

## Parallelization of the Lagrangian Particle Dispersion Model

by

R. L. Buckley

Westinghouse Savannah River Company  
Savannah River Site  
Aiken, South Carolina 29808

B. L. O'Steen

**MASTER**

DOE Contract No. **DE-AC09-96SR18500**

---

This paper was prepared in connection with work done under the above contract number with the U. S. Department of Energy. By acceptance of this paper, the publisher and/or recipient acknowledges the U. S. Government's right to retain a nonexclusive, royalty-free license in and to any copyright covering this paper, along with the right to reproduce and to authorize others to reproduce all or part of the copyrighted paper.

**DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED**

# PARALLELIZATION OF THE LAGRANGIAN PARTICLE DISPERSION MODEL (U)

Robert L. Buckley and B. Lance O'Steen

Technical Reviewer

Dan P. Griggs Date 8/28/97  
D. P. Griggs

Approval

Robert Addis Date 8/28/97  
R. P. Addis, Manager, Atmospheric Technologies Group

A. L. Boni Date 8-29-97  
A. L. Boni, Manager, Nonproliferation Technologies Section

August 1997

Westinghouse Savannah River Company  
Savannah River Site  
Aiken, SC 29808



SAVANNAH RIVER SITE

PREPARED FOR THE U.S. DEPARTMENT OF ENERGY UNDER CONTRACT NO. DE-AC09-96SR18500

## DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from the Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, TN 37831; prices available from (615) 576-8401.

Available to the public from the National Technical Information Service, U.S. Department of Commerce, 5285 Port Royal Road, Springfield, VA 22161.

## **DISCLAIMER**

**Portions of this document may be illegible electronic image products. Images are produced from the best available original document.**

Key Words

Atmosphere  
Particle dispersion  
Parallel processing

**Retention: Lifetime**

## **PARALLELIZATION OF THE LAGRANGIAN PARTICLE DISPERSION MODEL (U)**

**Robert L. Buckley and B. Lance O'Steen**

**Issued: August 1997**

**SRTC**

**SAVANNAH RIVER TECHNOLOGY CENTER  
AIKEN, SC 29808**  
**Westinghouse Savannah River Company  
Savannah River Site  
Aiken, SC 29808**

---

**PREPARED FOR THE U.S. DEPARTMENT OF ENERGY UNDER CONTRACT NO. DE-AC09-96SR18500**

**ABSTRACT**

An advanced stochastic Lagrangian Particle Dispersion Model (LPDM) is used by the Atmospheric Technologies Group (ATG) to simulate contaminant transport. The model uses time-dependent three-dimensional fields of wind and turbulence to determine the location of individual particles released into the atmosphere. This report describes modifications to LPDM using the Message Passing Interface (MPI) which allows for execution in a parallel configuration on the Cray Supercomputer facility at the SRS. Use of a parallel version allows for many more particles to be released in a given simulation, with little or no increase in computational time. This significantly lowers (greater than an order of magnitude) the minimum resolvable concentration levels without ad hoc averaging schemes and/or without reducing spatial resolution. The general changes made to LPDM are discussed and a series of tests are performed comparing the serial (single processor) and parallel versions of the code.

**TABLE OF CONTENTS**

---

1. INTRODUCTION .....	1
2. GENERAL MODIFICATIONS TO CODE STRUCTURE .....	1
3. BENCHMARK CASES .....	3
A. Input conditions.....	3
B. Results .....	4
4. CONCLUSIONS.....	6
REFERENCES.....	8
APPENDIX .....	15

**LIST OF TABLES**

---

Table 1: Timing comparisons between serial and parallel LPDM versions .....	4
---	---

**LIST OF FIGURES**

---

Figure 1: Flow diagram for parallel version of LPDM .....	9
Figure 2: 'Node' continuation of flow diagram. ....	10
Figure 3: 'Master' continuation of flow diagram.....	11
Figure 4: Integrated surface concentration at 00Z, 25 June 1997 (42 hours after release) over the southeastern United States using the serial version of LPDM. ....	12
Figure 5: Integrated surface concentration at 00Z, 25 June 1997 (42 hours after release) over the southeastern United States using the parallel version of LPDM (using 5 node processors).....	13
Figure 6: Time-dependent concentration at sampler location T (near Tallahassee, Florida) comparing serial and parallel versions. ....	14

# PARALLELIZATION OF THE LAGRANGIAN PARTICLE DISPERSION MODEL (U)

Robert L. Buckley and B. Lance O'Steen

Westinghouse Savannah River Company  
Savannah River Site  
Aiken, SC 29808

## 1. INTRODUCTION

Prognostic atmospheric modeling at the Savannah River Site is performed using the Regional Atmospheric Modeling System (RAMS, Pielke et al. 1992). The Lagrangian Particle Dispersion Model (LPDM, Uliasz 1993) is used to examine the atmospheric transport of passive tracers in the RAMS generated meteorological fields. Tracer particles released from sources of varying number and shape are subject to advective and turbulent disturbances using input from RAMS. Applications using these models include non-proliferation and emergency response consequence assessments.

A serial (single processor) version of LPDM exists, and is currently used operationally on the Cray Supercomputer System at the Savannah River Site. A parallel version of LPDM has now been implemented which allows many more particles to be released for the same computational time. This improves the accuracy and sensitivity of concentration calculations by lowering the minimum concentration level that can be computed without spatial averaging. In addition, error estimates from different realizations of the turbulence are possible in a parallel sequence. Since long-range transport calculations typically lead to low concentrations, increased particle release rates should improve the statistical reliability of these calculations.

A variety of scenarios can be envisioned in which to apply a parallel network to the particle dispersion code. Each processor could be assigned a release from a slightly different source location, and a number of concurrent simulations could be performed to determine the most probable source location. The number of release locations would be limited only by the number of available processors. In a similar manner, source releases at different times could be performed to determine the most likely time at which a release occurred. Use of LPDM rather than simple one-dimensional, time-invariant Gaussian plume models allows for the inclusion of more realistic diffusion processes often very important near the source release.

In this study, different realizations are made by assuming a different seed number from which to apply random number generation for the turbulent velocity components (Buckley and O'Steen 1997). Since LPDM contains an input parameter allowing for multiple sources, the forward trajectory application could be used here as well, where the different realizations are applied at each source release.

## 2. GENERAL MODIFICATIONS TO CODE STRUCTURE

The serial (single processor) version of LPDM is contained essentially within a single module. This module consists of a primary subroutine which calls a series of subroutines or functions responsible for emitting and transporting the particles, and outputting particle locations and concentrations. Within the primary routine, memory is first

allocated for the various parameters used in the code (stored contiguously in a large array, 'A'), and a time-loop is established in which a majority of the computations are performed.

The parallel version contains the same series of subroutines as in the serial version except for a revision of the primary subroutine. A flow diagram of the parallel version is shown in Figs. 1 to 3. Boxes with bold borders indicate tasks which were created as a result of parallel processing. Communications between the different processors is accomplished using the Message Passing Interface (MPI, Gropp et al. 1994). MPI is a library of routines created specifically for parallel programming applications. This library is commonly used on a variety of platforms and is used here for the parallelization of LPDM. Note that the Cray Supercomputer at SRS is a shared-memory platform. This implies that a processor assigned to an LPDM simulation may have to share a portion of its computing time with other jobs currently being executed. Logic for the code before the time-loop is encountered is shown in Fig. 1. A short program first establishes which processors are assigned to which machines. Once this designation is made, the machine numbered '0' is assigned to be the 'master' processor, while the remaining machines are assigned to be 'node' processors. Separate subroutines have been created for the 'master' and 'node' processors.

The master subroutine will first read in analysis files (if necessary) and process them for use by the nodes. These large LPDM data files contain information from RAMS at a given fixed time increment (usually 1 hour). The master processor then tells the node processors that the LPDM data file is ready so that each node may begin transport calculations. Each node processor begins reading data from the analysis file. The first node to complete this process sends the time-independent topography, latitude, and longitude arrays associated with the horizontal grid to the master processor to be used for graphical displays.

At this point, the node processors begin the actual transport calculations (Fig. 2). A timestep loop is initiated and each processor releases an identical number of particles which are subject to slightly different random turbulent motions via the input seed value to the random number generator. If it is time to output results, the quickest node first sends the current time to the master. The machine number is then sent, followed by particle attributes (source, location). The master must receive information from all node processors before proceeding, so a signal is sent by the master back to the nodes, telling them they may continue their transport calculations. This implies that the speed of the simulation is dependent on the slowest processor. When all of the nodes complete the calculations, the master is notified by receiving a 'stopping' time.

When the master processor receives the particle attributes associated with all of the nodes (Fig. 3), it calls up routines to produce the National Center for Atmospheric Research (NCAR, Clare and Kennison 1989) graphics. Two types of graphics are created. Particle plots as viewed from both horizontal (plan) and vertical perspectives (looking from south to north) are illustrated at a user-specified increment of time (usually one or two hours). It has been determined that plotting excessively large numbers of particles ( $> 20000$ ) on a frame does not add information to the transport illustration. Thus, the total number of particles plotted on each frame is limited ( $10^2$  to  $10^4$  particles) which also reduces the time needed for graphical output. This is accomplished by using the remainder function in Fortran. For instance, if 50000 particles are released and 5000 particles are plotted, every 10th particle is selected for graphical display. When the particle plots are made, the three-dimensional particle locations are stored in attribute files for additional post-processing if needed.

Contours of concentration at user-specified vertical model levels are also shown for a specified concentration grid and increment of time (which may or may not be the same as for the particle plots). Instantaneous, time-averaged (also user-specified), and integrated (summed over all time since release) concentrations are produced on separate plots for each source. The total number of particles released for all node processors is used in the calculation of concentration. If a time received from one of the node processors is the stopping time, the master will finish program execution.

### 3. BENCHMARK CASES

Making direct comparisons of results (i.e. particle locations, surface concentrations, etc.) between the serial and parallel versions is complicated by the fact that the random number generating routine (RANF) used in calculations of the turbulent velocity components is not guaranteed to produce the same order of random numbers in a multi-tasked environment. This is because the current random number depends on the previously-generated random number. Therefore, there is no guarantee that each task (process) will obtain a distinct, reproducible set of random number values (Cray Programming Environments, 1995). However, if only one node processor is utilized, one-to-one correspondence between the serial and parallel version is expected.

Computing time comparisons must be approached with caution as well. Processors on the SRS Cray are not dedicated. Therefore, if other jobs are active concurrently with the LPDM code, a given node (or the master) may be temporarily stopped while these calculations are being performed. Since the master requires information from all nodes before making concentration and other output calculations, this results in all of the processes waiting on the slowest process to 'catch up' to the others. The simulations described here for the parallel version were performed during times of little or no other user interaction. In addition, the parallel version utilized a Fortran 90 compiler, whereas the older serial version uses a Fortran 77 compiler.

#### *A. Input Conditions*

For the benchmark case, a RAMS simulation is performed over the southeastern United States. The 48-hr simulation begins at 00Z, 23 June 1997. The grid is roughly 1800 km in horizontal extent (with a grid spacing of 60 km) in both the east-west and north-south directions and centered just off the Atlantic coastline near Savannah, Georgia and Jacksonville, Florida (latitude/longitude coordinates of 31°N, 80°W). The vertical grid is terrain-following and stretches from 100 m at the surface to roughly 1 km at the model top (~17 km). Three-dimensional, time-dependent lateral boundary conditions are created using the National Meteorological Center (NMC) aviation and forecast objective weather analysis product (Sela, 1982).

The particle model simulations begin at 06Z, 23 June 1997 (to allow the wind fields in RAMS to 'spin up') and continue for 42 hours. Analysis files of winds and turbulence created in RAMS are ingested into LPDM at hourly increments over this period. In the particle model, an instantaneous 'puff' release for a user-specified number of particles is simulated for a timestep of 60 seconds. The concentration calculations are performed using the 'cell' method of counting particles in a grid volume. This is a discrete method in which the concentration estimate is assumed to be constant throughout the sampling volume (Moran 1992). In this instance, the horizontal extent of the concentration grid is 10 km, while the vertical depth varies with the model height. Results depicted here are

for surface concentrations. An example of the input file containing parameters for use by LPDM is given in the appendix.

Comparisons between the serial and parallel versions of LPDM are performed by examining variations in several different input characteristics: the number of particles released (per source), the number of source release points, the frequency of output, and the number of node processors utilized in the parallel version. The nominal case assumed here is for a single source (located near Miami, Florida at 50 m above ground) with a release of 20000 particles using 5 node processors and displaying output at 2-hr intervals. These values were chosen as likely parameters to be used in an operational mode. Increasing the number of particles released into the domain (either through the release rate or by increasing the number of sources) allows one to see how much time the model spends transporting particles due to advection and turbulence. Multiple sources (located near Tampa Bay and Jacksonville, Florida) imply not only more particles, but more graphical output, as each source contains its own output page (for both particle plots and concentration isopleths). This latter point is also investigated by varying the output frequency. Finally, by increasing the number of node processes, one can gain insight into how much time is spent in passing information between the node and master processes.

### B. Results

Table 1 lists 14 simulations (10 unique) which were made using both the serial and parallel versions of LPDM. In the table header,  $N$  indicates the number of particles released per source,  $S$  indicates the number of source locations, and  $P$  is the number of node processes for the parallel version. (Note that the total number of processors is one more than  $P$  to account for the master).

Table 1: Timing comparisons between serial and parallel LPDM versions

Case	$N$	$P$ (parallel)	$S$	Output Freq. (hr)	CPU $t_S$ (s)	CPU $t_P$ (s)	Concurrent CPU	Ratio ( $t_P/t_S$ )
I.a	200	5	1	2	159.4	287.9	5.60	1.81
I.b	2000	5	1	2	212.6	327.1	5.56	1.54
I.c*	20000	5	1	2	590.9	577.3	5.64	0.98
I.d	50000	5	1	2	1219.1	1172.8	5.84	0.96
II.a	20000	1	1	2	590.9	558.0	1.89	0.94
II.b*	20000	5	1	2	590.9	577.3	5.64	0.98
II.c	20000	7	1	2	590.9	599.7	7.47	1.01
III.a*	20000	5	1	2	590.9	577.3	5.64	0.98
III.b	20000	5	2	2	1086.2	980.9	5.80	0.90
III.c	20000	5	3	2	1587.3	1406.2	5.79	0.89
IV.a*	20000	5	1	2	590.9	577.3	5.64	0.98
IV.b	20000	5	1	6	520.7	566.2	5.76	1.09
IV.c	20000	5	1	12	505.6	551.6	5.82	1.09

The rows in Table 1 marked with asterisks represent the nominal case and are repeated for clarity in the various comparisons. Note that particles released for the parallel version implies the number per process. Thus, the total number of particles released for the parallel versions is represented by  $(N \times P)$ , all of which are used in the concentration

calculations. The output frequency includes plots of particle locations and concentration isopleths and illustrates the effect of input/output (I/O) on overall performance.

The computer processor unit (CPU) time for the serial and parallel versions is given by  $t_s$  and  $t_p$ , respectively. The concurrent CPU column represents the balance efficiency of the parallel simulation. It is the average number of processors which were active during the course of the simulation. Thus, if  $P = 5$  (implies 6 total processors), the best number one could hope for is 6.0. However, there are several places in which the master alone performs calculations, such as establishing the LPDM input data file from which the nodes gather meteorological data, as well as possible output at the final time, after the nodes have all reported in with their information. Sharing the memory with other users during the simulation will also result in a decrease in this number, as all processors must wait upon the slowest processor (Figs. 2 and 3). Note that the product of  $t_s$  and the concurrent CPU yields the total required CPU for all of the processors in the parallel simulation.

From a practical standpoint, the most important column of numbers is the ratio of parallel to serial version CPU. From the Case I results, it is evident that for a small particle release ( $N \leq 2000$ ), the parallel version is slower than its serial counterpart, while for increasingly larger particle releases, the efficiency of the parallel version improves. For small  $N$ , the node processors finish the transport calculations before the master processor has completed graphical output, and hence, the nodes must wait upon the master processor before continuing their calculations.

By increasing the number of processors (Case II), the amount of message passing and I/O is increased, leading to slight slowdowns in  $t_p$ . However, the bottom line in this instance is that a simulation using 140000 particles took only 40 seconds longer to execute than one with 20000 particles (Case II.a). The multiple-source simulations (Case III) reveal much the same information as Case I, in that releasing more particles results in greater parallel efficiency. However, since each source requires a separate series of concentration plots, the parallel version has an even greater advantage over its serial counterpart since the node processors can execute the transport calculations while the master processor generates the extra graphical plots.

The final set of simulations (Case IV) illustrate the independence of the master and node processors. Although there is a slight decrease in CPU time for the parallel version with reduced output frequency, the overall decrease is not as dramatic as for the serial version. Again, in the parallel version, the transport calculations carry on independently of the graphical output calculations (unless so few particles are released that the nodes are forced to wait on the master processor to complete the graphical output). In the majority of cases in Table 1, the ratio of CPU times is at or around unity. It is also worth noting that the concurrent CPU value tends to be higher for large particle releases ( $\sim 50000$ ). This is because less time is spent in the message passing mode; hence, less time is spent waiting on other processors to complete their calculations.

The purpose of releasing more particles is to improve predictions of concentration at low concentration levels. Results from the serial and parallel version for the nominal case ( $N = 20000$ ,  $P = 5$ ,  $S = 1$ ) illustrate this point. Figures 4 and 5 show integrated concentration over the entire 42 hours for the serial and parallel versions, respectively. The map of the southeast United States contains contours of topography (200 m increments, light blue) and concentration contours in red. Three sampler locations are given by the letters 'A', 'T', and 'C'. These figures summarize the major advantages of releasing more particles.

In the serial version (Fig. 4), concentration isopleths near the source (Miami, Florida) are well-defined, but break down at large distances from the source ( $> 500$  km). The patchy appearance of the footprint at lower concentrations is indicative of concentration cells containing very few particles. This is a direct consequence of the discrete nature of the particle-in-cell method of determining concentration. It is likely that only one or two particles are present in each of these sampling volumes. With a small number of particles per cell, the computed concentration is very uncertain. However, for the parallel version (Fig. 5), the cohesiveness of the isopleths is evident throughout, extending to the north and west portions of Alabama. In addition, the lowest predicted concentration for the parallel version is an order of magnitude lower, due to the five-fold increase in the number of particles released during the simulation. As transport time increases and maximum concentration levels decrease, the importance of increased particle release rates becomes more apparent. Thus, the parallel version of LPDM will be most useful for long-range transport simulations ( $\geq 3000$  km).

Time-dependent instantaneous surface concentrations at the sample location T (in northwest Florida) are shown in Fig. 6 for the nominal case (i.e. Case I.c) where sampling is performed at 2-hr intervals. It is evident that the release of more particles in the parallel version results in earlier detection of the source at lower magnitudes. For the results depicted in Figs. 5 and 6, the user should bear in mind that successive simulations using the parallel version will result in a different sequence of random numbers due to the multi-tasked environment. Thus, slight variations in the results for the parallel version for successive simulations can be expected. Nonetheless, the general conclusions remain unchanged. It is clear that releasing more particles into the simulation domain has many advantages.

#### 4. CONCLUSIONS

A parallel version of LPDM has been written using the Message Passing Interface (MPI) on a shared-memory computer system. A 'master' processor coordinates the activities of a user-defined number of 'node' processors. Each node performs transport calculations for a given number of particles, while the master is responsible for graphical output. The difference between node calculations is the initial random seed value assigned in the generation of turbulent velocity components. At user-defined times, information from each node processor pertaining to particle location and source is returned to the master processor.

Timing comparisons between the original serial version and the parallel version indicate successful speedup using the latter, with dependence on several items. It has been determined that balancing the message passing between the master and node processors with the amount of input/output which must be performed is required to obtain good efficiency ( $t_p/t_s \leq 1$ ) in the parallel version. If too few particles are released, then the node processors must wait on the master processor to complete graphical output. Also, since the Cray J90 at SRS is a shared-memory platform, if other users are simultaneously utilizing the computer's memory, then the parallel version efficiency is reduced since calculations are dependent on the slowest processor (i.e. if it is time to produce graphical output, computations must be halted until all node processors have sent information to the master processor). The latter is usually not a problem since current operational use of RAMS and LPDM occurs using batch jobs during the overnight period when memory use is at a minimum.

The use of more particles in a simulation is advantageous because the minimum concentration which can be computed, without ad hoc 'averaging' or kernel methods, is reduced and statistical uncertainty at low concentration levels is decreased.

## REFERENCES

Buckley, R. L. and B. L. O'Steen, 1997: Modifications to the Lagrangian Particle Dispersion Model (U). WSRC-TR-97-0059, Savannah River Site, Aiken SC.

Clare, F. and D. Kennison, 1989: *NCAR Graphics Guide to New Utilities, Version 3.00*. NCAR/TN-341+STR, University Corporation for Atmospheric Research, National Center for Atmospheric Research.

Cray Programming Environments, 1995: *Math Library Reference Manual, SR-2138 2.0*, Cray Research, Inc., pp. 94-99.

Gropp, W., E. Lusk, and A. Skjellum, 1994: *Using MPI: Portable Parallel Programming with the Message-Passing Interface*, The MIT Press, Cambridge, Massachusetts, 307 pp.

Moran, M. D., 1992: *Numerical Modelling of Mesoscale Atmospheric Dispersion*, Ph. D. dissertation, Colorado State University, Department of Atmospheric Science, Paper No. 513, 758 pp.

Pielke R. A., W. R. Cotton, R. L. Walko, C. J. Tremback, W. A. Lyons, L. D. Grasso, M. E. Nicholls, M. D. Moran, D. A. Wesley, T. J. Lee, and J. H. Copeland, 1992: A comprehensive meteorological modeling system--RAMS. *Meteor. Atmos. Phys.*, **49**, 69-91.

Sela, J. G., 1982: *The NMC Spectral Model*. NOAA Technical Report NWS 30. U. S. Department of Commerce, NOAA, Silver Spring, MD.

Uliasz, M., 1993: The atmospheric mesoscale dispersion modeling system. *J. Appl. Meteor.*, **32**, 139-149.

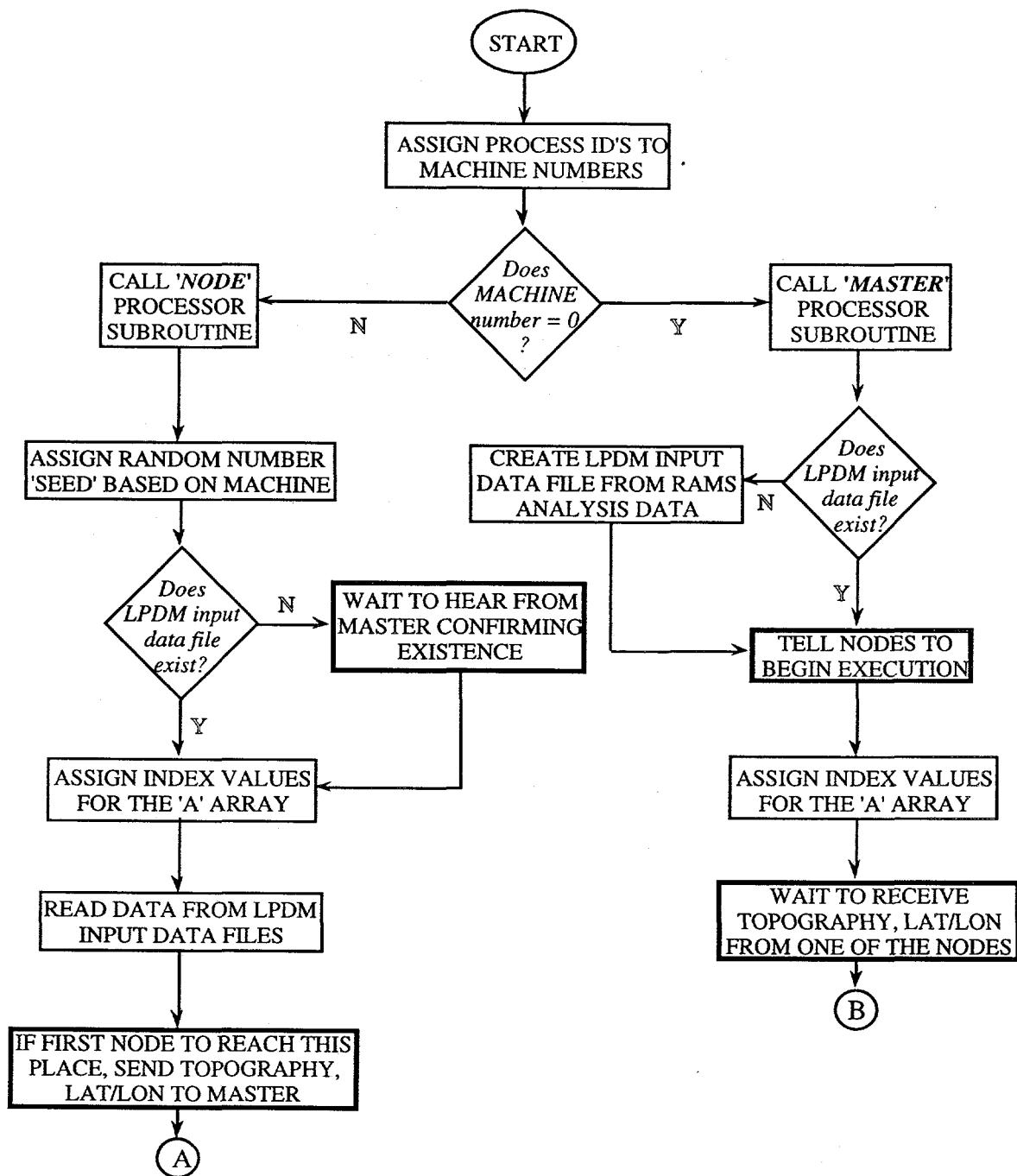


Figure 1: Flow diagram for parallel version of LPDM.

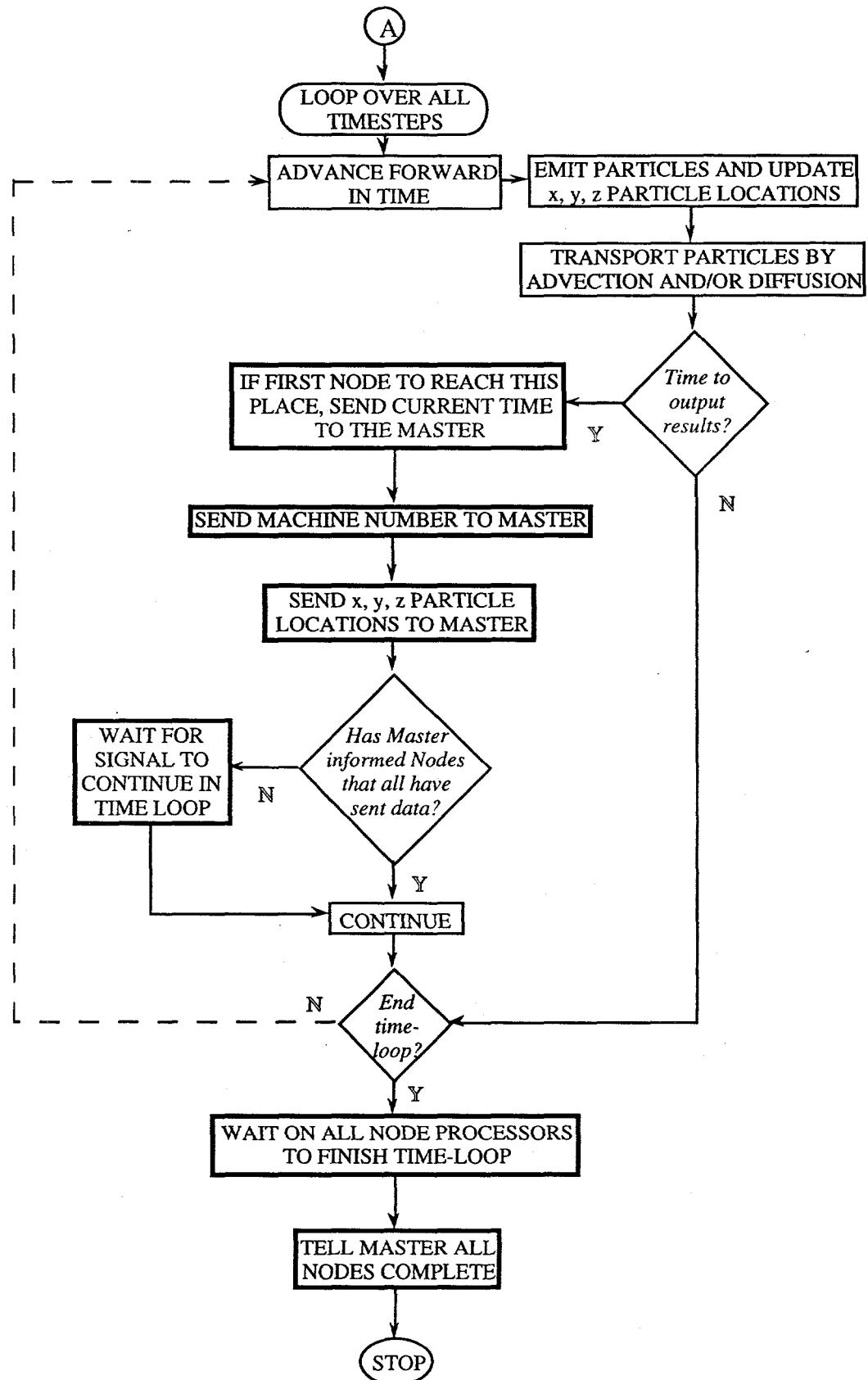


Figure 2: 'Node' continuation of flow diagram

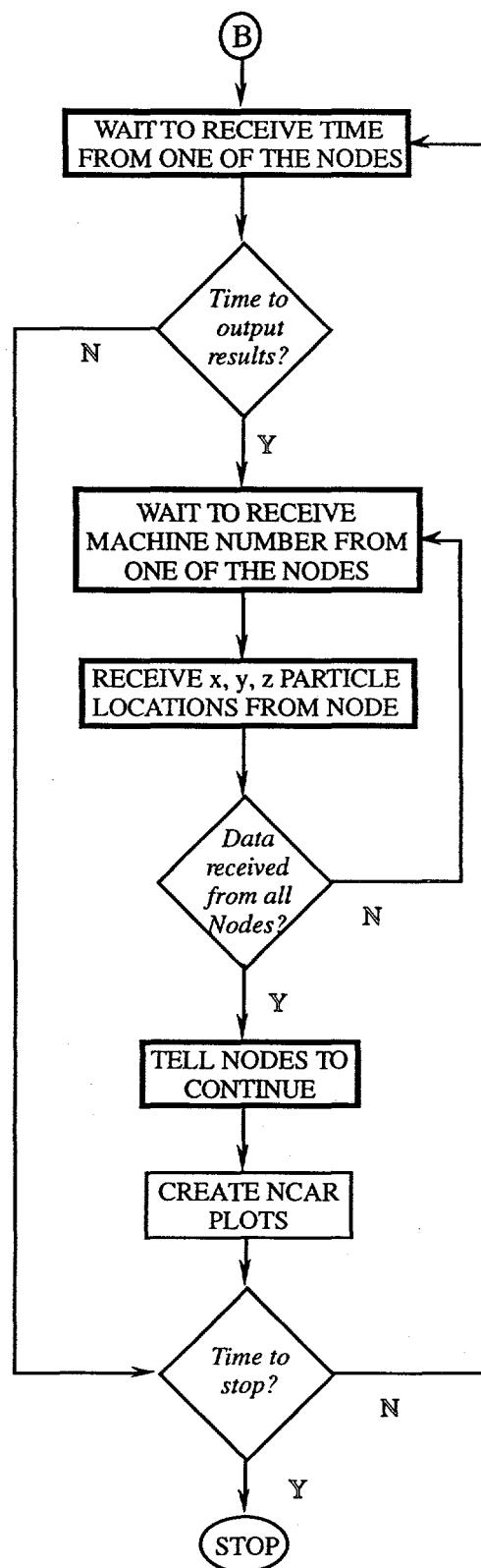
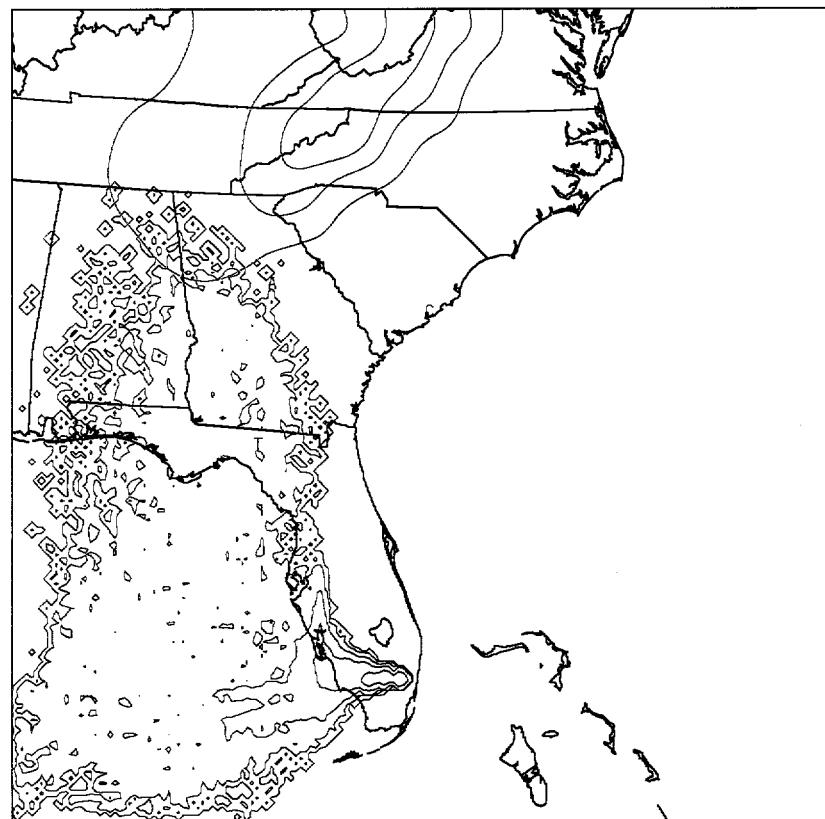


Figure 3: 'Master' continuation of flow diagram

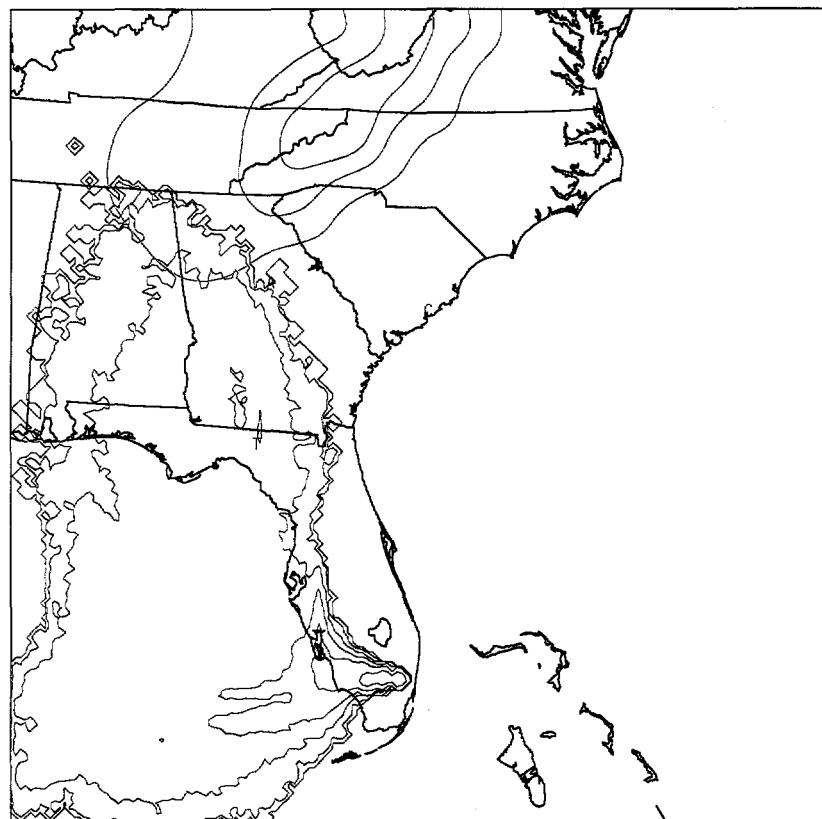
Integrated Surface Concentration  
From Source 01  
Release Time 06Z 06/23/97  
Forecast Time 00Z 06/25/97



Contours -  
from  $0.1000E-04$  ng/m<sup>3</sup>  
to  $0.1000E+00$  ng/m<sup>3</sup>  
by a factor of 10

Figure 4. Integrated surface concentration at 00Z, 25 June 1997  
(42 hours after release) over the southeastern United States  
using the serial version of LPDM.

Integrated Surface Concentration  
From Source 01  
Release Time 06Z 06/23/97  
Forecast Time 00Z 06/25/97



Contours -  
from  $0.1000E-05$  ng/m<sup>3</sup>  
to  $0.1000E+00$  ng/m<sup>3</sup>  
by a factor of 10

Figure 5. Integrated surface concentration at 00Z, 25 June 1997  
(42 hours after release) over the southeastern United States  
using the parallel version of LPDM (using 5 node processors).

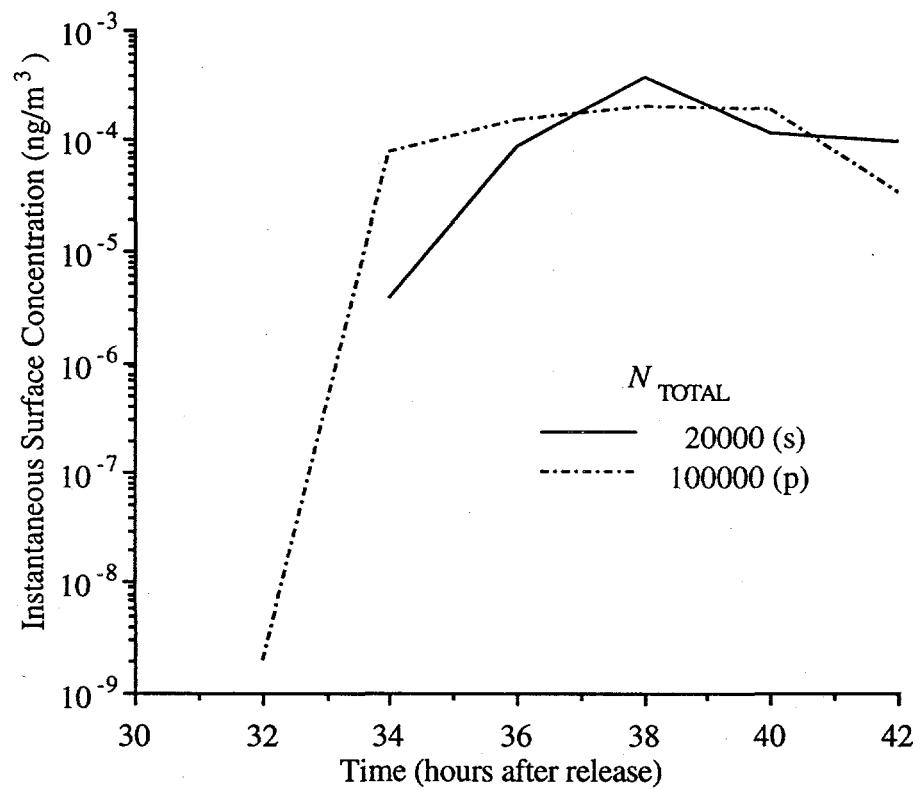


Figure 6: Time-dependent concentration at sampler location T.  
The total number of particles released is given in the legend  
where (s) denotes serial and (p) denotes parallel

## APPENDIX

Example input file used for the LPDM simulations.

```

$CONTROL
  NFILES   = 43,          ! Number of input files
  ANSPEC   = 'AUTO',       ! Type of file spec: 'AUTO' or 'MANUAL'

          ! If AUTO:
  ANPREF  = '/tmp/d0929/rams3a/ps.a',   ! Prefix for the file names
  ANINC   = '60m',          ! Increment of the file names
  ANBEGIN = '360m',         ! Suffix of first file name

          ! If MANUAL:
  FNAME(1) = '/u/smith/myrun/ps.a0m',   ! Input file names
  FNAME(2) = '/u/smith/myrun/ps.a60m',  ! Input file names
  ANATYPE = 'PART',          ! Type of run :
          ! 'PART' - run LPDM
          ! 'SPACE' - all plots have spatial axes
          ! 'TIME' - all plots have a time axis
          ! 'ANIMATE' - prepare Stardent animation files
          ! 'ANIM3D' - prepare Stardent 3-d anim. files
  INFILE   = 'ANAL',         ! Type of input files (FNAMEs)
          ! 'ANAL' - analysis files

$END

$LPDM
  PANNAME = 'LPfile', ! Particle analysis file name
  PHDNAME = 'LPfileh', ! Particle header file name
  POUTNAME = 'LPout',  ! Particle history file name
  NLPSTEP = 2520,      ! Number of timesteps in the current LPDM run
  DTLP   = 60.,        ! Length of particle model timestep in seconds
  FRQLPW = 7200.,      ! Interval (sec) between particle history writes
  LPDIMEN = 3,          ! Number of dimensions in which particles move
  ILPTURB = 1,          ! Turbulence flag: 1=yes, 2=no (traj's only)
  INTSPC  = 0,          ! Spatial interpolation flag: 1=yes, 2=no

  NLPSRC  = 1,          ! Number of particle source regions
  SRCX   = 14.865, 10.0, 10.0, ! X-location of source centers
  SRCY   = 5.771, 10.0, 10.0, ! Y-location of source centers
  SRCZ   = 1.5, 1.5, 1.5,   ! Z-location of source centers
  XSIZE  = 0.0, 0.0, 0.0,   ! X-dimensions of sources
  YSIZE  = 0.0, 0.0, 0.0,   ! Y-dimensions of sources
  ZSIZE  = 0.0, 0.0, 0.0,   ! Z-dimensions of sources
  NXLATT = 1, 1, 1,        ! # of srce lattice pts, X-dir
  NYLATT = 1, 1, 1,        ! # of srce lattice pts, Y-dir
  NZLATT = 1, 1, 1,        ! # of srce lattice pts, Z-dir
  RELSTRT = 21600., 21600., 21600., ! Start time (s), srce release
  RELEND  = 21660., 21660., 21660., ! Stop time (s), srce release
  RELRATE = 20000., 20000., 20000., ! Release rate (#/timestep) for srce
  PHTIME = 86400.0, ! Particle/conc. history write time for restart
  UNITS   = 'GRAMS', ! Conc. units: CURIES (pCi/m3), GRAMS (ng/m3)
  DOSE    = 'N',       ! Calculate dose (Y/N)
  PMRATE = 1.0, 1.0, 1.0, ! Part. mass release rate (mass/sec)
  HALFLIFE = 0., 0., 0., ! Halflife of isotope for dose calc.
  DOSEFAC = 5.0E04, 3.2E04, 5.2E03, ! Dose factor (rem/(Ci/m3)/hr)
  RELTYPE = 'LATTICE', 'LATTICE', 'LATTICE',
          ! Src type ('LATTICE', 'RANDOM')

```

```

ILPFALL = 0,           ! Gravitational settling flag: 1=yes, 0=no
SZMIN   = 1.E-7, 1.E-7, 1.E-7, ! Min particle dia (m) in each srce
SZMAX   = 1.E-7, 1.E-7, 1.E-7, ! Max particle dia (m) in each srce
SZPWR   = 0.33, 0.33, 0.33,     ! Size distrib. power in each srce

CNTYPE  = 'CELL',      ! Concen. algorithm: 'CELL' or 'KERNEL'
FRQCN   = 7200.0,      ! Period (s) between conc. computations
FRQAVE   = 3.,          ! Avg.time for conc.avg.calc as mult. of FRQCN
DXAN    = 0.1667,       ! X,Y,Z-dimensions of concentration
DYAN    = 0.1667,       ! grid cell relative to atmospheric
DZAN    = 1.,           ! model grid cell
ANISW   = 1.0,          ! Location in atmospheric model grid of
ANJSW   = 1.0,          ! western, southern, and bottom
ANKB    = 1.0,          ! boundaries, respectively
ANINE   = 29.0,         ! Location in atmospheric model grid
ANJNE   = 29.0,         ! eastern, northern, and top
ANKT    = 3.0,          ! boundaries, respectively
NCPASS  = 1,            ! Number of passes of concentration filter
CNCOEF  = 0.10,         ! Concentration filter coefficient
CONOUT  = 'n',          ! Concentration output flag: 'Y' =yes
PCONNAME = 'lpconc',   ! Concentration output file name
RELLOC  = 'n',          ! If 1: Plot release locations, otherwise, don't
OBSLOC  = 2,             ! If 1: Read obs.loc's, Calc. conc. @ obs.loc's,
                         ! Write to std. out & Make conc.-vs-time plots
                         ! If 2: Plot obs.loc's on output map plots
MAPDISP = 'Y',          ! If 'Y', put poltcl bnds,topo,coasts in plots
! ISEED   = 5,            ! Random number generator seed value (serial)
TOPO_MIN = 100.,         ! Minimum topography contour level for NCAR (m)
TOPO_MAX = 1000.,        ! Maximum topography contour level for NCAR (m)
TOPO_INC = 100.,          ! Topography contour increment for NCAR (m)
VPLTMAX = 8000.,         ! Max. ht (m) for vertical particle NCAR plots
NLCONC  = 1,              ! Number of vertical levels for concentration
LCONC   = 2,10,13,        ! K-levels to examine for concentration contours
$END

```

**PARALLELIZATION OF THE LAGRANGIAN PARTICLE DISPERSION  
MODEL (U)**

**DISTRIBUTION**

P. Deason, 773-A  
A. L. Boni, 773-A  
R. P. Addis, 773-A  
A. J. Garrett, 773-A  
D. P. Griggs, 773-A  
R. J. Kurzeja, 773-A  
B. L. O'Steen, 773-A  
A. H. Weber, 773-A  
R. L. Buckley, 773-A

SRTC Records(4), 773-52A  
ATG Records(5), 773-A