Probabilistic Physical Modeling on Distributed Computing Systems

Andrei Smirnov, Steven Rowan, and James McCormick
West Virginia University
Morgantown, WV

Abstract

A method is proposed which facilitates express analysis of the results of large distributed simulations of physical systems. The method uses the domain decomposition technique based on the idea of transition probabilities. The proposed approach opens the possibility of conducting parallel simulations of strongly coupled systems on loosely coupled distributed computer platforms, such as grid computing environments. The results of the simulations can then be retrieved in a timely manner for express risk analysis. The case of aerosol dispersion in urban environments was considered, and a prototype solver was developed and tested. Fast response of the system is achieved by replacing complex 3D simulations with information retrieval from aerosol dispersion database. The database was created by extensive simulations of different aerosol dispersion scenarios in a grid computing environment. The reduction of system response from a few days to several seconds can be achieved compared to real-time 3D simulations.

Keywords: Monte Carlo methods and applications, Parallel and distributed simulation, Domain decomposition methods Fast Response Systems, Risk analysis, Information Retrieval, Aerosol dispersion modeling, Urban environment modeling

1 Introduction

The emerging grid computing technologies offer a great promise for a number of applications, such as distributed databases and transaction processing. However in the field of scientific computing these system often present formidable obstacles, since they are essentially based on loosely coupled collection of computing nodes. The lack of adequate communication band-width prevents their effective usage in solving tightly coupled problems, such as those based on various continuum approximations and partial differential equations (PDE).

It is well recognized that parallel or distributed computing is not well suited for many types of problems, especially in the field of high-performance (HP) computing, because of tight coupling or procedural bottlenecks in the solution methods used.

There are two main issues involved in distributed simulations of aerosol dispersion in an urban environments: (1) tightly coupled nature of the problem does not fit with the loosely coupled parallelism of the grid, and (2) the simulations need to be performed in a timely manner, which is hard to accomplish given the physical complexity of the problem.

In this both issues have been solved using the idea of *domain transition probabilities* (DTP) and *probabilistic implicit tracking* (PIT) technique [1]. The approach can facilitate the solutions to some of the tightly coupled problems in grid computing environments. The technique was applied to the problem of express analysis of aerosols dispersion in a city landscape.

2 Method

In the method of this study a complex space of objects such as a city landscape is subdivided into domains in a similar manner as in a conventional domain decomposition technique [2]. However, instead of using a tightly coupled scheme, the domain is completely decoupled, using the concept of *domain transition probabilities* (DTPs). Each DTP represents the probability for an event A occur-

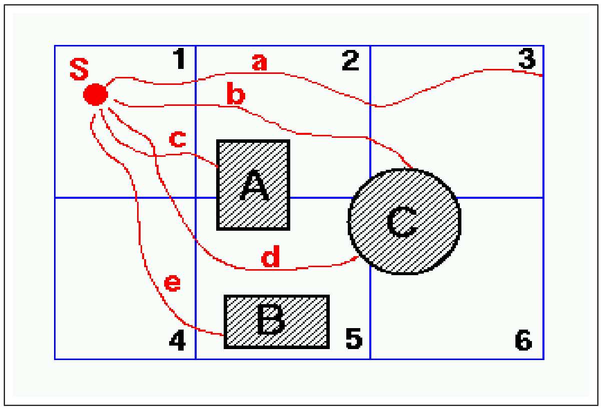


Figure 1: Particle passage through a decomposed domain: S - particle source, A...C - physical objects where the particle fallout occurs, a...e - particle trajectories, 1..6 - domains of a decomposed computational space.

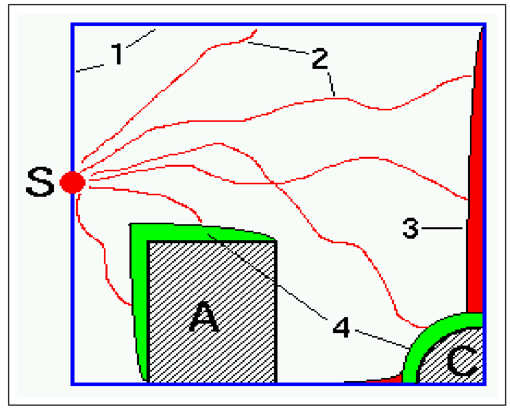


Figure 2: Assembling domain transfer probabilities: S - boundary particle source, A-C - objects, 1 - domain boundaries, 2 - particle trajectories, 3 - source-to-boundary transfer probability, 4 - source-to-object transfer probabilities.

ring at the boundary of a domain to cause an event B at some other point on the boundary.

To illustrate the idea, figure 1 shows the conventional way of tracking the particle through a decomposed multi-domain space. In contrast, the current approach is conducted in two stages. First the information on DTPs is collected for each domain in prior simulations (Fig.2), and then a probabilistic procedure of particle tracking is applied, which we call *probabilistic implicit tracking* (PIT), using the obtained DTP information (Fig.3).

In the first stage of physical modeling the DTP data is assembled into two sets: the external transition probabilities, which represent boundary-to-boundary transition events, and internal transition probabilities for the events of particle fallout on the objects inside the domain. It should be noted that by sub-dividing each domain further into objects it is possible to reduce the overall size of the DTP data set. This is because this representation uses point-to-object rather than point-to-point transfer probabilities, and the number of objects in the domain is smaller than the number of all possible discrete locations.

In this study we used a simplified aerosol transport model based on the equation of particle motion, expressed in terms of particle velocity, $\mathbf{v}(\mathbf{x}, t)$, in a given mean flow field, $\mathbf{u}(\mathbf{x}, t)$:

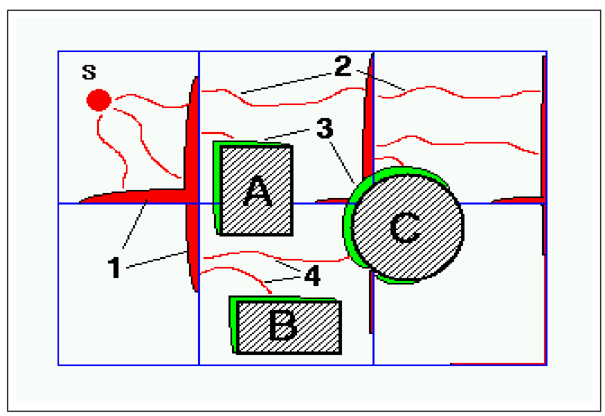


Figure 3: Implicit particle tracking using DTPs: 1 - boundary particle sources, 2 - boundary-to-boundary transfer probabilities, 3 - object sinks, 4 - boundary-to-object transfer probabilities,

$$\frac{d\mathbf{v}}{dt} = C_D(\mathbf{u} - \mathbf{v}) + C_T\mathbf{u}' - \mathbf{g} \qquad (1)$$

where C_D is the drag coefficient, C_T the turbulent diffusivity, $\mathbf{u}'$ is the instantaneous turbulent fluctuation vector, and $\mathbf{g}$ is the gravity acceleration vector [3]. The position of the particle at each time step is computed using second order Runge-Kutta time-stepping scheme:

$$\mathbf{x} = \mathbf{x}_0 + \frac{\mathbf{v} + \mathbf{v}_0}{2} dt$$

where $(\mathbf{x}_0, \mathbf{v}_0)$ and $(\mathbf{x}, \mathbf{v})$ are the old and new coordinates and velocities of a particle. The effects of turbulent dispersion on the particles encapsulated in the second term of (1) which was modeled using the RFG technique [4].

Each aerosol particle is convected in a velocity field $\mathbf{u}$, and is traced inside the computational domain until it crosses the domain boundary or hits an object inside the domain. In the second event the hit count for that object is incremented. The final DTP is obtained by dividing all the hit counts by the number of particles released.

The number of particles released for each realization will determine the accuracy of predictions, since the variance of the result, σ, is inversely proportional to the square root of the number of particles $n(x)$ deposited at position x: $\sigma(x) \approx 1/\sqrt{n(x)}$. Figure 4 shows a typical probability distribution of aerosol fallout on a particular object as a function of angle and one of the three spatial positions (height).

The storage of the DTP data set is an important factor since it is presumed that the database should be stored locally or accessed remotely. The size of the data set can be reduced significantly considering that the data on angular distribution are usually sparse, that is, many wind directions carry particles away from the objects causing zero deposition counts. It can be seen from Fig.4 that the size of the data set can be reduced by not storing the zero counts or very low probability counts. On the average, by avoiding zero counts a reduction by about a factor of 4 is possible. Another size reduction can be achieved by using data compression. Standard data compression methods, such as LZW, LZ77, and others can easily reduce the size of the file by a factor of 3 or more.

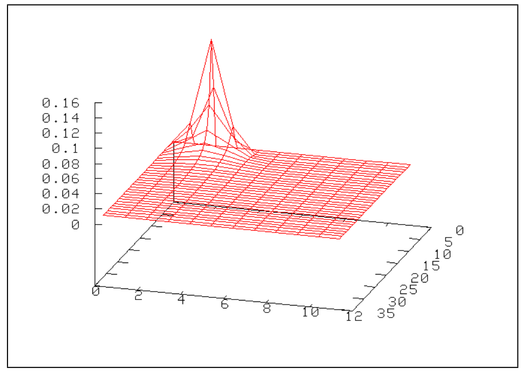


Figure 4: A typical DTP for different space locations and angles

3 Results and Discussion

To validate the method a generic city landscape was set up and prototyped after the Pittsburgh downtown area (Fig.5), using the voxel-based 3D graphics system [5, 6]. The whole domain was discretized on the grid with spacial resolution of $92 \times 92 \times 32$, which corresponds to the total of $N_n = N_x N_y N_z = 270848$ nodes or possible particle release points. The landscape was populated with characteristic features like rivers, hills, bridges, park area, pavements and buildings.(Fig.5).

Three sets of simulations were conducted: (1) parallel runs on a cluster using Lagrangian particle solver (LPD) to collect the DTP data; (2) probabilistic implicit tracking (PIT) using the DTP data, and (3) particle tracking using a conventional scheme. The last two simulations were conducted on a 1.8 GHz Pentium 4 laptop with a 1 Gig of RAM. All three simulators were implemented in a C++ language.

In the parallel simulations the whole scene was subdivided into 16 domains and the runs were conducted on a computer cluster with 4GB, 2GHz computing nodes (teragrid.org). One sub-domain was assigned per each node. Three simulations were performed for different number of particles released: $10^3, 10^4$, and 10^5. Thus, the total of 3 runs were conducted. The processor time required per each run was in an almost linear relation to the number of particles, and for the 10^5 processor run the average time for executing the DTP calculations on a single node

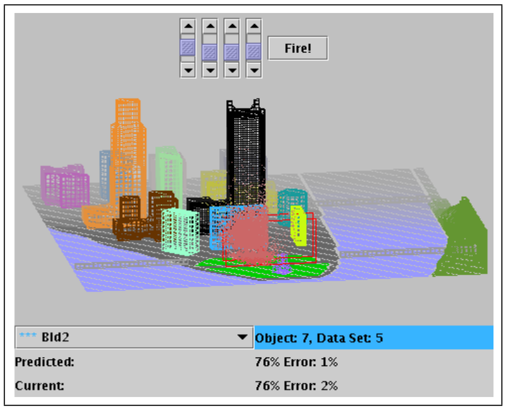


Figure 5: Web interface to simulate aerosol release in a city

http://mulphys.com/sim/demo

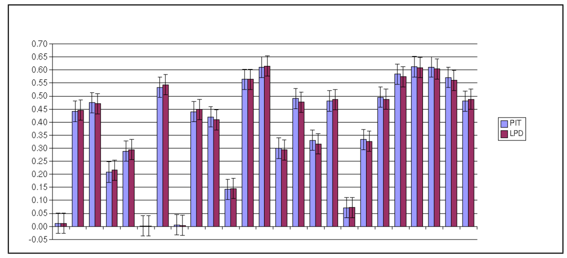
of the cluster was almost linear in the number of particles (Tab.1).

N_p	10^2	10^3	10^4	10^5
CPU [min]	3	27	266	2695

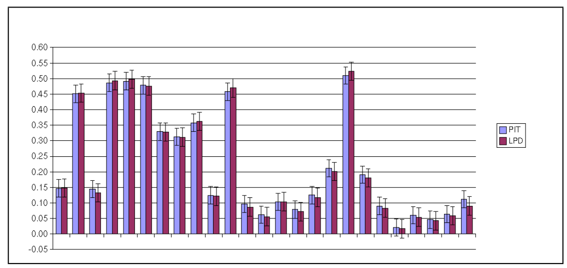
Table 1: Wall clock time of parallel runs for different number of particles.

The execution time of the stochastic algorithm on the laptop and was almost linearly proportional to the number of particles. It should be noted that the overall accuracy of simulations depends on the number of particles used in both parallel and stochastic simulations. The number of particles used in stochastic simulations can be selected so as to achieve a reasonable response time. Once this number is set, the time of stochastic simulations will not significantly depend on the number of particles used to produce the DTP data. This is because the number of particles engaged in aerosol simulations affect the accuracy of probability distributions, but not the size of probability data sets. Thus, the number of particles for stochastic simulations was set to 10^4, which provided a 20 sec response time on the average for DTP sets produced with any number of particles.

Figure 6 shows typical comparison histograms between



(a) Scenario 1



(b) Scenario 2

Figure 6: Comparison of aerosol deposition data computed with LPD and PIT methods

LPD and PIT methods obtained for two random aerosol release scenarios. The number of scenarios can be very large. The data collected so far for different scenarios show a very good agreement between the LPD and PIT methods with the average deviation of the results typically within 5% for the 10^5 particle case. However this deviation has a statistical character and may vary dependent on the number of particles deposited on each particular object.

To demonstrate the viability of this method for express risk analysis a prototype web interface was developed, which enabled us to test different aerosol release and dispersion scenarios (mulphys.com/sim/demo). The interface was written in Java language and provides a 3D representation of a city landscape with the possibility of navigating through the landscape, arbitrary positioning of the aerosol source, and setting wind direction (Fig.5). The applet also performs a real time simulation of aerosol propagation and dispersion for a limited number of particles as well as a web-retrieval the particle deposition data from a remote database.

The simulation in the applet is performed with a limited number of particles, and serves the purpose of comparing its execution time with a fast retrieval of the pre-assembled simulation data. The accuracy of the applet results is also less than that obtained from the data, since the latter were performed with a much higher number of particles.

4 Conclusions and Future Work

The method presented in this study serves dual purpose of (1) providing an *embarrassingly parallel* implementation of certain class of transport problems on grid computing environments and (2) facilitating express risk analysis of multiple scenarios of a complex physical event.

Using the idea of domain transition probabilities (DTP) and implicit probabilistic tracking (PIT) it was possible to replace a complex physical simulator with a simpler and more flexible stochastic simulator for the purpose of express analysis of simulation data.

A particularly relevant problem is the express analysis of possible aerosol contamination in the city. The results show that this method provides a viable and efficient tool for fast analysis of different contamination scenarios.

A significant saving of retrieval time and data space was achieved by querying objects rather than particular space locations for fallout data. If a more differentiated approach is needed, this approximation can easily be refined by splitting large objects into smaller ones, like buildings can be split into floors, etc.

The method is especially effective when implemented in multi-processor and distributed computing environments, since no inter-processor communication is required to produce the DTP data sets.

Acknowledgments

The authors would like to acknowledge West Virginia University Research Corporation for sponsoring this work.

References

[1] A. Smirnov and I. Celik. A Lagrangian particle dynamics model with an implicit four-way coupling scheme. In *The 2000 ASME International Mechanical Engineering Congress and Exposition. Fluids Engineering Division*, volume FED-253, pages 93–100, Orlando, Fl, 2000.

[2] B. Smith, P. Bjørstad, and W. Gropp. *Domain Decomposition, Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press, 1996.

[3] C. Crowe, M. Sommerfeld, and Y. Tsuji. *Multiphase Flows with Droplets and Particles*. CRC Press, 1998.

[4] A. Smirnov, S. Shi, and I. Celik. Random flow generation technique for large eddy simulations and particle-dynamics modeling. *Trans. ASME. Journal of Fluids Engineering*, 123:359–371, 2001.

[5] A.V. Smirnov, H. Zhang, and B. Sowers. Voxel-based volume graphics system for multi-physics modeling. In *The 8th World Multi-Conference on Systemics, Cybernetics and Informatics*, volume 5 of *Computer Science and Engineering*, pages 144–149. International Institute of Informatics and Systemics, 2004.

[6] A.V. Smirnov, H. Zhang, A. Burt, and I. Celik. Fuel-cell simulator interface. *Journal of Power Sources*, 138/1-2:187–193, 2004.