

# Parallel Implementation of a Lagrangian Stochastic Model for Pollution Dispersion

Debora R. Roberti<sup>1</sup>, Roberto P. Souto<sup>2</sup>, Haroldo F. de Campos Velho<sup>2</sup>,  
Gervasio A. Degrazia<sup>1</sup>, Domenico Anfossi<sup>3</sup>

<sup>1</sup> Department of Physics, Federal University of Santa Maria (UFSM)  
Santa Maria (RS), Brazil

<sup>2</sup> Laboratory for Computing and Applied Mathematics (LAC)  
National Institute for Space Research (INPE)  
São José dos Campos (SP), Brazil

<sup>3</sup> Institute of Atmospheric Sciences and Climate (ISAC)  
Italian National Research Council (CNR)

E-mails: d\_r\_roberti@yahoo.com.br, roberto@lac.inpe.br, degrazia@cnme.ufsm.br,  
haroldo@lac.inpe.br, D.Anfossi@isac.cnr.it

## Abstract

*Pollutant dispersion models in the atmosphere can be described by Eulerian or Lagrangian approaches. Lagrangian models belong to the class of Monte Carlo methods. This type of method is very flexible, solving more complex problems, however this computational cost is greater than Eulerian models, as it is well established in the atmospheric pollutant and nuclear engineering communities. A parallel version of the Lagrangian particle model – LAMBDA – is developed using the MPI message passing communication library. Performance tests were executed in a distributed memory parallel machine, a multicomputer based on IA-32 architecture. Portions of the pollutant in the air are considered particles emitted from a pollutant source, evolving under stochastic forcing. This yields independent evolution equations for each particle of the model, that can be executed by a different processor in a parallel implementation. Speed-up results show that the parallel implementation is suitable for the used architecture.*

## 1. Introduction

The pollutant dispersion in the atmosphere is a topic with high interest nowadays. However, inside of the Planetary Boundary Layer (PBL) the turbulence is a permanent feature [3, 26], becoming this problem in a very hard. The turbulence in the PBL is a complex process, depending on the thermodynamic state of the atmosphere. Therefore, the PBL can be characterized as a Convective Boundary Layer, where thermal processes govern the turbulence; Neutral Boundary Layer, when the turbulence is maintained by mechanical production

only; or a Stable Boundary Layer, occurring in situations in which the heat flux is coming from the atmosphere to the Earth (removing energy from turbulent flow) and the mechanical turbulent production is present. Different types of PBL implies in a different parameterizations for the atmospheric turbulence. In addition, complex topography is an additional difficulty in real world applications. The Lagrangian dispersion models are good alternatives dealing on complex terrain. However, these models are computationally intensive.

Models for air monitoring are not only important for describing the pollutant impact over urban or rural areas, or for a urban planning consideration (these are examples where the forward problem is important considering several scenarios), but other issues are also relevant, such as the pollutant source strength estimation, CO<sub>2</sub> diurnal cycle, and total ozone in the atmosphere. The last three issues are examples of inverse problems in atmospheric pollution. Inverse problems are usually solved by an *implicit technique*, where the inverse problem is formulated as a constrained nonlinear optimization problem: the forward problem is iteratively solved for successive approximations of the unknown parameters. The associated forward problem is the solution of the dispersion model. In a typical inversion, thousands of iterations may be required and, therefore, the choice of an algorithm that is suitable for parallelization has an important role.

## 2. Description of the LAMBDA Model

The Lagrangian particle model LAMBDA [10] was developed to study the transport process and pollutants

diffusion, starting from the Brownian random walk modeling. This model is based on a three-dimensional form of the Langevin equation for the random velocity [28]. The velocity and the displacement of each particle are given by the following equations [24]:

$$du_i = a_i(\mathbf{x}, \mathbf{u}, t)dt + b_{ij}(\mathbf{x}, \mathbf{u}, t)dW_j(t) \quad (1a)$$

and

$$dx = (\mathbf{U} + \mathbf{u})dt, \quad (1b)$$

where  $i, j = 1, 2, 3$ , and  $x$  is the displacement vector,  $\mathbf{U}$  is the mean wind velocity vector,  $\mathbf{u}$  is the Lagrangian velocity vector,  $a_i(\mathbf{x}, \mathbf{u}, t)dt$  is a deterministic term and  $b_{ij}(\mathbf{x}, \mathbf{u}, t)dW_j(t)$  is a stochastic term and the quantity  $dW_j(t)$  is the incremental Wiener process.

Thomson considered the Fokker-Planck equation [24] as Eulerian complement of the Langevin equation to obtain the deterministic coefficient  $a_i(\mathbf{x}, \mathbf{u}, t)$ . The stationary Fokker-Planck equation is given as:

$$\frac{\partial}{\partial x_i}(u_i P_E) = -\frac{\partial}{\partial u_i}(a_i P_E) + \frac{\partial^2}{\partial u_i \partial u_j} \left( \frac{1}{2} b_{ij} b_{jk} P_E \right) \quad (2)$$

where  $P_E$  is the non-conditional probability density function (PDF) of the Eulerian velocity fluctuations and the other symbols have the same definitions as in equations (1a, 1b). The deterministic coefficient  $a_i(\mathbf{x}, \mathbf{u}, t)$  is obtained from:

$$a_i P_E = -\frac{\partial}{\partial x_j} \left( \frac{1}{2} b_{ij} b_{jk} P_E \right) + \phi_i(x, u, t) \quad (3a)$$

$$\frac{\partial \phi_i}{\partial u_j} = -\frac{\partial P_E}{\partial t} - \frac{\partial}{\partial x_i}(u_i P_E), \quad (3b)$$

subject to the condition

$$\phi_i \rightarrow 0 \text{ when } \mathbf{u} \rightarrow \infty. \quad (3c)$$

While in the two horizontal directions the  $P_E$  is considered to be Gaussian, in the vertical direction the PDF is assumed to be non-Gaussian (to deal with non-uniform turbulent conditions and/or convection). The non-Gaussian PDF more used is the bi-Gaussian one, truncated to the third order [2].

The coefficient  $b_{ij}(\mathbf{x}, \mathbf{u}, t)$  is obtained from the following Lagrangian structure function (the ensemble average of the square of the change in Lagrangian velocity in the time interval  $\Delta t$ ):

$$D(\Delta t) = \left\langle [u_i(t + \Delta t) - u_i(t)]^2 \right\rangle = \left\langle (\Delta u_i)^2 \right\rangle. \quad (4)$$

For the inertial subrange ( $\tau_K \ll \Delta t \ll \tau_L$ ), where  $\tau_K$  is the Kolmogorov time scale and  $\tau_L$  is the Lagrangian

decorrelation time scale, the structure function, according to Kolmogorov, is given by

$$D(\Delta t) = C_0 \varepsilon(x, t) \Delta t, \quad (5)$$

where  $C_0$  is the Kolmogorov constant and  $\varepsilon(x, t)$  is the ensemble-average rate of dissipation of turbulent kinetic energy. Using the square of equation (1a) and applying the conditions for the Wiener process, the change in Lagrangian velocity is:

$$\left\langle (\Delta u_i)^2 \right\rangle = b_{ij} b_{jk} \Delta t. \quad (6)$$

Then, from the equations (5) e (6), it is possible to show that  $b_{ij}(\mathbf{x}, \mathbf{u}, t)$  is related to  $C_0$  of the following way:

$$\left\langle b_{ij} b_{jk} \right\rangle = \delta_{ij} C_0 \varepsilon, \quad (7)$$

where  $\delta_{ij}$  is the Kronecker delta.

LAMBDA uses a determined number of fictitious particle to simulate the atmospheric diffusion. Each particle can be marked for a mass. Which can be constant or variable in the time, due to deposition effects and reaction chemistry [30]. However, the spatial distribution of particle mass on computational domain allows the computation of a three-dimensional concentration field. The concentration (mass by  $m^3$ ), in a given time and position, is determined to account of the particle number in a cell or imaginary volume centered in  $x, y, z$ :

$$C(x, y, z, t) = m_p \frac{N_v}{V_c} \quad (8)$$

where  $m_p$  is the mass of the each particle,  $N_v$  is the particle number on the volume concentration and  $V_c$  is the concentration volume. The mass of the each particle is determined to following equation:

$$m_p = \frac{Q(t) N_{\Delta t}}{N_p}, \quad (9)$$

where  $Q(t)$  is the emission range,  $N_{\Delta t}$  is the steps in time and  $N_p$  is the particle number emitted.

## 2.1 Parameterization of drift and diffusion terms in the Langevin equation

In order to simulate the turbulent diffusion employing the Langevin equation (1.a) the turbulent velocity need to be expressed as variances  $\sigma_i^2$  and decorrelation time scales ( $\tau_{L_i}$ ). Accounting for the current knowledge of the PBL structure and characteristics, Degrazia et al. [8] have derived parameterizations for  $\sigma_i^2$  and  $\tau_{L_i}$ . This parameterization is obtained from the Taylor statistical diffusion theory; observed spectral properties, where a

linear combination of the two turbulent forcing mechanisms (shear + buoyancy) is assumed [13], and the value of the wavelength associated to the energy containing eddies. These parameterizations give continuous values for the PBL at all elevations ( $z_0 \leq z \leq h$ ) and all stability conditions from unstable to stable ( $-\infty < L < \infty$ ), where  $h$  is the PBL height,  $z_0$  is the aerodynamic roughness length and  $L$  is the Monin-Obukhov length. The general expressions for  $\sigma_i^2$  and  $\tau_{L_i}$  are the following:

$$\sigma_i^2 = \frac{1.06 c_i \left( \psi_\varepsilon^c \frac{z}{z_i} \right)^{2/3} w_*^2}{[(f_m^*)_i]^2/3} + \frac{2.32 c_i (\phi_\varepsilon^{n+s})^{2/3} u_*^2}{[(f_m^*)_i]^{2/3}} \quad (11)$$

$$\tau_L = \frac{z}{\sqrt{c_i}} \left\{ \frac{0.14 \left( \frac{-\bar{L}}{z_i} \frac{z_i}{-L} \right)^{1/2}}{[(f_m^*)_i]^2/3 w_* \left( \psi_\varepsilon^c \frac{z}{z_i} \right)^{1/3}} + \frac{0.059}{[(f_m^*)_i]^{n+s} (\phi_\varepsilon^{n+s})^{1/3} u_*} \right\} \quad (12)$$

where  $w_*$  is the convective velocity scale,  $u_*$  is the local friction velocity,  $(f_m^*)_i^c$  is the reduced frequency of the convective spectral peak,  $(f_m^*)_i^{n+s}$  is the reduced frequency of the neutral or stable spectral peak,  $L$  is the Monin-Obukhov length,  $-\bar{L}/z_i$  is an average stability parameter for the convective PBL,  $\psi_\varepsilon^c = \varepsilon h / w_*^3$  and  $\phi_\varepsilon^{n+s} = (\varepsilon_s \kappa z) / u_*^3$  are the nondimensional molecular dissipation rate functions associated to buoyancy and mechanical productions, respectively,  $\varepsilon_b$  is the buoyant rate of Turbulence Kinetic energy (TKE) dissipation  $\varepsilon_s$  is the mechanical rate of TKE dissipation,  $\kappa$  is the von Karman constant, and  $c_i = \alpha_i \alpha_u (2\pi\kappa)^{-2/3}$  with  $\alpha_u = 0.5 \pm 0.05$  and  $\alpha_i = 1, 4/3, 4/3$  for  $u$ ,  $v$  and  $w$  components, respectively.

The turbulence parameterization from equation (1) will be complete having expressions for  $w_*$ ,  $u_*$ ,  $(f_m^*)_i^c$  and  $(f_m^*)_i^{n+s}$ . For a convective PBL

$w_* = (u_*)_0 (-h / \kappa L)^{1/3}$  and, recalling that  $(f_m^*)_i^c = z / (\lambda_m)_i$ , being  $(\lambda_m)_i$  the peak wavelength of the turbulent velocity spectra,  $(f_m^*)_i^c$  expressions for all wind velocity components can be derived. According to Kaimal et al. [19], Caughey and Palmer [5],

$$(\lambda_m)_u = (\lambda_m)_v = 1.5 z_i \quad (13)$$

and

$$(\lambda_m)_w \approx \begin{cases} z / (0.55 - 0.38 |z/L|) & 0 \leq z \leq |L| \\ 5.9z & |L| < z \leq 0.1h \\ 1.8h [1 - \exp(-4z/h) - 3 \times 10^{-4} \exp(-8z/h)] & 0.1h < z \leq h \end{cases} \quad (14)$$

so that

$$(f_m^*)_i^c = \frac{z}{(\lambda_m)_w} \quad (15)$$

For neutral or stable PBL,  $u_*^2 = (u_*)_0 (1 - z/h)^{\alpha_1}$ , in which  $\alpha_1 = 1.7$  for the neutral case [29]. Then, following Delage [9], and Sorbjan [25], and Stull [26], it is obtained:

$$(f_m^*)_i^{n+s} = (f_m)_{is}^n \left( 1 + a_i \frac{f_c z}{G} + 3.7 \frac{z}{\Lambda} \right), \quad (16)$$

where  $(f_m)_{is}^n$  is the frequency of the spectral peak in the surface for neutral conditions,  $G$  is the geostrophic wind speed,  $f_c = 10^{-4} s^{-1}$  is the Coriolis parameter and  $\Lambda = L(1 - z/h)^{(1.5\alpha_1 - \alpha_2)}$  [6] is the local Monin-Obukhov length. For a shear dominated stable boundary layer,  $\alpha_1 = 1.5$  and  $\alpha_2 = 1.0$  [21]. According to Olesen et al. [22] and Sorbjan [25],  $(f_m)_{us}^n = 0.045$ ,  $(f_m)_{vs}^n = 0.16$  and  $(f_m)_{ws}^n = 0.33$ . Furthermore,  $a_w = 500$  [16, 17] as a consequence of the Blackadar's mixing length hypothesis [4] (i.e., the asymptotic length scale  $l_\infty \approx G / f_c$  is limited by a constant value, equal for all the components) we found  $a_u = 3889$  and  $a_v = 1094$ . Then, by writing  $f_c z / G = [f_c z / (u_*)_0] [(u_*)_0 / G]$  where  $(u_*)_0 / G$  is the neutral geostrophic drag coefficient, equation (16) results

$$(f_m^*)_i^{n+s} = (f_m)_{is}^n \left( 1 + a_i \frac{f_c z (u_*)_0}{(u_*)_0 G} + 3.7 \frac{z}{\Lambda} \right) \quad (17)$$

By considering  $(u_*)_0 / G = 0.03$  [18, 27] equation (16) can be written as

$$(f_m^*)_i^{n+s} = (f_m)_{is}^n \left( 1 + 0.03 a_i \frac{f_c z}{(u_*)_0} + 3.7 \frac{z}{\Lambda} \right). \quad (18)$$

It is important to emphasize the meaning of continuous parameterization for all stabilities in the present paper is the following. At the same time and location stable and unstable conditions cannot coexist. Unstable and neutral effects may be jointly take into account due to the contemporary presence of mechanical and convective turbulence. Stable and neutral effects may also jointly be accounted for because of the competition between wind-shear generated turbulence and stabilising effects of stratification. Therefore, in unstable-neutral conditions the last term of equation (5) is set equal to zero, whereas in the neutral-stable conditions the term  $1/(f_m^*)_i^c$  is set to zero.

### 3. Parallel Implementation

A preliminary study of the LAMBDA code has led to the chosen parallelization strategy. Immediately two strategies can be thought: (i) a domain decomposition, where the space integration domain is partitioned in small volumes, where each volume (sub-domain) is addressed for different processor; (ii) the particles, representing the pollutant substance, are divided among the processors.

The main difficulty in the first strategy is that the particles would be migrating from a sub-domain to other. Therefore a dynamic load balancing is necessary at each time-stepping.

The second strategy provides a parallel implementation that distributes these particles uniformly among processors, becoming the dispersion problem into an independent ensemble of the Lagrangian models. Parallelization based on assigning different particles to different processors is straightforward and was employed in this work. Therefore, a set of particles are released to evolving on different processors.

Another significant point is that, for any strategy, as stated on the Amdahl's law [23], the gain in processing time is limited to the fraction of the code that can be executed in parallel. This profile was obtained by means of the gprof Unix/Linux profiling tool. The adopted strategy was then confirmed.

The related code was parallelized using calls to the message passing communication library MPI and executed on a distributed memory parallel machine. This machine is a cluster of 17 standard IA-32 architecture processors connected by a standard FastEthernet interconnection network and a 24-port switch. High communication latencies are characteristic of this hardware arrangement and, therefore, either coarse granularity or latency hiding programming techniques are a must.

### 4. Performance Results

The proposed parallel implementation of the LAMBDA code was tested with a characteristic data from Copenhagen experiment. This dispersion experiment was carried out in the northern part of Copenhagen [14, 15] The pollutant ( $\text{SF}_6$ ) was released without buoyancy from a tower at a height of 115 m and collected at the ground-level positions in up to three crosswind arcs of tracer sampling units. The sampling units were positioned 2-6 km from the point of release.

The site was mainly residential with a roughness length of 0.6 m. The available data (see Table 1) were used to create the input for the simulation. The profiles of wind standard deviations and the Lagrangian decorrelation time scale were calculated according the equations (11) and (12), respectively. Wind speeds at 10 and 115 meters were used to calculate the coefficient for the exponential wind vertical profile, which is used to obtain the wind speed as follows:

$$\gamma = \left[ \frac{\log(U(115)/U(10))}{\log(115/10)} \right] \quad (19a)$$

$$U(z) = U(10) \left[ \frac{z}{10} \right]^\gamma, \quad (19b)$$

where  $U(10)$  is the wind speed at 10 m and  $U(115)$  is the wind speed at 115 m.

Parameters	Numerical value
$-L$ (m)	292
$h$ (m)	1920
$u_*$ ( $\text{ms}^{-1}$ )	0.73
$U$ 10 m ( $\text{ms}^{-1}$ )	4.9
$U$ 115 m ( $\text{ms}^{-1}$ )	10.6
$Q$ ( $\text{gs}^{-1}$ )	3.2

**Table 1. Meteorological parameters during the Copenhagen experiment**

In LAMBDA code, the horizontal domain was determined according to sampler distances and the vertical domain was set equal to the observed mixing height. The time step was maintained constant and it was obtained according to the value of the Lagrangian decorrelation time scale ( $\Delta t = \tau_{L_i} / c$ ), where  $\tau_{L_i}$  must be the smaller value between  $\tau_{L_u}, \tau_{L_v}, \tau_{L_w}$  and  $c$  is an empirical coefficient set equal to 10. Two thousand particles were released in each time step during 500 time steps. The PDF Gram-Charlier truncated to the fourth order was chosen [1, 8].

For the simulation a wind field is necessary for computing the pollutant dispersion. In the example presented here an empirical relation for determining wind field is used. However, the wind field could be given from a meso-scale meteorological model [8], given a more accurate result.

The simulation is run with a main wind blowing on the direction- $x$  over a flat terrain for a point source emitting 2000 particles per second. Final results for time integration of the Lambda model are shown in Figures 1a-1d, ranging from 1 up to 10 processors. The similarity among simulations is indicating that all simulations are carried out successfully.

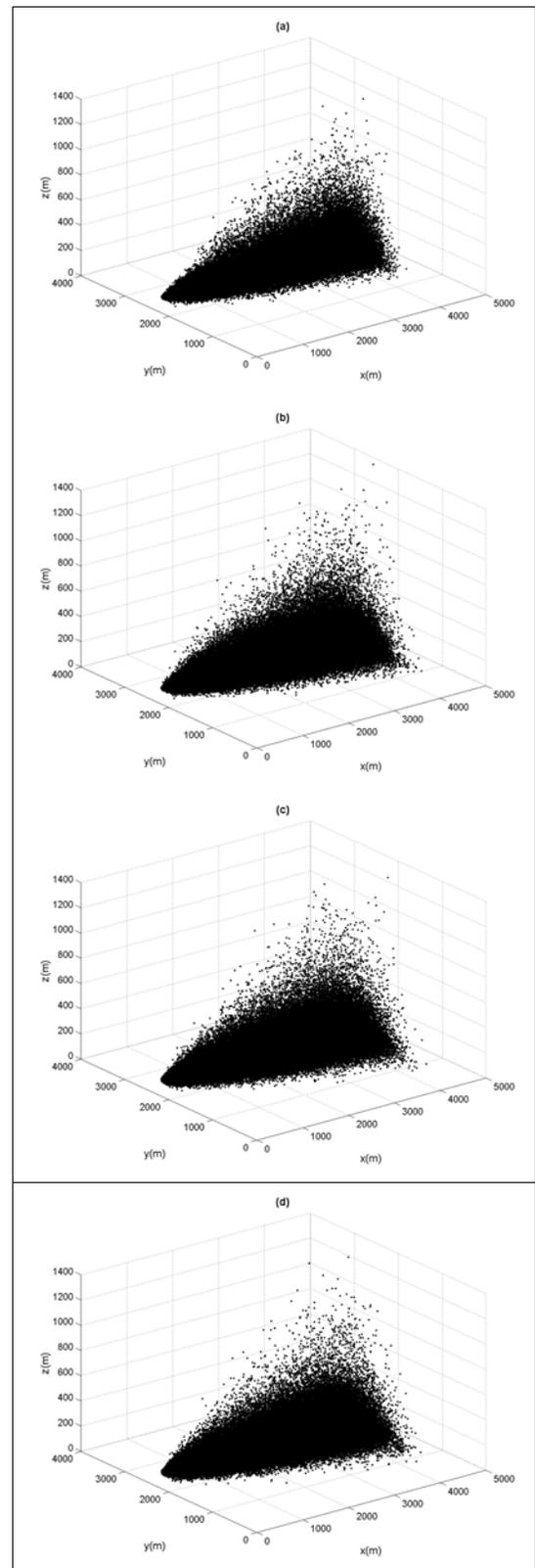
The little disagreement among the solutions displayed in the Figure 1 is due to the stochastic nature of the Lagrangian models, as a consequence of pseudo-random numbers generated during the execution. But, in terms of pollutant concentrations, the values are pretty similar, i.e., the parallelization does not affect the result, as expected.

Observing Table-2, one can realize that the parallelization was effective, presenting a good performance for the parallel code for this complex computational simulator – where  $p$  and  $seq.$  denote the number of processors and the sequential execution time, respectively. Figure 2 also shows the CPU-time according to the number of processors.

This good parallel performance could be predicted observing the corresponding sequential code profiling, like the partial gprof/Linux call graph profile shown in Figure 3. Almost all computing processing of generation and particles dispersion is done by the function `lambda` and its children routines.

$p$	Time(sec.)	Speed-up	Efficiency
Seq.	174.43	-	-
2	88.27	1.98	0.99
5	40.07	4.35	0.87
10	24.85	7.02	0.70

**Table 2. Speed-up and efficiency for  $p$  processors**



**Figure 1. Dispersion: (a) sequential, (b) 2 processors, (c) 5 processors, and (d) 10 processors.**

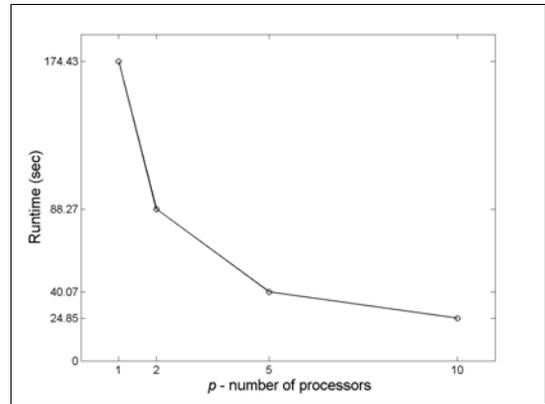
The corresponding percentage of 81.9%, is not as greater as expected because were identified another 19 “spontaneous” type functions, that spent 31.55 seconds, or 18.1% of the total execution time (174.43 seconds). Parents of this kind of function can not be determined. Therefore, 81.9% means the whole effective remaining processing time of this profiling concerns to `lambda` function and its subsequent routines. Consequently, it is supposed that the sequential part of the respective parallel code, will not contribute expressively for the efficiency decreasing with the number of processors employed. It occurs mainly due to the communication overhead of the `MPI_REDUCE` function at each step time to join the particles, separated in each processor.

In order to give a support to this affirmation, in Table-3 can be seen parallel execution time, subtracted to the reduction operation execution time for each number of processor. For 2, 5 and 10 processors, the communication overhead of reduction operation was 2.39, 4.51 and 6.09 seconds, respectively. Note that the efficiency now is superlinear with two processors, close to linear using five and, using ten processors, the efficiency increased a lot compared with Table-2.

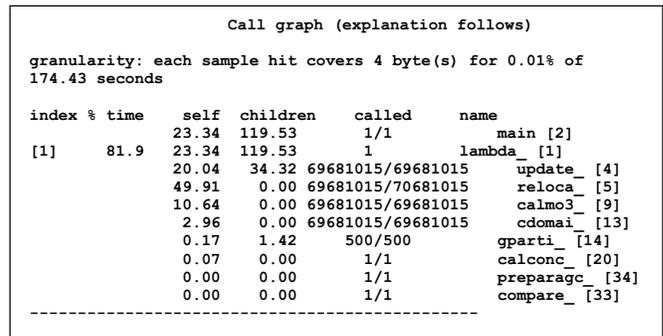
<i>p</i>	Time(sec.)	Speed-up	Efficiency
Seq.	174.43	-	-
2	85.88	2.03	1.02
5	35.56	4.91	0.98
10	18.76	9.30	0.93

**Table 3. Speed-up and efficiency for *p* processors, supposing absence of reduction operation overhead**

The load balancing is not necessary for the present strategy, because the number of particles released is fixed at the 2000, producing a perfect balance during all simulations. This represents an additional advantage for this parallel implementation.



**Figure 2. Processing time versus number of processors (*p*).**



**Figure 3. Partial gprof/Linux call graph profile.**

## 5. Final Remarks

An important issue in parallel programming is to maximize the amount of computation done by each processor and to minimize the amount of communication, due in this case to MPI calls, in order to achieve good performance. This is particularly important in multicomputers as the communication latency is relatively high. Therefore, a pollutant dispersion model running independently from a complex meso-scale meteorological simulator will not have the same difficulties, as a dynamic load balancing.

The distribution of the virtual particles among the processors allows good load balancing and requires a small amount of communication. Therefore, good values of speed-up and efficiency are achieved in the parallel implementation up to 10 processors. The parallel processing was nearly linear for 2 processors (reduction of 50% in processing time), but the performance was not maintained when more processors were considered. The time processing was improved 55% (related to the 2 processors) and 38% (comparing

to 5 processors), speed-up 4.35 and 7.02 respectively. As the runtime of the sequential part of the parallel code is not relevant, most of the efficiency decreasing may be associated with the communication overhead of the particles reduction operation, at each step of time loop.

This work has shown that a cost effective architecture standard software tools can be successfully employed to efficiently solve the Lagrangian atmospheric dispersion models.

## Acknowledgements

The authors would like to thank to the Brazilians and Italian financial support agencies, becoming possible this scientific cooperation. The first author also thanks to the LAC-INPE by the technical visit period, where this work was done. Authors thanks S. Stephany and A. J. Preto, both LAC-INPE researchers, by the support in the parallel machine instructions and also thanks FAPESP, The State of São Paulo Research Foundation, for the support given to this study through a Research Project grant (process 01/03100-9). Author R.P. Souto acknowledges financial support by CNPq, the Brazilian Council for Scientific and Technological Development.

## References

[1] D. Anfossi, and E. Ferrero, "Comparison among Empirical Probability Density Functions of the Vertical Velocity in the Surface Layer Based on Higher Order Correlations", *Boundary-Layer Meteorol.*, **82**, pp. 193-218, 1997.

[2] J.H. Baerentsen, R. Berkowicz (1984): "Monte Carlo simulation of plume dispersion in the convective boundary layer", *Atmos. Environ.* **18**, 701-712, 1984.

[3] A.K. Blackadar, *Turbulence and Diffusion in the Atmosphere*. Springer Verlag, 1997

[4] A.K. Blackadar, "The vertical distribution of wind and turbulent exchange in a neutral atmosphere", *J. Geophys. Res.*, **67**, pp. 3095-3102, 1982

[5] S.J. Caughey, and S.G. Palmer, "Some aspects of turbulence structure through the depth of the convective boundary layer", *Quart. J. Roy. Meteor. Soc.*, **105**, pp. 811-827, 1979.

[6] G.A. Degrazia, and O.L.L. Moraes, "New model for eddy diffusivity in a stable boundary layer", *Boundary-Layer Meteorol.*, **58**, 205-221, 1992.

[7] G.A. Degrazia, "Modelling dispersion from elevated sources in a planetary boundary layer dominated by moderate convection", *Nuovo Ciment.*, **21**(C), pp. 345-358, 1998.

[8] G.A. Degrazia, D. Anfossi, J.C. Carvalho, C. Mangia, T. Tirabassi, and H.F. Campos Velho, "Turbulence parameterization for PBL dispersion models in all stability conditions", *Atmos. Environ.* **34**, pp. 3575-3583, 2000.

[9] Y.A. Delage, Numerical Study of the Nocturnal Atmospheric Boundary Layer, *Quart. J. Roy. Meteor. Soc.* **100**, pp. 351-364, 1974

[10] E. Ferrero, D. Anfossi, G. Brusasca, (1995): Lagrangian particle model LAMBDA: evaluation against tracer data, *Int. J. Environment and Pollution*, **5**, pp. 360-374, 1995

[11] E. Ferrero, and D. Anfossi, "Sensitivity analysis of Lagrangian Stochastic models for CBL with different PDF's and turbulence parameterizations", in: *Air Pollution Modelling and its Applications XII*, S.E. Gryning and N. Chaumerliac eds., Plenum Press, New York, pp. 673-680, 1998a.

[12] E. Ferrero, and D. Anfossi, "Comparison of PDFs, closures schemes and turbulence parameterizations in Lagrangian Stochastic Models", *Int. J. Environ. and Pollution* **9**, pp. 384-410, 1998b.

[13] U. Frisch. 1995, *Turbulence*, Cambridge University Press.

[14] S.E. Gryning, and E. Lyck, "Atmospheric dispersion from elevated source in an urban area: comparison between tracer experiments and model calculations", *Journal of Climate Applied Meteorology*, **23**, pp. 651-654, 1984.

[15] S.E. Gryning, and E. Lyck, The Copenhagen tracer experiments: reporting of measurements. RIS National Laboratory, ISBN 87-550-2395-9. 54 p (only available in an electronic version at: <http://www.risoe.dk/rispubl/VEA/ris-r-1054.htm>), 1998.

[16] S.R. Hanna, "A method of estimation vertical eddy transport in the planetary boundary layer using characteristics if the vertical velocity spectrum", *J. Atmos. Sci.*, **25**, pp. 1026-1032, 1968.

[17] S.R. Hanna, "Lagrangian and Eulerian time-scale in the daytime boundary layer", *J. Appl. Meteorol.*, **20**, pp. 242-249, 1981.

[18] S.R. Hanna, "Applications in air pollution modeling", *Atmospheric turbulence and air pollution modeling*, F.T.M. Nieuwstad and H. von Dop, Eds, D. Reidel Publishing, pp. 275-310, 1982.

[19] J.C. Kaimal, J.C. Wyngaard, D.A. Haugen, O.R. Cote, Y. Izumi, S.J. Caughey, and C.J. Readings, 1976: "Turbulence structure in the convective boundary layer", *J. Atmos. Sci.*, **33**, pp. 2152-2169, 1976.

[20] MPI Forum, MPI: a message-passing interface standard, *International Journal of Supercomputer Applications*, **8**(3-4), 1994.

- [21] F.T. Nieuwstadt, "The Turbulent Structure of the Stable Nocturnal Boundary Layer", *J. Atmos. Sci.* **41**, pp. 2002-2216, 1984
- [22] H.R. Olesen, S.E. Larsen and J. Hojstrup: "Modelling velocity spectra in the lower part of the planetary boundary layer", *Boundary-Layer Meteorol.*, **29**, pp. 285-312, 1984.
- [23] P. Pacheco, *Parallel Programming with MPI*. Morgan Kaufmann Publishers, San Francisco, USA, 1996.
- [24] H.C. Rodean, *Stochastic Lagrangian models of turbulent diffusion*. Boston: American Meteorology Society, 1996.
- [25] Z. Sorbjan, *Structure of the Atmospheric Boundary Layer*, Prentice Hall, EnglewoodCliffs, NJ, 317 pp., (Sorbjan) 1996.
- [26] R.B. Stull, *An Introduction to Boundary Layer Meteorology*. Kluwer Academic Publishers, 1988.
- [27] H. Tennekes, "Similarity relation, scaling laws and spectral dynamics", *Atmospheric turbulence and Air pollution Modeling* (Editors: F.T.M. Nieuwstadt and H. von Dop), pp. 37-68, 1982.
- [28] D.J. Thomson, "Criteria for the selection of stochastic models of particle trajectories in turbulent flows", *J. Fluid Mech.*, **180**, pp. 529-556, 1987.
- [29] J.C. Wyngaard, O.R. Cote, K.S. Rao, Modeling of the atmospheric Boundary Layer. – *Adv. In Geophys.* **18(A)**, pp. 193-212, 1974.
- [30] P. Zanetti, *Air Pollution Modeling*, Computational Mechanics Publications, UK, 1990.