

## GRID APPROACH TO PATH INTEGRAL MONTE CARLO CALCULATIONS

Danica Stojiljković, Antun Balaž, Aleksandar Bogojević, Aleksandar Belić  
Scientific Computing Laboratory, Institute of Physics, Belgrade, Serbia

**Abstract** – Approach taken for the gridification of the developed Monte Carlo code for calculation of path integrals is described. Brief introduction to path integrals and Grids is given, and details on the implementation of SPEEDUP in the Grid environment are described. The numerical results obtained by the gridified version of the application are shortly presented, demonstrating its usefulness in the research in physics and related areas.

## 1. INTRODUCTION TO PATH INTEGRALS

Feynman functional formalism is known to be the most concise and flexible formulation of quantum theories [1, 2]. It enabled us to easily extend quantization procedure to more complicated systems, from multidimensional many particle systems, to theory of fields, strings, etc. Many different areas of physics, such as high-energy physics, condensed matter, statistical and quantum physics, but also chemistry, material science, mathematics and even modern finance are treated in the same manner, using the same mathematical tool. This initiated an exchange of key ideas between diverse research areas. Its general mathematical framework gives simple and natural setup for using and applying symmetries, deriving various approximation techniques and non-perturbative results and making connections between different theories.

Although the number of exactly solvable models was not enlarged by the introduction of functional formalism, the analytical and numerical approaches to path integrals gave us possibility to treat large number of systems that were not previously accessible. Path integral is the main mathematical object in the functional formalism and it represents an infinite limit of multiple integrals. Most often use of approach for calculation of path integrals is Monte Carlo (MC) method [3], which is defined in quite general terms as any algorithm that uses random numbers for solving numerical problems. Advance gained by using MC simulations for calculating multiple integrals is that the standard deviation of results vanishes as  $1/\sqrt{N_{MC}}$  regardless on integral dimensionality.

$N_{MC}$  here represents the number of Monte Carlo samples and it is clear that by increasing this number we can improve the precision of the results as desired.

As we already said, path integral is defined as an  $N \rightarrow \infty$  limit of an  $N$ -fold integral. Since we can numerically calculate only finite number of integrals, here in contrast to calculating ordinary  $N$ -fold integrals an additional source of error is present. Naive  $N$ -discretized value of the path integral differs from its continuum limit by a term of the order  $1/N$ . A recent series of papers [4, 5, 6, and 7] has focused on the dynamical implications of stochastic self-similarity by studying the relation between path integral discretizations of different coarseness. This has resulted in a systematic analytical construction of a hierarchy of  $N$ -fold discretized effective actions  $S_N^{(p)}$  labeled by an integer number  $p$  and built up from the naively discretized action in the mid-point

ordering prescription (corresponding to  $p=1$ ). The level  $p$  effective actions lead to discretized transition amplitudes expressed through path integrals and expectation values differing from the continuum limit by a term of the order  $1/N^p$ . The MC code implementing this approach can be found at [8].

## 2. INTRODUCTION TO GRIDS

The above described effective action approach confirms to the common experience that any analytical knowledge of the considered system can be translated into vast speedups in the convergence of numerical approaches used to tackle those complex systems. However, path integrals require substantial computing resources, even in such dramatically improved algorithms. In MC approach this means that we still need to use large numbers of MC samples  $N_{MC}$ . This is usually not possible on a single parallel cluster in one research institute, or is very inefficient to wait until enough resources are available in one's home institute. Therefore, the use of computing resources distributed across different institutions collaborating on joint RRD projects is the most efficient way of obtaining considerable computing resources on demand. In fact, this way each institution involved can obtain when needed computing resources much greater than it possesses, while other institutions can use its resources when idle. Resource sharing can be organized on reciprocity principle, on a commercial basis, or on some other suitable arrangement.

While the Internet provides a framework service for sharing the information, a framework for sharing available computing and data resources is known as Grid. It consists of a number of clusters that provides computing power and storage resources, and that are connected and enabled to interoperate through a series of Grid services. Here we will shortly describe Grid established by the EGEE-II [9] and SEE-GRID-2 [10] projects, as well as by the AEGIS (Academic and Educational Grid Initiative of Serbia [11]). The services and resources offered to interested users represent an ideal framework in which MC simulations can be executed, taking advantage of any idle resources for improving statistics of path integral calculations. Basic Grid operations offered to a user are job submission management, file transfer, database access, database management and monitoring and indexing system information. To allow these operations, core services Grid-nodes collect information published by the local information systems on each cluster. Grid user does not need to know where his/her job is executed, or where the data is stored. The user does not need to be familiar with the architecture and organization of Grid services. He or she just needs to log on to a node with installed user interface (UI) software, describe job requirements in a simple file and accompany it with necessary programs and input files, and submit it. Later, the user can monitor the job and retrieve output files when the

job is done. When job is submitted, unique job ID is generated which must be saved by the user in order to retrieve output or monitor job status at any time. Grid services allow UI to find available clusters that fulfill job requirements specified by user, and distribute it to one of those clusters. They monitor status of a job, and collect output files after the job is done, so the user could retrieve it later. If large amounts of data are used or produced by a program, data can be placed on a storage element. That way we can avoid unnecessary transfers of data and reduce possibility of problems that can occur due to "overloaded" network connections of Grid nodes.

In order to ensure secure operations on the Grid, it was necessary to introduce authentication and authorization mechanisms. Authentication of users and services is done through digital certificates based on Public Key Infrastructure (PKI). The certificates are issued through a network and national and regional certification authorities. The authorization for the use of resources and other operations is done through Virtual Organizations (VO). One needs to become a member of some VO which represents group of people working together on the same activity/project. One can be a member of more than one VO. What can one do on a Grid is determined by his/her VO privileges meaning that one can access just the resources on clusters supporting his/her VO, and with the fair-share of those resources specified by the local policies. Here the user does not need to adjust his/her behavior according to those policies, but rather the policies will be automatically implemented and imposed by Grid services.

At the time of submission of a job and during its execution, one needs to have valid credentials, *mini-certificate* (proxy) which has no password, propagates through Grid and has a significantly limited life time, specified at creation time. If job is not finished by the proxy expiration time, it would be aborted, unless the proxy is renewed. This should be kept in mind when running long jobs.

### 3. GRID APPROACH TO MC SIMULATIONS

Running one or small number of long running jobs on Grid would not be its optimal use. There is a large number of processors available and if we want to get our results faster it is necessary to divide all computational tasks to a large number of shorter programs which can be executed at the same time. This can always be done for MC simulations, but for some types of problems it may not be possible. There are three ways in which we can perform this division of tasks: parallelization, gridification, or combination of both.

Parallel program on a usual PC-cluster consists of cooperative processes, each with its own memory, which can exchange data. Communication and distribution of tasks to different processors is implemented in a program code by using libraries of functions and macros provided by implementation of e.g. Message Passing Interface (MPI). Parallelization of MC simulations is trivial. Each node is processing same number of MC samples independently, and after processing is done master node collects the raw data, performs final calculations and saves the results in a file. For running parallel program on Grid site it must have support for MPI, which is not standard configuration..

Gridification of program assumes separation of calculation tasks into a number of shorter independent programs. In MC simulations this division is performed naturally. Instead of processing, for example,  $10^9$  samples on a single node, task can be divided to 1000 nodes, each processing just  $10^6$  samples. Each process is saving raw data and when all data is collected, final processing can be done via script or some other results processing program. This is usually performed on desktop computer or user interface. This approach is based on independence of individual programs. If there is a malfunction on one node, we loose only a small amount of data, while in parallel approach this leads to breaking of the whole program and loss of all data. In addition, we can submit as many jobs as needed for statistics requested, thus replacing failed jobs with the new ones seamlessly. Separate programs can run on several sites, at different times, which enables us to divide the task into thousands of smaller programs, while parallel approach demands that all the nodes are available on one site at the same time.

Before submitting a job to the Grid it must be specified in a file written in the Job Description Language (JDL file). This file contains specifications on the type of job, executable file and its arguments, location of input files required by the job (input sandbox), names of output files that user wants to retrieve (output sandbox), many optional requirements for site properties like number of working nodes, MPI support, physical closeness to some Storage Element, specific site name, maximal allowed expected site response time for this job, etc. Executable file can be a compiled program or a script that contains sequence of commands that needs to be executed on a worker node (WN). That way code can be compiled on the worker node itself, at the execution time user can manipulate data directly from WN, pack or unpack files, transfer files from and to storage elements and execute any command according to his/her VO privileges.

There are few different middleware systems for managing jobs on grid such as gLite, LCG, Globus, UNICORE, etc. The Grids established by AEGIS, EGEE-II and SEE-GRID-2 are based on LCG [12] and gLite [13], but concept on the user side is basically the same. User submits job from command line of user interface and gets unique job ID. Sandbox and JDL files are sent to a series of Grid services which then search for available resources that meet job requirements. The job is then being dispatched to one of those sites and scheduled in the local site's queue. After job is done, files defined to be in the output sandbox are sent back to grid services (the actual component used is the Workload Management, WMS) which are waiting for the user to retrieve the output. This is also done from the command line of the user interface. User can also monitor status of submitted job at any time after submission.

For submission, monitoring and retrieving the output of large number of MC simulations for calculation of path integrals, we developed a number of scripts. Scripts for job submission use the concept of a template JDL file (TJDL) for creating a large number of JDL files which differs only in few program arguments. Jobs are then being submitted one by one, and all job IDs are stored in a file. Scripts for monitoring of job statuses and retrieving data or canceling the jobs are using this file. The scripts can be downloaded from [8].

This concept can also be used for other applications, and scripts need to be adapted for their use for particular problems. Submission of hundreds of jobs can take several hours if they are submitted one by one. New component of grid services - WMProxy in gLite - is developed recently to handle a large number of job submissions efficiently. New type of job - parametric job - is introduced. It represents a set of very similar jobs, differing only in the values of parameters, defined by some of JDL attributes. With this type of jobs, user just needs to send one JDL file with the parameter changing range specified. This new type of job then creates final JDL files and submits them to Grid on behalf of the user. Another advantage offered is that jobs can share input sandboxes, so when many single jobs are using the same input files, those can be transferred only once from UI. Disadvantage of this type of job is that, for now, there is only one parameter that can be changed.

#### 4. PRELIMINARY RESULTS AND PERFORMANCE

We are using path integrals for the calculation of quantum mechanics probability amplitudes. Here we will present some numerical results obtained using the gridified version of the SPEEDUP code [8], using the scripts described in the previous section. The evolution of a physical system during the time  $T$  can be expressed as a path integral of the quantity  $e^{-S}$ , where  $S$  is the action of the system. The integration is made over all possible trajectories of the system (thus making the infinite number of ordinary integrals necessary), and the action reduces to the energy of the system in the Euclidean time approach.

Definition of path integrals makes it necessary to make transition from continuum to the discretized theory. Time of the evolution is divided in  $N$  equal time steps, and instead of using continuum value of the action  $S$  for each trajectory, we are using its discretized value  $S_N$ . Actions  $S_N$  are not uniquely defined, and the value of the discretized path integral can approach continuum limit in different ways. Although the continuum limit of different discretizations will remain the same, the convergence properties can differ.

We recently developed a series of effective actions  $S_N^{(p)}$  that speeds up the convergence [4, 5]. They have been applied here for calculation of quantum mechanics transition amplitudes, energy levels [6] and expectation values [7], and statistical partition functions. It is analytically shown in those papers that transition amplitudes approach their continuum values as  $1/N^p$ . Expressions for  $S_N^{(p)}$  are derived for  $p$  up to 10, but the developed procedure enables us to find these actions for arbitrary value of  $p$ . This allows us to use smaller values of  $N$  for evaluation of path integrals and we gain speedup of several orders of magnitude. But there is a price to be paid. Numerical complexities of expressions for effective actions have exponential growth with  $p$ , and implementation of this analytical speedup will increase the required computational time. This growth is shown in Figure 1 for  $p$  up to 9. We can see that for  $p=9$  the simulation is about 10 times slower, which in fact is small price to pay for gain in precision of nine orders of magnitude.

In statistical mechanics path integrals can be applied for calculating values of partition function and thermal free energy. MC simulations for calculations of the free energy numerically confirmed that the use of effective actions leads to the same speed up in convergence [6]. This is illustrated in

Figure 2 where dependence of discretized free energy values (calculated using the corresponding discretized effective actions) on the number of discretization points  $N$  is shown for levels  $p$  from 1 to 5. We see that for  $p=5$  discretized values differ from their continuum limit by a term which is smaller than the stochastic error introduced by MC method. In order to further gain something from the speedup offered by higher  $p$  levels we need to reduce stochastic error by increasing number of MC samples. This is why we need to use distributed computing resources that allow large statistics to be obtained in a short time periods.

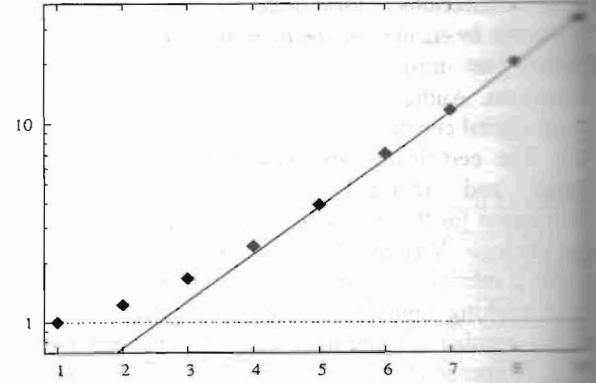


Fig. 1: Relative increase in computation time that comes about from the increased complexity of expression for higher  $p$  levels effective action

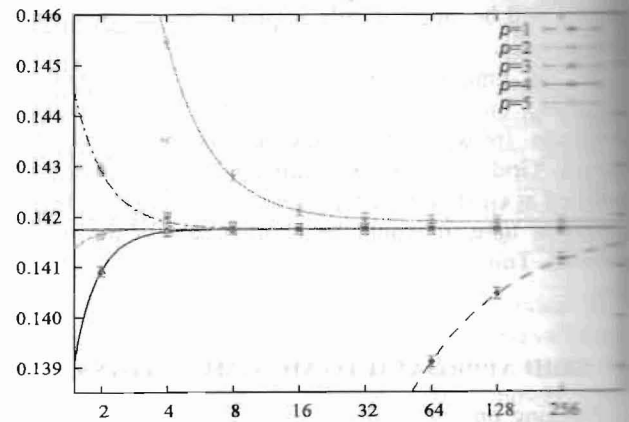


Fig. 2: The dependence of free energy  $F_N^{(p)}$  on  $N$  for different levels of  $p$ . The plot is for the anharmonic oscillator with quartic coupling  $g=1$ , inverse temperature  $\beta=1$  and  $N_{MC}=10^7$  MC samples. The same kind of behavior is seen for other parameters, as well as for other potentials.

Free energy of the physical system can be expressed in terms of its energy levels

$$e^{-\beta F(\beta)} = \sum_{n=0}^{\infty} e^{-\beta E_n}, \quad (1)$$

where  $\beta$  is inverse temperature. As we can see, the free energy is completely determined by system's energy spectrum and vice-versa. By calculating free energy on a set of temperatures, it is possible to extract several lowest energy levels. Calculations are conducted for several different models. In case of models that are exactly solvable, our results agree with exact values, but our approach provides means for calculation of spectra for models that can not be solved analytically, and are not in the perturbative regime.



i.e. which can not be treated by the usual methods. As an illustration we give energy levels of anharmonic oscillator with the potential of the form:

$$V(q) = \frac{1}{2}q^2 + \frac{g}{4!}q^4 \quad (2)$$

Several values of coupling constant  $g$  are considered, and results are given in Table 1. Note that the case  $g=0$  is exactly solvable, while  $g=1000$  lies deep in non-perturbative domain.

Table 1: Low lying energy levels of the anharmonic oscillator with quartic coupling  $g$ , calculated using  $N=256$ ,  $p=9$ , and  $N_{MC}=10^7$

$g$	$E_0$	$E_1$	$E_2$	$E_3$
0	0.49993(2)	1.502(2)	2.48(6)	3.6(5)
0.1	0.50301(2)	1.516(1)	2.54(5)	3.5(2)
1	0.52765(2)	1.6295(8)	2.85(2)	3.98(7)
10	0.67335(2)	2.230(1)	4.12(2)	
100	1.16247(4)	4.058(6)		
1000	2.3578(2)			

## 5. CONCLUSIONS

We presented the Grid approach that can be applied for large scale computations if some generic Monte Carlo simulations are used. On the important SPEEDUP application we demonstrated how an MC application can be gridified and efficiently deployed in production Grid environments, such as e-Infrastructure offered by the EGEE-II of SEE-GRID-2 projects, or the national AEGIS infrastructure. The usefulness of this approach is demonstrated through solving numerically complex problem of energy spectra calculation of different quantum mechanical systems.

## 6. ACKNOWLEDGMENTS

This work was supported in part by the Ministry of Science and Environmental Protection of the Republic of Serbia under project no. OI141035. D. Stojiljković has received support under EC FP6 project CX-CMCS (INCO-026343). The presented numerical results were obtained on the AEGIS GRID e-infrastructure whose operation is supported in part by EC FP6 projects EEE-II (INFSO-RI-031688) and SEE-GRID-2 (INFSO-RI-031775).

## 7. REFERENCES

- [1] R.P. Feynman and A.R. Hibbs, Quantum mechanics and Path Integrals, New York: McGraw-Hill, 1972.
- [2] R. P. Feynman, Statistical Mechanics, New York: Benjamin, 1972
- [3] M. H. Kalos and P. A. Whitlock, Monte Carlo Methods, vol. 1: Basics, New York: John Wiley and Sons, 1986
- [4] A. Bogojević, A. Balaž and A. Belić, Phys. Rev. Lett. 94 (2005) 180403
- [5] A. Bogojević, A. Balaž and A. Belić, Phys. Rev. B 72 (2005) 064302
- [6] D. Stojiljković, A. Bogojević, A. Balaž, Efficient Calculation of Energy Spectra Using Path Integrals, Phys. Lett. A (2006), In press, doi:10.1016/j.physleta.2006.08.035
- [7] J. Grujić, A. Bogojević, A. Balaž, Energy Estimators and Calculation of Energy Expectation Values in the Path Integral Formalism, Phys. Lett. A (2006), In press, doi:10.1016/j.physleta.2006.08.044
- [8] <http://scl.phy.bg.ac.yu/speedup>
- [9] <http://www.eu-egge.org>
- [10] <http://www.see-grid.eu>
- [11] <http://aegis.phy.bg.ac.yu>
- [12] <http://lcg.web.cern.ch/LCG>
- [13] <http://glite.web.cern.ch>

**Садржај** – Описан је приступ гридификацији развијеног Монте Карло алгоритма за рачунање функционалних интеграла. Дат је кратак увод у функционалне интеграле и основни концепт Грид-а, и описани су детаљи имплементације "SPEEDUP" апликације у Грид окружење. Представљени су нумерички резултати добијени помоћу гридификоване верзије апликације који илуструју њихову корисност за истраживања у физици и сродним областима.

## ГРИД ПРИСТУП МОНТЕ КАРЛО РАЧУНАЊУ ФУНКЦИОНАЛНИХ ИНТЕГРАЛА

Даница Стојиљковић, Антун Балаж,  
Александар Богојевић, Александар Белић