INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2000; **00**:1–6 *Prepared using* fldauth.cls *[Version: 2002/09/18 v1.01]*

Aerosols Dispersion Modeling Using Probabilistic Particle Tracking

Andrei Smirnov*, Steven Rowan, James McCormick

West Virginia University
Morgantown, WV 26506-6106

SUMMARY

A method is proposed which can facilitate parallel computations of particle transport in complex environments, such as urban landscapes. A two stage-approach is used, where in the first stage, physical simulations of various aerosol release scenarios are conducted on a high-performance distributed computing facility, such as a Beowulf cluster or a computing grid, and stored in a database as a set of transfer probabilities. In this stage the method provides a partially decoupled parallel implementation of a tightly coupled physical system. In the second stage,various aerosol release scenarios can be analyzed in a timely manner, using obtained probability distributions and a simpler stochastic simulator, which can be executed on a commodity computer, such as a workstation or a laptop. Using the proposed approach and developed graphical tools, a case of aerosol dispersion in a typical urban landscape has been studied. A considerable speedup of analysis time for different aerosol dispersion scenarios has been demonstrated. The method is appropriate for the development of express risk analysis systems. Copyright © 2000 John Wiley & Sons, Ltd.

KEY WORDS: Parallel simulations; Stochastic modeling; CFD; Domain decomposition; Risk analysis; Aerosols; Urban environments; Monte Carlo methods

NOMENCLATURE

CFD	Computational Fluid Dynamics
LPD	Lagrangian Particle Dyanmics
MPI	Message Passing Interface
NS	Navier Stokes
PIT	Probabilistic Implicit Tracking

*Correspondence to: andrei.v.smirnov@gmail.com

Contract/grant sponsor: West Virginia University Research Corporation; contract/grant number: PSCoR-2005

Received 3 July 1999
Revised 18 September 2002

1. BACKGROUND

The problem of urban aerosols closely relates to environmental issues, and has also received a special attention in the United States following the terrorist attacks on September 11, 2001. The U.S. Department of Homeland Security has sponsored planning scenarios in order to help prepare municipalities for the possibility of an event related to the release of biological, chemical, or radioactive material in urban areas. One approach is to conduct computer simulations of these scenarios using computational fluid dynamics (CFD) [[10]]. While these simulations can provide accurate results, the time required to run a simulation of a specific incident might very well take from several hours to several days to complete, depending upon the size of the simulation grid and the computing resources at hand. Considering the time critical nature of the need to respond to these incidents, it is all too likely that the information necessary for first responders to carry out their jobs would not be available until after the fact.

The problem becomes even harder when several physical models are involved in simulations. One common example is a combined CFD/ LPD (Lagrangian particle dynamics) approach [[2]], which is commonly used in aerosol tracking applications. Running a time-critical simulation based on a variety of physical models can be very time-consuming. One approach is to run a series of simulations using a cluster or grid-computing network to generate a set of data from which accurate predictions of the dispersion of a contaminant could be made in a timely manner. In other words, utilizing a computing grid, it is possible to run simulations of multiple dispersion situations on parallel systems in order to quickly develop data sets for a wide range of scenarios. With these data sets, a fast response of the system can be achieved by replacing complex 3D simulations with information retrieval from a database. This idea has been proposed in the past by [10], and implemented by [5, 4] in a prototype risk analysis system. However, for realistic scenarios it proved to be difficult to realize because of the enormous number of scenarios, which need to be considered in order to cover a representative range of all possibilities. Nothing to say, that three dimensional data sets produced in CFD simulations are usually very large. The method of this study provides an improvement on this idea, which consists in replacing the database of scenarios with the database of *domain transfer probabilities* (DTP), and augmenting the database query procedure with a simplified stochastic simulator suggested by [6, 7].

Another problem addressed in this study is the efficient parallel implementation of a physically-based particle transport simulator. Distributed computer platforms, such as workstation clusters or grid computing environments, offer two main advantages for large scale simulations: (1) they enable processing of very large data sets, which typically can not fit onto a memory of a single workstation, and (2) they speed up the simulation process. However, the major obstacle in the deployment of CFD solvers on distributed computing systems is the tightly coupled nature of CFD discretization schemes, based on continuum approximations. This is especially evident for multi-phase fluid - particle systems, including particle laden flows, aerosol transport, etc, which present even more formidable parallelization problems than pure fluid dynamical systems. In particular, when discrete solvers are used in a simulation, such as LPD solvers, one may encounter serious load balancing issues, related to non-uniform particle distributions inside the domain. Also, particle transport across the sub-domain boundaries may contribute significantly to the communication overhead.

Computational clusters can be improved to some extent to suit the need of high-performance computing by using high-bandwidth communication switches and fast local area networks (LANs). However, this approach reaches its limits as the number of nodes increases. Firstly because the application of conventional domain decomposition technique will increase the communication overhead for highly decomposed domains due to the increasing boundary-to-volume ratios for smaller domains. Also, a tight coupling of the computing nodes negatively affects the performance since the cumulative failure rate from all nodes becomes a problem on a large distributed system. In addition to these, in a grid computing environment one can no longer guarantee the required high-bandwidth, since the underlying communication network is inherently slow.

In this study we combine the idea of domain transition probabilities (DTP) with a *probabilistic implicit tracking* (PIT) algorithm to develop a technique, which provides a partially decoupled domain decomposition strategy. Using this strategy, one can formulate and solve a tightly coupled physical model as a loosely coupled system, thus enabling an efficient multi-processor implementation. The approach is effective when a large number of aerosol release scenarios need to be analyzed in a timely manner. This method can form a basis for an express risk analysis system of aerosol dispersion and tracking. A prototype of such system was developed and demonstrated in the course of this study. To test the effectiveness of the approach, a hypothetical case of aerosol dispersion in an urban environment has been considered.

Finally, the accessibility of data and their storage, retrieval, and visualization issues are addressed by developing a prototype risk assessment system based on a web application for submitting database queries and analyzing the results. This application will allow the user to select a location from which a contaminant has been released and obtain accurate results in far shorter time than it would take to run the physical model. In this way, not only can the issue of accurate dispersion scenario be addressed accurately and quickly, it would also allow emergency first responders to immediately access this information stored on a common media (CD, DVD) or via a laptop computer and a wireless Internet connection from close proximity to the actual contaminant release location.

2. METHOD

2.1. Domain Transfer Probabilities

The main idea of the method is to replace one complex tightly coupled multi-physics simulator of particle transport with a sequence of two simpler simulators: (1) a partially decoupled multi-physics simulator run for each sub-domain separately and independently of other sub-domains and (2) stochastic (Monte-Carlo type) simulator applied to all sub-domains using the input from the previous decoupled simulations.

As mentioned in the previous section the improvement offered by the current method consists in replacing the database of scenarios with what we call *domain transfer probabilities* (DTPs), and substituting a stochastic simulator for the database query procedure. This method essentially reduces the size of the data that needs to be stored, or transfered over the network, since a typical DTP set can incorporate many scenarios. Figure 1 illustrates this concept. In essence DTPs determine the probability for a particle released at a certain location inside a domain or at its boundary to exit the domain at any other location on the boundary, or being deposited on the objects inside the domain. These DTPs replace the actual physical model, and enable one to compute the aerosol concentrations and depositions using a simpler stochastic simulator rather than more complex CFD/LPD schemes. Thus, using DTPs one can analyze various aerosol deposition scenarios with a commodity workstation or even a laptop, and in a shorter time than would be required for a full physically-based simulation.

An important step in constructing the DTPs is to represent the whole simulation space as a collection of well identified objects. In the case of an urban environment this representation comes naturally, since the objects can be associated with buildings, bridges, etc. Figure 2 shows a generic city map where

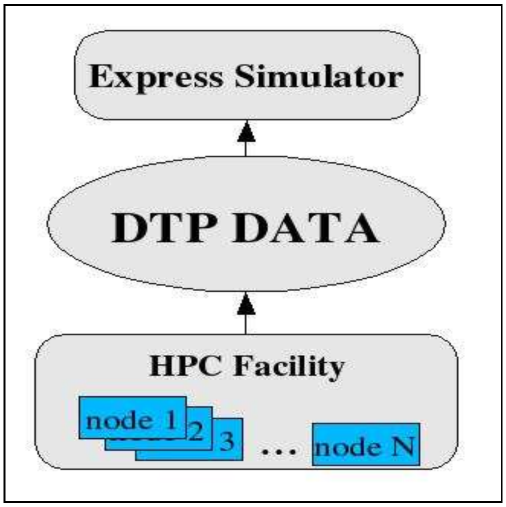


Figure 1. Express simulation scheme using DTP data

different objects of importance are marked by letters (A,B,C...). Aerosol *release scenario* may include many parameters, such as the release location, aerosol size distribution, volatility, etc., and indeed, the whole air flow field present at the time of the release is a part of the scenario parameter space.

To describe a set of solutions for various aerosol release scenarios we define DTP as the probability for a particle to be transfered from any point of the domain to any object inside the domain, or to a boundary point of the neighboring domain. The probabilistic nature of the DTP sets enables one to obtain them separately for each domain, and independently of other domains. Thus the solution procedure to construct the DTP sets can be implemented in a completely decoupled manner, and leads to the so-called *embarrassingly parallel* simulation.

We should note that this procedural decoupling applies only to the discrete phase, involving particle dynamics routines. The continuum flow field will still have to be solved using conventional domain decomposition schemes. Still, this represents a considerable improvement in overall efficiency of parallel computations, since particle tracking across domain boundaries is notoriously expensive in terms of communication overhead.

Once the DTP sets are constructed for each sub-domain, particle depositions on objects arising from an initial source (S) can be reproduced by stochastically generating particles and letting them transit from domain to domain and from domain boundaries to objects, following the assigned probabilities. Figure 3 shows the conventional way of particle tracking 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 physically-based simulations (Fig.4), and then a probabilistic procedure of particle tracking is applied, which we call *probabilistic implicit tracking* (PIT). This procedure is essentially a stochastic simulator, which uses the DTP data set generated in prior physically based simulations (Fig.5).

In particular, the DTP data produced in the first stage of physical modeling is assembled in two sets: the *transfer probabilities*, which represent boundary-to-boundary transition events, and internal *deposition probabilities* for the events of particle fallout on the objects inside each domain.

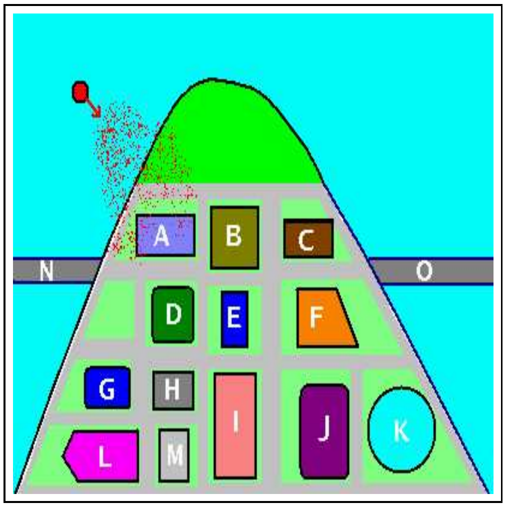


Figure 2. Aerosol dispersion in a city: A-O - domain objects.

It should be noted, that the size of DTP sets can still be too large, since the number of all point-to-point correspondences is itself a product of large numbers. However, this size can be reduced significantly, if one considers only boundary-to-boundary transition sets, and limits the number of boundary elements, which hold the probability data. This can be done by exploiting the uneven nature of aerosol distributions at the boundaries. In this case only the probabilities with non-zero values, or those above a certain threshold can be stored, which can reduce the total data storage and processing times considerably. More discussion of this technique is provided below in Sec.2.3.

Another reason why the DTP data sets can be made very compact lies in the coarse-grained representation of DTPs on the basis of discrete objects rather than continuum points. This is because the size of point-to-object relationships is much smaller than point-to-point relationships, since the number of objects in the domain is usually much smaller than the number of all possible locations.

2.2. Physical Modeling

The algorithms for fluid dynamics and particle tracking used in the first phase are based on conventional continuum diescretization schemes, and particle dynamics routines. The results of these simulations, however, have to be represented in a probabilistic manner, as discussed above, so that a simpler stochastic modeling can be used in the subsequent data analysis phase.

In this study we used an open source CFD solver OpenFOAM (openfoam.org) to compute the fluid phase and 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, $\mathbf{u}(\mathbf{x}, t)$:

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

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 [[1]]. The position of the particle at each time step is computed using midpoint interpolation scheme: $\mathbf{x} \ = \ \mathbf{x}_0 \ + \ dt(\mathbf{v} + \mathbf{v}_0)/2$.

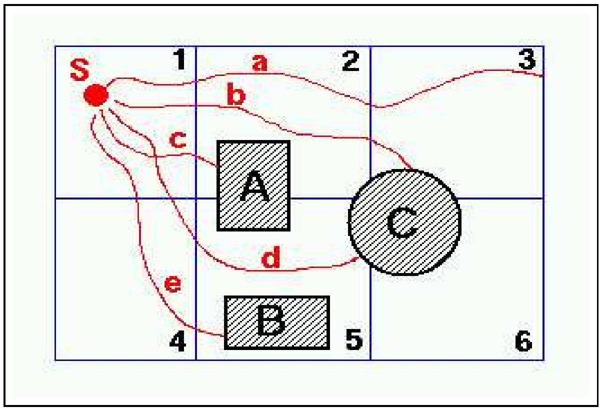


Figure 3. Particle passage through a decomposed domain: S - particle source, A...C - physical objects where the particle fallout occurs, a...e - particle trajectories, 1..6 - domains of a decomposed computational space.

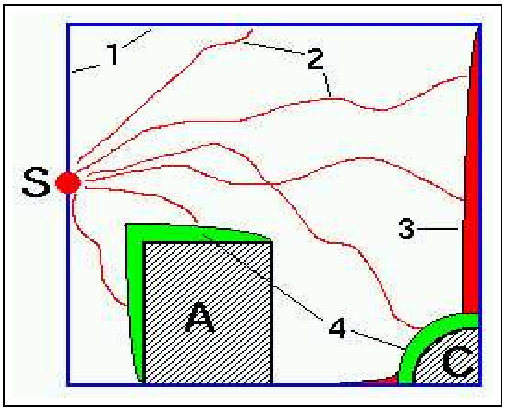


Figure 4. Assembling domain transfer probabilities: S - boundary particle source, A-C - objects, 1 - domain boundaries, 2 - particle trajectories, 3 - source-to-boundary transfer probability, 4 - source-to-object transfer probabilities.

The effects of turbulent dispersion on the particles, which are encapsulated in the second term of (1), can be accounted for by an appropriate sub-grid scale turbulence model. In particular, the RFG technique developed earlier by [3] was used to simulate the effects of turbulence on the aerosols.

In the stage of physically based modeling, a CFD/LPD solver is used to simulate each aerosol particle as it is convected in a velocity field. The particle is traced inside the computational domain until it crosses the domain boundary or hits an object inside the domain. In the first event the corresponding boundary hit count is incremented, and in the second event the hit count for that object is updated. The final DTP is obtained by dividing all the hit counts by the number of particles released. The statistical error of transfer and deposition probabilities computed in this manner will be inversely proportional to the square root of the number of particles: $p \sim 1/\sqrt{Np}$. This provides a flexibility of adjusting the

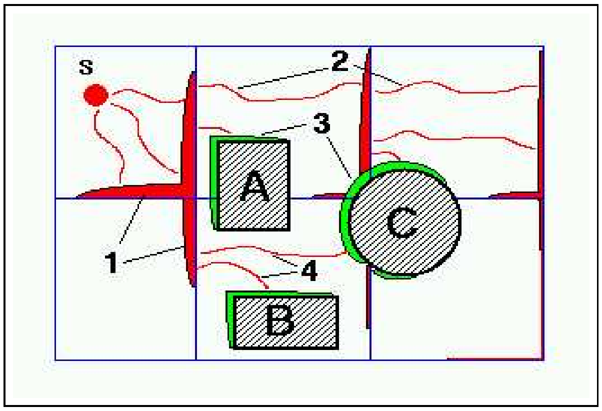


Figure 5. Implicit particle tracking using DTPs: S - initial particle source, 1 - boundary particle sources, 2 - boundary-to-boundary transfer probabilities, 3 - object sinks, 4 - boundary-to-object transfer probabilities,

accuracy of the simulations by selecting the particle sample of a suitable size.

2.3. *Probabilistic Implicit Tracking*

There are two approaches to find the probability of particle deposition at a certain location of a some domain, given its release from a different location of possibly another domain. In the first approach one can use the appropriate relations from the probability theory, in the second approach one can use stochastic simulation.

Let's consider the probability $P(A|B_i)$ of particle deposition on object A when it was released from location i at boundary B[†]. Then the total probability of particle deposition at A is given by [[11]]:

$$P(A) = \sum_{i}^{N_B} P(B_i)P(A|B_i)$$

where N_B is the total number of elements on the boundary B, and $P(B_i)$ is the probability of particle's occurrence at the boundary. The latter can be computed using the cross-boundary transfer probabilities from the neighboring domain:

$$P(B_i) = \sum_{i}^{N_B^n} P(B_j^n)P(B_i|B_j^n)$$

where N_B^n is the total number of elements on the boundary of the neighbor domain, $P(B_j^n)$ is the probability of particle's occurrence at the neighbor domain boundary, and $P(B_i|B_j^n)$ is the domain transfer probability between element B_j^n at the neighbor domain boundary, and element B_i at the current domain boundary[‡]. To find $P(B_j^n)$ one has to apply this formula again but this time using

[†] More exactly, B_i represents boundary element i, such as obtained by surface triangulation

[‡] Neighbor domain boundary common with the current boundary should not be considered

DTPs of the neighbor to the neighbor domain, etc. until the all domains have been exhausted. The whole procedure can be formulated as a recursive algorithm spanning all the boundary elements on all the domains. The initial non-zero boundary particle probabilities will come from the domain containing the particle source. Although possible in principle, this procedure will involve repeated summations over all the boundary elements, and with a growing number of domains it can become prohibitively expensive.

In contrast the PIT procedure uses a stochastic simulator, which generates a certain number of particles at the source and moves them randomly from boundary to boundary following the assigned DTPs. This process stops when all particles have crossed the outer boundaries, which have no neighbors across, or are deposited on inside objects. Thus, the procedure follows this simplified algorithm:

```
While the number of particles > 0 do:
   For each particle do:
       Transfer particle
```

This procedure can be organized by looping through particles, or by looping through all boundary elements. The former will run faster, since empty elements will be automatically avoided, but it will require additional bookkeeping for each particle to identify its current host element at the boundary.

The transfer operator for each particle provides a simple assignment of a new object Y given an old object X and the object-to-object transfer probability function, P:

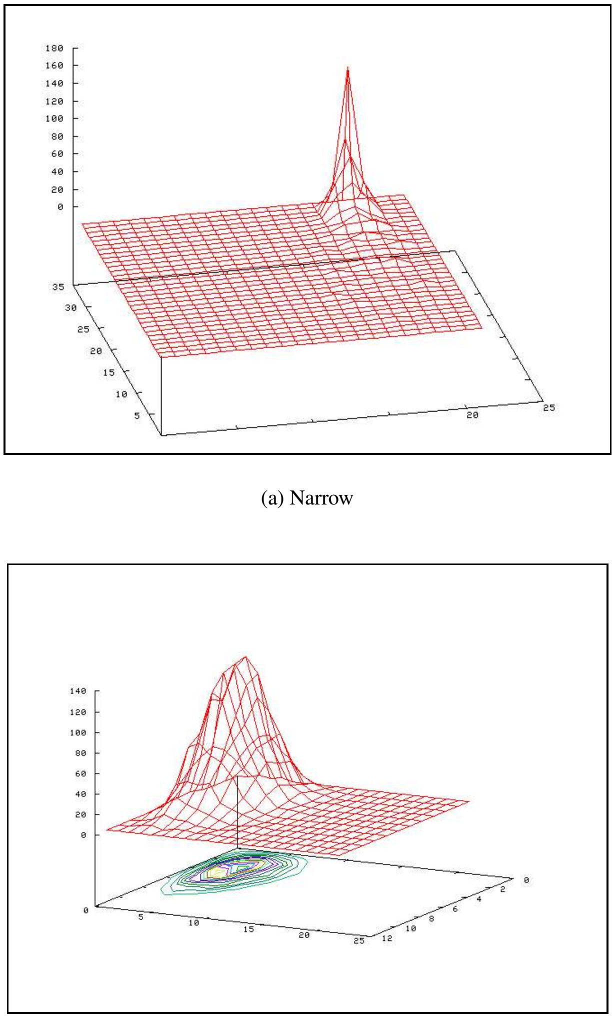
$$X \xrightarrow{P} Y$$

where the input object, X, should be a boundary element, the output object Y can be either a boundary element or a domain object, and the transfer probability function P is given by the corresponding DTP data.

Figure 6 shows typical probability distributions of aerosol fallout on a particular object as a function of wind direction and one of the three spatial positions (in this case the height). The dependence in two other spatial directions shows similar features. The shape of the distributions depends on wind velocity and turbulence levels, and is usually very spiky (Fig.6(a)). This shape is attributed to the fact that in most cases the particles are carried away from the objects causing zero deposition counts. It can be seen from Fig.6 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.

To achieve further reduction in both required memory and execution time one can simply ignore the DTP values lower than a certain specified threshold, and use only statistically significant contributions from the DTP set. With the appropriate selection of the threshold, such reduction will not affect the outcome of the simulations, unless one is interested in very low deposition areas, which is usually not the case. For example, if one is interested in the most contaminated areas, and can ignore the fallout of less than 1% of the total, then this cutoff technique will make sense. Since there is usually a considerable number of low count events in DTP distributions, introducing such a cutoff will lead to another drastic reduction of the data size and computing time. The results of using the *DTP-cutoff* technique are presented in Sec.3.

Another data 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.



(a) Narrow

(b) Wide

Figure 6. Typical DTPs for different space locations and wind directions

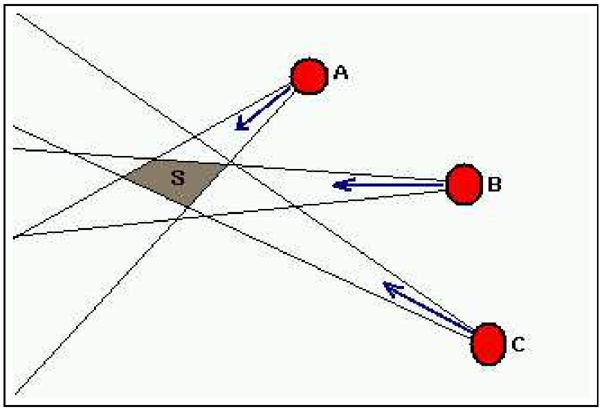


Figure 7. Backtracking particles to the source: A,B,C - fallout measurement locations. Shaded area S - possible particle source location.

2.4. The Inverse Problem

One of the further advantages of PIT method is the possibility to solve the inverse problem almost as easily as the direct problem. This comes from the fact that using the domain transfer probability data, and applying the Bayes' Rule [[11]], one can run the particles in reverse, that is, from their fallout locations all the way to the release location. This idea is illustrated in Figure 7. Suppose there are available measurements of fallout at some locations, like A, B, and C as shown in the figure. Then one can release fictitious particles from these measurements locations and run them backwards using the inverted DTP data. The area where most of the reverse paths intersect will identify the possible location of particle release.

It should be noted that backtracking particles is also possible with conventional deterministic methods, such as those based on CFD/LPD schemes. However, using a probabilistic method, such as presented here, has certain advantages. In particular, reversing the velocity of a particle in an LPD scheme can in fact introduce a systematic error in predicting its original location, which will be consistently shifted away from the true origin (Fig.8). This effect is due to accumulation of discretization errors in numerical schemes for PDEs, and is not present in a stochastic PIT scheme. In a stochastic algorithm the particles are not traced through the domain following a complex iterative scheme, but instead are transferred from boundary to boundary in a single step using a known probability distribution. Thus, the accumulation of errors, as happens during the reverse trajectory tracking in LPD schemes does not take place in PIT scheme. Moreover, any systematic errors of this type that might have occurred during the compilation of the DTP sets using LPD simulations, are completely reversed in probabilistic backtracking. Thus in principle, the particle release source can be traced more accurately with the reversed PIT scheme.

3. RESULTS

In this work the issues discussed above have been addressed and implemented in a prototype risk assessment system for aerosol transport (RASAT) in urban environments. There are several aspects

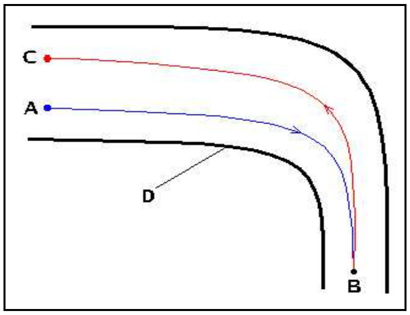


Figure 8. Backtracking particles in LDP introduces systematic errors: A - particle source, B - particle backtracking start point, C - particle backtracking end point, D - duct wall.

required to implement the method:

1. Physical model enabling accurate simulations of events of interest;
2. The possibility to conduct parallel and possibly exhaustive sets of simulations;
3. Data reduction and efficient storage;
4. Data retrieval and visualization.

The prototype system consists of a simulator and front-end interface. The simulator incorporates a physically based model of aerosol transport and is executed on high-performance computing facilities, such as computer clusters or grid computing environments. It collects the data of possible aerosol release scenarios. The front-end interface runs as a web-based application and retrieves the data for particular scenarios and their outcomes.

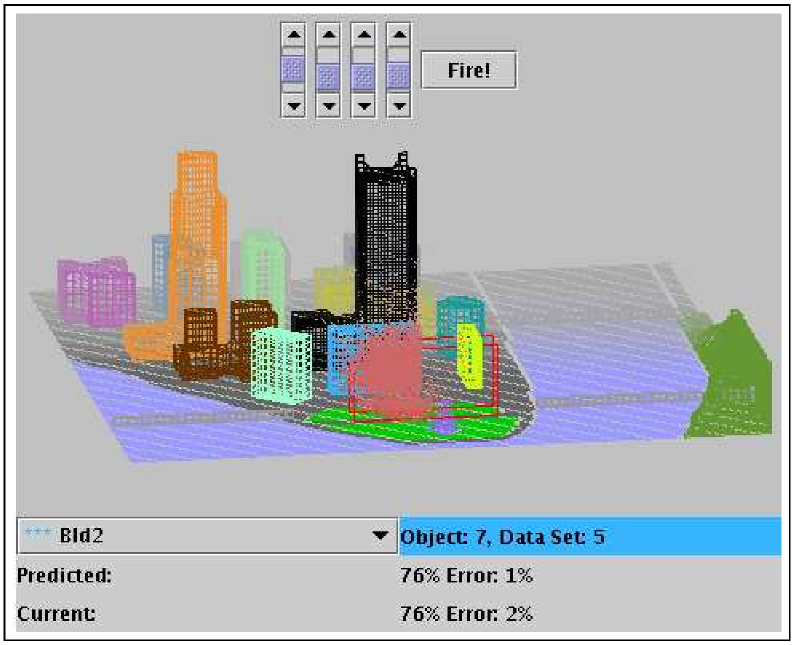
In our feasibility study a generic city landscape was set up and prototyped after the Pittsburgh downtown area (Fig.9), using the voxel-based 3D graphics system developed by [9, 8]. The whole domain was discretized on the $92 \times 92 \times 32$ grid and populated with characteristic features like rivers, hills, bridges, park area, pavements and buildings.

Three sets of simulations were performed: (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 purpose of the last two simulations was to conduct a validation study and compare the results and performances between the PIT scheme and a conventional method. All three simulators were implemented in a C++ language and run on different hardware platforms (see Sec.3.2).

3.1. Parallel Simulations

In the simulations of aerosol transport and dispersion the whole scene was sub-divided 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. Figure 10 shows the typical velocity and turbulent kinetic energy distribution in a horizontal cross-section taken in the middle of the domain.

The processor time required followed a near linear dependence on the number of particles (Tab.I), and for the 10^5 particles run the average time for executing the DTP calculations on a single node was close to 5.5 hours. The variance of the execution time was due to the differences in the number and shapes of objects inside each domain.



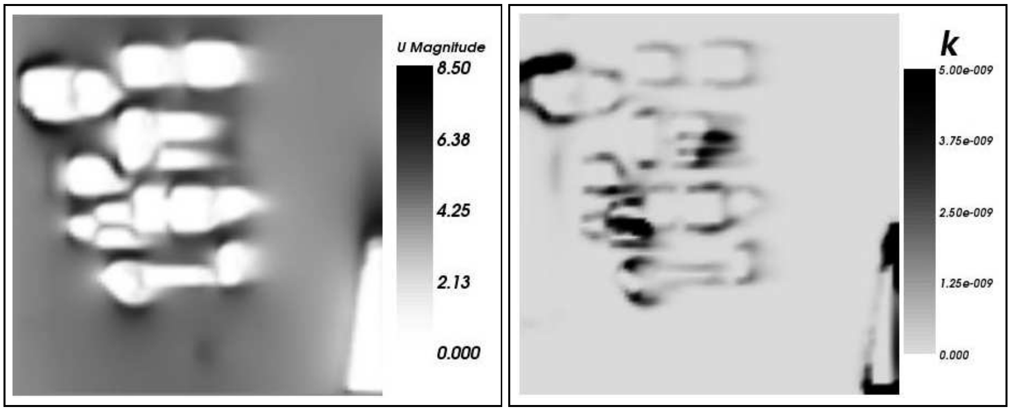
*** Bld2

Object: 7, Data Set: 5

Predicted: 76% Error: 1%

Current: 76% Error: 2%

Figure 9. Web interface to simulate aerosol release in a city (http://mulphys.com/rasat/demo)



(a) Mean velocity

(b) Turbulent energy

Figure 10. Velocity and turbulent kinetic energy fields

N_p	CPU[s]	VAR[s]
10^2	19	6
10^3	189	70
10^4	1894	711
10^5	19980	7930

Table I. Wall clock time of parallel runs for different number of particles: CPU=execution time, VAR=variance between different processors.

3.2. Validation

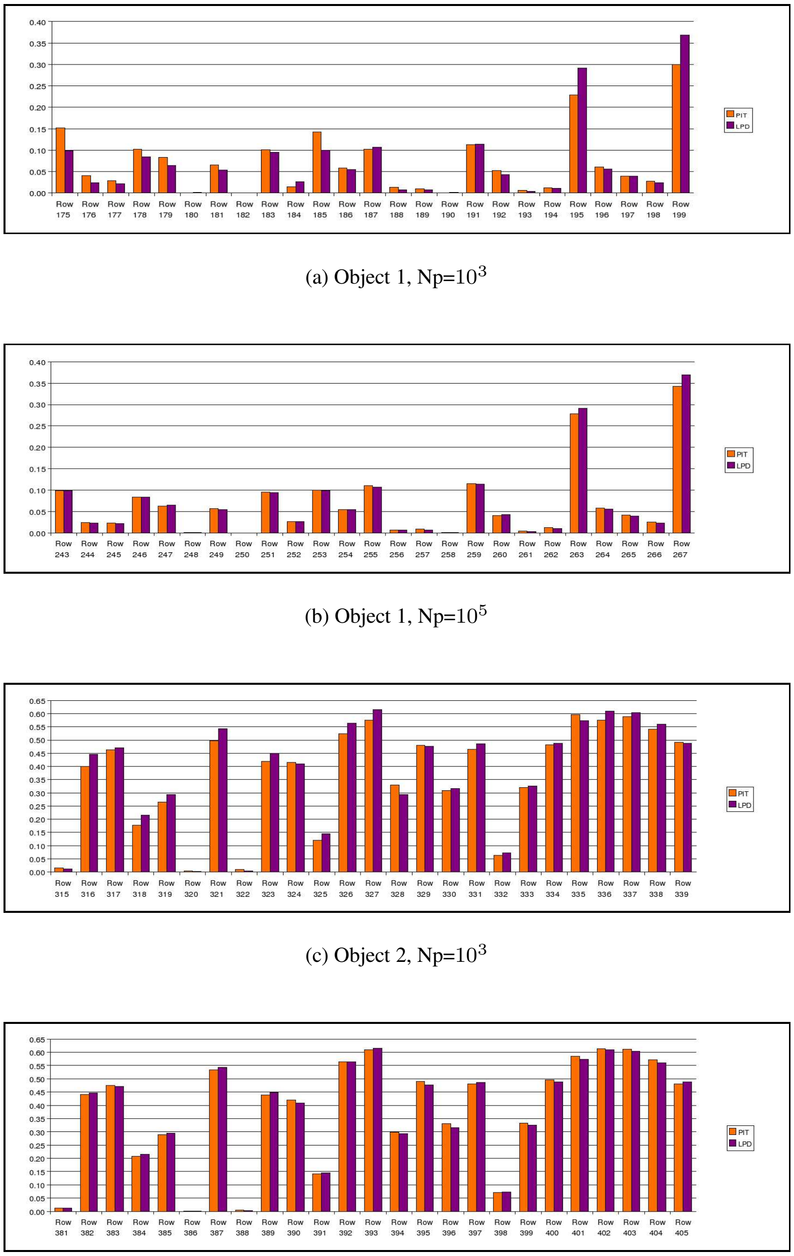
To validate the method, separate simulations were performed in a conventional (non-parallel) manner, where the physical CFD/LPD solver was applied to the whole computational space without using domain decomposition. The validation simulations consisted of two separate runs: a CFD solver (OpenFOAM) computes several scenarios of flow fields, which were fed into LPD solver that computed different particle release scenarios for each given flow-field. The results were compared with those obtained from the stochastic simulator, using the PIT scheme.

Figure 11 shows typical histograms of particle distributions for LPD and PIT methods obtained for two objects, one experiencing a relatively low and another - large particle fallout, and 25 different aerosol release scenarios. The comparison is shown for two DTP sets used: one computed for 10^3 and another for 10^5 particles. The data collected for different scenarios show a very good agreement between the two methods with the average deviation of the results typically within 3% for 10^5 particles. However, this value may vary dependent on the number of particles deposited on each particular object.

The validation (CFD/LPD) simulation was conducted on the 3GHz, 2GB workstation, and the typical computing time for one flow scenario with a standard turbulence model (k-epsilon) took on the average one day. The stochastic simulations using PIT scheme were conducted on a 1.8 GHz, 1GB Pentium 4 laptop. The execution times for stochastic simulations are given in Table II. As can be seen from this result, the execution time of stochastic algorithm is largely insensitive to the number of particles used to generate the DTP sets, especially for larger number of particles. This is because the execution time of the stochastic algorithm depends primarily on the size of the DTP sets, and the latter is only weakly affected by the number of particles used in physically-based aerosol simulations. This weak non-linear dependence is attributed to small changes in DTP set size due to more frequent occurrences of near-zero counts (see discussion of DTP-cutoff below). In contrast the execution time of the LPD scheme is linearly proportional in the number of particles, as it should. Thus, the speedup of the simulations with the PIT scheme can be quite considerable, especially when high-accuracy computations with large number of particles are involved. This speedup will become even more pronounced as more sophisticated physical models are used to produce the DTP sets.

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 small response time with a reasonably good accuracy. In our tests we used 10^4 particles for the stochastic algorithm, since that guaranteed less than one minute response time even for the largest DTP sets.

As discussed above the size of the DTP sets and consequently the execution time of the PIT scheme can be considerably reduced by ignoring the low probability data. It was noted that because of a non-uniform nature of the DTP distributions, most of the size of the DTP sets is taken by very



(a) Object 1, Np=10^3

(b) Object 1, Np=10^5

(c) Object 2, Np=10^3

(d) Object 2, Np=10^5

Figure 11. Comparison of aerosol deposition data computed with LPD and PIT methods for two different objects and different number of particles. X-axis corresponds to different aerosol release scenarios.

small probability counts. These low counts can be safely ignored for most practical purposes with a substantial gain in execution time and reduction in data size. To analyze the effects of DTP cutoff on DTP sizes and execution times three cutoff thresholds were applied to the probability sets: 1%, 2%, and 5% respectively. The results are presented in Table III. As can be seen, already a 1% cutoff can lead to a considerable reduction in CPU time. However, further increase in cutoff did not introduce a considerable time reduction. Even greater savings were achieved in total data size, also shown in Table III.

N_p	10^2	10^3	10^4	10^5
CPU	3	15	21	27

Table II. Execution time of PIT algorithm dependent on the number of particles in the DTP set.

3.3. Data Retrieval and Visualization

From the perspective of express risk analysis it would be highly desirable to have a direct access to the simulation data on a local workstation, laptop, or via the Internet. In addition to this the application should be platform independent, and should run equally well on different operating systems. A simple solution would be an efficient web interface to aerosol dispersion data base. This interface should provide a reasonably short access time to the data no matter how large the total data set can be, and at the same time it should be simple and intuitive to use.

Such prototype web interface was developed in the course of this study, which enabled us to test different scenarios of aerosol release and dispersion. The application is written in Java language and provides a 3D representation of a city with the possibility of navigating through the landscape, arbitrary positioning of the aerosol source, and setting wind direction (Fig.9). 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.

When tested on a 1GB 1HG laptop with 100MB Internet connection the speed of retrieval of each dataset was on the average 10 to 20 times faster than the Java execution of the simulation for 10^4 particles. Considering that the retrieved data set consisted of 1000 realizations, it leads to a much greater speedup if multiple local scenarios need to be analyzed.

CUTOFF	0	1%	2	5%
CPU	27 ± 1.2	4.5 ± 0.13	$3.1 \pm 0.06,$	2.3 ± 0.06
SIZE (MB)	559	59	35	19
SIZE (%)	100	11	6	3

Table III. Execution time of PIT algorithm dependent on the probability cutoff in the reduced DTP sets of 10^5 particles computed for 25 runs.

4. CONCLUSIONS AND FUTURE WORK

The method of this study is based on information retrieval from compressed data sets obtained in prior exhaustive simulations of different aerosol release scenarios.

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. The physical simulator still needs to produce the DTP data sets, but it can be implemented in a completely decoupled (*embarrassingly parallel*) manner, since no inter-processor communication is required in LPD schemes to produce the DTP data sets. The algorithm can then be efficiently executed in multi-processor and distributed computing environments.

As such the method serves a dual purpose of (1) providing an *embarrassingly parallel* implementation for certain classes of transport problems on grid computing environments and (2) facilitating express risk analysis of multiple scenarios of a complex physical event. A particularly relevant problem is the express analysis of possible aerosol contamination in urban environments. 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 represented as a set of floors, etc.

A unique advantage of the method is the possibility to accurately solve the inverse problem, that is, identification of particle release source based on available fallout data as discussed in Sec.2.4. A validation of this feature will be addressed in the future work.

Acknowledgments

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

REFERENCES

1. C. Crowe, M. Sommerfeld, and Y. Tsuji. *Multiphase Flows with Droplets and Particles*. CRC Press, 1998.
2. C. Growe, M. Sommerfeld, and Y. Tsuji. *Multiphase Flows with Droplets and Particles*. CRC Press, 1999.
3. 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.
4. Andrei Smirnov and Steven Rowan. Express risk assessment through web access to simulation data. In *Third International Conference on Information Technology: New Generations*, number 114 in ITNG-06, pages 584–585, Las Vegas, NV, 2006. IEEE Computer Society.
5. Andrei Smirnov and Steven Rowan. A fast response system for aerosol dispersion forecasting. In *17th IASTED International Conference: Modeling and Simulation*, number 530-050 in MS, pages 37–42, 2006.
6. Andrei Smirnov, Steven Rowan, and James McCormick. Aerosol dispersion modeling with probabilistic implicit particle tracking algorithm. In *2006 International Aerosol Conference*, number 9G25 in IAC-06, St. Paul, Minnesota, 2006. American Association for Aerosol Research (AAAR).
7. Andrei Smirnov, Steven Rowan, and James McCormick. Probabilistic physical modeling on distributed computing systems. In *The 2006 World Congress in Computer Science, Computer Engineering, and Applied Computing*, number PDP-4004 in WCC-06, Las Vegas, NV, 2006.
8. A.V. Smirnov, H. Zhang, A. Burt, and I. Celik. Fuel-cell simulator interface. *Journal of Power Sources*, 138/1-2:187–193, 2004.

9. 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.
10. J. Thilmen. Harm's way. *Mechanical Engineering*, 127:22–24, 2005. bioterror CFD.
11. W.Navidi. *Statistics for Engineers and Scientists*. McGraw-Hill, 2006.