaussian type. In Section 4 we describe details of quasi-MC implementation of the SPEEDUP code and present numerical results obtained for several one-dimensional quantum models. Section 5 summarizes the obtained results.

2. SPEEDUP ALGORITHM

In the standard Path Integral Monte Carlo approach based on the use of effective actions, the time of propagation T is divided into N time steps, such that $\varepsilon = T/N$ is sufficiently small and that the effective potential (which has the finite radius of convergence) can be used. This applies to the original Feynman's definition of path integrals [11, 12, 18], which corresponds to $p=1$, as well as to the higher-order effective actions. The discretization of the propagation time leads to the following expression for the discretized amplitude

$$A_N^{(p)}(a,b;T) = \int \frac{dq_1 \ldots dq_{N-1}}{(2\pi\varepsilon)^{N/2}} \exp(-s_N^{(p)}) , \qquad (5)$$

where $S_N^{(p)}$ stands for the discretized level p effective action,

$$S_N^{(p)} = \sum_{k=0}^{N-1} \left\lfloor \frac{(q_{k+1}-q_k)^2}{2\varepsilon} + \varepsilon W_p(x_k, \bar{x}_k; \varepsilon) \right\rfloor , \qquad (6)$$

and we have used the abbreviations $q_0 = a$, $q_N = b$, $x_k = (q_{k+1}+q_k)/2$, $\bar{x}_k = (q_{k+1}-q_k)/2$.

The trajectory of q's is constructed using the bisection method [19]. The procedure starts from bisection level $n=0$, where we only have initial and final position of the particle (i.e. the trajectory consists of only these two points). At the next bisection level $n=1$, the propagation is divided into two time-steps, and we have to generate a coordinate q of the particle at the moment $T/2$, thus constructing the piecewise trajectory connecting points a at the time $t=0$, q at $t=T/2$, and b at $t=T$. The coordinate q is generated from the Gaussian probability density function centered at the mid-point $(a+b)/2$, with the width $\sigma_1 = \sqrt{T/2}$. The procedure continues iteratively, and each time a set of points is added to the piecewise trajectory. At each bisection level n the coordinates are generated from the Gaussian centered at mid-points of coordinates generated at previous level $n-1$, with the width $\sigma_n = \sqrt{T/2^n}$. To generate numbers η from the Gaussian centered at zero we use the standard Box-Müller method [20],

$$\eta = \sqrt{-2\sigma_n^2 \ln \xi_1} \cos 2\pi\xi_2 , \qquad (7)$$

where numbers ξ_1 and ξ_2 are generated from the uniform distribution on the interval [0,1], using the SPRNG library [21]. If the target (maximal) bisection level is s, then at bisection level $n \le s$ we have to generate 2^{n-1} numbers from the Gaussian distribution using the above formula, and to construct the new trajectory by adding to already existing points the new ones, according to

$$q[(1+2i)\cdot 2^{s-n}] = \eta_i + \frac{1}{2}\left(q[i\cdot 2^{s-n+1}] + q[(i+1)\cdot 2^{s-n+1}]\right) , \qquad (8)$$

where i runs from 0 to $2^{n-1}-1$. This ensures that at bisection level s we get trajectory with $N = 2^s$ time-steps, consisting of $N+1$ points, with boundary conditions $q[0] = a$ and $q[N] = b$. At each lower bisection level n, the trajectory consists of 2^n+1 points obtained from the maximal one (level s trajectory) as a subset of points $q[i\cdot 2^{s-n}]$ for $i = 0,1,\ldots,2^n$.

3. LOW-DIMENSIONAL QUASI-MC

In this Section we will first introduce low-discrepancy sequences and then apply quasi-Monte Carlo method for calculation of low-dimensional integrals of the Gaussian type. Since the SPEEDUP code requires calculation of high-dimensional integrals of the similar type, this preliminary study is done to verify the quasi-MC algorithm to be implemented in the QSPEEDUP code. It is also used to determine the appropriate distribution of deviations from the exact result on an ensemble of independent quasi-MC runs, which then could be instrumental in estimating errors of quasi-MC results. In addition to this, such study can be used to estimate the expected improvement in the performance of the SPEEDUP code when quasi-MC is implemented.

The usual implementation of the Monte Carlo method [22] uses pseudo-random numbers for calculation of the integrals. In the simplest case, if we are calculating d-dimensional integral on a unit cube U^d, and if ξ_i is a sequence of pseudo-random d-dimensional points in U^d, then the MC estimate of the integral of the function $f(\mathbf{x})$ is given by the average of the function f evaluated at the MC sample of points ξ_i. According to the central limit theorem, such estimate converges to the exact value of the integral when the number of MC samples N_{MC} goes to infinity. Furthermore, central limit theorem states that the statistical distribution of numerical results obtained using large number of independent MC samples is a Gaussian, centered at the exact value of the integral, with the variance $\sigma^2(f)/N_{\mathrm{MC}}$, where $\sigma^2(f)$ is given by the analytic formula

$$\sigma^2(f) = \int_{U^d} f^2(\mathbf{x}) d\mathbf{x} - \left(\int_{U^d} f(\mathbf{x}) d\mathbf{x} \right)^2 , \qquad (9)$$

and can be estimated as well from a single MC run. This gives clear statistical interpretation of errors when MC method is used: the distribution of deviations

$$\Delta = \int_{U^d} f(\mathbf{x})d\mathbf{x} - \frac{1}{N_{\mathrm{MC}}}\sum_{i=1}^{N_{MC}} f(\xi_i)\ , \qquad (10)$$

is a Gaussian with the expected standard deviation

$$E(\Delta) = \sqrt{\frac{\sigma^2(f)}{N_{\mathrm{MC}}}}\ . \qquad (11)$$

This shows the familiar convergence rate of $N_{\mathrm{MC}}^{-1/2}$ associated with MC pseudo-random methods, and its main advantage: we can decrease the deviation of numerical results (i.e. increase the accuracy of results) by simply increasing N_{MC}, the number of MC samples.

The key property of (pseudo-)random sequences is their uniformity, so that any contiguous subsequence is well spread throughout the domain of integration. This idea has lead to the suggestion that using other sequences, which are more uniformly distributed than a random sequence, may produce even better results. Such sequences are called quasi-random or low-discrepancy sequences [23].

Initially it may appear that a simple d-dimensional grid would provide optimal uniformity. However, grids suffer from several difficulties. First, the number of points required to create even a coarse mesh grows exponentially with the number of dimensions. Also, grids tend to have rather high discrepancy, a quantity measuring the deviation from the uniformity of a set of points. Finally, the size of the grid cannot be increased incrementally. The only obvious method for increasing the size of a uniform grid is to halve the mesh size, which requires addition of 2^d times the current number of points. Even such an exponential increase in the number of points would only yield a polynomial (depending on the discretization approach) increase in the accuracy.

Solution to this problem is to use infinite sequences of points such that for every N, the first N terms of a sequence are uniformly distributed throughout the cube. In order to quantify this, we introduce the discrepancy D_N of the sequence ξ_i of N points, defined as

$$D_N = \sup_{Q\in U^d}\left|\frac{\text{number of points in } Q}{N} - m(Q)\right|\ , \qquad (12)$$

where Q is any d-dimensional rectangle contained within U^d, with surfaces parallel to coordinate axes, and $m(Q)$ is its volume. By the law of the iterated logarithm [24], the expectation of the discrepancy of a random sequence is bounded by $(\log\log N)N^{-1/2}$.

There are many known quasi-random sequences (Halton [25], Sobol' [26], Faure [27], etc.) for which the discrepancy is bounded by $(\log N)^d/N$, which suggests greater uniformity than a (pseudo-)random sequence. In this paper we will use Sobol's sequence [28, 29] for implementation of the quasi-MC algorithm within the existing SPEEDUP code, as well as for a comparison with the earlier developed MC algorithm.

In order to verify the quasi-MC algorithm, which will be later used in the improved version of the SPEEDUP code, we have considered calculation of the Gaussian-type integrals

$$I = \int_{U^d} \exp\left(-\sum_{i=1}^{d} x_i^2\right)\ d\mathbf{x}\ , \qquad (13)$$

where d is the number of dimensions. In order to verify the expected approximate $1/N_{\mathrm{QMC}}$ scaling of deviations from the exact value of the integral, we have performed the numerical calculation using the large number of independent QMC samples for different values of dimensionality d. The first important observation is that the obtained distribution of numerical estimates for the value of the integral was always found to be a Gaussian, whose parameters can be found by fitting. The obtained distributions were centered on the exact values of integrals (13) within the errors estimated by the fitted widths of Gaussians. The typical deviations of quasi-MC results are shown in Fig. 1 as a function of the size of a quasi-random sample N_{QMC}. As can be seen from this log-log graph, the deviations are proportional to N_{QMC}^{-1}.

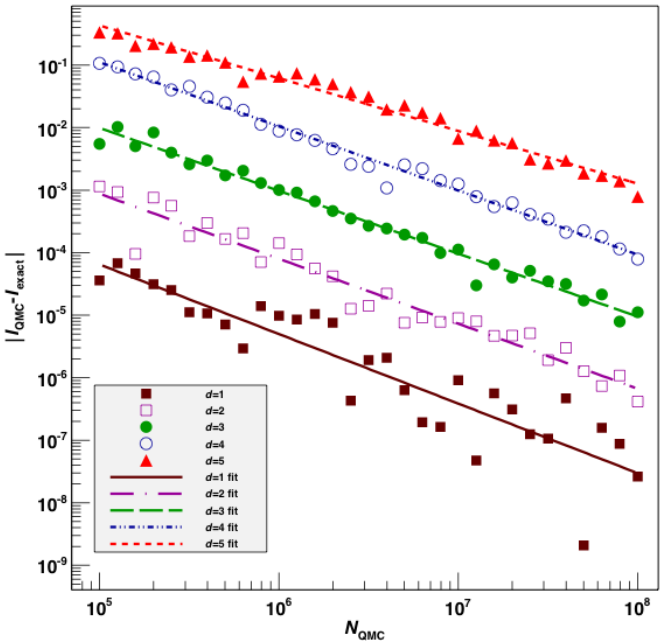

Fig. 1. Deviations of numerically calculated values of the integral (13) from the corresponding exact values, as a function of the number of quasi-random numbers N_{QMC} for different dimensionalities d of the integral. Sobol' set in the appropriate number of dimensions was used.

4. QUASI-MC IMPLEMENTATION OF THE SPEEDUP CODE

In this Section we present application of the quasi-MC method for calculation of quantum mechanical transition amplitudes, based on the modified version of the SPEEDUP code. We also study the statistical distribution of the obtained results on a large ensemble of samples, identify the appropriate estimate of deviations from the exact amplitudes and their dependence on the size of the sample, and assess the performance of the code. This will be done on a simple model of a quartic anharmonic oscillator, which however exhibits all features relevant for the proper assessment of the method and improved implementation of the algorithm.

QSPEEDUP, the modified version of the code, uses Sobol's set [28, 29] of low-discrepancy quasi-random numbers, instead of the pseudo-random numbers generated by the SPRNG library [21] in the original SPEEDUP code. In the previous Section we presented results for the case of low-dimensional integrals of the Gaussian type. This allowed us to determine that the distribution of numerical results obtained by the quasi-MC algorithm is also of the Gaussian type. Although calculation of general transition amplitudes assumes calculation of discretized path integrals of much higher dimensionality and of a more complex type,

depending on the potential, the dominant behavior is still mostly given by the Gaussian integrals stemming from the kinetic part of the Hamiltonian.

The only modification in the code was related to the use of quasi-random instead of pseudo-random numbers. The implementation of Sobol's sequence [28, 29] we used allows generation of quasi-random numbers in a large number of dimensions, which were used in the Box-Müller method to obtain the trajectories according to the bisection algorithm. QSPEEDUP code is tested on the example of calculation of the transition amplitudes $A(0,1;1)$ for the anharmonic potential

$$V(x) = \frac{1}{2} m\omega^2 x^2 + \frac{1}{24} g x^4 , \qquad (14)$$

for the values of parameters $m=1, g=1$, with the level $p=4$ effective action, and using the target bisection level $s=8$, corresponding to 255-dimensional integrals. For such a physical system we have first considered a distribution the ensemble of 10^3 independently calculated transition amplitudes, each obtained from the sample of $N_{QMC}=10^8$ trajectories. The distribution is shown in Fig. 2. As expected from the results of previous Section, the observed distribution of transition amplitudes is again Gaussian. If we fit these numerical results to a Gaussian function, for the mean value and the associated error we get the estimate $\left\langle A^{p=4}_{QMC}(0,1;1)\right\rangle = 0.187029267(3)$, while for the standard deviation the fitting gives $\sigma_{\left\langle A^{p=4}_{QMC}\right\rangle} = 5.6(2)\times 10^{-9}$. If the standard Monte Carlo method is used, for the same size of the sample we would get the standard deviation of around 3×10^{-6}, which is substantially higher than the standard deviation of quasi-MC results presented in Fig. 2. Since the generation of qusi-random numbers has very similar complexity to the generation of pseudo-random numbers using the SPRNG library, the obtained increase in the accuracy directly translates to the increase in the performance of the QSPEEDUP code.

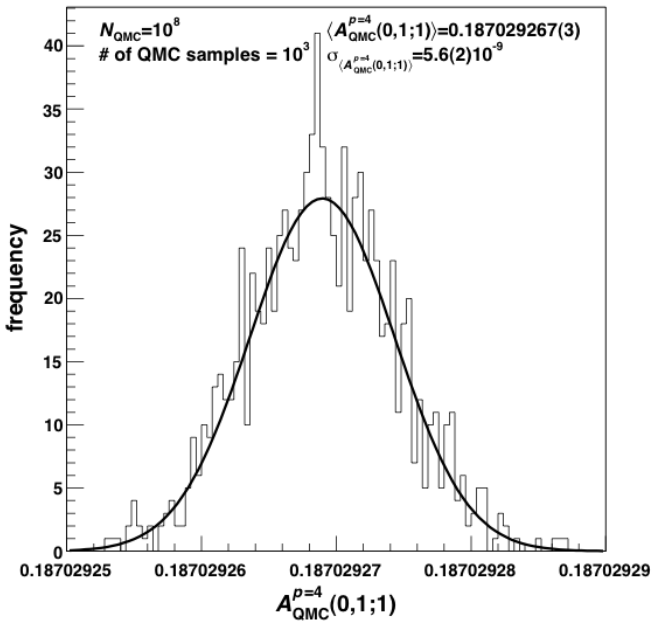

Fig. 2. Distribution of numerically calculated transition amplitudes for anharmonic potential (14) for the parameters given in the text using the QSPEEDUP code. Each amplitude is obtained using the sample of $N_{QMC}=10^8$. The histogram is obtained from the ensemble of 10^3 samples.

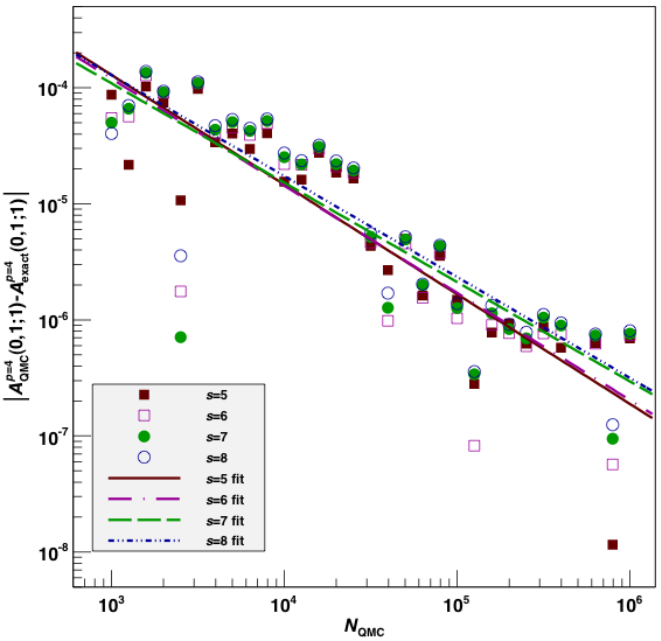

Fig. 3. Deviations of transition amplitudes calculated using the QSPEEDUP code from the exact value as a function of the quasi-MC sample N_{QMC}. The results are shown for different values of the target bisection level s, corresponding to the discretization with 2^s-1 time steps.

In order to assess if the obtained estimate for the amplitude (mean value of the Gaussian) is correct, i.e. if it is consistent with the exact value of the amplitude, we have used the MC SPEEDUP code with the exceedingly large number of samples $N_{MC}=10^{12}$. With such a sample we achieved the comparable precision for the amplitude, $A^{p=4}_{exact}=0.18702926(3)$, which is used as our estimate for the exact value in further calculation of deviations of numerical results obtained from the QSPEEDUP code. As we see, this value is in excellent agreement with the mean value of the distribution from Fig. 2, i.e. the deployed quasi-MC algorithm is found to give the correct value of the amplitude.

Since the standard deviation of quasi-MC results cannot be estimated using the MC approach, where one simply calculates the standard deviation of the sample according to equation (11), we have next studied the dependence of the deviation from the exact value of the amplitude as a function of the size of the sample N_{QMC}. Another approach would be to always study the distribution as in Fig. 2 and estimate the standard deviation from an ensemble of samples. However, this takes a considerable amount of time, which is not justified if there are other means to reliably estimate the deviation. Here we use the exact value of the amplitude obtained by the MC algorithm, and therefore simplify the numerical analysis considerably. The results are given in Fig. 3. As we see, the earlier observed approximate N_{QMC}^{-1} scaling of deviations is present for all values of the target bisection level.

Such scaling leads to the improved performance of quasi-MC algorithm compared to the standard MC method. As explained earlier, generation of pseudo-random and quasi-random numbers is of similar complexity, and therefore the fact that one needs much smaller size of quasi-MC sample in order to obtain the same accuracy as when MC algorithm is used presents a significant advantage. This is illustrated in Fig. 4, where we plot the speedup, i.e. the ratio of required

CPU time for the execution of MC and quasi-MC code in order to achieve the same deviation from the exact value of the amplitude. As we can see, even for a moderate value of the precision Δ, one obtains improvement of many orders of magnitude, approximately proportional to $1/\Delta$.

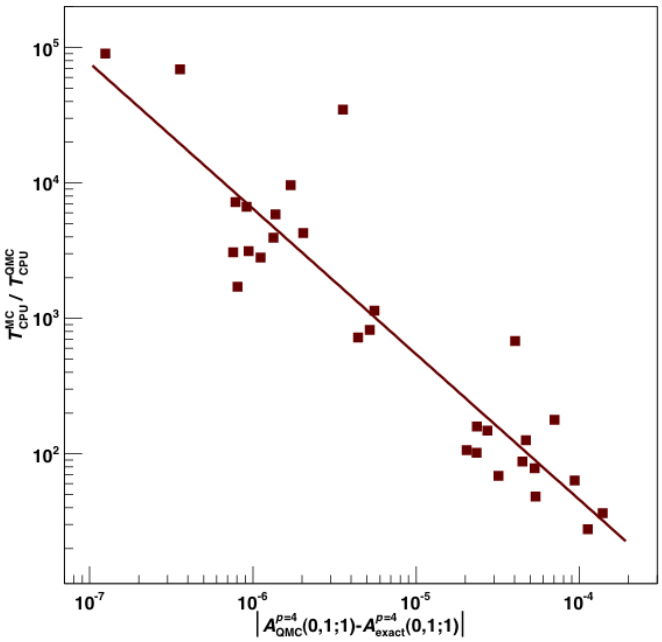

Fig. 4. The speedup in the calculation of transition amplitudes at the given precision using the QPEEDUP CODE, compared with the required CPU time for the calculation using MC algorithm implemented in the SPEEDUP code.

5. CONCLUSIONS

In the paper we have presented quasi-MC extension of the SPEEDUP code for calculation of quantum mechanical transition amplitudes using the effective action approach. The improved QSPEEDUP algorithm uses Sobol's set of quasi-random numbers for generation of trajectories relevant for calculation of transition amplitudes in the path integral formalism. At first, we studied the distribution of numerical results obtained using the quasi-MC algorithm for lower-dimensional integrals similar to the ones being calculated in the path integral code. The distribution is found to be Gaussian, with the standard deviation inversely proportional to the size of the sample. The same conclusions are obtained when quasi-MC QSPEEDUP code is applied to the calculation of transition amplitudes of a quartic anharmonic oscillator. The obtained $1/N_{QMC}$ scaling of deviations leads to a significant speedup of the quasi-MC algorithm compared to the standard MC approach for the same accuracy of results.

ACKNOWLEDGEMENTS

We gratefully acknowledge useful discussions with Emanouil Atanassov. This work was supported in part by the Ministry of Science of the Republic of Serbia, under project No. OI141035. Numerical simulations were run on the AEGIS e-Infrastructure, supported in part by FP7 projects EGEE-III and SEE-GRID-SCI.

REFERENCES

[1] A. Bogojević, A. Balaž, and A. Belić, Phys. Rev. Lett. **94**, 180403 (2005).
[2] A. Bogojević, A. Balaž, and A. Belić, Phys. Rev. B **72**, 064302 (2005).
[3] A. Bogojević, A. Balaž, and A. Belić, Phys. Lett. A **344**, 84 (2005).
[4] A. Bogojević , A. Balaž, and A. Belić, Phys. Rev. E **72**, 036128 (2005).
[5] A. Balaž, A. Bogojević, I. Vidanović, and A. Pelster, Phys. Rev. E **79**, 036701 (2009).
[6] I. Vidanović, A. Bogojević, and A. Belić, Phys. Rev. E **80**, 066705 (2009).
[7] I. Vidanović, A. Bogojević, A. Balaž, and A. Belić, Phys. Rev. E **80**, 066706 (2009).
[8] A. Balaž, I. Vidanović, A. Bogojević, and A. Pelster, Accepted for publication in Phys. Lett. A (2010); arXiv:1001.1463.
[9] J. Grujić, A. Bogojević, and A. Balaž, Phys. Lett. A **360**, 217 (2006).
[10] A. Bogojević, I. Vidanović, A. Balaž, and A. Belić, Phys. Lett. A **372**, 3341 (2008).
[11] R. P. Feynman and A. R. Hibbs, *Quantum Mechanics and Path Integrals* (McGraw-Hill, New York, 1965).
[12] H. Kleinert, *Path Integrals in Quantum Mechanics, Statistics, Polymer Physics, and Financial Markets*, 5th ed. (World Scientific, Singapore, 2009).
[13] D. M. Ceperley, Rev. Mod. Phys. **67**, 279 (1995).
[14] M. Takahashi and M. Imada, J. Phys. Soc. Jpn. **53**, 3765 (1984).
[15] X. P. Li and J. Q. Broughton, J. Chem. Phys. **86**, 5094 (1987).
[16] H. De Raedt and B. De Raedt, Phys. Rev. A **28**, 3575 (1983).
[17] SPEEDUP code, http://www.scl.rs/speedup/
[18] R. P. Feynman, Rev. Mod. Phys. **20**, 367 (1948).
[19] D. M. Ceperley, Rev. Mod. Phys. **67**, 279 (1995).
[20] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing* (Cambridge University Press, 1992)
[21] Scalable Parallel Random Number Generator library, http://sprng.fsu.edu/
[22] M. H. Kalos and P. A. Whitlock, *Monte Carlo Methods*, vol. 1: Basics (John Wiley and Sons, New York, 1986).
[23] E. I. Atanassov, I. T. Dimov, M. K. Durchova, LNCS **2907**, 128-135 (2004).
[24] A. Khinchine. Fundamenta Mathematica, **6**, 9-20, (1924).
[25] E. I. Atanassov, Mathematica Balkanica **18**, 15-32, (2004).
[26] I. M. Sobol', U.S.S.R Comput. Maths. Math. Phys. **7,** 86–112 (1967).
[27] H. Faure, J. Number Theory **42**, 47-56, (1992)
[28] B. Fox, ACM Trans. Math. Softw. **12,** 362-376 (1986)
[29] S. Joe, F. Kuo, ACM Trans. Math. Softw. **29**, 49-57, (2003).